



AWS Livre blanc

# Meilleures pratiques WordPress pour AWS



# Meilleures pratiques WordPress pour AWS: AWSLivre blanc

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 appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

---

# Table of Contents

Résumé .....	1
Êtes-vous Well-Architected ? .....	1
Introduction .....	2
Déploiement simple .....	3
Considérations .....	3
Approches disponibles .....	3
Amazon Lightsail .....	4
Sélection d'un plan tarifaire Amazon Lightsail .....	5
Installation WordPress .....	5
Reprise après un échec .....	6
Améliorer les performances et la rentabilité .....	7
Accélérer la diffusion de contenu .....	7
Déchargement de contenu statique .....	8
Contenu dynamique .....	8
Mise en cache des base de données .....	10
Mise en cache du bytecode .....	10
Déploiement élastique .....	12
Architecture de référence .....	12
Élargir le niveau Web .....	14
niveau Web sans état .....	16
Stockage partagé (Amazon S3 et AmazonEFS) .....	16
Niveau de données (Amazon Aurora et Amazon ElastiCache) .....	17
Conclusion .....	19
Participants .....	20
Révisions du document .....	21
Annexe A : CloudFront configuration .....	22
Origines et comportements .....	22
CloudFront création de distribution .....	22
Annexe B : configuration de contenu statique .....	26
Création d'utilisateur .....	26
Création de compartiment Amazon S3 .....	26
Création d'origine statique .....	28
Annexe C : Sauvegarde et restauration .....	29
Annexe D : Déploiement de nouveaux plug-ins et thèmes .....	31

---

Avis .....	32
AWS Glossaire .....	33
.....	xxxiv

# Meilleures pratiques WordPress pour AWS

Date de publication : 19 octobre 2021 ([Révisions du document](#))

Ce livre blanc fournit aux administrateurs système des conseils spécifiques sur la manière de démarrer avec WordPress Amazon Web Services (AWS) et d'améliorer à la fois la rentabilité du déploiement et l'expérience de l'utilisateur final. Il décrit également une architecture de référence qui répond aux exigences courantes d'évolutivité et de haute disponibilité.

## Êtes-vous Well-Architected ?

Le [AWS Well-Architected](#) Framework vous aide à comprendre les avantages et les inconvénients des décisions que vous prenez lors de la création de systèmes dans le cloud. Les six piliers du Framework vous permettent d'apprendre les bonnes pratiques d'architecture pour concevoir et exploiter des systèmes fiables, sécurisés, efficaces, rentables et durables. À l'aide du [AWS Well-Architected Tool](#), disponible gratuitement dans le [AWS Management Console](#), vous pouvez évaluer votre charge de travail par rapport à ces meilleures pratiques en répondant à une série de questions pour chaque pilier.

[Pour obtenir des conseils d'experts supplémentaires et les meilleures pratiques relatives à votre architecture cloud \(déploiements d'architecture de référence, diagrammes et livres blancs\), consultez le Centre d'architecture.AWS](#)

# Introduction

WordPress est un outil de blog open source et un système de gestion de contenu (CMS) basé sur PHP et MySQL, utilisé pour alimenter de nombreux contenus, des blogs personnels aux sites web à fort trafic.

Lorsque la première version de WordPress a été publiée en 2003, elle n'a pas été conçue dans l'optique d'utiliser des infrastructures modernes, élastiques et évolutives basées sur le cloud. Grâce au travail de la communauté WordPress et la parution de divers modules WordPress, les capacités de cette solution CMS sont en constante évolution. Aujourd'hui, il est possible de concevoir une architecture WordPress qui tire profit de la plupart des avantages du cloud AWS.

# Déploiement simple

Pour les blogs ou sites Web à faible trafic sans exigences strictes en matière de haute disponibilité, un simple déploiement d'un seul serveur peut être approprié. Ce déploiement n'est pas l'architecture la plus résiliente ou la plus évolutive, mais c'est le moyen le plus rapide et le plus économique de rendre votre site Web opérationnel.

## Rubriques

- [Considérations](#)
- [Approches disponibles](#)
- [Amazon Lightsail](#)

## Considérations

Cette discussion commence par le déploiement d'un seul serveur Web. Il peut arriver que vous deviez trop grand, par exemple :

- La machine virtuelle sur laquelle votre WordPress site Web est déployé constitue un point de défaillance unique. Un problème avec cette instance entraîne une perte de service pour votre site Web.
- La mise à l'échelle des ressources pour améliorer les performances ne peut être réalisée que par une « mise à l'échelle verticale », c'est-à-dire en augmentant la taille de la machine virtuelle qui exécute votre WordPress site Web.

## Approches disponibles

AWS propose différentes options pour le provisionnement des machines virtuelles. Il existe trois manières principales d'héberger votre propre WordPress site Web sur AWS :

- Amazon Lightsail
- Amazon Elastic Compute Cloud (AmazonEC2)
- AWS Marketplace

[Amazon Lightsail](#) est un service qui vous permet de lancer rapidement un serveur privé virtuel (une instance Lightsail) pour héberger un site Web. WordPress Lightsail est le moyen le plus facile de se

familiariser avec si vous n'avez pas besoin de types d'instances hautement configurables ou d'accès à des fonctionnalités réseau avancées.

[Amazon EC2](#) est un service Web qui fournit une capacité de calcul redimensionnable afin que vous puissiez lancer un serveur virtuel en quelques minutes. Amazon EC2 propose davantage d'options de configuration et de gestion que Lightsail, ce qui est souhaitable dans les architectures plus avancées. Vous disposez d'un accès administratif à vos EC2 instances et pouvez installer les packages logiciels de votre choix, notamment WordPress.

[AWS Marketplace](#) est une boutique en ligne dans laquelle vous pouvez trouver, acheter et déployer rapidement des logiciels compatibles avec AWS. Vous pouvez utiliser le déploiement en 1 clic pour lancer WordPress des images préconfigurées directement sur Amazon EC2 depuis votre propre AWS compte en quelques minutes. Plusieurs fournisseurs de Marketplace proposent des ready-to-run WordPress instances.

Ce livre blanc décrit l'option Lightsail en tant qu'implémentation recommandée pour un site Web à serveur unique. WordPress

## Amazon Lightsail

Lightsail est le moyen le plus facile de faire ses premiers pas avec pour AWS les développeurs, les petites entreprises, les étudiants et les autres utilisateurs ayant besoin d'une solution simple de serveur VPS privé virtuel ( ).

Le service soustrait à l'utilisateur la plupart des éléments les plus complexes de la gestion de l'infrastructure. Il s'agit donc d'un point de départ idéal si vous avez moins d'expérience en matière d'infrastructure ou lorsque vous devez vous concentrer sur la gestion de votre site Web et qu'un produit simplifié est suffisant pour répondre à vos besoins.

Avec Amazon Lightsail, vous pouvez choisir les systèmes d'exploitation Windows ou Linux/Unix et les applications Web populaires, WordPress notamment, et les déployer en un seul clic à partir de modèles préconfigurés.

À mesure que vos besoins augmentent, vous pouvez facilement dépasser les limites initiales et vous connecter à des services de AWS base de données, de stockage d'objets, de mise en cache et de distribution de contenu supplémentaires.

## Sélection d'un plan tarifaire Amazon Lightsail

Un forfait [Lightsail](#) définit le coût mensuel des ressources Lightsail que vous utilisez pour héberger votre site Web. WordPress Un certain nombre de plans sont disponibles pour couvrir divers cas d'utilisation, avec différents niveaux de CPU ressources, de mémoire, de stockage sur disque SSD (SSD) et de transfert de données. Si votre site Web est complexe, vous aurez peut-être besoin d'une instance plus grande avec davantage de ressources. Vous pouvez y parvenir en migrant votre serveur vers un forfait plus important à [l'aide de la console Web](#) ou comme décrit dans la documentation [Amazon Lightsail CLI](#).

## Installation WordPress

Lightsail fournit des modèles pour les applications couramment utilisées telles que. WordPress Ce modèle est un excellent point de départ pour gérer votre propre WordPress site Web car il est préinstallé avec la plupart des logiciels dont vous avez besoin. Vous pouvez installer des logiciels supplémentaires ou personnaliser la configuration logicielle à l'aide du terminal intégré au navigateur ou de votre propre SSH client, ou via l'interface Web d' WordPressadministration.

Amazon Lightsail a conclu un partenariat GoDaddy avec le produit Pro Sites pour WordPress aider les clients à gérer facilement et gratuitement leurs instances. Les serveurs virtuels WordPress Lightsail sont préconfigurés et optimisés pour des performances et une sécurité rapides, ce qui facilite la mise en service de WordPress votre site en un rien de temps. Les clients utilisant plusieurs WordPress instances trouvent qu'il est difficile et fastidieux de mettre à jour, de maintenir et de gérer l'ensemble de leurs sites. Grâce à cette intégration, vous pouvez facilement gérer vos multiples WordPress instances en quelques minutes et en quelques clics seulement.

Pour plus d'informations WordPress sur la gestion de Lightsail après son installation, consultez la section [Commencer à l' WordPress utiliser depuis votre instance Amazon Lightsail](#). Une fois que vous avez terminé de personnaliser votre WordPress site Web, nous vous recommandons de prendre un instantané de votre instance.

Un [instantané](#) permet de créer une image de sauvegarde de votre instance Lightsail. Il s'agit d'une copie du disque système qui stocke également la configuration d'origine de la machine (c'est-à-dire la mémoireCPU, la taille du disque et le taux de transfert de données). Les snapshots peuvent être utilisés pour revenir à une bonne configuration connue après un mauvais déploiement ou une mise à niveau.

Ce snapshot vous permet de récupérer votre serveur si nécessaire, mais également de lancer de nouvelles instances avec les mêmes personnalisations.

## Reprise après un échec

Un serveur Web unique est un point de défaillance unique. Vous devez donc vous assurer que les données de votre site Web sont sauvegardées. Le mécanisme de capture d'écran décrit précédemment peut également être utilisé à cette fin. Pour récupérer après un échec, vous pouvez restaurer une nouvelle instance à partir de votre dernier instantané. Pour réduire la quantité de données susceptibles d'être perdues lors d'une restauration, vos instantanés doivent être aussi récents que possible.

Pour minimiser le risque de perte de données, assurez-vous que des instantanés sont pris régulièrement. Vous pouvez planifier des instantanés automatiques de vos instances Lightsail Linux/Unix. Pour les étapes à suivre, veuillez consulter [Activation ou désactivation des instantanés automatiques pour des instances ou des disques dans Amazon Lightsail](#).

AWS vous recommande d'utiliser une adresse IP statique : une adresse IP publique fixe dédiée à votre compte Lightsail. Si vous devez remplacer votre instance par une autre, vous pouvez réattribuer l'adresse IP statique à la nouvelle instance. De cette manière, vous n'avez pas besoin de reconfigurer tous les systèmes externes (tels que les DNS registres) afin qu'ils pointent vers une nouvelle adresse IP à chaque fois que vous souhaitez remplacer votre instance.

# Améliorer les performances et la rentabilité

Il se peut que votre déploiement sur un serveur unique finisse par être dépassé. Dans ce cas, vous devrez peut-être envisager des options pour améliorer les performances de votre site web. Avant de migrer vers un déploiement multi-serveurs et évolutif (abordé plus loin dans ce document), vous pouvez réaliser un certain nombre d'économies en termes de performances et de coûts. Il s'agit de bonnes pratiques que vous devriez suivre dans tous les cas, même si vous passez à une architecture multi-serveurs.

Les sections suivantes présentent un certain nombre d'options qui peuvent améliorer certains aspects des performances et de la capacité de mise à l'échelle de votre site web WordPress. Certaines peuvent être appliquées à un déploiement sur un serveur unique, tandis que d'autres exploitent la capacité de mise à l'échelle de plusieurs serveurs. Bon nombre de ces modifications nécessitent l'utilisation d'un ou plusieurs plugins WordPress. Bien que différentes options soient disponibles, [W3 Total Cache](#) est un choix répandu qui combine bon nombre de ces modifications dans un seul plugin.

## Rubriques

- [Accélérer la diffusion de contenu](#)
- [Mise en cache des base de données](#)
- [Mise en cache du bytecode](#)

## Accélérer la diffusion de contenu

Tout site web WordPress doit fournir un mélange de contenu statique et de contenu dynamique. Le contenu statique inclut les images, les fichiers JavaScript ou les feuilles de style. Le contenu dynamique inclut tout ce qui est généré côté serveur à l'aide du code PHP WordPress, par exemple, les éléments de votre site qui sont générés à partir de la base de données ou personnalisés pour chaque visiteur.

Un aspect important de l'expérience de l'utilisateur final est la latence du réseau impliquée lors de la diffusion du contenu précédent aux utilisateurs du monde entier. L'accélération de la diffusion du contenu précédent améliore l'expérience de l'utilisateur final, en particulier celle des utilisateurs dispersés géographiquement dans le monde entier. Cette accélération peut être obtenue grâce à un réseau de diffusion de contenu (CDN) tel qu'Amazon CloudFront.

[Amazon CloudFront](#) est un service web qui permet de distribuer du contenu de façon simple et rentable avec une faible latence et des vitesses de transfert de données élevées par le biais de plusieurs emplacements périphériques dans le monde. Les demandes des utilisateurs sont automatiquement acheminées vers un [emplacement périphérique](#) CloudFront approprié afin de réduire la latence. Si le contenu peut être mis en cache (pendant quelques secondes, quelques minutes, voire plusieurs jours) et qu'il est déjà stocké dans un emplacement périphérique particulier, CloudFront le fournit immédiatement. Si le contenu ne doit pas être mis en cache, a expiré ou ne se trouve pas actuellement dans cet emplacement périphérique, CloudFront récupère le contenu d'une ou plusieurs sources de vérité, appelées origines (dans ce cas, l'instance Lightsail) dans la configuration CloudFront. Cette récupération s'effectue via des connexions réseau optimisées, qui permettent d'accélérer la diffusion de contenu sur votre site web. Outre l'amélioration de l'expérience de l'utilisateur final, le modèle présenté réduit également la charge sur vos serveurs d'origine et peut générer d'importantes économies de coûts.

## Déchargement de contenu statique

Le contenu statique inclut les fichiers CSS, JavaScript et image (soit les fichiers qui font partie de vos thèmes WordPress, soit les fichiers multimédias chargés par les administrateurs de contenu). Tous ces fichiers peuvent être stockés dans Amazon Simple Storage Service (Amazon S3) à l'aide d'un plugin tel que W3 Total Cache et fournis aux utilisateurs de manière évolutive et hautement disponible. [Amazon S3](#) offre une infrastructure de stockage de données hautement évolutive, fiable, à faible latence et à faible coût, accessible via des API REST. Amazon S3 stocke de manière redondante vos objets, non seulement sur plusieurs appareils, mais également sur plusieurs sites d'une région AWS, offrant ainsi des niveaux de durabilité exceptionnellement élevés.

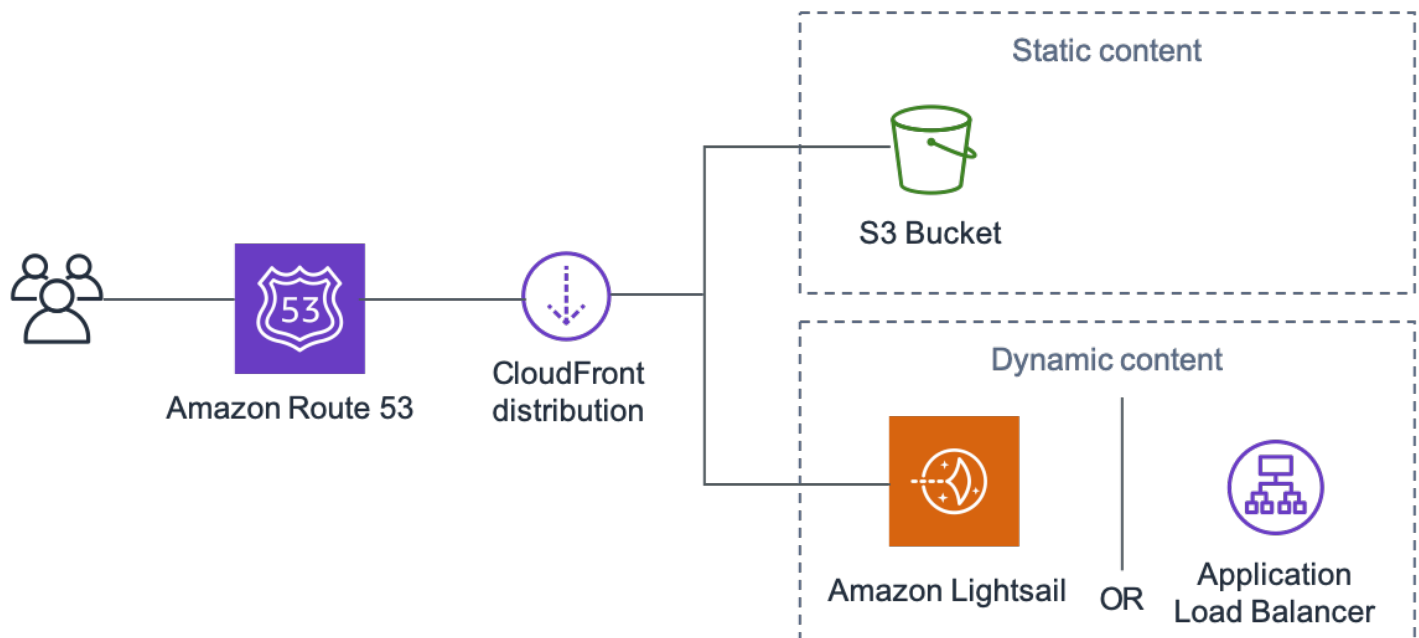
Cela a pour effet positif de délester votre instance Lightsail de cette charge de travail et de la laisser se concentrer sur la génération de contenu dynamique. Cela réduit la charge sur le serveur et constitue une étape importante vers la création d'une architecture sans état (condition préalable à la mise en œuvre de la scalabilité automatique).

Vous pouvez ensuite configurer Amazon S3 en tant qu'origine pour CloudFront afin d'améliorer la livraison de ces ressources statiques aux utilisateurs du monde entier. Bien que WordPress ne soit pas pré-intégré à Amazon S3 et CloudFront, plusieurs plugins permettent de prendre en charge ces services (par exemple, W3 Total Cache).

## Contenu dynamique

Le contenu dynamique inclut la sortie de scripts PHP WordPress côté serveur. Le contenu dynamique peut également être diffusé via CloudFront en configurant le site web WordPress en

tant qu'origine. Étant donné que le contenu dynamique inclut du contenu personnalisé, vous devez configurer CloudFront pour transférer certains cookies HTTP et en-têtes HTTP dans le cadre d'une demande envoyée à votre serveur d'origine personnalisé. CloudFront utilise les valeurs de cookie transférées dans la clé qui identifie un objet unique dans son cache. Pour optimiser l'efficacité de la mise en cache, vous devez configurer CloudFront pour ne transférer que les cookies HTTP et les en-têtes HTTP qui modifient réellement le contenu (et non les cookies qui ne sont utilisés que du côté client ou par des applications tierces, par exemple, pour l'analyse web).



### Livraison de l'ensemble du site web via Amazon CloudFront

La figure précédente inclut deux origines : l'une pour le contenu statique et l'autre pour le contenu dynamique. Pour plus d'informations sur l'implémentation, reportez-vous à l'[Annexe A : Configuration de CloudFront](#) et à l'[Annexe B : Installation et configuration des plug-ins](#).

CloudFront utilise des en-têtes de contrôle de cache standard pour déterminer s'il doit mettre en cache des réponses HTTP spécifiques et pendant combien de temps il doit le faire. Ces mêmes en-têtes de contrôle de cache sont également utilisés par les navigateurs web pour définir quand et pendant combien de temps le contenu doit être mis en cache localement pour une expérience utilisateur final plus optimale (par exemple, un fichier `.css` déjà téléchargé ne sera pas téléchargé de nouveau chaque fois qu'un visiteur récurrent consulte une page). Vous pouvez configurer les en-têtes de contrôle de cache au niveau du serveur web (par exemple, via des fichiers `.htaccess` ou des modifications du fichier `httpd.conf`) ou installer un plugin WordPress (par exemple, W3 Total Cache) pour dicter la façon dont ces en-têtes sont définis pour le contenu statique et le contenu dynamique.

## Mise en cache des base de données

La mise en cache des bases de données peut réduire considérablement la latence et augmenter le débit des applications à forte charge de lecture comme WordPress. Les performances des applications sont améliorées en stockant en mémoire les données fréquemment consultées pour un accès à faible latence (par exemple, les résultats de requêtes de base de données à forte intensité d'E/S). Lorsqu'un pourcentage élevé de requêtes est traité à partir du cache, le nombre de requêtes qui doivent atteindre la base de données est réduit, ce qui se traduit par une diminution des coûts associés au fonctionnement de la base de données.

Bien que WordPress dispose de capacités de mise en cache limitées à la base, divers plugins prennent en charge l'intégration avec [Memcached](#), un système de mise en cache d'objets mémoire largement répandu. Le plugin W3 Total Cache en est un bon exemple.

Dans les scénarios les plus simples, vous installez Memcached sur votre serveur web et vous capturez le résultat sous la forme d'un nouvel instantané. Dans ce cas, vous êtes responsable des tâches administratives associées à l'exécution d'un cache.

Une autre option consiste à utiliser un service géré tel qu'[Amazon ElastiCache](#) et à éviter cette charge opérationnelle. ElastiCache facilite le déploiement, l'exploitation et la mise à l'échelle d'un cache en mémoire distribué dans le cloud. Vous trouverez des informations sur la façon de vous connecter à vos nœuds de cluster ElastiCache dans la [documentation Amazon ElastiCache](#).

Si vous utilisez Lightsail et souhaitez accéder à un cluster ElastiCache de votre compte AWS en privé, vous pouvez le faire en utilisant l'appariage de VPC. Pour obtenir des instructions sur l'activation de l'appariage de VPC, consultez le document [Configurer l'appariage de VPC Amazon pour travailler avec les ressources AWS en dehors d'Amazon Lightsail](#).

## Mise en cache du bytecode

Chaque fois qu'un script PHP est exécuté, il est analysé et compilé. Si vous utilisez un cache de bytecode PHP, le résultat de la compilation PHP est stocké dans la RAM afin que le même script n'ait pas à être compilé encore et encore. Cela réduit la surcharge liée à l'exécution des scripts PHP, et permet d'obtenir de meilleures performances et une réduction des besoins en CPU.

Un cache de bytecode peut être installé sur n'importe quelle instance Lightsail hébergeant WordPress et peut réduire considérablement sa charge. Pour PHP 5.5 et versions ultérieures, AWS recommande d'utiliser [OPcache](#), une extension fournie avec cette version de PHP.

Notez qu'OPcache est activé par défaut dans le modèle WordPress Lightsail Bitnami, donc aucune autre action n'est requise.

# Déploiement élastique

Il existe de nombreux scénarios dans lesquels le déploiement d'un seul serveur peut ne pas être suffisant pour votre site Web. Dans ces situations, vous avez besoin d'une architecture évolutive multiserveur.

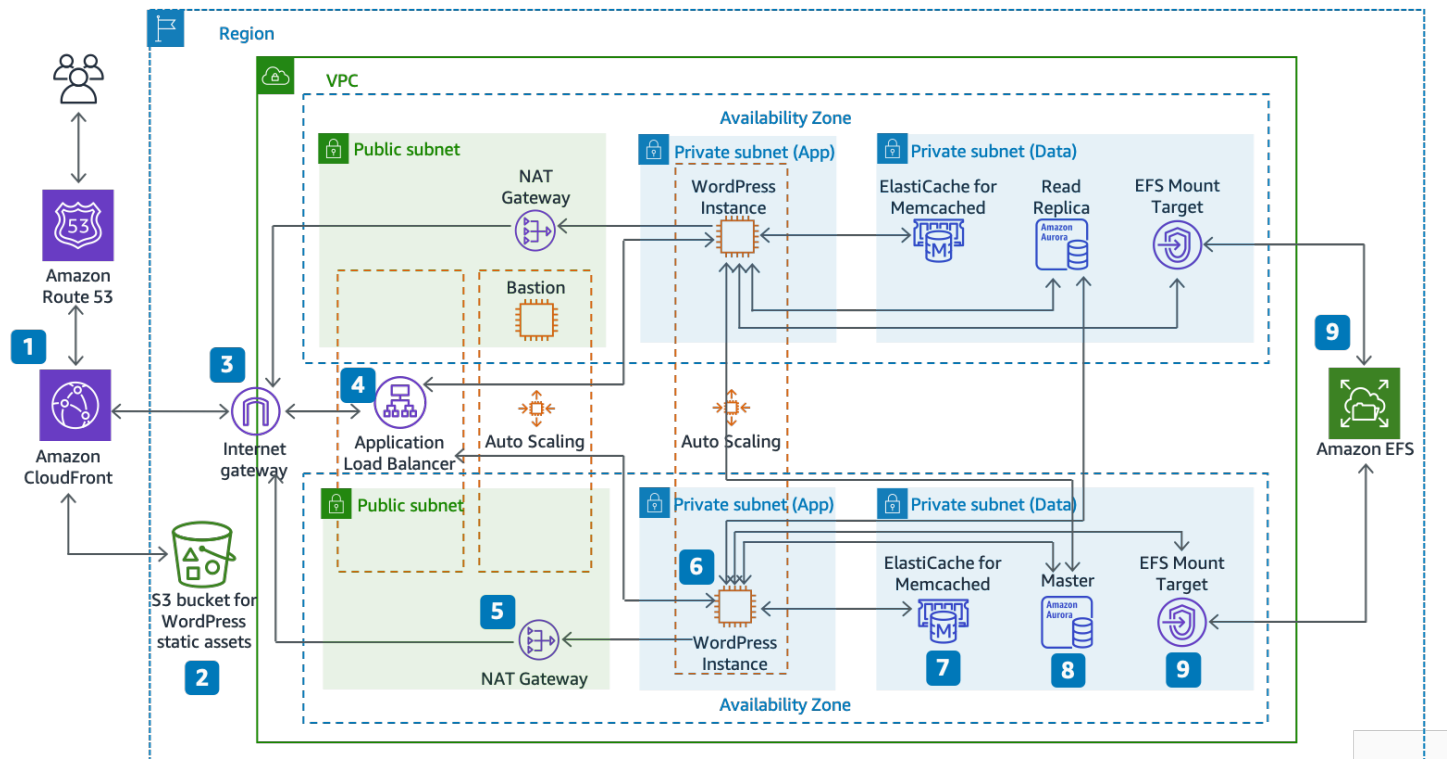
## Rubriques

- [Architecture de référence](#)
- [Élargir le niveau Web](#)
- [niveau Web sans état](#)

## Architecture de référence

L'[architecture d'hébergement WordPress sur AWS référence](#) disponible sur GitHub décrit les meilleures pratiques de déploiement WordPress AWS et inclut un ensemble de AWS CloudFormation modèles pour vous permettre d'être rapidement opérationnel. L'architecture suivante est basée sur cette architecture de référence. Dans le reste de cette section, nous passerons en revue les raisons des choix architecturaux.

La base AMI du GitHub est passée d'Amazon Linux1 à Amazon Linux2 en juillet 2021. Cependant, les modèles de déploiement de S3 n'ont pas encore été modifiés. Il est recommandé d'utiliser des modèles en GitHub cas de problème lors du déploiement de l'architecture de référence avec des modèles sur S3.



## Architecture de référence pour l'hébergement WordPress sur AWS

### Composants d'architecture

L'architecture de référence illustre un déploiement complet des meilleures pratiques pour un WordPress site Web sur AWS.

- Tout commence par la mise en cache périphérique dans Amazon CloudFront (1) pour mettre en cache le contenu à proximité des utilisateurs finaux afin d'accélérer la diffusion.
- CloudFront extrait le contenu statique d'un compartiment S3 (2) et le contenu dynamique d'un Application Load Balancer (4) situé devant les instances Web.
- Les instances Web s'exécutent dans un groupe Auto Scaling d'EC2 instances Amazon (6).
- Un ElastiCache cluster (7) met en cache les données fréquemment demandées afin d'accélérer les réponses.

Une SQL instance Amazon Aurora My (8) héberge la WordPress base de données.

- Les WordPress EC2 instances accèdent aux WordPress données partagées sur un système de fichiers Amazon EFS via un EFS Mount Target (9) dans chaque zone de disponibilité.
- Une passerelle Internet (3) permet la communication entre vos ressources VPC et celles d'Internet.

- NAT Les passerelles (5) de chaque zone de disponibilité permettent aux EC2 instances des sous-réseaux privés (App et Data) d'accéder à Internet.

Au sein d'Amazon, VPC il existe deux types de sous-réseaux : public (sous-réseau public) et privé (sous-réseau d'applications et sous-réseau de données). Les ressources déployées dans les sous-réseaux publics recevront une adresse IP publique et seront visibles publiquement sur Internet. L'Application Load Balancer (4) et un hôte Bastion pour l'administration sont déployés ici. Les ressources déployées dans les sous-réseaux privés ne reçoivent qu'une adresse IP privée et ne sont donc pas visibles publiquement sur Internet, ce qui améliore la sécurité de ces ressources. Les instances de serveur WordPress Web (6), les instances de ElastiCache cluster (7), les instances de SQL base de données Aurora My (8) et EFS Mount Targets (9) sont toutes déployées dans des sous-réseaux privés.

Le reste de cette section traite plus en détail de chacune de ces considérations.

## Élargir le niveau Web

Pour faire évoluer votre architecture mono-serveur vers une architecture évolutive multiserveur, vous devez utiliser cinq composants clés :

- EC2 Instances Amazon
- Amazon Machine Images Images (AMIs)
- Équilibreurs de charge
- Dimensionnement automatique
- Surveillance de l'état

AWS fournit une grande variété de types d'EC2 instance afin que vous puissiez choisir la meilleure configuration de serveur en termes de performances et de coûts. D'une manière générale, le type d'instance optimisé pour le calcul (par exemple, C4) peut être un bon choix pour un WordPress serveur Web. Vous pouvez déployer vos instances dans plusieurs zones de disponibilité au sein d'une même AWS région afin d'améliorer la fiabilité de l'architecture globale.

Comme vous avez le contrôle total de votre EC2 instance, vous pouvez vous connecter avec un accès root pour installer et configurer tous les composants logiciels nécessaires au fonctionnement d'un WordPress site Web. Après avoir terminé, vous pouvez enregistrer cette configuration sous

forme de fichier AMI que vous pouvez utiliser pour lancer de nouvelles instances avec toutes les personnalisations que vous avez effectuées.

Pour distribuer les demandes des utilisateurs finaux à plusieurs nœuds de serveur Web, vous avez besoin d'une solution d'équilibrage de charge. AWS fournit cette fonctionnalité via [Elastic Load Balancing](#), un service hautement disponible qui distribue le trafic à plusieurs EC2 instances. Étant donné que votre site Web diffuse du contenu à vos utilisateurs via HTTP ou HTTPS, nous vous recommandons d'utiliser l'Application Load Balancer, un équilibreur de charge de la couche application avec routage du contenu et capable d'exécuter plusieurs WordPress sites Web sur différents domaines, si nécessaire.

Elastic Load Balancing prend en charge la distribution des demandes entre plusieurs zones de disponibilité au sein d'une même AWS région. Vous pouvez également configurer un contrôle de santé afin que l'Application Load Balancer arrête automatiquement d'envoyer du trafic aux instances individuelles défaillantes (par exemple, en raison d'un problème matériel ou d'un crash logiciel). AWS recommande d'utiliser la page de connexion de l'WordPress administrateur (`/wp-login.php`) pour le contrôle de santé, car cette page confirme à la fois que le serveur Web fonctionne et qu'il est configuré pour servir correctement PHP les fichiers.

Vous pouvez choisir de créer une page de vérification de l'état personnalisée qui vérifie d'autres ressources dépendantes, telles que les ressources de base de données et de cache. Pour plus d'informations, reportez-vous à [la section Contrôles de santé de vos groupes cibles](#) dans le Guide Application Load Balancer.

L'élasticité est une caractéristique essentielle du AWS Cloud. Vous pouvez augmenter la capacité de calcul (par exemple, des serveurs Web) lorsque vous en avez besoin et en exécuter moins lorsque ce n'est pas le cas. [Amazon EC2 Auto Scaling](#) est un AWS service qui vous aide à automatiser ce provisionnement afin d'augmenter ou de diminuer votre EC2 capacité Amazon en fonction des conditions que vous définissez, sans intervention manuelle. Vous pouvez configurer Amazon EC2 Auto Scaling de telle sorte que le nombre d'EC2 instances que vous utilisez augmente de manière fluide pendant les pics de demande afin de maintenir les performances et qu'il diminue automatiquement lorsque le trafic diminue, afin de minimiser les coûts.

Elastic Load Balancing prend également en charge l'ajout et la suppression dynamiques d'EC2 hôtes Amazon dans le cadre de la rotation d'équilibrage de charge. Elastic Load Balancing lui-même augmente et diminue de manière dynamique la capacité d'équilibrage de charge afin de s'adapter aux demandes de trafic sans intervention manuelle.

## niveau Web sans état

Pour tirer parti de plusieurs serveurs Web dans une configuration de dimensionnement automatique, votre niveau Web doit être apatride. Une application sans état est une application qui n'a pas besoin de connaître les interactions précédentes et qui ne stocke aucune information de session. Dans le cas de WordPress, cela signifie que tous les utilisateurs finaux reçoivent la même réponse, quel que soit le serveur Web qui a traité leur demande. Une application sans état peut évoluer horizontalement car toute demande peut être traitée par l'une des ressources informatiques disponibles (c'est-à-dire les instances de serveur Web). Lorsque cette capacité n'est plus requise, toute ressource individuelle peut être interrompue en toute sécurité (une fois que les tâches en cours ont été épuisées). Ces ressources n'ont pas besoin d'être conscientes de la présence de leurs pairs ; il leur suffit de trouver un moyen de leur répartir la charge de travail.

En ce qui concerne le stockage des données de session utilisateur, le WordPress noyau est totalement apatride car il repose sur des cookies stockés dans le navigateur Web du client. Le stockage de session n'est pas un problème, sauf si vous avez installé un code personnalisé (par exemple, un WordPress plugin) qui repose plutôt sur PHP des sessions natives.

Cependant, WordPress il a été initialement conçu pour fonctionner sur un seul serveur. Par conséquent, il stocke certaines données dans le système de fichiers local du serveur. Lors de WordPress l'exécution dans une configuration multiserveur, cela crée un problème car il existe des incohérences entre les serveurs Web. Par exemple, si un utilisateur télécharge une nouvelle image, celle-ci n'est stockée que sur l'un des serveurs.

Cela montre pourquoi nous devons améliorer la configuration d' WordPress exécution par défaut pour déplacer les données importantes vers le stockage partagé. L'architecture des meilleures pratiques comporte une base de données en tant que couche séparée en dehors du serveur Web et utilise le stockage partagé pour stocker les téléchargements, les thèmes et les plugins des utilisateurs.

### Stockage partagé (Amazon S3 et AmazonEFS)

Par défaut, WordPress stocke les téléchargements des utilisateurs sur le système de fichiers local et n'est donc pas apatride. Par conséquent, nous devons déplacer l' WordPress installation et toutes les personnalisations utilisateur (telles que la configuration, les plugins, les thèmes et les téléchargements générés par les utilisateurs) vers une plate-forme de données partagée afin de réduire la charge sur les serveurs Web et de rendre le niveau Web apatride.

[Amazon Elastic File System](#) (AmazonEFS) fournit des systèmes de fichiers réseau évolutifs à utiliser avec des EC2 instances. Les systèmes de EFS fichiers Amazon sont répartis sur un nombre illimité

de serveurs de stockage, ce qui permet aux systèmes de fichiers de croître de manière élastique et permet un accès massivement parallèle depuis les instances. EC2 La conception distribuée d'Amazon EFS permet d'éviter les goulots d'étranglement et les contraintes inhérents aux serveurs de fichiers traditionnels.

En déplaçant l'intégralité du répertoire WordPress d'installation sur un système de EFS fichiers et en le montant dans chacune de vos EC2 instances au démarrage, votre WordPress site et toutes ses données sont automatiquement stockés sur un système de fichiers distribué qui ne dépend d'aucune EC2 instance, ce qui rend votre niveau Web complètement apatriote. L'avantage de cette architecture est que vous n'avez pas besoin d'installer de plugins et de thèmes à chaque lancement d'une nouvelle instance, et vous pouvez accélérer considérablement l'installation et la restauration des WordPress instances. Il est également plus facile de déployer les modifications apportées aux plugins et aux thèmes dans WordPress, comme indiqué dans la section [Considérations relatives au déploiement](#) de ce document.

Pour garantir des performances optimales de votre site Web lorsqu'il est exécuté à partir d'un système de EFS fichiers, vérifiez les paramètres de configuration recommandés pour Amazon EFS et Opcache sur l'[architecture de AWS référence pour WordPress](#).

Vous avez également la possibilité de télécharger toutes les ressources statiques, telles que les images et JavaScript les fichiersCSS, dans un compartiment S3 avec CloudFront mise en cache à l'avant. Le mécanisme permettant de le faire dans une architecture multiserveur est exactement le même que pour une architecture à serveur unique, comme indiqué dans la section [Contenu statique](#) de ce livre blanc. Les avantages sont les mêmes que ceux de l'architecture à serveur unique : vous pouvez déléguer le travail associé à la diffusion de vos actifs statiques sur Amazon S3 CloudFront, permettant ainsi à vos serveurs Web de se concentrer uniquement sur la génération de contenu dynamique et de répondre à un plus grand nombre de demandes d'utilisateurs par serveur Web.

## Niveau de données (Amazon Aurora et Amazon ElastiCache)

L' WordPress installation étant stockée sur un système de fichiers réseau partagé, évolutif et distribué, et les ressources statiques étant fournies par Amazon S3, vous pouvez concentrer votre attention sur le composant dynamique restant : la base de données. Comme pour le niveau de stockage, la base de données ne doit pas dépendre d'un seul serveur, elle ne peut donc pas être hébergée sur l'un des serveurs Web. Hébergez plutôt la WordPress base de données sur Amazon Aurora.

[Amazon Aurora](#) est une base de données relationnelle SQL compatible avec My SQL et Postgre conçue pour le cloud, qui associe les performances et la disponibilité des bases de données

commerciales haut de gamme à la simplicité et à la rentabilité des bases de données open source. Aurora My SQL améliore mes SQL performances et ma disponibilité en intégrant au moteur de base de données un système de stockage distribué spécialement conçu, soutenu par. SSD Il est tolérant aux pannes et autoréparant, réplique six copies de vos données dans trois zones de disponibilité, est conçu pour une disponibilité supérieure à 99,99 % et sauvegarde en permanence vos données dans Amazon S3. Amazon Aurora est conçu pour détecter automatiquement les incidents de base de données et redémarrer sans effectuer de récupération sur incident et sans régénérer le cache de la base de données.

Amazon Aurora propose un certain nombre de [types d'instances](#) adaptés à différents profils d'applications, notamment des instances optimisées pour la mémoire et des instances évolutives. Pour améliorer les performances de votre base de données, vous pouvez sélectionner un type d'instance de grande taille afin de fournir davantage de ressources CPU et de mémoire.

Amazon Aurora gère automatiquement le basculement entre l'instance principale et [Aurora Replicas](#) afin que vos applications puissent reprendre les opérations de base de données aussi rapidement que possible sans intervention administrative manuelle. Le basculement dure généralement moins de 30 secondes.

Après avoir créé au moins une réplique Aurora, connectez-vous à votre instance principale à l'aide du point de terminaison du cluster pour permettre à votre application de basculer automatiquement en cas de défaillance de l'instance principale. Vous pouvez créer jusqu'à 15 réplicas en lecture à faible latence dans trois zones de disponibilité.

Au fur et à mesure que votre base de données évolue, votre cache de base de données devra également évoluer. Comme indiqué précédemment dans la section Mise en [cache de base de données](#), ElastiCache possède des fonctionnalités permettant de dimensionner le cache sur plusieurs nœuds d'un ElastiCache cluster et sur plusieurs zones de disponibilité d'une région pour une meilleure disponibilité. Lorsque vous dimensionnez votre ElastiCache cluster, assurez-vous de configurer votre plug-in de mise en cache pour qu'il se connecte à l'aide du point de terminaison de configuration afin qu'il WordPress puisse utiliser les nouveaux nœuds de cluster lorsqu'ils sont ajoutés et arrêter d'utiliser les anciens nœuds de cluster lorsqu'ils sont supprimés. Vous devez également configurer vos serveurs Web pour utiliser le [client de ElastiCache cluster PHP et les mettre à jour AMI pour](#) enregistrer cette modification.

## Conclusion

AWS présente de nombreuses options d'architecture pour l'exécution de WordPress. L'option la plus simple consiste à installer un serveur unique pour les sites web à faible trafic. Pour les sites web plus avancés, les administrateurs de sites peuvent ajouter plusieurs autres options, chacune représentant une amélioration progressive en termes de disponibilité et de capacité de mise à l'échelle. Les administrateurs peuvent sélectionner les fonctionnalités qui correspondent le mieux à leurs exigences et à leur budget.

# Participants

Ont contribué à la préparation du présent document :

- Paul Lewis, architecte de solutions, Amazon Web Services
- Ronan Guilfoyle, architecte de solutions, Amazon Web Services
- Andreas Chatzakis, responsable des architectes de solutions, Amazon Web Services
- Jibril Touzi, gestionnaire des comptes techniques, Amazon Web Services
- Hakmin Kim, architecte de solutions partenaires de migration, Amazon Web Services

## Révisions du document

Pour recevoir une notification des mises à jour de ce livre blanc, abonnez-vous au RSS flux.

Modification	Description	Date
<a href="#">Livre blanc mis à jour</a>	Mis à jour pour modifier l'architecture de référence et AWS pour WordPress le plugin.	19 octobre 2021
<a href="#">Livre blanc mis à jour</a>	Mis à jour pour inclure de nouvelles approches de déploiement et AWS pour WordPress le plugin.	30 octobre 2019
<a href="#">Livre blanc mis à jour</a>	Mise à jour pour clarifier les messages relatifs aux produits Amazon Aurora.	1 février 2018
<a href="#">Livre blanc mis à jour</a>	Mis à jour pour inclure AWS les services lancés depuis la première publication.	1er décembre 2017
<a href="#">Publication initiale</a>	Publication initiale.	1 décembre 2014

## Annexe A : CloudFront configuration

Pour obtenir des performances et une efficacité optimales lors de l'utilisation d'Amazon CloudFront avec votre WordPress site Web, il est important de configurer le site Web correctement en fonction des différents types de contenu proposés.

Rubriques

- [Origines et comportements](#)
- [CloudFront création de distribution](#)

### Origines et comportements

Une [origine](#) est un emplacement où CloudFront envoie des demandes de contenu qu'elle distribue via les emplacements périphériques. En fonction de votre implémentation, vous pouvez avoir une ou deux origines. Un pour le contenu dynamique (l'instance Lightsail dans l'option de [déploiement sur un seul serveur, ou l'Application Load Balancer](#) dans l'option de [déploiement élastique](#)) utilisant une [origine](#) personnalisée. Vous pouvez avoir une deuxième origine vers laquelle vous CloudFront diriger pour votre contenu statique. Dans l'[architecture de référence](#) précédente, il s'agit d'un compartiment S3. Lorsque vous utilisez Amazon S3 comme origine de votre distribution, vous devez utiliser une [politique de compartiment](#) pour rendre le contenu accessible au public.

[Les comportements](#) vous permettent de définir des règles qui régissent la manière dont votre contenu est mis en CloudFront cache et, par conséquent, de déterminer l'efficacité du cache. Les comportements vous permettent de contrôler le protocole et les HTTP méthodes par lesquels votre site Web est accessible. Ils vous permettent également de contrôler si vous souhaitez transmettre HTTP des en-têtes, des cookies ou des chaînes de requête à votre backend (et, dans l'affirmative, lesquels). Les comportements s'appliquent à des modèles de URL trajectoire spécifiques.

### CloudFront création de distribution

Créez une distribution CloudFront Web en suivant la distribution. L'origine et le comportement par défaut créés automatiquement seront utilisés pour le contenu dynamique. Créez quatre comportements supplémentaires pour personnaliser davantage la façon dont les demandes statiques et dynamiques sont traitées. Le tableau suivant résume les propriétés de configuration des cinq comportements.

Tableau 1 : Résumé des propriétés de configuration pour les CloudFront comportements

Propriété	Statique	Dynamique (administrateur)	Dynamique (frontal)
Chemins (comportements)	wp-content/* wp-includes/*	wp-admin/* wp-login.php	par défaut (*)
Protocoles	HTTP et HTTPS	Rediriger vers HTTPS	HTTP et HTTPS
HTTP méthodes	GET, HEAD	ALL	ALL
HTTP en-têtes	NONE	ALL	Host (Hôte) CloudFront-Forwarded-Proto CloudFront-Is-Mobile-Viewer CloudFront-Is-Tablet-Viewer CloudFront-Is-Desktop-Viewer
Cookies	NONE	ALL	commentaire_* wordpress_* wp-paramètres-*
Chaînes de requête	YES (invalidation)	YES	YES

Pour le comportement par défaut, AWS recommande la configuration suivante :

- Permettez à la politique du protocole d'origine de correspondre à celle du visualiseur, de sorte que si les spectateurs CloudFront se connectent à CloudFront l'utilisateur HTTPS, ils se connectent

également à votre origine HTTPS en utilisant, pour obtenir end-to-end le cryptage. Notez que cela nécessite l'installation d'un SSL certifié sécurisé sur l'équilibreur de charge. Pour plus de détails, reportez-vous à [la section HTTPS Demande de communication entre CloudFront et votre origine personnalisée](#).

- Autorisez toutes les HTTP méthodes, car les parties dynamiques du site Web nécessitent à la fois des POST demandes GET et des demandes (par exemple, POST pour prendre en charge les formulaires de soumission de commentaires).
- Transférez uniquement les cookies qui font varier le WordPress résultat, par exemple `>wordpress_*wp-settings-*`, `etcomment_*`. Vous devez étendre cette liste si vous avez installé des plugins qui dépendent d'autres cookies ne figurant pas dans la liste.
- Transférez uniquement HTTP les en-têtes qui affectent la sortie de WordPress, par exemple, `Host`, `CloudFront-Forwarded-Proto`, `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer`, et `CloudFront-is-Tablet-Viewer` :
  - `Host` permet d'WordPress héberger plusieurs sites Web sur le même site d'origine.
  - `CloudFront-Forwarded-Proto` permet de mettre en cache différentes versions de pages selon qu'elles sont accessibles via HTTP ou HTTPS.
  - `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer`, et `CloudFront-is-Tablet-Viewer` permettent de personnaliser le résultat de vos thèmes en fonction du type d'appareil de l'utilisateur final.
- Transférez toutes les chaînes de requête vers le cache en fonction de leurs valeurs, car WordPress elles peuvent également être utilisées pour invalider des objets mis en cache.

Si vous souhaitez diffuser votre site Web sous un nom de domaine personnalisé (ce n'est pas le cas `*.cloudfront.net`), entrez le nom approprié URIs sous Noms de domaine alternatifs dans les paramètres de distribution. Dans ce cas, vous avez également besoin d'un SSL certifié de votre nom de domaine personnalisé. Vous pouvez [demander](#) SSL des certificats via le AWS Certificate Manager et les configurer par rapport à une CloudFront distribution.

À présent, créez deux autres comportements de cache pour le contenu dynamique : un pour la page de connexion (modèle de chemin `:wp-login.php`) et un pour le tableau de bord d'administration (modèle de chemin `:wp-admin/*`). Les paramètres de ces deux comportements sont les mêmes :

- Appliquez une politique de protocole de visualisation de HTTPS Only.
- Autorisez toutes les HTTP méthodes.
- Cache basé sur tous les HTTP en-têtes.

- Forward all cookies (Transmettre les cookies)
- Forward (Transmettre et mettre en cache) en fonction de toutes les chaînes de requête.

La raison de cette configuration est que cette section du site Web est hautement personnalisée et ne compte généralement que quelques utilisateurs. L'efficacité de la mise en cache n'est donc pas une préoccupation majeure. L'objectif est de simplifier la configuration afin de garantir une compatibilité maximale avec tous les plugins installés en transmettant tous les cookies et en-têtes à l'origine.

Par défaut, tout est WordPress stocké localement sur le serveur Web, à savoir le stockage par blocs (AmazonEBS) pour le [déploiement sur un seul serveur](#) et le stockage de fichiers (AmazonEFS) pour [le déploiement élastique](#). Outre la réduction des coûts de stockage et de transfert de données, le transfert de ressources statiques vers Amazon S3 offre évolutivité, disponibilité des données, sécurité et performances. Plusieurs plug-ins facilitent le transfert de contenu statique vers Amazon S3 ; l'un d'entre eux est [W3 Total Cache](#), également abordé dans [l'annexe B : Installation et configuration des plug-ins](#).

## Annexe B : configuration de contenu statique

Par défaut, tout est WordPress stocké localement sur le serveur Web, à savoir le stockage par blocs (AmazonEBS) pour le [déploiement sur un seul serveur](#) et le stockage de fichiers (AmazonEFS) pour [le déploiement élastique](#). Outre la réduction des coûts de stockage et de transfert de données, le transfert de ressources statiques vers Amazon S3 offre évolutivité, disponibilité des données, sécurité et performances.

Dans cet exemple, le plugin W3 Total Cache (W3TC) est utilisé pour stocker des actifs statiques sur Amazon S3. Cependant, il existe d'autres plugins dotés de fonctionnalités similaires. Si vous souhaitez utiliser une alternative, vous pouvez ajuster les étapes suivantes en conséquence. Les étapes se réfèrent uniquement aux fonctionnalités ou aux paramètres pertinents dans cet exemple. Le présent document n'a pas pour objectif de décrire en détail tous les paramètres. Reportez-vous à la [page du plugin W3 Total Cache](#) sur [wordpress.org](#) pour plus d'informations.

### Création d'utilisateur

Vous devez créer un utilisateur pour que le WordPress plugin stocke les actifs statiques dans Amazon S3. Pour connaître les étapes à suivre, reportez-vous à [la section Création d'un utilisateur dans votre AWS compte](#).

Remarque : Les rôles constituent un meilleur moyen de gérer l'accès aux AWS ressources, mais au moment de la rédaction de cet article, le plugin W3 Total Cache ne prend pas en charge [les rôles](#).

Prenez note des informations d'identification de sécurité de l'utilisateur et stockez-les de manière sécurisée. Vous aurez besoin de ces informations d'identification ultérieurement.

### Création de compartiment Amazon S3

1. Créez d'abord un compartiment Amazon S3 dans la AWS région de votre choix. Pour connaître les étapes à suivre, reportez-vous à [la section Création d'un bucket](#). Activez l'hébergement de site Web statique pour le compartiment en suivant le [didacticiel : configuration d'un site web statique sur Amazon S3](#).
2. Créez une politique pour fournir à l'utilisateur créé précédemment l'accès au compartiment S3 spécifié, et attachez-la à l'utilisateur. Pour connaître les étapes de création de la politique suivante, reportez-vous à la section [Gestion des politiques](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1389783689000",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::wp-demo",
        "arn:aws:s3:::wp-demo/*"
      ]
    }
  ]
}
```

3. Installez et activez le plugin W3TC depuis le panneau d' WordPress administration.
4. Accédez à la section Paramètres généraux de la configuration du plugin et assurez-vous que le cache du navigateur CDNest activé.
5. Dans la liste déroulante de la CDN configuration, choisissez Origin Push : Amazon CloudFront (Amazon S3 est à l'origine de cette option).
6. Accédez à la section Browser Cache de la configuration du plugin et activez les en-têtes expires, cache control et entity tag (ETag).
7. Activez également l'option Empêcher la mise en cache des objets après la modification des paramètres afin qu'une nouvelle chaîne de requête soit générée et ajoutée aux objets chaque fois que des paramètres sont modifiés.
8. Accédez à la CDN section de configuration du plugin et entrez les informations de sécurité de l'utilisateur que vous avez créé précédemment, ainsi que le nom du compartiment S3.
9. Si vous diffusez votre site Web via le CloudFront URL, entrez le nom de domaine de distribution dans la case correspondante. Sinon, saisissez-en un ou plusieurs CNAMEs pour vos noms de domaine personnalisés.

10 Enfin, exportez la bibliothèque multimédia et téléchargez les fichiers wp-includes, les fichiers de thème et les fichiers personnalisés sur Amazon S3 à l'aide du plugin W3TC. Ces fonctions de téléchargement sont disponibles dans la section Général de la page de CDN configuration.

## Création d'origine statique

Maintenant que les fichiers statiques sont stockés sur Amazon S3, revenez à la CloudFront configuration dans la CloudFront console et configurez Amazon S3 comme origine pour le contenu statique. Pour ce faire, ajoutez une deuxième origine pointant vers le compartiment S3 que vous avez créé à cette fin. Créez ensuite deux autres comportements de cache, un pour chacun des deux dossiers (wp-content et wp-includes) qui devraient utiliser l'origine S3 plutôt que l'origine par défaut pour le contenu dynamique. Configurez les deux de la même manière :

- Ne HTTP GET répondez qu'aux demandes.
- Amazon S3 ne modifie pas sa sortie en fonction des cookies ou des HTTP en-têtes. Vous pouvez donc améliorer l'efficacité de la mise en cache en ne les transférant pas vers l'origine via CloudFront
- Bien que ces comportements ne s'appliquent qu'au contenu statique (qui n'accepte aucun paramètre), vous transmettez les chaînes de requête à l'origine. Cela vous permet d'utiliser des chaînes de requête comme identificateurs de version pour invalider instantanément, par exemple, les anciens CSS fichiers lors du déploiement de nouvelles versions. Pour plus d'informations, consultez le manuel [Amazon CloudFront Developer Guide](#).

### Note

Après avoir ajouté les comportements d'origine statiques à votre CloudFront distribution, vérifiez l'ordre afin de vous assurer que les comportements relatifs wp-admin/\* au contenu statique wp-login.php ont une priorité supérieure à celle des comportements relatifs au contenu statique. Sinon, vous risquez de constater un comportement étrange lors de l'accès à votre panneau d'administration.

## Annexe C : Sauvegarde et restauration

La reprise sur incident est plus rapide et plus facile à réaliser dans AWS que dans les environnements d'hébergement traditionnels. Par exemple, vous pouvez lancer une instance de remplacement en quelques minutes en réponse à une défaillance matérielle ou utiliser le basculement automatique dans bon nombre de nos services managés pour annuler l'impact d'un redémarrage dû à une maintenance de routine.

Cependant, vous devez toujours vous assurer d'avoir sauvegardé les bonnes données pour pouvoir les récupérer avec succès. Pour rétablir la disponibilité d'un site web WordPress, vous devez être capable de récupérer les composants suivants :

- Installation et configuration du système d'exploitation (OS) et des services (Apache, MySQL, etc.)
- Code et configuration de l'application WordPress
- Thèmes et plug-ins WordPress
- Chargements (par exemple, fichiers multimédias pour des publications)
- Contenu de la base de données (publications, commentaires, etc.)

AWS propose différentes méthodes pour sauvegarder et restaurer les données et les ressources de vos applications web.

Ce livre blanc a abordé précédemment l'utilisation des instantanés Lightsail pour protéger toutes les données stockées sur le stockage local de l'instance. Si votre site web WordPress ne fonctionne que sur l'instance Lightsail, des instantanés Lightsail réguliers devraient suffire pour que vous puissiez récupérer votre site web WordPress dans son intégralité. Toutefois, vous perdrez toujours toutes les modifications appliquées à votre site web depuis la prise du dernier instantané si vous effectuez une restauration à partir d'un instantané.

Dans un déploiement multiserveur, vous devez sauvegarder chacun des composants décrits précédemment à l'aide de différents mécanismes. Chaque composant peut avoir des exigences différentes en matière de fréquence de sauvegarde, par exemple, l'installation et la configuration du système d'exploitation et de WordPress changeront beaucoup moins fréquemment que le contenu généré par les utilisateurs et, par conséquent, pourront être sauvegardés moins fréquemment sans perdre de données en cas de récupération.

Pour sauvegarder l'installation et la configuration du système d'exploitation et des services, ainsi que le code et la configuration de l'application WordPress, vous pouvez créer une AMI d'une instance

EC2 correctement configurée. Les AMI peuvent avoir deux objectifs : servir de sauvegarde de l'état de l'instance et servir de modèle lors du lancement de nouvelles instances.

Pour sauvegarder le code et la configuration de l'application WordPress, vous devez utiliser des AMI et également des sauvegardes Aurora.

Pour sauvegarder les thèmes et plugins WordPress installés sur votre site web, sauvegardez le compartiment Amazon S3 ou le système de fichiers Amazon EFS sur lequel ils sont stockés.

- Pour les thèmes et les plugins stockés dans un compartiment S3, vous pouvez activer la [réplication entre régions](#) afin que tous les objets chargés dans votre compartiment principal soient automatiquement répliqués dans votre compartiment de sauvegarde dans une autre région AWS. La réplication entre régions nécessite que la [gestion des versions](#) soit activée à la fois sur vos compartiments source et de destination. Vous bénéficiez alors d'une couche de protection supplémentaire et pouvez revenir à une version précédente d'un objet donné de votre compartiment.
- Pour les thèmes et les plugins stockés sur un système de fichiers EFS, vous pouvez créer un AWS Data Pipeline pour copier les données de votre système de fichiers EFS de production vers un autre système de fichiers EFS, comme indiqué dans la page de documentation [Sauvegarde de vos systèmes de fichiers Amazon EFS](#). Vous pouvez également sauvegarder un système de fichiers EFS à l'aide de n'importe quelle application de sauvegarde que vous utilisez déjà.
- Pour sauvegarder les chargements des utilisateurs, vous devez suivre les étapes décrites précédemment pour la sauvegarde des thèmes et plugins WordPress.
- Pour sauvegarder le contenu de la base de données, vous devez utiliser la [sauvegarde Aurora](#). Aurora sauvegarde automatiquement votre volume de cluster et conserve les données de restauration pendant la durée de la période de rétention des sauvegardes. Comme les sauvegardes Aurora sont continues et progressives, vous pouvez rapidement procéder à une restauration à un point quelconque de la période de rétention des sauvegardes. Aucun impact sur les performances ni interruption du service de base de données ne se produit lors de l'écriture des données de sauvegarde. Vous pouvez définir la période de rétention des sauvegardes sur une période de 1 à 35 jours. Vous pouvez également créer des [instantanés de base de données manuels](#), qui sont conservés jusqu'à ce que vous les supprimiez. Les instantanés de base de données manuels sont utiles pour les sauvegardes et l'archivage à long terme.

## Annexe D : Déploiement de nouveaux plugins et thèmes

Peu de sites web restent statiques. Dans la plupart des cas, vous ajouterez périodiquement des thèmes et des plugins WordPress accessibles au public ou mettez à niveau vers une version WordPress plus récente. Dans d'autres cas, vous développerez vos propres thèmes et plugins personnalisés à partir de zéro.

Chaque fois que vous apportez un changement structurel à votre installation WordPress, il existe un certain risque d'introduire des problèmes imprévus. Au minimum, faites une sauvegarde du code de votre application, de la configuration et de la base de données avant d'appliquer tout changement important (comme l'installation d'un nouveau plugin). Pour les sites web commerciaux ou autres, testez d'abord ces modifications dans un environnement intermédiaire distinct. Avec AWS, il est facile de répliquer la configuration de votre environnement de production et d'exécuter l'ensemble du processus de déploiement en toute sécurité. Une fois vos tests terminés, vous pouvez simplement supprimer votre environnement de test et cesser de payer pour ces ressources. Ce livre blanc abordera ultérieurement certaines considérations spécifiques à WordPress.

Certains plugins écrivent des informations de configuration dans la table de base de données `wp_options` (ou introduisent des modifications de schéma de base de données), tandis que d'autres créent des fichiers de configuration dans le répertoire d'installation de WordPress. Étant donné que nous avons déplacé la base de données et le stockage vers des plateformes partagées, ces modifications sont immédiatement disponibles sur toutes vos instances en cours d'exécution, sans aucun effort supplémentaire de votre part.

Lors du déploiement de nouveaux thèmes dans WordPress, quelques efforts supplémentaires pourront être nécessaires. Si vous utilisez Amazon EFS uniquement pour stocker tous vos fichiers d'installation WordPress, les nouveaux thèmes seront immédiatement disponibles sur toutes les instances en cours d'exécution. Toutefois, si vous déchargez du contenu statique vers Amazon S3, vous devez en effectuer une copie sur le bon emplacement de compartiment. Les plugins tels que W3 Total Cache vous permettent de lancer manuellement cette tâche. Vous pouvez également automatiser cette étape dans le cadre d'un processus de génération.

Étant donné que les ressources de thème peuvent être mises en cache sur CloudFront et dans le navigateur, vous devez invalider les anciennes versions lorsque vous déployez des modifications. La meilleure façon d'y parvenir est d'inclure une sorte d'identifiant de version dans votre objet. Cet identifiant peut être une chaîne de requête avec un horodatage ou une chaîne aléatoire. Si vous utilisez le plugin W3 Total Cache, vous pouvez mettre à jour une chaîne de requête multimédia qui est ajoutée aux URL des fichiers multimédias.

## Avis

Il incombe aux clients de procéder à une évaluation indépendante des informations contenues dans le présent document. Ce document : (a) est fourni à titre informatif uniquement, (b) représente les offres de AWS produits et les pratiques actuelles, qui sont sujettes à modification sans préavis, et (c) ne donne lieu à aucun engagement ni aucune assurance de la part d'AWS et de ses sociétés apparentées, fournisseurs ou concédants de licence. AWS les produits ou services sont fournis « tels quels » sans garanties, déclarations ou conditions d'aucune sorte, qu'elles soient explicites ou implicites. Les responsabilités et obligations d'AWS vis-à-vis de ses clients sont régies par AWS les contrats. Le présent document ne fait partie d'aucun, et ne modifie aucun, contrat entre AWS et ses clients.

© 2023, Amazon Web Services, Inc. ou ses sociétés apparentées. Tous droits réservés.

# AWS Glossaire

Pour la AWS terminologie la plus récente, consultez le [AWS glossaire](#) dans la Glossaire AWS référence.

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.