



Panduan Developerr

Manajer Sesi NICE DCV



Manajer Sesi NICE DCV: Panduan Developerr

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

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

Table of Contents

Apa Session Manager?	1
Cara Session Manager bekerja	1
Fitur	3
Mulai	4
Menghasilkan Klien.	4
Daftarkan Klien Anda.	5
Dapatkan token akses dan buat permintaan API	5
Referensi API Session Manager sesi Session Manager API	9
CloseServers	9
Permintaan parameter	5
Parameter respons	10
Contoh	11
CreateSessions	12
Permintaan parameter	5
Parameter respons	10
Contoh	11
DescribeServers	20
Permintaan parameter	5
Parameter respons	10
Contoh	11
DescribeSessions	31
Permintaan parameter	5
Parameter respons	10
Contoh	11
DeleteSessions	37
Permintaan parameter	5
Parameter respons	10
Contoh	11
GetSessionConnectionData	40
Permintaan parameter	5
Parameter respons	10
Info tambahan	43
Contoh	11
GetSessionScreenshots	46

Permintaan parameter	5
Parameter respons	10
Contoh	11
OpenServers	50
Permintaan parameter	5
Parameter respons	10
Contoh	11
UpdateSessionPermissions	52
Permintaan parameter	5
Parameter respons	10
Contoh	11
Catatan Rilis dan Riwayat Dokumen	55
Catatan rilis	55
2023.1— November 9, 2023	56
2023.0-15065— 4 Mei 2023	56
2023.0-14852— 28 Maret 2023	56
2022.2-13907— November 11, 2022	56
2022.1-13067— 29 Juni 2022	57
2022.0-11952— 23 Februari 2022	57
2021.3-11591— Desember 20, 2021	57
2021.2-11445— November 18, 2021	58
2021.2-11190— 11 Oktober 2021	58
2021.2-11042— September 01, 2021	58
2021.1-10557— 31 Mei 2021	59
2021.0-10242— 12 April 2021	59
2020.2-9662— 04 Desember 2020	60
.....	60
Riwayat dokumen	60
.....	lxiii

Apa itu Manajer Sesi NICE DCV?

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

Panduan ini menjelaskan cara menggunakan API Session Manager untuk mengelola siklus hidup sesi NICE DCV. Untuk informasi selengkapnya tentang cara menginstal dan mengkonfigurasi Broker dan Agen Manajer Sesi, lihat Panduan Administrator Manajer Sesi NICE DCV.

Prasyarat

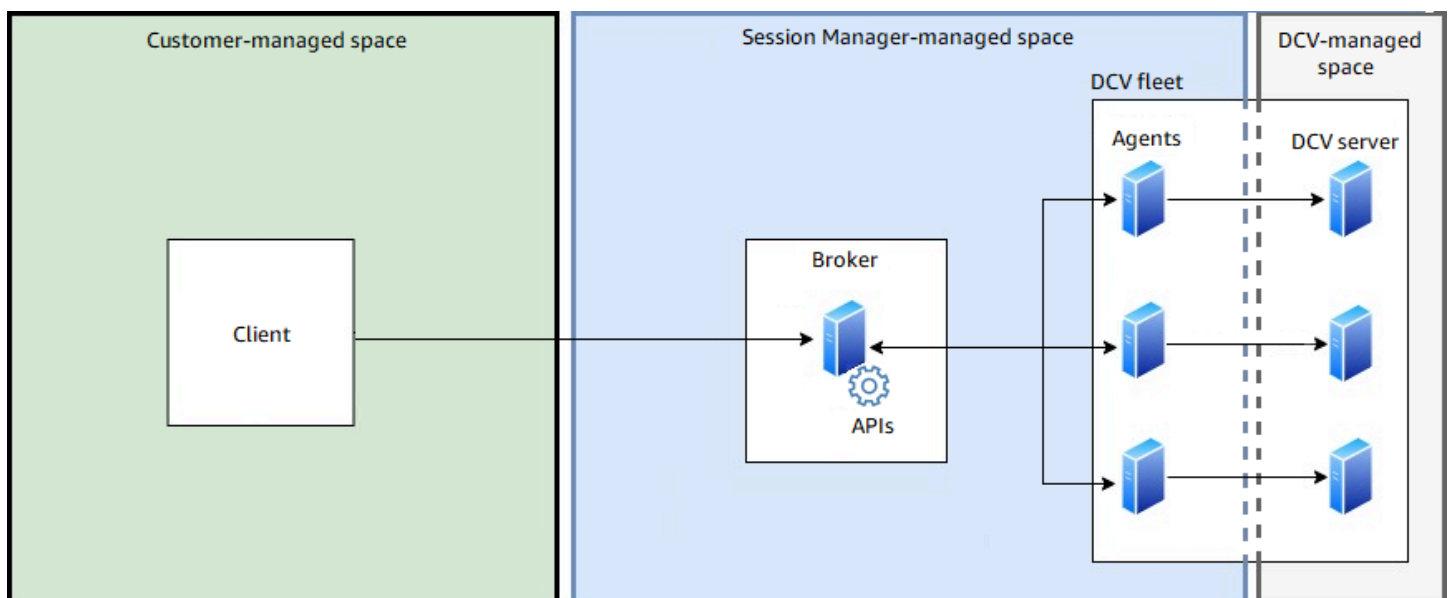
Sebelum Anda mulai bekerja dengan Session Manager API, pastikan Anda terbiasa dengan sesi NICE DCV dan NICE DCV. Untuk informasi selengkapnya, lihat [Panduan Administrator NICE DCV](#).

Topik

- [Cara Session Manager bekerja](#)
- [Fitur](#)

Cara Session Manager bekerja

Diagram berikut menunjukkan komponen tingkat tinggi Session Manager.



Pialang

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

Agen

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

API

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

Klien

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

Token akses

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

API Klien

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

Sesi NICE DCV

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

- CREATING—Broker sedang dalam proses pembuatan sesi.

- READYSession siap menerima koneksi klien.
- DELETINGSession sedang dihapus.
- DELETEDSession telah dihapus.
- UNKNOWN—able untuk menentukan keadaan sesi. Broker dan Agen mungkin tidak dapat berkomunikasi.

Fitur

Manajer Session DCV menawarkan fitur-fitur berikut:

- Menyediakan informasi sesi NICE DCV —dapatkan informasi tentang sesi yang berjalan di beberapa server NICE DCV.
- Kelola siklus hidup untuk beberapa sesi NICE DCV —buat atau hapus beberapa sesi untuk beberapa pengguna di beberapa server NICE DCV dengan satu permintaan API.
- Mendukung tag —use custom tags untuk menargetkan sekelompok server NICE DCV saat membuat sesi.
- Mengelola izin untuk beberapa sesi NICE DCV —memodifikasi izin pengguna untuk beberapa sesi dengan satu permintaan API.
- Menyediakan informasi koneksi —retrieve informasi koneksi klien untuk sesi NICE DCV.
- Mendukung cloud dan di tempat —gunakan Session Manager diAWS, di tempat, atau dengan server berbasis cloud alternatif.

Mulai

Bagian ini menjelaskan cara memulai dengan Klien.

Pada bagian ini, kami akan menunjukkan kepada Anda bagaimana melakukan ini dengan menggunakan `DescribeSessions` API sebagai contoh.

Topik

- [Menghasilkan Klien.](#)
- [Daftarkan Klien Anda.](#)
- [Dapatkan token akses dan buat permintaan API](#)

Menghasilkan Klien.

API Session Manager didefinisikan dalam satu file YAKL. API didasarkan pada spesifikasi OpenAPI3.0, yang mendefinisikan antarmuka standar, bahasa-agnostik untuk API RESTful. Untuk informasi lebih lanjut, lihat [Spesifikasi OpenKlien](#).

Dengan menggunakan file YAKL, Anda dapat menghasilkan klien API dalam salah satu bahasa yang didukung. Untuk melakukannya, Anda harus menggunakan Swagger Codegen 3.0 atau yang lebih baru. Untuk informasi selengkapnya tentang bahasa yang didukung, lihat [repo kodegen yang angkuh](#).

Untuk menghasilkan klien API

1. Unduh file YAKL API Pengelola Sesi dari Broker Manajer Sesi. File YAKL tersedia di URL berikut.

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

2. Instal Swagger Codegen.

- macOS

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

- Platform lainnya

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



```
$ cd swagger-codegen
```

3. Hasilkan klien API.

- macOS

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

- Platform lainnya

```
$ 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
```

Daftarkan Klien Anda.

Untuk membuat permintaan API, Anda harus terlebih dahulu mengambil token akses dari Broker. Untuk mendapatkan token akses dari Broker, Anda harus memberikan kredensi kepada Broker untuk API klien Anda. Kredensialnya didasarkan pada ID klien dan kata sandi klien yang dihasilkan saat klien Anda terdaftar di Broker. Jika Anda tidak memiliki ID klien dan kata sandi klien untuk klien Anda, Anda harus memintanya dari administrator Broker Anda. Untuk informasi lebih lanjut tentang mendaftarkan API klien Anda ke Broker dan mendapatkan ID klien dan kata sandi, lihat [register-api-client](#).

Dapatkan token akses dan buat permintaan API

Pertama kita mengimpor model yang dibutuhkan untuk aplikasi.

Kemudian kita mendeklarasikan variabel untuk ID klien (`__CLIENT_ID`), password klien (`__CLIENT_SECRET`), dan URL Broker, termasuk nomor port (`__PROTOCOL_HOST_PORT`).

Selanjutnya, kita membuat `fungsi build_client_credentials` yang disebut yang menghasilkan kredensial klien. Untuk menghasilkan kredensi klien, Anda harus terlebih dahulu menggabungkan ID klien dan kata sandi klien dan memisahkan nilai dengan titik dua (`client_ID:client_password`), dan kemudian Base64 menyandikan seluruh string.

```

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 = 'MzcxNzJhN2UtYjEzNS00MjNjLTg2N2YtMjF1ZmRlZWJmDU1'
__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')

```

Sekarang setelah kami memiliki kredensi klien kami, kami dapat menggunakannya untuk meminta token akses dari Broker. Untuk melakukan ini, kita membuat fungsi yang disebut `get_access_token`. Anda harus memanggil POST on `https://Broker_IP:8443/oauth2/token?grant_type=client_credentials`, dan menyediakan header otorisasi, yang mencakup kredensial klien basic-dikodekan, dan jenis konten `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
```

Sekarang, kita membuat fungsi yang diperlukan untuk instantiate API klien. Untuk instantiate API klien, Anda harus menentukan konfigurasi klien dan header yang akan digunakan untuk permintaan. `get_client_configuration` Fungsi ini membuat objek konfigurasi yang mencakup alamat IP dan port Broker dan jalur ke sertifikat yang ditandatangani sendiri Broker, yang seharusnya Anda terima dari administrator Broker. `set_request_headers` Fungsi ini menciptakan objek header permintaan yang mencakup kredensial klien dan token akses.

```
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
```

Akhirnya, kita membuat metode utama yang memanggil `DescribeSessions` API. Untuk informasi selengkapnya, lihat [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'}])
```

Referensi API Session Manager sesi Session Manager API

Bagian ini memberikan deskripsi, sintaks, dan contoh penggunaan untuk setiap tindakan API Session Manager.

Topik

- [CloseServers](#)
- [CreateSessions](#)
- [DescribeServers](#)
- [DescribeSessions](#)
- [DeleteSessions](#)
- [GetSessionConnectionData](#)
- [GetSessionScreenshots](#)
- [OpenServers](#)
- [UpdateSessionPermissions](#)

CloseServers

Menutup satu atau lebih server NICE DCV. Ketika Anda menutup server NICE DCV, Anda membuatnya tidak tersedia untuk penempatan sesi NICE DCV. Anda tidak dapat membuat sesi NICE DCV pada server tertutup. Menutup server memastikan bahwa tidak ada sesi yang berjalan di atasnya dan bahwa pengguna tidak dapat membuat sesi baru di atasnya.

Topik

- [Permintaan parameter](#)
- [Parameter respons](#)
- [Contoh](#)

Permintaan parameter

ServerId

ID server untuk menutup.

Tipe: String

Wajib: Ya

Force

Memaksa operasi dekat. Jika Anda menentukan `true`, server ditutup meskipun telah menjalankan sesi. Session terus berjalan.

Tipe: Boolean

Wajib: Tidak

Parameter respons

RequestId

ID unik dari permintaan tersebut.

SuccessfulList

Informasi tentang server NICE DCV yang berhasil ditutup. Struktur data ini mencakup parameter respons bersarang berikut:

ServerId

ID server yang berhasil ditutup.

UnsuccessfulList

Informasi tentang server NICE DCV yang tidak dapat ditutup. Struktur data ini mencakup parameter respons bersarang berikut:

CloseServerRequestData

Informasi tentang permintaan asli yang gagal. Struktur data ini mencakup parameter respons bersarang berikut:

ServerId

ID server NICE DCV yang tidak dapat ditutup.

Force

Parameter kekuatan yang diminta.

FailureCode

Kode kegagalan.

FailureReason

Sebab kegagalan.

Contoh

Python

Permintaan

Contoh berikut menutup dua server NICE DCV (serverId1 dan serverId2). Server serverId2 tidak ada dan mengakibatkan kegagalan.

```
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"])
```

Response

Berikut adalah contoh output.

```
{
```

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

CreateSessions

Menciptakan sesi NICE DCV baru dengan rincian yang ditentukan.

Tindakan API

- [Permintaan parameter](#)
- [Parameter respons](#)
- [Contoh](#)

Permintaan parameter

Name

Nama untuk sesi.

Tipe: String

Wajib: Ya

Owner

Nama pemilik sesi sesi sesi sesi sesi sesi sesi sesi sesi sesi ini. Ini harus nama pengguna yang ada di server DCV target NICE.

Tipe: String

Wajib: Ya

Type

Jenis sesi. Untuk informasi selengkapnya tentang jenis sesi, lihat [Pengantar Sesi NICE DCV](#) di Panduan Administrator NICE DCV.

Nilai yang valid: CONSOLE | VIRTUAL

Tipe: String

Wajib: Ya

InitFile

Didukung dengan sesi virtual pada server Linux NICE DCV. Hal ini tidak didukung dengan sesi konsol pada server Windows dan Linux NICE DCV. Jalur ke skrip kustom pada server NICE DCV untuk dijalankan untuk menginisialisasi sesi ketika dibuat. Path file relatif terhadap direktori init yang ditentukan untuk parameter konfigurasi `agent.init_folder` Agen. Jika file berada di direktori init yang ditentukan, tentukan nama file saja. Jika file tidak dalam direktori init yang ditentukan, tentukan path relatif. Untuk informasi selengkapnya, lihat [File konfigurasi agen](#) di Panduan Administrator Manajer Sesi NICE DCV.

Tipe: String

Wajib: Tidak

MaxConcurrents

Jumlah maksimum klien NICE DCV bersamaan.

Tipe: Integer

Wajib: Tidak

DcvGLEnabled

Menunjukkan apakah sesi virtual dikonfigurasi untuk menggunakan OpenGL berbasis perangkat keras. Didukung dengan sesi virtual saja. Parameter ini tidak didukung dengan server DCV Windows NICE.

Nilai yang valid: benar/salah

Tipe: Boolean

Wajib: Tidak

PermissionsFile

Konten terkode Base64 dari file izin. Default ke server default jika dihilangkan. Untuk informasi selengkapnya, lihat [Mengonfigurasi Otorisasi NICE DCV](#) NICE di Panduan Administrator DCV NICE.

Tipe: String

Wajib: Tidak

EnqueueRequest

Menunjukkan apakah akan mengantri permintaan jika tidak dapat segera dipenuhi.

Jenis: Boolean

Default: salah

Wajib: Tidak

AutorunFile

Didukung dengan sesi konsol pada server Windows NICE DCV dan sesi virtual di server Linux NICE DCV. Ini tidak didukung dengan sesi konsol di server Linux NICE DCV.

Path ke file pada server host yang akan dijalankan di dalam sesi. Jalur file relatif terhadap direktori autorun yang ditentukan untuk parameter konfigurasi `agent . autorun_folder` Agen. Jika file berada di direktori autorun yang ditentukan, tentukan nama file saja. Jika file tidak ada dalam direktori autorun yang ditentukan, tentukan jalur relatifnya. Untuk informasi selengkapnya, lihat [File konfigurasi agen](#) di Panduan Administrator Manajer Sesi NICE DCV.

File dijalankan atas nama Pemilik yang ditentukan. Pemilik yang ditentukan harus memiliki izin untuk menjalankan file di server. Pada server Windows NICE DCV, file dijalankan ketika pemilik log ke sesi. Pada server Linux NICE DCV, file dijalankan saat sesi dibuat.

Tipe: String

Wajib: Tidak

AutorunFileArguments

Didukung dengan sesi virtual pada server Linux NICE DCV. Hal ini tidak didukung dalam sesi konsol pada server Windows dan Linux NICE DCV. Argumen baris perintah diteruskan ke

AutorunFileatas eksekusi di dalam sesi. Argumen dilewatkan dalam urutan mereka muncul ke dalam array yang diberikan. Jumlah argumen maksimum yang diizinkan dan panjang maksimum yang diizinkan dari setiap argumen dapat dikonfigurasi. Untuk informasi selengkapnya, lihat [File konfigurasi broker](#) di Panduan Administrator Manajer Sesi NICE DCV.

Tipe: Array string

Wajib: Tidak

DisableRetryOnFailure

Menunjukkan apakah tidak akan mencoba lagi permintaan sesi buat setelah gagal pada host NICE DCV karena alasan apa pun. Untuk informasi selengkapnya tentang mekanisme coba ulang sesi, lihat [File konfigurasi broker](#) di Panduan Administrator Manajer Sesi NICE DCV.

Jenis: Boolean

Default: salah

Wajib: Tidak

Requirements

Persyaratan yang harus dipenuhi server untuk menempatkan sesi. Persyaratan dapat mencakup tag server dan/atau properti server, baik tag server dan properti server diambil dengan memanggil DescribeServersAPI.

Persyaratan ekspresi kondisi:

- *Sebuah!* = *b* true jika *a* tidak sama dengan *b*
- *a = b* true jika *a* sama dengan *b*
- *a > b* true jika *a* lebih besar dari *b*
- *Sebuah>* = *b* true jika *a* lebih besar dari atau sama dengan *b*
- *a <b* true jika *a* kurang dari *b*
- *a <= b* true jika *a* kurang dari atau sama dengan *b*
- *a = b true jika berisi string b*

Persyaratan operator boolean:

- *a* dan *b* true jika *a* dan *b* benar
- *a* atau *b* true jika *a* atau *b* benar

- tidak *benar jika adalah palsu*

Kunci tag harus diawali oleh tag :, properti server harus diawali server : dengan. Persyaratan ekspresi mendukung kurung ().

Persyaratan contoh:

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

Nilai numerik dapat ditentukan menggunakan notasi eksponensial, misalnya: `"server:Host.Memory.TotalBytes > 1024E6"`.

Properti server yang didukung adalah:

- 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`

Tipe: String

Wajib: Tidak

StorageRoot

Menentukan jalur ke folder yang digunakan untuk penyimpanan sesi. Untuk informasi selengkapnya tentang penyimpanan sesi NICE DCV, lihat [Mengaktifkan Penyimpanan Sesi](#) di Panduan Administrator NICE DCV.

Tipe: String

Wajib: Tidak

Parameter respons

Id

ID unik sesi sesi sesi sesi sesi sesi ini.

Name

Nama sesi sesi.

Owner

Pemilik sesi sesi sesi sesi sesi sesi sesi ini.

Type

Jenis sesi.

State

Keadaan sesi. Jika permintaan selesai berhasil, sesi memasuki CREATING negara.

Substate

Substate sesi sesi sesi sesi sesi sesi sesi sesi sesi ini. Jika permintaan selesai dengan sukses, substate memasuki `SESSION_PLACING` substate.

Contoh

Python

Permintaan

Contoh berikut menciptakan tiga sesi.

```
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),
    ])
]
```

Response

Berikut adalah contoh output.

```
{
    "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

Menjelaskan satu atau lebih server NICE DCV.

Topik

- [Permintaan parameter](#)
- [Parameter respons](#)
- [Contoh](#)

Permintaan parameter

ServerIds

ID dari server NICE DCV untuk menggambarkan. Jika tidak ada ID yang ditentukan, semua server dikembalikan dalam output paginasi.

Tipe: Array string

Wajib: Tidak

NextToken

Token yang akan digunakan untuk mengambil halaman hasil berikutnya.

Tipe: String

Wajib: Tidak

MaxResults

Jumlah hasil maksimum yang akan dikembalikan oleh permintaan dalam output yang diberi nomor halaman. Ketika parameter ini digunakan, permintaan hanya mengembalikan jumlah tertentu hasil dalam satu halaman bersama dengan elemenNextToken respon. Hasil yang tersisa dari permintaan awal dapat dilihat dengan mengirimkan permintaan lain denganNextToken nilai yang dikembalikan.

Kisaran yang valid: 1 - 1000

Default: 1000

Tipe: Integer

Wajib: Tidak

Parameter respons

RequestId

ID unik dari permintaan tersebut.

Servers

Informasi tentang server NICE DCV. Struktur data ini mencakup parameter respons bersarang berikut:

Id

ID unik server DCV NICE DCV.

Ip

Alamat IP server DCV NICE DCV server NICE DCV server NICE DCV server DCV.

Hostname

Nama host server DCV NICE DCV.

Endpoints

Informasi tentang titik akhir server NICE DCV. Struktur data ini mencakup parameter respons bersarang berikut:

IpAddress

Alamat IP titik akhir server server endpoint server endpoint server endpoint server endpoint server endpoint.

Port

Port titik akhir server server akhir server akhir akhir server titik akhir server titik akhir server endpoint.

Protocol

Protokol yang digunakan oleh titik akhir server endpoint server endpoint server endpoint. Nilai yang mungkin termasuk:

- HTTP- Endpoint menggunakan protokol WebSocket (TCP).

- QUIC- Endpoint menggunakan protokol QUIC (UDP).

WebUrlPath

Jalur URL web dari titik akhir server. Tersedia untuk protokol HTTP saja.

Version

Versi server NICE DCV.

SessionManagerAgentVersion

Versi Session Manager Agen berjalan pada server NICE DCV.

Availability

Ketersediaan server NICE DCV. Nilai yang mungkin termasuk:

- AVAILABLE- Server tersedia dan siap untuk penempatan sesi.
- UNAVAILABLE- Server tidak tersedia dan tidak dapat menerima penempatan sesi.

UnavailabilityReason

Alasan tidak tersedianya server NICE DCV. Nilai yang mungkin termasuk:

- SERVER_FULL- Server NICE DCV telah mencapai jumlah maksimum sesi bersamaan yang dapat dijalankan.
- SERVER_CLOSED- Server NICE DCV telah dibuat tidak tersedia menggunakan CloseServerAPI.
- UNREACHABLE_AGENT- Broker Manajer Sesi tidak dapat berkomunikasi dengan Agen Manajer Sesi di server NICE DCV.
- UNHEALTHY_DCV_SERVER- Agen Manajer Sesi tidak dapat berkomunikasi dengan server NICE DCV.
- EXISTING_LOGGED_IN_USER- (Hanya server Windows NICE DCV) Seorang pengguna saat ini masuk ke server NICE DCV menggunakan RDP.
- UNKNOWN— Broker Manajer Sesi tidak dapat menentukan alasannya.

ConsoleSessionCount

Jumlah sesi konsol pada server NICE DCV.

VirtualSessionCount

Jumlah sesi virtual pada server NICE DCV.

Host

Informasi tentang server host tempat server NICE DCV berjalan. Struktur data ini mencakup parameter respons bersarang berikut:

Os

Informasi tentang sistem operasi server host. Struktur data ini mencakup parameter respons bersarang berikut:

Family

Keluarga sistem operasi sistem operasi. Nilai yang mungkin termasuk:

- windows— Server host sedang menjalankan sistem operasi Windows.
- linux- Server host menjalankan sistem operasi Linux.

Name

Nama sistem operasi.

Version

Versi sistem operasi.

KernelVersion

(Hanya Linux) Versi kernel dari sistem operasi.

BuildNumber

(Hanya Windows) Nomor build dari sistem operasi.

Memory

Informasi tentang memori server host. Struktur data ini mencakup parameter respons bersarang berikut:

TotalBytes

Total memori, dalam byte, di server host.

UsedBytes

Memori yang digunakan, dalam byte, pada server host.

Swap

Informasi tentang file swap server host. Struktur data ini mencakup parameter respons bersarang berikut:

TotalBytes

Ukuran file swap total, dalam byte, di server host.

UsedBytes

Ukuran file swap yang digunakan, dalam byte, di server host.

Aws

Hanya untuk server NICE DCV yang berjalan di instans Amazon EC2. AWS-informasi spesifik. Struktur data ini mencakup parameter respons bersarang berikut:

Region

AWSWilayah instans Amazon EC2.

Ec2InstanceType

Tipe instans Amazon EC2 EC2.

Ec2InstanceId

ID instans Amazon EC2 EC2.

Ec2ImageId

ID gambar Amazon EC2 EC2.

CpuInfo

Informasi tentang CPU server host. Struktur data ini mencakup parameter respons bersarang berikut:

Vendor

Vendor CPU server host server server server server host server server server server host server CPU server server server

ModelName

Nama model CPU server host.

Architecture

Arsitektur CPU server host server host.

NumberOfCpus

Jumlah CPU di server host server server host server server host server server host server server host server server host server server

PhysicalCorePerCpu

Jumlah inti CPU per CPU CPU.

CpuLoadAverage

Informasi tentang beban CPU server host. Struktur data ini mencakup parameter respons bersarang berikut:

OneMinute

Rata-rata beban CPU selama periode 1 menit terakhir.

FiveMinutes

Rata-rata beban CPU selama 5 menit terakhir.

FifteenMinutes

Rata-rata beban CPU selama periode 15 menit terakhir.

Gpus

Informasi tentang GPU server host. Struktur data ini mencakup parameter respons bersarang berikut:

Vendor

Vendor GPU server host server server server host server server GPU server host server GPU server host server

ModelName

Nama model GPU server host.

LoggedInUsers

Pengguna yang saat ini login ke server host. Struktur data ini mencakup parameter respons bersarang berikut:

Username

Nama pengguna pengguna yang masuk log in pengguna.

Tags

Tag yang ditugaskan ke server server server. Struktur data ini mencakup parameter respons bersarang berikut:

Key

Kunci tag.

Value

Nilai tag.

Contoh

Python

Permintaan

Contoh berikut menjelaskan semua server NICE DCV yang tersedia. Hasilnya dipaginasi untuk menunjukkan dua hasil per halaman.

```
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)
```

Response

Berikut adalah contoh output.

```
{
  "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

Menjelaskan satu atau lebih sesi NICE DCV.

Topik

- [Permintaan parameter](#)
- [Parameter respons](#)
- [Contoh](#)

Permintaan parameter

SessionIds

ID sesi untuk menggambarkan.

Tipe: String

Wajib: Tidak

NextToken

Token yang akan digunakan untuk mengambil halaman hasil berikutnya.

Tipe: String

Wajib: Tidak

Filters

Filter tambahan untuk diterapkan pada permintaan. Filter yang didukung meliputi:

- Tag:kunci - tag yang ditugaskan untuk sesi sesi sesi sesi sesi sesi ini.
- Pemilik—pemilik sesi.

Tipe: String

Wajib: Tidak

Parameter respons

Id

ID unik sesi sesi sesi sesi sesi ini.

Name

Nama sesi.

Owner

Pemilik sesi sesi sesi ini.

Server

Informasi tentang server tempat sesi berjalan. Struktur data ini mencakup parameter respons bersarang berikut:

Ip

Alamat IP host server NICE DCV.

Hostname

Hostname dari host server NICE DCV.

Port

Port di mana server NICE DCV berkomunikasi dengan klien NICE DCV.

Endpoints

Informasi tentang titik akhir server NICE DCV. Struktur data ini mencakup parameter respons bersarang berikut:

IpAddress

Alamat IP titik akhir server server endpoint server endpoint server endpoint server endpoint server endpoint.

Port

Port titik akhir server server akhir server akhir akhir server titik akhir server titik akhir server endpoint.

Protocol

Protokol yang digunakan oleh titik akhir server endpoint server endpoint server endpoint. Nilai yang mungkin termasuk:

- HTTP- Endpoint menggunakan protokol WebSocket (TCP).
- QUIC- Endpoint menggunakan protokol QUIC (UDP).

WebUrlPath

Jalur URL web dari titik akhir server. Tersedia untuk protokol HTTP saja.

Tags

Tag yang ditugaskan ke server server server. Struktur data ini mencakup parameter respons bersarang berikut:

Key

Kunci tag.

Value

Nilai tag.

Type

Jenis sesi.

State

Status sesi saat ini sesi sesi sesi sesi sesi sesi sesi sesi sesi ini. Nilai yang mungkin adalah:

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

Substate

Substate sesi saat ini. Nilai yang mungkin adalah:

- SESSION_PLACING- sesi sedang menunggu untuk ditempatkan pada Server DCV yang tersedia.
- PENDING_PREPARATION- sesi dibuat tetapi tidak dapat digunakan; terkait dengan Server DCV.

CreationTime

Tanggal dan waktu saat sesi dibuat.

LastDisconnectionTime

Tanggal dan waktu pemutusan klien terakhir.

NumOfConnections

Jumlah koneksi klien aktif klien.

StorageRoot

Menentukan jalur ke folder yang digunakan untuk penyimpanan sesi. Untuk informasi selengkapnya tentang penyimpanan sesi NICE DCV, lihat [Mengaktifkan Penyimpanan Sesi](#) di Panduan Administrator NICE DCV.

Tipe: String

Wajib: Tidak

Contoh

Python

Permintaan

Contoh berikut menjelaskan sesi yang dimiliki oleh `user1` dan memiliki `tagos=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'}])
```

Response

Berikut adalah contoh output.

```
{
  "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

Menghapus sesi NICE DCV yang ditentukan dan menghapusnya dari cache Broker.

Topik

- [Permintaan parameter](#)
- [Parameter respons](#)
- [Contoh](#)

Permintaan parameter

SessionId

ID sesi yang akan dihapus.

Tipe: String

Wajib: Ya

Owner

Pemilik sesi yang akan dihapus.

Tipe: String

Wajib: Ya

Force

Menghapus sesi dari cache Broker dengan mencoba menghapusnya dari server NICE DCV. Ini berguna untuk menghapus sesi usang dari cache Broker. Misalnya, jika server NICE DCV dihentikan, tetapi sesi masih terdaftar di Broker, gunakan flag ini untuk membersihkan sesi dari cache Broker.

Perlu diingat bahwa jika sesi masih aktif, itu di-cache ulang oleh Broker.

Nilai yang valid: `true` | `false`

Tipe: Boolean

Wajib: Tidak

Parameter respons

SessionId

ID sesi

State

Hanya dikembalikan jika sesi berhasil dihapus. Menunjukkan status sesi saat ini. Jika permintaan selesai berhasil, sesi transisi keDELETING negara. Perlu waktu beberapa menit sampai sesi dihapus. Ketika telah dihapus, transisi negara dariDELETING keDELETED.

FailureReason

Hanya dikembalikan jika beberapa sesi tidak dapat dihapus. Menunjukkan mengapa sesi tidak dapat dihapus.

Contoh

Python

Permintaan

Contoh berikut menghapus dua sesi — sesi dengan IDSessionId123 yang dimiliki olehuser1, dan sesi dengan IDSessionIdabc yang dimiliki olehuser99.

```
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')])
```

Response

Berikut adalah contoh output. SessionId123 telah berhasil dihapus, sementara tidak SessionIdabc dapat dihapus.

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

GetSessionConnectionData

Mendapat informasi koneksi untuk koneksi pengguna tertentu ke sesi NICE DCV tertentu.

Topik

- [Permintaan parameter](#)
- [Parameter respons](#)
- [Info tambahan](#)
- [Contoh](#)

Permintaan parameter

SessionId

ID sesi untuk melihat informasi koneksi.

Tipe: String

Wajib: Ya

User

Nama pengguna untuk melihat informasi koneksi.

Tipe: String

Wajib: Ya

Parameter respons

Id

ID unik sesi sesi sesi sesi sesi ini.

Name

Nama sesi.

Owner

Pemilik sesi sesi sesi ini.

Server

Informasi tentang server tempat sesi berjalan. Struktur data ini mencakup parameter respons bersarang berikut:

Ip

Alamat IP host server NICE DCV.

Hostname

Hostname dari host server NICE DCV.

Port

Port di mana server NICE DCV berkomunikasi dengan klien NICE DCV.

Endpoints

Informasi tentang titik akhir server NICE DCV. Struktur data ini mencakup parameter respons bersarang berikut:

IpAddress

Alamat IP titik akhir server server endpoint server endpoint server endpoint server endpoint server endpoint.

Port

Port titik akhir server server akhir server akhir akhir server titik akhir server titik akhir server endpoint.

Protocol

Protokol yang digunakan oleh endpoint server. Nilai yang mungkin termasuk:

- HTTP- Endpoint menggunakan protokol WebSocket (TCP).
- QUIC- Endpoint menggunakan protokol QUIC (UDP).

WebUrlPath

Jalur URL web dari titik akhir server. Tersedia untuk protokol HTTP saja.

WebUrlPath

Jalur menuju file konfigurasi server DCV NICE DCV.

Tags

Tag yang ditugaskan ke server server server. Struktur data ini mencakup parameter respons bersarang berikut:

Key

Kunci tag.

Value

Nilai tag.

Type

Jenis sesi.

State

Status sesi saat ini sesi sesi sesi sesi sesi sesi sesi sesi sesi ini. Nilai yang mungkin adalah:

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

CreationTime

Tanggal dan waktu saat sesi dibuat.

LastDisconnectionTime

Tanggal dan waktu pemutusan klien terakhir.

NumOfConnections

Jumlah koneksi bersamaan pengguna memiliki sesi.

ConnectionToken

Token otentikasi yang digunakan untuk menyambung ke sesi.

Info tambahan

Informasi yang diperoleh dari API ini dapat diteruskan ke klien NICE DCV untuk terhubung ke sesi NICE DCV.

Dalam kasus klien Web NICE DCV, Anda dapat membuat URL yang dapat dibuka di browser. URL memiliki format berikut:

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

Dalam kasus klien asli NICE DCV, Anda dapat membangun URL dengandcv:// skema. Ketika klien asli NICE DCV diinstal, ia mendaftarkan sendiri dengan sistem sebagai penanganan untukdcv:// URL. URL memiliki format berikut:

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

Note

Jika Anda menggunakan Amazon EC2, alamat IP haruslah alamat IP publik. Jika konfigurasi Anda memiliki host NICE DCV di belakang gateway, tentukan alamat gateway daripada yang dikembalikan oleh SessionConnectionData API.

Contoh

Python

Permintaan

Contoh berikut mendapat informasi koneksi untuk pengguna dengan nama pengguna `user1` dan sesi dengan ID `sessionId12345`.

```
def get_session_connection_api():
    api_instance =
    swagger_client.SessionConnectionDataApi(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')
```

Response

Berikut adalah contoh output.

```
{
  "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":
"EXAMPLEi0iJm0WM1YTRhZi1jZmU0LTQ0ZjEtYjZlOC04ZjY0YjM4ZTE2ZDkiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUz
tngiKXevUxhhJm3BPJYRs9NPE4GCJRTc13EXAMPLEIxNEPPh5IMcVmR0fU1WKPnry4ypPTp3rsZ7YWjCTSfs1GoN3R_
Kqtpd5GH0D-E8FwsedV-
Q2bRQ4y9y1q0MgFU4QjaSMypUuYR0YjkCaoainjmEZew4A33fG40wATrBvoivBiNwdNpytHX2CD0uk_k0k_DWeZjMvv9
h_GaMgHmltqBIA4jdPD7i0CmC2e7413KFy-
EQ4Ej1cM7RjLwhFuWpKWAVJxogJjYpfoKkaPo4KxvJjJIPYhkscklINQpe2W5rnLxCq7sC7ptcGw17DUobP7egRv9H37
hK1G4G8erHv19HIrTR9_c884fNrTCC8DvC062e4KYdLkAhhJmboN9CAGIGFyd2c1AY_CzzvDL0EXAMLE"
}

```

GetSessionScreenshots

Mendapat screenshot dari satu atau lebih sesi NICE DCV.

Jenis file gambar dan resolusi tangkapan layar tergantung pada konfigurasi Broker Manajer Sesi. Untuk memodifikasi jenis file gambar, konfigurasi `session-screenshot-format` parameter. Untuk memodifikasi resolusi, konfigurasi `session-screenshot-max-width` dan `session-screenshot-max-height` parameter. Untuk informasi selengkapnya, lihat [File konfigurasi broker](#) di Panduan Administrator Manajer Sesi NICE DCV.

Topik

- [Permintaan parameter](#)

- [Parameter respons](#)
- [Contoh](#)

Permintaan parameter

SessionId

ID sesi NICE DCV untuk mendapatkan tangkapan layar.

Tipe: String

Wajib: Ya

Parameter respons

RequestId

ID unik dari permintaan tersebut.

SuccessfulList

Informasi tentang screenshot sukses. Struktur data ini mencakup parameter respons bersarang berikut:

SessionScreenshot

Informasi tentang tangkapan layar. Struktur data ini mencakup parameter respons bersarang berikut:

SessionId

ID sesi NICE DCV dari mana tangkapan layar diambil.

Images

Informasi tentang gambar. Struktur data ini mencakup parameter respons bersarang berikut:

Format

Format gambar. Nilai yang mungkin termasuk: jpeg dan png.

Data

Gambar screenshot base64 format dikodekan.

CreationTime

Tanggal dan waktu saat tangkapan layar diambil.

Primary

Menunjukkan apakah tangkapan layar adalah tampilan utama sesi NICE DCV.

UnsuccessfulList

Informasi tentang tangkapan layar yang gagal. Struktur data ini mencakup parameter respons bersarang berikut:

GetSessionScreenshotRequestData

Permintaan asli yang gagal.

SessionId

ID sesi NICE DCV dari mana tangkapan layar akan diambil.

FailureReason

Sebab kegagalan.

Contoh

Python

Permintaan

Contoh berikut mendapat screenshot dari dua sesi (`sessionId1` dan `sessionId2`).
`sessionId2` tidak ada dan mengakibatkan kegagalan.

```
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"])
```

Response

Berikut adalah contoh output.

```
{
  "RequestId": "542735ef-f6ab-47d8-90e5-23df31d8d166",
  "SuccessfulList": [
    {
      "SessionScreenshot": {
        "SessionId": "sessionId1",
        "Images": [
          {
            "Format": "png",
            "Data": "iVBORw0KGgoAAAANSUhEUgAAAEEXAMPLE",
            "CreationTime": "2021-03-30T15:47:06.822Z",
            "Primary": true
          }
        ]
      }
    }
  ],
  "UnsuccessfulList": [
    {
      "GetSessionScreenshotRequestData": {
        "SessionId": "sessionId2"
      },
      "FailureReason": "Dcv session not found."
    }
  ]
}
```

OpenServers

Membuka satu atau lebih server NICE DCV. Sebelum Anda dapat membuat sesi NICE DCV pada server NICE DCV, Anda harus mengubah status server untuk membuka. Setelah server NICE DCV terbuka, Anda dapat membuat sesi NICE DCV di server.

Topik

- [Permintaan parameter](#)
- [Parameter respons](#)
- [Contoh](#)

Permintaan parameter

ServerId

ID server untuk membuka.

Tipe: String

Wajib: Ya

Parameter respons

RequestId

ID unik dari permintaan tersebut.

SuccessfulList

Informasi tentang server NICE DCV yang berhasil dibuka. Struktur data ini mencakup parameter respons bersarang berikut:

ServerId

ID server yang berhasil dibuka.

UnsuccessfulList

Informasi tentang server NICE DCV yang tidak dapat dibuka. Struktur data ini mencakup parameter respons bersarang berikut:

OpenServerRequestData

Informasi tentang permintaan asli yang gagal. Struktur data ini mencakup parameter respons bersarang berikut:

ServerId

ID server NICE DCV yang tidak dapat dibuka.

FailureCode

Kode kegagalan.

FailureReason

Sebab kegagalan.

Contoh

Python

Permintaan

Contoh berikut membuka dua server NICE DCV (`serverId1` dan `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"])
```

Response

Berikut adalah contoh output.

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

UpdateSessionPermissions

Memperbarui izin pengguna untuk sesi NICE DCV tertentu.

Topik

- [Permintaan parameter](#)
- [Parameter respons](#)
- [Contoh](#)

Permintaan parameter

SessionId

ID sesi untuk mengubah izin.

Tipe: String

Wajib: Ya

Owner

Pemilik sesi untuk mengubah izin.

Tipe: String

Wajib: Ya

PermissionFile

Konten terkode Base64 dari file izin yang akan digunakan. Untuk informasi selengkapnya, lihat [Mengonfigurasi Otorisasi NICE DCV NICE](#) di Panduan Administrator DCV NICE.

Tipe: String

Wajib: Ya

Parameter respons

SessionId

ID sesi ini.

Contoh

Python

Permintaan

Contoh berikut menetapkan izin baru untuk sesi dengan ID sesi `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_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']])
```

Response

Berikut adalah contoh output.

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

Catatan rilis dan riwayat dokumen untuk Manajer Sesi NICE DCV

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

Topik

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

Catatan rilis Manajer Sesi DCV NICE

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

Topik

- [2023.1— November 9, 2023](#)
- [2023.0-15065— 4 Mei 2023](#)
- [2023.0-14852— 28 Maret 2023](#)
- [2022.2-13907— November 11, 2022](#)
- [2022.1-13067— 29 Juni 2022](#)
- [2022.0-11952— 23 Februari 2022](#)
- [2021.3-11591— Desember 20, 2021](#)
- [2021.2-11445— November 18, 2021](#)
- [2021.2-11190— 11 Oktober 2021](#)
- [2021.2-11042— September 01, 2021](#)
- [2021.1-10557— 31 Mei 2021](#)
- [2021.0-10242— 12 April 2021](#)
- [2020.2-9662— 04 Desember 2020](#)
- [2020.2-9508— 11 November 2020](#)

2023.1— November 9, 2023

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

2023.0-15065— 4 Mei 2023

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

2023.0-14852— 28 Maret 2023

Bangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> • Pialang: 392 • Agen: 642 • CLI: 132 	<ul style="list-style-type: none"> • Menambahkan dukungan untuk Red Hat Enterprise Linux 9, Rocky Linux 9, dan CentOS Stream 9.

2022.2-13907— November 11, 2022

Bangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> • Pialang: 382 • Agen: 612 • CLI: 123 	<ul style="list-style-type: none"> • Menambahkan Substate bidang DescribeSessions sebagai tanggapan.

Bangun angka	Perubahan dan perbaikan bug
	<ul style="list-style-type: none"> • Memperbaiki masalah yang dapat menyebabkan CLI gagal terhubung ke broker tergantung pada URL yang digunakan.

2022.1-13067— 29 Juni 2022

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

2022.0-11952— 23 Februari 2022

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

2021.3-11591— Desember 20, 2021

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

2021.2-11445— November 18, 2021

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

2021.2-11190— 11 Oktober 2021

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

2021.2-11042— September 01, 2021

Bangun angka	Fitur baru	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> • Pialang: 254 • Agen: 413 • CLI: 37 	<ul style="list-style-type: none"> • NICE DCV Session Manager sekarang menawarkan dukungan antarmuka baris perintah (CLI). Anda dapat membuat dan mengelola sesi NICE DCV di CLI, alih-alih memanggil API. • Manajer Sesi NICE DCV memperkenalkan ketekunan data Broker. Untuk ketersediaan yang lebih tinggi, broker dapat mempertahankan informasi status server pada penyimpanan data 	<ul style="list-style-type: none"> • Saat mendaftarkan server otorisasi eksternal, Anda sekarang dapat menentukan algoritme yang digunakan server otorisasi untuk menandatangani Token Web berformat JSON. Dengan perubahan ini, Anda dapat menggunakan Azure AD sebagai server otorisasi eksternal.

Bangun angka	Fitur baru	Perubahan dan perbaikan bug
	eksternal dan memulihkan data saat startup.	

2021.1-10557— 31 Mei 2021

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

2021.0-10242— 12 April 2021

Bangun angka	Perubahan dan perbaikan bug
<ul style="list-style-type: none"> Pialang: 183 Agen: 318 	<ul style="list-style-type: none"> Manajer Sesi NICE DCV memperkenalkan API baru berikut: <ul style="list-style-type: none"> OpenServers CloseServers DescribeServers GetSessionScreenshots Ini juga memperkenalkan parameter konfigurasi baru berikut: <ul style="list-style-type: none"> Parameter broker: <code>session-screenshot-max-width</code>, <code>session-screenshot-max-height</code>, <code>session-screenshot-format</code>, <code>create-sessions-queue-max-size</code>, dan <code>create-sessions-queue-max-time-seconds</code>.

Bangun angka	Perubahan dan perbaikan bug
	<ul style="list-style-type: none"> • Parameter agen:<code>agent.autorun_folder</code> ,<code>max_virtual_sessions</code> , dan <code>max_concurrent_sessions_per_user</code> . <p>Parameter agen:<code>agent.autorun_folder</code> ,<code>max_virtual_sessions</code> , dan <code>max_concurrent_sessions_per_user</code> .</p> <p>Parameter agen:<code>agent.autorun_folder</code> ,<code>max_virtual_sessions</code> , dan <code>max_concurrent_sessions_per_user</code> .</p>

2020.2-9662— 04 Desember 2020

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

2020.2-9508— 11 November 2020

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

Riwayat dokumen

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

Perubahan	Deskripsi	Tanggal
NICE DCV Versi 2023.1	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2023.1. Untuk informasi selengkapnya, lihat 2023.1—November 9, 2023 .	November 9, 2023
NICE DCV Versi 2023.0	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2023.0. Untuk informasi selengkapnya, lihat 2023.0-14852—28 Maret 2023 .	Maret 28, 2023
NICE DCV Versi 2022.2	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2022.2. Untuk informasi selengkapnya, lihat 2022.2-13907—November 11, 2022 .	November 11, 2022
NICE DCV Versi 2022.1	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2022.1. Untuk informasi selengkapnya, lihat 2022.1-13067—29 Juni 2022 .	Juni 29, 2022
NICE DCV Versi 2022.0	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2022.0. Untuk informasi selengkapnya, lihat 2022.0-11952—23 Februari 2022 .	Februari 23, 2022
NICE DCV Versi 2021.3	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2021.3. Untuk informasi selengkapnya, lihat 2021.3-11591—Desember 20, 2021 .	Desember 20, 2021
NICE DCV Versi 2021.2	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2021.2. Untuk informasi selengkapnya, lihat 2021.2-11042—September 01, 2021 .	September 01, 2021

Perubahan	Deskripsi	Tanggal
NICE DCV Versi 2021.1	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2021.1. Untuk informasi selengkapnya, lihat 2021.1-10 557— 31 Mei 2021 .	Mei 31, 2021
NICE DCV Versi 2021.0	Manajer Sesi NICE DCV telah diperbarui untuk NICE DCV 2021.0. Untuk informasi selengkapnya, lihat 2021.0-10 242— 12 April 2021 .	12 April 2021
Rilis Awal Session Manager NICE DCV	Publikasi pertama dari konten ini.	11 November 2020

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