



Guía del administrador

NICE DCV Session Manager



NICE DCV Session Manager: Guía del administrador

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es Session Manager?	1
Cómo funciona Session Manager	1
Características	3
Limitaciones	3
Precios	4
Requisitos	4
Requisitos de red y conectividad	5
Configuración	7
Paso 1: Preparar los servidores NICE DCV	7
Paso 2: Configurar el broker	8
Paso 3: Configurar el agente	10
Paso 4: Configurar el servidor NICE DCV	15
Paso 5: Verificar las instalaciones	16
Verificar el agente	17
Comprobar el broker	18
Configuración	19
Session Manager de escalado	19
Paso 1: Crear un perfil de instancia	20
Paso 2: Preparar el certificado SSL para el equilibrador de carga	21
Paso 3: Crear el equilibrador de carga de aplicación de broker	21
Paso 4: Lanzar los brokers	23
Paso 5: Crear un equilibrador de carga de aplicación del agente	24
Paso 6: Lanzar los agentes	25
Utilizar etiquetas	26
Configurar un servidor de autorización externo	28
Configurar la persistencia del broker	33
Configurar el broker para que persista en DynamoDB	34
Configurar el broker para que persista en MariaDB/MySQL	35
Integración con la puerta de enlace de conexión NICE DCV	36
Configurar el broker de Session Manager como Solucionador de sesiones para la puerta de enlace NICE DCV	36
Opcional: habilitar autenticación de cliente TLS	37
Servidor NICE DCV: asignación de DNS	39
Integración de Amazon CloudWatch	41

Mejora	43
Actualizar el agente de NICE DCV Session Manager	43
Actualizar el broker de NICE DCV Session Manager	45
Referencia de la CLI del broker	48
register-auth-server	49
Sintaxis	49
Opciones	49
Ejemplo	49
list-auth-servers	50
Sintaxis	49
Output	50
Ejemplo	49
unregister-auth-server	51
Sintaxis	49
Opciones	49
Output	50
Ejemplo	49
register-api-client	52
Sintaxis	49
Opciones	49
Output	50
Ejemplo	49
describe-api-clients	53
Sintaxis	49
Output	50
Ejemplo	49
unregister-api-client	55
Sintaxis	49
Opciones	49
Ejemplo	49
renew-auth-server-api-key	56
Sintaxis	49
Ejemplo	49
generate-software-statement	56
Sintaxis	49
Output	50

Ejemplo	49
describe-software-statements	58
Sintaxis	49
Output	50
Ejemplo	49
deactivate-software-statement	59
Sintaxis	49
Opciones	49
Ejemplo	49
describe-agent-clients	60
Sintaxis	49
Output	50
Ejemplo	49
unregister-agent-client	61
Sintaxis	49
Opciones	49
Ejemplo	49
register-server-dns-mappings	62
Sintaxis	49
Opciones	49
Ejemplo	49
describe-server-dns-mappings	63
Sintaxis	49
Output	50
Ejemplo	49
Referencia de archivos de configuración	66
Archivo de configuración del broker	66
Archivo de configuración del agente	82
Notas de la versión e historial de revisión	89
Notas de la versión	89
2023.1— 9 de noviembre de 2023	90
2023.0-15065 — 4 de mayo de 2023	90
2023.0-14852— 28 de marzo de 2023	90
2022.2-13907— 11 de noviembre de 2022	91
2022.1-13067— 29 de junio de 2022	91
2022.0-11952— 23 de febrero de 2022	91

2021.3-11591— 20 de diciembre de 2021	92
2021.2-11445— 18 de noviembre de 2021	92
2021.2-11190— 11 de octubre de 2021	92
2021.2-11042— 01 de septiembre de 2021	93
2021.1-10557— 31 de mayo de 2021	93
2021.0-10242— 12 de abril de 2021	94
2020.2-9662— 04 de diciembre de 2020	95
.....	95
Historial de documentos	95
.....	xcviii

¿Qué es NICE DCV Session Manager?

NICE DCV Session Manager es un conjunto de paquetes de software instalables (un agente y un broker) y una interfaz de programación de aplicaciones (API) que facilita a los desarrolladores y proveedores de software independientes (ISV) la creación de aplicaciones front-end que crean y gestionan mediante programación el ciclo de vida de las sesiones de NICE DCV en una flota de servidores NICE DCV.

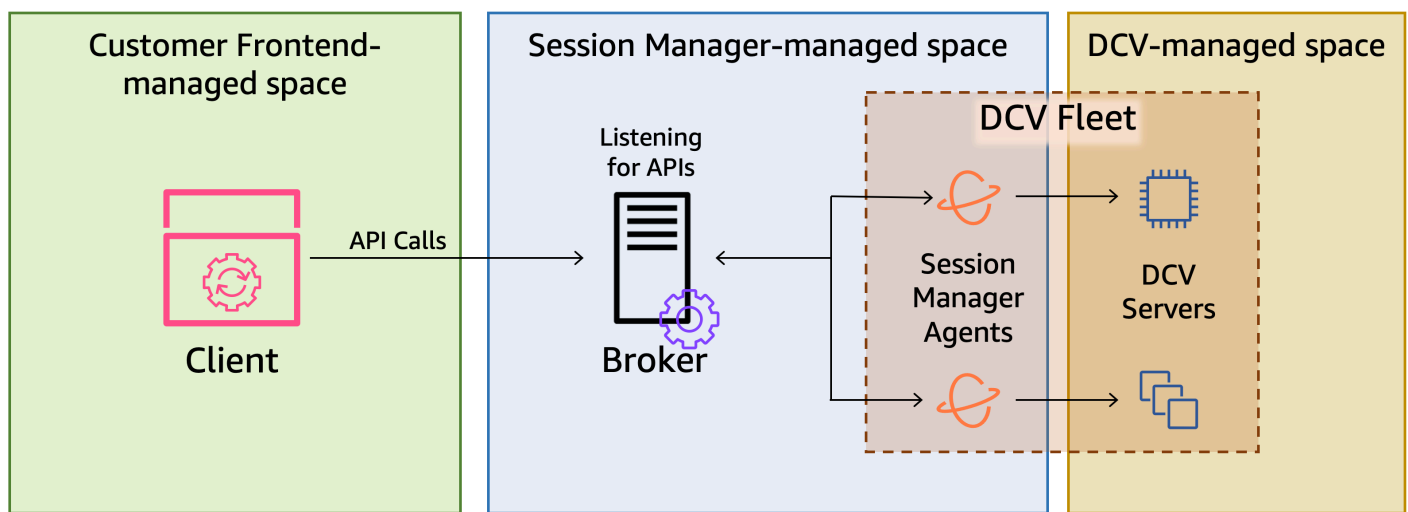
Esta guía explica cómo instalar y configurar el agente y el broker de Session Manager. Para obtener más información sobre el uso de las API de Session Manager, consulte la Guía para desarrolladores de NICE DCV Session Manager.

Temas

- [Cómo funciona Session Manager](#)
- [Características](#)
- [Limitaciones](#)
- [Precios](#)
- [Requisitos de NICE DCV Session Manager](#)

Cómo funciona Session Manager

En el siguiente diagrama se muestran los componentes de alto nivel de Session Manager.



Broker

El broker es un servidor web que aloja y expone las API de Session Manager. Recibe y procesa solicitudes de API para gestionar las sesiones de NICE DCV del cliente y, a continuación, pasa las instrucciones a los agentes correspondientes. El broker debe estar instalado en un host distinto de los servidores NICE DCV, pero debe ser accesible para el cliente y debe poder acceder a los agentes.

Agente

El agente está instalado en cada servidor NICE DCV de la flota. Los agentes reciben instrucciones del broker y las ejecutan en sus servidores NICE DCV correspondientes. Los agentes también supervisan el estado de los servidores NICE DCV y envían actualizaciones de estado periódicas al broker.

API

Session Manager expone un conjunto de interfaces de programación de aplicaciones (API) REST que se pueden utilizar para administrar las sesiones de NICE DCV en una flota de servidores NICE DCV. El broker aloja y expone las API. Los desarrolladores pueden crear clientes de administración de sesiones personalizados que llamen a las API.

Cliente

El cliente es la aplicación o el portal front-end que se desarrolla para llamar a las API de Session Manager expuestas por el broker. Los usuarios finales utilizan el cliente para gestionar las sesiones alojadas en los servidores NICE DCV de la flota.

Token de acceso

Para realizar una solicitud de API, debe proporcionar un token de acceso. Las API de cliente registradas pueden solicitar tokens al broker o a un servidor de autorización externo. Para solicitar el token y acceder a él, la API del cliente debe proporcionar credenciales válidas.

API del cliente

La API del cliente se genera a partir del archivo YAML de definición de la API de Session Manager, utilizando Swagger Codegen. La API del cliente se utiliza para realizar solicitudes de API.

Sesión de NICE DCV

Debe crear una sesión de NICE DCV en el servidor NICE DCV al que puedan conectarse sus clientes. Los clientes solo pueden conectarse a un servidor NICE DCV si hay una sesión activa.

NICE DCV admite sesiones virtuales y de consola. Las API de Session Manager sirven para gestionar el ciclo de vida de las sesiones de NICE DCV. Las sesiones de NICE DCV pueden tener uno de los siguientes estados:

- CREATING: el broker está creando la sesión.
- READY: la sesión está lista para aceptar conexiones de clientes.
- DELETING: se está eliminando la sesión.
- DELETED: se ha eliminado la sesión.
- UNKNOWN: no se puede determinar el estado de la sesión. Es posible que el agente y el broker no puedan comunicarse.

Características

DCV Session Manager ofrece las siguientes características:

- Proporciona información sobre sesiones de NICE DCV: obtenga información sobre las sesiones que se ejecutan en varios servidores NICE DCV.
- Gestione el ciclo de vida de varias sesiones de NICE DCV: cree o elimine distintas sesiones para varios usuarios en múltiples servidores NICE DCV con una sola solicitud de API.
- Admite etiquetas: utilice etiquetas personalizadas para identificar un grupo de servidores NICE DCV al crear sesiones.
- Administra los permisos para varias sesiones de NICE DCV: modifique los permisos de usuario para varias sesiones con una solicitud de API.
- Proporciona información de conexión: recupera la información de conexión del cliente para las sesiones de NICE DCV.
- Compatible con servidores locales y en la nube: utilice Session Manager en servidores de AWS, locales o alternativos basados en la nube.

Limitaciones

Session Manager no proporciona capacidades de aprovisionamiento de recursos. Si ejecuta NICE DCV en instancias de Amazon EC2, es posible que necesite utilizar servicios adicionales de AWS, como Amazon EC2 Auto Scaling, para gestionar el escalado de su infraestructura.

Precios

Session Manager está disponible sin coste alguno para los clientes de AWS que utilicen instancias de EC2.

Los clientes locales requieren una licencia NICE DCV Plus o DCV Professional Plus. Para obtener información sobre cómo comprar una licencia de NICE DCV Plus o NICE DCV Professional Plus, consulte [Cómo comprar](#) en el sitio web de NICE y busque un distribuidor o revendedor de NICE en su región. Para permitir que todos los clientes locales experimenten con DCV Session Manager, los requisitos de licencia solo se aplicarán a partir de NICE DCV versión 2021.0.

Para obtener más información, consulte [Licencias del servidor NICE DCV](#) en la Guía del administrador de NICE DCV.

Requisitos de NICE DCV Session Manager

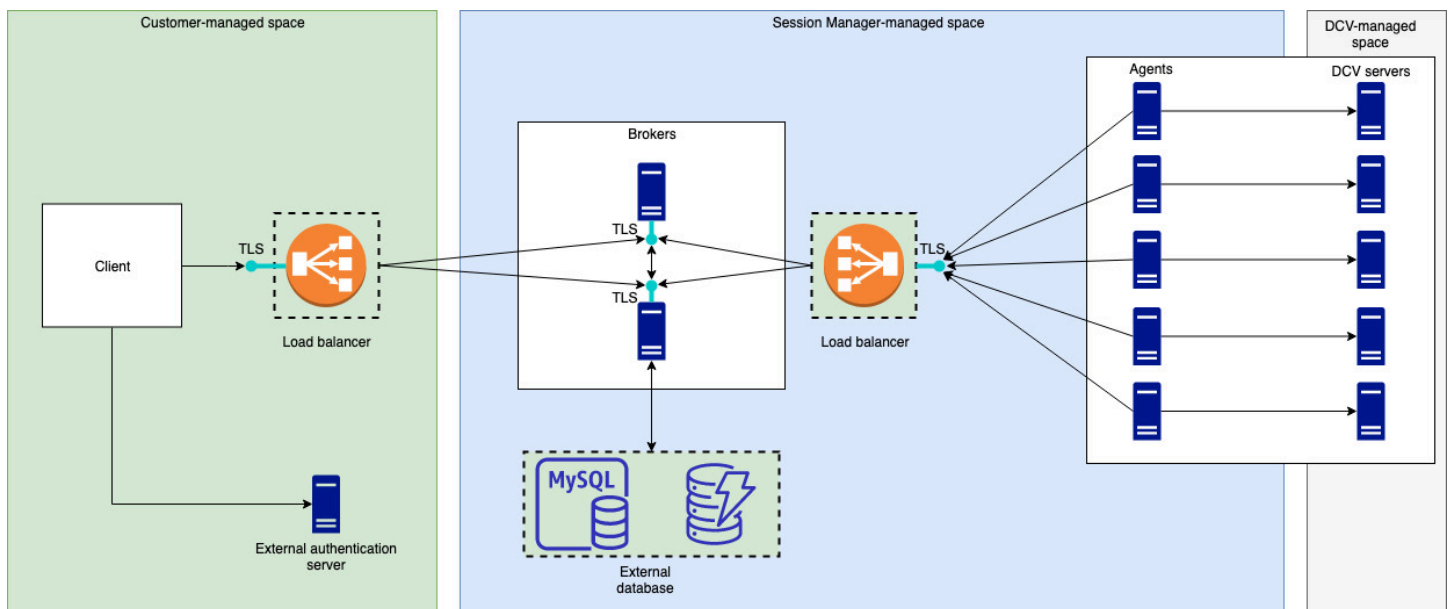
El agente y el broker de NICE DCV Session Manager tienen los siguientes requisitos.

	Agente	Agente
Sistema operativo	<ul style="list-style-type: none"> • Amazon Linux 2 • CentOS 7.6 o versiones posteriores • Flujo 8 de CentOS • RHEL 7.6 o versiones posteriores • RHEL 8.x • Rocky Linux 8.5 o versiones posteriores • Ubuntu 20.04 • Ubuntu 22.04 	<ul style="list-style-type: none"> • Windows <ul style="list-style-type: none"> • Windows Server 2019 • Windows Server 2016 • Windows Server 2012 R2 • Servidor Linux <ul style="list-style-type: none"> • Amazon Linux 2 • CentOS 7.6 o versiones posteriores • Flujo 8 de CentOS • RHEL 7.6 o versiones posteriores • RHEL 8.x • Rocky Linux 8.5 o versiones posteriores • Ubuntu 20.04 • Ubuntu 22.04

	Agente	Agente
		<ul style="list-style-type: none"> • SUSE Linux Enterprise 12 con SP4 o posterior • SUSE Linux Enterprise 15
Arquitectura	<ul style="list-style-type: none"> • x86 de 64 bits • ARM de 64 bits 	<ul style="list-style-type: none"> • x86 de 64 bits • ARM de 64 bits (Amazon Linux 2, CentOS 7.x/8.x y RHEL 7.x/8.x solo) • ARM de 64 bits (Ubuntu 22.04)
Memoria	8 GB	4 GB
Versión de NICE DCV	NICE DCV 2020.2 y versiones posteriores	NICE DCV 2020.2 y versiones posteriores
Requisitos adicionales	Java 11	-

Requisitos de red y conectividad

En el siguiente diagrama se ofrece información de alto nivel sobre los requisitos de conectividad y red de Session Manager.



El broker debe estar instalado en un host aparte, pero debe tener conectividad de red con los agentes de los servidores NICE DCV. Si opta por tener varios brokers para mejorar la disponibilidad, debe instalar y configurar cada uno en un host aparte y utilizar uno o más equilibradores de carga para gestionar el tráfico entre el cliente y los brokers, y entre los brokers y los agentes. Los brokers también deberían poder comunicarse entre sí para intercambiar información sobre los servidores y las sesiones de NICE DCV. Los brokers pueden almacenar sus claves y datos de estado en una base de datos externa y tener esta información disponible tras el reinicio o la finalización. Esto ayuda a mitigar el riesgo de perder información importante sobre los brokers gracias a que se mantiene en la base de datos externa. Podrá recuperarla más adelante. Si decide tenerla, debe configurar la base de datos externa y configurar los brokers. DynamoDB, MariaDB y MySQL son compatibles. Puede ver que los parámetros de configuración aparecen en el [archivo de configuración del broker](#).

Los agentes deben poder iniciar conexiones HTTP seguras, persistentes y bidireccionales con el broker.

Su cliente, o su aplicación front-end, debe poder acceder al broker para llamar a las API. El cliente también debe poder acceder a su servidor de autenticación.

Configuración de NICE DCV Session Manager

En la siguiente sección se explica cómo instalar Session Manager con un único broker y varios agentes. Puede utilizar varios brokers para mejorar la escalabilidad y el rendimiento. Para obtener más información, consulte [Session Manager de escalado](#).

Para configurar NICE DCV Session Manager, haga lo siguiente:

Pasos

- [Paso 1: Preparar los servidores NICE DCV](#)
- [Paso 2: Configurar el broker de NICE DCV Session Manager](#)
- [Paso 3: Configurar el agente de NICE DCV Session Manager](#)
- [Paso 4: Configurar el servidor NICE DCV para utilizar el broker como servidor de autenticación](#)
- [Paso 5: Verificar las instalaciones](#)

Paso 1: Preparar los servidores NICE DCV

Debe tener una flota de servidores NICE DCV con los que desee utilizar Session Manager. Para obtener más información sobre los requisitos del servidor NICE DCV, consulte [Installing the NICE DCV server](#) en la Guía del administrador de NICE DCV.

En los servidores NICE DCV de Linux, Session Manager utiliza un usuario de servicio local denominado `dcvsmagent`. Este usuario se crea automáticamente cuando se instala el agente de Session Manager. Debe conceder a este servicio privilegios de administrador de usuarios para NICE DCV de modo que pueda realizar acciones en nombre de otros usuarios. Para conceder privilegios de administrador de usuarios al servicio de Session Manager, haga lo siguiente:

Para agregar el usuario de servicio local para servidores NICE DCV de Linux

1. Abra `/etc/dcv/dcv.conf` con el editor de texto que prefiera.
2. Agregue el parámetro `administrators` a la sección `[security]` y especifique el usuario de Session Manager. Por ejemplo:

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

3. Guarde y cierre el archivo.
4. Detenga y reinicie el servidor NICE DCV.

Session Manager solo puede crear sesiones NICE DCV en nombre de los usuarios que ya existen en el servidor NICE DCV. Si se realiza una solicitud para crear una sesión para un usuario que no existe, se produce un error en la solicitud. Por tanto, debe asegurarse de que cada usuario final previsto tiene un usuario de sistema válido en el servidor NICE DCV.

Tip

Si tiene intención de utilizar varios hosts de broker o servidores NICE DCV con agentes, le recomendamos que configure solo un broker y un servidor NICE DCV con un agente. Para ello, siga estos pasos: cree imágenes de máquina de Amazon (AMI) de los hosts con las configuraciones completadas y, a continuación, utilice las AMI para lanzar el resto de los brokers y servidores NICE DCV. Como alternativa, puede usar AWS Systems Manager para ejecutar los comandos en varias instancias de forma remota.

Paso 2: Configurar el broker de NICE DCV Session Manager

El broker debe estar instalado en un host de Linux. Para obtener más información acerca de las distribuciones de Linux admitidas, consulte [Requisitos de NICE DCV Session Manager](#). Instale el broker en un host independiente del agente y del host del servidor NICE DCV. El host puede estar instalado en una red privada diferente, pero debe poder conectarse y comunicarse con el agente.

Para instalar e iniciar el broker

1. Conéctese al host en el que desea instalar el broker.
2. Los paquetes del se firman digitalmente con una firma GPG segura. Para permitir al administrador de paquetes comprobar la firma de paquetes, debe importar la clave de NICE GPG. Ejecute el siguiente comando para importar la clave de NICE GPG.

- Amazon Linux 2, RHEL, CentOS y 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. Descargue el paquete de instalación.

- Amazon Linux 2, RHEL 7.x y 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 y 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. Instale el paquete.

- Amazon Linux 2, RHEL 7.x y 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 y 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. Compruebe que la versión predeterminada del entorno Java es la 11

```
$ java -version
```

Si no es así, puede configurar de forma explícita el directorio principal de Java que el broker utilizará para identificar la versión correcta de Java. Esto se hace configurando el parámetro `broker-java-home` en el archivo de configuración del broker. Para obtener más información, consulte [Archivo de configuración del broker](#).

6. Inicie el servicio del broker y asegúrese de que se inicie de forma automática cuando se inicia la instancia.

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

7. Coloque una copia del certificado autofirmado del broker en su directorio de usuario. Lo necesitará al instalar los agentes en el siguiente paso.

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

Paso 3: Configurar el agente de NICE DCV Session Manager

El agente debe estar instalado en todos los hosts del servidor NICE DCV de la flota. El agente puede estar instalado en servidores Windows y Linux. Para obtener más información acerca de los sistemas operativos compatibles, consulte [Requisitos de NICE DCV Session Manager](#).

Requisitos previos

El servidor NICE DCV debe estar instalado en todos el host antes de instalar el agente.

Linux host

Note

[El agente Session Manager está disponible para las distribuciones y arquitecturas de Linux que se indican en Requisitos:](#)

Siga estas instrucciones para instalar el agente en hosts x86 de 64 bits. Para instalar el agente en hosts ARM de 64 bits, sustituya *x86_64* por. *aarch64* *Para Ubuntu, sustituya amd64 por. arm64*

Para instalar el agente en un host Linux

1. Los paquetes del se firman digitalmente con una firma GPG segura. Para permitir al administrador de paquetes comprobar la firma de paquetes, debe importar la clave de NICE GPG. Ejecute el siguiente comando para importar la clave de NICE GPG.

- Amazon Linux 2, RHEL, CentOS y 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. Descargue el paquete de instalación.

- Amazon Linux 2, RHEL 7.x y 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 y 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. Instale el paquete.

- Amazon Linux 2, RHEL 7.x y 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 y 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. Coloque una copia del certificado autofirmado del broker (que copió en el paso anterior) en el directorio `/etc/dcv-session-manager-agent/` del agente.
5. Abra `/etc/dcv-session-manager-agent/agent.conf` con su editor de texto preferido y haga lo siguiente.
 - Para `broker_host`, especifique el nombre DNS del host en el que está instalado el broker.

 Important

Si el broker se ejecuta en una instancia de Amazon EC2, para `broker_host` debe especificar la dirección IPv4 privada de la instancia.

- (Opcional) Para `broker_port`, especifique el puerto a través del cual se va a comunicar con el broker. De forma predeterminada, el agente y el broker se comunican a través del puerto 8445. Cambie esto solo si necesita utilizar otro puerto. Si lo cambia, asegúrese de que el broker está configurado para utilizar el mismo puerto.
- Para `ca_file`, especifique la ruta completa del archivo de certificado que copió en el paso anterior. Por ejemplo:

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

Como alternativa, si desea deshabilitar la verificación de TLS, establezca `tls_strict` en `false`.


6. Guarde y cierre el archivo.
7. Para iniciar el agente, ejecute el comando siguiente.

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

Windows host

Para instalar el agente en un host de Windows

1. Descargue el [instalador del agente](#).
2. Ejecute el instalador. En la pantalla de bienvenida, elija Next.
3. En la pantalla del EULA, lea atentamente el contrato de licencia y, si está de acuerdo, seleccione Acepto los términos y condiciones y elija Siguiente.
4. Elija Instalar para comenzar la instalación.
5. Coloque una copia del certificado autofirmado del broker (que copió en el paso anterior) en la carpeta C:\Program Files\NICE\DCVSessionManagerAgent\conf\ del agente.
6. Abra C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf con su editor de texto preferido y, a continuación, haga lo siguiente:
 - Para broker_host, especifique el nombre DNS del host en el que está instalado el broker.

 Important

Si el broker se ejecuta en una instancia de Amazon EC2, para broker_host debe especificar la dirección IPv4 privada de la instancia.

- (Opcional) Para broker_port, especifique el puerto a través del cual se va a comunicar con el broker. De forma predeterminada, el agente y el broker se comunican a través del puerto 8445. Cambie esto solo si necesita utilizar otro puerto. Si lo cambia, asegúrese de que el broker está configurado para utilizar el mismo puerto.
- Para ca_file, especifique la ruta completa del archivo de certificado que copió en el paso anterior. Por ejemplo:

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

Como alternativa, si desea deshabilitar la verificación de TLS, establezca tls_strict en false.

7. Guarde y cierre el archivo.
8. Pare y reinicie el servicio del agente para que los cambios surtan efecto. En el símbolo del sistema, ejecute los siguientes comandos.

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

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

Paso 4: Configurar el servidor NICE DCV para utilizar el broker como servidor de autenticación

Configure el servidor NICE DCV para que utilice el broker como servidor de autenticación externo para validar los tokens de conexión del cliente. También debe configurar el servidor NICE DCV para que confíe en la CA autofirmada del broker.

Linux NICE DCV server

Para agregar el usuario de servicio local para servidores NICE DCV de Linux

1. Abra `/etc/dcv/dcv.conf` con el editor de texto que prefiera.
2. Agregue los parámetros `ca-file` y `auth-token-verifier` a la sección `[security]`.

Para `ca-file`, especifique la ruta a la CA autofirmada del broker que copió en el host en el paso anterior.

Para `auth-token-verifier`, especifique la URL del verificador de tokens del broker con el siguiente formato: `https://broker_ip_or_dns:port/agent/validate-authentication-token`. Especifique el puerto utilizado para la comunicación entre el broker y el agente, que es 8445 de forma predeterminada. Si ejecuta el broker en una instancia de Amazon EC2, debe utilizar el DNS privado o la dirección IP privada.

Por ejemplo

```
[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. Guarde y cierre el archivo.

4. Detenga y reinicie el servidor NICE DCV. Para obtener más información, consulte [Stopping the NICE DCV Server](#) y [Starting the NICE DCV Server](#) en la Guía del administrador de NICE DCV.

Windows NICE DCV server

Servidores NICE DCV de Windows

1. Abra el Editor del Registro de Windows y vaya a la clave HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/security/.
2. Abra el parámetro ca-file. Para Value data, especifique la ruta de la CA autofirmada del broker que copió en el host en el paso anterior.

Note

Si el parámetro no existe, cree un nuevo parámetro de cadena y llámelo ca-file.

3. Abra el parámetro. auth-token-verifier Para Value data, especifique la URL del verificador de tokens del broker con el siguiente formato: `https://broker_ip_or_dns:port/agent/validate-authentication-token`. Especifique el puerto utilizado para la comunicación entre el broker y el agente, que es 8445 de forma predeterminada. Si ejecuta el broker en una instancia de Amazon EC2, debe utilizar el DNS privado o la dirección IP privada.

Note

Si el parámetro no existe, cree un nuevo parámetro de cadena y llámelo auth-token-verifier.

4. Haga clic en Aceptar y cierre el Editor del Registro de Windows.
5. Detenga y reinicie el servidor NICE DCV. Para obtener más información, consulte [Stopping the NICE DCV Server](#) y [Starting the NICE DCV Server](#) en la Guía del administrador de NICE DCV.

Paso 5: Verificar las instalaciones

Temas

- [Verificar el agente](#)

- [Comprobar el broker](#)

Verificar el agente

Una vez que haya instalado el broker y el agente, asegúrese de que el agente está ejecutándose y de que puede conectarse al broker.

Host del agente de Linux

El comando que se debe ejecutar depende de la versión.

- A partir de la versión 2022.0

En el host del agente, ejecute el siguiente comando:

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

- Versiones anteriores a 2022.0

En el host del agente, ejecute el siguiente comando y especifique el año, el mes y el día actuales.

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

Por ejemplo

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

Si el agente está en ejecución y puede conectarse al broker, el comando debería devolver success.

Si el comando devuelve una salida diferente, inspeccione el archivo de registro del agente para obtener más información. Los archivos de registro se encuentran aquí: `/var/log/dcv-session-manager-agent/`.

Host del agente de Windows

Abra el archivo de registro del agente, que se encuentra en `C:\ProgramData\NICE\DCVSessionManagerAgent\log`.

Si el archivo de registro incluye una línea similar a la siguiente, significa que el agente está en ejecución y puede conectarse al broker.

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

Si el archivo de registro no tiene una línea similar, inspeccione el archivo de registro en busca de errores.

Comprobar el broker

Una vez que haya instalado el broker y el agente, asegúrese de que el broker está ejecutándose y que pueden acceder a él los usuarios y las aplicaciones front-end.

En una computadora que debería poder comunicarse con el broker, ejecute el siguiente comando:

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

Si la verificación se realiza correctamente, el broker devuelve lo siguiente:

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


Configurar NICE DCV Session Manager

En esta sección se explica cómo realizar la configuración avanzada de Session Manager.

Temas

- [Session Manager de escalado](#)
- [Utilizar etiquetas para identificar servidores NICE DCV](#)
- [Configurar un servidor de autorización externo](#)
- [Configurar la persistencia del broker](#)
- [Integración con la puerta de enlace de conexión NICE DCV](#)
- [Integración de Amazon CloudWatch](#)

Session Manager de escalado

Para permitir una alta disponibilidad y mejorar el rendimiento puede configurar Session Manager para que utilice varios agentes y brokers. Si piensa utilizar varios agentes y brokers, le recomendamos que instale y configure solo un host de agente y broker, cree imágenes de máquina de Amazon (AMI) a partir de esos hosts y, a continuación, lance los hosts restantes desde las AMI.

De forma predeterminada, Session Manager admite el uso de varios agentes sin necesidad de configuración adicional. Sin embargo, si tiene intención de utilizar varios brokers, debe utilizar un equilibrador de carga para equilibrar el tráfico entre el cliente frontend y los brokers, y entre estos y los agentes. Usted es el propietario exclusivo y el gestor de la instalación y la configuración del equilibrador de carga.

En la siguiente sección, se explica cómo configurar Session Manager para utilizar varios hosts con un Equilibrador de carga de aplicación.

Pasos

- [Paso 1: Crear un perfil de instancia](#)
- [Paso 2: Preparar el certificado SSL para el equilibrador de carga](#)
- [Paso 3: Crear el equilibrador de carga de aplicación de broker](#)
- [Paso 4: Lanzar los brokers](#)
- [Paso 5: Crear un equilibrador de carga de aplicación del agente](#)
- [Paso 6: Lanzar los agentes](#)

Paso 1: Crear un perfil de instancia

Debe adjuntar un perfil de instancia a los hosts Broker y Agent que les concedan permiso para usar las API de Elastic Load Balancing. Para obtener más información, consulte [Roles de IAM para Amazon EC2](#) en la Guía del usuario de Amazon EC2.

Cómo crear un perfil de instancia

1. Cree un rol AWS Identity and Access Management (de IAM) que defina los permisos que se van a usar en el perfil de la instancia. Utilice la siguiente política de confianza:

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

Asocie la siguiente política:

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

```
}  
]  
}
```

Para obtener más información, consulte [Creación de un rol de IAM \(AWS CLI\)](#) en Guía del usuario de IAM.

2. Cree un nuevo perfil de instancia. Para obtener más información, consulte [create-instance-profile](#) en la Referencia de comandos de la AWS CLI .
3. Agregue el Rol de IAM al perfil de instancias. Para obtener más información, consulte [add-role-to-instance-profile](#) en la Referencia de comandos de la AWS CLI .
4. Asocie el perfil de instancia a los hosts de broker. Para obtener más información, consulte [Adjuntar un rol de IAM a una instancia](#) en la Guía del usuario de Amazon EC2.

Paso 2: Preparar el certificado SSL para el equilibrador de carga

Cuando se utiliza HTTPS para el agente de escucha del equilibrador de carga, es preciso implementar en él un certificado SSL. El equilibrador de carga usará dicho certificado para terminar la conexión y descifrar las solicitudes de los clientes antes de enviárselas a los destinos.

Para preparar el certificado SSL

1. Cree una entidad de certificación (CA) privada AWS Certificate Manager Private Certificate Authority (ACM PCA). Para obtener más información, consulte [Procedimientos para crear una CA](#) en la Guía del usuario de AWS Certificate Manager Private Certificate Authority.
2. Instale la CA. Para obtener más información, consulte [Instalación de un certificado de CA raíz](#) en la Guía del usuario de AWS Certificate Manager Private Certificate Authority.
3. Solicite un nuevo certificado privado firmado por la CA. Como nombre de dominio, utilice `*.region.elb.amazonaws.com` y especifique la región en la que desea crear el equilibrador de carga. Para obtener más información, consulte [Solicitud de un certificado privado](#) en la Guía del usuario de AWS Certificate Manager Private Certificate Authority.

Paso 3: Crear el equilibrador de carga de aplicación de broker

Cree un equilibrador de carga de aplicación para equilibrar el tráfico entre sus clientes front-end y los brokers.

Para crear el equilibrador de carga

1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.

En el panel de navegación, elija Equilibradores de carga y, a continuación, elija Crear equilibrador de carga. En Tipo de equilibrador de carga, elija Equilibrador de carga de aplicación.

2. En Paso 1: Configurar equilibrador de carga, haga lo siguiente:
 - a. En Nombre, escriba un nombre descriptivo para el equilibrador de carga.
 - b. En Esquema, seleccione Con acceso a Internet.
 - c. En Protocolo de equilibrador de carga, seleccione HTTPS y, en Puerto de equilibrador de carga, introduzca 8443.
 - d. Para VPC, seleccione la VPC que desee utilizar y, a continuación, seleccione todas las subredes de esa VPC.
 - e. Elija Siguiente.
3. En Paso 2: Configurar grupo de seguridad, haga lo siguiente:
 - a. En Tipo de certificado, seleccione Elegir un certificado de ACM.
 - b. En Nombre del certificado, seleccione el certificado privado que solicitó anteriormente.
 - c. Elija Siguiente.
4. En el Paso 3: Configurar grupos de seguridad, cree un nuevo grupo de seguridad o seleccione un grupo de seguridad existente que permita tráfico entrante y saliente entre su cliente front-end y los brokers a través de HTTPS y el puerto 8443.

Elija Siguiente.

5. En Paso 4: Configurar enrutamiento, haga lo siguiente:
 - a. En Grupo de destino, seleccione Nuevo grupo de destino.
 - b. En Name (Nombre), escriba un nombre para el grupo de destino.
 - c. En Tipo de destino, elija Instancia.
 - d. En Protocolo, seleccione HTTPS. En Puerto, escriba 8443. En Versión del protocolo, seleccione HTTP1.
 - e. En Protocolo de comprobación de estado, elija HTTPS y en Ruta, introduzca /health.
 - f. Elija Siguiente.

6. En Paso 5: Registrar destinos, seleccione Siguiente.

7. Seleccione Crear.

Paso 4: Lanzar los brokers

Cree un broker inicial y configúrelo para utilizar el equilibrador de carga, cree una AMI desde el broker y, a continuación, utilícela para lanzar los brokers restantes. Esto garantiza que todos los brokers estén configurados para utilizar la misma CA y la misma configuración de equilibrador de carga.

Para lanzar los brokers

1. Inicie y configure el host del broker inicial. Para obtener más información sobre cómo instalar y configurar el broker, consulte [Paso 2: Configurar el broker de NICE DCV Session Manager](#).

Note

No se necesita un certificado autofirmado del broker, ya que utilizamos un equilibrador de carga de aplicación.

2. Conéctese al broker, abra `/etc/dcv-session-manager-broker/session-manager-broker.properties` con su editor de texto preferido y haga lo siguiente:
 - a. Comente el parámetro `broker-to-broker-discovery-addresses` colocando un hash (`#`) al principio de la línea.
 - b. Para `broker-to-broker-discovery-aws-region`, introduzca la región en la que creó el equilibrador de carga de aplicación.
 - c. Para `broker-to-broker-discovery-aws-alb-target-group-arn`, introduzca el ARN del grupo de destino asociado al equilibrador de carga del broker.
 - d. Guarde y cierre el archivo.
3. Detenga la instancia de broker.
4. Cree una AMI a partir de la instancia de broker detenida. Para obtener más información, consulte [Crear una AMI de Linux desde una instancia](#) en Amazon EC2 Guía del usuario de instancias de Linux.
5. Utilice la AMI para lanzar los brokers restantes.
6. Asigne el perfil de instancia que creó a todas las instancias de broker.

7. Asigne un grupo de seguridad que permita a tráfico de red de broker a broker y de broker a equilibrador de carga a todas las instancias de broker. Para obtener más información sobre puertos de red, consulte [Archivo de configuración del broker](#).
8. Registre todas las instancias de broker como destinos para el equilibrador de carga de broker. Para obtener más información, consulte [Registro de destinos con el grupo de destino](#) en la Guía del usuario de los equilibradores de carga de aplicación.

Paso 5: Crear un equilibrador de carga de aplicación del agente

Cree un equilibrador de carga de aplicación para equilibrar los agentes y los brokers.

Para crear el equilibrador de carga

1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.

En el panel de navegación, elija Equilibradores de carga y, a continuación, elija Crear equilibrador de carga. En Tipo de equilibrador de carga, elija Equilibrador de carga de aplicación.

2. En Paso 1: Configurar equilibrador de carga, haga lo siguiente:
 - a. En Nombre, escriba un nombre descriptivo para el equilibrador de carga.
 - b. En Esquema, seleccione Con acceso a Internet.
 - c. En Protocolo de equilibrador de carga, seleccione HTTPS y, en Puerto de equilibrador de carga, introduzca 8445.
 - d. Para VPC, seleccione la VPC que desee utilizar y, a continuación, seleccione todas las subredes de esa VPC.
 - e. Elija Siguiente.
3. En Paso 2: Configurar grupo de seguridad, haga lo siguiente:
 - a. En Tipo de certificado, seleccione Elegir un certificado de ACM.
 - b. En Nombre del certificado, seleccione el certificado privado que solicitó anteriormente.
 - c. Elija Siguiente.
4. En Paso 3: Configurar grupos de seguridad, cree un nuevo grupo de seguridad o seleccione un grupo de seguridad existente que permita tráfico entrante y saliente de agentes y brokers a través de HTTPS y el puerto 8445.

Elija Siguiente.

5. En Paso 4: Configurar enrutamiento, haga lo siguiente:
 - a. En Grupo de destino, seleccione Nuevo grupo de destino.
 - b. En Name (Nombre), escriba un nombre para el grupo de destino.
 - c. En Tipo de destino, elija Instancia.
 - d. En Protocolo, seleccione HTTPS. En Puerto, escriba 8445. En Versión del protocolo, seleccione HTTP1.
 - e. En Protocolo de comprobación de estado, elija HTTPS y, en Ruta, introduzca `/health`.
 - f. Elija Siguiente.
6. En Paso 5: Registrar destinos, seleccione todas las instancias de broker y elija Agregar a registrado. Elija Siguiente: Revisar.
7. Seleccione Crear.

Paso 6: Lanzar los agentes

Cree un agente inicial y configúrelo para utilizar el equilibrador de carga, cree una AMI desde el agente y, a continuación, utilícela para lanzar los agentes restantes. Esto garantiza que todos los agentes estén configurados para utilizar la misma configuración de equilibrador de carga.

Para iniciar los agentes

1. Prepare el servidor NICE DCV. Para obtener más información, consulte [Paso 1: Preparar los servidores NICE DCV](#).
2. Coloque una copia de la clave pública de CA creada en [Paso 2: Preparar el certificado SSL para el equilibrador de carga](#). Elija o cree un directorio que pueda leer cualquier usuario. El archivo de clave pública de CA también debe ser legible para cualquier usuario.
3. Instale y configure el agente. Para obtener más información acerca de cómo instalar y configurar el agente, consulte [Paso 3: Configurar el agente de NICE DCV Session Manager](#).

Important

Cuando modifique el archivo de configuración del agente:

- para el parámetro `broker_host`, introduzca el DNS del equilibrador de carga del agente

- para el parámetro `ca_file`, introduzca la ruta del archivo de clave pública de CA creado en el paso anterior

4. Configure el servidor NICE DCV para utilizar el broker como servidor de autenticación. Para obtener más información, consulte [Paso 4: Configurar el servidor NICE DCV para utilizar el broker como servidor de autenticación](#).

 Important

Al modificar el archivo de configuración del servidor NICE DCV:

- para el parámetro `ca-file`, introduzca la misma ruta del archivo de clave pública de CA creado en el paso anterior
- para el parámetro `auth-token-verifier`, utilice el DNS del equilibrador de carga del agente para *broker_ip_or_dns*

5. Detenga la instancia del agente.
6. Cree una AMI a partir de la instancia de agente detenida. Para obtener más información, consulte [Crear una AMI de Linux desde una instancia](#) en Amazon EC2 Guía del usuario de instancias de Linux.
7. Utilice la AMI para lanzar los agentes restantes y asignarles el perfil de instancia que creó a todos ellos.
8. Asigne un grupo de seguridad que permita a tráfico de red de agente a equilibrador de carga a todas las instancias de agente. Para obtener más información sobre puertos de red, consulte [Archivo de configuración del agente](#).

Utilizar etiquetas para identificar servidores NICE DCV

Puede asignar etiquetas personalizadas a los agentes de Session Manager para ayudar a identificarlos y clasificarlos a ellos y a los servidores NICE DCV a los que están asociados. Al crear una nueva sesión de NICE DCV, puede identificar un grupo de servidores NICE DCV con las etiquetas que se asignan a sus respectivos agentes. Para obtener más información sobre cómo identificar servidores NICE DCV con etiquetas de agente, consulte [CreateSessionRequests](#) en la Guía para desarrolladores de Session Manager.

Una etiqueta consta de un par de clave y valor de etiqueta, y se puede utilizar cualquier par de información que sea adecuado para el caso de uso o el entorno. Puede etiquetar a los agentes en función de la configuración de hardware de su host. Por ejemplo, puede etiquetar todos los agentes con hosts que tengan 4 GB de memoria con `ram=4GB`. O puede etiquetar a los agentes según su propósito. Por ejemplo, puede etiquetar todos los agentes que se ejecutan en hosts de producción con `purpose=production`.

Para asignar etiquetas a un agente

1. Con su editor de texto preferido, cree un nuevo archivo y asígnele un nombre descriptivo, por ejemplo, `agent_tags.toml`. El tipo de archivo debe ser `.toml` y el contenido del archivo debe especificarse en el formato de archivo TOML.
2. En el archivo, añada cada nuevo par de clave y valor de etiqueta en una línea nueva utilizando el formato `key=value`. Por ejemplo:

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

3. Abra el archivo de configuración del agente (`/etc/dcv-session-manager-agent/agent.conf` para Linux o `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` para Windows). En `tags_folder`, especifique la ruta del directorio en el que se encuentra el archivo de etiqueta.

Si el directorio contiene varios archivos de etiqueta, se aplican al agente todas las etiquetas definidas en los archivos. Los archivos se leen en orden alfabético. Si hay archivos que contienen una etiqueta con la misma clave, el valor se sobrescribe con el valor del último archivo leído.

4. Guarde y cierre el archivo .
5. Detenga y reinicie el agente.

- Windows

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

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

- Linux

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

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

Configurar un servidor de autorización externo

El servidor de autorización es el servidor responsable de autenticar y autorizar los SDK y los agentes del cliente.

De forma predeterminada, Session Manager utiliza el broker como servidor de autorización para generar los tokens de acceso de OAuth 2.0 para SDK cliente e instrucciones de software para agentes. Si utiliza el broker como servidor de autorización, no se requiere configuración adicional.

Puede configurar Session Manager para que utilice Amazon Cognito como servidor de autorización externo en lugar del broker. Para obtener más información sobre Amazon Cognito, consulte la [Guía para desarrolladores de Amazon Cognito](#).

Para utilizar Amazon Cognito como servidor de autorización

1. Cree un grupo de usuarios de Amazon Cognito. Para obtener más información sobre grupos de usuarios, consulte [Características de Amazon Cognito](#) en la Guía para desarrolladores de Amazon Cognito.

Utilice el comando [create-user-pool](#) y especifique el nombre de un grupo y la región en la que desea crearlo.

En este ejemplo, llamamos al grupo `dcv-session-manager-client-app` y lo creamos en `us-east-1`.

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

Ejemplo de salida

```
{  
  "UserPoolClient": {  
    "UserPoolId": "us-east-1_QLEXAMPLE",  
    "ClientName": "dcv-session-manager-client-app",
```

```

    "ClientId": "15hhd8jij74hf32f24uEXAMPLE",
    "LastModifiedDate": 1602510048.054,
    "CreationDate": 1602510048.054,
    "RefreshTokenValidity": 30,
    "AllowedOAuthFlowsUserPoolClient": false
  }
}

```

Tome nota del `userPoolId`, ya que lo necesitará en el siguiente paso.

2. Cree un nuevo dominio para el grupo de usuarios. Utilice el comando [create-user-pool-domain](#) y especifique un nombre de dominio y el `userPoolId` del grupo de usuarios que creó en el paso anterior.

En este ejemplo, el nombre del dominio es `mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE` y lo creamos en `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

```

Ejemplo de salida

```

{
  "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": {}
  }
}

```

El formato del dominio del grupo de usuarios es el siguiente:

`https://domain_name.auth.region.amazoncognito.com`. En este ejemplo, el dominio del grupo de usuarios es `https://mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE.auth.us-east-1.amazoncognito.com`.

3. Cree un cliente del grupo de usuarios. Utilice el comando [create-user-pool-client](#) y especifique el `userPoolId` del grupo de usuarios que creó, un nombre para el cliente y la región en la que desea crearlo. Además, incluya la opción `--generate-secret` para especificar si desea generar un secreto para el cliente del grupo de usuarios que está creando.

En este caso, el nombre del cliente es `dcv-session-manager-client-app` y lo creamos en la región `us-east-1`.

```
$ 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
```

Ejemplo de salida

```
{  
  "UserPoolClient": {  
    "UserPoolId": "us-east-1_QLEXAMPLE",  
    "ClientName": "dcv-session-manager-client-app",  
    "ClientId": "219273hp6k2ut5cugg9EXAMPLE",  
    "ClientSecret": "1vp5e8nec7cbf4m9me55mbmht91u61h1h0a78rq1qki11EXAMPLE",  
    "LastModifiedDate": 1602510291.498,  
    "CreationDate": 1602510291.498,  
    "RefreshTokenValidity": 30,  
    "AllowedOAuthFlowsUserPoolClient": false  
  }  
}
```

Note

Tome nota de `ClientId` y `ClientSecret`. Deberá facilitar a los desarrolladores esta información que necesitarán cuando soliciten tokens de acceso para las solicitudes de la API.

4. Cree un nuevo servidor de recursos OAuth2.0 para el grupo de usuarios. Un servidor de recursos contiene recursos de acceso protegido. Gestiona las solicitudes autenticadas de tokens de acceso.

Utilice el comando [create-resource-server](#) y especifique el `userPoolId` del grupo de usuarios, un identificador único y un nombre para el servidor de recursos, el ámbito y la región en la que desea crearlo.

En este ejemplo, utilizamos `dcv-session-manager` como identificador y nombre, y `sm_scope` como nombre del ámbito y descripción.

```
$ 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
```

Ejemplo de salida

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

5. Actualice el cliente del grupo de usuarios.

Utilice el comando [update-user-pool-client](#). Especifique el `userPoolId` del grupo de usuarios, el `ClientId` del cliente del grupo de usuarios y la región. Para `--allowed-o-auth-flows`, especifique `client_credentials` si desea indicar que el cliente debe obtener los tokens de acceso del punto de conexión del token utilizando una combinación de ID de cliente y de secreto de cliente. Para `--allowed-o-auth-scopes`, especifique el identificador del servidor de recursos y el nombre del ámbito de este modo: *resource_server_identifier/scope_name*. Incluya el `--allowed-o-auth-flows-user-pool-client` para indicar que el cliente puede seguir el protocolo OAuth al interactuar con grupos de usuarios de Cognito.

```
$ aws cognito-idp update-user-pool-client --user-pool-id us-east-1_QLEXAMPLE --
client-id 219273hp6k2ut5cugg9EXAMPLE --allowed-o-auth-flows client_credentials --
allowed-o-auth-scopes dcv-session-manager/sm_scope --allowed-o-auth-flows-user-
pool-client --region us-east-1
```

Ejemplo de salida

```
{
  "UserPoolClient": {
    "UserPoolId": "us-east-1_QLEXAMPLE",
    "ClientName": "dcv-session-manager-client-app",
    "ClientId": "219273hp6k2ut5cugg9EXAMPLE",
    "ClientSecret": "1vp5e8nec7cbf4m9me55mbmht91u61h1h0a78rq1qki11EXAMPLE",
    "LastModifiedDate": 1602512103.099,
    "CreationDate": 1602510291.498,
    "RefreshTokenValidity": 30,
    "AllowedOAuthFlows": [
      "client_credentials"
    ],
    "AllowedOAuthScopes": [
      "dcv-session-manager/sm_scope"
    ],
    "AllowedOAuthFlowsUserPoolClient": true
  }
}
```

Note

El grupo de usuarios está ahora listo para proporcionar y autenticar tokens de acceso. En este ejemplo, la URL del servidor de autorización es `https://cognito-idp.us-east-1.amazonaws.com/us-east-1_QLEXAMPLE/.well-known/jwks.json`.

6. Pruebe la configuración.

```
$ curl -H "Authorization: Basic `echo -
n 219273hp6k2ut5cugg9EXAMPLE:1vp5e8nec7cbf4m9me55mbmht91u61h1h0a78rq1qki11EXAMPLE
| 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"
```

Ejemplo de salida

```
{
```

```
"access_token": "eyJraWQiOiJGQ0VaRFpJUUpT3NSaW41MmtqaDdEbTZYb0RnSTQ5b2VUT0cxUU11Q2VJPSIsImF0IjoiZm9udjTlZGScR0dZtId5dThkpEZiSx0YwiiWe9crAlqoazlDcCsUJHIXDtGKW64pSj3-uQQGg1Jv_tyVjhrA4JbD0k67WS2V9NW-uZ7t4zwwaUm0i3KzpBmi54fpVgPaewiVlUm_aS4LUFcWT6hVJjiZF7om7984qb2g0a14iZxpXPBJTZX_gtG9EtvnS9U",
"expires_in": 3600,
"token_type": "Bearer"
}
```

7. Registre el servidor de autorización externo para utilizarlo con el broker mediante el comando [register-auth-server](#).

```
$ 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
```

Los desarrolladores pueden ahora utilizar el servidor para solicitar tokens de acceso. Cuando solicite tokens de acceso, proporcione el ID del cliente, el secreto del cliente y la URL del servidor generados aquí. Para obtener más información sobre la solicitud de tokens de acceso, consulte [Get an access token and make an API request](#) en la Guía para desarrolladores de NICE DCV Session Manager.

Configurar la persistencia del broker

Los brokers de Session Manager admiten la integración con bases de datos externas. La base de datos externa permite a Session Manager conservar los datos de estado y las claves para que estén disponibles posteriormente. De hecho, los datos del broker se distribuyen por todo el clúster, lo que los hace susceptibles a la pérdida de datos si un host necesita reiniciarse o si se termina un clúster. Con esta característica habilitada, puede añadir y eliminar nodos de broker. Además, puede detener un clúster y reiniciarlo sin necesidad de regenerar claves ni perder información sobre qué servidor NICE DCV está abierto o cerrado.

Se pueden configurar los siguientes tipos de información para que persistan:

- Claves para configurar las sesiones a fin de establecer conexión con los clientes
- Datos de las sesiones en vuelo
- Estado del servidor NICE DCV

NICE DCV Session Manager es compatible con bases de datos DynamoDB, MariaDB y MySQL. Debe configurar y administrar una de estas bases de datos para utilizar esta característica. Si sus

máquinas de broker están alojadas en Amazon EC2, le recomendamos que utilice DynamoDB como base de datos externa, ya que no requiere configuración adicional.

Note

Puede incurrir en costes adicionales al ejecutar una base de datos externa. Para obtener información sobre precios de DynamoDB, consulte [Precios de la capacidad aprovisionada](#).

Configurar el broker para que persista en DynamoDB

Configure los brokers para que comiencen a almacenar sus datos en DynamoDB:

1. Abra `/etc/dcv-session-manager-broker/session-manager-broker.properties` con su editor de texto preferido y haga lo siguiente:
 - Establezca `enable-persistence = true`
 - Establezca `persistence-db = dynamodb`
 - Para `dynamodb-region` especifique la región de `&aws;` en la que desea almacenar las tablas que contienen los datos del broker. Para ver una lista de las regiones admitidas, consulte [Puntos de conexión del servicio de DynamoDB](#).
 - Para `dynamodb-table-rcu` especifique la cantidad de unidades de capacidad de lectura (RCU) que admite cada tabla. Para obtener más información sobre RCU, consulte [Capacidad aprovisionada de DynamoDB](#).
 - Para `dynamodb-table-wcu` especifique la cantidad de unidades de capacidad de escritura (RCU) que admite cada tabla. Para obtener más información sobre WCU, consulte [Capacidad aprovisionada de DynamoDB](#).
 - Para `dynamodb-table-name-prefix`, especifique el prefijo que se añade a cada tabla de DynamoDB (resulta útil para distinguir distintos clústeres de brokers que utilizan la misma cuenta de). Solo se permiten caracteres alfanuméricos, punto, guion y guion bajo.
2. Detenga todos los brokers en el clúster. Para cada broker, ejecute el siguiente comando:

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

3. Asegúrese de que todos los brokers del clúster estén detenidos y, a continuación, reinícelos todos. Inicie cada broker ejecutando el siguiente comando:


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

El host del broker debe tener permiso para llamar a las API de DynamoDB. En instancias de Amazon EC2, las credenciales se recuperan automáticamente a través del servicio de metadatos de Amazon EC2. Si necesita especificar credenciales diferentes, puede configurarlas mediante una de las técnicas de recuperación de credenciales compatibles (como las propiedades del sistema Java o las variables de entorno). Para obtener más información, consulte [Suministro y recuperación de credenciales de &aws;](#).

Configurar el broker para que persista en MariaDB/MySQL

Note

El archivo `/etc/dcv-session-manager-broker/session-manager-broker.properties` contiene información confidencial. De forma predeterminada, el acceso de escritura está limitado a la raíz y el acceso de lectura está limitado a la raíz y al usuario que ejecuta el broker. De forma predeterminada, este es el usuario `dcvsmbroker`. El broker comprueba al inicio que el archivo tiene los permisos esperados.

Configure los brokers para que comiencen a conservar sus datos en MariaDB/MySQL:

1. Abra `/etc/dcv-session-manager-broker/session-manager-broker.properties` con su editor de texto preferido y haga lo siguiente:
 - Establezca `enable-persistence = true`
 - Establezca `persistence-db = mysql`
 - Establezca `jdbc-connection-url = jdbc:mysql://<db_endpoint>:<db_port>/<db_name>?createDatabaseIfNotExist=true`

En esta configuración, `<db_endpoint>` es el punto de conexión de la base de datos, `<db_port>` es el puerto de la base de datos y `<db_name>` es el nombre de la base de datos.

- Para `jdbc-user` especifique el nombre del usuario que tiene acceso a la base de datos.
- Para `jdbc-password` especifique la contraseña del usuario que tiene acceso a la base de datos.

2. Detenga todos los brokers en el clúster. Para cada broker, ejecute el siguiente comando:

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

3. Asegúrese de que todos los brokers del clúster estén detenidos y, a continuación, reinícelos todos. Para cada broker, ejecute el siguiente comando:

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

Integración con la puerta de enlace de conexión NICE DCV

La [puerta de enlace de conexión NICE DCV](#) es un paquete de software instalable que permite a los usuarios acceder a una flota de servidores NICE DCV a través de un único punto de acceso a LAN o VPC.

Si su infraestructura incluye servidores NICE DCV a los que se puede acceder a través de la puerta de enlace de conexión NICE DCV, puede configurar Session Manager para integrar la puerta de enlace de conexión NICE DCV. Siguiendo los pasos que se describen en la siguiente sección, el broker actuará como [Solucionador de sesiones](#) para la puerta de enlace de conexión. Es decir, el broker expondrá un punto de conexión HTTP adicional. La puerta de enlace de conexión realizará llamadas API al punto de conexión para recuperar la información necesaria para enrutar las conexiones NICE DCV al host seleccionado por el broker.

Configurar el broker de Session Manager como Solucionador de sesiones para la puerta de enlace NICE DCV

Lado del broker de Session Manager

1. Abra `/etc/dcv-session-manager-broker/session-manager-broker.properties` con el editor de texto que prefiera y aplique los cambios siguientes:
 - Establezca `enable-gateway = true`
 - Establezca `gateway-to-broker-connector-https-port` en un puerto TCP libre (el predeterminado es 8447)
 - Establezca `gateway-to-broker-connector-bind-host` en la dirección IP del host al que se va a vincular el broker para las conexiones de puerta de enlace de conexión NICE DCV (el valor predeterminado es 0.0.0.0)

2. A continuación, ejecute los siguientes comandos para detener y reiniciar el broker:

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

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

3. Obtenga una copia del certificado autofirmado del broker y colóquelo en su directorio de usuario.

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

La necesitará cuando instale la puerta de enlace de conexión NICE DCV en el siguiente paso.

Lado de la puerta de enlace de conexión NICE DCV

- Siga la [sección](#) de la documentación sobre la puerta de enlace de conexión NICE DCV.

Dado que la puerta de enlace de conexión NICE DCV realiza llamadas API HTTP al broker, si el broker utiliza un certificado autofirmado, tendrá que copiar el certificado de broker en el host de la puerta de enlace de conexión NICE DCV (recuperado en el paso anterior) y establecer el parámetro `ca-file` en la sección `[resolver]` de la configuración de la puerta de enlace de conexión NICE DCV.

Opcional: habilitar autenticación de cliente TLS

Una vez completado el paso anterior, Session Manager y la puerta de enlace de conexión pueden comunicarse a través de un canal seguro, en el que la puerta de enlace de conexión puede verificar la identidad de los brokers de Session Manager. Si necesita que los brokers de Session Manager también validen la identidad de la puerta de enlace de conexión antes de establecer el canal seguro, debe habilitar la característica de autenticación de cliente TLS siguiendo los pasos de la siguiente sección.

Note

Si Session Manager utiliza un equilibrador de carga, la autenticación de cliente TLS no se puede habilitar con equilibradores de carga que tengan una terminación de conexión TLS, como los equilibradores de carga de aplicación (ALB) o los equilibradores de carga de puerta de enlace (GLB). Solo se admiten los equilibradores de carga sin terminación TLS, como los

equilibradores de carga de red (NLB). Si utiliza ALB o GLB, puede forzar que solo grupos de seguridad específicos puedan contactar con los equilibradores de carga, lo que garantiza un nivel de seguridad adicional; encontrará más información sobre grupos de seguridad aquí:

[Security groups for your VPC](#)

Lado del broker de Session Manager

1. Para habilitar la autenticación del cliente TLS para la comunicación entre los brokers de Session Manager y la puerta de enlace de conexión NICE DCV, siga estos pasos:
2. Genere las claves y los certificados necesarios ejecutando: la salida del comando le indicará la carpeta en la que se generaron las credenciales y la contraseña utilizada para crear el archivo TrustStore.

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

3. Coloque una copia de la clave privada y del certificado autofirmado de la puerta de enlace de conexión NICE DCV en su directorio de usuario. La necesitará cuando instale la puerta de enlace de conexión NICE DCV en el siguiente paso.

```
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. A continuación, abra `/etc/dcv-session-manager-broker/session-manager-broker.properties` en el editor de texto que prefiera y haga lo siguiente:

- Establezca `enable-tls-client-auth-gateway` en `true`
- Establezca `gateway-to-broker-connector-trust-store-file` en la ruta del archivo TrustStore creado en el paso anterior
- Establezca `gateway-to-broker-connector-trust-store-pass` en la contraseña utilizada para crear el archivo TrustStore en el paso anterior

5. A continuación, ejecute el siguiente comando para detener y reiniciar el broker:

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

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

Lado de la puerta de enlace de conexión NICE DCV

- Siga la [sección](#) de la documentación sobre la puerta de enlace de conexión NICE DCV.
 - Utilice la ruta completa del archivo de certificado que copió en el paso anterior al establecer el parámetro `cert-file` en la sección `[resolver]`
 - Utilice la ruta completa del archivo de clave que copió en el paso anterior cuando al establecer el parámetro `cert-key-file` en la sección `[resolver]`

Servidor NICE DCV Session Manager: asignación de DNS

La puerta de enlace de conexión NICE DCV requiere los nombres DNS de los servidores NICE DCV para conectarse a las instancias del servidor DCV. En esta sección se muestra cómo se puede definir un archivo JSON que contenga la asignación entre cada servidor DCV y su nombre DNS asociado.

Estructura de archivos

La asignación consiste en una lista de objetos JSON con los siguientes campos:

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

Donde:

ServerIdType :

Identifica el tipo de identificador al que hace referencia el valor; los valores disponibles actualmente son `ipAddress`, `agentServerId` e `instanceId`:

Ip:

Disponible tanto para Amazon EC2 como para infraestructuras locales; los administradores del sistema pueden recuperarlo rápidamente con un comando `ifconfig` (Linux) o `ipconfig` (Windows). Esta información también está disponible en la respuesta de la API `DescribeServers`.

Id:

Disponible tanto para Amazon EC2 como para infraestructuras locales; el agente de Session Manager crea un nuevo UUID cada vez que cambia el nombre de host o la dirección IP. Esta información está disponible en la respuesta de la API `DescribeServers`.

Host.Aws.Ec2InstanceId:

Disponible solo para instancias de Amazon EC2, identifica de forma exclusiva una máquina; no cambia tras el reinicio de la instancia. Se puede recuperar en el host estableciendo contacto con `http://169.254.169.254/latest/meta-data/instance-id`. Esta información también está disponible en la respuesta de la API `DescribeServers`.

ServerId:

Un identificador del tipo especificado que identifica de forma exclusiva a cada servidor NICE DCV de la red.

DnsNames :

El objeto que contiene los nombres DNS asociados al servidor NICE DCV. Este objeto contendrá:

InternalDnsNames :

El nombre DNS que utiliza la puerta de enlace de conexión NICE DCV para conectarse a la instancia.

Utilice los comandos `register-server-dns-mapping` de la CLI del broker de Session Manager para cargar la asignación desde un archivo (referencia de la página de comandos: [register-server-dns-mapping](#)) y `describe-server-dns-mappings` para enumerar las asignaciones cargadas actualmente en el broker de Session Manager (referencia de la página de comandos: [describe-server-dns-mappings](#)).

Persistencia

Le recomendamos encarecidamente que habilite la característica de persistencia del broker de Session Manager para evitar la pérdida de asignación en caso de que varios brokers o todo el clúster dejen de funcionar. Para obtener más información sobre cómo habilitar la persistencia de datos, consulte [Configure Broker Persistence](#)

Integración de Amazon CloudWatch

Session Manager admite la integración con Amazon CloudWatch para brokers que se ejecutan en instancias de Amazon EC2 y también para brokers que se ejecutan en hosts locales.

Amazon CloudWatch monitorea los recursos y las aplicaciones de Amazon Web Services (AWS) que ejecuta en AWS en tiempo real. Puede utilizar CloudWatch para recopilar y hacer un seguimiento de métricas, que son las variables que puede medir en los recursos y aplicaciones. Para obtener más información, consulte la [Guía del usuario de Amazon CloudWatch](#).


Puede configurar el broker de Session Manager para que envíe los siguientes datos de métricas a Amazon CloudWatch:

- `Number of DCV servers`: número de servidores DCV administrados por el broker.
- `Number of ready DCV servers`: número de servidores DCV que se encuentran en estado READY administrados por el broker.
- `Number of DCV sessions`: número de sesiones de DCV gestionadas por el broker.
- `Number of DCV console sessions`: número de sesiones de consola DCV gestionadas por el broker.
- `Number of DCV virtual sessions`: número de sesiones virtuales de DCV gestionadas por el broker.
- `Heap memory used`: cantidad de memoria dinámica que utiliza el broker.
- `Off-heap memory used`: cantidad de memoria no dinámica que utiliza el broker.
- `Describe sessions request time`: tiempo que se tarda en completar las solicitudes a la API `DescribeSessions`.
- `Delete sessions request time`: tiempo que se tarda en completar las solicitudes a la API `DeleteSessions`.
- `Create sessions request time`: tiempo que se tarda en completar las solicitudes a la API `CreateSessions`.

- `Get session connection data request time`: tiempo que se tarda en completar las solicitudes de la API `GetSessionConnectionData`.
- `Update session permissions request time`: tiempo que se tarda en completar las solicitudes de la API `UpdateSessionPermissions`.

Para configurar el broker para que envíe datos de métricas a Amazon CloudWatch

1. Abra `/etc/dcv-session-manager-broker/session-manager-broker.properties` con su editor de texto preferido y haga lo siguiente.
 - Establezca `enable-cloud-watch-metrics` en `true`
 - Para `cloud-watch-region`, especifique la región en la que desea recopilar los datos de las métricas.

 Note

Si su broker se ejecuta en una instancia de Amazon EC2, este parámetro es opcional. La región se recupera automáticamente del servicio de metadatos de instancias (IMDS). Si ejecuta el broker en un host local, este parámetro es obligatorio.

2. Detenga y reinicie el broker.

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

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

El host del broker también debe tener permiso para llamar a la API `cloudwatch:PutMetricData`. Las credenciales de AWS se pueden recuperar mediante una de las técnicas de recuperación de credenciales compatibles. Para obtener más información, consulte [Suministro y recuperación de credenciales de AWS](#).

Actualización de NICE DCV Session Manager

En el siguiente tema se describe cómo actualizar Session Manager.

Note

Le recomendamos encarecidamente que actualice todos los agentes de Session Manager antes de actualizar los brokers de Session Manager para evitar problemas de incompatibilidad en el caso de que se introduzcan nuevas características.

Temas

- [Actualizar el agente de NICE DCV Session Manager](#)
- [Actualizar el broker de NICE DCV Session Manager](#)

Actualizar el agente de NICE DCV Session Manager

Linux host

Note

Siga estas instrucciones para instalar el agente en hosts x86 de 64 bits. Para instalar el agente en hosts ARM de 64 bits, para Amazon Linux, RHEL y CentOS, sustituya *x86_64* por *aarch64* y, para Ubuntu, sustituya *amd64* por *arm64*.

Para actualizar el agente en un host Linux

1. Ejecute el siguiente comando para detener el agente.

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

2. Descargue el paquete de instalación.
 - Amazon Linux 2, RHEL 7.x y 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 y 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. Instale el paquete.

- Amazon Linux 2, RHEL 7.x y 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 y 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. Para iniciar el agente, ejecute el comando siguiente.

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

Windows host

Para instalar el agente en un host Windows

1. Detenga el servicio del agente. En el símbolo del sistema, ejecute los siguientes comandos.

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

2. Descargue el [instalador del agente](#).
3. Ejecute el instalador. En la pantalla de bienvenida, elija Next.
4. En la pantalla del EULA, lea atentamente el contrato de licencia y, si está de acuerdo, seleccione Acepto los términos y condiciones y elija Siguiente.
5. Elija Instalar para comenzar la instalación.
6. Reinicie el servicio del agente. En el símbolo del sistema, ejecute los siguientes comandos.

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

Actualizar el broker de NICE DCV Session Manager

Para actualizar el broker

1. Conéctese al host en el que desea actualizar el broker.
2. Detenga el servicio del broker.

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

3. Descargue el paquete de instalación.

- Amazon Linux 2, RHEL 7.x y 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 y 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. Instale el paquete.

- Amazon Linux 2, RHEL 7.x y 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 y 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. Inicie el servicio del broker y asegúrese de que se inicie de forma automática cuando se inicia la instancia.

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

Referencia de la CLI del broker

En esta sección, se explica cómo utilizar los comandos de la interfaz de la línea de comandos (CLI) del broker.

Utilice los siguientes comandos si utiliza un servidor de autenticación externo para generar tokens de acceso de OAuth 2.0:

- [register-auth-server](#)
- [list-auth-servers](#)
- [unregister-auth-server](#)

Utilice los siguientes comandos si utiliza el broker de Session Manager como servidor de autenticación de OAuth 2.0.

- [register-api-client](#)
- [describe-api-clients](#)
- [unregister-api-client](#)
- [renew-auth-server-api-key](#)

Utilice los siguientes comandos para administrar el agente de Session Manager.

- [generate-software-statement](#)
- [describe-software-statements](#)
- [deactivate-software-statement](#)
- [describe-agent-clients](#)
- [unregister-agent-client](#)

Utilice los siguientes comandos para administrar el archivo de asignación de nombres de servidor DNS - DCV.

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

register-auth-server

Registra un servidor de autenticación externo para su uso con el broker.

De forma predeterminada, Session Manager usa el broker como servidor de autenticación para generar tokens de acceso de OAuth 2.0. Si utiliza el broker como servidor de autenticación, no se requiere configuración adicional.

Sin embargo, si decide utilizar un servidor de autenticación externo, como Active Directory o Amazon Cognito, debe utilizar este comando para registrar el servidor de autenticación externo.

Temas

- [Sintaxis](#)
- [Opciones](#)
- [Ejemplo](#)

Sintaxis

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

Opciones

--url

URL del servidor de autenticación externo que se va a utilizar. Debe añadir `.well-known/jwks.json` a la URL del servidor de autenticación.

Tipo: String

Obligatorio: sí

Ejemplo

El siguiente ejemplo registra un servidor de autenticación externo con una URL de `https://my-auth-server.com/`.

Comando

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

Salida

```
Jwk url registered.
```

list-auth-servers

Muestra los servidores de autenticación externos que se han registrado.

Temas

- [Sintaxis](#)
- [Output](#)
- [Ejemplo](#)

Sintaxis

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

Output

Urls

URL de los servidores de autenticación externos registrados.

Ejemplo

El ejemplo siguiente muestra todos los servidores de autenticación externos que se han registrado.

Comando

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

Salida


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

unregister-auth-server

Anula el registro de un servidor de autenticación externo. Después de anular el registro de un servidor de autenticación externo, ya no se podrá utilizar para generar tokens de acceso de OAuth 2.0.

Temas

- [Sintaxis](#)
- [Opciones](#)
- [Output](#)
- [Ejemplo](#)

Sintaxis

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

Opciones

--url

URL del servidor de autenticación externo cuyo registro se va a anular. Debe añadir `.well-known/jwks.json` a la URL del servidor de autenticación.

Tipo: String

Obligatorio: sí

Output

Url

URL del servidor de autenticación externo cuyo registro se va a anular.

Ejemplo

El siguiente ejemplo registra un servidor de autenticación externo con una URL de `https://my-auth-server.com/`.

Comando

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

Salida

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

register-api-client

Registra un cliente de Session Manager en el broker y genera credenciales de cliente que el este puede utilizar para recuperar un token de acceso de OAuth 2.0, necesario para realizar solicitudes a la API.

Important

Asegúrese de guardar las credenciales en un lugar seguro. No se pueden recuperar más tarde.

Este comando solo se utiliza si el broker se utiliza como servidor de autenticación de OAuth 2.0.

Temas

- [Sintaxis](#)
- [Opciones](#)
- [Output](#)
- [Ejemplo](#)

Sintaxis

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

Opciones

--name

Nombre único que se utiliza para identificar al cliente de Session Manager.

Tipo: String

Obligatorio: sí

Output

client-id

ID de cliente único que utilizará el cliente de Session Manager para recuperar un token de acceso de OAuth 2.0.

client-password

Contraseña que utilizará el cliente de Session Manager para recuperar un token de acceso de OAuth 2.0.

Ejemplo

El siguiente ejemplo registra un cliente llamado `my-sm-client`.

Comando

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

Salida

```
client-id: 21cfe9cf-61d7-4c53-b1b6-cf248EXAMPLE  
client-password: NjVmZDR1N2ItNjNmYS00M2QxLWF1ZmMtZmNmMDNkMEXAMPLE
```

describe-api-clients

Muestra los clientes de Session Manager que se han registrado con el broker.

Temas

- [Sintaxis](#)
- [Output](#)
- [Ejemplo](#)

Sintaxis

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

Output

name

Nombre único del cliente de Session Manager.

id

ID exclusivo del cliente de Session Manager.

active

Indica el estado del cliente de Session Manager. Si el cliente está activo, el valor es `true`; de lo contrario, es `false`.

Ejemplo

En el siguiente ejemplo se enumeran los clientes de Session Manager registrados.

Comando

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

Salida

```
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

Desactiva un cliente de Session Manager registrado. Un cliente de Session Manager desactivado ya no puede utilizar sus credenciales para recuperar los tokens de acceso de OAuth 2.0.

Temas

- [Sintaxis](#)
- [Opciones](#)
- [Ejemplo](#)

Sintaxis

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

Opciones

--client -id

ID del cliente de Session Manager que se va a desactivar.

Tipo: String

Obligatorio: sí

Ejemplo

El siguiente ejemplo desactiva un cliente de Session Manager con un ID de cliente de f855b54b-40d4-4769-b792-b727bEXAMPLE.

Comando

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

Salida

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

renew-auth-server-api-key

Renueva las claves públicas y privadas utilizadas por el broker para firmar los tokens de acceso de OAuth 2.0 que se ofrecen al cliente de Session Manager. Si renueva las claves, debe proporcionar la nueva clave privada al desarrollador, ya que es necesaria para realizar solicitudes a la API.

Temas

- [Sintaxis](#)
- [Ejemplo](#)

Sintaxis

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

Ejemplo

En el siguiente ejemplo se renuevan las claves pública y privada.

Comando

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

Salida

```
Keys renewed.
```

generate-software-statement

Genera una instrucción de software.

Los agentes deben estar registrados en el broker para habilitar la comunicación. Los agentes necesitan una instrucción de software para poder registrarse en el broker. Cuando el agente

disponga de una instrucción de software podrá registrarse automáticamente a sí mismo en el broker mediante el [Protocolo de registro dinámico de clientes OAuth 2.0](#). Cuando el agente se registra en el broker, recibe un identificador de cliente y un secreto de cliente que utiliza para autenticarse en el broker.

El broker y el agente reciben y utilizan una instrucción de software predeterminada cuando se instalan por primera vez. Puede seguir utilizando la instrucción de software predeterminada o generar una nueva. Si genera una nueva instrucción de software, debe colocarla en un nuevo archivo del agente y, a continuación, añadir la ruta de archivo al parámetro `agent.software_statement_path` del archivo `agent.conf`. Cuando termine, detenga y reinicie el agente para que pueda utilizar la nueva instrucción de software y registrarse en el broker.

Temas

- [Sintaxis](#)
- [Output](#)
- [Ejemplo](#)

Sintaxis

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

Output

software-statement

Instrucción de software.

Ejemplo

El siguiente ejemplo genera una instrucción de software.

Comando

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

Salida

```
software-statement:
```

```
ewogICJpZCIgOiAiYjc1NTVhN2Q0tNWI0MC00OTJhLWJjOTU0tNmUzOWhhYzkyMDcxIiwKICAiYWN0aXZlIiA6IHRydWUsc3I
```

describe-software-statements

Describe las instrucciones de software existentes.

Temas

- [Sintaxis](#)
- [Output](#)
- [Ejemplo](#)

Sintaxis

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

Output

software-statement

Instrucción de software.

issued-at

Fecha y hora en que se generó el software.

is-active

Estado actual de la instrucción de software. `true` si la instrucción de software está activa; de lo contrario, es `false`.

Ejemplo

El siguiente ejemplo genera una instrucción de software.

Comando

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


Ejemplo

El siguiente ejemplo desactiva una instrucción de software.

Comando

```
sudo -u root dcv-session-manager-broker deactivate-software-statement --software-statement EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWI0MC00TJhLWJj0TUtNmUz0WNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj
```

Salida

```
Software statement  
EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWI0MC00TJhLWJj0TUtNmUz0WNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj  
deactivated
```

describe-agent-clients

Describe los agentes que están registrados en el broker.

Temas

- [Sintaxis](#)
- [Output](#)
- [Ejemplo](#)

Sintaxis

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

Output

name

Nombre del agente

id

ID exclusivo del agente.

active

Estado del agente. `true` si el agente está activo; de lo contrario, es `false`.

Ejemplo

En el siguiente ejemplo se describen los agentes.

Comando

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

Salida

```
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

Anula el registro de un agente del broker.

Temas

- [Sintaxis](#)
- [Opciones](#)

- [Ejemplo](#)

Sintaxis

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

Opciones

--client-id

ID del agente cuyo registro se va a anular.

Tipo: String

Obligatorio: sí

Ejemplo

El siguiente ejemplo anula el registro de un agente.

Comando

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

Salida

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

register-server-dns-mappings

Registra las asignaciones de servidores DCV - nombres DNS procedentes de un archivo JSON.

Sintaxis

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

Opciones

--file-path

Ruta del archivo que contiene las asignaciones de servidores DCV - nombres DNS.

Tipo: String

Obligatorio: sí

Ejemplo

El siguiente ejemplo registra los las asignaciones de servidores DCV - nombres DNS del archivo /tmp/mappings.json.

Comando

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

Salida

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

describe-server-dns-mappings

Describe las asignaciones de servidores DCV - nombres DNS disponibles actualmente.

Sintaxis

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

Output

serverIdType

Tipo de identificador del servidor.

serverId

ID exclusivo del servidor.

dnsNames

Nombres DNS internos y externos

internalDnsNames

Nombres DNS internos

externalDnsNames

Nombres DNS externos

Ejemplo

En el siguiente ejemplo se muestran las asignaciones de servidores DCV - nombres DNS registradas.

Comando

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

Salida

```
[
{
  "serverIdType" : "Id",
  "serverId" : "192.168.0.1",
  "dnsNames" : {
    "internalDnsName" : "internal1",
    "externalDnsName" : "external1"
  }
},
{
  "serverIdType" : "Host.Aws.Ec2InstanceId",
  "serverId" : "i-0648aee30bc78bdff",
  "dnsNames" : {
    "internalDnsName" : "internal2",
    "externalDnsName" : "external2"
  }
}
```

```
}  
]
```

Referencia de archivos de configuración

Esta guía proporciona información sobre los archivos de configuración de agentes y brokers.

Temas

- [Archivo de configuración del broker](#)
- [Archivo de configuración del agente](#)

Archivo de configuración del broker

El archivo de configuración del broker (`/etc/dcv-session-manager-broker/session-manager-broker.properties`) incluye parámetros que se pueden configurar para personalizar la funcionalidad de Session Manager. Puede editar el archivo de configuración con el editor de texto que prefiera.

Note

El archivo `/etc/dcv-session-manager-broker/session-manager-broker.properties` contiene información confidencial. De forma predeterminada, el acceso de escritura está limitado a la raíz y el acceso de lectura está limitado a la raíz y al usuario que ejecuta el broker. De forma predeterminada, este es el usuario `dcvsmbroker`. El broker comprueba al inicio que el archivo tiene los permisos esperados.

En la siguiente tabla se muestran los parámetros del archivo de configuración del broker.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
<code>broker.java.home</code>	No		Especifica la ruta del directorio principal de Java que utilizará el broker en lugar del directorio o predeterminado del sistema. Si está configura

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
			<p>do, el broker gente utilizará <code><broker-java-home>/bin/java</code> al inicio.</p> <p>Sugerecia: el broker requiere Java Runtime Environment 11 y está instalado si no se encuentra como dependencia tras una instalación correcta. Si la versión 11 no está configurada como entorno Java predeterminado, se puede capturar su directori o principal mediante el siguiente comando:</p> <pre>\$ sudo alternatives --display java</pre>
sessionscreenshots-max-width	No	160	Especifica el ancho máximo, en píxeles, de las capturas de pantalla de las sesiones realizadas con la API <code>GetSessionScreenshots</code> .

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
session-screenshots-max-height	No	100	Especifica el alto máximo, en píxeles, de las capturas de pantalla de las sesiones realizadas con la API GetSessionScreenshots.
session-screenshots-format	No	png	Formato de archivo de imagen de las capturas de pantalla de las sesiones realizadas con la API GetSessionScreenshots.
create-sessions-queue-max-size	No	1000	Número máximo de solicitudes de la API CreateSessions no atendidas que se pueden poner en cola. Cuando la cola está llena, se rechazan las nuevas solicitudes no atendidas.
create-sessions-queue-max-time-seconds	No	1800	Tiempo máximo, en segundos, que puede permanecer en la cola una solicitud de API CreateSessions no atendida. Si la solicitud no se puede atender en el período de tiempo especificado, se produce un error.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
session-manager-working-path	Sí	/tmp	Especifica la ruta del directorio en el que el broker escribe los archivos necesarios para funcionar. Solo el broker debe poder acceder a este directorio.
enable-authorization-server	Sí	true	Especifica si el broker es el servidor de autenticación que se utiliza para generar los tokens de acceso de OAuth 2.0 para las API de los clientes.
enable-authorization	Sí	true	Habilita o deshabilita la autorización del cliente. Si habilita la autorización del cliente, la API del cliente debe proporcionar un token de acceso al realizar las solicitudes de la API. Si deshabilita la autorización del cliente, las API del cliente pueden realizar solicitudes sin tokens de acceso.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
enable-agent-authorization	Sí	true	Habilita o deshabilita la autorización del agente. Si habilita la autorización del agente, este debe proporcionar un token de acceso cuando se comunique con el broker.
delete-session-duration-hours	No	1	Especifica el número de horas después de las cuales las sesiones eliminadas pasan a ser invisibles y las llamadas a la API DescribeSession ya no las devuelven.
connect-session-token-duration-minutes	No	60	Especifica el número de minutos durante los cuales sigue siendo válido el token ConnectSession.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
client-to-broker-connect-https-port	Sí	8443	Especifica el puerto HTTPS en el que el broker escucha las conexiones de clientes.
client-to-broker-connect-bind-host	No	0.0.0.0	Especifica la dirección IP del host a la que se enlaza el broker para las conexiones de clientes.
client-to-broker-connect-key-store-file	Sí		Especifica el almacén de claves utilizado para las conexiones de clientes TLS.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
client-to-broker-connect-key-store-pass	Sí		Especifica la contraseña del almacén de claves.
agent-to-broker-connect-https-port	Sí	8445	Especifica el puerto HTTPS en el que el broker escucha las conexiones de agentes.
agent-to-broker-connect-bind-host	No	0.0.0.0	Especifica la dirección IP del host a la que se enlaza el broker para las conexiones de agentes.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
agent-to-broker-connectokey-store-file	Sí		Especifica el almacén de claves utilizado para las conexiones de agentes TLS.
agent-to-broker-connectokey-store-pass	Sí		Especifica la contraseña del almacén de claves.
broker-to-broker-port	Sí	47100	Especifica el puerto utilizado para conexiones de broker a broker.
broker-to-broker-bind-host	No	0.0.0.0	Especifica la dirección IP del host a la que se enlaza el broker para conexiones de broker a broker.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
broker-to-broker-discovery-port	Sí	47500	Especifica el puerto que utilizan los brokers para detectarse entre sí.
broker-to-broker-discovery-address	No		Especifica las direcciones IP y los puertos de los demás brokers de la flota con el formato <i>dirección_ip :puerto</i> . Si hay varios brokers, separe los valores con una coma. Si especifica broker-to-broker-discovery-multicast-group , broker-to-broker-discovery-multicast-port , broker-to-broker-discovery-AWS-region o broker-to-broker-discovery-AWS-alb-target-group-arn , omite este parámetro.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
broker-to-broker-discovery-multicast-group	No		Especifica el grupo de multidifusión para la detección de broker a broker. Si especifica <code>broker-to-broker-discovery-addresses</code> , <code>broker-to-broker-discovery-aws-region</code> o <code>broker-to-broker-discovery-AWS-alb-target-group-arn</code> , omite este parámetro.
broker-to-broker-discovery-multicast-port	No		Especifica el puerto de multidifusión para la detección de broker a broker. Si especifica <code>broker-to-broker-discovery-addresses</code> , <code>broker-to-broker-discovery-AWS-region</code> o <code>broker-to-broker-discovery-AWS-alb-target-group-arn</code> , omite este parámetro.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
broker-to-broker-discovery-AWS-region	No		Especifica la región de AWS del equilibrador de carga de aplicación que se utiliza para la detección de broker to broker. Si especifica <code>broker-to-broker-discovery-multicast-group</code> , <code>broker-to-broker-discovery-multicast-port</code> o <code>broker-to-broker-discovery-addresses</code> , omite este parámetro.
broker-to-broker-discovery-AWS-alb-target-group-arn	No		El ARN del usuario del grupo de destino del equilibrador de carga de aplicación para detección de broker a broker. Si especifica <code>broker-to-broker-discovery-multicast-group</code> , <code>broker-to-broker-discovery-multicast-port</code> o <code>broker-to-broker-discovery-addresses</code> , omite este parámetro.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
broker-to-broker-distributed-memory-max-size-mb	No	4096	Especifica la cantidad máxima de memoria fuera del montón que utilizará un broker para almacenar datos de sesión de NICE DCV.
broker-to-broker-key-store-file	Sí		Especifica el almacén de claves utilizado para las conexiones de agentes TLS.
broker-to-broker-key-store-pass	Sí		Especifica la contraseña del almacén de claves.
enable-cloud-watch-metrics	No	false	Habilita o deshabilita las métricas de Amazon CloudWatch. Si habilita las métricas de CloudWatch quizá deba especificar un valor para <code>cloud-watch-region</code> .

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
cloud-watch-region	No	Solo es obligatorio si <code>enable-cloud-watch-metrics</code> se establece en <code>true</code> . Si el broker está instalado en una instancia de Amazon EC2, la región se recupera del IMDS.	La región de AWS en la que se publican las métricas de CloudWatch.
max-api-requests-per-second	No	1000	Especifica el número máximo de solicitudes que la API de broker puede procesar cada segundo antes de que se limite.
enable-rottlir-forwarder	No	false	Si se establece <code>true</code> , la limitación recupera el IP del intermediario del encabezado X-Forwarded-For, si está presente.
create-session-number-of-retries-on-failure	No	2	Especifica el número máximo de reintentos que se realizarán después de que falle una solicitud de creación de sesión en un host de servidor NICE DCV. Establézcalo en 0 para no realizar reintentos en caso de error.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
autorun-file-arguments-max-size	No	50	Especifica el número máximo de argumentos que se pueden pasar al archivo de ejecución automática.
autorun-file-arguments-max-argument-length	No	150	Especifica la longitud máxima en caracteres de cada argumento del archivo de ejecución automática.
enable-persistence	Sí	false	Si se establece en true, los datos de estado del broker se conservan en una base de datos externa.
persistence-db	No	Solo es obligatorio si enable-persistence se establece en true.	Especifica qué base de datos se utiliza para persistencia. Los únicos valores que se admiten son: dynamodb y mysql.
dynamodb-region	No	Solo es necesario si enable-persistence se establece en true y persistence-db se establece en dynamodb.	Especifica la región en la que se crean las tablas de DynamoDB y se accede a ellas.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
dynamoc table- rcu	No	Solo es necesario si enable-persistence se establece en true y persistence-db se establece en dynamodb.	Especifica las unidades de capacidad de lectura (RCU) de cada tabla de DynamoDB. Para obtener más información sobre las RCU, consulte Precios de la capacidad aprovisionada .
dynamoc table- wcu	No	Solo es necesario si enable-persistence se establece en true y persistence-db se establece en dynamodb.	Especifica las unidades de capacidad de escritura (WCU) de cada tabla de DynamoDB. Para obtener más información sobre las WCU, consulte Precios de la capacidad aprovisionada .
dynamoc table- nam e- prefix	No	Solo es necesario si enable-persistence se establece en true y persistence-db se establece en dynamodb.	Especifica el prefijo que se añade a cada tabla de DynamoDB (resulta útil para distinguir distintos clústeres de brokers que utilizan la misma cuenta de AWS). Solo se permiten caracteres alfanuméricos, punto, guion y guion bajo.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
jdbc-connection-url	No	Solo es necesario si <code>enable-persistence</code> se establece en <code>true</code> y <code>persistence-db</code> se establece en <code>mysql</code> .	<p>Especifica la URL de conexión con la base de datos MariaDB/MySQL; contiene el punto de conexión y el nombre de la base de datos. La URL debe tener este formato:</p> <pre data-bbox="1117 680 1507 919">jdbc:mysql://<db_endpoint>:<db_port>/<db_name>?createDatabaseIfNotExist=true</pre> <p>Donde <code><db_endpoint></code> es el punto de conexión de la base de datos MariaDB/MySQL, <code><db_port></code> es el puerto de la base de datos y <code><db_name></code> es el nombre de la base de datos.</p>
jdbc-user	No	Solo es necesario si <code>enable-persistence</code> se establece en <code>true</code> y <code>persistence-db</code> se establece en <code>mysql</code> .	Especifica el nombre del usuario que tiene acceso a la base de datos MariaDB/MySQL.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
jdbcpass word	No	Solo es necesario si <code>enable-persistence</code> se establece en <code>true</code> y <code>persistence-db</code> se establece en <code>mysql</code> .	Especifica la contraseña del usuario que tiene acceso a la base de datos MariaDB/MySQL.
secondsbefore- deleting- unreachable- dcv-server	No	1800	Especifica el número de segundos después de los cuales se elimina del sistema un servidor al que no se puede acceder.

Archivo de configuración del agente

El archivo de configuración del agente (`/etc/dcv-session-manager-agent/agent.conf` para Linux y `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` para Windows) incluye parámetros que se pueden configurar para personalizar la funcionalidad de Session Manager. Puede editar el archivo de configuración con el editor de texto que prefiera.

En la siguiente tabla se muestran los parámetros del archivo de configuración del agente.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
agent.ker_	Sí		Especifica el nombre de DNS del host del broker.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
<code>agent.tls_broker_port</code>	Sí	8445	Especifica el puerto a través del cual se va a comunicar con el broker.
<code>agent.tls_certificate_file</code>	No		Solo es necesario si <code>tls_strict</code> se establece en <code>true</code> . Especifica la ruta del archivo de certificado (.pem) necesario para validar el certificado TLS. Copie el certificado autofirmado del broker al agente.
<code>agent.tls_script_folder</code>	No	<ul style="list-style-type: none"> <code>/var/lib/dcv-session-manager-agent/init</code> (Linux) 	Especifica la ruta a una carpeta del servidor host que se utiliza para almacenar los scripts personalizados que permiten inicializar las sesiones del servidor NICE DCV cuando se crean. Debe especificar una ruta absoluta. Los usuarios que utilicen el parámetro de solicitud <code>InitFile</code> de la API <code>CreateSessions</code> deben poder acceder a la carpeta y ejecutar los archivos.
<code>agent.tls_strict</code>	No	<code>true</code>	Indica si se debe utilizar validación TLS estricta.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
agent_software_statement_path	No		Solo es necesario si no se utiliza la instrucción de software predeterminada. Especifica la ruta al archivo de instrucción de software. Para obtener más información, consulte generate-software-statement .
agent_software_statement_path	No	<ul style="list-style-type: none"> /etc/dcv-session-manager-agent (Linux) C:\Program Files\NICE\DCVSessionManagerAgent\conf\tags (Windows) 	Especifica la ruta de acceso a la carpeta en la que se encuentran los archivos de etiquetas. Para obtener más información, consulte Utilizar etiquetas para identificar servidores NICE DCV .

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
agent.authorun_file	No	<ul style="list-style-type: none"> • /var/lib/dcv-session-manager-agent/autorun (Linux) • C:\ProgramData\NICE\DcvSessionManagerAgent\autorun (Windows) 	<p>Especifica la ruta a una carpeta del servidor host que se utiliza para almacenar los scripts y las aplicaciones que se pueden ejecutar automáticamente en el inicio de la sesión. Debe especificar una ruta absoluta. Los usuarios que utilicen el parámetro de solicitud AutorunFile de la API CreateSessions deben poder acceder a la carpeta y ejecutar los archivos.</p>
agent._virtual_sessions	No	-1 (sin límite)	<p>Número máximo de sesiones virtuales que se pueden crear en un servidor NICE DCV mediante NICE DCV Session Manager.</p>
agent._concurrent_sessions_per_user	No	1	<p>Número máximo de sesiones virtuales que un solo usuario puede crear en un servidor NICE DCV mediante NICE DCV Session Manager.</p>

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
agent.t ker_upc e_inte l	No	30	Especifica el número de segundos que deben esperarse antes de enviar datos actualizados al broker. Los datos enviados incluyen el estado del servidor y el host NICE DCV, así como información actualizada de la sesión. Los valores más bajos permiten que Session Manager conozca mejor los cambios que se producen en el sistema en el que se ejecuta el agente, pero aumentan la carga del sistema y el tráfico de red. Los valores más altos reducen la carga del sistema y de la red, pero Session Manager es menos receptivo a los cambios del sistema, por lo que no se recomiendan valores superiores a 120.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
log.level	No	info	<p>Especifica el nivel de detalle de los archivos de registro. Están disponibles los niveles de detalle siguientes:</p> <ul style="list-style-type: none"> • error: proporciona el mínimo de detalle. Solo incluye errores. • warning: incluye errores y advertencias. • info: nivel de detalle predeterminado. Incluye errores, advertencias y mensajes de información. • debug: proporciona el máximo de detalle. Proporciona información detallada que resulta útil para depurar problemas.
log.directory	No	<ul style="list-style-type: none"> • /var/log/dcv-session-manager-agent/ (Linux) • C:\ProgramData\nice\DCVSessionManagerAgent\log (Windows) 	Especifica el directorio en el que se crean los archivos de registro.

Nombre del parámetro	Obligatorio	Valor predeterminado	Descripción
log.rotation	No	daily	<p>Especifica la rotación del archivo de registro. Los valores válidos son:</p> <ul style="list-style-type: none"> hourly: los archivos de registro rotan cada hora. daily: los archivos de registro rotan cada día.
log.maxlogfile-size	No	10485760	<p>Cuando el tamaño de un archivo de registro alcanza el tamaño especificado en bytes, rota. Se creará un nuevo archivo de registro y se colocarán más eventos de registro en el nuevo archivo.</p>
log.rotation-count	No	9	<p>El número máximo de archivos de registro que se conservan en la rotación. Cada vez que se produzca una rotación y se alcance este número, se eliminará el archivo de registro más antiguo.</p>

Notas de la versión e historial de revisión de NICE DCV Session Manager

Esta página recoge las notas de la versión y el historial de revisión de NICE DCV Session Manager.

Temas

- [Notas de la versión de NICE DCV Session Manager](#)
- [Historial de documentos](#)

Notas de la versión de NICE DCV Session Manager

Esta sección proporciona una descripción general de las principales actualizaciones, lanzamientos de características y correcciones de errores de NICE DCV Session Manager. Todas las actualizaciones están organizadas por fecha de lanzamiento. Actualizamos la documentación con frecuencia para dar respuesta a los comentarios que se nos envían.

Temas

- [2023.1— 9 de noviembre de 2023](#)
- [2023.0-15065 — 4 de mayo de 2023](#)
- [2023.0-14852— 28 de marzo de 2023](#)
- [2022.2-13907— 11 de noviembre de 2022](#)
- [2022.1-13067— 29 de junio de 2022](#)
- [2022.0-11952— 23 de febrero de 2022](#)
- [2021.3-11591— 20 de diciembre de 2021](#)
- [2021.2-11445— 18 de noviembre de 2021](#)
- [2021.2-11190— 11 de octubre de 2021](#)
- [2021.2-11042— 01 de septiembre de 2021](#)
- [2021.1-10557— 31 de mayo de 2021](#)
- [2021.0-10242— 12 de abril de 2021](#)
- [2020.2-9662— 04 de diciembre de 2020](#)
- [2020.2-9508— 11 de noviembre de 2020](#)

2023.1— 9 de noviembre de 2023

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 410• Agente: 732• CLI: 140	<ul style="list-style-type: none">• Correcciones de errores y mejoras de rendimiento.

2023.0-15065 — 4 de mayo de 2023

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 392• Agente: 675• CLI: 132	<ul style="list-style-type: none">• Se ha agregado compatibilidad con Red Hat Enterprise Linux 9, Rocky Linux 9 y CentOS Stream 9 en plataformas ARM.

2023.0-14852— 28 de marzo de 2023

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 392• Agente: 642• CLI: 132	<ul style="list-style-type: none">• Se ha agregado compatibilidad con Red Hat Enterprise Linux 9, Rocky Linux 9 y CentOS Stream 9.

2022.2-13907— 11 de noviembre de 2022

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none"> • Broker: 382 • Agente: 612 • CLI: 123 	<ul style="list-style-type: none"> • Se ha agregado un campo Substate en respuesta a DescribeSessions . • Se ha corregido un problema que podía provocar que la CLI no pudiera conectarse al broker en función de la URL utilizada.

2022.1-13067— 29 de junio de 2022

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none"> • Broker: 355 • Agente: 592 • CLI: 114 	<ul style="list-style-type: none"> • Se ha agregado compatibilidad para ejecutar el broker en instancias de AWS Graviton. • Se ha agregado compatibilidad con agentes y brokers para Ubuntu 22.04.

2022.0-11952— 23 de febrero de 2022

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none"> • Broker: 341 • Agente: 520 • CLI: 112 	<ul style="list-style-type: none"> • Se ha agregado la capacidad de rotación de registros al agente. • Se ha agregado un parámetro de configuración para establecer Java Home en el broker. • Se ha mejorado el vaciado de datos de la memoria caché al disco en el broker. • Validación de URL fija en la CLI.

2021.3-11591— 20 de diciembre de 2021

Números de compilación	Nuevas características
<ul style="list-style-type: none">• Broker: 307• Agente: 453• CLI: 92	<ul style="list-style-type: none">• Se ha agregado compatibilidad para la integración con la puerta de enlace de conexión NICE DCV.• Se ha agregado compatibilidad con Ubuntu 18.04 y Ubuntu 20.04.

2021.2-11445— 18 de noviembre de 2021

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 288• Agente: 413• CLI: 54	<ul style="list-style-type: none">• Se ha corregido un problema con la validación de los nombres de inicio de sesión que incluyen un dominio de Windows.

2021.2-11190— 11 de octubre de 2021

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 254• Agente: 413• CLI: 54	<ul style="list-style-type: none">• Se ha corregido un problema en la interfaz de la línea de comandos que impedía iniciar sesiones de Windows.

2021.2-11042— 01 de septiembre de 2021

Números de compilación	Nuevas características	Cambios y correcciones de errores
<ul style="list-style-type: none"> • Broker: 254 • Agente: 413 • CLI: 37 	<ul style="list-style-type: none"> • NICE DCV Session Manager ofrece ahora compatibilidad con la interfaz de la línea de comandos (CLI). Puede crear y administrar sesiones de NICE DCV en la CLI, en lugar de llamar a las API. • NICE DCV Session Manager ha introducido persistencia de datos del broker. Para mayor disponibilidad, los agentes pueden conservar la información de estado del servidor en un almacén de datos externo y restaurarlos al iniciar. 	<ul style="list-style-type: none"> • Al registrar un servidor de autorización externo, ahora puede especificar el algoritmo que utiliza el servidor de autorización para firmar los Web Tokens con formato JSON. Con este cambio, puede utilizar Azure AD como servidor de autorización externo.

2021.1-10557— 31 de mayo de 2021

Números de compilación	Nuevas características	Cambios y correcciones de errores
<ul style="list-style-type: none"> • Broker: 214 • Agente: 365 	<ul style="list-style-type: none"> • NICE DCV Session Manager ha agregado compatibilidad con los parámetros de entrada pasados al archivo de ejecución automática en Linux. 	<ul style="list-style-type: none"> • Hemos solucionado un problema con el archivo de ejecución automática en Windows.

Números de compilación	Nuevas características	Cambios y correcciones de errores
	<ul style="list-style-type: none"> Las propiedades del servidor ahora se pueden pasar como requisitos a la API CreateSessions. 	

2021.0-10242— 12 de abril de 2021

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none"> Broker: 183 Agente: 318 	<ul style="list-style-type: none"> NICE DCV Session Manager ha introducido las siguientes API nuevas: <ul style="list-style-type: none"> OpenServers CloseServers DescribeServers GetSessionScreenshots También ha introducido los siguientes parámetros de configuración nuevos: <ul style="list-style-type: none"> Parámetros del broker: <code>session-screenshot-max-width</code>, <code>session-screenshot-max-height</code>, <code>session-screenshot-format</code>, <code>create-sessions-queue-max-size</code> y <code>create-sessions-queue-max-time-seconds</code>. Parámetros del agente: <code>agent.autorun_folder</code>, <code>max_virtual_sessions</code> y <code>max_concurrent_sessions_per_user</code>. Parámetros del agente: <code>agent.autorun_folder</code>, <code>max_virtual_sessions</code> y <code>max_concurrent_sessions_per_user</code>. Parámetros del agente: <code>agent.autorun_folder</code>, <code>max_virtual_sessions</code> y <code>max_concurrent_sessions_per_user</code>.

2020.2-9662— 04 de diciembre de 2020

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none"> • Broker: 114 • Agente: 211 	<ul style="list-style-type: none"> • Hemos solucionado un problema con los certificados TLS generados automáticamente que impedían que el broker se iniciara.

2020.2-9508— 11 de noviembre de 2020

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none"> • Broker: 78 • Agente: 183 	<ul style="list-style-type: none"> • Versión inicial de NICE DCV Session Manager.

Historial de documentos

En la siguiente tabla se describe la documentación de esta versión de NICE DCV Session Manager.

Cambio	Descripción	Fecha
NICE DCV versión 2023.1	NICE DCV Session Manager se ha actualizado para NICE DCV 2023.1. Para obtener más información, consulte 2023.1— 9 de noviembre de 2023 .	9 de noviembre de 2023
NICE DCV versión 2023.0	NICE DCV Session Manager se ha actualizado para NICE DCV 2023.0. Para obtener más información, consulte 2023.0-14852— 28 de marzo de 2023 .	28 de marzo de 2023
NICE DCV versión 2022.2	NICE DCV Session Manager se ha actualizado para NICE DCV 2022.2. Para obtener más información, consulte	11 de noviembre de 2022

Cambio	Descripción	Fecha
	2022.2-13907— 11 de noviembre de 2022.	
NICE DCV versión 2022.1	NICE DCV Session Manager se ha actualizado para NICE DCV 2022.1. Para obtener más información, consulte 2022.1-13067— 29 de junio de 2022.	29 de junio de 2022
NICE DCV versión 2022.0	NICE DCV Session Manager se ha actualizado para NICE DCV 2022.0. Para obtener más información, consulte 2022.0-11952— 23 de febrero de 2022.	23 de febrero de 2022
NICE DCV versión 2021.3	NICE DCV Session Manager se ha actualizado para NICE DCV 2021.3. Para obtener más información, consulte 2021.3-11591— 20 de diciembre de 2021.	20 de diciembre de 2021
NICE DCV versión 2021.2	NICE DCV Session Manager se ha actualizado para NICE DCV 2021.2. Para obtener más información, consulte 2021.2-11042— 01 de septiembre de 2021.	01 de septiembre de 2021
NICE DCV versión 2021.1	NICE DCV Session Manager se ha actualizado para NICE DCV 2021.1. Para obtener más información, consulte 2021.1-10557— 31 de mayo de 2021.	31 de mayo de 2021
NICE DCV versión 2021.0	NICE DCV Session Manager se ha actualizado para NICE DCV 2021.0. Para obtener más información, consulte 2021.0-10242— 12 de abril de 2021.	12 de abril de 2021

Cambio	Descripción	Fecha
Versión inicial de NICE DCV Session Manager	Primera publicación de este contenido.	11 de noviembre de 2020

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.