
Administrador NICE DCV del Administrador del programa

Guía para desarrolladores



Administrador NICE DCV del Administrador del programa: Guía para desarrolladores

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas comerciales que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es Session Manager?	1
Cómo funciona el Administrador de sesiones	1
Características	2
Introducción	4
Generar la API del cliente	4
Registre su cliente de API	5
Obtenga un token de acceso y realice una solicitud de API	5
Referencia de la API del Administrador de sesiones	8
CloseServers	8
Parámetros de solicitud	5
Parámetros de respuesta	9
Ejemplo	9
CreateSessions	10
Parámetros de solicitud	5
Parámetros de respuesta	9
Ejemplo	9
DescribeServers	15
Parámetros de solicitud	5
Parámetros de respuesta	9
Ejemplo	9
DescribeSessions	23
Parámetros de solicitud	5
Parámetros de respuesta	9
Ejemplo	9
DeleteSessions	28
Parámetros de solicitud	5
Parámetros de respuesta	9
Ejemplo	9
GetSessionConnectionData	30
Parámetros de solicitud	5
Parámetros de respuesta	9
Información adicional	32
Ejemplo	9
GetSessionScreenshots	34
Parámetros de solicitud	5
Parámetros de respuesta	9
Ejemplo	9
OpenServers	36
Parámetros de solicitud	5
Parámetros de respuesta	9
Ejemplo	9
UpdateSessionPermissions	38
Parámetros de solicitud	5
Parámetros de respuesta	9
Ejemplo	9
Notas de la versión e historial de revisión	40
Notas de la versión	40
2023.0-14852 — 28 de marzo de 2023	40
2022.2-13907 — 11 de noviembre de 2022	41
2022.1-13067 — 29 de junio de 2022	41
2022.0-11952 — 23 de febrero de 2022	41
2021.3-11591 — 20 de diciembre de 2021	41
2021.2-11445 — 18 de noviembre de 2021	42
2021.2-11190 — 11 de octubre de 2021	42

2021.2-11042 — 01 de septiembre de 2021	42
2021.1-10557 — 31 de mayo de 2021	42
2021.0-10242 — 12 de abril de 2021	43
2020.2-9662— 4 de diciembre de 2020	43
.....	43
Historial de documentos	43
.....	xlv

¿Qué es NICE DCV Session Manager?

El administrador de sesiones de NICE DCV es un conjunto de paquetes de software instalables (un agente y un agente) y una interfaz de programación de aplicaciones (API) que facilita a los desarrolladores y proveedores de software independientes (ISV) la creación de aplicaciones de interfaz que creen y gestionen mediante programación el ciclo de vida de las sesiones de NICE DCV en una flota de servidores NICE DCV.

Esta guía explica cómo utilizar las API de Session Manager para administrar el ciclo de vida de las sesiones de NICE DCV. Para obtener más información sobre cómo instalar y configurar el agente y los agentes de Session Manager, consulte la Guía del administrador de sesiones de NICE DCV.

Requisitos previos

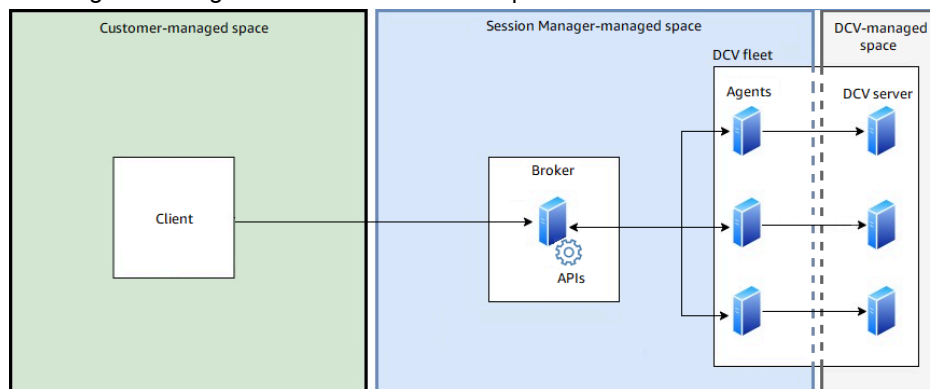
Antes de empezar a trabajar con las API de Session Manager, asegúrese de estar familiarizado con las sesiones NICE DCV y NICE DCV. Para obtener más información, consulte la [Guía del administrador de NICE DCV](#).

Temas

- [Cómo funciona el Administrador de sesiones \(p. 1\)](#)
- [Características \(p. 2\)](#)

Cómo funciona el Administrador de sesiones

En el siguiente diagrama se muestran los componentes de alto nivel de Session Manager.



Agente

El Broker es un servidor web que aloja y expone las API del administrador de sesiones. Recibe y procesa las solicitudes de API para gestionar las sesiones de NICE DCV del cliente y, a continuación, pasa las instrucciones a los agentes pertinentes. El Broker debe estar instalado en un host que esté separado de los servidores NICE DCV, pero debe ser accesible para el cliente y debe poder acceder a los agentes.

Agente

El agente se instala en cada servidor NICE DCV de la flota. Los agentes reciben instrucciones del corredor y las ejecutan en sus respectivos servidores NICE DCV. Los agentes también supervisan el estado de los servidores de NICE DCV y envían actualizaciones de estado periódicas al corredor.

API

El administrador de sesiones expone un conjunto de interfaces de programación de aplicaciones (API) REST que se pueden utilizar para administrar las sesiones de NICE DCV en una flota de servidores NICE DCV. Las API están alojadas y expuestas por el Broker. Los desarrolladores pueden crear clientes de administración de sesiones personalizados que llamen a las API.

Cliente

El cliente es la aplicación frontal o el portal que se desarrolla para llamar a las API del administrador de sesiones expuestas por el corredor. Los usuarios finales utilizan el cliente para administrar las sesiones alojadas en los servidores NICE DCV de la flota.

Token de acceso

Para realizar una solicitud de API, debes proporcionar un token de acceso. Las API de clientes registradas pueden solicitar los tokens al Broker o a un servidor de autorización externo. Para solicitar un token de acceso, la API del cliente debe proporcionar credenciales válidas.

API de cliente

La API del cliente se genera a partir del archivo YAML de definición de la API de Session Manager, mediante Swagger Codegen. La API del cliente se usa para realizar solicitudes de API.

Sesión de NICE DCV

Debe crear una sesión de NICE DCV en su servidor NICE DCV a la que puedan conectarse sus clientes. Los clientes solo pueden conectarse a un servidor NICE DCV si hay una sesión activa. NICE DCV admite sesiones virtuales y de consola. Las API de Session Manager se utilizan para administrar el ciclo de vida de las sesiones de NICE DCV. Las sesiones de NICE DCV pueden tener uno de los siguientes estados:

- CREATING—El bróker está en proceso de crear la sesión.
- READY—la sesión está lista para aceptar conexiones de clientes.
- DELETING: se está eliminando la sesión.
- DELETED—se ha eliminado la sesión.
- UNKNOWN—no se puede determinar el estado de la sesión. Es posible que el corredor y el agente no puedan comunicarse.

Características

El Administrador de sesiones de DCV ofrece las siguientes características:

- Proporciona información sobre las sesiones de NICE DCV: obtenga información sobre las sesiones que se ejecutan en varios servidores de NICE DCV.
- Gestione el ciclo de vida de varias sesiones de NICE DCV: cree o elimine varias sesiones para varios usuarios en varios servidores de NICE DCV con una solicitud de API.
- Admite etiquetas: utilice etiquetas personalizadas para dirigirse a un grupo de servidores NICE DCV al crear sesiones.
- Administra los permisos de varias sesiones de NICE DCV: modifique los permisos de usuario para varias sesiones con una solicitud de API.

- Proporciona información de conexión: recupera la información de conexión del cliente para las sesiones de NICE DCV.
- Compatible con servidores locales y en la nube: utilice Session ManagerAWS in, in situ o con servidores alternativos basados en la nube.

Introducción

En esta sección se explica cómo empezar a utilizar las API del administrador de sesiones.

En esta sección, le mostraremos cómo hacerlo utilizando la `DescribeSessions` API como ejemplo.

Temas

- [Generar la API del cliente \(p. 4\)](#)
- [Registre su cliente de API \(p. 5\)](#)
- [Obtenga un token de acceso y realice una solicitud de API \(p. 5\)](#)

Generar la API del cliente

Las API de Session Manager se definen en un único archivo YAML. Las API se basan en la especificación OpenAPI3.0, que define una interfaz estándar independiente del idioma para las API RESTful. Para obtener más información, consulte [Especificación de OpenAPI](#).

Con el archivo YAML, puede generar clientes de API en uno de los idiomas admitidos. Para ello, debe usar Swagger Codegen 3.0 o superior. Para obtener más información acerca de los idiomas compatibles, consulte el [repositorio swagger-codegen](#).

Para generar el cliente de API

1. Descargue el archivo YAML de la API de Session Manager desde el Session Manager Broker. El archivo YAML está disponible en la siguiente URL.

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

2. Instale Swagger Codegen.

- macOS

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

- Otras plataformas

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

```
$ cd swagger-codegen
```

3. Genstre el cliente de API.

- macOS

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

- Otras 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
```

Registre su cliente de API

Para realizar una solicitud de API, primero debe recuperar un token de acceso del Broker. Para obtener un token de acceso del corredor, debe proporcionarle las credenciales de la API de su cliente. Las credenciales se basan en un identificador de cliente y una contraseña de cliente que se generan cuando su cliente se registra en el Broker. Si no tiene un identificador de cliente ni una contraseña de cliente para su cliente, debe solicitarlos al administrador de su bróker. Para obtener más información sobre cómo registrar la API de su cliente en el Broker y obtener un ID de cliente y una contraseña, consulte [register-api-client](#).

Obtenga un token de acceso y realice una solicitud de API

Primero importamos los modelos necesarios para la aplicación.

A continuación, declaramos las variables para el ID del cliente (`__CLIENT_ID`), la contraseña del cliente (`__CLIENT_SECRET`) y la URL del bróker, incluido el número de puerto (`__PROTOCOL_HOST_PORT`).

A continuación, creamos una función llamada `build_client_credentials` que genera las credenciales del cliente. Para generar las credenciales del cliente, primero debe concatenar el ID del cliente y la contraseña del cliente y separar los valores con dos puntos (`client_id:client_password`) y, a continuación, codificar en Base64 toda la cadena.

```
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 = 'MzcxNzJhN2UtYjEzNS00MjNjLTg2N2YtMjF1ZmRlZWVjMDU1'  
__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')
```

Ahora que tenemos nuestras credenciales de cliente, podemos usarlas para solicitar un token de acceso al Broker. Para ello, creamos una función llamada `get_access_token`. Debe llamar a `POST` on `https://Broker_IP:8443/oauth2/token?grant_type=client_credentials` y proporcionar un encabezado de autorización, que incluya las credenciales del cliente codificadas en Basic y un tipo de contenido 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
```

Ahora, creamos las funciones necesarias para crear una instancia de la API de un cliente. Para crear una instancia de una API de cliente, debe especificar la configuración del cliente y los encabezados que se utilizarán en las solicitudes. La `get_client_configuration` función crea un objeto de configuración que incluye la dirección IP y el puerto del Broker y la ruta al certificado autofirmado del Broker, que debería haber recibido del administrador del Broker. La `set_request_headers` función crea un objeto de encabezado de solicitud que incluye las credenciales del cliente y el token de acceso.

```
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 último, creamos un método principal que llama a la `DescribeSessions` API. Para obtener más información, consulte [DescribeSessions \(p. 23\)](#).

```
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'}])
```

Referencia de la API del Administrador de sesiones

En esta sección se ofrecen descripciones, sintaxis y ejemplos de uso de cada una de las acciones de las acciones de la API de Session Manager.

Temas

- [CloseServers \(p. 8\)](#)
- [CreateSessions \(p. 10\)](#)
- [DescribeServers \(p. 15\)](#)
- [DescribeSessions \(p. 23\)](#)
- [DeleteSessions \(p. 28\)](#)
- [GetSessionConnectionData \(p. 30\)](#)
- [GetSessionScreenshots \(p. 34\)](#)
- [OpenServers \(p. 36\)](#)
- [UpdateSessionPermissions \(p. 38\)](#)

CloseServers

Cierra uno o más servidores NICE DCV. Al cerrar un servidor de NICE DCV, deja de estar disponible para la ubicación de una sesión de NICE DCV. No puede crear sesiones de NICE DCV en servidores cerrados. Al cerrar un servidor, se garantiza que no se esté ejecutando ninguna sesión en él y que los usuarios no puedan crear nuevas sesiones en él.

Temas

- [Parámetros de solicitud \(p. 5\)](#)
- [Parámetros de respuesta \(p. 9\)](#)
- [Ejemplo \(p. 9\)](#)

Parámetros de solicitud

ServerId

El ID del servidor que se va a cerrar.

Tipo: String

Obligatorio: sí

Force

Obliga a cerrar la operación. Si lo especifica `true`, el servidor se cierra incluso si tiene sesiones en ejecución. Las sesiones siguen ejecutándose.

Tipo: booleano

Obligatorio: no

Parámetros de respuesta

RequestId

El ID único de la solicitud.

SuccessfulList

Información sobre los servidores NICE DCV que se cerraron correctamente. Esta estructura de datos incluye el siguiente parámetro de respuesta anidada:

ServerId

El ID del servidor que se cerró correctamente.

UnsuccessfulList

Información sobre los servidores NICE DCV que no se pudo cerrar. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

CloseServerRequestData

Información sobre la solicitud original que falló. Esta estructura de datos incluye el siguiente parámetro de respuesta anidada:

ServerId

El ID del servidor NICE DCV que no se pudo cerrar.

Force

El parámetro de fuerza solicitado.

FailureCode

El código del error.

FailureReason

El motivo del error.

Ejemplo

Python

Solicitud

El siguiente ejemplo cierra dos servidores NICE DCV (`serverId1` y `serverId2`). `serverId2` El servidor no existe y se produce un error.

```
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"])
```

Respuesta

A continuación, se muestra el resultado de este ejemplo.

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

CreateSessions

Crea una nueva sesión de NICE DCV con los detalles especificados.

Acciones de API

- [Parámetros de solicitud \(p. 5\)](#)
- [Parámetros de respuesta \(p. 9\)](#)
- [Ejemplo \(p. 9\)](#)

Parámetros de solicitud

Name

El nombre de la sesión.

Tipo: String

Obligatorio: sí

Owner

El nombre del propietario de la sesión. Debe ser el nombre de un usuario existente en el servidor NICE DCV de destino.

Tipo: String

Obligatorio: sí

Type

El tipo de sesión. Para obtener más información sobre los tipos de sesiones, consulte [Introducción a las sesiones de NICE DCV](#) en la Guía del administrador de NICE DCV.

Valores válidos: CONSOLE | VIRTUAL

Tipo: String

Obligatorio: sí

InitFile

Compatible con sesiones virtuales en servidores NICE DCV de Linux. No es compatible con las sesiones de consola de los servidores NICE DCV de Windows y Linux. La ruta al script personalizado en el servidor NICE DCV que se ejecutará para inicializar la sesión cuando se cree. La ruta del archivo es relativa al directorio de inicio especificado para el parámetro de configuración `delagent.init_folder` agente. Si el archivo está en el directorio de inicio especificado, especifique solo el nombre del archivo. Si el archivo no está en el directorio de inicio especificado, especifique la ruta relativa. Para obtener más información, consulte el [archivo de configuración del agente](#) en la Guía del administrador de sesiones de NICE DCV.

Tipo: String

Requerido: No

MaxConcurrents

Número máximo de clientes NICE DCV que se ejecutan simultáneamente.

Tipo: entero

Obligatorio: no

DcvGLEnabled

Indica si la sesión virtual está configurada para usar OpenGL basado en hardware. Compatible solo con sesiones virtuales. Este parámetro no es compatible con los servidores Windows NICE DCV.

Valores válidos: true | false

Tipo: booleano

Obligatorio: no

PermissionsFile

Contenido codificado en Base64 del archivo de permisos. El valor predeterminado son los valores predeterminados del servidor si se omite. Para obtener más información, consulte [Configuración de la autorización de NICE DCV](#) en la Guía del administrador de NICE DCV.

Tipo: String

Requerido: No

EnqueueRequest

Indica si se debe poner en cola la solicitud si no se puede cumplir inmediatamente.

Tipo: booleano

Valor predeterminado: falso

Obligatorio: no

AutorunFile

Compatible con sesiones de consola en servidores NICE DCV de Windows y sesiones virtuales en servidores NICE DCV de Linux. No es compatible con las sesiones de consola de los servidores NICE DCV de Linux.

La ruta a un archivo del servidor host que se va a ejecutar dentro de la sesión. La ruta del archivo es relativa al directorio de ejecución automática especificado para el parámetro de configuración `delagent . autorun_folder agente`. Si el archivo está en el directorio de ejecución automática especificado, especifique únicamente el nombre del archivo. Si el archivo no está en el directorio de ejecución automática especificado, especifique la ruta relativa. Para obtener más información, consulte el [archivo de configuración del agente](#) en la Guía del administrador de sesiones de NICE DCV.

El archivo se ejecuta en nombre del propietario especificado. El propietario especificado debe tener permiso para ejecutar el archivo en el servidor. En los servidores NICE DCV de Windows, el archivo se ejecuta cuando el propietario inicia sesión. En los servidores NICE DCV de Linux, el archivo se ejecuta cuando se crea la sesión.

Tipo: String

Requerido: No

AutorunFileArguments

Compatible con sesiones virtuales en servidores NICE DCV de Linux. No se admite en las sesiones de consola de los servidores NICE DCV de Windows y Linux. Los argumentos de la línea de comandos `AutorunFile` se transmiten al ejecutarlos dentro de la sesión. Los argumentos se pasan en el orden en que aparecen en la matriz dada. Se puede configurar el número máximo permitido de argumentos y la longitud máxima permitida de cada argumento. Para obtener más información, consulte el [archivo de configuración del corredor](#) en la Guía del administrador de sesiones de NICE DCV.

Tipo: matriz de cadenas

Obligatorio: no

DisableRetryOnFailure

Indica si no se debe volver a intentar la solicitud de creación de sesión después de que se produzca un error en un host de NICE DCV por cualquier motivo. Para obtener más información sobre el mecanismo de creación de reintentos de sesión, consulte el [archivo de configuración del corredor](#) en la Guía del administrador de sesiones de NICE DCV.

Tipo: booleano

Valor predeterminado: falso

Obligatorio: no

Requirements

Los requisitos que debe cumplir el servidor para poder realizar la sesión. Los requisitos pueden incluir etiquetas de servidor y/o propiedades del servidor; tanto las etiquetas como las propiedades del servidor se recuperan llamando a la `DescribeServersAPI`.

Expresiones de condiciones de requisitos:

- $un! = b$ verdadero si a no es igual a b
- $a = b$ es cierto si a es igual a b
- $a > b$ es cierto si a es mayor que b
- $a \geq b$ es verdadero si a es mayor o igual que b
- $a < b$ es verdadero si a es menor que b

- $a \leq b$ verdadero si a es menor o igual que b
- $a = b$ es cierto si a contiene la cadena b

Requisitos de operadores booleanos:

- a y b son verdaderas si a y b son verdaderas
- a o b son verdaderas si a o b son verdaderas
- no es verdadero si a es falso

Las claves de etiqueta deben tener el prefijo de tag: , las propiedades del servidor deben tener el prefijo server: .Las expresiones de requisitos admiten paréntesis().

Ejemplos 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"`

Los valores numéricos se pueden especificar mediante la notación exponencial, por ejemplo:"server:Host.Memory.TotalBytes > 1024E6".

Las propiedades del servidor compatibles son:

- 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

Requerido: No

StorageRoot

Especifica la ruta a la carpeta usada para el almacenamiento de sesión. Para obtener más información sobre el almacenamiento de sesiones de NICE DCV, consulte [Habilitar el almacenamiento de sesiones](#) en la Guía del administrador de NICE DCV.

Tipo: String

Requerido: No

Parámetros de respuesta

Id

El ID único de la sesión.

Name

El nombre de la sesión.

Owner

El propietario de la sesión.

Type

El tipo de sesión.

State

El estado de la sesión. Si la solicitud se completa correctamente, la sesión pasa alCREATING estado.

Substate

El subestado de la sesión. Si la solicitud se completa correctamente, el subestado entra en elSESSION_PLACING subestado.

Ejemplo

Python

Solicitud

En el ejemplo siguiente se crean tres sesiones.

```
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),  
    ])
```

Respuesta

A continuación, se muestra el resultado de este ejemplo.

```
{  
    "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

Describe uno o varios servidores NICE DCV.

Temas

- [Parámetros de solicitud \(p. 5\)](#)
- [Parámetros de respuesta \(p. 9\)](#)
- [Ejemplo \(p. 9\)](#)

Parámetros de solicitud

ServerIds

Los identificadores de los servidores de NICE DCV que se van a describir. Si no se especifica ningún identificador, todos los servidores se devuelven en una salida paginada.

Tipo: matriz de cadenas

Obligatorio: no

NextToken

El token que se utilizará para recuperar la siguiente página de resultados.

Tipo: String

Requerido: No

MaxResults

El número máximo de resultados que devolverá la solicitud en la salida paginada. Cuando se utiliza este parámetro, la solicitud devuelve solo el número especificado de resultados en una sola página junto con un elemento deNextToken respuesta. Los resultados restantes de la solicitud inicial se pueden ver enviando otra solicitud con elNextToken valor devuelto.

Rango válido: 1-1000

Predeterminado: 1000

Tipo: entero

Obligatorio: no

Parámetros de respuesta

RequestId

El ID único de la solicitud.

Servers

Información sobre los servidores NICE DCV. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Id

El ID único del servidor NICE DCV.

Ip

Dirección IP del servidor NICE DCV.

Hostname

El nombre de host del servidor NICE DCV.

Endpoints

Información sobre los extremos del servidor NICE DCV. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

IpAddress

Dirección IP del punto de conexión del servidor.

Port

El puerto del punto de conexión del servidor.

Protocol

Protocolo utilizado por el punto de conexión del servidor. Entre los valores posibles se incluyen:

- HTTP— El punto final utiliza el protocolo WebSocket (TCP).
- QUIC— El punto final utiliza el protocolo QUIC (UDP).

WebUrlPath

La ruta URL web del extremo del servidor. Disponible solo para el protocolo HTTP.

Version

La versión del servidor NICE DCV.

SessionManagerAgentVersion

La versión del agente Session Manager que se ejecuta en el servidor NICE DCV.

Availability

La disponibilidad del servidor NICE DCV. Entre los valores posibles se incluyen:

- AVAILABLE— El servidor está disponible y listo para la colocación de la sesión.
- UNAVAILABLE— El servidor no está disponible y no puede aceptar la ubicación de la sesión.

UnavailabilityReason

El motivo de la falta de disponibilidad del servidor NICE DCV. Entre los valores posibles se incluyen:

- SERVER_FULL— El servidor NICE DCV ha alcanzado el número máximo de sesiones simultáneas que puede ejecutar.
- SERVER_CLOSED— El servidor NICE DCV no está disponible mediante la CloseServerAPI.
- UNREACHABLE_AGENT— El agente de gestión de sesiones no puede comunicarse con el agente de gestión de sesiones del servidor NICE DCV.
- UNHEALTHY_DCV_SERVER— El agente de gestión de sesiones no puede comunicarse con el servidor NICE DCV.
- EXISTING_LOGGED_IN_USER— (Solo servidores NICE DCV de Windows) Un usuario ha iniciado sesión actualmente en el servidor NICE DCV mediante RDP.
- UNKNOWN— El agente de Session Manager no puede determinar el motivo.

ConsoleSessionCount

El número de sesiones de consola en el servidor NICE DCV.

VirtualSessionCount

El número de sesiones virtuales en el servidor NICE DCV.

Host

Información sobre el servidor host en el que se ejecuta el servidor NICE DCV. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Os

Información sobre el sistema operativo del servidor anfitrión. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Family

La familia de sistemas operativos. Entre los valores posibles se incluyen:

- `windows`— El servidor host ejecuta un sistema operativo Windows.
- `linux`— El servidor host ejecuta un sistema operativo Linux.

Name

Nombre del sistema operativo.

Version

Versión del sistema operativo.

KernelVersion

(Solo para Linux) Versión del núcleo del sistema operativo.

BuildNumber

(Solo Windows) El número de compilación del sistema operativo.

Memory

Información sobre la memoria del servidor anfitrión. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

TotalBytes

La memoria total, en bytes, del servidor host.

UsedBytes

La memoria utilizada, en bytes, en el servidor host.

Swap

Información sobre el archivo de intercambio del servidor anfitrión. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

TotalBytes

El tamaño total del archivo de intercambio, en bytes, en el servidor host.

UsedBytes

El tamaño del archivo de intercambio utilizado, en bytes, en el servidor host.

Aws

Solo para servidores NICE DCV que se ejecuten en una instancia de Amazon EC2. AWS-información específica. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Region

AWSRegión de la instancia de Amazon EC2.

Ec2InstanceType

Tipo de instancia de Amazon EC2.

Ec2InstanceId

El ID de la instancia Amazon EC2.

Ec2ImageId

El ID de la imagen de Amazon EC2.

CpuInfo

Información sobre las CPU del servidor anfitrión. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Vendor

El proveedor de la CPU del servidor host.

ModelName

El nombre del modelo de la CPU del servidor host.

Architecture

La arquitectura de la CPU del servidor anfitrión.

NumberOfCpus

La cantidad de CPU del servidor host.

PhysicalCorePerCpu

El número de núcleos de CPU por CPU.

CpuLoadAverage

Información sobre la carga de la CPU del servidor host. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

OneMinute

Carga media de la CPU durante el último período de 1 minuto.

FiveMinutes

Carga media de la CPU durante los últimos 5 minutos.

FifteenMinutes

Carga media de la CPU durante los últimos 15 minutos.

Gpus

Información sobre las GPU del servidor anfitrión. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Vendor

El proveedor de la GPU del servidor anfitrión.

ModelName

El nombre del modelo de la GPU del servidor host.

LoggedInUsers

Los usuarios que han iniciado sesión actualmente en el servidor host. Esta estructura de datos incluye el siguiente parámetro de respuesta anidada:

Username

El nombre de usuario del usuario que ha iniciado sesión.

Tags

Las etiquetas asignadas al servidor. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Key

La clave de la etiqueta.

Value

El valor de la etiqueta.

Ejemplo

Python

Solicitud

En el siguiente ejemplo se obtienen todos los servidores NICE DCV disponibles. Los resultados se pagan para mostrar dos 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)
```

Respuesta

A continuación, se muestra el resultado de este ejemplo.

```
{
  "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,
    }
  ]
}
```


Administrador NICE DCV del Administrador
del programa Guía para desarrolladores
Ejemplo

```
"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

Describe una o varias sesiones de NICE DCV.

Temas

- [Parámetros de solicitud \(p. 5\)](#)
- [Parámetros de respuesta \(p. 9\)](#)
- [Ejemplo \(p. 9\)](#)

Parámetros de solicitud

SessionIds

Los ID de las sesiones que se van a describir.

Tipo: String

Requerido: No

NextToken

El token que se utilizará para recuperar la siguiente página de resultados.

Tipo: String

Requerido: No

Filters

Filtros adicionales para aplicar a la solicitud. Los filtros compatibles incluyen:

- Etiqueta: clave: las etiquetas asignadas a la sesión.
- Propietario: propietario de la sesión.

Tipo: String

Requerido: No

Parámetros de respuesta

Id

El ID único de la sesión.

Name

El nombre de la sesión.

Owner

El propietario de la sesión.

Server

Información sobre el servidor en el que se ejecuta la sesión. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Ip

Dirección IP del anfitrión del servidor de NICE DCV.

Hostname

El nombre de host del servidor NICE DCV.

Port

El puerto a través del cual el servidor NICE DCV se comunica con los clientes de NICE DCV.

Endpoints

Información sobre los extremos del servidor NICE DCV. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

IpAddress

Dirección IP del punto de conexión del servidor.

Port

El puerto del punto de conexión del servidor.

Protocol

Protocolo utilizado por el punto de conexión del servidor. Entre los valores posibles se incluyen:

- HTTP— El punto final utiliza el protocolo WebSocket (TCP).
- QUIC— El punto final utiliza el protocolo QUIC (UDP).

WebUrlPath

La ruta URL web del extremo del servidor. Disponible solo para el protocolo HTTP.

Tags

Las etiquetas asignadas al servidor. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Key

La clave de la etiqueta.

Value

El valor de la etiqueta.

Type

El tipo de sesión.

State

El estado actual de la sesión. Los valores posibles son los siguientes:

- CREATING- el Broker está en proceso de crear la sesión.
- READY- la sesión está lista para aceptar conexiones de clientes.
- DELETING- se está eliminando la sesión.
- DELETED- se ha eliminado la sesión.
- UNKNOWN- no se puede determinar el estado de la sesión. Es posible que el corredor y el agente no puedan comunicarse.

Substate

El subestado actual de la sesión. Los valores posibles son los siguientes:

- SESSION_PLACING- la sesión está esperando ser colocada en un servidor DCV disponible.
- PENDING_PREPARATION- la sesión se ha creado pero no se puede utilizar; está vinculada a un servidor DCV.

CreationTime

La fecha y hora de creación de la sesión.

LastDisconnectionTime

La fecha y hora de desconexión del cliente por última vez.

NumOfConnections

El número de conexiones de cliente activas.

StorageRoot

Especifica la ruta a la carpeta usada para el almacenamiento de sesión. Para obtener más información sobre el almacenamiento de sesiones de NICE DCV, consulte [Habilitar el almacenamiento de sesiones](#) en la Guía del administrador de NICE DCV.

Tipo: String

Requerido: No

Ejemplo

Python

Solicitud

En el siguiente ejemplo, se describen las sesiones que son propiedad de user1 y tienen una etiqueta deos=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'}])
```

Respuesta

A continuación, se muestra el resultado de este ejemplo.

```
{
  "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"
          }
        ]
      }
    }
  ]
}
```

Administrador NICE DCV del Administrador
del programa Guía para desarrolladores
Ejemplo

```
        "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

Elimina la sesión de NICE DCV especificada y la elimina de la memoria caché del Broker.

Temas

- [Parámetros de solicitud \(p. 5\)](#)
- [Parámetros de respuesta \(p. 9\)](#)
- [Ejemplo \(p. 9\)](#)

Parámetros de solicitud

SessionId

El ID de la sesión que se va a eliminar.

Tipo: String

Obligatorio: sí

Owner

El propietario de la sesión que se va a eliminar.

Tipo: String

Obligatorio: sí

Force

Elimina una sesión de la memoria caché del Broker al intentar eliminarla del servidor NICE DCV. Esto es útil para eliminar las sesiones desactualizadas de la memoria caché del Broker. Por ejemplo, si se detuvo un servidor NICE DCV, pero las sesiones siguen registradas en el Broker, utilice este indicador para purgar las sesiones de la memoria caché del Broker.

Tenga en cuenta que si la sesión sigue activa, el Broker la vuelve a almacenar en caché.

Valores válidos: true | false

Tipo: booleano

Obligatorio: no

Parámetros de respuesta

SessionId

El ID de la sesión

State

Solo se devuelve si las sesiones se eliminaron correctamente. Indica el estado actual de la sesión. Si la solicitud se completa correctamente, la sesión pasa alDELETING estado. Puede que transcurran unos minutos hasta que se elimine la sesión. Cuando se ha eliminado, el estado pasa deDELETING aDELETED.

FailureReason

Solo se devuelve si no se pueden eliminar algunas sesiones. Indica por qué no se pudo eliminar la sesión.

Ejemplo

Python

Solicitud

En el siguiente ejemplo se eliminan dos sesiones: una sesión con un ID delSessionId123 que es propiedad de user1 de y una sesión con un IDSessionIdabc que es propiedad de 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')])
```

Respuesta

A continuación, se muestra el resultado de este ejemplo. SessionId123se ha eliminado correctamente, pero no se haSessionIdabc podido eliminar.

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

```
}  
  ]  
}
```

GetSessionConnectionData

Obtiene información de conexión para la conexión de un usuario específico a una sesión de NICE DCV específica.

Temas

- [Parámetros de solicitud \(p. 5\)](#)
- [Parámetros de respuesta \(p. 9\)](#)
- [Información adicional \(p. 32\)](#)
- [Ejemplo \(p. 9\)](#)

Parámetros de solicitud

SessionId

El ID de sesión de la que se puede ver información de conexión.

Tipo: String

Obligatorio: sí

User

Nombre del usuario del que se puede ver la información de conexión.

Tipo: String

Obligatorio: sí

Parámetros de respuesta

Id

El ID único de la sesión.

Name

El nombre de la sesión.

Owner

El propietario de la sesión.

Server

Información sobre el servidor en el que se ejecuta la sesión. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Ip

Dirección IP del anfitrión del servidor de NICE DCV.

Hostname

El nombre de host del servidor NICE DCV.

Port

El puerto a través del cual el servidor NICE DCV se comunica con los clientes de NICE DCV.

Endpoints

Información sobre los extremos del servidor NICE DCV. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

IpAddress

Dirección IP del punto de conexión del servidor.

Port

El puerto del punto de conexión del servidor.

Protocol

Protocolo utilizado por el punto de conexión del servidor. Entre los valores posibles se incluyen:

- HTTP— El punto final utiliza el protocolo WebSocket (TCP).
- QUIC— El punto final utiliza el protocolo QUIC (UDP).

WebUrlPath

La ruta URL web del extremo del servidor. Disponible solo para el protocolo HTTP.

WebUrlPath

La ruta al archivo de configuración del servidor NICE DCV.

Tags

Las etiquetas asignadas al servidor. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Key

La clave de la etiqueta.

Value

El valor de la etiqueta.

Type

El tipo de sesión.

State

El estado actual de la sesión. Los valores posibles son los siguientes:

- CREATING- el Broker está en proceso de crear la sesión.
- READY- la sesión está lista para aceptar conexiones de clientes.
- DELETING- se está eliminando la sesión.
- DELETED- se ha eliminado la sesión.

- UNKNOWN- no se puede determinar el estado de la sesión. Es posible que el corredor y el agente no puedan comunicarse.

CreationTime

La fecha y hora de creación de la sesión.

LastDisconnectionTime

La fecha y hora de desconexión del cliente por última vez.

NumOfConnections

El número de conexiones simultáneas que el usuario tiene a la sesión.

ConnectionToken

El token de autenticación utilizado para conectarse a la sesión.

Información adicional

La información obtenida de esta API se puede pasar a un cliente de NICE DCV para conectarse a la sesión de NICE DCV.

En el caso del cliente web NICE DCV, puede crear una URL que se pueda abrir en el navegador. La URL tiene el siguiente formato:

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

En el caso del cliente nativo NICE DCV, puede crear una URL con el `dcv://` esquema. Cuando se instala el cliente nativo de NICE DCV, se registra en el sistema como controlador de `dcv://` las URL. La URL tiene el siguiente formato:

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

Note

Si utiliza Amazon EC2, la dirección IP debe ser la pública. Si su configuración tiene servidores NICE DCV detrás de una puerta de enlace, especifique la dirección de la puerta de enlace en lugar de la que devuelve la `SessionConnectionData` API.

Ejemplo

Python

Solicitud

En el ejemplo siguiente se obtiene la información de conexión de un usuario con un nombre de usuario `user1` y una sesión con un ID de `sessionId12345`.

```
def get_session_connection_api():  
    api_instance =  
    swagger_client.GetSessionConnectionDataApi(swagger_client.ApiClient(get_client_configuration()))
```

```
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')
```

Respuesta

A continuación, se muestra el resultado de este ejemplo.

```
{
  "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":
  "EXAMPLEi0iJm0WM1YTRhZi1jZmU0LTQ0ZjEtYjZlOC04ZjY0YjM4ZTE2ZDkiLCJ0eXAI0iJKV1QiLCJhbGciOiJSUzI1NiJ9.
  tngiKXevUxhhJvm3BPJYRs9NPE4GCJRTc13EXAMPLEIxNEPPh5IMcVmR0fU1WKPNry4ypPTp3rsZ7YWjCTSfs1GoN3R_nLFyAxk
  Kqtpd5GH0D-E8FwsedV-
  Q2bRQ4y9y1q0MgFU4QjaSMypUuYR0YjkCaoainjMEZew4A33fG40wATrBvoivBiNwdNpytHX2CD0uk_k0k_DWeZjMvv9jF1f5EX
  h_GaMgHm1tqBIA4jdPD7i0CmC2e7413KFy-
  EQ4Ej1cM7RjLwhFuWpKWAVJxogJjYpfoKkaPo4KxvJjJIPYhkscklINQpe2W5rn1xCq7sC7ptcGw17DUobP7egRv9H37VD8SrkL
  hK1G4G8erHv19HIrTR9_c884fNrTCC8DvC062e4KYdLkAhhJmboN9CAGIGFyd2c1AY_CzzvDL0EXAMLE"
}
}
```

GetSessionScreenshots

Obtiene capturas de pantalla de una o más sesiones de NICE DCV.

El tipo de archivo de imagen y la resolución de la captura de pantalla dependen de la configuración de Session Manager Broker. Para modificar el tipo de archivo de imagen, configure el `session-screenshot-format` parámetro. Para modificar la resolución, configure los `session-screenshot-max-height` y `session-screenshot-max-width` parámetros. Para obtener más información, consulte el [archivo de configuración del corredor](#) en la Guía del administrador de sesiones de NICE DCV.

Temas

- [Parámetros de solicitud \(p. 5\)](#)
- [Parámetros de respuesta \(p. 9\)](#)
- [Ejemplo \(p. 9\)](#)

Parámetros de solicitud

SessionId

El ID de la sesión de NICE DCV de la que se va a obtener la captura de pantalla.

Tipo: String

Obligatorio: sí

Parámetros de respuesta

RequestId

El ID único de la solicitud.

SuccessfulList

Información sobre las capturas de pantalla realizadas correctamente. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

SessionScreenshot

Información sobre las capturas de pantalla. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

SessionId

El ID de la sesión de NICE DCV desde la que se tomó la captura de pantalla.

Images

: información acerca de las imágenes. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Format

El formato de la imagen. Los valores posibles incluyen: jpeg y png.

Data

El formato codificado en base64 de la imagen de captura de pantalla.

CreationTime

La fecha y hora en que se realizó la captura de pantalla.

Primary

Indica si la captura de pantalla es de la pantalla principal de la sesión de NICE DCV.

UnsuccessfulList

Información sobre las capturas de pantalla fallidas. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

GetSessionScreenshotRequestData

La solicitud original que se ha producido el siguiente error.

SessionId

El ID de la sesión de NICE DCV desde la que se iba a tomar la captura de pantalla.

FailureReason

El motivo del error.

Ejemplo

Python

Solicitud

En el siguiente ejemplo se obtienen capturas de pantalla de dos sesiones (sessionId1 y sessionId2). sessionId2La sesión no existe y se produce un error.

```
from swagger_client.models.describe_servers_request_data import
    DescribeServersRequestData

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):
```

```
request = [GetSessionScreenshotRequestData(session_id=session_id) 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"])
```

Respuesta

A continuación, se muestra el resultado de este ejemplo.

```
{
  "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 uno o más servidores NICE DCV. Antes de poder crear sesiones de NICE DCV en un servidor de NICE DCV, debe cambiar el estado del servidor para que se abra. Una vez abierto el servidor NICE DCV, puede crear sesiones NICE DCV en el servidor.

Temas

- [Parámetros de solicitud \(p. 5\)](#)
- [Parámetros de respuesta \(p. 9\)](#)
- [Ejemplo \(p. 9\)](#)

Parámetros de solicitud

ServerId

El ID del servidor que se va a abrir.

Tipo: String

Obligatorio: sí

Parámetros de respuesta

RequestId

El ID único de la solicitud.

SuccessfulList

Información sobre los servidores NICE DCV que se abrieron correctamente. Esta estructura de datos incluye el siguiente parámetro de respuesta anidada:

ServerId

El ID del servidor que se abrió correctamente.

UnsuccessfulList

Información sobre los servidores NICE DCV que no se pudo abrir. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

OpenServerRequestData

Información sobre la solicitud original que falló. Esta estructura de datos incluye el siguiente parámetro de respuesta anidada:

ServerId

El ID del servidor NICE DCV que no se pudo abrir.

FailureCode

El código del error.

FailureReason

El motivo del error.

Ejemplo

Python

Solicitud

El siguiente ejemplo abre dos servidores NICE DCV (serverId1serverId2).

```
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"])
```

Respuesta

A continuación, se muestra el resultado de este ejemplo.

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

UpdateSessionPermissions

Actualiza los permisos de usuario para una sesión específica de NICE DCV.

Temas

- [Parámetros de solicitud \(p. 5\)](#)
- [Parámetros de respuesta \(p. 9\)](#)
- [Ejemplo \(p. 9\)](#)

Parámetros de solicitud

SessionId

El ID de sesión para la que se van a cambiar los permisos.

Tipo: String

Obligatorio: sí

Owner

El propietario de la sesión para la que se van a cambiar los permisos.

Tipo: String

Obligatorio: sí

PermissionFile

Contenido codificado en Base64 del archivo de permisos que se va a utilizar. Para obtener más información, consulte [Configuración de la autorización de NICE DCV](#) en la Guía del administrador de NICE DCV.

Tipo: String

Obligatorio: sí

Parámetros de respuesta

SessionId

Token de continuación si todavía no se arrojaron todas las instrucciones.

Ejemplo

Python

Solicitud

El siguiente ejemplo establece los permisos nuevos para una sesión con un identificador de sesión deSessionId1897.

```
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_instance
def 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']])
```

Respuesta

A continuación, se muestra el resultado de este ejemplo.

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

Notas de publicación e historial de documentos de NICE DCV Session Manager

Esta página proporciona las notas de publicación y el historial de documentos de NICE DCV Session Manager.

Temas

- [Notas de lanzamiento de NICE DCV Session Manager \(p. 40\)](#)
- [Historial de documentos \(p. 43\)](#)

Notas de lanzamiento de NICE DCV Session Manager

Esta sección proporciona una descripción general de las principales actualizaciones, versiones de funciones y correcciones de errores de NICE DCV Session Manager. Todas las actualizaciones están organizadas por fecha de lanzamiento. Actualizamos la documentación con frecuencia para responder a los comentarios que nos envía.

Temas

- [2023.0-14852 — 28 de marzo de 2023 \(p. 40\)](#)
- [2022.2-13907 — 11 de noviembre de 2022 \(p. 41\)](#)
- [2022.1-13067 — 29 de junio de 2022 \(p. 41\)](#)
- [2022.0-11952 — 23 de febrero de 2022 \(p. 41\)](#)
- [2021.3-11591 — 20 de diciembre de 2021 \(p. 41\)](#)
- [2021.2-11445 — 18 de noviembre de 2021 \(p. 42\)](#)
- [2021.2-11190 — 11 de octubre de 2021 \(p. 42\)](#)
- [2021.2-11042 — 01 de septiembre de 2021 \(p. 42\)](#)
- [2021.1-10557 — 31 de mayo de 2021 \(p. 42\)](#)
- [2021.0-10242 — 12 de abril de 2021 \(p. 43\)](#)
- [2020.2-9662 — 4 de diciembre de 2020 \(p. 43\)](#)
- [2020.2-9508 — 11 de noviembre de 2020 \(p. 43\)](#)

2023.0-14852 — 28 de marzo de 2023

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• agente	<ul style="list-style-type: none">• Se agregó soporte para Red Hat Enterprise Linux 9, Rocky Linux 9 y CentOS Stream 9.

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• agente• CLIPS: 132	

2022.2-13907 — 11 de noviembre de 2022

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• agente• agente• CLIPS: 123	<ul style="list-style-type: none">• Se agregó unSubstate campo enDescribeSessions respuesta.• Se solucionó un problema que podía provocar que la CLI no se conectara al corredor en función de la URL en uso.

2022.1-13067 — 29 de junio de 2022

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• agente• agente• CLIPS: 114	<ul style="list-style-type: none">• Se agregó soporte para ejecutar el corredor en instancias deAWS Graviton.• Se agregó soporte de agentes y corredores para Ubuntu 22.04.

2022.0-11952 — 23 de febrero de 2022

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• agente• agente• CLIPS: 112	<ul style="list-style-type: none">• Se agregó la capacidad de rotación de registros al agente.• Se agregó un parámetro de configuración para establecer el directorio principal de Java en el Broker.• Se mejoró el vaciado de datos de la memoria caché al disco en el Broker.• Validación de URL fija en la CLI.

2021.3-11591 — 20 de diciembre de 2021

Números de compilación	Nuevas características de
<ul style="list-style-type: none">• agente• agente• CLIPS: 92	<ul style="list-style-type: none">• Se agregó soporte para la integración con la pasarela de conexión NICE DCV.• Se agregó el agente

2021.2-11445 — 18 de noviembre de 2021

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• agente• agente• CLIPS: 54	<ul style="list-style-type: none">• Se ha corregido un problema con la validación de los nombres de inicio de sesión que incluyen un dominio de Windows.

2021.2-11190 — 11 de octubre de 2021

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• agente• agente• CLIPS: 54	<ul style="list-style-type: none">• Se ha corregido un problema en la interfaz de línea de comandos que impedía iniciar sesiones de Windows.

2021.2-11042 — 01 de septiembre de 2021

Números de compilación	Nuevas características de	Cambios y correcciones de errores
<ul style="list-style-type: none">• agente• agente• CLIPS: 37	<ul style="list-style-type: none">• NICE DCV administrador de sesión del agente Puede crear y administrar sesiones de NICE DCV en la CLI, en lugar de llamar a las API.• El administrador de sesiones NICE DCV introdujo la persistencia de datos de Broker. Para una mayor disponibilidad, los corredores pueden conservar la información del estado del servidor en un banco de datos externo y restaurar los datos al inicio.	<ul style="list-style-type: none">• Al registrar un servidor de autorización externo, ahora puede especificar el algoritmo que utiliza el servidor de autorización para firmar los tokens web con formato JSON. Con este cambio, puede usar Azure AD como servidor de autorización externo.

2021.1-10557 — 31 de mayo de 2021

Números de compilación	Nuevas características de	Cambios y correcciones de errores
<ul style="list-style-type: none">• agente• agente	<ul style="list-style-type: none">• El administrador de sesiones NICE DCV agregó soporte para los parámetros de entrada que se pasan al archivo de ejecución automática en Linux.• Las propiedades del servidor ahora se pueden transferir como requisitos a la CreateSessionsAPI.	<ul style="list-style-type: none">• Hemos solucionado un problema con el archivo de ejecución automática en Windows.

2021.0-10242 — 12 de abril de 2021

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• agente• agente	<ul style="list-style-type: none">• El administrador de sesiones NICE DCV introdujo las siguientes API nuevas:<ul style="list-style-type: none">• OpenServers• CloseServers• DescribeServers• GetSessionScreenshots• También introdujo los siguientes parámetros de configuración nuevos:<ul style="list-style-type: none">• Parámetros del bróker:<code>session-screenshot-max-width</code>,<code>session-screenshot-max-height</code>,<code>session-screenshot-format</code>,<code>create-sessions-queue-max-size</code>,<code>create-sessions-queue-max-time-seconds</code>.• Parámetros del agente:<code>agent.autorun_folder</code>,<code>max_virtual_sessions</code>,<code>ymax_concurrent_sessions_per_user</code>. <p>Parámetros del agente:<code>agent.autorun_folder</code>,<code>max_virtual_sessions</code>,<code>ymax_concurrent_sessions_per_user</code>.</p> <p>Parámetros del agente:<code>agent.autorun_folder</code>,<code>max_virtual_sessions</code>,<code>ymax_concurrent_sessions_per_user</code>.</p>

2020.2-9662— 4 de diciembre de 2020

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• agente• agente	<ul style="list-style-type: none">• Se solucionó un problema con los certificados TLS generados automáticamente que impedían que el Broker se iniciara.

2020.2-9508— 11 de noviembre de 2020

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• agente• agente	<ul style="list-style-type: none">• Versión NICE DCV agente

Historial de documentos

En la siguiente tabla, se indica la documentación de esta versión de NICE DCV.

Administrador NICE DCV del Administrador
del programa Guía para desarrolladores
Historial de documentos

Cambio	Descripción	Fecha
NICE DCV versión 2023.0	El administrador de sesiones de NICE DCV se ha actualizado para NICE DCV 2023.0. Para obtener más información, consulte 2023.0-14852 — 28 de marzo de 2023 (p. 40) .	28 de marzo de 2023
NICE DCV versión 2022.2	El administrador de sesiones de NICE DCV se ha actualizado para NICE DCV 2022.2. Para obtener más información, consulte 2022.2-13907 — 11 de noviembre de 2022 (p. 41) .	11 de noviembre de 2022
NICE DCV versión 2022.1	El administrador de sesiones de NICE DCV se ha actualizado para NICE DCV 2022.1. Para obtener más información, consulte 2022.1-13067 — 29 de junio de 2022 (p. 41) .	29 de junio de 2022
NICE DCV versión 2022.0	El administrador de sesiones de NICE DCV se ha actualizado para NICE DCV 2022.0. Para obtener más información, consulte 2022.0-11952 — 23 de febrero de 2022 (p. 41) .	23 de febrero de 2022
NICE DCV versión 2021.3	El administrador de sesiones de NICE DCV se ha actualizado para NICE DCV 2021.3. Para obtener más información, consulte 2021.3-11591 — 20 de diciembre de 2021 (p. 41) .	20 de diciembre de 2021
NICE DCV versión 2021.2	El administrador de sesiones de NICE DCV se ha actualizado para NICE DCV 2021.2. Para obtener más información, consulte 2021.2-11042 — 01 de septiembre de 2021 (p. 42) .	01 de septiembre de 2021
NICE DCV versión 2021.1	El administrador de sesiones de NICE DCV se ha actualizado para NICE DCV 2021.1. Para obtener más información, consulte 2021.1-10557 — 31 de mayo de 2021 (p. 42) .	31 de mayo de 2021
NICE DCV versión 2021.0	El administrador de sesiones de NICE DCV se ha actualizado para NICE DCV 2021.0. Para obtener más información, consulte 2021.0-10242 — 12 de abril de 2021 (p. 43) .	12 de abril de 2021
Versión inicial de NICE DCV Session Manager	La primera publicación de este contenido.	11 de noviembre de 2020

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.