aws

Administratorhandbuch

Amazon DCV-Sitzungsmanager

Amazon DCV-Sitzungsmanager: Administratorhandbuch

Table of Contents

Was ist Amazon DCV Session Manager?

> **Note**
>
> Amazon DCV war zuvor als NICE DCV bekannt.

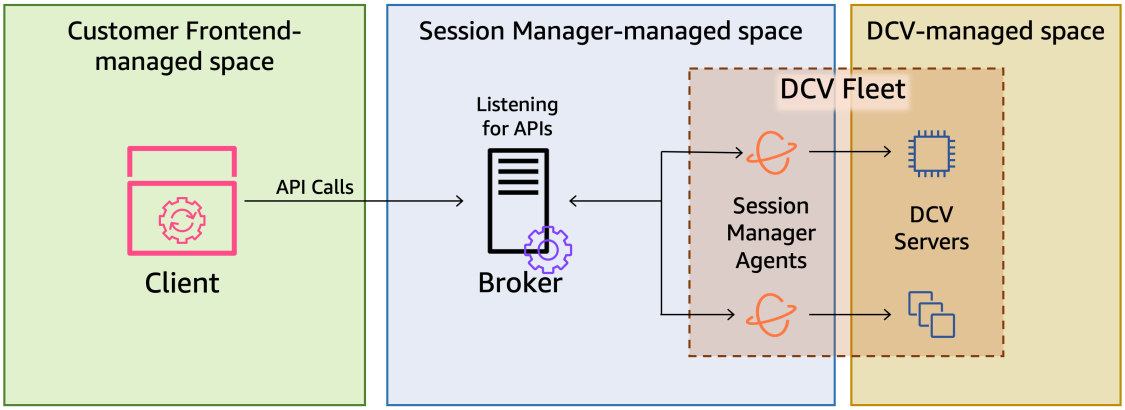
Amazon DCV Session Manager besteht aus installierbaren Softwarepaketen (einem Agenten und einem Broker) und einer Anwendungsprogrammierschnittstelle (API), die es Entwicklern und unabhängigen Softwareanbietern (ISVs) erleichtern, Frontend-Anwendungen zu erstellen, die den Lebenszyklus von Amazon DCV-Sitzungen auf einer Flotte von Amazon DCV-Servern programmgesteuert erstellen und verwalten.

In diesem Handbuch wird erklärt, wie Sie den Session Manager Agent und den Broker installieren und konfigurieren. Weitere Informationen zur Verwendung des Session Managers APIs finden Sie im Amazon DCV Session Manager Developer Guide.

Themen

Wie funktioniert Session Manager

Das folgende Diagramm zeigt die allgemeinen Komponenten von Session Manager.



Broker

Der Broker ist ein Webserver, der den Session Manager hostet und verfügbar macht. APIs Es empfängt und verarbeitet API-Anfragen zur Verwaltung von Amazon DCV-Sitzungen vom Kunden und leitet die Anweisungen dann an die entsprechenden Agenten weiter. Der Broker muss auf einem Host installiert sein, der von Ihren Amazon DCV-Servern getrennt ist, aber er muss für den Client zugänglich sein und er muss auf die Agents zugreifen können.

Kundendienstmitarbeiter

Der Agent ist auf jedem Amazon DCV-Server in der Flotte installiert. Die Agenten erhalten Anweisungen vom Broker und führen sie auf ihren jeweiligen Amazon DCV-Servern aus. Die Agenten überwachen auch den Status der Amazon DCV-Server und senden regelmäßig Status-Updates an den Broker zurück.

APIs

Session Manager stellt eine Reihe von REST-Anwendungsprogrammierschnittstellen (APIs) zur Verfügung, die zur Verwaltung von Amazon DCV-Sitzungen auf einer Flotte von Amazon DCV-Servern verwendet werden können. Sie APIs werden auf dem Broker gehostet und von diesem bereitgestellt. Entwickler können benutzerdefinierte Sitzungsverwaltungsclients erstellen, die den aufrufen APIs.

Client

Der Client ist die Front-End-Anwendung oder das Portal, das Sie entwickeln, um den Session Manager aufzurufen APIs, die vom Broker verfügbar gemacht werden. Endbenutzer verwenden den Client, um die auf den Amazon DCV-Servern der Flotte gehosteten Sitzungen zu verwalten.

Zugriffstoken

Um eine API-Anfrage zu stellen, müssen Sie ein Zugriffstoken bereitstellen. Token können vom registrierten Client vom Broker oder einem externen Autorisierungsserver angefordert werden APIs. Um Token anzufordern und darauf zuzugreifen, muss die Client-API gültige Anmeldeinformationen bereitstellen.

Client-API

Die Client-API wird mithilfe von Swagger Codegen aus der Session Manager-API-Definitionsdatei generiert. Die Client-API wird verwendet, um API-Anfragen zu stellen.

Amazon DCV-Sitzung

Eine Amazon DCV-Sitzung ist eine Zeitspanne, in der der Amazon DCV-Server Verbindungen von einem Client annehmen kann. Bevor Ihre Kunden eine Verbindung zu einer Amazon DCV-Sitzung herstellen können, müssen Sie eine Amazon DCV-Sitzung auf dem Amazon DCV-Server erstellen. Amazon DCV unterstützt sowohl Konsolen- als auch virtuelle Sitzungen, und jede Sitzung hat einen bestimmten Besitzer und eine Reihe von Berechtigungen. Sie verwenden den Session ManagerAPIs , um den Lebenszyklus von Amazon DCV-Sitzungen zu verwalten. Amazon DCV-Sitzungen können sich in einem der folgenden Zustände befinden:

- `CREATING`— Der Broker ist dabei, die Sitzung zu erstellen.
- `READY`— Die Sitzung ist bereit, Client-Verbindungen anzunehmen.
- `DELETING`— Die Sitzung wird gelöscht.
- `DELETED`— Die Sitzung wurde gelöscht.
- `UNKNOWN`— Der Status der Sitzung konnte nicht ermittelt werden. Der Broker und der Agent können möglicherweise nicht kommunizieren.

Feature

DCV Session Manager bietet die folgenden Funktionen:

- Stellt Amazon DCV-Sitzungsinformationen bereit — ruft Informationen über die Sitzungen ab, die auf mehreren Amazon DCV-Servern ausgeführt werden.
- Verwalten Sie den Lebenszyklus für mehrere Amazon DCV-Sitzungen — erstellen oder löschen Sie mehrere Sitzungen für mehrere Benutzer auf mehreren Amazon DCV-Servern mit einer API-Anfrage.

- Unterstützt Tags — Verwenden Sie benutzerdefinierte Tags, um beim Erstellen von Sitzungen eine Gruppe von Amazon DCV-Servern als Ziel zu verwenden.
- Verwaltet Berechtigungen für mehrere Amazon DCV-Sitzungen — ändern Sie Benutzerberechtigungen für mehrere Sitzungen mit einer API-Anfrage.
- Stellt Verbindungsinformationen bereit — ruft Client-Verbindungsinformationen für Amazon DCV-Sitzungen ab.
- Unterstützt Cloud- und lokale Server — Verwenden Sie Session Manager auf AWS, vor Ort oder mit alternativen Cloud-basierten Servern.

Einschränkungen

Session Manager bietet keine Funktionen zur Ressourcenbereitstellung. Wenn Sie Amazon DCV auf EC2 Amazon-Instances ausführen, müssen Sie möglicherweise zusätzliche AWS Dienste wie Amazon EC2 Auto Scaling verwenden, um die Skalierung Ihrer Infrastruktur zu verwalten.

Preisgestaltung

Session Manager ist für AWS Kunden, die EC2 Instances ausführen, kostenlos verfügbar.

Kunden vor Ort benötigen eine Amazon DCV Plus- oder Amazon DCV Professional Plus-Lizenz. Informationen zum Kauf einer Amazon DCV Plus- oder Amazon DCV Professional Plus-Lizenz finden Sie unter So kaufen Sie auf der Amazon DCV-Website und finden Sie einen Amazon DCV-Händler oder -Wiederverkäufer in Ihrer Region. Damit alle Kunden vor Ort mit dem Amazon DCV Session Manager experimentieren können, werden die Lizenzanforderungen erst ab Amazon DCV Version 2021.0 durchgesetzt.

Weitere Informationen finden Sie unter Lizenzierung des Amazon DCV-Servers im Amazon DCV-Administratorhandbuch.

Anforderungen für Amazon DCV Session Manager

Der Amazon DCV Session Manager Agent und der Broker haben die folgenden Anforderungen.

	Broker	Kundendienstmitarbeiter
Betriebssystem	• Amazon Linux 2 • Amazon Linux 2023	• Windows

	Broker	Kundendienstmitarbeiter

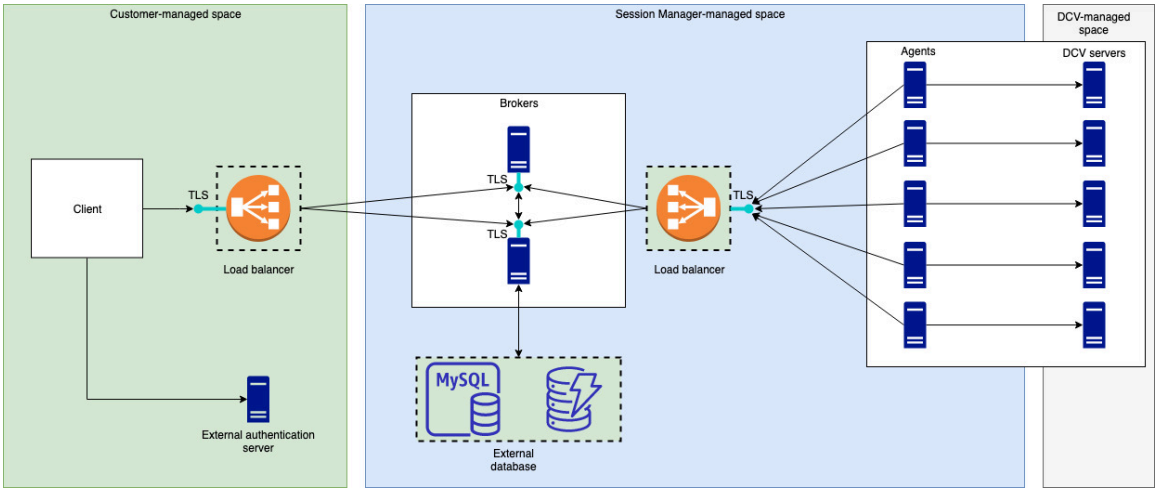
	• CentOS Stream 9
	• RHEL 8.x
	• RHEL 9.x
	• Rocky Linux 8.5 oder höher
	• Rocky Linux 9.x
	• Ubuntu 22.04
	• Ubuntu 24.04
		• Windows 11
		• Windows Server 2025
		• Windows Server 2022
		• Windows Server 2019
		• Windows Server 2016
		• Linux-Server
		  • Amazon Linux 2
		  • Amazon Linux 2023
		  • CentOS Stream 9
		  • RHEL 8.x
		  • RHEL 9.x
		  • Rocky Linux 8.5 oder höher
		  • Rocky Linux 9.x
		  • Ubuntu 22.04
		  • Ubuntu 24.04
		  • SUSE Linux Enterprise 15 mit oder höher SP6
		• macOS
		  • macOS 13 (Ventura)
		  • macOS 14 (Sonoma)
		  • macOS 15 (Sequoia)

Architektur	• 64-Bit x86 • 64-Bit-ARM	• 64-Bit x86 • 64-Bit-ARM (Amazon Linux 2, Amazon Linux 2023, CentOS 9.x, RHEL 8.x/9.x, Rocky 8.x/9.x, Ubuntu 22.04/24.04 und macOS 13/14/15)

	Broker	Kundendienstmitarbeiter
Arbeitsspeicher	8 GB	4 GB
Amazon DCV-Version	Amazon DCV 2020.2 und höher	Amazon DCV 2020.2 und höher
Zusätzliche Anforderungen	Java 11	-

Netzwerk- und Konnektivitätsanforderungen

Das folgende Diagramm bietet einen allgemeinen Überblick über die Netzwerk- und Konnektivitätsanforderungen von Session Manager.



Der Broker muss auf einem separaten Host installiert sein, er muss jedoch über eine Netzwerkverbindung mit den Agenten auf den Amazon DCV-Servern verfügen. Wenn Sie sich zur Verbesserung der Verfügbarkeit für mehrere Broker entscheiden, müssen Sie jeden Broker auf einem separaten Host installieren und konfigurieren und einen oder mehrere Load Balancer verwenden, um den Datenverkehr zwischen dem Client und den Brokern sowie den Brokern und den Agenten zu verwalten. Die Broker sollten auch in der Lage sein, miteinander zu kommunizieren, um Informationen über die Amazon DCV-Server und -Sitzungen auszutauschen. Die Broker können ihre Schlüssel und Statusdaten in einer externen Datenbank speichern und haben diese Informationen

nach einem Neustart oder einer Kündigung zur Verfügung. Dies trägt dazu bei, das Risiko des Verlusts wichtiger Broker-Informationen zu verringern, indem sie in der externen Datenbank gespeichert werden. Sie können sie später abrufen. Wenn Sie sich dafür entscheiden, müssen Sie die externe Datenbank einrichten und die Broker konfigurieren. DynamoDB, MariaDB und MySQL werden unterstützt. Die Konfigurationsparameter sind in der Broker-Konfigurationsdatei aufgeführt.

Die Agents müssen in der Lage sein, sichere, persistente, bidirektionale HTTPs Verbindungen mit dem Broker herzustellen.

Ihr Client oder Ihre Frontend-Anwendung muss auf den Broker zugreifen können, um den aufrufen zu können. APIs Der Client sollte auch auf Ihren Authentifizierungsserver zugreifen können.

Amazon DCV Session Manager einrichten

Im folgenden Abschnitt wird erklärt, wie Sie Session Manager mit einem einzelnen Broker und mehreren Agenten installieren. Sie können mehrere Broker verwenden, um die Skalierbarkeit und Leistung zu verbessern. Weitere Informationen finden Sie unter Sitzungsmanager skalieren.

Gehen Sie wie folgt vor, um Amazon DCV Session Manager einzurichten:

Schritte

- Schritt 1: Bereiten Sie die Amazon DCV-Server vor
- Schritt 2: Den Amazon DCV Session Manager-Broker einrichten
- Schritt 3: Den Amazon DCV Session Manager-Agenten einrichten
- Schritt 4: Konfigurieren Sie den Amazon DCV-Server so, dass er den Broker als Authentifizierungsserver verwendet
- Schritt 5: Überprüfen Sie die Installationen

Schritt 1: Bereiten Sie die Amazon DCV-Server vor

Sie benötigen eine Flotte von Amazon DCV-Servern, mit denen Sie Session Manager verwenden möchten. Weitere Informationen zur Installation von Amazon DCV-Servern finden Sie unter Installation des Amazon DCV-Servers im Amazon DCV-Administratorhandbuch.

Auf Linux- und macOS-Amazon-DCV-Servern verwendet Session Manager einen lokalen Dienstbenutzer mit dem Namen`dcvsmagent`. Dieser Benutzer wird automatisch erstellt, wenn der Session Manager-Agent installiert wird. Sie müssen diesem Service-Benutzer Administratorrechte für Amazon DCV gewähren, damit er Aktionen im Namen anderer Benutzer ausführen kann. Gehen Sie wie folgt vor, um dem Benutzer des Session Manager-Service Administratorrechte zu gewähren:

So fügen Sie den lokalen Dienstbenutzer für Linux- und macOS-Amazon-DCV-Server hinzu

1. Öffnen Sie `/etc/dcv/dcv.conf` mit Ihrem bevorzugten Texteditor.
2. Fügen Sie den `administrators` Parameter dem `[security]` Abschnitt hinzu und geben Sie den Session Manager-Benutzer an. Zum Beispiel:

```
[security]
administrators=["dcvsmagent"]
```

3. Speichern und schließen Sie die Datei.
4. Stoppen Sie den Amazon DCV-Server und starten Sie ihn neu.

Session Manager kann Amazon DCV-Sitzungen nur im Namen von Benutzern erstellen, die bereits auf dem Amazon DCV-Server existieren. Wenn eine Anfrage zur Erstellung einer Sitzung für einen Benutzer gestellt wird, der nicht existiert, schlägt die Anfrage fehl. Daher müssen Sie sicherstellen, dass jeder vorgesehene Endbenutzer über einen gültigen Systembenutzer auf dem Amazon DCV-Server verfügt.

> **Tip**
>
> Wenn Sie beabsichtigen, mehrere Broker-Hosts oder Amazon DCV-Server mit Agenten zu verwenden, empfehlen wir Ihnen, nur einen Broker und einen Amazon DCV-Server mit einem Agenten zu konfigurieren, indem Sie die folgenden Schritte ausführen: Amazon Machine Images (AMI) der Hosts mit den abgeschlossenen Konfigurationen erstellen und dann verwenden, AMIs um die verbleibenden Broker und Amazon DCV-Server zu starten. Alternativ können Sie AWS Systems Manager verwenden, um die Befehle auf mehreren Instanzen remote auszuführen.

Schritt 2: Den Amazon DCV Session Manager-Broker einrichten

Der Broker muss auf einem Linux-Host installiert sein. Weitere Informationen zu den unterstützten Linux-Distributionen finden Sie unterAnforderungen für Amazon DCV Session Manager. Installieren Sie den Broker auf einem Host, der vom Agenten und dem Amazon DCV-Serverhost getrennt ist. Der Host kann in einem anderen privaten Netzwerk installiert werden, muss jedoch in der Lage sein, eine Verbindung zum Agenten herzustellen und mit ihm zu kommunizieren.

Um den Broker zu installieren und zu starten

1. Connect zu dem Host her, auf dem Sie den Broker installieren möchten.
2. Die -Pakete sind digital mit einer sicheren GPG-Signatur signiert. Damit der Paketmanager die Paketsignatur überprüfen kann, müssen Sie den Amazon DCV-GPG-Schlüssel importieren. Führen Sie den folgenden Befehl aus, um den Amazon DCV-GPG-Schlüssel zu importieren.

   - Amazon Linux 2, RHEL, CentOS und Rocky Linux

```
$ sudo rpm --import https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY
```

- Ubuntu

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY
```

```
$ gpg --import NICE-GPG-KEY
```

3. Laden Sie das Installationspaket herunter.

- Amazon Linux 2

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerBrokers/nice-
dcv-session-manager-broker-2025.0.539-1.el7.noarch.rpm
```

- Amazon Linux 2023

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerBrokers/nice-
dcv-session-manager-broker-2025.0.539-1.amzn2023.noarch.rpm
```

- RHEL 8.x und Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerBrokers/nice-
dcv-session-manager-broker-2025.0.539-1.el8.noarch.rpm
```

- CentOS 9.x, RHEL 9.x und Rocky Linux 9.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerBrokers/nice-
dcv-session-manager-broker-2025.0.539-1.el9.noarch.rpm
```

- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerBrokers/nice-
dcv-session-manager-broker_2025.0.539-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerBrokers/nice-
dcv-session-manager-broker_2025.0.539-1_all.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerBrokers/nice-
dcv-session-manager-broker_2025.0.539-1_all.ubuntu2404.deb
```

4. Installieren Sie das Paket .

- Amazon Linux 2

```
$ sudo yum install -y ./nice-dcv-session-manager-
broker-2025.0.539-1.el7.noarch.rpm
```

- Amazon Linux 2023

```
$ sudo yum install -y ./nice-dcv-session-manager-
broker-2025.0.539-1.amzn2023.noarch.rpm
```

- RHEL 8.x und Rocky Linux 8.x

```
$ sudo yum install -y ./nice-dcv-session-manager-
broker-2025.0.539-1.el8.noarch.rpm
```

- CentOS 9.x, RHEL 9.x und Rocky Linux 9.x

```
$ sudo yum install -y ./nice-dcv-session-manager-
broker-2025.0.539-1.el9.noarch.rpm
```

- Ubuntu 20.04

```
$ sudo apt install -y ./nice-dcv-session-manager-
broker_2025.0.539-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ sudo apt install -y ./nice-dcv-session-manager-
broker_2025.0.539-1_all.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ sudo apt install -y ./nice-dcv-session-manager-
broker_2025.0.539-1_all.ubuntu2404.deb
```

5. Stellen Sie sicher, dass die Standardversion der Java-Umgebung 11 ist

```
$ java -version
```

   Wenn nicht, können Sie explizit das Java-Home-Verzeichnis festlegen, das der Broker als Ziel für die richtige Java-Version verwendet. Dies erfolgt durch das Einstellen des Parameters `broker-java-home` in der Broker-Konfigurationsdatei. Weitere Informationen finden Sie unter Broker-Konfigurationsdatei.

6. Starten Sie den Brokerdienst und stellen Sie sicher, dass er bei jedem Start der Instanz automatisch gestartet wird.

```
$  sudo systemctl start dcv-session-manager-broker && sudo systemctl enable dcv-session-manager-broker
```

7. Platzieren Sie eine Kopie des selbstsignierten Zertifikats des Brokers in Ihrem Benutzerverzeichnis. Sie benötigen es, wenn Sie die Agenten im nächsten Schritt installieren.

```
sudo cp /var/lib/dcvsmbroker/security/dcvsmbroker_ca.pem $HOME
```

Schritt 3: Den Amazon DCV Session Manager-Agenten einrichten

Der Agent muss auf allen Amazon DCV-Serverhosts in der Flotte installiert sein. Der Agent kann sowohl auf Windows-, Linux- als auch auf MacOS-Hosts installiert werden. Weitere Informationen zu den unterstützten Betriebssystemen finden Sie unterAnforderungen für Amazon DCV Session Manager.

Voraussetzungen

Der Amazon DCV-Server muss auf dem Host installiert werden, bevor der Agent installiert wird.

Linux host

> **Note**
>
> Der Session Manager-Agent ist für die unter Anforderungen aufgeführten Linux-Distributionen und -Architekturen verfügbar:

Die folgenden Anweisungen beziehen sich auf die Installation des Agenten auf 64-Bit-x86-Hosts. Um den Agenten auf 64-Bit-ARM-Hosts zu installieren, *x86_64* ersetzen Sie ihn durch`aarch64`. Ersetzen Sie für Ubuntu *amd64* durch`arm64`.

Um den Agenten auf einem Linux-Host zu installieren

1. Die -Pakete sind digital mit einer sicheren GPG-Signatur signiert. Damit der Paketmanager die Paketsignatur überprüfen kann, müssen Sie den Amazon DCV-GPG-Schlüssel importieren. Führen Sie den folgenden Befehl aus, um den Amazon DCV-GPG-Schlüssel zu importieren.

   - Amazon Linux 2, RHEL, CentOS und SUSE Linux Enterprise

     ```
     $ sudo rpm --import https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY
     ```

   - Ubuntu

     ```
     $ wget https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY
     ```

     ```
     $ gpg --import NICE-GPG-KEY
     ```

2. Laden Sie das Installationspaket herunter.

   - Amazon Linux 2

     ```
     $  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
     nice-dcv-session-manager-agent-2025.0.801-1.el7.x86_64.rpm
     ```

   - Amazon Linux 2023

     ```
     $  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
     nice-dcv-session-manager-agent-2025.0.888-1.amzn2023.x86_64.rpm
     ```

   - RHEL 8.x und Rocky Linux 8.x

     ```
     $  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
     nice-dcv-session-manager-agent-2025.0.888-1.el8.x86_64.rpm
     ```

   - CentOS 9.x, RHEL 9.x und Rocky Linux 9.x

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent-2025.0.888-1.el9.x86_64.rpm
```

- Ubuntu 20.04

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent_2025.0.888-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent_2025.0.888-1_amd64.ubuntu2204.deb
```

- Ubuntu 24.04

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent_2025.0.888-1_amd64.ubuntu2404.deb
```

- SUSE Linux Enterprise 12

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent-2025.0.888-1.sles12.x86_64.rpm
```

- SUSE Linux Enterprise 15

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent-2025.0.888-1.sles15.x86_64.rpm
```

3. Installieren Sie das Paket .

- Amazon Linux 2

```
$  sudo yum install -y ./nice-dcv-session-manager-
agent-2025.0.888-1.el7.x86_64.rpm
```

- Amazon Linux 2023

```
$  sudo yum install -y ./nice-dcv-session-manager-
agent-2025.0.888-1.amzn2023.x86_64.rpm
```

- RHEL 8.x und Rocky Linux 8.x

```
$  sudo yum install -y ./nice-dcv-session-manager-
agent-2025.0.888-1.el8.x86_64.rpm
```

- CentOS 9.x, RHEL 9.x und Rocky Linux 9.x

```
$  sudo yum install -y ./nice-dcv-session-manager-
agent-2025.0.888-1.el9.x86_64.rpm
```

- Ubuntu 20.04

```
$  sudo apt install ./nice-dcv-session-manager-
agent_2025.0.888-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04

```
$  sudo apt install ./nice-dcv-session-manager-
agent_2025.0.888-1_amd64.ubuntu2204.deb
```

- Ubuntu 24.04

```
$  sudo apt install ./nice-dcv-session-manager-
agent_2025.0.888-1_amd64.ubuntu2404.deb
```

- SUSE Linux Enterprise 12

```
$ sudo zypper install ./nice-dcv-session-manager-
agent-2025.0.888-1.sles12.x86_64.rpm
```

- SUSE Linux Enterprise 15

```
$ sudo zypper install ./nice-dcv-session-manager-
agent-2025.0.888-1.sles15.x86_64.rpm
```

4. Platzieren Sie eine Kopie des selbstsignierten Zertifikats des Brokers (das Sie im vorherigen Schritt kopiert haben) im `/etc/dcv-session-manager-agent/` Verzeichnis auf dem Agenten.
5. Öffnen Sie `/etc/dcv-session-manager-agent/agent.conf` mit Ihrem bevorzugten Texteditor und gehen Sie wie folgt vor.

- Geben Sie für `broker_host` den DNS-Namen des Hosts an, auf dem der Broker installiert ist.

  > **Important**
  >
  > Wenn der Broker auf einer EC2 Amazon-Instance läuft, müssen `broker_host` Sie die private IPv4-Adresse der Instance angeben.

- (Optional) Geben Sie für den Port an`broker_port`, über den mit dem Broker kommuniziert werden soll. Standardmäßig kommunizieren der Agent und der Broker über den Port`8445`. Ändern Sie dies nur, wenn Sie einen anderen Port verwenden müssen. Wenn Sie es ändern, stellen Sie sicher, dass der Broker so konfiguriert ist, dass er denselben Port verwendet.
- Geben Sie für `ca_file` den vollständigen Pfad der Zertifikatsdatei an, die Sie im vorherigen Schritt kopiert haben. Zum Beispiel:

  ```
  ca_file = '/etc/dcv-session-manager-agent/broker_cert.pem'
  ```

  Wenn Sie die TLS-Überprüfung deaktivieren möchten, legen Sie alternativ die Einstellung `tls_strict` auf fest`false`.

6. Speichern und schließen Sie die Datei.
7. Führen Sie den folgenden Befehl aus, um den Agenten zu starten.

   ```
   $ sudo systemctl start dcv-session-manager-agent
   ```

Windows host

Um den Agenten auf einem Windows-Host zu installieren

1. Laden Sie das Agent-Installationsprogramm herunter.
2. Führen Sie das Installationsprogramm aus. Klicken Sie auf der Willkommensseite auf Weiter.
3. Lesen Sie auf dem EULA-Bildschirm die Lizenzvereinbarung sorgfältig durch. Wenn Sie damit einverstanden sind, wählen Sie Ich akzeptiere die Bedingungen und dann Weiter.
4. Um mit der Installation zu beginnen, wählen Sie Installieren.

5. Platzieren Sie eine Kopie des selbstsignierten Zertifikats des Brokers (das Sie im vorherigen Schritt kopiert haben) in den `C:\Program Files\NICE\DCVSessionManagerAgent\conf\` Ordner auf dem Agenten.
6. Öffnen Sie das Programm `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` mit Ihrem bevorzugten Texteditor, und gehen Sie dann wie folgt vor:

   - Geben Sie für `broker_host` den DNS-Namen des Hosts an, auf dem der Broker installiert ist.

     > **Important**
     >
     > Wenn der Broker auf einer EC2 Amazon-Instance läuft, müssen `broker_host` Sie die private IPv4 Adresse der Instance angeben.

   - (Optional) Geben Sie für den Port an`broker_port`, über den mit dem Broker kommuniziert werden soll. Standardmäßig kommunizieren der Agent und der Broker über den Port`8445`. Ändern Sie dies nur, wenn Sie einen anderen Port verwenden müssen. Wenn Sie es ändern, stellen Sie sicher, dass der Broker so konfiguriert ist, dass er denselben Port verwendet.
   - Geben Sie für `ca_file` den vollständigen Pfad der Zertifikatsdatei an, die Sie im vorherigen Schritt kopiert haben. Zum Beispiel:

     ```
     ca_file = 'C:\Program Files\NICE\DCVSessionManagerAgent\conf\broker_cert.pem'
     ```

     Wenn Sie die TLS-Überprüfung deaktivieren möchten, legen Sie alternativ die Einstellung `tls_strict` auf fest`false`.
7. Speichern und schließen Sie die Datei.
8. Beenden Sie den Agent-Dienst und starten Sie ihn neu, damit die Änderungen wirksam werden. Führen Sie die folgenden Befehle an der Eingabeaufforderung aus.

   ```
   C:\> sc stop DcvSessionManagerAgentService
   ```

   ```
   C:\> sc start DcvSessionManagerAgentService
   ```

macOS host

Um den Agenten auf einem macOS-Host zu installieren

1. Laden Sie das Agent-Installationsprogramm herunter oder verwenden Sie die Befehlszeile:

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent-2025.0.888-macos-arm64.pkg
```

2. Installieren Sie das Paket . Sie können entweder die Befehlszeile oder den Agent-Installer verwenden:

   - Befehlszeile:

```
$ sudo installer -pkg ./nice-dcv-session-manager-agent-2025.0.888-macos-
arm64.pkg -target /
```

   - Agent-Installationsprogramm: Doppelklicken Sie auf die `.pkg` Datei und folgen Sie dem Installationsassistenten.

3. Platzieren Sie eine Kopie des selbstsignierten Zertifikats des Brokers (das Sie im vorherigen Schritt kopiert haben) in das `/etc/dcv-session-manager-agent/` Verzeichnis auf dem Agenten.

4. Öffnen Sie `/etc/dcv-session-manager-agent/agent.conf` mit Ihrem bevorzugten Texteditor und gehen Sie wie folgt vor.

   - Geben Sie für `broker_host` den DNS-Namen des Hosts an, auf dem der Broker installiert ist.

     > **Important**
     >
     > Wenn der Broker auf einer EC2 Amazon-Instance läuft, müssen `broker_host` Sie die private IPv4 Adresse der Instance angeben.

   - (Optional) Geben Sie für den Port an`broker_port`, über den mit dem Broker kommuniziert werden soll. Standardmäßig kommunizieren der Agent und der Broker über den Port`8445`. Ändern Sie dies nur, wenn Sie einen anderen Port verwenden müssen. Wenn Sie es ändern, stellen Sie sicher, dass der Broker so konfiguriert ist, dass er denselben Port verwendet.

- Geben Sie für `ca_file` den vollständigen Pfad der Zertifikatsdatei an, die Sie im vorherigen Schritt kopiert haben. Zum Beispiel:

  ```
  ca_file = '/usr/local/etc/dcv-session-manager-agent/broker_cert.pem'
  ```

  Wenn Sie die TLS-Überprüfung deaktivieren möchten, legen Sie alternativ die Einstellung `tls_strict` auf fest`false`.

5. Speichern und schließen Sie die Datei.
6. Starten Sie den Agent-Dienst.

   ```
   $ sudo launchctl load /Library/LaunchDaemons/com.amazon.dcv.session-
   manager.agent.plist
   ```

Schritt 4: Konfigurieren Sie den Amazon DCV-Server so, dass er den Broker als Authentifizierungsserver verwendet

Konfigurieren Sie den Amazon DCV-Server so, dass er den Broker als externen Authentifizierungsserver für die Validierung von Client-Verbindungstoken verwendet. Sie müssen den Amazon DCV-Server auch so konfigurieren, dass er der selbstsignierten CA des Brokers vertraut.

Linux Amazon DCV server

So fügen Sie den lokalen Dienstbenutzer für Linux-Amazon-DCV-Server hinzu

1. Öffnen Sie `/etc/dcv/dcv.conf` mit Ihrem bevorzugten Texteditor.
2. Fügen Sie die `auth-token-verifier` Parameter `ca-file` und zum `[security]` Abschnitt hinzu.

   - Geben Sie für `ca-file` den Pfad zur selbstsignierten Zertifizierungsstelle des Brokers an, die Sie im vorherigen Schritt auf den Host kopiert haben.
   - Geben Sie für `auth-token-verifier` die URL für den Token-Verifier auf dem Broker im folgenden Format an:. `https://broker_ip_or_dns:port/agent/validate-authentication-token` Geben Sie den Port an, der für die Broker-Agent-Kommunikation verwendet wird. Dieser ist standardmäßig 8445. Wenn Sie den Broker auf einer EC2 Amazon-Instance ausführen, müssen Sie die private DNS- oder private IP-Adresse verwenden.

Beispiel

```
[security]
ca-file="/etc/dcv-session-manager-agent/broker_cert.pem"
auth-token-verifier="https://my-sm-broker.com:8445/agent/validate-
authentication-token"
```

3. Speichern und schließen Sie die Datei.
4. Stoppen Sie den Amazon DCV-Server und starten Sie ihn neu. Weitere Informationen finden Sie unter Stoppen des Amazon DCV-Servers und Starten des Amazon DCV-Servers im Amazon DCV-Administratorhandbuch.

Windows Amazon DCV server

Auf Windows Amazon DCV-Servern

1. Öffnen Sie den Windows-Registrierungseditor und navigieren Sie zur Taste HKEY_/USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/security.
2. Öffnen Sie den Parameter ca-file.
3. Geben Sie unter Wertdaten den Pfad zur selbstsignierten Zertifizierungsstelle des Brokers an, die Sie im vorherigen Schritt auf den Host kopiert haben.

> **Note**
>
> Wenn der Parameter nicht existiert, erstellen Sie einen neuen Zeichenkettenparameter und geben Sie ihm `ca-file` einen Namen.

4. Öffnen Sie den auth-token-verifierParameter.
5. Geben Sie für Wertdaten die URL für den Token-Verifier auf dem Broker im folgenden Format an:`https://`*`broker_ip_or_dns`*`:`*`port`*`/agent/validate-authentication-token`.
6. Geben Sie den Port an, der für die Broker-Agent-Kommunikation verwendet wird. Dieser ist standardmäßig 8445. Wenn Sie den Broker auf einer EC2 Amazon-Instance ausführen, müssen Sie die private DNS- oder private IP-Adresse verwenden.

> **Note**
>
> Wenn der Parameter nicht existiert, erstellen Sie einen neuen Zeichenkettenparameter und geben Sie ihm einen Namen`auth-token-verifier`.

7. Klicken Sie auf OK und schließen Sie den Windows Registrierungs-Editor.
8. Stoppen Sie den Amazon DCV-Server und starten Sie ihn neu. Weitere Informationen finden Sie unter Stoppen des Amazon DCV-Servers und Starten des Amazon DCV-Servers im Amazon DCV-Administratorhandbuch.

macOS Amazon DCV server

So fügen Sie den lokalen Dienstbenutzer für macOS Amazon DCV-Server hinzu

1. Öffnen Sie `/etc/dcv/dcv.conf` mit Ihrem bevorzugten Texteditor.
2. Fügen Sie die `auth-token-verifier` Parameter `ca-file` und zum `[security]` Abschnitt hinzu.
   - Geben Sie für `ca-file` den Pfad zur selbstsignierten Zertifizierungsstelle des Brokers an, die Sie im vorherigen Schritt auf den Host kopiert haben.
   - Geben Sie für `auth-token-verifier` die URL für den Token-Verifier auf dem Broker im folgenden Format an:. `https://`*`broker_ip_or_dns`*`:`*`port`*`/agent/validate-authentication-token` Geben Sie den Port an, der für die Broker-Agent-Kommunikation verwendet wird. Dieser ist standardmäßig 8445. Wenn Sie den Broker auf einer EC2 Amazon-Instance ausführen, müssen Sie die private DNS- oder private IP-Adresse verwenden.

     Beispiel

     ```
     [security]
     ca-file="/usr/local/etc/dcv-session-manager-agent/broker_cert.pem"
     auth-token-verifier="https://my-sm-broker.com:8445/agent/validate-
     authentication-token"
     ```
3. Speichern und schließen Sie die Datei.

4. Stoppen Sie den Amazon DCV-Server und starten Sie ihn neu. Weitere Informationen finden Sie unter Stoppen des Amazon DCV-Servers und Starten des Amazon DCV-Servers im Amazon DCV-Administratorhandbuch.

Schritt 5: Überprüfen Sie die Installationen

Nachdem Sie den Agenten und den Broker eingerichtet und beide auf dem Amazon DCV-Server konfiguriert haben, müssen Sie überprüfen, ob die Installationen ordnungsgemäß funktionieren.

Themen

Überprüfen Sie den Agenten

Nachdem Sie den Broker und den Agenten installiert haben, stellen Sie sicher, dass der Agent läuft und eine Verbindung zum Broker herstellen kann.

Agentenhosts für Linux und macOS

Der auszuführende Befehl hängt von der Version ab.

- Seit Version 2022.0

  Führen Sie auf dem Agent-Host den folgenden Befehl aus:

  ```
  $ grep 'sessionsUpdateResponse' /var/log/dcv-session-manager-agent/agent.log | tail
   -1 | grep -o success
  ```

- Versionen vor 2022.0

  Führen Sie auf dem Agent-Host den folgenden Befehl aus und geben Sie das aktuelle Jahr, den aktuellen Monat und den aktuellen Tag an.

  ```
  $ grep 'sessionsUpdateResponse' /var/log/dcv-session-manager-agent/
  agent.log.yyyy-mm-dd | tail -1 | grep -o success
  ```

  Beispiel

```
$ grep 'sessionsUpdateResponse' /var/log/dcv-session-manager-agent/
agent.log.2020-11-19 | tail -1 | grep -o success
```

Wenn der Agent läuft und eine Verbindung zum Broker herstellen kann, sollte der Befehl zurückkehren`success`.

Wenn der Befehl eine andere Ausgabe zurückgibt, finden Sie weitere Informationen in der Agent-Protokolldatei. Die Protokolldateien befinden sich hier: `/var/log/dcv-session-manager-agent/`.

Windows-Agent-Host

Öffnen Sie die Agent-Protokolldatei, die sich in befindet`C:\ProgramData\NICE\DCVSessionManagerAgent\log`.

Wenn die Protokolldatei eine Zeile enthält, die der folgenden ähnelt, läuft der Agent und kann eine Verbindung zum Broker herstellen.

```
2020-11-02 12:38:03,996919  INFO ThreadId(05) dcvsessionmanageragent::agent:Processing
 broker message "{\n  \"sessionsUpdateResponse\" : {\n    \"requestId\" :
 \"69c24a3f5f6d4f6f83ffbb9f7dc6a3f4\",\n    \"result\" : {\n      \"success\" : true\n
   }\n  }\n}"
```

Wenn Ihre Protokolldatei keine ähnliche Zeile enthält, überprüfen Sie die Protokolldatei auf Fehler.

Überprüfen Sie den Broker

Nachdem Sie den Broker und den Agenten installiert haben, stellen Sie sicher, dass Ihr Broker läuft und dass er von Ihren Benutzern und Frontend-Anwendungen aus erreichbar ist.

Führen Sie von einem Computer aus, der in der Lage sein sollte, den Broker zu erreichen, den folgenden Befehl aus:

```
$ curl -X GET https://broker_host_ip:port/sessionConnectionData/aSession/aOwner --
insecure
```

Wenn die Überprüfung erfolgreich ist, gibt der Broker Folgendes zurück:

```
{
```

```
    "error": "No authorization header"
}
```

Konfiguration von Amazon DCV Session Manager

Um eine reibungslose und sichere Benutzererfahrung zu gewährleisten, ist es wichtig, den Session Manager entsprechend den Bedürfnissen und Anforderungen Ihres Unternehmens richtig zu konfigurieren. Dieser Abschnitt führt Sie durch die wichtigsten Schritte zur Einrichtung und Konfiguration des Session Managers, einschließlich der Verwaltung des Benutzerzugriffs, der Konfiguration der Netzwerkeinstellungen und der Anpassung der Sitzungseinstellungen.

Themen

Sitzungsmanager skalieren

Um eine hohe Verfügbarkeit zu gewährleisten und die Leistung zu verbessern, können Sie Session Manager so konfigurieren, dass mehrere Agenten und Broker verwendet werden. Wenn Sie beabsichtigen, mehrere Agents und Brokers zu verwenden, empfehlen wir, nur einen Agent- und Broker-Host zu installieren und zu konfigurieren, Amazon Machines Images (AMI) von diesen Hosts zu erstellen und dann die verbleibenden Hosts von zu starten AMIs.

Standardmäßig unterstützt Session Manager die Verwendung mehrerer Agents ohne zusätzliche Konfiguration. Wenn Sie jedoch beabsichtigen, mehrere Broker zu verwenden, müssen Sie einen Load Balancer verwenden, um den Verkehr zwischen dem Frontend-Client und den Brokern sowie zwischen den Brokern und den Agenten auszugleichen. Die Einrichtung und Konfiguration des Load Balancers gehört vollständig Ihnen und wird von Ihnen verwaltet.

Im folgenden Abschnitt wird erklärt, wie Sie Session Manager für die Verwendung mehrerer Hosts mit einem Application Load Balancer konfigurieren.

Schritte

Schritt 1: Erstellen eines Instance-Profils

Sie müssen den Broker- und Agent-Hosts ein Instance-Profil anhängen, das ihnen die Erlaubnis erteilt, Elastic Load Balancing zu verwenden APIs. Weitere Informationen finden Sie unter IAM-Rollen für Amazon EC2 im Amazon-EC2-Benutzerhandbuch.

So erstellen Sie ein Instance-Profil

1. Erstellen Sie eine AWS Identity and Access Management (IAM-) Rolle, die die im Instance-Profil zu verwendenden Berechtigungen definiert. Verwenden Sie die folgende Vertrauensrichtlinie:

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "Service": "ec2.amazonaws.com"
         },
         "Action": "sts:AssumeRole"
       }
     ]
   }
   ```

   Fügen Sie dann die folgende Richtlinie bei:

   JSON

   ```
   {
   ```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeInstances"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "elasticloadbalancing:DescribeTargetHealth"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Weitere Informationen finden Sie unter Erstellen einer IAM-Rolle im IAM-Benutzerhandbuch.

2. Erstellen Sie ein neues Instanzprofil. Weitere Informationen finden Sie unter create-instance-profile in der Referenz zum AWS CLI -Befehl.
3. Fügen Sie dem Instance-Profil die IAM-Rolle hinzu. Weitere Informationen finden Sie unter add-role-to-instance-profile in der AWS CLI Befehlsreferenz.
4. Hängen Sie das Instanzprofil an die Broker-Hosts an. Weitere Informationen finden Sie unter Anhängen einer IAM-Rolle an eine Instance im Amazon EC2 EC2-Benutzerhandbuch.

Schritt 2: Bereiten Sie das SSL-Zertifikat für den Load Balancer vor

Wenn Sie HTTPS für Ihre Load Balancer-Listener verwenden, müssen Sie ein SSL-Zertifikat auf dem Load Balancer bereitstellen. Der Load Balancer verwendet dieses Zertifikat, um die Verbindung zu beenden und Anfragen von Clients zu entschlüsseln, bevor er sie an die Ziele sendet.

Um das SSL-Zertifikat vorzubereiten

1. Erstellen Sie eine private Zertifizierungsstelle (CA) AWS Certificate Manager Private Certificate Authority (ACM PCA). Weitere Informationen finden Sie unter Verfahren zum Erstellen einer CA im AWS Certificate Manager Private Certificate Authority User Guide.

2. Installieren Sie die CA. Weitere Informationen finden Sie unter Installation eines Root-CA-Zertifikats im AWS Certificate Manager Private Certificate Authority User Guide.
3. Fordern Sie ein neues privates Zertifikat an, das von der CA signiert wurde. Verwenden Sie für den Domainnamen die Region, in der Sie den Load Balancer erstellen möchten, `*.region.elb.amazonaws.com` und geben Sie sie an. Weitere Informationen finden Sie unter Anfordern eines privaten Zertifikats im AWS Certificate Manager Private Certificate Authority User Guide.

Schritt 3: Erstellen Sie den Load Balancer für die Broker-Applikation

Erstellen Sie einen Application Load Balancer, um den Datenverkehr zwischen Ihren Front-End-Clients und den Brokern auszugleichen.

So erstellen Sie den Load Balancer

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/.

   Wählen Sie im Navigationsbereich Load Balancers und dann Create Load Balancer aus. Wählen Sie als Load Balancer-Typ Application Load Balancer aus.
2. Führen Sie für Step 1: Configure Load Balancer (Schritt 1; Konfigurieren von Load Balancer) die folgenden Schritte aus:
   a. Geben Sie unter Name einen aussagekräftigen Namen für den Load Balancer ein.
   b. Wählen Sie für Schema die Option Internet-facing aus.
   c. Wählen Sie für Load Balancer Protocol die Option HTTPS aus, und geben Sie für Load Balancer Port ein. `8443`
   d. Wählen Sie für VPC die zu verwendende VPC und dann alle Subnetze in dieser VPC aus.
   e. Wählen Sie Weiter aus.
3. Gehen Sie für Schritt 2: Sicherheitseinstellungen konfigurieren wie folgt vor:
   a. Wählen Sie als Zertifikatstyp die Option Zertifikat aus ACM auswählen aus.
   b. Wählen Sie unter Zertifikatsname das private Zertifikat aus, das Sie zuvor angefordert haben.
   c. Wählen Sie Weiter aus.

4. Für Schritt 3: Sicherheitsgruppen konfigurieren, eine neue Sicherheitsgruppe erstellen oder eine vorhandene Sicherheitsgruppe auswählen, die eingehenden und ausgehenden Datenverkehr zwischen Ihrem Frontend-Client und den Brokern über HTTPS und Port 8443 zulässt.

   Wählen Sie Weiter aus.

5. Gehen Sie für Schritt 4: Routing konfigurieren wie folgt vor:

   a. Wählen Sie für Zielgruppe die Option Neue Zielgruppe aus.

   b. Geben Sie unter Name einen Namen für die Zielgruppe ein.

   c. Wählen Sie als Zieltyp die Option Instanz aus.

   d. Wählen Sie als Protokoll die Option HTTPS aus. Geben Sie im Feld Port `8443` ein. Wählen Sie für Protokollversion die Option HTTP1.

   e. Wählen Sie für das Health Check-Protokoll die Option HTTPS aus, und geben Sie als Pfad ein`/health`.

   f. Wählen Sie Weiter aus.

6. Wählen Sie für Schritt 5: Ziele registrieren die Option Weiter aus.

7. Wählen Sie Erstellen aus.

Schritt 4: Starten Sie die Broker

Erstellen Sie einen ersten Broker und konfigurieren Sie ihn für die Verwendung des Load Balancers, erstellen Sie ein AMI vom Broker und verwenden Sie dann das AMI, um die verbleibenden Broker zu starten. Dadurch wird sichergestellt, dass alle Broker so konfiguriert sind, dass sie dieselbe CA und dieselbe Load Balancer-Konfiguration verwenden.

Um die Brokers zu starten

1. Starten und konfigurieren Sie den ersten Broker-Host. Weitere Informationen zur Installation und Konfiguration des Brokers finden Sie unterSchritt 2: Den Amazon DCV Session Manager-Broker einrichten.

   > **Note**
   >
   > Das selbstsignierte Zertifikat des Brokers ist nicht erforderlich, da wir einen Application Load Balancer verwenden.

2. Connect zum Broker her, öffnen Sie ihn `/etc/dcv-session-manager-broker/session-manager-broker.properties` mit Ihrem bevorzugten Texteditor und gehen Sie wie folgt vor:

   a. Kommentieren Sie den `broker-to-broker-discovery-addresses` Parameter aus, indem Sie am Anfang der Zeile einen Hash (#) platzieren.

   b. Geben Sie für die Region ein`broker-to-broker-discovery-aws-region`, in der Sie den Application Load Balancer erstellt haben.

   c. Geben Sie für den ARN der Zielgruppe ein`broker-to-broker-discovery-aws-alb-target-group-arn`, die dem Broker Load Balancer zugeordnet ist.

   d. Speichern und schließen Sie die Datei.

3. Stoppen Sie die Broker-Instanz.
4. Erstellen Sie ein AMI aus der gestoppten Broker-Instance. Weitere Informationen finden Sie unter Erstellen eines Linux-AMI aus einer Instance im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances.
5. Verwenden Sie das AMI, um die verbleibenden Broker zu starten.
6. Weisen Sie das Instance-Profil, das Sie erstellt haben, allen Broker-Instances zu.
7. Weisen Sie eine Sicherheitsgruppe zu, mit der Broker to Broker und Broker den Load Balancer-Netzwerkverkehr für alle Broker-Instanzen ausgleichen können. Weitere Informationen zu Netzwerkports finden Sie unter Broker-Konfigurationsdatei.
8. Registrieren Sie alle Broker-Instanzen als Ziele für den Broker Load Balancer. Weitere Informationen finden Sie unter Registrieren von Zielen bei Ihrer Zielgruppe im Benutzerhandbuch für Application Load Balancers.

Schritt 5: Erstellen Sie den Agent Application Load Balancer

Erstellen Sie einen Application Load Balancer, um das Gleichgewicht zwischen Agenten und Brokern zu verteilen.

So erstellen Sie den Load Balancer

1. Öffnen Sie die Amazon-EC2-Konsole unter https://console.aws.amazon.com/ec2/.

   Wählen Sie im Navigationsbereich Load Balancers und dann Create Load Balancer aus. Wählen Sie als Load Balancer-Typ Application Load Balancer aus.

2. Führen Sie für Step 1: Configure Load Balancer (Schritt 1; Konfigurieren von Load Balancer) die folgenden Schritte aus:

a. Geben Sie unter Name einen aussagekräftigen Namen für den Load Balancer ein.

b. Wählen Sie für Schema die Option Internet-facing aus.

c. Wählen Sie für Load Balancer Protocol die Option HTTPS aus, und geben Sie für Load Balancer Port ein. `8445`

d. Wählen Sie für VPC die zu verwendende VPC und dann alle Subnetze in dieser VPC aus.

e. Wählen Sie Weiter aus.

3. Gehen Sie für Schritt 2: Sicherheitseinstellungen konfigurieren wie folgt vor:

a. Wählen Sie als Zertifikatstyp die Option Zertifikat aus ACM auswählen aus.

b. Wählen Sie unter Zertifikatsname das private Zertifikat aus, das Sie zuvor angefordert haben.

c. Wählen Sie Weiter aus.

4. Für Schritt 3: Sicherheitsgruppen konfigurieren, eine neue Sicherheitsgruppe erstellen oder eine vorhandene Sicherheitsgruppe auswählen, die eingehenden und ausgehenden Datenverkehr zwischen den Agents und Brokern über HTTPS und Port 8445 zulässt.

   Wählen Sie Weiter aus.

5. Gehen Sie für Schritt 4: Routing konfigurieren wie folgt vor:

a. Wählen Sie für Zielgruppe die Option Neue Zielgruppe aus.

b. Geben Sie unter Name einen Namen für die Zielgruppe ein.

c. Wählen Sie als Zieltyp die Option Instanz aus.

d. Wählen Sie als Protokoll die Option HTTPS aus. Geben Sie im Feld Port `8445` ein. Wählen Sie für Protokollversion die Option HTTP1.

e. Wählen Sie für das Health Check-Protokoll die Option HTTPS aus, und geben Sie als Pfad ein`/health`.

f. Wählen Sie Weiter aus.

6. Wählen Sie für Schritt 5: Ziele registrieren alle Broker-Instances aus und wählen Sie Zu registrierten hinzufügen aus. Wählen Sie Weiter: Prüfen aus.

7. Wählen Sie Erstellen aus.

Schritt 6: Starten Sie die Agents

Erstellen Sie einen ersten Agenten und konfigurieren Sie ihn für die Verwendung des Load Balancers, erstellen Sie ein AMI aus dem Agenten und verwenden Sie dann das AMI, um die verbleibenden Agents zu starten. Dadurch wird sichergestellt, dass alle Agents für die Verwendung derselben Load Balancer-Konfiguration konfiguriert sind.

Um die Agents zu starten

1. Bereiten Sie den Amazon DCV-Server vor. Weitere Informationen finden Sie unter Schritt 1: Bereiten Sie die Amazon DCV-Server vor.
2. Platzieren Sie eine Kopie des öffentlichen CA-Schlüssels, der in Schritt 2: Bereiten Sie das SSL-Zertifikat für den Load Balancer vor erstellt wurde. Wählen oder erstellen Sie ein Verzeichnis, das für jeden Benutzer lesbar ist. Die Datei mit dem öffentlichen Schlüssel der CA muss auch für jeden Benutzer lesbar sein.
3. Installieren und konfigurieren Sie den Agenten. Weitere Informationen zur Installation und Konfiguration des Agenten finden Sie unterSchritt 3: Den Amazon DCV Session Manager-Agenten einrichten.

   > **Important**
   >
   > Gehen Sie beim Ändern der Agenten-Konfigurationsdatei wie folgt vor:
   >
   > - Geben Sie für den `broker_host` Parameter den DNS des Agenten-Loadbalancers ein
   > - Geben Sie für den `ca_file` Parameter den Pfad zur Datei mit dem öffentlichen Schlüssel der CA ein, die im vorherigen Schritt erstellt wurde

4. Konfigurieren Sie den Amazon DCV-Server so, dass er den Broker als Authentifizierungsserver verwendet. Weitere Informationen finden Sie unter Schritt 4: Konfigurieren Sie den Amazon DCV-Server so, dass er den Broker als Authentifizierungsserver verwendet.

   > **Important**
   >
   > Gehen Sie beim Ändern der Amazon DCV-Serverkonfigurationsdatei wie folgt vor:
   >
   > - Geben Sie für den `ca-file` Parameter denselben Pfad zur öffentlichen CA-Schlüsseldatei ein, der im vorherigen Schritt verwendet wurde

- Verwenden Sie für den `auth-token-verifier` Parameter den DNS des Agent-Loadbalancers für *`broker_ip_or_dns`*

5. Stoppen Sie die Agent-Instanz.
6. Erstellen Sie ein AMI aus der gestoppten Agent-Instanz. Weitere Informationen finden Sie unter Erstellen eines Linux-AMI aus einer Instance im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances.
7. Verwenden Sie das AMI, um die verbleibenden Agents zu starten und ihnen das von Ihnen erstellte Instance-Profil zuzuweisen.
8. Weisen Sie eine Sicherheitsgruppe zu, die es dem Agenten ermöglicht, den Netzwerkverkehr über den Load Balancer auf alle Agent-Instances auszudehnen. Weitere Informationen zu Netzwerkports finden Sie in der Agent-Konfigurationsdatei.

Verwenden von Tags als Ziel für Amazon DCV-Server

Sie können Session Manager-Agenten benutzerdefinierte Tags zuweisen, um sie und die Amazon DCV-Server, mit denen sie verknüpft sind, zu identifizieren und zu kategorisieren. Wenn Sie eine neue Amazon DCV-Sitzung erstellen, können Sie eine Gruppe von Amazon DCV-Servern anhand der Tags ansprechen, die ihren jeweiligen Agenten zugewiesen sind. Weitere Informationen zum Targeting von Amazon DCV-Servern auf der Grundlage von Agent-Tags finden Sie CreateSessionRequests im Session Manager Developer Guide.

Ein Tag besteht aus einem Tag-Schlüssel- und Wertepaar, und Sie können jedes Informationspaar verwenden, das für Ihren Anwendungsfall oder Ihre Umgebung sinnvoll ist. Sie können festlegen, ob Agenten auf der Grundlage der Hardwarekonfiguration ihres Hosts markiert werden sollen. Sie können beispielsweise alle Agents mit Hosts, die über 4 GB Arbeitsspeicher verfügen, mit taggen`ram=4GB`. Oder Sie können Agenten je nach Zweck taggen. Sie können beispielsweise alle Agents, die auf Produktions-Hosts laufen, mit taggen`purpose=production`.

Um einem Agenten Tags zuzuweisen

1. Erstellen Sie mit Ihrem bevorzugten Texteditor eine neue Datei und geben Sie ihr beispielsweise `agent_tags.toml` einen aussagekräftigen Namen. Der Dateityp und der Dateiinhalt müssen im TOML-Dateiformat angegeben werden. `.toml`
2. Fügen Sie in der Datei jedes neue Tag-Schlüssel-Wert-Paar in einer neuen Zeile unter Verwendung des folgenden `key=value` Formats hinzu. Zum Beispiel:

```
tag1="abc"
tag2="xyz"
```

3. Öffnen Sie die Agent-Konfigurationsdatei (`/etc/dcv-session-manager-agent/agent.conf`für Linux und macOS oder `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` für Windows). Für`tags_folder`, und geben Sie den Pfad zu dem Verzeichnis an, in dem sich die Tag-Datei befindet.

   Wenn das Verzeichnis mehrere Tag-Dateien enthält, wenden alle in den Dateien definierten Tags den Agenten an. Die Dateien werden in alphabetischer Reihenfolge gelesen. Wenn mehrere Dateien ein Tag mit demselben Schlüssel enthalten, wird der Wert mit dem Wert aus der zuletzt gelesenen Datei überschrieben.

4. Speichern und schließen Sie die Datei.
5. Beenden Sie den Agenten und starten Sie ihn neu.

   - Windows

     ```
     C:\> sc stop DcvSessionManagerAgentService
     ```

     ```
     C:\> sc start DcvSessionManagerAgentService
     ```

   - Linux

     ```
     $ sudo systemctl stop dcv-session-manager-agent
     ```

     ```
     $ sudo systemctl start dcv-session-manager-agent
     ```

   - macOS

     ```
     $ sudo launchctl unload /Library/LaunchDaemons/com.amazon.dcv.session-
     manager.agent.plist
     ```

     ```
     $ sudo launchctl load /Library/LaunchDaemons/com.amazon.dcv.session-
     manager.agent.plist
     ```

Konfiguration eines externen Autorisierungsservers

Der Autorisierungsserver ist der Server, der für die Authentifizierung und Autorisierung des Clients SDKs und der Agenten verantwortlich ist.

Standardmäßig verwendet Session Manager den Broker als Autorisierungsserver, um OAuth 2.0-Zugriffstoken für Clients SDKs und Softwareanweisungen für Agenten zu generieren. Wenn Sie den Broker als Autorisierungsserver verwenden, ist keine zusätzliche Konfiguration erforderlich.

Sie können Session Manager so konfigurieren, dass Amazon Cognito anstelle des Brokers als externen Autorisierungsserver verwendet wird. Weitere Informationen zu Amazon Cognito finden Sie im Amazon Cognito Developer Guide.

So verwenden Sie Amazon Cognito als Autorisierungsserver

1. Erstellen Sie einen neuen Amazon Cognito Cognito-Benutzerpool. Weitere Informationen zu Benutzerpools finden Sie unter Funktionen von Amazon Cognito im Amazon Cognito Developer Guide.

   Verwenden Sie den create-user-poolBefehl und geben Sie einen Poolnamen und die Region an, in der er erstellt werden soll.

   In diesem Beispiel geben wir dem Pool einen Namen `dcv-session-manager-client-app` und erstellen ihn in`us-east-1`.

   ```
   $ aws cognito-idp create-user-pool --pool-name dcv-session-manager-client-app --
   region us-east-1
   ```

   Beispielausgabe

   ```
   {
       "UserPoolClient": {
           "UserPoolId": "us-east-1_QLEXAMPLE",
           "ClientName": "dcv-session-manager-client-app",
           "ClientId": "15hhd8jij74hf32f24uEXAMPLE",
           "LastModifiedDate": 1602510048.054,
           "CreationDate": 1602510048.054,
           "RefreshTokenValidity": 30,
           "AllowedOAuthFlowsUserPoolClient": false
       }
   ```

```
}
```

Notieren Sie sich das`userPoolId`, Sie werden es im nächsten Schritt benötigen.

2. Erstellen Sie eine neue Domain für Ihren Benutzerpool. Verwenden Sie den create-user-pool-domainBefehl und geben Sie einen Domänennamen und den Namen `userPoolId` des Benutzerpools an, den Sie im vorherigen Schritt erstellt haben.

   In diesem Beispiel lautet der Domainname `mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE` und wir erstellen ihn in`us-east-1`.

   ```
   $ aws cognito-idp create-user-pool-domain --domain mydomain-544fa30f-
   c0e5-4a02-8d2a-a3761EXAMPLE --user-pool-id us-east-1_QLEXAMPLE --region us-east-1
   ```

   Beispielausgabe

   ```
   {
       "DomainDescription": {
           "UserPoolId": "us-east-1_QLEXAMPLE",
           "AWSAccountId": "123456789012",
           "Domain": "mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE",
           "S3Bucket": "aws-cognito-prod-pdx-assets",
           "CloudFrontDistribution": "dpp0gtexample.cloudfront.net",
           "Version": "20201012133715",
           "Status": "ACTIVE",
           "CustomDomainConfig": {}
       }
   }
   ```

   Das Format der Benutzerpool-Domäne lautet wie folgt:`https://domain_name.auth.region.amazoncognito.com`. In diesem Beispiel lautet die Benutzerpool-Domäne`https://mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE.auth.us-east-1.amazoncognito.com`.

3. Erstellen Sie einen Benutzerpool-Client. Verwenden Sie den create-user-pool-clientBefehl und geben Sie den `userPoolId` Benutzerpool an, den Sie erstellt haben, einen Namen für den Client und die Region, in der er erstellt werden soll. Fügen Sie außerdem die `--generate-secret` Option hinzu, mit der Sie angeben können, dass Sie ein Geheimnis für den Benutzerpool-Client generieren möchten, der gerade erstellt wird.

In diesem Fall lautet der Kundenname `dcv-session-manager-client-app` und wir erstellen ihn in der `us-east-1` Region.

```
$ aws cognito-idp create-user-pool-client --user-pool-id us-east-1_QLEXAMPLE --
client-name dcv-session-manager-client-app --generate-secret --region us-east-1
```

Beispielausgabe

```
{
    "UserPoolClient": {
        "UserPoolId": "us-east-1_QLEXAMPLE",
        "ClientName": "dcv-session-manager-client-app",
        "ClientId": "2l9273hp6k2ut5cugg9EXAMPLE",
        "ClientSecret": "1vp5e8nec7cbf4m9me55mbmht91u61hlh0a78rq1qki1lEXAMPLE",
        "LastModifiedDate": 1602510291.498,
        "CreationDate": 1602510291.498,
        "RefreshTokenValidity": 30,
        "AllowedOAuthFlowsUserPoolClient": false
    }
}
```

> **Note**
>
> Notieren Sie sich das `ClientId` und`ClientSecret`. Sie müssen diese Informationen den Entwicklern zur Verfügung stellen, wenn sie Zugriffstoken für die API-Anfragen anfordern.

4. Erstellen Sie einen neuen OAuth2 2.0-Ressourcenserver für den Benutzerpool. Ein Ressourcenserver ist ein Server für zugriffsgeschützte Ressourcen. Er verarbeitet authentifizierte Anfragen nach Zugriffstoken.

   Verwenden Sie den create-resource-serverBefehl und geben Sie den `userPoolId` Benutzerpool, eine eindeutige Kennung und einen Namen für den Ressourcenserver, den Bereich und die Region an, in der er erstellt werden soll.

   In diesem Beispiel verwenden wir `dcv-session-manager` als Bezeichner und Namen sowie `sm_scope` als Bereichsnamen und Beschreibung.

```
$ aws cognito-idp create-resource-server --user-pool-id us-east-1_QLEXAMPLE
 --identifier dcv-session-manager --name dcv-session-manager --scopes
 ScopeName=sm_scope,ScopeDescription=sm_scope --region us-east-1
```

Beispielausgabe

```
{
    "ResourceServer": {
        "UserPoolId": "us-east-1_QLEXAMPLE",
        "Identifier": "dcv-session-manager",
        "Name": "dcv-session-manager",
        "Scopes": [
        {
            "ScopeName": "sm_scope",
            "ScopeDescription": "sm_scope"
        }]
    }
}
```

5. Aktualisieren Sie den Benutzerpool-Client.

   Verwenden Sie den update-user-pool-client-Befehl. Geben Sie `userPoolId` den Benutzerpool, den `ClientId` des Benutzerpool-Clients und die Region an. Geben Sie für `client_credentials` an`--allowed-o-auth-flows`, dass der Client mithilfe einer Kombination aus einer Client-ID und einem geheimen Client-Schlüssel Zugriffstoken vom Token-Endpunkt abrufen soll. Geben Sie für `--allowed-o-auth-scopes` die Ressourcenserver-ID und den Bereichsnamen wie folgt an:*resource_server_identifier*/*scope_name*. Fügen Sie das ein, `--allowed-o-auth-flows-user-pool-client` um anzugeben, dass der Client bei der Interaktion mit Cognito-Benutzerpools OAuth das Protokoll befolgen darf.

   ```
   $ aws cognito-idp update-user-pool-client --user-pool-id us-east-1_QLEXAMPLE --
   client-id 2l9273hp6k2ut5cugg9EXAMPLE --allowed-o-auth-flows client_credentials --
   allowed-o-auth-scopes dcv-session-manager/sm_scope --allowed-o-auth-flows-user-
   pool-client --region us-east-1
   ```

   Beispielausgabe

   ```
   {
       "UserPoolClient": {
   ```

```
        "UserPoolId": "us-east-1_QLEXAMPLE",
        "ClientName": "dcv-session-manager-client-app",
        "ClientId": "2l9273hp6k2ut5cugg9EXAMPLE",
        "ClientSecret": "1vp5e8nec7cbf4m9me55mbmht91u61hlh0a78rq1qki1lEXAMPLE",
        "LastModifiedDate": 1602512103.099,
        "CreationDate": 1602510291.498,
        "RefreshTokenValidity": 30,
        "AllowedOAuthFlows": [
            "client_credentials"
        ],
        "AllowedOAuthScopes": [
            "dcv-session-manager/sm_scope"
        ],
        "AllowedOAuthFlowsUserPoolClient": true
    }
}
```

> **Note**
>
> Der Benutzerpool ist jetzt bereit, Zugriffstoken bereitzustellen und zu authentifizieren. In diesem Beispiel lautet `https://cognito-idp.us-east-1.amazonaws.com/us-east-1_QLEXAMPLE/.well-known/jwks.json` die URL für den Autorisierungsserver.

6. Testen Sie die Konfiguration.

```
$ curl -H "Authorization: Basic `echo -
n 2l9273hp6k2ut5cugg9EXAMPLE:1vp5e8nec7cbf4m9me55mbmht91u61hlh0a78rq1qki1lEXAMPLE
 | base64`" -H "Content-Type: application/x-www-form-urlencoded" -X
 POST "https://mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE.auth.us-
east-1.amazoncognito.com/oauth2/token?grant_type=client_credentials&scope=dcv-
session-manager/sm_scope"
```

Beispielausgabe

```
{
"access_token":"eyJraWQiOiJGQ0VaRFpJUUptT3NSaW41MmtqaDdEbTZYb0RnSTQ5b2VUT0cxUUI1Q2VJPSIsImF
Zkfi0HIDsd6audjTXKzHlZGScr6ROdZtId5dThkpEZiSx0YwiiWe9crAlqoazlDcCsUJHIXDtgKW64pSj3-
uQQGg1Jv_tyVjhrA4JbD0k67WS2V9NW-
uZ7t4zwwaUmOi3KzpBMi54fpVgPaewiVlUm_aS4LUFcWT6hVJjiZF7om7984qb2gOa14iZxpXPBJTZX_gtG9EtvnS9u
"expires_in":3600,
```

```
"token_type":"Bearer"
}
```

7. Registrieren Sie den externen Autorisierungsserver für die Verwendung mit dem Broker, indem Sie den register-auth-server Befehl verwenden.

```
$ sudo -u root dcv-session-manager-broker register-auth-server --url  https://
cognito-idp.us-east-1.amazonaws.com/us-east-1_QLEXAMPLE/.well-known/jwks.json
```

Entwickler können jetzt den Server verwenden, um Zugriffstoken anzufordern. Wenn Sie Zugriffstoken anfordern, geben Sie die Client-ID, den geheimen Client-Schlüssel und die hier generierte Server-URL an. Weitere Informationen zum Anfordern von Zugriffstoken finden Sie unter Erstellen, Zugriffstoken erstellen und API-Anfrage stellen im Amazon DCV Session Manager Developer Guide.

Konfiguration der Broker-Persistenz

Session Manager-Broker unterstützen die Integration mit externen Datenbanken. Die externe Datenbank ermöglicht es Session Manager, Statusdaten und Schlüssel beizubehalten, sodass sie anschließend verfügbar sind. Tatsächlich sind die Broker-Daten über den Cluster verteilt, sodass dieser anfällig für Datenverluste ist, wenn ein Host neu gestartet werden muss oder ein Cluster beendet wird. Wenn diese Funktion aktiviert ist, können Sie Brokerknoten hinzufügen und entfernen. Außerdem können Sie einen Cluster stoppen und neu starten, ohne Schlüssel neu generieren zu müssen oder Informationen darüber zu verlieren, welcher Amazon DCV-Server geöffnet oder geschlossen ist.

Die folgenden Arten von Informationen können so eingestellt werden, dass sie dauerhaft gespeichert werden:

- Schlüssel zum Einrichten von Sitzungen zum Herstellen einer Verbindung mit Clients
- Daten zu Sitzungen während des Fluges
- Amazon DCV-Serverstatus

Amazon DCV Session Manager unterstützt DynamoDB-, MariaDB- und MySQL-Datenbanken. Sie müssen eine dieser Datenbanken einrichten und verwalten, um diese Funktion nutzen zu können. Wenn Ihre Broker-Computer bei Amazon gehostet werden EC2, empfehlen wir, DynamoDB als externe Datenbank zu verwenden, da hierfür keine zusätzliche Einrichtung erforderlich ist.

> **Note**
>
> Beim Betrieb einer externen Datenbank können zusätzliche Kosten anfallen. Informationen zu den Preisen von DynamoDB finden Sie unter Preise für bereitgestellte Kapazität.

Den Broker so konfigurieren, dass er auf DynamoDB persistiert

Konfigurieren Sie die Broker so, dass sie mit dem Speichern ihrer Daten auf DynamoDB beginnen:

1. Öffnen Sie `/etc/dcv-session-manager-broker/session-manager-broker.properties` mit Ihrem bevorzugten Texteditor und nehmen Sie die folgenden Änderungen vor:

   - Legen Sie `enable-persistence = true` fest.
   - Legen Sie `persistence-db = dynamodb` fest.
   - `dynamodb-region`Geben Sie für die &aws; -Region an, in der Sie die Tabellen mit den Brokerdaten speichern möchten. Eine Liste der unterstützten Regionen finden Sie unter DynamoDB-Dienstendpunkte.
   - `dynamodb-table-rcu`Geben Sie die Anzahl der Read Capacity Units (RCU) an, die jede Tabelle unterstützt. Weitere Informationen zu RCU finden Sie unter Bereitgestellte Kapazität von DynamoDB.
   - `dynamodb-table-wcu`Geben Sie die Anzahl der Schreibkapazitätseinheiten (WCU) an, die jede Tabelle unterstützt. Weitere Informationen zu WCU finden Sie unter Bereitgestellte Kapazität von DynamoDB.
   - dynamodb-table-name-prefixGeben Sie für das Präfix an, das jeder DynamoDB-Tabelle hinzugefügt wird (nützlich, um mehrere Broker-Cluster zu unterscheiden, die dasselbe Konto verwenden). Nur alphanumerische Zeichen, Punkt, Bindestrich und Unterstrich sind zulässig.

2. Stoppen Sie alle Broker im Cluster. Führen Sie für jeden Broker den folgenden Befehl aus:

   ```
   sudo systemctl stop dcv-session-manager-broker
   ```

3. Stellen Sie sicher, dass alle Broker im Cluster gestoppt sind, und starten Sie sie dann alle neu. Starten Sie jeden Broker, indem Sie den folgenden Befehl ausführen:

   ```
   sudo systemctl start dcv-session-manager-broker
   ```

Der Broker-Host muss berechtigt sein, DynamoDB APIs aufzurufen. Auf EC2 Amazon-Instances werden die Anmeldeinformationen automatisch mithilfe des EC2 Amazon-Metadatendienstes abgerufen. Wenn Sie andere Anmeldeinformationen angeben müssen, können Sie diese mithilfe einer der unterstützten Techniken zum Abrufen von Anmeldeinformationen (z. B. Java-Systemeigenschaften oder Umgebungsvariablen) festlegen. Weitere Informationen finden Sie unter &aws; -Anmeldeinformationen bereitstellen und abrufen.

Konfigurieren Sie den Broker so, dass er auf MariaDB/MySQL persistiert

> **Note**
>
> Die `/etc/dcv-session-manager-broker/session-manager-broker.properties` Datei enthält sensible Daten. Standardmäßig ist der Schreibzugriff auf Root und der Lesezugriff auf Root und den Benutzer beschränkt, der den Broker ausführt. Standardmäßig ist dies der `dcvsmbroker` Benutzer. Der Broker überprüft beim Start, ob die Datei über die erwarteten Berechtigungen verfügt.

Konfigurieren Sie die Broker so, dass sie beginnen, ihre Daten auf MariaDB/MySQL MySQL:

1. Öffnen Sie `/etc/dcv-session-manager-broker/session-manager-broker.properties` mit Ihrem bevorzugten Texteditor und nehmen Sie die folgenden Änderungen vor:

   - Legen Sie `enable-persistence = true` fest.
   - Legen Sie `persistence-db = mysql` fest.
   - Legen Sie `jdbc-connection-url = jdbc:mysql://<db_endpoint>:<db_port>/<db_name>?createDatabaseIfNotExist=true` fest.

     In dieser Konfiguration <db_endpoint>ist der Datenbankendpunkt, <db_port>der Datenbankport und <db_name>der Datenbankname.
   - `jdbc-user`Geben Sie für den Namen des Benutzers an, der Zugriff auf die Datenbank hat.
   - `jdbc-password`Geben Sie für das Passwort des Benutzers an, der Zugriff auf die Datenbank hat.
2. Stoppen Sie alle Broker im Cluster. Führen Sie für jeden Broker den folgenden Befehl aus:

```
sudo systemctl stop dcv-session-manager-broker
```

3. Stellen Sie sicher, dass alle Broker im Cluster gestoppt sind, und starten Sie dann alle neu. Führen Sie für jeden Broker den folgenden Befehl aus:

```
sudo systemctl start dcv-session-manager-broker
```

Integration mit dem Amazon DCV Connection Gateway

Amazon DCV Connection Gateway ist ein installierbares Softwarepaket, mit dem Benutzer über einen einzigen Zugriffspunkt zu einem LAN oder einer VPC auf eine Flotte von Amazon DCV-Servern zugreifen können.

Wenn Ihre Infrastruktur Amazon DCV-Server umfasst, auf die über das Amazon DCV Connection Gateway zugegriffen werden kann, können Sie den Session Manager so konfigurieren, dass er das Amazon DCV Connection Gateway integriert. Wenn Sie die im folgenden Abschnitt beschriebenen Schritte befolgen, fungiert der Broker als Session Resolver für das Connection Gateway. Mit anderen Worten, der Broker macht einen zusätzlichen HTTP-Endpunkt verfügbar. Das Connection Gateway sendet API-Aufrufe an den Endpunkt, um die Informationen abzurufen, die für die Weiterleitung von Amazon DCV-Verbindungen an den vom Broker ausgewählten Host erforderlich sind.

Themen

Richten Sie den Session Manager Broker als Session Resolver für das Amazon DCV Connection Gateway ein

Session Manager Broker-Seite

1. Öffnen Sie `/etc/dcv-session-manager-broker/session-manager-broker.properties` die Datei mit Ihrem bevorzugten Texteditor und nehmen Sie die folgenden Änderungen vor:

- Legen Sie `enable-gateway = true` fest.
- Auf `gateway-to-broker-connector-https-port` einen freien TCP-Port eingestellt (Standard ist 8447)
- Auf `gateway-to-broker-connector-bind-host` die IP-Adresse des Hosts eingestellt, an den der Broker für Amazon DCV Connection Gateway-Verbindungen bindet (Standard ist 0.0.0.0)

2. Führen Sie dann die folgenden Befehle aus, um den Broker zu beenden und neu zu starten:

```
sudo systemctl stop dcv-session-manager-broker
```

```
sudo systemctl start dcv-session-manager-broker
```

3. Rufen Sie eine Kopie des selbstsignierten Zertifikats des Brokers ab und platzieren Sie es in Ihrem Benutzerverzeichnis.

```
sudo cp /var/lib/dcvsmbroker/security/dcvsmbroker_ca.pem $HOME
```

Sie benötigen es, wenn Sie das Amazon DCV Connection Gateway im nächsten Schritt installieren.

Seite Amazon DCV Connection Gateway

- Bitte folgen Sie dem Abschnitt in der Amazon DCV Connection Gateway-Dokumentation.

  Da das Amazon DCV Connection Gateway HTTP-API-Aufrufe an den Broker sendet, müssen Sie, wenn der Broker ein selbstsigniertes Zertifikat verwendet, das Broker-Zertifikat auf den Amazon DCV Connection Gateway-Host kopieren (im vorherigen Schritt abgerufen) und den `ca-file` Parameter im `[resolver]` Abschnitt der Amazon DCV Connection Gateway-Konfiguration festlegen.

Optional — Aktivieren Sie die TLS-Client-Authentifizierung

Sobald Sie den vorherigen Schritt abgeschlossen haben, können der Session Manager und das Connection Gateway über einen sicheren Kanal kommunizieren, über den das Connection Gateway die Identität der Session Manager-Broker überprüfen kann. Wenn Sie möchten, dass auch die Session Manager-Broker die Identität des Connection Gateways überprüfen, bevor der sichere Kanal

eingerichtet wird, müssen Sie die TLS-Client-Authentifizierungsfunktion aktivieren, indem Sie die Schritte im nächsten Abschnitt befolgen.

> **Note**
>
> Wenn sich der Session Manager hinter einem Load Balancer befindet, kann die TLS-Client-Authentifizierung nicht für Load Balancer aktiviert werden, die über einen TLS-Verbindungsabbruch verfügen, wie z. B. Application Load Balancers (ALBs) oder Gateway Load Balancers (). GLBs Es können nur Load Balancer ohne TLS-Terminierung unterstützt werden, z. B. Network Load Balancers (). NLBs Wenn Sie ALBs oder verwenden GLBs, können Sie erzwingen, dass nur bestimmte Sicherheitsgruppen Kontakt zu den Load Balancers aufnehmen können, wodurch eine zusätzliche Sicherheitsstufe gewährleistet wird. Weitere Informationen zu Sicherheitsgruppen finden Sie hier: Sicherheitsgruppen für Ihre VPC

Session Manager Broker-Seite

1. Gehen Sie wie folgt vor, um die TLS-Client-Authentifizierung für die Kommunikation zwischen den Session Manager Brokers und dem Amazon DCV Connection Gateway zu aktivieren:
2. Generieren Sie die erforderlichen Schlüssel und Zertifikate, indem Sie Folgendes ausführen: In der Ausgabe des Befehls erfahren Sie, in welchem Ordner die Anmeldeinformationen generiert wurden, und welches Passwort für die Erstellung der TrustStore Datei verwendet wurde.

   ```
   sudo /usr/share/dcv-session-manager-broker/bin/gen-gateway-certificates.sh
   ```

3. Platzieren Sie eine Kopie des privaten Schlüssels und des selbstsignierten Zertifikats von Amazon DCV Connection Gateway in Ihrem Benutzerverzeichnis. Sie benötigen es, wenn Sie im nächsten Schritt die TLS-Client-Authentifizierung im Amazon DCV Connection Gateway aktivieren.

   ```
   sudo cp /etc/dcv-session-manager-broker/resolver-creds/dcv_gateway_key.pem $HOME
   ```

   ```
   sudo cp /etc/dcv-session-manager-broker/resolver-creds/dcv_gateway_cert.pem $HOME
   ```

4. Öffnen Sie dann mit Ihrem bevorzugten Texteditor/etc/dcv-session-manager-broker/session-manager-broker.properties und gehen Sie wie folgt vor:

- Setzen Sie `enable-tls-client-auth-gateway` auf `true`
- Geben Sie `gateway-to-broker-connector-trust-store-file` den Pfad der TrustStore Datei ein, die im vorherigen Schritt erstellt wurde
- Stellen Sie `gateway-to-broker-connector-trust-store-pass` das Passwort ein, das für die Erstellung der TrustStore Datei im vorherigen Schritt verwendet wurde

5. Führen Sie dann den folgenden Befehl aus, um den Broker zu beenden und neu zu starten:

```
sudo systemctl stop dcv-session-manager-broker
```

```
sudo systemctl start dcv-session-manager-broker
```

Seite Amazon DCV Connection Gateway

- Bitte folgen Sie dem Abschnitt in der Amazon DCV Connection Gateway-Dokumentation.
  - verwenden Sie den vollständigen Pfad der Zertifikatsdatei, die Sie im vorherigen Schritt kopiert haben, wenn Sie den `cert-file` Parameter im Abschnitt festlegen `[resolver]`
  - verwenden Sie den vollständigen Pfad der Schlüsseldatei, die Sie im vorherigen Schritt kopiert haben, wenn Sie den `cert-key-file` Parameter im `[resolver]` Abschnitt festlegen

Amazon DCV Session Manager Amazon DCV-Server — Referenz zur DNS-Zuordnung

Das Amazon DCV Connection Gateway benötigt die DNS-Namen der Amazon DCV-Server, um eine Verbindung zu den DCV-Server-Instances herzustellen. In diesem Abschnitt wird veranschaulicht, wie Sie eine JSON-Datei definieren können, die die Zuordnung zwischen jedem DCV-Server und dem zugehörigen DNS-Namen enthält.

Dateistruktur

Die Zuordnung besteht aus einer Liste von JSON-Objekten mit den folgenden Feldern:

```
[
 {
  “ServerIdType”: "Ip",
  “ServerId”: "192.168.0.1",
```

```
  “DnsNames”:
  {
   “InternalDnsName”: "internal"
  }
 },
 ...
]
```

Wobei Folgendes gilt:

`ServerIdType:`

Identifiziert, auf welchen ID-Typ sich der Wert bezieht. Derzeit sind die verfügbaren Werte ipAddress agentServerId, und instanceId:

`Ip:`

Sowohl für Amazon EC2 - als auch für lokale Infrastrukturen verfügbar; kann von Systemadministratoren mit einem Befehl ifconfig (Linux/macOS) oder ipconfig (Windows) schnell abgerufen werden. Diese Informationen sind auch in der API-Antwort verfügbar. DescribeServers

`Id:`

Der Session Manager Agent ist sowohl für Amazon EC2 - als auch für lokale Infrastrukturen verfügbar. Jedes Mal, wenn sich der Hostname oder die IP-Adresse ändert, erstellt er eine neue UUID. Diese Informationen sind in der API-Antwort verfügbar. DescribeServers

`Host.Aws.Ec2InstanceId:`

Sie ist nur für EC2 Amazon-Instances verfügbar und identifiziert eine Maschine eindeutig. Sie ändert sich nach einem Instance-Neustart nicht. Kann auf dem Host abgerufen werden, indem Sie sich an http://169.254.169.254/ latest/meta-data/instance -id wenden. Diese Informationen sind auch in der DescribeServers API-Antwort verfügbar.

`ServerId:`

Eine ID des angegebenen Typs, die jeden Amazon DCV-Server im Netzwerk eindeutig identifiziert.

`DnsNames:`

Das Objekt, das die DNS-Namen enthält, die mit dem Amazon DCV-Server verknüpft sind. Dieses Objekt enthält:

`InternalDnsNames:`

Der DNS-Name, der vom Amazon DCV Connection Gateway verwendet wird, um eine Verbindung mit der Instance herzustellen.

Bitte verwenden Sie die CLI-Befehle von Session Manager Broker, `register-server-dns-mapping` um das Mapping aus einer Datei zu laden (Befehlsseiten-Referenz: register-server-dns-mapping) und `describe-server-dns-mappings` um die aktuell im Session Manager Broker geladenen Mappings aufzulisten (Befehlsseiten-Referenz: describe-server-dns-mappings).

Persistenz

Es wird dringend empfohlen, die Persistenzfunktion des Session Manager Brokers zu aktivieren, um sich vor dem Verlust der Zuordnung zu schützen, wenn mehrere Broker oder der gesamte Cluster ausfallen. Weitere Informationen zur Aktivierung der Datenpersistenz finden Sie unter Broker-Persistenz konfigurieren

Integration mit Amazon CloudWatch

Session Manager unterstützt die Integration mit Amazon CloudWatch für Brokers, die auf EC2 Amazon-Instances ausgeführt werden, sowie für Brokers, die auf lokalen Hosts ausgeführt werden.

Amazon CloudWatch überwacht Ihre Amazon Web Services (AWS) -Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können CloudWatch damit Metriken sammeln und verfolgen. Dabei handelt es sich um Variablen, die Sie für Ihre Ressourcen und Anwendungen messen können. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Sie können den Session Manager Broker so konfigurieren, dass er die folgenden Metrikdaten an Amazon sendet CloudWatch:

- `Number of DCV servers`— Die Anzahl der vom Broker verwalteten DCV-Server.
- `Number of ready DCV servers`— Die Anzahl der DCV-Server, die sich in dem vom Broker verwalteten `READY` Status befinden.
- `Number of DCV sessions`— Die Anzahl der vom Broker verwalteten DCV-Sitzungen.
- `Number of DCV console sessions`— Die Anzahl der vom Broker verwalteten DCV-Konsolensitzungen.

- `Number of DCV virtual sessions`— Die Anzahl der vom Broker verwalteten virtuellen DCV-Sitzungen.
- `Heap memory used`— Die Menge des vom Broker verwendeten Heap-Speichers.
- `Off-heap memory used`— Die Menge des vom Broker verwendeten Off-Heap-Speichers.
- `Describe sessions request time`— Die Zeit, die zum Abschließen von DescribeSessions API-Anfragen benötigt wurde.
- `Delete sessions request time`— Die Zeit, die für die Fertigstellung von DeleteSessions API-Anfragen benötigt wurde.
- `Create sessions request time`— Die Zeit, die für die Fertigstellung von CreateSessions API-Anfragen benötigt wurde.
- `Get session connection data request time`— Die Zeit, die für die Fertigstellung von GetSessionConnectionData API-Anfragen benötigt wurde.
- `Update session permissions sequest time`— Die Zeit, die für die Fertigstellung von UpdateSessionPermissions API-Anfragen benötigt wurde.

Um den Broker so zu konfigurieren, dass er Metrikdaten an Amazon sendet CloudWatch

1. Öffnen Sie `/etc/dcv-session-manager-broker/session-manager-broker.properties` mit Ihrem bevorzugten Texteditor und gehen Sie wie folgt vor:
   - Stellen Sie ein `enable-cloud-watch-metrics` auf `true`
   - Geben Sie für die Region an`cloud-watch-region`, in der die metrischen Daten erfasst werden sollen.

   > **Note**
   >
   > Wenn Ihr Broker auf einer EC2 Amazon-Instance läuft, ist dieser Parameter optional. Die Region wird automatisch aus dem Instance Metadata Service (IMDS) abgerufen. Wenn Sie den Broker auf einem lokalen Host ausführen, ist dieser Parameter obligatorisch.

2. Stoppen Sie den Broker und starten Sie ihn neu.

```
$  sudo systemctl stop dcv-session-manager-broker
```

```
$  sudo systemctl start dcv-session-manager-broker
```

Der Broker-Host muss außerdem über die Berechtigung verfügen, die `cloudwatch:PutMetricData` API aufzurufen. AWS Anmeldeinformationen können mit einer der unterstützten Techniken zum Abrufen von Anmeldeinformationen abgerufen werden. Weitere Informationen finden Sie unter Angeben und Abrufen AWS von Anmeldeinformationen.

Den Amazon DCV Session Manager aktualisieren

Da Amazon DCV-Systeme immer größer und komplexer werden, ist es wichtig, sicherzustellen, dass der Session Manager auch weiterhin up-to-date in der Lage ist, die steigenden Anforderungen zu bewältigen. Sowohl die Agenten- als auch die Broker-Pakete müssen von Zeit zu Zeit aktualisiert werden. In diesem Abschnitt wird der Upgrade-Prozess für den Amazon DCV Session Manager beschrieben, der Upgrade-Vorgang und Empfehlungen für die Wartung Ihres Systems behandelt.

Im folgenden Thema wird beschrieben, wie Sie den Session Manager aktualisieren.

> **Note**
>
> Es wird dringend empfohlen, alle Session Manager-Agenten zu aktualisieren, bevor Sie die Session Manager-Broker aktualisieren, um Inkompatibilitätsprobleme bei der Einführung neuer Funktionen zu vermeiden.

Themen

Den Amazon DCV Session Manager-Agenten aktualisieren

Amazon DCV Session Manager-Agenten erhalten Anweisungen vom Broker und führen sie auf ihren jeweiligen Amazon DCV-Servern aus. Im Rahmen der routinemäßigen Wartung müssen die Agenten aktualisiert werden, um neuen Standards und Anforderungen gerecht zu werden. In diesem Abschnitt werden Sie Schritt für Schritt durch den Upgrade-Prozess Ihrer Session Manager-Agenten geführt.

Linux host

> **Note**
>
> Die folgenden Anweisungen beziehen sich auf die Installation des Agenten auf 64-Bit-x86-Hosts. Um den Agenten auf 64-Bit-ARM-Hosts für Amazon Linux, RHEL und Centos zu installieren`aarch64`, *`x86_64`* ersetzen Sie ihn durch und für Ubuntu durch*`amd64`*. `arm64`

Um den Agenten auf einem Linux-Host zu aktualisieren

1. Führen Sie den folgenden Befehl aus, um den Agenten zu beenden.

```
$  sudo systemctl stop dcv-session-manager-agent
```

2. Laden Sie das Installationspaket herunter.

   - Amazon Linux 2 und RHEL 7.x

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent-2025.0.888-1.el7.x86_64.rpm
```

   - RHEL 8.x und Rocky Linux 8.x

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent-2025.0.888-1.el8.x86_64.rpm
```

   - Ubuntu 20.04

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent_2025.0.888-1_amd64.ubuntu2004.deb
```

   - Ubuntu 22.04

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent_2025.0.888-1_amd64.ubuntu2204.deb
```

   - Ubuntu 24.04

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent_2025.0.888-1_amd64.ubuntu2404.deb
```

   - SUSE Linux Enterprise 12

```
$  curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent-2025.0.888-1.sles12.x86_64.rpm
```

   - SUSE Linux Enterprise 15

```
$  curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
nice-dcv-session-manager-agent-2025.0.888-1.sles15.x86_64.rpm
```

3. Installieren Sie das Paket .

   - Amazon Linux 2 und RHEL 7.x

     ```
     $  sudo yum install -y nice-dcv-session-manager-
     agent-2025.0.888-1.el7.x86_64.rpm
     ```

   - RHEL 8.x und Rocky Linux 8.x

     ```
     $  sudo yum install -y nice-dcv-session-manager-
     agent-2025.0.888-1.el8.x86_64.rpm
     ```

   - Ubuntu 20.04

     ```
     $  sudo apt install ./nice-dcv-session-manager-
     agent_2025.0.888-1_amd64.ubuntu2004.deb
     ```

   - Ubuntu 22.04

     ```
     $  sudo apt install ./nice-dcv-session-manager-
     agent_2025.0.888-1_amd64.ubuntu2204.deb
     ```

   - Ubuntu 24.04

     ```
     $  sudo apt install ./nice-dcv-session-manager-
     agent_2025.0.888-1_amd64.ubuntu2404.deb
     ```

   - SUSE Linux Enterprise 12

     ```
     $  sudo zypper install nice-dcv-session-manager-
     agent-2025.0.888-1.sles12.x86_64.rpm
     ```

   - SUSE Linux Enterprise 15

     ```
     $  sudo zypper install nice-dcv-session-manager-
     agent-2025.0.888-1.sles15.x86_64.rpm
     ```

4. Führen Sie den folgenden Befehl aus, um den Agenten zu starten.

```
$  sudo systemctl start dcv-session-manager-agent
```

Windows host

Um den Agenten auf einem Windows-Host zu aktualisieren

1. Beenden Sie den Agent-Dienst. Führen Sie die folgenden Befehle an der Eingabeaufforderung aus.

   ```
   C:\>  sc stop DcvSessionManagerAgentService
   ```

2. Laden Sie das Agent-Installationsprogramm herunter.
3. Führen Sie das Installationsprogramm aus. Klicken Sie auf der Willkommensseite auf Weiter.
4. Lesen Sie auf dem EULA-Bildschirm die Lizenzvereinbarung sorgfältig durch. Wenn Sie damit einverstanden sind, wählen Sie Ich akzeptiere die Bedingungen und dann Weiter.
5. Um mit der Installation zu beginnen, wählen Sie Installieren.
6. Starten Sie den Agent-Dienst neu. Führen Sie die folgenden Befehle an der Eingabeaufforderung aus.

   ```
   C:\>  sc start DcvSessionManagerAgentService
   ```

macOS host

Um den Agenten auf einem macOS-Host zu aktualisieren

1. Führen Sie den folgenden Befehl aus, um den Agenten zu beenden.

   ```
   $  sudo launchctl unload /Library/LaunchDaemons/com.amazon.dcv.session-
   manager.agent.plist
   ```

2. Laden Sie das Agent-Installationsprogramm herunter oder verwenden Sie die Befehlszeile:

   ```
   $ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerAgents/
   nice-dcv-session-manager-agent-2025.0.888-macos-arm64.pkg
   ```

3. Installieren Sie das Paket . Sie können entweder die Befehlszeile oder den Agent-Installer verwenden:

- Befehlszeile:

```
$ sudo installer -pkg ./nice-dcv-session-manager-agent-2025.0.888-macos-arm64.pkg -target /
```

- Grafisch: Doppelklicken Sie auf die `.pkg` Datei und folgen Sie dem Installationsassistenten.

4. Führen Sie den folgenden Befehl aus, um den Agenten zu starten.

```
$  sudo launchctl load /Library/LaunchDaemons/com.amazon.dcv.session-manager.agent.plist
```

Den Amazon DCV Session Manager-Broker aktualisieren

Amazon DCV Session Manager-Broker leiten API-Anfragen an ihre jeweiligen Agenten weiter. Sie werden auf einem Host installiert, der von den Amazon DCV-Servern getrennt ist. Im Rahmen der routinemäßigen Wartung müssen Broker aktualisiert werden, um neuen Standards und Anforderungen gerecht zu werden. In diesem Abschnitt werden Sie Schritt für Schritt durch den Upgrade-Prozess Ihrer Session Manager-Broker geführt.

Um den Broker zu aktualisieren

1. Connect zu dem Host her, auf dem Sie den Broker aktualisieren möchten.
2. Beenden Sie den Broker-Service.

```
$  sudo systemctl stop dcv-session-manager-broker
```

3. Laden Sie das Installationspaket herunter.

- Amazon Linux 2 und RHEL 7.x

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2025.0.539-1.el7.noarch.rpm
```

- RHEL 8.x und Rocky Linux 8.x

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2025.0.539-1.el8.noarch.rpm
```

- Ubuntu 20.04

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerBrokers/nice-
dcv-session-manager-broker-2025.0.539-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerBrokers/nice-
dcv-session-manager-broker-2025.0.539-1_all.ubuntu2204.deb
```

- Ubuntu 24.04

```
$  wget https://d1uj6qtbmh3dt5.cloudfront.net/2025.0/SessionManagerBrokers/nice-
dcv-session-manager-broker-2025.0.539-1_all.ubuntu2404.deb
```

4. Installieren Sie das Paket .

- Amazon Linux 2 und RHEL 7.x

```
$  sudo yum install -y nice-dcv-session-manager-
broker-2025.0.539-1.el7.noarch.rpm
```

- RHEL 8.x und Rocky Linux 8.x

```
$  sudo yum install -y nice-dcv-session-manager-
broker-2025.0.539-1.el8.noarch.rpm
```

- Ubuntu 20.04

```
$  sudo apt install -y nice-dcv-session-manager-
broker-2025.0.539-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$  sudo apt install -y nice-dcv-session-manager-
broker-2025.0.539-1_all.ubuntu2204.deb
```

- Ubuntu 24.04

```
$  sudo apt install -y nice-dcv-session-manager-
broker-2025.0.539-1_all.ubuntu2404.deb
```

5. Starten Sie den Broker-Service und stellen Sie sicher, dass er bei jedem Start der Instanz automatisch gestartet wird.

```
$  sudo systemctl start dcv-session-manager-broker && sudo systemctl enable dcv-
session-manager-broker
```

Broker-CLI-Referenz

Der Amazon DCV Session Manager Broker ist ein Befehlszeilenschnittstellentool (CLI), das die administrative Kontrolle über den Session Manager ermöglicht. Diese Referenz behandelt den vollständigen Satz von CLI-Befehlen, die für die Verwaltung von Sitzungen, Benutzern, Ressourcen und anderen Aspekten des Session Managers verfügbar sind. Administratoren können routinemäßige Verwaltungsaufgaben automatisieren, Probleme beheben und die Leistung ihrer Amazon DCV-Infrastruktur optimieren.

Verwenden Sie die folgenden Befehle, wenn Sie einen externen Authentifizierungsserver verwenden, um OAuth 2.0-Zugriffstoken zu generieren:

- register-auth-server
- list-auth-servers
- unregister-auth-server

Verwenden Sie die folgenden Befehle, wenn Sie den Session Manager-Broker als OAuth 2.0-Authentifizierungsserver verwenden.

- register-api-client
- describe-api-clients
- unregister-api-client
- renew-auth-server-api-Taste

Verwenden Sie die folgenden Befehle, um den Session Manager-Agent zu verwalten.

- generate-software-statement
- describe-software-statements
- deactivate-software-statement
- describe-agent-clients
- unregister-agent-client

Verwenden Sie die folgenden Befehle, um die Datei für die Zuordnung von DCV-Servern und DNS-Namen zu verwalten.

- register-server-dns-mappings
- describe-server-dns-mappings

register-auth-server

Registriert einen externen Authentifizierungsserver zur Verwendung mit dem Broker.

Standardmäßig verwendet Session Manager den Broker als Authentifizierungsserver, um OAuth 2.0-Zugriffstoken zu generieren. Wenn Sie den Broker als Authentifizierungsserver verwenden, ist keine zusätzliche Konfiguration erforderlich.

Wenn Sie sich jedoch dafür entscheiden, einen externen Authentifizierungsserver wie Active Directory oder Amazon Cognito zu verwenden, müssen Sie diesen Befehl verwenden, um den externen Authentifizierungsserver zu registrieren.

Themen

Syntax

```
sudo -u root dcv-session-manager-broker register-auth-server --url  server_url.well-known/jwks.json
```

Optionen

`--url`

Die URL des externen Authentifizierungsservers, der verwendet werden soll. Sie müssen die URL `.well-known/jwks.json` an den Authentifizierungsserver anhängen.

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel

Im folgenden Beispiel wird ein externer Authentifizierungsserver mit der `https://my-auth-server.com/` URL registriert.

Befehl

```
sudo -u root dcv-session-manager-broker register-auth-server --url https://my-auth-server.com/.well-known/jwks.json
```

Ausgabe

```
Jwk url registered.
```

list-auth-servers

Listet die externen Authentifizierungsserver auf, die registriert wurden.

Themen

Syntax

```
sudo -u root dcv-session-manager-broker list-auth-servers
```

Output

`Urls`

Der URLs der registrierten externen Authentifizierungsserver.

Beispiel

Das folgende Beispiel listet alle externen Authentifizierungsserver auf, die registriert wurden.

Befehl

```
sudo -u root dcv-session-manager-broker list-auth-servers
```

Ausgabe

```
Urls: [ "https://my-auth-server.com/.well-known/jwks.json" ]
```

unregister-auth-server

Hebt die Registrierung eines externen Authentifizierungsservers auf. Nachdem Sie die Registrierung eines externen Authentifizierungsservers aufgehoben haben, kann er nicht mehr zum Generieren von OAuth 2.0-Zugriffstoken verwendet werden.

Themen

Syntax

```
sudo -u root dcv-session-manager-broker unregister-auth-server --url  server_url.well-known/jwks.json
```

Optionen

`--url`

Die URL des externen Authentifizierungsservers, für den die Registrierung aufgehoben werden soll. Sie müssen die URL `.well-known/jwks.json` an den Authentifizierungsserver anhängen.

Typ: Zeichenfolge

Erforderlich: Ja

Output

`Url`

Die URL des nicht registrierten externen Authentifizierungsservers.

Beispiel

Im folgenden Beispiel wird ein externer Authentifizierungsserver mit der `https://my-auth-server.com/` URL registriert.

Befehl

```
sudo -u root dcv-session-manager-broker unregister-auth-server --url https://my-auth-server.com/.well-known/jwks.json
```

Ausgabe

```
Jwk urlhttps://my-auth-server.com/.well-known/jwks.json unregistered
```

register-api-client

Registriert einen Session Manager-Client beim Broker und generiert Client-Anmeldeinformationen, die vom Client verwendet werden können, um ein OAuth 2.0-Zugriffstoken abzurufen, das für API-Anfragen benötigt wird.

> **Important**
>
> Stellen Sie sicher, dass Sie die Anmeldeinformationen an einem sicheren Ort aufbewahren. Sie können später nicht wiederhergestellt werden.

Dieser Befehl wird nur verwendet, wenn der Broker als OAuth 2.0-Authentifizierungsserver verwendet wird.

Themen

Syntax

```
sudo -u root dcv-session-manager-broker register-api-client --client-name client_name
```

Optionen

`--name`

Ein eindeutiger Name, der zur Identifizierung des Session Manager-Clients verwendet wird.

Typ: Zeichenfolge

Erforderlich: Ja

Output

`client-id`

Die eindeutige Client-ID, die vom Session Manager-Client zum Abrufen eines OAuth 2.0-Zugriffstokens verwendet werden soll.

`client-password`

Das Passwort, das vom Session Manager-Client zum Abrufen eines OAuth 2.0-Zugriffstokens verwendet werden soll.

Beispiel

Im folgenden Beispiel wird ein Client mit dem Namen registriert`my-sm-client`.

Befehl

```
sudo -u root dcv-session-manager-broker register-api-client --client-name my-sm-client
```

Ausgabe

```
client-id: 21cfe9cf-61d7-4c53-b1b6-cf248EXAMPLE
client-password: NjVmZDRlN2ItNjNmYS00M2QxLWFlZmMtZmNmMDNkMEXAMPLE
```

describe-api-clients

Listet die Session Manager-Clients auf, die beim Broker registriert wurden.

Themen

Syntax

```
sudo -u root dcv-session-manager-broker describe-api-clients
```

Output

`name`

Der eindeutige Name des Session Manager-Clients.

`id`

Die eindeutige ID des Session Manager-Clients.

`active`

Zeigt den Status des Session Manager-Clients an. Wenn der Client aktiv ist, lautet der Wert`true`; andernfalls ist er`false`.

Beispiel

Das folgende Beispiel listet die registrierten Session Manager-Clients auf.

Befehl

```
sudo -u root dcv-session-manager-broker describe-api-clients
```

Ausgabe

```
Api clients
[ {
"name" : "client-abc",
"id" : "f855b54b-40d4-4769-b792-b727bEXAMPLE",
"active" : false
}, {
"name" : "client-xyz",
"id" : "21cfe9cf-61d7-4c53-b1b6-cf248EXAMPLE",
"active" : true
}]
```

unregister-api-client

Deaktiviert einen registrierten Session Manager-Client. Ein deaktivierter Session Manager-Client kann seine Anmeldeinformationen nicht mehr zum Abrufen von OAuth 2.0-Zugriffstoken verwenden.

Themen

Syntax

```
sudo -u root dcv-session-manager-broker unregister-api-client --client-id client_id
```

Optionen

`--client -id`

Die Client-ID des Session Manager-Clients, der deaktiviert werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel

Im folgenden Beispiel wird ein Session Manager-Client mit der Client-ID von `f855b54b-40d4-4769-b792-b727bEXAMPLE` deaktiviert.

Befehl

```
sudo -u root dcv-session-manager-broker unregister-api-client --client-id
 f855b54b-40d4-4769-b792-b727bEXAMPLE
```

Ausgabe

```
Client f855b54b-40d4-4769-b792-b727bEXAMPLE unregistered.
```

renew-auth-server-api-Taste

Erneuert die öffentlichen und privaten Schlüssel, die vom Broker zum Signieren der OAuth 2.0-Zugriffstoken verwendet werden, die an den Session Manager-Client verkauft werden. Wenn Sie die Schlüssel erneuern, müssen Sie dem Entwickler den neuen privaten Schlüssel zur Verfügung stellen, da er für API-Anfragen benötigt wird.

Themen

Syntax

```
sudo -u root dcv-session-manager-broker renew-auth-server-api-key
```

Beispiel

Im folgenden Beispiel werden die öffentlichen und privaten Schlüssel erneuert.

Befehl

```
sudo -u root dcv-session-manager-broker renew-auth-server-api-key
```

Ausgabe

```
Keys renewed.
```

generate-software-statement

Generiert eine Softwareanweisung.

Agenten müssen beim Broker registriert sein, um die Kommunikation zu ermöglichen. Agenten benötigen eine Softwareerklärung, um sich beim Broker registrieren zu können. Sobald der Agent über eine Softwareanweisung verfügt, kann er sich mithilfe des OAuth 2.0 Dynamic Client Registration Protocol automatisch beim Broker registrieren. Nachdem sich der Agent beim Broker registriert hat, erhält er eine Client-ID und einen geheimen Client-Schlüssel, mit denen er sich beim Broker authentifiziert.

Der Broker und der Agent erhalten und verwenden bei der ersten Installation eine Standard-Softwareanweisung. Sie können weiterhin die Standard-Softwareanweisung verwenden, oder Sie können sich dafür entscheiden, eine neue zu generieren. Wenn Sie eine neue Softwareanweisung generieren, müssen Sie die Softwareanweisung in einer neuen Datei auf dem Agenten platzieren und dann den Dateipfad zum `agent.software_statement_path` Parameter in der `agent.conf` Datei hinzufügen. Nachdem Sie dies getan haben, beenden Sie den Agenten und starten ihn neu, damit er sich mit der neuen Softwareanweisung beim Broker registrieren kann.

Themen

Syntax

```
sudo -u root dcv-session-manager-broker generate-software-statement
```

Output

`software-statement`

Die Softwareanweisung.

Beispiel

Das folgende Beispiel generiert eine Softwareanweisung.

Befehl

```
sudo -u root dcv-session-manager-broker generate-software-statement
```

Ausgabe

```
software-statement:
 ewogICJpZCIgOiAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiYWN0aXZlIiA6IHRydWUsCi
```

describe-software-statements

Beschreibt die vorhandenen Softwareanweisungen.

Themen

Syntax

```
sudo -u root dcv-session-manager-broker describe-software-statements
```

Output

`software-statement`

Die Softwareanweisung.

`issued-at`

Datum und Uhrzeit der Softwaregenerierung.

`is-active`

Der aktuelle Status der Softwareanweisung. `true`wenn die Softwareanweisung aktiv ist; andernfalls ist sie es`false`.

Beispiel

Das folgende Beispiel generiert eine Softwareanweisung.

Befehl

```
sudo -u root dcv-session-manager-broker describe-software-statements
```

Ausgabe

```
Software Statements
[ {
"software-statement" :
 "ewogICJpZCIgOiAiYmEEXAMPLEYtNzUwNy00YmFhLTliZWItYTA1MmJjZTE3NDJjIiwKICAiaXNzdWVkQXQiIDogMTU5N
"issued-at" : "2020.08.05 15:38:32 +0000",
"is-active" : "true"
}, {
"software-statement" :
 "EXAMPLEpZCIgOiAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNzdWEXAMPLEDogMTU5N
"issued-at" : "2020.08.07 10:24:41 +0000",
"is-active" : "true"
} ]
```

deactivate-software-statement

Deaktiviert eine Softwareanweisung. Wenn Sie eine Softwareanweisung deaktivieren, kann sie nicht mehr für Agentenregistrierungen verwendet werden.

Themen

Syntax

```
sudo -u root dcv-session-manager-broker deactivate-software-statement --software-
statement software_statement
```

Optionen

`--software-statement`

Die Softwareanweisung, die deaktiviert werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel

Im folgenden Beispiel wird eine Softwareanweisung deaktiviert.

Befehl

```
sudo -u root dcv-session-manager-broker deactivate-software-statement --software-
statement
 EXAMPLEpZCIgOiAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj
```

Ausgabe

```
Software statement
 EXAMPLEpZCIgOiAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj
 deactivated
```

describe-agent-clients

Beschreibt die Agenten, die beim Broker registriert sind.

Themen

Syntax

```
sudo -u root dcv-session-manager-broker describe-agent-clients
```

Output

`name`

Der Name des Agenten.

`id`

Die eindeutige ID des Agenten.

`active`

Der Status des Agenten. `true`ob der Agent aktiv ist; andernfalls ist er es`false`.

Beispiel

Das folgende Beispiel beschreibt die Agenten.

Befehl

```
sudo -u root dcv-session-manager-broker describe-agent-clients
```

Ausgabe

```
Session manager agent clients
[ {
"name" : "test",
"id" : "6bc05632-70cb-4410-9e54-eaf9bEXAMPLE",
"active" : true
}, {
"name" : "test",
"id" : "27131cc2-4c71-4157-a4ca-bde38EXAMPLE",
"active" : true
}, {
"name" : "test",
"id" : "308dd275-2b66-443f-95af-33f63EXAMPLE",
"active" : false
}, {
"name" : "test",
"id" : "ce412d1b-d75c-4510-a11b-9d9a3EXAMPLE",
"active" : true
} ]
```

unregister-agent-client

Heben Sie die Registrierung eines Agenten beim Broker auf.

Themen

Syntax

```
sudo -u root dcv-session-manager-broker unregister-agent-client --client-id client_id
```

Optionen

`--client-id`

Die ID des Agenten, dessen Registrierung aufgehoben werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel

Im folgenden Beispiel wird die Registrierung eines Agenten aufgehoben.

Befehl

```
sudo -u root dcv-session-manager-broker unregister-agent-client --client-id
 3b0d7b1d-78c7-4e79-b2e1-b976dEXAMPLE
```

Ausgabe

```
agent client 3b0d7b1d-78c7-4e79-b2e1-b976dEXAMPLE unregistered
```

register-server-dns-mappings

Registrieren Sie die DCV-Server — DNS-Namenszuordnungen, die aus einer JSON-Datei stammen.

Syntax

```
sudo -u root dcv-session-manager-broker register-server-dns-mappings --file-
path file_path
```

Optionen

`--file-path`

Der Pfad der Datei, die die Zuordnungen der DCV-Server — DNS-Namen enthält.

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel

Im folgenden Beispiel werden die Zuordnungen von DCV-Servern und DNS-Namen aus der Datei /tmp/mappings.json registriert.

Befehl

```
sudo -u root dcv-session-manager-broker register-server-dns-mappings --file-path /tmp/
mappings.json
```

Ausgabe

```
 Successfully loaded 2 server id - dns name mappings from file /tmp/mappings.json
```

describe-server-dns-mappings

Beschreiben Sie die derzeit verfügbaren Zuordnungen von DCV-Servern und DNS-Namen.

Syntax

```
sudo -u root dcv-session-manager-broker describe-server-dns-mappings
```

Output

`serverIdType`

Der Typ der Server-ID.

`serverId`

Die eindeutige ID des Servers.

`dnsNames`

Die internen und externen DNS-Namen

`internalDnsNames`

Die internen DNS-Namen

`externalDnsNames`

Die externen DNS-Namen

Beispiel

Im folgenden Beispiel werden die registrierten Zuordnungen von DCV-Servern und DNS-Namen aufgeführt.

Befehl

```
sudo -u root dcv-session-manager-broker describe-server-dns-mappings
```

Ausgabe

```
[
{
 "serverIdType" : "Id",
 "serverId" : "192.168.0.1",
 "dnsNames" : {
```

```
   "internalDnsName" : "internal1",
   "externalDnsName" : "external1"
  }
 },
 {
  "serverIdType" : "Host.Aws.Ec2InstanceId",
  "serverId" : "i-0648aee30bc78bdff",
  "dnsNames" : {
   "internalDnsName" : "internal2",
   "externalDnsName" : "external2"
  }
 }
 ]
```

Referenz zur Konfigurationsdatei

Dieser Referenzabschnitt bietet einen Überblick über die verfügbaren Konfigurationsoptionen für den Session Manager. Konfigurationen beinhalten Änderungen sowohl an der Agent- als auch an der Brokerdatei. Jede Konfiguration enthält eine Erläuterung des Zwecks, der akzeptierten Werte und der Auswirkungen auf das allgemeine Systemverhalten. Amazon DCV Session Manager kann an die individuellen Anforderungen eines Amazon DCV-Systems angepasst werden.

Themen

Broker-Konfigurationsdatei

Die Broker-Konfigurationsdatei (`/etc/dcv-session-manager-broker/session-manager-broker.properties`) enthält Parameter, die konfiguriert werden können, um die Session Manager-Funktionalität anzupassen. Sie können die Konfigurationsdatei mit Ihrem bevorzugten Texteditor bearbeiten.

> **Note**
>
> Die `/etc/dcv-session-manager-broker/session-manager-broker.properties` Datei enthält sensible Daten. Standardmäßig ist der Schreibzugriff auf Root und der Lesezugriff auf Root und den Benutzer beschränkt, der den Broker ausführt. Standardmäßig ist dies der `dcvsmbroker` Benutzer. Der Broker überprüft beim Start, ob die Datei über die erwarteten Berechtigungen verfügt.

In der folgenden Tabelle sind die Parameter in der Broker-Konfigurationsdatei aufgeführt.

Name des Parameters	Erforderlich	Standardwert	Description
`broker-java-home`	Nein		Gibt den Pfad zum Java-Home-Verzeichnis an, das der Broker anstelle des Standardverzeichnisses des Systems verwenden wird. Wenn diese Option gesetzt ist, verwendet der Broker sie `<broker-java-home>/bin/java` beim Start. Tipp: Der Broker benötigt Java Runtime Environment 11 und wird nach erfolgreicher Installation als Abhängigkeit installiert, falls es fehlt. Wenn Version 11 nicht als Standard-Java-Umgebung festgelegt ist, kann das zugehörige Home-Verzeichnis mit dem folgenden Befehl abgerufen werden: `$ sudo alternatives --display java`
`sessions-creensh-max-`	Nein	`160`	Gibt die maximale Breite von Sitzungs-Screenshots, die mit der GetSessionScreensh

Name des Parameters	Erforderlich	Standardwert	Description
`width`			otsAPI aufgenommen wurden, in Pixeln an.
`sessions-creenshot-max-height`	Nein	`100`	Gibt die maximale Höhe von Sitzungs-Screenshots, die mit der GetSessionScreenshotsAPI aufgenommen wurden, in Pixeln an.
`sessions-creenshot-format`	Nein	`png`	Das Bilddateiformat von Sitzungs-Screenshots, die mit der GetSessionScreenshotsAPI aufgenommen wurden.
`create-sessions-queue-max-size`	Nein	`1000`	Die maximale Anzahl unerfüllter CreateSessionsAPI-Anfragen, die in die Warteschlange gestellt werden können. Wenn die Warteschlange voll ist, werden neue unerfüllte Anfragen abgelehnt.

Name des Parameters	Erforderlich	Standardwert	Description
`create-sessions-queue-max-time-seconds`	Nein	`1800`	Die maximale Zeit in Sekunden, für die eine unerfüllte CreateSessionsAPI-Anfrage in der Warteschlange verbleiben kann. Wenn die Anfrage nicht innerhalb des angegebenen Zeitraums erfüllt werden kann, schlägt sie fehl.
`session-manager-working-path`	Ja	`/tmp`	Gibt den Pfad zu dem Verzeichnis an, in das der Broker die für den Betrieb erforderlichen Dateien schreibt. Dieses Verzeichnis darf nur für den Broker zugänglich sein.
`enable-authorization-server`	Ja	`true`	Gibt an, ob der Broker der Authentifizierungsserver ist, der verwendet wird, um OAuth 2.0-Zugriffstoken für den Client zu generieren APIs.

Name des Parameters	Erforderlich	Standardwert	Description
`enable-authorization`	Ja	`true`	Aktiviert oder deaktiviert die Client-Autorisierung. Wenn Sie die Client-Autorisierung aktivieren, muss die Client-API bei API-Anfragen ein Zugriffstoken bereitstellen. Wenn Sie die Client-Autorisierung deaktivieren, APIs kann der Client Anfragen ohne Zugriffstoken stellen.
`enable-agent-authorizatio`	Ja	`true`	Aktiviert oder deaktiviert die Agentenautorisierung. Wenn Sie die Agentenautorisierung aktivieren, muss der Agent bei der Kommunikation mit dem Broker ein Zugriffstoken bereitstellen.

Name des Parameters	Erforderlich	Standardwert	Description
`delete-session-duration-hours`	Nein	1	Gibt die Anzahl der Stunden an, nach denen gelöschte Sitzungen unsichtbar werden und nicht mehr durch `DescribeSession` API-Aufrufe zurückgegeben werden. Veraltet: `delete-session-duration-hours` Änderung zu `delete-session-duration-seconds` — Verfügbar seit Version 2024.0-493.
`delete-session-duration-seconds`	Nein	`3600`	Gibt die Anzahl der Sekunden an, nach denen gelöschte Sitzungen unsichtbar werden und nicht mehr von API-Aufrufen zurückgegeben werden. `DescribeSession` Dieser Parameter ersetzt den veralteten `delete-session-duration-hours` Parameter — verfügbar seit Version 2024.0-493.

Name des Parameters	Erforderlich	Standardwert	Description
connect-session-token-duration-minutes	Nein	60	Gibt die Anzahl der Minuten an, für die das Token gültig bleibt. ConnectSession
client-to-broker-connector-https-port	Ja	8443	Gibt den HTTPS-Port an, an dem der Broker auf Client-Verbindungen wartet.
client-to-broker-connector-bind-host	Nein	0.0.0.0	Gibt die IP-Adresse des Hosts an, an den der Broker für Clientverbindungen bindet.

Name des Parameters	Erforderlich	Standardwert	Description
client-to-broker-connector-key-store-file	Ja		Gibt den Schlüsselspeicher an, der für TLS-Clientverbindungen verwendet wird.
client-to-broker-connector-key-store-pass	Ja		Gibt den Schlüsselspeicherpass an.
agent-to-broker-connector-https-port	Ja	8445	Gibt den HTTPS-Port an, an dem der Broker auf Agentenverbindungen wartet.

Name des Parameters	Erforderlich	Standardwert	Description
agent-to-broker-connector-bind-host	Nein	0.0.0.0	Gibt die IP-Adresse des Hosts an, an den der Broker für Agentenverbindungen bindet.
agent-to-broker-connector-key-store-file	Ja		Gibt den Schlüsselspeicher an, der für TLS-Agent-Verbindungen verwendet wird.
agent-to-broker-connector-key-store-pass	Ja		Gibt den Schlüsselspeicherpass an.
broker-to-broker-port	Ja	47100	Gibt den Port an, der für broker-to-broker Verbindungen verwendet wird.

Name des Parameters	Erforderlich	Standardwert	Description
`broker-to-broker-bind-host`	Nein	`0.0.0.0`	Gibt die IP-Adresse des Hosts an, an den der Broker für broker-to-broker Verbindungen bindet.
`broker-to-broker-discover-port`	Ja	`47500`	Gibt den Port an, der von Brokern verwendet wird, um sich gegenseitig zu erkennen.

Name des Parameters	Erforderlich	Standardwert	Description
`broker-to-broker-discovery-address`	Nein		Gibt die IP-Adressen und Ports der anderen Broker in der Flotte im *`port`* Format*`ip_address`* : an. Wenn es mehrere Broker gibt, trennen Sie die Werte durch ein Komma. Wenn Sie`broker-to-broker-discovery-multicast-group` ,, oder angeben `broker-to-broker-discovery-multicast-port` `broker-to-broker-discovery-AWS-region` `broker-to-broker-discovery-AWS-alb-target-group-arn`, lassen Sie diesen Parameter weg.

Name des Parameters	Erforderlich	Standardwert	Description
broker-to-broker-discovery-multicast-group	Nein		Gibt die Multicast-Gruppe für broker-to-roker die Erkennung an. Wenn Sie, oder angeben `broker-to-broker-discovery-addresses` `broker-to-broker-discovery-aws-region` `broker-to-broker-discovery-AWS-alb-target-group-arn` , lassen Sie diesen Parameter weg.
broker-to-broker-discovery-multicast-port	Nein		Gibt den Multicast-Port für broker-to-broker die Erkennung an. Wenn Sie, oder angeben `broker-to-broker-discovery-addresses` `broker-to-broker-discovery-AWS-region` `broker-to-broker-discovery-AWS-alb-target-group-arn` , lassen Sie diesen Parameter weg.

Name des Parameters	Erforderlich	Standardwert	Description
`broker-to-broker-discovery-AWS-region`	Nein		Gibt die AWS Region des Application Load Balancers an, der für die Erkennung von Broker zu Broker verwendet wird. Wenn Sie, oder angeben `broker-to-broker-discovery-multicast-group` `broker-to-broker-discovery-multicast-port` `broker-to-broker-discovery-addresses` , lassen Sie diesen Parameter weg.
`broker-to-broker-discovery-AWS-alb-target-group-arn`	Nein		Der ARN des Application Load Balancer-Zielgruppenbenutzers für die broker-to-broker Erkennung. Wenn Sie, oder angeben `broker-to-broker-discovery-multicast-group` `broker-to-broker-discovery-multicast-port` `broker-to-broker-discovery-addresses` , lassen Sie diesen Parameter weg.

Name des Parameters	Erforderlich	Standardwert	Description
broker-to-broker-distributed-memory-max-size-mb	Nein	4096	Gibt die maximale Menge an Off-Heap-Speicher an, die von einem einzelnen Broker zum Speichern von Amazon DCV-Sitzungsdaten verwendet werden soll.
broker-to-broker-key-store-file	Ja		Gibt den Schlüsselspeicher an, der für TLS-Brokerverbindungen verwendet wird.
broker-to-broker-key-store-pass	Ja		Gibt den Schlüsselspeicherpass an.

Name des Parameters	Erforderlich	Standardwert	Description
`enable-cloud-watch-metrics`	Nein	`false`	Aktiviert oder deaktiviert CloudWatch Amazon-Metriken. Wenn Sie CloudWatch Metrics aktivieren, müssen Sie möglicherweise einen Wert für `cloud-watch-region` angeben.
`cloud-watch-region`	Nein	Nur erforderlich, wenn auf gesetzt `enable-cloud-watch-metrics` ist`true`. Wenn der Broker auf einer Amazon EC2 EC2-Instance installiert ist, wird die Region aus dem IMDS abgerufen.	Die AWS Region, in der die CloudWatch Metriken veröffentlicht werden.
`max-api-requests-per-second`	Nein	`1000`	Gibt die maximale Anzahl von Anfragen an, die die Broker-API jede Sekunde verarbeiten kann, bevor sie gedrosselt wird.

Name des Parameters	Erforderlich	Standardwert	Description
`enable-throttling-forwarded-for-header`	Nein	`false`	Wenn diese Option aktiviert ist, `true` wird die IP des Anrufers aus dem Header abgerufen, falls vorhanden. X-Forwared-For
`create-sessions-number-of-retries-on-failure`	Nein	`2`	Gibt die maximale Anzahl von Wiederholungen an, die ausgeführt werden sollen, nachdem eine Anfrage zum Erstellen einer Sitzung auf einem Amazon DCV-Serverhost fehlgeschlagen ist. Auf 0 setzen, um bei Fehlern niemals Wiederholungen durchzuführen.
`autorun-file-arguments-max-size`	Nein	`50`	Gibt die maximale Anzahl von Argumenten an, die an die Autorun-Datei übergeben werden können.

Name des Parameters	Erforderlich	Standardwert	Description
`autorun-file-arguments-max-argument-length`	Nein	`150`	Gibt die maximale Länge der einzelnen Autorun-Dateiargumente in Zeichen an.
`enable-persister`	Ja	`false`	Wenn auf gesetzt`true`, werden die Broker-Statusdaten in einer externen Datenbank gespeichert.
`persistence-db`	Nein	Nur erforderlich, wenn auf `true` gesetzt `enable-persistence` ist.	Gibt an, welche Datenbank für die Persistenz verwendet wird. Die einzigen unterstützten Werte sind: `dynamodb` und`mysql`.
`dynamodb-region`	Nein	Nur erforderlich, wenn auf gesetzt `enable-persistence` `persistence-db` ist `true` und auf gesetzt ist`dynamodb`.	Gibt die Region an, in der die DynamoDB-Tabellen erstellt werden und auf die zugegriffen wird.

Name des Parameters	Erforderlich	Standardwert	Description
dynamodb-table-rcu	Nein	Nur erforderlich, wenn auf gesetzt enable-persistence ist true und auf gesetzt persistence-db ist. dynamodb	Gibt die Lesekapazitätseinheiten (RCU) für jede DynamoDB-Tabelle an. Weitere Informationen zu RCU finden Sie unter Preise für bereitgestellte Kapazität.
dynamodb-table-wcu	Nein	Nur erforderlich, wenn auf gesetzt enable-persistence ist und auf true gesetzt ist. persistence-db dynamodb	Gibt die Schreibkapazitätseinheiten (WCU) für jede DynamoDB-Tabelle an. Weitere Informationen zu WCU finden Sie unter Preise für bereitgestellte Kapazität.
dynamodb-table-name-prefix	Nein	Nur erforderlich, wenn auf eingestellt enable-persistence ist true und auf gesetzt persistence-db ist. dynamodb	Gibt das Präfix an, das jeder DynamoDB-Tabelle hinzugefügt wird (nützlich, um mehrere Broker-Cluster zu unterscheiden, die dasselbe AWS Konto verwenden). Nur alphanumerische Zeichen, Punkt, Bindestrich und Unterstrich sind zulässig.

Name des Parameters	Erforderlich	Standardwert	Description
`jdbc-connection-url`	Nein	Nur erforderlich, wenn auf gesetzt `enable-persistence` ist `true` und auf gesetzt `persistence-db` ist. `mysql`	Gibt die Verbindungs-URL zur MariaDB/MySQL Datenbank an; sie enthält den Endpunkt und den Datenbanknamen. Die URL sollte dieses Format haben: `jdbc:mysql://<db_endpoint>:<db_port>/<db_name>?createDatabaseIfNotExist=true` Wo `<db_endpoint>` ist der MariaDB/MySQL Datenbankendpunkt, `<db_port>` ist der Datenbankport und `<db_name>` ist der Datenbankname.
`jdbc-user`	Nein	Nur erforderlich, wenn auf gesetzt `enable-persistence` `persistence-db` ist `true` und auf gesetzt ist`mysql`.	Gibt den Namen des Benutzers an, der Zugriff auf die MariaDB/MySQL Datenbank hat.

Name des Parameters	Erforderlich	Standardwert	Description
`jdbc-password`	Nein	Nur erforderlich, wenn auf gesetzt `enable-persistence` `persistence-db` ist `true` und auf gesetzt ist`mysql`.	Gibt das Passwort des Benutzers an, der Zugriff auf die MariaDB/MySQL Datenbank hat.
`seconds-before-deleting-unreachable-dcv-server`	Nein	`1800`	Gibt die Anzahl der Sekunden an, nach denen ein Amazon DCV-Server, der nicht erreichbar ist, aus dem System gelöscht wird.

Name des Parameters	Erforderlich	Standardwert	Description
`seconds-before-deleting-sessions-unreachable-server`	Nein		Gibt die Anzahl der Sekunden an, nach denen Sitzungen auf einem nicht erreichbaren Amazon DCV-Server aus dem System gelöscht werden. Das Entfernen von Sitzungen von einem Server, der nicht erreichbar ist, ist standardmäßig deaktiviert. Um das Entfernen von Sitzungen von Servern zu ermöglichen, die nicht erreichbar sind, geben Sie einen gültigen Wert an.

Name des Parameters	Erforderlich	Standardwert	Description
`sessions-screenshot-max-width`	Nein	`160`	Gibt die maximale Breite von Sitzungs-Screenshots, die mit der GetSessionScreenshotsAPI aufgenommen wurden, in Pixeln an. Wenn in der Webclient-Konfigurationsdatei festgelegt `session-screenshot-max-width` ist, hat sie Vorrang und überschreibt diesen Standardwert. Beachten Sie, dass dies die maximale Breite ist, sodass die tatsächliche Bildschirmauflösung möglicherweise niedriger ist.

Name des Parameters	Erforderlich	Standardwert	Description
`sessions creenshot-max-height`	Nein	`100`	Gibt die maximale Höhe von Sitzungs-Screenshots, die mit der GetSessionScreenshotsAPI aufgenommen wurden, in Pixeln an. Wenn `session-screenshot-max-height` dies in der Webclient-Konfigurationsdatei festgelegt ist, hat es Vorrang und überschreibt diesen Standardwert. Beachten Sie, dass dies die maximale Höhe ist, sodass die tatsächliche Bildschirmauflösung möglicherweise niedriger ist.

Agent-Konfigurationsdatei

Die Agentenkonfigurationsdatei (`/etc/dcv-session-manager-agent/agent.conf`für Linux und macOS sowie `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` für Windows) enthält Parameter, die konfiguriert werden können, um die Session Manager-Funktionalität anzupassen. Sie können die Konfigurationsdatei mit Ihrem bevorzugten Texteditor bearbeiten.

In der folgenden Tabelle sind die Parameter in der Agenten-Konfigurationsdatei aufgeführt.

Name des Parameters	Erforderlich	Standardwert	Description
`agent.broker_host`	Ja		Gibt den DNS-Namen des Broker-Hosts an.
`agent.broker_port`	Ja	`8445`	Gibt den Port an, über den mit dem Broker kommuniziert werden soll.
`agent.cafile`	Nein		Wird nur benötigt, wenn auf gesetzt `tls_strict` ist `true`. Gibt den Pfad zur Zertifikatsdatei (.pem) an, die zur Validierung des TLS-Zertifikats benötigt wird. Kopieren Sie das selbstsignierte Zertifikat vom Broker auf den Agenten.
`agent.init_folder`	Nein	• `/var/lib/dcv-session-manager-agent/init` (Linux)	Gibt den Pfad zu einem Ordner auf dem Host-Server an, der zum Speichern von benutzerdefinierten Skripten verwendet wird, die Amazon DCV-Serversitzungen initialisieren dürfen, wenn sie erstellt werden. Sie müssen einen absoluten Pfad angeben. Auf den Ordner muss zugegriffen werden können und die Dateien müssen von Benutzern

Name des Parameters	Erforderlich	Standardwert	Description
			ausgeführt werden können, die den InitFileAnforderungsparameter der CreateSessionsAPI verwenden.
`agent._strict`	Nein	`true`	Gibt an, ob eine strikte TLS-Validierung verwendet werden soll.
`agent.software_statement_path`	Nein		Wird nur benötigt, wenn die Standard-Softwareanweisung nicht verwendet wird. Gibt den Pfad zur Datei mit den Softwareanweisungen an. Weitere Informationen finden Sie unter generate-software-statement.
`agent.tags_folder`	Nein	• `/etc/dcv-session-manager-agent` (Linux/macOS) • `C:\Program Files\NICE\DCVSessionManagerAgent\conf\tags` (Windows)	Gibt den Pfad zu dem Ordner an, in dem sich die Tag-Dateien befinden. Weitere Informationen finden Sie unter Verwenden von Tags als Ziel für Amazon DCV-Server.

Name des Parameters	Erforderlich	Standardwert	Description
agent.autorun_folder	Nein	• /var/lib/dcv-session-manager-agent/autorun (Linux) • C:\ProgramData\NICE\DcvSessionManagerAgent\autorun (Windows)	Gibt den Pfad zu einem Ordner auf dem Hostserver an, in dem Skripts und Apps gespeichert werden, die beim Sitzungsstart automatisch ausgeführt werden dürfen. Sie müssen einen absoluten Pfad angeben. Auf den Ordner muss zugegriffen werden können und die Dateien müssen von Benutzern ausgeführt werden können, die den AutorunFileAnforderungsparameter der CreateSessionsAPI verwenden.
agent._virtual_sessions	Nein	-1 (kein Limit)	Die maximale Anzahl virtueller Sitzungen, die mit Amazon DCV Session Manager auf einem Amazon DCV-Server erstellt werden können.
agent._concurrent_sessions_per_user	Nein	1	Die maximale Anzahl virtueller Sitzungen, die auf einem Amazon DCV-Server von einem einzelnen Benutzer mit Amazon DCV Session Manager erstellt werden können.

Name des Parameters	Erforderlich	Standardwert	Description
`agent.broker_update_interval`	Nein	30	Gibt an, wie viele Sekunden gewartet werden soll, bevor aktualisierte Daten an den Broker gesendet werden. Zu den gesendeten Daten gehören der Amazon DCV-Server- und Hoststatus sowie aktualisierte Sitzungsinformationen. Niedrigere Werte sorgen dafür, dass der Sitzungsmanager besser auf Änderungen auf dem System reagiert, auf dem der Agent ausgeführt wird, erhöhen jedoch die Systemlast und den Netzwerkverkehr. Höhere Werte verringern die System- und Netzwerklast, aber der Sitzungsmanager reagiert weniger schnell auf Systemänderungen, weshalb höhere Werte als nicht empfohlen `120` werden.

Name des Parameters	Erforderlich	Standardwert	Description
`agent.enable_queried_logged_on_users`	Nein	true	Aktivieren Sie das Abrufen von angemeldeten Benutzern unter Windows. Wenn der Wert wahr ist, wird das System nach tatsächlich angemeldeten Benutzern abgefragt. Wenn der Wert falsch ist, wird „unbekannt“ zurückgegeben
`agent.preferred_network_interface`	Nein		Gibt den Namen der Netzwerkschnittstelle an, der für die Host-IP-Erkennung verwendet werden soll. Beispiele: `eth0` (Linux), `Ethernet 4` (Windows). Linux/macOS Namen unterscheiden Groß- und Kleinschreibung, Windows-Namen unterscheiden nicht zwischen Groß- und Kleinschreibung. Wenn nicht angegeben oder wenn die angegebene Netzwerkschnittstelle nicht gefunden wird, wird die erste Nicht-Loopback-Schnittstelle ausgewählt.

Name des Parameters	Erforderlich	Standardwert	Description
`log.level`	Nein	`info`	Gibt den Ausführlichkeitsgrad der Protokolldateien an. Die folgenden Ausführlichkeitsstufen sind verfügbar: • `error`— Stellt die wenigsten Details bereit. Umfasst nur Fehler. • `warning`— Beinhaltet Fehler und Warnungen. • `info`— Die standardmäßige Ausführlichkeitsstufe. Umfasst Fehler, Warnungen und Informationsmeldungen. • `debug`— Bietet die meisten Details. Bietet detaillierte Informationen, die nützlich für das Debugging sind.
`log.directory`	Nein	• `/var/log/dcv-session-manager-agent/`(Linux/macOS) • `C:\ProgramData\NICE\DCVSessionManagerAgent\log` (Windows)	Gibt das Verzeichnis an, in dem Protokolldateien erstellt werden sollen.

Name des Parameters	Erforderlich	Standardwert	Description
`log.rotation`	Nein	`daily`	Gibt die Rotation der Protokolldateien an. Folgende sind gültige Werte: • `hourly`— Die Protokolldateien werden stündlich rotiert. • `daily`— Die Protokolldateien werden täglich rotiert.
`log.maxfile-size`	Nein	`10485760`	Wenn eine Protokolldatei die angegebene Größe in Byte erreicht, wird sie rotiert. Eine neue Protokolldatei wird erstellt und weitere Protokollereignisse werden in der neuen Datei gespeichert.
`log.rotate`	Nein	`9`	Die maximale Anzahl von Protokolldateien, die in der Rotation aufbewahrt werden. Jedes Mal, wenn eine Rotation stattfindet und diese Zahl erreicht wird, wird die älteste Protokolldatei gelöscht.

Versionshinweise und Dokumentverlauf für Amazon DCV Session Manager

Diese Seite enthält die Versionshinweise und den Dokumentverlauf für Amazon DCV Session Manager.

Themen

Versionshinweise zu Amazon DCV Session Manager

Dieser Abschnitt bietet einen Überblick über die wichtigsten Updates, Feature-Releases und Bugfixes für Amazon DCV Session Manager. Alle Updates sind nach Veröffentlichungsdatum geordnet. Wir aktualisieren die Dokumentation regelmäßig, um das Feedback zu berücksichtigen, das Sie uns senden.

Themen

- 2022.1-13067 — 29. Juni 2022
- 2022.0-11952 — 23. Februar 2022
- 2021.3-11591 — 20. Dezember 2021
- 2021.2-11445 — 18. November 2021
- 2021.2-11190 — 11. Oktober 2021
- 2021.2-11042 — 01. September 2021
- 2021.1-10557 — 31. Mai 2021
- 2021.0-10242 — 12. April 2021
- 2020.2-9662 — 04. Dezember 2020
- 2020.2-9508 — 11. November 2020

2025.0-544 — 2. Februar 2026

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 544 • Makler: 902 • CLI: 159	• `preferred_network_interface` Konfigurationsparameter für die Host-IP-Erkennung hinzugefügt.

2025.0-544 — 23. Dezember 2025

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 544 • Makler: 893 • CLI: 159	• Die Beschränkung der WebSocket Nachrichtengröße wurde erhöht, um Fehler beim Abrufen von Screenshots auf macOS-Hosts zu beheben. • Die Windows-Build-Umgebung wurde auf Visual Studio 2022 aktualisiert.

2025.0-539 — 12. November 2025

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 539 • Makler: 888 • CLI: 159	• Die macOS Agent Bundle-ID wurde von NICE Software in Amazon umbenannt.

2025.0-539 — 22. Oktober 2025

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 539 • Makler: 886 • CLI: 159	• Der Konfigurationsparameter enable_query_logged_in_users wurde zur Agenten-Konfigurationsdatei hinzugefügt, um das Abfrageverhalten angemeldeter Benutzer auf Windows-Systemen zu spezifizieren. • Befehle wurden durch systemeigene PowerShell Windows-Befehle APIs (WMI und Windows-Registrierung) ersetzt, um die Leistung und Zuverlässigkeit beim Abrufen von Systeminformationen zu verbessern. • Die DNS-Namensauflösung auf Windows Amazon EC2 EC2-Instances wurde behoben, indem die Amazon EC2 EC2-Erkennung mit einem Fallback auf den AWS Metadaten-Service verbessert wurde, wenn die UUID-basierte Erkennung fehlschlägt. • Aktualisierte Version auf 2025.

2024.0-531 — 17. Juni 2025

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 531 • Makler: 852 • CLI: 154	• Funktion hinzugefügt, um Zertifikate vor Ablauf zu verlängern. • NICE DCV wurde in Amazon DCV umbenannt. • Fehlerbehebungen

2024.0-504 — 31. März 2025

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 504 • Makler: 817 • CLI: 154	• Unterstützung für hinzugefügt AL2023. • Fehlerbehebungen und Leistungsverbesserungen.

2024.0-493 — 15. Januar 2025

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 493 • Makler: 801 • CLI: 152	• Der `GetSessionScreenshot` Anfrage wurden Parameter hinzugefügt, um die maximale Höhe und Breite des Screenshots anzugeben. • Der Broker-Konfigurationsdatei wurde ein Parameter hinzugefügt, der die Anzahl der Sekunden angibt, nach denen Sitzungen auf einem nicht erreichbaren Amazon DCV-Server aus dem System gelöscht werden. • Es wurde ein Problem behoben, bei dem der `seconds-before-deleting-unreachable-dcv-server` Parameter in der Broker-Konfigurationsdatei nicht berücksichtigt wurde. • Fehlerbehebungen und Leistungsverbesserungen.

2024.0-457 — 1. Oktober 2024

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 457 • Makler: 748 • CLI: 140	• NICE DCV wurde in Amazon DCV umbenannt. • Unterstützung für Ubuntu 24.04 hinzugefügt.

2023.1-17652 — 1. August 2024

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 426 • Makler: 748 • CLI: 140	• Fehlerbehebungen und Leistungsverbesserungen.

2023.1-16388 — 26. Juni 2024

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 417 • Makler: 748 • CLI: 140	• Es wurde ein Fehler behoben, durch den Speicher fälschlicherweise als TB und nicht als GB angezeigt wurde. • Fehlerbehebungen und Leistungsverbesserungen.

2023.1 — 9. November 2023

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 410 • Makler: 732 • CLI: 140	• Fehlerbehebungen und Leistungsverbesserungen

2023.0-15065 — 4. Mai 2023

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 392 • Makler: 675 • CLI: 132	• Unterstützung für Red Hat Enterprise Linux 9, Rocky Linux 9 und CentOS Stream 9 auf ARM-Plattformen hinzugefügt.

2023.0-14852 — 28. März 2023

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 392 • Makler: 642 • CLI: 132	• Unterstützung für Red Hat Enterprise Linux 9, Rocky Linux 9 und CentOS Stream 9 hinzugefügt.

2022.2-13907 — 11. November 2022

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 382 • Makler: 612 • CLI: 123	• `DescribeSessions` Als Antwort wurde ein `Substate` Feld hinzugefügt. • Es wurde ein Problem behoben, das dazu führen konnte, dass die CLI je nach verwendeter URL keine Verbindung zum Broker herstellen konnte.

2022.1-13067 — 29. Juni 2022

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 355 • Makler: 592 • CLI: 114	• Unterstützung für die Ausführung des Brokers AWS auf Graviton-Instances hinzugefügt. • Agenten- und Broker-Unterstützung für Ubuntu 22.04 hinzugefügt.

2022.0-11952 — 23. Februar 2022

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 341	• Dem Agenten wurde die Funktion zur Rotation von Protokollen hinzugefügt.

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 520 • CLI: 112	• Konfigurationsparameter hinzugefügt, um Java Home im Broker festzulegen. • Die Übertragung von Daten vom Cache auf die Festplatte im Broker wurde verbessert. • Die URL-Validierung in der CLI wurde behoben.

2021.3-11591 — 20. Dezember 2021

Build-Nummern	Neue Features
• Makler: 307 • Makler: 453 • CLI: 92	• Unterstützung für die Integration mit dem Amazon DCV Connection Gateway wurde hinzugefügt. • Broker-Unterstützung für Ubuntu 18.04 und Ubuntu 20.04 hinzugefügt.

2021.2-11445 — 18. November 2021

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 288 • Makler: 413 • CLI: 54	• Ein Problem mit der Überprüfung von Anmeldenamen, die eine Windows-Domäne enthalten, wurde behoben.

2021.2-11190 — 11. Oktober 2021

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 254 • Makler: 413 • CLI: 54	• Es wurde ein Problem in der Befehlszeilenschnittstelle behoben, das das Starten von Windows-Sitzungen verhinderte.

2021.2-11042 — 01. September 2021

Build-Nummern	Neue Features	Änderungen und Fehlerbehebungen
• Makler: 254 • Makler: 413 • CLI: 37	• Amazon DCV Session Manager bietet jetzt Unterstützung für die Befehlszeilenschnittstelle (CLI). Sie können Amazon DCV-Sitzungen in der CLI erstellen und verwalten, anstatt sie aufzurufen APIs. • Amazon DCV Session Manager führte Broker-Datenpersistenz ein. Für eine höhere Verfügbarkeit können Broker Serverstatusinformationen in einem externen Datenspeicher speichern und die Daten beim Start wiederherstellen.	• Bei der Registrierung eines externen Autorisierungsservers können Sie jetzt den Algorithmus angeben, den der Autorisierungsserver zum Signieren von Web-Token im JSON-Format verwendet. Mit dieser Änderung können Sie Azure AD als externen Autorisierungsserver verwenden.

2021.1-10557 — 31. Mai 2021

Build-Nummern	Neue Features	Änderungen und Fehlerbehebungen
• Makler: 214 • Makler: 365	• Amazon DCV Session Manager hat Unterstützung für Eingabeparameter hinzugefügt, die an die Autorun-Datei unter Linux übergeben werden. • Servereigenschaften können jetzt als Anforderungen an die CreateSessionsAPI übergeben werden.	• Wir haben ein Problem mit der Autorun-Datei unter Windows behoben.

2021.0-10242 — 12. April 2021

Build-Nummern	Änderungen und Fehlerbehebungen

• Makler: 183
• Makler: 318
	• Amazon DCV Session Manager hat die folgenden Neuerungen APIs eingeführt:
	  • OpenServers
	  • CloseServers
	  • DescribeServers
	  • GetSessionScreenshots
	• Außerdem wurden die folgenden neuen Konfigurationsparameter eingeführt:
	  • Broker-Parameter: `session-screenshot-max-width session-screenshot-max-height ,session-screenshot-format ,create-sessions-queue-max-size , undcreate-sessions-queue-max-time-seconds .`
	  • Agentenparameter: `agent.autorun_folder max_virtual_sessions , undmax_concurrent_sessions_per_user .`
	
	  Agentenparameter: `agent.autorun_folder max_virtual_sessions , undmax_concurrent_sessions_per_user .`
	
	  Agentenparameter: `agent.autorun_folder max_virtual_sessions , undmax_concurrent_sessions_per_user .`

2020.2-9662 — 04. Dezember 2020

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 114 • Makler: 211	• Wir haben ein Problem mit den automatisch generierten TLS-Zertifikaten behoben, das den Start des Brokers verhinderte.

2020.2-9508 — 11. November 2020

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 78 • Makler: 183	• Die erste Version von Amazon DCV Session Manager.

Dokumentverlauf

In der folgenden Tabelle wird die Dokumentation für diese Version von Amazon DCV Session Manager beschrieben.

Änderungen	Beschreibung	Date
Amazon DCV Version 2025.0-544	Amazon DCV Session Manager wurde für Amazon DCV 2025.0-544 aktualisiert. Weitere Informationen finden Sie unter 2025.0-544 — 2. Februar 2026.	2. Februar 2026
Amazon DCV Version 2025.0-544	Amazon DCV Session Manager wurde für Amazon DCV 2025.0-544 aktualisiert. Weitere Informationen finden Sie unter 2025.0-544 — 23. Dezember 2025.	23. Dezember 2025
Amazon DCV Version 2025.0-539	Amazon DCV Session Manager wurde für Amazon DCV 2025.0-539 aktualisiert. Weitere Informationen finden Sie unter 2025.0-539 — 12. November 2025.	12. November 2025
Amazon DCV Version 2025.0-539	Amazon DCV Session Manager wurde für Amazon DCV 2025.0-539 aktualisiert. Weitere Informationen finden Sie unter 2025.0-539 — 22. Oktober 2025.	22. Oktober 2025

Änderungen	Beschreibung	Date
Amazon DCV-Version 2024.0-531	Amazon DCV Session Manager wurde für Amazon DCV 2024.0-531 aktualisiert. Weitere Informationen finden Sie unter 2024.0-531 — 17. Juni 2025.	17. Juni 2025
Amazon DCV version 2024.0-504	Amazon DCV Session Manager wurde für Amazon DCV 2024.0-504 aktualisiert. Weitere Informationen finden Sie unter 2024.0-504 — 31. März 2025.	31. März 2025
Amazon DCV version 2024.0-493	Amazon DCV Session Manager wurde für Amazon DCV 2024.0-493 aktualisiert. Weitere Informationen finden Sie unter 2024.0-493 — 15. Januar 2025.	15. Januar 2025
Amazon DCV version 2024.0-457	Amazon DCV Session Manager wurde für Amazon DCV 2024.0-457 aktualisiert. Weitere Informationen finden Sie unter 2024.0-457 — 1. Oktober 2024.	30. September 2024
Amazon DCV Version 2023.1-17652	Amazon DCV Session Manager wurde für Amazon DCV 2023.1-17652 aktualisiert. Weitere Informationen finden Sie unter 2023.1-17652 — 1. August 2024.	1. August 2024
Amazon DCV Version 2023.1-1638	Amazon DCV Session Manager wurde für Amazon DCV 2023.1-16388 aktualisiert. Weitere Informationen finden Sie unter 2023.1-16388 — 26. Juni 2024.	26. Juni 2024

Änderungen	Beschreibung	Date
Amazon DCV Version 2023.1	Amazon DCV Session Manager wurde für Amazon DCV 2023.1 aktualisiert. Weitere Informationen finden Sie unter 2023.1 — 9. November 2023.	9. November 2023
Amazon DCV versie 2023.0	Amazon DCV Session Manager wurde für Amazon DCV 2023.0 aktualisiert. Weitere Informationen finden Sie unter 2023.0-14852 — 28. März 2023.	28. März 2023
Amazon DCV Version 2022.2	Amazon DCV Session Manager wurde für Amazon DCV 2022.2 aktualisiert. Weitere Informationen finden Sie unter 2022.2-13907 — 11. November 2022.	11. November 2022
Amazon DCV Version 2022.1	Amazon DCV Session Manager wurde für Amazon DCV 2022.1 aktualisiert. Weitere Informationen finden Sie unter 2022.1-13067 — 29. Juni 2022.	29. Juni 2022
Amazon DCV versie 2022.0	Amazon DCV Session Manager wurde für Amazon DCV 2022.0 aktualisiert. Weitere Informationen finden Sie unter 2022.0-11952 — 23. Februar 2022.	23. Februar 2022
Amazon DCV Version 2021.3	Amazon DCV Session Manager wurde für Amazon DCV 2021.3 aktualisiert. Weitere Informationen finden Sie unter 2021.3-11591 — 20. Dezember 2021.	20. Dezember 2021
Amazon DCV versie 2021.2	Amazon DCV Session Manager wurde für Amazon DCV 2021.2 aktualisiert. Weitere Informationen finden Sie unter 2021.2-11042 — 01. September 2021.	01. September 2021

Änderungen	Beschreibung	Date
Amazon DCV version 2021.1	Amazon DCV Session Manager wurde für Amazon DCV 2021.1 aktualisiert. Weitere Informationen finden Sie unter 2021.1-10557 — 31. Mai 2021.	31. Mai 2021
Amazon DCV version 2021.0	Amazon DCV Session Manager wurde für Amazon DCV 2021.0 aktualisiert. Weitere Informationen finden Sie unter 2021.0-10242 — 12. April 2021.	12. April 2021
Erste Version von Amazon DCV Session Manager	Die erste Veröffentlichung dieses Inhalts.	11. November 2020

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.