POST EDIT. ADDED PROOFREAD. ADDED PP1

NICE DCV-Sitzungsmanager

NICE DCV-Sitzungsmanager: POST EDIT. ADDED PROOFREAD. ADDED PP1

Table of Contents

Was ist NICE DCV Session Manager?

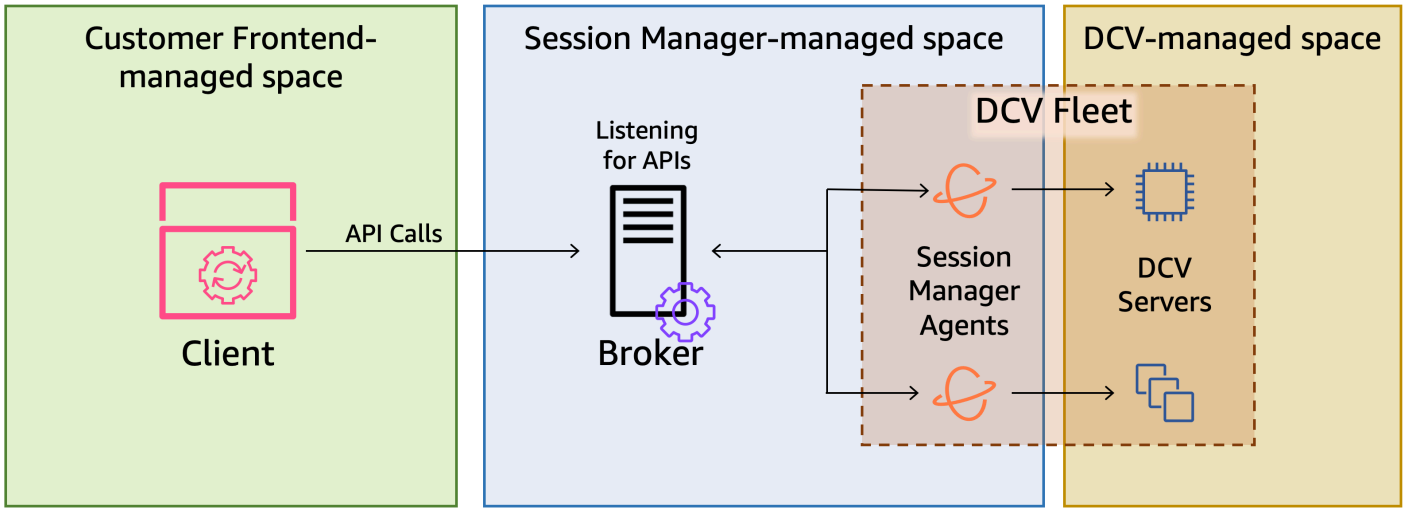
NICE DCV Session Manager besteht aus installierbaren Softwarepaketen (einem Agenten und einem Broker) und einer Anwendungsprogrammierschnittstelle (API), die es Entwicklern und unabhängigen Softwareanbietern (ISVs) leicht machen, Frontend-Anwendungen zu erstellen und zu verwalten, die den Lebenszyklus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

In diesem Handbuch wird erklärt, wie der Session Manager Agent und der Broker installiert und konfiguriert werden. Weitere Informationen zur Verwendung der Session Manager-APIs finden Sie im NICE DCV Session Manager Developer Guide.

Themen

Wie funktioniert Session Manager

Das folgende Diagramm zeigt die allgemeinen Komponenten von Session Manager.



Broker

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

Kundendienstmitarbeiter

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

APIs

Session Manager stellt eine Reihe von REST-APIs (Application Programming Interfaces) zur Verfügung, mit denen NICE-DCV-Sitzungen auf einer Flotte von NICE-DCV-Servern verwaltet werden können. Die APIs werden auf dem Broker gehostet und von diesem bereitgestellt. Entwickler können benutzerdefinierte Sitzungsverwaltungsclients erstellen, die die APIs aufrufen.

Client

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

Zugriffstoken

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

Client-API

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

NICE DCV-Sitzung

Sie müssen auf Ihrem NICE-DCV-Server eine NICE-DCV-Sitzung erstellen, mit der sich Ihre Clients verbinden können. Clients können nur dann eine Verbindung zu einem NICE-DCV-Server

herstellen, wenn eine aktive Sitzung besteht. NICE DCV unterstützt Konsolen- und virtuelle Sitzungen. Sie verwenden die Session Manager-APIs, um den Lebenszyklus von NICE-DCV-Sitzungen zu verwalten. NICE-DCV-Sitzungen können sich in einem der folgenden Zustände befinden:

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

Features

DCV Session Manager bietet die folgenden Funktionen:

- Stellt NICE-DCV-Sitzungsinformationen bereit — ruft Informationen über die Sitzungen ab, die auf mehreren NICE-DCV-Servern ausgeführt werden.
- Verwalten Sie den Lebenszyklus für mehrere NICE-DCV-Sitzungen — erstellen oder löschen Sie mehrere Sitzungen für mehrere Benutzer auf mehreren NICE-DCV-Servern mit einer API-Anfrage.
- Unterstützt Tags — Verwenden Sie benutzerdefinierte Tags, um beim Erstellen von Sitzungen eine Gruppe von NICE-DCV-Servern als Ziel zu verwenden.
- Verwaltet Berechtigungen für mehrere NICE-DCV-Sitzungen — Ändern Sie Benutzerberechtigungen für mehrere Sitzungen mit einer API-Anfrage.
- Stellt Verbindungsinformationen bereit — ruft Client-Verbindungsinformationen für NICE-DCV-Sitzungen ab.
- Unterstützt Cloud- und lokale Server: Verwenden Sie Session Manager aufAWS, vor Ort oder mit alternativen cloudbasierten Servern.

Einschränkungen

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

Preisgestaltung

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

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

Weitere Informationen finden Sie unter Licensing the NICE DCV Server im NICE DCV Administrator Guide.

Anforderungen für NICE DCV Session Manager

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

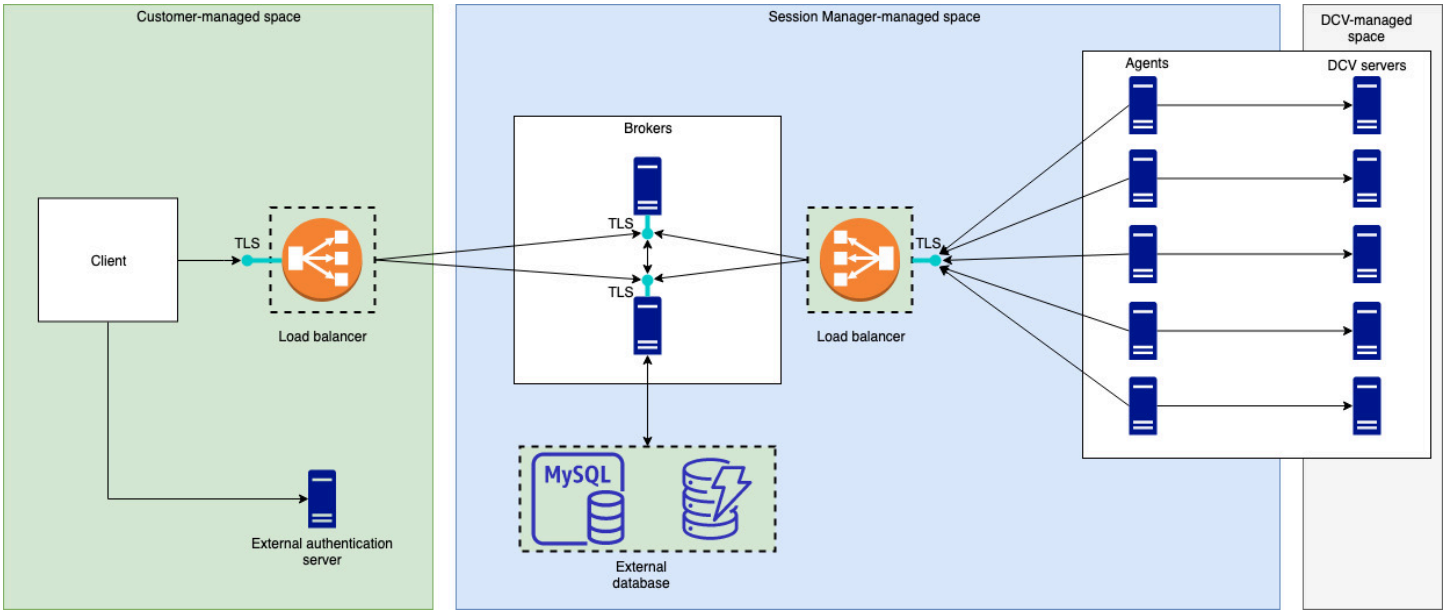
	Broker	Kundendienstmitarbeiter

Betriebssystem	• Amazon Linux 2
	• CentOS 7.6 oder höher
	• CentOS Stream 8
	• CentOS Stream 9
	• RHEL 7.6 oder höher
	• RHEL 8.x
	• RHEL 9.x
	• Rocky Linux 8.5 oder höher
	• Rocky Linux 9.x
	• Ubuntu 20.04
	• Ubuntu 22.04
		• Windows
		  • Windows Server 2022
		  • Windows Server 2019
		  • Windows Server 2016
		• Linux-Server
		  • Amazon Linux 2
		  • CentOS 7.6 oder höher
		  • CentOS Stream 8
		  • CentOS Stream 9
		  • RHEL 7.6 oder höher
		  • RHEL 8.x
		  • RHEL 9.x
		  • Rocky Linux 8.5 oder höher
		  • Rocky Linux 9.x

	Broker	Kundendienstmitarbeiter
		• Ubuntu 20.04 • Ubuntu 22.04 • SUSE Linux Enterprise 12 mit SP4 oder höher • SUSE Linux Enterprise 15
Architektur	• 64-Bit x86 • 64-Bit-ARM	• 64-Bit x86 • 64-Bit-ARM (nur Amazon Linux 2, CentOS 7.x/8.x/9.x, RHEL 7.x/8.x/9.x und Rocky 8.x/9.x) • 64-Bit-ARM (Ubuntu 22.04)
Arbeitsspeicher	8 GB	4 GB
NICE DCV DCV-Version	NICE DCV 2020.2 und später	NICE DCV 2020.2 und später
Zusätzliche Anforderungen	Java 11	-

Netzwerk- und Konnektivitätsanforderungen

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



Der Broker muss auf einem separaten Host installiert sein, aber er muss über eine Netzwerkverbindung mit den Agenten auf den NICE-DCV-Servern verfügen. Wenn Sie sich zur Verbesserung der Verfügbarkeit für mehrere Broker entscheiden, müssen Sie jeden Broker auf einem separaten Host installieren und konfigurieren und einen oder mehrere Load Balancer verwenden, um den Datenverkehr zwischen dem Client und den Brokern sowie den Brokern und den Agenten zu verwalten. Die Broker sollten auch in der Lage sein, miteinander zu kommunizieren, um Informationen über die NICE-DCV-Server und -Sitzungen auszutauschen. Die Broker können ihre Schlüssel und Statusdaten in einer externen Datenbank speichern und haben diese Informationen nach einem Neustart oder einer Beendigung zur Verfügung. Dies trägt dazu bei, das Risiko des Verlusts wichtiger Broker-Informationen zu verringern, indem sie in der externen Datenbank gespeichert werden. Sie können sie später abrufen. Wenn Sie sich dafür entscheiden, müssen Sie die externe Datenbank einrichten und die Broker konfigurieren. DynamoDB, MariaDB und MySQL werden unterstützt. Die Konfigurationsparameter sind in der Broker-Konfigurationsdatei aufgeführt.

Die Agents müssen in der Lage sein, sichere, persistente, bidirektionale HTTPS-Verbindungen mit dem Broker zu initiieren.

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

NICE DCV Session Manager einrichten

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

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

Schritte

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

Schritt 1: Bereiten Sie die NICE DCV-Server vor

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

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

Um den lokalen Dienstbenutzer für Linux NICE DCV-Server hinzuzufügen

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

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

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

Der Sitzungsmanager kann nur NICE-DCV-Sitzungen im Namen von Benutzern erstellen, die bereits auf dem NICE-DCV-Server vorhanden sind. Wenn eine Anfrage zum Erstellen einer Sitzung für einen Benutzer gestellt wird, der nicht existiert, schlägt die Anfrage fehl. Daher müssen Sie sicherstellen, dass jeder vorgesehene Endbenutzer über einen gültigen Systembenutzer auf dem NICE-DCV-Server verfügt.

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

Schritt 2: Den NICE DCV Session Manager Broker einrichten

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

Um den Broker zu installieren und zu starten

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

   - Amazon Linux 2, RHEL, CentOS und Rocky Linux

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

- Ubuntu

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

3. Laden Sie das Installationspaket herunter.

   - Amazon Linux 2, RHEL 7.x und 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 und 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. Installieren Sie das Paket .

   - Amazon Linux 2, RHEL 7.x und 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 und 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. Stellen Sie sicher, dass die Standardversion der Java-Umgebung 11 ist

```
$ java -version
```

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

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

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

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

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

Schritt 3: NICE DCV Session Manager Agent einrichten

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

Voraussetzungen

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

Linux host

> **Note**
>
> Der Session Manager Agent ist für die unter Anforderungen aufgeführten Linux-Distributionen und -Architekturen verfügbar:
> Die folgenden Anweisungen beziehen sich auf die Installation des Agenten auf 64-Bit-x86-Hosts. Um den Agenten auf 64-Bit-ARM-Hosts zu installieren, ersetzen Sie *x86_64* durch. aarch64 *Ersetzen Sie für Ubuntu amd64 durch.* arm64

Um den Agenten auf einem Linux-Host zu installieren

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

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

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

   - Ubuntu

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

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

2. Laden Sie das Installationspaket herunter.

   - Amazon Linux 2, RHEL 7.x und 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 und 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. Installieren Sie das Paket .

- Amazon Linux 2, RHEL 7.x und 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 und 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. Platzieren Sie eine Kopie des selbstsignierten Zertifikats des Brokers (das Sie im vorherigen Schritt kopiert haben) im `/etc/dcv-session-manager-agent/` Verzeichnis auf dem Agenten.
5. Öffnen Sie `/etc/dcv-session-manager-agent/agent.conf` mit Ihrem bevorzugten Texteditor und gehen Sie wie folgt vor.

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

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

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

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

     Wenn Sie die TLS-Überprüfung deaktivieren möchten, legen Sie alternativ die Einstellung `tls_strict` auf fest`false`.
6. Speichern und schließen Sie die Datei.
7. Führen Sie den folgenden Befehl aus, um den Agenten zu starten.

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

Windows host

Um den Agenten auf einem Windows-Host zu installieren

1. Laden Sie das Agent-Installationsprogramm herunter.
2. Führen Sie das Installationsprogramm aus. Klicken Sie auf der Willkommensseite auf Weiter.
3. Lesen Sie auf dem EULA-Bildschirm die Lizenzvereinbarung sorgfältig durch. Wenn Sie damit einverstanden sind, wählen Sie Ich akzeptiere die Bedingungen und dann Weiter.
4. Um mit der Installation zu beginnen, wählen Sie Installieren.
5. Platzieren Sie eine Kopie des selbstsignierten Zertifikats des Brokers (das Sie im vorherigen Schritt kopiert haben) in den `C:\Program Files\NICE\DCVSessionManagerAgent\conf\` Ordner auf dem Agenten.
6. Öffnen Sie `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` mit Ihrem bevorzugten Texteditor und gehen Sie dann wie folgt vor:

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

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

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

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

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

7. Speichern und schließen Sie die Datei.
8. Beenden Sie den Agent-Dienst und starten Sie ihn neu, damit die Änderungen wirksam werden. Führen Sie die folgenden Befehle an der Eingabeaufforderung aus.

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

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

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

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

Linux NICE DCV server

Um den lokalen Dienstbenutzer für Linux NICE DCV-Server hinzuzufügen

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

   Geben Sie für `ca-file` den Pfad zur selbstsignierten Zertifizierungsstelle des Brokers an, die Sie im vorherigen Schritt auf den Host kopiert haben.

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

   Beispiel

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

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

Windows NICE DCV server

Auf Windows NICE DCV-Servern

1. Öffnen Sie den Windows-Registrierungseditor und navigieren Sie zum Schlüssel HKEY_Users/S-1-5-18/Software/GSettings/com/NiceSoftware/DCV/Security/.
2. Öffnen Sie den Parameter ca-file. Geben Sie unter Wertdaten den Pfad zur selbstsignierten Zertifizierungsstelle des Brokers an, die Sie im vorherigen Schritt auf den Host kopiert haben.

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

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

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

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

Schritt 5: Überprüfen Sie die Installationen

Themen

- Überprüfen Sie den Agenten
- Überprüfen Sie den Broker

Überprüfen Sie den Agenten

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

Linux-Agent-Host

Der auszuführende Befehl hängt von der Version ab.

- Seit Version 2022.0

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

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

- Versionen vor 2022.0

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

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

  Beispiel

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

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

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

Windows Agent-Host

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

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

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

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

Überprüfen Sie den Broker

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

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

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

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

```
{
```

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

Konfiguration des NICE DCV Session Managers

In diesem Abschnitt wird beschrieben, wie Sie die erweiterte Konfiguration für die -Speicherverwaltung durchführen.

Themen

Sitzungsmanager skalieren

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

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

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

Schritte

Schritt 1: Erstellen eines Instance-Profils

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

So erstellen Sie ein Instance-Profil

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

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

   Fügen Sie dann die folgende Richtlinie bei:

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

```
        "elasticloadbalancing:DescribeTargetHealth"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

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

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

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

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

Um das SSL-Zertifikat vorzubereiten

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

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

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

So erstellen Sie den Load Balancer

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

   Wählen Sie im Navigationsbereich Load Balancers und dann Create Load Balancer aus. Wählen Sie als Load Balancer-Typ Application Load Balancer aus.
2. Führen Sie für Step 1: Configure Load Balancer (Schritt 1; Konfigurieren von Load Balancer) die folgenden Schritte aus:

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

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

   c. Wählen Sie für Load Balancer Protocol die Option HTTPS aus, und geben Sie für Load Balancer Port ein. 8443

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

   e. Wählen Sie Weiter aus.
3. Gehen Sie für Schritt 2: Sicherheitseinstellungen konfigurieren wie folgt vor:

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

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

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

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

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

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

   c. Wählen Sie als Zieltyp die Option Instanz aus.

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

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

f. Wählen Sie Weiter aus.

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

7. Wählen Sie Erstellen.

Schritt 4: Starten Sie die Broker

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

Um die Brokers zu starten

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

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

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

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

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

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

   d. Speichern und schließen Sie die Datei.

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

Schritt 5: Erstellen Sie den Agent Application Load Balancer

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

So erstellen Sie den Load Balancer

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

   Wählen Sie im Navigationsbereich Load Balancers und dann Create Load Balancer aus. Wählen Sie als Load Balancer-Typ Application Load Balancer aus.
2. Führen Sie für Step 1: Configure Load Balancer (Schritt 1; Konfigurieren von Load Balancer) die folgenden Schritte aus:

   a. Geben Sie unter Name einen aussagekräftigen Namen für den Load Balancer ein.
   b. Wählen Sie für Schema die Option Internet-facing aus.
   c. Wählen Sie für Load Balancer Protocol die Option HTTPS aus, und geben Sie für Load Balancer Port ein. `8445`
   d. Wählen Sie für VPC die zu verwendende VPC und dann alle Subnetze in dieser VPC aus.
   e. Wählen Sie Weiter aus.
3. Gehen Sie für Schritt 2: Sicherheitseinstellungen konfigurieren wie folgt vor:

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

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

c. Wählen Sie Weiter aus.

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

   Wählen Sie Weiter aus.

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

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

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

   c. Wählen Sie als Zieltyp die Option Instanz aus.

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

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

   f. Wählen Sie Weiter aus.

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

7. Wählen Sie Erstellen.

Schritt 6: Starten Sie die Agents

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

Um die Agents zu starten

1. Bereiten Sie den NICE-DCV-Server vor. Weitere Informationen finden Sie unter Schritt 1: Bereiten Sie die NICE DCV-Server vor.

2. Platzieren Sie eine Kopie des öffentlichen CA-Schlüssels, der in Schritt 2: Bereiten Sie das SSL-Zertifikat für den Load Balancer vor erstellt wurde. Wählen oder erstellen Sie ein Verzeichnis,

das von jedem Benutzer gelesen werden kann. Die Datei mit dem öffentlichen Schlüssel der CA muss auch für jeden Benutzer lesbar sein.

3. Installieren und konfigurieren Sie den Agenten. Weitere Informationen zur Installation und Konfiguration des Agenten finden Sie unterSchritt 3: NICE DCV Session Manager Agent einrichten.

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

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

> **Important**
>
> Gehen Sie beim Ändern der NICE DCV-Serverkonfigurationsdatei wie folgt vor:
>
> - Geben Sie für den `ca-file` Parameter denselben Pfad zur öffentlichen CA-Schlüsseldatei ein, der im vorherigen Schritt verwendet wurde
> - *`Verwenden Sie für den auth-token-verifier Parameter den DNS des Agent-Loadbalancers für broker_ip_or_dns`*

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

Verwenden von Tags als Ziel für NICE-DCV-Server

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

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

Um einem Agenten Stichwörter zuzuweisen

1. Erstellen Sie mit Ihrem bevorzugten Texteditor eine neue Datei und geben Sie ihr zum Beispiel einen aussagekräftigen Namen`agent_tags.toml`. Der Dateityp und der Dateiinhalt müssen im TOML-Dateiformat angegeben sein.`.toml`
2. Fügen Sie in der Datei jedes neue Tag-Schlüssel- und Wertepaar in einer neuen Zeile hinzu, indem Sie das`key=value` Format verwenden. Beispiel:

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

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

   Wenn das Verzeichnis mehrere Tag-Dateien enthält, wenden alle in den Dateien definierten Tags den Agenten an. Die Dateien werden in alphabetischer Reihenfolge gelesen. Wenn mehrere Dateien ein Tag mit demselben Schlüssel enthalten, wird der Wert mit dem Wert aus der zuletzt gelesenen Datei überschrieben.
4. Speichern und schließen Sie die Datei.
5. Stoppen Sie den Agenten und starten Sie ihn neu.

- Windows

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

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

- Linux

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

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

Konfiguration eines externen Autorisierungsservers

Der Autorisierungsserver ist der Server, der für die Authentifizierung und Autorisierung der Client-SDKs und -Agenten verantwortlich ist.

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

Sie können Session Manager so konfigurieren, dass Amazon Cognito anstelle des Brokers als externen Autorisierungsserver verwendet wird. Weitere Informationen zu Amazon Cognito erhalten Sie im Entwicklerhandbuch von Amazon Cognito Entwicklerhandbuch von Amazon Cognito Entwicklerhandbuch.

Um Amazon Cognito als Autorisierungsserver zu verwenden

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

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

   In diesem Beispiel benennen wir den Pool`dcv-session-manager-client-app` und erstellen ihn in`us-east-1`.

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

Beispielausgabe

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

Notieren Sie sich den`userPoolId`, den Sie im nächsten Schritt benötigen.

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

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

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

Beispielausgabe

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

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

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

3. Erstellen Sie einen Benutzerpool-Client. Verwenden Sie den create-user-pool-clientBefehl und geben Sie den`userPoolId` des Benutzerpools an, den Sie erstellt haben, einen Namen für den Client und die Region, in der er erstellt werden soll. Fügen Sie auch die`--generate-secret` Option hinzu, um festzulegen, dass Sie ein Secret für den zu erstellenden Benutzerpool-Client generieren möchten.

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

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

   Beispielausgabe

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

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

4. Erstellen Sie einen neuen OAuth2.0-Ressourcenserver für den Benutzerpool. Ein Ressourcenserver ist ein Server für zugriffsgeschützte Ressourcen. Es verarbeitet authentifizierte Anfragen nach Zugriffstoken.

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

   In diesem Beispiel verwenden wird`dcv-session-manager` als Bezeichner und den Namen und`sm_scope` als Gültigkeitsbereich den Namen und die Beschreibung.

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

   Beispielausgabe

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

5. Aktualisieren Sie den Benutzerpool-Client

   Verwenden Sie den update-user-pool-client-Befehl. Geben Sie den`userPoolId` des Benutzerpools, den`ClientId` des Benutzerpool-Clients und die Region an. Geben

Sie für an`--allowed-o-auth-flows`,`client_credentials` um anzugeben, dass der Client Zugriffstoken vom Token-Endpunkt erhalten soll, indem er eine Kombination aus einer Client-ID und einem Client-Schlüssel verwendet. Geben Sie für`--allowed-o-auth-scopes` die Ressourcenserver-ID und den Bereichsnamen wie folgt an:*`resource_server_identifier`*/*`scope_name`*. Setzen Sie den ein,`--allowed-o-auth-flows-user-pool-client` um anzugeben, dass der Client bei der Interaktion mit Cognito-Benutzerpools dem OAuth-Protokoll folgen darf.

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

Beispielausgabe

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

> **Note**
>
> Der Benutzerpool ist jetzt bereit, Zugriffstoken bereitzustellen und zu authentifizieren. In diesem Beispiel lautet die URL für den Autorisierungsserverhttps://cognito-

```
    idp.us-east-1.amazonaws.com/us-east-1_QLEXAMPLE/.well-known/
    jwks.json.
```

6. Testen Sie die Konfiguration.

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

Beispielausgabe

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

7. Registrieren Sie den externen Autorisierungsserver für die Verwendung mit dem Broker mithilfe desregister-auth-server Befehls.

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

Entwickler können jetzt den Server verwenden, um Zugriffstoken anzufordern. Geben Sie bei der Anforderung von Zugriffstoken die Client-ID, das Client-Geheimnis und die hier generierte Server-URL an. Weitere Informationen zum Anfordern von Zugriffstoken finden Sie unter Erstellen, Abrufen eines Zugriffstokens und Senden einer API-Anfrage im NICE DCV Session Manager Developer Guide.

Konfiguration der Broker-Persistenz

Session Manager-Broker unterstützen die Integration mit externen Datenbanken. Die externe Datenbank ermöglicht es Session Manager, Statusdaten und Schlüssel so zu speichern, dass sie

danach verfügbar sind. Tatsächlich sind die Brokerdaten über den Cluster verteilt, sodass sie anfällig für Datenverluste sind, wenn ein Host neu gestartet werden muss oder ein Cluster beendet wird. Wenn diese Funktion aktiviert ist, können Sie Brokerknoten hinzufügen und entfernen. Außerdem können Sie einen Cluster stoppen und neu starten, ohne Schlüssel neu generieren zu müssen oder Informationen darüber zu verlieren, welcher NICE DCV Server geöffnet oder geschlossen ist.

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

- Tasten zum Einrichten von Sitzungen zum Herstellen einer Verbindung mit Kunden
- Daten von Flugsitzungen
- NICE DCV-Serverstatus

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

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

Konfigurieren Sie den Broker so, dass er auf DynamoDB bestehen bleibt

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

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

   - Legen Sie `enable-persistence = true` fest.
   - Legen Sie `persistence-db = dynamodb` fest.
   - Für`dynamodb-region` geben Sie die &aws; -Region an, in der Sie die Tabellen mit den Brokerdaten speichern möchten. Eine Liste der unterstützten Regionen finden Sie unter DynamoDB-Dienstendpunkte.

- Für`dynamodb-table-rcu` geben Sie die Anzahl der Lesekapazitätseinheiten (RCU) an, die jede Tabelle unterstützt. Weitere Informationen zu RCU finden Sie unter Bereitgestellte Kapazität von DynamoDB.
- Für`dynamodb-table-wcu` geben Sie die Anzahl der Schreibkapazitätseinheiten (WCU) an, die jede Tabelle unterstützt. Weitere Informationen zur WCU finden Sie unter Bereitgestellte Kapazität von DynamoDB.
- Für dynamodb-table-name-prefix geben Sie das Präfix an, das jeder DynamoDB-Tabelle hinzugefügt wird (nützlich, um mehrere Brokercluster zu unterscheiden, die dasselbe Konto verwenden). Nur alphanumerische Zeichen, Punkte, Bindestriche und Unterstriche sind zulässig.

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

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

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

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

Der Broker-Host muss über die Berechtigung verfügen, die DynamoDB-APIs aufzurufen. Auf Amazon-EC2-Instances werden die Anmeldeinformationen über den Amazon-EC2-Metadaten-Service automatisch abgerufen. Wenn Sie unterschiedliche Anmeldeinformationen angeben müssen, können Sie diese mithilfe einer der unterstützten Techniken zum Abrufen von Anmeldeinformationen festlegen (z. B. Java-Systemeigenschaften oder Umgebungsvariablen). Weitere Informationen finden Sie unter Bereitstellen und Abrufen von &aws; -Anmeldeinformationen.

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

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

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

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

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

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

   ```
   sudo systemctl stop dcv-session-manager-broker
   ```
3. Stellen Sie sicher, dass alle Broker im Cluster gestoppt sind, und starten Sie sie dann alle neu. Führen Sie für jeden Broker den folgenden Befehl aus:

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

Integration mit dem NICE DCV Connection Gateway

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

Wenn Ihre Infrastruktur NICE-DCV-Server umfasst, auf die über das NICE DCV Connection Gateway zugegriffen werden kann, können Sie den Session Manager so konfigurieren, dass er das NICE DCV Connection Gateway integriert. Wenn Sie die im folgenden Abschnitt beschriebenen Schritte ausführen, fungiert der Broker als Session Resolver für das Connection Gateway. Mit anderen Worten: Der Broker wird einen zusätzlichen HTTP-Endpunkt verfügbar machen. Das Connection

Gateway führt API-Aufrufe an den Endpunkt durch, um die Informationen abzurufen, die für die Weiterleitung von NICE-DCV-Verbindungen an den vom Broker ausgewählten Host erforderlich sind.

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

Session-ManagManagManager-Manager-Manag

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

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

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

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

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

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

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

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

NICE DCV DCV-Verbindung auf der Gateway-Seite

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

  Da das NICE DCV Connection Gateway HTTP-API-Aufrufe an den Broker sendet, müssen Sie, falls der Broker ein selbstsigniertes Zertifikat verwendet, das Brokerzertifikat auf den NICE DCV Connection Gateway-Host kopieren (abgerufen im vorherigen Schritt) und denca-

`file` Parameter im`[resolver]` Abschnitt der NICE DCV Connection Gateway-Konfiguration festlegen.

Optional: Aktivieren Sie die TLS-Client-Authentifizierung

Sobald Sie den vorherigen Schritt abgeschlossen haben, können der Session Manager und das Connection Gateway über einen sicheren Kanal kommunizieren, über den das Connection Gateway die Identität der Session Manager Brokers überprüfen kann. Wenn Sie verlangen, dass auch die Session Manager Brokers die Identität des Connection Gateways überprüfen, bevor Sie den sicheren Kanal einrichten, müssen Sie die TLS-Client-Authentifizierungsfunktion aktivieren. Folgen Sie dazu den Schritten im nächsten Abschnitt.

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

Session-ManagManagManager-Manager-Manag

1. Um die TLS-Client-Authentifizierung für die Kommunikation zwischen den Session Manager Brokers und dem NICE DCV Connection Gateway zu aktivieren, folgen Sie bitte den nächsten Schritten:
2. Generieren Sie die erforderlichen Schlüssel und Zertifikate, indem Sie Folgendes ausführen: Die Ausgabe des Befehls zeigt Ihnen den Ordner, in dem die Anmeldeinformationen generiert wurden, und das Passwort, das für die Erstellung der TrustStore Datei verwendet wurde.

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

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

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

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

4. Öffnen Sie anschließenddcv-session-manager-brokersession-manager-broker /etc/etc/etc/etc/ etc/etc/etc/etc/etc/etc/etc/etc/etc/.properties mit Ihrem bevorzugten Texteditor und

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

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

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

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

NICE DCV DCV-Verbindung auf der Gateway-Seite

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

  - verwenden Sie den vollständigen Pfad der Zertifikatsdatei, die Sie im vorherigen Schritt kopiert haben, wenn Sie den`cert-file` Parameter im`[resolver]` Abschnitt festlegen
  - verwenden Sie den vollständigen Pfad der Schlüsseldatei, die Sie im vorherigen Schritt kopiert haben, wenn Sie den`cert-key-file` Parameter im`[resolver]` Abschnitt festlegen

NICE DCV Session Manager NICE DCV Server - DNS-Zuordnung

Das NICE DCV Connection Gateway benötigt die DNS-Namen der NICE-DCV-Server, um eine Verbindung zu den DCV-Serverinstanzen herzustellen. In diesem Abschnitt wird veranschaulicht, wie

Sie eine JSON-Datei definieren können, die die Zuordnung zwischen jedem DCV-Server und dem zugehörigen DNS-Namen enthält.

Dateistruktur

Das Mapping besteht aus einer Liste von JSON-Objekten mit den folgenden Feldern:

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

Wobei gilt:

`ServerIdType:`

Identifiziert, auf welchen ID-Typ sich der Wert bezieht; derzeit sind die verfügbaren Werte ipAddress agentServerId, und InstanceId:

`Ip:`

Verfügbar sowohl für Amazon EC2 als auch für lokale Infrastrukturen; kann von Systemadministratoren mit einem Befehl ifconfig (Linux) oder ipconfig (Windows) schnell abgerufen werden. Diese Information ist auch in der DescribeServers API-Antwort verfügbar.

`Id:`

Verfügbar sowohl für Amazon EC2- als auch für lokale Infrastrukturen. Der Session Manager Agent erstellt jedes Mal eine neue UUID, wenn sich der Hostname oder die IP-Adresse ändern. Diese Information ist in der DescribeServers API-Antwort verfügbar.

`Host.Aws.Ec2InstanceId:`

Es ist nur für Amazon EC2 EC2-Instances verfügbar und identifiziert eine Maschine eindeutig. Es ändert sich nicht nach einem Instance-Neustart. Kann auf dem Host abgerufen werden,

indem Sie http://169.254.169.254/latest/meta-data/instance-id kontaktieren. Diese Information ist auch in der DescribeServers API-Antwort verfügbar.

`ServerId:`

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

`DnsNames:`

Das Objekt, das die DNS-Namen enthält, die dem NICE-DCV-Server zugeordnet sind, dieses Objekt wird enthalten:

`InternalDnsNames:`

Der DNS-Name, der vom NICE DCV Connection Gateway für die Verbindung mit der Instanz verwendet wird.

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

Persistenz

Es wird dringend empfohlen, die Persistenzfunktion des Session Manager Brokers zu aktivieren, um sich vor Mapping-Verlusten zu schützen, wenn mehrere Broker oder der gesamte Cluster ausfallen. Weitere Informationen zum Aktivieren der Datenpersistenz finden Sie unter Brokerpersistenz konfigurieren

Integration mit Amazon CloudWatch

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

Amazon CloudWatch überwacht Ihre Amazon Web Services (AWS) -Ressourcen und die Anwendungen, die Sie auf ausführen,AWS in Echtzeit. Sie können Metriken verwenden CloudWatch , die Variablen sind, die Sie für Ihre Ressourcen und Anwendungen messen können. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

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

- `Number of DCV servers`— Die Anzahl der vom Broker verwalteten DCV-Server.
- `Number of ready DCV servers`— Die Anzahl der DCV-Server, die sich in dem vom Broker verwaltetenREADY Status befinden.
- `Number of DCV sessions`— Die Anzahl der vom Broker verwalteten DCV-Sitzungen.
- `Number of DCV console sessions`— Die Anzahl der vom Broker verwalteten DCV-Konsolensitzungen.
- `Number of DCV virtual sessions`— Die Anzahl der virtuellen DCV-Sitzungen, die vom Broker verwaltet werden.
- `Heap memory used`— Die Menge des Heap-Speichers, der vom Broker verwendet wird.
- `Off-heap memory used`— Die Menge des vom Broker verwendeten Off-Heap-Speichers.
- `Describe sessions request time`— Die Zeit, die zum Abschließen von DescribeSessions API-Anfragen benötigt wird.
- `Delete sessions request time`— Die Zeit, die zum Abschließen von DeleteSessions API-Anfragen benötigt wird.
- `Create sessions request time`— Die Zeit, die zum Abschließen von CreateSessions API-Anfragen benötigt wird.
- `Get session connection data request time`— Die Zeit, die zum Abschließen von GetSessionConnectionData API-Anfragen benötigt wird.
- `Update session permissions sequest time`— Die Zeit, die zum Abschließen von UpdateSessionPermissions API-Anfragen benötigt wird.

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

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

     > **Note**
     >
     > Wenn Ihr Broker auf einer Amazon-EC2-Instance ausgeführt wird, ist dieser Parameter optional. Die Region wird automatisch aus dem Instance-Metadatendienst (IMDS)

abgerufen. Wenn Sie den Broker auf einem lokalen Host ausführen, ist dieser Parameter obligatorisch.

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

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

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

Der Broker-Host muss auch die Erlaubnis haben, die`cloudwatch:PutMetricData` API aufzurufen. AWSAnmeldeinformationen können mit einer der unterstützten Techniken zum Abrufen von Anmeldeinformationen abgerufen werden. Weitere Informationen finden Sie unter Angeben und Abrufen vonAWS Anmeldeinformationen.

Den NICE DCV Session Manager aktualisieren

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

> **Note**
>
> Es wird dringend empfohlen, alle Session Manager Agents zu aktualisieren, bevor Sie die Session Manager Brokers aktualisieren, um Inkompatibilitätsprobleme zu vermeiden, falls neue Funktionen eingeführt werden.

Themen

Den NICE DCV Session Manager Agent aktualisieren

Linux host

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

Um den Agenten auf einem Linux-Host zu aktualisieren

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

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

2. Laden Sie das Installationspaket herunter.

   - Amazon Linux 2, RHEL 7.x und 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 und 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. Installieren Sie das Paket .

- Amazon Linux 2, RHEL 7.x und 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 und 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. Führen Sie den folgenden Befehl aus, um den Agenten zu starten.

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

Windows host

Um den Agenten auf einem Windows-Host zu aktualisieren

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

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

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

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

Den NICE DCV Session Manager Broker aktualisieren

Um den Broker zu aktualisieren

1. Connect zu dem Host her, auf dem Sie den Broker aktualisieren möchten.

2. Beenden Sie den Broker-Dienst.

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

3. Laden Sie das Installationspaket herunter.

   - Amazon Linux 2, RHEL 7.x und 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 und 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. Installieren Sie das Paket .

   - Amazon Linux 2, RHEL 7.x und 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 und 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. Starten Sie den Broker-Service und stellen Sie sicher, dass er bei jedem Start der Instanz automatisch gestartet wird.

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

CLI Referenz für Broker

In diesem Abschnitt wird die Verwendung der Befehle der Broker-Befehlszeilenschnittstelle (CLI) beschrieben.

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

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

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

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

Verwenden Sie die folgenden Befehle zur Verwaltung des Session Manager Agents.

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

Verwenden Sie die folgenden Befehle, um die Zuordnungsdatei DCV-Server — DNS-Namen zu verwalten.

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

register-auth-server

Registriert einen externen Authentifizierungsserver für die Verwendung mit dem Broker.

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

Wenn Sie sich jedoch für die Verwendung eines externen Authentifizierungsservers wie Active Directory oder Amazon Cognito entscheiden, müssen Sie diesen Befehl verwenden, um den externen Authentifizierungsserver zu registrieren.

Themen

Syntax

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

Optionen

`--url`

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

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel

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

Befehl

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

Ausgabe

```
Jwk url registered.
```

list-auth-servers

Listet die externen Authentifizierungsserver auf, die registriert wurden.

Themen

Syntax

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

Ausgabe

`Urls`

Die URLs der registrierten externen Authentifizierungsserver.

Beispiel

Im folgenden Beispiel werden alle externen Authentifizierungsserver aufgelistet, die registriert wurden.

Befehl

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

Ausgabe

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

unregister-auth-server

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

Themen

Syntax

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

Optionen

`--url`

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

Typ: Zeichenfolge

Erforderlich: Ja

Ausgabe

`Url`

Die URL des nicht registrierten externen Authentifizierungsservers.

Beispiel

Im folgenden Beispiel wird ein externer Authentifizierungsserver mit einer URL von registrierthttps://my-auth-server.com/.

Befehl

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

Ausgabe

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

register-api-client

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

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

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

Themen

Syntax

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

Optionen

`--name`

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

Typ: Zeichenfolge

Erforderlich: Ja

Ausgabe

`client-id`

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

`client-password`

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

Beispiel

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

Befehl

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

Ausgabe

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

describe-api-clients

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

Themen

Syntax

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

Ausgabe

`name`

Der eindeutige Name des Session Manager-Clients.

`id`

Die eindeutige ID des Session Manager-Clients.

`active`

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

Beispiel

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

Befehl

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

Ausgabe

```
Api clients
```

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

unregister-api-client

Deaktiviert einen registrierten Session Manager-Client. Ein deaktivierter Session Manager-Client kann seine Anmeldeinformationen nicht mehr verwenden, um OAuth 2.0-Zugriffstoken abzurufen.

Themen

Syntax

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

Optionen

`--client -id`

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

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel

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

Befehl

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

Ausgabe

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

renew-auth-server-api-Schlüssel

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

Themen

Syntax

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

Beispiel

Im folgenden Beispiel werden die öffentlichen und privaten Schlüssel erneuert.

Befehl

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

Ausgabe

```
Keys renewed.
```

generate-software-statement

Generiert eine Softwareanweisung.

Agenten müssen beim Broker registriert sein, um die Kommunikation zu ermöglichen. Agenten benötigen eine Softwareerklärung, um sich beim Broker registrieren zu können. Nachdem der Agent eine Softwareanweisung erhalten hat, kann er sich mithilfe des OAuth 2.0 Dynamic Client Registration Protocol automatisch beim Broker registrieren. Nachdem sich der Agent beim Broker registriert hat, erhält er eine Client-ID und ein Client-Geheimnis, mit dem er sich beim Broker authentifiziert.

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

Themen

Syntax

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

Ausgabe

`software-statement`

Die Softwareerklärung.

Beispiel

Das folgende Beispiel generiert eine Softwareanweisung.

Befehl

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

Ausgabe

```
software-statement:
 ewogICJpZCIgOiAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiYWN0aXZlIiA6IHRydWUsCi
```

describe-software-statements

Beschreibt die vorhandenen Softwareanweisungen.

Themen

Syntax

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

Ausgabe

`software-statement`

Die Softwareerklärung.

`issued-at`

Datum und Uhrzeit der Generierung der Software.

`is-active`

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

Beispiel

Das folgende Beispiel generiert eine Softwareanweisung.

Befehl

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

Ausgabe

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

deactivate-software-statement

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

Themen

Syntax

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

Optionen

`--software-statement`

Die zu deaktivierende Softwareanweisung.

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel

Das folgende Beispiel deaktiviert eine Softwareanweisung.

Befehl

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

Ausgabe

```
Software statement
 EXAMPLEpZCIgOiAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj
 deactivated
```

describe-agent-clients

Beschreibt die Agenten, die beim Broker registriert sind.

Themen

Syntax

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

Ausgabe

`name`

Der Name des Agenten.

`id`

Die eindeutige ID des Agenten.

`active`

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

Beispiel

Das folgende Beispiel beschreibt die Agents.

Befehl

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

Ausgabe

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

unregister-agent-client

Heben Sie die Registrierung eines Agenten beim Broker ab.

Themen

Syntax

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

Optionen

`--client-id`

Die ID des Agenten, für den die Registrierung aufgehoben werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel

Im folgenden Beispiel wird die Registrierung eines Agenten aufgehoben.

Befehl

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

Ausgabe

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

register-server-dns-mappings

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

Syntax

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

Optionen

`--file-path`

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

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel

Das folgende Beispiel registriert die DCV-Server — DNS-Namenszuordnungen aus der Datei /tmp/mappings.json.

Befehl

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

Ausgabe

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

describe-server-dns-mappings

Beschreiben Sie die derzeit verfügbaren DCV-Server — DNS-Namenszuordnungen.

Syntax

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

Ausgabe

`serverIdType`

Der Typ der Server-ID.

`serverId`

Die eindeutige ID des Servers.

`dnsNames`

Die internen und externen DNS-Namen

`internalDnsNames`

Die internen DNS-Namen

`externalDnsNames`

Die externen DNS-Namen

Beispiel

Im folgenden Beispiel werden die Zuweisungen der registrierten DCV-Server mit DNS-Namen aufgelistet.

Befehl

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

Ausgabe

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

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

Referenz der Konfigurationsdatei

Dieser Abschnitt enthält Informationen der Agent- und Broker-Konfigurationsdateien.

Themen

Broker-Konfigurationsdatei

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

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

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

Parameter name	Erforderlich	Standardwert	Beschreibung
`broker-java-home`	Nein		Gibt den Pfad zum Java-Home-Verzeichnis an, das der Broker anstelle des Standardverzeichnisses des Systems verwendet. Wenn diese Option gesetzt ist, wird sie vom

Parameter name	Erforderlich	Standardwert	Beschreibung
			Broker<broker-java-home>/bin/java beim Start verwendet. Tipp: Der Broker benötigt Java Runtime Environment 11 und es wird installiert, falls es nach erfolgreicher Installation als Abhängigkeit fehlt. Wenn Version 11 nicht als Standard-Java-Umgebung festgelegt ist, kann das Basisverzeichnis mit dem folgenden Befehl abgerufen werden: `$ sudo alternatives --display java`
sessions-creenshot-max-width	Nein	160	Gibt die maximale Breite von Sitzungs-Screenshots in Pixeln an, die mit der GetSessionScreenshotsAPI aufgenommen wurden.
sessions-creenshot-max-height	Nein	100	Gibt die maximale Höhe von Sitzungs-Screenshots in Pixeln an, die mit der GetSessionScreenshotsAPI aufgenommen wurden.

Paramet name	Erforderlich	Standardwert	Beschreibung
sessior s creensh - format	Nein	png	Das Bilddateiformat von Sitzungs-Screenshots, die mit der GetSessio nScreenshotsAPI aufgenommen wurden.
create- se ssions- qu eue- max-s ize	Nein	1000	Die maximale Anzahl unerfüllter CreateSes sionsAPI-Anfragen, die in die Warteschlange gestellt werden können. Wenn die Warteschlange voll ist, werden neue unerfüllte Anfragen abgelehnt.
create- se ssions- qu eue- max-t ime- secon ds	Nein	1800	Die maximale Zeit in Sekunden, die eine unerfüllte CreateSes sionsAPI-Anforderung in der Warteschlange verbleiben kann. Wenn die Anfrage nicht innerhalb der angegebenen Zeit erfüllt werden kann, schlägt sie fehl.
sessior m anager- wo rking- path	Ja	/tmp	Gibt den Pfad zu dem Verzeichnis an, in das der Broker die für den Betrieb erforderlichen Dateien schreibt. Dieses Verzeichn is darf nur für den Broker zugänglich sein.

Paramet name	Erforderlich	Standardwert	Beschreibung
enable-authorization-server	Ja	true	Gibt an, ob der Broker der Authentifizierungsserver ist, der zum Generieren von OAuth 2.0-Zugriffstoken für Client-APIs verwendet wird.
enable-authorization	Ja	true	Aktiviert oder deaktiviert die Client-Autorisierung. Wenn Sie die Client-Autorisierung aktivieren, muss die Client-API bei API-Anfragen ein Zugriffstoken bereitstellen. Wenn Sie die Client-Autorisierung deaktivieren, können Client-APIs Anfragen ohne Zugriffstoken stellen.
enable-agent-authorizatic	Ja	true	Aktiviert oder deaktiviert die Agentenautorisierung. Wenn Sie die Agentenautorisierung aktivieren, muss der Agent bei der Kommunikation mit dem Broker ein Zugriffstoken bereitstellen.

Parameter name	Erforderlich	Standardwert	Beschreibung
delete-session-duration-hours	Nein	1	Gibt die Anzahl der Stunden an, nach denen gelöschte Sitzungen unsichtbar werden und nicht mehr durch`DescribeSession` API-Aufrufe zurückgegeben werden.
connect-session-token-duration-minutes	Nein	60	Gibt die Anzahl der Minuten an, für die das ConnectSession Token gültig bleibt.
client-to-broker-connector-https-port	Ja	8443	Gibt den HTTPS-Port an, an dem der Broker auf Client-Verbindungen wartet.
client-to-broker-connector-bind-host	Nein	0.0.0.0	Gibt die IP-Adresse des Hosts an, den der Broker die Verbindung der Client-Verbindungen herstellt.

Paramet name	Erforderlich	Standardwert	Beschreibung
client-to-broker-connect key-store-file	Ja		Gibt den Schlüsselspeicher an, der für TLS-Client-Verbindungen verwendet wird.
client-to-broker-connect key-store-pass	Ja		Gibt den Schlüssel speicherpass an.
agent-to-broker-connector https-port	Ja	8445	Gibt den HTTPS-Port an, an dem der Broker auf Agentenverbindungen wartet.

Parameter name	Erforderlich	Standardwert	Beschreibung
agent-to-broker-connector-bind-host	Nein	0.0.0.0	Gibt die IP-Adresse des Hosts an, den der Broker die Verbindung der Agentenverbindung herstellt.
agent-to-broker-connector-key-store-file	Ja		Gibt den Schlüsselspeicher an, der für TLS-Agent-Verbindungen verwendet wird.
agent-to-broker-connector-key-store-pass	Ja		Gibt den Schlüssel speicherpass an.
broker-to-broker-port	Ja	47100	Gibt den Port an, der für Broker-to-Broker-V erbindungen verwendet wird.

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

Paramet name	Erforderlich	Standardwert	Beschreibung
broker- to- broker- d iscove address	Nein		Gibt die IP-Adressen und Ports der anderen Broker in der Flotte im Format *ip_address* : *port* an. Wenn es mehrere Broker gibt, trennen Sie die Werte durch ein Komma. Wenn Siebroker-to-broker-discovery-multicast-group,broker-to-broker-discovery-multicast-port , oder angebenbroker-to-broker-discovery-AWS-region ,broker-to-broker-discovery-AWS-alb-target-group-arn lassen Sie diesen Parameter weg.

Paramet name	Erforderlich	Standardwert	Beschreibung
broker- to- broker- d iscove multic - group	Nein		Gibt die Multicast-Gruppe für Broker-to -roker Discovery an. Wenn Siebroker-to-broker-discovery-addresses , oder angebenbroker-to-broker-discovery-aws-region ,broker-to-broker-d iscovery-AWS-alb-target-group-arn lassen Sie diesen Parameter weg.
broker- to- broker- d iscove multic - port	Nein		Gibt den Multicast-Port für Broker-o-oker Discovery an. Wenn Siebroker-to -broker-discovery-addresses , oder angebenbroker-to-broker-discovery-AWS-region ,broker-to-broker-d iscovery-AWS-alb-target-group-arn lassen Sie diesen Parameter weg.

Paramet name	Erforderlich	Standardwert	Beschreibung
broker-to-broker-discovery-AWS-region	Nein		Gibt dieAWS Region des Anwendungs-Load Balancers an, der für die Broker-to-Broker-Discovery verwendet wird. Wenn Siebroker-to-broker-discovery-multicast-group , oder angebenbroker-to-broker-discovery-multicast-port ,broker-to-broker-discovery-addresses lassen Sie diesen Parameter weg.
broker-to-broker-discovery-AWS-alb-target-group-arn	Nein		Der ARN des Application Load Balancer-Zielgruppenbenutzers für die Broker-to-Broker-Discovery. Wenn Siebroker-to-broker-discovery-multicast-group , oder angebenbroker-to-broker-discovery-multicast-port ,broker-to-broker-discovery-addresses lassen Sie diesen Parameter weg.

Parameter name	Erforderlich	Standardwert	Beschreibung
broker-to-broker-distributed-memory-max-size-mb	Nein	4096	Gibt die maximale Menge an Off-Heap-Speicher an, die von einem einzelnen Broker zum Speichern von NICE-DCV-Sitzungsdaten verwendet werden soll.
broker-to-broker-key-store-file	Ja		Gibt den Schlüsselspeicher an, der für TLS-Broker-Verbindungen verwendet wird.
broker-to-broker-key-store-pass	Ja		Gibt den Schlüssel speicherpass an.
enable-cloud-watch-metrics	Nein	false	Aktiviert oder deaktiviert CloudWatch Amazon-Metriken. Wenn Sie CloudWatch Metrics aktivieren, müssen Sie möglicherweise einen Wert für angeben`cloud-watch-region` .

Paramet name	Erforderlich	Standardwert	Beschreibung
`cloud-watch-region`	Nein	Nur erforderlich, wenn auf gesetzt`enable-cloud-watch-metrics` ist`true`. Wenn der Broker auf einer Amazon EC2 EC2-Instance installiert ist, wird die Region aus dem IMDS abgerufen.	DieAWS Region, in der die CloudWatch Kennzahlen veröffentlicht werden.
max-api-requests-per-Sekunde	Nein	`1000`	Gibt die maximale Anzahl von Anfragen an, die die Broker-API pro Sekunde verarbeiten kann, bevor sie gedrosselt werden.
`enable-throttling-forwarded-for-header`	Nein	`false`	Wenn diese Option auf`true` Drosselung gesetzt ist, wird die Anrufer-IP aus dem X-Forwared-For-Header abgerufen, falls vorhanden.

Paramet name	Erforderlich	Standardwert	Beschreibung
create-sessions-number-of-retries-on-failure	Nein	2	Gibt die maximale Anzahl von Wiederholungsversuchen an, die ausgeführt werden sollen, nachdem eine Sitzungsanforderung zum Erstellen einer Sitzung auf einem NICE-DCV-Serverhost fehlgeschlagen ist. Setzen Sie den Wert auf 0, um bei Fehlern keine Wiederholungsversuche durchzuführen.
autorunfile-arguments-max-size	Nein	50	Gibt die maximale Anzahl von Argumenten an, die an die Autorun-Datei übergeben werden können.
autorunfile-arguments-max-argumer length	Nein	150	Gibt die maximale Länge jedes Autorun-Dateiargumente in Zeichen an.
enable-persister	Ja	false	Wenn auf gesetzttrue, werden die Broker-Statusdaten in einer externen Datenbank gespeichert.

Paramet name	Erforderlich	Standardwert	Beschreibung
`persist ce-db`	Nein	Nur erforderlich, wenn auf gesetzt`enable-pe rsistence` ist`true`.	Gibt an, welche Datenbank für die Persistenz verwendet wird. Die einzigen unterstützten Werte sind:`dynamodb` und`mysql`.
`dynamoc region`	Nein	Nur erforderlich, wenn auf gesetzt`enable-pe rsistence persisten ce-db` ist`true` und auf gesetzt ist`dynamodb`.	Gibt die Region an, in der die DynamoDB-Tabellen erstellt und auf die zugegriffen wird.
`dynamoc table- rcu`	Nein	Nur erforderlich, wenn auf gesetzt`enable-pe rsistence persisten ce-db` ist`true` und auf gesetzt ist`dynamodb`.	Gibt die Lesekapaz itätseinheiten (RCU) für jede DynamoDB-Tabelle an. Weitere Informationen zur RCU finden Sie unter Preise für bereitgestellte Kapazität.
`dynamoc table- wcu`	Nein	Nur erforderlich, wenn auf gesetzt`enable-pe rsistence persisten ce-db` ist`true` und auf gesetzt ist`dynamodb`.	Gibt die Schreibkapazitätse inheiten (WCU) für jede DynamoDB-Tabelle an. Weitere Informationen zur WCU finden Sie unter Preise für bereitgestellte Kapazität.

Paramet name	Erforderlich	Standardwert	Beschreibung
dynamoc table-nam e-prefix	Nein	Nur erforderlich, wenn auf gesetztenable-pe rsistence persisten ce-db isttrue und auf gesetzt istdynamodb.	Gibt das Präfix an, das jeder DynamoDB-Tabelle hinzugefügt wird (nützlich, um mehrere Brokercluster zu untersche iden, die dasselbeAWS Konto verwenden). Nur alphanumerische Zeichen, Punkte, Bindestriche und Unterstriche sind zulässig.
jdbc-conn ection-url	Nein	Nur erforderlich, wenn auf gesetztenable-pe rsistence persisten ce-db isttrue und auf gesetzt istmysql.	Gibt die Verbindungs-URL zur MariaDB/MySQL-Datenbank an; sie enthält den Endpunkt und den Datenbanknamen. Die URL sollte dieses Format haben: `jdbc:mysql://<db_endpoint>:<db_port>/<db_name>?createDatabaseIfNotExist=true` Wo<db_endpoint> ist der MariaDB/MySQL-Datenbankendpunkt,<db_port> ist der Datenbankport und<db_name> ist der Datenbankname.

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

Agent-Konfigurationsdatei

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

In der folgenden Tabelle sind die Parameter in der Agent-Konfigurationsdatei aufgelistet.

Paramet name	Erforderlich	Standardwert	Beschreibung
agent.b ker_hos	Ja		Gibt den DNS-Namen des Broker-Hosts an.
agent.b ker_por	Ja	8445	Gibt den Port an, über den mit dem Broker kommunizi ert werden soll.
agent.c file	Nein		Wird nur benötigt, wenn auf gesetzt`tls_stric t` ist`true`. Gibt den Pfad zur Zertifikatsdatei (.pem) an, die zur Validierung des TLS-Zertifikats benötigt wird. Kopieren Sie das selbstsignierte Zertifikat vom Broker zum Agenten.
agent.i t_folde	Nein	• `/var/lib/dcv-session-manager-agent/init` (Linux)	Gibt den Pfad zu einem Ordner auf dem Hostserve r an, in dem benutzerd efinierte Skripts gespeiche rt werden, die zur Initialis ierung von NICE-DCV-Serversitzungen zugelasse n sind, wenn sie erstellt werden. Sie müssen einen absoluten Pfad angeben. Der Ordner muss zugänglich sein und die Dateien müssen für Benutzer ausführbar sein, die den InitFileAnforderu ngsparameter der

Paramet name	Erforderlich	Standardwert	Beschreibung
			CreateSessionsAPI verwenden.
`agent._strict`	Nein	`true`	Gibt an, ob eine strikte TLS-Validierung verwendet werden sollte.
`agent.s tware_s tement_ th`	Nein		Nur erforderlich, wenn die Standard-Softwarea nweisung nicht verwendet wird. Gibt den Pfad zur Software-Anweisungsdatei an. Weitere Informationen finden Sie unter generate-software-statement.
`agent.t s_folde`	Nein	• `/etc/dcv-session-manager-ag ent` (Linux) • `C:\Program Files \NICE\DCVSess ionManagerAgent \conf\tags` (Windows)	Gibt den Pfad des Ordners an, in dem sich die Tag-Dateien befinden. Weitere Informationen finden Sie unter Verwenden von Tags als Ziel für NICE-DCV-Server.

Paramet name	Erforderlich	Standardwert	Beschreibung
agent.a orun_fo er	Nein	• /var/lib/dcv-session-manager-agent/autorun (Linux) • C:\ProgramData\NICE\DcvSessionManagerAgent\autorun (Windows)	Gibt den Pfad zu einem Ordner auf dem Hostserver an, der zum Speichern von Skripten und Apps verwendet wird, die beim Sitzungsstart automatisch ausgeführt werden dürfen. Sie müssen einen absoluten Pfad angeben. Der Ordner muss zugänglich sein und die Dateien müssen für Benutzer ausführbar sein, die den AutorunFileAnforderungsparameter der CreateSessionsAPI verwenden.
agent.m _virtua sessior	Nein	-1 (kein Limit)	Die maximale Anzahl virtueller Sitzungen, die mit dem NICE DCV Session Manager auf einem NICE DCV-Server erstellt werden können.
agent.m _concur nt_sess ns_per_ er	Nein	1	Die maximale Anzahl virtueller Sitzungen, die von einem einzelnen Benutzer mit dem NICE DCV Session Manager auf einem NICE DCV-Server erstellt werden können.

Paramet name	Erforderlich	Standardwert	Beschreibung
agent.b ker_upd e_inter l	Nein	30	Gibt die Wartezeit in Sekunden an, bevor aktualisierte Daten an den Broker gesendet werden. Zu den gesendete n Daten gehören der NICE DCV-Server- und Hoststatus sowie aktualisi erte Sitzungsinformatio nen. Niedrigere Werte sorgen dafür, dass sich der Session Manager Änderungen auf dem System, auf dem der Agent ausgeführt wird, besser bewusst, erhöhen jedoch die Systemlast und den Netzwerkverkehr. Höhere Werte verringern die System- und Netzwerkl ast, der Session Manager reagiert jedoch weniger schnell auf Systemänd erungen, weshalb höhere Werte als nicht empfohlen `120` werden.

Paramet name	Erforderlich	Standardwert	Beschreibung
log.lev	Nein	info	Gibt den Ausführli chkeitsgrad der Protokoll dateien an. Die folgenden Ausführlichkeitsstufen sind verfügbar: • error— Stellt die wenigsten Details bereit. Umfasst nur Fehler. • warning— Schließt Fehler und Warnungen ein. • info— Die standardm äßige Ausführlichkeitsst ufe. Umfasst Fehler, Warnungen und Informationsmeldungen. • debug— Bietet die meisten Details. Bietet detaillierte Informati onen, die nützlich für das Debugging sind.
log.dir tory	Nein	• /var/log/dcv-session-manager-agent/(Linux) • C:\ProgramData \NICE\DCVSess ionManagerAgent \log (Windows)	Gibt das Verzeichnis an, in dem Protokolldateien erstellt werden

Parameter name	Erforderlich	Standardwert	Beschreibung
`log.rotation`	Nein	`daily`	Gibt die Rotation der Protokolldatei an. Folgende Werte sind zulässig: • `hourly`— Protokolldateien werden stündlich rotiert. • `daily`— Protokolldateien werden täglich rotiert.
`log.maxfile-size`	Nein	`10485760`	Wenn die Größe einer Protokolldatei die angegebene Größe in Byte erreicht, wird sie rotiert. Eine neue Protokolldatei wird erstellt und weitere Protokollereignisse werden in die neue Datei aufgenommen.
`log.rotate`	Nein	`9`	Die maximale Anzahl der Protokolldateien, die bei der Rotation beibehalten wurden. Jedes Mal, wenn eine Rotation stattfindet und diese Zahl erreicht wird, wird die älteste Protokolldatei gelöscht.

Versionshinweise und Dokumentenverlauf für NICE DCV Session Manager

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

Themen

Versionshinweise zu NICE DCV Session Manager

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

Themen

2023.1-16388 — 26. Juni 2024

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

2023.1 — 9. November 2023

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

2023.0-15065 — 4. Mai 2023

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

2023.0-14852 — 28. März 2023

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

2022.2-13907 — 11. November 2022

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

2022.1-13067 — 29. Juni 2022

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

2022.0-11952 — 23. Februar 2022

Build-Nummern	Änderungen und Fehlerbehebungen
• Makler: 341 • Makler: 520 • CLI: 112	• Dem Agenten wurde die Funktion zur Rotation von Protokollen hinzugefügt. • Konfigurationsparameter hinzugefügt, um Java Home im Broker festzulegen. • Die Übertragung von Daten vom Cache auf die Festplatte im Broker wurde verbessert. • Die URL-Validierung in der CLI wurde behoben.

2021.3-11591 — 20. Dezember 2021

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

2021.2-11445 — 18. November 2021

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

2021.2-11190 — 11. Oktober 2021

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

2021.2-11042 — 01. September 2021

Build-Nummern	Neue Features	Änderungen und Fehlerbehebungen
• Makler: 254	• NICE DCV Session Manager bietet jetzt Unterstützung für die Befehlszeilenschnittstelle (CLI). Sie können NICE-DCV-Sitzungen in der CLI	• Bei der Registrierung eines externen Autorisierungsservers können Sie jetzt den Algorithmus angeben, den der Autorisierungsserver zum

Build-Nummern	Neue Features	Änderungen und Fehlerbehebungen
• Makler: 413 • CLI: 37	erstellen und verwalten, anstatt APIs aufzurufen. • NICE DCV Session Manager führte Broker-Datenpersistenz ein. Für eine höhere Verfügbarkeit können Broker Serverstatusinformationen in einem externen Datenspeicher speichern und die Daten beim Start wiederherstellen.	Signieren von Web-Token im JSON-Format verwendet. Mit dieser Änderung können Sie Azure AD als externen Autorisierungsserver verwenden.

2021.1-10557 — 31. Mai 2021

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

2021.0-10242 — 12. April 2021

Build-Nummern	Änderungen und Fehlerbehebungen

• Makler: 183
• Makler: 318
	• NICE DCV Session Manager führte die folgenden neuen APIs ein:
	  • OpenServers
	  • CloseServers
	  • DescribeServers

Build-Nummern	Änderungen und Fehlerbehebungen

	• GetSessionScreenshots
	• Außerdem wurden die folgenden neuen Konfigurationsparameter eingeführt:
	  • Broker-Parameter: `session-screenshot-max-width session-screenshot-max-height ,session-screenshot-format ,create-sessions-queue-max-size , undcreate-sessions-queue-max-time-seconds .`
	  • Agentenparameter: `agent.autorun_folder max_virtual_sessions , undmax_concurrent_sessions_per_user .`
	
	  Agentenparameter: `agent.autorun_folder max_virtual_sessions , undmax_concurrent_sessions_per_user .`
	
	  Agentenparameter: `agent.autorun_folder max_virtual_sessions , undmax_concurrent_sessions_per_user .`

2020.2-9662 — 04. Dezember 2020

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

2020.2-9508 — 11. November 2020

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

Dokumentverlauf

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

Änderung	Beschreibung	Datum
NICE DCV DCV-Ausführung 2023.1-16388	NICE DCV Session Manager wurde für NICE DCV 2023.1-16388 aktualisiert. Weitere Informationen finden Sie unter 2023.1-16388 — 26. Juni 2024.	26. Juni 2024
NICE DCV Version 2023.1	NICE DCV Session Manager wurde für NICE DCV 2023.1 aktualisiert. Weitere Informationen finden Sie unter 2023.1 — 9. November 2023.	9. November 2023
NICE DCV Version 2023.0	NICE DCV Session Manager wurde für NICE DCV 2023.0 aktualisiert. Weitere Informationen finden Sie unter 2023.0-14852 — 28. März 2023.	28. März 2023
NICE DCV DCV-Version 2022.2	NICE DCV Session Manager wurde für NICE DCV 2022.2 aktualisiert. Weitere Informationen finden Sie unter 2022.2-13907 — 11. November 2022.	11. November 2022
NICE DCV Version 2022.1	NICE DCV Session Manager wurde für NICE DCV 2022.1 aktualisiert. Weitere Informationen finden Sie unter 2022.1-13067 — 29. Juni 2022.	29. Juni 2022
NICE DCV Version 2022.0	NICE DCV Session Manager wurde für NICE DCV 2022.0 aktualisiert. Weitere Informationen finden Sie unter 2022.0-11952 — 23. Februar 2022.	23. Februar 2022

Änderung	Beschreibung	Datum
NICE DCV Version 2021.3	NICE DCV Session Manager wurde für NICE DCV 2021.3 aktualisiert. Weitere Informationen finden Sie unter 2021.3-11591 — 20. Dezember 2021.	20. Dezember 2021
NICE DCV Version 2021.2	NICE DCV Session Manager wurde für NICE DCV 2021.2 aktualisiert. Weitere Informationen finden Sie unter 2021.2-11042 — 01. September 2021.	01. September 2021
NICE DCV Version 2021.1	NICE DCV Session Manager wurde für NICE DCV 2021.1 aktualisiert. Weitere Informationen finden Sie unter 2021.1-10557 — 31. Mai 2021.	31. Mai 2021
NICE DCV Version 2021.0	NICE DCV Session Manager wurde für NICE DCV 2021.0 aktualisiert. Weitere Informationen finden Sie unter 2021.0-10242 — 12. April 2021.	12. April 2021
Erste Version von NICE DCV Session Manager	Die erste Veröffentlichung dieses Inhalts.	11. November 2020

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