Guia do Desenvolvedor

Gerenciador de Sessões do Amazon DCV

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Gerenciador de Sessões do Amazon DCV: Guia do Desenvolvedor

Copyright © 2025 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 desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é o Gerenciador de Sessões? ... 1
Como o Gerenciador de Sessões funciona ... 1
Recursos ... 3
Introdução à API do Gerenciador de Sessões ... 5
Etapa 1: gerar seu cliente de API ... 5
Etapa 2: registrar a API de cliente ... 6
Etapa 3: obter um token de acesso e faça uma solicitação de API ... 7
Referência da API do Gerenciador de Sessões ... 10
CloseServers ... 10
Parâmetros de solicitação ... 7
Parâmetros de resposta ... 11
Exemplo ... 12
CreateSessions ... 13
Parâmetros de solicitação ... 7
Parâmetros de resposta ... 11
Exemplo ... 12
DescribeServers ... 21
Parâmetros de solicitação ... 7
Parâmetros de resposta ... 11
Exemplo ... 12
DescribeSessions ... 32
Parâmetros de solicitação ... 7
Parâmetros de resposta ... 11
Exemplo ... 12
DeleteSessions ... 39
Parâmetros de solicitação ... 7
Parâmetros de resposta ... 11
Exemplo ... 12
GetSessionConnectionData ... 41
Parâmetros de solicitação ... 7
Parâmetros de resposta ... 11
Informações adicionais ... 44
Exemplo ... 12
GetSessionScreenshots ... 47

Parâmetros de solicitação 7
Parâmetros de resposta 11
Exemplo 12
OpenServers 52
Parâmetros de solicitação 7
Parâmetros de resposta 11
Exemplo 12
UpdateSessionPermissions 54
Parâmetros de solicitação 7
Parâmetros de resposta 11
Exemplo 12
Notas de versão e histórico de documentos 57
Notas da versão 57
2024.0-531 — 17 de junho de 2025 58
2024.0-504— 31 de março de 2025 58
2024.0-493 — 15 de janeiro de 2025 58
2024.0-457: 1º de outubro de 2024 59
2023.1-17652: 1º de agosto de 2024 59
2023.1-16388: 26 de junho de 2024 60
2023.1 — 9 de novembro de 2023 60
2023.0-15065 — 4 de maio de 2023 60
2023.0-14852 — 28 de março de 2023 61
2022.2-13907 — 11 de novembro de 2022 61
2022.1-13067 — 29 de junho de 2022 61
2022.0-11952 — 23 de fevereiro de 2022 62
2021.3-11591 — 20 de dezembro de 2021 62
2021.2-11445 — 18 de novembro de 2021 62
2021.2-11190 — 11 de outubro de 2021 63
2021.2-11042 — 1º de setembro de 2021 63
2021.1-10557 — 31 de maio de 2021 64
2021.0-10242 — 12 de abril de 2021 64
2020.2-9662 — 4 de dezembro de 2020 65
..... 65
Histórico de documentos 65
..... lxix

O que é o Gerenciador de Sessões do Amazon DCV?

> **Note**
>
> O Amazon DCV era conhecido anteriormente como NICE DCV.

O Amazon DCV Session Manager é um conjunto de pacotes de software instaláveis (um agente e um agente) e uma interface de programação de aplicativos (API) que facilita que desenvolvedores e fornecedores independentes de software (ISVs) criem aplicativos front-end que criem e gerenciem programaticamente o ciclo de vida das sessões do Amazon DCV em uma frota de servidores Amazon DCV.

Este guia explica como usar o Session Manager APIs para gerenciar o ciclo de vida das sessões do Amazon DCV. Para obter mais informações sobre como instalar e configurar o Agente do Gerenciador de Sessões e os Atendentes, consulte o Guia do administrador do Gerenciador de Sessões do Amazon DCV.

Pré-requisitos

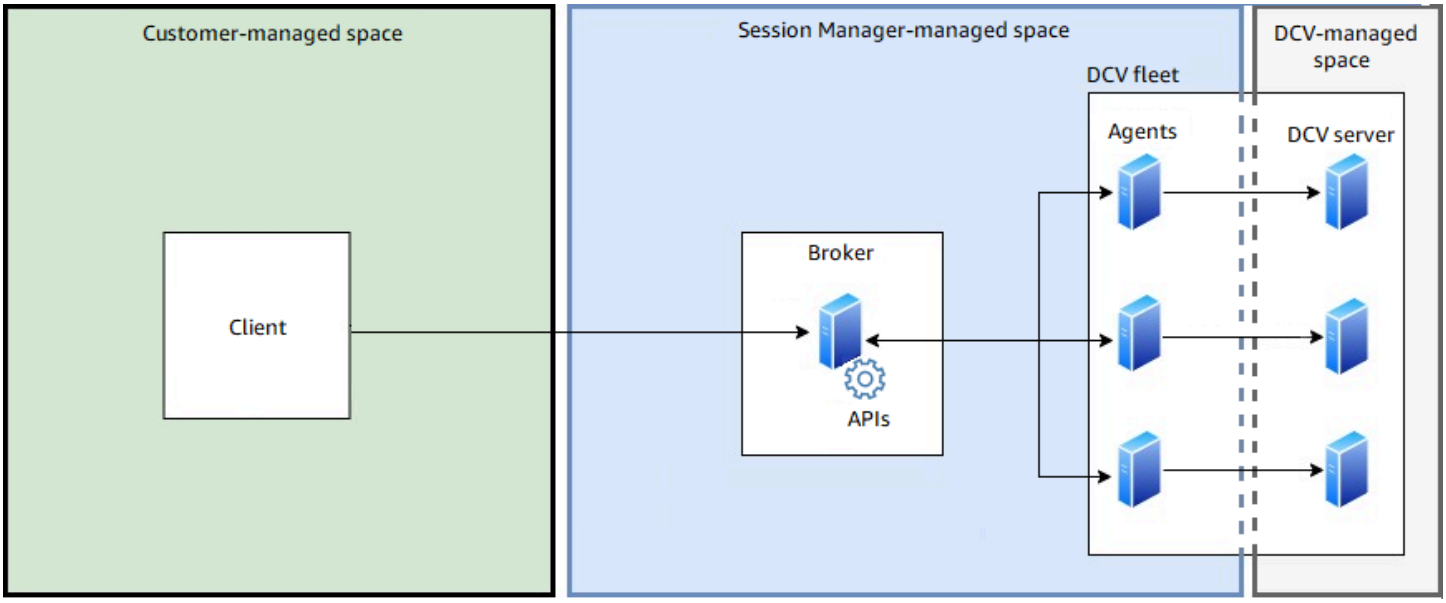
Antes de começar a trabalhar com o Session Manager APIs, certifique-se de estar familiarizado com as sessões do Amazon DCV e do Amazon DCV. Para obter mais informações, consulte o Guia do Administrador do Amazon DCV.

Tópicos

- Como o Gerenciador de Sessões funciona
- Recursos

Como o Gerenciador de Sessões funciona

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



Agente

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

Agente

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

APIs

O Session Manager expõe um conjunto de interfaces de programação de aplicativos REST (APIs) que podem ser usadas para gerenciar sessões do Amazon DCV em uma frota de servidores Amazon DCV. Eles APIs são hospedados e expostos pelo Corretor. Os desenvolvedores podem criar clientes personalizados de gerenciamento de sessões que chamam APIs o.

Cliente

O cliente é o aplicativo ou portal front-end que você desenvolve para chamar o Gerenciador de Sessões APIsque é exposto pelo Broker. Os usuários finais usam o cliente para gerenciar as sessões hospedadas nos servidores Amazon 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, pelo cliente registrado APIs. 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 Amazon DCV

Uma sessão do Amazon DCV é um período em que o servidor Amazon DCV é capaz de aceitar conexões de um cliente. Antes que seus clientes possam se conectar a uma sessão do Amazon DCV, você deve criar uma sessão do Amazon DCV no servidor do Amazon DCV. O Amazon DCV oferece suporte a sessões virtuais e de console, e cada sessão tem um proprietário e um conjunto de permissões especificados. Você usa o Session Manager APIs para gerenciar o ciclo de vida das sessões do Amazon DCV. As sessões do Amazon DCV podem estar em um dos seguintes estados:

- `CREATING` — o Agente 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 Agente e o Atendente podem não conseguir se comunicar.

Recursos

O Gerenciador de Sessões do DCV oferece os seguintes atributo:

- Fornece informações sobre a sessão do Amazon DCV: obtenha informações sobre as sessões em execução em vários servidores Amazon DCV.
- Gerencie o ciclo de vida de várias sessões do Amazon DCV: crie ou exclua várias sessões para vários usuários em vários servidores Amazon DCV com uma solicitação de API.
- Compatível com tags: use tags personalizadas para segmentar um grupo de servidores Amazon DCV ao criar sessões.

- Gerencia permissões para várias sessões do Amazon 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 Amazon DCV.
- Compatível com nuvem e on-premises: use o Gerenciador de Sessões no AWS, on-premises ou com servidores alternativos baseados na nuvem.

Introdução à API do Gerenciador de Sessões

A API do Gerenciador de Sessões do Amazon DCV fornece uma interface automatizada para gerenciar sessões de área de trabalho remota. Por meio dessa API, os desenvolvedores podem criar, listar, iniciar, parar e controlar sessões do DCV programaticamente. Isso permite a integração da funcionalidade do Amazon DCV em aplicativos e fluxos de trabalho personalizados. Ao aproveitar esta API, as organizações podem otimizar o gerenciamento de workloads de visualização remota, automatizando muitas tarefas comuns.

Antes de começar a fazer chamadas para a API do Amazon DCV, você precisará obter um token de acesso que autentique seu aplicativo e o autorize a acessar os recursos necessários. A API Amazon DCV usa OAuth 2.0 para autenticação, então você precisará registrar seu aplicativo e recuperar as credenciais necessárias. Depois de ter seu token de acesso, você pode começar a enviar solicitações aos endpoints da API do Amazon DCV para começar a processar dados.

Tópicos

- Etapa 1: gerar seu cliente de API
- Etapa 2: registrar a API de cliente
- Etapa 3: obter um token de acesso e faça uma solicitação de API

Etapa 1: gerar seu cliente de API

O Gerenciador de Sessões APIs é definido em um único arquivo YAML. Eles APIs são baseados na especificação Open API3 4.0, que também define uma interface padrão independente de idioma. RESTful APIs Para obter mais informações, consulte Especificação de OpenAPI.

Usando o arquivo YAML, você pode gerar clientes de API em um dos idiomas compatíveis. Para fazer isso, você deve usar o Swagger Codegen versão 3.0 ou posterior. Para obter mais informações sobre os idiomas compatíveis, consulte o repositório swagger-codegen.

Para gerar o cliente da API

1. Baixe o arquivo YAML da API do Gerenciador de Sessões no Agente do Gerenciador de Sessões. O arquivo YAML está disponível no seguinte URL.

```
https://broker_host_ip:port/dcv-session-manager-api.yaml
```

2. Instale o Swagger Codegen.

   - macOS

```
$ brew install swagger-codegen
```

   - Outras plataformas

```
$ git clone https://github.com/swagger-api/swagger-codegen --branch 3.0.0
```

```
$ cd swagger-codegen
```

3. Gere o cliente da API.

   - macOS

```
$ swagger-codegen generate -i /path_to/yaml_file -l language -o $output_folder
```

   - Outras plataformas

```
$ mvn clean package
```

```
$ java -jar modules/swagger-codegen-cli/target/swagger-codegen-cli.jar generate -
i /path_to/yaml_file -l language -o output_folder
```

Etapa 2: registrar a API de cliente

As solicitações de API usam um token de acesso para verificar suas credenciais. Essas credenciais se baseiam em um ID de cliente e senha de cliente gerados quando seu cliente é registrado no Agente.

Para acessar esse token, você precisa se registrar com o agente. Use register-api-clientpara registrar a API do cliente.

Se você não tiver um ID de cliente e uma senha de cliente para seu cliente, deverá solicitá-los ao administrador do Agente.

Etapa 3: obter um token de acesso e faça uma solicitação de API

Este exemplo explicará as etapas para configurar seu token de acesso e, em seguida, mostrará como fazer uma solicitação básica de API. Isso fornecerá o conhecimento básico para começar a criar aplicativos mais avançados com a API do Amazon DCV.

Neste exemplo, mostraremos como fazer isso usando a API `DescribeSessions`.

Example

Primeiro, importamos os modelos necessários para o aplicativo.

Em seguida, declaramos variáveis para o ID do cliente (`__CLIENT_ID`), a senha do cliente (`__CLIENT_SECRET`) e o URL do Agente, incluindo o número da porta (`__PROTOCOL_HOST_PORT`).

Em seguida, criamos uma função chamada `build_client_credentials` que gera as credenciais do cliente. Para gerar as credenciais do cliente, primeiro você deve concatenar o ID do cliente e a senha do cliente e separar os valores com dois pontos (*`client_ID`*`:`*`client_password`*) e, em seguida, codificar a string inteira em Base64.

```
import swagger_client
import base64
import requests
import json
from swagger_client.models.describe_sessions_request_data import
 DescribeSessionsRequestData
from swagger_client.models.key_value_pair import KeyValuePair
from swagger_client.models.delete_session_request_data import DeleteSessionRequestData
from swagger_client.models.update_session_permissions_request_data import
 UpdateSessionPermissionsRequestData
from swagger_client.models.create_session_request_data import CreateSessionRequestData

__CLIENT_ID = '794b2dbb-bd82-4707-a2f7-f3d9899cb386'
__CLIENT_SECRET = 'MzcxNzJhN2UtYjEzNS00MjNjLTg2N2YtMjFlZmRlZWNjMDU1'
__PROTOCOL_HOST_PORT = 'https://<broker-hostname>:8443'

def build_client_credentials():
    client_credentials = '{client_id}:{client_secret}'.format(client_id=__CLIENT_ID,
 client_secret=__CLIENT_SECRET)
    return base64.b64encode(client_credentials.encode('utf-8')).decode('utf-8')
```

Agora que temos nossas credenciais de cliente, podemos usá-las para solicitar um token de acesso ao Agente. Para fazer isso, criamos uma função chamada `get_access_token`. Você deve chamar um POST em `https://Broker_IP:8443/oauth2/token?grant_type=client_credentials` e fornecer um cabeçalho de autorização, que inclua as credenciais do cliente codificadas em Basic e um tipo de conteúdo de `application/x-www-form-urlencoded`.

```
def get_access_token():
    client_credentials = build_client_credentials()
    headers = {
        'Authorization': 'Basic {}'.format(client_credentials),
        'Content-Type': 'application/x-www-form-urlencoded'
    }
    endpoint = __PROTOCOL_HOST_PORT + '/oauth2/token?grant_type=client_credentials'
    print('Calling', endpoint, 'using headers', headers)
    res = requests.post(endpoint, headers=headers, verify=True)
    if res.status_code != 200:
        print('Cannot get access token:', res.text)
        return None
    access_token = json.loads(res.text)['access_token']
    print('Access token is', access_token)
    return access_token
```

Agora, criamos as funções necessárias para instanciar uma API de cliente. Para instanciar uma API de cliente, você deve especificar a configuração do cliente e os cabeçalhos a serem usados nas solicitações. A função `get_client_configuration` cria um objeto de configuração que inclui o endereço IP e a porta do Agente e o caminho para o certificado autoassinado do Agente, que você deve ter recebido do administrador do Agente. A função `set_request_headers` cria um objeto de cabeçalho de solicitação que inclui as credenciais do cliente e o token de acesso.

```
def get_client_configuration():
    configuration = swagger_client.Configuration()
    configuration.host = __PROTOCOL_HOST_PORT
    configuration.verify_ssl = True
    # configuration.ssl_ca_cert = cert_file.pem
    return configuration

def set_request_headers(api_client):
    access_token = get_access_token()
    api_client.set_default_header(header_name='Authorization',
```

```
                                    header_value='Bearer {}'.format(access_token))

def get_sessions_api():
    api_instance =
 swagger_client.SessionsApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance
```

Por fim, criamos um método principal que chama a API `DescribeSessions`. Para obter mais informações, consulte DescribeSessions.

```
def describe_sessions(session_ids=None, next_token=None, tags=None, owner=None):
    filters = list()
    if tags:
        for tag in tags:
            filter_key_value_pair = KeyValuePair(key='tag:' + tag['Key'],
 value=tag['Value'])
            filters.append(filter_key_value_pair)
    if owner:
        filter_key_value_pair = KeyValuePair(key='owner', value=owner)
        filters.append(filter_key_value_pair)

    request = DescribeSessionsRequestData(session_ids=session_ids, filters=filters,
 next_token=next_token)
    print('Describe Sessions Request:', request)
    api_instance = get_sessions_api()
    api_response = api_instance.describe_sessions(body=request)
    print('Describe Sessions Response', api_response)

def main():
    describe_sessions(
        session_ids=['SessionId1895', 'SessionId1897'],
        owner='an owner 1890',
        tags=[{'Key': 'ram', 'Value': '4gb'}])
```

Referência da API do Gerenciador de Sessões

Essa referência fornece detalhes sobre as ações de API disponíveis, parâmetros necessários e formatos de resposta para permitir que você aproveite de forma eficaz a API do Gerenciador de Sessões em seus próprios sistemas. Usando a API do Gerenciador de Sessões, é possível iniciar, interromper e ver detalhes sobre sessões interativas. Isso permite que você automatize e integre funcionalidades em seus aplicativos e fluxos de trabalho.

Tópicos

- CloseServers
- CreateSessions
- DescribeServers
- DescribeSessions
- DeleteSessions
- GetSessionConnectionData
- GetSessionScreenshots
- OpenServers
- UpdateSessionPermissions

CloseServers

Fecha um ou mais servidores Amazon DCV. Ao fechar um servidor Amazon DCV, você o torna indisponível para colocação de sessões do Amazon DCV. Você não pode criar sessões do Amazon DCV em servidores fechados. Fechar um servidor garante que nenhuma sessão seja executada nele e que os usuários não podem criar sessões nele.

Tópicos

- Parâmetros de solicitação
- Parâmetros de resposta
- Exemplo

Parâmetros de solicitação

`ServerId`

O ID do servidor a ser fechado.

Tipo: string

Obrigatório: Sim

`Force`

Força a operação de fechamento. Se você especificar `true`, o servidor será fechado mesmo que tenha sessões em execução. As sessões continuam sendo executadas.

Tipo: booliano

Obrigatório: não

Parâmetros de resposta

`RequestId`

O ID exclusivo da solicitação.

`SuccessfulList`

Informações sobre os servidores Amazon DCV que foram fechados com sucesso. Essa estrutura de dados inclui o seguinte parâmetro de resposta aninhada:

`ServerId`

O ID do servidor que foi fechado com sucesso.

`UnsuccessfulList`

Informações sobre os servidores Amazon DCV que não puderam ser fechados. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

`CloseServerRequestData`

Informações sobre a solicitação original com falha. Essa estrutura de dados inclui o seguinte parâmetro de resposta aninhada:

ServerId

O ID do servidor Amazon DCV que não pôde ser fechado.

Force

O parâmetro de força solicitado.

FailureCode

O código da falha.

FailureReason

O motivo da falha.

Exemplo

Python

Solicitação

O exemplo a seguir fecha dois servidores Amazon DCV (`serverId1` and `serverId2`). O servidor `serverId2` não existe e resulta em uma falha.

```
from swagger_client.models import CloseServerRequestData

def get_servers_api():
    api_instance =
 swagger_client.ServersApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def close_servers(server_ids):
    request = [CloseServerRequestData(server_id=server_id) for server_id in
 server_ids]
    print('Close Servers Request:', request)
    api_instance = get_servers_api()
    api_response = api_instance.close_servers(body=request)
    print('Close Servers Response:', api_response)
    open_servers(server_ids)

def main():
    close_servers(["serverId1", "serverId2"])
```

Resposta

Esta é uma saída de exemplo.

```
{
    "RequestId": "4d7839b2-a03c-4b34-a40d-06c8b21099e6",
    "SuccessfulList": [
        {
            "ServerId": "serverId1"
        }
    ],
    "UnsuccessfulList": [
        {
            "OpenServerRequestData": {
                "ServerId": "serverId2"
            },
            "FailureCode": "DCV_SERVER_NOT_FOUND",
            "FailureReason": "Dcv server not found."
        }
    ]
}
```

CreateSessions

Cria uma nova sessão do Amazon DCV com os detalhes especificados.

Ações da API

- Parâmetros de solicitação
- Parâmetros de resposta
- Exemplo

Parâmetros de solicitação

Name

Um nome para a sessão.

Tipo: string

Obrigatório: Sim

`Owner`

O nome do proprietário da sessão. Deve ser o nome de um usuário existente no servidor Amazon DCV de destino.

Tipo: string

Obrigatório: Sim

`Type`

O tipo de sessão. Para obter mais informações sobre os tipos de sessões, consulte Introdução às sessões do Amazon DCV no Guia do administrador do Amazon DCV.

Valores válidos: CONSOLE | VIRTUAL

Tipo: string

Obrigatório: Sim

`InitFile`

Compatível com sessões virtuais nos servidores Amazon DCV do Linux. Não é compatível com sessões do console nos servidores Amazon DCV do Windows e do Linux. O caminho para o script personalizado no servidor Amazon DCV a ser executado para inicializar a sessão quando ela for criada. O caminho do arquivo é relativo ao diretório init especificado para o parâmetro de configuração do Atendente `agent.init_folder`. Se o arquivo estiver no diretório init especificado, especifique somente o nome do arquivo. Se o arquivo não estiver no diretório init especificado, especifique o caminho relativo. Para obter mais informações, consulte o arquivo de configuração do Atendente no Guia do administrador do Gerenciador de Sessões do Amazon DCV.

Tipo: String

Obrigatório: Não

`MaxConcurrents`

O número máximo de clientes simultâneos do Amazon DCV.

Tipo: número inteiro

Obrigatório: não

`DcvGlEnabled`

Indica se a sessão virtual está configurada para usar OpenGL baseado em hardware. Compatível somente com sessões virtuais. Este parâmetro não é compatível com servidores Amazon DCV do Windows.

Valores válidos: true | false

Tipo: booliano

Obrigatório: não

`PermissionsFile`

O conteúdo codificado em Base64 do arquivo de permissões. O padrão é o servidor padrão, se omitido. Para obter mais informações, consulte Configurar a autorização do Amazon DCV no Guia do administrador do Amazon DCV.

Tipo: String

Obrigatório: Não

`EnqueueRequest`

Indica se a solicitação deverá ser colocada em fila se ela não puder ser atendida imediatamente.

Tipo: booliano

Padrão: False

Obrigatório: não

`AutorunFile`

Compatível com sessões de console em servidores Amazon DCV do Windows e sessões virtuais em servidores Amazon DCV do Linux. Não é compatível com sessões do console nos servidores Amazon DCV do Linux.

O caminho para um arquivo no servidor host que será executado dentro da sessão. O caminho do arquivo é relativo ao diretório autorun especificado para o parâmetro de configuração do Atendente do `agent.autorun_folder`. Se o arquivo estiver no diretório autorun especificado, especifique somente o nome do arquivo. Se o arquivo não estiver no diretório autorun especificado, especifique o caminho relativo. Para obter mais informações, consulte o arquivo de configuração do Atendente no Guia do administrador do Gerenciador de Sessões do Amazon DCV.

O arquivo é executado em nome do proprietário especificado. O proprietário especificado deve ter permissão para executar o arquivo no servidor. Nos servidores Amazon DCV do Windows, o arquivo é executado quando o proprietário faz login na sessão. Nos servidores Amazon DCV do Linux, o arquivo é executado quando a sessão é criada.

Tipo: String

Obrigatório: Não

`AutorunFileArguments`

Compatível com sessões virtuais nos servidores Amazon DCV do Linux. Não é compatível com sessões do console nos servidores Amazon DCV do Windows e do Linux. Argumentos de linha de comando transmitidos AutorunFileapós sua execução dentro da sessão. Os argumentos são transmitidos na ordem em que aparecem na matriz fornecida. O número máximo permitido de argumentos e o tamanho máximo permitido de cada argumento podem ser configurados. Para obter mais informações, consulte o arquivo de configuração do Agente no Guia do administrador do Gerenciador de Sessões do Amazon DCV.

Tipo: matriz de strings

Obrigatório: não

`DisableRetryOnFailure`

Indica se a solicitação de criação de sessão não deve ser repetida após ela falhar em um host do Amazon DCV por algum motivo. Para obter mais informações sobre criar o mecanismo de nova tentativa de sessão, consulte Arquivo de configuração do Agente no Guia do administrador do Gerenciador de Sessões do Amazon DCV.

Tipo: booliano

Padrão: False

Obrigatório: não

`Requirements`

Os requisitos que o servidor deve atender para realizar a sessão. Os requisitos podem incluir tags do servidor e/ou propriedades do servidor. Tanto as tags quanto as propriedades do servidor são recuperadas chamando a DescribeServersAPI.

Expressões de condição de requisitos:

- *a*! = *b* verdadeiro se não *a* for igual a *b*

- *a*= *b* verdadeiro se *a* for igual a *b*
- *a*> *b* verdadeiro se *a* for maior que *b*
- *a*>= *b* verdadeiro se *a* for maior ou igual a *b*
- *a*< *b* verdadeiro se *a* for menor que *b*
- *a*<= *b* verdadeiro se *a* for menor ou igual a *b*
- *a*= *b* verdadeiro se *a* contiver a string *b*

Requisitos para operadores booleanos:

- *a*e *b* verdadeiro se *a* e *b* forem verdadeiros
- *a*ou *b* verdadeiro se *a* ou *b* são verdadeiros
- não é *a* verdadeiro se *a* for falso

As chaves de tag devem ser prefixadas por `tag:`, as propriedades do servidor devem ser prefixadas por `server:`. As expressões de requisitos oferecem suporte a parênteses `()`.

Exemplos de requisitos:

- `tag:color = 'pink' and (server:Host.Os.Family = 'windows' or tag:color := 'red')`
- `"server:Host.Aws.Ec2InstanceType := 't2' and server:Host.CpuInfo.NumberOfCpus >= 2"`

Valores numéricos podem ser especificados usando a notação exponencial, por exemplo: `"server:Host.Memory.TotalBytes > 1024E6"`.

As propriedades do servidor compatíveis são:

- `Id`
- `Hostname`
- `Version`
- `SessionManagerAgentVersion`
- `Host.Os.BuildNumber`
- `Host.Os.Family`
- `Host.Os.KernelVersion`
- `Host.Os.Name`
- `Host.Os.Version`

- `Host.Memory.TotalBytes`
- `Host.Memory.UsedBytes`
- `Host.Swap.TotalBytes`
- `Host.Swap.UsedBytes`
- `Host.CpuLoadAverage.OneMinute`
- `Host.CpuLoadAverage.FiveMinutes`
- `Host.CpuLoadAverage.FifteenMinutes`
- `Host.Aws.Ec2InstanceId`
- `Host.Aws.Ec2InstanceType`
- `Host.Aws.Region`
- `Host.Aws.Ec2ImageId`
- `Host.CpuInfo.Architecture`
- `Host.CpuInfo.ModelName`
- `Host.CpuInfo.NumberOfCpus`
- `Host.CpuInfo.PhysicalCoresPerCpu`
- `Host.CpuInfo.Vendor`

Tipo: String

Obrigatório: Não

`StorageRoot`

Especifica o caminho para a pasta a ser usada para armazenamento de sessão. Para obter mais informações sobre o armazenamento de sessão do Amazon DCV, consulte Ativar o armazenamento de sessão no Guia do administrador do Amazon DCV.

Tipo: String

Obrigatório: Não

Parâmetros de resposta

`Id`

O ID exclusivo da sessão.

`Name`

O nome da sessão.

`Owner`

O proprietário da sessão.

`Type`

O tipo de sessão.

`State`

O estado da sessão. Se a solicitação for concluída com êxito, a sessão entrará no estado `CREATING`.

`Substate`

O subestado da sessão. Se a solicitação for concluída com êxito, o subestado entrará no subestado `SESSION_PLACING`.

Exemplo

Python

Solicitação

O exemplo a seguir cria três sessões.

```
from swagger_client.models.create_session_request_data import
 CreateSessionRequestData

def get_sessions_api():
    api_instance =
 swagger_client.SessionsApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def create_sessions(sessions_to_create):
    create_sessions_request = list()
    for name, owner, session_type, init_file_path, autorun_file,
 autorun_file_arguments, max_concurrent_clients,\
```

```
            dcv_gl_enabled, permissions_file, requirements, storage_root in
 sessions_to_create:
        a_request = CreateSessionRequestData(
            name=name, owner=owner, type=session_type,
            init_file_path=init_file_path, autorun_file=autorun_file,
 autorun_file_arguments=autorun_file_arguments,
 max_concurrent_clients=max_concurrent_clients,
            dcv_gl_enabled=dcv_gl_enabled, permissions_file=permissions_file,
 requirements=requirements, storage_root=storage_root)
        create_sessions_request.append(a_request)

    api_instance = get_sessions_api()
    print('Create Sessions Request:', create_sessions_request)
    api_response = api_instance.create_sessions(body=create_sessions_request)
    print('Create Sessions Response:', api_response)

def main():
    create_sessions([
    ('session1', 'user1', 'CONSOLE', None, None, None, 1, None, '/dcv/
permissions.file', "tag:os = 'windows' and server:Host.Memory.TotalBytes > 1024", "/
storage/root"),
    ('session2', 'user1', 'VIRTUAL', None, 'myapp.sh', None, 1, False, None, "tag:os
 = 'linux'", None),
    ('session3', 'user1', 'VIRTUAL', '/dcv/script.sh', 'myapp.sh', ['argument1',
 'argument2'], 1, False, None, "tag:os = 'linux'", None),
])
```

Resposta

Esta é uma saída de exemplo.

```
{
              "RequestId": "e32d0b83-25f7-41e7-8c8b-e89326ecc87f",
    "SuccessfulList": [
    {
            "Id": "78b45deb-1163-46b1-879b-7d8fcbe9d9d6",
            "Name": "session1",
            "Owner": "user1",
            "Type": "CONSOLE",
            "State": "CREATING"
    },
    {
            "Id": " a0c743c4-9ff7-43ce-b13f-0c4d55a268dd",
            "Name": "session2",
```

```
            "Owner": "user1",
            "Type": "VIRTUAL",
            "State": "CREATING"
    },
    {
            "Id": " 10311636-df90-4cd1-bcf7-474e9675b7cd",
            "Name": "session3",
            "Owner": "user1",
            "Type": "VIRTUAL",
            "State": "CREATING"
    }
    ],
    "UnsuccessfulList": [
    ]
}
```

DescribeServers

Descreve um ou mais servidores Amazon DCV.

Tópicos

- Parâmetros de solicitação
- Parâmetros de resposta
- Exemplo

Parâmetros de solicitação

`ServerIds`

Os IDs servidores Amazon DCV a serem descritos. Se nenhum IDs for especificado, todos os servidores serão retornados na saída paginada.

Tipo: matriz de strings

Obrigatório: não

`NextToken`

O token a ser usado para recuperar da próxima página de resultados.

Tipo: String

Obrigatório: Não

MaxResults

O número máximo de resultados a serem retornados por solicitação na saída paginada. Quando esse parâmetro é usado, a solicitação retorna resultados somente o número especificado de resultados em uma única página junto com um elemento de resposta `NextToken`. Os resultados restantes da solicitação inicial podem ser vistos enviando outra solicitação com o valor `NextToken` retornado.

Intervalo válido: 1 a 1000

Padrão: 1000

Tipo: número inteiro

Obrigatório: não

Parâmetros de resposta

RequestId

O ID exclusivo da solicitação.

Servers

Informações sobre os servidores Amazon DCV. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

Id

O ID exclusivo do servidor Amazon DCV.

Ip

O endereço IP do servidor Amazon DCV.

Hostname

O nome do host do servidor Amazon DCV.

Endpoints

Informações sobre os endpoints do servidor Amazon DCV. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

IpAddress

O endereço IP do endpoint do servidor.

Port

A porta do endpoint do servidor.

Protocol

O protocolo usado pelo endpoint do servidor. Os possíveis valores incluem:

- `HTTP`— O endpoint usa o protocolo WebSocket (TCP).
- `QUIC` — O endpoint usa o protocolo QUIC (UDP).

WebUrlPath

O caminho do URL da web do endpoint do servidor. Disponível somente para o protocolo HTTP.

Version

A versão do servidor Amazon DCV.

SessionManagerAgentVersion

A versão do Atendente do Gerenciador de Sessões em execução no servidor Amazon DCV.

Availability

A disponibilidade do servidor Amazon DCV. Os possíveis valores incluem:

- `AVAILABLE` — O servidor está disponível e pronto para o posicionamento da sessão.
- `UNAVAILABLE` — O servidor não está disponível e não pode aceitar o posicionamento da sessão.

UnavailabilityReason

O motivo da indisponibilidade do servidor Amazon DCV. Os possíveis valores incluem:

- `SERVER_FULL` — O servidor Amazon DCV atingiu o número máximo de sessões simultâneas que pode ser executado.
- `SERVER_CLOSED`— O servidor Amazon DCV ficou indisponível usando a CloseServerAPI.
- `UNREACHABLE_AGENT` — O Agente do Gerenciador de Sessões não consegue se comunicar com o Atendente do Gerenciador de Sessões no servidor de Amazon DCV.
- `UNHEALTHY_DCV_SERVER` — O Atendente do Gerenciador de Sessões não consegue se comunicar com o servidor Amazon DCV.

- `EXISTING_LOGGED_IN_USER` — (Somente servidores Amazon DCV do Windows) Um usuário está conectado no momento ao servidor Amazon DCV usando RDP.
- `UNKNOWN` — O Agente do Gerenciador de Sessões não consegue determinar o motivo.

`ConsoleSessionCount`

O número de sessões do console no servidor Amazon DCV.

`VirtualSessionCount`

O número de sessões virtuais no servidor Amazon DCV.

`Host`

Informações sobre o servidor host no qual o servidor Amazon DCV está em execução. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

`Os`

Informações sobre o sistema operacional do servidor host. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

`Family`

A família do sistema operacional. Os possíveis valores incluem:

- `windows` — O servidor host está executando um sistema operacional Windows.
- `linux` — O servidor host está executando um sistema operacional Linux.

`Name`

O nome do sistema operacional.

`Version`

A versão do sistema operacional.

`KernelVersion`

(Somente para Linux) A versão do kernel do sistema operacional.

`BuildNumber`

(Somente para Windows) O número da compilação do sistema operacional.

`Memory`

Informações sobre a memória do servidor host. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

TotalBytes

A memória total, em bytes, no servidor host.

UsedBytes

A memória usada, em bytes, no servidor host.

Swap

Informações sobre o arquivo de troca do servidor host. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

TotalBytes

O tamanho total do arquivo de troca, em bytes, no servidor host.

UsedBytes

O tamanho usado do arquivo de troca, em bytes, no servidor host.

Aws

Somente para servidores Amazon DCV em execução em uma EC2 instância da Amazon. AWS-informações específicas. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

Region

A AWS região da EC2 instância da Amazon.

Ec2InstanceType

O tipo de EC2 instância da Amazon.

Ec2InstanceId

O ID da EC2 instância da Amazon.

Ec2ImageId

O ID da EC2 imagem da Amazon.

CpuInfo

Informações sobre o servidor host CPUs. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

Vendor

O fornecedor da CPU do servidor host.

ModelName

O nome do modelo da CPU do servidor host.

Architecture

A arquitetura da CPU do servidor host.

NumberOfCpus

O número de CPUs no servidor host.

PhysicalCorePerCpu

O número de núcleos de CPU por CPU.

CpuLoadAverage

Informações sobre a carga de CPU do servidor host. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

OneMinute

A carga média de CPU durante o último período de 1 minuto.

FiveMinutes

A carga média de CPU durante o último período de 5 minutos.

FifteenMinutes

A carga média de CPU durante o último período de 15 minutos.

Gpus

Informações sobre o servidor host GPUs. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

Vendor

O fornecedor da GPU do servidor host.

ModelName

O nome do modelo da GPU do servidor host.

LoggedInUsers

Os usuários que estão atualmente conectados ao servidor host. Essa estrutura de dados inclui o seguinte parâmetro de resposta aninhada:

Username

O nome de usuário do usuário conectado.

Tags

As tags atribuídas ao servidor. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

Key

A chave de tags.

Value

O valor da tag.

Exemplo

Python

Solicitação

O exemplo a seguir descreve todos os servidores Amazon DCV disponíveis. Os resultados são paginados para mostrar dois resultados por página.

```
from swagger_client.models.describe_servers_request_data import
 DescribeServersRequestData

def get_servers_api():
    api_instance =
 swagger_client.ServersApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def describe_servers(server_ids=None, next_token=None, max_results=None):
    request = DescribeServersRequestData(server_ids=server_ids,
 next_token=next_token, max_results=max_results)
    print('Describe Servers Request:', request)
    api_instance = get_servers_api()
    api_response = api_instance.describe_servers(body=request)
    print('Describe Servers Response', api_response)

def main():
```

```
    describe_servers(max_results=2)
```

Resposta

Esta é uma saída de exemplo.

```
{
    "RequestId": "request-id-123",
    "Servers": [
        {
            "Id": "ServerId123",
            "Ip": "1.1.1.123",
            "Hostname": "node001",
            "DefaultDnsName": "node001",
            "Endpoints": [
                {
                    "IpAddress": "x.x.x.x",
                    "Port": 8443,
                    "WebUrlPath": "/",
                    "Protocol": "HTTP"
                }
            ],
            "Version": "2021.0.10000",
            "SessionManagerAgentVersion": "2021.0.300",
            "Availability": "UNAVAILABLE",
            "UnavailabilityReason": "SERVER_FULL",
            "ConsoleSessionCount": 1,
            "VirtualSessionCount": 0,
            "Host": {
                "Os": {
                    "Family": "windows",
                    "Name": "Windows Server 2016 Datacenter",
                    "Version": "10.0.14393",
                    "BuildNumber": "14393"
                },
                "Memory": {
                    "TotalBytes": 8795672576,
                    "UsedBytes": 1743886336
                },
                "Swap": {
                    "TotalBytes": 0,
                    "UsedBytes": 0
                },
                "Aws": {
```

```
                    "Region": "us-west-2b",
                    "EC2InstanceType": "t2.large",
                    "EC2InstanceId": "i-123456789",
                    "EC2ImageId": "ami-12345678987654321"
                },
                "CpuInfo": {
                    "Vendor": "GenuineIntel",
                    "ModelName": "Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz",
                    "Architecture": "x86_64",
                    "NumberOfCpus": 2,
                    "PhysicalCoresPerCpu": 3
                },
                "CpuLoadAverage": {
                    "OneMinute": 0.04853546,
                    "FiveMinutes": 0.21060601,
                    "FifteenMinutes": 0.18792416
                },
                "Gpus": [],
                "LoggedInUsers": [
                    {
                        "Username": "Administrator"
                    }
                ]
            },
            "Tags": [
                {
                    "Key": "color",
                    "Value": "pink"
                },
                {
                    "Key": "dcv:os-family",
                    "Value": "windows"
                },
                {
                    "Key": "size",
                    "Value": "small"
                },
                {
                    "Key": "dcv:max-virtual-sessions",
                    "Value": "0"
                }
            ]
        },
        {
```

```
            "Id": "server-id-12456897",
            "Ip": "1.1.1.145",
            "Hostname": "node002",
            "DefaultDnsName": "node002",
            "Endpoints": [
                {
                    "IpAddress": "x.x.x.x",
                    "Port": 8443,
                    "WebUrlPath": "/",
                    "Protocol": "HTTP"
                },
                {
                    "IpAddress": "x.x.x.x",
                    "Port": 8443,
                    "Protocol": "QUIC"
                }
            ],
            "Version": "2021.0.10000",
            "SessionManagerAgentVersion": "2021.0.0",
            "Availability": "AVAILABLE",
            "ConsoleSessionCount": 0,
            "VirtualSessionCount": 5,
            "Host": {
                "Os": {
                    "Family": "linux",
                    "Name": "Amazon Linux",
                    "Version": "2",
                    "KernelVersion": "4.14.203-156.332.amzn2.x86_64"
                },
                "Memory": {
                    "TotalBytes": 32144048128,
                    "UsedBytes": 2184925184
                },
                "Swap": {
                    "TotalBytes": 0,
                    "UsedBytes": 0
                },
                "Aws": {
                    "Region": "us-west-2a",
                    "EC2InstanceType": "g3s.xlarge",
                    "EC2InstanceId": "i-123456789",
                    "EC2ImageId": "ami-12345678987654321"
                },
                "CpuInfo": {
```

```
                    "Vendor": "GenuineIntel",
                    "ModelName": "Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz",
                    "Architecture": "x86_64",
                    "NumberOfCpus": 4,
                    "PhysicalCoresPerCpu": 2
                },
                "CpuLoadAverage": {
                    "OneMinute": 2.24,
                    "FiveMinutes": 0.97,
                    "FifteenMinutes": 0.74
                },
                "Gpus": [
                    {
                        "Vendor": "NVIDIA Corporation",
                        "ModelName": "GM204GL [Tesla M60]"
                    }
                ],
                "LoggedInUsers": [
                    {
                        "Username" : "user45687"
                    },
                    {
                        "Username" : "user789"
                    }
                ]
            },
            "Tags": [
                {
                    "Key": "size",
                    "Value": "big"
                },
                {
                    "Key": "dcv:os-family",
                    "Value": "linux"
                },
                {
                    "Key": "dcv:max-virtual-sessions",
                    "Value": "10"
                },
                {
                    "Key": "color",
                    "Value": "blue"
                }
            ]
```

```
        }
    ]
}
```

DescribeSessions

Descreve uma ou mais sessões do Amazon DCV.

Tópicos

- Parâmetros de solicitação
- Parâmetros de resposta
- Exemplo

Parâmetros de solicitação

`SessionIds`

A IDs das sessões a serem descritas.

Tipo: String

Obrigatório: Não

`NextToken`

O token a ser usado para recuperar da próxima página de resultados.

Tipo: String

Obrigatório: Não

`Filters`

Filtros adicionais a serem aplicados à solicitação. Os filtros compatíveis incluem:

- tag:key — As tags atribuídas à sessão.
- owner — O proprietário da sessão.

Tipo: String

Obrigatório: Não

Parâmetros de resposta

`Id`

O ID exclusivo da sessão.

`Name`

O nome da sessão.

`Owner`

O proprietário da sessão.

`Server`

Informações sobre o servidor no qual a sessão está sendo executada. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

`Ip`

O endereço IP ou o nome do host do servidor Amazon DCV.

`Hostname`

O nome do host do host do servidor Amazon DCV.

`Port`

A porta pela qual o servidor Amazon DCV se comunica com os clientes do Amazon DCV.

`Endpoints`

Informações sobre os endpoints do servidor Amazon DCV. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

`IpAddress`

O endereço IP do endpoint do servidor.

`Port`

A porta do endpoint do servidor.

`Protocol`

O protocolo usado pelo endpoint do servidor. Os possíveis valores incluem:

- `HTTP`— O endpoint usa o protocolo WebSocket (TCP).
- `QUIC` — O endpoint usa o protocolo QUIC (UDP).

`WebUrlPath`

O caminho do URL da web do endpoint do servidor. Disponível somente para o protocolo HTTP.

`Tags`

As tags atribuídas ao servidor. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

`Key`

A chave de tags.

`Value`

O valor da tag.

`Type`

O tipo de sessão.

`State`

O estado atual da sessão. Os valores possíveis são:

- `CREATING` - o Agente 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 Agente e o Atendente podem não conseguir se comunicar.

`Substate`

O subestado atual da sessão. Os valores possíveis são:

- `SESSION_PLACING` - a sessão está esperando para ser colocada em um servidor DCV disponível.
- `PENDING_PREPARATION` - a sessão foi criada, mas não pode ser usada; vinculada a um servidor DCV.

`CreationTime`

A data e a hora em que a sessão foi criada.

`LastDisconnectionTime`

A data e a hora da última desconexão do cliente.

`NumOfConnections`

O número de conexões de cliente ativas.

`StorageRoot`

Especifica o caminho para a pasta a ser usada para armazenamento de sessão. Para obter mais informações sobre o armazenamento de sessão do Amazon DCV, consulte Ativar o armazenamento de sessão no Guia do administrador do Amazon DCV.

Tipo: String

Obrigatório: Não

Exemplo

Python

Solicitação

O exemplo a seguir descreve sessões que são de propriedade de `user1` e têm uma tag de `os=windows`.

```
from swagger_client.models.describe_sessions_request_data import
 DescribeSessionsRequestData
from swagger_client.models.key_value_pair import KeyValuePair
```

```
def get_sessions_api():
    api_instance =
 swagger_client.SessionsApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def describe_sessions(session_ids=None, next_token=None, tags=None, owner=None):
    filters = list()
    if tags:
        for tag in tags:
            filter_key_value_pair = KeyValuePair(key='tag:' + tag['Key'],
 value=tag['Value'])
            filters.append(filter_key_value_pair)
    if owner:
        filter_key_value_pair = KeyValuePair(key='owner', value=owner)
        filters.append(filter_key_value_pair)

    request = DescribeSessionsRequestData(session_ids=session_ids, filters=filters,
 next_token=next_token)
    print('Describe Sessions Request:', request)
    api_instance = get_sessions_api()
    api_response = api_instance.describe_sessions(body=request)
    print('Describe Sessions Response', api_response)

def main():
    describe_sessions(
        owner='user1',
        tags=[{'Key': 'os', 'Value': 'windows'}])
```

Resposta

Esta é uma saída de exemplo.

```
{
    "Sessions": [
        {
            "Id": "SessionId1897",
            "Name": "a session name",
            "Owner": "an owner 1890",
            "Server": {
                "Ip": "1.1.1.123",
                "Hostname": "server hostname",
                "Port": "1222",
                "Endpoints": [
```

```
                {
                    "IpAddress": "x.x.x.x",
                    "Port": 8443,
                    "WebUrlPath": "/",
                    "Protocol": "HTTP"
                },
                {
                    "IpAddress": "x.x.x.x",
                    "Port": 9443,
                    "WebUrlPath": "/",
                    "Protocol": "HTTP"
                },
                {
                    "IpAddress": "x.x.x.x",
                    "Port": 8443,
                    "WebUrlPath": "",
                    "Protocol": "QUIC"
                }
            ],
            "Tags": [
                {
                    "Key": "os",
                    "Value": "windows"
                },
                {
                    "Key": "ram",
                    "Value": "4gb"
                }
            ]
        },
        "Type": "VIRTUAL",
        "State": "READY",
        "CreationTime": "2020-10-06T10:15:31.633Z",
        "LastDisconnectionTime": "2020-10-06T10:15:31.633Z",
        "NumOfConnections": 2,
        "StorageRoot" : "/storage/root"
    },
    {
        "Id": "SessionId1895",
        "Name": "a session name",
        "Owner": "an owner 1890",
        "Server": {
            "Ip": "1.1.1.123",
            "Hostname": "server hostname",
```

```
                "Port": "1222",
                "Endpoints": [
                    {
                        "IpAddress": "x.x.x.x",
                        "Port": 8443,
                        "WebUrlPath": "/",
                        "Protocol": "HTTP"
                    },
                    {
                        "IpAddress": "x.x.x.x",
                        "Port": 9443,
                        "WebUrlPath": "/",
                        "Protocol": "HTTP"
                    },
                    {
                        "IpAddress": "x.x.x.x",
                        "Port": 8443,
                        "WebUrlPath": "",
                        "Protocol": "QUIC"
                    }
                ],
                "Tags": [
                    {
                        "Key": "os",
                        "Value": "windows"
                    },
                    {
                        "Key": "ram",
                        "Value": "4gb"
                    }
                ]
            },
            "Type": "VIRTUAL",
            "State": "DELETING",
            "CreationTime": "2020-10-06T10:15:31.633Z",
            "LastDisconnectionTime": "2020-10-06T10:15:31.633Z",
            "NumOfConnections": 2,
            "StorageRoot" : "/storage/root"
        }
    ]
}
```

DeleteSessions

Exclui a sessão especificada do Amazon DCV e a remove do cache do Agente.

Tópicos

- Parâmetros de solicitação
- Parâmetros de resposta
- Exemplo

Parâmetros de solicitação

`SessionId`

O ID da sessão a ser excluído.

Tipo: string

Obrigatório: Sim

`Owner`

O proprietário da sessão a ser excluído.

Tipo: string

Obrigatório: Sim

`Force`

Remove uma sessão do cache do agente ao tentar excluí-la do servidor Amazon DCV. Isso é útil para remover sessões desatualizadas do cache do Agente. Por exemplo, se um servidor Amazon DCV foi interrompido, mas as sessões ainda estiverem registradas no Agente, use esse sinalizador para limpar as sessões do cache do Agente.

Lembre-se de que, se a sessão ainda estiver ativa, ela será armazenada novamente em cache pelo Agente.

Valores válidos: `true` | `false`

Tipo: booliano

Obrigatório: não

Parâmetros de resposta

`SessionId`

O ID da sessão

`State`

Retornado somente se as sessões foram excluídas com sucesso. Indica o estado atual da sessão. Se a solicitação for concluída com êxito, a sessão passará para o estado `DELETING`. Pode demorar alguns minutos para que a sessão seja excluída. Quando ele é excluído, o estado muda de `DELETING` para `DELETED`.

`FailureReason`

Retornado somente se algumas sessões não puderem ser excluídas. Indica por que a sessão não pôde ser excluída.

Exemplo

Python

Solicitação

O exemplo a seguir exclui duas sessões: uma sessão com um ID `SessionId123` que pertence ao `user1` e uma sessão com um ID `SessionIdabc` que pertence ao `user99`.

```
from swagger_client.models.delete_session_request_data import
 DeleteSessionRequestData

def get_sessions_api():
    api_instance =
 swagger_client.SessionsApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def delete_sessions(sessions_to_delete, force=False):
    delete_sessions_request = list()
    for session_id, owner in sessions_to_delete:
        a_request = DeleteSessionRequestData(session_id=session_id, owner=owner,
 force=force)
        delete_sessions_request.append(a_request)
```

```
    print('Delete Sessions Request:', delete_sessions_request)
    api_instance = get_sessions_api()
    api_response = api_instance.delete_sessions(body=delete_sessions_request)
    print('Delete Sessions Response', api_response)

def main():
    delete_sessions([('SessionId123', 'an owner user1'), ('SessionIdabc',
 'user99')])
```

Resposta

A seguir está a saída de exemplo. O `SessionId123` foi excluído com sucesso, mas o `SessionIdabc` não pôde ser excluído.

```
{
    "RequestId": "10311636-df90-4cd1-bcf7-474e9675b7cd",
    "SuccessfulList": [
        {
            "SessionId": "SessionId123",
            "State": "DELETING"
        }
    ],
    "UnsuccessfulList": [
        {
            "SessionId": "SessionIdabc",
            "FailureReason": "The requested dcvSession does not exist"
        }
    ]
}
```

GetSessionConnectionData

Obtém informações de conexão para a conexão de um usuário específico com uma sessão especificada do Amazon DCV.

Tópicos

- Parâmetros de solicitação
- Parâmetros de resposta
- Informações adicionais
- Exemplo

Parâmetros de solicitação

`SessionId`

O ID da sessão para a qual exibir as informações de conexão.

Tipo: string

Obrigatório: Sim

`User`

O nome do usuário para o qual exibir as informações de conexão.

Tipo: string

Obrigatório: Sim

Parâmetros de resposta

`Id`

O ID exclusivo da sessão.

`Name`

O nome da sessão.

`Owner`

O proprietário da sessão.

`Server`

Informações sobre o servidor no qual a sessão está sendo executada. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

`Ip`

O endereço IP ou o nome do host do servidor Amazon DCV.

Hostname

O nome do host do host do servidor Amazon DCV.

Port

A porta pela qual o servidor Amazon DCV se comunica com os clientes do Amazon DCV.

Endpoints

Informações sobre os endpoints do servidor Amazon DCV. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

IpAddress

O endereço IP do endpoint do servidor.

Port

A porta do endpoint do servidor.

Protocol

O protocolo usado pelo endpoint do servidor. Os possíveis valores incluem:

- `HTTP`— O endpoint usa o protocolo WebSocket (TCP).
- `QUIC` — O endpoint usa o protocolo QUIC (UDP).

WebUrlPath

O caminho do URL da web do endpoint do servidor. Disponível somente para o protocolo HTTP.

WebUrlPath

O caminho para o arquivo de configuração do servidor Amazon DCV.

Tags

As tags atribuídas ao servidor. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

Key

A chave de tags.

Value

O valor da tag.

`Type`

O tipo de sessão.

`State`

O estado atual da sessão. Os valores possíveis são:

- `CREATING` - o Agente 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 Agente e o Atendente podem não conseguir se comunicar.

`CreationTime`

A data e a hora em que a sessão foi criada.

`LastDisconnectionTime`

A data e a hora da última desconexão do cliente.

`NumOfConnections`

O número de conexões simultâneas que o usuário tem com a sessão.

`ConnectionToken`

O token de autenticação usado para conectar a sessão.

Informações adicionais

As informações obtidas dessa API podem ser transmitidas para um cliente do Amazon DCV para se conectar à sessão do Amazon DCV.

No caso do cliente da Web do Amazon DCV, você pode criar um URL que possa ser aberto no navegador. O URL tem o seguinte formato:

```
https://{Ip}:{Port}{WebUrlPath}?authToken={ConnectionToken}#{SessionId}.
```

No caso do cliente nativo do Amazon DCV, você pode criar um URL com o esquema `dcv://`. Quando o cliente nativo Amazon DCV é instalado, ele se registra no sistema como manipulador do. `dcv://` URLs O URL tem o seguinte formato:

```
dcv://{Ip}:{Port}{WebUrlPath}?authToken={ConnectionToken}#{SessionId}.
```

> **Note**
>
> Se você estiver usando a Amazon EC2, o endereço IP deve ser público. Se sua configuração tiver hosts Amazon DCV atrás de um gateway, especifique o endereço do gateway em vez daquele retornado pela SessionConnectionData API.

Exemplo

Python

Solicitação

O exemplo a seguir obtém informações de conexão de um usuário com nome de usuário `user1` e sessão com ID `sessionId12345`.

```
def get_session_connection_api():
    api_instance =
 swagger_client.GetSessionConnectionDataApi(swagger_client.ApiClient(get_client_configuratio
    set_request_headers(api_instance.api_client)
    return api_instance


def get_url_to_connect(api_response):
    ip_address = api_response.session.server.ip
    port = api_response.session.server.port
    web_url_path = api_response.session.server.web_url_path
    connection_token = api_response.connection_token
    session_id = api_response.session.id
    url = f'https://{ip_address}:{port}{web_url_path}?
authToken={connection_token}#{session_id}'
    return url
```

```
def get_session_connection_data(session_id, user):
    api_response =
 get_session_connection_api().get_session_connection_data(session_id=session_id,
 user=user)
    url_to_connect = get_url_to_connect(api_response)
    print('Get Session Connection Data Response:', api_response)
    print('URL to connect: ', url_to_connect)


def main():
    get_session_connection_data('sessionId12345', 'user1')
```

Resposta

Esta é uma saída de exemplo.

```
{
    "Session": {
        "Id": "sessionId12345",
        "Name": "a session name",
        "Owner": "an owner 1890",
        "Server": {
            "Ip": "1.1.1.123",
            "Hostname": "server hostname",
            "Port": "1222",
            "endpoints": [
                {
                    "port": 8443,
                    "web_url_path": "/",
                    "protocol": "HTTP"
                },
                {
                    "port": 9443,
                    "web_url_path": "/",
                    "protocol": "HTTP"
                },
                {
                    "port": 8443,
                    "web_url_path": "",
                    "protocol": "QUIC"
                }
            ],
```

```
            "WebUrlPath": "/path",
            "Tags": [
                {
                    "Key": "os",
                    "Value": "windows"
                },
                {
                    "Key": "ram",
                    "Value": "4gb"
                }
            ]
        },
        "Type": "VIRTUAL",
        "State": "UNKNOWN",
        "CreationTime": "2020-10-06T10:15:31.633Z",
        "LastDisconnectionTime": "2020-10-06T10:15:31.633Z",
        "NumOfConnections": 2
    },
    "ConnectionToken":
 "EXAMPLEiOiJmOWM1YTRhZi1jZmU0LTQ0ZjEtYjZlOC04ZjY0YjM4ZTE2ZDkiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUz
tngiKXevUxhhJvm3BPJYRs9NPE4GCJRTc13EXAMPLEIxNEPPh5IMcVmROfU1WKPnry4ypPTp3rsZ7YWjCTSfs1GoN3R_
Kqtpd5GH0D-E8FwsedV-
Q2bRQ4y9y1q0MgFU4QjaSMypUuYR0YjkCaoainjmEZew4A33fG40wATrBvoivBiNWdNpytHX2CDOuk_k0k_DWeZjMvv9
h_GaMgHmltqBIA4jdPD7i0CmC2e7413KFy-
EQ4Ej1cM7RjLwhFuWpKWAVJxogJjYpfoKKaPo4KxvJjJIPYhksclINQpe2W5rnlxCq7sC7ptcGw17DUobP7egRv9H37
hK1G4G8erHvl9HIrTR9_c884fNrTCC8DvC062e4KYdLkAhhJmboN9CAGIGFyd2c1AY_CzzvDL0EXAMLE"
}
```


GetSessionScreenshots

Obtém capturas de tela de uma ou mais sessões do Amazon DCV.

Para modificar o formato da imagem, configure o `session-screenshot-format` parâmetro na configuração do Session Manager Broker. Consulte o arquivo de configuração do agente no Guia do administrador do Amazon DCV Session Manager.

Quando os `MaxHeight` parâmetros `MaxWidth` ou da `GetSessionScreenshots` solicitação não forem especificados, os `session-screenshot-max-height` valores `session-screenshot-max-width` e definidos no arquivo de configuração do Session Manager Broker serão usados. Para modificar esses parâmetros, consulte também o arquivo de configuração do agente no Guia do administrador do Amazon DCV Session Manager.

O valor superior da resolução da captura de tela é limitado à resolução da sessão remota. Se os `MaxHeight` parâmetros `MaxWidth` e forem definidos para valores maiores do que a resolução da sessão remota atual, a captura de tela resultante será limitada à resolução real da sessão.

> **Note**
>
> Para modificar esses valores no Access Console, consulte o arquivo de configuração do Web Client no Guia do administrador do console de acesso Amazon DCV. Para modificar esses valores com a CLI do Session Manager, consulte `get-session-screenshots` o Guia da CLI do Amazon DCV.

Tópicos

- Parâmetros de solicitação
- Parâmetros de resposta
- Exemplo

Parâmetros de solicitação

`SessionId`

O ID da sessão do Amazon DCV da qual obter a captura de tela.

Tipo: string

Obrigatório: Sim

`MaxWidth`

A largura máxima, em pixels, das capturas de tela da sessão. Se não for especificado, os valores da configuração do Session Manager Broker serão aplicados. Se fornecido, deve ser um número maior que 0.

Tipo: número inteiro

Obrigatório: não

`MaxHeight`

A altura máxima, em pixels, das capturas de tela da sessão. Se não for especificado, os valores da configuração do Session Manager Broker serão aplicados. Se fornecido, deve ser um número maior que 0.

Tipo: inteiro

Obrigatório: sim

Parâmetros de resposta

`RequestId`

O ID exclusivo da solicitação.

`SuccessfulList`

Informações sobre as capturas de tela bem-sucedidas. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

`SessionScreenshot`

Informações sobre as capturas de tela. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

`SessionId`

O ID da sessão do Amazon DCV da qual a captura de tela foi tirada.

`Images`

Informações sobre as imagens. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

`Format`

O formato da imagem. Os valores possíveis incluem: `jpeg` e `png`.

`Data`

O formato codificado em base64 da imagem da captura de tela.

`CreationTime`

A data e a hora em que a captura de tela foi tirada.

Primary

Indica se a captura de tela é da tela principal da sessão do Amazon DCV.

UnsuccessfulList

Informações sobre as capturas de tela malsucedidas. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

GetSesionScreenshotRequestData

A solicitação original que falhou.

SessionId

O ID da sessão do Amazon DCV da qual a captura de tela foi tirada.

FailureReason

O motivo da falha.

GetSessionScreenshotRequestData

A solicitação original que falhou.

Exemplo

Python

Solicitação

O exemplo a seguir obtém capturas de tela de duas sessões (`sessionId1esessionId2`) com a largura máxima definida em 800 e a altura máxima definida em 600. A sessão `sessionId2` não existe e resulta em uma falha.

```
from swagger_client.models.describe_servers_request_data import
 GetSessionScreenshotRequestData

def get_sessions_api():
    api_instance =
 swagger_client.ServersApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance
```

```
def get_session_screenshots(session_ids, max_width=None, max_height=None):
    request = [GetSessionScreenshotRequestData(session_id=session_id,
 max_width=max_width, max_height=max_height) for session_id in session_ids]
    print('Get Session Screenshots Request:', request)
    api_instance = get_sessions_api()
    api_response = api_instance.get_session_screenshots(body=request)
    print('Get Session Screenshots Response:', api_response)

def main():
    get_session_screenshots(["sessionId1", "sessionId2"]), 800, 600)
```

Resposta

Esta é uma saída de exemplo.


```
{
    "RequestId": "542735ef-f6ab-47d8-90e5-23df31d8d166",
    "SuccessfulList": [
        {
            "SessionScreenshot": {
                "SessionId": "sessionId1",
                "Images": [
                    {
                        "Format": "png",
                        "Data": "iVBORw0KGgoAAAANSUhEUgAAAEXAMPLE",
                        "CreationTime": "2021-03-30T15:47:06.822Z",
                        "Primary": true
                    }
                ]
            }
        }
    ],
    "UnsuccessfulList": [
        {
            "GetSessionScreenshotRequestData": {
                "SessionId": "sessionId2"
            },
            "FailureReason": "Dcv session not found."
        }
    ]
}
```

OpenServers

Abre um ou mais servidores Amazon DCV. Antes de criar sessões do Amazon DCV em um servidor Amazon DCV, você deve alterar o estado do servidor para aberto. Depois que o servidor Amazon DCV estiver aberto, você poderá criar sessões do Amazon DCV no servidor.

Tópicos

- Parâmetros de solicitação
- Parâmetros de resposta
- Exemplo

Parâmetros de solicitação

`ServerId`

O ID do servidor a ser aberto.

Tipo: string

Obrigatório: Sim

Parâmetros de resposta

`RequestId`

O ID exclusivo da solicitação.

`SuccessfulList`

Informações sobre os servidores Amazon DCV que foram abertos com sucesso. Essa estrutura de dados inclui o seguinte parâmetro de resposta aninhada:

`ServerId`

O ID do servidor que foi aberto com sucesso.

`UnsuccessfulList`

Informações sobre os servidores Amazon DCV que não puderam ser abertos. Essa estrutura de dados inclui os seguintes parâmetros de resposta aninhada:

`OpenServerRequestData`

Informações sobre a solicitação original com falha. Essa estrutura de dados inclui o seguinte parâmetro de resposta aninhada:

`ServerId`

O ID do servidor Amazon DCV que não pôde ser aberto.

`FailureCode`

O código da falha.

`FailureReason`

O motivo da falha.

Exemplo

Python

Solicitação

O exemplo a seguir abre dois servidores Amazon DCV (`serverId1` e `serverId2`).

```
from swagger_client.models import OpenServerRequestData


def get_servers_api():
    api_instance =
 swagger_client.ServersApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def open_servers(server_ids):
    request = [OpenServerRequestData(server_id=server_id) for server_id in
 server_ids]
    print('Open Servers Request:', request)
    api_instance = get_servers_api()
    api_response = api_instance.open_servers(body=request)
    print('Open Servers Response:', api_response)

def main():
    open_servers(["serverId1", "serverId2"])
```

Resposta

Esta é uma saída de exemplo.

```
{
    "RequestId": "1e64830f-0a27-41bf-8147-0f3411791b64",
    "SuccessfulList": [
        {
            "ServerId": "serverId1"
        }
    ],
    "UnsuccessfulList": [
        {
            "OpenServerRequestData": {
                "ServerId": "serverId2"
            },
            "FailureCode": "DCV_SERVER_NOT_FOUND",
            "FailureReason": "Dcv server not found."
        }
    ]
}
```

UpdateSessionPermissions

Atualiza as permissões do usuário para uma sessão especificada do Amazon DCV.

Tópicos

- Parâmetros de solicitação
- Parâmetros de resposta
- Exemplo

Parâmetros de solicitação

`SessionId`

O ID da sessão para o qual as permissões devem ser alteradas.

Tipo: string

Obrigatório: Sim

`Owner`

O proprietário da sessão para o qual as permissões devem ser alteradas.

Tipo: string

Obrigatório: Sim

`PermissionFile`

O conteúdo codificado em Base64 do arquivo de permissões a ser usado. Para obter mais informações, consulte Configurar a autorização do Amazon DCV no Guia do administrador do Amazon DCV.

Tipo: string

Obrigatório: Sim

Parâmetros de resposta

`SessionId`

O ID da sessão.

Exemplo

Python

Solicitação

O exemplo a seguir define as novas permissões para uma sessão com um ID de sessão de `SessionId1897`.

```
from swagger_client.models.update_session_permissions_request_data import
 UpdateSessionPermissionsRequestData

def get_session_permissions_api():
    api_instance =
 swagger_client.SessionPermissionsApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instancedef
 update_session_permissions(session_permissions_to_update):
```

```
    update_session_permissions_request = list()
    for session_id, owner, permissions_base64_encoded in
 session_permissions_to_update:
        a_request = UpdateSessionPermissionsRequestData(
            session_id=session_id, owner=owner,
 permissions_file=permissions_base64_encoded)
        update_session_permissions_request.append(a_request)
    print('Update Session Permissions Request:', update_session_permissions_request)
    api_instance = get_session_permissions_api()
    api_response =
 api_instance.update_session_permissions(body=update_session_permissions_request)
    print('Update Session Permissions Response:', api_response)

def main():
    update_session_permissions([('SessionId1897', 'an owner 1890',
 'file_base64_encoded')])
```

Resposta

Esta é uma saída de exemplo.

```
{
   'request_id': 'd68ebf66-4022-42b5-ba65-99f89b18c341',
    'successful_list': [
        {'
            session_id': 'SessionId1897'
        }
    ],
    'unsuccessful_list': []
}
```

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

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

Tópicos

- Notas de versão do Gerenciador de Sessões do Amazon DCV
- Histórico de documentos

Notas de versão do Gerenciador de Sessões do Amazon DCV

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

Tópicos

- 2024.0-531 — 17 de junho de 2025
- 2024.0-504— 31 de março de 2025
- 2024.0-493 — 15 de janeiro de 2025
- 2024.0-457: 1º de outubro de 2024
- 2023.1-17652: 1º de agosto de 2024
- 2023.1-16388: 26 de junho de 2024
- 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

2024.0-531 — 17 de junho de 2025

Números de compilação	Alterações e correções de bugs
• Corretor: 531 • Agente: 852 • CLI: 154	• Recurso adicionado para renovar certificados antes da expiração. • NICE DCV foi renomeado como Amazon DCV. • Correções de erros.

2024.0-504— 31 de março de 2025

Números de compilação	Alterações e correções de bugs
• Corretor: 504 • Agente: 817 • CLI: 154	• Foi adicionado suporte para AL2 023. • Correções de erros e melhorias na performance.

2024.0-493 — 15 de janeiro de 2025

Números de compilação	Alterações e correções de bugs
• Corretor: 493 • Agente: 801	• Parâmetros adicionados à `GetSessionScreenshot` solicitação para especificar a altura e a largura máximas da captura de tela.

Números de compilação	Alterações e correções de bugs
• CLI: 152	• Foi adicionado um parâmetro ao arquivo de configuração do Broker que especifica o número de segundos após os quais as sessões em um servidor Amazon DCV inacessível são excluídas do sistema. • Corrigido um problema em que o `seconds-before-deleting-unreachable-dcv-server` parâmetro no arquivo de configura ção do Broker não estava sendo respeitado. • Correções de erros e melhorias na performance.

2024.0-457: 1º de outubro de 2024

Números de compilação	Alterações e correções de bugs
• Agente: 457 • Atendente: 748 • CLI: 140	• NICE DCV foi renomeado como Amazon DCV. • Foi adicionado suporte para Ubuntu 24.04.

2023.1-17652: 1º de agosto de 2024

Números de compilação	Alterações e correções de bugs
• Agente: 426 • Atendente: 748 • CLI: 140	• Correções de erros e melhorias na performance.

2023.1-16388: 26 de junho de 2024

Números de compilação	Alterações e correções de bugs
• Agente: 417 • Atendente: 748 • CLI: 140	• Corrigido um bug que mostrava a memória incorretamente como TB, não GB. • Correções de erros e melhorias na performance.

2023.1 — 9 de novembro de 2023

Números de compilação	Alterações e correções de bugs
• Agente: 410 • Atendente: 732 • CLI: 140	• Correções de erros e melhorias no desempenho

2023.0-15065 — 4 de maio de 2023

Números de compilação	Alterações e correções de bugs
• Agente: 392 • Atendente: 675 • CLI: 132	• 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
• Agente: 392 • Atendente: 642 • CLI: 132	• 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
• Agente: 382 • Atendente: 612 • CLI: 123	• Foi adicionado um campo `Substate` na resposta `DescribeSessions` . • 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
• Agente: 355 • Atendente: 592 • CLI: 114	• Foi adicionado suporte para executar o broker em instâncias do AWS Graviton. • 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
• Agente: 341 • Atendente: 520 • CLI: 112	• Foi adicionada a capacidade de rotação de log ao Atendente. • Parâmetro de configuração adicionado para definir a página inicial do Java no Agente. • Melhoria na descarga de dados do cache para o disco no Agente. • Validação de URL corrigida na CLI.

2021.3-11591 — 20 de dezembro de 2021

Números de compilação	Novos atributos
• Agente: 307 • Atendente: 453 • CLI: 92	• Suporte adicionado para integração com o Gateway de Conexão do Amazon DCV. • Suporte adicionado para Agente para Ubuntu 18.04 e Ubuntu 20.04.

2021.2-11445 — 18 de novembro de 2021

Números de compilação	Alterações e correções de bugs
• Agente: 288 • Atendente: 413 • CLI: 54	• 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
• Agente: 254 • Atendente: 413 • CLI: 54	• 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 atributos	Alterações e correções de bugs
• Agente: 25 • Atendente: 413 • CLI: 37	• O Gerenciador de Sessões do Amazon DCV agora oferece suporte à interface de linha de comandos (CLI). Você pode criar e gerenciar sessões do Amazon DCV na CLI, em vez de ligar. APIs • O Gerenciador de Sessões do Amazon DCV introduziu a persistência de dados do Agente. 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.	• 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 atributos	Alterações e correções de bugs
• Agente: 21 • Atendente: 365	• O Gerenciador de Sessões do Amazon DCV adicionou suporte para parâmetros de entrada transmitidos para o arquivo autorun no Linux. • Agora, as propriedades do servidor podem ser passadas como requisitos para a CreateSessionsAPI.	• Corrigimos um problema no arquivo autorun no Windows.

2021.0-10242 — 12 de abril de 2021

Números de compilação	Alterações e correções de bugs

• Agente: 183
• Atendente: 318
	• O Amazon DCV Session Manager apresentou as seguintes novidades APIs:
	  • OpenServers
	  • CloseServers
	  • DescribeServers
	  • GetSessionScreenshots
	• Ele também introduziu os seguintes novos parâmetros de configura ção:
	  • Parâmetros do broker: `session-screenshot-max-widt h` , `session-screenshot-max-height` , `session-s creenshot-format` , `create-sessions-queue-max-s ize` e `create-sessions-queue-max-time-seconds` .
	  • Parâmetros do Atendente: `agent.autorun_folder` , `max_virtual_sessions` e `max_concurrent_ses sions_per_user` .

Números de compilação	Alterações e correções de bugs
	Parâmetros do Atendente: `agent.autorun_folder`, `max_virtual_sessions` e `max_concurrent_sessions_per_user`. Parâmetros do Atendente: `agent.autorun_folder`, `max_virtual_sessions` e `max_concurrent_sessions_per_user`.

2020.2-9662 — 4 de dezembro de 2020

Números de compilação	Alterações e correções de bugs
• Agente: 114 • Atendente: 211	• Corrigimos um problema com os certificados TLS gerados automaticamente que impediam o Agente de iniciar.

2020.2-9508 — 11 de novembro de 2020

Números de compilação	Alterações e correções de bugs
• Agente: 78 • Atendente: 183	• Versão inicial do Gerenciador de Sessões do Amazon DCV.

Histórico de documentos

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

Alteração	Descrição	Data
Amazon DCV versão 2024.0-531	O Amazon DCV Session Manager foi atualizado para o Amazon DCV 2024.0-531. Para obter mais informações, consulte ???.	17 de junho de 2025
Amazon DCV versão 2024.0-504	O Amazon DCV Session Manager foi atualizado para o Amazon DCV 2024.0-504. Para obter mais informações, consulte ???.	31 de março de 2025
Amazon DCV versão 2024.0-493	O Amazon DCV Session Manager foi atualizado para o Amazon DCV 2024.0-493. Para obter mais informações, consulte 2024.0-493 — 15 de janeiro de 2025.	15 de janeiro de 2025
Amazon DCV versão 2024.0-457	O Gerenciador de Sessões do Amazon DCV foi atualizado para o Amazon DCV 2024.0-457. Para obter mais informações, consulte 2024.0-457: 1º de outubro de 2024.	30 de setembro de 2024
Amazon DCV versão 2023.1-17652	O Gerenciador de Sessões do Amazon DCV foi atualizado para o Amazon DCV 2023.1-17652. Para obter mais informações, consulte 2023.1-17652: 1º de agosto de 2024.	1.º de agosto de 2024
Amazon DCV versão 2023.1-16388	O Gerenciador de Sessões do Amazon DCV foi atualizado para o Amazon DCV 2023.1-16388. Para obter mais informações, consulte 2023.1-16388: 26 de junho de 2024.	26 de junho de 2024
Amazon DCV	O Gerenciador de Sessões do Amazon DCV foi atualizado para o Amazon DCV	9 de novembro de 2023

Alteração	Descrição	Data
versão 2023.1	2023.1. Para obter mais informações, consulte 2023.1 — 9 de novembro de 2023.	
Amazon DCV versão 2023.0	O Gerenciador de Sessões do Amazon DCV foi atualizado para o Amazon DCV 2023.0. Para obter mais informações, consulte 2023.0-14852 — 28 de março de 2023.	28 de março de 2023
Amazon DCV versão 2022.2	O Gerenciador de Sessões do Amazon DCV foi atualizado para o Amazon DCV 2022.2. Para obter mais informaçõ es, consulte 2022.2-13907 — 11 de novembro de 2022.	11 de novembro de 2022
Amazon DCV versão 2022.1	O Gerenciador de Sessões do Amazon DCV foi atualizado para o Amazon DCV 2022.1. Para obter mais informações, consulte 2022.1-13067 — 29 de junho de 2022.	29 de junho de 2022
Amazon DCV versão 2022.0	O Gerenciador de Sessões do Amazon DCV foi atualizado para o Amazon DCV 2022.0. Para obter mais informaçõ es, consulte 2022.0-11952 — 23 de fevereiro de 2022.	23 de fevereiro de 2022
Amazon DCV versão 2021.3	O Gerenciador de Sessões do Amazon DCV foi atualizado para o Amazon DCV 2021.3. Para obter mais informaçõ es, consulte 2021.3-11591 — 20 de dezembro de 2021.	20 de dezembro de 2021

Alteração	Descrição	Data
Amazon DCV versão 2021.2	O Gerenciador de Sessões do Amazon DCV foi atualizado para o Amazon DCV 2021.2. Para obter mais informaçõ es, consulte 2021.2-11042 — 1º de setembro de 2021.	1º de setembro de 2021
Amazon DCV versão 2021.1	O Gerenciador de Sessões do Amazon DCV foi atualizado para o Amazon DCV 2021.1. Para obter mais informações, consulte 2021.1-10557 — 31 de maio de 2021.	31 de maio de 2021
Amazon DCV versão 2021.0	O Gerenciador de Sessões do Amazon DCV foi atualizado para o Amazon DCV 2021.0. Para obter mais informações, consulte 2021.0-10242 — 12 de abril de 2021.	12 de abril de 2021
Versão inicial do Gerenciad or de Sessões do Amazon 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.