



Panduan Administrator

# Manajer Sesi NICE DCV



# Manajer Sesi NICE DCV: Panduan Administrator

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

---

# Table of Contents

Apa itu Manajer Sesi? .....	1
Bagaimana Manajer Sesi bekerja .....	1
Fitur .....	3
Batasan .....	3
Penetapan harga .....	3
Persyaratan .....	4
Persyaratan jaringan dan konektivitas .....	5
Menyiapkan .....	7
Langkah 1: Siapkan server NICE DCV .....	7
Langkah 2: Siapkan Broker .....	8
Langkah 3: Siapkan Agen .....	10
Langkah 4: Konfigurasi server NICE DCV .....	15
Langkah 5: Verifikasi instalasi .....	16
Verifikasi Agen .....	17
Verifikasi Broker .....	18
Mengkonfigurasi .....	19
Manajer Sesi Penskalaan .....	19
Langkah 1: Buat profil instance .....	20
Langkah 2: Siapkan sertifikat SSL untuk penyeimbang beban .....	21
Langkah 3: Buat penyeimbang beban aplikasi Broker .....	22
Langkah 4: Luncurkan Broker .....	23
Langkah 5: Buat penyeimbang beban aplikasi Agen .....	24
Langkah 6: Luncurkan Agen .....	25
Menggunakan tag .....	26
Mengonfigurasi server otorisasi eksternal .....	28
Mengkonfigurasi ketekunan broker .....	33
Konfigurasi broker untuk bertahan di DynamoDB .....	33
Konfigurasi broker untuk bertahan di Mariadb/MySQL .....	34
Mengintegrasikan dengan Gateway Koneksi NICE DCV .....	35
Siapkan Session Manager Broker sebagai Session Resolver untuk NICE DCV Connection Gateway .....	36
Opsional - Aktifkan otentikasi klien TLS .....	37
Server NICE DCV - pemetaan DNS .....	38
Mengintegrasikan dengan Amazon CloudWatch .....	40

---

Meningkatkan .....	43
Meningkatkan Agen Manajer Sesi NICE DCV .....	43
Meningkatkan Broker Manajer Sesi NICE DCV .....	45
Referensi CLI Broker .....	48
register-auth-server .....	49
Sintaksis .....	49
Opsis .....	49
Contoh .....	49
list-auth-servers .....	50
Sintaksis .....	49
Output .....	50
Contoh .....	49
unregister-auth-server .....	51
Sintaksis .....	49
Opsis .....	49
Output .....	50
Contoh .....	49
register-api-client .....	52
Sintaksis .....	49
Opsis .....	49
Output .....	50
Contoh .....	49
describe-api-clients .....	53
Sintaksis .....	49
Output .....	50
Contoh .....	49
unregister-api-client .....	55
Sintaksis .....	49
Opsis .....	49
Contoh .....	49
renew-auth-server-api-kunci .....	56
Sintaksis .....	49
Contoh .....	49
generate-software-statement .....	56
Sintaksis .....	49
Output .....	50

Contoh .....	49
describe-software-statements .....	58
Sintaksis .....	49
Output .....	50
Contoh .....	49
deactivate-software-statement .....	59
Sintaksis .....	49
Opsi .....	49
Contoh .....	49
describe-agent-clients .....	60
Sintaksis .....	49
Output .....	50
Contoh .....	49
unregister-agent-client .....	61
Sintaksis .....	49
Opsi .....	49
Contoh .....	49
register-server-dns-mappings .....	62
Sintaksis .....	49
Opsi .....	49
Contoh .....	49
describe-server-dns-mappings .....	63
Sintaksis .....	49
Output .....	50
Contoh .....	49
Referensi Referensi .....	65
File .....	65
File konfigurasi agen .....	81
Catatan Rilis dan Riwayat Dokumen .....	88
Catatan Rilis .....	88
2023.1-16388— 26 Juni 2024 .....	89
2023.1— November 9, 2023 .....	89
2023.0-15065— 4 Mei 2023 .....	89
2023.0-14852— 28 Maret 2023 .....	89
2022.2-13907— November 11, 2022 .....	90
2022.1-13067— 29 Juni 2022 .....	90

---

2022.0-11952— 23 Februari 2022 .....	90
2021.3-11591— Desember 20, 2021 .....	91
2021.2-11445— November 18, 2021 .....	91
2021.2-11190— 11 Oktober 2021 .....	91
2021.2-11042— September 01, 2021 .....	91
2021.1-10557— 31 Mei 2021 .....	92
2021.0-10242— 12 April 2021 .....	92
2020.2-9662— 04 Desember 2020 .....	93
.....	93
Riwayat dokumen .....	94
.....	xcvi

# Apa itu Manajer Sesi NICE DCV?

NICE DCV Session Manager adalah seperangkat paket perangkat lunak yang dapat diinstal (Agen dan Broker) dan antarmuka pemrograman aplikasi (API) yang memudahkan pengembang dan vendor perangkat lunak independen (ISV) untuk membangun aplikasi front-end yang secara terprogram membuat dan mengelola siklus hidup sesi NICE DCV di seluruh armada server NICE DCV.

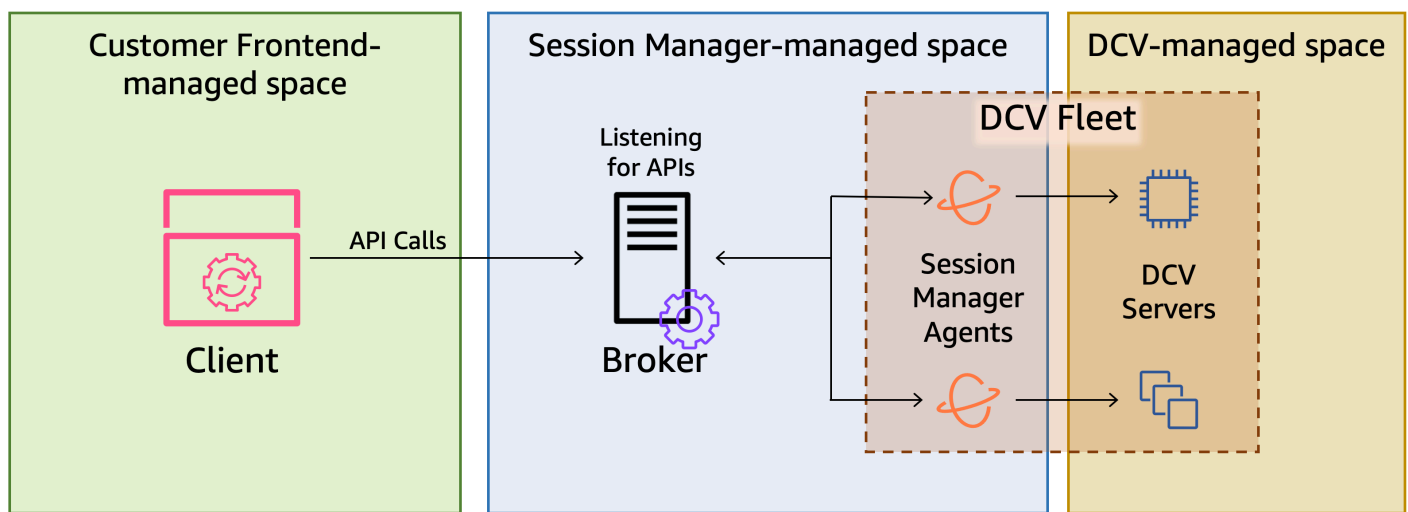
Panduan ini menjelaskan cara menginstal dan mengkonfigurasi Agen Manajer Sesi dan Broker. Untuk informasi selengkapnya tentang menggunakan API Manajer Sesi, lihat Panduan Pengembang Manajer Sesi DCV NICE DCV.

## Topik

- [Bagaimana Manajer Sesi bekerja](#)
- [Fitur](#)
- [Batasan](#)
- [Penetapan harga](#)
- [Persyaratan Manajer Sesi NICE DCV](#)

## Bagaimana Manajer Sesi bekerja

Diagram berikut menunjukkan komponen tingkat tinggi Manajer Sesi.



## Pialang

Broker adalah server web yang menghosting dan mengekspos Session Manager API. Ini menerima dan memproses permintaan API untuk mengelola sesi NICE DCV dari klien, dan kemudian meneruskan instruksi ke Agen yang relevan. Broker harus diinstal pada host yang terpisah dari server NICE DCV Anda, tetapi harus dapat diakses oleh klien, dan harus dapat mengakses Agen.

## Agen

Agen diinstal pada setiap server NICE DCV di armada. Agen menerima instruksi dari Broker dan menjalankannya di server NICE DCV masing-masing. Agen juga memantau keadaan server NICE DCV, dan mengirim pembaruan status berkala kembali ke Broker.

## API

Session Manager memaparkan satu set antarmuka pemrograman aplikasi REST (API) yang dapat digunakan untuk mengelola sesi NICE DCV pada armada server NICE DCV. API di-host dan diekspos oleh Broker. Pengembang dapat membangun klien manajemen sesi khusus yang memanggil API.

## Klien

Klien adalah aplikasi front-end atau portal yang Anda kembangkan untuk memanggil Session Manager API yang diekspos oleh Broker. Pengguna akhir menggunakan klien untuk mengelola sesi yang dihosting di server NICE DCV di armada.

## Token akses

Untuk membuat permintaan API, Anda harus memberikan token akses. Token dapat diminta dari Broker, atau server otorisasi eksternal, oleh API klien terdaftar. Untuk meminta dan mengakses token, API klien harus memberikan kredensi yang valid.

## API Klien

API klien dihasilkan dari file YAMM definisi API Session Manager, menggunakan Swagger Codegen. API klien digunakan untuk membuat permintaan API.

## Sesi DCV NICE

Anda harus membuat sesi DCV NICE di server NICE DCV Anda yang dapat terhubung dengan klien Anda. Klien hanya dapat terhubung ke server NICE DCV jika ada sesi aktif. NICE DCV mendukung konsol dan sesi virtual. Anda menggunakan Session Manager API untuk mengelola siklus hidup sesi NICE DCV. Sesi DCV NICE dapat berada di salah satu negara bagian berikut:



- CREATING—Broker sedang dalam proses membuat sesi.
- READY—sesi siap menerima koneksi klien.
- DELETING—sesi sedang dihapus.
- DELETED—sesi telah dihapus.
- UNKNOWN—tidak dapat menentukan status sesi. Broker dan Agen mungkin tidak dapat berkomunikasi.

## Fitur

Manajer Sesi DCV menawarkan fitur-fitur berikut:

- Menyediakan informasi sesi NICE DCV —dapatkan informasi tentang sesi yang berjalan di beberapa server NICE DCV.
- Kelola siklus hidup untuk beberapa sesi NICE DCV —buat atau hapus beberapa sesi untuk beberapa pengguna di beberapa server NICE DCV dengan satu permintaan API.
- Mendukung tag —gunakan tag khusus untuk menargetkan sekelompok server NICE DCV saat membuat sesi.
- Mengelola izin untuk beberapa sesi NICE DCV —ubah izin pengguna untuk beberapa sesi dengan satu permintaan API.
- Menyediakan informasi koneksi —mengambil informasi koneksi klien untuk sesi NICE DCV.
- Mendukung cloud dan lokal —gunakan Pengelola Sesi diAWS, lokal, atau dengan server berbasis Internet alternatif.

## Batasan

Session Manager tidak menyediakan kemampuan penyediaan sumber daya. Jika Anda menjalankan NICE DCV di instans Amazon EC2, Anda mungkin perlu menggunakan layanan AWS tambahan, seperti Amazon EC2 Auto Scaling untuk mengelola penskalaan infrastruktur Anda.

## Penetapan harga

Session Manager tersedia tanpa biaya untuk AWS pelanggan yang menjalankan instans EC2.

Pelanggan lokal memerlukan lisensi NICE DCV Plus atau DCV Professional Plus. Untuk informasi tentang cara membeli lisensi NICE DCV Plus atau NICE DCV Professional Plus, [lihat Cara Membeli](#) di situs web NICE dan temukan distributor atau pengecer NICE di wilayah Anda. Untuk memungkinkan semua pelanggan lokal bereksperimen dengan Manajer Sesi DCV, persyaratan lisensi hanya akan diberlakukan mulai dari NICE DCV versi 2021.0.

Untuk informasi selengkapnya, lihat [Melisensikan Server DCV NICE](#) di Panduan Administrator DCV NICE.

## Persyaratan Manajer Sesi NICE DCV

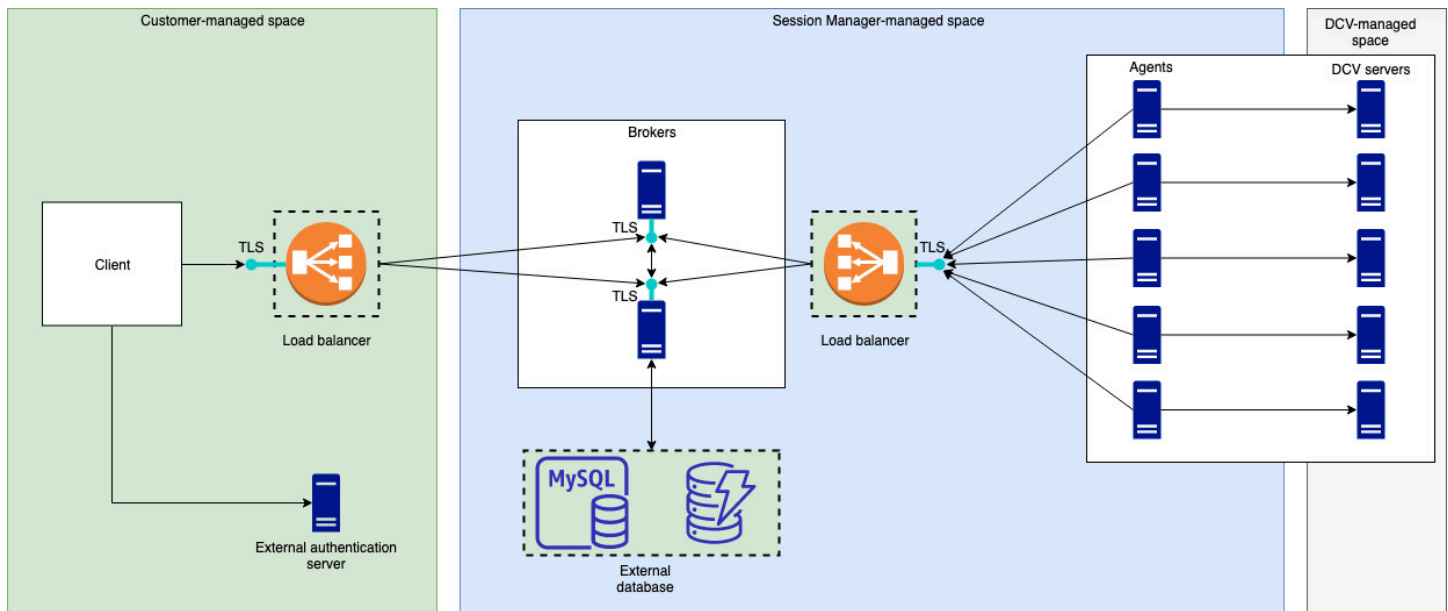
Agen dan Broker Session Manager NICE DCV memiliki persyaratan sebagai berikut.

	Pialang	Agen
Sistem operasi	<ul style="list-style-type: none"> <li>• Amazon Linux 2</li> <li>• CentOS 7.6 atau yang lebih baru</li> <li>• CentOS Aliran 8</li> <li>• CentOS Aliran 9</li> <li>• RHEL 7.6 atau yang lebih baru</li> <li>• RHEL 8.x</li> <li>• RHEL 9.x</li> <li>• Rocky Linux 8.5 atau yang lebih baru</li> <li>• Linux berbatu 9.x</li> <li>• Ubuntu 20.04</li> <li>• Ubuntu 22.04</li> </ul>	<ul style="list-style-type: none"> <li>• Windows <ul style="list-style-type: none"> <li>• Windows Server 2022</li> <li>• Windows Server 2019</li> <li>• Windows Server 2016</li> </ul> </li> <li>• Server Linux <ul style="list-style-type: none"> <li>• Amazon Linux 2</li> <li>• CentOS 7.6 atau yang lebih baru</li> <li>• CentOS Aliran 8</li> <li>• CentOS Aliran 9</li> <li>• RHEL 7.6 atau yang lebih baru</li> <li>• RHEL 8.x</li> <li>• RHEL 9.x</li> <li>• Rocky Linux 8.5 atau yang lebih baru</li> <li>• Linux berbatu 9.x</li> <li>• Ubuntu 20.04</li> <li>• Ubuntu 22.04</li> </ul> </li> </ul>

	Pialang	Agen
		<ul style="list-style-type: none"> <li>• SUSE Linux Enterprise 12 dengan SP4 atau yang lebih baru</li> <li>• SUSE Linux Perusahaan 15</li> </ul>
Arsitektur	<ul style="list-style-type: none"> <li>• 64-bit x86</li> <li>• LENGAN 64-bit</li> </ul>	<ul style="list-style-type: none"> <li>• 64-bit x86</li> <li>• ARM 64-bit (Amazon Linux 2, CentOS 7.x/8.x/9.x, RHEL 7.x/8.x/9.x dan Rocky 8.x/9.x saja)</li> <li>• ARM 64-bit (Ubuntu 22.04)</li> </ul>
Memori	8 GB	4 GB
Versi NICE DCV	NICE DCV 2020.2 dan setelahnya	NICE DCV 2020.2 dan setelahnya
Persyaratan tambahan	Java 11	-

## Persyaratan jaringan dan konektivitas

Diagram berikut memberikan gambaran tingkat tinggi dari jaringan Manajer Sesi dan persyaratan konektivitas.



Broker harus diinstal pada host terpisah, tetapi harus memiliki konektivitas jaringan dengan Agen di server NICE DCV. Jika Anda memilih untuk memiliki beberapa Broker untuk meningkatkan ketersediaan, maka Anda harus menginstal dan mengkonfigurasi setiap broker pada host terpisah, dan menggunakan satu atau lebih penyeimbang beban untuk mengelola lalu lintas antara klien dan Broker, dan Broker dan Agen. Pialang juga harus dapat berkomunikasi satu sama lain untuk bertukar informasi tentang server dan sesi NICE DCV. Broker dapat menyimpan kunci dan data status mereka pada database eksternal dan memiliki informasi ini tersedia setelah reboot atau penghentian. Ini membantu mengurangi risiko kehilangan informasi Broker penting dengan menyimpannya di database eksternal. Anda dapat mengambilnya nanti. Jika Anda memilih untuk memilikinya, maka Anda harus mengatur database eksternal dan mengkonfigurasi broker. DynamoDB, MariaDB, dan MySQL didukung. Anda dapat menemukan parameter konfigurasi yang tercantum pada [File Konfigurasi Broker](#).

Agen harus dapat memulai koneksi HTTPS dua arah yang aman, persisten, dan dua arah dengan Broker.

Klien Anda, atau aplikasi frontend, harus dapat mengakses Broker untuk memanggil API. Klien juga harus dapat mengakses server otentikasi Anda.

# Menyiapkan Manajer Sesi NICE DCV

Bagian berikut menjelaskan cara menginstal Session Manager dengan satu Broker dan beberapa Agen. Anda dapat menggunakan beberapa Broker untuk meningkatkan skalabilitas dan kinerja. Untuk informasi selengkapnya, lihat [Manajer Sesi Penskalaan](#).

Untuk menyiapkan Manajer Sesi NICE DCV, lakukan hal berikut:

## Langkah-langkah

- [Langkah 1: Siapkan server NICE DCV](#)
- [Langkah 2: Siapkan Broker Manajer Sesi NICE DCV](#)
- [Langkah 3: Siapkan Agen Manajer Sesi NICE DCV](#)
- [Langkah 4: Konfigurasi server NICE DCV untuk menggunakan Broker sebagai server otentikasi](#)
- [Langkah 5: Verifikasi instalasi](#)

## Langkah 1: Siapkan server NICE DCV

Anda harus memiliki armada server NICE DCV yang ingin Anda gunakan Session Manager. Untuk informasi lebih lanjut tentang menginstal server NICE DCV, lihat [Menginstal server NICE DCV di Panduan Administrator DCV NICE](#).

Di server Linux NICE DCV, Session Manager menggunakan pengguna layanan lokal bernama `dcvsmagent`. Pengguna ini secara otomatis dibuat ketika Agen Manajer Sesi diinstal. Anda harus memberikan hak administrator pengguna layanan ini untuk NICE DCV sehingga dapat melakukan tindakan atas nama pengguna lain. Untuk memberikan hak administrator pengguna layanan Session Manager, lakukan hal berikut:

Untuk menambahkan pengguna layanan lokal untuk server Linux NICE DCV

1. Buka `/etc/dcv/dcv.conf` menggunakan editor teks pilihan Anda.
2. Tambahkan `administrators` parameter ke `[security]` bagian dan tentukan pengguna Session Manager. Sebagai contoh:

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

3. Simpan dan tutup file .
4. Berhenti dan restart server NICE DCV.

Session Manager hanya mampu membuat sesi NICE DCV atas nama pengguna yang sudah ada di server NICE DCV. Jika permintaan dibuat untuk membuat sesi untuk pengguna yang tidak ada, permintaan gagal. Oleh karena itu, Anda harus memastikan bahwa setiap pengguna akhir yang dituju memiliki pengguna sistem yang valid di server NICE DCV.

#### Tip

Jika Anda berniat menggunakan beberapa host Broker atau server NICE DCV dengan Agen, kami sarankan Anda mengonfigurasi hanya satu Broker dan satu server NICE DCV dengan Agen dengan melakukan langkah-langkah berikut, membuat Amazon Machine Images (AMI) dari host dengan konfigurasi yang lengkap, dan kemudian menggunakan AMI untuk meluncurkan Broker yang tersisa dan server NICE DCV. Atau, Anda dapat menggunakan AWS Systems Manager untuk menjalankan perintah pada beberapa instance dari jarak jauh.

## Langkah 2: Siapkan Broker Manajer Sesi NICE DCV

Broker harus diinstal pada host Linux. Untuk informasi selengkapnya tentang distribusi Linux yang didukung, lihat [Persyaratan Manajer Sesi NICE DCV](#). Instal Broker pada host yang terpisah dari Agen dan host server NICE DCV. Host dapat diinstal pada jaringan pribadi yang berbeda, tetapi harus dapat terhubung dan berkomunikasi dengan Agen.

Untuk menginstal dan memulai Broker

1. Connect ke host tempat Anda ingin menginstal Broker.
2. Paket-paket tersebut ditandatangani secara digital dengan tanda tangan GPG yang aman. Untuk memungkinkan manajer paket memverifikasi tanda tangan paket, Anda harus mengimpor kunci NICE GPG. Jalankan perintah berikut untuk mengimpor kunci NICE GPG.
  - Amazon Linux 2, RHEL, CentOS, dan 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. Unduh paket instalasi.

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

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

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

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

- Ubuntu 20.04

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

- Ubuntu 22.04

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

### 4. Instal paket .

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

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

- RHEL 8.x, Streaming CentOS 8, dan Rocky Linux 8.x

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

- Ubuntu 20.04

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

- Ubuntu 22.04

```
$ sudo apt install -y ./nice-dcv-session-manager-broker_2023.1.410-1_all.ubuntu2204.deb
```

5. Periksa apakah versi lingkungan Java default adalah 11

```
$ java -version
```

Jika tidak, Anda dapat secara eksplisit mengatur direktori home Java yang akan digunakan Broker untuk menargetkan versi Java yang tepat. Ini dilakukan pengaturan parameter `broker-java-home` dalam file konfigurasi Broker. Untuk informasi selengkapnya, lihat [File Konfigurasi Broker](#).

6. Mulai layanan Broker dan pastikan bahwa itu dimulai secara otomatis setiap kali instance dimulai.

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

7. Tempatkan salinan sertifikat yang ditandatangani sendiri Broker di direktori pengguna Anda. Anda akan membutuhkannya ketika Anda menginstal Agen di langkah berikutnya.

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

## Langkah 3: Siapkan Agen Manajer Sesi NICE DCV

Agen harus diinstal pada semua host server NICE DCV di armada. Agen dapat diinstal pada server Windows dan Linux. Untuk informasi selengkapnya tentang sistem operasi yang didukung, lihat [Persyaratan Manajer Sesi NICE DCV](#).

### Prasyarat

Server NICE DCV harus diinstal pada host sebelum menginstal Agen.



## Linux host

### Note

[Agen Manajer Sesi tersedia untuk distribusi dan arsitektur Linux yang tercantum dalam Persyaratan:](#)

Instruksi berikut adalah untuk menginstal Agen pada host 64-bit x86. Untuk menginstal Agen pada host ARM 64-bit, ganti `x86_64` dengan `aarch64` Untuk Ubuntu, ganti `amd64` dengan `arm64`

Untuk menginstal Agen pada host Linux

1. Paket-paket tersebut ditandatangani secara digital dengan tanda tangan GPG yang aman. Untuk memungkinkan manajer paket memverifikasi tanda tangan paket, Anda harus mengimpor kunci NICE GPG. Jalankan perintah berikut untuk mengimpor kunci NICE GPG.

- Amazon Linux 2, RHEL, CentOS, dan 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. Unduh paket instalasi.

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

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

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

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

- Ubuntu 20.04

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

- Ubuntu 22.04

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

- SUSE Linux Perusahaan 12

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

- SUSE Linux Perusahaan 15

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

### 3. Instal paket .

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

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

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

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

- Ubuntu 20.04

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

- Ubuntu 22.04

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


- SUSE Linux Perusahaan 12

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

- SUSE Linux Perusahaan 15

```
$ sudo zypper install ./nice-dcv-session-manager-agent-2023.1.732-1.sles15.x86_64.rpm
```

4. Tempatkan salinan sertifikat yang ditandatangani sendiri Broker (yang Anda salin pada langkah sebelumnya) di `/etc/dcv-session-manager-agent/` direktori pada Agen.
5. Buka `/etc/dcv-session-manager-agent/agent.conf` menggunakan editor teks pilihan Anda dan lakukan hal berikut.
  - Untuk `broker_host`, tentukan nama DNS dari host tempat Broker diinstal.

 Important

Jika Broker berjalan pada instans Amazon EC2, `broker_host` Anda harus menentukan alamat Ipv4 pribadi instans tersebut.

- (Opsional) Untuk `broker_port`, tentukan port untuk berkomunikasi dengan Broker. Secara default Agen dan Broker berkomunikasi melalui port 8445. Hanya ubah ini jika Anda perlu menggunakan port yang berbeda. Jika Anda mengubahnya, pastikan Broker dikonfigurasi untuk menggunakan port yang sama.
- Untuk `ca_file`, tentukan jalur lengkap file sertifikat yang Anda salin di langkah sebelumnya. Sebagai contoh:

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

Atau, jika Anda ingin menonaktifkan verifikasi TLS, atur `tls_strict` ke `false`.

6. Simpan dan tutup file .
7. Jalankan perintah berikut untuk memulai Agen.

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

## Windows host

Untuk menginstal Agen pada host Windows

1. Unduh [penginstal Agen](#).
2. Jalankan pemasang. Pada layar Selamat Datang, pilih Berikutnya.
3. Pada layar EULA, baca perjanjian lisensi dengan cermat, dan jika Anda setuju, pilih Saya menerima persyaratan dan pilih Berikutnya.
4. Untuk memulai instalasi, pilih Instal.
5. Tempatkan salinan sertifikat yang ditandatangani sendiri Broker (yang Anda salin pada langkah sebelumnya) di C:\Program Files\NICE\DCVSessionManagerAgent\conf\ folder pada Agen.
6. Buka C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf menggunakan editor teks pilihan Anda, lalu lakukan hal berikut:
  - Untuk `broker_host`, tentukan nama DNS dari host tempat Broker diinstal.

### Important

Jika Broker berjalan pada instans Amazon EC2, `broker_host` Anda harus menentukan alamat IPv4 pribadi instans tersebut.

- (Opsional) Untuk `broker_port`, tentukan port untuk berkomunikasi dengan Broker. Secara default Agen dan Broker berkomunikasi melalui port 8445. Hanya ubah ini jika Anda perlu menggunakan port yang berbeda. Jika Anda mengubahnya, pastikan Broker dikonfigurasi untuk menggunakan port yang sama.
- Untuk `ca_file`, tentukan jalur lengkap file sertifikat yang Anda salin di langkah sebelumnya. Sebagai contoh:

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

Atau, jika Anda ingin menonaktifkan verifikasi TLS, atur `tls_strict` ke `false`.

7. Simpan dan tutup file .
8. Hentikan dan mulai ulang layanan Agen agar perubahan diterapkan. Jalankan perintah berikut di command prompt.

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

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

## Langkah 4: Konfigurasi server NICE DCV untuk menggunakan Broker sebagai server otentikasi

Konfigurasi server NICE DCV untuk menggunakan Broker sebagai server otentikasi eksternal untuk memvalidasi token koneksi klien. Anda juga harus mengonfigurasi server NICE DCV untuk mempercayai CA yang ditandatangani sendiri Broker.

### Linux NICE DCV server

Untuk menambahkan pengguna layanan lokal untuk server Linux NICE DCV

1. Buka `/etc/dcv/dcv.conf` menggunakan editor teks pilihan Anda.
2. Tambahkan `auth-token-verifier` parameter `ca-file` dan ke `[security]` bagian.

Untuk `ca-file`, tentukan jalur ke CA yang ditandatangani sendiri Broker yang Anda salin ke host pada langkah sebelumnya.

Untuk `auth-token-verifier`, tentukan URL untuk verifikasi token pada Broker dalam format berikut: `https://broker_ip_or_dns:port/agent/validate-authentication-token`. Tentukan port yang digunakan untuk komunikasi Broker-Agent, yaitu 8445 secara default. Jika Anda menjalankan Broker pada instans Amazon EC2, Anda harus menggunakan DNS pribadi atau alamat IP pribadi.

Sebagai contoh

```
[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. Simpan dan tutup file .

4. Berhenti dan restart server NICE DCV. Untuk informasi lebih lanjut, lihat [Menghentikan Server DCV NICE dan Memulai Server DCV NICE di Panduan Administrator DCV NICE](#).

## Windows NICE DCV server

Di Windows, server NICE DCV

1. Buka Windows Registry Editor dan arahkan ke kunci HKEY\_USERS/S-1-5-18/Software/Gsettings/com/niceSoftware/DCV/Security/.
2. Buka parameter ca-file. Untuk data Nilai, tentukan jalur ke CA yang ditandatangani sendiri Broker yang Anda salin ke host pada langkah sebelumnya.

### Note

Jika parameter tidak ada, buat parameter string baru dan beri nama ca-file.

3. Buka auth-token-verifierparameter-nya. Untuk data Nilai, tentukan URL untuk verifikasi token pada Broker dalam format berikut: `https://broker_ip_or_dns:port/agent/validate-authentication-token`. Tentukan port yang digunakan untuk komunikasi Broker-Agent, yaitu 8445 secara default. Jika Anda menjalankan Broker pada instans Amazon EC2, Anda harus menggunakan DNS pribadi atau alamat IP pribadi.

### Note

Jika parameter tidak ada, buat parameter string baru dan beri nama auth-token-verifier.

4. Pilih OK dan tutup Windows Registry Editor.
5. Berhenti dan restart server NICE DCV. Untuk informasi lebih lanjut, lihat [Menghentikan Server DCV NICE dan Memulai Server DCV NICE di Panduan Administrator DCV NICE](#).

## Langkah 5: Verifikasi instalasi

Topik

- [Verifikasi Agen](#)
- [Verifikasi Broker](#)

## Verifikasi Agen

Setelah Anda menginstal Broker dan Agen, pastikan Agen berjalan dan dapat terhubung ke Broker.

### Host Agen Linux

Perintah untuk menjalankan tergantung pada versinya.

- Sejak versi 2022.0

Dari host Agen, jalankan perintah berikut:

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

- Versi sebelum 2022.0

Dari host Agen, jalankan perintah berikut, dan tentukan tahun, bulan, dan hari saat ini.

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

Sebagai contoh

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

Jika Agen berjalan dan dapat terhubung ke broker, perintah harus kembali success.

Jika perintah mengembalikan output yang berbeda, periksa file log Agen untuk informasi lebih lanjut. File log terletak di sini: `/var/log/dcv-session-manager-agent/`.

### Tuan rumah Agen Windows

Buka file log Agen, yang terletak di `C:\ProgramData\NICE\DCVSessionManagerAgent\log`.

Jika file log menyertakan baris yang mirip dengan yang di bawah ini, Agen berjalan dan dapat terhubung ke 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}"
```

Jika file log Anda tidak memiliki baris yang sama, periksa file log untuk kesalahan.

## Verifikasi Broker

Setelah Anda menginstal Broker dan Agen, pastikan Broker Anda berjalan dan dapat dijangkau dari pengguna dan aplikasi front-end Anda.

Dari komputer yang seharusnya bisa menjangkau Broker, jalankan perintah berikut:

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

Jika verifikasi berhasil, Broker mengembalikan yang berikut:

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



# Mengkonfigurasi Manajer Sesi NICE DCV

Bagian ini menjelaskan cara melakukan konfigurasi lanjutan untuk Manajer Sesi.

Topik

- [Manajer Sesi Penskalaan](#)
- [Menggunakan tag untuk menargetkan server NICE DCV](#)
- [Mengonfigurasi server otorisasi eksternal](#)
- [Mengkonfigurasi ketekunan broker](#)
- [Mengintegrasikan dengan Gateway Koneksi NICE DCV](#)
- [Mengintegrasikan dengan Amazon CloudWatch](#)

## Manajer Sesi Penskalaan

Untuk mengaktifkan ketersediaan tinggi dan meningkatkan kinerja, Anda dapat mengonfigurasi Manajer Sesi untuk menggunakan beberapa Agen dan Broker. Jika Anda berniat menggunakan beberapa Agen dan Broker, kami sarankan Anda menginstal dan mengonfigurasi hanya satu host Agen dan Broker, membuat Gambar Mesin Amazon (AMI) dari host tersebut, dan kemudian meluncurkan host yang tersisa dari AMI.

Secara default, Session Manager mendukung penggunaan beberapa Agen tanpa konfigurasi tambahan. Namun, jika Anda berniat menggunakan beberapa Broker, Anda harus menggunakan penyeimbang beban untuk menyeimbangkan lalu lintas antara klien frontend dan Broker, dan antara Broker dan Agen. Pengaturan dan konfigurasi penyeimbang beban sepenuhnya dimiliki dan dikelola oleh Anda.

Bagian berikut menjelaskan cara mengkonfigurasi Session Manager untuk menggunakan beberapa host dengan Application Load Balancer.

Langkah-langkah

- [Langkah 1: Buat profil instance](#)
- [Langkah 2: Siapkan sertifikat SSL untuk penyeimbang beban](#)
- [Langkah 3: Buat penyeimbang beban aplikasi Broker](#)
- [Langkah 4: Luncurkan Broker](#)
- [Langkah 5: Buat penyeimbang beban aplikasi Agen](#)

- [Langkah 6: Luncurkan Agen](#)

## Langkah 1: Buat profil instance

Anda harus melampirkan profil instans ke host Broker dan Agen yang memberi mereka izin untuk menggunakan Elastic Load Balancing API. Untuk informasi selengkapnya, lihat [peran IAM untuk Amazon EC2](#) di Panduan Pengguna Amazon EC2.

### Buat profil instans

1. Buat peran AWS Identity and Access Management (IAM) yang mendefinisikan izin untuk digunakan dalam profil instance. Gunakan kebijakan kepercayaan berikut:

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

Kemudian lampirkan kebijakan berikut:

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

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

Untuk informasi selengkapnya, lihat [Membuat peran IAM](#) di Panduan Pengguna IAM.

2. Buat profil instance baru. Untuk informasi selengkapnya, lihat [create-instance-profile](#) di Command Reference.AWS CLI
3. Tambahkan peran IAM ke profil instance. Untuk informasi selengkapnya, lihat [menambahkan role-to-instance-profile](#) di Command Reference.AWS CLI
4. Lampirkan profil instance ke host Broker. Untuk informasi selengkapnya, lihat [Melampirkan peran IAM ke instans](#) di Panduan Pengguna Amazon EC2.

## Langkah 2: Siapkan sertifikat SSL untuk penyeimbang beban

Saat Anda menggunakan HTTPS untuk listener penyeimbang beban, Anda harus men-deploy sertifikat SSL pada penyeimbang beban. Penyeimbang beban menggunakan sertifikat untuk mengakhiri koneksi dan mendekripsi permintaan dari klien sebelum mengirimkannya mereka ke target.

Untuk menyiapkan sertifikat SSL

1. Buat otoritas sertifikat pribadi (CA) AWS Certificate Manager Private Certificate Authority (ACM PCA). Untuk informasi selengkapnya, lihat [Prosedur untuk Membuat CA](#) di Panduan Pengguna Otoritas AWS Certificate Manager Private Certificate Authority.
2. Instal CA. Untuk informasi selengkapnya, lihat [Menginstal Sertifikat Root CA](#) di Panduan Pengguna Otoritas AWS Certificate Manager Private Certificate Authority.
3. Minta sertifikat pribadi baru yang ditandatangani oleh CA. Untuk nama domain, gunakan `*.region.elb.amazonaws.com` dan tentukan Wilayah tempat Anda ingin membuat penyeimbang beban. Untuk informasi selengkapnya, lihat [Meminta Sertifikat Pribadi](#) di Panduan Pengguna Otoritas AWS Sertifikat Pribadi Certificate Manager Certificate.

## Langkah 3: Buat penyeimbang beban aplikasi Broker

Buat penyeimbang beban aplikasi untuk menyeimbangkan lalu lintas antara klien front-end Anda dan Broker.

Untuk membuat penyeimbang beban

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.

Di panel navigasi, pilih Load Balancers dan kemudian pilih Create Load Balancer. Untuk tipe load balancer, pilih Application Load Balancer.

2. Untuk Langkah 1: Konfigurasi Penyeimbang Beban, lakukan hal berikut:
  - a. Untuk Nama, masukkan nama deskriptif untuk penyeimbang beban.
  - b. Untuk Skema, pilih yang menghadap ke internet.
  - c. Untuk Protokol Load Balancer, pilih HTTPS, dan untuk Port Load Balancer, masukkan 8443
  - d. Untuk VPC, pilih VPC yang akan digunakan dan kemudian pilih semua subnet di VPC itu.
  - e. Pilih Selanjutnya.
3. Untuk Langkah 2: Konfigurasikan Pengaturan Keamanan, lakukan hal berikut:
  - a. Untuk jenis Sertifikat, pilih Pilih sertifikat dari ACM.
  - b. Untuk nama Sertifikat, pilih sertifikat pribadi yang Anda minta sebelumnya.
  - c. Pilih Selanjutnya.
4. Untuk Langkah 3: Konfigurasikan Grup Keamanan, buat grup keamanan baru, atau pilih grup keamanan yang ada yang memungkinkan lalu lintas masuk dan keluar antara klien frontend Anda dan Broker melalui HTTPS dan port 8443.

Pilih Selanjutnya.

5. Untuk Langkah 4: Konfigurasikan Routing, lakukan hal berikut:
  - a. Untuk grup Target, pilih Grup target baru.
  - b. Untuk Name, masukkan nama untuk grup target.
  - c. Untuk tipe Target, pilih Instance.
  - d. Untuk Protokol, pilih HTTPS. Untuk Port, masukkan 8443. Untuk versi Protokol, pilih HTTP1.

- e. Untuk pemeriksaan kesehatan Protokol, pilih HTTPS, dan untuk Path, masukkan/health.
  - f. Pilih Selanjutnya.
6. Untuk Langkah 5: Daftarkan Target, pilih Berikutnya.
  7. Pilih Buat.

## Langkah 4: Luncurkan Broker

Buat Broker awal dan konfigurasi untuk menggunakan penyeimbang beban, buat AMI dari Broker, dan kemudian gunakan AMI untuk meluncurkan Broker yang tersisa. Ini memastikan bahwa semua Broker dikonfigurasi untuk menggunakan CA yang sama dan konfigurasi penyeimbang beban yang sama.

Untuk meluncurkan Broker

1. Luncurkan dan konfigurasi host Broker awal. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi Broker, lihat [Langkah 2: Siapkan Broker Manajer Sesi NICE DCV](#).

### Note

Sertifikat yang ditandatangani sendiri oleh broker tidak diperlukan karena kami menggunakan penyeimbang beban aplikasi.

2. Hubungkan ke Broker, buka `/etc/dcv-session-manager-broker/session-manager-broker.properties` menggunakan editor teks pilihan Anda, dan lakukan hal berikut:
  - a. Komentari `broker-to-broker-discovery-addresses` parameter dengan menempatkan hash (#) di awal baris.
  - b. Untuk `broker-to-broker-discovery-aws-region`, masukkan Wilayah tempat Anda membuat penyeimbang beban aplikasi.
  - c. Untuk `broker-to-broker-discovery-aws-alb-target-group-arn`, masukkan ARN dari kelompok sasaran yang terkait dengan penyeimbang beban Broker.
  - d. Simpan dan tutup file .
3. Hentikan contoh Broker.
4. Buat AMI dari instance Broker yang dihentikan. Untuk informasi selengkapnya, lihat [Membuat AMI Linux dari instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
5. Gunakan AMI untuk meluncurkan Broker yang tersisa.

6. Tetapkan profil instans yang Anda buat untuk semua instance Broker.
7. Tetapkan grup keamanan yang memungkinkan Broker to Broker dan Broker memuat lalu lintas jaringan penyeimbang ke semua instance Broker. Untuk informasi selengkapnya tentang port jaringan, lihat [File Konfigurasi Broker](#).
8. Daftarkan semua instance Broker sebagai target untuk load balancer Broker. Untuk informasi selengkapnya, lihat [Mendaftarkan target dengan grup target Anda](#) di Panduan Pengguna untuk Penyeimbang Beban Aplikasi.

## Langkah 5: Buat penyeimbang beban aplikasi Agen

Buat penyeimbang beban aplikasi untuk menyeimbangkan Agen dan Pialang.

Untuk membuat penyeimbang beban

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.

Di panel navigasi, pilih Load Balancers dan kemudian pilih Create Load Balancer. Untuk tipe load balancer, pilih Application Load Balancer.

2. Untuk Langkah 1: Konfigurasi Penyeimbang Beban, lakukan hal berikut:
  - a. Untuk Nama, masukkan nama deskriptif untuk penyeimbang beban.
  - b. Untuk Skema, pilih yang menghadap ke internet.
  - c. Untuk Protokol Load Balancer, pilih HTTPS, dan untuk Port Load Balancer, masukkan 8445
  - d. Untuk VPC, pilih VPC yang akan digunakan dan kemudian pilih semua subnet di VPC itu.
  - e. Pilih Selanjutnya.
3. Untuk Langkah 2: Konfigurasi Pengaturan Keamanan, lakukan hal berikut:
  - a. Untuk jenis Sertifikat, pilih Pilih sertifikat dari ACM.
  - b. Untuk nama Sertifikat, pilih sertifikat pribadi yang Anda minta sebelumnya.
  - c. Pilih Selanjutnya.
4. Untuk Langkah 3: Konfigurasi Grup Keamanan, buat grup keamanan baru, atau pilih grup keamanan yang ada yang memungkinkan lalu lintas masuk dan keluar Agen dan Broker melalui HTTPS dan port 8445.

Pilih Selanjutnya.

5. Untuk Langkah 4: Konfigurasi Routing, lakukan hal berikut:
  - a. Untuk grup Target, pilih Grup target baru.
  - b. Untuk Name, masukkan nama untuk grup target.
  - c. Untuk tipe Target, pilih Instance.
  - d. Untuk Protokol, pilih HTTPS. Untuk Port, masukkan 8445. Untuk versi Protokol, pilih HTTP1.
  - e. Untuk pemeriksaan kesehatan Protokol, pilih HTTPS, dan untuk Path, masukkan /health.
  - f. Pilih Selanjutnya.
6. Untuk Langkah 5: Daftarkan Target, pilih semua instans Broker dan pilih Tambahkan ke terdaftar. Pilih Berikutnya: Tinjau.
7. Pilih Buat.

## Langkah 6: Luncurkan Agen

Buat Agen awal dan konfigurasi untuk menggunakan penyeimbang beban, buat AMI dari Agen, lalu gunakan AMI untuk meluncurkan Agen yang tersisa. Ini memastikan bahwa semua Agen dikonfigurasi untuk menggunakan konfigurasi penyeimbang beban yang sama.

Untuk meluncurkan Agen

1. Siapkan server NICE DCV. Untuk informasi selengkapnya, lihat [Langkah 1: Siapkan server NICE DCV](#).
2. Tempatkan salinan kunci publik CA yang dibuat di [Langkah 2: Siapkan sertifikat SSL untuk penyeimbang beban](#). Pilih atau buat direktori yang dapat dibaca oleh pengguna mana pun. File kunci publik CA harus dapat dibaca oleh pengguna mana pun juga.
3. Instal dan konfigurasi Agen. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi Agen, lihat [Langkah 3: Siapkan Agen Manajer Sesi NICE DCV](#).

### Important

Saat memodifikasi file konfigurasi Agen:

- untuk `broker_host` parameter, masukkan DNS Agent load balancer

- untuk `ca_file` parameter, masukkan path ke file kunci publik CA yang dibuat pada langkah sebelumnya

4. Konfigurasi server NICE DCV untuk menggunakan Broker sebagai server otentikasi. Untuk informasi selengkapnya, lihat [Langkah 4: Konfigurasi server NICE DCV untuk menggunakan Broker sebagai server otentikasi](#).

#### Important

Saat memodifikasi file konfigurasi server NICE DCV:

- untuk `ca-file` parameter, masukkan jalur yang sama ke file kunci publik CA yang digunakan pada langkah sebelumnya
- *untuk `auth-token-verifier` parameter, gunakan `DNS Agent load balancer` untuk `broker_ip_or_dns`*

5. Hentikan contoh Agen.
6. Buat AMI dari instance Agen yang dihentikan. Untuk informasi selengkapnya, lihat [Membuat AMI Linux dari instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
7. Gunakan AMI untuk meluncurkan Agen yang tersisa dan tetapkan profil instans yang Anda buat untuk semuanya.
8. Tetapkan grup keamanan yang memungkinkan Agen memuat lalu lintas jaringan penyeimbang ke semua instance Agen. Untuk informasi selengkapnya tentang port jaringan, lihat [File Konfigurasi Agen](#).

## Menggunakan tag untuk menargetkan server NICE DCV

Anda dapat menetapkan tag khusus ke Agen Manajer Sesi untuk membantu mengidentifikasi dan mengkategorikan mereka dan server NICE DCV yang terkait dengannya. Saat membuat sesi NICE DCV baru, Anda dapat menargetkan sekelompok server NICE DCV berdasarkan tag yang ditugaskan ke Agen masing-masing. Untuk informasi selengkapnya tentang cara menargetkan server NICE DCV berdasarkan tag Agen, lihat [CreateSessionRequests](#) di Panduan Pengembang Session Manager.

Tag terdiri dari kunci tag dan pasangan nilai, dan Anda dapat menggunakan pasangan informasi apa pun yang masuk akal untuk kasus penggunaan atau lingkungan Anda. Anda dapat memilih untuk menandai Agen berdasarkan konfigurasi perangkat keras host mereka. Misalnya, Anda



dapat menandai semua Agen dengan host yang memiliki memori 4 GBram=4GB. Atau Anda dapat menandai Agen berdasarkan tujuan. Misalnya, Anda dapat menandai semua Agen yang berjalan di host produksi dengan `purpose=production`.

Untuk menetapkan tag ke Agen

1. Menggunakan editor teks pilihan Anda, membuat file baru dan memberikan nama deskriptif, misalnya `agent_tags.toml`. Jenis file harus `.toml`, dan isi file harus ditentukan dalam format file TOLL.
2. Dalam file, tambahkan setiap kunci tag baru dan pasangan nilai pada baris baru menggunakan `key=value` format. Misalnya:

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

3. Buka file konfigurasi Agen (`/etc/dcv-session-manager-agent/agent.conf` untuk Linux atau `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` Windows). Untuk `tags_folder`, dan tentukan path ke direktori tempat file tag berada.

Jika direktori berisi beberapa file tag, semua tag yang didefinisikan di seluruh file menerapkan Agen. File dibaca dalam urutan abjad. Jika beberapa file berisi tag dengan kunci yang sama, nilainya ditimpa dengan nilai dari file baca terakhir.

4. Simpan dan tutup file.
5. Berhenti dan mulai ulang Agen.

- Windows

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

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

- Linux

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

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

## Mengonfigurasi server otorisasi eksternal

Server otorisasi adalah server yang bertanggung jawab untuk mengautentikasi dan mengotorisasi SDK dan Agen klien.

Secara default, Session Manager menggunakan Broker sebagai server otorisasi untuk menghasilkan token akses OAuth 2.0 untuk SDK klien dan pernyataan perangkat lunak untuk Agen. Jika Anda menggunakan Broker sebagai server otorisasi, tidak diperlukan konfigurasi tambahan.

Anda dapat mengonfigurasi Pengelola Sesi untuk menggunakan Amazon Cognito sebagai server otorisasi eksternal, bukan Broker. Untuk informasi selengkapnya, lihat [Panduan Developer Amazon Cognito](#).

Menggunakan Amazon Cognito sebagai server otorisasi

1. Membuat kolam pengguna Amazon Cognito. Untuk informasi selengkapnya tentang kolam pengguna, lihat [Fitur Amazon Cognito](#) di Panduan Developer Amazon Cognito.

Gunakan [create-user-pool](#) perintah, dan tentukan nama pool dan Region untuk membuatnya.

Dalam contoh ini, kita beri nama `pooldcv-session-manager-client-app` dan kita membuatnya `us-east-1`.

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

Contoh keluaran

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

Catata `userPoolId`, Anda akan membutuhkannya di langkah berikutnya.

2. Buat domain baru untuk kolam pengguna Anda. Gunakan [create-user-pool-domain](#) perintah, dan tentukan nama domain dan kumpulan pengguna yang Anda buat di langkah sebelumnya. `userPoolId`

Dalam contoh ini, nama domain `mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE` dan kita membuatnya di `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
```

Contoh keluaran

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

Format domain pool pengguna adalah sebagai

berikut: `https://domain_name.auth.region.amazoncognito.com`. Dalam contoh ini, domain pool pengguna adalah `https://mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE.auth.us-east-1.amazoncognito.com`.

3. Membuat klien kolam pengguna. Gunakan [create-user-pool-client](#) perintah dan tentukan kumpulan pengguna yang Anda buat, nama untuk klien, dan Wilayah untuk membuatnya. `userPoolId` Juga, sertakan `--generate-secret` opsi untuk menentukan bahwa Anda ingin menghasilkan rahasia untuk klien kolam pengguna yang sedang dibuat.

Dalam hal ini, nama klien `dcv-session-manager-client-app` dan kami membuatnya di `us-east-1` Region.

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

## Contoh keluaran

```
{
  "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

Catat `ClientId` dan `ClientSecret`. Anda harus memberikan informasi ini kepada pengembang ketika mereka meminta token akses untuk permintaan API.

4. Buat server sumber daya OAuth2.0 baru untuk kumpulan pengguna. Sebuah server sumber daya adalah server untuk sumber daya yang dilindungi akses. Ini menangani permintaan otentikasi untuk token akses.

Gunakan [create-resource-server](#) perintah dan tentukan kumpulan pengguna, pengenal unik dan nama untuk server sumber daya, ruang lingkup, dan Wilayah tempat membuatnya. `userPoolId`

Dalam contoh ini, kita gunakan `dcv-session-manager` sebagai identifier dan nama, dan kita gunakan `sm_scope` sebagai nama lingkup dan deskripsi.

```
$ 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
```

## Contoh keluaran

```
{
  "ResourceServer": {
    "UserPoolId": "us-east-1_QLEXAMPLE",
```

```

    "Identifier": "dcv-session-manager",
    "Name": "dcv-session-manager",
    "Scopes": [
      {
        "ScopeName": "sm_scope",
        "ScopeDescription": "sm_scope"
      }
    ]
  }
}

```

## 5. Memperbarui klien kolam pengguna.

Gunakan perintah [update-user-pool-client](#). Tentukan `userPoolId` kumpulan pengguna, klien kumpulan pengguna, dan Wilayah. `ClientId` Untuk `--allowed-o-auth-flows`, tentukan `client_credentials` untuk menunjukkan bahwa klien harus mendapatkan token akses dari titik akhir token dengan menggunakan kombinasi ID klien dan rahasia klien. Untuk `--allowed-o-auth-scopes`, tentukan pengidentifikasi server sumber daya dan nama lingkup sebagai berikut: `resource_server_identifier/scope_name`. Sertakan protokol OAuth saat berinteraksi dengan kolam pengguna Cognito. `--allowed-o-auth-flows-user-pool-client`

```

$ 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

```

## Contoh keluaran

```

{
  "UserPoolClient": {
    "UserPoolId": "us-east-1_QLEXAMPLE",
    "ClientName": "dcv-session-manager-client-app",
    "ClientId": "219273hp6k2ut5cugg9EXAMPLE",
    "ClientSecret": "1vp5e8nec7cbf4m9me55mbmht91u61hlh0a78rq1qki11EXAMPLE",
    "LastModifiedDate": 1602512103.099,
    "CreationDate": 1602510291.498,
    "RefreshTokenValidity": 30,
    "AllowedOAuthFlows": [
      "client_credentials"
    ],
    "AllowedOAuthScopes": [

```

```

    "dcv-session-manager/sm_scope"
  ],
  "AllowedAuthFlowsUserPoolClient": true
}
}

```

### Note

Pool pengguna sekarang siap untuk menyediakan dan mengotentikasi token akses. Dalam contoh ini, URL untuk server otorisasi adalah `https://cognito-idp.us-east-1.amazonaws.com/us-east-1_QLEXAMPLE/.well-known/jwks.json`.

## 6. Uji konfigurasi.

```

$ 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"

```

### Contoh keluaran

```

{
  "access_token": "eyJraWQiOiJGQ0VaRFPJUUptT3NSaW41MmtqaDdEbTZYb0RnSTQ5b2VUT0cxUU1Q2VJPSIsImF1dG8iOiJkZXkiLCJhdGUiOiJkZXkiLCJ0eXBlIjoiYXV0IiwiaWF0Ijoi",
  "expires_in": 3600,
  "token_type": "Bearer"
}

```

## 7. Daftarkan server otorisasi eksternal untuk digunakan dengan broker dengan menggunakan `register-auth-server` perintah.

```

$ 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

```

Pengembang sekarang dapat menggunakan server untuk meminta token akses. Saat meminta token akses, dan URL server yang dihasilkan di sini. Untuk informasi selengkapnya tentang meminta token akses, lihat [Membuat token akses dan membuat permintaan API](#) di Panduan Pengembang Manajer Sesi NICE DCV.

## Mengkonfigurasi ketekunan broker

Broker Session Manager mendukung integrasi dengan database eksternal. Database eksternal memungkinkan Session Manager untuk bertahan data status dan kunci sehingga mereka tersedia setelah itu. Bahkan data broker didistribusikan melalui cluster, sehingga rentan terhadap kehilangan data jika host perlu reboot atau cluster dihentikan. Dengan fitur ini diaktifkan, Anda dapat menambah dan menghapus node broker. Juga, Anda dapat menghentikan cluster dan restart, tanpa perlu regenerasi kunci atau kehilangan informasi tentang yang NICE DCV Server terbuka atau tertutup.

Tipe informasi berikut dapat disetel untuk bertahan:

- Kunci untuk menyiapkan sesi untuk menjalin koneksi dengan klien
- Data sesi dalam penerbangan
- NICE DCV server

NICE DCV Session Manager mendukung database DynamoDB, MariaDB, dan MySQL. Anda harus menyiapkan dan mengonfigurasi salah satu database ini untuk menggunakan fitur ini. Jika mesin broker Anda di-host di Amazon EC2, sebaiknya gunakan DynamoDB sebagai database eksternal, karena tidak memerlukan pengaturan tambahan.

### Note

Anda mungkin dikenakan biaya tambahan saat menjalankan database eksternal. Untuk melihat informasi tentang harga DynamoDB, lihat [Harga untuk Kapasitas yang Disediakan](#).

## Konfigurasi broker untuk bertahan di DynamoDB

Konfigurasi broker untuk mulai menyimpan data mereka di DynamoDB:

1. Buka `/etc/dcv-session-manager-broker/session-manager-broker.properties` menggunakan editor teks yang Anda pilih dan lakukan suntingan berikut:

- `Setenable-persistence = true`
  - `Setpersistence-db = dynamodb`
  - Untuk `dynamodb-region` menentukan &aws; Wilayah di mana Anda ingin menyimpan tabel yang berisi data broker. Untuk daftar wilayah yang didukung, lihat [titik akhir layanan DynamoDB](#).
  - Untuk `dynamodb-table-rcu` tentukan jumlah Unit Kapasitas Baca (RCU) yang didukung setiap tabel. Untuk informasi selengkapnya tentang RCU, lihat [kapasitas yang disediakan DynamoDB](#).
  - Untuk `dynamodb-table-wcu` menentukan jumlah Write Capacity Units (WCU) yang didukung setiap tabel. Untuk info lebih lanjut tentang WCU, lihat [kapasitas yang disediakan DynamoDB](#).
  - Untuk `dynamodb-table-name-prefix` tentukan awalan yang ditambahkan ke setiap tabel DynamoDB (berguna untuk membedakan beberapa klaster broker menggunakan akun yang sama). Hanya karakter alfanumerik, titik, dasbor, dan garis bawah yang diizinkan.
2. Hentikan semua broker di klaster. Untuk setiap broker, jalankan perintah berikut:

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

3. Pastikan semua broker di cluster dihentikan dan kemudian restart semuanya. Mulai setiap broker mulai dengan jalankan perintah berikut:

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

Tuan rumah broker harus memiliki izin untuk memanggil API DynamoDB. Pada instans Amazon EC2, kredensi secara otomatis diambil menggunakan layanan metadata Amazon EC2. Jika Anda perlu menentukan kredensi yang berbeda, Anda dapat mengaturnya menggunakan salah satu teknik pengambilan kredensi yang didukung (seperti properti sistem Java atau variabel lingkungan). Untuk informasi selengkapnya, lihat [Memasok dan Mengambil &aws; Kredensial](#).

## Konfigurasi broker untuk bertahan di Mariadb/MySQL

### Note

`/etc/dcv-session-manager-broker/session-manager-broker.propertiesFile` ini berisi data sensitif. Secara default, akses tulisnya terbatas pada root dan akses bacanya dibatasi untuk root dan pengguna yang menjalankan Broker. Secara default, ini



adalah `dcvsmbroker` pengguna. Broker memeriksa saat startup bahwa file tersebut memiliki izin yang diharapkan.

Konfigurasi broker untuk mulai mempertahankan data mereka di MariaDB/MySQL:

1. Buka `/etc/dcv-session-manager-broker/session-manager-broker.properties` dengan editor teks pilihan Anda dan lakukan pengeditan berikut:

- `Setenable-persistence = true`
- `Setpersistence-db = mysql`
- `Setjdbc-connection-url =  
jdbc:mysql://<db_endpoint>:<db_port>/<db_name>?  
createDatabaseIfNotExist=true`

Dalam konfigurasi ini, `<db_endpoint>` adalah endpoint database, `<db_port>` adalah port database, dan `<db_name>` nama database.

- Untuk `jdbc-user` menentukan nama pengguna yang memiliki akses ke database.
  - Untuk `jdbc-password` menentukan password dari pengguna yang memiliki akses ke database.
2. Hentikan semua broker di kluster. Untuk setiap broker, jalankan perintah berikut:

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

3. Pastikan semua broker di cluster dihentikan, dan kemudian restart semuanya. Untuk setiap broker, jalankan perintah berikut:

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

## Mengintegrasikan dengan Gateway Koneksi NICE DCV

[NICE DCV Connection Gateway](#) adalah paket perangkat lunak yang dapat diinstal yang memungkinkan pengguna mengakses armada server NICE DCV melalui satu titik akses ke LAN atau VPC.

Jika infrastruktur Anda termasuk server NICE DCV yang dapat diakses melalui NICE DCV Connection Gateway, Anda dapat mengkonfigurasi Session Manager untuk mengintegrasikan NICE





3. Tempatkan salinan kunci pribadi NICE DCV Connection Gateway dan sertifikat yang ditandatangani sendiri di direktori pengguna Anda. Anda akan membutuhkannya ketika Anda mengaktifkan otentikasi klien TLS di NICE DCV Connection Gateway pada langkah berikutnya.

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

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

4. Kemudian buka `/etc/dcv-session-manager-broker/session-manager-broker.properties` menggunakan editor teks pilihan Anda dan lakukan hal berikut:
  - Aturenable-tls-client-auth-gateway ketruue
  - Tetapkangateway-to-broker-connector-trust-store-file ke jalur TrustStore file yang dibuat pada langkah sebelumnya
  - Aturgateway-to-broker-connector-trust-store-pass ke kata sandi yang digunakan untuk membuat TrustStore file di langkah sebelumnya
5. Kemudian jalankan perintah berikut untuk menghentikan dan me-restart Broker:

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

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

## Sisi Gateway Koneksi NICE DCV

- Silakan ikuti [bagian](#) dalam dokumentasi NICE DCV Connection Gateway.
  - gunakan jalur lengkap dari file sertifikat yang Anda salin pada langkah sebelumnya saat mengaturncert-file parameter di[resolver] bagian
  - gunakan path lengkap dari file kunci yang Anda salin pada langkah sebelumnya saat mengaturncert-key-file parameter di[resolver] bagian

## Manajer Sesi NICE DCV BAGUS Server DCV NICE - Pemetaan DNS

NICE DCV Connection Gateway memerlukan nama DNS server NICE DCV untuk terhubung ke instance server DCV. Bagian ini menggambarkan bagaimana Anda dapat menentukan file JSON yang berisi pemetaan antara setiap Server DCV dan nama DNS yang terkait.



**ServerId:**

Id dari jenis tertentu yang secara unik mengidentifikasi setiap server NICE DCV dalam jaringan.

**DnsNames:**

Objek yang berisi nama-nama DNS yang terkait dengan server NICE DCV objek ini akan berisi:

**InternalDnsNames:**

Nama DNS yang digunakan oleh NICE DCV Connection Gateway untuk terhubung ke instance.

Silakan gunakan perintah Session Manager Broker `CLIregister-server-dns-mapping` untuk memuat pemetaan dari file (halaman perintah referensi: [register-server-dns-mapping](#)) dan `danddescribe-server-dns-mappings` untuk daftar pemetaan saat ini dimuat di Session Manager Broker (halaman perintah referensi: [describe-server-dns-mappings](#)).

**Tetap**

Kami sangat menyarankan agar Anda mengaktifkan fitur persistensi Broker Manajer Sesi, untuk melindungi dari kerugian pemetaan ketika beberapa broker atau seluruh kluster turun. Untuk informasi selengkapnya tentang mengaktifkan persistensi data, lihat [Mengkonfigurasi Persistensi Broker](#)

## Mengintegrasikan dengan Amazon CloudWatch

Session Manager mendukung integrasi dengan Amazon CloudWatch for Brokers yang berjalan di instans Amazon EC2, dan juga Broker yang berjalan di host lokal.

Amazon CloudWatch memantau sumber daya Amazon Web Services (AWS) Anda dan aplikasi yang Anda jalankan AWS di secara waktu nyata. Anda dapat menggunakan CloudWatch untuk mengumpulkan dan menelusuri metrik, yang merupakan variabel yang dapat diukur untuk sumber daya dan aplikasi Anda. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).


Anda dapat mengonfigurasi Broker Manajer Sesi untuk mengirim data metrik berikut ke Amazon CloudWatch:

- `Number of DCV servers`—Jumlah server DCV yang dikelola oleh Broker.

- `Number of ready DCV servers`—Jumlah server DCV yang berada diREADY negara bagian yang dikelola oleh Broker.
- `Number of DCV sessions`—Jumlah sesi DCV yang dikelola oleh Broker.
- `Number of DCV console sessions`—Jumlah sesi konsol DCV yang dikelola oleh Broker.
- `Number of DCV virtual sessions`—Jumlah sesi virtual DCV yang dikelola oleh Broker.
- `Heap memory used`—Jumlah memori tumpukan yang digunakan oleh Broker.
- `Off-heap memory used`—Jumlah memori off-heap yang digunakan oleh Broker.
- `Describe sessions request time`—Jumlah waktu yang dibutuhkan untuk menyelesaikan permintaan `DescribeSessions` API.
- `Delete sessions request time`—Jumlah waktu yang dibutuhkan untuk menyelesaikan permintaan `DeleteSessions` API.
- `Create sessions request time`—Jumlah waktu yang dibutuhkan untuk menyelesaikan permintaan `CreateSessions` API.
- `Get session connection data request time`—Jumlah waktu yang dibutuhkan untuk menyelesaikan permintaan `GetSessionConnectionData` API.
- `Update session permissions request time`—Jumlah waktu yang dibutuhkan untuk menyelesaikan permintaan `UpdateSessionPermissions` API.

Untuk mengkonfigurasi Broker untuk mengirim data metrik ke Amazon CloudWatch

1. Buka `/etc/dcv-session-manager-broker/session-manager-broker.properties` menggunakan editor teks pilihan Anda dan lakukan hal berikut:
  - `Atunable-cloud-watch-metrics` ke `true`
  - Untuk `cloud-watch-region`, tentukan Wilayah tempat mengumpulkan data metrik.

 Note

Jika Broker Anda berjalan di instans Amazon EC2, parameter ini bersifat opsional. Wilayah diambil secara otomatis dari Instance Metadata Service (IMDS). Jika Anda menjalankan Broker di host lokal, parameter ini wajib dilakukan.

2. Berhenti dan restart Broker.

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

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

Tuan rumah Broker juga harus memiliki izin untuk memanggil `cloudwatch:PutMetricData` API. AWS kredensi dapat diambil menggunakan salah satu teknik pengambilan kredensi yang didukung. Untuk informasi selengkapnya, lihat [Memasok dan menelusuri AWS Kredensi](#).



# Meningkatkan Manajer Sesi NICE DCV

Topik berikut menjelaskan cara meng-upgrade Session Manager.

## Note

Kami sangat menyarankan agar Anda meningkatkan semua Agen Manajer Sesi sebelum meningkatkan Broker Manajer Sesi, untuk menghindari masalah ketidakcocokan jika fitur baru diperkenalkan.

Topik

- [Meningkatkan Agen Manajer Sesi NICE DCV](#)
- [Meningkatkan Broker Manajer Sesi NICE DCV](#)

# Meningkatkan Agen Manajer Sesi NICE DCV

Linux host

## Note

Petunjuk berikut adalah untuk menginstal Agen pada host 64-bit x86. *Untuk menginstal Agen pada host ARM 64-bit, untuk Amazon Linux, RHEL, dan Centos, ganti x86\_64 dengan, dan untuk Ubuntuarch64, ganti amd64 dengan. arm64*

Untuk memperbarui Agen pada host Linux

1. Jalankan perintah berikut untuk menghentikan Agen.

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

2. Unduh paket instalasi.
  - Amazon Linux 2, RHEL 7.x, dan CentOS 7.x

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

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

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

- Ubuntu 20.04

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

- SUSE Linux Perusahaan 12

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

- SUSE Linux Perusahaan 15

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

### 3. Instal paket .

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

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

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

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

- Ubuntu 20.04

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

- SUSE Linux Perusahaan 12

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

- SUSE Linux Perusahaan 15

```
$ sudo zypper install nice-dcv-session-manager-agent-2023.1.732-1.sles15.x86_64.rpm
```

4. Jalankan perintah berikut untuk memulai Agen.

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

## Windows host

Untuk memperbarui Agen pada host Windows

1. Hentikan layanan agen. Jalankan perintah berikut di command prompt.

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

2. Unduh [penginstal Agen](#).
3. Jalankan pemasang. Pada layar Selamat Datang, pilih Berikutnya.
4. Pada layar EULA, baca perjanjian lisensi dengan cermat, dan jika Anda setuju, pilih Saya menerima persyaratan dan pilih Berikutnya.
5. Untuk memulai instalasi, pilih Instal.
6. Mulai ulang layanan Agen. Jalankan perintah berikut di command prompt.

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

## Meningkatkan Broker Manajer Sesi NICE DCV

Untuk meng-upgrade Broker

1. Connect ke host tempat Anda ingin meng-upgrade Broker.
2. Hentikan layanan Broker.

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

### 3. Unduh paket instalasi.

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

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

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

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

- Ubuntu 20.04

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

### 4. Instal paket .

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

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

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

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

- Ubuntu 20.04

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

### 5. Mulai layanan Broker dan pastikan bahwa itu dimulai secara otomatis setiap kali instance dimulai.

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

## Referensi CLI Broker

Bagian ini menjelaskan cara menggunakan perintah antarmuka baris perintah Broker (CLI).

Gunakan perintah berikut jika Anda menggunakan server otentikasi eksternal untuk menghasilkan token akses OAuth 2.0:

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

Gunakan perintah berikut jika Anda menggunakan Session Manager Broker sebagai server otentikasi OAuth 2.0.

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

Gunakan perintah berikut untuk mengelola Agen Manajer Sesi.

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

Gunakan perintah berikut untuk mengelola file pemetaan nama DNS server DCV.

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

# register-auth-server

Mendaftarkan server otentikasi eksternal untuk digunakan dengan Broker.

Secara default, Session Manager menggunakan Broker sebagai server otentikasi untuk menghasilkan token akses OAuth 2.0. Jika Anda menggunakan Broker sebagai server otentikasi, tidak diperlukan konfigurasi tambahan.

Namun, jika Anda memilih untuk menggunakan server otentikasi eksternal, seperti Active Directory atau Amazon Cognito, Anda harus menggunakan perintah ini untuk mendaftarkan server otentikasi eksternal.

Topik

- [Sintaksis](#)
- [Ops](#)
- [Contoh](#)

## Sintaksis

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

## Ops

### **--url**

URL dari server otentikasi eksternal yang akan digunakan. Anda harus menambahkan `.well-known/jwks.json` ke URL server otentikasi.

Tipe: String

Wajib: Ya

## Contoh

Contoh berikut register server otentikasi eksternal dengan URL `https://my-auth-server.com/`.

## Perintah

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

## Keluaran

```
Jwk url registered.
```

## list-auth-servers

Daftar server otentikasi eksternal yang telah terdaftar.

### Topik

- [Sintaksis](#)
- [Output](#)
- [Contoh](#)

## Sintaksis

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

## Output

### Ur1s

URL dari server otentikasi eksternal terdaftar.

## Contoh

Contoh berikut mencantumkan semua server otentikasi eksternal yang telah terdaftar.

### Perintah

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



## Keluaran

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

## unregister-auth-server

Membatalkan registrasi server otentikasi eksternal. Setelah Anda membatalkan pendaftaran server otentikasi eksternal, server tidak dapat lagi digunakan untuk menghasilkan token akses OAuth 2.0.

### Topik

- [Sintaksis](#)
- [Opsis](#)
- [Output](#)
- [Contoh](#)

## Sintaksis

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

## Opsis

### **--url**

URL server otentikasi eksternal untuk membatalkan pendaftaran. Anda harus menambahkan `.well-known/jwks.json` ke URL server otentikasi.

Tipe: String

Wajib: Ya

## Output

### **Url**

URL dari server otentikasi eksternal terdaftar.

## Contoh

Contoh berikut register server otentikasi eksternal dengan URL `https://my-auth-server.com/`.

### Perintah

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

### Keluaran

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

## register-api-client

Mendaftarkan klien Session Manager dengan Broker dan menghasilkan kredensi klien yang dapat digunakan oleh klien untuk mengambil token akses OAuth 2.0, yang diperlukan untuk membuat permintaan API.

### Important

Pastikan Anda menyimpan kredensialnya di tempat yang aman. Mereka tidak bisa dipulihkan nanti.

Perintah ini hanya digunakan jika Broker digunakan sebagai server otentikasi OAuth 2.0.

### Topik

- [Sintaksis](#)
- [Opsi](#)
- [Output](#)
- [Contoh](#)

## Sintaksis

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

## Opsi

### **--name**

Nama unik yang digunakan untuk mengidentifikasi klien Session Manager.

Tipe: String

Wajib: Ya

## Output

### **client-id**

ID klien unik yang akan digunakan oleh klien Session Manager untuk mengambil token akses OAuth 2.0.

### **client-password**

Kata sandi yang akan digunakan oleh klien Session Manager untuk mengambil token akses OAuth 2.0.

## Contoh

Contoh berikut meregistrasikan klien bernama `my-sm-client`.

### Perintah

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

### Keluaran

```
client-id: 21cfe9cf-61d7-4c53-b1b6-cf248EXAMPLE  
client-password: NjVmZDRlN2ItNjNmYS00M2QxLWF1ZmMtZmNmMDNkMEXAMPLE
```

## describe-api-clients

Mencantumkan klien Manajer Sesi yang telah terdaftar di Broker.

## Topik

- [Sintaksis](#)
- [Output](#)
- [Contoh](#)

## Sintaksis

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

## Output

### **name**

Nama unik klien Manajer Sesi.

### **id**

ID unik klien Session Manager.

### **active**

Menunjukkan status klien Manajer Sesi. Jika klien aktif, nilainya adalah `true`; jika tidak, itu `false`.

## Contoh

Contoh berikut mencantumkan klien Manajer Sesi terdaftar.

### Perintah

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

### Keluaran

```
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

Menonaktifkan klien Manajer Sesi terdaftar. Klien Session Manager yang dinonaktifkan tidak dapat lagi menggunakan kredensialnya untuk mengambil token akses OAuth 2.0.

Topik

- [Sintaksis](#)
- [Ops](#)
- [Contoh](#)

### Sintaksis

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

### Ops

#### **--client -id**

ID klien klien Session Manager untuk menonaktifkan.

Tipe: String

Wajib: Ya

### Contoh

Contoh berikut menonaktifkan klien Session Manager dengan ID klien dari f855b54b-40d4-4769-b792-b727bEXAMPLE.

Perintah

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

## Keluaran

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

## renew-auth-server-api-kunci

Memperbaharui kunci publik dan pribadi yang digunakan oleh Broker untuk menandatangani token akses OAuth 2.0 yang diberikan kepada klien Session Manager. Jika Anda memperbarui kunci, maka Anda harus memberikan kunci pribadi baru kepada pengembang, karena diperlukan untuk membuat permintaan API.

### Topik

- [Sintaksis](#)
- [Contoh](#)

## Sintaksis

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

## Contoh

Contoh berikut memperbarui kunci publik dan privat.

### Perintah

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

### Keluaran

```
Keys renewed.
```

## generate-software-statement

Menghasilkan pernyataan perangkat lunak.

Agen harus terdaftar di Broker untuk memungkinkan komunikasi. Agen membutuhkan pernyataan perangkat lunak untuk mendaftar ke Broker. Setelah Agen memiliki pernyataan perangkat lunak,

ia dapat secara otomatis mendaftarkan dirinya dengan Broker dengan menggunakan [Protokol Pendaftaran Klien Dinamis OAuth 2.0](#). Setelah Agen terdaftar di Broker, ia menerima ID klien dan rahasia klien yang digunakan untuk mengotentikasi dengan Broker.

Broker dan Agen menerima dan menggunakan pernyataan perangkat lunak default saat pertama kali diinstal. Anda dapat terus menggunakan pernyataan perangkat lunak default, atau Anda dapat memilih untuk membuat yang baru. Jika Anda menghasilkan pernyataan perangkat lunak baru, Anda harus menempatkan pernyataan perangkat lunak ke dalam file baru pada Agen, dan kemudian menambahkan path file ke `agent . software_statement_path` parameter dalam `agent . conf` file. Setelah Anda melakukan ini, hentikan dan mulai ulang Agen sehingga dapat menggunakan pernyataan perangkat lunak baru untuk mendaftar ke Broker.

Topik

- [Sintaksis](#)
- [Output](#)
- [Contoh](#)

## Sintaksis

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

## Output

### **software-statement**

Pernyataan perangkat lunak.

## Contoh

Contoh berikut menghasilkan pernyataan perangkat lunak.

Perintah

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

Keluaran

```
software-statement:
```

```
ewogICJpZCIgOiAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkyMDcxIiwKICAiYWN0aXZlIiA6IHRydWUsCi
```

## describe-software-statements

Menjelaskan pernyataan perangkat lunak yang ada.

Topik

- [Sintaksis](#)
- [Output](#)
- [Contoh](#)

### Sintaksis

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

### Output

#### **software-statement**

Pernyataan perangkat lunak.

#### **issued-at**

Tanggal dan waktu saat perangkat lunak dibuat.

#### **is-active**

Status pernyataan perangkat lunak saat ini. `true` jika pernyataan perangkat lunak aktif; jika tidak `false`.

### Contoh

Contoh berikut menghasilkan pernyataan perangkat lunak.

Perintah

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





## Contoh

Contoh berikut menonaktifkan pernyataan perangkat lunak.

Perintah

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

Keluaran

```
Software statement  
EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj  
deactivated
```

## describe-agent-clients

Menjelaskan Agen yang terdaftar di Broker.

Topik

- [Sintaksis](#)
- [Output](#)
- [Contoh](#)

## Sintaksis

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

## Output

**name**

Nama Agen.

**id**

ID Agen.

## active

Keadaan Agen. true jika Agen aktif; jika tidak, itu false.

## Contoh

Contoh berikut menjelaskan Agen.

### Perintah

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

### Keluaran

```
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

Batalkan pendaftaran Agen dari Broker.

### Topik

- [Sintaksis](#)
- [Ops](#)

- [Contoh](#)

## Sintaksis

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

## Opsi

### **--client-id**

ID Agen untuk membatalkan pendaftaran.

Tipe: String

Wajib: Ya

## Contoh

Contoh berikut unregisters Agen.

### Perintah

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

### Keluaran

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

## register-server-dns-mappings

Daftarkan Server DCV - pemetaan nama DNS yang berasal dari file JSON.

## Sintaksis

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

## Opsi

### **--file-path**

Jalur file yang berisi Server DCV - nama DNS pemetaan.

Tipe: String

Wajib: Ya

## Contoh

Contoh berikut register Server DCV - DNS nama pemetaan dari file /tmp/mappings.json.

Perintah

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

Keluaran

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

## describe-server-dns-mappings

Jelaskan Server DCV yang tersedia saat ini - pemetaan nama DNS.

### Sintaksis

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

## Output

### **serverIdType**

Jenis Id server.

### **serverId**

ID unik Server.

## dnsNames

Nama dns internal dan eksternal

### **internalDnsNames**

Nama dns internal

### **externalDnsNames**

Nama dns eksternal

## Contoh

Contoh berikut mencantumkan Server DCV terdaftar - nama DNS pemetaan.

Perintah

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

Keluaran

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

# Referensi Referensi

Bagian ini menyediakan informasi tentang file konfigurasi Agen dan Broker.

Topik

- [File](#)
- [File konfigurasi agen](#)

## File

File konfigurasi Broker (`/etc/dcv-session-manager-broker/session-manager-broker.properties`) mencakup parameter yang dapat dikonfigurasi untuk menyesuaikan fungsionalitas Manajer Sesi. Anda dapat mengedit file konfigurasi menggunakan editor teks pilihan Anda.

### Note

`/etc/dcv-session-manager-broker/session-manager-broker.properties` berisi data sensitif. Secara default, akses tulisnya terbatas pada root dan akses bacanya dibatasi untuk root dan pengguna yang menjalankan Broker. Secara default, ini adalah `dcvsmbroker` pengguna. Broker memeriksa saat startup bahwa file tersebut memiliki izin yang diharapkan.

Tabel berikut mencantumkan parameter dalam file konfigurasi Broker.

Nama Parameter	Diperlukan	Nilai default	Deskripsi
<code>broker-java-home</code>	Tidak		Menentukan path ke direktori home Java Broker akan menggunakan bukan sistem default satu. Jika diatur, Broker akan digunakan <code>&lt;broker-j</code>

Nama Paramet	Diperlukan	Nilai default	Deskripsi
			<p>ava-home&gt;/bin/java saat startup.</p> <p>Tip: Broker membutuhkan Java Runtime Environment 11 dan diinstal jika hilang sebagai ketergantungan pada instalasi sukses. Jika versi 11 tidak ditetapkan sebagai lingkungan Java default, direktori home dapat diambil dengan menggunakan perintah berikut:</p> <pre data-bbox="1117 934 1507 1054">\$ sudo alternatives --display java</pre>
session-screenshots-max-width	Tidak	160	Menentukan lebar maksimum, dalam piksel, screenshot sesi diambil menggunakan GetSessionScreenshotsAPI.
session-screenshots-max-height	Tidak	100	Menentukan tinggi maksimum, dalam piksel, tangkapan layar sesi yang diambil menggunakan GetSessionScreenshotsAPI.



Nama Parameter	Diperlukan	Nilai default	Deskripsi
session-screenshots-format	Tidak	png	Format file gambar dari tangkapan layar sesi yang diambil menggunakan GetSessionScreenshotsAPI.
create-sessions-queue-max-size	Tidak	1000	Jumlah maksimum permintaan CreateSessionsAPI yang tidak terpenuhi yang dapat diantrekan. Saat antrian penuh, permintaan baru yang belum terpenuhi akan ditolak.
create-sessions-queue-max-time-seconds	Tidak	1800	Waktu maksimum, dalam detik, dimana permintaan CreateSessionsAPI yang tidak terpenuhi dapat tetap dalam antrean. Jika permintaan tidak dapat dipenuhi dalam jumlah waktu yang ditentukan, itu gagal.
session-manager-working-path	Ya	/tmp	Menentukan path ke direktori di mana Broker menulis file yang dibutuhkan untuk beroperasi. Direktori ini harus dapat diakses hanya oleh Broker.

Nama Parameter	Diperlukan	Nilai default	Deskripsi
enable-authorization-server	Ya	true	Menentukan apakah broker adalah server otentikasi yang digunakan untuk menghasilkan token akses OAuth 2.0 untuk API klien.
enable-authorization	Ya	true	Mengaktifkan atau menonaktifkan otorisasi klien. Jika Anda mengaktifkan otorisasi klien, API klien harus menyediakan token akses saat membuat permintaan API. Jika Anda menonaktifkan otorisasi klien, API klien dapat membuat permintaan tanpa token akses.
enable-agent-authorization	Ya	true	Mengaktifkan atau menonaktifkan otorisasi Agen. Jika Anda mengaktifkan otorisasi Agen, Agen harus memberikan token akses saat berkomunikasi dengan Pialang.
delete-session-duration-hours	Tidak	1	Menentukan jumlah jam setelah sesi dihapus menjadi tidak terlihat dan tidak lagi dikembalikan oleh panggilan DescribeSession API.

Nama Parameter	Diperlukan	Nilai default	Deskripsi
connect-session-to-keep-duration-minutes	Tidak	60	Menentukan jumlah menit yang ConnectSession token tetap valid.
client-to-broker-connect-https-port	Ya	8443	Menentukan port HTTPS di mana Broker mendengarkan koneksi klien.
client-to-broker-connect-bind-host	Tidak	0.0.0.0	Menentukan alamat IP dari host di mana Broker mengikat untuk koneksi klien.

Nama Parameter	Diperlukan	Nilai default	Deskripsi
client-to-broker-connect-key-store-file	Ya		Menentukan toko kunci yang digunakan untuk koneksi klien TLS.
client-to-broker-connect-key-store-pass	Ya		Menentukan lolos penyimpanan kunci.
agent-to-broker-connection-https-port	Ya	8445	Menentukan port HTTPS di mana Broker mendengarkan koneksi Agen.

Nama Parameter	Diperlukan	Nilai default	Deskripsi
agent-to-broker-connectonbind-host	Tidak	0.0.0.0	Menentukan alamat IP dari host di mana Broker mengikat untuk koneksi Agen.
agent-to-broker-connectonkey-store-file	Ya		Menentukan toko kunci yang digunakan untuk koneksi Agen TLS.
agent-to-broker-connectonkey-store-pass	Ya		Menentukan lolos penyimpanan kunci.
broker-to-broker-port	Ya	47100	Menentukan port yang digunakan untuk koneksi Broker-to-broker.

Nama Parameter	Diperlukan	Nilai default	Deskripsi
broker-to-broker-bind-host	Tidak	0.0.0.0	Menentukan alamat IP dari host di mana Broker mengikat untuk koneksi Broker-to-Broker.
broker-to-broker-discover-port	Ya	47500	Menentukan port yang digunakan oleh Broker untuk menemukan satu sama lain.

Nama Parameter	Diperlukan	Nilai default	Deskripsi
broker-to-broker-discovery-address	Tidak		<p>Menentukan alamat IP dan port dari broker lain dalam armada di <i>ip_addresses</i> : format <i>port</i>. Jika ada beberapa Broker, pisahkan nilai dengan koma.</p> <p>Jika Anda menentukan <code>broker-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> , menghilangkan parameter ini.</p>

Nama Parameter	Diperlukan	Nilai default	Deskripsi
broker-to-broker-discovery-multicast-group	Tidak		Menentukan kelompok multicast untuk broker-to-broker penemuan B. Jika Anda menentukan <code>broker-to-broker-discovery-addresses</code> , <code>broker-to-broker-discovery-aws-region</code> , atau <code>broker-to-broker-discovery-AWS-alb-target-group-arn</code> , menghilangkan parameter ini.
broker-to-broker-discovery-multicast-port	Tidak		Menentukan port multicast untuk broker-to-broker penemuan B. Jika Anda menentukan <code>broker-to-broker-discovery-addresses</code> , <code>broker-to-broker-discovery-AWS-region</code> , atau <code>broker-to-broker-discovery-AWS-alb-target-group-arn</code> , menghilangkan parameter ini.



Nama Parameter	Diperlukan	Nilai default	Deskripsi
broker-to-broker-discovery-AWS-region	Tidak		Menentukan AWS Wilayah load balancer aplikasi yang digunakan untuk Broker untuk penemuan Broker. Jika Anda menentukan <code>broker-to-broker-discovery-multicast-group</code> , <code>broker-to-broker-discovery-multicast-port</code> , atau <code>broker-to-broker-discovery-addresses</code> , menghilangkan parameter ini.
broker-to-broker-discovery-AWS-alb-target-group-arn	Tidak		ARN pengguna grup target penyeimbang beban aplikasi untuk penemuan Broker-to-Broker. Jika Anda menentukan <code>broker-to-broker-discovery-multicast-group</code> , <code>broker-to-broker-discovery-multicast-port</code> , atau <code>broker-to-broker-discovery-addresses</code> , menghilangkan parameter ini.

Nama Paramet	Diperlukan	Nilai default	Deskripsi
broker-to-broker-distribut-d-memory-max-size-mb	Tidak	4096	Menentukan jumlah maksimum memori off-heap yang akan digunakan oleh Broker tunggal untuk menyimpan data sesi NICE DCV.
broker-to-broker-key-store-file	Ya		Menentukan toko kunci yang digunakan untuk koneksi TLS Broker.
broker-to-broker-key-store-pass	Ya		Menentukan lolos penyimpanan kunci.
enable-cloud-watch-metrics	Tidak	false	Mengaktifkan atau menonaktifkan CloudWatch metrik Amazon. Jika Anda mengaktifkan CloudWatch Metrik, Anda mungkin perlu menentukan nilainya <code>cloud-watch-region</code> .

Nama Parameter	Diperlukan	Nilai default	Deskripsi
cloud-watch-ch-region	Tidak	Hanya diperlukan jika <code>enable-cloud-watch-metrics</code> diatur ke <code>true</code> . Jika Broker diinstal pada instans Amazon EC2, wilayah tersebut diambil dari IMDS.	AWS Wilayah tempat CloudWatch metrik diposting.
max-api-requests-per-detik	Tidak	1000	Menentukan jumlah permintaan yang dapat diproses oleh API Broker setiap detik sebelum dicekik.
enable-throttling-forwarder	Tidak	false	Jika diatur ke <code>true</code> ke throttling mengambil ip pemanggil dari header X-Forwarded-For jika ada.
create-session-number-of-retries-on-failure	Tidak	2	Menentukan jumlah maksimum percobaan ulang yang akan dilakukan setelah permintaan sesi membuat gagal pada host server NICE DCV. Atur ke 0 untuk tidak pernah melakukan percobaan ulang pada kegagalan.

Nama Paramet	Diperlukan	Nilai default	Deskripsi
autorun-file-arguments-max-size	Tidak	50	Menentukan jumlah argumen maksimum yang dapat diteruskan ke file autorun.
autorun-file-arguments-max-arguement-length	Tidak	150	Menentukan panjang maksimum dalam karakter dari setiap argumen file autorun.
enable-persistence	Ya	false	Jika diatur ke true, data status broker tetap ada di database eksternal.
persiste-db	Tidak	Hanya diperlukan jika enable-persistence diatur ke true.	Menentukan database yang digunakan untuk ketekunan. Satu-satunya nilai yang didukung adalah: dynamodb dan mysql.

Nama Parameter	Diperlukan	Nilai default	Deskripsi
dynamodb-region	Tidak	Hanya diperlukan jika enable-persistence diatur ke true dan persistence-db diatur ke dynamodb.	Menentukan wilayah di mana tabel DynamoDB dibuat dan diakses.
dynamodb-table-rcu	Tidak	Hanya diperlukan jika enable-persistence diatur ke true dan persistence-db diatur ke dynamodb.	Menentukan Read Capacity Units (RCU) untuk setiap tabel DynamoDB. Untuk informasi selengkapnya tentang RCU, lihat <a href="#">Harga untuk Kapasitas yang Disediakan</a> .
dynamodb-table-wcu	Tidak	Hanya diperlukan jika enable-persistence diatur ke true dan persistence-db diatur ke dynamodb.	Menentukan Write Capacity Units (WCU) untuk setiap tabel DynamoDB. Untuk informasi lebih lanjut tentang WCU, lihat <a href="#">Harga untuk Kapasitas yang Disediakan</a> .

Nama Parameter	Diperlukan	Nilai default	Deskripsi
dynamodb-table-name-prefix	Tidak	Hanya diperlukan jika <code>enable-persistence</code> diatur ke <code>true</code> dan <code>persistency-db</code> diatur ke <code>dynamodb</code> .	Menentukan awalan yang ditambahkan ke setiap tabel DynamoDB (berguna untuk membedakan beberapa kluster broker menggunakan AWS akun yang sama). Hanya karakter alfanumerik, titik, dasbor dan garis bawah yang diizinkan.
jdbc-connection-url	Tidak	Hanya diperlukan jika <code>enable-persistence</code> diatur ke <code>true</code> dan <code>persistency-db</code> diatur ke <code>mysql</code> .	<p>Menentukan URL koneksi ke database MariaDB/MySQL; berisi endpoint dan nama database. Url harus memiliki format ini:</p> <pre data-bbox="1117 1045 1507 1285">jdbc:mysql://&lt;db_endpoint&gt;:&lt;db_port&gt;/&lt;db_name&gt;?createDatabaseIfNotExist=true</pre> <p>Dimana <code>&lt;db_endpoint&gt;</code> adalah endpoint database MariaDB/MySQL, <code>&lt;db_port&gt;</code> adalah port database dan <code>&lt;db_name&gt;</code> nama database.</p>

Nama Parameter	Diperlukan	Nilai default	Deskripsi
jdbc-user	Tidak	Hanya diperlukan jika enable-persistence diatur ke true dan persistence-db diatur ke mysql.	Menentukan nama pengguna yang memiliki akses ke database MariaDB/MySQL.
jdbc-password	Tidak	Hanya diperlukan jika enable-persistence diatur ke true dan persistence-db diatur ke mysql.	Menentukan password dari pengguna yang memiliki akses ke database MariaDB/MySQL.
seconds-before-deleting-unreachable-dcv-server	Tidak	1800	Menentukan jumlah detik setelah server yang tidak terjangkau dihapus dari sistem.

## File konfigurasi agen

File konfigurasi Agen (/etc/dcv-session-manager-agent/agent.conf untuk Linux dan C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf Windows) menyertakan parameter yang dapat dikonfigurasi untuk menyesuaikan fungsionalitas Session Manager. Anda dapat mengedit file konfigurasi menggunakan editor teks pilihan Anda.

Tabel berikut mencantumkan parameter dalam file konfigurasi Agen.

Nama Parameter	Diperlukan	Nilai default	Deskripsi
<code>agent.ker_hostname</code>	Ya		Menentukan nama DNS dari host Broker.
<code>agent.ker_port</code>	Ya	8445	Menentukan port di mana untuk berkomunikasi dengan Broker.
<code>agent.config_file</code>	Tidak		Hanya diperlukan jika <code>tls_strict</code> diatur ke <code>true</code> . Menentukan jalur ke berkas sertifikat (.pem) yang diperlukan untuk memvalidasi sertifikat TLS. Salin sertifikat yang ditandatangani sendiri dari Broker ke Agen.
<code>agent.config_folder</code>	Tidak	<ul style="list-style-type: none"> <li><code>/var/lib/dcv-session-manager-agent/init</code> (Linux)</li> </ul>	Menentukan path ke folder pada server host yang digunakan untuk menyimpan skrip kustom diizinkan untuk menginisialisasi sesi server NICE DCV ketika mereka dibuat. Anda harus menentukan jalur absolut. Folder harus dapat diakses dan file harus dapat dieksekusi oleh pengguna yang menggunakan parameter <code>InitFilepermintaan CreateSessionsAPI</code> .



Nama Parameter	Diperlukan	Nilai default	Deskripsi
agent.tls_strict	Tidak	true	Menunjukkan apakah validasi TLS yang ketat harus digunakan.
agent.software_statement_path	Tidak		Hanya diperlukan jika pernyataan perangkat lunak default tidak digunakan. Menentukan jalur ke file pernyataan software. Untuk informasi selengkapnya, lihat <a href="#">generate-software-statement</a> .
agent.tls_session_folder	Tidak	<ul style="list-style-type: none"> <li>/etc/dcv-session-manager-agent (Linux)</li> <li>C:\Program Files\NICE\DCVSessionManagerAgent\conf\tags (Jendela)</li> </ul>	Menentukan jalur ke folder tempat file tag berada. Untuk informasi selengkapnya, lihat <a href="#">Menggunakan tag untuk menargetkan server NICE DCV</a> .

Nama Parameter	Diperlukan	Nilai default	Deskripsi
agent.a orun_folder	Tidak	<ul style="list-style-type: none"> <li>• /var/lib/dcv-session-manager/autorun (Linux)</li> <li>• C:\ProgramData\NICE\DcvSessionManagerAgent\autorun (Jendela)</li> </ul>	Menentukan jalur ke folder pada server host yang digunakan untuk menyimpan skrip dan aplikasi yang diizinkan untuk dijalankan secara otomatis pada sesi startup. Anda harus menentukan jalur absolut. Folder harus dapat diakses dan file harus dapat dieksekusi oleh pengguna yang menggunakan parameter AutorunFilepermintaan CreateSessionsAPI.
agent.n _virtual sessions	Tidak	-1 (tidak ada batas)	Jumlah maksimum sesi virtual yang dapat dibuat pada server NICE DCV menggunakan NICE DCV Session Manager.
agent.n _concurrent sessions_per user	Tidak	1	Jumlah maksimum sesi virtual yang dapat dibuat pada server NICE DCV oleh satu pengguna menggunakan NICE DCV Session Manager.

Nama Parameter	Diperlukan	Nilai default	Deskripsi
agent.ker_upc e_inte l	Tidak	30	Menentukan jumlah detik untuk menunggu sebelum mengirim data yang diperbarui ke Broker. Data yang dikirim termasuk server NICE DCV dan status host, serta informasi sesi yang diperbarui. Nilai yang lebih rendah membuat Manajer Sesi lebih sadar akan perubahan yang terjadi pada sistem tempat Agen berjalan, tetapi meningkatkan beban sistem dan lalu lintas jaringan. Nilai yang lebih tinggi mengurangi beban sistem dan jaringan, tetapi Manajer Sesi menjadi kurang responsif terhadap perubahan sistem, sehingga nilai lebih tinggi dari 120 yang tidak disarankan.

Nama Parameter	Diperlukan	Nilai default	Deskripsi
log.level	Tidak	info	<p>Menentukan tingkat verbositas file log. Level verbositas berikut tersedia:</p> <ul style="list-style-type: none"> <li>• <b>error</b>—Menyediakan detail paling sedikit. Termasuk kesalahan saja.</li> <li>• <b>warning</b>—Termasuk kesalahan dan peringatan.</li> <li>• <b>info</b>—Tingkat verbositas default. Termasuk kesalahan, peringatan, dan pesan informasi.</li> <li>• <b>debug</b>—Menyediakan paling detail. Memberikan informasi rinci yang berguna untuk masalah debugging.</li> </ul>
log.directory	Tidak	<ul style="list-style-type: none"> <li>• /var/log/dcv-session-manager-agent/(Linux)</li> <li>• C:\ProgramData\NICE\DCVSessionManagerAgent\log (Jendela)</li> </ul>	Menentukan direktori untuk membuat file log.

Nama Parameter	Diperlukan	Nilai default	Deskripsi
log.rotation	Tidak	daily	Menentukan rotasi file log. Nilai yang valid adalah: <ul style="list-style-type: none"> <li>hourlyFile —Log dirotasi setiap jam.</li> <li>dailyFile —Log dirotasi setiap hari.</li> </ul>
log.maxfile-size	Tidak	10485760	Ketika ukuran file log mencapai ukuran yang ditentukan dalam byte, itu akan diputar. File log baru akan dibuat dan peristiwa log lebih lanjut akan ditempatkan di file baru.
log.rotate	Tidak	9	Jumlah maksimum file log yang diawetkan dalam rotasi. Setiap kali rotasi terjadi dan jumlah ini tercapai, file log tertua akan dihapus.

# Catatan rilis dan riwayat dokumen untuk Manajer Sesi NICE DCV

Halaman ini menyediakan catatan rilis dan riwayat dokumen untuk Manajer Sesi NICE DCV.

Topik

- [Catatan rilis Manajer Sesi DCV NICE](#)
- [Riwayat dokumen](#)

## Catatan rilis Manajer Sesi DCV NICE

Bagian ini memberikan gambaran umum tentang pembaruan utama, rilis fitur, dan perbaikan bug untuk Pengelola Sesi DCV NICE. Semua pembaruan diatur berdasarkan tanggal rilis. Kami sering memperbarui dokumentasi untuk mengatasi umpan balik yang Anda kirimkan kepada kami.

Topik

- [2023.1-16388— 26 Juni 2024](#)
- [2023.1— November 9, 2023](#)
- [2023.0-15065— 4 Mei 2023](#)
- [2023.0-14852— 28 Maret 2023](#)
- [2022.2-13907— November 11, 2022](#)
- [2022.1-13067— 29 Juni 2022](#)
- [2022.0-11952— 23 Februari 2022](#)
- [2021.3-11591— Desember 20, 2021](#)
- [2021.2-11445— November 18, 2021](#)
- [2021.2-11190— 11 Oktober 2021](#)
- [2021.2-11042— September 01, 2021](#)
- [2021.1-10557— 31 Mei 2021](#)
- [2021.0-10242— 12 April 2021](#)
- [2020.2-9662— 04 Desember 2020](#)

- [2020.2-9508— 11 November 2020](#)

## 2023.1-16388— 26 Juni 2024

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 417</li> <li>• Agen: 748</li> <li>• CLI: 140</li> </ul>	<ul style="list-style-type: none"> <li>• Memperbaiki bug yang menunjukkan memori secara tidak benar sebagai TB, bukan GB.</li> <li>• Perbaiki bug dan peningkatan performa.</li> </ul>

## 2023.1— November 9, 2023

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 410</li> <li>• Agen: 732</li> <li>• CLI: 140</li> </ul>	<ul style="list-style-type: none"> <li>• Perbaiki bug dan peningkatan kinerja</li> </ul>

## 2023.0-15065— 4 Mei 2023

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 392</li> <li>• Agen: 675</li> <li>• CLI: 132</li> </ul>	<ul style="list-style-type: none"> <li>• Menambahkan dukungan untuk Red Hat Enterprise Linux 9, Rocky Linux 9, dan CentOS Stream 9 pada platform ARM.</li> </ul>

## 2023.0-14852— 28 Maret 2023

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 392</li> </ul>	<ul style="list-style-type: none"> <li>• Menambahkan dukungan untuk Red Hat Enterprise Linux 9, Rocky Linux 9, dan CentOS Stream 9.</li> </ul>

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Agen: 642</li> <li>• CLI: 132</li> </ul>	

## 2022.2-13907— November 11, 2022

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 382</li> <li>• Agen: 612</li> <li>• CLI: 123</li> </ul>	<ul style="list-style-type: none"> <li>• Menambahkan Substate bidang DescribeSessions sebagai tanggapan.</li> <li>• Memperbaiki masalah yang dapat menyebabkan CLI gagal terhubung ke broker tergantung pada URL yang digunakan.</li> </ul>

## 2022.1-13067— 29 Juni 2022

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 355</li> <li>• Agen: 592</li> <li>• CLI: 114</li> </ul>	<ul style="list-style-type: none"> <li>• Menambahkan dukungan untuk menjalankan broker pada instance AWS Graviton.</li> <li>• Menambahkan dukungan agen dan broker untuk Ubuntu 22.04.</li> </ul>

## 2022.0-11952— 23 Februari 2022

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 341</li> <li>• Agen: 520</li> <li>• CLI: 112</li> </ul>	<ul style="list-style-type: none"> <li>• Menambahkan kemampuan rotasi log ke Agen.</li> <li>• Ditambahkan parameter konfigurasi untuk mengatur rumah Java di Broker.</li> <li>• Peningkatan pembilasan data dari cache ke disk di Broker.</li> <li>• Validasi URL tetap di CLI.</li> </ul>



## 2021.3-11591— Desember 20, 2021

Membangun angka	Fitur baru
<ul style="list-style-type: none"> <li>• Pialang: 307</li> <li>• Agen: 453</li> <li>• CLI: 92</li> </ul>	<ul style="list-style-type: none"> <li>• Menambahkan dukungan untuk mengintegrasikan dengan NICE DCV Connection Gateway.</li> <li>• Menambahkan dukungan Broker untuk Ubuntu 18.04 dan Ubuntu 20.04.</li> </ul>

## 2021.2-11445— November 18, 2021

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 288</li> <li>• Agen: 413</li> <li>• CLI: 54</li> </ul>	<ul style="list-style-type: none"> <li>• Memperbaiki masalah dengan validasi nama login yang mencakup domain Windows.</li> </ul>

## 2021.2-11190— 11 Oktober 2021

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 254</li> <li>• Agen: 413</li> <li>• CLI: 54</li> </ul>	<ul style="list-style-type: none"> <li>• Memperbaiki masalah di antarmuka baris perintah yang mencegah peluncuran sesi Windows.</li> </ul>

## 2021.2-11042— September 01, 2021

Membangun angka	Fitur baru	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 254</li> </ul>	<ul style="list-style-type: none"> <li>• NICE DCV Session Manager sekarang menawarkan dukungan antarmuka baris perintah (CLI). Anda</li> </ul>	<ul style="list-style-type: none"> <li>• Saat mendaftarkan server otorisasi eksternal, Anda sekarang dapat menentukan algoritme yang</li> </ul>

Membangun angka	Fitur baru	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Agen: 413</li> <li>• CLI: 37</li> </ul>	<ul style="list-style-type: none"> <li>• dapat membuat dan mengelola sesi NICE DCV di CLI, alih-alih memanggil API.</li> <li>• Manajer Sesi NICE DCV memperkenalkan ketekunan data Broker. Untuk ketersediaan yang lebih tinggi, broker dapat mempertahankan informasi status server pada penyimpanan data eksternal dan memulihkan data saat startup.</li> </ul>	<p>digunakan server otorisasi untuk menandatangani Token Web berformat JSON. Dengan perubahan ini, Anda dapat menggunakan Azure AD sebagai server otorisasi eksternal.</p>

## 2021.1-10557— 31 Mei 2021

Membangun angka	Fitur baru	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 214</li> <li>• Agen: 365</li> </ul>	<ul style="list-style-type: none"> <li>• NICE DCV Session Manager menambahkan dukungan untuk parameter input yang diteruskan ke file autorun di Linux.</li> <li>• Properti server sekarang dapat diteruskan sebagai persyaratan ke <a href="#">CreateSessionsAPI</a>.</li> </ul>	<ul style="list-style-type: none"> <li>• Kami memperbaiki masalah dengan file autorun di Windows.</li> </ul>

## 2021.0-10242— 12 April 2021

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 183</li> <li>• Agen: 318</li> </ul>	<ul style="list-style-type: none"> <li>• Manajer Sesi NICE DCV memperkenalkan API baru berikut: <ul style="list-style-type: none"> <li>• <a href="#">OpenServers</a></li> <li>• <a href="#">CloseServers</a></li> </ul> </li> </ul>

Membangun angka	Perubahan dan perbaikan bug
	<ul style="list-style-type: none"> <li>• <a href="#">DescribeServers</a></li> <li>• <a href="#">GetSessionScreenshots</a></li> <li>• Ini juga memperkenalkan parameter konfigurasi baru berikut: <ul style="list-style-type: none"> <li>• <a href="#">Parameter broker</a>:<code>session-screenshot-max-width</code> ,<code>session-screenshot-max-height</code> ,<code>session-screenshot-format</code> ,<code>create-sessions-queue-max-size</code> ,dan<code>create-sessions-queue-max-time-seconds</code> .</li> <li>• <a href="#">Parameter agen</a>:<code>agent.autorun_folder</code> ,<code>max_virtual_sessions</code> ,dan<code>max_concurrent_sessions_per_user</code> .</li> </ul> </li> </ul> <p><a href="#">Parameter agen</a>:<code>agent.autorun_folder</code> ,<code>max_virtual_sessions</code> ,dan<code>max_concurrent_sessions_per_user</code> .</p> <p><a href="#">Parameter agen</a>:<code>agent.autorun_folder</code> ,<code>max_virtual_sessions</code> ,dan<code>max_concurrent_sessions_per_user</code> .</p>

## 2020.2-9662— 04 Desember 2020

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 114</li> <li>• Agen: 211</li> </ul>	<ul style="list-style-type: none"> <li>• Kami memperbaiki masalah dengan sertifikat TLS yang dibuat secara otomatis yang mencegah Broker memulai.</li> </ul>

## 2020.2-9508— 11 November 2020

Membangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> <li>• Pialang: 78</li> <li>• Agen: 183</li> </ul>	<ul style="list-style-type: none"> <li>• Rilis awal Pengelola Sesi NICE DCV.</li> </ul>

## Riwayat dokumen

Tabel berikut menjelaskan dokumentasi untuk rilis NICE DCV Session Manager ini.

Perubahan	Deskripsi	Tanggal
NICE DCV Versi 2023.1-16388	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2023.1-16388. Untuk informasi selengkapnya, lihat <a href="#">2023.1-16388— 26 Juni 2024</a> .	Juni 26, 2024
NICE DCV Versi 2023.1	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2023.1. Untuk informasi selengkapnya, lihat <a href="#">2023.1— November 9, 2023</a> .	9 November 2023
NICE DCV Versi 2023.0	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2023.0. Untuk informasi selengkapnya, lihat <a href="#">2023.0-14852— 28 Maret 2023</a> .	Maret 28, 2023
NICE DCV Versi 2022.2	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2022.2. Untuk informasi selengkapnya, lihat <a href="#">2022.2-13907— November 11, 2022</a> .	11 November 2022
NICE DCV Versi 2022.1	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2022.1. Untuk informasi selengkapnya, lihat <a href="#">2022.1-13067— 29 Juni 2022</a> .	Juni 29, 2022
NICE DCV Versi 2022.0	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2022.0. Untuk informasi selengkapnya, lihat <a href="#">2022.0-11952— 23 Februari 2022</a> .	Februari 23, 2022

Perubahan	Deskripsi	Tanggal
NICE DCV Versi 2021.3	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2021.3. Untuk informasi selengkapnya, lihat <a href="#">2021.3-11591— Desember 20, 2021</a> .	Desember 20, 2021
NICE DCV Versi 2021.2	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2021.2. Untuk informasi selengkapnya, lihat <a href="#">2021.2-11042— September 01, 2021</a> .	September 01, 2021
NICE DCV Versi 2021.1	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2021.1. Untuk informasi selengkapnya, lihat <a href="#">2021.1-10557— 31 Mei 2021</a> .	Mei 31, 2021
NICE DCV Versi 2021.0	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2021.0. Untuk informasi selengkapnya, lihat <a href="#">2021.0-10242— 12 April 2021</a> .	12 April 2021
Rilis Awal Session Manager NICE DCV	Publikasi pertama dari konten ini.	11 November 2020

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.