



Guia do administrador

Gerenciador de Sessões do NICE DCV



Gerenciador de Sessões do NICE DCV: Guia do administrador

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

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não são propriedade da Amazon pertencem aos respectivos proprietários, os quais podem ou não ser afiliados, estar conectados ou ser patrocinados pela Amazon.

Table of Contents

O que é o Gerenciador de Sessões?	1
Como o Gerenciador de Sessões funciona	1
Atributos	3
Limitações	3
Preços	4
Requisitos	4
Requisitos de rede e conectividade	5
Configuração	7
Etapa 1: Preparar os servidores do NICE DCV	7
Etapa 2: Configurar o Broker	8
Etapa 3: Configurar o Atendente	10
Etapa 4: Configurar o servidor NICE DCV	15
Etapa 5: Verificar as instalações	17
Verificar o Atendente	17
Verificar o Broker	18
Configuração	19
Gerenciador de Sessões de Escalabilidade	19
Etapa 1: Criar um perfil de instância	20
Etapa 2: Preparar o certificado SSL para o balanceador de carga	21
Etapa 3: Criar o Application Load Balancer do Agente	21
Etapa 4: Iniciar os Agentes	23
Etapa 5: Criar o Application Load Balancer do Atendente	24
Etapa 6: Iniciar os atendentes	25
Usar tags	26
Configurar um servidor de autorização externo	28
Como configurar a persistência do agente	33
Configure o agente para persistir no DynamoDB	34
Configure o agente para persistir no MariaDB/MySQL	35
Integração com o gateway de conexão do NICE DCV	36
Configurar o agente do gerenciador de sessões como um resolvedor de sessões para o gateway de conexão do NICE DCV	36
Opcional – Ativar a autenticação do cliente TLS	37
Servidor NICE DCV – mapeamento de DNS	39
Fazer a integração com o Amazon CloudWatch	40

Fazer upgrade	43
Fazer upgrade do Atendente do Gerenciador de Sessões do NICE DCV	43
Fazer upgrade do Agente do Gerenciador de Sessões do NICE DCV	45
Referência da CLI do agente	48
register-auth-server	49
Sintaxe	49
Opções	49
Exemplo	49
list-auth-servers	50
Sintaxe	49
Resultado	50
Exemplo	49
unregister-auth-server	51
Sintaxe	49
Opções	49
Resultado	50
Exemplo	49
register-api-client	52
Sintaxe	49
Opções	49
Resultado	50
Exemplo	49
describe-api-clients	54
Sintaxe	49
Resultado	50
Exemplo	49
unregister-api-client	55
Sintaxe	49
Opções	49
Exemplo	49
renew-auth-server-api-key	56
Sintaxe	49
Exemplo	49
generate-software-statement	57
Sintaxe	49
Resultado	50

Exemplo	49
describe-software-statements	58
Sintaxe	49
Resultado	50
Exemplo	49
deactivate-software-statement	59
Sintaxe	49
Opções	49
Exemplo	49
describe-agent-clients	60
Sintaxe	49
Resultado	50
Exemplo	49
unregister-agent-client	62
Sintaxe	49
Opções	49
Exemplo	49
register-server-dns-mappings	63
Sintaxe	49
Opções	49
Exemplo	49
describe-server-dns-mappings	63
Sintaxe	49
Resultado	50
Exemplo	49
Referência do arquivo de configuração	66
Arquivo de configuração do Broker	66
Arquivo de configuração do Atendente	82
Notas de versão e histórico de documentos	89
Notas de versão	89
2023.1 — 9 de novembro de 2023	90
2023.0-15065 — 4 de maio de 2023	90
2023.0-14852 — 28 de março de 2023	90
2022.2-13907 — 11 de novembro de 2022	91
2022.1-13067 — 29 de junho de 2022	91
2022.0-11952 — 23 de fevereiro de 2022	91

2021.3-11591 — 20 de dezembro de 2021	92
2021.2-11445 — 18 de novembro de 2021	92
2021.2-11190 — 11 de outubro de 2021	92
2021.2-11042 — 1º de setembro de 2021	93
2021.1-10557 — 31 de maio de 2021	93
2021.0-10242 — 12 de abril de 2021	94
2020.2-9662 — 4 de dezembro de 2020	95
.....	95
Histórico do documento	95
.....	xcviii

O que é o Gerenciador de Sessões do NICE DCV?

O Gerenciador de Sessões do NICE DCV é um conjunto de pacotes de software instaláveis (um Atendente e um Broker) e uma interface de programação de aplicações (API) que facilita que desenvolvedores e provedores independentes de software (ISVs) desenvolvam aplicativos front-end que criam e gerenciam programaticamente o ciclo de vida das sessões do NICE DCV em uma frota de servidores do NICE DCV.

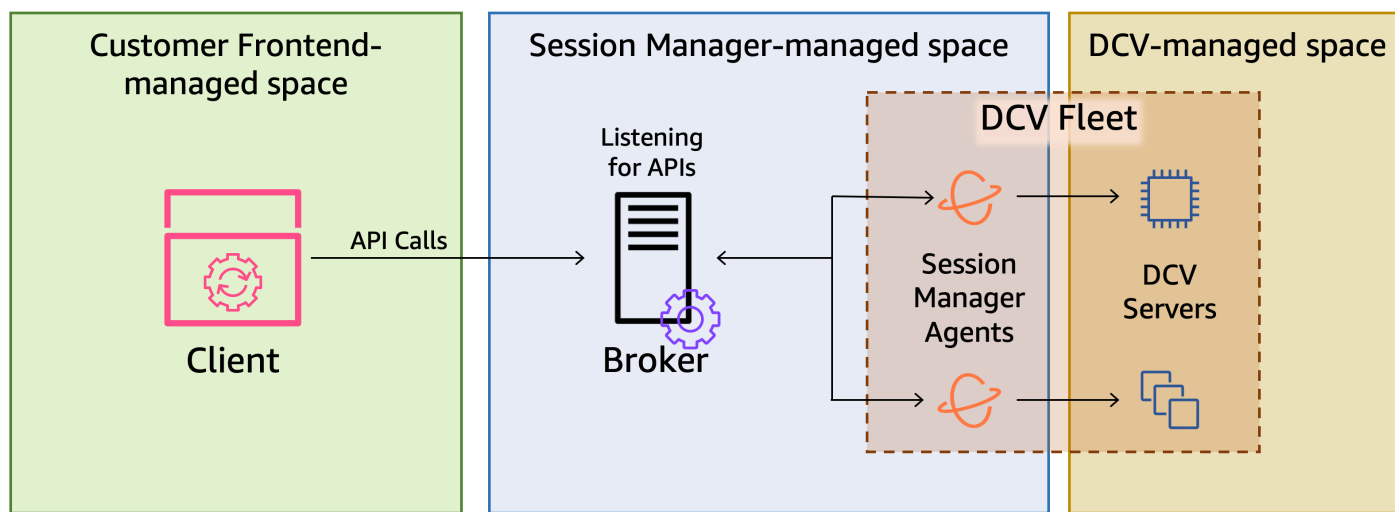
Este guia explica como instalar e configurar o Atendente e o Broker do Gerenciador de Sessões. Para obter mais informações sobre o uso das APIs do Gerenciador de Sessões, consulte o Guia do desenvolvedor do Gerenciador de Sessões do NICE DCV.

Tópicos

- [Como o Gerenciador de Sessões funciona](#)
- [Atributos](#)
- [Limitações](#)
- [Preços](#)
- [Requisitos do Gerenciador de Sessões do NICE DCV](#)

Como o Gerenciador de Sessões funciona

O diagrama a seguir mostra os componentes de alto nível do Gerenciador de Sessões.



Broker

O Broker é um servidor web que hospeda e expõe as APIs do Gerenciador de Sessões. Ele recebe e processa solicitações de API para gerenciar sessões do NICE DCV do cliente e, em seguida, transmite as instruções para os Atendentes relevantes. O Broker deve ser instalado em um host separado dos servidores do NICE DCV, mas deve estar acessível ao cliente e aos Atendentes.

Atendente

O Atendente é instalado em cada servidor do NICE DCV da frota. Os Atendentes recebem instruções do Broker e as executam em seus respectivos servidores do NICE DCV. Os Atendentes também monitoram o estado dos servidores do NICE DCV e enviam atualizações periódicas de status de volta ao Broker.

APIs

O Gerenciador de Sessões expõe um conjunto de interfaces de programação de aplicações (APIs) REST que pode ser usado para gerenciar sessões do NICE DCV em uma frota de servidores do NICE DCV. As APIs são hospedadas e expostas pelo Broker. Os desenvolvedores podem criar clientes personalizados de gerenciamento de sessões que chamam as APIs.

Cliente

O cliente é o aplicativo ou portal front-end que você desenvolve para chamar as APIs do Gerenciador de Sessões que são expostas pelo Broker. Os usuários finais usam o cliente para gerenciar as sessões hospedadas nos servidores do NICE DCV da frota.

Token de acesso

Para fazer uma solicitação de API, você deve fornecer um token de acesso. Os tokens podem ser solicitados ao Broker, ou a um servidor de autorização externo, por meio de APIs de clientes registrados. Para solicitar e acessar o token, a API do cliente deve fornecer credenciais válidas.

API do cliente

A API do cliente é gerada a partir do arquivo YAML de definição da API do Gerenciador de Sessões, usando o Swagger Codegen. A API do cliente é usada para fazer solicitações de API.

Sessão do NICE DCV

Você deve criar uma sessão do NICE DCV no servidor do NICE DCV à qual seus clientes possam se conectar. Os clientes só poderão se conectar a um servidor do NICE DCV se houver

uma sessão ativa. O NICE DCV é compatível com sessões virtuais e console. Use as APIs do Gerenciador de Sessões para gerenciar o ciclo de vida das sessões do NICE DCV. As sessões do NICE DCV podem estar em um dos seguintes estados:

- CREATING — o Broker está criando a sessão.
- READY — a sessão está pronta para aceitar conexões de clientes.
- DELETING — a sessão está sendo excluída.
- DELETED — a sessão foi excluída.
- UNKNOWN — incapaz de determinar o estado da sessão. O Broker e o Atendente podem não conseguir se comunicar.

Atributos

O Gerenciador de Sessões do DCV oferece os seguintes atributo:

- Fornece informações sobre a sessão do NICE DCV — obtenha informações sobre as sessões em execução em vários servidores do NICE DCV.
- Gerencie o ciclo de vida de várias sessões do NICE DCV — crie ou exclua várias sessões para vários usuários em vários servidores do NICE DCV com uma solicitação de API.
- Compatível com tags — use tags personalizadas para segmentar um grupo de servidores do NICE DCV ao criar sessões.
- Gerencia permissões para várias sessões do NICE DCV — modifique as permissões do usuário para várias sessões com uma solicitação de API.
- Fornece informações de conexão — recupera informações de conexão do cliente para sessões do NICE DCV.
- Compatível com nuvem e on-premises — use o Gerenciador de Sessões no AWS, on-premises ou com servidores alternativos baseados na nuvem.

Limitações

O Gerenciador de Sessões não fornece recursos de provisionamento de atributos. Se você estiver executando o NICE DCV em instâncias do Amazon EC2, talvez seja necessário usar serviços adicionais da AWS, como o Amazon EC2 Auto Scaling, para gerenciar a escalabilidade da sua infraestrutura.

Preços

O Gerenciador de Sessões está disponível gratuitamente para clientes da AWS que executam instâncias do EC2.

Clientes locais precisam de uma licença do NICE DCV Plus ou do DCV Professional Plus. Para obter informações sobre como comprar uma licença perpétua ou uma assinatura do NICE DCV ou do NICE DCV Professional Plus, consulte [Como comprar](#) no site do NICE e encontre um distribuidor ou um revendedor do NICE na sua região. Para permitir que todos os clientes on-premises experimentem o Gerenciador de Sessões do DCV, os requisitos de licenciamento só serão aplicados a partir da versão 2021.0 do NICE DCV.

Para obter mais informações, consulte [Licenciar o Servidor do NICE DCV](#) no Guia do administrador do NICE DCV.

Requisitos do Gerenciador de Sessões do NICE DCV

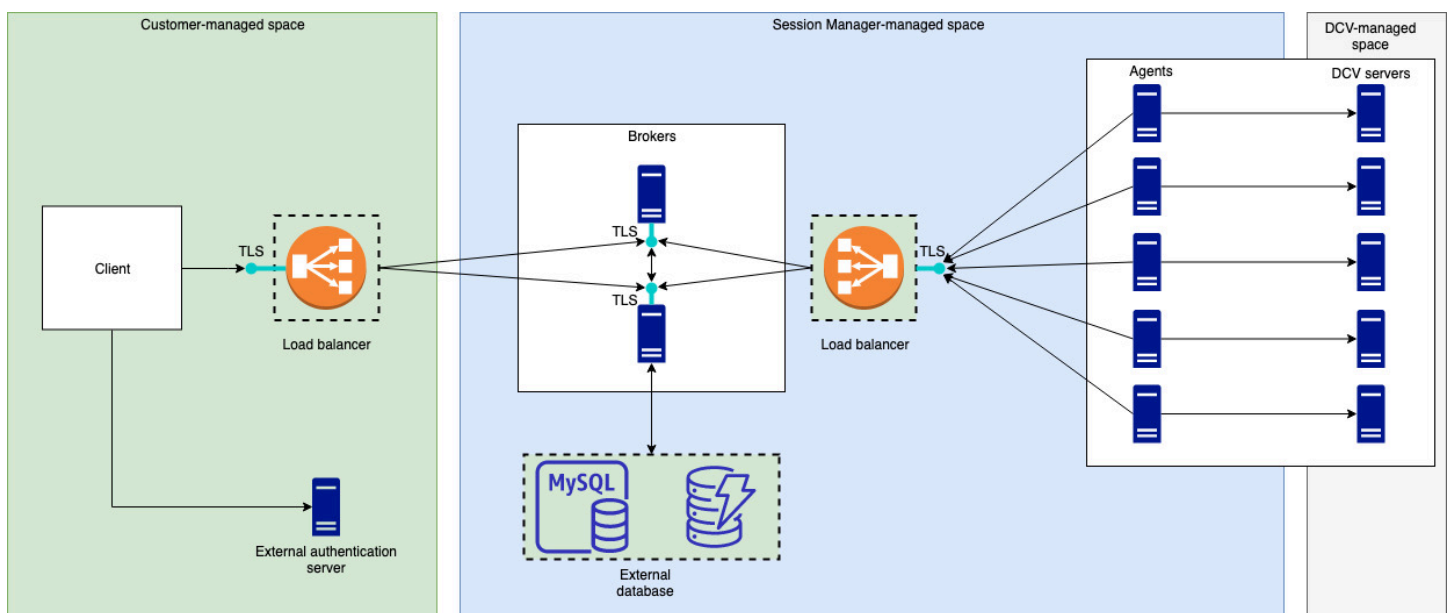
O Atendente e Broker do Gerenciador de Sessões do NICE DCV têm os seguintes requisitos.

	Broker	Atendente
Sistema operacional	<ul style="list-style-type: none"> • Amazon Linux 2 • CentOS 7.6 ou posterior • CentOS Stream 8 • RHEL 7.6 ou posterior • RHEL 8.x • Rocky Linux 8.5 ou posterior • Ubuntu 20.04 • Ubuntu 22.04 	<ul style="list-style-type: none"> • Windows <ul style="list-style-type: none"> • Windows Server 2019 • Windows Server 2016 • Windows Server 2012 R2 • Servidor do Linux <ul style="list-style-type: none"> • Amazon Linux 2 • CentOS 7.6 ou posterior • CentOS Stream 8 • RHEL 7.6 ou posterior • RHEL 8.x • Rocky Linux 8.5 ou posterior • Ubuntu 20.04 • Ubuntu 22.04

	Broker	Atendente
		<ul style="list-style-type: none"> • SUSE Linux Enterprise 12 com SP4 ou posterior • SUSE Linux Enterprise 15
Arquitetura	<ul style="list-style-type: none"> • x86 de 64 bits • ARM de 64 bits 	<ul style="list-style-type: none"> • x86 de 64 bits • ARM de 64 bits (somente Amazon Linux 2, CentOS 7.x/8.x e RHEL 7.x/8.x) • ARM de 64 bits (Ubuntu 22.04)
Memória	8 GB	4 GB
Versão do NICE DCV	NICE DCV 2020.2 e posterior	NICE DCV 2020.2 e posterior
Requisitos adicionais	Java 11	-

Requisitos de rede e conectividade

O diagrama a seguir fornece uma visão geral de alto nível dos requisitos de rede e conectividade do Gerenciador de Sessões.



O Broker deve ser instalado em um host separado, mas deve ter conectividade de rede com os Atendentes nos servidores do NICE DCV. Se você optar por ter vários Brokers para melhorar a disponibilidade, deverá instalar e configurar cada broker em um host separado e usar um ou mais balanceadores de carga para gerenciar o tráfego entre o cliente e os Brokers, entre os Brokers e os Atendentes. Os Brokers também devem ser capazes de se comunicar entre si para trocar informações sobre os servidores e sessões do NICE DCV. Os Brokers podem armazenar suas chaves e dados de status em um banco de dados externo e ter essas informações disponíveis após a reinicialização ou encerramento. Isso ajuda a reduzir o risco de perda de informações importantes do Broker ao persisti-las no banco de dados externo. Não será possível recuperá-las posteriormente. Se você optar por fazê-lo, deverá configurar o banco de dados externo e configurar os brokers. DynamoDB, MariaDB e MySQL são compatíveis. Você pode ver que os parâmetros de configuração estão listados no [Arquivo de Configuração do Broker](#).

Os Atendentes devem ser capazes de iniciar conexões HTTPs seguras, persistentes e bidirecionais com o Broker.

Seu cliente, ou aplicativo de front-end, deve ser capaz de acessar o Broker para chamar as APIs. O cliente também deve ser capaz de acessar o servidor de autenticação.

Configurar o Gerenciador de Sessões do NICE DCV

A seção a seguir explica como instalar o Gerenciador de Sessões com um único Broker e vários Atendentes. Você pode usar vários Brokers para melhorar a escalabilidade e o desempenho. Para ter mais informações, consulte [Gerenciador de Sessões de Escalabilidade](#).

Para configurar o Gerenciador de Sessões do NICE DCV, faça o seguinte:

Etapas

- [Etapa 1: Preparar os servidores do NICE DCV](#)
- [Etapa 2: Configurar o Broker do Gerenciador de Sessões do NICE DCV](#)
- [Etapa 3: Configurar o Atendente do Gerenciador de Sessões do NICE DCV](#)
- [Etapa 4: Configure o servidor do NICE DCV para usar o Broker como servidor de autenticação](#)
- [Etapa 5: Verificar as instalações](#)

Etapa 1: Preparar os servidores do NICE DCV

Você deve ter uma frota de servidores do NICE DCV com os quais pretende usar o Gerenciador de Sessões. Para obter mais informações sobre a instalação dos servidores do NICE DCV, consulte [Instalar o servidor do NICE DCV](#) no Guia do administrador do NICE DCV.

Nos servidores do NICE DCV do Linux, o Gerenciador de Sessões usa um usuário de serviço local chamado `dcvsmagent`. Esse usuário é criado automaticamente quando o Atendente do Gerenciador de Sessões é instalado. Você deve conceder a esse usuário do serviço privilégios de administrador para NICE DCV para que ele possa realizar ações em nome de outros usuários. Para conceder privilégios de administrador ao usuário do serviço Gerenciador de Sessões, faça o seguinte:

Para adicionar o usuário do serviço local para servidores do NICE DCV do Linux

1. Abra o `/etc/dcv/dcv.conf` usando o editor de texto de sua preferência.
2. Adicione o parâmetro `administrators` à seção `[security]` e especifique o usuário do Gerenciador de Sessões. Por exemplo: .

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

3. Salve e feche o arquivo.

4. Interrompa e reinicie o servidor NICE DCV.

O Gerenciador de Sessões só pode criar sessões do NICE DCV em nome de usuários que já existem no servidor do NICE DCV. Se for feita uma solicitação para criar uma sessão para um usuário que não existe, a solicitação falhará. Portanto, você deve garantir que cada usuário final desejado tenha um usuário de sistema válido no servidor do NICE DCV.

Tip

Se você pretende usar vários hosts do Broker ou servidores do NICE DCV com Atendentes, recomendamos que você configure somente um Broker e um servidor do NICE DCV com um Atendente executando as etapas a seguir, criando Imagens da Máquina da Amazon (AMI) dos hosts com as configurações concluídas e, em seguida, usando as AMIs para iniciar os demais Brokers e servidores do NICE DCV. Como alternativa, você pode usar o AWS Systems Manager para executar os comandos em várias instâncias remotamente.

Etapa 2: Configurar o Broker do Gerenciador de Sessões do NICE DCV

O Broker deve estar instalado em um host Linux. Para obter mais informações sobre as distribuições do Linux compatíveis, consulte [Requisitos do Gerenciador de Sessões do NICE DCV](#). Instale o Broker em um host separado do Atendente e do host do servidor do NICE DCV. O host pode ser instalado em uma rede privada diferente, mas deve ser capaz de se conectar e se comunicar com o Atendente.

Para instalar e iniciar o Broker

1. Conecte-se ao host no qual você pretende instalar o Broker.
2. Os pacotes do são assinados digitalmente com uma assinatura GPG segura. Para permitir que o gerenciador de pacotes verifique a assinatura do pacote, é necessário importar a chave GPG do NICE. Execute o comando a seguir para importar a chave NICE GPG.
 - Amazon Linux 2, RHEL, CentOS e Rocky Linux

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

- Ubuntu

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

3. Baixe o pacote de instalação.

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

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

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

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

- Ubuntu 20.04

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

- Ubuntu 22.04

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

4. Instale o pacote .

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

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

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

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

- Ubuntu 20.04

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

- Ubuntu 22.04

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

5. Verifique se a versão padrão do ambiente Java é 11

```
$ java -version
```

Caso contrário, você poderá definir explicitamente o diretório inicial do Java que o Broker usará para direcionar a versão correta do Java. Isso é feito definindo o parâmetro `broker-java-home` no arquivo de configuração do Broker. Para obter mais informações, consulte [Arquivo de configuração do Broker](#).

6. Inicie o serviço do Broker e certifique-se de que ele seja iniciado automaticamente sempre que a instância for executada.

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

7. Coloque uma cópia do certificado assinado pelo Broker no seu diretório de usuários. Você precisará dele ao instalar os Atendentes na próxima etapa.

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

Etapa 3: Configurar o Atendente do Gerenciador de Sessões do NICE DCV

O Atendente deve estar instalado em todos os hosts do servidor do NICE DCV da frota. O Atendente pode ser instalado em servidores Windows e Linux. Para obter mais informações sobre os sistemas operacionais compatíveis, consulte [Requisitos do Gerenciador de Sessões do NICE DCV](#).

Pré-requisitos

O servidor do NICE DCV deve ser instalado no host antes de instalar o Atendente.

Linux host

Note

[O Session Manager Agent está disponível para as distribuições e arquiteturas Linux listadas nos Requisitos:](#)

As instruções a seguir são para instalar o Atendente em hosts x86 de 64 bits. Para instalar o Agente em hosts ARM de 64 bits, substitua `x86_64` por `aarch64`. Para o Ubuntu, substitua `amd64` por `arm64`.

Para instalar o Atendente em um host do Linux

1. Os pacotes são assinados digitalmente com uma assinatura GPG segura. Para permitir que o gerenciador de pacotes verifique a assinatura do pacote, é necessário importar a chave GPG do NICE. Execute o comando a seguir para importar a chave NICE GPG.

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

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

- Ubuntu

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

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

2. Baixe o pacote de instalação.

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

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

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

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

- Ubuntu 20.04

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

- Ubuntu 22.04

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

- SUSE Linux Enterprise 12

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

- SUSE Linux Enterprise 15

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

3. Instale o pacote .

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

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

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

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

- Ubuntu 20.04

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

- Ubuntu 22.04

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

- SUSE Linux Enterprise 12

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

- SUSE Linux Enterprise 15

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

4. Coloque uma cópia do certificado autoassinado do Broker (que você copiou na etapa anterior) no diretório `/etc/dcv-session-manager-agent/` do Atendente.
5. Abra `/etc/dcv-session-manager-agent/agent.conf` usando o editor de texto que preferir e adicione o seguinte:
 - Para `broker_host`, especifique o nome DNS do host no qual o Broker está instalado.

 Important

Se o Broker estiver sendo executado em uma instância do Amazon EC2, para `broker_host`, você deverá especificar o endereço IPv4 privado da instância.

- (Opcional) Para `broker_port`, especifica a porta pela qual se comunicar com o Broker. Por padrão, o Atendente e o Broker se comunicam pela porta 8445. Altere isso somente se precisar usar uma porta diferente. Se você alterá-la, certifique-se de que o Broker esteja configurado para usar a mesma porta.
- Para `ca_file`, especifique o caminho completo do arquivo de certificado que você copiou na etapa anterior. Por exemplo: .

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

Opcionalmente, se você quiser desativar a verificação do TLS, defina `tls_strict` como `false`.

6. Salve e feche o arquivo.
7. Para iniciar o Atendente, execute o comando a seguir.

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

Windows host

Para instalar o Atendente em um host do Windows

1. Baixe o [instalador do Atendente](#).
2. Execute o instalador. Na tela Welcome, escolha Next.
3. Na tela do EULA (Contrato de Licença do Usuário Final), leia com atenção o contrato de licença e, se você concordar, selecione Aceito os termos e escolha Avançar.
4. Para iniciar a instalação, escolha Instalar.
5. Coloque uma cópia do certificado autoassinado do Broker (que você copiou na etapa anterior) na pasta C:\Program Files\NICE\DCVSessionManagerAgent\conf\ do Atendente.
6. Abra o C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf usando o editor de texto que preferir e adicione o seguinte:
 - Para broker_host, especifique o nome DNS do host no qual o Broker está instalado.

Important

Se o Broker estiver sendo executado em uma instância do Amazon EC2, para broker_host, você deverá especificar o endereço IPv4 privado da instância.

- (Opcional) Para broker_port, especifica a porta pela qual se comunicar com o Broker. Por padrão, o Atendente e o Broker se comunicam pela porta 8445. Altere isso somente se precisar usar uma porta diferente. Se você alterá-la, certifique-se de que o Broker esteja configurado para usar a mesma porta.
- Para ca_file, especifique o caminho completo do arquivo de certificado que você copiou na etapa anterior. Por exemplo: .

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

Opcionalmente, se você quiser desativar a verificação do TLS, defina tls_strict como false.

7. Salve e feche o arquivo.
8. Para e reinicie o serviço do Atendente para que as alterações entrem em vigor. Execute os comandos a seguir no prompt de comando.

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

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

Etapa 4: Configure o servidor do NICE DCV para usar o Broker como servidor de autenticação

Configure o servidor do NICE DCV para usar o Broker como servidor de autenticação externa para validar os tokens de conexão do cliente. Você também deve configurar o servidor do NICE DCV para confiar na CA autoassinada do Broker.

Linux NICE DCV server

Para adicionar o usuário do serviço local para servidores do NICE DCV do Linux

1. Abra o `/etc/dcv/dcv.conf` usando o editor de texto de sua preferência.
2. Adicione os parâmetros `ca-file` e `auth-token-verifier` à seção `[security]`.

Para `ca-file`, especifique o caminho para a CA autoassinada do Broker que você copiou para o host na etapa anterior.

Para `auth-token-verifier`, especifique a URL do verificador de token no Broker no seguinte formato: `https://broker_ip_or_dns:port/agent/validate-authentication-token`. Especifique a porta usada para a comunicação Broker-Atendente, que é 8445 por padrão. Se você estiver executando o Broker em uma instância do Amazon EC2, deverá usar o DNS privado ou o endereço IP privado.

Por exemplo

```
[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. Salve e feche o arquivo.

4. Interrompa e reinicie o servidor NICE DCV. Para obter mais informações, consulte [Interromper o servidor do NICE DCV](#) e [Iniciar o servidor do NICE DCV](#) no Guia do administrador do NICE DCV.

Windows NICE DCV server

Em servidores do NICE DCV do Windows

1. Abra o Editor de Registro do Windows e navegue até a chave HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/.
2. Abra o parâmetro ca-file. Para dados de Valores, especifique o caminho para a CA autoassinada do Broker que você copiou para o host na etapa anterior.

Note

Se o parâmetro não existir, crie um novo parâmetro de string e nomeie-o como ca-file.

3. Abra o auth-token-verifierparâmetro. Para dados de Valores, especifique o URL do verificador de token no Broker no seguinte formato: `https://broker_ip_or_dns:port/agent/validate-authentication-token`. Especifique a porta usada para a comunicação Broker-Atendente, que é 8445 por padrão. Se você estiver executando o Broker em uma instância do Amazon EC2, deverá usar o DNS privado ou o endereço IP privado.

Note

Se o parâmetro não existir, crie um novo parâmetro de string e nomeie-o como auth-token-verifier.

4. Escolha OK e feche o Editor do Registro do Windows.
5. Interrompa e reinicie o servidor NICE DCV. Para obter mais informações, consulte [Interromper o servidor do NICE DCV](#) e [Iniciar o servidor do NICE DCV](#) no Guia do administrador do NICE DCV.

Etapa 5: Verificar as instalações

Tópicos

- [Verificar o Atendente](#)
- [Verificar o Broker](#)

Verificar o Atendente

Depois de instalar o Broker e o Atendente, certifique-se de que o Atendente esteja em execução e que seja capaz de se conectar ao Broker.

Host do Atendente do Linux

O comando a ser executado depende da versão.

- Desde a versão 2022.0

No host do Atendente, execute o comando a seguir:

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

- Versões anteriores à versão 2022.0

No host do Atendente, execute o comando a seguir e especifique o ano, o mês e o dia atuais.

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

Por exemplo

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

Se o Atendente estiver em execução e conseguir se conectar ao broker, o comando deverá retornar `success`.

Se o comando retornar uma saída diferente, inspecione o arquivo de log do Atendente para obter mais informações. Os arquivos de log estão localizados aqui: `/var/log/dcv-session-manager-agent/`.

Host do Atendente do Windows

Abra o arquivo de log do Atendente, localizado em `C:\ProgramData\NICE\DCVSessionManagerAgent\log`.

Se o arquivo de log incluir uma linha semelhante à abaixo, o Atendente estará em execução e será capaz de se conectar ao Broker.

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

Se seu arquivo de log não tiver uma linha semelhante, inspecione o arquivo de log em busca de erros.

Verificar o Broker

Depois de instalar o Broker e o Atendente, certifique-se de que seu Broker esteja em execução e possa ser acessado por seus usuários e aplicativos de front-end.

Em um computador que deveria ser capaz de acessar o Broker, execute o seguinte comando:

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

Se a verificação for bem-sucedida, o Broker retornará o seguinte:

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


Configurar o gerenciador de sessões NICE DCV

Esta seção explica como executar a configuração avançada do gerenciador de sessões.

Tópicos

- [Gerenciador de Sessões de Escalabilidade](#)
- [Usar tags para direcionar servidores NICE DCV](#)
- [Configurar um servidor de autorização externo](#)
- [Como configurar a persistência do agente](#)
- [Integração com o gateway de conexão do NICE DCV](#)
- [Fazer a integração com o Amazon CloudWatch](#)

Gerenciador de Sessões de Escalabilidade

Para permitir a alta disponibilidade e melhorar o desempenho, você pode configurar o gerenciador de sessões para usar vários Atendentes e Agentes. Se você pretende usar vários Atendentes e Agentes, recomendamos instalar e configurar somente um host de Atendente e Agente, criar imagens de máquina da Amazon (AMI) a partir desses hosts e, em seguida, iniciar os hosts restantes pelas AMIs.

Por padrão, o Gerenciador de Sessões é compatível com o uso de vários Atendentes sem nenhuma configuração adicional. No entanto, se você pretende usar vários Agentes, deve usar um balanceador de carga para equilibrar o tráfego entre o cliente de front-end e os Agentes e entre os Agentes e os Atendentes. A instalação e configuração do balanceador de carga são inteiramente de sua propriedade e são gerenciadas por você.

A seção a seguir explica como configurar o Gerenciador de Sessões para usar vários hosts com um Application Load Balancer.

Etapas

- [Etapa 1: Criar um perfil de instância](#)
- [Etapa 2: Preparar o certificado SSL para o balanceador de carga](#)
- [Etapa 3: Criar o Application Load Balancer do Agente](#)
- [Etapa 4: Iniciar os Agentes](#)
- [Etapa 5: Criar o Application Load Balancer do Atendente](#)

- [Etapa 6: Iniciar os atendentes](#)

Etapa 1: Criar um perfil de instância

Você deve anexar um perfil de instância aos hosts do Agente e do Atendente que lhes dê permissão para usar as APIs do Elastic Load Balancing. Para obter mais informações, consulte [Perfis do IAM para o Amazon EC2](#) no Guia do usuário do Amazon EC2.

Como criar um perfil de instância

1. Crie uma função AWS Identity and Access Management (IAM) que defina as permissões a serem usadas no perfil da instância. Use a seguinte política de confiança:

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

Em seguida, anexe a seguinte política:

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

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

Para obter mais informações, consulte [Criar um perfil do IAM](#) no Guia do usuário do IAM.

2. Crie um novo perfil de instância. Para obter mais informações, consulte [create-instance-profile](#) na Referência de comandos da AWS CLI .
3. Adicione o perfil do IAM ao perfil de instância. Para obter mais informações, consulte [add-role-to-instance-profile](#) na Referência de comandos da AWS CLI .
4. Anexe o perfil de instância aos hosts do Agente. Para obter mais informações, consulte [Anexar uma função do IAM a uma instância](#) no Guia do usuário do Amazon EC2.

Etapa 2: Preparar o certificado SSL para o balanceador de carga

Quando você usa HTTPS para o listener do seu load balancer, você deve implantar um certificado SSL no load balancer. O load balancer usa esse certificado para encerrar a conexão e descriptografar solicitações dos clientes antes de enviá-las aos destinos.

Para preparar o certificado SSL

1. Crie uma autoridade de certificação privada (CA) AWS Certificate Manager Private Certificate Authority (ACM PCA). Para obter mais informações, consulte [Procedimentos para criar uma CA](#) no Guia do usuário da AWS Autoridade de Certificação Privada do Certificate Manager.
2. Instale a CA. Para obter mais informações, consulte [Instalando um certificado CA raiz](#) no Guia do usuário da Autoridade de AWS Certificação Privada do Certificate Manager.
3. Solicite um novo certificado privado assinado pela CA. Para o nome do domínio, use `*.region.elb.amazonaws.com` e especifique a Região na qual você pretende criar o balanceador de carga. Para obter mais informações, consulte [Solicitando um certificado privado](#) no Guia do usuário da Autoridade de AWS Certificação Privada do Certificate Manager.

Etapa 3: Criar o Application Load Balancer do Agente

Crie um Application Load Balancer para equilibrar o tráfego entre seus clientes de front-end e os Agentes.

Para criar o load balancer

1. Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.

No painel de navegação, escolha Balanceadores de carga, Criar balanceador de carga. Para o tipo de balanceador de carga, escolha Application Load Balancer.

2. Para Step 1: Configure Load Balancer, faça o seguinte:
 - a. Em Nome, insira um nome descritivo para o balanceador de carga.
 - b. Para Esquema, selecione internet-facing.
 - c. Em Protocolo do balanceador de carga, selecione HTTPS e, em Porta do balanceador de carga, insira 8443.
 - d. Para VPC, selecione a VPC a ser usada e, em seguida, selecione todas as sub-redes nessa VPC.
 - e. Escolha Próximo.
3. Para Etapa 2: Definir configurações de segurança, faça o seguinte:
 - a. Em Tipo de certificado, escolha Escolher um certificado do ACM.
 - b. Em Nome do certificado, selecione o certificado privado que você solicitou anteriormente.
 - c. Escolha Próximo.
4. Para a Etapa 3: Configurar grupos de segurança, crie um novo grupo de segurança ou selecione um grupo de segurança existente que permita o tráfego de entrada e saída entre o cliente de front-end e os Agentes por HTTPS e pela porta 8443.

Escolha Próximo.

5. Para Etapa 4: Configurar roteamento, faça o seguinte:
 - a. Em Grupo-alvo, selecione Novo grupo-alvo.
 - b. Em Nome, insira um nome para o destino.
 - c. Para Tipo de destino, escolha Instância.
 - d. Para Protocolo, selecione HTTPS. Em Porta, insira 8443. Em Versão do protocolo, escolha HTTP1.
 - e. Para o Protocolo de verificação de integridade, escolha HTTPS e, em Caminho, insira /health.
 - f. Escolha Próximo.

6. Para Etapa 5: Registrar alvos, escolha Avançar.
7. Escolha Criar.

Etapa 4: Iniciar os Agentes

Crie um Agente inicial e configure-o para usar o balanceador de carga, criar uma AMI a partir do Agente e, em seguida, usar a AMI para iniciar os Agentes restantes. Isso garante que todos os Agentes estejam configurados para usar a mesma CA e a mesma configuração do balanceador de carga.

Para iniciar os Agentes

1. Inicie e configure o host do Agente inicial. Para obter mais informações sobre como instalar e configurar o Agente, consulte [Etapa 2: Configurar o Broker do Gerenciador de Sessões do NICE DCV](#).

Note

O certificado autoassinado do Agente não é necessário, pois estamos usando um Application Load Balancer.

2. Conecte-se ao Agente, abra o `/etc/dcv-session-manager-broker/session-manager-broker.properties` usando o editor de texto de sua preferência e faça o seguinte:
 - a. Comente o parâmetro `broker-to-broker-discovery-addresses` colocando um hash (`#`) no início da linha.
 - b. Para `broker-to-broker-discovery-aws-region`, insira a Região na qual você criou o application load balancer.
 - c. Para `broker-to-broker-discovery-aws-alb-target-group-arn`, insira o ARN do grupo de destino associado ao balanceador de carga do Agente.
 - d. Salve e feche o arquivo.
3. Pare a instância do Agente.
4. Crie uma AMI a partir da instância do Agente interrompida. Para obter mais informações, consulte [Criar uma AMI do Linux a partir de uma instância](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.
5. Use a AMI para iniciar os Agentes restantes.

6. Atribua o perfil de instância que você criou a todas as instâncias do Agente.
7. Atribua um grupo de segurança que permita o tráfego de rede de Agente para Agente e de Agente para balanceador de carga para todas as instâncias do Agente. Para obter mais informações sobre portas de rede, consulte [Arquivo de configuração do Agente](#).
8. Registre todas as instâncias do Agente como destinos para o balanceador de carga do Agente. Para obter mais informações, consulte [Registrar destinos no grupo de destino](#) no Guia do usuário dos Application Load Balancers.

Etapa 5: Criar o Application Load Balancer do Atendente

Crie um Application Load Balancer para equilibrar os Atendentes e os Agentes.

Para criar o load balancer

1. Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.

No painel de navegação, escolha Balanceadores de carga, Criar balanceador de carga. Para o tipo de balanceador de carga, escolha Application Load Balancer.

2. Para Step 1: Configure Load Balancer, faça o seguinte:
 - a. Em Nome, insira um nome descritivo para o balanceador de carga.
 - b. Para Esquema, selecione internet-facing.
 - c. Em Protocolo do balanceador de carga, selecione HTTPS e, em Porta do balanceador de carga, insira 8445.
 - d. Para VPC, selecione a VPC a ser usada e, em seguida, selecione todas as sub-redes nessa VPC.
 - e. Escolha Próximo.
3. Para Etapa 2: Definir configurações de segurança, faça o seguinte:
 - a. Em Tipo de certificado, escolha Escolher um certificado do ACM.
 - b. Em Nome do certificado, selecione o certificado privado que você solicitou anteriormente.
 - c. Escolha Próximo.
4. Para a Etapa 3: Configurar grupos de segurança, crie um novo grupo de segurança ou selecione um existente que permita o tráfego de entrada e saída dos Atendentes e dos Agentes por HTTPS e pela porta 8445.

Escolha Próximo.

5. Para Etapa 4: Configurar roteamento, faça o seguinte:
 - a. Em Grupo-alvo, selecione Novo grupo-alvo.
 - b. Em Nome, insira um nome para o destino.
 - c. Para Tipo de destino, escolha Instância.
 - d. Para Protocolo, selecione HTTPS. Em Porta, insira 8445. Em Versão do protocolo, escolha HTTP1.
 - e. Para o Protocolo de verificação de integridade, escolha HTTPS e, em Caminho, insira /health.
 - f. Escolha Próximo.
6. Para a Etapa 5: Registrar destinos, selecione todas as instâncias do Agente e escolha Adicionar ao registrado. Escolha Próximo: revisar.
7. Escolha Criar.

Etapa 6: Iniciar os atendentes

Crie um Atendente inicial e configure-o para usar o balanceador de carga, criar uma AMI a partir do Atendente e, em seguida, usar a AMI para iniciar os Atendentes restantes. Isso garante que todos os Atendentes estejam configurados para usar a mesma configuração do balanceador de carga.

Para iniciar os Atendentes

1. Prepare o servidor do NICE DCV. Para ter mais informações, consulte [Etapa 1: Preparar os servidores do NICE DCV](#).
2. Coloque uma cópia da chave pública da CA criada em [Etapa 2: Preparar o certificado SSL para o balanceador de carga](#). Escolha ou crie um diretório legível por qualquer usuário. O arquivo de chave pública da CA também deve ser legível por qualquer usuário.
3. Instale e configure o Atendente. Para obter mais informações sobre como instalar e configurar o Atendente, consulte [Etapa 3: Configurar o Atendente do Gerenciador de Sessões do NICE DCV](#).

Important

Ao modificar o arquivo de configuração do Atendente:

- para o parâmetro `broker_host`, insira o DNS do balanceador de carga do Atendente
- para o parâmetro `ca_file`, insira o caminho para o arquivo de chave pública da CA criado na etapa anterior

4. Configure o servidor do NICE DCV para usar o Agente como servidor de autenticação. Para ter mais informações, consulte [Etapa 4: Configure o servidor do NICE DCV para usar o Broker como servidor de autenticação](#).

Important

Ao modificar o arquivo de configuração do servidor do NICE DCV:

- para o parâmetro `ca-file`, insira o mesmo caminho para o arquivo de chave pública da CA usado na etapa anterior
- para o parâmetro `auth-token-verifier`, use o DNS do balanceador de carga do Atendente para *`broker_ip_or_dns`*

5. Pare a instância do Atendente.
6. Crie uma AMI a partir da instância do Atendente interrompida. Para obter mais informações, consulte [Criar uma AMI do Linux a partir de uma instância](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.
7. Use a AMI para iniciar os Atendentes restantes e atribuir o perfil de instância que você criou a todos eles.
8. Atribua um grupo de segurança que permita ao Atendente balancear a carga do tráfego de rede para todas as instâncias do Atendente. Para obter mais informações sobre as portas de rede, consulte o [arquivo de configuração do Atendente](#).

Usar tags para direcionar servidores NICE DCV

Você pode atribuir tags personalizadas aos Atendentes do Gerenciador de Sessões para ajudar a identificá-los e categorizá-los e os servidores NICE DCV aos quais estão associados. Ao criar uma nova sessão NICE DCV, você pode segmentar um grupo de servidores NICE DCV com base nas tags atribuídas aos respectivos Atendentes. Para obter mais informações sobre como direcionar servidores NICE DCV com base nas tags do Atendente, consulte [CreateSessionRequests](#) no Guia do Desenvolvedor do Gerenciador de Sessões.

Uma tag consiste em um par de chave e valor de tag, e você pode usar qualquer par de informações que faça sentido para seu caso de uso ou ambiente. Você pode escolher marcar Atendentes com base na configuração de hardware do host. Por exemplo, você pode marcar todos os Atendentes com hosts que tenham 4 GB de memória com `ram=4GB`. Ou você pode marcar Atendentes com base na finalidade. Por exemplo, você pode marcar todos os Atendentes em execução nos hosts de produção de `purpose=production`.

Para atribuir tags a um Atendente

1. Usando seu editor de texto preferido, crie um novo arquivo e dê a ele um nome descritivo, por exemplo `agent_tags.toml`. O tipo de arquivo deve ser `.toml`, e o conteúdo do arquivo deve ser especificado no formato de arquivo TOML.
2. No arquivo, adicione cada novo par de chave e valor de tag em uma nova linha usando o formato `key=value`. Por exemplo:

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

3. Abra o arquivo de configuração do Atendente (`/etc/dcv-session-manager-agent/agent.conf` para Linux ou `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` para Windows). Para `tags_folder`, e especifique o caminho para o diretório no qual o arquivo de tag está localizado.

Se o diretório contiver vários arquivos de tag, todas as tags definidas nos arquivos se aplicarão ao Atendente. Os arquivos são lidos em ordem alfabética. Se vários arquivos contiverem uma tag com a mesma chave, o valor será substituído pelo valor do último arquivo lido.

4. Salve e feche o arquivo.
5. Interrompa e reinicie o Atendente.

- Windows

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

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

- Linux

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

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

Configurar um servidor de autorização externo

O servidor de autorização é o servidor que autentica e autoriza os SDKs e atendentes do cliente.

Por padrão, o gerenciador de sessões usa o agente como servidor de autorização para gerar tokens de acesso OAuth 2.0 para SDKs de clientes e declarações de software para atendentes. Se você usar o agente como servidor de autorização, não será preciso configurar mais nada.

Você pode configurar o gerenciador de sessões para usar o Amazon Cognito como um servidor de autorização externo em vez do agente. Para mais informações sobre o Amazon Cognito, consulte o [Guia do desenvolvedor do Amazon Cognito](#).

Usar o Amazon Cognito como servidor de autorização

1. Criar um grupo de usuários do Amazon Cognito Para mais informações sobre grupos de usuários, consulte [Atributos do Amazon Cognito](#) no Guia do desenvolvedor do Amazon Cognito.

Use o comando [create-user-pool](#) e especifique o nome do pool e a região na qual criá-lo.

Neste exemplo, demos o nome de `dcv-session-manager-client-app` ao pool e o criamos em `us-east-1`.

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

Exemplo de saída

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

```
}
```

Anote o valor do `userPoolId`, pois será necessário na próxima etapa.

2. Crie um novo domínio para o grupo de usuários. Use o comando [create-user-pool-domain](#) e especifique um nome de domínio e o `userPoolId` do grupo de usuários que você criou na etapa anterior.

Neste exemplo, o domínio se chama `mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE` e o criamos em `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
```

Exemplo de saída

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

O formato do domínio do grupo de usuários é o seguinte:

`https://domain_name.auth.region.amazoncognito.com`. Neste exemplo, o domínio do grupo de usuários é `https://mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE.auth.us-east-1.amazoncognito.com`.

3. Crie um cliente de grupo de usuários. Use o comando [create-user-pool-client](#) e especifique o `userPoolId` do grupo de usuários que você criou, o nome do cliente e a região na qual criá-lo. Além disso, inclua a opção `--generate-secret` para especificar que deseja gerar um segredo para o cliente do grupo de usuários que está sendo criado.

Nesse caso, o nome do cliente é `dcv-session-manager-client-app` e o criamos na região `us-east-1`.

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

Exemplo de saída

```
{  
  "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

Anote o `ClientId` e o `ClientSecret`. Você precisará fornecer essas informações aos desenvolvedores quando eles solicitarem tokens de acesso para as solicitações das APIs.

4. Crie um servidor de recursos OAuth2.0 para o grupo de usuários. Um servidor de recursos é um servidor para recursos com acesso protegido. Ele trabalha com solicitações autenticadas para tokens de acesso.

Use o comando [create-resource-server](#) e especifique o `userPoolId` do grupo de usuários, o identificador e nome exclusivos para o servidor de recursos, o escopo e a região na qual criá-lo.

Neste exemplo, usamos `dcv-session-manager` como identificador e nome e usamos `sm_scope` como nome e descrição do escopo.

```
$ 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
```

Exemplo de saída

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

5. Atualize o cliente do grupo de usuários.

Use o comando [update-user-pool-client](#). Especifique o `userPoolId` do grupo de usuários, o `ClientId` do cliente do grupo de usuários e a região. Para `--allowed-o-auth-flows`, especifique `client_credentials` para indicar que o cliente deve obter tokens de acesso do endpoint do token usando uma combinação de ID e segredo do cliente. Para `--allowed-o-auth-scopes`, especifique o identificador do servidor de recursos e o nome do escopo da seguinte forma: *resource_server_identifier/scope_name*. Inclua `--allowed-o-auth-flows-user-pool-client` para indicar que o cliente tem permissão para seguir o protocolo OAuth ao interagir com grupos de usuários do Cognito.

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

Exemplo de saída

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

```

    "Allowed0AuthFlows": [
      "client_credentials"
    ],
    "Allowed0AuthScopes": [
      "dcv-session-manager/sm_scope"
    ],
    "Allowed0AuthFlowsUserPoolClient": true
  }
}

```

Note

O grupo de usuários agora está pronto para fornecer e autenticar tokens de acesso. Neste exemplo, o URL do servidor de autorização é `https://cognito-idp.us-east-1.amazonaws.com/us-east-1_QLEXAMPLE/.well-known/jwks.json`.

6. Teste a configuração.

```

$ 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"

```

Exemplo de saída

```

{
  "access_token": "eyJraWQiOiJGQ0VaRFpJUUpT3NSaW41MmtqaDdEbTZYb0RnSTQ5b2VUT0cxUU1Q2VJPSIsImF0IjoiZm9udjTlZGScR0dZtId5dThkpEziSx0YwiiWe9crAlqoazlDcCsUJHIXDtgKW64pSj3-uQQGg1Jv_tyVjhrA4JbD0k67WS2V9NW-uZ7t4zwwaUm0i3KzpBmi54fpVgPaewiVlUm_aS4LUFcWT6hVJjiZF7om7984qb2g0a14iZxpXPBJTZX_gtG9EtvnS9U",
  "expires_in": 3600,
  "token_type": "Bearer"
}

```

7. Registre o servidor de autorização externo para uso com o corretor usando o comando [register-auth-server](#).

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

Agora, desenvolvedores podem usar o servidor para solicitar tokens de acesso. Ao solicitar tokens de acesso, forneça o ID do cliente, o segredo do cliente e o URL do servidor gerados aqui. Para mais informações sobre a solicitação de tokens de acesso, consulte [Criar um token de acesso e fazer uma solicitação de API](#) no Guia do desenvolvedor para o gerenciador de sessões do NICE DCV.

Como configurar a persistência do agente

Os agentes do Gerenciador de Sessões oferecem suporte à integração com bancos de dados externos. O banco de dados externo permite que o Gerenciador de Sessões mantenha os dados de status e as chaves para que estejam disponíveis posteriormente. Na verdade, os dados do agente são distribuídos pelo cluster, tornando-o suscetível à perda de dados se um host precisar ser reinicializado ou se um cluster for encerrado. Com esse atributo ativado, você pode adicionar e remover nós do agente. Além disso, você pode interromper um cluster e reiniciá-lo, sem a necessidade de regenerar chaves ou perder informações sobre qual Servidor NICE DCV Server está aberto ou fechado.

Os seguintes tipos de informações podem ser configurados para persistir:

- Chaves para configurar sessões para estabelecer conexão com clientes
- Dados das sessões de bordo
- Status do servidor NICE DCV

O Gerenciador de Sessões NICE DCV oferece suporte aos bancos de dados DynamoDB, MariaDB e MySQL. Você deve configurar e gerenciar um desses bancos de dados para usar esse atributo. Se suas máquinas com agentes estiverem hospedadas no Amazon EC2, recomendamos usar o DynamoDB como o banco de dados externo, pois ele não requer nenhuma configuração adicional.

Note

Você pode incorrer em custos adicionais ao executar um banco de dados externo. Para ver informações sobre os preços do DynamoDB, consulte [Preços da capacidade provisionada](#).

Configure o agente para persistir no DynamoDB

Configure os agentes para começarem a armazenar seus dados no DynamoDB:

1. Abra o `/etc/dcv-session-manager-broker/session-manager-broker.properties` usando o editor de texto de sua preferência e faça as seguintes edições:
 - Definir `enable-persistence = true`
 - Definir `persistence-db = dynamodb`
 - Para `dynamodb-region`, especifique a região da `&aws;` em que você deseja armazenar as tabelas que contêm os dados do agente. Para obter uma lista das regiões compatíveis, consulte [Endpoints de serviço do DynamoDB](#).
 - Para `dynamodb-table-rcu`, especifique a quantidade de Unidades de capacidade de leitura (RCU) para as quais cada tabela dá suporte. Para obter mais informações sobre a RCU, consulte [Capacidade provisionada do DynamoDB](#).
 - Para `dynamodb-table-wcu`, especifique a quantidade de Unidades de capacidade de leitura (WCU) para as quais cada tabela dá suporte. Para obter mais informações sobre a WCU, consulte [Capacidade provisionada do DynamoDB](#).
 - Para `dynamodb-table-name-prefix`, especifique o prefixo que é adicionado a cada tabela do DynamoDB (útil para distinguir vários clusters de agentes usando a mesma conta). São permitidos somente caracteres alfanuméricos, ponto, traço e sublinhado.
2. Pare todos os agentes no cluster. Para cada agente, execute o seguinte comando:

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

3. Certifique-se de que todos os agentes no cluster sejam interrompidos e reinicie todos eles. Inicie cada agente executando o seguinte comando:

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

O host do agente deve ter permissão para chamar as APIs do DynamoDB. Em instâncias do Amazon EC2, as credenciais são recuperadas automaticamente usando o serviço de metadados do Amazon EC2. Se precisar especificar credenciais diferentes, você pode defini-las usando uma das técnicas de recuperação de credenciais compatíveis (como propriedades do sistema Java ou variáveis de ambiente). Para obter mais informações, consulte [Fornecimento e recuperação de credenciais da &aws;](#).

Configure o agente para persistir no MariaDB/MySQL

Note

O arquivo `/etc/dcv-session-manager-broker/session-manager-broker.properties` contém dados confidenciais. Por padrão, seu acesso de gravação é restrito à raiz e seu acesso de leitura é restrito à raiz e ao usuário que executa o Broker. Por padrão, esse é o usuário `dcvsmbroker`. O Broker verifica na inicialização se o arquivo tem as permissões esperadas.

Configure os agentes para começarem a persistir seus dados no MariaDB/MySQL:

1. Abra o `/etc/dcv-session-manager-broker/session-manager-broker.properties` usando o editor de texto de sua preferência e faça as seguintes edições:

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

Nessa configuração, `<db_endpoint>` é o endpoint do banco de dados, `<db_port>` é a porta do banco de dados e `<db_name>` é o nome do banco de dados.

- Para `jdbc-user`, especifique o nome do usuário que tem acesso ao banco de dados.
 - Para `jdbc-password`, especifique o nome do usuário que tem acesso ao banco de dados.
2. Pare todos os agentes no cluster. Para cada agente, execute o seguinte comando:

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

3. Certifique-se de que todos os agentes no cluster sejam interrompidos e reinicie todos eles. Para cada agente, execute o seguinte comando:

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

Integração com o gateway de conexão do NICE DCV

O [gateway de conexão do NICE DCV](#) é um pacote de software instalável que permite aos usuários acessar uma frota de servidores NICE DCV usando apenas um ponto de acesso a uma LAN ou VPC.

Se a sua infraestrutura tem servidores NICE DCV que podem ser acessados por meio do gateway de conexão do NICE DCV, você pode configurar o gerenciador de sessões para integrar o gateway de conexão do NICE DCV. Seguindo as etapas descritas na seção a seguir, o agente atuará como um [resolvedor de sessões](#) para o gateway de conexão. Em outras palavras, o agente vai expor um endpoint HTTP adicional. O gateway de conexão fará chamadas de API para o endpoint a fim de trazer as informações necessárias para rotear as conexões NICE DCV para o host que o agente selecionar.

Configurar o agente do gerenciador de sessões como um resolvedor de sessões para o gateway de conexão do NICE DCV

Lado do agente do gerenciador de sessões

1. Abra o `/etc/dcv-session-manager-broker/session-manager-broker.properties` usando o editor de texto que preferir e aplique as mudanças a seguir:
 - Definir `enable-gateway = true`
 - Defina `gateway-to-broker-connector-https-port` como uma porta TCP livre (o padrão é 8447)
 - Defina `gateway-to-broker-connector-bind-host` como o endereço IP do host em que o agente se vincula às conexões do gateway de conexão do NICE DCV (o padrão é 0.0.0.0)
2. Em seguida, execute os seguintes comandos para parar e reiniciar o agente:

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

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

3. Traga uma cópia do certificado assinado pelo agente e coloque no seu diretório de usuários.

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

Você precisará dele quando for instalar o gateway de conexão do NICE DCV na próxima etapa.

Lado do gateway de conexão do NICE DCV

- Siga a [seção](#) na documentação do gateway de conexão do NICE DCV.

Como o gateway de conexão do NICE DCV faz chamadas de API HTTP para o agente, se o agente estiver usando um certificado que ele mesmo assinou, você precisará copiar o certificado do agente para o host do gateway de conexão do NICE DCV (obtido na etapa anterior) e definir o parâmetro `ca-file` na seção `[resolver]` para configuração do gateway de conexão do NICE DCV.

Opcional – Ativar a autenticação do cliente TLS

Depois de concluir a etapa anterior, o gerenciador de sessões e o gateway de conexão conseguem se comunicar através de um canal seguro, em que o gateway pode verificar a identidade dos agentes do gerenciador de sessões. Se você precisa que os agentes do gerenciador de sessões também validem a identidade do gateway de conexão antes de estabelecer o canal seguro, habilite o atributo de autenticação do cliente TLS, seguindo as etapas na próxima seção.

Note

Se o gerenciador de sessões estiver por trás de um balanceador de carga, a autenticação do cliente TLS não poderá ser habilitada com balanceadores de carga que tenham terminação de conexão TLS, como Application Load Balancers (ALBs) ou Gateway Load Balancers (GLBs). São aceitos apenas balanceadores de carga sem terminação TLS, como Network Load Balancers (NLBs). Se você usa ALBs ou GLBs, é possível assegurar um nível a mais de segurança ao garantir que somente grupos de segurança específicos consigam contatar os balanceadores de carga. Mais informações sobre grupos de segurança aqui: [Grupos de segurança para sua VPC](#)

Lado do agente do gerenciador de sessões

1. Para habilitar a autenticação do cliente TLS para a comunicação entre os agentes do gerenciador de sessões e o gateway de conexão do NICE DCV, siga as próximas etapas:
2. Gere as chaves e os certificados necessários executando: A saída do comando informará a pasta em que as credenciais foram geradas e a senha usada para criar o arquivo TrustStore.

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

3. Coloque no diretório de usuários uma cópia da chave privada e do certificado autoassinado do gateway de conexão do NICE DCV. Você precisará dela quando for ativar a autenticação de cliente TLS no gateway de conexão do NICE DCV na próxima etapa.

```
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. Em seguida, abra `/etc/dcv-session-manager-broker/session-manager-broker.properties` usando o editor de texto que preferir e faça o seguinte:

- Defina `enable-tls-client-auth-gateway` como `true`
- Defina `gateway-to-broker-connector-trust-store-file` para o caminho do arquivo TrustStore criado na etapa anterior
- Defina `gateway-to-broker-connector-trust-store-pass` para a senha usada para criar o arquivo TrustStore na etapa anterior

5. Em seguida, execute o seguinte comando para parar e reiniciar o agente:

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

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

Lado do gateway de conexão do NICE DCV

- Siga a [seção](#) na documentação do gateway de conexão do NICE DCV.
 - Use o caminho completo do arquivo de certificado que você copiou na etapa anterior ao definir o parâmetro `cert-file` na seção `[resolver]`
 - Use o caminho completo do arquivo de chave que você copiou na etapa anterior ao definir o parâmetro `cert-key-file` na seção `[resolver]`

Servidor NICE DCV do gerenciador de sessões do NICE DCV – mapeamento de DNS

O gateway de conexão do NICE DCV requer os nomes DNS dos servidores NICE DCV para se conectar às instâncias do servidor DCV. Esta seção mostra como você pode definir um arquivo JSON contendo o mapeamento entre cada servidor DCV e seu respectivo nome DNS.

Estrutura do arquivo

O mapeamento consiste em uma lista de objetos JSON com os seguintes campos:

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

Onde:

ServerIdType:

Identifica a qual tipo de id o valor se refere; atualmente, os valores disponíveis são `ipAddress`, `agentServerId` e `instanceId`:

Ip:

Disponível para o Amazon EC2 e para infraestruturas on premise; pode ser obtido rapidamente pelos administradores do sistema com um comando `ifconfig` (Linux) ou `ipconfig` (Windows). Essas informações também estão disponíveis na resposta da API `DescribeServers`.

Id:

Disponível tanto para o Amazon EC2 quanto para infraestruturas on premise; o atendente do gerenciador de sessões cria um novo identificador único universal (UUID) toda vez que o

nome do host ou o endereço IP são alterados. Essas informações também estão disponíveis na resposta da API DescribeServers.

Host.Aws.Ec2InstanceId:

Disponível somente para instâncias do Amazon EC2, identifica exclusivamente uma máquina e não muda após a reinicialização da instância. Pode ser recuperado no host contatando <http://169.254.169.254/latest/meta-data/instance-id>. Essas informações também estão disponíveis na resposta da API DescribeServers.

ServerId:

Um ID do tipo especificado que identifica exclusivamente cada servidor NICE DCV na rede.

DnsNames:

O objeto que contém os nomes DNS associados ao servidor NICE DCV que esse objeto conterá:

InternalDnsNames:

O nome DNS usado pelo gateway de conexão do NICE DCV para se conectar à instância.

Use os comandos `register-server-dns-mapping` da CLI do agente do gerenciador de sessões para carregar o mapeamento de um arquivo (referência da página de comando: [register-server-dns-mapping](#)) e `describe-server-dns-mappings` para listar os mapeamentos atualmente carregados no agente do gerenciador de sessões (referência da página de comando: [describe-server-dns-mappings](#)).

Persistência

É altamente recomendável que você ative o atributo de persistência do agente do gerenciador de sessões para se proteger contra a perda de mapeamento quando vários agentes ou todo o cluster ficam inativos. Para mais informações sobre como habilitar a persistência de dados, consulte [Configurar a persistência do agente](#)

Fazer a integração com o Amazon CloudWatch

O gerenciador de sessões faz integração com o Amazon CloudWatch para agentes rodando em instâncias do Amazon EC2 e também para agentes executados em hosts on-premises.

O Amazon CloudWatch monitora os recursos da Amazon Web Services (AWS) e as aplicações executadas na AWS em tempo real. Você pode usar o CloudWatch para coletar e monitorar

métricas, que são as variáveis mensuráveis que ajudam você a avaliar seus recursos e aplicativos. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch](#).

Você pode configurar o agente do gerenciador de sessões para enviar as seguintes métricas ao Amazon CloudWatch:

- `Number of DCV servers` – O número de servidores DCV gerenciados pelo agente.
- `Number of ready DCV servers` – O número de servidores DCV no status READY gerenciados pelo agente.
- `Number of DCV sessions` – O número de sessões DCV gerenciadas pelo agente.
- `Number of DCV console sessions` – O número de sessões do console DCV gerenciadas pelo agente.
- `Number of DCV virtual sessions` – O número de sessões virtuais DCV gerenciadas pelo agente.
- `Heap memory used` – A quantidade de memória heap usada pelo agente.
- `Off-heap memory used` – A quantidade de memória off-heap usada pelo agente.
- `Describe sessions request time` – Tempo necessário para concluir as solicitações da API `DescribeSessions`.
- `Delete sessions request time` – Tempo necessário para concluir as solicitações da API `DeleteSessions`.
- `Create sessions request time` – Tempo necessário para concluir as solicitações da API `CreateSessions`.
- `Get session connection data request time` – O tempo necessário para concluir as solicitações da API `GetSessionConnectionData`.
- `Update session permissions request time` – O tempo necessário para concluir as solicitações da API `UpdateSessionPermissions`.

Configurar o Broker para enviar métricas ao Amazon CloudWatch

1. Abra o `/etc/dcv-session-manager-broker/session-manager-broker.properties` usando o editor de texto que preferir e adicione o seguinte:
 - Defina `enable-cloud-watch-metrics` como `true`
 - Para `cloud-watch-region`, especifique a região na qual coletar as métricas.

Note

Se o agente estiver rodando em uma instância do Amazon EC2, esse parâmetro é opcional. A região é trazida automaticamente do serviço de metadados de instância (IMDS). Se você estiver rodando o agente em um host on-premises, esse parâmetro é obrigatório.

2. Interrompa e reinicie o agente.

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

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

O host do agente também deve ter permissão para chamar a API `cloudwatch:PutMetricData`. As credenciais da AWS podem ser trazidas usando uma das técnicas de obtenção de credenciais aceitas. Para obter mais informações, consulte [Fornecimento e recuperação de credenciais da AWS](#).

Fazer upgrade do Gerenciador de Sessões do NICE DCV

O tópico a seguir descreve como atualizar o Gerenciador de Sessões.

Note

É altamente recomendável que você atualize todos os Atendentes do Gerenciador de Sessões antes de atualizar os Agentes do Gerenciador de Sessões para evitar problemas de incompatibilidade caso novos atributo sejam introduzidos.

Tópicos

- [Fazer upgrade do Atendente do Gerenciador de Sessões do NICE DCV](#)
- [Fazer upgrade do Agente do Gerenciador de Sessões do NICE DCV](#)

Fazer upgrade do Atendente do Gerenciador de Sessões do NICE DCV

Linux host

Note

As instruções a seguir são para instalar o Atendente em hosts x86 de 64 bits. Para instalar o Atendente em hosts ARM de 64 bits, para Amazon Linux, RHEL e Centos, substitua *x86_64* por *aarch64* e, para Ubuntu, substitua *amd64* por *arm64*.

Para atualizar o Atendente em um host Linux

1. Execute o seguinte comando para interromper o Atendente.

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

2. Baixe o pacote de instalação.
 - Amazon Linux 2, RHEL 7.x e CentOS 7.x

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

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

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

- Ubuntu 20.04

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

- SUSE Linux Enterprise 12

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

- SUSE Linux Enterprise 15

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

3. Instale o pacote .

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

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

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

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

- Ubuntu 20.04

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

- SUSE Linux Enterprise 12

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

- SUSE Linux Enterprise 15

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

4. Para iniciar o Atendente, execute o comando a seguir.

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

Windows host

Para atualizar o Atendente em um host Windows

1. Interrompa o serviço do Atendente. Execute os comandos a seguir no prompt de comando.

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

2. Baixe o [instalador do Atendente](#).
3. Execute o instalador. Na tela Welcome, escolha Next.
4. Na tela do EULA (Contrato de Licença do Usuário Final), leia com atenção o contrato de licença e, se você concordar, selecione Aceito os termos e escolha Avançar.
5. Para iniciar a instalação, escolha Instalar.
6. Reinicie o serviço do Atendente. Execute os comandos a seguir no prompt de comando.

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

Fazer upgrade do Agente do Gerenciador de Sessões do NICE DCV

Para atualizar o Agente

1. Conecte-se ao host no qual você pretende atualizar o Agente.

2. Pare o serviço do Agente.

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

3. Baixe o pacote de instalação.

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

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

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

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

- Ubuntu 20.04

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

4. Instale o pacote .

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

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

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

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

- Ubuntu 20.04

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

5. Inicie o serviço do Agente e certifique-se de que ele seja iniciado automaticamente sempre que a instância for executada.

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

Referência da CLI do agente

Esta seção explica como usar os comandos da interface de linha de comandos (CLI) do agente.

Use os comandos a seguir se você usar um servidor de autenticação externo para gerar tokens de acesso do OAuth 2.0:

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

Use os comandos a seguir se você usar o agente do gerenciador de sessões como o servidor de autenticação OAuth 2.0.

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

Use os comandos a seguir para gerenciar o atendente do gerenciador de sessões.

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

Use os comandos a seguir para gerenciar o servidor DCV – arquivo de mapeamento de nomes DNS.

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

register-auth-server

Registra um servidor de autenticação externo para uso com o agente.

Por padrão, o gerenciador de sessões usa o agente como servidor de autorização para gerar tokens de acesso do OAuth 2.0. Se você usar o agente como servidor de autenticação, não será preciso configurar mais nada.

No entanto, se você optar por usar um servidor de autenticação externo, como o Active Directory ou o Amazon Cognito, use esse comando para registrar o servidor de autenticação externo.

Tópicos

- [Sintaxe](#)
- [Opções](#)
- [Exemplo](#)

Sintaxe

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

Opções

--url

O URL do servidor de autenticação externo a ser usado. Você deve anexar `.well-known/jwks.json` ao URL do servidor de autenticação.

Tipo: string

Obrigatório: sim

Exemplo

O exemplo a seguir registra um servidor de autenticação externo com um URL de `https://my-auth-server.com/`.

Comando

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

Resultado

```
Jwk url registered.
```

list-auth-servers

Lista os servidores de autenticação externos que foram registrados.

Tópicos

- [Sintaxe](#)
- [Resultado](#)
- [Exemplo](#)

Sintaxe

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

Resultado

Urls

Os URLs dos servidores de autenticação externos registrados.

Exemplo

O exemplo a seguir lista todos os servidores de autenticação externos que foram registrados.

Comando

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

Resultado


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

unregister-auth-server

Cancela o registro de um servidor de autenticação externo. Quando você cancela o registro de um servidor de autenticação externo, ele não poderá mais ser usado para gerar tokens de acesso do OAuth 2.0.

Tópicos

- [Sintaxe](#)
- [Opções](#)
- [Resultado](#)
- [Exemplo](#)

Sintaxe

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

Opções

--url

O URL do servidor de autenticação externo cujo registro será cancelado. Você deve anexar `.well-known/jwks.json` ao URL do servidor de autenticação.

Tipo: string

Obrigatório: sim

Resultado

Url

O URL do servidor de autenticação externo cujo registro foi cancelado.

Exemplo

O exemplo a seguir registra um servidor de autenticação externo com um URL de `https://my-auth-server.com/`.

Comando

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

Resultado

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

register-api-client

Registra um cliente do gerenciador de sessões no agente e gera credenciais de cliente que podem ser usadas pelo cliente para trazer um token de acesso do OAuth 2.0, o que é necessário para fazer solicitações de API.

Important

Certifique-se de armazenar as credenciais em um lugar seguro. Eles não podem ser recuperados mais tarde.

Esse comando é usado somente se o agente for usado como o servidor de autenticação do OAuth 2.0.

Tópicos

- [Sintaxe](#)
- [Opções](#)
- [Resultado](#)
- [Exemplo](#)

Sintaxe

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

Opções

--name

Um nome exclusivo usado para identificar o cliente do gerenciador de sessões.

Tipo: string

Obrigatório: sim

Resultado

client-id

O ID exclusivo do cliente a ser usado pelo cliente do gerenciador de sessões para trazer um token de acesso do OAuth 2.0.

client-password

A senha a ser usada pelo cliente do gerenciador de sessões para trazer um token de acesso do OAuth 2.0.

Exemplo

O exemplo a seguir registra o cliente `my-sm-client`.

Comando

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

Resultado

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

describe-api-clients

Lista os clientes do gerenciador de sessões que foram registrados no agente.

Tópicos

- [Sintaxe](#)
- [Resultado](#)
- [Exemplo](#)

Sintaxe

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

Resultado

name

O nome exclusivo do cliente do gerenciador de sessões.

id

O ID exclusivo do cliente do gerenciador de sessões.

active

Indica o status do cliente do gerenciador de sessões. Se o cliente estiver ativo, o valor é `true`; caso contrário, é `false`.

Exemplo

O exemplo a seguir lista os clientes registrados do gerenciador de sessões.

Comando

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

Resultado

```
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

Desativa um cliente registrado do gerenciador de sessões. Um cliente desativado do gerenciador de sessões não consegue mais usar as credenciais para recuperar tokens de acesso do OAuth 2.0.

Tópicos

- [Sintaxe](#)
- [Opções](#)
- [Exemplo](#)

Sintaxe

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

Opções

--client -id

O ID do cliente do gerenciador de sessões a ser desativado.

Tipo: string

Obrigatório: sim

Exemplo

O exemplo a seguir mostra a desativação de um cliente do gerenciador de sessões com um ID de cliente de f855b54b-40d4-4769-b792-b727bEXAMPLE.

Comando

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

Resultado

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

renew-auth-server-api-key

Renova as chaves públicas e privadas usadas pelo agente para assinar os tokens de acesso do OAuth 2.0 fornecidos ao cliente do gerenciador de sessões. Se você renovar as chaves, informe a nova chave privada ao desenvolvedor, pois ela é necessária para fazer solicitações de API.

Tópicos

- [Sintaxe](#)
- [Exemplo](#)

Sintaxe

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

Exemplo

O exemplo a seguir renova as chaves pública e privada.

Comando

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

Resultado

```
Keys renewed.
```

generate-software-statement

Gera uma instrução de software.

Os atendentes devem estar registrados no agente para permitir a comunicação. Os atendentes precisam de uma instrução de software para se registrarem no agente. Depois que o atendente tiver uma instrução de software, ele poderá se registrar automaticamente no agente usando o [Protocolo de Registro de Cliente Dinâmico do OAuth 2.0](#). Depois que o agente se registra no atendente, ele recebe um ID e um segredo de cliente, que ele usa para se autenticar no agente.

O agente e o atendente recebem e usam uma instrução de software padrão quando são instalados pela primeira vez. É possível continuar a usar a instrução de software padrão ou gerar uma nova. Se você gerar uma nova instrução de software, coloque a instrução de software em um novo arquivo no atendente e, em seguida, adicione o caminho do arquivo ao parâmetro `agent.software_statement_path` no arquivo `agent.conf`. Depois de fazer isso, pare e reinicie o atendente para que ele possa usar a nova instrução de software para se registrar no agente.

Tópicos

- [Sintaxe](#)
- [Resultado](#)
- [Exemplo](#)

Sintaxe

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

Resultado

software-statement

A instrução do software.

Exemplo

O exemplo a seguir gera uma instrução de software.

Comando

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

Resultado

```
software-statement:  
ewogICJpZCIgOiAiYjc1NTVhN2QtNWl0MC00OTJhLWJjOTUtNmUzOWNhYzkyMDcxIiwKICAiYWN0aXZlIiA6IHRydWUsCi
```

describe-software-statements

Descreve as instruções de software existentes.

Tópicos

- [Sintaxe](#)
- [Resultado](#)
- [Exemplo](#)

Sintaxe

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

Resultado

software-statement

A instrução do software.

issued-at

A data e a hora em que o software foi gerado.

is-active

O estado atual da declaração do software. `true` se a instrução do software estiver ativa; `false` se inativa.

Opções

--software-statement

A instrução do software a ser desativada.

Tipo: string

Obrigatório: sim

Exemplo

O exemplo a seguir desativa uma instrução de software.

Comando

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

Resultado

```
Software statement  
EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWI0MC00TJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj  
deactivated
```

describe-agent-clients

Descreve os atendentes que estão registrados no agente.

Tópicos

- [Sintaxe](#)
- [Resultado](#)
- [Exemplo](#)

Sintaxe

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

Resultado

name

O nome do Atendente.

id

O ID exclusivo do atendente.

active

O status do atendente. `true` se o atendente estiver ativo; `false` se não estiver.

Exemplo

O exemplo a seguir descreve os Atendentes.

Comando

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

Resultado

```
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

Cancele o registro de um atendente do agente.

Tópicos

- [Sintaxe](#)
- [Opções](#)
- [Exemplo](#)

Sintaxe

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

Opções

--client-id

O ID do atendente cujo registro será cancelado.

Tipo: string

Obrigatório: sim

Exemplo

O exemplo a seguir mostra um cancelamento do registro de um atendente.

Comando

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

Resultado

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

register-server-dns-mappings

Registram os servidores DCV – mapeamentos de nomes DNS provenientes de um arquivo JSON.

Sintaxe

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

Opções

--file-path

O caminho do arquivo contendo os servidores DCV – mapeamentos de nomes DNS.

Tipo: string

Obrigatório: sim

Exemplo

O exemplo a seguir mostra o registro dos servidores DCV – mapeamentos de nomes DNS do arquivo `/tmp/mappings.json`.

Comando

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

Resultado

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

describe-server-dns-mappings

Descrevem os servidores DCV atualmente disponíveis – mapeamentos de nomes DNS.

Sintaxe

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

Resultado

serverIdType

O tipo do ID do servidor.

serverId

O ID exclusivo do servidor.

dnsNames

Os nomes DNS internos e externos

internalDnsNames

Os nomes DNS internos

externalDnsNames

Os nomes DNS externos

Exemplo

O exemplo a seguir lista os servidores DCV registrados – mapeamentos de nomes DNS.

Comando

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

Resultado

```
[
{
  "serverIdType" : "Id",
  "serverId" : "192.168.0.1",
  "dnsNames" : {
    "internalDnsName" : "internal1",
    "externalDnsName" : "external1"
  }
}
```

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

Referência do arquivo de configuração

Esta seção fornece informações sobre os arquivos de configuração do Atendente e do Broker.

Tópicos

- [Arquivo de configuração do Broker](#)
- [Arquivo de configuração do Atendente](#)

Arquivo de configuração do Broker

O arquivo de configuração do Broker (`/etc/dcv-session-manager-broker/session-manager-broker.properties`) inclui parâmetros que podem ser configurados para personalizar a funcionalidade Gerenciador de Sessões. Você pode editar o arquivo de configuração usando o editor de texto de sua preferência.

Note

O arquivo `/etc/dcv-session-manager-broker/session-manager-broker.properties` contém dados confidenciais. Por padrão, seu acesso de gravação é restrito à raiz e seu acesso de leitura é restrito à raiz e ao usuário que executa o Broker. Por padrão, esse é o usuário `dcvsmbroker`. O Broker verifica na inicialização se o arquivo tem as permissões esperadas.

A tabela a seguir lista os parâmetros no arquivo de configuração do Broker.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
<code>broker-java-home</code>	Não		Especifica o caminho para o diretório inicial Java que o Broker usará em vez do diretório padrão do sistema. Se definido, o Broker usará <code><broker-j</code>

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
			<p>ava-home>/bin/java na inicialização.</p> <p>Dica: o Broker requer o Java Runtime Environment 11 e, caso ele não exista no sistema, será instalado como uma dependência após a instalação bem-sucedida. Se a versão 11 não estiver definida como ambiente Java padrão, seu diretório inicial poderá ser obtido usando o seguinte comando:</p> <pre>\$ sudo alternatives --display java</pre>
session-screenshots-max-width	Não	160	Especifica a largura máxima, em pixels, das capturas de tela da sessão feitas usando a API GetSessionScreenshots.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
session-screenshots-max-height	Não	100	Especifica a altura máxima, em pixels, das capturas de tela da sessão feitas usando a API GetSessionScreenshots.
session-screenshots-format	Não	png	O formato do arquivo de imagem das capturas de tela da sessão feitas usando a API GetSessionScreenshots.
create-sessions-queue-max-size	Não	1000	O número máximo de solicitações não atendidas da API CreateSessions que podem ser colocadas em fila. Quando a fila está cheia, novas solicitações não atendidas são rejeitadas.
create-sessions-queue-max-timer-seconds	Não	1800	O tempo máximo, em segundos, durante o qual uma solicitação da API CreateSessions não atendida pode permanecer na fila. Se a solicitação não puder ser atendida dentro do período de tempo especificado, ela falhará.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
session-manager-working-path	Sim	/tmp	Especifica o caminho para o diretório no qual o Broker grava os arquivos necessários para operar. Esse diretório deve estar acessível somente para o Broker.
enable-authentication-server	Sim	true	Especifica se o Broker é o servidor de autenticação usado para gerar tokens de acesso do OAuth 2.0 para as APIs do cliente.
enable-authentication	Sim	true	Ativa ou desativa a autorização do cliente. Se você ativar a autorização do cliente, a API do cliente deverá fornecer um token de acesso ao fazer solicitações de API. Se você desativar a autorização do cliente, as APIs do cliente poderão fazer solicitações sem tokens de acesso.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
enable-agent-authorization	Sim	true	Ativa ou desativa a autorização do Atendente. Se você ativar a autorização do Atendente, ele deverá fornecer um token de acesso ao se comunicar com o Broker.
delete-session-duration-hours	Não	1	Especifica o número de horas após as quais as sessões excluídas se tornam invisíveis e não são mais retornadas por chamadas de API DescribeSession .
connect-session-to-token-duration-minutes	Não	60	Especifica o número de minutos durante os quais o token ConnectSession permanece válido.
client-to-broker-connect-https-port	Sim	8443	Especifica a porta HTTPS na qual o Broker escuta as conexões do cliente.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
client-to-broker-connect-bind-host	Não	0.0.0.0	Especifica o endereço IP do host ao qual o Broker se vincula às conexões do cliente.
client-to-broker-connect-key-store-file	Sim		Especifica o armazenamento de chaves usado para conexões de clientes TLS.
client-to-broker-connect-key-store-pass	Sim		Especifica a transmissão do armazenamento de chaves.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
agent-to-broker-connectonhttps-port	Sim	8445	Especifica a porta HTTPS na qual o Broker escuta as conexões do Atendente.
agent-to-broker-connectonbind-host	Não	0.0.0.0	Especifica o endereço IP do host ao qual o Broker se vincula às conexões do Atendente.
agent-to-broker-connectonkey-store-file	Sim		Especifica o armazenamento de chaves usado para conexões de Atendentes TLS.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
agent-to-broker-connectokey-store-pass	Sim		Especifica a transmissão do armazenamento de chaves.
broker-to-broker-port	Sim	47100	Especifica a porta usada para conexões Broker para Broker.
broker-to-broker-bind-host	Não	0.0.0.0	Especifica o endereço IP do host ao qual o Broker se vincula às conexões do Broker para Broker.
broker-to-broker-discovery-port	Sim	47500	Especifica a porta usada pelos Brokers para descobrirem uns aos outros.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
broker-to-broker-discovery-address	Não		<p>Especifica os endereços IP e as portas dos outros brokers na frota no formato <i>ip_address: :port</i>.</p> <p>Se houver vários Brokers, separe os valores com uma vírgula. Se você especificar <code>broker-to-broker-discovery-multicast-group</code>, <code>broker-to-broker-discovery-multicast-port</code>, <code>broker-to-broker-discovery-AWS-region</code> ou <code>broker-to-broker-discovery-AWS-alb-target-group-arn</code>, omite esse parâmetro.</p>

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
<code>broker-to-broker-discovery-multicast-group</code>	Não		Especifica o grupo multicast para a descoberta do Broker para Broker. Se você especificar <code>broker-to-broker-discovery-addresses</code> , <code>broker-to-broker-discovery-aws-region</code> ou <code>broker-to-broker-discovery-AWS-alb-target-group-arn</code> , omite esse parâmetro.
<code>broker-to-broker-discovery-multicast-port</code>	Não		Especifica a porta multicast para a descoberta do Broker para Broker. Se você especificar <code>broker-to-broker-discovery-addresses</code> , <code>broker-to-broker-discovery-AWS-region</code> ou <code>broker-to-broker-discovery-AWS-alb-target-group-arn</code> , omite esse parâmetro.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
<code>broker-to-broker-discovery-AWS-region</code>	Não		Especifica a Região AWS do application load balancer usado para a descoberta de Broker para Broker. Se você especificar <code>broker-to-broker-discovery-multicast-group</code> , <code>broker-to-broker-discovery-multicast-port</code> ou <code>broker-to-broker-discovery-addresses</code> , omite esse parâmetro.
<code>broker-to-broker-discovery-AWS-alb-target-group-arn</code>	Não		O ARN do grupo alvo do application load balancer para descoberta de Broker para Broker. Se você especificar <code>broker-to-broker-discovery-multicast-group</code> , <code>broker-to-broker-discovery-multicast-port</code> ou <code>broker-to-broker-discovery-addresses</code> , omite esse parâmetro.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
broker-to-broker-distributed-memory-max-size-mb	Não	4096	Especifica a quantidade e máxima de memória off-heap a ser usada por um único Broker para armazenar dados da sessão NICE DCV.
broker-to-broker-key-store-file	Sim		Especifica o armazenamento de chaves usado para conexões de Brokers do TLS.
broker-to-broker-key-store-pass	Sim		Especifica a transmissão do armazenamento de chaves.
enable-cloud-watch-metrics	Não	false	Ativa ou desativa as métricas do Amazon CloudWatch. Se você ativar as métricas do CloudWatch, talvez seja necessário especificar um valor para <code>cloud-watch-region</code> .

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
<code>cloud-watch-region</code>	Não	Necessário somente se <code>enable-cloud-watch-metrics</code> estiver definido como <code>true</code> . Se o Broker estiver instalado em uma instância do Amazon EC2, a região será recuperada do IMDS.	A região AWS na qual as métricas do CloudWatch são publicadas.
<code>max-api-requests-per-second</code>	Não	1000	Especifica o número máximo de solicitações que a API do Broker pode processar a cada segundo antes de ser limitada.
<code>enable-rottlir-forwarder</code>	Não	false	Se definido como <code>true</code> , o controle de utilização recupera o IP do chamador do cabeçalho X-Forwarded-For, se presente.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
<code>create- sessions- number- of- retries- on- failure</code>	Não	2	Especifica o número máximo de novas tentativas a serem executadas após a falha de uma solicitação de criação de sessão em um host de servidor NICE DCV. Defina como 0 para nunca realizar novas tentativas em caso de falhas.
<code>autorun- file- arguments- max- size</code>	Não	50	Especifica o número máximo de argumentos que podem ser transmitidos ao arquivo de execução automática.
<code>autorun- file- arguments- max- argument- length</code>	Não	150	Especifica o tamanho máximo em caracteres de cada argumento do arquivo de execução automática.
<code>enable- persistent</code>	Sim	false	Se definido como true, os dados de status do broker serão mantidos em um banco de dados externo.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
<code>persistencedb</code>	Não	Necessário somente se <code>enable-persistence</code> estiver definido como <code>true</code> .	Especifica qual banco de dados é usado para persistência. Os únicos valores compatíveis são: <code>dynamodb</code> e <code>mysql</code> .
<code>dynamodbregion</code>	Não	Só é necessário se <code>enable-persistence</code> estiver definido como <code>true</code> e <code>persistencedb</code> estiver definido como <code>dynamodb</code> .	Especifica a região na qual as tabelas do DynamoDB são criadas e acessadas.
<code>dynamodbtable-rcu</code>	Não	Só é necessário se <code>enable-persistence</code> estiver definido como <code>true</code> e <code>persistencedb</code> estiver definido como <code>dynamodb</code> .	Especifica as Unidades de Capacidade de Leitura (RCU) para cada tabela do DynamoDB. Para obter mais informações sobre RCU, consulte Preços da capacidade provisionada .
<code>dynamodbtable-wcu</code>	Não	Só é necessário se <code>enable-persistence</code> estiver definido como <code>true</code> e <code>persistencedb</code> estiver definido como <code>dynamodb</code> .	Especifica as Unidades de Capacidade de Gravação (WCU) para cada tabela do DynamoDB. Para obter mais informações sobre WCU, consulte Preço da capacidade provisionada .

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
dynamodb-table-name-prefix	Não	Só é necessário se <code>enable-persistence</code> estiver definido como <code>true</code> e <code>persistence-db</code> estiver definido como <code>dynamodb</code> .	<p>Especifique o prefixo que é adicionado a cada tabela do DynamoDB (útil para distinguir vários clusters de brokers usando a mesma conta do AWS). São permitidos somente caracteres alfanuméricos, ponto, traço e sublinhado.</p>
jdbc-connection-url	Não	Só é necessário se <code>enable-persistence</code> estiver definido como <code>true</code> e <code>persistence-db</code> estiver definido como <code>mysql</code> .	<p>Especifica o URL de conexão com o banco de dados MariaDB/MySQL; ele contém o endpoint e o nome do banco de dados. O URL deve ter este formato:</p> <pre>jdbc:mysql://<db_endpoint>:<db_port>/<db_name>?createDatabaseIfNotExist=true</pre> <p>Sendo que <code><db_endpoint></code> é o endpoint do banco de dados MariaDB/MySQL, <code><db_port></code> é a porta do banco de dados e <code><db_name></code> é o nome do banco de dados.</p>

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
<code>jdbc-user</code>	Não	Só é necessário se <code>enable-persistence</code> estiver definido como <code>true</code> e <code>persistence-db</code> estiver definido como <code>mysql</code> .	Especifica o nome do usuário que tem acesso ao banco de dados MariaDB/MySQL.
<code>jdbc-password</code>	Não	Só é necessário se <code>enable-persistence</code> estiver definido como <code>true</code> e <code>persistence-db</code> estiver definido como <code>mysql</code> .	Especifica a senha do usuário que tem acesso ao banco de dados MariaDB/MySQL.
<code>seconds-before-deleting-unreachable-connections</code>	Não	1800	Especifica a quantidade de segundos depois da qual um servidor inacessível é excluído do sistema.

Arquivo de configuração do Atendente

O arquivo de configuração do Atendente (`/etc/dcv-session-manager-agent/agent.conf` para Linux e `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` para Windows) inclui parâmetros que podem ser configurados para personalizar a funcionalidade Gerenciador de Sessões. Você pode editar o arquivo de configuração usando o editor de texto de sua preferência.

A tabela a seguir lista os parâmetros no arquivo de configuração do Atendente.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
<code>agent.tls_host</code>	Sim		Especifica o nome DNS do host do Broker.
<code>agent.tls_port</code>	Sim	8445	Especifica a porta pela qual se comunicar com o Broker.
<code>agent.tls_certificate_file</code>	Não		Só é necessário se <code>tls_strict</code> estiver definido como <code>true</code> . Especifica o caminho para o arquivo do certificado (.pem) necessário para validar o certificado TLS. Copie o certificado autoassinado do Broker para o Atendente.
<code>agent.tls_script_folder</code>	Não	<ul style="list-style-type: none"> <code>/var/lib/dcv-session-manager-agent/init</code> (Linux) 	Especifica o caminho para uma pasta no servidor host usada para armazenar scripts personalizados com permissão para inicializar sessões do servidor NICE DCV quando eles são criados. Você deve especificar um caminho absoluto. A pasta deve estar acessível, e os arquivos devem ser executáveis por usuários que usam o parâmetro de

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
			solicitação InitFile da API CreateSessions.
agent.t _strict	Não	true	Indica se deve ser usada a validação de TLS estrita.
agent.s tware_s tatement_ th	Não		Só é necessário se não for usada a instrução de software padrão. Especifique a o caminho para o arquivo de instruções do software. Para obter mais informações, consulte generate-software-statement .
agent.t s_folde	Não	<ul style="list-style-type: none"> • /etc/dcv-session-manager-agent (Linux) • C:\Program Files\NICE\DCVSessionManagerAgent\conf\tags (Windows) 	Especifica o caminho para a pasta na qual os arquivos de tag estão localizados. Para obter mais informações, consulte Usar tags para direcionar servidores NICE DCV .

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
agent.authorun_file	Não	<ul style="list-style-type: none"> /var/lib/dcv-session-manager-agent/autorun (Linux) C:\ProgramData\NICE\DcvSessionManagerAgent\autorun (Windows) 	<p>Especifica o caminho para uma pasta no servidor host usada para armazenar scripts e aplicativos que podem ser executados automaticamente na inicialização da sessão. Você deve especificar um caminho absoluto. A pasta deve estar acessível, e os arquivos devem ser executáveis por usuários que usam o parâmetro de solicitação AutorunFile da API CreateSessions.</p>
agent._virtual_sessions	Não	-1 (sem limite)	<p>O número máximo de sessões virtuais que podem ser criadas em um servidor NICE DCV usando o Gerenciador de Sessões NICE DCV.</p>
agent._concurrent_sessions_per_user	Não	1	<p>O número máximo de sessões virtuais que podem ser criadas em um servidor NICE DCV por um único usuário usando o Gerenciador de Sessões NICE DCV.</p>

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
agent.ker_upc e_inte l	Não	30	Especifica a quantidade de segundos de espera antes de enviar dados atualizados para o Broker. Os dados enviados incluem o status do servidor NICE DCV e do host, bem como informações atualizadas da sessão. Valores mais baixos tornam o Gerenciador de Sessões mais consciente das mudanças que acontecem no sistema em que o Atendente é executado, mas aumentam a carga do sistema e o tráfego da rede. Valores mais altos diminuem a carga do sistema e da rede, mas o Gerenciador de Sessões se torna menos responsivo às alterações do sistema, portanto, valores maiores do que 120 não são recomendados.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
log.level	Não	info	<p>Especifica o nível de verbosidade dos arquivos de log. Os seguintes níveis de detalhamento estão disponíveis:</p> <ul style="list-style-type: none"> • <code>error</code> — Fornece os mínimos detalhes. Inclui apenas erros. • <code>warning</code> — Inclui erros e avisos. • <code>info</code> — O nível padrão de detalhamento. Inclui erros, avisos e mensagens informativas. • <code>debug</code> — Fornece a maioria dos detalhes. Fornece informações detalhadas que são úteis para depurar problemas.
log.directory	Não	<ul style="list-style-type: none"> • <code>/var/log/dcv-session-manager-agent/</code> (Linux) • <code>C:\ProgramData\NICE\DCVSessionManagerAgent\log</code> (Windows) 	Especifica o diretório no qual os arquivos de log devem ser criados.

Nome do parâmetro	Obrigatório	Valor padrão	Descrição
log.rotation	Não	daily	<p>Especifica a rotação do arquivo de log. Os valores válidos são:</p> <ul style="list-style-type: none"> hourly — Arquivos de log são rotacionados de hora em hora. daily — Arquivos de log são rotacionados diariamente.
log.maxfilesize	Não	10485760	<p>Quando o tamanho de um arquivo de log atingir o tamanho especificado em bytes, ele será rotacionado. Um novo arquivo de log será criado e outros eventos de log serão colocados no novo arquivo.</p>
log.rotationcount	Não	9	<p>O número máximo de arquivos de log preservados na rotação. Sempre que ocorrer uma rotação e esse número for atingido, o arquivo de log mais antigo será excluído.</p>

Notas de versão e histórico de documentos do Gerenciador de Sessões do NICE DCV

Esta página traz as notas da versão e o histórico de documentos do Gerenciador de Sessões do NICE DCV.

Tópicos

- [Notas de versão do Gerenciador de Sessões do NICE DCV](#)
- [Histórico do documento](#)

Notas de versão do Gerenciador de Sessões do NICE DCV

Esta seção fornece uma visão geral das principais atualizações, lançamentos de atributos e correções de erros do Gerenciador de Sessões do NICE DCV. Todas as atualizações são organizadas por data de lançamento. Atualizamos a documentação com frequência para incluir os comentários que recebemos de vocês.

Tópicos

- [2023.1 — 9 de novembro de 2023](#)
- [2023.0-15065 — 4 de maio de 2023](#)
- [2023.0-14852 — 28 de março de 2023](#)
- [2022.2-13907 — 11 de novembro de 2022](#)
- [2022.1-13067 — 29 de junho de 2022](#)
- [2022.0-11952 — 23 de fevereiro de 2022](#)
- [2021.3-11591 — 20 de dezembro de 2021](#)
- [2021.2-11445 — 18 de novembro de 2021](#)
- [2021.2-11190 — 11 de outubro de 2021](#)
- [2021.2-11042 — 1º de setembro de 2021](#)
- [2021.1-10557 — 31 de maio de 2021](#)
- [2021.0-10242 — 12 de abril de 2021](#)
- [2020.2-9662 — 4 de dezembro de 2020](#)
- [2020.2-9508 — 11 de novembro de 2020](#)

2023.1 — 9 de novembro de 2023

Números de compilação	Alterações e correções de bugs
<ul style="list-style-type: none">• Broker: 410• Atendente: 732• CLI: 140	<ul style="list-style-type: none">• Correções de erros e melhorias no desempenho

2023.0-15065 — 4 de maio de 2023

Números de compilação	Alterações e correções de bugs
<ul style="list-style-type: none">• Broker: 392• Atendente: 675• CLI: 132	<ul style="list-style-type: none">• Suporte adicionado para Red Hat Enterprise Linux 9, Rocky Linux 9 e CentOS Stream 9 nas plataformas ARM.

2023.0-14852 — 28 de março de 2023

Números de compilação	Alterações e correções de bugs
<ul style="list-style-type: none">• Broker: 392• Atendente: 642• CLI: 132	<ul style="list-style-type: none">• Suporte adicionado para Red Hat Enterprise Linux 9, Rocky Linux 9 e CentOS Stream 9.

2022.2-13907 — 11 de novembro de 2022

Números de compilação	Alterações e correções de bugs
<ul style="list-style-type: none">• Broker: 382• Atendente: 612• CLI: 123	<ul style="list-style-type: none">• Foi adicionado um campo <code>Substate</code> na resposta <code>DescribeSessions</code>.• Corrigido um problema que poderia fazer com que a CLI não se conectasse ao broker, dependendo do URL em uso.

2022.1-13067 — 29 de junho de 2022

Números de compilação	Alterações e correções de bugs
<ul style="list-style-type: none">• Broker: 355• Atendente: 592• CLI: 114	<ul style="list-style-type: none">• Foi adicionado suporte para executar o broker em instâncias do Graviton AWS.• Suporte adicionado para atendente e broker para Ubuntu 22.04.

2022.0-11952 — 23 de fevereiro de 2022

Números de compilação	Alterações e correções de bugs
<ul style="list-style-type: none">• Broker: 341• Atendente: 520• CLI: 112	<ul style="list-style-type: none">• Foi adicionada a capacidade de rotação de log ao Atendente.• Parâmetro de configuração adicionado para definir a página inicial do Java no Broker.• Melhoria na descarga de dados do cache para o disco no Broker.• Validação de URL corrigida na CLI.

2021.3-11591 — 20 de dezembro de 2021

Números de compilação	Novos recursos
<ul style="list-style-type: none">• Broker: 307• Atendente: 453• CLI: 92	<ul style="list-style-type: none">• Suporte adicionado para integração com o Gateway de Conexão do NICE DCV.• Suporte adicionado para Broker para Ubuntu 18.04 e Ubuntu 20.04.

2021.2-11445 — 18 de novembro de 2021

Números de compilação	Alterações e correções de bugs
<ul style="list-style-type: none">• Broker: 288• Atendente: 413• CLI: 54	<ul style="list-style-type: none">• Corrigido um problema com a validação de nomes de login que incluem um domínio do Windows.

2021.2-11190 — 11 de outubro de 2021

Números de compilação	Alterações e correções de bugs
<ul style="list-style-type: none">• Broker: 254• Atendente: 413• CLI: 54	<ul style="list-style-type: none">• Corrigido um problema na interface da linha de comandos que impedia o início de sessões do Windows.

2021.2-11042 — 1º de setembro de 2021

Números de compilação	Novos recursos	Alterações e correções de bugs
<ul style="list-style-type: none"> • Broker: 254 • Atendente : 413 • CLI: 37 	<ul style="list-style-type: none"> • O Gerenciador de Sessões do NICE DCV agora oferece suporte à interface de linha de comandos (CLI). Você pode criar e gerenciar sessões do NICE DCV na CLI, em vez de chamar as APIs. • O Gerenciador de Sessões do NICE DCV introduziu a persistência de dados do Broker. Para maior disponibilidade, os brokers podem manter as informações do estado do servidor em um armazenamento de dados externo e restaurar os dados na inicialização. 	<ul style="list-style-type: none"> • Ao registrar um servidor de autorização externo, agora você pode especificar o algoritmo que o servidor de autorização usa para assinar Tokens da Web formatados em JSON. Com essa alteração, você pode usar o Azure AD como um servidor de autorização externo.

2021.1-10557 — 31 de maio de 2021

Números de compilação	Novos recursos	Alterações e correções de bugs
<ul style="list-style-type: none"> • Broker: 214 • Atendente : 365 	<ul style="list-style-type: none"> • O Gerenciador de Sessões do NICE DCV adicionou suporte para parâmetros de entrada transmitidos para o arquivo autorun no Linux. • Agora, as propriedades do servidor podem ser transmitidas como requisitos para a API CreateSessions. 	<ul style="list-style-type: none"> • Corrigimos um problema no arquivo autorun no Windows.

2021.0-10242 — 12 de abril de 2021

Números de compilação	Alterações e correções de bugs
<ul style="list-style-type: none"> • Broker: 183 • Atendente: 318 	<ul style="list-style-type: none"> • O Gerenciador de Sessões do NICE DCV introduziu as seguintes novas APIs: <ul style="list-style-type: none"> • OpenServers • CloseServers • DescribeServers • GetSessionScreenshots • Ele também introduziu os seguintes novos parâmetros de configuração: <ul style="list-style-type: none"> • Parâmetros do broker: <code>session-screenshot-max-width</code> , <code>session-screenshot-max-height</code> , <code>session-screenshot-format</code> , <code>create-sessions-queue-max-size</code> e <code>create-sessions-queue-max-time-seconds</code> . • Parâmetros do Atendente: <code>agent.autorun_folder</code> , <code>max_virtual_sessions</code> e <code>max_concurrent_sessions_per_user</code> . • Parâmetros do Atendente: <code>agent.autorun_folder</code> , <code>max_virtual_sessions</code> e <code>max_concurrent_sessions_per_user</code> . • Parâmetros do Atendente: <code>agent.autorun_folder</code> , <code>max_virtual_sessions</code> e <code>max_concurrent_sessions_per_user</code> .

2020.2-9662 — 4 de dezembro de 2020

Números de compilação	Alterações e correções de bugs
<ul style="list-style-type: none"> • Broker: 114 • Atendente: 211 	<ul style="list-style-type: none"> • Corrigimos um problema com os certificados TLS gerados automaticamente que impediam o Broker de iniciar.

2020.2-9508 — 11 de novembro de 2020

Números de compilação	Alterações e correções de bugs
<ul style="list-style-type: none"> • Broker: 78 • Atendente: 183 	<ul style="list-style-type: none"> • A versão inicial do Gerenciador de Sessões do NICE DCV.

Histórico do documento

A tabela a seguir descreve a documentação desta versão do Gerenciador de Sessões do NICE DCV.

Alteração	Descrição	Data
NICE DCV versão 2023.1	O Gerenciador de Sessões do NICE DCV foi atualizado para o NICE DCV 2023.1. Para obter mais informações, consulte 2023.1 — 9 de novembro de 2023 .	9 de novembro de 2023
NICE DCV versão 2023.0	O Gerenciador de Sessões do NICE DCV foi atualizado para o NICE DCV 2023.0. Para obter mais informações, consulte 2023.0-14852 — 28 de março de 2023 .	28 de março de 2023

Alteração	Descrição	Data
NICE DCV versão 2022.2	O Gerenciador de Sessões do NICE DCV foi atualizado para o NICE DCV 2022.2. Para obter mais informações, consulte 2022.2-13907 — 11 de novembro de 2022 .	11 de novembro de 2022
NICE DCV versão 2022.1	O Gerenciador de Sessões do NICE DCV foi atualizado para o NICE DCV 2022.1. Para obter mais informações, consulte 2022.1-13067 — 29 de junho de 2022 .	29 de junho de 2022
NICE DCV versão 2022.0	O Gerenciador de Sessões do NICE DCV foi atualizado para o NICE DCV 2022.0. Para obter mais informações, consulte 2022.0-11952 — 23 de fevereiro de 2022 .	23 de fevereiro de 2022
NICE DCV versão 2021.3	O Gerenciador de Sessões do NICE DCV foi atualizado para o NICE DCV 2021.3. Para obter mais informações, consulte 2021.3-11591 — 20 de dezembro de 2021 .	20 de dezembro de 2021
NICE DCV versão 2021.2	O Gerenciador de Sessões do NICE DCV foi atualizado para o NICE DCV 2021.2. Para obter mais informações, consulte 2021.2-11042 — 1º de setembro de 2021 .	1º de setembro de 2021
NICE DCV versão 2021.1	O Gerenciador de Sessões do NICE DCV foi atualizado para o NICE DCV 2021.1. Para obter mais informações, consulte 2021.1-10557 — 31 de maio de 2021 .	31 de maio de 2021

Alteração	Descrição	Data
NICE DCV versão 2021.0	O Gerenciador de Sessões do NICE DCV foi atualizado para o NICE DCV 2021.0. Para obter mais informações, consulte 2021.0-10242 — 12 de abril de 2021 .	12 de abril de 2021
Versão inicial do Gerenciador de Sessões do NICE DCV	Primeira publicação deste conteúdo.	11 de novembro de 2020

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.