



Guida per l'amministratore

Gestore di sessioni NICE DCV



Gestore di sessioni NICE DCV: Guida per l'amministratore

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Che cos'è Session Manager?	1
Come funziona Session Manager	1
Funzionalità	3
Limitazioni	3
Prezzi	3
Requisiti	4
Requisiti di rete e connettività	5
Configurazione	7
Fase 1: Preparare i server NICE DCV	7
Fase 2: Configurare il broker	8
Fase 3: Configurare l'agente	10
Fase 4: Configurazione del server NICE DCV	15
Fase 5: Verificare le installazioni	16
Verifica l'agente	17
Verifica il broker	18
Configurazione	19
Scalabilità di Session Manager	19
Fase 1: creazione di un profilo dell'istanza	20
Fase 2: Preparare il certificato SSL per il sistema di bilanciamento del carico	21
Fase 3: Creare il sistema di bilanciamento del carico dell'applicazione Broker	22
Passaggio 4: avviare i broker	23
Fase 5: Creare il sistema di bilanciamento del carico dell'applicazione Agent	24
Passaggio 6: Avvia gli agenti	25
Utilizzo dei tag	26
Configurazione di un server di autorizzazione esterno	28
Configurazione della persistenza del broker	33
Configura il broker in modo che resti su DynamoDB	34
Configura il broker per persistere su MariaDB/MySQL	35
Integrazione con il gateway di connessione NICE DCV	36
Configurare il broker Session Manager come risolutore di sessione per il gateway di connessione NICE DCV	36
Facoltativo: abilita l'autenticazione del client TLS	37
Server NICE DCV - mappatura DNS	39
Integrazione con Amazon CloudWatch	40

Aggiornamento	43
Aggiornamento dell'agente NICE DCV Session Manager	43
Aggiornamento del broker NICE DCV Session Manager	45
Riferimento CLI del broker	47
register-auth-server	47
Sintassi	48
Opzioni	48
Esempio	48
list-auth-servers	49
Sintassi	48
Output	49
Esempio	48
unregister-auth-server	50
Sintassi	48
Opzioni	48
Output	49
Esempio	48
register-api-client	51
Sintassi	48
Opzioni	48
Output	49
Esempio	48
describe-api-clients	53
Sintassi	48
Output	49
Esempio	48
unregister-api-client	54
Sintassi	48
Opzioni	48
Esempio	48
renew-auth-server-api-chiave	55
Sintassi	48
Esempio	48
generate-software-statement	56
Sintassi	48
Output	49

Esempio	48
describe-software-statements	57
Sintassi	48
Output	49
Esempio	48
deactivate-software-statement	58
Sintassi	48
Opzioni	48
Esempio	48
describe-agent-clients	59
Sintassi	48
Output	49
Esempio	48
unregister-agent-client	61
Sintassi	48
Opzioni	48
Esempio	48
register-server-dns-mappings	62
Sintassi	48
Opzioni	48
Esempio	48
describe-server-dns-mappings	62
Sintassi	48
Output	49
Esempio	48
Riferimento al file di configurazione	65
File di configurazione del broker	65
File di configurazione dell'agente	80
Note di rilascio e cronologia dei documenti	87
Note di rilascio	87
2023.1-16388— 26 giugno 2024	88
2023.1 — 9 novembre 2023	88
2023.0-15065— 4 maggio 2023	88
2023.0-14852— 28 marzo 2023	88
2022.2-13907 — 11 novembre 2022	89
2022.1-13067— 29 giugno 2022	89

2022.0-11952— 23 febbraio 2022	89
2021.3-11591— 20 dicembre 2021	90
2021.2-11445— 18 novembre 2021	90
2021.2-11190— 11 ottobre 2021	90
2021.2-11042— 01 settembre 2021	90
2021.1-10557— 31 maggio 2021	91
2021.0-10242 — 12 aprile 2021	91
2020.2-9662— 04 dicembre 2020	92
.....	92
Cronologia dei documenti	92
.....	XCV

Cos'è NICE DCV Session Manager?

NICE DCV Session Manager è un insieme di pacchetti software installabili (un agente e un broker) e un'interfaccia di programmazione delle applicazioni (API) che facilita agli sviluppatori e ai fornitori di software indipendenti (ISV) la creazione di applicazioni front-end che creano e gestiscono in modo programmatico il ciclo di vita delle sessioni NICE DCV su una flotta di server NICE DCV.

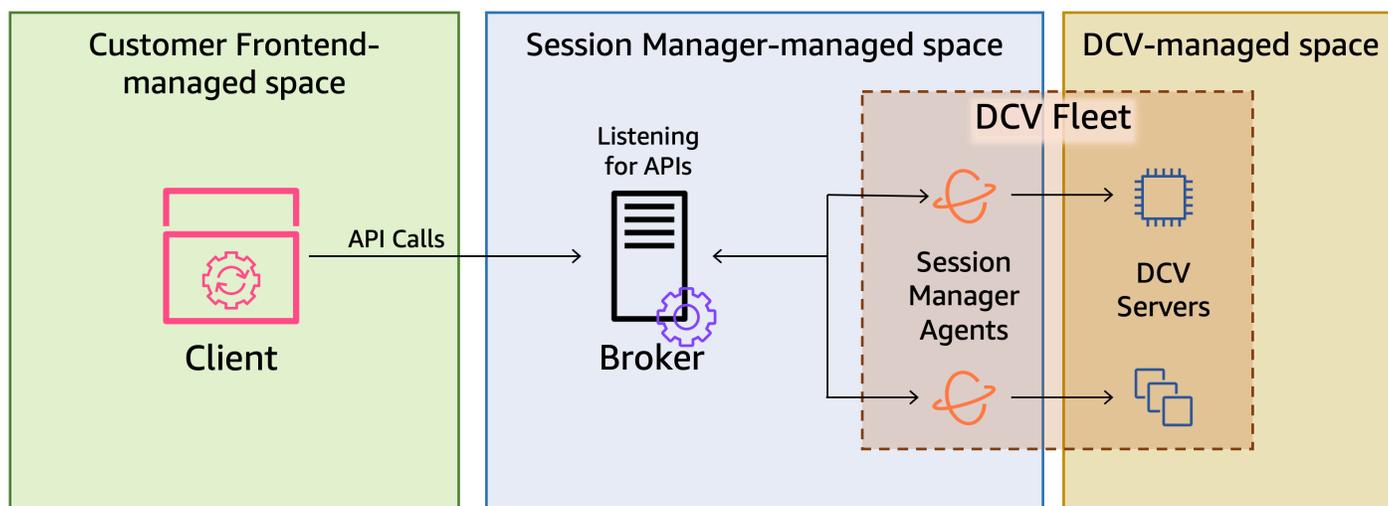
Questa guida spiega come installare e configurare l'agente e il broker di Session Manager. Per ulteriori informazioni sull'utilizzo delle API di Session Manager, consulta la Guida per gli sviluppatori di NICE DCV Session Manager.

Argomenti

- [Come funziona Session Manager](#)
- [Funzionalità](#)
- [Limitazioni](#)
- [Prezzi](#)
- [Requisiti di NICE DCV Session Manager](#)

Come funziona Session Manager

Il diagramma seguente mostra i componenti di alto livello di Session Manager.



Broker

Il Broker è un server web che ospita ed espone le API di Session Manager. Riceve ed elabora le richieste API per gestire le sessioni NICE DCV dal client, quindi trasmette le istruzioni agli agenti pertinenti. Il Broker deve essere installato su un host separato dai server NICE DCV, ma deve essere accessibile al client e deve poter accedere agli agenti.

Agente

L'agente è installato su ogni server NICE DCV del parco macchine. Gli agenti ricevono istruzioni dal Broker e le eseguono sui rispettivi server NICE DCV. Gli agenti monitorano inoltre lo stato dei server NICE DCV e inviano aggiornamenti periodici sullo stato al Broker.

API

Session Manager espone una serie di interfacce di programmazione delle applicazioni (API) REST che possono essere utilizzate per gestire le sessioni NICE DCV su una flotta di server NICE DCV. Le API sono ospitate ed esposte dal Broker. Gli sviluppatori possono creare client di gestione delle sessioni personalizzati che chiamano le API.

Cliente

Il client è l'applicazione o il portale front-end che sviluppi per chiamare le API di Session Manager esposte dal Broker. Gli utenti finali utilizzano il client per gestire le sessioni ospitate sui server NICE DCV della flotta.

Token di accesso

Per effettuare una richiesta API, devi fornire un token di accesso. I token possono essere richiesti al Broker o a un server di autorizzazione esterno tramite API client registrate. Per richiedere e accedere al token, l'API client deve fornire credenziali valide.

API del client

L'API client viene generata dal file YAML di definizione dell'API di Session Manager, utilizzando Swagger Codegen. L'API client viene utilizzata per effettuare richieste API.

Sessione NICE DCV

Devi creare una sessione NICE DCV sul tuo server NICE DCV a cui i tuoi clienti possano connettersi. I client possono connettersi a un server NICE DCV solo se c'è una sessione attiva. NICE DCV supporta console e sessioni virtuali. Utilizzate le API di Session Manager per gestire il ciclo di vita delle sessioni NICE DCV. Le sessioni NICE DCV possono trovarsi in uno dei seguenti stati:

- CREATING—il Broker sta creando la sessione.
- READY—la sessione è pronta ad accettare le connessioni client.
- DELETING—la sessione viene eliminata.
- DELETED—la sessione è stata eliminata.
- UNKNOWN—impossibile determinare lo stato della sessione. Il Broker e l'Agente potrebbero non essere in grado di comunicare.

Funzionalità

DCV Session Manager offre le seguenti funzionalità:

- Fornisce informazioni sulla sessione NICE DCV: ottieni informazioni sulle sessioni in esecuzione su più server NICE DCV.
- Gestisci il ciclo di vita di più sessioni NICE DCV: crea o elimina più sessioni per più utenti su più server NICE DCV con una sola richiesta API.
- Supporta i tag: utilizza tag personalizzati per indirizzare un gruppo di server NICE DCV durante la creazione di sessioni.
- Gestisce le autorizzazioni per più sessioni NICE DCV: modifica le autorizzazioni utente per più sessioni con una sola richiesta API.
- Fornisce informazioni sulla connessione: recupera le informazioni di connessione del client per le sessioni NICE DCV.
- Supporti per il cloud e l'ambiente locale: utilizza Session Manager in localeAWS, o con server alternativi basati sul cloud.

Limitazioni

Session Manager non fornisce funzionalità di provisioning delle risorse. Se utilizzi NICE DCV su istanze Amazon EC2, potresti dover utilizzare servizi AWS aggiuntivi, come Amazon EC2 Auto Scaling per gestire la scalabilità della tua infrastruttura.

Prezzi

Session Manager è disponibile gratuitamente per i clienti che utilizzano istanze EC2. AWS

I clienti locali richiedono una licenza NICE DCV Plus o DCV Professional Plus. Per informazioni su come acquistare una licenza NICE DCV Plus o NICE DCV Professional Plus, [consulta Come acquistare](#) sul sito Web NICE e trova un distributore o rivenditore NICE nella tua regione. Per consentire a tutti i clienti locali di sperimentare DCV Session Manager, i requisiti di licenza verranno applicati solo a partire dalla versione 2021.0 di NICE DCV.

Per ulteriori informazioni, consulta la sezione [Licenze del server NICE DCV nella NICE DCV Administrator Guide](#).

Requisiti di NICE DCV Session Manager

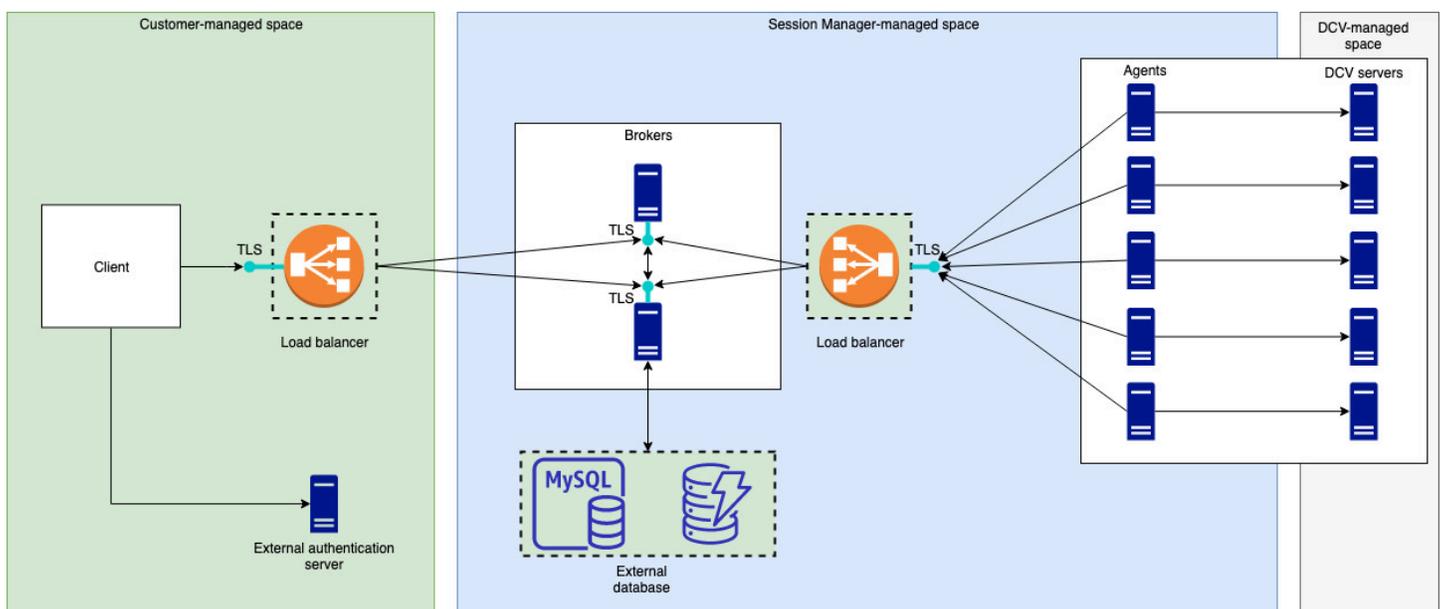
L'agente e il broker di NICE DCV Session Manager hanno i seguenti requisiti.

	Broker	Agente
Sistema operativo	<ul style="list-style-type: none"> • Amazon Linux 2 • CentOS 7.6 o versione successiva • CentOS Stream 8 • CentOS Stream 9 • RHEL 7.6 o versione successiva • RHEL 8.x • RHEL 9.x • Rocky Linux 8.5 o versione successiva • Rocky Linux 9.x • Ubuntu 20.04 • Ubuntu 22.04 	<ul style="list-style-type: none"> • Windows <ul style="list-style-type: none"> • Windows Server 2022 • Windows Server 2019 • Windows Server 2016 • Server Linux <ul style="list-style-type: none"> • Amazon Linux 2 • CentOS 7.6 o versione successiva • CentOS Stream 8 • CentOS Stream 9 • RHEL 7.6 o versione successiva • RHEL 8.x • RHEL 9.x • Rocky Linux 8.5 o versione successiva • Rocky Linux 9.x • Ubuntu 20.04 • Ubuntu 22.04

	Broker	Agente
		<ul style="list-style-type: none"> • SUSE Linux Enterprise 12 con SP4 o versione successiva • SUSE Linux Enterprise 15
Architettura	<ul style="list-style-type: none"> • x86 a 64 bit • ARM a 64 bit 	<ul style="list-style-type: none"> • x86 a 64 bit • ARM a 64 bit (solo Amazon Linux 2, CentOS 7.x/8.x/9.x, RHEL 7.x/8.x/9.x e Rocky 8.x/9.x) • ARM a 64 bit (Ubuntu 22.04)
Memoria	8 GB	4 GB
Versione NICE DCV	NICE DCV 2020.2 e versioni successive	NICE DCV 2020.2 e versioni successive
Requisiti aggiuntivi	Java 11	-

Requisiti di rete e connettività

Il diagramma seguente fornisce una panoramica di alto livello dei requisiti di rete e connettività di Session Manager.



Il Broker deve essere installato su un host separato, ma deve disporre di connettività di rete con gli agenti sui server NICE DCV. Se si sceglie di avere più broker per migliorare la disponibilità, è necessario installare e configurare ciascun broker su un host separato e utilizzare uno o più sistemi di bilanciamento del carico per gestire il traffico tra il client e i broker e tra i broker e gli agenti. I Broker dovrebbero inoltre essere in grado di comunicare tra loro per scambiarsi informazioni sui server e sulle sessioni del NICE DCV. I Broker possono memorizzare le proprie chiavi e i dati di stato su un database esterno e rendere queste informazioni disponibili dopo il riavvio o la chiusura. Questo aiuta a mitigare il rischio di perdere importanti informazioni sul Broker mantenendole sul database esterno. Puoi recuperarlo in un secondo momento. Se scegli di averlo, devi configurare il database esterno e configurare i broker. DynamoDB, MariaDB e MySQL sono supportati. [I parametri di configurazione sono elencati nel file di configurazione del broker.](#)

Gli agenti devono essere in grado di avviare connessioni HTTPS sicure, persistenti e bidirezionali con il Broker.

Il client, o l'applicazione frontend, deve essere in grado di accedere al Broker per chiamare le API. Il client dovrebbe inoltre essere in grado di accedere al server di autenticazione.

Configurazione del gestore di sessioni NICE DCV

La sezione seguente spiega come installare Session Manager con un singolo broker e più agenti. È possibile utilizzare più broker per migliorare la scalabilità e le prestazioni. Per ulteriori informazioni, consulta [Gestione delle sessioni di scalabilità](#).

Per configurare NICE DCV Session Manager, procedi come segue:

Fasi

- [Fase 1: Preparare i server NICE DCV](#)
- [Fase 2: Configurazione del broker NICE DCV Session Manager](#)
- [Fase 3: Configurare l'agente NICE DCV Session Manager](#)
- [Fase 4: Configurare il server NICE DCV per utilizzare il Broker come server di autenticazione](#)
- [Fase 5: Verificare le installazioni](#)

Fase 1: Preparare i server NICE DCV

È necessario disporre di una flotta di server NICE DCV con cui si intende utilizzare Session Manager. Per ulteriori informazioni sull'installazione dei server NICE DCV, consultate [Installazione del server NICE DCV nella NICE DCV Administrator Guide](#).

Sui server Linux NICE DCV, Session Manager utilizza un utente del servizio locale denominato `dcvsmagent`. Questo utente viene creato automaticamente quando viene installato l'agente Session Manager. È necessario concedere a questo utente del servizio i privilegi di amministratore per NICE DCV in modo che possa eseguire azioni per conto di altri utenti. Per concedere i privilegi di amministratore agli utenti del servizio Session Manager, procedi come segue:

Per aggiungere l'utente del servizio locale per i server Linux NICE DCV

1. Apri `/etc/dcv/dcv.conf` con qualsiasi editor di testo.
2. Aggiungere il `administrators` parametro alla `[security]` sezione e specificare l'utente del Session Manager. Per esempio:

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

3. Salva e chiudi il file.
4. Arresta e riavvia il server NICE DCV.

Session Manager è in grado di creare sessioni NICE DCV solo per conto di utenti già esistenti sul server NICE DCV. Se viene effettuata una richiesta per creare una sessione per un utente che non esiste, la richiesta fallisce. Pertanto, è necessario assicurarsi che ogni utente finale previsto disponga di un utente di sistema valido sul server NICE DCV.

Tip

Se intendi utilizzare più host Broker o server NICE DCV con agenti, ti consigliamo di configurare solo un Broker e un server NICE DCV con un agente eseguendo i seguenti passaggi, creando Amazon Machine Images (AMI) degli host con le configurazioni completate e quindi utilizzando le AMI per avviare i Brokers e i server NICE DCV rimanenti. In alternativa, è possibile utilizzare AWS Systems Manager per eseguire i comandi su più istanze in remoto.

Fase 2: Configurazione del broker NICE DCV Session Manager

Il Broker deve essere installato su un host Linux. Per ulteriori informazioni sulle distribuzioni Linux supportate, vedere [Requisiti di NICE DCV Session Manager](#). Installa il Broker su un host separato dall'agente e dall'host del server NICE DCV. L'host può essere installato su una rete privata diversa, ma deve essere in grado di connettersi e comunicare con l'agente.

Per installare e avviare il Broker

1. Connettiti all'host su cui intendi installare il Broker.
2. I pacchetti del sono firmati digitalmente con una firma GPG sicura. Per consentire al gestore di pacchetti di verificare la firma del pacchetto, è necessario importare la chiave NICE GPG. Eseguite il seguente comando per importare la chiave NICE GPG.

- Amazon Linux 2, RHEL, CentOS e 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. Scarica il pacchetto di installazione.

- Amazon Linux 2, RHEL 7.x e 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 e 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. Installare il pacchetto .

- Amazon Linux 2, RHEL 7.x e 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 e 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. Verifica che la versione predefinita dell'ambiente Java sia 11

```
$ java -version
```

In caso contrario, è possibile impostare in modo esplicito la home directory Java che il Broker utilizzerà per indirizzare la versione Java corretta. Questo viene fatto impostando il parametro `broker-java-home` nel file di configurazione del Broker. Per ulteriori informazioni, consulta [Broker Configuration File](#).

6. Avvia il servizio Broker e assicurati che si avvii automaticamente a ogni avvio dell'istanza.

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

7. Inserisci una copia del certificato autofirmato del Broker nella tua directory utente. Ne avrai bisogno quando installerai gli agenti nella fase successiva.

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

Fase 3: Configurare l'agente NICE DCV Session Manager

L'agente deve essere installato su tutti gli host di server NICE DCV del parco macchine. L'agente può essere installato su server Windows e Linux. Per ulteriori informazioni sui sistemi operativi supportati, vedere [Requisiti di NICE DCV Session Manager](#).

Prerequisiti

Il server NICE DCV deve essere installato sull'host prima di installare l'agente.

Linux host

 Note

[L'agente Session Manager è disponibile per le distribuzioni e le architetture Linux elencate in Requisiti:](#)

Le seguenti istruzioni riguardano l'installazione dell'agente su host x86 a 64 bit. Per installare l'agente su host ARM a 64 bit, sostituire `x86_64` con. `aarch64` *Per Ubuntu, sostituisci `amd64` con. `arm64`*

Per installare l'agente su un host Linux

1. I pacchetti del sono firmati digitalmente con una firma GPG sicura. Per consentire al gestore di pacchetti di verificare la firma del pacchetto, è necessario importare la chiave NICE GPG. Eseguite il seguente comando per importare la chiave NICE GPG.

- Amazon Linux 2, RHEL, CentOS e 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. Scarica il pacchetto di installazione.

- Amazon Linux 2, RHEL 7.x e 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 e 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. Installare il pacchetto .

- Amazon Linux 2, RHEL 7.x e 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 e 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
```

- Inserisci una copia del certificato autofirmato del Broker (che hai copiato nel passaggio precedente) nella `/etc/dcv-session-manager-agent/` directory dell'agente.
- Apri `/etc/dcv-session-manager-agent/agent.conf` utilizzando il tuo editor di testo preferito e procedi come segue.
 - `Perbroker_host`, specifica il nome DNS dell'host su cui è installato il Broker.

 Important

Se il Broker è in esecuzione su un'istanza Amazon EC2, `broker_host` devi specificare l'indirizzo IPv4 privato dell'istanza.

- (Facoltativo) `Perbroker_port`, specifica la porta tramite la quale comunicare con il Broker. Per impostazione predefinita, l'agente e il broker comunicano tramite porta 8445. Modificalo solo se devi usare una porta diversa. Se la modifichi, assicurati che il Broker sia configurato per utilizzare la stessa porta.
- `Perca_file`, specifica il percorso completo del file di certificato che hai copiato nel passaggio precedente. Per esempio:

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

In alternativa, se desideri disabilitare la verifica TLS, imposta `sutls_strict: false`

- Salva e chiudi il file.
- Esegui il comando seguente per avviare l'agente.

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

Windows host

Per installare l'agente su un host Windows

- Scarica il programma di [installazione dell'agente](#).

2. Eseguire il programma di installazione. Nella schermata iniziale, selezionare Next (Successivo).
3. Nella schermata EULA, leggi attentamente il contratto di licenza e, se sei d'accordo, seleziona Accetto i termini e scegli Avanti.
4. Per iniziare l'installazione, scegli Installa.
5. Inserisci una copia del certificato autofirmato del Broker (che hai copiato nel passaggio precedente) nella C:\Program Files\NICE\DCVSessionManagerAgent\conf\ cartella dell'agente.
6. Apri C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf utilizzando il tuo editor di testo preferito, quindi procedi come segue:
 - Perbroker_host, specifica il nome DNS dell'host su cui è installato il Broker.

 Important

Se il Broker è in esecuzione su un'istanza Amazon EC2, broker_host devi specificare l'indirizzo IPv4 privato dell'istanza.

- (Facoltativo) Perbroker_port, specifica la porta tramite la quale comunicare con il Broker. Per impostazione predefinita, l'agente e il broker comunicano tramite porta8445. Modificalo solo se devi usare una porta diversa. Se la modifichi, assicurati che il Broker sia configurato per utilizzare la stessa porta.
- Perca_file, specifica il percorso completo del file di certificato che hai copiato nel passaggio precedente. Per esempio:

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

In alternativa, se desideri disabilitare la verifica TLS, imposta sutls_strict. false

7. Salva e chiudi il file.
8. Arresta e riavvia il servizio Agent per rendere effettive le modifiche. Esegui i seguenti comandi al prompt dei comandi.

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

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

Fase 4: Configurare il server NICE DCV per utilizzare il Broker come server di autenticazione

Configurare il server NICE DCV per utilizzare il Broker come server di autenticazione esterno per la convalida dei token di connessione del client. È inoltre necessario configurare il server NICE DCV in modo che consideri attendibile la CA autofirmata del Broker.

Linux NICE DCV server

Per aggiungere l'utente del servizio locale per i server Linux NICE DCV

1. Apri `/etc/dcv/dcv.conf` con qualsiasi editor di testo.
2. Aggiungere i `auth-token-verifier` parametri `ca-file` and alla `[security]` sezione.

`Perca-file`, specifica il percorso della CA autofirmata del Broker che hai copiato sull'host nel passaggio precedente.

`Perauth-token-verifier`, specifica l'URL per il verificatore del token sul Broker nel seguente formato: `https://broker_ip_or_dns:port/agent/validate-authentication-token` Specificare la porta utilizzata per la comunicazione Broker-Agent, che per impostazione predefinita è 8445. Se esegui il Broker su un'istanza Amazon EC2, devi utilizzare il DNS privato o l'indirizzo IP privato.

Ad esempio

```
[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. Salva e chiudi il file.
4. Arresta e riavvia il server NICE DCV. Per ulteriori informazioni, consulta [Arresto del server NICE DCV](#) e [Avvio del server NICE DCV nella Guida per l'amministratore di NICE DCV](#).

Windows NICE DCV server

Sui server Windows NICE DCV

1. Apri l'editor del registro di Windows e vai alla chiave HKEY_USERS/S-1-5-18/software/gsettings/com/NiceSoftware/DCV/Security/.
2. Apri il parametro ca-file. Per Value data, specifica il percorso della CA autofirmata del Broker che hai copiato sull'host nel passaggio precedente.

Note

Se il parametro non esiste, crea un nuovo parametro di stringa e assegnagli un nome. ca-file

3. Aprite il auth-token-verifierparametro. Per i dati Value, specifica l'URL per il verificatore del token sul Broker nel seguente formato: `https://broker_ip_or_dns:port/agent/validate-authentication-token` Specificare la porta utilizzata per la comunicazione Broker-Agent, che per impostazione predefinita è 8445. Se esegui il Broker su un'istanza Amazon EC2, devi utilizzare il DNS privato o l'indirizzo IP privato.

Note

Se il parametro non esiste, crea un nuovo parametro di stringa e assegnagli un nome. auth-token-verifier

4. Seleziona OK e chiudi l'editor del Registro di sistema di Windows.
5. Arresta e riavvia il server NICE DCV. Per ulteriori informazioni, consulta [Arresto del server NICE DCV](#) e [Avvio del server NICE DCV nella Guida per l'amministratore di NICE DCV](#).

Fase 5: Verificare le installazioni

Argomenti

- [Verifica l'agente](#)
- [Verifica il broker](#)

Verifica l'agente

Dopo aver installato il broker e l'agente, assicurati che l'agente sia in esecuzione e che sia in grado di connettersi al broker.

Host Linux Agent

Il comando da eseguire dipende dalla versione.

- Dalla versione 2022.0

Dall'host dell'agente, esegui il seguente comando:

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

- Versioni precedenti alla 2022.0

Dall'host dell'agente, esegui il comando seguente e specifica l'anno, il mese e il giorno correnti.

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

Ad esempio

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

Se l'agente è in esecuzione ed è in grado di connettersi al broker, il comando dovrebbe tornare success.

Se il comando restituisce un output diverso, controllate il file di registro dell'agente per ulteriori informazioni. I file di registro si trovano qui: `/var/log/dcv-session-manager-agent/`.

Host di Windows Agent

Aprire il file di registro dell'agente, che si trova in `C:\ProgramData\NICE\DCVSessionManagerAgent\log`.

Se il file di registro include una riga simile a quella riportata di seguito, l'agente è in esecuzione ed è in grado di connettersi 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}"
```

Se il file di registro non ha una riga simile, ispeziona il file di registro per individuare eventuali errori.

Verifica il broker

Dopo aver installato il Broker e l'Agente, assicurati che il Broker sia in esecuzione e che sia raggiungibile dagli utenti e dalle applicazioni front-end.

Da un computer che dovrebbe essere in grado di raggiungere il Broker, esegui il seguente comando:

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

Se la verifica ha esito positivo, il Broker restituisce quanto segue:

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


- [Fase 5: Creare il sistema di bilanciamento del carico dell'applicazione Agent](#)
- [Passaggio 6: Avvia gli agenti](#)

Fase 1: creazione di un profilo dell'istanza

È necessario collegare un profilo di istanza agli host Broker e Agent che consentano loro di utilizzare le API Elastic Load Balancing. Per ulteriori informazioni, consulta [i ruoli IAM per Amazon EC2 nella Amazon EC2 User Guide](#).

Come creare un profilo dell'istanza

1. Crea un ruolo AWS Identity and Access Management (IAM) che definisca le autorizzazioni da utilizzare nel profilo dell'istanza. Utilizza la seguente politica di fiducia:

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

Quindi allega la seguente politica:

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

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

Per ulteriori informazioni, consulta [Creazione di un ruolo IAM](#) nella Guida per l'utente IAM.

2. Crea un nuovo profilo di istanza. Per ulteriori informazioni, consulta [create-instance-profile](#) nel Command Reference.AWS CLI
3. Aggiungere il ruolo IAM al profilo dell'istanza. Per ulteriori informazioni, vedere [add-role-to-instance-profile](#) nella Guida ai comandi.AWS CLI
4. Allega il profilo dell'istanza agli host del Broker. Per ulteriori informazioni, consulta [Associare un ruolo IAM a un'istanza](#) nella Amazon EC2 User Guide.

Fase 2: Preparare il certificato SSL per il sistema di bilanciamento del carico

Quando utilizzi HTTPS per il listener del load balancer, devi distribuire un certificato SSL al load balancer. Il load balancer utilizzerà il certificato per terminare la connessione e decrittografare le richieste provenienti dai client prima di inoltrarle ai target.

Per preparare il certificato SSL

1. Creare un'autorità di certificazione privata (CA) AWS Certificate Manager Private Certificate Authority (ACM PCA). Per ulteriori informazioni, vedere [Procedure per la creazione di una CA](#) nella Guida per l' AWS utente dell'autorità di certificazione privata di Certificate Manager.
2. Installa la CA. Per ulteriori informazioni, vedere [Installazione di un certificato CA root](#) nella Guida per l' AWS utente dell'autorità di certificazione privata di Certificate Manager.
3. Richiedi un nuovo certificato privato firmato dalla CA. Per il nome di dominio, usa `*.region.elb.amazonaws.com` e specifica la regione in cui intendi creare il load balancer. Per ulteriori informazioni, vedere [Richiesta di un certificato privato](#) nella Guida per l'utente dell'autorità di AWS certificazione privata di Certificate Manager.

Fase 3: Creare il sistema di bilanciamento del carico dell'applicazione Broker

Crea un sistema di bilanciamento del carico delle applicazioni per bilanciare il traffico tra i tuoi client front-end e i broker.

Per creare il sistema di bilanciamento del carico

1. Apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.

Nel riquadro di navigazione, scegli Load Balancer, quindi scegli Create Load Balancer. Per il tipo di load balancer, scegli Application Load Balancer.

2. Per Step 1: Configure Load Balancer (Fase 1: configurare il sistema di bilanciamento del carico), procedere come segue:
 - a. In Nome, inserisci un nome descrittivo per il load balancer.
 - b. Per Scheme, selezionate Internet-facing.
 - c. Per Load Balancer Protocol, seleziona HTTPS e per Load Balancer Port, inserisci. 8443
 - d. Per VPC, seleziona il VPC da utilizzare, quindi seleziona tutte le sottoreti in quel VPC.
 - e. Seleziona Successivo.
3. Per il passaggio 2: configurare le impostazioni di sicurezza, procedi come segue:
 - a. Per Tipo di certificato, scegli Scegli un certificato da ACM.
 - b. Per Nome del certificato, seleziona il certificato privato richiesto in precedenza.
 - c. Seleziona Successivo.
4. Per la fase 3: configura i gruppi di sicurezza, crea un nuovo gruppo di sicurezza o seleziona un gruppo di sicurezza esistente che consenta il traffico in entrata e in uscita tra il client frontend e i broker tramite HTTPS e la porta 8443.

Seleziona Successivo.

5. Per il Passaggio 4: Configurazione del routing, procedi come segue:
 - a. Per Gruppo target, seleziona Nuovo gruppo target.
 - b. In Name (Nome), inserire un nome per il gruppo di destinazione.
 - c. Per Tipo di destinazione, scegli Istanza.

- d. Per Protocollo, seleziona HTTPS. Per Port (Porta), immettere 8443. Per la versione del protocollo, scegli HTTP1.
 - e. Per il protocollo di controllo dello stato, scegli HTTPS e per Path, inserisci/health.
 - f. Seleziona Successivo.
6. Per il passaggio 5: registrazione degli obiettivi, scegli Avanti.
 7. Scegli Crea.

Passaggio 4: avviare i broker

Crea un broker iniziale e configuralo per utilizzare il load balancer, crea un'AMI dal broker e quindi utilizza l'AMI per avviare i broker rimanenti. Ciò garantisce che tutti i broker siano configurati per utilizzare la stessa CA e la stessa configurazione del load balancer.

Per avviare i Broker

1. Avvia e configura l'host iniziale del Broker. Per ulteriori informazioni sull'installazione e la configurazione del Broker, vedere [Fase 2: Configurazione del broker NICE DCV Session Manager](#).

Note

Il certificato autofirmato del broker non è necessario poiché utilizziamo un sistema di bilanciamento del carico delle applicazioni.

2. Connettiti al Broker, apri `/etc/dcv-session-manager-broker/session-manager-broker.properties` utilizzando il tuo editor di testo preferito ed esegui le seguenti operazioni:
 - a. Commenta il `broker-to-broker-discovery-addresses` parametro inserendo un cancelletto (`#`) all'inizio della riga.
 - b. Per `broker-to-broker-discovery-aws-region`, inserite la regione in cui avete creato il sistema di bilanciamento del carico dell'applicazione.
 - c. Per `broker-to-broker-discovery-aws-alb-target-group-arn`, inserire l'ARN del gruppo target associato al sistema di bilanciamento del carico del Broker.
 - d. Salva e chiudi il file.
3. Arresta l'istanza Broker.

4. Crea un AMI dall'istanza Broker interrotta. Per ulteriori informazioni, consulta [Creazione di un'AMI Linux da un'istanza](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.
5. Usa l'AMI per avviare i broker rimanenti.
6. Assegna il profilo di istanza che hai creato a tutte le istanze Broker.
7. Assegna un gruppo di sicurezza che consenta a Broker a Broker e Broker di bilanciare il traffico di rete su tutte le istanze Broker. Per ulteriori informazioni sulle porte di rete, vedere [Broker Configuration File](#).
8. Registra tutte le istanze del Broker come destinazioni per il bilanciamento del carico del Broker. Per ulteriori informazioni, consulta [Registrare gli obiettivi con il proprio gruppo target](#) nella Guida utente per Application Load Balancers.

Fase 5: Creare il sistema di bilanciamento del carico dell'applicazione Agent

Crea un sistema di bilanciamento del carico delle applicazioni per bilanciare gli agenti e i broker.

Per creare il sistema di bilanciamento del carico

1. Apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.

Nel riquadro di navigazione, scegli Load Balancer, quindi scegli Create Load Balancer. Per il tipo di load balancer, scegli Application Load Balancer.

2. Per Step 1: Configure Load Balancer (Fase 1: configurare il sistema di bilanciamento del carico), procedere come segue:
 - a. In Nome, inserisci un nome descrittivo per il load balancer.
 - b. Per Scheme, selezionate Internet-facing.
 - c. Per Load Balancer Protocol, seleziona HTTPS e per Load Balancer Port, inserisci. 8445
 - d. Per VPC, seleziona il VPC da utilizzare, quindi seleziona tutte le sottoreti in quel VPC.
 - e. Seleziona Successivo.
3. Per il passaggio 2: configurare le impostazioni di sicurezza, procedi come segue:
 - a. Per Tipo di certificato, scegli Scegli un certificato da ACM.
 - b. Per Nome del certificato, seleziona il certificato privato richiesto in precedenza.
 - c. Seleziona Successivo.

4. Per la fase 3: configurare i gruppi di sicurezza, creare un nuovo gruppo di sicurezza o selezionare un gruppo di sicurezza esistente che consenta il traffico in entrata e in uscita, gli agenti e i broker su HTTPS e la porta 8445.

Seleziona Successivo.

5. Per il Passaggio 4: Configurazione del routing, procedi come segue:
 - a. Per Gruppo target, seleziona Nuovo gruppo target.
 - b. In Name (Nome), inserire un nome per il gruppo di destinazione.
 - c. Per Tipo di destinazione, scegli Istanza.
 - d. Per Protocollo, seleziona HTTPS. Per Port (Porta), immettere 8445. Per la versione del protocollo, scegli HTTP1.
 - e. Per il protocollo di controllo dello stato, scegli HTTPS e per Path, inserisci/health.
 - f. Seleziona Successivo.
6. Per il passaggio 5: Registrazione degli obiettivi, seleziona tutte le istanze del Broker e scegli Aggiungi ai registrati. Scegli Prossimo: Rivedi.
7. Scegli Crea.

Passaggio 6: Avvia gli agenti

Crea un agente iniziale e configuralo per utilizzare il sistema di bilanciamento del carico, crea un'AMI dall'agente e quindi utilizza l'AMI per avviare gli agenti rimanenti. Ciò garantisce che tutti gli agenti siano configurati per utilizzare la stessa configurazione del load balancer.

Per avviare gli agenti

1. Preparate il server NICE DCV. Per ulteriori informazioni, consulta [Fase 1: Preparare i server NICE DCV](#).
2. Inserisci una copia della chiave pubblica CA creata in [Fase 2: Preparare il certificato SSL per il sistema di bilanciamento del carico](#). Scegli o crea una directory leggibile da qualsiasi utente. Il file della chiave pubblica CA deve essere leggibile anche da qualsiasi utente.
3. Installa e configura l'agente. Per ulteriori informazioni sull'installazione e la configurazione dell'agente, vedere [Fase 3: Configurare l'agente NICE DCV Session Manager](#).

⚠ Important

Quando si modifica il file di configurazione dell'agente:

- per il `broker_host` parametro, inserite il DNS dell'Agent Load Balancer
- per il `ca_file` parametro, immettere il percorso del file di chiave pubblica CA creato nel passaggio precedente

4. Configurare il server NICE DCV per utilizzare il Broker come server di autenticazione. Per ulteriori informazioni, consulta [Fase 4: Configurare il server NICE DCV per utilizzare il Broker come server di autenticazione](#).

⚠ Important

Quando si modifica il file di configurazione del server NICE DCV:

- per il `ca-file` parametro, immettete lo stesso percorso del file di chiave pubblica CA utilizzato nel passaggio precedente
- *per il `auth-token-verifier` parametro, utilizzate il DNS dell'Agent load balancer per `broker_ip_or_dns`*

5. Arresta l'istanza dell'agente.
6. Crea un'AMI dall'istanza Agent interrotta. Per ulteriori informazioni, consulta [Creazione di un'AMI Linux da un'istanza](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.
7. Utilizza l'AMI per avviare gli agenti rimanenti e assegnare il profilo di istanza che hai creato a tutti.
8. Assegna un gruppo di sicurezza che consenta all'agente di bilanciare il traffico di rete a tutte le istanze dell'agente. Per ulteriori informazioni sulle porte di rete, vedere [Agent Configuration File](#).

Utilizzo dei tag per indirizzare i server NICE DCV

È possibile assegnare tag personalizzati agli agenti di Session Manager per aiutarli a identificarli e classificarli e i server NICE DCV a cui sono associati. Quando si crea una nuova sessione NICE DCV, è possibile scegliere come target un gruppo di server NICE DCV in base ai tag assegnati ai rispettivi agenti. Per ulteriori informazioni su come indirizzare i server NICE DCV [CreateSessionRequests](#) in base ai tag Agent, consulta la Session Manager Developer Guide.

Un tag è composto da una chiave di tag e una coppia di valori ed è possibile utilizzare qualsiasi coppia di informazioni utile per il caso d'uso o l'ambiente. Puoi scegliere di etichettare gli agenti in base alla configurazione hardware del loro host. Ad esempio, puoi etichettare tutti gli agenti con host con 4 GB di memoria ram=4GB. Oppure puoi taggare gli agenti in base allo scopo. Ad esempio, puoi applicare tag a tutti gli agenti in esecuzione sugli host di produzione con purpose=production.

Per assegnare tag a un agente

1. Utilizzando l'editor di testo preferito, creare un nuovo file e assegnargli un nome descrittivo, ad esempio `agent_tags.toml`. Il tipo di file deve essere `.toml` e il contenuto del file deve essere specificato nel formato di file TOML.
2. Nel file, aggiungi ogni nuova coppia di tag, chiave e valore su una nuova riga utilizzando il `key=value` formato. Ad esempio:

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

3. Aprire il file di configurazione dell'agente (`/etc/dcv-session-manager-agent/agent.conf` per Linux o `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` per Windows). Per `tags_folder` e specifica il percorso della directory in cui si trova il file di tag.

Se la directory contiene più file di tag, tutti i tag definiti nei file applicano l'Agent. I file vengono letti in ordine alfabetico. Se più file contengono un tag con la stessa chiave, il valore viene sovrascritto con il valore dell'ultimo file letto.

4. Salva e chiudi il file.
5. Arrestare e riavviare l'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
```

Configurazione di un server di autorizzazione esterno

Il server di autorizzazione è il server responsabile dell'autenticazione e dell'autorizzazione degli SDK e degli agenti client.

Per impostazione predefinita, Session Manager utilizza il Broker come server di autorizzazione per generare token di accesso OAuth 2.0 per gli SDK dei client e le istruzioni software per gli agenti. Se si utilizza il Broker come server di autorizzazione, non è richiesta alcuna configurazione aggiuntiva.

Puoi configurare Session Manager per utilizzare Amazon Cognito come server di autorizzazione esterno anziché come Broker. Per ulteriori informazioni su Amazon Cognito, consulta la [Guida per sviluppatori di Amazon Cognito](#).

Per utilizzare Amazon Cognito come server di autorizzazione

1. Crea un nuovo pool di utenti di Amazon Cognito. Per ulteriori informazioni sui pool di utenti, consulta [Caratteristiche di Amazon Cognito](#) nella Amazon Cognito Developer Guide.

Utilizzate il [create-user-pool](#) comando e specificate il nome del pool e la regione in cui crearlo.

In questo esempio, diamo un nome al pool `dcv-session-manager-client-app` e lo creiamo in `us-east-1`.

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

Output di esempio

```
{  
  "UserPoolClient": {  
    "UserPoolId": "us-east-1_QLEXAMPLE",  
    "ClientName": "dcv-session-manager-client-app",  
    "ClientId": "15hhd8jjj74hf32f24uEXAMPLE",  
    "LastModifiedDate": 1602510048.054,  
    "CreationDate": 1602510048.054,  
    "RefreshTokenValidity": 30,  
    "AllowedOAuthFlowsUserPoolClient": false
```

```
}
}
```

Prendi nota di `userPoolId` dato poiché sarà necessario nella fase successiva.

2. Crea un nuovo dominio per il dominio di utenti. Usa il [create-user-pool-domain](#) comando e specifica un nome di dominio e il nome del `userPoolId` di dominio e il nome del client di utenti che hai creato nella fase precedente.

In questo esempio, il nome di dominio è `mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE` e lo creiamo in `us-east-1`.

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

Output di esempio

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

Il formato del dominio del pool di utenti è il seguente: `https://domain_name.auth.region.amazoncognito.com`. In questo esempio, il dominio del pool di utenti è `https://mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE.auth.us-east-1.amazoncognito.com`.

3. Creazione di un client pool di utenti Usa il [create-user-pool-client](#) comando e specifica il `userPoolId` di utenti che hai creato, un nome per il client e la regione in cui crearlo. Inoltre, includi l' `--generate-secret` opzione per specificare che si desidera generare un segreto per il client di utenti che si sta creando.

In questo caso, il nome del cliente è `dcv-session-manager-client-app` e lo creiamo nella `us-east-1` Regione.

```
$ 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
```

Output di esempio

```
{
  "UserPoolClient": {
    "UserPoolId": "us-east-1_QLEXAMPLE",
    "ClientName": "dcv-session-manager-client-app",
    "ClientId": "219273hp6k2ut5cugg9EXAMPLE",
    "ClientSecret": "1vp5e8nec7cbf4m9me55mbmht91u61hlh0a78rq1qki11EXAMPLE",
    "LastModifiedDate": 1602510291.498,
    "CreationDate": 1602510291.498,
    "RefreshTokenValidity": 30,
    "AllowedOAuthFlowsUserPoolClient": false
  }
}
```

Note

Prendi nota della `ClientId` e `ClientSecret`. Dovrai fornire queste informazioni agli sviluppatori quando richiederanno i token di accesso per le richieste API.

4. Creazione di un nuovo server di risorse OAuth2.0 per il server di risorse OAuth2.0 per il server di utenti. Un server di risorse è un server di risorse ad accesso protetto. Gestisce le richieste autenticate di token di accesso.

Usa il [create-resource-server](#) comando e specifica il nome `userPoolId` del pool di utenti, un identificatore e un nome univoci per il server di risorse, l'ambito e la regione in cui crearlo.

In questo esempio, utilizziamo `dcv-session-manager` come identificatore e nome `esm_scope` come nome dell'ambito.

```
$ 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
```

Output di esempio

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

5. Aggiorna il client di utenti.

Utilizza il comando [update-user-pool-client](#). Specificare il nome `userPoolId` del `poolClientId` di utenti, il client del pool di utenti e la regione. Per `--allowed-o-auth-flows` esempio, specifica `client_credentials` di indicare che il client deve ottenere i token di accesso dall'endpoint del token utilizzando una combinazione di un ID client e un segreto del client. Per `--allowed-o-auth-scopes`, specificare l'identificatore del server di risorse e il nome dell'ambito come segue: `resource_server_identifier/scope_name`. `--allowed-o-auth-flows-user-pool-client` includi per indicare che il client è autorizzato a seguire il protocollo OAuth durante l'interazione con i pool di utenti di 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
```

Output di esempio

```
{
  "UserPoolClient": {
    "UserPoolId": "us-east-1_QLEXAMPLE",
    "ClientName": "dcv-session-manager-client-app",
    "ClientId": "219273hp6k2ut5cugg9EXAMPLE",
    "ClientSecret": "1vp5e8nec7cbf4m9me55mbmht91u61h1h0a78rq1qki11EXAMPLE",
    "LastModifiedDate": 1602512103.099,
```



```
$ 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
```

Gli sviluppatori ora possono utilizzare il server per richiedere i token di accesso. Quando richiedi i token di accesso, fornisci l'ID client, il segreto del client e l'URL del server generati qui. Per ulteriori informazioni sulla richiesta di token di accesso, consulta [Creare, ottenere un token di accesso ed effettuare una richiesta API](#) nella Guida per gli sviluppatori di NICE DCV Session Manager.

Configurazione della persistenza del broker

I broker Session Manager supportano l'integrazione con database esterni. Il database esterno consente a Session Manager di conservare i dati e le chiavi di stato in modo che siano disponibili in seguito. In effetti, i dati del broker sono distribuiti sul cluster, rendendoli suscettibili di perdita di dati se un host deve riavviarsi o un cluster viene terminato. Con questa funzionalità abilitata, puoi aggiungere e rimuovere nodi del broker. Inoltre, è possibile arrestare un cluster e riavviarlo, senza la necessità di rigenerare le chiavi o perdere informazioni su quale server NICE DCV è aperto o chiuso.

È possibile impostare i seguenti tipi di informazioni in modo che persistano:

- Tasti per configurare le sessioni per stabilire una connessione con i client
- Dati delle sessioni in volo
- Stato del server NICE DCV

NICE DCV Session Manager supporta i database DynamoDB, MariaDB e MySQL. È necessario configurare e gestire uno di questi database per utilizzare questa funzionalità. Se le macchine del tuo broker sono ospitate su Amazon EC2, ti consigliamo di utilizzare DynamoDB come database esterno, poiché non richiede alcuna configurazione aggiuntiva.

Note

L'esecuzione di un database esterno può comportare costi aggiuntivi. Per informazioni sui prezzi di DynamoDB, vedi [Prezzi per la capacità fornita](#).

Configura il broker in modo che resti su DynamoDB

Configura i broker per iniziare a archiviare i loro dati su DynamoDB:

1. Aprire il file `/etc/dcv-session-manager-broker/session-manager-broker.properties` utilizzando qualsiasi editor di testo e apportare le seguenti modifiche:
 - Imposta `enable-persistence = true`
 - Imposta `persistence-db = dynamodb`
 - Per `dynamodb-region` specificare la regione &aws; in cui desideri archiviare le tabelle contenenti i dati del broker. Per l'elenco delle regioni supportate, consulta gli [endpoint del servizio DynamoDB](#).
 - Per `dynamodb-table-rcu` specificare la quantità di unità di capacità di lettura (RCU) supportate da ogni tabella. Per ulteriori informazioni su RCU, vedi [Capacità fornita da DynamoDB](#).
 - Per `dynamodb-table-wcu` specificare la quantità di unità di capacità di scrittura (WCU) supportate da ogni tabella. Per maggiori informazioni sulla WCU, vedi [Capacità fornita da DynamoDB](#).
 - Per `dynamodb-table-name-prefix` specificare il prefisso che viene aggiunto a ciascuna tabella DynamoDB (utile per distinguere più cluster di broker utilizzando lo stesso account). Sono consentiti solo caratteri alfanumerici, punti, trattini e trattini bassi.
2. Blocca tutti i gestori di credito nel cluster. Per ogni broker, esegui il comando riportato di seguito:

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

3. Assicurati che tutti i broker del cluster siano bloccati, quindi riavviali tutti. Avviare ogni broker eseguendo il comando riportato di seguito:

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

L'host del broker deve disporre dell'autorizzazione per chiamare le API di DynamoDB. Nelle istanze Amazon EC2, le credenziali vengono recuperate automaticamente utilizzando il servizio di metadati Amazon EC2. Se è necessario specificare credenziali diverse, è possibile impostarle utilizzando una delle tecniche di recupero delle credenziali supportate (come le proprietà di sistema Java o le variabili di ambiente). Per ulteriori informazioni, consulta [Fornitura e recupero delle credenziali &aws;](#).

Configura il broker per persistere su MariaDB/MySQL

Note

Il file `/etc/dcv-session-manager-broker/session-manager-broker.properties` contiene dati sensibili. Per impostazione predefinita, il suo accesso in scrittura è limitato a root e il suo accesso in lettura è limitato a root e all'utente che esegue il Broker. Per impostazione predefinita, questo è l'utente `dcvsmbroker`. Il Broker verifica all'avvio che il file disponga delle autorizzazioni previste.

Configura i broker per iniziare a conservare i loro dati su MariaDB/MySQL:

1. Aprire `/etc/dcv-session-manager-broker/session-manager-broker.properties` il file utilizzando qualsiasi editor di testo e apportare le seguenti modifiche:

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

In questa configurazione, `<db_endpoint>` è l'endpoint del database, `<db_port>` è la porta del database e `<db_name>` è il nome del database.

- Per `jdbc-user` specificare il nome dell'utente che ha accesso al database.
 - Per `jdbc-password` specificare la password dell'utente che ha accesso al database.
2. Blocca tutti i gestori di credito nel cluster. Per ogni broker, esegui il comando riportato di seguito:

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

3. Assicurati che tutti i broker del cluster siano bloccati, quindi riavviali tutti. Per ogni broker, esegui il comando riportato di seguito:

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

Integrazione con il gateway di connessione NICE DCV

[NICE DCV Connection Gateway](#) è un pacchetto software installabile che consente agli utenti di accedere a una flotta di server NICE DCV tramite un unico punto di accesso a una LAN o VPC.

Se la tua infrastruttura include server NICE DCV accessibili tramite il gateway di connessione NICE DCV, puoi configurare il Session Manager per integrare il gateway di connessione NICE DCV. Seguendo i passaggi descritti nella sezione seguente, il broker fungerà da [Session Resolver](#) per il Connection Gateway. In altre parole, il broker esporrà un endpoint HTTP aggiuntivo. Il Connection Gateway effettuerà chiamate API all'endpoint per recuperare le informazioni necessarie per indirizzare le connessioni NICE DCV all'host selezionato dal broker.

Configurare il broker Session Manager come risolutore di sessione per il gateway di connessione NICE DCV

Session Manager, lato Session

1. Aprire il file `/etc/dcv-session-manager-broker/session-manager-broker.properties` utilizzando qualsiasi editor di testo e aggiungere il seguente script:
 - Imposta `enable-gateway = true`
 - `gateway-to-broker-connector-https-port` Impostato su una porta TCP libera (l'impostazione predefinita è 8447)
 - Impostato sull'indirizzo IP dell'host `gateway-to-broker-connector-bind-host` a cui si associa il Broker per le connessioni NICE DCV Connection Gateway (l'impostazione predefinita è 0.0.0.0)
2. Quindi esegui i seguenti comandi per interrompere e riavviare il Broker:

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

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

3. Recupera una copia del certificato autofirmato del Broker e inseriscilo nella tua directory utente.

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

Ne avrai bisogno quando installerai il gateway di connessione NICE DCV nel passaggio successivo.

Connessione NICE DCV (lato gateway)

- Segui la [sezione nella](#) documentazione del gateway di connessione NICE DCV.

Poiché il NICE DCV Connection Gateway effettua chiamate API HTTP al broker, se il broker utilizza un certificato autofirmato, sarà necessario copiare il certificato del broker sull'host NICE DCV Connection Gateway (recuperato nel passaggio precedente) e impostare il `ca-file` parametro nella `[resolver]` sezione della configurazione del NICE DCV Connection Gateway.

Facoltativo: abilita l'autenticazione del client TLS

Una volta completato il passaggio precedente, Session Manager e Connection Gateway possono comunicare su un canale sicuro, dove il Connection Gateway può verificare l'identità dei broker di Session Manager. Se è necessario che anche i broker di Session Manager convalidino l'identità del gateway di connessione prima di stabilire il canale sicuro, è necessario abilitare la funzionalità di autenticazione del client TLS, seguendo i passaggi nella sezione successiva.

Note

Se Session Manager utilizza un sistema di bilanciamento del carico, l'autenticazione del client TLS non può essere abilitata con sistemi di bilanciamento del carico con terminazione della connessione TLS, come Application Load Balancers (ALB) o Gateway Load Balancers (GLB). Possono essere supportati solo i sistemi di bilanciamento del carico senza terminazione TLS, come i Network Load Balancer (NLB). Se utilizzi ALB o GLB, puoi far sì che solo gruppi di sicurezza specifici possano contattare i sistemi di bilanciamento del carico, garantendo un livello di sicurezza aggiuntivo; maggiori informazioni sui gruppi di sicurezza qui: [Gruppi di sicurezza per il tuo VPC](#)

Session Manager, lato Session

1. Per abilitare l'autenticazione del client TLS per la comunicazione tra i broker di Session Manager e il gateway di connessione NICE DCV, segui i passaggi seguenti:

2. Genera le chiavi e i certificati richiesti eseguendo: L'output del comando ti indicherà la cartella in cui sono state generate le credenziali e la password utilizzata per creare il TrustStore file.

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

3. Inserisci una copia della chiave privata e del certificato autofirmato del NICE DCV Connection Gateway nella tua directory utente. Ne avrai bisogno quando abiliti l'autenticazione del client TLS nel gateway di connessione NICE DCV nel passaggio successivo.

```
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. Aprire il file `dcv-session-manager-broker/etc/session-manager-broker.properties` utilizzando qualsiasi editor di testo e fare il seguente script:
 - `Impostatoenable-tls-client-auth-gateway` su `true`
 - `gateway-to-broker-connector-trust-store-file` Imposta il percorso del TrustStore file creato nel passaggio precedente
 - `gateway-to-broker-connector-trust-store-pass` Imposta la password utilizzata per creare il TrustStore file nel passaggio precedente
5. Quindi, esegui il comando riportato di seguito per arrestare e riavviare il Broker:

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

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

Connessione NICE DCV (lato gateway)

- Segui la [sezione nella](#) documentazione del gateway di connessione NICE DCV.
 - usa il percorso completo del file del certificato che hai copiato nel passaggio precedente quando imposti il `cert-file` parametro nella `[resolver]` sezione
 - utilizzate il percorso completo del file chiave che avete copiato nel passaggio precedente quando impostate il `cert-key-file` parametro nella `[resolver]` sezione

NICE DCV Session Manager Server NICE DCV - Mappatura DNS

Il gateway di connessione NICE DCV richiede i nomi DNS dei server NICE DCV per connettersi alle istanze del server DCV. Questa sezione illustra come definire un file JSON contenente la mappatura tra ogni server DCV e il nome DNS associato.

Struttura dei file

La mappatura è costituita da un elenco di oggetti JSON con i seguenti campi:

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

Dove:

ServerIdType:

Identifica il tipo di id a cui si riferisce il valore; attualmente i valori disponibili sono `agentServerId` `IPAddress` e `instanceId`:

Ip:

Disponibile sia per Amazon EC2 che per infrastrutture locali; può essere recuperato rapidamente dagli amministratori di sistema con un comando `ifconfig` (Linux) o `ipconfig` (Windows). Queste informazioni sono disponibili anche nella risposta dell' `DescribeServers` API.

Id:

Disponibile sia per Amazon EC2 che per infrastrutture locali, il Session Manager Agent crea un nuovo UUID ogni volta che il nome host o l'indirizzo IP cambiano. Queste informazioni sono disponibili nella risposta dell' `DescribeServers` API.

Host.Aws.Ec2InstanceId:

Disponibile solo per le istanze Amazon EC2, identifica in modo univoco una macchina e non cambia dopo il riavvio di un'istanza. Può essere recuperato sull'host contattando <http://169.254.169.254/latest/meta-data/instance-id>. Queste informazioni sono disponibili anche nella risposta dell' DescribeServers API.

ServerId:

Un ID del tipo specificato che identifica in modo univoco ogni server NICE DCV della rete.

DnsNames:

L'oggetto contenente i nomi DNS associati al server NICE DCV, questo oggetto conterrà:

InternalDnsNames:

Il nome DNS utilizzato dal gateway di connessione NICE DCV per connettersi all'istanza.

Utilizza i comandi CLI di Session Manager `brokerregister-server-dns-mapping` per caricare la mappatura da un file (riferimento alla pagina di comando: [register-server-dns-mapping](#)) ed `describe-server-dns-mappings` per elencare le mappature attualmente caricate nel Session Manager Broker (riferimento alla pagina dei comandi: [describe-server-dns-mappings](#)).

Persistenza

Ti consigliamo vivamente di abilitare la funzione di persistenza di Session Manager Broker, per proteggerti dalla perdita della mappatura quando più broker o l'intero cluster falliscono. Per ulteriori informazioni sull'attivazione della persistenza dei dati, consulta [Configura la persistenza del broker](#)

Integrazione con Amazon CloudWatch

Session Manager supporta l'integrazione con Amazon CloudWatch for Brokers in esecuzione su istanze Amazon EC2 e anche con i broker che funzionano su host locali.

Amazon CloudWatch monitora le risorse Amazon Web Services (AWS) e le applicazioni che esegui su AWS in tempo reale. Si possono utilizzare CloudWatch per raccogliere e tenere traccia dei parametri, che sono delle variabili che si possono misurare per le risorse e le applicazioni. Per ulteriori informazioni, consulta [la Guida per CloudWatch l'utente di Amazon](#).

Puoi configurare il Session Manager Broker per inviare i seguenti dati metrici ad Amazon CloudWatch:

- `Number of DCV servers`—Il numero di server DCV gestiti dal Broker.
- `Number of ready DCV servers`—Il numero di server DCV che si trovano nello `READY` stato gestito dal Broker.
- `Number of DCV sessions`—Il numero di sessioni DCV gestite dal Broker.
- `Number of DCV console sessions`—Il numero di sessioni della console DCV gestite dal Broker.
- `Number of DCV virtual sessions`—Il numero di sessioni virtuali DCV gestite dal Broker.
- `Heap memory used`—La quantità di memoria heap allocata al Broker.
- `Off-heap memory used`—La quantità di memoria off-heap utilizzata dal Broker.
- `Describe sessions request time`—Il tempo impiegato per completare le richieste `DescribeSessions` API.
- `Delete sessions request time`—Il tempo impiegato per completare le richieste `DeleteSessions` API.
- `Create sessions request time`—Il tempo impiegato per completare le richieste `CreateSessions` API.
- `Get session connection data request time`—Il tempo impiegato per completare le richieste `GetSessionConnectionData` API.
- `Update session permissions request time`—Il tempo impiegato per completare le richieste `UpdateSessionPermissions` API.

Per configurare il Broker per l'invio di dati metrici ad Amazon CloudWatch

1. Aprire il file `/etc/dcv-session-manager-broker/session-manager-broker.properties` utilizzando qualsiasi editor di testo e fare come segue script.
 - `Impostatoenable-cloud-watch-metrics` su `true`
 - `Percloud-watch-region`, specifica la regione in cui raccogliere i dati metrici.

Note

Se il Broker è in esecuzione su un'istanza Amazon EC2, questo parametro è facoltativo. La regione viene recuperata automaticamente dall'Instance Metadata Service (Instance Metadata Service). Se si esegue il Broker su un host locale, questo parametro è obbligatorio.

Aggiornamento del Session Manager NICE DCV

L'argomento seguente descrive come aggiornare il Session Manager.

Note

Si consiglia vivamente di aggiornare tutti gli agenti di Session Manager prima di aggiornare i Session Manager Brokers, per evitare problemi di incompatibilità nel caso in cui vengano introdotte nuove funzionalità.

Argomenti

- [Aggiornamento dell'agente NICE DCV Session Manager](#)
- [Aggiornamento del broker NICE DCV Session Manager](#)

Aggiornamento dell'agente NICE DCV Session Manager

Linux host

Note

Le seguenti istruzioni riguardano l'installazione dell'agente su host x86 a 64 bit. *Per installare l'agente su host ARM a 64 bit, per Amazon Linux, RHEL e Centos, sostituisci x86_64 con e per Ubuntu sostituisci amd64 con aarch64.* arm64

Per aggiornare l'agente su un host Linux

1. Esegui il comando seguente per arrestare l'agente.

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

2. Scarica il pacchetto di installazione.
 - Amazon Linux 2, RHEL 7.x e 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 e 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. Installare il pacchetto .

- Amazon Linux 2, RHEL 7.x e 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 e 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. Esegui il seguente comando per avviare l'agente.

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

Windows host

Per aggiornare l'agente su un host Windows

1. Interrompere il servizio Agent. Esegui i seguenti comandi al prompt dei comandi.

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

2. Scaricate il programma di [installazione dell'agente](#).
3. Eseguire il programma di installazione. Nella schermata iniziale, selezionare Next (Successivo).
4. Nella schermata EULA, leggi attentamente il contratto di licenza e, se sei d'accordo, seleziona Accetto i termini e scegli Avanti.
5. Per iniziare l'installazione, scegli Installa.
6. Riavvia il servizio Agent. Esegui i seguenti comandi al prompt dei comandi.

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

Aggiornamento del broker NICE DCV Session Manager

Per aggiornare il Broker

1. Connettiti all'host su cui intendi aggiornare il Broker.
2. Interrompi il servizio Broker.

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

3. Scarica il pacchetto di installazione.

- Amazon Linux 2, RHEL 7.x e 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 e 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. Installare il pacchetto .

- Amazon Linux 2, RHEL 7.x e 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 e 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. Avvia il servizio Broker e assicurati che si avvii automaticamente a ogni avvio dell'istanza.

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

Riferimento CLI del broker

Questa sezione spiega come utilizzare i comandi dell'interfaccia a riga di comando (CLI) del Broker.

Utilizza i seguenti comandi se utilizzi un server di autenticazione esterno per generare token di accesso OAuth 2.0:

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

Usa i seguenti comandi se usi il Session Manager Broker come server di autenticazione OAuth 2.0.

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

Utilizzate i seguenti comandi per gestire il Session Manager Agent.

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

Usa i seguenti comandi per gestire il server DCV - file di mappatura dei nomi DNS.

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

register-auth-server

Registra un server di autenticazione esterno da utilizzare con il Broker.

Per impostazione predefinita, Session Manager utilizza il Broker come server di autenticazione per generare token di accesso OAuth 2.0. Se si utilizza il Broker come server di autenticazione, non è richiesta alcuna configurazione aggiuntiva.

Tuttavia, se scegli di utilizzare un server di autenticazione esterno, come Active Directory o Amazon Cognito, devi utilizzare questo comando per registrare il server di autenticazione esterno.

Argomenti

- [Sintassi](#)
- [Opzioni](#)
- [Esempio](#)

Sintassi

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

Opzioni

--url

L'URL del server di autenticazione esterno da utilizzare. È necessario aggiungere `.well-known/jwks.json` all'URL del server di autenticazione.

Tipo: String

Campo obbligatorio: sì

Esempio

L'esempio seguente registra un server di autenticazione esterno con un URL di `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
```

Output

```
Jwk url registered.
```

list-auth-servers

Elenca i server di autenticazione esterni che sono stati registrati.

Argomenti

- [Sintassi](#)
- [Output](#)
- [Esempio](#)

Sintassi

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

Output

Urls

Gli URL dei server di autenticazione esterni registrati.

Esempio

Nell'esempio seguente sono elencati tutti i server di autenticazione esterni registrati.

Comando

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

Output

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

unregister-auth-server

Annulla la registrazione di un server di autenticazione esterno. Dopo aver annullato la registrazione di un server di autenticazione esterno, non è più possibile utilizzarlo per generare token di accesso OAuth 2.0.

Argomenti

- [Sintassi](#)
- [Opzioni](#)
- [Output](#)
- [Esempio](#)

Sintassi

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

Opzioni

--url

L'URL del server di autenticazione esterno da annullare la registrazione. È necessario aggiungere `.well-known/jwks.json` all'URL del server di autenticazione.

Tipo: String

Campo obbligatorio: sì

Output

Url

L'URL del server di autenticazione esterno non registrato.

Esempio

L'esempio seguente registra un server di autenticazione esterno con un URL di `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
```

Output

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

register-api-client

Registra un client di Session Manager con il Broker e genera credenziali client che possono essere utilizzate dal client per recuperare un token di accesso OAuth 2.0, necessario per effettuare richieste API.

Important

Assicurati di archiviare le credenziali in un luogo sicuro. Non potranno quindi essere recuperati in seguito.

Questo comando viene utilizzato solo se il Broker viene utilizzato come server di autenticazione OAuth 2.0.

Argomenti

- [Sintassi](#)
- [Opzioni](#)
- [Output](#)
- [Esempio](#)

Sintassi

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

Opzioni

--name

Un nome univoco utilizzato per identificare il client di Session Manager.

Tipo: String

Campo obbligatorio: sì

Output

client-id

L'ID client univoco che deve essere utilizzato dal client Session Manager per recuperare un token di accesso OAuth 2.0.

client-password

La password che deve essere utilizzata dal client Session Manager per recuperare un token di accesso OAuth 2.0.

Esempio

Nell'esempio seguente viene registrato un client denominato `my-sm-client`.

Comando

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

Output

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

describe-api-clients

Elenca i client di Session Manager registrati con il Broker.

Argomenti

- [Sintassi](#)
- [Output](#)
- [Esempio](#)

Sintassi

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

Output

name

Il nome univoco del client Session Manager.

id

L'ID univoco del client Session Manager.

active

Indica lo stato del client Session Manager. Se il client è attivo, il valore è `true`; in caso contrario, è `false`.

Esempio

L'esempio seguente elenca i client registrati di Session Manager.

Comando

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

Output

```
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

Disattiva un client registrato di Session Manager. Un client Session Manager disattivato non può più utilizzare le sue credenziali per recuperare i token di accesso OAuth 2.0.

Argomenti

- [Sintassi](#)
- [Opzioni](#)
- [Esempio](#)

Sintassi

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

Opzioni

--client -id

L'ID client del client Session Manager da disattivare.

Tipo: String

Campo obbligatorio: sì

Esempio

L'esempio seguente disattiva un client di Session Manager con un ID client dif855b54b-40d4-4769-b792-b727bEXAMPLE.

Comando

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

Output

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

renew-auth-server-api-chiave

Rinnova le chiavi pubbliche e private utilizzate dal Broker per firmare i token di accesso OAuth 2.0 che vengono venduti al client Session Manager. Se rinnovi le chiavi, devi fornire la nuova chiave privata allo sviluppatore, poiché è necessaria per effettuare richieste API.

Argomenti

- [Sintassi](#)
- [Esempio](#)

Sintassi

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

Esempio

L'esempio seguente rinnova le chiavi pubbliche e private.

Comando

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

Output

```
Keys renewed.
```

generate-software-statement

Genera una dichiarazione software.

Gli agenti devono essere registrati presso il Broker per consentire la comunicazione. Gli agenti hanno bisogno di una dichiarazione software per registrarsi presso il Broker. Dopo aver ricevuto un'istruzione software, l'agente può registrarsi automaticamente presso il Broker utilizzando il [protocollo di registrazione dinamico del client OAuth 2.0](#). Dopo che l'Agente si è registrato presso il Broker, riceve un ID cliente e un segreto del cliente che utilizza per autenticarsi con il Broker.

Il broker e l'agente ricevono e utilizzano un'istruzione software predefinita quando vengono installati per la prima volta. Puoi continuare a utilizzare l'istruzione software predefinita; in alternativa, puoi scegliere di generarne una nuova. Se si genera una nuova istruzione software, è necessario inserire l'istruzione software in un nuovo file sull'agente e quindi aggiungere il percorso del file `alagent . software_statement_path` parametro nel `alagent . conf` file. Dopo averlo fatto, interrompi e riavvia l'Agent in modo che possa utilizzare la nuova istruzione software per registrarsi presso il Broker.

Argomenti

- [Sintassi](#)
- [Output](#)
- [Esempio](#)

Sintassi

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

Output

software-statement

La dichiarazione del software.

Esempio

L'esempio seguente genera un'istruzione software.

Comando

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

Output

```
software-statement:  
ewogICJpZCIgOiAiYjc1NTVhN2QtNWl0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiYWN0aXZlIiA6IHRydWUsCi
```

describe-software-statements

Descrive le istruzioni software esistenti.

Argomenti

- [Sintassi](#)
- [Output](#)
- [Esempio](#)

Sintassi

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

Output

software-statement

La dichiarazione del software.

issued-at

La data e l'ora di generazione del software.

is-active

Lo stato corrente dell'istruzione software. `true` se l'istruzione software è attiva; altrimenti lo è `false`.

Opzioni

--software-statement

La dichiarazione del software da disattivare.

Tipo: String

Campo obbligatorio: sì

Esempio

L'esempio seguente disattiva un'istruzione software.

Comando

```
sudo -u root dcv-session-manager-broker deactivate-software-statement --software-statement EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWI0MC00TJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj
```

Output

```
Software statement  
EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWI0MC00TJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj  
deactivated
```

describe-agent-clients

Descrive gli agenti registrati presso il Broker.

Argomenti

- [Sintassi](#)
- [Output](#)
- [Esempio](#)

Sintassi

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

Output

name

Il nome dell'agente.

id

L'ID univoco dell'agente.

active

Lo stato dell'agente. `true` se l'agente è attivo; altrimenti lo è `false`.

Esempio

L'esempio seguente descrive gli agenti.

Comando

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

Output

```
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

Annullare la registrazione di un agente dal broker.

Argomenti

- [Sintassi](#)
- [Opzioni](#)
- [Esempio](#)

Sintassi

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

Opzioni

--client-id

L'ID dell'agente da annullare la registrazione.

Tipo: String

Campo obbligatorio: sì

Esempio

L'esempio seguente annulla la registrazione di un agente.

Comando

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

Output

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

register-server-dns-mappings

Registra i server DCV: mappature dei nomi DNS provenienti da un file JSON.

Sintassi

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

Opzioni

--file-path

Il percorso del file contenente i server DCV - mappature dei nomi DNS.

Tipo: String

Campo obbligatorio: sì

Esempio

L'esempio seguente registra i server DCV - mappature dei nomi DNS dal file /tmp/mappings.json.

Comando

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

Output

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

describe-server-dns-mappings

Descrivi i server DCV - mappature dei nomi DNS attualmente disponibili.

Sintassi

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

Output

serverIdType

Il tipo di ID del server.

serverId

L'ID univoco del server.

dnsNames

I nomi DNS interni ed esterni

internalDnsNames

I nomi DNS interni

externalDnsNames

I nomi DNS esterni

Esempio

Nell'esempio seguente sono elencati i server DCV registrati - mappature dei nomi DNS.

Comando

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

Output

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

```
"dnsNames" : {  
  "internalDnsName" : "internal2",  
  "externalDnsName" : "external2"  
}  
}  
]
```

Riferimento al file di configurazione

In questa sezione vengono fornite informazioni sui file di configurazione dell'agente e del broker.

Argomenti

- [File di configurazione del broker](#)
- [File di configurazione dell'agente](#)

File di configurazione del broker

Il file di configurazione del Broker (`/etc/dcv-session-manager-broker/session-manager-broker.properties`) include parametri che possono essere configurati per personalizzare la funzionalità di Session Manager. Puoi modificare il file di configurazione utilizzando l'editor di testo preferito.

Note

Il file `/etc/dcv-session-manager-broker/session-manager-broker.properties` contiene dati sensibili. Per impostazione predefinita, il suo accesso in scrittura è limitato a root e il suo accesso in lettura è limitato a root e all'utente che esegue il Broker. Per impostazione predefinita, questo è `dcvsmbrokerutente`. Il Broker verifica all'avvio che il file disponga delle autorizzazioni previste.

Nella tabella seguente sono elencati i parametri nel file di configurazione Broker.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
<code>broker.java-home</code>	No		Specifica il percorso della home directory Java che il Broker utilizzerà al posto di quello predefinito del sistema. Se impostato, il Broker lo utilizzerà.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
			<p>à<broker-java-home>/bin/java all'avvio.</p> <p>Suggerimento: il Broker richiede Java Runtime Environment 11 e viene installato se manca come dipendenza dalla corretta installazione. Se la versione 11 non è impostata come ambiente Java predefinito, la relativa home directory può essere acquisita utilizzando il seguente comando:</p> <pre data-bbox="1117 1031 1507 1150">\$ sudo alternatives --display java</pre>
session-screenshot-max-width	No	160	<p>Specifica la larghezza massima, in pixel, degli screenshot delle sessioni acquisiti utilizzando l'GetSessionScreenshotsAPI.</p>

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
session-screenshot-max-height	No	100	Specifica l'altezza massima, in pixel, degli screenshot delle sessioni acquisiti utilizzando l'GetSessionScreenshotsAPI.
session-screenshot-format	No	png	Il formato del file di immagine degli screenshot delle sessioni acquisiti utilizzando l'GetSessionScreenshotsAPI.
create-sessions-queue-max-size	No	1000	Il numero massimo di richieste CreateSessionsAPI non soddisfatte che possono essere messe in coda. Quando la coda è piena, le nuove richieste non soddisfatte vengono rifiutate.
create-sessions-queue-max-time-seconds	No	1800	Il tempo massimo, espresso in secondi, in cui una richiesta di verifica preliminare dell>CreateSessionsAPI non soddisfatta può rimanere in coda. Se la richiesta non può essere soddisfatta entro il periodo di tempo specificato, fallisce.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
session-manager-working-path	Sì	/tmp	Specifica il percorso della directory in cui il Broker scrive i file necessari per operare. Questa directory deve essere accessibile solo al Broker.
enable-authentication-server	Sì	true	Specifica se il broker è il server di autenticazione utilizzato per generare token di accesso OAuth 2.0 per le API client.
enable-authentication	Sì	true	Abilita o disabilita l'autorizzazione del client. Se si abilita l'autorizzazione del client, l'API del client deve fornire un token di accesso quando si effettuano richieste API. Se disabiliti l'autorizzazione del client, le API client possono effettuare richieste senza token di accesso.
enable-agent-authorization	Sì	true	Abilita o disabilita l'autorizzazione dell'agente. Se abiliti l'autorizzazione dell'agente, l'agente deve fornire un token di accesso quando comunica con il Broker.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
delete-session-duration-hours	No	1	Specifica il numero di ore dopo le quali le sessioni eliminate diventano invisibili e non vengono più restituite dalle chiamate DescribeSession API.
connect-session-token-duration-minutes	No	60	Specifica il numero di minuti per i quali il ConnectSession token rimane valido.
client-to-broker-connect-https-port	Sì	8443	Specifica la porta HTTPS in cui il Broker ascolta le connessioni client.
client-to-broker-connect-bind-host	No	0.0.0.0	Specifica l'indirizzo IP dell'host a cui il broker si lega per le connessioni client.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
client-to-broker-connect-key-store-file	Sì		Specifica l'archivio chiavi utilizzato per le connessioni client TLS.
client-to-broker-connect-key-store-pass	Sì		Specifica il pass per l'archiviazione delle chiavi.
agent-to-broker-connect-https-port	Sì	8445	Specifica la porta HTTPS in cui il Broker ascolta le connessioni dell'agente.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
agent-to-broker-connecto b ind- host	No	0.0.0.0	Specifica l'indirizzo IP dell'host a cui il broker si lega alle connessioni dell'agente.
agent-to-broker-connecto key- store- file	Sì		Specifica l'archivio chiavi utilizzato per le connessioni dell'agente TLS.
agent-to-broker-connecto key- store- pass	Sì		Specifica il pass per l'archiviazione delle chiavi.
broker-to-broker- port	Sì	47100	Specifica la porta utilizzata per le connessioni da broker a broker.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
broker-to-broker-broadcast-ind-host	No	0.0.0.0	Specifica l'indirizzo IP dell'host a cui si collega il broker per le connessioni da broker a broker.
broker-to-broker-discovery-port	Sì	47500	Specifica la porta utilizzata dai broker per scoprirsi a vicenda.
broker-to-broker-discovery-address	No		Specifica gli indirizzi IP e le porte degli altri broker della flotta nel formato <i>ip_address : port</i> . Se sono presenti più broker, separa i valori con una virgola. Se si specifica <code>broker-to-broker-discovery-multicast-group</code> , <code>broker-to-broker-discovery-multicast-port</code> , <code>broker-to-broker-discovery-AWS-region</code> o <code>broker-to-broker-discovery-AWS-alb-target-group-arn</code> , si omette questo parametro.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
broker-to-broker-discovery-multicast-group	No		Specifica il gruppo multicast per l'broker-to-roker individuazione B. Se si specifica broker-to-broker-discovery-addresses-obroker-to-broker-discovery-AWS-alb-target-group-arn si omette questo parametro.broker-to-broker-discovery-aws-region
broker-to-broker-discovery-multicast-port	No		Specifica la porta multicast per il broker-to-roker rilevamento B. Se si specificabroker-to-broker-discovery-addresses-obroker-to-broker-discovery-AWS-alb-target-group-arn si omette questo parametro .broker-to-broker-discovery-AWS-region

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
broker-to-broker-discovery-AWS-region	No		Specifica laAWS regione del sistema di bilanciamento del carico dell'applicazione utilizzato per l'individuazione da Broker a Broker. Se si specificabroker-to-broker-discovery-multicast-group obroker-to-broker-discovery-addresses si omette questo parametro.broker-to-broker-discovery-multicast-port
broker-to-broker-discovery-AWS-alb-target-group-arn	No		L'ARN dell'utente del gruppo target dell'applicazione load balancer per l'individuazione da broker a broker. Se si specificabroker-to-broker-discovery-multicast-group obroker-to-broker-discovery-addresses si omette questo parametro.broker-to-broker-discovery-multicast-port

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
broker-to-broker-distributed-memory-max-size-mb	No	4096	Specifica la quantità massima di memoria off-heap che può essere utilizzata da un singolo broker per archiviare i dati della sessione NICE DCV.
broker-to-broker-key-store-file	Sì		Specifica l'archivio chiavi utilizzato per le connessioni TLS Broker.
broker-to-broker-key-store-pass	Sì		Specifica il pass per l'archiviazione delle chiavi.
enable-cloud-watch-metrics	No	false	Abilita o disabilita le CloudWatch metriche Amazon. Se abiliti le CloudWatch metriche, potresti dover specificare un valore per <code>cloud-watch-region</code> .

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
<code>cloud-watch-region</code>	No	Necessario solo se <code>enable-cloud-watch-metrics</code> è impostato su <code>true</code> . Se il Broker è installato su un'istanza Amazon EC2, la regione viene recuperata dall'IMDS.	La AWS regione in cui vengono pubblicate le CloudWatch metriche.
<code>max-api-requests-per-second</code>	No	1000	Specifica il numero massimo di richieste che l'API Broker può elaborare ogni secondo prima di essere limitate.
<code>enable-throw-headers-forwarder</code>	No	<code>false</code>	Se impostato su <code>true</code> sulla limitazione, recupera l'IP del chiamante dall'intestazione X-Forwarded-For, se presente.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
create-session-number-of-retries-on-failure	No	2	Specifica il numero massimo di tentativi da eseguire dopo il fallimento di una richiesta di sessione di creazione su un host del server NICE DCV. Impostare su 0 per non ripetere mai più tentativi in caso di errore.
autorun-file-arguments-max-size	No	50	Specifica il numero massimo di argomenti che possono essere passati al file autorun.
autorun-file-arguments-max-argument-length	No	150	Specifica la lunghezza massima in caratteri di ogni argomento del file autorun.
enable-peersister	Sì	false	Se impostato su true, i dati sullo stato del broker vengono mantenuti in un database esterno.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
<code>persist- ce- db</code>	No	Necessario solo se <code>enable-persistence</code> è impostato su <code>true</code> .	Specifica quale database viene utilizzato per la persistenza. Gli unici valori supportati sono: <code>dynamodb</code> e <code>mysql</code> .
<code>dynamodb- region</code>	No	Necessario solo se <code>enable-persistence</code> è impostato su <code>true</code> ed <code>persist- ce- db</code> è impostato su <code>dynamodb</code> .	Specifica la regione in cui vengono create e accessibili le tabelle DynamoDB.
<code>dynamodb- table- rcu</code>	No	Necessario solo se <code>enable-persistence</code> è impostato su <code>true</code> ed <code>persist- ce- db</code> è impostato su <code>dynamodb</code> .	Specifica le unità di capacità di lettura (RCU) per ogni tabella DynamoDB. Per ulteriori informazioni su RCU, vedi Prezzi per la capacità fornita .
<code>dynamodb- table- wcu</code>	No	Necessario solo se <code>enable-persistence</code> è impostato su <code>true</code> ed <code>persist- ce- db</code> è impostato su <code>dynamodb</code> .	Specifica le unità di capacità di scrittura (WCU) per ogni tabella DynamoDB. Per ulteriori informazioni sulla WCU, consulta Pricing for Provisioned Capacity .

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
dynamodb-table-name-prefix	No	Obbligatorio solo se enable-persistence è impostato su true ed persistence-db è impostato su dynamodb.	Specifica il prefisso che viene aggiunto a ciascuna tabella DynamoDB (utile per distinguere più cluster di broker utilizzando lo stesso AWS account). Sono consentiti solo caratteri alfanumerici, punti, trattini e trattini bassi.
jdbc-connection-url	No	Obbligatorio solo se enable-persistence è impostato su true ed persistence-db è impostato su mysql.	<p>Specifica l'URL di connessione al database MariaDB/MySQL; contiene l'endpoint e il nome del database. L'url deve avere questo formato:</p> <pre>jdbc:mysql://<db_endpoint>:<db_port>/<db_name>?createDatabaseIfNotExist=true</pre> <p><db_endpoint> Dov'è l'endpoint del database MariaDB/MySQL, <db_port> è la porta del database e <db_name> è il nome del database.</p>

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
jdbc-user	No	Obbligatorio solo se enable-persistence è impostato su true ed persistence-db è impostato su mysql.	Specifica il nome dell'utente che ha accesso al database MariaDB/MySQL.
jdbc-password	No	Obbligatorio solo se enable-persistence è impostato su true ed persistence-db è impostato su mysql.	Specifica la password dell'utente che ha accesso al database MariaDB/MySQL.
seconds-before-deleting-unreachable-agent-server	No	1800	Specifica la quantità di secondi dopo i quali un server non raggiungibile viene eliminato dal sistema.

File di configurazione dell'agente

Il file di configurazione dell'agente (/etc/dcv-session-manager-agent/agent.conf per Linux e C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf per Windows) include parametri che possono essere configurati per personalizzare la funzionalità di Session Manager. Puoi modificare il file di configurazione utilizzando l'editor di testo preferito.

Nella tabella seguente sono elencati i parametri nel file di configurazione dell'agente.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
<code>agent.tlscer_host</code>	Sì		Specifica il nome DNS dell'host del broker.
<code>agent.tlscer_port</code>	Sì	8445	Specifica la porta su cui comunicare con il Broker.
<code>agent.configfile</code>	No		Necessario solo se <code>setls_strict</code> è impostato su <code>true</code> . Specifica il percorso del file del certificato (.pem) necessario per convalidare il certificato TLS. Copia il certificato autofirmato dal broker all'agente.
<code>agent.configfolder</code>	No	<ul style="list-style-type: none"> <code>/var/lib/dcv-session-manager-agent/init</code> (Linux) 	Specifica il percorso di una cartella sul server host utilizzata per archiviare gli script personalizzati autorizzati a inizializzare le sessioni del server NICE DCV quando vengono create. È necessario specificare un percorso assoluto. La cartella deve essere accessibile e i file devono essere eseguibili dagli utenti che utilizzano il parametro di <code>InitFileRichiesta</code> dell' <code>CreateSessionsAPI</code> .

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
agent.1 _strict	No	true	Indica se deve essere utilizzata una convalida TLS rigorosa.
agent.s tware_s tatement_ th	No		Necessario solo se non viene utilizzata l'istruzione software predefinita. Specifica il percorso del file di istruzioni del software. Per ulteriori informazioni, consulta generate-software-statement .
agent.1 s_folde	No	<ul style="list-style-type: none"> /etc/dcv-session-manager-agent (Linux) C:\Program Files\NICE\DCVSessionManagerAgent\conf\tags (Windows) 	Specifica il percorso della cartella in cui si trovano i file di tag. Per ulteriori informazioni, consulta Utilizzo dei tag per indirizzare i server NICE DCV .

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
agent.a orun_ er	No	<ul style="list-style-type: none"> • /var/lib/dcv-session-manager/autorun (Linux) • C:\ProgramData\NICE\DcvSessionManagerAgent\autorun (Windows) 	Specifica il percorso di una cartella sul server host utilizzata per archiviare script e app che possono essere eseguiti automaticamente all'avvio della sessione. È necessario specificare un percorso assoluto. La cartella deve essere accessibile e i file devono essere eseguibili dagli utenti che utilizzano il parametro di AutorunFilerichiesta dell>CreateSessionsAPI.
agent.n _virtua sessor	No	-1 (nessun limite)	Il numero massimo di sessioni virtuali che possono essere create su un server NICE DCV utilizzando NICE DCV Session Manager.
agent.n _concu nt_ses ns_per_ er	No	1	Il numero massimo di sessioni virtuali che possono essere create su un server NICE DCV da un singolo utente utilizzando NICE DCV Session Manager.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
agent.t ker_upc e_inte l	No	30	Specifica la quantità di secondi da attendere prima di inviare dati aggiornati al Broker. I dati inviati includono lo stato del server e dell'host NICE DCV, nonché informazioni aggiornate sulla sessione. Valori più bassi rendono il Session Manager più consapevole delle modifiche che avvengono nel sistema in cui viene eseguito l'agente, ma aumentano il carico del sistema e il traffico di rete. Valori più alti riducono il carico del sistema e della rete, ma il Session Manager diventa meno reattivo alle modifiche del sistema, quindi valori superiori a 120 quelli non consigliati.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
log.level	No	info	<p>Specifica il livello di verbosità dei file di registro. Sono disponibili i seguenti livelli di dettaglio:</p> <ul style="list-style-type: none"> • <code>error</code>—Fornisce il minimo dettaglio. Include solo gli errori. • <code>warning</code>—Include errori e avvisi. • <code>info</code>—Il livello di verbosità predefinito. Include errori, avvisi e messaggi informativi. • <code>debug</code>—Fornisce il maggior numero di dettagli. Fornisce informazioni dettagliate utili per la risoluzione dei problemi.
log.directory	No	<ul style="list-style-type: none"> • <code>/var/log/dcv-session-manager-agent/</code> (Linux) • <code>C:\ProgramData\NICE\DCVSessionManagerAgent\log</code> (Windows) 	Specifica la directory in cui creare i file di log.

Nome del parametro	Obbligatorio	Valore predefinito	Descrizione
log.rotation	No	daily	<p>Specifica la rotazione del file di log. I valori validi sono:</p> <ul style="list-style-type: none"> hourly—I file di log vengono ruotati ogni ora. daily—I file di log vengono ruotati ogni giorno.
log.max-file-size	No	10485760	<p>Quando la dimensione di un file di registro raggiunge la dimensione specificata in byte, verrà ruotato. Verrà creato un nuovo file di registro e nel nuovo file verranno inseriti altri eventi di registro.</p>
log.rotate	No	9	<p>Il numero massimo di file di log conservati nella rotazione. Ogni volta che si verifica una rotazione e viene raggiunto questo numero, il file di registro più vecchio verrà eliminato.</p>

Note di rilascio e cronologia dei documenti per NICE DCV Session Manager

Questa pagina fornisce le note di rilascio e la cronologia dei documenti per NICE DCV Session Manager.

Argomenti

- [Note di rilascio di NICE DCV Session Manager](#)
- [Cronologia dei documenti](#)

Note di rilascio di NICE DCV Session Manager

Questa sezione fornisce una panoramica dei principali aggiornamenti, versioni di funzionalità e correzioni di bug per NICE DCV Session Manager. Tutti gli aggiornamenti sono organizzati per data di rilascio. Aggiorniamo frequentemente la documentazione per rispondere al feedback che ci inviate.

Argomenti

- [2023.1-16388— 26 giugno 2024](#)
- [2023.1 — 9 novembre 2023](#)
- [2023.0-15065— 4 maggio 2023](#)
- [2023.0-14852— 28 marzo 2023](#)
- [2022.2-13907 — 11 novembre 2022](#)
- [2022.1-13067— 29 giugno 2022](#)
- [2022.0-11952— 23 febbraio 2022](#)
- [2021.3-11591— 20 dicembre 2021](#)
- [2021.2-11445— 18 novembre 2021](#)
- [2021.2-11190— 11 ottobre 2021](#)
- [2021.2-11042— 01 settembre 2021](#)
- [2021.1-10557— 31 maggio 2021](#)
- [2021.0-10242 — 12 aprile 2021](#)
- [2020.2-9662— 04 dicembre 2020](#)
- [2020.2-9508— 11 novembre 2020](#)

2023.1-16388— 26 giugno 2024

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none">• Intermediario: 417• Agente: 748• CLI: 140	<ul style="list-style-type: none">• Risolto un bug che mostrava erroneamente la memoria come TB, non GB.• Correzione dei bug e miglioramenti delle prestazioni.

2023.1 — 9 novembre 2023

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none">• Intermediario: 410• Agente: 732• CLI: 140	<ul style="list-style-type: none">• Correzioni di bug e miglioramenti delle prestazioni

2023.0-15065— 4 maggio 2023

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none">• Intermediario: 392• Agente: 675• CLI: 132	<ul style="list-style-type: none">• È stato aggiunto il supporto per Red Hat Enterprise Linux 9, Rocky Linux 9 e CentOS Stream 9 su piattaforme ARM.

2023.0-14852— 28 marzo 2023

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none">• Intermediario: 392• Agente: 642• CLI: 132	<ul style="list-style-type: none">• È stato aggiunto il supporto per Red Hat Enterprise Linux 9, Rocky Linux 9 e CentOS Stream 9.

2022.2-13907 — 11 novembre 2022

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none">• Intermediario: 382• Agente: 612• CLI: 123	<ul style="list-style-type: none">• Aggiunto un Substate campo in DescribeSessions risposta.• È stato risolto un problema che poteva impedire alla CLI di connettersi al broker a seconda dell'URL in uso.

2022.1-13067— 29 giugno 2022

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none">• Intermediario: 355• Agente: 592• CLI: 114	<ul style="list-style-type: none">• Aggiunto il supporto per eseguire il broker su istanze AWS Graviton.• Aggiunto il supporto per agenti e broker per Ubuntu 22.04.

2022.0-11952— 23 febbraio 2022

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none">• Intermediario: 341• Agente: 520• CLI: 112	<ul style="list-style-type: none">• Aggiunta la funzionalità di rotazione dei log all'agente.• È stato aggiunto un parametro di configurazione per impostare la home page di Java nel Broker.• Migliore trasferimento dei dati dalla cache al disco nel Broker.• Convalida URL fissa nella CLI.

2021.3-11591— 20 dicembre 2021

Numeri build	Nuove funzionalità
<ul style="list-style-type: none"> • Intermediario: 307 • Agente: 453 • CLI: 92 	<ul style="list-style-type: none"> • È stato aggiunto il supporto per l'integrazione con il gateway di connessione NICE DCV. • È stato aggiunto il supporto Broker per Ubuntu 18.04 e Ubuntu 20.04.

2021.2-11445— 18 novembre 2021

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none"> • Intermediario: 288 • Agente: 413 • CLI: 54 	<ul style="list-style-type: none"> • È stato risolto un problema con la convalida dei nomi di accesso che includono un dominio Windows.

2021.2-11190— 11 ottobre 2021

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none"> • Intermediario: 254 • Agente: 413 • CLI: 54 	<ul style="list-style-type: none"> • È stato risolto un problema nell'interfaccia della riga di comando che impediva l'avvio di sessioni di Windows.

2021.2-11042— 01 settembre 2021

Numeri build	Nuove funzionalità	Modifiche e correzioni di bug
<ul style="list-style-type: none"> • Intermediario: 254 	<ul style="list-style-type: none"> • NICE DCV Session Manager ora offre il supporto dell'interfaccia a riga di comando (CLI). Puoi creare e gestire 	<ul style="list-style-type: none"> • Quando si registra un server di autorizzazione esterno, è ora possibile specificare l'algoritmo

Numeri build	Nuove funzionalità	Modifiche e correzioni di bug
<ul style="list-style-type: none"> • Agente: 413 • CLI: 37 	<ul style="list-style-type: none"> • sessioni NICE DCV nella CLI, invece di chiamare le API. • NICE DCV Session Manager ha introdotto la persistenza dei dati del Broker. Per una maggiore disponibilità, i broker possono conservare le informazioni sullo stato del server su un archivio dati esterno e ripristinare i dati all'avvio. 	<ul style="list-style-type: none"> • utilizzato dal server di autorizzazione per firmare token Web in formato JSON. Con questa modifica, puoi usare Azure AD come server di autorizzazione esterno.

2021.1-10557— 31 maggio 2021

Numeri build	Nuove funzionalità	Modifiche e correzioni di bug
<ul style="list-style-type: none"> • Intermediario: 214 • Agente: 365 	<ul style="list-style-type: none"> • NICE DCV Session Manager ha aggiunto il supporto per i parametri di input passati al file autorun su Linux. • Le proprietà del server possono ora essere passate come requisiti all'CreateSessionsAPI. 	<ul style="list-style-type: none"> • Abbiamo risolto un problema con il file autorun su Windows.

2021.0-10242 — 12 aprile 2021

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none"> • Intermediario: 183 • Agente: 318 	<ul style="list-style-type: none"> • NICE DCV Session Manager ha introdotto le seguenti nuove API: <ul style="list-style-type: none"> • OpenServers • CloseServers • DescribeServers • GetSessionScreenshots

Numeri build	Modifiche e correzioni di bug
	<ul style="list-style-type: none"> • Ha inoltre introdotto i seguenti nuovi parametri di configurazione: <ul style="list-style-type: none"> • Parametri del broker: session-screenshot-max-width session-screenshot-max-height session-screenshot-format ,create-sessions-queue-max-size ,,ecreate-sessions-queue-max-time-seconds . • Parametri dell'agente: agent.autorun_folder max_virtual_sessions ,emax_concurrent_sessions_per_user . • Parametri dell'agente: agent.autorun_folder max_virtual_sessions ,emax_concurrent_sessions_per_user . • Parametri dell'agente: agent.autorun_folder max_virtual_sessions ,emax_concurrent_sessions_per_user .

2020.2-9662— 04 dicembre 2020

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none"> • Intermediario: 114 • Agente: 211 	<ul style="list-style-type: none"> • Abbiamo risolto un problema con i certificati TLS generati automaticamente che impedivano l'avvio del Broker.

2020.2-9508— 11 novembre 2020

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none"> • Intermediario: 78 • Agente: 183 	<ul style="list-style-type: none"> • La versione iniziale di NICE DCV Session Manager.

Cronologia dei documenti

La tabella seguente descrive la documentazione per questa versione di NICE DCV Session Manager.

Modifica	Descrizione	Data
NICE DCV versione 2023.1-16388	NICE DCV Session Manager è stato aggiornato per NICE DCV 2023.1-16388. Per ulteriori informazioni, consulta 2023.1-16388— 26 giugno 2024.	26 giugno 2024
NICE DCV versione 2023.1	NICE DCV Session Manager è stato aggiornato per NICE DCV 2023.1. Per ulteriori informazioni, consulta 2023.1 — 9 novembre 2023.	9 novembre 2023
NICE DCV versione 2023.0	NICE DCV Session Manager è stato aggiornato per NICE DCV 2023.0. Per ulteriori informazioni, consulta 2023.0-14852— 28 marzo 2023.	28 marzo 2023
NICE DCV versione 2022.2	NICE DCV Session Manager è stato aggiornato per NICE DCV 2022.2. Per ulteriori informazioni, consulta 2022.2-13907 — 11 novembre 2022.	11 novembre 2022
NICE DCV versione 2022.1	NICE DCV Session Manager è stato aggiornato per NICE DCV 2022.1. Per ulteriori informazioni, consulta 2022.1-13067— 29 giugno 2022.	29 giugno 2022
NICE DCV Versione 2022.0	NICE DCV Session Manager è stato aggiornato per NICE DCV 2022.0. Per ulteriori informazioni, consulta 2022.0-11952— 23 febbraio 2022.	23 febbraio 2022
NICE DCV versione 2021.3	NICE DCV Session Manager è stato aggiornato per NICE DCV 2021.3. Per ulteriori informazioni, consulta 2021.3-11591— 20 dicembre 2021.	20 dicembre 2021

Modifica	Descrizione	Data
NICE DCV versione 2021.2	NICE DCV Session Manager è stato aggiornato per NICE DCV 2021.2. Per ulteriori informazioni, consulta 2021.2-11042— 01 settembre 2021 .	01 settembre 2021
NICE DCV versione 2021.1	NICE DCV Session Manager è stato aggiornato per NICE DCV 2021.1. Per ulteriori informazioni, consulta 2021.1-10557— 31 maggio 2021 .	31 maggio 2021
NICE DCV versione 2021.0	NICE DCV Session Manager è stato aggiornato per NICE DCV 2021.0. Per ulteriori informazioni, consulta 2021.0-10242 — 12 aprile 2021 .	12 Aprile 2021
Versione iniziale di NICE DCV Session Manager	La prima pubblicazione di questo contenuto.	11 novembre 2020

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.