



Guide du développeur

Amazon Kinesis Video Streams



Amazon Kinesis Video Streams: Guide du développeur

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon sont la propriété de leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

| | |
|--|----|
| Qu'est-ce qu'Amazon Kinesis Video Streams ? | 1 |
| Disponibilité dans les Régions | 2 |
| Comment ça marche | 3 |
| Bibliothèques d'API et de producteurs | 4 |
| Modèle de données | 8 |
| Configuration système requise | 17 |
| Exigences relatives à la caméra | 18 |
| Systèmes d'exploitation testés | 19 |
| Exigences de stockage du SDK | 19 |
| Quotas | 19 |
| Quotas de service API du plan de contrôle | 19 |
| Quotas de service d'API pour les médias et les médias archivés | 23 |
| Fragment-metadata et quotas de médias fragmentés | 29 |
| Quotas du service de métadonnées en streaming | 32 |
| Quotas du SDK pour les producteurs | 32 |
| Stockage hiérarchisé | 36 |
| Présentation de | 36 |
| Concepts clés | 37 |
| Configuration du stockage hiérarchisé | 37 |
| Bonnes pratiques | 37 |
| Configuré un compte | 38 |
| Inscrivez-vous pour un Compte AWS | 38 |
| Créez un Compte AWS clé | 38 |
| Avantages | 38 |
| Prise en main | 40 |
| Création d'un flux vidéo Amazon Kinesis | 40 |
| Création d'un flux vidéo à l'aide de la console | 41 |
| Créez un flux vidéo à l'aide du AWS CLI | 41 |
| Envoyer des données vers un flux vidéo Amazon Kinesis | 41 |
| Créez le SDK et les exemples | 42 |
| Exécutez les exemples pour télécharger du contenu multimédia sur Kinesis Video Streams | 45 |
| Examiner les objets d'accusé de réception | 47 |
| Consommez des données multimédia | 48 |

| | |
|--|----|
| Afficher le contenu multimédia dans la console | 48 |
| Consommez des données multimédia à l'aide du protocole HLS | 48 |
| Transférer vers Kinesis Video Streams | 49 |
| Client producteur Kinesis Video Streams | 49 |
| Bibliothèque destinée aux producteurs de Kinesis Video Streams | 50 |
| Découvrez ce que sont les bibliothèques destinées aux producteurs | 51 |
| Java | 51 |
| Procédure : utiliser le SDK Java Producer | 52 |
| Conditions préalables | 52 |
| Téléchargez et configurez le code | 53 |
| Rédiger et examiner le code | 54 |
| nettoyer des ressources ; | 56 |
| Exécutez et vérifiez le code | 56 |
| Android | 57 |
| Procédure : utilisez le SDK Android Producer | 58 |
| Conditions préalables | 58 |
| Téléchargez et configurez le code | 62 |
| Examinez le code | 63 |
| Exécutez et vérifiez le code | 65 |
| C++ | 67 |
| Modèle d'objet | 67 |
| Insérez du contenu multimédia dans le stream | 67 |
| Interfaces de rappel | 67 |
| Procédure : utiliser le SDK du producteur C++ | 68 |
| Conditions préalables | 68 |
| Téléchargez et configurez le code | 69 |
| Rédiger et examiner le code | 75 |
| Exécutez et vérifiez le code | 82 |
| Utiliser le SDK en tant que plugin GStreamer | 82 |
| Utiliser le SDK C++ Producer en tant que plugin GStreamer dans un conteneur Docker | 82 |
| Utiliser la journalisation | 83 |
| Producteur C | 83 |
| Modèle d'objet | 84 |
| Insérez du contenu multimédia dans le stream | 85 |
| Procédure : utiliser le SDK du producteur C | 85 |
| Conditions préalables | 85 |

| | |
|--|-----|
| Téléchargez le code | 86 |
| Rédiger et examiner le code | 87 |
| Exécutez et vérifiez le code | 90 |
| C++ sur Raspberry Pi | 91 |
| Conditions préalables | 92 |
| Créez un utilisateur | 92 |
| Joignez-vous à votre réseau | 94 |
| Connect à distance | 95 |
| Configuration de la caméra | 96 |
| Installation des prérequis logiciels | 98 |
| Téléchargez et compilez le SDK C++ Producer | 99 |
| Diffusez une diffusion en direct | 102 |
| Lire du contenu multimédia | 113 |
| Résolution des problèmes | 117 |
| Référence du code d'erreur | 121 |
| Erreurs et codes d'état renvoyés par les PutFrame rappels - Code indépendant de la plate- forme (PIC) | 121 |
| Erreurs et codes d'état renvoyés par les PutFrame rappels - bibliothèque C Producer | 175 |
| drapeaux d'adaptation NAL | 182 |
| Structures des producteurs | 184 |
| DeviceInfo/DefaultDeviceInfoProvider | 184 |
| StorageInfo | 184 |
| Structures fluviales | 186 |
| StreamDefinition/StreamInfo | 186 |
| ClientMetrics | 204 |
| StreamMetrics | 205 |
| Rappels | 207 |
| ClientCallbackProvider | 208 |
| StreamCallbackProvider | 208 |
| ClientCallbacks | 209 |
| Implémentations de callback pour réessayer le streaming | 214 |
| Utilisation des métadonnées de streaming | 215 |
| Ajouter des métadonnées à un flux vidéo Kinesis | 216 |
| Utiliser IPv6 avec Kinesis Video Streams | 218 |
| Configurez le AWS SDK pour IPv6 | 218 |
| Configuration du SDK Kinesis Video Streams Producer pour IPv6 | 220 |

| | |
|--|-----|
| Configurez le AWS CLI pour IPv6 | 222 |
| Exemples de configuration | 223 |
| Considérations | 223 |
| Clients mis à niveau pour inclure IPv6 | 224 |
| Résolution des problèmes | 225 |
| Lecture vidéo | 227 |
| Exigences relatives à la lecture | 228 |
| GetClip | 228 |
| GetDASHStreamingSessionURL | 229 |
| GetHLSStreamingSessionURL | 230 |
| GetImages | 231 |
| Lecture avec HLS | 232 |
| Utilisez la commande AWS CLI pour récupérer l'URL d'une session de streaming HLS | 232 |
| Exemple : utilisez HLS en HTML et JavaScript | 236 |
| Résolution des problèmes liés au HLS | 240 |
| Lecture avec MPEG-DASH | 243 |
| Exemple : utilisation MPEG-DASH en HTML et JavaScript | 243 |
| Configurer les notifications | 248 |
| Gérer les configurations de notifications | 248 |
| UpdateNotificationConfiguration | 248 |
| DescribeNotificationConfiguration | 249 |
| À propos des tags MKV du producteur | 249 |
| Syntaxe pour les tags MKV du producteur | 249 |
| Limites de balises MKV | 249 |
| Messages Amazon SNS | 250 |
| Charge utile de la rubrique Amazon SNS | 250 |
| Afficher vos messages Amazon SNS | 251 |
| Cross-account Publication de notifications Amazon SNS | 252 |
| Identity-based configuration des politiques | 252 |
| Resource-based configuration des politiques | 253 |
| Exigences et considérations | 253 |
| Extraire des images à partir de flux vidéo | 254 |
| Real-time génération d'images | 254 |
| Ajouter des balises de génération d'images aux fragments | 257 |
| Chemin d'accès à l'objet Amazon S3 (image) | 260 |
| Récupération des métadonnées d'image | 260 |

| | |
|---|-----|
| Recommandations relatives aux URI Amazon S3 pour se protéger contre le ralentissement | 261 |
| Résolution des problèmes | 262 |
| Accédez à l'analyse vidéo | 265 |
| Consommer des métadonnées | 265 |
| Diffuser à l'aide de la bibliothèque d'analyseurs | 266 |
| Conditions préalables | 267 |
| Téléchargez le code | 267 |
| Examinez le code | 268 |
| Exécutez le code | 275 |
| Contrôle | 275 |
| Surveillez les métriques avec CloudWatch | 276 |
| Surveillez l'agent Amazon Kinesis Video Streams Edge avec CloudWatch | 296 |
| Enregistrez les appels d'API avec CloudTrail | 301 |
| Limites de métadonnées de diffusion | 307 |
| Planifier l'enregistrement et le stockage vidéo | 308 |
| Opérations de l'API Amazon Kinesis Video Streams Edge Agent | 309 |
| Surveillance de l'agent Amazon Kinesis Video Streams Edge | 309 |
| Déploiement en AWS IoT Greengrass mode non | 309 |
| Installation des dépendances | 310 |
| Créez des ressources pour les URL RTSP de vos caméras IP | 312 |
| Création d'une politique d'autorisations IAM | 314 |
| Créer un rôle IAM | 316 |
| Création de l'alias de AWS IoT rôle | 317 |
| Création de la AWS IoT politique | 318 |
| Créez AWS IoT quelque chose et obtenez des AWS IoT Core informations d'identification .. | 320 |
| Création de l'agent Edge | 323 |
| Installation de l' CloudWatch agent | 334 |
| Exécuter l'agent Edge en tant que processus natif | 337 |
| Déployer vers AWS IoT Greengrass | 339 |
| Création d'une instance Ubuntu | 340 |
| Configuration de l'appareil AWS IoT Greengrass principal | 342 |
| Créez des ressources pour les URL RTSP de vos caméras IP | 343 |
| Ajouter des autorisations au rôle TES | 345 |
| Installation du composant Secret Manager | 348 |
| Déployer l'agent Edge sur l'appareil | 351 |

| | |
|--|-----|
| Installation du composant du gestionnaire de AWS IoT Greengrass journaux | 360 |
| FAQ | 364 |
| Quels sont les systèmes d'exploitation pris en charge par Amazon Kinesis Video Streams | |
| Edge Agent ? | 365 |
| L'agent Amazon Kinesis Video Streams Edge est-il compatible avec les médias H.265 ? ... | 365 |
| L'agent Amazon Kinesis Video Streams Edge fonctionne-t-il dans AL2 ? | 365 |
| Comment puis-je exécuter plusieurs flux dans le AWS IoT objet ou appareil ? | 365 |
| Comment puis-je modifier un document une StartEdgeConfigurationUpdatefois qu'il a été envoyé ? | 366 |
| Avez-vous des exemples courants ScheduleConfigs? | 366 |
| Y a-t-il une limite de diffusion maximale ? | 367 |
| Comment puis-je redémarrer une tâche qui a donné lieu à une erreur ? | 367 |
| Comment puis-je surveiller l'état de santé de mon agent Amazon Kinesis Video Streams Edge ? | 367 |
| Diffusez des vidéos via un VPC | 368 |
| Informations supplémentaires | 368 |
| Procédures relatives aux points de terminaison VPC | 369 |
| Exemples | 371 |
| Exemples : envoi de données vers Kinesis Video Streams | 371 |
| Exemples : extraction de données depuis Kinesis Video Streams | 371 |
| Exemples : lecture de données vidéo | 371 |
| Conditions préalables | 371 |
| Plug-in GStreamer - kvssink | 372 |
| Téléchargez, compilez et configurez l'élément GStreamer | 373 |
| Exécutez l'élément GStreamer | 373 |
| Commandes de lancement | 374 |
| Exécutez l'élément GStreamer dans un conteneur Docker | 376 |
| Référence des paramètres | 379 |
| PutMedia API | 394 |
| Téléchargez et configurez le code | 395 |
| Rédiger et examiner le code | 396 |
| Exécutez et vérifiez le code | 398 |
| RTSP et Docker | 399 |
| Didacticiels vidéo | 399 |
| Conditions préalables | 400 |
| Créez l'image Docker | 400 |

| | |
|---|-----|
| Exécutez l'exemple d'application RTSP | 401 |
| Moteur de rendu | 402 |
| Conditions préalables | 403 |
| Exemple d'exécution du moteur de rendu | 403 |
| Comment ça marche | 404 |
| Sécurité | 406 |
| Protection des données | 407 |
| Qu'est-ce que le chiffrement côté serveur pour Kinesis Video Streams ? | 407 |
| Considérations relatives aux coûts, aux régions et aux performances | 407 |
| Comment démarrer avec le chiffrement côté serveur ? | 408 |
| Création et utilisation d'une clé gérée par le client | 409 |
| Autorisations d'utilisation d'une clé gérée par le client | 410 |
| Contrôle de l'accès aux ressources Kinesis Video Streams à l'aide d'IAM | 412 |
| Syntaxe d'une politique | 412 |
| Actions relatives à Kinesis Video Streams | 413 |
| Amazon Resource Names (ARN) pour Kinesis Video Streams | 414 |
| Autoriser d'autres comptes IAM à accéder à un flux vidéo Kinesis | 414 |
| Exemples de stratégies | 418 |
| Contrôle de l'accès aux ressources Kinesis Video Streams à l'aide de AWS IoT | 420 |
| AWS IoT ThingName comme nom de flux | 421 |
| AWS IoT CertificateId comme nom de flux | 428 |
| Utilisation AWS IoT informations d'identification pour diffuser vers un nom de flux codé en dur | 429 |
| Validation de la conformité | 431 |
| Résilience | 431 |
| Sécurité de l'infrastructure | 431 |
| Bonnes pratiques de sécurité | 432 |
| Implémentation d'un accès sur la base du moindre privilège | 432 |
| Utilisation des rôles IAM | 433 |
| CloudTrail À utiliser pour surveiller les appels d'API | 433 |
| Résolution des problèmes | 434 |
| Problèmes généraux | 434 |
| Latence trop élevée | 434 |
| Problèmes liés à l'API | 435 |
| Erreur : « Options inconnues » | 435 |
| Erreur : « Impossible de déterminer service/operation le nom à autoriser » | 435 |

| | |
|--|-----|
| Erreur : « Failed to put a frame in the stream » (Échec du placement d'une trame dans le flux) | 436 |
| Erreur : « Le service AckEvent a fermé la connexion avant la réception de la version finale » | 436 |
| Erreur : « STATUS_STORE_OUT_OF_MEMORY » | 436 |
| Erreur : « Les informations d'identification doivent être limitées à une région valide. » | 437 |
| Problèmes liés au HLS | 437 |
| Problèmes liés à Java | 437 |
| Activation des journaux Java | 437 |
| Problèmes liés à la bibliothèque des producteurs | 438 |
| Impossible de compiler le SDK du producteur | 439 |
| Un flux vidéo n'apparaît pas dans la console | 439 |
| Erreur : « Security token included in the request is invalid » (le jeton de sécurité inclus dans la requête n'est pas valide) lors de la diffusion de données à l'aide de l'application de démonstration GStreamer | 440 |
| Erreur : « Failed to submit frame to Kinesis Video client » (Échec d'envoi d'image au client Kinesis Video) | 441 |
| L'application GStreamer s'arrête avec le message « streaming arrêté, motif non négocié » sur OS X | 441 |
| Erreur : « Failed to allocate heap » (échec de l'allocation du tas) lors de la création du Client vidéo Kinesis dans la démonstration de GStreamer sur Raspberry Pi | 442 |
| Erreur : « Illegal Instruction » (instruction non conforme) lors de l'exécution de la démonstration de GStreamer sur Raspberry Pi | 442 |
| Le chargement de la caméra échoue sur Raspberry Pi | 442 |
| La caméra n'est pas trouvée sur macOS High Sierra | 443 |
| Fichier jni.h introuvable lors de la compilation sur macOS High Sierra | 443 |
| Erreurs curl lors de l'exécution de l'application de démonstration GStreamer | 443 |
| Timestamp/range assertion lors de l'exécution sur Raspberry Pi | 443 |
| Assertion sur gst_value_set_fraction_range_full sur Raspberry Pi | 443 |
| 0x3200000d STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (erreur sur Android) | 444 |
| La durée maximale du fragment a été atteinte (erreur) | 444 |
| Message d'erreur « Invalid thing name passed » (Nom d'objet non valide attribué) lors de l'utilisation d'une autorisation IoT | 445 |
| Problèmes liés à la bibliothèque d'analyseurs de flux | 445 |
| Impossible d'accéder à une seule image depuis le flux | 445 |

| | |
|--------------------------------------|--------|
| Erreur de décodage de fragment | 446 |
| Problèmes liés au réseau | 446 |
| Historique de la documentation | 448 |
| | cdliii |

Qu'est-ce qu'Amazon Kinesis Video Streams ?

Vous pouvez utiliser Amazon Kinesis Video Streams, une solution entièrement Service AWS gérée, pour diffuser des vidéos en direct depuis des appareils vers AWS Cloud le, ou créer des applications pour le traitement vidéo en temps réel ou l'analyse vidéo par lots.

Kinesis Video Streams ne se limite pas au stockage de données vidéo. Vous pouvez l'utiliser pour surveiller vos flux vidéo en temps réel au fur et à mesure de leur réception dans le cloud. Vous pouvez soit surveiller vos diffusions en direct dans le AWS Management Console, soit développer votre propre application de surveillance qui utilise la bibliothèque d'API Kinesis Video Streams pour afficher des vidéos en direct.

Vous pouvez utiliser Kinesis Video Streams pour capturer d'énormes quantités de données vidéo en direct provenant de millions de sources, notamment des smartphones, des caméras de sécurité, des webcams, des caméras embarquées dans des voitures, des drones et d'autres sources. Vous pouvez également envoyer des données chronologiques non vidéo, telles que des données audio, des images thermiques, des données de profondeur et des données RADAR. Lorsque vous diffusez des vidéos en direct depuis ces sources vers un flux vidéo Kinesis, vous pouvez créer des applications permettant d'accéder aux données, image par image, en temps réel pour un traitement à faible latence. Kinesis Video Streams fonctionne indépendamment de la source. Vous pouvez diffuser des vidéos à partir de la webcam d'un ordinateur à l'aide de la [Plug-in GStreamer - kvssink](#) bibliothèque ou à partir d'une caméra de votre réseau à l'aide du protocole de diffusion en temps réel (RTSP).

Vous pouvez également configurer votre flux vidéo Kinesis pour stocker durablement les données multimédia pendant la période de conservation spécifiée. Kinesis Video Streams stocke automatiquement ces données et les chiffre au repos. En outre, Kinesis Video Streams indexe temporellement les données stockées en fonction à la fois de l'horodatage du producteur et de l'horodatage d'ingestion. Vous pouvez créer des applications qui traitent régulièrement les données vidéo par lots, ou vous pouvez créer des applications qui nécessitent un accès unique aux données historiques pour différents cas d'utilisation.

Vos applications personnalisées, en temps réel ou orientées par lots, peuvent s'exécuter sur des instances Amazon EC2. Ces applications peuvent traiter les données à l'aide d'algorithmes d'apprentissage en profondeur open source ou utiliser des applications tierces intégrées à Kinesis Video Streams.

Disponibilité dans les Régions

Amazon Kinesis Video Streams est disponible dans les régions suivantes :

| Nom de la région | AWS Code de région |
|----------------------------|--------------------|
| USA Est (Ohio) | us-east-2 |
| USA Est (Virginie du Nord) | us-east-1 |
| USA Ouest (Oregon) | us-west-2 |
| AWS GovCloud (US-East) | us-gov-east-1 |
| AWS GovCloud (US-West) | us-gov-west-1 |
| Afrique (Le Cap) | af-south-1 |
| Asie-Pacifique (Hong Kong) | ap-east-1 |
| Asie-Pacifique (Mumbai) | ap-south-1 |
| Asia Pacific (Seoul) | ap-northeast-2 |
| Asie-Pacifique (Singapour) | ap-southeast-1 |
| Asie-Pacifique (Sydney) | ap-southeast-2 |
| Asie-Pacifique (Malaisie) | ap-southeast-3 |
| Asie-Pacifique (Tokyo) | ap-northeast-1 |
| Canada (Central) | ca-central-1 |
| China (Beijing) | cn-north-1 |
| Europe (Francfort) | eu-central-1 |
| Europe (Irlande) | eu-west-1 |
| Europe (Londres) | eu-west-2 |

| Nom de la région | AWS Code de région |
|-----------------------------|--------------------|
| Europe (Paris) | eu-west-3 |
| Europe (Espagne) | eu-south-2 |
| Moyen-Orient (Bahreïn) | me-south-1 |
| Amérique du Sud (Sao Paulo) | sa-east-1 |

Kinesis Video Streams : comment ça marche

Rubriques

- [API Kinesis Video Streams et prise en charge des bibliothèques de production](#)
- [Modèle de données Kinesis Video Streams](#)

Vous pouvez utiliser Amazon Kinesis Video Streams, une solution entièrement Service AWS gérée, pour diffuser des vidéos en direct depuis des appareils et AWS Cloud les stocker de manière durable. Vous pouvez ensuite créer vos propres applications de traitement vidéo en temps réel ou effectuer des analyses vidéo de traitement par lot.

Le schéma suivant donne un aperçu du fonctionnement de Kinesis Video Streams.

Le diagramme montre l'interaction entre les composants suivants :

- **Producteur** : toute source qui insère des données dans un flux vidéo Kinesis. Un producteur peut être n'importe quel appareil de génération vidéo, tel qu'une caméra de sécurité, une caméra portée sur le corps, une caméra pour smartphone ou une caméra de tableau de bord. Un producteur peut également envoyer des données non vidéo, par exemple du contenu audio, des images, ou des données RADAR.

Un même producteur peut générer plusieurs flux de données vidéo. Par exemple, une caméra vidéo peut transférer des données vidéo vers un flux vidéo Kinesis et des données audio vers un autre.

- Bibliothèques de production Kinesis Video Streams : ensemble de logiciels et de bibliothèques que vous pouvez installer et configurer sur vos appareils. Vous pouvez utiliser ces bibliothèques pour vous connecter en toute sécurité et diffuser des vidéos de manière fiable de différentes manières, notamment en temps réel, après l'avoir mise en mémoire tampon pendant quelques secondes ou sous forme de téléchargement multimédia après coup.
- Kinesis Video Stream : ressource que vous pouvez utiliser pour transporter des données vidéo en direct, les stocker éventuellement et les mettre à disposition pour qu'elles soient consommées en temps réel, par lots ou en une seule fois. Dans une configuration classique, un seul producteur publie des données dans un flux vidéo Kinesis.

Le flux peut porter de l'audio, de la vidéo et des flux de données avec un timecode similaire, par exemple des flux de détection de profondeur, des flux RADAR et bien plus encore. Vous créez un flux vidéo Kinesis à l'aide des SDK AWS Management Console ou par programmation. AWS

Plusieurs applications indépendantes peuvent consommer un flux vidéo Kinesis en parallèle.

- Consommateur : obtient des données, telles que des fragments et des images, à partir d'un flux vidéo Kinesis pour les visualiser, les traiter ou les analyser. Ces consommateurs sont généralement appelés applications Kinesis Video Streams. Vous pouvez écrire des applications qui consomment et traitent des données dans Kinesis Video Streams en temps réel, ou une fois les données stockées et indexées dans le temps lorsqu'un traitement à faible latence n'est pas requis. Vous pouvez créer ces applications grand public à exécuter sur des instances Amazon EC2.
- [Regardez le résultat des caméras à l'aide de la bibliothèque d'analyseurs](#)— Permet aux applications Kinesis Video Streams d'obtenir du contenu multimédia de manière fiable à partir du flux vidéo Kinesis avec une faible latence. De plus, elle analyse les limites de l'image dans les médias afin que les applications puissent se concentrer sur le traitement et l'analyse des images elles-mêmes.

API Kinesis Video Streams et prise en charge des bibliothèques de production

Kinesis Video Streams fournit des API qui vous permettent de créer et de gérer des flux et de lire ou d'écrire des données multimédia depuis et vers un flux. Outre les fonctionnalités d'administration, la console Kinesis Video Streams prend également en charge la lecture en direct et la lecture de vidéos à la demande. Kinesis Video Streams fournit également un ensemble de bibliothèques de production que vous pouvez utiliser dans le code de votre application pour extraire des données de vos sources multimédia et les télécharger sur votre flux vidéo Kinesis.

Rubriques

- [API Kinesis Video Streams](#)
- [Modèle de découverte des terminaux](#)
- [Bibliothèques pour producteurs](#)

API Kinesis Video Streams

Kinesis Video Streams fournit des API pour créer et gérer des Kinesis Video Streams. Il fournit également des API pour la lecture et l'écriture des données média sur un flux, comme suit :

- **API Producer** — Kinesis Video Streams fournit `PutMedia` une API permettant d'écrire des données multimédia dans un flux vidéo Kinesis. Dans une requête `PutMedia`, le producteur envoie un flux de fragments média. Un fragment est une séquence d'images autonomes. Les images appartenant à un fragment ne doivent avoir aucune dépendance vis-à-vis des images provenant d'autres fragments. Pour de plus amples informations, veuillez consulter [PutMedia](#).

À mesure que les fragments arrivent, Kinesis Video Streams attribue un numéro de fragment unique, par ordre croissant. Il stocke également les horodatages côté producteur et côté serveur pour chaque fragment, sous forme de métadonnées Kinesis Video. Streams-specific

- **API grand public** : les consommateurs peuvent utiliser les API suivantes pour obtenir des données à partir d'un flux :
 - **GetMedia** - Lors de l'utilisation de cet API, les consommateurs doivent identifier le fragment de départ. L'API renvoie ensuite les fragments dans leur ordre d'ajout au flux (en ordre croissant par numéro de fragment). Les données médias dans les fragments sont compressées dans un format structuré, comme le format [Matroska \(MKV\)](#). Pour de plus amples informations, veuillez consulter [GetMedia](#).

Note

`GetMedia` connaît l'emplacement des fragments (archivés dans le magasin de données ou disponibles en temps réel). Par exemple, si `GetMedia` détermine que le fragment de départ est archivé, il commence à renvoyer des fragments à partir du magasin de données. Lorsqu'il doit renvoyer des fragments plus récents qui ne sont pas encore archivés, `GetMedia` passe à la lecture de fragments depuis une mémoire tampon de flux en mémoire.

Ceci est un exemple d'un consommateur continue qui traite les fragments dans l'ordre dans lequel ils sont assimilés par le flux.

`GetMedia` permet aux applications de traitement vidéo d'échouer ou de prendre du retard, puis de rattraper sans aucun effort supplémentaire. Par l'utilisation de `GetMedia`, les applications peuvent traiter les données archivées dans le magasin de données et lorsque l'application rattrape les tâches, `GetMedia` continue d'alimenter les données média en temps réel à leur arrivée.

- `GetMediaFromFragmentList` (et `ListFragments`) - Les applications de traitement par lot sont considérées comme consommateurs hors ligne. Les consommateurs hors ligne peuvent décider de récupérer explicitement certains fragments de médias spécifiques ou certaines plages de vidéo en combinant les API `ListFragments` et `GetMediaFromFragmentList`. `ListFragments` et `GetMediaFromFragmentList` permettent à une application d'identifier les segments d'une vidéo pour une plage de temps ou de fragments particulière, puis de récupérer ces fragments soit de façon séquentielle soit en parallèle pour traitement. Cette approche convient aux gammes d'application MapReduce, qui doit traiter rapidement de grandes quantités de données en parallèle.

Par exemple, supposons qu'un client souhaite traiter les fragments de vidéos d'une journée. Le consommateur doit procéder comme suit :

1. Obtenez une liste des fragments en appelant l'API `ListFragments` et en spécifiant une période pour sélectionner la collection de fragments souhaitée.

L'API renvoie les métadonnées à partir de tous les fragments dans la période spécifiée. Les métadonnées fournissent des informations telles que le numéro du fragment, les horodatages côté producteur et côté serveur, etc.

2. Prenez la liste de métadonnées de fragment et récupérez les fragments, dans n'importe quel ordre. Par exemple, pour traiter tous les fragments de la journée, le consommateur peut choisir de diviser la liste en sous-listes et de demander aux employés (par exemple, plusieurs instances Amazon EC2) de récupérer les fragments en parallèle à l'aide `GetMediaFromFragmentList` de, et de les traiter en parallèle.

Le schéma suivant illustre le flux de données pour des fragments et fragments au cours de ces appels d'API.

Lorsqu'un producteur envoie une requête PutMedia, il envoie les métadonnées média dans la charge utile, puis envoie une séquence de fragments de données média. Dès réception des données, Kinesis Video Streams stocke les données multimédia entrantes sous forme de fragments Kinesis Video Streams. Chaque morceau se compose des éléments suivants :

- Une copie des métadonnées média
- Un fragment
- Streams-specific Métadonnées Kinesis Video ; par exemple, le numéro du fragment et les horodatages côté serveur et côté producteur

Lorsqu'un consommateur demande des métadonnées multimédia, Kinesis Video Streams renvoie un flux de fragments, en commençant par le numéro de fragment que vous spécifiez dans la demande.

Si vous activez la persistance des données pour le flux, après avoir reçu un fragment dans le flux, Kinesis Video Streams enregistre également une copie du fragment dans le magasin de données.

Modèle de découverte des terminaux

API REST du plan de contrôle

Pour accéder aux [API REST Kinesis Video Streams Control Plane](#), utilisez les points de terminaison [du service Kinesis Video Streams](#).

API REST Data Plane

Kinesis Video Streams repose sur une architecture [cellulaire afin de garantir une](#) meilleure évolutivité et de meilleures propriétés d'isolation du trafic. Chaque flux étant mappé à une cellule spécifique d'une région, votre application doit utiliser les points de terminaison spécifiques à la cellule auxquels votre flux a été mappé. Lorsque vous accédez aux API REST de Data Plane, vous devez gérer et mapper vous-même les points de terminaison appropriés. Ce processus, le modèle de découverte des terminaux, est décrit ci-dessous :

1. Le modèle de découverte des terminaux commence par un appel à l'une des GetEndpoints actions. Ces actions appartiennent au plan de contrôle.
 1. Si vous recherchez les points de terminaison pour les services [Amazon Kinesis Video Streams Media API](#) ou [Amazon Kinesis Video Streams Archived Media API](#), utilisez [GetDataEndpoint](#)

2. [Si vous recherchez les points de terminaison pour l'API Amazon Kinesis Video Signaling Channels, l'API de stockage Amazon Kinesis Video WebRTC ou la signalisation vidéo Kinesis, utilisez `GetSignalingChannelEndpoint`](#)
2. Mettez en cache et réutilisez le point de terminaison.
3. Si le point de terminaison mis en cache ne fonctionne plus, effectuez un nouvel appel pour `GetEndpoints` actualiser le point de terminaison.

Bibliothèques pour producteurs

Après avoir créé un flux vidéo Kinesis, vous pouvez commencer à lui envoyer des données. Dans le code de votre application, vous pouvez utiliser ces bibliothèques pour extraire des données de vos sources multimédia et les télécharger dans votre flux vidéo Kinesis. Pour plus d'informations sur les bibliothèques producteur disponibles, consultez [Transférer vers Kinesis Video Streams](#).

Modèle de données Kinesis Video Streams

Les [Transférer vers Kinesis Video Streams](#) et [the section called “Diffuser à l'aide de la bibliothèque d'analyseurs”](#) envoient et reçoivent des données vidéo dans un format prenant en charge les informations d'incorporation avec les données vidéo. Ce format repose sur la spécification Matroska (MKV).

Le [Format MKV](#) est une spécification ouverte de données média. Toutes les bibliothèques et tous les exemples de code du guide du développeur Amazon Kinesis Video Streams envoient ou reçoivent des données au format MKV.

Il [Transférer vers Kinesis Video Streams](#) utilise les Frame types `StreamDefinition` et pour produire des en-têtes de flux MKV, des en-têtes de cadre et des données d'image.

Pour obtenir des informations sur la spécification MKV, consultez [Spécifications Matroska](#).

Les sections suivantes décrivent les composants des MKV-formatted données produites par le [C++](#).

Rubriques

- [Diffuser des éléments d'en-tête](#)
- [Diffusez les données de suivi](#)
- [Éléments d'en-tête du cadre](#)
- [Données de trame MKV](#)

Diffuser des éléments d'en-tête

Les éléments d'en-tête MKV suivants sont utilisés par `StreamDefinition` (défini dans `StreamDefinition.h`).

| Element | Description | Valeurs typiques |
|-------------------------------|--|--------------------------------------|
| <code>stream_name</code> | Correspond au nom du flux vidéo Kinesis. | mon-flux |
| <code>retention_period</code> | Durée, en heures, pendant laquelle les données de streaming sont conservées par Kinesis Video Streams. Spécifiez 0 pour un flux qui ne conserve pas de données. | 24 |
| <code>tags</code> | Une collection clé-valeur de données utilisateur. Ces données sont affichées dans la AWS Management Console et peuvent être lues par les applications client afin de filtrer ou d'obtenir des informations sur un flux. | |
| <code>kms_key_id</code> | Si elle est présente, la AWS KMS clé définie par l'utilisateur est utilisée pour chiffrer les données du flux. En cas d'absence, les données sont cryptées par la Kinesis-s applied clé (<code>aws/kinesisvideo</code>). | 01234567-89ab-cdef-0123-456789ab |
| <code>streaming_type</code> | Actuellement, le seul type de flux valide est <code>STREAMING_TYPE_REALTIME</code> . | <code>STREAMING_TYPE_REALTIME</code> |

| Element | Description | Valeurs typiques |
|-------------------------|--|------------------|
| content_type | Le type de contenu défini par l'utilisateur. Pour le streaming de données vidéo que la console va lire, le contenu doit être de type <code>video/h264</code> . | video/h264 |
| max_latency | Cette valeur n'est pas utilisée actuellement et doit être définie sur 0. | 0 |
| fragment_duration | L'estimation de la durée de vos fragments, utile à l'optimisation. La durée de fragment réelle est déterminée par les données de streaming. | 2 |
| timecode_scale | Indique l'échelle utilisée par les horodatages des images. La valeur par défaut est 1 milliseconde. Spécifier 0 attribue également la valeur par défaut d'1 milliseconde. La valeur peut être comprise entre 100 nanosecondes et 1 seconde. Pour plus d'informations, consultez TimecodeScale la documentation de Matroska. | |
| key_frame_fragmentation | S'il est défini sur <code>true</code> , le flux démarre un nouveau cluster lorsqu'une image clé est reçue. | true |

| Element | Description | Valeurs typiques |
|------------------------|---|-------------------|
| frame_timecodes | Si <code>true</code> , Kinesis Video Streams utilise les valeurs d'horodatage de présentation (pts) et d'horodatage de décodage (dts) des images reçues. Si <code>false</code> , Kinesis Video Streams tamponne les images lorsqu'elles sont reçues avec des valeurs temporelles générées par le système. | <code>true</code> |
| absolute_fragment_time | Si ce paramètre est défini sur <code>true</code> , les codes horaires du cluster sont interprétés comme utilisant l'heure absolue (par exemple, selon l'horloge système du producteur). Si <code>false</code> , les timescodes du cluster sont interprétés comme étant relatifs à l'heure de début du flux. | <code>true</code> |
| fragment_acks | Si <code>true</code> , des accusés de réception (ACK) sont envoyés lorsque Kinesis Video Streams reçoit les données. Les accusés de réception peuvent être reçus à l'aide des rappels <code>KinesisVideoStreamFragmentAck</code> ou <code>KinesisVideoStreamParseFragmentAck</code> . | <code>true</code> |

| Element | Description | Valeurs typiques |
|----------------------|---|----------------------------|
| restart_on_error | Indique si le flux doit reprendre la transmission après le signalement d'une erreur de flux. | true |
| nal_adaptation_flags | Indique si l'adaptation NAL (Network Abstraction Layer) ou les données privées codec sont présentes dans le contenu. Les indicateurs valides comprennent NAL_ADAPTATION_ANNEXB_NALS et NAL_ADAPTATION_ANNEXB_CPD_NALS . | NAL_ADAPTATION_ANNEXB_NALS |
| frame_rate | Une estimation du taux d'image du contenu. Cette valeur est utilisée pour l'optimisation ; le taux réel d'image est déterminé par le taux des données entrantes. Spécifier 0 attribue la valeur par défaut 24. | 24 |

| Element | Description | Valeurs typiques |
|-------------------|--|------------------|
| avg_bandwidth_bps | Estimation de la bande passante du contenu, en Mbits/s. Cette valeur est utilisée pour l'optimisation ; le taux réel est déterminé par la bande passante des données entrantes. Par exemple, pour un flux vidéo d'une résolution de 720 p s'exécutant à 25 FPS, vous pouvez vous attendre à une bande passante moyenne de 5 Mbit/s. | 5 |
| buffer_duration | La durée de mise en tampon du contenu sur le producteur. Si la latence du réseau est faible, cette valeur peut être réduite. Si la latence du réseau est élevée, l'augmentation de cette valeur empêche les trames d'être supprimées avant de pouvoir être envoyées, car l'allocation ne permet pas de placer les trames dans la plus petite mémoire tampon. | |

| Element | Description | Valeurs typiques |
|----------------------|---|------------------|
| replay_duration | Durée pendant laquelle le flux de données vidéo est « rembobiné » en cas de perte de connexion. Cette valeur peut être nulle si la perte d'images due à une perte de connexion n'est pas un problème. La valeur peut être augmentée si l'application utilisatrice parvient à supprimer les trames redondantes. Cette valeur doit être inférieure à la durée de la mémoire tampon, sinon la durée de la mémoire tampon est utilisée. | |
| connection_staleness | La durée pendant laquelle une connexion est maintenue quand aucune donnée n'est reçue. | |
| codec_id | Le codec utilisé par le contenu. Pour plus d'informations, consultez CodecID dans la spécification Matroska. | V_MPEG2 |
| track_name | Le nom défini par l'utilisateur de la piste. | ma_piste |

| Element | Description | Valeurs typiques |
|-----------------------|--|--|
| codec PrivateData | Les données fournies par l'encodeur utilisées pour coder les données d'image, notamment la largeur et la hauteur de l'image en pixels, nécessaires à de nombreux consommateurs en aval. Dans la bibliothèque C++ Producer , le gMkvTrackVideoBits tableau MkvStatics.cpp inclut la largeur et la hauteur en pixels du cadre. | |
| codec PrivateDataSize | La taille des données dans le paramètre codecPrivateData . | |
| track_type | Type de piste pour le flux. | MKV_TRACK_INFO_TYPE_AUDIO ou MKV_TRACK_INFO_TYPE_VIDEO |
| segment_uuid | User-defined uuid du segment (16 octets). | |
| default_track_id | Numéro unique différent de zéro pour la piste. | 1 |

Diffusez les données de suivi

Les éléments de piste MKV suivants sont utilisés par StreamDefinition (défini dans StreamDefinition.h).

| Element | Description | Valeurs typiques |
|------------|--|---------------------------|
| track_name | User-defined nom de la piste. Par exemple, « audio » pour la piste audio. | audio |
| codec_id | ID de codec de la piste Par exemple, « A_AAC » pour une piste audio. | A_AAC |
| cpd | Données fournies par l'encodeur, utilisées pour décoder les données de trame. Ces données peuvent inclure la largeur et la hauteur de trame en pixels, qui sont requis par de nombreux consommateurs en aval. Dans la bibliothèque de production C++ , le MkvTrackVideoBits tableau g MkvStatics.cpp inclut la largeur et la hauteur en pixels du cadre. | |
| cpd_size | Taille des données dans le PrivateData paramètre du codec. | |
| track_type | Type de la piste. Par exemple, vous pouvez utiliser la valeur enum MKV_TRACK_INFO_TYPE_AUDIO pour l'audio. | MKV_TRACK_INFO_TYPE_AUDIO |

Éléments d'en-tête du cadre

Les éléments d'en-tête MKV suivants sont utilisés par `Frame` (défini dans le package `KinesisVideoPic`, dans `mkvgen/Include.h`) :

- Index d'image : Une valeur en croissance monotone.
- Indicateurs : Le type d'image. Les valeurs valides sont les suivantes :
 - `FRAME_FLAGS_NONE`
 - `FRAME_FLAG_KEY_FRAME` : Si `key_frame_fragmentation` est définie sur le flux, les images clés commencent un nouveau fragment.
 - `FRAME_FLAG_DISCARDABLE_FRAME` : Indique au décodeur qu'il peut ignorer cette image si le décodage est lent.
 - `FRAME_FLAG_INVISIBLE_FRAME` : La durée de ce bloc est de 0.
- Horodatage du décodage : date à laquelle cette image a été décodée. Si les images précédentes dépendent de cette image pour le décodage, cet horodatage peut être antérieur à celui des images précédentes. Cette valeur est appliquée par rapport au début du fragment.
- Horodatage de la présentation : horodatage de l'affichage de ce cadre. Cette valeur est appliquée par rapport au début du fragment.
- Durée : La durée de lecture de l'image.
- Taille : La taille des données de l'image en octets

Données de trame MKV

Selon le schéma de codage utilisé, les données de `frame.frameData` peuvent uniquement contenir les données médias de l'image ou contenir des informations d'en-tête imbriquées. Pour être affichées dans le AWS Management Console, les données doivent être codées dans le [H.264](#) codec, mais Kinesis Video Streams peut recevoir des flux de données sérialisés dans le temps dans n'importe quel format.

Configuration système requise pour Amazon Kinesis Video Streams

Les sections suivantes décrivent les exigences matérielles, logicielles et de stockage pour Amazon Kinesis Video Streams.

Rubriques

- [Exigences relatives à la caméra](#)
- [Systèmes d'exploitation testés](#)
- [Exigences de stockage du SDK](#)

Exigences relatives à la caméra

Les caméras utilisées pour exécuter le kit de développement logiciel Kinesis Video Streams Producer et les exemples présentent les exigences de mémoire suivantes :

- L'affichage de contenu SDK nécessite 16 Mo de mémoire.
- La configuration par défaut de l'exemple d'application est de 512 Mo. Cette valeur convient pour les producteurs qui ont une bonne connectivité réseau et aucune des exigences relatives au tampon supplémentaire. Si la connectivité réseau est faible et si davantage de mémoire tampon est requise, vous pouvez calculer la mémoire requise par seconde de mise en mémoire tampon en multipliant la fréquence de trames par seconde par la taille de la mémoire de trame. Pour plus d'informations sur l'allocation de mémoire, consultez [StorageInfo](#).

Nous vous recommandons d'utiliser des caméras USB ou RTSP (Real Time Streaming Protocol) qui encodent les données à l'aide de cette technologie, H.264 car cela supprime la charge de travail d'encodage du processeur.

Actuellement, l'application de démonstration ne prend pas en charge le protocole UDP (User Datagram Protocol) pour le streaming RTSP. Cette capacité sera ajouté ultérieurement.

Le SDK du fabricant prend en charge les types de caméras suivants :

- Caméras Web.
- Caméras USB.
- Caméras avec H.264 encodage (de préférence).
- Caméras sans H.264 encodage.
- Module caméra Raspberry Pi. Cette solution est préférable pour les appareils Raspberry Pi, car elle permet la connexion à GPU pour le transfert de données vidéo et, par conséquent, il n'y a pas de frais généraux pour le traitement de l'UC.
- Caméras (réseau) RTSP. Ces caméras sont préférées car les flux vidéo sont déjà encodés avec H.264.

Systèmes d'exploitation testés

Nous avons testé les caméras Web et les caméras RTSP avec les appareils et systèmes d'exploitation suivants :

- Mac mini
 - High Sierra
- MacBook Ordinateurs portables Pro
 - Sierra (10.12)
 - El Capitan (10.11)
- Ordinateurs portables HP exécutant Ubuntu 16.04
- Ubuntu 17.10 (conteneur Docker)
- Raspberry Pi 3

Exigences de stockage du SDK

L'installation de [Transférer vers Kinesis Video Streams](#) requiert un espace de stockage minimal de 170 Mo et un espace de 512 Mo est recommandé.

Quotas du service Amazon Kinesis Video Streams

Kinesis Video Streams dispose des quotas de service suivants :


Important

Les quotas de service suivants sont soit souples [s], qui peuvent être améliorés en soumettant un ticket d'assistance, soit stricts [h], qui ne peuvent pas être augmentés. Vous verrez [s] et [h] à côté du quota de service individuel dans les tableaux ci-dessous.

Quotas de service API du plan de contrôle

La section suivante décrit les quotas de service pour les API du plan de contrôle. TPS représente le nombre de transactions par seconde.

Lorsqu'une limite de demandes au niveau du compte ou au niveau des ressources est atteinte, un est émis. `ClientLimitExceededException`

| API | Limite de compte : demande | Limite du compte : Streams | Stream-level limite | Exceptions et remarques pertinentes |
|--------------|----------------------------|---|---------------------|---|
| CreateStream | 50 TPS [s] | 10 000 streams par compte [s] dans toutes les régions prises en charge. <div data-bbox="592 699 792 1879" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Cette limite peut être augmenté jusqu'à 100 000 flux (ou plus) par compte(s).</p> <p>Connectez-vous à l'AWS Management Console adresse https://</p> </div> | | Les appareils, les CLI, les SDK-driven accès et la console peuvent tous invoquer cette API. Un seul appel d'API réussit si le flux n'existe pas encore. |

| API | Limite de compte : demande | Limite du compte : Streams | Stream-level limite | Exceptions et remarques pertinentes |
|--------------------------------------|----------------------------|--|---------------------|-------------------------------------|
| | | console.aws.amazon.com/ et demandez une augmentation de cette limite. | | |
| DeleteEdgeConfiguration | 10 TPS [h] | N/A | 1 TPS [h] | |
| DeleteStream | 50 TPS [h] | N/A | 5 TPS [h] | |
| DescribeEdgeConfiguration | 50 TPS [h] | N/A | 5 TPS [h] | |
| DescribeImageGenerationConfiguration | 50 TPS [h] | N/A | 5 TPS [h] | |
| DescribeMappedResourceConfiguration | 50 TPS [h] | N/A | 5 TPS [h] | |

| API | Limite de compte : demande | Limite du compte : Streams | Stream-level limite | Exceptions et remarques pertinentes |
|-----------------------------------|----------------------------|----------------------------|---------------------|--|
| DescribeNotificationConfiguration | 50 TPS [h] | N/A | 5 TPS [h] | |
| DescribeStream | 300 TPS [h] | N/A | 5 TPS [h] | |
| GetDataEndpoint | 300 TPS [h] | N/A | 5 TPS [h] | Appelée toutes les 45 minutes pour actualiser le jeton de streaming pour la plupart des cas d'utilisation PutMedia/GetMedia. La mise en cache des points de terminaison de données est sécurisée si l'application les recharge en cas d'échec. |
| ListEdgeAgentConfigurations | 50 TPS [h] | N/A | N/A | |
| ListStreams | 50 TPS [h] | N/A | | |
| ListTagsForStream | 50 TPS [h] | N/A | 5 TPS [h] | |
| StartEdgeConfigurationUpdate | 10 TPS [h] | N/A | 1 TPS [h] | |
| TagStream | 50 TPS [h] | N/A | 5 TPS [h] | |
| UntagStream | 50 TPS [h] | N/A | 5 TPS [h] | |

| API | Limite de compte : demande | Limite du compte : Streams | Stream-level limite | Exceptions et remarques pertinentes |
|------------------------------------|----------------------------|----------------------------|---------------------|-------------------------------------|
| UpdateDataRetention | 50 TPS [h] | N/A | 5 TPS [h] | |
| UpdateImageGenerationConfiguration | 50 TPS [h] | N/A | 5 TPS [h] | |
| UpdateNotificationConfiguration | 50 TPS [h] | N/A | 5 TPS [h] | |
| UpdateStream | 50 TPS [h] | N/A | 5 TPS [h] | |
| UpdateStreamStorageConfiguration | 50 TPS [h] | N/A | 5 TPS [h] | |
| DescribeStreamStorageConfiguration | 50 TPS [h] | N/A | 5 TPS [h] | |

Quotas de service d'API pour les médias et les médias archivés

La section suivante décrit les quotas de service pour les API multimédia et multimédia archivées.

Lorsqu'une limite de demandes au niveau du compte ou au niveau des ressources est atteinte, un est émis. `ClientLimitExceededException`

Lorsqu'une limite au niveau de la connexion est atteinte, une exception `ConnectionLimitExceededException` est déclenchée.

Les erreurs ou accusés de réception suivants sont déclenchés lorsqu'une limite au niveau du fragment est atteinte :

- Un accusé de réception `MIN_FRAGMENT_DURATION_REACHED` est renvoyé pour un fragment inférieur à la durée minimale.
- Un accusé de réception `MAX_FRAGMENT_DURATION_REACHED` est renvoyé pour un fragment supérieur à la durée maximale.
- Un accusé de réception `MAX_FRAGMENT_SIZE` est renvoyé pour un fragment supérieur à la taille de données maximale.
- Une exception `FragmentLimitExceeded` est déclenchée si une limite de fragment est atteinte dans une opération `GetMediaForFragmentList`.


Quotas de service d'API de plan de données

| API | Stream-level limite | Connection-level limite | Limite de bande passante | Fragment-level limite | Exceptions et remarques pertinentes |
|----------|---------------------|-------------------------|---|---|--|
| PutMedia | 5 TPS [h] | 1 [h] | 12,5 MB/second ou 100 Mbps [s] par flux | <ul style="list-style-type: none"> • Durée minimale d'un fragment : 1 seconde • Durée maximale du fragment : 20 secondes [h] • Taille maximale d'un fragment : 50 Mo [h] | <p>PutMedia les demandes sont des connexions en streaming et de longue durée. Il n'est pas nécessaire d'ouvrir une nouvelle connexion pour chaque élément de données, car vous pouvez envoyer plusieurs fragments au cours d'une seule connexion persistante. Si vous tentez plusieurs <code>PutMedia</code> connexions simultanées, Kinesis Video Streams limite les dernières connexions en affichant un</p> |

| API | Stream-level limite | Connection-level limite | Limite de bande passante | Fragment-level limite | Exceptions et remarques pertinentes |
|----------------------------|---------------------|-------------------------|--------------------------------|---|--|
| | | | | <ul style="list-style-type: none"> • Nombre maximal de pistes : 3 [s] • Nombre maximum de fragments envoyés par seconde : 5 [h] • Limite maximale de métadonnées de fragments : 10 balises [h] | message d'ConnectionLimitExceededException erreur. |
| GetClip | N/A | N/A | Limite de taille de 100 Mo [h] | Nombre maximal de fragments : 200 [h] | |
| GetDASHStreamingSessionURL | 25 TPS [h] | N/A | N/A | N/A | |

| API | Stream-level limite | Connection-level limite | Limite de bande passante | Fragment-level limite | Exceptions et remarques pertinentes |
|---------------------------|---------------------|-------------------------|--------------------------|-----------------------|---|
| GetHLSStreamingSessionURL | 25 TPS [h] | N/A | N/A | N/A | |
| GetImages | N/A | N/A | 100 Mo [h] | N/A | <p>Le nombre maximum d'images par demande est de 100 [h].</p> <div data-bbox="1159 701 1451 1129" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px; background-color: #e6f2ff;"> <p>Note</p> <p>La valeur minimale pour <code>SamplingInterval</code> est de 200 millisecondes (ms), soit 5 images par seconde.</p> </div> |

| API | Stream-level limite | Connection-level limite | Limite de bande passante | Fragment-level limite | Exceptions et remarques pertinentes |
|----------|---------------------|-------------------------|----------------------------|--|--|
| GetMedia | 5 TPS [h] | 3 [h] | 25 MB/s ou 200 Mbits/s [s] | Maximum de 5 fragments envoyés par seconde [h] | <p>GetMediales demandes sont des connexions en streaming et de longue durée. Il n'est pas nécessaire d'ouvrir une nouvelle connexion pour chaque élément de données, car vous pouvez envoyer plusieurs fragments au cours d'une seule connexion persistante. Si vous tentez plus de trois GetMedia connexions simultanées, Kinesis Video Streams limite les dernières connexions en affichant un message d'ConnectionLimitExceededException erreur.</p> <p>Si la taille d'un fragment typique est d'environ 5 Mo, cette limite correspond à environ 75 Mo/s par flux vidéo Kinesis. Un tel flux aurait un débit binaire sortant égal à 2 fois le débit binaire entrant maximal des flux.</p> |

| API | Stream-level limite | Connection-level limite | Limite de bande passante | Fragment-level limite | Exceptions et remarques pertinentes |
|-------------------------|---------------------|-------------------------|----------------------------|--|--|
| | | | | | <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E1F5FE;"> <p> Note</p> <p>GetMedian'est pas utilisé pour HLS/DASH la lecture.</p> </div> |
| GetMediaForFragmentList | N/A | 5 [s] | 25 MB/s ou 200 Mbits/s [s] | Nombre maximal de fragments : 1000 [h] | Cinq applications consommatrices basées sur des fragments peuvent être invoquées simultanément. GetMediaForFragmentList Les autres connexions sont rejetées. |

Quotas de service API du protocole de lecture vidéo

| API | Session-level limite | Fragment-level limite |
|-------------------------|----------------------|--|
| GetDASHManifestPlaylist | 5 TPS [h] | Nombre maximum de fragments par playlist : 5 000 [h] |
| GetHLSMasterPlaylist | 5 TPS [h] | N/A |
| GetHLSMediaPlaylist | 5 TPS [h] | Nombre maximum de fragments par playlist : 5 000 [h] |
| GetMP4InitFragment | 5 TPS [h] | N/A |
| GetMP4MediaFragment | 20 TPS [h] | N/A |

| API | Session-level limite | Fragment-level limite |
|---------------|----------------------|-----------------------|
| GetTSFragment | 20 TPS [h] | N/A |

Fragment-metadata et quotas de médias fragmentés

Les [API Kinesis Video Streams permettant d'accéder aux médias archivés](#) sont limitées en fonction du nombre de fragments demandés plutôt que du nombre d'appels d'API. Le débit des API est limité à la fois par le nombre de métadonnées de fragments et par le nombre de fragments multimédias demandés. Les métadonnées des fragments et les quotas de fragments multimédias sont appliqués par flux. En d'autres termes, les demandes de métadonnées fragmentées ou de contenu multimédia dans un flux ne s'appliquent pas aux quotas d'un autre flux. Toutefois, au sein d'un flux donné, chaque quota est partagé entre plusieurs API. Cela signifie que, pour un flux donné, les demandes de fragments provenant de différentes API consomment le même quota. Lorsque les métadonnées du fragment ou le quota multimédia du fragment pour un flux sont dépassés, l'API renvoie un `ClientLimitExceededException`. Les tableaux suivants montrent comment les API consomment chacun des deux types de quotas. Pour la deuxième colonne de ces tableaux, supposons que si un flux a un quota de N, cela signifie que les API ont N points à consommer à partir de ce type de quota pour ce flux. L'GetClipAPI apparaît dans les deux tableaux.

Fragment-metadata consommation de quotas

| API | Nombre de points de quota consommés par demande | Quota partagé (N) |
|---------------------|--|--|
| ListFragments | Valeur du <code>MaxResults</code> paramètre | 10 000 points de quota par seconde, par stream [h] |
| GetClip | Nombre de fragments dans le clip obtenu | |
| GetHLSMediaPlaylist | Valeur du <code>MaxMediaPlaylistFragmentResults</code> paramètre | |

| API | Nombre de points de quota consommés par demande | Quota partagé (N) |
|-----------------|---|-------------------|
| GetDASHManifest | Valeur du MaxManifestFragmentResults paramètre | |
| GetImages | Valeur de 400 + nombre maximum d'images demandées | |

Fragment-media consommation de quotas

| API | Nombre de points de quota consommés par demande | Quota partagé (N) |
|-------------------------|---|---|
| GetMediaForFragmentList | Nombre de fragments dans le paramètre Fragments | 500 points de quota par seconde, par stream [h] |
| GetClip | Nombre de fragments dans le clip obtenu | |
| GetMP4MediaFragment | 1 | |
| GetTSFragment | 1 | |
| GetImages | Nombre maximum d'images demandées | |

Par exemple, avec un quota de 500 fragments de média par seconde, les modèles d'appel suivants sont pris en charge pour un flux donné :

- 5 requêtes par seconde GetClip avec 100 fragments dans chaque clip.
- 100 requêtes par seconde GetClip avec 5 fragments dans chaque clip.
- 2 requêtes par seconde pour GetClip 100 fragments par clip et 3 requêtes par seconde pour GetMediaForFragmentList chaque clip.

- 400 demandes par seconde vers `GetMP4MediaFragment` et 100 demandes par seconde vers `GetTSFragment`.

Ces quotas ont une incidence importante sur le nombre de HLS et de MPEG-DASH sessions pouvant être prises en charge par flux. Il n'y a aucune limite au nombre de sessions HLS et DASH pouvant être utilisées par les lecteurs multimédias à un moment donné. Il est donc important que l'application de lecture n'autorise pas l'utilisation simultanée d'un trop grand nombre de sessions. Les deux exemples suivants décrivent comment déterminer le nombre de sessions de lecture simultanées pouvant être prises en charge :

Exemple 1 : diffusion en direct

Dans un scénario de diffusion en direct avec HLS avec des fragments d'une durée d'une seconde, une piste audio et vidéo, et `MaxMediaPlaylistFragmentResults` réglé sur cinq, un lecteur multimédia effectue généralement deux appels `GetHLSMediaPlaylist` par seconde. Un appel porte sur les dernières métadonnées vidéo et un autre sur les métadonnées audio correspondantes. Les deux appels consomment chacun cinq points de quota de métadonnées de fragments. Il effectue également deux appels `GetMP4MediaFragment` par seconde : un appel pour la dernière vidéo et un autre pour le son correspondant. Chaque appel consomme un jeton multimédia fragmenté, donc deux jetons sont consommés au total.

Dans ce scénario, jusqu'à 250 sessions de lecture simultanées peuvent être prises en charge. Avec 250 sessions, ce scénario consomme 2 500 points de quota de métadonnées de fragments par seconde (bien en deçà du quota de 10 000) et 500 points de quota de fragments de média par seconde.

Exemple 2 : On-demand lecture

Dans un scénario de lecture à la demande d'un événement passé avec MPEG-DASH une piste audio et vidéo `MaxManifestFragmentResults` définie sur 1 000, un lecteur multimédia appelle généralement une `GetDASHManifest` fois au début de la session (consommation de 1 000 points de quota de métadonnées de fragments) et il appelle jusqu'à `GetMP4MediaFragment` à 5 fois par seconde (consommation de 5 points de quota de fragments de contenu multimédia) jusqu'à ce que tous les fragments soient chargés. Dans ce scénario, jusqu'à 10 nouvelles sessions peuvent être démarrées par seconde (juste au quota de 10 000 fragments de métadonnées par seconde), et jusqu'à 100 sessions peuvent charger activement des fragments de média à un rythme de 5 par seconde (juste au quota de 500 fragments de média par seconde).

Vous pouvez utiliser `ArchivedFragmentsConsumed.Metadata` et `ArchivedFragmentsConsumed.Media` surveiller la consommation de métadonnées de fragments et de points de quota de fragments de média, respectivement. Pour plus d'informations sur la surveillance, consultez [the section called "Contrôle"](#).

Quotas du service de métadonnées en streaming

Les quotas de service suivants s'appliquent à l'ajout de métadonnées de streaming à un flux vidéo Kinesis :

- Vous pouvez ajouter jusqu'à 10 éléments de métadonnées à un fragment.
- La longueur d'un nom de métadonnées de fragment ne doit pas dépasser 128 octets.
- La longueur d'une valeur de métadonnées de fragment ne doit pas dépasser 256 octets.
- Le nom des métadonnées d'un fragment ne peut pas commencer par la chaîne « AWS ». Si un tel élément de métadonnées est ajouté, la méthode `putFragmentMetadata` du PIC renvoie une erreur `STATUS_INVALID_METADATA_NAME` (code d'erreur `0x52000077`). Votre application peut alors ignorer l'erreur (le PIC n'ajoute pas l'élément de métadonnées), ou répondre à l'erreur.

Quotas du SDK pour les producteurs

Le tableau suivant contient les quotas actuels pour les valeurs du SDK. Pour plus d'informations, consultez [Transférer vers Kinesis Video Streams](#).

Note

Avant de définir ces valeurs, vous devez valider vos entrées. Le kit SDK ne valide pas ces limites et une erreur d'exécution se produit en cas de dépassement des limites.

| Value | Limite | Remarques |
|---------------------|--------|--|
| Nombre max. de flux | 128 | Nombre maximal de flux qu'un objet producteur peut créer. Il s'agit d'une limite flexible (vous pouvez demander une augmentation). Cela garantit que le producteur ne crée pas |

| Value | Limite | Remarques |
|--|---|---|
| | | accidentellement des flux de manière récursive. |
| Longueur max. du nom de l'appareil | 128 caractères | |
| Nombre max. de balises | 50 par flux | |
| Longueur max. de nom de flux | 256 caractères | |
| Taille de stockage min. | 10 Mio = $10 * 1024 * 1024$ octets | |
| Taille de stockage max. | 10 Gio = $10 * 1024 * 1024 * 1024$ octets | |
| Longueur max. du chemin du répertoire racine | 4,096 caractères | |
| Longueur max. des infos d'authentification | 10 000 octets | |
| Longueur max. de chaîne d'URI | 10 000 caractères | |
| Longueur max. de nom de balise | 128 caractères | |
| Longueur max. de valeur de balise | 1,024 caractères | |
| Période min. de jeton de sécurité | 30 secondes | |
| Période de grâce de jeton de sécurité | 40 minutes | Si la durée spécifiée est plus longue, elle est limitée à cette valeur. |

| Value | Limite | Remarques |
|--|-----------------------------|---|
| Période de conservation | 0 ou supérieure à une heure | 0 indique l'absence de conservation. |
| Durée de cluster min. | 1 seconde | La valeur est spécifiée en unités de 100 ns, ce qui est le standard du kit SDK. |
| Durée de cluster max. | 30 secondes | La valeur est spécifiée en unités de 100 ns, ce qui est le standard du kit SDK. L'API principale peut imposer une durée de cluster plus courte. |
| Taille de fragment max. | 50 Mo | Pour de plus amples informations, veuillez consulter Quotas du service Amazon Kinesis Video Streams . |
| Durée de fragment max. | 20 secondes | Pour de plus amples informations, veuillez consulter Quotas du service Amazon Kinesis Video Streams . |
| Durée de connexion max. | 45 minutes | Le backend ferme la connexion après cette période. Le kit SDK effectue une rotation du jeton et établit une nouvelle connexion au cours de cette période. |
| Longueur max. de segment ACK | 1,024 caractères | Longueur max. de segment de la confirmation envoyée à la fonction d'analyseur ACK. |
| Longueur max. de chaîne de type de contenu | 128 caractères | |

| Value | Limite | Remarques |
|---|--------------------------------|--|
| Longueur max. de chaîne d'ID de codec | 32 caractères | |
| Longueur max. de chaîne de nom de piste | 32 caractères | |
| Longueur max. des données privées codec | 1 Mio = 1 * 1024 * 1024 octets | |
| Longueur min. de valeur d'échelle de code horaire | 100 ns | La valeur minimale d'échelle de code horaire pour représenter les horodatages des images dans le cluster MKV obtenu. Cette valeur est spécifiée sous forme d'incrément de 100 ns, ce qui est le standard du kit SDK. |
| Longueur max. de valeur d'échelle de code horaire | 1 seconde | La valeur maximale d'échelle de code horaire pour représenter les horodatages des images dans le cluster MKV obtenu. Cette valeur est spécifiée sous forme d'incrément de 100 ns, ce qui est le standard du kit SDK. |
| Nombre min. d'éléments d'affichage de contenu | 10 | |
| Durée min. de mise en tampon | 20 secondes | Cette valeur est spécifiée sous forme d'incrément de 100 ns, ce qui est le standard du kit SDK. |

| Value | Limite | Remarques |
|---|-----------------|-----------|
| Longueur max. de version de mise à jour | 128 caractères | |
| Longueur max. de nom ARN | 1024 caractères | |
| Longueur max. de séquence de fragment | 128 caractères | |
| Période de conservation maximale | 10 Ans | |

Stockage hiérarchisé Amazon Kinesis Video Streams

Le stockage hiérarchisé Amazon Kinesis Video Streams vous permet d'optimiser les coûts associés au stockage et à la consommation de contenu multimédia en fonction des besoins de consommation attendus, de la fréquence d'accès aux données et de la durée de conservation. KVS propose deux niveaux de stockage pour vous aider à équilibrer les coûts et les exigences de performance.

Rubriques

- [Présentation de](#)
- [Concepts clés](#)
- [Configuration du stockage hiérarchisé](#)
- [Bonnes pratiques](#)

Présentation de

Kinesis Video Streams propose 2 niveaux de stockage :

- Niveau Kinesis Video Streams Standard (Hot) : fournit un accès immédiat avec une latence minimale aux médias fréquemment consultés.
- Kinesis Video Streams Warm Tier Cost-effective : pour les cas d'utilisation dans lesquels le contenu multimédia est stocké pendant une plus longue durée, est consulté moins fréquemment, mais nécessite un accès rapide en cas de besoin.

Concepts clés

La compréhension de ces concepts vous aidera à configurer des politiques de stockage optimales :

- Coût de stockage — Le niveau chaud est moins cher que le niveau chaud.
- Coût d'ingestion : les frais du niveau Warm sont facturés pour 1 000 fragments, tandis que ceux du niveau Hot sont facturés pour 1 Go de contenu multimédia ingéré.
- Durée de stockage minimale : le stockage à chaud est facturé pour une durée de stockage minimale de 30 jours. Si le contenu multimédia est supprimé ou déplacé avant cette période de 30 jours, vous devrez payer les frais d'utilisation normaux du stockage, plus des frais calculés au prorata pour le reste de la durée de stockage minimale. Les supports stockés pendant une durée supérieure à la durée de stockage minimale ne seront pas soumis à des frais minimaux.
- Latence — Légère augmentation (millisecondes) de la latence d'ingestion et de consommation lorsque le contenu multimédia est stocké dans le niveau Warm.

Configuration du stockage hiérarchisé

Créez un nouveau stream avec Warm Tier Ingestion

1. Ouvrez la console à l'adresse <https://console.aws.amazon.com/kinesisvideo/home>.
2. Sur la page Video streams (Flux vidéo) choisissez Create video stream (Créer un flux vidéo).
3. Sur la page Créer un nouveau flux vidéo, saisissez *YourStreamName* le nom du flux.
4. Choisissez Configuration personnalisée. Définissez une conservation des données (par exemple *31 days*). Définissez le niveau de stockage sur *Warm tier*.
5. Choisissez Create video stream (Créer un flux vidéo).
6. Une fois qu'Amazon Kinesis Video Streams a créé le flux, consultez les informations sur *YourStreamName* la page.

Votre flux ingère désormais le contenu multimédia directement dans le Warm Tier, offrant ainsi un accès immédiat avec une latence minimale.

Bonnes pratiques

- Analysez vos modèles d'accès aux médias avant de sélectionner les niveaux de stockage
- Utilisez Hot Tier pour les médias fréquemment consultés nécessitant un accès immédiat

- Configurer le niveau Warm pour les médias accessibles occasionnellement avec une légère latence acceptable
- Tenez compte du coût total de propriété, y compris les coûts de stockage et de récupération

Configuré un compte .

Avant d'utiliser Amazon Kinesis Video Streams pour la première fois, effectuez les tâches suivantes.

Rubriques

- [Inscrivez-vous pour un Compte AWS](#)
- [Créez un Compte AWS clé](#)

Inscrivez-vous pour un Compte AWS

Pour commencer AWS, vous avez besoin d'un Compte AWS. Pour plus d'informations sur la création d'un Compte AWS, voir [Getting started with an Compte AWS](#) dans le Guide de Gestion de compte AWS référence.

Créez un Compte AWS clé

Vous aurez besoin d'une Compte AWS clé pour accéder à Amazon Kinesis Video Streams par programmation.

Pour créer une Compte AWS clé, procédez comme suit :

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
2. Choisissez Utilisateurs dans la barre de navigation, puis choisissez l'utilisateur administrateur.
3. Choisissez l'onglet Informations d'identification de sécurité, puis Créer une clé d'accès.
4. Enregistrez l'ID de clé d'accès. Choisissez Afficher sous Clé d'accès secrète. Enregistrez la clé d'accès secrète.

Avantages

Les avantages de l'utilisation de Kinesis Video Streams sont les suivants :

- Connectez-vous et diffusez depuis des millions d'appareils : vous pouvez utiliser Kinesis Video Streams pour connecter et diffuser des données vidéo, audio et autres provenant de millions d'appareils, tels que des smartphones grand public, des drones et des caméras embarquées. Vous pouvez utiliser les bibliothèques de production Kinesis Video Streams pour configurer vos appareils et diffuser des contenus de manière fiable en temps réel, ou sous forme de téléchargement de contenu multimédia après coup.
- Stockez, chiffrez et indexez les données de manière durable : vous pouvez configurer votre flux vidéo Kinesis pour stocker durablement les données multimédia pendant des périodes de conservation personnalisées. Kinesis Video Streams génère également un index des données stockées en fonction des horodatages générés par le producteur ou du côté du service. Vos applications peuvent récupérer des données spécifiées dans un flux à l'aide de l'index temporel.
- Concentrez-vous sur la gestion des applications plutôt que sur l'infrastructure : Kinesis Video Streams fonctionne sans serveur, il n'y a donc aucune infrastructure à configurer ou à gérer. Vous n'avez pas à vous soucier du déploiement, de la configuration ou de la mise à l'échelle élastique de l'infrastructure sous-jacente, car vos flux de données et le nombre d'applications consommatrices augmentent et diminuent. Kinesis Video Streams effectue automatiquement toutes les tâches d'administration et de maintenance nécessaires à la gestion des flux, afin que vous puissiez vous concentrer sur les applications, et non sur l'infrastructure.
- Créez des applications en temps réel et par lots sur des flux de données : vous pouvez utiliser Kinesis Video Streams pour créer des applications personnalisées en temps réel qui fonctionnent sur des flux de données en direct, et créer des applications par lots ou ponctuelles qui fonctionnent sur des données persistantes de manière durable sans exigences de latence strictes. Vous pouvez créer, déployer et gérer des applications personnalisées : solutions open source (Apache MXnet, OpenCV), locales ou tierces en utilisant le AWS Marketplace pour traiter et analyser vos flux. Vous pouvez utiliser les API Kinesis Video Get Streams pour créer plusieurs applications simultanées traitant des données en temps réel ou par lots.
- Diffusez des données de manière plus sécurisée : Kinesis Video Streams chiffre toutes les données lorsqu'elles transitent par le service et lorsqu'elles sont conservées. Kinesis Video Streams applique le chiffrement basé sur le protocole TLS (Transport Layer Security) aux données diffusées en continu depuis les appareils, et chiffre toutes les données inactives à l'aide de (). AWS Key Management Service AWS KMS En outre, vous pouvez gérer l'accès à vos données à l'aide de Gestion des identités et des accès AWS (IAM).
- Payez au fur et à mesure — Pour plus d'informations, consultez [Calculateur de tarification AWS](#).

Commencer à utiliser Amazon Kinesis Video Streams

Cette section explique comment effectuer les tâches suivantes dans Amazon Kinesis Video Streams :

- Configurez votre Compte AWS compte et créez-en un administrateur, si ce n'est pas déjà fait.
- Créez un flux vidéo Kinesis.
- Envoyez des données vers le flux vidéo Kinesis depuis votre caméra et visionnez le contenu multimédia dans la console.

Si vous utilisez Amazon Kinesis Video Streams pour la première fois, nous vous recommandons de [Kinesis Video Streams : comment ça marche](#) lire d'abord.

Note

Le fait de suivre l'exemple de mise en route ne vous facturera aucun frais. Compte AWS Pour connaître les coûts liés aux données dans votre région, consultez la [tarification d'Amazon Kinesis Video Streams](#).

Rubriques

- [Création d'un flux vidéo Amazon Kinesis](#)
- [Envoyer des données vers un flux vidéo Amazon Kinesis](#)
- [Consommez des données multimédia](#)

Création d'un flux vidéo Amazon Kinesis

Cette section explique comment créer un flux vidéo Kinesis.

Cette section contient les procédures suivantes :

- [the section called “Création d'un flux vidéo à l'aide de la console”](#)
- [the section called “Créez un flux vidéo à l'aide du AWS CLI”](#)

Création d'un flux vidéo à l'aide de la console

1. Ouvrez la console à l'adresse <https://console.aws.amazon.com//kinesisvideo/home>.
2. Sur la page Video streams (Flux vidéo) choisissez Create video stream (Créer un flux vidéo).
3. Sur la page Créer un nouveau flux vidéo, saisissez *YourStreamName* le nom du flux. Laissez le bouton de configuration par défaut sélectionné.
4. Choisissez Create video stream (Créer un flux vidéo).
5. Une fois qu'Amazon Kinesis Video Streams a créé le flux, consultez les informations sur YourStreamName la page.

Créez un flux vidéo à l'aide du AWS CLI

1. Vérifiez que vous les avez AWS CLI installés et configurés. Pour plus d'informations, consultez la documentation [AWS Command Line Interface](#).
2. Exécutez la commande Create-Stream suivante dans l' AWS CLI :

```
aws kinesisvideo create-stream --stream-name "YourStreamName" --data-retention-in-hours 24
```

La réponse se présente comme suit :

```
{
  "StreamARN": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/YourStreamName/123456789012"
}
```

Envoyer des données vers un flux vidéo Amazon Kinesis

Cette section décrit comment envoyer des données multimédia depuis une caméra vers le flux vidéo Kinesis que vous avez créé dans la section précédente. Cette section utilise le kit [Utiliser la bibliothèque de production C++](#) en tant que plugin [Exemple : Plug-in GStreamer du SDK Kinesis Video Streams Producer - kvssink](#).

Pour envoyer du contenu multimédia depuis différents appareils sur différents systèmes d'exploitation, ce didacticiel utilise la bibliothèque de production Kinesis Video Streams C++ [et](#)

GStreamer, un framework multimédia open source qui normalise l'accès aux caméras et aux autres sources multimédia.

Rubriques

- [Créez le SDK et les exemples](#)
- [Exécutez les exemples pour télécharger du contenu multimédia sur Kinesis Video Streams](#)
- [Examiner les objets d'accusé de réception](#)

Créez le SDK et les exemples

Vous pouvez créer le SDK et les exemples sur votre ordinateur ou dans AWS Cloud9. Suivez les procédures appropriées ci-dessous.

Build on your computer

Utilisez les instructions du [fichier readme](#) pour créer la bibliothèque du producteur et un exemple d'application.

Cela inclut notamment les éléments suivants :

- Installation des dépendances
- Clonage du référentiel
- Utiliser CMake pour générer des makefiles
- Création des fichiers binaires à l'aide de make

Build in AWS Cloud9

Suivez ces procédures pour effectuer le téléchargement sur Kinesis Video Streams AWS Cloud9 in. Vous n'aurez pas besoin de télécharger quoi que ce soit sur votre ordinateur.

1. Dans le AWS Management Console, ouvrez [AWS Cloud9](#).

Sélectionnez Créer un environnement.

2. Sur l'écran Créer un environnement, effectuez les opérations suivantes :
 - Nom : saisissez le nom de votre nouvel environnement.
 - Plateforme - Sélectionnez Ubuntu Server 22.04 LTS.


```
gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-bad gstreamer1.0-plugins-good \  
gstreamer1.0-plugins-ugly gstreamer1.0-tools
```

- Utilisez git pour cloner le SDK du producteur C++.

```
git clone https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp.git
```

- Préparez un répertoire de compilation.

```
cd amazon-kinesis-video-streams-producer-sdk-cpp  
mkdir build  
cd build
```

- Utilisez CMake pour générer des makefiles.

```
cmake .. -DBUILD_GSTREAMER_PLUGIN=TRUE -DBUILD_DEPENDENCIES=OFF
```

La fin de la sortie attendue se présente comme suit :

```
-- Build files have been written to: /home/ubuntu/environment/amazon-kinesis-video-streams-producer-sdk-cpp/build
```

- Utilisez make pour compiler le SDK et les exemples d'applications, ainsi que pour créer les exécutables finaux.

```
make
```

La fin de la sortie attendue se présente comme suit :

```
[100%] Linking CXX executable kvs_gstreamer_file_uploader_sample  
[100%] Built target kvs_gstreamer_file_uploader_sample
```

- Vérifiez que les fichiers d'exemple ont été créés. Répertoriez les fichiers du répertoire en cours :

```
ls
```

Vérifiez que les fichiers suivants sont présents :

- kvs_gstreamer_sample
 - libgstkvssink.so
11. (Facultatif) Vous pouvez ajouter le réglage de la variable d'environnement `GST_PLUGIN_PATH` au script de démarrage de votre shell. Cela garantit que `GST_PLUGIN_PATH` est correctement défini lors d'une nouvelle session de terminal. Dans AWS Cloud9, le script de démarrage du shell est `~/ .bashrc`.

Exécutez la commande suivante pour ajouter la commande à la fin du script de démarrage du shell.

```
echo "export GST_PLUGIN_PATH=~/.environment/amazon-kinesis-video-streams-producer-sdk-cpp/build" >> ~/.bashrc
```

Tapez ce qui suit pour exécuter le script de démarrage du shell :

```
source ~/.bashrc
```

Vérifiez que `GST_PLUGIN_PATH` est défini.

```
echo $GST_PLUGIN_PATH
```

Si vous réglez correctement la sortie, vous verrez la sortie suivante. Si la sortie est vide, la variable d'environnement n'est pas définie correctement.

```
/home/ubuntu/environment/amazon-kinesis-video-streams-producer-sdk-cpp/build
```

Exécutez les exemples pour télécharger du contenu multimédia sur Kinesis Video Streams

L'exemple d'application ne prend pas en charge les informations d'identification IMDS. Dans votre terminal, exportez les AWS informations d'identification de votre utilisateur ou de votre rôle IAM et de la région dans laquelle se trouve votre diffusion.

```
export AWS_ACCESS_KEY_ID=YourAccessKey  
export AWS_SECRET_ACCESS_KEY=YourSecretKey
```

```
export AWS_DEFAULT_REGION=YourAWSRegion
```

Si vous utilisez des AWS informations d'identification temporaires, exportez également votre jeton de session :

```
export AWS_SESSION_TOKEN=YourSessionToken
```

.mp4 files

Téléchargez un exemple de vidéo .mp4 à télécharger sur Kinesis Video Streams.

```
wget https://awsj-iot-handson.s3-ap-northeast-1.amazonaws.com/kvs-workshop/sample.mp4
```

Caractéristiques de la vidéo :

- Résolution : 1280 x 720 pixels
- Fréquence d'images - 30 images par seconde
- Durée - 14,0 secondes
- Encodage vidéo - H.264, dans la piste 1
- Images-clés : toutes les 3 secondes, ce qui donne une durée de fragment (également connue sous le nom de taille d'un groupe d'images (GoP)) de 3 secondes, le fragment final étant de 2 secondes.

Exécutez la commande suivante avec le nom du flux que vous avez créé précédemment. Si vous n'avez pas encore créé de stream, consultez [the section called “Création d'un flux vidéo Amazon Kinesis”](#).

```
./kvs_gstreamer_sample YourStreamName ./sample.mp4
```

Sample video from GStreamer

Utilisez la commande suivante pour générer une vidéo à l'aide de GStreamer.

Indiquez à GStreamer où se trouve le plugin kvssink GStreamer. Dans votre répertoire de construction, spécifiez le chemin d'accès au dossier contenant le `libgstkvssink.so` fichier.

À partir de votre répertoire de compilation, exécutez la commande suivante :

```
export GST_PLUGIN_PATH=`pwd`
```

Ce pipeline GStreamer génère un flux vidéo de test en direct avec un modèle de test standard qui s'exécute à 10 images par seconde avec une résolution de 640 x 480 pixels. Une superposition est ajoutée pour afficher l'heure et la date actuelles du système. La vidéo est ensuite encodée dans un H.264 format et des images-clés sont générées au maximum toutes les 10 images, ce qui donne une durée de fragment (également connue sous le nom de taille d'un groupe d'images (GoP)) d'une seconde. kvssink prend le flux H.264-encodé vidéo, le regroupe dans le format conteneur Matroska (MKV) et le télécharge sur votre flux vidéo Kinesis.

Exécutez la commande suivante :

```
gst-launch-1.0 -v videotestsrc is-live=true \  
! video/x-raw,framerate=10/1,width=640,height=480 \  
! clockoverlay time-format="%a %B %d, %Y %I:%M:%S %p" \  
! x264enc bframes=0 key-int-max=10 \  
! h264parse \  
! kvssink stream-name="YourStreamName"
```

Pour arrêter le pipeline GStreamer, sélectionnez la fenêtre du terminal et appuyez sur CTRL+C.

Note

Pour plus d'informations sur l'utilisation du plugin GStreamer pour diffuser des vidéos à partir d'un flux RTSP depuis une caméra ou depuis une caméra USB, consultez. [Exemple : Plug-in GStreamer du SDK Kinesis Video Streams Producer - kvssink](#)

Examiner les objets d'accusé de réception

Pendant le téléchargement, Kinesis Video Streams renvoie des objets d'accusé de réception au client effectuant le téléchargement. Vous devriez les voir imprimés dans la sortie de commande. Voici un exemple :

```
{"EventType": "PERSISTED", "FragmentTimecode": 1711124585823, "FragmentNumber": "1234567890123456789"}
```

Si l'accusé de réception l'EventType est PERSISTED, cela signifie que Kinesis Video Streams a stocké et crypté de manière durable cette partie du contenu multimédia à des fins de récupération, d'analyse et de stockage à long terme.

Pour plus d'informations sur les remerciements, consultez [PutMedia](#)

Consommez des données multimédia

Vous pouvez consommer des données multimédia soit en les visualisant dans la console, soit en créant une application qui lit les données multimédia d'un flux à l'aide du protocole HLS (Hypertext Live Streaming).

Afficher le contenu multimédia dans la console

Dans un autre onglet du navigateur, ouvrez le AWS Management Console. Dans le tableau de bord Kinesis Video Streams, [sélectionnez Streams vidéo](#).

Sélectionnez le nom de votre stream dans la liste des streams. Utilisez la barre de recherche, si nécessaire.

Développez la section Lecture multimédia. Si la vidéo est toujours en cours de téléchargement, elle sera affichée. Si le téléchargement est terminé, sélectionnez la double flèche gauche.

Consommez des données multimédia à l'aide du protocole HLS

Vous pouvez créer une application cliente qui consomme les données d'un flux vidéo Kinesis à l'aide du protocole HLS. Pour plus d'informations sur la création d'une application consommant des données média via HLS, consultez [Lecture vidéo](#).

Transférer vers Kinesis Video Streams

Les bibliothèques de production Amazon Kinesis Video Streams sont un ensemble de bibliothèques du SDK de production Kinesis Video Streams. Le client utilise les bibliothèques et le SDK pour créer l'application intégrée à l'appareil afin de se connecter en toute sécurité à Kinesis Video Streams et de diffuser des données multimédia à visualiser sur la console ou dans les applications clientes en temps réel.

Les données média peuvent être diffusées en continu des manières suivantes :

- En temps réel
- Après l'avoir mis en mémoire tampon pendant quelques secondes
- Après le téléchargement des médias

Après avoir créé un flux Kinesis Video Streams, vous pouvez commencer à y envoyer des données. Vous pouvez utiliser le SDK pour créer un code d'application qui extrait les données vidéo, appelées images, de la source multimédia et les télécharge sur Kinesis Video Streams. Ces applications sont également désignées comme applications producteur.

Les bibliothèques du producteur contiennent les composants suivants :

- [Client producteur Kinesis Video Streams](#)
- [Bibliothèque destinée aux producteurs de Kinesis Video Streams](#)

Client producteur Kinesis Video Streams

Le client Kinesis Video Streams Producer inclut une `KinesisVideoClient` seule classe. Cette classe gère les sources multimédia, reçoit les données des sources et gère le cycle de vie des flux lorsque les données circulent d'une source multimédia vers Kinesis Video Streams. Il fournit également une `MediaSource` interface permettant de définir l'interaction entre Kinesis Video Streams et votre matériel et logiciels propriétaires.

Une source média peut être constituée de presque tout élément. Par exemple, vous pouvez utiliser une source média sous la forme d'une caméra ou d'un microphone. Les sources média ne se limitent pas aux sources audio et vidéo. Par exemple, les journaux de données peuvent être des fichiers texte mais ils peuvent tout de même être envoyés comme flux de données. Vous pourriez également avoir plusieurs appareils photo sur votre téléphone diffusant les données simultanément.

Pour obtenir des données à partir de l'une de ces sources, vous pouvez implémenter l'interface `MediaSource`. Cette interface permet d'autres scénarios pour lesquels nous ne proposons aucune prise en charge intégrée. Par exemple, vous pouvez choisir d'envoyer ce qui suit à Kinesis Video Streams :

- Un flux de données de diagnostic (par exemple, des journaux d'application et des événements)
- Des données de caméras infrarouges, de caméras RADAR ou de caméras de profondeur

Kinesis Video Streams ne fournit pas d'implémentations intégrées pour les appareils de production multimédia tels que les caméras. Pour extraire les données de ces appareils, vous devez mettre en place le code, créant ainsi votre propre implémentation de source média personnalisée. Vous pouvez ensuite enregistrer explicitement vos sources multimédia personnalisées auprès de laquelle `KinesisVideoClient` les données sont transférées vers Kinesis Video Streams.

Le client Kinesis Video Streams Producer est disponible pour les applications Java et Android. Pour plus d'informations, consultez [Utiliser la bibliothèque Java Producer](#) et [Utiliser la bibliothèque du producteur Android](#).

Bibliothèque destinée aux producteurs de Kinesis Video Streams

La bibliothèque de production Kinesis Video Streams est contenue dans le client de production Kinesis Video Streams. La bibliothèque peut également être utilisée directement par ceux qui souhaitent une intégration plus poussée avec Kinesis Video Streams. Il permet l'intégration à partir d'appareils avec des systèmes d'exploitation brevetés, des piles de réseau ou des ressources limitées sur l'appareil.

La bibliothèque de production Kinesis Video Streams implémente la machine à états pour le streaming vers Kinesis Video Streams. Elle fournit des hooks de rappel, qui nécessitent que vous fournissiez votre propre implémentation de transport et que vous traitiez explicitement entrant et sortant du service.

Vous pouvez choisir d'utiliser directement la bibliothèque de production Kinesis Video Streams pour les raisons suivantes :

- Le périphérique sur lequel vous souhaitez exécuter l'application ne dispose d'aucune machine virtuelle Java.
- Vous souhaitez écrire le code d'application dans d'autres langages que celle de Java.

- Vous souhaitez réduire la surcharge de votre code et la limiter au strict minimum d'abstraction, en raison de limitations telles que la mémoire et la puissance de traitement.

Actuellement, la bibliothèque de production Kinesis Video Streams est disponible pour les applications Android, C, C++ et Java. Pour plus d'informations, consultez les langues prises en charge dans les rubriques connexes suivantes.

Découvrez ce que sont les bibliothèques destinées aux producteurs

[Utiliser la bibliothèque Java Producer](#)

[Utiliser la bibliothèque du producteur Android](#)

[Utiliser la bibliothèque de production C++](#)

[Utiliser la bibliothèque C Producer](#)

[Utiliser le SDK du producteur C++ sur Raspberry Pi](#)

Utiliser la bibliothèque Java Producer

Vous pouvez utiliser la bibliothèque de production Java fournie par Amazon Kinesis Video Streams pour écrire du code d'application avec une configuration minimale, afin d'envoyer des données multimédia d'un appareil vers un flux vidéo Kinesis.

Procédez comme suit pour intégrer votre code à Kinesis Video Streams afin que votre application puisse commencer à diffuser des données vers votre flux vidéo Kinesis :

1. Créez une instance de l'objet `KinesisVideoClient`.
2. Créez un objet `MediaSource` en fournissant les informations de la source média. Par exemple, lors de la création d'une caméra en tant que source média, vous fournissez des informations permettant d'identifier la caméra et de spécifier le codage utilisé par la caméra.

Lorsque vous souhaitez commencer le streaming, vous devez créer une source média personnalisée.

3. Inscrivez la source média avec `KinesisVideoClient`.

Après avoir inscrit la source média avec `KinesisVideoClient`, chaque fois que les données deviennent disponibles avec la source média, il invoque `KinesisVideoClient` avec les données.

Procédure : utiliser le SDK Java Producer

Cette procédure explique comment utiliser le client Kinesis Video Streams Java Producer dans votre application Java pour envoyer des données à votre flux vidéo Kinesis.

Ces étapes n'exigent pas que vous disposez d'une source média telle qu'une caméra ou un microphone. Par ailleurs, à des fins de test, le code génère des exemples d'images qui comprennent une série d'octets. Vous pouvez utiliser le même modèle de codage lorsque vous envoyez des données média provenant de sources réelles, par exemple des appareils photos ou des microphones.

La procédure comprend les étapes suivantes :

- [Téléchargez et configurez le code](#)
- [Rédiger et examiner le code](#)
- [Exécutez et vérifiez le code](#)

Conditions préalables

Avant de configurer le SDK Java Producer, assurez-vous de remplir les conditions préalables suivantes :

- Dans l'exemple de code, vous fournissez des informations d'identification en spécifiant un profil que vous avez configuré dans votre fichier de profil AWS d'informations d'identification. Si vous n'y avez pas encore procédé, commencez par définir votre profil à informations d'identification. Pour plus d'informations, voir [Configurer les AWS informations d'identification et la région pour le développement](#) dans le AWS SDK pour Java.

Note

L'exemple Java utilise un `SystemPropertiesCredentialsProvider` objet pour obtenir vos informations d'identification. Le fournisseur extrait ces informations d'identification à partir des propriétés système Java `aws.accessKeyId` et `aws.secretKey`. Vous

définissez ces propriétés système dans votre environnement de développement Java. Pour plus d'informations sur la manière de définir les propriétés du système Java, consultez la documentation relative à votre environnement de développement intégré (IDE).

- Vous `NativeLibraryPath` devez contenir votre `KinesisVideoProducerJNI` dossier, disponible à l'adresse <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp>. L'extension du nom de fichier de ce fichier dépend de votre système d'exploitation :
 - `KinesisVideoProducerJNI.so` pour Linux
 - `KinesisVideoProducerJNI.dylib` pour macOS
 - `KinesisVideoProducerJNI.dll` pour Windows

Note

Pre-built les bibliothèques pour macOS, Ubuntu, Windows et Raspbian sont disponibles sur `src/main/resources/lib` <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java.git> Pour les autres environnements, compilez le [C++](#).

Téléchargez et configurez le code de la bibliothèque Java Producer

Dans cette section de la procédure de bibliothèque Java Producer, vous allez télécharger l'exemple de code Java, importer le projet dans votre IDE Java et configurer les emplacements des bibliothèques.

Pour les prérequis et d'autres informations sur cet exemple, consultez la section [Utilisation de la bibliothèque Java Producer](#).

1. Créez un répertoire, puis clonez l'exemple de code source à partir du GitHub référentiel.

```
git clone https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java
```

2. Ouvrez l'environnement de développement intégré (IDE) Java que vous utilisez (par exemple, [Eclipse](#) ou [JetBrains IntelliJ IDEA](#)) et importez le projet Apache Maven que vous avez téléchargé :
 - Dans IntelliJ IDEA : Choisissez Importer. Naviguez vers le fichier `pom.xml` dans la racine du package téléchargé.

- Dans Eclipse : Choisissez Fichier, Importer, Maven, Projets Maven existants. Puis, accédez au répertoire `kinesis-video-java-demo`.

Pour plus d'informations, consultez la documentation relative à votre IDE.

3. L'exemple de code Java utilise les AWS informations d'identification actuelles. Pour utiliser un autre profil à informations d'identification, recherchez le code suivant dans `DemoAppMain.java` :

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
        Regions.US_WEST_2,
        AuthHelper.getSystemPropertiesCredentialsProvider());
```

Modifiez le code comme suit :

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
        Regions.US_WEST_2,
        new ProfileCredentialsProvider("credentials-profile-name"));
```

Pour plus d'informations, voir [ProfileCredentialsProvider](#) dans la référence AWS SDK pour Java.

Rédiger et examiner le code

Dans cette section de la [procédure de la bibliothèque Java Producer](#), vous allez écrire et examiner l'exemple de code Java que vous avez téléchargé dans la section précédente.

L'application de test Java ([DemoAppMain](#)) présente le modèle de codage suivant :

- Créez une instance de `KinesisVideoClient`.
- Créez une instance de `MediaSource`.
- Inscrivez la `MediaSource` avec le client.
- Démarrez le streaming. Démarrez le `MediaSource` et il commence à envoyer des données au client.

Consultez les sections suivantes pour obtenir des détails.

Créez une instance de KinesisVideoClient

Vous créez l'objet `KinesisVideoClient` en appelant l'opération `createKinesisVideoClient`.

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
        Regions.US_WEST_2,
        AuthHelper.getSystemPropertiesCredentialsProvider());
```

Pour que `KinesisVideoClient` passe des appels au réseau, il a besoin des informations d'identification. Vous transmettez une instance de `SystemPropertiesCredentialsProvider`, qui lit `AWSCredentials` pour le profil par défaut dans le fichier d'informations d'identification :

```
[default]
aws_access_key_id = ABCDEFGHIJKLMNOPQRSTU
aws_secret_access_key = AbCd1234EfGh5678IjKl9012MnOp3456QrSt7890
```

Créez une instance de MediaSource

Pour envoyer des octets à votre flux vidéo Kinesis, vous devez produire les données. Amazon Kinesis Video Streams fournit `MediaSource` l'interface, qui représente la source de données.

Par exemple, la bibliothèque Java Kinesis Video Streams fournit `ImageFileMediaSource` l'implémentation de `MediaSource` l'interface. Cette classe lit uniquement les données d'une série de fichiers multimédia plutôt que d'un flux vidéo Kinesis, mais vous pouvez l'utiliser pour tester le code.

```
final MediaSource bytesMediaSource = createImageFileMediaSource();
```

Enregistrez le MediaSource auprès du client

Inscrivez la source média que vous avez créée avec le `KinesisVideoClient` afin qu'elle prenne connaissance du client (et puisse ensuite envoyer des données au client).

```
kinesisVideoClient.registerMediaSource(mediaSource);
```

Démarrez la source multimédia

Démarrez la source multimédia afin qu'elle puisse commencer à générer des données et à les envoyer au client.

```
bytesMediaSource.start();
```

nettoyer des ressources ;

Afin d'éviter les fuites de mémoire, procédez comme suit pour annuler l'enregistrement d'une source multimédia auprès du client et libérer le client.

```
try {
    kinesisisVideoClient.unregisterMediaSource(mediaSource);
    kinesisisVideoClient.free();
} catch (final KinesisVideoException e) {
    throw new RuntimeException(e);
}
```

Si vous avez ajouté des éléments au cache à l'aide [CachedInfoMultiAuthServiceCallbacks](#), par exemple :

```
serviceCallbacks.addStreamInfoToCache(streamName, streamInfo);
serviceCallbacks.addStreamingEndpointToCache(streamName, dataEndpoint);
```

Videz le cache lorsque vous avez terminé :

```
serviceCallbacks.removeStreamFromCache(streamName);
```

Exécutez et vérifiez le code

Pour exécuter le harnais de test Java pour la [bibliothèque Java Producer](#), procédez comme suit.

1. Sélectionnez DemoAppMain.
2. Choisissez Exécuter, Exécuter « DemoAppMain ».
3. Ajoutez vos informations d'identification aux arguments JVM pour l'application :

- Pour les informations d' AWS identification non temporaires : "-
Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey}
-Djava.library.path={NativeLibraryPath}"
 - Pour les AWS informations d'identification temporaires : "-
Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey}
-Daws.sessionToken={YourAwsSessionToken} -
Djava.library.path={NativeLibraryPath}"
4. Connectez-vous à la console [Kinesis Video Streams AWS Management Console](#) et ouvrez-la.

Sur la page Gérer les flux, choisissez votre flux.
 5. L'échantillon vidéo sera lu dans le lecteur intégré. Vous devrez peut-être patienter un court instant (jusqu'à dix secondes dans des conditions de processeur et de bande passante classiques) pendant que les images s'accumulent avant l'affichage de la vidéo.

L'exemple de code crée un flux. Lorsque l'élément `MediaSource` du code démarre, il commence à envoyer des échantillons d'images à `KinesisVideoClient`. Le client envoie ensuite les données vers votre flux vidéo Kinesis.

Utiliser la bibliothèque du producteur Android

Vous pouvez utiliser la bibliothèque de production Android fournie par Amazon Kinesis Video Streams pour écrire du code d'application, avec une configuration minimale, afin d'envoyer des données multimédia d'un appareil Android vers un flux vidéo Kinesis.

Procédez comme suit pour intégrer votre code à Kinesis Video Streams afin que votre application puisse commencer à diffuser des données vers votre flux vidéo Kinesis :

1. Créez une instance de l'objet `KinesisVideoClient`.
2. Créez un objet `MediaSource` en fournissant les informations de la source média. Par exemple, lors de la création d'une caméra en tant que source média, vous fournissez des informations permettant d'identifier la caméra et de spécifier le codage utilisé par la caméra.

Lorsque vous souhaitez commencer le streaming, vous devez créer une source média personnalisée.

Procédure : utilisez le SDK Android Producer

Cette procédure explique comment utiliser le client Kinesis Video Streams Android Producer dans votre application Android pour envoyer des données vers votre flux vidéo Kinesis.

La procédure comprend les étapes suivantes :

- [the section called “Conditions préalables”](#)
- [the section called “Téléchargez et configurez le code”](#)
- [the section called “Examinez le code”](#)
- [the section called “Exécutez et vérifiez le code”](#)

Conditions préalables

Nous vous recommandons [Android Studio](#) pour la vérification, la modification et l'exécution du code de l'application. Nous vous recommandons d'utiliser la dernière version stable.

Dans l'exemple de code, vous fournissez les informations d'identification Amazon Cognito.

Suivez ces procédures pour configurer un groupe d'utilisateurs et un pool d'identités Amazon Cognito.

- [Configuration d'un groupe d'utilisateurs](#)
- [Configuration d'un pool d'identités](#)

Configuration d'un groupe d'utilisateurs

Pour configurer un groupe d'utilisateurs

1. Connectez-vous à la [console Amazon Cognito](#) et vérifiez que la région est correcte.
2. Dans le menu de navigation de gauche, choisissez Groupes d'utilisateurs.
3. Dans la section Groupes d'utilisateurs, choisissez Créer un groupe d'utilisateurs.
4. Complétez les sections suivantes :
 - a. Étape 1 : Configuration de l'expérience de connexion - Dans la section des options de connexion au groupe d'utilisateurs de Cognito, sélectionnez les options appropriées.

Sélectionnez Suivant.

- b. Étape 2 : Configuration des exigences de sécurité - Sélectionnez les options appropriées.
Sélectionnez Suivant.
 - c. Étape 3 : Configuration de l'expérience d'inscription - Sélectionnez les options appropriées.
Sélectionnez Suivant.
 - d. Étape 4 : Configuration de la livraison des messages - Sélectionnez les options appropriées.
Dans le champ de sélection du rôle IAM, sélectionnez un rôle existant ou créez-en un nouveau.
Sélectionnez Suivant.
 - e. Étape 5 : Intégrez votre application - Sélectionnez les options appropriées.
Dans le champ Client initial de l'application, sélectionnez Client confidentiel.
Sélectionnez Suivant.
 - f. Étape 6 : révision et création : passez en revue les sélections des sections précédentes, puis choisissez Créer un groupe d'utilisateurs.
5. Sur la page Groupes d'utilisateurs, sélectionnez le pool que vous venez de créer.
Copiez l'ID du groupe d'utilisateurs et notez-le pour plus tard. Dans le `awsconfiguration.json` dossier, c'est `CognitoUserPool.Default.PoolId`.
 6. Sélectionnez l'onglet Intégration des applications et rendez-vous au bas de la page.
 7. Dans la section Liste des clients de l'application, choisissez le nom du client de l'application que vous venez de créer.
Copiez l'ID client et notez-le pour plus tard. Dans le `awsconfiguration.json` dossier, c'est `CognitoUserPool.Default.AppClientId`.
 8. Affichez le secret du client et notez-le pour plus tard. Dans le `awsconfiguration.json` dossier, c'est `CognitoUserPool.Default.AppClientSecret`.

Configuration d'un pool d'identités

Pour configurer un pool d'identités

1. Connectez-vous à la [console Amazon Cognito](#) et vérifiez que la région est correcte.

2. Dans le menu de navigation de gauche, choisissez Identity pools.
3. Choisissez Créer un groupe d'identités.
4. Configurez le pool d'identités.
 - a. Étape 1 : Configurer la confiance du pool d'identités - Complétez les sections suivantes :
 - Accès utilisateur : sélectionnez Accès authentifié
 - Sources d'identité authentifiées : sélectionnez le groupe d'utilisateurs Amazon Cognito

Sélectionnez Suivant.
 - b. Étape 2 : Configuration des autorisations - Dans la section Rôle authentifié, renseignez les champs suivants :
 - Rôle IAM - Sélectionnez Créer un nouveau rôle IAM
 - Nom du rôle IAM : entrez un nom et notez-le pour une étape ultérieure.


Sélectionnez Suivant.
 - c. Étape 3 : Connecter les fournisseurs d'identité - Dans la section Détails du groupe d'utilisateurs, complétez les champs suivants :
 - ID du groupe d'utilisateurs : sélectionnez le groupe d'utilisateurs que vous avez créé précédemment.
 - ID du client de l'application : sélectionnez l'ID du client de l'application que vous avez créé précédemment.

Sélectionnez Suivant.
 - d. Étape 4 : Configuration des propriétés - Entrez un nom dans le champ Nom du pool d'identités.

Sélectionnez Suivant.
 - e. Étape 5 : révision et création : passez en revue vos sélections dans chacune des sections, puis sélectionnez Créer un pool d'identités.
5. Sur la page Groupes d'identités, sélectionnez votre nouveau pool d'identités.

Copiez l'ID du pool d'identités et notez-le pour plus tard. Dans le `awsconfiguration.json` dossier, `c'estCredentialsProvider.CognitoIdentity.Default.PoolId`.

6. Mettez à jour les autorisations pour le rôle IAM.
 - a. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
 - b. Dans le menu de navigation de gauche, sélectionnez Rôles.
 - c. Recherchez et sélectionnez le rôle que vous avez créé ci-dessus.

 Note

Utilisez la barre de recherche, si nécessaire.

- d. Sélectionnez la politique d'autorisation jointe.

Tâche de sélection Modifier.

- e. Sélectionnez l'onglet JSON et remplacez la politique par la suivante :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cognito-identity:*",
        "kinesisvideo:*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Sélectionnez Suivant.

- f. Cochez la case à côté de Définir cette nouvelle version comme version par défaut si elle n'est pas déjà sélectionnée.

Cliquez sur Enregistrer les modifications.

Téléchargez et configurez le code de la bibliothèque du producteur Android

Dans cette section de la procédure de la bibliothèque Android Producer, vous allez télécharger l'exemple de code Android et ouvrir le projet dans Android Studio.

Pour les prérequis et d'autres informations concernant cet exemple, consultez la section [Utilisation de la bibliothèque du producteur Android](#).

1. Créez un répertoire, puis clonez-le AWS Mobile SDK pour Android à partir du GitHub référentiel.

```
git clone https://github.com/aws-labs/aws-sdk-android-samples
```

2. Ouvrez [Android Studio](#).
3. Dans l'écran qui s'affiche, choisissez Open an existing Android Studio project.
4. Accédez au répertoire `aws-sdk-android-samples/AmazonKinesisVideoDemoApp` et choisissez OK.
5. Ouvrez le fichier `AmazonKinesisVideoDemoApp/src/main/res/raw/awsconfiguration.json`.

Dans le `CredentialsProvider` nœud, indiquez l'ID du pool d'identités indiqué dans la procédure Pour configurer un pool d'identités de la section [Conditions préalables](#), ainsi que le votre Région AWS (par exemple, **us-west-2**).

Dans le `CognitoUserPool` nœud, indiquez le secret du client de l'application, l'identifiant du client de l'application et l'identifiant du pool indiqués dans la procédure Pour configurer un groupe d'utilisateurs de la section [Conditions préalables](#), et indiquez le votre Région AWS (par exemple, **us-west-2**).

6. Votre fichier `awsconfiguration.json` est alors similaire à ce qui suit :

```
{
  "Version": "1.0",
  "CredentialsProvider": {
    "CognitoIdentity": {
```

```
        "Default": {
            "PoolId": "us-west-2:01234567-89ab-cdef-0123-456789abcdef",
            "Region": "us-west-2"
        }
    },
    "IdentityManager": {
        "Default": {}
    },
    "CognitoUserPool": {
        "Default": {
            "AppClientSecret": "abcdefghijklmnopqrstuvwxy0123456789abcdefghijklmnop",
            "AppClientId": "0123456789abcdefghijklmnop",
            "PoolId": "us-west-2_qRsTuVwXy",
            "Region": "us-west-2"
        }
    }
}
```

7. Mettez-la à jour `AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/KinesisVideoDemoApp.java` avec votre région (dans l'exemple suivant, elle est définie sur `US_WEST_2`) :

```
public class KinesisVideoDemoApp extends Application {
    public static final String TAG = KinesisVideoDemoApp.class.getSimpleName();
    public static Regions KINESIS_VIDEO_REGION = Regions.US_WEST_2;
```

Pour plus d'informations sur les Région AWS constantes, reportez-vous à la section [Régions](#).

Examinez le code

Dans cette section de la [procédure de la bibliothèque Android Producer](#), vous allez examiner l'exemple de code.

L'application de test Android (`AmazonKinesisVideoDemoApp`) présente le modèle de codage suivant :

- Créez une instance de `KinesisVideoClient`.
- Créez une instance de `MediaSource`.

- Démarrez le streaming. Démarrez le `MediaSource`, et il commence à envoyer des données au client.

Consultez les sections suivantes pour obtenir des détails.

Créez une instance de `KinesisVideoClient`

Vous créez l'objet [KinesisVideoClient](#) en appelant l'opération [createKinesisVideoClient](#).

```
mKinesisVideoClient = KinesisVideoAndroidClientFactory.createKinesisVideoClient(
    getActivity(),
    KinesisVideoDemoApp.KINESIS_VIDEO_REGION,
    KinesisVideoDemoApp.getCredentialsProvider());
```

Pour que `KinesisVideoClient` passe des appels au réseau, il a besoin des informations d'identification. Vous transmettez une instance de `AWSCredentialsProvider`, qui lit vos informations d'identification Amazon Cognito dans le `awsconfiguration.json` fichier que vous avez modifié dans la section précédente.

Créez une instance de `MediaSource`

Pour envoyer des octets à votre flux vidéo Kinesis, vous devez produire les données. Amazon Kinesis Video Streams fournit [MediaSource](#) l'interface, qui représente la source de données.

Par exemple, la bibliothèque Android Kinesis Video Streams fournit [AndroidCameraMediaSource](#) l'implémentation de `MediaSource` l'interface. Cette classe lit des données à partir de l'une des caméras de l'appareil.

Dans l'exemple de code suivant (provenant du fichier [fragment/StreamConfigurationFragment.java](#)), la configuration de la source multimédia est créée :

```
private AndroidCameraMediaSourceConfiguration getCurrentConfiguration() {
    return new AndroidCameraMediaSourceConfiguration(
        AndroidCameraMediaSourceConfiguration.builder()
            .withCameraId(mCamerasDropdown.getSelectedItem().getCameraId())

            .withEncodingMimeType(mMimeTypeDropdown.getSelectedItem().getMimeType())

            .withHorizontalResolution(mResolutionDropdown.getSelectedItem().getWidth())

            .withVerticalResolution(mResolutionDropdown.getSelectedItem().getHeight())
```

```
        .withCameraFacing(mCamerasDropdown.getSelectedItem().getCameraFacing())
        .withIsEncoderHardwareAccelerated(
mCamerasDropdown.getSelectedItem().isEncoderHardwareAccelerated())
        .withFrameRate(FRAMERATE_20)
        .withRetentionPeriodInHours(RETENTION_PERIOD_48_HOURS)
        .withEncodingBitRate(BITRATE_384_KBPS)
        .withCameraOrientation(-
mCamerasDropdown.getSelectedItem().getCameraOrientation())

        .withNalAdaptationFlags(StreamInfo.NalAdaptationFlags.NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS)
        .withIsAbsoluteTimecode(false));
    }
```

Dans l'exemple de code suivant (provenant du fichier [fragment/StreamingFragment.java](#)), la source multimédia est créée :

```
mCameraMediaSource = (AndroidCameraMediaSource) mKinesisVideoClient
    .createMediaSource(mStreamName, mConfiguration);
```

Démarrez la source multimédia

Démarrez la source média afin qu'elle puisse commencer à générer des données et les envoyer au client. L'exemple de code suivant provient du fichier [fragment/StreamingFragment.java](#) :


```
mCameraMediaSource.start();
```

Exécutez et vérifiez le code

Pour exécuter l'exemple d'application Android pour la [bibliothèque du producteur Android](#), procédez comme suit.

1. Connectez-vous à un appareil Android.
2. Choisissez Run, Run..., puis Edit configurations
3. Choisissez l'icône plus (+), application Android. Dans le champ Nom, saisissez **AmazonKinesisVideoDemoApp**. Dans le menu déroulant Module, choisissez AmazonKinesisVideoDemoApp Choisissez OK.
4. Choisissez Run, Run.

5. Dans l'écran Select Deployment Target, sélectionnez votre appareil connecté, puis choisissez OK.
6. Dans l'AWSKinesisVideoDemoAppapplication de l'appareil, choisissez Créer un nouveau compte.
7. Entrez les valeurs de USERNAME, Password, Given name, Email address et Phone number, puis choisissez Sign up.

 Note


Ces valeurs sont soumises aux contraintes suivantes :

- Password : Doit contenir des lettres en majuscules et minuscules, des nombres et des caractères spéciaux. Vous pouvez modifier ces contraintes sur la page de votre groupe d'utilisateurs sur la console [Amazon Cognito](#).
- Email address : Doit être une adresse valide afin que vous puissiez recevoir un code de confirmation.
- Numéro de téléphone : Doit être au format suivant : **+<Country code><Number>**, par exemple, **+12065551212**.

8. Entrez le code que vous recevez par e-mail, puis cliquez sur Confirmer. Choisissez OK.
9. Sur la page suivante, conservez les valeurs par défaut et choisissez Stream.
10. Connectez-vous à la console [Kinesis Video Streams AWS Management Console](#) et ouvrez-la dans la région USA Ouest (Oregon).

Dans la page Manage Streams, choisissez demo-stream.

11. La vidéo en streaming est lue dans le lecteur intégré. Vous devrez peut-être patienter un court instant (jusqu'à dix secondes dans des conditions de processeur et de bande passante classiques) pendant que les images s'accumulent avant l'affichage de la vidéo.

 Note

Si l'écran de l'appareil effectue une rotation (par exemple, de portrait à paysage), l'application cesse la diffusion en continu de la vidéo.

L'exemple de code crée un flux. Lorsque l'élément `MediaSource` démarre dans le code, il commence à envoyer des images de la caméra vers le `KinesisVideoClient`. Le client envoie ensuite les données vers un flux vidéo Kinesis nommé `demo-stream`.

Utiliser la bibliothèque de production C++

Vous pouvez utiliser la bibliothèque de production C++ fournie par Amazon Kinesis Video Streams pour écrire du code d'application afin d'envoyer des données multimédia d'un appareil vers un flux vidéo Kinesis.

Modèle d'objet

La bibliothèque C++ fournit les objets suivants pour gérer l'envoi de données vers un flux vidéo Kinesis :

- `KinesisVideoProducer`: contient des informations sur votre source multimédia et vos AWS informations d'identification, et gère des rappels pour signaler les événements liés à Kinesis Video Streams.
- `KinesisVideoStream`: représente le flux vidéo Kinesis. Contient des informations sur les paramètres du flux vidéo, tels que le nom, la durée de conservation des données et le type de contenu multimédia.

Insérez du contenu multimédia dans le stream

Vous pouvez utiliser les méthodes fournies par la bibliothèque C++ (par exemple, `PutFrame`) pour insérer des données dans l'`KinesisVideoStream` objet. La bibliothèque gère ensuite l'état interne des données, ce qui peut inclure les tâches suivantes :

- Réalisation de l'authentification.
- Supervision de la latence du réseau. Si la latence est trop élevée, la bibliothèque peut choisir d'abandonner des images.
- Suivi de l'état du streaming en cours.

Interfaces de rappel

Cette couche propose un ensemble d'interfaces de rappel, ce qui lui permet de consulter la couche d'application. Parmi ces interfaces de rappel, on trouve :

- Interface de rappel de service (**CallbackProvider**) : la bibliothèque invoque les événements obtenus via cette interface lorsqu'elle crée un flux, obtient une description de flux et supprime un flux.
- Client-ready state or low storage events interface (**ClientCallbackProvider**) : La bibliothèque invoque des événements sur cette interface lorsque le client est prêt ou lorsqu'elle détecte qu'il n'y a peut-être plus de stockage ou de mémoire disponible.
- Interface de rappel des événements de flux (**StreamCallbackProvider**) : la bibliothèque invoque des événements sur cette interface lorsque des événements de flux se produisent, tels que l'entrée du flux dans l'état prêt, la perte d'images ou des erreurs de flux.

Kinesis Video Streams fournit des implémentations par défaut pour ces interfaces. Vous pouvez également fournir votre propre implémentation personnalisée, par exemple, si vous avez besoin d'une logique réseau personnalisée ou si vous souhaitez exposer l'interface utilisateur à un faible niveau de stockage.

Pour plus d'informations sur les rappels dans les bibliothèques du producteur, consultez [Rappels du SDK pour les producteurs](#).

Procédure : utiliser le SDK du producteur C++

Cette procédure explique comment utiliser le client Kinesis Video Streams et les sources multimédia dans une application C++ pour envoyer des données à votre flux vidéo Kinesis.

La procédure comprend les étapes suivantes :

Conditions préalables

Avant de configurer le SDK C++ Producer, assurez-vous de remplir les conditions préalables suivantes :

- Informations d'identification : dans l'exemple de code, vous fournissez des informations d'identification en spécifiant un profil que vous avez configuré dans votre fichier de profil AWS d'informations d'identification. Si vous n'y avez pas encore procédé, commencez par définir votre profil à informations d'identification.

Pour plus d'informations, voir [Configurer les AWS informations d'identification et la région pour le développement](#).

- Intégration du magasin de certificats : la bibliothèque du producteur Kinesis Video Streams doit établir un lien de confiance avec le service qu'elle appelle. Cela se fait en validant les autorités de certification (CA) dans le magasin de certificats public. Sur Linux-based les modèles, ce magasin se trouve dans le répertoire `/etc/ssl/`.

Télécharger le certificat depuis l'emplacement suivant vers votre magasin de certificats :

<https://www.amazontrust.com/repository/SFSRootCAG2.pem>

- Installez les dépendances de build suivantes pour macOS :
 - [Autoconf 2.69](#) (License GPLv3+/ Autoconf : GNU GPL version 3 ou ultérieure)
 - [CMake 3.7 ou 3.8](#)
 - [Pkg-Config](#)
 - xCode (macOS) / clang / gcc (xcode-select version 2347)
 - Kit de développement Java (JDK) (pour compilation Java JNI)
 - [Lib-Pkg](#)
- Installez les dépendances de build suivantes pour Ubuntu :
 - Git : `sudo apt install git`
 - [CMake](#) : `sudo apt install cmake`
 - G++ : `sudo apt install g++`
 - configuration du fichier : `sudo apt install pkg-config`
 - OpenJDK : `sudo apt install openjdk-8-jdk`

Note

Cela n'est nécessaire que si vous créez une interface native Java (JNI).

- Définissez la variable d'environnement : `JAVA_HOME export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

Téléchargez et configurez le code de la bibliothèque C++ Producer

Pour plus d'informations sur le téléchargement et la configuration de la bibliothèque de production C++, consultez [Amazon Kinesis Video Streams CPP Producer, GStreamer Plugin et JNI](#).

Pour les prérequis et plus d'informations sur cet exemple, consultez [the section called "C++"](#).

Arguments CMake

Vous trouverez ci-dessous un tableau de référence pour les arguments SDK-specific CMake du producteur C++. Vous pouvez également transmettre les [options standard de CMake](#) à CMake.

Important

Elles sont toutes facultatives.

Drapeaux pour inclure ou exclure certaines fonctionnalités

| Argument CMake | Type | Par défaut | Explication |
|------------------------|---------|------------|---|
| BUILD_DEPENDENCIES | Booléen | ON | Créez des dépendances à partir de la source. Dans le cas contraire, utilisez les dépendances déjà installées sur le système. Si l'une des dépendances requises est introuvable, une erreur sera renvoyée. |
| BUILD_GSTREAMER_PLUGIN | Booléen | OFF | Construit le plugin kvssink GStreamer. |
| BUILD_JNI | Booléen | OFF | Construit l'interface native Java (JNI) pour pouvoir appeler ce code depuis un environnement d'exécution Java. |
| ALIGNED_MEMORY_MODEL | Booléen | OFF | Si les allocations de mémoire doivent |

| Argument CMake | Type | Par défaut | Explication |
|--------------------|---------|------------|--|
| | | | être alignées sur des limites de 8 octets. Certaines architectures n'autorisent pas l'accès à la mémoire non aligné. |
| CONSTRAINED_DEVICE | Booléen | OFF | Non-Windows uniquement. Lorsque cette option est activée, définit la taille de la pile de threads sur 0.5 MiB. Nécessaire pour les versions d' Alpine Linux . Dans le cas contraire, c'est le système d'exploitation par défaut qui est utilisé. |
| BUILD_STATIC | Booléen | OFF | Créez des bibliothèques et des exécutables partagés (OFF) ou statiques (ON). |
| ADD_MUCLIBC | Booléen | OFF | Lien vers UClibc au lieu de la bibliothèque C standard, qui est une bibliothèque standard C plus petite conçue pour les systèmes embarqués. |

| Argument CMake | Type | Par défaut | Explication |
|-----------------------------|--------|--------------------------|---|
| OPEN_SRC_ INSTALL_PREFIX | String | .. /ouvert/ source/local | Emplacement où installer les dépendances open source, si vous créez à partir des sources. |

Drapeaux pour la compilation croisée

Important

Définissez ces paramètres si les architectures de processeur de votre machine cible et de votre machine hôte sont différentes.

| Argument CMake | Type | Par défaut | Explication |
|--------------------------|--------|------------|---|
| BUILD_LOG 4CPLUS_HOST | String | "" | Créez la log4cplus dépendance pour l'architecture de processeur spécifiée . S'il n'est pas défini, log4cplus il détectera automatiquement et utilisera l'architecture du processeur de la machine hôte. |
| BUILD_OPENSSL_PLATFORM | String | "" | Créez la OpenSSL dépendance pour l'architecture de processeur spécifiée . S'il n'est pas défini, OpenSSL il détectera |

| Argument CMake | Type | Par défaut | Explication |
|----------------|------|------------|---|
| | | | automatiquement et utilisera l'architecture du processeur de la machine hôte. |

Drapeaux liés aux tests

| Argument CMake | Type | Par défaut | Explication |
|-------------------|---------|------------|--|
| BUILD_TEST | Booléen | OFF | Créer les tests unitaires et d'intégration. Pour exécuter tous les tests, exécutez-les <code>./tst/producerTest</code> depuis le répertoire de construction. AWS Des informations d'identification sont nécessaires pour exécuter les tests. |
| CODE_COVERAGE | Booléen | OFF | Disponible uniquement pour les GNU/Clang compilateurs. Activez la collecte de la couverture du code avec gcov et la génération de rapports. |
| COMPILER_WARNINGS | Booléen | OFF | Disponible uniquement pour les GNU/Clang compilateurs. Activez |

| Argument CMake | Type | Par défaut | Explication |
|------------------------------|---------|------------|--|
| | | | tous les avertissements du compilateur. |
| ADDRESS_SANITIZER | Booléen | OFF | Disponible uniquement pour les GNU/Clang compilateurs. Construisez avec AddressSanitizer . |
| MEMORY_SANITIZER | Booléen | OFF | Disponible uniquement pour les GNU/Clang compilateurs. Construisez avec MemorySanitizer . |
| THREAD_SANITIZER | Booléen | OFF | Disponible uniquement pour les GNU/Clang compilateurs. Construisez avec ThreadSanitizer . |
| UNDEFINED_BEHAVIOR_SANITIZER | Booléen | OFF | Disponible uniquement pour les GNU/Clang compilateurs. Construisez avec UndefinedBehaviorSanitizer . |

Pour utiliser ces arguments CMake, transmettez-les sous forme de liste de `-Dkey=value` paires séparées par des espaces après la commande. `cmake ..` Par exemple :

```
cmake .. -DBUILD_GSTREAMER_PLUGIN=ON -DBUILD_DEPENDENCIES=OFF -
DALIGNED_MEMORY_MODEL=ON
```

CMake recherchera la chaîne d'outils du compilateur en suivant la \$PATH variable. Avant d'exécuter CMake, définissez les variables d'CXEnvironnement CC et pour définir explicitement la chaîne d'outils à utiliser pour la compilation croisée.

Rédiger et examiner le code

Dans cette section du [the section called "C++"](#), vous examinez le code du harnais de test C++ (`tst/ProducerTestFixture.h` et d'autres fichiers). Vous avez téléchargé ce code dans la section précédente.

L'exemple de Plateforme indépendante C++ suivant illustre le modèle de codage suivant :

- Créez une instance de `KinesisVideoProducer` pour accéder à Kinesis Video Streams.
- Créez une instance de `KinesisVideoStream`. Cela crée un flux vidéo Kinesis dans votre Compte AWS si aucun flux du même nom n'existe déjà.
- Appelez `putFrame` sur le `KinesisVideoStream` pour chaque image de données à mesure de sa disponibilité, pour l'envoyer au flux.

Les sections suivantes fournissent plus d'informations sur ce modèle de codage.

Créez une instance de `KinesisVideoProducer`

Vous créez l'objet `KinesisVideoProducer` en appelant la méthode `KinesisVideoProducer::createSync`. L'exemple suivant crée l'`KinesisVideoProducer` dans le fichier `ProducerTestFixture.h` :

```
kinesis_video_producer_ = KinesisVideoProducer::createSync(move(device_provider_),
    move(client_callback_provider_),
    move(stream_callback_provider_),
    move(credential_provider_),
    defaultRegion_);
```

La méthode `createSync` prend les paramètres suivants :

- Un objet `DeviceInfoProvider` qui renvoie un objet `DeviceInfo` contenant des informations relatives à la configuration de l'appareil ou du stockage.

Note

Vous configurez la taille de votre magasin de contenu à l'aide du paramètre `deviceInfo.storageInfo.storageSize`. Vos flux de contenu partagent le magasin de contenu. Pour déterminer les exigences en termes de taille de stockage, multipliez la taille d'image moyenne par le nombre d'images stockées pendant la durée maximale pour tous les flux. Ensuite, multipliez le résultat par 1,2 pour prendre en compte la défragmentation. Par exemple, supposez que votre application a la configuration suivante :

- Trois flux
- 3 minutes de durée maximale
- Chaque flux comprend 30 images par seconde (FPS)
- Chaque image a une taille de 10 000 Ko

L'exigence de stockage de contenu pour cette application est de $3 \text{ (flux)} * 3 \text{ (minutes)} * 60 \text{ (secondes par minute)} * 10000 \text{ (ko)} * 1,2 \text{ (tolérance de défragmentation)} = 194,4 \text{ Mo} \sim 200 \text{ Mo}$.

- Un objet `ClientCallbackProvider` objet qui renvoie les pointeurs de fonction rapportant les événements spécifiques au client.
- Un objet `StreamCallbackProvider` qui renvoie les pointeurs de fonction qui sont appelés lors d'événements spécifiques au flux.
- Un `CredentialProvider` objet qui donne accès aux variables d'environnement AWS d'identification.
- Le Région AWS (« us-west-2 »). Le point de terminaison de service est déterminé selon la région.

Créez une instance de `KinesisVideoStream`

Vous créez l'objet `KinesisVideoStream` en appelant la méthode `KinesisVideoProducer::CreateStream` avec un paramètre `StreamDefinition`. L'exemple crée le flux `KinesisVideoStream` dans le fichier `ProducerTestFixture.h` avec le type de piste vidéo et l'ID de piste 1 :

```
auto stream_definition = make_unique<StreamDefinition>(stream_name,  
                                                       hours(2),  
                                                       tags,  
                                                       "");
```

```
        STREAMING_TYPE_REALTIME,  
        "video/h264",  
        milliseconds::zero(),  
        seconds(2),  
        milliseconds(1),  
        true,  
        true,  
        true);  
return kinesis_video_producer_->createStream(move(stream_definition));
```

L'objet `StreamDefinition` comporte les champs suivants :

- Nom du flux.
- Durée de conservation.
- Balises pour le flux. Ces balises peuvent être utilisées par des applications de consommateurs pour trouver le bon flux ou pour obtenir plus d'informations sur le flux. Les balises peuvent également être affichés dans la AWS Management Console.
- AWS KMS clé de chiffrement pour le flux. Pour de plus amples informations, veuillez consulter [the section called "Protection des données"](#).
- Type de streaming. Actuellement, la seule valeur valide est `STREAMING_TYPE_REALTIME`.
- Type de contenu média.
- Latence de média. Cette valeur n'est pas utilisée actuellement et doit être définie sur 0.
- Durée de lecture de chaque fragment.
- Mise à l'échelle du timecode média.
- Si le média utilise la fragmentation des images de la clé.
- Si le média utilise des timecodes.
- Si le média utilise des fragments absolus.

Ajouter une piste audio au flux vidéo Kinesis

Vous pouvez ajouter les détails d'une piste audio à la définition d'un flux de piste vidéo en utilisant la `addTrack` méthode suivante `StreamDefinition` :

```
stream_definition->addTrack(DEFAULT_AUDIO_TRACKID, DEFAULT_AUDIO_TRACK_NAME,  
    DEFAULT_AUDIO_CODEC_ID, MKV_TRACK_INFO_TYPE_AUDIO);
```

La `addTrack` méthode nécessite les paramètres suivants :

- Identifiant de piste (identique à celui de l'audio). Il doit s'agir d'une valeur unique différente de zéro.
- User-defined nom de piste (par exemple, « audio » pour la piste audio).
- Identifiant du codec pour cette piste (par exemple, pour la piste audio « A_AAC »).
- Type de piste (par exemple, utilisez la valeur enum de `MKV_TRACK_INFO_TYPE_AUDIO` pour l'audio).

Si vous avez des données privées codec pour la piste audio, vous pouvez les transmettre lors de l'appel de la fonction `addTrack`. Vous pouvez également envoyer les données privées du codec après avoir créé l' `KinesisVideoStream` objet lors de l'appel de la méthode `start` in `KinesisVideoStream`.

Insérer une image dans le flux vidéo Kinesis

Vous insérez du contenu multimédia dans le flux vidéo Kinesis en utilisant `KinesisVideoStream::putFrame`, en transmettant un `Frame` objet contenant l'en-tête et les données multimédia. L'exemple appelle `putFrame` dans le fichier `ProducerApiTest.cpp` :

```
frame.duration = FRAME_DURATION_IN_MICROS * HUNDREDS_OF_NANOS_IN_A_MICROSECOND;
frame.size = sizeof(frameBuffer_);
frame.frameData = frameBuffer_;
memset(frame.frameData, 0x55, frame.size);

while (!stop_producer_) {
    // Produce frames
    timestamp = std::chrono::duration_cast<std::chrono::nanoseconds>(
        std::chrono::system_clock::now().time_since_epoch()).count() /
    DEFAULT_TIME_UNIT_IN_NANOS;
    frame.index = index++;
    frame.decodingTs = timestamp;
    frame.presentationTs = timestamp;

    // Key frame every 50th
    frame.flags = (frame.index % 50 == 0) ? FRAME_FLAG_KEY_FRAME : FRAME_FLAG_NONE;
    ...

    EXPECT_TRUE(kinesis_video_stream->putFrame(frame));
}
```

Note

L'exemple de producteur C++ précédent envoie une mémoire tampon de données de test. Dans une application réelle, vous devez obtenir le tampon et la taille de l'image depuis les données de l'image à partir d'une source média (par exemple, une caméra).

L'objet `Frame` comporte les champs suivants :

- Index d'image. Ce doit être une valeur croissante de manière monotone.
- Indicateurs associés à l'image. Par exemple, si l'encodeur a été configuré pour produire un image de clé, cette image sera attribuée à l'indicateur `FRAME_FLAG_KEY_FRAME`.
- Horodatage de décodage.
- Horodatage de la présentation.
- Durée de l'image (jusqu'à 100 unités ns).
- Taille de l'image, en octets.
- Données de l'image.

Pour plus d'informations sur le format du cadre, consultez [the section called "Modèle de données"](#).

Placer un `KinesisVideoFrame` dans une piste spécifique de `KinesisVideoStream`

Vous pouvez utiliser la `PutFrameHelper` classe pour placer des données de trame dans une piste spécifique. Tout d'abord, appelez le `getFrameDataBuffer` pour obtenir un pointeur vers l'un des tampons préalloués afin de renseigner les `KinesisVideoFrame` données. Ensuite, vous pouvez appeler le `putFrameMultiTrack` pour envoyer le `KinesisVideoFrame` avec la valeur booléenne pour indiquer le type de données de trame. Utilisez la valeur `true` s'il s'agit de données vidéo ou `false` si la trame contient des données audio. La `putFrameMultiTrack` méthode utilise un mécanisme de mise en file d'attente pour garantir que les fragments MKV conservent des horodatages d'image qui augmentent de façon monotone et que deux fragments ne se chevauchent pas. Par exemple, l'horodatage MKV de la première image d'un fragment doit toujours être supérieur à l'horodatage MKV de la dernière image du fragment précédent.

`PutFrameHelper` comporte les champs suivants :

- Nombre maximal d'images audio dans la file d'attente.
- Nombre maximum d'images vidéo dans la file d'attente.

- Taille à allouer pour une seule image audio.
- Taille à allouer pour une seule image vidéo.

Métriques d'accès et journalisation des métriques

Le SDK C++ Producer inclut des fonctionnalités pour les métriques et la journalisation des métriques.

Vous pouvez utiliser les opérations `getKinesisVideoMetrics` et `getKinesisVideoStreamMetrics` API pour récupérer des informations sur Kinesis Video Streams et vos flux actifs.

Le code suivant provient du fichier `kinesis-video-pic/src/client/include/com/amazonaws/kinesis/video/client/Include.h`.

```
/**
 * Gets information about the storage availability.
 *
 * @param 1 CLIENT_HANDLE - the client object handle.
 * @param 2 PKinesisVideoMetrics - OUT - Kinesis Video metrics to be filled.
 *
 * @return Status of the function call.
 */
PUBLIC_API STATUS getKinesisVideoMetrics(CLIENT_HANDLE, PKinesisVideoMetrics);

/**
 * Gets information about the stream content view.
 *
 * @param 1 STREAM_HANDLE - the stream object handle.
 * @param 2 PStreamMetrics - Stream metrics to fill.
 *
 * @return Status of the function call.
 */
PUBLIC_API STATUS getKinesisVideoStreamMetrics(STREAM_HANDLE, PStreamMetrics);
```

L'objet `PClientMetrics` rempli par `getKinesisVideoMetrics` contient les informations suivantes :

- contenu `StoreSize` : taille globale en octets du magasin de contenu (mémoire utilisée pour stocker les données de streaming).
- contenu `StoreAvailableSize` : mémoire disponible dans le magasin de contenu, en octets.

- `total StoreAllocatedSize` : mémoire allouée dans le magasin de contenu.
- `total ContentViewsSize` : mémoire totale utilisée pour l'affichage du contenu. L'affichage du contenu est une série d'indices d'informations dans le magasin de contenu.
- `total FrameRate` : nombre total d'images par seconde pour tous les flux actifs.
- `total TransferRate` : nombre total de bits par seconde (bps) envoyés dans tous les flux.

L'objet `PStreamMetrics` rempli par `getKinesisVideoStreamMetrics` contient les informations suivantes :

- `current ViewDuration` : différence en unités de 100 ns entre la tête de la vue du contenu (lorsque les images sont codées) et la position actuelle (lorsque les données d'image sont envoyées à Kinesis Video Streams).
- `global ViewDuration` : différence en unités de 100 ns entre la tête de la vue du contenu (lorsque les images sont codées) et la fin (lorsque les images sont vidées de la mémoire, soit parce que l'espace total alloué à l'affichage du contenu est dépassé, soit parce qu'un `PersistedAck` message est reçu de Kinesis Video Streams et que les images connues pour être persistantes sont vidées).
- `current ViewSize` : taille en octets du contenu affiché entre la tête (lorsque les images sont codées) et la position actuelle (lorsque les images sont envoyées à Kinesis Video Streams).
- `global ViewSize` : taille totale en octets de l'affichage du contenu.
- `current FrameRate` : dernière fréquence mesurée du flux, en images par seconde.
- `current TransferRate` : dernier débit mesuré du flux, en octets par seconde.

Démontage

Si vous souhaitez envoyer les octets restants dans un tampon et attendre le ACK, vous pouvez utiliser `stopSync` :

```
kinesis_video_stream->stopSync();
```

Vous pouvez également appeler `stop` pour mettre fin au streaming :

```
kinesis_video_stream->stop();
```

Après avoir arrêté le flux, vous pouvez le libérer en appelant l'API suivante :

```
kinesis_video_producer_ ->freeStream(kinesis_video_stream);
```

Exécutez et vérifiez le code

Pour exécuter et vérifier le code du [the section called “C++”](#), consultez les OS-specific instructions suivantes :

- [Linux](#)
- [macOS](#)
- [Windows](#)
- [OS Raspberry Pi](#)

Vous pouvez surveiller le trafic sur votre stream en observant les métriques associées à votre stream dans la CloudWatch console Amazon, telles que `PutMedia.IncomingBytes`.

Utiliser le SDK C++ Producer en tant que plugin GStreamer

[GStreamer](#) est un framework multimédia populaire utilisé par plusieurs caméras et sources vidéo pour créer des pipelines multimédias personnalisés en combinant des plugins modulaires. Le plugin Kinesis Video Streamer rationalise l'intégration de votre pipeline multimédia GStreamer existant à Kinesis Video Streams.

Pour plus d'informations sur l'utilisation du SDK C++ Producer en tant que plugin GStreamer, consultez. [Exemple : Plug-in GStreamer du SDK Kinesis Video Streams Producer - kvssink](#)

Utiliser le SDK C++ Producer en tant que plugin GStreamer dans un conteneur Docker

[GStreamer](#) est un framework multimédia populaire utilisé par plusieurs caméras et sources vidéo pour créer des pipelines multimédias personnalisés en combinant des plugins modulaires. Le plugin Kinesis Video Streamer rationalise l'intégration de votre pipeline multimédia GStreamer existant à Kinesis Video Streams.

En outre, l'utilisation de [Docker](#) pour créer le pipeline GStreamer normalise l'environnement d'exploitation de Kinesis Video Streams, ce qui rationalise la création et l'exécution de l'application.

Pour plus d'informations sur l'utilisation du SDK C++ Producer en tant que plugin GStreamer dans un conteneur Docker, consultez. [Exécutez l'élément GStreamer dans un conteneur Docker](#)

Utiliser la journalisation avec le SDK C++ Producer

Vous configurez la journalisation pour les applications du SDK C++ Producer dans le `kvs_log_configuration` fichier du `kinesis-video-native-build` dossier.

L'exemple suivant montre la première ligne du fichier de configuration par défaut, qui configure l'application pour écrire les entrées de journal de niveau DEBUG dans la AWS Management Console:

```
log4cpplus.rootLogger=DEBUG, KvsConsoleAppender
```

Vous pouvez définir le niveau de journalisation sur INFO pour une journalisation avec moins de commentaires.

Pour configurer l'application afin qu'elle écrive des entrées de journal dans un fichier journal, mettez à jour la première ligne du fichier comme suit :

```
log4cpplus.rootLogger=DEBUG, KvsConsoleAppender, KvsFileAppender
```

Cela configure l'application pour écrire les entrées de journal dans `kvs.log`, dans le dossier `kinesis-video-native-build/log`.

Pour modifier l'emplacement du fichier journal, mettez à jour la ligne suivante avec le nouveau chemin d'accès :

```
log4cpplus.appender.KvsFileAppender.File=../log/kvs.log
```

Note

Si la journalisation de niveau DEBUG s'effectue dans un fichier, le fichier journal peut rapidement utiliser tout l'espace de stockage disponible sur l'appareil.

Utiliser la bibliothèque C Producer

Vous pouvez utiliser la bibliothèque C Producer fournie par Amazon Kinesis Video Streams pour écrire le code d'application permettant d'envoyer des données multimédia d'un appareil vers un flux vidéo Kinesis.

Modèle d'objet

La bibliothèque de production Kinesis Video Streams C est basée sur un composant commun appelé Platform Independent Codebase (PIC), qui est disponible sur GitHub . <https://github.com/aws-labs/amazon-kinesis-video-streams-pic/> Le PIC contient une logique métier indépendante de la plate-forme pour les composants de base. La bibliothèque de production Kinesis Video Streams C intègre à PIC une couche supplémentaire d'API qui permet des rappels et des événements spécifiques au scénario et à la plate-forme. La bibliothèque de production Kinesis Video Streams C comprend les composants suivants basés sur PIC :

- Fournisseurs d'informations sur les appareils : expose la `DeviceInfo` structure qui peut être directement fournie à l'API PIC. Vous pouvez configurer un ensemble de fournisseurs, y compris un fournisseur optimisé pour les scénarios d'application qui peut optimiser le magasin de contenu en fonction du nombre et des types de flux gérés par votre application et de la quantité de mémoire tampon requise configurée en fonction de la quantité de RAM disponible.
- Fournisseur d'informations de flux — Expose la `StreamInfo` structure qui peut être directement fournie à l'API PIC. Il existe un ensemble de fournisseurs spécifiques aux types d'applications et aux types courants de scénarios de streaming. Il s'agit notamment de fournisseurs tels que la vidéo, l'audio et le multipiste audio et vidéo. Chacun de ces scénarios comporte des paramètres par défaut que vous pouvez personnaliser en fonction des exigences de votre application.
- Fournisseur de rappel : expose la `ClientCallbacks` structure qui peut être directement fournie à l'API PIC. Cela inclut un ensemble de fournisseurs de rappels pour la mise en réseau (rappels d'API-based CURL), l'autorisation (API d'AWS identification) et le rappel des tentatives de diffusion en cas d'erreur. L'API Callback Provider utilise un certain nombre d'arguments pour être configurée, tels que les informations d'autorisation Région AWS et. Cela se fait en utilisant des certificats IoT ou en utilisant AWS AccessKeyId SecretKey, ou SessionToken. Vous pouvez améliorer le fournisseur de rappel avec des rappels personnalisés si votre application a besoin du traitement ultérieur d'un rappel particulier pour répondre à une logique spécifique à une application.
- `FrameOrderCoordinator`— Permet de gérer la synchronisation audio et vidéo pour les scénarios multipistes. Il possède un comportement par défaut, que vous pouvez personnaliser pour gérer la logique spécifique de votre application. Il rationalise également l'empaquetage des métadonnées de trame dans la structure PIC Frame avant de le soumettre à l'API PIC de couche inférieure. Pour les scénarios non multipistes, ce composant est une passerelle vers l'API `putFrame` PIC.

La bibliothèque C fournit les objets suivants pour gérer l'envoi de données vers un flux vidéo Kinesis :

- `KinesisVideoClient`— Contient des informations sur votre appareil et gère des rappels pour signaler les événements liés à Kinesis Video Streams.
- `KinesisVideoStream`— Représente des informations sur les paramètres du flux vidéo, tels que le nom, la durée de conservation des données et le type de contenu multimédia.

Insérez du contenu multimédia dans le stream

Vous pouvez utiliser les méthodes fournies par la bibliothèque C (par exemple, `PutKinesisVideoFrame`) pour insérer des données dans l'`KinesisVideoStream` objet. La bibliothèque gère ensuite l'état interne des données, ce qui peut inclure les tâches suivantes :

- Réalisation de l'authentification.
- Supervision de la latence du réseau. Si la latence est trop élevée, la bibliothèque peut choisir d'abandonner des images.
- Suivi de l'état du streaming en cours.

Procédure : utiliser le SDK du producteur C

Cette procédure explique comment utiliser le client Kinesis Video Streams et les sources multimédia dans une application C pour H.264-encodé envoyer des images vidéo vers votre flux vidéo Kinesis.

La procédure comprend les étapes suivantes :

- [Téléchargez le code de la bibliothèque C Producer](#)
- [Rédiger et examiner le code](#)
- [Exécutez et vérifiez le code](#)

Conditions préalables

Avant de configurer le SDK C Producer, assurez-vous de remplir les conditions préalables suivantes :

- Informations d'identification — Dans l'exemple de code, vous fournissez des informations d'identification en spécifiant un profil que vous avez configuré dans votre fichier de profil AWS d'informations d'identification. Si vous n'y avez pas encore procédé, commencez par définir votre profil à informations d'identification.

Pour plus d'informations, voir [Configurer les AWS informations d'identification et la région pour le développement](#).

- Intégration du magasin de certificats — La bibliothèque du producteur Kinesis Video Streams doit établir un lien de confiance avec le service qu'elle appelle. Cela se fait en validant les autorités de certification (CA) dans le magasin de certificats public. Sur Linux-based les modèles, ce magasin se trouve dans le répertoire `/etc/ssl/`.

Télécharger le certificat depuis l'emplacement suivant vers votre magasin de certificats :

<https://www.amazontrust.com/repository/SFSRootCAG2.pem>

- Installez les dépendances de build suivantes pour macOS :
 - [Autoconf 2.69](#) (License GPLv3+/ Autoconf : GNU GPL version 3 ou ultérieure)
 - [CMake 3.7 ou 3.8](#)
 - [Pkg-Config](#)
 - xCode (macOS) / clang / gcc (xcode-select version 2347)
 - Kit de développement Java (JDK) (pour compilation Java JNI)
 - [Lib-Pkg](#)
- Installez les dépendances de build suivantes pour Ubuntu :
 - Git : `sudo apt install git`
 - [CMake](#) : `sudo apt install cmake`
 - G++ : `sudo apt install g++`
 - configuration du fichier : `sudo apt install pkg-config`
 - OpenJDK : `sudo apt install openjdk-8-jdk`
 - Définissez la variable d'environnement : `JAVA_HOME export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

Téléchargez le code de la bibliothèque C Producer

Dans cette section, vous téléchargez les bibliothèques de bas niveau. Pour les conditions préalables et autres informations concernant cet exemple, consultez [the section called "C++"](#).

1. Créez un répertoire, puis clonez l'exemple de code source à partir du GitHub référentiel.

```
git clone --recursive https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c.git
```

Note

Si vous n'exécutez pas la commande `git clone` avec `--recursive`, exécutez `git submodule update --init` dans le répertoire `amazon-kinesis-video-streams-producer-c/open-source`. Vous devez également installer `pkg-config`, `CMake` et un environnement de construction.

Pour obtenir plus d'informations, consultez `README.md` dans <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c.git>.

- Ouvrez le code dans l'environnement de développement intégré (IDE) de votre choix (par exemple, [Eclipse](#)).

Rédiger et examiner le code

Dans cette section, vous allez examiner le code de l'exemple d'application `KvsVideoOnlyStreamingSample.c` dans le dossier `samples` du <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c> repo on GitHub. Vous avez téléchargé ce code lors de l'étape précédente. Cet exemple montre comment utiliser la bibliothèque C Producer pour envoyer des images H.264-encodées vidéo contenues dans le dossier `samples/h264SampleFrames` vers votre flux vidéo Kinesis.

Cet exemple d'application contient trois sections :

- Initialisation et configuration :
 - Initialisation et configuration du pipeline de médias spécifique à la plateforme.
 - Initialisation et configuration `KinesisVideoClient` et `KinesisVideoStream` pour le pipeline, définition des rappels, intégration de l'authentification spécifique au scénario, extraction et soumission des données privées du codec et mise du flux à l'état `READY`.
- Boucle principale :
 - Obtention de l'image de la part du pipeline multimédias avec les horodatages et les balises.
 - Soumission du cadre au `KinesisVideoStream`.
- Démontage :

- Arrêt (synchronisation) KinesisVideoStream, libération KinesisVideoStream, libération KinesisVideoClient.

Cet exemple d'application exécute les tâches suivantes :

- Appelez l'API `createDefaultDeviceInfo` pour créer l'objet `deviceInfo` qui contient des informations sur le périphérique ou la configuration du stockage.

```
// default storage size is 128MB. Use setDeviceInfoStorageSize after create to change
// storage size.
CHK_STATUS(createDefaultDeviceInfo(&pDeviceInfo));
// adjust members of pDeviceInfo here if needed
pDeviceInfo->clientInfo.loggerLogLevel = LOG_LEVEL_DEBUG;
```

- Appelez l'API `createRealtimeVideoStreamInfoProvider` pour créer l'objet `StreamInfo`.

```
CHK_STATUS(createRealtimeVideoStreamInfoProvider(streamName,
    DEFAULT_RETENTION_PERIOD, DEFAULT_BUFFER_DURATION, &pStreamInfo));
// adjust members of pStreamInfo here if needed
```

- Appelez l'`createDefaultCallbacksProviderWithAwsCredentialsAPI` pour créer le fournisseur de rappels par défaut basé sur des AWS informations d'identification statiques.

```
CHK_STATUS(createDefaultCallbacksProviderWithAwsCredentials(accessKey,
    secretKey,
    sessionToken,
    MAX_UINT64,
    region,
    cacertPath,
    NULL,
    NULL,
    FALSE,
    &pClientCallbacks));
```

- Appelez l'`createKinesisVideoClientAPI` pour créer l'`KinesisVideoClient`objet qui contient des informations sur le stockage de votre appareil et gère les rappels pour signaler les événements Kinesis Video Streams.

```
CHK_STATUS(createKinesisVideoClient(pDeviceInfo, pClientCallbacks, &clientHandle));
```

- Appelez l'API `createKinesisVideoStreamSync` pour créer l'objet `KinesisVideoStream`.

```
CHK_STATUS(createKinesisVideoStreamSync(clientHandle, pStreamInfo, &streamHandle));
```

- Configurez un exemple d'image et appelez l'API `PutKinesisVideoFrame` pour envoyer cette image à l'objet `KinesisVideoStream`.

```
// setup sample frame
MEMSET(frameBuffer, 0x00, frameSize);
frame.frameData = frameBuffer;
frame.version = FRAME_CURRENT_VERSION;
frame.trackId = DEFAULT_VIDEO_TRACK_ID;
frame.duration = HUNDREDS_OF_NANOS_IN_A_SECOND / DEFAULT_FPS_VALUE;
frame.decodingTs = defaultGetTime(); // current time
frame.presentationTs = frame.decodingTs;

while(defaultGetTime() > streamStopTime) {
    frame.index = frameIndex;
    frame.flags = fileIndex % DEFAULT_KEY_FRAME_INTERVAL == 0 ?
FRAME_FLAG_KEY_FRAME : FRAME_FLAG_NONE;
    frame.size = sizeof(frameBuffer);

    CHK_STATUS(readFrameData(&frame, frameFilePath));

    CHK_STATUS(putKinesisVideoFrame(streamHandle, &frame));
    defaultThreadSleep(frame.duration);

    frame.decodingTs += frame.duration;
    frame.presentationTs = frame.decodingTs;
    frameIndex++;
    fileIndex++;
    fileIndex = fileIndex % NUMBER_OF_FRAME_FILES;
}
```

- Démontage :

```
CHK_STATUS(stopKinesisVideoStreamSync(streamHandle));
CHK_STATUS(freeKinesisVideoStream(&streamHandle));
CHK_STATUS(freeKinesisVideoClient(&clientHandle));
```

Exécutez et vérifiez le code

Pour exécuter et vérifier le code [du the section called "C++"](#), procédez comme suit :

1. Exécutez les commandes suivantes pour créer un build répertoire dans le [SDK C que vous avez téléchargé](#), puis cmake lancez-le :

```
mkdir -p amazon-kinesis-video-streams-producer-c/build;
cd amazon-kinesis-video-streams-producer-c/build;
cmake ..
```

Vous pouvez transmettre les options suivantes à cmake ..

- -DBUILD_DEPENDENCIES- s'il faut construire des bibliothèques dépendantes à partir des sources.
- -DBUILD_TEST=TRUE- construire des tests unitaires et d'intégration. Cela peut être utile pour confirmer que votre appareil est compatible.

```
./tst/webrtc_client_test
```

- -DCODE_COVERAGE- activer les rapports de couverture.
- -DCOMPILER_WARNINGS- active tous les avertissements du compilateur.
- -DADDRESS_SANITIZER- construire avec AddressSanitizer.
- -DMEMORY_SANITIZER- construire avec MemorySanitizer.
- -DTHREAD_SANITIZER- construire avec ThreadSanitizer.
- -DUNDEFINED_BEHAVIOR_SANITIZER- construire avec UndefinedBehaviorSanitizer.
- -DALIGNED_MEMORY_MODEL - créer pour les appareils avec modèle de mémoire alignés uniquement. La valeur par défaut est OFF.

2. Accédez au build répertoire que vous venez de créer à l'étape précédente et lancez-le make pour créer le SDK WebRTC C et les exemples fournis.

```
make
```

3. L'exemple d'application `kinesis_video_cproducer_video_only_sample` envoie à Kinesis Video Streams des images vidéo codées en H.264 contenues dans `samples/h264SampleFrames` le dossier. La commande suivante envoie les images vidéo en boucle pendant dix secondes à Kinesis Video Streams :

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10
```

Si vous souhaitez envoyer H.264-encoded des images depuis un autre dossier (par exemple, MyH264FramesFolder), exécutez l'exemple avec les arguments suivants :

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10 MyH264FramesFolder
```

4. Pour activer les journaux détaillés, définissez le HEAP_DEBUG et LOG_STREAMING C-defines en décommentant les lignes appropriées. CMakeList.txt

Vous pouvez surveiller la progression de la gamme de test dans la sortie de débogage dans votre IDE. Vous pouvez également surveiller le trafic sur votre stream en regardant les métriques associées à votre stream dans la CloudWatch console Amazon, telles que PutMedia.IncomingBytes.

Note

La console n'affiche pas les données sous forme de flux vidéo, dans la mesure où le harnais de test envoie uniquement des images vides en octets

Utiliser le SDK du producteur C++ sur Raspberry Pi

Le Raspberry Pi est un petit ordinateur peu onéreux qui peut être utilisé pour enseigner et apprendre des compétences élémentaires de programmation d'ordinateur. Ce didacticiel explique comment configurer et utiliser le SDK de production Amazon Kinesis Video Streams C++ sur un appareil Raspberry Pi. Cette procédure inclut également la vérification de l'installation à l'aide de la démonstration GStreamer.

Rubriques

- [Conditions préalables](#)
- [Création d'un utilisateur IAM autorisé à écrire sur Kinesis Video Streams](#)
- [Connectez votre Raspberry Pi à votre Wi-Fi réseau](#)
- [Connectez-vous à distance à votre Raspberry Pi](#)
- [Configuration de la caméra Raspberry Pi](#)

- [Installation des prérequis logiciels](#)
- [Téléchargez et créez le SDK Kinesis Video Streams C++ pour le producteur](#)
- [Diffusez de la vidéo sur votre flux vidéo Kinesis](#)
- [Lire le contenu multimédia de votre flux vidéo Kinesis](#)
- [Résolution des problèmes de compilation sur le SDK C++ Producer pour Raspberry Pi](#)

Conditions préalables

Avant de configurer le SDK du producteur C++ sur votre Raspberry Pi, assurez-vous de remplir les conditions préalables suivantes :

- Un appareil Raspberry Pi doté de la configuration suivante :
 - Version de carte : Modèle 3 B ou version ultérieure.
 - Un [module de caméra](#) connecté ou une caméra USB connectée (webcam).
 - Une carte SD d'une capacité d'au moins 8 Go.
 - Le système d'exploitation Raspbian (version de noyau 4.9 ou ultérieure) installé. Vous pouvez télécharger la dernière image du système d'exploitation Raspberry Pi (précédemment appelé Raspbian) sur le [site Web du Raspberry Pi](#). Suivez les instructions de Raspberry Pi pour [installer l'image téléchargée sur une carte SD](#).
- Et Compte AWS avec un flux vidéo Kinesis. Pour plus d'informations, consultez [Démarez avec Flux vidéo Kinesis](#).

Création d'un utilisateur IAM autorisé à écrire sur Kinesis Video Streams

Si ce n'est pas déjà fait, configurez un utilisateur Gestion des identités et des accès AWS (IAM) autorisé à écrire sur un flux vidéo Kinesis.

Ces procédures ont pour but de vous aider à commencer rapidement à utiliser une paire de clés d' AWS accès. Les appareils peuvent utiliser X.509 des certificats pour se connecter à AWS IoT. Consultez [the section called “Contrôle de l'accès aux ressources Kinesis Video Streams à l'aide de AWS IoT”](#) pour plus d'informations sur la configuration de votre appareil pour utiliser l'authentification basée sur des certificats.

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.

2. Dans le menu de navigation de gauche, choisissez Utilisateurs.
3. Pour créer un nouvel utilisateur, choisissez Add user (Ajouter un utilisateur).
4. Choisissez un nom d'utilisateur descriptif pour l'utilisateur, par exemple **kinesis-video-raspberry-pi-producer**.
5. Sous Access type (Type d'accès), choisissez Programmatic access (Accès par programme).
6. Choisissez Suivant : Autorisations.
7. Sous Set permissions for kinesis-video-raspberry-pi-producer (Définir les autorisations pour kinesis-video-raspberry-pi-producer), choisissez Attach existing policies directly (Attacher directement les stratégies existantes).
8. Choisissez Create Policy (Créer une politique). La page Create policy (Créer la stratégie) s'ouvre dans un nouvel onglet du navigateur web.
9. Choisissez l'onglet JSON.
10. Copiez la stratégie JSON suivante et collez-la dans la zone de texte. Cette politique autorise votre utilisateur à créer et à écrire des données dans les flux vidéo Kinesis.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "kinesisvideo:DescribeStream",
      "kinesisvideo:CreateStream",
      "kinesisvideo:GetDataEndpoint",
      "kinesisvideo:PutMedia"
    ],
    "Resource": [
      "*"
    ]
  }]
}
```

11. Choisissez Examiner une politique.
12. Attribuez un nom à votre stratégie, par exemple **kinesis-video-stream-write-policy**.
13. Choisissez Create Policy (Créer une politique).

14. Revenez à l'onglet Add user (Ajouter un utilisateur) dans votre navigateur, puis choisissez Refresh (Actualiser).
15. Dans la zone de recherche, saisissez le nom de la stratégie que vous avez créée.
16. Cochez la case en regard de la nouvelle stratégie dans la liste.
17. Choisissez Next: Review (Suivant : Vérification).
18. Choisissez Create user.
19. La console affiche l'ID de clé d'accès pour votre nouvel utilisateur. Choisissez Afficher pour afficher la clé d'accès secrète. Enregistrez ces valeurs ; elles sont nécessaires lorsque vous configurez l'application.

Connectez votre Raspberry Pi à votre Wi-Fi réseau

Si vous utilisez un clavier et un moniteur connectés, passez à [Configuration de la caméra Raspberry Pi](#).

Ces instructions sont écrites pour vous aider à configurer votre Raspberry Pi lorsqu'il est exécuté en mode headless, c'est-à-dire sans clavier, moniteur ou câble réseau connecté. Suivez les instructions ci-dessous pour configurer votre Raspberry Pi afin qu'il tente automatiquement de se connecter au réseau spécifié, permettant ainsi à votre machine hôte de s'y connecter par SSH.

1. Sur votre ordinateur, créez un dossier nommé `wpa_supplicant.conf`.
2. Copiez le texte suivant et collez-le dans le `wpa_supplicant.conf` fichier :

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
  ssid="Your Wi-Fi SSID"
  scan_ssid=1
  key_mgmt=WPA-PSK
  psk="Your Wi-Fi Password"
}
```

Remplacez les psk valeurs ssid et par les informations relatives à votre Wi-Fi réseau.

3. Copiez le fichier `wpa_supplicant.conf` dans la carte SD. Il doit être copié à la racine du volume boot.

4. Insérez la carte SD dans l'appareil Raspberry Pi et allumez ce dernier. Il rejoint votre Wi-Fi réseau et SSH est activé.

Connectez-vous à distance à votre Raspberry Pi

Si vous utilisez un clavier et un moniteur connectés, passez à [Configuration de la caméra Raspberry Pi](#).

Ces instructions sont écrites pour vous aider à configurer votre Raspberry Pi lorsqu'il est exécuté en mode headless, c'est-à-dire sans clavier, moniteur ou câble réseau connecté. Suivez les instructions ci-dessous pour localiser votre Raspberry Pi sur votre réseau et y accéder par SSH depuis votre machine hôte.

1. Avant de vous connecter à distance à votre appareil Raspberry Pi, effectuez l'une des actions suivantes pour déterminer son adresse IP :
 - Si vous avez accès au Wi-Fi routeur de votre réseau, examinez les Wi-Fi appareils connectés. Recherchez l'appareil nommé `Raspberry Pi` pour trouver l'adresse IP de votre appareil.
 - Si vous n'avez pas accès au Wi-Fi routeur de votre réseau, vous pouvez utiliser un autre logiciel pour rechercher des appareils sur votre réseau. [Fing](#) est une application populaire disponible à la fois pour les appareils Android et iOS. Vous pouvez utiliser la version gratuite de cette application pour trouver les adresses IP des appareils figurant sur votre réseau.
2. Lorsque vous connaissez l'adresse IP de l'appareil Raspberry Pi, vous pouvez utiliser n'importe quelle application de terminal pour vous connecter.
 - Sur macOS ou Linux, utilisez `ssh` :

```
ssh pi@<IP address>
```

- Sur Windows, utilisez [PuTTY](#), un client SSH gratuit pour Windows.

Pour une nouvelle installation de Raspbian, le nom d'utilisateur est **pi** et le mot de passe est **raspberry**. Nous vous recommandons de [changer le mot de passe par défaut](#).

Configuration de la caméra Raspberry Pi

Procédez comme suit pour configurer le [module de caméra Raspberry Pi](#) afin qu'il envoie de la vidéo depuis l'appareil vers un flux vidéo Kinesis.

Note

Si vous utilisez une webcam USB, passez directement à [the section called "Installation des prérequis logiciels"](#).

Camera module 1

Suivez ces instructions pour mettre à jour le fichier des modules, activer l'interface de la caméra et vérifier le fonctionnement de votre caméra. La mise à jour du fichier de modules indique au Raspberry Pi quels modules du noyau charger au moment du démarrage. Le pilote de caméra n'est pas chargé par défaut afin de préserver les ressources du système sur les appareils Raspberry Pi qui n'utilisent pas de caméra.

1. Ouvrez un éditeur pour apporter des modifications au fichier des modules. Ouvrez le terminal et utilisez la commande suivante pour modifier le fichier à l'aide de l'éditeur nano :

```
sudo nano /etc/modules
```

2. Ajoutez la ligne suivante à la fin du fichier, si ce n'est pas déjà fait :

```
bcm2835-v4l2
```

3. Enregistrez le fichier et quittez l'éditeur. Pour enregistrer et quitter à l'aide de l'éditeur nano, utilisez Ctrl+X.

4. Redémarrez l'appareil Raspberry Pi :

```
sudo reboot
```

5. Lorsque l'appareil redémarre, connectez-vous y à nouveau par le biais de votre application de terminal si vous vous connectez à distance.

6. Ouvrir `raspi-config`:

```
sudo raspi-config
```

7. Choisissez Options d'interfaçage, Legacy Camera. Dans les anciennes versions du système d'exploitation Raspbian, cette option de menu peut se trouver sous Options d'interface, Caméra.

Activez la caméra si elle n'est pas déjà activée, et redémarrez si vous y êtes invité.

8. Vérifiez que la caméra fonctionne en tapant la commande suivante :

```
raspistill -v -o test.jpg
```

Si votre appareil photo est correctement configuré, cette commande capture une image de l'appareil photo, l'enregistre dans un fichier nommé `test.jpg` et affiche des messages d'information.

Camera module 2 or 3

Si vous utilisez un module de caméra 2, vous utilisez soit `bcm2835-v4l2` (ancien) soit `libcamera` (moderne). Cependant, la `libcamera` pile est recommandée pour un meilleur support et de meilleures fonctionnalités. Suivez les étapes ci-dessous pour vérifier que votre système `libcamera` est à jour.

1. [libcamera](#) devrait être préinstallé sur votre Raspberry Pi. Vérifiez les mises à jour et mettez à jour la dernière version pour les corrections de bogues et les mises à jour de sécurité. Ouvrez le terminal et saisissez les commandes suivantes :

```
sudo apt-get update  
sudo apt-get upgrade
```

2. Redémarrez votre système pour que les mises à jour prennent effet.

```
sudo reboot
```

3. Testez votre appareil photo. Cette application lance un flux d'aperçu de la caméra et l'affiche à l'écran.

```
libcamera-hello
```

Si vous rencontrez des problèmes avec votre module de caméra, consultez la [documentation du Raspberry Pi](#) pour le dépannage.

Installation des prérequis logiciels

Le SDK C++ Producer nécessite que vous installiez les prérequis logiciels suivants sur le Raspberry Pi.

1. Mettez à jour la liste des packages et installez les bibliothèques nécessaires à la création du SDK. Ouvrez le terminal et saisissez les commandes suivantes :

```
sudo apt-get update
sudo apt-get install -y \
  automake \
  build-essential \
  cmake \
  git \
  gstreamer1.0-plugins-base-apps \
  gstreamer1.0-plugins-bad \
  gstreamer1.0-plugins-good \
  gstreamer1.0-plugins-ugly \
  gstreamer1.0-tools \
  gstreamer1.0-omx-generic \
  libcurl4-openssl-dev \
  libgstreamer1.0-dev \
  libgstreamer-plugins-base1.0-dev \
  liblog4cplus-dev \
  libssl-dev \
  pkg-config
```

2. Si vous utilisez la `libcamera` pile, installez également le plugin `libcamerasrc` GStreamer. Ce plugin GStreamer n'est pas installé par défaut.

```
sudo apt-get install gstreamer1.0-libcamera
```

3. Copiez le fichier PEM suivant dans `/etc/ssl/cert.pem`:

```
sudo curl https://www.amazontrust.com/repository/AmazonRootCA1.pem -o /etc/ssl/
AmazonRootCA1.pem
sudo chmod 644 /etc/ssl/AmazonRootCA1.pem
```

Téléchargez et créez le SDK Kinesis Video Streams C++ pour le producteur

Suivez les procédures ci-dessous pour télécharger et créer le SDK [Kinesis Video Streams C++ + Producer](#). Assurez-vous d'avoir installé les prérequis logiciels ; consultez [the section called "Installation des prérequis logiciels"](#) ces étapes.

1. Accédez au répertoire de téléchargement. Ouvrez un terminal et accédez à votre répertoire de téléchargement préféré.

Par exemple :

```
cd ~/Downloads
```

2. Clonez le référentiel du SDK. Utilisez la `git clone` commande pour télécharger le SDK depuis le GitHub référentiel. Type :

```
git clone https://github.com/awslabs/amazon-kinesis-video-streams-producer-sdk-cpp.git --single-branch -b master kvs-producer-sdk-cpp
```

Cette commande ne clone qu'une seule branche (la `master` branche), réduisant ainsi la taille et la durée du téléchargement. Il place également le contenu téléchargé dans un dossier appelé `kvs-producer-sdk-cpp` dans le répertoire actuel.

3. Vérifiez le téléchargement. Une fois le processus de clonage terminé, listez le contenu du `kvs-producer-sdk-cpp` dossier pour vérifier que le SDK a été téléchargé.

```
ls kvs-producer-sdk-cpp
```

4. Préparez un répertoire de compilation. Type :


```
mkdir -p kvs-producer-sdk-cpp/build  
cd kvs-producer-sdk-cpp/build
```

5. Configurez le build. Exécutez la `cmake` commande suivante pour configurer l'environnement de génération avec des options spécifiques :

```
cmake .. -DBUILD_GSTREAMER_PLUGIN=ON -DBUILD_DEPENDENCIES=OFF -  
DALIGNED_MEMORY_MODEL=ON
```

[CMake](#) utilise les options suivantes pour générer les informations appropriées `Makefiles` :

- Utiliser le dossier du projet (. .) comme répertoire source.
- Utilisation du répertoire actuel (.) (build /) pour la sortie de compilation.
- -DBUILD_GSTREAMER_PLUGIN=0N permet la création du plugin GStreamer kvssink.
- -DBUILD_DEPENDENCIES=0FF désactive la création de dépendances externes à partir de la source. Le projet trouvera et utilisera les dépendances externes installées lors d'une étape précédente.
- -DALIGNED_MEMORY_MODEL=0N désactive le modèle de mémoire non aligné. L'accès à la mémoire non aligné n'est pas pris en charge par certains appareils Raspberry Pi.

 Note

Pour une liste complète des arguments CMake, voir [the section called “Téléchargez et configurez le code”](#).

6. Générez le projet. Après avoir configuré le build, utilisez la make commande pour compiler à l'aide de la commande Makefile générée par cmake.

```
make -j$(nproc)
```

L'- j argument to make lui permet d'exécuter plusieurs tâches de compilation en parallèle. Pour réduire les temps de construction, utilisez la nproc commande pour calculer dynamiquement le nombre de cœurs de processeur de votre Raspberry Pi.

7. Confirmez qu'libgstkvssink.so est présent.

Répertoriez les fichiers du répertoire en cours.

Prompt :

```
ls
```

Réponse :

```
CMakeCache.txt      dependency          kvs_gstreamer_sample
CMakeFiles          kvs_gstreamer_audio_video_sample  kvssink_gstreamer_sample
Makefile            kvs_gstreamer_file_uploader_sample libKinesisVideoProducer.so
```

```
cmake_install.cmake kvs_gstreamer_multistream_sample libgstkvssink.so
```

8. Vérifiez que GStreamer peut se charger. kvssink

Définissez la variable d'GST_PLUGIN_PATH environnement sur le répertoire contenant `libgstkvssink.so`.

```
export GST_PLUGIN_PATH=`pwd`
```

Faites charger GStreamer : kvssink

```
gst-inspect-1.0 kvssink
```

Vous devriez voir de la documentation sur `kvssink`. Utilisez les touches fléchées pour naviguer et appuyez `q` pour quitter.

9. (Facultatif) Mettez à jour le script de démarrage de votre shell pour inclure la définition de la variable d'GST_PLUGIN_PATH environnement. Cela garantit GST_PLUGIN_PATH un réglage correct lors d'une nouvelle session de terminal. Sur les appareils Raspberry Pi, le script de démarrage du shell est `~/ .bashrc`.

Exécutez la commande suivante pour ajouter la commande à la fin du script de démarrage du shell.

```
echo "export GST_PLUGIN_PATH=~/Downloads/kvs-producer-sdk-cpp/build" >> ~/.bashrc
```

Tapez ce qui suit pour exécuter le script de démarrage du shell ou fermer le shell actuel et en ouvrir un nouveau.

```
source ~/.bashrc
```

Confirmez que le GST_PLUGIN_PATH paramètre est réglé et que vous pouvez charger `kvssink`.

```
echo $GST_PLUGIN_PATH
```

```
gst-inspect-1.0 kvssink
```

Diffusez de la vidéo sur votre flux vidéo Kinesis

Pour exécuter cet exemple d'application, vous avez besoin des informations suivantes :

- Le nom du flux que vous avez créé dans la section [Conditions préalables](#).
- Les informations d'identification de compte (ID de clé d'accès et clé d'accès secrète) que vous avez créées dans [Création d'un utilisateur IAM autorisé à écrire sur Kinesis Video Streams](#).
- GStreamer est capable de localiser le kvssink plugin. Pour plus d'informations, consultez [the section called "Téléchargez et compilez le SDK C++ Producer"](#).

1. Définissez les informations d'identification et la région.

```
export AWS_ACCESS_KEY_ID=YourAccessKey
export AWS_SECRET_ACCESS_KEY=YourSecretKey
export AWS_DEFAULT_REGION=us-west-2
```

Pour les autres méthodes d'authentification, consultez [the section called "Fournir des informations d'identification à kvssink"](#).

Note

Le SDK C++ Producer utilise par défaut la région USA Ouest (Oregonus-west-2) (). Pour utiliser la valeur par défaut, Région AWS créez votre flux vidéo Kinesis dans la région USA Ouest (Oregon).

Pour utiliser une autre région pour votre flux vidéo Kinesis, définissez la variable d'environnement suivante sur votre région (par exemple, *us-east-1*) :

```
export AWS_DEFAULT_REGION=us-east-1
```

2. En fonction de votre support d'entrée, choisissez l'une des options suivantes :

Sample GStreamer video

Ce pipeline GStreamer génère un flux vidéo de test en direct avec un modèle de test standard qui s'exécute à 10 images par seconde avec une résolution de 640 x 480 pixels. Une superposition est ajoutée pour afficher l'heure et la date actuelles du système. La vidéo est ensuite encodée au H.264 format et des images-clés sont générées au maximum toutes

les 10 images, ce qui donne une durée de fragment (également appelée taille d'un groupe d'images (GoP)) de 1 seconde. kvssink prend le flux H.264-encodé vidéo, le regroupe dans le format conteneur Matroska (MKV) et le télécharge sur votre flux vidéo Kinesis.

Exécutez la commande suivante :

```
gst-launch-1.0 -v videotestsrc is-live=true \  
! video/x-raw,framerate=10/1,width=640,height=480 \  
! clockoverlay time-format="%a %B %d, %Y %I:%M:%S %p" \  
! x264enc bframes=0 key-int-max=10 \  
! h264parse \  
! kvssink stream-name="YourStreamName"
```

Pour arrêter le pipeline GStreamer, sélectionnez la fenêtre du terminal et appuyez sur CTRL +C.

L'exemple de pipeline vidéo GStreamer ressemble à ceci :

USB web cam

Exécutez la commande suivante pour que GStreamer détecte automatiquement votre caméra USB :

```
gst-launch-1.0 autovideosrc \  
! videoconvert \  
! video/x-raw,format=I420,width=640,height=480 \  
! x264enc bframes=0 key-int-max=45 tune=zerolatency byte-stream=true speed-  
preset=ultrafast \  
! h264parse \  
! video/x-h264,stream-format=avc,alignment=au,profile=baseline \  
! kvssink stream-name="YourStreamname"
```

Pour arrêter le pipeline GStreamer, sélectionnez la fenêtre du terminal et appuyez sur CTRL +C.

Plutôt que de laisser GStreamer effectuer la détection automatique, vous pouvez utiliser un identifiant v4l2src d'appareil spécifique. Exécutez la commande suivante :

```
gst-device-monitor-1.0
```

Dans le résultat, vous verrez certains appareils et le début d'un pipeline GStreamer expliquant comment utiliser l'appareil :

Device found:

```

name : H264 USB Camera: USB Camera
class : Video/Source
caps : video/x-h264, stream-format=(string)byte-stream,
alignment=(string)au, width=(int)1920, height=(int)1080, pixel-aspect-
ratio=(fraction)1/1, colorimetry=(string){ 2:4:7:1 }, framerate=(fraction)
{ 30/1, 25/1, 15/1 };
...
properties:
device.path = /dev/video4
udev-probed = false
device.api = v4l2
v4l2.device.driver = uvcvideo
v4l2.device.card = "H264\ USB\ Camera:\ USB\ Camera"
v4l2.device.bus_info = usb-3f980000.usb-1.3
v4l2.device.version = 265767 (0x00040e27)
v4l2.device.capabilities = 2216689665 (0x84200001)
v4l2.device.device_caps = 69206017 (0x04200001)
gst-launch-1.0 v4l2src device=/dev/video4 ! ...

```

Pour arrêter le pipeline GStreamer, sélectionnez la fenêtre du terminal et appuyez sur CTRL +C.

Raspberry Pi camera module 1

Si vous utilisez le module de caméra Pi 1 ou le module de caméra Pi 2 avecbcm2835-v4l2, utilisez ce qui suit :

```

gst-launch-1.0 v4l2src device=/dev/video0 \
! videoconvert \
! video/x-raw,format=I420,width=640,height=480 \
! x264enc bframes=0 key-int-max=45 bitrate=500 tune=zerolatency \
! h264parse ! video/x-h264,stream-format=avc,alignment=au,profile=baseline \
! kvssink stream-name="YourStreamname"

```

Pour arrêter le pipeline GStreamer, sélectionnez la fenêtre du terminal et appuyez sur CTRL +C.

Raspberry Pi camera module 2 or 3

Si vous utilisez la `libcamera` pile moderne, utilisez le pipeline GStreamer suivant :

```
gst-launch-1.0 libcamerasrc \  
  ! video/x-raw,width=640,height=480,framerate=30/1,format=I420 \  
  ! videoconvert \  
  ! x264enc speed-preset=ultrafast tune=zerolatency byte-stream=true key-int-  
max=75 \  
  ! video/x-h264,level='(string)4' \  
  ! h264parse \  
  ! video/x-h264,stream-  
format=avc,alignment=au,width=640,height=480,framerate=30/1 \  
  ! kvssink stream-name="YourStreamname"
```

Pour arrêter le pipeline GStreamer, sélectionnez la fenêtre du terminal et appuyez sur CTRL +C.

Sample RTSP camera

Dans cet exemple, nous hébergeons `Gst-Rtsp-Server` localement un flux de caméra RTSP de démonstration. Nous construisons ensuite un pipeline GStreamer pour télécharger ce flux de caméra RTSP vers le flux vidéo Kinesis spécifié.

À configurer `Gst-Rtsp-Server` sur votre Raspberry Pi

1. Installez les bibliothèques de dépendances nécessaires pour créer le `Gst-Rtsp-Server` projet. Assurez-vous que les prérequis logiciels sont également installés. Tapez ce qui suit dans votre terminal :

```
sudo apt-get update  
sudo apt-get install libgststrtspserver-1.0
```

2. Téléchargez la version 1.22 de GStreamer sur votre Raspberry Pi.

```
git clone https://gitlab.freedesktop.org/gstreamer/gstreamer.git --single-  
branch -b 1.22
```

3. Remplacez les répertoires par le répertoire `examples` dans le serveur `gst-rtsp`.

```
cd gstreamer  
cd subprojects
```

```
cd gst-rtsp-server
cd examples
```

4. Compilez le fichier test-launch.c dans un exécutable appelé test-launch en utilisant gcc.

```
gcc -o test-launch test-launch.c `pkg-config --cflags --libs gstreamer-rtsp-server-1.0`
```

5. Exécutez le fichier exécutable avec les arguments suivants. Remarque : le premier chargement de GStreamer peut prendre un certain temps.

```
./test-launch "videotestsrc is-live=true ! video/x-raw,height=480,width=640,framerate=10/1 ! videoconvert ! x264enc tune=zerolatency bitrate=512 key-int-max=25 bframes=0 ! h264parse ! rtph264pay ! name=pay0 pt=96"
```

Vous devriez voir la sortie suivante :

```
stream ready at rtsp://127.0.0.1:8554/test
```

6. Vérifiez le flux vidéo RTSP. Vous pouvez utiliser n'importe quel visualiseur RTSP. Par exemple, le lecteur multimédia VLC. Pour utiliser le lecteur multimédia VLC pour visionner votre diffusion en direct, ouvrez un nouveau terminal et tapez :

```
sudo apt-get install vlc
```

pour installer le lecteur multimédia VLC. Tapez ensuite :

```
vlc rtsp://127.0.0.1:8554/test
```

Une fenêtre VLC devrait apparaître avec le flux en direct. Si ce n'est pas le cas, vérifiez que le fichier exécutable de lancement du test est toujours en cours d'exécution et vérifiez qu'il n'y a pas d'erreur dans le résultat.

Un autre moyen de vérifier le flux RTSP consiste à utiliser l'utilitaire gst-discoverer-1.0. Tapez :

```
gst-discoverer-1.0 "rtsp://127.0.0.1:8554/test"
```

Le résultat attendu ressemble à ceci :

```
Analyzing rtsp://127.0.0.1:8554/test
Done discovering rtsp://127.0.0.1:8554/test

Properties:
  Duration: 99:99:99.999999999
  Seekable: no
  Live: yes
  unknown #0: application/x-rtp
    video #1: H.264 (Constrained Baseline Profile)
      Stream ID:
      359314d7d4bba383223927d7e57d4244d0800e629c626be81c505055c62170e2/
video:0:0:RTP:AVP:96
  Width: 640
  Height: 480
  Depth: 24
  Frame rate: 10/1
  Pixel aspect ratio: 1/1
  Interlaced: false
  Bitrate: 0
  Max bitrate: 0
```

Pour envoyer le flux RTSP vers votre flux vidéo Kinesis à l'aide de kvssink

Ce pipeline GStreamer est utilisé `rtspsrc` pour se connecter au serveur RTSP afin de récupérer le flux vidéo RTP. Il transmet les images `rtph264depay`, qui extrait les images H.264-encodées vidéo des paquets RTP. `h264parseregroup` les images vidéo dans le format que je `kvssink` peut comprendre. `kvssink` prend le flux H.264-encodé vidéo, le regroupe dans le format conteneur Matroska (MKV) et le télécharge sur votre flux vidéo Kinesis.

Exécutez la commande suivante :

```
gst-launch-1.0 -v rtspsrc location="rtsp://127.0.0.1:8554/test" short-
header=true \
! rtph264depay \
! h264parse \
! video/x-h264,format=avc,alignment=au \
```

```
! kvssink stream-name="YourStreamName"
```

Pour arrêter le pipeline GStreamer, sélectionnez la fenêtre du terminal et appuyez sur CTRL +C.

Utiliser du matériel

Certains modèles de Raspberry Pi sont équipés d'encodeurs accélérés par matériel H.264 . Vous pouvez les utiliser à la place x264enc d'un encodeur logiciel.

1. Assurez-vous que les plugins GStreamer sont installés :

```
sudo apt-get install gstreamer1.0-tools gstreamer1.0-plugins-bad
```

2. Type :

```
gst-inspect-1.0 | grep h264
```

Déterminez si les éléments suivants sont disponibles :

- omxh264enc
- v4l2h264enc

S'ils sont disponibles, vous pouvez les utiliser. Voici quelques exemples de pipelines utilisant ces éléments :

omxh264enc:

```
gst-launch-1.0 v4l2src device=/dev/video0 \  
! videoconvert \  
! video/x-raw,format=I420,width=640,height=480 \  
! omxh264enc control-rate=2 target-bitrate=512000 periodicity-idr=45 inline-  
header=FALSE \  
! h264parse ! video/x-h264,stream-format=avc,alignment=au,profile=baseline \  
! kvssink stream-name="raspberry"
```

v4l2h264encet v4l2convert :

```
gst-launch-1.0 libcamerasrc \  

```

```
! video/x-raw,width=640,height=480,framerate=30/1,format=I420 \  
! v4l2convert \  
! v4l2h264enc extra-controls="controls,repeat_sequence_header=1" \  
! video/x-h264,level=(string)4' \  
! h264parse \  
! video/x-h264,stream-format=avc,alignment=au,width=640,height=480,framerate=30/1 \  
\   
! kvssink stream-name="test-stream"
```

Problèmes d'exécution

Vous trouverez ci-dessous certains problèmes d'exécution fréquemment rencontrés et la manière de les résoudre.

Aucun élément de ce type « xxxxxxxx »

Si vous recevez un message d'erreur comme celui-ci, cela signifie qu'il vous manque un plugin GStreamer :

```
WARNING: erroneous pipeline: no element "videoconvert"
```

Résolution :

En fonction de l'élément manquant, déterminez l'action appropriée :

- kvssink: Voir [the section called “Téléchargez et compilez le SDK C++ Producer”](#).
- libcamerasrc: Voir [the section called “Erreur « Échec de l'activation du pool de mémoire tampon »”](#) pour installer l'élément libcamerasrc GStreamer.
- omxh264enc ou v4l2h264enc :

Suivez [the section called “Installation des prérequis logiciels”](#) pour installer toutes les bibliothèques GStreamer. Si vous les avez tous installés et que ces éléments n'apparaissent pas, cela signifie que votre Raspberry Pi n'a pas le matériel nécessaire. Utilisez x264enc plutôt que l'encodeur logiciel.

- Autre : suivez [the section called “Installation des prérequis logiciels”](#) pour installer toutes les bibliothèques GStreamer. Différents éléments de GStreamer se trouvent dans les différents groupes de plugins GStreamer (bons, mauvais, moches), alors assurez-vous de tous les installer.

Erreur « Échec de l'activation du pool de mémoire tampon »

Si vous recevez une erreur comme celle-ci, cela signifie que le pipeline utilisé est en cours d'utilisation `v4l2src`, mais qu'il devrait l'utiliser à la place `libcamerasrc`.

```
ERROR bufferpool gstbufferpool.c:572:gst_buffer_pool_set_active:source:pool0:src start failed
WARN v4l2src gstv4l2src.c:976:gst_v4l2src_decide_allocation: error: Failed to allocate required memory.
WARN v4l2src gstv4l2src.c:976:gst_v4l2src_decide_allocation: error: Buffer pool activation failed
WARN basesrc gstbasesrc.c:3352:gst_base_src_prepare_allocation: Subclass failed to decide allocation
Error received from element source: Failed to allocate required memory.
WARN basesrc gstbasesrc.c:3132:gst_base_src_loop: error: Internal data stream error.
Debugging information: ../sys/v4l2/gstv4l2src.c(976): gst_v4l2src_decide_allocation(): /GstPipeline:live-kinesis-pipeline/GstV4l2Src:source:
Buffer pool activation failed
WARN basesrc gstbasesrc.c:3132:gst_base_src_loop: error: streaming stopped, reason not-negotiated (-4)
```

Par exemple, si vous utilisez le pipeline suivant alors que le module de caméra 2 n'est pas `libcamerasrc` installé, vous pouvez rencontrer cette erreur lorsque GStreamer essaie de détecter automatiquement les éléments à utiliser.

```
gst-launch-1.0 autovideosrc ! videoconvert ! autovideosink
```

Résolution :

Assurez-vous qu'il `libcamerasrc` est installé et utilisez-le comme élément source, plutôt que `v4l2src`. Tapez ce qui suit pour installer l'élément `libcamerasrc` GStreamer :

```
sudo apt-get update
sudo apt-get install gstreamer1.0-libcamera
```

Une fois `libcamerasrc` installé, si vous utilisez l'`autovideosrc` élément, GStreamer devrait automatiquement passer à la bonne source `libcamerasrc` au lieu de `v4l2src`.

Erreur de bus

Si vous recevez une erreur de bus peu après le démarrage kvssink (généralement, au moment où l'appel HTTP est PutMedia terminé), cela signifie que votre Raspberry Pi ne prend pas en charge l'accès à la mémoire non aligné. Les journaux ressembleront à ce qui suit :

```
INFO Camera camera.cpp:1197 configuring streams: (0) 640x480-YUV420
INFO RPI pisp.cpp:1450 Sensor: /base/axi/pcie@120000/rp1/i2c@88000/imx708@1a - Selected
  sensor format: 1536x864-SBGGR10_1X10 - Selected CFE format: 1536x864-PC1B
[INFO ] kinesisVideoStreamFormatChanged(): Stream format changed.
[DEBUG] setRequestHeader(): Appending header to request: user-agent -> AWS-SDK-KVS-CPP-
CLIENT/3.4.2/1.5.3 GCC/12.2.0 Linux/6.6.51+rpt-rpi-v8 aarch64 CPPSDK
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-stream-name -> demo-
stream
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-producer-start-
timestamp -> 1732012345.678
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-fragment-
acknowledgment-required -> 1
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-fragment-timecode-type
-> ABSOLUTE
[DEBUG] setRequestHeader(): Appending header to request: transfer-encoding -> chunked
[DEBUG] setRequestHeader(): Appending header to request: connection -> keep-alive
[INFO ] putStreamResultEvent(): Put stream result event. New upload handle 0
[WARN ] notifyDataAvailable(): [demo-stream] Failed to un-pause curl with error: 43.
  Curl object 0xe2f6f418
Bus error
```

Kinesis Video Streams PIC utilise un accès mémoire non aligné pour optimiser l'utilisation de la mémoire, ce qui n'est pas pris en charge par tous les appareils.

Résolution :

Pour utiliser le SDK en mode d'accès à la mémoire aligné, vous devez définir explicitement l'indicateur `ALIGNED_MEMORY_MODEL` CMake sur `ON` lors de la compilation `kvssink`, car il est défini par défaut sur `OFF`. Voir [the section called “Téléchargez et compilez le SDK C++ Producer”](#) pour des instructions plus détaillées.

L'horodatage se bloque et le pipeline s'arrête

Lors de l'utilisation `x264enc` dans un pipeline `GStreamer`, vous pouvez rencontrer des situations où la chronologie du pipeline ralentit considérablement ou s'arrête complètement en quelques secondes.

Cela se produit parce que les paramètres `x264enc` par défaut peuvent introduire une latence de codage élevée, qui dépasse la capacité de la mémoire tampon d'entrée par défaut. Par conséquent, la mémoire tampon d'entrée se remplit, ce qui provoque le blocage des éléments en amont et le blocage du pipeline.

Pour plus d'informations, consultez la documentation de [GStreamer](#).

Résolution :

Configurez `x264enc` avec l'option de `zerolatency` réglage. Cela réduit considérablement la latence d'encodage en optimisant les scénarios en temps réel, garantissant ainsi un traitement et une sortie plus rapides des images.

Exemple de configuration :

```
... ! x264enc tune=zerolatency byte-stream=true speed-preset=ultrafast bframes=0 key-int-max=60 ! ...
```

Note

Bien que cette solution empêche efficacement le blocage du pipeline, elle peut avoir un impact sur l'efficacité et la qualité de l'encodage. Pour les scénarios nécessitant à la fois une faible latence et une haute qualité, envisagez d'autres approches, telles que l'utilisation d'optimisations matérielles ou la recherche d'une webcam qui émet H.264 directement, en sautant cette étape de codage.

Pour de plus amples informations, veuillez consulter [the section called "Utiliser du matériel"](#).

Impossible d'exécuter plusieurs pipelines à partir du même périphérique **v4l2** en même temps

De tels appareils ne sont `/dev/video0` accessibles que par un seul processus à la fois. Si plusieurs processus tentent d'y accéder en même temps, le second attend que le premier soit terminé.

Résolution :

Créez un dispositif de bouclage permettant à plusieurs processus d'utiliser l'interface de bouclage en même temps. Pour plus d'informations, consultez [Stack Exchange](#).

Erreur de flux de données interne

Lorsque vous créez un pipeline GStreamer, vous connectez des éléments en liant le pad source d'un élément au pad récepteur d'un autre élément. Ce processus de liaison permet le flux de données de l'élément source vers l'élément récepteur, formant ainsi un pipeline de données.

Le message d'erreur « Pad link failed » dans le journal indique que GStreamer a rencontré un problème en essayant d'établir une connexion (lien) entre les pads de deux éléments de votre pipeline.

```
Pad link failed
Error received from element udpsrc0: Internal data stream error.
```

Résolution :

Déterminez quels éléments ne sont pas liés les uns aux autres. Pour réduire la portée du pipeline, supprimez des éléments du pipeline. Remplacez l'élément le plus à droite par `fakesink` des éléments et supprimez-les un par un.

Vous devrez peut-être ajuster les éléments [capsfilter](#), and/or modifier les éléments utilisés par votre pipeline.

Les cas courants demandent un `framerate` ou `resolution` que l'appareil photo ne prend pas en charge. `gst-device-monitor-1.0` À utiliser dans le terminal pour obtenir les valeurs prises en charge `frameratesresolutions`, et `formats`. Vous pouvez utiliser l'élément [Videoscale](#) GStreamer pour régler la résolution vidéo, et [Videorate](#) pour ajuster la fréquence d'images vidéo.

Pour vérifier les formats pris en charge pour un élément GStreamer individuel, saisissez `gst-inspect-1.0 element-name` le terminal.

Lire le contenu multimédia de votre flux vidéo Kinesis

Ouvrez la console [Kinesis Video Streams](#) et sélectionnez le nom du flux que vous avez créé.

Le flux vidéo qui est envoyé à partir de l'appareil Raspberry Pi apparaît dans la console.

Note

Quelques secondes peuvent s'écouler avant que la vidéo n'apparaisse dans la console.

Consultez [the section called “Fournir une région à kvssink”](#) pour plus d'informations sur la façon dont kvssink détermine la région à utiliser.

Le contenu multimédia met beaucoup de temps à se charger dans le AWS Management Console

⚠ Important

L'expérience de lecture sur console est différente de l'expérience de lecture HLS et DASH. Utilisez également l'exemple de [page Web hébergée par](#) le lecteur multimédia GitHub pour tester la lecture. Le code source de la page Web se trouve [ici](#).

Le contenu multimédia peut se charger lentement dans la console en raison d'une faible bande passante réseau ou d'un périphérique restreint, mais cela peut également être dû au codage vidéo et à la fragmentation.

Principes de base de l'encodage vidéo :

- H.264 et H.265 les encodeurs utilisent key-frames (I-Frames) et predicted-frames (P-Frames) pour une compression efficace.
- Key-frames contiennent des données d'image complètes, P-frames mais ne contiennent que des modifications par rapport aux images précédentes.
- L' « intervalle entre images clés » détermine la fréquence à laquelle les images-clés apparaissent dans le flux vidéo.

Fragmentation dans le streaming :

- Dans Kinesis Video Streams, les nouveaux fragments commencent par chaque image-clé. Pour de plus amples informations, veuillez consulter [the section called “Modèle de données”](#).
- La longueur du fragment (en secondes) peut être estimée comme suit : intervalle entre images clé ÷ fréquence d'images

Exemple :

Pour un flux avec un intervalle d'images clé de 30 et une fréquence d'images de 15 images par seconde :

Longueur du fragment = $30 \div 15 = 2$ secondes

En raison de l'allongement des intervalles entre les images-clés, les fragments plus longs augmentent la latence dans le streaming multimédia.

Résolution :

Pour améliorer les temps de chargement, pensez à réduire l'intervalle entre les images-clés. Cela créera des fragments plus courts, diminuera la latence, mais augmentera également la taille du fichier vidéo.

Pour l'élément `x264enc` GStreamer, vous pouvez définir explicitement l'intervalle entre les images-clés via la propriété : `key-int-max`

```
x264enc bframes=0 key-int-max=60
```

Lorsque vous examinez les résultats du journal, notez la fréquence à laquelle le client de téléchargement reçoit des ACK de Kinesis Video Streams. Plus le nombre d'images-clés générées est élevé, plus les ACK renvoyés sont nombreux.

Le média est déformé ou contient des artefacts

Pour résoudre ce problème, assurez-vous que tous les câbles sont bien connectés. Vérifiez la sortie `libcamera-hello` (ou `raspistill` pour les anciennes caméras Pi) des modules de caméra.

Dans votre pipeline GStreamer, remplacez `kvssink` par `autovideosink` ou `matroskamux` et `filesink` Par exemple :

```
... x264enc tune=zerolatency speed-preset=ultrafast bframes=0 key-int-max=60 byte-stream=true ! h264parse ! matroskamux ! filesink location=output.mkv
```

Vérifiez le fichier de sortie `filesink` ou le lecteur multimédia qui s'ouvre lors de l'utilisation `autovideosink` pour voir si les artefacts sont également présents.

Passez également en revue le résultat du pipeline suivant :

```
gst-launch-1.0 autovideosrc ! videoconvert ! autovideosink
```

L'ajout d'éléments à votre pipeline, tels que le [dewarp](#), peut corriger les résultats des caméras Fish Eye.

Vérifiez les codecs de sortie pris en charge par votre caméra et ajustez les éléments selon vos besoins.

Par exemple, si votre caméra USB ne prend en charge que la sortie JPEG, vous devrez utiliser les `jpegdec` éléments `jpegparse` et pour transformer le support avant de l'encoder pour H.264 l'utiliser `x264enc`. Recherchez de l'aide sur les forums GStreamer pour d'autres utilisateurs utilisant des configurations de `and/or webcams` similaires à des pipelines.

Résolution des problèmes de compilation sur le SDK C++ Producer pour Raspberry Pi

Si vous rencontrez un problème de compilation et que vous souhaitez essayer différents arguments CMake, assurez-vous d'effectuer une nouvelle génération. Supprimez les `build` dossiers `open-sourcedependency`, et avant de réessayer.

Problèmes de création avec OpenSSL

Si vous recevez un résultat similaire à ce qui suit, cela indique qu'OpenSSL a mal détecté l'architecture de votre système.

```
crypto/md5/md5-aarch64.S: Assembler messages:
crypto/md5/md5-aarch64.S:3: Error: unrecognized symbol type ""
crypto/md5/md5-aarch64.S:6: Error: bad instruction `stp x19,x20,[sp,#-80]!'
crypto/md5/md5-aarch64.S:7: Error: bad instruction `stp x21,x22,[sp,#16]'
crypto/md5/md5-aarch64.S:8: Error: bad instruction `stp x23,x24,[sp,#32]'
crypto/md5/md5-aarch64.S:9: Error: bad instruction `stp x25,x26,[sp,#48]'
```

Dans cet exemple, il tente de créer une version 64 bits (`linux-aarch64`) alors que ce Raspberry Pi est en réalité 32 bits. Certains appareils Raspberry Pi ont un noyau 64 bits, mais un espace utilisateur 32 bits.

Déterminez l'architecture pour laquelle OpenSSL essaie de créer. Vous pouvez trouver la ligne de journal lors de l'`configure` étape pour OpenSSL :

```
[ 33%] Performing update step for 'project_libopenssl'
-- Already at requested tag: OpenSSL_1_1_1t
[ 44%] No patch step for 'project_libopenssl'
[ 55%] Performing configure step for 'project_libopenssl'
Operating system: x86_64-whatever-linux2
Configuring OpenSSL version 1.1.1t (0x1010114fL) for linux-x86_64
Using os-specific seed configuration
Creating configdata.pm
Creating Makefile
```

Vérifiez l'architecture de votre système :

- Vérifiez le nombre de bits du noyau : exécutez `uname -m`
- Vérifiez le nombre de bits de l'espace utilisateur : exécuter `getconf LONG_BIT`

Vous pouvez également consulter les informations relatives à votre processeur à l'aide des `lscpu` commandes `cat /proc/cpuinfo` ou.

Résolution :

Pour résoudre ce problème, ajoutez l'argument CMake suivant lors de la création, afin de vous assurer qu'OpenSSL se construit correctement pour l'architecture ARM 32 bits :

```
-DBUILD_OPENSSL_PLATFORM=linux-armv4
```

Résoudre les problèmes de chargement de **kvssink** dans GStreamer

Confirmer `GST_PLUGIN_PATH`

Assurez-vous que la variable d'`GST_PLUGIN_PATH` environnement de votre session shell en cours pointe vers le répertoire contenant `kvssink`. Les variables d'environnement étant spécifiques à une session, vous devez les définir pour chaque nouvelle session. Pour rendre cette modification permanente, consultez « Mettre à jour le script de démarrage de votre shell pour inclure la définition de la variable d'environnement `GST_PLUGIN_PATH` ».

Erreur : Impossible d'ouvrir le fichier objet partagé : aucun fichier ou répertoire de ce type

Si le message d'erreur s'affiche `Cannot open shared object file: No such file or directory`, exécutez la commande suivante :

```
gst-inspect-1.0 /path/to/libgstkvssink.so
```

Si vous recevez le résultat suivant, cela indique que l'éditeur de liens dynamiques ne parvient pas à localiser les bibliothèques requises pour `kvssink`. Cela se produit généralement pour les raisons suivantes :

- Déménagement `kvssink` vers un endroit différent de celui où il a été construit.
- Cross-compiling pour une architecture de processeur incorrecte.
- Il manque une dépendance obligatoire.

Sortie :

```
WARNING: erroneous pipeline: no element "kvssink"  
error while loading shared libraries: libcproducer.so: cannot open shared object file:  
No such file or directory
```

Résolution :

Pour les bibliothèques déplacées, ajoutez le répertoire contenant les bibliothèques manquantes à `LD_LIBRARY_PATH`.

À partir du répertoire racine du dépôt d'origine, vous pouvez localiser la bibliothèque manquante à l'aide de l'utilitaire. Dans le terminal, tapez :

```
find . -name "*libcproducer*"
```

Sortie :

```
./build/dependency/libkvscproducer/kvscproducer-src/libcproducer.so
```

Le séparateur de chemin de fichier sur les appareils Linux est : `.` La commande ci-dessous ajoute un nouveau chemin de dossier à la variable d'environnement existante `LD_LIBRARY_PATH`, en préservant les valeurs précédentes.

Dans votre terminal, saisissez :

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:./path/to/build/dependency/libkvscproducer/  
kvscproducer-src
```

Important

Les variables d'environnement sont spécifiques à la session. Pour conserver les modifications entre les sessions, modifiez le script de démarrage de votre shell.

Vous devrez peut-être également ajouter le `open-source/local/lib` à votre `$LD_LIBRARY_PATH`.

Erreur `./path/to/libcproducer.so.1` : en-tête ELF non valide

Si vous recevez cette erreur lors du chargement des bibliothèques partagées, cela peut être dû à des liens symboliques rompus (symlinks). Les liens symboliques peuvent être interrompus si le système d'exploitation de la machine hôte ne correspond pas à celui de la machine cible. Par exemple, la compilation croisée sur un MacBook pour un Raspberry Pi.

Une autre cause possible est que les fichiers binaires créés étaient destinés à la mauvaise architecture. Par exemple, si les fichiers binaires ont été conçus pour x86 (le Raspberry Pi utilise des processeurs ARM).

Accédez à l'emplacement de bibliothèque spécifié dans l'erreur et tapez : `ls -la` pour inspecter la bibliothèquesymlinks.

Réponse :

```
drwxr-xr-x 16 me staff      512 Sep 10 17:16 .
drwxr-xr-x  7 me staff      224 Jan  6 23:46 ..
drwxr-xr-x  4 me staff      128 Sep 10 17:16 engines-1.1
-rwxr-xr-x  1 me staff 2294496 Sep 10 17:16 libcrypto.1.1.so
-rw-r--r--  1 me staff 4002848 Sep 10 17:16 libcrypto.a
lrwxr-xr-x  1 me staff      19 Sep 10 17:16 libcrypto.so -> libcrypto.1.1.so
-rwxr-xr-x  1 me staff 631176 Sep 10 17:12 liblog4cplus-2.0.3.so
lrwxr-xr-x  1 me staff      24 Sep 10 17:12 liblog4cplus.so -> liblog4cplus-2.0.3.so
-rwxr-xr-x  1 me staff   1012 Sep 10 17:12 liblog4cplus.a
-rwxr-xr-x  1 me staff 694328 Sep 10 17:12 liblog4cplusU-2.0.3.so
lrwxr-xr-x  1 me staff      25 Sep 10 17:12 liblog4cplusU.dylib ->
liblog4cplusU-2.0.3.so
-rwxr-xr-x  1 me staff   1017 Sep 10 17:12 liblog4cplusU.a
-rwxr-xr-x  1 me staff 536416 Sep 10 17:16 libssl.1.1.so
-rw-r--r--  1 me staff 795184 Sep 10 17:16 libssl.a
lrwxr-xr-x  1 me staff      16 Sep 10 17:16 libssl.so -> libssl.1.1.so
drwxr-xr-x  6 me staff      192 Sep 10 17:16 pkgconfig
```

Dans l'exemple de sortie ci-dessus, symlinks ils ne sont pas cassés. Les Broken n'symlinksauront pas de flèches pointant vers leurs cibles.

Résolution :

Il existe deux options pour corriger les liens symboliques :

- **Recommandé** : Recréez le symlink avec la `ln` commande. Type :

```
ln -s /path/to/actual/library /path/to/symlink
```

- Copiez le fichier de bibliothèque actuel et renommez-le pour qu'il corresponde `ausymLink`.

Note

Cette option entraîne une augmentation de l'utilisation du stockage.

La meilleure pratique consiste à compiler sur le même système d'exploitation à l'aide d'outils tels que Docker pour éviter les problèmes de compilation croisée.

Dépendances manquantes :

Si le nom de bibliothèque manquant commence par `libkvs`, consultez la section « bibliothèques déplacées » ci-dessus pour installer les bibliothèques Kinesis Video Streams de l'appareil hôte vers l'appareil cible.

Dans le cas [the section called “Installation des prérequis logiciels”](#) contraire, assurez-vous que tous les logiciels open source requis sont installés sur l'appareil cible.

Référence du code d'erreur

Cette section contient des informations relatives aux erreurs et codes d'état pour le [Transférer vers Kinesis Video Streams](#).

Pour obtenir plus d'informations sur les solutions concernant des problèmes courants, consultez [Résolution des problèmes](#).

Rubriques

- [Erreurs et codes d'état renvoyés par les PutFrame rappels - Code indépendant de la plate-forme \(PIC\)](#)
- [Erreurs et codes d'état renvoyés par les PutFrame rappels - bibliothèque C Producer](#)

Erreurs et codes d'état renvoyés par les PutFrame rappels - Code indépendant de la plate-forme (PIC)

Les sections suivantes contiennent des informations d'erreur et d'état renvoyées par des rappels pour l'PutFrameopération dans le code PIC (Platform Independent Code).

Rubriques

- [Codes d'erreur et d'état renvoyés par la bibliothèque cliente](#)
- [Codes d'erreur et d'état renvoyés par la bibliothèque de durées](#)
- [Codes d'erreur et d'état renvoyés par la bibliothèque commune](#)
- [Codes d'erreur et d'état renvoyés par la bibliothèque heap](#)
- [Codes d'erreur et d'état renvoyés par la bibliothèque MKVGen](#)
- [Codes d'erreur et d'état renvoyés par la bibliothèque Trace](#)
- [Codes d'erreur et d'état renvoyés par la bibliothèque Utils](#)
- [Codes d'erreur et d'état renvoyés par la bibliothèque View](#)

Codes d'erreur et d'état renvoyés par la bibliothèque cliente

Le tableau suivant contient les informations d'erreur et d'état renvoyées par les méthodes de la bibliothèque Kinesis Video Client Streams.

| Code | Message | Description | Action recommandée |
|------------|-----------------------------------|--|--|
| 0x52000001 | STATUS_MAX_STREAM_COUNT | Le nombre maximal de flux a été atteint. | Spécifiez un plus grand nombre max. de flux dans DeviceInfo que le nombre spécifié dans Quotas du SDK pour les producteurs . |
| 0x52000002 | STATUS_MIN_STREAM_COUNT | Erreur relative au nombre minimal de flux. | Spécifiez le nombre maximum de flux entrants supérieurs à zeroDeviceInfo . |
| 0x52000003 | STATUS_INVALID_DEVICE_NAME_LENGTH | La longueur du nom de périphérique n'est pas valide. | Reportez-vous à la longueur maximale du nom de l'appareil en caractères spécifiée dans Quotas du SDK pour les producteurs . |

| Code | Message | Description | Action recommandée |
|------------|------------------------------------|---|--|
| 0x52000004 | STATUS_INVALID_DEVICE_INFO_VERSION | La version de la structure DeviceInfo est non valide. | Spécifiez la version actuelle correcte de la structure. |
| 0x52000005 | STATUS_MAX_TAG_COUNT | Le nombre maximal de balises a été atteint. | Reportez-vous au nombre maximal de balises actuel spécifié dans Quotas du SDK pour les producteurs . |
| 0x52000006 | STATUS_DEVICE_FINGERPRINT_LENGTH | | |
| 0x52000007 | STATUS_INVALID_CALLBACKS_VERSION | La version de la structure Callbacks est non valide. | Spécifiez la version actuelle correcte de la structure. |
| 0x52000008 | STATUS_INVALID_STREAM_INFO_VERSION | La version de la structure StreamInfo est non valide. | Spécifiez la version actuelle correcte de la structure. |
| 0x52000009 | STATUS_INVALID_STREAM_NAME_LENGTH | La longueur du nom de flux n'est pas valide. | Reportez-vous à la longueur maximale du nom du flux en caractères spécifiée dans Quotas du SDK pour les producteurs . |
| 0x5200000a | STATUS_INVALID_STORAGE_SIZE | Une taille de stockage non valide a été spécifiée. | La taille de stockage en octets doit être comprise dans les limites spécifiées dans Quotas du SDK pour les producteurs . |

| Code | Message | Description | Action recommandée |
|------------|-------------------------------------|--|--|
| 0x5200000b | STATUS_INVALID_DIRECTORY_LENGTH | La longueur de chaîne du répertoire et racine est non valide. | Reportez-vous à la longueur maximale du chemin du répertoire et racine spécifiée dans Quotas du SDK pour les producteurs . |
| 0x5200000c | STATUS_INVALID_SPILL_RATIO | Taux de déversement non valide. | Exprimez le taux de déversement sous forme de pourcentage compris entre 0 et 100. |
| 0x5200000d | STATUS_INVALID_STORAGE_INFO_VERSION | La version de la structure StorageInfo est non valide. | Spécifiez la version actuelle correcte de la structure. |
| 0x5200000e | STATUS_INVALID_STREAM_STATE | Le flux est dans un état qui ne permet pas d'effectuer l'opération actuelle. | Le plus souvent, cette erreur se produit lorsque le SDK n'atteint pas l'état requis pour effectuer l'opération demandée. Par exemple, elle se produit si l'appel de l'API GetStreamingEndpoint échoue et que l'application cliente l'ignore et continue de placer des images dans le flux. |

| Code | Message | Description | Action recommandée |
|------------|-------------------------------------|--|--|
| 0x5200000f | STATUS_SERVICE_CALLBACKS_MISSING | La structure <code>Callbacks</code> a des points d'entrée de fonction manquants pour certaines fonctions obligatoires. | Vérifiez que les rappels obligatoires sont implémentés dans l'application cliente. Cette erreur n'est visible que pour les clients PIC (Platform Independent Code). C++ et d'autres wrappers de plus haut niveau satisfont à ces appels. |
| 0x52000010 | STATUS_SERVICE_NOT_AUTHORIZED_ERROR | Non autorisé. | Vérifiez le jeton de sécurité, le certificat, l'intégration du jeton de sécurité et l'expiration. Vérifiez que le jeton possède les droits appropriés associés. Pour les exemples d'applications Kinesis Video Streams, vérifiez que la variable d'environnement est correctement définie. |
| 0x52000011 | STATUS_DESCRIPTOR_CALL_FAILED | Échec de l'API <code>DescribeStream</code> . | Cette erreur est renvoyée après l'échec de la nouvelle tentative d'exécution de l'API <code>DescribeStream</code> . Le client PIC renvoie cette erreur après avoir arrêté de réessayer. |

| Code | Message | Description | Action recommandée |
|------------|--|--|---|
| 0x52000012 | STATUS_INVALID_DESCRIBE_STREAM_RESPONSE | La structure DescribeStreamResponse est non valide. | La structure qui a été transmise à DescribeStreamResultEvent a pour valeur NULL ou contient des éléments non valides, tels qu'un nom Amazon Resource Name (ARN) de valeur NULL. |
| 0x52000013 | STATUS_STREAM_IS_BEING_DELETED_ERROR | Le flux est en cours de suppression. | Un échec d'API a été causé par le flux en cours de suppression. Vérifiez qu'aucun autre processus n'essaie de supprimer le flux pendant qu'il est en cours d'utilisation. |
| 0x52000014 | STATUS_SERVICE_CALL_INVALID_ARG_ERROR | Des arguments non valides ont été spécifiés pour l'appel de service. | Le backend renvoie cette erreur lorsqu'un argument d'appel de service n'est pas valide ou lorsque le SDK rencontre une erreur qu'il ne peut pas interpréter. |
| 0x52000015 | STATUS_SERVICE_CALL_DEVICE_NOT_FOUND_ERROR | L'appareil est introuvable. | Vérifiez que l'appareil n'est pas supprimé pendant son utilisation. |
| 0x52000016 | STATUS_SERVICE_CALL_DEVICE_NOT_PROVISIONED_ERROR | L'appareil n'a pas été mis en service. | Vérifiez que l'appareil a été approvisionné. |

| Code | Message | Description | Action recommandée |
|------------|--|--|--|
| 0x52000017 | STATUS_SERVICE_CALL_RESOURCE_NOT_FOUND_ERROR | Une ressource générique introuvable a été renvoyée du service. | Cette erreur se produit lorsque le service ne peut pas localiser la ressource (par exemple, un flux). Cela peut signifier des choses différentes dans différents contextes, mais la cause probable est l'utilisation d'API avant la création du flux. L'utilisation du SDK confirme que le flux a été créé en premier. |
| 0x52000018 | STATUS_INVALID_AUTH_LEN | Longueur non valide des infos d'authentification. | Reportez-vous aux valeurs actuelles spécifiées dans Quotas du SDK pour les producteurs . |
| 0x52000019 | STATUS_CREATE_STREAM_CALL_FAILED | L'appel de l'API <code>CreateStream</code> a échoué. | Reportez-vous à la chaîne d'erreur pour obtenir de plus amples informations sur la raison pour laquelle l'opération a échoué. |
| 0x5200002a | STATUS_GET_STREAMING_TOKEN_CALL_FAILED | L'appel de <code>GetStreamingToken</code> a échoué. | Reportez-vous à la chaîne d'erreur pour obtenir de plus amples informations sur la raison pour laquelle l'opération a échoué. |

| Code | Message | Description | Action recommandée |
|------------|---|---|---|
| 0x5200002b | STATUS_GET_STREAMING_ENDPOINT_CALL_FAILED | L'appel de l'API <code>GetStreamingEndpoint</code> a échoué. | Reportez-vous à la chaîne d'erreur pour obtenir de plus amples informations sur la raison pour laquelle l'opération a échoué. |
| 0x5200002c | STATUS_INVALID_URI_LEN | Un URI d'une longueur de chaîne non valide a été renvoyé de l'API <code>GetStreamingEndpoint</code> . | Reportez-vous aux valeurs maximales actuelles spécifiées dans Quotas du SDK pour les producteurs . |
| 0x5200002d | STATUS_PUT_STREAM_CALL_FAILED | L'appel de l'API <code>PutMedia</code> a échoué. | Reportez-vous à la chaîne d'erreur pour obtenir de plus amples informations sur la raison pour laquelle l'opération a échoué. |

| Code | Message | Description | Action recommandée |
|------------|----------------------------|--|---|
| 0x5200002e | STATUS_STORE_OUT_OF_MEMORY | Le magasin de contenu manque de mémoire. | <p>Le magasin de contenu est partagé entre les flux et doit avoir une capacité suffisante pour stocker les durées maximales pour tous les flux + ~20 % (pour tenir compte de la défragmentation). Il est important de ne pas saturer l'espace de stockage. Choisissez des valeurs pour la durée maximale par flux qui correspondent aux tolérances cumulées de la taille de stockage et de la latence. Nous recommandons de supprimer les cadres lorsqu'ils sortent de la fenêtre d'affichage du contenu plutôt que d'être simplement placés (pression sur la mémoire du magasin de contenu). Cela est dû au fait que la suppression des cadres déclenche les rappels de notification de la pression du cours d'eau. Ensuite, l'application peut ajuster les composants multimédias en amont (comme l'encodeur) pour fluidifier la vitesse de transmission,</p> |

| Code | Message | Description | Action recommandée |
|------------|---|--|---|
| | | | supprimer des images ou agir en conséquence. |
| 0x5200002f | STATUS_NO_MORE_DATA_AVAILABLE | Aucune donnée n'est plus disponible actuellement pour un flux. | Il s'agit d'un résultat potentiellement valide lorsque le pipeline multimédia produit plus lentement que le thread réseau ne consomme les trames à envoyer au service. Higher-level les clients (par exemple, C++, Java ou Android) ne voient pas cet avertissement car il est géré en interne. |
| 0x52000030 | STATUS_INVALID_TAG_VERSION | La version de la structure Tag est non valide. | Spécifiez la version actuelle correcte de la structure. |
| 0x52000031 | STATUS_SERVICE_CALL_UNKNOWN_ERROR | Une erreur inconnue ou générique a été renvoyée à partir de la pile de mise en réseau. | Consultez les journaux pour obtenir des informations plus détaillées. |
| 0x52000032 | STATUS_SERVICE_CALL_RESOURCE_IN_USE_ERROR | Ressource en cours d'utilisation. | Renvoyée du service. Pour plus d'informations, consultez le manuel de référence de l'API Kinesis Video Streams. |

| Code | Message | Description | Action recommandée |
|------------|---------------------------------------|--|--|
| 0x52000033 | STATUS_SERVICE_CLIENT_LIMIT_ERROR | Limite du client. | Renvoyée du service. Pour plus d'informations, consultez le manuel de référence de l'API Kinesis Video Streams. |
| 0x52000034 | STATUS_SERVICE_DEVICE_LIMIT_ERROR | Limite de l'appareil. | Renvoyée du service. Pour plus d'informations, consultez le manuel de référence de l'API Kinesis Video Streams. |
| 0x52000035 | STATUS_SERVICE_STREAM_LIMIT_ERROR | Limite du flux. | Renvoyée du service. Pour plus d'informations, consultez le manuel de référence de l'API Kinesis Video Streams. |
| 0x52000036 | STATUS_SERVICE_RESOURCE_DELETED_ERROR | La ressource a été supprimée ou est en cours de suppression. | Renvoyée du service. Pour plus d'informations, consultez le manuel de référence de l'API Kinesis Video Streams. |
| 0x52000037 | STATUS_SERVICE_TIMEOUT_ERROR | L'appel du service a expiré. | L'appel d'une API de service particulière a entraîné le dépassement d'un délai d'attente. Vérifiez que vous disposez d'une connexion réseau valide. Le code PIC retente automatiquement l'opération. |

| Code | Message | Description | Action recommandée |
|------------|---|--|---|
| 0x52000038 | STATUS_STREAM_READ_CALLBACK_FAILED | Notification de préparation de flux. | Cette notification est envoyée à partir du code PIC au client pour indiquer que le flux asynchrone a été créé. |
| 0x52000039 | STATUS_DEVICE_TAGSCOUNT_NON_ZERO_TAGSNONE | Des balises non valides ont été spécifiées. | Le nombre de balises n'est pas nul, mais les balises sont vides. Vérifiez que les balises sont spécifiées ou que le nombre est égal à zéro. |
| 0x5200003a | STATUS_INVALID_STREAM_DESCRIPTION_VERSION | La version de la structure StreamDescription est non valide. | Spécifiez la version actuelle correcte de la structure. |
| 0x5200003b | STATUS_INVALID_TAG_NAME_LEN | La longueur du nom de balise n'est pas valide. | Reportez-vous aux limites pour le nom de balise qui sont spécifiées dans Quotas du SDK pour les producteurs . |
| 0x5200003c | STATUS_INVALID_TAG_VALUE_LEN | La longueur de la valeur de balise n'est pas valide. | Reportez-vous aux limites pour la valeur de balise qui sont spécifiées dans Quotas du SDK pour les producteurs . |

| Code | Message | Description | Action recommandée |
|------------|---------------------------------------|---|---|
| 0x5200003d | STATUS_TAG_STREAM_CALL_FAILED | L'API TagResource a échoué. | L'appel de l'API TagResource a échoué. Recherchez une connexion réseau valide. Consultez les journaux pour obtenir plus d'informations sur l'échec. |
| 0x5200003e | STATUS_INVALID_CUSTOM_DATA | Des données personnalisées non valides appellent les API de code PIC. | Des données personnalisées non valides ont été spécifiées dans un appel aux API de code PIC. Cela peut se produire uniquement dans les clients qui utilisent directement le code PIC. |
| 0x5200003f | STATUS_INVALID_CREATE_STREAM_RESPONSE | La structure CreateStreamResponse est non valide. | La structure ou ses champs membres ne sont pas valides (c'est-à-dire que le nom ARN est null ou plus grand que ce qui est spécifié dans Quotas du SDK pour les producteurs). |

| Code | Message | Description | Action recommandée |
|------------|-------------------------------------|---|---|
| 0x52000040 | STATUS_CLIENT_AUTH_CALL_FAILED | L'authentification du client a échoué. | Le PIC n'a pas réussi à obtenir les informations d'authentification appropriées (AccessKeyId ou SecretAccessKey) après plusieurs tentatives. Vérifiez l'intégration de l'authentification. Les exemples d'application utilisent des variables d'environnement pour transmettre des informations d'identification à la bibliothèque Producteur C++. |
| 0x52000041 | STATUS_GET_CLIENT_TOKEN_CALL_FAILED | L'appel pour obtenir le jeton de sécurité a échoué. | Cela peut se produire uniquement pour les clients qui utilisent directement le code PIC. Après un certain nombre de tentatives, l'appel échoue avec cette erreur. |
| 0x52000042 | STATUS_CLIENT_PROVISION_CALL_FAILED | Erreur de mise en service. | Le provisionnement n'est pas implémenté. |
| 0x52000043 | STATUS_CREATE_CLIENT_CALL_FAILED | Impossible de créer le client producteur. | Une erreur générique a été renvoyée par le code PIC après un certain nombre de tentatives lors de l'échec de la création du client. |

| Code | Message | Description | Action recommandée |
|------------|---------------------------------------|---|---|
| 0x52000044 | STATUS_CLIENT_READY_CALLBACK_FAILED | Impossible d'obtenir le client producteur à l'état READY. | Erreur renvoyée par la machine d'état PIC si le code PIC ne passe pas à l'état READY. Consultez les journaux pour obtenir plus d'informations sur la cause principale. |
| 0x52000045 | STATUS_TAG_CLIENT_CALL_FAILED | Échec de TagResource pour le client producteur. | L'appel de l'API TagResource a échoué pour le client producteur. Consultez les journaux pour obtenir plus d'informations sur la cause principale. |
| 0x52000046 | STATUS_INVALID_CREATE_DEVICE_RESPONSE | Device/Producer échec de création. | Les SDK de niveau supérieur (par exemple, C++ ou Java) n'implémentent pas encore l'API de création d'appareils ou de producteurs. Les clients qui utilisent le code PIC directement peuvent indiquer un échec en utilisant la notification des résultats. |

| Code | Message | Description | Action recommandée |
|------------|---|---|---|
| 0x52000047 | STATUS_ACK_TIMESTAMP_NOT_IN_VIEW_WINDOW | L'horodatage de l'ACK reçu n'est pas dans la vue. | Cette erreur se produit si l'image correspondant à l'ACK reçu sort de la fenêtre d'affichage de contenu. En règle générale, cela se produit si la livraison ACK est lente. Cela peut être interprété comme un avertissement et une indication que la liaison descendante est lente. |
| 0x52000048 | STATUS_INVALID_FRAGMENT_ACK_VERSION | La version de la structure <code>FragmentAck</code> est non valide. | Spécifiez la version actuelle correcte de la structure <code>FragmentAck</code> . |
| 0x52000049 | STATUS_INVALID_TOKEN_EXPIRATION | Expiration non valide du jeton de sécurité. | À l'avenir, l'expiration du jeton de sécurité devrait avoir un horodatage absolu supérieur à l'horodatage actuel, avec un délai de grâce. Pour les limites de la période de grâce, reportez-vous à Quotas du SDK pour les producteurs . |

| Code | Message | Description | Action recommandée |
|------------|---------------------------------|-------------------------------------|---|
| 0x5200004a | STATUS_END_OF_STREAM | Indicateur de fin de flux (EOS). | Dans l'appel de l'API <code>GetStreamData</code> , indique que la session de la poignée de chargement en cours est terminée. Cela se produit si la session se termine ou présente des erreurs, ou si le jeton de session a expiré et que la session est en cours de rotation. |
| 0x5200004b | STATUS_DUPLICATE_STREAM_NAME | Nom de flux en double. | Plusieurs flux ne peuvent pas avoir le même nom de flux. Choisissez un nom unique pour le flux. |
| 0x5200004c | STATUS_INVALID_RETENTION_PERIOD | Période de conservation non valide. | Une période de conservation non valide est spécifiée dans la structure <code>StreamInfo</code> . Pour plus d'informations sur la plage de valeurs valides pour la période de conservation, reportez-vous à Quotas du SDK pour les producteurs . |

| Code | Message | Description | Action recommandée |
|------------|-----------------------------------|-----------------------------|--|
| 0x5200004d | STATUS_INVALID_ACK_KEY_START | Non valide FragmentAck . | Impossible d'analyser la chaîne ACK de fragment. Indicateur de début de clé non valide. La chaîne ACK de fragment peut être endommagée. Elle peut se corriger automatiquement et cette erreur peut être traitée comme un avertissement. |
| 0x5200004e | STATUS_INVALID_DUPLICATE_KEY_NAME | Non valide FragmentAck . | Impossible d'analyser la chaîne ACK de fragment. Plusieurs clés ont le même nom. La chaîne ACK de fragment peut être endommagée. Elle peut se corriger automatiquement et cette erreur peut être traitée comme un avertissement. |
| 0x5200004f | STATUS_INVALID_VALUE_START | Non valide FragmentAck . | Impossible d'analyser la chaîne ACK de fragment en raison d'un indicateur de début de valeur de clé non valide. La chaîne ACK de fragment peut être endommagée. Elle peut se corriger automatiquement, et cette erreur peut être traitée comme un avertissement. |

| Code | Message | Description | Action recommandée |
|------------|--|--|---|
| 0x52000050 | STATUS_INVALID_ACK_ININVALID_VALUE_END | Non valide FragmentAck . | Impossible d'analyser la chaîne ACK de fragment en raison d'un indicateur de fin de valeur de clé non valide. La chaîne ACK de fragment peut être endommagée. Elle peut se corriger automatiquement et cette erreur peut être traitée comme un avertissement. |
| 0x52000051 | STATUS_INVALID_PARSED_ACK_TYPE | Non valide FragmentAck . | Impossible d'analyser la chaîne ACK de fragment car un type ACK non valide a été spécifié. |
| 0x52000052 | STATUS_STREAM_HAS_BEEN_STOPPED | Le flux a été arrêté. | Le flux a été arrêté, mais une image est encore en train d'être placée dans le flux. |
| 0x52000053 | STATUS_INVALID_STREAM_METRICS_VERSION | La version de la structure StreamMetrics est non valide. | Spécifiez la version actuelle correcte de la structure StreamMetrics . |
| 0x52000054 | STATUS_INVALID_CLIENT_METRICS_VERSION | La version de la structure ClientMetrics est non valide. | Spécifiez la version actuelle correcte de la structure ClientMetrics . |

| Code | Message | Description | Action recommandée |
|------------|--------------------------------------|--|---|
| 0x52000055 | STATUS_INVALID_CLIENT_READY_STATE | L'initialisation du producteur n'a pas atteint un état READY. | Impossible d'atteindre l'état READY lors de l'initialisation du client producteur. Consultez les journaux pour en savoir plus. |
| 0x52000056 | STATUS_STATE_MACHINE_STATE_NOT_FOUND | Erreur interne de machine d'état. | Erreur qui n'est pas visible publiquement. |
| 0x52000057 | STATUS_INVALID_FRAGMENT_ACK_TYPE | Un type ACK non valide est spécifié dans la structure <code>FragmentAck</code> . | La structure <code>FragmentAck</code> doit contenir les types ACK définis dans l'en-tête public. |
| 0x52000058 | STATUS_INVALID_STREAM_READY_STATE | Erreur de transition de machine d'état interne. | Erreur qui n'est pas visible publiquement. |
| 0x52000059 | STATUS_CLIENT_FREE_BEFORE_STREAM | L'objet de flux a été libéré après la libération du producteur. | Il y a eu une tentative pour libérer un objet de flux après la libération de l'objet producteur. Cela peut se produire uniquement dans des clients qui utilisent directement PIC. |

| Code | Message | Description | Action recommandée |
|------------|---|--------------------------------|--|
| 0x5200005a | STATUS_ALLOCATION_SIZE_SMALLER_THAN_REQUESTED | Erreur interne de stockage. | Une erreur interne indiquant que la taille d'allocation réelle provenant du magasin de contenu est inférieure à la taille du cadre et du fragment empaquetés. |
| 0x5200005b | STATUS_VIEW_ITEM_SIZE_GREATER_THAN_ALLOCATION | Erreur interne de stockage. | La taille stockée de l'allocation dans l'affichage du contenu est supérieure à la taille d'allocation dans le magasin de contenu. |
| 0x5200005c | STATUS_ACK_ERROR_STREAM_READ_ERROR | Erreur de lecture de flux ACK. | Une erreur renvoyée par l'ACK depuis le backend indiquant une erreur de lecture ou d'analyse du flux. Cela se produit généralement lorsque le backend ne parvient pas à récupérer le flux. Auto-restreaming peut généralement corriger cette erreur. |

| Code | Message | Description | Action recommandée |
|------------|--------------------------------------|--|--|
| 0x5200005d | STATUS_ACK_ERR_FRAGMENT_SIZE_REACHED | La taille maximale de fragment a été atteinte. | La taille maximale de fragment en octets est définie dans Quotas du SDK pour les producteurs . Cette erreur indique la présence d'images très volumineuses ou l'absence d'images clés pour créer des fragments de taille gérable. Vérifiez les paramètres de l'encodeur et vérifiez que les images clés sont produites correctement. Pour les flux à très haute densité, configurez l'encodeur pour produire des fragments de plus petites durées pour gérer la taille maximale. |

| Code | Message | Description | Action recommandée |
|------------|--|--|--|
| 0x5200005e | STATUS_AK_ERR_FRAGMENT_DURATION_REACHED | La durée maximale de fragment a été atteinte. | La durée maximale de fragment est définie dans Quotas du SDK pour les producteurs . Cette erreur indique une très faible fréquence d'images par seconde ou l'absence d'images clés pour créer des fragments de durée gérable. Vérifiez les paramètres de l'encodeur et vérifiez que les images clés sont produites correctement à intervalles réguliers. |
| 0x5200005f | STATUS_AK_ERR_CONNECTION_DURATION_REACHED | La durée maximale de connexion a été atteinte. | Kinesis Video Streams applique la durée maximale de connexion spécifiée dans le Quotas du SDK pour les producteurs . Le SDK Producer fait automatiquement pivoter le flux ou le jeton avant que le maximum ne soit atteint. Les clients utilisant le SDK ne devraient pas recevoir cette erreur. |
| 0x52000060 | STATUS_AK_ERR_FRAGMENT_TIMESTAMP_NOT_MONOTONIC | Les codes horaires n'augmentent pas de façon monotone. | Le SDK Producer applique les horodatages, de sorte que les clients utilisant le SDK ne devraient pas recevoir cette erreur. |

| Code | Message | Description | Action recommandée |
|------------|---|-------------------------------|--|
| 0x52000061 | STATUS_AC K_ERR_MUL TI_TRACK_MKV | Pistes multiples dans le MKV. | Le SDK Producer applique les flux à piste unique, de sorte que les clients utilisant le SDK ne devraient pas recevoir cette erreur. |
| 0x52000062 | STATUS_AC K_ERR_INV ALID_MKV_DATA | Données MKV non valides. | L'analyseur MKV du backend a rencontré une erreur lors de l'analyse du flux. Les clients utilisant le SDK peuvent rencontrer cette erreur si le flux est endommagé pendant la transition. Cela peut également se produire si les pressions de la zone tampon obligent le SDK à abandonner les châssis partiellement transmis. Dans ce dernier cas, nous vous recommandons de réduire le nombre d'images par seconde et la résolution, d'augmenter le taux de compression ou (s'il s'agit d'un réseau « surchargé ») d'augmenter la durée du stockage du contenu et de la mémoire tampon pour faire face aux pressions temporaires. |

| Code | Message | Description | Action recommandée |
|------------|---|--|---|
| 0x52000063 | STATUS_ACK_ERR_INVALID_PRODUCER_TIMESTAMP | Horodatage non valide du kit Producer. | Le service renvoie cette erreur ACK si l'horloge du producteur présente une dérive importante dans le futur. Higher-level Les SDK (par exemple, Java ou C++) utilisent une version de l'horloge système pour satisfaire le rappel horaire actuel du PIC. Vérifiez que l'horloge du système est correctement réglée. Les clients utilisant directement le PIC doivent vérifier que leurs fonctions de rappel renvoient l'horodatage correct. |
| 0x52000064 | STATUS_ACK_STREAM_NOT_ACTIVE | Flux inactif. | Un appel à une API de backend a été effectué pendant que le flux n'était pas dans un état « Actif ». Cela se produit lorsque le client crée le flux et continue immédiatement à lui transmettre (push) des images. Le kit SDK gère ce scénario via la machine d'état et le mécanisme de récupération. |

| Code | Message | Description | Action recommandée |
|------------|--|--|---|
| 0x52000065 | STATUS_AK_ERR_KMS_KEY_ACCESS_DENIED | AWS KMS erreur d'accès refusé. | Erreur renvoyée lorsque le compte n'a pas accès à la clé spécifiée. |
| 0x52000066 | STATUS_AK_ERR_KMS_KEY_DISABLED | AWS KMS la touche est désactivée. | La clé spécifiée a été désactivée. |
| 0x52000067 | STATUS_AK_ERR_KMS_KEY_VALIDATION_ERROR | AWS KMS erreur de validation des clés. | Erreur de validation générique. Pour plus d'informations, consultez la Référence des API AWS Key Management Service . |
| 0x52000068 | STATUS_AK_ERR_KMS_KEY_UNAVAILABLE | AWS KMS key indisponible. | La clé est indisponible. Pour plus d'informations, consultez la Référence des API AWS Key Management Service . |
| 0x52000069 | STATUS_AK_ERR_KMS_KEY_INVALID_USAGE | Utilisation non valide de la clé KMS. | Le n° AWS KMS key est pas configuré pour être utilisé dans ce contexte. Pour plus d'informations, consultez la Référence des API AWS Key Management Service . |
| 0x5200006a | STATUS_AK_ERR_KMS_KEY_INVALID_STATE | AWS KMS état non valide. | Pour plus d'informations, consultez la Référence des API AWS Key Management Service . |

| Code | Message | Description | Action recommandée |
|------------|--|---|--|
| 0x5200006b | STATUS_ACK_ERR_KMS_KEY_NOT_FOUND | Clé KMS introuvable. | La clé est introuvable. Pour plus d'informations, consultez la Référence des API AWS Key Management Service . |
| 0x5200006c | STATUS_ACK_ERR_STREAM_DELETED | Le flux a été ou est en cours de suppression. | Le flux est en cours de suppression par une autre application ou via la AWS Management Console. |
| 0x5200006d | STATUS_ACK_ERR_INTERNAL_ERROR | Erreur interne. | Erreur interne du service générique. |
| 0x5200006e | STATUS_ACK_ERR_FRAGMENT_ARCHIVAL_ERROR | Erreur d'archivage du fragment. | Erreur renvoyée lorsque le service ne parvient pas à indexer et conserver durablement le fragment. Bien que rare, elle peut se produire pour diverses raisons. Par défaut, le kit SDK tente d'envoyer à nouveau le fragment. |
| 0x5200006f | STATUS_ACK_ERR_UNKNOWN_ACK_ERROR | Erreur inconnue. | Le service a renvoyé une erreur inconnue. |
| 0x52000070 | STATUS_MISSING_ERR_ACK_ID | Informations ACK manquantes. | L'analyseur ACK a terminé l'analyse, mais les informations FragmentAck sont manquantes. |

| Code | Message | Description | Action recommandée |
|------------|--|---|--|
| 0x52000071 | STATUS_INVALID_ACK_SEGMENT_LEN | Longueur non valide de segment ACK. | Une chaîne de segment ACK d'une longueur non valide a été spécifiée pour l'analyseur ACK. Pour de plus amples informations, veuillez consulter Quotas du SDK pour les producteurs . |
| 0x52000074 | STATUS_MAXIMUM_FRAGMENT_METADATA_COUNT | Le nombre maximal d'éléments de métadonnées a été ajouté à un fragment. | Un flux vidéo Kinesis peut ajouter jusqu'à 10 éléments de métadonnées à un fragment, soit en ajoutant un élément non persistant à un fragment, soit en ajoutant un élément persistant à la file d'attente de métadonnées. Pour de plus amples informations, veuillez consulter Utilisation des métadonnées de streaming avec Kinesis Video Streams . |

| Code | Message | Description | Action recommandée |
|------------|---|---|---|
| 0x52000075 | STATUS_ACK_ERR_FRAGMENT_METADATA_LIMIT_REACHED | Une limite (nombre maximal de métadonnées, longueur de nom de métadonnées ou longueur de valeur de métadonnées) a été atteinte. | Le kit SDK Producer limite le nombre et la taille des éléments de métadonnées. Cette erreur ne se produit que si les limites du code du SDK du producteur sont modifiées. Pour de plus amples informations, veuillez consulter Utilisation des métadonnées de streaming avec Kinesis Video Streams . |
| 0x52000076 | STATUS_BLOCKING_PUT_INTERRUPTED_STREAM_TERMINATED | Non implémenté. | |
| 0x52000077 | STATUS_INVALID_METADATA_NAME | Le nom des métadonnées n'est pas valide. | Le nom des métadonnées ne peut pas commencer par la chaîne « AWS ». Si cette erreur se produit, l'élément de métadonnées n'est pas ajouté au fragment ou à la file d'attente de métadonnées. Pour de plus amples informations, veuillez consulter Utilisation des métadonnées de streaming avec Kinesis Video Streams . |

| Code | Message | Description | Action recommandée |
|------------|---|--|---|
| 0x52000078 | STATUS_END_OF_FRAGMENT_FRAME_INVALID_STATE | La fin d'une trame de fragment est à un état non valide. | La fin du fragment ne doit pas être envoyée dans un flux fragmenté sans image-clé. |
| 0x52000079 | STATUS_TRACK_INFO_MISSING | Il manque les informations de piste. | Le numéro de piste doit être supérieur à zéro et correspondre à l'identifiant de piste. |
| 0x5200007a | STATUS_MAXIMUM_TRACK_COUNT_EXCEEDED | Le nombre maximum de pistes est dépassé. | Vous pouvez avoir un maximum de trois titres par stream. |
| 0x5200007b | STATUS_OFFLINE_MODE_WITH_ZERO_RETENTION | La durée de conservation du mode de streaming hors ligne est définie sur zéro. | La durée de rétention du mode streaming hors ligne ne doit pas être fixée à zéro. |
| 0x5200007c | STATUS_ACK_ERROR_TRACK_NUMBER_MISMATCH | Le numéro de piste de l'erreur ACK ne correspond pas. | |
| 0x5200007d | STATUS_ACK_ERROR_FRAMES_MISSING_FOR_TRACK | Trames manquantes pour une piste. | |
| 0x5200007e | STATUS_ACK_ERROR_MORE_THAN_ALLOWED_TRACKS_FOUND | Le nombre maximal de pistes autorisé est dépassé. | |

| Code | Message | Description | Action recommandée |
|------------|--|--|--------------------|
| 0x5200007f | STATUS_UP LOAD_HAND LE_ABORTED | La poignée de chargement est abandonnée. | |
| 0x52000080 | STATUS_IN VALID_CER T_PATH_LENGTH | Longueur de chemin de certificat non valide. | |
| 0x52000081 | STATUS_DU PLICATE_T RACK_ID_FOUND | ID de piste en double trouvé. | |
| 0x52000082 | STATUS_IN VALID_CLI ENT_INFO_VERSION | | |
| 0x52000083 | STATUS_IN VALID_CLI ENT_ID_ST RING_LENGTH | | |
| 0x52000084 | STATUS_SE TTING_KEY _FRAME_FL AG_WHILE_ USING_EOFR | | |
| 0x52000085 | STATUS_MA X_FRAME_T IMESTAMP_ DELTA_BET WEEN_TRAC KS_EXCEEDED | | |

| Code | Message | Description | Action recommandée |
|------------|---|-------------|--------------------|
| 0x52000086 | STATUS_STREAM_SHUTTING_DOWN | | |
| 0x52000087 | STATUS_CLIENT_SHUTTING_DOWN | | |
| 0x52000088 | STATUS_PUT_MEDIA_LAST_PERSIST_ACK_NOT_RECEIVED | | |
| 0x52000089 | STATUS_NON_ALIGNED_HEAP_WITHIN_CONTENT_STORE_ALLOCATORS | | |
| 0x5200008a | STATUS_MULTIPLE_CONSECUTIVE_EOFR | | |
| 0x5200008b | STATUS_DUPLICATE_STREAM_EVENT_TYPE | | |
| 0x5200008c | STATUS_STREAM_NOT_STARTED | | |
| 0x5200008d | STATUS_INVALID_IMAGE_PREFIX_LENGTH | | |

| Code | Message | Description | Action recommandée |
|------------|--------------------------------------|-------------|--------------------|
| 0x5200008e | STATUS_INVALID_METADATA_KEY_LENGTH | | |
| 0x5200008f | STATUS_INVALID_METADATA_VALUE_LENGTH | | |

Codes d'erreur et d'état renvoyés par la bibliothèque de durées

Le tableau suivant contient les informations d'erreur et d'état renvoyées par les méthodes de la `Duration` bibliothèque.

| Code | Message |
|--------------------|------------------------|
| 0xFFFFFFFFFFFFFFFF | INVALID_DURATION_VALUE |

Codes d'erreur et d'état renvoyés par la bibliothèque commune

Le tableau suivant contient les informations d'erreur et d'état renvoyées par les méthodes de la `Common` bibliothèque.

Note

Les codes des informations relatives aux erreurs et aux états sont communs à de nombreuses API.

| Code | Code sans 0 en début | Message | Description |
|------------|----------------------|-----------------------------|--|
| 0x00000001 | 0x1 | STATUS_NULL_ARG | La valeur NULL a été transmise pour un argument obligatoire. |
| 0x00000002 | 0x2 | STATUS_INVALID_ARG | Une valeur non valide a été spécifiée pour un argument. |
| 0x00000003 | 0x3 | STATUS_INVALID_ARG_LEN | Une longueur d'argument non valide a été spécifiée. |
| 0x00000004 | 0x4 | STATUS_NOT_ENOUGH_MEMORY | Impossible d'allouer suffisamment de mémoire. |
| 0x00000005 | 0x5 | STATUS_BUFFER_TOO_SMALL | La taille de tampon spécifiée est trop petite. |
| 0x00000006 | 0x6 | STATUS_UNEXPECTED_EOF | Une fin de fichier inattendue a été atteinte. |
| 0x00000007 | 0x7 | STATUS_FORMAT_ERROR | Un format non valide a été rencontré. |
| 0x00000008 | 0 x 8 | STATUS_INVALID_HANDLE_ERROR | Valeur de handle non valide. |
| 0x00000009 | 0x9 | STATUS_OPEN_FILE_FAILED | Échec d'ouverture d'un fichier. |
| 0x0000000a | 0xa | STATUS_READ_FILE_FAILED | Impossible de lire à partir d'un fichier. |

| Code | Code sans 0 en début | Message | Description |
|------------|----------------------|-----------------------------|---|
| 0x0000000b | 0xb | STATUS_WRITE_TO_FILE_FAILED | Impossible d'écrire dans un fichier. |
| 0x0000000c | 0xc | STATUS_INTERNAL_ERROR | Erreur interne qui ne se produit normalement pas et qui peut indiquer un bogue au niveau de l'API du service ou du kit SDK. |
| 0x0000000d | 0xd | STATUS_INVALID_OPERATION | Une opération non valide a eu lieu ou l'opération n'est pas autorisée. |
| 0x0000000e | 0xe | STATUS_NOT_IMPLEMENTED | La fonctionnalité n'est pas mise en œuvre. |
| 0x0000000f | 0xf | STATUS_OPERATION_TIMED_OUT | L'opération a expiré. |
| 0x00000010 | 0 x 10 | STATUS_NOT_FOUND | Une ressource requise est introuvable. |
| 0x00000011 | 0 x 11 | STATUS_CREATE_THREAD_FAILED | Impossible de créer un fil de discussion. |

| Code | Code sans 0 en début | Message | Description |
|------------|----------------------|--|---|
| 0x00000012 | 0 x 12 | STATUS_TH READ_NOT_ ENOUGH_RE SOURCES | Des ressources insuffisantes pour créer un autre thread ou une limite du nombre de threads imposée par le système a été rencontrée. |
| 0x00000013 | 0x13 | STATUS_TH READ_INVA LID_ARG | Attributs de thread non valides spécifiés , ou un autre thread attend déjà de rejoindre ce thread. |
| 0x00000014 | 0x14 | STATUS_TH READ_PERM ISSIONS | Aucune autorisation pour définir la politique de planification et les paramètres spécifiés dans les attributs du thread. |
| 0x00000015 | 0 x 15 | STATUS_TH READ_DEAD LOCKED | Un blocage a été détecté ou le thread de jonction indique le thread appelant. |
| 0x00000016 | 0 x 16 | STATUS_TH READ_DOES _NOT_EXIST | Impossible de trouver le thread avec l'ID de thread spécifié. |
| 0x00000017 | 0x17 | STATUS_JO IN_THREAD _FAILED | Une erreur inconnue ou générique a été renvoyée lors de l'opération de jointure des threads. |

| Code | Code sans 0 en début | Message | Description |
|------------|----------------------|---|---|
| 0x00000018 | 0 x 18 | STATUS_WAIT_FAILED | Le temps d'attente maximal pour la variable conditionnelle a été dépassé. |
| 0x00000019 | 0x19 | STATUS_CANCEL_THREAD_FAILED | Une erreur inconnue ou générique a été renvoyée par l'opération d'annulation du thread. |
| 0x0000001a | 0 x 1 a | STATUS_THREAD_READ_IS_NOT_JOINABLE | L'opération de jointure de thread est demandée sur un thread non joignable. |
| 0x0000001b | 0x1b | STATUS_DETACH_THREAD_FAILED | Une erreur inconnue ou générique a été renvoyée lors de l'opération de détachement du thread. |
| 0x0000001c | 0 x 1 c | STATUS_THREAD_READ_ATTR_INIT_FAILED | Impossible d'initialiser l'objet des attributs du thread. |
| 0x0000001d | 0x1d | STATUS_THREAD_READ_ATTR_SET_STACK_SIZE_FAILED | Impossible de définir la taille de pile pour l'objet des attributs de thread. |

| Code | Code sans 0 en début | Message | Description |
|------------|----------------------|--------------------------------------|---|
| 0x0000001e | 0x1e | STATUS_MEMORY_NOT_FREED | Utilisé uniquement dans les tests. Indique que toute la mémoire demandée n'a pas été libérée. |
| 0x0000001f | 0x1f | STATUS_INVALID_THREAD_PARAMS_VERSION | Version de structure ThreadParams « » non valide. Spécifiez la version actuelle correcte de la structure. |

Codes d'erreur et d'état renvoyés par la bibliothèque heap

Le tableau suivant contient les informations d'erreur et d'état renvoyées par les méthodes de la Heap bibliothèque.

| Code | Message | Description |
|------------|-----------------------------|--|
| 0x10000001 | STATUS_HEAP_FLAGS_ERROR | Une combinaison non valide d'indicateurs a été spécifiée. |
| 0x10000002 | STATUS_HEAP_NOT_INITIALIZED | Une opération a été tentée avant l'initialisation du segment de mémoire. |
| 0x10000003 | STATUS_HEAP_CORRUPTED | Le segment de mémoire a été corrompu ou la bande de garde (en mode de débogage) a été remplacée. Un dépassement de mémoire tampon dans le code client peut conduire à la corruption d'un segment de mémoire. |

| Code | Message | Description |
|------------|---|--|
| 0x10000004 | STATUS_HEAP_VRAM_L IB_MISSING | L'utilisateur VRAM (RAM vidéo) ou la bibliothèque en mode noyau ne peuvent pas être chargés ou sont absents. Vérifiez si la plateforme sous-jacente prend en charge les allocations de VRAM. |
| 0x10000005 | STATUS_HEAP_VRAM_L IB_REOPEN | Impossible d'ouvrir la bibliothèque VRAM. |
| 0x10000006 | STATUS_HEAP_VRAM_I INIT_FUNC_SYMBOL | Impossible de charger l'exportation de la fonction INIT. |
| 0x10000007 | STATUS_HEAP_VRAM_A LLOC_FUNC_SYMBOL | Impossible de charger l'exportation de la fonction ALLOC. |
| 0x10000008 | STATUS_HEAP_VRAM_F REE_FUNC_SYMBOL | Impossible de charger l'exportation de la fonction FREE. |
| 0x10000009 | STATUS_HEAP_VRAM_L OCK_FUNC_SYMBOL | Impossible de charger l'exportation de la fonction LOCK. |
| 0x1000000a | STATUS_HEAP_VRAM_U NLOCK_FUNC_SYMBOL | Impossible de charger l'exportation de la fonction UNLOCK. |
| 0x1000000b | STATUS_HEAP_VRAM_U NINIT_FUNC_SYMBOL | Impossible de charger l'exportation de la fonction UNINIT. |

| Code | Message | Description |
|------------|-------------------------------------|---|
| 0x1000000c | STATUS_HEAP_VRAM_GETMAX_FUNC_SYMBOL | Impossible de charger l'exportation de la fonction GETMAX. |
| 0x1000000d | STATUS_HEAP_DIRECT_MEM_INIT | Impossible d'initialiser le pool de segments de mémoire principal dans le segment de mémoire hybride. |
| 0x1000000e | STATUS_HEAP_VRAM_INIT_FAILED | Échec de l'initialisation dynamique de VRAM. |
| 0x1000000f | STATUS_HEAP_LIBRARY_FREE_FAILED | Impossible de désallouer et de libérer la bibliothèque VRAM. |
| 0x10000010 | STATUS_HEAP_VRAM_ALLOC_FAILED | Échec de l'allocation de VRAM. |
| 0x10000011 | STATUS_HEAP_VRAM_FREE_FAILED | Échec de la libération de VRAM. |
| 0x10000012 | STATUS_HEAP_VRAM_MAP_FAILED | Échec du mappage de VRAM. |
| 0x10000013 | STATUS_HEAP_VRAM_UNMAP_FAILED | Échec de l'annulation du mappage de VRAM. |
| 0x10000014 | STATUS_HEAP_VRAM_UNINIT_FAILED | Échec de la désinitialisation de VRAM. |
| 0x10000015 | STATUS_INVALID_ALLOCATION_SIZE | |
| 0x10000016 | STATUS_HEAP_REALLOC_ERROR | |

| Code | Message | Description |
|------------|--|-------------|
| 0x10000017 | STATUS_HEAP_FILE_H EAP_FILE_CORRUPT | |

Codes d'erreur et d'état renvoyés par la bibliothèque MKVGen

Le tableau suivant contient les informations d'erreur et d'état renvoyées par les méthodes de la MKVGen bibliothèque.

| Code | Message | Description/Action recommandée |
|------------|------------------------------------|---|
| 0x32000001 | STATUS_MKV_INVALID_FRAME_DATA | Membres non valides de la structure de données Frame. Assurez-vous que la durée, la taille et les données de trame sont valides et respectent les limites spécifiées dans Quotas du SDK pour les producteurs . |
| 0x32000002 | STATUS_MKV_INVALID_FRAME_TIMESTAMP | Horodatage d'image non valide. Les valeurs calculées de PTS (horodatage de présentation) et DTS (horodatage de décodage) sont supérieures ou égales à l'horodatage de l'image de début du fragment. Ceci indique un redémarrage potentiel de pipeline média ou un problème de stabilité d'encodeur. Pour plus d'informations sur le dépannage, consultez Erreur : « Failed to submit frame to |

| Code | Message | Description/Action recommandée |
|------------|--|---|
| | | Kinesis Video client » (Échec d'envoi d'image au client Kinesis Video) |
| 0x32000003 | STATUS_MKV_INVALID_CLUSTER_DURATION | Une durée de fragment non valide a été spécifiée. Pour de plus amples informations, veuillez consulter Quotas du SDK pour les producteurs . |
| 0x32000004 | STATUS_MKV_INVALID_CONTENT_TYPE_LENGTH | Longueur non valide de chaîne de type de contenu. Pour de plus amples informations, veuillez consulter Quotas du SDK pour les producteurs . |
| 0x32000005 | STATUS_MKV_NUMBER_TOO_BIG | Une tentative a eu lieu d'encoder un nombre trop grand pour être représenté dans le format EBML (Extensible Binary Meta Language). Cela ne doit pas être exposé aux clients du kit SDK. |
| 0x32000006 | STATUS_MKV_INVALID_CODEC_ID_LENGTH | Longueur non valide de chaîne d'ID de codec. Pour de plus amples informations, veuillez consulter Quotas du SDK pour les producteurs . |

| Code | Message | Description/Action recommandée |
|------------|---|---|
| 0x32000007 | STATUS_MKV_INVALID_TRACK_NAME_LENGTH | Longueur non valide de chaîne de nom de piste. Pour de plus amples informations, veuillez consulter Quotas du SDK pour les producteurs . |
| 0x32000008 | STATUS_MKV_INVALID_CODEC_PRIVATE_LENGTH | Longueur non valide des données privées codec. Pour de plus amples informations, veuillez consulter Quotas du SDK pour les producteurs . |
| 0x32000009 | STATUS_MKV_CODEC_PRIVATE_NULL | Les données privées du codec (CPD) sont NULL, alors que la taille du CPD est supérieure à zéro. |
| 0x3200000a | STATUS_MKV_INVALID_TIMECODE_SCALE | Valeur d'échelle de code horaire non valide. Pour de plus amples informations, veuillez consulter Quotas du SDK pour les producteurs . |
| 0x3200000b | STATUS_MKV_MAX_FRAME_TIMECODE | Le code horaire d'image est supérieur au maximum. Pour de plus amples informations, veuillez consulter Quotas du SDK pour les producteurs . |

| Code | Message | Description/Action recommandée |
|------------|---------------------------------|--|
| 0x3200000c | STATUS_MKV_LARGE_FRAME_TIMECODE | <p>Le code horaire d'image maximal a été atteint. Le format MKV utilise 16 bits signés pour représenter le code horaire relatif de l'image au début du cluster. L'erreur est générée si le code temporel de l'image ne peut pas être représenté. Cette erreur indique une mauvaise sélection de l'échelle du code horaire ou une durée de cluster trop longue, de sorte que la représentation du code horaire de l'image sature l'espace des 16 bits signés.</p> |

| Code | Message | Description/Action recommandée |
|------------|--|---|
| 0x3200000d | STATUS_MKV_INVALID _ANNEXB_NALU_IN_FR AME_DATA | Un code de Annex-B démarrage non valide a été détecté. Par exemple, l'indicateur Annex-B d'adaptation a été spécifié et le code rencontre une séquence de départ non valide de plus de trois zéros. Un Annex-B format valide doit comporter une séquence de « prévention de l'émulation » pour échapper à une séquence de trois zéros ou plus dans le flux d'octets. Pour plus d'informations, consultez la norme MPEG. Pour plus d'informations sur cette erreur sur Android, consultez 0x3200000d STATUS_MKV_INVALID _ANNEXB_NALU_IN_FR AME_DATA (erreur sur Android) . |
| 0x3200000e | STATUS_MKV_INVALID _AVCC_NALU_IN_FRAM E_DATA | Packaging AVCC NALU non valide lorsque l'indicateur AVCC d'adaptation est spécifié. Vérifiez que le flux d'octets est au format AVCC valide. Pour plus d'informations, consultez la norme MPEG. |

| Code | Message | Description/Action recommandée |
|------------|---|--|
| 0x3200000f | STATUS_MKV_BOTH_ANNEXB_AND_AVCC_SPECIFIED | L'adaptation de l'AVCC et du Annex-B NALU a été spécifiée . N'en spécifiez qu'un ou n'en spécifiez aucun. |
| 0x32000010 | STATUS_MKV_INVALID_ANNEXB_NALU_IN_CPD | Annex-B Format de CPD non valide lorsque l' Annex-B indicateur d'adaptation est spécifié. Vérifiez que le Annex-B format du CPD est valide. Si ce n'est pas le cas, supprimez l'indicateur Annex-B d'adaptation CPD. |
| 0x32000011 | STATUS_MKV_PTS_DTS_ARE_NOT_SAME | Kinesis Video Streams impose que le PTS (horodatage de présentation) et le DTS (horodatage de décodage) soient identiques pour les images de début du fragment. Il s'agit des images clés qui démarrent le fragment. |
| 0x32000012 | STATUS_MKV_INVALID_H264_H265_CPD | Impossible d'analyser les données privées du H264/H265 codec. |
| 0x32000013 | STATUS_MKV_INVALID_H264_H265_SPS_WIDTH | Impossible d'extraire la largeur des données privées codec. |
| 0x32000014 | STATUS_MKV_INVALID_H264_H265_SPS_HEIGHT | Impossible d'extraire la hauteur des données privées codec. |

| Code | Message | Description/Action recommandée |
|------------|---|--|
| 0x32000015 | STATUS_MKV_INVALID_H264_H265_SPS_NALU | NALU H264/H265 SPS non valide. |
| 0x32000016 | STATUS_MKV_INVALID_BIH_CPD | Format non valide d'en-tête d'informations de bitmap dans les données privées codec. |
| 0x32000017 | STATUS_MKV_INVALID_HEVC_NALU_COUNT | Nombre d'unités NALU (Network Abstraction Layer units) HEVC (High Efficiency Video Coding) non valide. |
| 0x32000018 | STATUS_MKV_INVALID_HEVC_FORMAT | Format HEVC non valide. |
| 0x32000019 | STATUS_MKV_HEVC_SPS_NALU_MISSING | Unités NALU HEVC manquantes dans l'ensemble de paramètres de séquence (SPS). |
| 0x3200001a | STATUS_MKV_INVALID_HEVC_SPS_NALU_SIZE | Taille NALU SPS HEVC non valide. |
| 0x3200001b | STATUS_MKV_INVALID_HEVC_SPS_CHROMA_FORMAT_IDC | IDC de format Chroma non valide. |
| 0x3200001c | STATUS_MKV_INVALID_HEVC_SPS_RESERVED | SPS réservé HEVC non valide. |

| Code | Message | Description/Action recommandée |
|------------|---|--|
| 0x3200001d | STATUS_MKV_MIN_ANNEX_B_CPD_SIZE | Taille minimale de la valeur bêta privée du AnnexBb codec. Pour H264, cette valeur doit être supérieure ou égale à 11. Pour H265, cette valeur doit être supérieure ou égale à 15. |
| 0x3200001e | STATUS_MKV_ANNEXB_CPD_MISSING_NALUS | Données privées du codec manquantes dans Annex-B NALus. |
| 0x3200001f | STATUS_MKV_INVALID_ANNEXB_CPD_NALUS | Bêta privée du codec non valide dans Annex-B NALus. |
| 0x32000020 | STATUS_MKV_INVALID_TAG_NAME_LENGTH | La longueur du nom de balise n'est pas valide. La valeur valide est supérieure à zéro et inférieure à 128. |
| 0x32000021 | STATUS_MKV_INVALID_TAG_VALUE_LENGTH | La longueur de la valeur de balise n'est pas valide. La valeur valide est supérieure à zéro et inférieure à 256. |
| 0x32000022 | STATUS_MKV_INVALID_GENERATOR_STATE_TAGS | Balises d'état de générateur non valides. |
| 0x32000023 | STATUS_MKV_INVALID_AAC_CPD_SAMPLING_FREQUENCY_INDEX | Index de fréquence d'échantillonnage des données privées codec AAC non valide. |

| Code | Message | Description/Action recommandée |
|------------|---|---|
| 0x32000024 | STATUS_MKV_INVALID_AAC_CPD_CHANNEL_CONFIG | Configuration de canal de données privées codec AAC non valide. |
| 0x32000025 | STATUS_MKV_INVALID_AAC_CPD | Données privées codec AAC non valides |
| 0x32000026 | STATUS_MKV_TRACK_INFO_NOT_FOUND | Informations de piste introuvables. |
| 0x32000027 | STATUS_MKV_INVALID_SEGMENT_UUID | UUID de segment non valide. |
| 0x32000028 | STATUS_MKV_INVALID_TRACK_UID | UID de piste non valide. |
| 0x32000029 | STATUS_MKV_INVALID_CLIENT_ID_LENGTH | |
| 0x3200002a | STATUS_MKV_INVALID_AMS_ACM_CPD | |
| 0x3200002b | STATUS_MKV_MISSING_SPS_FROM_H264_CPD | |
| 0x3200002c | STATUS_MKV_MISSING_PPS_FROM_H264_CPD | |
| 0x3200002d | STATUS_MKV_INVALID_PARENT_TYPE | |

Codes d'erreur et d'état renvoyés par la bibliothèque Trace

Le tableau suivant contient les informations d'erreur et d'état renvoyées par les méthodes de la Trace bibliothèque.

| Code | Message |
|------------|----------------------------|
| 0x10100001 | STATUS_MIN_PROFILER_BUFFER |

Codes d'erreur et d'état renvoyés par la bibliothèque Utils

Le tableau suivant contient les informations d'erreur et d'état renvoyées par les méthodes de la `Utils` bibliothèque.

| Code | Message |
|------------|-----------------------------------|
| 0x40000001 | STATUS_INVALID_BASE64_ENCODE |
| 0x40000002 | STATUS_INVALID_BASE |
| 0x40000003 | STATUS_INVALID_DIGIT |
| 0x40000004 | STATUS_INT_OVERFLOW |
| 0x40000005 | STATUS_EMPTY_STRING |
| 0x40000006 | STATUS_DIRECTORY_OPEN_FAILED |
| 0x40000007 | STATUS_PATH_TOO_LONG |
| 0x40000008 | STATUS_UNKNOWN_DIR_ENTRY_TYPE |
| 0x40000009 | STATUS_REMOVE_DIRECTORY_FAILED |
| 0x4000000a | STATUS_REMOVE_FILE_FAILED |
| 0x4000000b | STATUS_REMOVE_LINK_FAILED |
| 0x4000000c | STATUS_DIRECTORY_ACCESS_DENIED |
| 0x4000000d | STATUS_DIRECTORY_MISSING_PATH |
| 0x4000000e | STATUS_DIRECTORY_ENTRY_STAT_ERROR |

| Code | Message |
|------------|--|
| 0x4000000f | STATUS_STRFTIME_FALIED |
| 0x40000010 | STATUS_MAX_TIMESTAMP_FORMAT_STR_LEN_EXCEEDED |
| 0x40000011 | STATUS_UTIL_MAX_TAG_COUNT |
| 0x40000012 | STATUS_UTIL_INVALID_TAG_VERSION |
| 0x40000013 | STATUS_UTIL_TAGS_COUNT_NON_ZERO_TAGS_NULL |
| 0x40000014 | STATUS_UTIL_INVALID_TAG_NAME_LEN |
| 0x40000015 | STATUS_UTIL_INVALID_TAG_VALUE_LEN |
| 0x4000002a | STATUS_EXPONENTIAL_BACKOFF_INVALID_STATE |
| 0x4000002b | STATUS_EXPONENTIAL_BACKOFF_RETRIES_EXHAUSTED |
| 0x4000002c | STATUS_THREADPOOL_MAX_COUNT |
| 0x4000002d | STATUS_THREADPOOL_INTERNAL_ERROR |
| 0x40100001 | STATUS_HASH_KEY_NOT_PRESENT |
| 0x40100002 | STATUS_HASH_KEY_ALREADY_PRESENT |
| 0x40100003 | STATUS_HASH_ENTRY_ITERATION_ABORT |
| 0x41000001 | STATUS_BIT_READER_OUT_OF_RANGE |
| 0x41000002 | STATUS_BIT_READER_INVALID_SIZE |

| Code | Message |
|------------|--|
| 0x41100001 | STATUS_TIMER_QUEUE_STOP_SCHEDULING |
| 0x41100002 | STATUS_INVALID_TIMER_COUNT_VALUE |
| 0x41100003 | STATUS_INVALID_TIMER_PERIOD_VALUE |
| 0x41100004 | STATUS_MAX_TIMER_COUNT_REACHED |
| 0x41100005 | STATUS_TIMER_QUEUE_SHUTDOWN |
| 0x41200001 | STATUS_SEMAPHORE_OPERATION_AFTER_SHUTDOWN |
| 0x41200002 | STATUS_SEMAPHORE_ACQUIRE_WHEN_LOCKED |
| 0x41300001 | STATUS_FILE_LOGGER_INDEX_FILE_INVALID_SIZE |

Codes d'erreur et d'état renvoyés par la bibliothèque View

Le tableau suivant contient les informations d'erreur et d'état renvoyées par les méthodes de la View bibliothèque.

| Code | Message | Description |
|------------|--------------------------------------|--|
| 0x30000001 | STATUS_MIN_CONTENT_VIEW_ITEMS | Un nombre non valide d'éléments d'affichage de contenu a été spécifié. Pour de plus amples informations, veuillez consulter Quotas du SDK pour les producteurs . |
| 0x30000002 | STATUS_INVALID_CONTENT_VIEW_DURATION | Une durée non valide d'affichage de contenu a été spécifiée. |

| Code | Message | Description |
|------------|-----------------------------------|---|
| | | Pour de plus amples informations, veuillez consulter Quotas du SDK pour les producteurs . |
| 0x30000003 | STATUS_CONTENT_VIEW_NO_MORE_ITEMS | Une tentative a été effectuée pour dépasser la position de tête. |
| 0x30000004 | STATUS_CONTENT_VIEW_INVALID_INDEX | Un index non valide est spécifié. |

| Code | Message | Description |
|------------|---------------------------------------|--|
| 0x30000005 | STATUS_CONTENT_VIEW_INVALID_TIMESTAMP | <p>Il y a eu un horodatage non valide ou un chevauchement d'horodatages. L'horodatage du décodage d'image doit être supérieur ou égal à l'horodatage de l'image précédente, plus la durée de l'image précédente : $DTS(n) \geq DTS(n-1) + Duration(n-1)$. Cette erreur indique souvent une « instabilité » de l'encodeur. L'encodeur produit un pic d'images encodées et leurs horodatages sont plus petits que les durées intra-image. Ou le flux est configuré pour utiliser les horodatages du kit SDK et les images sont envoyées plus rapidement que les durées des images. Pour atténuer la « instabilité » de l'encodeur, vous pouvez ajuster la durée de l'image dans la structure <code>Frame</code> lorsque vous appelez <code>putFrame()</code> ou <code>putKinesisVideoFrame()</code>. Certains flux requièrent un contrôle de temporisation plus précis pour la détection des erreurs.</p> |
| 0x30000006 | STATUS_INVALID_CONTENT_VIEW_LENGTH | <p>Une longueur non valide des données d'éléments d'affichage de contenu a été spécifiée.</p> |

Erreurs et codes d'état renvoyés par les PutFrame rappels - bibliothèque C Producer

La section suivante contient les informations d'erreur et d'état renvoyées par les rappels pour l'PutFrameopération dans la bibliothèque du producteur C.

| Code | Message | Description | Action recommandée |
|------------|---|--|---|
| 0x15000001 | STATUS_STOP_CALLBACK_CHAIN | La chaîne de rappel a été arrêtée. | |
| 0x15000002 | STATUS_MAXIMUM_CALLBACK_CHAIN | La chaîne de rappel maximum a été atteinte. | |
| 0x15000003 | STATUS_INVALID_PLATFORM_CALLBACKS_VERSION | La version de la structure PlatformCallbacks est non valide. | Spécifiez la version actuelle correcte de la structure. |
| 0x15000004 | STATUS_INVALID_PRODUCER_CALLBACKS_VERSION | La version de la structure ProducerCallbacks est non valide. | Spécifiez la version actuelle correcte de la structure. |
| 0x15000005 | STATUS_INVALID_STREAM_CALLBACKS_VERSION | La version de la structure StreamCallbacks est non valide. | Spécifiez la version actuelle correcte de la structure. |
| 0x15000006 | STATUS_INVALID_AUTH_CALLBACKS | La version de la structure AuthCallbacks est non valide. | Spécifiez la version actuelle correcte de la structure. |

| Code | Message | Description | Action recommandée |
|------------|--|--|---|
| | H_CALLBACKS_VERSION | | |
| 0x15000007 | STATUS_INVALID_API_CALLBACKS_VERSION | La version de la structure ApiCallbacks est non valide. | Spécifiez la version actuelle correcte de la structure. |
| 0x15000008 | STATUS_INVALID_AWS_CREDENTIALS_VERSION | La version de la structure AwsCredentials est non valide. | Spécifiez la version actuelle correcte de la structure. |
| 0x15000009 | STATUS_MAX_REQUEST_HEADER_COUNT | Le nombre maximal d'en-têtes de demande a été atteint. | |
| 0x1500000a | STATUS_MAX_REQUEST_HEADER_NAME_LEN | La longueur maximale du nom d'en-tête de demande a été atteint. | |
| 0x1500000b | STATUS_MAX_REQUEST_HEADER_VALUE_LEN | La longueur maximale de la valeur d'en-tête de demande a été atteinte. | |
| 0x1500000c | STATUS_INVALID_API_RETURN_JSON | Retour JSON non valide pour un appel d'API. | |

| Code | Message | Description | Action recommandée |
|------------|---|---|---|
| 0x1500000d | STATUS_CURL_INIT_FAILED | L'initialisation de Curl a échoué. | |
| 0x1500000e | STATUS_CURL_LIBRARY_INIT_FAILED | L'initialisation de la bibliothèque Curl a échoué. | |
| 0x1500000f | STATUS_INVALID_DESCRIBE_STREAM_RETURN_JSON | JSON de retour non valide pour DescribeStream. | |
| 0x15000010 | STATUS_HMAC_GENERATION_ERROR | Erreur de génération HMAC. | |
| 0x15000011 | STATUS_IOT_FAILED | L'autorisation IoT a échoué. | |
| 0x15000012 | STATUS_MAX_ALIAS_LENGTH_EXCEEDED | La longueur maximale d'alias de rôle a été atteinte. | Spécifiez une longueur d'alias plus courte. |
| 0x15000013 | STATUS_MAX_AGENT_NAME_POSTFIX_LENGTH_EXCEEDED | La longueur maximale de postfix du nom d'agent a été atteinte. | |
| 0x15000014 | STATUS_MAX_CUSTOM_USER_AGENT_LENGTH_EXCEEDED | La longueur maximale de l'agent utilisateur du client a été atteinte. | |

| Code | Message | Description | Action recommandée |
|------------|---|--|--|
| 0x15000015 | STATUS_INVALID_USER_AGENT_LENGTH | Longueur d'agent utilisateur non valide. | |
| 0x15000016 | STATUS_INVALID_ENDPOINT_TERMINATION_PERIOD | Période de mise en cache de point de terminaison non valide. | Spécifiez une période de mise en cache inférieure à 24 heures. |
| 0x15000017 | STATUS_IOT_EXPIRATION_OCCURRED_IN_PAST | L'horodatage d'expiration de l'IoT se produit dans le passé. | |
| 0x15000018 | STATUS_IOT_EXPIRATION_PARSING_FAILED | L'analyse de l'expiration de l'IoT a échoué. | |
| 0x15000019 | STATUS_DUPLICATE_PRODUCER_CALLBACK_FUNCTION | | |
| 0x1500001a | STATUS_DUPLICATE_STREAM_CALLBACK_FREE_FUNC | | |
| 0x1500001b | STATUS_DUPLICATE_AUTH_CALLBACK_FREE_FUNC | | |

| Code | Message | Description | Action recommandée |
|------------|---|-------------|--------------------|
| 0x1500001c | STATUS_DUPLICATE_API_CALLBACK_FREE_FUNC | | |
| 0x1500001d | STATUS_FILE_LOGGER_INDEX_FILE_TOO_LARGE | | |
| 0x1500001e | STATUS_MAXIMUM_IOT_THING_NAME_LENGTH | | |
| 0x1500001f | STATUS_IOT_CREATE_CONTENTS_FAILED | | |
| 0x15000020 | STATUS_INVALID_CERT_PATH | | |
| 0x15000022 | STATUS_FILE_CREDENTIAL_PROVIDER_OPEN_FILE_FAILED | | |
| 0x15000023 | STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_LENGTH | | |

| Code | Message | Description | Action recommandée |
|------------|---|-------------|--------------------|
| 0x15000024 | STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_FORMAT | | |
| 0x15000026 | STATUS_STREAM_BEING_SHUTDOWN | | |
| 0x15000027 | STATUS_CLIENT_BEING_SHUTDOWN | | |
| 0x15000028 | STATUS_CONTINUOUS_RETRY_RETRY_FAILED | | |

| Code | Message | Description | Action recommandée |
|------------|------------------------------------|--|--|
| 0x16000001 | STATUS_CURL_PERFORM_FAILED | CURL a renvoyé un code d'échec. | <p>Consultez les journaux pour plus d'informations. Une erreur CURL courante est « Impossible de résoudre le nom d'hôte. », vérifiez la connectivité Internet de l'appareil.</p> <p>Une autre erreur courante est le code d'erreur 403. Cela indique que les certificats IoT ne sont pas créés ou spécifiés correctement. Vérifiez que les chemins d'accès aux certificats IoT sont correctement définis et que les autorisations sont correctement définies. Pour plus d'informations, consultez the section called “Contrôle de l'accès aux ressources Kinesis Video Streams à l'aide de AWS IoT”.</p> |
| 0x16000002 | STATUS_IOT_INVALID_RESPONSE_LENGTH | J'ai reçu une réponse de longueur nulle lors de la récupération des informations d'identification IoT. | Consultez le tableau AWS de bord de santé et réessayez ultérieurement. |

| Code | Message | Description | Action recommandée |
|------------|---|---|--|
| 0x16000003 | STATUS_IOT_NULL_AWS_CREDS | Le JSON renvoyé par le point de terminaison des informations d'identification IoT ne contenait pas l'objet des informations d'identification. | Consultez l'élément « message » dans le JSON pour plus d'informations. |
| 0x16000004 | STATUS_IOT_INVALID_URI_LEN | L'URL transmise à la fonction de récupération des informations d'identification IoT n'a pas une longueur comprise entre 1 et 10 000. | Vérifiez l'URL transmise à cette fonction. |
| 0x16000005 | STATUS_TIMESTAMP_STRING_UNRECOGNIZED_FORMAT | L'élément « expiration » du JSON provenant de la récupération des informations d'identification IoT n'est pas au format :YYYY-MM-DDTHH:mm:ssZ . | Consultez le tableau AWS de bord de santé et réessayez ultérieurement. |

Référence concernant les indicateurs d'adaptation NAL (Network Abstraction Layer)

Cette section contient des informations sur les indicateurs disponibles pour l'énumération `StreamInfo.NalAdaptationFlags`.

Le [flux élémentaire](#) d'une application peut être au format AVCC Annex-B ou au format AVCC :

- Le Annex-B format délimite les [NALU \(unités de couche d'abstraction réseau\)](#) par deux octets de zéros, suivis d'un ou trois octets de zéros, suivis du chiffre 1 (appelé code de démarrage, par exemple, 00000001).
- Le format AVCC encapsule également les NALU, mais chaque unité NALU est précédée d'une valeur qui indique la taille de l'unité NALU (généralement quatre octets).

De nombreux encodeurs produisent le format Annex-B bitstream. Certains processeurs de flux binaire de niveau supérieur (tels qu'un moteur de lecture ou le lecteur [Media Source Extensions \(MSE\) intégré au AWS Management Console](#)) utilisent le format AVCC pour leurs images.

Les données privées du codec (CPD), qui sont SPS/PPS (ensemble de Set/Picture paramètres de séquence) pour le H.264 codec, peuvent également être au Annex-B format AVCC. Cependant, pour les CPD, les formats sont différents de ceux décrits précédemment.

Les indicateurs indiquent au SDK d'adapter les NALU à l'AVCC ou Annex-B pour les données de trame et le CPD comme suit :

| Indicateur | Adaptation |
|--|--|
| NAL_ADAPTATION_FLAG_NONE | Aucune adaptation. |
| NAL_ADAPTATION_ANNEXB_NALS | Adaptez les Annex-B NALU aux NALU AVCC. |
| NAL_ADAPTATION_AVCC_C_NALS | Adaptez les NALU AVCC aux NALU. Annex-B |
| NAL_ADAPTATION_ANNEXB_CPD_NALS | Adaptez les Annex-B NALU pour les données privées du codec au format AVCC NALU. |
| NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS | Adaptez les Annex-B NALU au codec et transformez les données privées en NALU au format AVCC. |

Pour plus d'informations sur les types d'unités NALU, consultez Section 1.3: Network Abstraction Layer Unit Types dans [RFC 3984](#).

Structures du SDK pour les producteurs

Cette section contient des informations sur les structures que vous pouvez utiliser pour fournir des données à l'objet Kinesis Video Streams Producer.

Rubriques

- [DeviceInfo/DefaultDeviceInfoProvider](#)
- [StorageInfo](#)

DeviceInfo/DefaultDeviceInfoProvider

Les DefaultDeviceInfoProviderobjets DeviceInfoet contrôlent le comportement de l'objet Kinesis Video Streams Producer.

Champs réservés aux membres

- version — Valeur entière utilisée pour s'assurer que la version correcte de la structure est utilisée avec la version actuelle de la base de code. La version actuelle est spécifiée à l'aide de la macro `DEVICE_INFO_CURRENT_VERSION`.
- name — Le nom lisible par l'homme du périphérique.
- tagCount/tags— Non utilisé actuellement.
- StreamCount — Le nombre maximum de flux que l'appareil peut gérer. Il pré-alloue initialement le stockage des pointeurs vers les objets de flux, mais les objets de flux réels sont créés ultérieurement. La valeur par défaut est de 16 flux, mais vous pouvez la modifier dans le fichier `DefaultDeviceInfoProvider.cpp`.
- storageInfo : Un objet décrivant la configuration du stockage principal. Pour de plus amples informations, veuillez consulter [StorageInfo](#).

StorageInfo

Spécifie la configuration du stockage principal pour Kinesis Video Streams.

L'implémentation par défaut est basée sur une implémentation de tas rapide à fragmentation faible, optimisée pour le streaming. Elle utilise l'allocateur MEMALLOC, qui peut être remplacé sur une plateforme donnée. Certaines plates-formes ont une allocation de mémoire virtuelle sans renforcement d'allocation par des pages physiques. À mesure que la mémoire est utilisée, les pages virtuelles sont renforcées par les pages physiques. Cela entraîne une sollicitation moindre de la mémoire sur l'ensemble du système lorsque le stockage est sous-utilisé.

Calculez le volume de stockage par défaut selon la formule suivante. Le paramètre `DefragmentationFactor` doit être défini sur 1.2 (20 pour cent).

```
Size = NumberOfStreams * AverageFrameSize * FramesPerSecond * BufferDurationInSeconds *
      DefragmentationFactor
```

Dans l'exemple suivant, un appareil dispose de flux audio et vidéo. Le flux audio possède 512 échantillons par seconde avec un échantillon moyen de 100 octets. Le flux vidéo possède 25 images par seconde avec une moyenne de 10 000 octets. Chaque flux comporte 3 minutes de durée de mise en tampon.

```
Size = (512 * 100 * (3 * 60) + 25 * 10000 * (3 * 60)) * 1.2 = (9216000 + 45000000) *
1.2 = 65059200 = ~ 66MB.
```

Si le périphérique dispose de plus de mémoire disponible, nous vous recommandons d'ajouter de la mémoire au stockage afin d'éviter une fragmentation importante.

Vérifiez que la taille de stockage est suffisante pour accueillir les tampons complets pour tous les flux présentant une complexité de codage élevée (lorsque la taille d'image est plus grande en raison de mouvements élevés) ou lorsque la bande passante est faible. Si le producteur atteint une pression de mémoire, il émet des rappels de pression de dépassement de capacité de stockage (`StorageOverflowPressureFunc`). Toutefois, lorsqu'aucune mémoire n'est disponible dans le magasin de contenu, l'image envoyée dans Kinesis Video Streams est supprimée avec une erreur `STATUS_STORE_OUT_OF_MEMORY = 0x5200002e` (`STATUS_STORE_OUT_OF_MEMORY`). Pour de plus amples informations, veuillez consulter [Codes d'erreur et d'état renvoyés par la bibliothèque cliente](#). Cela peut également se produire si les accusés de réception (ACK) d'application ne sont pas disponibles ou si les ACK de conservation sont retardés. Dans ce cas, les tampons atteignent leur capacité de « durée de mémoire tampon » avant que les images précédentes ne commencent à disparaître.

Champs réservés aux membres

- `version` — Valeur entière utilisée pour s'assurer que la version correcte de la structure est utilisée avec la version actuelle de la base de code.
- `StorageType` : `DEVICE_STORAGE_TYPE` énumération qui indique le support sous-jacent et l'implémentation du stockage. Actuellement, la seule valeur prise en charge est `DEVICE_STORAGE_TYPE_IN_MEM`. `DEVICE_STORAGE_TYPE_HYBRID_FILE` sera prise en charge dans une prochaine implémentation, ce qui indique que le stockage revient au magasin de contenu basé sur les fichiers.
- `StorageSize` — Taille de stockage en octets à préallouer. L'allocation minimum est 10 Mo et l'allocation maximum est 10 Go. (Cela évoluera avec la prochaine implémentation du magasin de contenu basé sur les fichiers.)
- `SpillRatio` — Valeur entière qui représente le pourcentage de stockage à allouer à partir du type de stockage de mémoire directe (RAM), par opposition au stockage secondaire par débordement (stockage de fichiers). Non utilisé actuellement.
- `rootDirectory` : Le chemin d'accès au répertoire où se situe le magasin de contenu basé sur les fichiers. Non utilisé actuellement.

Structures de flux vidéo Kinesis

Vous pouvez utiliser les structures suivantes pour fournir des données à une instance d'un flux vidéo Kinesis.

Rubriques

- [StreamDefinition/StreamInfo](#)
- [ClientMetrics](#)
- [StreamMetrics](#)

StreamDefinition/StreamInfo

L'objet `StreamDefinition` dans la couche C++ englobe l'objet `StreamInfo` dans le code indépendant de la plateforme et fournit des valeurs par défaut dans le constructeur.

Champs réservés aux membres

| Champ | Type de données | Description | Valeur par défaut |
|------------------|---------------------------------|---|--|
| stream_name | string | Nom de flux facultatif. Pour plus d'informations sur la longueur du nom de flux, consultez Quotas du SDK pour les producteurs . Chaque flux doit avoir un nom unique. | Si aucun nom n'est spécifié, un nom est généré de façon aléatoire. |
| retention_period | duration<uint64_t, ratio<3600>> | La période de conservation du flux en secondes. 0 indique l'absence de conservation. | 3600 (Une heure) |
| tags | const map<string, string>* | Un mappage de paires clé-valeur contenant les informations d'utilisateur. Si le flux possède déjà un ensemble de balises, les nouvelles balises sont ajoutées à l'ensemble existant. | Pas de balise |
| kms_key_id | string | L'ID de AWS KMS clé à utiliser pour chiffrer le flux. Pour de plus amples informations, veuillez consulter Protection des | Clé KMS par défaut (aws/kinesisvideo .) |

| Champ | Type de données | Description | Valeur par défaut |
|----------------|-------------------------------|---|----------------------|
| | | données dans Kinesis Video Streams. | |
| streaming_type | Énumération STREAMING_TYPE | La seule valeur prise en charge est STREAMING_TYPE_REALTIME . | |
| content_type | string | Format du contenu du flux. La console Kinesis Video Streams peut lire du contenu dans ce formatvideo/h264 . | video/h264 |
| max_latency | duration< uint64_t, milli> | Latence maximale en millisecondes pour le flux. Le rappel de sollicitation de latence de flux (si spécifié) est appelé lorsque la durée de mise en tampon dépasse cette durée. 0 indique qu'aucun rappel de sollicitation de latence de flux ne sera appelé. | milliseconds::zero() |

| Champ | Type de données | Description | Valeur par défaut |
|--------------------------------|---------------------------------------|---|-------------------|
| <code>fragment_duration</code> | <code>duration<uint64_t></code> | <p>La durée de fragment souhaitée en secondes. Cette valeur est utilisée en combinaison avec la valeur <code>key_frame_fragmentation</code>. Si cette valeur est définie <code>false</code>, Kinesis Video Streams génère des fragments sur une image clé une fois cette durée écoulée. Par exemple, un flux audio Advanced Audio Coding (AAC) utilise chaque image en tant qu'image clé. La spécification de <code>key_frame_fragmentation = false</code> entraîne une fragmentation sur une image clé lorsque cette durée arrive à expiration, ce qui se traduit par des fragments de 2 secondes.</p> | 2 |

| Champ | Type de données | Description | Valeur par défaut |
|----------------|----------------------------|---|-------------------|
| timecode_scale | duration< uint64_t, milli> | L'échelle de code horaire MKV en millisecondes qui spécifie la granularité des codes horaires pour les images au sein du cluster MKV. Le code horaire d'image MKV est toujours relatif par rapport au démarrage du cluster. MKV utilise une valeur 16 bits signée (0-32767) pour représenter le code horaire au sein du cluster (fragment). Vérifiez que le code temporel de l'image peut être représenté avec l'échelle de code temporel donnée. La valeur d'échelle de code horaire par défaut de 1 ms garantit la représentation de la plus grande image en 32767 ms ~ = 32 secondes. Cela dépasse la durée de fragment maximum spécifiée dans Quotas du service | 1 |

| Champ | Type de données | Description | Valeur par défaut |
|--------------------------------------|-------------------|---|-------------------|
| | | Amazon Kinesis Video Streams qui est de 10 secondes. | |
| <code>key_frame_fragmentation</code> | <code>bool</code> | Indique s'il faut générer des fragments sur une image clé. Si la valeur est <code>true</code> , le kit SDK génère un début de fragment à chaque image clé. Si <code>false</code> , Kinesis Video Streams attend <code>fragment_duration</code> au moins et produit un nouveau fragment sur l'image clé qui le suit. | <code>true</code> |

| Champ | Type de données | Description | Valeur par défaut |
|------------------------------|-------------------|---|-------------------|
| <code>frame_timecodes</code> | <code>bool</code> | Indique s'il faut utiliser codes horaires d'image ou générer des horodatages à l'aide du rappel actuel. De nombreux encodeurs ne génèrent pas d'horodatages avec les images. La spécification <code>false</code> de ce paramètre garantit donc que les images sont horodatées lorsqu'elles sont placées dans Kinesis Video Streams. | <code>true</code> |

| Champ | Type de données | Description | Valeur par défaut |
|--------------------------------------|-------------------|---|-------------------|
| <code>absolute_fragment_times</code> | <code>bool</code> | Kinesis Video Streams utilise MKV comme mécanisme d'empaquetage sous-jacent. La spécification MKV est stricte en ce qui concerne l'aspect relatif du code horaire d'image par rapport au début du cluster (fragment). Toutefois, les codes horaires de cluster peuvent être absolus ou relatifs par rapport à l'heure de démarrage du flux. Si les horodatages sont relatifs, l'appel de l'API de service <code>PutMedia</code> utilisera l'horodatage de démarrage de flux facultatif et ajustera les horodatages de cluster. Le service stocke toujours les fragments avec leurs horodatages absolus. | <code>true</code> |

| Champ | Type de données | Description | Valeur par défaut |
|----------------------------------|-------------------|--|--|
| <code>fragment_acks</code> | <code>bool</code> | S'il faut recevoir des ACK de fragments au niveau de l'application (accusés de réception). | <code>true</code> signifie que le kit SDK recevra les accusés de réception et agira en conséquence. |
| <code>restart_on_error</code> | <code>bool</code> | Indique s'il faut redémarrer en cas d'erreurs spécifiques. | La valeur <code>true</code> signifie que le kit SDK essaie de redémarrer le streaming en cas d'erreur. |
| <code>recalculate_metrics</code> | <code>bool</code> | Indique s'il faut recalculer les métriques. Chaque appel à extraire les métriques peut recalculer celles qui permettent d'obtenir la dernière valeur « en cours », ce qui peut avoir un léger impact sur le processeur. Vous devrez peut-être régler ce paramètre <code>false</code> sur des power/ footprint appareils extrêmement bas pour économiser les cycles du processeur. Dans le cas contraire, nous vous déconseillons <code>false</code> d'utiliser cette valeur. | <code>true</code> |

| Champ | Type de données | Description | Valeur par défaut |
|-----------------------------------|-----------------------|---|---|
| <code>nal_adaptation_flags</code> | <code>uint32_t</code> | <p>Spécifie les indicateurs d'adaptation d'unité Network Abstraction Layer (NAL). Si le flux binaire est H.264 codé, il peut ensuite être traité comme brut ou empaqueté dans des NALU. Ils sont au format AVCC Annex-B ou au format AVCC. La plupart des producteurs et consommateurs de flux élémentaires (encodeurs de lecture et décodeurs) utilisent ce Annex-B format car il présente des avantages, tels que la récupération des erreurs. Higher-levelles systèmes utilisent le format AVCC, qui est le format par défaut pour MPEG, HLS, DASH, etc. La lecture de console utilise les extensions MSE (media source extensions) du navigateur pour décoder et lire le flux</p> | Par défaut, le format est adapté au Annex-B format AVCC à la fois pour les données de trame et pour les données privées du codec. |

| Champ | Type de données | Description | Valeur par défaut |
|-------|-----------------|---|-------------------|
| | | <p>utilisant le format AVCC. Pour H.264 (et pour M-JPEG et H.265), le SDK fournit des fonctionnalités d'adaptation.</p> <p>De nombreux flux élémentaires sont au format suivant. Dans cet exemple, Ab il s'agit du code de Annex-B démarrage (001 ou 0001).</p> <div data-bbox="829 888 1149 1283" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>Ab(Sps)Ab (Pps)Ab(I- frame)Ab(P/B- frame) Ab(P/B-fr ame)... Ab(Sps)Ab (Pps)Ab(I- frame)Ab(P/B- frame) Ab(P/B-fr ame)</pre> </div> <p>Dans le cas de H.264, les données privées du codec (CPD) se trouvent dans les paramètres SPS (ensemble de paramètres de séquence) et PPS (ensemble de paramètres d'image) et peuvent être</p> | |

| Champ | Type de données | Description | Valeur par défaut |
|------------|-----------------|--|-------------------|
| | | <p>adaptées au format AVCC. L'application peut extraire les CPD de l'image, sauf si le pipeline multimédia les fournit séparément. Il peut le faire en recherchant la première trame IDR (qui doit contenir le SPS et le PPS), en extrayant les deux NALU (qui le sont Ab (Sps) Ab (Pps)) et en la configurant dans le CPD in. StreamDefinition</p> <p>Pour de plus amples informations, veuillez consulter the section called "drapeaux d'adaptation NAL".</p> | |
| frame_rate | uint32_t | Débit d'images attendu. Cette valeur est utilisée pour mieux calculer les besoins de mise en mémoire tampon. | 25 |

| Champ | Type de données | Description | Valeur par défaut |
|-------------------|-----------------|--|-------------------|
| avg_bandwidth_bps | uint32_t | Bande passante moyenne attendue pour le flux. Cette valeur est utilisée pour mieux calculer les besoins de mise en mémoire tampon. | 4 * 1024 * 1024 |

| Champ | Type de données | Description | Valeur par défaut |
|------------------------------|---------------------------------------|--|-------------------|
| <code>buffer_duration</code> | <code>duration<uint64_t></code> | Durée de la mémoire tampon de flux en secondes. Le SDK conserve les cadres du magasin de contenu pendant une durée maximale de <code>buffer_duration</code> , après quoi les cadres précédents sont supprimés au fur et à mesure que la fenêtre avance. Si le cadre supprimé n'a pas été envoyé au backend, le rappel du cadre supprimé est appelé. Si la durée de mise en tampon actuelle est supérieure à <code>max_latency</code> , le rappel de sollicitation de latence de flux est appelé. La mémoire tampon est tronquée au début du fragment suivant lorsque l'accusé de réception de conservation de ce dernier est reçu. Cela indique que le contenu a été conservé durablement. | 120 |

| Champ | Type de données | Description | Valeur par défaut |
|-------|-----------------|--|-------------------|
| | | nt dans le cloud, afin de le stockage du contenu sur l'appareil local ne soit plus nécessaire. | |

| Champ | Type de données | Description | Valeur par défaut |
|-----------------|--------------------|--|-------------------|
| replay_duration | duration<uint64_t> | Durée, en secondes, nécessaire pour faire reculer le lecteur actuel pour le rejouer en cas d'erreur si le redémarrage est activé. La restauration s'arrête au début du tampon (si le streaming vient de commencer ou si l'accusé de réception de conservation est arrivé). La restauration essaie de se poser sur une image clé qui indique un début de fragment. Si l'erreur « à l'origine du redémarrage » n'indique pas un hôte mort (l'hôte est toujours en vie et contient les données de trame dans ses tampons internes) , le rollback s'arrête à la dernière trame ACK reçue. Ensuite, la restauration se poursuit jusqu'à l'image clé suivante, car l'ensemble du fragment est déjà | 40 |

| Champ | Type de données | Description | Valeur par défaut |
|----------------------|--------------------|--|-------------------|
| | | stocké dans la mémoire hôte. | |
| connection_staleness | duration<uint64_t> | Durée, en secondes, après laquelle le rappel de stagnation du flux est appelé si le SDK ne reçoit pas l'ACK de mise en mémoire tampon. Cela indique que les images sont envoyées depuis l'appareil, mais le backend ne les accuse pas de réception. Cet état indique une connexion coupée au niveau du saut intermédiaire ou de l'équilibreur de charge. | 30 |
| codec_id | string | ID de codec de la piste MKV. | V_MPEG4/ISO/AVC |
| track_name | string | Nom de la piste MKV. | kinesis_video |

| Champ | Type de données | Description | Valeur par défaut |
|-----------------------|-----------------|---|-------------------|
| codec PrivateData | unsigned char* | La mémoire tampon des données privées codec (CPD). Si le pipeline média contient les informations sur les données de type CPD avant que le flux démarre, elle peut être définie dans <code>StreamDefinition.c</code> <code>odecPrivateData</code> . Les bits sont copiés et la mémoire tampon peut être réutilisée ou libérée après l'appel pour créer le flux. Toutefois, si les données ne sont pas disponibles lors de la création du flux, elles peuvent être définies dans l'une des surcharges de la <code>KinesisVideoStream</code> <code>.start(cpd)</code> fonction. | null |
| codec PrivateDataSize | uint32_t | La taille de la mémoire tampon des données privées codec. | 0 |

ClientMetrics

L'ClientMetricsobjet est rempli en appelantgetKinesisVideoMetrics.

Champs réservés aux membres

| Champ | Type de données | Description |
|----------------------------|-----------------|---|
| Version | UINT32 | Version de la structure, définie dans la macro CLIENT_METRICS_CURRENT_VERSION . |
| contentStoreSize | UINT64 | Taille du magasin de contenu global, en octets. Il s'agit de la valeur spécifiée dans DeviceInfo.StorageInfo.storageSize . |
| contenu StoreAvailableSize | UINT64 | Taille de stockage actuellement disponible en octets. |
| contenu StoreAllocatedSize | UINT64 | Taille actuellement allouée. Le total de la taille allouée et de la taille disponible doit être légèrement inférieur à la taille de stockage globale, en raison des données comptables internes et de l'implémentation du magasin de contenu. |
| total ContentViewsSize | UINT64 | Taille de la mémoire allouée pour l'ensemble des affichages de contenu pour tous les flux. Cela n'est pas pris en compte dans la taille de stockage. Cette mémoire est allouée à l'aide de la macro |

| Champ | Type de données | Description |
|--------------------|-----------------|--|
| | | MEMALLOC, qui peut être remplacée pour fournir un allocateur personnalisé. |
| total FrameRate | UINT64 | Débit d'images total observé pour tous les flux. |
| total TransferRate | UINT64 | Débit de flux total observé, en octets par seconde, pour tous les flux. |

StreamMetrics

L'StreamMetricsobjet est rempli en appelantgetKinesisVideoMetrics.

Champs réservés aux membres

| Champ | Type de données | Description |
|---------------------|-----------------|--|
| Version | UINT32 | Version de la structure, définie dans la macro STREAM_METRICS_CURRENT_VERSION . |
| actuel ViewDuration | UINT64 | Durée des images accumulées. Dans le cas d'un réseau rapide, cette durée est soit nulle, soit la durée de la trame (lorsque la trame est transmise). Si la durée devient plus longue que celle max_latency spécifiée dans leStreamDefinition , le rappel de latence du flux est appelé s'il est spécifié. La durée est spécifiée en unités |

| Champ | Type de données | Description |
|------------------------------|-----------------|--|
| | | de 100ns, qui constitue l'unité de temps par défaut pour la couche PIC. |
| dans l'ensemble ViewDuration | UINT64 | Durée d'affichage globale. Si le flux est configuré sans ACK ni persistance, cette valeur augmente au fur et à mesure que les images sont insérées dans le flux vidéo Kinesis et devient égale à celle de <code>buffer_duration</code> de <code>StreamDefinition</code> . Lorsque les ACK sont activés et que l'ACK persistant est reçu, la mémoire tampon est réduite à l'image clé suivante. Cela est dû au fait que l'horodatage ACK indique le début de l'ensemble du fragment. La durée est spécifiée en unités de 100-ns qui constitue l'unité de temps par défaut pour la couche PIC. |
| actuel ViewSize | UINT64 | Taille, en octets, de la mémoire tampon actuelle. |
| dans l'ensemble ViewSize | UINT64 | Taille de l'affichage global, en octets. |
| actuel FrameRate | UINT64 | Débit d'images observé pour le flux actuel. |

| Champ | Type de données | Description |
|---------------------|-----------------|---|
| actuel TransferRate | UINT64 | Débit de transfert observé, en octets par seconde, pour le flux actuel. |

Rappels du SDK pour les producteurs

Les classes et les méthodes du SDK Amazon Kinesis Video Streams Producer ne gèrent pas leurs propres processus. À la place, elles utilisent les appels et les événements entrants de la fonction afin de planifier les rappels pour communiquer avec l'application.

Il existe deux schémas de rappel que l'application peut utiliser pour interagir avec le kit SDK :

- [CallbackProvider](#)— Cet objet expose tous les rappels du composant de code indépendant de la plate-forme (PIC) à l'application. Ce schéma permet des fonctionnalités complètes, mais cela signifie également que l'implémentation doit gérer toutes les méthodes d'API publique et les signatures dans la couche C++.
- [StreamCallbackProvider](#) et [ClientCallbackProvider](#) — Ces objets exposent les rappels spécifiques au flux et au client, et la couche C++ du SDK expose le reste des rappels. Il s'agit du schéma de rappel préféré pour interagir avec le kit SDK Producteur.

Le diagramme suivant illustre le modèle d'objet des objets de rappel :

Dans le diagramme précédent, `DefaultCallbackProvider` dérive de `CallbackProvider` (qui expose tous les rappels dans le PIC) et contient `StreamCallbackProvider` et `ClientCallbackProvider`.

Cette rubrique contient les sections suivantes :

- [ClientCallbackProvider](#)
- [StreamCallbackProvider](#)
- [ClientCallbacks structure](#)
- [Implémentations de callback pour réessayer le streaming](#)

ClientCallbackProvider

L'objet `ClientCallbackProvider` expose des fonctions de rappel au niveau du client. Les détails des fonctions sont décrites dans la section [the section called “ClientCallbacks”](#).

Méthodes de rappel :

- `getClientReadyCallback`— Signale un état de préparation pour le client.
- `getStorageOverflowPressureCallback`— Signale un débordement ou une pression de stockage. Ce rappel est appelé lorsque l'utilisation du stockage est inférieure à la valeur `STORAGE_PRESSURE_NOTIFICATION_THRESHOLD`, qui correspond à 5 pour cent du volume de stockage global. Pour de plus amples informations, veuillez consulter [StorageInfo](#).

StreamCallbackProvider

L'objet `StreamCallbackProvider` expose des fonctions de rappel au niveau du flux.

Méthodes de rappel :

- `getDroppedFragmentReportCallback` : Signale un fragment abandonné.
- `getDroppedFrameReportCallback`— Signale une image perdue.
- `getFragmentAckReceivedCallback`— Signale qu'un fragment ACK a été reçu pour le flux.
- `getStreamClosedCallback`— Signale une condition de fermeture d'un cours d'eau.
- `getStreamConnectionStaleCallback`— Signale un état de connexion périmé. Dans cette situation, le producteur envoie des données au service mais ne reçoit pas d'accusé de réception.
- `getStreamDataAvailableCallback`— Indique que les données sont disponibles dans le flux.
- `getStreamErrorReportCallback`— Signale une condition d'erreur de flux.
- `getStreamLatencyPressureCallback`— Signale une condition de latence du flux, c'est-à-dire lorsque la taille de la mémoire tampon cumulée est supérieure à la `max_latency` valeur. Pour de plus amples informations, veuillez consulter [StreamDefinition/StreamInfo](#).
- `getStreamReadyCallback`: —Signale une condition de disponibilité du flux.
- `getStreamUnderflowReportCallback`— Signale un état de sous-débit d'un cours d'eau. Cette fonction n'est pas utilisée actuellement et est réservée pour une utilisation future.

Pour le code source de `StreamCallbackProvider`, voir [StreamCallbackProvider.h](#).

ClientCallbacks structure

La structure `ClientCallbacks` contient les points d'entrée de la fonction de rappel que le PIC appelle en cas d'événements spécifiques. La structure contient également des informations de version dans le champ `CALLBACKS_CURRENT_VERSION` ainsi qu'un champ `customData` pour les données définies par l'utilisateur et renvoyées avec les fonctions de rappel individuelles.

L'application client peut utiliser un pointeur `this` pour le champ `custom_data` afin de mapper les fonctions de membre vers les fonctions `ClientCallback` statiques lors de l'exécution, comme illustré dans l'exemple de code suivant :

```
STATUS TestStreamCallbackProvider::streamClosedHandler(UINT64 custom_data,
    STREAM_HANDLE stream_handle, UINT64 stream_upload_handle) {
    LOG_INFO("Reporting stream stopped.");

    TestStreamCallbackProvider* streamCallbackProvider =
        reinterpret_cast<TestStreamCallbackProvider*> (custom_data);
    streamCallbackProvider->streamClosedHandler(...);
```

Événements

| Fonction | Description | Type |
|---------------------------------------|---|----------------|
| <code>CreateDeviceFunc</code> | Pas implémenté actuellement sur le backend. Cet appel échoue lorsque appelé à partir de Java ou C++. Les autres clients effectuent une initialisation propre à la plateforme. | API de backend |
| <code>CreateStreamFunc</code> | Appelé lors de la création du flux. | API de backend |
| <code>DescribeStreamFunc</code> | Appelé lorsque <code>DescribeStream</code> est appelé. | API de backend |
| <code>GetStreamingEndpointFunc</code> | Appelé lorsque <code>GetStreamingEndpoint</code> est appelé. | API de backend |

| Fonction | Description | Type |
|---------------------------|---|-----------------|
| GetStreamingTokenFunc | Appelé lorsque GetStreamingToken est appelé. | API de backend |
| PutStreamFunc | Appelé lorsque PutStream est appelé. | API de backend |
| TagResourceFunc | Appelé lorsque TagResource est appelé. | API de backend |
| | | |
| CreateMutexFunc | Crée un mutex de synchronisation. | Synchronisation |
| FreeMutexFunc | Libère les mutex. | Synchronisation |
| LockMutexFunc | Verrouille le mutex de synchronisation. | Synchronisation |
| TryLockMutexFunc | Essaie de verrouiller les mutex. Pas implémenté actuellement. | Synchronisation |
| UnlockMutexFunc | Déverrouille le mutex. | Synchronisation |
| | | |
| ClientReadyFunc | Appelé lorsque le client passe en l'état prêt. | Notification |
| DroppedFrameReportFunc | Signale lorsqu'une image est abandonnée. | Notification |
| DroppedFragmentReportFunc | Signale lorsqu'un fragment est abandonné. Cette fonction n'est pas utilisée actuellement et est réservée pour une utilisation future. | Notification |

| Fonction | Description | Type |
|-----------------------------|--|--------------|
| FragmentAckReceiveFunc | Appelé lorsqu'un accusé de réception de fragment (mise en mémoire tampon, réception, conservation et erreur) est reçu. | Notification |
| StorageOverflowPressureFunc | Appelé lorsque l'utilisation du stockage est inférieure à la valeur STORAGE_PRESSURE_NOTIFICATION_THRESHOLD qui correspond à 5 pour cent du volume de stockage global. | Notification |
| StreamClosedFunc | Appelé lorsque les derniers bits des images restantes sont diffusés en continu. | Notification |
| StreamConnectionStaleFunc | Appelé lorsque le flux passe en état de connexion obsolète. Dans cet état, l'application Producteur envoie des données vers le service, mais ne reçoit pas d'accusés de réception. | Notification |
| StreamDataAvailableFunc | Appelé lorsque les données de flux sont disponibles. | Notification |
| StreamErrorReportFunc | Appelé en cas d'erreur de flux. Dans cet état, le PIC ferme automatiquement le flux. | Notification |

| Fonction | Description | Type |
|--|---|---------------------------|
| <code>StreamLatencyPressureFunc</code> | Appelé lorsque le flux passe en état de latence qui correspond au moment où la taille de la mémoire tampon accumulée est supérieure à la valeur <code>max_latency</code> . Pour de plus amples informations, veuillez consulter StreamDefinition/StreamInfo . | Notification |
| <code>StreamReadyFunc</code> | Appelé lorsque le flux passe en état prêt. | Notification |
| <code>StreamUnderflowReportFunc</code> | Cette fonction n'est pas utilisée actuellement et est réservée pour une utilisation future. | Notification |
| <code>DeviceCertToTokenFunc</code> | Renvoie le certificat de connexion sous la forme d'un jeton. | Intégration de plateforme |
| <code>GetCurrentTimeFunc</code> | Renvoie l'heure actuelle. | Intégration de plateforme |
| <code>GetDeviceCertificateFunc</code> | Renvoie le certificat de l'appareil. Cette fonction n'est pas utilisée actuellement et est réservée pour une utilisation future. | Intégration de plateforme |

| Fonction | Description | Type |
|---------------------------------------|--|---------------------------|
| <code>GetDeviceFingerprintFunc</code> | Renvoie l'empreinte de l'appareil. Cette fonction n'est pas utilisée actuellement et est réservée pour une utilisation future. | Intégration de plateforme |
| <code>GetRandomNumberFunc</code> | Renvoie un nombre aléatoire entre 0 et <code>RAND_MAX</code> . | Intégration de plateforme |
| <code>GetSecurityTokenFunc</code> | Renvoie le jeton de sécurité transmis aux fonctions qui communiquent avec l'API principale. L'implémentation peut spécifier les <code>AccessKeyId</code> , <code>SecretKeyId</code> sérialisés et le jeton de session. | Intégration de plateforme |
| <code>LogPrintFunc</code> | Consigne une ligne de texte avec la balise et le niveau de journalisation. Pour de plus amples informations, veuillez consulter <code>PlatformUtils.h</code> . | Intégration de plateforme |

Pour les fonctions d'intégration de plateforme dans le tableau précédent, le dernier paramètre est une structure `ServiceCallContext` qui comporte les champs suivants :

- `version` : La version de la structure.
- `callAfter` : Un délai absolu après lequel appeler la fonction.
- `timeout` : Le délai d'expiration de l'opération en unités de 100 nanosecondes.
- `customData` : Une valeur définie par l'utilisateur à retransmettre au client.
- `pAuthInfo` : les informations d'identification pour l'appel. Pour plus d'informations, consultez la structure (`__AuthInfo`) suivante.

Les informations d'autorisation sont fournies à l'aide de la structure `__AuthInfo` sous la forme d'informations d'identification sérialisées ou de jeton d'authentification propre au fournisseur. Cette structure possède les champs suivants :

- `version` : La version de la structure `__AuthInfo`.
- `type` : Une valeur `AUTH_INFO_TYPE` définissant le type d'informations d'identification (certificat ou jeton de sécurité).
- `data` : Une matrice à octets contenant les informations d'authentification.
- `size` : La taille du paramètre `data`.
- `expiration` : Le délai d'expiration des informations d'identification en unités de 100 nanosecondes.

Implémentations de callback pour réessayer le streaming

Le kit SDK Kinesis Video Producer fournit le statut de diffusion en continu par le biais de fonctions de rappel. Nous vous recommandons de mettre en œuvre les mécanismes de rappel suivants pour remédier à tout problème réseau momentanément rencontré pendant le streaming.

- **Rappel de la pression de latence du flux** : ce mécanisme de rappel est lancé lorsque le SDK rencontre une condition de latence du flux. Cela se produit lorsque la taille de mémoire tampon accumulée est supérieure à la valeur `MAX_LATENCY`. Lorsque le flux est créé, l'application de diffusion en continu définit une valeur par défaut de 60 secondes pour `MAX_LATENCY`. L'implémentation standard pour ce rappel est de réinitialiser la connexion. Vous pouvez utiliser l'exemple d'implémentation <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp/blob/master/kinesis-video-c-producer/src/source/StreamLatencyStateMachine.cs> selon vos besoins. Notez qu'il n'est pas possible de stocker les trames non livrées en raison d'une panne du réseau dans un stockage secondaire à des fins de remblayage.
- **Rappel de temporisation des flux** : ce rappel est lancé lorsque le producteur peut envoyer des données au service Amazon Kinesis Data Streams (liaison montante) mais qu'il n'est pas en mesure de récupérer les accusés de réception (ACK mis en mémoire tampon) à temps (la valeur par défaut est de 60 secondes). Selon les paramètres du réseau, le rappel de la pression de latence du flux ou le rappel de l'obsolescence du flux, ou les deux, peuvent être initiés. À l'image de l'implémentation d'un nouveau rappel de sollicitation de latence de flux, l'implémentation standard consiste à réinitialiser la connexion et à en démarrer une nouvelle pour la diffusion en continu. Vous pouvez utiliser l'exemple d'implémentation <https://github.com/aws-labs/amazon->

[kinesis-video-streams-producer-c/blob/master/src/source/ConnectionStateStateMachine.c](#) selon vos besoins.

- Rappel d'erreur de flux : ce rappel est lancé lorsque le SDK rencontre un délai d'expiration sur la connexion réseau ou d'autres erreurs lors de l'appel aux appels du service API KVS.
- Rappel de trame supprimé : ce rappel est lancé lorsque la taille de stockage est pleine, soit en raison d'une lenteur du réseau, soit d'une erreur de flux. Si la vitesse du réseau entraîne des pertes d'images, vous pouvez augmenter la taille de stockage, réduire la taille des images vidéo ou la fréquence d'images pour qu'elle corresponde à la vitesse du réseau.

Utilisation des métadonnées de streaming avec Kinesis Video Streams

Vous pouvez utiliser le SDK Amazon Kinesis Video Streams Producer pour intégrer des métadonnées au niveau du fragment individuel dans un flux vidéo Kinesis. Dans Kinesis Video Streams, les métadonnées sont une paire clé-valeur modifiable. Vous pouvez l'utiliser pour décrire le contenu du fragment, intégrer les mesures de capteur associées qui doivent être transférées avec le fragment lui-même ou répondre à d'autres besoins personnalisés. Les métadonnées sont mises à disposition dans le cadre des opérations d'API [GetMedia](#) ou [GetMediaForFragmentList](#). Il est stocké avec les fragments pendant toute la durée de conservation du stream. Vos applications consommatrices peuvent lire, traiter et réagir en fonction des métadonnées à l'aide du [Regardez le résultat des caméras à l'aide de la bibliothèque d'analyseurs](#).

Il existe deux modes dans lequel les métadonnées peuvent être intégrées à des fragments dans un flux :

- Non persistant : vous pouvez apposer des métadonnées de manière ponctuelle ou ad hoc sur des fragments d'un flux, en fonction de critères spécifiques à l'entreprise qui se sont produits. Par exemple, une caméra intelligente détecte un mouvement et ajoute des métadonnées aux fragments correspondants contenant le mouvement avant de les envoyer vers son flux vidéo Kinesis. Vous pouvez appliquer les métadonnées au fragment selon le format suivant : `Motion = true`.
- Persistant : vous pouvez apposer des métadonnées sur des fragments consécutifs et successifs d'un flux en fonction d'un besoin continu. Par exemple, une caméra intelligente envoie les coordonnées de latitude et de longitude actuelles associées à tous les fragments qu'elle envoie à son flux vidéo Kinesis. Vous pouvez appliquer les métadonnées à tous les fragments selon le format suivant : `Lat = 47.608013N , Long = -122.335167W`.

Vous pouvez associer simultanément les métadonnées au même fragment dans ces deux modes, en fonction des besoins de votre application. Les métadonnées intégrées peuvent inclure les objets détectés, l'activité suivie, les coordonnées GPS ou d'autres données personnalisées que vous souhaitez associer aux fragments du flux. Les métadonnées sont codées sous forme de paires de chaînes clé-valeur.

Ajouter des métadonnées à un flux vidéo Kinesis

Les métadonnées que vous ajoutez à un flux vidéo Kinesis sont modélisées sous forme de balises MKV, qui sont implémentées sous forme de paires clé-valeur.

Les métadonnées peuvent être transitoires, pour marquer un événement dans le flux par exemple, ou persistantes, pour identifier les fragments dans lesquels un événement donné est en cours par exemple. Un élément de métadonnées persistant est conservé et est appliqué à chaque fragment consécutif jusqu'à son annulation.

Note

Les éléments de métadonnées ajoutés avec [Transférer vers Kinesis Video Streams](#) sont distincts de la mise en œuvre des API de balisage au niveau du flux avec [TagStream](#), [UntagStream](#) et [ListTagsForStream](#).

API de métadonnées de streaming

Vous pouvez utiliser les opérations suivantes dans le SDK du producteur pour implémenter les métadonnées de streaming.

Rubriques

- [PIC](#)
- [SDK pour producteurs de C++](#)
- [SDK pour le producteur Java](#)
- [Métadonnées persistantes et non persistantes](#)

PIC

```
PUBLIC_API STATUS putKinesisVideoFragmentMetadata(STREAM_HANDLE streamHandle,  
PCHAR name,
```

```
PCHAR value,  
BOOL persistent);
```

SDK pour producteurs de C++

```
/**  
 * Appends a "tag" or metadata - a key/value string pair into the stream.  
 */  
bool putFragmentMetadata(const std::string& name, const std::string& value, bool  
    persistent = true);
```

SDK pour le producteur Java

Vous pouvez utiliser le SDK Java Producer pour ajouter des métadonnées à un utilisateur `MediaSource` : `MediaSourceSink.onCodecPrivateData`

```
void onFragmentMetadata(final @NonNull String metadataName, final @NonNull String  
    metadataValue, final boolean persistent)  
    throws KinesisVideoException;
```

Métadonnées persistantes et non persistantes

Pour les métadonnées non persistantes, vous pouvez ajouter plusieurs éléments de métadonnées portant le même nom. Le SDK du producteur collecte les éléments de métadonnées dans la file d'attente de métadonnées jusqu'à ce qu'ils soient ajoutés au fragment suivant. La file d'attente des métadonnées est effacée lorsque les éléments de métadonnées sont appliqués au flux. Pour reproduire les métadonnées, appelez à nouveau `putKinesisVideoFragmentMetadata` ou `putFragmentMetadata`.

Pour les métadonnées persistantes, le SDK du producteur collecte les éléments de métadonnées dans la file d'attente de métadonnées de la même manière que pour les métadonnées non persistantes. Toutefois, les éléments de métadonnées ne sont pas supprimés de la file d'attente lorsqu'ils sont ajoutés au fragment suivant.

L'appel de `putKinesisVideoFragmentMetadata` ou `putFragmentMetadata` avec `persistent` défini sur `true` a le comportement suivant :

- L'appel de l'API place l'élément des métadonnées dans la file d'attente. Les métadonnées sont ajoutées sous la forme d'une balise MVK à chaque fragment tandis que l'élément se trouve dans la file d'attente.

- L'appel de l'API avec le même nom et une autre valeur en tant qu'élément de métadonnées précédemment ajouté remplace l'élément.
- L'appel de l'API avec une valeur vide supprime (annule) l'élément de métadonnées de la file d'attente des métadonnées.

Utiliser IPv6 avec Kinesis Video Streams

Vous pouvez configurer Kinesis Video Streams pour utiliser IPv6 à la fois pour les opérations du plan de contrôle et du plan de données. Cela permet à vos applications de communiquer avec les services Kinesis Video Streams à l'aide d'adresses IPv6 via des points de terminaison à double pile.

Note

La prise en charge d'IPv6 nécessite des versions du SDK et des paramètres de configuration spécifiques. Assurez-vous que vos versions du SDK et du SDK Kinesis Video Streams prennent en charge AWS les points de terminaison à double pile IPv6. Dual-stack les points de terminaison prennent en charge le trafic IPv4 et IPv6 et sont disponibles pour certains services dans certaines régions.

Kinesis Video Streams prend en charge le protocole IPv6 via des points de terminaison à double pile pour les applications destinées aux producteurs et aux consommateurs. Vous pouvez configurer vos applications pour utiliser IPv6 pour les appels d'API du plan de contrôle et les opérations de streaming du plan de données.

Configurez le AWS SDK pour IPv6

Si vous utilisez le AWS SDK pour appeler les API du plan de contrôle Kinesis Video Streams dans votre configuration de production, vous pouvez activer IPv6 en configurant des points de terminaison à double pile. Le AWS SDK fournit plusieurs méthodes standardisées pour activer les points de terminaison à double pile.

⚠ Important

Lorsque les points de terminaison à double pile sont activés, le SDK tente d'utiliser des points de terminaison à double pile pour effectuer des requêtes réseau. S'il n'existe pas de point de terminaison à double pile pour le service ou la région, la demande échoue.

Utiliser des variables d'environnement

Définissez la variable d'environnement suivante pour activer les points de terminaison IPv6 à double pile :

```
export AWS_USE_DUALSTACK_ENDPOINT=true
```

Utilisez la commande AWS Fichier de configuration

Ajoutez le paramètre suivant à votre fichier AWS de configuration (~/.aws/config) :

```
[default]
use_dualstack_endpoint = true
```

Utiliser les propriétés du système JVM (SDK Java et Kotlin uniquement)

Pour les applications Java et Kotlin, définissez la propriété système JVM suivante :

```
-Daws.useDualstackEndpoint=true
```

Ou par programmation dans votre code Java :

```
System.setProperty("aws.useDualstackEndpoint", "true");
```

Prise en charge de SDK

Les AWS SDK suivants prennent en charge la configuration des terminaux à double pile :

| Kit SDK | Pris en charge | Méthodes de configuration |
|------------|----------------|--|
| AWS CLI v2 | Oui | Variable d'environnement, fichier de configuration |

| Kit SDK | Pris en charge | Méthodes de configuration |
|----------------------------|----------------|---|
| SDK pour C++ | Oui | Variable d'environnement, fichier de configuration |
| SDK pour Go V2 (1.x) | Oui | Variable d'environnement, fichier de configuration |
| SDK pour Java 2.x | Oui | Variable d'environnement, fichier de configuration, propriété JVM |
| Kit SDK pour Java 1.x | Non | Non pris en charge |
| SDK pour 3.x JavaScript | Oui | Variable d'environnement, fichier de configuration |
| Kit SDK for Python (Boto3) | Oui | Variable d'environnement, fichier de configuration |

Après avoir configuré des points de terminaison à double pile, le AWS SDK utilise automatiquement les points de terminaison IPv6 lorsqu'il appelle les API du plan de contrôle Kinesis Video Streams.

Configuration du SDK Kinesis Video Streams Producer pour IPv6

Le SDK Kinesis Video Streams Producer fournit des options de configuration IPv6 pour les opérations du plan de contrôle et du plan de données. Ces paramètres fonctionnent avec la configuration du point de terminaison à double pile du AWS SDK.

Configuration du SDK C/C ++ Producer

Les points de terminaison par défaut et la chaîne de résolution DNS sont implémentés par le Producer-C SDK KVS version 1.6.0. Il vérifie séquentiellement chaque endroit où vous pouvez définir la configuration de ces paramètres, puis sélectionne le premier que vous définissez. La séquence prédéfinie est la suivante :

Pour plus d'informations sur les SDK Producer, consultez les rubriques relatives au [SDK Producer pour C](#), au [SDK Producer pour C++](#) et au [SDK Producer](#).

Configuration du terminal

1. Le `controlPlaneUrl` paramètre pour `createAbstractDefaultCallbacksProvider`.
2. Paramètres CMake de configuration du point de terminaison : (-
`DAWS_KVS_USE_LEGACY_ENDPOINT_ONLY=TRUE,-`
`DAWS_KVS_USE_DUAL_STACK_ENDPOINT_ONLY=TRUE`)
3. Variables d'environnement : (`export AWS_USE_DUALSTACK_ENDPOINT=TRUE`)
 - Si tel `AWS_USE_DUALSTACK_ENDPOINT` est le cas `TRUE` (sans distinction majuscules/majuscules), le point de terminaison à double pile sera utilisé.
4. Dans le cas contraire, l'ancien point de terminaison sera construit et utilisé.

Avec 2, 3 et 4, le point de terminaison sera construit en fonction de la région fournie à `createAbstractDefaultCallbacksProvider`.

Filtrage DNS

Le SDK KVS Producer définira le `CURLOPT_IPRESOLVE` paramètre approprié en fonction de la configuration :

1. Paramètres CMake de résolution DNS : (-`DAWS_KVS_IPV4_ONLY=TRUE,-`
`DAWS_KVS_IPV6_ONLY=TRUE,-DAWS_KVS_IPV4_AND_IPV6_ONLY=TRUE`)
2. Variables d'environnement (`export AWS_KVS_USE_IPV4=TRUE,export`
`AWS_KVS_USE_IPV6=TRUE`)
3. Dans le cas contraire, aucun filtrage n'aura lieu. Les adresses IP IPv4 et IPv6, si elles sont renvoyées par le DNS, peuvent être utilisées.

Note

Si les paramètres du filtre DNS sont définis pour filtrer les adresses IP IPv6, mais que la configuration du SDK consiste à utiliser l'ancien point de terminaison (IPv4-only adresses renvoyées), la demande échouera.

La version 3.5.0 du SDK C++ Producer utilise le Producer-C SDK 1.6.0 pour les appels d'API KVS.

Configurer le plugin GStreamer

Le plugin GStreamer utilise le SDK C Producer sous-jacent, de sorte que la configuration IPv6 est gérée automatiquement lorsque vous configurez le SDK C pour IPv6 comme décrit précédemment.

Il n'est pas nécessaire de modifier le code, il suffit de créer le SDK avec les paramètres CMake ou de définir les variables d'environnement appropriées comme décrit dans la section précédente.

Résolution des extrémités du plan de données

Pour les opérations sur le plan de données, utilisez l'GetDataEndpointAPI pour récupérer le point de terminaison du plan de données à double pile approprié. Le service renvoie le point de terminaison correspondant en fonction de l'URL de la demande.

Exemple :

- Si la demande GetDataEndpoint d'API est adressée à l'ancien point de terminaison se terminant par `.amazonaws.com`, Kinesis Video Streams renvoie l'ancien point de terminaison du plan de données se terminant par `.amazonaws.com`
- Si la demande GetDataEndpoint d'API est envoyée au point de terminaison à double pile se terminant par `.api.aws`, Kinesis Video Streams renvoie le point de terminaison du plan de données à double pile se terminant par `.api.aws`

Configurez le AWS CLI pour IPv6

Si vous utilisez les opérations AWS CLI pour Kinesis Video Streams (généralement pour des travaux de validation de concept), vous pouvez activer IPv6 en configurant des points de terminaison à double pile.

Utiliser une variable d'environnement

```
export AWS_USE_DUALSTACK_ENDPOINT=true
```

Utilisez la commande AWS Fichier de configuration

Ajoutez ce qui suit à votre fichier AWS CLI de configuration (`~/.aws/config`) :

```
[default]
```

```
use_dualstack_endpoint = true
```

Exemples de configuration

Exemple de SDK C

Pour créer le Producer-C SDK KVS en IPV6-only mode et ignorer la configuration des variables d'environnement, utilisez les commandes suivantes pour créer le SDK :

```
cmake .. -DAWS_KVS_USE_DUAL_STACK_ENDPOINT_ONLY=TRUE -DAWS_KVS_IPV6_ONLY=TRUE
make -j
```

Note

Si vous avez déjà créé le SDK, vous devrez effectuer une nouvelle génération. Supprimez les dossiers de compilation, de code source ouvert et de dépendance existants avant d'exécuter les commandes de génération.

Considérations

Exigences réseau

- Assurez-vous que votre infrastructure réseau prend en charge la connectivité IPv6
- Vérifiez que vos groupes de sécurité et vos ACL réseau autorisent le trafic IPv6
- Testez la connectivité aux points de terminaison AWS IPv6 depuis votre environnement de déploiement
- Dual-stack les points de terminaison sont disponibles pour certains services dans certaines régions. Vérifiez la disponibilité pour vos régions cibles

Compatibilité avec le SDK

- Assurez-vous que vous utilisez une version du AWS SDK prise en charge
- Le AWS SDK pour Java 1.x ne prend pas en charge la configuration des points de terminaison à double pile
- Pour le SDK pour Go 1.x (V1), vous devez activer le chargement depuis le fichier de configuration pour utiliser les paramètres du fichier de configuration partagés

Tests et validation

Avant de déployer des applications IPv6-enabled Kinesis Video Streams en production :

- Tester les opérations du plan de contrôle (création de flux, suppression, listage)
- Vérifier le fonctionnement du plan de données (ingestion et consommation de vidéos)
- Validez les performances et la connectivité dans votre environnement réseau
- Exécutez des tests Canary pour garantir la cohérence des fonctionnalités IPv6
- Testez le comportement du basculement lorsque les points de terminaison à double pile ne sont pas disponibles

Clients mis à niveau pour inclure IPv6

Lorsque vous activez IPv6 pour Kinesis Video Streams, vous devrez peut-être mettre à jour vos configurations et politiques existantes dans plusieurs domaines afin de garantir le fonctionnement continu.

Politiques IAM et filtrage des adresses IP

Si vous utilisez le filtrage des adresses IP source dans vos politiques utilisateur IAM, vos politiques de rôle ou vos politiques basées sur les ressources, vous devez mettre à jour ces politiques pour inclure les plages d'adresses IPv6.

Important

Les politiques IAM existantes qui utilisent des blocs d'adresse CIDR IPv4 dans `IpAddress` ou dans des `NotIpAddress` conditions ne fonctionneront pas automatiquement avec les adresses IPv6. Vous devez ajouter explicitement des plages IPv6 pour maintenir le contrôle d'accès.

Exemple de mise à jour de la politique IAM pour IPv6 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
"Effect": "Allow",
"Principal": "*",
"Action": "kinesisvideo:*",
"Resource": "*",
"Condition": {
  "IpAddress": {
    "aws:SourceIp": [
      "192.0.2.0/24",
      "203.0.113.0/24",
      "2001:db8::/32"
    ]
  }
}
}
```

Journalisation et surveillance

Les adresses IPv6 ont un format différent de celui des adresses IPv4, ce qui peut avoir un impact sur vos systèmes de journalisation, de surveillance et d'analyse.

journaux

les journaux contiendront les adresses IPv6 dans le `sourceIPAddress` champ lorsque les demandes sont effectuées via IPv6. Mettez à jour vos outils et scripts d'analyse des journaux pour gérer les formats d'adresse IPv6.

Exemple d'adresse IPv6 dans les journaux :

```
{
  "sourceIPAddress": "2001:db8::1",
  "eventName": "CreateStream",
  "eventSource": "kinesisvideo.amazonaws.com"
}
```

Résolution des problèmes

Problèmes courants

- Défaillances de connexion — Vérifiez la connectivité réseau IPv6 et la résolution DNS

- Erreurs du SDK : assurez-vous que vous utilisez des versions du SDK compatibles qui prennent en charge les points de terminaison à double pile
- Problèmes d'authentification — Vérifiez que les politiques et les informations d'identification IAM fonctionnent avec les points de terminaison IPv6
- Point de terminaison non disponible : si aucun point de terminaison à double pile n'existe pour le service ou la région, les demandes échouent

Étapes de vérification

- Vérifiez que `AWS_USE_DUALSTACK_ENDPOINT=true` est défini ou que `use_dualstack_endpoint = true` se trouve dans votre fichier de configuration
- Vérifiez que les indicateurs de configuration IPv6 du SDK Kinesis Video Streams sont correctement définis
- Tester la connectivité réseau aux points de terminaison AWS IPv6
- Consultez les journaux des applications pour détecter les messages IPv6-specific d'erreur
- Vérifiez que votre région prend en charge les points de terminaison à double pile pour Kinesis Video Streams

Validation de configuration

Vous pouvez vérifier la configuration de votre point de terminaison à double pile en vérifiant :

- Variables d'environnement : `echo $AWS_USE_DUALSTACK_ENDPOINT`
- AWS fichier de configuration : `cat ~/.aws/config | grep use_dualstack_endpoint`
- Propriétés de la JVM (Java) : vérifiez les propriétés du système dans les journaux de vos applications

Lecture de Kinesis Video Streams

Vous pouvez visionner un flux vidéo Kinesis en utilisant les méthodes suivantes :

- **GetMedia**— Vous pouvez utiliser l'GetMediaAPI pour créer vos propres applications afin de traiter Kinesis Video Streams. GetMediaest une API en temps réel à faible latence. Pour créer un joueur qui l'utiliseGetMedia, vous devez le créer vous-même. Pour plus d'informations sur le développement d'une application qui affiche un flux vidéo Kinesis à l'aide GetMedia de. [Diffuser à l'aide de la bibliothèque d'analyseurs](#)
- **HLS** — [HTTP Live Streaming \(HLS\)](#) est un protocole de communication par streaming HTTP-based multimédia standard. Vous pouvez utiliser le HLS pour visionner un flux vidéo Kinesis, que ce soit pour le visionner en direct ou pour visionner une vidéo archivée.

Vous pouvez utiliser HLS pour la lecture en direct. La latence est généralement comprise entre 3 et 5 secondes, mais elle peut être comprise entre 1 et 10 secondes, selon le cas d'utilisation, le lecteur et les conditions du réseau. Vous pouvez utiliser un lecteur tiers (tel que [Video.js](#)[Google Shaka Player](#)) pour afficher le flux vidéo en fournissant l'URL de la session de streaming HLS, par programmation ou manuellement. Vous pouvez également lire une vidéo en saisissant l'URL de la session de streaming HLS dans la barre d'adresse des navigateurs [Apple Safari](#) ou [Microsoft Edge](#).

- **MPEG-DASH**— Le [Dynamic Adaptive Streaming over HTTP \(DASH\)](#), également connu sous le nom de MPEG-DASH, est un protocole de streaming à débit adaptatif qui permet de diffuser du contenu multimédia de haute qualité sur Internet à partir de serveurs Web HTTP conventionnels.

Vous pouvez l'utiliser MPEG-DASH pour la diffusion en direct. La latence est généralement comprise entre 3 et 5 secondes, mais elle peut être comprise entre 1 et 10 secondes, selon le cas d'utilisation, le lecteur et les conditions du réseau. Vous pouvez utiliser un lecteur tiers (tel que [dash.js](#) ou [Google Shaka Player](#)) pour afficher le flux vidéo en fournissant l'URL de la session de MPEG-DASH diffusion, par programmation ou manuellement.

- **GetClip**— Vous pouvez utiliser l'GetClipAPI pour télécharger un clip (dans un fichier MP4) contenant le contenu multimédia archivé à la demande à partir du flux vidéo spécifié pendant la période spécifiée. Pour plus d'informations, consultez la référence de l'API [GetClip](#).

Rubriques

- [Exigences relatives aux pistes de lecture vidéo](#)

- [Lecture vidéo avec HLS](#)
- [Lecture vidéo avec MPEG-DASH](#)

Exigences relatives aux pistes de lecture vidéo

Amazon Kinesis Video Streams prend en charge le contenu multimédia encodé dans plusieurs formats. Si votre flux vidéo Kinesis utilise un format non pris en charge par l'une des quatre API répertoriées ci-dessous, utilisez [GetMedia](#) ou [GetMediaForFragmentList](#), car elles ne sont pas limitées par type de piste.

Rubriques

- [GetClip exigences](#)
- [GetDASHStreamingSessionURL exigences](#)
- [GetHLSStreamingSessionURL exigences](#)
- [GetImages exigences](#)

GetClip exigences

Pour plus d'informations sur cette API, consultez [GetClip](#).

| Description de la piste 1 | ID du codec Track 1 | Description de la piste 2 | ID du codec Track 2 |
|---------------------------|---------------------|--------------------------------|---------------------|
| H.264 vidéo | V_MPEG4/ISO/AVC | N/A | N/A |
| H.264 vidéo | V_MPEG4/ISO/AVC | Audio AAC | A_AAC |
| H.264 vidéo | V_MPEG4/ISO/AVC | G.711 audio (A-Law uniquement) | A_MS/ACM |
| H.265 vidéo | V_MPEGH/ISO/HEVC | N/A | N/A |
| H.265 vidéo | V_MPEGH/ISO/HEVC | Audio AAC | A_AAC |

⚠ Important

Les données privées du codec (CPD) contenues dans chaque fragment contiennent des informations d'initialisation spécifiques au codec, telles que la fréquence d'images, la résolution et le profil de codage, qui sont nécessaires pour décoder correctement le fragment. Les modifications du CPD ne sont pas prises en charge entre les fragments cibles du clip obtenu. Le CPD doit rester cohérent sur le support demandé, sinon une erreur sera renvoyée.

⚠ Important

Le suivi des modifications n'est pas pris en charge. Les pistes doivent rester cohérentes sur l'ensemble du support demandé. Une erreur est renvoyée si les fragments du flux ne contiennent plus uniquement de la vidéo mais contiennent à la fois du son et de la vidéo, ou si une piste audio AAC est remplacée par une piste A-Law audio.

GetDASHStreamingSessionURL exigences

Pour plus d'informations sur cette API, consultez [GetDASHStreamingSessionURL](#).

| Description de la piste 1 | ID du codec Track 1 | Description de la piste 2 | ID du codec Track 2 |
|---------------------------|---------------------|--------------------------------|---------------------|
| H.264 vidéo | V_MPEG4/ISO/AVC | N/A | N/A |
| H.264 vidéo | V_MPEG4/ISO/AVC | Audio AAC | A_AAC |
| H.264 vidéo | V_MPEG4/ISO/AVC | G.711 audio (A-Law uniquement) | A_MS/ACM |
| H.264 vidéo | V_MPEG4/ISO/AVC | G.711 audio (U-Law uniquement) | A_MS/ACM |
| Audio AAC | A_AAC | N/A | N/A |
| H.265 vidéo | V_MPEGH/ISO/HEVC | N/A | N/A |
| H.265 vidéo | V_MPEGH/ISO/HEVC | Audio AAC | A_AAC |

⚠ Important

Les données privées du codec (CPD) contenues dans chaque fragment contiennent des informations d'initialisation spécifiques au codec, telles que la fréquence d'images, la résolution et le profil de codage, qui sont nécessaires pour décoder correctement le fragment. Les modifications du CPD ne sont pas prises en charge pendant une session de streaming. Le CPD doit rester cohérent à travers les médias interrogés.

⚠ Important

Le suivi des modifications n'est pas pris en charge. Les pistes doivent rester cohérentes sur l'ensemble du support demandé. Le streaming échouera si les fragments du flux ne contiennent plus uniquement de la vidéo mais contiennent à la fois du son et de la vidéo, ou si une piste audio AAC est remplacée par une piste A-Law audio.

GetHLSStreamingSessionURL exigences

Pour plus d'informations sur cette API, consultez [GetHLSStreamingSessionURL](#).

HLS Mp4

| Description de la piste 1 | ID du codec Track 1 | Description de la piste 2 | ID du codec Track 2 |
|---------------------------|---------------------|---------------------------|---------------------|
| H.264 vidéo | V_MPEG4/ISO/AVC | N/A | N/A |
| H.264 vidéo | V_MPEG4/ISO/AVC | Audio AAC | A_AAC |
| Audio AAC | A_AAC | N/A | N/A |
| H.265 vidéo | V_MPEGH/ISO/HEVC | N/A | N/A |
| H.265 vidéo | V_MPEGH/ISO/HEVC | Audio AAC | A_AAC |

HLS TS

| Description de la piste 1 | ID du codec Track 1 | Description de la piste 2 | ID du codec Track 2 |
|---------------------------|---------------------|---------------------------|---------------------|
| H.264 vidéo | V_MPEG4/ISO/AVC | N/A | N/A |
| H.264 vidéo | V_MPEG4/ISO/AVC | Audio AAC | A_AAC |
| Audio AAC | A_AAC | N/A | N/A |

Note

Les données privées du codec (CPD) contenues dans chaque fragment contiennent des informations d'initialisation spécifiques au codec, telles que la fréquence d'images, la résolution et le profil de codage, qui sont nécessaires pour décoder correctement le fragment. Pour TS et MP4, les modifications du CPD sont prises en charge pendant une session de streaming. Par conséquent, les fragments d'une session peuvent avoir des informations différentes dans le CPD sans interrompre la lecture. Pour chaque session de streaming, seules 500 modifications du CPD sont autorisées.

Important

Le suivi des modifications n'est pas pris en charge. Les pistes doivent rester cohérentes sur l'ensemble du support demandé. Le streaming échouera si les fragments du flux ne contiennent plus uniquement de la vidéo mais contiennent à la fois du son et de la vidéo, ou si une piste audio AAC est remplacée par une piste A-Law audio.

GetImages exigences

Pour plus d'informations sur cette API, consultez [GetImages](#).

Note

Le `GetImages` média doit contenir une piste vidéo dans la piste 1.

Lecture vidéo avec HLS

Le [HTTP Live Streaming \(HLS\)](#) est un protocole de communication par streaming HTTP-based multimédia standard. Vous pouvez utiliser le HLS pour visionner un flux vidéo Kinesis, que ce soit pour le visionner en direct ou pour visionner une vidéo archivée.

Vous pouvez utiliser HLS pour la lecture en direct. La latence est généralement comprise entre 3 et 5 secondes, mais elle peut être comprise entre 1 et 10 secondes, selon le cas d'utilisation, le lecteur et les conditions du réseau. Vous pouvez utiliser un lecteur tiers (tel que [Video.js](#) [Google Shaka Player](#)) pour afficher le flux vidéo en fournissant l'URL de la session de streaming HLS, par programmation ou manuellement. Vous pouvez également lire une vidéo en saisissant l'URL de la session de streaming HLS dans la barre d'adresse des navigateurs [Apple Safari](#) ou [Microsoft Edge](#).

Pour visionner un flux vidéo Kinesis à l'aide du protocole HLS, créez d'abord une session de streaming à l'aide de [GetHLSStreamingSessionURL](#). Cette action renvoie une URL (contenant un jeton de session) permettant d'accéder à la session HLS. Vous pouvez alors utiliser cette URL dans un lecteur multimédia ou une application autonome pour afficher le flux.

Important

Les contenus multimédias envoyés à Kinesis Video Streams ne peuvent pas tous être lus via le protocole HLS. Consultez [GetHLSStreamingSessionURL](#) les exigences spécifiques en matière de téléchargement.

Rubriques

- [Utilisez la commande AWS CLI pour récupérer l'URL d'une session de streaming HLS](#)
- [Exemple : utilisez HLS en HTML et JavaScript](#)
- [Résolution des problèmes liés au HLS](#)

Utilisez la commande AWS CLI pour récupérer l'URL d'une session de streaming HLS

La procédure suivante explique comment utiliser le AWS CLI pour générer une URL de session de streaming HLS pour un flux vidéo Kinesis.

Pour les instructions d'installation, consultez le [guide de AWS Command Line Interface l'utilisateur](#). Après l'installation, [configurez le AWS CLI avec les](#) informations d'identification et la région.

Vous pouvez également ouvrir le AWS CloudShell terminal sur lequel ils AWS CLI sont installés et configurés. Pour plus d'informations, consultez le [Guide de l'utilisateur AWS CloudShell](#).

Récupérez le point de terminaison URL HLS pour votre flux vidéo Kinesis.

1. Tapez ce qui suit dans le terminal :

```
aws kinesismvideo get-data-endpoint \  
  --api-name GET_HLS_STREAMING_SESSION_URL \  
  --stream-name YourStreamName
```

Vous recevrez une réponse qui ressemble à ceci :

```
{  
  "DataEndpoint": "https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com"  
}
```

2. Envoyez la demande d'URL de session de streaming HLS à ce point de terminaison renvoyé.

Live

Pour la lecture en direct, la liste de lecture multimédia HLS est continuellement mise à jour avec les derniers médias dès qu'ils sont disponibles. Lorsque vous jouez ce type de session dans un lecteur multimédia, l'interface utilisateur affiche généralement une notification « en direct », sans aucune commande permettant de choisir la position à afficher dans la fenêtre de lecture.

Assurez-vous que vous téléchargez du contenu multimédia sur ce flux lorsque vous exécutez cette commande.

```
aws kinesismvideo-archived-media get-hls-streaming-session-url \  
  --endpoint-url https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com \  
  --stream-name YourStreamName \  
  --playback-mode LIVE
```

Live replay

Pour la rediffusion en direct, la lecture commence à partir d'une heure de début spécifiée. La liste de lecture multimédia HLS est également continuellement mise à jour avec les derniers médias dès qu'ils sont disponibles. La session continuera à inclure les médias nouvellement ingérés jusqu'à son expiration ou jusqu'à l'heure de fin spécifiée, selon la première éventualité. Ce mode est utile pour démarrer la lecture dès qu'un événement est détecté et continuer à diffuser en direct du contenu multimédia qui n'a pas encore été ingéré au moment de la création de la session.

Déterminez un horodatage de début.

Pour cet exemple, nous utilisons le format Unix Epoch en secondes. Reportez-vous à la section [Horodatages](#) du guide de l' AWS Command Line Interface utilisateur pour plus d'informations sur le formatage des horodatages.

Voir [UnixTime.org](https://unixtime.org) pour un outil de conversion.

- 1708471800 est égal au 20 février 2024 à 15 h 30 GMT-08:00

Dans cet exemple, nous ne spécifions pas d'horodatage de fin, ce qui signifie que la session continuera à inclure le contenu multimédia récemment ingéré jusqu'à son expiration.

Appelez l'GetHLSStreamingSessionURLAPI avec le mode LIVE_REPLAY de lecture et un [sélectionneur de fragments HLS](#) spécifié.

```
aws kinesis-video-archived-media get-hls-streaming-session-url \  
  --endpoint-url https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com \  
  --stream-name YourStreamName \  
  --playback-mode LIVE_REPLAY \  
  --hls-fragment-selector \  
  
  "FragmentSelectorType=SERVER_TIMESTAMP, TimestampRange={StartTimestamp=1708471800}"
```

On-demand

Pour la lecture à la demande, la liste de lecture multimédia HLS contient le contenu multimédia spécifié par le sélectionneur de fragments HLS. Lorsque ce type de session est lu

dans un lecteur multimédia, l'interface utilisateur affiche généralement une commande de nettoyage permettant de choisir la position à afficher dans la fenêtre de lecture.

Pour créer une URL pour une section spécifique du flux, déterminez d'abord les horodatages de début et de fin.

Pour cet exemple, nous utilisons le format Unix Epoch en secondes. Reportez-vous à la section [Horodatages](#) du guide de l' AWS Command Line Interface utilisateur pour plus d'informations sur le formatage des horodatages.

Voir [UnixTime.org](#) pour un outil de conversion.

- 1708471800 est égal au 20 février 2024 à 15 h 30 GMT-08:00
- 1708471860 est égal au 20 février 2024 à 15 h 31 GMT-08:00

Appelez l'GetHLSStreamingSessionURLAPI avec le mode ON_DEMAND de lecture et un [sélecteur de fragments HLS](#) spécifié.

```
aws kinesis-video-archived-media get-hls-streaming-session-url \
  --endpoint-url https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com \
  --stream-name YourStreamName \
  --playback-mode ON_DEMAND \
  --hls-fragment-selector \
  "FragmentSelectorType=SERVER_TIMESTAMP, TimestampRange={StartTimestamp=1708471800, EndTime
```

Note

Les horodatages doivent être espacés de 24 heures l'un de l'autre, comme indiqué dans la documentation. [HLSTimestampRange](#)

Vous recevrez une réponse qui ressemble à ceci :

```
{
  "HLSStreamingSessionURL": "https://b-1234abcd.kinesisvideo.aws-
region.amazonaws.com/hls/v1/getHLSMasterPlaylist.m3u8?SessionToken=CiAz...DkRE6M~"
}
```

⚠ Important

Ne partagez pas et ne stockez pas ce jeton dans un endroit où une entité non autorisée pourrait y accéder. Le jeton donne accès au contenu du flux. Protégez le jeton avec les mêmes mesures que celles que vous utiliseriez avec vos AWS informations d'identification.

Vous pouvez utiliser cette URL et n'importe quel lecteur HLS pour afficher le flux HLS.

Par exemple, utilisez le lecteur multimédia VLC.

Vous pouvez également écouter le HLS Stream en saisissant l'URL de la session de streaming HLS dans la barre d'adresse des navigateurs Apple Safari ou Microsoft Edge.

Exemple : utilisez HLS en HTML et JavaScript

L'exemple suivant montre comment utiliser le AWS SDK pour la JavaScript version 2 afin de récupérer une session de streaming HLS pour un flux vidéo Kinesis et de la lire sur une page Web. Cet exemple montre comment lire la vidéo dans les lecteurs suivants :

- [Video.js](#)
- [Google Shaka Player](#)
- [hls.js](#)

Consultez l'[exemple de code complet](#) et la [page Web hébergée](#) dans GitHub. Cette page Web statique simplifie les tests et les expérimentations avec le HLS et la MPEG-DASH sortie du flux vidéo Amazon Kinesis. La page d'exemple fournit des champs de saisie pour les paramètres suivants :

- Région AWS: Région dans laquelle se trouve votre flux vidéo Kinesis
- Nom du flux : nom de votre flux vidéo Kinesis
- Mode de lecture : mode de lecture HLS (LIVE, LIVE_REPLAY ou ON_DEMAND)
- Type de sélecteur de fragment : méthode utilisée pour sélectionner des fragments (SERVER_TIMESTAMP ou PRODUCER_TIMESTAMP)
- Numéro de fragment : numéro de fragment de départ (le cas échéant)

- Format du conteneur : format du conteneur multimédia (FRAGMENTED_MP4 ou MPEG_TS)

L'application récupère ces valeurs dans les zones de saisie de la page HTML et les utilise pour créer la demande d'une session de streaming HLS qui s'affiche sur la page.

Sujets de présentation du code :

- [Importez le AWS SDK JavaScript pour navigateurs](#)
- [Configuration du client Kinesis Video Streams](#)
- [Récupérez le point de terminaison pour la lecture HLS](#)
- [Configuration du client multimédia archivé Kinesis Video Streams](#)
- [Récupérez l'URL de la session de streaming HLS](#)
- [Afficher le flux HLS sur la page Web](#)

Importez le AWS SDK JavaScript pour navigateurs

Dans la page Web, incluez la balise de script suivante pour importer le AWS SDK pour la JavaScript version 2 dans le projet.

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/aws-sdk/2.490.0/aws-sdk.min.js"></script>
```

Pour plus d'informations, reportez-vous à la JavaScript documentation du [AWS SDK](#).

Configuration du client Kinesis Video Streams

Pour accéder au streaming vidéo avec HLS, créez et configurez d'abord le client Kinesis Video Streams. Voir [Configuration des informations d'identification dans un navigateur Web](#) pour d'autres méthodes d'authentification.

```
const clientConfig = {
  accessKeyId: 'YourAccessKey',
  secretAccessKey: 'YourSecretKey',
  region: 'us-west-2'
};
const kinesisVideoClient = new AWS.KinesisVideo(clientConfig);
```

L'application récupère les valeurs nécessaires à partir des zones de saisie sur la page HTML.

Récupérez le point de terminaison pour la lecture HLS

Utilisez le client Kinesis Video Streams pour appeler [GetDataEndpoint](#) l'API afin de récupérer le point de terminaison.

```
const getDataEndpointOptions = {
  StreamName: 'YourStreamName',
  APIName: 'GET_HLS_STREAMING_SESSION_URL'
};
const getDataEndpointResponse = await kinesisVideoClient
  .getDataEndpoint(getDataEndpointOptions)
  .promise();
const hlsDataEndpoint = getDataEndpointResponse.DataEndpoint;
```

Ce code stocke le point de terminaison dans la `hlsDataEndpoint` variable.

Configuration du client multimédia archivé Kinesis Video Streams

Dans la configuration du client multimédia archivé Kinesis Video Streams, spécifiez le point de terminaison que vous avez obtenu à l'étape précédente.

```
const archivedMediaClientConfig = {
  accessKeyId: 'YourAccessKey',
  secretAccessKey: 'YourSecretKey',
  region: 'us-west-2',
  endpoint: hlsDataEndpoint
};
const kinesisVideoArchivedMediaClient = new
  AWS.KinesisVideoArchivedMedia(archivedMediaClientConfig);
```

Récupérez l'URL de la session de streaming HLS

Utilisez le client multimédia archivé Kinesis Video Streams pour appeler [GetHLSStreamingSessionURL](#) l'API afin de récupérer l'URL de lecture HLS.

```
const getHLSStreamingSessionURLOptions = {
  StreamName: 'YourStreamName',
  PlaybackMode: 'LIVE'
};
const getHLSStreamingSessionURLResponse = await kinesisVideoArchivedMediaClient
  .getHLSStreamingSessionURL(getHLSStreamingSessionURLOptions)
```

```
.promise();  
const hlsUrl = getHLSStreamingSessionURLResponse.HLSStreamingSessionURL;
```

Afficher le flux HLS sur la page Web

Lorsque vous disposez de l'URL de la session de streaming HLS, fournissez-la au lecteur vidéo. La méthode pour fournir l'URL au lecteur vidéo est spécifique au lecteur utilisé.

Video.js

Procédez comme suit pour importer [Video.js](#) et ses classes CSS dans le script de notre navigateur :

```
<link rel="stylesheet" href="https://vjs.zencdn.net/6.6.3/video-js.css">  
<script src="https://vjs.zencdn.net/6.6.3/video.js"></script>  
<script src="https://cdnjs.cloudflare.com/ajax/libs/videojs-contrib-hls/5.14.1/  
videojs-contrib-hls.js"></script>
```

Créez un élément video HTML pour afficher la vidéo :

```
<video id="videojs" class="player video-js vjs-default-skin" controls autoplay></  
video>
```

Définissez l'URL HLS comme source de l'élément vidéo HTML :

```
const playerElement = document.getElementById('videojs');  
const player = videojs(playerElement);  
player.src({  
  src: hlsUrl,  
  type: 'application/x-mpegURL'  
});  
player.play();
```

Shaka

Procédez comme suit pour importer le [lecteur Google Shaka](#) dans le script de notre navigateur :

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/shaka-player/2.4.1/shaka-  
player.compiled.js"></script>
```

Créez un élément video HTML pour afficher la vidéo :

```
<video id="shaka" class="player" controls autoplay></video>
```

Créez un lecteur Shaka en spécifiant l'élément vidéo et appelez la méthode de chargement :

```
const playerElement = document.getElementById('shaka');  
const player = new shaka.Player(playerElement);  
player.load(hlsUrl);
```

hls.js

Procédez comme suit pour importer le [fichier hls.js](#) dans le script de votre navigateur :

```
<script src="https://cdn.jsdelivr.net/npm/hls.js@latest"></script>
```

Créez un élément video HTML pour afficher la vidéo :

```
<video id="hlsjs" class="player" controls autoplay></video>
```

Créez un lecteur hls.js, donnez-lui l'URL HLS et dites-lui de jouer :

```
const playerElement = document.getElementById('hlsjs');  
const player = new Hls();  
player.loadSource(hlsUrl);  
player.attachMedia(playerElement);  
player.on(Hls.Events.MANIFEST_PARSED, function() {  
    video.play();  
});
```

Résolution des problèmes liés au HLS

Cette section décrit les problèmes que vous pouvez rencontrer lors de l'utilisation du protocole HTTP Live Streaming (HLS) avec Kinesis Video Streams.

Problèmes

- [Succès de la récupération de l'URL de la session de streaming HLS, mais échec de la lecture dans le lecteur vidéo](#)

- [Latence trop élevée entre producteur et lecteur](#)

Succès de la récupération de l'URL de la session de streaming HLS, mais échec de la lecture dans le lecteur vidéo

Cette situation se produit lorsque l'URL d'une session de streaming HLS peut être récupérée avec succès avec `GetHLSStreamingSessionURL`, mais que la vidéo ne peut pas être lue lorsque l'URL est fournie à un lecteur vidéo.

Pour résoudre ce problème, essayez l'une des actions suivantes :

- Déterminez si le flux vidéo est lu dans la console Kinesis Video Streams. Prenez en compte toutes les erreurs que la console affiche.
- Si la durée de fragment est inférieure à une seconde, augmentez-la à une seconde. Si la durée du fragment est trop courte, le service peut ralentir le lecteur car il demande trop fréquemment des fragments vidéo.
- Vérifiez que chaque URL de session de streaming HLS est utilisée par un seul lecteur. Si plusieurs lecteurs utilisent une même URL de session de streaming HLS, le service peut recevoir un trop grand nombre de demandes et peut les limiter.
- Vérifiez que votre lecteur prend en charge toutes les options que vous spécifiez pour la session de streaming HLS. Essayez des combinaisons différentes de valeurs pour les paramètres suivants :
 - `ContainerFormat`
 - `PlaybackMode`
 - `FragmentSelectorType`
 - `DiscontinuityMode`
 - `MaxMediaPlaylistFragmentResults`

Certains lecteurs multimédias (tels que les lecteurs HTML5 et mobiles) ne prennent généralement en charge que HLS avec le format de conteneur fMP4. Les autres lecteurs multimédias (tels que Flash et les lecteurs personnalisés) peuvent uniquement prendre en charge le HLS avec le format de conteneur MPEG TS. Nous vous recommandons de tester le `ContainerFormat` paramètre pour commencer à résoudre les problèmes.

- Vérifiez que chaque fragment possède un nombre de pistes cohérent. Vérifiez que les fragments du flux ne changent pas entre une piste audio et vidéo et une piste vidéo uniquement. Vérifiez également que les paramètres de l'encodeur (résolution et fréquence d'images) ne changent pas entre les fragments de chaque piste.

Latence trop élevée entre producteur et lecteur

Cette situation se produit lorsque la latence est trop élevée entre le moment où la vidéo est capturée et le moment où elle est lue dans le lecteur vidéo.

La vidéo est relue via HLS par fragment. Par conséquent, la latence ne peut pas être inférieure à la durée du fragment. La latence inclut également le temps nécessaire à la mise en tampon et au transfert des données. Si votre solution nécessite une latence inférieure à une seconde, envisagez d'utiliser l'API `GetMedia` à la place.

Les paramètres suivants peuvent être ajustés pour réduire la latence globale, mais l'ajustement de ces paramètres peut également réduire la qualité de la vidéo ou augmenter le taux de remise en tampon.

- **Durée du fragment** : la durée du fragment est la quantité de vidéo entre les divisions du flux, contrôlée par la fréquence des images-clés générées par l'encodeur vidéo. La valeur recommandée est d'une seconde. Le fait d'avoir une durée de fragment plus courte signifie que moins de temps s'écoule à attendre la fin du fragment avant de transmettre les données vidéo au service. Les fragments plus courts sont également plus rapides à traiter par le service. Toutefois, si la durée de fragment est trop courte, la probabilité augmente que le lecteur viendra à manquer de contenu et devra s'arrêter et mettre en tampon du contenu. Si la durée de fragment est inférieure à 500 millisecondes, le producteur peut créer un trop grand nombre de demandes, obligeant ainsi le service à les limiter.
- **Débit** — Un flux vidéo avec un débit inférieur prend moins de temps à lire, à écrire et à transmettre. Toutefois, un flux vidéo avec une vitesse de transmission plus faible a généralement une plus faible qualité vidéo.
- **Nombre de fragments dans les listes de lecture multimédia** : un lecteur sensible à la latence ne doit charger que les fragments les plus récents d'une liste de lecture multimédia. La plupart des joueurs commencent plutôt par le premier fragment. En réduisant le nombre de fragments dans la liste de lecture, vous réduisez l'intervalle de temps entre les fragments précédents et les nouveaux fragments. Avec une taille de playlist plus petite, il est possible qu'un fragment soit ignoré pendant la lecture si l'ajout de nouveaux fragments à la liste de lecture est retardé ou si le lecteur tarde à obtenir une liste de lecture mise à jour. Nous vous recommandons d'utiliser 3 à 5 fragments et d'utiliser un lecteur configuré pour charger uniquement les derniers fragments d'une playlist.
- **Taille de la mémoire tampon du lecteur** — La plupart des lecteurs vidéo ont une durée de mémoire tampon minimale configurable, généralement de 10 secondes par défaut. Pour la latence la plus faible, vous pouvez définir cette valeur sur 0 seconde. Cependant, cela signifie que le joueur

rebuffera en cas de retard de production de fragments, car le joueur n'aura pas de mémoire tampon pour absorber le retard.

- Lecteur « catch up » : les lecteurs vidéo ne reprennent généralement pas automatiquement la lecture jusqu'au début de la mémoire tampon lorsque celle-ci est pleine, par exemple lorsqu'un fragment retardé entraîne un retard de lecture de fragments. Un lecteur personnalisé peut éviter ce problème en supprimant des trames ou en augmentant la vitesse de lecture (par exemple, à 1,1x) pour rattraper l'avant du tampon. Cela entraîne des saccades ou des accélérations de la lecture lorsque le lecteur rattrape son retard, et une remise en tampon peut être plus fréquente lorsque la taille du tampon est maintenue courte.

Lecture vidéo avec MPEG-DASH

Pour visionner un flux vidéo Kinesis à l'aide de MPEG-DASH, vous devez d'abord créer une session de streaming à l'aide de [GetDASHStreamingSessionURL](#). Cette action renvoie une URL (contenant un jeton de session) permettant d'accéder à la MPEG-DASH session. Vous pouvez alors utiliser cette URL dans un lecteur multimédia ou une application autonome pour afficher le flux.

Un flux vidéo Amazon Kinesis doit répondre aux exigences suivantes pour la diffusion de vidéos via : MPEG-DASH

- Pour connaître les exigences relatives à la lecture de pistes de lecture vidéo en streaming, voir [the section called “GetDASHStreamingSessionURL”](#).
- La conservation des données doit être supérieure à 0.
- La piste vidéo de chaque fragment doit contenir des données privées du codec dans le codage vidéo avancé (AVC) pour le H.264 format et le format HEVC pour le format. H.265 Pour plus d'informations, consultez la [MPEG-4spécification ISO/IEC 14496-15](#). Pour plus d'informations sur l'adaptation des données de flux à un format donné, voir [Indicateurs d'adaptation NAL](#).
- [La piste audio \(le cas échéant\) de chaque fragment doit contenir des données privées du codec au format AAC \(spécification AAC ISO/IEC 13818-7\) ou au format MS Wave.](#)

Exemple : utilisation MPEG-DASH en HTML et JavaScript

L'exemple suivant montre comment récupérer une session de MPEG-DASH streaming pour un flux vidéo Kinesis et la lire sur une page Web. Cet exemple montre comment lire la vidéo dans les lecteurs suivants :

- [Google Shaka Player](#)
- [dash.js](#)

Rubriques

- [Configuration du client MPEG-DASH Kinesis Video Streams pour la lecture](#)
- [Récupérez le point de terminaison MPEG-DASH de contenu archivé Kinesis Video Streams pour le visionner](#)
- [Récupérez l'URL de la session de MPEG-DASH streaming](#)
- [Afficher la vidéo en streaming avec MPEG-DASH lecture](#)
- [Exemple terminé](#)

Configuration du client MPEG-DASH Kinesis Video Streams pour la lecture

Pour accéder au streaming vidéo avec MPEG-DASH, créez et configurez d'abord le client Kinesis Video Streams (pour récupérer le point de terminaison du service) et le client multimédia archivé (pour récupérer MPEG-DASH la session de streaming). L'application récupère les valeurs nécessaires à partir des zones de saisie sur la page HTML.

```
var streamName = $('#streamName').val();

// Step 1: Configure SDK Clients
var options = {
  accessKeyId: $('#accessKeyId').val(),
  secretAccessKey: $('#secretAccessKey').val(),
  sessionToken: $('#sessionToken').val() || undefined,
  region: $('#region').val(),
  endpoint: $('#endpoint').val() || undefined
}
var kinesisVideo = new AWS.KinesisVideo(options);
var kinesisVideoArchivedContent = new AWS.KinesisVideoArchivedMedia(options);
```

Récupérez le point de terminaison MPEG-DASH de contenu archivé Kinesis Video Streams pour le visionner

Une fois les clients initialisés, récupérez le point de terminaison de contenu archivé Kinesis Video Streams afin de pouvoir récupérer MPEG-DASH l'URL de la session de streaming comme suit :

```
// Step 2: Get a data endpoint for the stream
console.log('Fetching data endpoint');
kinesisVideo.getDataEndpoint({
  StreamName: streamName,
  APIName: "GET_DASH_STREAMING_SESSION_URL"
}, function(err, response) {
  if (err) { return console.error(err); }
  console.log('Data endpoint: ' + response.DataEndpoint);
  kinesisVideoArchivedContent.endpoint = new AWS.Endpoint(response.DataEndpoint);
});
```

Récupérez l'URL de la session de MPEG-DASH streaming

Lorsque vous disposez du point de terminaison du contenu archivé, appelez

[l'GetDASHStreamingSessionURL API](#) pour récupérer l'URL de la session de MPEG-DASH streaming comme suit :

```
// Step 3: Get a Streaming Session URL
var consoleInfo = 'Fetching ' + protocol + ' Streaming Session URL';
console.log(consoleInfo);

if (protocol === 'DASH') {
  kinesisVideoArchivedContent.getDASHStreamingSessionURL({
    StreamName: streamName,
    PlaybackMode: $('#playbackMode').val(),
    DASHFragmentSelector: {
      FragmentSelectorType: $('#fragmentSelectorType').val(),
      TimestampRange: $('#playbackMode').val() === "LIVE" ? undefined : {
        StartTimestamp: new Date($('#startTimestamp').val()),
        EndTimestamp: new Date($('#endTimestamp').val())
      }
    },
    DisplayFragmentTimestamp: $('#displayFragmentTimestamp').val(),
    DisplayFragmentNumber: $('#displayFragmentNumber').val(),
    MaxManifestFragmentResults: parseInt($('#maxResults').val()),
    Expires: parseInt($('#expires').val())
  }, function(err, response) {
    if (err) { return console.error(err); }
    console.log('DASH Streaming Session URL: ' + response.DASHStreamingSessionURL);
  });
}
```

Afficher la vidéo en streaming avec MPEG-DASH lecture

Lorsque vous avez l'URL de la session de MPEG-DASH streaming, indiquez-la au lecteur vidéo. La méthode pour fournir l'URL au lecteur vidéo est spécifique au lecteur utilisé.

L'exemple de code suivant montre comment fournir l'URL de la session de streaming à un lecteur

[Google Shaka Player](#) :

```
// Step 4: Give the URL to the video player.

//Shaka Player elements
<video id="shaka" class="player" controls autoplay></video>
<script src="https://cdnjs.cloudflare.com/ajax/libs/shaka-player/2.4.1/shaka-
player.compiled.js">
</script>
...

var playerName = $('#player').val();

if (playerName === 'Shaka Player') {
  var playerElement = $('#shaka');
  playerElement.show();

  var player = new shaka.Player(playerElement[0]);
  console.log('Created Shaka Player');

  player.load(response.DASHStreamingSessionURL).then(function() {
    console.log('Starting playback');
  });
  console.log('Set player source');
}
```

L'exemple de code suivant montre comment fournir l'URL de session de streaming à un lecteur

[dash.js](#) :

```
<!-- dash.js Player elements -->
<video id="dashjs" class="player" controls autoplay=""></video>
<script src="https://cdn.dashjs.org/latest/dash.all.min.js"></script>
...

```

```
var playerElement = $('#dashjs');
playerElement.show();

var player = dashjs.MediaPlayer().create();
console.log('Created DASH.js Player');

player.initialize(document.querySelector('#dashjs'), response.DASHStreamingSessionURL,
  true);
console.log('Starting playback');
console.log('Set player source');
}
```

Exemple terminé

Vous pouvez [télécharger ou consulter l'exemple de code complet](#) sur GitHub.

Configuration des notifications dans Kinesis Video Streams

Lorsqu'un fragment multimédia est disponible pour consommation, Kinesis Video Streams en informe les clients à Amazon Simple Notification Service l'aide de notifications (Amazon SNS).

Note

Amazon Kinesis Video Streams utilise les rubriques standard d'Amazon SNS pour la communication. Les sujets FIFO ne sont pas pris en charge actuellement.

Les rubriques suivantes expliquent comment démarrer avec les notifications.

Rubriques

- [Gérer les configurations de notifications](#)
- [À propos des tags MKV du producteur](#)
- [Messages Amazon SNS](#)
- [Cross-account Publication de notifications Amazon SNS](#)

Gérer les configurations de notifications

Pour gérer les configurations de notification, utilisez `UpdateNotificationConfiguration` et `DescribeNotificationConfiguration`. Voir ci-dessous pour plus d'informations.

UpdateNotificationConfiguration

Utilisez cette opération d'API pour mettre à jour les informations de notification d'un flux. Pour plus d'informations sur `UpdateNotificationConfiguration` cette fonctionnalité, consultez [UpdateNotificationConfiguration](#) le manuel Amazon Kinesis Video Streams Developer Guide.

Note

Le lancement de la notification prend au moins une minute après la mise à jour de la configuration des notifications. Attendez au moins une minute avant d'appeler `PutMedia` après l'appel de mise à jour.

DescribeNotificationConfiguration

Utilisez cette API pour décrire une configuration de notification attachée à un flux. Pour plus d'informations sur DescribeNotificationConfiguration cette fonctionnalité, consultez [DescribeNotificationConfiguration](#) le manuel Amazon Kinesis Video Streams Developer Guide.

À propos des tags MKV du producteur

Vous pouvez utiliser le SDK Kinesis Video Streams Producer pour baliser des fragments spécifiques présentant un intérêt en exposant une opération d'API dans le SDK. Vous trouverez un exemple de ce fonctionnement [dans cette section de code](#). Lors de l'appel de cette API, le SDK ajoutera un ensemble de balises MKV prédéfinies ainsi que les données du fragment. Kinesis Video Streams reconnaîtra ces balises MKV spéciales et lancera des notifications pour les fragments balisés.

Toutes les métadonnées de fragment fournies avec les balises Notification MKV seront publiées dans le cadre de la charge utile de la rubrique Amazon SNS.

Syntaxe pour les tags MKV du producteur

```
|+ Tags
| + Tag
| // MANDATORY: Predefined MKV tag to trigger the notification for the fragment
| + Simple
| + Name: AWS_KINESISVIDEO_NOTIFICATION
| + String
| // OPTIONAL: Key value pairs that will be sent as part of the Notification payload
| + Simple
| + Name: CUSTOM_KEY_1 // Max 128 bytes
| + String: CUSTOM_VALUE_1 // Max 256 bytes
| + Simple
| + Name: CUSTOM_KEY_2 // Max 128 bytes
| + String: CUSTOM_VALUE_2 // Max 256 bytes
```

Limites de balises MKV

Le tableau suivant répertorie les limites associées aux balises de métadonnées. Si la limite de balises de métadonnées est ajustable, vous pouvez demander une augmentation auprès de votre responsable de compte.

| Limite | Valeur maximale | Ajustable |
|--|-----------------|-----------|
| Longueur de clé de métadonnées facultative | 128 | Non |
| Longueur de valeur des métadonnées facultative | 256 | Non |
| Nombre maximum de métadonnées facultatives | 10 | Oui |

Messages Amazon SNS

Cette rubrique contient plus d'informations sur les messages Amazon SNS et les charges utiles.

Rubriques

- [Charge utile de la rubrique Amazon SNS](#)
- [Afficher vos messages Amazon SNS](#)

Charge utile de la rubrique Amazon SNS

Toute notification initiée par le biais du flux de travail précédent fournira la charge utile du sujet Amazon SNS, comme indiqué dans l'exemple suivant. Cet exemple est un message Amazon SNS qui apparaît après avoir consommé les données de notification d'une file d'attente Amazon Simple Queue Service (Amazon SQS).

```
{
  "Type" : "Notification",
  "MessageId" : Message ID,
  "TopicArn" : SNS ARN,
  "Subject" : "Kinesis Video Streams Notification",
  "Message" : "{\"StreamArn\":\\Stream Arn,\\\"FragmentNumber\\\":\\Fragment Number,
  \\\"FragmentStartProducerTimestamp\\\":FragmentStartProducerTimestamp,
    \\\"FragmentStartServerTimestamp\\\":FragmentStartServerTimestamp,
  \\\"NotificationType\\\":\\\"PERSISTED\\\",\\\"NotificationPayload\\\":{\\ CUSTOM_KEY_1:
  \\CUSTOM_VALUE_1,
    \\CUSTOM_KEY_2:\\CUSTOM_VALUE_2}}\",
```

```
"Timestamp" : "2022-04-25T18:36:29.194Z",  
"SignatureVersion" : Signature Version,  
"Signature" : Signature,  
"SigningCertURL" : Signing Cert URL,  
"UnsubscribeURL" : Unsubscribe URL  
}
```

```
Subject: "Kinesis Video Streams Notification"  
Message:  
{  
  "StreamArn":Stream Arn,  
  "FragmentNumber":Fragment Number,  
  "FragmentStartProducerTimestamp":Fragment Start Producer Timestamp,  
  "FragmentStartServerTimestamp":Fragment Start Server Timestamp,  
  "NotificationType": "PERSISTED",  
  "NotificationPayload": {  
    CUSTOM_KEY_1: CUSTOM_VALUE_1,  
    CUSTOM_KEY_2: CUSTOM_VALUE_2  
  }  
}
```

Afficher vos messages Amazon SNS

Vous ne pouvez pas lire les messages directement depuis une rubrique Amazon SNS car aucune API ne permet de le faire. Pour consulter les messages, abonnez une file d'attente SQS à la rubrique SNS ou choisissez une autre destination prise en charge par [Amazon SNS](#). Cependant, l'option la plus efficace pour afficher les messages est d'utiliser Amazon SQS.

Pour consulter vos messages Amazon SNS à l'aide d'Amazon SQS

1. Créez une file d'[attente Amazon SQS](#).
2. À partir de l'AWS Management Console, ouvrez la rubrique Amazon SNS définie comme destination sous. NotificationConfiguration
3. Choisissez Create Subscription, puis choisissez la file d'attente Amazon SQS créée lors de la première étape.
4. Lancez une PutMedia session avec la configuration de notification activée et avec les balises de notification MKV ajoutées aux fragments.
5. Choisissez la file d'attente Amazon SQS dans la console Amazon SQS, puis sélectionnez Envoyer et recevoir des messages pour la file d'attente Amazon SQS.

6. Sondage de messages. Cette commande doit afficher toutes les notifications générées par la PutMedia session. Pour plus d'informations sur le sondage, consultez le sondage [court et long d'Amazon SQS](#).

Cross-account Publication de notifications Amazon SNS

Pour publier des notifications Amazon SNS sur une rubrique d'un autre AWS compte, vous devez configurer des politiques basées à la fois sur l'identité et sur les ressources. Cette configuration permet à Kinesis Video Streams de publier des notifications d'un compte vers une rubrique Amazon SNS dans un autre compte.

Identity-based configuration des politiques

Le rôle ou l'utilisateur IAM qui appelle l'PutMediaAPI doit disposer des `sns:Publish` autorisations nécessaires pour accéder à la rubrique Amazon SNS entre comptes. Ajoutez la déclaration de politique suivante à la stratégie basée sur l'identité :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesisvideo:PutMedia",
      "Resource": "arn:aws:kinesisvideo:us-east-1:123456789012:stream/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:us-east-1:123456789012:*"
      ]
    }
  ]
}
```

Resource-based configuration des politiques

La rubrique Amazon SNS du compte de destination doit disposer d'une politique d'accès basée sur les ressources qui autorise le compte source à publier des messages. Configurez la politique d'accès aux rubriques Amazon SNS comme suit :

JSON

```
{
  "Version": "2012-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:us-east-1:123456789012:topic-name"
    }
  ]
}
```

<kvs_streams_account_id>Remplacez-le par l'ID du AWS compte sur lequel se trouvent vos flux Kinesis Video Streams <sns_topic_arn> et par l'ARN de votre rubrique Amazon SNS.

Exigences et considérations

- La stratégie basée sur l'identité (dans le compte source) et la stratégie basée sur les ressources (dans le compte de destination) doivent être configurées pour que la publication entre comptes fonctionne.
- Le rôle IAM utilisé pour les PutMedia opérations doit inclure des sns:Publish autorisations, même lors de l'utilisation de certificats IoT avec des alias de rôle.

Extraire des images à partir de flux vidéo

Vous pouvez utiliser les API et SDK Amazon Kinesis Video Streams pour effectuer une extraction d'images à la demande et une extraction automatique d'images en temps réel à partir de vos flux vidéo Kinesis. Vous pouvez utiliser ces images pour des applications de lecture améliorées telles que les vignettes ou le nettoyage amélioré, ou pour les utiliser dans des flux de travail d'apprentissage automatique.

Kinesis Video Streams permet d'extraire des images à partir de flux vidéo de deux manières :

- [On-demand Génération d'images](#) : utilisez l'[Get Images](#) API pour extraire une ou plusieurs images d'une vidéo stockée dans Kinesis Video Streams.
- [the section called “Real-time génération d'images”](#) - Configurez Kinesis Video Streams pour extraire automatiquement les images des données vidéo en temps réel en fonction des balises de fragments de la vidéo au fur et à mesure de leur ingestion, et pour envoyer les images dans un compartiment S3.

Génération automatique d'images en temps réel

Amazon Kinesis Video Streams permet de transcoder et de diffuser des images. Kinesis Video Streams extrait automatiquement les images des données vidéo en temps réel et les diffuse dans le compartiment Amazon S3 que vous avez spécifié. La mise en œuvre de l'extraction automatique d'images en temps réel implique les étapes suivantes :

- Création d'un compartiment S3 pour recevoir les images générées.
- Configuration de la propriété [ImageGenerationConfiguration](#) stream qui indique à Kinesis Video Streams comment créer les images et où les envoyer.
- Ajouter des balises de génération d'images : Kinesis Video Streams génère uniquement des images à l'aide de fragments dotés de la balise de génération d'image. Ces balises sont ajoutées lors du téléchargement d'une vidéo à l'aide du SDK Kinesis Video Streams Producer associé à la méthode `putKinesisVideoEventMetadata`

Les procédures suivantes fournissent des instructions pour effectuer chacune de ces étapes.

Si vous utilisez une clé gérée par le client, assurez-vous que le rôle effectuant les PutMedia appels (uploader) dispose des autorisations suivantes qui sont requises pour chiffrer et déchiffrer les données, ainsi que pour accéder au compartiment Amazon S3.

- kms:Encrypt
- kms:GenerateDataKey
- kms:Decrypt
- s3:PutObject

Pour de plus amples informations, veuillez consulter [the section called “Comment démarrer avec le chiffrement côté serveur ?”](#).

Pour configurer la destination des images générées

1. Créez un compartiment de destination S3 dans lequel les images seront envoyées.

Suivez le [guide de l'utilisateur Amazon S3](#) pour créer un compartiment Amazon S3.

Notez l'URI du bucket, dont vous aurez besoin à l'étape suivante lors de la mise à jour de la configuration de génération d'images du flux.

2. Vérifiez que vous les avez AWS CLI installés et configurés. Pour plus d'informations, consultez le [guide de AWS Command Line Interface l'utilisateur de la version 2](#).
3. Créez un nouveau fichier appelé `update-image-generation-input.json` avec le contenu suivant en entrée. Mettez à jour les valeurs de l'espace réservé avec les valeurs que vous souhaitez utiliser. Pour connaître les valeurs maximales et minimales prises en charge, consultez l'[UpdateImageGenerationConfigurationAPI](#).

```
{
  "StreamName": "demo-stream",
  "ImageGenerationConfiguration": {
    "Status": "ENABLED",
    "DestinationConfig": {
      "DestinationRegion": "us-east-1",
      "Uri": "s3://my-bucket-name"
    },
    "SamplingInterval": 200,
    "ImageSelectorType": "PRODUCER_TIMESTAMP",
    "Format": "JPEG",
    "FormatConfig": {
```

```
    "JPEGQuality": "80"  
  },  
  "WidthPixels": 320,  
  "HeightPixels": 240  
}  
}
```

4. Mettez à jour la configuration de génération d'images du flux à l'aide de l'[UpdateImageGenerationConfiguration](#) API et en joignant le fichier JSON en entrée, comme indiqué dans la commande suivante. Notez que le chemin du fichier pointe vers le fichier situé dans le répertoire en cours.

```
aws kinesisvideo update-image-generation-configuration \  
  --cli-input-json file://./update-image-generation-input.json
```

5. En cas de succès, une réponse vide est renvoyée et rien n'est imprimé dans votre terminal.

Note

Le lancement du flux de production d'images prend au moins 1 minute après la mise à jour de la configuration de génération d'images. Patientez au moins 1 minute avant de télécharger la vidéo sur votre stream.

6. Vérifiez les paramètres de configuration. Utilisez le AWS CLI pour appeler l'[DescribeImageGenerationConfiguration](#) API de votre flux.

```
aws kinesisvideo describe-image-generation-configuration \  
  --stream-name "demo-stream"
```

Kinesis Video Streams génère et diffuse des images uniquement pour les fragments dotés de la balise de génération d'image. Toutes les métadonnées de fragment supplémentaires fournies avec les balises de génération d'images Amazon S3 seront enregistrées en tant que métadonnées Amazon S3.

Note

Les balises de génération d'images font référence à des balises de métadonnées de fragments et non à des balises au niveau du flux.

⚠ Important

Les balises de génération d'images sont prises en compte dans le calcul de la limite de balises de métadonnées relatives aux fragments. Pour de plus amples informations, veuillez consulter [the section called “Quotas du service de métadonnées en streaming”](#).

Voici un exemple de ce à quoi ressemble la structure des balises de métadonnées des fragments à l'aide de l'`mkvinfo` utilitaire. La balise de génération d'image est une balise MKV simple avec une clé de `AWS_KINESISVIDEO_IMAGE_GENERATION` et aucune valeur. Pour plus d'informations, consultez [l'exemple de balises vidéo](#) dans la documentation de Matroska.

```
|+ Tags
| + Tag
| // MANDATORY: Predefined MKV tag to trigger image generation for the fragment
| + Simple
| + Name: AWS_KINESISVIDEO_IMAGE_GENERATION

| // OPTIONAL: S3 prefix which will be set as prefix for generated image.
| + Simple
| + Name: AWS_KINESISVIDEO_IMAGE_PREFIX
| + String: image_prefix_in_s3 // 256 bytes max

| // OPTIONAL: Key value pairs that will be persisted as S3 Image object metadata.
| + Simple
| + Name: CUSTOM_KEY_1 // Max 128 bytes
| + String: CUSTOM_VALUE_1 // Max 256 bytes
| + Simple
| + Name: CUSTOM_KEY_2 // Max 128 bytes
| + String: CUSTOM_VALUE_2 // Max 256 bytes
```

Ajouter des balises de génération d'images aux fragments

Kinesis Video Streams génère et diffuse des images uniquement pour les fragments dotés de la balise de génération d'image. Kinesis Video Streams reconnaît ces balises MKV spéciales et lance le flux de production d'images en fonction de la configuration de traitement d'image du flux.

Lorsque vous utilisez le SDK Kinesis Video Streams Producer pour télécharger du contenu multimédia, vous utilisez cette méthode pour ajouter `putKinesisVideoEventMetadata` la

balise de génération d'image à chaque fragment que vous souhaitez baliser. Un nouveau fragment commence lorsqu'il `putFrame` est appelé par un cadre contenant le `keyframe` drapeau.

Si vous mettez en ligne une vidéo préenregistrée, elle peut être mise en ligne à une vitesse différente de celle à laquelle elle a été enregistrée, en fonction de la vitesse de votre réseau. Nous vous recommandons d'utiliser l'horodatage du producteur pour configurer la génération d'images si vous souhaitez générer des images à intervalles réguliers en fonction des horodatages d'origine de la vidéo, et de ne pas utiliser les horodatages du serveur générés en fonction de la fréquence à laquelle Amazon Kinesis Video Streams a reçu votre vidéo.

Pour voir un exemple complet de ce code, consultez l'exemple de [VideoOnlyRealtimeStreamingSample](#) code dans GitHub.

```
// Setup sample frame
memset(frameBuffer, 0x00, frameSize);
frame.frameData = frameBuffer;
frame.version = FRAME_CURRENT_VERSION;
frame.trackId = DEFAULT_VIDEO_TRACK_ID;
frame.duration = HUNDREDS_OF_NANOS_IN_A_SECOND / DEFAULT_FPS_VALUE;
frame.decodingTs = defaultGetTime(); // current time
frame.presentationTs = frame.decodingTs;

Frame eofr = EOFR_FRAME_INITIALIZER;

while(defaultGetTime() > streamStopTime) {
    frame.index = frameIndex;
    frame.flags = fileIndex % DEFAULT_KEY_FRAME_INTERVAL == 0 ? FRAME_FLAG_KEY_FRAME :
FRAME_FLAG_NONE;
    frame.size = sizeof(frameBuffer);

    CHK_STATUS(readFrameData(&frame, frameFilePath));

    // 1. End the previous fragment
    if (frame.flags == FRAME_FLAG_KEY_FRAME && !firstFrame) {
        putKinesisVideoFrame(streamHandle, &eofr);
    }

    // 2. putFrame call
    CHK_STATUS(putKinesisVideoFrame(streamHandle, &frame));

    if (frame.flags == FRAME_FLAG_KEY_FRAME) {
        // 3. Adding the image generation tag
```

```
    CHK_STATUS(putKinesisVideoEventMetadata(streamHandle,
STREAM_EVENT_TYPE_IMAGE_GENERATION, NULL));

    // 4. Adding fragment metadata
    for (n = 1; n <= 5; n++) {
        SNPRINTF(metadataKey, METADATA_MAX_KEY_LENGTH, "SAMPLE_KEY_%d", n);
        SNPRINTF(metadataValue, METADATA_MAX_VALUE_LENGTH, "SAMPLE_VALUE_%d",
frame.index + n);
        CHK_STATUS(putKinesisVideoFragmentMetadata(streamHandle, metadataKey,
metadataValue, FALSE));
    }
}
defaultThreadSleep(frame.duration);

frame.decodingTs += frame.duration;
frame.presentationTs = frame.decodingTs;
frameIndex++;
fileIndex++;
fileIndex = fileIndex % NUMBER_OF_FRAME_FILES;
firstFrame = TRUE;
}

// 5. End the final fragment
putKinesisVideoFrame(streamHandle, &eofr);
```

Les éléments de l'exemple de code pour configurer des exemples de cadres sont expliqués comme suit :

1. Chaque fragment doit se terminer par une fin de fragment (`eofr`). Cette instruction indique que chaque fois qu'une nouvelle image-clé est reçue, signalant le début de l'image suivante, placez un `eofr` avant d'ajouter l'image suivante dans le flux.
2. Insérez l'image actuelle dans le flux.
3. Ajoutez la balise de génération d'image. La `putKinesisVideoEventMetadata` méthode peut être appelée à tout moment après l'`putFrame(keyFrame)` appel et avant le `putFrame(eofr)`. Il ne doit être appelé qu'une seule fois par fragment. Comme chaque fragment ne comportera qu'une seule image-clé, nous l'appelons actuellement pour des raisons de simplicité. La valeur renvoyée pour `putKinesisVideoEventMetadata` est vérifiée pour un code de réussite (0).
4. Ajoutez d'autres métadonnées de fragments personnalisées, que Kinesis Video Streams transformera en métadonnées d'objets Amazon S3.
5. Terminez le dernier fragment de cette session de téléchargement.

Utilisation des exemples pour ajouter des balises de génération d'images

Vous pouvez utiliser le `kvs_gstreamer_audio_video_sample` SDK C++ Producer si vous souhaitez une option de ligne de commande pour ajouter des balises de génération d'images. Activez cette fonctionnalité en ajoutant l'option `-e both` ou, comme indiqué dans l'exemple suivant.

```
./kvs_gstreamer_audio_video_sample stream-name \  
-f video-to-upload.mp4 \  
-e both
```

Pour plus d'informations sur cet exemple d'application, consultez le [producteur Amazon Kinesis Video Streams CPP](#), le [plug-in GStreamer](#) et le fichier JNI README dans [GitHub](#).

Chemin d'accès à l'objet Amazon S3 (image)

Le chemin de l'objet S3 décrit l'emplacement dans le compartiment S3 configuré où l'image générée sera livrée. Il utilise le format suivant :

```
ImagePrefix_AccountID_StreamName_ImageTimecode_RandomID.file-extension
```

Les éléments du chemin de l'objet sont définis comme suit :

- `ImagePrefix`- Valeur de `AWS_KINESISVIDEO_IMAGE_PREFIX` s'il est présent.
- `AccountID`- L'ID du compte AWS identifiant sous lequel le flux est créé.
- `StreamName`- Nom du flux à partir duquel l'image est générée.
- `ImageTimecode`- Code temporel d'époque (en millisecondes) dans le fragment dans lequel l'image est générée.
- `RandomID`- GUID aléatoire.
- `file-extension`- JPG ou PNG selon le format d'image demandé.

Dans cet exemple, le chemin de l'objet pour les images générées sera le suivant :

```
111122223333_demo-stream_16907729324_f20f9add-75e7-4399-a30f-fc7aefb1bab7.jpg
```

Récupération des métadonnées d'image

Vous pouvez utiliser la console S3 ou la CLI pour récupérer les métadonnées des images générées.

Kinesis Video Streams définit le numéro du fragment, l'horodatage du producteur et du serveur, ainsi que les métadonnées du type de contenu de l'image générée, le tout formaté en tant que métadonnées d'objet Amazon S3. Si des balises MKV supplémentaires sont présentes, elles seront également ajoutées en tant que métadonnées d'objet Amazon S3. L'exemple suivant montre comment utiliser la commande API `head-object` d'Amazon S3 pour récupérer les métadonnées de l'objet. La réponse inclut les métadonnées créées par Kinesis Video Streams.

```
aws s3api head-object --bucket my-bucket-name --key 111122223333_demo-stream_1690707290324_f20f9add-7e57-4399-a30f-fc7aefb1bab7.jpg
{
  "AcceptRanges": "bytes",
  "LastModified": "2023-07-30T08:54:51+00:00",
  "ContentLength": 22693,
  "ETag": "\"63e03cb6d57f77e2db984c1d344b1083\"",
  "ContentType": "image/jpeg",
  "ServerSideEncryption": "AES256",
  "Metadata": {
    "aws_kinesisvideo_producer_timestamp": "1690707290324",
    "aws_kinesisvideo_server_timestamp": "1690707289209",
    "aws_kinesisvideo_fragment_number":
"91343852333182036507421233921329142742245756394"
  }
}
```

Pour plus d'informations sur les métadonnées des objets S3, consultez <https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingMetadata.html>.

Recommandations relatives aux URI Amazon S3 pour se protéger contre le ralentissement

Si vous écrivez des milliers d'images sur Amazon S3, il existe un risque de limitation. Pour plus d'informations, consultez la section [Limites des demandes de publication du préfixe S3](#).

Un préfixe Amazon S3 commence par une limite de 3 500 requêtes PUT par seconde, puis augmentera progressivement au fil du temps pour les préfixes uniques. Évitez d'utiliser des dates et des heures comme préfixes Amazon S3. Les données à code temporel auront un impact sur un préfixe à la fois et changeront également régulièrement, invalidant ainsi les précédentes mises à l'échelle des préfixes.

Pour permettre un dimensionnement Amazon S3 plus rapide et cohérent, nous vous recommandons d'ajouter un préfixe aléatoire, tel qu'un code hexadécimal ou un UUID, à l'URI de destination Amazon S3. Par exemple, les préfixes de code hexadécimal répartiront naturellement vos demandes de manière aléatoire entre 16 préfixes différents (un préfixe pour chaque caractère hexadécimal unique), ce qui autorisera 56 000 requêtes PUT par seconde après le redimensionnement automatique d'Amazon S3.

Résolution des problèmes

Images non livrées dans le compartiment Amazon S3

Pour résoudre ce problème, il y a quelques points à prendre en compte :

1. Missing permissions (Autorisations manquantes)
2. La configuration de génération d'image est incorrecte
3. La balise n'a pas été ajoutée au fragment

Missing permissions (Autorisations manquantes)

Si vous utilisez une clé KMS gérée par le client, assurez-vous que le rôle effectuant les PutMedia appels (téléchargeur) dispose des autorisations de chiffrement et de déchiffrement appropriées et qu'il a accès au compartiment Amazon S3, comme suit :

- kms:Encrypt
- kms:GenerateDataKey
- kms:Decrypt
- s3:PutObject

Pour de plus amples informations, veuillez consulter [the section called “Comment démarrer avec le chiffrement côté serveur ?”](#).

Vérifiez la destination

Utilisez le AWS CLI pour appeler l' DescribeImageGenerationConfiguration API de votre flux.

```
aws kinesisvideo describe-image-generation-configuration \  
  --stream-name "demo-stream"
```

Vérifiez le contenu `DestinationConfig` de la réponse et confirmez qu'il semble correct.

Vérifiez que la balise de génération d'image a été ajoutée au fragment

1. Vérifiez que `putKinesisVideoEventMetadata` a réussi.

La `putKinesisVideoEventMetadata` méthode renvoie un code d'état 0 en cas de succès. Nous vous recommandons de vérifier les valeurs renvoyées par les fonctions pour 0. Si un code d'état différent de zéro est renvoyé, convertissez-le en hexadécimal et vérifiez le [the section called "Référence du code d'erreur"](#) pour plus d'informations.


Assurez-vous que les journaux d'erreurs sont activés et examinez-les pour détecter toute autre erreur dans l'application. Passez en revue et comparez le modèle d'appel de soumission de cadres de votre application par rapport à l'implémentation recommandée : [Ajouter des balises de génération d'images aux fragments](#).

2. Vérifiez le fichier MKV généré localement

Vérifiez que le SDK du producteur ou l'exemple d'application ont correctement ajouté les balises.

- a. Définissez la variable d'environnement `KVS_DEBUG_DUMP_DATA_FILE_DIR`. Si cette valeur est définie, le SDK Producer écrit les fichiers multimédia qu'il aurait envoyés à Kinesis Video Streams à l'emplacement spécifié.

```
export KVS_DEBUG_DUMP_DATA_FILE_DIR=/path/to/output/directory
```

 Note


Le SDK ne créera pas de nouveau répertoire si le chemin n'existe pas. Créez le dossier si nécessaire.

- b. Lancez à nouveau l'application. Vous devriez voir `.mkv` des fichiers être écrits dans le répertoire de sortie spécifié.
- c. Vérifiez le contenu pour vous assurer que la balise est présente à l'aide `MKVToolNix` d'un autre logiciel.
 - i. Installation `MKVToolNix` : `brew install mkvtoolnix`
 - ii. Exécutez `MKVToolNix` avec l'un des `.mkv` fichiers du répertoire de sortie.

```
mkvinfo -v ./path/to/video/file
```

- iii. Passez en revue le MKVToolNix résultat. Si la méthode du SDK du `KinesisVideoStream::PutFragmentMetadata` producteur a été invoquée correctement, vous devriez voir la balise MKV suivante.

```
|+ Tags
| + Tag
|   + Simple
|     + Name: AWS_KINESISVIDEO_IMAGE_GENERATION
```

 Note

Les balises appartiennent au cluster qui le précède.

- d. Si la balise MKV n'est pas présente, assurez-vous que la méthode SDK du `KinesisVideoStream::PutEventMetadata` producteur a été appelée avec l'`STREAM_EVENT_TYPE_IMAGE_GENERATION` argument et qu'elle a renvoyé un code de réussite (0).

Accédez à l'analyse vidéo

Cette section contient des informations sur la manière d'accéder à l'analyse vidéo à l'aide de la bibliothèque d'analyseurs et d'Amazon CloudWatch.

Rubriques

- [Utiliser les métadonnées intégrées dans un flux vidéo Kinesis](#)
- [Regardez le résultat des caméras à l'aide de la bibliothèque d'analyseurs](#)
- [Surveillance d'Amazon Kinesis Video Streams](#)
- [Limites de métadonnées de diffusion](#)

Utiliser les métadonnées intégrées dans un flux vidéo Kinesis

Pour utiliser les métadonnées d'un flux vidéo Kinesis, utilisez une implémentation de : `MkvTagProcessor`

```
public interface MkvTagProcessor {
    default void process(MkvTag mkvTag, Optional<FragmentMetadata>
currentFragmentMetadata) {
        throw new NotImplementedException("Default
FragmentMetadataVisitor.MkvTagProcessor");
    }
    default void clear() {
        throw new NotImplementedException("Default
FragmentMetadataVisitor.MkvTagProcessor");
    }
}
```

Cette interface se trouve dans la classe [FragmentMetadataVisitor](#) de la [Regardez le résultat des caméras à l'aide de la bibliothèque d'analyseurs](#).

La classe `FragmentMetadataVisitor` contient une implémentation de `MkvTagProcessor` :

```
public static final class BasicMkvTagProcessor implements
FragmentMetadataVisitor.MkvTagProcessor {
    @Getter
    private List<MkvTag> tags = new ArrayList<>();
}
```

```
@Override
public void process(MkvTag mkvTag, Optional<FragmentMetadata>
currentFragmentMetadata) {
    tags.add(mkvTag);
}

@Override
public void clear() {
    tags.clear();
}
}
```

La classe `KinesisVideoRendererExample` contient un exemple d'utilisation de `BasicMkvTagProcessor`. Dans l'exemple suivant, une `BasicMkvTagProcessor` est ajoutée au lien `MediaProcessingArguments` d'une application :

```
if (renderFragmentMetadata) {
    getMediaProcessingArguments =
    KinesisVideoRendererExample.GetMediaProcessingArguments.create(
        Optional.of(new FragmentMetadataVisitor.BasicMkvTagProcessor()));
}
```

La méthode `BasicMkvTagProcessor.process` est appelée lorsque les métadonnées de fragment arrivent. Vous pouvez récupérer les métadonnées accumulées avec `GetTags`. Pour récupérer un seul élément de métadonnées, appelez `clear` d'abord pour effacer les métadonnées collectées, puis récupérez à nouveau les éléments de métadonnées.

Regardez le résultat des caméras à l'aide de la bibliothèque d'analyseurs

La bibliothèque d'analyseurs de flux vidéo Kinesis est un ensemble d'outils que vous pouvez utiliser dans les applications Java pour consommer les données MKV d'un flux vidéo Kinesis.

La bibliothèque comporte les outils suivants :

- [StreamingMkvReader](#) : Cette classe lit les éléments spécifiés MKV à partir d'un flux vidéo.
- [FragmentMetadataVisitor](#) : Cette classe récupère les métadonnées pour les fragments (éléments média) et contrôle (flux de données individuels contenant des informations média, par exemple l'audio ou les sous-titres).

- [OutputSegmentMerger](#) : Cette classe fusionne les fragments ou les blocs consécutifs dans un flux vidéo.
- [KinesisVideoExample](#): il s'agit d'un exemple d'application qui montre comment utiliser la bibliothèque d'analyseurs de flux vidéo Kinesis.

La bibliothèque comprend également des tests qui montrent la façon dont les outils sont utilisés.

Conditions préalables

Vous devez disposer des éléments suivants pour examiner et utiliser la bibliothèque d'analyseur de flux vidéo Kinesis :

- Un compte Amazon Web Services (AWS). Si vous n'en avez pas déjà un Compte AWS, consultez [the section called “Inscrivez-vous pour un Compte AWS”](#).
- Un environnement de développement intégré (IDE) Java, tel qu'[Eclipse Java Neon](#) ou [JetBrains IntelliJ Idea](#).
- Java 11, tel qu'[Amazon Corretto](#) 11.

Téléchargez le code

Dans cette section, vous téléchargez la bibliothèque Java et le code test, et importez le projet dans votre IDE Java.

Pour les conditions préalables et autres informations concernant cette procédure, consultez [the section called “Diffuser à l'aide de la bibliothèque d'analyseurs”](#).

1. Créez un répertoire et clonez le code source de la bibliothèque à partir du GitHub référentiel (<https://github.com/aws/amazon-kinesis-video-streams-parser-library>).

```
git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. Ouvrez l'IDE Java que vous utilisez (par exemple, [Eclipse](#) ou [IntelliJ IDEA](#)) et importez le projet Apache Maven que vous avez téléchargé :
 - Dans Eclipse : Choisissez Fichier, Importer..., Maven, Projets Maven existants et accédez au dossier `kinesis-video-streams-parser-lib`.
 - Dans IntelliJ Idea : Choisissez Importer. Naviguez vers le fichier `pom.xml` dans la racine du package téléchargé.

Pour plus d'informations, consultez la documentation relative à IDE.

Examinez le code

Dans cette section, vous examinez la bibliothèque Java et le code test et apprenez à utiliser les outils de la bibliothèque dans votre propre code.

La bibliothèque d'analyseurs de flux vidéo Kinesis contient les outils suivants :

- [StreamingMkvReader](#)
- [FragmentMetadataVisitor](#)
- [OutputSegmentMerger](#)
- [KinesisVideoExample](#)

StreamingMkvReader

Cette classe lit les éléments spécifiés MKV à partir d'un flux sans blocage.

L'exemple de code suivant (à partir de `FragmentMetadataVisitorTest`) illustre comment créer et utiliser un `StreamingMkvReader` pour récupérer des objets `MkvElement` à partir d'un flux d'entrée appelé `inputStream` :

```
StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new
InputStreamParserByteSource(inputStream));
while (mkvStreamReader.mightHaveNext()) {
    Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
    if (mkvElement.isPresent()) {
        mkvElement.get().accept(fragmentVisitor);
        ...
    }
}
}
```

FragmentMetadataVisitor

Cette classe récupère les métadonnées des fragments (éléments multimédias) et suit les flux de données individuels contenant des informations multimédia, telles que les données privées du codec, la largeur des pixels ou la hauteur des pixels.

L'exemple de code suivant (à partir du fichier `FragmentMetadataVisitorTest`) illustre comment utiliser `FragmentMetadataVisitor` pour récupérer des données à partir d'un objet `MkvElement` :

```
FragmentMetadataVisitor fragmentVisitor = FragmentMetadataVisitor.create();
    StreamingMkvReader mkvStreamReader =
        StreamingMkvReader.createDefault(new InputStreamParserByteSource(in));
    int segmentCount = 0;
    while(mkvStreamReader.mightHaveNext()) {
        Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
        if (mkvElement.isPresent()) {
            mkvElement.get().accept(fragmentVisitor);
            if
(MkvTypeInfoos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetaData().getTypeInfo()))
{
                MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
                Frame frame =
((MkvValue<Frame>)dataElement.getValueCopy()).getVal();
                MkvTrackMetadata trackMetadata =
fragmentVisitor.getMkvTrackMetadata(frame.getTrackNumber());
                assertTrackAndFragmentInfo(fragmentVisitor, frame, trackMetadata);
            }
            if
(MkvTypeInfoos.SEGMENT.equals(mkvElement.get().getElementMetaData().getTypeInfo())) {
                if (mkvElement.get() instanceof MkvEndMasterElement) {
                    if (segmentCount < continuationTokens.size()) {
                        Optional<String> continuationToken =
fragmentVisitor.getContinuationToken();
                        Assert.assertTrue(continuationToken.isPresent());
                        Assert.assertEquals(continuationTokens.get(segmentCount),
continuationToken.get());
                    }
                    segmentCount++;
                }
            }
        }
    }
}
```

L'exemple précédent présente le modèle de codage suivant :

- Créez un `FragmentMetadataVisitor` pour analyser les données et un [StreamingMkvReader](#) pour fournir les données.
- Pour chaque `MkvElement` dans le flux, testez si ses métadonnées sont de type `SIMPLEBLOCK`.
- Si c'est le cas, récupérez le `MkvDataElement` à partir de `MkvElement`.
- Récupérez le `Frame` (données média) à partir de `MkvDataElement`.
- Récupérez les `MkvTrackMetadata` pour le `Frame` à partir de `FragmentMetadataVisitor`.
- Récupérez et vérifiez les données suivantes à partir du `Frame` et des objets `MkvTrackMetadata` :
 - Le numéro de suivi.
 - La hauteur de l'image en pixels.
 - La largeur de l'image en pixels.
 - L'ID du codec utilisé pour coder l'image.
 - Cette structure est arrivée en ordre. Vérifiez que le numéro de piste de l'image précédente, s'il est présent, est inférieur à celui de l'image actuelle.

Pour utiliser `FragmentMetadataVisitor` dans votre projet, transmettez les objets `MkvElement` au visiteur à l'aide de leur méthode `accept` :

```
mkvElement.get().accept(fragmentVisitor);
```

OutputSegmentMerger

Cette classe fusionne des métadonnées à partir de différentes pistes du flux dans un flux avec un seul segment.

L'exemple de code suivant (à partir du fichier `FragmentMetadataVisitorTest`) illustre comment utiliser `OutputSegmentMerger` pour fusionner les métadonnées de suivi à partir d'une matrice à octets appelée `inputBytes` :

```
FragmentMetadataVisitor fragmentVisitor = FragmentMetadataVisitor.create();

ByteArrayOutputStream outputStream = new ByteArrayOutputStream();

OutputSegmentMerger outputSegmentMerger =
    OutputSegmentMerger.createDefault(outputStream);
```

```
CompositeMkvElementVisitor compositeVisitor =
    new TestCompositeVisitor(fragmentVisitor, outputSegmentMerger);

final InputStream in = TestResourceUtil.getTestInputStream("output_get_media.mkv");

StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new InputStreamParserByteSource(in));

while (mkvStreamReader.mightHaveNext()) {
    Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
    if (mkvElement.isPresent()) {
        mkvElement.get().accept(compositeVisitor);
        if
(MkvTypeInfoos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetadata().getTypeInfo()))
        {
            MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
            Frame frame = ((MkvValue<Frame>) dataElement.getValueCopy()).getVal();
            Assert.assertTrue(frame.getFrameData().limit() > 0);
            MkvTrackMetadata trackMetadata =
fragmentVisitor.getMkvTrackMetadata(frame.getTrackNumber());
            assertTrackAndFragmentInfo(fragmentVisitor, frame, trackMetadata);
        }
    }
}
```

L'exemple précédent présente le modèle de codage suivant :

- Créez un [FragmentMetadataVisitor](#) pour extraire les métadonnées du flux.
- Créer un flux de sortie pour recevoir les métadonnées fusionnées.
- Créez un `OutputSegmentMerger` en passant dans le `ByteArrayOutputStream`.
- Créez un `CompositeMkvElementVisitor` qui contient les deux visiteurs.
- Créez un `InputStream` qui pointe vers le fichier spécifié.
- Fusionnez chaque élément des données d'entrée dans le flux de sortie.

KinesisVideoExample

Il s'agit d'un exemple d'application qui montre comment utiliser la bibliothèque d'analyseur de flux vidéo Kinesis.

Cette classe effectue les opérations suivantes :

- Crée un flux vidéo Kinesis. S'il existe déjà un flux portant ce nom, le flux est supprimé et recréé.
- Appels [PutMedia](#) pour diffuser des fragments vidéo dans le flux vidéo Kinesis.
- Appels [GetMedia](#) pour diffuser des fragments vidéo à partir du flux vidéo Kinesis.
- Elle utilise un [StreamingMkvReader](#) pour analyser les fragments renvoyés sur le flux et utilise une [FragmentMetadataVisitor](#) pour consigner les fragments.

Supprimer et recréer le flux

L'exemple de code suivant (extrait du `StreamOps.java` fichier) supprime un flux vidéo Kinesis donné :

```
//Delete the stream
amazonKinesisVideo.deleteStream(new
    DeleteStreamRequest().withStreamARN(streamInfo.get().getStreamARN()));
```

L'exemple de code suivant (extrait du `StreamOps.java` fichier) crée un flux vidéo Kinesis portant le nom spécifié :

```
amazonKinesisVideo.createStream(new CreateStreamRequest().withStreamName(streamName)
    .withDataRetentionInHours(DATA_RETENTION_IN_HOURS)
    .withMediaType("video/h264"));
```

Appel PutMedia

L'exemple de code suivant (extrait du `PutMediaWorker.java` fichier) appelle [PutMedia](#) le flux :

```
putMedia.putMedia(new PutMediaRequest().withStreamName(streamName)
    .withFragmentTimecodeType(FragmentTimecodeType.RELATIVE)
    .withProducerStartTimestamp(new Date())
    .withPayload(inputStream), new PutMediaAckResponseHandler() {
    ...
});
```

Appel GetMedia

L'exemple de code suivant (extrait du `GetMediaWorker.java` fichier) appelle [GetMedia](#) le flux :

```
GetMediaResult result = videoMedia.getMedia(new  
    GetMediaRequest().withStreamName(streamName).withStartSelector(startSelector));
```

Analyser le résultat GetMedia

Cette section décrit comment utiliser [StreamingMkvReader](#), [FragmentMetadataVisitor](#) et [CompositeMkvElementVisitor](#) pour analyser, enregistrer dans un fichier et consigner les données renvoyées par GetMedia.

Lisez le résultat de GetMedia with StreamingMkvReader

L'exemple de code suivant (extrait du `GetMediaWorker.java` fichier) crée un [StreamingMkvReader](#) et l'utilise pour analyser le résultat de l'[GetMedia](#) opération :

```
StreamingMkvReader mkvStreamReader = StreamingMkvReader.createDefault(new  
    InputStreamParserByteSource(result.getPayload()));  
log.info("StreamingMkvReader created for stream {}", streamName);  
try {  
    mkvStreamReader.apply(this.elementVisitor);  
} catch (MkvElementVisitException e) {  
    log.error("Exception while accepting visitor {}", e);  
}
```

Dans l'exemple de code précédent, l'élément [StreamingMkvReader](#) extrait des objets `MkvElement` à partir de la charge utile du résultat de GetMedia. Dans la section suivante, les éléments sont transmis à un élément [FragmentMetadataVisitor](#).

Récupérez des fragments avec [FragmentMetadataVisitor](#)

Les exemples de code suivants (en provenance des fichiers `KinesisVideoExample.java` et `StreamingMkvReader.java` files) créent un élément [FragmentMetadataVisitor](#). Les objets `MkvElement` itérés par l'élément [StreamingMkvReader](#) sont ensuite transmis au visiteur à l'aide de la méthode `accept`.

À partir de `KinesisVideoExample.java` :

```
FragmentMetadataVisitor fragmentMetadataVisitor = FragmentMetadataVisitor.create();
```

À partir de *StreamingMkvReader.java* :

```
if (mkvElementOptional.isPresent()) {
    //Apply the MkvElement to the visitor
    mkvElementOptional.get().accept(elementVisitor);
}
```

Consigner les éléments et les écrire dans un fichier

L'exemple de code suivant (en provenance du fichier *KinesisVideoExample.java*) crée les objets suivants et les renvoie dans le cadre de la valeur renvoyée par la fonction *GetMediaProcessingArguments* :

- Un élément *LogVisitor* (une extension de *MkvElementVisitor*) qui écrit des données dans le journal système.
- Un élément *OutputStream* qui copie les données entrantes dans un fichier MKV.
- Un élément *BufferedOutputStream* qui met en mémoire tampon les données de *OutputStream*.
- Un élément [the section called "OutputSegmentMerger"](#) qui fusionne les éléments consécutifs dans le résultat de *GetMedia* avec les mêmes données de suivi et EBML.
- Un *CompositeMkvElementVisitor* qui compose le [FragmentMetadataVisitor](#) [the section called "OutputSegmentMerger"](#), et *LogVisitor* en un seul élément visiteur.

```
//A visitor used to log as the GetMedia stream is processed.
LogVisitor logVisitor = new LogVisitor(fragmentMetadataVisitor);

//An OutputSegmentMerger to combine multiple segments that share track and ebml
metadata into one
//mkv segment.
OutputStream fileOutputStream =
Files.newOutputStream(Paths.get("kinesis_video_example_merged_output2.mkv"),
    StandardOpenOption.WRITE, StandardOpenOption.CREATE);
BufferedOutputStream outputStream = new BufferedOutputStream(fileOutputStream);
OutputSegmentMerger outputSegmentMerger =
OutputSegmentMerger.createDefault(outputStream);

//A composite visitor to encapsulate the three visitors.
CompositeMkvElementVisitor mkvElementVisitor =
```

```
new CompositeMkvElementVisitor(fragmentMetadataVisitor,  
outputSegmentMerger, logVisitor);  
  
return new GetMediaProcessingArguments(outputStream, logVisitor,  
mkvElementVisitor);
```

Les arguments du traitement multimédia sont ensuite transmis au `GetMediaWorker`, qui est à son tour transmis au `ExecutorService`, qui exécute le travailleur sur un thread distinct :

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),  
getCredentialsProvider(),  
getStreamName(),  
new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),  
amazonKinesisVideo,  
getMediaProcessingArgumentsLocal.getMkvElementVisitor());  
executorService.submit(getMediaWorker);
```

Exécutez le code

La bibliothèque d'analyseurs de flux vidéo Kinesis contient des outils que vous pouvez utiliser dans vos propres projets. Le projet contient des tests d'unité pour les outils que vous pouvez exécuter pour vérifier votre installation.

Les tests d'unité suivants sont inclus dans la bibliothèque :

- `mkv`
 - `ElementSizeAndOffsetVisitorTest`
 - `MkvValueTest`
 - `StreamingMkvReaderTest`
- `utilitaires`
 - `FragmentMetadataVisitorTest`
 - `OutputSegmentMergerTest`

Surveillance d'Amazon Kinesis Video Streams

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances d'Amazon Kinesis Video Streams et de AWS vos solutions. Nous vous recommandons

de collecter des données de surveillance provenant de toutes les parties de votre AWS solution afin de vous aider à corriger une défaillance multipoint, le cas échéant. Avant de commencer à surveiller Amazon Kinesis Video Streams, nous vous recommandons de créer un plan de surveillance incluant les réponses aux questions suivantes :

- Quels sont les objectifs de la surveillance ?
- Quelles sont les ressources à surveiller ?
- À quelle fréquence les ressources doivent-elles être surveillées ?
- Quels outils de surveillance utiliser ?
- Qui exécute les tâches de supervision ?
- Qui doit être informé en cas de problème ?

Après avoir défini vos objectifs de surveillance et créé votre plan de surveillance, l'étape suivante consiste à établir une base de référence pour les performances normales d'Amazon Kinesis Video Streams dans votre environnement. Vous devez mesurer les performances d'Amazon Kinesis Video Streams à différents moments et dans différentes conditions de charge. Lorsque vous surveillez Amazon Kinesis Video Streams, enregistrez l'historique des données de surveillance que vous avez collectées. Vous pouvez comparer les performances actuelles d'Amazon Kinesis Video Streams à ces données historiques pour vous aider à identifier les modèles de performances normaux et les anomalies de performances, et à concevoir des méthodes pour résoudre les problèmes susceptibles de survenir.

Rubriques

- [Surveillez les statistiques d'Amazon Kinesis Video Streams avec CloudWatch](#)
- [Surveillez l'agent Amazon Kinesis Video Streams Edge avec CloudWatch](#)
- [Enregistrez les appels d'API Amazon Kinesis Video Streams avec AWS CloudTrail](#)

Surveillez les statistiques d'Amazon Kinesis Video Streams avec CloudWatch

Vous pouvez surveiller un flux vidéo Kinesis à l'aide d'Amazon CloudWatch, qui collecte et traite les données brutes d'Amazon Kinesis Video Streams pour en faire des indicateurs lisibles en temps quasi réel. Ces statistiques sont enregistrées pendant une période de 15 mois afin que vous puissiez accéder aux informations historiques et avoir une meilleure idée des performances de votre application ou service Web.

Dans la console [Amazon Kinesis Video Streams](#), vous pouvez CloudWatch consulter les statistiques d'un flux vidéo Amazon Kinesis de deux manières :


- Sur la page Tableau de bord, choisissez l'onglet Streams vidéo dans la Account-level section Statistiques de la région actuelle.
- Choisissez l'onglet Monitoring (Surveillance) dans la page de détails du flux vidéo.

Amazon Kinesis Video Streams fournit les statistiques suivantes :


| Métrique | Description |
|------------------------------------|---|
| ArchivedFragmentsConsumed.Media | <p>Nombre de points de quota de fragments multimédias consommés par toutes les API. Pour une explication du concept de points de quota, voir the section called “Fragment-metadata et quotas de médias fragmentés”.</p> <p>Unités : nombre</p> |
| ArchivedFragmentsConsumed.Metadata | <p>Nombre de points de quota de métadonnées de fragments consommés par toutes les API. Pour une explication du concept de points de quota, voir the section called “Fragment-metadata et quotas de médias fragmentés”.</p> <p>Unités : nombre</p> |
| PutMedia.Requests | <p>Le nombre de demandes d'PutMediaAPI pour un flux donné.</p> <p>Unités : nombre</p> |
| PutMedia.IncomingBytes | <p>Le nombre d'octets reçus dans le PutMedia cadre du flux.</p> <p>Unités : octets</p> |
| PutMedia.IncomingFragments | <p>Le nombre de fragments complets reçus dans le PutMedia cadre du flux.</p> |


| Métrique | Description |
|--|---|
| | Unités : nombre |
| <code>PutMedia.IncomingFrames</code> | Le nombre d'images complètes reçues dans le <code>PutMedia</code> cadre du flux. Unités : nombre |
| <code>PutMedia.ActiveConnections</code> | Nombre total de connexions à l'hôte de service. Unités : nombre |
| <code>PutMedia.ConnectionErrors</code> | Les erreurs lors de l'établissement de la <code>PutMedia</code> connexion pour le flux. Unités : nombre |
| <code>PutMedia.FragmentIngestionLatency</code> | Décalage horaire entre le moment où le premier et le dernier octet d'un fragment sont reçus par Amazon Kinesis Video Streams. Unités : millisecondes |
| <code>PutMedia.FragmentPersistLatency</code> | Temps écoulé entre la réception et l'archivage de l'intégralité des données du fragment. Unités : nombre |
| <code>PutMedia.Latency</code> | La différence de temps entre la demande et la réponse HTTP <code>InletService</code> pendant l'établissement de la connexion. Unités : nombre |
| <code>PutMedia.BufferingAckLatency</code> | Décalage horaire entre le moment où le premier octet d'un nouveau fragment est reçu par Amazon Kinesis Video Streams et le moment où le <code>Buffering ACK</code> est envoyé pour le fragment. Unités : millisecondes |


| Métrique | Description |
|---|---|
| <code>PutMedia.ReceivedAckLatency</code> | <p>Décalage horaire entre le moment où le dernier octet d'un nouveau fragment est reçu par Amazon Kinesis Video Streams et le moment où l'ACK reçu est envoyé pour le fragment.</p> <p>Unités : millisecondes</p> |
| <code>PutMedia.PersistedAckLatency</code> | <p>Décalage horaire entre le moment où le dernier octet d'un nouveau fragment est reçu par Amazon Kinesis Video Streams et le moment où le Persisted ACK est envoyé pour le fragment.</p> <p>Unités : millisecondes</p> |
| <code>PutMedia.ErrorAckCount</code> | <p>Le nombre d'erreurs ACK envoyés lors de l'exécution du flux.</p> <p>Unités : nombre</p> |
| <code>PutMedia.Success</code> | <p>1 pour chaque fragment écrit avec succès ; 0 pour chaque fragment ayant échoué. La valeur moyenne de cette métrique indique le nombre de fragments valides complets envoyé.</p> <p>Unités : nombre</p> |
| <code>GetMedia.Requests</code> | <p>Le nombre de demandes d'GetMediaAPI pour un flux donné.</p> <p>Unités : nombre</p> |
| <code>GetMedia.OutgoingBytes</code> | <p>Nombre total d'octets envoyés par le service dans le cadre de l'GetMediaAPI pour un flux donné.</p> <p>Unités : octets</p> |


| Métrique | Description |
|---|---|
| <code>GetMedia.OutgoingFragments</code> | <p>Le nombre de fragments envoyés lors de l'GetMedia exécution du flux.</p> <p>Unités : nombre</p> |
| <code>GetMedia.OutgoingFrames</code> | <p>Le nombre de trames envoyées pendant GetMedia le flux donné.</p> <p>Unités : nombre</p> |
| <code>GetMedia.MillisBehindNow</code> | <p>La différence de temps entre l'horodatage actuel du serveur et l'horodatage du dernier fragment envoyé.</p> <p>Unités : millisecondes</p> |
| <code>GetMedia.ConnectionErrors</code> | <p>Nombre de connexions qui n'ont pas été correctement établies.</p> <p>Unités : nombre</p> |
| <code>GetMedia.Success</code> | <p>1 pour chaque fragment envoyé avec succès ; 0 pour chaque échec. La valeur moyenne indique le taux de réussite.</p> <div data-bbox="748 1230 1508 1688"><p> Note</p><p>Les échecs incluent à la fois 400 erreurs (utilisateur) et 500 erreurs (système). Pour plus d'informations sur l'activation d'un résumé des demandes et des réponses, y compris les identifiants des AWS demandes, consultez la section Enregistrement des Request/Response résumés.</p></div> <p>Unités : nombre</p> |


| Métrique | Description |
|--|---|
| <code>GetMediaForFragmentList.OutgoingBytes</code> | <p>Nombre total d'octets envoyés par le service dans le cadre de l'<code>GetMediaForFragmentList</code> API pour un flux donné.</p> <p>Unités : octets</p> |
| <code>GetMediaForFragmentList.OutgoingFragments</code> | <p>Nombre total de fragments envoyés par le service dans le cadre de l'<code>GetMediaForFragmentList</code> API pour un flux donné.</p> <p>Unités : nombre</p> |
| <code>GetMediaForFragmentList.OutgoingFrames</code> | <p>Le nombre total de trames envoyées par le service dans le cadre de l'<code>GetMediaForFragmentList</code> API pour un flux donné.</p> <p>Unités : nombre</p> |
| <code>GetMediaForFragmentList.Requests</code> | <p>Le nombre de demandes d'<code>GetMediaForFragmentList</code> API pour un flux donné.</p> <p>Unités : nombre</p> |

| Métrique | Description |
|--|---|
| <code>GetMediaForFragmentList.Success</code> | <p>1 pour chaque fragment envoyé avec succès ; 0 pour chaque échec. La valeur moyenne indique le taux de réussite.</p> <div data-bbox="748 401 1507 856"><p> Note</p><p>Les échecs incluent à la fois 400 erreurs (utilisateur) et 500 erreurs (système). Pour plus d'informations sur l'activation d'un résumé des demandes et des réponses, y compris les identifiants des AWS demandes, consultez la section Enregistrement des Request/Response résumés.</p></div> <p>Unités : nombre</p> |
| <code>ListFragments.Latency</code> | <p>La latence de l'<code>ListFragments</code> API appelle le nom de flux donné.</p> <p>Unités : millisecondes</p> |
| <code>ListFragments.Requests</code> | <p>Le nombre de demandes d'<code>ListFragments</code> API pour un flux donné.</p> <p>Unités : nombre</p> |


| Métrique | Description |
|------------------------------------|--|
| ListFragments.Success | <p>1 pour chaque demande ayant abouti ; 0 pour chaque échec. La valeur moyenne indique le taux de réussite.</p> <div data-bbox="748 401 1508 856"><p> Note</p><p>Les échecs incluent à la fois 400 erreurs (utilisateur) et 500 erreurs (système). Pour plus d'informations sur l'activation d'un résumé des demandes et des réponses, y compris les identifiants des AWS demandes, consultez la section Enregistrement des Request/Response résumés.</p></div> <p>Unités : nombre</p> |
| GetHLSStreamingSessionURL.Latency | <p>La latence de l'GetHLSStreamingSessionURL API appelle le nom de flux donné.</p> <p>Unités : millisecondes</p> |
| GetHLSStreamingSessionURL.Requests | <p>Le nombre de demandes d'GetHLSStreamingSessionURL API pour un flux donné.</p> <p>Unités : nombre</p> |


| Métrique | Description |
|--|--|
| <code>GetHLSStreamingSessionURL.Success</code> | <p>1 pour chaque demande ayant abouti ; 0 pour chaque échec. La valeur moyenne indique le taux de réussite.</p> <div data-bbox="748 401 1507 856"><p> Note</p><p>Les échecs incluent à la fois 400 erreurs (utilisateur) et 500 erreurs (système). Pour plus d'informations sur l'activation d'un résumé des demandes et des réponses, y compris les identifiants des AWS demandes, consultez la section Enregistrement des Request/Response résumés.</p></div> <p>Unités : nombre</p> |
| <code>GetHLSMasterPlaylist.Latency</code> | <p>La latence de l'GetHLSMasterPlaylist API appelle le nom de flux donné.</p> <p>Unités : millisecondes</p> |
| <code>GetHLSMasterPlaylist.Requests</code> | <p>Le nombre de demandes d'GetHLSMasterPlaylist API pour un flux donné.</p> <p>Unités : nombre</p> |


| Métrique | Description |
|---|---|
| <code>GetHLSMasterPlaylist.Success</code> | <p>1 pour chaque demande ayant abouti ; 0 pour chaque échec. La valeur moyenne indique le taux de réussite.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Les échecs incluent à la fois 400 erreurs (utilisateur) et 500 erreurs (système). Pour plus d'informations sur l'activation d'un résumé des demandes et des réponses, y compris les identifiants des AWS demandes, consultez la section Enregistrement des Request/Response résumés.</p> </div> <p>Unités : nombre</p> |
| <code>GetHLSMediaPlaylist.Latency</code> | <p>La latence de l'GetHLSMediaPlaylist API appelle le nom de flux donné.</p> <p>Unités : millisecondes</p> |
| <code>GetHLSMediaPlaylist.Requests</code> | <p>Le nombre de demandes d'GetHLSMediaPlaylis t API pour un flux donné.</p> <p>Unités : nombre</p> |


| Métrique | Description |
|--|---|
| <code>GetHLSMediaPlaylist.Success</code> | <p>1 pour chaque demande ayant abouti ; 0 pour chaque échec. La valeur moyenne indique le taux de réussite.</p> <div data-bbox="748 401 1507 856" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Les échecs incluent à la fois 400 erreurs (utilisateur) et 500 erreurs (système). Pour plus d'informations sur l'activation d'un résumé des demandes et des réponses, y compris les identifiants des AWS demandes, consultez la section Enregistrement des Request/Response résumés.</p></div> <p>Unités : nombre</p> |
| <code>GetMP4InitFragment.Latency</code> | <p>La latence de l'GetMP4InitFragment API appelle le nom de flux donné.</p> <p>Unités : millisecondes</p> |
| <code>GetMP4InitFragment.Requests</code> | <p>Le nombre de demandes d'GetMP4InitFragment API pour un flux donné.</p> <p>Unités : nombre</p> |

| Métrique | Description |
|---|--|
| <code>GetMP4InitFragment.Success</code> | <p>1 pour chaque demande ayant abouti ; 0 pour chaque échec. La valeur moyenne indique le taux de réussite.</p> <div data-bbox="748 401 1508 856"><p> Note</p><p>Les échecs incluent à la fois 400 erreurs (utilisateur) et 500 erreurs (système). Pour plus d'informations sur l'activation d'un résumé des demandes et des réponses, y compris les identifiants des AWS demandes, consultez la section Enregistrement des Request/Response résumés.</p></div> <p>Unités : nombre</p> |
| <code>GetMP4MediaFragment.Latency</code> | <p>La latence de l'GetMP4MediaFragment API appelle le nom de flux donné.</p> <p>Unités : millisecondes</p> |
| <code>GetMP4MediaFragment.Requests</code> | <p>Le nombre de demandes d'GetMP4MediaFragment API pour un flux donné.</p> <p>Unités : nombre</p> |

| Métrique | Description |
|--|---|
| <code>GetMP4MediaFragment.Success</code> | <p>1 pour chaque demande ayant abouti ; 0 pour chaque échec. La valeur moyenne indique le taux de réussite.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>Les échecs incluent à la fois 400 erreurs (utilisateur) et 500 erreurs (système). Pour plus d'informations sur l'activation d'un résumé des demandes et des réponses, y compris les identifiants des AWS demandes, consultez la section Enregistrement des Request/Response résumés.</p> </div> <p>Unités : nombre</p> |
| <code>GetMP4MediaFragment.OutgoingBytes</code> | <p>Nombre total d'octets envoyés par le service dans le cadre de l'GetMP4MediaFragment API pour un flux donné.</p> <p>Unités : octets</p> |
| <code>GetTSFragment.Latency</code> | <p>La latence de l'GetTSFragment API appelle le nom de flux donné.</p> <p>Unités : millisecondes</p> |
| <code>GetTSFragment.Requests</code> | <p>Le nombre de demandes d'GetTSFragment API pour un flux donné.</p> <p>Unités : nombre</p> |

| Métrique | Description |
|--|--|
| <code>GetTSFfragment.Success</code> | <p>1 pour chaque demande ayant abouti ; 0 pour chaque échec. La valeur moyenne indique le taux de réussite.</p> <div data-bbox="748 401 1507 856"><p> Note</p><p>Les échecs incluent à la fois 400 erreurs (utilisateur) et 500 erreurs (système). Pour plus d'informations sur l'activation d'un résumé des demandes et des réponses, y compris les identifiants des AWS demandes, consultez la section Enregistrement des Request/Response résumés.</p></div> <p>Unités : nombre</p> |
| <code>GetTSFfragment.OutgoingBytes</code> | <p>Nombre total d'octets envoyés par le service dans le cadre de l'GetTSFfragment API pour un flux donné.</p> <p>Unités : octets</p> |
| <code>GetDASHStreamingSessionURL.Latency</code> | <p>La latence de l'GetDASHStreamingSessionURL API appelle le nom de flux donné.</p> <p>Unités : millisecondes</p> |
| <code>GetDASHStreamingSessionURL.Requests</code> | <p>Le nombre de demandes d'GetDASHStreamingSessionURL API pour un flux donné.</p> <p>Unités : nombre</p> |

| Métrique | Description |
|--|--|
| GetDASHStreamingSessionURL. Success | <p>1 pour chaque demande ayant abouti ; 0 pour chaque échec. La valeur moyenne indique le taux de réussite.</p> <div data-bbox="748 401 1507 856"><p> Note</p><p>Les échecs incluent à la fois 400 erreurs (utilisateur) et 500 erreurs (système). Pour plus d'informations sur l'activation d'un résumé des demandes et des réponses, y compris les identifiants des AWS demandes, consultez la section Enregistrement des Request/Response résumés.</p></div> <p>Unités : nombre</p> |
| GetDASHManifest.Latency | <p>La latence de l'GetDASHManifest API appelle le nom de flux donné.</p> <p>Unités : millisecondes</p> |
| GetDASHManifest.Requests | <p>Le nombre de demandes d'GetDASHManifest API pour un flux donné.</p> <p>Unités : nombre</p> |

| Métrique | Description |
|-------------------------|--|
| GetDASHManifest.Success | <p>1 pour chaque demande ayant abouti ; 0 pour chaque échec. La valeur moyenne indique le taux de réussite.</p> <div data-bbox="748 401 1508 856"><p> Note</p><p>Les échecs incluent à la fois 400 erreurs (utilisateur) et 500 erreurs (système). Pour plus d'informations sur l'activation d'un résumé des demandes et des réponses, y compris les identifiants des AWS demandes, consultez la section Enregistrement des Request/Response résumés.</p></div> <p>Unités : nombre</p> |
| GetClip.Latency | <p>La latence de l' GetClip API nécessite le nom du flux vidéo donné.</p> <p>Unités : millisecondes</p> |
| GetClip.Requests | <p>Le nombre de demandes d' GetClip API pour un flux vidéo donné.</p> <p>Unités : nombre</p> |

| Métrique | Description |
|-----------------------|---|
| GetClip.Success | <p>1 pour chaque demande ayant abouti ; 0 pour chaque échec. La valeur moyenne indique le taux de réussite.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>Note</p> <p>Les échecs incluent à la fois 400 erreurs (utilisateur) et 500 erreurs (système). Pour plus d'informations sur l'activation d'un résumé des demandes et des réponses, y compris les identifiants des AWS demandes, consultez la section Enregistrement des Request/Response résumés.</p> </div> <p>Unités : nombre</p> |
| GetClip.OutgoingBytes | <p>Nombre total d'octets envoyés par le service dans le cadre de l' GetClip API pour un flux vidéo donné.</p> <p>Unités : octets</p> |

CloudWatch conseils sur les métriques

CloudWatch les métriques peuvent aider à trouver des réponses aux questions suivantes :

Rubriques

- [Les données parviennent-elles au service Amazon Kinesis Video Streams ?](#)
- [Pourquoi les données ne sont-elles pas correctement ingérées par le service Amazon Kinesis Video Streams ?](#)
- [Pourquoi les données ne peuvent-elles pas être lues à partir du service Amazon Kinesis Video Streams au même rythme que celles envoyées par le producteur ?](#)
- [Pourquoi la console ne lit-elle aucune vidéo, ou pourquoi la vidéo est-elle lue avec un retard ?](#)

- [Quel est le retard dans la lecture des données en temps réel et pourquoi le client est-il en retard par rapport à la tête du flux ?](#)
- [Le client lise-t-il les données du flux vidéo Kinesis, et à quel rythme ?](#)
- [Pourquoi le client ne peut-il pas lire les données du flux vidéo Kinesis ?](#)

Les données parviennent-elles au service Amazon Kinesis Video Streams ?

Métriques pertinentes :

- `PutMedia.IncomingBytes`
- `PutMedia.IncomingFragments`
- `PutMedia.IncomingFrames`

Éléments d'action :

- En cas de baisse de ces statistiques, vérifiez si votre application envoie toujours des données au service.
- Vérifiez la bande passante réseau. Si votre bande passante réseau est insuffisante, cela peut ralentir la vitesse de réception des données par le service.

Pourquoi les données ne sont-elles pas correctement ingérées par le service Amazon Kinesis Video Streams ?

Métriques pertinentes :

- `PutMedia.Requests`
- `PutMedia.ConnectionErrors`
- `PutMedia.Success`
- `PutMedia.ErrorAckCount`

Éléments d'action :

- En cas d'augmentation `PutMedia.ConnectionErrors`, examinez la réponse HTTP et les codes d'erreur reçus par le client producteur pour voir quelles erreurs se produisent lors de l'établissement de la connexion.

- En cas de baisse `PutMedia.Success` ou d'augmentation `PutMedia.ErrorAckCount`, examinez le code d'erreur ack dans les réponses ack envoyées par le service pour voir pourquoi l'ingestion des données échoue. Pour de plus amples informations, veuillez consulter [AckErrorCode.Values](#).

Pourquoi les données ne peuvent-elles pas être lues à partir du service Amazon Kinesis Video Streams au même rythme que celles envoyées par le producteur ?

Métriques pertinentes :

- `PutMedia.FragmentIngestionLatency`
- `PutMedia.IncomingBytes`

Éléments d'action :

- En cas de baisse de ces indicateurs, vérifiez la bande passante réseau de vos connexions. Low-bandwidth les connexions peuvent faire en sorte que les données atteignent le service à un débit inférieur.

Pourquoi la console ne lit-elle aucune vidéo, ou pourquoi la vidéo est-elle lue avec un retard ?

Métriques pertinentes :

- `PutMedia.FragmentIngestionLatency`
- `PutMedia.FragmentPersistLatency`
- `PutMedia.Success`
- `ListFragments.Latency`
- `PutMedia.IncomingFragments`

Éléments d'action :

- En cas d'augmentation `PutMedia.FragmentIngestionLatency` ou de baisse `PutMedia.IncomingFragments`, vérifiez la bande passante du réseau et vérifiez si les données sont toujours envoyées.
- En cas de baisse `PutMedia.Success`, vérifiez les codes d'erreur du pack. Pour de plus amples informations, veuillez consulter [AckErrorCode.Values](#).

- S'il y a une augmentation de `PutMedia.FragmentPersistLatency` ou `ListFragments.Latency`, il est fort probable que vous rencontriez un problème de service. Si le problème persiste pendant une période prolongée, contactez votre service client pour savoir s'il y a un problème avec votre service.

Quel est le retard dans la lecture des données en temps réel et pourquoi le client est-il en retard par rapport à la tête du flux ?

Métriques pertinentes :

- `GetMedia.MillisBehindNow`
- `GetMedia.ConnectionErrors`
- `GetMedia.Success`

Éléments d'action :

- En cas d'augmentation `GetMedia.ConnectionErrors`, le consommateur risque de prendre du retard dans la lecture du stream, en raison des fréquentes tentatives de reconnexion au stream. Regardez les `response/error codes HTTP` renvoyés pour la `GetMedia` requête.
- En cas de baisse `GetMedia.Success`, cela est probablement dû au fait que le service n'est pas en mesure d'envoyer les données au consommateur, ce qui entraînerait une perte de connexion, et des reconnexions avec les clients, ce qui entraînerait un retard par rapport à la tête du stream.
- En cas d'augmentation `GetMedia.MillisBehindNow`, examinez vos limites de bande passante pour voir si vous recevez les données à un rythme plus lent en raison d'une bande passante plus faible.

Le client lise-t-il les données du flux vidéo Kinesis, et à quel rythme ?

Métriques pertinentes :

- `GetMedia.OutgoingBytes`
- `GetMedia.OutgoingFragments`
- `GetMedia.OutgoingFrames`
- `GetMediaForFragmentList.OutgoingBytes`
- `GetMediaForFragmentList.OutgoingFragments`

- `GetMediaForFragmentList.OutgoingFrames`

Éléments d'action :

- Ces indicateurs indiquent le taux de lecture des données en temps réel et archivées.

Pourquoi le client ne peut-il pas lire les données du flux vidéo Kinesis ?

Métriques pertinentes :

- `GetMedia.ConnectionErrors`
- `GetMedia.Success`
- `GetMediaForFragmentList.Success`
- `PutMedia.IncomingBytes`

Éléments d'action :

- En cas d'augmentation `GetMedia.ConnectionErrors`, examinez la réponse HTTP et les codes d'erreur renvoyés par la `GetMedia` requête. Pour de plus amples informations, veuillez consulter [AckErrorCode.Values](#).
- Si vous essayez de lire les données les plus récentes ou en temps réel, vérifiez `PutMedia.IncomingBytes` si des données entrent dans le flux pour que le service les envoie aux clients.
- En `GetMedia.Success` cas de baisse `GetMediaForFragmentList.Success`, cela est probablement dû au fait que le service n'est pas en mesure d'envoyer les données au consommateur. Si le problème persiste pendant une période prolongée, contactez votre service client pour savoir s'il y a un problème avec votre service.


Surveillez l'agent Amazon Kinesis Video Streams Edge avec CloudWatch


Vous pouvez surveiller l'agent Amazon Kinesis Video Streams Edge à l'aide d' `CloudWatchAmazon`, qui collecte et traite les données brutes pour en faire des indicateurs lisibles en temps quasi réel. Ces statistiques sont enregistrées pour une période de 15 mois. Grâce à ces informations historiques, vous pouvez avoir une meilleure idée des performances de votre application Web ou du service Amazon Kinesis Video Streams Edge Agent.

Pour consulter les statistiques, procédez comme suit :

1. Connectez-vous à la CloudWatch console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le volet de navigation de gauche, sous Mesures, sélectionnez Toutes les mesures.
3. Choisissez l'onglet Parcourir, puis sélectionnez l'espace de noms EdgeRuntimeAgentpersonnalisé.

L'agent Amazon Kinesis Video Streams Edge publie les statistiques suivantes sous l'EdgeRuntimeAgentespace de noms :

| Dimension s | State | Description |
|------------------------|----------------------|---|
| Nom du flux, RecordJob | En cours d'exécution | Publie en continu lorsque le RecordJob est en cours d'exécution. Unités : Aucune. « 1 » est publié tant qu'RecordJob il est dans cet état. |
| | FatalError | Publie en cas d'erreur RecordJob fatale. Unités : Aucune. « 1 » est publié une fois, lorsque cet événement se produit. <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note Consultez les journaux pour plus d'informations.</p> </div> |
| | Terminé | Publie lorsqu'un RecordJob est terminé. Unités : Aucune. « 1 » est publié une fois, lorsque cet événement se produit. |
| Nom du flux, UploadJob | En cours d'exécution | Publie en continu lorsque le UploadJob est en cours d'exécution. Unités : Aucune. « 1 » est publié tant qu'UploadJob il est dans cet état. |

| Dimension s | State | Description |
|----------------|----------------------------|--|
| | FatalError | <p>Publie en cas d'erreur UploadJob fatale.</p> <p>Unités : Aucune. « 1 » est publié une fois, lorsque cet événement se produit.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note Consultez les journaux pour plus d'informations.</p> </div> |
| | Terminé | <p>Publie lorsque le UploadJob est terminé.</p> <p>Unités : Aucune. « 1 » est publié une fois, lorsque cet événement se produit.</p> |
| Nom du stream | PercentageSpaceUsed | <p>Il s'agit du pourcentage utilisé par rapport à l'espace total alloué dans les configurations de l'agent Amazon Kinesis Video Streams Edge pour l'enregistrement de supports. Pour plus d'informations, consultez LocalSizeConfig.</p> <p>Unités : pourcentage (échelle de 0 à 1).</p> |
| Nom de l'objet | Vivant | <p>Publie chaque minute depuis l'agent Amazon Kinesis Video Streams Edge, quelles que soient les configurations exécutées sur celui-ci.</p> <p>Cela peut être utilisé pour déterminer si l'agent Amazon Kinesis Video Streams Edge est actif et prêt à accepter des configurations.</p> <p>Unités : Aucune. « 1 » est publié toutes les minutes.</p> |
| | RecordJobs.HealthyJobCount | <p>Nombre total de tâches d'enregistrement en cours et planifiées sur Amazon Kinesis Video Streams Edge Agent.</p> <p>Unités : Nombre.</p> |

| Dimensions | State | Description |
|------------|------------------------------|--|
| | UploadJobs.HealthyJobCount | <p>Nombre total de tâches de téléchargement en cours et planifiées sur Amazon Kinesis Video Streams Edge Agent.</p> <p>Unités : Nombre.</p> |
| | RecordJobs.UnhealthyJobCount | <p>Nombre total de tâches d'enregistrement actuellement erronées.</p> <p>Unités : Nombre.</p> |
| | UploadJobs.UnhealthyJobCount | <p>Nombre total de tâches de téléchargement actuellement erronées.</p> <p>Unités : Nombre.</p> |
| | RecordJobs.RunningJobCount | <p>Nombre total de tâches d'enregistrement en cours d'exécution active.</p> <p>Unités : Nombre.</p> |
| | UploadJobs.RunningJobCount | <p>Nombre total de tâches de téléchargement en cours d'exécution active.</p> <p>Unités : Nombre.</p> |
| | RecordJobs.EdgeConfigCount | <p>Nombre total de configurations d'enregistrement en cours de traitement sur Amazon Kinesis Video Streams Edge Agent.</p> <p>Unités : Nombre.</p> |
| | UploadJobs.EdgeConfigCount | <p>Nombre total de configurations de téléchargement en cours sur Amazon Kinesis Video Streams Edge Agent.</p> <p>Unités : Nombre.</p> |

CloudWatch conseils relatifs aux métriques pour Amazon Kinesis Video Streams Edge Agent

CloudWatch les métriques peuvent être utiles pour trouver des réponses aux questions suivantes :

Rubriques

- [L'agent Amazon Kinesis Video Streams Edge dispose-t-il de suffisamment d'espace pour enregistrer ?](#)
- [L'agent Amazon Kinesis Video Streams Edge est-il actif ?](#)
- [Y a-t-il des emplois malsains ?](#)
- [Certains emplois nécessitent-ils une intervention extérieure ?](#)

L'agent Amazon Kinesis Video Streams Edge dispose-t-il de suffisamment d'espace pour enregistrer ?

Indicateurs pertinents : `PercentageSpaceUsed`

Action : Aucune action requise.

L'agent Amazon Kinesis Video Streams Edge est-il actif ?

Indicateurs pertinents : `Alive`

Action : Si, à un moment donné, vous ne recevez plus cette métrique, cela signifie que l'agent Amazon Kinesis Video Streams Edge a rencontré un ou plusieurs des problèmes suivants :

- Un problème d'exécution de l'application : contrainte de mémoire ou autre contrainte de ressources, bogue, etc.
- L' AWS IoT appareil sur lequel l'agent est en cours d'exécution lors de l'arrêt, s'est écrasé ou s'est arrêté
- L' AWS IoT appareil n'est pas connecté au réseau


Y a-t-il des emplois malsains ?

Métriques pertinentes :

- `RecordJobs.UnhealthyJobCount`
- `UploadJobs.UnhealthyJobCount`

Action : Inspectez les journaux et recherchez la `FatalError` métrique.

- Si la `FatalError` métrique est présente, une erreur fatale s'est produite et vous devez redémarrer le travail manuellement. Inspectez les journaux et corrigez le problème avant `StartEdgeConfigurationUpdate` de recommencer manuellement le travail.
- Si la `FatalError` métrique n'est pas présente, une erreur passagère (non fatale) s'est produite et l'agent Amazon Kinesis Video Streams Edge tente à nouveau la tâche.

 Note

Pour que l'agent tente à nouveau une tâche fatalement erronée, utilisez [StartEdgeConfigurationUpdate](#)

Certains emplois nécessitent-ils une intervention extérieure ?

Métriques pertinentes :

- `PercentageSpaceUsed`— Si cette valeur dépasse une certaine valeur, le travail d'enregistrement est suspendu et ne reprend que lorsque l'espace est disponible (lorsque le support n'est plus conservé). Vous pouvez envoyer une configuration mise à jour avec une valeur supérieure `MaxLocalMediaSizeInMB` pour mettre à jour la tâche immédiatement.
- `RecordJob.FatalError/UploadJob.FatalError`— Examinez les journaux de l'agent et renvoyez la configuration pour que le travail reprenne.

Action : effectuez un appel d'API avec la configuration pour redémarrer les tâches qui rencontrent ce problème.

Enregistrez les appels d'API Amazon Kinesis Video Streams avec AWS CloudTrail

Amazon Kinesis Video Streams fonctionne AWS CloudTrail avec un service qui fournit un enregistrement des actions effectuées par un utilisateur, un rôle ou Service AWS un utilisateur dans Amazon Kinesis Video Streams. CloudTrail capture tous les appels d'API pour Amazon Kinesis Video Streams sous forme d'événements. Les appels capturés incluent des appels provenant de la console Amazon Kinesis Video Streams et des appels de code vers les opérations de l'API Amazon Kinesis Video Streams. Si vous créez un suivi, vous pouvez activer la diffusion continue d'

CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour Amazon Kinesis Video Streams. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Amazon Kinesis Video Streams, l'adresse IP à partir de laquelle la demande a été effectuée, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires.

Pour en savoir plus CloudTrail, notamment comment le configurer et l'activer, consultez le [guide de AWS CloudTrail l'utilisateur](#).

Amazon Kinesis Video Streams et CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité événementielle prise en charge se produit dans Amazon Kinesis Video Streams, cette activité est enregistrée dans CloudTrail un événement avec AWS d'autres événements de service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section [Affichage des événements avec l'historique des CloudTrail événements](#).

Pour un enregistrement continu des événements de votre AWS compte, y compris les événements relatifs à Amazon Kinesis Video Streams, créez une trace. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions Régions AWS. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez en configurer d'autres Services AWS pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les ressources suivantes :

- [Vue d'ensemble de la création d'un journal d'activité](#)
- [CloudTrail Services et intégrations pris en charge](#)
- [Configuration des notifications Amazon SNS pour CloudTrail](#)
- [Réception de fichiers CloudTrail journaux de plusieurs régions](#) et [réception de fichiers CloudTrail journaux de plusieurs comptes](#)

Amazon Kinesis Video Streams prend en charge l'enregistrement des actions suivantes sous forme d'événements CloudTrail dans des fichiers journaux :

- [CreateStream](#)

- [DeleteStream](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [ListStreams](#)
- [ListTagsForStream](#)
- [TagStream](#)
- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateStream](#)

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les informations d'identification utilisateur racine ou
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré
- Si la requête a été effectuée par un autre Service AWS.

Pour plus d'informations, consultez la section [Élément userIdentity CloudTrail](#).

Exemple : entrées du fichier journal Amazon Kinesis Video Streams

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande individuelle à partir d'une source quelconque et comprend des informations sur l'action demandée, sur tous les paramètres, les paramètres de la demande, etc. Les fichiers journaux CloudTrail ne sont pas des séries ordonnées retraçant les appels d'API publics. Ils ne suivent aucun ordre précis.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'[CreateStream](#) action.

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
```

```

        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2018-05-25T00:16:31Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "CreateStream",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
        "streamName": "VideoStream",
        "dataRetentionInHours": 2,
        "mediaType": "mediaType",
        "kmsKeyId": "arn:aws:kms::us-east-1:123456789012:alias",
        "deviceName": "my-device"
    },
    "responseElements": {
        "streamARN": "arn:aws:kinesisvideo:us-east-1:123456789012:stream/VideoStream/12345"
    },
    "requestID": "db6c59f8-c757-11e3-bc3b-57923b443c1c",
    "eventID": "b7acfcfd0-6ca9-4ee1-a3d7-c4e8d420d99b"
},
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2018-05-25:17:06Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "DeleteStream",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {

```

```
        "streamARN": "arn:aws:kinesisvideo:us-east-1:012345678910:stream/
VideoStream/12345",
        "currentVersion": "keqrjeqkj9"
    },
    "responseElements": null,
    "requestID": "f0944d86-c757-11e3-b4ae-25654b1d3136",
    "eventID": "0b2f1396-88af-4561-b16f-398f8eaea596"
},
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-04-19T00:15:02Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "DescribeStream",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
        "streamName": "VideoStream"
    },
    "responseElements": null,
    "requestID": "a68541ca-c757-11e3-901b-cbcfe5b3677a",
    "eventID": "22a5fb8f-4e61-4bee-a8ad-3b72046b4c4d"
},
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-04-19T00:15:03Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "GetDataEndpoint",
```

```

    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
      "streamName": "VideoStream",
      "aPName": "LIST_FRAGMENTS"
    }
  },
  "responseElements": null,
  "requestID": "a6e6e9cd-c757-11e3-901b-cbcfe5b3677a",
  "eventID": "dcd2126f-c8d2-4186-b32a-192dd48d7e33"
},
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/Alice",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2018-05-25T00:16:56Z",
  "eventSource": "kinesisvideo.amazonaws.com",
  "eventName": "ListStreams",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
  "requestParameters": {
    "maxResults": 100,
    "streamNameCondition": {"comparisonValue":"MyVideoStream"
comparisonOperator":"BEGINS_WITH"}}
  },
  "responseElements": null,
  "requestID": "e9f9c8eb-c757-11e3-bf1d-6948db3cd570",
  "eventID": "77cf0d06-ce90-42da-9576-71986fec411f"
}
]
}

```

Limites de métadonnées de diffusion

Consultez [the section called “Quotas du service de métadonnées en streaming”](#) pour plus d'informations sur les limites applicables à l'ajout de métadonnées de streaming à un flux vidéo Kinesis.

Planifiez l'enregistrement et le stockage vidéo avec Amazon Kinesis Video Streams Edge Agent

Amazon Kinesis Video Streams offre un moyen efficace et économique de se connecter aux caméras IP dans les locaux du client. Avec l'agent Amazon Kinesis Video Streams Edge, vous pouvez enregistrer et stocker localement des vidéos à partir des caméras et diffuser des vidéos vers le cloud selon un calendrier défini par le client pour un stockage, une lecture et un traitement analytique à long terme.

Note

Pour accéder à l'agent Amazon Kinesis Video Streams Edge, complétez [ce bref formulaire](#).

Vous pouvez télécharger l'agent Amazon Kinesis Video Streams Edge et le déployer sur vos appareils informatiques de pointe sur site. Vous pouvez également les déployer facilement dans des conteneurs Docker exécutés sur des instances Amazon EC2. Après le déploiement, vous pouvez utiliser l'API Amazon Kinesis Video Streams pour mettre à jour les configurations d'enregistrement vidéo et de téléchargement dans le cloud. Cette fonctionnalité fonctionne avec n'importe quelle caméra IP capable de diffuser via le protocole RTSP. Il ne nécessite aucun déploiement de microprogramme supplémentaire sur les caméras.

Nous proposons les installations suivantes pour l'agent Amazon Kinesis Video Streams Edge :

- En tant que AWS IoT Greengrass V2 composant : vous pouvez installer l'agent Amazon Kinesis Video Streams Edge en AWS IoT Greengrass tant que composant sur AWS IoT Greengrass n'importe quel appareil certifié. Pour en savoir plus AWS IoT Greengrass, consultez le [guide du AWS IoT Greengrass Version 2 développeur](#).
- Activé AWS Snowball Edge : vous pouvez exécuter l'agent Amazon Kinesis Video Streams Edge sur les appareils Snowball Edge. Pour en savoir plus, consultez le [guide du développeur AWS Snowball Edge Edge](#).
- Dans le cadre d'un AWS IoT déploiement natif : vous pouvez installer l'agent Amazon Kinesis Video Streams Edge de manière native sur n'importe quelle instance de calcul. Le SDK Edge est utilisé [AWS IoT Core](#) pour gérer l'Edge via les opérations [d'API Amazon Kinesis Video Streams](#).

Pour commencer à utiliser Amazon Kinesis Video Streams Edge Agent, suivez les procédures appropriées ci-dessous.

Rubriques

- [Opérations de l'API Amazon Kinesis Video Streams Edge Agent](#)
- [Surveillance de l'agent Amazon Kinesis Video Streams Edge](#)
- [Déployez dans un environnement non gouvernemental AWS IoT Greengrass mode](#)
- [Déployez l'agent Amazon Kinesis Video Streams Edge pour AWS IoT Greengrass](#)
- [FAQ sur l'agent Amazon Kinesis Video Streams Edge](#)

Opérations de l'API Amazon Kinesis Video Streams Edge Agent

Utilisez les opérations d'API suivantes pour configurer l'agent Amazon Kinesis Video Streams Edge :

- [StartEdgeConfigurationUpdate](#)
- [DescribeEdgeConfiguration](#)
- [DeleteEdgeConfiguration](#)
- [ListEdgeAgentConfigurations](#)

Surveillance de l'agent Amazon Kinesis Video Streams Edge

Pour surveiller votre agent Amazon Kinesis Video Streams Edge, [the section called “Surveillez l'agent Amazon Kinesis Video Streams Edge avec CloudWatch”](#) consultez.

Déployez dans un environnement non gouvernemental AWS IoT Greengrass mode

Cette section fournit un guide complet sur l'utilisation d'Amazon Kinesis Video Streams en dehors de AWS IoT Greengrass l'environnement. Que vous travailliez avec des appareils périphériques ou d'autres plateformes, ces informations vous aideront à configurer et à utiliser Kinesis Video Streams de manière efficace.

Vous trouverez des informations détaillées sur :

- Configuration de votre environnement de développement

- Création d'un flux vidéo Kinesis
- Téléchargement et compilation du SDK Kinesis Video Streams Producer
- Rédaction et examen d'un exemple de candidature
- Exécution de l'exemple d'application

Suivez les étapes ci-dessous pour exécuter l'agent Amazon Kinesis Video Streams Edge AWS IoT avec MQTT en tant que déploiement autonome.

Rubriques

- [Installation des dépendances](#)
- [Créez des ressources pour les URL RTSP de vos caméras IP](#)
- [Création d'une politique d'autorisations IAM](#)
- [Créer un rôle IAM](#)
- [Créez le AWS IoT alias de rôle](#)
- [Créez le AWS IoT policy](#)
- [Créez un AWS IoT Chose et obtiens AWS IoT Core informations d'identification](#)
- [Création de l'agent Amazon Kinesis Video Streams Edge](#)
- [Installation de l' CloudWatch agent sur l'appareil](#)
- [Exécutez l'agent Amazon Kinesis Video Streams Edge en tant que processus natif](#)

Installation des dépendances

Avant de commencer à utiliser le SDK Amazon Kinesis Video Streams Producer, vous devez configurer votre environnement de développement avec les dépendances nécessaires. Cette page vous guide tout au long du processus d'installation des composants logiciels et des bibliothèques requis sur votre système.

Note

Pour obtenir la liste des systèmes d'exploitation pris en charge, consultez [the section called “Quels sont les systèmes d'exploitation pris en charge par Amazon Kinesis Video Streams Edge Agent ?”](#).

Installer les dépendances sur l'appareil

1. Pour exécuter l'agent Amazon Kinesis Video Streams Edge, installez les bibliothèques appropriées suivantes sur votre appareil :

Ubuntu

Type :

```
wget -O- https://apt.corretto.aws/corretto.key | sudo apt-key add -
sudo add-apt-repository 'deb https://apt.corretto.aws stable main'
sudo apt-get update

sudo apt-get install -y gcc libssl-dev libcurl4-openssl-dev liblog4cplus-dev \
libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev \
gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-bad \
gstreamer1.0-plugins-good gstreamer1.0-tools \
unzip java-11-amazon-corretto-jdk maven
```

Amazon Linux 2

Type :

```
sudo yum update -y && sudo yum upgrade -y && sudo yum clean all -y
sudo yum install -y gcc-c++ openssl-devel libcurl-devel gstreamer1* wget \
java-11-amazon-corretto tar
```

Installez `log4cplus-2.1.0` à partir de la source.

```
wget https://github.com/log4cplus/log4cplus/releases/download/REL_2_1_0/
log4cplus-2.1.0.tar.gz
tar -xzvf log4cplus-2.1.0.tar.gz
cd log4cplus-2.1.0 && \
mkdir build && \
cd build && \
cmake .. && \
sudo make && \
sudo make install
```

Installez `apache-maven-3.9.2` à partir de la source.

```
wget https://d1cdn.apache.org/maven/maven-3/3.9.2/binaries/apache-maven-3.9.2-bin.tar.gz
RUN tar -xzvf apache-maven-3.9.2-bin.tar.gz -C /opt
```

Important

Si un écran vous indique que certains services doivent être redémarrés, appuyez sur Entrée pour sélectionner OK.

Pour plus d'informations, consultez le guide de l'utilisateur [d'Amazon Corretto 11](#).

2. Installez le AWS Command Line Interface. Consultez la section [Installation ou mise à jour de la dernière version des AWS CLI](#) procédures dans le guide de AWS Command Line Interface l'utilisateur.

Créez des ressources pour les URL RTSP de vos caméras IP

Suivez ces procédures pour créer les flux et les secrets nécessaires dans AWS Secrets Manager. Effectuez d'abord cette étape, car vous avez besoin des ARN des ressources créées dans les politiques.

Création d'Amazon Kinesis Video Streams

Créez Amazon Kinesis Video Streams à l'aide AWS Management Console de l' AWS CLI API, ou.

Dans le AWS Management Console, ouvrez la console [Amazon Kinesis Video Streams](#). Choisissez Streams vidéo dans le menu de navigation de gauche.

Pour de plus amples informations, veuillez consulter [the section called "Création d'un flux vidéo Amazon Kinesis"](#).

Créez des secrets dans AWS Secrets Manager

Dans le AWS Management Console, ouvrez la [AWS Secrets Manager console](#). Choisissez Secrets dans le menu de navigation de gauche.

Vérifiez que la région appropriée est sélectionnée.

1. Choisissez Store a new secret (Stocker un nouveau secret).

a. Étape 1 : Choisissez le type de secret

- Sélectionnez Autre type de secret.
- Dans la section Key/Value Pairs, ajoutez une paire clé-valeur.

Clé : MediaURI

Note

La clé doit être MediaURI. Cela distingue les majuscules et minuscules. Si vous ne le saisissez pas correctement, l'application ne fonctionne pas.

Valeur : *Your MediaURI*.

Exemple

Exemple : `rtsp://<YourCameraIPAddress>:<YourCameraRTSPPort>/
YourCameraMediaURI.`

- b. Étape 2 : configurer le secret Donnez un nom à ce secret. Nomme-le comme tu veux.
- c. Étape 3 : Configuration de la rotation (facultatif). Choisissez Suivant.
- d. Étape 4 : Révision. Choisissez Stocker.

2. Si votre secret ne s'affiche pas immédiatement, sélectionnez le bouton d'actualisation.

Choisissez le nom de votre secret. Prenez note de l'ARN secret.

3. Répétez ce processus pour chaque MediaURI à partir duquel vous souhaitez diffuser.

Note

Le AWS réseau bloque certaines sources RTSP publiques. Vous ne pouvez pas y accéder depuis l'instance Amazon EC2 ou si vous les exécutez de manière non gérée lorsque vous êtes connecté au VPN.

⚠ Important

L'URL RTSP de votre caméra doit diffuser de la vidéo au format h.264. La durée du fragment ne doit pas dépasser la limite mentionnée dans [the section called “Quotas du SDK pour les producteurs”](#).

L'agent Amazon Kinesis Video Streams Edge prend uniquement en charge la vidéo.

Exécutez `gst-discoverer-1.0 Your RtspUrl` pour vous assurer que votre caméra est accessible depuis votre appareil.

Enregistrez les ARN de tous les flux et secrets que vous avez créés. Vous en aurez besoin pour passer à l'étape suivante.

Création d'une politique d'autorisations IAM

Suivez ces procédures pour créer une politique IAM. Cette politique d'autorisation permet un contrôle d'accès sélectif (un sous-ensemble des opérations prises en charge) pour une AWS ressource. Dans ce cas, les AWS ressources sont les flux vidéo vers lesquels vous souhaitez que l'agent Amazon Kinesis Video Streams Edge diffuse. Les ressources incluent également les AWS Secrets Manager secrets que l'agent Amazon Kinesis Video Streams Edge peut récupérer. Pour plus d'informations, consultez la section [Politiques IAM](#).

Créez une politique à l'aide de l'éditeur de stratégie JSON

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
2. Dans le volet de navigation de gauche, choisissez Politiques.

Si vous sélectionnez Politiques pour la première fois, la page Bienvenue dans les politiques gérées s'affiche. Sélectionnez Mise en route.

3. En haut de la page, sélectionnez Créer une politique.
4. Dans la section Éditeur de politique, choisissez l'option JSON.
5. Entrez le document de politique JSON suivant :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData",
        "kinesisvideo:ListStreams",
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": [
        "arn:aws:kinesisvideo:*:*:stream/streamName1/*",
        "arn:aws:kinesisvideo:*:*:stream/streamName2/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": [
        "arn:aws:secretsmanager:*:*:secret:*",
        "arn:aws:secretsmanager:*:*:secret:*"
      ]
    }
  ]
}
```

```
}
```

Note

Remplacez `arn:aws:kinesisvideo:*:*:stream/streamName1/*` et `arn:aws:kinesisvideo:*:*:stream/streamName2/*` par les ARN des flux vidéo, et remplacez-les par les `arn:aws:secretsmanager:*:*:secret:*` ARN contenant les secrets MediaURI que vous avez créés dans [the section called “Créez des ressources pour les URL RTSP de vos caméras IP”](#) Utilisez les ARN pour les secrets auxquels vous souhaitez que l'agent Amazon Kinesis Video Streams Edge accède.

6. Choisissez Suivant.

Note

Vous pouvez basculer à tout moment entre les options des éditeurs visuel et JSON. Toutefois, si vous apportez des modifications ou si vous choisissez Suivant dans l'éditeur visuel, IAM peut restructurer votre politique afin de l'optimiser pour l'éditeur visuel. Pour plus d'informations, consultez la section [Restructuration des politiques](#) dans le guide de l'utilisateur IAM.

7. Sur la page Réviser et créer, entrez le nom de la politique et une description facultative pour la stratégie que vous créez. Vérifiez les Autorisations définies dans cette politique pour voir les autorisations accordées par votre politique.
8. Choisissez Create policy (Créer une politique) pour enregistrer votre nouvelle politique.

Créer un rôle IAM

Le rôle que vous créez au cours de cette étape peut être assumé par AWS IoT afin d'obtenir des informations d'identification temporaires auprès de AWS Security Token Service (AWS STS). Cela est effectué lorsque vous effectuez des demandes d'autorisation d'identification depuis l'agent Amazon Kinesis Video Streams Edge.

Création du rôle de service pour Amazon Kinesis Video Streams (console IAM)

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.

2. Dans le volet de navigation de la console IAM, sélectionnez Roles (Rôles), puis Create role (Créer un rôle).
3. Choisissez le type de rôle de politique de confiance personnalisée et collez la politique suivante :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {
      "Service": "credentials.iot.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
}
```

4. Cochez la case à côté de la politique IAM que vous avez créée dans [the section called “Création d'une politique d'autorisations IAM”](#).
5. Choisissez Suivant.
6. Entrez un nom de rôle ou un suffixe de nom de rôle pour vous aider à identifier l'objectif de ce rôle.

Exemple

Exemple : KvsEdgeAgentRole

7. (Facultatif) Pour Description, saisissez une description pour le nouveau rôle.
8. (Facultatif) Ajoutez des métadonnées au rôle en attachant des balises par key/value paires.

Pour plus d'informations sur l'utilisation des balises dans IAM, consultez la section [Balisage des ressources IAM](#) dans le guide de l'utilisateur IAM.

9. Passez en revue les informations du rôle, puis choisissez Créer un rôle.

Créez le AWS IoT alias de rôle

Suivez ces procédures pour créer un alias de AWS IoT rôle pour le rôle IAM que vous avez créé dans [the section called “Créer un rôle IAM”](#). Un alias de rôle est un modèle de données alternatif qui pointe vers le rôle IAM. Une demande de fournisseur AWS IoT d'informations d'identification doit

inclure un alias de rôle indiquant le rôle IAM à assumer afin d'obtenir des informations d'identification temporaires auprès de AWS Security Token Service (AWS STS). Pour plus d'informations, consultez [Comment utiliser un certificat pour obtenir un jeton de sécurité](#).

Créez le AWS IoT alias de rôle

1. Connectez-vous à la AWS IoT Core console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/iot/>.
2. Vérifiez que la région appropriée est sélectionnée.
3. Dans le volet de navigation de gauche, sélectionnez Sécurité, puis cliquez sur Alias de rôle.
4. Choisissez Créer un alias de rôle.
5. Entrez un nom pour votre alias de rôle.

Exemple

Exemple : `KvsEdgeAgentRoleAlias`

6. Dans le menu déroulant Rôle, sélectionnez le rôle IAM que vous avez créé dans [the section called "Créer un rôle IAM"](#)
7. Choisissez Créer. Sur la page suivante, vous pouvez voir une note indiquant que votre alias de rôle a été créé avec succès.
8. Recherchez et sélectionnez le nouvel alias de rôle. Prenez note de l'alias ARN du rôle. Vous en aurez besoin pour la AWS IoT politique à l'étape suivante.

Créez le AWS IoT policy

Suivez ces procédures pour créer une AWS IoT politique qui sera attachée au certificat de l'appareil. Cela donne des autorisations sur les AWS IoT fonctionnalités et permet de prendre en charge l'alias de rôle à l'aide du certificat.

AWS IoT Core Les politiques vous permettent de contrôler l'accès au plan AWS IoT Core de données. Le plan de AWS IoT Core données comprend des opérations que vous pouvez utiliser pour effectuer les opérations suivantes :

- Connectez-vous au courtier de AWS IoT Core messages
- Envoyer et recevoir des messages MQTT
- Obtenir ou mettre à jour l'ombre de l'appareil d'un objet

Pour plus d'informations, consultez [AWS IoT Core Stratégies](#).

Utilisation AWS IoT éditeur de politiques pour créer un AWS IoT policy

1. Connectez-vous à la AWS IoT Core console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/iot/>.
2. Dans le volet de navigation de gauche, sélectionnez Sécurité, puis Politiques.
3. Choisissez Create Policy (Créer une politique).
4. Entrez un nom pour votre politique.

Exemple

Un exemple de nom de politique est KvsEdgeAccessIoTPolicy.

5. (Facultatif) Ajoutez des métadonnées à la politique en associant les balises sous forme de paires clé-valeur.

Pour plus d'informations sur l'utilisation des balises dans IAM, consultez la section [Marquage de vos AWS IoT ressources](#) dans le Guide du AWS IoT Core développeur.

6. Choisissez l'onglet JSON.
7. Collez le document de politique JSON suivant :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

```
        "Effect": "Allow",
        "Action": [
            "sts:AssumeRoleWithWebIdentity"
        ],
        "Resource": "arn:aws:iot:us-west-2:123456789012:rolealias/your-  
role-alias"
    }
]
```

Note

your-role-alias-arn Remplacez-le par l'ARN de l'alias de rôle que vous avez créé dans [the section called "Création de l'alias de AWS IoT rôle"](#).

8. Choisissez Créer pour enregistrer votre travail.

Créez un AWS IoT Chose et obtiens AWS IoT Core informations d'identification

À ce stade, vous avez créé :

- Une politique d'autorisations IAM. Consultez [the section called "Création d'une politique d'autorisations IAM"](#).
- Un rôle IAM, auquel est attachée la politique d'autorisation. Consultez [the section called "Créer un rôle IAM"](#).
- Alias de AWS IoT rôle pour le rôle IAM. Consultez [the section called "Création de l'alias de AWS IoT rôle"](#).
- Une AWS IoT politique, actuellement indépendante d'une AWS ressource. Consultez [the section called "Création de la AWS IoT politique"](#).

Pour créer et enregistrer un AWS IoT Chose et obtiens AWS IoT Core informations d'identification d'accès

1. Enregistrez l'appareil en tant qu' AWS IoT objet et générez le X.509 certificat pour l'appareil.

- a. Connectez-vous à la AWS IoT Core console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/iot/>.
- b. Sélectionnez la région appropriée.
- c. Dans le volet de navigation de gauche, sélectionnez Tous les appareils, puis Objets.
- d. Choisissez Create things.
- e. Sélectionnez Créer un objet unique, puis cliquez sur Suivant.

1. Étape 1. Spécifier les propriétés de l'objet

Tapez un nom pour votre objet, puis choisissez Next.

2. Étape 2. Configurer le certificat de l'appareil

Sélectionnez Auto-generate un nouveau certificat (recommandé), puis cliquez sur Suivant.

3. Étape 3. Joindre des politiques au certificat

Recherchez la politique d'autorisation que vous avez créée dans [the section called "Création de la AWS IoT politique"](#).

Cochez la case à côté de votre politique et choisissez Create thing.

- f. Dans la fenêtre qui apparaît, téléchargez les fichiers suivants :

- Certificat de l'appareil. Voici le X.509 certificat.
- Fichier de clé publique
- Fichier de clé privée
- Point de terminaison Amazon Trust Services (clé RSA 2048 bits : Amazon Root CA 1)

Notez l'emplacement de chacun de ces fichiers pour une étape ultérieure.

- g. Sélectionnez Exécuté. Sur la page suivante, vous pouvez voir une note indiquant que votre objet a été créé avec succès.
 - h. Transférez les fichiers téléchargés ci-dessus sur votre appareil AWS IoT , s'ils ne s'y trouvent pas déjà.
2. Obtenez le point de terminaison du fournisseur d'informations d'identification pour votre AWS compte.

AWS CLI

Exécutez la commande suivante :

```
aws iot describe-endpoint --endpoint-type iot:CredentialProvider
```

AWS Management Console

Dans [AWS CloudShell](#), exécutez la commande suivante :

```
aws iot describe-endpoint --endpoint-type iot:CredentialProvider
```

Prenez note de ces informations pour une étape ultérieure.

3. Obtenez le point de terminaison des données de l'appareil associé à votre AWS compte.

AWS CLI

Exécutez la commande suivante :

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

AWS Management Console

Procédez comme suit :

1. Connectez-vous à la AWS IoT Core console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/iot/>.
2. Dans le volet de navigation de gauche, sélectionnez Réglages.
3. Localisez le point de terminaison des données de l'appareil.

Prenez note de ces informations pour une étape ultérieure.

4. (Facultatif) Vérifiez que vos certificats ont été générés correctement.

Exécutez la commande suivante pour vérifier que vos articles ont été générés correctement.

```
curl --header "x-amzn-iot-thingname:your-thing-name" \  
--cert /path/to/certificateID-certificate.pem.crt \  

```

```
--key /path/to/certificateID-private.pem.key \  
--cacert /path/to/AmazonRootCA1.pem \  
https://your-credential-provider-endpoint/role-aliases/your-role-alias-name/  
credentials
```

Pour plus d'informations, consultez [Comment utiliser un certificat pour obtenir un jeton de sécurité](#).

Création de l'agent Amazon Kinesis Video Streams Edge

Création de l'agent Amazon Kinesis Video Streams Edge

1. Téléchargez le tar fichier en utilisant le lien qui vous a été fourni.

Si vous avez rempli le formulaire d'intérêt pour Amazon Kinesis Video Streams Edge Agent, consultez vos e-mails pour trouver le lien de téléchargement. Si vous n'avez pas rempli le formulaire, complétez-le [ici](#).

2. Vérifiez le checksum.
3. Extrayez les fichiers binaires et le fichier jar de votre appareil.

```
Type :tar -xvf kvs-edge-agent.tar.gz.
```

Après l'extraction, la structure de vos dossiers ressemblera à ce qui suit :

```
kvs-edge-agent/LICENSE  
kvs-edge-agent/THIRD-PARTY-LICENSES  
kvs-edge-agent/pom.xml  
kvs-edge-agent/KvsEdgeComponent  
kvs-edge-agent/KvsEdgeComponent/recipes  
kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml  
kvs-edge-agent/KvsEdgeComponent/artifacts  
kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent  
kvs-edge-agent/KvsEdgeComponent/artifacts/  
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion  
kvs-edge-agent/KvsEdgeComponent/artifacts/  
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/edge_log_config  
kvs-edge-agent/KvsEdgeComponent/artifacts/  
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/kvs-edge-agent.jar  
kvs-edge-agent/KvsEdgeComponent/artifacts/  
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libgstkvssink.so
```

```
kvs-edge-agent/KvsEdgeComponent/artifacts/  
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libIngestorPipelineJNI.so  
kvs-edge-agent/KvsEdgeComponent/artifacts/  
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib  
kvs-edge-agent/KvsEdgeComponent/artifacts/  
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libcproducer.so  
kvs-edge-agent/KvsEdgeComponent/artifacts/  
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libKinesisVideoProducer.so
```

Note

Le nom du dossier de version doit être configuré de manière à refléter le dernier numéro de version binaire. Par exemple, le nom du dossier d'une version 1.0.0 sera défini comme 1.0.0.

4. Créez le fichier jar de dépendances.

Note

Le fichier jar inclus avec le `kvs-edge-agent.tar.gz` ne contient pas les dépendances. Suivez les étapes ci-dessous pour créer ces bibliothèques.

Accédez au `kvs-edge-agent` dossier qui contient `pom.xml`.

Tapez `mvn clean package`.

Cela génère un fichier jar contenant les dépendances requises par l'agent Amazon Kinesis Video Streams Edge. `kvs-edge-agent/target/libs.jar`

5. `libs.jar` Placez-le dans le dossier contenant les artefacts du composant.

Tapez `mv ./target/libs.jar ./KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/`.

6. Définissez les variables d'environnement à l'aide des valeurs des étapes précédentes. Le tableau suivant décrit les variables.

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|---|
| AWS_REGION | Oui | La région utilisée. Exemple : us-west-2 |
| AWS_IOT_CA_CERT | Oui | Chemin du fichier vers le certificat CA utilisé pour établir la confiance avec le service principal via TLS. Exemple : <i>/file/path/to/AmazonRootCA1.pem</i> |
| AWS_IOT_CORE_CERT | Oui | Chemin du fichier vers le X.509 certificat. Exemple : <i>/file/path/to/certificateID-certificate.pem.crt</i> |
| AWS_IOT_CORE_CREDENTIAL_ENDPOINT | Oui | Le point de terminaison du fournisseur d'AWS IoT Core identifiants pour votre AWS compte. Exemple : <i>credential-account-specific-prefix.credentials.iot.region.amazonaws.com</i> |

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|--|
| AWS_IOT_CORE_DATA_ATS_ENDPOINT | Oui | <p>Le point de terminaison du plan de AWS IoT Core données de votre AWS compte.</p> <p>Exemple : <i>data-account-specific-prefix.iot.aws-region.amazonaws.com</i></p> |
| AWS_IOT_CORE_PRIVATE_KEY | Oui | <p>Chemin du fichier vers la clé privée utilisée dans la paire de public/private clés. Pour plus d'informations, consultez la section Gestion des clés dans AWS IoT.</p> <p>Exemple : <i>/file/path/to/certificateID-private.pem.key</i></p> |
| AWS_IOT_CORE_ROLE_ALIAS | Oui | <p>Nom de l'alias de rôle pointant vers le rôle AWS IAM à utiliser lors de la connexion. AWS IoT Core</p> <p>Exemple : <i>kvs-edge-role-alias</i></p> |

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|---|
| AWS_IOT_CORE_THING_NAME | Oui | <p>Nom de l' AWS IoT objet sur lequel l'application est exécutée.</p> <p>Exemple : my-edge-device-thing</p> |
| GST_PLUGIN_PATH | Oui | <p>Chemin du fichier pointant vers le dossier contenant les bibliothèques dépendantes de la IngestorPipelineJNI plate-forme gstkvsink et de celle-ci. Laisse GStreamer charger ces plugins. Pour de plus amples informations, veuillez consulter the section called “Téléchargez, compilez et configurez l'élément GStreamer”.</p> <p>Exemple : <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /</i></p> |

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|---|
| LD_LIBRARY_PATH | Oui | <p>Chemin du fichier pointant vers le répertoire contenant les bibliothèques <code>cproducer</code> dépendantes de la <code>KinesisVideoProducer</code> plate-forme et</p> <p>Exemple : <code>/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/</code></p> |
| AWS_KVS_EDGE_CLOUD_WATCH_ENABLED | Non | <p>Détermine si l'agent Amazon Kinesis Video Streams Edge publiera les indicateurs Amazon CloudWatch de santé au travail sur.</p> <p>Valeurs acceptées : TRUE/FALSE (sans distinction majuscules/minuscules). La valeur par défaut est FALSE si elle n'est pas fournie.</p> <p>Exemple : FAUX</p> |

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|---|
| AWS_KVS_EDGE_LOG_LEVEL | Non | <p>Niveau de journalisation des sorties de l'agent Amazon Kinesis Video Streams Edge.</p> <p>Valeurs acceptées :</p> <ul style="list-style-type: none">• OFF• ALL• FATAL• ERROR• WARN• INFO, par défaut, s'il n'est pas fourni• DEBUG• TRACE <p>Exemple : INFO</p> |
| AWS_KVS_EDGE_LOG_MAX_FILE_SIZE | Non | <p>Une fois que le fichier journal atteint cette taille, un roulement se produit.</p> <ul style="list-style-type: none">• Minimum : 0• Maximum : 10000• Par défaut : 20, s'il n'est pas fourni• Unités : mégaoctets (Mo) <p>Exemple : 5</p> |

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|---|
| AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY | Non | <p>Le chemin du fichier pointant vers le répertoire dans lequel les journaux de l'agent Amazon Kinesis Video Streams Edge sont générés. La valeur par défaut est <code>./log</code> si elle n'est pas fournie.</p> <p>Exemple : <code>./file/path/</code></p> |
| AWS_KVS_EDGE_LOG_ROLLOVER_COUNT | Non | <p>Le nombre de journaux reportés à conserver avant de les supprimer.</p> <ul style="list-style-type: none">• Minimum : 1• Maximum : 100• Par défaut : 10, s'il n'est pas fourni <p>Exemple : 20</p> |
| AWS_KVS_EDGE_RECORDING_DIRECTORY | Non | <p>Le chemin du fichier pointant vers le répertoire dans lequel le support enregistré sera écrit. La valeur par défaut est le répertoire actuel s'il n'est pas fourni.</p> <p>Exemple : <code>./file/path/</code></p> |

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|---|
| GST_DEBUG | Non | Spécifie le niveau des journaux GStreamer à générer. Pour plus d'informations, consultez la documentation de GStreamer . Exemple : 0 |
| GST_DEBUG_FILE | Non | Spécifie le fichier de sortie des journaux de débogage de GStreamer. S'ils ne sont pas définis, les journaux de débogage sont renvoyés à une erreur standard. Pour plus d'informations, consultez la documentation de GStreamer . Exemple : <i>/tmp/gstreamer-logging.log</i> |

7. Videz le cache de GStreamer. Type :


```
rm ~/.cache/gstreamer-1.0/registry.your-os-architecture.bin
```

Pour plus d'informations, consultez la documentation du [registre GStreamer](#).

8. Préparez et exécutez la commande Java. L'agent Amazon Kinesis Video Streams Edge accepte les arguments suivants :

| Nom de la propriété Java | Obligatoire | Description |
|--------------------------|-------------|--|
| java.library.path | Non | Chemin du fichier pointant vers le dossier contenant |

| Nom de la propriété Java | Obligatoire | Description |
|--------------------------|-------------|--|
| | | gstkvsink les bibliothèques IngestorPipelineJNI dépendantes. S'ils ne sont pas fournis, l'agent Amazon Kinesis Video Streams Edge les recherchera dans le répertoire actuel. |

 Important
L'agent Amazon Kinesis Video Streams Edge ne fonctionnera pas correctement s'il ne parvient pas à localiser ces fichiers.

Exemple : */file/path/*

Pour les définir, ajoutez-les `-Djava-property-name=value` à la commande java utilisée pour exécuter le fichier jar.

Par exemple :

```
java -Djava.library.path=download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion \  
  --add-opens java.base/jdk.internal.misc=ALL-UNNAMED \  
  -Dio.netty.tryReflectionSetAccessible=true \  
  -cp kvs-edge-agent.jar:libs.jar \  
  com.amazonaws.kinesisvideo.edge.controller.ControllerApp
```

⚠ Important

Exécutez la commande java ci-dessus depuis le même répertoire que `/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion`.

9. Envoyez les configurations à l'application à l'aide du AWS CLI.

a. Créez un nouveau fichier, `example-edge-configuration.json`.

Collez le code suivant dans le fichier. Il s'agit d'un exemple de configuration qui enregistre tous les jours de 9 h 00 à 16 h 59 min 59 s (en fonction de l'heure système de votre appareil). AWS IoT II télécharge également les médias enregistrés tous les jours de 19 h 00 à 21 h 59 min 59 s.

Pour de plus amples informations, veuillez consulter [StartEdgeConfigurationUpdate](#).

```
{
  "StreamARN": "arn:aws:kinesisvideo:your-region:your-account-id:stream/your-stream/0123456789012",
  "EdgeConfig": {
    "HubDeviceArn": "arn:aws:iot:your-region:your-account-id:thing/kvs-edge-agent-demo",
    "RecorderConfig": {
      "MediaSourceConfig": {
        "MediaUriSecretArn": "arn:aws:secretsmanager:your-region:your-account-id:secret:your-secret-dRbHJQ",
        "MediaUriType": "RTSP_URI"
      },
      "ScheduleConfig": {
        "ScheduleExpression": "0 0 9,10,11,12,13,14,15,16 ? * * **",
        "DurationInSeconds": 3599
      }
    },
    "UploaderConfig": {
      "ScheduleConfig": {
        "ScheduleExpression": "0 0 19,20,21 ? * * **",
        "DurationInSeconds": 3599
      }
    },
    "DeletionConfig": {
```

```
        "EdgeRetentionInHours": 15,
        "LocalSizeConfig": {
            "MaxLocalMediaSizeInMB": 2800,
            "StrategyOnFullSize": "DELETE_OLDEST_MEDIA"
        },
        "DeleteAfterUpload": true
    }
}
```

- b. Pour envoyer le fichier à l'agent Amazon Kinesis Video Streams Edge, saisissez ce qui suit dans AWS CLI le champ :

```
aws kinesishvideo start-edge-configuration-update --cli-input-json
"file://example-edge-configuration.json"
```

10. Répétez l'étape précédente pour chaque flux pour l'agent Amazon Kinesis Video Streams Edge.

Installation de l' CloudWatch agent sur l'appareil

Note

Soyez conscient des [CloudWatchquotas](#).

Suivez ces procédures pour installer et configurer l' CloudWatch agent afin qu'il télécharge automatiquement les journaux générés par l'agent CloudWatch Amazon Kinesis Video Streams Edge vers. Il s'agit d'une étape facultative.

Pour connaître [les procédures](#) d'installation de l' CloudWatch agent sur votre appareil, consultez le guide de CloudWatch l'utilisateur Amazon.

Lorsque vous êtes invité à saisir la configuration, sélectionnez l'une des configurations suivantes.

Important

Les `file_path` configurations suivantes supposent que l'emplacement de sortie de journalisation par défaut est utilisé.

Le chemin de fichier utilisé suppose que vous exécutez l'agent Amazon Kinesis Video Streams Edge à partir de l'emplacement `:download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/version`.

- Pour configurer l' CloudWatch agent afin qu'il télécharge les journaux et publie les métriques relatives à la RAM et au processeur de l'appareil, collez ce qui suit dans le fichier de configuration.

```
{
  "agent": {
    "run_as_user": "ubuntu",
    "metrics_collection_interval": 60
  },
  "metrics": {
    "metrics_collected": {
      "mem": {
        "measurement": [
          "mem_used_percent"
        ],
        "append_dimensions": {
          "IotThing": "YourIotThingName"
        }
      },
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          "usage_active"
        ],
        "totalcpu": true,
        "append_dimensions": {
          "IotThing": "YourIotThingName"
        }
      }
    }
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
```



```

    {
      "file_path": "download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/version/log/cpp_kvs_edge.log*",
      "log_group_name": "/aws/kinesisvideo/EdgeRuntimeAgent",
      "log_stream_name": "YourIoTThingName-cpp_kvs_edge.log"
    },
    {
      "file_path": "download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/version/log/cpp_kvs_streams.log*",
      "log_group_name": "/aws/kinesisvideo/EdgeRuntimeAgent",
      "log_stream_name": "YourIoTThingName-cpp_kvs_streams.log"
    },
    {
      "file_path": "download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/version/log/cpp_kvssink.log*",
      "log_group_name": "/aws/kinesisvideo/EdgeRuntimeAgent",
      "log_stream_name": "YourIoTThingName-cpp_kvssink.log"
    }
  ]
}
}
}
}
}

```

Exécutez l'agent Amazon Kinesis Video Streams Edge en tant que processus natif

Configurez l'agent Amazon Kinesis Video Streams Edge en tant que service systemd. Il s'agit d'une étape facultative.

systemd est un gestionnaire de systèmes et de services sur les appareils Linux. systemd est la méthode recommandée pour gérer le processus, car elle redémarrera l'agent Amazon Kinesis Video Streams Edge en cas d'erreur ou en cas de panne d'alimentation de l'appareil exécutant l'application.

Procédez comme suit :

Exécutez l'agent Amazon Kinesis Video Streams Edge en tant que processus natif

1. Créez un nouveau fichier `/etc/systemd/system` et nommez-le `aws.kinesisvideo.edge-runtime-agent.service`.

Collez ce qui suit :

```
[Unit]
Description=AWS Kinesis Video Streams edge agent
After=network.target
StartLimitBurst=3
StartLimitInterval=30

[Service]
Type=simple
Restart=on-failure
RestartSec=10
WorkingDirectory=/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion
Environment="GST_PLUGIN_PATH=/download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion"
Environment="LD_LIBRARY_PATH=/download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib"
...
Environment="AWS_IOT_CORE_DATA_ATS_ENDPOINT=data-account-specific-prefix.iot.aws-
region.amazonaws.com"
ExecStart=/usr/lib/jvm/java-11-amazon-corretto/bin/java --add-opens java.base/
jdk.internal.misc=ALL-UNNAMED -Dio.netty.tryReflectionSetAccessible=true -cp kvs-
edge-agent.jar:libs.jar com.amazonaws.kinesisvideo.edge.controller.ControllerApp

[Install]
WantedBy=multi-user.target
```

Pour plus d'informations sur les paramètres acceptés par le fichier de configuration du `systemd` service, consultez la [documentation](#).

Note

Ajoutez les variables d'environnement requises à l'. . . emplacement, comme indiqué dans [the section called "Création de l'agent Edge"](#).

2. Rechargez les fichiers de service pour inclure le nouveau service.

Tapez `sudo systemctl daemon-reload`.

3. Lancez le service .

Tapez `sudo systemctl start aws.kinesisvideo.edge-runtime-agent.service`.

4. Vérifiez l'état du service Amazon Kinesis Video Streams Edge Agent pour vérifier qu'il fonctionne.

Tapez `sudo systemctl status aws.kinesisvideo.edge-runtime-agent.service`.

Voici un exemple du résultat que vous allez voir.

```
aws.kinesisvideo.edge-runtime-agent.service - AWS Kinesis Video Streams edge agent
  Loaded: loaded (/etc/systemd/system/aws.kinesisvideo.edge-runtime-agent.service; disabled; vendor preset: enabled)
  Active: active (running) since Thu 2023-06-08 19:15:02 UTC; 6s ago
  Main PID: 506483 (java)
  Tasks: 23 (limit: 9518)
  Memory: 77.5M
  CPU: 4.214s
  CGroup: /system.slice/aws.kinesisvideo.edge-runtime-agent.service
          ##506483 /usr/lib/jvm/java-11-amazon-corretto/bin/java -cp kvs-edge-agent.jar:libs.jar com.amazonaws.kinesisvideo.edge.controller.ControllerApp
```

5. Inspectez les journaux pour détecter toute erreur.

Tapez `journalctl -e -u aws.kinesisvideo.edge-runtime-agent.service`.

6. Entrez `systemctl --help` la liste complète des options à utiliser pour gérer le `processussystemctl`.

Voici quelques commandes courantes pour gérer l'agent Amazon Kinesis Video Streams Edge :

- Pour redémarrer, tapez `sudo systemctl restart aws.kinesisvideo.edge-runtime-agent.service`.
- Pour arrêter, tapez `sudo systemctl stop aws.kinesisvideo.edge-runtime-agent.service`.
- Pour démarrer automatiquement à chaque redémarrage de l'appareil, tapez `sudo systemctl enable aws.kinesisvideo.edge-runtime-agent.service`.

Déployez l'agent Amazon Kinesis Video Streams Edge pour AWS IoT Greengrass

Cette section fournit un guide complet sur l'utilisation d'Amazon Kinesis Video Streams AWS IoT Greengrass avec. En combinant ces services, vous pouvez diffuser efficacement des vidéos depuis des appareils de pointe vers le cloud, permettant ainsi un large éventail d'applications dans les domaines de l'IoT, de la surveillance, etc.

Vous trouverez des informations détaillées sur :

- Configuration de votre environnement de développement
- Création d'un flux vidéo Kinesis
- Création et empaquetage d'une fonction Lambda
- Configuration de l'appareil principal Kinesis Video Streams
- Déploiement sur le périphérique principal
- Vérification de votre stream

Suivez ces étapes pour déployer l'agent Amazon Kinesis Video Streams Edge AWS IoT Greengrass afin d'enregistrer et de télécharger du contenu multimédia à partir de caméras IP.

Rubriques

- [Création d'une instance Amazon EC2 d'Ubuntu](#)
- [Configurez le AWS IoT Greengrass V2 périphérique principal sur l'appareil](#)
- [Créez les Amazon Kinesis Video Streams et AWS Secrets Manager ressources pour les URL RTSP de votre caméra IP](#)
- [Ajouter des autorisations au rôle de service d'échange de jetons \(TES\)](#)
- [Installer la AWS IoT Greengrass Composant Secret Manager sur l'appareil](#)
- [Déployez l'agent Amazon Kinesis Video Streams Edge AWS IoT Greengrass composant de l'appareil](#)
- [Installer la AWS IoT Greengrass composant du gestionnaire de journaux sur le périphérique](#)

Création d'une instance Amazon EC2 d'Ubuntu

Procédez comme suit pour créer une instance Ubuntu Amazon EC2.

Création d'une instance Amazon EC2 d'Ubuntu

1. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse.
<https://console.aws.amazon.com/ec2/>

Vérifiez que la région appropriée est sélectionnée.

2. Choisissez Launch Instances (Lancer les instances).

Renseignez les champs suivants :

- Nom — Entrez le nom de l'instance.
 - Images d'applications et de systèmes d'exploitation (Amazon Machine Image) : sélectionnez Ubuntu.
 - Type d'instance — Sélectionnez t2.large.
 - Connexion par paire de clés — Créez votre propre paire de clés.
 - Paramètres réseau : conservez les paramètres par défaut.
 - Configuration du stockage : augmentez le volume à 256 GiB.
 - Paramètres avancés : conservez les paramètres par défaut.
3. Lancez l'instance et utilisez-la en SSH.

Procédez comme suit :

1. Sélectionnez Instances dans le menu de navigation de gauche, puis sélectionnez l'ID de l'instance.
 2. Choisissez Connect en haut à droite.
 3. Choisissez le client SSH et suivez les instructions affichées à l'écran.
 4. Ouvrez un terminal et naviguez jusqu'au .pem fichier téléchargé (probablement dans~/Downloads).
 5. La première fois que vous suivez ces procédures, vous recevrez le message « L'authenticité de l'hôte (...) ne peut pas être établie ». Tapez « oui ».
4. Installez les bibliothèques système pour intégrer l'agent Amazon Kinesis Video Streams Edge sur l'instance.

```
wget -O- https://apt.corretto.aws/corretto.key | sudo apt-key add -  
sudo add-apt-repository 'deb https://apt.corretto.aws stable main'  
  
sudo apt-get update
```

```
sudo apt-get install -y gcc libssl-dev libcurl4-openssl-dev liblog4cplus-dev \  
libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev \  
gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-bad \  
gstreamer1.0-plugins-good gstreamer1.0-tools \  
unzip java-11-amazon-corretto-jdk maven
```

Important

Si un écran vous indique que certains services doivent être redémarrés, appuyez sur Entrée pour sélectionner OK.

Pour plus d'informations, consultez le guide de l'utilisateur [d'Amazon Corretto 11](#).

Configurez le AWS IoT Greengrass V2 périphérique principal sur l'appareil

Suivez ces procédures pour installer le logiciel Nucleus AWS IoT Greengrass principal sur l'instance Amazon EC2.

Configurez le AWS IoT Greengrass appareil principal

1. Connectez-vous au AWS Management Console, <https://console.aws.amazon.com/iot/>.

Vérifiez que la région appropriée est sélectionnée.

2. Dans le menu de navigation de gauche, sélectionnez Appareils Greengrass, Appareils Core.
3. Choisissez Configurer un appareil principal.
4. Effectuez les étapes affichées à l'écran.
 - Étape 1 : Enregistrez un appareil principal Greengrass. Entrez le nom de l'appareil.
 - Étape 2 : ajouter à un groupe d'objets pour appliquer un déploiement continu. Sélectionnez Aucun groupe.
 - Étape 3 : installez le logiciel Greengrass Core. Sélectionnez Linux.
 - Étape 3.1 : Installation de Java sur l'appareil

Java est installé dans le cadre de [the section called "Création d'une instance Ubuntu"](#).


Revenez à cette étape si Java n'est pas encore installé.

- Étape 3.2 : Copier les AWS informations d'identification sur l'appareil

Ouvrez l'bash/zshoption et collez les commandes d'exportation dans l'instance Amazon EC2.


- Étape 3.3 : Exécuter le programme d'installation

1. Copiez et exécutez les commandes Télécharger le programme d'installation et Exécuter le programme d'installation dans l'instance Ubuntu Amazon EC2.

 Note

La commande Exécuter le programme d'installation sera automatiquement mise à jour en fonction du nom que vous avez choisi à l'étape précédente.

2. Notez le rôle de service d'échange de jetons (TES) créé. Vous en aurez besoin ultérieurement.

 Note

Par défaut, le rôle créé est appelé GreengrassV2TokenExchangeRole.

Créez les Amazon Kinesis Video Streams et AWS Secrets Manager ressources pour les URL RTSP de votre caméra IP

Suivez ces procédures pour créer les flux et les secrets nécessaires dans AWS Secrets Manager. Effectuez d'abord cette étape, car vous avez besoin des ARN des ressources créées dans les politiques.

Création d'Amazon Kinesis Video Streams

Créez Amazon Kinesis Video Streams à l'aide AWS Management Console de l' AWS CLI API, ou.

Dans le AWS Management Console, ouvrez la console [Amazon Kinesis Video Streams](#). Choisissez Streams vidéo dans le menu de navigation de gauche.

Pour de plus amples informations, veuillez consulter [the section called "Création d'un flux vidéo Amazon Kinesis"](#).

Créez des secrets dans AWS Secrets Manager

Dans le AWS Management Console, ouvrez la [AWS Secrets Manager console](#). Choisissez Secrets dans le menu de navigation de gauche.

Vérifiez que la région appropriée est sélectionnée.

1. Choisissez Store a new secret (Stocker un nouveau secret).

a. Étape 1 : Choisissez le type de secret

- Sélectionnez Autre type de secret.
- Dans la section Key/Value Pairs, ajoutez une paire clé-valeur.

Clé : MediaURI

Note

La clé doit être MediaURI. Cela distingue les majuscules et minuscules. Si vous ne le saisissez pas correctement, l'application ne fonctionne pas.

Valeur : *Your MediaURI*.

Exemple

Exemple : `rtsp://<YourCameraIPAddress>:<YourCameraRTSPPort>/
YourCameraMediaURI`.

- Étape 2 : configurer le secret Donnez un nom à ce secret. Nomme-le comme tu veux.
- Étape 3 : Configuration de la rotation (facultatif). Choisissez Suivant.
- Étape 4 : Révision. Choisissez Stocker.

2. Si votre secret ne s'affiche pas immédiatement, sélectionnez le bouton d'actualisation.

Choisissez le nom de votre secret. Prenez note de l'ARN secret.

3. Répétez ce processus pour chaque MediaURI à partir duquel vous souhaitez diffuser.

Note

Le AWS réseau bloque certaines sources RTSP publiques. Vous ne pouvez pas y accéder depuis l'instance Amazon EC2 ou si vous les exécutez de manière non gérée lorsque vous êtes connecté au VPN.

⚠ Important

L'URL RTSP de votre caméra doit diffuser de la vidéo au format h.264. La durée du fragment ne doit pas dépasser la limite mentionnée dans [the section called “Quotas du SDK pour les producteurs”](#).

L'agent Amazon Kinesis Video Streams Edge prend uniquement en charge la vidéo.

Exécutez `gst-discoverer-1.0 Your RtspUrl` pour vous assurer que votre caméra est accessible depuis votre appareil.

Enregistrez les ARN de tous les flux et secrets que vous avez créés. Vous en aurez besoin pour passer à l'étape suivante.


Ajouter des autorisations au rôle de service d'échange de jetons (TES)

Accordez le rôle de service d'échange de jetons (TES) à l'appareil autorisé à consulter les secrets. Cela est nécessaire pour que le AWS Secrets Manager AWS IoT Greengrass composant fonctionne correctement.

Ajouter des autorisations au rôle TES

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
2. Choisissez Rôles dans le menu de navigation de gauche et recherchez le rôle TES que vous avez créé plus tôt dans le processus.
3. Dans le menu déroulant Ajouter des autorisations, sélectionnez Joindre des politiques.
4. Choisissez Create Policy (Créer une politique).
5. Faites défiler l'écran vers le bas et sélectionnez Modifier.
6. Dans l'éditeur de stratégie, choisissez JSON et modifiez la politique.

Remplacez la politique par la suivante :

 Note

Remplacez `arn:aws:kinesisvideo:*:*:stream/streamName1/*` et `arn:aws:kinesisvideo:*:*:stream/streamName2/*` par les ARN des flux que vous avez créés à l'étape précédente.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:ListStreams"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": [
        "arn:aws:kinesisvideo:*:*:stream/streamName1/*",
        "arn:aws:kinesisvideo:*:*:stream/streamName2/*"
      ]
    }
  ]
}
```

7. Sur la page Add tags (Ajouter des identifications), choisissez Next: Review (Suivant : Révision).
8. Donnez un nom à votre politique, puis choisissez Créer une politique.

Un exemple de nom de politique est KvsEdgeAccessPolicy.

9. Fermez l'onglet et revenez à l'onglet dans lequel vous associez une politique au rôle TES.


Cliquez sur le bouton d'actualisation, puis recherchez la politique nouvellement créée.

Cochez la case et choisissez Joindre les politiques.

Sur l'écran suivant, vous voyez une note indiquant que la politique a été correctement attachée au rôle.

10. Créez et joignez une autre politique, cette fois pour vos secrets.

Remplacez la politique par la suivante :

 Note

arn:aws:secretsmanager:*:*:secret:* Remplacez-les par les ARN contenant les secrets MediaURI que vous avez créés dans. [the section called "Créez des ressources pour les URL RTSP de vos caméras IP"](#)

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": [
        "arn:aws:secretsmanager:*:*:secret:*",
        "arn:aws:secretsmanager:*:*:secret:*"
      ]
    }
  ]
}
```

11. Créez et attachez une autre politique, cette fois pour les Amazon CloudWatch métriques.

Remplacez la politique par la suivante :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Installer la AWS IoT Greengrass Composant Secret Manager sur l'appareil

L'agent Amazon Kinesis Video Streams Edge nécessite que AWS IoT Greengrass le composant Secret Manager soit d'abord installé sur l'appareil.

Installation du composant Secret Manager

1. Connectez-vous à la AWS IoT Core console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/iot/>. Vérifiez que la région appropriée est sélectionnée.
2. Dans le menu de navigation de gauche, sélectionnez Appareils Greengrass, Déploiements.

Choisissez le déploiement avec le même objectif que celui dans lequel nous l'avons créé [the section called "Configuration de l'appareil AWS IoT Greengrass principal"](#).

3. Dans le menu déroulant Actions en haut à droite, sélectionnez Réviser.

Dans la fenêtre contextuelle qui apparaît, choisissez Revise le déploiement.

4. Complétez les sections suivantes :
 - Étape 1 : Spécifiez la cible Choisissez Suivant.
 - Étape 2 : Sélectionnez les composants

- Vérifiez que le composant AWS.Greengrass.cli est sélectionné. Ne désinstallez pas ce composant.
- Activez le bouton Afficher uniquement les composants sélectionnés et recherchez aws.greengrass. SecretManager.
- Cochez la case à côté de aws.greengrass. SecretManager, puis choisissez Next.
- Étape 3 : configurer les composants Configurez le composant AWS IoT Greengrass Secret Manager pour télécharger les secrets depuis l' AWS IoT Greengrass environnement.

Sélectionnez aws.greengrass. SecretManagercomposant, puis choisissez Configurer le composant.

Dans l'écran qui apparaît, mettez à jour les AWS Secrets Manager ARN dans la zone Configuration à fusionner.

Note

arn:aws:secretsmanager:*:*:secret:* Remplacez-les par les ARN des secrets que vous avez créés dans [the section called "Créez des ressources pour les URL RTSP de vos caméras IP"](#).

```
{
  "cloudSecrets": [
    {
      "arn": "arn:aws:secretsmanager:*:*:secret:*"
    },
    {
      "arn": "arn:aws:secretsmanager:*:*:secret:*"
    }
  ]
}
```

Note

cloudSecrets est une liste d'objets avec la clé arn. Pour plus d'informations, consultez la section relative à la [configuration du gestionnaire secret](#) dans le guide du AWS IoT Greengrass Version 2 développeur.

Lorsque vous avez terminé, sélectionnez Confirmer, puis Suivant.

- Étape 4 : configurer les paramètres avancés Sélectionnez Suivant.
 - Étape 5 : Révision. Sélectionnez Deploy (Déployer).
5. Vérifiez que le AWS Secrets Manager composant et les autorisations ont été correctement installés.

Sur l'instance Amazon EC2 d'Ubuntu, tapez `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.SecretManager` pour vérifier que le composant a reçu la configuration mise à jour.

6. Inspectez les AWS IoT Greengrass principaux journaux.

Tapez `sudo less /greengrass/v2/logs/greengrass.log`.

Vérifiez les erreurs de déploiement.

En cas d'erreur, revoyez le déploiement pour supprimer le `aws.greengrass.SecretManager` composant.

Tapez `sudo service greengrass restart` pour redémarrer le service AWS IoT Greengrass principal.

Si l'erreur de déploiement est liée à des autorisations manquantes, consultez la [the section called “Ajouter des autorisations au rôle TES”](#) section pour vous assurer que le rôle TES dispose des autorisations appropriées. Ensuite, répétez cette section.

7. Mettre à jour les secrets du composant AWS IoT Greengrass Secret Manager

 Important

Le composant AWS IoT Greengrass Secret Manager récupère et met en cache les secrets uniquement lorsque le déploiement est mis à jour.

Pour mettre à jour les secrets du composant AWS IoT Greengrass Secret Manager, suivez les étapes 1 à 6 précédentes, avec les modifications suivantes.

Étape 3 : configurer les composants Configurez le composant AWS IoT Greengrass Secret Manager pour télécharger les secrets depuis l' AWS IoT Greengrass environnement.

Sélectionnez `aws.greengrass.SecretManagercomposant`, puis choisissez Configurer le composant.

Dans l'écran qui apparaît, collez [""] le champ Réinitialiser les chemins et mettez à jour AWS Secrets Manager les ARN dans le champ Configuration pour fusionner.

Pour plus d'informations, voir [Réinitialiser les mises à jour](#).

Déployez l'agent Amazon Kinesis Video Streams Edge AWS IoT Greengrass composant de l'appareil

Procédez comme suit pour déployer le composant Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent sur l'appareil :

Déployer le composant

1. Téléchargez le `tar` fichier à l'aide du lien fourni.

Si vous avez rempli le formulaire d'intérêt pour Amazon Kinesis Video Streams Edge Agent, consultez vos e-mails pour trouver le lien de téléchargement. Si vous n'avez pas rempli le formulaire, complétez-le [ici](#).

2. Vérifiez le checksum.
3. Extrayez les fichiers binaires et le fichier `jar` de votre appareil.

```
Type :tar -xvf kvs-edge-agent.tar.gz.
```

Après l'extraction, la structure de vos dossiers ressemblera à ce qui suit :

```
kvs-edge-agent/LICENSE
kvs-edge-agent/THIRD-PARTY-LICENSES
kvs-edge-agent/pom.xml
kvs-edge-agent/KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/recipes
kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml
kvs-edge-agent/KvsEdgeComponent/artifacts
kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion
```

```

kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/edge_log_config

kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/kvs-edge-agent.jar
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libgstkvssink.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libIngestorPipelineJNI.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libcproducer.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libKinesisVideoProducer.so

```

Note

Le nom du dossier de version doit être configuré de manière à refléter le dernier numéro de version binaire. Par exemple, le nom du dossier d'une version 1.0.0 sera défini comme 1.0.0.

4. Créez le fichier jar de dépendances.

Note

Le fichier jar inclus dans le fichier kvs-edge-agent.tar.gz ne contient pas les dépendances. Suivez les étapes ci-dessous pour créer ces bibliothèques.

Accédez au kvs-edge-agent dossier qui contient pom.xml.

Tapez `mvn clean package`.

Cela générera un fichier jar contenant les dépendances requises par l'agent Amazon Kinesis Video Streams Edge. `kvs-edge-agent/target/libs.jar`

5. Placez le fichier libs.jar dans le dossier contenant les artefacts du composant.

Tapez `mv ./target/libs.jar ./KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/`.

6. Facultatif. Configurez les propriétés. L'agent Amazon Kinesis Video Streams Edge accepte les variables d'environnement suivantes AWS IoT Greengrass en mode :

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|---|
| AWS_REGION | Oui | La région utilisée. Exemple : us-west-2 AWS IoT Greengrass Le logiciel principal définit automatiquement cette valeur pour vous. Pour plus d'informations, consultez la rubrique de référence sur les variables d'environnement des composants dans le Guide du AWS IoT Greengrass Version 2 développeur. |

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|--|
| GST_PLUGIN_PATH | Oui | <p>Chemin du fichier pointant vers le dossier contenant les bibliothèques gstreamer dépendantes de la IngestorPipelineJNI plate-forme et Cela permet à GStreamer de charger ces plugins. Pour de plus amples informations, veuillez consulter the section called “Téléchargez, compilez et configurez l'élément GStreamer”.</p> <p>Exemple : <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /</i></p> |

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|---|
| LD_LIBRARY_PATH | Oui | <p>Chemin du fichier pointant vers le répertoire contenant les bibliothèques <code>cproducer</code> dépendantes de la <code>KinesisVideoProducer</code> plate-forme et</p> <p>Exemple : <code>/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/</code></p> |
| AWS_KVS_EDGE_CLOUD_WATCH_ENABLED | Non | <p>Détermine si l'agent Amazon Kinesis Video Streams Edge publiera les indicateurs Amazon CloudWatch de santé au travail sur.</p> <p>Valeurs acceptées : TRUE/FALSE (sans distinction majuscules/minuscules). La valeur par défaut est FALSE si elle n'est pas fournie.</p> <p>Exemple : FAUX</p> |

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|---|
| AWS_KVS_EDGE_LOG_LEVEL | Non | <p>Niveau de journalisation des sorties de l'agent Amazon Kinesis Video Streams Edge.</p> <p>Valeurs acceptées :</p> <ul style="list-style-type: none">• OFF• ALL• FATAL• ERROR• WARN• INFO, par défaut, s'il n'est pas fourni• DEBUG• TRACE <p>Exemple : INFO</p> |
| AWS_KVS_EDGE_LOG_MAX_FILE_SIZE | Non | <p>Une fois que le fichier journal atteint cette taille, un roulement se produit.</p> <ul style="list-style-type: none">• Minimum : 1• Maximum : 100• Par défaut : 20, s'il n'est pas fourni• Unités : mégaoctets (Mo) <p>Exemple : 5</p> |

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|---|
| AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY | Non | <p>Le chemin du fichier pointant vers le répertoire dans lequel les journaux de l'agent Amazon Kinesis Video Streams Edge sont générés. La valeur par défaut est <code>./log</code> si elle n'est pas fournie.</p> <p>Exemple : <code>./file/path/</code></p> |
| AWS_KVS_EDGE_LOG_ROLLOVER_COUNT | Non | <p>Le nombre de journaux reportés à conserver avant de les supprimer.</p> <ul style="list-style-type: none">• Minimum : 1• Maximum : 100• Par défaut : 10, s'il n'est pas fourni <p>Exemple : 20</p> |
| AWS_KVS_EDGE_RECORDING_DIRECTORY | Non | <p>Le chemin du fichier pointant vers le répertoire dans lequel le support enregistré sera écrit. La valeur par défaut est le répertoire actuel s'il n'est pas fourni.</p> <p>Exemple : <code>./file/path/</code></p> |

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|---|
| GREENGRASS_ROOT_DIRECTORY | Non | <p>Le chemin du fichier vers le répertoire AWS IoT Greengrass racine.</p> <p>Ce paramètre est défini par défaut <code>/greengrass/v2/</code> s'il n'est pas fourni.</p> <p>Exemple : <code>/file/path/</code></p> |
| GST_DEBUG | Non | <p>Spécifie le niveau des journaux GStreamer à générer. Pour plus d'informations, consultez la documentation de GStreamer.</p> <p>Exemple : 0</p> |
| GST_DEBUG_FILE | Non | <p>Spécifie le fichier de sortie des journaux de débogage de GStreamer. S'ils ne sont pas définis, les journaux de débogage sont renvoyés à une erreur standard. Pour plus d'informations, consultez la documentation de GStreamer.</p> <p>Exemple :</p> <p><code>/tmp/gstreamer-logging.log</code></p> |

Ouvrez `kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml` et modifiez le script d'exécution pour ajouter l'une des variables d'environnement précédentes.

⚠ Important

Assurez-vous que le script d'exécution modifié ne contient aucun caractère de tabulation. Le logiciel AWS IoT Greengrass principal ne pourra pas lire la recette.

7. Déployez le composant Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent.

Type :

```
sudo /greengrass/v2/bin/greengrass-cli deployment create \  
  --recipeDir <download location>/kvs-edge-agent/KvsEdgeComponent/recipes/ \  
  --artifactDir <download location>/kvs-edge-agent/KvsEdgeComponent/artifacts/ \  
  --merge "aws.kinesisvideo.KvsEdgeComponent=EdgeAgentVersion"
```

Pour plus d'informations, consultez les sections suivantes du guide du AWS IoT Greengrass Version 2 développeur :

- [AWS IoT Greengrass Commandes CLI](#)
- [Déployer AWS IoT Greengrass des composants sur des appareils](#)

8. Envoyez les configurations à l'application à l'aide du AWS CLI.

- a. Créez un nouveau fichier, *example-edge-configuration.json*.

Collez le code suivant dans le fichier. Il s'agit d'un exemple de configuration qui enregistre tous les jours de 9 h 00 à 16 h 59 min 59 s (en fonction de l'heure système de votre appareil). AWS IoT II télécharge également les médias enregistrés tous les jours de 19 h 00 à 21 h 59 min 59 s.

Pour de plus amples informations, veuillez consulter [StartEdgeConfigurationUpdate](#).

```
{  
  "StreamARN": "arn:aws:kinesisvideo:your-region:your-account-id:stream/your-stream/0123456789012",  
  "EdgeConfig": {  
    "HubDeviceArn": "arn:aws:iot:your-region:your-account-id:thing/kvs-edge-agent-demo",  
    "RecorderConfig": {  
      "MediaSourceConfig": {
```

```

        "MediaUriSecretArn": "arn:aws:secretsmanager:your-region:your-account-id:secret:your-secret-dRbHJQ",
        "MediaUriType": "RTSP_URI"
    },
    "ScheduleConfig": {
        "ScheduleExpression": "0 0 9,10,11,12,13,14,15,16 ? * * **",
        "DurationInSeconds": 3599
    }
},
"UploaderConfig": {
    "ScheduleConfig": {
        "ScheduleExpression": "0 0 19,20,21 ? * * **",
        "DurationInSeconds": 3599
    }
},
"DeletionConfig": {
    "EdgeRetentionInHours": 15,
    "LocalSizeConfig": {
        "MaxLocalMediaSizeInMB": 2800,
        "StrategyOnFullSize": "DELETE_OLDEST_MEDIA"
    },
    "DeleteAfterUpload": true
}
}
}

```

- b. Entrez ce qui suit AWS CLI pour envoyer le fichier à l'agent Amazon Kinesis Video Streams Edge :

```
aws kinesisvideo start-edge-configuration-update --cli-input-json
"file://example-edge-configuration.json"
```

9. Répétez l'étape précédente pour chaque flux pour l'agent Amazon Kinesis Video Streams Edge.

Installer la AWS IoT Greengrass composant du gestionnaire de journaux sur le périphérique

Note

Soyez conscient des [CloudWatchquotas](#).

Suivez ces procédures pour configurer les journaux de l'agent Amazon Kinesis Video Streams Edge afin qu'ils soient automatiquement téléchargés CloudWatch à l'aide AWS IoT Greengrass du composant du gestionnaire de journaux. Il s'agit d'une étape facultative.

Installer la AWS IoT Greengrass composant du gestionnaire de journaux

1. Vérifiez que le rôle de l' AWS IoT Greengrass appareil dispose des [autorisations appropriées](#).
 - a. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
 - b. Cliquez sur Rôles dans le menu de navigation de gauche.
 - c. Choisissez le nom du rôle TES créé dans [the section called "Configuration de l'appareil AWS IoT Greengrass principal"](#). Utilisez la barre de recherche si nécessaire.
 - d. Sélectionnez la GreengrassV2TokenExchangeRoleAccess politique.
 - e. Sélectionnez l'onglet JSON et vérifiez que la politique ressemble à ce qui suit :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams",
        "s3:GetBucketLocation"
      ],
      "Resource": "*"
    }
  ]
}
```

- f. Si la GreengrassV2TokenExchangeRoleAccess politique n'existe pas ou si certaines autorisations requises sont manquantes, créez une nouvelle stratégie IAM avec ces autorisations et associez-la au rôle TES créé dans [the section called "Configuration de l'appareil AWS IoT Greengrass principal"](#).


2. Connectez-vous à la AWS IoT Core console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/iot/>. Vérifiez que la région appropriée est sélectionnée.
3. Dans le menu de navigation de gauche, sélectionnez Appareils Greengrass, Déploiements.

Choisissez le déploiement avec le même objectif que celui dans lequel vous l'avez créé [the section called "Configuration de l'appareil AWS IoT Greengrass principal"](#).

4. Dans le coin supérieur droit, sélectionnez Actions, puis sélectionnez Réviser.

Dans la fenêtre contextuelle qui apparaît, choisissez Revise le déploiement.

5. Complétez les sections suivantes :
 - a. Étape 1 : Spécifiez la cible Choisissez Suivant.
 - b. Étape 2 : Sélectionnez les composants
 - i. Vérifiez que le composant AWS.Greengrass.cli et aws.greengrass.SecretManagerles composants sont toujours sélectionnés.

 Important

Ne désinstallez pas ces composants.

- ii. Activez le bouton Afficher uniquement les composants sélectionnés et recherchez aws.greengrass.LogManager.
 - iii. Cochez la case à côté de aws.greengrass.LogManager, puis choisissez Next.
 - c. Étape 3 : configurer les composants Configurez le composant du gestionnaire de AWS IoT Greengrass journaux pour télécharger les journaux générés par l'agent Amazon Kinesis Video Streams Edge.

Sélectionnez aws.greengrass.LogManagercomposant, puis choisissez Configurer le composant.

Dans l'écran qui apparaît, collez la configuration suivante du gestionnaire de journaux dans la zone Configuration à fusionner.

```
{
  "logsUploaderConfiguration": {
    "componentLogsConfigurationMap": {
      "aws.kinesisvideo.KvsEdgeComponent/java_kvs.log": {
        "diskSpaceLimit": "100",
```

```

        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "java_kvs.log\\w*"
    },
    "aws.kinesisvideo.KvsEdgeComponent/cpp_kvs_edge.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "cpp_kvs_edge.log\\w*"
    },
    "aws.kinesisvideo.KvsEdgeComponent/cpp_kvssink.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "cpp_kvssink.log\\w*"
    },
    "aws.kinesisvideo.KvsEdgeComponent/cpp_kvs_streams.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "cpp_kvs_streams.log\\w*"
    }
}
},
"periodicUploadIntervalSec": "1"
}

```

Important

`logFileDirectoryPath` Dans la configuration précédente, on suppose que l'emplacement de sortie de journalisation par défaut est utilisé.

Note

Pour plus d'informations sur chacun des paramètres de configuration du gestionnaire de journaux, consultez la section [Gestionnaire de journaux](#) du guide du AWS IoT Greengrass Version 2 développeur.

Une fois que vous avez terminé, sélectionnez Confirmer, puis cliquez sur Suivant.

- d. Étape 4 : configurer les paramètres avancés Sélectionnez Suivant.
 - e. Étape 5 : Révision. Sélectionnez Deploy (Déployer).
6. Vérifiez que le composant du gestionnaire de AWS journaux et les autorisations ont été correctement installés.
 7. Sur l'instance Amazon EC2 d'Ubuntu, tapez `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.LogManager` pour vérifier que le composant a reçu la configuration mise à jour.
 8. Inspectez les AWS IoT Greengrass principaux journaux.

Tapez `sudo less /greengrass/v2/logs/greengrass.log`.

Vérifiez les erreurs de déploiement.

En cas d'erreur, revoyez le déploiement pour supprimer le `aws.greengrass.LogManager` composant.

Tapez `sudo service greengrass restart` pour redémarrer le service AWS IoT Greengrass principal.

Si l'erreur de déploiement est liée à des autorisations manquantes, vérifiez [the section called "Ajouter des autorisations au rôle TES"](#) que le rôle TES dispose des autorisations appropriées. Ensuite, répétez cette section.

FAQ sur l'agent Amazon Kinesis Video Streams Edge

Voici quelques questions fréquemment posées au sujet du service Amazon Kinesis Video Streams Edge Agent.

Quels sont les systèmes d'exploitation pris en charge par Amazon Kinesis Video Streams Edge Agent ?

Amazon Kinesis Video Streams Edge Agent prend actuellement en charge les systèmes d'exploitation suivants :

Ubuntu

- 22.x
 - AMD64
- 18.x
 - ARM

AL 2

- amzn2
 - AMD64 amazonlinux:2.0.20210219.0-amd64 (Snowball)

L'agent Amazon Kinesis Video Streams Edge est-il compatible avec les médias H.265 ?

Amazon Kinesis Video Streams Edge Agent ne H.264 prend en charge que les flux élémentaires.

L'agent Amazon Kinesis Video Streams Edge fonctionne-t-il dans AL2 ?

Oui.

Comment puis-je exécuter plusieurs flux dans le AWS IoT objet ou appareil ?

Envoyez-en un autre [StartEdgeConfigurationUpdate](#) vers des AWS Secrets Manager streams/ARN Amazon Kinesis identiques `HubDeviceArn`, mais différents.

Comment puis-je modifier un document une **StartEdgeConfigurationUpdate** fois qu'il a été envoyé ?

Envoyez une mise [StartEdgeConfigurationUpdate](#) à jour `HubDeviceArn` avec le même ARN Amazon Kinesis Video Streams. Lorsque l'application reçoit le message d'Amazon Kinesis Video Streams, elle remplace la configuration précédente pour ce flux. Des modifications auront alors lieu.

Avez-vous des exemples courants **ScheduleConfigs** ?

L'agent Amazon Kinesis Video Streams Edge utilise l'heure système de l'appareil sur lequel il s'exécute.

| Description | ScheduleExpression | DurationInSeconds |
|---|------------------------------------|-------------------|
| 24/7 enregistrement, téléchargement toutes les heures | (null ScheduleConfig) | |
| 9 h 00 à 16 h 59 min 59 s tous les jours | 0 0 9-16 * * ? * | 3599 |
| 9 h 00 à 16 h 59 min 59 s en semaine | 0 0 9-16 ? * 2-6 * | 3599 |
| | 0 0 9-16 ? * 2,3,4,5,6 * | 3599 |
| | 0 0 9-16 ? * MON-FRI * | 3599 |
| | 0 0 9-16 ? * MON,TUE,WED,THU,FRI * | 3599 |
| De 9 h 00 à 16 h 59 min 59 s les fins de semaine | 0 0 9-16 ? * SAT,SUN * | 3599 |
| 22 h 00 - 23 h 59 min 59 s en semaine | 0 0 22,23 ? * MON-FRI * | 3599 |
| 9 h 00 à 10 h 00 tous les jours | 0 0 9 * * ? * | 3600 |
| 16 h 00 à 17 h 59 min 59 s tous les jours | 0 0 16-17 * * ? * | 3599 |

Pour plus d'exemples, consultez la [documentation de Quartz](#).

Y a-t-il une limite de diffusion maximale ?

L'agent Amazon Kinesis Video Streams Edge est actuellement soumis à une limite stricte de 16 flux par appareil. Utilisez l'[DeleteEdgeConfiguration](#) API pour supprimer les flux d'un appareil. La mise à jour d'une configuration pour le même flux à l'aide du [StartEdgeConfigurationUpdate](#) n'augmente pas le nombre de flux de l'appareil.

Comment puis-je redémarrer une tâche qui a donné lieu à une erreur ?

En cas d'erreur, l'agent Amazon Kinesis Video Streams Edge tentera de redémarrer la tâche. Toutefois, en cas d'erreurs (telles que des erreurs de configuration), vous devez redémarrer le travail manuellement.

Pour déterminer les tâches qui doivent être redémarrées manuellement, consultez la FatalError métrique dans [the section called “Surveillez l'agent Amazon Kinesis Video Streams Edge avec CloudWatch”](#).

Renvoyez-le [StartEdgeConfigurationUpdate](#) pour redémarrer le travail pour le stream.

Comment puis-je surveiller l'état de santé de mon agent Amazon Kinesis Video Streams Edge ?

Pour de plus amples informations, veuillez consulter [the section called “Surveillez l'agent Amazon Kinesis Video Streams Edge avec CloudWatch”](#).

Diffusez des vidéos via un VPC

Cette version bêta est disponible en avant-première dans la région Europe (Paris), eu-west-3. Pour accéder à ces composants et à notre guide de démarrage, [envoyez-nous un e-mail](#).

Le service de point de terminaison Amazon Kinesis Video Streams VPC vous permet de diffuser et de consommer des vidéos via le réseau Amazon sans qu'aucune donnée ne transite par l'Internet public.

Pour demander l'accès, [envoyez-nous les informations suivantes par e-mail](#) :

- ID de compte
- Diffusez des ARN
- ID du VPC

Note

Cela peut prendre jusqu'à une semaine pour que nous vous ajoutions au service.

Si vous n'avez jamais utilisé de points de terminaison VPC par le passé, consultez les informations suivantes pour vous familiariser avec le concept :

- [AWS PrivateLink arrière-plan](#)
- [Guide de démarrage en VPC](#)

Informations supplémentaires

Une fois que vous aurez été ajouté à la version bêta, nous vous enverrons par e-mail un lien vers des informations supplémentaires sur cette fonctionnalité.

Procédures relatives aux points de terminaison VPC

Quotas

Les principales différences de quotas sont les suivantes :

- Quota inférieur pour toutes les API de bande passante (2 Mbits/s) :
 - PutMedia
 - GetMedia
 - GetMediaForFragmentList
- 10 streams autorisés par client

Créer un point de terminaison

Une fois que vous serez autorisé dans la liste, vous recevrez le nom du service de point de terminaison VPC pour Amazon Kinesis Video Streams. Cela ressemblera à `com.amazonaws.region.kinesisvideo`.

Créez un point de [terminaison VPC d'interface](#) pour Amazon Kinesis Video Streams à l'aide de la console Amazon VPC ou du (). AWS Command Line Interface AWS CLI

Dans le AWS CLI, saisissez ce qui suit :

```
aws ec2 create-vpc-endpoint \  
--vpc-id customer-provided-vpc-id \  
--service-name com.amazonaws.eu-west-2.kinesisvideo \  
--private-dns-enabled
```

Important

Le trafic au sein de votre VPC utilisera un DNS privé pour passer par le point de terminaison. Si vous ne l'activez pas, vous devrez implémenter votre propre logique DNS. Pour plus d'informations sur le DNS privé, consultez [AWS PrivateLink la documentation](#).

Pour plus d'informations sur AWS CLI cette option, consultez [create-vpc-endpoint](#).

Contrôlez l'accès aux terminaux

Vous pouvez associer une politique de point de terminaison à votre point de terminaison VPC qui contrôle l'accès à Amazon Kinesis Video Streams. La politique spécifie les informations suivantes :

- le principal qui peut effectuer des actions,
- les actions qui peuvent être effectuées, et
- les ressources sur lesquelles les actions peuvent être effectuées.

Pour plus d'informations, consultez la section [Contrôle de l'accès aux services avec des points de terminaison VPC à l'aide de politiques relatives aux points de terminaison dans le Guide. AWS PrivateLink](#)

Voici un exemple de politique relative aux terminaux pour Amazon Kinesis Video Streams. Lorsqu'elle est attachée à un point de terminaison, cette politique refuse l'accès aux PutMedia actions répertoriées pour tous les principaux sur toutes les ressources.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Deny",
      "Action": [
        "kinesisvideo:PutMedia"
      ],
      "Resource": "*"
    }
  ]
}
```

Exemples d'Amazon Kinesis Video Streams

Les exemples de code suivants montrent comment utiliser l'API Kinesis Video Streams :

Exemples : envoi de données vers Kinesis Video Streams

- [Exemple : Plug-in GStreamer du SDK Kinesis Video Streams Producer - kvssink](#): montre comment créer le SDK de production Kinesis Video Streams à utiliser comme destination GStreamer.
- [Exécutez l'élément GStreamer dans un conteneur Docker](#): montre comment utiliser une image Docker prédéfinie pour envoyer une vidéo RTSP (Real-Time Streaming Protocol) depuis une caméra IP vers Kinesis Video Streams.
- [Exemple : diffusion en continu depuis une source RTSP](#): montre comment créer votre propre image Docker et envoyer une vidéo RTSP depuis une caméra IP vers Kinesis Video Streams.
- [Exemple : envoi de données à Kinesis Video Streams à l'aide PutMedia de l'API](#): montre comment utiliser le pour envoyer [Utiliser la bibliothèque Java Producer](#) à Kinesis Video Streams des données déjà au format conteneur (MKV) à l'aide [PutMedia](#) de l'API.

Exemples : extraction de données depuis Kinesis Video Streams

- [KinesisVideoExample](#): montre comment analyser et enregistrer des fragments vidéo à l'aide de la bibliothèque d'analyseurs Kinesis Video Streams.
- [Exemple : analyse et rendu de fragments Kinesis Video Streams: montre comment analyser et afficher des fragments de flux vidéo Kinesis à l'aide de JCodec et de JFrame.](#)

Exemples : lecture de données vidéo

- [Exemple : utilisez HLS en HTML et JavaScript](#): montre comment récupérer une session de streaming HLS pour un flux vidéo Kinesis et la lire sur une page Web.

Conditions préalables

- Dans l'exemple de code, vous fournissez des informations d'identification en spécifiant un profil que vous définissez dans votre fichier de profil AWS d'informations d'identification ou en fournissant des informations d'identification dans les propriétés du système Java de votre environnement

de développement intégré (IDE). Si ce n'est pas déjà fait, configurez d'abord vos informations d'identification. Pour plus d'informations, voir [Configurer les AWS informations d'identification et la région pour le développement](#).

- Nous vous recommandons d'utiliser un IDE Java pour afficher et exécuter le code, par exemple :
 - [Eclipse Java Néon](#)
 - [JetBrains IntelliJ IDEA](#)

Exemple : Plug-in GStreamer du SDK Kinesis Video Streams

Producer - kvssink

Cette rubrique explique comment créer le SDK Amazon Kinesis Video Streams Producer à utiliser comme plugin GStreamer.

Rubriques

- [Téléchargez, compilez et configurez l'élément GStreamer](#)
- [Exécutez l'élément GStreamer](#)
- [Exemples de commandes de lancement de GStreamer](#)
- [Exécutez l'élément GStreamer dans un conteneur Docker](#)
- [Référence des paramètres de l'élément GStreamer](#)

[GStreamer](#) est un framework multimédia populaire utilisé par plusieurs caméras et sources vidéo pour créer des pipelines multimédias personnalisés en combinant des plugins modulaires. Le plugin Kinesis Video Streamer rationalise l'intégration de votre pipeline multimédia GStreamer existant à Kinesis Video Streams. Après avoir intégré GStreamer, vous pouvez diffuser des vidéos depuis une webcam ou une caméra RTSP (Real Time Streaming Protocol) vers Kinesis Video Streams pour une lecture, un stockage et une analyse plus approfondie en temps réel ou ultérieurement.

Le plugin GStreamer gère automatiquement le transfert de votre flux vidéo vers Kinesis Video Streams en encapsulant les fonctionnalités fournies par le SDK du producteur Kinesis Video Streams dans un élément récepteur GStreamer, `kvssink`. L'infrastructure GStreamer fournit un environnement géré standard pour créer un flux multimédia à partir d'un appareil tel qu'une caméra ou que d'autres sources vidéo en vue d'un traitement, rendu ou stockage ultérieurs.

Le pipeline GStreamer se compose généralement du lien entre une source (caméra vidéo) et l'élément récepteur (un joueur pour restituer la vidéo ou un stockage pour une récupération

hors ligne). Dans cet exemple, vous utilisez le kit SDK Producteur en tant que récepteur ou une destination multimédia pour votre source vidéo (webcam ou caméra IP). L'élément du plugin qui encapsule le SDK envoie ensuite le flux vidéo à Kinesis Video Streams.

Cette rubrique explique comment créer un pipeline multimédia GStreamer capable de diffuser des vidéos à partir d'une source vidéo, telle qu'une caméra Web ou un flux RTSP, généralement connectée via des étapes de codage intermédiaires (à l'aide de l' H.264 encodage) à Kinesis Video Streams. Lorsque votre flux vidéo est disponible sous forme de flux vidéo Kinesis, vous pouvez l'utiliser [the section called “Diffuser à l'aide de la bibliothèque d'analyseurs”](#) pour poursuivre le traitement, la lecture, le stockage ou l'analyse de votre flux vidéo.

Téléchargez, compilez et configurez l'élément GStreamer

L'exemple du plugin GStreamer est inclus dans le SDK de production Kinesis Video Streams C++. Pour plus d'informations sur les prérequis et le téléchargement du kit SDK, consultez [Téléchargez et configurez le code de la bibliothèque C++ Producer](#).

Vous pouvez créer le récepteur GStreamer du kit de développement GStreamer sous forme de bibliothèque dynamique sur macOS, Ubuntu, Raspberry Pi ou Windows. Le plug-in GStreamer se trouve dans votre répertoire `build`. Pour charger ce plugin, il doit se trouver dans votre `GST_PLUGIN_PATH`. Exécutez la commande suivante :

```
export GST_PLUGIN_PATH=`pwd`/build
```

Note

Sur macOS, vous pouvez uniquement diffuser en continu du contenu vidéo à partir d'une caméra en réseau lors de l'exécution de GStreamer dans un conteneur Docker. Le streaming de contenu vidéo à partir d'une caméra USB sur macOS dans un conteneur Docker n'est pas pris en charge.

Exécutez l'élément GStreamer

Pour exécuter GStreamer avec l'élément SDK du producteur Kinesis Video Streams comme récepteur, utilisez la commande. `gst-launch-1.0` Utilisez des éléments en amont adaptés à

l'utilisation du plugin GStreamer. Par exemple, [v4l2src](#) pour les appareils v4l2 sur les systèmes Linux ou [rtspsrc](#) pour les appareils RTSP. Spécifiez `kvssink` en tant qu'élément récepteur (destination finale du pipeline) pour envoyer des vidéos au kit SDK Producteur.

En plus de [fournir des informations d'identification](#) et [une région](#), l'élément `kvssink` possède le paramètre obligatoire suivant :

- `stream-name`— Le nom du Kinesis Video Streams de destination.

Pour plus d'informations sur les paramètres facultatifs `kvssink`, consultez [Référence des paramètres de l'élément GStreamer](#).

Pour obtenir les dernières informations sur les plugins et les paramètres de GStreamer, consultez la section Plugins [GStreamer](#). Vous pouvez également utiliser le `gst-inspect-1.0` suivi du nom d'un élément ou d'un plugin GStreamer pour imprimer ses informations et vérifier qu'il est disponible sur votre appareil :

```
gst-inspect-1.0 kvssink
```

Si la construction `kvssink` échoue ou si `GST_PLUGIN_PATH` n'est pas correctement défini, votre sortie ressemblera à ceci :

```
No such element or plugin 'kvssink'
```

Exemples de commandes de lancement de GStreamer

Les exemples suivants montrent comment utiliser le plugin `kvssink` GStreamer pour diffuser des vidéos à partir de différents types d'appareils.

Exemple 1 : Diffuser une vidéo depuis une caméra RTSP sur Ubuntu

La commande suivante crée un pipeline GStreamer sur Ubuntu qui diffuse en continu à partir d'une caméra RTSP en réseau, à l'aide du plug-in GStreamer [rtspsrc](#) :

```
gst-launch-1.0 -v rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !  
rtph264depay ! h264parse ! kvssink stream-name="YourStreamName" storage-size=128
```

Exemple 2 : Encoder et diffuser une vidéo à partir d'une caméra USB sur Ubuntu

La commande suivante crée un pipeline GStreamer sur Ubuntu qui encode au H.264 format le flux provenant d'une caméra USB et le diffuse vers Kinesis Video Streams. Cet exemple utilise le plugin GStreamer [v4l2src](#).

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert ! video/x-raw,format=I420,width=640,height=480,framerate=30/1 ! x264enc bframes=0 key-int-max=45 bitrate=500 ! video/x-h264,stream-format=avc,alignment=au,profile=baseline ! kvssink stream-name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

Exemple 3 : Diffuser une vidéo pré-encodée à partir d'une caméra USB sur Ubuntu

La commande suivante crée un pipeline GStreamer sur Ubuntu qui diffuse des vidéos déjà encodées par la caméra au H.264 format Kinesis Video Streams. Cet exemple utilise le plugin GStreamer [v4l2src](#).

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! h264parse ! video/x-h264,stream-format=avc,alignment=au ! kvssink stream-name="plugin" storage-size=512 access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

Exemple 4 : Diffuser une vidéo à partir d'une caméra réseau sous macOS

La commande suivante crée un pipeline GStreamer sur macOS qui diffuse des vidéos vers Kinesis Video Streams à partir d'une caméra réseau. L'exemple suivant illustre le plug-in GStreamer [rtspsrc](#).

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE ! rtpH264depay ! h264parse ! video/x-h264, format=avc,alignment=au ! kvssink stream-name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

Exemple 5 : Diffuser une vidéo à partir d'une caméra réseau sous Windows

La commande suivante crée un pipeline GStreamer sous Windows qui diffuse des vidéos vers Kinesis Video Streams à partir d'une caméra réseau. L'exemple suivant illustre le plug-in GStreamer [rtspsrc](#).

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE ! rtpH264depay ! video/x-h264, format=avc,alignment=au ! kvssink stream-
```

```
name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-  
key="YourSecretKey" aws-region="YourAWSRegion"
```

Exemple 6 : Diffuser une vidéo depuis une caméra sur un Raspberry Pi

La commande suivante crée un pipeline GStreamer sur le Raspberry Pi qui diffuse des vidéos vers Kinesis Video Streams. Cet exemple utilise le plugin GStreamer [v4l2src](#).

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert !  
video/x-raw,format=I420,width=640,height=480,framerate=30/1 !  
omxh264enc control-rate=1 target-bitrate=5120000 periodicity-  
idr=45 inline-header=FALSE ! h264parse ! video/x-h264,stream-  
format=avc,alignment=au,width=640,height=480,framerate=30/1,profile=baseline ! kvssink  
stream-name="YourStreamName" access-key="YourAccessKey" secret-key="YourSecretKey"  
aws-region="YourAWSRegion"
```

Exemple 7 : Diffuser de l'audio et de la vidéo dans Raspberry Pi et Ubuntu

Découvrez comment [exécuter la commande gst-launch-1.0 pour démarrer le streaming audio et vidéo](#) dans Ubuntu. Raspberry-Pi

Exemple 8 : Diffuser du son et de la vidéo à partir des sources de l'appareil sous macOS

Découvrez comment [exécuter la commande gst-launch-1.0 pour commencer à diffuser en streaming du contenu audio et vidéo dans MacOS](#).

Exemple 9 : télécharger un fichier MKV contenant à la fois de l'audio et de la vidéo

Découvrez comment [exécuter la commande gst-launch-1.0 pour télécharger le fichier MKV ayant du contenu audio et vidéo](#). Vous aurez besoin d'un fichier de test MKV avec des supports codés H.264 et AAC.

Exécutez l'élément GStreamer dans un conteneur Docker

Docker est une plateforme permettant de développer, de déployer et d'exécuter des applications à l'aide de conteneurs. L'utilisation de Docker pour créer le pipeline GStreamer normalise l'environnement d'exploitation de Kinesis Video Streams, ce qui rationalise la création et l'utilisation de l'application.

Pour installer et configurer Docker, consultez les éléments suivants :

- [Instructions pour télécharger Docker](#)
- [Démarrez avec Docker](#)

Après avoir installé Docker, vous pouvez télécharger le SDK Kinesis Video Streams C++ Producer (et le plugin GStreamer) depuis Amazon Elastic Container Registry à l'aide de l'une des commandes ci-dessous. `docker pull`

Pour exécuter GStreamer avec l'élément SDK du producteur Kinesis Video Streams comme récepteur dans un conteneur Docker, procédez comme suit :

Rubriques

- [Authentifiez votre client Docker](#)
- [Téléchargez l'image Docker pour Ubuntu, macOS, Windows ou Raspberry Pi](#)
- [Exécutez l'image Docker](#)

Authentifiez votre client Docker

Authentifiez votre client Docker auprès du registre Amazon ECR à partir duquel l'image doit être extraite. Vous devez obtenir des jetons d'authentification pour chaque registre utilisé. Les jetons sont valides pendant 12 heures. Pour plus d'informations, veuillez consulter [Authentification de registre](#) dans le Guide de l'utilisateur Amazon Elastic Container Registry.

Exemple Authentification avec Amazon ECR

Pour vous authentifier auprès d'Amazon ECR, copiez et collez la commande suivante comme indiqué.

```
sudo aws ecr get-login-password --region us-west-2 | docker login -u AWS --password-stdin https://546150905175.dkr.ecr.us-west-2.amazonaws.com
```

En cas de succès, la sortie imprime `Login Succeeded`.

Téléchargez l'image Docker pour Ubuntu, macOS, Windows ou Raspberry Pi

Téléchargez l'image Docker dans votre environnement Docker à l'aide d'une des commandes suivantes en fonction de votre système d'exploitation :

Téléchargez l'image Docker pour Ubuntu

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux:latest
```

Téléchargez l'image Docker pour macOS

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux:latest
```

Téléchargez l'image Docker pour Windows

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-windows:latest
```

Téléchargez l'image Docker pour Raspberry Pi

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-raspberry-pi:latest
```

Pour vérifier que l'image a été ajoutée avec succès, utilisez la commande suivante :

```
docker images
```

Exécutez l'image Docker

Utilisez l'une des commandes suivantes pour exécuter l'image Docker en fonction de votre système d'exploitation :

Exécutez l'image Docker sur Ubuntu

```
sudo docker run -it --network="host" --device=/dev/video0 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

Exécutez l'image Docker sur macOS

```
sudo docker run -it --network="host" 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

Exécutez l'image Docker sous Windows

```
docker run -it 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-windows AWS_ACCESS_KEY_ID AWS_SECRET_ACCESS_KEY RTSP_URL STREAM_NAME
```

Exécutez l'image Docker sur Raspberry Pi

```
sudo docker run -it --device=/dev/video0 --device=/dev/vchiq -v /opt/vc:/opt/vc 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-raspberry-pi /bin/bash
```

Docker lance le conteneur et vous présente une invite de commande pour utiliser les commandes dans le conteneur.

Dans le conteneur, définissez les variables d'environnement à l'aide de la commande suivante :

```
export LD_LIBRARY_PATH=/opt/awssdk/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:$LD_LIBRARY_PATH
export PATH=/opt/awssdk/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/bin:$PATH
export GST_PLUGIN_PATH=/opt/awssdk/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:$GST_PLUGIN_PATH
```

Commencez à diffuser en kvssink utilisant le `gst-launch-1.0` pour exécuter un pipeline adapté à votre appareil et à votre source vidéo. Par exemple, les pipelines, voir [Exemples de commandes de lancement de GStreamer](#).

Référence des paramètres de l'élément GStreamer

Pour envoyer une vidéo au SDK C++ Amazon Kinesis Video Streams Producer, vous devez le `kvssink` spécifier comme récepteur ou destination finale du pipeline. Cette référence fournit des informations sur les paramètres obligatoires et facultatifs de l'élément `kvssink`. Pour de plus amples informations, veuillez consulter [the section called "Plug-in GStreamer - kvssink"](#).

Rubriques

- [the section called "Fournir des informations d'identification à kvssink"](#)
- [the section called "Fournir une région à kvssink"](#)
- [the section called "paramètres optionnels de kvssink"](#)

Fournir des informations d'identification à **kvssink**

Pour autoriser l'élément `kvssink` GStreamer à envoyer des demandes AWS, fournissez des AWS informations d'identification à utiliser lorsqu'il appelle le service Amazon Kinesis Video Streams. La chaîne de fournisseurs d'informations d'identification recherche les informations d'identification dans l'ordre suivant :

1. AWS IoT informations d'identification

Pour configurer les AWS IoT informations d'identification, voir [the section called “Contrôle de l'accès aux ressources Kinesis Video Streams à l'aide de AWS IoT”](#).

La valeur du `iot-credentials` paramètre doit commencer par `iot-certificate`, et être suivie d'une liste séparée par des virgules des paires *key = value* suivantes.

| Clé | Obligatoire | Description |
|------------------------|-------------|--|
| <code>ca-path</code> | Oui | Chemin du fichier vers le certificat CA utilisé pour établir la confiance avec le service principal via TLS. Exemple Exemple : <code>/file/path/to/certificate.pem</code> |
| <code>cert-path</code> | Oui | Chemin du fichier vers le X.509 certificat. Exemple Exemple : <code>/file/path/to/certificateID -certificate.pem.crt</code> |
| <code>endpoint</code> | Oui | Le point de terminaison AWS IoT Core du fournisseur |

| Clé | Obligatoire | Description |
|--------------|-------------|---|
| | | <p>d'informations d'identification pour votre AWS compte. Consultez le guide AWS IoT du développeur.</p> <p>Exemple</p> <p>Exemple : <i>credential-account-specific-prefix</i> .credentials.iot. <i>aws-region</i> .amazonaws.com</p> |
| key-path | Oui | <p>Chemin du fichier vers la clé privée utilisée dans la paire de public/private clés.</p> <p>Exemple</p> <p>Exemple : <i>/file/path/to/certificateID</i> -private.pem.key</p> |
| role-aliases | Oui | <p>Nom de l'alias de rôle pointant vers le rôle AWS IAM à utiliser lors de la connexion. AWS IoT Core</p> <p>Exemple</p> <p>Exemple : <i>KvsCameraIoTRoleAlias</i></p> |

| Clé | Obligatoire | Description |
|----------------|-------------|--|
| iot-thing-name | Non | L'option <code>iot-thing-name</code> est facultative. Si <code>iot-thing-name</code> ce n'est pas le cas, la valeur du <code>stream-name</code> paramètre est utilisée. Exemple Exemple : <i>kvs_examp le_camera</i> |

Exemple

Exemple :

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
  iot-certificate="iot-certificate,endpoint=credential-account-specific-
  prefix.credentials.iot.aws-region.amazonaws.com,cert-path=certificateID-
  certificate.pem.crt,key-path=certificateID-private.pem.key,ca-
  path=certificate.pem,role-aliases=YourRoleAlias,iot-thing-name=YourThingName"
```

2. Variables d'environnement

Pour `kvssink` utiliser les informations d'identification de l'environnement, définissez les variables d'environnement suivantes :

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|--|
| AWS_ACCESS_KEY_ID | Oui | La clé AWS d'accès utilisée pour accéder à Amazon Kinesis Video Streams. |
| AWS_SECRET_ACCESS_KEY | Oui | La clé AWS secrète associée à la clé d'accès. |

| Nom de la variable d'environnement | Obligatoire | Description |
|------------------------------------|-------------|--|
| AWS_SESSION_TOKEN | Non | Spécifie la valeur du jeton de session requis si vous utilisez des informations d'identification de sécurité temporaires provenant directement AWS STS des opérations. |

La définition de la variable d'environnement permet de modifier la valeur utilisée jusqu'à la fin de votre session shell, ou jusqu'à ce que vous définissiez la variable sur une autre valeur. Pour que les variables soient persistantes au cours des futures sessions, définissez-les dans le script de démarrage de votre shell.

3. paramètres de clé d'accès, de clé secrète

Pour spécifier les informations d'identification directement en tant que `kvssink` paramètre, définissez les paramètres suivants :

| kvssink Nom du paramètre | Obligatoire | Description |
|---------------------------------|-------------|--|
| access-key | Oui | La clé AWS d'accès utilisée pour accéder à Amazon Kinesis Video Streams. |
| secret-key | Oui | La clé AWS secrète associée à la clé d'accès. |
| session-token | Non | Spécifie la valeur du jeton de session requis si vous utilisez des informations d'identification de sécurité temporaires provenant directement AWS STS des opérations. |

Exemple

À l'aide d'informations d'identification statiques :

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
  access-key="AKIDEXAMPLE" secret-key="SKEEXAMPLE"
```

Exemple

À l'aide d'informations d'identification temporaires :

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
  access-key="AKIDEXAMPLE" secret-key="SKEEXAMPLE" session-token="STEXAMPLE"
```

4. Fichier d'informations d'identification

Important

Si vous avez sélectionné l'une des méthodes précédentes, vous ne pouvez pas utiliser le `credential-path` `kvssink` paramètre.

| kvssink Nom du paramètre | Obligatoire | Description |
|---------------------------------|-------------|--|
| <code>credential-path</code> | Oui | Chemin d'accès au fichier texte contenant les informations d'identification dans un format spécifique. |

Le fichier texte doit contenir des informations d'identification dans l'un des formats suivants :

- INFORMATIONS D'IDENTIFICATION *YourAccessKey YourSecretKey*
- INFORMATIONS D'IDENTIFICATION *YourAccessKey Expiration YourSecretKey SessionToken*

Exemple

Exemple : votre `credentials.txt` fichier se trouve à l'adresse suivante `/home/ubuntu` et contient les éléments suivants :

```
CREDENTIALS AKIDEXAMPLE 2023-08-10T22:43:00Z SKEXAMPLE STEXAMPLE
```

Pour l'utiliser dans `kvssink`, tapez :

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
credential-path="/home/ubuntu/credentials.txt"
```

Note

Le délai d'expiration doit être d'au moins $5 + 30 + 3 = 38$ secondes dans le futur. Le délai de grâce est défini comme la `IOT_CREDENTIAL_FETCH_GRACE_PERIOD` variable dans [IotCredentialProvider.h](#). Si les informations d'identification sont trop proches de leur date d'expiration lorsque vous démarrez `kvssink`, le code d'erreur s'affiche `0x52000049 - STATUS_INVALID_TOKEN_EXPIRATION`.

Important

`kvssink` ne modifie pas le fichier d'informations d'identification. Si vous utilisez des informations d'identification temporaires, le fichier d'informations d'identification doit être mis à jour par une source externe avant la date d'expiration, déduction faite du délai de grâce.


Fournir une région à `kvssink`

L'ordre de recherche des régions est le suivant :



1. `AWS_DEFAULT_REGION` La variable d'environnement est examinée en premier. Si elle est définie, cette région est utilisée pour configurer le client.
2. `aws-region` le paramètre est ensuite revu. Si elle est définie, cette région est utilisée pour configurer le client.
3. Si aucune des méthodes précédentes n'a été utilisée, la `kvssink` valeur par défaut est `us-west-2`

paramètres optionnels de kvssink

Les paramètres facultatifs pour l'élément `kvssink` sont les suivants. Pour obtenir plus d'informations sur ces paramètres, consultez [Structures de flux vidéo Kinesis](#).

| Paramètre | Description | Unité / Type | Par défaut |
|--------------------------------------|--|--------------|------------|
| <code>stream-name</code> | <p>Nom du flux vidéo Amazon Kinesis de destination.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Important</p> <p>Si aucun nom de flux n'est spécifié, le nom de flux par défaut sera utilisé : « DEFAULT_STREAM ». Si un flux portant ce nom par défaut n'existe pas déjà, il sera créé.</p> </div> | | |
| <code>absolute-fragment-times</code> | Indique s'il faut utiliser des fragments absolus. | Booléen | true |
| <code>access-key</code> | <p>La clé AWS d'accès utilisée pour accéder à Kinesis Video Streams.</p> <p>Vous devez soit avoir défini les AWS</p> | | |

| Paramètre | Description | Unité / Type | Par défaut |
|-------------------|--|------------------|------------|
| | <p>informations d'identification, soit fournir ce paramètre. Pour fournir ces informations, saisissez ce qui suit :</p> <pre>export AWS_ACCESS_KEY_ID=</pre> | | |
| avg-bandwidth-bps | Bande passante moyenne attendue pour le flux. | Bits par seconde | 4194304 |


| Paramètre | Description | Unité / Type | Par défaut |
|-------------------------|---|--------------|-------------|
| <code>aws-region</code> | <p>Le Région AWS à utiliser.</p> <div data-bbox="472 352 792 1381"><p> Note</p><p>Vous pouvez également fournir la variable d'AWS_DEFAULT_REGION environnement à la région. Les variables d'environnement sont prioritaires si la variable d'environnement et les paramètres kvssink sont définis.</p></div> <div data-bbox="472 1451 792 1864"><p> Important</p><p>La région sera définie par <code>us-west-2</code> défaut sur sauf indication contraire.</p></div> | String | "us-west-2" |

| Paramètre | Description | Unité / Type | Par défaut |
|--------------------------------------|---|--------------------|-------------------|
| <code>buffer-duration</code> | Durée de la mémoire tampon du flux. | Secondes | 120 |
| <code>codec-id</code> | L'ID de codec du flux. | String | "V_MPEG4/ISO/AVC" |
| <code>connection-staleness</code> | Le délai après lequel le rappel de l'obsolescence du flux est appelé. | Secondes | 60 |
| <code>content-type</code> | Le type de contenu du flux. | String | "video/h264" |
| <code>fragment-acks</code> | Indique s'il faut utiliser des ACK de fragment. | Booléen | true |
| <code>fragment-duration</code> | La durée de fragment souhaitée. | Millisecondes | 2000 |
| <code>framerate</code> | Débit d'images attendu. | Images par seconde | 25 |
| <code>frame-timecodes</code> | Indique s'il faut utiliser codes horaires d'image ou générer des horodatages à l'aide du rappel actuel. | Booléen | true |
| <code>key-frame-fragmentation</code> | Indique s'il faut générer des fragments sur une image clé. | Booléen | true |

| Paramètre | Description | Unité / Type | Par défaut |
|---------------------|--|--------------|----------------------------|
| log-config | Le chemin d'accès de configuration du journal. | String | "../kvs_log_configuration" |
| max-latency | La latence maximum pour le flux. | Secondes | 60 |
| recalculate-metrics | Indique s'il faut recalculer les métriques. | Booléen | true |
| replay-duration | La durée de restauration du lecteur actuel pour relire en cas d'erreur si le redémarrage est activé. | Secondes | 40 |
| restart-on-error | Indique s'il faut redémarrer en cas d'erreur. | Booléen | true |
| retention-period | La durée pendant laquelle le flux est conservé. | heures | 2 |
| rotation-period | La période de rotation des clés d'accès. Pour plus d'informations, voir Rotation AWS KMS des touches . | Secondes | 3600 |

| Paramètre | Description | Unité / Type | Par défaut |
|---------------|---|----------------|------------|
| secret-key | <p>La clé AWS secrète utilisée pour accéder à Kinesis Video Streams.</p> <p>Vous devez soit avoir défini les AWS informations d'identification, soit fournir ce paramètre.</p> <pre>export AWS_SECRET_ACCESS_KEY=</pre> | | |
| session-token | <p>Spécifie la valeur du jeton de session requis si vous utilisez des informations d'identification de sécurité temporaires provenant directement AWS STS des opérations.</p> | | |
| storage-size | <p>Taille de stockage de l'appareil en mébioctet (MiB). Pour plus d'informations sur la configuration du stockage de l'appareil, consultez StorageInfo.</p> | Mebiocet (MiB) | 128 |

| Paramètre | Description | Unité / Type | Par défaut |
|-----------------------------|---|------------------------------|-----------------|
| <code>streaming-type</code> | <p>Le type de streaming . Les valeurs valides sont les suivantes :</p> <ul style="list-style-type: none">• 0 : temps réel• 1 : temps quasi réel (non pris en charge actuellement)• 2 : hors ligne | Enum GstKvsSinkStreamingType | 0 : temps réel |
| <code>timecode-scale</code> | L'échelle de code horaire MKV. | Millisecondes | 1 |
| <code>track-name</code> | Nom de la piste MKV. | String | "kinesis_video" |

| Paramètre | Description | Unité / Type | Par défaut |
|------------------------------|---|--------------|------------|
| <code>iot-certificate</code> | <p>AWS IoT informations d'identification à utiliser dans l'élément.</p> <p><code>iot-certificate</code> accepte les clés et les valeurs suivantes :</p> <div data-bbox="472 667 792 1318" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p><code>iot-thing-name</code> C'est facultatif. Si <code>iot-thing-name</code> ce n'est pas le cas, la valeur du <code>stream-name</code> paramètre est utilisée.</p> </div> <ul style="list-style-type: none"> • <code>endpoint=iotcredentialsproviderendpoint</code> • <code>cert-path=/localdirectorypath/to/certificate</code> • <code>key-path=/localdir</code> | String | Aucune |

| Paramètre | Description | Unité / Type | Par défaut |
|-----------|--|--------------|------------|
| | <p>ectorypath / to/private/ key</p> <ul style="list-style-type: none"> • ca-path=/ localdir ectorypath/ to/ca-cert • role-aliases =role-aliases • iot-thing-name=YourIotThingName | | |

Exemple : envoi de données à Kinesis Video Streams à l'aide PutMedia de l'API

Cet exemple montre comment utiliser l'[PutMedia](#) API. Il montre comment envoyer des données déjà au format conteneur (MKV). Si vos données doivent être assemblées dans un format conteneur avant d'être envoyées (par exemple, si vous assemblez les données vidéo de la caméra en images), voir [Transférer vers Kinesis Video Streams](#).

Note

L'PutMedia opération n'est disponible que dans les SDK C++ et Java. Cela est dû à la gestion en duplex intégral des connexions, du flux de données et des accusés de réception. Il n'est pas pris en charge dans les autres langues.

Cet exemple comprend les étapes suivantes :

- [Téléchargez et configurez le code](#)
- [Rédiger et examiner le code](#)
- [Exécutez et vérifiez le code](#)

Téléchargez et configurez le code

Suivez les étapes pour télécharger l'exemple de code Java, importer le projet dans votre IDE Java, configurer les emplacements des bibliothèques et configurer le code pour utiliser vos AWS informations d'identification.

1. Créez un répertoire et clonez l'exemple de code source à partir du GitHub référentiel. L'exemple PutMedia fait partie de [Java](#).

```
git clone https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java
```

2. Ouvrez l'IDE Java que vous utilisez (par exemple, [Eclipse](#) ou [IntelliJ IDEA](#)) et importez le projet Apache Maven que vous avez téléchargé :
 - Dans Eclipse : Choisissez Fichier, Importer, Maven, Projets Maven existants et accédez à la racine du package téléchargé. Sélectionnez le fichier pom.xml.
 - Dans IntelliJ Idea : Choisissez Importer. Naviguez vers le fichier pom.xml dans la racine du package téléchargé.

Pour plus d'informations, consultez la documentation relative à IDE.

3. Mettez à jour le projet de manière à ce que IDE trouve l'emplacement des bibliothèques que vous avez importées.
 - Pour IntelliJ IDEA, procédez comme suit :
 - a. Ouvrez le menu contextuel (via un clic droit) du répertoire lib du projet, puis choisissez Ajouter en tant que bibliothèque.
 - b. Choisissez Fichier, puis Structure du projet.
 - c. Sous Paramètres du projet, choisissez Modules.
 - d. Dans l'onglet Sources, définissez Niveau de langue sur **7** ou une valeur supérieure.
 - Pour Eclipse, procédez comme suit :
 - a. Ouvrez le menu contextuel (clic droit) du projet et choisissez Propriétés, Chemin de création Java, Source. Ensuite, procédez comme suit :
 1. Sur l'onglet Source, double-cliquez sur Emplacement de la bibliothèque native.
 2. Dans l'assistant Configuration du dossier de la bibliothèque Native, choisissez Workspace.

3. Dans la sélection Dossier de la bibliothèque native, choisissez le répertoire lib dans le projet.
- b. Ouvrez le menu contextuel (clic droit) pour le projet et choisissez Propriétés. Ensuite, procédez comme suit :
1. Dans l'onglet Bibliothèques, choisissez Ajouter Jars.
 2. Dans l'assistant Sélection JAR, choisissez tous les fichiers .jars dans le répertoire lib du projet.

Rédiger et examiner le code

L'exemple d'API PutMedia () présente le modèle de codage suivant :PutMediaDemo

Rubriques

- [Créez le PutMediaClient](#)
- [Diffuser du média et interrompre le thread](#)

Les exemples de code de cette section proviennent de la classe PutMediaDemo.

Créez le PutMediaClient

La création de PutMediaClient l'objet nécessite les paramètres suivants :

- L'URI pour le point de terminaison PutMedia.
- Un InputStream indiquant le fichier MKV à diffuser.
- Nom du flux. Cet exemple utilise le flux qui a été créé dans [Utiliser la bibliothèque Java Producer](#) (my-stream). Pour utiliser un flux différent, modifiez le paramètre suivant :

```
private static final String STREAM_NAME="my-stream";
```

Note

L'exemple PutMedia d'API ne crée pas de flux. Vous devez créer un flux soit à l'aide de l'application de test pour la [Utiliser la bibliothèque Java Producer](#) console Kinesis Video Streams, soit AWS CLI à l'aide du.

- L'horodatage actuel.
- Le type de time code. L'exemple utilise RELATIVE, indiquant que l'horodatage est relatif au début du conteneur.
- Un objet `AWSKinesisVideoV4Signer` qui vérifie que les paquets reçus ont été envoyés par l'expéditeur autorisé.
- Le montant maximum de la bande passante en Kbps.
- Un objet `AckConsumer` pour recevoir les accusés de réception des paquets reçus.

Le code suivant crée l'objet `PutMediaClient` :

```
/* actually URI to send PutMedia request */
final URI uri = URI.create(KINESIS_VIDEO_DATA_ENDPOINT + PUT_MEDIA_API);

/* input stream for sample MKV file */
final InputStream inputStream = new FileInputStream(MKV_FILE_PATH);

/* use a latch for main thread to wait for response to complete */
final CountDownLatch latch = new CountDownLatch(1);

/* a consumer for PutMedia ACK events */
final AckConsumer ackConsumer = new AckConsumer(latch);

/* client configuration used for AWS SigV4 signer */
final ClientConfiguration configuration = getClientConfiguration(uri);

/* PutMedia client */
final PutMediaClient client = PutMediaClient.builder()
    .putMediaDestinationUri(uri)
    .mkvStream(inputStream)
    .streamName(STREAM_NAME)
    .timestamp(System.currentTimeMillis())
    .fragmentTimeCodeType("RELATIVE")
    .signWith(getKinesisVideoSigner(configuration))
    .upstreamKbps(MAX_BANDWIDTH_KBPS)
    .receiveAcks(ackConsumer)
    .build();
```

Diffuser du média et interrompre le thread

Une fois le client créé, l'échantillon commence la diffusion asynchrone avec `putMediaInBackground`. Le thread principal est alors interrompu avec `latch.await` jusqu'au retour de `AckConsumer`, le client est alors fermé.

```
/* start streaming video in a background thread */
    client.putMediaInBackground();

    /* wait for request/response to complete */
    latch.await();

    /* close the client */
    client.close();
```

Exécutez et vérifiez le code

Pour exécuter l'exemple d'API `PutMedia`, procédez comme suit :

1. Créez un flux nommé `my-stream` dans la console Kinesis Video Streams ou à l'aide AWS CLI du.
2. Remplacez votre répertoire de travail par le répertoire du kit SDK Producteur Java :

```
cd /<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-video-streams-
producer-sdk-java/
```

3. Compilez le kit SDK Java et l'application de démonstration :

```
mvn package
```

4. Créez un nom de fichier temporaire dans le répertoire `/tmp` :

```
jar_files=$(mktemp)
```

5. Créez une chaîne de chemin de classe de dépendances à partir du référentiel local vers un fichier :

```
mvn -Dmdep.outputFile=$jar_files dependency:build-classpath
```

6. Définissez la valeur de la variable d'environnement `LD_LIBRARY_PATH` comme suit :

```
export LD_LIBRARY_PATH=/<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-  
video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:  
$LD_LIBRARY_PATH  
$ classpath_values=$(cat $jar_files)
```

7. Exécutez la démo à partir de la ligne de commande comme suit, en fournissant vos AWS informations d'identification :

```
java -classpath target/kinesisvideo-java-demo-1.0-SNAPSHOT.jar:$classpath_values -  
Daws.accessKeyId=${ACCESS_KEY} -Daws.secretKey=${SECRET_KEY} -Djava.library.path=  
opt/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build  
com.amazonaws.kinesisvideo.demoapp.DemoAppMain
```

8. Ouvrez la console [Kinesis Video Streams](#) et choisissez votre flux sur la page Gérer les flux. La vidéo est lue dans le volet Aperçu vidéo.

Exemple : diffusion en continu depuis une source RTSP

[C++](#)Il contient la définition d'un conteneur [Docker](#) qui se connecte à une caméra réseau RTSP (Real-Time Streaming Protocol). L'utilisation de Docker normalise l'environnement d'exploitation de Kinesis Video Streams, ce qui rationalise la création et l'utilisation de l'application.

La procédure suivante illustre comment configurer et utiliser l'application de démonstration RTSP.

Rubriques

- [Didacticiels vidéo](#)
- [Conditions préalables](#)
- [Créez l'image Docker](#)
- [Exécutez l'exemple d'application RTSP](#)

Didacticiels vidéo

Cette vidéo explique comment configurer un Raspberry Pi pour envoyer des flux RTSP vers le AWS cloud et Amazon Kinesis Video Streams. Il s'agit d'une démonstration de bout en bout.

Cette vidéo montre comment capturer des images à partir d'un flux pour utiliser la vision par ordinateur et Amazon Rekognition pour traiter les images et envoyer des alertes.

Conditions préalables

Pour exécuter l'exemple d'application Kinesis Video Streams RTSP, vous devez disposer des éléments suivants :

- Docker : Pour plus d'informations sur l'installation et l'utilisation de Docker, consultez les liens suivants :
 - [Instructions pour télécharger Docker](#)
 - [Démarrez avec Docker](#)
- Source de la caméra RTSP en réseau : Pour plus d'informations sur les caméras recommandées, consultez [Configuration système requise](#).

Créez l'image Docker

Créez d'abord l'image Docker dans laquelle l'application de démonstration sera exécutée.

1. Clonez le référentiel de démos d'Amazon Kinesis Video Streams.

```
git clone https://github.com/aws-samples/amazon-kinesis-video-streams-demos.git
```

2. Accédez au répertoire contenant le Dockerfile. Dans ce cas, il s'agit du répertoire [docker-rtsp](#).

```
cd amazon-kinesis-video-streams-demos/producer-cpp/docker-rtsp/
```

3. Utilisez la commande suivante pour créer l'image Docker. Cette commande crée l'image et la balise `rtspdockertest`.

```
docker build -t rtspdockertest .
```

4. Exécutez `docker images` et recherchez l'identifiant de l'image marqué `avecrtspdockertest`.

Par exemple, dans l'exemple de sortie ci-dessous, le `IMAGE ID` est `54f0d65f69b2`.

| REPOSITORY | TAG | IMAGE ID | CREATED | PLATFORM | SIZE |
|----------------|--------|--------------|----------------|-------------|-----------|
| rtspdockertest | latest | 54f0d65f69b2 | 10 minutes ago | linux/arm64 | 653.1 MiB |

Vous en aurez besoin ultérieurement.

Exécutez l'exemple d'application RTSP

Vous pouvez exécuter l'exemple d'application RTSP à l'intérieur ou à l'extérieur du conteneur Docker. Suivez les instructions appropriées ci-dessous.

Rubriques

- [Dans le conteneur Docker](#)
- [À l'extérieur du conteneur Docker](#)

Dans le conteneur Docker

Exécutez l'exemple d'application RTSP

1. Démarrez le conteneur Docker Amazon Kinesis Video Streams à l'aide de la commande suivante :

```
docker run -it YourImageId /bin/bash
```

2. Pour démarrer l'exemple d'application, entrez vos informations AWS d'identification, le nom du flux vidéo Amazon Kinesis et l'URL de la caméra réseau RTSP.

Important

Si vous utilisez des informations d'identification temporaires, vous devrez également fournir votre `AWS_SESSION_TOKEN`. Voir le deuxième exemple ci-dessous.

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId  
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId  
export AWS_DEFAULT_REGION=YourAWSRegion  
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

Informations d'identification temporaires :

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId
```

```
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId
export AWS_SESSION_TOKEN=YourSessionToken
export AWS_DEFAULT_REGION=YourAWSRegion
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

3. Connectez-vous à la console [Kinesis Video Streams AWS Management Console](#) et ouvrez-la.
Visionnez le stream.
4. Pour quitter le conteneur Docker, fermez la fenêtre du terminal ou tapez `exit`.

À l'extérieur du conteneur Docker

Depuis l'extérieur du conteneur Docker, utilisez la commande suivante :

```
docker run -it YourImageId /bin/bash -c "export AWS_ACCESS_KEY_ID=YourAccessKeyId;  
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId; export  
AWS_SESSION_TOKEN=YourSessionToken; export AWS_DEFAULT_REGION=Your AWS Region; ./  
kvs_gstreamer_sample YourStreamName YourRtspUrl"
```

Exemple : analyse et rendu de fragments Kinesis Video Streams

[Diffuser à l'aide de la bibliothèque d'analyseurs](#) Il contient une application de démonstration nommée `KinesisVideoRenderExample` qui explique l'analyse et le rendu de fragments de flux vidéo Amazon Kinesis. L'exemple utilise [JCodec](#) pour décoder les trames H.264 codées ingérées à l'aide de l'application. [Exemple : Plug-in GStreamer du SDK Kinesis Video Streams Producer - kvssink](#) Une fois que l'image est décodée à l'aide de JCodec, l'image visible est rendue à l'aide de [JFrame](#).

Cet exemple montre comment effectuer les opérations suivantes :

- Récupérez des images d'un flux vidéo Kinesis à l'aide de l'`GetMediaAPI` et restituez le flux pour le visionner.
- Affichez le contenu vidéo des flux dans une application personnalisée au lieu d'utiliser la console Kinesis Video Streams.

Vous pouvez également utiliser les classes de cet exemple pour afficher le contenu d'un flux vidéo Kinesis qui n'est pas codé H.264, tel qu'un flux de fichiers JPEG qui ne nécessitent pas de décodage avant d'être affichés.

La procédure suivante montre comment configurer et utiliser la démonstration du moteur de rendu.

Conditions préalables

Pour examiner et utiliser l'exemple de bibliothèque de moteur de rendu, vous avez besoin des éléments suivants :

- Un compte Amazon Web Services (AWS). Si vous n'avez pas encore de AWS compte, consultez [Getting Started with Kinesis Video Streams](#).
- Un environnement de développement intégré (IDE) Java, tel qu'[Eclipse Java Neon](#) ou [JetBrains IntelliJ Idea](#).

Exemple d'exécution du moteur de rendu

1. Créez un répertoire, puis clonez l'exemple de code source à partir du GitHub référentiel.

```
git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. Ouvrez l'IDE Java que vous utilisez (par exemple, [Eclipse](#) ou [IntelliJ IDEA](#)) et importez le projet Apache Maven que vous avez téléchargé :
 - Dans Eclipse : Choisissez Fichier, Importer, Maven, Projets Maven existants. Accédez au répertoire `kinesis-video-streams-parser-lib`.
 - Dans IntelliJ Idea : Choisissez Importer. Naviguez vers le fichier `pom.xml` dans la racine du package téléchargé.

Note

Si IntelliJ ne parvient pas à trouver vos dépendances, vous pourriez avoir à exécuter les tâches suivantes :

- Nettoyage de build : Choisissez File (Fichier), Settings (Paramètre), Build, Execution, Deployment (Génération, exécution, déploiement), Compiler (Compilateur). Vérifiez que l'option Effacer le répertoire de sortie lors de la reconstruction est sélectionnée, puis choisissez Build, Build Project.
- Réimportation du projet : Ouvrez le menu contextuel (clic droit) du projet, puis choisissez Maven, Reimport (Réimporter).

Pour plus d'informations, consultez la documentation relative à IDE.

3. À partir de votre IDE Java, ouvrez `src/test/java/com.amazonaws.kinesisvideo.parser/examples/KinesisVideoRendererExampleTest`.
4. Supprimez la directive `@Ignore` du fichier.
5. Mettez à jour le `.stream` paramètre avec le nom de votre flux vidéo Kinesis.
6. Exécutez le test `KinesisVideoRendererExample`.

Comment ça marche

L'exemple d'application illustre les éléments suivants :

- [Envoi des données MKV](#)
- [Analyse de fragments MKV en cadres](#)
- [Décodage et affichage du cadre](#)

Envoi des données MKV

Cet exemple envoie des exemples de données MKV à partir du fichier `rendering_example_video.mkv` à `PutMedia`, en utilisant pour envoyer des données vidéo à un flux nommé `render-example-stream`.

L'application crée un objet `PutMediaWorker` :

```
PutMediaWorker putMediaWorker = PutMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    inputStream,
    streamOps.amazonKinesisVideo);
executorService.submit(putMediaWorker);
```

Pour obtenir des informations sur la classe `PutMediaWorker`, consultez [Appel PutMedia](#) dans la documentation de [Diffuser à l'aide de la bibliothèque d'analyseurs](#).

Analyse de fragments MKV en cadres

Cet exemple récupère et analyse ensuite les fragments MKV depuis le flux à l'aide d'un objet `GetMediaWorker` :

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    streamOps.amazonKinesisVideo,
    getMediaProcessingArgumentsLocal().getFrameVisitor());
executorService.submit(getMediaWorker);
```

Pour plus d'informations sur la classe `GetMediaWorker`, consultez [Appel GetMedia](#) dans la documentation de [Diffuser à l'aide de la bibliothèque d'analyseurs](#).

Décodage et affichage du cadre

L'exemple décode et affiche ensuite l'image à l'aide de [JFrame](#).

L'exemple de code suivant provient de la classe `KinesisVideoFrameViewer`, qui étend `JFrame` :

```
public void setImage(BufferedImage bufferedImage) {
    image = bufferedImage;
    repaint();
}
```

L'image est affichée sous la forme d'une instance de [java.awt.image. BufferedImage](#). Pour des exemples montrant comment travailler avec `BufferedImage`, consultez [Reading/Loading une image](#).

Sécurité dans Amazon Kinesis Video Streams

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficierez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le [modèle de responsabilité partagée](#) décrit cette notion par les termes sécurité du cloud et sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. L'efficacité de notre sécurité est régulièrement testée et vérifiée par des auditeurs tiers dans le cadre des [programmes de conformité AWS](#). Pour en savoir plus sur les programmes de conformité qui s'appliquent à Kinesis Video Streams, [AWS consultez la section Services concernés par programme de conformité](#).
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre organisation, et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de Kinesis Video Streams. Les rubriques suivantes expliquent comment configurer Kinesis Video Streams pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui peuvent vous aider à surveiller et à sécuriser vos ressources Kinesis Video Streams.

Rubriques

- [Protection des données dans Kinesis Video Streams](#)
- [Contrôle de l'accès aux ressources Kinesis Video Streams à l'aide d'IAM](#)
- [Contrôle de l'accès aux ressources Kinesis Video Streams à l'aide de AWS IoT](#)
- [Validation de conformité pour Amazon Kinesis Video Streams](#)
- [Résilience dans Amazon Kinesis Video Streams](#)
- [Sécurité de l'infrastructure dans Kinesis Video Streams](#)
- [Bonnes pratiques de sécurité pour Kinesis Video Streams](#)

Protection des données dans Kinesis Video Streams

Vous pouvez utiliser le chiffrement côté serveur (SSE) à l'aide de clés AWS Key Management Service (AWS KMS) pour répondre à des exigences strictes en matière de gestion des données en chiffrant vos données au repos dans Amazon Kinesis Video Streams.

Rubriques

- [Qu'est-ce que le chiffrement côté serveur pour Kinesis Video Streams ?](#)
- [Considérations relatives aux coûts, aux régions et aux performances](#)
- [Comment démarrer avec le chiffrement côté serveur ?](#)
- [Création et utilisation d'une clé gérée par le client](#)
- [Autorisations d'utilisation d'une clé gérée par le client](#)

Qu'est-ce que le chiffrement côté serveur pour Kinesis Video Streams ?

Server-side Le chiffrement est une fonctionnalité de Kinesis Video Streams qui chiffre automatiquement les données avant qu'elles ne soient stockées au repos à l'aide AWS KMS d'une clé que vous spécifiez. Les données sont chiffrées avant leur écriture sur la couche de stockage du flux Kinesis Video Streams et déchiffrées après leur extraction de l'espace de stockage. Par conséquent, vos données sont toujours chiffrées au repos dans le service Kinesis Data Streams.

Grâce au chiffrement côté serveur, les producteurs et les consommateurs de vos flux vidéo Kinesis n'ont pas besoin de gérer les clés KMS ou les opérations cryptographiques. Si la conservation des données est activée, vos données sont automatiquement cryptées lorsqu'elles entrent dans Kinesis Video Streams et en sortent. Vos données inactives sont donc cryptées. AWS KMS fournit toutes les clés utilisées par la fonction de chiffrement côté serveur. AWS KMS rationalise l'utilisation d'une clé KMS pour Kinesis Video Streams gérée AWS par une clé AWS KMS spécifiée par l'utilisateur importée dans le service. AWS KMS

Considérations relatives aux coûts, aux régions et aux performances

Lorsque vous appliquez le chiffrement côté serveur, vous êtes soumis à l'utilisation de AWS KMS l'API et au coût des clés. Contrairement aux AWS KMS clés personnalisées, la clé `aws/kinesisvideo` KMS par défaut est proposée gratuitement. Toutefois, vous devez toujours payer les frais d'utilisation de l'API engagés par Kinesis Video Streams en votre nom.

Les coûts d'utilisation de l'API s'appliquent à chaque clé KMS, y compris les clés personnalisées. Les AWS KMS coûts varient en fonction du nombre d'informations d'identification utilisateur que vous utilisez pour vos producteurs et consommateurs de données, car chaque identifiant d'utilisateur nécessite un appel d' AWS KMS API unique.

Voici une description des coûts par ressource :

Clés

- La clé KMS pour Kinesis Video Streams gérée AWS par (alias aws/kinesisvideo =) est gratuite.
- User-generated Les clés KMS sont AWS KMS key payantes. Pour plus d'informations, consultez [AWS Key Management Service Tarification](#).

AWS KMS Utilisation de l'API

Les demandes d'API pour générer de nouvelles clés de chiffrement de données ou pour récupérer des clés de chiffrement existantes augmentent à mesure que le trafic augmente, et sont soumises à des coûts AWS KMS d'utilisation. Pour plus d'informations, voir [AWS Key Management Service Tarification : utilisation](#).

Kinesis Video Streams génère des demandes clés même lorsque la rétention est définie sur 0 (aucune rétention).

Disponibilité du chiffrement côté serveur par région

Server-side le chiffrement des flux vidéo Kinesis est disponible partout Régions AWS où Kinesis Video Streams est disponible.

Comment démarrer avec le chiffrement côté serveur ?

Server-side le chiffrement est toujours activé sur Kinesis Video Streams. Si aucune clé fournie par l'utilisateur n'est spécifiée lors de la création du flux, la Clé gérée par AWS (fournie par Kinesis Video Streams) est utilisée.

Une clé KMS fournie par l'utilisateur doit être attribuée à un flux vidéo Kinesis lors de sa création. Vous ne pourrez pas attribuer une clé différente à un flux à l'aide de l'[UpdateStream](#)API ultérieurement.

Vous pouvez attribuer une clé KMS fournie par l'utilisateur à un flux vidéo Kinesis de deux manières :

- Lorsque vous créez un flux vidéo Kinesis dans le AWS Management Console, spécifiez la clé KMS dans l'onglet Chiffrement de la page Créer un nouveau flux vidéo.
- Lorsque vous créez un flux vidéo Kinesis à l'aide de l'[CreateStream](#) API, spécifiez l'ID de clé dans le `KmsKeyId` paramètre.

Création et utilisation d'une clé gérée par le client

Cette section explique comment créer et utiliser vos propres clés KMS au lieu d'utiliser la clé administrée par Amazon Kinesis Video Streams.

Création d'une clé gérée par le client

Pour plus d'informations sur la création de vos propres clés, consultez la section [Création de clés](#) dans le guide du AWS Key Management Service développeur. Une fois que vous avez créé les clés de votre compte, le service Kinesis Video Streams les renvoie dans la liste des clés gérées par le client.

Utilisation d'une clé gérée par le client

Une fois que les autorisations appropriées ont été appliquées à vos consommateurs, producteurs et administrateurs, vous pouvez utiliser des clés KMS personnalisées, que ce Compte AWS soit vous-même ou une autre Compte AWS. Toutes les clés KMS de votre compte apparaissent dans la liste des clés gérées par le client sur la console.

Pour utiliser des clés KMS personnalisées situées dans un autre compte, vous devez être autorisé à utiliser ces clés. Vous devez également créer le flux à l'aide de l'API `CreateStream`. Vous ne pouvez pas utiliser les clés KMS de différents comptes dans les flux créés dans la console.

Note

La clé KMS n'est pas accessible tant que l'opération `PutMedia` ou `GetMedia` n'est pas effectuée. Ceci génère les résultats suivants :

- Si la clé que vous spécifiez n'existe pas, l'opération `CreateStream` réussit, mais les opérations `PutMedia` et `GetMedia` sur le flux échouent.
- Si vous utilisez la clé fournie (`aws/kinesisvideo`), elle n'est pas présente dans votre compte tant que la première opération `PutMedia` ou `GetMedia` n'est pas effectuée.

Autorisations d'utilisation d'une clé gérée par le client

Avant de pouvoir utiliser le chiffrement côté serveur avec une clé gérée par le client, vous devez configurer les politiques relatives aux clés KMS pour autoriser le chiffrement des flux ainsi que le chiffrement et le déchiffrement des enregistrements des flux. Pour des exemples et plus d'informations sur AWS KMS les autorisations, voir [Permissions AWS KMS d'API : référence sur les actions et les ressources](#).

Note

L'utilisation de la clé de service par défaut pour le chiffrement ne nécessite pas l'application d'autorisations IAM personnalisées.

Avant d'utiliser une clé gérée par le client, vérifiez que les producteurs et les consommateurs de vos flux vidéo Kinesis (principaux IAM) sont des utilisateurs conformément à la AWS KMS politique de clé par défaut. Si ce n'est pas le cas, les écritures et les lectures à partir d'un flux échoueront, ce qui pourrait entraîner la perte de données, des retards de traitement ou la suspension d'applications. Vous pouvez gérer les autorisations pour les clés KMS à l'aide de politiques IAM. Pour plus d'informations, consultez la section [Utilisation des politiques IAM avec AWS KMS](#).

Exemple d'autorisations pour les producteurs

Les producteurs de vos flux vidéo Kinesis doivent être autorisés à : `kms:GenerateDataKey`

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey"
      ],
      "Resource": "arn:aws:kms:us-west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    {
```

```
    "Effect": "Allow",
    "Action": [
        "kinesisvideo:PutMedia"
    ],
    "Resource": "arn:aws:kinesisvideo:*:123456789012:stream/MyStream/*"
}
]
```

Exemples d'autorisations accordées aux consommateurs

Les utilisateurs de votre flux vidéo Kinesis doivent être autorisés à : kms:Decrypt

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:us-west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:GetMedia"
      ],
      "Resource": "arn:aws:kinesisvideo:*:123456789012:stream/MyStream/*"
    }
  ]
}
```

Contrôle de l'accès aux ressources Kinesis Video Streams à l'aide d'IAM

Vous pouvez utiliser Gestion des identités et des accès AWS (IAM) avec Amazon Kinesis Video Streams pour contrôler si les utilisateurs de votre organisation peuvent effectuer une tâche à l'aide d'opérations spécifiques de l'API Kinesis Video Streams et s'ils peuvent utiliser des ressources spécifiques. AWS

Pour plus d'informations sur IAM, consultez les ressources suivantes :

- [Gestion des identités et des accès AWS \(JE SUIS\)](#)
- [Commencer à utiliser IAM](#)
- [Guide de l'utilisateur IAM](#)

Table des matières

- [Syntaxe d'une politique](#)
- [Actions relatives à Kinesis Video Streams](#)
- [Amazon Resource Names \(ARN\) pour Kinesis Video Streams](#)
- [Autoriser d'autres comptes IAM à accéder à un flux vidéo Kinesis](#)
- [Exemples de politiques pour Kinesis Video Streams](#)

Syntaxe d'une politique

Une politique IAM est un document JSON qui se compose d'une ou de plusieurs déclarations. Chaque déclaration est structurée comme suit :

```
{
  "Statement": [{
    "Effect": "effect",
    "Action": "action",
    "Resource": "arn",
    "Condition": {
      "condition": {
        "key": "value"
      }
    }
  ]
}
```

```
]
}
```

Une déclaration se compose de différents éléments :

- **Effet** — L'effet peut être `Allow` ou `Deny`. Comme, par défaut, les utilisateurs n'ont pas la permission d'utiliser les ressources et les actions d'API, toutes les demandes sont refusées. Une autorisation explicite remplace l'autorisation par défaut. Un refus explicite remplace toute autorisation.
- **Action** : il s'agit de l'action d'API spécifique pour laquelle vous accordez ou refusez l'autorisation.
- **Ressource** : ressource affectée par l'action. Pour spécifier une ressource dans la déclaration, vous devez utiliser son Amazon Resource Name (ARN).
- **État** — Les conditions sont facultatives. Elles permettent de contrôler à quel moment votre politique est effective.

Lorsque vous créez et gérez des politiques IAM, nous vous recommandons d'utiliser le [générateur de politiques IAM et le simulateur de politiques IAM](#).

Actions relatives à Kinesis Video Streams

Dans une déclaration de politique IAM, vous pouvez spécifier une action d'API à partir de n'importe quel service prenant en charge IAM. Pour Kinesis Video Streams, utilisez le préfixe suivant avec le nom de l'action d'API : `kinesisvideo:`. Par exemple : `kinesisvideo:CreateStream`, `kinesisvideo:ListStreams` et `kinesisvideo:DescribeStream`.

Pour spécifier plusieurs actions dans une seule déclaration, séparez-les par des virgules comme suit :

```
"Action": ["kinesisvideo:action1", "kinesisvideo:action2"]
```

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques. Par exemple, vous pouvez spécifier toutes les actions dont le nom commence par le mot « `Get` » comme suit :

```
"Action": "kinesisvideo:Get*"
```

Pour spécifier toutes les opérations Kinesis Video Streams, utilisez le caractère générique astérisque (*) comme suit :

```
"Action": "kinesisvideo:*"
```

Pour obtenir la liste complète des actions de l'API Kinesis Video Streams, consultez [le manuel de référence de l'API Kinesis Video Streams](#).

Amazon Resource Names (ARN) pour Kinesis Video Streams

Chaque déclaration de politique IAM s'applique aux ressources que vous spécifiez à l'aide de leur ARN.

Utilisez le format de ressource ARN suivant pour Kinesis Video Streams :

```
arn:aws:kinesisvideo:region:account-id:stream/stream-name/code
```

Par exemple :

```
"Resource": arn:aws:kinesisvideo:*:111122223333:stream/my-stream/0123456789012
```

Vous pouvez obtenir l'ARN d'un flux en utilisant [DescribeStream](#).

Autoriser d'autres comptes IAM à accéder à un flux vidéo Kinesis

Vous devrez peut-être autoriser d'autres comptes IAM à effectuer des opérations sur des flux dans Kinesis Video Streams. La présentation suivante décrit les étapes générales pour accorder l'accès à des flux vidéo parmi les comptes :

1. Obtenez l'identifiant de compte à 12 chiffres du compte auquel vous souhaitez accorder l'autorisation d'effectuer des opérations sur la ressource de streaming créée dans votre compte.

Exemple : Dans les étapes suivantes, nous utiliserons 111111111111 comme identifiant de compte pour le compte auquel vous souhaitez accorder l'autorisation, et 999999999999 comme identifiant pour vos Kinesis Video Streams

2. Créez une politique gérée par IAM dans le compte propriétaire du flux (999999999999) qui autorise le niveau d'accès que vous souhaitez accorder.

Exemple de politique :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:GetDataEndpoint",
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia"
      ],
      "Resource": "arn:aws:kinesisvideo:us-west-2:999999999999:stream/custom-stream-name/1613732218179"
    }
  ]
}
```

Pour d'autres exemples de politiques relatives aux ressources Kinesis Video Streams, [Exemples de stratégies](#) reportez-vous à la section suivante.

3. Créez un rôle dans le compte propriétaire du flux (999999999999) et spécifiez le compte pour lequel vous souhaitez accorder des autorisations (111111111111). Cela ajoutera une entité de confiance au rôle.

Exemple de politique fiable :


JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111111111111:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
]
}
```

Associez à ce rôle la politique que vous avez créée à l'étape précédente.

Vous avez maintenant créé un rôle dans le compte 999999999999 qui est autorisé à effectuer des opérations telles que `DescribeStreamGetDataEndpoint`, et sur PutMedia un ARN de ressource de flux dans la politique gérée. Ce nouveau rôle fait également confiance à l'autre compte, 111111111111, pour assumer ce rôle.

 Important

Notez l'ARN du rôle, vous en aurez besoin à l'étape suivante.

4. Créez une politique gérée dans l'autre compte, 111111111111, qui autorise l'`AssumeRoleAction` sur le rôle que vous avez créé dans le compte 999999999999 à l'étape précédente. Vous devrez mentionner l'ARN du rôle indiqué à l'étape précédente.

Exemple de politique :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::999999999999:role/CustomRoleName"
    }
  ]
}
```

5. Associez la politique créée à l'étape précédente à une entité IAM, telle qu'un rôle ou un utilisateur dans le compte 111111111111. Cet utilisateur est désormais autorisé à assumer le rôle `CustomRoleName` dans le compte 999999999999.

Les informations d'identification de cet utilisateur appellent l' `AWS STS AssumeRoleAPI` pour obtenir les informations d'identification de session, qui sont ensuite utilisées pour appeler les API Kinesis Video Streams sur le stream créé dans le compte 999999999999.

```
aws sts assume-role --role-arn "arn:aws:iam::999999999999:role/CustomRoleName" --
role-session-name "kvs-cross-account-assume-role"
{
  "Credentials": {
    "AccessKeyId": "",
    "SecretAccessKey": "",
    "SessionToken": "",
    "Expiration": ""
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "",
    "Arn": ""
  }
}
```

6. Définissez la clé d'accès, la clé secrète et les informations d'identification de session en fonction de la configuration précédente dans l'environnement.

```
set AWS_ACCESS_KEY_ID=
set AWS_SECRET_ACCESS_KEY=
set AWS_SESSION_TOKEN=
```

7. Exécutez les API Kinesis Video Streams pour décrire et obtenir le point de terminaison de données du flux dans le compte 999999999999.

```
aws kinesismedia describe-stream --stream-arn "arn:aws:kinesisvideo:us-
west-2:999999999999:stream/custom-stream-name/1613732218179"
{
  "StreamInfo": {
    "StreamName": "custom-stream-name",
    "StreamARN": "arn:aws:kinesisvideo:us-west-2:999999999999:stream/custom-
stream-name/1613732218179",
    "KmsKeyId": "arn:aws:kms:us-west-2:999999999999:alias/aws/kinesisvideo",
    "Version": "abcd",
    "Status": "ACTIVE",
    "CreationTime": "2018-02-19T10:56:58.179000+00:00",
    "DataRetentionInHours": 24
  }
}

aws kinesismedia get-data-endpoint --stream-arn "arn:aws:kinesisvideo:us-
west-2:999999999999:stream/custom-stream-name/1613732218179" --api-name "PUT_MEDIA"
```

```
{
  "DataEndpoint": "https://s-b12345.kinesisvideo.us-west-2.amazonaws.com"
}
```

Pour obtenir des instructions générales étape par étape sur l'octroi d'un accès entre comptes, consultez la section [Accès délégué à Comptes AWS l'aide de rôles IAM](#).

Exemples de politiques pour Kinesis Video Streams

Les exemples de règles suivants montrent comment contrôler l'accès des utilisateurs à vos Kinesis Video Streams.

Exemple 1 : Autoriser les utilisateurs à obtenir des données depuis n'importe quel flux vidéo Kinesis

Cette politique permet à un utilisateur ou à un groupe d'effectuer les `ListTagsForStream` opérations `DescribeStreamGetDataEndpoint`, `GetMedia`, `ListStreams`, et sur n'importe quel flux vidéo Kinesis. Cette stratégie est adaptée aux utilisateurs qui peuvent obtenir des données depuis n'importe quel flux vidéo.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:Describe*",
        "kinesisvideo:Get*",
        "kinesisvideo:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

Exemple 2 : Autoriser un utilisateur à créer un flux vidéo Kinesis et à y écrire des données

Cette stratégie permet à un utilisateur ou groupe d'effectuer les opérations `CreateStream` et `PutMedia`. Cette stratégie est adaptée à une caméra de sécurité pouvant créer un flux vidéo et y envoyer des données.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:CreateStream",
        "kinesisvideo:PutMedia"
      ],
      "Resource": "*"
    }
  ]
}
```

Exemple 3 : Autoriser un utilisateur à accéder pleinement à toutes les ressources Kinesis Video Streams

Cette politique permet à un utilisateur ou à un groupe d'effectuer n'importe quelle opération Kinesis Video Streams sur n'importe quelle ressource. Cette stratégie est adaptée aux administrateurs.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesisvideo:*",
      "Resource": "*"
    }
  ]
}
```

Exemple 4 : Autoriser un utilisateur à écrire des données dans un flux vidéo Kinesis spécifique

Cette stratégie permet à un utilisateur ou groupe d'écrire des données sur un flux vidéo spécifique. Cette stratégie est adaptée à un appareil pouvant envoyer des données vers un même flux.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesisvideo:PutMedia",
      "Resource": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/your_stream/0123456789012"
    }
  ]
}
```

Contrôle de l'accès aux ressources Kinesis Video Streams à l'aide de AWS IoT

Cette section explique comment permettre à un appareil (par exemple, une caméra) d'envoyer des données audio et vidéo à un seul flux vidéo Kinesis en particulier. Vous pouvez le faire en utilisant le fournisseur AWS IoT d'informations d'identification et un rôle Gestion des identités et des accès AWS (IAM).

Les appareils peuvent utiliser X.509 des certificats pour se connecter à AWS IoT l'aide des protocoles d'authentification mutuelle TLS. D'autres Services AWS (par exemple, Kinesis Video Streams) ne prennent pas en charge l'authentification basée sur des certificats, mais peuvent être appelés à l'AWS aide d'informations d'identification AWS au format Signature Version 4. L'algorithme Signature Version 4 exige généralement que l'appelant dispose d'un identifiant de clé d'accès et d'une clé d'accès secrète. AWS IoT dispose d'un fournisseur d'informations d'identification qui vous permet d'utiliser le X.509 certificat intégré comme identité unique de l'appareil pour authentifier les AWS demandes (par exemple, les demandes adressées à Kinesis Video Streams). Il n'est donc plus nécessaire de stocker un identifiant de clé d'accès et une clé d'accès secrète sur votre appareil.

Le fournisseur d'informations d'identification authentifie un client (dans ce cas, un SDK Kinesis Video Streams exécuté sur la caméra à laquelle vous souhaitez envoyer des données vers un flux vidéo) à l'aide d'un certificat et émet X.509 un jeton de sécurité temporaire à privilèges limités. Vous pouvez utiliser le jeton pour signer et authentifier n'importe quelle AWS demande (dans ce cas, un appel aux Kinesis Video Streams). Pour plus d'informations, consultez la section [Autorisation des appels directs vers les AWS services](#).

Pour authentifier les demandes de votre caméra adressées à Kinesis Video Streams, vous devez créer et configurer un rôle IAM et associer les politiques IAM appropriées au rôle afin que le fournisseur d'informations d'identification puisse assumer AWS IoT le rôle en votre nom.

Pour plus d'informations AWS IoT, consultez [AWS IoT Core la documentation](#). Pour plus d'informations sur (IAM), consultez [Gestion des identités et des accès AWS \(IAM\)](#).

Rubriques

- [AWS IoT ThingName comme nom de flux](#)
- [AWS IoT Certificateld comme nom de flux](#)
- [Utilisation AWS IoT informations d'identification pour diffuser vers un nom de flux codé en dur](#)

AWS IoT ThingName comme nom de flux

Rubriques

- [Étape 1 : Création d'un AWS IoT type d'objet et un AWS IoT thing](#)
- [Étape 2 : Création d'un rôle IAM à assumer par AWS IoT](#)
- [Étape 3 : créer et configurer le X.509 certificat](#)
- [Étape 4 : Testez le AWS IoT informations d'identification associées à votre flux vidéo Kinesis](#)
- [Étape 5 : Déploiement AWS IoT certificats et informations d'identification sur le système de fichiers de votre caméra et diffusion de données vers votre flux vidéo](#)

Étape 1 : Création d'un AWS IoT type d'objet et un AWS IoT thing

Dans AWS IoT, un objet est la représentation d'un périphérique ou d'une entité logique spécifique. Dans ce cas, un AWS IoT objet représente le flux vidéo Kinesis pour lequel vous souhaitez configurer le contrôle d'accès au niveau des ressources. Pour créer un objet, vous devez d'abord créer un type d' AWS IoT objet. Vous pouvez utiliser des types d' AWS IoT objet pour stocker des informations de description et de configuration communes à tous les éléments associés au même type d'objet.

1. L'exemple de commande suivant crée un type d'objet `kvs_example_camera` :

```
aws --profile default iot create-thing-type --thing-type-name kvs_example_camera >
iot-thing-type.json
```

2. Cet exemple de commande crée l'`kvs_example_camera_stream` objet du type d'`kvs_example_camera` objet :

```
aws --profile default iot create-thing --thing-name kvs_example_camera_stream --
thing-type-name kvs_example_camera > iot-thing.json
```

Étape 2 : Création d'un rôle IAM à assumer par AWS IoT

Les rôles IAM sont similaires aux utilisateurs, dans la mesure où un rôle est une AWS identité dotée de politiques d'autorisation qui déterminent ce que l'identité peut et ne peut pas faire. AWS Un rôle peut être assumé par toute personne qui en a besoin. Lorsque vous assumez un rôle, il vous fournit des informations d'identification de sécurité temporaires pour votre session de rôle.

Le rôle que vous créez au cours de cette étape peut être assumé AWS IoT pour obtenir des informations d'identification temporaires auprès du service de jetons de sécurité (STS) lors de l'exécution de demandes d'autorisation d'informations d'identification auprès du client. Dans ce cas, le client est le SDK Kinesis Video Streams qui s'exécute sur votre caméra.

Effectuez les étapes suivantes pour créer et configurer ce rôle IAM :

1. Créez un rôle IAM.

L'exemple de commande suivant crée un rôle IAM `KVSCameraCertificateBasedIAMRole` :

```
aws --profile default iam create-role --role-name KVSCameraCertificateBasedIAMRole
--assume-role-policy-document 'file://iam-policy-document.json' > iam-role.json
```

Vous pouvez utiliser le fichier JSON de stratégie d'approbation suivant pour `iam-policy-document.json` :

JSON

```
{
  "Version": "2012-10-17",
```

```

    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "credentials.iot.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }
}

```

2. Ensuite, associez une politique d'autorisation au rôle IAM que vous avez créé précédemment. Cette politique d'autorisation permet un contrôle d'accès sélectif (un sous-ensemble des opérations prises en charge) pour une AWS ressource. Dans ce cas, la AWS ressource est le flux vidéo auquel vous souhaitez que votre caméra envoie des données. En d'autres termes, une fois toutes les étapes de configuration terminées, cette caméra sera en mesure d'envoyer des données uniquement à ce flux vidéo.

```

aws --profile default iam put-role-policy --role-name
KVSCameraCertificateBasedIAMRole --policy-name KVSCameraIAMPolicy --policy-
document 'file://iam-permission-document.json'

```

Vous pouvez utiliser le JSON de politique IAM suivant pour : iam-permission-document.json

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": "arn:aws:kinesisvideo:*:*:stream/${credentials-
iot:ThingName}/*"
    }
  ]
}

```

```
    ]
  }
```

Notez que cette règle autorise les actions spécifiées uniquement sur un flux vidéo (AWS ressource) spécifié par l'espace réservé(`${credentials-iot:ThingName}`). Cet espace réservé prend la valeur de l'attribut AWS IoT thing ThingName lorsque le fournisseur AWS IoT d'informations d'identification envoie le nom du flux vidéo dans la demande.

3. Créez ensuite un alias de rôle pour votre rôle IAM. L'alias de rôle est un autre modèle de données qui pointe vers le rôle IAM. Une demande de fournisseur AWS IoT d'informations d'identification doit inclure un alias de rôle indiquant le rôle IAM à assumer pour obtenir les informations d'identification temporaires auprès du STS.

L'exemple de commande suivant crée un alias de rôle appelé `KvsCameraIoTRoleAlias`,

```
aws --profile default iot create-role-alias --role-alias KvsCameraIoTRoleAlias --
role-arn $(jq --raw-output '.Role.Arn' iam-role.json) --credential-duration-seconds
3600 > iot-role-alias.json
```

4. Vous pouvez maintenant créer la politique qui permettra AWS IoT de jouer un rôle avec le certificat (une fois qu'il est attaché) en utilisant l'alias de rôle.

L'exemple de commande suivant crée une politique pour AWS IoT called `KvsCameraIoTPolicy`.

```
aws --profile default iot create-policy --policy-name KvsCameraIoTPolicy --policy-
document 'file://iot-policy-document.json'
```

Vous pouvez utiliser la commande suivante pour créer le `iot-policy-document.json` document JSON :

```
cat > iot-policy-document.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:AssumeRoleWithCertificate"
      ],
    }
  ],
}
```

```

        "Resource": "$(jq --raw-output '.roleAliasArn' iot-role-alias.json)"
    }
]
}
EOF

```

Étape 3 : créer et configurer le X.509 certificat

La communication entre un appareil (votre flux vidéo) AWS IoT est protégée par l'utilisation de X.509 certificats.

1. Créez le certificat auquel vous devez joindre la politique AWS IoT que vous avez créée précédemment.

```

aws --profile default iot create-keys-and-certificate --set-as-active --
certificate-pem-outfile certificate.pem --public-key-outfile public.pem.key --
private-key-outfile private.pem.key > certificate

```

2. Joignez la politique pour AWS IoT (KvsCameraIoTPolicy créée précédemment) à ce certificat.

```

aws --profile default iot attach-policy --policy-name KvsCameraIoTPolicy --target
$(jq --raw-output '.certificateArn' certificate)

```

3. Joignez votre AWS IoT objet (kvs_example_camera_stream) au certificat que vous venez de créer :

```

aws --profile default iot attach-thing-principal --thing-name
kvs_example_camera_stream --principal $(jq --raw-output '.certificateArn'
certificate)

```

4. Pour autoriser les demandes via le fournisseur AWS IoT d'informations d'identification, vous avez besoin du point de terminaison des AWS IoT informations d'identification, qui est unique à votre Compte AWS identifiant. Vous pouvez utiliser la commande suivante pour obtenir le point de terminaison des AWS IoT informations d'identification.

```

aws --profile default iot describe-endpoint --endpoint-type iot:CredentialProvider
--output text > iot-credential-provider.txt

```

5. Outre le X.509 certificat créé précédemment, vous devez également disposer d'un certificat CA pour établir la confiance avec le service principal via TLS. Vous pouvez obtenir le certificat CA à l'aide de la commande suivante :

```
curl --silent 'https://www.amazontrust.com/repository/SFSRootCAG2.pem' --output cacert.pem
```

Étape 4 : Testez le AWS IoT informations d'identification associées à votre flux vidéo Kinesis

Vous pouvez maintenant tester les AWS IoT informations d'identification que vous avez configurées jusqu'à présent.

1. Tout d'abord, créez un flux vidéo Kinesis avec lequel vous souhaitez tester cette configuration.

Important

Créez un flux vidéo avec un nom identique au nom de l' AWS IoT objet que vous avez créé à l'étape précédente (`kvs_example_camera_stream`).

```
aws kinesismedia create-stream --data-retention-in-hours 24 --stream-name kvs_example_camera_stream
```

2. Appelez ensuite le fournisseur AWS IoT d'informations d'identification pour obtenir les informations d'identification temporaires :

```
curl --silent -H "x-amzn-iot-thingname:kvs_example_camera_stream" --cert certificate.pem --key private.pem.key https://IOT_GET_CREDENTIAL_ENDPOINT/role-aliases/KvsCameraIoTRoleAlias/credentials --cacert ./cacert.pem > token.json
```

Note

Vous pouvez utiliser la commande suivante pour obtenir `IOT_GET_CREDENTIAL_ENDPOINT` :

```
IOT_GET_CREDENTIAL_ENDPOINT=`cat iot-credential-provider.txt`
```

Le JSON de sortie contient le `accessKeysecretKey`, et le `sessionToken`, que vous pouvez utiliser pour accéder aux Kinesis Video Streams.

3. Pour votre test, vous pouvez utiliser ces informations d'identification pour appeler l'API Kinesis Video `DescribeStream` Streams pour l'`kvs_example_camera_stream` exemple de flux vidéo.

```
AWS_ACCESS_KEY_ID=$(jq --raw-output '.credentials.accessKeyId' token.json)
AWS_SECRET_ACCESS_KEY=$(jq --raw-output '.credentials.secretAccessKey' token.json)
AWS_SESSION_TOKEN=$(jq --raw-output '.credentials.sessionToken' token.json) aws
kinesisvideo describe-stream --stream-name kvs_example_camera_stream
```

Étape 5 : Déploiement AWS IoT certificats et informations d'identification sur le système de fichiers de votre caméra et diffusion de données vers votre flux vidéo

Note

Les étapes décrites dans cette section décrivent l'envoi de contenu multimédia vers un flux vidéo Kinesis à partir d'une caméra utilisant le [the section called "C++"](#)

1. Copiez le X.509 certificat, la clé privée et le certificat CA générés lors des étapes précédentes dans le système de fichiers de votre caméra. Spécifiez les chemins où ces fichiers sont stockés, le nom de l'alias du rôle et le point de terminaison des AWS IoT informations d'identification pour exécuter la `gst-launch-1.0` commande ou votre exemple d'application.
2. L'exemple de commande suivant utilise l'autorisation de AWS IoT certificat pour envoyer une vidéo à Kinesis Video Streams :

```
gst-launch-1.0 rtspsrc location=rtsp://YourCameraRtspUrl short-header=TRUE !
rtph264depay ! video/x-h264,format=avc,alignment=au ! h264parse ! kvssink stream-
name="kvs_example_camera_stream" aws-region="YourAWSRegion" iot-certificate="iot-
certificate,endpoint=credential-account-specific-prefix.credentials.iot.aws-
```

```
region.amazonaws.com,cert-path=/path/to/certificate.pem,key-path=/path/to/private.pem,key,ca-path=/path/to/cacert.pem,role-aliases=KvsCameraIoTRoleAlias"
```

AWS IoT CertificateId comme nom de flux

Pour représenter votre appareil (par exemple, votre caméra) par le biais d'un AWS IoT objet, tout en autorisant un autre nom de flux, vous pouvez utiliser l' AWS IoT `certificateId` attribut comme nom de votre flux et fournir des autorisations Kinesis Video Streams sur le flux AWS IoT en utilisant. Les étapes pour y parvenir sont similaires à celles décrites précédemment, avec quelques modifications.

- Modifiez la politique d'autorisations pour votre rôle IAM (`iam-permission-document.json`) comme suit :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": "arn:aws:kinesisvideo:*:*:stream/${credentials-iot:AwsCertificateId}/*"
    }
  ]
}
```

Note

L'ARN de ressource utilise l'ID de certificat comme espace réservé pour le nom de flux. L'autorisation IAM fonctionne lorsque vous utilisez l'ID du certificat comme nom de flux. Obtenez l'ID du certificat à partir du certificat afin de pouvoir l'utiliser comme nom de flux dans l'appel d'API de description du flux suivant.

```
export CERTIFICATE_ID=`cat certificate | jq --raw-output '.certificateId'`
```

- Vérifiez cette modification à l'aide de la commande CLI Kinesis Video Streams `describe-stream` :

```
AWS_ACCESS_KEY_ID=$(jq --raw-output '.credentials.accessKeyId' token.json)
AWS_SECRET_ACCESS_KEY=$(jq --raw-output '.credentials.secretAccessKey' token.json)
AWS_SESSION_TOKEN=$(jq --raw-output '.credentials.sessionToken' token.json) aws
kinesisvideo describe-stream --stream-name ${CERTIFICATE_ID}
```

- Transmettez le `CertificateId` au fournisseur d'informations d'identification dans AWS IoT l'[exemple d'application du SDK](#) Kinesis Video Streams C++ :

```
credential_provider =
    make_unique<IotCertCredentialProvider>(iot_get_credential_endpoint,
        cert_path,
        private_key_path,
        role_alias,
        ca_cert_path,
        certificateId);
```

Note

Notez que vous transmettez le `thingname` au fournisseur AWS IoT d'informations d'identification. Vous pouvez l'utiliser `getenv` pour transmettre le nom de l'objet à l'application de démonstration de la même manière que pour les autres AWS IoT attributs. Utilisez l'ID de certificat comme nom de flux dans les paramètres de ligne de commande lorsque vous exécutez l'exemple d'application.

Utilisation AWS IoT informations d'identification pour diffuser vers un nom de flux codé en dur

Pour représenter votre appareil (par exemple, votre caméra) à travers un AWS IoT objet, mais autoriser le streaming vers un flux vidéo Amazon Kinesis spécifique, accordez des autorisations Amazon Kinesis Video Streams sur le flux à l'aide de. AWS IoT Le processus est similaire aux sections précédentes, avec quelques modifications.

Modifiez la politique d'autorisations pour votre rôle IAM (`iam-permission-document.json`) comme suit :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": "arn:aws:kinesisvideo:*:*:stream/YourStreamName/*"
    }
  ]
}
```

Copiez le X.509 certificat, la clé privée et le certificat CA générés lors des étapes précédentes dans le système de fichiers de votre caméra.

Spécifiez les chemins où ces fichiers sont stockés, le nom de l'alias du rôle, le nom de l'AWS IoT objet et le point de terminaison des AWS IoT informations d'identification pour exécuter la `gst-launch-1.0` commande ou votre exemple d'application.

L'exemple de commande suivant utilise l'autorisation de AWS IoT certificat pour envoyer une vidéo à Amazon Kinesis Video Streams :

```
gst-launch-1.0 rtspsrc location=rtsp://YourCameraRtspUrl short-header=TRUE !
rtph264depay ! video/x-h264,format=avc,alignment=au ! h264parse ! kvssink
stream-name="YourStreamName" aws-region="YourAWSRegion" iot-certificate="iot-
certificate,endpoint=credential-account-specific-prefix.credentials.iot.aws-
region.amazonaws.com,cert-path=/path/to/certificate.pem,key-path=/path/to/
private.pem.key,ca-path=/path/to/cacert.pem,role-aliases=KvsCameraIoTRoleAlias,iot-
thing-name=YourThingName"
```

Validation de conformité pour Amazon Kinesis Video Streams

Pour savoir si un [programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité](#) spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de [AWS conformité Programmes AWS](#) de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir [Téléchargement de rapports dans AWS Artifact](#) .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez [AWS la documentation de sécurité](#).

Résilience dans Amazon Kinesis Video Streams

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section [Infrastructure AWS mondiale](#).

Sécurité de l'infrastructure dans Kinesis Video Streams

En tant que service géré, Amazon Kinesis Video Streams est protégé par AWS les procédures de sécurité du réseau mondial décrites dans [le livre blanc Amazon Web Services : présentation des processus de sécurité](#).

Vous utilisez des appels d'API AWS publiés pour accéder à Kinesis Video Streams via le réseau. Les clients doivent prendre en charge le protocole TLS (Transport Layer Security) 1.2 ou version ultérieure. Les clients doivent également prendre en charge les suites de chiffrement à parfaite

confidentialité (PFS), telles que Ephemeral (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Diffie-Hellman La plupart des systèmes modernes telles que Java 7 et versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un identifiant de clé d'accès et d'une clé d'accès secrète associés à un principal IAM. Vous pouvez également utiliser [AWS Security Token Service](#) (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Bonnes pratiques de sécurité pour Kinesis Video Streams

Amazon Kinesis Video Streams propose un certain nombre de fonctionnalités de sécurité à prendre en compte lors de l'élaboration et de la mise en œuvre de vos propres politiques de sécurité. Les bonnes pratiques suivantes doivent être considérées comme des instructions générales et ne représentent pas une solution de sécurité complète. Étant donné que ces bonnes pratiques peuvent ne pas être appropriées ou suffisantes pour votre environnement, considérez-les comme des remarques utiles plutôt que comme des recommandations.

Pour découvrir les bonnes pratiques de sécurité applicables à vos appareils à distance, consultez [Bonnes pratiques de sécurité pour les agents d'appareil](#).

Implémentation d'un accès sur la base du moindre privilège

Lorsque vous accordez des autorisations, vous décidez qui obtient quelles autorisations pour quelles ressources Kinesis Video Streams. Vous activez des actions spécifiques que vous souhaitez autoriser sur ces ressources. Par conséquent, vous devez accorder uniquement les autorisations qui sont requises pour exécuter une tâche. L'implémentation d'un accès sur la base du moindre privilège est fondamentale pour réduire les risques en matière de sécurité et l'impact que pourraient avoir des d'erreurs ou des actes de malveillance.

Par exemple, un producteur qui envoie des données à Kinesis Video Streams a PutMedia uniquement besoin de `GetStreamingEndpoint`, `DescribeStream` et. N'octroyez pas aux applications de type producteur des autorisations pour toutes les actions (*), ou pour d'autres actions du type `GetMedia`.

Pour plus d'informations, consultez la section [En quoi consiste le principe de moindre privilège et pourquoi est-il nécessaire ?](#)

Utilisation des rôles IAM

Les applications productrices et clientes doivent disposer d'informations d'identification valides pour accéder à Kinesis Video Streams. Vous ne devez pas stocker les informations d'identification AWS directement dans une application client ni dans un compartiment Amazon S3. Il s'agit d'informations d'identification à long terme qui ne font pas l'objet d'une rotation automatique et qui pourraient avoir un impact commercial significatif si elles sont compromises.

Vous devez plutôt utiliser un rôle IAM pour gérer les informations d'identification temporaires permettant à vos applications productrices et clientes d'accéder à Kinesis Video Streams. Lorsque vous utilisez un rôle, vous n'avez pas besoin d'utiliser des informations d'identification à long terme (telles qu'un nom d'utilisateur et un mot de passe ou des clés d'accès) pour accéder à d'autres ressources.

Pour plus d'informations, consultez les rubriques suivantes dans le Guide de l'utilisateur IAM :

- [Rôles IAM](#)
- [Scénarios courants pour les rôles : utilisateurs, applications et services.](#)

CloudTrail À utiliser pour surveiller les appels d'API

Kinesis Video Streams fonctionne AWS CloudTrail avec un service qui fournit un enregistrement des actions effectuées par un utilisateur, un rôle ou Service AWS un utilisateur dans Kinesis Video Streams.

Vous pouvez utiliser les informations collectées CloudTrail pour déterminer la demande envoyée à Kinesis Video Streams, l'adresse IP à partir de laquelle la demande a été effectuée, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires.

Pour de plus amples informations, veuillez consulter [the section called “Enregistrez les appels d'API avec CloudTrail”](#).

Résolution des problèmes liés à Kinesis Video Streams

Utilisez les informations suivantes pour résoudre les problèmes courants rencontrés avec Amazon Kinesis Video Streams.

Rubriques

- [Problèmes généraux](#)
- [Problèmes liés à l'API](#)
- [Problèmes liés au HLS](#)
- [Problèmes liés à Java](#)
- [Problèmes liés à la bibliothèque des producteurs](#)
- [Problèmes liés à la bibliothèque d'analyseurs de flux](#)
- [Problèmes liés au réseau](#)

Problèmes généraux

Cette section décrit les problèmes généraux que vous pouvez rencontrer lors de l'utilisation de Kinesis Video Streams.

Problèmes

- [Latence trop élevée](#)

Latence trop élevée

La latence peut être due à la durée des fragments envoyés au service Kinesis Video Streams. Une solution pour réduire la latence entre le producteur et le service consiste à configurer le pipeline média pour produire des durées de fragment plus courtes.

Pour réduire le nombre d'images envoyées dans chaque fragment, réduisez la valeur suivante dans `kinesis_video_gstreamer_sample_app.cpp` :

```
g_object_set(G_OBJECT (data.encoder), "bframes", 0, "key-int-max", 45, "bitrate", 512, NULL);
```

Note

Les latences sont plus élevées dans le navigateur Mozilla Firefox en raison de l'implémentation interne du rendu vidéo.

Problèmes liés à l'API

Cette section décrit les problèmes d'API que vous pouvez rencontrer lorsque vous travaillez avec Kinesis Video Streams.

Problèmes

- [Erreur : « Options inconnues »](#)
- [Erreur : « Impossible de déterminer service/operation le nom à autoriser »](#)
- [Erreur : « Failed to put a frame in the stream » \(Échec du placement d'une trame dans le flux\)](#)
- [Erreur : « Le service AckEvent a fermé la connexion avant la réception de la version finale »](#)
- [Erreur : « STATUS_STORE_OUT_OF_MEMORY »](#)
- [Erreur : « Les informations d'identification doivent être limitées à une région valide. »](#)

Erreur : « Options inconnues »

`GetMedia` et `GetMediaForFragmentList` peuvent échouer avec l'erreur suivante :

```
Unknown options: <filename>.mkv
```

Cette erreur se produit si vous avez configuré le AWS CLI avec un output type de json. Reconfigurez le AWS CLI avec le type de sortie par défaut (none). Pour plus d'informations sur la configuration du AWS CLI, voir [configure](#) dans le manuel de référence des AWS CLI commandes.

Erreur : « Impossible de déterminer service/operation le nom à autoriser »

`GetMedia` peut échouer avec l'erreur suivante :

```
Unable to determine service/operation name to be authorized
```

Cette erreur peut se produire si le point de terminaison n'est pas correctement spécifié. Lorsque vous obtenez le point de terminaison, veillez à inclure le paramètre suivant dans l'`GetDataEndpoint` appel, en fonction de l'API à appeler :

```
--api-name GET_MEDIA
--api-name PUT_MEDIA
--api-name GET_MEDIA_FOR_FRAGMENT_LIST
--api-name LIST_FRAGMENTS
```

Erreur : « Failed to put a frame in the stream » (Échec du placement d'une trame dans le flux)

`PutMedia` peut échouer avec l'erreur suivante :

```
Failed to put a frame in the stream
```

Cette erreur peut se produire si la connectivité ou les autorisations ne sont pas disponibles pour le service. Exécutez ce qui suit dans le AWS CLI, et vérifiez que les informations du flux peuvent être récupérées :

```
aws kinesismedia describe-stream --stream-name StreamName --endpoint https://
ServiceEndpoint.kinesisvideo.region.amazonaws.com
```

Si l'appel échoue, voir [Résolution AWS CLI des erreurs](#) pour plus d'informations.

Erreur : « Le service AckEvent a fermé la connexion avant la réception de la version finale »

`PutMedia` peut échouer avec l'erreur suivante :

```
com.amazonaws.SdkClientException: Service closed connection before final AckEvent was
received
```

Cette erreur peut se produire si `PushbackInputStream` n'est pas correctement implémenté. Vérifiez que les `unread()` méthodes sont correctement mises en œuvre.

Erreur : « STATUS_STORE_OUT_OF_MEMORY »

`PutMedia` peut échouer avec l'erreur suivante :

```
The content store is out of memory.
```

Cette erreur se produit lorsque le magasin de contenu n'est pas alloué avec une taille suffisante. Pour augmenter la taille du magasin de contenu, augmentez la valeur de `StorageInfo.storageSize`. Pour de plus amples informations, veuillez consulter [StorageInfo](#).

Erreur : « Les informations d'identification doivent être limitées à une région valide. »

Cette erreur se produit si la région de signature ne correspond pas à la région du point de terminaison.

Par exemple, si vous spécifiez `us-west-2` comme région de signature, mais que vous essayez de vous connecter au point de terminaison `kinesisvideo.us-east-1.amazonaws.com` (`us-east-1`), vous recevrez cette erreur.

Dans certaines applications, comme [kvssink](#), la chaîne de repli des régions est définie par défaut sur `us-west-2`. Vérifiez que vous avez correctement défini votre région en fonction de l'application que vous utilisez.

Problèmes liés au HLS

Si la lecture de votre flux vidéo n'est pas correcte, consultez [the section called “Résolution des problèmes liés au HLS”](#).

Problèmes liés à Java

Cette section explique comment résoudre les problèmes Java courants rencontrés lors de l'utilisation de Kinesis Video Streams.

Problèmes

- [Activation des journaux Java](#)

Activation des journaux Java

Pour résoudre les problèmes liés aux exemples et aux bibliothèques Java, il est utile d'activer et d'examiner les journaux de débogage. Pour activer les journaux de débogage, procédez comme suit :

1. Ajoutez log4j au fichier pom.xml dans le nœud dependencies :

```
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
```

2. Dans le répertoire target/classes, créez un fichier nommé log4j.properties avec les contenus suivants :

```
# Root logger option
log4j.rootLogger=DEBUG, stdout

# Redirect log messages to console
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:
%L - %m%n

log4j.logger.org.apache.http.wire=DEBUG
```

Les journaux de débogage s'impriment ensuite dans la console IDE.

Problèmes liés à la bibliothèque des producteurs

Cette section décrit les problèmes que vous pouvez rencontrer lors de l'utilisation des [Transférer vers Kinesis Video Streams](#).

Problèmes

- [Impossible de compiler le SDK du producteur](#)
- [Un flux vidéo n'apparaît pas dans la console](#)
- [Erreur : « Security token included in the request is invalid » \(le jeton de sécurité inclus dans la requête n'est pas valide\) lors de la diffusion de données à l'aide de l'application de démonstration GStreamer](#)
- [Erreur : « Failed to submit frame to Kinesis Video client » \(Échec d'envoi d'image au client Kinesis Video\)](#)

- [L'application GStreamer s'arrête avec le message « streaming arrêté, motif non négocié » sur OS X](#)
- [Erreur : « Failed to allocate heap » \(échec de l'allocation du tas\) lors de la création du Client vidéo Kinesis dans la démonstration de GStreamer sur Raspberry Pi](#)
- [Erreur : « Illegal Instruction » \(instruction non conforme\) lors de l'exécution de la démonstration de GStreamer sur Raspberry Pi](#)
- [Le chargement de la caméra échoue sur Raspberry Pi](#)
- [La caméra n'est pas trouvée sur macOS High Sierra](#)
- [Fichier jni.h introuvable lors de la compilation sur macOS High Sierra](#)
- [Erreurs curl lors de l'exécution de l'application de démonstration GStreamer](#)
- [Timestamp/range assertion lors de l'exécution sur Raspberry Pi](#)
- [Assertion sur `gst_value_set_fraction_range_full` sur Raspberry Pi](#)
- [0x3200000d STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA \(erreur sur Android\)](#)
- [La durée maximale du fragment a été atteinte \(erreur\)](#)
- [Message d'erreur « Invalid thing name passed » \(Nom d'objet non valide attribué\) lors de l'utilisation d'une autorisation IoT](#)

Impossible de compiler le SDK du producteur

Vérifiez que les bibliothèques requises se trouvent dans votre chemin d'accès. Pour le vérifier, utilisez la commande suivante :

```
env | grep LD_LIBRARY_PATH
LD_LIBRARY_PATH=/home/local/awslabs/amazon-kinesis-video-streams-producer-sdk-cpp/
kinesis-video-native-build/downloads/local/lib
```

Un flux vidéo n'apparaît pas dans la console

Pour afficher votre flux vidéo dans la console, il doit être encodé H.264 au format aVCC. Si votre flux ne s'affiche pas, vérifiez les points suivants :

- Vous [drapeaux d'adaptation NAL](#) êtes configuré pour savoir `NAL_ADAPTATION_ANNEXB_NALS` | `NAL_ADAPTATION_ANNEXB_CPD_NALS` si le flux d'origine est au Annex-B format. Il s'agit de la valeur par défaut dans le constructeur `StreamDefinition`.

- Vous fournissez correctement les données privées codec. En effet H.264, il s'agit du jeu de paramètres de séquence (SPS) et du jeu de paramètres d'image (PPS). En fonction de votre source média, ces données peuvent être extraites de la source média séparément ou codées dans l'image.

De nombreux flux élémentaires sont au format suivant, où se Ab trouve le code de Annex-B départ (001 ou 0001) :

```
Ab(Sps)Ab(Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame)... Ab(Sps)Ab(Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame)
```

Le CPD (Codec Private Data), s'il H.264 est présent dans le flux sous forme de SPS ou PPS, peut être adapté au format aVCC. À moins que le pipeline multimédia ne fournisse le CPD séparément, l'application peut extraire le CPD de la trame en recherchant la première trame Idr (qui doit contenir le SPS et le PPS), extraire les deux NALU (qui seront Ab (Sps) Ab (Pps)) et le configurer dans le CPD in. `StreamDefinition`

Erreur : « Security token included in the request is invalid » (le jeton de sécurité inclus dans la requête n'est pas valide) lors de la diffusion de données à l'aide de l'application de démonstration GStreamer

Si cette erreur se produit, il y a un problème avec vos informations d'identification. Vérifiez les paramètres suivants :

- Si vous utilisez des informations d'identification temporaires, vous devez spécifier le jeton de session.
- Vérifiez que vos informations d'identification temporaires ne sont pas arrivées à expiration.
- Vérifiez que vous disposez des droits appropriés.
- Sur macOS, vérifiez que vous n'avez pas d'informations d'identification mises en cache dans Keychain.

Erreur : « Failed to submit frame to Kinesis Video client » (Échec d'envoi d'image au client Kinesis Video)

Si cette erreur se produit, les horodatages ne sont pas correctement définis dans le flux source. Essayez les éléments suivants :

- Utilisez le dernier exemple de kit SDK qui peut avoir une mise à jour qui corrige votre problème.
- Réglez le flux de haute qualité sur un débit plus élevé et corrigez toute instabilité dans le flux source si la caméra le permet.

L'application GStreamer s'arrête avec le message « streaming arrêté, motif non négocié » sur OS X

Le streaming peut s'interrompre sur OS X avec le message suivant :

```
Debugging information: gstbasesrc.c(2939): void gst_base_src_loop(GstPad *) (): /
GstPipeline:test-pipeline/GstAutoVideoSrc:source/GstAVFVideoSrc:source-actual-src-
avfvide:
streaming stopped, reason not-negotiated (-4)
```

Une solution possible consiste à supprimer les paramètres de fréquence d'images de l'appel `gst_caps_new_simple` dans `kinesis_video_gstreamer_sample_app.cpp` :

```
GstCaps *h264_caps = gst_caps_new_simple("video/x-h264",
                                         "profile", G_TYPE_STRING, "baseline",
                                         "stream-format", G_TYPE_STRING, "avc",
                                         "alignment", G_TYPE_STRING, "au",
                                         "width", GST_TYPE_INT_RANGE, 320, 1920,
                                         "height", GST_TYPE_INT_RANGE, 240, 1080,
                                         "framerate", GST_TYPE_FRACTION_RANGE, 0,
                                         1, 30, 1,
                                         NULL);
```

Erreur : « Failed to allocate heap » (échec de l'allocation du tas) lors de la création du Client vidéo Kinesis dans la démonstration de GStreamer sur Raspberry Pi

L'exemple d'application GStreamer essaie d'allouer 512 Mo de RAM, qui peuvent ne pas être disponibles sur votre système. Vous pouvez réduire cette allocation en réduisant la valeur suivante dans `KinesisVideoProducer.cpp` :

```
device_info.storageInfo.storageSize = 512 * 1024 * 1024;
```

Erreur : « Illegal Instruction » (instruction non conforme) lors de l'exécution de la démonstration de GStreamer sur Raspberry Pi

Si vous rencontrez l'erreur suivante lors de l'exécution de la démo de GStreamer, vérifiez que vous avez compilé l'application pour la bonne version de votre appareil. (Par exemple, vérifiez que vous n'êtes pas en train de compiler pour Raspberry Pi 3 lorsque vous utilisez Raspberry Pi 2.)

```
INFO - Initializing curl.  
Illegal instruction
```

Le chargement de la caméra échoue sur Raspberry Pi

Pour vérifier si la caméra est chargée, procédez comme suit :

```
ls /dev/video*
```

Si rien n'est trouvé, procédez comme suit :

```
vcgencmd get_camera
```

La sortie doit ressembler à ce qui suit :

```
supported=1 detected=1
```

Si le pilote ne détecte pas la caméra, procédez comme suit :

1. Vérifiez la configuration de la caméra physique et vérifiez qu'elle est connectée correctement.

2. Effectuez les étapes suivantes pour mettre à jour le micrologiciel :

```
sudo rpi-update
```

3. Redémarrez le périphérique.
4. Effectuez les étapes suivantes pour charger le pilote :

```
sudo modprobe bcm2835-v4l2
```

5. Vérifiez que la caméra a été détectée :

```
ls /dev/video*
```

La caméra n'est pas trouvée sur macOS High Sierra

Sur macOS High Sierra, l'application de démonstration ne trouve pas la caméra si plus d'une caméra est disponible.

Fichier jni.h introuvable lors de la compilation sur macOS High Sierra

Pour résoudre cette erreur, mettez à jour votre installation de Xcode vers la version la plus récente.

Erreurs curl lors de l'exécution de l'application de démonstration GStreamer

Pour résoudre les erreurs curl lorsque vous exécutez l'application de démonstration GStreamer, copiez [ce fichier de certificat](#) dans `/etc/ssl/cert.pem`.

Timestamp/range assertion lors de l'exécution sur Raspberry Pi

Si une assertion de plage d'horodatage se produit lors de l'exécution, mettez à jour le micrologiciel et redémarrez le périphérique :

```
sudo rpi-update  
$ sudo reboot
```

Assertion sur `gst_value_set_fraction_range_full` sur Raspberry Pi

L'assertion suivante s'affiche si le service `uv4l` est en cours d'exécution :

```
gst_util_fraction_compare (numerator_start, denominator_start, numerator_end,  
denominator_end) < 0' failed
```

Si cela se produit, arrêtez le service uv4l et redémarrez l'application.

0x3200000d STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (erreur sur Android)

L'erreur suivante s'affiche si les [drapeaux d'adaptation NAL](#) sont incorrects pour la diffusion multimédia :

```
putKinesisVideoFrame(): Failed to put a frame with status code 0x3200000d
```

Si cette erreur se produit, fournissez le bon indicateur `.withNalAdaptationFlags` pour votre média (par exemple, `NAL_ADAPTATION_ANNEXB_CPD_NALS`). Fournissez cet indicateur dans la ligne suivante [Android](#) :

<https://github.com/aws-labs/aws-sdk-android-samples/blob/master/AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/fragment/StreamConfigurationFragment.java#L169>

La durée maximale du fragment a été atteinte (erreur)

Cette erreur se produit lorsqu'un fragment de média dans un flux dépasse la limite maximale de durée de fragment. Consultez la limite de durée maximale des fragments dans la [the section called "Quotas de service d'API pour les médias et les médias archivés"](#) section.

Pour résoudre ce problème, essayez ce qui suit :

- Si vous utilisez un webcam/USB appareil photo, effectuez l'une des opérations suivantes :
 - Si vous utilisez la fragmentation basée sur les images clés, configurez l'encodeur pour qu'il fournisse des images clés dans les 10 secondes.
 - Si vous n'utilisez pas la fragmentation basée sur des images clés, lors de la définition du flux [Rédiger et examiner le code](#) entrant, définissez la limite de durée maximale du fragment sur une valeur inférieure à 10 secondes.
 - Si vous utilisez des encodeurs logiciels (tels que x264) dans le pipeline GStreamer, vous pouvez attribuer à l'attribut `key-int-max` une valeur dans les 10 secondes. Par exemple, réglez `key-`

int-max sur 60, avec des images par seconde sur 30, pour activer les images clés toutes les 2 secondes.

- Si vous utilisez une caméra RPI, définissez l'attribut `keyframe-interval` sur une valeur inférieure à 10 secondes.
- Si vous utilisez une caméra IP (RTSP), réglez la taille du GOP sur 60.

Message d'erreur « Invalid thing name passed » (Nom d'objet non valide attribué) lors de l'utilisation d'une autorisation IoT

Pour éviter cette erreur (HTTP Error 403: Response: {"message":"Invalid thing name passed"}) lorsque vous utilisez des informations d'identification IoT pour l'autorisation, assurez-vous que la valeur de `stream-name` (paramètre obligatoire de `kvssink` élément) est identique à la valeur `deiot-thingname`. Pour de plus amples informations, veuillez consulter [Référence des paramètres de l'élément GStreamer](#).

Problèmes liés à la bibliothèque d'analyseurs de flux

Cette section décrit les problèmes que vous pouvez rencontrer lors de l'utilisation des [Diffuser à l'aide de la bibliothèque d'analyseurs](#).

Problèmes

- [Impossible d'accéder à une seule image depuis le flux](#)
- [Erreur de décodage de fragment](#)

Impossible d'accéder à une seule image depuis le flux

Pour accéder à une seule image depuis une source de streaming dans votre application grand public, vérifiez que votre flux contient les bonnes données privées du codec. Pour en savoir plus sur le format des données dans le flux, consultez [Modèle de données](#).

Pour savoir comment utiliser les données privées d'un codec pour accéder à un cadre, consultez le fichier de test suivant sur le GitHub site Web : [KinesisVideoRendererExampleTest.java](#)

Erreur de décodage de fragment

Si vos fragments ne sont pas correctement encodés dans un H.264 format et un niveau pris en charge par le navigateur, le message d'erreur suivant peut s'afficher lors de la diffusion de votre stream sur la console :

```
Fragment Decoding Error
There was an error decoding the video data. Verify that the stream contains valid H.264
content
```

Si cela se produit, vérifiez les points suivants :

- La résolution des images correspond à la résolution spécifiée dans les données privées codec.
- Le H.264 profil et le niveau des trames codées correspondent au profil et au niveau spécifiés dans les données privées du codec.
- Le navigateur prend en charge cette profile/level combinaison. La plupart des navigateurs actuels prennent en charge toutes les combinaisons de profil et de niveau.
- Les horodatages sont exacts et dans le bon ordre, et aucun horodatage en double n'est créé.
- Votre application code les données du cadre à l'aide du H.264 format.

Problèmes liés au réseau

Si vous constatez des erreurs de connexion, telles que « Expiration de la connexion » ou « Échec de la connexion », lorsque vous tentez de vous connecter à Kinesis Video Streams, cela peut être dû à des restrictions de plage d'adresses IP dans votre configuration réseau.

Si votre configuration comporte des restrictions de plage d'adresses IP pour Kinesis Video Streams, mettez à jour votre configuration réseau pour autoriser la liste des plages d'adresses IP Kinesis [Video Streams](#).

Important

La liste des plages d'adresses IP n'est pas une liste exhaustive des adresses IP Kinesis Video Streams. Incluez les plages d'adresses IP que vous voyez et sachez que les adresses IP peuvent changer au fil du temps.

Pour plus d'informations, consultez la section Plages d'[AWS adresses IP](#). Pour être averti lorsque les plages d'adresses IP changent, suivez la [procédure d'abonnement](#).

Historique du document pour Amazon Kinesis Video Streams

Le tableau suivant décrit les modifications importantes apportées à la documentation depuis la dernière version d'Amazon Kinesis Video Streams.

- Dernière version d'API : 29-11-2017
- Dernière mise à jour de la documentation : 6 janvier 2025

| Modifier | Description | Date |
|---|--|-------------------|
| C++ sur Raspberry Pi | Documentation actualisée pour l'utilisation du SDK C++ Producer sur Raspberry Pi. | 6 janvier 2025 |
| Connexion à l'agent Amazon Kinesis Video Streams Edge Edge-to-Cloud | Nouvelle version de fonctionnalité. Pour de plus amples informations, veuillez consulter Planifier l'enregistrement et le stockage vidéo . | 27 juin 2023 |
| Mise en route : envoyer des données vers un flux vidéo Kinesis | Tutoriel de base pour envoyer des données multimédia d'une caméra vers un flux vidéo Kinesis. Pour de plus amples informations, veuillez consulter Envoyer des données vers un flux vidéo Amazon Kinesis . | 21 janvier 2019 |
| Métadonnées de streaming | Vous pouvez utiliser le SDK Producer pour intégrer des métadonnées dans un flux vidéo Kinesis. Pour de plus amples informations, veuillez consulter Utilisation des | 28 septembre 2018 |

| Modifier | Description | Date |
|---|--|-----------------|
| | métadonnées de streaming avec Kinesis Video Streams. | |
| Journalisation du kit SDK C++ Producer | Vous pouvez configurer la journalisation pour les applications du kit SDK C++ Producer. Pour de plus amples informations, veuillez consulter Utiliser la journalisation avec le SDK C++ Producer. | 18 juillet 2018 |
| Streaming vidéo HLS | Vous pouvez désormais visionner un flux vidéo Kinesis via HTTP Live Streaming. Pour de plus amples informations, veuillez consulter Lecture de Kinesis Video Streams. | 13 juillet 2018 |
| Streaming à partir d'une source RTSP | Exemple d'application pour Kinesis Video Streams qui s'exécute dans un conteneur Docker et diffuse des vidéos à partir d'une source RTSP. Pour de plus amples informations, veuillez consulter RTSP et Docker. | le 20 juin 2018 |
| Plug-in GStreamer du kit SDK Producteur C++ | Illustre comment générer le C++ afin de l'utiliser en tant que destination GStreamer. Pour de plus amples informations, veuillez consulter Plug-in GStreamer - kvssink. | 15 juin 2018 |

| Modifier | Description | Date |
|---|---|--------------|
| Documentation de référence sur les rappels du kit SDK Producteur | Documentation de référence sur les rappels utilisés par le Transférer vers Kinesis Video Streams . Pour de plus amples informations, veuillez consulter Rappels du SDK pour les producteurs . | 12 juin 2018 |
| Configuration système requise | Documentation relative aux exigences en termes de mémoire et de stockage pour les appareils de producteurs et le kit SDK. Pour de plus amples informations, veuillez consulter Configuration système requise pour Amazon Kinesis Video Streams . | 30 mai 2018 |
| CloudTrail soutien | Documentation CloudTrail à utiliser pour surveiller l'utilisation des API. Pour de plus amples informations, veuillez consulter Enregistrez les appels d'API Amazon Kinesis Video Streams avec AWS CloudTrail . | 24 mai 2018 |
| Documentation de référence sur les structures du kit SDK Producteur | Documentation de référence sur les structures utilisées par le Transférer vers Kinesis Video Streams . Pour plus d'informations, consultez Structures du SDK pour les producteurs et Structures de flux vidéo Kinesis . | 7 mai 2018 |

| Modifier | Description | Date |
|---|--|-----------------|
| Documentation relative à l'exemple de moteur de rendu | Documentation relative à l'exemple d'application <code>Renderer</code> , qui montre comment décoder et afficher les images d'un flux vidéo Kinesis. Pour de plus amples informations, veuillez consulter Exemple : analyse et rendu de fragments Kinesis Video Streams . | 15 mars 2018 |
| Documentation de référence sur les limites du kit SDK Producteur | Informations sur les limites des opérations dans le C++ . Pour de plus amples informations, veuillez consulter Quotas du SDK pour les producteurs . | 13 mars 2018 |
| Contrôle | Informations sur la surveillance des métriques et des appels d'API Kinesis Video Streams à l'aide d' <code>CloudWatch</code> et <code>Amazon AWS CloudTrail</code> . Pour de plus amples informations, veuillez consulter Surveillance d'Amazon Kinesis Video Streams . | 5 février 2018 |
| Référence concernant les indicateurs d'adaptation NAL (Network Abstraction Layer) | Informations sur la configuration des indicateurs d'adaptation NAL lors de la consommation de vidéo en streaming. Pour de plus amples informations, veuillez consulter drapeaux d'adaptation NAL . | 15 janvier 2018 |

| Modifier | Description | Date |
|---|--|------------------|
| Prise en charge Android de la vidéo en streaming | Kinesis Video Streams prend désormais en charge le streaming vidéo à partir d'appareils Android. Pour de plus amples informations, veuillez consulter Android . | 12 janvier 2018 |
| Documentation sur l'exemple d'application Kinesis Video | Documentation relative à l'exemple d'application Kinesis Video, qui montre comment l'utiliser Regardez le résultat des caméras à l'aide de la bibliothèque d'analyseurs dans une application. Pour de plus amples informations, veuillez consulter KinesisVideoExemple . | 9 janvier 2018 |
| Documentation publiée pour Kinesis Video Streams | Il s'agit de la première version du Manuel du développeur Amazon Kinesis Video. | 29 novembre 2017 |

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.