aws

Guide de l’administrateur

Gestionnaire de sessions Amazon DCV

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

Gestionnaire de sessions Amazon DCV: Guide de l’administrateur

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

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

Table of Contents

Qu'est-ce que Session Manager ? 1
Comment fonctionne le gestionnaire de session 1
Fonctionnalités 3
Limitations 4
Tarification 4
Prérequis 4
Exigences en matière de réseau et de connectivité 6
Configuration du gestionnaire de session 8
Étape 1 : préparer les serveurs Amazon DCV 8
Étape 2 : configurer le courtier 9
Étape 3 : configurer l'agent 12
Étape 4 : Configuration du serveur Amazon DCV 19
Étape 5 : vérifier les installations 21
Vérifiez l'agent 22
Vérifiez le courtier 23
Configuration du gestionnaire de session 24
Gestionnaire de sessions de dimensionnement 24
Étape 1 : Créer un profil d'instance 25
Étape 2 : préparer le certificat SSL pour l'équilibreur de charge 26
Étape 3 : Création de l'équilibreur de charge de l'application Broker 27
Étape 4 : Lancez les courtiers 28
Étape 5 : Création de l'équilibreur de charge de l'application Agent 29
Étape 6 : Lancer les agents 31
Utilisation de balises sur les serveurs Amazon DCV 32
Configuration d'un serveur d'autorisation externe 33
Configuration de la persistance du broker 39
Configurer le broker pour qu'il persiste sur DynamoDB 40
Configurer le broker pour qu'il persiste sur MariaDB/MySQL 41
Intégration à la passerelle de connexion Amazon DCV 42
Configurer le Session Manager Broker en tant que résolveur de session pour la passerelle de connexion Amazon DCV 42
Facultatif : activez l'authentification du client TLS 43
Serveur Amazon DCV - Référence de mappage DNS 45
Intégration à Amazon CloudWatch 47

Mise à niveau du gestionnaire de session 49
Mise à niveau de l'agent Amazon DCV Session Manager 49
Mise à niveau du broker Amazon DCV Session Manager 53
Référence Broker CLI 56
register-auth-server 57
Syntaxe 57
Options 57
exemple 58
list-auth-servers 58
Syntaxe 57
Sortie 58
exemple 58
unregister-auth-server 59
Syntaxe 57
Options 57
Sortie 58
exemple 58
register-api-client 60
Syntaxe 57
Options 57
Sortie 58
exemple 58
describe-api-clients 62
Syntaxe 57
Sortie 58
exemple 58
unregister-api-client 63
Syntaxe 57
Options 57
exemple 58
renew-auth-server-api-clé 64
Syntaxe 57
exemple 58
generate-software-statement 65
Syntaxe 57
Sortie 58

exemple ... 58
describe-software-statements ... 66
Syntaxe ... 57
Sortie ... 58
exemple ... 58
deactivate-software-statement ... 67
Syntaxe ... 57
Options ... 57
exemple ... 58
describe-agent-clients ... 68
Syntaxe ... 57
Sortie ... 58
exemple ... 58
unregister-agent-client ... 70
Syntaxe ... 57
Options ... 57
exemple ... 58
register-server-dns-mappings ... 71
Syntaxe ... 57
Options ... 57
exemple ... 58
describe-server-dns-mappings ... 71
Syntaxe ... 57
Sortie ... 58
exemple ... 58
Référence du fichier de configuration ... 74
Fichier de configuration du broker ... 74
Fichier de configuration de l'agent ... 94
Notes de mise à jour et historique des documents ... 102
Notes de mise à jour ... 102
2025.0-561 — 25 juin 2026 ... 103
2025.0-544 — 2 février 2026 ... 103
2025.0-544 — 23 décembre 2025 ... 104
2025.0-539 — 12 novembre 2025 ... 104
2025.0-539 — 22 octobre 2025 ... 104
2024.0-531 — 17 juin 2025 ... 105

2024.0-504— 31 mars 2025 105
2024.0-493— 15 janvier 2025 105
2024.0-457— 1er octobre 2024 106
2023.1-17652 — 1er août 2024 106
2023.1-16388 — 26 juin 2024 106
2023.1 — 9 novembre 2023 107
2023.0-15065— 4 mai 2023 107
2023.0-14852 — 28 mars 2023 107
2022.2-13907 — 11 novembre 2022 107
2022.1-13067 — 29 juin 2022 108
2022.0-11952 — 23 février 2022 108
2021.3-11591 — 20 décembre 2021 108
2021.2-11445 — 18 novembre 2021 109
2021.2-11190 — 11 octobre 2021 109
2021.2-11042 — 1er septembre 2021 109
2021.1-10557 — 31 mai 2021 110
2021.0-10242 — 12 avril 2021 110
2020.2-9662 — 04 décembre 2020 111
..... 111
Historique de la documentation 111
..... cxvi

Qu'est-ce qu'Amazon DCV Session Manager ?

> **Note**
>
> Amazon DCV était auparavant connu sous le nom de NICE DCV.

Amazon DCV Session Manager est un ensemble de progiciels installables (un agent et un courtier) et une interface de programmation d'applications (API) qui permettent aux développeurs et aux fournisseurs de logiciels indépendants (ISVs) de créer facilement des applications frontales qui créent et gèrent par programmation le cycle de vie des sessions Amazon DCV sur un parc de serveurs Amazon DCV.

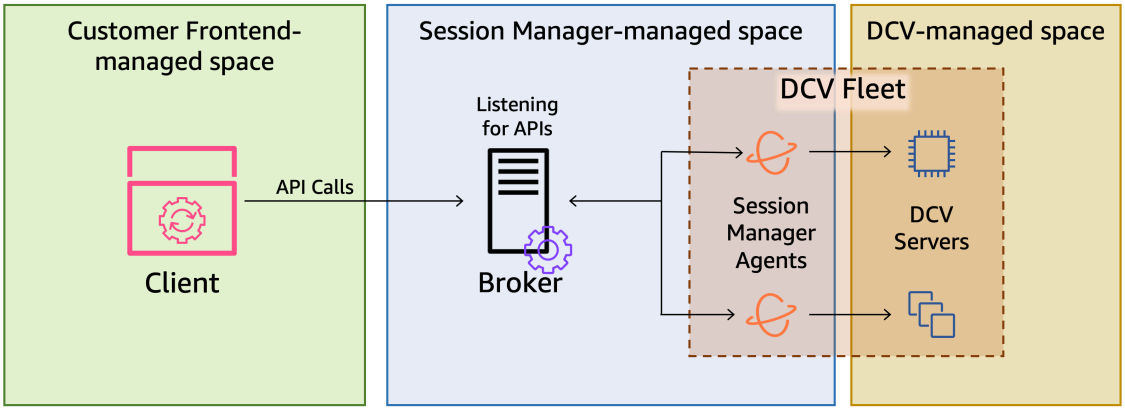
Ce guide explique comment installer et configurer l'agent et le courtier du gestionnaire de session. Pour plus d'informations sur l'utilisation du gestionnaire de session APIs, consultez le guide du développeur d'Amazon DCV Session Manager.

Rubriques

- Comment fonctionne le gestionnaire de session
- Fonctionnalités
- Limitations
- Tarification
- Exigences relatives au gestionnaire de sessions Amazon DCV

Comment fonctionne le gestionnaire de session

Le schéma suivant montre les composants de haut niveau du gestionnaire de session.



Agent

Le Broker est un serveur Web qui héberge et expose le gestionnaire APIs de session. Il reçoit et traite les demandes d'API pour gérer les sessions Amazon DCV provenant du client, puis transmet les instructions aux agents concernés. Le Broker doit être installé sur un hôte distinct de vos serveurs Amazon DCV, mais il doit être accessible au client et il doit pouvoir accéder aux agents.

Agent

L'agent est installé sur chaque serveur Amazon DCV de la flotte. Les agents reçoivent des instructions du courtier et les exécutent sur leurs serveurs Amazon DCV respectifs. Les agents surveillent également l'état des serveurs Amazon DCV et envoient des mises à jour périodiques au courtier.

APIs

Le gestionnaire de session expose un ensemble d'interfaces de programmation d'applications REST (APIs) qui peuvent être utilisées pour gérer les sessions Amazon DCV sur un parc de serveurs Amazon DCV. Ils APIs sont hébergés et exposés par le courtier. Les développeurs peuvent créer des clients de gestion de session personnalisés qui appellent le APIs.

Client

Le client est l'application frontale ou le portail que vous développez pour appeler le gestionnaire de session et APIsqui est exposé par le courtier. Les utilisateurs finaux utilisent le client pour gérer les sessions hébergées sur les serveurs Amazon DCV du parc.

Jeton d'accès

Pour effectuer une demande d'API, vous devez fournir un jeton d'accès. Les jetons peuvent être demandés au courtier ou à un serveur d'autorisation externe par le client enregistré APIs. Pour demander un jeton et y accéder, l'API client doit fournir des informations d'identification valides.

API client

L'API client est générée à partir du fichier YAML de définition de l'API Session Manager, à l'aide de Swagger Codegen. L'API client est utilisée pour effectuer des demandes d'API.

Séance Amazon DCV

Une session Amazon DCV est une période pendant laquelle le serveur Amazon DCV est en mesure d'accepter les connexions d'un client. Avant que vos clients puissent se connecter à une session Amazon DCV, vous devez créer une session Amazon DCV sur le serveur Amazon DCV. Amazon DCV prend en charge à la fois les sessions de console et les sessions virtuelles, et chaque session est associée à un propriétaire et à un ensemble d'autorisations spécifiques. Vous utilisez le gestionnaire de session APIs pour gérer le cycle de vie des sessions Amazon DCV. Les sessions Amazon DCV peuvent être dans l'un des états suivants :

- `CREATING`—Le courtier est en train de créer la session.
- `READY`: la session est prête à accepter les connexions des clients.
- `DELETING`: la session est en cours de suppression.
- `DELETED`: la session a été supprimée.
- `UNKNOWN`: impossible de déterminer l'état de la session. Le courtier et l'agent peuvent ne pas être en mesure de communiquer.

Fonctionnalités

Le gestionnaire de sessions DCV offre les fonctionnalités suivantes :

- Fournit des informations sur les sessions Amazon DCV : obtenez des informations sur les sessions exécutées sur plusieurs serveurs Amazon DCV.
- Gérez le cycle de vie de plusieurs sessions Amazon DCV : créez ou supprimez plusieurs sessions pour plusieurs utilisateurs sur plusieurs serveurs Amazon DCV avec une seule demande d'API.
- Supporte les balises : utilisez des balises personnalisées pour cibler un groupe de serveurs Amazon DCV lors de la création de sessions.

- Gère les autorisations pour plusieurs sessions Amazon DCV : modifiez les autorisations utilisateur pour plusieurs sessions avec une seule demande d'API.
- Fournit des informations de connexion : récupère les informations de connexion client pour les sessions Amazon DCV.
- Supports pour le cloud et sur site : utilisez le gestionnaire de session sur AWS, sur site ou avec d'autres serveurs basés sur le cloud.

Limitations

Le gestionnaire de session ne fournit pas de fonctionnalités de provisionnement des ressources. Si vous exécutez Amazon DCV sur des EC2 instances Amazon, vous devrez peut-être utiliser des AWS services supplémentaires, tels qu'Amazon EC2 Auto Scaling pour gérer le dimensionnement de votre infrastructure.

Tarification

Le gestionnaire de session est disponible gratuitement pour les AWS clients qui exécutent EC2 des instances.

Les clients sur site ont besoin d'une licence Amazon DCV Plus ou Amazon DCV Professional Plus. Pour plus d'informations sur l'achat d'une licence Amazon DCV Plus ou Amazon DCV Professional Plus, consultez Comment acheter sur le site Web Amazon DCV et trouvez un distributeur ou un revendeur Amazon DCV dans votre région. Pour permettre à tous les clients sur site de tester le gestionnaire de session Amazon DCV, les exigences de licence ne seront appliquées qu'à partir de la version 2021.0 d'Amazon DCV.

Pour plus d'informations, consultez la section Octroi de licences au serveur Amazon DCV dans le guide de l'administrateur Amazon DCV.

Exigences relatives au gestionnaire de sessions Amazon DCV

L'agent et le courtier Amazon DCV Session Manager répondent aux exigences suivantes.

	Broker	Agent

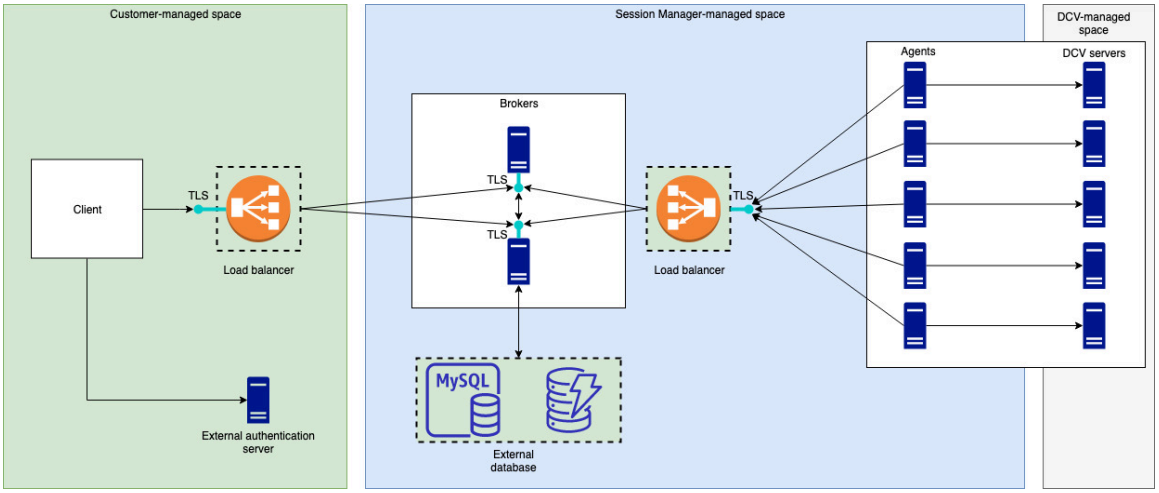
Système d'exploitation	• Amazon Linux 2
	• Amazon Linux 2023
	• CentOS Stream 9
	• RHEL 8.x
	• RHEL 9.x
	• Rocky Linux 8.5 ou version ultérieure
	• Rocky Linux 9.x
	• Ubuntu 22.04
	• Ubuntu 24.04
		• Windows
		  • Windows 11
		  • Windows Server 2025
		  • Windows Server 2022
		  • Windows Server 2019
		  • Windows Server 2016
		• Serveur Linux
		  • Amazon Linux 2
		  • Amazon Linux 2023
		  • CentOS Stream 9
		  • RHEL 8.x
		  • RHEL 9.x
		  • Rocky Linux 8.5 ou version ultérieure
		  • Rocky Linux 9.x
		  • Ubuntu 22.04
		  • Ubuntu 24.04
		  • SUSE Linux Enterprise 15 avec SP6 ou version ultérieure
		• macOS
		  • macOS 13 (Ventura)
		  • macOS 14 (Sonoma)
		  • macOS 15 (Sequoia)

Architecture	• 64 bits x86 • ARM 64 bits	• 64 bits x86 • ARM 64 bits (Amazon Linux 2, Amazon Linux 2023, CentOS 9.x,

	Broker	Agent
		RHEL 8.x/9.x, Rocky 8.x/9.x, Ubuntu 24.04/22.04 et macOS 13/14/15)
Mémoire	8 Go	4 Go
Version DCV d'Amazon	Amazon DCV 2020.2 et versions ultérieures	Amazon DCV 2020.2 et versions ultérieures
Exigences supplémentaires	Java 11	-

Exigences en matière de réseau et de connectivité

Le schéma suivant fournit un aperçu général des exigences de mise en réseau et de connectivité du gestionnaire de session.



Le Broker doit être installé sur un hôte distinct, mais il doit disposer d'une connectivité réseau avec les agents des serveurs Amazon DCV. Si vous choisissez d'avoir plusieurs courtiers pour améliorer la disponibilité, vous devez installer et configurer chaque courtier sur un hôte distinct, et utiliser un ou plusieurs équilibreurs de charge pour gérer le trafic entre le client et les courtiers, et entre les courtiers et les agents. Les courtiers devraient également être en mesure de communiquer entre eux afin d'échanger des informations sur les serveurs et les sessions Amazon DCV. Les courtiers peuvent stocker leurs clés et leurs données d'état dans une base de données externe et disposer de

ces informations après le redémarrage ou la résiliation. Cela permet d'atténuer le risque de perdre des informations importantes sur le courtier en les conservant dans la base de données externe. Vous pourrez le récupérer ultérieurement. Si vous choisissez de l'avoir, vous devez configurer la base de données externe et configurer les courtiers. DynamoDB, MariaDB et MySQL sont pris en charge. Les paramètres de configuration sont répertoriés dans le fichier de configuration du courtier.

Les agents doivent être en mesure d'établir des HTTPs connexions bidirectionnelles sécurisées et persistantes avec le courtier.

Votre client, ou application frontale, doit être en mesure d'accéder au courtier pour pouvoir appeler le APIs. Le client doit également être en mesure d'accéder à votre serveur d'authentification.

Configuration d'Amazon DCV Session Manager

La section suivante explique comment installer le gestionnaire de session avec un seul broker et plusieurs agents. Vous pouvez utiliser plusieurs courtiers pour améliorer l'évolutivité et les performances. Pour de plus amples informations, veuillez consulter Gestionnaire de sessions de dimensionnement.

Pour configurer Amazon DCV Session Manager, procédez comme suit :

Étapes

- Étape 1 : préparer les serveurs Amazon DCV
- Étape 2 : configurer le broker Amazon DCV Session Manager
- Étape 3 : configurer l'agent Amazon DCV Session Manager
- Étape 4 : configurer le serveur Amazon DCV pour utiliser le broker comme serveur d'authentification
- Étape 5 : vérifier les installations

Étape 1 : préparer les serveurs Amazon DCV

Vous devez disposer d'un parc de serveurs Amazon DCV avec lesquels vous avez l'intention d'utiliser Session Manager. Pour plus d'informations sur l'installation des serveurs Amazon DCV, consultez la section Installation du serveur Amazon DCV dans le manuel Amazon DCV Administrator Guide.

Sur les serveurs Amazon DCV Linux et macOS, le gestionnaire de session utilise un nom `dcvsmagent` d'utilisateur du service local. Cet utilisateur est automatiquement créé lors de l'installation de l'agent Session Manager. Vous devez accorder à ce service des privilèges d'administrateur pour Amazon DCV afin qu'il puisse effectuer des actions pour le compte d'autres utilisateurs. Pour accorder des privilèges d'administrateur à l'utilisateur du service Session Manager, procédez comme suit :

Pour ajouter l'utilisateur du service local pour les serveurs Amazon DCV Linux et macOS

1. Ouvrez `/etc/dcv/dcv.conf` à l'aide de votre éditeur de texte préféré.
2. Ajoutez le `administrators` paramètre à la `[security]` section et spécifiez l'utilisateur du gestionnaire de session. Par exemple :

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

3. Enregistrez et fermez le fichier .
4. Arrêtez et redémarrez le serveur Amazon DCV.

Le gestionnaire de session est uniquement en mesure de créer des sessions Amazon DCV pour le compte d'utilisateurs qui existent déjà sur le serveur Amazon DCV. Si une demande est faite pour créer une session pour un utilisateur qui n'existe pas, la demande échoue. Par conséquent, vous devez vous assurer que chaque utilisateur final prévu dispose d'un utilisateur système valide sur le serveur Amazon DCV.

> **Tip**
>
> Si vous avez l'intention d'utiliser plusieurs hôtes de courtage ou des serveurs Amazon DCV avec des agents, nous vous recommandons de configurer un seul courtier et un seul serveur Amazon DCV avec un agent en effectuant les étapes suivantes, en créant des Amazon Machine Images (AMI) des hôtes avec les configurations complètes, puis en les utilisant AMIs pour lancer les autres courtiers et serveurs Amazon DCV. Vous pouvez également utiliser AWS Systems Manager pour exécuter les commandes sur plusieurs instances à distance.

Étape 2 : configurer le broker Amazon DCV Session Manager

Le broker doit être installé sur un hôte Linux. Pour plus d'informations sur les distributions Linux prises en charge, consultezExigences relatives au gestionnaire de sessions Amazon DCV. Installez le broker sur un hôte distinct de l'agent et de l'hôte du serveur Amazon DCV. L'hôte peut être installé sur un autre réseau privé, mais il doit être en mesure de se connecter à l'agent et de communiquer avec celui-ci.

Pour installer et démarrer le broker

1. Connectez-vous à l'hôte sur lequel vous souhaitez installer le broker.
2. Les packages du sont signés numériquement avec une signature GPG sécurisée. Pour permettre au gestionnaire de package de vérifier la signature du package, vous devez importer la clé GPG Amazon DCV. Exécutez la commande suivante pour importer la clé Amazon DCV GPG.

- Amazon Linux 2, RHEL, CentOS et 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. Téléchargez le package d'installation.

- 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 et 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 et 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. Installez le package .

- 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 et 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 et 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. Vérifiez que la version de l'environnement Java par défaut est 11

   ```
   $ java -version
   ```

   Si ce n'est pas le cas, vous pouvez définir explicitement le répertoire de base Java que le broker utilisera pour cibler la bonne version de Java. Cela se fait en définissant le paramètre `broker-java-home` dans le fichier de configuration du courtier. Pour plus d'informations, consultez le fichier de configuration du broker.

6. Démarrez le service de courtage et assurez-vous qu'il démarre automatiquement à chaque démarrage de l'instance.

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

7. Placez une copie du certificat autosigné du courtier dans votre répertoire d'utilisateurs. Vous en aurez besoin lors de l'installation des agents à l'étape suivante.

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

Étape 3 : configurer l'agent Amazon DCV Session Manager

L'agent doit être installé sur tous les hôtes du serveur Amazon DCV du parc. L'agent peut être installé sur les hôtes Windows, Linux et macOS. Pour plus d'informations sur les systèmes d'exploitation pris en charge, consultezExigences relatives au gestionnaire de sessions Amazon DCV.

Prérequis

Le serveur Amazon DCV doit être installé sur l'hôte avant d'installer l'agent.

Linux host

> **Note**
>
> L'agent Session Manager est disponible pour les distributions et architectures Linux répertoriées dans Exigences :

Les instructions suivantes concernent l'installation de l'agent sur des hôtes x86 64 bits. Pour installer l'agent sur des hôtes ARM 64 bits, remplacez *x86_64* par `aarch64`. Pour Ubuntu, remplacez *amd64* par `arm64`.

Pour installer l'agent sur un hôte Linux

1. Les packages du sont signés numériquement avec une signature GPG sécurisée. Pour permettre au gestionnaire de package de vérifier la signature du package, vous devez importer la clé GPG Amazon DCV. Exécutez la commande suivante pour importer la clé Amazon DCV GPG.

   - Amazon Linux 2, RHEL, CentOS et 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. Téléchargez le package d'installation.

   - 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 et 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 et 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. Installez le package .

- 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 et 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 et 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. Placez une copie du certificat autosigné du courtier (que vous avez copié à l'étape précédente) dans le `/etc/dcv-session-manager-agent/` répertoire de l'agent.
5. Ouvrez `/etc/dcv-session-manager-agent/agent.conf` à l'aide de votre éditeur de texte préféré et procédez comme suit.

   - Pour`broker_host`, spécifiez le nom DNS de l'hôte sur lequel le broker est installé.

> **Important**
>
> Si le broker s'exécute sur une EC2 instance Amazon, `broker_host` vous devez spécifier l'adresse IPv4 privée de l'instance.

- (Facultatif) Pour`broker_port`, spécifiez le port par lequel communiquer avec le courtier. Par défaut, l'agent et le broker communiquent via le port`8445`. Ne modifiez cette option que si vous devez utiliser un autre port. Si vous le modifiez, assurez-vous que le broker est configuré pour utiliser le même port.
- Pour`ca_file`, spécifiez le chemin complet du fichier de certificat que vous avez copié à l'étape précédente. Par exemple :

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

Sinon, si vous souhaitez désactiver la vérification TLS, définissez le paramètre sur`tls_strict`. `false`

6. Enregistrez et fermez le fichier .
7. Exécutez la commande suivante pour démarrer l'agent.

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

Windows host

Pour installer l'agent sur un hôte Windows

1. Téléchargez le programme d'installation de l'agent.
2. Exécutez le programme d'installation. Sur l'écran d'accueil, choisissez Next.
3. Sur l'écran EULA, lisez attentivement le contrat de licence et, si vous êtes d'accord, sélectionnez J'accepte les termes, puis Next.
4. Pour commencer l'installation, choisissez Installer.
5. Placez une copie du certificat autosigné du courtier (que vous avez copié à l'étape précédente) dans le `C:\Program Files\NICE\DCVSessionManagerAgent\conf\` dossier de l'agent.

6. Ouvrez `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` à l'aide de votre éditeur de texte préféré, puis effectuez les opérations suivantes :

   - Pour`broker_host`, spécifiez le nom DNS de l'hôte sur lequel le broker est installé.

     > **Important**
     >
     > Si le broker s'exécute sur une EC2 instance Amazon, `broker_host` vous devez spécifier l' IPv4 adresse privée de l'instance.

   - (Facultatif) Pour`broker_port`, spécifiez le port par lequel communiquer avec le courtier. Par défaut, l'agent et le broker communiquent via le port`8445`. Ne modifiez cette option que si vous devez utiliser un autre port. Si vous le modifiez, assurez-vous que le broker est configuré pour utiliser le même port.
   - Pour`ca_file`, spécifiez le chemin complet du fichier de certificat que vous avez copié à l'étape précédente. Par exemple :

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

     Sinon, si vous souhaitez désactiver la vérification TLS, définissez le paramètre sur`tls_strict`. `false`

7. Enregistrez et fermez le fichier .
8. Arrêtez et redémarrez le service d'agent pour que les modifications prennent effet. Exécutez les commandes suivantes à l'invite de commandes.

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

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

macOS host

Pour installer l'agent sur un hôte macOS

1. Téléchargez le programme d'installation de l'agent ou utilisez la ligne de commande :

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

2. Installez le package . Vous pouvez utiliser la ligne de commande ou via le programme d'installation de l'agent :

   - Ligne de commande :

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

   - Programme d'installation de l'agent : double-cliquez sur le `.pkg` fichier et suivez les instructions de l'assistant d'installation.

3. Placez une copie du certificat autosigné du courtier (que vous avez copié à l'étape précédente) dans le `/etc/dcv-session-manager-agent/` répertoire de l'agent.

4. Ouvrez `/etc/dcv-session-manager-agent/agent.conf` à l'aide de votre éditeur de texte préféré et procédez comme suit.

   - Pour`broker_host`, spécifiez le nom DNS de l'hôte sur lequel le broker est installé.

     > **Important**
     >
     > Si le broker s'exécute sur une EC2 instance Amazon, `broker_host` vous devez spécifier l' IPv4 adresse privée de l'instance.

   - (Facultatif) Pour`broker_port`, spécifiez le port par lequel communiquer avec le courtier. Par défaut, l'agent et le broker communiquent via le port`8445`. Ne modifiez cette option que si vous devez utiliser un autre port. Si vous le modifiez, assurez-vous que le broker est configuré pour utiliser le même port.

   - Pour`ca_file`, spécifiez le chemin complet du fichier de certificat que vous avez copié à l'étape précédente. Par exemple :

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

     Sinon, si vous souhaitez désactiver la vérification TLS, définissez le paramètre sur`tls_strict`. `false`

5. Enregistrez et fermez le fichier .

6. Démarrez le service d'agent.

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

Étape 4 : configurer le serveur Amazon DCV pour utiliser le broker comme serveur d'authentification

Configurez le serveur Amazon DCV pour utiliser le broker comme serveur d'authentification externe pour valider les jetons de connexion client. Vous devez également configurer le serveur Amazon DCV pour qu'il fasse confiance à l'autorité de certification autosignée du courtier.

Linux Amazon DCV server

Pour ajouter l'utilisateur du service local pour les serveurs Linux Amazon DCV

1. Ouvrez `/etc/dcv/dcv.conf` à l'aide de votre éditeur de texte préféré.
2. Ajoutez les `auth-token-verifier` paramètres `ca-file` et à la `[security]` section.

   - Pour`ca-file`, spécifiez le chemin d'accès à l'autorité de certification autosignée du courtier que vous avez copiée sur l'hôte à l'étape précédente.
   - Pour`auth-token-verifier`, spécifiez l'URL du vérificateur de jetons sur le courtier au format suivant :`https://`*`broker_ip_or_dns`*`:`*`port`*`/agent/validate-authentication-token`. Spécifiez le port utilisé pour la communication courtier-agent, qui est 8445 par défaut. Si vous exécutez le broker sur une EC2 instance Amazon, vous devez utiliser le DNS privé ou l'adresse IP privée.

     Par exemple

     ```
     [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. Enregistrez et fermez le fichier .

4. Arrêtez et redémarrez le serveur Amazon DCV. Pour plus d'informations, consultez les sections Stopping the Amazon DCV Server et Starting the Amazon DCV Server dans le manuel Amazon DCV Administrator Guide.

Windows Amazon DCV server

Sur les serveurs Windows Amazon DCV

1. Ouvrez l'éditeur de registre Windows et accédez à la touche USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/security HKEY_/.
2. Ouvrez le paramètre ca-file.
3. Pour les données de valeur, spécifiez le chemin d'accès à l'autorité de certification autosignée du courtier que vous avez copiée sur l'hôte à l'étape précédente.

> **Note**
>
> Si le paramètre n'existe pas, créez un nouveau paramètre de chaîne et nommez-le `ca-file`.

4. Ouvrez le auth-token-verifierparamètre.
5. Pour les données de valeur, spécifiez l'URL du vérificateur de jetons sur le courtier au format suivant : `https://`*`broker_ip_or_dns`*`:`*`port`*`/agent/validate-authentication-token`
6. Spécifiez le port utilisé pour la communication courtier-agent, qui est 8445 par défaut. Si vous exécutez le broker sur une EC2 instance Amazon, vous devez utiliser le DNS privé ou l'adresse IP privée.

> **Note**
>
> Si le paramètre n'existe pas, créez un nouveau paramètre de chaîne et nommez-le `auth-token-verifier`.

7. Choisissez OK, puis fermez l'Éditeur du Registre Windows.
8. Arrêtez et redémarrez le serveur Amazon DCV. Pour plus d'informations, consultez les sections Stopping the Amazon DCV Server et Starting the Amazon DCV Server dans le manuel Amazon DCV Administrator Guide.

macOS Amazon DCV server

Pour ajouter l'utilisateur du service local pour macOS, Amazon DCV

1. Ouvrez `/etc/dcv/dcv.conf` à l'aide de votre éditeur de texte préféré.
2. Ajoutez les `auth-token-verifier` paramètres `ca-file` et à la `[security]` section.
   - Pour`ca-file`, spécifiez le chemin d'accès à l'autorité de certification autosignée du courtier que vous avez copiée sur l'hôte à l'étape précédente.
   - Pour`auth-token-verifier`, spécifiez l'URL du vérificateur de jetons sur le courtier au format suivant :`https://`*`broker_ip_or_dns`*`:`*`port`*`/agent/validate-authentication-token`. Spécifiez le port utilisé pour la communication courtier-agent, qui est 8445 par défaut. Si vous exécutez le broker sur une EC2 instance Amazon, vous devez utiliser le DNS privé ou l'adresse IP privée.

     Par exemple

     ```
     [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. Enregistrez et fermez le fichier .
4. Arrêtez et redémarrez le serveur Amazon DCV. Pour plus d'informations, consultez les sections Stopping the Amazon DCV Server et Starting the Amazon DCV Server dans le manuel Amazon DCV Administrator Guide.

Étape 5 : vérifier les installations

Après avoir configuré l'agent, configuré le broker et configuré les deux sur le serveur Amazon DCV, vous devez vérifier que les installations fonctionnent correctement.

Rubriques

- Vérifiez l'agent
- Vérifiez le courtier

Vérifiez l'agent

Après avoir installé le broker et l'agent, assurez-vous que l'agent est en cours d'exécution et qu'il est capable de se connecter au broker.

Hôtes d'agents Linux et macOS

La commande à exécuter dépend de la version.

- Depuis la version 2022.0

  Depuis l'hôte de l'agent, exécutez la commande suivante :

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

- Versions antérieures à 2022.0

  Depuis l'hôte de l'agent, exécutez la commande suivante et spécifiez l'année, le mois et le jour en cours.

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

  Par exemple

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

Si l'agent est en cours d'exécution et qu'il est capable de se connecter au broker, la commande doit être renvoyée`success`.

Si la commande renvoie une sortie différente, consultez le fichier journal de l'agent pour plus d'informations. Les fichiers journaux se trouvent ici : `/var/log/dcv-session-manager-agent/`.

Hôte de l'agent Windows

Ouvrez le fichier journal de l'agent, qui se trouve dans`C:\ProgramData\NICE \DCVSessionManagerAgent\log`.

Si le fichier journal inclut une ligne similaire à celle ci-dessous, l'agent est en cours d'exécution et peut se connecter au courtier.

```
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}"
```

Si votre fichier journal ne comporte pas de ligne similaire, inspectez-le pour détecter d'éventuelles erreurs.

Vérifiez le courtier

Après avoir installé le broker et l'agent, assurez-vous que votre broker est en cours d'exécution et qu'il est accessible depuis vos utilisateurs et vos applications frontales.

À partir d'un ordinateur qui devrait pouvoir accéder au broker, exécutez la commande suivante :

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

Si la vérification est réussie, le courtier renvoie les informations suivantes :

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

Configuration du gestionnaire de session Amazon DCV

Pour offrir une expérience fluide et sécurisée, il est important de configurer correctement le gestionnaire de session en fonction des besoins et des exigences de votre organisation. Cette section décrit les principales étapes de l'installation et de la configuration du gestionnaire de session, notamment la gestion de l'accès des utilisateurs, la configuration des paramètres réseau et la personnalisation des paramètres de session.

Rubriques

- Gestionnaire de sessions de dimensionnement
- Utilisation de balises pour cibler les serveurs Amazon DCV
- Configuration d'un serveur d'autorisation externe
- Configuration de la persistance du broker
- Intégration à la passerelle de connexion Amazon DCV
- Intégration à Amazon CloudWatch

Gestionnaire de sessions de dimensionnement

Pour garantir une haute disponibilité et améliorer les performances, vous pouvez configurer le gestionnaire de session pour qu'il utilise plusieurs agents et courtiers. Si vous avez l'intention d'utiliser plusieurs agents et courtiers, nous vous recommandons d'installer et de configurer un seul hôte agent et courtier, de créer des images Amazon Machines (AMI) à partir de ces hôtes, puis de lancer les hôtes restants à partir du AMIs.

Par défaut, le gestionnaire de session prend en charge l'utilisation de plusieurs agents sans configuration supplémentaire. Toutefois, si vous avez l'intention d'utiliser plusieurs courtiers, vous devez utiliser un équilibreur de charge pour équilibrer le trafic entre le client frontal et les courtiers, et entre les courtiers et les agents. L'installation et la configuration de l'équilibreur de charge sont entièrement détenues et gérées par vous.

La section suivante explique comment configurer le gestionnaire de session pour utiliser plusieurs hôtes avec un Application Load Balancer.

Étapes

- Étape 1 : Créer un profil d’instance

- Étape 2 : préparer le certificat SSL pour l'équilibreur de charge
- Étape 3 : Création de l'équilibreur de charge de l'application Broker
- Étape 4 : Lancez les courtiers
- Étape 5 : Création de l'équilibreur de charge de l'application Agent
- Étape 6 : Lancer les agents

Étape 1 : Créer un profil d’instance

Vous devez associer un profil d'instance aux hôtes Broker et Agent qui leur donne l'autorisation d'utiliser Elastic Load Balancing APIs. Pour de plus amples informations sur les rôles IAM, consultez Rôles IAM pour Amazon EC2 dans le Guide de l’utilisateur Amazon EC2.

Pour créer un profil d’instance

1. Créez un rôle Gestion des identités et des accès AWS (IAM) qui définit les autorisations à utiliser dans le profil d'instance. Utilisez la politique de confiance suivante :

   JSON

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

   Joignez ensuite la politique suivante :

   JSON

   ```
   {
   ```

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

Pour plus d'informations, consultez la section Création d'un rôle IAM dans le Guide de l'utilisateur IAM.

2. Créez un nouveau profil d'instance. Pour plus d'informations, consultez create-instance-profile dans la Référence des commandes de l'AWS CLI .
3. Ajoutez le rôle IAM au profil d'instance. Pour plus d'informations, consultez add-role-to-instance-profile dans le manuel de référence des AWS CLI commandes.
4. Attachez le profil d'instance aux hôtes Broker. Pour plus d'informations, consultez la section Attacher un rôle IAM à une instance dans le guide de l'utilisateur Amazon EC2.

Étape 2 : préparer le certificat SSL pour l'équilibreur de charge

Lorsque vous utilisez HTTPS pour votre écouteur d'équilibreur de charge, vous devez déployer un certificat SSL sur votre équilibreur de charge. L'équilibreur de charge utilise ce certificat pour mettre fin à la connexion et déchiffrer les demandes des clients avant de les envoyer aux cibles.

Pour préparer le certificat SSL

1. Créez une autorité de certification privée (CA) AWS Certificate Manager Private Certificate Authority (ACM PCA). Pour plus d'informations, consultez la section Procédures de création

d'une autorité de certification dans le AWS guide de l'utilisateur de Certificate Manager Private Certificate Authority.

2. Installez le CA. Pour plus d'informations, consultez la section Installation d'un certificat CA racine dans le guide de l'utilisateur de AWS Certificate Manager Private Certificate Authority.
3. Demandez un nouveau certificat privé signé par l'autorité de certification. Pour le nom de domaine, utilisez `*.region.elb.amazonaws.com` et spécifiez la région dans laquelle vous souhaitez créer l'équilibreur de charge. Pour plus d'informations, consultez la section Requering a Private Certificate dans le AWS guide de l'utilisateur de Certificate Manager Private Certificate Authority.

Étape 3 : Création de l'équilibreur de charge de l'application Broker

Créez un équilibreur de charge d'application pour équilibrer le trafic entre vos clients frontaux et les courtiers.

Pour créer l'équilibreur de charge

1. Ouvrez la console Amazon EC2 à l'adresse https://console.aws.amazon.com/ec2/.

   Dans le volet de navigation, choisissez Load Balancers, puis Create Load Balancer. Pour le type d'équilibreur de charge, choisissez Application Load Balancer.
2. Pour Étape 1 : Configurer l'équilibreur de charge, procédez comme suit :

   a. Dans Nom, entrez un nom descriptif pour l'équilibreur de charge.
   b. Pour Schéma, sélectionnez connecté à Internet.
   c. Pour Load Balancer Protocol, sélectionnez HTTPS, et pour Load Balancer Port, entrez. `8443`
   d. Pour le VPC, sélectionnez le VPC à utiliser, puis sélectionnez tous les sous-réseaux de ce VPC.
   e. Choisissez Suivant.
3. Pour l'étape 2 : Configuration des paramètres de sécurité, procédez comme suit :

   a. Pour Type de certificat, choisissez Choisir un certificat auprès d'ACM.
   b. Dans Nom du certificat, sélectionnez le certificat privé que vous avez demandé précédemment.
   c. Choisissez Suivant.

4. Pour l'étape 3 : configurer les groupes de sécurité, créer un nouveau groupe de sécurité ou sélectionner un groupe de sécurité existant qui autorise le trafic entrant et sortant entre votre client frontal et les courtiers via HTTPS et le port 8443.

   Choisissez Suivant.

5. Pour l'étape 4 : Configuration du routage, procédez comme suit :

   a. Pour Groupe cible, sélectionnez Nouveau groupe cible.

   b. Dans Nom, entrez un nom pour le groupe cible.

   c. Pour Type de cible, choisissez Instance.

   d. Pour Protocole, sélectionnez HTTPS. Pour Port, entrez `8443`. Pour la version du protocole, choisissez HTTP1.

   e. Pour le protocole de vérification de l'état, choisissez HTTPS, et pour Path, entrez `/health`.

   f. Choisissez Suivant.

6. Pour l'étape 5 : Enregistrer les cibles, choisissez Next.

7. Choisissez Créer.

Étape 4 : Lancez les courtiers

Créez un Broker initial et configurez-le pour utiliser l'équilibreur de charge, créez une AMI à partir du Broker, puis utilisez l'AMI pour lancer les Brokers restants. Cela garantit que tous les courtiers sont configurés pour utiliser la même autorité de certification et la même configuration d'équilibreur de charge.

Pour lancer les Brokers

1. Lancez et configurez l'hôte Broker initial. Pour plus d'informations sur l'installation et la configuration du Broker, consultezÉtape 2 : configurer le broker Amazon DCV Session Manager.

   > **Note**
   >
   > Le certificat auto-signé du courtier n'est pas nécessaire car nous utilisons un équilibreur de charge d'application.

2. Connectez-vous au courtier, ouvrez-le à `/etc/dcv-session-manager-broker/session-manager-broker.properties` l'aide de votre éditeur de texte préféré et procédez comme suit :

   a. Commentez le `broker-to-broker-discovery-addresses` paramètre en plaçant un hachage (#) au début de la ligne.

   b. Pour`broker-to-broker-discovery-aws-region`, entrez la région dans laquelle vous avez créé l'équilibreur de charge de l'application.

   c. Pour`broker-to-broker-discovery-aws-alb-target-group-arn`, entrez l'ARN du groupe cible associé à l'équilibreur de charge Broker.

   d. Enregistrez et fermez le fichier .

3. Arrêtez l'instance Broker.

4. Créez une AMI à partir de l'instance de Broker arrêtée. Pour plus d'informations, consultez la section Création d'une AMI Linux à partir d'une instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

5. Utilisez l'AMI pour lancer les Brokers restants.

6. Attribuez le profil d'instance que vous avez créé à toutes les instances de Broker.

7. Attribuez un groupe de sécurité qui permet à Broker to Broker et Broker d'équilibrer la charge du trafic réseau vers toutes les instances du courtier. Pour plus d'informations sur les ports réseau, consultez le fichier de configuration du courtier.

8. Enregistrez toutes les instances de Broker en tant que cibles pour l'équilibreur de charge Broker. Pour plus d'informations, consultez la section Enregistrer des cibles auprès de votre groupe cible dans le Guide de l'utilisateur des équilibreurs de charge d'application.

Étape 5 : Création de l'équilibreur de charge de l'application Agent

Créez un équilibreur de charge d'application pour équilibrer les agents et les courtiers.

Pour créer l'équilibreur de charge

1. Ouvrez la console Amazon EC2 à l'adresse https://console.aws.amazon.com/ec2/.

   Dans le volet de navigation, choisissez Load Balancers, puis Create Load Balancer. Pour le type d'équilibreur de charge, choisissez Application Load Balancer.

2. Pour Étape 1 : Configurer l'équilibreur de charge, procédez comme suit :

a. Dans Nom, entrez un nom descriptif pour l'équilibreur de charge.

b. Pour Schéma, sélectionnez connecté à Internet.

c. Pour Load Balancer Protocol, sélectionnez HTTPS, et pour Load Balancer Port, entrez. `8445`

d. Pour le VPC, sélectionnez le VPC à utiliser, puis sélectionnez tous les sous-réseaux de ce VPC.

e. Choisissez Suivant.

3. Pour l'étape 2 : Configuration des paramètres de sécurité, procédez comme suit :

a. Pour Type de certificat, choisissez Choisir un certificat auprès d'ACM.

b. Dans Nom du certificat, sélectionnez le certificat privé que vous avez demandé précédemment.

c. Choisissez Suivant.

4. Pour l'étape 3 : configurer les groupes de sécurité, créer un nouveau groupe de sécurité ou sélectionner un groupe de sécurité existant qui autorise le trafic entrant et sortant des agents et des courtiers via HTTPS et le port 8445.

   Choisissez Suivant.

5. Pour l'étape 4 : Configuration du routage, procédez comme suit :

a. Pour Groupe cible, sélectionnez Nouveau groupe cible.

b. Dans Nom, entrez un nom pour le groupe cible.

c. Pour Type de cible, choisissez Instance.

d. Pour Protocole, sélectionnez HTTPS. Pour Port, entrez `8445`. Pour la version du protocole, choisissez HTTP1.

e. Pour le protocole de vérification de l'état, choisissez HTTPS, et pour Path, entrez`/health`.

f. Choisissez Suivant.

6. Pour l'étape 5 : Enregistrer les cibles, sélectionnez toutes les instances de courtier et choisissez Ajouter aux instances enregistrées. Choisissez Suivant : Vérification.

7. Choisissez Créer.

Étape 6 : Lancer les agents

Créez un agent initial et configurez-le pour utiliser l'équilibreur de charge, créez une AMI à partir de l'agent, puis utilisez l'AMI pour lancer les agents restants. Cela garantit que tous les agents sont configurés pour utiliser la même configuration d'équilibreur de charge.

Pour lancer les agents

1. Préparez le serveur Amazon DCV. Pour de plus amples informations, veuillez consulter Étape 1 : préparer les serveurs Amazon DCV.
2. Placez une copie de la clé publique CA créée dansÉtape 2 : préparer le certificat SSL pour l'équilibreur de charge. Choisissez ou créez un répertoire lisible par n'importe quel utilisateur. Le fichier de clé publique de l'autorité de certification doit également être lisible par tous les utilisateurs.
3. Installez et configurez l'agent. Pour plus d'informations sur l'installation et la configuration de l'agent, consultezÉtape 3 : configurer l'agent Amazon DCV Session Manager.

   > **Important**
   >
   > Lors de la modification du fichier de configuration de l'agent :
   >
   > - pour le `broker_host` paramètre, entrez le DNS de l'équilibreur de charge de l'agent
   > - pour le `ca_file` paramètre, entrez le chemin d'accès au fichier de clé publique CA créé à l'étape précédente

4. Configurez le serveur Amazon DCV pour utiliser le Broker comme serveur d'authentification. Pour de plus amples informations, veuillez consulter Étape 4 : configurer le serveur Amazon DCV pour utiliser le broker comme serveur d'authentification.

   > **Important**
   >
   > Lorsque vous modifiez le fichier de configuration du serveur Amazon DCV :
   >
   > - pour le `ca-file` paramètre, entrez le même chemin vers le fichier de clé publique CA utilisé à l'étape précédente
   > - pour le `auth-token-verifier` paramètre, utilisez le DNS de l'équilibreur de charge de l'agent pour *`broker_ip_or_dns`*

5. Arrêtez l'instance de l'agent.
6. Créez une AMI à partir de l'instance d'agent arrêtée. Pour plus d'informations, consultez la section Création d'une AMI Linux à partir d'une instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.
7. Utilisez l'AMI pour lancer les agents restants et attribuez le profil d'instance que vous avez créé à chacun d'entre eux.
8. Attribuez un groupe de sécurité qui permet à l'agent d'équilibrer la charge du trafic réseau vers toutes les instances de l'agent. Pour plus d'informations sur les ports réseau, consultez le fichier de configuration de l'agent.

Utilisation de balises pour cibler les serveurs Amazon DCV

Vous pouvez attribuer des balises personnalisées aux agents Session Manager afin de les identifier et de les classer, ainsi que les serveurs Amazon DCV auxquels ils sont associés. Lorsque vous créez une nouvelle session Amazon DCV, vous pouvez cibler un groupe de serveurs Amazon DCV en fonction des balises attribuées à leurs agents respectifs. Pour plus d'informations sur la manière de cibler les serveurs Amazon DCV en fonction des balises d'agent, consultez CreateSessionRequests le guide du développeur de Session Manager.

Une balise se compose d'une paire clé-valeur de balise, et vous pouvez utiliser n'importe quelle paire d'informations adaptée à votre cas d'utilisation ou à votre environnement. Vous pouvez choisir de baliser les agents en fonction de la configuration matérielle de leur hôte. Par exemple, vous pouvez étiqueter tous les agents dotés d'hôtes dotés de 4 Go de mémoire`ram=4GB`. Vous pouvez également étiqueter les agents en fonction de leur objectif. Par exemple, vous pouvez étiqueter tous les agents exécutés sur les hôtes de production avec`purpose=production`.

Pour attribuer des tags à un agent

1. À l'aide de votre éditeur de texte préféré, créez un nouveau fichier et donnez-lui un nom descriptif, par exemple`agent_tags.toml`. Le type de fichier doit être`.toml`, et le contenu du fichier doit être spécifié au format de fichier TOML.
2. Dans le fichier, ajoutez chaque nouvelle paire clé-valeur de balise sur une nouvelle ligne en utilisant le `key=value` format. Par exemple :

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

3. Ouvrez le fichier de configuration de l'agent (`/etc/dcv-session-manager-agent/agent.conf`pour Linux et macOS ou `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` pour Windows). Pour`tags_folder`, et spécifiez le chemin d'accès au répertoire dans lequel se trouve le fichier de balises.

   Si le répertoire contient plusieurs fichiers de balises, toutes les balises définies dans les fichiers appliquent l'agent. Les fichiers sont lus par ordre alphabétique. Si plusieurs fichiers contiennent une balise avec la même clé, la valeur est remplacée par la valeur du dernier fichier lu.

4. Enregistrez et fermez le fichier .
5. Arrêtez et redémarrez l'agent.

   - 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
     ```

Configuration d'un serveur d'autorisation externe

Le serveur d'autorisation est le serveur chargé d'authentifier et d'autoriser le client SDKs et les agents.

Par défaut, le gestionnaire de session utilise le courtier comme serveur d'autorisation pour générer des jetons d'accès OAuth 2.0 pour les instructions client SDKs et logicielle pour les agents. Si vous utilisez le Broker comme serveur d'autorisation, aucune configuration supplémentaire n'est requise.

Vous pouvez configurer le gestionnaire de session pour utiliser Amazon Cognito comme serveur d'autorisation externe au lieu du Broker. Pour plus d'informations sur Amazon Cognito, consultez le guide du développeur Amazon Cognito.

Pour utiliser Amazon Cognito comme serveur d'autorisation

1. Créez un nouveau groupe d'utilisateurs Amazon Cognito. Pour plus d'informations sur les groupes d'utilisateurs, consultez la section Fonctionnalités d'Amazon Cognito dans le guide du développeur Amazon Cognito.

   Utilisez la create-user-poolcommande et spécifiez le nom du pool et la région dans laquelle vous souhaitez le créer.

   Dans cet exemple, nous nommons le pool `dcv-session-manager-client-app` et nous le créons dans`us-east-1`.

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

   Exemple de sortie

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

   Prenez note du`userPoolId`, vous en aurez besoin à l'étape suivante.

2. Créez un nouveau domaine pour votre groupe d'utilisateurs. Utilisez la create-user-pool-domaincommande et spécifiez un nom de domaine et le nom `userPoolId` du groupe d'utilisateurs que vous avez créé à l'étape précédente.

   Dans cet exemple, le nom de domaine est `mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE` et nous l'avons créé dans`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
   ```

   Exemple de sortie

   ```
   {
       "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": {}
       }
   }
   ```

   Le format du domaine du pool d'utilisateurs est le suivant :`https://domain_name.auth.region.amazoncognito.com`. Dans cet exemple, le domaine du groupe d'utilisateurs est`https://mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE.auth.us-east-1.amazoncognito.com`.

3. Créez un client de groupe d'utilisateurs. Utilisez la create-user-pool-clientcommande et spécifiez le `userPoolId` groupe d'utilisateurs que vous avez créé, le nom du client et la région dans laquelle le créer. Incluez également l'`--generate-secret`option permettant de spécifier que vous souhaitez générer un secret pour le client du groupe d'utilisateurs en cours de création.

   Dans ce cas, le nom du client est `dcv-session-manager-client-app` et nous le créons dans la `us-east-1` région.

   ```
   $ 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
   ```

Exemple de sortie

```
{
    "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**
>
> Prenez note de la `ClientId` et`ClientSecret`. Vous devrez fournir ces informations aux développeurs lorsqu'ils demanderont des jetons d'accès pour les demandes d'API.

4. Créez un nouveau serveur de ressources OAuth2 .0 pour le groupe d'utilisateurs. Un serveur de ressources est un serveur dédié aux ressources à accès protégé. Il gère les demandes authentifiées de jetons d'accès.

   Utilisez la create-resource-servercommande et spécifiez le `userPoolId` groupe d'utilisateurs, un identifiant unique et un nom pour le serveur de ressources, l'étendue et la région dans laquelle le créer.

   Dans cet exemple, nous utilisons `dcv-session-manager` comme identifiant et nom, et nous utilisons `sm_scope` comme nom et description de la portée.

   ```
   $ 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
   ```

   Exemple de sortie

   ```
   {
       "ResourceServer": {
   ```

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

5. Mettez à jour le client du groupe d'utilisateurs.

   Utilisez la commande update-user-pool-client. Spécifiez le `userPoolId` groupe d'utilisateurs, le client `ClientId` du groupe d'utilisateurs et la région. Pour`--allowed-o-auth-flows`, spécifiez `client_credentials` pour indiquer que le client doit obtenir des jetons d'accès depuis le point de terminaison du jeton en utilisant une combinaison d'un identifiant client et d'un secret client. Pour`--allowed-o-auth-scopes`, spécifiez l'identifiant du serveur de ressources et le nom de la portée comme suit :*resource_server_identifier*/*scope_name*. Incluez le `--allowed-o-auth-flows-user-pool-client` pour indiquer que le client est autorisé à suivre le OAuth protocole lorsqu'il interagit avec les groupes d'utilisateurs de Cognito.

   ```
   $ 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
   ```

   Exemple de sortie

   ```
   {
       "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**
>
> Le groupe d'utilisateurs est désormais prêt à fournir et à authentifier des jetons d'accès. Dans cet exemple, l'URL du serveur d'autorisation esthttps://cognito-idp.*us-east-1*.amazonaws.com/*us-east-1_QLEXAMPLE*/.well-known/jwks.json.

6. Testez la configuration.

```
$ 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"
```

Exemple de sortie

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

7. Enregistrez le serveur d'autorisation externe à utiliser avec le courtier à l'aide de la register-auth-server commande.

```
$ 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
```

Les développeurs peuvent désormais utiliser le serveur pour demander des jetons d'accès. Lorsque vous demandez des jetons d'accès, fournissez l'ID client, le secret du client et l'URL du serveur générés ici. Pour plus d'informations sur la demande de jetons d'accès, consultez la section Créer, obtenir un jeton d'accès et effectuer une demande d'API dans le guide du développeur Amazon DCV Session Manager.

Configuration de la persistance du broker

Les courtiers Session Manager prennent en charge l'intégration avec des bases de données externes. La base de données externe permet au gestionnaire de session de conserver les données d'état et les clés afin qu'elles soient disponibles ultérieurement. En fait, les données du courtier sont distribuées sur le cluster, ce qui le rend vulnérable à la perte de données si un hôte doit redémarrer ou si un cluster est arrêté. Lorsque cette fonctionnalité est activée, vous pouvez ajouter et supprimer des nœuds de courtage. Vous pouvez également arrêter un cluster et le redémarrer, sans avoir à régénérer les clés ou à perdre les informations concernant le serveur Amazon DCV ouvert ou fermé.

Les types d'informations suivants peuvent être définis pour être conservés :

- Clés pour configurer des sessions afin d'établir une connexion avec les clients
- Données relatives aux sessions en vol
- État du serveur Amazon DCV

Amazon DCV Session Manager prend en charge les bases de données DynamoDB, MariaDB et MySQL. Vous devez configurer et gérer l'une de ces bases de données pour utiliser cette fonctionnalité. Si vos machines de courtage sont hébergées sur Amazon EC2, nous vous recommandons d'utiliser DynamoDB comme base de données externe, car elle ne nécessite aucune configuration supplémentaire.

> **Note**
>
> L'utilisation d'une base de données externe peut entraîner des coûts supplémentaires. Pour obtenir des informations sur la tarification de DynamoDB, consultez la section Tarification de la capacité allouée.

Configurer le broker pour qu'il persiste sur DynamoDB

Configurez les courtiers pour qu'ils commencent à stocker leurs données sur DynamoDB :

1. Ouvrez `/etc/dcv-session-manager-broker/session-manager-broker.properties` à l'aide de votre éditeur de texte préféré et apportez les modifications suivantes :

   - Définir `enable-persistence = true`
   - Définir `persistence-db = dynamodb`
   - Pour `dynamodb-region` spécifier la région &aws ; dans laquelle vous souhaitez stocker les tables contenant les données du courtier. Pour obtenir la liste des régions prises en charge, consultez la section Points de terminaison du service DynamoDB.
   - Pour `dynamodb-table-rcu` spécifier le nombre d'unités de capacité de lecture (RCU) prises en charge par chaque table. Pour plus d'informations sur la RCU, consultez la section Capacité allouée DynamoDB.
   - Pour `dynamodb-table-wcu` spécifier le nombre d'unités de capacité d'écriture (WCU) prises en charge par chaque table. Pour plus d'informations sur la WCU, consultez la section Capacité allouée DynamoDB.
   - Pour dynamodb-table-name-prefix spécifier le préfixe ajouté à chaque table DynamoDB (utile pour distinguer plusieurs clusters de courtiers utilisant le même compte). Seuls les caractères alphanumériques, les points, les tirets et les traits de soulignement sont autorisés.
2. Arrêtez tous les courtiers du cluster. Pour chaque broker, exécutez la commande suivante :

   ```
   sudo systemctl stop dcv-session-manager-broker
   ```
3. Assurez-vous que tous les courtiers du cluster sont arrêtés, puis redémarrez-les tous. Démarrez chaque broker en exécutant la commande suivante :

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

L'hôte du broker doit être autorisé à appeler DynamoDB APIs. Sur les EC2 instances Amazon, les informations d'identification sont automatiquement récupérées à l'aide du service de EC2 métadonnées Amazon. Si vous devez spécifier des informations d'identification différentes, vous pouvez les définir à l'aide de l'une des techniques de récupération d'informations d'identification prises en charge (telles que les propriétés du système Java ou les variables d'environnement). Pour

plus d'informations, consultez la section Fourniture et récupération des informations d'identification &aws ;.

Configurer le broker pour qu'il persiste sur MariaDB/MySQL

> **Note**
>
> Le `/etc/dcv-session-manager-broker/session-manager-broker.properties` fichier contient des données sensibles. Par défaut, son accès en écriture est limité à root et son accès en lecture est limité à root et à l'utilisateur exécutant le Broker. Par défaut, il s'agit de l'`dcvsmbroker`utilisateur. Le Broker vérifie au démarrage que le fichier possède les autorisations attendues.

Configurez les courtiers pour qu'ils commencent à conserver leurs données sur MariaDB/MySQL :

1. Ouvrez `/etc/dcv-session-manager-broker/session-manager-broker.properties` avec votre éditeur de texte préféré et apportez les modifications suivantes :

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

     Dans cette configuration, <db_endpoint>il s'agit du point de terminaison de la base de données, <db_port>du port de base de données et du <db_name>nom de la base de données.
   - Pour `jdbc-user` spécifier le nom de l'utilisateur ayant accès à la base de données.
   - Pour `jdbc-password` spécifier le mot de passe de l'utilisateur ayant accès à la base de données.
2. Arrêtez tous les courtiers du cluster. Pour chaque broker, exécutez la commande suivante :

   ```
   sudo systemctl stop dcv-session-manager-broker
   ```
3. Assurez-vous que tous les courtiers du cluster sont arrêtés, puis redémarrez-les tous. Pour chaque broker, exécutez la commande suivante :

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

Intégration à la passerelle de connexion Amazon DCV

Amazon DCV Connection Gateway est un package logiciel installable qui permet aux utilisateurs d'accéder à un parc de serveurs Amazon DCV via un point d'accès unique à un réseau local ou à un VPC.

Si votre infrastructure inclut des serveurs Amazon DCV accessibles via Amazon DCV Connection Gateway, vous pouvez configurer le gestionnaire de session pour intégrer Amazon DCV Connection Gateway. En suivant les étapes décrites dans la section suivante, le courtier agira en tant que résolveur de session pour la passerelle de connexion. En d'autres termes, le broker exposera un point de terminaison HTTP supplémentaire. La passerelle de connexion effectuera des appels d'API vers le point de terminaison pour récupérer les informations nécessaires pour acheminer les connexions Amazon DCV vers l'hôte sélectionné par le courtier.

Rubriques

- Configurer le Session Manager Broker en tant que résolveur de session pour la passerelle de connexion Amazon DCV
- Facultatif : activez l'authentification du client TLS
- Gestionnaire de sessions Amazon DCV (serveur Amazon DCV) - Référence de mappage DNS

Configurer le Session Manager Broker en tant que résolveur de session pour la passerelle de connexion Amazon DCV

Gestionnaire de sessions côté courtier

1. Ouvrez `/etc/dcv-session-manager-broker/session-manager-broker.properties` à l'aide de votre éditeur de texte préféré et appliquez les modifications suivantes :
   - Définir `enable-gateway = true`
   - Réglé `gateway-to-broker-connector-https-port` sur un port TCP libre (le port par défaut est 8447)

- Défini `gateway-to-broker-connector-bind-host` sur l'adresse IP de l'hôte sur lequel le broker se lie pour les connexions Amazon DCV Connection Gateway (la valeur par défaut est 0.0.0.0)

2. Exécutez ensuite les commandes suivantes pour arrêter et redémarrer le Broker :

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

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

3. Récupérez une copie du certificat auto-signé du courtier et placez-le dans votre répertoire d'utilisateurs.

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

   Vous en aurez besoin lors de l'installation de la passerelle de connexion Amazon DCV à l'étape suivante.

Côté Amazon DCV Connection Gateway

- Veuillez suivre la section de la documentation Amazon DCV Connection Gateway.

  Étant donné que Amazon DCV Connection Gateway envoie des appels d'API HTTP au courtier, si le courtier utilise un certificat auto-signé, vous devrez copier le certificat du courtier sur l'hôte Amazon DCV Connection Gateway (récupéré à l'étape précédente) et définir le `ca-file` paramètre dans la `[resolver]` section de configuration d'Amazon DCV Connection Gateway.

Facultatif : activez l'authentification du client TLS

Une fois l'étape précédente terminée, le gestionnaire de session et la passerelle de connexion peuvent communiquer via un canal sécurisé, où la passerelle de connexion peut vérifier l'identité des courtiers du gestionnaire de session. Si vous demandez aux Session Manager Brokers de valider également l'identité de la passerelle de connexion avant d'établir le canal sécurisé, vous devez activer la fonctionnalité d'authentification du client TLS, en suivant les étapes de la section suivante.

> **Note**
>
> Si le gestionnaire de session se trouve derrière un équilibreur de charge, l'authentification du client TLS ne peut pas être activée avec les équilibreurs de charge dont la connexion TLS est interrompue, tels que les équilibreurs de charge d'application (ALBs) ou les équilibreurs de charge de passerelle (). GLBs Seuls les équilibreurs de charge sans terminaison TLS peuvent être pris en charge, tels que les équilibreurs de charge réseau (). NLBs Si vous utilisez ALBs ou GLBs, vous pouvez faire en sorte que seuls des groupes de sécurité spécifiques puissent contacter les équilibreurs de charge, garantissant ainsi un niveau de sécurité supplémentaire ; plus d'informations sur les groupes de sécurité ici : Groupes de sécurité pour votre VPC

Gestionnaire de sessions côté courtier

1. Pour activer l'authentification du client TLS pour la communication entre les Session Manager Brokers et Amazon DCV Connection Gateway, veuillez suivre les étapes suivantes :
2. Générez les clés et les certificats requis en exécutant : La sortie de la commande vous indiquera le dossier dans lequel les informations d'identification ont été générées et le mot de passe utilisé pour créer le TrustStore fichier.

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

3. Placez une copie de la clé privée et du certificat auto-signé d'Amazon DCV Connection Gateway dans votre répertoire d'utilisateurs. Vous en aurez besoin lorsque vous activerez l'authentification du client TLS dans Amazon DCV Connection Gateway à l'étape suivante.

```
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. Ouvrez ensuite/etc/dcv-session-manager-broker/session-manager-broker.properties à l'aide de votre éditeur de texte préféré et procédez comme suit :

   - Définissez `enable-tls-client-auth-gateway` sur `true`
   - Définissez `gateway-to-broker-connector-trust-store-file` le chemin du TrustStore fichier créé à l'étape précédente

- Définissez `gateway-to-broker-connector-trust-store-pass` le mot de passe utilisé pour créer le TrustStore fichier à l'étape précédente

5. Exécutez ensuite la commande suivante pour arrêter et redémarrer le Broker :

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

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

Côté Amazon DCV Connection Gateway

- Veuillez suivre la section de la documentation Amazon DCV Connection Gateway.
  - utilisez le chemin complet du fichier de certificat que vous avez copié à l'étape précédente lors de la définition du `cert-file` paramètre dans la `[resolver]` section
  - utilisez le chemin complet du fichier clé que vous avez copié à l'étape précédente lors de la définition du `cert-key-file` paramètre dans la `[resolver]` section

Gestionnaire de sessions Amazon DCV (serveur Amazon DCV) - Référence de mappage DNS

La passerelle de connexion Amazon DCV a besoin des noms DNS des serveurs Amazon DCV pour se connecter aux instances du serveur DCV. Cette section explique comment définir un fichier JSON contenant le mappage entre chaque serveur DCV et son nom DNS associé.

Structure de fichier

Le mappage consiste en une liste d'objets JSON avec les champs suivants :

```
[
 {
  “ServerIdType”: "Ip",
  “ServerId”: "192.168.0.1",
  “DnsNames”:
  {
   “InternalDnsName”: "internal"
  }
 },
```

```
  ...
]
```

Où :

`ServerIdType:`

Identifie le type d'identifiant auquel la valeur fait référence ; actuellement, les valeurs disponibles sont IPAddress et agentServerId InstanceId :

`Ip:`

Disponible à la fois pour Amazon EC2 et pour les infrastructures sur site ; peut être rapidement récupéré par les administrateurs système à l'aide d'une commande ifconfig (Linux/macOS) ou ipconfig (Windows). Ces informations sont également disponibles dans la réponse de DescribeServers l'API.

`Id:`

Disponible à la fois pour Amazon EC2 et pour les infrastructures sur site ; l'agent Session Manager crée un nouvel UUID chaque fois que le nom d'hôte ou l'adresse IP changent. Ces informations sont disponibles dans la réponse de DescribeServers l'API.

`Host.Aws.Ec2InstanceId:`

Disponible uniquement pour les EC2 instances Amazon, il identifie de manière unique une machine ; il ne change pas après le redémarrage de l'instance. Peut être récupéré sur l'hôte en contactant http://169.254.169.254/ latest/meta-data/instance -id. Ces informations sont également disponibles dans la réponse de DescribeServers l'API.

`ServerId:`

Un identifiant du type spécifié qui identifie de manière unique chaque serveur Amazon DCV du réseau.

`DnsNames:`

L'objet contenant les noms DNS associés au serveur Amazon DCV, cet objet contiendra :

`InternalDnsNames:`

Le nom DNS utilisé par Amazon DCV Connection Gateway pour se connecter à l'instance.

Utilisez les commandes CLI de Session Manager Broker `register-server-dns-mapping` pour charger le mappage à partir d'un fichier (référence de page de commande : register-server-dns-

mapping) et `describe-server-dns-mappings` pour répertorier les mappages actuellement chargés dans le Session Manager Broker (référence de page de commande : describe-server-dns-mappings).

Persistance

Nous vous recommandons vivement d'activer la fonctionnalité de persistance du courtier du gestionnaire de session, afin de vous protéger contre la perte de mappage lorsque plusieurs courtiers ou l'ensemble du cluster tombent en panne. Pour plus d'informations sur l'activation de la persistance des données, voir Configurer la persistance du courtier

Intégration à Amazon CloudWatch

Le gestionnaire de session prend en charge l'intégration avec Amazon CloudWatch pour les courtiers exécutés sur EC2 des instances Amazon, ainsi que pour les courtiers exécutés sur des hôtes locaux.

Amazon CloudWatch surveille vos ressources Amazon Web Services (AWS) et les applications que vous utilisez AWS en temps réel. Vous pouvez les utiliser CloudWatch pour collecter et suivre les métriques, qui sont des variables que vous pouvez mesurer pour vos ressources et vos applications. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.

Vous pouvez configurer le Session Manager Broker pour envoyer les données métriques suivantes à Amazon CloudWatch :

- `Number of DCV servers`: le nombre de serveurs DCV gérés par le Broker.
- `Number of ready DCV servers`: le nombre de serveurs DCV dont l'`READY`état est géré par le Broker.
- `Number of DCV sessions`: le nombre de sessions DCV gérées par le Broker.
- `Number of DCV console sessions`: le nombre de sessions de console DCV gérées par le Broker.
- `Number of DCV virtual sessions`: le nombre de sessions virtuelles DCV gérées par le Broker.
- `Heap memory used`: la quantité de mémoire de segment utilisée par le broker.
- `Off-heap memory used`: la quantité de mémoire hors segment utilisée par le Broker.
- `Describe sessions request time`: le temps nécessaire pour traiter les demandes DescribeSessions d'API.

- `Delete sessions request time`: le temps nécessaire pour traiter les demandes DeleteSessions d'API.
- `Create sessions request time`: le temps nécessaire pour traiter les demandes CreateSessions d'API.
- `Get session connection data request time`: le temps nécessaire pour traiter les demandes GetSessionConnectionData d'API.
- `Update session permissions sequest time`: le temps nécessaire pour traiter les demandes UpdateSessionPermissions d'API.

Pour configurer le courtier afin qu'il envoie des données métriques à Amazon CloudWatch

1. Ouvrez `/etc/dcv-session-manager-broker/session-manager-broker.properties` à l'aide de votre éditeur de texte préféré et procédez comme suit :
   - Réglé `enable-cloud-watch-metrics` sur `true`
   - Pour`cloud-watch-region`, spécifiez la région dans laquelle vous souhaitez collecter les données métriques.

   > **Note**
   >
   > Si votre broker s'exécute sur une EC2 instance Amazon, ce paramètre est facultatif. La région est automatiquement extraite du service de métadonnées d'instance (IMDS). Si vous exécutez le Broker sur un hôte local, ce paramètre est obligatoire.

2. Arrêtez et redémarrez le Broker.

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

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

L'hôte du Broker doit également être autorisé à appeler l'`cloudwatch:PutMetricData`API. AWS les informations d'identification peuvent être récupérées à l'aide de l'une des techniques de récupération d'informations d'identification prises en charge. Pour plus d'informations, consultez la section Fourniture et récupération des AWS informations d'identification.

Mise à niveau du gestionnaire de sessions Amazon DCV

À mesure que les systèmes Amazon DCV gagnent en ampleur et en complexité, il est important de s'assurer que le gestionnaire de session reste up-to-date capable de répondre aux demandes croissantes. Les packages d'agent et de courtier nécessiteront une mise à niveau de temps à autre. Cette section décrit le processus de mise à niveau du gestionnaire de session Amazon DCV, en abordant la procédure de mise à niveau et les recommandations pour la maintenance de votre système.

La rubrique suivante décrit comment mettre à niveau le gestionnaire de session.

> **Note**
>
> Nous vous recommandons vivement de mettre à niveau tous les agents du gestionnaire de session avant de mettre à niveau les courtiers du gestionnaire de session afin d'éviter les problèmes d'incompatibilité lors de l'introduction de nouvelles fonctionnalités.

Rubriques

- Mise à niveau de l'agent Amazon DCV Session Manager
- Mise à niveau du broker Amazon DCV Session Manager

Mise à niveau de l'agent Amazon DCV Session Manager

Les agents Amazon DCV Session Manager reçoivent des instructions du courtier et les exécutent sur leurs serveurs Amazon DCV respectifs. Dans le cadre de la maintenance de routine, les agents doivent être mis à niveau pour répondre aux nouvelles normes et exigences. Cette section explique le processus de mise à niveau de vos agents du gestionnaire de session.

Linux host

> **Note**
>
> Les instructions suivantes concernent l'installation de l'agent sur des hôtes x86 64 bits. Pour installer l'agent sur des hôtes ARM 64 bits, pour Amazon Linux, RHEL et Centos, remplacez *x86_64* par `aarch64` et pour Ubuntu, remplacez par*amd64*. `arm64`

Pour mettre à jour l'agent sur un hôte Linux

1. Exécutez la commande suivante pour arrêter l'agent.

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

2. Téléchargez le package d'installation.

   - Amazon Linux 2 et 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 et 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. Installez le package .

   - Amazon Linux 2 et RHEL 7.x

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

   - RHEL 8.x et 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. Exécutez la commande suivante pour démarrer l'agent.

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

Windows host

Pour mettre à jour l'agent sur un hôte Windows

1. Arrêtez le service d'agent. Exécutez les commandes suivantes à l'invite de commandes.

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

2. Téléchargez le programme d'installation de l'agent.
3. Exécutez le programme d'installation. Sur l'écran d'accueil, choisissez Next.
4. Sur l'écran EULA, lisez attentivement le contrat de licence et, si vous êtes d'accord, sélectionnez J'accepte les termes, puis Next.
5. Pour commencer l'installation, choisissez Installer.
6. Redémarrez le service d'agent. Exécutez les commandes suivantes à l'invite de commandes.

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

macOS host

Pour mettre à jour l'agent sur un hôte macOS

1. Exécutez la commande suivante pour arrêter l'agent.

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

2. Téléchargez le programme d'installation de l'agent ou utilisez la ligne de commande :

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

3. Installez le package . Vous pouvez utiliser la ligne de commande ou via le programme d'installation de l'agent :

   - Ligne de commande :

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

- Graphique : double-cliquez sur le `.pkg` fichier et suivez les instructions de l'assistant d'installation.

4. Exécutez la commande suivante pour démarrer l'agent.

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

Mise à niveau du broker Amazon DCV Session Manager

Les courtiers Amazon DCV Session Manager transmettent les demandes d'API à leurs agents concernés. Ils sont installés sur un hôte distinct des serveurs Amazon DCV. Dans le cadre de la maintenance de routine, les courtiers doivent être mis à niveau pour répondre aux nouvelles normes et exigences. Cette section vous explique le processus de mise à niveau de vos courtiers Session Manager.

Pour mettre à niveau le courtier

1. Connectez-vous à l'hôte sur lequel vous souhaitez mettre à niveau le broker.
2. Arrêtez le service de courtage.

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

3. Téléchargez le package d'installation.

- Amazon Linux 2 et 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 et 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. Installez le package .

- Amazon Linux 2 et RHEL 7.x

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

- RHEL 8.x et 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. Démarrez le service de courtage et assurez-vous qu'il démarre automatiquement à chaque démarrage de l'instance.

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

Référence Broker CLI

Le broker Amazon DCV Session Manager est un outil d'interface de ligne de commande (CLI) qui fournit un contrôle administratif sur le gestionnaire de session. Cette référence couvre l'ensemble complet des commandes CLI disponibles pour la gestion des sessions, des utilisateurs, des ressources et d'autres aspects du gestionnaire de session. Les administrateurs peuvent automatiser les tâches de gestion de routine, résoudre les problèmes et optimiser les performances de leur infrastructure Amazon DCV.

Utilisez les commandes suivantes si vous utilisez un serveur d'authentification externe pour générer des jetons d'accès OAuth 2.0 :

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

Utilisez les commandes suivantes si vous utilisez le courtier Session Manager comme serveur d'authentification OAuth 2.0.

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

Utilisez les commandes suivantes pour gérer l'agent Session Manager.

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

Utilisez les commandes suivantes pour gérer le serveur DCV - fichier de mappage des noms DNS.

- register-server-dns-mappings

- describe-server-dns-mappings

register-auth-server

Enregistre un serveur d'authentification externe à utiliser avec le courtier.

Par défaut, le gestionnaire de session utilise le courtier comme serveur d'authentification pour générer des jetons d'accès OAuth 2.0. Si vous utilisez le broker comme serveur d'authentification, aucune configuration supplémentaire n'est requise.

Toutefois, si vous choisissez d'utiliser un serveur d'authentification externe, tel qu'Active Directory ou Amazon Cognito, vous devez utiliser cette commande pour enregistrer le serveur d'authentification externe.

Rubriques

- Syntaxe
- Options
- exemple

Syntaxe

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

Options

`--url`

URL du serveur d'authentification externe à utiliser. Vous devez ajouter `.well-known/jwks.json` à l'URL du serveur d'authentification.

Type : String

Obligatoire : oui

exemple

L'exemple suivant enregistre un serveur d'authentification externe avec une URL de`https://my-auth-server.com/`.

Commande

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

Sortie

```
Jwk url registered.
```

list-auth-servers

Répertorie les serveurs d'authentification externes qui ont été enregistrés.

Rubriques

- Syntaxe
- Sortie
- exemple

Syntaxe

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

Sortie

`Urls`

Le URLs des serveurs d'authentification externes enregistrés.

exemple

L'exemple suivant répertorie tous les serveurs d'authentification externes qui ont été enregistrés.

Commande

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

Sortie

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

unregister-auth-server

Désenregistre un serveur d'authentification externe. Une fois que vous avez désinscrit un serveur d'authentification externe, il ne peut plus être utilisé pour générer des jetons d'accès OAuth 2.0.

Rubriques

- Syntaxe
- Options
- Sortie
- exemple

Syntaxe

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

Options

`--url`

URL du serveur d'authentification externe dont vous souhaitez annuler l'enregistrement. Vous devez ajouter `.well-known/jwks.json` à l'URL du serveur d'authentification.

Type : String

Obligatoire : oui

Sortie

`Url`

URL du serveur d'authentification externe non enregistré.

exemple

L'exemple suivant enregistre un serveur d'authentification externe avec une URL de`https://my-auth-server.com/`.

Commande

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

Sortie

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

register-api-client

Enregistre un client Session Manager auprès du courtier et génère des informations d'identification client qui peuvent être utilisées par le client pour récupérer un jeton d'accès OAuth 2.0, nécessaire pour effectuer des demandes d'API.

> **Important**
>
> Assurez-vous de stocker les informations d'identification dans un endroit sûr. Ils ne pourront pas être récupérés plus tard.

Cette commande est utilisée uniquement si le broker est utilisé comme serveur d'authentification OAuth 2.0.

Rubriques

- Syntaxe

- Options
- Sortie
- exemple

Syntaxe

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

Options

`--name`

Nom unique utilisé pour identifier le client du gestionnaire de session.

Type : String

Obligatoire : oui

Sortie

`client-id`

L'ID client unique à utiliser par le client Session Manager pour récupérer un jeton d'accès OAuth 2.0.

`client-password`

Le mot de passe à utiliser par le client Session Manager pour récupérer un jeton d'accès OAuth 2.0.

exemple

L'exemple suivant enregistre un client nommé`my-sm-client`.

Commande

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

Sortie

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

describe-api-clients

Répertorie les clients du gestionnaire de session enregistrés auprès du courtier.

Rubriques

- Syntaxe
- Sortie
- exemple

Syntaxe

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

Sortie

`name`

Nom unique du client Session Manager.

`id`

L'identifiant unique du client Session Manager.

`active`

Indique l'état du client du gestionnaire de session. Si le client est actif, la valeur est `true` ; dans le cas contraire, c'est le cas`false`.

exemple

L'exemple suivant répertorie les clients du gestionnaire de session enregistrés.

Commande

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

Sortie

```
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

Désactive un client Session Manager enregistré. Un client Session Manager désactivé ne peut plus utiliser ses informations d'identification pour récupérer des jetons d'accès OAuth 2.0.

Rubriques

- Syntaxe
- Options
- exemple

Syntaxe

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

Options

`--client -id`

ID client du client Session Manager à désactiver.

Type : String

Obligatoire : oui

exemple

L'exemple suivant désactive un client Session Manager dont l'ID client est. `f855b54b-40d4-4769-b792-b727bEXAMPLE`

Commande

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

Sortie

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

renew-auth-server-api-clé

Renouvelle les clés publiques et privées utilisées par le courtier pour signer les jetons d'accès OAuth 2.0 qui sont envoyés au client Session Manager. Si vous renouvelez les clés, vous devez fournir la nouvelle clé privée au développeur, car elle est nécessaire pour effectuer des demandes d'API.

Rubriques

- Syntaxe
- exemple

Syntaxe

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

exemple

L'exemple suivant renouvelle les clés publique et privée.

Commande

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

Sortie

```
Keys renewed.
```

generate-software-statement

Génère une instruction logicielle.

Les agents doivent être enregistrés auprès du courtier pour permettre la communication. Les agents ont besoin d'une déclaration logicielle pour s'inscrire auprès du courtier. Une fois que l'agent a reçu une instruction logicielle, il peut s'enregistrer automatiquement auprès du courtier à l'aide du protocole d'enregistrement dynamique des clients OAuth 2.0. Une fois que l'agent s'est enregistré auprès du courtier, il reçoit un identifiant client et un secret client qu'il utilise pour s'authentifier auprès du courtier.

Le courtier et l'agent reçoivent et utilisent une déclaration logicielle par défaut lors de leur première installation. Vous pouvez continuer à utiliser l'instruction logicielle par défaut ou choisir d'en générer une nouvelle. Si vous générez une nouvelle instruction logicielle, vous devez la placer dans un nouveau fichier sur l'agent, puis ajouter le chemin du fichier au `agent.software_statement_path` paramètre du `agent.conf` fichier. Ensuite, arrêtez et redémarrez l'agent afin qu'il puisse utiliser la nouvelle instruction logicielle pour s'enregistrer auprès du courtier.

Rubriques

- Syntaxe
- Sortie
- exemple

Syntaxe

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

Sortie

`software-statement`

La déclaration du logiciel.

exemple

L'exemple suivant génère une instruction logicielle.

Commande

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

Sortie

```
software-statement:
 ewogICJpZCIgOiAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiYWN0aXZlIiA6IHRydWUsCi
```

describe-software-statements

Décrit les instructions logicielles existantes.

Rubriques

- Syntaxe
- Sortie
- exemple

Syntaxe

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

Sortie

`software-statement`

La déclaration du logiciel.

`issued-at`

Date et heure de génération du logiciel.

`is-active`

État actuel de la déclaration logicielle. `true`si l'instruction logicielle est active ; dans le cas contraire, elle l'est`false`.

exemple

L'exemple suivant génère une instruction logicielle.

Commande

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

Sortie

```
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

Désactive une instruction logicielle. Lorsque vous désactivez une instruction logicielle, elle ne peut plus être utilisée pour les enregistrements d'agents.

Rubriques

- Syntaxe
- Options
- exemple

Syntaxe

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

Options

`--software-statement`

Déclaration logicielle à désactiver.

Type : String

Obligatoire : oui

exemple

L'exemple suivant désactive une instruction logicielle.

Commande

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

Sortie

```
Software statement
 EXAMPLEpZCIgOiAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj
 deactivated
```

describe-agent-clients

Décrit les agents enregistrés auprès du courtier.

Rubriques

- Syntaxe
- Sortie
- exemple

Syntaxe

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

Sortie

`name`

Le nom de l'agent.

`id`

L'identifiant unique de l'agent.

`active`

État de l'agent. `true`si l'agent est actif, sinon il l'est`false`.

exemple

L'exemple suivant décrit les agents.

Commande

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

Sortie

```
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

Désenregistrez un agent auprès du courtier.

Rubriques

- Syntaxe
- Options
- exemple

Syntaxe

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

Options

`--client-id`

ID de l'agent dont l'enregistrement doit être annulé.

Type : String

Obligatoire : oui

exemple

L'exemple suivant annule l'enregistrement d'un agent.

Commande

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

Sortie

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

register-server-dns-mappings

Enregistrez les serveurs DCV - mappages de noms DNS provenant d'un fichier JSON.

Syntaxe

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

Options

`--file-path`

Le chemin du fichier contenant les mappages de noms DNS entre serveurs DCV et serveurs.

Type : String

Obligatoire : oui

exemple

L'exemple suivant enregistre les serveurs DCV - mappages de noms DNS à partir du fichier /tmp/mappings.json.

Commande

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

Sortie

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

describe-server-dns-mappings

Décrivez les serveurs DCV - mappages de noms DNS actuellement disponibles.

Syntaxe

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

Sortie

`serverIdType`

Type de l'identifiant du serveur.

`serverId`

L'ID unique du serveur.

`dnsNames`

Les noms DNS internes et externes

`internalDnsNames`

Les noms DNS internes

`externalDnsNames`

Les noms DNS externes

exemple

L'exemple suivant répertorie les serveurs DCV enregistrés - mappages de noms DNS.

Commande

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

Sortie

```
[
{
 "serverIdType" : "Id",
 "serverId" : "192.168.0.1",
 "dnsNames" : {
  "internalDnsName" : "internal1",
  "externalDnsName" : "external1"
```

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

Référence du fichier de configuration

Cette section de référence fournit une vue d'ensemble des options de configuration disponibles pour le gestionnaire de session. Les configurations incluent des modifications apportées à la fois au fichier de l'agent et au fichier du courtier. Chaque configuration inclut une explication de l'objectif, des valeurs acceptées et de l'impact sur le comportement global du système. Amazon DCV Session Manager peut être personnalisé pour répondre aux exigences uniques d'un système Amazon DCV.

Rubriques

- Fichier de configuration du broker
- Fichier de configuration de l'agent

Fichier de configuration du broker

Le fichier de configuration du broker (`/etc/dcv-session-manager-broker/session-manager-broker.properties`) inclut des paramètres qui peuvent être configurés pour personnaliser les fonctionnalités du gestionnaire de session. Vous pouvez modifier le fichier de configuration à l'aide de votre éditeur de texte préféré.

> **Note**
>
> Le `/etc/dcv-session-manager-broker/session-manager-broker.properties` fichier contient des données sensibles. Par défaut, son accès en écriture est limité à root et son accès en lecture est limité à root et à l'utilisateur exécutant le broker. Par défaut, il s'agit de l'`dcvsmbroker`utilisateur. Le broker vérifie au démarrage que le fichier possède les autorisations attendues.

Le tableau suivant répertorie les paramètres du fichier de configuration du broker.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`broker-ja`	Non		Spécifie le chemin d'accès au répertoire de base Java

Nom du paramètr	Obligatoire	Valeur par défaut	Description
va-home			que le courtier utilisera à la place du répertoire par défaut du système. Si cette option est définie, le broker l'utilisera <broker-java-home>/bin/java au démarrage. Conseil : le broker nécessite Java Runtime Environment 11 et il est installé s'il est absent en tant que dépendance une fois l'installation réussie. Si la version 11 n'est pas définie comme environnement Java par défaut, son répertoire personnel peut être récupéré à l'aide de la commande suivante : `$ sudo alternatives --display java`
sessions creensh-max-width	Non	160	Spécifie la largeur maximale, en pixels, des captures d'écran de session prises à l'aide de l'GetSessionScreenshotsAPI.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`sessior s creensh - max- heig ht`	Non	`100`	Spécifie la hauteur maximale, en pixels, des captures d'écran de session prises à l'aide de l'GetSessionScreensh otsAPI.
`sessior s creensh - format`	Non	`png`	Format de fichier image des captures d'écran de session prises à l'aide de l'GetSessionScreensh otsAPI.
`create- se ssions- qu eue- max-s ize`	Non	`1000`	Le nombre maximum de demandes d'CreateSes sionsAPI non satisfaites qui peuvent être mises en file d'attente. Lorsque la file d'attente est pleine, les nouvelles demandes non satisfaites sont rejetées.
`create- se ssions- qu eue- max-t ime- secon ds`	Non	`1800`	Durée maximale, en secondes, pendant laquelle une demande d'CreateSes sionsAPI non satisfaite peut rester dans la file d'attente. Si la demande ne peut pas être satisfaite dans le délai imparti, elle échoue.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`sessionmanager-working-path`	Oui	`/tmp`	Spécifie le chemin d'accès au répertoire dans lequel le broker écrit les fichiers nécessaires à son fonctionnement. Ce répertoire doit être accessible uniquement au courtier.
`enable-authorization-server`	Oui	`true`	Spécifie si le broker est le serveur d'authentification utilisé pour générer des jetons d'accès OAuth 2.0 pour le client APIs.
`enable-authorization`	Oui	`true`	Active ou désactive l'autorisation du client. Si vous activez l'autorisation du client, l'API client doit fournir un jeton d'accès lors des demandes d'API. Si vous désactivez l'autorisation du client, celui-ci APIs peut faire des demandes sans jetons d'accès.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`enable-agent-authorizatic`	Oui	`true`	Active ou désactive l'autorisation de l'agent. Si vous activez l'autorisation de l'agent, celui-ci doit fournir un jeton d'accès lorsqu'il communique avec le courtier.
`delete-session-duration-hours`	Non	`1`	Spécifie le nombre d'heures après lesquelles les sessions supprimées deviennent invisibles et ne sont plus renvoyées par les appels `DescribeSession` d'API. Obsolète : `delete-session-duration-hours` passage à `delete-session-duration-seconds` — Disponible depuis la version 2024.0-493.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
delete-session-duration-seconds	Non	3600	Spécifie le nombre de secondes après lesquelles les sessions supprimées deviennent invisibles et ne sont plus renvoyées par les appels DescribeSession d'API. Ce paramètre remplace le delete-session-duration-hours paramètre obsolète, disponible depuis la version 2024.0-493.
connect-session-token-duration-minutes	Non	60	Spécifie le nombre de minutes pendant lesquelles le ConnectSession jeton reste valide.
client-to-broker-connect-https-port	Oui	8443	Spécifie le port HTTPS sur lequel le broker écoute les connexions des clients.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
client-to-broker-connect bind-host	Non	0.0.0.0	Spécifie l'adresse IP de l'hôte sur lequel le broker se lie pour les connexions client.
client-to-broker-connect key-store-file	Oui		Spécifie le magasin de clés utilisé pour les connexions client TLS.
client-to-broker-connect key-store-pass	Oui		Spécifie le code d'accès au magasin de clés.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
agent-to-broker-connector-https-port	Oui	8445	Spécifie le port HTTPS sur lequel le broker écoute les connexions des agents.
agent-to-broker-connector-bind-host	Non	0.0.0.0	Spécifie l'adresse IP de l'hôte sur lequel le broker se lie pour les connexions aux agents.
agent-to-broker-connector-key-store-file	Oui		Spécifie le magasin de clés utilisé pour les connexions aux agents TLS.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
agent-to-broker-connector-key-store-pass	Oui		Spécifie le code d'accès au magasin de clés.
broker-to-broker-port	Oui	47100	Spécifie le port utilisé pour les broker-to-broker connexions.
broker-to-broker-bind-host	Non	0.0.0.0	Spécifie l'adresse IP de l'hôte sur lequel le broker se lie pour les broker-to-broker connexions.
broker-to-broker-discover-port	Oui	47500	Spécifie le port utilisé par les courtiers pour se découvrir mutuellement.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`broker-to-broker-discover address`	Non		Spécifie les adresses IP et les ports des autres courtiers de la flotte au *`port`* format *`ip_address`* :. S'il existe plusieurs courtiers, séparez les valeurs par une virgule. Si vous spécifiez `broker-to-broker-discovery-multicast-group` ,`broker-to-broker-discovery-multicast-port` , ou `broker-to-broker-discovery-AWS-region` `broker-to-broker-discovery-AWS-alb-target-group-arn` , omettez ce paramètre.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
broker-to-broker-discovery-multica-group	Non		Spécifie le groupe de multidiffusion pour la broker-to-roker découverte. Si vous spécifiez broker-to-broker-discovery-addresses ou broker-to-broker-discovery-AWS-alb-target-group-arn omettez ce paramètre. broker-to-broker-discovery-aws-region
broker-to-broker-discovery-multica-port	Non		Spécifie le port de multidiffusion pour la broker-to-broker découverte. Si vous spécifiez broker-to-broker-discovery-addresses ou broker-to-broker-discovery-AWS-alb-target-group-arn omettez ce paramètre. broker-to-broker-discovery-AWS-region

Nom du paramètr	Obligatoire	Valeur par défaut	Description
broker-to-broker-discover-AWS-region	Non		Spécifie la AWS région de l'équilibreur de charge d'application utilisé pour la découverte entre courtiers. Si vous spécifiez broker-to-broker-discovery-multicast-group ou broker-to-broker-discovery-addresses omettez ce paramètre. broker-to-broker-discovery-multicast-port
broker-to-broker-discover-AWS-alb-target-group-arn	Non		L'ARN de l'utilisateur du groupe cible de l'équilibreur de charge d'application à des fins de broker-to-broker découverte. Si vous spécifiez broker-to-broker-discovery-multicast-group ou broker-to-broker-discovery-addresses omettez ce paramètre. broker-to-broker-discovery-multicast-port

Nom du paramètr	Obligatoire	Valeur par défaut	Description
broker-to-broker-distributed-memory-max-size-mb	Non	4096	Spécifie la quantité maximale de mémoire hors segment à utiliser par un seul courtier pour stocker les données de session Amazon DCV.
broker-to-broker-key-store-file	Oui		Spécifie le magasin de clés utilisé pour les connexions entre courtiers TLS.
broker-to-broker-key-store-pass	Oui		Spécifie le code d'accès au magasin de clés.
enable-cloud-watch-metrics	Non	false	Active ou désactive les CloudWatch métriques Amazon. Si vous activez CloudWatch les métriques , vous devrez peut-être spécifier une valeur pour `cloud-watch-region` .

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`cloud-watch-region`	Non	Nécessaire uniquement s'`enable-cloud-watch-metrics` il est défini sur`true`. Si le broker est installé sur une instance Amazon EC2, la région est extraite de l'IMDS.	AWS Région dans laquelle les CloudWatch statistiques sont publiées.
`max-api-requests-per-second`	Non	`1000`	Spécifie le nombre maximum de demandes que l'API du courtier peut traiter chaque seconde avant d'être limitée.
`enable-throttling-forwarded-for-header`	Non	`false`	S'il est réglé sur `true` le throttling, il récupère l'adresse IP de l'appelant dans l'en-tête s'il est X-Forwared-For présent.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
create-sessions-number-of-retries-on-failure	Non	2	Spécifie le nombre maximal de tentatives à effectuer après l'échec d'une demande de création de session sur un hôte de serveur Amazon DCV. Définissez cette valeur sur 0 pour ne jamais effectuer de nouvelles tentatives en cas d'échec.
autorun-file-arguments-max-size	Non	50	Spécifie le nombre maximum d'arguments pouvant être transmis au fichier autorun.
autorun-file-arguments-max-argumer length	Non	150	Spécifie la longueur maximale en caractères de chaque argument de fichier autorun.
enable-persister	Oui	false	Si ce paramètre est défini surtrue, les données d'état du courtier sont conservées dans une base de données externe.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
persist ce-db	Non	Nécessaire uniquement s'enable-persistence il est défini surtrue.	Spécifie la base de données utilisée pour la persistance. Les seules valeurs prises en charge sont : dynamodb etmysql.
dynamoc region	Non	Nécessaire uniquement s'enable-persistence il est défini sur true et persistence-db est défini surdynamodb.	Spécifie la région dans laquelle les tables DynamoDB sont créées et accessibles.
dynamoc table-rcu	Non	Nécessaire uniquement s'enable-persistence il est défini sur true et persistence-db est défini surdynamodb.	Spécifie les unités de capacité de lecture (RCU) pour chaque table DynamoDB. Pour plus d'informations sur les RCU, voir Tarification de la capacité allouée.
dynamoc table-wcu	Non	Nécessaire uniquement s'enable-persistence il est défini sur true et persistence-db est défini surdynamodb.	Spécifie les unités de capacité d'écritur e (WCU) pour chaque table DynamoDB. Pour plus d'informations sur la WCU, consultez la section Tarification de la capacité allouée.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`dynamoc table- nam e- prefix`	Non	Nécessaire uniquement s'`enable-persistence` il est défini sur `true` et `persistence-db` est défini sur`dynamodb`.	Spécifie le préfixe ajouté à chaque table DynamoDB (utile pour distinguer plusieurs clusters de courtiers utilisant le même compte). AWS Seuls les caractères alphanumé riques, les points, les tirets et les traits de soulignem ent sont autorisés.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`jdbc-connection-url`	Non	Nécessaire uniquement s'`enable-persistence` il est défini sur `true` et `persistence-db` est défini sur`mysql`.	Spécifie l'URL de connexion à la MariaDB/MySQL base de données ; elle contient le point de terminaison et le nom de la base de données. L'URL doit avoir le format suivant : `jdbc:mysql://<db_endpoint>:<db_port>/<db_name>?createDatabaseIfNotExist=true` Où `<db_endpoint>` se trouvent le point MariaDB/MySQL de terminaison de la base de données, `<db_port>` le port de base de données et `<db_name>` le nom de la base de données.
`jdbc-user`	Non	Nécessaire uniquement s'`enable-persistence` il est défini sur `true` et `persistence-db` est défini sur`mysql`.	Spécifie le nom de l'utilisateur ayant accès à la MariaDB/MySQL base de données.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`jdbc-password`	Non	Nécessaire uniquement s'`enable-persistence` il est défini sur `true` et `persistence-db` est défini sur`mysql`.	Spécifie le mot de passe de l'utilisateur ayant accès à la MariaDB/MySQL base de données.
`seconds-before-deleting-unreachable-dcv-server`	Non	`1800`	Spécifie le nombre de secondes après lesquelles un serveur Amazon DCV inaccessible est supprimé du système.
`seconds-before-deleting-sessions-unreachable-server`	Non		Spécifie le nombre de secondes après lesquelles les sessions sur un serveur Amazon DCV inaccessible sont supprimées du système. La suppression des sessions d'un serveur inaccessible est désactivée par défaut. Pour activer la suppression de sessions sur des serveurs inaccessibles, fournissez une valeur valide.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`sessior s creensh - max- widt h`	Non	`160`	Spécifie la largeur maximale, en pixels, des captures d'écran de session prises à l'aide de l'GetSessionScreensh otsAPI. S'il `session-s creenshot-max-widt` h est défini dans le fichier de configuration du client Web, il est prioritaire et remplace cette valeur par défaut. Notez qu'il s'agit de la largeur maximale. La résolution réelle de la capture d'écran peut donc être inférieure.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`sessior s creensh - max- heig ht`	Non	`100`	Spécifie la hauteur maximale, en pixels, des captures d'écran de session prises à l'aide de l'GetSessionScreensh otsAPI. S'il `session-screenshot-max-height` est défini dans le fichier de configura tion du client Web, il est prioritaire et remplace cette valeur par défaut. Notez qu'il s'agit de la hauteur maximale. La résolution réelle de la capture d'écran peut donc être inférieure.

Fichier de configuration de l'agent

Le fichier de configuration de l'agent (`/etc/dcv-session-manager-agent/agent.conf`pour Linux, macOS et `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` Windows) inclut des paramètres qui peuvent être configurés pour personnaliser les fonctionnalités du gestionnaire de session. Vous pouvez modifier le fichier de configuration à l'aide de votre éditeur de texte préféré.

Le tableau suivant répertorie les paramètres du fichier de configuration de l'agent.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`agent.b ker_hos`	Oui		Spécifie le nom DNS de l'hôte du courtier.
`agent.b ker_por`	Oui	`8445`	Spécifie le port par lequel communiquer avec le courtier.
`agent.c file`	Non		Nécessaire uniquemen t s'`tls_strict` il est défini sur`true`. Spécifie le chemin d'accès au fichier de certificat (.pem) nécessaire pour valider le certificat TLS. Copiez le certificat auto-signé du courtier vers l'agent.
`agent.i t_folde`	Non	• `/var/lib/dcv-session-manager-agent/init` (Linux)	Spécifie le chemin d'accès à un dossier sur le serveur hôte utilisé pour stocker les scripts personnal isés autorisés à initialiser les sessions du serveur Amazon DCV lors de leur création. Vous devez spécifier un chemin absolu. Le dossier doit être accessible et les fichiers doivent être exécutabl es par les utilisateurs qui utilisent le paramètre de InitFilerequête de l'CreateSessionsAPI.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`agent.t _strict`	Non	`true`	Indique si une validatio n TLS stricte doit être utilisée.
`agent.s tware_s tement_ th`	Non		Nécessaire uniquemen t si l'instruction logiciell e par défaut n'est pas utilisée. Spécifie le chemin d'accès au fichier d'instruc tions logicielles. Pour de plus amples informati ons, veuillez consulter generate-software-statement.
`agent.t s_folde`	Non	• `/etc/dcv-session-manager-ag ent` (Linux/macOS) • `C:\Program Files \NICE\DCVSess ionManagerAgent \conf\tags` (Windows)	Spécifie le chemin d'accès au dossier dans lequel se trouvent les fichiers de balises. Pour de plus amples informations, veuillez consulter Utilisati on de balises pour cibler les serveurs Amazon DCV.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`agent.a orun_fo er`	Non	• `/var/lib/dcv-session-manage r-agent/autorun` (Linux) • `C:\ProgramData \NICE\DcvSess ionManagerAgent \autorun` (Windows)	Spécifie le chemin d'accès à un dossier sur le serveur hôte utilisé pour stocker les scripts et les applications autorisés à être exécutés automatiquement au démarrage de la session. Vous devez spécifier un chemin absolu. Le dossier doit être accessibl e et les fichiers doivent être exécutables par les utilisateurs qui utilisent le paramètre de AutorunFi lerequête de l'CreateSes sionsAPI.
`agent.m _virtua sessior`	Non	-1 (aucune limite)	Nombre maximal de sessions virtuelles pouvant être créées sur un serveur Amazon DCV à l'aide d'Amazon DCV Session Manager.
`agent.m _concur nt_sess ns_per_ er`	Non	1	Nombre maximal de sessions virtuelles pouvant être créées sur un serveur Amazon DCV par un seul utilisateur à l'aide d'Amazon DCV Session Manager.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`agent.broker_update_interval`	Non	30	Spécifie le nombre de secondes à attendre avant d'envoyer des données mises à jour au courtier. Les données envoyées incluent le statut du serveur Amazon DCV et de l'hôte, ainsi que les informations de session mises à jour. Des valeurs plus faibles permettent au gestionnaire de session de mieux connaître les modifications qui se produisent sur le système sur lequel l'agent s'exécute , mais augmentent la charge du système et le trafic réseau. Des valeurs plus élevées diminuent la charge du système et du réseau, mais le gestionnaire de session réagit moins bien aux modifications du système, d'où des valeurs supérieures à celles `120` qui ne sont pas recommandées.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`agent.e ble_que _logged n_users`	Non	true	Activez l'enregistrement des utilisateurs connectés sous Windows. Lorsque c'est vrai, interroge le système pour connaître les utilisateurs réellemen t connectés. Si la valeur est fausse, renvoie « inconnu »
`agent.p ferred_ twork_i erface`	Non		Spécifie le nom de l'interfa ce réseau à utiliser pour la détection de l'adresse IP de l'hôte. Exemples : `eth0` (Linux), `Ethernet 4` (Windows). Linux/mac OS les noms distinguent majuscules et minuscule s, les noms Windows ne distinguent pas les majuscules et minuscules. Si elle n'est pas spécifiée ou si l'interface réseau spécifiée est introuvable, la première interface sans boucle est sélectionnée.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`log.lev`	Non	`info`	Spécifie le niveau de verbosité des fichiers journaux. Les différents niveaux de détails disponibles sont les suivants : • `error`: fournit le moins de détails possible. Inclut les erreurs uniquement. • `warning`—Inclut des erreurs et des avertissements. • `info`: le niveau de verbosité par défaut. Inclut les erreurs, les avertissements et les messages d'information. • `debug`—Fournit le plus de détails. Fournit des informations détaillées utiles au débogage des problèmes.
`log.dir tory`	Non	• `/var/log/dcv-session-manager-agent/`(Linux/macOS) • `C:\ProgramData\NICE\DCVSessionManagerAgent\log` (Windows)	Spécifie le répertoire dans lequel créer les fichiers journaux.

Nom du paramètr	Obligatoire	Valeur par défaut	Description
`log.rot ion`	Non	`daily`	Spécifie la rotation du fichier journal. Les valeurs valides sont : • `hourly`—Les fichiers journaux font l'objet d'une rotation toutes les heures. • `daily`—Les fichiers journaux font l'objet d'une rotation quotidien ne.
`log.ma f ile- size`	Non	`10485760`	Lorsque la taille d'un fichier journal atteint la taille spécifiée en octets, il fait l'objet d'une rotation. Un nouveau fichier journal sera créé et d'autres événements de journal seront placés dans le nouveau fichier.
`log.rot e`	Non	`9`	Le nombre maximum de fichiers journaux conservés pendant la rotation. Chaque fois qu'une rotation a lieu et que ce nombre est atteint, le fichier journal le plus ancien est supprimé.

Notes de mise à jour et historique des documents pour Amazon DCV Session Manager

Cette page fournit les notes de publication et l'historique des documents pour Amazon DCV Session Manager.

Rubriques

- Notes de mise à jour d'Amazon DCV Session Manager
- Historique de la documentation

Notes de mise à jour d'Amazon DCV Session Manager

Cette section fournit un aperçu des principales mises à jour, versions de fonctionnalités et corrections de bogues pour Amazon DCV Session Manager. Toutes les mises à jour sont organisées par date de sortie. Nous mettons fréquemment à jour la documentation pour répondre aux commentaires que vous nous envoyez.

Rubriques

- 2025.0-561 — 25 juin 2026
- 2025.0-544 — 2 février 2026
- 2025.0-544 — 23 décembre 2025
- 2025.0-539 — 12 novembre 2025
- 2025.0-539 — 22 octobre 2025
- 2024.0-531 — 17 juin 2025
- 2024.0-504— 31 mars 2025
- 2024.0-493— 15 janvier 2025
- 2024.0-457— 1er octobre 2024
- 2023.1-17652 — 1er août 2024
- 2023.1-16388 — 26 juin 2024
- 2023.1 — 9 novembre 2023
- 2023.0-15065— 4 mai 2023
- 2023.0-14852 — 28 mars 2023

- 2022.2-13907 — 11 novembre 2022
- 2022.1-13067 — 29 juin 2022
- 2022.0-11952 — 23 février 2022
- 2021.3-11591 — 20 décembre 2021
- 2021.2-11445 — 18 novembre 2021
- 2021.2-11190 — 11 octobre 2021
- 2021.2-11042 — 1er septembre 2021
- 2021.1-10557 — 31 mai 2021
- 2021.0-10242 — 12 avril 2021
- 2020.2-9662 — 04 décembre 2020
- 2020.2-9508 — 11 novembre 2020

2025.0-561 — 25 juin 2026

Numéros de version	Modifications et correctifs de bogues
• Courtier : 561	• Mise à niveau d'Apache Ignite vers la version 2.18.0. • Correction de la dépendance au package broker pour Ubuntu 24.04 afin d'installer la version d'exécution Java correcte.

2025.0-544 — 2 février 2026

Numéros de version	Modifications et correctifs de bogues
• Courtier : 544 • Agent : 902 • CLI : 159	• Ajout d'un paramètre `preferred_network_interface` de configuration pour la détection de l'adresse IP de l'hôte.

2025.0-544 — 23 décembre 2025

Numéros de version	Modifications et correctifs de bogues
• Courtier : 544 • Agent : 893 • CLI : 159	• Augmentation de la taille limite des WebSocket messages pour résoudre les échecs de récupération de captures d'écran sur les hôtes macOS. • Environnement de génération Windows mis à jour vers Visual Studio 2022.

2025.0-539 — 12 novembre 2025

Numéros de version	Modifications et correctifs de bogues
• Courtier : 539 • Agent : 888 • CLI : 159	• Identifiant du bundle macOS Agent renommé de NICE Software à Amazon.

2025.0-539 — 22 octobre 2025

Numéros de version	Modifications et correctifs de bogues
• Courtier : 539 • Agent : 886 • CLI : 159	• Le paramètre de configuration enable_query_logged_in_users a été ajouté au fichier de configuration de l'agent pour spécifier le comportement des requêtes des utilisateurs connectés sur les systèmes Windows. • PowerShell Les commandes ont été remplacées par des API Windows natives (WMI et registre Windows) pour améliorer les performances et la fiabilité lors de la récupération des informations système. • Correction de la résolution des noms DNS sur les instances Windows Amazon EC2 en améliorant la détection Amazon EC2 avec un retour au service de métadonnées en cas d'échec AWS de la détection. UUID-based

Numéros de version	Modifications et correctifs de bogues
	• Version mise à jour jusqu'en 2025.

2024.0-531 — 17 juin 2025

Numéros de version	Modifications et correctifs de bogues
• Courtier : 531 • Agent : 852 • CLI : 154	• Ajout d'une fonctionnalité permettant de renouveler les certificats avant leur expiration. • Rebaptisé NICE DCV en Amazon DCV. • Correctifs de bogue.

2024.0-504— 31 mars 2025

Numéros de version	Modifications et correctifs de bogues
• Courtier : 504 • Agent : 817 • CLI : 154	• Ajout du support pour AL2023. • Correctifs de bogues et améliorations de performances

2024.0-493— 15 janvier 2025

Numéros de version	Modifications et correctifs de bogues
• Courtier : 493 • Agent : 801 • CLI : 152	• Des paramètres ont été ajoutés à la `GetSessionScreenshot` demande pour spécifier la hauteur et la largeur maximales de la capture d'écran. • Ajout d'un paramètre au fichier de configuration du courtier qui indique le nombre de secondes après lesquelles les sessions sur un serveur Amazon DCV inaccessible sont supprimées du système.

Numéros de version	Modifications et correctifs de bogues
	• Problème résolu : le `seconds-before-deleting-unreachable-dcv-server` paramètre du fichier de configuration du broker n'était pas respecté. • Correctifs de bogues et améliorations de performances

2024.0-457— 1er octobre 2024

Numéros de version	Modifications et correctifs de bogues
• Courtier : 457 • Agent : 748 • CLI : 140	• Rebaptisé NICE DCV en Amazon DCV. • Ajout du support pour Ubuntu 24.04.

2023.1-17652 — 1er août 2024

Numéros de version	Modifications et correctifs de bogues
• Courtier : 426 • Agent : 748 • CLI : 140	• Correctifs de bogues et améliorations de performances

2023.1-16388 — 26 juin 2024

Numéros de version	Modifications et correctifs de bogues
• Courtier : 417 • Agent : 748 • CLI : 140	• Correction d'un bug qui affichait incorrectement la mémoire en To, et non en Go. • Correctifs de bogues et améliorations de performances

2023.1 — 9 novembre 2023

Numéros de version	Modifications et correctifs de bogues
• Courtier : 410 • Agent : 732 • CLI : 140	• Correctifs de bogues et améliorations de performances

2023.0-15065— 4 mai 2023

Numéros de version	Modifications et correctifs de bogues
• Courtier : 392 • Agent : 675 • CLI : 132	• Ajout du support pour Red Hat Enterprise Linux 9, Rocky Linux 9 et CentOS Stream 9 sur les plateformes ARM.

2023.0-14852 — 28 mars 2023

Numéros de version	Modifications et correctifs de bogues
• Courtier : 392 • Agent : 642 • CLI : 132	• Ajout du support pour Red Hat Enterprise Linux 9, Rocky Linux 9 et CentOS Stream 9.

2022.2-13907 — 11 novembre 2022

Numéros de version	Modifications et correctifs de bogues
• Courtier : 382 • Agent : 612 • CLI : 123	• Ajout d'un `Substate` champ en `DescribeSessions` réponse. • Correction d'un problème qui empêchait la CLI de se connecter au broker en fonction de l'URL utilisée.

2022.1-13067 — 29 juin 2022

Numéros de version	Modifications et correctifs de bogues
• Courtier : 355 • Agent : 592 • CLI : 114	• Ajout du support pour exécuter le broker sur des instances AWS Graviton. • Ajout du support des agents et des courtiers pour Ubuntu 22.04.

2022.0-11952 — 23 février 2022

Numéros de version	Modifications et correctifs de bogues
• Courtier : 341 • Agent : 520 • CLI : 112	• Ajout de la fonctionnalité de rotation des journaux à l'agent. • Ajout d'un paramètre de configuration pour définir Java Home dans le Broker. • Amélioration du transfert des données du cache vers le disque dans le Broker. • Validation d'URL fixe dans la CLI.

2021.3-11591 — 20 décembre 2021

Numéros de version	Nouvelles fonctionnalités
• Courtier : 307 • Agent : 453 • CLI : 92	• Ajout de la prise en charge de l'intégration avec Amazon DCV Connection Gateway. • Ajout du support Broker pour Ubuntu 18.04 et Ubuntu 20.04.

2021.2-11445 — 18 novembre 2021

Numéros de version	Modifications et correctifs de bogues
• Courtier : 288 • Agent : 413 • CLI : 54	• Correction d'un problème de validation des noms de connexion incluant un domaine Windows.

2021.2-11190 — 11 octobre 2021

Numéros de version	Modifications et correctifs de bogues
• Courtier : 254 • Agent : 413 • CLI : 54	• Correction d'un problème dans l'interface de ligne de commande qui empêchait le lancement de sessions Windows.

2021.2-11042 — 1er septembre 2021

Numéros de version	Nouvelles fonctionnalités	Modifications et correctifs de bogues
• Courtier : 254 • Agent : 413 • CLI : 37	• Amazon DCV Session Manager prend désormais en charge l'interface de ligne de commande (CLI). Vous pouvez créer et gérer des sessions Amazon DCV dans la CLI, au lieu d'appeler des API. • Amazon DCV Session Manager a introduit la persistance des données Broker. Pour une disponibilité accrue, les courtiers peuvent conserver les informations sur l'état du serveur dans	• Lorsque vous enregistrez un serveur d'autorisation externe, vous pouvez désormais spécifier l'algorithme utilisé par le serveur d'autorisation pour signer les jetons JSON-formatted Web. Avec cette modification, vous pouvez utiliser Azure AD comme serveur d'autorisation externe.

Numéros de version	Nouvelles fonctionnalités	Modifications et correctifs de bogues
	un magasin de données externe et restaurer les données au démarrage.	

2021.1-10557 — 31 mai 2021

Numéros de version	Nouvelles fonctionnalités	Modifications et correctifs de bogues
• Courtier : 214 • Agent : 365	• Amazon DCV Session Manager a ajouté la prise en charge des paramètres d'entrée transmis au fichier autorun sous Linux. • Les propriétés du serveur peuvent désormais être transmises en tant qu'exigences à l'CreateSessionsAPI.	• Nous avons résolu un problème avec le fichier autorun sous Windows.

2021.0-10242 — 12 avril 2021

Numéros de version	Modifications et correctifs de bogues

• Courtier : 183
• Agent : 318
	• Amazon DCV Session Manager a introduit les nouvelles API suivantes :
	  • OpenServers
	  • CloseServers
	  • DescribeServers
	  • GetSessionScreenshots
	• Il a également introduit les nouveaux paramètres de configuration suivants :
	  • Paramètres du courtier : `session-screenshot-max-widt h session-screenshot-max-height ,session-s`

Numéros de version	Modifications et correctifs de bogues
	creenshot-format ,create-sessions-queue-max-size , etcreate-sessions-queue-max-time-seconds . • Paramètres de l'agent : agent.autorun_folder max_virtual_sessions , etmax_concurrent_sessions_per_user . Paramètres de l'agent : agent.autorun_folder max_virtual_sessions , etmax_concurrent_sessions_per_user . Paramètres de l'agent : agent.autorun_folder max_virtual_sessions , etmax_concurrent_sessions_per_user .

2020.2-9662 — 04 décembre 2020

Numéros de version	Modifications et correctifs de bogues
• Courtier : 114 • Agent : 211	• Nous avons résolu un problème lié aux certificats TLS générés automatiquement qui empêchaient le Broker de démarrer.

2020.2-9508 — 11 novembre 2020

Numéros de version	Modifications et correctifs de bogues
• Courtier : 78 • Agent : 183	• La version initiale d'Amazon DCV Session Manager.

Historique de la documentation

Le tableau suivant décrit la documentation de cette version d'Amazon DCV Session Manager.

Modifier	Description	Date
Version 2025.0-561 d'Amazon DCV	Le gestionnaire de session Amazon DCV a été mis à jour pour Amazon DCV 2025.0-561. Pour de plus amples informations, veuillez consulter 2025.0-561 — 25 juin 2026.	25 juin 2026
Version 2025.0-544 d'Amazon DCV	Le gestionnaire de session Amazon DCV a été mis à jour pour Amazon DCV 2025.0-544. Pour de plus amples informations, veuillez consulter 2025.0-544 — 2 février 2026.	2 février 2026
Version 2025.0-544 d'Amazon DCV	Le gestionnaire de session Amazon DCV a été mis à jour pour Amazon DCV 2025.0-544. Pour de plus amples informations, veuillez consulter 2025.0-544 — 23 décembre 2025.	23 décembre 2025
Version 2025.0-539 d'Amazon DCV	Le gestionnaire de session Amazon DCV a été mis à jour pour Amazon DCV 2025.0-539. Pour de plus amples informations, veuillez consulter 2025.0-539 — 12 novembre 2025.	12 novembre 2025
Version 2025.0-539 d'Amazon DCV	Le gestionnaire de session Amazon DCV a été mis à jour pour Amazon DCV 2025.0-539. Pour de plus amples informations, veuillez consulter 2025.0-539 — 22 octobre 2025.	22 octobre 2025
Version 2024.0-531 d'Amazon DCV	Le gestionnaire de session Amazon DCV a été mis à jour pour Amazon DCV 2024.0-531. Pour de plus amples informations, veuillez consulter 2024.0-531 — 17 juin 2025.	17 juin 2025

Modifier	Description	Date
Version 2024.0-504 d'Amazon DCV	Le gestionnaire de session Amazon DCV a été mis à jour pour Amazon DCV 2024.0-504. Pour de plus amples informations, veuillez consulter 2024.0-504— 31 mars 2025.	31 mars 2025
Version 2024.0-493 d'Amazon DCV	Le gestionnaire de session Amazon DCV a été mis à jour pour Amazon DCV 2024.0-493. Pour de plus amples informations, veuillez consulter 2024.0-493— 15 janvier 2025.	15 janvier 2025
Version 2024.0-457 d'Amazon DCV	Le gestionnaire de session Amazon DCV a été mis à jour pour Amazon DCV 2024.0-457. Pour de plus amples informations, veuillez consulter 2024.0-457— 1er octobre 2024.	30 septembre 2024
Version 2023.1-17652 d'Amazon DCV	Le gestionnaire de session Amazon DCV a été mis à jour pour Amazon DCV 2023.1-17652. Pour de plus amples informations, veuillez consulter 2023.1-17652 — 1er août 2024.	1er août 2024
Version 2023.1-16388 d'Amazon DCV	Amazon DCV Session Manager a été mis à jour pour Amazon DCV 2023.1-16388. Pour de plus amples informations, veuillez consulter 2023.1-16388 — 26 juin 2024.	26 juin 2024
Amazon DCV version 2023.1	Amazon DCV Session Manager a été mis à jour pour Amazon DCV 2023.1. Pour de plus amples informations, veuillez consulter 2023.1 — 9 novembre 2023.	9 novembre 2023

Modifier	Description	Date
Amazon DCV version 2023.0	Amazon DCV Session Manager a été mis à jour pour Amazon DCV 2023.0. Pour de plus amples informations, veuillez consulter 2023.0-14852 — 28 mars 2023.	28 mars 2023
Amazon DCV version 2022.2	Amazon DCV Session Manager a été mis à jour pour Amazon DCV 2022.2. Pour de plus amples informations, veuillez consulter 2022.2-13907 — 11 novembre 2022.	11 novembre 2022
Amazon DCV version 202.1	Amazon DCV Session Manager a été mis à jour pour Amazon DCV 2022.1. Pour de plus amples informations, veuillez consulter 2022.1-13067 — 29 juin 2022.	29 juin 2022
Amazon DCV version 2022.0	Amazon DCV Session Manager a été mis à jour pour Amazon DCV 2022.0. Pour de plus amples informations, veuillez consulter 2022.0-11952 — 23 février 2022.	23 février 2022
Amazon DCV version 2021.3	Amazon DCV Session Manager a été mis à jour pour Amazon DCV 2021.3. Pour de plus amples informations, veuillez consulter 2021.3-11591 — 20 décembre 2021.	20 décembre 2021
Amazon DCV version 2021.2	Amazon DCV Session Manager a été mis à jour pour Amazon DCV 2021.2. Pour de plus amples informations, veuillez consulter 2021.2-11042 — 1er septembre 2021.	01 septembre 2021

Modifier	Description	Date
Amazon DCV version 2021.1	Amazon DCV Session Manager a été mis à jour pour Amazon DCV 2021.1. Pour de plus amples informations, veuillez consulter 2021.1-10557 — 31 mai 2021.	31 mai 2021
Amazon DCV version 2021.0	Amazon DCV Session Manager a été mis à jour pour Amazon DCV 2021.0. Pour de plus amples informations, veuillez consulter 2021.0-10242 — 12 avril 2021.	12 avril 2021
Version initiale d'Amazon DCV Session Manager	Il s'agit de la première publication de ce contenu.	11 novembre 2020

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.