



관리자 안내서

NICE DCV



NICE DCV: 관리자 안내서

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

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 관련하여 고객에게 혼동을 일으킬 수 있는 방식이나 Amazon 브랜드 이미지를 떨어뜨리는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon 계열사, 관련 업체 또는 Amazon의 지원 업체 여부에 상관없이 해당 소유자의 자산입니다.

Table of Contents

NICE DCV란 무엇인가요?	1
NICE DCV의 작동 원리	1
기능	1
요금	2
NICE DCV 서버	3
요구 사항	3
지원되는 기능	5
설정	7
설치	7
Windows	7
Linux	14
라이선싱	50
NICE DCV 라이선스 요구 사항	51
확장 평가판 라이선스 설치	54
프로덕션 라이선스 설치	55
프로덕션 라이선스 업데이트	67
업그레이드	67
호환성 고려 사항	67
Windows에서 NICE DCV 서버 업그레이드	68
Linux에서 NICE DCV 서버 업그레이드	68
설치 제거	69
Windows에서 NICE DCV 서버 제거	69
Linux에서 NICE DCV 서버 제거	70
이미징	71
이미지 구축	71
이미지 파이프라인에 추가	72
서버 관리	73
서버 시작	73
서버 중지	75
QUIC UDP 활성화	76
TCP/UDP 포트 및 주소 변경	78
서버 TCP/UDP 포트 변경	79
특정 엔드포인트에서 수신	81
TLS 인증서 관리	83

유휴 클라이언트 연결 해제	84
Linux에서 GPU 공유 활성화	87
터치스크린 및 스타일러스 지원 활성화	89
스타일러스 구성	91
게임패드 지원 활성화	92
Xbox 360 컨트롤러 지원	92
USB 원격화 활성화	93
스마트 카드 캐싱 구성	95
WebAuthn 리디렉션 구성	96
리디렉션 브라우저 확장 설정 WebAuthn	97
세션 스토리지 활성화	99
Windows에서 세션 스토리지 활성화	99
Linux에서 세션 스토리지 활성화	100
Linux에서 프린터 설정	101
프린터 문제 해결	103
Linux에서 클립보드 구성	104
기본 선택 항목에 클라이언트 클립보드 콘텐츠 붙여 넣기	104
클라이언트 클립보드에 기본 선택 내용 복사	105
다중 채널 오디오 구성	105
Windows NICE DCV 서버의 오디오 채널 구성	107
Linux NICE DCV 서버의 오디오 채널 구성	107
HTTP 헤더 구성	109
Windows NICE DCV 서버에서 HTTP 헤더 구성	109
Linux NICE DCV 서버에서 HTTP 헤더 구성	110
인증 구성	110
Windows에서 인증 구성	111
Linux에서 인증 구성	112
외부 인증자를 사용한 권한 부여 구성	114
권한 부여 구성	114
기본 권한 파일	114
사용자 지정 권한 파일	114
권한 파일 작업	115
세션 관리	121
NICE DCV 세션 소개	121
콘솔 세션	122
가상 세션	122

명령줄 도구를 사용하여 세션 관리	123
Windows NICE DCV 서버에서 명령줄 도구 사용	123
Linux NICE DCV 서버에서 명령줄 도구 사용	124
명령줄 도구 사용	124
세션 시작	126
콘솔 및 가상 세션 수동 시작	126
자동 콘솔 세션 활성화	131
세션 중지	132
명령문	133
예	133
실행 중인 세션 관리	133
세션 스토리지 관리	133
세션 권한 부여 관리	135
세션 표시 레이아웃 관리	136
세션 이름 관리	139
세션 시간대 관리	141
세션 보기	142
모든 활성 세션 나열	142
특정 세션에 대한 정보 보기	142
스크린샷 찍기	144
명령문	144
옵션	144
예	146
방법	148
외부 인증 사용	148
NICE DCV 서버 구성	149
토큰 사용	150
인증 서비스 요구 사항	150
유휴 세션 찾기 및 중지	152
X 서버로의 원격 X 연결 활성화	153
X 서버로의 원격 X 연결 활성화	154
NICE DCV 웹 브라우저 클라이언트를 iFrame 안에 임베드	155
문제 해결	158
로그 파일 사용	158
Windows에서 로그 파일 세부 수준 변경	159
Linux에서 로그 파일 세부 수준 변경	159

Linux에서의 가상 세션 생성 문제 해결	160
Linux에서 가상 세션 생성 실패 조사	160
Linux에서 파일세이프 가상 세션 생성	161
UID 변경 후 Linux 세션이 시작되지 않음	163
Windows에서의 커서 문제 해결	163
IntelliJ IDEA에 복사 및 붙여넣기 문제 해결	163
자체 서명된 인증서에 대한 리디렉션 설명	164
Windows에서 NVIDIA GPU 사용 시 다중 모니터/전체 화면 오류	165
NICE DCV 성능 및 통계 모니터링	166
카운터 세트	167
파라미터 참조	175
audio 파라미터	176
clipboard 파라미터	177
connectivity 파라미터	180
display 파라미터	187
display/linux 파라미터	194
input 파라미터	195
license 파라미터	196
log 파라미터	198
printer 파라미터	202
redirection 파라미터	203
security 파라미터	205
session-management 파라미터	214
session-management/automatic-console-session 파라미터	218
session-management/defaults 파라미터	220
smartcard 파라미터	221
webcam 파라미터	222
windows 파라미터	223
구성 파라미터 수정	224
Windows NICE DCV 서버	224
Linux NICE DCV 서버	225
NICE DCV 지원 기간 종료	227
EOSL 타임라인	227
고객을 위한 EOSL 경로	228
EOSL FAQ	228
보안	230

데이터 보호	230
데이터 암호화	231
규정 준수 검증	232
릴리스 정보 및 문서 기록	233
릴리스 정보	233
NICE DCV 2023.1-16388	235
NICE DCV 2023.1-16388	236
NICE DCV 2023.1-16220	237
NICE DCV 2023.0-15487	239
NICE DCV 2023.0-15065	240
NICE DCV 2023.0-15022	241
NICE DCV 2023.0-14852	243
NICE DCV 2022.2-14521	244
NICE DCV 2022.2-14357	245
NICE DCV 2022.2-14175	245
NICE DCV 2022.2-14126	246
NICE DCV 2022.2-13907	247
NICE DCV 2022.1-13300	248
NICE DCV 2022.1-13216	249
NICE DCV 2022.1-13067	249
NICE DCV 2022.0-12760	250
NICE DCV 2022.0-12627	251
NICE DCV 2022.0-12123	251
NICE DCV 2022.0-11954	252
NICE DCV 2021.3-11591	254
NICE DCV 2021.2-11445	254
NICE DCV 2021.2-11190	255
NICE DCV 2021.2-11135	255
NICE DCV 2021.2-11048	256
DCV 2021.1-10851	258
DCV 2021.1-10598	259
DCV 2021.1-10557	259
DCV 2021.0-10242	260
DCV 2020.2-9662	261
DCV 2020.2-9508	261
DCV 2020.1-9012	262

DCV 2020.1-9012	263
DCV 2020.1-8942	263
DCV 2020.0-8428	264
DCV 2019.1-7644	266
DCV 2019.1-7423	266
DCV 2019.0-7318	267
DCV 2017.4-6898	268
DCV 2017.3-6698	269
DCV 2017.2-6182	271
DCV 2017.1-5870	273
DCV 2017.1-5777	274
DCV 2017.0-5600	274
DCV 2017.0-5121	275
DCV 2017.0-4334	275
DCV 2017.0-4100	276
사용 설명서 기록	276
.....	cclxxxii

NICE DCV란 무엇인가요?

NICE DCV는 고성능 원격 디스플레이 프로토콜입니다. 이를 통해 다양한 네트워크 조건에서 클라우드 또는 데이터 센터에서 모든 디바이스로 원격 데스크톱 및 애플리케이션 스트리밍을 안전하게 제공할 수 있습니다. Amazon EC2와 함께 NICE DCV를 사용하면 그래픽 집약적 애플리케이션을 Amazon EC2 인스턴스에서 원격으로 실행할 수 있습니다. 그런 다음 결과를 적절한 클라이언트 머신으로 스트리밍할 수 있으므로 비용이 많이 드는 전용 워크스테이션이 필요하지 않습니다.

주제

- [NICE DCV의 작동 원리](#)
- [NICE DCV의 기능](#)
- [NICE DCV 가격](#)

NICE DCV의 작동 원리

NICE DCV를 사용하려면 NICE DCV 서버 소프트웨어를 서버에 설치합니다. NICE DCV 서버 소프트웨어는 보안 [세션](#)을 생성하는 데 사용됩니다. 서버에 애플리케이션을 설치하고 실행합니다. 서버는 하드웨어를 사용하여 설치된 애플리케이션이 요구하는 고성능 처리를 수행합니다. 사용자는 NICE DCV 클라이언트 애플리케이션을 사용해 세션에 원격으로 연결하여 애플리케이션에 액세스합니다. 연결이 설정되면 NICE DCV 서버 소프트웨어는 애플리케이션의 시각적 출력을 압축한 다음 이를 암호화된 픽셀 스트림으로 다시 스트리밍합니다. 클라이언트 애플리케이션은 이렇게 압축된 픽셀 스트림을 받아 해독한 다음 로컬 디스플레이에 출력합니다.

NICE DCV의 기능

NICE DCV는 다음 기능을 제공합니다.

- 전체 데스크톱 공유 - 고성능 NICE DCV 프로토콜을 사용하여 전체 원격 데스크톱에 대한 완전한 제어를 공유합니다.
- 이미지만 전송 - 렌더링된 이미지를 지오메트리 및 장면 정보 대신 픽셀로 전송합니다. 네트워크를 통해 전송되는 독점적 고객 정보가 없으므로 추가 보안 계층을 제공합니다.
- H.264 기반 인코딩 지원 - H.264 기반 비디오 압축 및 인코딩을 사용하여 대역폭 소비를 줄입니다.
- 무손실 품질 비디오 압축 지원 - 네트워크 및 프로세서 상태가 허용하는 경우 무손실 품질 비디오 압축을 지원합니다.

- 디스플레이 레이아웃 일치 - 서버의 화면 해상도와 디스플레이 레이아웃을 클라이언트 창 크기에 맞게 자동으로 조정합니다.
- 멀티스크린 지원 - 세션 데스크톱을 최대 4대의 모니터로 확장할 수 있습니다. Windows 및 macOS 용 네이티브 클라이언트에서는 픽셀 밀도가 높은 모니터가 지원됩니다.
- 압축 수준 조정 - 네트워크의 가용 대역폭과 지연 시간을 기반으로 비디오 압축 수준을 자동으로 조정합니다.
- 공동 작업 지원 - 여러 공동 작업 클라이언트를 지원하는 동적 세션을 제공합니다. 클라이언트는 세션 중에 언제든지 연결 및 연결 해제할 수 있습니다.
- 서버당 다중 세션 지원(Linux NICE DCV 서버만 해당) - Linux NICE DCV 서버당 여러 개의 가상 세션을 지원하여 비용 절감을 극대화합니다.
- GPU 공유 지원(Linux NICE DCV 서버만 해당) - Linux NICE DCV 서버에서 실행되는 여러 가상 세션 간에 하나 이상의 물리적 GPU를 공유할 수 있습니다.
- 터치 입력, 스타일러스 입력 및 게임패드 지원 - 로컬 컴퓨터에 연결된 입력 디바이스를 사용하여 원격 NICE DCV 세션과 상호 작용할 수 있습니다.
- WebAuthn, 스마트 카드, 스타일러스 및 USB 원격화 지원 - 로컬 컴퓨터에서처럼 NICE DCV 세션에서 주변 디바이스를 사용할 수 있습니다.
- 오디오 입/출력, 인쇄, 복사 및 붙여넣기 지원 - 세션과 로컬 컴퓨터 간에 이러한 주요 작업을 수행할 수 있습니다.
- 파일 전송 지원 - 세션과 로컬 컴퓨터 간에 파일을 전송할 수 있습니다.
- HTML5 클라이언트 제공 - Windows 및 Linux의 최신 웹 브라우저에서 사용할 수 있는 HTML5 클라이언트를 제공합니다.
- 최신 Linux 데스크톱 환경 지원 - RHEL 8의 Gnome 3과 같은 최신 Linux 데스크톱을 지원합니다.

NICE DCV 가격

Amazon EC2 인스턴스에서 NICE DCV 서버를 사용할 경우 발생하는 추가 요금은 없습니다. 사용하는 인스턴스와 기타 Amazon EC2 기능에 대한 표준 요금만 지불하면 됩니다.

그렇지 않은 경우 라이선스가 있어야 합니다. 자세한 내용은 [NICE DCV 서버 라이선스](#) 섹션을 참조하세요.

NICE DCV 서버

NICE DCV 서버는 Windows 및 Linux에서 사용할 수 있습니다. 두 서버 모두 유사한 기능을 제공하지만 몇 가지 차이점이 있습니다. 요건에 가장 적합한 NICE DCV 서버를 선택합니다. 다음 표에서는 Windows 및 Linux NICE DCV 서버에서 지원하는 기능을 비교합니다.

주제

- [요구 사항](#)
- [지원되는 기능](#)

요구 사항

NICE DCV에서 양질의 사용자 경험을 제공하려면 서버가 다음과 같은 최소 요구 사항을 충족해야 합니다. 사용자 경험은 NICE DCV 서버에서 NICE DCV 클라이언트로 스트리밍되는 픽셀 수에 좌우된다는 점에 유의하세요.

Amazon EC2 인스턴스에 NICE DCV 서버를 설치하는 경우 Amazon EC2 G3, G4dn, G4ad, 또는 G5 인스턴스 유형을 사용하는 것이 좋습니다. 이러한 인스턴스 유형은 하드웨어 기반 OpenGL 및 GPU 공유를 지원하는 GPU를 제공합니다. 자세한 내용은 [Amazon EC2 G3 인스턴스](#), [Amazon EC2 G4 인스턴스](#), [Amazon EC2 G5 인스턴스](#)를 참조하세요.

다른 인스턴스 유형에도 NICE DCV 서버를 설치할 수 있지만, 화면 해상도 제한이 있을 수 있습니다. Windows Server 2016에서 이러한 제한을 우회하려면 [EC2용 NICE DCV 가상 디스플레이 드라이버](#)를 다운로드하여 설치하세요. DCV 2023.1 이상을 실행하는 Windows Server 2019 이상에서는 추가 조치가 필요하지 않습니다.

서버는 다음 표에 나와 있는 최소 요구 사항을 충족해야 합니다.

	Windows 서버	Linux 서버
운영 체제	<ul style="list-style-type: none"> • Windows 10 • Windows 11 • Windows Server 2016 • Windows Server 2019 • Windows Server 2022 	<ul style="list-style-type: none"> • Amazon Linux 2 • CentOS 7.6 이상 • CentOS Stream 8 • CentOS Stream 9 • RHEL 7.6 이상 • RHEL 8.x

	Windows 서버	Linux 서버
	<p>Note</p> <p>지원되는 모든 Windows 운영 체제에 .NET Framework 4.5가 필요하고 x86-64 아키텍처를 지원해야 합니다.</p>	<ul style="list-style-type: none"> • RHEL 9.x • SUSE Linux Enterprise 12 SP5 이상 • SUSE Linux Enterprise 15 SP5 • Rocky Linux 8.5 이상 • Rocky Linux 9 • Ubuntu 20.04 • Ubuntu 22.04
지원되는 아키텍처	64비트 x86	<ul style="list-style-type: none"> • 64비트 x86 • 64비트 ARM(Amazon Linux 2, RHEL 7.x/8.x 및 Ubuntu 22.04를 실행하는 Amazon EC2 인스턴스에서만 지원됨)
GPU	<p>(선택 사항) 하드웨어 기반 비디오 인코딩에는 NVIDIA 또는 AMD GPU가 필요합니다. 서버에 GPU가 없는 경우 소프트웨어 기반 비디오 인코딩이 사용됩니다.</p> <p>Note</p> <ul style="list-style-type: none"> • NVIDIA GPU에는 하드웨어 기반 비디오 인코딩을 위한 NVENC가 필요합니다. 컴퓨팅 기능이 3.5 >=인 NVIDIA GPU가 필요합니다. • AMD GPU에는 하드웨어 기반 비디오 인코딩을 위해 Linux 또는 Windows용 Advanced Media Framework(AMF) 또는 Windows용 Rapidfire가 필요합니다. Linux의 경우 AMD 드라이버에서 제공하는 amf-amdgpu-pro 추가 패키지를 설치하여 Ubuntu 인스턴스에서 AMF 인코더를 사용할 수 있습니다. <p>가상 세션에서 GPU 공유 시 NVIDIA GPU가 필요합니다.</p> <p>Note</p> <p>AMD GPU가 설치된 Linux 서버에서는 콘솔 세션만 지원됩니다.</p>	

	Windows 서버	Linux 서버
네트워크	기본적으로 NICE DCV 서버는 포트 8443을 통해 통신합니다. 포트는 구성 가능하지만 1024보다 커야 합니다. 서버가 필수 포트를 통한 통신을 허용하는지 확인합니다.	

Note

NICE DCV는 수명 주기가 끝난 운영 체제는 지원하지 않습니다. 사용 중인 운영 체제에 대해서는 공급업체에 문의하세요.

NICE DCV 클라이언트 요구 사항에 대한 자세한 내용은 NICE DCV 사용 설명서의 [NICE DCV 클라이언트 요구 사항](#)을 참조하세요.

지원되는 기능

다음 표에서는 Windows 및 Linux NICE DCV 서버에서 지원하는 기능을 비교합니다.

기능	Windows NICE DCV 서버	Linux NICE DCV 서버
콘솔 세션	✓	✓
가상 세션	✗	✓
QUIC(UDP) 전송 프로토콜	✓	✓
구성 가능한 TCP/UDP 포트 및 주소	✓	✓
사용자 지정 TLS 인증서	✓	✓
유휴 클라이언트 연결 해제	✓	✓
GPU 공유	✗	✓
USB 원격화	✓	✓
스마트 카드 지원	✓	✓

기능	Windows NICE DCV 서버	Linux NICE DCV 서버
웹캠 지원	✓ (Windows 10 및 Server 2016 이상)	✗
세션 저장 및 파일 전송	✓	✓
복사 및 붙여넣기	✓	✓
사용자 지정 HTTP 헤더	✓	✓
세션에서 인쇄	✓	✓
스테레오 2.0 오디오 재생	✓	✓
서라운드 사운드 오디오 재생	✓ (최대 7.1)	✓ (최대 5.1)
스테레오 2.0 오디오 레코딩	✓	✓
터치스크린 지원	✓ (Windows 10 및 Server 2012 이상)	✓
스타일러스 지원	✓ (Windows 10 및 Server 2019)	✓
게임패드 지원	✓ (Windows 10 및 Server 2016 이상)	✗
전체 화면 선택 모니터	✓	✗
시간대 리디렉션	✓	✓
WebAuthn 리디렉션	✓	✗

NICE DCV 클라이언트 기능에 대한 자세한 내용은 NICE DCV 사용 설명서의 [NICE DCV 클라이언트 기능](#)을 참조하세요.

NICE DCV 서버 설정

NICE DCV를 사용하려면 NICE DCV 세션을 호스팅하려는 서버에 NICE DCV 서버 소프트웨어를 설치합니다. 적절한 소프트웨어 라이선스가 부여되어 있어야 합니다.

다음 주제에서는 NICE DCV 서버를 설치하고 라이선스를 부여하는 방법에 대해 설명합니다. [라이선스](#) 주제는 온프레미스 및 기타 클라우드 기반 서버에 설치하는 경우에만 적용됩니다. Amazon EC2 인스턴스에서 NICE DCV 서버를 사용하는 데는 라이선스가 필요하지 않기 때문입니다.

주제

- [NICE DCV 서버 설치](#)
- [NICE DCV 서버 라이선스](#)
- [NICE DCV 서버 업그레이드](#)
- [NICE DCV 서버 제거](#)
- [이미징 NICE DCV 서버](#)

NICE DCV 서버 설치

다음 주제에서는 Windows와 Linux에 NICE DCV 서버의 최신 버전을 설치하는 방법을 설명합니다. Amazon EC2 인스턴스나 기타 온프레미스 또는 클라우드 기반 서버에 NICE DCV를 설치하는 경우 이러한 단계를 따릅니다.

Note

NICE DCV 서버의 이전 버전에서 최신 버전으로 업그레이드하려는 경우 [NICE DCV 서버 업그레이드](#) 섹션을 참조하세요.

주제

- [Windows에 NICE DCV 서버 설치](#)
- [Linux에서 NICE DCV 서버 설치](#)

Windows에 NICE DCV 서버 설치

이 섹션에서는 Windows에 NICE DCV 서버를 설치하는 방법을 설명합니다.

주제

- [Amazon EC2 인스턴스의 Windows NICE DCV 서버에 대한 사전 조건](#)
- [Windows에 NICE DCV 서버 설치](#)

Amazon EC2 인스턴스의 Windows NICE DCV 서버에 대한 사전 조건

이 주제에서는 NICE DCV 서버를 설치하기 전에 Windows Amazon EC2 인스턴스를 구성하는 방법을 설명합니다. Amazon EC2 Windows 인스턴스에 NICE DCV 서버를 설치하지 않는 경우 이 사전 조건을 건너뛰세요.

주제

- [가속 컴퓨팅 인스턴스에 대한 사전 조건](#)
- [기타 인스턴스 패밀리에 대한 사전 조건](#)

가속 컴퓨팅 인스턴스에 대한 사전 조건

GPU 그래픽 인스턴스에 대한 사전 조건

GPU 그래픽 인스턴스(예: G2, G3, G4dn, G4ad 또는 G5)를 사용하는 경우 적절한 NVIDIA 또는 AMD GPU 드라이버를 설치하고 구성하는 것이 좋습니다. GPU 드라이버는 다음을 허용합니다.

- 응용 프로그램을 위한 DirectX 및 OpenGL 하드웨어 가속
- H.264 비디오 스트리밍 인코딩을 위한 하드웨어 가속
- 사용자 지정 가능한 서버 모니터 해상도
- 서버 모니터의 최대 해상도 향상(최대 4096x2160)
- 서버 모니터 수 증가

GPU 그래픽 인스턴스에 NVIDIA GPU 드라이버를 설치하는 방법에 대한 안내는 Amazon EC2 사용 설명서의 다음 주제를 참조하세요.

- NVIDIA GPU를 사용하는 인스턴스(예: G2, G3, G4dn 또는 G5 인스턴스)의 경우 [Windows에 NVIDIA 드라이버 설치](#)를 참조하세요.
- AMD GPU를 사용하는 인스턴스(예: G4ad 인스턴스)의 경우 [Windows 인스턴스에 AMD 드라이버 설치](#)를 참조하세요.

Amazon EC2 G4ad 인스턴스에 대한 자세한 내용은 [새 Amazon EC2 G4ad 인스턴스에 대한 심층 분석 블로그 게시물](#)을 참조하세요.

기타 가속 컴퓨팅 인스턴스에 대한 사전 조건

GPU 그래픽 인스턴스가 아닌 가속 컴퓨팅 인스턴스를 사용하는 경우(예: P2, P3 또는 P3dn 인스턴스), 적절한 NVIDIA GPU 드라이버를 설치하고 구성하는 것이 좋습니다. NVIDIA GPU 드라이버는 H.264 비디오 스트리밍 인코딩을 위한 하드웨어 가속을 활성화합니다.

가속 컴퓨팅 인스턴스에 NVIDIA GPU 드라이버를 설치하는 방법에 대한 안내는 Windows 인스턴스용 Amazon EC2 사용 설명서의 [퍼블릭 NVIDIA 드라이버](#)를 참조하세요.

가속 컴퓨팅 인스턴스에 NVIDIA GPU 드라이버를 설치해도 서버 모니터 제한이나 해상도가 향상되지 않습니다. 서버 모니터 해상도 지원을 추가하려면 NVIDIA GRID 드라이버를 설치할 수 있습니다. 자세한 내용은 NVIDIA 웹사이트의 [NVIDIA vGPU 소프트웨어](#)를 참조하세요.

기타 인스턴스 패밀리에 대한 사전 조건

가속 컴퓨팅 인스턴스 이외의 인스턴스에 대해서는, Windows 2016을 사용 중이거나 2023.1 이전 버전의 NICE DCV 서버를 실행 중인 경우 NICE DCV 가상 디스플레이 드라이버를 설치하는 것이 좋습니다. 여기에는 범용, 컴퓨팅 최적화, 메모리 최적화 및 스토리지 최적화 인스턴스 패밀리의 인스턴스가 포함됩니다.

NICE DCV 가상 디스플레이 드라이버를 설치하면 다음 사항이 지원됩니다.

- 최대 4개 모니터 지원
- 사용자 지정 해상도 지원
- 4K UHD 해상도 지원

NICE DCV 서버에서 연결된 서버 모니터는 Windows 제어판을 사용하여 관리할 수 없습니다.

Note

NICE DCV 가상 디스플레이 드라이버는 Windows Server 2016 이상에서 지원됩니다. DCV 서버 2023.1 이상 버전이 설치된 Windows Server 2019 이상 버전을 사용하는 경우 이 드라이버가 필요하지 않습니다.

⚠ Important

NICE DCV 가상 디스플레이 드라이버를 NVIDIA GPU 드라이버와 같은 다른 GPU 드라이버와 함께 설치하면 충돌이 발생할 수 있습니다. 충돌을 피하려면 NICE DCV 가상 디스플레이 드라이버를 다른 GPU 드라이버와 함께 설치하지 않는 것이 좋습니다.

인스턴스에 NICE DCV 가상 디스플레이 드라이버를 설치하려면 다음과 같이 하세요.

1. [NICE DCV 웹사이트](#)에서 NICE DCV 가상 디스플레이 드라이버 설치 프로그램을 다운로드합니다.
2. 마법사를 실행하여 드라이버를 설치하려면 설치 파일을 열거나 두 번 클릭합니다. 또는 다음 명령을 사용하여 무인 설치를 실행합니다.

```
C:\> nice-dcv-virtual-display-x64-Release-78.msi /quiet /norestart
```

3. 인스턴스를 재부팅한 다음 인스턴스에 다시 연결합니다.

Windows에 NICE DCV 서버 설치

설치 마법사를 사용하여 Windows 호스트 서버에 NICE DCV 서버를 설치할 수 있습니다. 마법사는 NICE DCV 서버 설치를 사용자 지정하는 방법을 보여주는 일련의 단계를 안내합니다. 아니면 명령줄을 사용하여 무인 설치를 수행할 수도 있습니다. 이렇게 하면 기본 설정을 사용하여 설치 절차를 자동화할 수 있습니다.

목차

- [마법사 사용](#)
- [무인 설치](#)

마법사 사용

안내 설치를 원할 경우 NICE DCV 서버 설치 마법사를 사용합니다.

마법사를 사용하여 Windows에 NICE DCV 서버를 설치하려면 다음과 같이 하세요.

1. NICE DCV 서버를 설치할 서버를 시작하고 연결합니다.
2. [NICE DCV](#) 웹사이트에서 NICE DCV 서버 설치 프로그램을 다운로드합니다.

Note

NICE DCV 서버는 64비트 버전에서만 사용할 수 있으며 64비트 Windows 운영 체제에서 지원됩니다.

Tip

다운로드 웹사이트의 [최신 패키지](#) 페이지에는 항상 사용 가능한 최신 버전을 가리키는 링크가 있습니다. 이러한 링크를 사용하여 최신 NICE DCV 패키지를 자동으로 찾을 수 있습니다.

3. nice-dcv-server-x64-Release-2023.1-*version_number*.msi를 실행합니다.
4. 시작 화면에서 다음을 선택합니다.
5. 최종 사용자 라이선스 계약 화면에서 라이선스 계약을 읽습니다. 약관에 동의하는 경우 라이선스 계약의 약관에 동의합니다 확인란을 선택하고 다음을 선택합니다.
6. (선택 사항) 구성 요소 선택 화면에서 원하는 항목을 선택하여 설치할 구성 요소를 구성합니다. 설치할 구성 요소를 표시하려면 해당 항목을 선택하고 로컬 하드 드라이브에 설치를 선택합니다. 설치에서 구성 요소를 생략하려면 생략할 항목을 선택하고 전체 기능을 사용할 수 없음을 선택합니다.
7. DCV 서비스 구성 화면에서:
 - a. (선택 사항) 필수 포트를 통한 통신을 허용하도록 서버의 방화벽을 수동으로 구성하려면 No, I will manually configure my firewall later(아니요, 나중에 방화벽을 수동으로 구성하겠습니다)를 선택합니다.
 - b. (선택 사항) 설치 후 NICE DCV 서버를 수동으로 시작하려면 아니요, DCV 서비스를 수동으로 시작하겠습니다를 선택합니다. 이 옵션을 선택하면 설치가 완료된 후 콘솔 세션을 자동으로 시작할 수 없습니다. 이 옵션을 선택하는 경우 9단계를 건너뛴니다.
8. 다음을 선택합니다.
9. DCV 세션 관리 구성 화면에서 자동 콘솔 세션의 소유자를 지정합니다. 또는 설치가 완료된 후 자동 콘솔 세션이 시작되지 않도록 하려면 No, I will create the session manually(아니요, 세션을 수동으로 생성하겠습니다)를 선택합니다.

Note

서버가 자동으로 시작되도록 이전에 선택한 경우에만 이 단계를 수행합니다.

10. 설치를 선택합니다.

무인 설치

무인 설치는 기본적으로 다음을 수행합니다.

- 포트 8443을 통한 통신을 허용하도록 방화벽 규칙을 추가합니다.
- NICE DCV 서버 자동 시작을 활성화합니다.
- 자동 콘솔 세션을 생성합니다.
- 콘솔 세션 소유자를 설치를 수행하는 사용자로 설정합니다.

설치 명령에 다음 옵션을 추가하여 기본 작업을 재정의할 수 있습니다.

- `DISABLE_FIREWALL=1` - 설치 관리자가 방화벽 규칙을 추가하지 못하게 합니다.
- `DISABLE_SERVER_AUTOSTART=1` - 설치 후 NICE DCV 서버가 자동으로 시작되지 못하게 합니다.
- `DISABLE_AUTOMATIC_SESSION_CREATION=1` - 설치 관리자가 자동 콘솔 세션을 시작하지 못하게 합니다.
- `AUTOMATIC_SESSION_OWNER=owner_name` - 자동 콘솔 세션에 다른 소유자를 지정합니다.
- `ADDLOCAL=component_list` - 설치할 요소 세트에 요소를 추가합니다.
- `REMOVE=component_list` - 설치할 요소 세트에서 요소를 제거합니다.

Note

REMOVE 옵션은 ADDLOCAL 옵션 다음에 평가됩니다. 두 목록에 모두 있는 요소는 설치되지 않습니다.

`component_list`는 쉼표로 구분된 목록이며 다음 값을 포함할 수 있습니다.

- `audioMicDriver`: 마이크 드라이버
- `audioSpkDriver`: 스피커 드라이버

- printerDriver: 프린터 드라이버
- usbDriver: USB 디바이스 원격화 드라이버(기본적으로 비활성화됨)
- webcamDriver: 웹캠 드라이버
- gamepadDriver: 게임패드 드라이버
- webClient: 웹 클라이언트
- webauthn: WebAuthn 리디렉션
- ALL: 모든 구성 요소

무인 설치를 사용하여 Windows에 NICE DCV 서버를 설치하려면 다음과 같이 하세요.

1. NICE DCV 서버를 설치할 서버를 시작하고 연결합니다.
2. [NICE](#) 웹사이트에서 NICE DCV 서버 설치 프로그램을 다운로드합니다.

Note

NICE DCV 서버는 64비트 버전에서만 사용할 수 있으며 64비트 Windows 운영 체제에서 지원됩니다.

3. 명령 프롬프트 창을 열고 설치 관리자를 다운로드한 폴더로 이동합니다.
4. 다음 예제 중 하나에 나와 있는 것처럼 자동 설치 프로그램을 실행합니다.

- 기본 구성 요소 설치:

```
C:\> msixexec.exe /i nice-dcv-server-x64-Release-2023.1-version_number.msi
/quiet /norestart /l*v dcv_install_msi.log
```

- 모든 구성 요소 설치:

```
C:\> msixexec.exe /i nice-dcv-server-x64-Release-2023.1-version_number.msi
ADDLOCAL=ALL /quiet /norestart /l*v dcv_install_msi.log
```

- 구성 요소 하위 세트 설치:

```
C:\> msixexec.exe /i nice-dcv-server-x64-Release-2023.1-version_number.msi
ADDLOCAL=audioMicDriver,audioSpkDriver,printerDriver,webcamDriver /quiet /
norestart /l*v dcv_install_msi.log
```

Linux에서 NICE DCV 서버 설치

이 섹션에서는 Linux에서 NICE DCV 서버를 설치하는 방법을 설명합니다.

주제

- [Linux NICE DCV 서버에 대한 사전 조건](#)
- [Linux에 NICE DCV 서버 설치](#)
- [설치 후 점검](#)

Linux NICE DCV 서버에 대한 사전 조건

NICE DCV를 사용하면 클라이언트가 Linux 서버의 원격 그래픽 X 세션에 액세스할 수 있게 됩니다. 이렇게 하면 해당 Linux 데스크톱에 액세스할 수 있습니다. NICE DCV는 콘솔 세션과 가상 세션이라는 두 가지 유형의 Linux 데스크톱 스트리밍을 지원합니다. 콘솔 및 가상 세션에 대한 자세한 내용은 [NICE DCV 세션 관리](#) 단원을 참조하십시오.

이 주제에서는 Linux 서버에서 NICE DCV를 사용하는 데 필요한 사전 조건을 설치하는 방법을 살펴봅니다.

내용

- [데스크톱 환경 및 데스크톱 관리자 설치](#)
- [Wayland 프로토콜 비활성화\(GDM3만 해당\)](#)
- [X 서버 구성](#)
- [glxinfo 유틸리티 설치](#)
- [OpenGL 소프트웨어 렌더링 확인](#)
- [그래픽 인스턴스용 GPU 드라이버 설치](#)
- [비GPU 인스턴스용 XDummy 드라이버 설치](#)

데스크톱 환경 및 데스크톱 관리자 설치

Linux 서버에서 NICE DCV 사용 환경을 개선할 수 있도록 데스크톱 환경 및 데스크톱 관리자를 설치할 수 있습니다.

데스크톱 환경은 Linux 운영 체제와의 상호 작용에 도움이 되는 그래픽 사용자 인터페이스(GUI)입니다. 데스크톱 환경으로는 몇 가지가 있으며 NICE DCV는 이들 중 다수와 연동합니다. 데스크톱 관리자

는 사용자 로그인 화면을 관리하는 프로그램이며 데스크톱 환경 세션과 X 서버를 시작하고 중지합니다.

다음 탭 내용은 지원되는 운영 체제에 기본 데스크톱 환경과 데스크톱 관리자를 설치하는 단계를 보여주며, 지원되는 운영 체제에서 X 서버를 구성하고 시작하는 방법을 안내합니다.

RHEL, CentOS, and Rocky Linux

RHEL, CentOS, Rocky Linux의 기본 데스크톱 환경은 Gnome30이고 기본 데스크톱 관리자는 GDM입니다.

RHEL, CentOS, Rocky Linux에서 데스크톱 환경과 데스크톱 관리자를 설치하고 구성하려면 다음과 같이 하세요.

1. 데스크톱 환경 및 데스크톱 관리자 패키지를 설치합니다.

- RHEL과 록키 리눅스

```
$ sudo yum groupinstall 'Server with GUI'
```

- CentOS

```
$ sudo yum groupinstall "GNOME Desktop"
```

2. 소프트웨어 패키지를 업데이트하여 Linux 서버를 최신 상태로 유지합니다.

```
$ sudo yum upgrade
```

3. Linux 서버를 재부팅합니다.

```
$ sudo reboot
```

Amazon Linux 2

Note

현재 NICE DCV는 아마존 리눅스 2023과 호환되지 않습니다. AL2023에는 NICE DCV를 실행하는 데 필요한 그래픽 데스크톱 환경이 포함되어 있지 않습니다.

Amazon Linux 2의 기본 데스크톱 환경은 Gnome3이고 기본 데스크톱 관리자는 GDM입니다.

Amazon Linux 2에서 데스크톱 환경과 데스크톱 관리자를 설치하고 구성하려면

1. 데스크톱 환경 및 데스크톱 관리자 패키지를 설치합니다.

```
$ sudo yum install gdm gnome-session gnome-classic-session gnome-session-xsession
```

```
$ sudo yum install xorg-x11-server-Xorg xorg-x11-fonts-Type1 xorg-x11-drivers
```

```
$ sudo yum install gnome-terminal gnu-free-fonts-common gnu-free-mono-fonts gnu-free-sans-fonts gnu-free-serif-fonts
```

2. 소프트웨어 패키지를 업데이트하여 Linux 서버를 최신 상태로 유지합니다.

```
$ sudo yum upgrade
```

3. Linux 서버를 재부팅합니다.

```
$ sudo reboot
```

Ubuntu 20.x and 22.x

Ubuntu 20.x/22.x의 기본 데스크톱 환경은 Gnome3이고 기본 데스크톱 관리자는 GDM3입니다.

Ubuntu 20.x부터 LightDM은 NICE DCV에서 더 이상 지원되지 않습니다.

Ubuntu 20.x/22.x에서 데스크톱 환경과 데스크톱 관리자를 설치하고 구성하려면 다음과 같이 하세요.

1. 데스크톱 환경 및 데스크톱 관리자 패키지를 설치합니다.

```
$ sudo apt update
```

```
$ sudo apt install ubuntu-desktop
```

GDM3 설치


```
$ sudo apt install gdm3
```

2. GDM3이 기본 데스크톱 관리자로 설정되어 있는지 확인합니다.

```
$ cat /etc/X11/default-display-manager
```

출력값은 다음과 같습니다.

```
/usr/sbin/gdm3
```

GDM3이 기본 데스크톱 관리자로 설정되어 있지 않은 경우, 다음 명령을 사용하여 기본값으로 설정하세요.

```
$ sudo dpkg-reconfigure gdm3
```

3. 소프트웨어 패키지를 업데이트하여 Linux 서버를 최신 상태로 유지합니다.

```
$ sudo apt upgrade
```

4. Linux 서버를 재부팅합니다.

```
$ sudo reboot
```

Note

가상 세션과 함께 NICE DCV의 2022.2 이전 버전을 사용하는 경우 [알려진 GDM 문제가 발생](#)할 수 있습니다. 가상 세션이 제대로 작동하도록 하려면 다음 솔루션 중 하나를 채택할 수 있습니다.

- GPU가 없는 서버에서는 가상 세션을 실행할 필요가 없으므로 데스크톱 관리자를 사용하지 않도록 설정할 수 있습니다. 가상 세션을 생성하기 전에 다음 명령을 실행하여 다중 사용자 모드에서 실행되도록 시스템을 구성하세요.

```
sudo systemctl isolate multi-user.target
```

- GPU가 있는 서버에서는 데스크톱 관리자를 사용하지 않도록 설정하는 것 외에도 가상 세션을 생성하기 전에 시스템에서 X 서버를 시작해야 합니다. 이를 위해 다음 명령을 실행합니다.

```
sudo systemctl isolate multi-user.target
```

```
sudo dcvstartx &
```

NICE DCV 2022.2 이상은 이 문제의 영향을 받지 않습니다.

SUSE Linux Enterprise 12.x

SUSE Linux Enterprise 12.x의 기본 데스크톱 환경은 SLE Classic이고 기본 데스크톱 관리자는 GDM입니다.

SUSE Linux Enterprise 12.x에서 데스크톱 환경과 데스크톱 관리자를 설치하고 구성하려면

1. 데스크톱 환경 및 데스크톱 관리자 패키지를 설치합니다.

```
$ sudo zypper install -t pattern gnome-basic
```

2. GDM이 기본 데스크톱 관리자로 설정되어 있는지 확인합니다.

```
$ sudo update-alternatives --set default-displaymanager /usr/lib/X11/  
displaymanagers/gdm
```

```
$ sudo sed -i "s/DEFAULT_WM=\"\"/DEFAULT_WM=\"gnome\"/" /etc/sysconfig/  
windowmanager
```

3. 소프트웨어 패키지를 업데이트하여 Linux 서버를 최신 상태로 유지합니다.

```
$ sudo zypper update
```

4. Linux 서버를 재부팅합니다.

```
$ sudo reboot
```

SUSE Linux Enterprise 15.x

SUSE Linux Enterprise 15.x의 기본 데스크톱 환경은 SLE Classic이고 기본 데스크톱 관리자는 GDM3입니다.

SUSE Linux Enterprise 15.x에서 데스크톱 환경과 데스크톱 관리자를 설치하고 구성하려면 다음과 같이 하세요.

1. 데스크톱 환경 및 데스크톱 관리자 패키지를 설치합니다.

```
$ sudo zypper install -t pattern gnome_basic
```

2. GDM이 기본 데스크톱 관리자로 설정되어 있는지 확인합니다.

```
$ sudo update-alternatives --set default-displaymanager /usr/lib/X11/displaymanagers/gdm
```

```
$ sudo sed -i "s/DEFAULT_WM=\"\"/DEFAULT_WM=\"gnome\"/" /etc/sysconfig/windowmanager
```

3. 소프트웨어 패키지를 업데이트하여 Linux 서버를 최신 상태로 유지합니다.

```
$ sudo zypper update
```

4. Linux 서버를 재부팅합니다.

```
$ sudo reboot
```

Note

가상 세션과 함께 NICE DCV의 2022.2 이전 버전을 사용하는 경우 [알려진 GDM 문제](#)가 발생할 수 있습니다. 가상 세션이 제대로 작동하도록 하려면 다음 솔루션 중 하나를 채택할 수 있습니다.

- GPU가 없는 서버에서는 가상 세션을 실행할 필요가 없으므로 데스크톱 관리자를 사용하지 않도록 설정할 수 있습니다. 가상 세션을 생성하기 전에 다음 명령을 실행하여 다중 사용자 모드에서 실행되도록 시스템을 구성하세요.

```
sudo systemctl isolate multi-user.target
```

- GPU가 있는 서버에서는 데스크톱 관리자를 사용하지 않도록 설정하는 것 외에도 가상 세션을 생성하기 전에 시스템에서 X 서버를 시작해야 합니다. 이를 위해 다음 명령을 실행합니다.

```
sudo systemctl isolate multi-user.target
```

```
sudo dcvstartx &
```

NICE DCV 2022.2 이상은 이 문제의 영향을 받지 않습니다.

Wayland 프로토콜 비활성화(GDM3만 해당)

NICE DCV는 Wayland 프로토콜을 지원하지 않습니다. GDM3 데스크톱 관리자를 사용하는 경우 Wayland 프로토콜을 비활성화해야 합니다. GDM3를 사용하지 않는 경우 이 단계를 건너뜁니다.

Wayland 프로토콜을 비활성화하려면 다음과 같이 하세요.

1. 원하는 텍스트 편집기를 사용하여 파일을 엽니다.

- RHEL, CentOS, SUSE Linux Enterprise 15.x

```
/etc/gdm/custom.conf
```

- Ubuntu 18.x/20.x/22.x

```
/etc/gdm3/custom.conf
```

2. [daemon] 섹션에서 WaylandEnable을 false로 설정합니다.

```
[daemon]
WaylandEnable=false
```

3. GDM 서비스를 다시 시작합니다.

- RHEL, CentOS

```
$ sudo systemctl restart gdm
```

- Ubuntu 18.x/20.x/22.x

```
$ sudo systemctl restart gdm3
```

- SUSE Linux Enterprise 15.x

```
$ sudo systemctl restart xdm
```

X 서버 구성

콘솔 세션이나 GPU 공유를 사용하려는 경우 Linux 서버에 제대로 구성되고 실행 중인 X 서버가 있는지 확인해야 합니다.

Note

GPU 공유 없이 가상 세션을 사용하려는 경우 X 서버가 필요하지 않습니다.

X 서버 패키지는 일반적으로 데스크톱 환경 및 데스크톱 관리자의 종속 항목으로 설치됩니다. Linux 서버를 부팅할 때 X 서버가 자동으로 시작되도록 구성하는 것이 좋습니다.

RHEL, CentOS, Rocky Linux, Amazon Linux 2, Ubuntu 18.x, 20.x, 22.x, SUSE Linux Enterprise 12.x, 15.x에서 X 서버를 구성하고 시작하려면 다음과 같이 하세요.

1. Linux 서버를 부팅할 때 X 서버가 자동으로 시작되도록 구성합니다.

```
$ sudo systemctl get-default
```

명령이 `graphical.target`를 반환한다면, X 서버가 자동으로 시작되도록 이미 구성된 것입니다. 다음 단계로 이동합니다.

명령이 `multi-user.target`을 반환한다면 X 서버가 자동으로 시작되도록 구성되지 않은 것입니다. 다음 명령을 실행합니다.

```
$ sudo systemctl set-default graphical.target
```

2. X 서버를 시작합니다.

```
$ sudo systemctl isolate graphical.target
```

3. X 서버가 실행되는지 확인합니다.

```
$ ps aux | grep X | grep -v grep
```

다음은 X 서버가 실행되는 경우의 출력 예를 보여줍니다.

```
root 1891 0.0 0.7 277528 30448 tty7 Ssl+ 10:59 0:00 /usr/bin/Xorg :0 -
background none -verbose -auth /run/gdm/auth-for-gdm-wltseN/database -
seat seat0 vt7
```

glxinfo 유틸리티 설치

glxinfo 유틸리티는 Linux 서버의 OpenGL 구성에 대한 정보를 제공합니다. 이 유틸리티를 사용하여 Linux 서버가 OpenGL 하드웨어 또는 소프트웨어 렌더링을 지원하도록 구성되어 있는지 확인할 수 있습니다. 이 유틸리티는 드라이버 및 지원되는 확장 프로그램에 대한 정보를 제공합니다.

glxinfo 유틸리티가 DCV GL의 패키지 종속 항목으로 설치됩니다. 따라서 DCV GL을 설치한 경우 glxinfo 유틸리티가 Linux 서버에 이미 설치되어 있습니다.

RHEL, CentOS, Rocky Linux, and Amazon Linux 2

glxinfo 유틸리티를 설치하려면

다음 명령을 실행합니다:

```
$ sudo yum install glx-utils
```

Ubuntu

glxinfo 유틸리티를 설치하려면

다음 명령을 실행합니다:

```
$ sudo apt install mesa-utils
```

SUSE Linux Enterprise

glxinfo 유틸리티를 설치하려면

다음 명령을 실행합니다:

```
$ sudo zypper in Mesa-demo-x
```

OpenGL 소프트웨어 렌더링 확인

비 GPU Linux 서버에서 OpenGL은 Mesa 드라이버를 사용하는 소프트웨어 렌더링 모드에서만 지원됩니다. 비GPU Linux 서버를 사용 중이며 OpenGL을 사용할 계획이라면, Mesa 드라이버가 Linux 서버에 설치되어 올바르게 구성되어 있어야 합니다.

Note

이는 비 GPU Linux 서버에만 적용됩니다.

OpenGL 소프트웨어 렌더링이 사용 가능한지 확인하려면

X 서버가 실행 중인지 확인하고 다음 명령을 사용합니다.

```
$ sudo DISPLAY=:0 XAUTHORITY=$(ps aux | grep "X.*\-auth" | grep -v grep | sed -n 's/.*-auth \([^ ]*\+\.*/\1/p') glxinfo | grep -i "opengl.*version"
```

다음에서는 OpenGL 소프트웨어 렌더링을 사용할 수 있는 경우의 출력 예를 보여줍니다.

```
OpenGL core profile version string: 3.3 (Core Profile) Mesa 17.0.5
OpenGL core profile shading language version string: 3.30
OpenGL version string: 3.0 Mesa 17.0.5
OpenGL shading language version string: 1.30
OpenGL ES profile version string: OpenGL ES 3.0 Mesa 17.0.5
OpenGL ES profile shading language version string: OpenGL ES GLSL ES 3.00
```

그래픽 인스턴스용 GPU 드라이버 설치

주제

- [NVIDIA 드라이버 설치 및 구성](#)
- [AMD 드라이버 설치 및 구성](#)

NVIDIA 드라이버 설치 및 구성

전용 NVIDIA GPU를 탑재한 Linux 서버의 경우, 적절한 NVIDIA 드라이버가 Linux 서버에 설치되어 올바르게 구성되어 있는지 확인해야 합니다. Amazon EC2 Linux 인스턴스에 NVIDIA 드라이버를 설치하는 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Linux 서버에 NVIDIA 드라이버 설치](#)를 참조하세요.

Note

- 이는 NVIDIA GPU를 탑재한 Linux 서버에만 적용됩니다.
- GRID 드라이버는 설치된 각 GPU에 대해 최대 4개의 4K 디스플레이를 지원합니다. 게임 드라이버는 설치된 각 GPU에 대해 하나의 4K 디스플레이만 지원합니다.

NVIDIA 드라이버를 Linux 서버에 설치한 후에는 `xorg.conf`를 업데이트해야 합니다.

업데이트된 `xorg.conf` 파일을 생성하려면

1. 다음 명령을 실행합니다.

```
sudo nvidia-xconfig --preserve-busid --enable-all-gpus
```

G3, G4 또는 G5 Amazon EC2 인스턴스를 사용 중인 상황에서 다중 모니터 콘솔 세션을 사용하려는 경우 `--connected-monitor=DFP-0,DFP-1,DFP-2,DFP-3` 파라미터를 포함합니다. 방법은 다음과 같습니다.

```
sudo nvidia-xconfig --preserve-busid --enable-all-gpus --connected-monitor=DFP-0,DFP-1,DFP-2,DFP-3
```

Note

서버에 레거시 `/etc/X11/XF86Config` 파일이 없는지 확인하세요. 이 파일이 있으면 `nvidia-xconfig`는 필수 `/etc/X11/xorg.conf` 파일을 생성하는 대신 해당 구성 파일을 업데이트합니다. 다음 명령을 실행하여 레거시 `XF86Config` 파일을 제거하십시오.

```
sudo rm -rf /etc/X11/XF86Config*
```

2. X 서버를 다시 시작하여 변경 사항을 적용합니다.

- ```
$ sudo systemctl isolate multi-user.target
```

```
$ sudo systemctl isolate graphical.target
```



NVIDIA GPU가 하드웨어 기반 비디오 인코딩을 지원하는지 확인하려면 다음과 같이 하세요.

NVENC 인코딩을 지원하는지, 컴퓨팅 기능이 3.0 이상 또는 Ubuntu 20의 경우 3.5 이상인지 확인하세요.

NVENC 지원을 확인하려면 [NVIDIA 비디오 인코딩 및 디코딩 GPU 지원 매트릭스](#)를 참조하세요. 컴퓨팅 성능을 확인하려면 [NVIDIA 컴퓨팅 용량 표](#)를 참조하세요.

NVIDIA GPU가 NVENC 인코딩을 지원하지 않거나 필요한 컴퓨팅 기능이 없는 경우 소프트웨어 기반 비디오 인코딩이 사용됩니다.

OpenGL 하드웨어 렌더링이 사용 가능한지 확인하려면

다음 명령을 사용하여 X 서버가 실행 중인지 확인합니다.

```
$ sudo DISPLAY=:0 XAUTHORITY=$(ps aux | grep "X.*\-auth" | grep -v grep | sed -n 's/.*-auth \([^]+\).*\1/p') glxinfo | grep -i "opengl.*version"
```

다음에서는 OpenGL 하드웨어 렌더링을 사용할 수 있는 경우의 출력 예를 보여줍니다.

```
OpenGL core profile version string: 4.4.0 NVIDIA 390.75
OpenGL core profile shading language version string: 4.40 NVIDIA via Cg compiler
OpenGL version string: 4.6.0 NVIDIA 390.75
OpenGL shading language version string: 4.60 NVIDIA
OpenGL ES profile version string: OpenGL ES 3.2 NVIDIA 390.75
OpenGL ES profile shading language version string: OpenGL ES GLSL ES 3.20
```

## AMD 드라이버 설치 및 구성

G4ad 인스턴스와 같이 연결된 AMD GPU가 있는 인스턴스에는 적절한 AMD 드라이버가 설치되어 있어야 합니다. 호환되는 Amazon EC2 인스턴스에 AMD GPU 드라이버를 설치하는 방법에 대한 안내는 [Linux 인스턴스에 AMD 드라이버 설치](#)를 참조하세요.

Amazon EC2 G4ad 인스턴스에 대한 자세한 내용은 [새 Amazon EC2 G4ad 인스턴스에 대한 심층 분석](#) 블로그 게시물을 참조하세요.

## 비GPU 인스턴스용 XDummy 드라이버 설치

### 주제

- [XDummy 드라이버 설치 및 구성](#)

## XDummy 드라이버 설치 및 구성

전용 GPU가 없는 Linux 서버에서 콘솔 세션을 사용하려면 Xdummy 드라이버가 설치되고 제대로 구성 되어 있어야 합니다. XDummy 드라이버를 사용하면 실제 GPU가 없을 때 가상 프레임 버퍼로 X 서버를 실행할 수 있습니다.

### Note

- 가상 세션을 사용하려는 경우 필수 사항은 아닙니다.
- XDummy 드라이버는 해당 구성에 정의된 해상도만 지원할 수 있습니다.

## RHEL, CentOS, Rocky Linux, and Amazon Linux 2

XDummy 드라이버를 설치하려면 다음과 같이 하세요.

다음 명령을 실행합니다:

```
$ sudo yum install xorg-x11-drv-dummy
```

## Ubuntu

XDummy 드라이버를 설치하려면 다음과 같이 하세요.

다음 명령을 실행합니다:

```
$ sudo apt install xserver-xorg-video-dummy
```

## SUSE Linux Enterprise

XDummy 드라이버를 설치하려면 다음과 같이 하세요.

다음 명령을 실행합니다:

```
$ sudo zypper in xf86-video-dummy
```

XDummy 드라이버를 Linux 서버에 설치한 후에는 `xorg.conf`를 업데이트해야 합니다.

xorg.conf에서 XDummy를 구성하려면 다음과 같이 하세요.

1. 선호하는 텍스트 편집기로 /etc/X11/xorg.conf 파일을 엽니다.
2. 구성에 다음 섹션을 추가합니다.

```
Section "Device"
 Identifier "DummyDevice"
 Driver "dummy"
 Option "UseEDID" "false"
 VideoRam 512000
EndSection

Section "Monitor"
 Identifier "DummyMonitor"
 HorizSync 5.0 - 1000.0
 VertRefresh 5.0 - 200.0
 Option "ReducedBlanking"
EndSection

Section "Screen"
 Identifier "DummyScreen"
 Device "DummyDevice"
 Monitor "DummyMonitor"
 DefaultDepth 24
 SubSection "Display"
 Viewport 0 0
 Depth 24
 Virtual 4096 2160
 EndSubSection
EndSection
```

#### Note

제공된 구성은 예제입니다. 모드를 더 추가하고 다른 virtual 해상도를 설정할 수 있습니다. 더미 모니터를 두 개 이상 구성할 수도 있습니다.

3. X 서버를 다시 시작하여 변경 사항을 적용합니다.

- ```
$ sudo systemctl isolate multi-user.target
```

```
$ sudo systemctl isolate graphical.target
```

Linux에 NICE DCV 서버 설치

NICE DCV 서버는 호스트 서버의 운영 체제에 따라 일련의 RPM 또는 .deb 패키지를 사용하여 설치됩니다. 이 패키지는 모든 필수 패키지와 종속 파일을 설치하고, 필요한 서버 구성을 수행합니다.

Note

NICE DCV 서버를 설치하려면 루트 사용자로 로그인해야 합니다.

NICE DCV 서버 설치

Amazon Linux 2 and RHEL/CentOS

NICE DCV 서버는 64비트 x86 및 64비트 ARM 아키텍처를 기반으로 하는 Amazon Linux 2, RHEL, CentOS 서버에서 사용할 수 있습니다.

Important

64비트 ARM 아키텍처 기반 서버에는 nice-dcv-gl 및 nice-dcv-gltest 패키지를 사용할 수 없습니다.

Amazon Linux 2, RHEL, CentOS에 NICE DCV 서버를 설치하려면 다음과 같이 하세요.

1. NICE DCV 서버를 설치할 서버를 시작하고 연결합니다.
2. NICE DCV 서버 패키지는 안전한 GPG 서명으로 디지털 서명됩니다. 패키지 관리자가 패키지 서명을 확인할 수 있도록 하려면 NICE GPG 키를 가져와야 합니다. 이렇게 하려면 터미널 창을 열고 NICE GPG 키를 가져옵니다.

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

3. [NICE DCV 다운로드 웹 사이트](#)에서 패키지를 다운로드합니다. RPM 및 deb 패키지는 .tgz 아카이브로 패키징되어 있습니다. 사용 중인 운영 체제에 알맞은 아카이브를 다운로드해야 합니다.

- 64비트 x86

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/Servers/nice-dcv-2023.1-16388-e17-x86_64.tgz
```

- 64비트 ARM

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/Servers/nice-dcv-2023.1-16388-e17-aarch64.tgz
```

Tip

다운로드 웹사이트의 [최신 패키지](#) 페이지에는 항상 사용 가능한 최신 버전을 가리키는 링크가 있습니다. 이러한 링크를 사용하여 최신 NICE DCV 패키지를 자동으로 찾을 수 있습니다.

- 64비트 x86

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-e17-x86_64.tgz
```

- 64비트 ARM

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-e17-aarch64.tgz
```

4. .tgz 아카이브의 콘텐츠를 추출하고 추출된 디렉터리로 이동합니다.

- 64비트 x86

```
$ tar -xvzf nice-dcv-2023.1-16388-e17-x86_64.tgz && cd nice-dcv-2023.1-16388-e17-x86_64
```

- 64비트 ARM

```
$ tar -xvzf nice-dcv-2023.1-16388-e17-aarch64.tgz && cd nice-dcv-2023.1-16388-e17-aarch64
```

5. NICE DCV 서버를 설치합니다.

- 64비트 x86

```
$ sudo yum install nice-dcv-server-2023.1.16388-1.el7.x86_64.rpm
```

- 64비트 ARM

```
$ sudo yum install nice-dcv-server-2023.1.16388-1.el7.aarch64.rpm
```

6. (선택 사항) NICE DCV 버전 2021.2 이상에서 웹 클라이언트를 사용하려면 `nice-dcv-web-viewer` 패키지를 설치합니다.

- 64비트 x86

```
$ sudo yum install nice-dcv-web-viewer-2023.1.16388-1.el7.x86_64.rpm
```

- 64비트 ARM

```
$ sudo yum install nice-dcv-web-viewer-2023.1.16388-1.el7.aarch64.rpm
```

7. (선택 사항) 가상 세션을 사용하려면 `nice-xdcv` 패키지를 설치합니다.

- 64비트 x86

```
$ sudo yum install nice-xdcv-2023.1.565-1.el7.x86_64.rpm
```

- 64비트 ARM

```
$ sudo yum install nice-xdcv-2023.1.565-1.el7.aarch64.rpm
```

8. (선택 사항) GPU 공유를 사용할 계획인 경우 `nice-dcv-gl` 패키지를 설치합니다.

- 64비트 x86

```
$ sudo yum install nice-dcv-gl-2023.1.1047-1.el7.x86_64.rpm
```

Note

필요할 경우 `nice-dcv-gltest` 패키지를 설치할 수 있습니다. 이 패키지에는 가상 세션이 하드웨어 기반 OpenGL을 사용하도록 올바르게 구성되었는지 확인하는 데 사용할 수 있는 간단한 OpenGL 애플리케이션이 포함되어 있습니다.

9. (선택 사항) NICE와 함께 NICE DCV를 사용하려는 경우 패키지를 EnginFrame 설치하십시오.
nice-dcv-simple-external-authenticator

- 64비트 x86

```
$ sudo yum install nice-dcv-simple-external-  
authenticator-2023.1.228-1.el7.x86_64.rpm
```

- 64비트 ARM

```
$ sudo yum install nice-dcv-simple-external-  
authenticator-2023.1.228-1.el7.aarch64.rpm
```

10. (선택 사항) USB 원격화를 사용하여 특수화된 USB 디바이스를 지원하려면 DCV USB 드라이버를 설치합니다.

DCV USB 드라이버를 설치하려면 서버에 DKMS(Dynamic Kernel Module Support)가 설치되어 있어야 합니다. 다음 명령을 사용하여 DKMS를 설치합니다.

DKMS를 EPEL(Extra Packages for Enterprise Linux) 리포지토리에서 설치할 수 있습니다. 다음 명령을 사용하여 EPEL 리포지토리를 활성화합니다.

```
$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-  
latest-7.noarch.rpm
```

EPEL 리포지토리를 활성화한 후, 다음 명령을 실행하여 DKMS를 설치합니다.

```
$ sudo yum install dkms
```

DKMS를 설치한 후, 다음 명령을 실행하여 DCV USB 드라이버를 설치합니다.

```
$ sudo dcvusbdriverinstaller
```

11. (선택 사항) 마이크 리디렉션을 지원하려는 경우 pulseaudio-utils 패키지가 시스템에 설치되어 있는지 확인합니다. 다음 명령을 사용하여 설치합니다.

```
$ sudo yum install pulseaudio-utils
```

RHEL, CentOS, and Rocky Linux 8.5

NICE DCV 서버는 64비트 x86 및 64비트 ARM 아키텍처, Rocky Linux 8.5 이상을 기반으로 하는 RHEL, CentOS 서버에서 사용할 수 있습니다.

Important

64비트 ARM 아키텍처 기반 서버에는 nice-dcv-gl 및 nice-dcv-gltest 패키지를 사용할 수 없습니다.

RHEL, CentOS 및/또는 Rocky Linux 8.5에 NICE DCV 서버를 설치하려면 다음과 같이 하세요.

1. NICE DCV 서버를 설치할 서버를 시작하고 연결합니다.
2. NICE DCV 서버 패키지는 안전한 GPG 서명으로 디지털 서명됩니다. 패키지 관리자가 패키지 서명을 확인할 수 있도록 하려면 NICE GPG 키를 가져와야 합니다. 이렇게 하려면 터미널 창을 열고 NICE GPG 키를 가져옵니다.

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

3. [NICE DCV 다운로드 웹 사이트](#)에서 패키지를 다운로드합니다. RPM 및 deb 패키지는 .tgz 아카이브로 패키징되어 있습니다. 사용 중인 운영 체제에 알맞은 아카이브를 다운로드해야 합니다.

- 64비트 x86

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/Servers/nice-dcv-2023.1-16388-el8-x86_64.tgz
```

- 64비트 ARM

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/Servers/nice-dcv-2023.1-16388-el8-aarch64.tgz
```

Tip

다운로드 웹사이트의 [최신 패키지](#) 페이지에는 항상 사용 가능한 최신 버전을 가리키는 링크가 있습니다. 이러한 링크를 사용하여 최신 NICE DCV 패키지를 자동으로 찾을 수 있습니다.

- 64비트 x86

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-el8-x86_64.tgz
```

- 64비트 ARM

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-el8-aarch64.tgz
```

4. .tgz 아카이브의 콘텐츠를 추출하고 추출된 디렉터리로 이동합니다.

- 64비트 x86

```
$ tar -xvzf nice-dcv-2023.1-16388-el8-x86_64.tgz && cd nice-dcv-2023.1-16388-el8-x86_64
```

- 64비트 ARM

```
$ tar -xvzf nice-dcv-2023.1-16388-el8-aarch64.tgz && cd nice-dcv-2023.1-16388-el8-aarch64
```

5. NICE DCV 서버를 설치합니다.

- 64비트 x86

```
$ sudo yum install nice-dcv-server-2023.1.16388-1.el8.x86_64.rpm
```

- 64비트 ARM

```
$ sudo yum install nice-dcv-server-2023.1.16388-1.el8.aarch64.rpm
```

6. (선택 사항) NICE DCV 버전 2021.2 이상에서 웹 클라이언트를 사용하려면 nice-dcv-web-viewer 패키지를 설치합니다.

- 64비트 x86

```
$ sudo yum install nice-dcv-web-viewer-2023.1.16388-1.el8.x86_64.rpm
```

- 64비트 ARM

```
$ sudo yum install nice-dcv-web-server-2023.1.16388-1.el8.aarch64.rpm
```

7. (선택 사항) 가상 세션을 사용하려면 nice-xdcv 패키지를 설치합니다.

- 64비트 x86

```
$ sudo yum install nice-xdcv-2023.1.565-1.el8.x86_64.rpm
```


- 64비트 ARM

```
$ sudo yum install nice-xdcv-2023.1.565-1.el8.aarch64.rpm
```

8. (선택 사항) GPU 공유를 사용할 계획인 경우 nice-dcv-g1 패키지를 설치합니다.

- 64비트 x86

```
$ sudo yum install nice-dcv-g1-2023.1.1047-1.el8.x86_64.rpm
```

 Note

필요할 경우 nice-dcv-glttest 패키지를 설치할 수 있습니다. 이 패키지에는 가상 세션이 하드웨어 기반 OpenGL을 사용하도록 올바르게 구성되었는지 확인하는 데 사용할 수 있는 간단한 OpenGL 애플리케이션이 포함되어 있습니다.

9. (선택 사항) NICE와 함께 NICE DCV를 사용하려는 경우 패키지를 EnginFrame 설치하십시오. nice-dcv-simple-external-authenticator

- 64비트 x86

```
$ sudo yum install nice-dcv-simple-external-authenticator-2023.1.228-1.el8.x86_64.rpm
```

- 64비트 ARM

```
$ sudo yum install nice-dcv-simple-external-authenticator-2023.1.228-1.el8.aarch64.rpm
```

10. (옵션 사항) USB 원격화를 사용하여 특수화된 USB 기기를 지원할 계획인 경우, DCV USB 드라이버를 설치합니다.

DCV USB 드라이버를 설치하려면 서버에 DKMS(Dynamic Kernel Module Support)가 설치되어 있어야 합니다. 다음 명령을 사용하여 DKMS를 설치합니다.

DKMS를 EPEL(Extra Packages for Enterprise Linux) 리포지토리에서 설치할 수 있습니다. 다음 명령을 사용하여 EPEL 리포지토리를 활성화합니다.

```
$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

EPEL 리포지토리를 활성화한 후, 다음 명령을 실행하여 DKMS를 설치합니다.

```
$ sudo yum install dkms
```

DKMS를 설치한 후, 다음 명령을 실행하여 DCV USB 드라이버를 설치합니다.

```
$ sudo dcvusbdriverinstaller
```

11. (선택 사항) 마이크 리디렉션을 지원하려는 경우 pulseaudio-utils 패키지가 시스템에 설치되어 있는지 확인합니다. 다음 명령을 사용하여 설치합니다.

```
$ sudo yum install pulseaudio-utils
```

RHEL, CentOS, and Rocky Linux 9

NICE DCV 서버는 64비트 x86 및 64비트 ARM 아키텍처, Rocky Linux 9 이상을 기반으로 하는 RHEL, CentOS 서버에서 사용할 수 있습니다.

Important

64비트 ARM 아키텍처 기반 서버에는 nice-dcv-gl 및 nice-dcv-gltest 패키지를 사용할 수 없습니다.

RHEL, CentOS 및/또는 Rocky Linux 9에 NICE DCV 서버를 설치하려면 다음과 같이 하세요.

1. NICE DCV 서버를 설치할 서버를 시작하고 연결합니다.
2. NICE DCV 서버 패키지는 안전한 GPG 서명으로 디지털 서명됩니다. 패키지 관리자가 패키지 서명을 확인할 수 있도록 하려면 NICE GPG 키를 가져와야 합니다. 이렇게 하려면 터미널 창을 열고 NICE GPG 키를 가져옵니다.

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

3. [NICE DCV 다운로드 웹 사이트](#)에서 패키지를 다운로드합니다. RPM 및 deb 패키지는 .tgz 아카이브로 패키징되어 있습니다. 사용 중인 운영 체제에 알맞은 아카이브를 다운로드해야 합니다.

- 64비트 x86

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/Servers/nice-dcv-2023.1-16388-e19-x86_64.tgz
```

- 64비트 ARM

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/Servers/nice-dcv-2023.1-16388-e19-aarch64.tgz
```

Tip

다운로드 웹사이트의 [최신 패키지](#) 페이지에는 항상 사용 가능한 최신 버전을 가리키는 링크가 있습니다. 이러한 링크를 사용하여 최신 NICE DCV 패키지를 자동으로 찾을 수 있습니다.

- 64비트 x86

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-e19-x86_64.tgz
```

- 64비트 ARM

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-e19-aarch64.tgz
```

4. .tgz 아카이브의 콘텐츠를 추출하고 추출된 디렉터리로 이동합니다.

- 64비트 x86

```
$ tar -xvzf nice-dcv-2023.1-16388-e19-x86_64.tgz && cd nice-dcv-2023.1-16388-e19-x86_64
```

- 64비트 ARM

```
$ tar -xvzf nice-dcv-2023.1-16388-el9-aarch64.tgz && cd nice-dcv-2023.1-16388-el9-aarch64
```

5. NICE DCV 서버를 설치합니다.

- 64비트 x86

```
$ sudo yum install nice-dcv-server-2023.1.16388-1.el9.x86_64.rpm
```

- 64비트 ARM

```
$ sudo yum install nice-dcv-server-2023.1.16388-1.el9.aarch64.rpm
```

6. (선택 사항) NICE DCV 버전 2021.2 이상에서 웹 클라이언트를 사용하려면 nice-dcv-web-viewer 패키지를 설치합니다.

- 64비트 x86

```
$ sudo yum install nice-dcv-web-viewer-2023.1.16388-1.el9.x86_64.rpm
```

- 64비트 ARM

```
$ sudo yum install nice-dcv-web-server-2023.1.16388-1.el9.aarch64.rpm
```

7. (선택 사항) 가상 세션을 사용하려면 nice-xdcv 패키지를 설치합니다.

- 64비트 x86

```
$ sudo yum install nice-xdcv-2023.1.565-1.el9.x86_64.rpm
```

- 64비트 ARM

```
$ sudo yum install nice-xdcv-2023.1.565-1.el9.aarch64.rpm
```

8. (선택 사항) GPU 공유를 사용할 계획인 경우 nice-dcv-gl 패키지를 설치합니다.

- 64비트 x86

```
$ sudo yum install nice-dcv-gl-2023.1.1047-1.el9.x86_64.rpm
```

Note

필요할 경우 nice-dcv-glttest 패키지를 설치할 수 있습니다. 이 패키지에는 가상 세션이 하드웨어 기반 OpenGL을 사용하도록 올바르게 구성되었는지 확인하는 데 사용할 수 있는 간단한 OpenGL 애플리케이션이 포함되어 있습니다.

9. (선택 사항) NICE와 함께 NICE DCV를 사용하려는 경우 패키지를 EnginFrame 설치하십시오.
nice-dcv-simple-external-authenticator

- 64비트 x86

```
$ sudo yum install nice-dcv-simple-external-
authenticator-2023.1.228-1.el9.x86_64.rpm
```

- 64비트 ARM

```
$ sudo yum install nice-dcv-simple-external-
authenticator-2023.1.228-1.el9.aarch64.rpm
```

10. (옵션 사항) USB 원격화를 사용하여 특수화된 USB 기기를 지원할 계획인 경우, DCV USB 드라이버를 설치합니다.

DCV USB 드라이버를 설치하려면 서버에 DKMS(Dynamic Kernel Module Support)가 설치되어 있어야 합니다. 다음 명령을 사용하여 DKMS를 설치합니다.

DKMS를 EPEL(Extra Packages for Enterprise Linux) 리포지토리에서 설치할 수 있습니다. 다음 명령을 사용하여 EPEL 리포지토리를 활성화합니다.

```
$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-
latest-9.noarch.rpm
```

EPEL 리포지토리를 활성화한 후, 다음 명령을 실행하여 DKMS를 설치합니다.

```
$ sudo yum install dkms
```

DKMS를 설치한 후, 다음 명령을 실행하여 DCV USB 드라이버를 설치합니다.

```
$ sudo dcvusbdriverinstaller
```

11. (선택 사항) 마이크 리디렉션을 지원하려는 경우 pulseaudio-utils 패키지가 시스템에 설치되어 있는지 확인합니다. 다음 명령을 사용하여 설치합니다.

```
$ sudo yum install pulseaudio-utils
```

SLES 12.x/15.x

NICE DCV 서버는 64비트 x86 아키텍처를 기반으로 하는 SUSE Linux Enterprise Server(SLES) 12.x/15.x 서버에서만 사용할 수 있습니다.

SLES 12.x/15.x에 NICE DCV 서버를 설치하려면 다음과 같이 하세요.

1. NICE DCV 서버를 설치할 서버를 시작하고 연결합니다.
2. NICE DCV 서버 패키지는 안전한 GPG 서명으로 디지털 서명됩니다. 패키지 관리자가 패키지 서명을 확인할 수 있도록 하려면 NICE GPG 키를 가져와야 합니다. 이렇게 하려면 터미널 창을 열고 NICE GPG 키를 가져옵니다.

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

3. [NICE DCV 다운로드 웹 사이트](#)에서 패키지를 다운로드합니다. RPM 및 deb 패키지는 .tgz 아카이브로 패키징되어 있습니다. 사용 중인 운영 체제에 알맞은 아카이브를 다운로드해야 합니다.

- SLES 12.x

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/Servers/nice-dcv-2023.1-16388-sles12-x86_64.tgz
```

- SLES 15.x

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/Servers/nice-dcv-2023.1-16388-sles15-x86_64.tgz
```

i Tip

다운로드 웹사이트의 [최신 패키지](#) 페이지에는 항상 사용 가능한 최신 버전을 가리키는 링크가 있습니다. 이러한 링크를 사용하여 최신 NICE DCV 패키지를 자동으로 찾을 수 있습니다.

- SLES 12.x

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-sles12-x86_64.tgz
```

- SLES 15.x

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-sles15-x86_64.tgz
```

4. .tgz 아카이브의 콘텐츠를 추출하고 추출된 디렉터리로 이동합니다.

- SLES 12.x

```
$ tar -xvzf nice-dcv-2023.1-16388-sles12-x86_64.tgz && cd nice-dcv-2023.1-16388-sles12-x86_64
```

- SLES 15.x

```
$ tar -xvzf nice-dcv-2023.1-16388-sles15-x86_64.tgz && cd nice-dcv-2023.1-16388-sles15-x86_64
```

5. NICE DCV 서버를 설치합니다.

- SLES 12.x

```
$ sudo zypper install nice-dcv-server-2023.1.16388-1.sles12.x86_64.rpm
```

- SLES 15.x

```
$ sudo zypper install nice-dcv-server-2023.1.16388-1.sles15.x86_64.rpm
```


- (선택 사항) NICE DCV 버전 2021.2 이상에서 웹 클라이언트를 사용하려면 `nice-dcv-web-viewer` 패키지를 설치합니다.

- SLES 12.x

```
$ sudo zypper install nice-dcv-web-viewer-2023.1.16388-1.sles12.x86_64.rpm
```

- SLES 15.x

```
$ sudo zypper install nice-dcv-web-viewer-2023.1.16388-1.sles15.x86_64.rpm
```

- (선택 사항) 가상 세션을 사용하려면 `nice-xdcv` 패키지를 설치합니다.

- SLES 12.x

```
$ sudo zypper install nice-xdcv-2023.1.565-1.sles12.x86_64.rpm
```

- SLES 15.x

```
$ sudo zypper install nice-xdcv-2023.1.565-1.sles15.x86_64.rpm
```

- (선택 사항) GPU 공유를 사용할 계획인 경우 `nice-dcv-gl` 패키지를 설치합니다.

- SLES 12.x

```
$ sudo zypper install nice-dcv-gl-2023.1.1047-1.sles12.x86_64.rpm
```

- SLES 15.x

```
$ sudo zypper install nice-dcv-gl-2023.1.1047-1.sles15.x86_64.rpm
```

Note

필요할 경우 `nice-dcv-gltest` 패키지를 설치할 수 있습니다. 이 패키지에는 가상 세션이 하드웨어 기반 OpenGL을 사용하도록 올바르게 구성되었는지 확인하는 데 사용할 수 있는 간단한 OpenGL 애플리케이션이 포함되어 있습니다.

- (선택 사항) NICE와 함께 NICE DCV를 사용하려는 경우 패키지를 `EnginFrame` 설치하십시오. `nice-dcv-simple-external-authenticator`

- SLES 12.x

```
$ sudo zypper install nice-dcv-simple-external-
authenticator-2023.1.228-1.sles12.x86_64.rpm
```

- SLES 15.x

```
$ sudo zypper install nice-dcv-simple-external-
authenticator-2023.1.228-1.sles15.x86_64.rpm
```

10. (옵션 사항) USB 원격화를 사용하여 특수화된 USB 기기를 지원할 계획인 경우, DCV USB 드라이버를 설치합니다.

DCV USB 드라이버를 설치하려면 서버에 DKMS(Dynamic Kernel Module Support)가 설치되어 있어야 합니다. 다음 명령을 사용하여 DKMS를 설치합니다.

다음 명령을 실행하여 DKMS를 설치합니다.


- SLES 12.x

```
$ sudo zypper install http://download.opensuse.org/repositories/home:/
Ximi1970:/Dkms:/Staging/SLE_12_SP4/noarch/dkms-2.5-11.1.noarch.rpm
```

- SLES 15

리포지토리를 활성화합니다. PackageHub

```
$ sudo SUSEConnect -p PackageHub/15/x86_64
```

 Note

SLES 15 SP1 또는 SP2를 사용하는 경우 위 명령의 **15**를 15.1 또는 15.2로 바꿉니다.

DKMS를 설치합니다.

```
$ sudo zypper refresh
```

```
$ sudo zypper install dkms
```

커널 소스를 설치합니다.

```
$ sudo zypper install -y kernel-source
```

인스턴스를 재부팅합니다.

```
$ sudo reboot
```

DKMS를 설치한 후, 다음 명령을 실행하여 DCV USB 드라이버를 설치합니다.

```
$ sudo dcvusbdriverinstaller
```

11. (선택 사항) 마이크 리디렉션을 지원하려는 경우 pulseaudio-utils 패키지가 시스템에 설치되어 있는지 확인합니다. 다음 명령을 사용하여 설치합니다.

```
$ sudo zypper install pulseaudio-utils
```

Ubuntu 20.04/22.04

NICE DCV 서버는 64비트 x86 및 64비트 ARM 아키텍처를 기반으로 하는 Ubuntu 서버에서 사용할 수 있습니다.

Important

64비트 ARM 아키텍처 기반 서버에는 nice-dcv-gl 및 nice-dcv-gltest 패키지를 사용할 수 없습니다.

Ubuntu 20.04/22.04에 NICE DCV 서버를 설치하려면 다음과 같이 하세요.

1. NICE DCV 서버를 설치할 서버를 시작하고 연결합니다.
2. NICE DCV 서버 패키지는 안전한 GPG 서명으로 디지털 서명됩니다. 패키지 관리자가 패키지 서명을 확인할 수 있도록 하려면 NICE GPG 키를 가져와야 합니다. 이렇게 하려면 터미널 창을 열고 NICE GPG 키를 가져옵니다.

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

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

3. [NICE DCV 다운로드 웹 사이트](#)에서 패키지를 다운로드합니다. RPM 및 deb 패키지는 .tgz 아카이브로 패키징되어 있습니다. 사용 중인 운영 체제에 알맞은 아카이브를 다운로드해야 합니다.

- Ubuntu 20.04(64비트 x86)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/Servers/nice-dcv-2023.1-16388-ubuntu2004-x86_64.tgz
```

- Ubuntu 22.04(64비트 x86)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/Servers/nice-dcv-2023.1-16388-ubuntu2204-x86_64.tgz
```

- Ubuntu 22.04(64비트 ARM)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2023.1/Servers/nice-dcv-2023.1-16388-ubuntu2204-aarch64.tgz
```

Tip

다운로드 웹사이트의 [최신 패키지](#) 페이지에는 항상 사용 가능한 최신 버전을 가리키는 링크가 있습니다. 이러한 링크를 사용하여 최신 NICE DCV 패키지를 자동으로 찾을 수 있습니다.

- Ubuntu 20.04(64비트 x86)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-ubuntu2004-x86_64.tgz
```

- Ubuntu 22.04(64비트 x86)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-ubuntu2204-x86_64.tgz
```

- Ubuntu 22.04(64비트 ARM)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-ubuntu2204-aarch64.tgz
```

4. .tgz 아카이브의 콘텐츠를 추출하고 추출된 디렉터리로 이동합니다.

- Ubuntu 20.04(64비트 x86)

```
$ tar -xvzf nice-dcv-2023.1-16388-ubuntu2004-x86_64.tgz && cd nice-dcv-2023.1-16388-ubuntu2004-x86_64
```

- Ubuntu 22.04(64비트 x86)

```
$ tar -xvzf nice-dcv-2023.1-16388-ubuntu2204-x86_64.tgz && cd nice-dcv-2023.1-16388-ubuntu2204-x86_64
```

- Ubuntu 22.04(64비트 ARM)

```
$ tar -xvzf nice-dcv-2023.1-16388-ubuntu2204-aarch64.tgz && cd nice-dcv-2023.1-16388-ubuntu2204-aarch64
```

5. NICE DCV 서버를 설치합니다.

- Ubuntu 20.04(64비트 x86)

```
$ sudo apt install ./nice-dcv-server_2023.1.16388-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04(64비트 x86)

```
$ sudo apt install ./nice-dcv-server_2023.1.16388-1_amd64.ubuntu2204.deb
```

- Ubuntu 22.04(64비트 ARM)

```
$ sudo apt install ./nice-dcv-server_2023.1.16388-1_arm64.ubuntu2204.deb
```

6. (선택 사항) NICE DCV 버전 2021.2 이상에서 웹 클라이언트를 사용하려면 nice-dcv-web-viewer 패키지를 설치합니다.

- Ubuntu 20.04(64비트 x86)

```
$ sudo apt install ./nice-dcv-web-viewer_2023.1.16388-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04(64비트 x86)

```
$ sudo apt install ./nice-dcv-web-viewer_2023.1.16388-1_amd64.ubuntu2204.deb
```

- Ubuntu 22.04(64비트 ARM)

```
$ sudo apt install ./nice-dcv-web-viewer_2023.1.16388-1_arm64.ubuntu2204.deb
```

7. dcv 사용자를 video 그룹에 추가합니다.

```
$ sudo usermod -aG video dcv
```

8. (선택 사항) 가상 세션을 사용할 계획인 경우 nice-xdcv 패키지를 설치합니다.

- Ubuntu 20.04(64비트 x86)

```
$ sudo apt install ./nice-xdcv_2023.1.565-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04(64비트 x86)

```
$ sudo apt install ./nice-xdcv_2023.1.565-1_amd64.ubuntu2204.deb
```

- Ubuntu 22.04(64비트 ARM)

```
$ sudo apt install ./nice-xdcv_2023.1.565-1_arm64.ubuntu2204.deb
```

9. (선택 사항) GPU 공유를 사용할 계획인 경우 nice-dcv-g1 패키지를 설치합니다.

- Ubuntu 22.04(64비트 x86)

```
$ sudo apt install ./nice-dcv-g1_2023.1.1047-1_amd64.ubuntu2204.deb
```

Note

필요할 경우 nice-dcv-glttest 패키지를 설치할 수 있습니다. 이 패키지에는 가상 세션이 하드웨어 기반 OpenGL을 사용하도록 올바르게 구성되었는지 확인하는 데 사용할 수 있는 간단한 OpenGL 애플리케이션이 포함되어 있습니다.

10. (선택 사항) NICE와 함께 NICE DCV를 사용하려는 경우 패키지를 EnginFrame 설치하십시오.
nice-dcv-simple-external-authenticator

- Ubuntu 20.04(64비트 x86)

```
$ sudo apt install ./nice-dcv-simple-external-
authenticator_2023.1.228-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04(64비트 x86)

```
$ sudo apt install ./nice-dcv-simple-external-
authenticator_2023.1.228-1_amd64.ubuntu2204.deb
```

- Ubuntu 22.04(64비트 ARM)

```
$ sudo apt install ./nice-dcv-simple-external-
authenticator_2023.1.228-1_arm64.ubuntu2204.deb
```

11. (옵션 사항) USB 원격화를 사용하여 특수화된 USB 기기를 지원할 계획인 경우, DCV USB 드라이버를 설치합니다.

DCV USB 드라이버를 설치하려면 서버에 DKMS(Dynamic Kernel Module Support)가 설치되어 있어야 합니다. 다음 명령을 사용하여 DKMS를 설치합니다.

DKMS는 공식 Ubuntu 리포지토리에서 사용할 수 있습니다. 다음 명령을 실행하여 DKMS를 설치합니다.

```
$ sudo apt install dkms
```

DKMS를 설치한 후, 다음 명령을 실행하여 DCV USB 드라이버를 설치합니다.

```
$ sudo dcvusbdriverinstaller
```

12. (선택 사항) 마이크 리디렉션을 지원하려는 경우 pulseaudio-utils 패키지가 시스템에 설치되어 있는지 확인합니다. 다음 명령을 사용하여 설치합니다.

```
$ sudo apt install pulseaudio-utils
```

설치 후 점검

이 주제에서는 NICE DCV를 설치한 후 NICE DCV 서버가 올바르게 구성되었는지 확인하기 위해 수행해야 할 몇 가지 설치 후 점검 사항을 살펴봅니다.

내용

- [NICE DCV 서버에 연결할 수 있는지 확인](#)
- [X 서버에 액세스할 수 있는지 확인](#)
- [DCV GL이 제대로 설치되어 있는지 확인](#)
- [NICE DCV DEB 패키지 서명 확인](#)

NICE DCV 서버에 연결할 수 있는지 확인

기본적으로 NICE DCV는 TCP 포트 8443을 통해 통신하도록 구성됩니다. 서버가 이 포트를 통해 연결 가능한지 확인합니다. 포트 8443을 통한 액세스를 차단하는 방화벽이 있는 경우, NICE DCV 서버의 통신에 사용되는 포트를 변경해야 합니다. 자세한 정보는 [NICE DCV 서버 TCP/UDP 포트 및 수신 주소 변경](#)을 참조하세요.

또한 EC2 인스턴스에 NICE DCV를 설정하는 경우 보안 그룹을 생성해야 합니다. 이는 NICE DCV 서버가 통신하는 포트에 대한 액세스를 가능하게 하기 위한 것입니다. 자세한 내용은 [EC2에서 보안 그룹을 구성하는 방법](#)을 참조하십시오.

X 서버에 액세스할 수 있는지 확인

NICE DCV 콘솔과 가상 세션이 X 서버에 액세스할 수 있는지 확인해야 합니다.

콘솔 세션

NICE DCV 서버가 설치되면 dcv 사용자가 생성됩니다. 이 사용자가 X 서버에 액세스할 수 있는지 확인해야 합니다.

dcv 사용자가 X 서버에 액세스할 수 있는지 확인하려면

다음 명령을 실행합니다:

```
$ sudo DISPLAY=:0 XAUTHORITY=$(ps aux | grep "X.*\-auth" | grep -v grep | sed -n 's/.*-auth \([^ ]*\+\.)*\1/p') xhost | grep "SI:localuser:dcv$"
```

명령이 SI:localuser:dcv 반환하는 경우, dcv 사용자가 X 서버에 액세스할 수 있습니다.

명령이 SI:localuser:dcv를 반환하지 않는 경우, dcv 사용자가 X 서버에 액세스할 수 없습니다. 다음 명령을 실행하여 X 서버를 다시 시작합니다.

- RHEL, CentOS, Amazon Linux 2, Ubuntu 18.x, SUSE Linux Enterprise 12.x

```
$ sudo systemctl isolate multi-user.target
```

```
$ sudo systemctl isolate graphical.target
```

가상 세션

DCV GL 패키지를 설치한 경우, 로컬 사용자가 X 서버에 액세스할 수 있는지 확인해야 합니다. 이를 통해 OpenGL 하드웨어 가속화가 가상 세션에서 올바르게 작동하도록 할 수 있습니다.

로컬 사용자가 X 서버에 액세스할 수 있는지 확인하려면

다음 명령을 실행합니다:

```
$ sudo DISPLAY=:0 XAUTHORITY=$(ps aux | grep "X.*\-auth" | grep -v grep | sed -n 's/.*-auth \([^ ]*\+\.)*\1/p') xhost | grep "LOCAL:$"
```

명령이 LOCAL:을 반환하는 경우, 로컬 사용자가 X 서버에 액세스할 수 있습니다.

명령이 LOCAL:을 반환하지 않는 경우, 로컬 사용자가 X 서버에 액세스할 수 없습니다. 다음 명령을 실행하여 X 서버를 다시 시작하고 DCV GL을 비활성화한 다음 다시 시작합니다.

- RHEL, CentOS, Amazon Linux 2, Ubuntu 18.x, SUSE Linux Enterprise 12.x

```
$ sudo systemctl isolate multi-user.target
```

```
$ sudo dcvgladmin disable
```

```
$ sudo dcvgladmin enable
```

```
$ sudo systemctl isolate graphical.target
```

DCV GL이 제대로 설치되어 있는지 확인

DCV GL 패키지를 설치하면 dcvgldiag 유틸리티가 자동으로 설치됩니다. 이 유틸리티를 사용하여 Linux 서버 구성이 DCV GL 요구 사항을 충족하는지 확인할 수 있습니다.

dcvgldiag 유틸리티를 실행하려면

다음 명령을 사용합니다.

```
$ sudo dcvgldiag
```

유틸리티가 실행 가능한 솔루션과 함께 경고 및 오류 목록을 반환합니다.

NICE DCV DEB 패키지 서명 확인

NICE DCV가 설치되면 Debian 패키지(DEB)에서 서명을 확인할 수 있습니다. 이 확인 프로세스를 수행하려면 GPG 버전 1을 사용해야 합니다.

DEB 패키지 서명을 확인하려면 다음과 같이 하세요.

다음 명령을 사용합니다.

```
gpg1 --import NICE-GPG-KEY-SECRET
dpgk-sig --verify nice-dcv-server_2023.1.16388-1_amd64.deb
```

그러면 서명이 확인되었음을 확인하는 GOODSIG 용어가 포함된 메시지가 반환됩니다. 다음 예제에서는 서명 확인 메시지를 보여줍니다. **## #** 대신 키가 표시됩니다.

```
Processing nice-dcv-server_2017.0.0-1_amd64.deb...
GOODSIG _gpgbuilder Example Key
```

NICE DCV 서버 라이선스

NICE DCV 라이선스 요구 사항은 NICE DCV 서버를 설치 및 사용하는 위치에 따라 다릅니다.

⚠ Important

다음 라이선스 요구 사항은 NICE DCV 버전 2017.0 이상에만 적용됩니다.

NICE DCV 라이선스 요구 사항

주제

- [Amazon EC2의 NICE DCV](#)
- [NICE DCV의 기타 사용 사례](#)
- [Windows Server에 원격으로 액세스하기 위한 Microsoft 라이선스 요구 사항](#)

Amazon EC2의 NICE DCV

Local Zones에서 실행되는 인스턴스를 포함하여 EC2 인스턴스에 NICE DCV 서버를 설치하고 사용하는 데는 라이선스 서버가 필요하지 않습니다. AWS Outposts AWS NICE DCV 서버는 Amazon EC2 인스턴스에서 실행 중인지를 자동으로 감지하고 S3 버킷에 주기적으로 연결하여 유효한 라이선스를 사용할 수 있는지 확인합니다.

인스턴스에 다음 속성이 있는지 확인하세요.

- 인스턴스가 Amazon S3 엔드포인트에 도달할 수 있습니다. 인스턴스에 인터넷 액세스 권한이 있는 경우, Amazon S3 퍼블릭 엔드포인트를 사용하여 연결합니다. 인스턴스가 인터넷에 액세스할 수 없는 경우, HTTPS를 통해 Amazon S3에 연결하도록 허용하는 아웃바운드 보안 그룹 규칙 또는 액세스 제어 목록(ACL) 정책을 사용하여 VPC에 대한 게이트웨이 엔드포인트를 구성합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [게이트웨이 VPC 엔드포인트](#)를 참조하세요. S3 버킷에 연결하는데 문제가 있는 경우 AWS 지식 센터에서 [게이트웨이 VPC 엔드포인트를 사용하여 S3 버킷에 연결할 수 없는 이유](#)를 참조하세요.
- 인스턴스에 필수 Amazon S3 객체에 액세스할 권한이 있습니다. 다음 Amazon S3 액세스 정책을 인스턴스의 IAM 역할에 추가하고 **## ## ##### ## AWS ##** (예:) 으로 교체합니다. us-east-1 자세한 내용은 [IAM 역할 생성](#) 단원을 참조하십시오.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```

    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::dcv-license.region/*"
  }
]
}

```

- Windows 인스턴스를 사용하는 경우 인스턴스가 인스턴스 메타데이터 서비스에 액세스할 수 있는지 확인합니다. NICE DCV 서버에 제대로 라이선스를 부여할 수 있도록 하려면 이 서비스에 대한 액세스가 필요합니다. 인스턴스 메타데이터 서비스 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 메타데이터 및 사용자 데이터](#)를 참조하세요.

사용자 지정 Windows AMI를 사용하는 경우 EC2Config Service(Windows Server 2012 R2 이하) 또는 EC2Launch(Windows Server 2016 이상)를 설치해야 합니다. 이렇게 하면 인스턴스가 인스턴스 메타데이터 서비스에 액세스할 수 있습니다. 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [EC2Config 서비스를 사용하여 Windows 인스턴스 구성](#) 또는 [EC2Launch를 사용하여 Windows 인스턴스 구성](#)을 참조하세요.

Amazon EC2 인스턴스에 NICE DCV 서버를 설치하고 사용 중인 경우, 이 장의 나머지를 건너뛸 수 있습니다. 이 장의 나머지 부분은 NICE DCV 서버의 다른 모든 사용 사례에만 적용됩니다.

NICE DCV의 기타 사용 사례

다른 모든 사용 사례의 경우, NICE DCV 서버를 설치하고 사용하려면 라이선스가 필요합니다. 다음과 같은 라이선스 옵션을 사용할 수 있습니다.

- 자동 평가판 라이선스 - 이 유형의 라이선스는 NICE DCV 서버를 설치할 때 자동으로 설치됩니다. 이 유형의 라이선스는 설치 후 30일 동안 유효합니다. 라이선스가 만료된 후에는 더 이상 서버에서 NICE DCV 세션을 생성하고 호스팅할 수 없습니다. 이러한 라이선스는 단기 테스트 및 평가에 적합합니다. 장기간 테스트하려면 확장 평가판 라이선스를 요청하십시오.

Note

다른 라이선스가 구성되지 않은 경우 NICE DCV 서버는 자동 평가판 라이선스로 기본 설정됩니다.

- 확장 평가판 라이선스 - 확장 평가판 라이선스는 자동 평가판 라이선스에서 제공하는 초기 30일 평가 기간을 연장하는 평가판 라이선스입니다. 기간은 NICE에 따라 결정됩니다. case-by-case 확장 평가판 라이선스는 만료 날짜가 지나면 유효하지 않게 되며, 더 이상 서버에서 NICE DCV 세션을 생성하고 호스팅할 수 없습니다. 확장 평가판 라이선스는 NICE 웹사이트의 [구매 방법](#) 페이지에 나열된

NICE 배포자 또는 리셀러에게 요청해야 합니다. 이 라이선스는 NICE DCV 서버에 설치해야 하는 라이선스 파일로 제공됩니다.

- 프로덕션 라이선스 - 프로덕션 라이선스는 NICE에서 구매하는 정식 라이선스입니다. 프로덕션 라이선스는 라이선스 서버에서 관리하는 부동 라이선스입니다. 부동 라이선스를 사용하면 네트워크에서 여러 NICE DCV 서버를 실행할 수 있습니다. 동시에 모든 서버에서 생성할 수 있는 동시 NICE DCV 세션 수를 제한할 수도 있습니다. 각 동시 NICE DCV 세션마다 라이선스가 하나씩 필요합니다. 프로덕션 라이선스는 RLM(Reprise License Manager) 서버에 설치해야 하는 라이선스 파일로 배포됩니다. 프로덕션 라이선스에는 다음과 같은 두 가지 유형이 있습니다.
- 영구 라이선스 - 영구 라이선스에는 만료일이 없으며 무기한 사용할 수 있습니다.
- 구독 - 구독은 제한된 기간(일반적으로 1년) 동안 유효합니다. 라이선스의 만료일은 라이선스 파일에 표시됩니다. 라이선스가 만료된 후에는 더 이상 NICE DCV 서버에서 NICE DCV 세션을 생성하고 호스팅할 수 없습니다.

NICE DCV 영구 라이선스 또는 구독을 구매하는 방법에 대한 자세한 내용은 NICE 웹사이트의 [구매 방법](#)을 참조하고 해당 리전의 NICE 유통업체 또는 리셀러를 찾아보세요.

라이선스 요구 사항

- NICE DCV 클라이언트에는 라이선스가 필요하지 않습니다.
- NICE DCV 서버 라이선스 파일은 이전 버전의 NICE DCV 서버와 역호환됩니다. 예를 들어, NICE DCV 서버 버전 2021 라이선스를 NICE DCV 서버 버전 2019와 함께 사용할 수 있습니다.
- NICE DCV 서버 버전에는 최소한 동일한 버전의 NICE DCV 서버 라이선스가 필요합니다. 예를 들어 NICE DCV 서버 버전 2021을 사용하는 경우 라이선스 버전 2021 이상이 필요합니다. NICE DCV 서버의 상위 버전으로 업그레이드하는 경우 호환되는 라이선스 파일을 요청해야 합니다. 자세한 내용은 NICE DCV 배포자 또는 리셀러에게 문의하세요.

Note

NICE DCV 서버 호환성에 대한 자세한 내용은 [호환성 고려 사항](#) 섹션을 참조하세요.

Windows Server에 원격으로 액세스하기 위한 Microsoft 라이선스 요구 사항

Microsoft에서는 Windows Server CAL(클라이언트 액세스 라이선스) 외에도 서버의 그래픽 사용자 인터페이스(GUI)에 원격으로 액세스하는 각 사용자에 대해 사용 중인 Windows Server 버전용 Windows Server 원격 데스크톱 서비스(RDS) CAL을 요구합니다. 이는 사용하는 원격 디스플레이 프로토콜과는

무관합니다. NICE DCV를 사용하여 원격 Windows Server 호스트의 GUI에 액세스하는 경우에도 이 라이선스가 필요합니다.

Amazon EC2 인스턴스에서 NICE DCV 서버를 실행하고 [Windows Server AMI](#)를 사용하는 경우, Amazon은 Windows Server CAL에 대한 라이선스 비용을 부담하고, 관리 목적으로만 사용되는 두 개의 Windows Server RDS CAL을 제공합니다. 이는 테스트, 유지 보수 및 관리 전용입니다.

자세한 내용은 [Microsoft 제품 약관 사이트](#)를 참조하세요. Microsoft 소프트웨어의 라이선스 또는 소유권에 대한 질문이 있는 경우 해당 법률팀이나 Microsoft 또는 Microsoft 리셀러에게 문의하세요.

확장 평가판 라이선스 설치

NICE에서 확장 평가판 라이선스를 요청하면 라이선스를 정의하는 `license.lic` 파일을 받게 됩니다.

확장 평가판 라이선스를 설치하려면

`license.lic` 파일을 서버의 다음 폴더에 놓습니다.

- Windows 서버

```
C:\Program Files\NICE\DCV\Server\license\license.lic
```

- Linux 서버

```
/usr/share/dcv/license/license.lic
```

또는 `license.lic`를 서버의 다른 폴더에 배치하려면 라이선스 파일의 전체 경로를 지정하도록 `license-file` 구성 파라미터를 업데이트해야 합니다.

주제

- [Windows Server에서 라이선스 경로 변경](#)
- [Linux 서버에서 라이선스 경로 변경](#)

Windows Server에서 라이선스 경로 변경

Windows 서버에서 **license-file** 구성 파라미터를 업데이트하려면

1. Windows 레지스트리 편집기를 엽니다.

2. HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/license/ 키로 이동하여 license-file 파라미터를 선택합니다.

레지스트리 키에 license-file 파라미터가 없는 경우 하나를 만듭니다.

- a. 왼쪽 창에서 라이선스 키의 컨텍스트(오른쪽 클릭) 메뉴를 열고 새로 만들기, 문자열 값을 선택합니다.
 - b. 이름에 license-file을 입력하고 Enter 키를 누릅니다.
3. license-file 파라미터를 엽니다. 값 데이터에 license.lic 파일의 전체 경로를 입력합니다.
 4. 확인을 선택하고 Windows 레지스트리 편집기를 닫습니다.

Linux 서버에서 라이선스 경로 변경

Linux 서버에서 **license-file** 구성 파라미터를 업데이트하려면

1. /etc/dcv/로 이동하고 원하는 텍스트 편집기에서 dcv.conf를 엽니다.
2. license-file 섹션에서 [license] 파라미터를 찾은 후, 기존 경로를 license.lic 파일의 새 전체 경로로 바꿉니다.

license-file 섹션에 [license] 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
license-file = "/custom-path/license.lic"
```

3. 파일을 저장하고 닫습니다.

프로덕션 라이선스 설치

이 주제의 다음 섹션에서는 프로덕션 라이선스(영구 라이선스 또는 구독)를 구매하고 사용하는 방법을 설명합니다.

주제

- [1단계: RLM 서버 설치](#)
- [2단계: RLM 서버의 호스트 ID 가져오기](#)
- [3단계: 영구 라이선스 또는 구독 구매](#)
- [4단계: 라이선스 파일 수정](#)

- [5단계: RLM 서버 구성](#)
- [6단계: NICE DCV 서버 구성](#)

1단계: RLM 서버 설치

영구 라이선스 또는 구독을 구매하면 라이선스 조항을 정의하는 라이선스 파일이 제공됩니다. RLM(Reprise License Manager) 서버에 라이선스 파일을 설치해야 합니다.

RLM에 대한 자세한 내용은 [Reprise Software](#) 웹 사이트를 참조하십시오.

주제

- [Windows에 RLM 서버 설치](#)
- [Linux에 RLM 서버 설치](#)

Windows에 RLM 서버 설치

Windows에 RLM 서버를 설치하려면

1. [Reprise Software 웹 사이트](#)에서 RLM 라이선스 관리 번들을 다운로드합니다.

Note

NICE DCV 버전 2022.1부터는 RLM 서버 14 이상 버전이 필요합니다. 이전 버전에는 RLM 12 이상 버전이 필요합니다.
안정적인 최신 버전의 RLM 라이선스 관리 번들을 설치하는 것이 좋습니다.

2. RLM 라이선스 관리 번들을 C:\RLM에 설치합니다.

Linux에 RLM 서버 설치

Linux에 RLM 서버를 설치하려면

1. [Reprise Software 웹 사이트](#)에서 RLM 라이선스 관리 번들을 다운로드합니다.

Note

NICE DCV 버전 2022.1부터는 RLM 서버 14 이상 버전이 필요합니다. 이전 버전에는 RLM 12 이상 버전이 필요합니다.

안정적인 최신 버전의 RLM 라이선스 관리 번들을 설치하는 것이 좋습니다.

2. 사용자 그룹 및 rlm 사용자를 생성합니다. 이는 유효한 사용자 또는 서비스 계정일 수 있습니다. 이 값에는 루트 계정을 사용하지 않는 것이 좋습니다.

```
$ groupadd -r rlm
```

```
$ useradd -r -g rlm -d "/opt/nice/rlm" -s /sbin/nologin -c "RLM License Server" rlm
```

3. RLM 서버에 필요한 /opt/nice/rlm 및 /opt/nice/rlm/license 디렉터리를 생성합니다.

```
$ mkdir -p /opt/nice/rlm/license
```

4. RLM 라이선스 관리 번들의 내용을 /opt/nice/rlm/에 압축 해제하고 rlm 사용자가 파일을 소유하는지 확인합니다.

```
$ tar xvf x64_l1.admin.tar.gz -C /opt/nice/rlm/ --strip-components 1
```

```
$ chown -R rlm:rlm /opt/nice/rlm
```

2단계: RLM 서버의 호스트 ID 가져오기

RLM 서버를 설치한 후에는 RLM 서버의 호스트 ID를 가져와야 합니다. 영구 라이선스 또는 구독을 구매할 때 이 호스트 ID를 제공해야 합니다.

Windows에서 RLM 서버 호스트 ID 가져오기

서버의 호스트 ID를 가져오려면 명령 프롬프트를 엽니다.

C:\RLM\으로 이동하여 다음 명령을 실행합니다.

```
C:\> rlmutil.exe rlmhostid ether
```

이 명령은 다음과 같이 RLM 서버의 호스트 ID를 반환합니다.

```
Hostid of this machine: 06814example
```

호스트 ID를 기록해 둡니다. 이 정보는 다음 단계에 필요합니다.

Linux에서 RLM 서버 호스트 ID 가져오기

서버의 호스트 ID를 단일 반환으로 가져옵니다.

1. /opt/nice/rlm/로 이동합니다.
2. 다음 명령을 실행합니다:

```
$ ./rlmutil rlmhostid ether
```

이 명령은 다음과 같이 각 네트워크 인터페이스에 대한 RLM 서버의 호스트 ID를 반환합니다.

3. 호스트 ID를 기록합니다. 이 정보는 다음 단계에 필요합니다.

Example

이 절차가 실행된 후 다음과 같은 단일 ID가 반환됩니다.

```
Hostid of this machine: 0a1b2c3d4e5f
```

그러면 이 ID가 기록되어 DCV용 라이선스를 구매하는 데 사용됩니다.

서버의 호스트 ID를 다중 반환으로 가져옵니다.

1. /opt/nice/rlm/로 이동합니다.
2. 다음 명령을 실행합니다:

```
$ ./rlmutil rlmhostid ether
```

여러 개의 ID가 목록으로 반환됩니다.

3. 다음 명령을 실행합니다.

```
iface=$(route -n | grep " UG " | tr -s " " | cut -d" " -f8)
ip link show $iface | grep link/ether | tr -s " " | cut -d" " -f3 | tr -d ":"
```

이 명령은 게이트웨이 네트워크 인터페이스에 대한 RLM 서버의 호스트 ID를 반환합니다.

4. 호스트 ID를 기록합니다. 이 정보는 다음 단계에 필요합니다.

Example

이 절차가 실행된 후 여러 ID가 다수의 ID 목록으로 반환됩니다.

```
Hostid of this machine: 0a1b2c3d4e5f 1b2c3d4e5f6a 2c3d4e5f6a7b 3d4e5f6a7b8c
```

인터페이스 명령이 실행되고 다음 ID가 반환됩니다.

```
Hostid of this machine: 0a1b2c3d4e5f
```

그러면 이 ID가 기록되어 DCV용 라이선스를 구매하는 데 사용됩니다.

3단계: 영구 라이선스 또는 구독 구매

NICE DCV 영구 라이선스 또는 구독을 구매하는 방법에 대한 자세한 내용은 NICE 웹사이트의 [구매 방법](#)을 참조하고 해당 리전의 NICE 유통업체 또는 리셀러를 찾아보세요.

RLM 서버의 호스트 ID를 제공해야 합니다. 호스트 ID는 NICE에서 제공하는 라이선스 파일에 포함되어 있습니다.

4단계: 라이선스 파일 수정

NICE DCV 영구 라이선스 또는 구독을 구매하면 라이선스를 정의하는 `license.lic` 파일을 받게 됩니다. `license.lic` 테이블에 포함되는 정보는 다음과 같습니다.

- RLM 서버의 호스트 이름
- 라이선스를 구매할 때 제공한 RLM 서버의 호스트 ID
- RLM 서버의 TCP 포트 번호. 기본값은 5053입니다.
- ISV 포트 번호입니다. 이 포트는 RLM 서버가 NICE DCV 라이선스 요청을 수신하는 선택적 포트입니다. 포트를 지정하지 않으면 시작 시 RLM이 임의의 포트를 선택합니다.
- 라이선스에 포함된 NICE DCV 제품과 각 제품에 대한 다음 세부 정보:
 - 라이선스가 적용되는 메이저 버전(예: 2017 NICE DCV 제품의 경우 2017)
 - 만료 날짜. Permanent는 라이선스가 만료되지 않음을 나타냅니다.
 - 최대 동시 세션 수입니다(예: 서버에 10개의 동시 세션이 있는 경우 10).
 - 라이선스 체크섬.
 - 라이선스 서명.

다음 코드 블록은 `license.lic` 파일의 형식을 보여 줍니다.

```
HOST RLM_server_hostname RLM_server_host_id RLM_server_port
ISV nice port=port_number
LICENSE product_1 major_version expiration_date concurrent_sessions share=hi
  _ck=checksum sig="signature"
LICENSE product_2 major_version expiration_date concurrent_sessions share=hi
  _ck=checksum sig="signature"
```

다음 코드 블록은 ISV 포트가 생략된 license.lic 파일의 예를 보여줍니다. 라이선스 파일에는 DCV 및 dcv-gl이라는 두 개의 NICE 제품에 대한 라이선스가 포함되어 있습니다.

```
HOST My-RLM-server abcdef123456 5053
ISV nice
LICENSE nice dcv 2017 permanent 10 share=hi _ck=456789098a
  sig="abcdefghijklmnopqrstuvwxy1234567890abcdefghijklmnopqrstuvwxy1234567890ab"
LICENSE nice dcv-gl 2017 permanent 10 share=hi _ck=123454323x
  sig="1234567890abcdefghijklmnopqrstuvwxy1234567890abcdefghijklmnopqrstuvwxy12"
```

license.lic 파일을 편집하려면

1. 선호하는 텍스트 편집기로 파일을 엽니다.
2. HOST로 시작하는 파일의 첫 줄에 RLM 서버의 호스트 이름과 TCP 포트 번호를 추가합니다.

Warning

*RLM_server_host_id*는 라이선스를 구매할 때 제공한 호스트 ID입니다.
*RLM_server_host_id*를 편집할 수 없습니다.

3. (선택 사항) port=*port_number*를 추가하여 ISV로 시작하는 파일의 행에 ISV 포트 번호를 추가합니다. 이 포트는 DCV 서버와 통신할 수 있도록 하는 데 필요합니다.

ISV 포트를 지정하지 않으려는 경우 port=*port_number*를 생략합니다. ISV 포트를 지정하지 않으면 RLM은 시작할 때마다 임의 포트를 사용합니다.

Warning

임의로 선택한 포트를 사용하지 못하도록 방화벽을 설정한 경우 HOST 행에 지정된 RLM 포트 외에도 이 포트를 지정하고 이 포트를 활성화하도록 방화벽을 구성해야 합니다.

4. 파일을 저장하고 닫습니다.

⚠ Warning

라이선스 파일의 다른 부분을 편집하면 파일의 서명이 손상되고 라이선스가 무효화됩니다.

5단계: RLM 서버 구성

라이선스 파일을 수정한 후에는 파일을 RLM 서버에 배치한 다음 RLM 서비스를 시작해야 합니다.

주제

- [Windows에서 RLM 서버 구성](#)
- [Linux에서 RLM 서버 구성](#)

Windows에서 RLM 서버 구성

Windows에서 RLM 서버를 구성하려면

1. RLM 서버에 연결합니다.
2. 편집된 `license.lic` 파일을 `C:\RLM\license\`에 복사합니다.
3. `C:\Program Files\NICE\DCV\Server\license\nice.set` 파일을 NICE DCV 서버에서 복사하여 RLM 서버의 `C:\RLM\` 폴더에 배치합니다.
4. RLM 서버를 Windows 서비스로 설치합니다.

```
C:\> rlm.exe -nows -dlog C:\RLM\rlm.log -c C:\RLM\license -install_service -
service_name dcv-rlm
```

RLM 시작 옵션에 대한 자세한 내용은 [Reprise 소프트웨어 라이선스 관리자\(RLM\)](#) 제품 페이지를 참조하세요.

5. RLM 서버를 시작합니다.

```
C:\> net start dcv-rlm
```

6. RLM 서버가 실행 중인지 확인합니다.

- a. 선호하는 텍스트 편집기로 `C:\RLM\nice.dlog`를 열고 다음 줄이 나타나는지 확인합니다.

```
date_time (nice) Server started on license1 (hostid: host_id) for: dcv dcv-gl
```

Note

rlm.log 파일의 내용은 RLM 서버 버전에 따라 조금씩 다를 수 있습니다.

- b. 다음 명령을 실행합니다.

```
C:\RLM\rlmutil rlmstat -a -c rlm_server_hostname@5053
```

이 명령은 RLM 서버에 대한 정보를 반환합니다.

Linux에서 RLM 서버 구성

Linux에서 RLM 서버를 구성하려면

1. 편집된 license.lic 파일을 /opt/nice/rlm/license/에 복사합니다.
2. /usr/share/dcv/license/nice.set 파일을 NICE DCV 서버에서 복사하여 RLM 서버의 /opt/nice/rlm에 배치합니다.
3. RLM 서버 서비스를 생성하고 시작할 때 자동으로 시작되는지 확인합니다.
 - a. dcv-rlm 폴더에 /opt/nice/rlm/이라는 파일을 만듭니다.

```
$ touch /opt/nice/rlm/dcv-rlm
```

- b. 원하는 텍스트 편집기를 사용하여 파일을 열고 다음 스크립트를 추가합니다. 파일을 저장하고 닫습니다.

```
#!/bin/sh
# chkconfig: 35 99 01
# description: The Reprise License Manager daemon.
# processname: dcv-rlm

### BEGIN INIT INFO
# Provides: dcv-rlm
# Required-Start: $local_fs $remote_fs $syslog
# Required-Stop: $local_fs $remote_fs $syslog
# Default-Start: 3 4 5
# Default-Stop: 0 1 2 6
# Short-Description: The Reprise License Manager daemon.
# Description: A service that runs the Reprise License Manager daemon.
```

```
### END INIT INFO

# user used to run the daemon
RLM_USER="rlm"

# root of rlm installation
RLM_ROOT="/opt/nice/rlm"

# license directory (license files should have .lic extension)
RLM_LICENSE_DIR="/opt/nice/rlm/license"

# log file
RLM_LOG_FILE="/var/log/rlm.log"

_getpid() {
    pidof -o $$ -o $PPID -o %PPID -x "$1"
}

start() {
    echo -n "Starting rlm: "
    touch ${RLM_LOG_FILE}
    chown "${RLM_USER}" ${RLM_LOG_FILE}
    su -p -s /bin/sh "${RLM_USER}" -c "${RLM_ROOT}/rlm -c ${RLM_LICENSE_DIR} \
        -nows -dlog +${RLM_LOG_FILE} &"
    if [ $? -ne 0 ]; then
        echo "FAILED"
        return 1
    fi
    echo "OK"
}

stop() {
    echo -n "Stopping rlm: "
    pid=`_getpid ${RLM_ROOT}/rlm`
    if [ -n "$pid" ]; then
        kill $pid >/dev/null 2>&1
        sleep 3
        if [ -d "/proc/$pid" ] ; then
            echo "FAILED"
            return 1
        fi
    fi
    echo "OK"
}
```

```
status() {
    pid=`_getpid ${RLM_ROOT}/rlm`
    if [ -z "$pid" ]; then
        echo "rlm is stopped"
        return 3
    fi
    echo "rlm (pid $pid) is running..."
    return 0
}

restart() {
    stop
    start
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status
        ;;
    restart)
        restart
        ;;
    *)
        echo $"Usage: $0 {start|stop|status|restart}"
        exit 1
esac

exit $?

# ex:ts=4:et:
```

- c. 스크립트를 실행 파일로 만들고, 이를 `/etc/init.d/`에 복사한 후, `chkconfig` 유틸리티에 추가합니다.

```
chmod +x /opt/nice/rlm/dcv-rlm
```



```
cp -a /opt/nice/rlm/dcv-rlm /etc/init.d/
```

```
chkconfig --add dcv-rlm
```

4. RLM 서버를 시작합니다.

```
$ service dcv-rlm start
```

5. RLM 서버가 실행되고 정상 작동하는지 확인합니다. 원하는 텍스트 편집기로 var/log/rlm.log를 열고 다음 줄이 표시되는지 확인합니다.

```
date_time (nice) Server started on license1 (hostid: host_id) for: dcv dcv-gl
```

Note

rlm.log 파일의 내용은 RLM 서버 버전에 따라 조금씩 다를 수 있습니다.

6단계: NICE DCV 서버 구성

RLM 서버를 사용하도록 NICE DCV 서버를 구성합니다. 이렇게 하려면 NICE DCV 서버에서 license-file 구성 파라미터를 구성해야 합니다.

license-file 파라미터는 연결할 RLM 서버의 사양에 따라 *RLM_server_port@RLM_server* 형식으로 설정해야 합니다. RLM 서버는 호스트 이름 또는 IP 주소로 지정할 수 있습니다. 명시적으로 구성되지 않은 경우 기본적으로 RLM 서버 포트 5053이 사용됩니다.

여러 RLM 서버를 사용하는 경우 Linux에서는 :으로, Windows에서는 ;으로 구분하여 여러 RLM 서버 사양 목록을 지정할 수 있습니다. 그러면 서버는 해당 RLM 서버와 한 개의 연결을 설정할 수 있을 때까지 각 서버에 차례로 연결을 시도합니다. 예를 들어 기본 RLM 서버에 연결할 수 없는 경우 RLM 페일오버 서버를 사용하여 인계할 때 특히 유용할 수 있습니다. 이 경우 *RLM_primary_server_port@RLM_primary_server:RLM_failover_server_port@RLM_failover_server* 같은 형식으로 라이선스를 지정할 수 있습니다.

Note

NICE DCV 서버가 Windows에 설치되어 있는 경우 사양의 항목을 ;으로 구분해야 합니다.

주제

- [Windows NICE DCV 서버 구성](#)
- [Linux NICE DCV 서버 구성](#)

Windows NICE DCV 서버 구성

Windows 서버에서 **license-file** 구성 파라미터를 구성하려면

1. Windows 레지스트리 편집기를 엽니다.
2. HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/license/ 키로 이동하여 license-file 파라미터를 선택합니다.

레지스트리 키에 license-file 파라미터가 없는 경우 이 파라미터를 생성해야 합니다.

- a. 왼쪽 창에서 라이선스 키의 컨텍스트(오른쪽 클릭) 메뉴를 열고 새로 만들기, 문자열 값을 선택합니다.
 - b. 이름에 license-file을 입력하고 Enter 키를 누릅니다.
3. license-file 파라미터를 엽니다. 값 데이터에 RLM 서버의 포트 번호와 호스트 이름을 *RLM_server_port@RLM_server* 형식으로 입력합니다. 여러 RLM 서버에 대한 연결을 설정해야 하는 경우 위의 내용을 확인하세요.
 4. 확인을 선택하고 Windows 레지스트리 편집기를 닫습니다.

Linux NICE DCV 서버 구성

Linux 서버에서 **license-file** 구성 파라미터를 구성하려면

1. /etc/dcv/로 이동하고 원하는 텍스트 편집기에서 dcv.conf를 엽니다.
2. [license] 섹션에서 license-file 파라미터를 찾습니다. 그런 다음, 기존 경로를 *RLM_server_port@RLM_server* 형식의 RLM 서버 포트와 호스트 이름으로 바꿉니다.

license-file 섹션에 [license] 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
license-file = "RLM_server_port@RLM_server"
```

여러 RLM 서버에 대한 연결을 설정해야 하는 경우 위의 내용을 확인하세요.

3. 파일을 저장하고 닫습니다.

프로덕션 라이선스 업데이트

NICE DCV 서버는 몇 분마다 RLM 서버의 라이선스를 확인합니다. RLM 서버에서 라이선스가 업데이트되는 경우 NICE DCV 서버는 실행 중인 세션의 사용된 라이선스를 자동으로 업데이트합니다. 다음 절차는 RLM에서 DCV 라이선스를 업데이트하는 방법을 자세히 설명합니다.

RLM 서버에서 DCV 라이선스를 업데이트하려면 다음과 같이 하세요.

1. 이전에 [설치된](#) 라이선스 파일을 업데이트합니다. Linux에서는 /opt/dcv/rlm/license/license.lic에, Windows에서는 C:\RLM\license\license.lic에 배치되었어야 합니다.
2. rlmutil rlmreread 명령을 실행하여 라이선스 파일을 강제로 다시 로드합니다.

RLM 서버에서 라이선스가 업데이트된 후 NICE DCV 서버는 몇 분(보통 5분 이내) 내에 새 라이선스의 사용을 확인해야 합니다.

NICE DCV 버전 2021.0부터 관리자 권한으로 다음 명령을 사용하여 라이선스 업데이트를 즉시 적용할 수 있습니다.

```
$ dcv reload-licenses
```

NICE DCV 서버 업그레이드

다음 주제에서는 NICE DCV 서버를 업그레이드하는 방법에 대해 설명합니다.

내용

- [호환성 고려 사항](#)
- [Windows에서 NICE DCV 서버 업그레이드](#)
- [Linux에서 NICE DCV 서버 업그레이드](#)

호환성 고려 사항

NICE DCV 서버 버전 2017 이상은 NICE DCV 클라이언트 버전 2017 이상과 호환됩니다.

Note

온프레미스 및 비EC2 기반 서버의 NICE DCV 서버 라이선스 호환성 요구 사항에 대한 자세한 내용은 [라이선스 요구 사항](#) 섹션을 참조하세요.

Windows에서 NICE DCV 서버 업그레이드

Windows에서 NICE DCV 서버를 업그레이드하려면 다음과 같이 하세요.

1. RDP 클라이언트를 사용하여 NICE DCV 서버에 관리자로 연결합니다.
2. 실행 중인 NICE DCV 세션이 없는지 확인합니다. `dcv list-sessions` NICE DCV 명령을 사용하여 실행 중인 세션을 확인합니다. 실행 중인 세션이 있는 경우 `dcv close-session` NICE DCV 명령을 사용하여 세션을 중지합니다.
3. 실행 중인 세션이 없는지 확인한 후 NICE DCV 서버를 중지합니다. 자세한 정보는 [NICE DCV 서버 중지](#)를 참조하세요.
4. NICE DCV 서버 구성을 백업합니다. 레지스트리 편집기를 열고 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv`로 이동한 다음, `dcv` 키를 마우스 오른쪽 버튼으로 클릭하고 내보내기를 선택합니다.
5. [NICE](#) 웹사이트에서 최신 버전의 NICE DCV 서버를 다운로드합니다.
6. 3단계부터 시작하여 [마법사 사용](#)에 설명된 단계를 따릅니다.
7. 설치가 완료된 후 NICE DCV 서버 구성이 여전히 올바른지 확인합니다. 레지스트리 편집기를 열고 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv`로 이동한 다음, 파라미터를 4단계에서 내보낸 구성과 비교합니다.
8. 새 NICE DCV 세션을 시작하여 NICE DCV 서버를 테스트합니다. 자세한 정보는 [NICE DCV 세션 시작](#)를 참조하세요.

Linux에서 NICE DCV 서버 업그레이드

Linux에서 NICE DCV 서버를 업그레이드하려면 다음과 같이 하세요.

1. SSH를 사용하여 `root` 사용자로 서버에 로그인합니다.
2. 실행 중인 NICE DCV 세션이 없는지 확인합니다. `dcv list-sessions` NICE DCV 명령을 사용하여 실행 중인 세션을 확인합니다. 실행 중인 세션이 있는 경우 `dcv close session` NICE DCV 명령을 사용하여 세션을 중지합니다.

3. 실행 중인 세션이 없는지 확인한 후 NICE DCV 서버를 중지합니다. 자세한 정보는 [NICE DCV 서버 중지](#)을 참조하세요.
4. NICE DCV 서버 구성을 백업합니다. /etc/dcv/dcv.conf 파일을 안전한 위치에 복사합니다.
5. [NICE DCV 서버 설치](#)에 설명된 단계를 따릅니다.
6. 설치가 완료된 후 NICE DCV 서버 구성이 여전히 올바른지 확인합니다. 4단계에서 복사한 파일을 열고 이 파일을 /etc/dcv/dcv.conf 파일과 비교합니다.
7. 새 NICE DCV 세션을 시작하여 NICE DCV 서버를 테스트합니다. 자세한 정보는 [NICE DCV 세션 시작](#)을 참조하세요.

NICE DCV 서버 제거

다음 주제에서는 NICE DCV 서버를 제거하는 방법에 대해 설명합니다.

내용

- [Windows에서 NICE DCV 서버 제거](#)
- [Linux에서 NICE DCV 서버 제거](#)

Windows에서 NICE DCV 서버 제거

Windows에서 NICE DCV 서버를 제거하려면 다음과 같이 하세요.

1. RDP 클라이언트를 사용하여 NICE DCV 서버에 관리자로 연결합니다.
2. 실행 중인 NICE DCV 세션이 없는지 확인합니다. dcv list-sessions NICE DCV 명령을 사용하여 실행 중인 세션을 확인합니다. 실행 중인 세션이 있는 경우 dcv close session NICE DCV 명령을 사용하여 세션을 중지합니다.
3. 실행 중인 세션이 없는지 확인한 후 NICE DCV 서버를 중지합니다. 자세한 정보는 [NICE DCV 서버 중지](#)을 참조하세요.
4. Windows 설정 애플리케이션을 열고 앱 및 기능 패널로 이동합니다.
5. NICE DCV 서버를 선택한 다음 제거를 누릅니다.
6. (선택 사항) NICE DCV 서버에서 생성된 모든 로그 파일을 제거할 수도 있습니다. 제거가 완료되면 C:\\NICEProgramData\\dcv\ 로 이동하여 로그 폴더를 삭제합니다.

Linux에서 NICE DCV 서버 제거

NICE DCV 서버는 호스트 서버의 운영 체제에 따라 일련의 RPM 또는 .deb 패키지를 사용하여 설치됩니다.

Note

NICE DCV 서버를 제거하려면 루트 사용자로 로그인해야 합니다.

Linux에서 NICE DCV 서버를 제거하려면 다음과 같이 하세요.

1. 실행 중인 NICE DCV 세션이 없는지 확인합니다. `dcv list-sessions NICE DCV` 명령을 사용하여 실행 중인 세션을 확인합니다. 실행 중인 세션이 있는 경우 `dcv close session NICE DCV` 명령을 사용하여 세션을 중지합니다.
2. 실행 중인 세션이 없는지 확인한 후 NICE DCV 서버를 중지합니다. 자세한 정보는 [NICE DCV 서버 중지](#)를 참조하세요.
3. NICE DCV 서버 패키지를 제거합니다. 설치를 수행한 방법에 따라 일부 패키지는 시스템에 설치되어 있지 않을 수 있으며 명령에서 생략할 수 있습니다. 선택 패키지 목록은 [Linux에서 NICE DCV 서버 설치](#) 섹션을 참조하세요.

Amazon Linux 2 and RHEL, CentOS

```
$ sudo yum remove nice-dcv-server nice-xdcv nice-dcv-gl nice-dcv-gltest nice-dcv-simple-external-authenticator
```

SLES 12.x/15.x

```
$ sudo zypper remove nice-dcv-server nice-xdcv nice-dcv-gl nice-dcv-gltest nice-dcv-simple-external-authenticator
```

Ubuntu 22.04

```
$ sudo apt remove nice-dcv-server nice-xdcv nice-dcv-gl nice-dcv-gltest nice-dcv-simple-external-authenticator
```

4. (선택 사항) NICE DCV 서버에서 생성된 모든 로그 파일을 제거할 수도 있습니다. 제거가 완료되면 `/var/log`로 이동하여 `dcv` 폴더를 삭제합니다.

이미징 NICE DCV 서버

[Amazon EC2 인스턴스를 사용자 지정한 후 이러한 변경 사항을 Amazon 머신](#) 이미지 (AMI) 로 캡처할 수 있습니다. 이 기능을 사용하면 필요할 때 단일 AMI에서 모두 동일한 구성으로 여러 인스턴스를 시작할 수 있습니다. 고성능 원격 디스플레이 프로토콜로 안전하게 스트리밍해야 하는 경우 Amazon EC2 인스턴스의 이미지를 생성하기 전에 운영 체제에 NICE DCV를 추가할 수 있습니다. NICE DCV 구성이 이미지에 포함되어 있어 이미지 수준에서 사업부를 분리하거나 배포된 인스턴스에서 특정 DCV 구성을 설정할 수 있습니다.

예를 들어 단일 AMI에서 여러 Amazon EC2 인스턴스를 배포하는 경우 로컬 사용자 계정에 대한 자동 콘솔 생성을 사용하고 최종 사용자에게 NICE DCV 권한을 위임할 수 있습니다. 또는 [NICE DCV 세션 관리자와 같은 브로커를 사용하여 NICE DCV 세션](#) 생성을 대규모로 관리할 수도 있습니다.

다음 두 가지 방법 중 하나로 NICE DCV AMI를 생성할 수 있습니다.

NICE DCV 이미지 만들기

먼저 시스템에 NICE DCV가 설치되어 있어야 합니다. [그렇지 않은 경우, 시스템이 NICE DCV에서 지원되는지 확인한 다음 설치 지침을 따르십시오.](#) NICE DCV를 [설치하고](#) 구성한 후 인스턴스의 [AMI를 가져옵니다.](#)

또는 Windows [또는 Linux용](#) NICE DCV 사전 요구 사항을 충족한 경우 아마존에서 관리하는 Image Builder NICE DCV 구성 요소를 실행하여 NICE DCV를 설치하고 구성할 수 있습니다. 다음 작업을 수행하여 구성 요소를 검색할 수 있습니다.

1. [Amazon EC2 Image Builder](#) 콘솔 내에서 구성 요소 페이지로 이동합니다.
2. 필터 소유자 드롭다운 메뉴를 선택하고 빠른 시작 (Amazon 관리) 을 선택합니다.
3. 필터 텍스트 상자를 사용하여 또는 를 검색하십시오. dcv-server-windows dcv-server-linux
4. 구성 요소의 하이퍼링크를 선택합니다.
5. NICE DCV 구성 요소 페이지의 콘텐츠 섹션에서 구성 요소 콘텐츠를 검색합니다.
6. [AWS 태스크 오케스트레이터 및 Executor](#) (AWSTOE) 를 사용하여 인스턴스에서 구성 요소를 로컬로 실행합니다.

Note

자세한 내용은 [시작하기](#)를 참조하십시오. AWSTOE

구성 요소 내 매개 변수 사용에 대한 내용은 아래 섹션을 참조하십시오.

이미지 파이프라인에 NICE DCV 추가

[EC2 Image Builder](#) 레시피는 이미지를 사용자 지정하고 모든 것이 예상대로 작동하는지 확인하기 위해 추가하는 구성 요소 집합과 함께 새 이미지를 생성하기 위한 시작점으로 사용할 기본 이미지를 정의합니다. 이 레시피에서 `dcv-server-windows` 또는 `dcv-server-linux` 구성 요소를 선택하여 파이프라인 내 NICE DCV 설치를 자동화합니다. 이러한 구성 요소 중 하나를 선택하면 요구 사항에 맞게 매개변수를 미세 조정할 수 있습니다.

Note

Linux의 경우 모든 [사전 요구 사항을 충족해야](#) 합니다. 이 작업은 기본 AMI 또는 이전 Image Builder 구성 요소에서 수행할 수 있습니다.

파라미터

Windows

- `sessionOwner`—자동으로 생성된 세션의 기본 소유자를 설정합니다. 지정하지 않으면 자동 콘솔 생성이 비활성화됩니다. 자세한 내용은 NICE DCV [관리 가이드의 자동 콘솔 세션 활성화를](#) 참조하십시오.
- `dcvPermissions`—세션의 NICE DCV 권한을 설정합니다. 자세한 내용은 DCV 관리 [안내서의 권한 파일 작업을](#) 참조하십시오.

Linux

- `SessionOwner`—자동으로 생성된 세션의 기본 소유자를 설정합니다. 지정하지 않으면 자동 콘솔 생성이 비활성화됩니다. 자세한 내용은 NICE DCV [관리 가이드의 자동 콘솔 세션 활성화를](#) 참조하십시오.
- `Packages`—설치할 NICE DCV 패키지를 정의합니다. 비어 있는 경우 사용 가능한 모든 NICE DCV 패키지가 설치됩니다. 자세한 내용은 [NICE DCV 관리 안내서의 Linux에 NICE DCV 서버 설치를](#) 참조하십시오.

구성 요소를 수정하려면 [새 구성 요소 버전을 만들](#) 수 있습니다.

NICE DCV 서버 관리

NICE DCV 서버를 시작, 중지 또는 구성하려면 관리자(Windows) 또는 루트(Linux)로 로그인해야 합니다.

주제

- [NICE DCV 서버 시작](#)
- [NICE DCV 서버 중지](#)
- [QUIC UDP 전송 프로토콜 활성화](#)
- [NICE DCV 서버 TCP/UDP 포트 및 수신 주소 변경](#)
- [TLS 인증서 관리](#)
- [유휴 클라이언트 연결 해제](#)
- [Linux NICE DCV 서버에서 GPU 공유 활성화](#)
- [터치스크린 및 스타일러스 지원 활성화](#)
- [게임패드 지원 활성화](#)
- [USB 원격화 활성화](#)
- [스마트 카드 캐싱 구성](#)
- [WebAuthn 리디렉션 구성](#)
- [세션 스토리지 활성화](#)
- [Linux NICE DCV 서버에서 프린터 설정](#)
- [Linux NICE DCV 서버에서 클립보드 구성](#)
- [다중 채널 오디오 구성](#)
- [HTTP 헤더 구성](#)
- [NICE DCV 인증 구성](#)
- [NICE DCV 권한 부여 구성](#)

NICE DCV 서버 시작

세션을 호스팅하려면 NICE DCV 서버가 실행되고 있어야 합니다.

기본적으로 NICE DCV 서버는 호스팅되는 서버가 시작될 때마다 시작됩니다. NICE DCV 서버를 설치할 때 자동 시작을 비활성화하도록 선택한 경우 서버를 수동으로 시작하거나 자동 시작을 다시 설정해야 합니다. 두 방법 중 하나를 수행하려면 다음 절차 중 하나를 따르세요.

Windows NICE DCV server

Microsoft Management Console의 Services 스냅인을 사용하여 NICE DCV 서버를 수동으로 시작합니다.

Windows에서 NICE DCV 서버를 시작하려면 다음과 같이 하세요.

1. MMC(Microsoft Management Console)의 Services 스냅인을 엽니다.
2. 오른쪽 창에서 DCV Server(DCV 서버)를 엽니다.
3. 시작을 선택합니다.

Note

서버가 이미 실행 중인 경우 시작 버튼이 비활성화되어 있습니다.

Microsoft Management Console의 Services 스냅인을 사용하여 자동 시작을 구성합니다.

Windows에서 NICE DCV 서버가 자동으로 시작되도록 구성하려면 다음과 같이 하세요.

1. MMC(Microsoft Management Console)의 Services 스냅인을 엽니다.
2. 오른쪽 창에서 DCV Server(DCV 서버)를 엽니다.
3. Startup service(시작 서비스)에서 Automatic(자동)을 선택합니다.

Linux NICE DCV server

명령줄을 사용하여 NICE DCV 서버를 수동으로 시작합니다.

Linux에서 NICE DCV 서버를 시작하려면 다음과 같이 하세요.

다음 명령을 사용합니다.

- RHEL, CentOS, SUSE Linux Enterprise 12, Ubuntu 18.x

```
$ sudo systemctl start dcvserver
```

명령줄을 사용하여 NICE DCV 서버가 자동으로 시작되도록 구성합니다.

Linux에서 NICE DCV 서버가 자동으로 시작되도록 구성하려면 다음과 같이 하세요.

다음 명령을 사용합니다.

- RHEL, CentOS, SUSE Linux Enterprise 12, Ubuntu 18.x

```
$ sudo systemctl enable dcvserver
```

NICE DCV 서버 중지

언제든지 NICE DCV 서버를 중지할 수 있습니다. 서버를 중지하면 모든 활성 NICE DCV 세션이 종료됩니다. 서버를 다시 시작한 후에는 새로운 세션을 시작할 수 없습니다.

Windows NICE DCV server

Microsoft Management Console의 Services 스냅인을 사용하여 NICE DCV 서버를 수동으로 중지합니다.

Windows에서 NICE DCV 서버를 중지하려면 다음과 같이 하세요.

1. MMC(Microsoft Management Console)의 Services 스냅인을 엽니다.
2. 오른쪽 창에서 DCV Server(DCV 서버)를 엽니다.
3. 중지를 선택합니다.

Note

서버가 이미 중지된 경우 중지 버튼이 비활성화되어 있습니다.

Microsoft Management Console의 Services 스냅인을 사용하여 자동 시작을 비활성화합니다.

Windows에서 NICE DCV 서버가 자동으로 시작되지 않도록 하려면 다음과 같이 하세요.

1. MMC(Microsoft Management Console)의 Services 스냅인을 엽니다.
2. 오른쪽 창에서 DCV Server(DCV 서버)를 엽니다.
3. Startup service(시작 서비스)에서 수동을 선택합니다.

Linux NICE DCV server

명령줄을 사용하여 NICE DCV 서버를 중지합니다.

Linux에서 NICE DCV 서버를 중지하려면 다음과 같이 하세요.

다음 명령을 사용합니다.

- RHEL, CentOS, SUSE Linux Enterprise 12

```
$ sudo systemctl stop dcvserver
```

명령줄을 사용하여 NICE DCV 서버 자동 시작을 비활성화합니다.

Linux에서 NICE DCV 서버가 자동으로 시작되지 않도록 하려면 다음과 같이 하세요.

다음 명령을 사용합니다.

- RHEL, CentOS, SUSE Linux Enterprise 12

```
$ sudo systemctl disable dcvserver
```

QUIC UDP 전송 프로토콜 활성화

기본적으로 NICE DCV는 WebSocket TCP를 기반으로 하는 프로토콜을 데이터 전송에 사용합니다.

데이터 전송에 QUIC 프로토콜을 사용하도록 NICE DCV를 구성할 수 있습니다. 이 전송 프로토콜은 UDP를 기반으로 합니다. 네트워크 지연 시간이 길고 패킷 손실이 발생하는 경우 QUIC를 사용하면 성능이 향상될 수 있습니다. QUIC를 활성화하면 NICE DCV 서버는 데이터 전송에 QUIC 프로토콜을 사용합니다. 하지만 인증 트래픽에는 계속 WebSocket 사용됩니다.

Note

네트워크 및 보안 구성에서 UDP 트래픽을 허용하는 경우에만 QUIC를 사용할 수 있습니다.

QUIC를 활성화하면 클라이언트는 NICE DCV 서버 세션에 연결할 때 QUIC 프로토콜을 사용하여 데이터를 전송할 수 있습니다. 클라이언트가 연결할 때 QUIC 프로토콜을 사용하지 않는 경우 QUIC 프로토

콜을 사용합니다 WebSocket. QUIC 프로토콜에 대한 자세한 내용은 NICE DCV 사용 설명서의 [NICE DCV 세션에 연결](#)을 참조하세요.

Windows NICE DCV server

데이터 전송에 QUIC(UDP)를 사용하도록 NICE DCV를 구성하려면 다음과 같이 하세요.

1. Windows 레지스트리 편집기를 열고 HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/ 키로 이동합니다.
2. enable-quic-frontend 파라미터를 엽니다. 값 데이터에 1을 입력합니다.

Note

파라미터를 찾을 수 없는 경우, 새 DWORD(32비트) 파라미터를 생성하고 이름을 enable-quic-frontend로 지정합니다.

3. Windows 레지스트리 편집기를 닫습니다.
4. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

Linux NICE DCV server

데이터 전송에 QUIC(UDP)를 사용하도록 NICE DCV를 구성하려면 다음과 같이 하세요.

1. 원하는 텍스트 편집기로 /etc/dcv/dcv.conf를 엽니다.
2. [connectivity] 섹션에서 다음을 수행합니다.
 - enable-quic-frontend에서 true를 지정합니다.

```
[connectivity]
enable-quic-frontend=true
```

3. 파일을 저장하고 닫습니다.
4. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

NICE DCV 서버 TCP/UDP 포트 및 수신 주소 변경

기본적으로 NICE DCV 서버는 TCP 8443 포트에서 수신하고 해당 서버가 실행되는 호스트의 모든 네트워크 인터페이스에서 통신하도록 구성됩니다.

NICE DCV 서버를 설치한 후 사용자 지정 TCP 포트를 지정할 수 있습니다. [QUIC를 활성화](#)하도록 NICE DCV 서버를 구성한 경우 QUIC 트래픽에 대한 사용자 지정 UDP 포트를 지정할 수도 있습니다. 포트 번호는 1024보다 커야 합니다.

NICE DCV 서버가 수신하는 네트워크 주소를 지정할 수 있습니다. 예를 들어, IPv4만 사용할지 아니면 IPv6만 사용할지를 지정할 수 있습니다. 또한 서버를 특정 네트워크 인터페이스에 바인딩하고 트래픽이 특정 네트워크를 통해 흐르도록 할 수 있습니다.

Important

NICE DCV 서버의 네트워크 구성을 변경할 때마다 변경 사항을 클라이언트에 전달해야 합니다. 예를 들어, 클라이언트는 세션에 연결하는 데 어떤 포트 번호가 사용되는지 알아야 합니다.

Tip

클라이언트에 노출되는 네트워크 주소 및 포트를 관리하는 또 다른 방법은 [NICE DCV Connection Gateway](#)나 다른 웹 프록시 또는 로드 밸런서를 서버의 프론트엔드로 사용하는 것입니다. 게이트웨이를 통해 NICE DCV 서버 호스트에 액세스하면 서버에 대한 단일 주소를 가질 수 있습니다. 또한 HTTPS의 표준 포트 번호인 443을 포함하여 1024보다 낮은 포트 번호를 사용할 수 있습니다.

게이트웨이의 네트워크 주소 및 포트 구성에 대한 자세한 내용은 게이트웨이 설명서를 참조하세요.

주제

- [NICE DCV 서버 TCP/UDP 포트 변경](#)
- [특정 엔드포인트에서 수신](#)

NICE DCV 서버 TCP/UDP 포트 변경

Windows NICE DCV server

NICE DCV 서버에서 사용하는 포트를 변경하려면 Windows 레지스트리 편집기를 사용하여 web-port 및 quic-port 파라미터를 구성해야 합니다.

Windows에서 서버의 포트를 변경하려면 다음과 같이 하세요.

1. Windows 레지스트리 편집기를 엽니다.
2. HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/ 키로 이동합니다.
3. TCP 포트를 구성하려면 web-port 파라미터를 선택합니다.

레지스트리 키에 web-port 파라미터가 없는 경우 하나를 만듭니다.

- a. 탐색 창에서 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 연결 키를 선택합니다. 그런 다음 새로 만들기, DWORD(32비트) 값을 선택합니다.
- b. 이름에 web-port을 입력하고 Enter 키를 누릅니다.
4. web-port 파라미터를 엽니다. Value data(값 데이터)에 새 TCP 포트 번호를 입력합니다. 이 파라미터를 구성하지 않으면 NICE DCV 서버는 기본적으로 TCP 포트 8443을 사용합니다.

Note

TCP 포트 번호는 1024보다 커야 합니다.

5. QUIC가 활성화된 경우 UDP 포트를 구성하려면 quic-port 파라미터를 선택합니다.

레지스트리 키에 quic-port 파라미터가 없는 경우 하나를 만듭니다.

- a. 탐색 창에서 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 연결 키를 선택합니다. 그런 다음 새로 만들기, DWORD(32비트) 값을 선택합니다.
- b. 이름에 quic-port을 입력하고 Enter 키를 누릅니다.
6. quic-port 파라미터를 엽니다. 값 데이터에 새 UDP 포트 번호를 입력합니다. 이 파라미터를 구성하지 않은 상태로 QUIC 지원이 활성화되어 있으면 NICE DCV 서버는 기본적으로 UDP 포트 8443을 사용합니다.

Note

UDP 포트 번호는 1024보다 커야 합니다.

7. 확인을 선택하고 Windows 레지스트리 편집기를 닫습니다.
8. NICE DCV 서버를 **중지**하고 **다시 시작**합니다.

Linux NICE DCV server

NICE DCV 서버에서 사용하는 포트를 변경하려면 `dcv.conf` 파일에 `web-port` 및 `quic-port` 파라미터를 구성해야 합니다.

Linux에서 서버의 포트를 변경하려면 다음과 같이 하세요.

1. `/etc/dcv/`로 이동하고 원하는 텍스트 편집기에서 `dcv.conf`를 엽니다.
2. `[connectivity]` 섹션에서 `web-port` 파라미터를 찾습니다. 그런 다음 기존 TCP 포트 번호를 새 TCP 포트 번호로 바꿉니다.

`[connectivity]` 섹션에 `web-port` 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[connectivity]
web-port=port_number
```

Note

TCP 포트 번호는 1024 이상이어야 합니다.

3. `[connectivity]` 섹션에서 `quic-port` 파라미터를 찾습니다. 그런 다음 기존 UDP 포트 번호를 새 UDP 포트 번호로 바꿉니다.

`[connectivity]` 섹션에 `web-port` 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[connectivity]
quic-port=port_number
```


Note

UDP 포트 번호는 1024 이상이어야 합니다.

4. 파일을 저장하고 닫습니다.
5. NICE DCV 서버를 **중지**하고 **다시 시작**합니다.

특정 엔드포인트에서 수신

특정 네트워크 주소에서만 수신하려면 NICE DCV 서버의 구성에서 `web-listen-endpoints` 및 `quic-listen-endpoints` 파라미터를 설정할 수 있습니다.

각 엔드포인트는 IPv4 또는 IPv6 주소로 표시되며 필요에 따라 `:`으로 구분된 포트 번호가 따라옵니다. 엔드포인트에 지정된 포트 번호는 `web-port` 및 `quic-port` 파라미터에 지정된 포트보다 우선합니다.

두 개 이상의 엔드포인트를 지정할 수 있으므로 엔드포인트 집합은 쉼표로 구분되어 대괄호로 묶인 목록으로 표시되며, 각 엔드포인트는 작은따옴표로 묶입니다. 예를 들어, `['0.0.0.0:8443', ':::8443']`은 포트 8443의 로컬 IPv4 주소 및 IPv6 주소를 나타내며, `['::%1]:8443'`는 인덱스가 1인 Windows 호스트의 네트워크 인터페이스에 바인딩된 IPv6 주소를 나타냅니다. `[':: %eth1]:8443'`은 Linux 호스트의 `eth1` 네트워크 인터페이스에 바인딩된 IPv6 주소를 나타냅니다.

Note

이러한 구성 파라미터는 NICE DCV 서버 2022.0부터 사용할 수 있습니다.

Windows NICE DCV server

Windows에서 서버의 엔드포인트를 변경하려면 다음과 같이 하세요.

1. Windows 레지스트리 편집기를 엽니다.
2. `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/` 키로 이동합니다.
3. TCP 엔드포인트를 구성하려면 매개변수를 선택합니다. `web-listen-endpoints`

레지스트리 키에 `web-listen-endpoints` 파라미터가 없는 경우 하나를 만듭니다.

- a. 탐색 창에서 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 연결 키를 선택합니다. 그런 다음 새로 만들기, 문자열 값을 선택합니다.
- b. 이름에 web-listen-endpoints을 입력하고 Enter 키를 누릅니다.
4. 파라미터를 엽니다. web-listen-endpoints 값 데이터에는 엔드포인트 목록을 입력합니다.
5. QUIC가 활성화된 경우 UDP 엔드포인트를 구성하려면 파라미터를 선택합니다. quic-listen-endpoints

레지스트리 키에 quic-listen-endpoints 파라미터가 없는 경우 하나를 만듭니다.

- a. 탐색 창에서 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 연결 키를 선택합니다. 그런 다음 새로 만들기, 문자열 값을 선택합니다.
- b. 이름에 quic-listen-endpoints을 입력하고 Enter 키를 누릅니다.
6. 파라미터를 엽니다. quic-listen-endpoints 값 데이터에는 엔드포인트 목록을 입력합니다.
7. 확인을 선택하고 Windows 레지스트리 편집기를 닫습니다.
8. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

Linux NICE DCV server

Linux에서 서버의 엔드포인트를 변경하려면 다음과 같이 하세요.

1. /etc/dcv/로 이동하고 원하는 텍스트 편집기에서 dcv.conf를 엽니다.
2. [connectivity] 섹션에서 web-listen-endpoints 파라미터를 찾습니다. 그런 다음 기존 엔드포인트 목록을 교체합니다.

[connectivity] 섹션에 web-listen-endpoints 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[connectivity]
web-listen-endpoints=[endpoint1, endpoint2]
```

3. [connectivity] 섹션에서 quic-listen-endpoints 파라미터를 찾습니다. 그런 다음 기존 엔드포인트 목록을 교체합니다.

[connectivity] 섹션에 quic-listen-endpoints 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[connectivity]
```

```
quic-listen-endpoints=[endpoint1, endpoint2]
```

4. 파일을 저장하고 닫습니다.
5. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

TLS 인증서 관리

NICE DCV는 NICE DCV 클라이언트와 NICE DCV 서버 간의 트래픽을 보호하는 데 사용되는 자체 서명된 인증서를 자동으로 생성합니다. 설치된 다른 인증서가 없는 경우, 기본적으로 이 인증서가 사용 됩니다. 기본 인증서에는 두 개의 파일이 포함되어 있습니다. 두 파일 자체가 인증서(dcv.pem) 및 키(dcv.key)입니다. 자세한 내용은 [the section called “자체 서명된 인증서에 대한 리디렉션 설명”](#) 단원을 참조하십시오.

DCV 클라이언트 사용자가 서버에 연결하면, 연결이 설정되기 전에 확인을 위해 조치를 취할 수 있는 서버 인증서 경고를 받을 수 있습니다.

웹 브라우저를 사용하여 연결하는 경우, 브라우저는 클라이언트 사용자에게 서버 인증서를 신뢰하는 것에 대해 경고하고, 관리자에게 문의하여 인증서의 신뢰성을 확인해야 한다고 경고할 수 있습니다.

마찬가지로 사용자가 Windows, Linux 또는 macOS 클라이언트를 사용하는 경우 NICE DCV 서버 관리자에게 해당 인증서의 지문을 확인하는 것이 좋습니다.

인증서 지문의 신뢰성을 확인하려면 `dcv list-endpoints -j`를 실행하여 인증서 지문에 대한 출력을 확인합니다.

기본 NICE DCV 인증서와 해당 키를 자체 인증서와 키로 바꿀 수 있습니다.

자체 인증서를 생성할 때는 특정 요구 사항에 맞는 인증서 속성을 선택하세요. 대부분의 경우 CN (Common Name) 속성은 호스트의 퍼블릭 호스트 이름과 일치해야 합니다. SAN (Subject Alternative Name) 속성을 지정하고 해당 속성에 호스트의 IP 주소를 설정할 수도 있습니다.

인증서 생성 방법에 대한 안내는 해당 인증 기관의 설명서를 참조하세요.

Important

자체 인증서와 키를 사용하는 경우 인증서 이름을 `dcv.pem`으로 지정하고 키 이름을 `dcv.key`로 지정해야 합니다.

Windows NICE DCV server

Windows에서 서버의 TLS 인증서를 변경하려면 다음과 같이 하세요.

- Windows NICE DCV 서버의 다음 위치에 인증서와 키를 둡니다.

```
C:\Windows\System32\config\systemprofile\AppData\Local\NICE\dcv\
```

Linux NICE DCV server

Linux에서 서버의 TLS 인증서를 변경하려면 다음과 같이 하세요.

- Linux NICE DCV 서버의 다음 위치에 인증서와 키를 둡니다.

```
/etc/dcv/
```

- 두 파일 모두의 소유권을 dcv 사용자에게 부여하고 해당 권한을 600(소유자만 파일을 읽거나 쓸 수 있음)으로 변경합니다.

```
$ sudo chown dcv dcv.pem dcv.key
```

```
$ sudo chmod 600 dcv.pem dcv.key
```

Note

NICE DCV 2022.0부터, NICE DCV 서버가 실행되는 동안 인증서 파일을 업데이트하면 새 인증서가 자동으로 다시 로드됩니다. 이전 버전의 NICE DCV의 경우 NICE DCV 서버를 수동으로 [중지](#)하고 [다시 시작](#)해야 합니다.

유휴 클라이언트 연결 해제

유휴 클라이언트의 연결을 해제하도록 NICE DCV를 구성할 수 있습니다. 더 구체적으로 설명하자면, 특정 기간 동안 NICE DCV 서버로 어떠한 키보드 또는 포인터 입력도 보내지 않은 클라이언트에 대해 이 작업을 수행할 수 있습니다. 기본적으로 NICE DCV 서버는 60분(1시간) 동안 유휴 상태가 된 후 NICE DCV 클라이언트의 연결을 해제합니다.

유휴 연결 해제 제한 시간을 재설정하는 특정 작업이 있습니다. 다음 작업 중 하나가 발생하면 유휴 제한 시간이 설정된 시간 프레임으로 재설정됩니다.

- 마우스 움직이기
- 마우스 버튼 누르기 또는 마우스 휠 움직이기
- 키보드의 아무 키나 누르기
- 터치스크린 터치 (활성화된 경우)
- 스타일러스 사용 (활성화된 경우)
- 게임패드 사용 (활성화된 경우)
- 웹캠으로 스트리밍 (활성화된 경우)
- 파일 업로드, 디렉터리 생성, 파일 다운로드 또는 항목 목록과 같은 모든 파일 저장 작업

Note

오디오 장치를 연결하고 사용해도 유휴 제한 시간이 재설정되지 않습니다.

유휴 클라이언트에 알림을 보내도록 NICE DCV 서버를 구성할 수도 있습니다. 알림은 세션 연결이 끊길 예정임을 알리기 위한 것입니다. 제한 시간 알림은 NICE DCV 서버 및 클라이언트 버전 2017.4 이상에서만 지원됩니다.

다음 절차에 따라 사용자 지정 유휴 제한 시간을 지정할 수 있습니다.

Windows NICE DCV server

NICE DCV 서버의 유휴 제한 시간을 변경하려면 Windows 레지스트리 편집기를 사용하여 `idle-timeout` 파라미터를 구성해야 합니다.

Windows에서 유휴 제한 시간을 변경하려면

1. Windows 레지스트리 편집기를 엽니다.
2. `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/` 키로 이동하여 `idle-timeout` 파라미터를 선택합니다.

파라미터를 찾을 수 없는 경우, 다음 단계에 따라 파라미터를 생성합니다.

- a. 탐색 창에서 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 연결 키를 선택합니다. 그런 다음 새로 만들기, `DWORD(32비트)` 값을 선택합니다.

- b. 이름에 `idle-timeout`을 입력하고 Enter 키를 누릅니다.
3. `idle-timeout` 파라미터를 엽니다. 값 데이터에 유휴 제한 시간 값(분 단위, 10진수)을 입력합니다. 유휴 클라이언트의 연결이 해제되지 않도록 하려면 0을 입력합니다.
4. 확인을 선택하고 Windows 레지스트리 편집기를 닫습니다.

(선택 사항) 유휴 클라이언트에 제한 시간 알림을 보내도록 NICE DCV 서버를 구성하려면 다음과 같이 하세요.

1. HKEY_USERS/S-1-5-18/Software/GSettings/com/NiceSoftware/DCV/Connectivity/ 키로 이동하여 매개변수를 선택합니다. `idle-timeout-warning`

파라미터를 찾을 수 없는 경우, 다음 단계에 따라 파라미터를 생성합니다.

- a. 탐색 창에서 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 연결 키를 선택합니다. 그런 다음 새로 만들기, DWORD(32비트) 값을 선택합니다.
- b. 이름에 `idle-timeout-warning`을 입력하고 Enter 키를 누릅니다.
2. `idle-timeout-warning`파라미터를 엽니다. 값 데이터의 경우 연결이 해제되기 전에 관련 경고 알림이 전송되는 시간(초 단위, 10진수)을 입력합니다. 예를 들어 유휴 제한 시간에 도달하기 2분 전에 알림을 보내려면 120을 입력합니다.
3. 확인을 선택하고 Windows 레지스트리 편집기를 닫습니다.

Linux NICE DCV server

NICE DCV 서버의 유휴 제한 시간을 변경하려면 `dcv.conf` 파일에서 `idle-timeout` 파라미터를 구성해야 합니다.

Linux에서 유휴 제한 시간을 변경하려면

1. 원하는 텍스트 편집기로 `/etc/dcv/dcv.conf`를 엽니다.
2. `[connectivity]` 섹션에서 `idle-timeout` 파라미터를 찾습니다. 그런 다음 기존 제한 시간을 새 제한 시간(분 단위, 10진수)으로 바꿉니다.

`[connectivity]` 섹션에 `idle-timeout` 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[connectivity]
idle-timeout=timeout_in_minutes
```

유휴 클라이언트의 연결이 해제되지 않도록 하려면 0을 입력합니다.

3. 파일을 저장하고 닫습니다.

(선택 사항) 유휴 클라이언트에 제한 시간 알림을 보내도록 NICE DCV 서버를 구성하려면 다음과 같이 하세요.

1. 원하는 텍스트 편집기로 `/etc/dcv/dcv.conf`를 엽니다.
2. `[connectivity]` 섹션에 `idle-timeout-warning` 파라미터를 추가하고 연결이 해제되기 까지 걸리는 시간(초 단위, 10진수)을 지정하여 관련 경고 알림을 보냅니다.

```
idle-timeout-warning=seconds_before_idle_timeout
```

예를 들어 유휴 제한 시간에 도달하기 2분 전에 알림을 보내려면 120을 지정합니다.

3. 파일을 저장하고 닫습니다.

Linux NICE DCV 서버에서 GPU 공유 활성화

GPU 공유를 사용하면 여러 NICE DCV 가상 세션 간에 하나 이상의 물리적 GPU를 공유할 수 있습니다. 세션에 대한 자세한 내용은 [NICE DCV 세션 관리](#) 단원을 참조하세요. GPU 공유를 사용하면 단일 NICE DCV 서버를 사용하여 이 서버의 물리적 GPU 리소스를 공유하는 여러 가상 세션을 호스팅할 수 있습니다.

Note

GPU 공유는 Linux NICE DCV 서버에서만 지원됩니다.

필수 조건

시작하기 전에 다음 필수 조건을 완료합니다.

- Linux 서버에 NICE DCV 서버를 설치합니다.
- 서버에 NICE DCV `dcv-g1` 및 `nice-xdcv` 패키지를 설치합니다.
- 서버에 지원되는 NVIDIA GPU가 하나 이상 있는지 확인합니다.
- 서버에 NVIDIA GPU 드라이버를 설치합니다. 공식 NVIDIA 드라이버가 필요합니다. 오픈 소스 NVIDIA 드라이버는 지원되지 않습니다.

- NVIDIA GPU 드라이버가 하드웨어 가속화된 OpenGL을 지원하는지 확인합니다.
- X 서버를 설치하고, `xorg.conf` 파일에서 Device 및 Screen 섹션을 구성합니다.

Note

`nvidia-xconfig` NVIDIA 유틸리티를 사용하여 `xorg.conf` 파일을 자동으로 생성하고 사용 가능한 모든 NVIDIA GPU에 대해 이를 구성할 수 있습니다.

- X 서버가 실행되는지 확인합니다.
- (선택 사항) `dcvgldiag` 도구를 실행하여 NICE DCV 서버 구성을 확인합니다. 자세한 정보는 [설치 후 점검](#)을 참조하세요.

`nice-dcv-gltest` 패키지를 설치하고 `dcvgltest` 테스트 애플리케이션을 실행하여 GPU 공유에 대해 서버가 올바르게 구성되었는지 확인할 수 있습니다.

GPU 공유를 활성화하려면 가상 세션에서 사용할 GPU 목록을 지정해야 합니다. GPU를 지정하지 않으면 표시 이름 `:0.0`과 함께 표준 X 서버에서 사용하는 GPU만 사용됩니다.

이 주제의 앞부분에서 설명한 사전 요구 사항을 완료한 후 `dcv.conf` 파일의 `gl-displays` 파라미터에 GPU를 지정합니다.

Linux NICE DCV 서버에서 GPU 공유를 활성화하려면

1. `/etc/dcv/`로 이동하고 원하는 텍스트 편집기에서 `dcv.conf` 파일을 엽니다.
2. `[display/linux]` 섹션과 `gl-displays` 파라미터를 추가합니다. 그런 다음, 사용 가능한 GPU를 다음 형식으로 지정합니다.

```
[display/linux]
gl-displays =
  [':xserver_port.screen_number_1', ':xserver_port.screen_number_2', ...]
```

여기서 `xserver_port`는 서버이고 `screen_number`는 GPU 관련 화면과 연관된 번호입니다. `screen_number`는 0으로 시작합니다.

다음 예제에서는 기본 X 서버 세션에서 실행되는 두 GPU의 `gl-displays` 파라미터를 보여 줍니다.

```
[display/linux]
```



```
gl-displays = [':0.0', ':0.1']
```

3. 파일을 저장하고 닫습니다.
4. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

터치스크린 및 스타일러스 지원 활성화

Note

터치스크린 및 스타일러스 디바이스의 USB 리디렉션은 필요하지 않습니다. 또한 NICE DCV 서버에 공급업체 드라이버를 설치할 필요가 없습니다.

NICE DCV는 기본 운영 체제 API를 사용하여 터치스크린과 스타일러스를 지원합니다.

Windows는 Windows Ink를 사용합니다.

Linux는 X11 입력 주입을 사용합니다.

• Windows 서버 지원

터치스크린은 지원되는 모든 Windows 운영 체제에서 지원됩니다. 스타일러스는 Windows 10 및 Windows 2019부터 지원되는 모든 Windows 운영 체제에서 지원되며, Windows 2016, Windows 8.1 및 이전 버전에서는 지원되지 않습니다. 이 기능은 Windows NICE DCV 서버에서 기본적으로 활성화됩니다. 추가 구성은 필요하지 않습니다.

• Linux 서버 지원

터치스크린과 스타일러스는 지원되는 모든 Linux 운영 체제에서 지원됩니다. 이 기능은 Linux NICE DCV 서버에서 호스팅되는 가상 세션에서 기본적으로 활성화됩니다. 그러나 Linux NICE DCV 서버에서 호스팅되는 콘솔 세션에서 기능을 활성화하려면 몇 가지 추가 구성이 필요합니다.

Important

NICE DCV를 통한 터치스크린 및 스타일러스 사용은 해당 기능이 클라이언트와 서버에서 모두 지원되고, 기능이 서버에서 사용 설정된 경우 활성화됩니다. 클라이언트 지원에 대한 자세한 내용은 NICE DCV 사용 설명서의 [클라이언트 기능](#)을 참조하세요.

Linux NICE DCV 서버에서 호스팅되는 콘솔 세션에 대한 터치스크린 및 스타일러스 지원을 활성화하려면 다음과 같이 하세요.

1. 원하는 텍스트 편집기에서 `/etc/X11/xorg.conf` 파일을 엽니다.
2. 파일에 다음 섹션을 추가합니다.

```
Section "InputDevice"
    Identifier "DCV Stylus Pen"
    Driver "dcvinput"
EndSection

Section "InputDevice"
    Identifier "DCV Stylus Eraser"
    Driver "dcvinput"
EndSection

Section "InputDevice"
    Identifier "DCV Touchscreen"
    Driver "dcvinput"
EndSection
```

3. `ServerLayout` 섹션 끝에 다음을 추가합니다.

```
InputDevice "DCV Stylus Pen"
InputDevice "DCV Stylus Eraser"
InputDevice "DCV Touchscreen"
```

예:

```
Section "ServerLayout"
    ...existing content...
    InputDevice "DCV Stylus Pen"
    InputDevice "DCV Stylus Eraser"
    InputDevice "DCV Touchscreen"
EndSection
```

4. 변경 내용을 저장하고 파일을 닫습니다.
5. X 서버를 시작합니다.
 - RHEL, Rocky, CentOS, 아마존 리눅스 2, 우분투, SUSE 리눅스 엔터프라이즈 12.x

```
$ sudo systemctl isolate multi-user.target
```

```
$ sudo systemctl isolate graphical.target
```

6. 입력 장치가 올바르게 구성되었는지 확인하려면 다음 명령을 실행합니다.

```
$ sudo DISPLAY=:0 xinput
```

DCV 스타일러스 펜, DCV 스타일러스 지우개, DCV 터치스크린이 명령 출력에 나타납니다. 출력의 예제는 다음과 같습니다.

```
| Virtual core pointer                id=2    [master pointer (3)]
|   | Virtual core XTEST pointer      id=4    [slave pointer (2)]
|   | dummy_mouse                     id=6    [slave pointer (2)]
|   | dummy_keyboard                  id=7    [slave pointer (2)]
|   | DCV Stylus Pen                   id=8    [slave pointer (2)]
|   | DCV Stylus Eraser                 id=9    [slave pointer (2)]
|   | DCV Touchscreen                  id=10   [slave pointer (2)]
| Virtual core keyboard               id=3    [master keyboard (2)]
|   | Virtual core XTEST keyboard     id=5    [slave keyboard (3)]
```

스타일러스 압력 범위 구성

일부 애플리케이션에서는 스타일러스 압력 범위를 0에서 2048 사이로 줄여야 합니다. `/etc/X11/xorg.conf` 파일에서 `Pressure2k` 옵션을 `true`로 설정하여 압력 범위를 구성할 수 있습니다.

스타일러스 압력을 구성하려면 다음과 같이 하세요.

1. 원하는 텍스트 편집기에서 `/etc/X11/xorg.conf` 파일을 엽니다.
2. 파일에 다음 섹션을 추가합니다.

```
Section "InputDevice"
    Identifier "DCV Stylus Pen"
    Driver "dcvinput"
    Option "Pressure2K" "true"
EndSection

Section "InputDevice"
```

```
Identifier "DCV Stylus Eraser"
Driver "dcvinput"
Option "Pressure2K" "true"
EndSection
```

3. 변경 내용을 저장하고 파일을 닫습니다.
4. X 서버를 시작합니다.

게임패드 지원 활성화

NICE DCV 서버 2022.0부터는 지원되는 모든 Windows 또는 Linux 운영 체제에 연결할 때 게임패드 디바이스를 사용할 수 있습니다.

지원되는 게임패드 디바이스는 다음과 같습니다.

- Xbox 360 컨트롤러
- DualShock 4 컨트롤러

위에 나열된 디바이스와 호환되거나, 지원되는 디바이스 중 하나를 에뮬레이션하도록 구성할 수 있는 다른 디바이스도 사용할 수 있습니다.

Note

게임패드 디바이스는 Windows 네이티브 NICE DCV 클라이언트를 사용하는 경우에만 지원됩니다. NICE DCV 클라이언트 2022.0 이상을 사용 중이어야 합니다.

게임패드 지원을 활성화하려면 최신 버전의 NICE DCV Server를 설치하고 Gamepad 드라이버를 설치하도록 선택했는지 확인하세요. 자세한 정보는 [Windows에 NICE DCV 서버 설치](#)를 참조하세요. 드라이버가 설치되면 Windows NICE DCV 서버에서 이 기능이 기본적으로 활성화됩니다.

Xbox 360 컨트롤러 지원

Xbox 360 컨트롤러를 사용하려면 Windows 드라이버를 설치해야 합니다. 이 드라이버는 Windows에 자동으로 설치되지 않으므로 공식 Windows 업데이트 웹사이트에서 검색해야 합니다.

최신 Xbox 360 컨트롤러 드라이버를 다운로드하고 설치하려면 다음과 같이 하세요.

1. Microsoft 업데이트 카탈로그 페이지(<https://www.catalog.update.microsoft.com/Search.aspx?q=game+devices+XBOX+360+Controller+For+Windows>)에서 드라이버를 검색합니다.
2. 사용 중인 운영 체제에 대한 최신 버전의 드라이버를 다운로드합니다.
3. .cab 파일을 열고 내용을 추출합니다.

```
expand filename.cab -F:* .
```

4. 다음 명령어를 사용해 드라이버의 .inf 파일을 설치합니다.

```
pnputil /add-driver filename.inf /install
```

USB 원격화 활성화

NICE DCV를 통해 클라이언트는 3D 포인팅 디바이스 또는 인증 디바이스와 같은 다양한 특수 USB 디바이스를 사용할 수 있습니다. 디바이스는 NICE DCV 서버에서 실행 중인 애플리케이션과 상호 작용하도록 컴퓨터에 물리적으로 연결되어 있습니다.

Important

NICE DCV는 USB 디바이스를 리디렉션하기 위한 일반적인 메커니즘을 제공합니다. 네트워크 지연에 민감한 일부 디바이스에서는 문제가 발생할 수 있습니다. 또한 드라이버 호환성 문제로 인해 일부 디바이스가 예상대로 작동하지 않을 수 있습니다. 프로덕션에 배포하기 전에 디바이스가 예상대로 작동하는지 확인하세요.

Note

USB 원격화는 Windows 클라이언트에서만 지원됩니다. 휴대용 Windows 클라이언트 또는 웹 브라우저 클라이언트에서는 지원되지 않습니다. NICE DCV 클라이언트에 추가적인 구성이 필요할 수 있습니다. 클라이언트에 USB 원격화를 설치하는 방법에 대한 자세한 내용은 NICE DCV 사용 설명서 의 [설치 가능한 Windows 클라이언트](#)의 선택 단계를 참조하세요.

NICE DCV 서버는 허용 목록을 사용하여 어떤 USB 디바이스 클라이언트를 사용하도록 허용할지 결정합니다. 일반적으로 사용되는 일부 USB 디바이스가 기본적으로 허용 목록에 추가됩니다. 따라서 클라

이언트가 이러한 USB 디바이스를 컴퓨터에 연결하고 추가 구성 없이 서버에서 사용할 수 있습니다. 자세한 내용은 NICE DCV 사용 설명서의 [USB 원격화 사용](#)을 참조하세요.

그러나 일부 특수화된 디바이스는 기본적으로 허용 목록에 추가되지 않을 수 있습니다. 이러한 디바이스는 클라이언트에서 사용하기 전에 NICE DCV 서버의 USB 허용 목록에 수동으로 추가해야 합니다. 추가된 후에는 Windows 클라이언트의 설정 메뉴에 나타납니다.

Windows NICE DCV server

USB 디바이스를 허용 목록에 추가하려면 클라이언트에서 USB 디바이스의 필터 문자열을 가져와 `usb-devices.conf` 파일에 추가해야 합니다.

Windows NICE DCV 서버의 허용 목록에 USB 디바이스를 추가하려면 다음과 같이 하세요.

1. NICE DCV 서버의 최신 버전을 설치했고 USB 원격화 드라이버를 설치하기로 선택했는지 확인합니다. 자세한 정보는 [Windows에 NICE DCV 서버 설치](#)을 참조하세요.
2. USB 디바이스의 하드웨어 드라이버를 NICE DCV 서버에 설치합니다.
3. Windows 클라이언트 머신에서 파일 관리자의 `C:\Program Files (x86)\NICE\DCV\Client\bin\` 경로로 이동합니다.
4. `dcvusblist.exe`를 실행합니다.
5. 목록에서 USB 디바이스를 마우스 오른쪽 버튼으로 클릭합니다.
6. 드롭다운 메뉴에서 필터 문자열 복사를 선택합니다.
7. 서버에서 원하는 텍스트 편집기를 사용하여 `C:\Program Files\NICE\DCV\Server\conf\usb-devices.conf`을 열고 필터 문자열을 파일 마지막의 새 라인에 추가합니다.
8. 파일을 저장하고 닫습니다.
9. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

Linux NICE DCV server

USB 디바이스를 허용 목록에 추가하려면 USB 디바이스의 필터 문자열을 `usb-devices.conf` 파일에 추가해야 합니다.

Linux NICE DCV 서버의 허용 목록에 USB 디바이스 추가

1. NICE DCV 서버의 최신 버전 및 DCV USB 드라이버를 설치했는지 확인합니다. 자세한 정보는 [Linux에서 NICE DCV 서버 설치](#)을 참조하세요.
2. USB 디바이스의 하드웨어 드라이버를 NICE DCV 서버에 설치합니다.

3. Windows 클라이언트 머신에서 파일 관리자의 C:\Program Files (x86)\NICE\DCV\Client\bin\ 경로로 이동합니다.
4. dcvusblist.exe를 실행합니다.
5. 목록에서 USB 디바이스를 마우스 오른쪽 버튼으로 클릭합니다.
6. 드롭다운 메뉴에서 필터 문자열 복사를 선택합니다.
7. 서버에서 원하는 텍스트 편집기를 사용하여 /etc/dcv/usb-devices.conf을 열고 필터 문자열을 파일 마지막의 새 라인에 추가합니다.
8. 파일을 저장하고 닫습니다.
9. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

스마트 카드 캐싱 구성

스마트 카드 캐싱 기능을 통해 NICE DCV 서버가 스마트 카드 값을 캐싱할 수 있습니다. 이 기능을 활성화하면 NICE DCV 서버가 최근 호출의 결과를 클라이언트의 스마트 카드에 캐싱합니다. 향후 호출은 클라이언트가 아니라 서버의 캐시에서 직접 검색됩니다. 이렇게 하면 클라이언트와 서버 간에 전송되는 트래픽 양을 감소시켜 성능을 개선할 수 있습니다. 이는 클라이언트의 인터넷 연결이 느린 경우에 특히 유용합니다.

스마트 카드 캐싱은 기본적으로 비활성화되어 있습니다. 클라이언트는 DCV_PCSC_ENABLE_CACHE 환경 변수를 설정하여, 실행 중인 각 애플리케이션에 대한 스마트 카드 캐싱을 수동으로 활성화할 수 있습니다. 자세한 안내는 NICE DCV 사용 설명서의 [스마트 카드 사용](#)을 참조하세요. 또는 DCV_PCSC_ENABLE_CACHE 환경 변수에 대해 지정한 값과 상관없이 스마트 카드 캐싱을 영구적으로 활성화하거나 비활성화하도록 NICE DCV 서버를 구성할 수 있습니다.

Linux NICE DCV server

Linux NICE DCV 서버에서 스마트 카드 캐싱을 영구적으로 활성화 또는 비활성화하려면 다음과 같이 하세요.

1. /etc/dcv/로 이동하고 원하는 텍스트 편집기에서 dcv.conf를 엽니다.
2. [smartcard] 섹션에서 enable-cache 파라미터를 찾습니다. 스마트 카드 캐싱을 영구적으로 활성화하려면 'always-on'을 입력합니다. 스마트 카드 캐싱을 영구적으로 비활성화하려면 'always-off'를 입력합니다.

[smartcard] 섹션에 enable-cache 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[smartcard]
enable-cache='always-on' | 'always-off'
```

3. 파일을 저장하고 닫습니다.
4. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

Windows NICE DCV server

Windows NICE DCV 서버에서 스마트 카드 캐싱을 영구적으로 활성화 또는 비활성화하려면 다음과 같이 하세요.

1. Windows 레지스트리 편집기를 엽니다.
2. HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/smartcard/ 키로 이동하여 enable-cache 파라미터를 선택합니다.

파라미터가 없으면 다음 단계에 따라 파라미터를 생성합니다.

- a. 왼쪽 창에서 스마트 카드 키의 컨텍스트(오른쪽 클릭) 메뉴를 열고 새로 만들기, 문자열 값을 선택합니다.
- b. 이름에 enable-cache을 입력하고 Enter 키를 누릅니다.
3. enable-cache 파라미터를 엽니다. 값 데이터의 경우 스마트 카드 캐싱을 영구적으로 활성화하려면 always-on을 입력하고, 스마트 카드 캐싱을 영구적으로 비활성화하려면 always-off를 입력합니다.
4. 확인을 선택하고 Windows 레지스트리 편집기를 닫습니다.

WebAuthn 리디렉션 구성

NICE DCV Server 2023.1부터 사용자는 원격 세션 내에서 지원되는 브라우저에서 웹 인증 WebAuthn () 표준을 사용하는 웹 애플리케이션을 인증할 수 있습니다. 이렇게 하려면 인증 프롬프트를 로컬로 연결된 FIDO2 인증자 (예: Windows Hello 또는 다른 FIDO2 호환 인증자) 로 리디렉션합니다. YubiKey

WebAuthn 리디렉션은 USB 리디렉션과 독립적으로 작동합니다. NICE DCV 서버에는 공급업체별 드라이버를 설치할 필요가 없습니다. WebAuthn 요청의 리디렉션은 브라우저의 기본 API를 통해 쉽게 이루어집니다.

사용하기 WebAuthn 전에 [지원되는 기능](#) 표를 다시 확인하여 모든 요구 사항을 충족하는지 확인하십시오.

지원되는 브라우저:

- Google Chrome 116 이상
- Microsoft Edge 116 이상

WebAuthn webauthn-redirection 권한을 사용하여 리디렉션을 활성화하거나 비활성화할 수 있습니다. 자세한 내용은 [권한 파일 작업을](#) 참조하십시오.

WebAuthn 리디렉션을 수행하려면 원격 서버에 브라우저 확장 프로그램을 설치해야 합니다. 이 기능을 사용하도록 설정하고 브라우저 확장을 설치하면 세션 내 브라우저에서 실행되는 웹 응용 프로그램에서 시작된 모든 WebAuthn 요청이 로컬 클라이언트로 원활하게 리디렉션됩니다. 그러면 사용자는 Windows Hello와 같은 장치를 사용하거나 인증을 YubiKey 완료하는 데 사용할 수 있습니다.

Note

이 기능을 사용하면 원격 세션 중에 브라우저 WebAuthn 내에서 사용할 수 있지만 인증자를 사용한 DCV 세션 인증은 지원하지 않습니다. WebAuthn

리디렉션 브라우저 확장 설정 WebAuthn

최초 브라우저 실행 시 자동 프롬프트

리디렉션이 활성화된 상태로 NICE DCV Server 2023.1을 WebAuthn 설치한 후 브라우저를 처음 시작할 때 브라우저 확장을 활성화하라는 메시지가 표시됩니다. 확장 프로그램을 설치하지 않거나 나중에 제거하도록 선택하면 리디렉션이 작동하지 않습니다. WebAuthn 관리자는 그룹 정책을 사용하여 설치를 강제할 수 있습니다.

그룹 정책을 사용하여 설치

확장 프로그램을 더 광범위하게 배포하려는 조직의 경우 그룹 정책을 활용할 수 있습니다.

Microsoft Edge를 사용하는 경우

1. [Microsoft Edge 관리 템플릿](#)을 다운로드하고 설치합니다.
2. 그룹 정책 관리 도구(gpmmc.msc)를 실행합니다.
3. 포리스트 > 도메인 > 내 FQDN(예: example.com) > 그룹 정책 객체로 이동합니다.
4. 원하는 정책을 선택하거나 새 정책을 만든 다음 마우스 오른쪽 버튼으로 클릭하여 "편집"을 선택합니다.

5. 컴퓨터 구성 > 관리 템플릿 > Microsoft Edge > 확장 프로그램 경로로 이동합니다.
6. "확장 프로그램 관리 설정"에서 "활성화"로 설정합니다.
7. 확장 프로그램 관리 설정 구성 필드에 다음을 입력합니다.

```

{"ihejeaahjpbegmaaeigiikmlphghlfmeh":
{"installation_mode":"force_installed","update_url":"https://edge.microsoft.com/
extensionwebstorebase/v1/crx"}}

```

8. 변경 내용을 저장하고 서버를 재부팅합니다.

Google Chrome을 사용하는 경우

1. [Google Chrome 관리 템플릿](#)을 확보 및 구현합니다.
2. Microsoft Edge의 단계와 마찬가지로 그룹 정책 관리 도구로 이동합니다.
3. 컴퓨터 구성 > 관리 템플릿 > Google Chrome > 확장 프로그램으로 이동합니다.
4. "확장 프로그램 관리 설정"에서 "활성화"로 설정합니다.
5. 확장 프로그램 관리 설정 구성 필드에 다음을 입력합니다.

```

{"mmiioagbgnbojdbcjoddlfahmccocfpmn":
{ "installation_mode":"force_installed","update_url":"https://clients2.google.com/
service/update2/crx"}}

```

6. 변경 내용을 저장하고 서버를 재부팅합니다.

수동으로 설치

확장 프로그램은 각 브라우저 스토어에서 구할 수 있습니다.

- [Microsoft Edge Add-ons](#)
- [Chrome 웹 스토어](#)

수동으로 설치하려면 다음과 같이 하세요.

1. NICE DCV 세션에 연결하세요.
2. 원하는 브라우저를 열고 관련 브라우저 스토어 (위 링크) 로 이동합니다.
3. "받기"(Microsoft Edge) 또는 "Chrome에 추가"(Google Chrome)를 선택하여 진행합니다.

4. 화면에 표시되는 지시 사항을 따릅니다. 확장 프로그램이 성공적으로 추가되면 확인 메시지가 표시됩니다.

시크릿 모드에서 WebAuthn 리디렉션 사용 (Chrome만 해당)

시크릿 모드를 사용하는 경우 Amazon DCV WebAuthn 리디렉션 확장 프로그램이 해당 확장 프로그램 내에서 실행되도록 특별히 허용해야 합니다. 그렇지 않으면 WebAuthn 리디렉션이 발생하지 않습니다. 방법:

1. 확장 프로그램 설정을 엽니다.
2. 세부 정보에서 시크릿 모드 허용을 찾아보세요.
3. 스위치를 켜기로 전환합니다.

세션 스토리지 활성화

세션 스토리지는 클라이언트가 특정 NICE DCV 세션에 연결되어 있을 때 액세스할 수 있는 NICE DCV 서버의 폴더입니다. 세션에 대해 세션 스토리지를 활성화하면 클라이언트가 파일을 다운로드하고, 파일을 지정된 폴더에 업로드할 수 있습니다. 이 기능을 통해 클라이언트는 세션에 연결된 동안 파일을 공유할 수 있습니다.

주제

- [Windows NICE DCV 서버에서 세션 스토리지 활성화](#)
- [Linux NICE DCV 서버에서 세션 스토리지 활성화](#)

Windows NICE DCV 서버에서 세션 스토리지 활성화

세션 스토리지를 활성화하려면 먼저 세션 스토리지에 사용할 폴더를 만듭니다. 그런 다음 Windows 레지스트리 편집기를 사용하여 storage-root 파라미터를 구성합니다.

Windows에서 세션 스토리지를 활성화하려면

1. 세션 스토리지에 사용할 폴더를 만듭니다(예: c:\session-storage).
2. storage-root 파라미터를 구성합니다.
 - a. Windows 레지스트리 편집기를 엽니다.

- b. automatic-console-session\HKEY_USERS\S-1-5-18\소프트웨어\GSettings\com\Nicesoftware\DCV\Session-Management/ 키로 이동하여 스토리지 루트 매개 변수를 선택합니다.

레지스트리 키에 storage-root 파라미터가 없는 경우 다음과 같이 하나를 만듭니다.

- i. 탐색 창에서 세션 관리/ 키의 컨텍스트 (마우스 오른쪽 버튼 클릭) 메뉴를 엽니다. automatic-console-session 그런 다음 새로 만들기, 문자열을 선택합니다.
 - ii. 이름에 storage-root을 입력하고 Enter 키를 누릅니다.
- c. storage-root 파라미터를 엽니다. 값 데이터에 1단계에서 만든 폴더의 전체 경로를 입력합니다.

경로에 %home%을 사용하여 현재 로그인한 사용자의 홈 디렉터리를 지정할 수도 있습니다. 예를 들어, 다음 경로는 세션 스토리지 디렉터리로 c:\Users*username*\storage\를 사용합니다.

```
%home%/storage/
```

Note

지정된 하위 디렉터리가 없으면 세션 스토리지가 비활성화됩니다.

- d. 확인을 선택하고 Windows 레지스트리 편집기를 닫습니다.
 - e. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.
3. 세션을 시작하고 --storage-root 옵션을 지정합니다. 자세한 정보는 [NICE DCV 세션 시작](#)을 참조하세요.

Linux NICE DCV 서버에서 세션 스토리지 활성화

세션 스토리지를 활성화하려면 세션 스토리지에 사용할 폴더를 만든 후 dcv.conf 파일에서 storage-root 파라미터를 구성합니다.

Linux에서 세션 스토리지를 활성화하려면

1. 세션 스토리지에 사용할 폴더를 만듭니다(예: /opt/session-storage/).
2. storage-root 파라미터를 구성합니다.

- a. /etc/dcv/로 이동하고 원하는 텍스트 편집기에서 dcv.conf를 엽니다.
- b. [session-management/automatic-console-session] 섹션에서 storage-root 파라미터를 찾습니다. 기존 경로를 1단계에서 만든 폴더의 전체 경로로 바꿉니다.

[session-management/automatic-console-session] 섹션에 storage-root 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[session-management/automatic-console-session]
storage-root="/opt/session-storage/"
```

경로에 %home%을 사용하여 현재 로그인한 사용자의 홈 디렉터리를 지정할 수도 있습니다. 예를 들어, 다음 파라미터는 세션 스토리지로 \$HOME/storage/ 디렉터리를 사용합니다.

```
[session-management/automatic-console-session]
storage-root="%home%/storage/"
```

Note

지정된 하위 디렉터리가 없으면 세션 스토리지가 비활성화됩니다.

3. 파일을 저장하고 닫습니다.
4. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.
5. 세션을 시작하고 --storage-root 옵션을 지정합니다. 자세한 정보는 [NICE DCV 세션 시작](#)을 참조하세요.

Linux NICE DCV 서버에서 프린터 설정

NICE DCV를 사용하면 로컬 리디렉션 프린터 또는 가상 NICE DCV 프린터로 인쇄할 수 있습니다.

지원되는 Linux 배포판을 사용하는 경우 인쇄를 지원하도록 NICE DCV 서버를 구성해야 합니다.

Windows NICE DCV 서버를 사용 중인 경우 추가 구성이 필요하지 않습니다.

Linux NICE DCV 서버에서 프린터 리디렉션을 활성화하려면 다음과 같이 하세요.

1. 서버에 CUPS 서비스를 설치합니다.
 - Amazon Linux 2, RHEL, CentOS

```
$ sudo yum install cups
```

- Ubuntu

```
$ sudo apt-get install cups
```

- SUSE Linux Enterprise

```
$ sudo zypper install cups
```

2. 프린터 관리자 그룹에 dcv 사용자를 추가합니다. 프린터 관리자 그룹의 이름은 운영 체제에 따라 다를 수 있습니다. 예를 들어 프린터 관리자 그룹의 이름이 lpadmin으로 지정되어 있는 경우 다음 명령을 실행합니다.

```
$ usermod -a -G lpadmin dcv
```

3. cups 구성 파일의 SystemGroup 파라미터에 프린터 관리자 그룹이 참조되어 있는지 확인합니다. 예를 들어 프린터 관리자 그룹에 이름이 lpadmin으로 지정되어 있는 경우 텍스트 편집기를 사용하여 /etc/cups/cups-files.conf를 열고 다음 행을 찾습니다.

```
SystemGroup lpadmin
```

구성 파일에 해당 행이 나타나면 설치가 완료된 것입니다. 다음 단계로 이동합니다.

구성 파일에 해당 행이 나타나지 않는 경우, 다음 형식으로 직접 추가한 다음 파일을 저장하고 닫습니다.

```
SystemGroup printer_admin_groupname
```

4. (SUSE Linux Enterprise만 해당) 프린터 관리자 그룹에 cups 로컬 인증서를 읽을 권한이 있는지 확인하세요. 이 인증서는 /var/run/cups/certs/ 디렉터리에 있습니다. 예를 들어 프린터 관리자 그룹의 이름이 lpadmin으로 지정되어 있는 경우 다음 명령을 실행합니다.

```
$ sudo chgrp -R lpadmin /var/run/cups/certs/ && chmod g+x /var/run/cups/certs
```

5. cups 서비스를 다시 시작합니다.

```
$ sudo systemctl restart cups
```

6. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

프린터 문제 해결

SUSE Linux Enterprise 및 RHEL 8이 프린터 소켓 연결을 방해할 수 있습니다. 이러한 운영 체제 중 하나를 실행 중이고 프린트 문제가 있는 경우, 로그 파일을 확인하여 이것이 원인인지 확인하세요.

텍스트 편집기를 사용하여 /var/log/audit/audit.log를 열고 로그에 다음과 비슷한 행이 있는지 확인합니다.

```
type=AVC msg=audit(1617716179.487:504): avc: denied { connectto } for pid=33933
comm="dcvcupsbackend"
path=002F636F6D2F6E696365736F66747761726552F6463762F637570732F636F6E736F6C65
scontext=system_u:system_r:cupsd_t:s0-s0:c0.c1023
tcontext=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
tclass=unix_stream_socket permissive=0
```

로그 파일에 비슷한 행이 있는 경우 운영 체제가 프린터 소켓에 대한 액세스를 차단하고 있는 것입니다.

이 문제를 해결하려면 프린터 소켓에 대한 액세스를 허용하는 cups 정책을 생성해야 합니다. 이렇게 하려면 다음 단계를 수행하십시오.

- 필수 정책 파일을 생성합니다. 선호하는 텍스트 편집기를 사용하여 cupsd_policy라는 이름의 새로운 파일을 생성하고 다음 내용을 추가합니다.

```
#===== cupsd_t =====
allow cupsd_t unconfined_t:unix_stream_socket connectto;
```

- 정책을 설치합니다.

```
$ ausearch -c 'dcvcupsbackend' --raw | audit2allow -M dcv-printer-policy
```

```
$ semodule -X 300 -i dcv-printer-policy.pp
```

Linux NICE DCV 서버에서 클립보드 구성

Linux 운영 체제는 내용을 복사하고 붙여 넣는 데 사용할 수 있는 두 가지 버퍼로 기본 선택 항목과 클립보드를 제공합니다. 내용을 기본 선택 항목으로 복사하려면 마우스 커서를 사용하여 콘텐츠를 강조 표시합니다. 기본 선택 항목에서 붙여 넣으려면 마우스의 커서 또는 Shift+Insert 키보드 단축키를 사용합니다. 콘텐츠를 클립보드로 복사하려면 콘텐츠를 강조 표시하고 컨텍스트(오른쪽 클릭) 메뉴에서 Copy(복사)를 선택합니다. 클립보드에서 붙여 넣으려면 컨텍스트(오른쪽 클릭) 메뉴에서 Paste(붙여 넣기)를 선택합니다.

Linux NICE DCV 서버에서는 클라이언트와 서버 간에 복사 및 붙여 넣기 작업을 수행할 때 기본 선택 항목이나 클립보드를 사용하도록 서버를 구성할 수 있습니다.

주제

- [기본 선택 항목에 클라이언트 클립보드 콘텐츠 붙여 넣기](#)
- [클라이언트 클립보드에 기본 선택 내용 복사](#)

기본 선택 항목에 클라이언트 클립보드 콘텐츠 붙여 넣기

기본적으로 클라이언트에서 복사된 내용은 클립보드에 저장됩니다. 이 콘텐츠를 서버에 붙여 넣으려면 컨텍스트(오른쪽 클릭) 메뉴를 사용하여 클립보드에서 붙여 넣어야 합니다.

클립보드 콘텐츠를 기본 선택 항목에 배치하도록 서버를 구성할 수 있습니다. 이렇게 하면 사용자가 컨텍스트(마우스 오른쪽 클릭) 메뉴를 사용하여 클립보드에 복사된 내용을 붙여 넣을 수 있습니다. 또는 마우스의 가운데 버튼 또는 Shift+Insert 키보드 단축키를 사용하여 기본 선택 항목에서 복사된 내용을 붙여 넣을 수 있습니다.

클립보드 콘텐츠를 기본 선택 항목에 배치하도록 서버를 구성하려면

1. /etc/dcv/로 이동하고 원하는 텍스트 편집기에서 dcv.conf를 엽니다.
2. [clipboard] 섹션에서 primary-selection-paste 파라미터를 찾은 후, 값을 true로 설정합니다.

[clipboard] 섹션에 primary-selection-paste 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[clipboard]
primary-selection-paste=true
```


3. 파일을 저장하고 닫습니다.
4. NICE DCV 세션을 [중지](#)하고 [다시 시작](#)합니다.

클라이언트 클립보드에 기본 선택 내용 복사

기본적으로 사용자는 클립보드를 사용하여 서버에서 클라이언트로만 콘텐츠를 복사할 수 있습니다. 즉, 기본 선택 항목에 복사된 내용을 클라이언트에 붙여 넣을 수 없습니다.

주요 선택 콘텐츠를 클립보드에 배치하도록 서버를 구성할 수 있습니다. 즉, 사용자가 서버의 기본 선택 항목에 콘텐츠를 복사하면 해당 콘텐츠도 클립보드에 복사됩니다. 사용자가 클립보드의 내용을 클라이언트에 붙여 넣을 수도 있다는 뜻입니다.

기본 선택 콘텐츠를 클립보드에 배치하도록 서버를 구성하려면

1. /etc/dcv/로 이동하고 원하는 텍스트 편집기에서 dcv.conf를 엽니다.
2. [clipboard] 섹션에서 primary-selection-copy 파라미터를 찾은 후, 값을 true로 설정합니다.

[clipboard] 섹션에 primary-selection-copy 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[clipboard]
primary-selection-copy=true
```

3. 파일을 저장하고 닫습니다.
4. NICE DCV 세션을 [중지](#)하고 [다시 시작](#)합니다.

다중 채널 오디오 구성

NICE DCV는 NICE DCV 네이티브 클라이언트를 사용할 때 7.1 오디오 채널까지 지원합니다. 웹 브라우저 클라이언트는 스테레오 2.0 오디오 채널만 지원합니다.

NICE DCV는 다음과 같은 다중 채널 오디오 구성을 지원합니다.

- 스테레오 2.0(2채널)
- 쿼드러포닉 4.0(4채널)
- 서라운드 5.1(6채널)

• 서라운드 7.1(8채널)—Windows NICE DCV 서버만 해당



클라이언트가 서버에서 제공한 채널 수보다 적은 수의 오디오 채널을 요청하면 서버는 채널 수를 다운믹스합니다. 이는 클라이언트가 요청한 채널 수와 일치시키기 위한 것입니다. 예를 들어 서버는 서라운드 사운드 7.1까지 지원하는 반면 클라이언트는 서라운드 사운드 5.1을 요청한다고 가정해 보겠습니다. 서버는 오디오를 5.1로 다운믹스합니다.

서버는 소스 애플리케이션의 오디오 출력과 일치하도록 오디오를 자동으로 다운믹스하지 않습니다. 예를 들어 소스 애플리케이션은 서라운드 사운드 7.1을 제공하는 반면 클라이언트는 스테레오 2.0만

지원한다고 가정해 보겠습니다. 전면 왼쪽 및 전면 오른쪽 오디오 채널만 클라이언트로 스트리밍됩니다. 나머지 채널은 손실됩니다. 이 경우 오디오 채널 손실을 방지하려면 NICE DCV 서버가 오디오 채널을 다운믹스하도록 구성하세요.

주제

- [Windows NICE DCV 서버의 오디오 채널 구성](#)
- [Linux NICE DCV 서버의 오디오 채널 구성](#)

Windows NICE DCV 서버의 오디오 채널 구성

Windows 서버는 서라운드 사운드 7.1(8개 오디오 채널)을 지원합니다. 기본 구성은 스테레오입니다. 하지만 서버가 다른 구성을 사용하도록 구성할 수 있습니다.

Windows 서버의 오디오 채널 구성

1. 사운드 제어판을 엽니다. 데스크톱의 작업 표시줄에서 스피커 아이콘을 마우스 오른쪽으로 클릭하고 사운드를 선택합니다.
2. 재생 탭을 열고 NICE DCV 스피커를 선택합니다.
3. 구성을 선택합니다.
4. 원하는 채널 구성을 선택합니다.
5. 확인을 선택합니다.

Linux NICE DCV 서버의 오디오 채널 구성

Linux 서버는 기본적으로 스테레오 2.0(2개 오디오 채널)을 지원하며 다중 채널 오디오를 지원하려면 몇 가지 추가 구성이 필요합니다.

이때 다음 작업을 수행해야 합니다.

1. 사운드 서버를 구성합니다. PulseAudio
2. 장치를 사용하도록 NICE DCV 서버를 구성합니다. PulseAudio
3. 사용할 채널 수를 구성합니다.

사운드 서버를 PulseAudio 구성하려면

1. 원하는 텍스트 편집기로 `/etc/pulse/default.pa`를 엽니다.

2. 파일 끝부분에 다음 행을 추가합니다.

```
load-module module-null-sink sink_name=dcv format=s16be channels=6
channel_map=front-left,front-right,rear-left,rear-right,front-center,lfe
rate=48000 sink_properties="device.description='DCV Audio Speakers'"
```

3. 파일을 저장하고 닫습니다.

사운드 서버를 구성한 후에는 PulseAudio 사운드 서버에서 PulseAudio 오디오를 캡처하도록 NICE DCV 서버를 구성해야 합니다.

장치를 사용하도록 NICE DCV 서버를 구성하려면 PulseAudio

1. 원하는 텍스트 편집기로 /etc/dcv/dcv.conf를 엽니다.
2. [audio] 섹션에서 grab-device 파라미터를 찾습니다. 그런 다음 기존 값을 이전 단계에서 검색한 디바이스 이름으로 바꿉니다.

[audio] 섹션에 grab-device 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[audio]
grab-device="DCV Audio Speakers"
```

3. 파일을 저장하고 닫습니다.

사운드 서버에서 오디오를 캡처하도록 NICE DCV 서버를 PulseAudio 구성한 후 사용할 채널 수를 지정할 수 있습니다.

사용할 채널 수를 구성하려면 다음과 같이 하세요.

1. 원하는 텍스트 편집기로 /etc/dcv/dcv.conf를 엽니다.
2. [audio] 섹션에서 source-channels 파라미터를 찾습니다. 그런 다음 기존 채널 수를 2.0용 2개, 4.0용 4개, 또는 5.1용 6개 중 하나로 바꿉니다.

[audio] 섹션에 source-channels 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[audio]
source-channels=channels
```

3. 파일을 저장하고 닫습니다.
4. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

HTTP 헤더 구성

사용자가 웹 브라우저 클라이언트를 사용하여 세션에 연결할 때 NICE DCV 클라이언트에 추가 HTTP 응답 헤더를 보내도록 NICE DCV 서버를 구성할 수 있습니다. 응답 헤더는 사용자가 연결하는 NICE DCV 서버에 대한 추가 정보를 제공할 수 있습니다.

주제

- [Windows NICE DCV 서버에서 HTTP 헤더 구성](#)
- [Linux NICE DCV 서버에서 HTTP 헤더 구성](#)

Windows NICE DCV 서버에서 HTTP 헤더 구성

Windows에서 HTTP 헤더를 구성하려면 Windows 레지스트리 편집기를 사용하여 `web-extra-http-headers` 파라미터를 구성합니다.

Windows에서 HTTP 헤더를 구성하려면

1. Windows 레지스트리 편집기를 엽니다.
2. `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/` 키로 이동합니다.
3. 탐색 창에서 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 연결 키를 선택합니다. 그런 다음 새로 만들기, 문자열을 선택합니다.
4. 이름에 `web-extra-http-headers`를 입력하고 Enter 키를 누릅니다.
5. 파라미터를 엽니다. `web-extra-http-headers` 값 데이터에 HTTP 헤더 이름과 값을 다음 형식으로 입력합니다.

```
[("header-name", "header-value")]
```

여러 헤더를 지정하려면 쉼표로 구분된 목록에 추가합니다.

```
[("header1-name", "header1-value"), ("header2-name", "header2-value")]
```

6. 확인을 선택하고 Windows 레지스트리 편집기를 닫습니다.

7. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

Linux NICE DCV 서버에서 HTTP 헤더 구성

Linux에서 HTTP 헤더를 구성하려면 `dcv.conf` 파일에서 `web-extra-http-headers` 파라미터를 구성합니다.

Linux에서 HTTP 헤더를 구성하려면

1. 원하는 텍스트 편집기로 `/etc/dcv/dcv.conf`를 엽니다.
2. `[connectivity]` 섹션을 찾습니다. HTTP 헤더 이름과 값을 다음 형식으로 지정합니다.

```
[connectivity]
web-extra-http-headers=[("header-name", "header-value")]
```

여러 헤더를 지정하려면 쉼표로 구분된 목록에 추가합니다.

```
[connectivity]
web-extra-http-headers=[("header1-name", "header1-value"), ("header2-name",
"header2-value")]
```

3. 파일을 저장하고 닫습니다.
4. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

NICE DCV 인증 구성

기본적으로 클라이언트는 NICE DCV 세션에 연결하기 전에 NICE DCV가 호스팅되는 서버에 대해 인증을 수행해야 합니다. 클라이언트가 인증에 실패하는 경우, 이는 클라이언트가 세션에 연결할 수 없기 때문일 수 있습니다. 클라이언트가 서버에 대해 인증하지 않고 세션에 연결할 수 있도록 클라이언트 인증 요구 사항을 비활성화할 수 있습니다.

NICE DCV는 다음 인증 방법을 지원합니다.

- `system` - 기본 인증 방법입니다. 클라이언트 인증이 기본 운영 체제에 위임됩니다. Windows NICE DCV 서버의 경우 인증이 위임됩니다. WinLogon Linux NICE DCV 서버의 경우 인증이 PAM에 위임됩니다. 클라이언트는 NICE DCV 세션에 연결할 때 시스템 보안 인증 정보를 제공합니다. 클라이언트에 NICE DCV 서버에 대한 적절한 로그인 보안 인증 정보가 있는지 확인하세요.

- none - NICE DCV 세션에 연결할 때 클라이언트 인증이 필요하지 않습니다. NICE DCV 서버는 세션 연결을 시도하는 모든 클라이언트에 액세스 권한을 부여합니다.

클라이언트가 NICE DCV 서버에서 사용하는 인증 방법을 알고 있는지 확인하세요. 또한 세션에 연결하는 데 필요한 정보를 가지고 있는지도 확인해야 합니다.

주제

- [Windows에서 인증 구성](#)
- [Linux에서 인증 구성](#)
- [외부 인증자를 사용한 권한 부여 구성](#)

Windows에서 인증 구성

NICE DCV 서버의 인증 방법을 변경하려면 Windows 레지스트리 편집기를 사용하여 authentication 파라미터를 구성해야 합니다.

Windows에서 인증 방법을 변경하려면

1. Windows 레지스트리 편집기를 엽니다.
2. HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/security/ 키로 이동하여 인증 파라미터를 선택합니다.

레지스트리 키에 authentication 파라미터가 없는 경우 하나를 만듭니다.

- a. 탐색 창에서 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 인증 키를 선택합니다. 그런 다음 새로 만들기, 문자열 값을 선택합니다.
 - b. 이름에 authentication을 입력하고 Enter 키를 누릅니다.
3. 인증 파라미터를 엽니다. Value data(값 데이터)에 system 또는 none을 입력합니다.
 4. 확인을 선택하고 Windows 레지스트리 편집기를 닫습니다.

Windows 자격 증명 공급자

Windows 보안 인증 공급자를 사용하면 사용자가 DCV 서버에 대해 성공적으로 인증하는 경우 Windows 로그인을 무시할 수 있습니다.

Windows 자격 증명 공급자는 DCV authentication 파라미터가 system으로 설정된 경우에만 지원됩니다. DCV authentication 파라미터를 none으로 설정하면 사용자가 DCV 서버에 대해 자동으로 인증된 후 Windows에 수동으로 로그인해야 합니다.

Windows 보안 인증 공급자는 NICE DCV 서버를 설치할 때 기본적으로 활성화됩니다.

Windows 자격 증명 공급자를 비활성화하려면

1. Windows 레지스트리 편집기를 엽니다.
2. HKEY_LOCAL_MACHINE\ 소프트웨어\ Microsoft\ Windows\ 인증\ 자격 증명 공급자 CurrentVersion\ {8A2C93D0-D55F-4045-99D7-B27F5E263407} 키로 이동합니다.
3. 편집, 새로 만들기, DWORD 값을 선택합니다.
4. 이름에 **Disabled**를 입력합니다.
5. 값을 엽니다. Value data(값 데이터)에 1을 입력한 후 확인을 선택합니다.
6. Windows 레지스트리 편집기를 닫습니다.

Windows 자격 증명 공급자를 다시 활성화하려면

1. Windows 레지스트리 편집기를 엽니다.
2. HKEY_LOCAL_MACHINE\ 소프트웨어\ Microsoft\ Windows\ 인증\ 자격 증명 공급자 CurrentVersion\ {8A2C93D0-D55F-4045-99D7-B27F5E263407} 키로 이동합니다.
3. Disabled(비활성) 값을 엽니다. Value data(값 데이터)에 0을 입력한 후 확인을 선택합니다.
4. Windows 레지스트리 편집기를 닫습니다.

Linux에서 인증 구성

NICE DCV 서버의 인증 방법을 변경하려면 dcv.conf 파일에서 authentication 파라미터를 구성해야 합니다.

Linux에서 인증 방법을 변경하려면

1. /etc/dcv/로 이동하고 원하는 텍스트 편집기에서 dcv.conf를 엽니다.
2. [security] 섹션에서 authentication 파라미터를 찾습니다. 그런 다음 기존 값을 system 또는 none으로 바꿉니다.

[security] 섹션에 authentication 파라미터가 없는 경우, 다음 형식을 사용하여 추가합니다.


```
[security]
authentication=method
```

3. 파일을 저장하고 닫습니다.

PAM 서비스

Linux에서 NICE DCV authentication 파라미터가 system으로 설정된 경우 PAM 서비스를 실행하여 인증을 수행합니다.

기본적으로 NICE DCV 서버에서 실행되는 PAM(권한 액세스 관리) 서비스는 /etc/pam.d/dcv입니다.

NICE DCV를 통해 사용자를 인증할 때 PAM이 수행하는 단계를 변경하려면 dcv.conf의 authentication 섹션에서 pam-service 파라미터를 설정할 수 있습니다.

PAM 서비스를 변경하려면 다음과 같이 하세요.

1. 루트로서 /etc/pam.d 디렉터리로 이동하여 dcv-custom과 같은 새 파일을 생성합니다.
2. 선호하는 텍스트 편집기를 사용하여 dcv-custom 파일을 엽니다. PAM 서비스 파일의 구문은 시스템 설명서를 참조하세요.
3. /etc/dcv/로 이동하고 원하는 텍스트 편집기에서 dcv.conf를 엽니다.
4. [authentication] 섹션에서 pam-service 파라미터를 찾습니다. 그런 다음 기존 서비스 이름을 새 PAM 서비스 이름으로 교체합니다.

[authentication] 섹션에 pam-service 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[authentication]
pam-service=service_name
```

Note

PAM 서비스 이름은 /etc/pam.d에서 생성한 파일 이름과 반드시 일치해야 합니다.

5. 파일을 저장하고 닫습니다.

외부 인증자를 사용한 권한 부여 구성

외부 인증자를 사용하도록 DCV를 구성할 수 있습니다. 이 프로세스에 대한 자세한 내용은 [외부 인증 사용](#) 섹션을 참조하세요.

NICE DCV 권한 부여 구성

권한 부여는 NICE DCV 클라이언트에 특정 NICE DCV 기능에 대한 권한을 부여하거나 거부하는 데 사용됩니다. NICE DCV에서 권한 부여는 권한 파일을 사용하여 구성됩니다. 권한 파일은 특정 사용자가 세션에 연결할 때 사용할 수 있는 특정 NICE DCV 기능을 정의합니다.

NICE DCV는 두 가지 유형의 권한 파일을 지원합니다.

주제

- [기본 권한 파일](#)
- [사용자 지정 권한 파일](#)
- [권한 파일 작업](#)

기본 권한 파일

세션을 생성할 때 사용자 지정 권한 파일을 지정하지 않으면 기본 권한 파일이 모든 세션에 사용됩니다. 기본 사용 권한 파일은 세션 소유자에게만 모든 기능에 대한 전체 액세스 권한을 부여합니다.

사용자 지정 권한 부여를 포함하도록 기본 권한 파일을 사용자 지정할 수 있습니다. 기본 사용 권한 파일은 Windows NICE DCV 서버에서는 C:\Program Files\NICE\DCV\Server\conf\default.perm, Linux NICE DCV 서버에서는 /etc/dcv/default.perm에 있습니다.

기본 권한 파일 사용자 지정에 대한 자세한 내용은 [권한 파일 작업](#) 단원을 참조하십시오.

사용자 지정 권한 파일

사용자 지정 권한 파일을 사용하여 특정 사용자 또는 그룹이 NICE DCV 세션에 연결할 때 액세스할 수 있는 기능을 명시적으로 정의할 수 있습니다. 사용자 지정 권한 파일을 사용할 때 기본 권한 파일을 재정의합니다.

사용자 지정 권한 파일을 사용하려면 먼저 권한 파일을 만들어야 합니다. 그런 다음 `dcv create-session` 명령으로 `--permissions-file` 옵션을 사용하여 세션을 시작할 때 지정합니다. 세션 시작에 대한 자세한 내용은 [NICE DCV 세션 시작](#) 단원을 참조하십시오.

사용자 지정 권한 파일 만들기에 대한 자세한 내용은 [권한 파일 작업](#) 단원을 참조하십시오.

권한 파일 작업

원하는 텍스트 편집기를 사용하여 사용자 지정 권한 파일을 만들거나 기존 권한 파일을 업데이트할 수 있습니다. 권한 파일은 일반적으로 다음과 같은 형식을 취합니다.

```
#import file_to_import

[groups]
group_definitions

[aliases]
alias_definitions

[permissions]
user_permissions
```

다음 섹션에서는 권한 파일을 업데이트하거나 만들 때 섹션을 채우는 방법에 대해 설명합니다.

목차

- [권한 파일 가져오기](#)
- [그룹 생성](#)
- [별칭 생성](#)
- [권한 추가](#)

권한 파일 가져오기

imports 섹션은 일반적으로 권한 파일의 첫 번째 섹션입니다. 이 섹션에서는 기존 권한 파일을 참조하고 포함시키는 방법을 알아봅니다. 또한 이를 사용하여 이전에 정의된 NICE DCV 권한을 권한 파일에 통합할 수 있습니다.

권한 파일에는 가져오기가 여러 개 포함될 수 있습니다. 가져온 권한 파일은 다른 권한 파일을 가져올 수 있습니다.

권한 파일을 사용자의 권한 파일로 가져오려면

- #import 문을 사용하고 파일 위치를 절대 경로 또는 상대 경로를 사용하여 지정합니다.

- Windows NICE DCV 서버

```
#import ..\file_path\file
```

- Linux NICE DCV 서버

```
#import ../file_path/file
```

예

다음 명령문은 절대 경로를 사용하여 이름이 `dcv-permissions.file`로 지정된 권한 파일을 가져옵니다. 파일은 Windows NICE DCV 서버의 NICE DCV 설치 폴더에 있습니다.

```
#import c:\Program Files\NICE\DCV\dcv-permissions.file
```

그룹 생성

권한 파일의 `[groups]` 섹션에서 비슷한 사용 사례 또는 권한 요구 사항이 있는 사용자들에 대해 사용자 그룹을 정의할 수 있습니다. 그룹에 특정 권한을 할당할 수 있습니다. 그룹에 할당된 권한은 그룹에 포함된 모든 사용자에게 적용됩니다.

권한 파일에서 그룹을 만들려면 먼저 파일에 그룹 섹션 머리글을 추가해야 합니다.

```
[groups]
```

그런 다음 섹션 머리글 아래에 그룹을 만들 수 있습니다. 새 그룹을 만들려면 그룹 이름을 입력한 후, 쉼표로 구분된 목록에서 그룹 구성원을 지정합니다. 그룹 구성원은 개별 사용자, 다른 그룹 및 운영 체제 사용자 그룹일 수 있습니다.

```
group_name=member_1, member_2, member_3
```

그룹에 사용자를 추가하는 방법

사용자 이름을 지정합니다.

Note

사용자 이름에 `user:` 접두사를 붙일 수 있습니다. Windows 도메인 사용자 이름에는 도메인 이름이 포함될 수 있습니다.

```
group_name=user_1, user:user_2, domain_name\user_3
```

그룹에 기존 그룹을 추가하려면

`group:`으로 시작하는 그룹 이름을 지정합니다.

```
group_name=group:group_1, group:group_2
```

그룹에 운영 체제 사용자 그룹을 추가하려면(Linux NICE DCV 서버만 해당)

`osgroup:`으로 시작하는 그룹의 이름을 지정합니다.

```
group_name=osgroup:os_group_1, osgroup:os_group2
```

예

다음 예제에서는 그룹 섹션 제목을 추가하고 이름이 `my-group`으로 지정된 그룹을 생성합니다. 이 그룹에는 개별 사용자가 포함됩니다. 사용자의 이름은 `john`과 `jane`입니다. 그중 하나는 이름이 `observers`로 지정된 기존 그룹입니다. 다른 하나는 이름이 `guests`로 지정된 다음과 같은 운영 체제 사용자 그룹입니다.

```
[groups]
my-group=john, user:jane, group:observers, osgroup:guests
```

별칭 생성

권한 파일의 `[aliases]` 섹션에서 NICE DCV 기능 세트를 생성할 수 있습니다. 별칭을 정의한 후 그룹 또는 개별 사용자에게 이를 사용할 권한을 부여하거나 거부할 수 있습니다. 별칭에 대한 권한을 부여하거나 거부하면 별칭에 포함된 모든 기능에 대한 권한을 부여하거나 거부하는 것입니다.

권한 파일에서 별칭을 만들려면 먼저 파일에 별칭 섹션 머리글을 추가해야 합니다.

```
[aliases]
```

그런 다음 섹션 머리글 아래에 별칭을 만들 수 있습니다. 새 별칭을 만들려면 별칭 이름을 입력한 후, 쉼표로 구분된 목록에서 별칭 구성원을 지정합니다. 별칭 구성원은 개별 NICE DCV 기능 또는 다른 별칭일 수 있습니다.

```
alias_name=member_1, member_2, member_3
```

예

다음 예제에서는 별칭 섹션 제목을 추가하고 이름이 file-management로 지정된 별칭을 생성합니다. 여기에는 file-upload 및 file-download 기능과 이름이 clipboard-management으로 지정된 기존 별칭이 포함됩니다.

```
[aliases]
file-management=file-upload, file-download, clipboard-management
```

권한 추가

권한 파일의 [permissions] 섹션에서 특정 기능 또는 별칭에 대한 사용자 및 그룹 액세스를 제어할 수 있습니다.

권한 파일에 권한을 추가하려면 먼저 파일에 권한 섹션 머리글을 추가합니다.

```
[permissions]
```

그런 다음 섹션 머리글 아래에 권한을 추가할 수 있습니다. 권한을 추가하려면 이를 제어하는 액터, 적용할 규칙 및 규칙이 적용되는 기능을 지정합니다.

```
actor rule features
```

액터는 사용자, 그룹 또는 운영 체제 그룹일 수 있습니다. 그룹에는 group:으로 시작하는 접두사가 붙어야 합니다. 운영 체제 그룹에는 osgroup: 접두사가 붙어야 합니다. NICE DCV에는 세션 소유자를 참조하는 데 사용할 수 있는 %owner% 참조가 기본으로 포함되어 있습니다. 또한 모든 사용자를 참조하는 데 사용할 수 있는 기본 %any% 참조를 참조하는 데에도 사용할 수 있습니다.

다음 규칙을 권한 문에서 사용할 수 있습니다.

- allow - 기능에 대한 액세스 권한을 부여합니다.
- disallow - 기능에 대한 액세스 권한을 거부하지만, 후속 권한으로 재정의할 수 있습니다.

- deny - 기능에 대한 액세스 권한을 거부하며 후속 권한으로 재정의할 수 없습니다.

기능에는 개별 NICE DCV 기능, 별칭 또는 이들의 조합이 포함될 수 있습니다. 기능 목록은 공백으로 구분해야 합니다. NICE DCV에는 NICE DCV 기능을 모두 포함하는 기본 builtin 별칭이 포함되어 있습니다.

권한 파일에서 다음 기능을 참조할 수 있습니다.

- audio-in - 클라이언트의 오디오를 NICE DCV 서버에 삽입합니다.
- audio-out - 클라이언트에서 NICE DCV 서버 오디오를 재생합니다.
- builtin - 모든 기능입니다.
- clipboard-copy - 데이터를 NICE DCV 서버에서 클라이언트 클립보드로 복사합니다.
- clipboard-paste - 데이터를 클라이언트 클립보드에서 NICE DCV 서버로 붙여 넣습니다.
- display - NICE DCV 서버로부터 시각 데이터를 수신합니다.
- extensions-client - NICE DCV 클라이언트에서 설치된 확장 프로그램을 시작할 수 있습니다.
- extensions-server - NICE DCV 서버에서 설치된 확장 프로그램을 시작할 수 있습니다.
- file-download - 세션 스토리지에서 파일을 다운로드합니다.
- file-upload - 파일을 세션 스토리지에 업로드합니다.
- gamepad - 세션에서 클라이언트 컴퓨터에 연결된 게임패드를 사용합니다. NICE DCV 2022.0 이상에서 지원됩니다.
- keyboard - 클라이언트 키보드에서 NICE DCV 서버로 입력합니다.
- keyboard-sas - 보안 주의 시퀀스를 사용합니다(Ctrl+Alt+Del). keyboard 기능이 필요합니다. NICE DCV 2017.3 이상에서 지원됩니다.
- mouse - 클라이언트 포인터에서 NICE DCV 서버로 입력합니다.
- pointer - NICE DCV 서버 마우스 위치 이벤트 및 포인터 모양을 봅니다. NICE DCV 2017.3 이상에서 지원됩니다.
- printer - NICE DCV 서버에서 클라이언트로 PDF 또는 XPS 파일을 생성합니다.
- screenshot - 원격 데스크톱의 스크린샷을 저장합니다. NICE DCV 2021.2 이상에서 지원됩니다.

screenshot 권한 부여를 제거할 때는 clipboard-copy 권한을 비활성화하는 것이 좋습니다. 이렇게 하면 사용자가 서버의 클립보드에서 스크린샷을 캡처한 다음 클라이언트에 붙여 넣을 수 없습니다. screenshot 권한 부여가 거부되면 Windows 및 macOS는 외부 도구가 클라이언트의 스크린샷을 캡처하지 못하도록 차단합니다. 예를 들어 NICE DCV 클라이언트 창에서 Windows 캡처 도구를 사용하면 검은색 이미지가 나타납니다.

- smartcard - 클라이언트에서 스마트 카드를 읽습니다.
- stylus - 3D 포인팅 디바이스 또는 그래픽 태블릿 등 특수화된 USB 디바이스에서 입력합니다.
- touch - 기본 터치 이벤트를 사용합니다. DCV 2017.3 이상에서 지원됩니다.
- unsupervised-access - 공동 작업 세션에서 사용자의 소유자 없는 액세스 권한을 설정하는 데 사용합니다.
- usb - 클라이언트의 USB 디바이스를 사용합니다.
- webcam - 세션에서 클라이언트 컴퓨터에 연결된 웹캠을 사용합니다. NICE DCV 2021.0 이상에서 지원됩니다.
- webauthn-redirect - 원격 브라우저의 Webauthn 요청을 로컬 클라이언트로 리디렉션합니다. NICE DCV 2023.1 이상에서 지원됩니다.

예

다음 예제에서는 권한 섹션 머리글을 추가하고 권한 네 개를 추가합니다. 첫 번째 권한은 john이라는 사용자에게 display, file-upload 및 file-download 기능에 대한 액세스를 부여합니다. 두 번째 권한은 audio-in 및 audio-out 기능과, clipboard-management 기능 별칭에 대한 observers 그룹 액세스를 거부합니다. 세 번째 권한은 guests 운영 체제 그룹에 clipboard-management 및 file-management 별칭에 대한 액세스를 부여합니다. 네 번째 권한은 세션 소유자에게 모든 기능에 대한 액세스를 부여합니다.

```
[permissions]
john allow display file-upload file-download
group:observers deny audio-in audio-out clipboard-management
osgroup:guests allow clipboard-management file-management
%owner% allow builtin
```


NICE DCV 세션 관리

클라이언트가 서버에 연결하려면 먼저 NICE DCV 서버에 NICE DCV 세션을 생성해야 합니다. 활성 세션이 있는 경우에만 클라이언트가 NICE DCV 서버에 연결할 수 있습니다.

모든 NICE DCV 세션에는 다음 속성이 있습니다.

- 세션 ID - NICE DCV 서버의 특정 세션을 식별하는 데 사용됩니다.
- 소유자 - 세션을 생성한 NICE DCV 사용자입니다. 기본적으로 소유자만 세션에 연결할 수 있습니다.

NICE DCV 클라이언트가 세션에 연결하려면 이 정보가 필요합니다.

주제

- [NICE DCV 세션 소개](#)
- [명령줄 도구를 사용하여 NICE DCV 세션 관리](#)
- [NICE DCV 세션 시작](#)
- [NICE DCV 세션 중지](#)
- [실행 중인 NICE DCV 세션 관리](#)
- [세션 시간대 관리](#)
- [NICE DCV 세션 보기](#)
- [NICE DCV 세션 스크린샷 찍기](#)

NICE DCV 세션 소개

NICE DCV는 콘솔 세션과 가상 세션이라는 두 가지 유형의 세션을 제공합니다. 다음 표에는 두 가지 유형의 세션 간 차이점이 요약되어 있습니다.

세션 유형	지원	다중 세션	필요한 권한	직접 화면 캡처	GPU 가속 OpenGL 지원
콘솔	Linux 및 Windows NICE DCV 서버	아니요, 각 서버 에는 콘솔 세션이	관리자만 세션을 시작하고 종료할 수 있습니다.	예	예, 추 가 소 프트웨

세션 유형	지원	다중 세션	필요한 권한	직접 화면 캡처	GPU 가속 OpenGL 지원
		하나만 허용됩니다.			어가 필요하지 않습니다.
가상	Linux NICE DCV 서버만	예, 단일 서버에서 여러 가상 세션이 허용됩니다.	모든 사용자가 세션을 시작하고 종료할 수 있습니다.	아니요, 가상 세션마다 전용 X 서버(Xdcb)가 실행됩니다. 화면은 X 서버에서 캡처됩니다.	예, 하지만 DCV-GL 패키지가 필요합니다.

Note

동일한 NICE DCV 서버에서 콘솔 세션과 가상 세션을 동시에 실행할 수 없습니다.

콘솔 세션

콘솔 세션은 Windows 및 Linux NICE DCV 서버에서 지원됩니다. Windows NICE DCV 서버를 사용하는 경우 콘솔 세션만 사용할 수 있습니다.

NICE DCV 서버에서는 한 번에 콘솔 세션 하나만 호스팅할 수 있습니다. 콘솔 세션은 Windows NICE DCV 서버에서는 관리자가, Linux NICE DCV 서버에서는 루트 사용자가 만들고 관리합니다.

NICE DCV는 콘솔 세션을 통해 데스크톱 화면의 내용을 직접 캡처합니다. 서버가 GPU로 구성된 경우 NICE DCV 콘솔 세션은 GPU에 직접 액세스할 수 있습니다.

가상 세션

가상 세션은 Linux NICE DCV 서버에서만 지원됩니다.

동일한 NICE DCV 서버에서 여러 가상 세션을 동시에 호스팅할 수 있습니다. 가상 세션은 NICE DCV 사용자가 생성하고 관리합니다. NICE DCV 사용자는 자신이 생성한 세션만 관리할 수 있습니다. 루트 사용자는 현재 NICE DCV 서버에서 실행되는 모든 가상 세션을 관리할 수 있습니다.

가상 세션을 통해 NICE DCV는 X 서버 인스턴스(Xdcb)를 시작하고 X 서버 내에서 데스크톱 환경을 실행합니다. NICE DCV는 각 가상 세션에 대해 새로운 전용 X 서버 인스턴스를 시작합니다. 각 가상 세션은 X 서버 인스턴스에서 제공하는 디스플레이를 사용합니다.

Note

NICE DCV는 각 가상 세션이 독립적인 Xdcb로 표시되도록 보장하지만 사용자의 홈 폴더에 있는 파일, D-Bus 서비스 및 디바이스를 비롯한 기타 많은 시스템 리소스는 사용자별로 제공되므로 동일한 사용자가 여러 가상 세션에서 이를 공유하고 액세스할 수 있습니다. 공유 리소스에 대해 발생 가능한 문제를 완화할 수 있도록 운영 체제를 설정하지 않은 한, 동일한 사용자에게 대해 동일한 NICE DCV 서버에서 동시에 여러 가상 세션을 실행해서는 안 됩니다.

dcb-g1 패키지가 설치되고 라이선스가 부여된 경우 NICE DCV 가상 세션은 서버의 GPU에 대한 액세스를 공유합니다. 여러 가상 세션에서 하드웨어 기반 OpenGL을 공유하려면 dcb-g1.conf 파일을 구성하여 가상 X 서버 인스턴스를 GPU에 연결해야 합니다.

명령줄 도구를 사용하여 NICE DCV 세션 관리

NICE DCV 서버에는 NICE DCV 세션을 시작하고, 중지하고, 확인하는 데 사용할 수 있는 명령줄 도구가 포함되어 있습니다.

Windows NICE DCV 서버에서 명령줄 도구 사용

Windows NICE DCV 서버에서 명령줄 도구를 사용하려면 NICE DCV 설치 디렉터리에서 명령을 실행하거나 PATH 환경 변수에 NICE DCV 디렉터를 추가해야 합니다. NICE DCV 디렉터를 PATH 환경 변수에 추가하는 경우 모든 디렉터리의 명령을 사용할 수 있습니다.

NICE DCV 설치 디렉터리에서 명령줄 도구를 사용하려면 다음과 같이 하세요.

dcb.exe 파일이 있는 폴더(기본적으로 C:\Program Files\NICE\DCV\Server\bin\)로 이동하여 명령 프롬프트 창을 엽니다.

또는 다른 디렉터리에서 명령을 실행할 때 전체 경로를 지정할 수 있습니다.

```
"C:\> Program Files\NICE\DCV\Server\bin\dcv.exe" list-sessions
```

PATH 환경 변수에 NICE DCV 디렉터리를 추가하려면 다음과 같이 하세요.

1. 파일 탐색기에서 이 PC를 마우스 오른쪽 단추로 클릭하고 속성을 선택합니다.
2. 고급 시스템 설정을 선택합니다.
3. 고급 탭에서 환경 변수를 선택합니다.
4. 시스템 변수 섹션에서 Path(경로) 변수를 선택하고 Edit(편집)를 선택합니다.
5. 새로 만들기를 선택하고 NICE DCV 설치 디렉터리의 bin 폴더에 대한 전체 경로(예: C:\Program Files\NICE\DCV\Server\bin\)를 지정합니다.
6. 확인을 선택하고 환경 변수 창을 닫습니다.

Linux NICE DCV 서버에서 명령줄 도구 사용

Linux NICE DCV 서버에서는 명령줄 도구가 \$PATH 환경 변수에 자동으로 구성되어 있습니다. 모든 폴더에서 도구를 사용할 수 있습니다. 터미널 창을 열고 실행할 명령을 입력합니다.

명령줄 도구 사용

다음 표에는 사용 가능한 명령줄 도구 옵션이 나와 있습니다. 이 목록은 dcv 호출할 --help 때 를 사용하여 검색할 수 있습니다. 각 명령을 사용하는 방법에 대한 자세한 내용을 보려면 사용 정보를 보려는 명령 --help 뒤에 전달하십시오. 예를 들면 dcv create-session --help입니다.

Command	설명
create-session	새 DCV 세션 생성
close-session	활성 DCV 세션 닫기
describe-session	DCV 세션에 대해 설명하기
list-sessions	활성 DCV 세션 목록
	DCV 세션의 클라이언트 연결 목록

Command	설명
<code>list-connections</code>	
<code>close-connection</code>	활성 클라이언트 연결 닫기
<code>get-screenshot</code>	DCV 콘솔의 스크린샷 가져오기
<code>set-display-layout</code>	활성 DCV 세션의 디스플레이 레이아웃 설정
<code>set-name</code>	DCV 세션의 이름 설정
<code>set-permissions</code>	활성 DCV 세션의 권한 설정
<code>set-storage-root</code>	활성 DCV 세션의 스토리지 루트 설정
<code>reload-licenses</code>	실행 중인 모든 세션의 라이선스를 강제로 다시 로드합니다.
<code>get-config</code>	서버 구성 가져오기
<code>list-endpoints</code>	DCV 엔드포인트 목록
<code>set-config</code>	서버 구성 설정
<code>version</code>	DCV 버전 보기
<code>help</code>	도움말 보기

NICE DCV 세션 시작

기본값을 사용하여 [Windows NICE DCV 서버를 설치](#)하면 서버가 설치된 후 [콘솔 세션](#)이 자동으로 생성되고 활성화됩니다. 기본 콘솔 세션은 Administrator가 소유하며, 기본 세션 ID는 console입니다. 이 세션을 사용할 수도 있고, 세션을 [종료하고](#) 새 세션을 생성할 수도 있습니다.

NICE DCV 서버를 설치할 때 콘솔 세션이 자동으로 생성되지 않도록 선택한 경우, 콘솔 세션을 수동으로 생성해야 합니다. 또한 NICE DCV 서버를 설치한 후 언제든지 [자동 콘솔 세션 생성](#)을 활성화하거나 비활성화할 수 있습니다.

Note

Linux NICE DCV 서버는 설치 후 기본 콘솔 세션을 가져오지 않습니다.

온프레미스 또는 대체 클라우드 기반 서버에서 부동 라이선스를 사용 중인 상황에서 라이선스가 지원하는 최대 동시 세션 수를 초과했다고 가정해 보겠습니다. 이때 no licenses 오류가 발생할 수 있습니다. 이 오류가 발생한 경우 사용하지 않는 세션을 중지하여 라이선스를 해제하고 다시 시도합니다.

세션을 시작하려면 NICE DCV 서버가 실행되고 있어야 합니다. 자세한 설명은 [NICE DCV 서버 시작](#) 섹션을 참조하세요.

주제

- [콘솔 및 가상 세션 수동 시작](#)
- [자동 콘솔 세션 활성화](#)

콘솔 및 가상 세션 수동 시작

언제든지 NICE DCV 서버를 시작할 수 있습니다. 한 번에 콘솔 세션 하나만 실행할 수 있습니다. Linux NICE DCV 서버를 사용하는 경우 여러 가상 세션을 동시에 실행할 수 있습니다.

특히 Windows NICE DCV 서버를 사용하는 경우 세션을 생성하기 전에 `dcv list-sessions`를 실행하는 것이 좋습니다.

Windows 또는 Linux NICE DCV 서버에서 콘솔 세션 또는 가상 세션을 생성하려면 `dcv create-session` 명령을 사용합니다.

주제

- [명령문](#)

- [옵션](#)
- [예](#)

명령문

세션을 시작하기 위한 명령의 최소 구문은 다음과 같습니다.

```
dcv create-session session_ID
```

모든 옵션이 포함된 전체 구문은 다음과 같습니다.

```
dcv create-session \  
  --type console|virtual \  
  --name session_name \  
  --user username \  
  --owner owner_name \  
  --permissions-file /path_to/permissions_file \  
  --storage-root /path_to/storage_directory \  
  --gl on|off \  
  --max-concurrent-clients number_of_clients \  
  --init /path_to/init_script \  
  session_ID
```

Note

\ 기호는 구문에서 명령을 여러 줄로 분할하는 것을 나타냅니다.

dcv create-session --help를 사용하여 구문에 대한 빠른 참조를 표시할 수도 있습니다.

옵션

dcv create-session 명령과 함께 다음 옵션을 사용할 수 있습니다.

--type

이 옵션은 Linux NICE DCV 서버에서만 지원됩니다. 이 옵션은 만들 세션 유형을 지정합니다. 세션 유형은 console 또는 virtual일 수 있습니다.

타입: 문자열

허용된 값: `console | virtual`

필수 여부: 아니요

--name

세션의 이름을 지정합니다. 세션 이름은 최대 256자의 문자열로 지정할 수 있습니다. 문자열이 256자를 초과하면 명령이 실패합니다. 세션 이름은 실행 중인 세션에서 고유하지 않아도 됩니다.

언제든지 `dcv set-name` 명령을 사용하여 세션 이름을 변경할 수 있습니다. 자세한 정보는 [세션 이름 관리](#)를 참조하세요.

타입: 문자열

필수 항목 여부: 예

--user

이 옵션은 Linux NICE DCV의 가상 세션에서만 지원됩니다. 이 값은 세션을 만드는 데 사용할 사용자입니다. 루트 사용자만 다른 사용자를 가장할 수 있습니다.

타입: 문자열

필수 항목 여부: 아니요

--owner

세션 소유자를 지정합니다. 생략할 경우 현재 로그인한 사용자로 기본 설정됩니다.

타입: 문자열

필수 항목 여부: 아니요

--permissions-file

사용자 지정 권한 파일 경로를 지정합니다. 생략할 경우 서버 기본값이 기본 설정됩니다.

타입: 문자열

필수 항목 여부: 아니요

--storage-root

세션 스토리지에 사용되는 폴더 경로를 지정합니다.

`%home%`를 사용하여 현재 로그인되어 있는 사용자의 홈 디렉터리를 지정할 수 있습니다. 예를 들어, 다음은 세션 스토리지를 위한 디렉터리를 Windows 서버의 경우 `c:\Users\username\storage\`, Linux 서버의 경우 `$HOME/storage/`로 설정합니다.


```
--storage-root %home%/storage/
```

Note

지정된 하위 디렉터리가 없으면 세션 스토리지가 비활성화됩니다.

타입: 문자열

필수 항목 여부: 아니요

--gl

이 옵션은 Linux NICE DCV의 가상 세션에서만 지원됩니다. 기본 dcv-gl 상태를 재정의하며, on 또는 off일 수 있습니다.

타입: 문자열

허용된 값: on | off

필수 여부: 아니요

--max-concurrent-clients

세션에 연결하도록 허용된 최대 NICE DCV 클라이언트 수를 지정합니다. 생략할 경우 무제한 연결이 기본 설정됩니다.

유형: 정수

필수 항목 여부: 아니요

--init

이 옵션은 Linux NICE DCV 서버의 가상 세션에서만 지원됩니다. 사용자 지정 init 스크립트 경로를 지정합니다. 스크립트를 사용하여 세션을 시작할 때 특정 데스크톱 환경을 시작하고 특정 애플리케이션을 자동으로 시작할 수 있습니다. 스크립트는 실행 가능해야 합니다. 생략할 경우 기본 데스크톱 환경을 시작하는 스크립트로 기본 설정됩니다.

타입: 문자열

필수 항목 여부: 아니요

session ID

명령 종료 시 세션의 ID를 제공합니다.

타입: 문자열

필수 항목 여부: 예

예

예제 1 - 콘솔 세션

다음 명령은 소유자가 `dcv-user`이고 고유 세션 ID가 `my-session`이며 이름이 `my graphics session`인 세션을 생성합니다. 또한 이 명령은 `perm-file.txt`라는 권한 파일을 지정합니다.

- Windows NICE DCV 서버

```
C:\> dcv create-session^
  --owner dcv-user^
  --name "my graphics session"^
  --permissions-file perm-file.txt^
my-session
```

- Linux NICE DCV 서버

```
$ sudo dcv create-session \
  --type=console \
  --owner dcv-user \
  --name "my graphics session" \
  --permissions-file perm-file.txt \
my-session
```

예제 2 - 가상 세션(Linux NICE DCV 서버만 해당)

다음 명령은 루트 사용자를 통해 가상 세션을 생성하여 의도된 세션 소유자인 `dcv-user`를 가장합니다. 루트 사용자가 생성한 세션일지라도 소유자는 `dcv-user`입니다.

```
$ sudo dcv create-session \
  --owner dcv-user \
  --user dcv-user \
my-session
```

예제 3 - 가상 세션(Linux NICE DCV 서버만 해당)

다음 명령은 세션을 생성한 사용자가 소유하는 가상 세션을 생성합니다.

```
$ dcv create-session my-session
```

자동 콘솔 세션 활성화

자동 콘솔 세션을 활성화하면 NICE DCV 서버를 시작할 때마다 콘솔 세션이 자동으로 생성됩니다. 자동 콘솔 세션은 owner 구성 파라미터로 지정된 NICE DCV 사용자가 소유합니다. 세션 ID는 항상 console입니다.

자동 콘솔 세션에 영향을 미치는 다른 파라미터로는 max-concurrent-clients, permissions-file, 및 storage-root가 있습니다. 이런 파라미터에 대한 자세한 내용은 [session-management/automatic-console-session 파라미터](#) 섹션을 참조하세요.

Note

NICE DCV는 자동 가상 세션을 지원하지 않습니다.

Windows NICE DCV server

Windows NICE DCV 서버에서 자동 콘솔 세션을 활성화하려면 다음과 같이 하세요.

1. Windows 레지스트리 편집기를 엽니다.
2. HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/session-management/ 키로 이동합니다.
3. create-session 파라미터를 생성합니다.
 - a. 탐색 창에서 session-management 키의 컨텍스트(마우스 오른쪽 클릭) 메뉴를 열고 새로 만들기, DWORD(32비트) 값을 선택합니다.
 - b. 이름에 create-session을 입력하고 Enter 키를 누릅니다.
 - c. create-session 파라미터를 엽니다. Value data(값 데이터)에 1을 입력하고 확인을 선택합니다.
4. automatic-console-sessionHKEY_USERS/S-1-5-18/소프트웨어/GSettings/com/NiceSoftware/DCV/세션 관리/ 키로 이동합니다.
5. owner 파라미터를 생성합니다.

- a. 탐색 창에서 해당 키의 컨텍스트 메뉴 (마우스 오른쪽 버튼 클릭) 를 열고 새로 만들기, 문자열 값을 선택합니다. `automatic-console-session`
 - b. 이름에 `owner`을 입력하고 Enter 키를 누릅니다.
 - c. 소유자 파라미터를 엽니다. Value data(값 데이터)에 세션 소유자의 이름을 입력하고 확인을 선택합니다.
6. 확인을 선택하고 Windows 레지스트리 편집기를 닫습니다.
 7. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

Linux NICE DCV server

Linux NICE DCV 서버에서 자동 콘솔 세션을 활성화하려면 다음과 같이 하세요.

1. `/etc/dcv/`로 이동하고 원하는 텍스트 편집기에서 `dcv.conf`를 엽니다.
2. 다음 형식을 사용하여 `create-session` 섹션에 `owner` 및 `[session-management/automatic-console-session]` 파라미터를 추가합니다.

```
[session-management]
create-session = true

[session-management/automatic-console-session]
owner="session-owner"
```

3. 파일을 저장하고 닫습니다.
4. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

NICE DCV 세션 중지

콘솔 세션은 Windows NICE DCV 서버의 관리자와 Linux NICE DCV 서버의 루트 사용자만 중지할 수 있습니다. Linux NICE DCV 서버의 가상 세션은 루트 사용자 또는 해당 세션을 만든 NICE DCV 사용자만 중지할 수 있습니다.

Note

세션을 중지하면 세션에서 실행 중인 모든 애플리케이션이 닫힙니다.

Windows 또는 Linux NICE DCV 서버에서 콘솔 세션 또는 가상 세션을 중지하려면 `dcv close-session` 명령을 사용하고 고유 세션 ID를 지정합니다.

주제

- [명령문](#)
- [예](#)

명령문

```
dcv close-session session-id
```

예

예를 들어 다음 명령은 고유 ID가 `my-session`인 세션을 중지합니다.

```
dcv close-session my-session
```

실행 중인 NICE DCV 세션 관리

다음 섹션에서는 실행 중인 NICE DCV 세션을 관리하는 방법에 대해 설명합니다.

주제

- [NICE DCV 세션 스토리지 관리](#)
- [NICE DCV 세션 권한 부여 관리](#)
- [NICE DCV 세션 표시 레이아웃 관리](#)
- [세션 이름 관리](#)

NICE DCV 세션 스토리지 관리

세션 스토리지는 클라이언트가 NICE DCV 세션에 연결되어 있을 때 액세스할 수 있는 NICE DCV 서버의 디렉터리입니다.

NICE DCV 서버에서 세션 스토리지가 활성화된 경우 `dcv set-storage-root` 명령을 사용하여 세션 스토리지에 사용할 디렉터리를 지정할 수 있습니다. NICE DCV 서버에서 세션 스토리지를 활성화하는 방법에 대한 자세한 내용은 [세션 스토리지 활성화](#) 섹션을 참조하세요.

세션 스토리지 경로를 설정하려면 `dcv set-storage-root` 명령을 사용하여 세션 ID와 사용할 디렉터리의 경로를 지정합니다.

주제

- [명령문](#)
- [옵션](#)
- [예](#)

명령문

```
dcv set-storage-root --session session_id /path_to/directory
```

디렉터리 경로의 경우 `%home%`을 사용하여 현재 로그인되어 있는 사용자의 홈 디렉터리를 지정할 수 있습니다. 예를 들어, `%home%/storage/` 경로는 Windows 서버에서 `c:\Users\username\storage\`로 확인됩니다. Linux 서버에서는 `$HOME/storage/`로 확인됩니다.

옵션

`dcv set-storage-root` 명령과 함께 다음 옵션을 사용할 수 있습니다.

--session

스토리지 디렉터를 지정하는 데 사용할 세션 ID입니다.

타입: 문자열

필수 항목 여부: 예

예

Windows NICE DCV 서버 예제

다음 예제에서는 세션 ID가 `my-session`인 세션의 스토리지 경로로 `c:\session-storage`를 설정합니다.

```
C:\> dcv set-storage-root --session my-session c:\session-storage
```

Linux NICE DCV 서버 예제

다음 예제에서는 세션 ID가 `my-session`인 세션에 대해 스토리지 경로를 현재 사용자의 홈 디렉터리의 `session-storage`라는 디렉터리로 설정합니다.

```
$ dcv set-storage-root --session my-session %home%/session-storage/
```

NICE DCV 세션 권한 부여 관리

권한 부여는 NICE DCV 클라이언트에 특정 NICE DCV 기능에 대한 권한을 부여하거나 거부하는 데 사용됩니다. 권한 부여는 일반적으로 NICE DCV 세션이 시작될 때 구성됩니다. 그러나 실행 중인 세션에 대한 권한을 편집할 수 있습니다. NICE DCV 권한 부여에 대한 자세한 내용은 [NICE DCV 권한 부여 구성](#) 섹션을 참조하세요.

실행 중인 세션에 대한 권한을 수정하려면 `dcv set-permissions` 명령을 사용하세요.

주제

- [명령문](#)
- [옵션](#)
- [예](#)

명령문

```
dcv set-permissions --session session-id --none | --reset-builtin | --file /path_to/permissions_file
```

`--none`, `--reset-builtin` 또는 `--file`을 지정해야 합니다.

옵션

`dcv set-permissions` 명령과 함께 다음 옵션을 사용할 수 있습니다.

`--session`

권한을 설정할 세션의 ID를 지정합니다.

`--reset-builtin`

세션의 권한을 기본 세션 권한으로 재설정합니다. 기본 권한은 세션 소유자에게만 모든 기능에 대한 전체 액세스 권한을 부여합니다.

--none

세션에 대한 모든 권한을 취소합니다.

--file

사용자 지정 권한 파일 경로를 지정합니다. 지정한 파일이 비어 있으면 모든 권한이 취소됩니다. 사용자 지정 권한 파일 작성에 대한 자세한 내용은 [권한 파일 작업](#) 단원을 참조하십시오.

예**예제 1 - 모든 권한 취소**

다음 예제에서는 ID가 `my-session`인 세션에 대한 모든 클라이언트 권한을 취소합니다.

```
C:\> dcv set-permissions --session my-session --none
```

예제 2 - 사용자 지정 권한 지정

다음 예제에서는 ID가 `my-session`인 세션에 대해 `perm-file.txt`라는 사용자 지정 권한 파일을 지정합니다. 이 파일은 `c:\dcv\` 디렉터리에 위치합니다.

```
C:\> dcv set-permissions --session my-session --file c:\dcv\perm-file.txt
```

예제 3 - 권한 재설정

다음 예제에서는 ID가 `my-session`인 세션의 권한을 기본값으로 재설정합니다.

```
C:\> dcv set-permissions --session my-session --reset-builtin
```

NICE DCV 세션 표시 레이아웃 관리

NICE DCV 세션 실행에 대한 표시 레이아웃을 설정할 수 있습니다. 표시 레이아웃은 클라이언트가 세션에 연결할 때 사용되는 기본 구성을 지정합니다. 그러나 클라이언트는 NICE DCV 클라이언트 설정 또는 기본 운영 체제 표시 설정을 사용하여 레이아웃을 수동으로 재정의할 수 있습니다.

호스트 서버의 하드웨어 및 소프트웨어 구성이 지정된 해상도 또는 화면 수를 지원하지 않는 경우 NICE DCV 서버는 지정된 표시 레이아웃을 적용하지 않습니다.

NICE DCV는 설정 및 서버 시스템 구성에 따라 해상도를 구성할 수 있습니다.

- 웹 클라이언트 해상도는 기본적으로 1920x1080 (-resolution 서버 설정 기준) 으로 제한됩니다. `web-client-max-head`
- 네이티브 클라이언트는 기본적으로 4096x2160 (from) 으로 제한됩니다. `max-head-resolution`

사용 가능한 해상도와 모니터 수는 서버 구성에 따라 다르므로 [사전 요구 사항 가이드](#)에 따라 시스템 환경과 드라이버를 올바르게 설정하여 성능을 최적화하십시오.

Note

네이티브 클라이언트의 경우 최대 4대의 모니터를 사용할 수 있습니다.
 웹 클라이언트의 경우 최대 2대의 모니터를 사용할 수 있습니다.
 어떤 구성에서도 해상도가 높거나 최대 모니터 수를 초과하는 것은 지원되지 않습니다.

주제

- [표시 레이아웃 제한](#)
- [표시 레이아웃 지정](#)
- [표시 레이아웃 보기](#)

표시 레이아웃 제한

클라이언트가 지정된 범위를 벗어나는 표시 레이아웃을 요청하지 않도록 NICE DCV 서버를 구성할 수 있습니다. 표시 레이아웃 변경을 제한하려면 다음 NICE DCV 서버 파라미터를 구성합니다.

- [enable-client-resize](#) - 클라이언트가 표시 레이아웃을 변경하지 못하도록 하려면 이 파라미터를 `false`로 설정합니다.
- [min-head-resolution](#) 및 [max-head-resolution](#) - 허용되는 최소 해상도와 최대 해상도를 각각 지정합니다.
- [web-client-max-head-resolution](#) - 웹 브라우저 클라이언트에 허용되는 최대 해상도를 지정합니다. `max-head-resolution` 제한은 `web-client-max-head-resolution` 제한에 추가로 적용됩니다. 기본적으로 웹 브라우저 클라이언트의 최대 해상도는 1920x1080입니다. 더 높은 해상도를 지정하면 웹 브라우저와 클라이언트 컴퓨터의 사양에 따라 성능 문제가 발생할 수 있습니다.
- [max-num-heads](#) - 최대 디스플레이 수를 지정합니다.
- [max-layout-area](#) - 화면 영역에 허용되는 최대 픽셀 수를 지정합니다. 픽셀 단위로 표현된 총 화면 영역이 지정된 값을 초과하는 요청은 무시됩니다.

이러한 파라미터에 대한 자세한 내용은 파라미터 참조의 [display 파라미터](#) 단원을 참고하십시오.

표시 레이아웃 지정

실행 중인 NICE DCV 세션의 표시 레이아웃을 구성하려면 다음과 같이 하세요.

`dcv set-display-layout` 명령을 사용하여 표시 레이아웃 및 표시 레이아웃 설명자를 설정할 세션을 지정합니다.

```
dcv set-display-layout --session session-id display-layout-descriptor
```

표시 레이아웃 설명자는 디스플레이의 수와 각 디스플레이의 해상도 및 위치 오프셋을 지정합니다. 설명은 다음 형식으로 지정해야 합니다.

```
widthxheight+|-x-position-offset+|-y-position-offset
```

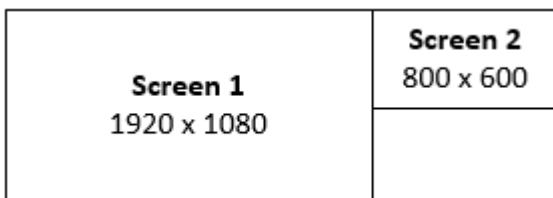
화면을 두 개 이상 지정하는 경우 화면 설명자를 쉼표로 구분합니다. 화면 위치 오프셋은 화면 1을 기준으로 화면의 왼쪽 위 모서리의 위치를 지정합니다. 스크린에 대한 위치 오프셋을 지정하지 않으면 기본 값은 $x=0, y=0$ 입니다.

⚠ Important

둘 이상의 스크린을 지정하는 경우 화면 겹침이 발생하지 않도록 각 스크린의 위치 오프셋을 올바르게 설정해야 합니다.

예를 들어, 다음 표시 레이아웃 설명자는 두 개의 스크린을 지정합니다.

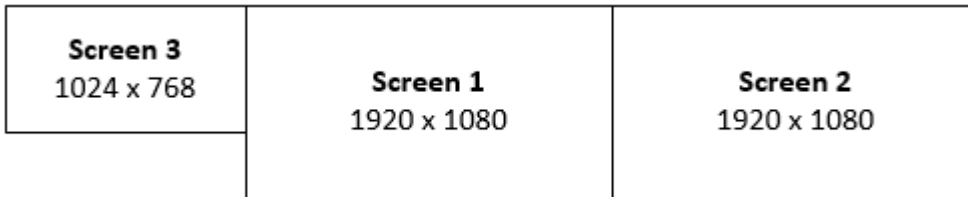
- 화면 1: 1920x1080 해상도 오프셋을 $x=0, y=0$ 으로 설정합니다.
- 화면 2: 화면 1의 오른쪽에 나타나도록 800x600 해상도 오프셋을 $x=1920, y=0$ 으로 설정합니다.



```
1920x1080+0+0,800x600+1920+0
```

다음 표시 레이아웃 기술자는 세 개의 화면을 지정합니다.

- 화면 1: 1920x1080 해상도 오프셋을 $x=0, y=0$ 으로 설정합니다.
- 화면 2: 화면 1의 오른쪽에 나타나도록 1920x1080 해상도 오프셋을 $x=1920, y=0$ 으로 설정합니다.
- 화면 3: 화면 1의 왼쪽에 나타나도록 1024x768 해상도 오프셋을 $x=-1024, y=0$ 으로 설정합니다.



```
1920x1080+0+0, 1920x1080+1920+0, 1024x768-1024+0
```

표시 레이아웃 보기

세션에 대한 표시 레이아웃을 보려면

`dcv describe-session` 명령을 사용하여 출력의 `display layout` 요소를 검토합니다. 자세한 설명은 [NICE DCV 세션 보기](#) 섹션을 참조하세요.

세션 이름 관리

실행 중인 세션의 이름은 언제든지 변경이 가능합니다. 세션의 특정 이름을 사용하여 해당 이름을 기반으로 세션을 빠르게 식별할 수 있습니다. 세션 이름은 실행 중인 세션에서 고유하지 않아도 됩니다.

실행 중인 세션의 이름을 변경하려면 `dcv set-name` 명령을 사용합니다.

주제

- [명령문](#)
- [옵션](#)
- [예](#)

명령문

```
$ dcv set-name --session session_id --none | --name "session-name"
```

--name 또는 --none를 지정해야 합니다.

옵션

dset-name 명령과 함께 다음 옵션을 사용할 수 있습니다.

--session

이름을 설정할 세션의 ID입니다.

타입: 문자열

필수 항목 여부: 예

--name

세션에 할당할 이름입니다. 세션에 이름을 할당하려는 경우에만 이 옵션을 지정하세요. 이름을 제거하려면 이 파라미터를 생략하세요. 이름은 최대 256자까지 지정할 수 있습니다. 이름에는 문자, 숫자 및 특수 문자를 포함할 수 있습니다. 지정된 문자열이 256자를 초과하면 명령이 실패합니다.

타입: 문자열

필수 항목 여부: 아니요

--none

세션에서 기존 이름을 제거하려면 이 파라미터를 지정하세요. 세션 이름을 제거하지 않으려면 이 옵션을 생략합니다.

필수 여부: 아니요

예

예제 1 - 세션 이름 변경

다음 예제에서는 ID가 my-session인 세션의 이름을 my graphics session으로 설정합니다.

```
$ dcv set-name --session my-session --name "my graphics sessions"
```

예제 2 - 세션 이름 제거

다음 예제에서는 ID가 my-session인 세션의 이름을 제거합니다.

```
$ dcv set-name --session my-session --none
```

세션 시간대 관리

DCV를 사용하면 세션 소유자와 사용자가 DCV 서버의 위치 또는 현재 위치를 반영하도록 세션의 시간대를 설정할 수 있습니다.

시간대 리디렉션 활성화

특정 세션의 모든 사용자에게 이 기능을 활성화하거나 비활성화할 수 있습니다.

1. [enable-timezone-redirection](#) 파라미터를 다음 값 중 하나로 설정합니다.

- `always-on`: 시간대 리디렉션이 항상 활성화됩니다.

이 기능이 켜지고 세션에 클라이언트의 시간대 정보가 표시됩니다. 사용자는 이 기능을 끌 수 없습니다.

- `always-off`: 시간대 리디렉션이 항상 비활성화됩니다.

이 기능이 꺼지고 세션에 자체 시간대 정보가 표시됩니다. 사용자는 이 기능을 켤 수 없습니다.

- `client-decides`: 시간대 리디렉션이 기본적으로 켜져 있습니다.

세션에서 기능이 활성화되고 클라이언트 시간대가 표시되며 사용자는 이 기능을 사용하지 않도록 설정하여 서버 시간대가 표시되도록 할 수 있습니다.

Note

이 설정이 표준 기본 설정입니다.

Note

세션의 개별 사용자만 이 기능을 사용해야 하는 경우, 먼저 모든 사용자에게 대한 중앙 집중식 파라미터를 설정한 다음 [권한 추가](#)에서 사용자 지정 권한 파일을 만들어 개인의 권한을 개별적으로 조정해야 합니다.

2. 변경 사항이 나타나도록, 영향을 받는 세션을 모두 다시 시작하세요.

NICE DCV 세션 보기

Windows NICE DCV 서버의 관리자 또는 Linux NICE DCV 서버의 루트 사용자는 서버에서 실행 중인 모든 활성 세션을 볼 수 있습니다. NICE DCV 사용자는 자신이 생성한 세션만 볼 수 있습니다.

주제

- [모든 활성 세션 나열](#)
- [특정 세션에 대한 정보 보기](#)

모든 활성 세션 나열

Windows 또는 Linux NICE DCV 서버에서 활성 콘솔 세션 또는 가상 세션을 나열하려면 `dcv list-sessions` 명령을 사용합니다.

주제

- [명령문](#)
- [출력](#)

명령문

```
dcv list-sessions
```

출력

이 명령은 다음 형식으로 활성 세션 목록을 반환합니다.

```
Session: session-id (owner:session-owner type:virtual/console name:'my session')
```

특정 세션에 대한 정보 보기

세션에 대한 정보를 보려면 `dcv describe-session` 명령을 사용하고 고유한 세션 ID를 지정합니다.

주제

- [명령문](#)
- [출력](#)

명령문

```
$ dcv describe-session session-id
```

출력

다음 예제 출력에서 `display-layout` 요소는 세션의 표시 레이아웃이 두 개의 800x600 화면을 사용하도록 설정되어 있음을 나타냅니다. 이 중 두 번째 화면은 첫 번째 화면의 `x=800`(오른쪽)으로 오프셋됩니다.

```
Session: test
  owner: session-id
  name: session-name
  x display: :1
  x authority: /run/user/1009/dcv/test.xauth
  display layout: 800x600+0+0,800x600+800+0
```

`--json`(또는 `-j`) 옵션을 포함하여 명령이 출력을 JSON 형식으로 반환하도록 할 수도 있습니다. JSON 출력은 세션에 대한 추가 세부 정보를 제공합니다.

```
$ dcv describe-session session-id --json
```

다음은 JSON 출력의 예입니다.

```
{
  "id" : "session-id",
  "owner" : "dcvuser",
  "name" : "session-name",
  "num-of-connections" : 0,
  "creation-time" : "2020-03-02T16:08:50Z",
  "last-disconnection-time" : "",
  "licenses" : [
    {
      "product" : "dcv",
      "status" : "licensed",
      "check-timestamp" : "2020-03-02T16:08:50Z",
      "expiration-date" : "2020-03-29T00:00:00Z"
    },
    {
      "product" : "dcv-gl",
```

```

    "status" : "licensed",
    "check-timestamp" : "2020-03-02T16:08:50Z",
    "expiration-date" : "2020-03-29T00:00:00Z"
  }
],
"storage-root" : "",
"type" : "virtual",
"x11-display" : ":2",
"x11-authority" : "/run/user/1009/dcv/vsession.xauth",
"display-layout" : [
  {
    "width" : 800,
    "height" : 600,
    "x" : 0,
    "y" : 0
  },
  {
    "width" : 800,
    "height" : 600,
    "x" : 800,
    "y" : 0
  }
]
}

```

NICE DCV 세션 스크린샷 찍기

dcv get-screenshot 명령을 사용하여 실행 중인 세션의 데스크톱 스크린샷을 찍을 수 있습니다.

명령문

```
dcv get-screenshot --max-width pixels --max-height pixels --format JPEG/PNG --primary
--json --output /path_to/destination session_name
```

옵션

--max-width

스크린샷의 최대 너비(픽셀 단위)를 지정합니다. 너비나 높이를 지정하지 않는 경우 스크린샷은 세션의 디스플레이 해상도를 사용합니다. 높이만 지정하는 경우 가로 세로 비율을 유지하기 위해 너비가 자동으로 조정됩니다.

유형: 정수

필수 항목 여부: 아니요

--max-height

스크린샷의 최대 높이(픽셀 단위)를 지정합니다. 너비나 높이를 지정하지 않는 경우 스크린샷은 세션의 디스플레이 해상도를 사용합니다. 너비만 지정하는 경우 가로 세로 비율을 유지하기 위해 높이가 자동으로 조정됩니다.

유형: 정수

필수 항목 여부: 아니요

--format

스크린샷의 파일 형식입니다. 현재 JPEG 및 PNG 형식만 지원됩니다. --format 및 --output 옵션에 대해 충돌하는 파일 유형을 지정하는 경우 --format에 지정된 값이 우선 적용됩니다. 예를 들어, --format JPEG 및 --output myfile.png를 지정하면 NICE DCV는 JPEG 이미지 파일을 생성합니다.

타입: 문자열

허용된 값: JPEG | PNG

필수 여부: 아니요

--primary

기본 디스플레이의 스크린샷만 찍을지 여부를 나타냅니다. 기본 디스플레이의 스크린샷만 찍으려면 --primary를 지정하세요. 모든 디스플레이의 스크린샷을 찍으려면 이 옵션을 생략합니다. 모든 디스플레이의 스크린샷을 찍도록 선택하면 모든 디스플레이가 단일 스크린샷으로 결합됩니다.

필수 여부: 아니요

--json, -j

출력을 base64로 인코딩된 JSON 형식으로 전달할지 여부를 나타냅니다. JSON 출력을 가져오려면 --json을 지정하세요. 그렇지 않으면 생략합니다.

필수 여부: 아니요

--output, -o

스크린샷의 대상 경로, 파일 이름, 파일 유형을 지정합니다. 예를 들어 Windows의 경우 c:\directory\filename.format을 지정하고, Linux의 경우 /directory/

filename.format을 지정합니다. 형식은 .png 또는 .jpeg여야 합니다. --format 및 --output 옵션에 대해 충돌하는 파일 유형을 지정하는 경우 --format에 지정된 값이 우선 적용됩니다. 예를 들어, --format JPEG 및 --output myfile.png를 지정하면 NICE DCV는 JPEG 이미지 파일을 생성합니다.

타입: 문자열

필수 항목 여부: 아니요

예

예 1

다음 예제 명령은 이름이 my-session으로 지정된 세션의 스크린샷을 찍습니다. 스크린샷은 서버의 해상도를 사용합니다.

```
dcv get-screenshot --output myscreenshot.png my-session
```

예제 2

다음 예제 명령은 너비 200 x 높이 100 픽셀인 스크린샷을 찍습니다. 이름이 my-session으로 지정된 세션의 스크린샷을 찍습니다. 스크린샷을 현재 디렉터리에 myscreenshot.png 파일 이름으로 저장합니다.

```
dcv get-screenshot --max-width 200 --max-height 100 --output myscreenshot.png my-session
```

예 3

다음 예제 명령은 이름이 my-session으로 지정된 세션의 스크린샷을 찍습니다. 스크린샷은 기본 디스플레이에만 해당됩니다. 파일을 현재 디렉터리에 저장하고 스크린샷 이름을 myscreenshot.png로 지정합니다.

```
dcv get-screenshot --primary --output myscreenshot.jpeg my-session
```

예 4

다음 예제 명령은 이름이 my-session으로 지정된 세션의 스크린샷을 찍습니다. 이 명령은 base64와 JSON 형식으로 인코딩된 파일을 출력합니다.

```
dcv get-screenshot --json --format png my-session
```

방법...

주제

- [외부 인증 사용](#)
- [유휴 세션 찾기 및 중지](#)
- [X 서버로의 원격 X 연결 활성화](#)
- [NICE DCV 웹 브라우저 클라이언트를 iFrame 안에 임베드](#)

외부 인증 사용

기본적으로 NICE DCV 클라이언트 인증이 기본 운영 체제에 위임됩니다. Windows NICE DCV 서버에서는 인증이 위임됩니다. WinLogon Linux NICE DCV 서버의 경우 인증이 Linux PAM에 위임됩니다.

외부 인증 서버를 사용하여 클라이언트를 인증하도록 NICE DCV를 구성할 수 있습니다. 이렇게 하면 기존 인증 시스템을 사용할 수 있습니다. NICE DCV는 외부 인증을 통해 기존 로그인 메커니즘을 활용하여 외부 인증 서버에 인증을 위임합니다.

외부 인증은 DCV 서버 액세스 권한이 있는 사용자를 인증하여 세션 생성을 사용할 수 있도록 합니다. 자체 외부 인증자를 직접 설정하지 않는 한 시스템 인증처럼 기본 OS에 대해 사용자를 인증하지 않습니다.

[DCV Session Manager](#)에는 외부 인증자가 내장되어 있습니다. 이 기능을 사용하려면 DCV 서버에서 세션 관리자 주소를 사용하여 [auth-token-verifier](#) 매개 변수를 설정해야 합니다.

외부 인증자 서버를 사용하려면 다음을 갖추고 있어야 합니다.

- 로그인 메커니즘 - 사용자가 로그인할 때 사용하는 프론트엔드 메커니즘입니다. 기존의 보안 인증 확인 시스템을 사용하여 사용자를 확인할 수 있어야 하며 토큰을 생성하여 NICE DCV 서버에 제공할 수 있어야 합니다. 자세한 설명은 [토큰 사용](#) 섹션을 참조하세요.
- 인증 서버 - 로그인 메커니즘으로 생성된 토큰을 인증하는 서버입니다. 이 서버는 토큰이 포함된 NICE DCV 서버로부터 HTTP(S) POST 요청을 수신하고 필요한 인증을 수행한 다음, NICE DCV 서버로 응답을 다시 보낼 수 있어야 합니다. 인증 서버를 구현하는 방법에 대한 자세한 내용은 [인증 서비스 요구 사항](#) 섹션을 참조하세요.
- NICE DCV 서버 구성 - NICE DCV 서버는 외부 인증 서버를 사용하도록 구성해야 합니다. 자세한 설명은 [NICE DCV 서버 구성](#) 섹션을 참조하세요.

주제

- [NICE DCV 서버 구성](#)
- [토큰 사용](#)
- [인증 서비스 요구 사항](#)

NICE DCV 서버 구성

외부 인증 서비스를 사용하도록 NICE DCV 서버를 구성해야 합니다.

Linux NICE DCV server

Linux에서 외부 인증 서버를 지정하려면 다음과 같이 하세요.

1. /etc/dcv/로 이동하고 원하는 텍스트 편집기에서 dcv.conf를 엽니다.
2. [security] 섹션에서 auth-token-verifier 파라미터를 찾고 기존 값을 외부 인증 서버의 URL과 통신에 사용되는 포트로 바꿉니다. 이때 형식은 *url:port*와 같습니다. 예를 들어 DcvSimpleExternalAuthenticator 사용하는 경우 다음을 지정하십시오.
http://127.0.0.1:8444

auth-token-verifier 섹션에 [security] 파라미터가 없는 경우, 다음 형식을 사용하여 수동으로 추가합니다.

```
[security] auth-token-verifier=url:port
```

3. 파일을 저장하고 닫습니다.

Windows NICE DCV server

Windows에서 외부 인증 서버를 지정하려면 다음과 같이 하세요.

1. Windows 레지스트리 편집기를 엽니다.
2. HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/ 키로 이동합니다.
3. [보안 auth-token-verifier 매개변수에서 매개변수를](#) 찾습니다.
4. 다음 중 하나를 수행합니다.
 - 값 데이터의 경우 외부 인증 서버의 URL과 통신할 포트를 *url:port* 형식으로 입력합니다.

Example

예를 들어 를 DcvSimpleExternalAuthenticator 사용하는 경우 다음을 지정하십시오
 오 `http://127.0.0.1:8444`.

- 보안 섹션에 auth-token-verifier 매개변수가 없는 경우 매개 변수를 에 추가하십시오 PowerShell. [구성 파라미터 수정](#)을 참조하세요.

5. Windows 레지스트리 편집기를 닫습니다.
6. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

토큰 사용

토큰을 생성한 후에는 NICE DCV 서버로 보낼 수 있어야 합니다. 웹 브라우저 클라이언트를 사용하여 다음과 같이 연결 URL에 토큰을 추가합니다.

```
https://server_hostname_or_IP:port?authToken=token#session_id
```

예:

```
https://my-dcv-server.com:8443/?authToken=1234567890abcdef#my-session
```

인증 서비스 요구 사항

사용자 지정 인증 서비스는 NICE DCV 서버의 동일한 호스트에서 실행하거나 별도의 호스트에서 실행할 수 있습니다. 인증 서비스는 NICE DCV 서버의 HTTP(S) POST 요청을 수신해야 합니다.

다음은 NICE DCV 서버에서 사용하는 POST 요청 형식을 보여줍니다.

```
POST / HTTP/1.1
Content-Type: application/x-www-form-urlencoded
sessionId=session_id&authenticationToken=token&clientAddress=client_address
```

인증 서비스는 제공된 토큰이 유효한지 여부를 판단합니다.

토큰의 유효성을 검사한 후 인증 서버는 NICE DCV 서버에 응답을 반환해야 합니다. 응답 본문에는 인증 프로세스의 결과에 따라 다음 중 하나가 반드시 포함되어야 합니다.

- 인증에 성공하면 인증 서비스는 결과 `yes` 및 사용자 식별자를 반환합니다. 예:

```
<auth result="yes"><username>username</username></auth>
```

- 인증에 실패하면 인증 서비스는 결과 no를 반환합니다. 예:

```
<auth result="no"><message>message</message></auth>
```

DcvSimpleExternalAuthenticator

NICE DCV는 DcvSimpleExternalAuthenticator라는 참조 외부 인증 서버와 함께 제공됩니다. DcvSimpleExternalAuthenticator 는 사용자 지정 인증 서버를 만들기 위한 시작점으로 사용할 수 있는 단일 Python 스크립트입니다.

DcvSimpleExternalAuthenticator 서버는 HTTP 및 HTTPS를 지원하며 NICE DCV 서버가 설치된 동일한 서버에서 실행되어야 합니다. 기본적으로 는 포트에서 요청을 DcvSimpleExternalAuthenticator 수신합니다. 8444 필요한 경우 포트를 변경할 수 있습니다. 이렇게 하려면 원하는 텍스트 편집기로 /etc/dcv/simpleextauth.conf를 열고 EXTAUTH_PORT 파라미터를 찾은 다음 기존 값을 필요한 포트 번호로 바꾸세요.

사용하려면 DcvSimpleExternalAuthenticator nice-dcv-simple-external-authenticator 패키지를 설치해야 합니다. 자세한 설명은 [NICE DCV 서버 설치](#) 섹션을 참조하세요.

단순 외부 인증자 사용

1. 인증 디렉터리로 이동합니다.

```
sudo mkdir -p /var/run/dcvsimpleextauth
```

2. 인증 토큰을 생성합니다.

Example

이 예제에서 123456은 샘플 인증자 토큰, session-123은 샘플 세션 ID, 는 username은 사용자입니다.

```
echo "123456" | sudo dcvsimpleextauth add-user --session session-123 --auth-dir /var/run/dcvsimpleextauth/ --user username -append
```

3. 서버를 시작합니다.

```
sudo dcvsimpleextauth --port 8444 --auth-dir /var/run/dcvsimpleextauth/
start-server
```

4. 서버가 실행되면 구성을 테스트하여 유효성을 검사합니다.

Example

다시 한 번 이 예제를 사용하면 다음과 같이 테스트가 실행됩니다.

```
curl -k http://localhost:8444 -d sessionId=session-123 -d
authenticationToken=123456
```

테스트에 성공하면 인증 결과로 `yes`를 받게 됩니다.

유휴 세션 찾기 및 중지

`-j` 명령 옵션과 함께 `dcv describe-sessions` CLI 명령을 사용하여 유휴 NICE DCV 세션을 식별할 수 있습니다. `-j` 옵션을 지정하면 명령이 JSON 형식으로 출력을 반환하도록 구성되며, 세션에 대한 추가 세부 정보를 제공합니다.

예를 들어 다음 명령은 `my-session`이라는 세션에 대한 정보를 반환합니다.

```
$ dcv describe-session my-session -j
```

출력:

```
{
  "id" : "my-session",
  "owner" : "dcvuser",
  "x11-display" : ":1",
  "x11-authority" : "/run/user/1009/dcv/test3.xauth",
  "num-of-connections" : 1,
  "creation-time" : "2019-05-13T13:21:19.262883Z",
  "last-disconnection-time" : "2019-05-14T12:32:14.357567Z",
  "licensing-mode" : "DEMO",
  "licenses" : [
    {
      "product" : "dcv",
      "status" : "LICENSED",
      "check-timestamp" : "2019-05-14T12:35:40Z",
```



```

    "expiration-date" : "2019-05-29T00:00:00Z"
  },
  {
    "product" : "dcv-gl",
    "status" : "LICENSED",
    "check-timestamp" : "2019-05-14T12:35:40Z",
    "expiration-date" : "2019-05-29T00:00:00Z"
  }
]
}

```

명령 출력에서 `num-of-connections` 파라미터는 활성 클라이언트 연결 수를 나타냅니다. 0 값은 활성 클라이언트 연결이 없고 세션이 현재 유휴 상태임을 나타냅니다. `last-disconnection-time` 파라미터를 사용하여 세션에 마지막으로 활성 클라이언트 연결이 있었던 시점을 확인할 수도 있습니다.

이 정보를 사용하여 유휴 세션을 식별하는 스크립트 또는 cron 작업을 만들 수 있습니다. 그런 다음 [dcv close-session](#) 명령을 사용하여 사용을 중지할 수 있습니다.

Note

세션을 중지하면 세션에서 실행 중인 모든 애플리케이션이 닫힙니다.

X 서버로의 원격 X 연결 활성화

기본적으로, Xdcv는 내재된 보안 위험 때문에 X 전송을 사용하지 못하도록 합니다. NICE DCV는 최신 버전의 Xorg 서버에서 이 동작을 상속합니다. NICE DCV 서버는 다음의 기본 완화 작업을 실행하여 보안 위험을 최소화합니다.

- X 서버는 네트워크로부터의 X 연결을 방지합니다. X 서버는 `-nolisten tcp` 명령줄 옵션으로 시작하도록 구성되어 있습니다. 그러나 기본 동작을 변경하여 X 서버로의 원격 X 연결을 활성화할 수는 없습니다. 차선책에 대한 자세한 내용은 [X 서버로의 원격 X 연결 활성화](#)를 참조하세요.
- X 서버는 GLX 간접 컨텍스트를 사용하지 않습니다. DCV-GL과의 충돌 때문에 현재 GLX 간접 컨텍스트를 활성화하는 차선책은 없습니다.

보안 위험 및 완화 방법에 대한 자세한 내용은 [X.Org 보안 공지](#)를 참조하세요.

X 서버로의 원격 X 연결 활성화

기본적으로 Xdcv는 `-nolisten tcp` 명령줄 옵션으로 시작하여 보안 위험에 대한 노출을 줄이도록 구성되어 있습니다. 그러나 기본 동작을 변경하여 X 전송을 활성화할 수 있습니다.

X 전송을 활성화하려면

원하는 텍스트 편집기에서 `/etc/dcv/dcv.conf` 파일을 엽니다. 파일 끝에 다음을 추가합니다.

- IPv4 및 IPv6 X 전송을 활성화하려면

```
[session-management]
virtual-session-xdcv-args="-listen tcp"
```

- IPv4 전용 X 전송을 활성화하려면

```
[session-management]
virtual-session-xdcv-args="-listen tcp -nolisten inet6"
```

Note

X 전송을 활성화해도 기존 세션에는 영향을 주지 않으며, 활성화된 후 시작된 새 세션에만 영향을 줍니다.

X 전송을 테스트하려면

1. NICE DCV 세션을 연결합니다.
2. NICE DCV가 6000~6063 범위의 포트에서 수신하고 있는지 확인합니다.

```
$ netstat -punta | grep 600
```

3. NICE DCV 서버 호스트 액세스 목록에 원격 서버를 추가합니다.

```
$ xhost +remote_server
```

4. NICE DCV 세션 표시 번호를 검색합니다.

```
$ dcv describe-session session_name | grep display
```

5. 애플리케이션이 호스팅되는 원격 서버에 대한 SSH 접속입니다.

```
$ ssh user@remote_server
```

6. 원격 서버에서 표시 환경 변수를 내보내어 NICE DCV 세션의 X 서버를 가리킵니다.

```
$ export DISPLAY=dcv_server_ip:display_number
```

7. 원격 서버에서 애플리케이션을 실행하여 X 전송 기능을 테스트합니다. 예:

```
xterm
```

이 경우 xterm인 테스트 애플리케이션은 NICE DCV 서버의 데스크톱 환경에 나타나야 합니다.

NICE DCV 웹 브라우저 클라이언트를 iFrame 안에 임베드

기본적으로 NICE DCV는 클릭재킹 공격을 방어하기 위해 웹 브라우저 클라이언트를 iFrame에 임베드하는 것을 허용하지 않습니다. 하지만 이 기본 동작을 재정의하여 웹 브라우저 클라이언트가 iFrame 내에서 실행되도록 할 수 있습니다.

클릭재킹 공격 방지에 대한 자세한 내용은 [Content Security Policy Cheat Sheet](#)를 참조하세요.

웹 브라우저가 iFrame 내에서 실행되도록 하려면 웹 브라우저 클라이언트에 다음과 같은 추가 HTTP 응답 헤더를 전송하도록 NICE DCV 서버를 구성해야 합니다.

- web-x-frame-options
- web-extra-http-headers

웹 브라우저 간 호환성을 극대화하려면 두 헤더를 모두 추가하는 것이 좋습니다.

Note

NICE DCV 연결 게이트웨이를 통해 연결하는 경우 게이트웨이 구성 내에서 x-프레임 옵션을 정의해야 합니다. 이 작업은 게이트웨이 구성 [\[web-resources\] 섹션](#) 내의 local-resources-http-headers 매개 변수를 사용하여 수행됩니다.

Windows server

1. Windows 레지스트리 편집기를 열고 HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/ 키로 이동합니다.
2. web-x-frame-options 파라미터를 엽니다. 값 데이터에 "ALLOW-FROM https://*server_hostname*"을 입력합니다.

Note

파라미터가 없는 경우 새 String 파라미터를 생성하고 이름을 web-x-frame-options로 지정합니다.

3. web-extra-http-headers 파라미터를 엽니다. 값 데이터에 [("Content-Security-Policy", "frame-ancestors https://*server_hostname*")]을 입력합니다.

Note

파라미터가 없는 경우 새 String 파라미터를 생성하고 이름을 web-extra-http-headers로 지정합니다.

4. Windows 레지스트리 편집기를 닫습니다.
5. NICE DCV 서버를 중지하고 다시 시작합니다.

Linux server

1. 원하는 텍스트 편집기로 /etc/dcv/dcv.conf를 엽니다.
2. [connectivity] 섹션에서 다음을 수행합니다.
 - web-x-frame-options에 "ALLOW-FROM https://*server_hostname*"를 입력합니다.
 - web-extra-http-headers에 [("Content-Security-Policy", "frame-ancestors https://*server_hostname*")]를 입력합니다.

예:

```
[connectivity]
web-x-frame-options="ALLOW-FROM https://my-dcv-server.com"
```

```
web-extra-http-headers=[("Content-Security-Policy", "frame-ancestors https://my-dcv-server.com")]
```

3. 파일을 저장하고 닫습니다.
4. NICE DCV 서버를 [중지](#)하고 [다시 시작](#)합니다.

기본적으로 대부분의 브라우저는 마이크 액세스 및 전체 화면 액세스와 같은 일부 기능에 대한 액세스를 차단합니다. 이러한 기능에 대한 액세스를 허용하려면 웹 페이지의 iFrame 요소를 수정하세요. 예를 들어, 마이크 및 전체 화면 모드에 대한 액세스를 허용하려면 다음과 같이 iFrame 요소를 수정하세요.

```
<iframe src="..." allow="microphone; fullscreen">/iframe>
```

NICE DCV 문제 해결

이 장에서는 NICE DCV에서 발생할 수 있는 문제를 식별하고 해결하는 방법을 설명합니다.

주제

- [로그 파일 사용](#)
- [Linux에서의 가상 세션 생성 문제 해결](#)
- [UID 변경 후 Linux 세션이 시작되지 않음](#)
- [Windows에서의 커서 문제 해결](#)
- [IntelliJ IDEA에 복사 및 붙여넣기 문제 해결](#)
- [자체 서명된 인증서에 대한 리디렉션 설명](#)
- [Windows에서 NVIDIA GPU 사용 시 다중 모니터/전체 화면 오류](#)
- [NICE DCV 성능 및 통계 모니터링](#)

추가 지원이 필요한 경우 다음 리소스를 사용하세요.

- NICE DCV 온프레미스 고객이고 추가 지원이 필요한 경우 NICE DCV 리셀러에게 문의하세요.
- Amazon EC2에서 NICE DCV를 사용하는 경우 [AWS 지원](#)을 통해 지원 티켓을 기록할 수 있습니다.
- AWS 지원 계획이 없는 경우 [AWS re:Post](#)에 질문을 게시하여 NICE DCV 커뮤니티에 도움을 요청할 수 있습니다.

로그 파일 사용

NICE DCV 로그 파일을 사용하여 NICE DCV 서버 문제를 식별하고 해결할 수 있습니다. NICE DCV 로그 파일은 NICE DCV 서버의 다음 위치에서 찾을 수 있습니다.

- Windows 서버

```
C:\ProgramData\NICE\dcv\log\server.log
```

Note

ProgramData 폴더는 기본적으로 숨길 수 있습니다. ProgramData 폴더가 표시되지 않으면 숨겨진 항목을 표시하도록 파일 브라우저를 설정합니다. 또는 주소 표시줄에 %programdata%를 입력하고 Enter 키를 누릅니다.

- Linux 서버

```
/var/log/dcv/server.log
```

NICE DCV 서버를 통해 로그 파일의 세부 수준을 구성할 수 있습니다. 다음 세부 수준을 사용할 수 있습니다.

- error - 최소 세부 정보를 제공합니다. 오류만 포함합니다.
- warn - 오류와 경고를 포함합니다.
- info - 기본 세부 수준입니다. 오류, 경고 및 정보 메시지를 포함합니다.
- debug - 최대 세부 정보를 제공합니다. 문제를 디버깅하는 데 유용한 세부 정보를 제공합니다.

Windows에서 로그 파일 세부 수준 변경

로그 파일 세부 수준을 구성하려면 Windows 레지스트리 편집기를 사용하여 level 파라미터를 구성해야 합니다.

Windows에서 로그 파일 세부 수준을 변경하려면

1. Windows 레지스트리 편집기를 엽니다.
2. HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/log/ 키로 이동합니다.
3. 두 번 클릭하여 레벨 파라미터를 엽니다. 필요한 세부 수준에 따라 Value data(값 데이터)에 error, warn, info 또는 debug를 입력합니다.
4. 확인을 선택하고 Windows 레지스트리 편집기를 닫습니다.

Linux에서 로그 파일 세부 수준 변경

로그 파일 세부 수준을 구성하려면 level 파일에서 dcv.conf 파라미터를 구성해야 합니다.

Linux에서 로그 파일 세부 수준을 변경하려면

1. `/etc/dcv/`로 이동하고 원하는 텍스트 편집기에서 `dcv.conf`를 엽니다.
2. `level` 섹션에서 `[log]` 파라미터를 찾은 후, 기존 세부 수준을 `error`, `warn`, `info` 또는 `debug`로 바꿉니다.

```
[log]
level="verbosity_level"
```

3. 파일을 저장하고 닫습니다.

Linux에서의 가상 세션 생성 문제 해결

주제

- [Linux에서 가상 세션 생성 실패 조사](#)
- [Linux에서 페일세이프 가상 세션 생성](#)

가상 세션에 연결할 때 `No session available` 또는 `The sessionId session is not available` 오류가 발생하는 경우, 이는 가상 세션 생성이 실패하고 종료되었기 때문일 수 있습니다.

`dcv list-sessions` 명령을 사용하여 세션이 존재하는지 확인할 수 있습니다. 실행 중인 세션 검사에 대한 자세한 내용은 [the section called “세션 보기”](#) 섹션을 참조하세요. 세션이 목록에 없으면 실패한 것일 수 있습니다.

Linux에서 가상 세션 생성 실패 조사

Linux에서는 다음 명령을 사용하여 가상 세션이 [생성](#)됩니다.

```
$ dcv create-session session
```

이 명령은 세션 생성이 실패한 경우에만 오류를 반환합니다. 하지만 세션이 처음에는 성공적으로 생성되었으나 사용자가 연결하기 전에 종료되는 경우가 발생할 수 있습니다. 예를 들어 `dcv list-sessions` 명령이나 `dcv describe-session session`을 사용하여 기존 세션을 확인할 때 아무 세션도 표시되지 않을 수 있기 때문에 이런 현상이 나타날 수 있습니다.

대부분의 경우 데스크톱 세션이 생성되었다가 즉시 실패하기 때문에 이런 현상이 발생합니다. 예를 들어 init 스크립트로 실행한 애플리케이션 중 하나가 충돌하거나 장애가 발생한 경우, 필요한 도구 중 하나가 누락된 경우 등이 있습니다.

세션 생성에 실패할 경우 다음 사항을 확인하세요.

- 새 세션 프로세스를 만드는 dcv 구성 요소와 관련된 로그가 포함된 `/var/log/dcv/sessionlauncher.log` 파일을 확인하세요.
- `dcv init` 스크립트와 관련된 로그가 포함된 `/var/log/dcv/dcv-session.user.session.log` 파일을 확인하세요.
- 세션 소유자에 해당하는 홈 디렉터리의 `$HOME/.xsession-errors` 파일을 확인하세요. 이 파일은 시스템 X 세션 init 스크립트에서 생성된 로그를 포함하고 있으며, 일반적으로 데스크톱 세션 관리자 또는 스크립트에서 호출한 다른 애플리케이션에서 생성된 로그를 포함하고 있습니다.
- 시스템 로그를 확인하여 장애가 발생한 시스템 및 구성 요소에 대한 자세한 내용을 확인하세요. 먼저 `dmesg`(예: 프로세스 장애 발생 시) 및 `journalctl -xe`의 출력을 확인하세요.
- [페일세이프\(failsafe\) 세션으로 테스트](#)하여 사용 중인 세션 관리자에 따라 문제가 달라지지 않는지 확인하세요.

특정 사용자에게만 실패가 발생하는 경우 다음을 시도해 볼 수도 있습니다.

- 사용자 구성, 특히 사용자 구성을 삭제하거나 이름을 바꿀 때 어떤 일이 발생하는지 확인하세요.
데스크톱 환경 및 버전에 따라 구성 디렉터리가 사용자 디렉토리의 `.gnome`, `.kde` 또는 `.config`에 있을 수 있습니다.
- 사용자 PATH 또는 환경에 영향을 미치는 특정 사용자 구성이 있는지 확인하세요. 특정 사용자의 세션 시작 실패는 세션 초기화 시 `dbus` 연결이 실패할 수 있는 일부 표준 기본 명령을 재정의하는 `anaconda` 등의 프레임워크 때문에 발생하는 경우가 많습니다.
- 권한 문제를 확인하세요. 로컬 `~/.dbus` 또는 `~/.Xauthority`에 잘못된 권한이 설정된 경우(예: 사용자 대신 `root`가 소유) 데스크톱 세션이 즉시 종료될 수 있습니다.

Linux에서 페일세이프 가상 세션 생성

세션 생성 실패가 데스크톱 환경 시작과 관련이 있는지 확인하는 일반적인 전략은 최소 세션을 생성하는 것입니다. 이 세션을 "페일세이프(failsafe)" 세션이라고 합니다. 페일세이프 세션이 제대로 생성된다면 기본 시스템 데스크톱 환경이 시작되지 않아 일반 세션이 실패한 것으로 추론할 수 있습니다. 반대로 페일세이프 세션도 실패하면 NICE DCV 서버 설정과 관련된 문제일 가능성이 높습니다.

파일세이프 세션은 보통 단순한 창 관리자와 터미널만 포함하는 데스크톱 세션으로 구성됩니다. 이를 통해 사용자는 사용 중인 특정 세션 환경(일반적으로 gnome 또는 KDE)과 관련된 세션 생성 문제가 있는지 확인할 수 있습니다.

파일세이프 세션을 생성하려면 다음과 같은 내용을 포함하는 사용자용 init 스크립트를 만들어야 합니다.

```
#!/bin/sh
metacity &
xterm
```

그러면 metacity 창 관리자가 시작되고 xterm 터미널이 시작되며, xterm 프로세스가 종료되는 즉시 세션도 종료됩니다.

시스템에서 사용할 수 있는 경우 원하는 다른 세션 관리자 또는 터미널을 사용할 수 있습니다.

Note

스크립트가 즉시 종료되지 않도록 해야 합니다. 이를 위해서는 스크립트가 끝날 때까지 즉시 종료되지 않는 프로그램을 실행해야 합니다. 마지막 명령이 종료되면(이 예제에서는 xterm) init 세션도 함께 종료됩니다. 동시에, 창 관리자 이후로 다른 도구를 실행할 때는 도구가 백그라운드에서 실행되는지 확인한 뒤(이 예제에서는 &를 추가) 다음 명령이 호출되도록 해야 합니다.

그런 다음 init 스크립트가 실행 가능한지 다음과 같이 확인해야 합니다.

```
$ chmod a+x init.sh
```

사용자 셸에서 지정된 init 스크립트를 사용하여 세션을 생성하려면 다음 명령을 실행합니다. `init.sh`는 이전에 생성된 스크립트입니다.

```
$ dcv create-session dummy --init init.sh
```

슈퍼 사용자로 다른 사용자를 위한 세션을 생성하려면 다음 명령을 대신 실행할 수 있습니다.

```
$ sudo dcv create-session test --user user --owner user --init init.sh
```

마지막으로 테스트 애플리케이션 `dcvgltest`(`nice-dcv-glttest` 패키지가 설치된 경우에만) 또는 `glxgears` 등을 실행하여 Open GL이나 기타 애플리케이션이 제대로 작동하는지 확인할 수 있습니다.

UID 변경 후 Linux 세션이 시작되지 않음

Linux 호스트에서 사용자의 사용자 ID(UID)를 변경하거나 사용자의 UID를 수정하는 다른 Active Directory 구성을 사용하면 호스트에서 NICE DCV 세션을 시작하지 못할 수 있습니다.

이 문제는 새 UID로 실행되는 DCV 세션의 프로세스가 이전 UID가 남아 있는 파일 및 폴더에 액세스할 수 있는 권한이 없기 때문에 발생합니다. 중요 사항:

- NICE DCV 로그 디렉터리의 [로그 파일](#)
- 사용자의 홈 폴더

이 문제는 콘솔 세션과 가상 세션 모두에 영향을 미칩니다.

이 문제를 해결하려면 사용자의 홈 폴더와 여기에 포함된 파일의 UID가 올바른지 확인하고 이전 UID가 있는 기존의 [NICE DCV 로그 파일](#)을 제거합니다.

Windows에서의 커서 문제 해결

NICE DCV 서버를 Windows Server 2012 또는 Windows 10 이상에서 실행 중인 경우 마우스 커서가 항상 화살표 모양으로 나타납니다. 이 문제는 텍스트 입력 필드 또는 한 번 클릭 탐색 항목에서 일시 중지하는 경우에도 발생합니다. 서버에 연결된 물리적 마우스가 없거나 장치 관리자에 마우스 장치가 나열되지 않은 경우 이러한 문제가 발생할 수 있습니다.

문제를 해결하려면

1. 제어판을 열고 Ease of Access Center(접근성 센터)를 선택합니다.
2. Make the mouse easier to use(마우스를 사용하기 쉽게 설정)을 선택합니다.
3. Turn on Mouse Keys(마우스 키 켜기)를 선택합니다.
4. 적용을 선택하고 확인을 선택합니다.

IntelliJ IDEA에 복사 및 붙여넣기 문제 해결

macOS NICE DCV 클라이언트에서 IntelliJ IDEA로 텍스트를 복사하여 붙여넣으려고 할 때 텍스트를 붙여넣을 수 없는 문제가 있습니다. IntelliJ는 NICE DCV가 기본적으로 사용하는 교차 플랫폼 형식을

받아들이지 못합니다. IntelliJ에 텍스트를 붙여넣을 수 있도록 NICE DCV에서 교차 플랫폼 텍스트를 비활성화하려면 NICE DCV 서버에서 `disabled-targets` 필드를 수정하세요.

이렇게 변경하면 NICE DCV 웹 클라이언트에서 복사 및 붙여넣기가 작동하지 않게 됩니다. 이 변경 사항을 적용하기 전에 IntelliJ IDEA에 대한 복사 및 붙여넣기가 NICE DCV 클라이언트에서만 작동하는지 확인한 후 변경해야 합니다.

IntelliJ IDEA에 텍스트를 붙여넣도록 서버를 구성하려면

1. `/etc/dcv/`로 이동하고 원하는 텍스트 편집기에서 `dcv.conf`를 엽니다.
2. `[clipboard]` 섹션에서 `disabled-targets` 파라미터를 찾습니다. `disabled-targets` 또는 `[clipboard]` 섹션이 없는 경우 직접 추가합니다.
3. 다음 내용을 추가하여 `disabled-targets`에 대한 값을 정의합니다.

```
[clipboard]
disabled-targets = ['dcv/text', 'JAVA_DATAFLAVOR:application/x-java-jvm-local-objectref; class=com.intellij.codeInsight.editorActions.FoldingData']
```

4. 파일을 저장하고 닫습니다.
5. NICE DCV 세션을 [중지](#)하고 [다시 시작](#)합니다.

자체 서명된 인증서에 대한 리디렉션 설명

웹 기반 포털 또는 애플리케이션에서 NICE DCV 세션으로 리디렉션할 때, 신뢰할 수 없는 자체 서명 인증서를 이전에 사용한 경우 해당 인증서가 세션과의 브라우저 신뢰를 깨뜨릴 수 있습니다. 이러한 상황의 예는 다음과 같습니다.

1. 사용자가 앱이 로드되는 회사 포털 사이트에 연결합니다.
2. 앱이 자체 서명된 인증서를 사용하여 NICE DCV 서버와의 직접적이고 안전한 연결을 시도합니다.
3. 인증서가 자체 서명된 것이기 때문에 브라우저가 보안 연결을 거부합니다.
4. 연결이 설정되지 않았으므로 사용자가 원격 서버를 사용할 수 없습니다.

신뢰 문제는 3단계에만 해당됩니다. 사용자가 자체 서명된 인증서를 사용하여 웹사이트에 연결하면 (예: `https://example.com`으로 이동) 브라우저가 인증서를 신뢰하도록 요청합니다. 그러나 HTTP 또는 HTTPS를 통해 제공되는 웹 앱/페이지가 DCV 서버에 대한 안전한 WebSocket 연결을 설정하려고 시도하는 경우가 있습니다. 자체 서명된 인증서의 경우 브라우저는 인증서가 이전에 신뢰할 수 있었는지

확인합니다. 이전에 신뢰할 수 없었던 인증서는 사용자에게 인증서를 신뢰할지 묻는 메시지를 표시하지 않고 연결을 거부합니다.

이 경우 가능한 해결 방법은 다음과 같습니다.

- 회사에서 컴퓨터에 사용자 지정 도메인을 사용하는 경우 DCV 서버 컴퓨터에 대한 유효한 인증서가 있어야 합니다. 인증서의 경우 기업 인증서를 DCV에 배포할 수 있습니다.

Example

사용자 ---[유효한 인증서]---> DCV 서버 인스턴스

- 프록시/게이트웨이에서 DCV 서버 플릿을 보호합니다. 이 경우에만 프록시/게이트웨이에 유효한 인증서가 있어야 하며 DCV 서버 인스턴스는 자체 서명된 인증서를 유지할 수 있습니다. 이 옵션의 경우 [DCV 연결 게이트웨이](#), ALB/NLB 또는 다른 프록시 솔루션을 사용할 수 있습니다.

Example

사용자/Cx ---[유효한 인증서 필요]---> 프록시/게이트웨이---[자체 서명 인증서]---> DCV 서버 인스턴스

- [SDK](#)를 통해 연결을 시작하기 전에 사용자가 자체 서명된 인증서를 신뢰하도록 하세요. 다른 탭/창/팝업에서 이 URL(<https://example.com/version>)을 열기만 하면 이 작업이 가능합니다.

Note

/version 엔드포인트는 HTTPS 연결 상태의 DCV 서버 버전에 대한 간단한 웹 페이지로 응답합니다.

동일한 자체 서명 인증서를 나중에 실제 DCV 서버 연결에서 사용할 수 있습니다.

Windows에서 NVIDIA GPU 사용 시 다중 모니터/전체 화면 오류

Windows 서버 호스트에 NVIDIA GPU가 있는 경우 DCV 전체 화면/다중 모니터 기능이 작동하지 않을 수 있습니다. 이 경우 디스플레이가 전체 화면 모드로 전환되지 않거나 서버가 여러 원격 모니터로 디스플레이 레이아웃을 구성하지 못하게 됩니다.

이 문제의 원인은 NVIDIA 드라이버와의 통합 실패입니다.

서버 호스트의 C:\ProgramData\NICE\dcv\log\ 경로를 살펴보면 오류를 확인할 수 있습니다.

WARN display - Cannot change display layout

이 오류는 다음과 같이 표시되기 전에 여러 번(20~30회) 표시됩니다.

```
EDID not set on output x gpu x after attempt x INFO DLMNVAPI:display -
Unable to set EDID on output x, gpu x: NVAPI_ERROR (-1)
```

문제가 재현되면 호스트가 비정상 상태인 것입니다. 즉, 서버가 다중 모니터 레이아웃을 일관되게 구성할 수 없고, 문제를 영구적으로 해결할 수 있는 방법이 없습니다(몇 가지 임시 방편만 있음).

다중 모니터를 사용하는 동안, 즉 호스트가 종료될 때 서버 호스트에 가상 모니터가 있는 경우 서버 OS를 재부팅하면 문제가 발생합니다. 따라서 이 문제를 방지하려면 서버를 종료하기 전에 서버 측의 모든 모니터를 제거해야 합니다. 다음 명령(관리자 권한으로 실행)을 사용하여 이를 확인할 수 있습니다.

```
C:\Program Files\NICE\DCV\Server\bin\dcvnvedid.exe --remove
```

가능한 임시 방편은 NVIDIA 드라이버를 재설치 또는 업데이트하고 호스트를 재부팅하는 것입니다.

NICE DCV 성능 및 통계 모니터링

NICE DCV 서버 2023.1 버전부터 Windows 성능 카운터를 사용하여 프로토콜 성능의 다양한 측면을 모니터링하고 NICE DCV 세션 및 연결에 대한 통계를 수집할 수 있습니다.

성능 카운터를 수집하는 도구:

- [PerfMon\(성능 모니터\)](#): 실시간으로 또는 로그 파일에서 성능 데이터를 시각화할 수 있는 Windows 기본 도구입니다.
- [LogMan](#): 지정된 기준에 따라 로깅을 시작하고 중지할 수 있는 명령줄 도구입니다.
- [TypePerf](#): 명령 창이나 로그 파일에 성능 데이터를 쓰는 명령줄 도구입니다.
- [PowerShell](#): 성능 데이터를 수집하고 조작하는 데 사용할 수 있는 Windows 스크립팅 언어입니다.
- 서드 파티 도구: 이러한 카운터를 수집하고 심층적인 정보를 제공하는 여러 서드 파티 모니터링 솔루션이 있습니다.

DCV 성능 카운터는 5개의 카운터 세트로 그룹화됩니다.

카운터 세트

NICE DCV 서버

이 카운터 세트에는 호스트의 DCV 서버 서비스에 대한 전역 통계가 포함되어 있습니다. 또한 이 카운터 세트는 다른 카운터 세트에서도 사용할 수 있는 여러 카운터를 집계하여 정적 경로를 통해 서버의 전체 수명 동안 집계된 정보에 액세스할 수 있는 방법을 제공합니다. 이 카운터 세트의 카운터를 읽기 위해 세션 또는 연결 식별자를 검색할 필요가 없습니다.

Note

다른 카운터 세트 중 하나에서 집계된 인스턴스(예: "\DCV Server Connections(_Total)\Sent Bytes")는 모든 활성 연결에 대한 합계를 반환하며, 전역 카운터는 서버가 시작된 순간부터 누적되고 종료된 연결을 포함합니다.

카운터 이름	설명	단위	참고
활성 세션	호스트의 활성 세션 수	개수	
총 세션	종료된 세션을 포함하여 호스트에 생성된 세션 증가 수	개수	
활성 연결	서버에 대한 활성 연결 수	개수	
총 연결	활성 클라이언트, 다시 연결된 클라이언트 및 연결이 끊긴 클라이언트를 포함하여 서버에 대한 연결의 증가 수	개수	
유휴 연결 해제	비활성 상태로 인해 끊긴 연결의 증가 수	개수	

카운터 이름	설명	단위	참고
수신 속도 비트/초	서버에서 데이터를 수신하는 속도(초당 비트 수)	비트/초	
수신된 바이트	서비스 시작 이후 수신된 총 바이트 수	바이트	
전송 속도 비트/초	서버에서 데이터를 전송하는 속도(초당 비트 수)	비트/초	
전송된 바이트	서비스 시작 이후 전송된 총 바이트 수	바이트	
HTTP 다운로드 속도 비트/초	발신 HTTP 트래픽의 대역폭(초당 비트 수)	비트/초	파일 스토리지에 대한 클라이언트-서버 트래픽은 수신 속도에 포함됨
HTTP 다운로드 바이트	서비스 시작 이후 HTTP를 통해 전송된 총 바이트 수	바이트	파일 스토리지에 대한 클라이언트-서버 트래픽은 수신된 바이트 수에 포함됨
왕복 시간 ms	서버와 클라이언트 간 평균 왕복 지연 시간 (밀리초)	밀리초	5초에 한 번씩 측정 및 업데이트
최소 왕복 시간 ms	서비스 시작 이후 감지된 최소 왕복 지연 시간 (밀리초)	밀리초	5초에 한 번씩 업데이트

DCV 서버 프로세스

이 카운터 세트에는 개별 NICE DCV 프로세스에 대한 정보가 포함되어 있습니다.

agent_type can be one of: session_agent, system_agent, user_agent

카운터는 1초에 한 번씩 업데이트됩니다.

카운터 이름	설명	단위	참고
% 프로세서 시간	프로세스에서 사용된 프로세서 시간 비율	%	비율은 CPU 코어 1개를 기준으로 합니다 (즉, 100%는 프로세스가 하나의 스레드를 차지하고 있음을 의미). \Process(NAME)\% Processor Time과 동일
물리적 메모리 바이트	프로세스에서 사용된 물리적 메모리의 현재 양(바이트)	바이트	\Process(NAME)\Working Set와 동일
가상 메모리 바이트	프로세스의 가상 주소 공간 현재 크기(바이트)	바이트	
프로세스 식별자	숫자 프로세스 식별자 (PID)	-	

NICE DCV 서버 세션

이 세트의 카운터는 단일 세션에 대한 정보를 제공합니다. 사용자의 연결 여부에 관계없이, 생성된 세션마다 이 카운터 세트의 인스턴스가 하나씩 있습니다.

관리자가 세션을 닫으면 해당 인스턴스가 제거되고, 관리자가 같은 이름의 세션을 다시 만들면 모든 카운터가 0부터 다시 시작됩니다.

카운터 이름	설명	단위
세션 지속 시간 sec	세션이 열려 있던 총 시간(초)	초

카운터 이름	설명	단위
총 픽셀	디스플레이 영역의 픽셀 수. 세션 내 모든 디스플레이의 픽셀 수를 합한 값	픽셀
디스플레이 수	세션 내 디스플레이 수	개수

다음 카운터는 설명에 약간의 차이가 있지만 NICE DCV 서버 카운터 세트의 카운터와 동일합니다.

카운터 이름	설명
활성 연결	세션 인스턴스에 대한 활성 연결 수
총 연결	활성 클라이언트, 다시 연결된 클라이언트 및 연결이 끊긴 클라이언트를 포함하여 세션 인스턴스에 대한 연결의 증가 수
유휴 연결 해제	세션 인스턴스에서 비활성 상태로 인해 끊긴 연결의 증가 수
비정상적 연결 해제	세션 인스턴스에서 오류로 인해 끊긴 연결의 증가 수
수신 속도 비트/초	세션 내에서 데이터를 수신하는 속도(초당 비트 수)
수신된 바이트	세션 시작 이후 수신된 총 바이트 수
전송 속도 비트/초	세션 내에서 데이터가 전송되는 속도(초당 비트 수)
전송된 바이트	세션 시작 이후 전송된 총 바이트 수
HTTP 다운로드 속도 비트/초	세션 내에서 발신 HTTP 데이터에 대한 대역폭(초당 비트 수)
HTTP 다운로드 바이트	세션 내에서 HTTP를 통해 전송된 총 바이트 수

카운터 이름	설명
왕복 시간 ms	서버와 클라이언트 간 세션 내 평균 왕복 지연 시간(밀리초)
최소 왕복 시간 ms	세션 구성 이후 감지된 최소 왕복 지연 시간(밀리초)

NICE DCV 서버 연결

이 세트의 카운터는 단일 클라이언트 연결에 대한 정보를 제공합니다. 카운터 세트 인스턴스는 클라이언트가 서버에 연결될 때 생성되고 클라이언트 연결이 끊어지면 삭제됩니다. connection_id는 숫자이며 한 서버 세션 내에서만 고유합니다.

카운터 이름	설명	단위
연결 지속 시간 초	연결이 열려 있던 총 시간(초)	초

다음 카운터는 설명에 약간의 차이가 있지만 “DCV 서버” 카운터 세트의 카운터와 동일합니다.

카운터 이름	설명
수신 속도 비트/초	연결 내에서 데이터가 수신되는 속도(초당 비트 수)
수신된 바이트	연결이 설정된 이후 수신된 총 바이트 수
전송 속도 비트/초	연결 내에서 데이터가 전송되는 속도(초당 비트 수)
전송된 바이트	연결이 설정된 이후 전송된 총 바이트 수
HTTP 다운로드 속도 비트/초	연결 내에서 발신 HTTP 데이터에 대한 대역폭(초당 비트 수)
HTTP 다운로드 바이트	연결이 설정된 이후 HTTP를 통해 전송된 총 바이트 수

카운터 이름	설명
왕복 시간 ms	연결의 평균 왕복 지연 시간(밀리초)
최소 왕복 시간 ms	연결이 설정된 이후 감지된 최소 왕복 지연 시간(밀리초)

NICE DCV 서버 채널

이 세트의 카운터는 클라이언트 연결의 개별 채널에 대한 정보를 제공합니다. 확장을 위한 추가 채널이 있을 수 있습니다.

채널 이름은 다음과 같습니다.

- `dcv::main`
- `dcv::display`
- `dcv::input`
- `dcv::audio`
- `dcv::filestorage`
- `dcv::clipboard`

수신되는 파일 스토리지 트래픽은 `dcv::filestorage` 채널에 귀속됩니다.

발신되는 파일 스토리지 트래픽은 DCV 서버 연결의 HTTP 다운로드 카운터에 포함됩니다.

Note

이 세트의 카운터는 DCV 서버 연결에 있는 카운터의 하위 집합입니다.

카운터 이름	설명
수신 속도 비트/초	채널을 통해 데이터를 수신하는 속도(초당 비트 수)
수신된 바이트	채널을 통해 수신된 총 바이트 수

카운터 이름	설명
전송 속도 비트/초	채널을 통해 데이터가 전송되는 속도(초당 비트 수)
전송된 바이트	채널을 통해 전송된 총 바이트 수

NICE DCV 서버 이미징

이 세트의 카운터는 화면 캡처, 인코딩 및 전송을 담당하는 하위 시스템에 대한 정보를 제공합니다.

이 세트의 카운터는 두 그룹으로 나뉩니다.

- 첫 번째 그룹의 경우 NICE DCV는 세션당 하나의 값을 수집하여 `$session_name` 인스턴스에 게시합니다.
- 두 번째 그룹의 경우 NICE DCV는 세션의 각 인코더에 대해 하나의 값을 수집합니다. 다음과 같은 세 가지 활성 인코더가 있습니다.
 - 풀프레임 인코더 1개
 - 타일 기반 인코더 1개
 - 무손실 인코더 1개

이러한 카운터는 `$session_name:$encoder_name` 인스턴스에 게시됩니다.

카운터 이름	설명	단위	인스턴스
초당 캡처 프레임	캡처된 프레임 속도(초당 프레임 수)	개수/초	세션
캡처된 프레임	세션 시작 이후 캡처된 총 프레임 수	개수	세션
초당 전송 프레임	연결된 클라이언트로 초당 전송된 화면 프레임 비율	개수/초	세션

카운터 이름	설명	단위	인스턴스
초당 드롭 프레임	연결된 클라이언트로 초당 전송되지 않은 화면 프레임 비율	개수/초	세션
디스플레이 지연 시간 ms	프레임 캡처와 프레젠테이션 사이의 평균 시간(밀리초)	밀리초	세션
가용 대역폭 비트/초	연결에서 사용할 수 있는 예상 대역폭(초당 비트 수)	비트/초	세션
초당 인코딩 프레임	초당 인코딩된 화면 프레임 비율	개수/초	세션:인코더
인코딩 시간 ms	화면 프레임 하나를 인코딩하는 데 사용되는 평균 시간(밀리초)	밀리초	세션:인코더
메가픽셀당 인코딩 시간 ms	100만 픽셀을 인코딩하는 데 사용되는 평균 시간(밀리초)	밀리초	세션:인코더
프레임 품질 %	평균 프레임 압축 품질 (백분율로 표시)	%	세션:인코더
프레임 압축률 %	프레임 크기(바이트)와 압축된 프레임 크기 간의 비율로 정의되는 평균 프레임 압축률	%	세션:인코더

NICE DCV 서버 파라미터 참조

다음 표에 NICE DCV 서버를 사용자 지정하도록 구성할 수 있는 파라미터가 나와 있습니다.

Note

각 테이블의 컨텍스트 다시 로드 열은 파라미터가 다시 로드되는 시기를 나타냅니다. 가능한 컨텍스트는 다음과 같습니다.

- **server**—서버가 시작될 때 파라미터가 한 번 로드됩니다. 파라미터 값이 업데이트되면 서버가 다시 시작될 때 새 값이 로드됩니다.
- **session**—세션이 생성될 때 파라미터가 로드됩니다. 파라미터 값이 업데이트되면 후속 세션에 대해 새 값이 로드됩니다.
- **connection**—새 클라이언트 연결이 설정되면 파라미터가 로드됩니다. 파라미터 값이 업데이트되면 후속 클라이언트 연결에 새 값이 사용됩니다.
- **custom**—파라미터가 로드되는 조건이 이 파라미터에 고유합니다. 자세한 내용은 파라미터 설명을 참조하세요.

주제

- [audio 파라미터](#)
- [clipboard 파라미터](#)
- [connectivity 파라미터](#)
- [display 파라미터](#)
- [display/linux 파라미터](#)
- [input 파라미터](#)
- [license 파라미터](#)
- [log 파라미터](#)
- [printer 파라미터](#)
- [redirection 파라미터](#)
- [security 파라미터](#)
- [session-management 파라미터](#)

- [session-management/automatic-console-session](#) 파라미터
- [session-management/defaults](#) 파라미터
- [smartcard](#) 파라미터
- [webcam](#) 파라미터
- [windows](#) 파라미터
- [구성 파라미터 수정](#)

audio 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[audio]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 audio 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
<code>avsync-support</code>	문자열	세션	'auto'	클라이언트가 오디오/비디오 동기화를 활성화할 수 있는지 확인 — 연결된 클라이언트가 오디오/비디오 동기화를 활성화할 수 있도록 허용합니다. 유효한 값은 'enabled', 'disabled' 또는 'auto'(기본값 = 'auto')입니다. 'auto'를 지정하면 오디오/비디오 동기화는 콘솔 세션에서만, 그리고 가속화된 비디오 압축을 사용할 수 있는 경우에만 활성화됩니다. — 버전 2021.1-10557 부터 사용 가능합니다.
<code>source-channels</code>	정수 - DWORD(비트)	세션	2	Linux 스피커 장치의 채널 수 — Linux 스피커 장치의 채널 수를 설정합니다. 값은 장치

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				에서 지원하는 채널 수보다 작거나 같아야 합니다. 허용되는 값은 2(스테레오), 4(4.0 쿼드러포닉), 6(5.1 서라운드), 8(7.1 서라운드)입니다. 기본값은 2(스테레오)입니다. — 버전 2020.0-8428 부터 사용 가능합니다.

clipboard 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[clipboard]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 `clipboard` 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
<code>enabled</code>	<code>true</code> 또는 <code>false</code> - DWORD(; 비트)	세션	Linux: <code>true</code> - Windows: <code>1</code>	클립보드 기능 활성화 여부 — 클립보드 기능을 활성화할지 여부를 지정합니다. 클립보드 기능이 비활성화된 경우 사용자는 클립보드 원격화를 사용할 수 없습니다. 클립보드 모니터링도 비활성화됩니다. — 버전 2020.0-8428 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
max-image-area	정수 - DWORD(비트)	세션	-1	클립보드 이미지의 최대 영역 — 서버와 클라이언트가 주고 받을 수 있는 클립보드 이미지 의 최대 영역(픽셀 수)을 지정 합니다. 이 값이 없거나 -1로 설 정되어 있는 경우, 어떠한 제한 도 적용되지 않습니다. — 버전 2017.0-4334 부터 사용 가능합 니다.
max-payload-size	정수 - DWORD(비트)	세션	20971520	클립보드 데이터의 최대 크기 — 서버와 클라이언트가 주고 받을 수 있는 클립보드 데이터 의 최대 크기(바이트 단위)를 지정합니다. 지원되는 최대 값 은 20MB입니다. 이 값이 없는 경우, 최대 제한 값이 적용됩니 다. — 버전 2017.0-4334 부터 사용 가능합니다.
max-text-len	정수 - DWORD(비트)	세션	-1	클립보드 텍스트의 최대 문자 수 — 서버에서 클라이언트로 전송 가능한 클립보드 텍스트 의 최대 문자 수를 지정합니다. 최대 문자 수를 초과하는 경우 초과 문자는 잘립니다. 이 값 이 없거나 -1로 설정되어 있는 경우, 어떠한 제한도 적용되지 않습니다. — 버전 2017.0-43 34 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
primary-selection-copy	true 또는 false - DWORD(32비트)	세션	Linux: false - Windows: 0	Linux에서 기본 선택 사본 활성화 — Linux 데스크톱이 여러 클립보드(일반 클립보드 및 기본 선택)를 지원합니다. 콘텐츠를 선택하면 기본 선택 항목이 업데이트되거나 복사됩니다. 그런 다음 마우스 가운데 버튼을 사용하거나 Shift+Insert 키 조합을 사용하여 붙여 넣을 수 있습니다. 사용하도록 설정하면 기본 선택이 모니터링되고 업데이트가 클라이언트에 전파됩니다. — 버전 2019.0-7318 부터 사용 가능합니다.
primary-selection-paste	true 또는 false - DWORD(32비트)	세션	Linux: false - Windows: 0	Linux에서 기본 선택 붙여 넣기 활성화 — Linux 데스크톱이 여러 클립보드(클립보드 및 기본 선택)를 지원합니다. 콘텐츠를 선택하면 기본 선택 항목이 업데이트되거나 복사됩니다. 그런 다음 마우스의 가운데 버튼 또는 Shift+Insert 키 조합을 사용하여 붙여 넣을 수 있습니다. 활성화하면 클라이언트의 클립보드 콘텐츠도 기본 선택에 삽입됩니다. — 버전 2019.0-7318 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
update-timeout	정수 - DWORD(; 비트)	세션	200	업데이트 이벤트 알림 제한 시 간 - 마지막 업데이트 이벤트로 부터 클라이언트에 알림을 보 내기 위해 대기할 시간을 밀리 초 단위로 지정합니다. 기본값 은 200밀리초입니다. — 버전 2020.1-8942 부터 사용 가능합 니다.

connectivity 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[connectivity]` 섹션에 대
한 구성 파라미터와 Windows NICE DCV 서버용 `connectivity` 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
disconnect-on-lock	true 또는 false - DWORD(; 비트)	사용자 지정	Linux: false - Windows: 0	OS 세션 잠금 시 클라이언트 연결이 끊겼는지 여부 - 원격 OS 세션이 잠겼을 때 클라이 언트 연결을 강제로 끊으려면 이 옵션을 활성화합니다. 그렇 지 않으면 클라이언트가 원격 세션을 계속 스트리밍합니다. 현재는 콘솔 세션에서만 지원 됩니다. 이 파라미터 값은 모든 원격 OS 세션 잠금에서 읽습

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				니다. — 버전 2023.1부터 사용 가능합니다.
disconnect-on-logout	true 또는 false - DWORD(; 비트)	사용자 지정	Linux: false - Windows: 0	OS 사용자 로그아웃 시 클라이언트 연결이 끊어지는지 여부 — 원격 OS 사용자가 로그아웃할 때(예: OS 세션이 닫힌 경우) 클라이언트 연결을 강제로 끊으려면 이 옵션을 활성화합니다. 그렇지 않으면 클라이언트가 원격 세션을 계속 스트리밍합니다. 현재는 콘솔 세션에서만 지원됩니다. 이 파라미터 값은 원격 OS 사용자가 로그아웃할 때마다 읽습니다. — 버전 2023.1부터 사용 가능합니다.
enable-quick-frontend	true 또는 false - DWORD(; 비트)	서버	Linux: false - Windows: 0	QUIC 프론트엔드 활성화 여부 — QUIC 프론트엔드를 활성화할지 여부를 지정합니다. — 버전 2020.2-9508 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
idle-timeout	정수 - DWORD(비트)	사용자 지정	60	유희 제한 시간 — 유희 클라이언트 연결을 끊기 전에 대기하는 시간(분 단위)을 지정합니다. 0을 지정하면 유희 클라이언트의 연결을 절대 끊지 않습니다. 이 파라미터 값은 5초마다 읽습니다. — 버전 2017.0-4100 부터 사용 가능합니다.
idle-timeout-warning	정수 - DWORD(비트)	사용자 지정	350	유희 제한 시간 경고 — 유희 제한 시간 연결 해제에 대해 유희 클라이언트에 경고하기 전에 대기할 idle-timeout 기준 시간(초)을 지정합니다. 유희 클라이언트에 경고하지 않으려면 0을 지정합니다. — 버전 2017.4-6898 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
quic-listen-endpoints	문자열	서버	['0.0.0.0', '::']	<p>DCV가 들어오는 QUIC 연결을 수신하는 엔드포인트 지정 — DCV가 들어오는 QUIC 연결을 수신할 엔드포인트 목록을 지정합니다. 엔드포인트는 로컬 바인딩 가능한 IPv4 주소 목록('0.0.0.0' - 가능한 모든 주소를 와일드카드로 지정) 또는 콜론(':')으로 구분된 선택적 포트를 포함하는 바인딩 가능한 IPv6 주소('::' - 가능한 모든 주소를 와일드카드로 지정)의 목록일 수 있습니다. 예를 들어, '1.2.3.4:5678'은 포트 5678에서 '1.2.3.4' 주소와 연결된 인터페이스에서 들어오는 연결을 수신합니다. 포트가 지정되지 않은 경우 'quic-port'의 설정이 기본값으로 사용됩니다. IPv6 주소로 포트를 지정하려면 주소를 대괄호로 묶으세요 (예: ':::1]:8443'). 명시적 인터페이스를 포함한 IPv6 주소도 지원됩니다(예: ':::%eth1]:8443').</p> <p>— 버전 2022.0-11954부터 사용 가능합니다.</p>

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
quic-port	정수 - DWORD(; 비트)	서버	8443	QUIC 프론트엔드용 UDP 포트 — DCV 서버가 클라이언트 연결을 수신하는 UDP 포트를 지정합니다. 포트 번호는 1024~65535 사이여야 합니다. 이 설정이 적용되는 방법에 대한 자세한 내용은 'quic-listen-endpoints' 설정을 참조하세요. — 버전 2020.2-9508 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
web-listen-endpoints	문자열	서버	['0.0.0.0', '::']	<p>DCV가 들어오는 웹 연결을 수신하는 엔드포인트 지정 — DCV가 들어오는 웹 연결을 수신할 엔드포인트 목록을 지정합니다. 엔드포인트는 로컬 바인딩 가능한 IPv4 주소 목록 ('0.0.0.0' - 가능한 모든 주소를 와일드카드로 지정) 또는 콜론(':')으로 구분된 선택적 포트를 포함하는 바인딩 가능한 IPv6 주소('::' - 가능한 모든 주소를 와일드카드로 지정)의 목록일 수 있습니다. 예를 들어, '1.2.3.4:5678'은 포트 5678에서 '1.2.3.4' 주소와 연결된 인터페이스에서 들어오는 연결을 수신합니다. 포트가 지정되지 않은 경우 'web-port'의 설정이 기본값으로 사용됩니다. IPv6 주소로 포트를 지정하려면 주소를 대괄호로 묶으세요 (예: ':::1]:8443'). 명시적 인터페이스를 포함한 IPv6 주소도 지원됩니다(예: ':::%eth1]:8443'). — 버전 2022.0-11954부터 사용 가능합니다.</p>

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
web-port	정수 - DWORD(비트)	서버	8443	클라이언트용 TCP 포트 — DCV 서버가 클라이언트 연결을 수신하는 TCP 포트를 지정합니다. 포트 번호는 1024~65535 사이여야 합니다. 이 설정이 적용되는 방법에 대한 자세한 내용은 'web-listen-endpoints' 설정을 참조하세요. — 버전 2017.0-4100 부터 사용 가능합니다.
web-root	문자열	서버	"	포함된 웹 서버용 문서 루트 — 포함된 웹 서버용 문서 루트를 지정합니다. — 버전 2017.0-4100 부터 사용 가능합니다.
web-url-path	문자열	서버	'/'	포함된 웹 서버용 URL 경로 — 포함된 웹 서버용 URL 경로를 지정하며 '/'로 시작해야 합니다. 예를 들면 /test/foo로 설정하는 것은 웹 서버가 https://host:port/test/foo에 접속할 수 있음을 의미합니다. — 버전 2017.0-4100 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
web-use-hsts	true 또는 false - DWORD(32- 비트)	서버	Linux: true - Windows: 1	HSTS 사용 여부 — 브라우저가 HTTP를 통해 전송되는 통신을 방지하도록 할 수 있습니다. 웹 페이지(및 모든 하위 도메인)에 대한 모든 전송은 대신 HTTPS를 사용하여 수행됩니다. — 버전 2017.0-4100 부터 사용 가능합니다.
ws-keepalive-interval	정수 - DWORD(32- 비트)	서버	10	WebSocket keepalive 간격 — keepalive 메시지를 보낼 시간 간격을 초 단위로 지정합니다. 0으로 설정하면 keepalive 메시지가 비활성화됩니다. — 버전 2017.0-4100 부터 사용 가능합니다.

display 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[display]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 `display` 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
console-session-default-layout	문자열	세션	[]	기본 화면 해상도 및 콘솔 세션 위치 — 기본 화면 해상도

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				<p>와 콘솔 세션의 위치를 지정합니다. 이 항목을 설정하면 DCV가 시작 시 요청되는 레이아웃으로 설정됩니다. 각 모니터를 해상도(w,h)와 위치(x,y)로 구성할 수 있습니다. 모든 지정된 모니터가 활성화됩니다. 기본 레이아웃 예제 값: <code>[{'w':<800>, 'h':<600>, 'x':<0>, 'y': <0>}, {'w':<1024>, 'h':<768>, 'x':<800>,'y':<0>}]</code> — 버전 2017.0-5600부터 사용 가능합니다.</p>

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
cuda-devices	문자열	ODBC	[]	스트림 인코딩에 사용되는 CUDA 장치 — DCV에서 인코딩 및 CUDA 워크로드를 분산하는 데 사용하는 로컬 CUDA 장치 목록을 지정합니다. 각 장치는 nvidia-smi 명령에서 검색할 수 있는 숫자로 식별됩니다. 예를 들어, cuda-devices=['0', '2']는 DCV가 ID 0과 2로 두 개의 GPU를 사용함을 나타냅니다. 이 설정은 CUDA_VISIBLE_DEVICES 환경 변수와 유사하지만 DCV에만 적용됩니다. 이 옵션을 설정하지 않으면 DCV는 0부터 시작하는 증분 세션 인덱스를 사용하여 사용할 다음 디바이스를 선택합니다. — 버전 2017.2-6182 부터 사용 가능합니다.
enable-client-resize	true 또는 false - DWORD(; 비트)	세션	Linux: true - Windows: 1	클라이언트의 표시 레이아웃 설정에 대한 허용 여부 — 클라이언트가 표시 레이아웃을 설정하도록 허용할지 여부를 지정합니다. — 버전 2017.0-4100 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
enable-qu	true 또는 false - DWORD(32 비트)	세션	Linux: true - Windows: 1	품질 업데이트 전송 여부 — 품질 업데이트를 전송할지 여부를 지정합니다. — 버전 2017.0-4100 부터 사용 가능합 니다.
enable-yuv444- encoding	문자열	세션	'default-off'	YUV444 인코딩 활성화 여 부 — YUV444 인코딩을 활 성화하거나 비활성화합니다. 'always-on' 상태인 경우 서버 는 높은 색상 정확도에 최적화 된 YUV444 형식을 우선 사용 합니다. 'always-off' 상태인 경 우 서버는 스트리밍 성능에 최 적화된 형식을 우선 사용합니 다. 'default-on' 및 'default-off' 값은 의미가 같아서 클라이언 트에 결정을 맡깁니다. 허용된 값은 'always-on', 'always-off', 'default-on', 'default-off'입니 다. — 버전 2022.0-11954 부터 사 용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
grabber-target-fps	정수 - DWORD(비트)	세션	0	프레임 그래버의 초당 대상 프레임 수 — 초당 프레임을 확보하기 위한 상한을 설정합니다. 값이 0이면 각 특정 프레임 버퍼 리더의 표준 동작이 기본적으로 적용됩니다(예: target-fps로 폴백하거나 확보를 제한하지 않음). 모든 프레임 캡처 백엔드가 이 설정을 준수하는 것은 아닙니다. — 버전 2017.1-5870 부터 사용 가능합니다.
max-compressor-threads	정수 - DWORD(비트)	세션	4	최대 압축 스레드 — 최대 압축 스레드 수를 지정합니다. — 버전 2017.0-4100 부터 사용 가능합니다.
max-head-resolution	문자열	사용자 지정	(4096, 2160)	최대 헤드 해상도 — 클라이언트에서 요청 가능한 디스플레이 헤드의 최대 해상도입니다. 디스플레이 헤드는 호스트 모니터와 동일합니다. 클라이언트 레이아웃 요청별로 설정이 다시 로드됩니다. 클라이언트가 더 큰 헤드 해상도를 요청하면 서버는 해상도를 조정하여 이 옵션으로 설정된 최대 너비 및 높이 값과 일치하는지 확인합니다. — 버전 2017.0-4100 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
max-layout-area	정수 - DWORD(비트)	사용자 지정	0	최대 레이아웃 영역(픽셀 단위) — 클라이언트가 요청한 디스 플레이 레이아웃의 최대 영역 (픽셀 단위)입니다. 이 제한보 다 큰 레이아웃은 무시됩니다. 이 최대값은 표시 레이아웃 지 오메트리에 제약을 가하지 않 으면서 전송해야 하는 표시 데 이터 양의 상한에 해당합니다. 0으로 설정하면 레이아웃 영역 에 제한이 적용되지 않습니다. 클라이언트 레이아웃 요청별로 설정이 다시 로드됩니다. — 버 전 2019.1-7423 부터 사용 가능 합니다.
max-num-heads	정수 - DWORD(비트)	사용자 지정	4	최대 헤드 수 — 클라이언트에 서 요청 가능한 최대 디스플레 이 헤드 수를 지정합니다. 디스 플레이 헤드는 호스트 모니터 와 동일합니다. 클라이언트 레 이아웃 요청별로 설정이 다시 로드됩니다. 클라이언트가 더 많은 수의 헤드를 요청하면 서 버는 이 옵션으로 설정된 값을 초과하지 않도록 헤드 수를 조 정합니다. — 버전 2017.0-41 00 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
min-head-resolution	문자열	사용자 지정	(640, 480)	최소 헤드 해상도 — 클라이언트에서 요청 가능한 디스플레이 헤드의 최소 해상도를 설정합니다. 디스플레이 헤드는 호스트 모니터와 동일합니다. 클라이언트 레이아웃 요청별로 설정이 다시 로드됩니다. 클라이언트가 더 작은 해상도를 요청하면 서버는 해상도를 조정하여 이 옵션으로 설정된 최소 너비 및 높이 값과 일치하는지 확인합니다. — 버전 2017.0-4100 부터 사용 가능합니다.
target-fps	정수 - DWORD(비트)	세션	-1	초당 타겟 프레임 — 초당 허용되는 최대 프레임을 지정합니다. 값 0은 제한이 없음을 의미합니다. 값이 -1이면 서버 특성 및 세션 유형에 따라 target-fps 값이 결정됩니다. 버전 2020.2 이하에서는 -1 값이 인식되지 않으며, 기본값은 25입니다. — 버전 2017.0-4100 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
use-grabber-dirty-region	true 또는 false - DWORD(; 비트)	세션	Linux: true - Windows: 1	더티 영역 사용 여부 — 화면의 더티 영역을 사용할지 여부를 지정합니다. 활성화하면, 그라버가 화면의 더티 영역에서 새로운 프레임의 계산을 시도합니다. — 버전 2017.0-4100 부터 사용 가능합니다.
web-client-max-head-resolution	문자열	사용자 지정	(1920, 1080)	웹 클라이언트의 최대 헤드 해상도 — 웹 클라이언트에서 요청 가능한 디스플레이 헤드의 최대 해상도를 설정합니다. 디스플레이 헤드는 호스트 모니터와 동일합니다. 클라이언트 레이아웃 요청별로 설정이 다시 로드됩니다. 이 설정은 웹 클라이언트가 최대 해상도를 명시적으로 설정하는 경우 무시됩니다. max-head-resolution 제한 옵션은 이 옵션으로 설정된 최대 너비 및 높이 값에 추가로 적용됩니다. 값이 (0, 0)으로 설정된 경우 무시됩니다. — 버전 2020.0-8428 부터 사용 가능합니다.

display/linux 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[display/linux]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 `display/linux` 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
gl-displays	문자열	세션	[:0.0]	3D 가속 X 디스플레이 — DCV가 가상 세션에서 OpenGL 렌더링을 위해 사용하는 로컬 3D 가속 X 디스플레이와 화면의 목록을 지정합니다. 이 값이 없으면, 가상 세션에서 OpenGL 애플리케이션을 실행할 수 없습니다. 콘솔 세션에서는 이 설정이 무시됩니다. — 버전 2017.0-4100 부터 사용 가능합니다.

input 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[input]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 `input` 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
enable-autorepeat	true 또는 false - DWORD(비트)	세션	Linux: true - Windows: 1	Linux에서 자동 반복 허용 여부 — 단일 키에 대해 자동 반복을 허용할지 여부를 지정합니다. — 버전 2017.2-6182 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
enable-gamepad	true 또는 false - DWORD(32- 비트)	세션	Linux: true - Windows: 1	게임패드 입력 허용 여부 — 게 임패드 활성화 여부를 지정합 니다. — 버전 2022.0-11954 부 터 사용 가능합니다.
enable-relative-mo use	true 또는 false - DWORD(32- 비트)	세션	Linux: true - Windows: 1	상대적 마우스 이동 허용 여부 — 상대적 마우스 이동을 허용 할지 여부를 지정합니다. — 버 전 2017.0-5121 부터 사용 가능 합니다.
enable-stylus	true 또는 false - DWORD(32- 비트)	세션	Linux: true - Windows: 1	스타일러스 입력 허용 여부 — 스타일러스를 활성화할지 여부를 지정합니다. — 버전 2019.0-7318 부터 사용 가능하 입니다.
enable-touch	true 또는 false - DWORD(32- 비트)	세션	Linux: true - Windows: 1	터치 입력 허용 여부 — 터치를 활성화할지 여부를 지정합니 다. — 버전 2017.3-6698 부터 사용 가능합니다.

license 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[license]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 `license` 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
license-file	문자열	세션	"	라이선스 - EC2가 아닌 인스턴스에서 DCV 서버를 실행할 때 사용할 라이선스를 지정합니다. 라이선스는 RLM 라이선스를 통해 부여됩니다. Windows에서는 ';'으로 구분되고 Linux에서는 ':'로 구분된 라이선스 사양 목록이 포함될 수 있습니다. 각 라이선스 사양은 확장 평가 라이선스의 경우 로컬 라이선스 파일이거나 부동 라이선스의 경우 PORT@HOSTNAME 형식으로 지정된 RLM 서버 포트 및 호스트 이름일 수 있습니다. 여러 라이선스가 지정된 경우 서버는 첫 번째 라이선스가 검증될 때까지 각 라이선스를 차례로 시도합니다(예: 라이선스 파일이 올바르게 인식되거나 원격 RLM 서버에 연결할 수 있음). 값을 지정하지 않으면 서버는 Linux에서는 기본 라이선스 파일인 '/usr/share/license/license.lic'를, Windows에서는 'C:\Program Files\NICE\DCV\Server\license\license.lic'를 찾습니다. 기본 라이선스 파일을 찾을 수 없는 경우 데모 라이선스를 사용합니다. EC2 인스턴스에서는 이 파라

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				미터가 무시됩니다. — 버전 2017.0-4100 부터 사용 가능합니다.

log 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[log]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 `log` 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
<code>directory</code>	문자열	서버	"	로그 출력 디렉터리 — 로그를 저장할 대상을 지정합니다. 지정하지 않으면 기본값은 Windows에서는 "C:\ProgramData\NICE\DCV\log", Linux에서는 "/var/log/dcv/"입니다. — 버전 2017.0-4100 부터 사용 가능합니다.
<code>enable-image-audit</code>	true 또는 false - DWORD(; 비트)	서버	Linux: false - Windows: 0	전송된 이미지의 내용 감사 활성화 — 전송된 이미지의 내용을 별도의 파일에 저장해야 하는지 여부를 지정합니다. 이미지는 로그 하위 디렉터리에 저장되며 파일 이름은 감사 CSV

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				파일에 보고됩니다. transfer-audit이 비활성화된 경우 값은 무시됩니다. — 버전 2023.0-14852 부터 사용 가능합니다.
level	문자열	사용자 지정	'info'	로그 수준 — 로그 파일 세부 수준을 지정합니다. 세부 수준 (세부 정보가 많은 순서로)에는 'error', 'warn', 'info', 'debug'가 있습니다. 새 값은 구성에서 변경되고 DCV 에이전트 프로세스로 전파되는 즉시 유효합니다. 2019.1 이전 버전에서는 에이전트 프로세스가 시작될 때만 DCV 에이전트 프로세스의 로그 수준이 설정됩니다. — 버전 2017.0-4100 부터 사용 가능합니다.
max-file-size	정수 - DWORD(비트)	서버	0	순환 전 최대 로그 파일 크기 (메가바이트) — 순환이 트리거되기 전에 최대 로그 파일 크기를 지정합니다. 값이 '0'이면 크기별 순환이 비활성화되고, 대신 파일을 생성하는 프로세스가 다시 시작될 때 파일이 순환됩니다. — 버전 2022.1-13067 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
rotate	정수 - DWORD(비트)	서버	10	로그 파일 순환 수수 — 로그 파 일이 제거되기 전에 순환하는 횟수를 지정합니다. 값이 0이 면, 이전 버전이 순환하지 않고 제거됩니다. — 버전 2017.0-41 00 부터 사용 가능합니다.
rotation-interval	문자열	서버	'none'	두 개의 연속 로그 파일 순환 사 이의 최대 시간 간격 — 두 개의 연속 로그 파일 순환 사이의 최 대 시간 간격을 지정합니다. 값 이 'none'인 경우 파일은 시간 에 따라 순환되지 않습니다. 가 능한 다른 값으로는 'every-mi nute', 'every-twenty-minutes', 'every-hour', 'every-day' 등이 있습니다. — 버전 2022.1-13 067 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
rotation-suffix	문자열	서버	'counter'	순환된 로그 파일에 추가할 접미사 — 순환된 로그 파일에 추가할 접미사를 지정합니다. 'counter'를 지정한 경우 순환된 각 로그 파일에 단순히 증가하는 카운터 접미사가 추가됩니다. 'timestamp'가 지정된 경우 'YYYY-MM-DD-HH-MM' 형식의 타임스탬프가 로그 파일에 적용됩니다. 해당 타임스탬프가 포함된 순환된 파일이 로그 폴더에 이미 있는 경우, 타임스탬프에 숫자 카운터가 더 추가됩니다. — 버전 2022.1-13067 부터 사용 가능합니다.
transfer-audit	문자열	서버	'none'	감사할 전송 방향 — 감사할 전송 방향을 지정합니다. 이 파라미터가 활성화되면, 새로운 CSV 파일 로그가 서버와 클라이언트 간에 전송됩니다. 허용되는 값은 'none', 'server-to-client', 'client-to-server', 'all'입니다. 이 값이 없거나 'none'에 해당되면, 전송 감사가 비활성화되고 파일이 생성되지 않습니다. — 버전 2017.0-4100 부터 사용 가능합니다.

printer 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[printer]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 printer 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
<code>file-printer-name</code>	문자열	사용자 지정	'DCV Printer'	파일 다운로드용 가상 DCV 프린터 이름 — DCV 서버의 가상 DCV 프린터 이름을 나타내는 문자열입니다. Linux에서는 새로운 Linux DCV 세션이 생성될 때마다 구성에서 이 값을 읽습니다. 이 설정이 비어 있지 않고 값으로 PREFIX 문자열을 포함하는 경우, 이름이 'PREFIX - 세션 번호'인 새 가상 프린터가 CUPS에 등록됩니다. 이 설정이 비어 있는 경우 DCV 가상 프린터는 등록되지 않습니다. Windows에서 이 설정은 시스템의 기본 프린터를 변경하는데 사용됩니다. 빈 문자열로 설정하면 DCV는 현재 기본 프린터를 변경하지 않습니다. — 버전 2022.0-11954 부터 사용 가능합니다.
<code>use-default-printer</code>	문자열	사용자 지정	'client-decides'	기본 프린터 설정 방법 결정 — 서버가 기본 프린터로 설정할 프린터를 결정합니다. 허용되는 값은 'client-decides', 'always-on', 'always-off'입니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				다. 이 설정의 값이 'always-off'인 경우 서버는 어떤 프린터도 기본 프린터로 설정하지 않습니다. 값이 'always-on'인 경우 'printer' 섹션의 'file-printer-name' 설정에 지정된 프린터를 설정합니다. 값이 'client-decides'인 경우 클라이언트에서 전송하는 기본 프린터가 설정됩니다. 클라이언트가 기본 프린터를 전송하지 않는 경우 'printer' 섹션의 'file-printer-name'에 지정된 프린터가 설정됩니다. 기본값은 'client-decides'입니다. 현재 Windows에서만 지원됩니다. — 버전 2022.2-13907 부터 사용 가능합니다.

redirection 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[redirection]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 `redirection` 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
enable-timezone-re direction	문자열	세션	'client-decides'	클라이언트에서 서버로의 시 간대 리디렉션 허용 또는 거 부 — 클라이언트에서 서버로 의 시간대 리디렉션을 허용하 거나 거부합니다. 허용되는 값 은 'always-on', 'always-off', 'client-decides'입니다. 'always- on'으로 설정된 경우 우선 순 위 사용자 클라이언트가 자신 의 시간대를 서버로 보내고 이 시간대가 서버의 시간대가 됩 니다. 'always-off'로 설정된 경 우 서버는 자체 시간대를 클라 이언트에 표시합니다. 모든 클 라이언트 시간대 메시지는 삭 제됩니다. 'client-decides'로 설 정된 경우 우선 순위 사용자 클라이언트가 자신의 시간대 를 서버로 보낼 수 있고 이 시 간대가 서버의 시간대가 됩니 다. 클라이언트는 자신의 시간 대를 서버로 전송하지 않도록 선택할 수 있습니다. 기본값은 'client-decides'입니다. — 버전 2022.2-13907 부터 사용 가능 합니다.

security 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[security]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 `security` 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
<code>allowed-http-host-regex</code>	문자열	서버	<code>'^.+\$\$'</code>	허용되는 호스트 정규식 — 이 DCV 서버가 제공할 수 있는 호스트 이름을 나타내는 정규식 패턴을 지정합니다. 수신되는 HTTP 요청의 호스트 헤더가 이 패턴과 일치하지 않는 경우, 요청 자체가 실패하여 상태 코드 403 Forbidden을 표시합니다. 이는 HTTP 호스트 헤더 공격을 차단하기 위한 보안 조치입니다. 패턴은 Javascript 부류의 유효한 정규식이어야 합니다. 패턴에 포함된 문자의 대문자/소문자 여부가 일치합니다. 예: <code>^(www\.)?example\.com\$</code> . — 버전 2017.0-4100 부터 사용 가능합니다.
<code>allowed-ws-origin-regex</code>	문자열	서버	<code> '^https://.+\$\$'</code>	허용되는 오리진 — 이 DCV 서버가 허용하는 오리진을 나타내는 정규식 패턴을 지정합니다. WebSocket 연결을 설정하는 경우, 클라이언트의 핸드셰이크의 오리진 헤더 필드는 연결을 설정하는 스크립트의 오리진을 나타냅니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				<p>니다. 수신되는 HTTP 요청의 오리진 헤더가 이 패턴과 일치하지 않는 경우, 요청 자체가 실패하여 상태 코드 403 Forbidden을 표시합니다. 이는 교차 사이트 WebSocket 탈취(CSWSH) 공격을 차단하기 위한 보안 조치입니다. 패턴은 Javascript 부류의 유효한 정규식이어야 합니다. 패턴에 포함된 문자의 대문자/소문자 여부가 일치합니다. 오리진 헤더는 다음과 같은 형식입니다. <scheme> "://" <host> [":" <port>]. 예: '^https://(www\.)?example\.com(:443)?\$'. — 버전 2017.0-4100부터 사용 가능합니다.</p>
auth-connection-setup-timeout	정수 - DWORD(; 비트)	서버	120	<p>인증 채널 연결 설정 제한 시간 — 인증 채널 연결 설정 절차를 완료하는 데 허용되는 제한 시간(초 단위)을 지정합니다. 절차 수행 도중 제한 시간을 넘기면, 채널이 닫힙니다. 0으로 설정된 경우, 인증 채널 연결 설정에 대한 제한 시간이 비활성화됩니다. — 버전 2017.0-4100부터 사용 가능합니다.</p>

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
auth-token-verifier	문자열	서버	"	인증 토큰 식별자의 엔드포인트 — DCV 서버에서 사용하는 인증 토큰 식별자의 엔드포인트(URL)를 지정합니다. 비어 있는 경우, 내부 인증 토큰 식별자가 사용됩니다. — 버전 2017.0-4100 부터 사용 가능합니다.
auth-token-verifier-timeout	정수 - DWORD(; 비트)	서버	100	인증 토큰 식별자의 제한 시간(초 단위) — DCV 서버에서 사용하는 인증 토큰 식별자를 기다리는 시간(초 단위)을 지정합니다. — 버전 2023.0-14852 부터 사용 가능합니다.
authentication	문자열	서버	'system'	인증 방법 — DCV 서버에서 사용하는 클라이언트 인증 방법을 지정합니다. 'system'을 사용하여 클라이언트 인증을 기본 운영 체제에 위임합니다. 'none'을 사용하여 클라이언트 인증을 비활성화하고 모든 클라이언트에 대한 액세스를 허용합니다. — 버전 2017.0-4100 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
authentication-threshold	정수 - DWORD(; 비트)	서버	3	인증 임계값 — 서버에서 연결을 종료하기 전에 각 클라이언트에 허용되는 인증 실패 횟수를 지정합니다. 인증 시도를 무제한 허용하려면 0으로 설정합니다. — 버전 2017.0-4100 부터 사용 가능합니다.
ca-file	문자열	서버	"	CA 파일 — DCV 서버에서 신뢰하는 인증 기관(CA) 정보가 담긴 파일을 지정합니다. 비어 있는 경우, 시스템에서 제공하는 기본 트러스트 스토어를 사용합니다. — 버전 2017.0-4100 부터 사용 가능합니다.
certificate-to-user-file	문자열	사용자 지정	"	사용자 매핑 파일 인증서 - 인증서가 포함된 파일을 사용자 매핑 목록에 지정합니다. — 버전 2022.0-11954 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
ciphers	문자열	서버	'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384'	TLS 연결에 사용되는 암호 목록 — TLS 연결에 사용되는 암호 목록을 지정합니다. 암호 목록은 ":" 문자를 사용하여 구분되어야 하며 openssl과 클라이언트의 지원이 있어야 합니다. — 버전 2017.0-4100 부터 사용 가능합니다.
connection-estab-timeout	정수 - DWORD(; 비트)	서버	5	연결 설정 제한 시간 — 연결 절차를 완료하는 데 허용되는 제한 시간(초 단위)을 지정합니다. 절차 수행 도중 제한 시간을 넘기면, 연결이 닫힙니다. 0으로 설정된 경우, 연결 설정에 대한 제한 시간이 없습니다. — 버전 2017.0-4100 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
connection-setup-timeout	정수 - DWORD(비트)	서버	5	채널 연결 설정 제한 시간 — 채널 연결 설정 절차를 완료하는데 허용되는 제한 시간(초 단위)을 지정합니다. 절차 수행 도중 제한 시간을 넘기면, 채널이 닫힙니다. 0으로 설정된 경우, 채널 연결 설정에 대한 제한 시간이 없습니다. — 버전 2017.0-4100 부터 사용 가능합니다.
crl-file	문자열	사용자 지정	"	CRL 파일 — 인증서 취소 목록 (CRL)이 포함된 파일을 지정합니다. — 버전 2022.0-11954 부터 사용 가능합니다.
enable-gssapi	true 또는 false - DWORD(비트)	서버	Linux: false - Windows: 0	GSSAPI SASL 메커니즘 활성화 — Kerberos를 사용하여 DCV 인증을 허용하는 GSSAPI SASL 메커니즘을 활성화 또는 비활성화합니다. — 버전 2017.3-6698 부터 사용 가능합니다.
max-connections-per-user	정수 - DWORD(비트)	서버	10	사용자의 최대 연결 수 — 사용자당 허용되는 최대 동시 연결 수를 지정합니다. 그 이상의 연결은 거부됩니다. — 버전 2017.0-4100 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
no-tls-strict	true 또는 false - DWORD(32 비트)	서버	Linux: false - Windows: 0	엄격한 인증서 검증 비활성화 — 외부 인증 토큰 식별자에 연결할 때 엄격한 인증서 검증 을 활성화하거나 비활성화합 니다. 인증 토큰 식별자가 자 체 서명된 인증서를 사용하는 경우 엄격한 인증서 검증을 비 활성화해야 합니다. — 버전 2017.0-4100 부터 사용 가능합 니다.
os-auto-lock	true 또는 false - DWORD(32 비트)	세션	Linux: true - Windows: 1	마지막 클라이언트 연결이 종 료될 때 OS 세션을 잠글지 여부 — 활성화하면 마지막 클라이언트 연결이 닫힐 때 OS 세션이 잠깁니다. — 버전 2017.1-5777 부터 사용 가능합 니다.
pam-service-name	문자열	서버	'dcv'	PAM 서비스 이름 — DCV에 서 사용하는 PAM 구성 파일 의 이름을 지정합니다. 기본적 인 PAM 서비스 이름은 'dcv'이 며, /etc/pam.d/dcv 구성 파일 에 해당합니다. 이 파라미터는 'system' 인증 방법이 적용되는 경우에만 사용됩니다. — 버전 2017.0-4100 부터 사용 가능합 니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
passwd-file	문자열	서버	"	암호 파일 — 사용자 보안 인증 정보를 확인하는 데 사용할 암호 파일을 지정합니다 (dcv 인증 모드에만 해당). 비어 있다면, Windows의 경우 <code>\${XDG_CONFIG_HOME}/NICE/dcv/passwd</code> for Linux, or <code>%CSIDL_LOCAL_APPDATA%\NICE\dcv\passwd</code> 의 기본 파일을 사용합니다. — 버전 2017.0-4100 부터 사용 가능합니다.
server-fqdn	문자열	서버	"	서버 FQDN — 서버의 정규화된 도메인 이름을 지정합니다. 비어 있으면 <code>gethostname()</code> 입니다. — 버전 2017.3-6698 부터 사용 가능합니다.
service-name	문자열	서버	'dcv'	서비스 이름 — 서비스의 등록된 이름(일반적으로 프로토콜 이름)입니다. — 버전 2020.0-8428 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
supervision-control	문자열	사용자 지정	'disabled'	세션의 감독 제어 유형 — 세션의 감독 제어 유형을 지정합니다. 가능한 값은 'disabled' 및 'enforced'입니다. 이 값을 'enforced'로 설정하면 감독되지 않은 액세스 권한을 설정하여 공동 작업 세션에서 사용자의 소유자 없는 액세스를 허용하거나 거부할 수 있습니다. 사용자에게 감독되지 않은 액세스가 허용된 경우 사용자는 소유자 없이 세션에 액세스할 수 있습니다. 소유자를 제외한 모든 사용자에게는 기본적으로 이 권한이 거부됩니다. 이 값을 'disabled'(기본값)로 설정하면 서버는 이 감독 제어 및 권한을 적용하지 않습니다. 새 값은 구성에서 변경되는 즉시 적용됩니다. — 버전 2021.3-11591 부터 사용 가능합니다.
user-realm	문자열	서버	"	서버 사용자 영역 — 서버에 대한 사용자 영역을 지정합니다. — 버전 2017.3-6698 부터 사용 가능합니다.

session-management 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[session-management]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 `session-management` 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
<code>create-session</code>	true 또는 false - DWORD(; 비트)	서버	Linux: false - Windows: 0	서버 시작 시 콘솔 세션 생성 — 서버 시작 시 (ID가 "console" 인) 콘솔 세션을 자동으로 생성할지 여부를 지정합니다. — 버전 2017.0-4100 부터 사용 가능합니다.
<code>enable-gi-in-virtual-sessions</code>	문자열	세션	'default-on'	<code>dcv-gi</code> 기능의 사용 여부 — <code>dcv-gi</code> 기능의 사용 여부를 지정합니다(라이선스가 필요함). 허용된 값은 'always-on', 'always-off', 'default-on', 'default-off'입니다. — 버전 2017.0-4100 부터 사용 가능합니다.
<code>max-concurrent-clients</code>	정수 - DWORD(; 비트)	세션	-1	세션당 최대 동시 클라이언트 수 — 세션당 최대 동시 클라이언트 수를 지정합니다. -1로 설정하면 어떠한 제한도 적용되지 않습니다. 자동 세션에만 제한을 설정하려면, 'session-management/automatic-console-session' 섹션의 'max-concurrent-clients'를 사용합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				— 버전 2017.0-4100 부터 사용 가능합니다.
max-concurrent-sessions	정수 - DWORD(비트)	서버	0	최대 동시 세션 수 — 허용되는 최대 동시 세션 수를 지정합니다. 콘솔 세션은 본질적으로 하나로 제한되므로 이 제한은 현재 가상 세션에만 적용됩니다. 제한을 적용하지 않으려면 0을 지정합니다. — 버전 2019.0-7318 부터 사용 가능합니다.
max-sessions-per-user	정수 - DWORD(비트)	서버	0	사용자당 최대 세션 수 — 각 사용자가 소유할 수 있는 최대 허용 동시 세션 수를 지정합니다. 이 제한은 현재 가상 세션에만 적용됩니다. 제한을 적용하지 않으려면 0을 지정합니다. — 버전 2021.0-10242 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
virtual-session-default-layout	문자열	세션	[]	<p>가상 세션의 기본 레이아웃</p> <p>— 이 설정이 설정되어 있으면 시작 시 지정된 레이아웃을 만들도록 Xdcv가 구성됩니다. 각 모니터를 해상도(w,h)와 위치(x,y)로 구성할 수 있습니다. 모든 지정된 모니터가 활성화됩니다. 기본 레이아웃 예제 값: [{'w':<800>, 'h':<600>, 'x':<0>, 'y': <0>}, {'w':<1024>, 'h':<768>, 'x':<800>, 'y':<0>}]</p> <p>이 설정의 경우 최대 모니터 수 (virtual-session-monitors 설정)는 어레이의 요소 수보다 우선 순위가 높습니다. 예를 들어 모니터 다섯 개가 설정되었지만 최대 모니터 수가 네 개이면 처음 네 개의 모니터만 만들어집니다. 이 키를 설정하면 virtual-session-monitors 설정에 지정된 활성화된 모니터 수가 무시됩니다. — 버전 2017.0-5600부터 사용 가능합니다.</p>

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
virtual-session-font-path	문자열	세션	"	특수 폰트 경로의 추가 여부 — 특수 폰트의 경로를 지정합니다. 일부 애플리케이션의 경우 X 서버로 전달하기 위해서는 특수 폰트가 필요합니다. — 버전 2017.0-4100 부터 사용 가능합니다.
virtual-session-source-profile	true 또는 false - DWORD(32비트)	세션	Linux: false - Windows: 0	세션 스타터에서 사용자 프로파일을 소싱할지 여부 — 세션 스타터 스크립트를 실행하는 셸이 사용자 프로파일을 소싱해야 하는지 여부를 지정합니다. 기본적으로 이 값은 false이며 DCV는 버전 2021.3-11591 부터 사용 가능한 "bash --noprofile --norc"를 사용하여 세션 스타터 스크립트를 실행합니다.
virtual-session-xdcv-args	문자열	세션	"	Xdcv에 전달할 추가 인수 — Xdcv에 전달할 모든 추가 인수를 지정합니다. — 버전 2017.0-4334 부터 사용 가능합니다.

session-management/automatic-console-session 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[session-management/automatic-console-session]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 `session-management/automatic-console-session` 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
<code>client-eviction-policy</code>	문자열	서버	'reject-new-connection'	한도에 도달했을 때 클라이언트 연결을 처리하는 방법 지정 — 세션당 최대 동시 클라이언트 수에 도달했을 때 새 연결을 거부할지 아니면 기존 연결을 자동으로 종료할지를 지정합니다. 허용되는 값은 'reject-new-connection'(들어오는 연결이 종료됨) 및 'same-user-oldest-connection'(세션과 가장 오랫동안 상호 작용하지 않았거나, 해당 정보가 없는 경우 가장 오래된 연결 시간을 가진 동일한 사용자의 연결을 서버에서 종료함)입니다. — 버전 2022.1-13067 부터 사용 가능합니다.
<code>max-concurrent-clients</code>	정수 - DWORD(비트)	서버	-1	세션당 최대 동시 클라이언트 수 — 세션당 허용되는 최대 동시 클라이언트 수를 지정합니다. -1로 설정하면 어떠한 제한도 적용되지 않습니다. — 버전

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				2017.0-5600 부터 사용 가능합니다.
owner	문자열	서버	"	자동 생성된 "콘솔" 세션의 소유자 — "콘솔" 세션 소유자의 사용자 이름을 지정합니다. 비어 있는 경우, 소유자는 DCV 서버를 시작한 사용자입니다. 이 설정은 create-session 설정이 true로 설정된 경우에 한하여 서버 시작 시 자동 생성된 "콘솔" 세션에만 적용됩니다. — 버전 2017.0-5600 부터 사용 가능합니다.
permissions-file	문자열	서버	"	자동 "콘솔" 세션에 대한 권한 파일 — DCV 기능에 대한 사용자 액세스를 확인하기 위해 사용될 권한 파일의 경로를 지정합니다. 비어 있는 경우, 소유자만이 세션에 대한 모든 액세스가 가능합니다. — 버전 2017.0-5600 부터 사용 가능합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
storage-root	문자열	서버	"	파일 스토리지 루트 폴더의 경 로 — 콘솔 세션 스토리지에 사 용할 폴더의 전체 경로를 지정 합니다. storage-root가 비어 있 거나 폴더가 없는 경우, 파일 스 토리지가 비활성화됩니다. — 버전 2017.0-5600 부터 사용 가 능합니다.

session-management/defaults 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[session-management/defaults]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 `session-management/defaults` 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
permissions-file	문자열	세션	"	모든 세션에 포함된 기본 권 한 — 각 세션에 대해 사용자 가 선택한 권한과 자동으로 병 합될 권한 파일의 경로를 지정 합니다. 비어 있는 경우, Linux 의 경우 <code>/etc/dcv/</code> 에 위치하고 Windows의 경우 DCV 설치 폴 더(예: <code>'C:\Program Files\NICE \DCV\Server\conf'</code>)에 위치한

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				'default.perm' 파일을 사용합니다. — 버전 2017.0-5600 부터 사용 가능합니다.

smartcard 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[smartcard]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 smartcard 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
enable-cache	문자열	사용자 지정	'default-on'	스마트카드 캐싱 메시지 활성화 여부 — 스마트 카드 캐싱을 활성화하거나 비활성화합니다. 사용하도록 설정하면 NICE DCV 서버가 클라이언트의 스마트 카드에서 마지막으로 받은 값을 캐시합니다. 향후 호출은 클라이언트가 아니라 서버의 캐시에서 직접 검색됩니다. 이렇게 하면 클라이언트와 서버 간에 전송되는 트래픽 양을 감소시켜 성능을 개선할 수 있습니다. 허용되는 값에는 'always-on', 'always-off', 'default-on', and 'default-off'가

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				있습니다. 이 값은 클라이언트 스마트 카드 애플리케이션이 시작될 때마다 구성에서 읽습니다. — 버전 2017.2-6182 부터 사용 가능합니다.

webcam 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[webcam]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 webcam 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
<code>max-resolution</code>	문자열	ODBC	(1280, 720)	최대 웹캠 해상도 — 애플리케이션에 표시되는 최대 웹캠 해상도를 지정합니다. — 버전 2021.0-10242 부터 사용 가능합니다.
<code>preferred-resolution</code>	문자열	ODBC	(480, 360)	기본 웹캠 해상도 — 클라이언트가 제공하는 해상도 중에서 선호하는 웹캠 해상도를 지정합니다. 지정된 해상도가 지원되지 않는 경우 가장 근접하게 일치하는 해상도가 선택되어 애플리케이션에 표시됩니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				지정된 값 중 하나가 0이면 웹캠 공유가 비활성화됩니다. — 버전 2021.0-10242 부터 사용 가능합니다.

windows 파라미터

다음 표에서는 Linux NICE DCV 서버용 `/etc/dcv/dcv.conf` 파일의 `[windows]` 섹션에 대한 구성 파라미터와 Windows NICE DCV 서버용 windows 레지스트리 키를 설명합니다.

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
<code>disable-display-sleep</code>	true 또는 false - DWORD(비트)	세션	Linux: true - Windows: 1	디스플레이의 저전력 모드 전환 차단 — 디스플레이의 저전력 모드 전환을 차단할지 여부를 지정합니다. — 버전 2017.0-4100 부터 사용 가능합니다.
<code>printer</code>	문자열	세션	"	기본값으로 설정해야 할 프린터 — 가상 DCV 프린터의 이름을 지정합니다. 이 이름은 시스템의 기본 프린터를 변경하는데 사용됩니다. 빈 문자열로 설정하면 DCV는 현재 기본 프린터를 변경하지 않습니다. 지원

파라미터	유형 - Windows 레지스 트리 유 형	컨텍스 트 다시 로드	기본값	설명
				중단됨: 'printer' 섹션의 'file-printer-name'을 사용하세요. — 버전 2017.0-4100 부터 사용 가능합니다.

구성 파라미터 수정

이 섹션에서는 NICE DCV 서버에 대한 구성 파라미터를 수정하는 방법을 설명합니다. Windows 서버용 레지스트리 키, Linux 서버에 대한 단원, 파라미터 이름, 유형 및 유효한 값에 대한 자세한 내용은 [NICE DCV 서버 파라미터 참조](#) 섹션을 참조하세요.

주제

- [Windows NICE DCV 서버](#)
- [Linux NICE DCV 서버](#)

Windows NICE DCV 서버

Windows NICE DCV 서버의 경우, Windows 레지스트리 편집기, PowerShell 또는 명령줄을 사용하여 구성 파라미터를 수정합니다.

Windows 레지스트리 편집기를 사용하여 구성 파라미터를 수정하려면

1. Windows 레지스트리 편집기를 엽니다.
2. 다음 레지스트리 경로로 이동합니다.

```
HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/
```

3. 파라미터가 있는 레지스트리 키를 선택합니다. 레지스트리 키가 없는 경우 [NICE DCV 서버 파라미터 참조](#)에 나와 있는 정확한 키 이름을 사용하여 레지스트리 키를 생성합니다.
4. 해당 파라미터를 엽니다(두 번 클릭). 해당 파라미터가 없는 경우 [NICE DCV 서버 파라미터 참조](#)에 나와 있는 유형 및 이름을 사용하여 추가합니다.

PowerShell을 사용하여 구성 파라미터를 수정하려면

1. PowerShell을 관리자로 실행합니다.
2. [NICE DCV 서버 파라미터 참조](#)에 나와 있는 키 이름을 사용하여 레지스트리 키를 추가합니다.

```
PS C:\> New-Item -Path "Microsoft.PowerShell.Core\Registry::\HKEY_USERS
\S-1-5-18\Software\GSettings\com\nicesoftware\dcv\" -Name registry_key -Force
```

3. [NICE DCV 서버 파라미터 참조](#)에 나와 있는 유형 및 이름을 사용하여 레지스트리 키에 파라미터를 생성합니다.

```
PS C:\> New-ItemProperty -Path "Microsoft.PowerShell.Core\Registry::
\HKEY_USERS\S-1-5-18\Software\GSettings\com\nicesoftware\dcv\registry_key" -
Name parameter_name -PropertyType parameter_type -Value parameter_value -Force
```

명령줄을 사용하여 구성을 수정하려면

1. 명령줄을 관리자로서 실행합니다.
2. [NICE DCV 서버 파라미터 참조](#)에 나와 있는 키 이름, 파라미터 유형 및 이름을 사용하여 레지스트리 키를 생성하고 파라미터를 추가합니다.

```
C:\> reg.exe ADD "HKEY_USERS\S-1-5-18\Software\GSettings\com\nicesoftware\dcv
\registry_key" /v parameter_name /t parameter_type /d parameter_value /f
```

Linux NICE DCV 서버

Linux NICE DCV의 경우, 텍스트 편집기 또는 crudini 등 명령줄 도구를 사용하여 구성 파라미터를 수정할 수 있습니다.

텍스트 편집기를 사용하여 구성 파라미터를 수정하려면

1. 원하는 텍스트 편집기에서 /etc/dcv/dcv.conf 파일을 엽니다.
2. 파일에서 해당 섹션을 찾습니다. 해당 섹션이 존재하지 않는 경우 [NICE DCV 서버 파라미터 참조](#)에 나와 있는 섹션 이름을 사용하여 추가합니다.

```
[section]
```

3. 해당 섹션에서 해당 파라미터를 찾아 값을 수정합니다. 해당 섹션에 해당 파라미터가 존재하지 않는 경우 [NICE DCV 서버 파라미터 참조](#)에 나와 있는 파라미터 이름을 사용하여 추가합니다.

```
parameter_name="parameter_value"
```

4. 파일을 저장하고 닫습니다.

crudini를 사용하여 구성 파라미터를 수정하려면

섹션을 생성하고 [NICE DCV 서버 파라미터 참조](#)에 나와 있는 섹션 및 파라미터 이름을 사용하여 파라미터를 추가합니다.

```
$ sudo crudini --set /etc/dcv/dcv.conf section_name parameter_name 'parameter_value'
```

NICE DCV 지원 기간 종료

NICE DCV 지원 기간 종료(EOSL)는 NICE DCV의 특정 메이저 버전(및 모든 마이너 버전)이 더 이상 지원되지 않고 새 버전과의 호환성 테스트가 더 이상 진행되지 않는 시점을 정의합니다.

EOSL 날짜 이전까지 NICE DCV 지원팀에서 구성 문제에 대한 모든 지원을 계속 제공합니다. 결함 해결 및 기능 요청은 최신 버전의 NICE DCV 서버 및 NICE DCV 클라이언트에만 구현됩니다. 이전 버전에는 구현되지 않습니다.

EOSL 날짜 이후에는 추가 지원이나 유지 관리가 제공되지 않습니다. 또한 호환성 문제에 대한 테스트도 중단할 예정입니다. 지속적인 지원을 받으려면 최신 NICE DCV 버전으로 업그레이드해야 합니다.

주제

- [EOSL 타임라인](#)
- [고객을 위한 EOSL 경로](#)
- [EOSL FAQ](#)

EOSL 타임라인

다음 표는 NICE DCV 메이저 버전에 대한 EOSL 타임라인을 보여줍니다.

NICE DCV 메이저 버전	최초 릴리스 날짜	EOSL 날짜
NICE DCV 2016.x	2015년 12월 31일	2021년 3월 31일
NICE DCV 2017.x	2017년 12월 18일	2021년 12월 31일
NICE DCV 2019.x	2019년 8월 5일	2022년 12월 31일
NICE DCV 2020.x	2020년 4월 16일	2023년 12월 31일

NICE DCV 메이저 버전	최초 릴리스 날짜	EOSL 날짜
NICE DCV 2021.x	2021년 4월 12일	2024년 12월 31일
NICE DCV 2022.x	2022년 2월 23일	2025년 12월 31일
NICE DCV 2023.x	2023년 5월 3일	2026년 12월 31일

고객을 위한 EOSL 경로

AWS에서 NICE DCV를 실행하는 경우 NICE DCV에 대한 라이선스가 필요하지 않습니다. 워크로드에 사용하는 기본 AWS 리소스에 대해서만 비용을 지불합니다. EOSL 날짜가 지난 NICE DCV 버전을 현재 사용 중인 경우, [NICE 다운로드 페이지](#)를 사용하여 최신 NICE DCV 버전으로 업그레이드하거나 AWS Marketplace에서 [NICE DCV AMI](#)를 사용하여 계속 지원을 받을 수 있습니다.

NICE DCV를 온프레미스로 실행 중이거나 서드 파티 클라우드 서비스 제공업체를 이용 중인 상황에서 현재 사용 중인 NICE DCV 버전이 EOSL 날짜가 지난 경우, 리셀러 또는 배포자 측에 문의하여 이용 가능한 업그레이드 경로를 평가하세요. 유효한 지원 계약이 있는 경우 무료로 최신 버전의 NICE DCV로 업그레이드할 수 있습니다. NICE DCV 배포자 및 리셀러에 대한 자세한 내용은 [NICE 웹사이트](#)를 참조하세요.

EOSL FAQ

1. 온프레미스 또는 서드 파티 클라우드 서비스 제공업체를 통해 EOSL에 도달한 NICE DCV 버전을 사용하고 있지만 기존 지원 계약이 있습니다. EOSL의 영향을 받게 되나요?

유효한 지원 계약이 있는 경우 NICE DCV 지원 계약 약관에 따라 추가 비용 없이 NICE DCV 라이선스를 최신 버전으로 업그레이드할 수 있습니다. 질문하신 상황에서는 영향이 최소화됩니다. 지원 계약이 만료된 경우, 계속해서 모든 지원을 받으려면 다음 방법 중 하나를 사용하세요.

1. 새로운 유료 라이선스로 NICE DCV를 최신 버전으로 업그레이드하세요.
2. EOSL 타임라인이 시작되기 전에 지원 계약을 갱신하세요. 그러면 NICE DCV의 최신 버전으로 업그레이드할 수 있습니다.

3. 지원 계약이 만료된 이후의 기간 동안, 지원 서비스에 대한 현재 요금의 70% 에 해당하는 복원 수수료를 지불하시면 기존 지원 계약을 복원할 수 있습니다.

2. Amazon EC2에서 EOSL에 도달한 NICE DCV 버전을 사용하고 있습니다. 지원되는 버전으로 업그레이드하려면 어떻게 해야 하나요?

언제든지 추가 비용 없이 NICE DCV를 완전히 지원되는 버전으로 업그레이드하여 Amazon EC2에서 사용할 수 있습니다.

3. 지원되는 NICE DCV 서버가 있는 상태에서 EOSL에 도달한 NICE DCV 클라이언트 버전을 사용할 수 있나요? 또, 그 반대의 경우도 가능한가요?

예, 하지만 EOSL에 도달한 버전에는 버그 수정이 더 이상 적용되지 않으므로 클라이언트와 서버 소프트웨어를 모두 최신 버전으로 업그레이드하는 것을 적극 권장합니다.

보안

AWS에서 클라우드 보안을 가장 중요하게 생각합니다. AWS 고객은 보안에 매우 민감한 조직의 요구 사항에 부합하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 AWS와 귀하의 공동 책임입니다. [공동 책임 모델](#)은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안 - AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호합니다. AWS는 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 [AWS 규정 준수 프로그램](#)의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. NICE DCV에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 [규정 준수 프로그램의 범위에 속하는 AWS 서비스](#)를 참조하세요.
- 클라우드 내 보안 - 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 NICE DCV 사용 시 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표를 충족하도록 NICE DCV를 구성하는 방법을 보여줍니다. 또한 NICE DCV 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

목차

- [NICE DCV의 데이터 보호](#)
- [NICE DCV의 규정 준수 검증](#)

NICE DCV의 데이터 보호

AWS [공동 책임 모델](#)은 NICE DCV의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 이 인프라에서 호스팅되는 콘텐츠에 대한 제어를 유지하는 것은 사용자의 책임입니다. 이 콘텐츠에는 사용하는 AWS 서비스 서비스의 보안 구성과 관리 작업이 포함돼 있습니다. 데이터 프라이버시에 대한 자세한 내용은 [데이터 프라이버시 FAQ](#)를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 [AWS 공동 책임 모델 및 GDPR](#) 블로그 게시물을 참조하세요.

데이터를 보호하려면 AWS 계정 보안 인증 정보를 보호하고 AWS IAM Identity Center 또는 AWS Identity and Access Management(IAM)를 통해 개별 사용자 계정을 설정하는 것이 좋습니다. 이러한 방

식에는 각 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정합니다.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용합니다.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-2 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 [Federal Information Processing Standard\(FIPS\) 140-2](#)를 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 NICE DCV 또는 기타 AWS 서비스에서 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명 정보를 URL에 포함시켜서는 안 됩니다.

데이터 암호화

보안 서비스의 주요 특징은 정보가 활발히 사용되지 않을 때 암호화된다는 것입니다.

유휴 시 암호화

NICE DCV는 자체적으로 고객 데이터를 저장하지 않습니다. NICE DCV 서버 호스트의 데이터는 저장 시 암호화될 수 있습니다. AWS에서 NICE DCV를 사용하는 경우 Linux 인스턴스용 Amazon EC2 사용 설명서의 [저장 시 암호화](#) 섹션 및 Windows 인스턴스용 Amazon EC2 사용 설명서의 [저장 시 암호화](#) 섹션을 참조하세요.

전송 중 데이터 암호화

NICE DCV 클라이언트 및 NICE DCV 서버에서 전송되는 모든 데이터는 HTTPS/TLS 연결을 통해 모든 항목을 전송하여 암호화됩니다.

인증서를 구성하려면 [TLS 인증서 관리](#) 섹션을 참조하세요.

NICE DCV의 규정 준수 검증

타사 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 AWS 서비스의 보안 및 규정 준수를 평가합니다. NICE DCV를 사용하여 서비스에 액세스하는 경우에도 해당 서비스의 규정 준수는 변경되지 않습니다.

특정 규정 준수 프로그램의 범위 내에 있는 AWS 서비스 목록은 [규정 준수 프로그램 제공 범위 내 AWS 서비스](#)를 참조하세요. 일반적인 내용은 [AWS 규정 준수 프로그램](#)을 참조하세요.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. AWS Artifact 자세한 정보는 [AWS Artifact에서 보고서 다운로드](#)를 참조하세요.

NICE DCV 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다. AWS에서는 규정 준수를 지원할 다음과 같은 리소스를 제공합니다.

- [보안 및 규정 준수 빠른 시작 안내서](#) - 이 배포 안내서에서는 아키텍처 고려 사항에 관해 설명하고 AWS에서 보안 및 규정 준수에 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다.
- [AWS 규정 준수 리소스](#) - 고객 조직이 속한 산업 및 위치에 적용될 수 있는 워크북 및 가이드 모음입니다.
- AWS Config 개발자 가이드의 [규칙을 사용하여 리소스 평가](#) - AWS Config 서비스는 내부 사례, 산업 지침 및 규제에 대한 리소스 구성의 준수 상태를 평가합니다.
- [AWS Security Hub](#) - 이 AWS 서비스는 보안 산업 표준 및 모범 사례 규정 준수 여부를 확인하는 데 도움이 되도록 AWS 내 보안 상태를 종합적으로 보여줍니다.

NICE DCV에 대한 릴리스 정보 및 문서 기록

이 페이지에서는 NICE DCV에 대한 릴리스 정보와 문서 기록을 제공합니다.

주제

- [NICE DCV 릴리스 정보](#)
- [사용 설명서 기록](#)

NICE DCV 릴리스 정보

이 섹션에서는 NICE DCV의 주요 업데이트, 기능 릴리스 정보 및 버그 수정에 대한 개요를 제공합니다. 모든 업데이트는 릴리스 데이터별로 정리되어 있습니다. 사용자로부터 받은 의견을 수렴하기 위해 설명서가 자주 업데이트됩니다.

주제

- [DCV 2023.1-16388 — 2024년 3월 5일](#)
- [DCV 2023.1-16388 - 2023년 12월 19일](#)
- [DCV 2023.1-16220 — 2023년 11월 9일](#)
- [DCV 2023.0-15487 — 2023년 6월 29일](#)
- [DCV 2023.0-15065 — 2023년 5월 3일](#)
- [DCV 2023.0-15022 — 2023년 4월 21일](#)
- [DCV 2023.0-14852 — 2023년 3월 28일](#)
- [DCV 2022.2-14521 — 2023년 2월 17일](#)
- [DV 2022.2-14357 — 2023년 1월 18일](#)
- [DCV 2022.2-14175 — 2022년 12월 21일](#)
- [DCV 2022.2-14126 — 2022년 12월 9일](#)
- [DCV 2022.2-13907 — 2022년 11월 11일](#)
- [DCV 2022.1-13300 — 2022년 8월 4일](#)
- [DCV 2022.1-13216 — 2022년 7월 21일](#)
- [DCV 2022.1-13067 — 2022년 6월 29일](#)
- [DCV 2022.0-12760 — 2022년 5월 23일](#)

- [DCV 2022.0-12627 — 2022년 5월 19일](#)
- [DCV 2022.0-12123 — 2022년 3월 23일](#)
- [DCV 2022.0-11954 — 2022년 2월 23일](#)
- [DCV 2021.3-11591 — 2021년 12월 20일](#)
- [DCV 2021.2-11445 — 2021년 11월 18일](#)
- [DCV 2021.2-11190 — 2021년 10월 11일](#)
- [DCV 2021.2-11135 — 2021년 9월 24일](#)
- [DCV 2021.2-11048 — 2021년 9월 1일](#)
- [DCV 2021.1-10851 — 2021년 7월 30일](#)
- [DCV 2021.1-10598 — 2021년 6월 10일](#)
- [DCV 2021.1-10557 — 2021년 5월 31일](#)
- [DCV 2021.0-10242 — 2021년 4월 12일](#)
- [DCV 2020.2-9662 — 2020년 12월 4일](#)
- [DCV 2020.2-9508 — 2020년 11월 11일](#)
- [DCV 2020.1-9012 — 2020년 9월 30일](#)
- [DCV 2020.1-9012 — 2020년 8월 24일](#)
- [DCV 2020.1-8942 — 2020년 8월 3일](#)
- [DCV 2020.0-8428 — 2020년 4월 16일](#)
- [DCV 2019.1-7644 — 2019년 10월 24일](#)
- [DCV 2019.1-7423 — 2019년 9월 10일](#)
- [DCV 2019.0-7318 — 2019년 8월 5일](#)
- [DCV 2017.4-6898 — 2019년 4월 16일](#)
- [DCV 2017.3-6698 — 2019년 2월 24일](#)
- [DCV 2017.2-6182 — 2018년 10월 8일](#)
- [DCV 2017.1-5870 — 2018년 8월 6일](#)
- [DCV 2017.1-5777 — 2018년 6월 29일](#)
- [DCV 2017.0-5600 — 2018년 6월 4일](#)
- [DCV 2017.0-5121 — 2018년 3월 18일](#)
- [DCV 2017.0-4334 — 2018년 1월 24일](#)
- [DCV 2017.0-4100 — 2017년 12월 18일](#)

DCV 2023.1-16388 — 2024년 3월 5일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 16388 	<ul style="list-style-type: none"> 디스플레이 스케일링이 다른 값으로 설정된 경우 Windows Client의 확장 프로그램 관련 문제가 수정되었습니다. 100%
<ul style="list-style-type: none"> nice-dcv-client(윈도우): 8993 	<ul style="list-style-type: none"> Windows 클라이언트의 상대 마우스 모드 및 High DPI 마우스 관련 문제를 수정했습니다. Windows 클라이언트에서 Shift 키를 사용하여 키보드 조합을 릴리스할 때 발생하는 문제를 수정했습니다.
<ul style="list-style-type: none"> nice-dcv-viewer (맥 OS): 6203 	
<ul style="list-style-type: none"> nice-dcv-viewer (리눅스): 6203 	
<ul style="list-style-type: none"> nice-xdcv: 565 	
<ul style="list-style-type: none"> nice-dcv-gl: 1047 	
<ul style="list-style-type: none"> nice-dcv-gltest: 325 	

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-simple-ex-ternal-인증자: 228 	

DCV 2023.1-16388 - 2023년 12월 19일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 16388 	<ul style="list-style-type: none"> Windows에서 에이전트 시작 시 스트리밍 실패 및 과도한 로깅을 일으킬 수 있는 경합 문제를 해결했습니다. 런타임 시 유휴 제한 시간 설정이 변경될 때 dcV list-connections 에 보고되는 마지막 상호 작용 시간을 수정했습니다.
<ul style="list-style-type: none"> nice-dcv-client(윈도우): 8934 	<ul style="list-style-type: none"> Windows 서버에서 NVIDIA GRID 드라이버 528.89와의 호환성 문제를 해결했습니다. 스트리밍 실패로 이어질 수 있는 웹 클라이언트의 비디오 디코딩 문제를 해결했습니다.
<ul style="list-style-type: none"> nice-dcv-viewer (맥 OS): 6203 	<ul style="list-style-type: none"> 서버에서 디스플레이 해상도 변경이 비활성화된 경우 Windows 클라이언트에서 여러 모니터에 전체 화면이 표시되는 문제를 해결했습니다.
<ul style="list-style-type: none"> nice-dcv-viewer (리눅스): 6203 	<ul style="list-style-type: none"> Linux 및 macOS 클라이언트의 웹캠 해상도 문제를 해결했습니다. Linux 및 macOS 클라이언트에서 마우스를 두 번, 세 번 클릭할 때 발생하는 문제를 해결했습니다.

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-xdcv: 565 nice-dcv-gl: 1047 nice-dcv-gltest: 325 nice-dcv-simple-external-인증자: 228 	<ul style="list-style-type: none"> Linux 및 macOS 클라이언트에서 WebAuth N 리디렉션 문제가 해결되었습니다.

DCV 2023.1-16220 — 2023년 11월 9일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 16220 nice-dcv-client(원도우): 8908 nice-dcv- 	<p>NICE DCV에 다음 기능이 추가되었습니다.</p> <ul style="list-style-type: none"> 원격 Google 크롬 또는 Microsoft Edge 브라우저에서 실행되는 웹 애플리케이션으로부터의 세션 내 WebAuth N 요청의 리디렉션을 지원합니다. 리디렉션된 요청은 클라이언트로 전달되므로 Windows Hello와 같은 YubiKey FIDO2 준수 인증자가 사용자 ID를 검증할 수 있습니다. 새로운 Windows 호스트용 간접 디스플레이 드라이버(IDD)는 그래픽 파이프라인을 최적 	<ul style="list-style-type: none"> Windows의 클립보드에 투명 이미지 지원이 추가되었습니다. Windows에서 클립보드에 동시 액세스할 때 일부 애플리케이션에서 잘라내기 및 붙여넣기 작업이 제대로 이루어지지 않던 문제를 해결했습니다. Windows에서 모니터 배율 인수가 100% NICE DCV 서버로 재설정되는 문제가 해결되었습니다.

빌드 번호	새로운 기능	변경 및 버그 수정
viewer (맥 OS): 6125 • nice-dcv-viewer (리눅스): 6125 • nice-xdcv: 565 • nice-dcv-gl: 1047 • nice-dcv-gltest: 325 • nice-dcv-simple-external-인증자: 228	화하고 프로토콜별로 전체적인 CPU 사용량을 대폭 줄여 줍니다. • 이제 Windows 성능 카운터를 사용하여 프레임 속도, 네트워크 대역폭, CPU 사용량 등과 같은 다양한 DCV 프로토콜 지표를 추적할 수 있어 사용자가 네트워크 및 DCV 프로토콜의 성능을 이해하는 데 도움이 될 수 있습니다.	<ul style="list-style-type: none"> • Windows 및 Linux의 콘솔 세션에서 사용자 로그아웃 및 화면 잠금 시 클라이언트 연결을 자동으로 끊는 설정을 추가했습니다. • 오디오 스택에서 잡음 및 사운드 아티팩트가 발생할 수 있는 문제를 해결했습니다. • 서버에서 애플리케이션을 종료하지 않고 다시 연결하면 웹캠 스트리밍을 재개할 수 있습니다. • Windows 네이티브 클라이언트에서 높은 dpi 마우스를 사용할 때의 상대적 마우스 동작이 개선되었습니다. • macOS 네이티브 클라이언트의 SmartCard 지원 관련 문제가 해결되었습니다. • Linux 네이티브 클라이언트에서 높은 픽셀 밀도에 대한 지원이 수정되었습니다. • 웹 클라이언트 및 Windows 네이티브 클라이언트의 사용자 인터페이스 접근성이 향상되었습니다. • macOS에서 웹 클라이언트를 사용할 때 일부 키보드 레이아웃의 제한 사항을 수정했습니다. • 서드 파티 종속성을 최신 버전으로 업데이트했습니다. • Xdcv가 XServer 21.1.9 버전으로 업데이트되었습니다. • Windows Server 2012R2, Ubuntu 18.04 및 SUSE Enterpris

빌드 번호	새로운 기능	변경 및 버그 수정
		<ul style="list-style-type: none"> e Linux 15SP4 지원이 제거되었습니다. • 버그 수정 및 성능 향상

DCV 2023.0-15487 — 2023년 6월 29일

빌드 번호	변경 및 버그 수정	
<ul style="list-style-type: none"> • nice-dcv-server: 15487 • nice-dcv-client(윈도우): 8771 • nice-dcv-viewer (macOS) 5629 • nice-dcv-viewer (리눅스): 5629 • nice-xdcv: 551 	<ul style="list-style-type: none"> • Chrome 114 이상을 사용할 때 잘못된 색상이 표시될 수 있는 웹 클라이언트 문제를 해결했습니다. • 제거 시 오류가 발생하지 않도록 NICE DCV 서버 및 Xdcv의 el7 rpm 패키지를 수정했습니다. • Windows 서버에서 NVIDIA GRID 드라이버 528.89와의 호환성 문제를 해결했습니다. • 일부 Windows 애플리케이션에서 클립보드가 제대로 작동하지 않을 수 있는 문제를 해결했습니다. • 이제 dcv-gl 패키지를 설치하거나 업데이트할 때 구성이 올바른지 확인하려면 최신 버전의 NICE DCV 서버 패키지가 필요합니다. • Windows 클라이언트에서 크기 조정 후 잘못된 해상도가 사용될 수 있는 문제를 해결했습니다. • macOS 및 Linux 클라이언트에서 IPv6 주소에 대한 지원이 수정되었습니다. • 이제 macOS 클라이언트에서 Control + 클릭을 오른쪽 클릭으로 구성할 수 있습니다. • 이제 웹 클라이언트는 지원되는 브라우저에서 전체 화면일 때 특수 키와 조합을 사용할 수 있습니다. 	

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-gl: 1039 nice-dcv-gltest: 318 nice-dcv-simple-external-인증자: 208 	<ul style="list-style-type: none"> OpenSSL 서드 파티 라이브러리를 업데이트했습니다.

DCV 2023.0-15065 — 2023년 5월 3일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 15065 nice-dcv-client(윈도우): 8671 nice-dcv-viewer(맥) 	<ul style="list-style-type: none"> 라이선스 토큰이 릴리스되지 않을 수 있는 <code>close-session</code> 관련 문제를 해결했습니다. macOS 네이티브 클라이언트에서 충돌이 발생하는 문제를 수정했습니다. BigSur

빌드 번호	변경 및 버그 수정	
OS): 5483		
• nice-dcv-viewer (리눅스): 5483		
• nice-xdcv: 547		
• nice-dcv-gl: 1027		
• nice-dcv-gltest: 318		
• nice-dcv-simple-external-인증자: 208		

DCV 2023.0-15022 — 2023년 4월 21일

빌드 번호	변경 및 버그 수정	
• nice-dcv-	• 화면 크기 조정 후 스트리밍이 제대로 작동하지 않는 동시성 문제를 해결했습니다.	

빌드 번호	변경 및 버그 수정
server: 15022	<ul style="list-style-type: none"> NICE DCV 서버에서 QUIC 연결에 장애가 발생할 수 있는 경합 상태를 해결했습니다.
<ul style="list-style-type: none"> nice-dcv-client(윈도우): 8671 	<ul style="list-style-type: none"> 숨겨진 커서가 있는 애플리케이션과 관련된 NICE DCV 서버의 충돌을 해결했습니다. Windows 서버의 일본어 키보드 입력 문제가 해결되었습니다. 웹캠 스트림의 오디오/비디오 동기화가 개선되었습니다.
<ul style="list-style-type: none"> nice-dcv-viewer(맥 OS): 5456 	<ul style="list-style-type: none"> ICU 및 libxml2 서드 파티 라이브러리를 업데이트했습니다. Xdcv를 XServer 21.1.8 버전으로 업데이트하고 가상 세션이 시작되지 않을 수 있는 XKB 관련 문제를 해결했습니다.
<ul style="list-style-type: none"> nice-dcv-viewer(리눅스): 5456 	<ul style="list-style-type: none"> Windows, macOS 및 Linux 네이티브 클라이언트에서 비디오 디코딩 실패를 일으킬 수 있는 문제를 해결했습니다. macOS 및 Linux 네이티브 클라이언트의 설정 관련 문제를 해결했습니다.
<ul style="list-style-type: none"> nice-xdcv: 547 	
<ul style="list-style-type: none"> nice-dcv-gl: 1027 	
<ul style="list-style-type: none"> nice-dcv-gltest: 318 	
<ul style="list-style-type: none"> nice-dcv-simple-ex 	

빌드 번호	변경 및 버그 수정
ternal- 인증자: 206	

DCV 2023.0-14852 — 2023년 3월 28일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 14852 nice-dcv-client(윈도우): 8655 nice-dcv-viewer (맥 OS): 5388 nice-dcv-viewer (리눅스): 5388 nice-xdcv: 527 	<p>NICE DCV에 다음 기능이 추가되었습니다.</p> <ul style="list-style-type: none"> macOS 및 Linux의 NICE DCV 클라이언트에 대해 선택된 모니터에서 전체 화면에 대한 지원이 추가되었습니다. 모든 클라이언트에 대해 드래그 앤 드롭으로 파일 업로드를 시작할 수 있는 지원이 추가되었습니다. Red Hat Enterprise Linux 9, Rocky Linux 9, CentOS Stream 9가 추가되었습니다. Linux에서 NICE DCV 서버의 시간대 리디렉션에 대한 지원이 추가되었습니다. 	<ul style="list-style-type: none"> 잘못된 대역폭 추정 및 시각적 아티팩트를 유발할 수 있는 QUIC 전송의 일부 문제를 해결했습니다. macOS 및 Linux 클라이언트의 사용자 인터페이스가 업데이트되었습니다. 이제 Windows 설치 프로그램에서 사용자가 볼 수 있는 애플리케이션 이름에 NICE DCV를 일관되게 사용합니다. 안정성을 높이기 위해 Windows의 클립보드 지원 구현을 재작업했습니다. Windows에서 독일어 키보드 레이아웃을 사용할 때 Caps Lock 키와 관련된 문제를 해결했습니다.

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-gl: 1022 nice-dcv-gltest: 318 nice-dcv-simple-external-인증자: 206 		

DCV 2022.2-14521 — 2023년 2월 17일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 14521 nice-dcv-client(윈도우): 8570 nice-dcv-viewer (맥OS): 5125 nice-dcv-viewer (리눅스): 4804 nice-xdcv: 519 nice-dcv-gl: 1012 nice-dcv-gltest: 307 nice-dcv-simple-external-인증자: 198 	<ul style="list-style-type: none"> macOS 클라이언트의 일본어 및 스페인어 키보드 문제를 해결했습니다. Windows NICE DCV 서버의 숫자 키패드 키와 관련된 문제를 해결했습니다. QUIC 연결 시 발생하는 메모리 누수 문제를 해결했습니다. 이전 비디오 드라이버 사용 시 Windows NICE DCV 클라이언트의 안정성이 향상되었습니다. OpenSSL 및 libsoup 서드 파티 라이브러리를 업데이트했습니다. Xdcv가 XServer 21.1.7 버전으로 업데이트되었습니다.

DV 2022.2-14357 — 2023년 1월 18일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 14357 nice-dcv-client(윈도우): 8522 nice-dcv-viewer (맥OS): 4804 nice-dcv-viewer (리눅스): 4804 nice-xdcv: 487 nice-dcv-gl: 1012 nice-dcv-gltest: 307 nice-dcv-simple-external-인증자: 198 	<ul style="list-style-type: none"> SUSE Linux 12의 가상 세션에서 발생하는 충돌이 SUSE 패키지의 최신 업데이트와 함께 발생하기 시작하던 문제를 해결했습니다. X Pixmap 처리와 관련된 DCV-GL의 메모리 누수를 해결했습니다. DCV-GL을 xrestop 도구와 통합하여 X Pixmap이 해당 프로세스에 연결되도록 했습니다. Windows 서버의 웹캠 및 오디오 리디렉션을 Windows의 기본 동작과 더 일치하도록 개선했습니다. 즉, OS 이벤트가 발생해도 스트림이 중단되지 않습니다. Windows NICE DCV 클라이언트가 입력 방법을 처리하는 방식을 개선했습니다. Windows NICE DCV 클라이언트에서 캐리지 리턴 문자만 줄 구분자로 사용하는 텍스트와 관련된 클립보드 문제를 해결했습니다.

DCV 2022.2-14175 — 2022년 12월 21일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 14175 nice-dcv-client(윈도우): 8472 nice-dcv-viewer (맥OS): 4804 nice-dcv-viewer (리눅스): 4804 nice-xdcv: 487 nice-dcv-gl: 983 nice-dcv-gltest: 307 	<ul style="list-style-type: none"> 연결을 사용할 때 서버에서 파일 디스크립터가 유출되는 문제를 수정했습니다. WebSocket Xdcv가 XServer 21.1.6 버전으로 업데이트되었습니다.

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-simple-external-인증자: 198 	

DCV 2022.2-14126 — 2022년 12월 9일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 14126 nice-dcv-client(윈도우): 8472 nice-dcv-viewer (맥OS): 4804 nice-dcv-viewer (리눅스): 4804 nice-xdcv: 481 nice-dcv-gl: 983 nice-dcv-gltest: 301 nice-dcv-simple-external-인증자: 198 	<ul style="list-style-type: none"> Windows 서버에서 한국어 키보드를 사용할 때 발생하는 문제를 해결했습니다. Windows 서버의 USB 리디렉션 관련 문제로 인해 Windows 11에서 작동이 중단될 수 있던 문제를 해결했습니다. 'rotate' 파라미터가 0으로 설정된 경우 서버의 로그 순환 문제가 해결되었습니다. macOS 및 Linux 클라이언트에서 특정 네트워크 조건하에 스트림이 정지되는 문제를 해결했습니다. Windows 네이티브 클라이언트가 전체 화면으로 전환할 때 크기가 제대로 조정되지 않는 문제를 해결했습니다. macOS 및 Linux 클라이언트에서 파일 업로드 중에 충돌이 발생할 수 있는 문제를 해결했습니다. macOS 클라이언트에서 오디오가 작동하지 않는 문제를 해결했습니다. Linux 클라이언트에서 NVIDIA GPU를 사용할 때 충돌이 발생할 수 있는 문제를 해결했습니다. 웹 클라이언트에서 시간대 리디렉션 UI가 서버와 동기화되지 않는 문제를 해결했습니다. 세션 후 페이지가 로드되지 않는 웹 클라이언트 문제를 해결했습니다.

빌드 번호	변경 및 버그 수정
	<ul style="list-style-type: none"> libTIFF 및 MIT-Kerberos 오픈 소스 종속성을 업데이트했습니다.

DCV 2022.2-13907 — 2022년 11월 11일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 13907 	<ul style="list-style-type: none"> NICE DCV에 다음 기능이 추가되었습니다. Windows의 NICE DCV 클라이언트에 대해 선택된 모니터에서 전체 화면에 대한 지원이 추가되었습니다. 	<ul style="list-style-type: none"> 웹 클라이언트 사용자 인터페이스를 Cloudscape 디자인 스타일로 업데이트했습니다.
<ul style="list-style-type: none"> nice-dcv-client(윈도우): 8427 	<ul style="list-style-type: none"> macOS에서 높은 픽셀 밀도 디스플레이 네이티브 클라이언트에 대한 지원이 추가되었습니다. 	<ul style="list-style-type: none"> 클라이언트 재연결로 인한 에이전트 내부 메모리 누수를 해결했습니다.
<ul style="list-style-type: none"> nice-dcv-viewer (맥 OS): 4653 	<ul style="list-style-type: none"> macOS 및 Linux에 NICE DCV 클라이언트를 위한 프린터 리디렉션을 추가했습니다. Windows에서 NICE DCV 서버의 시간대 리디렉션에 대한 지원이 추가되었습니다. 	<ul style="list-style-type: none"> Ubuntu 20.04에서 가상 세션을 사용할 때 GDM3를 사용하는 시스템에 대한 지원이 추가되었습니다.
<ul style="list-style-type: none"> nice-dcv-viewer (리눅스): 4653 	<ul style="list-style-type: none"> 콘솔 세션의 SSO(싱글 사인온)를 지원하는 Ubuntu 22.04용 Gnome-Shell 확장 프로그램을 추가했습니다. 	<ul style="list-style-type: none"> Ubuntu 20.04의 가상 세션에서 간헐적으로 검은 화면이 나타나는 문제를 해결했습니다.
<ul style="list-style-type: none"> nice-dcv-viewer (리눅스): 4653 	<ul style="list-style-type: none"> 오픈 소스 드라이버 사용 시 AMD GPU에 VA-API 기반 인코더가 추가되었습니다. 	<ul style="list-style-type: none"> 웹 클라이언트에서 탭을 변경할 때 클립보드 업데이트가 누락되는 문제를 해결했습니다.
<ul style="list-style-type: none"> nice-xdcv: 481 		<ul style="list-style-type: none"> 숫자 키패드의 Enter 키 관련 문제를 해결했습니다.

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-gl: 983 nice-dcv-gltest: 301 nice-dcv-simple-external-인증자: 198 		

DCV 2022.1-13300 — 2022년 8월 4일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 13300 nice-dcv-client(윈도우): 8261 nice-dcv-viewer (맥OS): 4279 nice-dcv-viewer (리눅스): 4251 nice-xdcv: 433 nice-dcv-gl: 973 nice-dcv-gltest: 295 nice-dcv-simple-external-인증자: 193 	<ul style="list-style-type: none"> 한 세션에 둘 이상의 공동 작업자가 연결되어 있는 경우 Windows를 자동으로 잠금 해제하지 마세요. 서버가 지정된 인증서 파일을 로드하지 못하는 문제가 해결되었습니다. macOS 클라이언트에서 오디오 왜곡이 발생하는 문제를 해결했습니다.

DCV 2022.1-13216 — 2022년 7월 21일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 13216 nice-dcv-client(윈도우): 8261 nice-dcv-viewer (맥OS): 4251 nice-dcv-viewer (리눅스): 4251 nice-xdcv: 433 nice-dcv-gl: 966 nice-dcv-gltest: 295 nice-dcv-simple-external-인증자: 193 	<ul style="list-style-type: none"> 모든 클라이언트에서 NICE DCV 서버 2019.1 및 이전 버전에 연결하지 못하는 문제를 해결했습니다. Windows 서버의 SmartCard 리디렉션 관련 문제를 수정했습니다. GPU가 있는 호스트의 NICE DCV 서버에 연결할 때 스트리밍이 실패하는 문제를 해결했습니다.

DCV 2022.1-13067 — 2022년 6월 29일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 13067 nice-dcv-client(윈도우): 8248 nice-dcv-viewer (맥OS): 4241 	<p>NICE DCV에 다음 기능이 추가되었습니다.</p> <ul style="list-style-type: none"> 서버에 Ubuntu 22.04와 Rocky Linux 8.5 이상에 대한 지원이 추가되었습니다. 네이티브 클라이언트용 Ubuntu 22.04에 대한 지원이 추가되었습니다. Windows, macOS 및 Linux 네이티브 클라이언트의 공동 작업 환경을 개선했습니다. 	<ul style="list-style-type: none"> 성능이 향상되어 비GPU 서버의 전체 CPU 사용량을 최대 30%까지 줄일 수 있습니다. 이제 시간 간격 또는 크기 제한을 지정하는 설정에서 로그 순환을 구성할 수 있습니다. 초기 핸드셰이크가 실패할 수 있었던 QUIC 전송 문제를 해결했습니다. 일부 애플리케이션에서 Linux 서버의 상대적 마우스 동작이 예상대로 작동하지 않는 문제를 해결했습니다.

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-viewer (리눅스): 4241 nice-xdcv: 433 nice-dcv-gl: 966 nice-dcv-gltest: 295 nice-dcv-simple-external-인증자: 193 		

DCV 2022.0-12760 — 2022년 5월 23일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 12760 nice-dcv-client(윈도우): 8145 nice-dcv-viewer (맥OS): 4131 nice-dcv-viewer (리눅스): 4131 	<p>변경 사항:</p> <p>옵션을 지정할 때 웹 클라이언트가 제대로 연결되지 않던 문제를 수정했습니다. <code>web-url-path</code></p>

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-xdcv: 424 nice-dcv-gl: 961 nice-dcv-gltest: 291 nice-dcv-simple-external-인증자: 188 	

DCV 2022.0-12627 — 2022년 5월 19일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 12627 nice-dcv-client(윈도우): 8145 nice-dcv-viewer (맥OS): 4131 nice-dcv-viewer (리눅스): 4131 nice-xdcv: 424 nice-dcv-gl: 961 nice-dcv-gltest: 291 nice-dcv-simple-external-인증자: 188 	<p>변경 사항:</p> <ul style="list-style-type: none"> 잘못된 대역폭 추정 및 시각적 아티팩트를 유발할 수 있는 QUIC 전송의 일부 문제를 해결했습니다. Windows 서버 설치 프로그램의 오디오 서비스 관련 문제로 인해 업데이트 프로세스가 실패할 수 있던 문제를 해결했습니다. Windows 클라이언트 설치 프로그램의 USB 처리 문제로 인해 제거 프로세스가 실패할 수 있었던 문제를 해결했습니다. macOS 및 Linux 클라이언트에서 스크린샷을 저장할 때 발생하는 문제를 해결했습니다. OpenSSL, zlib, gdk-pixbuf 서드 파티 라이브러리를 업데이트했습니다.

DCV 2022.0-12123 — 2022년 3월 23일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 12123 nice-dcv-client(윈도우): 7920 	NICE DCV에 다음 기능이 추가되었습니다.	<p>변경 사항:</p>

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-viewer (맥OS): 3973 nice-dcv-viewer (리눅스): 3973 nice-xdcv: 424 nice-dcv-gi: 961 nice-dcv-gitest: 291 nice-dcv-simple-external-인증자: 188 	<ul style="list-style-type: none"> macOS 및 Linux 클라이언트에 높은 색상 정확도를 지원하는 옵션이 추가되었습니다. 	<ul style="list-style-type: none"> QUIC 전송을 사용할 때 대역폭 추정 및 이미지 품질이 개선되었습니다. <p>수정 사항:</p> <ul style="list-style-type: none"> NVIDIA 드라이버 510.xx 버전을 사용할 때 Linux의 콘솔 세션에서 발생하는 시각적 아티팩트를 수정했습니다. Windows 네이티브 클라이언트에서 블루투스를 통해 컨트롤러 DualShock 4개가 연결되던 문제를 수정했습니다. 웹캠을 활성화할 때 macOS 클라이언트에서 발생할 수 있는 충돌을 수정했습니다.

DCV 2022.0-11954 — 2022년 2월 23일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 11954 nice-dcv-client(윈도우): 7866 nice-dcv-viewer (macOS): 3929 nice-dcv-viewer (리눅스): 3929 nice-xdcv: 424 nice-dcv-gi: 961 	<p>NICE DCV에 다음 기능이 추가되었습니다.</p> <ul style="list-style-type: none"> Windows Server 및 Windows 네이티브 클라이언트에 대한 게임 컨트롤러를 지원합니다. NICE DCV 웹 클라이언트는 이제 이를 지원하는 브라우저를 WebCodecs 활용합니다. 	<p>변경 사항:</p> <ul style="list-style-type: none"> 이제 NICE DCV 서버를 다시 시작하지 않고도 TLS 인증서를 업데이트할 수 있습니다. 이제 특정 네트워크 인터페이스 또는 특정 IPv4 또는 IPv6 주소에서 수신하도록 NICE DCV 서버를 구성할 수 있습니다.

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-glttest: 291 nice-dcv-simple-external-인증자: 188 	<ul style="list-style-type: none"> Windows 및 웹 클라이언트에 높은 색상 정확도를 지원하는 옵션이 추가되었습니다. 공동 작업 환경 개선: 누군가 세션에 참여하면 사용자에게 알림이 전송됩니다. 지원되는 Linux 배포판 목록에 CentOS 8 Stream이 추가되었습니다. 	<ul style="list-style-type: none"> 이제 Linux 시스템에서도 'DCV 프린터'가 자동으로 구성됩니다. 이제 Windows의 NICE DCV 프로세스가 더 높은 우선 순위로 실행됩니다. <p>수정 사항:</p> <ul style="list-style-type: none"> Windows 2016에서 GPU가 있는 인스턴스 사용 시 에이전트를 다시 시작할 때 발생하는 충돌을 수정했습니다. 일부 USB 디바이스가 NICE DCV 클라이언트에서 리디렉션되는 동안 세션에서 로그아웃할 때 Windows에서 충돌이 발생하는 문제를 해결했습니다. 권한 검사를 수행할 때 Windows 도메인이 포함된 사용자 이름을 정규화합니다. Windows 클라이언트의 상대 마우스 모드를 개선했습니다. 키 동기화 관련 문제가 수정되었습니다. CapsLock

DCV 2021.3-11591 — 2021년 12월 20일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 11591 nice-dcv-client(윈도우): 7801 nice-dcv-viewer (macOS): 3829 nice-dcv-viewer (리눅스): 3829 nice-xdcv: 415 nice-dcv-gl: 952 nice-dcv-gltest: 284 nice-dcv-simple-external-인증자: 176 	<p>NICE DCV에 다음 기능이 추가되었습니다.</p> <ul style="list-style-type: none"> 웹 클라이언트의 사용자 인터페이스가 업데이트되었습니다. 이제 EC2 G5 및 G5g 인스턴스가 지원됩니다. 이제 Windows Server 2022와 Windows 11 운영 체제가 지원됩니다. 	<ul style="list-style-type: none"> Linux 가상 세션용 init 스크립트는 더 이상 사용자의 bash 프로파일을 로드하지 않으므로 시스템 기본값을 재정의하는 환경 변수와 관련된 문제가 반복되지 않습니다. nice-dcv-ext-authenticator 이제 Python 3이 필요합니다.

DCV 2021.2-11445 — 2021년 11월 18일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 11445 nice-dcv-client(윈도우): 7792 nice-dcv-viewer (맥OS): 3797 nice-dcv-viewer (리눅스): 3797 nice-xdcv: 411 nice-dcv-gl: 946 nice-dcv-gltest: 279 nice-dcv-simple-external-인증자: 160 	<p>수정 사항:</p> <ul style="list-style-type: none"> macOS Monterey에서 클라이언트가 제대로 작동하지 않는 문제를 해결했습니다. Windows에서 서버에 대한 보안을 개선했습니다. 특히 웹 클라이언트를 사용할 때 다중 모니터 레이아웃이 제대로 적용되지 않는 버그가 수정되었습니다. 일부 Windows 애플리케이션에서 Delete 키가 제대로 작동하지 않는 문제를 해결했습니다.

빌드 번호	변경 및 버그 수정
	<ul style="list-style-type: none"> Linux의 웹 클라이언트 패키지를 웹 클라이언트 자체가 포함된 이전 버전의 서버 패키지와 상호 배타적인 것으로 표시했습니다.

DCV 2021.2-11190 — 2021년 10월 11일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 11190 nice-dcv-client(윈도우): 7788 nice-dcv-viewer (맥OS): 3776 nice-dcv-viewer (리눅스): 3776 nice-xdcv: 411 nice-dcv-gl: 946 nice-dcv-gltest: 279 nice-dcv-simple-external-인증자: 160 	<p>수정 사항:</p> <ul style="list-style-type: none"> 만료된 인증서로 서버에 연결할 때 사용자가 인증서 유효성 검사 대화 상자를 닫지 못하던 Windows 클라이언트 문제를 해결했습니다. 네이티브 클라이언트에서 스타일러스 펜의 가운데 클릭 버튼이 예상대로 작동하지 않는 문제를 해결했습니다. Xdcv에서 레거시 X11 글꼴을 로드할 수 없었던 회귀 현상을 해결했습니다. macOS 및 Linux 클라이언트에서 데드 키를 사용하는 키보드 레이아웃 사용 시 키보드 조합이 제대로 작동하지 않는 문제를 해결했습니다.

DCV 2021.2-11135 — 2021년 9월 24일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 11135 nice-dcv-client(윈도우): 7781 nice-dcv-viewer (맥OS): 3740 nice-dcv-viewer (리눅스): 3740 nice-xdcv: 408 	<p>수정 사항:</p> <ul style="list-style-type: none"> 2021.2 클라이언트를 사용하여 이전 서버에 연결할 때 연결 및 성능 문제를 일으킬 수 있는 QUIC 패킷 크기 협상 문제를 해결했습니다.

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-gl: 944 nice-dcv-gltest: 279 nice-dcv-simple-external-인증자: 160 	<ul style="list-style-type: none"> NVENC 인코더가 실패할 수 있는 NVIDIA 디바이스 선택 관련 버그가 수정되었습니다. Windows 및 NVIDIA GPU가 설치된 시스템에서 압축 아티팩트와 색상 정확도 아티팩트가 발생할 수 있는 문제를 해결했습니다. 일부 키보드 조합이 예상대로 작동하지 않을 수 있는 Linux 서버의 보조 키 관련 버그가 수정되었습니다. M1 CPU가 설치된 컴퓨터에서 macOS 클라이언트의 성능 저하를 해결했습니다. 일부 키보드 조합이 예상대로 작동하지 않는 macOS 클라이언트의 버그가 수정되었습니다. Linux 가상 세션에서 터치 이벤트가 처리되는 방식으로 인해 세션이 종료될 수 있었던 문제를 해결했습니다.

DCV 2021.2-11048 — 2021년 9월 1일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 11048 nice-dcv-client(윈도우): 7774 nice-dcv-viewer (맥OS): 3690 nice-dcv-viewer (리눅스): 3690 nice-xdcv: 406 nice-dcv-gl: 944 nice-dcv-gltest: 279 	<p>NICE DCV에 다음 기능이 추가되었습니다.</p> <ul style="list-style-type: none"> 웹 클라이언트 클립보드 개선. 이러한 개선 사항으로 이제 Google Chrome과 Microsoft Edge에서 NICE DCV 웹 클라이언트를 사용하여 PNG 형식 이미지를 복사하여 붙여넣을 수 있습니다. 	<p>변경 사항:</p> <ul style="list-style-type: none"> NICE DCV 웹 클라이언트는 이제 Linux에서는 별도의 패키지이며 Windows 설치 프로그램에서는 선택적 구성 요소입니다. 이번 변경으로 고객은 웹 클라이언트를 배포할지 여부를 결정할 수 있습니다. 이제 NVENC 인코더를 사용할 때 H.264 High Profile이

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-simple-external-인증자: 160 	<ul style="list-style-type: none"> Windows 및 macOS 클라이언트를 위한 스크린샷 차단 기능. 이 기능은 사용자가 NICE DCV 세션 콘텐츠의 스크린샷을 찍지 못하도록 하여 보안을 한층 더 강화합니다. 이 기능을 활성화하면 사용자가 캡처한 모든 스크린샷은 빈 화면이 됩니다. 스트리밍 품질 개선. 특히 QUIC 프로토콜을 사용할 때 “build-to-lossless” 성능이 향상되어 스트리밍 품질이 향상되었습니다. 클라이언트의 동작을 지정하는 certificate-validation-policy 옵션이 추가되었습니다. 서버가 신뢰할 수 없는 X.509 인증서(예: 자체 서명된 인증서)를 제공할 때 이 옵션을 사용할 수 있습니다. 런타임 시 오디오 드라이버에 구성된 채널 수를 변경할 수 있습니다. Pressure2K 옵션이 dcvinput Xorg 모듈에 추가되었습니다. 이 옵션을 사용하여 스타일러스의 압력 민감도 범위를 0~65335에서 0~2048로 변경하여 Mari 및 Nuke와 같은 애플리케이션과의 호환성을 유지할 수 있습니다. 	<p>지원됩니다. NVIDIA GPU와 함께 NVENC 인코더를 사용하면 동일한 이미지 품질을 유지하면서 대역폭 사용량을 줄일 수 있습니다.</p> <ul style="list-style-type: none"> NICE DCV 서버는 이제 GPU가 두 개 이상인 시스템에서 압축에 사용 가능한 모든 GPU를 사용합니다. NICE DCV와 함께 제공되는 모든 Windows 드라이버는 이제 WHQL 인증을 받았습니다. OpenSSL이 1.1.1 버전으로 업데이트되었습니다. Xdcv가 XServer 1.20.13 버전으로 업데이트되었습니다. <p>수정 사항:</p> <ul style="list-style-type: none"> macOS 클라이언트의 숫자 키패드 키 관련 문제를 해결했습니다. 일부 USB 디바이스(예: 게임패드)가 Windows 서버로 제대로 리디렉션되지 않던 문제를 해결했습니다. 연결이 끊겼을 때 보조 키를 제대로 해제할 수 없는 버그가 수정되었습니다. Ubuntu 20.04와 인텔 GPU를 사용할 때 Linux 네이티브

빌드 번호	새로운 기능	변경 및 버그 수정
	<ul style="list-style-type: none"> 구글 크롬과 마이크로소프트 엣지의 실험용 WebCodecs API에 대한 지원이 추가되었습니다. 브라우저에서 이 API를 활성화하면 NICE DCV 웹 클라이언트가 이를 사용하여 비디오 디코딩을 가속화하고 더 높은 프레임 속도를 제공할 수 있습니다. 	<p>클라이언트에서 충돌이 발생하는 문제를 해결했습니다.</p>

DCV 2021.1-10851 — 2021년 7월 30일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 10851 nice-dcv-client(윈도우): 7744 nice-dcv-viewer(macOS): 3590 nice-dcv-viewer(리눅스): 3560 nice-xdcv: 392 nice-dcv-gi: 937 nice-dcv-gitest: 275 nice-dcv-simple-external-인증자: 154 	<p>변경 사항:</p> <ul style="list-style-type: none"> Windows, Linux, macOS 클라이언트의 안정성을 개선했습니다. <p>수정 사항:</p> <ul style="list-style-type: none"> Windows 서버에서 AMD 및 NVIDIA 그래픽 어댑터를 사용할 때 화면이 깜박이는 버그가 수정되었습니다. 여러 세션을 실행하는 Linux 서버에 연결할 때 간헐적으로 발생하는 문제를 해결했습니다. Linux 서버에서 서양어가 아닌 키보드 레이아웃 처리와 관련된 버그가 수정되었습니다. Windows 클라이언트의 연결 창에서 나타나는 시각적 아티팩트를 수정했습니다. Windows용 USB 리디렉션 드라이버의 여러 버그를 수정하고 디바이스 호환성을 개선했습니다.

DCV 2021.1-10598 — 2021년 6월 10일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 10598 nice-dcv-client(윈도우): 7713 nice-dcv-viewer(맥OS): 3473 nice-dcv-viewer(리눅스): 3473 nice-xdcv: 392 nice-dcv-gl: 937 nice-dcv-gltest: 275 nice-dcv-simple-external-인증자: 154 	<ul style="list-style-type: none"> 서버의 Windows 설치 프로그램에서 session owner 필드를 현재 사용자로 미리 채우는 문제를 해결했습니다. macOS 및 Linux 클라이언트의 전반적인 안정성을 개선했습니다.

DCV 2021.1-10557 — 2021년 5월 31일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 10557 nice-dcv-client(윈도우): 7713 nice-dcv-viewer (맥OS): 3450 nice-dcv-viewer (리눅스): 3454 nice-xdcv: 392 nice-dcv-gl: 937 nice-dcv-gltest: 275 nice-dcv-simple-external-인증자: 154 	<ul style="list-style-type: none"> NICE DCV에 GPU로 서버에 연결할 때 정확한 오디오/비디오 동기화가 가능하도록 클라이언트 옵션을 추가했습니다. NICE DCV에 Linux 콘솔 세션의 마이크에 대한 지원을 추가했습니다. 	<ul style="list-style-type: none"> GPU가 없는 Windows 서버 호스트에서 CPU 사용량을 줄였습니다. macOS 및 Linux 클라이언트에서 .dcv 연결 파일을 읽을 때 발생하는 문제를 해결했습니다. 하드웨어 가속 디코딩을 지원하지 않는 macOS 시스템의 소프트웨어 디코딩에 대한 폴백을 추가했습니다. macOS 클라이언트가 시스템 키체인에 저장된 CA 인증서를 읽을 수 있도록 지원이 추가되었습니다.

DCV 2021.0-10242 — 2021년 4월 12일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 10242 nice-dcv-client(윈도우): 7643 nice-dcv-viewer (맥OS): 3186 nice-dcv-viewer (리눅스): 3294 nice-xdcv: 380 nice-dcv-gi: 912 nice-dcv-gitest: 266 nice-dcv-simple-external-인증자: 134 	<ul style="list-style-type: none"> Windows NICE DCV 서버에 대한 웹캠 리디렉션 지원이 추가되었습니다. Linux NICE DCV 서버에 대한 프린터 리디렉션 지원이 추가되었습니다. macOS 클라이언트에 M1 프로세서 지원이 추가되었습니다. macOS 클라이언트를 위한 다중 모니터 디스플레이 지원이 추가되었습니다. 	<ul style="list-style-type: none"> NVIDIA GPU를 사용하는 Linux 서버 및 Amazon EC2 인스턴스에서 GPU 및 CPU 리소스 사용을 최적화했습니다. Linux NICE DCV 서버용 Amazon EC2 G4ad 인스턴스에서 AMD GPU를 사용한 GPU 가속 비디오 인코딩에 대한 지원이 추가되었습니다. 오디오 처리를 최적화하여 오디오 지연 시간을 줄였습니다. 서버에서 프로토콜이 활성화된 경우 클라이언트의 기본 값을 QUIC 프로토콜로 변경했습니다. DCV 명령줄 도구에 새로운 get-screenshot 명령을 추가했습니다. close-session 명령의 --logout-user 옵션을 사용하는 강제 로그아웃 옵션을 추가했습니다. 콘솔 세션을 종료할 때 이 옵션을 사용할 수 있습니다.

DCV 2020.2-9662 — 2020년 12월 4일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 9662 nice-dcv-client(윈도우): 7490 nice-dcv-viewer (맥OS): 2117 nice-dcv-viewer (리눅스): 3007 nice-xdcv: 359 nice-dcv-gi: 881 nice-dcv-glttest: 259 nice-dcv-simple-external-인증자: 125 	<ul style="list-style-type: none"> 웹 브라우저 클라이언트에서 사용되는 보안 프로토콜을 개선했습니다. Windows 클라이언트와 함께 사용되는 Amazon EC2 G4ad 인스턴스의 성능과 안정성이 향상되었습니다. Windows 클라이언트의 연결 설정 대화 상자에서 포트를 선택할 때 발생하는 문제를 해결했습니다.

DCV 2020.2-9508 — 2020년 11월 11일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 9508 nice-dcv-client(윈도우): 7459 nice-dcv-viewer (맥OS): 2078 nice-dcv-viewer (리눅스): 1737 nice-xdcv: 359 nice-dcv-gi: 881 nice-dcv-glttest: 259 nice-dcv-simple-external-인증자: 125 	<ul style="list-style-type: none"> QUIC(UDP 기반) 전송 프로토콜에 대한 지원이 추가되었습니다. SLES 15 및 Ubuntu 20.4에 대한 지원이 추가되었습니다. Windows NICE DCV 서버에 대한 스마트 카드 지원이 추가되었습니다. 	<ul style="list-style-type: none"> NVIDIA GPU를 사용하는 서버 및 EC2 인스턴스에서 호스팅되는 콘솔 세션의 NICE DCV 프레임 속도 제한 기본값을 60FPS로 변경했습니다. NVIDIA GPU가 있는 EC2 인스턴스에 호스팅되는 Windows NICE DCV 서버의 GPU 및 CPU 리소스를 최적화했습니다. list-endpoints NICE DCV CLI 명령이 추가되었습니다. 여기에는 현재 활성 엔드포인트가 나열됩니다.

빌드 번호	새로운 기능	변경 및 버그 수정
		<ul style="list-style-type: none"> • <code>version</code> NICE DCV CLI 명령은 <code>--json</code> 옵션을 지원합니다. • 이제 Linux 서버에서 <code>create-session</code> NICE DCV CLI 명령이 <code>--disable-login-monitor</code> 옵션을 지원합니다. • Linux NICE DCV 서버의 다양한 디스플레이 관리자와의 호환성이 향상되었습니다. • 키보드 입력 처리 시 발생하는 몇 가지 문제를 해결했습니다. • 이제 USB 디바이스 허용 목록 파일이 동적으로 다시 로드됩니다.

DCV 2020.1-9012 — 2020년 9월 30일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> • nice-dcv-server: 9012 • nice-dcv-client (윈도우): 7342 • nice-dcv-viewer (맥OS): 1986 • nice-dcv-viewer (리눅스): 1545 • nice-xdcv: 338 • nice-dcv-gi: 840 • nice-dcv-glttest: 246 • nice-dcv-simple-external-인증자: 111 	<ul style="list-style-type: none"> • 누락된 macOS 클라이언트 아이콘을 추가했습니다.

DCV 2020.1-9012 — 2020년 8월 24일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 9012 nice-dcv-client (윈도우): 7342 nice-dcv-viewer (맥OS): 1910 nice-dcv-viewer (리눅스): 1545 nice-xdcv: 338 nice-dcv-gi: 840 nice-dcv-glttest: 246 nice-dcv-simple-external-인증자: 111 	<ul style="list-style-type: none"> AWS GovCloud 지역에서의 Amazon S3 액세스를 수정했습니다. 웹 기반 클라이언트 개선 사항

DCV 2020.1-8942 — 2020년 8월 3일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 8942 nice-dcv-client (윈도우): 7342 nice-dcv-viewer (맥OS): 1910 nice-dcv-viewer (리눅스): 1545 nice-xdcv: 338 nice-dcv-gi: 840 nice-dcv-glttest: 246 nice-dcv-simple-external-인증자: 111 	<ul style="list-style-type: none"> 리눅스 NICE DCV 서버는 이제 M6g, C6g, R6g와 같은 그라비톤2 기반 Arm 인스턴스를 AWS 지원합니다. 자세한 내용은 AWS Graviton 프로세서를 참조하세요. Linux NICE DCV 서버에서 RHEL 8.x 및 CentOS 8.x에 대한 지원이 추가되었습니다. Windows NICE DCV 서버 및 Windows NICE DCV 클라이언트 사용 시 프린터 리디렉션에 대한 지원이 추가되었습니다. 	<ul style="list-style-type: none"> GPU가 없는 Amazon EC2 인스턴스에 새로운 NICE DCV 가상 디스플레이 드라이버에 대한 지원이 추가되었습니다. NVENC 인코더를 사용할 때 색 공간 변환으로 인해 시각적 아티팩트가 발생하는 문제가 해결되었습니다. 이제 <code>dcv list-sessions</code> 명령에는 콘솔 세션(있는 경우)이 항상 포함됩니다. 이제 최신 Linux 배포판에서 콘솔 세션용 에이전트가 데스크톱 세션의 일부로 시작

빌드 번호	새로운 기능	변경 및 버그 수정
	<ul style="list-style-type: none"> • macOS 및 Linux 네이티브 NICE DCV 클라이언트에서 압력 감지 기능을 갖춘 스타 일러스 지원이 추가되었습니다. • Linux NICE DCV 서버 및 Linux NICE DCV 클라이언트에 대한 서라운드 사운드 5.1 지원이 추가되었습니다. • Linux NICE DCV 네이티브 클라이언트에 대한 터치 스크린 지원이 추가되었습니다. • 이제 사용자 지정 이름을 NICE DCV 세션에 연결할 수 있습니다. • macOS 네이티브 NICE DCV 클라이언트에서 하드웨어 가속 디코딩 및 렌더링을 지원합니다. 	<ul style="list-style-type: none"> • 되어 GDM3와 같은 최신 디스플레이 관리자 지원이 향상되었습니다. • 이제 <code>dcv://</code> 스키마를 사용하여 URL을 활성화하면 네이티브 클라이언트가 자동으로 열립니다. • macOS 네이티브 클라이언트와 웹 클라이언트가 키보드 보조 키를 처리하는 방식을 개선했습니다. • DCV-GL의 시각적 및 <code>fbconfig</code> 선택을 개선하여 일부 애플리케이션에 대한 지원을 개선했습니다. • 파일 전송 중 CPU 사용량을 줄였습니다. • 웹 브라우저 클라이언트의 WebGL 렌더링을 개선하여 리소스 사용량을 줄였습니다.

DCV 2020.0-8428 — 2020년 4월 16일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> • nice-dcv-server: 8428 • nice-dcv-client (윈도우): 7238 • nice-dcv-viewer (맥OS): 1716 	<ul style="list-style-type: none"> • Linux 서버에 온스크린 스타 일러스 및 터치 지원을 추가했습니다. • Windows 서버의 서라운드 사운드 7.1 재생 지원을 Windows 네이티브 클라이언트에 추가했습니다. 	<ul style="list-style-type: none"> • 이제 전체 화면 모드에서 Windows 클라이언트의 도구 모음 그림을 숨길 수 있습니다. • Windows 네이티브 클라이언트에 NTLM 프록시 지원을 추가했습니다.

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> • nice-dcv-viewer (리눅스): 1358 • nice-xdcv: 296 • nice-dcv-gl: 759 • nice-dcv-gltest: 229 • nice-dcv-simple-external-인증자: 87 	<ul style="list-style-type: none"> • Linux 네이티브 클라이언트에 하드웨어 가속 및 스타일러스 지원을 추가했습니다. • 서버 측에서 디스플레이 레이아웃을 설정하는 새로운 API 명령을 추가했습니다. • Microsoft Edge 브라우저 (79.0.309 버전 이상)에 다중 모니터 웹 클라이언트 디스플레이 지원을 추가했습니다. 	<ul style="list-style-type: none"> • NVIDIA 어댑터를 사용하여 Windows 헤드리스 물리적 호스트에 대한 지원을 개선했습니다. • 레거시 NVIDIA NvIFR 라이브러리에 대한 지원을 없앴습니다. • 최신 Windows 10에서 Windows 그래픽 캡처 API에 대한 지원을 추가했습니다. • Amazon EC2 지원이 추가되었습니다. <p>EC2 인스턴스의 인스턴스 메타데이터 서비스(IMDS) v2.</p> <ul style="list-style-type: none"> • DCV CLI는 클라이언트가 세션에 연결되거나 연결 해제되는 시점을 감지하는 새로운 on-client-connected /disconnected 명령을 제공합니다. • 외부 인증자에 대한 인증서를 바인딩하기 위해 호스트 이름을 지정하는 지원을 추가했습니다. • DCV-GL이 이제 GLvnd(GL Vendor-Neutral Dispatch library)를 지원하는 시스템에서 이를 사용합니다.

DCV 2019.1-7644 — 2019년 10월 24일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 764 nice-dcv-client (윈도우): 7114 nice-dcv-viewer (맥OS): 1535 nice-dcv-viewer (리눅스): 1124 nice-xdcv: 226 nice-dcv-gl: 544 nice-dcv-gltest: 220 nice-dcv-simple-external-인증자: 77 	<ul style="list-style-type: none"> NICE EnginFrame 및 기타 세션 관리자가 사용하는 통합 API의 문제를 수정했습니다. 32비트 버전의 Windows 네이티브 클라이언트 문제를 해결했습니다.

DCV 2019.1-7423 — 2019년 9월 10일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 7423 nice-dcv-client (윈도우): 7087 nice-dcv-viewer (맥OS): 1535 nice-dcv-viewer (리눅스): 1124 nice-xdcv: 226 nice-dcv-gl: 544 nice-dcv-gltest: 220 nice-dcv-simple-external-인증자: 77 	<ul style="list-style-type: none"> Windows에서 DCV 서버에 대한 보안을 개선했습니다. Linux에서 Autodesk Maya의 렌더링 문제를 해결했습니다. 키보드 처리와 관련된 개선 사항 및 버그 수정을 추가했습니다.

DCV 2019.0-7318 — 2019년 8월 5일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> • nice-dcv-server: 7318 • nice-dcv-client (윈도우): 7059 • nice-dcv-viewer (맥OS): 1530 • nice-dcv-viewer (리눅스): 968 • nice-xdcv: 224 • nice-dcv-gi: 529 • nice-dcv-gitest: 218 • nice-dcv-simple-external-인증자: 72 	<ul style="list-style-type: none"> • 웹 클라이언트에서 다중 모니터 지원. • Windows Server 2019에서 스타일러스 입력 지원. • MacOS 및 Linux 네이티브 클라이언트에서 오디오 입/출력. • Linux 서버에서 클립보드 기능 개선(가운데 클릭 붙여넣기). 	<ul style="list-style-type: none"> • Windows 터치 입력의 압력 감지 호환성을 개선했습니다. • Windows에서 이기종 그래픽 어댑터가 있는 시스템의 동작을 개선했습니다. • 비활성 연결을 감지하는 데 필요한 시간을 줄였습니다 (예: 클라이언트에서 유선에서 Wi-Fi 네트워크로의 변경에 대한 대응). • Linux에서 커서 아이콘을 캡처할 수 없는 경우 로깅을 줄였습니다. • 가상 세션 Xdcv 구성 요소에서 Composite 확장 비활성화를 지원합니다. • 동시 가상 세션 수에 대한 제한을 설정할 수 있는 옵션을 추가했습니다. • Bash 5가 설치된 시스템과의 스크립트 호환성을 개선했습니다. • Linux 클라이언트에서 OpenGL 및 GLES를 자동으로 감지하여 렌더링에 사용하도록 기본값을 변경했습니다. • GL 창의 가시성이 바뀔 때 DCV-GL 화면 버퍼를 업데이트했습니다.

빌드 번호	새로운 기능	변경 및 버그 수정
		<ul style="list-style-type: none"> Windows 7의 Windows 클라이언트에서 마우스 휠 감지를 수정했습니다. 일부 Windows 7 시스템에서 라이브러리를 로드할 때 Windows 클라이언트가 실패하는 문제를 해결했습니다. 가로 방향으로 문서를 인쇄할 때 Windows 클라이언트의 인쇄 기능을 개선했습니다.

DCV 2017.4-6898 — 2019년 4월 16일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 6898 nice-dcv-client (윈도우): 6969 nice-dcv-viewer (맥OS): 1376 nice-dcv-viewer (리눅스): 804 nice-xdcv: 210 nice-dcv-gi: 490 nice-dcv-gitest: 216 nice-dcv-simple-external-인증자: 70 	<ul style="list-style-type: none"> MacOS를 위한 새로운 네이티브 클라이언트. 	<ul style="list-style-type: none"> Windows 네이티브 클라이언트가 이제 디코딩 및 렌더링에 하드웨어 가속을 사용합니다(시스템에서 사용할 수 있는 경우). dcv 명령줄 도구가 이제 Windows와 Linux 모두에서 동일한 옵션과 출력 형식을 사용합니다. dcv 명령줄 도구가 이제 라이선스에 대한 정보를 보고합니다. 클라이언트가 이제 비활성으로 인해 연결을 끊기 전에 사용자에게 경고를 표시합니다.

빌드 번호	새로운 기능	변경 및 버그 수정
		<ul style="list-style-type: none"> 여러 한정자를 사용하는 키보드 조합에 대한 지원을 개선했습니다. 통신 실패에 대한 Reprise License Manager와의 상호작용의 견고성을 개선했습니다. dcvusers 명령줄 도구가 이제 기본적으로 Linux의 dcv 사용자 홈 디렉터리에 데이터를 저장합니다. Linux에서 여러 GPU와 함께 NVENC 하드웨어 인코더를 사용할 때 nvidia-smi 도구에서 사용하는 것과 동일한 순서를 따랐습니다. Linux 클라이언트가 이제 Windows DCV 프린터에서 인쇄된 파일을 수신하고 처리합니다.

DCV 2017.3-6698 — 2019년 2월 24일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 6698 nice-dcv-client: 5946 nice-dcv-viewer (리눅스): 683 nice-xdcv: 207 nice-dcv-gi: 471 nice-dcv-gltest: 210 	<ul style="list-style-type: none"> Kerberos(GSSAPI) 인증에 대한 지원을 추가했습니다. 터치 이벤트를 지원하는 Windows 버전에서 터치 이벤트에 대한 지원을 추가했습니다. 시스템 인증(Windows 자격 증명 공급자)을 사용할 때 	<ul style="list-style-type: none"> Y'UV444 인코딩을 옵트인하는 옵션을 추가했습니다. EL6 RPM에 이제 NVENC 인코더 모듈이 포함됩니다. Windows 시스템 인증에서 이제 name@domain 형식을 수락합니다.

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-simple-external-인증자: 66 	<p>Windows 세션을 자동으로 잠금 해제합니다.</p>	<ul style="list-style-type: none"> Yubikey USB 디바이스가 이제 허용 목록에 추가됩니다. 일본어 키보드 지원을 개선했습니다. 입력 권한 부여 권한이 보다 세분화됩니다. 가상 커서를 처리할 수 있는 pointer 권한을 추가했습니다. 상대 마우스 모드는 마우스(모션 삽입)와 포인터(모션 피드백)에 따라 달라집니다. Windows에서 SAS를 처리할 keyboard-sas 권한을 추가했습니다(Control+Alt + Del). keyboardsas 는 keyboard 권한에 따라 달라집니다. 비동기 클립보드 API를 지원하는 브라우저의 웹 클라이언트에서 빈 클립보드 이벤트 문제를 해결했습니다. 캡처 모듈에서 클라이언트가 첫 번째 프레임을 수신하지 못하는 경합 상태를 수정했습니다. 동시 파일 스토리지 전송 처리를 개선했습니다. 최신 NVIDIA 드라이버가 장착된 Windows에서 NvIFR을 수정했습니다. 새 드라이버가 동작을 변경했습니다. 드라이버 버전이 이제 자동으로 감지되고 그에 따라 메모리 처리가 수행됩니다.

빌드 번호	새로운 기능	변경 및 버그 수정
		<ul style="list-style-type: none"> • RLM 라이선스 토큰을 다시 획득하려는 재시도를 중지하지 마세요. 이렇게 하면 연장된 기간이 지난 후에도 licensing error 상태에서 복구할 수 있습니다. • Windows 클라이언트에서 전체 화면 바로 가기 키보드 단축키를 설정하는 옵션을 추가했습니다. • Windows 클라이언트에서 여러 모니터에 걸쳐 창을 드래그할 때의 자동 맞춤 논리를 개선했습니다. • Windows 클라이언트에서 UI에 의해 연결 해제가 트리거되지 않을 때 프롬프트 다시 연결 옵션을 수정했습니다. • NVIDIA 드라이버 410.xx와 DCV-GL 비호환성을 수정했습니다. • Matlab 및 Blender 애플리케이션에서 DCV-GL의 회귀를 수정했습니다.

DCV 2017.2-6182 — 2018년 10월 8일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> • nice-dcv-server: 6182 • nice-dcv-client: 5890 	<ul style="list-style-type: none"> • Linux 가상 세션에 오디오 재생 지원을 추가했습니다. • 스마트 카드 성능을 개선했습니다. 	<ul style="list-style-type: none"> • 키보드 처리와 관련된 개선 및 버그 수정.

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> • nice-dcv-viewer (리눅스): 503 • nice-xdcv: 180 • nice-dcv-gi: 427 • nice-dcv-gitest: 201 • nice-dcv-simple-external-인증자: 58 	<ul style="list-style-type: none"> • Linux 클라이언트에서 파일 전송 지원을 추가했습니다. 	<ul style="list-style-type: none"> • 구성에서 로그 수준을 변경할 때 더 이상 서버를 다시 시작할 필요가 없습니다. • 이제 Windows 서버 설치 프로그램이 Microsoft C 런타임 재배포 가능 패키지가 이미 설치되어 있는 경우 패키지 설치를 건너뛵니다. • EC2에서 실행할 때 라이선스의 S3 액세스가 실패하면 사용자 인터페이스에 알림이 표시됩니다. • Linux dcv 명령줄 도구가 이제 <code>list-connections</code> 및 <code>describe-session</code> 하위 명령을 지원하며 JSON 출력을 내보내는 옵션을 포함합니다. • <code>display</code> 섹션에 <code>cuda-devices</code> 설정을 추가했습니다. 이를 통해 NVENC 인코딩을 서로 다른 CUDA 디바이스에 배포하도록 서버를 구성할 수 있습니다. • 여러 개의 동시 명령을 처리할 때 세션 생성 코드의 견고성을 개선했습니다. • 기본 클립보드 제한을 20MB로 늘렸습니다. • Windows 클라이언트가 이제 레거시 <code>.dcv</code> 파일을 검색하고 DCV 2016 Endstation(설치된 경우)을 시작합니다.

빌드 번호	새로운 기능	변경 및 버그 수정
		<ul style="list-style-type: none"> • DCV 단순 외부 인증자가 이제 환경에 설정된 것 대신에 항상 시스템 Python 인터프리터를 사용합니다. • DCV-GL의 다시 읽기 전략을 개선하여 성능 및 견고성을 개선했습니다. • DCV-GL이 이제 전면 버퍼 다시 읽기 후 창의 크기가 변경되었는지 확인합니다. 이렇게 하면 Coot 애플리케이션의 렌더링 문제가 수정됩니다.

DCV 2017.1-5870 — 2018년 8월 6일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> • nice-dcv-server: 5870 • nice-dcv-client: 5813 • nice-dcv-viewer (리눅스): 450 • nice-xdcv: 170 • nice-dcv-gl: 366 • nice-dcv-gltest: 198 • nice-dcv-simple-external-인증자: 53 	<p>Ubuntu 18.04용 패키지를 릴리스했습니다. 콘솔 모드에서 작업하는 경우 GDM은 필요한 X11 디스플레이 정보를 노출하지 않기 때문에 LightDM 또는 사용자가 선택한 다른 디스플레이 관리자를 사용하도록 시스템을 구성해야 합니다. 가상 세션은 이 제한의 영향을 받지 않습니다.</p>	<ul style="list-style-type: none"> • 이제 세션이 생성되면 라이선스 설정을 읽을 수 있습니다. 이를 통해 관리자는 서버를 다시 시작하지 않고도 이 설정을 변경할 수 있습니다. • 일부 시스템에서 프로그램이 예기치 않게 종료되는 Windows 클라이언트의 안정성 문제를 해결했습니다. • 가능한 오류 조건에서 로깅을 줄였습니다.

DCV 2017.1-5777 — 2018년 6월 29일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 5777 nice-dcv-client: 5777 nice-dcv-viewer (리눅스): 438 nice-xdcv: 166 nice-dcv-gl: 366 nice-dcv-gltest: 189 nice-dcv-simple-external-인증자: 51 	<ul style="list-style-type: none"> Linux 네이티브 클라이언트를 추가했습니다. 3DConnexion 마우스 및 USB 스토리지 디바이스에 대한 지원을 추가했습니다. 마지막 클라이언트의 연결이 끊어지면 Windows 세션이 자동으로 잠깁니다. 	<ul style="list-style-type: none"> Linux 버전의 성능 개선. 새 NVIDIA 드라이버에서 NvIFR 문제를 방지하기 위해 NVIDIA 디바이스의 기본 HW 인코더를 NVENC로 변경했습니다. Linux에서 스마트 카드 지원을 개선했습니다. Linux 콘솔 세션을 사용할 때 업로드된 파일에 대한 파일 권한을 수정했습니다.

DCV 2017.0-5600 — 2018년 6월 4일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 5600 nice-dcv-client: 5600 nice-xdcv: 160 nice-dcv-gl: 279 nice-dcv-gltest: 184 nice-dcv-simple-external-인증자: 48 	<ul style="list-style-type: none"> Linux에서 여러 모니터에 대한 지원을 추가했습니다. Windows 클라이언트 성능 개선. Chrome 66 이상에서 새로운 클립보드 API를 사용했습니다. Windows용 NVENC 인코더를 추가했습니다. 	<ul style="list-style-type: none"> 이제 EC2를 사용하려면 DCV 서버를 실행하는 인스턴스에서 S3에 도달할 수 있어야 합니다. 서버 프레임 처리 및 Windows 클라이언트 디코딩의 성능을 개선했습니다. 수정자와 관련된 키보드 문제 NumPad 및 차단된 수정자를 수정했습니다. Linux에서 외부 인증자를 사용할 때 파일 설명자 누출을 방지합니다.

빌드 번호	새로운 기능	변경 및 버그 수정
		<ul style="list-style-type: none"> 스마트 카드 연결에서 발생할 수 있는 오류를 수정했습니다.

DCV 2017.0-5121 — 2018년 3월 18일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 5121 nice-dcv-client: 5121 nice-xdcv: 146 nice-dcv-gi: 270 nice-dcv-glttest: 184 nice-dcv-simple-external-인증자: 46 	<ul style="list-style-type: none"> Windows 네이티브 클라이언트가 이제 DPI를 인식합니다. 상대 마우스 이동 모드에 대한 지원을 추가했습니다. 	<ul style="list-style-type: none"> Linux에서 Ansys cfx5solve가 중단되지 않도록 조치했습니다. Windows 10에서 에이전트가 중단될 수 있는 문제를 해결했습니다. 웹 클라이언트 사용자 인터페이스가 개선되었습니다. 도메인이 지정될 때 Windows 사용자 이름을 정규화했습니다. RHEL6에서 외부 인증자를 수정했습니다.

DCV 2017.0-4334 — 2018년 1월 24일

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-server: 4334 nice-dcv-client: 4334 nice-xdcv: 137 nice-dcv-gi: 254 nice-dcv-glttest: 184 	<ul style="list-style-type: none"> 키보드 처리를 개선했습니다. 세션을 닫으면 새 세션을 만들 수 없는 RHEL6의 DBus 문제를 해결했습니다. 기본 클라이언트에서 SOCKS5 프록시에 대한 지원을 개선했습니다.

빌드 번호	변경 및 버그 수정
<ul style="list-style-type: none"> nice-dcv-simple-external-인증자: 45 	<ul style="list-style-type: none"> Headwave 및 Chimera를 가상 세션에서 실행할 때 충돌을 일으키는 버그가 수정되었습니다. 가상 세션에서 글꼴 지원을 개선했습니다.

DCV 2017.0-4100 — 2017년 12월 18일

빌드 번호
<ul style="list-style-type: none"> nice-dcv-server: 4100 nice-dcv-client: 4100 nice-xdcv: 118 nice-dcv-gl: 229 nice-dcv-gltest: 158 nice-dcv-simple-external-인증자: 35

사용 설명서 기록

다음 표는 본 NICE DCV 릴리스 관련 설명서를 소개합니다.

변경 사항	설명	날짜
NICE DCV 버전 2023.1	NICE DCV는 윈도우 클라이언트 2023.1에 수정되었습니다. 자세한 정보는 DCV 2023.1-16388 — 2024년 3월 5일 을 참조하세요.	2024년 3월 5일
NICE DCV 버전 2023.1	NICE DCV가 2023.1 버전으로 수정되었습니다. 자세한 정보는 DCV 2023.1-16388 - 2023년 12월 19일 을 참조하세요.	2023년 12월 19일

변경 사항	설명	날짜
NICE DCV 버전 2023.1	이제 NICE DCV 2023.1 버전을 사용할 수 있습니다. 자세한 정보는 DCV 2023.1-16220 — 2023년 11월 9일 을 참조하세요.	2023년 11월 9일
NICE DCV 버전 2023.0	NICE DCV는 수명이 종료된 운영 체제를 더 이상 지원하지 않습니다.	2023년 6월 30일
NICE DCV 버전 2023.0	NICE DCV가 2023.0 버전으로 수정되었습니다. 자세한 정보는 DCV 2023.0-15487 — 2023년 6월 29일 을 참조하세요.	2023년 6월 29일
NICE DCV 버전 2023.0	NICE DCV가 2023.0 버전으로 수정되었습니다. 자세한 정보는 DCV 2023.0-15065 — 2023년 5월 3일 을 참조하세요.	2023년 5월 3일
NICE DCV 버전 2023.0	NICE DCV가 2023.0 버전으로 업데이트 및 수정되었습니다. 자세한 정보는 DCV 2023.0-15022 — 2023년 4월 21일 을 참조하세요.	2023년 4월 21일
NICE DCV 버전 2023.0	이제 NICE DCV 2023.0 버전을 사용할 수 있습니다. 자세한 정보는 DCV 2023.0-14852 — 2023년 3월 28일 을 참조하세요.	2023년 3월 28일

변경 사항	설명	날짜
NICE DCV 버전 2022.2	이제 NICE DCV 2022.2 버전을 사용할 수 있습니다. 자세한 정보는 DCV 2022.2-13907 — 2022년 11월 11일 을 참조하세요.	2022년 11월 11일
NICE DCV 버전 2022.1	이제 NICE DCV 2022.1 버전을 사용할 수 있습니다. 자세한 정보는 DCV 2022.1-13067 — 2022년 6월 29일 을 참조하세요.	2022년 6월 29일
NICE DCV 버전 2022.0	이제 NICE DCV 2022.0 버전을 사용할 수 있습니다. 자세한 정보는 DCV 2022.0-11954 — 2022년 2월 23일 을 참조하세요.	2022년 2월 23일
NICE DCV 버전 2021.3	이제 NICE DCV 2021.3 버전을 사용할 수 있습니다. 자세한 정보는 DCV 2021.3-11591 — 2021년 12월 20일 을 참조하세요.	2021년 12월 20일
NICE DCV 버전 2021.2	이제 NICE DCV 2021.2 버전을 사용할 수 있습니다. 자세한 정보는 DCV 2021.2-11048 — 2021년 9월 1일 을 참조하세요.	2021년 9월 1일
NICE DCV 버전 2021.1	이제 NICE DCV 2021.1 버전을 사용할 수 있습니다. 자세한 정보는 DCV 2021.1-10557 — 2021년 5월 31일 을 참조하세요.	2021년 5월 31일

변경 사항	설명	날짜
NICE DCV 버전 2021.0	이제 NICE DCV 2021.0 버전을 사용할 수 있습니다. 자세한 정보는 DCV 2021.0-10242 — 2021년 4월 12일 을 참조하세요.	2021년 4월 12일
NICE DCV 웹 클라이언트 SDK	이제 NICE DCV 웹 클라이언트 SDK를 사용할 수 있습니다. NICE DCV 웹 클라이언트 SDK는 JavaScript 최종 사용자가 실행 중인 NICE DCV 세션에 연결하고 상호 작용하는 데 사용할 수 있는 고유한 NICE DCV 웹 브라우저 클라이언트 애플리케이션을 개발하는 데 사용할 수 있는 라이브러리입니다. 자세한 내용은 NICE DCV 웹 클라이언트 SDK 개발자 안내서 를 참조하세요.	2021년 3월 24일
NICE DCV 버전 2020.2	이제 NICE DCV 2020.2 버전을 사용할 수 있습니다. 자세한 정보는 DCV 2020.2-9508 — 2020년 11월 11일 을 참조하세요.	2020년 11월 11일
NICE DCV 버전 2020.1	이제 NICE DCV 2020.1 버전을 사용할 수 있습니다. 자세한 정보는 DCV 2020.1-8942 — 2020년 8월 3일 을 참조하세요.	2020년 8월 3일

변경 사항	설명	날짜
NICE DCV 버전 2020.0	NICE DCV 2020.0 버전에는 새로운 Microsoft Edge 브라우저를 사용하는 서라운드 사운드 7.1, 터치 및 스타일러스, 다중 모니터에 대한 지원이 포함되어 있습니다. 자세한 내용은 NICE DCV 관리자 안내서의 NICE DCV 서버 설치 를 참조하세요.	2020년 4월 16일
HTTP 응답 헤더	추가 HTTP 응답 헤더를 보내도록 NICE DCV를 구성할 수 있습니다.	2019년 8월 26일
macOS 클라이언트	이제 NICE DCV에서 macOS 클라이언트를 제공합니다. 자세한 내용은 NICE DCV 사용 설명서의 macOS 클라이언트 를 참조하세요.	2019년 4월 18일
스마트 카드 캐싱	이제 NICE DCV 서버가 클라이언트로부터 수신한 스마트 카드 데이터를 캐싱하여 성능을 높일 수 있습니다. 자세한 내용은 NICE DCV 관리자 안내서의 스마트 카드 캐싱 구성 을 참조하세요.	2018년 10월 8일
Linux 클라이언트	NICE DCV는 RHEL 7, CentOS 7, SLES 12, Ubuntu 16.04/18.04용 Linux 클라이언트를 제공합니다. 자세한 내용은 NICE DCV 사용 설명서의 Linux 클라이언트 를 참조하세요.	2018년 8월 29일

변경 사항	설명	날짜
업데이트된 파라미터 참조	파라미터 참조가 업데이트되었습니다. 자세한 내용은 NICE DCV 관리자 안내서의 NICE DCV 서버 파라미터 참조 를 참조하세요.	2018년 8월 7일
USB 원격화	NICE DCV를 통해 클라이언트가 3D 포인팅 디바이스 또는 그래픽 태블릿 등 특수화된 USB 디바이스를 사용할 수 있습니다. 자세한 내용은 NICE DCV 관리자 안내서의 USB 원격화 활성화 를 참조하세요.	2018년 8월 7일
NICE DCV의 최초 릴리스	이 내용의 첫 번째 발행입니다.	2018년 6월 05일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.