



管理員指南

# Amazon DCV Session Manager



# Amazon DCV Session Manager: 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，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

# Table of Contents

什麼是 Session Manager ? .....	1
Session Manager 的運作方式 .....	1
功能 .....	3
限制 .....	3
定價 .....	3
要求 .....	3
網路和連線需求 .....	5
設定 Session Manager .....	6
步驟 1：準備 Amazon DCV 伺服器 .....	6
步驟 2：設定代理程式 .....	7
步驟 3：設定客服人員 .....	9
步驟 4：設定 Amazon DCV 伺服器 .....	14
步驟 5：驗證安裝 .....	15
驗證客服人員 .....	16
驗證代理程式 .....	17
設定 Session Manager .....	18
Scaling Session Manager .....	18
步驟 1：建立執行個體設定檔 .....	19
步驟 2：準備負載平衡器的SSL憑證 .....	20
步驟 3：建立代理程式應用程式負載平衡器 .....	20
步驟 4：啟動代理程式 .....	21
步驟 5：建立客服人員應用程式負載平衡器 .....	22
步驟 6：啟動客服人員 .....	24
在 Amazon 伺服器上使用標籤 DCV .....	25
設定外部授權伺服器 .....	26
設定代理程式持續性 .....	31
將代理程式設定為在 DynamoDB 上保留 .....	31
將代理程式設定為保留在 MariaDB /My 上SQL .....	32
與 Amazon DCV Connection Gateway 整合 .....	33
將 Session Manager Broker 設定為 Amazon DCV Connection Gateway 的 Session Resolver .....	34
選用 - 啟用TLS用戶端身分驗證 .....	35
Amazon DCV 伺服器 - DNS映射參考 .....	36
與亞馬遜集成 CloudWatch .....	38

升級 Session Manager .....	40
升級 Amazon DCV Session Manager 代理程式 .....	40
升級 Amazon DCV Session Manager 代理程式 .....	43
代理程式CLI參考 .....	46
register-auth-server .....	47
語法 .....	47
選項 .....	47
範例 .....	47
list-auth-servers .....	48
語法 .....	47
輸出 .....	48
範例 .....	47
unregister-auth-server .....	49
語法 .....	47
選項 .....	47
輸出 .....	48
範例 .....	47
register-api-client .....	50
語法 .....	47
選項 .....	47
輸出 .....	48
範例 .....	47
describe-api-clients .....	51
語法 .....	47
輸出 .....	48
範例 .....	47
unregister-api-client .....	53
語法 .....	47
選項 .....	47
範例 .....	47
renew-auth-server-api-金鑰 .....	54
語法 .....	47
範例 .....	47
generate-software-statement .....	54
語法 .....	47
輸出 .....	48

範例 .....	47
describe-software-statements .....	55
語法 .....	47
輸出 .....	48
範例 .....	47
deactivate-software-statement .....	57
語法 .....	47
選項 .....	47
範例 .....	47
describe-agent-clients .....	58
語法 .....	47
輸出 .....	48
範例 .....	47
unregister-agent-client .....	59
語法 .....	47
選項 .....	47
範例 .....	47
register-server-dns-mappings .....	60
語法 .....	47
選項 .....	47
範例 .....	47
describe-server-dns-mappings .....	61
語法 .....	47
輸出 .....	48
範例 .....	47
組態檔案參考 .....	63
代理程式組態檔案 .....	63
代理程式組態檔案 .....	77
版本備註和文件歷史記錄 .....	82
版本備註 .....	82
2024.0-457 — 2024 年 10 月 1 日 .....	83
2023.1-17652 — 2024 年 8 月 1 日 .....	83
2023.1-16388 — 2024 年 6 月 26 日 .....	83
2023.1 — 2023 年 11 月 9 日 .....	83
2023.0-15065 — 2023 年 5 月 4 日 .....	84
2023.0-14852 — 2023 年 3 月 28 日 .....	84

---

2022.2-13907 — 2022 年 11 月 11 日 .....	84
2022.1-13067 — 2022 年 6 月 29 日 .....	84
2022.0-11952 — 2022 年 2 月 23 日 .....	85
2021.3-11591 — 2021 年 12 月 20 日 .....	85
2021 年 2 月 11445 日 — 2021 年 11 月 18 日 .....	85
2021.2-11190 — 2021 年 10 月 11 日 .....	85
2021.2-11042 — 2021 年 9 月 1 日 .....	86
2021.1-10557 — 2021 年 5 月 31 日 .....	86
2021 年 0 月 10242 日 — 2021 年 4 月 12 日 .....	86
2020.2-9662 — 2020 年 12 月 4 日 .....	87
.....	87
文件歷史紀錄 .....	88
.....	XC

# 什麼是 Amazon DCV Session Manager ？

## Note

Amazon DCV 先前稱為 NICE DCV。

Amazon DCV Session Manager 是一組可安裝的軟體套件（代理程式和代理程式）和應用程式程式設計介面（API），可讓開發人員和獨立軟體供應商（ISVs）輕鬆建立前端應用程式，以程式設計方式在 Amazon DCV 伺服器機群中建立和管理 Amazon DCV 工作階段的生命週期。

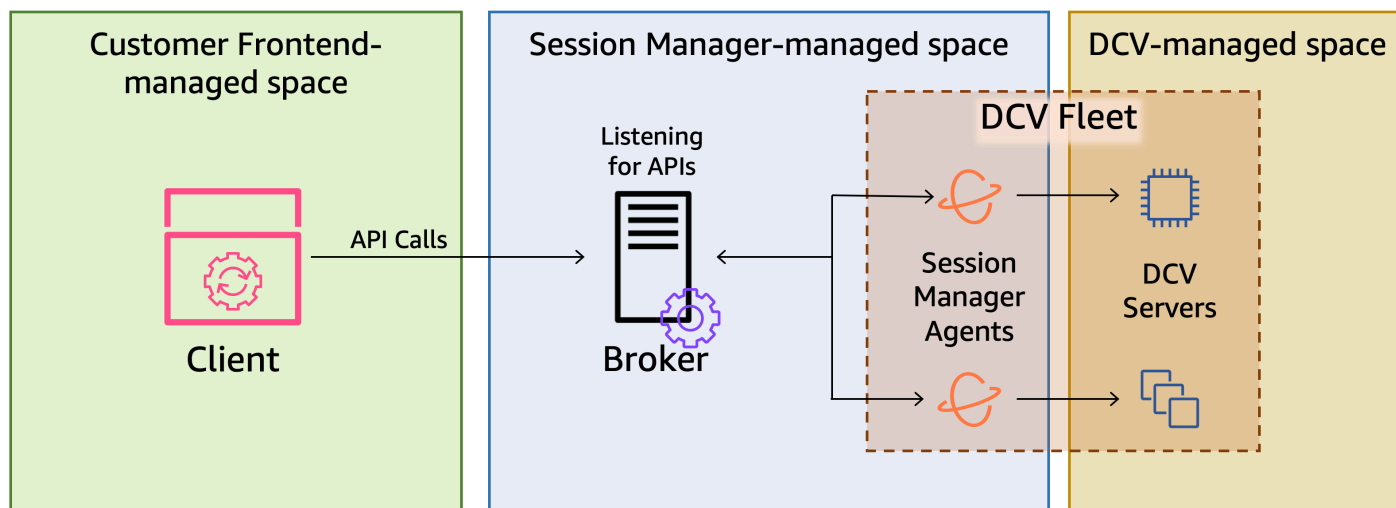
本指南說明如何安裝和設定 Session Manager Agent and Broker。如需使用 Session Manager 的詳細資訊 APIs，請參閱 Amazon DCV Session Manager 開發人員指南。

## 主題

- [Session Manager 的運作方式](#)
- [功能](#)
- [限制](#)
- [定價](#)
- [Amazon DCV Session Manager 需求](#)

## Session Manager 的運作方式

下圖顯示 Session Manager 的高階元件。



## 中介裝置

Broker 是託管並公開 Session Manager 的 Web 伺服器APIs。它會接收並處理來自用戶端的 Amazon DCV工作階段管理API請求，然後將指示傳遞給相關的客服人員。代理程式必須安裝在與您的 Amazon DCV 伺服器分開的主機上，但用戶端必須可存取，且必須能夠存取代理程式。

## 代理程式

代理程式安裝在機群中的每個 Amazon DCV 伺服器上。客服人員會收到代理程式的指示，並在各自的 Amazon DCV伺服器上執行這些指示。客服人員也會監控 Amazon DCV 伺服器的狀態，並將定期狀態更新傳回給代理程式。

## APIs

Session Manager 會公開一組REST應用程式程式設計介面（APIs），可用於管理 Amazon DCV 伺服器機群上的 Amazon DCV工作階段。APIs 託管於 並由代理程式 公開。開發人員可以建立呼叫的自訂工作階段管理用戶端APIs。

## 用戶端

用戶端是您開發的前端應用程式或入口網站，用於呼叫代理程式 APIs公開的 Session Manager。最終使用者使用 用戶端來管理機群中 Amazon DCV伺服器上託管的工作階段。

## 存取權杖

若要提出API請求，您必須提供存取權杖。註冊的用戶端 可以從代理程式或外部授權伺服器請求權杖APIs。若要請求和存取字符，用戶端API必須提供有效的憑證。

## 用戶端 API

用戶端是使用 Swagger Codegen 從 Session Manager API定義YAML檔案API產生。用戶端API用於提出API請求。

## Amazon DCV工作階段

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

- CREATING- 代理程式正在建立工作階段。
- 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。代理程式和客服人員可能無法通訊。

## 功能

DCV Session Manager 提供下列功能：

- 提供 Amazon DCV工作階段資訊 — 取得在多個 Amazon DCV 伺服器上執行之工作階段的相關資訊。
- 管理多個 Amazon DCV工作階段的生命週期：透過單一API請求為多個 Amazon DCV 伺服器中的多個使用者建立或刪除多個工作階段。
- 支援標籤 — 在建立工作階段時，使用自訂標籤鎖定一組 Amazon DCV 伺服器。
- 管理多個 Amazon DCV工作階段的許可 — 透過一個API請求修改多個工作階段的使用者許可。
- 提供連線資訊 — 擷取 Amazon DCV工作階段的用戶端連線資訊。
- 支援雲端和內部部署 — 在 、 AWS內部部署上使用 Session Manager ，或搭配替代的雲端型伺服器。

## 限制

Session Manager 不提供資源佈建功能。如果您在 Amazon EC2執行個體DCV上執行 Amazon ，您可能需要使用 AWS 其他服務，例如 Amazon EC2 Auto Scaling 來管理基礎設施的擴展。

## 定價

執行EC2執行個體 AWS 的客戶可免費使用 Session Manager。

內部部署客戶需要 Amazon DCV Plus 或 Amazon DCV Professional Plus 授權。如需有關如何購買 Amazon DCV Plus 或 Amazon DCV Professional Plus 授權的資訊，請參閱[如何在 Amazon 網站上購買](#)，以及尋找您所在區域的 Amazon DCV經銷商或經銷商。DCV若要允許所有內部部署客戶嘗試 Amazon DCV Session Manager，許可要求只會從 Amazon 2021.0 DCV版開始強制執行。

如需詳細資訊，請參閱 [Amazon 管理員指南 中的授權 Amazon DCV Server](#)。 DCV

## Amazon DCV Session Manager 需求

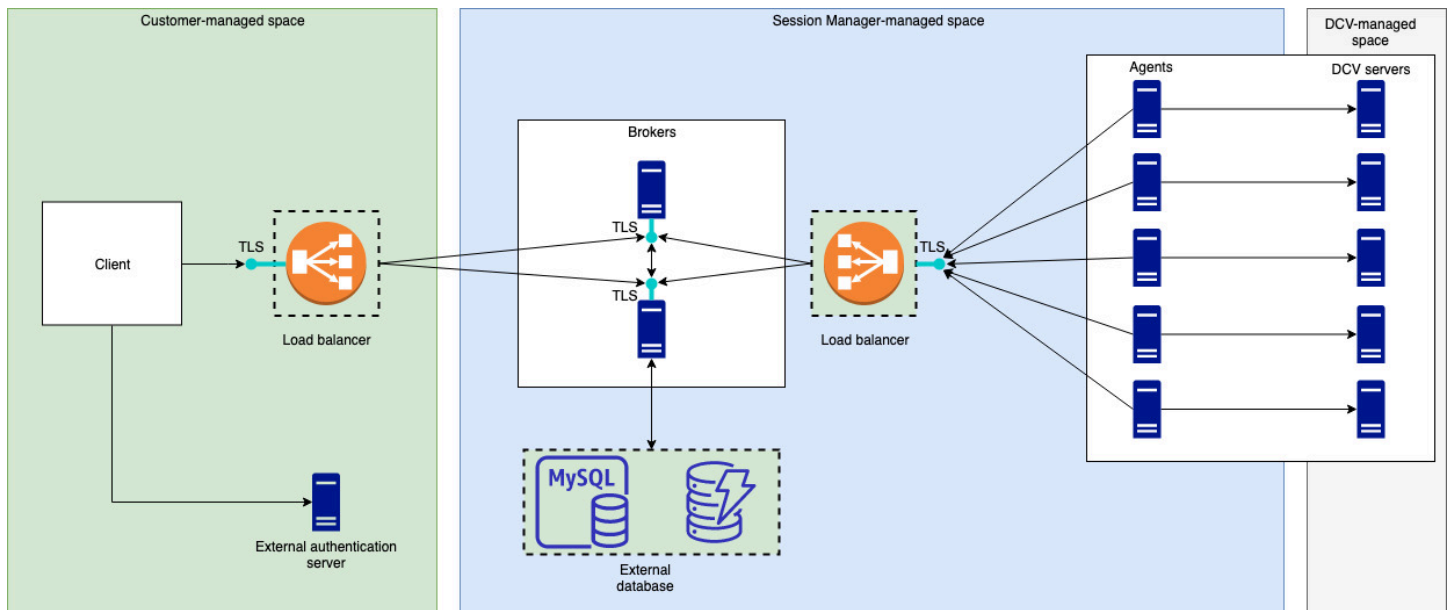
Amazon DCV Session Manager 代理程式和代理程式有下列要求。

	代理程式	代理程式
作業系統	<ul style="list-style-type: none"> <li>• Amazon Linux 2</li> <li>• CentOS Stream 9</li> <li>• RHEL 7.6 或更新版本</li> <li>• RHEL 8.x</li> <li>• RHEL 9.x</li> <li>• Rocky Linux 8.5 或更新版本</li> <li>• Rocky Linux 9.x</li> <li>• Ubuntu 20.04</li> <li>• Ubuntu 22.04</li> <li>• Ubuntu 24.04</li> </ul>	<ul style="list-style-type: none"> <li>• Windows <ul style="list-style-type: none"> <li>• Windows Server 2022</li> <li>• Windows Server 2019</li> <li>• Windows Server 2016</li> </ul> </li> <li>• Linux 伺服器 <ul style="list-style-type: none"> <li>• Amazon Linux 2</li> <li>• CentOS Stream 9</li> <li>• RHEL 8.x</li> <li>• RHEL 9.x</li> <li>• Rocky Linux 8.5 或更新版本</li> <li>• Rocky Linux 9.x</li> <li>• Ubuntu 20.04</li> <li>• Ubuntu 22.04</li> <li>• Ubuntu 24.04</li> <li>• SUSE 具有 SP4或更新版本的 Linux Enterprise 12</li> <li>• SUSE Linux Enterprise 15</li> </ul> </li> </ul>
架構	<ul style="list-style-type: none"> <li>• 64 位元 x86</li> <li>• 64 位元 ARM</li> </ul>	<ul style="list-style-type: none"> <li>• 64 位元 x86</li> <li>• 64 位元 ARM ( 僅限 Amazon Linux 2、CentOS 9.x、8.x/9.x RHEL 和 Rocky 8.x/9.x )</li> <li>• 64 位元 ARM ( Ubuntu 22.04 和 24.04 )</li> </ul>
記憶體	8 GB	4 GB
Amazon DCV版本	Amazon DCV 2020.2 及更新版本	Amazon DCV 2020.2 及更新版本

	代理程式	代理程式
其他要求	Java 11	-

## 網路和連線需求

下圖提供 Session Manager 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 Amazon DCV 伺服器上的代理程式具有網路連線。如果您選擇有多個代理程式來改善可用性，則必須在個別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與代理程式，以及代理程式與代理程式之間的流量。代理程式也應該能夠彼此通訊，以交換 Amazon DCV 伺服器和工作階段的相關資訊。代理程式可以在外部資料庫中儲存其金鑰和狀態資料，並在重新啟動或終止後提供此資訊。這有助於降低遺失重要代理程式資訊的風險，方法是將其保留在外部資料庫中。您可以稍後再擷取。如果您選擇擁有它，則必須設定外部資料庫並設定代理程式。支援 DynamoDB、MariaDB 和 MySQL。您可以在[代理程式組態檔案中找到組態參數](#)。

客服人員必須能夠啟動與代理程式的安全、持久、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 APIs。用戶端也應該能夠存取您的身分驗證伺服器。

# 設定 Amazon DCV Session Manager

下一節說明如何使用單一代理程式和多個代理程式安裝 Session Manager。您可以使用多個代理程式來改善可擴展性和效能。如需詳細資訊，請參閱[Scaling Session Manager](#)。

若要設定 Amazon DCV Session Manager，請執行下列動作：

## 步驟

- [步驟 1：準備 Amazon DCV 伺服器](#)
- [步驟 2：設定 Amazon DCV Session Manager 代理程式](#)
- [步驟 3：設定 Amazon DCV Session Manager 代理程式](#)
- [步驟 4：將 Amazon DCV 伺服器設定為使用代理程式作為身分驗證伺服器](#)
- [步驟 5：驗證安裝](#)

## 步驟 1：準備 Amazon DCV 伺服器

您必須擁有要搭配 Session Manager 使用的 Amazon DCV 伺服器機群。如需安裝 Amazon DCV 伺服器的詳細資訊，請參閱[Amazon 管理員指南](#) 中的 [安裝 Amazon DCV 伺服器](#)。DCV

在 Linux Amazon DCV 伺服器上，Session Manager 會使用名為 `dcvsmagent` 的本機服務使用者。此使用者會在安裝 Session Manager 代理程式時自動建立。您必須授予 Amazon 的此服務使用者管理員權限，DCV 才能代表其他使用者執行動作。若要授予 Session Manager 服務使用者管理員權限，請執行下列動作：

新增 Linux Amazon 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` 參數新增至 `[security]` 區段，並指定 Session Manager 使用者。例如：

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

3. 儲存並關閉檔案。
4. 停止並重新啟動 Amazon DCV 伺服器。

Session Manager 只能代表已存在於 Amazon DCV 伺服器上的使用者建立 Amazon DCV 工作階段。如果請求為不存在的使用者建立工作階段，則請求會失敗。因此，您必須確保每個預期的最終使用者在 Amazon DCV 伺服器上都有有效的系統使用者。

### Tip

如果您打算將多個代理程式主機或 Amazon DCV 伺服器與代理程式搭配使用，建議您執行 DCV 下列步驟，以建立主機的 Amazon Machine Images (AMI)，然後使用 AMIs 啟動其餘代理程式和 Amazon DCV 伺服器。或者，您可以使用 AWS Systems Manager 遠端在多個執行個體上執行命令。

## 步驟 2：設定 Amazon DCV Session Manager 代理程式

代理程式必須安裝在 Linux 主機上。如需支援 Linux 發行版本的詳細資訊，請參閱 [Amazon DCV Session Manager 需求](#)。在與代理程式和 Amazon 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有網路上，但必須能夠連線至代理程式並與代理程式通訊。

### 安裝和啟動代理程式

1. 連線到您要在其中安裝代理程式的主機。
2. 套件使用安全簽章進行數位 GPG 簽章。若要允許套件管理員驗證套件簽章，您必須匯入 Amazon DCV GPG 金鑰。執行下列命令以匯入 Amazon DCV GPG 金鑰。

- Amazon Linux 2、RHEL、CentOS 和 Rocky 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、RHEL7.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1.el7.noarch.rpm
```

- RHEL 8.x 和 Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1.el8.noarch.rpm
```

- CentOS 9.x、9.x RHEL 和 Rocky Linux 9.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1.el9.noarch.rpm
```

- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker_2024.0.457-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker_2024.0.457-1_all.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker_2024.0.457-1_all.ubuntu2404.deb
```

#### 4. 安裝套件。

- Amazon Linux 2、RHEL7.x

```
$ sudo yum install -y ./nice-dcv-session-manager-broker-2024.0.457-1.el7.noarch.rpm
```

- RHEL 8.x 和 Rocky Linux 8.x

```
$ sudo yum install -y ./nice-dcv-session-manager-broker-2024.0.457-1.el8.noarch.rpm
```

- Ubuntu 20.04

```
$ sudo apt install -y ./nice-dcv-session-manager-broker_2024.0.457-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ sudo apt install -y ./nice-dcv-session-manager-broker_2024.0.457-1_all.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ sudo apt install -y ./nice-dcv-session-manager-broker_2024.0.457-1_all.ubuntu2404.deb
```

## 5. 檢查預設 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用來鎖定正確 Java 版本的 Java 主目錄。這會在 `broker-java-home` 代理程式組態檔案中設定 參數。如需詳細資訊，請參閱 [代理程式組態檔案](#)。

## 6. 啟動代理程式服務，並確保它在每次執行個體啟動時自動啟動。

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

## 7. 將代理程式自我簽署憑證的副本放入您的使用者目錄中。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 Amazon DCV Session Manager 代理程式

代理程式必須安裝在機群中的所有 Amazon DCV 伺服器主機上。代理程式可以同時安裝在 Windows 和 Linux 伺服器上。如需支援的作業系統的詳細資訊，請參閱 [Amazon DCV Session Manager 需求](#)。

### 必要條件

在安裝代理程式之前，必須在主機上安裝 Amazon DCV 伺服器。

## Linux host

### Note

Session Manager 代理程式適用於[需求](#) 中列出的 Linux 發行版本和架構：

下列指示適用於在 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，請取代 `x86_64` 使用 `aarch64`。對於 Ubuntu，請取代 `amd64` 使用 `arm64`。

### 在 Linux 主機上安裝代理程式

1. 套件使用安全簽章進行數位 GPG 簽章。若要允許套件管理員驗證套件簽章，您必須匯入 Amazon DCV GPG 金鑰。執行下列命令以匯入 Amazon DCV GPG 金鑰。

- Amazon Linux 2、RHEL、CentOS 和 SUSE Linux Enterprise

```
$ 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 和 RHEL 7.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.el7.x86_64.rpm
```

- RHEL 8.x 和 Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.el8.x86_64.rpm
```

- CentOS 9.x、9.x RHEL 和 Rocky Linux 9.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.el9.x86_64.rpm
```



- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2404.deb
```

- SUSE Linux Enterprise 12

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.sles12.x86_64.rpm
```

- SUSE Linux Enterprise 15

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.sles15.x86_64.rpm
```

### 3. 安裝套件。

- Amazon Linux 2 和 RHEL 7.x

```
$ sudo yum install -y ./nice-dcv-session-manager-agent-2024.0.781-1.el7.x86_64.rpm
```

- RHEL 8.x 和 Rocky Linux 8.x

```
$ sudo yum install -y ./nice-dcv-session-manager-agent-2024.0.781-1.el8.x86_64.rpm
```

- CentOS 9.x、9.x RHEL 和 Rocky Linux 9.x

```
$ sudo yum install -y ./nice-dcv-session-manager-agent-2024.0.781-1.el9.x86_64.rpm
```

- Ubuntu 20.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2404.deb
```


- SUSE Linux Enterprise 12

```
$ sudo zypper install ./nice-dcv-session-manager-agent-2024.0.781-1.sles12.x86_64.rpm
```

- SUSE Linux Enterprise 15

```
$ sudo zypper install ./nice-dcv-session-manager-agent-2024.0.781-1.sles15.x86_64.rpm
```

4. 將代理程式自我簽署憑證（您在上一步中複製的憑證）的複本放入代理程式的 `/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。
  - 針對 `broker_host`，指定代理程式安裝所在的DNS主機名稱。

 Important

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

- （選用）針對 `broker_port`，指定要透過其與代理程式通訊的連接埠。根據預設，代理程式和代理程式會透過連接埠 進行通訊8445。只有在您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，請確定代理程式已設定為使用相同的連接埠。

- 針對 `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，請將 `tls_strict` 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，然後選取下一步。
4. 若要開始安裝，請選擇安裝。
5. 將代理程式的自我簽署憑證（您在上一步中複製的憑證）副本放入代理程式的 `C:\Program Files\NICE\DCVSessionManagerAgent\conf\資料夾` 中。
6. `C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf` 使用您偏好的文字編輯器開啟，然後執行下列動作：
  - 針對 `broker_host`，指定代理程式安裝所在的 DNS 主機名稱。

### Important

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

- (選用) 針對 `broker_port`，指定要透過其與代理程式通訊的連接埠。根據預設，代理程式和代理程式會透過連接埠 進行通訊 8445。只有在您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，請確定代理程式已設定為使用相同的連接埠。
- 針對 `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，請將 `tls_strict` 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。在命令提示中執行下列命令。

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

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

## 步驟 4：將 Amazon DCV 伺服器設定為使用代理程式作為身分驗證伺服器

將 Amazon DCV 伺服器設定為使用代理程式作為外部身分驗證伺服器，以驗證用戶端連線權杖。您還必須設定 Amazon DCV 伺服器以信任代理程式的自我簽署 CA。

### Linux Amazon DCV server

新增 Linux Amazon 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` 參數新增至 `[security]` 區段。
  - 針對 `ca-file`，指定您在上一個步驟中複製到主機之代理程式自我簽署 CA 的路徑。
  - 對於 `auth-token-verifier`，請在代理程式上指定權杖驗證器 URL 的，格式如下：`https://broker_ip_or_dns:port/agent/validate-authentication-token`。指定用於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 Amazon DCV 伺服器。如需詳細資訊，請參閱 [Amazon 管理員指南 中的停止 Amazon DCV Server](#) 和 [啟動 Amazon DCV Server](#)。 DCV

## Windows Amazon DCV server

在 Windows Amazon DCV 伺服器上

1. 開啟 Windows Registry Editor 並導覽至 HKEY\_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/security/ 金鑰。
2. 開啟 ca-file 參數。
3. 針對值資料，指定您在上一個步驟中複製到主機之代理程式自我簽署 CA 的路徑。

### Note

如果參數不存在，請建立新的字串參數並命名為 ca-file。

4. 開啟 auth-token-verifier 參數。
5. 對於值資料，請以下列格式在代理程式上指定權杖驗證器 URL 的  
: https://*broker\_ip\_or\_dns:port*/agent/validate-authentication-token。
6. 指定用於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

如果參數不存在，請建立新的字串參數並命名為 auth-token-verifier。

7. 選擇 OK (確定) 並關閉 Windows 登錄編輯程式。
8. 停止並重新啟動 Amazon DCV 伺服器。如需詳細資訊，請參閱 [Amazon 管理員指南 中的停止 Amazon DCV Server](#) 和 [啟動 Amazon DCV Server](#)。 DCV

## 步驟 5：驗證安裝

設定代理程式、設定代理程式並在 Amazon DCV 伺服器上同時設定兩者後，您需要驗證安裝是否正常運作。

## 主題

- [驗證客服人員](#)
- [驗證代理程式](#)

## 驗證客服人員

安裝代理程式和代理程式後，請確定代理程式正在執行，且能夠連線至代理程式。

### Linux 代理程式主機

要執行的命令取決於 版本。

- 自 2022.0 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 之前的版本

從代理程式主機執行下列命令，並指定目前年份、月份和日期。

```
$ 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，且能夠連線至代理程式，則命令應傳回 success。

如果命令傳回不同的輸出，請檢查客服人員日誌檔案以取得詳細資訊。日誌檔案位於此處：`/var/log/dcv-session-manager-agent/`。

### Windows 代理程式主機

開啟位於 中的代理程式日誌檔案 `C:\ProgramData\NICE\DCVSessionManagerAgent\log`。

如果日誌檔案包含類似下列的行，代理程式正在執行中，且能夠連線至代理程式。

```
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}"
```

如果您的日誌檔案沒有類似的行，請檢查日誌檔案是否有錯誤。

## 驗證代理程式

安裝代理程式和代理程式後，請確定您的代理程式正在執行中，且可從使用者和前端應用程式連線。

從應該能夠到達代理程式的電腦執行下列命令：

```
$ 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項目：

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

# 設定 Amazon DCV Session Manager

為了提供無縫且安全的體驗，請務必根據組織的需求和要求正確設定 Session Manager。本節將逐步解說設定和設定 Session Manager 所涉及的關鍵步驟，包括管理使用者存取、設定網路設定和自訂工作階段設定。

## 主題

- [Scaling Session Manager](#)
- [使用標籤鎖定 Amazon DCV 伺服器](#)
- [設定外部授權伺服器](#)
- [設定代理程式持續性](#)
- [與 Amazon DCV Connection Gateway 整合](#)
- [與亞馬遜集成 CloudWatch](#)

## Scaling Session Manager

若要啟用高可用性並改善效能，您可以將 Session Manager 設定為使用多個代理程式和代理程式。如果您打算使用多個代理程式和代理程式，建議您只安裝和設定一個代理程式和代理程式主機，從這些主機建立 Amazon Machine Images (AMI)，然後從 啟動剩餘的主機AMIs。

依預設，Session Manager 支援使用多個代理程式，而不需要任何額外的組態。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與代理程式之間的流量，以及代理程式與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。

下一節說明如何將 Session Manager 設定為搭配 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：建立代理程式應用程式負載平衡器](#)
- [步驟 4：啟動代理程式](#)
- [步驟 5：建立客服人員應用程式負載平衡器](#)
- [步驟 6：啟動客服人員](#)



## 步驟 1：建立執行個體設定檔

您必須將執行個體設定檔連接至代理程式和代理程式主機，以授予他們使用 Elastic Load Balancing 的許可APIs。如需詳細資訊，請參閱 [IAMAmazon 使用者指南](#) 中的 [Amazon 角色EC2](#)。 EC2

### 建立執行個體設定檔

1. 建立 AWS Identity and Access Management ( IAM ) 角色，定義執行個體設定檔中使用的許可。使用下列信任政策：

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

然後連接下列政策：

```
{
  "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。如需詳細資訊，請參閱 命令參考 [create-instance-profile](#) 中的 。 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 AWS CLI 命令參考 中的 [add-role-to-instance設定檔](#)。
4. 將執行個體設定檔連接至代理程式主機。如需詳細資訊，請參閱 Amazon EC2使用者指南 中的[將IAM角色連接至執行個體](#)。

## 步驟 2：準備負載平衡器的SSL憑證

當您使用 HTTPS作為負載平衡器接聽程式時，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. 建立私有憑證授權機構（CA）AWS Certificate Manager 私有憑證授權機構（ACM PCA）。如需詳細資訊，請參閱 AWS Certificate Manager Private Certificate Authority 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 Private Certificate Authority 使用者指南中的[安裝根CA憑證](#)。 AWS
3. 請求由 CA 簽署的新私有憑證。針對網域名稱，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 [Private Certificate Authority 使用者指南中的請求](#)私有憑證。 AWS

## 步驟 3：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，以平衡前端用戶端與代理程式之間的流量。

建立負載平衡器

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

在導覽窗格中，選擇負載平衡器，然後選擇建立Load Balancer。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. 針對名稱，輸入負載平衡器的描述性名稱。
  - b. 針對結構描述，選取面向網際網路的。
  - c. 針對Load Balancer通訊協定，選取 HTTPS，針對Load Balancer連接埠，輸入 8443。
  - d. 對於 VPC，選取要VPC使用的，然後選擇該中的所有子網路VPC。
  - e. 選擇 Next (下一步)。
3. 對於步驟 2：設定安全設定，執行下列動作：
  - a. 針對憑證類型，選擇從 選擇憑證ACM。
  - b. 針對憑證名稱，選取您先前請求的私有憑證。
  - c. 選擇 Next (下一步)。
4. 對於步驟 3：設定安全群組、建立新的安全群組，或選取允許前端用戶端與代理程式之間透過 HTTPS連接埠 8443 傳入和傳出流量的現有安全群組。

選擇 Next (下一步)。
5. 對於步驟 4：設定路由，請執行下列動作：
  - a. 針對目標群組，選取新目標群組。
  - b. 針對 Name (名稱)，輸入目標群組的名稱。
  - c. 針對目標類型，選擇執行個體。
  - d. 對於通訊協定，請選取 HTTPS。針對連接埠，輸入 8443。針對通訊協定版本，選擇 HTTP1。
  - e. 針對運作狀態檢查通訊協定，選擇 HTTPS，針對路徑，輸入 /health。
  - f. 選擇 Next (下一步)。
6. 對於步驟 5：註冊目標，選擇下一步。
7. 選擇 Create (建立)。

## 步驟 4：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、AMI從代理程式建立，然後使用 AMI啟動其餘代理程式。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。

## 若要啟動代理程式

1. 啟動並設定初始代理程式主機。如需安裝和設定代理程式的詳細資訊，請參閱 [步驟 2：設定 Amazon DCV Session Manager 代理程式](#)。

### Note

由於我們使用應用程式負載平衡器，因此不需要代理程式的自我簽署憑證。

2. 連線至代理程式，`/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. 在行開頭放置雜湊（#）來評論 `broker-to-broker-discovery-addresses` 參數。
  - b. 針對 `broker-to-broker-discovery-aws-region`，輸入您建立應用程式負載平衡器的區域。
  - c. 針對 `broker-to-broker-discovery-aws-alb-target-group-arn`，輸入與代理程式負載平衡器相關聯的目標群組ARN的。
  - d. 儲存並關閉檔案。
3. 停止代理程式執行個體。
4. AMI 從已停止的代理程式執行個體建立。如需詳細資訊，請參閱 Amazon [Linux 執行個體使用者指南](#) 中的 [AMI 從執行個體建立 Linux](#)。EC2
5. 使用 AMI 啟動其餘代理程式。
6. 將您建立的執行個體設定檔指派給所有代理程式執行個體。
7. 指派安全群組，允許代理程式向代理程式和代理程式載入平衡器網路流量至所有代理程式執行個體。如需網路連接埠的詳細資訊，請參閱 [代理程式組態檔案](#)。
8. 將所有代理程式執行個體註冊為代理程式負載平衡器的目標。如需詳細資訊，請參閱 Application Load Balancer 使用者指南中的 [向目標群組註冊目標](#)。

## 步驟 5：建立客服人員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. 在 開啟 Amazon EC2 主控台 <https://console.aws.amazon.com/ec2/>。

在導覽窗格中，選擇負載平衡器，然後選擇建立Load Balancer。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. 針對名稱，輸入負載平衡器的描述性名稱。
  - b. 針對結構描述，選取面向網際網路的。
  - c. 針對Load Balancer通訊協定，選取 HTTPS，針對Load Balancer連接埠，輸入 8445。
  - d. 對於 VPC，選取要VPC使用的，然後選擇該中的所有子網路VPC。
  - e. 選擇 Next (下一步)。
3. 對於步驟 2：設定安全設定，執行下列動作：
  - a. 針對憑證類型，選擇從 選擇憑證ACM。
  - b. 針對憑證名稱，選取您先前請求的私有憑證。
  - c. 選擇 Next (下一步)。
4. 對於步驟 3：設定安全群組，建立新的安全群組，或選取允許代理程式和代理程式透過 HTTPS和連接埠 8445 傳入和傳出流量的現有安全群組。

選擇 Next (下一步)。
5. 對於步驟 4：設定路由，請執行下列動作：
  - a. 針對目標群組，選取新目標群組。
  - b. 針對 Name (名稱)，輸入目標群組的名稱。
  - c. 針對目標類型，選擇執行個體。
  - d. 對於通訊協定，請選取 HTTPS。針對連接埠，輸入 8445。針對通訊協定版本，選擇 HTTP1。
  - e. 針對運作狀態檢查通訊協定，選擇 HTTPS，針對路徑，輸入 /health。
  - f. 選擇 Next (下一步)。
6. 對於步驟 5：註冊目標，選取所有代理程式執行個體，然後選擇新增至已註冊的。選擇下一步：檢閱。
7. 選擇 Create (建立)。

## 步驟 6：啟動客服人員

建立初始代理程式，並將其設定為使用負載平衡器、AMI從代理程式建立，然後使用 AMI 啟動其餘代理程式。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。

若要啟動客服人員

1. 準備 Amazon DCV 伺服器。如需詳細資訊，請參閱[步驟 1：準備 Amazon DCV 伺服器](#)。
2. 放置在 中建立的 CA 公有金鑰複本[步驟 2：準備負載平衡器的SSL憑證](#)。選擇或建立可由任何使用者讀取的目錄。CA 公有金鑰檔案也必須可供任何使用者讀取。
3. 安裝和設定代理程式。如需安裝和設定代理程式的詳細資訊，請參閱[步驟 3：設定 Amazon DCV Session Manager 代理程式](#)。

### Important

修改客服人員組態檔案時：

- 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，輸入在上一個步驟中建立的 CA 公有金鑰檔案路徑

4. 將 Amazon DCV 伺服器設定為使用代理程式作為身分驗證伺服器。如需詳細資訊，請參閱[步驟 4：將 Amazon DCV 伺服器設定為使用代理程式作為身分驗證伺服器](#)。

### Important

修改 Amazon DCV 伺服器組態檔案時：

- 針對 `ca-file` 參數，輸入與上一個步驟中使用的 CA 公有金鑰檔案相同的路徑
- 針對 `auth-token-verifier` 參數，使用的客服人員負載平衡器 DNS  
*broker\_ip\_or\_dns*

5. 停止客服人員執行個體。
6. AMI 從已停止的客服人員執行個體建立。如需詳細資訊，請參閱 Amazon [Linux 執行個體使用者指南](#) 中的[AMI從執行個體建立 Linux](#)。 EC2
7. 使用 AMI 啟動其餘的客服人員，並將您建立的執行個體設定檔指派給所有客服人員。
8. 指派安全群組，允許客服人員將平衡器網路流量載入所有客服人員執行個體。如需網路連接埠的詳細資訊，請參閱[客服人員組態檔案](#)。

## 使用標籤鎖定 Amazon DCV 伺服器

您可以為 Session Manager Agents 指派自訂標籤，以協助識別和分類這些標籤，以及與它們相關聯的 Amazon DCV 伺服器。建立新的 Amazon DCV 工作階段時，您可以根據指派給各自代理程式的標籤，鎖定 Amazon DCV 伺服器群組。如需如何根據代理程式標籤鎖定 Amazon DCV 伺服器的詳細資訊，請參閱 Session Manager 開發人員指南 [CreateSessionRequests](#) 中的。

標籤由標籤索引鍵和值對組成，您可以使用任何對您的使用案例或環境有意義的資訊對。您可以選擇根據其主機硬體組態來標記客服人員。例如，您可以將所有代理程式標記為具有 4 GB 記憶體的主機 `ram=4GB`。或者，您可以根據用途標記客服人員。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`。

### 將標籤指派給客服人員

1. 使用您偏好的文字編輯器，建立新的檔案並為其提供描述性名稱，例如 `agent_tags.toml`。檔案類型必須為 `.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` 格式在新行上新增每個新標籤索引鍵和值對。例如：

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

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

如果目錄包含多個標籤檔案，跨檔案定義的所有標籤都會套用代理程式。檔案會以字母順序讀取。如果多個檔案包含具有相同索引鍵的標籤，則該值會以上次讀取檔案的值覆寫。

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SDKs和客服人員的伺服器。

依預設，Session Manager 會使用代理程式作為授權伺服器，為代理程式的用戶端SDKs和軟體陳述式產生 OAuth 2.0 存取權杖。如果您使用代理程式作為授權伺服器，則不需要額外的組態。

您可以設定 Session Manager 將 Amazon Cognito 用作外部授權伺服器，而不是代理程式。如需 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](#)。Amazon Cognito

使用 [create-user-pool](#) 命令，並指定集區名稱和要在其中建立它的區域。

在此範例中，我們命名集區 `dcv-session-manager-client-app`，並在 `us-east-1` 中建立集區 `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": "15hhd8jij74hf32f24uEXAMPLE",  
    "LastModifiedDate": 1602510048.054,  
    "CreationDate": 1602510048.054,  
    "RefreshTokenValidity": 30,  
    "AllowedOAuthFlowsUserPoolClient": false  
  }  
}
```



```
}
```

記下 `userPoolId`，您需要在下一個步驟中使用它。

2. 為您的使用者集區建立新的網域。使用 [create-user-pool-domain](#) 命令，並指定網域名稱和您在上一個步驟中建立 `userPoolId` 的使用者集區的。

在此範例中，網域名稱為 `mydomain-544fa30f-c0e5-4a02-8d2a-a3761EXAMPLE`，我們會在 `us-east-1` 中建立網域名稱 `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 [create-user-pool-client](#) 命令，並指定您建立 `userPoolId` 的使用者集區的、用戶端的名稱，以及建立集區所在的區域。此外，請包含 `--generate-secret` 選項，以指定您要為要建立的用戶端產生秘密。

在這種情況下，用戶端名稱為 `dcv-session-manager-client-app`，我們會在 `us-east-1` 區域中建立用戶端名稱。

```
$ 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。當開發人員請求 API 請求存取字符時，您需要提供此資訊給開發人員。

4. 為使用者集區建立新的 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。它處理已驗證的存取權杖請求。

使用 [create-resource-server](#) 命令，並指定 userPoolId 使用者集區的、資源伺服器的唯一識別符和名稱、範圍，以及建立該集區的區域。

在此範例中，我們使用 `dcv-session-manager` 作為識別符和名稱，而我們使用 `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 `userPoolId` 使用者集區的、使用者集區用戶端 `ClientId` 的 和 區域。對於 `--allowed-o-auth-flows`，指定以 `client_credentials` 指出用戶端應該使用用戶端 ID 和用戶端秘密的組合，從權杖端點取得存取權杖。對於 `--allowed-o-auth-scopes`，請指定資源伺服器識別符和範圍名稱，如下所示：`resource_server_identifier/scope_name`。包含 `--allowed-o-auth-flows-user-pool-client` 表示用戶端在與 Cognito 使用者集區互動時，可遵循 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並驗證存取權杖。在此範例中，URL授權伺服器的為 `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": "eyJraWQiOiJGQ0VaRFPJUUptT3NSaW41MmtqaDdEbTZyYb0RnSTQ5b2VUT0cxUUU1Q2VJPSIsImF1dG8iOiJkZXkiOj0HIDsd6audjTXKzHlZGScr6R0dZtId5dThkpEziSx0YwiiWe9crAlqoazlDcCsUJHIXDtGKW64pSj3-uQQGg1Jv_tyVjhrA4JbD0k67WS2V9NW-uZ7t4zwwaUm0i3KzpBmi54fpVgPaewiVlUm_aS4LUFcWT6hVJjiZF7om7984qb2g0a14iZxpXPBJTZX_gtG9EtvnS9u",
  "expires_in": 3600,
  "token_type": "Bearer"
}
```

## 7. 使用 `register-auth-server` 命令註冊要與代理程式搭配使用的外部授權伺服器。

```
$ 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 伺服器來請求存取權杖。請求存取權杖時，請在此處提供用戶端 ID、用戶端秘密和URL產生的伺服器。如需請求存取權杖的詳細資訊，請參閱Amazon DCV Session Manager 開發人員指南中的[建立取得存取權杖和提出API請求](#)。

## 設定代理程式持續性

Session Manager 代理程式支援與外部資料庫整合。外部資料庫允許 Session Manager 保留狀態資料和金鑰，以便之後可用。事實上，代理程式資料會分散在叢集上，因此如果主機需要重新啟動或終止叢集，就容易發生資料遺失。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，而無需重新產生金鑰或遺失有關 Amazon DCV Server 開啟或關閉的資訊。

下列類型的資訊可以設定為持續：

- 用於設定工作階段以建立用戶端連線的金鑰
- 機上工作階段資料
- Amazon DCV 伺服器狀態

Amazon DCV Session Manager 支援 DynamoDB、MariaDB 和 MySQL 資料庫。您必須設定和管理其中一個資料庫，才能使用此功能。如果您的代理程式機器託管在 Amazon 上 EC2，我們建議您使用 DynamoDB 作為外部資料庫，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](#)。

## 將代理程式設定為在 DynamoDB 上保留

設定代理程式開始將資料儲存在 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，並進行下列編輯：
  - 設定 `enable-persistence = true`
  - 設定 `persistence-db = dynamodb`
  - 對於 `dynamodb-region` 指定 `&aws`；您要存放包含代理程式資料的資料表的區域。如需支援的區域清單，請參閱 [DynamoDB 服務端點](#)。
  - 對於 `dynamodb-table-rcu` 指定每個資料表支援的讀取容量單位（RCU）數量。如需的詳細資訊 RCU，請參閱 [DynamoDB 佈建容量](#)。

- 對於 `dynamodb-table-wcu` 指定每個資料表支援的寫入容量單位 ( WCU ) 數量。如需的詳細資訊 WCU，請參閱 [DynamoDB 佈建容量](#)。
  - 對於 `dynamodb-table-name-prefix` 指定新增至每個 DynamoDB 資料表的字首 ( 用於使用相同帳戶區分多個代理程式叢集 )。僅允許英數字元、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。執行下列命令來啟動每個代理程式：

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

代理程式主機必須具有呼叫 DynamoDB 的許可 APIs。在 Amazon EC2 執行個體上，會使用 Amazon EC2 中繼資料服務自動擷取憑證。如果您需要指定不同的憑證，您可以使用其中一個支援的憑證擷取技術 ( 例如 Java 系統屬性或環境變數 ) 來設定它們。如需詳細資訊，請參閱 [供應和擷取 AWS 憑證](#)。

。

## 將代理程式設定為保留在 MariaDB /My 上 SQL

### Note

`/etc/dcv-session-manager-broker/session-manager-broker.properties` 檔案包含敏感資料。依預設，其寫入存取權會限制為根目錄，其讀取存取權會限制為根目錄和執行代理程式的使用者。根據預設，這是 `dcvsmbroker` 使用者。代理程式會在啟動時檢查檔案是否具有預期的許可。

設定代理程式開始在 MariaDB /My 上保留其資料 SQL：

1. `/etc/dcv-session-manager-broker/session-manager-broker.properties` 使用您偏好的文字編輯器開啟，並進行下列編輯：
  - 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>` 是資料庫連接埠，`<db_name>` 是資料庫名稱。

- 對於 `jdbc-user`，請指定有權存取資料庫的使用者名稱。
  - `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。針對每個代理程式，執行下列命令：

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

## 與 Amazon DCV Connection Gateway 整合

[Amazon DCV Connection Gateway](#) 是一種可安裝的軟體套件，可讓使用者透過單一存取點存取 Amazon DCV 伺服器機群至 LAN 或 VPC。

如果您的基礎設施包含可透過 Amazon DCV Connection Gateway 存取的 Amazon DCV 伺服器，您可以設定 Session Manager 來整合 Amazon DCV Connection Gateway。遵循下一節中概述的步驟，代理程式將充當 Connection Gateway 的 [Session Resolver](#)。換句話說，代理程式會公開其他 HTTP 端點。Connection Gateway 將 API 呼叫端點，以擷取將 Amazon DCV 連線路由至代理程式選取的主機所需的資訊。

### 主題

- [將 Session Manager Broker 設定為 Amazon DCV Connection Gateway 的 Session Resolver](#)
- [選用 - 啟用 TLS 用戶端身分驗證](#)
- [Amazon DCV Session Manager Amazon DCV 伺服器 - DNS 映射參考](#)

# 將 Session Manager Broker 設定為 Amazon DCV Connection Gateway 的 Session Resolver

## Session Manager 代理程式端

1. `/etc/dcv-session-manager-broker/session-manager-broker.properties` 使用您偏好的文字編輯器開啟，並套用下列變更：
  - 設定 `enable-gateway = true`
  - `gateway-to-broker-connector-https-port` 設為可用TCP連接埠（預設為 8447）
  - `gateway-to-broker-connector-bind-host` 設為代理程式為 Amazon DCV Connection Gateway 連線繫結的主機 IP 地址（預設為 0.0.0.0）
2. 然後執行下列命令來停止和重新啟動代理程式：

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

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

3. 擷取代理程式自我簽署憑證的副本，並將其放入您的使用者目錄中。

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

在下一個步驟中安裝 Amazon DCV Connection Gateway 時，您需要它。

## Amazon DCV Connection Gateway 端

- 請遵循 Amazon DCV Connection Gateway 文件中的 [一節](#)。

由於 Amazon DCV Connection Gateway 會HTTPAPI呼叫代理程式，如果代理程式使用自我簽署憑證，您將需要將代理程式憑證複製到 Amazon DCV Connection Gateway 主機（在上一步中擷取），並在 Amazon DCV Connection Gateway 組態的 `[resolver]` 區段中設定 `ca-file` 參數。



## 選用 - 啟用TLS用戶端身分驗證

完成上一個步驟後，Session Manager 和 Connection Gateway 可以透過安全頻道進行通訊，連線閘道可以驗證 Session Manager Brokers 的身分。如果您需要 Session Manager Brokers 在建立安全頻道之前驗證 Connection Gateway 的身分，則需要按照下一節中的步驟啟用TLS用戶端身分驗證功能。

### Note

如果 Session Manager 位於負載平衡器之後，則無法使用TLS連線終止的負載平衡器啟用 TLS用戶端身分驗證，例如 Application Load Balancer ( ALBs ) 或 Gateway Load Balancer ( ) GLBs。僅支援沒有TLS終止的負載平衡器，例如 Network Load Balancer ( NLBs )。如果您使用 ALBs或 GLBs，您可以強制執行只有特定安全群組可以聯絡負載平衡器，確保額外的安全層級；有關安全群組的詳細資訊，請參閱此處：[您的安全群組 VPC](#)

### Session Manager 代理程式端

1. 若要啟用 Session Manager Brokers 與 Amazon DCV Connection Gateway 之間通訊的TLS用戶端身分驗證，請遵循下列步驟：
2. 透過執行來產生所需的金鑰和憑證：命令的輸出會告訴您憑證產生所在的資料夾，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. 將 Amazon DCV Connection Gateway 的私有金鑰和自我簽署憑證的副本放入您的使用者目錄中。在下一個步驟中，當您在 Amazon DCV Connection Gateway 中啟用TLS用戶端身分驗證時，您將需要該身分驗證。

```
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. 然後使用您偏好的文字編輯器開啟 `/etc/dcv-session-manager-broker/session-manager-broker.properties`，然後執行下列動作：
  - `enable-tls-client-auth-gateway` 設定為 `true`
  - `gateway-to-broker-connector-trust-store-file` 設定為在上一個步驟中建立之 TrustStore 檔案的路徑

- `gateway-to-broker-connector-trust-store-pass` 設定為在上一個步驟中用來建立 TrustStore 檔案的密碼

5. 然後執行下列命令來停止並重新啟動代理程式：

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

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

## Amazon DCV Connection Gateway 端

- 請遵循 Amazon DCV Connection Gateway 文件中的 [一節](#)。
  - 在 [resolver] 區段中設定 `cert-file` 參數時，請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，請使用您在上一個步驟中複製之金鑰檔案的完整路徑

## Amazon DCV Session Manager Amazon DCV 伺服器 - DNS 映射參考

Amazon DCV Connection Gateway 需要 Amazon DCV 伺服器 DNS 的名稱，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 伺服器及其關聯 DNS 名稱之間的映射。

### 檔案結構

映射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:**

識別值所指的 ID 類型；目前可用的值為 `ipAddress` `agentServerId`、和 `instanceId`：

#### **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。此資訊也可用於 `DescribeServers` API 回應。

#### **Id:**

適用於 Amazon EC2 和內部部署基礎設施；`Session Manager Agent UUID` 會在每次主機名稱或 IP 地址變更時建立新的。此資訊可在回應中使用 `DescribeServers` API。

### **Host.Aws.Ec2InstanceId:**

僅適用於 Amazon EC2 執行個體，它可唯一識別機器；執行個體重新啟動後不會變更。可透過聯絡 `http://169.254.169.254/latest/meta-data/instance-id` 在主機上擷取。此資訊也可用於 `DescribeServers` API 回應。

### **ServerId:**

指定類型的 ID，可唯一識別網路中的每個 Amazon DCV 伺服器。

### **DnsNames:**

包含與此物件相關聯之 Amazon DCV 伺服器 DNS 名稱的物件將包含：

#### **InternalDnsNames:**

Amazon DCV Connection Gateway 用來連線至執行個體 DNS 的名稱。

請使用 `Session Manager Broker CLI` 命令從檔案 `register-server-dns-mapping` 載入映射（命令頁面參考：[register-server-dns-mapping](#)），並 `describe-server-dns-mappings` 列出目前載入 `Session Manager Broker` 中的映射（命令頁面參考：[describe-server-dns-mappings](#)）。

## Persistence

我們強烈建議您啟用 `Session Manager Broker` 的持久性功能，以便在多個代理程式或整個叢集停機時防止映射遺失。如需啟用資料持久性的詳細資訊，請參閱[設定代理程式持久性](#)

## 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 Amazon DCV Session Manager

隨著 Amazon DCV 系統規模和複雜性的成長，請務必確保 Session Manager 保持 up-to-date 且能夠處理不斷增加的需求。代理程式和代理程式套件都需要不時升級。本節概述升級 Amazon DCV Session Manager 的程序，涵蓋維護系統的升級程序和建議。

下列主題說明如何升級 Session Manager。

## Note

我們強烈建議您在升級 Session Manager 代理程式之前升級所有 Session Manager 代理程式，以避免在引入新功能時出現不相容問題。

## 主題

- [升級 Amazon DCV Session Manager 代理程式](#)
- [升級 Amazon DCV Session Manager 代理程式](#)

## 升級 Amazon DCV Session Manager 代理程式

Amazon DCV Session Manager 代理程式會收到代理程式的指示，並在各自的 Amazon DCV 伺服器上執行這些指示。作為例行維護的一部分，客服人員需要升級以符合新的標準和要求。本節會逐步引導您完成 Session Manager 客服人員的升級程序。

### Linux host

## Note

下列指示適用於在 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，對於 Amazon Linux、RHEL 和 Centos，請取代 `x86_64` 使用 `aarch64`、和對於 Ubuntu，取代 `amd64` 使用 `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 和 RHEL 7.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.el7.x86_64.rpm
```

- RHEL 8.x 和 Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.el8.x86_64.rpm
```

- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2404.deb
```

- SUSE Linux Enterprise 12

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.sles12.x86_64.rpm
```

- SUSE Linux Enterprise 15

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerAgents/nice-dcv-session-manager-agent-2024.0.781-1.sles15.x86_64.rpm
```

## 3. 安裝套件。

- Amazon Linux 2 和 RHEL 7.x

```
$ sudo yum install -y nice-dcv-session-manager-agent-2024.0.781-1.el7.x86_64.rpm
```

- RHEL 8.x 和 Rocky Linux 8.x

```
$ sudo yum install -y nice-dcv-session-manager-agent-2024.0.781-1.el8.x86_64.rpm
```

- Ubuntu 20.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ sudo apt install ./nice-dcv-session-manager-agent_2024.0.781-1_amd64.ubuntu2404.deb
```

- SUSE Linux Enterprise 12

```
$ sudo zypper install nice-dcv-session-manager-agent-2024.0.781-1.sles12.x86_64.rpm
```

- SUSE Linux Enterprise 15

```
$ sudo zypper install nice-dcv-session-manager-agent-2024.0.781-1.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

### 更新 Windows 主機上的代理程式

1. 停止客服人員服務。在命令提示字元下執行下列命令。

```
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，然後選取下一步。
5. 若要開始安裝，請選擇安裝。
6. 重新啟動代理程式服務。在命令提示字元下執行下列命令。

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

## 升級 Amazon DCV Session Manager 代理程式

Amazon DCV Session Manager 代理程式會將API請求傳遞給相關的代理程式。它們安裝在與 Amazon DCV 伺服器分開的主機上。作為例行維護的一部分，代理程式需要升級以滿足新的標準和要求。本節會逐步引導您完成 Session Manager 代理程式的升級程序。

### 升級代理程式

1. 連線到您要升級代理程式的主機。
2. 停止代理程式服務。

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

3. 下載安裝套件。

- Amazon Linux 2 和 RHEL 7.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1.el7.noarch.rpm
```

- RHEL 8.x 和 Rocky Linux 8.x

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1.el8.noarch.rpm
```

- Ubuntu 20.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1_all.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/SessionManagerBrokers/nice-dcv-session-manager-broker-2024.0.457-1_all.ubuntu2404.deb
```

#### 4. 安裝套件。

- Amazon Linux 2 和 RHEL 7.x

```
$ sudo yum install -y nice-dcv-session-manager-broker-2024.0.457-1.el7.noarch.rpm
```

- RHEL 8.x 和 Rocky Linux 8.x

```
$ sudo yum install -y nice-dcv-session-manager-broker-2024.0.457-1.el8.noarch.rpm
```

- Ubuntu 20.04

```
$ sudo apt install -y nice-dcv-session-manager-broker-2024.0.457-1_all.ubuntu2004.deb
```

- Ubuntu 22.04

```
$ sudo apt install -y nice-dcv-session-manager-broker-2024.0.457-1_all.ubuntu2204.deb
```

- Ubuntu 24.04

```
$ sudo apt install -y nice-dcv-session-manager-  
broker-2024.0.457-1_all.ubuntu2404.deb
```

5. 啟動代理程式服務，並確保它在每次執行個體啟動時自動啟動。

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

## 代理程式CLI參考

Amazon DCV Session Manager 代理程式是命令列介面（CLI）工具，可對 Session Manager 提供管理控制。此參考涵蓋可用於管理 Session Manager 的工作階段、使用者、資源和其他層面的完整CLI命令集。管理員可以自動化例行管理任務、疑難排解問題，並最佳化其 Amazon DCV基礎設施的效能。

如果您使用外部身分驗證伺服器產生 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 Session Manager 代理程式作為 OAuth 2.0 身分驗證伺服器，請使用下列命令。

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

使用下列命令來管理 Session Manager 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名稱映射檔案。

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

# register-auth-server

註冊外部身分驗證伺服器，以便與代理程式搭配使用。

依預設，Session Manager 會使用代理程式作為身分驗證伺服器來產生 OAuth 2.0 存取權杖。如果您使用代理程式作為身分驗證伺服器，則不需要額外的組態。

不過，如果您選擇使用外部身分驗證伺服器，例如 Active Directory 或 Amazon Cognito，則必須使用此命令來註冊外部身分驗證伺服器。

## 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URL的外部身分驗證伺服器的。您必須附加 `.well-known/jwks.json` 至身分驗證伺服器URL。

類型：字串

必要：是

## 範例

下列範例會使用 URL 的 註冊外部身分驗證伺服器 `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

列出已註冊的外部身分驗證伺服器。

### 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URLs外部身分驗證伺服器的。

## 範例

下列範例列出所有已註冊的外部身分驗證伺服器。

### 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

取消註冊外部身分驗證伺服器。取消註冊外部身分驗證伺服器之後，就無法再用來產生 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URL的外部身分驗證伺服器的。您必須附加 `.well-known/jwks.json` 至身分驗證伺服器 URL。

類型：字串

必要：是

### 輸出

#### **Url**

未註冊URL外部身分驗證伺服器的。

## 範例

下列範例會使用 URL 的 註冊外部身分驗證伺服器 `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 Session Manager 用戶端，並產生用戶端憑證，供用戶端用來擷取 2.0 OAuth 存取權杖，這是提出 API 請求所需的。

### Important

確保您將憑證存放在安全的地方。它們稍後無法復原。

只有在代理程式用作 OAuth 2.0 身分驗證伺服器時，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 Session Manager 用戶端的唯一名稱。

類型：字串

必要：是

## 輸出

### **client-id**

Session Manager 用戶端用來擷取 2.0 OAuth 存取權杖的唯一用戶端 ID。

### **client-password**

Session Manager 用戶端用來擷取 2.0 OAuth 存取權杖的密碼。

## 範例

下列範例會註冊名為 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

列出已向代理程式註冊的 Session Manager 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

Session Manager 用戶端的唯一名稱。

### id

Session Manager 用戶端的唯一 ID。

### active

指示 Session Manager 用戶端的狀態。如果用戶端處於作用中狀態，則值為 true；否則，則為 false。

## 範例

下列範例列出已註冊的 Session Manager 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 Session Manager 用戶端。停用的 Session Manager 用戶端無法再使用其憑證來擷取 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**

要停用之 Session Manager 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 `f855b54b-40d4-4769-b792-b727bEXAMPLE` 的 Session Manager 用戶端。

### 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-金鑰

更新代理程式用來簽署 2.0 OAuth 存取權杖的公有和私有金鑰，這些金鑰會轉換為 Session Manager 用戶端。如果您更新金鑰，則必須將新的私有金鑰提供給開發人員，因為需要它才能提出API請求。

### 主題

- [語法](#)
- [範例](#)

### 語法

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

### 範例

下列範例會更新公有金鑰和私有金鑰。

### 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。客服人員需要軟體陳述式才能向代理程式註冊。代理程式擁有軟體陳述式後，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](#)。客服人員向代理程式註冊後，會收到用戶端 ID 和用戶端秘密，用於向代理程式進行身分驗證。

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，也可以選擇產生新的軟體陳述式。如果您產生新的軟體陳述式，則必須將軟體陳述式放入代理程式的新

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代理程式註冊。

## 主題

- [語法](#)
- [輸出](#)
- [範例](#)

## 語法

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

## 輸出

### **software-statement**

軟體陳述式。

## 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:  
ewogICJpZCIgOiAiYjc1NTVhN2QtNWl0MC00OTJhLWJjOTU0TU0tNmUzOWNhYzkyMDcxIiwKICAiYWN0aXZlIiA6IHRydWUsCi
```

## describe-software-statements

描述現有的軟體陳述式。

## 主題

- [語法](#)



```
"is-active" : "true"  
} ]
```

## deactivate-software-statement

停用軟體陳述式。當您停用軟體陳述式時，就無法再將其用於客服人員註冊。

主題

- [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  
EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWl0MC00TJhLWJj0TUtNmUz0WNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj
```

輸出

**Software statement**

```
EXAMPLEpZCIg0iAiYjc1NTVhN2QtNWI0MC00OTJhLWJjOTUtNmUzOWNhYzkxMDcxIiwKICAiaXNEXAMPLEQiIDogMTU5Nj  
deactivated
```

## describe-agent-clients

描述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，否則為 false。

### 範例

下列範例說明客服人員。

#### 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 - DNS名稱映射來自JSON檔案。

## 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 - DNS名稱映射的檔案路徑。

類型：字串

必要：是

## 範例

下列範例會註冊 DCV 伺服器 - DNS名稱映射來自 檔案 /tmp/mappings.json。

## 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名稱映射。

## 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 dns 名稱

#### **externalDnsNames**

外部 dns 名稱

## 範例

下列範例列出已註冊的DCV伺服器 - DNS名稱映射。

## 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

本參考章節提供 Session Manager 可用組態選項的概觀。組態包括對代理程式和代理程式檔案的變更。每個組態都包含目的說明、接受的值，以及對整體系統行為的影響。Amazon DCV Session Manager 可以自訂以滿足 Amazon DCV 系統的獨特需求。

### 主題

- [代理程式組態檔案](#)
- [代理程式組態檔案](#)

## 代理程式組態檔案

代理程式組態檔案 ( `/etc/dcv-session-manager-broker/session-manager-broker.properties` ) 包含可設定為自訂 Session Manager 功能的參數。您可以使用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。依預設，其寫入存取權會限制為根目錄，其讀取存取權會限制為根目錄和執行代理程式的使用者。根據預設，這是 `dcvsmbroker` 使用者。代理程式會在啟動時檢查檔案是否具有預期的許可。

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
<code>broker-ja-home</code>	否		指定代理程式將使用的 Java 主目錄路徑，而不是系統預設目錄。如果設定，代理程式將在啟動 <code>&lt;broker-java-home&gt;/bin/java</code> 時使用。

參數名稱	必要	預設值	描述
			<p>秘訣：代理程式需要 Java Runtime Environment 11，如果缺少作為成功安裝的相依項目，則會進行安裝。如果第 11 版未設定為預設 Java 環境，則可以使用下列命令擷取其主目錄：</p> <pre>\$ sudo alternatives --display java</pre>
session-screenshots-max-width	否	160	指定使用 GetSessionScreenshots 擷取之工作階段螢幕擷取畫面的最大寬度，以像素為單位API。
session-screenshots-max-height	否	100	以像素為單位，指定使用 GetSessionScreenshots 擷取之工作階段螢幕擷取畫面的最大高度API。
session-screenshots-format	否	png	使用 GetSessionScreenshots 擷取的工作階段螢幕擷取畫面的影像檔案格式API。

參數名稱	必要	預設值	描述
create-sessions-queue-max-size	否	1000	可排入佇列的未滿足 CreateSessionsAPI 請求數量上限。當佇列已滿時，新的未完成請求會遭到拒絕。
create-sessions-queue-max-time-seconds	否	1800	未完成 CreateSessionsAPI 請求可保留在佇列中的時間上限，以秒為單位。如果無法在指定的時間內完成請求，則會失敗。
session-manager-working-path	是	/tmp	指定代理程式寫入操作所需檔案的目錄路徑。此目錄必須只能由代理程式存取。
enable-authorization-server	是	true	指定代理程式是否為用來產生用戶端 OAuth 2.0 存取權杖的身分驗證伺服器 APIs。

參數名稱	必要	預設值	描述
enable-user-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必須在提出API請求時提供存取權杖。如果您停用用戶端授權，用戶端APIs可以在沒有存取權杖的情況下提出請求。
enable-agent-authorization	是	true	啟用或停用客服人員授權。如果您啟用代理程式授權，代理程式必須在與代理程式通訊時提供存取權杖。
delete-session-duration-hours	否	1	指定刪除的工作階段變成不可見且不再被DescribeSession API呼叫傳回的小時數。
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	指定代理程式接聽用戶端連線的HTTPS連接埠。
client-to-broker-connect-bind-host	否	0.0.0.0	指定代理程式為用戶端連線繫結的主機 IP 地址。
client-to-broker-connect-key-store-file	是		指定用於TLS用戶端連線的金鑰存放區。

參數名稱	必要	預設值	描述
client-to-broker-connect-key-store-pass	是		指定金鑰存放區傳遞。
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	指定代理程式為代理程式連線繫結的主機 IP 地址。

參數名稱	必要	預設值	描述
agent-to-broker-connectokey-store-file	是		指定用於TLS代理程式連線的金鑰存放區。
agent-to-broker-connectokey-store-pass	是		指定金鑰存放區傳遞。
broker-to-broker-port	是	47100	指定用於連線的 broker-to-broker連接埠。
broker-to-broker-bind-host	否	0.0.0.0	指定代理程式為 broker-to-broker連線繫結的主機 IP 地址。

參數名稱	必要	預設值	描述
broker-to-broker-discovery-port	是	47500	指定代理程式用來互相探索的連接埠。
broker-to-broker-discovery-address	否		指定 機群中其他代理程式的 IP 地址和連接埠 <i>ip_address :port</i> 格式。如果有多個代理程式，請使用逗號分隔值。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		指定探索的 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		指定探索的 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		指定用於代理程式探索的應用程式負載平衡器 AWS 區域。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。
broker-to-broker-discovery-AWS-alb-target-group-arn	否		用於 broker-to-broker 探索 ARN 的應用程式負載平衡器目標群組使用者的。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	指定單一代理程式用來存放 Amazon DCV 工作階段資料之關閉堆積記憶體的最大數量。
broker-to-broker-key-store-file	是		指定用於 TLS 代理程式連線的金鑰存放區。
broker-to-broker-key-store-pass	是		指定金鑰存放區傳遞。
enable-cloud-watch-metrics	否	false	啟用或停用 Amazon CloudWatch 指標。如果您啟用 CloudWatch 指標，您可能需要指定的值 <code>cloud-watch-region</code> 。

參數名稱	必要	預設值	描述
cloud-watch-region	否	只有在 enable-cloud-watch-metrics 設定為 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 擷取該區域 IMDS。	發佈 CloudWatch 指標 AWS 的區域。
max-api-requests-per-second	否	1000	指定代理程式 api 每秒可以處理的最大請求數，然後再進行限流。
enable-throw-forwarder-header	否	false	如果設定為 true 限流，則會在出現時從 X-Forwarded-For 標頭擷取呼叫者 ip。
create-session-number-of-retries-on-failure	否	2	指定在 Amazon DCV 伺服器主機上建立工作階段請求失敗後要執行的重試次數上限。將 設為 0 以永遠不對失敗執行重試。



參數名稱	必要	預設值	描述
autorun-file-arguments-max-size	否	50	指定可傳遞至自動執行檔案的引數數目上限。
autorun-file-arguments-max-argument-length	否	150	指定每個自動執行檔案引數的字元長度上限。
enable-persistence	是	false	如果設定為 true，代理程式狀態資料會保留在外部資料庫上。
persistence-db	否	只有在 enable-persistence 設定為 true 時才需要 true。	指定用於持久性的資料庫。唯一支援的值是：dynamodb 和 mysql。
dynamodb-region	否	只有在 enable-persistence 設為 true 且 persistence-db 設為 true 時才需要 dynamodb。	指定建立和存取 DynamoDB 資料表的區域。

參數名稱	必要	預設值	描述
dynamodb-table-rcu	否	只有在 enable-persistence 設為 true 且 persistence-db 設為 true 時才需要 dynamodb。	指定每個 DynamoDB 資料表的讀取容量單位 (RCU) 所需的詳細資訊 RCU，請參閱 <a href="#">佈建容量的定價</a> 。
dynamodb-table-wcu	否	只有在 enable-persistence 設為 true 且 persistence-db 設為 true 時才需要 dynamodb。	指定每個 DynamoDB 資料表的寫入容量單位 (WCU)。所需的詳細資訊 WCU，請參閱 <a href="#">佈建容量的定價</a> 。
dynamodb-table-name-prefix	否	只有在 enable-persistence 設為 true 且 persistence-db 設為 true 時才需要 dynamodb。	指定新增至每個 DynamoDB 資料表的字首 (可用來區分使用相同 AWS 帳戶的多個代理程式叢集)。僅允許英數字元、點、破折號和底線。
jdbc-connection-url	否	只有在 enable-persistence 設為 true 且 persistence-db 設為 true 時才需要 mysql。	<p>指定 URL 與 MariaDB / MySQL 資料庫的連線；其中包含端點和資料庫名稱。URL 應具有此格式：</p> <pre>jdbc:mysql://&lt;db_endpoint&gt;:&lt;db_port&gt;/&lt;db_name&gt;?createDatabaseIfNotExist=true</pre> <p>其中 &lt;db_endpoint&gt; 是 MariaDB/MySQL 資料庫端點，&lt;db_port&gt; 是資料庫連接埠，而 &lt;db_name&gt; 是資料庫名稱。</p>

參數名稱	必要	預設值	描述
jdbc-user	否	只有在 enable-persistence 設為 true 且 persistence-db 設為 時才需要mysql。	指定有權存取 MariaDB/我的SQL資料庫的使用者名稱。
jdbc-password	否	只有在 enable-persistence 設為 true 且 persistence-db 設為 時才需要mysql。	指定有權存取 MariaDB/我的SQL資料庫的使用者的密碼。
seconds-before-deleting-unreachable-dcv-server	否	1800	指定從系統刪除無法連線伺服器後的秒數。

## 代理程式組態檔案

代理程式組態檔案 ( /etc/dcv-session-manager-agent/agent.conf 適用於 Linux 和 C:\Program Files\NICE\DCVSessionManagerAgent\conf\agent.conf 適用於 Windows ) 包含可設定為自訂 Session Manager 功能的參數。您可以使用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
<code>agent.ker_hostname</code>	是		指定代理程式主機DNS的名稱。
<code>agent.ker_port</code>	是	8445	指定要與代理程式通訊的連接埠。
<code>agent.config_file</code>	否		只有在 <code>tls_strict</code> 設定為 <code>true</code> 時才需要。指定驗證憑證所需的TLS憑證 ( <code>.pem</code> ) 檔案路徑。將自我簽署憑證從代理程式複製到代理程式。
<code>agent.init_folder</code>	否	<ul style="list-style-type: none"> <li><code>/var/lib/dcv-session-manager-agent/init</code> ( 林克斯 )</li> </ul>	指定主機伺服器上資料夾的路徑，用於存放建立 Amazon DCV 伺服器工作階段時允許初始化的自訂指令碼。您必須指定絕對路徑。資料夾必須可供存取，且檔案必須由使用 <code>CreateSessions InitFile</code> 請求參數的使用者執行API。
<code>agent.tls_strict</code>	否	<code>true</code>	指示是否應使用嚴格TLS驗證。
<code>agent.software_statement_path</code>	否		只有在不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「 <a href="#">generate-software-statement</a> 」。
<code>agent.tags_folder</code>	否	<ul style="list-style-type: none"> <li><code>/etc/dcv-session-manager-agent</code> ( 林克斯 )</li> </ul>	指定標籤檔案所在的資料夾路徑。如需詳細資訊，請參閱「 <a href="#">generate-software-statement</a> 」。

參數名稱	必要	預設值	描述
		<ul style="list-style-type: none"> <li>C:\Program Files\NICE\DCVSessionManagerAgent\conf\tags (Windows)</li> </ul>	閱「 <a href="#">使用標籤鎖定 Amazon DCV 伺服器</a> 」。
agent.autorun_folder	否	<ul style="list-style-type: none"> <li>/var/lib/dcv-session-manager-agent/autorun (林克斯)</li> <li>C:\ProgramData\NICE\DcvSessionManagerAgent\autorun (Windows)</li> </ul>	指定主機伺服器上資料夾的路徑，用於存放允許在工作階段啟動時自動執行的指令碼和應用程式。您必須指定絕對路徑。資料夾必須可供存取，且檔案必須由使用 CreateSessions AutorunFile 請求參數的使用者執行 API。
agent.max_virtual_sessions	否	-1 (無限制)	可以使用 Amazon Session DCV Manager 在 Amazon DCV 伺服器上建立的虛擬工作階段數量上限。
agent.max_concurrent_sessions_per_user	否	1	可由單一使用者使用 Amazon DCV Session Manager 在 Amazon DCV 伺服器上建立的虛擬工作階段數量上限。

參數名稱	必要	預設值	描述
agent.updater_interval	否	30	指定傳送更新後的資料給代理程式前要等待的秒數。傳送的資料包括 Amazon 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 Session Manager 更了解代理程式執行所在的系統上發生的變化，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 Session Manager 對系統變更的回應會變少，因此120不建議高於 的值。
log.level	否	info	指定日誌檔案的動量層級。可用的詳細資訊等級如下： <ul style="list-style-type: none"> <li>• error- 提供最少的詳細資訊。僅包含錯誤。</li> <li>• warning- 包含錯誤和警告。</li> <li>• info- 預設動詞層級。包含錯誤、警告和資訊訊息。</li> <li>• debug- 提供詳細資訊。提供有助於偵錯問題的詳細資訊。</li> </ul>

參數名稱	必要	預設值	描述
log.directory	否	<ul style="list-style-type: none"> <li>/var/log/dcv-session-manager/ ( 林克斯 )</li> <li>C:\ProgramData\NICE\DCVSessionManagerAgent\log (Windows)</li> </ul>	指定要在其中建立日誌檔案的目錄。
log.rotation	否	daily	<p>指定日誌檔案輪換。有效的值如下：</p> <ul style="list-style-type: none"> <li>hourly- 日誌檔案每小時輪換一次。</li> <li>daily- 日誌檔案每天輪換。</li> </ul>
log.max-file-size	否	10485760	當日誌檔案大小達到指定的位元組大小時，它將被輪換。將建立新的日誌檔案，並將進一步的日誌事件放置在新的檔案中。
log.rotate	否	9	輪換中保留的日誌檔案數目上限。每次輪換發生且達到此數字時，都會刪除最舊的日誌檔案。

# Amazon DCV Session Manager 的版本備註和文件歷史記錄

此頁面提供 Amazon DCV Session Manager 的版本備註和文件歷史記錄。

## 主題

- [Amazon DCV Session Manager 版本備註](#)
- [文件歷史紀錄](#)

## Amazon DCV Session Manager 版本備註

本節提供 Amazon DCV Session Manager 主要更新、功能版本和錯誤修正的概觀。所有更新都會依發行日期進行組織。我們會經常更新文件，以處理您傳送給我們的意見回饋。

## 主題

- [2024.0-457 — 2024 年 10 月 1 日](#)
- [2023.1-17652 — 2024 年 8 月 1 日](#)
- [2023.1-16388 — 2024 年 6 月 26 日](#)
- [2023.1 — 2023 年 11 月 9 日](#)
- [2023.0-15065 — 2023 年 5 月 4 日](#)
- [2023.0-14852 — 2023 年 3 月 28 日](#)
- [2022.2-13907 — 2022 年 11 月 11 日](#)
- [2022.1-13067 — 2022 年 6 月 29 日](#)
- [2022.0-11952 — 2022 年 2 月 23 日](#)
- [2021.3-11591 — 2021 年 12 月 20 日](#)
- [2021 年 2 月 11445 日 — 2021 年 11 月 18 日](#)
- [2021.2-11190 — 2021 年 10 月 11 日](#)
- [2021.2-11042 — 2021 年 9 月 1 日](#)
- [2021.1-10557 — 2021 年 5 月 31 日](#)
- [2021 年 0 月 10242 日 — 2021 年 4 月 12 日](#)
- [2020.2-9662 — 2020 年 12 月 4 日](#)
- [2020.2-9508 — 2020 年 11 月 11 日](#)



## 2024.0-457 — 2024 年 10 月 1 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"><li>代理程式：457</li><li>客服人員：748</li><li>CLI：140</li></ul>	<ul style="list-style-type: none"><li>重新命名NICEDCV為 Amazon DCV。</li><li>新增對 Ubuntu 24.04 的支援。</li></ul>

## 2023.1-17652 — 2024 年 8 月 1 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"><li>代理程式：426</li><li>客服人員：748</li><li>CLI：140</li></ul>	<ul style="list-style-type: none"><li>錯誤修正與效能改進。</li></ul>

## 2023.1-16388 — 2024 年 6 月 26 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"><li>代理程式：417</li><li>客服人員：748</li><li>CLI：140</li></ul>	<ul style="list-style-type: none"><li>已修正錯誤顯示記憶體為 TB 而非 GB 的錯誤。</li><li>錯誤修正與效能改進。</li></ul>

## 2023.1 — 2023 年 11 月 9 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"><li>代理程式：410</li><li>客服人員：732</li><li>CLI：140</li></ul>	<ul style="list-style-type: none"><li>錯誤修正與效能改進</li></ul>

## 2023.0-15065 — 2023 年 5 月 4 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"><li>代理程式：392</li><li>客服人員：675</li><li>CLI：132</li></ul>	<ul style="list-style-type: none"><li>新增對ARM平台上 Red Hat Enterprise Linux 9、Rocky Linux 9 和 CentOS Stream 9 的支援。</li></ul>

## 2023.0-14852 — 2023 年 3 月 28 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"><li>代理程式：392</li><li>客服人員：642</li><li>CLI：132</li></ul>	<ul style="list-style-type: none"><li>新增對 Red Hat Enterprise Linux 9、Rocky Linux 9 和 CentOS Stream 9 的支援。</li></ul>

## 2022.2-13907 — 2022 年 11 月 11 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"><li>代理程式：382</li><li>客服人員：612</li><li>CLI：123</li></ul>	<ul style="list-style-type: none"><li>在DescribeSessions 回應中新增Substate欄位。</li><li>已修正可能導致 CLI無法連線至代理程式的問題，具體取決於使用URL 中的。</li></ul>

## 2022.1-13067 — 2022 年 6 月 29 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"><li>代理程式：355</li><li>客服人員：592</li><li>CLI：114</li></ul>	<ul style="list-style-type: none"><li>新增在 AWS Graviton 執行個體上執行代理程式的支援。</li><li>新增了對 Ubuntu 22.04 的代理程式和代理程式支援。</li></ul>

## 2022.0-11952 — 2022 年 2 月 23 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"><li>代理程式：341</li><li>客服人員：520</li><li>CLI：112</li></ul>	<ul style="list-style-type: none"><li>已將日誌輪換功能新增至客服人員。</li><li>新增組態參數，以在代理程式中設定 Java 首頁。</li><li>改善了從快取到代理程式中磁碟的資料排清。</li><li>已修正 中的URL驗證CLI。</li></ul>

## 2021.3-11591 — 2021 年 12 月 20 日

建置編號	新功能
<ul style="list-style-type: none"><li>代理程式：307</li><li>客服人員：453</li><li>CLI：92</li></ul>	<ul style="list-style-type: none"><li>新增了與 Amazon DCV Connection Gateway 整合的支援。</li><li>新增了對 Ubuntu 18.04 和 Ubuntu 20.04 的代理程式支援。</li></ul>

## 2021 年 2 月 11445 日 — 2021 年 11 月 18 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"><li>代理程式：288</li><li>客服人員：413</li><li>CLI：54</li></ul>	<ul style="list-style-type: none"><li>已修正包含 Windows 網域的登入名稱驗證問題。</li></ul>

## 2021.2-11190 — 2021 年 10 月 11 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"><li>代理程式：254</li><li>客服人員：413</li><li>CLI：54</li></ul>	<ul style="list-style-type: none"><li>已修正命令列介面中無法啟動 Windows 工作階段的問題。</li></ul>

## 2021.2-11042 — 2021 年 9 月 1 日

建置編號	新功能	變更與錯誤修正
<ul style="list-style-type: none"> <li>代理程式：254</li> <li>客服人員：413</li> <li>CLI：37</li> </ul>	<ul style="list-style-type: none"> <li>Amazon DCV Session Manager 現在提供命令列介面（CLI）支援。您可以在 中建立和管理 Amazon DCV 工作階段 CLI，而不是呼叫 APIs。</li> <li>Amazon DCV Session Manager 引入了代理程式資料持久性。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。</li> </ul>	<ul style="list-style-type: none"> <li>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 格式化 Web 權杖的演算法。透過此變更，您可以使用 Azure AD 作為外部授權伺服器。</li> </ul>

## 2021.1-10557 — 2021 年 5 月 31 日

建置編號	新功能	變更與錯誤修正
<ul style="list-style-type: none"> <li>代理程式：214</li> <li>客服人員：365</li> </ul>	<ul style="list-style-type: none"> <li>Amazon DCV Session Manager 新增了對傳遞至 Linux 上自動執行檔案之輸入參數的支援。</li> <li>現在可以將伺服器屬性作為需求傳遞至 <a href="#">CreateSessions</a> API。</li> </ul>	<ul style="list-style-type: none"> <li>我們修復了 Windows 上自動執行檔案的問題。</li> </ul>

## 2021 年 0 月 10242 日 — 2021 年 4 月 12 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> <li>代理程式：183</li> <li>客服人員：318</li> </ul>	<ul style="list-style-type: none"> <li>Amazon DCV Session Manager 推出下列新的 APIs： <ul style="list-style-type: none"> <li><a href="#">OpenServers</a></li> <li><a href="#">CloseServers</a></li> <li><a href="#">DescribeServers</a></li> <li><a href="#">GetSessionScreenshots</a></li> </ul> </li> </ul>

建置編號	變更與錯誤修正
	<ul style="list-style-type: none"> <li>它還引入了下列新組態參數： <ul style="list-style-type: none"> <li><a href="#">代理程式參數</a>：<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>。</li> <li><a href="#">客服人員參數</a>：<code>max_virtual_sessions</code>、<code>agent.autorun_folder</code> 和 <code>max_concurrent_sessions_per_user</code>。</li> </ul> </li> </ul> <p><a href="#">客服人員參數</a>：<code>agent.autorun_folder</code>、<code>max_virtual_sessions</code>、和 <code>max_concurrent_sessions_per_user</code>。</p> <p><a href="#">客服人員參數</a>：<code>agent.autorun_folder</code>、<code>max_virtual_sessions</code>、和 <code>max_concurrent_sessions_per_user</code>。</p>

## 2020.2-9662 — 2020 年 12 月 4 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> <li>代理程式：114</li> <li>客服人員：211</li> </ul>	<ul style="list-style-type: none"> <li>我們修復了自動產生的 TLS 憑證的問題，這些憑證會阻止代理程式啟動。</li> </ul>

## 2020.2-9508 — 2020 年 11 月 11 日

建置編號	變更與錯誤修正
<ul style="list-style-type: none"> <li>代理程式：78</li> <li>客服人員：183</li> </ul>	<ul style="list-style-type: none"> <li>Amazon DCV Session Manager 的初始版本。</li> </ul>

## 文件歷史紀錄

下表說明此版本 Amazon DCV Session Manager 的文件。

變更	描述	日期
Amazon 2024.0-457 DCV版	已更新 Amazon 2024.0-457 DCV 的 Amazon DCV Session Manager。如需詳細資訊，請參閱 <a href="#">2024.0-457 — 2024 年 10 月 1 日</a> 。	2024 年 9 月 30 日
Amazon 2023.1-17 652 DCV版	已更新 Amazon 2023.1-17652 DCV 的 Amazon DCV Session Manager。如需詳細資訊，請參閱 <a href="#">2023.1-17652 — 2024 年 8 月 1 日</a> 。	2024 年 8 月 1 日
Amazon 2023.1-16 388 DCV版	已更新 Amazon 2023.1-16388 DCV 的 Amazon DCV Session Manager。如需詳細資訊，請參閱 <a href="#">2023.1-16388 — 2024 年 6 月 26 日</a> 。	2024 年 6 月 26 日
Amazon 2023.1 DCV版	已更新 Amazon 2023.1 DCV 的 Amazon DCV Session Manager。如需詳細資訊，請參閱 <a href="#">2023.1 — 2023 年 11 月 9 日</a> 。	2023 年 11 月 9 日
Amazon 2023.0 DCV版	已更新 Amazon 2023.0 DCV 的 Amazon DCV Session Manager。如需詳細資訊，請參閱 <a href="#">2023.0-14852 — 2023 年 3 月 28 日</a> 。	2023 年 3 月 28 日
Amazon 2022.2 DCV版	已更新 Amazon 2022.2 DCV 的 Amazon DCV Session Manager。如需詳細資訊，請參閱 <a href="#">2022.2-13907 — 2022 年 11 月 11 日</a> 。	2022 年 11 月 11 日

變更	描述	日期
Amazon 2022.1 DCV版	已更新 Amazon 2022.1 DCV 的 Amazon DCV Session Manager。如需詳細資訊，請參閱 <a href="#">2022.1-13067 — 2022 年 6 月 29 日</a> 。	2022 年 6 月 29 日
Amazon 2022.0 DCV版	已更新 Amazon 2022.0 DCV 的 Amazon DCV Session Manager。如需詳細資訊，請參閱 <a href="#">2022.0-11952 — 2022 年 2 月 23 日</a> 。	2022 年 2 月 23 日
Amazon 2021.3 DCV版	已更新 Amazon 2021.3 DCV 的 Amazon DCV Session Manager。如需詳細資訊，請參閱 <a href="#">2021.3-11591 — 2021 年 12 月 20 日</a> 。	2021 年 12 月 20 日
Amazon 2021.2 DCV版	已更新 Amazon 2021.2 DCV 的 Amazon DCV Session Manager。如需詳細資訊，請參閱 <a href="#">2021.2-11042 — 2021 年 9 月 1 日</a> 。	2021 年 9 月 1 日
Amazon 2021.1 DCV版	已更新 Amazon 2021.1 DCV 的 Amazon DCV Session Manager。如需詳細資訊，請參閱 <a href="#">2021.1-10557 — 2021 年 5 月 31 日</a> 。	2021 年 5 月 31 日
Amazon 2021.0 DCV版	已更新 Amazon 2021.0 DCV 的 Amazon DCV Session Manager。如需詳細資訊，請參閱 <a href="#">2021 年 0 月 10242 日 — 2021 年 4 月 12 日</a> 。	2021 年 4 月 12 日
Amazon DCV Session Manager 的初始版本	此內容的第一次發佈。	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。