



管理員指南

NICE DCV DCV 會話管理器



NICE DCV DCV 會話管理器: 管理員指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能隸屬於 Amazon，或與 Amazon 有合作關係，或由 Amazon 贊助。

Table of Contents

什麼是會話管理器？	1
工作階段管理員如何	1
功能	3
限制	3
定價	3
要求	3
網路和連線需求	5
設定	6
步驟 1：準備好的 DCV 伺服器	6
步驟 2：設置經紀人	7
步驟 3：設定代理程式	9
步驟 4：配置 NICE DCV DCV 服務器	13
步驟 5：驗證安裝	14
驗證代理程式	15
驗證經紀人	16
設定	17
縮放工作階段管	17
步驟 1：建立執行個體設定檔	18
步驟 2：準備負載平衡器的 SSL 憑證	19
步驟 3：建立 Broker 應用程式負載平衡器	19
步驟 4：啟動經紀人	20
步驟 5：建立代理程式應用程式負載平衡器	21
步驟 6：啟動代理程式	23
使用標籤	24
設定外部授權伺服器	25
配置代理持續性	30
設定代理程式以保留在 DynamoDB 上	30
配置代理持久在瑪麗亞布/MySQL	31
與 NICE DCV 連線閘道器整合	32
將工作階段管理員代理人設定為 NICE DCV 連線閘道的工作階段解析程式	32
選用-啟用 TLS 用戶端驗證	33
NICE DCV DCV 服務器-DNS 映射	35
與亞馬遜集成 CloudWatch	36
升級	38

升級 NICE DCV 工作階段管理員代理程式	38
升級 NICE DCV 工作階段管理員代理程式	40
代理 CLI 參考資料	42
register-auth-server	42
語法	43
選項	43
範例	43
list-auth-servers	44
語法	43
輸出	44
範例	43
unregister-auth-server	44
語法	43
選項	43
輸出	44
範例	43
register-api-client	46
語法	43
選項	43
輸出	44
範例	43
describe-api-clients	47
語法	43
輸出	44
範例	43
unregister-api-client	48
語法	43
選項	43
範例	43
renew-auth-server-api-鍵	49
語法	43
範例	43
generate-software-statement	50
語法	43
輸出	44
範例	43

describe-software-statements	51
語法	43
輸出	44
範例	43
deactivate-software-statement	52
語法	43
選項	43
範例	43
describe-agent-clients	53
語法	43
輸出	44
範例	43
unregister-agent-client	55
語法	43
選項	43
範例	43
register-server-dns-mappings	56
語法	43
選項	43
範例	43
describe-server-dns-mappings	57
語法	43
輸出	44
範例	43
組態檔案參考	59
程式組態檔案	59
代理程式組態檔案	73
版本備註和文件歷史記錄	78
版本備註	78
2023.1— 二二三年十一月九日	79
2023.0-15065— 二二二三年五月四日	79
2023.0-14852— 二二三年三月二十八日	79
2022.2-13907— 二〇〇二年十一月十一日	79
2022.1-13067— 二零二二年六月二十九日	80
2022.0-11952— 二零二二年二月二十三日	80
2021.3-11591— 二零二一年十二月二十日	80

2021.2-11445— 二零二一年十一月十八日	80
2021.2-11190— 二零二一年十月十一日	81
2021.2-11042— 二零二一年九月一日	81
2021.1-10557— 二零二一年五月三十一日	81
2021.0-10242— 二零二一年四月十二日	82
2020.2-9662— 二零二零年十二月四日	82
.....	83
文件歷史紀錄	83
.....	lxxxv

什麼是 NICE DCV 會話管理器？

NICE DCV 工作階段管理器是一組可安裝的軟體套件 (代理程式和代理程式) 和應用程式設計介面 (API)，可讓開發人員和獨立軟體廠商 (ISV) 輕鬆建立前端應用程式，以程式設計方式在 NICE DCV 伺服器叢集中建立和管理 NICE DCV 工作階段的生命週期。

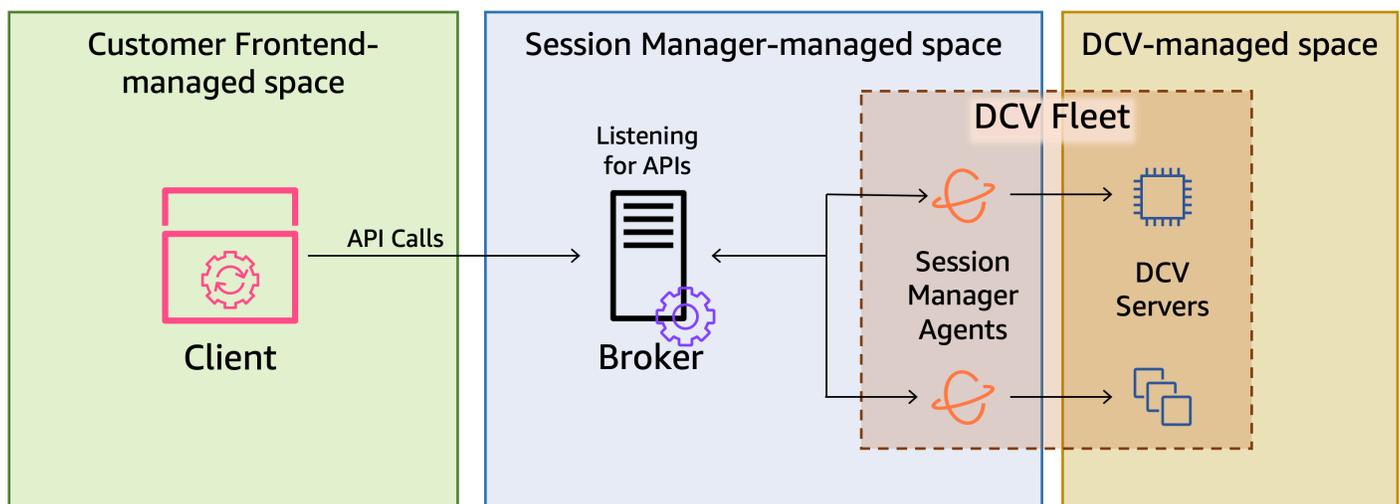
本指南說明如何安裝和設定工作階段管理員代理程式和代理程式。如需有關使用工作階段管理員 API 的詳細資訊，請參閱 NICE DCV 工作階段管理員開發人員指南。

主題

- [工作階段管理員如何](#)
- [功能](#)
- [限制](#)
- [定價](#)
- [NICE DCV DCV 會話管理器要求](#)

工作階段管理員如何

下圖顯示了會話管理器的高級組件。



中介裝置

代理程式是託管和公開工作階段管理員 API 的 Web 伺服器。它會從用戶端接收並處理 API 要求以管理 NICE DCV 工作階段，然後將指示傳送給相關代理程式。代理程式必須安裝在與 NICE DCV 伺服器分開的主機上，但用戶端必須可存取代理程式，而且必須能夠存取代理程式。

代理程式

代理程式會安裝在叢集中的每個 NICE DCV 伺服器上。代理程式會從代理程式接收指示，並在各自的 NICE DCV 伺服器上執行這些指示。代理程式也會監控 NICE DCV 伺服器的狀態，並將定期狀態更新傳送回代理程式。

API

會話管理器公開了一組 REST 應用程式編程接口 (API)，可用於管理 NICE DCV 服務器隊列上的 NICE DCV 會話。這些 API 託管在代理上並由公開。開發人員可以建置呼叫 API 的自訂工作階段管理用戶端。

客戶端

用戶端是您開發用來呼叫 Broker 公開的工作階段管理員 API 的前端應用程式或入口網站。使用者可以使用用戶端來管理叢集中 NICE DCV 伺服器上託管的工作階段。

訪問令牌

為了發出 API 請求，您必須提供訪問令牌。可以透過註冊的用戶端 API 從代理程式或外部授權伺服器要求權杖。要請求和訪問令牌，客戶端 API 必須提供有效的憑據。

用戶端 API

用戶端 API 是從工作階段管理員 API 定義 YAML 檔案產生的，使用 Swagger Codegen。客戶端 API 用於發出 API 請求。

NICE DCV DCV 會議

您必須在用戶端可以連線的 NICE DCV 伺服器上建立 NICE DCV 工作階段。用戶端只能在有作用中的工作階段時連線到 NICE DCV 伺服器。NICE DCV 支援主控台和虛擬工作階段。您可以使用工作階段管理員 API 來管理 NICE DCV 工作階段的生命週期。NICE DCV 工作階段可以處於下列其中一種狀態：

- CREATING— 代理人正在建立工作階段。
- READY工作階段已準備好接受用戶端連線。
- DELETING正在刪除工作階段。

- DELETED已刪除工作階段。
- UNKNOWN無法判斷工作階段的狀態。代理程式和代理程式可能無法通訊。

功能

DCV 工作階段管理員提供下列功能：

- 提供 NICE DCV 工作階段資訊 — 取得在多個 NICE DCV 伺服器上執行之工作階段的相關資訊。
- 管理多個 NICE DCV 工作階段的生命週期 — 透過一個 API 請求，跨多個 NICE DCV 伺服器建立或刪除多個使用者的多個工作階段。
- 支援標籤 — 建立工作階段時，使用自訂標籤以 NICE DCV 伺服器群組為目標。
- 管理多個 NICE DCV 工作階段的權限 — 使用一個 API 請求修改多個工作階段的使用者權限。
- 提供連線資訊 — 擷取 NICE DCV 工作階段的用戶端連線資訊。
- 支援雲端和內部部署 — 在AWS、內部部署或替代雲端伺服器上使用工作階段管理員。

限制

工作階段管理員不提供資源佈建功能。如果您在 Amazon EC2 執行個體上執行 NICE DCV，則可能需要使用其他AWS服務，例如 Amazon EC2 Auto Scaling 來管理基礎設施的擴展。

定價

對於執行 EC2 執行個體的AWS客戶，可免費使用工作階段管理員。

內部部署客戶需要 NICE DCV 加強版或 DCV 專業版增強版授權。如需有關如何購買 NICE DCV Plus 或 NICE DCV 專業版 Plus 授權的詳細資訊，請參閱[如何在 NICE 網站上購買](#)並尋找您所在地區的 NICE 經銷商或經銷商。若要允許所有內部部署客戶試用 DCV 工作階段管理員，只會從 NICE DCV 版本 2021.0 開始強制執行授權要求。

如需詳細資訊，請參閱 [《NICE DCV 管理員指南》](#) 中的授權 [NICE DCV 伺服器](#)。

NICE DCV DCV 會話管理器要求

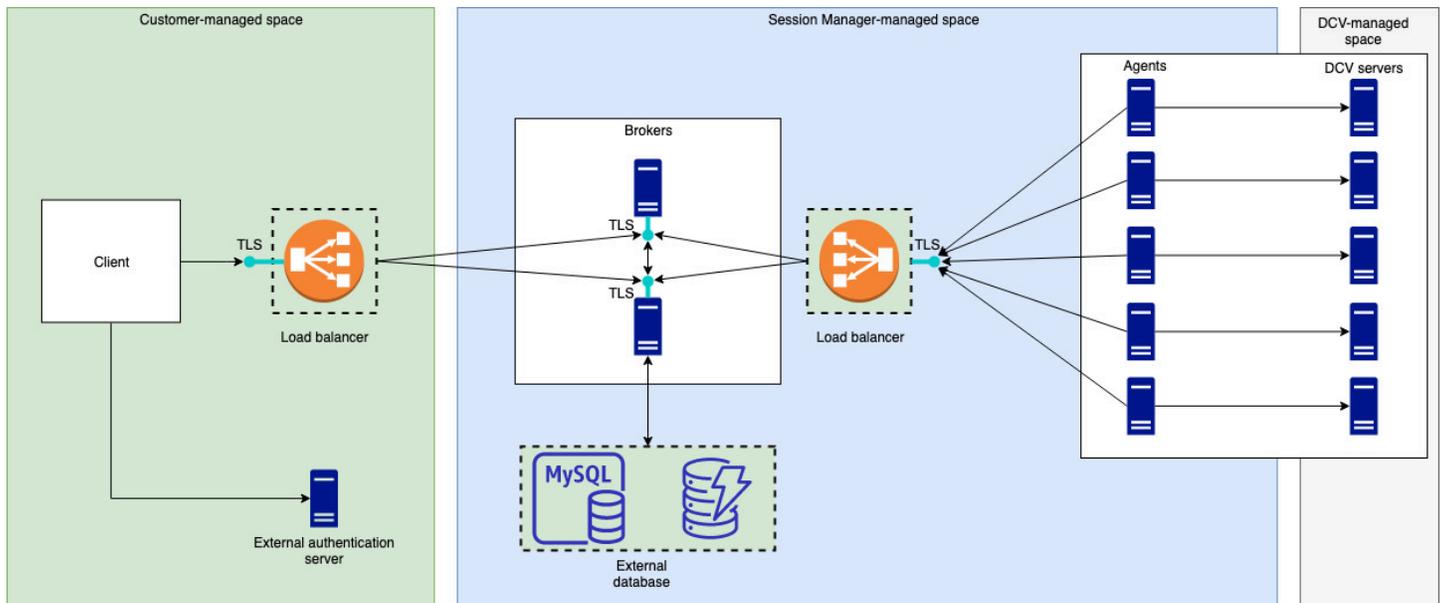
NICE DCV 工作階段管理員代理程式和代理程式具有下列需求。

	代理程式	客服人員
作業系統	<ul style="list-style-type: none"> • Amazon Linux 2 • CentOS 7.6 或以上版本 • CentOS Stream 8 • CentOS Stream 9 • RHEL 7.6 或更高版本 • RHEL 8.x • RHEL 9.x • 洛基 Linux 8.5 或更高版本 • 洛基 Linux 9.x • Ubuntu 20.04 • Ubuntu 	<ul style="list-style-type: none"> • Windows <ul style="list-style-type: none"> • Windows Server 2022 • Windows Server 2019 • Windows Server 2016 • Linux 伺服器 <ul style="list-style-type: none"> • Amazon Linux 2 • CentOS 7.6 或以上版本 • CentOS Stream 8 • CentOS Stream 9 • RHEL 7.6 或更高版本 • RHEL 8.x • RHEL 9.x • 洛基 Linux 8.5 或更高版本 • 洛基 Linux 9.x • Ubuntu 20.04 • Ubuntu • 使用 SP4 或更新版本的 SUSE Linux 企業版 12 • 瑞士企業版
架構	<ul style="list-style-type: none"> • 64 位元 • 64 位元手臂 	<ul style="list-style-type: none"> • 64 位元 • 64 位元 ARM (Amazon Linux 2, CentOS 7.x/8.x /9.x, 瑞氏 7.x/8.x /9.x 和洛基 8.x 只有 9.x) • 64 位元手臂
記憶體	8 GB	4 GB
NICE DCV DCV 版本	NICE DCV 救援隊 2020.2 及更高版本	NICE DCV 救援隊 2020.2 及更高版本

	代理程式	客服人員
其他需求	Java 11	-

網路和連線需求

下圖提供工作階段管理員網路和連線需求的高階概觀。



代理程式必須安裝在不同的主機上，但必須與 NICE DCV 伺服器上的代理程式具有網路連線。如果您選擇使用多個代理程式來改善可用性，則必須在不同的主機上安裝和設定每個代理程式，並使用一或多個負載平衡器來管理用戶端和代理程式之間的流量，以及代理程式和代理程式。經紀人還應該能夠相互通信，以便交換有關 NICE DCV 服務器和會話的信息。Broker 可以將其金鑰和狀態資料儲存在外部資料庫中，並在重新啟動或終止後提供此資訊。這有助於減輕遺失重要 Broker 資訊的風險，方法是將其保留在外部資料庫上。您可以稍後檢索它。如果您選擇擁有它，則必須設置外部數據庫並配置代理程序。支援數據庫、MariaDB 和 MySQL。您可以在[代理程式組態檔中找到組態參數](#)。

代理程式必須能夠與代理程式起始安全、持續的雙向 HTTPS 連線。

您的用戶端或前端應用程式必須能夠存取代理程式，才能呼叫 API。客戶端還應該能夠訪問您的身份驗證服務器。

設置 NICE DCV DCV 會話管理器

下節說明如何使用單一代理程式和多個代理程式安裝工作階段管理員。您可以使用多個代理程式來改善可擴充性和效能。如需詳細資訊，請參閱 [縮放工作階段管](#)。

若要設定 NICE DCV 工作階段管理員，請執行下列動作：

步驟

- [步驟 1：準備好的 DCV 伺服器](#)
- [步驟 2：設定 NICE DCV 工作階段管理員代理程式](#)
- [步驟 3：設定 NICE DCV 工作階段管理員代理程式](#)
- [步驟 4：將 NICE DCV 伺服器設定為使用代理程式做為驗證伺服器](#)
- [步驟 5：驗證安裝](#)

步驟 1：準備好的 DCV 伺服器

您必須擁有要使用工作階段管理員的 NICE DCV 伺服器群。如需有關安裝 NICE DCV 伺服器的詳細資訊，請參閱 [《NICE DCV 管理員指南》](#) 中的〈[安裝 NICE DCV 伺服器](#)〉。

在 Linux NICE DCV 伺服器上，工作階段管理員使用名為 dcvsmagent 的本機服務使用者。安裝工作階段管理員代理程式時，會自動建立此使用者。您必須授與此服務使用者 NICE DCV 的管理員權限，以便它可以代表其他使用者執行動作。若要授與工作階段管理員服務使用者管理員權限，請執行下列動作

若要新增 Linux NICE DCV 伺服器的本機服務使用者

1. 使用您偏好的文字編輯器開啟 /etc/dcv/dcv.conf。
2. 將 administrators 參數加入至 [security] 區段，並指定工作階段管理員使用者。例如：

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

3. 儲存並關閉檔案。
4. 停止並重新啟動 NICE DCV 伺服器。

工作階段管理員只能代表 NICE DCV 伺服器上已存在的使用者建立 NICE DCV 工作階段。如果要求為不存在的使用者建立工作階段，則要求會失敗。因此，您必須確保每個預定的最終使用者在 NICE DCV 伺服器上都有一個有效的系統使用者。

Tip

如果您打算搭配代理程式使用多個代理程式主機或 NICE DCV 伺服器，建議您透過執行下列步驟、使用完整組態建立主機的 Amazon 機器映像 (AMI)，然後使用 AMI 啟動剩餘的代理程式和 NICE DCV 伺服器。或者，您可以使用「AWS Systems Manager」在多個執行個體上遠端執行指令。

步驟 2：設定 NICE DCV 工作階段管理員代理程式

代理程式必須安裝在 Linux 主機上。如需有關支援 Linux 發行版本的詳細資訊，請參閱[NICE DCV DCV 會話管理器要求](#)。在與代理程式和 NICE DCV 伺服器主機分開的主機上安裝代理程式。主機可以安裝在不同的私人網路上，但必須能夠連線到代理程式並與其通訊。

安裝並啟動代理

1. Connect 至您要安裝代理程式的主機。
2. 套件是使用安全 GPG 簽章來進行數位簽署。若要允許套件管理員驗證套件簽章，您必須匯入 NICE GPG 金鑰。執行下列命令以匯入 NICE GPG 金鑰。

- Amazon Linux 2, RHEL, CentOS, 和岩石 Linux

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

- Ubuntu

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

3. 下載安裝套件。

- Amazon Linux 2、雷爾 7.x 及 CentOS 7.x

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

- RHEL 8.x、CentOS 8 流及落基 Linux 8.x 版

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

- Ubuntu 20.04

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

- Ubuntu

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

4. 安裝套件。

- Amazon Linux 2、雷爾 7.x 及 CentOS 7.x

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

- RHEL 8. X, 流 CentOS 8, 和岩石 Linux 8.x

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

- Ubuntu 20.04

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

- Ubuntu

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

5. 檢查預設的 Java 環境版本是否為 11

```
$ java -version
```

如果沒有，您可以明確設定代理程式將用來鎖定正確 Java 版本的 Java 主目錄。這樣就可以在 Broker 組態檔案 `broker-java-home` 中設定參數。如需詳細資訊，請參閱[代理程式組態檔](#)。

6. 啟動 Broker 服務，並確保每次執行個體啟動時都會自動啟動。

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

7. 將 Broker 的自我簽署憑證複本放在您的使用者目錄中。在下一個步驟中安裝代理程式時，您將需要它。

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

步驟 3：設定 NICE DCV 工作階段管理員代理程式

代理程式必須安裝在叢集中的所有 NICE DCV 伺服器主機上。代理程式可以安裝在 Windows 和 Linux 伺服器上。如需支援作業系統的詳細資訊，請參閱[NICE DCV DCV 會話管理器要求](#)。

必要條件

在安裝代理程式之前，必須先在主機上安裝 NICE DCV 伺服器。

Linux host

Note

工作階段管理員代理程式可用於[需求](#)中列出的 Linux 發行版和架構：

下列說明適用於在 64 位元 x86 主機上安裝代理程式。若要在 64 位元 ARM 主機上安裝代理程式，請將 `x86_64` 取代為 `aarch64`。對於 Ubuntu，請使用 `.arm64`

在 Linux 主機上安裝代理程式

1. 套件是使用安全 GPG 簽章來進行數位簽署。若要允許套件管理員驗證套件簽章，您必須匯入 NICE GPG 金鑰。執行下列命令以匯入 NICE GPG 金鑰。
 - Amazon Linux 2、RHEL、CentOS 和瑞士企業版

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

- Ubuntu

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

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

2. 下載安裝套件。

- Amazon Linux 2、雷爾 7.x 及 CentOS 7.x

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

- RHEL 8.x、CentOS 8 流及落基 Linux 8.x 版

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

- Ubuntu 20.04

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

- Ubuntu

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

- 瑞士企業版

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

- 瑞士企業版

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

3. 安裝套件。

- Amazon Linux 2、雷爾 7.x 及 CentOS 7.x

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

- RHEL 8.x、CentOS 8 流及落基 Linux 8.x 版

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

- Ubuntu 20.04

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

- Ubuntu

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

- 瑞士企業版

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

- 瑞士企業版

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

4. 將 Broker 的自我簽署憑證 (您在上一個步驟中複製的) 複本放在代理程式的 `/etc/dcv-session-manager-agent/`目錄中。
5. `/etc/dcv-session-manager-agent/agent.conf`使用您偏好的文字編輯器開啟，然後執行下列動作。
 - 對於`broker_host`，指定安裝代理程式之主機的 DNS 名稱。

Important

如果代理程式在 Amazon EC2 執行個體上執行，`broker_host`您必須指定執行個體的私有 IPv4 地址。

- (選擇性) 對於 `broker_port`，指定要與代理人通訊的連接埠。依預設，代理程式和代理程式會透過連接埠通訊8445。只有在需要使用不同的連接埠時才變更此選項。如果您確實變更了它，請確定 Broker 已設定為使用相同的連接埠。
- 對於 `ca_file`，指定您在上一個步驟中複製的憑證檔案的完整路徑。例如：

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

或者，如果您想要停用 TLS 驗證，請 `tls_strict` 將設定為 `false`。

6. 儲存並關閉檔案。
7. 執行下列命令以啟動代理程式。

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

Windows host

在 Windows 主機上安裝代理程式

1. 下載[代理程式安裝程式](#)。
2. 執行安裝程式。在 Welcome (歡迎) 畫面上，選擇 Next (下一步)。
3. 在 EULA 畫面上，請仔細閱讀授權合約，如果您同意，請選取 [我接受條款]，然後選擇 [下一步]。
4. 若要開始安裝，請選擇 [安裝]。
5. 將 Broker 的自我簽署憑證 (您在上一個步驟中複製的) 複本放在代理程式上的 `C:\Program Files\NICE\DCVSessionManagerAgent\conf` 資料夾中。
6. `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` 使用偏好的文字編輯器開啟，然後執行下列動作：
 - 對於 `broker_host`，指定安裝代理程式之主機的 DNS 名稱。

Important

如果代理程式在 Amazon EC2 執行個體上執行，`broker_host` 您必須指定執行個體的私有 IPv4 地址。

- (選擇性) 對於 `broker_port`，指定要與代理人通訊的連接埠。依預設，代理程式和代理程式會透過連接埠通訊8445。只有在需要使用不同的連接埠時才變更此選項。如果您確實變更了它，請確定 Broker 已設定為使用相同的連接埠。
- 對於 `ca_file`，指定您在上一個步驟中複製的憑證檔案的完整路徑。例如：

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

或者，如果您想要停用 TLS 驗證，請 `tls_strict` 將設定為 `false`。

7. 儲存並關閉檔案。
8. 停止並重新啟動「代理程式」服務，以使變更生效。在命令提示字元中執行下列命令。

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

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

步驟 4：將 NICE DCV 伺服器設定為使用代理程式做為驗證伺服器

將 NICE DCV 伺服器設定為使用代理程式做為外部驗證伺服器，以驗證用戶端連線權杖。您也必須設定 NICE DCV 伺服器，以信任代理程式的自我簽署 CA。

Linux NICE DCV server

若要新增 Linux NICE DCV 伺服器的本機服務使用者

1. 使用您偏好的文字編輯器開啟 `/etc/dcv/dcv.conf`。
2. 將 `ca-file` 和 `auth-token-verifier` 參數加入至 `[security]` 區段。

針對 `ca-file`，指定您在上一個步驟中複製到主機的 Broker 自我簽署 CA 路徑。

對於 `auth-token-verifier`，請以下列格式指定 Broker 上 Token 驗證器的 URL：`https://broker_ip_or_dns:port/agent/validate-authentication-token` 指定用於代理程式-代理程式通訊的連接埠，預設為 8445。如果您在 Amazon EC2 執行個體上執行代理程式，則必須使用私有 DNS 或私有 IP 地址。

例如

```
[security]
```

```
ca-file="/etc/dcv-session-manager-agent/broker_cert.pem"
auth-token-verifier="https://my-sm-broker.com:8445/agent/validate-
authentication-token"
```

3. 儲存並關閉檔案。
4. 停止並重新啟動 NICE DCV 伺服器。如需詳細資訊，請參閱 [《NICE DCV 管理員指南》中的停止 NICE DCV 伺服器和啟動 NICE DCV 伺服器](#)。

Windows NICE DCV server

在視窗上 NICE DCV DCV 服務器

1. 打開 Windows 註冊表編輯器，然後導航到 HKEY_用戶 /S-1-5-18 軟件/G 設置/通信/優化/電視/DCV /安全/密鑰。
2. 開啟 CA 檔案參數。針對值資料，指定您在上一個步驟中複製到主機的 Broker 自我簽署 CA 路徑。

Note

如果該參數不存在，請創建一個新的字符串參數並命名它ca-file。

3. 開啟auth-token-verifier參數。對於「值」資料，請以下列格式指定代理程式上權杖驗證器的 URL : `https://broker_ip_or_dns:port/agent/validate-authentication-token`指定用於代理程式-代理程式通訊的連接埠，預設為 8445。如果您在 Amazon EC2 執行個體上執行代理程式，則必須使用私有 DNS 或私有 IP 地址。

Note

如果該參數不存在，請創建一個新的字符串參數並命名它auth-token-verifier。

4. 選擇 OK (確定) 並關閉 Windows 登錄編輯程式。
5. 停止並重新啟動 NICE DCV 伺服器。如需詳細資訊，請參閱 [《NICE DCV 管理員指南》中的停止 NICE DCV 伺服器和啟動 NICE DCV 伺服器](#)。

步驟 5：驗證安裝

主題

- [驗證代理程式](#)
- [驗證經紀人](#)

驗證代理程式

安裝代理程式和代理程式之後，請確定代理程式正在執行，而且它能夠連線至代理程式。

代理程式主機

要執行的命令取決於版本。

- 自第二版以來

從「代理程式」主機執行下列命令：

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

- 2022.0 版本之前的版本

從代理程式主機執行下列命令，並指定目前的年、月和日。

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

例如

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

如果代理程式正在執行，而且可以連線到 Broker，則命令應該會傳回success。

如果命令傳回不同的輸出，請檢查代理程式記錄檔以取得詳細資訊。日誌文件位於此處：`/var/log/dcv-session-manager-agent/`。

視窗代理程式

開啟位於中的代理程式記錄檔C:\ProgramData\NICE\DCVSessionManagerAgent\log。

如果記錄檔包含類似下列的行，表示代理程式正在執行，而且可以連線至 Broker。

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

如果您的記錄檔沒有類似的行，請檢查記錄檔是否有錯誤。

驗證經紀人

安裝代理程式和代理程式之後，請確定您的 Broker 正在執行，而且可從使用者和前端應用程式存取代理程式。

從應該能夠連線到 Broker 的電腦上，執行下列命令：

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

如果驗證成功，代理程式會傳回下列資訊：

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

設定 NICE DCV 工作階段管理員

本節說明如何執行工作管理管理員。

主題

- [縮放工作階段管](#)
- [使用標籤以 NICE DCV 伺服器為目標](#)
- [設定外部授權伺服器](#)
- [配置代理持續性](#)
- [與 NICE DCV 連線閘道器整合](#)
- [與亞馬遜集成 CloudWatch](#)

縮放工作階段管

若要啟用高可用性並改善效能，您可以設定工作階段管理員使用多個代理程式和代理程式。如果您確實打算使用多個代理程式和代理程式，建議您僅安裝和設定一個代理程式和代理主機，從這些主機建立 Amazon Machines 映像 (AMI)，然後從 AMI 啟動剩餘的主機。

依預設，工作階段管理員支援使用多個代理程式，無需任何其他設定。但是，如果您打算使用多個代理程式，則必須使用負載平衡器來平衡前端用戶端和代理程式之間以及代理程式和代理程式之間的流量。負載平衡器設定和組態完全由您擁有和管理。

下節說明如何將工作階段管理員設定為搭配應用 Application Load Balancer 使用多個主機。

步驟

- [步驟 1：建立執行個體設定檔](#)
- [步驟 2：準備負載平衡器的 SSL 憑證](#)
- [步驟 3：建立 Broker 應用程式負載平衡器](#)
- [步驟 4：啟動經紀人](#)
- [步驟 5：建立代理程式應用程式負載平衡器](#)
- [步驟 6：啟動代理程式](#)

步驟 1：建立執行個體設定檔

您必須將執行個體設定檔附加至代理程式和代理程式主機，以授與其使用 Elastic Load Balancing API 的權限。如需詳細資訊，請參閱 [Amazon EC2 使用者指南中的適用於 Amazon EC2 的 IAM 角色](#)。

建立執行個體設定檔

1. 建立 AWS Identity and Access Management (IAM) 角色，以定義要在執行個體設定檔中使用的許可。請使用下列信任原則：

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

然後附上以下策略：

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

```
]
}
```

如需詳細資訊，請參閱 [IAM 使用者指南中的建立 IAM 角色](#)。

2. 建立新的執行個體設定檔。若要取得更多資訊，請參閱《指令參考》中的〈[建立執行個體-設定檔](#)〉。AWS CLI
3. 將 IAM 角色新增至執行個體設定檔。如需詳細資訊，請參閱命令參考中的[新增角色至執行個體設定檔](#)。AWS CLI
4. 將執行個體設定檔附加至 Broker 主機。[如需詳細資訊，請參閱 Amazon EC2 使用者指南中的將 IAM 角色附加到執行個體](#)。

步驟 2：準備負載平衡器的 SSL 憑證

當您對負載平衡器接聽程式使用 HTTPS 時，您必須在負載平衡器上部署 SSL 憑證。負載平衡器會使用此憑證來終止連線，並解密來自用戶端的請求，然後才將這些請求傳送到目標。

若要準備 SSL 憑證

1. 建立私有憑證授權單位 (CA) Certificate Manager 專用憑證授權單位 (ACM PCA)。如需詳細資訊，請參閱 [Certificate Manager 專用 AWS 憑證授權單位使用者指南中的建立 CA](#) 的程序。
2. 安裝 CA。如需詳細資訊，請參閱 [Certificate Manager 專用憑證授權單位使用者指南中的安裝根 CA 憑證](#)。AWS
3. 要求 CA 簽署的新私有憑證。對於網域名稱，請使用 `*.region.elb.amazonaws.com` 並指定您要在其中建立負載平衡器的區域。如需詳細資訊，請參閱 [Certificate Manager 專用憑證授權單位使用者指南中的要求私人憑證](#)。AWS

步驟 3：建立 Broker 應用程式負載平衡器

建立應用程式負載平衡器，以平衡前端用戶端和代理程式之間的流量。

建立負載平衡器

1. 前往 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。

在瀏覽窗格中，選擇 [Load Balancer]，然後選擇 [建立負載平衡器]。針對負載平衡器類型，選擇 Application Load Balancer

2. 在 Step 1: Configure Load Balancer (步驟 1：設定負載平衡器) 中，執行下列動作：

- a. 在名稱中，輸入負載平衡器的描述性名稱。
 - b. 針對「方案」，選取網際網路對向。
 - c. 對於 Load Balancer 通訊協定，選取 HTTPS，然後在 Load Balancer 連接埠中輸入8443。
 - d. 對於 VPC，請選取要使用的 VPC，然後選取該 VPC 中的所有子網路。
 - e. 選擇下一步。
3. 對於步驟 2：設定安全性設定，請執行下列動作：
- a. 針對 [憑證類型]，選擇 [從 ACM 選擇憑證]。
 - b. 在「憑證名稱」中，選取您先前要求的私人憑證。
 - c. 選擇下一步。
4. 對於步驟 3：設定安全群組、建立新的安全群組，或選取現有的安全性群組，以允許前端用戶端和代理程式之間透過 HTTPS 和 8443 連接埠允許輸入和輸出流量。
- 選擇下一步。
5. 對於步驟 4：設定路由，請執行下列操作：
- a. 針對目標群組，選取新增目標群組。
 - b. 針對 Name (名稱)，輸入目標群組的名稱。
 - c. 針對「目標類型」，選擇「例證」
 - d. 針對通訊協定，選取 HTTPS。針對連接埠，輸入 8443。針對通訊協定版本，請選擇 HTTP1。
 - e. 對於健全狀況檢查通訊協定，請選擇 HTTPS，然後在路徑中輸入/health。
 - f. 選擇下一步。
6. 對於步驟 5：註冊目標，選擇下一步。
7. 選擇建立。

步驟 4：啟動經紀人

建立初始代理程式並將其設定為使用負載平衡器、從代理程式建立 AMI，然後使用 AMI 啟動剩餘的代理程式。這可確保所有代理程式都設定為使用相同的 CA 和相同的負載平衡器組態。

啟動經紀人

1. 啟動並設定初始代理主機。如需有關安裝和設定 Broker 的詳細資訊，請參閱[步驟 2：設定 NICE DCV 工作階段管理員代理程式](#)。

Note

由於我們使用的是應用程式負載平衡器，因此不需要代理的自簽名證書。

2. Connect 至代理程式，`/etc/dcv-session-manager-broker/session-manager-broker.properties`使用您偏好的文字編輯器開啟，然後執行下列動作：
 - a. 通過在行的開頭放置一個散列 (#) 註釋掉`broker-to-broker-discovery-addresses`參數。
 - b. 在中`broker-to-broker-discovery-aws-region`，輸入您在其中建立應用程式負載平衡器的區域。
 - c. 在中`broker-to-broker-discovery-aws-alb-target-group-arn`，輸入與代理程式負載平衡器相關聯之目標群組的 ARN。
 - d. 儲存並關閉檔案。
3. 停止代理程式執行個體。
4. 從停止的代理程式執行個體建立 AMI。[如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的從執行個體建立 Linux AMI](#)。
5. 使用 AMI 啟動剩餘的經紀人。
6. 將您建立的執行個體設定檔指派給所有 Broker 執行個體。
7. 為所有 Broker 執行個體指派一個安全群組，該群組允許代理人和代理人負載平衡器網路流量。如需有關網路連接埠的詳細資訊，請參閱[代理人組態檔](#)。
8. 將所有 Broker 執行個體註冊為 Broker 負載平衡器的目標。如需詳細資訊，請參閱[《應用程式負載平衡器使用者指南》中的向目標群組註冊目標](#)。

步驟 5：建立代理程式應用程式負載平衡器

建立應用程式負載平衡器以平衡代理程式和代理程式。

建立負載平衡器

1. 前往 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。

在瀏覽窗格中，選擇 [Load Balancer]，然後選擇 [建立負載平衡器]。針對負載平衡器類型，選擇 Application Load Balancer

2. 在 Step 1: Configure Load Balancer (步驟 1：設定負載平衡器) 中，執行下列動作：
 - a. 在名稱中，輸入負載平衡器的描述性名稱。
 - b. 針對「方案」，選取網際網路對向。
 - c. 對於 Load Balancer 通訊協定，選取 HTTPS，然後在 Load Balancer 連接埠中輸入 8445。
 - d. 對於 VPC，請選取要使用的 VPC，然後選取該 VPC 中的所有子網路。
 - e. 選擇下一步。
3. 對於步驟 2：設定安全性設定，請執行下列動作：
 - a. 針對 [憑證類型]，選擇 [從 ACM 選擇憑證]。
 - b. 在「憑證名稱」中，選取您先前要求的私人憑證。
 - c. 選擇下一步。
4. 對於步驟 3：設定安全群組、建立新的安全群組，或選取現有的安全性群組，以允許代理程式和代理程式透過 HTTPS 和 8445 進行輸入和輸出流量。
選擇下一步。
5. 對於步驟 4：設定路由，請執行下列操作：
 - a. 針對目標群組，選取新增目標群組。
 - b. 針對 Name (名稱)，輸入目標群組的名稱。
 - c. 針對「目標類型」，選擇「例證」
 - d. 針對通訊協定，選取 HTTPS。針對連接埠，輸入 8445。針對通訊協定版本，請選擇 HTTP1。
 - e. 對於健全狀況檢查通訊協定，請選擇 HTTPS，然後在路徑中輸入/health。
 - f. 選擇下一步。
6. 在步驟 5：註冊目標中，選取所有代理程式執行處理，然後選擇新增至已註冊。選擇下一步：檢閱。
7. 選擇建立。

步驟 6：啟動代理程式

建立初始代理程式並將其設定為使用負載平衡器、從代理程式建立 AMI，然後使用 AMI 啟動剩餘的代理程式。這可確保所有代理程式都設定為使用相同的負載平衡器組態。

啟動代理程式

1. 準備好的 DCV 伺服器。如需詳細資訊，請參閱 [步驟 1：準備好的 DCV 伺服器](#)。
2. 放置在中建立的 CA 公開金鑰的複本 [步驟 2：準備負載平衡器的 SSL 憑證](#)。選擇或建立任何使用者可讀取的目錄。CA 公鑰文件必須是由任何用戶以及可讀。
3. 安裝並設定代理程式。如需安裝和設定代理程式的詳細資訊，請參閱 [步驟 3：設定 NICE DCV 工作階段管理員代理程式](#)。

Important

修改代理程式組態檔時：

- 針對 `broker_host` 參數，輸入代理程式負載平衡器的 DNS
- 針對 `ca_file` 參數，輸入在上一步驟中建立之 CA 公開金鑰檔案的路徑

4. 將 NICE DCV 伺服器設定為使用代理程式做為驗證伺服器。如需詳細資訊，請參閱 [步驟 4：將 NICE DCV 伺服器設定為使用代理程式做為驗證伺服器](#)。

Important

修改 NICE DCV 伺服器組態檔案時：

- 針對 `ca-file` 參數，輸入與上一步所使用之 CA 公開金鑰檔案相同的路徑
- 針對 `auth-token-verifier` 參數，請針對代理程式負載平衡器使用代理程式負載平衡器的 *DNS*

5. 停止代理程式執行個體。
6. 從停止的代理程式執行個體建立 AMI。如需詳細資訊，請參閱 [Amazon EC2 Linux 執行個體使用者指南中的從執行個體建立 Linux AMI](#)。
7. 使用 AMI 啟動剩餘的代理程式，並將您建立的執行個體設定檔指派給所有代理程式。
8. 指派可讓代理程式負載平衡器網路流量至所有代理程式執行個體的安全群組。如需網路連接埠的詳細資訊，請參閱 [代理程式組態檔](#)。

使用標籤以 NICE DCV 伺服器為目標

您可以將自訂標記指派給工作階段管理員代理程式，以協助識別和分類它們以及與它們相關聯的 NICE DCV 伺服器。建立新的 NICE DCV 工作階段時，您可以根據指派給各自代理程式的標籤來鎖定一組 NICE DCV 伺服器。如需有關如何根據代理程式標籤鎖定 NICE DCV 伺服器的詳細資訊，請參閱工作階段管理員開發人員指南[CreateSessionRequests](#) 中的。

標籤由標籤鍵和值配對組成，您可以使用對您的使用案例或環境有意義的任何資訊配對。您可以選擇根據代理程式主機硬體組態標記代理程式。例如，您可以使用具有 4 GB 記憶體的主機來標記所有代理程式ram=4GB。或者，您可以根據目的標記代理程式。例如，您可以使用，標記在生產主機上執行的所有代理程式purpose=production。

若要將標記指派給代理程式

1. 使用您偏好的文字編輯器，建立例如，建立新檔案，然後為其命名agent_tags.toml。檔案類型必須是.toml，且檔案內容必須以 TOML 檔案格式指定。
2. 在檔案中，使用key=value格式將每個新的標籤鍵和值配對新行加入。例如：

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

3. 開啟代理程式組態檔案 (/etc/dcv-session-manager-agent/agent.conf適用C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf於 Linux 或 Windows 系統)。對於tags_folder，並指定標籤檔案所在目錄的路徑。

如果目錄包含多個標記檔案，則跨檔案定義的所有標籤都會套用代理程式。檔案會依字母順序讀取。如果多個檔案包含具有相同索引鍵的標籤，則會以上次讀取檔案中的值覆寫該值。

4. 儲存並關閉檔案。
5. 停止並重新啟動代理程式。

- Windows

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

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

- Linux

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

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

設定外部授權伺服器

授權伺服器是負責驗證和授權用戶端 SDK 和代理程式的伺服器。

依預設，工作階段管理員會使用代理程式做為授權伺服器，為用戶端 SDK 和代理程式的軟體陳述式產生 OAuth 2.0 存取權杖。如果您使用 Broker 做為授權伺服器，則不需額外進行設定。

您可以將工作階段管理員設定為使用 Amazon Cognito 做為外部授權伺服器，而不是代理人。如需有關 Amazon Cognito 的詳細資訊，請參閱 [Amazon Cognito 開發人員指南](#)。

使用 Amazon Cognito 作為授權伺服器

1. 建立新的 Amazon Cognito 使用者集區。如需有關使用者集區的詳細資訊，請參閱 [Amazon Cognito 開發人員指南中的 Amazon Cognito 功能](#)。

使用指 `create-user-pool` 令，並指定儲存池名稱以及要在其中建立該集區的「區域」。

在此範例中，我們為池命名，`dcv-session-manager-client-app` 然後在中建立它 `us-east-1`。

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

範例輸出

```
{  
  "UserPoolClient": {  
    "UserPoolId": "us-east-1_QLEXAMPLE",  
    "ClientName": "dcv-session-manager-client-app",  
    "ClientId": "15hhd8jjj74hf32f24uEXAMPLE",  
    "LastModifiedDate": 1602510048.054,  
    "CreationDate": 1602510048.054,  
    "RefreshTokenValidity": 30,  
    "AllowedOAuthFlowsUserPoolClient": false  
  }  
}
```

```
}
```

請記下userPoolId，在後續步驟您將會用到它。

2. 建立使用者集區的新網域。使用命[create-user-pool-domain](#)令，並指定您在第一個步驟中所建立使用者集區userPoolId的網域名稱和。

在這個例子中，域名是mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE，我們在中創建它us-east-1。

```
$ aws cognito-idp create-user-pool-domain --domain mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE --user-pool-id us-east-1_QLEXAMPLE --region us-east-1
```

範例輸出

```
{
  "DomainDescription": {
    "UserPoolId": "us-east-1_QLEXAMPLE",
    "AWSAccountId": "123456789012",
    "Domain": "mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE",
    "S3Bucket": "aws-cognito-prod-pdx-assets",
    "CloudFrontDistribution": "dpp0gtexample.cloudfront.net",
    "Version": "20201012133715",
    "Status": "ACTIVE",
    "CustomDomainConfig": {}
  }
}
```

使用者集區網域的格式如

下：https://domain_name.auth.region.amazoncognito.com在此範例中，使用者集區網域為<https://mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE.auth.us-east-1.amazoncognito.com>。

3. 建立使用者集區用戶端。使用命[create-user-pool-client](#)令並指定您建立userPoolId的使用者集區的名稱、用戶端的名稱，以及要在其中建立它的區域。此外，請包含選--generate-secret項，指定您想要產生所建立使用者集區用戶端的秘密。

在這種情況下，客戶名稱是dcv-session-manager-client-app，我們在us-east-1區域中創建它。

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

範例輸出

```
{
  "UserPoolClient": {
    "UserPoolId": "us-east-1_QLEXAMPLE",
    "ClientName": "dcv-session-manager-client-app",
    "ClientId": "219273hp6k2ut5cugg9EXAMPLE",
    "ClientSecret": "1vp5e8nec7cbf4m9me55mbmht91u61h1h0a78rq1qki11EXAMPLE",
    "LastModifiedDate": 1602510291.498,
    "CreationDate": 1602510291.498,
    "RefreshTokenValidity": 30,
    "AllowedOAuthFlowsUserPoolClient": false
  }
}
```

Note

記下ClientId和ClientSecret。您需要將此信息提供給開發人員，以了解他們何時請求 API 請求的訪問令牌。

4. 建立使用者集區的新的 OAuth2.0 資源伺服器。資源伺服器是用於存取受保護之資源的伺服器。它處理訪問令牌的身份驗證請求。

使用命 [create-resource-server](#) 令並指定使用者集區userPoolId的、資源伺服器的唯一識別碼和名稱、範圍以及要在其中建立它的區域。

在這個例子中，我們使用dcv-session-manager作為標識符和名稱，我們使用sm_scope作為範圍名稱和描述。

```
$ aws cognito-idp create-resource-server --user-pool-id us-east-1_QLEXAMPLE
--identifier dcv-session-manager --name dcv-session-manager --scopes
ScopeName=sm_scope,ScopeDescription=sm_scope --region us-east-1
```

範例輸出

```
{
```

```

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

```

5. 更新使用者集區用戶端。

使用 [update-user-pool-client](#) 命令。指定使用 `userPoolId` 使用者集區、使用者集區用戶端 `ClientId` 的，以及區域。對於 `--allowed-o-auth-flows`，指定 `client_credentials` 以指示用戶端應該使用用戶端 ID 和用戶端密碼的組合，從 Token 端點取得存取權杖。對於 `--allowed-o-auth-scopes`，指定資源伺服器識別碼和範圍名稱，如下所示：`resource_server_identifier/scope_name` 包含，表示 `--allowed-o-auth-flows-user-pool-client` 允許用戶端與 Cognito 使用者集區互動時遵循 OAuth 通訊協定。

```

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

```

範例輸出

```

{
  "UserPoolClient": {
    "UserPoolId": "us-east-1_QLEXAMPLE",
    "ClientName": "dcv-session-manager-client-app",
    "ClientId": "219273hp6k2ut5cugg9EXAMPLE",
    "ClientSecret": "1vp5e8nec7cbf4m9me55mbmht91u61h1h0a78rq1qki11EXAMPLE",
    "LastModifiedDate": 1602512103.099,
    "CreationDate": 1602510291.498,
    "RefreshTokenValidity": 30,
    "AllowedOAuthFlows": [
      "client_credentials"
    ],
    "AllowedOAuthScopes": [
      "dcv-session-manager/sm_scope"
    ]
  }
}

```

```
    ],
    "AllowedOAuthFlowsUserPoolClient": true
  }
}
```

Note

使用者集區現在已準備好提供和驗證存取權杖。在此範例中，授權伺服器的 URL 為 `https://cognito-idp.us-east-1.amazonaws.com/us-east-1_QLEXAMPLE/.well-known/jwks.json`。

6. 測試組態。

```
$ curl -H "Authorization: Basic `echo -
n 219273hp6k2ut5cugg9EXAMPLE:1vp5e8nec7cbf4m9me55mbmht91u61h1h0a78rq1qki11EXAMPLE
| base64`" -H "Content-Type: application/x-www-form-urlencoded" -X
POST "https://mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE.auth.us-
east-1.amazoncognito.com/oauth2/token?grant_type=client_credentials&scope=dcv-
session-manager/sm_scope"
```

範例輸出

```
{
  "access_token": "eyJraWQiOiJGQ0VaRFPjJUuptT3NSaW41MmtqaDdEbTZyYb0RnSTQ5b2VUT0cxUU11Q2VJPSIsImF0IjoiZkfi0HIDsd6audjTXKzHlZGScr6R0dZtId5dThkpEziSx0YwiiWe9crAlqoazlDcCsUJHIXDtGKW64pSj3-
uQQGg1Jv_tyVjhrA4JbD0k67WS2V9NW-
uZ7t4zwwaUm0i3KzpBmi54fpVgPaewiVlUm_aS4LUFcWT6hVJjiZF7om7984qb2g0a14iZxpXPBJTZX_gtG9EtvnS9u
",
  "expires_in": 3600,
  "token_type": "Bearer"
}
```

7. 使用 `register-auth-server` 指令註冊外部授權伺服器，以便與 Broker 搭配使用。

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

開發人員現在可以使用伺服器來請求存取符記。請求存取符記時，請提供用戶端 ID、用戶端秘密和此處產生的伺服器 URL。有關請求訪問令牌的更多信息，請參閱 [NICE DCV 會話管理器開發人員指南中的創建獲取訪問令牌並發出 API 請求](#)。

配置代理持續性

會話管理器代理程序支持與外部數據庫集成。外部資料庫允許工作階段管理員保留狀態資料和金鑰，以便之後可以使用它們。事實上，Broker 資料會分散在叢集上，因此如果主機需要重新開機或叢集終止，則容易遺失資料。啟用此功能後，您可以新增和移除代理程式節點。此外，您可以停止叢集並重新啟動叢集，而不需要重新產生金鑰或遺失開啟或關閉 NICE DCV Server 的相關資訊。

可以將下列類型的資訊設定為持續存在：

- 設定工作階段以建立與用戶端連線的金鑰
- 飛行中工作階段資料
- NICE DCV DCV 服務器狀態

NICE DCV DCV 工作階段管理員支援動態資料庫、MariaDB 和 MySQL 資料庫。您必須設定和管理這些資料庫之一，才能使用此功能。如果您的代理機器託管在 Amazon EC2 上，我們建議您使用 DynamoDB 做為外部資料庫，因為它不需要任何額外的設定。

Note

執行外部資料庫時，可能會產生額外費用。若要查看 DynamoDB 定價的相關資訊，請參閱[佈建容量的定價](#)。

設定代理程式以保留在 DynamoDB 上

設定代理程式以開始將其資料儲存在 DynamoDB 上：

1. `/etc/dcv-session-manager-broker/session-manager-broker.properties` 使用您偏好的文字編輯器開啟，然後執行下列命令：
 - 設定 `enable-persistence = true`
 - 設定 `persistence-db = dynamodb`
 - 用於 `dynamodb-region` 指定要存儲包含代理數據的表的 `&aws; 区域`。如需支援區域的清單，請參閱 [DynamoDB 服務端點](#)。
 - 用於 `dynamodb-table-rcu` 指定每個表格支援的讀取容量單位 (RCU) 數量。如需有關 RCU 的詳細資訊，請參閱 [DynamoDB 佈建的容量](#)。

- 用於 `dynamodb-table-wcu` 指定每個表格支援的寫入容量單位 (WCU) 數量。如需有關 WCU 的詳細資訊，請參閱 [DynamoDB 佈建的容量](#)。
 - 用於 `dynamodb-table-name-prefix` 指定新增至每個 DynamoDB 表的前置詞 (有助於區分使用相同帳戶的多個代理程式叢集)。僅允許字母數字字元、點、破折號和底線。
2. 停止叢集中的所有代理程式。針對每個代理程式，執行下列命令：

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

3. 確保叢集中的所有代理程式都已停止，然後重新啟動所有代理程式。執行下列命令，啟動每個代理程式：

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

代理程式主機必須具有呼叫 DynamoDB API 的權限。憑證是在 Amazon EC2 執行個體上，憑證是使用 Amazon EC2 中繼資料服務自動擷取。如果您需要指定不同的認證，可以使用其中一種支援的認證擷取技術 (例如 Java 系統屬性或環境變數) 來設定它們。有關詳情，請參閱 [提供和检索 AWS 证书](#)。

配置代理持久在瑪麗亞布/MySQL

Note

`/etc/dcv-session-manager-broker/session-manager-broker.properties` 檔案中包含敏感資料。根據預設，其寫入存取權限制為 `root`，其讀取存取權限僅限於 `root` 使用者和執行 Broker 的使用者。依預設，這是 `dcvsmbroker` 使用者。Broker 會在啟動時檢查檔案是否具有預期的權限。

配置代理程序以開始在瑪麗亞德/MySQL 上保存其數據：

1. 使 `/etc/dcv-session-manager-broker/session-manager-broker.properties` 用您偏好的文字編輯器開啟，然後執行下列命令：
 - 設定 `enable-persistence = true`
 - 設定 `persistence-db = mysql`
 - 設定 `jdbc-connection-url = jdbc:mysql://<db_endpoint>:<db_port>/<db_name>?createDatabaseIfNotExist=true`

在此組態中，<db_endpoint>是資料庫端點、<db_port>資料庫連接埠，而且<db_name>是資料庫名稱。

- 用於jdbc-user指定具有資料庫存取權的使用者名稱。
- 用於jdbc-password指定有權存取資料庫之使用者的密碼。

2. 停止叢集中的所有代理程式。針對每個代理程式，執行下列命令：

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

3. 確定叢集中的所有代理程式都已停止，然後重新啟動所有代理程式。針對每個代理程式，執行下列命令：

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

與 NICE DCV 連線閘道器整合

[NICE DCV 連線閘道](#)是一個可安裝的軟體套件，可讓使用者透過 LAN 或 VPC 的單一存取點存取 NICE DCV 伺服器群。

如果您的基礎結構包含可透過 NICE DCV 連線閘道存取的 NICE DCV 伺服器，您可以設定工作階段管理員以整合 NICE DCV 連線閘道。透過遵循下一節所述的步驟，代理人將充當連線閘道的[工作階段解析](#)程式。換言之，代理人將公開額外的 HTTP 端點公開額外的 HTTP 端點公開一連線閘道會對端點進行 API 呼叫，以擷取將 NICE DCV 連線路由到代理人選取的主機所需的資訊。

將工作階段管理員代理人設定為 NICE DCV 連線閘道的工作階段解析程式

Session Manager 連線 Session Manager 連

1. /etc/dcv-session-manager-broker/session-manager-broker.properties使用您偏好的文字編輯器開啟，然後套用下列命令：
 - 設定 enable-gateway = true
 - 設定gateway-to-broker-connector-https-port為可用的 TCP 連接埠 (預設值為 8447)
 - 設定gateway-to-broker-connector-bind-host為代理人繫結 NICE DCV 連線閘道連線之主機的 IP 位址 (預設值為 0.0.0.0)
2. 然後執行下列命令以停止並重新啟動 Broker：

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

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

3. 擷取 Broker 自我簽署憑證的副本，並將其放置在您的使用者目錄中。

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

在下一步中安裝 NICE DCV 連接閘道時，您將需要它。

NICE DCV DCV 連接網關端

- 請按照 NICE DCV 連接閘道文檔中的[部分](#)進行操作。

由於 NICE DCV 連線閘道會對代理程式進行 HTTP API 呼叫，因此如果代理程式使用自我簽署憑證，您必須將代理程式憑證複製到 NICE DCV 連線閘道主機 (在上一個步驟中擷取)，並在 NICE DCV 連線閘道組態的 [resolver] 區段中設定 ca-file 參數。

選用-啟用 TLS 用戶端驗證

完成上一個步驟後，工作階段管理員和連線閘道就可以透過安全通道進行通訊，連線閘道可以驗證工作階段管理員代理程式的身分識別。如果您還需要工作階段管理員代理程式在建立安全通道之前驗證連線閘道的身分，則需要按照下一節中的步驟啟用 TLS 用戶端驗證功能。

Note

如果工作階段管理員位於負載平衡器後方，則無法使用具有 TLS 連線終止的負載平衡器 (例如應用程式負載平衡器 (ALB) 或閘道負載平衡器 (GLB) 來啟用 TLS 用戶端驗證。僅支援未使用 TLS 終止的負載平衡器，例如網路負載平衡器 (NLB)。如果您使用 ALB 或 GLB，則可以強制執行只有特定安全群組才能連絡負載平衡器，以確保額外的安全性層級；此處有關安全群組的詳細資訊：[VPC 的安全性群組](#)

Session Manager 連線 Session Manager 連

1. 若要為工作階段管理員代理程式和 NICE DCV 連線閘道之間的通訊啟用 TLS 用戶端驗證，請依照下列步驟執行：

2. 執行以產生所需的金鑰和憑證：命令的輸出會告訴您產生認證的資料夾，以及用於建立 TrustStore 檔案的密碼。

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

3. 在您的使用者目錄中放置 NICE DCV 連線閘道私密金鑰和自我簽署憑證的複本。當您在下一個步驟中啟用 NICE DCV 連線閘道中的 TLS 用戶端驗證時，將會需要此驗證。

```
sudo cp /etc/dcv-session-manager-broker/resolver-creds/dcv_gateway_key.pem $HOME
```

```
sudo cp /etc/dcv-session-manager-broker/resolver-creds/dcv_gateway_cert.pem $HOME
```

4. 使用您偏好的文字編輯器開啟 `/etc/dcv-session-manager-broker/session-manager-broker` 連接，然後執行下列命令：

- 設定 `enable-tls-client-auth-gateway` 為 `true`
- 設定 `gateway-to-broker-connector-trust-store-file` 為在上一步中建立的 TrustStore 檔案的路徑
- 設定 `gateway-to-broker-connector-trust-store-pass` 為在上一步中用於建立 TrustStore 檔案的密碼

5. 然後執行下列命令以停止然後重新啟動 Bro 連線，然後停止

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

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

NICE DCV DCV 連接網關端

- 請按照 NICE DCV 連接閘道文檔中的 [部分](#) 進行操作。
 - 在 `[resolver]` 區段中設定 `cert-file` 參數時，使用您在上一個步驟中複製的憑證檔案的完整路徑
 - 在 `[resolver]` 區段中設定 `cert-key-file` 參數時，使用您在上一個步驟中複製的金鑰檔案的完整路徑

NICE DCV DCV 會話管理器 NICE DCV DCV 服務器-DNS 映射

NICE DCV 連線閘道需要 NICE DCV 伺服器的 DNS 名稱，才能連線至 DCV 伺服器執行個體。本節說明如何定義 JSON 檔案，其中包含每個 DCV 伺服器及其相關聯 DNS 名稱之間的對應。

檔案結構

對應包含具有下列欄位的 JSON 物件清單：

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

其中：

ServerIdType:

識別值所參考的識別碼類型；目前可用的值為 IPAddress 和執行個體 ID：agentServerId

Ip:

適用於 Amazon EC2 和內部部署基礎設施；系統管理員可以使用 ifconfig (Linux) 或 ipconfig (Windows) 命令快速擷取。DescribeServers API 回應中也提供此資訊。

Id:

適用於 Amazon EC2 和內部部署基礎設施；每當主機名稱或 IP 位址變更時，工作階段管理員代理程式都會建立新的 UUID。DescribeServers API 回應中提供此資訊。

Host.Aws.Ec2InstanceId:

它僅適用於 Amazon EC2 執行個體，可唯一識別機器；在執行個體重新啟動後不會變更。可以通過聯繫 <http://169.254.169.254/latest/meta-data/instance-id> 在主機上檢索。DescribeServers API 回應中也提供此資訊。

ServerId:

指定類型的 ID，可唯一識別網路中的每個 NICE DCV 伺服器。

DnsNames:

包含與此物件 NICE DCV 伺服器相關聯之 DNS 名稱的物件將包含：

InternalDnsNames:

NICE DCV 連線閘道用來連線至執行個體的 DNS 名稱。

請使用工作階段管理員代理人 CLI 命令 `register-server-dns-mapping` 從檔案載入對應 (命令頁面參考:[register-server-dns-mapping](#))，並列出目前載入 `describe-server-dns-mappings` 到工作階段管理員代理程式中的對映 (命令頁面參考:[describe-server-dns-mappings](#))。

Persistence

我們強烈建議您啟用工作階段管理員代理程式的持續性功能，以防止多個代理程式或整個叢集關閉時造成對應遺失。如需啟用資料持續性的詳細資訊，請參閱[設定代理人持續](#)

與亞馬遜集成 CloudWatch

工作階段管理器支援與 Amazon EC2 執 CloudWatch 行個體上執行的代理程式整合，以及在現場部署主機上執行的代理程式。

Amazon 會即時 CloudWatch 監控您的 Amazon Web Services (AWS) 資源及您 AWS 在上執行的應用程式。您可以使 CloudWatch 用收集和追蹤指標，這些是您可以為您的資源和應用程式測量的變數。如需詳細資訊，請參閱 [Amazon CloudWatch 使用者指南](#)。

您可以設定工作階段管理員代理程式，將下列指標資料傳送至 Amazon CloudWatch：

- `Number of DCV servers` 由代理人管理的 DCV 伺服器數目。
- `Number of ready DCV servers` 處於代理人管理之 READY 狀態的 DCV 伺服器數目。
- `Number of DCV sessions` 代理人管理的 DCV 工作階段數目。
- `Number of DCV console sessions` 由代理人管理的 DCV 主控台工作階段數目。
- `Number of DCV virtual sessions` 代理人管理的 DCV 虛擬工作階段數目。
- `Heap memory used` 代理程式使用的堆積記憶體數量。
- `Off-heap memory used` 代理程式使用的堆積外記憶體數量。

- Describe sessions request time完成 DescribeSessions API 要求所花費的時間量。
- Delete sessions request time完成 DeleteSessions API 要求所花費的時間量。
- Create sessions request time完成 CreateSessions API 要求所花費的時間量。
- Get session connection data request time完成 GetSessionConnectionData API 要求所花費的時間量。
- Update session permissions request time完成 UpdateSessionPermissions API 要求所花費的時間量。

設定代理程式以將指標資料傳送至 Amazon CloudWatch

1. `/etc/dcv-session-manager-broker/session-manager-broker.properties` 使用您偏好的文字編輯器開啟，然後執行下列命令：
 - 設定 `enable-cloud-watch-metrics` 為 `true`
 - 在中 `cloud-watch-region`，指定要收集測量結果資料的「區域」。

Note

如果您的代理程式是在 Amazon EC2 執行個體上執行的代理程式，此參數是選用的。會自動從執行個體中繼資料服務 (IMDS) 擷取區域。如果您在內部部署主機上執行 Broker，則此參數為必要參數。

2. 停止並重新啟動代理。

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

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

代理程式主機也必須具有呼叫 `cloudwatch:PutMetricData` API 的權限。AWS 您可以使用其中一種支援的認證擷取技術來擷取身份證明。如需詳細資訊，請參閱 [提供和擷取AWS認證](#)。

升級 NICE DCV 工作階段管理員

下列主題說明如何升級工作階段管理員。

Note

我們強烈建議您在升級工作階段管理員代理程式之前，先升級所有工作階段管理員代理程式，以避免引入新功能時發生不相容的問題。

主題

- [升級 NICE DCV 工作階段管理員代理程式](#)
- [升級 NICE DCV 工作階段管理員代理程式](#)

升級 NICE DCV 工作階段管理員代理程式

Linux host

Note

下列說明適用於在 64 位元 x86 主機上安裝代理程式。### 64 ## ARM #####
Amazon Linux#RHEL # CentOS### x86_64 ##### Ubuntuarch64###
amd64 ###. arm64

更新 Linux 主機上的代理程式

1. 執行下列命令以停止代理程式。

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

2. 下載安裝套件。

- Amazon Linux 2、雷爾 7.x 及 CentOS 7.x

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

- RHEL 8.x、CentOS 8 流及落基 Linux 8.x 版

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

- Ubuntu 20.04

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

- 瑞士企業版 12

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

- 瑞士企業版

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

3. 安裝套件。

- Amazon Linux 2、雷爾 7.x 及 CentOS 7.x

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

- RHEL 8.x、CentOS 8 流及落基 Linux 8.x 版

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

- Ubuntu 20.04

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

- 瑞士企業版 12

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

- 瑞士企業版

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

4. 執行下列命令以啟動代理程式。

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

Windows host

在 Windows 主機上更新代理程式

1. 停止代理程式服務。在命令提示字元中執行下列命令。

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

2. 下載[代理程式安裝程式](#)。
3. 執行安裝程式。在 Welcome (歡迎) 畫面上，選擇 Next (下一步)。
4. 在 EULA 畫面上，請仔細閱讀授權合約，如果您同意，請選取 [我接受條款]，然後選擇 [下一步]。
5. 若要開始安裝，請選擇 [安裝]。
6. 重新啟動代理程式服務。在命令提示字元中執行下列命令。

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

升級 NICE DCV 工作階段管理員代理程式

升級代理

1. Connect 至您要升級代理程式的主機。
2. 停止代理服務。

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

3. 下載安裝套件。

- Amazon Linux 2、雷爾 7.x 及 CentOS 7.x

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

- RHEL 8.x、CentOS 8 流及落基 Linux 8.x 版

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

- Ubuntu 20.04

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

4. 安裝套件。

- Amazon Linux 2、雷爾 7.x 及 CentOS 7.x

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

- RHEL 8.x、CentOS 8 流及落基 Linux 8.x 版

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

- Ubuntu 20.04

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

5. 啟動 Broker 服務，並確保每次執行個體啟動時都會自動啟動。

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

代理 CLI 參考資料

本節說明如何使用代理程式命令列界面 (CLI) 命令。

如果您使用外部驗證伺服器產生 OAuth 2.0 存取權杖，請使用下列命令：

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

如果您使用工作階段管理員代理人作為 OAuth 2.0 驗證伺服器，請使用下列命令。

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

使用下列命令來管理工作階段管理員代理程式。

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

使用下列命令來管理 DCV 伺服器-DNS 名稱對應檔案。

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

register-auth-server

註冊外部驗證伺服器以與 Broker 搭配使用。

依預設，工作階段管理員會使用代理程式做為驗證伺服器來產生 OAuth 2.0 存取權杖。如果您使用 Broker 作為驗證伺服器，則不需額外進行設定。

但是，如果您選擇使用外部身份驗證伺服器 (例如 Active Directory 或 Amazon Cognito)，則必須使用此命令來註冊外部身份驗證伺服器。

主題

- [語法](#)
- [選項](#)
- [範例](#)

語法

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

選項

--url

要使用的外部驗證伺服器的 URL。您必須附加 `.well-known/jwks.json` 至驗證伺服器 URL。

類型：字串

必要：是

範例

下列範例會使用的 URL 註冊外部驗證伺服器 `https://my-auth-server.com/`。

命令

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

輸出

```
Jwk url registered.
```

list-auth-servers

列出已註冊的外部驗證伺服器。

主題

- [語法](#)
- [輸出](#)
- [範例](#)

語法

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

輸出

Urls

已註冊外部驗證伺服器的 URL。

範例

以下範例列出所有已註冊的外部驗證伺服器。

命令

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

輸出

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

unregister-auth-server

取消註冊外部驗證伺服器。取消註冊外部驗證伺服器後，就無法再使用它來產生 OAuth 2.0 存取權杖。

主題

- [語法](#)
- [選項](#)
- [輸出](#)
- [範例](#)

語法

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

選項

--url

要取消註冊之外部驗證伺服器的 URL。您必須附加 `.well-known/jwks.json` 至驗證伺服器 URL。

類型：字串

必要：是

輸出

Url

未註冊外部驗證伺服器的 URL。

範例

下列範例會使用的 URL 註冊外部驗證伺服器 `https://my-auth-server.com/`。

命令

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

輸出

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

register-api-client

向代理程式註冊工作階段管理員用戶端，並產生用戶端憑證，用戶端可用來擷取 OAuth 2.0 存取權杖，這是發出 API 要求所需的。

Important

確保將憑證儲存在安全的地方。以後無法復原自動備份內容。

僅當代理人用作 OAuth 2.0 驗證伺服器時，才會使用此命令。

主題

- [語法](#)
- [選項](#)
- [輸出](#)
- [範例](#)

語法

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

選項

--name

用於識別工作階段管理員用戶端的唯一名稱。

類型：字串

必要：是

輸出

client-id

工作階段管理員用戶端用來擷取 OAuth 2.0 存取權杖的唯一用戶端識別碼。

client-password

工作階段管理員用戶端用來擷取 OAuth 2.0 存取權杖的密碼。

範例

以下範例會註冊名為的用戶端my-sm-client。

命令

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

輸出

```
client-id: 21cfe9cf-61d7-4c53-b1b6-cf248EXAMPLE  
client-password: NjVmZDRlN2ItNjNmYS00M2QxLWF1ZmMtZmNmMDNkMEXAMPLE
```

describe-api-clients

列出已向代理人註冊的階段作業管理員用戶端。

主題

- [語法](#)
- [輸出](#)
- [範例](#)

語法

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

輸出

name

工作階段管理員用戶端的唯一名稱。

id

工作階段管理員用戶端的唯一 ID。

active

指示階段作業管理員用戶端的狀態。如果用戶端處於作用中狀態，則值為true；否則為false。

範例

下列範例列出已註冊的工作階段管理員用戶端。

命令

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

輸出

```
Api clients
[ {
  "name" : "client-abc",
  "id" : "f855b54b-40d4-4769-b792-b727bEXAMPLE",
  "active" : false
}, {
  "name" : "client-xyz",
  "id" : "21cfe9cf-61d7-4c53-b1b6-cf248EXAMPLE",
  "active" : true
}]
```

unregister-api-client

停用已註冊的工作階段管理員用戶端。停用的工作階段管理員用戶端無法再使用其認證擷取 OAuth 2.0 存取權杖。

主題

- [語法](#)
- [選項](#)
- [範例](#)

語法

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

選項

--client -id

要停用的工作階段管理員用戶端的用戶端 ID。

類型：字串

必要：是

範例

下列範例會停用用戶端 ID 為的工作階段管理員用戶端 f855b54b-40d4-4769-b792-b727bEXAMPLE。

命令

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

輸出

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

renew-auth-server-api-鍵

更新代理人用於簽署 OAuth 2.0 訪問令牌的公鑰和私鑰，這些令牌提供給會話管理器客戶端。如果您續訂金鑰，則必須提供新的私密金鑰給開發人員，因為它是提出 API 要求所需的。

主題

- [語法](#)
- [範例](#)

語法

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

範例

下列範例會更新公開金鑰和私密金鑰。

命令

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

輸出

```
Keys renewed.
```

generate-software-statement

產生軟體陳述式。

代理程式必須向代理程式註冊，才能啟用通訊。代理程式需要軟體陳述式才能向代理商註冊。代理程式具有軟體陳述式之後，它可以使用 [OAuth 2.0 動態用戶端註冊通訊協定](#)，自動向代理程式註冊自己。代理程式向代理程式註冊後，會收到用戶端 ID 和用戶端密碼，用於向 Broker 進行驗證。

代理程式和代理程式會在第一次安裝時接收並使用預設軟體陳述式。您可以繼續使用預設軟體陳述式，或者您可以選擇產生新的陳述式。如果您產生新的軟體陳述式，則必須將軟體陳述式置於代理程式上的新檔案中，然後將檔案路徑新增至 agent.conf 檔案中的 agent.software_statement_path 參數。完成此操作之後，請停止並重新啟動代理程式，以便它可以使使用新的軟體陳述式向 Broker 註冊。

主題

- [語法](#)
- [輸出](#)
- [範例](#)

語法

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

輸出

software-statement

軟件聲明。

範例

下列範例會產生軟體陳述式。

命令

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

輸出

```
software-statement:  
ewogICJpZCIgOiAiYjc1NTVhN2QtNWl0MC00OTJhLWJjOTUtNmUzOWNhYzkyMDcxIiwKICAiYWN0aXZlIiA6IHRydWUsCi
```

describe-software-statements

描述現有的軟體陳述式。

主題

- [語法](#)
- [輸出](#)
- [範例](#)

語法

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


- [語法](#)
- [選項](#)
- [範例](#)

語法

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

選項

--software-statement

要停用的軟體陳述式。

類型：字串

必要：是

範例

下列範例會停用軟體陳述式。

命令

```
sudo -u root dcv-session-manager-broker deactivate-software-statement --software-statement EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj
```

輸出

```
Software statement  
EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj  
deactivated
```

describe-agent-clients

描述已向代理程式註冊的代理程式。

主題

- [語法](#)
- [輸出](#)
- [範例](#)

語法

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

輸出

name

代理程式的名稱。

id

代理程式的唯一 ID。

active

代理程式的狀態。 true如果代理程式處於作用中狀態；否則就是false。

範例

以下範例說明代理程式。

命令

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

輸出

```
Session manager agent clients
[ {
  "name" : "test",
  "id" : "6bc05632-70cb-4410-9e54-eaf9bEXAMPLE",
  "active" : true
}, {
  "name" : "test",
```

```
"id" : "27131cc2-4c71-4157-a4ca-bde38EXAMPLE",
"active" : true
}, {
"name" : "test",
"id" : "308dd275-2b66-443f-95af-33f63EXAMPLE",
"active" : false
}, {
"name" : "test",
"id" : "ce412d1b-d75c-4510-a11b-9d9a3EXAMPLE",
"active" : true
} ]
```

unregister-agent-client

從代理程式取消註冊代理程式。

主題

- [語法](#)
- [選項](#)
- [範例](#)

語法

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

選項

--client-id

要取消註冊的代理程式 ID。

類型：字串

必要：是

範例

下列範例會取消註冊代理程式。

命令

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

輸出

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

register-server-dns-mappings

註冊 DCV 伺服器-來自 JSON 檔案的 DNS 名稱對應。

語法

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

選項

--file-path

包含 DCV 伺服器的檔案路徑-DNS 名稱對應。

類型：字串

必要：是

範例

下列範例會註冊 DCV 伺服器-來自檔案 /tmp/對應.json 的 DNS 名稱對應。

命令

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

輸出

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

describe-server-dns-mappings

描述目前可用的 DCV 伺服器-DNS 名稱對應。

語法

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

輸出

serverIdType

伺服器 ID 的類型。

serverId

伺服器的唯一 ID。

dnsNames

內部和外部 DNS 名稱

internalDnsNames

內部的 DNS 名稱

externalDnsNames

外部 DNS 名稱

範例

以下範例列出已註冊 DCV 伺服器-DNS 名稱對應。

命令

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

輸出

```
[
  {
    "serverIdType" : "Id",
    "serverId" : "192.168.0.1",
    "dnsNames" : {
      "internalDnsName" : "internal1",
      "externalDnsName" : "external1"
    }
  },
  {
    "serverIdType" : "Host.Aws.Ec2InstanceId",
    "serverId" : "i-0648aee30bc78bdff",
    "dnsNames" : {
      "internalDnsName" : "internal2",
      "externalDnsName" : "external2"
    }
  }
]
```

組態檔案參考

本節提供代理程式和代理程式組態檔案的相關資訊。

主題

- [程式組態檔案](#)
- [代理程式組態檔案](#)

程式組態檔案

Broker 組態檔案 (/etc/dcv-session-manager-broker/session-manager-broker.properties) 包含可設定為自訂工作階段管理員功能的參數。您可以使用您偏好的文字編輯器編輯組態檔案。

Note

/etc/dcv-session-manager-broker/session-manager-broker.properties 檔案包含敏感資料。根據預設，其寫入存取權限限制為 root，其讀取存取權限僅限於 root 使用者和執行 Broker 的使用者。依預設，這是dcvsmbroker使用者。Broker 會在啟動時檢查檔案是否具有預期的權限。

下表列出代理程式組態檔案中的參數。

參數名稱	必要	預設值	描述
broker- ja va- home	否		指定代理程式將使用的 Java 主目錄路徑，而非系統預設路徑。如果設置，代理將<broker-java-home>/bin/java 在啟動時使用。 提示：代理人需要 Java 運行時環境 11，如果缺少作

參數名稱	必要	預設值	描述
			<p>為成功安裝後的依賴關係，則會安裝它。如果版本 11 未設置為默認 Java 環境，則可以使用以下命令獲取其主目錄：</p> <pre>\$ sudo alternatives --display java</pre>
session-screenshots-max-width	否	160	指定使用 GetSessionScreenshotsAPI 擷取的工作階段螢幕擷取畫面的最大寬度 (以像素為單位)。
session-screenshots-max-height	否	100	指定使用 GetSessionScreenshotsAPI 擷取的工作階段螢幕快照的最大高度 (以像素為單位)。
session-screenshots-format	否	png	使用 GetSessionScreenshotsAPI 拍攝的會話螢幕截圖的圖像文件格式。

參數名稱	必要	預設值	描述
create-sessions-queue-max-size	否	1000	可排入佇列的未完成 CreateSessionsAPI 要求數目上限。當佇列已滿時，新的未履行要求會遭到拒絕。
create-sessions-queue-max-time-seconds	否	1800	未完成的 CreateSessionsAPI 要求可以保留在佇列中的時間上限 (以秒為單位)。如果請求無法在指定的時間內滿足，則會失敗。
session-manager-working-path	是	/tmp	指定 Broker 寫入操作所需檔案的目錄路徑。此目錄必須只能由代理人存取。
enable-authorization-server	是	true	指定代理程式是否為用來為用戶端 API 產生 OAuth 2.0 存取權杖的驗證伺服器。

參數名稱	必要	預設值	描述
enable-authorization	是	true	啟用或停用用戶端授權。如果您啟用用戶端授權，用戶端 API 在發出 API 要求時必須提供存取權杖。如果停用用戶端授權，用戶端 API 可在沒有存取權杖的情況下發出要求。
enable-agent-authorization	是	true	啟用或停用代理程式授權。如果您啟用代理程式授權，則代理程式在與 Broker 通訊時必須提供存取權杖。
delete-session-duration-hours	否	1	指定經過多少小時，刪除的工作階段會變成不可見且不再由 DescribeSession API 呼叫傳回的時數。
connect-session-token-duration-minutes	否	60	指定 ConnectSession 權杖保持有效的分鐘數。

參數名稱	必要	預設值	描述
client-to-broker-connect-https-port	是	8443	指定代理程式監聽用戶端連線的 HTTPS 連接埠。
client-to-broker-connect-bind-host	否	0.0.0.0	指定代理程式繫結的主機的 IP 地址。
client-to-broker-connect-key-store-file	是		指定 TLS 用戶端連線所使用的金鑰存放區。

參數名稱	必要	預設值	描述
client-to-broker-connect-key-store-pass	是		指定金鑰存放區通行證。
agent-to-broker-connect-https-port	是	8445	指定代理程式接聽代理程式連線的 HTTPS 連接埠。
agent-to-broker-connect-bind-host	否	0.0.0.0	指定代理程式繫結的主機的 IP 地址。

參數名稱	必要	預設值	描述
agent-to-broker-connectokey-store-file	是		指定 TLS 代理程式連線所使用的金鑰存放區。
agent-to-broker-connectokey-store-pass	是		指定金鑰存放區通行證。
broker-to-broker-port	是	47100	指定用於代理程式對代理商連線的連接埠。
broker-to-broker-bind-host	否	0.0.0.0	指定代理程式與代理商連線繫結的主機 IP 位址。

參數名稱	必要	預設值	描述
broker-to-broker-discovery-port	是	47500	指定代理程式用來探索彼此的連接埠。
broker-to-broker-discovery-address	否		以 <i>ip_address: ######## IP ##</i> 和連接 #。如果有多個代理，請用逗號分隔值。如果您指定 broker-to-broker-discovery-multicast-group、broker-to-broker-discovery-multicast-port、broker-to-broker-discovery-AWS-region、或 broker-to-broker-discovery-AWS-alb-target-group-arn，請省略此參數。

參數名稱	必要	預設值	描述
broker-to-broker-discovery-multicast-group	否		指定 Broker-to-roker 探查的多點傳送群組。如果您指定broker-to-broker-discovery-addresses broker-to-broker-discovery-aws-region 、或broker-to-broker-discovery-AWS-alb-target-group-arn ，請省略此參數。
broker-to-broker-discovery-multicast-port	否		指定 Broker-to-roker 探查的多點傳送連接埠。如果您指定broker-to-broker-discovery-addresses broker-to-broker-discovery-AWS-region 、或broker-to-broker-discovery-AWS-alb-target-group-arn ，請省略此參數。

參數名稱	必要	預設值	描述
broker-to-broker-discovery-AWS-region	否		指定用於代理對代理人探索之應用程式負載平衡器的AWS區域。如果您指定broker-to-broker-discovery-multicast-groupbroker-to-broker-discovery-multicast-port、或broker-to-broker-discovery-addresses，請省略此參數。
broker-to-broker-discovery-AWS-alb-target-group-arn	否		用於代理程式對代理人探索之應用程式負載平衡器目標群組使用者的ARN。如果您指定broker-to-broker-discovery-multicast-group broker-to-broker-discovery-multicast-port、或broker-to-broker-discovery-addresses，請省略此參數。

參數名稱	必要	預設值	描述
broker-to-broker-distributed-memory-max-size-mb	否	4096	指定單一代理人用來儲存 NICE DCV 工作階段資料的最大堆積記憶體數量。
broker-to-broker-key-store-file	是		指定 TLS 代理程式連線所使用的金鑰存放區。
broker-to-broker-key-store-pass	是		指定金鑰存放區通行證。
enable-cloud-watch-metrics	否	false	啟用或停用亞馬遜 CloudWatch 指標。如果啟用「CloudWatch 指標」，您可能需要指定的值cloud-watch-region。

參數名稱	必要	預設值	描述
cloud-watch-region	否	僅在設定enable-cloud-watch-metrics為true時才需要true。如果代理程式安裝在 Amazon EC2 執行個體上，則會從 IMDS 擷取該區域。	公佈 CloudWatch 量度的 AWS 區域。
max-api-requests-per-秒	否	1000	指定代理 API 在限制之前每秒可處理的要求數目上限。
enable-throttling-forwarder	否	false	如果設置為true節流，則從 X-福爾-用於頭 (如果存在) 檢索呼叫者 IP。
create-sessions-number-of-retries-on-failure	否	2	指定在 NICE DCV 伺服器主機上建立工作階段要求失敗後，要執行的重試次數上限。設定為 0 以永遠不會在失敗時執行重試。

參數名稱	必要	預設值	描述
autorun-file-arguments-max-size	否	50	指定可以傳送的最大多少個數。
autorun-file-arguments-max-argument-length	否	150	指定每個自動執行檔案引數的最大長度 (以字元為單位)。
enable-persistence	是	false	如果設定為true, Broker 狀態資料會保留在外部資料庫上。
persistence-db	否	僅在設定enable-persistence 為true時才需要true。	指定用於持續性的資料庫。唯一支援的值為:dynamodb和mysql。
dynamodb-region	否	僅當設定enable-persistence 為true且persistence-db 設定為true時才需要dynamodb。	指定建立和存取DynamoDB 表格所在的區域。

參數名稱	必要	預設值	描述
dynamoc table- rcu	否	只有在設定enable- persistence 為true且persisten ce-db 設定為時才需 要dynamodb。	指定每個 DynamoDB 表的 讀取容量單位 (RCU) 如需 有關 RCU 的詳細資訊，請 參閱 佈建容量的定價 。
dynamoc table- wcu	否	只有在設定enable- persistence 為true且persisten ce-db 設定為時才需 要dynamodb。	指定每個 DynamoDB 資料表的寫入容量單位 (WCU)。如需 WCU 的詳細 資訊，請參閱 已佈建容量的 定價 。
dynamoc table- nam e- prefix	否	只有在設定enable- persistence 為true且persisten ce-db 設定為時才需 要dynamodb。	指定新增至每個 DynamoDB 表的前置詞 (有 助於區分使用相同AWS帳 戶的多個代理程式叢集)。 只允許字母數字字元、點、 破折號和底線。
jdbc- conn ection- url	否	只有在設定enable- persistence 為true且persisten ce-db 設定為時才需 要mysql。	指定連接到 Mariadb/M ySQL 數據庫的 URL; 它包 含端點和數據庫名稱。網址 應具有以下格式： <pre>jdbc:mysql://<db_e ndpoint>:<db_port> /<db_name>?createD atabaseIfNotExist= true</pre> <p>其中<db_endpoint> 是 Mariadb/MySQL 數據庫端 點，<db_port> 是數據庫 端口，<db_name> 是數據 庫名稱。</p>

參數名稱	必要	預設值	描述
jdbc-user	否	只有在設定enable-persistence 為true且persistence-db 設定為時才需要mysql。	指定可以存取 Mariadb/MySQL 資料庫的使用者名稱。
jdbc-password	否	只有在設定enable-persistence 為true且persistence-db 設定為時才需要mysql。	指定可以訪問 Mariadb/MySQL 數據庫的用戶的密碼。
seconds-before-deleting-unreachable-dcv-server	否	1800	指定從系統中刪除無法連線之伺服器的秒數。

代理程式組態檔案

代理程式組態檔案 (/etc/dcv-session-manager-agent/agent.conf適用C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf於 Linux 和 Windows) 包含可設定為自訂工作階段管理員功能的參數。您可以使用您偏好的文字編輯器編輯組態檔案。

下表列出代理程式組態檔案中的參數。

參數名稱	必要	預設值	描述
agent.tls_ker_hostname	是		指定代理程式的 DNS 名稱。
agent.tls_ker_port	是	8445	指定要與代理人通訊的連接埠。
agent.config_file	否		僅當設定tls_strict 為 true 時才需要。指定驗證 TLS 憑證所需的憑證 (.pem) 檔案路徑。將自簽的證書從代理程式複製到代理程式。
agent.config_init_file	否	<ul style="list-style-type: none"> /var/lib/dcv-session-manager-agent/init (Linux) 	指定主機伺服器上用來儲存自訂指令碼的資料夾路徑，允許在建立時初始化 NICE DCV 伺服器工作階段。您必須指定絕對路徑。該文件夾必須是可訪問的，並且文件必須由使用 CreateSessionsAPI 的 InitFile 請求參數的用戶可執行。
agent.config_tls_strict	否	true	指出是否應使用嚴格的 TLS 驗證。
agent.config_software_statement_path	否		只有在未使用預設軟體陳述式時才需要。指定軟體陳述式檔案的路徑。如需詳細資訊，請參閱 generate-software-statement 。
agent.config_software_statement_path	否	<ul style="list-style-type: none"> /etc/dcv-session-manager-agent (Linux) 	指定標籤所在的資料夾的路徑。如需詳細資訊，請參閱 generate-software-statement 。

參數名稱	必要	預設值	描述
		<ul style="list-style-type: none"> C:\Program Files\NICE\DCVSessionManagerAgent\conf\tags (Windows) 	閱使用標籤以 NICE DCV 伺服器為目標。
agent.autorun_file	否	<ul style="list-style-type: none"> /var/lib/dcv-session-manager-autorun (Linux) C:\ProgramData\NICE\DcvSessionManagerAgent\autorun (Windows) 	指定主機伺服器上用來儲存允許在工作階段啟動時自動執行的指令碼和應用程式的資料夾路徑。您必須指定絕對路徑。該文件夾必須是可訪問的，並且文件必須由使用 CreateSessionsAPI 的AutorunFile請求參數的用戶可執行。
agent.max_virtual_sessions	否	-1 (無限制)	可以使用 NICE DCV 工作階段管理員在 NICE DCV 伺服器上建立的虛擬工作階段數目上限。
agent.max_concurrent_sessions_per_user	否	1	單一使用者使用 NICE DCV 工作階段管理員可在 NICE DCV 伺服器上建立的虛擬工作階段數目上限。

參數名稱	必要	預設值	描述
agent.t ker_upc e_inte l	否	30	指定將更新的資料傳送至代理程式之前要等待的秒數。傳送的資料包括 NICE DCV 伺服器 and 主機狀態，以及更新的工作階段資訊。較低的值可讓工作階段管理員更清楚執行代理程式的系統上發生的變更，但會增加系統負載和網路流量。較高的值會減少系統和網路負載，但工作階段管理員對系統變更的回應速度較低，因此值高於 120 不建議使用的值。
log.level	否	info	指定記錄檔的詳細程度層級。可用的詳細資訊等級如下： <ul style="list-style-type: none"> • error 提供最少的細節。僅包含錯誤。 • warning 包括錯誤和警告。 • info 預設詳細程度層級。包含錯誤、警告和資訊訊息。 • debug 提供最詳細的資訊。提供有助於偵錯問題的詳細資訊。

參數名稱	必要	預設值	描述
log.directory	否	<ul style="list-style-type: none"> • /var/log/dcv-session-manager-agent/(Linux) • C:\ProgramData\NICE\DCVSessionManagerAgent\log (Windows) 	指定欲建立日誌檔案的目錄。
log.rotation	否	daily	<p>指定日誌檔案的輪換。有效值為：</p> <ul style="list-style-type: none"> • hourly日誌檔案每小時輪換。 • daily日誌檔案每天輪換。
log.max-file-size	否	10485760	當記錄檔大小達到以位元組為單位的指定大小時，它將被旋轉。將會建立新的記錄檔，並將進一步的記錄事件放置在新檔案中。
log.rotate	否	9	旋轉中保留的記錄檔數目上限。每次輪替並達到此數字時，將刪除最舊的日誌文件。

NICE DCV 工作階段管理員的發行說明和文件歷史記錄

此頁面提供 NICE DCV 工作階段管理員的發行說明和文件歷史記錄。

主題

- [NICE DCV 工作階段管理員發行說明](#)
- [文件歷史紀錄](#)

NICE DCV 工作階段管理員發行說明

本節提供 NICE DCV 工作階段管理員的主要更新、功能版本和錯誤修正的概觀。所有更新都是按發布日期組織的。我們會經常更新文件，以解決您傳送給我們的意見反應。

主題

- [2023.1— 二二三年十一月九日](#)
- [2023.0-15065— 二二二三年五月四日](#)
- [2023.0-14852— 二二三年三月二十八日](#)
- [2022.2-13907— 二〇〇二年十一月十一日](#)
- [2022.1-13067— 二零二二年六月二十九日](#)
- [2022.0-11952— 二零二二年二月二十三日](#)
- [2021.3-11591— 二零二一年十二月二十日](#)
- [2021.2-11445— 二零二一年十一月十八日](#)
- [2021.2-11190— 二零二一年十月十一日](#)
- [2021.2-11042— 二零二一年九月一日](#)
- [2021.1-10557— 二零二一年五月三十一日](#)
- [2021.0-10242— 二零二一年四月十二日](#)
- [2020.2-9662— 二零二零年十二月四日](#)
- [2020.2-9508— 二零二零年十一月十一日](#)

2023.1— 二二三年十一月九日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人： 代理人： CLI：140 	<ul style="list-style-type: none"> 錯誤修正與效能改進

2023.0-15065— 二二二三年五月四日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人： 代理人： CLI: 132 	<ul style="list-style-type: none"> 在 ARM 平台上增加了對紅帽企業版 Linux 9，洛奇 Linux 9 和 CentOS 流 9 的支持。

2023.0-14852— 二二三年三月二十八日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人： 代理人： CLI: 132 	<ul style="list-style-type: none"> 增加了對紅帽企業版 Linux 9、洛基 Linux 9 和 CentOS 流 9 的支援。

2022.2-13907— 二〇〇二年十一月十一日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人 代理人： CLI: 123 	<ul style="list-style-type: none"> 在 DescribeSessions 響應中添加了一個 Substate 字段。 修正可能導致 CLI 無法根據使用中的 URL 連線至代理程式的問題。

2022.1-13067— 二零二二年六月二十九日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人 代理人: CLI : 114 	<ul style="list-style-type: none"> 已新增在 AWS Graviton 執行個體上執行代理程式的支援。 增加了對 Ubuntu 22.04 的代理程式和代理程式支援。

2022.0-11952— 二零二二年二月二十三日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人 : 代理人: CLI: 112 	<ul style="list-style-type: none"> 已將記錄輪換功能新增至代理程式。 已新增組態參數以在代理程式中設定 Java 本位目錄。 改善 Broker 中從快取到磁碟的資料清除功能。 在 CLI 中修正了 URL 驗證。

2021.3-11591— 二零二一年十二月二十日

建置編號	新的 功能
<ul style="list-style-type: none"> 經紀人 : 307 代理人: CLI : 92 	<ul style="list-style-type: none"> 新增了與 NICE DCV 連線閘道整合的支援。 增加了對 Ubuntu 18.04 和 Ubuntu 20.04 的代理支持。

2021.2-11445— 二零二一年十一月十八日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人 : 代理人: CLI 數 : 54 	<ul style="list-style-type: none"> 修正驗證包含 Windows 網域的登入名稱的問題。

2021.2-11190— 二零二一年十月十一日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人：254 代理人： CLI 數：54 	<ul style="list-style-type: none"> 修正命令列介面中無法啟動 Windows 工作階段的問題。

2021.2-11042— 二零二一年九月一日

建置編號	新的 功能	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人：254 代理人： CLI：37 	<ul style="list-style-type: none"> NICE DCV 工作階段管理員現在提供命令列介面 (CLI) 支援。您可以在 CLI 中建立和管理 NICE DCV 工作階段，而不是呼叫 API。 NICE DCV 會話管理器引入了代理人數據持久性。為了提高可用性，代理程式可以在外部資料存放區上保留伺服器狀態資訊，並在啟動時還原資料。 	<ul style="list-style-type: none"> 註冊外部授權伺服器時，您現在可以指定授權伺服器用來簽署 JSON 格式的 Web Token 的演算法。透過此變更，您可以使用 Azure AD 做為外部授權伺服器。

2021.1-10557— 二零二一年五月三十一日

建置編號	新的 功能	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人：214 代理人：365 	<ul style="list-style-type: none"> NICE DCV 會話管理器添加了對傳遞到 Linux 上自動運行文件的輸入參數的支持。 伺服器屬性現在可以根據需求傳遞給 CreateSessionsAPI。 	<ul style="list-style-type: none"> 我們修復了 Windows 上自動運行文件的問題。

2021.0-10242— 二零二一年四月十二日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人：183 代理人： 	<ul style="list-style-type: none"> NICE DCV DCV 會話管理器引入了以下新 API： <ul style="list-style-type: none"> OpenServers CloseServers DescribeServers GetSessionScreenshots 它還引入了以下新的配置參數： <ul style="list-style-type: none"> 代理人參數：<code>session-screenshot-max-width</code> <code>session-screenshot-max-height</code> <code>session-screenshot-format</code>、<code>create-sessions-queue-max-size</code>、和 <code>create-sessions-queue-max-time-seconds</code>。 代理程式參數：<code>agent.autorun_folder</code> <code>max_virtual_sessions</code>、和 <code>max_concurrent_sessions_per_user</code>。 <p>代理程式參數：<code>agent.autorun_folder</code> <code>max_virtual_sessions</code>、和 <code>max_concurrent_sessions_per_user</code>。</p> <p>代理程式參數：<code>agent.autorun_folder</code> <code>max_virtual_sessions</code>、和 <code>max_concurrent_sessions_per_user</code>。</p>

2020.2-9662— 二零二零年十二月四日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人：114 代理人:11 	<ul style="list-style-type: none"> 我們修正了自動產生的 TLS 憑證的問題，導致 Broker 無法啟動。

2020.2-9508— 二零二零年十一月十一日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> 經紀人：78 代理人： 	<ul style="list-style-type: none"> NICE DCV 工作階段管理員的初始版本。

文件歷史紀錄

下表說明此版本 NICE DCV 工作階段管理員的文件。

變更	描述	日期
NICE DCV 民政署	NICE DCV DCV 會話管理器已經更新了 NICE DCV DCV 2023.1。如需詳細資訊，請參閱 2023.1— 二二三年十一月九日 。	2023年11月9日
NICE DCV 市民商業	NICE DCV DCV 會話管理器已更新 NICE DCV DCV 2023.0。如需詳細資訊，請參閱 2023.0-14852— 二二三年三月二十八日 。	2023年3月28日
NICE DCV 民政署版本 2022.2	NICE DCV DCV 會話管理器已經更新了 NICE DCV DCV 2022.2。如需詳細資訊，請參閱 2022.2-13907— 二〇〇二年十一月十一日 。	2022年11月11日
NICE DCV 市民商業中心 2022.1 版	NICE DCV DCV 會話管理器已更新為 NICE DCV DCV 2022.1。如需詳細資訊，請參閱 2022.1-13067— 二零二二年六月二十九日 。	2022年6月29日
NICE DCV 市民商業中心	NICE DCV DCV 會話管理器已更新 NICE DCV DCV 2022.0。如需詳細資訊	2022年2月23日

變更	描述	日期
	訊，請參閱 2022.0-11952— 二零二二年二月二十三日 。	
NICE DCV 民政署版本 2021.3	NICE DCV DCV 會話管理器已更新為 NICE DCV DCV 2021.3。如需詳細資訊，請參閱 2021.3-11591— 二零二一年十二月二十日 。	2021 年 12 月 20 日
NICE DCV 民政署版本 2021.2	NICE DCV DCV 會話管理器已經更新了 NICE DCV DCV 2021.2。如需詳細資訊，請參閱 2021.2-11042— 二零二一年九月一日 。	2021年9月01 日
NICE DCV 市民商業中心	NICE DCV DCV 會話管理器已更新為 NICE DCV DCV 2021.1。如需詳細資訊，請參閱 2021.1-10557— 二零二一年五月三十一日 。	2021年5月31日
NICE DCV 民政署版本	NICE DCV DCV 會話管理器已更新 NICE DCV DCV 2021.0。如需詳細資訊，請參閱 2021.0-10242— 二零二一年四月十二日 。	2021 年 4 月 12 日
NICE DCV 會話管理器的初始版本	此內容的第一次出版。	2020 年 11 月 11 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。