



Guide de l'administrateur

Gestionnaire de sessions DCV NICE



Gestionnaire de sessions DCV NICE: Guide de l'administrateur

Copyright © 2024 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
Limites	3
Tarification	4
Prérequis	4
Exigences en matière de réseau et de connectivité	5
Configuration	7
Étape 1 : Préparation des serveurs DCV NICE	7
Étape 2 : configurer le courtier	8
Étape 3 : Configuration de l'agent	10
Étape 4 : Configuration du serveur DCV NICE	15
Étape 5 : vérifier les installations	17
Vérifiez l'agent	17
Vérifiez le courtier	18
Configuration	19
Gestionnaire de sessions de dimensionnement	19
Étape 1 : Créer un profil d'instance	20
Étape 2 : préparer le certificat SSL pour l'équilibreur de charge	21
Étape 3 : Création de l'équilibreur de charge de l'application Broker	22
Étape 4 : Lancez les courtiers	23
Étape 5 : Création de l'équilibreur de charge de l'application Agent	24
Étape 6 : Lancer les agents	25
Utilisation de balises	27
Configuration d'un serveur d'autorisation externe	28
Configuration de la persistance des courtiers	33
Configurer le broker pour qu'il persiste sur DynamoDB	34
Configurer le broker pour qu'il persiste sur MariaDB/MySQL	35
Intégration à la passerelle de connexion NICE DCV	36
Configurer le Session Manager Broker en tant que résolveur de session pour la passerelle de connexion NICE DCV	37
Facultatif : activer l'authentification du client TLS	38
Serveur NICE DCV - Cartographie DNS	39
Intégration à Amazon CloudWatch	41

Mise à niveau	44
Mise à niveau de l'agent du gestionnaire de session NICE DCV	44
Mise à niveau du broker NICE DCV Session Manager	46
Référence Broker CLI	49
register-auth-server	50
Syntaxe	50
Options	50
Exemple	50
list-auth-servers	51
Syntaxe	50
Sortie	51
Exemple	50
unregister-auth-server	52
Syntaxe	50
Options	50
Sortie	51
Exemple	50
register-api-client	53
Syntaxe	50
Options	50
Sortie	51
Exemple	50
describe-api-clients	54
Syntaxe	50
Sortie	51
Exemple	50
unregister-api-client	56
Syntaxe	50
Options	50
Exemple	50
renew-auth-server-api-clé	57
Syntaxe	50
Exemple	50
generate-software-statement	57
Syntaxe	50
Sortie	51

Exemple	50
describe-software-statements	59
Syntaxe	50
Sortie	51
Exemple	50
deactivate-software-statement	60
Syntaxe	50
Options	50
Exemple	50
describe-agent-clients	61
Syntaxe	50
Sortie	51
Exemple	50
unregister-agent-client	62
Syntaxe	50
Options	50
Exemple	50
register-server-dns-mappings	63
Syntaxe	50
Options	50
Exemple	50
describe-server-dns-mappings	64
Syntaxe	50
Sortie	51
Exemple	50
Référence sur le fichier de configuration	67
Fichier de configuration d'un agent	67
Fichier de configuration de l'agent	84
Notes de mise à jour et historique des documents	91
Notes de mise à jour	91
2023.1-16388 — 26 juin 2024	92
2023.1 — 9 novembre 2023	92
2023.0-15065— 4 mai 2023	92
2023.0-14852 — 28 mars 2023	92
2022.2-13907 — 11 novembre 2022	93
2022.1-13067 — 29 juin 2022	93

2022.0-11952 — 23 février 2022	93
2021.3-11591 — 20 décembre 2021	94
2021.2-11445 — 18 novembre 2021	94
2021.2-11190 — 11 octobre 2021	94
2021.2-11042 — 1er septembre 2021	94
2021.1-10557 — 31 mai 2021	95
2021.0-10242 — 12 avril 2021	95
2020.2-9662 — 04 décembre 2020	96
.....	96
Historique de la documentation	97
.....	xcix

Qu'est-ce que NICE DCV Session Manager ?

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

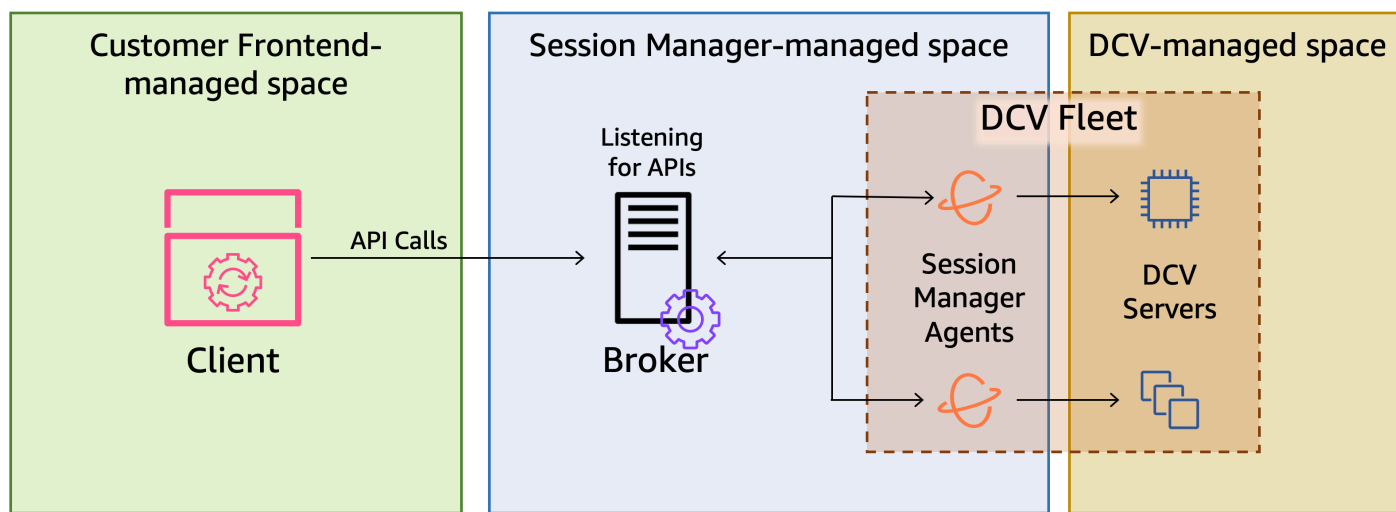
Ce guide explique comment installer et configurer l'agent et le courtier du gestionnaire de session. Pour plus d'informations sur l'utilisation des API du gestionnaire de session, consultez le guide du développeur du gestionnaire de session NICE DCV.

Rubriques

- [Comment fonctionne le gestionnaire de session](#)
- [Fonctionnalités](#)
- [Limites](#)
- [Tarification](#)
- [Exigences du gestionnaire de sessions NICE 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 les API du gestionnaire de session. Il reçoit et traite les demandes d'API pour gérer les sessions NICE DCV du client, puis transmet les instructions aux agents concernés. Le Broker doit être installé sur un hôte distinct de vos serveurs DCV NICE, mais il doit être accessible au client et il doit pouvoir accéder aux agents.

Agent

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

API

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

Client

Le client est l'application frontale ou le portail que vous développez pour appeler les API du gestionnaire de session exposées par le courtier. Les utilisateurs finaux utilisent le client pour gérer les sessions hébergées sur les serveurs DCV NICE 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 des API clientes enregistrées. 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 DCV NICE

Vous devez créer une session NICE DCV sur votre serveur NICE DCV à laquelle vos clients peuvent se connecter. Les clients ne peuvent se connecter à un serveur DCV NICE que s'il existe une session active. NICE DCV prend en charge les sessions de console et virtuelles. Vous

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

Limites

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

Tarifification

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

Les clients sur site ont besoin d'une licence NICE DCV Plus ou DCV Professional Plus. Pour plus d'informations sur l'achat d'une licence NICE DCV Plus ou NICE DCV Professional Plus, [consultez Comment acheter sur le site Web de NICE](#) et trouvez un distributeur ou un revendeur NICE dans votre région. Pour permettre à tous les clients locaux d'expérimenter le DCV Session Manager, les exigences de licence ne seront appliquées qu'à partir de la version 2021.0 de NICE DCV.

Pour plus d'informations, consultez la section [Licence du serveur DCV NICE](#) dans le Guide de l'administrateur NICE DCV.

Exigences du gestionnaire de sessions NICE DCV

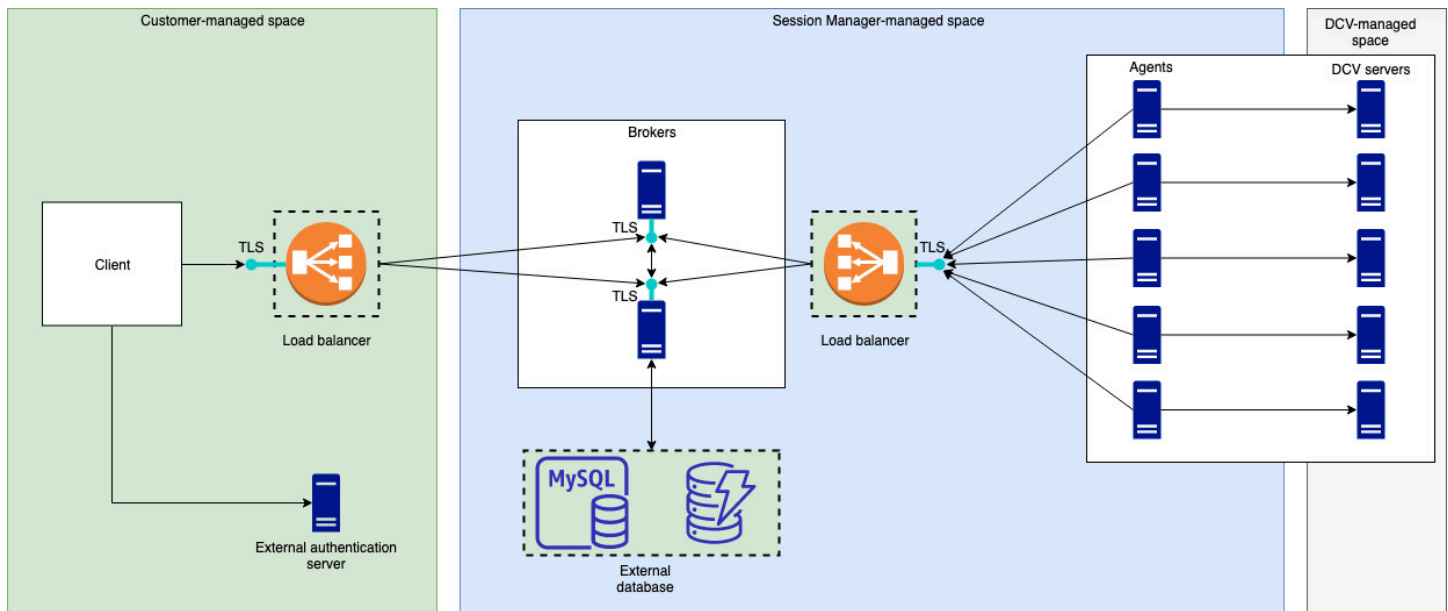
L'agent et le courtier du gestionnaire de session NICE DCV répondent aux exigences suivantes.

	Broker	Agent
Système d'exploitation	<ul style="list-style-type: none"> • Amazon Linux 2 • CentOS 7.6 ou version ultérieure • CentOS Stream 8 • CentOS Stream 9 • RHEL 7.6 ou version ultérieure • RHEL 8.x • RHEL 9.x • Rocky Linux 8.5 ou version ultérieure • Rocky Linux 9.x • Ubuntu 20.04 • Ubuntu 22.04 	<ul style="list-style-type: none"> • Windows <ul style="list-style-type: none"> • Windows Server 2022 • Windows Server 2019 • Windows Server 2016 • Serveur Linux <ul style="list-style-type: none"> • Amazon Linux 2 • CentOS 7.6 ou version ultérieure • CentOS Stream 8 • CentOS Stream 9 • RHEL 7.6 ou version ultérieure • RHEL 8.x • RHEL 9.x • Rocky Linux 8.5 ou version ultérieure

	Broker	Agent
		<ul style="list-style-type: none"> • Rocky Linux 9.x • Ubuntu 20.04 • Ubuntu 22.04 • SUSE Linux Enterprise 12 avec SP4 ou version ultérieure • SUSE Linux Enterprise 15
Architecture	<ul style="list-style-type: none"> • 64 bits x86 • ARM 64 bits 	<ul style="list-style-type: none"> • 64 bits x86 • ARM 64 bits (Amazon Linux 2, CentOS 7.x/8.x/9.x, RHEL 7.x/8.x/9.x et Rocky 8.x/9.x uniquement) • ARM 64 bits (Ubuntu 22.04)
Mémoire	8 Go	4 Go
Belle version DCV	NICE DCV 2020.2 et versions ultérieures	NICE 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 DCV NICE. 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 DCV de NICE. 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 connexions HTTPS sécurisées, persistantes et bidirectionnelles avec le Broker.

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

Configuration du gestionnaire de sessions DCV NICE

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 plus d'informations, consultez [Gestionnaire de sessions de dimensionnement](#).

Pour configurer le gestionnaire de session NICE DCV, procédez comme suit :

Étapes

- [Étape 1 : Préparation des serveurs DCV NICE](#)
- [Étape 2 : Configuration du broker NICE DCV Session Manager](#)
- [Étape 3 : Configuration de l'agent du gestionnaire de session NICE DCV](#)
- [Étape 4 : configurer le serveur NICE DCV pour utiliser le Broker comme serveur d'authentification](#)
- [Étape 5 : vérifier les installations](#)

Étape 1 : Préparation des serveurs DCV NICE

Vous devez disposer d'un parc de serveurs NICE DCV avec lesquels vous avez l'intention d'utiliser Session Manager. Pour plus d'informations sur l'installation des serveurs DCV NICE, consultez la section [Installation du serveur DCV NICE dans le](#) guide de l'administrateur NICE DCV.

Sur les serveurs DCV Linux NICE, le gestionnaire de session utilise un nom d'utilisateur du service local. `dcvsmagent` Cet utilisateur est automatiquement créé lors de l'installation de l'agent du gestionnaire de session. Vous devez accorder à ce service des privilèges d'administrateur pour NICE 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 Linux NICE DCV

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 DCV NICE.

Le gestionnaire de session est uniquement en mesure de créer des sessions DCV NICE pour le compte d'utilisateurs qui existent déjà sur le serveur DCV NICE. 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 DCV NICE.

Tip

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

Étape 2 : Configuration du broker NICE DCV Session Manager

Le Broker doit être installé sur un hôte Linux. Pour plus d'informations sur les distributions Linux prises en charge, consultez [Exigences du gestionnaire de sessions NICE DCV](#). Installez le Broker sur un hôte distinct de l'agent et de l'hôte du serveur NICE 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 packages de vérifier la signature du package, vous devez importer la clé NICE GPG. Exécutez la commande suivante pour importer la clé NICE 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, RHEL 7.x et CentOS 7.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerBrokers/nice-dcv-session-manager-broker-2023.1.410-1.el7.noarch.rpm
```

- RHEL 8.x, CentOS Stream 8 et Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerBrokers/nice-dcv-session-manager-broker-2023.1.410-1.el8.noarch.rpm
```

- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerBrokers/nice-dcv-session-manager-broker_2023.1.410-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerBrokers/nice-dcv-session-manager-broker_2023.1.410-1_all.ubuntu2204.deb
```

4. Installez le package .

- Amazon Linux 2, RHEL 7.x et CentOS 7.x

```
$ sudo yum install -y ./nice-dcv-session-manager-broker-2023.1.410-1.el7.noarch.rpm
```

- RHEL 8.x, Stream CentOS 8 et Rocky Linux 8.x

```
$ sudo yum install -y ./nice-dcv-session-manager-broker-2023.1.410-1.el8.noarch.rpm
```

- Ubuntu 20.04

```
$ sudo apt install -y ./nice-dcv-session-manager-broker_2023.1.410-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ sudo apt install -y ./nice-dcv-session-manager-broker_2023.1.410-1_all.ubuntu2204.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 la section [Fichier de configuration du courtier](#).

6. Démarrez le service Broker 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 : Configuration de l'agent du gestionnaire de session NICE DCV

L'agent doit être installé sur tous les hôtes du serveur DCV NICE du parc. L'agent peut être installé sur des serveurs Windows et Linux. Pour plus d'informations sur les systèmes d'exploitation pris en charge, consultez [Exigences du gestionnaire de sessions NICE DCV](#).

Prérequis

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

Linux host

Note

L'agent du gestionnaire de session 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 packages de vérifier la signature du package, vous devez importer la clé NICE GPG. Exécutez la commande suivante pour importer la clé NICE 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, RHEL 7.x et CentOS 7.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerAgents/nice-dcv-session-manager-agent-2023.1.732-1.el7.x86_64.rpm
```

- RHEL 8.x, CentOS Stream 8 et Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerAgents/nice-dcv-session-manager-agent-2023.1.732-1.el8.x86_64.rpm
```

- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerAgents/nice-dcv-session-manager-agent_2023.1.732-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerAgents/nice-dcv-session-manager-agent_2023.1.732-1_amd64.ubuntu2204.deb
```

- SUSE Linux Enterprise 12

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerAgents/nice-dcv-session-manager-agent-2023.1.732-1.sles12.x86_64.rpm
```

- SUSE Linux Enterprise 15

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerAgents/nice-dcv-session-manager-agent-2023.1.732-1.sles15.x86_64.rpm
```

3. Installez le package .

- Amazon Linux 2, RHEL 7.x et CentOS 7.x

```
$ sudo yum install -y ./nice-dcv-session-manager-agent-2023.1.732-1.el7.x86_64.rpm
```

- RHEL 8.x, CentOS Stream 8 et Rocky Linux 8.x

```
$ sudo yum install -y ./nice-dcv-session-manager-agent-2023.1.732-1.el8.x86_64.rpm
```

- Ubuntu 20.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2023.1.732-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2023.1.732-1_amd64.ubuntu2204.deb
```


- SUSE Linux Enterprise 12

```
$ sudo zypper install ./nice-dcv-session-manager-agent-2023.1.732-1.sles12.x86_64.rpm
```

- SUSE Linux Enterprise 15

```
$ sudo zypper install ./nice-dcv-session-manager-agent-2023.1.732-1.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.
 - `Broker_host`, spécifiez le nom DNS de l'hôte sur lequel le broker est installé.

 Important

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

- (Facultatif) `Broker_port`, spécifiez le port par lequel communiquer avec le broker. Par défaut, l'agent et le courtier 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.
- `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 `surtls_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 instance Amazon EC2, `broker_host` vous devez spécifier l'adresse IPv4 privée de l'instance.

- (Facultatif) Pour `broker_port`, spécifiez le port par lequel communiquer avec le broker. Par défaut, l'agent et le courtier 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 `surtls_strict.false`

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

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

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

Étape 4 : configurer le serveur NICE DCV pour utiliser le Broker comme serveur d'authentification

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

Linux NICE DCV server

Pour ajouter l'utilisateur du service local pour les serveurs Linux NICE 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 Broker au format suivant : `https://broker_ip_or_dns:port/agent/validate-authentication-token`. Spécifiez le port utilisé pour la communication Broker-Agent, qui est 8445 par défaut. Si vous exécutez le Broker sur une instance Amazon EC2, 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 DCV NICE. Pour plus d'informations, consultez les [sections Arrêt du serveur DCV NICE et Démarrage du serveur DCV NICE dans le Guide de l'administrateur NICE DCV](#).

Windows NICE DCV server

Sur les serveurs Windows NICE DCV

1. Ouvrez l'éditeur de registre Windows et accédez à la clé HKEY_USERS/S-1-5-18/Software/GSettings/com/NiceSoftware/DCV/Security/.
2. Ouvrez le paramètre ca-file. 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`.

3. Ouvrez le `auth-token-verifier` paramètre. 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`
Spécifiez le port utilisé pour la communication Broker-Agent, qui est 8445 par défaut. Si vous exécutez le Broker sur une instance Amazon EC2, 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`.

4. Choisissez OK, puis fermez l'Éditeur du Registre Windows.
5. Arrêtez et redémarrez le serveur DCV NICE. Pour plus d'informations, consultez les [sections Arrêt du serveur DCV NICE et Démarrage du serveur DCV NICE dans le Guide de l'administrateur NICE DCV](#).

Étape 5 : vérifier les installations

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.

Hôte de l'agent Linux

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 un résultat différent, 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 sessions NICE DCV

Cette section explique comment effectuer la configuration avancée de Session.

Rubriques

- [Gestionnaire de sessions de dimensionnement](#)
- [Utilisation de balises pour cibler les serveurs NICE DCV](#)
- [Configuration d'un serveur d'autorisation externe](#)
- [Configuration de la persistance des courtiers](#)
- [Intégration à la passerelle de connexion NICE 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 des AMI.

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 les API Elastic Load Balancing. Pour plus d'informations, consultez la section [Rôles IAM pour Amazon EC2](#) dans le guide de l'utilisateur Amazon EC2.

Pour créer un profil d'instance

1. Créez un rôle AWS Identity and Access Management (IAM) qui définit les autorisations à utiliser dans le profil d'instance. Utilisez la politique de confiance suivante :

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

Joignez ensuite la politique suivante :

```
{
  "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.AWS CLI
3. Ajoutez le rôle IAM au profil d'instance. Pour plus d'informations, consultez la section [add role-to-instance-profile](#) dans la référence des commandes.AWS CLI
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 HTTP/1.
 - 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, consultez [Étape 2 : Configuration du broker NICE 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 HTTP/1.
 - 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 DCV NICE. Pour plus d'informations, consultez [Étape 1 : Préparation des serveurs DCV NICE](#).
2. Placez une copie de la clé publique CA créée dans [Étape 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 CA 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, consultez [Étape 3 : Configuration de l'agent du gestionnaire de session NICE DCV](#).

⚠ 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 DCV NICE pour utiliser le Broker comme serveur d'authentification. Pour plus d'informations, consultez [Étape 4 : configurer le serveur NICE DCV pour utiliser le Broker comme serveur d'authentification](#).

⚠ Important

Lors de la modification du fichier de configuration du serveur NICE 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 NICE DCV

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

Une balise se compose d'une paire clé et 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 dont les hôtes disposent de 4 Go de mémoire `ram=4GB`. Vous pouvez également étiqueter les agents en fonction de leur objectif. Par exemple, vous pouvez baliser tous les agents exécutés sur des hôtes de production avec `purpose=production`.

Pour attribuer des balises à un agent

1. À l'aide de l'éditeur de texte de votre choix, créez un fichier et nommez-le de façon descriptive, 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 de clés et de valeurs 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 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 dans l'ordre alphabétique. Si plusieurs fichiers contiennent une étiquette 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
```

Configuration d'un serveur d'autorisation externe

Le serveur d'autorisation est le serveur chargé d'authentifier et d'autoriser les SDK et les agents clients.

Par défaut, Session Manager utilise le Broker comme serveur d'autorisation pour générer des jetons d'accès OAuth 2.0 pour les SDK clients et des instructions logicielles 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 Amazon Cognito](#).

Pour utiliser Amazon Cognito comme serveur d'autorisation

1. Créez un groupe d'utilisateurs Amazon Cognito. Pour plus d'informations sur les groupes d'utilisateurs, consultez les [fonctionnalités d'Amazon Cognito](#) dans le Guide du développeur Amazon Cognito.

Utilisez la [create-user-pool](#) commande et spécifiez le nom du pool et la région dans laquelle 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": "15hhd8jjj74hf32f24uEXAMPLE",  
    "LastModifiedDate": 1602510048.054,  
    "CreationDate": 1602510048.054,  
    "RefreshTokenValidity": 30,  
    "AllowedOAuthFlowsUserPoolClient": false  
  }  
}
```

Notez le `userPoolId`, car vous en aurez besoin lors de l'étape suivante.

2. Créez un nouveau domaine pour votre groupe d'utilisateurs. Utilisez la [create-user-pool-domain](#) commande et spécifiez un nom de domaine et celui `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 le créons 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 pool 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-client](#) commande et spécifiez le `userPoolId` groupe d'utilisateurs que vous avez créé, le nom du client et la région dans laquelle le créer. Ajoutez l'option `--generate-secret` qui permet de spécifier que vous souhaitez générer un secret pour le client de pool d'utilisateurs que vous créez.

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": "219273hp6k2ut5cugg9EXAMPLE",
    "ClientSecret": "1vp5e8nec7cbf4m9me55mbmht91u61h1h0a78rq1qki11EXAMPLE",
    "LastModifiedDate": 1602510291.498,
    "CreationDate": 1602510291.498,
    "RefreshTokenValidity": 30,
    "AllowedOAuthFlowsUserPoolClient": false
  }
}

```

Note

Prenez note du `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-server](#) commande et spécifiez le `userPoolId` groupe d'utilisateurs, un identifiant et un nom uniques 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 l'étendue.

```
$ aws cognito-idp create-resource-server --user-pool-id us-east-1_QLEXAMPLE
--identifiant 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 le client du groupe d'utilisateurs.

Utilisez la commande [update-user-pool-client](#). Spécifiez le `userPoolId` pool d'utilisateurs, le `clientId` du pool d'utilisateurs et la région. Pour `--allowed-o-auth-flows`, spécifiez `client_credentials` pour indiquer que le client doit obtenir des jetons d'accès à partir du 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 l'étendue comme suit `:resource_server_identifiant/scope_name`. Ajoutez le `--allowed-o-auth-flows-user-pool-client` pour indiquer que le client est autorisé à suivre le protocole OAuth lors de l'interaction avec les groupes d'utilisateurs Cognito.

```
$ aws cognito-idp update-user-pool-client --user-pool-id us-east-1_QLEXAMPLE --
client-id 219273hp6k2ut5cugg9EXAMPLE --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": "219273hp6k2ut5cugg9EXAMPLE",
    "ClientSecret": "1vp5e8nec7cbf4m9me55mbmht91u61hlh0a78rq1qki11EXAMPLE",
    "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 les jetons d'accès. Dans cet exemple, l'URL du serveur d'autorisation est `https://cognito-idp.us-east-1.amazonaws.com/us-east-1_QLEXAMPLE/.well-known/jwks.json`.

6. Tester la configuration.

```
$ curl -H "Authorization: Basic `echo -
n 219273hp6k2ut5cugg9EXAMPLE:1vp5e8nec7cbf4m9me55mbmht91u61hlh0a78rq1qki11EXAMPLE
| 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": "eyJraWQiOiJGQ0VaRFPJUUptT3NSaW41MmtqaDdEbTZYb0RnSTQ5b2VUT0cxUU11Q2VJPSIsImFzciI6ImF1dG8iLCJhdWQiOiJkZXZlZG50dZtId5dThkpEZiSx0YwiiWe9crAlqoaz1DcCsUJHIXDtGKW64pSj3-uQQGg1Jv_tyVjhrA4JbD0k67WS2V9NW-uZ7t4zwwaUm0i3KzpBmi54fpVgPaewiVlUm_aS4LUFcWT6hVJjiZF7om7984qb2g0a14iZxpXPBJTZX_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 client et l'URL du serveur générés ici. Pour plus d'informations sur la demande de jetons d'accès, voir [Création, obtention d'un jeton d'accès et demande d'API](#) dans le Guide du développeur du gestionnaire de session NICE DCV.

Configuration de la persistance des courtiers

Les courtiers Session Manager prennent en charge l'intégration avec des bases de données externes. La base de données externe permet à Session Manager 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 des informations concernant le serveur NICE DCV ouvert ou fermé.

Les types d'informations suivants peuvent être configurés pour persister :

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

Le gestionnaire de session NICE DCV 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 en savoir plus sur la tarification de DynamoDB, consultez la section [Tarification de la capacité provisionnée](#).

Configurer le broker pour qu'il persiste sur DynamoDB

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

1. Ouvrez le fichier `etc/dcv-session-manager-broker/session-manager-broker.properties` à l'aide de l'éditeur de texte de votre choix et effectuez ce qui suit :
 - 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 provisionnée de 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 provisionnée de DynamoDB](#).
 - Pour `dynamodb-table-name-prefix` spécifier le préfixe qui est 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 agents dans le cluster. Exécutez la commande suivante pour chaque agent :


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

3. Assurez-vous que tous les courtiers du cluster sont arrêtés, puis redémarrez-les tous. Démarrez chaque agent à l'aide de la commande suivante :

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

L'hôte courtier doit être autorisé à appeler les API DynamoDB. Sur les instances Amazon EC2, les informations d'identification sont automatiquement récupérées à l'aide du service de métadonnées Amazon EC2. 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 `dcvsmbrokerutilisateur`. 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 le fichier `/etc/dcv-session-manager-broker/session-manager-broker.properties` à l'aide de l'éditeur de texte de votre choix et effectuez ce qui suit :
 - 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 la 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 agents dans le cluster. Exécutez la commande suivante pour chaque agent :

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

3. Assurez-vous que tous les courtiers du cluster sont arrêtés, puis redémarrez-les tous. Exécutez la commande suivante pour chaque agent :

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

Intégration à la passerelle de connexion NICE DCV

La [passerelle de connexion NICE DCV](#) est un progiciel installable qui permet aux utilisateurs d'accéder à un parc de serveurs NICE DCV via un point d'accès unique à un réseau local ou à un VPC.

Si votre infrastructure inclut des serveurs NICE DCV accessibles via la passerelle de connexion NICE DCV, vous pouvez configurer le gestionnaire de session pour intégrer la passerelle de connexion NICE DCV. 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 courtier 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 à l'acheminement des connexions NICE DCV vers l'hôte sélectionné par le courtier.

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

Gestionnaire de session, côté Broker

1. Ouvrez le fichier `/etc/dcv-session-manager-broker/session-manager-broker.properties` à l'aide de l'éditeur de texte de votre choix et appliquez les modifications suivantes :
 - Définir `enable-gateway = true`
 - `gateway-to-broker-connector-https-port` Défini sur un port TCP libre (8447 par défaut)
 - Définissez `gateway-to-broker-connector-bind-host` l'adresse IP de l'hôte sur lequel le Broker se lie pour les connexions au Gestionnaire de connexion NICE DCV (0.0.0 par défaut)
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-la dans votre répertoire utilisateur.

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

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

Côté passerelle de connexion NICE DCV

- Veuillez suivre la [section](#) de la documentation de la passerelle de connexion NICE DCV.

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

Facultatif : activer 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 également aux Session Manager Brokers de valider 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 décrites dans 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 (ALB) ou les équilibreurs de charge de passerelle (GLB). Seuls les équilibreurs de charge sans terminaison TLS peuvent être pris en charge, tels que les équilibreurs de charge réseau (NLB). Si vous utilisez des ALB ou des GLB, 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. Pour plus d'informations sur les groupes de sécurité, cliquez ici : [Groupes de sécurité pour votre VPC](#)

Gestionnaire de session, côté Broker

1. Pour activer l'authentification du client TLS pour la communication entre les Session Manager Brokers et la passerelle de connexion NICE DCV, 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 de la passerelle de connexion NICE DCV dans votre répertoire utilisateur. Vous en aurez besoin lorsque vous activerez l'authentification du client TLS dans la passerelle de connexion NICE DCV 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
```

- Ouvrez ensuite `dcv-session-manager-broker /etc/session-manager-broker .properties` à l'aide de l'éditeur de texte de votre choix et procédez comme suit :
 - Réglez `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
- 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 passerelle de connexion NICE DCV

- Veuillez suivre la [section](#) de la documentation de la passerelle de connexion NICE DCV.
 - 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 NICE DCV Serveur NICE DCV - mappage DNS

La passerelle de connexion NICE DCV nécessite les noms DNS des serveurs NICE DCV pour se connecter aux instances du serveur DCV. Cette section montre 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 agentServerId IPAddress et InstanceId :

Ip :

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

Id :

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

Host.Aws.Ec2InstanceId :

Disponible uniquement pour les instances Amazon EC2, il identifie une machine de manière unique ; il ne change pas après le redémarrage d'une 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 NICE DCV du réseau.

DnsNames :

L'objet contenant les noms DNS associés au serveur NICE DCV contiendra :

InternalDnsNames :

Nom DNS utilisé par la passerelle de connexion NICE DCV 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 la page de commandes : [register-server-dns-mapping](#)) et `describe-server-dns-mappings` pour répertorier les mappages actuellement chargés dans le Session Manager Broker (référence de la page de commandes : [describe-server-dns-mappings](#)).

Persistance

Nous vous recommandons vivement d'activer la fonctionnalité de persistance du Session Manager Broker, afin de vous protéger contre la perte de mappage en cas de panne de plusieurs courtiers ou de l'ensemble du cluster. 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 for Brokers s'exécutant sur des instances Amazon EC2, ainsi qu'avec les Brokers exécutés sur des hôtes locaux.

Amazon CloudWatch contrôle vos ressources Amazon Web Services (AWS) et les applications que vous exécutez sur AWS en temps réel. Vous pouvez utiliser CloudWatch pour recueillir et suivre les métriques, qui sont des variables que vous pouvez mesurer pour vos ressources et 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`—Quantité de mémoire de tas 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 terminer les demandes `DescribeSessions` d'API.
- `Delete sessions request time`—Le temps nécessaire pour terminer les demandes `DeleteSessions` d'API.
- `Create sessions request time`—Le temps nécessaire pour terminer les demandes `CreateSessions` d'API.
- `Get session connection data request time`—Le temps nécessaire pour terminer les demandes `GetSessionConnectionData` d'API.
- `Update session permissions request time`—Le temps nécessaire pour terminer les demandes `UpdateSessionPermissions` d'API.

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

1. Ouvrez le fichier `/etc/dcv-session-manager-broker/session-manager-broker.properties` à l'aide de l'éditeur de texte de votre choix et procédez comme suit :
 - Réglez `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 instance Amazon EC2, ce paramètre est facultatif. La région est automatiquement récupérée 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 Broker doit également être autorisé à appeler l'`cloudwatch:PutMetricDataAPI`. 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 [Récupération des informations](#), [consultez Récupération des AWS informations](#), [consultez Récupération](#)

Mise à niveau du gestionnaire de sessions DCV NICE

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 en cas d'introduction de nouvelles fonctionnalités.

Rubriques

- [Mise à niveau de l'agent du gestionnaire de session NICE DCV](#)
- [Mise à niveau du broker NICE DCV Session Manager](#)

Mise à niveau de l'agent du gestionnaire de session NICE DCV

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, et pour Ubuntuarch64, remplacez amd64 par. 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, RHEL 7.x et CentOS 7.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerAgents/nice-dcv-session-manager-agent-2023.1.732-1.el7.x86_64.rpm
```

- RHEL 8.x, CentOS Stream 8 et Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerAgents/nice-dcv-session-manager-agent-2023.1.732-1.el8.x86_64.rpm
```

- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerAgents/nice-dcv-session-manager-agent_2023.1.732-1_amd64.ubuntu2004.deb
```

- SUSE Linux Enterprise 12

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerAgents/nice-dcv-session-manager-agent-2023.1.732-1.sles12.x86_64.rpm
```

- SUSE Linux Enterprise 15

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerAgents/nice-dcv-session-manager-agent-2023.1.732-1.sles15.x86_64.rpm
```

3. Installez le package .

- Amazon Linux 2, RHEL 7.x et CentOS 7.x

```
$ sudo yum install -y nice-dcv-session-manager-agent-2023.1.732-1.el7.x86_64.rpm
```

- RHEL 8.x, CentOS Stream 8 et Rocky Linux 8.x

```
$ sudo yum install -y nice-dcv-session-manager-agent-2023.1.732-1.el8.x86_64.rpm
```

- Ubuntu 20.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2023.1.732-1_amd64.ubuntu2004.deb
```

- SUSE Linux Enterprise 12

```
$ sudo zypper install nice-dcv-session-manager-agent-2023.1.732-1.sles12.x86_64.rpm
```

- SUSE Linux Enterprise 15

```
$ sudo zypper install nice-dcv-session-manager-agent-2023.1.732-1.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 Agent. Exécutez les commandes suivantes à l'invite de commandes.

```
C:\> sc start 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 Agent. Exécutez les commandes suivantes à l'invite de commandes.

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

Mise à niveau du broker NICE DCV Session Manager

Pour mettre à niveau le Broker

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

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

3. Téléchargez le package d'installation.

- Amazon Linux 2, RHEL 7.x et CentOS 7.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerBrokers/nice-dcv-session-manager-broker-2023.1.410-1.el7.noarch.rpm
```

- RHEL 8.x, CentOS Stream 8 et Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerBrokers/nice-dcv-session-manager-broker-2023.1.410-1.el8.noarch.rpm
```

- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/SessionManagerBrokers/nice-dcv-session-manager-broker-2023.1.410-1_all.ubuntu2004.deb
```

4. Installez le package .

- Amazon Linux 2, RHEL 7.x et CentOS 7.x

```
$ sudo yum install -y nice-dcv-session-manager-broker-2023.1.410-1.el7.noarch.rpm
```

- RHEL 8.x, CentOS Stream 8 et Rocky Linux 8.x

```
$ sudo yum install -y nice-dcv-session-manager-broker-2023.1.410-1.el8.noarch.rpm
```

- Ubuntu 20.04

```
$ sudo apt install -y nice-dcv-session-manager-broker-2023.1.410-1_all.ubuntu2004.deb
```

5. Démarrez le service Broker 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

Cette section explique comment utiliser les commandes de l'interface de ligne de commande (CLI) Broker.

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 Session Manager Broker 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 fichier de mappage des noms DNS et serveur DCV.

- [register-server-dns-mappings](#)
- [describe-server-dns-mappings](#)

register-auth-server

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

Par défaut, Session Manager utilise le Broker 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 l'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

Les URL des serveurs d'authentification externes enregistrés.

Exemple

L'exemple suivant répertorie tous les serveurs d'authentification externes 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

Annule l'enregistrement d'un serveur d'authentification externe. Une fois que vous avez annulé l'enregistrement d'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 à désenregistrer. Vous devez l'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 Broker 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 emplacement sûr. Elles ne seront pas récupérées plus tard.

Cette commande n'est utilisée que 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 Session Manager.

Type : String

Obligatoire : oui

Sortie

client-id

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

client-password

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: NjVmZDRlN2ItNjNmYS00M2QxLWF1ZmMtZmNmMDNkMEXAMPLE
```

describe-api-clients

Répertorie les clients Session Manager enregistrés avec le Broker.

Rubriques

- [Syntaxe](#)
- [Sortie](#)
- [Exemple](#)

Syntaxe

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

Sortie

name

Nom unique du client Session Manager.

id

ID unique du client Session Manager.

active

Indique l'état du client Session Manager. Si le client est actif, la valeur est `true` ; sinon, elle est `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 les 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

L'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 Broker pour signer les jetons d'accès OAuth 2.0 qui sont vendus 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 instruction logicielle pour s'inscrire auprès du courtier. Une fois que l'agent dispose

d'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 instruction logicielle par défaut lors de leur première installation. Vous pouvez continuer d'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 vers `leagent.software_statement_path` paramètre dans `leagent.conf` fichier. Une fois cela fait, arrêtez et redémarrez l'agent afin qu'il puisse utiliser la nouvelle instruction logicielle pour s'enregistrer auprès du Broker.

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:
```

```
ewogICJpZCIgOiAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkyMDcxIiwKICAiYWN0aXZlIiA6IHRydWUsCi
```

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 l'instruction logicielle. `true` si l'instruction logicielle est active ; sinon, elle l'est `false`.

Exemple

L'exemple suivant génère une instruction logicielle.

Commande

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


Exemple

L'exemple suivant désactive une instruction logicielle.

Commande

```
sudo -u root dcv-session-manager-broker deactivate-software-statement --software-statement EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj
```

Sortie

```
Software statement  
EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj  
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

ID unique de l'agent.

active

L'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

L'ID de l'agent à désinscrire.

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 mappages de serveurs DCV et 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 et serveurs DCV.

Type : String

Obligatoire : oui

Exemple

L'exemple suivant enregistre les mappages de noms DNS et serveurs DCV 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 mappages de noms DNS et serveurs DCV actuellement disponibles.

Syntaxe

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

Sortie

serverIdType

Type de l'identifiant du serveur.

serverId

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 mappages de noms DNS et de serveurs DCV enregistrés.

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 sur le fichier de configuration

Cette section fournit des informations sur les fichiers de configuration de l'agent et du agent.

Rubriques

- [Fichier de configuration d'un agent](#)
- [Fichier de configuration de l'agent](#)

Fichier de configuration d'un agent

Le fichier de configuration du Broker (`/etc/dcv-session-manager-broker/session-manager-broker.properties`) inclut des paramètres qui peuvent être configurés pour personnaliser la fonctionnalité 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 `dcvsmbrokerutilisateur`. Le Broker vérifie au démarrage que le fichier possède les autorisations attendues.

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

Nom du paramètre	Obligatoire	Valeur par défaut	Description
<code>broker.java.home</code>	Non		Spécifie le chemin d'accès au répertoire de base Java que le Broker utilisera à la place du répertoire par défaut du système. S'il est défini, le Broker l'utilise

Nom du paramètre	Obligatoire	Valeur par défaut	Description
			<p>ra<broker-java-home>/bin/java au démarrage.</p> <p>Conseil : le Broker nécessite l'environnement d'exécution Java 11 et il est installé s'il est manquant, car cela dépend de la réussite de l'installation. Si la version 11 n'est pas définie comme environnement Java par défaut, son répertoire de base peut être récupéré à l'aide de la commande suivante :</p> <pre>\$ sudo alternatives --display java</pre>
sessions-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 paramètre	Obligatoire	Valeur par défaut	Description
session-screenshots-max-height	Non	100	Spécifie la hauteur maximale, en pixels, des captures d'écran de session prises à l'aide de l'GetSessionScreenshotsAPI.
session-screenshots-format	Non	png	Format de fichier image des captures d'écran de session prises à l'aide de l'GetSessionScreenshotsAPI.
create-sessions-queue-max-size	Non	1000	Nombre maximum de demandes d>CreateSessionsAPI 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-sessions-queue-max-time-seconds	Non	1800	La durée maximale, en secondes, pendant laquelle une demande d>CreateSessionsAPI non satisfaite peut rester dans la file d'attente. Si la demande ne peut pas être satisfaite dans le délai spécifié, elle échoue.

Nom du paramètre	Obligatoire	Valeur par défaut	Description
session-manager-working-path	Oui	/tmp	Spécifie le chemin d'accès au répertoire dans lequel le Broker écrit les fichiers nécessaires au fonctionnement. Ce répertoire ne doit être accessible qu'au Broker.
enable-authentication-server	Oui	true	Spécifie si le broker est le serveur d'authentification utilisé pour générer des jetons d'accès OAuth 2.0 pour les API clientes.
enable-authentication	Oui	true	Active ou désactive l'autorisation client. Si vous activez l'autorisation client, l'API client doit fournir un jeton d'accès lors des demandes d'API. Si vous désactivez l'autorisation client, les API clientes peuvent effectuer des demandes sans jetons d'accès.

Nom du paramètre	Obligatoire	Valeur par défaut	Description
enable-agent-authorization	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 lequel les sessions supprimées deviennent invisibles et ne sont plus renvoyées par les appels DescribeSession d'API.
connect-session-to-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 clients.

Nom du paramètre	Obligatoire	Valeur par défaut	Description
client-to-broker-connect-bind-host	Non	0.0.0.0	Spécifie l'adresse IP de l'hôte où le agent 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 mot de passe du magasin de clés.

Nom du paramètre	Obligatoire	Valeur par défaut	Description
agent-to-broker-connection-https-port	Oui	8445	Spécifie le port HTTPS sur lequel le Broker écoute les connexions de l'agent.
agent-to-broker-connection-binding-host	Non	0.0.0.0	Spécifie l'adresse IP de l'hôte où le agent se lie pour les connexions d'agent.
agent-to-broker-connection-key-store-file	Oui		Spécifie le magasin de clés utilisé pour les connexions à l'agent TLS.

Nom du paramètre	Obligatoire	Valeur par défaut	Description
agent-to-broker-connectokey-store-pass	Oui		Spécifie le mot de passe du magasin de clés.
broker-to-broker-port	Oui	47100	Spécifie le port utilisé pour les connexions entre courtiers.
broker-to-broker-bind-host	Non	0.0.0.0	Spécifie l'adresse IP de l'hôte auquel le courtier se lie pour les connexions entre courtiers.
broker-to-broker-discovery-port	Oui	47500	Spécifie le port utilisé par les courtiers pour se découvrir mutuellement.

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

Nom du paramètre	Obligatoire	Valeur par défaut	Description
broker-to-broker-discovery-multicast-group	Non		Spécifie le groupe de multidiffusion pour 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-multicast-port	Non		Spécifie le port de multidiffusion pour Broker-to-broker Discovery. 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 paramètre	Obligatoire	Valeur par défaut	Description
broker-to-broker-discovery-AWS-region	Non		Spécifie laAWS région de l'équilibreur de charge de l'application utilisée pour la découverte entre courtiers. Si vous spécifiez broker-to-broker-discovery-multicast-group oubroker-to-broker-discovery-addresses omettez ce paramètre.broker-to-broker-discovery-multicast-port
broker-to-broker-discovery-AWS-alb-target-group-arn	Non		ARN du groupe cible de l'équilibreur de charge de l'application utilisé pour la découverte entre courtiers. Si vous spécifiez broker-to-broker-discovery-multicast-group oubroker-to-broker-discovery-addresses omettez ce paramètre.broker-to-broker-discovery-multicast-port

Nom du paramètre	Obligatoire	Valeur par défaut	Description
broker-to-broker-distributed-memory-max-size-mb	Non	4096	Spécifie la quantité maximale de mémoire off-heap à utiliser par un seul Broker pour stocker les données de session NICE DCV.
broker-to-broker-key-store-file	Oui		Spécifie le magasin de clés utilisé pour les connexions TLS Broker.
broker-to-broker-key-store-pass	Oui		Spécifie le mot de passe du magasin de clés.
enable-cloud-watch-metrics	Non	false	Active ou désactive les CloudWatch métriques Amazon. Si vous activez CloudWatch les mesures, vous devrez peut-être spécifier une valeur pour <code>cloud-watch-region</code> .

Nom du paramètre	Obligatoire	Valeur par défaut	Description
cloud-watch-region	Non	Obligatoire 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 Broker peut traiter chaque seconde avant d'être limitée.
enable-rottlir-forward-header	Non	false	Si ce paramètre est réglé sur true, l'adresse IP de l'appelant est extraite de l'en-tête X-Forwarded-For, le cas échéant.
create-session-number-of-retries-on-failure	Non	2	Spécifie le nombre maximum de tentatives à effectuer après l'échec d'une demande de création de session sur un hôte de serveur NICE DCV. Définissez ce paramètre sur 0 pour ne jamais réessayer en cas d'échec.

Nom du paramètre	Obligatoire	Valeur par défaut	Description
autorun-file-arguments-max-size	Non	50	Spécifie le nombre maximal d'arguments pouvant être transmis au fichier Autorun.
autorun-file-arguments-max-argument-length	Non	150	Spécifie la longueur maximale en caractères de chaque argument de fichier autorun.
enable-persistence	Oui	false	Si ce paramètre est défini sur true, les données d'état du courtier sont conservées dans une base de données externe.
persist-connection-db	Non	Obligatoire uniquement s'enable-persistence est défini sur true.	Spécifie la base de données utilisée pour la persistance. Les seules valeurs prises en charge sont : dynamodb et mysql.

Nom du paramètre	Obligatoire	Valeur par défaut	Description
dynamoc-region	Non	Obligatoire uniquement s'enable-persistence il est défini sur true et persistence-db est réglé sur dynamodb.	Spécifie la région dans laquelle les tables DynamoDB sont créées et accessibles.
dynamoc-table-rcu	Non	Obligatoire uniquement s'enable-persistence il est défini sur true et persistence-db est réglé sur dynamodb.	Spécifie les unités de capacité de lecture (RCU) pour chaque table DynamoDB. Pour plus d'informations sur les RCU, consultez la section Tarification de la capacité provisionnée .
dynamoc-table-wcu	Non	Obligatoire uniquement s'enable-persistence il est défini sur true et persistence-db est réglé sur dynamodb.	Spécifie les unités de capacité d'écriture (WCU) pour chaque table DynamoDB. Pour plus d'informations sur la WCU, consultez la section Tarification de la capacité provisionnée .

Nom du paramètre	Obligatoire	Valeur par défaut	Description
dynamodb-table-name-prefix	Non	Obligatoire uniquement s'enable-persistence est défini sur true et persistence-db est réglé sur dynamodb.	Spécifie le préfixe qui est ajouté à chaque table DynamoDB (utile pour distinguer plusieurs clusters de courtiers utilisant le même AWS compte). Seuls les caractères alphanumériques, les points, les tirets et les traits de soulignement sont autorisés.

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

Nom du paramètre	Obligatoire	Valeur par défaut	Description
jdbc-password	Non	Obligatoire uniquement s'enable-persistence est défini sur true et persistence-db est réglé sur mysql.	Spécifie le mot de passe de l'utilisateur ayant accès à la base de données MariaDB/MySQL.
seconds-before-deleting-unreachable-dcv-server	Non	1800	Spécifie le nombre de secondes après lequel un serveur inaccessible est supprimé du système.

Fichier de configuration de l'agent

Le fichier de configuration de l'agent (/etc/dcv-session-manager-agent/agent.conf pour Linux et C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf pour Windows) inclut des paramètres qui peuvent être configurés pour personnaliser la fonctionnalité 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 paramètre	Obligatoire	Valeur par défaut	Description
<code>agent.tls.ker_hostname</code>	Oui		Spécifie le nom DNS de l'hôte Broker.
<code>agent.tls.ker_port</code>	Oui	8445	Spécifie le port par lequel communiquer avec le Broker.
<code>agent.config.file</code>	Non		Nécessaire uniquement si <code>tls_strict</code> est réglé sur <code>true</code> . Spécifie le chemin d'accès au fichier de certificat (.pem) nécessaire pour valider le certificat TLS. Copiez le certificat auto-signé du agent à l'agent.
<code>agent.config.init_folder</code>	Non	<ul style="list-style-type: none"> <code>/var/lib/dcv-session-manager-agent/init</code> (Linux) 	Spécifie le chemin d'accès à un dossier sur le serveur hôte utilisé pour stocker les scripts personnalisés autorisés à initialiser les sessions du serveur NICE DCV lors de leur création. Vous devez spécifier un chemin absolu. Le dossier doit être accessible et les fichiers doivent être exécutables par les utilisateurs qui utilisent le paramètre de <code>InitFiledemande</code> de <code>CreateSessionsAPI</code> .

Nom du paramètre	Obligatoire	Valeur par défaut	Description
agent.tls_strict	Non	true	Indique si une validation TLS stricte doit être utilisée.
agent.software_statement_path	Non		Nécessaire uniquement si l'instruction logicielle par défaut n'est pas utilisée. Spécifie le chemin d'accès au fichier d'instructions du logiciel. Pour plus d'informations, veuillez consulter generate-software-statement .
agent.tls_folders	Non	<ul style="list-style-type: none"> • /etc/dcv-session-manager-agent (Linux) • C:\Program Files\NICE\DCVSessionManagerAgent\conf\tags (Windows) 	Spécifie le chemin d'accès au dossier où se trouvent les fichiers de balises. Pour plus d'informations, veuillez consulter Utilisation de balises pour cibler les serveurs NICE DCV .

Nom du paramètre	Obligatoire	Valeur par défaut	Description
agent.a orun_ er	Non	<ul style="list-style-type: none"> • /var/lib/dcv-session-manager/autorun (Linux) • C:\ProgramData\NICE\DcvSessionManagerAgent\autorun (Windows) 	Spécifie le chemin d'accès à un dossier sur le serveur hôte utilisé pour stocker les scripts et les applications dont l'exécution automatique est autorisée au démarrage de la session. Vous devez spécifier un chemin absolu. Le dossier doit être accessible et les fichiers doivent être exécutables par les utilisateurs qui utilisent le paramètre de AutorunFiledemande de l>CreateSessionsAPI.
agent.n _virtua ssion	Non	-1 (aucune limite)	Nombre maximum de sessions virtuelles pouvant être créées sur un serveur NICE DCV à l'aide du gestionnaire de sessions NICE DCV.
agent.n _concu nt_ses ns_per er	Non	1	Nombre maximum de sessions virtuelles pouvant être créées sur un serveur NICE DCV par un seul utilisateur à l'aide du gestionnaire de sessions NICE DCV.

Nom du paramètre	Obligatoire	Valeur par défaut	Description
agent.t ker_upc e_inte l	Non	30	Spécifie le nombre de secondes à attendre avant d'envoyer les données mises à jour au Broker. Les données envoyées incluent l'état du serveur et de l'hôte NICE DCV, ainsi que des informations de session mises à jour. Des valeurs plus faibles rendent le gestionnaire de session plus conscient des 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 réduisent la charge du système et du réseau, mais le gestionnaire de session réagit moins aux modifications du système. Par conséquent, des valeurs supérieures à celles120 qui ne sont pas recommandées.

Nom du paramètre	Obligatoire	Valeur par défaut	Description
log.level	Non	info	<p>Spécifie le niveau de verbosité des fichiers journaux. Les différents niveaux de détails disponibles sont les suivants :</p> <ul style="list-style-type: none"> • error: fournit le moins de détails possible. Inclut les erreurs uniquement. • warning: inclut des erreurs et des avertissements. • info—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.directory	Non	<ul style="list-style-type: none"> • /var/log/dcv-session-manager-agent/(Linux) • C:\ProgramData\nice\DCVSessionManagerAgent\log (Windows) 	Spécifie le répertoire dans lequel vous créez les fichiers journaux.

Nom du paramètre	Obligatoire	Valeur par défaut	Description
log.rotation	Non	daily	<p>Spécifie la rotation du fichier journal. Les valeurs valides sont :</p> <ul style="list-style-type: none"> hourly—Les fichiers journaux font l'objet d'une rotation horaire. daily—Les fichiers journaux font l'objet d'une rotation quotidienne.
log.maxfilesize	Non	10485760	<p>Lorsque la taille d'un fichier journal atteint la taille spécifiée en octets, il subit une rotation. Un nouveau fichier journal sera créé et d'autres événements seront placés dans le nouveau fichier.</p>
log.rotate	Non	9	<p>Nombre maximum de fichiers journaux conservés dans la rotation. Chaque fois qu'une rotation se produit et que ce nombre est atteint, le fichier journal le plus ancien est supprimé.</p>

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

Cette page fournit les notes de mise à jour et l'historique des documents pour le gestionnaire de session NICE DCV.

Rubriques

- [Notes de mise à jour du gestionnaire de sessions NICE DCV](#)
- [Historique de la documentation](#)

Notes de mise à jour du gestionnaire de sessions NICE DCV

Cette section fournit un aperçu des principales mises à jour, des versions de fonctionnalités et des corrections de bogues du gestionnaire de session NICE DCV. 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

- [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](#)

2023.1-16388 — 26 juin 2024

Numéros de version	Modifications et correctifs de bogues
<ul style="list-style-type: none">• Courtier : 417• Agent : 748• CLI : 140	<ul style="list-style-type: none">• 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
<ul style="list-style-type: none">• Courtier : 410• Agent : 732• CLI : 140	<ul style="list-style-type: none">• Correctifs de bogues et améliorations de performances

2023.0-15065— 4 mai 2023

Numéros de version	Modifications et correctifs de bogues
<ul style="list-style-type: none">• Courtier : 392• Agent : 675• CLI : 132	<ul style="list-style-type: none">• 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
<ul style="list-style-type: none">• Courtier : 392	<ul style="list-style-type: none">• Ajout du support pour Red Hat Enterprise Linux 9, Rocky Linux 9 et CentOS Stream 9.

Numéros de version	Modifications et correctifs de bogues
<ul style="list-style-type: none">• Agent : 642• CLI : 132	

2022.2-13907 — 11 novembre 2022

Numéros de version	Modifications et correctifs de bogues
<ul style="list-style-type: none">• Courtier : 382• Agent : 612• CLI : 123	<ul style="list-style-type: none">• 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
<ul style="list-style-type: none">• Courtier : 355• Agent : 592• CLI : 114	<ul style="list-style-type: none">• 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
<ul style="list-style-type: none">• Courtier : 341• Agent : 520• CLI : 112	<ul style="list-style-type: none">• 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
<ul style="list-style-type: none"> • Courtier : 307 • Agent : 453 • CLI : 92 	<ul style="list-style-type: none"> • Ajout de la prise en charge de l'intégration avec la passerelle de connexion NICE DCV. • 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
<ul style="list-style-type: none"> • Courtier : 288 • Agent : 413 • CLI : 54 	<ul style="list-style-type: none"> • Correction d'un problème lié à la validation des noms de connexion incluant un domaine Windows.

2021.2-11190 — 11 octobre 2021

Numéros de version	Modifications et correctifs de bogues
<ul style="list-style-type: none"> • Courtier : 254 • Agent : 413 • CLI : 54 	<ul style="list-style-type: none"> • 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
<ul style="list-style-type: none"> • Courtier : 254 	<ul style="list-style-type: none"> • Le gestionnaire de session NICE DCV prend désormais en charge l'interface de ligne de commande (CLI). Vous pouvez créer et gérer des 	<ul style="list-style-type: none"> • Lorsque vous enregistrez un serveur d'autorisation externe, vous pouvez désormais spécifier l'algorithme utilisé par le serveur d'autorisation

Numéros de version	Nouvelles fonctionnalités	Modifications et correctifs de bogues
<ul style="list-style-type: none"> Agent : 413 CLI : 37 	<ul style="list-style-type: none"> sessions NICE DCV dans la CLI, au lieu d'appeler des API. Le gestionnaire de session NICE DCV a introduit la persistance des données Broker. Pour une disponibilité accrue, les courtiers peuvent conserver les informations sur l'état du serveur dans un magasin de données externe et restaurer les données au démarrage. 	<ul style="list-style-type: none"> pour signer les jetons Web au format JSON. Avec cette modification, vous pouvez utiliser Azure AD comme serveur d'autorisation externe.

2021.1-10557 — 31 mai 2021

Numéros de version	Nouvelles fonctionnalités	Modifications et correctifs de bogues
<ul style="list-style-type: none"> Courtier : 214 Agent : 365 	<ul style="list-style-type: none"> Le gestionnaire de session NICE DCV 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. 	<ul style="list-style-type: none"> 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
<ul style="list-style-type: none"> Courtier : 183 Agent : 318 	<ul style="list-style-type: none"> NICE DCV Session Manager a introduit les nouvelles API suivantes : <ul style="list-style-type: none"> OpenServers CloseServers DescribeServers

Numéros de version	Modifications et correctifs de bogues
	<ul style="list-style-type: none"> • GetSessionScreenshots • Il a également introduit les nouveaux paramètres de configuration suivants : <ul style="list-style-type: none"> • Paramètres du courtier : <code>session-screenshot-max-width</code> , <code>session-screenshot-max-height</code> , <code>session-screenshot-format</code> , <code>create-sessions-queue-max-size</code> , et <code>create-sessions-queue-max-time-seconds</code> . • Paramètres de l'agent : <code>agent.autorun_folder</code> , <code>max_virtual_sessions</code> , et <code>max_concurrent_sessions_per_user</code> . <p>Paramètres de l'agent : <code>agent.autorun_folder</code> , <code>max_virtual_sessions</code> , et <code>max_concurrent_sessions_per_user</code> .</p> <p>Paramètres de l'agent : <code>agent.autorun_folder</code> , <code>max_virtual_sessions</code> , et <code>max_concurrent_sessions_per_user</code> .</p>

2020.2-9662 — 04 décembre 2020

Numéros de version	Modifications et correctifs de bogues
<ul style="list-style-type: none"> • Courtier : 114 • Agent : 211 	<ul style="list-style-type: none"> • 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
<ul style="list-style-type: none"> • Courtier : 78 • Agent : 183 	<ul style="list-style-type: none"> • La version initiale de NICE DCV Session Manager.

Historique de la documentation

Le tableau suivant décrit la documentation de cette version de NICE DCV Session Manager.

Modification	Description	Date
Version DCV NICE 2023.1-16388	Le gestionnaire de session NICE DCV a été mis à jour pour NICE DCV 2023.1-16388. Pour plus d'informations, consultez 2023.1-16388 — 26 juin 2024 .	26 juin 2024
NICE DCV Version 2023.1	Le gestionnaire de session NICE DCV a été mis à jour pour NICE DCV 2023.1. Pour plus d'informations, consultez 2023.1 — 9 novembre 2023 .	9 novembre 2023
NICE DCV Version 2023.0	Le gestionnaire de session NICE DCV a été mis à jour pour NICE DCV 2023.0. Pour plus d'informations, consultez 2023.0-14852 — 28 mars 2023 .	28 mars 2023
NICE DCV Version 2022.2	Le gestionnaire de session NICE DCV a été mis à jour pour NICE DCV 2022.2. Pour plus d'informations, consultez 2022.2-13907 — 11 novembre 2022 .	11 novembre 2022
NICE DCV Version 2022.1	Le gestionnaire de session NICE DCV a été mis à jour pour NICE DCV 2022.1. Pour plus d'informations, consultez 2022.1-13067 — 29 juin 2022 .	29 juin 2022
NICE DCV version 2022.0	Le gestionnaire de session NICE DCV a été mis à jour pour NICE DCV 2022.0. Pour plus d'informations, consultez 2022.0-11952 — 23 février 2022 .	23 février 2022

Modification	Description	Date
NICE DCV version 2021.3	Le gestionnaire de session NICE DCV a été mis à jour pour NICE DCV 2021.3. Pour plus d'informations, consultez 2021.3-11591 — 20 décembre 2021 .	20 décembre 2021
NICE DCV Version 2021.2	Le gestionnaire de session NICE DCV a été mis à jour pour NICE DCV 2021.2. Pour plus d'informations, consultez 2021.2-11042 — 1er septembre 2021 .	01 septembre 2021
NICE DCV Version 2021.1	Le gestionnaire de session NICE DCV a été mis à jour pour NICE DCV 2021.1. Pour plus d'informations, consultez 2021.1-10557 — 31 mai 2021 .	31 mai 2021
NICE DCV Version 2021.0	Le gestionnaire de session NICE DCV a été mis à jour pour NICE DCV 2021.0. Pour plus d'informations, consultez 2021.0-10242 — 12 avril 2021 .	12 avril 2021
Version initiale du gestionnaire de sessions DCV NICE	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.