



Guida per l'amministratore

Amazon DCV Session Manager



Amazon DCV Session Manager: 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	4
Prezzi	4
Requisiti	4
Requisiti di rete e connettività	5
Configurazione di Session Manager	7
Fase 1: Preparare i DCV server Amazon	7
Fase 2: Configurare il broker	8
Fase 3: configurare l'agente	11
Fase 4: Configurare il DCV server Amazon	16
Fase 5: Verificare le installazioni	17
Verifica l'agente	18
Verifica il broker	19
Configurazione del Session Manager	20
Scalabilità di Session Manager	20
Fase 1: creazione di un profilo dell'istanza	21
Fase 2: Preparare il SSL certificato per il sistema di bilanciamento del carico	22
Fase 3: Creare il sistema di bilanciamento del carico dell'applicazione Broker	23
Passaggio 4: avviare i broker	24
Fase 5: Creare il sistema di bilanciamento del carico dell'applicazione Agent	25
Passaggio 6: Avvia gli agenti	26
Utilizzo dei tag sui DCV server Amazon	27
Configurazione di un server di autorizzazione esterno	29
Configurazione della persistenza del broker	34
Configurare il broker per persistere su DynamoDB	34
Configura il broker per persistere su MariaDB/my SQL	35
Integrazione con Amazon DCV Connection Gateway	36
Configura Session Manager Broker come risolutore di sessione per Amazon DCV Connection Gateway	37
Facoltativo: abilita TLS l'autenticazione del client	38
DCVServer Amazon: riferimento alla DNS mappatura	40
Integrazione con Amazon CloudWatch	41

Aggiornamento del Session Manager	44
Aggiornamento dell'agente Amazon DCV Session Manager	44
Aggiornamento del broker Amazon DCV Session Manager	47
CLIRiferimento del broker	50
register-auth-server	51
Sintassi	51
Opzioni	51
Esempio	51
list-auth-servers	52
Sintassi	51
Output	52
Esempio	51
unregister-auth-server	53
Sintassi	51
Opzioni	51
Output	52
Esempio	51
register-api-client	54
Sintassi	51
Opzioni	51
Output	52
Esempio	51
describe-api-clients	56
Sintassi	51
Output	52
Esempio	51
unregister-api-client	57
Sintassi	51
Opzioni	51
Esempio	51
renew-auth-server-api-chiave	58
Sintassi	51
Esempio	51
generate-software-statement	59
Sintassi	51
Output	52

Esempio	51
describe-software-statements	60
Sintassi	51
Output	52
Esempio	51
deactivate-software-statement	61
Sintassi	51
Opzioni	51
Esempio	51
describe-agent-clients	62
Sintassi	51
Output	52
Esempio	51
unregister-agent-client	64
Sintassi	51
Opzioni	51
Esempio	51
register-server-dns-mappings	65
Sintassi	51
Opzioni	51
Esempio	51
describe-server-dns-mappings	65
Sintassi	51
Output	52
Esempio	51
Riferimento al file di configurazione	68
File di configurazione del broker	68
file di configurazione dell'agente	83
Note di rilascio e cronologia dei documenti	90
Note di rilascio	90
2024.0-457— 1 ottobre 2024	91
2023.1-17652 — 1 agosto 2024	91
2023.1-16388— 26 giugno 2024	91
2023.1 — 9 novembre 2023	92
2023.0-15065— 4 maggio 2023	92
2023.0-14852— 28 marzo 2023	92

2022.2-13907 — 11 novembre 2022	92
2022.1-13067 — 29 giugno 2022	93
2022.0-11952— 23 febbraio 2022	93
2021.3-11591— 20 dicembre 2021	93
2021.2-11445— 18 novembre 2021	94
2021.2-11190— 11 ottobre 2021	94
2021.2-11042— 01 settembre 2021	94
2021.1-10557— 31 maggio 2021	95
2021.0-10242 — 12 aprile 2021	95
2020.2-9662— 04 dicembre 2020	96
.....	96
Cronologia dei documenti	96
.....	xcix

Cos'è Amazon DCV Session Manager?

Note

Amazon DCV era precedentemente noto come NICE DCV.

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

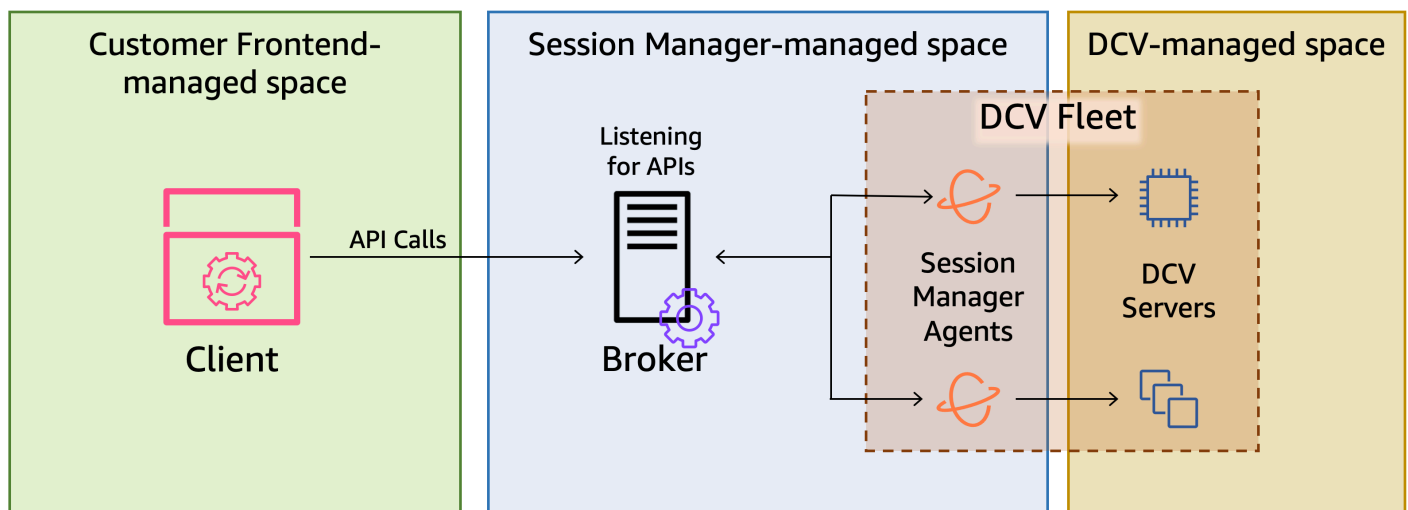
Questa guida spiega come installare e configurare l'agente e il broker di Session Manager. Per ulteriori informazioni sull'uso di Session Manager APIs, consulta la [Amazon DCV Session Manager Developer Guide](#).

Argomenti

- [Come funziona Session Manager](#)
- [Funzionalità](#)
- [Limitazioni](#)
- [Prezzi](#)
- [Requisiti DCV di Amazon 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 il Session Manager APIs. Riceve ed elabora API e richieste per gestire DCV le sessioni Amazon dal client, quindi trasmette le istruzioni agli agenti pertinenti. Il Broker deve essere installato su un host separato dai DCV server Amazon, ma deve essere accessibile al client e deve poter accedere agli agenti.

Agente

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

APIs

Session Manager espone una serie di interfacce di programmazione delle REST applicazioni (APIs) che possono essere utilizzate per gestire DCV le sessioni Amazon su una flotta di server AmazonDCV. APIs Sono ospitati ed esposti dal Broker. Gli sviluppatori possono creare client di gestione delle sessioni personalizzati che chiamano il APIs.

Cliente

Il client è l'applicazione o il portale front-end che sviluppi per chiamare il Session Manager e APIs che viene esposto dal Broker. Gli utenti finali utilizzano il client per gestire le sessioni ospitate sui DCV server Amazon della flotta.

Token di accesso

Per effettuare una API richiesta, è necessario fornire un token di accesso. I token possono essere richiesti al Broker o a un server di autorizzazione esterno dal client APIs registrato. Per richiedere e accedere al token, il client API deve fornire credenziali valide.

Cliente API

Il client API viene generato dal YAML file di API definizione di Session Manager, utilizzando Swagger Codegen. Il client API viene utilizzato per effettuare richieste. API

DCVSessione Amazon

Devi creare una DCV sessione Amazon sul tuo DCV server Amazon a cui i tuoi clienti possano connettersi. I client possono connettersi a un DCV server Amazon solo se c'è una sessione attiva. Amazon DCV supporta sessioni su console e virtuali. Utilizzi Session Manager APIs per gestire il ciclo di vita delle sessioni AmazonDCV. DCVLe sessioni Amazon 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

DCVSession Manager offre le seguenti funzionalità:

- Fornisce informazioni sulla DCV sessione Amazon: ottieni informazioni sulle sessioni in esecuzione su più DCV server Amazon.
- Gestisci il ciclo di vita di più DCV sessioni Amazon: crea o elimina più sessioni per più utenti su più DCV server Amazon con una sola richiesta. API
- Supporta i tag: utilizza tag personalizzati per indirizzare un gruppo di DCV server Amazon durante la creazione di sessioni.
- Gestisce le autorizzazioni per più DCV sessioni Amazon: modifica le autorizzazioni utente per più sessioni con una sola richiesta. API

- Fornisce informazioni sulla connessione: recupera le informazioni sulla connessione del client per le sessioni AmazonDCV.
- Supporti per il cloud e l'ambiente locale: utilizza Session Manager in locale AWS, o con server alternativi basati sul cloud.

Limitazioni

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

Prezzi

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

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

Per ulteriori informazioni, consulta [la sezione Licenze per Amazon DCV Server](#) nella Amazon DCV Administrator Guide.

Requisiti DCV di Amazon Session Manager

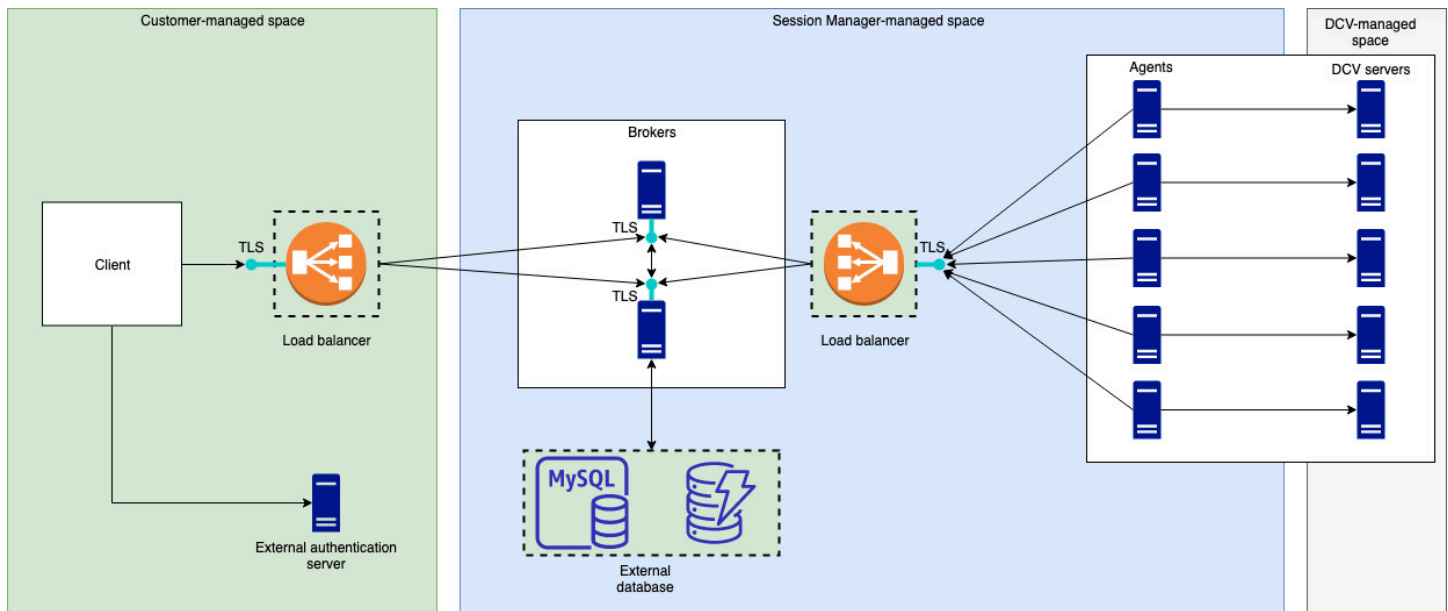
L'agente e il broker di Amazon DCV Session Manager hanno i seguenti requisiti.

	Broker	Agente
Sistema operativo	<ul style="list-style-type: none"> • Amazon Linux 2 • CentOS Stream 9 • RHEL7.6 o versione successiva • RHEL8.x • RHEL9. x 	<ul style="list-style-type: none"> • Windows <ul style="list-style-type: none"> • Windows Server 2022 • Windows Server 2019 • Windows Server 2016 • Server Linux

	Broker	Agente
	<ul style="list-style-type: none"> Rocky Linux 8.5 o versione successiva Rocky Linux 9.x Ubuntu 20.04 Ubuntu 22.04 Ubuntu 24.04 	<ul style="list-style-type: none"> Amazon Linux 2 CentOS Stream 9 RHEL8.x RHEL9. x Rocky Linux 8.5 o versione successiva Rocky Linux 9.x Ubuntu 20.04 Ubuntu 22.04 Ubuntu 24.04 SUSELinux Enterprise 12 con SP4 o versione successiva SUSELinux Enterprise 15
Architettura	<ul style="list-style-type: none"> x86 a 64 bit 64 bit ARM 	<ul style="list-style-type: none"> x86 a 64 bit 64 bit ARM (solo Amazon Linux 2, CentOS 9.x, 8.x/9.x e Rocky RHEL 8.x/9.x) 64 bit ARM (Ubuntu 22.04 e 24.04)
Memoria	8 GB	4 GB
DCVVersione Amazon	Amazon DCV 2020.2 e versioni successive	Amazon 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 DCV server Amazon. Se scegli di avere più broker per migliorare la disponibilità, devi 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, i broker e gli agenti. I Broker dovrebbero inoltre essere in grado di comunicare tra loro per scambiare informazioni sui DCV server e sulle sessioni di Amazon. I Broker possono archiviare le proprie chiavi e i dati di stato su un database esterno e rendere disponibili queste informazioni 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 My sono supportati. SQL [È possibile trovare i parametri di configurazione elencati nel file di configurazione del broker.](#)

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

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

Configurazione di Amazon DCV Session Manager

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 Amazon DCV Session Manager, procedi come segue:

Fasi

- [Fase 1: Preparare i DCV server Amazon](#)
- [Fase 2: configurare il broker Amazon DCV Session Manager](#)
- [Fase 3: configurare l'agente Amazon DCV Session Manager](#)
- [Fase 4: Configurare il DCV server Amazon per utilizzare il broker come server di autenticazione](#)
- [Passaggio 5: verifica le installazioni](#)

Fase 1: Preparare i DCV server Amazon

Devi disporre di una flotta di DCV server Amazon con cui intendi utilizzare Session Manager. Per ulteriori informazioni sull'installazione DCV dei server Amazon, consulta [Installazione del DCV server Amazon](#) nella Amazon DCV Administrator Guide.

DCVSui server Amazon Linux, Session Manager utilizza un utente del servizio locale denominato `dcvsmagent`. Questo utente viene creato automaticamente quando viene installato l'agente Session Manager. Devi concedere a questo servizio i privilegi di amministratore degli utenti del servizio per Amazon 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 DCV server Linux Amazon

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

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

3. Salva e chiudi il file.

4. Arresta e riavvia il DCV server Amazon.

Session Manager è in grado di creare DCV sessioni Amazon solo per conto di utenti già esistenti sul DCV server Amazon. Se viene effettuata una richiesta per creare una sessione per un utente che non esiste, la richiesta ha esito negativo. Pertanto, devi assicurarti che ogni utente finale previsto abbia un utente di sistema valido sul DCV server Amazon.

Tip

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

Fase 2: configurare il broker Amazon DCV Session Manager

Il broker deve essere installato su un host Linux. Per ulteriori informazioni sulle distribuzioni Linux supportate, vedere [Requisiti DCV di Amazon Session Manager](#). Installa il broker su un host separato dall'agente e dall'host del DCV server Amazon. 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 sono firmati digitalmente con una GPG firma sicura. Per consentire al gestore di pacchetti di verificare la firma del pacchetto, devi importare la DCV GPG chiave Amazon. Esegui il seguente comando per importare la DCV GPG chiave Amazon.

- Amazon Linux 2RHEL, 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

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1.el7.noarch.rpm
```

- RHEL8.x e Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1.el8.noarch.rpm
```

- CentOS 9.x, 9.x e RHEL Rocky Linux 9.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1.el9.noarch.rpm
```

- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker_2024.0.457-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker_2024.0.457-1_all.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker_2024.0.457-1_all.ubuntu2404.deb
```

4. Installare il pacchetto .

- Amazon Linux 2, RHEL 7.x

```
$ sudo yum install -y ./nice-dcv-session-manager-broker-2024.0.457-1.el7.noarch.rpm
```

- RHEL8.x e Rocky Linux 8.x

```
$ sudo yum install -y ./nice-dcv-session-manager-broker-2024.0.457-1.el8.noarch.rpm
```

- Ubuntu 20.04

```
$ sudo apt install -y ./nice-dcv-session-manager-broker_2024.0.457-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ sudo apt install -y ./nice-dcv-session-manager-broker_2024.0.457-1_all.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ sudo apt install -y ./nice-dcv-session-manager-broker_2024.0.457-1_all.ubuntu2404.deb
```

5. Verifica che la versione predefinita dell'ambiente Java sia 11

```
$ java -version
```

In caso contrario, puoi 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 il [file di configurazione del broker](#).

6. Avvia il servizio di brokeraggio 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 rubrica utenti. Ne avrai bisogno quando installerai gli agenti nel passaggio successivo.

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


Fase 3: configurare l'agente Amazon DCV Session Manager

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

Prerequisiti

Il DCV server Amazon 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 ARM host a 64 bit sostituirlo `x86_64` con `arm64`. Per Ubuntu, sostituisci `amd64` con `arm64`.

Per installare l'agente su un host Linux

1. I pacchetti sono firmati digitalmente con una GPG firma sicura. Per consentire al gestore di pacchetti di verificare la firma del pacchetto, devi importare la DCV GPG chiave Amazon. Esegui il seguente comando per importare la DCV GPG chiave Amazon.

- 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 e RHEL 7.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.el7.x86_64.rpm
```

- RHEL8.x e Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.el8.x86_64.rpm
```

- CentOS 9.x, 9.x e RHEL Rocky Linux 9.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.el9.x86_64.rpm
```

- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2404.deb
```

- SUSELinux Enterprise 12

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.sles12.x86_64.rpm
```

- SUSELinux Enterprise 15

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.sles15.x86_64.rpm
```

3. Installare il pacchetto .

- Amazon Linux 2 e RHEL 7.x

```
$ sudo yum install -y ./nice-dcv-session-manager-agent-2024.0.781-1.el7.x86_64.rpm
```

- RHEL8.x e Rocky Linux 8.x

```
$ sudo yum install -y ./nice-dcv-session-manager-agent-2024.0.781-1.el8.x86_64.rpm
```

- CentOS 9.x, 9.x e RHEL Rocky Linux 9.x

```
$ sudo yum install -y ./nice-dcv-session-manager-agent-2024.0.781-1.el9.x86_64.rpm
```

- Ubuntu 20.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2404.deb
```

- SUSELinux Enterprise 12

```
$ sudo zypper install ./nice-dcv-session-manager-agent-2024.0.781-1.sles12.x86_64.rpm
```


- SUSELinux Enterprise 15

```
$ sudo zypper install ./nice-dcv-session-manager-agent-2024.0.781-1.sles15.x86_64.rpm
```

4. Inserisci una copia del certificato autofirmato del broker (che hai copiato nel passaggio precedente) nella `/etc/dcv-session-manager-agent/` directory dell'agente.

5. Apri `/etc/dcv-session-manager-agent/agent.conf` usando il tuo editor di testo preferito e procedi come segue.

- `Perbroker_host`, specifica il DNS nome dell'host su cui è installato il broker.

 Important

Se il broker è in esecuzione su un'EC2istanza Amazon, `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 lo 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 TLS la verifica, imposta `sutls_strict.false`

6. Salva e chiudi il file.
7. 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

1. Scarica il programma di [installazione dell'agente](#).
2. Eseguire il programma di installazione. Nella schermata iniziale, selezionare Next (Successivo).
3. EULASullo schermo, 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 DNS nome dell'host su cui è installato il broker.

 Important

Se il broker è in esecuzione su un'EC2istanza Amazon, `broker_host` devi specificare l'IPv4indirizzo 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 lo 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 TLS la verifica, imposta `sutls_strict.false`

7. Salva e chiudi il file.
8. Arresta e riavvia il servizio dell'agente 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 DCV server Amazon per utilizzare il broker come server di autenticazione

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

Linux Amazon DCV server

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

1. Apri `/etc/dcv/dcv.conf` con qualsiasi editor di testo.
2. Aggiungi i `auth-token-verifier` parametri `ca-file` and alla `[security]` sezione.
 - Per `ca-file`, specifica il percorso della CA autofirmata del broker che hai copiato sull'host nel passaggio precedente.
 - Per `auth-token-verifier`, specifica il URL verificatore del token sul broker nel seguente formato: `https://broker_ip_or_dns:port/agent/validate-authentication-token` Specificate la porta utilizzata per la comunicazione broker-agente, che per impostazione predefinita è 8445. Se utilizzi il broker su un'EC2istanza Amazon, devi utilizzare l'indirizzo IP privato DNS o 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 DCV server Amazon. Per ulteriori informazioni, consulta [Stopping the Amazon DCV Server](#) e [Starting the Amazon DCV Server](#) nella Amazon DCV Administrator Guide.

Windows Amazon DCV server

Sui DCV server Windows Amazon

1. Apri l'editor del registro di Windows e vai al tasto HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/security/.
2. Apri il parametro ca-file.
3. 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

4. Aprite il auth-token-verifierparametro.
5. Per i dati Value, specifica il URL verificatore del token sul broker nel seguente formato:
`https://broker_ip_or_dns:port/agent/validate-authentication-token`
6. Specificate la porta utilizzata per la comunicazione broker-agente, che per impostazione predefinita è 8445. Se utilizzi il broker su un'EC2istanza Amazon, devi utilizzare l'indirizzo IP privato DNS o privato.

Note

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

7. Seleziona OK e chiudi l'editor del Registro di sistema di Windows.
8. Arresta e riavvia il DCV server Amazon. Per ulteriori informazioni, consulta [Stopping the Amazon DCV Server](#) e [Starting the Amazon DCV Server](#) nella Amazon DCV Administrator Guide.

Passaggio 5: verifica le installazioni

Dopo aver configurato l'agente, il broker e aver configurato entrambi sul DCV server Amazon, devi verificare che le installazioni funzionino correttamente.

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 dell'agente Linux

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 dell'agente Windows

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, controlla la presenza di errori nel file di registro.

Verifica il broker

Dopo aver installato il broker e l'agente, assicurati che il broker sia attivo 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"
}
```

Configurazione di Amazon DCV Session Manager

Per fornire un'esperienza fluida e sicura, è importante configurare correttamente Session Manager in base alle esigenze e ai requisiti dell'organizzazione. Questa sezione illustra i passaggi chiave coinvolti nella configurazione e nella configurazione del Session Manager, tra cui la gestione dell'accesso degli utenti, la configurazione delle impostazioni di rete e la personalizzazione delle impostazioni di sessione.

Argomenti

- [Gestione delle sessioni di scalabilità](#)
- [Utilizzo dei tag per indirizzare DCV i server Amazon](#)
- [Configurazione di un server di autorizzazione esterno](#)
- [Configurazione della persistenza del broker](#)
- [Integrazione con Amazon DCV Connection Gateway](#)
- [Integrazione con Amazon CloudWatch](#)

Gestione delle sessioni di scalabilità

Per garantire l'elevata disponibilità e migliorare le prestazioni, è possibile configurare Session Manager in modo che utilizzi più agenti e broker. Se intendi utilizzare più agenti e broker, ti consigliamo di installare e configurare un solo host Agent e Broker, creare Amazon Machines Images (AMI) da tali host e quindi avviare gli host rimanenti da AMIs

Per impostazione predefinita, Session Manager supporta l'uso di più agenti senza alcuna configurazione aggiuntiva. Tuttavia, se si intende utilizzare più broker, è necessario utilizzare un sistema di bilanciamento del carico per bilanciare il traffico tra il client frontend e i broker e tra i broker e gli agenti. L'installazione e la configurazione del sistema di bilanciamento del carico sono interamente di proprietà e gestite dall'utente.

La sezione seguente spiega come configurare Session Manager per utilizzare più host con un Application Load Balancer.

Fasi

- [Fase 1: creazione di un profilo dell'istanza](#)

- [Fase 2: Preparare il SSL certificato per il sistema di bilanciamento del carico](#)
- [Fase 3: Creare il sistema di bilanciamento del carico dell'applicazione Broker](#)
- [Passaggio 4: avviare i broker](#)
- [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 autorizzano l'uso di Elastic Load Balancing APIs. Per ulteriori informazioni, consulta [IAMi ruoli 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. Utilizzate 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 IAM ruolo](#) nella Guida IAM per l'utente.

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

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

Quando lo utilizzi HTTPS per il tuo load balancer listener, devi distribuire un SSL certificato sul tuo sistema di bilanciamento del carico. 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 (ACMPCA). 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 EC2 console Amazon 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 HTTPS, selezionare e per Load Balancer Port, immettere. 8443
 - d. Per VPC, seleziona la VPC sottoreti da usare, quindi seleziona tutte le sottoreti in essa contenute. VPC
 - e. Scegli Next (Successivo).
3. Per il passaggio 2: Configurazione delle 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 che hai richiesto in precedenza.
 - c. Scegli Next (Successivo).
4. Per il passaggio 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 la porta 8443.

Scegli Next (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. In Protocollo, seleziona HTTPS. Per Port (Porta), immettere 8443. Per la versione del protocollo, scegli HTTP1.
 - e. Per il controllo dello stato del protocollo, scegli HTTPS, e per Path, inserisci /health.
 - f. Scegli Next (Successivo).
6. Per la Fase 5: Registrazione degli obiettivi, scegliete Avanti.
 7. Scegli Create (Crea) .

Passaggio 4: avviare i broker

Crea un broker iniziale e configuralo per utilizzare il load balancer, creane uno AMI dal broker e quindi utilizzalo 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: configurare il broker Amazon 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`, inserisci il ARN gruppo target associato al load balancer del Broker.
 - d. Salva e chiudi il file.
3. Arresta l'istanza Broker.
4. Crea un'AMI istanza Broker interrotta. Per ulteriori informazioni, consulta [Creazione di un Linux AMI da un'istanza](#) nella Amazon EC2 User Guide for Linux Instances.

5. Usa il 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 EC2 console Amazon 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 HTTPS, selezionare e per Load Balancer Port, immettere. 8445
 - d. Per VPC, seleziona la VPC sottoreti da usare, quindi seleziona tutte le sottoreti in essa contenute. VPC
 - e. Scegli Next (Successivo).
3. Per il passaggio 2: Configurazione delle 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 che hai richiesto in precedenza.
 - c. Scegli Next (Successivo).
4. Per la Fase 3: Configurazione dei gruppi di sicurezza, creazione di un nuovo gruppo di sicurezza o selezione di un gruppo di sicurezza esistente che consenta il traffico in entrata e in uscita dagli agenti e dai broker HTTPS e sulla porta 8445.

Scegli Next (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. In Protocollo, seleziona HTTPS. Per Port (Porta), immettere 8445. Per la versione del protocollo, scegli HTTP1.
 - e. Per il controllo dello stato del protocollo, scegli HTTPS, e per Path, inserisci/health.
 - f. Scegli Next (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 Create (Crea) .

Passaggio 6: Avvia gli agenti

Crea un agente iniziale e configuralo per utilizzare il sistema di bilanciamento del carico, AMI creane uno dall'agente e quindi utilizza il 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. Prepara il DCV server Amazon. Per ulteriori informazioni, consulta [Fase 1: Preparare i DCV server Amazon](#).
2. Inserisci una copia della chiave pubblica CA creata in [Fase 2: Preparare il SSL certificato 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 Amazon DCV Session Manager](#).

Important

Quando si modifica il file di configurazione dell'agente:

- per il `broker_host` parametro, immettete Agent load balancer DNS

- per il `ca_file` parametro, immettere il percorso del file della chiave pubblica CA creato nel passaggio precedente

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

Important

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

- per il `ca-file` parametro, inserisci lo stesso percorso del file della chiave pubblica CA utilizzato nel passaggio precedente
- per il `auth-token-verifier` parametro, utilizzare Agent load balancer per DNS *broker_ip_or_dns*

5. Arresta l'istanza dell'agente.
6. Crea un'istanza Agent AMI dall'istanza dell'agente interrotta. Per ulteriori informazioni, consulta [Creazione di un Linux AMI da un'istanza](#) nella Amazon EC2 User Guide for Linux Instances.
7. Usa il AMI per avviare gli agenti rimanenti e assegna 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 su tutte le istanze dell'agente. Per ulteriori informazioni sulle porte di rete, vedere [Agent](#) Configuration File.

Utilizzo dei tag per indirizzare DCV i server Amazon

Puoi assegnare tag personalizzati agli agenti di Session Manager per identificare e classificare tali agenti e DCV i server Amazon a cui sono associati. Quando crei una nuova DCV sessione Amazon, puoi scegliere come target un gruppo di DCV server Amazon in base ai tag assegnati ai rispettivi agenti. Per ulteriori informazioni su come indirizzare DCV i server Amazon [CreateSessionRequests](#) in base ai tag Agent, consulta la Session Manager Developer Guide.

Un tag è costituito da una coppia di tag chiave-valore e puoi utilizzare qualsiasi coppia di informazioni che abbia senso per il tuo caso d'uso o ambiente. Puoi scegliere di etichettare gli agenti in base alla configurazione hardware del loro host. Ad esempio, puoi etichettare tutti gli agenti con host che dispongono di 4 GB di memoria con `ram=4GB`. Oppure puoi taggare gli agenti in

base allo scopo. Ad esempio, puoi etichettare tutti gli agenti in esecuzione su host di produzione con `purpose=production`.

Per assegnare tag a un agente

1. Usando il tuo editor di testo preferito, crea un nuovo file e assegnagli 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 del TOML file.
2. Nel file, aggiungi ogni nuova coppia di tag, chiave e valore su una nuova riga utilizzando il `key=value` formato. Per esempio:

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

3. Apri 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 specificate 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 si applicano all'agente. 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. Arresta e riavvia 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 del client e degli agenti. SDKs

Per impostazione predefinita, Session Manager utilizza il Broker come server di autorizzazione per generare token di accesso OAuth 2.0 per le istruzioni client SDKs e 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 Broker. Per ulteriori informazioni su Amazon Cognito, consulta la [Amazon Cognito Developer Guide](#).

Per utilizzare Amazon Cognito come server di autorizzazione

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

Usa il [create-user-pool](#) comando e specifica 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": "15hhd8jij74hf32f24uEXAMPLE",  
    "LastModifiedDate": 1602510048.054,  
    "CreationDate": 1602510048.054,  
    "RefreshTokenValidity": 30,  
    "AllowedOAuthFlowsUserPoolClient": false  
  }  
}
```

Prendi nota del `userPoolId`, ti servirà nel passaggio successivo.

2. Crea un nuovo dominio per il tuo pool di utenti. Usa il [create-user-pool-domain](#) comando e specifica un nome di dominio e il pool `userPoolId` di utenti che hai creato nel passaggio 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 Utilizzate il [create-user-pool-client](#) comando e specificate il pool `userPoolId` di utenti che avete creato, un nome per il client e la regione in cui crearlo. Inoltre, includi l' `--generate-secret` opzione per specificare che desideri generare un segreto per il client del pool di utenti che stai creando.

In questo caso, il nome del client 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 per quando richiedono i token di accesso per le API richieste.

4. Crea un nuovo server di risorse OAuth2 2.0 per il pool di utenti. Un server di risorse è un server di risorse ad accesso protetto. Gestisce le richieste autenticate di token di accesso.

Utilizza il [create-resource-server](#) comando e specifica il pool `userPoolId` 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 e utilizziamo `sm_scope` come nome e descrizione 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 del pool di utenti.

Utilizza il comando [update-user-pool-client](#). Specificare il pool `userPoolId` di utenti, il client `ClientId` del pool di utenti e la regione. Per `--allowed-o-auth-flows`, 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 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 il per indicare che il client è autorizzato a seguire il OAuth protocollo quando interagisce 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,
    "CreationDate": 1602510291.498,
    "RefreshTokenValidity": 30,
    "AllowedOAuthFlows": [
      "client_credentials"
    ],
    "AllowedOAuthScopes": [
      "dcv-session-manager/sm_scope"
    ],
  }
}

```

```

    "Allowed0AuthFlowsUserPoolClient": true
  }
}

```

Note

Il pool di utenti è ora pronto per fornire e autenticare i token di accesso. In questo esempio, il server di autorizzazione URL per il server di autorizzazione è. `https://cognito-idp.us-east-1.amazonaws.com/us-east-1_QLEXAMPLE/.well-known/jwks.json`

6. Verificate la configurazione.

```

$ curl -H "Authorization: Basic `echo -
n 2l9273hp6k2ut5cugg9EXAMPLE:1vp5e8nec7cbf4m9me55mbmht91u61h1h0a78rq1qki1lEXAMPLE
| base64`" -H "Content-Type: application/x-www-form-urlencoded" -X
POST "https://mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE.auth.us-
east-1.amazoncognito.com/oauth2/token?grant_type=client_credentials&scope=dcv-
session-manager/sm_scope"

```

Output di esempio

```

{
  "access_token": "eyJraWQiOiJGQ0VaRFPJUUptT3NSaW41MmtqaDdEbTZyYb0RnSTQ5b2VUT0cxUU1Q2VJPSIsImF0IjoiZm9udjTlZGScr6R0dZtId5dThkPEZiSx0YwiiWe9crAlqoazlDcCsUJHIXDtgKW64pSj3-uQQGg1Jv_tyVjhrA4JbD0k67WS2V9NW-uZ7t4zwwaUm0i3KzpBmi54fpVgPaewiVlUm_aS4LUFcWT6hVJjiZF7om7984qb2g0a14iZxpXPBJTZX_gtG9EtvnS9u",
  "expires_in": 3600,
  "token_type": "Bearer"
}

```

7. Registra il server di autorizzazione esterno per l'utilizzo con il broker utilizzando il [register-auth-server](#) comando.

```

$ 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 possono ora utilizzare il server per richiedere i token di accesso. Quando richiedi i token di accesso, fornisci l'ID client, il segreto del client e il server URL generati qui. Per ulteriori

informazioni sulla richiesta di token di accesso, consulta [Create get an access token ed effettua una API richiesta](#) nella Amazon DCV Session Manager Developer Guide.

Configurazione della persistenza del broker

I broker di Session Manager supportano l'integrazione con database esterni. Il database esterno consente a Session Manager di mantenere i dati e le chiavi di stato in modo che siano disponibili in seguito. In effetti, i dati del broker vengono distribuiti sul cluster, il che li rende suscettibili alla perdita di dati se un host deve riavviarsi o un cluster viene chiuso. Con questa funzionalità abilitata, puoi aggiungere e rimuovere i nodi del broker. Inoltre, puoi fermare un cluster e riavviarlo, senza dover rigenerare le chiavi o perdere informazioni su quali Amazon DCV Server è aperto o chiuso.

I seguenti tipi di informazioni possono essere impostati per essere persistenti:

- Chiavi per configurare le sessioni per stabilire la connessione con i client
- Dati delle sessioni in volo
- Stato DCV del server Amazon

Amazon DCV Session Manager supporta i database DynamoDB, MariaDB e My. SQL. È necessario configurare e gestire uno di questi database per utilizzare questa funzionalità. Se i tuoi computer broker sono ospitati su AmazonEC2, ti consigliamo di utilizzare DynamoDB come database esterno, poiché non richiede alcuna configurazione aggiuntiva.

Note

La gestione di un database esterno potrebbe comportare costi aggiuntivi. Per informazioni sui prezzi di DynamoDB, consulta Pricing for [Provisioned Capacity](#).

Configurare il broker per persistere su DynamoDB

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

1. Apri `/etc/dcv-session-manager-broker/session-manager-broker.properties` utilizzando il tuo editor di testo preferito e apporta le seguenti modifiche:
 - Imposta `enable-persistence = true`

- Imposta `persistence-db = dynamodb`
 - Per `dynamodb-region` specifica la regione `&aws;` in cui desideri memorizzare le tabelle contenenti i dati del broker. Per l'elenco delle regioni supportate, consulta Endpoint del [servizio DynamoDB](#).
 - Per `dynamodb-table-rcu` specificare la quantità di unità di capacità di lettura (RCU) supportate da ciascuna tabella. Per ulteriori informazioni RCU, consulta [DynamoDB provisioned capacity](#).
 - Per `dynamodb-table-wcu` specificare la quantità di unità di capacità di scrittura (WCU) supportate da ciascuna tabella. Per maggiori informazioni WCU, consulta [DynamoDB provisioned capacity](#).
 - 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. Fermate tutti i broker del cluster. Per ogni broker, esegui il seguente comando:

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

3. Assicurati che tutti i broker del cluster siano fermi, quindi riavviali tutti. Avvia ogni broker eseguendo il seguente comando:

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

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

Configura il broker per persistere su MariaDB/my SQL

Note

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

predefinita, si tratta dell'`dcvsmbrokerutente`. Il Broker verifica all'avvio che il file disponga delle autorizzazioni previste.

Configura i broker per iniziare a rendere persistenti i loro dati su MariaDB/MY: SQL

1. Apri `/etc/dcv-session-manager-broker/session-manager-broker.properties` con il tuo editor di testo preferito e apporta 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. Arresta tutti i broker del cluster. Per ogni broker, esegui il seguente comando:

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

3. Assicurati che tutti i broker del cluster siano fermi, quindi riavviali tutti. Per ogni broker, esegui il seguente comando:

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

Integrazione con Amazon DCV Connection Gateway

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

Se la tua infrastruttura include DCV server Amazon accessibili tramite Amazon DCV Connection Gateway, puoi configurare Session Manager per integrare Amazon DCV Connection Gateway. 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 aggiuntivo. HTTP Il Connection Gateway effettuerà API chiamate all'endpoint per recuperare le informazioni necessarie per instradare DCV le connessioni Amazon all'host selezionato dal broker.

Argomenti

- [Configura Session Manager Broker come risolutore di sessione per Amazon DCV Connection Gateway](#)
- [Facoltativo: abilita TLS l'autenticazione del client](#)
- [Amazon DCV Session Manager Amazon DCV server - riferimento alla DNS mappatura](#)

Configura Session Manager Broker come risolutore di sessione per Amazon DCV Connection Gateway

Lato Session Manager Broker

1. Apri `/etc/dcv-session-manager-broker/session-manager-broker.properties` utilizzando il tuo editor di testo preferito e applica le seguenti modifiche:
 - Imposta `enable-gateway = true`
 - Imposta `gateway-to-broker-connector-https-port` su una TCP porta libera (l'impostazione predefinita è 8447)
 - Impostato sull'indirizzo IP dell'host `gateway-to-broker-connector-bind-host` a cui il Broker si collega per le connessioni Amazon DCV Connection Gateway (l'impostazione predefinita è 0.0.0.0)
2. Quindi esegui i seguenti comandi per arrestare 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 Amazon DCV Connection Gateway nella fase successiva.

Lato Amazon DCV Connection Gateway

- Segui la [sezione](#) nella documentazione di Amazon DCV Connection Gateway.

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

Facoltativo: abilita TLS l'autenticazione del client

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

Note

Se il Session Manager utilizza un sistema di bilanciamento del carico, l'autenticazione TLS client non può essere abilitata con sistemi di bilanciamento del carico con interruzione della TLS connessione, come Application Load Balancers () o Gateway Load Balancers ()ALBs. GLBs Possono essere supportati solo i sistemi di bilanciamento del carico senza TLS terminazione, come Network Load Balancers (). NLBs [Se utilizzi ALBs orGLBs, puoi fare in modo che solo gruppi di sicurezza specifici possano contattare i sistemi di bilanciamento del carico, garantendo un ulteriore livello di sicurezza; maggiori informazioni sui gruppi di sicurezza sono disponibili qui: \[Security groups for your VPC\]\(#\)](#)

Session Manager (lato Broker)

1. Per abilitare l'autenticazione TLS client per la comunicazione tra i broker di Session Manager e Amazon DCV Connection Gateway, segui i passaggi seguenti:
2. Genera le chiavi e i certificati richiesti eseguendo: L'output del comando ti dirà 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
```

- Inserisci una copia della chiave privata e del certificato autofirmato di Amazon DCV Connection Gateway nella tua directory utente. Ne avrai bisogno quando abiliterai l'autenticazione del TLS client in Amazon DCV Connection Gateway nella fase successiva.

```
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
```

- Quindi open `/etc/dcv-session-manager-broker/session-manager-broker.properties` utilizza il tuo editor di testo preferito ed esegui quanto segue:
 - `enable-tls-client-auth-gateway` impostato 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
- Quindi esegui il seguente comando per arrestare e riavviare il Broker:

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

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

Lato Amazon DCV Connection Gateway

- Segui la [sezione](#) nella documentazione di Amazon DCV Connection Gateway.
 - usa il percorso completo del file di certificato che hai copiato nel passaggio precedente per impostare il `cert-file` parametro nella sezione `[resolver]`
 - usa il percorso completo del file chiave che hai copiato nel passaggio precedente per impostare il `cert-key-file` parametro nella sezione `[resolver]`

Amazon DCV Session Manager Amazon DCV server - riferimento alla DNS mappatura

Amazon DCV Connection Gateway richiede i DNS nomi DCV dei server Amazon per connettersi alle istanze del DCV server. Questa sezione illustra come definire un JSON file contenente la mappatura tra ciascun DCV server e il nome associato. DNS

Struttura dei file

La mappatura consiste in un elenco di JSON oggetti 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 ipAddress agentServerId, e: instanceld

Ip:

Disponibile sia per Amazon EC2 che per le 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. DescribeServers API

Id:

Disponibile sia per le infrastrutture Amazon EC2 che on-premise; l'agente Session Manager ne crea uno nuovo UUID ogni volta che il nome host o l'indirizzo IP cambiano. Queste informazioni sono disponibili nella risposta. DescribeServers API

Host.Aws.Ec2InstanceId:

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

ServerId:

Un ID del tipo specificato che identifica in modo univoco ogni DCV server Amazon nella rete.

DnsNames:

L'oggetto contenente i DNS nomi associati al DCV server Amazon conterrà:

InternalDnsNames:

Il DNS nome usato da Amazon DCV Connection Gateway per connettersi all'istanza.

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

Persistenza

Ti consigliamo vivamente di abilitare la funzionalità di persistenza del Session Manager Broker, per proteggerti dalla perdita di mappatura in caso di guasto di più broker o dell'intero cluster. [Per ulteriori informazioni sull'attivazione della persistenza dei dati, consulta Configure Broker Persistence](#)

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 di Amazon Session Manager DCV

Man mano che DCV i sistemi Amazon crescono in termini di scalabilità e complessità, è importante garantire che il Session Manager rimanga up-to-date e sia in grado di gestire le crescenti richieste. Sia il pacchetto agente che quello broker richiederanno di tanto in tanto un aggiornamento. Questa sezione descrive il processo di aggiornamento di Amazon DCV Session Manager, descrive la procedura di aggiornamento e i consigli per la manutenzione del sistema.

Il seguente argomento descrive come aggiornare il Session Manager.

Note

Si consiglia vivamente di aggiornare tutti gli agenti di Session Manager prima di aggiornare i broker di Session Manager per evitare problemi di incompatibilità quando vengono introdotte nuove funzionalità.

Argomenti

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

Aggiornamento dell'agente Amazon DCV Session Manager

Gli agenti di Amazon DCV Session Manager ricevono istruzioni dal broker e le eseguono sui rispettivi DCV server Amazon. Come parte della manutenzione ordinaria, gli agenti devono essere aggiornati per soddisfare nuovi standard e requisiti. Questa sezione illustra il processo di aggiornamento degli agenti di Session Manager.

Linux host

Note

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

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 e RHEL 7.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.el7.x86_64.rpm
```

- RHEL8.x e Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.el8.x86_64.rpm
```

- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2404.deb
```

- SUSELinux Enterprise 12

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.sles12.x86_64.rpm
```

- SUSELinux Enterprise 15

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.sles15.x86_64.rpm
```

3. Installare il pacchetto .

- Amazon Linux 2 e RHEL 7.x

```
$ sudo yum install -y nice-dcv-session-manager-agent-2024.0.781-1.el7.x86_64.rpm
```

- RHEL8.x e Rocky Linux 8.x

```
$ sudo yum install -y nice-dcv-session-manager-agent-2024.0.781-1.el8.x86_64.rpm
```

- Ubuntu 20.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2404.deb
```

- SUSELinux Enterprise 12

```
$ sudo zypper install nice-dcv-session-manager-agent-2024.0.781-1.sles12.x86_64.rpm
```

- SUSELinux Enterprise 15

```
$ sudo zypper install nice-dcv-session-manager-agent-2024.0.781-1.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 dell'agente. Esegui i seguenti comandi al prompt dei comandi.

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

2. Scarica il programma di [installazione dell'agente](#).
3. Eseguire il programma di installazione. Nella schermata iniziale, selezionare Next (Successivo).
4. EULASullo schermo, 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 dell'agente. Esegui i seguenti comandi al prompt dei comandi.

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

Aggiornamento del broker Amazon DCV Session Manager

I broker DCV di Amazon Session Manager trasmettono API le richieste agli agenti competenti. Sono installati su un host separato dai DCV server Amazon. Come parte della manutenzione ordinaria, i broker devono essere aggiornati per soddisfare nuovi standard e requisiti. Questa sezione illustra il processo di aggiornamento dei broker di Session Manager.

Per aggiornare il broker

1. Connettiti all'host su cui intendi aggiornare il broker.
2. Interrompi il servizio di brokeraggio.

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

3. Scarica il pacchetto di installazione.

- Amazon Linux 2 e RHEL 7.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1.el7.noarch.rpm
```

- RHEL8.x e Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1.el8.noarch.rpm
```

- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1_all.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1_all.ubuntu2404.deb
```

4. Installare il pacchetto .

- Amazon Linux 2 e RHEL 7.x

```
$ sudo yum install -y nice-dcv-session-manager-broker-2024.0.457-1.el7.noarch.rpm
```

- RHEL8.x e Rocky Linux 8.x

```
$ sudo yum install -y nice-dcv-session-manager-broker-2024.0.457-1.el8.noarch.rpm
```

- Ubuntu 20.04

```
$ sudo apt install -y nice-dcv-session-manager-broker-2024.0.457-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ sudo apt install -y nice-dcv-session-manager-  
broker-2024.0.457-1_all.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ sudo apt install -y nice-dcv-session-manager-  
broker-2024.0.457-1_all.ubuntu2404.deb
```

5. Avvia il servizio di brokeraggio 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
```

CLI Riferimento del broker

Il broker Amazon DCV Session Manager è uno strumento di interfaccia a riga di comando (CLI) che fornisce il controllo amministrativo su Session Manager. Questo riferimento copre il set completo di CLI comandi disponibili per la gestione di sessioni, utenti, risorse e altri aspetti di Session Manager. Gli amministratori possono automatizzare le attività di gestione di routine, risolvere i problemi e ottimizzare le prestazioni della propria infrastruttura Amazon. DCV

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

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

Utilizzate i seguenti comandi se utilizzate il broker Session Manager 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 l'agente Session Manager.

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

Utilizzate i seguenti comandi per gestire il file di mappatura DNS dei nomi del DCV server.

- [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

Il server URL di autenticazione esterno da utilizzare. È necessario aggiungerlo `.well-known/jwks.json` al server di autenticazione. URL

Tipo: stringa

Campo obbligatorio: sì

Esempio

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

I server URLs di autenticazione esterni registrati.

Esempio

L'esempio seguente elenca tutti i server di autenticazione esterni che sono stati 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, questo non può più essere utilizzato 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

Il server URL di autenticazione esterno di cui annullare la registrazione. È necessario aggiungerlo `.well-known/jwks.json` al server di autenticazione. URL

Tipo: stringa

Campo obbligatorio: sì

Output

Url

Il URL server di autenticazione esterno non registrato.

Esempio

L'esempio seguente registra un server di autenticazione esterno con un diURL. `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 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 possono essere recuperate 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 Session Manager.

Tipo: stringa

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

L'esempio seguente registra 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: NjVmZDR1N2ItNjNmYS00M2QxLWFlZmMtZmNmMDNkMEXAMPLE
```

describe-api-clients

Elenca i client di Session Manager che sono stati registrati presso 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 di Session Manager disattivato non può più utilizzare le proprie credenziali per recuperare OAuth i token di accesso 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: stringa

Campo obbligatorio: sì

Esempio

L'esempio seguente disattiva un client Session Manager con un ID client di.
f855b54b-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 forniti 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 un'istruzione software.

Gli agenti devono essere registrati presso il broker per consentire la comunicazione. Gli agenti necessitano di una dichiarazione software per registrarsi presso il broker. Dopo che l'agente ha una dichiarazione software, può registrarsi automaticamente presso il broker utilizzando il [Dynamic Client Registration Protocol OAuth 2.0](#). Dopo la registrazione presso il broker, l'agente riceve un ID cliente e un client secret che utilizza per l'autenticazione con il broker.

Il broker e l'agente ricevono e utilizzano un'istruzione software predefinita al momento della prima installazione. È possibile continuare a utilizzare l'istruzione software predefinita oppure 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 al `agent.software_statement_path` parametro nel `agent.conf` file. Dopo aver eseguito questa operazione, arrestate e riavviate l'agente 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:  
ewogICJpZCIgOiAiYjc1NTVhN2QtNWl0MC00OTJhLWJjOTUtNmUzOWNhYzIxMDcxIiwKICAiYWN0aXZlIiA6IHRydWUsCi
```

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

Data e 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

L'istruzione software da disattivare.

Tipo: stringa

Campo obbligatorio: sì

Esempio

L'esempio seguente disattiva un'istruzione software.

Comando

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

Output

```
Software statement  
EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWI0MC00TJhLWJj0TUtNmUz0WNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj  
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

Annulla 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 di cui annullare la registrazione.

Tipo: stringa

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 DCV server: DNS nomina le mappature provenienti da un JSON file.

Sintassi

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

Opzioni

--file-path

Il percorso del file contenente le mappature DCV Servers - DNS names.

Tipo: stringa

Campo obbligatorio: sì

Esempio

L'esempio seguente registra le mappature DCV Servers - DNS names da .json. file /tmp/mappings

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 le mappature dei DNS nomi dei DCV server 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

L'esempio seguente elenca le mappature DCV Servers - DNS names registrate.

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

Questa sezione di riferimento fornisce una panoramica delle opzioni di configurazione disponibili per il Session Manager. Le configurazioni includono modifiche sia al file dell'agente che al file del broker. Ogni configurazione include una spiegazione dello scopo, dei valori accettati e dell'impatto sul comportamento generale del sistema. Amazon DCV Session Manager può essere personalizzato per soddisfare i requisiti unici di un DCV sistema Amazon.

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. È possibile modificare il file di configurazione utilizzando l'editor di testo preferito.

Note

Il `/etc/dcv-session-manager-broker/session-manager-broker.properties` file 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.

La tabella seguente elenca i parametri nel file di configurazione del broker.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
<code>broker-ja</code>	No		Specifica il percorso della home directory Java che

Nome del parametro	Richiesto	Valore predefinito	Descrizione
java-home			<p>il broker utilizzerà al posto di quella predefinita del sistema. Se impostato <code><broker-java-home></code>, il broker lo utilizzerà <code><broker-java-home>/bin/java</code> all'avvio.</p> <p>Suggerimento: il broker richiede Java Runtime Environment 11 e, se manca, viene installato o come dipendenza al termine dell'installazione. Se la versione 11 non è impostata come ambiente Java predefinito, la sua home directory può essere recuperata utilizzando il seguente comando:</p> <pre>\$ sudo alternatives --display java</pre>
session-screenshots-max-width	No	160	<p>Specifica la larghezza massima, in pixel, delle schermate di sessione acquisite utilizzando <code>GetSessionScreenshotsAPI</code>.</p>

Nome del parametro	Richiesto	Valore predefinito	Descrizione
session-screenshots-max-height	No	100	Specifica l'altezza massima, in pixel, delle schermate di sessione acquisite utilizzando GetSessionScreenshotsAPI
session-screenshots-format	No	png	Il formato del file di immagine delle schermate della sessione acquisite utilizzando GetSessionScreenshotsAPI
create-sessions-queue-max-size	No	1000	Il numero massimo di CreateSessionsAPI richieste 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, in secondi, in cui una CreateSessionsAPI richiesta non soddisfatta può rimanere in coda. Se la richiesta non può essere soddisfatta entro il periodo di tempo specificato, ha esito negativo.

Nome del parametro	Richiesto	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-auth-on-server	Sì	true	Specifica se il broker è il server di autenticazione utilizzato per generare token di accesso OAuth 2.0 per il client. APIs
enable-auth-on	Sì	true	Abilita o disabilita l'autorizzazione del client. Se si abilita l'autorizzazione del client, il client API deve fornire un token di accesso quando effettua le richieste. API Se disabiliti l'autorizzazione del client, il client APIs può effettuare richieste senza token di accesso.
enable-agent-authorization	Sì	true	Abilita o disabilita l'autorizzazione dell'agente. Se si abilita l'autorizzazione dell'agente, l'agente deve fornire un token di accesso quando comunica con il broker.

Nome del parametro	Richiesto	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 DescribeSessions API chiamate.
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 HTTPS porta in cui il broker ascolta le connessioni dei client.
client-to-broker-connect-bind-host	No	0.0.0.0	Specifica l'indirizzo IP dell'host a cui il broker si collega per le connessioni client.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
client-to-broker-connect-key-store-file	Sì		Specifica l'archivio di chiavi utilizzato per TLS le connessioni client.
client-to-broker-connect-key-store-pass	Sì		Specifica il key store pass.
agent-to-broker-connect-https-port	Sì	8445	Specifica la HTTPS porta in cui il broker ascolta le connessioni degli agenti.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
agent-to-broker-connect-to-broker-host	No	0.0.0.0	Specifica l'indirizzo IP dell'host a cui il broker si connette per le connessioni degli agenti.
agent-to-broker-connect-key-store-file	Sì		Specifica l'archivio di chiavi utilizzato per TLS le connessioni degli agenti.
agent-to-broker-connect-key-store-pass	Sì		Specifica il key store pass.
broker-to-broker-port	Sì	47100	Specifica la porta utilizzata per broker-to-broker le connessioni.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
broker-to-broker-ind-host	No	0.0.0.0	Specifica l'indirizzo IP dell'host a cui il broker si collega per le connessioni broker-to-broker
broker-to-broker-discovery-port	Sì	47500	Specifica la porta utilizzata dai broker per scoprirsi l'un l'altro.
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 specifichi <code>ibroker-to-broker-discovery-multicast-group</code> , <code>broker-to-broker-discovery-multicast-port</code> , <code>broker-to-broker-discovery-AWS-region</code> , <code>broker-to-broker-discovery-AWS-alb-target-group-arn</code> , o ometti questo parametro.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
broker-to-broker-discovery-multicast-group	No		Specifica il gruppo multicast per il rilevamento. broker-to-broker-discovery-multicast-group Se si specifica o broker-to-broker-discovery-addresses broker-to-broker-discovery-aws-region broker-to-broker-discovery-AWS-alb-target-group-arn si omette questo parametro.
broker-to-broker-discovery-multicast-port	No		Specifica la porta multicast per il rilevamento. broker-to-broker-discovery-multicast-port Se si specifica o broker-to-broker-discovery-addresses broker-to-broker-discovery-AWS-region broker-to-broker-discovery-AWS-alb-target-group-arn si omette questo parametro.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
<code>broker-to-broker-discovery-AWS-region</code>	No		Specifica la AWS regione del sistema di bilanciamento del carico dell'applicazione utilizzato per l'individuazione da broker a broker. Se si specifica <code>broker-to-broker-discovery-multicast-group</code> <code>broker-to-broker-discovery-multicast-port</code> <code>broker-to-broker-discovery-addresses</code> si omette questo parametro.
<code>broker-to-broker-discovery-AWS-alb-target-group-arn</code>	No		L'utente ARN del gruppo target dell'Application Load Balancer per il broker-to-broker rilevamento. Se si specifica <code>broker-to-broker-discovery-multicast-group</code> <code>broker-to-broker-discovery-multicast-port</code> <code>broker-to-broker-discovery-addresses</code> si omette questo parametro.

Nome del parametro	Richiesto	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 delle sessioni AmazonDCV .
broker-to-broker-key-store-file	Sì		Specifica l'archivio di chiavi utilizzato per le connessioni del broker. TLS
broker-to-broker-key-store-pass	Sì		Specifica il key store pass.
enable-cloud-watch-metrics	No	false	Abilita o disabilita i parametri di Amazon CloudWatch . Se abiliti CloudWatch Metrics, potresti dover specificare un valore per. <code>cloud-watch-region</code>

Nome del parametro	Richiesto	Valore predefinito	Descrizione
<code>cloud-watch-region</code>	No	Richiesto solo se <code>enable-cloud-watch-metrics</code> è impostato su <code>true</code> . Se il broker è installato su un'EC2istanza Amazon, la regione viene recuperata da 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 del broker può elaborare ogni secondo prima di essere limitata.
<code>enable-throttling-forward-header</code>	No	false	Se impostato su <code>throttling</code> , recupera <code>true</code> l'ip del chiamante dall'intestazione, se presente. X-Forwarded-For
<code>create-session-number-of-retries-on-failure</code>	No	2	Specifica il numero massimo di tentativi da eseguire dopo che una richiesta di creazione di sessione non è riuscita su un host di DCV server Amazon. Imposta su 0 per non eseguire mai nuovi tentativi in caso di errori.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
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-persistence	Sì	false	Se impostato su true, i dati sullo stato del broker vengono conservati in un database esterno.
persistence-db	No	Richiesto solo se enable-persistence è impostato su true	Specifica quale database viene utilizzato per la persistenza. Gli unici valori supportati sono: dynamodb e mysql
dynamodb-region	No	Richiesto solo se enable-persistence è impostato su true ed persistence-db è impostato su dynamodb.	Specifica la regione in cui vengono create e accedute le tabelle DynamoDB.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
dynamoc-table-rcu	No	Richiesto solo se enable-persistence è impostato su true ed persistence-db è impostato su. dynamodb	Specifica le unità di capacità di lettura (RCU) per ogni tabella DynamoDB. Per ulteriori informazioni, consulta Pricing for RCU Provision ed Capacity .
dynamoc-table-wcu	No	Richiesto solo se enable-persistence è impostato su true ed persistence-db è impostato su. dynamodb	Specifica le unità di capacità di scrittura (WCU) per ogni tabella DynamoDB. Per ulteriori informazioni, consulta Pricing for WCU Provision ed Capacity .
dynamoc-table-name-prefix	No	Richiesto 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 che utilizzano lo stesso account). AWS Sono consentiti solo caratteri alfanumerici, punto, trattino e trattino basso.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
jdbc-connection-url	No	Richiesto solo se enable-persistence è impostato su true ed persistence-db è impostato su. mysql	<p>Specifica la connessione URL al database MariaDB/my; contiene l'endpoint e il nome del SQL database. L'url dovrebbe avere questo formato:</p> <pre>jdbc:mysql://<db_endpoint>:<db_port>/<db_name>?createDatabaseIfNotExist=true</pre> <p><db_endpoint> Dov'è l'endpoint del SQL database MariaDB/My, <db_port> è la porta del database ed <db_name> è il nome del database.</p>
jdbc-user	No	Richiesto 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	Richiesto 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.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
<code>seconds before-deleting-unreachable-dcv-server</code>	No	1800	Specifica il numero 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. È possibile modificare il file di configurazione utilizzando l'editor di testo preferito.

La tabella seguente elenca i parametri nel file di configurazione dell'agente.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
<code>agent.tls_ker_host</code>	Sì		Specifica il DNS nome dell'host del broker.
<code>agent.tls_ker_port</code>	Sì	8445	Specifica la porta tramite la quale comunicare con il broker.
<code>agent.tls_strict</code>	No		Necessario solo se <code>tls_strict</code> è impostato

Nome del parametro	Richiesto	Valore predefinito	Descrizione
			<p>su. <code>true</code> Specifica il percorso del file del certificato (.pem) necessari o per convalidare il certificato. TLS Copia il certificato autofirmato dal broker all'agente.</p>
<code>agent.init_folder</code>	No	<ul style="list-style-type: none"> <code>/var/lib/dcv-session-manager-agent/init</code> (Linux) 	<p>Specifica il percorso di una cartella sul server host utilizzata per archiviare gli script personalizzati autorizzati a inizializzare le sessioni del server DCV Amazon al momento della creazione. È necessario specificare un percorso assoluto. La cartella deve essere accessibile e i file devono essere eseguibili dagli utenti che utilizzano il parametro <code>InitFilerequest</code> di <code>CreateSessionsAPI</code>.</p>
<code>agent.init_strict</code>	No	<code>true</code>	<p>Indica se è necessari o utilizzare TLS una convalida rigorosa.</p>

Nome del parametro	Richiesto	Valore predefinito	Descrizione
agent.software_statement_path	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.tags_folder	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 dei tag. Per ulteriori informazioni, consulta Utilizzo dei tag per indirizzare DCV i server Amazon .

Nome del parametro	Richiesto	Valore predefinito	Descrizione
<code>agent.autorun_folder</code>	No	<ul style="list-style-type: none"> <code>/var/lib/dcv-session-manager-agent/autorun</code> (Linux) <code>C:\ProgramData\NICE\DcvSessionManagerAgent\autorun</code> (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 <code>AutorunFilerequest</code> di <code>CreateSessionsAPI</code> .
<code>agent._virtual_sessions</code>	No	-1 (nessun limite)	Il numero massimo di sessioni virtuali che possono essere create su un DCV server Amazon utilizzando Amazon DCV Session Manager.
<code>agent._concurrent_sessions_per_server</code>	No	1	Il numero massimo di sessioni virtuali che possono essere create su un DCV server Amazon da un singolo utente utilizzando Amazon DCV Session Manager.

Nome del parametro	Richiesto	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 DCV del server e dell'host Amazon, oltre a informazioni aggiornate sulla sessione. I valori più bassi rendono il Session Manager più consapevole delle modifiche che avvengono nel sistema su 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 quelli non 120 sono consigliati.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
log.level	No	info	<p>Specifica il livello di dettaglio dei file di registro. Sono disponibili i seguenti livelli di dettaglio:</p> <ul style="list-style-type: none"> • <code>error</code>—Fornisce il minor numero di dettagli. 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 registro.

Nome del parametro	Richiesto	Valore predefinito	Descrizione
log.rotation	No	daily	<p>Specifica la rotazione del file di registro. I valori validi sono:</p> <ul style="list-style-type: none"> hourly—I file di registro vengono ruotati ogni ora. daily—I file di registro vengono ruotati giornalmente.
log.max-file-size	No	10485760	<p>Quando la dimensione di un file di registro raggiunge la dimensione specificata in byte, viene ruotato. Verrà creato un nuovo file di registro e ulteriori eventi di registro verranno inseriti nel nuovo file.</p>
log.rotate	No	9	<p>Il numero massimo di file di registro conservati durante la 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 Amazon DCV Session Manager

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

Argomenti

- [Note sulla versione DCV di Amazon Session Manager](#)
- [Cronologia dei documenti](#)

Note sulla versione DCV di Amazon Session Manager

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

Argomenti

- [2024.0-457— 1 ottobre 2024](#)
- [2023.1-17652 — 1 agosto 2024](#)
- [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](#)

2024.0-457 — 1 ottobre 2024

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none">• Intermediario: 457• Agente: 748• CLI: 140	<ul style="list-style-type: none">• Rinominato in NICE DCV Amazon. DCV• Aggiunto il supporto per Ubuntu 24.04.

2023.1-17652 — 1 agosto 2024

Numeri build	Modifiche e correzioni di bug
<ul style="list-style-type: none">• Intermediario: 426• Agente: 748• CLI: 140	<ul style="list-style-type: none">• Correzione dei bug e miglioramenti delle prestazioni.

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">• È stato corretto 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 ARM su piattaforme.

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">• È stato aggiunto un Substate campo in DescribeSessions risposta.

Numeri build	Modifiche e correzioni di bug
	<ul style="list-style-type: none"> • Risolto un problema che poteva causare la CLI mancata connessione al broker a seconda del tipo 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"> • È stato 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. • È stato migliorato il trasferimento dei dati dalla cache al disco nel Broker. • URLValidazione fissa in. 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 Amazon DCV Connection Gateway. • 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 delle 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 • Agente: 413 • CLI: 37 	<ul style="list-style-type: none"> • Amazon DCV Session Manager ora offre il supporto dell'interfaccia a riga di comando (CLI). Puoi creare e gestire DCV sessioni Amazon inCLI, anziché chiamare APIs. • Amazon DCV Session Manager ha introdotto la persistenza dei dati di Broker. Per una maggiore disponibilità, i broker possono conservare le informazioni sullo stato del server in un archivio dati esterno e ripristinare i dati all'avvio. 	<ul style="list-style-type: none"> • Quando si registra un server di autorizzazione esterno, è ora possibile specificare l'algoritmo utilizzato dal server di autorizzazione per firmare JSON token Web in formato. 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"> • Amazon 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 a. 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"> • Amazon DCV Session Manager ha introdotto le seguenti novitàAPIs: <ul style="list-style-type: none"> • OpenServers • CloseServers • DescribeServers • GetSessionScreenshots • Ha inoltre introdotto i seguenti nuovi parametri di configurazione: <ul style="list-style-type: none"> • Parametri del broker: <code>session-screenshot-max-width</code> <code>session-screenshot-max-height</code> <code>session-screenshot-format</code> <code>create-sessions-queue-max-size</code> <code>create-sessions-queue-max-time-seconds</code> . • Parametri dell'agente: <code>agent.autorun_folder</code> <code>max_virtual_sessions</code> <code>max_concurrent_sessions_per_user</code> . • Parametri dell'agente: <code>agent.autorun_folder</code> <code>max_virtual_sessions</code> <code>max_concurrent_sessions_per_user</code> . • Parametri dell'agente: <code>agent.autorun_folder</code> <code>max_virtual_sessions</code> <code>max_concurrent_sessions_per_user</code> .

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 TLS certificati 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 Amazon DCV Session Manager.

Cronologia dei documenti

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

Modifica	Descrizione	Data
Amazon DCV versione 2024.0-457	Amazon DCV Session Manager è stato aggiornato per Amazon DCV 2024.0-457. Per ulteriori informazioni, consulta 2024.0-457— 1 ottobre 2024 .	30 settembre 2024
Amazon DCV versione 2023.1-17652	Amazon DCV Session Manager è stato aggiornato per Amazon DCV 2023.1-17652. Per ulteriori informazioni, consulta 2023.1-17652 — 1 agosto 2024 .	1° agosto 2024
Amazon DCV versione	Amazon DCV Session Manager è stato aggiornato per Amazon DCV 2023.1-16	26 giugno 2024

Modifica	Descrizione	Data
2023.1-16388	388. Per ulteriori informazioni, consulta 2023.1-16388— 26 giugno 2024.	
Amazon DCV versione 2023.1	Amazon DCV Session Manager è stato aggiornato per Amazon DCV 2023.1. Per ulteriori informazioni, consulta 2023.1 — 9 novembre 2023.	9 novembre 2023
Amazon DCV versione 2023.0	Amazon DCV Session Manager è stato aggiornato per Amazon DCV 2023.0. Per ulteriori informazioni, consulta 2023.0-14852— 28 marzo 2023.	28 marzo 2023
Amazon DCV versione 2022.2	Amazon DCV Session Manager è stato aggiornato per Amazon DCV 2022.2. Per ulteriori informazioni, consulta 2022.2-13907 — 11 novembre 2022.	11 novembre 2022
Amazon DCV versione 2022.1	Amazon DCV Session Manager è stato aggiornato per Amazon DCV 2022.1. Per ulteriori informazioni, consulta 2022.1-13067 — 29 giugno 2022.	29 giugno 2022
Amazon DCV versione 2022.0	Amazon DCV Session Manager è stato aggiornato per Amazon DCV 2022.0. Per ulteriori informazioni, consulta 2022.0-11952— 23 febbraio 2022.	23 febbraio 2022
Amazon DCV versione 2021.3	Amazon DCV Session Manager è stato aggiornato per Amazon DCV 2021.3. Per ulteriori informazioni, consulta 2021.3-11591— 20 dicembre 2021.	20 dicembre 2021
Amazon DCV versione 2021.2	Amazon DCV Session Manager è stato aggiornato per Amazon DCV 2021.2. Per ulteriori informazioni, consulta 2021.2-11042— 01 settembre 2021.	1 settembre 2021

Modifica	Descrizione	Data
Amazon DCV versione 2021.1	Amazon DCV Session Manager è stato aggiornato per Amazon DCV 2021.1. Per ulteriori informazioni, consulta 2021.1-10557— 31 maggio 2021 .	31 maggio 2021
Amazon DCV versione 2021.0	Amazon DCV Session Manager è stato aggiornato per Amazon DCV 2021.0. Per ulteriori informazioni, consulta 2021.0-10242 — 12 aprile 2021 .	12 Aprile 2021
Versione iniziale di Amazon 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.