

# 使用和管理 vRealize Automation Cloud Assembly

2022 年 10 月

vRealize Automation 8.0

您可以在 VMware 網站上找到最新的技術文件，網址如下：

<https://docs.vmware.com/tw/>

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

Copyright © 2022 VMware, Inc. 保留所有權利。 [版權與商標資訊](#)。

# 目錄

<b>1</b>	<b>什麼是 vRealize Automation Cloud Assembly</b>	<b>6</b>
	vRealize Automation Cloud Assembly 如何運作	6
<b>2</b>	<b>為您的組織設定 vRealize Automation Cloud Assembly</b>	<b>9</b>
	什麼是 vRealize Automation Cloud Assembly 使用者角色	9
	新增雲端帳戶	12
	使用雲端帳戶所需的認證	12
	在 vRealize Automation 中建立 Microsoft Azure 雲端帳戶	30
	在 vRealize Automation 中建立 Amazon Web Services 雲端帳戶	30
	建立 Google Cloud Platform 雲端帳戶	31
	建立 vCenter 雲端帳戶	32
	建立 NSX-V 雲端帳戶	33
	建立 NSX-T 雲端帳戶	34
	建立 VMware Cloud on AWS 雲端帳戶	36
	與其他應用程式整合	37
	如何使用 GitLab 和 GitHub 整合	37
	設定外部 IPAM 整合點	40
	如何升級到較新的 IPAM 整合套件	42
	在 vRealize Automation Cloud Assembly 中設定 My VMware 整合	42
	在 Cloud Assembly 中設定 vRealize Orchestrator 整合	43
	如何在 vRealize Automation Cloud Assembly 中使用 Kubernetes	45
	什麼是 vRealize Automation Cloud Assembly 中的組態管理	52
	如何在 vRealize Automation Cloud Assembly 中建立 Active Directory 整合	57
	上線計劃是什麼	58
	將選取的機器做為單一部署上線	59
	使規則篩選的機器上線做為單獨部署	61
	進階組態	66
	如何設定網際網路 Proxy 伺服器	66
	如何使用 cloud-init 或 cloudbase-init 設定 Windows 範本	70
	如何使用 IPAM SDK 建立提供者特定的外部 IPAM 整合套件	70
<b>3</b>	<b>vRealize Automation Cloud Assembly 使用案例</b>	<b>72</b>
	WordPress 使用案例	72
	建立基礎結構	72
	建立專案	80
	建立和擴充藍圖	81
	VMware Cloud on AWS 使用案例	97

設定基本 VMware Cloud on AWS 工作流程	98
在 VMware Cloud on AWS 中設定隔離網路	109
提供者特定的外部 IPAM 整合使用案例	113
部署下載套件之前，在 Infoblox 應用程式中新增必要的可擴充屬性	114
下載並部署外部 IPAM 提供者套件	116
為 IPAM 整合點建立執行環境	117
新增外部 IPAM 整合點	118
將網路和網路設定檔設定為使用 IPAM 提供者值	121
定義和部署使用 IPAM 提供者範圍指派的藍圖	122
針對 IPAM 整合使用 Infoblox 特定內容	125

## 4 建置資源基礎結構 127

如何新增雲端區域	127
進一步瞭解雲端區域	127
如何新增類型模板對應	129
進一步瞭解類型模板對應	129
如何新增映像對應	130
進一步瞭解映像對應	130
如何新增網路設定檔	132
進一步瞭解網路設定檔	133
使用網路和網路設定檔中的 IP 位址	137
使用網路和網路設定檔	138
使用負載平衡器設定	140
如何新增儲存區設定檔	142
進一步瞭解儲存區設定檔	142
如何使用標籤	142
建立標記策略	144
在 vRealize Automation Cloud Assembly 中使用功能標籤	145
在 vRealize Automation Cloud Assembly 中使用限制標籤	146
標準標籤	147
vRealize Automation Cloud Assembly 如何處理標籤	148
如何設定簡單的標記結構	148
如何使用資源	150
計算資源	150
網路資源	150
安全資源	151
儲存資源	152
機器資源	152
磁碟區資源	153
進一步瞭解資源	153



## 5 新增和管理專案 158

- 如何為我的開發團隊新增專案 158
- 進一步瞭解專案 159
  - 使用專案標籤和自訂內容 159
  - 專案在部署時的運作方式 161

## 6 設計部署 163

- 在建立藍圖之前 164
- 建立藍圖的方式 164
- 如何從頭建立簡單藍圖 166
  - 如何選取元件並將其新增至藍圖 166
  - 如何連線藍圖資源 167
  - 如何建立有效的藍圖代碼 168
- 如何強化簡單藍圖 169
  - 使用者輸入如何自訂藍圖 170
  - 如何設定元件部署順序 175
  - 如何使用運算式使藍圖代碼更加多樣化 176
  - 如何在藍圖中自動初始化機器 184
  - 如何在藍圖中啟用遠端存取 191
- 如何儲存不同版本的藍圖 194
- 如何自訂已部署資源的名稱 196
- 什麼是資源內容 197
- 可使用的一些藍圖代碼範例 197
  - 藍圖中的 vSphere 元件範例 198
  - 可檢閱的藍圖 201
  - 網路、安全性和負載平衡器藍圖範例 208
  - 啟用 Puppet 的具有使用者名稱和密碼存取權的藍圖 212
- 如何使用市集 221
- 如何使用擴充性延長和自動化應用程式生命週期 221
  - 擴充性動作訂閱 222
  - 擴充性工作流程訂閱 239
  - 進一步瞭解擴充性訂閱 245

## 7 管理部署 253

- 如何監控作用中部署 254
- 如果 vRealize Automation Cloud Assembly 部署失敗，該如何解決 255
- 如何管理已完成部署的生命週期 257
- 您可以對部署執行哪些動作 259

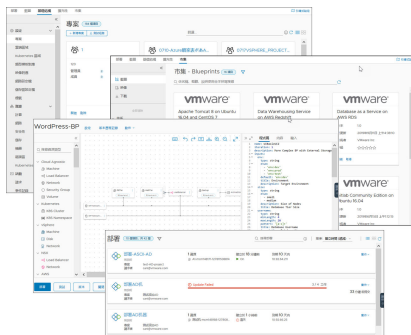
# 什麼是 vRealize Automation Cloud Assembly

1

您使用 vRealize Automation Cloud Assembly 連線到公有雲和私有雲提供者，以便您可以部署機器、應用程式，以及為這些資源建立的服務。您和您的團隊在環境中開發藍圖即程式碼，此環境支援反覆的工作流程，從開發到測試再到生產。在佈建期間，您可以跨一系列雲端廠商進行部署。服務是一個受管理的 VMware SaaS 和 NaaS 型架構。

vRealize Automation Cloud Assembly 的概觀包括下列基本功能。

- 您可以透過 [基礎結構] 索引標籤新增並組織整理雲端廠商資源和使用者。此標籤也提供已部署藍圖的相關資訊。
- [市集] 索引標籤提供 VMware Solution Exchange 藍圖和映像，可協助您建置藍圖程式庫和存取支援的 OVA 或 OVF。
- [藍圖] 索引標籤是開發首頁。您使用畫布和 YAML 編輯器開發，然後再部署機器和應用程式。
- [部署] 索引標籤會顯示您已佈建資源的目前狀態。您可以存取用來管理部署的詳細資料和歷程記錄。



本章節討論下列主題：

- [vRealize Automation Cloud Assembly 如何運作](#)

## vRealize Automation Cloud Assembly 如何運作

vRealize Automation Cloud Assembly 是一個藍圖開發和部署服務。您和您的團隊使用此服務將機器、應用程式和服務部署至您的雲端廠商資源。

做為 Cloud Assembly 雲端管理員 (通常稱為雲端管理員)，您可以設定佈建基礎結構，並建立將使用者和資源分組的專案。

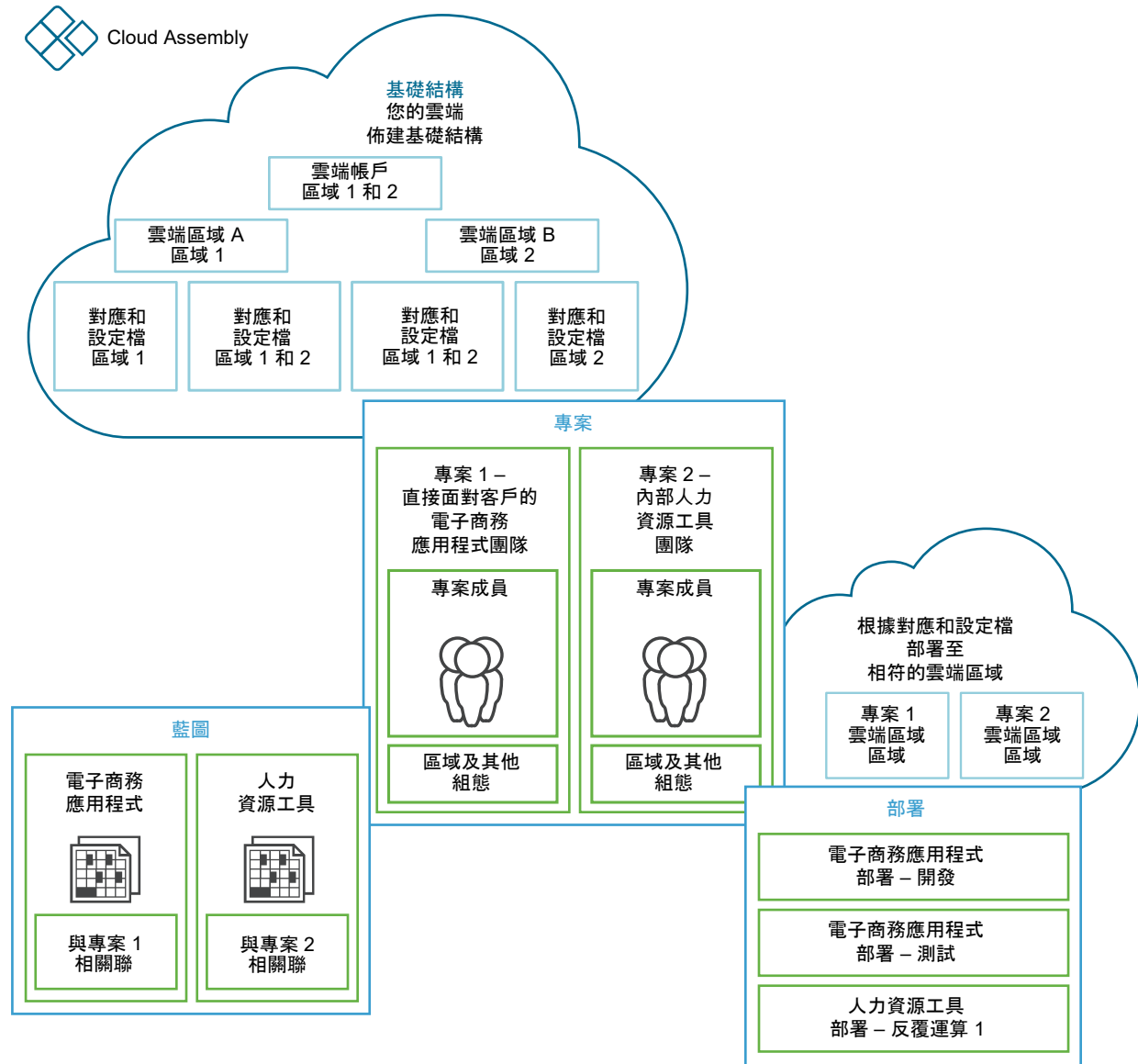
- 新增您的雲端廠商帳戶。請參閱[將雲端帳戶新增至 vRealize Automation Cloud Assembly](#)。

- 判定哪些區域或資料存放區是您想要開發人員部署到的雲端區域。請參閱[進一步瞭解 vRealize Automation Cloud Assembly 雲端區域](#)。
- 建立定義雲端區域的原則。請參閱[第 4 章 建置 vRealize Automation Cloud Assembly 資源基礎結構](#)。
- 建立將開發人員與雲端區域一起分組的專案。請參閱[使用 vRealize Automation Cloud Assembly 專案標籤和自訂內容](#)。

做為藍圖開發人員，您是一或多個專案的成員。建立藍圖並將其部署到與其中一個專案相關聯的雲端區域。

- 使用畫布開發專案的藍圖。您的專案管理員可以使用市集從 VMware Solution Exchange 下載藍圖和支援的映像。請參閱[第 6 章 設計 vRealize Automation Cloud Assembly 部署](#)和 [如何使用 vRealize Automation Cloud Assembly 市集](#)。
- 根據原則和限制，將您的藍圖部署到專案雲端區域。
- 管理您的部署，包括刪除未使用的應用程式。請參閱[第 7 章 管理 vRealize Automation Cloud Assembly 部署](#)。

歡迎使用 vRealize Automation Cloud Assembly。如果您想瞭解如何定義基礎結構，然後再建立並部署藍圖的範例，請參閱 [WordPress 使用案例](#)。



# 為您的組織設定 vRealize Automation Cloud Assembly

## 2

做為 Cloud Assembly 管理員，您必須瞭解使用者角色，並設定與雲端帳戶廠商和整合應用程式的連線。

設定雲端帳戶和整合時，您所設定的是 Cloud Assembly 與目標系統之間的通訊。

本章節討論下列主題：

- 什麼是 vRealize Automation Cloud Assembly 使用者角色
- 將雲端帳戶新增至 vRealize Automation Cloud Assembly
- 將 vRealize Automation 與其他應用程式整合
- vRealize Automation Cloud Assembly 中的上線計劃是什麼
- vRealize Automation Cloud Assembly 環境的進階組態

## 什麼是 vRealize Automation Cloud Assembly 使用者角色

使用者角色決定您可以在 vRealize Automation Cloud Assembly 中看到和執行什麼。一些角色在組織層級定義，而某些角色特定於 vRealize Automation Cloud Assembly。

### 使用者角色

由組織擁有者在 vRealize Automation 主控台中為組織定義使用者角色。有兩種類型的角色：組織角色和服務角色。

組織角色是全域性的，將套用至組織中的所有服務。組織層級的角色是「組織擁有者」或「組織成員」角色。

如需有關組織角色的詳細資訊，請參閱《[管理 vRealize Automation](#)》。

在主控台的組織層級上，還會指派屬於服務專屬權限的 vRealize Automation Cloud Assembly 服務角色。

表 2-1. 服務角色

角色	說明
Cloud Assembly 管理員	必須具有整個使用者介面和 API 資源的讀取和寫入權限。這是唯一可查看和執行所有操作的使用者角色，包括新增雲端帳戶、建立新專案，以及指派專案管理員。
Cloud Assembly 使用者	沒有 Cloud Assembly 管理員角色的任何使用者。 在 vRealize Automation Cloud Assembly 專案中，管理員會將使用者新增至專案做為專案成員。管理員也可以新增專案管理員。這兩種角色的權限定義如下。

## 專案角色和權限

專案角色、專案管理員和專案成員是在 vRealize Automation Cloud Assembly 中定義，可能會因專案而異。

在下列資料表 (其中已定義權限) 中，請記住雲端管理員具有 UI 中所有區域的完整權限。

專案管理員利用雲端管理員所建立的基礎結構，來確保其專案成員具有其開發工作所需的資源。

表 2-2. 專案管理員權限

索引標籤	節點或區域	檢視	建立	修改/刪除
基礎結構	設定 - 專案	是 (僅限您的專案)	否	是 (僅限您的專案)
	設定 - 雲端區域	否	否	否
	設定 - 類型模板對應	是	否	否
	設定 - 映像對應	是	否	否
	設定 - 網路設定檔	是	否	否
	設定 - 儲存區設定檔	是	否	否
	設定 - 標籤	是	否	否
	資源 - 計算	是	否	否
	資源 - 網路	是	否	否
	資源 - 儲存區	是	否	否
	資源 - 機器	是 (僅限您的專案)	是	是 (僅限您的專案)
	資源 - 磁碟區			
	活動 - 要求	是 (僅限您的專案)	N/A	是 (僅限您的專案)
	活動 - 事件	是 (僅限您的專案)	N/A	是 (僅限您的專案)
	連線 - 雲端帳戶	否	否	否
	連線 - 整合		否	否

表 2-2. 專案管理員權限 (續)

索引標籤	節點或區域	檢視	建立	修改/刪除
	連線 - 雲端 Proxy		否	否
	成本 - VMC 評估	是	否	否
	成本 - 私有雲	是	否	否
	上線		否	否
藍圖	藍圖	是 (僅適用於您的專案)	是 (僅適用於您的專案)	是 (僅適用於您的專案)
部署	部署	是 (僅適用於您的專案)	N/A	是 (僅適用於您的專案)

專案成員通常是建立和部署藍圖的開發人員。

表 2-3. 專案成員權限

索引標籤	節點或區域	檢視	建立	修改/刪除
基礎結構	設定 - 專案	是 (僅限您所屬的專案)	否	否
	設定 - 雲端區域	否	否	否
	設定 - 類型模板對應	是	否	否
	設定 - 映像對應	是	否	否
	設定 - 網路設定檔	是	否	否
	設定 - 儲存區設定檔	是	否	否
	設定 - 標籤	是	否	否
	資源 - 計算	是	否	否
	資源 - 網路	是	否	否
	資源 - 儲存區	是	否	否
	資源 - 機器	是 (僅限您部署的)	是	是 (僅限您部署的)
	資源 - 磁碟區			
	活動 - 要求	是 (僅限您部署的)	N/A	是 (僅限您部署的)
	活動 - 事件	是 (僅限您部署的)	N/A	是 (僅限您部署的)
	連線 - 雲端帳戶	否	否	否
	連線 - 整合			
	連線 - 雲端 Proxy			
	成本 - VMC 評估	是	否	否
	成本 - 私有雲	是	否	否

表 2-3. 專案成員權限 (續)

索引標籤	節點或區域	檢視	建立	修改/刪除
	上線			
藍圖	藍圖	是 (僅適用於您的專案)	是 (僅適用於您的專案)	是 (僅適用於您的專案)
部署	部署	是 (僅適用於您的部署，除非與所有專案成員共用專案部署)。	N/A	是 (僅適用於您的部署，除非與所有專案成員共用專案部署，且您有權執行第 2 天動作)。

## 將雲端帳戶新增至 vRealize Automation Cloud Assembly

雲端帳戶是已設定的權限，vRealize Automation Cloud Assembly 使用這些權限從區域或資料中心收集資料，以及將藍圖部署到這些區域。

收集的資料包括您稍後與雲端區域相關聯的區域。

稍後設定雲端區域、對應和設定檔時，您可以選取與其相關聯的雲端帳戶。

做為雲端帳戶管理員，您可以按照團隊成員的運作方式建立專案的雲端帳戶。將從您的雲端帳戶收集網路和安全性、計算資源、儲存區及標籤內容等資源資訊的資料。

**備註** 如果雲端帳戶具有已在區域中部署的關聯機器，則可以使用上線計劃將這些機器上架至 vRealize Automation Cloud Assembly 中進行管理。請參閱 [vRealize Automation Cloud Assembly 中的上線計劃是什麼](#)。

如果移除部署中使用的雲端帳戶，則屬於該部署的資源會變為未受管理。

## 在 vRealize Automation 中使用雲端帳戶所需的認證

若要在 vRealize Automation 中設定和使用雲端帳戶，請確認您具有下列認證。



## 所需的雲端帳戶認證

若要執行以下作業...	您需要...
註冊並登入 vRealize Automation Cloud Assembly	<p>VMware 識別碼。</p> <ul style="list-style-type: none"> <li>■ 使用您的企業電子郵件地址設定 <a href="#">My VMware</a> 帳戶。</li> </ul>
連線到 vRealize Automation Services	<p>HTTPS 連接埠 443 向傳出流量開啟，可透過防火牆存取：</p> <ul style="list-style-type: none"> <li>■ *.vmwareidentity.com</li> <li>■ gaz.csp-vidm-prod.com</li> <li>■ *.vmware.com</li> </ul> <p>如需有關連接埠和通訊協定的詳細資訊，請參閱〈<a href="#">VMware 連接埠和通訊協定</a>〉。</p> <p>如需所需連接埠和通訊協定的相關資訊，請參閱：</p> <ul style="list-style-type: none"> <li>■ 《安裝》說明中的<a href="#">連接埠和通訊協定</a></li> <li>■ 《參考架構》說明中的<a href="#">連接埠需求</a></li> </ul>

若要執行以下作業...	您需要...
新增 Amazon Web Services (AWS) 雲端帳戶	<p>提供具有讀取和寫入權限的進階使用者帳戶。使用者帳戶必須是 AWS 身分識別與存取管理 (IAM) 系統中的電源存取原則 (PowerUserAccess) 的成員。</p> <ul style="list-style-type: none"> <li>■ 20 位數的存取金鑰識別碼及對應的密碼存取金鑰</li> </ul> <p>如果您使用外部 HTTP 網際網路 Proxy，則必須針對 IPv4 進行設定。</p> <p>vRealize Automation 以動作為基礎的擴充性 (ABX) 和外部 IPAM 整合可能需要其他權限。</p> <p>若要允許自動調整功能，建議具備下列 AWS 權限：</p> <ul style="list-style-type: none"> <li>■ 自動調整動作： <ul style="list-style-type: none"> <li>■ autoscaling:DescribeAutoScalingInstances</li> <li>■ autoscaling:AttachInstances</li> <li>■ autoscaling&gt;DeleteLaunchConfiguration</li> <li>■ autoscaling:DescribeAutoScalingGroups</li> <li>■ autoscaling&gt;CreateAutoScalingGroup</li> <li>■ autoscaling:UpdateAutoScalingGroup</li> <li>■ autoscaling&gt;DeleteAutoScalingGroup</li> <li>■ autoscaling:DescribeLoadBalancers</li> </ul> </li> <li>■ 自動調整資源： <ul style="list-style-type: none"> <li>■ *</li> </ul> <p>提供所有自動調整資源權限。</p> </li> </ul> <p>若要允許 AWS Security Token Service (AWS STS) 功能支援用於 AWS 身分識別與存取的暫時有限權限認證，需要具備下列權限：</p> <ul style="list-style-type: none"> <li>■ AWS STS 資源： <ul style="list-style-type: none"> <li>■ *</li> </ul> <p>提供所有 STS 資源權限。</p> </li> </ul> <p>若要允許 EC2 功能，需要具備下列 AWS 權限：</p> <ul style="list-style-type: none"> <li>■ EC2 動作： <ul style="list-style-type: none"> <li>■ ec2:AttachVolume</li> <li>■ ec2:AuthorizeSecurityGroupIngress</li> <li>■ ec2&gt;DeleteSubnet</li> <li>■ ec2&gt;DeleteSnapshot</li> <li>■ ec2:DescribeInstances</li> <li>■ ec2&gt;DeleteTags</li> <li>■ ec2:DescribeRegions</li> <li>■ ec2:DescribeVolumesModifications</li> <li>■ ec2&gt;CreateVpc</li> <li>■ ec2:DescribeSnapshots</li> <li>■ ec2:DescribeInternetGateways</li> <li>■ ec2&gt;DeleteVolume</li> <li>■ ec2:DescribeNetworkInterfaces</li> <li>■ ec2:StartInstances</li> <li>■ ec2:DescribeAvailabilityZones</li> <li>■ ec2&gt;CreateInternetGateway</li> <li>■ ec2&gt;CreateSecurityGroup</li> <li>■ ec2:DescribeVolumes</li> <li>■ ec2&gt;CreateSnapshot</li> </ul> </li> </ul>

**若要執行以下作業...**

**您需要...**

- ec2:ModifyInstanceAttribute
- ec2:DescribeRouteTables
- ec2:DescribeInstanceStatus
- ec2:DetachVolume
- ec2:RebootInstances
- ec2:AuthorizeSecurityGroupEgress
- ec2:ModifyVolume
- ec2:TerminateInstances
- ec2:DescribeSpotFleetRequestHistory
- ec2:DescribeTags
- ec2:CreateTags
- ec2:RunInstances
- ec2:DescribeNatGateways
- ec2:StopInstances
- ec2:DescribeSecurityGroups
- ec2:CreateVolume
- ec2:DescribeSpotFleetRequests
- ec2:DescribeImages
- ec2:DescribeVpcs
- ec2>DeleteSecurityGroup
- ec2>DeleteVpc
- ec2:CreateSubnet
- ec2:DescribeSubnets
- ec2:RequestSpotFleet

**備註** vRealize Automation 以動作為基礎的擴充性 (ABX) 或外部 IPAM 整合不需要 SpotFleet 請求權限。

- EC2 資源：

- \*

提供所有 EC2 資源權限。

若要允許彈性負載平衡功能，需要具備下列 AWS 權限：

- 負載平衡器動作：

- elasticloadbalancing:DeleteLoadBalancer
- elasticloadbalancing:DescribeLoadBalancers
- elasticloadbalancing:RemoveTags
- elasticloadbalancing:CreateLoadBalancer
- elasticloadbalancing:DescribeTags
- elasticloadbalancing:ConfigureHealthCheck
- elasticloadbalancing:AddTags
- elasticloadbalancing:CreateTargetGroup
- elasticloadbalancing>DeleteLoadBalancerListeners
- elasticloadbalancing:DeregisterInstancesFromLoadBalancer
- elasticloadbalancing:RegisterInstancesWithLoadBalancer
- elasticloadbalancing:CreateLoadBalancerListeners

**若要執行以下作業...**

**您需要...**

- 負載平衡器資源：

- \*

提供所有負載平衡器資源權限。

您可以啟用下列 AWS 身分識別與存取管理 (IAM) 權限，但這些權限不是必要權限：

- iam:SimulateCustomPolicy
- iam:GetUser
- iam:ListUserPolicies
- iam:GetUserPolicy
- iam:ListAttachedUserPolicies
- iam:GetPolicyVersion
- iam:ListGroupsForUser
- iam:ListGroupPolicies
- iam:GetGroupPolicy
- iam:ListAttachedGroupPolicies
- iam:ListPolicyVersions

若要執行以下作業...	您需要...
新增 Microsoft Azure 雲端帳戶	<p>設定 Microsoft Azure 執行個體，並取得可使用訂閱識別碼的有效 Microsoft Azure 訂閱。</p> <p>建立 Active Directory 應用程式，如 Microsoft Azure 產品說明文件中的 <a href="#">〈操作方法：使用入口網站建立可存取資源的 Azure AD 應用程式和服務主體〉</a> 所述。</p> <p>如果您使用外部 HTTP 網際網路 Proxy，則必須針對 IPv4 進行設定。</p> <p>請記錄下列資訊：</p> <ul style="list-style-type: none"> <li>■ 訂閱識別碼 <ul style="list-style-type: none"> <li>可讓您存取 Microsoft Azure 訂閱。</li> </ul> </li> <li>■ 承租人識別碼 <ul style="list-style-type: none"> <li>在 Microsoft Azure 帳戶中建立的 Active Directory 應用程式的授權端點。</li> </ul> </li> <li>■ 用戶端應用程式識別碼 <ul style="list-style-type: none"> <li>提供 Microsoft Azure 個別帳戶中 Microsoft Active Directory 的存取權。</li> </ul> </li> <li>■ 用戶端應用程式秘密金鑰 <ul style="list-style-type: none"> <li>與您的用戶端應用程式識別碼配對而產生的唯一秘密金鑰。</li> </ul> </li> </ul> <p>建立和驗證 Microsoft Azure 雲端帳戶需要下列權限：</p> <ul style="list-style-type: none"> <li>■ Microsoft 計算 <ul style="list-style-type: none"> <li>■ Microsoft.Compute/virtualMachines/extensions/write</li> <li>■ Microsoft.Compute/virtualMachines/extensions/read</li> <li>■ Microsoft.Compute/virtualMachines/extensions/delete</li> <li>■ Microsoft.Compute/virtualMachines/deallocate/action</li> <li>■ Microsoft.Compute/virtualMachines/delete</li> <li>■ Microsoft.Compute/virtualMachines/powerOff/action</li> <li>■ Microsoft.Compute/virtualMachines/read</li> <li>■ Microsoft.Compute/virtualMachines/restart/action</li> <li>■ Microsoft.Compute/virtualMachines/start/action</li> <li>■ Microsoft.Compute/virtualMachines/write</li> <li>■ Microsoft.Compute/availabilitySets/write</li> <li>■ Microsoft.Compute/availabilitySets/read</li> <li>■ Microsoft.Compute/availabilitySets/delete</li> <li>■ Microsoft.Compute/disks/delete</li> <li>■ Microsoft.Compute/disks/read</li> <li>■ Microsoft.Compute/disks/write</li> </ul> </li> <li>■ Microsoft 網路 <ul style="list-style-type: none"> <li>■ Microsoft.Network/loadBalancers/backendAddressPools/join/action</li> <li>■ Microsoft.Network/loadBalancers/delete</li> <li>■ Microsoft.Network/loadBalancers/read</li> <li>■ Microsoft.Network/loadBalancers/write</li> <li>■ Microsoft.Network/networkInterfaces/join/action</li> <li>■ Microsoft.Network/networkInterfaces/read</li> <li>■ Microsoft.Network/networkInterfaces/write</li> <li>■ Microsoft.Network/networkInterfaces/delete</li> <li>■ Microsoft.Network/networkSecurityGroups/join/action</li> <li>■ Microsoft.Network/networkSecurityGroups/read</li> <li>■ Microsoft.Network/networkSecurityGroups/write</li> </ul> </li> </ul>

**若要執行以下作業...**

**您需要...**

- Microsoft.Network/networkSecurityGroups/delete
- Microsoft.Network/publicIPAddresses/delete
- Microsoft.Network/publicIPAddresses/join/action
- Microsoft.Network/publicIPAddresses/read
- Microsoft.Network/publicIPAddresses/write
- Microsoft.Network/virtualNetworks/read
- Microsoft.Network/virtualNetworks/subnets/delete
- Microsoft.Network/virtualNetworks/subnets/join/action
- Microsoft.Network/virtualNetworks/subnets/read
- Microsoft.Network/virtualNetworks/subnets/write
- Microsoft.Network/virtualNetworks/write
- Microsoft 資源
  - Microsoft.Resources/subscriptions/resourcegroups/delete
  - Microsoft.Resources/subscriptions/resourcegroups/read
  - Microsoft.Resources/subscriptions/resourcegroups/write
- Microsoft 儲存
  - Microsoft.Storage/storageAccounts/delete
  - Microsoft.Storage/storageAccounts/listKeys/action
  - Microsoft.Storage/storageAccounts/read
  - Microsoft.Storage/storageAccounts/write
- Microsoft Web
  - Microsoft.Web/sites/read
  - Microsoft.Web/sites/write
  - Microsoft.Web/sites/delete
  - Microsoft.Web/sites/config/read
  - Microsoft.Web/sites/config/write
  - Microsoft.Web/sites/config/list/action
  - Microsoft.Web/sites/publishxml/action
  - Microsoft.Web/serverfarms/write
  - Microsoft.Web/serverfarms/delete
  - Microsoft.Web/sites/hostruntime/functions/keys/read
  - Microsoft.Web/sites/hostruntime/host/read
  - Microsoft.web/sites/functions/masterkey/read

如果您要將 Microsoft Azure 與以動作為基礎的擴充性搭配使用，則除了最小權限之外，還需要下列權限：

- Microsoft.Web/sites/read
- Microsoft.Web/sites/write
- Microsoft.Web/sites/delete
- Microsoft.Web/sites/config/read
- Microsoft.Web/sites/config/write
- Microsoft.Web/sites/config/list/action
- Microsoft.Web/sites/publishxml/action
- Microsoft.Web/serverfarms/write
- Microsoft.Web/serverfarms/delete
- Microsoft.Web/sites/hostruntime/functions/keys/read

**若要執行以下作業...**

**您需要...**

- Microsoft.Web/sites/hostruntime/host/read
- Microsoft.Web/sites/functions/masterkey/read

如果您要將 Microsoft Azure 與具有延伸的以動作為基礎的擴充性搭配使用，則還需要下列權限：

- Microsoft.Compute/virtualMachines/extensions/write
  - Microsoft.Compute/virtualMachines/extensions/read
  - Microsoft.Compute/virtualMachines/extensions/delete
-

若要執行以下作業...	您需要...
新增 Google Cloud Platform (GCP) 雲端帳戶	<p>Google Cloud Platform 雲端帳戶與 Google Cloud Platform 計算引擎互動。</p> <p>建立並驗證 Google Cloud Platform 雲端帳戶時需要專案管理員和擁有者認證。</p> <p>如果您使用外部 HTTP 網際網路 Proxy，則必須針對 IPv4 進行設定。</p> <p>必須啟用計算引擎服務。在 vRealize Automation 中建立雲端帳戶時，請使用在初始化計算引擎時所建立的服務帳戶。</p> <p>還需要下列計算引擎權限，具體取決於使用者可以採取的動作：</p> <ul style="list-style-type: none"> <li>■ roles/compute.admin                     <p>提供所有計算引擎資源的完整控制權。</p> </li> <li>■ roles/iam.serviceAccountUser                     <p>為管理設定為做為服務帳戶執行的虛擬機器執行個體的使用者提供存取權。授與下列資源和服務的存取權：</p> <ul style="list-style-type: none"> <li>■ compute.*</li> <li>■ resourceManager.projects.get</li> <li>■ resourceManager.projects.list</li> <li>■ serviceUsage.quotas.get</li> <li>■ serviceUsage.services.get</li> <li>■ serviceUsage.services.list</li> </ul> </li> <li>■ roles/compute.imageUser                     <p>提供列出和讀取映像的權限，沒有映像的其他權限。在專案層級授與 compute.imageUser 角色，可讓使用者列出專案中的所有映像。還可以讓使用者根據專案中的映像建立資源，例如執行個體和持續性磁碟。</p> <ul style="list-style-type: none"> <li>■ compute.images.get</li> <li>■ compute.images.getFromFamily</li> <li>■ compute.images.list</li> <li>■ compute.images.useReadOnly</li> <li>■ resourceManager.projects.get</li> <li>■ resourceManager.projects.list</li> <li>■ serviceUsage.quotas.get</li> <li>■ serviceUsage.services.get</li> <li>■ serviceUsage.services.list</li> </ul> </li> <li>■ roles/compute.instanceAdmin                     <p>提供建立、修改和刪除虛擬機器執行個體的權限。這包括建立、修改和刪除磁碟以及設定受防護的 VMBETA 設定的權限。</p> <p>對於管理虛擬機器執行個體 (但不是網路或安全性設定，或做為服務帳戶執行的執行個體) 的使用者，將此角色授與包含執行個體的組織、資料夾或專案，或授與個別執行個體。</p> <p>管理設定為做為服務帳戶執行的虛擬機器執行個體的使用者還需要 roles/iam.serviceAccountUser 角色。</p> <ul style="list-style-type: none"> <li>■ compute.acceleratorTypes</li> <li>■ compute.addresses.get</li> <li>■ compute.addresses.list</li> <li>■ compute.addresses.use</li> <li>■ compute.autoscalers</li> <li>■ compute.diskTypes</li> <li>■ compute.disks.create</li> <li>■ compute.disks.createSnapshot</li> <li>■ compute.disks.delete</li> </ul> </li> </ul>



**若要執行以下作業...**

**您需要...**

- compute.disks.get
- compute.disks.list
- compute.disks.resize
- compute.disks.setLabels
- compute.disks.update
- compute.disks.use
- compute.disks.useReadOnly
- compute.globalAddresses.get
- compute.globalAddresses.list
- compute.globalAddresses.use
- compute.globalOperations.get
- compute.globalOperations.list
- compute.images.get
- compute.images.getFromFamily
- compute.images.list
- compute.images.useReadOnly
- compute.instanceGroupManagers
- compute.instanceGroups
- compute.instanceTemplates
- compute.instances
- compute.licenses.get
- compute.licenses.list
- compute.machineTypes
- compute.networkEndpointGroups
- compute.networks.get
- compute.networks.list
- compute.networks.use
- compute.networks.useExternallp
- compute.projects.get
- compute.regionOperations.get
- compute.regionOperations.list
- compute.regions
- compute.reservations.get
- compute.reservations.list
- compute.subnetworks.get
- compute.subnetworks.list
- compute.subnetworks.use
- compute.subnetworks.useExternallp
- compute.targetPools.get
- compute.targetPools.list
- compute.zoneOperations.get
- compute.zoneOperations.list
- compute.zones
- resourcemanager.projects.get

**若要執行以下作業...**

**您需要...**

- resourceManager.projects.list
- serviceusage.quotas.get
- serviceusage.services.get
- serviceusage.services.list
- roles/compute.instanceAdmin.v1

提供計算引擎執行個體、執行個體群組、磁碟、快照和映像的完整控制權。還提供對所有計算引擎網路資源的讀取權限。

**備註** 如果您在執行個體層級授與使用者此角色，則該使用者無法建立新的執行個體。

- compute.acceleratorTypes
- compute.addresses.get
- compute.addresses.list
- compute.addresses.use
- compute.autoscalers
- compute.backendBuckets.get
- compute.backendBuckets.list
- compute.backendServices.get
- compute.backendServices.list
- compute.diskTypes
- compute.disks
- compute.firewalls.get
- compute.firewalls.list
- compute.forwardingRules.get
- compute.forwardingRules.list
- compute.globalAddresses.get
- compute.globalAddresses.list
- compute.globalAddresses.use
- compute.globalForwardingRules.get
- compute.globalForwardingRules.list
- compute.globalOperations.get
- compute.globalOperations.list
- compute.healthChecks.get
- compute.healthChecks.list
- compute.httpHealthChecks.get
- compute.httpHealthChecks.list
- compute.httpsHealthChecks.get
- compute.httpsHealthChecks.list
- compute.images
- compute.instanceGroupManagers
- compute.instanceGroups
- compute.instanceTemplates
- compute.instances
- compute.interconnectAttachments.get
- compute.interconnectAttachments.list
- compute.interconnectLocations

**若要執行以下作業...**

**您需要...**

- compute.interconnects.get
- compute.interconnects.list
- compute.licenseCodes
- compute.licenses
- compute.machineTypes
- compute.networkEndpointGroups
- compute.networks.get
- compute.networks.list
- compute.networks.use
- compute.networks.useExternalIp
- compute.projects.get
- compute.projects.setCommonInstanceMetadata
- compute.regionBackendServices.get
- compute.regionBackendServices.list
- compute.regionOperations.get
- compute.regionOperations.list
- compute.regions
- compute.reservations.get
- compute.reservations.list
- compute.resourcePolicies
- compute.routers.get
- compute.routers.list
- compute.routes.get
- compute.routes.list
- compute.snapshots
- compute.sslCertificates.get
- compute.sslCertificates.list
- compute.sslPolicies.get
- compute.sslPolicies.list
- compute.sslPolicies.listAvailableFeatures
- compute.subnetworks.get
- compute.subnetworks.list
- compute.subnetworks.use
- compute.subnetworks.useExternalIp
- compute.targetHttpProxies.get
- compute.targetHttpProxies.list
- compute.targetHttpsProxies.get
- compute.targetHttpsProxies.list
- compute.targetInstances.get
- compute.targetInstances.list
- compute.targetPools.get
- compute.targetPools.list
- compute.targetSslProxies.get
- compute.targetSslProxies.list

若要執行以下作業...	您需要...
	<ul style="list-style-type: none"> <li>■ compute.targetTcpProxies.get</li> <li>■ compute.targetTcpProxies.list</li> <li>■ compute.targetVpnGateways.get</li> <li>■ compute.targetVpnGateways.list</li> <li>■ compute.urlMaps.get</li> <li>■ compute.urlMaps.list</li> <li>■ compute.vpnTunnels.get</li> <li>■ compute.vpnTunnels.list</li> <li>■ compute.zoneOperations.get</li> <li>■ compute.zoneOperations.list</li> <li>■ compute.zones</li> <li>■ resourceManager.projects.get</li> <li>■ resourceManager.projects.list</li> <li>■ serviceusage.quotas.get</li> <li>■ serviceusage.services.get</li> <li>■ serviceusage.services.list</li> </ul>
新增 NSX-T 雲端帳戶	<p>提供具有下列讀取和寫入權限的帳戶：</p> <ul style="list-style-type: none"> <li>■ NSX-T 企業管理員角色和存取認證</li> <li>■ NSX-T IP 位址或 FQDN</li> </ul> <p>管理員還需要存取 vCenter Server，如此頁面上以下〈基於 vCenter 的雲端帳戶的 vSphere 代理程式需求〉一節中所述。</p>
新增 NSX-V 雲端帳戶	<p>提供具有下列讀取和寫入權限的帳戶：</p> <ul style="list-style-type: none"> <li>■ NSX-V 企業管理員角色和存取認證</li> <li>■ NSX-V IP 位址或 FQDN</li> </ul> <p>管理員還需要存取 vCenter Server，如此頁面上以下〈基於 vCenter 的雲端帳戶的 vSphere 代理程式需求〉一節中所述。</p>
新增 vCenter 雲端帳戶	<p>提供具有下列讀取和寫入權限的帳戶：</p> <ul style="list-style-type: none"> <li>■ vCenter IP 位址或 FQDN</li> </ul> <p>管理員還需要存取 vCenter Server，如此頁面上以下〈基於 vCenter 的雲端帳戶的 vSphere 代理程式需求〉一節中所述。</p>
新增 VMware Cloud on AWS (VMC) 雲端帳戶	<p>提供具有下列讀取和寫入權限的帳戶：</p> <ul style="list-style-type: none"> <li>■ cloudadmin@vmc.local 帳戶或 CloudAdmin 群組中的任何使用者帳戶</li> <li>■ NSX 企業管理員角色和存取認證</li> <li>■ 對您組織的 VMware Cloud on AWS SDDC 環境的 NSX Cloud 管理員存取權</li> <li>■ 對您組織的 VMware Cloud on AWS SDDC 環境的管理員存取權</li> <li>■ 在您組織的 VMware Cloud on AWS 服務中適用於 VMware Cloud on AWS 環境的 VMware Cloud on AWS API Token</li> <li>■ vCenter IP 位址或 FQDN</li> </ul> <p>管理員還需要對您的目標 VMware Cloud on AWS SDDC 使用的 vCenter 的存取權，它具有本頁面上以下〈基於 vSphere 的雲端帳戶的 vCenter 代理程式需求〉一節中列出的所有權限。</p> <p>如需有關建立和使用 VMware Cloud on AWS 雲端帳戶所需權限的詳細資訊，請參閱 VMware Cloud on AWS <a href="#">產品說明文件</a> 中的《管理 VMware Cloud on AWS 資料中心》。</p>

## 基於 vCenter 的雲端帳戶的 vSphere 代理程式需求

下表列出了管理 VMware Cloud on AWS 和 vCenter 雲端帳戶所需的權限。必須針對 vCenter Server 中的所有叢集啟用這些權限，而非僅針對主控端點的叢集啟用。

對於所有以 vCenter Server 為基礎的雲端帳戶 (包括 NSX-V、NSX-T、vCenter 和 VMware Cloud on AWS)，管理員必須具有 vSphere 端點認證或用以在 vCenter 中執行代理程式服務的認證，以提供對主機 vCenter Server 的管理存取權。

如需有關 vSphere 代理程式需求的詳細資訊，請參閱 [VMware vSphere 產品說明文件](#)。

表 2-4. vSphere 代理程式管理 vCenter Server 執行個體所需的權限

屬性值	權限
資料存放區	<ul style="list-style-type: none"> <li>■ 配置空間</li> <li>■ 瀏覽資料存放區</li> <li>■ 低層級檔案作業</li> </ul>
資料存放區叢集	設定資料存放區叢集
資料夾	<ul style="list-style-type: none"> <li>■ 建立資料夾</li> <li>■ 刪除資料夾</li> </ul>
全域	<ul style="list-style-type: none"> <li>■ 管理自訂屬性</li> <li>■ 設定自訂屬性</li> </ul>
網路	指派網路
權限	修改權限
資源	<ul style="list-style-type: none"> <li>■ 將虛擬機器指派給資源集區</li> <li>■ 移轉已關閉電源的虛擬機器</li> <li>■ 移轉已開啟電源的虛擬機器</li> </ul>

表 2-4. vSphere 代理程式管理 vCenter Server 執行個體所需的權限 (續)

屬性值	權限
內容程式庫	<p>若要指派內容程式庫的權限，管理員必須將該權限做為全域權限授與使用者。如需相關資訊，請參閱 <a href="#">VMware vSphere 說明文件</a> 之《vSphere 虛擬機器管理》中的〈<a href="#">內容程式庫權限的階層式繼承</a>〉。</p> <ul style="list-style-type: none"> <li>■ 新增程式庫項目</li> <li>■ 建立本機程式庫</li> <li>■ 建立已訂閱程式庫</li> <li>■ 刪除程式庫項目</li> <li>■ 刪除本機程式庫</li> <li>■ 刪除已訂閱程式庫</li> <li>■ 下載檔案</li> <li>■ 收回程式庫項目</li> <li>■ 收回已訂閱程式庫</li> <li>■ 探查訂閱資訊</li> <li>■ 讀取儲存區</li> <li>■ 同步程式庫項目</li> <li>■ 同步已訂閱程式庫</li> <li>■ 類型自我檢查</li> <li>■ 更新組態設定</li> <li>■ 更新檔案</li> <li>■ 更新程式庫</li> <li>■ 更新程式庫項目</li> <li>■ 更新本機程式庫</li> <li>■ 更新已訂閱程式庫</li> <li>■ 檢視組態設定</li> </ul>
標籤	<ul style="list-style-type: none"> <li>■ 指派或取消指派 vSphere 標籤</li> <li>■ 建立 vSphere 標籤</li> <li>■ 建立 vSphere 標籤類別</li> <li>■ 刪除 vSphere 標籤</li> <li>■ 刪除 vSphere 標籤類別</li> <li>■ 編輯 vSphere 標籤</li> <li>■ 編輯 vSphere 標籤類別</li> <li>■ 修改類別的 [UsedBy] 欄位</li> <li>■ 修改標籤的 [UsedBy] 欄位</li> </ul>
vApp	<ul style="list-style-type: none"> <li>■ 匯入</li> <li>■ vApp 應用程式組態</li> </ul> <p>對於 OVF 範本以及從內容程式庫佈建虛擬機器時，需要 <code>vApp.Import</code> 應用程式組態。</p> <p>針對雲端組態指令碼使用 <code>cloud-init</code> 時，需要 <code>vApp.vApp</code> 應用程式組態。此設定允許修改 vApp 的內部結構，如產品資訊和內容。</p>
虛擬機器 - 詳細目錄	<ul style="list-style-type: none"> <li>■ 從現有項目建立</li> <li>■ 建立新項目</li> <li>■ 移動</li> <li>■ 移除</li> </ul>

表 2-4. vSphere 代理程式管理 vCenter Server 執行個體所需的權限 (續)

屬性值	權限
虛擬機器 - 互動	<ul style="list-style-type: none"> <li>■ 設定 CD 媒體</li> <li>■ 主控台互動</li> <li>■ 裝置連線</li> <li>■ 關閉電源</li> <li>■ 開啟電源</li> <li>■ 重設</li> <li>■ 暫停</li> <li>■ 工具安裝</li> </ul>
虛擬機器 - 組態	<ul style="list-style-type: none"> <li>■ 新增現有磁碟</li> <li>■ 新增磁碟</li> <li>■ 移除磁碟</li> <li>■ 進階</li> <li>■ 變更 CPU 計數</li> <li>■ 變更資源</li> <li>■ 延伸虛擬磁碟</li> <li>■ 磁碟變更追蹤</li> <li>■ 記憶體</li> <li>■ 修改裝置設定</li> <li>■ 重新命名</li> <li>■ 設定註解</li> <li>■ 設定</li> <li>■ 分頁檔放置</li> </ul>
虛擬機器 - 佈建	<ul style="list-style-type: none"> <li>■ 自訂</li> <li>■ 複製範本</li> <li>■ 複製虛擬機器</li> <li>■ 部署範本</li> <li>■ 讀取自訂規格</li> </ul>
虛擬機器 - 狀態	<ul style="list-style-type: none"> <li>■ 建立快照</li> <li>■ 移除快照</li> <li>■ 還原為快照</li> </ul>

## 設定 Microsoft Azure 與 vRealize Automation Cloud Assembly 搭配使用

您必須收集一些資訊並執行某些組態，才能在 vRealize Automation Cloud Assembly 中建立 Microsoft Azure 雲端帳戶。

### 程序

- 1 找到並記錄 Microsoft Azure 訂閱和承租人識別碼。
  - 訂閱識別碼 - 按一下 Azure 入口網站左側工具列中的 [訂閱] 圖示，以檢視訂閱識別碼。
  - 承租人識別碼 - 按一下 [說明] 圖示，然後在 Azure 入口網站中選取 [顯示診斷]。搜尋承租人，並在找到時記錄其識別碼。

- 2 您可以建立新的儲存區帳戶和資源群組以開始。也可以稍後在藍圖中建立儲存區帳戶和資源群組。

■ 儲存區帳戶 - 請使用下列程序設定帳戶。

- 1 在 Azure 入口網站中，找到側邊列上的 [儲存區帳戶] 圖示。請確定已選取正確的訂閱，然後按一下**新增**。您也可以在 Azure 搜尋欄位中搜尋儲存區帳戶。
- 2 輸入儲存區帳戶所需的資訊。需要提供訂閱識別碼。
- 3 選取是要使用現有資源群組還是建立一個新的資源群組。記下資源群組名稱，因為稍後需要用到它。

---

**備註** 儲存儲存區帳戶的位置，因為稍後需要用到它。

---

- 3 建立虛擬網路。或者，如果您有適當的現有網路，則可以選取該網路。

如果您要建立網路，必須選取 [使用現有的資源群組]，並指定您在上一個步驟中建立的群組。此外，請選取先前指定的相同位置。如果物件將使用的所有適用元件之間的位置不相符，Microsoft Azure 將不會部署虛擬機器或其他物件。

- a 在左面板中找到 [虛擬網路] 圖示，然後按一下該圖示或搜尋虛擬網路。請確保選取正確的訂閱，然後按一下**新增**。
- b 為新的虛擬網路輸入唯一名稱，並將其記錄以供稍後使用。
- c 在**位址空間**欄位中，為虛擬網路輸入適當的 IP 位址。
- d 請確定已選取正確的訂閱，然後按一下**新增**。
- e 輸入剩餘的基本組態資訊。
- f 您可以視需要修改其他選項，但對於大多數組態，您可以保留預設值。
- g 按一下**建立**。

- 4 設定 Azure Active Directory 應用程式，以便 vRA 可進行驗證。

- a 在 Azure 左側功能表中找到 Active Directory 圖示，然後按一下該圖示。
- b 按一下**應用程式登錄**，然後選取**新增**。
- c 針對應用程式輸入符合 Azure 名稱驗證的名稱。
- d 保留 [Web 應用程式/API] 做為應用程式類型。
- e 登入 URL 可以是適合您使用的任何內容。
- f 按一下**建立**。

- 5 建立秘密金鑰以驗證 Cloud Assembly 中的應用程式。

- a 按一下 Azure 中的應用程式的名稱。  
記下應用程式識別碼以供稍後使用。
- b 在下一個窗格中按一下**所有設定**，然後從 [設定] 清單中選取 [金鑰]。
- c 輸入新金鑰的說明，然後選擇持續時間。



- d 按一下**儲存**並確保將此金鑰值複製到安全的位置，因為您稍後將無法擷取該值。
  - e 在左側功能表中，針對應用程式選取 **API 權限**，然後按一下**新增權限**以建立新權限。
  - f 在 [選取 API] 頁面上選取 [Azure 服務管理]。
  - g 按一下**委派的權限**。
  - h 在 [選取權限] 下選取 user\_impersonation，然後按一下**新增權限**。
- 6 授權 Active Directory 應用程式連線至 Azure 訂閱，以便您可以部署和管理虛擬機器。
- a 在左側功能表中，按一下 [訂閱] 圖示，然後選取您的新訂閱。  
您可能需要按一下名稱的文字，才能使面板滑過。
  - b 選取 [存取控制 (IAM)] 選項，以查看訂閱的權限。
  - c 按一下 [新增角色指派] 標題下的**新增**。
  - d 從 [角色] 下拉式功能表中選擇 [參與者]。
  - e 在 [指派存取權至] 下拉式功能表中保留預設選擇。
  - f 在 [選取] 方塊中輸入應用程式的名稱。
  - g 按一下**儲存**。
  - h 新增其他角色，使新應用程式具有「擁有者」、「參與者」和「讀者」角色。
  - i 按一下**儲存**。

#### 後續步驟

您必須安裝 Microsoft Azure 命令列介面工具。這些工具可供 Windows 和 Mac 作業系統隨意使用。如需有關下載和安裝這些工具的詳細資訊，請參閱 Microsoft 說明文件。

安裝命令列介面後，您必須對新訂閱進行驗證。

- 1 開啟終端機視窗，然後輸入 Microsoft Azure 登入資訊。您將會收到一個 URL 和允許您進行驗證的簡短程式碼。
- 2 在瀏覽器中，輸入您從裝置上的應用程式接收的程式碼。
- 3 輸入驗證碼，然後按一下**繼續**。
- 4 選取 Azure 帳戶並登入。

如果您有多個訂閱，請確保使用 `azure account set <subscription-name>` 命令選取正確的訂閱。

- 5 繼續之前，您必須使用 `azure provider register microsoft.compute` 命令將 Microsoft.Compute 提供者註冊至新的 Azure 訂閱。

如果命令在您第一次執行時逾時並產生錯誤，請重新執行。

當您完成設定後，您可以使用 `azure vm image list` 命令擷取可用的虛擬機器映像名稱。您可以選擇所需的映像，並記錄為其提供的 URN 以便稍後在藍圖中使用。

## 在 vRealize Automation 中建立 Microsoft Azure 雲端帳戶

做為雲端管理員，您可以為團隊將要部署 vRealize Automation 藍圖的帳戶區域建立 Microsoft Azure 雲端帳戶。

若要檢視 Microsoft Azure 雲端帳戶在 vRealize Automation 中的運作方式的使用案例範例，請參閱 [WordPress 使用案例](#)。

### 必要條件

- 確認您擁有必要的管理員認證，並且已在連接埠 443 上啟用 HTTPS 存取。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有必要的使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 設定 Microsoft Azure 帳戶與 vRealize Automation 搭配使用。請參閱[設定 Microsoft Azure 與 vRealize Automation Cloud Assembly 搭配使用](#)。
- 如果沒有外部網際網路存取權，則設定網際網路伺服器 Proxy。請參閱[如何為 vRealize Automation 設定網際網路 Proxy 伺服器](#)。

### 程序

- 1 選取**基礎結構 > 連線 > 雲端帳戶**，然後按一下**新增雲端帳戶**。
- 2 選取 Microsoft Azure 帳戶類型，然後輸入認證和其他值。
- 3 按一下**驗證**。  
會收集與該帳戶相關聯的帳戶區域。
- 4 選取您想要佈建此資源的區域。
- 5 若要提高效率，請按一下**為所選區域建立雲端區域**。
- 6 如果需要新增標籤以支援標記策略，請輸入功能標籤。請參閱[如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署](#)和 [建立標記策略](#)。
- 7 按一下**儲存**。

### 結果

此帳戶將新增至 vRealize Automation，並且選取的區域可用於指定的雲端區域。

### 後續步驟

為此雲端帳戶建立基礎結構資源。

## 在 vRealize Automation 中建立 Amazon Web Services 雲端帳戶

做為雲端管理員，您可以為團隊將要部署 vRealize Automation 藍圖的帳戶區域建立 Amazon Web Services (AWS) 雲端帳戶。

### 必要條件

- 確認您擁有必要的管理員認證，並且已在連接埠 443 上啟用 HTTPS 存取。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有必要的使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 確認您具有所需的 AWS 管理員認證。
- 如果沒有外部網際網路存取權，則設定網際網路伺服器 Proxy。請參閱[如何為 vRealize Automation 設定網際網路 Proxy 伺服器](#)。

### 程序

- 1 選取**基礎結構 > 連線 > 雲端帳戶**，然後按一下**新增雲端帳戶**。
- 2 選取 AWS 帳戶類型，然後輸入認證和其他值。
- 3 按一下**驗證**。  
會收集與該帳戶相關聯的帳戶區域。
- 4 選取您想要佈建此資源的區域。
- 5 若要提高效率，請按一下**為所選區域建立雲端區域**。
- 6 如果需要新增標籤以支援標記策略，請輸入功能標籤。請參閱[如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署](#)和 [建立標記策略](#)。
- 7 按一下**新增**。

### 結果

此帳戶將新增至 vRealize Automation，並且選取的區域可用於指定的雲端區域。

### 後續步驟

為此雲端帳戶設定基礎結構資源。

## 在 vRealize Automation 中建立 Google Cloud Platform 雲端帳戶

做為雲端管理員，您可以為團隊將要部署 vRealize Automation 藍圖的帳戶區域建立 Google Cloud Platform (GCP) 雲端帳戶。

### 必要條件

- 確認您擁有必要的管理員認證，並且已在連接埠 443 上啟用 HTTPS 存取。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有必要的使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 確認您有權存取 Google Cloud Platform JSON 安全性金鑰。
- 確認您具有 Google Cloud Platform 執行個體所需的安全性資訊。這些資訊大多數可從您的執行個體或 Google 說明文件取得。

- 如果沒有外部網際網路存取權，則設定網際網路伺服器 Proxy。請參閱[如何為 vRealize Automation 設定網際網路 Proxy 伺服器](#)。

#### 程序

- 1 選取**基礎結構 > 連線 > 雲端帳戶**，然後按一下**新增雲端帳戶**。
- 2 選取 Google Cloud Platform 帳戶類型，然後輸入適當的認證和相關資訊。使用在來源 GCP 帳戶計算引擎初始化時建立的服務帳戶。

如上面的〈**必要條件**〉一節所述，可從在 [vRealize Automation 中使用雲端帳戶所需的認證](#) 取得認證需求。若要在 vRealize Automation 中成功建立雲端帳戶，來源 GCP 帳戶必須已啟用計算引擎服務。

在 vRealize Automation 中，專案識別碼是 Google Cloud Platform 端點的一部分。您可以在建立雲端帳戶時加以指定。對專案特定的私人映像進行資料收集期間，vRealize Automation GCP 介面卡會查詢 Google Cloud Platform API。

- 3 按一下**驗證**。  
會收集與該帳戶相關聯的帳戶區域。
- 4 選取您想要佈建此資源的區域。
- 5 若要提高效率，請按一下**為所選區域建立雲端區域**。
- 6 如果需要標籤以支援標記策略，請輸入功能標籤。請參閱[如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署](#)和 [建立標記策略](#)。
- 7 按一下**新增**。

#### 結果

此帳戶將新增至 vRealize Automation，並且選取的區域可用於指定的雲端區域。

#### 後續步驟

為此雲端帳戶建立基礎結構資源。

## 在 vRealize Automation Cloud Assembly 中建立 vCenter 雲端帳戶

您可以針對要部署 vRealize Automation Cloud Assembly 藍圖的帳戶區域新增 vCenter 雲端帳戶。

出於網路和安全性目的，您可以將 NSX-T 或 NSX-V 雲端帳戶與 vCenter 雲端帳戶相關聯。

#### 必要條件

- 確認您擁有必要的管理員認證，並且已在連接埠 443 上啟用 HTTPS 存取。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。

- 確認您已正確設定連接埠和通訊協定，以支援雲端帳戶。請參閱 [vRealize Automation 產品說明文件](#) 中的以下內容：《使用 vRealize Easy Installer 安裝 vRealize Automation》中的〈vRealize Automation 的連接埠和通訊協定〉主題和《vRealize Automation 參考架構指南》中的〈連接埠需求〉主題。

#### 程序

- 1 選取**基礎結構 > 連線 > 雲端帳戶**，然後按一下**新增雲端帳戶**。
- 2 選取 vCenter 帳戶類型，然後輸入 vCenter Server 主機 IP 位址。
- 3 輸入您的 vCenter Server 管理員認證，然後按一下**驗證**。  
會收集與該帳戶相關聯的資料中心。
- 4 在指定的 vCenter Server 上至少選取一個可用的資料中心，以允許佈建此雲端帳戶。
- 5 若要提高效率，請建立雲端區域以佈建到所選的資料中心。  
也可以根據組織的雲端策略在單獨的步驟中建立雲端區域。  
如需雲端區域的相關資訊，請參閱[進一步瞭解 vRealize Automation Cloud Assembly 雲端區域](#)。
- 6 選取現有的 NSX 雲端帳戶。  
您可以現在或稍後編輯雲端帳戶時選取 NSX 帳戶。  
如需 NSX-V 雲端帳戶的相關資訊，請參閱在 [vRealize Automation Cloud Assembly 中建立 NSX-V 雲端帳戶](#)。  
如需 NSX-T 雲端帳戶的相關資訊，請參閱在 [vRealize Automation Cloud Assembly 中建立 NSX-T 雲端帳戶](#)。
- 7 如果要新增標籤以支援標記策略，請輸入功能標籤。  
您可以現在或稍後編輯雲端帳戶時新增標籤。如需標記的相關資訊，請參閱[如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署](#)。
- 8 按一下**儲存**。

#### 結果

將新增雲端帳戶，並且選取的資料中心可用於指定的雲端區域。收集的資料 (例如機器和磁碟區) 會列在 [基礎結構] 索引標籤的 [資源] 區段中。

#### 後續步驟

為此雲端帳戶設定剩餘的基礎結構資源。請參閱第 4 章 [建置 vRealize Automation Cloud Assembly 資源基礎結構](#)。

## 在 vRealize Automation Cloud Assembly 中建立 NSX-V 雲端帳戶

出於網路和安全性目的，您可以建立 NSX-V 雲端帳戶並將其與 vCenter 雲端帳戶相關聯。

**必要條件**

- 確認您擁有必要的管理員認證，並且已在連接埠 443 上啟用 HTTPS 存取。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 確認您具有可與此 NSX 雲端帳戶搭配使用的 vCenter 雲端帳戶。請參閱在 [vRealize Automation Cloud Assembly 中建立 vCenter 雲端帳戶](#)。
- 確認您已正確設定連接埠和通訊協定，以支援雲端帳戶。請參閱 [vRealize Automation 產品說明文件](#) 中的以下內容：《使用 vRealize Easy Installer 安裝 vRealize Automation》中的〈vRealize Automation 的連接埠和通訊協定〉主題和《vRealize Automation 參考架構指南》中的〈連接埠需求〉主題。

**程序**

- 1 選取**基礎結構 > 連線 > 雲端帳戶**，然後按一下**新增雲端帳戶**。
- 2 選取 NSX-V 帳戶類型，然後輸入 NSX-V 主機 IP 位址。
- 3 輸入您的 NSX 管理員認證，然後按一下**驗證**。  
會收集與此帳戶相關聯的資產。  
如果 NSX 主機 IP 位址無法使用，則驗證失敗。
- 4 如果可用，請選取表示要與此 NSX-V 帳戶相關聯的 vCenter 雲端帳戶的 vCenter 端點。
- 5 如果要新增標籤以支援標記策略，請輸入功能標籤。  
您可以稍後新增或移除功能標籤。請參閱[如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署](#)。
- 6 按一下**儲存**。

**後續步驟**

可以建立或編輯 vCenter 雲端帳戶，並將其與此 NSX 雲端帳戶相關聯。請參閱在 [vRealize Automation Cloud Assembly 中建立 vCenter 雲端帳戶](#)。

建立並設定一或多個雲端區域，以便與此雲端帳戶使用的資料中心搭配使用。請參閱[進一步瞭解 vRealize Automation Cloud Assembly 雲端區域](#)。

為此雲端帳戶設定基礎結構資源。請參閱第 4 章 [建置 vRealize Automation Cloud Assembly 資源基礎結構](#)。

**在 vRealize Automation Cloud Assembly 中建立 NSX-T 雲端帳戶**

出於網路和安全性目的，您可以建立 NSX-T 雲端帳戶並將其與 vCenter 雲端帳戶相關聯。

為了促進部署中的容錯和高可用性，每個 NSX-T 資料中心端點均代表由三個 NSX Manager 組成的叢集。

- vRealize Automation 會指向其中一個 NSX Manager。透過使用此選項，一個 NSX Manager 會從 vRealize Automation 接收 API 呼叫。

- vRealize Automation 會指向叢集的虛擬 IP。透過使用此選項，一個 NSX Manager 便會承擔 VIP 的控制權。該 Manager 會從 vRealize Automation 接收 API 呼叫。如果發生故障，叢集中的另一個節點會承擔 VIP 的控制權，並從 vRealize Automation 接收 API 呼叫。

如需有關 VIP 組態的詳細資訊，請參閱 [VMware NSX-T Data Center 說明文件](#) 之《NSX-T Data Center 安裝指南》中的〈設定叢集的虛擬 IP (VIP) 位址〉。

- vRealize Automation 會指向負載平衡器 VIP，以將呼叫負載平衡到三個 NSX Manager。透過使用此選項，所有三個 NSX Manager 都將從 vRealize Automation 接收 API 呼叫。

您可以在第三方負載平衡器或 NSX-T 負載平衡器上設定 VIP。

對於大型環境，請考慮使用此選項，以分割三個 NSX Manager 之間的 vRealize Automation API 呼叫。

#### 必要條件

- 確認您擁有必要的管理員認證，並且已在連接埠 443 上啟用 HTTPS 存取。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 確認您具有可與此 NSX 雲端帳戶搭配使用的 vCenter 雲端帳戶。請參閱在 [vRealize Automation Cloud Assembly 中建立 vCenter 雲端帳戶](#)。
- 確認您已正確設定連接埠和通訊協定，以支援雲端帳戶。請參閱 [vRealize Automation 產品說明文件](#) 中的以下內容：《使用 vRealize Easy Installer 安裝 vRealize Automation》中的〈vRealize Automation 的連接埠和通訊協定〉主題和《vRealize Automation 參考架構指南》中的〈連接埠需求〉主題。

#### 程序

- 1 選取**基礎結構 > 連線 > 雲端帳戶**，然後按一下**新增雲端帳戶**。
- 2 選取 NSX-T 帳戶類型，然後輸入 NSX-T 端點管理程式執行個體或 VIP 的主機 IP 位址 (請參閱以上內容)。
- 3 輸入您的 NSX 管理員認證，然後按一下**驗證**。  
會收集與此帳戶相關聯的資產。  
如果 NSX 主機 IP 位址無法使用，則驗證失敗。
- 4 (如果可用) 選取代表要與此 NSX-T 雲端帳戶相關聯的 vCenter 雲端帳戶的 vCenter 端點。
- 5 如果要新增標籤以支援標記策略，請輸入功能標籤。  
您可以稍後新增或移除功能標籤。請參閱[如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署](#)。
- 6 按一下**儲存**。



## 後續步驟

您可以建立或編輯 vCenter 雲端帳戶，以便與此 NSX 雲端帳戶相關聯。請參閱在 [vRealize Automation Cloud Assembly 中建立 vCenter 雲端帳戶](#)。

建立並設定一或多個雲端區域，以便與此雲端帳戶使用的資料中心搭配使用。請參閱[進一步瞭解 vRealize Automation Cloud Assembly 雲端區域](#)。

為此雲端帳戶設定基礎結構資源。請參閱第 4 章 [建置 vRealize Automation Cloud Assembly 資源基礎結構](#)。

## 在 vRealize Automation 中建立 VMware Cloud on AWS 雲端帳戶

做為雲端管理員，您可以為團隊將要部署 vRealize Automation 藍圖的帳戶區域建立 VMware Cloud on AWS 雲端帳戶。

在 vRealize Automation 中，VMware Cloud on AWS 需要一些獨特的組態程序。若要針對 VMware Cloud on AWS 正確設定 vRealize Automation，包括為雲端帳戶設定 API Token 值，以及為其雲端 Proxy 設定閘道防火牆規則，請參閱 [VMware Cloud on AWS 使用案例](#) 工作流程。

### 必要條件

- 確認您擁有必要的 VMware Cloud on AWS 管理員認證，包括 vCenter 中目標 SDDC 的 VMware Cloud on AWS CloudAdmin 認證，並且已在連接埠 443 上啟用 HTTPS 存取。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 如果沒有外部網際網路存取權，則設定網際網路伺服器 Proxy。請參閱[如何為 vRealize Automation 設定網際網路 Proxy 伺服器](#)。

### 程序

- 1 選取**基礎結構 > 連線 > 雲端帳戶**，按一下**新增雲端帳戶**並選取 VMware Cloud on AWS 帳戶類型。
- 2 為您的組織新增 **VMC API Token** 以存取可用的 SDDC。

您可以在連結的 **API Token** 頁面上，為您的組織建立新 Token 或使用現有的 Token。如需詳細資料，請參閱在範例工作流程中於 [vRealize Automation 中建立 VMware Cloud on AWS 雲端帳戶](#)。

- 3 選取可用於部署的 SDDC。

不支援 NSX-V SDDC，因此它不會顯示在清單中。

會根據 SDDC 自動填入 vCenter 和 NSX-T Manager IP 位址/FQDN 值。

- 4 輸入指定 SDDC 的 vCenter 使用者名稱和密碼 (如果不是預設值 cloudadmin@vmc.local)。
- 5 按一下**驗證**以確認您對指定 vCenter 的存取權限，並檢查 vCenter 是否正在執行。

會收集與該帳戶相關聯的資料中心。

- 6 若要提高效率，請建立雲端區域以佈建到所選的 SDDC。

也可以根據組織的雲端策略在單獨的步驟中建立雲端區域。



7 如果需要新增標籤以支援標記策略，請輸入功能標籤。

您可以稍後新增或移除功能標籤。請參閱[如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署](#)。

8 按一下**儲存**。

#### 結果

將新增雲端帳戶，並且選取的 SDDC 可用於指定的雲端區域。

#### 後續步驟

若要針對 VMware Cloud on AWS 正確設定 vRealize Automation，請參閱[VMware Cloud on AWS 使用案例](#)。

如需 vRealize Automation 之外的 VMware Cloud on AWS 的相關資訊，請參閱[VMware Cloud on AWS 說明文件](#)。

## 將 vRealize Automation 與其他應用程式整合

整合可讓您將外部系統新增至 vRealize Automation。

整合包括 vRealize Orchestrator、組態管理，以及其他外部系統 (例如 GitHub、Ansible、Puppet) 和外部 IPAM 提供者 (例如 Infoblox)。

---

**備註** 如果您沒有外部網際網路存取權，但您的整合需要存取權，則可以設定網際網路伺服器 Proxy。請參閱[如何為 vRealize Automation 設定網際網路 Proxy 伺服器](#)。

---

## 如何在 vRealize Automation Cloud Assembly 中使用 GitLab 和 GitHub 整合

vRealize Automation Cloud Assembly 支援與 GitLab 和 GitHub 存放庫整合，以便您可以在原始檔控制下管理藍圖和動作指令碼。此功能可促進對部署程序的稽核和財務責任落實。

您必須具有適當的本機 Git 存放庫，並為指定的所有使用者設定存取權限，以便設定 Git 與 vRealize Automation Cloud Assembly 的整合。此外，您必須在特定結構中儲存藍圖，以便 Git 偵測到這些藍圖。若要建立與 GitLab 或 GitHub 的整合，請在 Cloud Assembly 中選取**基礎結構 > 連線 > 整合**，然後做出適當的選擇。您需要具備目標存放庫的 URL 和 Token。

使用現有存放庫設定 Git 整合時，與所選專案相關聯的所有藍圖將可供合格使用者使用。您可以將這些藍圖用於現有部署，也可以將其用作新部署的基礎。新增專案時，您必須選取與此專案在 Git 中的儲存位置和儲存方式相關的一些內容。

您可以直接從 vRealize Automation Cloud Assembly 將動作儲存至 Git 存放庫。您可以直接在 Git 中對動作指令碼進行版本設定，也可以在 vRealize Automation Cloud Assembly 中建立版本。如果您在 vRealize Automation Cloud Assembly 中建立動作的版本，則會自動將其做為版本儲存至 Git。藍圖稍微複雜些，因為您無法直接將其從 vRealize Automation Cloud Assembly 新增至 Git 整合。您必須將其直接儲存至 Git 執行個體，然後在使用 vRealize Automation Cloud Assembly 中的藍圖管理頁面時可以從 Git 中擷取這些藍圖。

## 開始之前

您必須在特定結構中建立並儲存藍圖，以便 GitLab 或 GitHub 偵測到它們。

- 設定和儲存藍圖，以便正確地與 GitLab 整合。只有有效的藍圖會匯入至 GitLab。
  - 為藍圖建立一或多個指定資料夾。
  - 所有藍圖都必須儲存在 `blueprint.yaml` 檔案中。
  - 確認藍圖的頂端包括 `name:` 和 `version:` 內容。
- 為適當的存放庫擷取 API 金鑰。在 Git 帳戶中，選取右上角的登入資訊，然後導覽至 [設定] 功能表。選取 **存取 Token**，並命名 Token，設定到期日期。然後，選取 API 並建立 Token。複製產生的值並儲存。

對於用於 Git 整合的所有藍圖，您必須遵循以下準則。

- 每個藍圖必須位於單獨的資料夾中。
- 所有藍圖都必須命名為 `blueprint.yaml`。
- 所有藍圖 YAML 檔案都必須使用 `name` 和 `version` 欄位。
- 僅會匯入有效的藍圖。
- 如果您更新從 Git 匯入的草稿藍圖，而且該藍圖的內容與最高版本中的不同，則在後續的同步中不會更新草稿，並且會建立新版本。如果您要更新藍圖，並同時允許來自 Git 的後續同步，則必須在完成最終變更後建立新版本。
- [在 vRealize Automation Cloud Assembly 中設定 GitLab 藍圖整合](#)  
此程序說明了如何在 vRealize Automation Cloud Assembly 中設定 GitLab 整合，以便您可以使用存放庫中的藍圖，並自動下載與指定專案相關聯的已儲存藍圖。若要將藍圖與 GitLab 搭配使用，您必須建立與適當 GitLab 執行個體的連線，然後將所需藍圖儲存至該執行個體。
- [在 vRealize Automation Cloud Assembly 中設定 GitHub 整合](#)  
您可以在 vRealize Automation Cloud Assembly 中整合 GitHub 雲端式存放庫主控服務

## 在 vRealize Automation Cloud Assembly 中設定 GitLab 藍圖整合

此程序說明了如何在 vRealize Automation Cloud Assembly 中設定 GitLab 整合，以便您可以使用存放庫中的藍圖，並自動下載與指定專案相關聯的已儲存藍圖。若要將藍圖與 GitLab 搭配使用，您必須建立與適當 GitLab 執行個體的連線，然後將所需藍圖儲存至該執行個體。

使用現有的存放庫設定 GitLab 整合時，與選取的專案相關聯的所有藍圖將可供合格使用者使用。您可以將這些藍圖用於現有部署，也可以將其用作新部署的基礎。新增專案時，您必須選取與此專案在 GitLab 中的儲存位置和儲存方式相關的一些內容。

---

**備註** 您無法從 vRealize Automation Cloud Assembly 將新的或已更新的藍圖推送至 Git 存放庫。此外，您無法從 vRealize Automation Cloud Assembly 將新藍圖推送至存放庫。若要將藍圖新增至存放庫，開發人員必須使用 Git 介面。

---

如果您更新從 Git 匯入的草稿藍圖，而且該藍圖的內容與最高版本中的不同，則在後續的同步中不會更新草稿，並且會建立新版本。如果您要更新藍圖，並同時允許來自 Git 的後續同步，則必須在完成最終變更後建立新版本。

設定要與 GitLab 搭配使用的藍圖，並收集所需資訊後，您必須設定與 GitLab 執行個體的整合。然後，您可以將指定的藍圖匯入 GitLab。您可以在 <https://www.youtube.com/watch?v=hOvqo63Sdgg> 中檢視此程序的視訊示範。

#### 必要條件

- 為適當的存放庫擷取 API 金鑰。在 GitLab 帳戶中，選取右上角的登入資訊，然後導覽至 [設定] 功能表。選取 [存取 Token]，並為 Token 命名，設定到期日期。然後，選取 API 並建立 Token。複製產生的值並儲存。

您必須具有適當的本機 Git 存放庫，並為指定的所有使用者設定存取權限，以便設定 Git 與 vRealize Automation Cloud Assembly 的整合。此外，您必須在特定結構中建立並儲存藍圖，以便 GitLab 偵測到它們。

- 設定和儲存藍圖，以便正確地與 GitLab 整合。只有有效的藍圖會匯入至 GitLab。請參閱[如何在 vRealize Automation Cloud Assembly 中使用 GitLab 和 GitHub 整合](#)。

#### 程序

- 1 在 vRealize Automation Cloud Assembly 中設定與 GitLab 環境的整合。
  - a 選取**基礎結構 > 整合 > 新增**，然後選擇 GitLab。
  - b 輸入 GitLab 執行個體的 **URL**。對於軟體即服務 GitLab 執行個體，大多數情況下，它是 gitlab.com。
  - c 為指定的 GitLab 執行個體輸入 **Token** (也稱為 API 金鑰)。如需從 GitLab 執行個體擷取 Token 的相關資訊，請參閱上述必要條件。
  - d 新增適當的名稱和說明。
  - e 按一下**驗證**來驗證連線。
  - f 視需要新增功能標籤。如需詳細資訊，請參閱[在 vRealize Automation Cloud Assembly 中使用功能標籤](#)。
  - g 按一下**新增**。
- 2 設定 GitLab 連線，以接受相應存放庫中的藍圖。
  - a 選取**基礎結構整合**，然後選擇適當的 GitLab 整合。
  - b 選取**專案**。
  - c 選取**新增專案**，並為此專案建立一個名稱。
  - d 在 GitLab 中輸入**存放庫路徑**。通常，這是附加至存放庫名稱的主要帳戶的使用者名稱。
  - e 輸入要使用的適當 GitLab **分支**。
  - f 輸入**資料夾名稱** (如適用)。如果保留空白，則所有資料夾皆可供使用。

- g 輸入適當的**類型**。輸入資料夾名稱 (如適用)。如果保留空白，則所有資料夾皆可供使用。
  - h 按下一步完成新增存放庫。
- 按下一步時，會起始將藍圖匯入平台的自動同步工作。
- 同步工作完成後，會顯示一則訊息，指出已匯入藍圖。

#### 結果

現在，您可以從 GitLab 擷取藍圖。

## 在 vRealize Automation Cloud Assembly 中設定 GitHub 整合

您可以在 vRealize Automation Cloud Assembly 中整合 GitHub 雲端式存放庫主控服務

您需要有效的 GitHub Token，才能在 vRealize Automation Cloud Assembly 中設定 GitHub 整合。如需建立和尋找 Token 的相關資訊，請參閱 GitHub 說明文件。

#### 必要條件

- 您必須具有 GitHub 的存取權。
- 設定和儲存藍圖，以便正確地與 GitHub 整合。只有有效的藍圖才會匯入至 GitHub。請參閱[如何在 vRealize Automation Cloud Assembly 中使用 GitLab 和 GitHub 整合](#)。

#### 程序

- 1 選取**基礎結構 > 連線 > 整合**，然後按一下**新增整合**。
- 2 選取 GitHub。
- 3 在 GitHub 組態頁面上，輸入所需資訊。
- 4 按一下**驗證**來檢查整合。
- 5 如果需要新增標籤以支援標記策略，請輸入功能標籤。請參閱[如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署](#)和 [建立標記策略](#)。
- 6 按一下**新增**。

#### 結果

GitHub 可在 vRealize Automation Cloud Assembly 藍圖中使用。

#### 後續步驟

現在，您可以從 GitHub 擷取藍圖。

## 在 vRealize Automation 中設定外部 IPAM 整合點

您可以建立提供者特定的外部 IPAM 整合點，以管理您的藍圖部署中使用的 IP 位址。使用外部 IPAM 整合點時，會從指定的 IPAM 提供者 (而非 vRealize Automation) 取得 IP 位址並由其進行管理。

您可以建立提供者特定的 IPAM 整合點，以在 vRealize Automation 中管理藍圖部署和虛擬機器的 IP 位址和 DNS 設定。

如需如何設定這些必要條件的相關資訊，以及如何在範例工作流程的內容中建立提供者特定外部 IPAM 整合點的範例，請參閱在 [vRealize Automation 中新增外部 IPAM 整合點](#)。

如需如何建立所需資產以便外部 IPAM 合作夥伴和廠商能夠將其 IPAM 解決方案與 vRealize Automation 整合的相關資訊，請參閱[如何使用 IPAM SDK 為 vRealize Automation 建立提供者特定的外部 IPAM 整合套件](#)。

#### 必要條件

- 確認您具有雲端管理員認證。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 確認您擁有外部 IPAM 提供者的帳戶，例如 [Infoblox](#) 或 [Bluecat](#)，並且您擁有對組織的 IPAM 提供者帳戶的正確存取認證。
- 確認您有權存取 IPAM 提供者 (例如 Infoblox 或 BlueCat) 的已部署整合套件。已部署的套件最初是從 IPAM 提供者或從 vRealize Automation 市集以 .zip 形式下載，然後部署至 vRealize Automation。
- 確認您可以存取為 IPAM 提供者設定的執行環境。
- 如果您使用以動作為基礎的擴充性 (ABX) 內部部署內嵌式執行環境，請確認 vRealize Automation 網路中具有可將傳出流量傳遞至外部站台 (例如 gcr.io 和 storage.googleapis.com) 的 HTTP Proxy 伺服器。如需詳細資料，請參閱〈在 [vRealize Automation 8.x 中提取位於 Proxy 後方的 Docker 映像 \(75180\)](#)〉。

#### 程序

- 1 選取**基礎結構 > 連線 > 整合**，然後按一下**新增整合**。
- 2 按一下**IPAM**。
- 3 在**提供者**下拉式清單中，從清單中選取已設定的 IPAM 提供者套件。  
如果清單為空白，請按一下**匯入提供者套件**，導覽至現有提供者套件 .zip 檔案，然後加以選取。如果沒有 .zip 檔案，可以從提供者的網站或從 vRealize Automation **市集**索引標籤取得該檔案。
- 4 輸入您的外部 IPAM 提供者帳戶的管理員使用者名稱和密碼認證，以及所有其他 (如有) 必填欄位 (例如提供者的主機名稱)。
- 5 在**執行環境**下拉式清單中，選取現有的執行環境，例如以動作為基礎的內部部署擴充性整合點。

執行環境支援 vRealize Automation 與 IPAM 提供者之間的通訊。

IPAM 架構僅支援以動作為基礎的擴充性 (ABX) 內部部署內嵌式執行環境。

---

**備註** 如果您使用 Amazon Web Services 或 Microsoft Azure 雲端帳戶做為整合執行環境，請確保 IPAM 提供者應用裝置可從網際網路進行存取，並且不在 NAT 或防火牆後方，且具有可公開解析的 DNS 名稱。如果 IPAM 提供者不可存取，則無法將 Amazon Web Services Lambda 或 Microsoft Azure 函數連線到該提供者，整合將會失敗。

---

- 6 按一下**驗證**。

- 當系統提示您信任來自外部 IPAM 提供者的自我簽署憑證時，請按一下**接受**。

接受自我簽署的憑證後，可以繼續完成驗證動作。

- 輸入此 IPAM 整合點的名稱，然後按一下**新增**以儲存新的 IPAM 整合點。

將啟動資料收集動作。將從外部 IPAM 提供者收集網路和 IP 位址的資料。

## 如何在 vRealize Automation 中升級到較新的 IPAM 整合套件

您可以升級現有的外部 IPAM 整合點，以取得更新版本的廠商特定的 IPAM 整合套件。

外部 IPAM 提供者或 VMware 可能會升級特定廠商的來源 IPAM 整合套件。例如，用於 Infoblox 的外部 IPAM 整合套件已多次升級。若要保留使用具名 IPAM 整合點的任何現有 vRealize Automation 基礎結構設定，您可以編輯 IPAM 整合點以取得更新的 IPAM 整合套件，而非建立新的 IPAM 整合點。

### 必要條件

此程序假設您已建立外部 IPAM 整合點，並且想要升級該整合點以使用更新版本的廠商 IPAM 整合套件。

如需有關如何建立外部 IPAM 整合點的資訊，請參閱在 [vRealize Automation 中新增外部 IPAM 整合點](#)。

- 確認您具有雲端管理員認證。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 確認您擁有外部 IPAM 提供者的帳戶，並擁有對組織 IPAM 提供者帳戶的正確存取認證。
- 確認您可以存取 IPAM 提供者的已部署整合套件。已部署的套件最初是從 IPAM 提供者網站或從 vRealize Automation 市集以 .zip 形式下載，然後部署至 vRealize Automation。

如需如何下載和部署提供者套件 .zip 檔案並在 [IPAM 整合] 頁面上將其做為**提供者**值使用的相關資訊，請參閱[下載並部署要在 vRealize Automation Cloud Assembly 中使用的外部 IPAM 提供者套件](#)。

- 確認您可以存取為 IPAM 提供者設定的執行環境。執行環境通常是以動作為基礎的擴充性 (ABX) 內部部署內嵌式整合點。

如需執行環境特性的相關資訊，請參閱在 [vRealize Automation 中為 IPAM 整合點建立執行環境](#)。

### 程序

- 選取**基礎結構 > 連線 > 整合 IPAM**，然後開啟現有的 IPAM 整合點。
- 按一下**管理提供者**。
- 導覽至並匯入更新的 IPAM 整合套件。
- 按一下**驗證**，然後按一下**儲存**。

## 在 vRealize Automation Cloud Assembly 中設定 My VMware 整合

您可以將 My VMware 與 vRealize Automation Cloud Assembly 整合，以支援與 VMware 相關的動作和功能，例如存取 VMware Marketplace 中的藍圖。



您只能為每個組織建立一個 My VMware 整合。

#### 必要條件

您必須擁有一個具有 My VMware 適當權限的使用者帳戶。

- 如需邀請使用者使用 My VMware 帳戶的相關資訊，請參閱[知識庫 2070555](#)。
- 如需在 My VMware 帳戶中指派使用者權限的相關資訊，請參閱[知識庫 2006977](#)。

#### 程序

- 1 選取**基礎結構 > 連線 > 整合**，然後按一下**新增整合**。
- 2 選取 My VMware。
- 3 在 [My VMware 組態] 頁面上，輸入所需的資訊。
- 4 如果需要標籤以支援標記策略，請輸入功能標籤。請參閱[如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署](#)和 [建立標記策略](#)。
- 5 按一下**新增**。

#### 結果

My VMware 可與藍圖搭配使用。

#### 後續步驟

將 My VMware 元件新增至所需藍圖。

## 在 Cloud Assembly 中設定 vRealize Orchestrator 整合

您可以設定一或多個 vRealize Orchestrator 整合，以便將工作流程用作擴充性的一部分。

vRealize Automation 包含可用於擴充性訂閱的預先設定的 vRealize Orchestrator 執行個體。您也可以從 vRealize Automation Cloud Services 主控台存取內嵌式 vRealize Orchestrator 的用戶端。

透過將 vRealize Orchestrator 整合至 vRealize Automation Cloud Assembly，您可以新增外部 vRealize Orchestrator 執行個體，並使用擴充性訂閱中包含的工作流程程式庫。如需詳細資訊，請參閱[擴充性工作流程訂閱](#)。

#### 必要條件

- 確認您具有雲端管理員認證。如需詳細資訊，請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 將 vRealize Orchestrator 7.x 執行個體移轉到版本 8.0。請參閱《安裝、設定和移轉 VMware vRealize Orchestrator》中的〈將獨立 vRealize Orchestrator 移轉到外部 vRealize Orchestrator 8.0〉。

#### 程序

- 1 選取**基礎結構 > 連線 > 整合**。
- 2 按一下**新增整合**。

- 3 選取 vRealize Orchestrator。
- 4 在 vRealize Automation Cloud Assembly 中，輸入 vRealize Orchestrator 執行個體的 URL。
- 5 若要驗證整合，請按一下**驗證**。
- 6 輸入 vRealize Orchestrator 整合的名稱。
- 7 (選擇性) 輸入 vRealize Orchestrator 整合的說明。
- 8 (選擇性) 新增功能標籤。如需有關功能標籤的詳細資訊，請參閱在 [vRealize Automation Cloud Assembly 中使用功能標籤](#)。

---

**備註** 功能標籤可以用來管理多個 vRealize Orchestrator 整合。請參閱[使用專案限制管理多個 vRealize Orchestrator 整合](#)。

---

- 9 按一下**新增**。

將儲存 vRealize Orchestrator 整合。

#### 後續步驟

若要確認整合已設定且工作流程已新增，請選取**擴充性 > 程式庫 > 工作流程**。

## 使用專案限制管理多個 vRealize Orchestrator 整合

您可以使用專案限制來管理工作流程訂閱中使用的 vRealize Orchestrator 整合。

vRealize Automation Cloud Assembly 支援整合可在工作流程訂閱中使用的多個 vRealize Orchestrator 伺服器。您可以使用專案軟限制或硬限制來管理由專案佈建的藍圖中使用的 vRealize Orchestrator 整合。如需有關專案限制的詳細資訊，請參閱[使用 vRealize Automation Cloud Assembly 專案標籤和自訂內容](#)。

#### 必要條件

- 確認您具有雲端管理員認證。如需詳細資訊，請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 在 vRealize Automation Cloud Assembly 中，設定兩個或多個 vRealize Orchestrator 整合。如需詳細資訊，請參閱在 [Cloud Assembly 中設定 vRealize Orchestrator 整合](#)。
- 將功能標籤新增至 vRealize Orchestrator 整合。如需有關功能標籤的詳細資訊，請參閱在 [vRealize Automation Cloud Assembly 中使用功能標籤](#)。

#### 程序

- 1 導覽至**基礎結構 > 設定 > 專案**，然後選取您的專案。
- 2 選取**佈建索引標籤**。
- 3 在**擴充性限制文字方塊**中輸入 vRealize Orchestrator 整合的功能標籤，並將其設定為專案軟限制或硬限制。
- 4 按一下**儲存**。



## 結果

部署藍圖時，vRealize Automation Cloud Assembly 使用專案限制來管理工作流程訂閱中使用的 vRealize Orchestrator 整合。

## 後續步驟

或者，您可以使用功能標籤來管理雲端帳戶層級上的多個 vRealize Orchestrator 整合。如需詳細資訊，請參閱[使用雲端帳戶功能標籤管理多個 vRealize Orchestrator 整合](#)。

## 使用雲端帳戶功能標籤管理多個 vRealize Orchestrator 整合

您可以使用功能標籤來管理工作流程訂閱中使用的 vRealize Orchestrator 整合。

vRealize Automation Cloud Assembly 支援整合可在工作流程訂閱中使用的多個 vRealize Orchestrator 伺服器。您可以透過將功能標籤新增至雲端帳戶來管理工作流程訂閱中使用的 vRealize Orchestrator 整合。

## 必要條件

- 確認您具有雲端管理員認證。如需詳細資訊，請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 在 vRealize Automation Cloud Assembly 中，設定兩個或多個 vRealize Orchestrator 整合。如需詳細資訊，請參閱在 [Cloud Assembly 中設定 vRealize Orchestrator 整合](#)。
- 將功能標籤新增至 vRealize Orchestrator 整合。如需有關功能標籤的詳細資訊，請參閱在 [vRealize Automation Cloud Assembly 中使用功能標籤](#)。

## 程序

- 1 導覽至**基礎結構 > 連線 > 雲端帳戶**。
- 2 選取您的雲端帳戶。
- 3 輸入您想要使用的 vRealize Orchestrator 整合的功能標籤。  
功能標籤會自動轉換為軟限制。若要在管理整合時使用硬限制，您必須使用專案限制。如需詳細資訊，請參閱[使用專案限制管理多個 vRealize Orchestrator 整合](#)。
- 4 按一下**儲存**。

## 結果

部署藍圖時，vRealize Automation Cloud Assembly 在相關聯的雲端帳戶中使用標記功能來管理工作流程訂閱中使用的 vRealize Orchestrator 整合。

## 如何在 vRealize Automation Cloud Assembly 中使用 Kubernetes

您可以將 Pivotal Container Service (PKS) 或 Red Hat OpenShift 與 vRealize Automation Cloud Assembly 整合，以管理和部署 Kubernetes 資源。也可以在 vRealize Automation Cloud Assembly 中整合外部 Kubernetes 資源。

建立 PKS 或 OpenShift 整合後，相應的 Kubernetes 叢集將可在 vRealize Automation Cloud Assembly 中使用，您可以新增和建立 Kubernetes 元件至 vRealize Automation Cloud Assembly，以支援叢集和容器應用程式的管理。這些應用程式構成可從 Service Broker 目錄取得的自助服務部署的基礎。

- [在 vRealize Automation Cloud Assembly 中設定 PKS 整合](#)

您可以在 vRealize Automation Cloud Assembly 中設定內部部署和雲端中的 PKS 資源連線，以支援 Kubernetes 整合及管理功能。

- [在 vRealize Automation Cloud Assembly 中使用 Kubernetes 叢集和命名空間](#)

您可以在 vRealize Automation Cloud Assembly 中新增、檢視和管理用作 Kubernetes 部署基礎的 Kubernetes 叢集和命名空間的組態。

- [在 vRealize Automation Cloud Assembly 中設定 Kubernetes 區域](#)

透過 Kubernetes 區域，雲端管理員可以定義在 vRealize Automation Cloud Assembly 部署中使用的 Kubernetes 叢集和命名空間的原則導向放置。管理員可使用此頁面來指定哪些叢集可用於佈建 Kubernetes 命名空間，以及可接受哪些內容用於叢集。

- [在 vRealize Automation Cloud Assembly 中將 Kubernetes 元件新增至藍圖](#)

將 Kubernetes 元件新增到 vRealize Automation Cloud Assembly 藍圖時，您可以選擇新增叢集或讓使用者能夠在各種組態中建立命名空間。此選項通常取決於存取控制需求、設定 Kubernetes 元件的方式以及部署需求。

- [將 vRealize Automation Cloud Assembly 擴充性與 Kubernetes 搭配使用](#)

vRealize Automation Cloud Assembly 提供一組標準事件主題，對應於與 Kubernetes 叢集部署相關的一般動作。使用者可以視需要訂閱這些主題，並且會在與已訂閱主題相關的事件發生時收到通知。您也可以設定 vRO 工作流程以根據事件通知執行。

## 在 vRealize Automation Cloud Assembly 中設定 PKS 整合

您可以在 vRealize Automation Cloud Assembly 中設定內部部署和雲端中的 PKS 資源連線，以支援 Kubernetes 整合及管理功能。

PKS 整合可讓您管理內部部署和雲端中的 PKS 執行個體，以及 PKS 上佈建的 Kubernetes 叢集和外部叢集。您必須建立 Kubernetes 設定檔並將其與專案相關聯，以支援以原則為基礎的資源放置。

### 必要條件

- 您必須使用 UAA 驗證設定適當設定的 Pivotal Container Service (PKS) 伺服器。
- 確認您具有雲端管理員認證。如需詳細資訊，請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。

### 程序

- 1 選取**基礎結構 > 連線 > 整合**，然後按一下**新增整合**。
- 2 選取 VMware Enterprise PKS。

- 3 輸入您要建立的 PKS 雲端帳戶的 IP 位址或 FQDN 和 PKS 位址。
  - IP 位址是 PKS 使用者驗證伺服器的 FQDN 或 IP 位址。
  - PKS 位址是主 PKS 伺服器的 FQDN 或 IP 位址。
- 4 選取此 PKS 伺服器是位於本機，還是位於公有雲或私有雲上。
- 5 輸入 PKS 伺服器的適當**使用者名稱和密碼**以及其他相關資訊。
- 6 如果使用標籤以支援標記策略，請輸入功能標籤。請參閱[如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署](#)和 [建立標記策略](#)。
- 7 按一下**新增**。

## 結果

您可以建立 Kubernetes 區域並將其指派給專案，也可以探索外部 Kubernetes 叢集並將其指派給專案。此外，您可以新增或建立 Kubernetes 命名空間，以便於在大型群組和組織之間管理叢集。

## 後續步驟

建立或選取適當的 Kubernetes 區域，選取一或多個叢集或命名空間，然後將其指派給專案。之後，您可以建立並發佈藍圖，讓使用者能夠產生使用 Kubernetes 的自助服務部署。

## 在 vRealize Automation Cloud Assembly 中使用 Kubernetes 叢集和命名空間

您可以在 vRealize Automation Cloud Assembly 中新增、檢視和管理用作 Kubernetes 部署基礎的 Kubernetes 叢集和命名空間的組態。

您可以在**基礎結構 > 資源 > Kubernetes**頁面上檢視、新增和管理您有權存取的 Kubernetes 叢集和命名空間。此頁面通常可協助管理已部署的叢集和命名空間。

- **叢集**：叢集是一組分散在一或多個實體機器上的 Kubernetes 節點。此頁面會顯示已佈建和取消部署的叢集，這些叢集已設定為在 vRealize Automation Cloud Assembly 執行個體上使用。您可以按一下叢集以檢視其目前狀態的相關資訊。部署叢集時，它包含 Kubconfig 檔案的連結，該檔案僅供雲端管理員存取。此檔案會授與對叢集的完整管理員權限，包括命名空間清單。
- **命名空間**：命名空間是為管理員提供區隔叢集資源的方式的虛擬叢集。它們有利於在大型使用者群組和組織中管理資源。做為角色型存取控制的一種形式，雲端管理員可讓使用者能夠在申請部署時將命名空間新增至專案，然後從 [Kubernetes 叢集] 頁面管理這些命名空間。部署命名空間時，它包含 Kubconfig 檔案的連結，該檔案讓有效的使用者 (例如開發人員) 能夠檢視和管理該命名空間的某些方面。

如果您要設定新的或現有的叢集，則必須選取是否要使用主要 IP 位址或主控主機名稱進行連線。

## 在 vRealize Automation Cloud Assembly 中使用 Kubernetes 叢集

您可以使用此頁面上的選項，將新的、現有的或外部叢集新增到 vRealize Automation Cloud Assembly。

- 1 選取**基礎結構 > 資源 > Kubernetes**，並確認 [叢集] 索引標籤處於作用中狀態。

如果目前為 vRealize Automation Cloud Assembly 執行個體設定了任何叢集，則這些叢集會顯示在此頁面上。

- 2 如果您要新增新的或現有的叢集或者部署叢集，請根據下表選取適當的選項。

選項	說明	詳細資料
部署	將新叢集新增到 vRealize Automation Cloud Assembly	您必須指定要將此叢集部署到的 PKS 雲端帳戶，以及所需的計劃和節點數目。
新增現有	將現有叢集設定為使用專案。	您必須指定 PKS 雲端帳戶、要使用的叢集，以及適用於目標開發人員的適當專案。此外，您還需要指定共用範圍。如果您想要全域共用，則必須適當設定 Kubernetes 區域和命名空間。
新增外部	將可能不會與 PKS 相關聯的 vanilla Kubernetes 叢集新增至 vRealize Automation Cloud Assembly。	您必須指定叢集所關聯的專案，並輸入所需叢集的 IP 位址，然後選取連線到此叢集所需的雲端 Proxy 和憑證資訊。

- 3 按一下**新增**以使叢集在 vRealize Automation Cloud Assembly 中可用。

### 在 vRealize Automation Cloud Assembly 中使用 Kubernetes 命名空間

如果您是雲端管理員，則命名空間可協助您分組和管理 Kubernetes 叢集資源。如果您是使用者，則命名空間是 Kubernetes 叢集中用於部署的區域。管理員和使用者可使用**基礎結構 > 資源 > Kubernetes** 頁面上的 [命名空間] 索引標籤存取命名空間。

有多種方式可將 Kubernetes 命名空間新增到 vRealize Automation Cloud Assembly 中的資源。下列程序概述了一種典型方法。

- 1 選取**基礎結構 > 資源 > Kubernetes**，然後按一下 [命名空間] 索引標籤。
- 2 若要新增命名空間，請按一下**新增命名空間**。若要新增現有命名空間，請按一下**新增命名空間**。
- 3 輸入命名空間的**名稱與說明**。

此時，您已新增與 Kubernetes 資源搭配使用的命名空間，但此命名空間不與任何特定內容相關聯。

- 4 指定要與此命名空間相關聯的**叢集**。
- 5 按一下**建立**以將命名空間新增至 vRealize Automation Cloud Assembly。

### 在 vRealize Automation Cloud Assembly 中設定 Kubernetes 區域

透過 Kubernetes 區域，雲端管理員可以定義在 vRealize Automation Cloud Assembly 部署中使用的 Kubernetes 叢集和命名空間的原則導向放置。管理員可使用此頁面來指定哪些叢集可用於佈建 Kubernetes 命名空間，以及可接受哪些內容用於叢集。

雲端管理員可以將 Kubernetes 區域與為 Cloud Assembly 設定的 PKS 雲端帳戶建立關聯，或與未與專案相關聯的外部 Kubernetes 叢集建立關聯。

當您建立 Kubernetes 區域時，您可以將提供者特定的多個資源指派給該區域，這些資源將根據工作節點數、主節點數、可用 CPU 數、記憶體數和其他組態設定，規定可針對新佈建的叢集設定哪些內容。對於 PKS 提供者，它們與 PKS 計劃相對應。管理員也可以向 Kubernetes 區域指派多個叢集，該區域將用於放置新佈建的 Kubernetes 命名空間。管理員只能指派未上線或不受 CMX 管理，且透過預先選取的叢集提供者進行佈建的叢集。管理員可將多個 Kubernetes 區域指派給單一專案，從而使其全部可用於此專案中發生的放置作業。

雲端管理員可以在多個層級上指派優先順序。

- 專案內的 Kubernetes 區域優先順序。
- Kubernetes 區域內的資源優先順序。
- Kubernetes 區域內的叢集優先順序。

雲端管理員也可以在多個層級上指派標籤：

- 每個 Kubernetes 區域的功能標籤。
- 每個資源指派的標籤。
- 每個叢集指派的標籤。

Service Broker 包含 [Kubernetes 區域] 頁面的版本，可讓 Service Broker 管理員存取現有的 Kubernetes 區域，以便他們能夠為從目錄佈建的 Kubernetes 命名空間和叢集建立放置原則。

#### 必要條件

設定與適當 PKS 部署的整合。請參閱在 [vRealize Automation Cloud Assembly](#) 中設定 PKS 整合

#### 程序

- 1 選取**基礎結構 > 設定 > Kubernetes 區域**，然後按一下**新增 Kubernetes 區域**。
- 2 輸入要套用此區域的 PKS 整合帳戶名稱。
- 3 為 Kubernetes 區域新增**名稱和說明**。
- 4 新增功能標籤 (如果適用)。如需詳細資訊，請參閱在 [vRealize Automation Cloud Assembly](#) 中使用**功能標籤**。
- 5 按一下**儲存**。

- 6 按一下 [隨選] 索引標籤，然後視需要針對用於叢集佈建的區域新增 PKS 計劃。

您可以選取一或多個計劃，並為其指派優先順序。數字越小，優先順序越高。優先順序指派次於以標籤為基礎的選取。

- 7 按一下 [叢集] 索引標籤，然後按一下**新增**按鈕以將 Kubernetes 叢集新增至區域。如果您使用的是外部叢集，則當您選取該叢集時，它會自動上線至 vRealize Automation Cloud Assembly。

您可以在 vRealize Automation Cloud Assembly 的 [Kubernetes 叢集] 頁面上將 Kubernetes 命名空間新增至叢集。

#### 結果

Kubernetes 區域已設定為與 vRealize Automation Cloud Assembly 部署搭配使用。

## 後續步驟

向專案指派 Kubernetes 區域。

- 1 選取**基礎結構 > 設定 > 專案**，然後選取要與 Kubernetes 區域相關聯的專案。
- 2 在 [專案] 頁面上，按一下 [Kubernetes 佈建] 索引標籤。
- 3 按一下**新增 Kubernetes 區域**，然後新增您剛建立的區域。如果適用，您也可以有多個區域，還可以對區域設定優先順序。
- 4 按一下**儲存**。

向專案指派區域後，您可以使用 [藍圖] 頁面來根據 Kubernetes 區域和專案組態佈建部署。

## 在 vRealize Automation Cloud Assembly 中將 Kubernetes 元件新增至藍圖

將 Kubernetes 元件新增到 vRealize Automation Cloud Assembly 藍圖時，您可以選擇新增叢集或讓使用者能夠在各種組態中建立命名空間。此選項通常取決於存取控制需求、設定 Kubernetes 元件的方式以及部署需求。

若要在 vRealize Automation Cloud Assembly 中將 Kubernetes 元件新增至藍圖，請按一下 [藍圖]，選取**新增**，然後在左側功能表中找到並展開 Kubernetes 選項。然後，透過將所需選項 (叢集或 KBS 命名空間) 拖曳至畫布來進行選取。

將與專案相關聯的 Kubernetes 叢集新增至藍圖，是使 Kubernetes 資源可供有效使用者使用的最簡單方法。您可以在叢集上使用標籤來控制其部署位置，如同處理其他 Cloud Assembly 資源一樣。在叢集部署的配置階段，您可以使用標籤來選取區域和 PKS 計劃。

一旦以這種方式新增叢集，該叢集就會自動可供所有有效的使用者使用。

## 藍圖範例

第一個藍圖範例會顯示透過標記來控制的簡單 Kubernetes 部署的藍圖。建立的 Kubernetes 區域包含兩個部署計劃，並在 [新增 Kubernetes 區域] 頁面上進行設定。在此案例中，名為 `placement:tag` 的標籤已新增為區域上的一項功能，用於比對藍圖上的類似限制。如果有多個區域設定了此標籤，則會選取具有最低優先順序編號的區域。

```
formatVersion: 1
inputs: {}
resources:
  Cluster_provisioned_from_tag:
    type: Cloud.K8S.Cluster
    properties:
      hostname: 109.129.209.125
      constraints:
        -tag: 'placement tag'
      port: 7003
      workers: 1
      connectBy: hostname
```

第二個藍圖範例顯示如何使用名為 `$(input.hostname)` 的變數設定藍圖，以便使用者能夠在申請部署時輸入所需的叢集主機名稱。在叢集部署的資源配置階段，也可以使用標籤來選取區域和 PKS 計劃。

```
formatVersion: 1
inputs:
  hostname:
    type: string
    title: Cluster hostname
resources:
  Cloud_K8S_Cluster_1:
    type: Cloud.K8S.Cluster
    properties:
      hostname: ${input.hostname}
      port: 8443
      connectBy: hostname
      workers: 1
```

如果您想要使用命名空間來管理叢集使用率，您可以在名為 `name: ${input.name}` 的藍圖中設定一個變數，以取代申請部署時使用者輸入的命名空間名稱。對於此類部署，您需要建立一個如以下範例所示的藍圖：

```
1 formatVersion: 1
2 inputs:
3   name:
4     type: string
5     title: "Namespace name"
6 resources:
7   Cloud_K8S_Namespace_1:
8     type: Cloud.K8S.Namespace
9     properties:
10      name: ${input.name}
```

使用者可以透過 kubeconfig 檔案來管理已部署的叢集，這些檔案可從**基礎結構 > 資源 > Kubernetes 叢集**頁面存取。在頁面上找到所需叢集對應的卡，然後按一下 **Kubeconfig**。

## 將 vRealize Automation Cloud Assembly 擴充性與 Kubernetes 搭配使用

vRealize Automation Cloud Assembly 提供一組標準事件主題，對應於與 Kubernetes 叢集部署相關的一般動作。使用者可以視需要訂閱這些主題，並且會在與已訂閱主題相關的事件發生時收到通知。您也可以設定 vRO 工作流程以根據事件通知執行。

下列主題適用於 vRealize Automation Cloud Assembly 中的**擴充性 > 程式庫 > 事件主題**頁面上的訂閱。若要檢視這些主題，請在 [事件主題搜尋] 文字方塊中搜尋 Kubernetes。

- Kubernetes 叢集配置
- Kubernetes 叢集佈建後
- Kubernetes 叢集移除後
- Kubernetes 叢集佈建
- Kubernetes 叢集移除

按一下其中一個主題可檢視該主題的架構，其中會顯示收集和傳輸的所有資訊。您可以使用任何架構資訊來設定各種通知和管理及報告工作。

您可以在**擴充性 > 程式庫 > 動作**頁面上設定 CMX 相關動作的動作指令碼。動作指令碼可用於多種用途：例如，建立 Kubernetes 叢集佈建的 DNS 記錄。如果您要建立 DNS 記錄，您可以在動作指令碼中使用 Kubernetes 叢集佈建後主題中的 `masternodeips` 欄位與 REST 命令，以建立 DNS 記錄。

[訂閱] 頁面定義事件主題和動作指令碼之間的關係。您可以在 vRealize Automation Cloud Assembly 的 [訂閱] 頁面上檢視和管理這些元件。

## 什麼是 vRealize Automation Cloud Assembly 中的組態管理

vRealize Automation Cloud Assembly 支援與 Puppet Enterprise 和 Ansible 開放原始碼整合，以便您可以針對組態和偏差管理部署。

### Puppet 整合

若要整合以 Puppet 為基礎的組態管理，您必須安裝在具有 vSphere 工作負載的公有或私有雲上安裝有效的 Puppet Enterprise 執行個體。您必須在此外部系統和 vRealize Automation Cloud Assembly 執行個體之間建立連線。然後，您可以透過將 Puppet 組態管理新增至適當的藍圖，使其可供 vRealize Automation Cloud Assembly 使用。

vRealize Automation Cloud Assembly 藍圖服務 Puppet 提供者會在已部署的計算資源上安裝、設定和執行 Puppet 代理程式。Puppet 提供者同時支援 SSH 和 WinRM 連線，並具備下列必要條件：

- SSH 連線：
  - 若要執行具有 `NOPASSWD` 的命令，使用者名稱必須是超級使用者或具有 `sudo` 權限的使用者。
  - 對指定的使用者停用 `requiretty`。
  - `cURL` 必須在部署計算資源上可用。
- WinRM 連線：
  - PowerShell 2.0 必須在部署計算資源上可用。
  - 設定 Windows 範本，如 vRealize Orchestrator 說明文件中所述。

DevOps 管理員負責管理 Puppet Master 連線，以及將 Puppet 角色或組態規則套用至特定的部署。執行下列部署後，設定為支援組態管理的虛擬機器將登錄到指定的 Puppet Master。

部署虛擬機器時，使用者可以新增或刪除做為外部系統的 Puppet Master 或更新指派給 Puppet Master 的專案。最後，當機器解除委任時，適當的使用者可以從 Puppet Master 解除登錄已部署的虛擬機器。

### Ansible 開放原始碼整合

設定 Ansible 整合時，請按照 Ansible 安裝指示安裝 Ansible 開放原始碼。如需有關安裝的詳細資訊，請參閱 Ansible 說明文件。



依預設，Ansible 啟用主機金鑰檢查。如果使用 `known_hosts` 檔案中的其他金鑰重新安裝了主機，則會出現錯誤訊息。如果主機未列在 `known_hosts` 檔案中，您必須在啟動時提供金鑰。您可以透過 `/etc/ansible/ansible.cfg` 或 `~/.ansible.cfg` 檔案中的下列設定停用主機金鑰檢查：

```
[defaults]
host_key_checking = False
localhost_warning = False

[paramiko_connection]
record_host_keys = False

[ssh_connection]
#ssh_args = -C -o ControlMaster=auto -o ControlPersist=60s
ssh_args = -o UserKnownHostsFile=/dev/null
```

若要避免主機金鑰檢查錯誤，請將 `host_key_checking` 和 `record_host_keys` 設定為 `False`，並新增在 `ssh_args` 中設定的額外選項 `UserKnownHostsFile=/dev/null`。此外，如果詳細目錄一開始為空白，則 Ansible 會發出主機清單為空白的警告。這會導致 playbook 語法檢查失敗。

透過 Ansible 保存庫，您可以在加密的檔案中而非純文字格式儲存敏感資訊（例如密碼或金鑰）。保存庫使用密碼進行加密。在 vRealize Automation Cloud Assembly 中，Ansible 使用保存庫對主機的 ssh 密碼等資料進行加密。它會假設已設定保存庫密碼的路徑。

您可以修改 `ansible.cfg` 檔案，以使用下列格式指定密碼檔案的位置。

```
vault_password_file = /path to/file.txt
```

也可以設定 `ANSIBLE_VAULT_PASSWORD_FILE` 環境變數，以便 Ansible 自動搜尋密碼。例如，`ANSIBLE_VAULT_PASSWORD_FILE=~/.vault_pass.txt`

vRealize Automation Cloud Assembly 管理 Ansible 詳細目錄檔案，因此您必須確保 vRealize Automation Cloud Assembly 使用者對詳細目錄檔案具有 `rwX` 存取權。

```
cat ~/var/tmp/vmware/provider/user_defined_script/${ls -t ~/var/tmp/vmware/provider/
user_defined_script/ | head -1)/log.txt
```

如果您想要使用具有 vRealize Automation Cloud Assembly 開放原始碼整合的非 root 使用者，使用者需要一組權限來執行 vRealize Automation Cloud Assembly 開放原始碼提供者所使用的命令。必須在使用者的 `sudoers` 檔案中設定下列命令。

```
Defaults:myuser !requiretty
```

如果使用者不屬於未指定 `askpass` 應用程式的管理員群組，則在使用者的 `sudoers` 檔案中設定下列命令：

```
myuser ALL=(ALL) NOPASSWD: ALL
```

如果在設定 Ansible 整合時遇到錯誤或其他問題，請參閱位於 Ansible 控制機器上的 `'cat~/var/tmp/vmware/provider/user_defined_script/${ls -t ~/var/tmp/vmware/provider/user_defined_script/ | head -1)'/` 中的 `log.txt` 檔案。

## 在 vRealize Automation Cloud Assembly 中設定 Puppet Enterprise 整合

vRealize Automation Cloud Assembly 支援與 Puppet Enterprise 組態管理整合。

將 Puppet Enterprise 做為外部系統新增至 Cloud Assembly 時，依預設它在所有專案上均可使用。您可以將其限制為僅限特定專案使用。

若要新增 Puppet Enterprise 整合，您必須具有 Puppet Master 名稱以及 Master 的主機名稱或 IP 位址。

如果需要檢查 Puppet 記錄是否有錯誤或出於資訊目的，您可以在下列位置中找到這些記錄。

說明	記錄位置
與建立及安裝相關的事件的記錄	這些記錄位於已部署機器上的 <code>~/var/tmp/vmware/provider/user_defined_script/\${ls -t ~/var/tmp/vmware/provider/user_defined_script/   head -1}/` 中。 如需完整記錄，請參閱 <code>log.txt</code> 檔案。如需詳細的 Puppet 代理程式記錄，請參閱 <a href="https://puppet.com/docs/puppet/4.8/services_agent_unix.html#logging">https://puppet.com/docs/puppet/4.8/services_agent_unix.html#logging</a></code>
與 Puppet 刪除及執行相關的工作的記錄	這些記錄位於 PE 上的 <code>~/var/tmp/vmware/provider/user_defined_script/\${ls -t ~/var/tmp/vmware/provider/user_defined_script/   head -1}/` 中。如需完整記錄，請參閱 <code>log.txt</code> 檔案。</code>

### 程序

- 1 選取**基礎結構 > 連線 > 整合**，然後按一下**新增整合**。
- 2 選取 [Puppet]。
- 3 在 [Puppet 組態] 頁面上，輸入所需的資訊。
- 4 按一下**驗證**來檢查整合。
- 5 按一下**新增**。

### 結果

Puppet 可與藍圖搭配使用。

### 後續步驟

將 Puppet 元件新增至所需藍圖。

- 1 在藍圖功能表上的 [內容管理] 標題下選取 [Puppet]，然後將 Puppet 元件拖曳至畫布。
- 2 在右側的窗格上輸入 Puppet 內容。

內容	說明
Master	輸入要與此藍圖搭配使用的 Puppet 主要機器的名稱。
環境	選取 Puppet 主要機器的環境。

內容	說明
角色	選取要與此藍圖搭配使用的 Puppet 角色。
代理程式執行間隔	希望 Puppet 代理程式輪詢 Puppet 主要機器以取得組態詳細資料的頻率，該組態詳細資料將套用至與此藍圖相關的已部署虛擬機器。

3 按一下右窗格中的 [程式碼] 索引標籤，以檢視 Puppet 組態內容的 YAML 程式碼。

## 在 vRealize Automation Cloud Assembly 中設定 Ansible 開放原始碼整合

vRealize Automation Cloud Assembly 支援與 Ansible 開放原始碼組態管理整合。設定整合後，您可以將 Ansible 元件新增至新部署或現有部署中。

將 Ansible 開放原始碼與 vRealize Automation Cloud Assembly 整合時，您可以將其設定為佈建新機器時，按照指定順序執行一或多個 Ansible playbook，以自動執行組態管理。可以在藍圖中為部署指定所需的 playbook。

設定 Ansible 整合時，您必須指定 Ansible 開放原始碼主機，以及定義用於管理資源的相關資訊的詳細目錄檔案路徑。此外，您必須提供用於存取 Ansible 開放原始碼執行個體的名稱和密碼。稍後，在將 Ansible 元件新增至部署時，您可以更新連線以使用基於金鑰的驗證。

依預設，Ansible 會使用 ssh 連線至實體機器。如果您使用的是在藍圖中透過 osType Windows 內容指定的 Windows 機器，則 connection\_type 變數會自動設定為 winm。

Ansible 整合支援不使用 IP 位址的實體機器。對於在公有雲 (例如 AWS、Azure 和 GCP) 上佈建的機器，僅當機器連線至公用網路時，才會在已建立資源中的 address 內容中填入機器的公用 IP 位址。對於未連線至公用網路的機器，Ansible 整合會從連結至機器的網路中尋找 IP 位址。如果連結了多個網路，則 Ansible 整合會尋找具有最少 deviceIndex 的網路；即連結至機器的網路介面卡 (NIC) 的索引。如果未在藍圖中指定 deviceIndex 內容，則整合會使用第一個連結的網路。

如需有關在 vRealize Automation Cloud Assembly 中設定 Ansible 開放原始碼進行整合的更多詳細資料，請參閱[什麼是 vRealize Automation Cloud Assembly 中的組態管理](#)。

### 必要條件

- Ansible 控制機器必須使用 Ansible 2.6.0 版或更新版本。
- 使用者必須對 Ansible 詳細目錄檔案所在的目錄具有讀取/寫入權限。此外，使用者還必須具有詳細目錄檔案 (如果已存在) 的讀取/寫入權限。
- 如果您使用具有 sudo 選項的非 root 使用者，請確保 sudoers 檔案中已設定下列內容：

```
Defaults:user_name !requiretty
```

和

```
username ALL=(ALL) NOPASSD: ALL
```

- 確保已透過在 /etc/ansible/ansible.cfg 或 ~/.ansible.cfg 中設定 host\_key\_checking = False 停用主機金鑰檢查。

- 確保已透過將以下行新增至 `/etc/ansible/ansible.cfg` 或 `~/.ansible.cfg` 檔案設定保存庫密碼：

```
vault_password_file = /path/to/password_file
```

保存庫密碼檔案包含純文字格式的密碼，僅在藍圖或部署提供要在 ACM 和節點之間使用的使用者名稱和密碼組合時才會使用該密碼，如下列範例所示。

```
echo 'myStr0ng9@88w0rd' > ~/.ansible_vault_password.txt
echo 'ANSIBLE_VAULT_PASSWORD_FILE=~/.ansible_vault_password.txt' > ~/.profile      #
Instead of this way, you can also set it setting
'vault_password_file=~/.ansible_vault_password.txt' in either /etc/ansible/ansible.cfg or
~/.ansible.cfg
```

- 若要在嘗試執行 playbook 時避免主機金鑰失敗，建議您在 `/etc/ansible/ansible config` 中包含下列設定。

```
[paramiko_connection]
record_host_keys = False

[ssh_connection]
#ssh_args = -C -o ControlMaster=auto -o ControlPersist=60s
ssh_args = -o UserKnownHostsFile=/dev/null      # If you already have any
options set for ssh_args, just add the additional option shown here at the end.
```

## 程序

- 1 選取**基礎結構 > 連線 > 整合**，然後按一下**新增整合**。
- 2 按一下 [Ansible]。  
隨即出現 [Ansible 組態] 頁面。
- 3 輸入 Ansible 開放原始碼執行個體的主機名稱、詳細目錄檔案路徑和其他所需資訊。
- 4 按一下**驗證**來檢查整合。
- 5 按一下**新增**。

## 結果

Ansible 可用於藍圖。

## 後續步驟

將 Ansible 元件新增至所需藍圖。

- 1 在藍圖畫布頁面上，選取藍圖選項功能表上 [組態管理] 標題下的 Ansible，然後將 Ansible 元件拖曳至畫布。
- 2 使用右側面板設定適當的 Ansible 內容，例如，指定要執行的 playbook。

在 Ansible 中，使用者可以將變數指派給單一主機，並稍後在 playbook 中使用。Ansible 開放原始碼整合可讓您在藍圖中指定這些主機變數。hostVariables 內容必須採用正確的 YAML 格式 (如 Ansible 控制機器的預期)，且此內容將放置在下列位置：

```
parent_directory_of_inventory_file/host_vars/host_ip_address/vra_user_host_vars.yml
```

Ansible 詳細目錄檔案的預設位置在 Cloud Assembly 中的 [整合] 頁面上新增的 Ansible 帳戶中進行定義。Ansible 整合將不會驗證藍圖中的 hostVariable YAML 語法，但是，如果在格式或語法錯誤的情況下執行 playbook，Ansible 控制機器會擲回錯誤。

下列藍圖 YAML 程式碼片段顯示了 hostVariables 內容的使用範例。

```
Cloud_Ansible_1:
  type: Cloud.Ansible
  properties:
    host: '${resource.AnsibleLinuxVM.*}'
    osType: linux
    account: ansible-CAVA
    username: ${input.username}
    password: ${input.password}
    maxConnectionRetries: 20
    groups:
      - linux_vms
    playbooks:
      provision:
        - /root/ansible-playbooks/install_web_server.yml
    hostVariables: |
      message: Hello ${env.requestedBy}
      project: ${env.projectName}
```

## 如何在 vRealize Automation Cloud Assembly 中建立 Active Directory 整合

vRealize Automation Cloud Assembly 支援與 Active Directory 伺服器進行整合，以便在佈建虛擬機器之前，於 Active Directory 伺服器內指定的組織單位 (OU) 中現成建立電腦帳戶。

Active Directory 整合僅支援與 Active Directory 伺服器建立 LDAP 連線。

### 必要條件

- 如果您要設定 Active Directory 與 vCenter 內部部署的整合，則必須為 Active Directory 整合設定擴充性雲端 Proxy。選取**擴充性 > 活動 > 整合**，然後選擇**擴充性動作內部部署**。
- 如果您要在雲端中設定與 Active Directory 的整合，則必須擁有 Microsoft Azure 或 Amazon Web Services 帳戶。
- 您的 Active Directory 伺服器必須使用 LDAP 伺服器連線。
- 您必須為專案設定適當的雲端區域以及用於 Active Directory 整合的映像和類型模板對應。
- 必須預先建立 Active Directory 中所需的 OU，然後才能將 Active Directory 整合與專案相關聯。

## 程序

- 1 選取**基礎結構 > 連線 > 整合**，然後選取**新增整合**。
- 2 按一下 **Active Directory**。
- 3 在**摘要**索引標籤上，輸入適當的 LDAP 主機和環境名稱。
- 4 輸入 LDAP 伺服器名稱和密碼。
- 5 為 Active Directory 中所需的使用者和群組輸入適當的基本 DN。

---

**備註** 每個 Active Directory 整合只能指定一個 DN。

---

- 6 按一下**驗證**，以確保整合正常運作。
- 7 輸入此整合的名稱與說明。
- 8 按一下**儲存**。
- 9 按一下**專案**索引標籤，將專案新增至 Active Directory 整合。

在**新增專案**對話方塊中，您必須選取專案名稱和相對 DN，即在 [摘要] 索引標籤上指定的基本 DN 中存在的 DN。

- 10 按一下**儲存**。

## 結果

您現在可以將具有 Active Directory 整合的專案關聯至藍圖。使用此藍圖佈建機器時，它將預先暫存在指定的 Active Directory 和組織單位中。

## vRealize Automation Cloud Assembly 中的上線計劃是什麼

您可以使用工作負載上線計劃來識別已從目標區域或資料中心的雲端帳戶類型收集資料，但尚未由 vRealize Automation Cloud Assembly 專案管理的機器。

新增包含在 vRealize Automation Cloud Assembly 之外部署的機器的雲端帳戶時，這些機器不會由 Cloud Assembly 進行管理，直到您使其上線。使用上線計劃可將未受管理的機器引入 vRealize Automation Cloud Assembly 管理。建立計劃，使用機器填入計劃，然後執行計劃以匯入機器。透過使用上線計劃，您可以建立藍圖，也可以建立一或多個部署。

您可以在一個計劃中將一或多個未受管理的機器上線。您可以手動或使用篩選規則選取機器。篩選規則根據機器名稱、狀態、IP 位址和標籤等準則選取要上線的機器。

- 您可以在每小時的單一上線計劃內，將最多 3,500 部未受管理的機器上線。
- 您可以在每小時多個上線計劃內，同時將最多 17,000 部未受管理的機器上線。

適用於工作負載上線的機器列在**資源 > 機器**頁面上，此頁面是相對於特定的雲端帳戶類型和區域並在 [來源] 資料行中標示為 *Discovered*。僅列出已進行資料收集的機器。在機器上線後，它們將在 [來源] 資料行中顯示為 *Deployed*。

執行工作負載上線計劃的人員將被自動指派為機器擁有者。

## 上線範例

如需上線技巧的範例，請參閱[範例：將選取的機器做為 vRealize Automation Cloud Assembly 中的單一部署上線](#)和[範例：在 vRealize Automation Cloud Assembly 中使規則篩選的機器上線做為單獨部署](#)。

## 上線事件訂閱

執行計劃時，會建立 Deployment Onboarded 事件。使用 [擴充性] 索引標籤中的選項，您可以訂閱這些部署事件，並對其執行動作。

## 範例：將選取的機器做為 vRealize Automation Cloud Assembly 中的單一部署上線

在此範例中，您將兩個未受管理的機器做為單一 vRealize Automation Cloud Assembly 部署上線，並為計劃中的所有機器建立單一藍圖。

建立雲端帳戶時，將為與此雲端帳戶相關聯的所有機器收集資料，然後在[基礎結構 > 資源 > 機器](#)頁面中顯示這些機器。如果雲端帳戶包含在 vRealize Automation Cloud Assembly 外部部署的機器，您可以使用上線計劃以允許 vRealize Automation Cloud Assembly 管理機器部署。

### 必要條件

- 確認您具有必要的使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 檢閱 [vRealize Automation Cloud Assembly 中的上線計劃是什麼](#)。
- 建立和準備 vRealize Automation Cloud Assembly 專案。

此程序涉及基本 Wordpress 使用案例中的一些步驟。請參閱 [WordPress 使用案例](#)。

- 建立專案、新增使用者，以及指派專案中的使用者角色。請參閱 [WordPress 使用案例：建立專案](#)。
- 為專案建立 Amazon Web Services 雲端帳戶。請參閱 [WordPress 使用案例：新增雲端帳戶](#)。

此程序中的 Amazon Web Services 雲端帳戶包含雲端帳戶新增至 vRealize Automation Cloud Assembly 之前部署的機器，以及由 vRealize Automation Cloud Assembly 以外的其他應用程式部署的機器。

- 確認[機器](#)頁面包含要上線的機器。請參閱[機器資源](#)。

### 程序

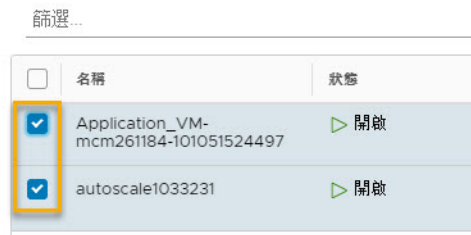
- 1 移至[基礎結構 > 上線](#)。
- 2 按一下[新增上線計劃](#)，然後輸入範例值。

設定	範例值
計劃名稱	VC-sqa-deployments
說明	OurCo-AWS 雲端帳戶之 AWS 機器的範例上線計劃

設定	範例值
雲端帳戶	OurCo-AWS
預設專案	WordPress

- 按一下**建立**。
- 在計劃的**部署索引標籤**上，按一下**選取機器**，選擇一或多個機器，然後按一下**確定**。

選取機器



- 選取**建立一個包含所有機器的部署**，然後按一下**建立**。
- 按一下新部署名稱旁的核取方塊，然後按一下**藍圖...**。
- 按一下以 **Cloud Assembly 格式建立藍圖**。



## 8 輸入藍圖名稱，然後按一下儲存。



**備註** 當您的上線計劃使用 vSphere 機器時，您必須在上線程序完成後編輯藍圖。此上線程序無法連結來源 vSphere 機器及其機器範本，且產生的藍圖將包含藍圖程式碼中的 `imageRef: "no image available"` 項目。在 `imageRef` 欄位中指定正確的範本名稱之前無法部署藍圖。若要在上線程序完成後更容易找到並更新藍圖，請在部署的藍圖組態頁面上使用藍圖名稱選項。記錄自動產生的藍圖名稱，或輸入並記錄您選擇的藍圖名稱。當上線程序完成後，找到並開啟藍圖，並將 `imageRef` 欄位中的 `"no image available"` 項目取代為正確的範本名稱。

## 9 按一下部署名稱核取方塊，按一下執行，然後在執行計劃頁面上再次按一下執行。

所選的 Amazon Web Services 機器將上線做為單一部署，並隨附一個藍圖。

## 10 透過按一下藍圖索引標籤，然後按一下藍圖名稱，開啟並檢查藍圖。

## 11 透過按一下部署索引標籤，然後按一下部署名稱，開啟並檢查部署。

## 範例：在 vRealize Automation Cloud Assembly 中使規則篩選的機器上線做為單獨部署

在此範例中，您將使用篩選規則將狀態為 [開啟] 且名稱以字母 BG 開頭的機器上線。您還將為計劃中的每個機器建立單獨的 vRealize Automation Cloud Assembly 藍圖和部署。

建立雲端帳戶時，將為與此雲端帳戶相關聯的所有機器收集資料，然後在**基礎結構 > 資源 > 機器**頁面中顯示這些機器。如果雲端帳戶包含在 vRealize Automation Cloud Assembly 外部部署的機器，您可以使用上線計劃以允許 vRealize Automation Cloud Assembly 管理機器部署。

### 必要條件

- 確認您具有必要的使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 檢閱 vRealize Automation Cloud Assembly 中的上線計劃是什麼。

- 建立並準備 vRealize Automation Cloud Assembly 專案，然後為其填入一或多個雲端帳戶。這包括引導式設定程序中的部分基本步驟。
  - 建立專案、新增使用者，以及指派專案中的使用者角色。請參閱 [WordPress 使用案例：建立專案](#)。
  - 在專案的指定區域中建立一或多個雲端帳戶。請參閱 [WordPress 使用案例：新增雲端帳戶](#)。
- 確認機器頁面包含要上線的機器。請參閱 [機器資源](#)。

## 程序

- 1 移至**基礎結構 > 上線**。
- 2 按一下**新增上線計劃**，然後輸入值。

設定	範例值
計劃名稱	ob_rules_1
說明	Machine onboarding with rules1
雲端帳戶	rs-aws
預設專案	rs-project

### 新的上線計劃



計劃名稱

說明

#### 必要條件

新增雲端帳戶，並為要上線的機器所在的計算資源建立雲端區域。  
建立具有至少一個使用者的專案，然後授與對雲端區域的專案存取權。

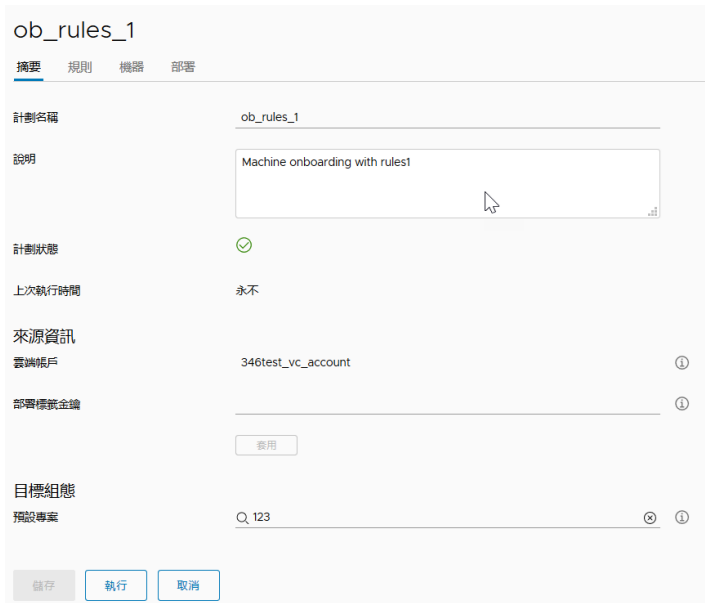
雲端帳戶

預設專案

取消

建立

### 3 按一下**建立**。



### 4 按一下**規則索引**標籤，然後按一下**新增規則**。

您可以建立一或多個規則，以根據特定機器特性選取要上線的一組機器。



### 5 輸入規則名稱，例如 **ob\_rules\_1**。

新增 規則

建立 篩選器的篩選規則將用於在此計劃中填入機器。

規則名稱 \*

### 6 新增篩選器以建置規則。

對於此範例，請使用**篩選器**下拉式功能表上的**狀態**和**名稱**篩選器來指定名稱包含 BG\* 且狀態為 On 的所有機器。

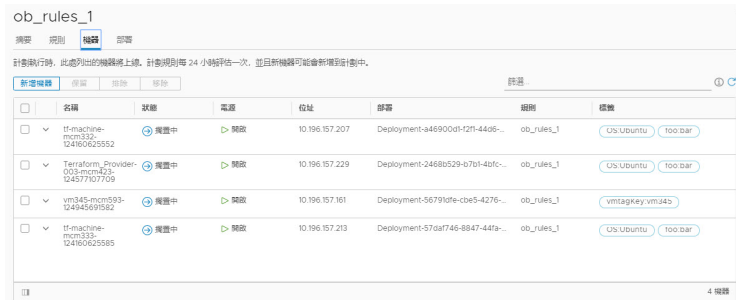


## 7 按一下儲存。

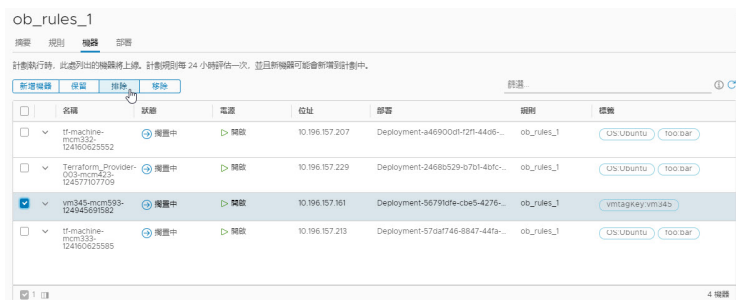
雖然您可以制定其他規則，但此範例僅使用單一規則。



## 8 按一下機器索引標籤。在此範例中，選取了 4 個機器，其中 3 個機器的名稱以字母 BG 開頭，還有一個機器的名稱中包含字母 BG。

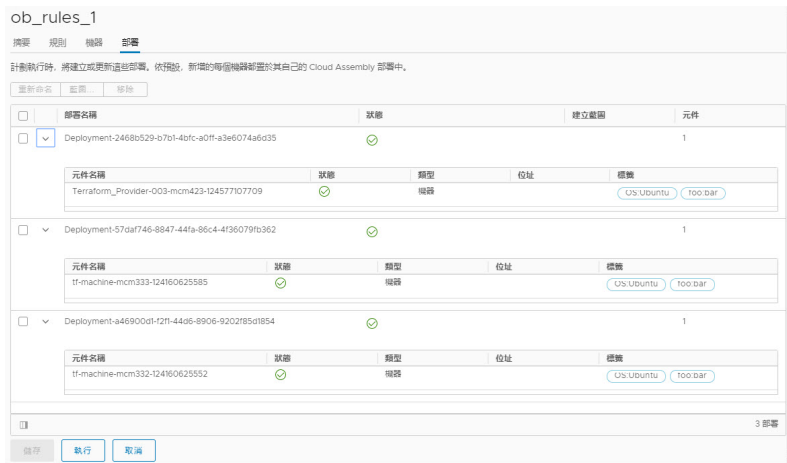


## 9 透過選取對應的核取方塊並按一下排除，來移除名稱不以 BG 開頭的機器。



## 10 按一下部署索引標籤。

名稱以字母 BG 開頭且 On 電源的 3 個機器已準備好進行部署。依預設，將為每個機器建立單獨的藍圖和部署。

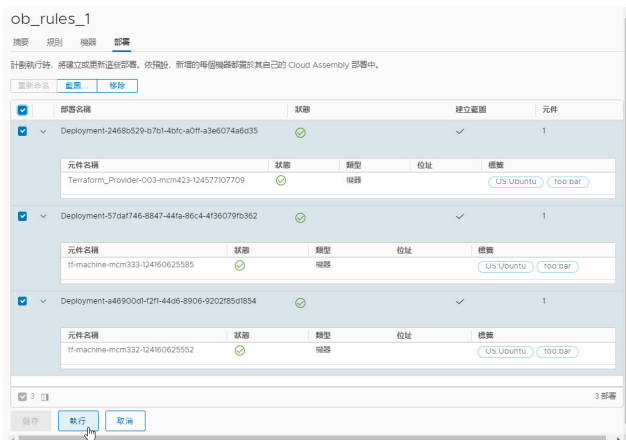


- 11 按一下三個部署名稱旁的核取方塊，然後依序按一下**藍圖**、以 Cloud Assembly 格式建立藍圖和儲存。



**備註** 當您的上線計劃使用 vSphere 機器時，您必須在上線程序完成後編輯藍圖。此上線程序無法連結來源 vSphere 機器及其機器範本，且產生的藍圖將包含藍圖程式碼中的 `imageRef: "no image available"` 項目。在 `imageRef` 欄位中指定正確的範本名稱之前無法部署藍圖。若要在上線程序完成後更容易找到並更新藍圖，請在部署的**藍圖組態**頁面上使用**藍圖名稱**選項。記錄自動產生的藍圖名稱，或輸入並記錄您選擇的藍圖名稱。當上線程序完成後，找到並開啟藍圖，並將 `imageRef` 欄位中的 `"no image available"` 項目取代為正確的範本名稱。

12 在部署頁面上，按一下三個部署名稱旁的核取方塊，然後按一下執行。



13 當系統提示您確認時，按一下執行將機器上線。



執行該計劃，並將機器引入 vRealize Automation Cloud Assembly 管理。將為每個機器建立單獨的藍圖和部署。

## vRealize Automation Cloud Assembly 環境的進階組態

您可以設定 vRealize Automation Cloud Assembly 環境，以進一步支援專案組態、整合及部署。

如需管理方法的相關資訊和其他資訊，例如使用使用者和記錄，以及加入或退出客戶經驗計劃，請參閱《[管理 vRealize Automation](#)》說明。

## 如何為 vRealize Automation 設定網際網路 Proxy 伺服器

對於不具有直接網際網路存取權的隔離網路上的 vRealize Automation 8.0.1 之前的安裝，您可以使用網際網路 Proxy 伺服器，以便允許透過 Proxy 功能存取網際網路。網際網路 Proxy 伺服器支援 HTTP 和 HTTPS。

若要透過 vRealize Automation 設定和使用公有雲提供者 (如 Amazon Web Services (AWS)、Microsoft Azure 和 Google Cloud Platform (GCP)) 以及外部整合點 (例如 IPAM、Ansible 和 Puppet)，您必須將網際網路 Proxy 伺服器設定為存取內部 vRealize Automation 網際網路 Proxy 伺服器。

vRealize Automation 包含與網際網路 Proxy 伺服器進行通訊的內部 Proxy 伺服器。如果您的 Proxy 伺服器已使用 `vracli proxy set ...` 命令進行設定，則此伺服器會與 Proxy 伺服器進行通訊。如果您尚未設定組織的網際網路 Proxy 伺服器，則 vRealize Automation 內部 Proxy 伺服器會嘗試直接連線至網際網路。

您可以使用提供的 `vracli` 命令列公用程式，將 vRealize Automation 設定為使用網際網路 Proxy 伺服器。有關如何使用 `vracli` API 的資訊，可透過在 `vracli` 命令列中使用 `--help` 引數取得，例如 `vracli proxy --help`。

存取網際網路 Proxy 伺服器時，需要使用 vRealize Automation 內建的以動作為基礎的擴充性 (ABX) 內部部署內嵌式控制。

---

**備註** 不支援透過網際網路 Proxy 存取 Workspace ONE Access (先前稱為 VMware Identity Manager)。您無法使用 `vracli set vidm` 命令透過網際網路 Proxy 伺服器存取 Workspace ONE Access。

---

內部 Proxy 伺服器需要 IPv4 做為其預設 IP 格式。不需要對 TLS (HTTPS) 憑證流量設定網際網路通訊協定限制、驗證或攔截式動作。

#### 必要條件

- 確認您擁有可在 vRealize Automation 網路中用作網際網路 Proxy 伺服器的現有 HTTP 或 HTTPS 伺服器，以將傳出流量傳遞至外部站台。必須針對 IPv4 設定連線。
- 確認目標網際網路 Proxy 伺服器已設定為支援 IPv4 (而非 IPv6) 做為其預設 IP 格式。
- 如果網際網路 Proxy 伺服器使用 TLS，並且需要與其用戶端建立 HTTPS 連線，您必須在設定 Proxy 組態之前，使用下列其中一個命令匯入伺服器憑證。

- `vracli certificate proxy --set path_to_proxy_certificate.pem`
- `vracli certificate proxy --set stdin`

使用 `stdin` 參數進行互動式輸入。

#### 程序

- 1 針對 Kubernetes 所使用的網蔴或容器建立 Proxy 組態。在此範例中，使用 HTTP 配置存取 Proxy 伺服器。

```
vracli proxy set --host http://proxy.vmware.com:3128
```

- 2 顯示 Proxy 組態。

```
vracli proxy show
```

結果類似下列內容：

```
{
  "enabled": true,
  "host": "10.244.4.51",
  "java-proxy-exclude": "*.local|*.localdomain|localhost|10.244.*|192.168.*|172.16.*|
kubernetes|sc2-rdops-vm06-dhcp-198-120.eng.vmware.com|10.192.204.9|*.eng.vmware.com|sc2-
rdops-vm06-dhcp-204-9.eng.vmware.com|10.192.213.146|sc2-rdops-vm06-
```

```

dhcp-213-146.eng.vmware.com|10.192.213.151|sc2-rdops-vm06-dhcp-213-151.eng.vmware.com",
  "java-user": null,
  "password": null,
  "port": 3128,
  "proxy-exclude":
".local,.localdomain,localhost,10.244.,192.168.,172.16.,kubernetes,sc2-rdops-vm06-
dhcp-198-120.eng.vmware.com,10.192.204.9,.eng.vmware.com,sc2-rdops-vm06-
dhcp-204-9.eng.vmware.com,10.192.213.146,sc2-rdops-vm06-
dhcp-213-146.eng.vmware.com,10.192.213.151,sc2-rdops-vm06-dhcp-213-151.eng.vmware.com",
  "scheme": "http",
  "upstream_proxy_host": null,
  "upstream_proxy_password_encoded": "",
  "upstream_proxy_port": null,
  "upstream_proxy_user_encoded": "",
  "user": null,
  "internal.proxy.config": "dns_v4_first on \nhttp_port 0.0.0.0:3128\nlogformat squid
%ts.%03tu %6tr %>a %Ss/%03>Hs %<st %rm %ru %[un %Sh/%<a %mt\naccess_log stdio:/tmp/logger
squid\ncoredump_dir /\ncache deny all \nappend_domain .prelude.svc.cluster.local\nacl
mylan src 10.0.0.0/8\nacl mylan src 127.0.0.0/8\nacl mylan src 192.168.3.0/24\nacl proxy-
exclude dstdomain .local\nacl proxy-exclude dstdomain .localdomain\nacl proxy-exclude
dstdomain localhost\nacl proxy-exclude dstdomain 10.244.\n\nacl proxy-exclude dstdomain
192.168.\n\nacl proxy-exclude dstdomain 172.16.\n\nacl proxy-exclude dstdomain kubernetes\nacl
proxy-exclude dstdomain 10.192.204.9\nacl proxy-exclude dstdomain .eng.vmware.com\nacl
proxy-exclude dstdomain 10.192.213.146\nacl proxy-exclude dstdomain
10.192.213.151\nalways_direct allow proxy-exclude\nhttp_access allow mylan\nhttp_access
deny all\n# End autogen configuration\n",
  "internal.proxy.config.type": "default"
}

```

**備註** 如果已為組織設定網際網路 Proxy 伺服器，則上述範例中會顯示

"internal.proxy.config.type": "non-default"，而不是 'default'。出於安全性考慮，不會顯示密碼。

**備註** 如果您使用 `-proxy-exclude` 參數，則必須編輯預設值。例如，如果您想要新增 `acme.com` 做為無法使用網際網路 Proxy 伺服器存取的網域，請使用下列步驟：

- a 輸入 `vraccli proxy default-no-proxy` 以取得預設 `proxy-exclude` 設定。這是自動產生的網域和網路的清單。
- b 編輯值以新增 `.acme.com`。
- c 輸入 `vraccli proxy set .... --proxy-exclude ...` 以更新組態設定。
- d 執行 `/opt/scripts/deploy.sh` 命令以重新部署環境。



- 3 (選擇性) 將 DNS 網域、FQDN 和 IP 位址排除在網際網路 Proxy 伺服器的存取範圍之外。

一律使用 `parameter --proxy-exclude` 修改 `proxy-exclude` 變數的預設值。若要新增網域 `exclude.vmware.com`，請先使用 `vrali proxy show` 命令，然後複製 `proxy-exclude` 變數，並使用 `vracli proxy set ...` 命令新增網域值，如下所示：

```
vracli proxy set --host http://proxy.vmware.com:3128 --proxy-exclude
"exclude.vmware.com,docker-
registry.prelude.svc.cluster.local,localhost,.local,.cluster.local,10.244.,192.,172.16.,sc-
rdops-vm11-dhcp-75-38.eng.vmware.com,10.161.75.38,.eng.vmware.com"
```

**備註** 將元素新增至 `proxy-exclude`，而不是取代值。如果您刪除 `proxy-exclude` 預設值，vRealize Automation 將無法正常運作。如果發生此情況，請刪除 Proxy 組態並重新啟動。

- 4 使用 `vracli proxy set ...` 命令設定網際網路 Proxy 伺服器後，您可以使用 `vracli proxy apply` 命令更新網際網路 Proxy 伺服器組態，並將最新的 Proxy 設定設為作用中狀態。
- 5 如果您尚未執行此操作，請透過執行下列命令來啟動指令碼變更：

```
/opt/scripts/deploy.sh
```

- 6 (選擇性) 如有需要，請將 Proxy 伺服器設定為支援連接埠 22 上的外部存取權。

若要支援 Puppet 和 Ansible 等整合，Proxy 伺服器必須允許連接埠 22 存取相關主機。

## 範例：範例 Squid 組態

相對於步驟 1，如果您設定的是 Squid Proxy，則可以透過將其調整為下列範例來調整 `/etc/squid/squid.conf` 中的組態：

```
acl localnet src 192.168.11.0/24

acl SSL_ports port 443

acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT

http_access allow !Safe_ports
http_access allow CONNECT !SSL_ports
http_access allow localnet

http_port 0.0.0.0:3128

maximum_object_size 5 GB
cache_dir ufs /var/spool/squid 20000 16 256
```

```

coredump_dir /var/spool/squid
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern (Release|Packages(.gz)*)$ 0 20% 2880
refresh_pattern . 0 20% 4320

client_persistent_connections on
server_persistent_connections on

```

## 如何在 vRealize Automation 中使用 cloud-init 或 cloudbase-init 設定 Windows 範本

您可以將 vRealize Automation 環境設定為支援 cloud-init 和 cloudbase-init 指令碼，以便在 Windows 部署中建立和使用自訂映像。

在映像對應和藍圖程式碼中支援雲端組態指令碼。雲端組態指令碼遵守 cloud-init 和 cloudbase-init 規則與格式。如需 cloud-init 和 cloudbase-init 的相關資訊，請參閱 <https://cloudbase.it/cloudbase-init>。

**備註** 如需針對 Windows 設定 cloud-init 或 cloudbase-init 的相關資訊，請參閱下列 VMware 部落格：

- [Windows Cloud-Init solution](#) 部落格文章
- [Windows guest initialization with Cloudbase-Init in vCenter](#) 部落格文章

如需為 Linux 設定 cloud-init 的相關資訊，請參閱部落格文章〈[為 vSphere 建立 vRealize Automation Cloud Ready Ubuntu 範本](#)〉。

如需在 vRealize Automation 中使用 cloud-init 與雲端組態指令碼的相關資訊，請參閱：

- [進一步瞭解 vRealize Automation Cloud Assembly 中的映像對應](#)
- [如何在 vRealize Automation Cloud Assembly 藍圖中自動初始化機器](#)

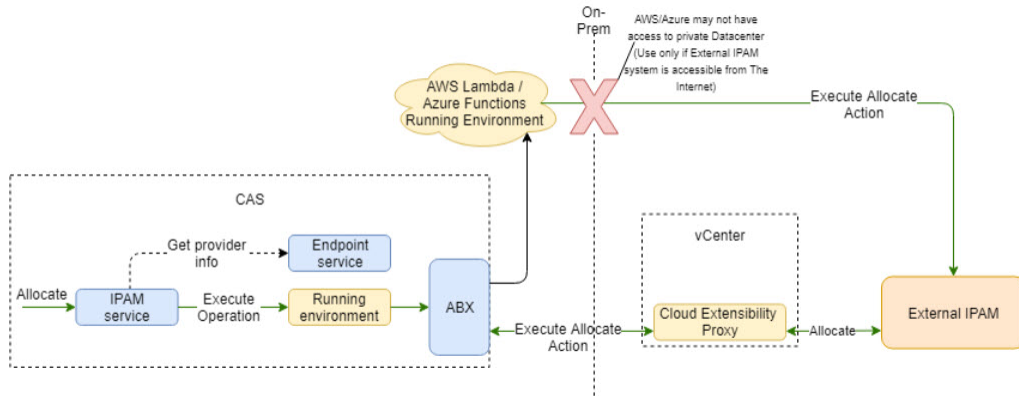
另請參閱這些 cloud-init 和 cloudbase-init 廠商頁面：

- <https://cloud-init.io>
- <https://cloudinit.readthedocs.io/en/latest/index.html>
- <https://cloudinit.readthedocs.io/en/latest/topics/examples.html#yaml-examples>
- <https://cloudbase.it/cloudbase-init>
- <https://cloudbase-init.readthedocs.io/en/latest/services.html>
- <https://cloudbase-init.readthedocs.io/en/latest/userdata.html#userdata>

## 如何使用 IPAM SDK 為 vRealize Automation 建立提供者特定的外部 IPAM 整合套件

外部 IPAM 廠商和合作夥伴可以下載並使用 IPAM SDK 來建立 IPAM 整合套件，以允許 vRealize Automation 支援其提供者特定的 IPAM 解決方案。

有關使用提供的 IPAM SDK 為 vRealize Automation 建立和部署自訂 IPAM 整合套件的程序，請參閱《為 VMware Cloud Assembly 建立和部署提供者特定的 IPAM 整合套件》文件。如文件中所述，您可以從 [VMware Solutions Exchange Marketplace](#) 下載 VMware vRealize Automation 第三方 IPAM SDK。



在花時間使用 IPAM SDK 建立廠商特定的 IPAM 整合套件之前，請先檢查 vRealize Automation 是否已存在一個整合套件。您可以在 IPAM 提供者的網站上、在 [VMware Solutions Exchange Marketplace](#) 以及從 vRealize Automation 市集索引標籤中檢查提供者特定的 IPAM 整合套件。

雖然 [提供者特定的外部 IPAM 整合使用案例](#) 範例特定於廠商，但也包含有用的參考資訊。

# vRealize Automation Cloud Assembly 使用案例

# 3

這些使用案例顯示了在 vRealize Automation Cloud Assembly 中建立資源基礎結構、進行設計並將應用程式部署到該基礎結構的範例。

這些使用案例僅提供範例值。您自己的環境結構和命名慣例會有所不同。

本章節討論下列主題：

- [WordPress 使用案例](#)
- [VMware Cloud on AWS 使用案例](#)
- [提供者特定的外部 IPAM 整合使用案例](#)

## WordPress 使用案例

此端對端 vRealize Automation Cloud Assembly 使用案例顯示建立基礎結構並將 WordPress 站台部署到該基礎結構的範例。

請查看連續設定，來瞭解可使 WordPress 站台完成的程序。

請注意，您看到的值只是使用案例範例而已。不能在您的環境中按原樣使用這些值。

請考量您可以在何處從範例值替換或推斷，以符合您自己的雲端基礎結構和部署需求。

### 程序

#### 1 [WordPress 使用案例：建立基礎結構](#)

做為雲端管理員，您首先需要設定一些資源，工程人員可以稍後在其中開發、測試並將 WordPress 站台放入生產環境。

#### 2 [WordPress 使用案例：建立專案](#)

專案可讓使用者可以佈建，並設定可能的佈建量。

#### 3 [WordPress 使用案例：建立和擴充藍圖](#)

做為開發人員，您可以採用通用 vRealize Automation Cloud Assembly 藍圖的形式定義 WordPress 站台，以便可以部署至任何雲端廠商。

## WordPress 使用案例：建立基礎結構

做為雲端管理員，您首先需要設定一些資源，工程人員可以稍後在其中開發、測試並將 WordPress 站台放入生產環境。

基礎結構包括雲端目標，以及 WordPress 站台將需要的機器、網路和儲存區相關的定義。

## 程序

### 1 WordPress 使用案例：新增雲端帳戶

在此步驟中，雲端管理員將新增兩個雲端帳戶。專案預期為在 AWS 上執行開發和測試工作，並在 Azure 上移到生產。

### 2 WordPress 使用案例：新增雲端區域

在此步驟中，雲端管理員將新增三個雲端區域，分別用於開發、測試和生產。

### 3 WordPress 使用案例：新增類型模板對應

在此步驟中，雲端管理員將新增類型模板對應至根據部署容量需求可能會有所不同的帳戶。

### 4 WordPress 使用案例：新增映像對應

在此步驟中，雲端管理員將為 Ubuntu (WordPress 伺服器及其 MySQL 資料庫伺服器的主機) 新增映像對應。

### 5 WordPress 使用案例：新增網路設定檔

在此步驟中，雲端管理員將網路設定檔新增到每個雲端區域。

### 6 WordPress 使用案例：新增儲存區設定檔

在此步驟中，雲端管理員將儲存區設定檔新增到每個雲端區域。

## WordPress 使用案例：新增雲端帳戶

在此步驟中，雲端管理員將新增兩個雲端帳戶。專案預期為在 AWS 上執行開發和測試工作，並在 Azure 上移到生產。

## 程序

1 移至**基礎結構 > 連線 > 雲端帳戶**。

2 按一下**新增雲端帳戶**，選取 Amazon Web Services，然後輸入值。

設定	範例值
存取金鑰識別碼	R5SDR3PXVV2ZW8B7YNSM
密碼存取金鑰	SZXAIXU4UHNQAQ1E156S
名稱	OurCo-AWS
說明	WordPress
功能	cloud:aws

請注意，所有值只是使用案例範例而已。您的帳戶特性會有所不同。

3 若要驗證認證，請按一下**驗證**。

4 按一下**新增**。

- 編輯新增的帳戶組態，並允許佈建到 us-east-1 和 us-west-2 區域。
- 按一下**新增雲端帳戶**，選取 Microsoft Azure，然後輸入值。

設定	範例值
訂閱識別碼	ef2avpf-dfdv-zxlugui1i-g4h0-i8ep2jwp4c9arbfe
承租人識別碼	dso9wv3-4zgc-5nrcy5h3m-4skf-nnovp40wfxsro22r
用戶端應用程式識別碼	bg224oq-3ptp-mbhi6aa05-q511-uf1yjr2sttyik6bs
用戶端應用程式秘密金鑰	7uqxi57-0wtn-kymgf9wcj-t2l7-e52e4nu5fig4pmdd
名稱	OurCo-Azure
說明	WordPress
功能	cloud:az

- 若要驗證認證，請按一下**驗證**。
- 按一下**新增**。
- 編輯新增的帳戶組態，並允許佈建到美國東部區域。

#### 後續步驟

新增專案將在其中部署 WordPress 站台的雲端區域。請參閱 [WordPress 使用案例：新增雲端區域](#)。

### WordPress 使用案例：新增雲端區域

在此步驟中，雲端管理員將新增三個雲端區域，分別用於開發、測試和生產。

雲端區域是一些資源，專案將部署機器、網路和儲存區至其中，以支援 WordPress 站台。

#### 必要條件

新增雲端帳戶。請參閱 [WordPress 使用案例：新增雲端帳戶](#)。

#### 程序

- 移至**基礎結構 > 設定 > 雲端區域**。
- 按一下**新增雲端區域**，然後輸入開發環境的值。

雲端區域設定	範例值
帳戶/區域	OurCo-AWS/us-east-1
名稱	OurCo-AWS-US-East
說明	WordPress
放置原則	預設值
功能標籤	env:dev

請注意，所有值只是使用案例範例而已。您的區域特性會有所不同。

- 3 按一下**計算**，並確認有您預期的區域。
- 4 按一下**建立**。
- 5 使用測試和生產環境的值，重複此程序兩次。

雲端區域設定	範例值
帳戶/區域	OurCo-AWS/us-west-2
名稱	OurCo-AWS-US-West
說明	WordPress
放置原則	預設值
功能標籤	env:test

雲端區域設定	範例值
帳戶/區域	OurCo-Azure/美國東部
名稱	OurCo-Azure-East-US
說明	WordPress
放置原則	預設值
功能標籤	env:prod

#### 後續步驟

透過新增類型模板對應，考慮不同大小的機器部署。請參閱 [WordPress 使用案例：新增類型模板對應](#)。

### WordPress 使用案例：新增類型模板對應

在此步驟中，雲端管理員將新增類型模板對應至根據部署容量需求可能會有所不同的帳戶。

類型模板對應通俗稱為「調整 T 恤衫大小」。

#### 必要條件

新增雲端區域。請參閱 [WordPress 使用案例：新增雲端區域](#)。

#### 程序

- 1 移至**基礎結構 > 設定 > 類型模板對應**。每個雲端區域必須允許小型、中型和大型類型模板。

## 2 按一下**新類型模板對應**，然後輸入開發雲端區域的值。

設定	範例值
類型模板名稱	小型
帳戶/區域 值	OurCo-AWS/us-east-1 t2.micro
帳戶/區域 值	OurCo-AWS/us-west-2 t2.micro
帳戶/區域 值	OurCo-Azure/美國東部 Standard_A0

請注意，所有值只是使用案例範例而已。您的類型模板會有所不同。

## 3 按一下**建立**。

## 4 使用中型和大型類型模板的值，重複此程序兩次。

設定	範例值
類型模板名稱	中型
帳戶/區域 值	OurCo-AWS/us-east-1 t2.medium
帳戶/區域 值	OurCo-AWS/us-west-2 t2.medium
帳戶/區域 值	OurCo-Azure/美國東部 Standard_A3

設定	範例值
類型模板名稱	大型
帳戶/區域 值	OurCo-AWS/us-east-1 t2.large
帳戶/區域 值	OurCo-AWS/us-west-2 t2.large
帳戶/區域 值	OurCo-Azure/美國東部 Standard_A7

### 後續步驟

透過新增映像對應，對作業系統進行規劃。請參閱 [WordPress 使用案例：新增映像對應](#)。

## WordPress 使用案例：新增映像對應

在此步驟中，雲端管理員將為 Ubuntu (WordPress 伺服器及其 MySQL 資料庫伺服器的主機) 新增映像對應。

每個雲端區域需要一個 Ubuntu 映像對應。



**必要條件**

新增雲端區域。請參閱 [WordPress 使用案例：新增雲端區域](#)。

**程序**

- 1 移至**基礎結構 > 設定 > 映像對應**。
- 2 按一下**新映像對應**，然後輸入 Ubuntu 伺服器的值。

設定	範例值
映像名稱	ubuntu-16
帳戶/區域 值	OurCo-AWS/us-east-1 ubuntu-16.04-server-cloudimg-amd64
帳戶/區域 值	OurCo-AWS/us-west-2 ubuntu-16.04-server-cloudimg-amd64
帳戶/區域 值	OurCo-Azure/美國東部 azul-zulu-ubuntu-1604-923eng

請注意，所有值只是使用案例範例而已。您的映像會有所不同。

- 3 按一下**建立**。

**後續步驟**

新增網路。請參閱 [WordPress 使用案例：新增網路設定檔](#)。

**WordPress 使用案例：新增網路設定檔**

在此步驟中，雲端管理員將網路設定檔新增到每個雲端區域。

在每個設定檔中，管理員會為 WordPress 機器新增網路，並新增位於最終負載平衡器另一端的另一個網路。第二個網路將是使用者最終透過其連線的網路。

**必要條件**

新增雲端區域。請參閱 [WordPress 使用案例：新增雲端區域](#)。

**程序**

- 1 移至**基礎結構 > 設定 > 網路設定檔**。
- 2 按一下**新網路設定檔**，然後為開發雲端區域建立設定檔。

網路設定檔設定	範例值
帳戶/區域	OurCo-AWS/us-east-1
名稱	devnets
說明	WordPress
功能標籤	env:dev

- 3 按一下**網路**，然後按一下**新增網路**。
- 4 選取 wpnet、appnet-public，然後按一下**新增**。  
請注意，所有值只是使用案例範例而已。您的網路名稱會有所不同。
- 5 按一下**建立**。  
此 Wordpress 範例不需要指定網路原則或網路安全性設定。
- 6 重複此程序兩次，為 Wordpress 範例測試和生產雲端區域建立網路設定檔。在每種情況下，新增 wpnet 和 appnet-public 網路。

網路設定檔設定	範例值
帳戶/區域	OurCo-AWS/us-west-2
名稱	testnets
說明	WordPress
功能標籤	env:test

網路設定檔設定	值
帳戶/區域	OurCo-Azure/美國東部
名稱	prodnets
說明	WordPress
功能標籤	env:prod

#### 後續步驟

新增儲存區。請參閱 [WordPress 使用案例：新增儲存區設定檔](#)。

### WordPress 使用案例：新增儲存區設定檔

在此步驟中，雲端管理員將儲存區設定檔新增到每個雲端區域。

管理員會將快速儲存區放置在生產區域，且將一般儲存區放置在開發和測試區域。

#### 必要條件

新增雲端區域。請參閱 [WordPress 使用案例：新增雲端區域](#)。

#### 程序

- 1 移至**基礎結構 > 設定 > 儲存區設定檔**。
- 2 按一下**新儲存區設定檔**，然後為開發雲端區域建立設定檔。  
選取帳戶/區域後，會顯示其他欄位。

儲存區設定檔設定	範例值
帳戶/區域	OurCo-AWS/us-east-1
名稱	OurCo-AWS-US-East-Disk
說明	WordPress
裝置類型	EBS
磁碟區類型	一般用途 SSD
功能標籤	使用量:一般

請注意，所有值只是使用案例範例而已。

- 按一下**建立**。
- 重複此程序，以建立測試雲端區域的設定檔。

儲存區設定檔設定	範例值
帳戶/區域	OurCo-AWS/us-west-2
名稱	OurCo-AWS-US-West-Disk
說明	WordPress
裝置類型	EBS
磁碟區類型	一般用途 SSD
功能標籤	使用量:一般

- 重複此程序，以建立生產雲端區域的設定檔，因為此區域是 Azure 區域，所以具有不同的設定。

儲存區設定檔設定	範例值
帳戶/區域	OurCo-Azure/美國東部
名稱	OurCo-Azure-East-US-Disk
說明	WordPress
儲存區類型	受管理的磁碟
磁碟類型	Premium LRS
作業系統磁碟快取	唯讀
資料磁碟快取	唯讀
功能標籤	usage:fast

## 後續步驟

建立專案以識別使用者和定義佈建設定。請參閱 [WordPress 使用案例：建立專案](#)。

## WordPress 使用案例：建立專案

專案可讓使用者可以佈建，並設定可能的佈建量。

專案定義使用者和佈建設定。

- 使用者及其角色權限層級
- 佈建至雲端區域時部署的優先順序
- 每個雲端區域的部署執行個體數目上限

### 必要條件

新增雲端區域。請參閱 [WordPress 使用案例：新增雲端區域](#)。

### 程序

- 1 移至**基礎結構 > 管理 > 專案**。
- 2 按一下**新專案**，然後輸入名稱 WordPress。
- 3 按一下**使用者**，然後按一下**新增使用者**。
- 4 為使用者新增電子郵件地址和角色。

若要成功新增使用者，VMware Cloud Services 管理員必須已為此使用者啟用 vRealize Automation Cloud Assembly 的存取權限。

請注意，此處顯示的位址只是使用案例範例而已。

- chris.ladd@ourco.com，成員
  - kerry.mott@ourco.com，成員
  - pat.tubb@ourco.com，管理員
- 5 按一下**佈建**，然後按一下**新增雲端區域**。
  - 6 新增使用者可以部署到其中的雲端區域。

專案雲端區域設定	範例值
雲端區域	OurCo-AWS-US-East
佈建優先順序	1
執行個體限制	5
雲端區域	OurCo-AWS-US-West
佈建優先順序	1
執行個體限制	5
雲端區域	OurCo-Azure-East-US
佈建優先順序	0
執行個體限制	1

- 7 按一下**建立**。
- 8 移至**基礎結構 > 設定 > 雲端區域**，然後開啟在 [WordPress 使用案例：新增雲端區域](#) 中建立的區域。
- 9 按一下**專案**，然後確認 WordPress 專案可以佈建到該區域。
- 10 檢查在 [WordPress 使用案例：新增雲端區域](#) 中建立的其他區域。

#### 後續步驟

建立基本藍圖。

## WordPress 使用案例：建立和擴充藍圖

做為開發人員，您可以採用通用 vRealize Automation Cloud Assembly 藍圖的形式定義 WordPress 站台，以便可以部署至任何雲端廠商。

使用案例藍圖包含一個 WordPress 應用程式伺服器、一個 MySQL 資料庫伺服器和可部署至 AWS、Azure 或以 vSphere 為基礎的雲端的支援元件。藍圖從幾個元件開始，然後隨著您修改現有元件和新增更多元件不斷增長。

[WordPress 使用案例：建立基礎結構](#) 中的範例包含由雲端管理員設定的基礎結構：

- 兩個雲端帳戶，AWS 和 Azure。
- 三個雲端區域環境：
  - 開發 - OurCo-AWS-US-East
  - 測試 - OurCo-AWS-US-West
  - 生產 - OurCo-Azure-East-US
- 針對每個區域具有小型、中型和大型計算資源的類型模板對應。
- 每個區域中設定了 Ubuntu 16 的映像對應。
- 針對每個區域具有內部和外部子網路的網路設定檔：devnets、testnets、prodnets。
- 支援封存磁碟的儲存區、用於開發和測試的一般儲存區、用於生產的快速儲存區。
- WordPress 專案包含全部三個雲端區域環境，以及可以嘗試此使用案例的使用者。

#### 必要條件

自行熟悉您的基礎結構值。例如，此使用案例範例將 AWS 用於開發和測試，而將 Azure 用於生產。建立您自己的藍圖時，可以取代為自己的值 (通常是由雲端管理員設定)。

#### 程序

##### 1 [WordPress 使用案例：建立基本藍圖](#)

做為開發人員，您可以從只包含最少 WordPress 元件 (例如，僅包含一個應用程式伺服器) 的 vRealize Automation Cloud Assembly 藍圖開始。

## 2 WordPress 使用案例：測試基本藍圖

在開發期間，您通常可以從基礎元件開始來建置 vRealize Automation Cloud Assembly 藍圖，然後隨著藍圖增長進行部署和測試。

## 3 WordPress 使用案例：擴充藍圖

建立並測試基本 vRealize Automation Cloud Assembly 藍圖後，將其擴充到可部署至開發、測試與最終生產環境的多階層應用程式。

## WordPress 使用案例：建立基本藍圖

做為開發人員，您可以從只包含最少 WordPress 元件 (例如，僅包含一個應用程式伺服器) 的 vRealize Automation Cloud Assembly 藍圖開始。

vRealize Automation Cloud Assembly 是基礎結構即代碼工具。您可以將元件拖曳至設計畫布以開始。然後，使用畫布右側的代碼編輯器填寫詳細資料。

透過代碼編輯器，您可以直接輸入、剪下和貼上代碼。如果不願意編輯代碼，您可以在畫布中選取資源，按一下代碼編輯器的內容索引標籤，然後在此輸入值。您輸入的值會出現在代碼中，如同您直接將其輸入一樣。

### 必要條件

自行熟悉您的基礎結構。此處顯示的範例使用 [WordPress 使用案例：建立基礎結構](#) 中的基礎結構值，但您可以將其取代為自己的基礎結構值。

### 程序

- 1 移至 **藍圖**，然後按一下 **新增**。
- 2 將藍圖命名為 **Wordpress-BP**。
- 3 選取 **WordPress 專案**，然後按一下 **建立**。
- 4 從藍圖設計頁面左側的元件中，將兩個非雲端機器拖曳至畫布。

這兩個機器分別用作 WordPress 應用程式伺服器 (WebTier) 和 MySQL 資料庫伺服器 (DBTier)。

- 5 在右側，編輯機器 YAML 代碼以新增名稱、映像、類型模板和限制標籤：

```
resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: 'ubuntu-16'
      flavor: 'small'
      constraints: - tag: env:dev
  WebTier:
    type: Cloud.Machine
    properties:
```

```

name: wordpress
image: 'ubuntu-16'
flavor: 'small'
constraints: - tag: env:dev

```

## 6 將非雲端網路拖曳至畫布，並編輯其代碼：

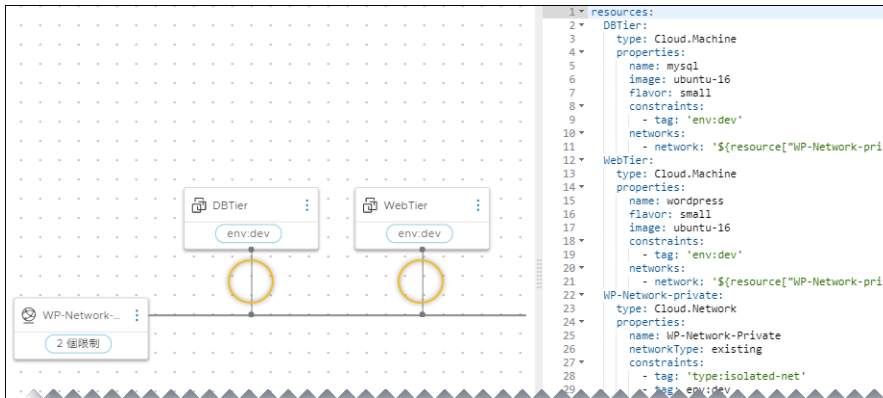
```

WP-Network-Private:
  type: Cloud.Network
  properties:
    name: WP-Network-Private
    networkType: existing
    constraints: - tag: 'type:isolated-net' - tag: 'env:dev'

```

## 7 將兩個機器連線至該網路：

在與網路區塊接觸的行處按住滑鼠按鈕，拖曳到機器區塊，然後放開滑鼠。



在編輯器中，您會發現網路代碼將新增到兩個機器：

```

resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: 'ubuntu-16'
      flavor: 'small'
      constraints:
        - tag: env:dev
      networks: - network: '${resource["WP-Network-Private"].id}'
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: 'ubuntu-16'
      flavor: 'small'

```

```
constraints:
  - tag: env:dev
  networks: - network: '${resource["WP-Network-Private"].id}'
```

## 8 新增使用者輸入提示。

在某些位置，使用案例基礎結構已設定用於多個選項。例如：

- 用於開發、測試和生產的雲端區域環境
- 用於小型、中型和大型機器的類型模板對應
- 適用於一般使用和快速使用的儲存區磁碟速度



您可以直接在藍圖中設定特定選項，但更好的方法是讓使用者在部署藍圖時選取選項。透過提示使用者輸入，您可以建立具有多種部署方式的單一藍圖，而不需要建立多個硬式編碼藍圖。

- a 在代碼中建立 `inputs` 區段，以便使用者可以在部署時選取機器大小和目標環境。定義可選取的值：

```
inputs:
  env:
    type: string
    enum:
      - 'env:dev'
      - 'env:prod'
      - 'env:test'
    default: 'env:dev'
    title: Environment
    description: Target Environment
  size:
    type: string
    enum:
      - small
      - medium
      - large
    description: Size of Nodes
    title: Tier Machine Size
```

- b 在代碼的 `resources` 區段中，新增 `${input.input-name}` 代碼以提示使用者選取：

```
resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: 'ubuntu-16'
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
      networks:
        - network: '${resource["WP-Network-Private"].id}'
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: 'ubuntu-16'
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
      networks:
        - network: '${resource["WP-Network-Private"].id}'
  WP-Network-Private:
    type: Cloud.Network
    properties:
      name: WP-Network-Private
```

```
networkType: existing
constraints:
  - tag: 'type:isolated-net'
  - tag: '${input.env}'
```

- 9 最後，使用下列範例增強 WebTier 和 DBTier 代碼。WP-Network-Private 代碼不需要其他變更。  
請注意，增強包括對資料庫伺服器、資料庫和部署時 cloudConfig 初始化指令碼的登入存取。

元件	範例
其他 DBTier 輸入	<pre> username:   type: string   minLength: 4   maxLength: 20   pattern: '[a-z]+'   title: Database Username   description: Database Username userpassword:   type: string   pattern: '[a-z0-9A-Z@#]+\$'   encrypted: true   title: Database Password   description: Database Password databaseDiskSize:   type: number   default: 4   maximum: 10   title: MySQL Data Disk Size   description: Database Disk Size </pre>
DBTier 資源	<pre> DBTier:   type: Cloud.Machine   properties:     name: mysql     image: ubuntu-16     flavor: '\${input.size}'     constraints:       - tag: '\${input.env}'     networks:       - network: '\${resource["WP-Network-Private"].id}'         assignPublicIpAddress: true   remoteAccess:     authentication: usernamePassword     username: '\${input.username}'     password: '\${input.userpassword}'   cloudConfig:       #cloud-config     repo_update: true     repo_upgrade: all      packages:       - mysql-server      runcmd:       - sed -e '/bind-address/ s/^#*#/' -i /etc/mysql/mysql.conf.d/ mysqlld.cnf       - service mysql restart       - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';"       - mysql -e "FLUSH PRIVILEGES;"   attachedDisks: [] </pre>
WebTier 資源	<pre> WebTier:   type: Cloud.Machine   properties:     name: wordpress     flavor: '\${input.size}' </pre>

元件	範例
	<pre> image: ubuntu-16 constraints:   - tag: '\${input.env}' networks:   - network: '\${resource["WP-Network-Private"].id}'     assignPublicIpAddress: true cloudConfig:     #cloud-config   repo_update: true   repo_upgrade: all  packages:   - apache2   - php   - php-mysql   - libapache2-mod-php   - php-mcrypt   - mysql-client  runcmd:   - mkdir -p /var/www/html/mywordpresssite &amp;&amp; cd /var/www/html &amp;&amp;   wget https://wordpress.org/latest.tar.gz &amp;&amp; tar -xzf /var/www/html/   latest.tar.gz -C /var/www/html/mywordpresssite --strip-components 1   - i=0; while [ \$i -le 5 ]; do mysql --connect-timeout=3 -h \$   {DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" &amp;&amp;   break    sleep 15; i=\$((i+1)); done   - mysql -u root -pmysqlpassword -h \${DBTier.networks[0].address}   -e "create database wordpress_blog;"   - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/   html/mywordpresssite/wp-config.php   - sed -i -e s/"define( 'DB_NAME',   'database_name_here' );"/"define( 'DB_NAME',   'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php &amp;&amp; sed   -i -e s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER',   'root' );"/ /var/www/html/mywordpresssite/wp-config.php &amp;&amp; sed -i -e   s/"define( 'DB_PASSWORD', 'password_here' );"/"define( 'DB_PASSWORD',   'mysqlpassword' );"/ /var/www/html/mywordpresssite/wp-config.php &amp;&amp; sed   -i -e s/"define( 'DB_HOST', 'localhost' );"/"define( 'DB_HOST', '\$   {DBTier.networks[0].address}' );"/ /var/www/html/mywordpresssite/wp-   config.php   - service apache2 reload </pre>

### 範例：基本藍圖代碼已完成的範例

```

inputs:
  env:
    type: string
    enum:
      - 'env:dev'
      - 'env:prod'
      - 'env:test'
    default: 'env:dev'
  title: Environment
  description: Target Environment
size:
  type: string
  enum:

```

```

    - small
    - medium
    - large
  description: Size of Nodes
  title: Tier Machine Size
username:
  type: string
  minLength: 4
  maxLength: 20
  pattern: '[a-z]+'
  title: Database Username
  description: Database Username
userpassword:
  type: string
  pattern: '[a-z0-9A-Z@#&$]+'
  encrypted: true
  title: Database Password
  description: Database Password
databaseDiskSize:
  type: number
  default: 4
  maximum: 10
  title: MySQL Data Disk Size
  description: Database Disk Size
resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: ubuntu-16
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
        assignPublicIpAddress: true
    remoteAccess:
      authentication: usernamePassword
      username: '${input.username}'
      password: '${input.userpassword}'
    cloudConfig: |
      #cloud-config
      repo_update: true
      repo_upgrade: all

    packages:
      - mysql-server

    runcmd:
      - sed -e '/bind-address/ s/^#*#/' -i /etc/mysql/mysql.conf.d/mysqld.cnf
      - service mysql restart
      - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
      - mysql -e "FLUSH PRIVILEGES;"
    attachedDisks: []
  WebTier:

```

```

type: Cloud.Machine
properties:
  name: wordpress
  flavor: '${input.size}'
  image: ubuntu-16
  constraints:
    - tag: '${input.env}'
  networks:
    - network: '${resource["WP-Network-Private"].id}'
      assignPublicIpAddress: true
  cloudConfig: |
    #cloud-config
    repo_update: true
    repo_upgrade: all

  packages:
    - apache2
    - php
    - php-mysql
    - libapache2-mod-php
    - php-mcrypt
    - mysql-client

  runcmd:
    - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://
wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
mywordpresssite --strip-components 1
    - i=0; while [ $i -le 5 ]; do mysql --connect-timeout=3 -h $
{DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;
i=$((i+1)); done
    - mysql -u root -pmysqlpassword -h ${DBTier.networks[0].address} -e "create database
wordpress_blog;"
    - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
mywordpresssite/wp-config.php
    - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME',
'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e
s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER', 'root' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_PASSWORD',
'password_here' );"/"define( 'DB_PASSWORD', 'mysqlpassword' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_HOST',
'localhost' );"/"define( 'DB_HOST', '${DBTier.networks[0].address}' );"/ /var/www/html/
mywordpresssite/wp-config.php
    - service apache2 reload
WP-Network-Private:
type: Cloud.Network
properties:
  name: WP-Network-Private
  networkType: existing
  constraints:
    - tag: 'type:isolated-net'
    - tag: '${input.env}'

```

## 後續步驟

透過檢查語法並進行部署來測試藍圖。

## WordPress 使用案例：測試基本藍圖

在開發期間，您通常可以從基礎元件開始來建置 vRealize Automation Cloud Assembly 藍圖，然後隨著藍圖增長進行部署和測試。

若要確保部署以您想要的方式運作，您可以多次測試和部署藍圖。可逐漸新增更多元件，並重新測試和重新部署。

### 必要條件

建立基本藍圖。請參閱 [WordPress 使用案例：建立基本藍圖](#)。

### 程序

- 1 按一下 **藍圖**，然後開啟 WordPress-BP 藍圖。

將在設計畫布和代碼編輯器中顯示基本藍圖。

- 2 若要檢查藍圖語法、放置和基本有效性，請按一下左下方的**測試**。
- 3 輸入值，然後按一下**測試**。

Environment	env:dev	▼ ⓘ
Database Tier Size *	small	▼ ⓘ
Database Username *	ouradmin	
Database Password *	••••••	
MySQL Data Disk Size	4	▼ ⓘ

此測試僅是一種模擬，不會實際部署虛擬機器或其他資源。模擬會暴露潛在問題，例如，未定義任何與藍圖中的硬限制相符的資源功能。

測試包含**佈建圖**的連結，您可以在這裡檢查模擬的部署流程並查看發生的任何錯誤。

請求詳細資料

NETWORK ALLOCATION

❗ 請求: WP-Network

錯誤: Could not find any profile to match network 'WP-Network' of type 'PUBLIC' with constraints '[env:dev]'.

請求類型: Allocation

網路類型: PUBLIC

面向網路: env.dev:hard

限制: env.dev:hard

❗ 專案: wordpress project

網路限制: 無

儲存區限制: 無

擴充性限制: 無

成功模擬並不能保證您可以部署藍圖而不發生任何錯誤。

- 藍圖通過模擬後，按一下左下方的**部署**。
- 選取**建立新部署**。
- 將部署命名為 **WordPress for OurCo**，然後按下一步。
- 輸入值，然後按一下**部署**。
- 若要確認藍圖是否已成功部署，請查看**部署**下方。

如果部署失敗，請按一下其名稱，然後按一下**歷程記錄**索引標籤以檢視可協助進行疑難排解的訊息。

時間戳記	狀態	資源類型	資源名稱	詳細資料
2020年1月21日 上午9:41:32	REQUEST_FINISHED			
2020年1月21日 上午9:41:31	COMPLETION_FINISHED			
2020年1月21日 上午9:41:14	COMPLETION_IN_PROGRESS			
2020年1月21日 上午9:40:51	CREATE_FINISHED	Cloud.Machine	Cloud_vSphere_Machine_1[...	
2020年1月21日 上午9:33:05	CREATE_IN_PROGRESS	Cloud.Machine	Cloud_vSphere_Machine_1[...	Request is in stage STARTED and substage RESOURCE_COUNTED
2020年1月21日 上午9:31:05	CREATE_IN_PROGRESS	Cloud.Machine	Cloud_vSphere_Machine_1[...	

某些歷程記錄項目的最右側可能有**佈建圖**連結。此圖類似於模擬圖，您可在此檢查佈建程序中 vRealize Automation Cloud Assembly 決定點的流程圖。

若要查看更多流程圖，請按一下**基礎結構 > 活動 > 申請**。

- 若要確認應用程式是否正在執行，請在瀏覽器中開啟 **WordPress 開始頁面**。
  - 等待 WordPress 伺服器完全建立和初始化。  
視環境而定，可能需要 30 分鐘或更多時間來進行初始化。
  - 若要尋找站台 FQDN 或 IP 位址，請移至**部署 > 拓撲**。



- c 在畫布中，按一下 WebTier，然後在右側的面板中尋找 IP 位址。
- d 輸入此 IP 位址做為 WordPress 開始頁面的完整 URL 的一部分。

在此使用案例中，完整 URL 為：

`http://{IP-address}/mywordpresssite`

或

`http://{IP-address}/mywordpresssite/wp-admin/install.php`

- 10 在瀏覽器中檢查 WordPress 之後，如果需要對應用程式進行其他處理，請進行藍圖變更，然後使用**更新現有部署**選項重新部署。
- 11 考慮對藍圖進行版本設定。如果變更導致部署失敗，您可以還原為正常運作的版本。
  - a 在藍圖設計頁面中，按一下**版本**。
  - b 在 [建立版本] 頁面中，輸入 **WP-1.0**。  
請勿在版本名稱中輸入空格。
  - c 按一下**建立**。

若要檢閱或還原為某個版本，請在設計頁面上，按一下**版本歷程記錄**索引標籤。

- 12 基本部署現已準備就緒，您可以透過增加應用程式伺服器 and 資料庫伺服器的 CPU 和記憶體來嘗試首次部署時增強功能。

將應用程式伺服器和資料庫伺服器更新為中型節點大小。使用相同的藍圖，在部署時選取 **medium**，重新部署並再次確認應用程式。

#### 後續步驟

透過新增更多元件，將藍圖擴充到值得生產的應用程式。

### WordPress 使用案例：擴充藍圖

建立並測試基本 vRealize Automation Cloud Assembly 藍圖後，將其擴充到可部署至開發、測試與最終生產環境的多階層應用程式。

若要擴充藍圖，請新增下列增強功能。

- 用於叢集化應用程式伺服器以增加容量的選項
- 應用程式伺服器前方的公用網路和負載平衡器
- 具有封存儲存區的備份伺服器

#### 必要條件

建立基本藍圖，並對其進行測試。請參閱 [WordPress 使用案例：建立基本藍圖](#) 和 [WordPress 使用案例：測試基本藍圖](#)。

## 程序

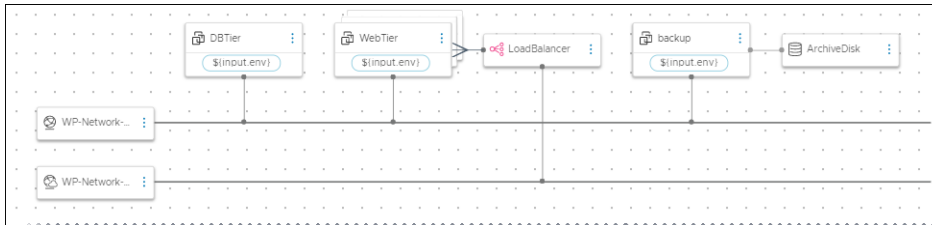
- 1 按一下**藍圖**，然後開啟 WordPress-BP 藍圖。

將在設計畫布和代碼編輯器中顯示基本藍圖。

- 2 請使用用於指引的代碼範例和圖進行新增和變更。

您可以使用 GUI 將新資源拖曳至畫布 (例如負載平衡器)，然後在代碼編輯器中完成組態。

- a 新增 `count` 輸入提示，以使 WordPress 應用程式伺服器加入叢集。
- b 新增非雲端負載平衡器。
- c 將負載平衡器連線到 WordPress 應用程式伺服器叢集。
- d 新增非雲端備份機器。
- e 將備份機器連線到私人/內部網路。
- f 新增非雲端公用/外部網路。
- g 將負載平衡器連線到公用網路。
- h 新增非雲端儲存磁碟區，以用作封存磁碟。
- i 將封存磁碟連線到備份機器。
- j 為儲存區磁碟速度新增 `archiveusage` 輸入提示。
- k 為儲存區磁碟大小新增 `archiveDiskSize` 輸入提示。



- 3 部署、測試和變更的方式與為基本藍圖執行的操作相同。

您可以更新現有部署，甚至部署新的執行個體，以便可以比較部署。

目標是取得可用於生產部署的固定、可重複的藍圖。

**範例：**擴充的藍圖代碼已完成的範例

```
inputs:
  env:
    type: string
    enum:
      - 'env:dev'
      - 'env:prod'
      - 'env:test'
    default: 'env:dev'
    title: Environment
    description: Target Environment
```

```

size:
  type: string
  enum:
    - small
    - medium
    - large
  description: Size of Nodes
  title: Tier Machine Size
username:
  type: string
  minLength: 4
  maxLength: 20
  pattern: '[a-z]+'
  title: Database Username
  description: Database Username
userpassword:
  type: string
  pattern: '[a-z0-9A-Z@#&]+'
  encrypted: true
  title: Database Password
  description: Database Password
databaseDiskSize:
  type: number
  default: 4
  maximum: 10
  title: MySQL Data Disk Size
  description: Database Disk Size

count: type: integer default: 2 maximum: 5 minimum: 2 title: WordPress Cluster Size
description: WordPress Cluster Size (Number of Nodes) archiveDiskSize: type: number default:
4 maximum: 10 title: WordPress Archive Disk Size description: Archive Storage Disk Speed
archiveusage: type: string enum: - 'usage:general' - 'usage:fast' description: Archive
Storage Disk Speed title: Archive Disk Speed
resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: ubuntu-16
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
      networks:
        - network: '${resource["WP-Network-Private"].id}'
          assignPublicIpAddress: true
      remoteAccess:
        authentication: usernamePassword
        username: '${input.username}'
        password: '${input.userpassword}'
      cloudConfig: |
        #cloud-config
        repo_update: true
        repo_upgrade: all

      packages:

```

```

- mysql-server

runcmd:
- sed -e '/bind-address/ s/^#*#/' -i /etc/mysql/mysql.conf.d/mysqld.cnf
- service mysql restart
- mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
- mysql -e "FLUSH PRIVILEGES;"

attachedDisks: []
WebTier:
  type: Cloud.Machine
  properties:
    name: wordpress
    flavor: '${input.size}'
    image: 'ubuntu-16'
    count: '${input.count}'
    constraints:
      - tag: '${input.env}'
  networks:
    - network: '${resource["WP-Network-Private"].id}'
      assignPublicIpAddress: true
  storage: disks: - capacityGb: '${input.archiveDiskSize}' name: ArchiveDisk
  cloudConfig: |
    #cloud-config
    repo_update: true
    repo_upgrade: all

  packages:
    - apache2
    - php
    - php-mysql
    - libapache2-mod-php
    - php-mcrypt
    - mysql-client

  runcmd:
    - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://
wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
mywordpresssite --strip-components 1
    - i=0; while [ $i -le 10 ]; do mysql --connect-timeout=3 -h $
{DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;
i=$((i+1)); done
    - mysql -u root -pmysqlpassword -h ${DBTier.networks[0].address} -e "create database
wordpress_blog;"
    - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
mywordpresssite/wp-config.php
    - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME',
'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e
s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER', 'root' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_PASSWORD',
'password_here' );"/"define( 'DB_PASSWORD', 'mysqlpassword' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_HOST',
'localhost' );"/"define( 'DB_HOST', '${DBTier.networks[0].address}' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_HOST',
'localhost' );"/"define( 'DB_HOST', '${DBTier.networks[0].address}' );"/ /var/www/html/
mywordpresssite/wp-config.php

```

```

- service apache2 reload

LoadBalancer: type: Cloud.LoadBalancer properties: name: myapp-lb network: '${resource["WP-
Network-Public"].id}' instances: - '${WebTier.id}' routes: - protocol: HTTP port: '80'
instanceProtocol: HTTP instancePort: '80' healthCheckConfiguration: protocol: HTTP port: '80'
urlPath: /mywordpresssite/wp-admin/install.php intervalSeconds: 6 timeoutSeconds: 5
unhealthyThreshold: 2 healthyThreshold: 2 internetFacing: true
WP-Network-Private:
  type: Cloud.Network
  properties:
    name: WP-Network-Private
    networkType: existing
    constraints:
      - tag: 'type:isolated-net'
      - tag: '${input.env}'

WP-Network-Public: type: Cloud.Network properties: name: WP-Network-Public networkType:
public constraints: - tag: 'type:public-net' - tag: '${input.env}' backup: type:
Cloud.Machine properties: name: backup flavor: '${input.size}' image: 'ubuntu-16' networks: -
network: '${resource["WP-Network-Private"].id}' constraints: - tag: '${input.env}'
attachedDisks: - source: '${ArchiveDisk.id}' ArchiveDisk: type: Cloud.Volume properties:
name: ArchiveDisk capacityGb: 5 constraints: - tag: '${input.archiveusage}' - tag: '$
{input.env}'

```

#### 後續步驟

定義您自己的基礎結構，並建立您自己的藍圖。

請參閱第 4 章 建置 vRealize Automation Cloud Assembly 資源基礎結構和 第 6 章 設計 vRealize Automation Cloud Assembly 部署。

## VMware Cloud on AWS 使用案例

此 vRealize Automation Cloud Assembly 使用案例顯示了定義資源基礎結構和藍圖設定，以部署到 VMware Cloud on AWS 環境的程序。

此程序要求雲端管理員已按照《VMware Cloud on AWS 入門》說明文件中的部署和管理軟體定義資料中心所述設定組織的 VMware Cloud on AWS SDDC 資料中心。

請查看連續設定，以瞭解為 VMware Cloud on AWS 設定環境的程序。請注意，您看到的值只是使用案例範例而已。請考量您可以在何處從範例值替換或推斷，以符合您自己的雲端基礎結構和部署需求。

如何為 Cloud Assembly 設定 VMware Cloud on AWS 的《VMware 雲端管理技術行銷》中提供了類似工作流程的詳細視訊。

#### 程序

##### 1 在 vRealize Automation Cloud Assembly 中設定基本 VMware Cloud on AWS 工作流程

此 vRealize Automation Cloud Assembly 使用案例說明了定義資源基礎結構的程序，以及用於部署至 VMware Cloud on AWS 環境的對應藍圖。

## 2 在 vRealize Automation Cloud Assembly 的 VMware Cloud on AWS 工作流程中設定隔離網路

在此程序中，您會在 vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署新增隔離網路。

## 在 vRealize Automation Cloud Assembly 中設定基本 VMware Cloud on AWS 工作流程

此 vRealize Automation Cloud Assembly 使用案例說明了定義資源基礎結構的程序，以及用於部署至 VMware Cloud on AWS 環境的對應藍圖。

在此程序中，您將設定支援將藍圖部署到現有 VMware Cloud on AWS 環境中資源的基礎結構。

### 必要條件

- 您必須屬於現有 VMware Cloud on AWS SDDC 環境中的組織，才能在 vRealize Automation Cloud Assembly 中建立和設定 VMware Cloud on AWS 雲端帳戶。如需設定 VMware Cloud on AWS 服務的相關資訊，請參閱 [VMware Cloud on AWS 說明文件](#)。
- 為了方便在 vCenter 中的現有 VMware Cloud on AWS 主機 SDDC 和 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 雲端帳戶之間建立所需的連線，您必須提供網路連線，並透過使用 VPN 或類似的網路方式新增防火牆規則。請參閱 [準備 VMware Cloud on AWS SDDC 以與 vRealize Automation 中的 VMware Cloud on AWS 雲端帳戶連線](#)。

### 程序

- 1 [準備 VMware Cloud on AWS SDDC 以與 vRealize Automation 中的 VMware Cloud on AWS 雲端帳戶連線](#)  
在 vRealize Automation Cloud Assembly 內部部署環境中使用 VMware Cloud on AWS 雲端帳戶時，您必須建立網路連線，以支援在 vCenter 中的 SDDC 和 vRealize Automation 中的任何 VMware Cloud on AWS 雲端帳戶之間進行通訊。
- 2 [在範例工作流程中於 vRealize Automation 中建立 VMware Cloud on AWS 雲端帳戶](#)  
在此步驟中，您在 vRealize Automation 中建立 VMware Cloud on AWS 雲端帳戶。
- 3 [在 vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署建立雲端區域](#)  
在此步驟中，您會建立一個雲端區域，來指定 CloudAdmin 使用者在 vRealize Automation Cloud Assembly 中使用 VMware Cloud on AWS 時可存取的計算資源。
- 4 [在 vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署設定網路和儲存區設定檔](#)  
在此步驟中，您會設定網路設定檔和儲存區設定檔，以指定可供 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS CloudAdmin 使用者使用的資源。
- 5 [在 vRealize Automation Cloud Assembly 中建立專案以支援 VMware Cloud on AWS 部署](#)  
在此步驟中，您會定義一個 vRealize Automation Cloud Assembly 專案，該專案可用於控制 VMware Cloud on AWS 部署可用的資源。

## 6 在藍圖設計中定義 vCenter 機器資源以支援 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署

在此步驟中，您將 vCenter 機器資源拖曳至設計畫布上，並為 VMware Cloud on AWS 部署新增設定。

### 準備 VMware Cloud on AWS SDDC 以與 vRealize Automation 中的 VMware Cloud on AWS 雲端帳戶連線

在 vRealize Automation Cloud Assembly 內部部署環境中使用 VMware Cloud on AWS 雲端帳戶時，您必須建立網路連線，以支援在 vCenter 中的 SDDC 和 vRealize Automation 中的任何 VMware Cloud on AWS 雲端帳戶之間進行通訊。

為了方便在 vCenter 中的現有 VMware Cloud on AWS 主機 SDDC 和 vRealize Automation 中的 VMware Cloud on AWS 雲端帳戶之間建立所需的連線，您必須使用 VPN 或類似的網路方式在兩個元素之間提供網路連線。

#### 程序

- 1 透過公用網際網路或 AWS Direct Connect 設定 VPN 連線。

請參閱 [VMware Cloud on AWS 說明文件](#) 中的《VMware Cloud on AWS 網路與安全性》。

- 2 確認是否可以在管理網路上的私人 IP 位址中解析 vCenter Server FQDN。

請參閱 [VMware Cloud on AWS 說明文件](#) 中的《VMware Cloud on AWS 網路與安全性》。

- 3 設定所需的防火牆規則。

必須在 SDDC 的 VMware Cloud on AWS 主控台中設定管理閘道防火牆規則以支援通訊。這些規則必須位於**管理閘道防火牆規則**區段中。透過使用 SDDC 主控台中**網路與安全性**索引標籤上的選項來建立防火牆規則。

- 將用於 HTTPS (TCP 443) 服務的 ESXi 的網路流量限制為已探索到的 vRealize Automation 應用裝置/伺服器或 vRealize Automation 負載平衡器 VIP 的 IP 位址。
- 將用於 ICMP (所有 ICMP)、SSO (TCP 7444) 和 HTTPS (TCP 443) 服務的 vCenter 的網路流量限制為已探索到的 vRealize Automation 應用裝置/伺服器或 vRealize Automation 負載平衡器 VIP 的 IP 位址。
- 將用於 HTTPS (TCP 443) 服務的 NSX-T Manager 的網路流量限制為已探索到的 vRealize Automation 應用裝置/伺服器或 vRealize Automation 負載平衡器 VIP 的 IP 位址。

下表概述了所需的防火牆規則。

表 3-1. 所需管理閘道防火牆規則摘要

名稱	來源	目的地	服務
vCenter	內部部署資料中心的 CIDR 區塊	vCenter	任何 (所有流量)
vCenter Ping	任何	vCenter	ICMP (所有 ICMP)
NSX Manager	內部部署資料中心的 CIDR 區塊	NSX Manager	任何 (所有流量)

表 3-1. 所需管理閘道防火牆規則摘要 (續)

名稱	來源	目的地	服務
內部部署至 ESXi Ping	內部部署資料中心的 CIDR 區塊	僅限 ESXi 管理	ICMP (所有 ICMP)
內部部署至 ESXi 遠端主控台和佈建	內部部署資料中心的 CIDR 區塊	僅限 ESXi 管理	TCP 902
內部部署至 SDDC 虛擬機器	內部部署資料中心的 CIDR 區塊	SDDC 邏輯網路的 CIDR 區塊	任何 (所有流量)
SDDC 虛擬機器至內部部署	SDDC 邏輯網路的 CIDR 區塊	內部部署資料中心的 CIDR 區塊	任何 (所有流量)

如需相關資訊，請參閱 [VMware Cloud on AWS 說明文件](#) 中的《VMware Cloud on AWS 網路與安全性》和《VMware Cloud on AWS 作業指南》。

## 結果

在設定所需的閘道存取和防火牆規則之後，您可以繼續執行建立 VMware Cloud on AWS 雲端帳戶的程序。

## 在範例工作流程中於 vRealize Automation 中建立 VMware Cloud on AWS 雲端帳戶

在此步驟中，您在 vRealize Automation 中建立 VMware Cloud on AWS 雲端帳戶。

如需相關資訊，請參閱 [VMware Cloud on AWS 說明文件](#)。

除非另有說明，否則在此程序中輸入的步驟值僅適用於此範例工作流程。

## 必要條件

- 此程序假設您擁有必要的管理員認證，包括 vCenter 中目標 SDDC 的 VMware Cloud on AWS CloudAdmin 認證，並且已在連接埠 443 上啟用 HTTPS 存取。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 此程序假設您具有雲端管理員使用者角色。請參閱 [什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 為了方便在 vCenter 中的現有 VMware Cloud on AWS 主機 SDDC 和 vRealize Automation 中的 VMware Cloud on AWS 雲端帳戶之間建立所需的連線，您必須使用 VPN 或類似的網路方式提供網路連線和防火牆規則。請參閱 [準備 VMware Cloud on AWS SDDC 以與 vRealize Automation 中的 VMware Cloud on AWS 雲端帳戶連線](#)。如果您使用外部 HTTP 網際網路 Proxy，則必須針對 IPv4 進行設定。
- 如果沒有外部網際網路存取權，則設定網際網路伺服器 Proxy。請參閱 [如何為 vRealize Automation 設定網際網路 Proxy 伺服器](#)。

## 程序

- 1 選取 **基礎結構 > 連線 > 雲端帳戶**。



## 2 按一下新增雲端帳戶，選取 VMware Cloud on AWS，並輸入值。

下表提供了範例值及支援資訊。

設定	範例值和指示	說明
VMC API Token	<ol style="list-style-type: none"> <li>按一下 <b>VMC API Token</b> 行結尾的 <b>i</b> 說明圖示，然後在說明文字方塊中按一下 <b>API Token 頁面</b>，以開啟您組織的<b>我的帳戶</b>頁面上的 <b>API Token</b> 索引標籤。</li> <li>按一下<b>產生 Token</b> 以顯示<b>產生新 API Token</b> 選項。</li> <li>輸入新 Token 名稱，例如 <b>myinitials_mytoken</b>。</li> <li>將 Token TTL 設定為<b>永不到期</b>。  如果您建立的 Token 設定為到期，則從 vRealize Automation 進行的 VMware Cloud on AWS 作業將在 Token 到期時停止運作，並持續失效直到您使用新的 Token 更新雲端帳戶為止。</li> <li>在<b>定義範圍</b>區段中，選取<b>所有角色</b>。    </li> <li>按一下<b>產生</b>。</li> <li>在產生的 Token 頁面中，按一下<b>複製</b>，然後按一下<b>繼續</b>。</li> <li>返回至<b>新增雲端帳戶</b>頁面，將複製的 Token 貼至 <b>VMC API Token</b> 列，然後按一下<b>套用 API Token</b>。    </li> </ol>	<p>您可以在連結的 <b>API Token</b> 頁面上，為您的組織建立新 Token 或使用現有的 Token。</p> <p>在<b>定義範圍</b>區段中，API Token 需要的最低角色為：</p> <ul style="list-style-type: none"> <li>■ <b>組織角色</b> <ul style="list-style-type: none"> <li>■ <b>組織成員</b></li> <li>■ <b>組織擁有者</b></li> </ul> </li> <li>■ <b>服務角色 - VMware Cloud on AWS</b> <ul style="list-style-type: none"> <li>■ <b>管理員</b></li> <li>■ <b>NSX Cloud 管理員</b></li> <li>■ <b>NSX Cloud 稽核員</b></li> </ul> </li> </ul> <p><b>備註</b> 複製、下載或列印產生的 Token。一旦離開此頁面，將無法擷取產生的 Token。</p> <p>套用產生的 Token 或提供的 Token，以連線至您組織 VMware Cloud on AWS 訂閱中可用的 SDDC 環境，並填入 SDDC 名稱清單。</p> <p>如果 vRealize Automation 和 VMware Cloud on AWS 服務位於不同的組織中，您應切換至 VMware Cloud on AWS 組織，然後產生 Token。</p> <p>如需有關 API Token 的詳細資訊，請參閱〈<a href="#">產生 API Token</a>〉。</p>
SDDC 名稱	<p>對於此範例，請選取 Datacenter:Datacenter-abz。</p> <p>有效的 SDDC 名稱會自動填入 vCenter 和 NSX-T FQDN 項目。如果雲端 Proxy 已部署至 SDDC，則雲端 Proxy 值也會自動填入。</p>	<p>從 VMware Cloud on AWS 訂閱中可用的 SDDC 清單選取。SDDC 清單以 VMware Cloud on AWS API Token 為基礎。</p> <p>vRealize Automation 不支援 NSX-V SDDC，因此它不會顯示在可用 SDDC 清單中。</p>
vCenter IP 位址/FQDN	<p>地址會根據您的 SDDC 選取項目自動填入。</p>	<p>輸入指定 SDDC 中 vCenter Server 的 IP 位址或 FQDN。</p> <p>IP 位址預設為私人 IP 位址。根據用來存取 SDDC 的網路連線類型，預設位址可能不同於指定 SDDC 中 NSX Manager 伺服器的 IP 位址。</p>

設定	範例值和指示	說明
NSX Manager IP 位址/FQDN	地址會根據您的 SDDC 選取項目自動填入。	指定所指定 SDDC 中的 NSX Manager 的 IP 位址或 FQDN。  IP 位址預設為私人 IP 位址。根據用來存取 SDDC 的網路連線類型，預設位址可能不同於指定 SDDC 中 NSX Manager 伺服器的 IP 位址。  VMware Cloud on AWS 雲端帳戶支援 NSX-T。
vCenter 使用者名稱和密碼	使用者名稱自動填入為 cloudadmin@vmc.local。	輸入指定 SDDC 的 vCenter 使用者名稱 (如果該使用者名稱與預設值不同)。  指定的使用者需要 CloudAdmin 認證。此使用者不需要 CloudGlobalAdmin 認證。  輸入使用者密碼。
驗證	按一下 <b>驗證</b> 。	驗證將確認您對指定 vCenter 的存取權限，並檢查 vCenter 是否正在執行。
名稱與說明	針對雲端帳戶名稱輸入 <b>OurCo-VMC</b> 。 針對雲端帳戶說明輸入 <b>VMC 的範例部署</b> 。	
允許佈建到這些資料中心	此資訊是唯讀的。	列出指定的 VMware Cloud on AWS SDDC 環境中可用的資料中心。
建立雲端區域	取消選取此核取方塊。對於此範例，您將稍後在工作流程中建立一個雲端區域。	請參閱 <a href="#">進一步瞭解 vRealize Automation Cloud Assembly 雲端區域</a> 。
功能標籤	將此設定保留空白。此工作流程不使用功能標籤。	根據組織的標籤策略使用標籤。請參閱 <a href="#">如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署</a> 和 <a href="#">建立標記策略</a> 。

### 3 按一下**新增**。

#### 結果

機器和磁碟區等資源從 VMware Cloud on AWS SDDC 資料中心進行資料收集，然後在 vRealize Automation **基礎結構**索引標籤的**資源**區段中列出。

#### 後續步驟

在 vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署建立雲端區域。

### 在 vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署建立雲端區域

在此步驟中，您會建立一個雲端區域，來指定 CloudAdmin 使用者在 vRealize Automation Cloud Assembly 中使用 VMware Cloud on AWS 時可存取的計算資源。

在 VMware Cloud on AWS 中，兩個主要管理員認證是雲端全域管理員和 CloudAdmin。vRealize Automation Cloud Assembly 設計為支援 CloudAdmin 使用者。部署到可供 VMware Cloud on AWS CloudAdmin 使用者使用的資源。請勿部署到需要 VMware Cloud on AWS CloudGlobalAdmin 認證的資源。

雲端區域可識別專案藍圖將機器、網路和儲存區部署至其上的計算資源。請參閱[進一步瞭解 vRealize Automation Cloud Assembly 雲端區域](#)。

除非另有說明，否則在此程序中輸入的步驟值僅適用於此範例工作流程。

#### 必要條件

- 完成在範例工作流程中於 vRealize Automation 中建立 VMware Cloud on AWS 雲端帳戶程序。
- 此程序假設您具有所需的管理員認證，包括 vCenter 中目標 SDDC 的 VMware Cloud on AWS CloudAdmin 認證。請參閱在 vRealize Automation 中使用雲端帳戶所需的認證。
- 此程序假設您具有雲端管理員使用者角色。請參閱什麼是 vRealize Automation Cloud Assembly 使用者角色。

#### 程序

- 1 選取**基礎結構 > 設定 > 雲端區域**。
- 2 按一下**新增雲端區域**，然後輸入 VMware Cloud on AWS 環境的值。

設定	範例值
帳戶/區域	OurCo-VMC / Datacenter:Datacenter-abz 這是您在上一步在範例工作流程中於 vRealize Automation 中建立 VMware Cloud on AWS 雲端帳戶中定義的雲端帳戶和相關聯的區域。
名稱	VMC_cloud_zone-1
說明	僅限 VMware Cloud on AWS 資源
放置原則	預設值
功能標籤	將此設定保留空白。此工作流程不使用功能標籤。

- 3 按一下**計算索引標籤**。
- 4 如下面的區域 1 中所示，尋找並選取可供 CloudAdmin 使用者使用的計算資源。對於此範例，使用名為 Cluster 1/ Compute-ResourcePool 的資源。

Cluster 1/ Compute-ResourcePool 是 VMware Cloud on AWS 的預設計算資源。

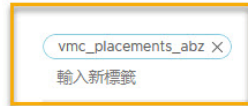


- 5 如上方的區域 2 中所示，新增標籤名稱 `vmc_placements_abz`。

標籤

已選取 1 個物件

新增標籤



移除標籤

沒有標籤 ⓘ

- 6 透過在篩選標籤區段中輸入 `vmc_placements_abz` 來篩選此雲端區域中所使用的計算資源。
- 7 按一下儲存。

名稱	帳戶/區域	類型	標籤
<input type="checkbox"/> ComputeClusterA-New	nsx-v 觀察器表 A 中在 vSphere 叢集 / NSX621-DataCenter	common title cluster	ComputeClusterA
<input checked="" type="checkbox"/> ComputeClusterA / Scale	270_VC_account 觀察器表 A 中在 vSphere 叢集 / NSX62-Scale-DC	ResourcePool	ComputeClusterA
<input type="checkbox"/> ComputeClusterA / Scale	LK-TEST 觀察器表 A 中在 vSphere 叢集 / NSX62-Scale-DC	ResourcePool	ComputeClusterA

對於此範例，只有名為 `Cluster 1/ Compute-ResourcePool` 的計算資源可供 CloudAdmin 使用者使用。

#### 後續步驟

在 vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署設定網路和儲存區設定檔。

### 在 vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署設定網路和儲存區設定檔

在此步驟中，您會設定網路設定檔和儲存區設定檔，以指定可供 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS CloudAdmin 使用者使用的資源。

雖然還需要映像和類型模板值，但是它們對於 VMware Cloud on AWS 使用者認證不具有任何獨特之處。對於此範例，在定義藍圖時，您將使用類型模板值 `small` 和映像值 `ubuntu-16`。

如需有關對應和設定檔的一般資訊，請參閱第 4 章 [建置 vRealize Automation Cloud Assembly 資源基礎結構](#)。

除非另有說明，否則在此程序中輸入的步驟值僅適用於此範例工作流程。

#### 必要條件

- 建立雲端區域。請參閱在 [vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署建立雲端區域](#)。
- 此程序假設您具有所需的管理員認證，包括 vCenter 中目標 SDDC 的 VMware Cloud on AWS CloudAdmin 認證。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。

- 此程序假設您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。

## 程序

- 1 為 VMware Cloud on AWS 部署定義網路設定檔。
  - a 選取**基礎結構 > 設定 > 網路設定檔**，然後按一下**新增網路設定檔**。

設定	範例值
帳戶/區域	OurCo-VMC / Datacenter:Datacenter-abz
名稱	vmc-network1
說明	包含具有 VMware Cloud on AWS CloudAdmin 認證的藍圖管理員可存取的網路。

- b 按一下**網路索引標籤**，然後按一下**新增網路**。
- c 選取具有 CloudAdmin 認證的 VMware Cloud on AWS 使用者可以部署到的網路，例如 `sddc-cgw-network-1`。

### 新增網路



<input type="checkbox"/>	名稱	帳戶/區域	區域	網路網域
<input checked="" type="checkbox"/>	ESO_PKS_VC01_VM_PKS	1114VCあ ア7中表議 @華ホ GE#U8aã æðeoãU*	ア7中表議	ESO_PKS_VC01_DVS01
<input type="checkbox"/>	ESO_PKS_VC01_Mgmt	1114VCあ ア7中表議	ア7中表議	ESO_PKS_VC01_DVS01

- 2 儲存網路設定檔。

### 3 為 VMware Cloud on AWS 部署定義儲存區設定檔。

設定目標為可供 CloudAdmin 使用者存取的資料存放區/叢集的儲存區設定檔。

- a 選取**基礎結構 > 設定 > 儲存區設定檔**，然後按一下**新增儲存區設定檔**。

設定	範例值
帳戶/區域	OurCo-VMC / Datacenter:Datacenter-abz 選取在在範例工作流程中於 vRealize Automation 中建立 VMware Cloud on AWS 雲端帳戶中建立的 VMware Cloud on AWS 雲端帳戶及其相符的 SDDC 資料中心。
名稱	vmc-storage1
說明	包含具有 VMware Cloud on AWS CloudAdmin 認證的藍圖管理員可以部署到的資料存放區叢集。

- b 從**資料存放區/叢集**下拉式功能表中，選取 **WorkloadDatastore** 資料存放區。



對於 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS，儲存區原則必須使用 **WorkloadDatastore** 資料存放區來支援 VMware Cloud on AWS 部署。

### 4 儲存儲存區設定檔。

#### 後續步驟

在 vRealize Automation Cloud Assembly 中建立專案以支援 VMware Cloud on AWS 部署。

### 在 vRealize Automation Cloud Assembly 中建立專案以支援 VMware Cloud on AWS 部署

在此步驟中，您會定義一個 vRealize Automation Cloud Assembly 專案，該專案可用於控制 VMware Cloud on AWS 部署可用的資源。

如需專案的相關資訊，請參閱 [vRealize Automation Cloud Assembly 專案在部署時的運作方式](#)。

除非另有說明，否則在此程序中輸入的步驟值僅適用於此範例工作流程。

#### 必要條件

- 完成在 vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署設定網路和儲存區設定檔程序。
- 此程序假設您具有所需的管理員認證，包括 vCenter 中目標 SDDC 的 VMware Cloud on AWS CloudAdmin 認證。請參閱在 vRealize Automation 中使用雲端帳戶所需的認證。
- 此程序假設您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。

## 程序

- 1 選取**基礎結構 > 設定 > 專案**。
- 2 按一下**新增專案**，然後輸入專案名稱 `VMC_proj-1_abz`。
- 3 按一下**使用者**，然後按一下**新增使用者**。

使用者需要 CloudAdmin 認證，才能存取其組織的 VMware Cloud on AWS 訂閱。

- `chris.gray@ourco.com`，管理員
- `kerry.white@ourco.com`，成員

- 4 按一下**佈建**，然後按一下**新增雲端區域**。
- 5 新增在之前步驟中設定的雲端區域。

設定	範例值
雲端區域	VMC_cloud_zone-1 您在之前步驟 (在 <a href="#">vRealize Automation Cloud Assembly</a> 中為 VMware Cloud on AWS 部署建立雲端區域) 中建立了此雲端區域。
佈建優先順序	1
執行個體限制	3

- 6 對於此範例，請忽略其他選項。

## 後續步驟

建立要在 VMware Cloud on AWS 環境中部署的藍圖。請參閱[在藍圖設計中定義 vCenter 機器資源以支援 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署](#)。

## 在藍圖設計中定義 vCenter 機器資源以支援 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署

在此步驟中，您將 vCenter 機器資源拖曳至設計畫布上，並為 VMware Cloud on AWS 部署新增設定。

建立藍圖設計，以將其部署到可用的 VMware Cloud on AWS 資源。

除非另有說明，否則在此程序中輸入的步驟值僅適用於此範例工作流程。

## 必要條件

- 此程序假設您具有藍圖設計人員認證。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 此程序假設您有適用於 vCenter 中的目標 SDDC (Datacenter:Datacenter-abz) 的 VMware Cloud on AWS CloudAdmin 認證。請參閱[在 vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 按照前幾節所述，設定資源基礎結構和專案。

## 程序

- 1 按一下設計索引標籤，然後按一下新增。

設定	範例值
名稱	vmc-bp_abz
說明	1
專案	VMC_proj-1_abz 這是之前建立的專案，可支援您之前建立的雲端區域。該專案現在與雲端區域相關聯，而雲端區域又與您之前建立的 VMware Cloud on AWS 雲端帳戶/區域相關聯。

- 2 將 vSphere 機器資源滑動至畫布。
- 3 在機器資源中編輯下列 (粗體) 藍圖資源代碼。

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: ubuntu-1604
      cpuCount: 1
      totalMemoryMB: 1024
      folderName: Workloads
```

image 可以是適用於部署需求的任何值。

您必須將 `folderName: Workloads` 陳述式新增至藍圖設計代碼，以支援 VMware Cloud on AWS 部署。`folderName: Workloads` 設定支援 VMware Cloud on AWS SDDC 環境中的 CloudAdmin 認證，並且是必要的。

附註：雖然上述代碼範例中所顯示的 `folderName: Workloads` 設定為必填，但可以直接在藍圖設計代碼中如上所述新增此設定，也可以將其新增到相關聯的雲端區域或專案中。如果在這三個位置中的多個位置指定了此設定，則優先順序如下所示：

- 專案設定會覆寫藍圖設計設定和雲端區域設定。
- 藍圖設計設定會覆寫雲端區域設定。

附註：您可以選擇性地將 `cpuCount` 和 `totalMemoryMB` 設定取代為 `flavor` (調整大小) 項目，如下所示：

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
```



```
properties:
  image: ubuntu-1604
  flavor: small
  folderName: Workloads
```

如果雲端區域有設定為 **Workloads** 的資料夾值，您不必在藍圖中設定 `folderName` 內容，除非您要覆寫雲端區域資料夾值。

#### 後續步驟

透過新增網路隔離，擴充此基本 VMware Cloud on AWS 工作流程。請參閱在 [vRealize Automation Cloud Assembly 的 VMware Cloud on AWS 工作流程中設定隔離網路](#)。

## 在 vRealize Automation Cloud Assembly 的 VMware Cloud on AWS 工作流程中設定隔離網路

在此程序中，您會在 vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署新增隔離網路。

定義 VMware Cloud on AWS 雲端帳戶時，您可以使用您在 VMware Cloud on AWS 服務中設定的 NSX-T 設定。如需在 VMware Cloud on AWS 服務中設定 NSX-T 設定的相關資訊，請參閱 VMware Cloud on AWS [產品說明文件](#)。

vRealize Automation Cloud Assembly 支援具有 NSX-T 的 VMware Cloud on AWS。不支援具有 NSX-V 的 VMware Cloud on AWS。

vRealize Automation Cloud Assembly 支援 VMware Cloud on AWS 部署的網路隔離。不支援 VMware Cloud on AWS 的其他網路方法。

基本 VMware Cloud on AWS 工作流程的這一延伸說明了下列建立隔離網路供在 vRealize Automation Cloud Assembly 藍圖中使用的方法：

- 設定以隨選網路為基礎的隔離
- 設定以隨選安全群組為基礎的隔離。

#### 必要條件

此程序擴充了基本 VMware Cloud on AWS 工作流程。它使用與您在 [VMware Cloud on AWS 使用案例](#) 工作流程中設定的相同雲端帳戶和區域、雲端區域、專案和網路設定檔。

#### 程序

- 1 在 [vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署定義隔離網路](#)  
您可以使用下列任一程序為 VMware Cloud on AWS 部署設定網路隔離：
- 2 在藍圖中定義網路元件以支援 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 的網路隔離

在此步驟中，將網路機器元件拖曳至 vRealize Automation Cloud Assembly 藍圖畫布上，並將隔離網路部署的設定新增至目標 VMware Cloud on AWS 環境。

## 在 vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署定義隔離網路

您可以使用下列任一程序為 VMware Cloud on AWS 部署設定網路隔離：

- 在 [vRealize Automation Cloud Assembly](#) 中設定以隨選網路為基礎的隔離
- 在 [vRealize Automation Cloud Assembly](#) 中設定以隨選安全群組為基礎的隔離

### 在 vRealize Automation Cloud Assembly 中設定以隨選網路為基礎的隔離

透過在 vRealize Automation Cloud Assembly 網路設定檔中指定和使用隨選網路設定，您可以根據 VMware Cloud on AWS 部署需求設定網路隔離。

您可以使用安全群組或隨選網路設定來指定隔離網路。在此範例中，您可以透過在網路設定檔中指定隨選網路設定來設定網路隔離。稍後，您可以在藍圖中存取網路，並在 VMware Cloud on AWS 部署中使用該藍圖。

除非另有說明，否則在此程序中輸入的步驟值僅適用於此範例工作流程。

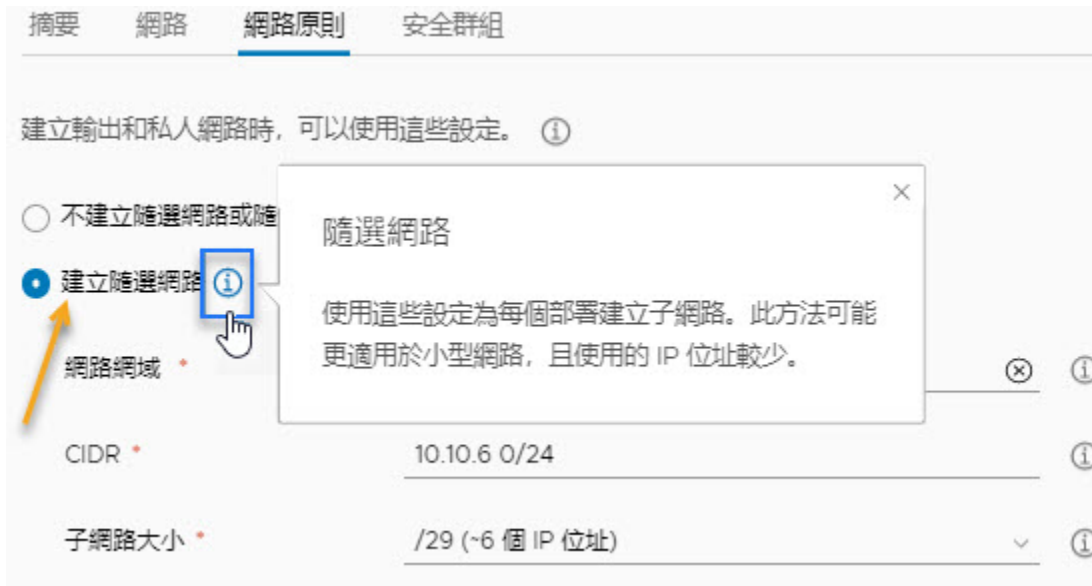
#### 必要條件

- 完成在 [vRealize Automation Cloud Assembly](#) 中設定基本 VMware Cloud on AWS 工作流程工作流程。
- 檢閱在 [vRealize Automation Cloud Assembly](#) 的 VMware Cloud on AWS 工作流程中設定隔離網路。
- 此程序假設您具有所需的管理員認證，包括 vCenter 中目標 SDDC 的 VMware Cloud on AWS CloudAdmin 認證。請參閱在 [vRealize Automation](#) 中使用雲端帳戶所需的認證。
- 此程序假設您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。

#### 程序

- 1 開啟在基本 VMware Cloud on AWS 工作流程中使用的網路設定檔，例如 `vmc-network1`。請參閱在 [vRealize Automation Cloud Assembly](#) 中為 VMware Cloud on AWS 部署設定網路和儲存區設定檔。
- 2 您不需要在網路索引標籤上進行任何選取。
- 3 按一下網路設定檔索引標籤。

- 4 選取**建立隨選網路**選項，然後選取預設的 `cgw` 網路網域。指定適當的 CIDR 和子網路大小。



- 5 按一下**儲存**。

使用此網路設定檔時，機器將部署到預設網路網域中的網路。透過使用私人或輸出網路存取將該網路與其他網路隔離。

#### 後續步驟

在藍圖中設定網路元件。請參閱[在藍圖中定義網路元件以支援 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 的網路隔離](#)

#### 在 vRealize Automation Cloud Assembly 中設定以隨選安全群組為基礎的隔離

透過在 vRealize Automation Cloud Assembly 網路設定檔中指定和使用隨選安全群組，您可以根據 VMware Cloud on AWS 部署需求設定網路隔離。

您可以使用安全群組或隨選網路設定來指定隔離網路。在此範例中，您可以透過在網路設定檔中指定隨選安全群組來設定網路隔離。稍後，您可以在藍圖中指定網路，並在 VMware Cloud on AWS 部署中使用該藍圖。

除非另有說明，否則在此程序中輸入的步驟值僅適用於此範例工作流程。

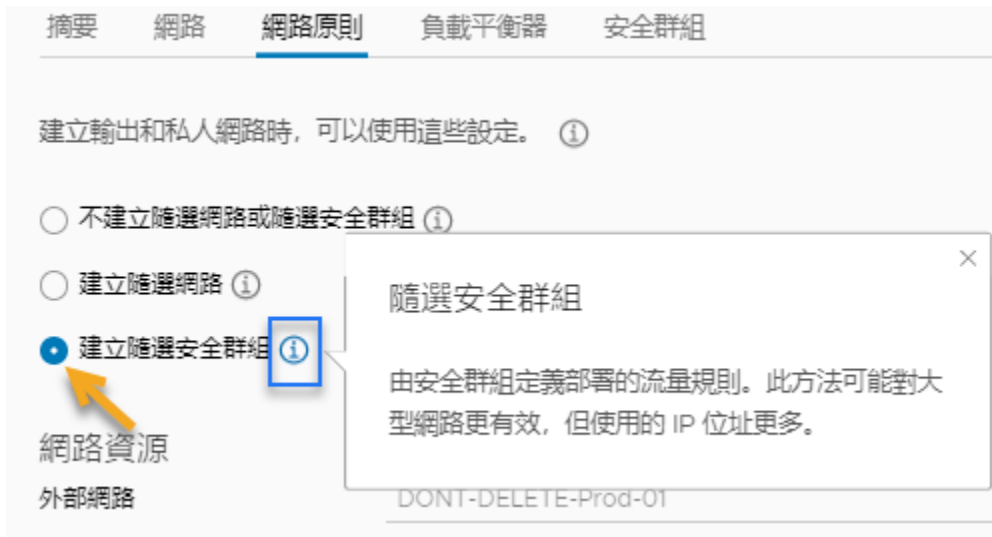
#### 必要條件

- 完成在 [vRealize Automation Cloud Assembly 中設定基本 VMware Cloud on AWS 工作流程](#) 工作流程。
- 檢閱在 [vRealize Automation Cloud Assembly 的 VMware Cloud on AWS 工作流程中設定隔離網路](#)。
- 此程序假設您具有所需的管理員認證，包括 vCenter 中目標 SDDC 的 VMware Cloud on AWS CloudAdmin 認證。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。

- 此程序假設您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。

#### 程序

- 1 開啟在基本 VMware Cloud on AWS 工作流程中使用的網路設定檔，例如 `vmc-network1`。請參閱在 [vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署設定網路和儲存區設定檔](#)。
- 2 選取在基本 VMware Cloud on AWS 工作流程中使用的現有網路，例如 `sddc-cgw-network-1`。請參閱在 [vRealize Automation Cloud Assembly 中為 VMware Cloud on AWS 部署設定網路和儲存區設定檔](#)。
- 3 按一下**網路設定檔索引標籤**。
- 4 選取**建立隨選安全群組**選項。



- 5 按一下**儲存**。

使用此網路設定檔時，機器將部署到所選網路，並由新的安全群組原則進行隔離。新的安全性原則允許私人或輸出網路存取。

#### 後續步驟

在藍圖中設定網路元件。請參閱[在藍圖中定義網路元件以支援 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 的網路隔離](#)

#### 在藍圖中定義網路元件以支援 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 的網路隔離

在此步驟中，將網路機器元件拖曳至 vRealize Automation Cloud Assembly 藍圖畫布上，並將隔離網路部署的設定新增至目標 VMware Cloud on AWS 環境。

將網路隔離新增至您先前建立的藍圖。此藍圖已與支援部署至 VMware Cloud on AWS 環境的專案和雲端區域，以及您為隔離設定的網路設定檔和網路相關聯。

除非另有說明，否則在此程序中輸入的步驟值僅適用於此範例工作流程。

#### 必要條件

- 完成在 [vRealize Automation Cloud Assembly](#) 中設定以隨選安全群組為基礎的隔離或在 [vRealize Automation Cloud Assembly](#) 中設定以隨選網路為基礎的隔離程序。
- 此程序假設您具有藍圖設計人員認證。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 此程序假設您有適用於 vCenter 中的目標 SDDC 的 VMware Cloud on AWS CloudAdmin 認證。請參閱在 [vRealize Automation](#) 中使用雲端帳戶所需的認證。

#### 程序

- 1 開啟您在上一個工作流程中建立的藍圖。請參閱在[藍圖設計中定義 vCenter 機器資源以支援 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署](#)。
- 2 從藍圖設計頁面左側的元件中，將網路元件拖曳至畫布。
- 3 編輯網路元件 YAML 代碼來指定網路類型 `private` 或 `outbound`，以粗體表示。

```
resources: Cloud_Network_1:
  type: Cloud.Network
  properties:
    name: vmc_isolated
    networkType: private
```

或

```
resources: Cloud_Network_1:
  type: Cloud.Network
  properties:
    name: vmc_isolated
    networkType: outbound
```

#### 後續步驟

您已準備好部署或關閉藍圖。

## 提供者特定的外部 IPAM 整合使用案例

您可以使用外部 IPAM 提供者來管理藍圖部署的 IP 位址指派。此範例程序說明如何在 vRealize Automation Cloud Assembly 中設定外部 IPAM 整合，並從外部 IPAM 提供者部署藍圖以取得其 IP 位址配置。此範例說明如何使用 Infoblox 做為外部 IPAM 提供者來設定 IPAM 整合。

在此程序中，您會使用現有的 IPAM 提供者套件 (在此案例中為 Infoblox 套件) 和現有的執行環境，以建立提供者特定的 IPAM 整合點。您可以設定現有網路並建立網路設定檔，以支援來自外部 IPAM 提供者的 IP 位址配置。最後，您可以建立與網路和網路設定檔相符的藍圖，並使用從外部 IPAM 提供者取得的 IP 值部署網路機器。

包含如何取得和設定 IPAM 提供者套件，以及如何設定存取雲端擴充性 Proxy 以支援 IPAM 提供者整合的執行環境的相關資訊，以供參考。

請注意，您看到的值是範例值。不能在您的環境中按原樣使用這些值。

請考量您可以在何處從範例值替換或推斷，以符合組織需求。

## 程序

### 1 在 Infoblox 應用程式中新增必要的可擴充屬性，以與 vRealize Automation 整合

您必須先在 Infoblox 中新增必要的擴充性屬性，然後才能從 Infoblox 網站或 VMware Marketplace 下載和部署 Infoblox 提供者套件 (infoblox.zip) 以便與 vRealize Automation 整合。

### 2 下載並部署要在 vRealize Automation Cloud Assembly 中使用的外部 IPAM 提供者套件

您必須先設定 IPAM 提供者套件，然後才能在 vRealize Automation Cloud Assembly 中定義外部 IPAM 整合點。您可以從 IPAM 提供者的網站或從 vRealize Automation Cloud Assembly 市集下載提供者套件。

### 3 在 vRealize Automation 中為 IPAM 整合點建立執行環境

您需要建立執行環境或存取現有的執行環境以充當 IPAM 提供者和 vRealize Automation 之間的仲介，然後才能在 vRealize Automation 中定義外部 IPAM 整合點。執行環境通常是 Amazon Web Services 或 Microsoft Azure 雲端帳戶，或是與雲端擴充性 Proxy 相關聯的以內部部署動作為基礎的擴充性整合點。

### 4 在 vRealize Automation 中新增外部 IPAM 整合點

vRealize Automation 支援與外部 IPAM 提供者整合。您可以使用提供者特定的 IPAM 整合點來取得並管理用於藍圖部署的 IP 位址和相關網路特性。

### 5 在 vRealize Automation Cloud Assembly 中將網路和網路設定檔設定為使用 IPAM 提供者值

您可以將網路定義為使用從外部 IPAM 提供者 (而非從 vRealize Automation Cloud Assembly 內部) 取得並由其進行管理的 IP 位址值。

### 6 定義和部署使用 IPAM 提供者範圍指派的 vRealize Automation Cloud Assembly 藍圖

您可以定義藍圖，以從外部 IPAM 提供者取得和管理 IP 位址指派。

### 7 在 vRealize Automation 中針對 IPAM 整合使用 Infoblox 特定內容和可擴充屬性

您可以針對包含用於 Infoblox 的外部 IPAM 整合的 vRealize Automation 專案使用 Infoblox 特定內容。

## 在 Infoblox 應用程式中新增必要的可擴充屬性，以與 vRealize Automation 整合

您必須先在 Infoblox 中新增必要的擴充性屬性，然後才能從 Infoblox 網站或 VMware Marketplace 下載和部署 Infoblox 提供者套件 (infoblox.zip) 以便與 vRealize Automation 整合。

如果要為與 vRealize Automation Cloud Assembly 的 Infoblox 整合建立外部 IPAM 整合點，則此程序適用。

您必須先使用組織帳戶管理員認證登入 Infoblox 帳戶，並預先建立下列 Infoblox 可擴充屬性，才能使用 infoblox.zip 下載：

- VMware NIC index
- VMware resource ID
- Tenant ID
- CMP Type
- VM ID
- VM Name

#### 必要條件

- 確認您擁有 [Infoblox](#) 帳戶，並且您擁有對組織 Infoblox 帳戶的正確存取認證。
- 確認 Infoblox WAPI 版本受支援。IPAM 與 Infoblox 的整合取決於 Infoblox WAPI 版本 v2.7。支援 WAPI v2.7 的所有 Infoblox 應用裝置均受支援。
- 檢閱在 [vRealize Automation](#) 中針對 IPAM 整合使用 Infoblox 特定內容和可擴充屬性。

#### 程序

- 1 使用管理員認證登入 Infoblox 帳戶。

這些是您使用[基礎結構 > 連線 > 整合](#) > 功能表順序在 vRealize Automation Cloud Assembly 中建立外部 IPAM 整合點時所指定的相同管理員使用者名稱和密碼認證。

- 2 請使用 Infoblox 說明文件中所述的程序，在 Infoblox 應用程式中建立下列必要的可擴充屬性。

- VMware NIC index - 「整數」類型
- VMware resource ID - 「字串」類型
- Tenant ID - 「字串」類型
- CMP Type - 「字串」類型
- VM ID - 「字串」類型
- VM Name - 「字串」類型

此程序會在 Infoblox 說明文件主題[關於可擴充屬性的〈新增可擴充屬性〉](#)一節中進行說明。另請參閱[〈管理可擴充屬性〉](#)。

#### 後續步驟

新增必要屬性之後，您可以繼續執行下載和部署 Infoblox 套件的程序，如[下載並部署要在 vRealize Automation Cloud Assembly 中使用的外部 IPAM 提供者套件](#)中所述。



## 下載並部署要在 vRealize Automation Cloud Assembly 中使用的外部 IPAM 提供者套件

您必須先設定 IPAM 提供者套件，然後才能在 vRealize Automation Cloud Assembly 中定義外部 IPAM 整合點。您可以從 IPAM 提供者的網站或從 vRealize Automation Cloud Assembly 市集下載提供者套件。

您可以從 IPAM 提供者的網站、從 [VMware Solutions Exchange Marketplace](#)，或從 vRealize Automation Cloud Assembly 市集索引標籤 (如果可用) 取得提供者特定的整合套件。

---

**備註** 此範例使用 VMware 提供的 Infoblox 套件 `Infoblox.zip`，可從 VMware Solutions Exchange Marketplace 下載此套件，如下所示：

- [vRA Cloud Infoblox 外掛程式版本 1.1](#) - 支援 vRealize Automation 8.1
- [vRA Cloud Infoblox 外掛程式版本 1.0](#) - 支援 vRealize Automation 8.0.1
- [vRA Cloud Infoblox 外掛程式版本 0.1](#) - 支援 vRealize Automation 8.0

IPAM 與 Infoblox 的整合取決於 Infoblox WAPI 版本 v2.7。支援 WAPI v2.7 的所有 Infoblox 應用裝置均受支援。

---

如需如何為其他 IPAM 提供者建立 IPAM 整合套件 (如果在市集中尚不存在) 的相關資訊，請參閱[如何使用 IPAM SDK 為 vRealize Automation 建立提供者特定的外部 IPAM 整合套件](#)。

IPAM 提供者套件包含隨中繼資料和其他組態一併封裝的指令碼。這些指令碼包含 vRealize Automation Cloud Assembly 與外部 IPAM 提供者協調執行的作業所使用的原始程式碼。範例作業包括 `Allocate an IP address for a virtual machine`、`Fetch a list of IP ranges from the provider` 和 `Update the MAC address of a host record in the provider`。

### 必要條件

- 確認您具有雲端管理員認證。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 確認您擁有外部 IPAM 提供者的帳戶，例如 [Infoblox](#) 或 [Bluecat](#)，並且您擁有對組織的 IPAM 提供者帳戶的正確存取認證。
- 如果您使用 Infoblox 做為外部 IPAM 提供者，請先確認您已在 Infoblox 帳戶中新增必要的可擴充屬性，然後再繼續。請參閱在 [Infoblox 應用程式中新增必要的可擴充屬性，以與 vRealize Automation 整合](#)。

### 程序

- 1 在 VMware Solutions Exchange Marketplace 中，導覽至 [vRA Cloud Infoblox 外掛程式版本 0.1](#) (vRealize Automation 8.0) 或 [vRA Cloud Infoblox 外掛程式版本 1.0](#) (vRealize Automation 8.0.1) 套件頁面。
- 2 登入並下載外掛程式套件。



- 3 如果尚未執行此操作，請在 Infoblox 中新增所需的可擴充屬性。請參閱在 [Infoblox 應用程式中新增必要的可擴充屬性，以與 vRealize Automation 整合](#)。

#### 結果

您現在可以使用此套件，透過 **整合 > 新增整合 > IPAM > 管理提供者 > 匯入套件** 功能表順序 (如在 [vRealize Automation 中新增外部 IPAM 整合點](#) 中所述) 進行部署。

## 在 vRealize Automation 中為 IPAM 整合點建立執行環境

您需要建立執行環境或存取現有的執行環境以充當 IPAM 提供者和 vRealize Automation 之間的仲介，然後才能在 vRealize Automation 中定義外部 IPAM 整合點。執行環境通常是 Amazon Web Services 或 Microsoft Azure 雲端帳戶，或是與雲端擴充性 Proxy 相關聯的以內部部署動作為基礎的擴充性整合點。

外部 IPAM 整合需要執行環境。當您定義 IPAM 整合點時，您可以透過指定可用的執行環境來建立 vRealize Automation Cloud Assembly 與 IPAM 提供者之間的連線。

IPAM 整合在執行環境中使用一組已下載的提供者特定的指令碼或外掛程式，該環境透過功能即服務 (FaaS) 提供者 (例如 Amazon Web Services Lambda、Microsoft Azure 函數或以動作為基礎的擴充性 (ABX) 內部部署內嵌式整合點) 提供便利。執行環境用於連線至外部 IPAM 提供者，例如 Infoblox。

---

**備註** Infoblox IPAM 整合點需要以動作為基礎的擴充性 (ABX) 內部部署內嵌式整合點。

---

每種類型的執行期間環境各有優缺點：

- 以動作為基礎的擴充性 (ABX) 整合點
  - 免費，無額外的廠商使用成本
  - 可連線至位於不可公開存取的 NAT/防火牆後方的內部部署資料中心內的 IPAM 廠商應用裝置，例如 Infoblox
  - 相較於商業雲端廠商，效能更慢且略不可靠
- Amazon Web Services
  - 具有相關聯的廠商 FaaS 連線/使用成本
  - 無法連線至位於不可公開存取的 NAT/防火牆後方的內部部署資料中心內的 IPAM 廠商應用裝置
  - 效能快速且高度可靠
- Microsoft Azure
  - 具有相關聯的廠商 FaaS 連線/使用成本
  - 無法連線至位於不可公開存取的 NAT/防火牆後方的內部部署資料中心內的 IPAM 廠商應用裝置
  - 效能快速且高度可靠

#### 必要條件

- 確認您具有雲端管理員認證。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有雲端管理員使用者角色。請參閱 [什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。

- 確認您擁有外部 IPAM 提供者的帳戶，例如 [Infoblox](#) 或 [Bluecat](#)，並且您擁有對組織的 IPAM 提供者帳戶的正確存取認證。
- 確認您可以存取 IPAM 提供者 (例如 Infoblox 或 BlueCat) 的已部署整合套件。已部署的套件最初是從 IPAM 提供者網站或從 vRealize Automation Cloud Assembly 市集以 .zip 形式下載，然後部署至 vRealize Automation Cloud Assembly。

如需如何部署提供者套件 .zip 檔案並在 [IPAM 整合] 頁面上將其做為提供者值使用的相關資訊，請參閱[下載並部署要在 vRealize Automation Cloud Assembly 中使用的外部 IPAM 提供者套件](#)。

#### 程序

- 1 若要建立以內部部署 FaaS 為基礎的擴充性動作以用作 IPAM 整合的執行環境，請選取**擴充性 > 程式庫 > 動作**。
- 2 按一下**新增動作**，然後輸入動作名稱和說明，並指定專案。
- 3 在 **FaaS 提供者**下拉式功能表中，然後選取**內部部署**。
- 4 完成表單以定義擴充性動作。



如需執行環境的相關資訊，請參閱此視訊大約 24 分鐘處播放的 [Infoblox IPAM 外掛程式 1.1 整合部落格視訊](#)。

## 在 vRealize Automation 中新增外部 IPAM 整合點

vRealize Automation 支援與外部 IPAM 提供者整合。您可以使用提供者特定的 IPAM 整合點來取得並管理用於藍圖部署的 IP 位址和相關網路特性。

在此範例中，您可以建立外部 IPAM 整合點，以支援存取組織的外部 IPAM 提供者帳戶。在此範例工作流程中，IPAM 提供者為 Infoblox，並且提供者特定的整合套件已存在。雖然這些指示特定於 Infoblox 整合，但如果為其他外部 IPAM 提供者建立 IPAM 整合，也可以將其用作參考。

您可以從 IPAM 提供者的網站、從 [VMware Solutions Exchange Marketplace](#)，或從 vRealize Automation Cloud Assembly **市集**索引標籤 (如果可用) 取得提供者特定的整合套件。

此範例使用 VMware 提供的 Infoblox 套件 `Infoblox.zip`，可從 VMware Solutions Exchange Marketplace 下載此套件，如下所示：

- [vRA Cloud Infoblox 外掛程式版本 1.1](#) - 支援 vRealize Automation 8.1
- [vRA Cloud Infoblox 外掛程式版本 1.0](#) - 支援 vRealize Automation 8.0.1
- [vRA Cloud Infoblox 外掛程式版本 0.1](#) - 支援 vRealize Automation 8.0

#### 必要條件

- 確認您具有雲端管理員認證。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 確認您擁有外部 IPAM 提供者的帳戶，並擁有對組織 IPAM 提供者帳戶的正確存取認證。

- 確認您可以存取 IPAM 提供者的已部署整合套件。已部署的套件最初是從 IPAM 提供者網站或從 VMware Solutions Exchange Marketplace 以 .zip 形式下載，然後部署至 vRealize Automation。如需如何下載和部署提供者套件 .zip 檔案並在 [IPAM 整合] 頁面上將其做為提供者值使用的相關資訊，請參閱[下載並部署要在 vRealize Automation Cloud Assembly 中使用的外部 IPAM 提供者套件](#)。
- 確認您可以存取為 IPAM 提供者設定的執行環境。執行環境通常是以動作為基礎的擴充性 (ABX) 內部部署內嵌式整合點。  
如需執行環境特性的相關資訊，請參閱在 [vRealize Automation](#) 中為 IPAM 整合點建立執行環境。
- 在 Infoblox 應用程式中啟用所需的可擴充屬性。請參閱在 [Infoblox 應用程式中新增必要的可擴充屬性，以與 vRealize Automation 整合](#)。
- 如果沒有外部網際網路存取權，您可以設定網際網路伺服器 Proxy。請參閱[如何為 vRealize Automation 設定網際網路 Proxy 伺服器](#)。

## 程序

- 1 選取**基礎結構 > 連線 > 整合**，然後按一下**新增整合**。
- 2 按一下 **IPAM**。
- 3 在**提供者**下拉式清單中，從清單中選取已設定的 IPAM 提供者套件，例如 *Infoblox\_hrg*。  
如果清單為空白，請按一下**匯入提供者套件**，導覽至現有提供者套件 .zip 檔案，然後加以選取。如果沒有提供者 .zip 檔案，您可以從 IPAM 提供者的網站或 vRealize Automation Cloud Assembly **市集** 索引標籤取得該檔案。  
如需如何在 vCenter 中部署提供者套件 .zip 檔案並在 [整合] 頁面上將其做為提供者值使用的相關資訊，請參閱[下載並部署要在 vRealize Automation Cloud Assembly 中使用的外部 IPAM 提供者套件](#)。  
如需有關如何升級現有 IPAM 整合以使用更新版本的廠商 IPAM 整合套件的資訊，請參閱[如何在 vRealize Automation 中升級到較新的 IPAM 整合套件](#)。
- 4 輸入您的外部 IPAM 提供者帳戶的管理員使用者名稱和密碼認證，以及所有其他 (如有) 必填欄位 (例如提供者的主機名稱)。  
在此範例中，使用下列步驟取得 Infoblox IPAM 提供者的主機名稱：
  - a 在單獨的瀏覽器索引標籤中，使用 Infoblox 管理員認證登入 IPAM 提供者帳戶。
  - b 複製您的主機名稱 URL。
  - c 將主機名稱 URL 貼到 [IPAM 整合] 頁面上的**主機名稱**欄位中。

- 5 在**執行環境**下拉式清單中，選取現有的以內部部署動作為基礎的擴充性整合點，例如 *Infoblox\_abx\_intg*。

執行環境支援 vRealize Automation 與外部 IPAM 提供者之間的通訊。

---

**備註** 如果您使用 Amazon Web Services 或 Microsoft Azure 雲端帳戶做為整合執行環境，請確保 IPAM 提供者應用裝置可從網際網路進行存取，並且不在 NAT 或防火牆後方，且具有可公開解析的 DNS 名稱。如果 IPAM 提供者不可存取，則無法將 Amazon Web Services Lambda 或 Microsoft Azure 函數連線到該提供者，整合將會失敗。如需相關資訊，請參閱在 [vRealize Automation](#) 中為 **IPAM 整合點建立執行環境**。

---

IPAM 架構僅支援以動作為基礎的擴充性 (ABX) 內部部署內嵌式執行環境。

---

**備註** Infoblox IPAM 整合點需要以動作為基礎的擴充性 (ABX) 內部部署內嵌式整合點。

---

設定的雲端帳戶或整合點允許透過相關聯的雲端擴充性 Proxy 在 vRealize Automation 與 IPAM 提供者 (在此範例中為 Infoblox) 之間進行通訊。您可以選取已建立的提供者，也可以建立一個提供者。

如需如何建立執行環境的相關資訊，請參閱在 [vRealize Automation](#) 中為 **IPAM 整合點建立執行環境**。

- 6 按一下**驗證**。

由於此範例將以內部部署動作為基礎的擴充性整合用於執行環境，因此，您可以檢視驗證動作。

- a 按一下**擴充性索引**標籤。
- b 按一下**活動 > 動作執行**，然後從篩選器中選取**所有執行**或**整合執行**，以說明端點驗證動作已啟動且正在執行。

- 7 當系統提示您信任來自 IPAM 提供者的自我簽署憑證時，請按一下**接受**。

接受自我簽署的憑證後，可以繼續完成驗證動作。

- 8 輸入此 IPAM 整合點的**名稱** (例如 *Infoblox\_Integration*) 以及**說明** (例如 *Infoblox IPAM with ABX integration for team HRG*)。

- 9 按一下**新增**以儲存新的外部 IPAM 整合點。

將啟動資料收集動作。將從 IPAM 提供者收集網路和 IP 範圍的資料。您可以按如下所示檢視資料收集動作：

- a 按一下**擴充性索引**標籤。
- b 按一下**活動 > 動作執行**，並注意資料收集動作已啟動且正在執行。您可以開啟並檢視動作執行內容。

## 結果

提供者特定的外部 IPAM 整合現在可與網路和網路設定檔一起使用。

## 在 vRealize Automation Cloud Assembly 中將網路和網路設定檔設定為使用 IPAM 提供者值

您可以將網路定義為使用從外部 IPAM 提供者 (而非從 vRealize Automation Cloud Assembly 內部) 取得並由其進行管理的 IP 位址值。

您可以定義網路以存取您在組織的外部 IPAM 提供者帳戶中定義的現有 IP 設定。此步驟會展開您在上一個步驟中建立的 Infoblox 提供者整合。

在此範例中，您可以使用從 vCenter 收集的現有網路資料來設定網路設定檔。然後，您可以將這些網路設定為從外部 IPAM 提供者取得 IP 資訊 (在此案例中為 Infoblox)。從 vRealize Automation Cloud Assembly 佈建的可與此網路設定檔相符的虛擬機器，會從外部 IPAM 提供者取得其 IP 及其他 TCP/IP 相關設定。

如需有關網路的詳細資訊，請參閱[網路資源](#)。如需有關網路設定檔的詳細資訊，請參閱[如何新增 vRealize Automation Cloud Assembly 網路設定檔](#)和[進一步瞭解 vRealize Automation Cloud Assembly 中的網路設定檔](#)。

### 必要條件

此步驟順序會顯示在 IPAM 提供者整合工作流程的內容中。請參閱[提供者特定的外部 IPAM 整合使用案例](#)。

- 確認您具有雲端管理員認證。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 確認您擁有外部 IPAM 提供者的帳戶，例如 [Infoblox](#) 或 [Bluecat](#)，並且您擁有對組織的 IPAM 提供者帳戶的正確存取認證。在此範例工作流程中，IPAM 提供者為 Infoblox。
- 確認您具有 IPAM 提供者的 IPAM 整合點。請參閱在 [vRealize Automation 中新增外部 IPAM 整合點](#)。

### 程序

- 1 若要設定網路，請按一下 **基礎結構 > 資源 > 網路**。
- 2 在 **網路索引** 標籤上，選取要與 IPAM 提供者整合點搭配使用的現有網路。在此範例中，網路名稱為 *net.23.117-only-IPAM*。

vRealize Automation Cloud Assembly 已從您組織中的 vCenter 收集列出的網路資料。

- 3 若要從外部 IPAM 提供者取得值，請確認除 **帳戶/區域**、**名稱** 和 **網路網域** 以外，所有其他網路設定均為空白，包括下列內容：
  - 網域 (請參閱步驟 8 中的附註)
  - CIDR
  - 預設閘道
  - DNS 伺服器
  - DNS 搜尋網域

- 4 按一下 **IP 範圍** 索引標籤，然後按一下 **新增 IPAM IP 範圍**。
- 5 從 **網路** 功能表中，選取您剛設定的網路，例如 *net.23.117-only-IPAM*。
- 6 從 **提供者** 功能表中，選取您先前在工作流程中建立的 *Infoblox\_Integration* IPAM 整合點
- 7 從現在可見的 **位址空間** 下拉式功能表中，選取其中一個列出的網路視圖。

Infoblox 中的位址空間稱為網路視圖。

將從您的 IPAM 提供者帳戶取得網路視圖。此範例使用您剛設定的網路子網路，例如 *net.23.117-only-IPAM*、您先前在工作流程中建立的 *Infoblox\_Integration* 整合點，以及名為 *default* 的位址空間。

將從外部 IPAM 提供者取得列出的位址空間值。

- 8 從可用於所選位址空間的已顯示網路清單中，選取一或多個網路，例如，選取 10.23.117.0/24。

在此範例中，所選網路的 **網域** 和 **DNS 伺服器** 資料行值包含來自 Infoblox 的值。

---

**備註** 如果您在步驟 3 中選取的網路具有為 vRealize Automation Cloud Assembly 指定的網域，然後從包含網域值的外部 IPAM 提供者位址空間中選取網路，則外部 IPAM 提供者網路中的網域值優先於 vRealize Automation Cloud Assembly 中指定的網域。如果 IPAM IP 範圍設定不具有在 Cloud Assembly 中或在外部 IPAM 提供者中指定的網域值，則佈建會失敗。

---

- 9 按一下 **新增** 以儲存網路的 IPAM IP 範圍。

範圍會顯示在 **IP 範圍** 資料表中。

- 10 按一下 **IP 位址** 索引標籤。

使用外部 IPAM 提供者中的新位址範圍佈建機器後，新記錄將顯示在 **IP 位址** 資料表中。

- 11 若要設定網路設定檔以使用該網路，請按一下 **基礎結構 > 設定 > 網路設定檔**。

- 12 為網路設定檔命名，例如 *Infoblox-NP*，然後新增下列範例設定。

- **[摘要] 索引標籤**

- 指定 vSphere 雲端帳戶/區域。
- 為網路設定檔新增功能標籤，例如，名為 *infoblox\_abx* 的標籤。

請記下此功能標籤，因為您還必須使用它做為藍圖限制標籤，以便在藍圖中進行佈建關聯。

- **[網路] 索引標籤**

- 新增您先前建立的網路，例如 *net.23.117-only-IPAM*。

- 13 按一下 **儲存**，以使用這些設定儲存網路設定檔。

## 結果

網路和網路設定檔設定現在已設定為支援外部 IPAM 整合，並且可在藍圖中使用。

## 定義和部署使用 IPAM 提供者範圍指派的 vRealize Automation Cloud Assembly 藍圖

您可以定義藍圖，以從外部 IPAM 提供者取得和管理 IP 位址指派。



在外部 IPAM 整合工作流程的最後一個步驟中，您可以定義和部署將先前定義的網路和網路設定檔連線至組織的 Infoblox 帳戶的藍圖，以從外部 IPAM 提供者 (而非 vRealize Automation Cloud Assembly) 取得和管理已部署虛擬機器的 IP 位址指派。

此工作流程使用 Infoblox 做為外部 IPAM 提供者，在某些步驟中，範例值對 Infoblox 而言是唯一的，不過目的是為了讓此程序能夠套用至其他外部 IPAM 整合。

部署藍圖並啟動虛擬機器後，部署中每個虛擬機器所使用的 IP 位址將顯示為**資源 > 網路**頁面中的網路項目、IPAM 提供者帳戶的 IPAM 提供者網路中的新主機記錄，以及主機 vCenter 中每個已部署虛擬機器的 vSphere Web Client 記錄中的新主機記錄。

#### 必要條件

這一系列步驟會顯示在外部 IPAM 提供者整合工作流程的內容中。請參閱[提供者特定的外部 IPAM 整合使用案例](#)。

- 確認您具有雲端管理員認證。請參閱在 [vRealize Automation 中使用雲端帳戶所需的認證](#)。
- 確認您具有雲端管理員使用者角色。請參閱[什麼是 vRealize Automation Cloud Assembly 使用者角色](#)。
- 確認您擁有外部 IPAM 提供者的帳戶，例如 Infoblox 或 BlueCat，並且您擁有對組織的 IPAM 提供者帳戶的正確存取認證。
- 確認您擁有主機帳戶的管理員存取權，以及在 vSphere Web Client 記錄中針對主機 vCenter 中已部署的虛擬機器顯示狀態記錄所需的任何角色需求。
- 確認您具有用於外部 IPAM 提供者的 IPAM 整合點。請參閱在 [vRealize Automation 中新增外部 IPAM 整合點](#)。
- 確認您已設定 vRealize Automation Cloud Assembly 網路和網路設定檔，以支援預期 IPAM 整合點的外部 IPAM 整合。請參閱在 [vRealize Automation Cloud Assembly 中將網路和網路設定檔設定為使用 IPAM 提供者值](#)。
- 確認您的專案和雲端區域已標記為符合 IPAM 整合點和網路或網路設定檔中的標籤。選擇性地設定專案以支援自訂資源命名。

如需有關專案和雲端區域角色以及藍圖中其他基礎結構元素角色的詳細資訊，請參閱 [WordPress 使用案例](#)。如需有關標記的詳細資訊，請參閱[如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署](#)。

如需使用專案中的設定自訂命名虛擬機器的相關資訊，請參閱[如何使用 vRealize Automation Cloud Assembly 自訂已部署資源的名稱](#)。

#### 程序

1 按一下**藍圖 > 新增**，在新**藍圖**頁面中輸入下列資訊，然後按一下**建立**。

- **名稱** = ipam-bpa
- **說明** = 使用 Infoblox IPAM 整合的藍圖
- **專案** = 123VC

- 2 在此範例中，將非雲端機器元件和非雲端網路元件新增至藍圖畫布，並連線這兩個元件。
- 3 編輯藍圖程式碼，將限制標籤新增到與已新增至網路設定檔的功能標籤相符的網路元件。在此範例中，標籤值為 *infoblox\_abx*。
- 4 編輯藍圖程式碼，以指定網路指派類型為靜態。

在此範例中，已知目前在為相關聯網路設定檔中選取的網路外部 IPAM 位址空間中提供指定的 IP 位址 10.23.117.4。雖然靜態指派設定是必要的，但位址值並非必要。如果您想要在特定的位址啟動外部 IP 位址選取，您可以將其啟動，但這不是必要的。如果您未指定位址值，則外部 IPAM 提供者會選取外部 IPAM 網路中的下一個可用位址。

- 5 對照下列範例驗證藍圖程式碼。

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      networkType: existing
      name: ipam
      限制:
        - 標籤: infoblox_abx
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: ubuntu
      flavor: small
      networks:
        - network: '${resource.Cloud_Network_1.id}'
          指派: 靜態
          位址: 10.23.117.4
          name: '${resource.Cloud_Network_1.name}'
```

**備註** 依預設，Infoblox 整合會在 Infoblox 的預設 DNS 視圖中建立 DNS 主機記錄。

如果 Infoblox 管理員已建立自訂 DNS 視圖，則可以覆寫預設整合行為，並使用機器元件中的 Infoblox.IPAM.Network.dnsView 內容指定一個具名視圖。例如，您可以將下列內容新增至 Cloud\_Machine\_1 元件，以在 Infoblox 中指定具名 DNS 視圖。

```
Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    image: ubuntu
    flavor: small
    Infoblox.IPAM.Network.dnsView:<dns-view-name>
```

如需設定和使用 DNS 視圖的相關資訊，請參閱 Infoblox 產品說明文件中的 [DNS 視圖](#)。

- 6 在 [藍圖] 頁面上按一下部署，為部署 *Infoblox-1* 命名，然後在部署類型頁面上按一下部署。



- 7 在部署藍圖時，按一下**擴充性**索引標籤，然後選取**活動 > 動作執行**，以查看 *Infoblox\_AllocateIP\_n* 擴充性動作執行。

在擴充性動作完成並佈建機器後，*Infoblox\_Update\_n* 動作會將 MAC 位址散佈到 Infoblox。

- 8 您可以登入並開啟 Infoblox 帳戶，以便在相關聯的 10.23.117.0/24 網路中查看 IPAM 位址的新主機記錄。您也可以在此 Infoblox 中開啟 DNS 索引標籤，以查看新的 DNS 主機記錄。
- 9 若要確認是否已佈建虛擬機器，請登入主機 vCenter 和 vSphere Web Client 找到已佈建的機器，並檢視 DNS 名稱和 IP 位址。  
啟動已佈建的虛擬機器後，MAC 位址會透過 *Infoblox\_AllocateIP* 擴充性動作散佈至 Infoblox。
- 10 若要在 vRealize Automation Cloud Assembly 中檢視新的網路記錄，請選取**基礎結構 > 資源 > 網路**，然後按一下以開啟 **IP 位址**索引標籤。
- 11 如果您刪除部署，則會釋放該部署中的虛擬機器的 IPAM 位址，且 IP 位址會再次供外部 IPAM 提供者用於其他配置。此事件在 vRealize Automation Cloud Assembly 中的擴充性動作為 *Infoblox\_Deallocate*。

## 在 vRealize Automation 中針對 IPAM 整合使用 Infoblox 特定內容和可擴充屬性

您可以針對包含用於 Infoblox 的外部 IPAM 整合的 vRealize Automation 專案使用 Infoblox 特定內容。

下列 Infoblox 內容可與 Infoblox IPAM 整合搭配使用。您可以在 vRealize Automation 中使用它們來進一步控制藍圖部署期間的 IP 位址配置。這些內容的使用是選擇性的。

雖然下列所有 Infoblox 內容均可用於適用於 vRealize Automation 8.0 的 [vRA Cloud Infoblox 外掛程式版本 0.1](#) 套件，但只有 `Infoblox.IPAM.Network.dnsView` 內容可與適用於 vRealize Automation 8.0.1 的 [vRA Cloud Infoblox 外掛程式版本 1.0](#) 和適用於 vRealize Automation 8.1 的 [vRA Cloud Infoblox 外掛程式版本 1.1](#) 套件搭配使用。

---

**備註** 提供者特定的外部 IPAM 整合使用案例範例工作流程中不包括這些內容的使用。

---

- `Infoblox.IPAM.createFixedAddress`

此內容可讓您在 Infoblox 內建立固定位址記錄。可能的值為 True 和 False。依預設，會建立主機記錄。預設值：False。

- `Infoblox.IPAM.Network.dnsView`

此內容可讓您在 Infoblox 內建立主機記錄時使用 DNS 視圖。預設值：default。

- `Infoblox.IPAM.Network.enableDns`

在 Infoblox 中配置 IP 時，此內容可讓您同時建立 DNS 記錄。可能的值為 True 和 False。預設值：True。

- `Infoblox.IPAM.Network.dnsSuffix`

此內容可讓您將 Infoblox 網路的 *domain* DHCP 選項覆寫為新選項。如果 Infoblox 網路未設定 *domain* DHCP 選項，或者必須覆寫 *domain* DHCP 選項，則此功能非常有用。預設值：None (空白字串)。

僅當 `Infoblox.IPAM.Network.enableDns` 設為 `True` 時，  
`Infoblox.IPAM.Network.dnsSuffix` 才適用。

您可以在 vRealize Automation Cloud Assembly 中使用下列其中一種方法來指定 Infoblox 內容：

- 您可以使用 **基礎結構 > 設定 > 專案** 頁面上的 **自訂內容** 區段來指定專案中的內容。透過使用此方法，指定的內容將會套用到在此專案範圍內佈建的所有機器。
- 您可以在藍圖中的每個機器元件上指定內容。以下是說明 `Infoblox.IPAM.Network.dnsView` 內容使用方法的範例藍圖代碼：

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      Infoblox.IPAM.Network.dnsView: default
      image: ubuntu
      cpuCount: 1
      totalMemoryMB: 1024
      networks:
        - network: '${resource.Cloud_Network_1.id}'
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      networkType: existing
      constraints:
        - tag: mk-ipam-demo
```

- 您可以使用擴充性訂閱指定內容。

如需與此使用案例相關的 Infoblox 可擴充屬性的相關資訊，請參閱在 [Infoblox 應用程式](#) 中新增必要的可擴充屬性，以與 vRealize Automation 整合。

# 建置 vRealize Automation Cloud Assembly 資源基礎結構

# 4

在 vRealize Automation Cloud Assembly 資源基礎結構中，您可以將雲端帳戶區域定義為可在其中部署藍圖及其工作負載的區域。

此外，資源基礎結構還涉及建立映像和機器大小的通用對應，以及建立定義整個雲端帳戶區域或資料中心的網路和儲存區功能的設定檔。

本章節討論下列主題：

- 如何新增定義 vRealize Automation Cloud Assembly 目標放置區域或資料中心的雲端區域
- 如何新增 vRealize Automation Cloud Assembly 類型模板對應以在 vRealize Automation Cloud Assembly 中建立通用機器大小
- 如何新增 vRealize Automation Cloud Assembly 映像對應以建立一般作業系統
- 如何新增 vRealize Automation Cloud Assembly 網路設定檔
- 如何新增因應不同需求的 vRealize Automation Cloud Assembly 儲存區設定檔
- 如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署
- 如何使用 vRealize Automation Cloud Assembly 中的資源

## 如何新增定義 vRealize Automation Cloud Assembly 目標放置區域或資料中心的雲端區域

vRealize Automation Cloud Assembly 雲端區域是雲端帳戶類型 (例如 AWS 或 vSphere) 中的一組資源。

特定帳戶區域中的雲端區域是您的藍圖部署工作負載的位置。每個雲端區域都與 vRealize Automation Cloud Assembly 專案相關聯。

選取**基礎結構 > 設定 > 雲端區域**，然後按一下**新增區域**。

## 進一步瞭解 vRealize Automation Cloud Assembly 雲端區域

vRealize Automation Cloud Assembly 雲端區域是您的雲端帳戶類型 (例如 AWS 或 vSphere) 中的部分。雲端區域是特定於一個專案的。

其他放置控制項包括放置原則選項、功能標籤和計算標籤。

- 放置原則

放置原則有助於在指定區域內選取主機。

- 預設值 - 將計算資源隨機放置在主機上。
- binpack - 將計算資源置於負載最多，且有足夠的資源來執行指定計算資源的主機上。
- spread - 將計算資源平均放置在主機上。

## ■ 功能標籤

藍圖包含協助您判斷部署放置的限制標籤。在部署期間，藍圖限制標籤將對應到與雲端區域中的相符功能標籤，以判定適用於計算資源放置的雲端區域。

## ■ 計算資源

您可以檢視和管理可佈建到此雲端區域的計算資源。

如果 vCenter 運算叢集已啟用 DRS，則雲端區域僅顯示計算資源清單中的叢集，不會顯示子系主機。

如果 vCenter 運算叢集未啟用 DRS，則雲端區域僅顯示獨立 ESXi 主機 (如果存在)。

計算資源標籤有助於進一步控制放置。您可以使用標籤來篩選可用的計算資源，以便僅列出與一或多個標籤相符的計算資源，如下列範例所示。

- 計算資源不包含任何標籤，並且未使用任何篩選。



- 兩個計算資源包含相同的標籤，但未使用任何篩選。



- 兩個計算資源包含相同的標籤，並且標籤篩選器與兩個計算資源上使用的標籤相符。



## ■ 專案

您可以檢視哪些專案已設定為支援將工作負載佈建到此雲端區域。

建立雲端區域後，您可以驗證其組態。

## 如何新增 vRealize Automation Cloud Assembly 類型模板對應以在 vRealize Automation Cloud Assembly 中建立通用機器大小

在 vRealize Automation Cloud Assembly 類型模板對應中，您可以使用自然語言為特定的雲端帳戶/區域定義目標部署大小。

類型模板對應表示對您的環境有意義的部署大小。一個範例可能是，對於已命名資料中心中的 vCenter 帳戶，以及已命名區域中的 Amazon Web Services 帳戶的 t2.nano 執行個體，small 表示 1 個 CPU 和 2 GB 記憶體，large 表示 2 個 CPU 和 8 GB 記憶體。

選取**基礎結構 > 設定 > 類型模板對應**，然後按一下**新增類型模板對應**。

## 進一步瞭解 vRealize Automation Cloud Assembly 中的類型模板對應

類型模板對應使用自然語言命名對 vRealize Automation Cloud Assembly 中的特定雲端帳戶/區域的一組目標部署大小進行分組。

類型模板對應可讓您建立一個具名對應，該對應包含跨帳戶區域的類似類型模板大小。例如，名為 standard\_small 的類型模板對應可能包含專案中部分或全部可用帳戶/區域的類似類型模板大小 (例如 1 個 CPU，2 GB RAM)。建置藍圖時，選取符合您需求的可用類型模板。

按部署目的組織整理專案的類型模板對應。

若要簡化藍圖建立，您可以在新增雲端帳戶時選取預先設定選項。選取預先設定選項時，將選取指定區域的組織最常用的類型模板對應和映像對應。

對於包含 vSphere 資源的藍圖中的映像對應，如果沒有為 vSphere 雲端區域定義類型模板對應，則可以使用藍圖中特定於 vSphere 的設定來設定無限制的記憶體和 CPU。如果為 vSphere 雲端區域定義了類型模板對應，則類型模板對應將用作藍圖中特定於 vSphere 之組態的限制。

如需基本類型模板對應範例，請參閱 [WordPress 使用案例：新增類型模板對應](#)。

## 如何新增 vRealize Automation Cloud Assembly 映像對應以建立一般作業系統

在 vRealize Automation Cloud Assembly 映像對應中，您可以使用自然語言為特定的雲端帳戶/區域定義目標部署作業系統。

選取**基礎結構 > 設定 > 映像對應**，然後按一下**新增映像對應**。

### 進一步瞭解 vRealize Automation Cloud Assembly 中的映像對應

映像對應使用自然語言命名對 vRealize Automation Cloud Assembly 中的特定雲端帳戶/區域的一組預先定義的目標作業系統規格進行分組。

Microsoft Azure 和 Amazon Web Services 等雲端廠商帳戶使用映像將一組目標部署條件分組到一起，其中包括作業系統和相關組態設定。基於 vCenter 和 NSX 的環境 (包括 VMware Cloud on AWS) 使用類似的分組機制來定義一組作業系統部署條件。建置並最終部署和反覆查看藍圖時，選取最符合您需求的可用映像。

按類似的作業系統設定、標記策略和正常運作的部署意圖組織整理專案的映像對應。

如需如何定義基本映像對應的範例，請參閱 [WordPress 使用案例：新增映像對應](#)。

若要簡化藍圖建立，您可以在新增雲端帳戶時選取預先設定選項。選取預先設定選項時，將選取指定區域的組織最常用的類型模板對應和映像對應。

將映像資訊新增至藍圖時，您可以使用機器元件的 `properties` 區段中的 `image` 或 `imageRef` 項目。例如，如果要從快照複製，請使用 `imageRef` 內容。

如需藍圖程式碼中 `image` 和 `imageRef` 項目的範例，請參閱 [第 6 章 設計 vRealize Automation Cloud Assembly 部署](#)。

若要指派內容程式庫的權限，管理員必須將該權限做為全域權限授與使用者。如需相關資訊，請參閱 [VMware vSphere 說明文件](#) 之《vSphere 虛擬機器管理》中的〈[內容程式庫權限的階層式繼承](#)〉。

### 同步雲端帳戶/區域的映像

您可以執行映像同步化，以確保您要在**基礎結構 > 設定 > 映像對應**頁面上為指定的雲端帳戶/區域新增或移除的映像是最新的。

- 1 若要開啟相關聯的**雲端帳戶/區域**，請選取**基礎結構 > 連線 > 雲端帳戶**。選取現有的雲端帳戶/區域。
- 2 按一下**同步映像**按鈕，並讓動作完成。



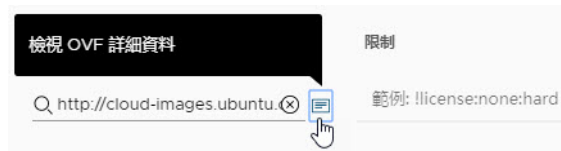
- 3 當動作完成時，按一下**基礎結構 > 設定 > 映像對應**。定義新的映像對應或編輯現有的映像對應，然後從步驟 1 中選取雲端帳戶/區域。
- 4 按一下**映像對應**頁面上的 [映像同步] 圖示。



5 在映像對應頁面上，為指定的雲端帳戶/區域設定映像對應設定。

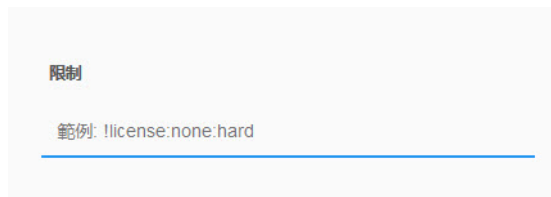
## 檢視 OVF 詳細資料

您可以將 OVF 規格納入 vRealize Automation Cloud Assembly 藍圖物件中，例如 vCenter 機器元件和映像對應。如果您的映像包含 OVF 檔案，您可以在未開啟檔案的情況下探索其內容。將游標暫留在 OVF 上以顯示 OVF 詳細資料，包括其名稱和位置。如需有關 OVF 檔案格式的詳細資訊，請參閱 [vcenter ovf:property](#)。



## 使用限制和標籤細分映像選取

若要在藍圖中進一步細分映像選取，您可以新增一或多項限制，以對可部署的映像類型指定以標籤為基礎的限制。當您建立或編輯映像對應組態時顯示的提供的**限制**範例為 `!license:none:hard`。此範例說明了以標籤為基礎的限制，僅當藍圖中不存在 `license:none` 標籤時，才可以使用映像。如果新增標籤 (例如 `license:88` 和 `license:92`)，則僅當 `license:88` 和 `license:92` 標籤存在於藍圖中時，才能使用指定的映像。



## 使用雲端組態指令碼控制部署

您可以在映像對應和/或藍圖中使用雲端組態指令碼，來定義要在 vRealize Automation Cloud Assembly 部署中使用的自訂作業系統特性。例如，根據您要將藍圖部署到公有雲或私有雲，您可以對映像套用特定的使用者權限、作業系統權限或其他條件。雲端組態指令碼遵守 `cloud-init` 格式 (Linux 系統的映像) 或 `cloudbase-init` 格式 (Windows 系統的映像)。vRealize Automation Cloud Assembly 支援適用於 Linux 系統的 `cloud-init` 工具和適用於 Windows 系統的 `cloudbase-init` 工具。

對於 Windows 機器，您可以使用 `cloudbase-init` 支援的任何雲端組態指令碼格式。

下列範例藍圖程式碼中的機器資源使用包含雲端組態指令碼的映像，該映像的內容顯示在 `image` 項目中。

```
resources:
  demo-machine:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      image: MyUbuntu16
```



```

https://cloud-images.ubuntu.com/releases/16.04/release-20170307/ami-
ubuntu-16.04-1.10.3-00-15269239.ova
cloudConfig: |
  ssh_pwauth: yes
  chpasswd:
    list: |
      ${input.username}:${input.password}
    expire: false
  users:
    - default
    - name: ${input.username}
      lock_passwd: false
      sudo: ['ALL=(ALL) NOPASSWD:ALL']
      groups: [wheel, sudo, admin]
      shell: '/bin/bash'
  runcmd:
    - echo "Defaults:${input.username} !requiretty" >> /etc/sudoers.d/${input.username}

```

## 映像對應和藍圖包含雲端組態指令碼時會發生什麼情況

當包含雲端組態指令碼的藍圖使用映像對應且該對應包含雲端組態指令碼時，則會組合使用兩個指令碼。合併動作會先處理映像對應指令碼的內容，接著處理藍圖指令碼的內容，同時考慮是否將指令碼設為 `#cloud-config` 格式。

- 對於採用 `#cloud-config` 格式的指令碼，合併動作將會合併每個模組的內容 (例如 `runcmd`、`users` 和 `write_files`)，如下所示：
  - 對於內容為清單的模組，會合併映像對應和藍圖中的命令清單，以排除兩個清單中相同的命令。
  - 對於內容為字典的模組，會合併命令，並且結果是兩個字典的組合。如果兩個字典中都存在相同的金鑰，則會保留映像對應指令碼字典中的金鑰，並忽略藍圖指令碼字典中的金鑰。
  - 對於內容為字串的模組，會保留映像對應指令碼中的內容值，並忽略藍圖指令碼中的內容值。
- 對於未採用 `#cloud-config` 格式的指令碼，或者一個指令碼採用 `#cloud-config` 格式，而另一個指令碼未採用此格式，則會按照以下方式合併兩個指令碼：先執行映像對應指令碼，然後在映像對應指令碼完成時執行藍圖指令碼。

如需相關資訊，請參閱[合併使用者資料區段](#)。

## 有關設定和使用雲端組態指令碼的詳細資訊

若要在 vRealize Automation Cloud Assembly 中為 Windows 部署設定 `cloud-init`，請參閱[如何在 vRealize Automation 中使用 cloud-init 或 cloudbase-init 設定 Windows 範本](#)。

如需有關使用雲端組態指令碼的詳細資訊，請參閱[如何在 vRealize Automation Cloud Assembly 藍圖中自動初始化機器](#)和 VMware 部落格文章[使用 Cloud-Init 自訂 Cloud Assembly 部署](#)。

## 如何新增 vRealize Automation Cloud Assembly 網路設定檔

vRealize Automation Cloud Assembly 網路設定檔說明了要部署之網路的行為。

例如，網路可能需要面向網際網路，而不是僅供內部使用。網路及其設定檔是特定於雲端的。



選取**基礎結構 > 設定 > 網路設定檔**，然後按一下**新增網路設定檔**。

## 進一步瞭解 vRealize Automation Cloud Assembly 中的網路設定檔

網路設定檔定義適用於 vRealize Automation Cloud Assembly 中特定區域或資料中心內的雲端帳戶的一組網路和網路設定。

通常，您可以定義網路設定檔以支援目標部署環境，例如小型測試環境 (現有網路於其中僅擁有輸出存取權) 或需要一組安全性原則的大型負載平衡生產環境。將網路設定檔視為工作負載特定網路特性的集合。

### 網路設定檔的內容

網路設定檔包含 vRealize Automation Cloud Assembly 中具名雲端帳戶類型和區域的特定資訊，包括下列設定：

- 網路設定檔的具名雲端帳戶/區域和選擇性功能標籤。
- 具名現有網路及其設定。
- 定義網路設定檔的隨選和其他方面的網路原則。
- (選擇性) 包含現有負載平衡器。
- (選擇性) 包含現有安全群組。

您可以根據網路設定檔判斷網路 IP 管理功能。

網路設定檔功能標籤與藍圖中的限制標籤相符，可協助控制網路選擇。此外，指派給網路設定檔所收集網路的所有標籤還與藍圖中的標籤相符，以協助您在部署藍圖時控制網路選擇。

功能標籤是選擇性的。功能標籤會套用到網路設定檔中的所有網路，但僅在網路用做該網路設定檔的一部分時才會套用。對於不包含功能標籤的網路設定檔，僅針對網路標籤執行標記比對。部署藍圖時，會套用相符網路設定檔中定義的網路和安全性設定。

使用靜態 IP 時，位址範圍由 vRealize Automation 管理。對於 DHCP，IP 起始和結束位址由獨立 DHCP 伺服器管理，而不是由 vRealize Automation 管理。使用 DHCP 或混合網路位址配置時，網路使用率值會設定為零。隨選網路配置範圍是根據網路設定檔中指定的 CIDR 和子網路大小而定。若要在部署中同時支援靜態和動態指派，配置的範圍會分為兩個範圍 - 一個用於靜態配置，而另一個用於動態配置。

### 網路

網路 (也稱為子網路) 是 IP 網路的邏輯細分。網路可將雲端帳戶、IP 位址或範圍及網路標籤分組，以控制佈建藍圖部署的方式及位置。設定檔中的網路參數定義部署中的機器如何透過 IP 第 3 層相互通訊。網路可以有標籤。

您可以將網路新增到網路設定檔，編輯網路設定檔所使用網路的各個方面，以及從網路設定檔中移除網路。

#### ■ 網路網域或傳輸區域

網路網域或傳輸區域是 vSphere vNetwork 分散式連接埠群組 (dvPortGroup) 的分散式虛擬交換器 (dvSwitch)。**傳輸區域**是一個現有的 NSX 概念，類似於 *dvSwitch* 或 *dvPortGroup* 等詞彙。

使用 NSX 雲端帳戶時，頁面上的元素名稱為**傳輸區域**，否則為**網路網域**。

對於標準交換器，網路網域或傳輸區域與交換器本身相同。網路網域或傳輸區域定義了 vCenter 中子網路的邊界。

傳輸區域會控制 NSX 邏輯交換器所能連線的主機。它可跨越一或多個 vSphere 叢集。傳輸區域會控制哪些叢集和哪些虛擬機器可以參與特定網路的使用。屬於相同 NSX 傳輸區域的子網路可用於相同的機器主機。

## ■ 網域

目標虛擬機器的 vCenter Single Sign-On 網域。網域由 vCenter 管理員在 vSphere 設定期間進行設定。網域確定了 vCenter 中的本機驗證空間。

## ■ IPv4 CIDR 和 IPv4 預設閘道

vSphere 雲端帳戶和藍圖中的 vSphere 機器元件支援雙 IPv6 和 IPv4 網際網路通訊協定方法。例如，192.168.100.14/24 表示 IPv4 位址 192.168.100.14 及其相關聯的路由前置詞 192.168.100.0，或相當於其子網路遮罩 255.255.255.0，它具有 24 個前置 1 位元。IPv4 區塊 192.168.100.0/22 表示從 192.168.100.0 到 192.168.103.255 的 1024 個 IP 網址。

## ■ IPv6 CIDR 和 IPv6 預設閘道

vSphere 雲端帳戶和藍圖中的 vSphere 機器元件支援雙 IPv6 和 IPv4 網際網路通訊協定方法。例如，2001:db8::/48 表示從 2001:db8:0:0:0:0:0:0 到 2001:db8:0:ffff:ffff:ffff:ffff:ffff 的 IPv6 位址的區塊。

隨選網路不支援 IPv6 格式。

## ■ DNS 伺服器和 DNS 搜尋網域

## ■ 支援公用 IP

選取此選項以將網路標記為公用。藍圖中具有 `network type: public` 內容的網路元件與標記為公用的網路相符。在藍圖部署期間進行進一步比對，以確定網路選擇。

## ■ 區域的預設值

選取此選項以將網路標記為雲端區域的預設網路。在藍圖部署期間，預設網路優先於其他網路。

## ■ 來源

識別網路來源。

## ■ 標籤

指定指派給網路的一或多個標籤。標籤是選擇性的。標籤比對會影響適用於您的藍圖部署的網路。

網路項目本身存在網路標籤，而與網路設定檔不相關。網路標籤套用至已新增該網路標籤的網路的每個事件，以及包含該網路的所有網路設定檔。可以將網路設為任意數目網路設定檔的執行個體。無論網路設定檔在何處，且無論在何處使用網路，網路標籤均與該網路相關聯。

部署藍圖時，藍圖的網路元件中的限制標籤與網路標籤相符，其中包括網路設定檔功能標籤。對於包含功能標籤的網路設定檔，功能標籤會套用到可用於該網路設定檔的所有網路。部署藍圖時，會套用相符網路設定檔中定義的網路和安全性設定。

## 網路原則

根據相關聯的雲端帳戶，您可以使用網路原則為 `outbound`、`private` 和 `routed` 網路類型以及隨選安全群組定義設定。當存在與該網路相關聯的負載平衡器時，也可以使用網路原則控制 `existing` 網路。

如需網路類型的相關資訊，請參閱在 [vRealize Automation Cloud Assembly 中使用網路和網路設定檔](#)。

下列隨選選擇的選項在網路設定檔畫面上的說明中進行說明，並概述如下。

### ■ 不建立隨選網路或隨選安全群組

指定 `existing` 或 `public` 網路類型時，您可以使用此選項。需要 `outbound`、`private` 或 `routed` 網路的藍圖與此設定檔不相符。

### ■ 建立隨選網路

指定 `outbound`、`private` 或 `routed` 網路類型時，您可以使用此選項。

Amazon Web Services、Microsoft Azure、NSX、vSphere 和 VMware Cloud on AWS 支援此選項。

### ■ 建立隨選安全群組

指定 `outbound` 或 `private` 網路類型時，您可以使用此選項。

如果網路類型為 `outbound` 或 `private`，則會為相符的藍圖建立新的安全群組。

Amazon Web Services、Microsoft Azure、NSX 和 VMware Cloud on AWS 支援此選項。

網路原則設定可特定於雲端帳戶類型。這些設定會在畫面上的路標說明中進行說明，並概述如下：

### ■ 網路網域或傳輸區域

網路網域或傳輸區域是 vSphere vNetwork 分散式連接埠群組 (dvPortGroup) 的分散式虛擬交換器 (dvSwitch)。*傳輸區域* 是一個現有的 NSX 概念，類似於 *dvSwitch* 或 *dvPortGroup* 等詞彙。

使用 NSX 雲端帳戶時，頁面上的元素名稱為 **傳輸區域**，否則為 **網路網域**。

對於標準交換器，網路網域或傳輸區域與交換器本身相同。網路網域或傳輸區域定義了 vCenter 中子網路的邊界。

傳輸區域會控制 NSX 邏輯交換器所能連線的主機。它可跨越一或多個 vSphere 叢集。傳輸區域會控制哪些叢集和哪些虛擬機器可以參與特定網路的使用。屬於相同 NSX 傳輸區域的子網路可用於相同的機器主機。

### ■ 外部子網路

具有輸出存取權的隨選網路需要具有輸出存取權的外部子網路。外部子網路將用於在藍圖中申請時提供輸出存取權，而不會控制網路放置。例如，外部子網路不會影響私人網路的放置。

### ■ CIDR

CIDR 標記法是 IP 位址及其相關聯的路由前置詞的精簡表示方式。CIDR 值指定了要在佈建期間用於建立子網路的網路位址範圍。**網路原則** 索引標籤上的此 CIDR 設定可接受以 `/nn` 結尾並包含介於 0 - 32 之間的值的 IPv4 標記法。

### ■ 子網路大小

此選項指定要為使用此網路設定檔的部署中的每個隔離網路建立的隨選網路大小 (使用 IPv4 標記法)。子網路大小設定可用於內部或外部 IP 位址管理。

隨選網路不支援 IPv6 格式。

## ■ 分散式邏輯路由器

針對隨選路由網路，您必須在使用 NSX-V 雲端帳戶時指定分散式邏輯網路。

分散式邏輯路由器 (DLR) 用於在 NSX-V 上的隨選路由網路之間路由東西向流量。僅當網路設定檔的帳戶/區域值與 NSX-V 雲端帳戶相關聯時，才會顯示此選項。

## ■ IP 範圍指派

此選項適用於支援 NSX 或 VMware Cloud on AWS 的雲端帳戶，包括 vSphere。

您可以選取下列三個選項之一來指定部署網路的 IP 範圍指派類型：

### ■ 靜態和 DHCP

預設選項，同時為建議選項。此混合選項使用配置的 CIDR 和子網路範圍設定，將 DHCP 伺服器集區設定為使用 DHCP (動態) 方法支援一半的位址空間配置，使用靜態方法支援另一半 IP 位址空間配置。當連線到隨選網路的部分機器需要指派的靜態 IP 位址，而部分機器需要動態 IP 位址時，請使用此選項。即會建立兩個 IP 範圍。

在機器連線到隨選網路的部署 (其中部分機器指派有靜態 IP，而其他機器則由 NSX DHCP 伺服器動態指派 IP) 以及負載平衡器 VIP 為靜態的部署中，此選項最為有效。

### ■ DHCP (動態)

此選項使用配置的 CIDR 在 DHCP 伺服器上設定 IP 集區。將會動態指派此網路的所有 IP 位址。會為每個已配置的 CIDR 建立單一 IP 範圍。

### ■ 靜態

此選項使用配置的 CIDR 以靜態配置 IP 位址。如果不需要為此網路設定 DHCP 伺服器，請使用此選項。會為每個已配置的 CIDR 建立單一 IP 範圍。

## ■ 網路資源 - 外部網路

外部網路也稱為現有網路。這些網路已進行資料收集並可供選取。

## ■ 網路資源 - 第 0 層邏輯路由器

NSX-T 使用第 0 層邏輯路由器做為 NSX 部署外部網路的閘道。第 0 層邏輯路由器為隨選網路設定輸出存取權。

## ■ 網路資源 - Edge 叢集

指定的 Edge 叢集提供路由服務。Edge 叢集用來設定隨選網路和負載平衡器的輸出存取權。將會識別要部署 Edge 應用裝置的 Edge 叢集或資源集區。

## ■ 網路資源 - Edge 資料存放區

指定的 Edge 資料存放區將用於佈建 Edge 應用裝置。此設定僅適用於 NSX-V。

## 負載平衡器

您可以在網路設定檔中新增負載平衡器。列出的負載平衡器根據從來源雲端帳戶收集的資料資訊提供。

如果網路設定檔中任何負載平衡器上的標籤與藍圖中負載平衡器元件中使用的標籤相符，則會在部署期間考慮負載平衡器。部署藍圖時，會使用相符網路設定檔中的負載平衡器。

如需詳細資訊，請參閱在 [vRealize Automation Cloud Assembly 中的網路設定檔和藍圖設計中使用負載平衡器設定](#) 和 [vRealize Automation Cloud Assembly 藍圖中的網路、安全性和負載平衡器範例](#)。

## 安全群組

部署藍圖時，其網路設定檔中的安全群組會套用到已佈建的機器 NIC。對於特定於 Amazon Web Services 的網路設定檔，網路設定檔中的安全群組在 [網路] 索引標籤上列出的網路所在的網路網域 (VPC) 中可用。如果網路設定檔在其 [網路] 索引標籤上未列出任何網路，則會顯示所有可用的安全群組。

您可以使用安全群組為隨選 `private` 或 `outbound` 網路進一步定義隔離設定。安全群組也適用於 `existing` 網路。

安全群組會套用到部署中連線至與網路設定檔相符的網路的所有機器。由於藍圖中可能存在多個網路，每個網路都與其他網路設定檔相符，因此您可以對不同的網路使用不同的安全群組。

透過將標籤新增至現有安全群組，您可以在藍圖 `Cloud.SecurityGroup` 元件中使用此安全群組。安全群組必須至少有一個標籤，否則無法在藍圖中使用。如需詳細資訊，請參閱 [安全資源](#) 和 [vRealize Automation Cloud Assembly 藍圖中的網路、安全性和負載平衡器範例](#)。

## 有關網路設定檔、網路、藍圖和標籤的詳細資訊

如需有關網路設定檔的詳細資訊，請參閱此部分說明中的其他主題以及 [WordPress 使用案例：新增網路設定檔](#)。

如需有關網路的詳細資訊，請參閱 [網路資源](#)。

如需藍圖中範例網路元件代碼的範例，請參閱 [vRealize Automation Cloud Assembly 藍圖中的網路、安全性和負載平衡器範例](#)。

如需範例網路自動化工作流程，請參閱 [〈使用 Cloud Assembly 和 NSX 實現網路自動化〉](#)。

如需有關標籤和標籤策略的詳細資訊，請參閱 [如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署](#)。

## 在 vRealize Automation Cloud Assembly 中使用網路和網路設定檔中的 IP 位址

透過使用網路和網路設定檔設定，您可以控制在 vRealize Automation Cloud Assembly 藍圖和部署中使用網路 IP 位址的方式。

透過使用網路設定檔，您可以為包含靜態、DHCP 或混合靜態和 DHCP IP 位址設定的現有網路網域定義子網路。

您可以使用 [網路原則](#) 索引標籤定義子網路並指定 IP 位址設定。如需詳細資訊，請參閱 [進一步瞭解 vRealize Automation Cloud Assembly 中的網路設定檔](#)。

## vRealize Automation Cloud Assembly 網路中的 IPv4 和 IPv6 支援

vRealize Automation Cloud Assembly 網路支援純 IPv4 或雙堆疊 IPv4 和 IPv6。目前不支援純 IPv6。

雖然所有雲端帳戶和整合類型均支援純 IPv4，但僅 vSphere 雲端帳戶及其端點支援雙堆疊 IPv4 和 IPv6。

目前不支援將 IPv6 用於負載平衡器、NSX 隨選網路或外部第三方 IPAM 提供者。

### 外部 IPAM 提供者端點

除了提供的內部 IPAM 支援之外，您還可以使用外部 IPAM 提供者為藍圖和部署中的網路動態或靜態配置 IP 位址。

對外部 IPAM 提供者 (例如 Infoblox) 的支援，可用於您使用 **基礎結構 > 連線 > 新增整合 > IPAM** 功能表順序建立的廠商特定 IPAM 整合點。

可透過**網路原則 > 新增 IPAM IP 範圍**頁面上的**新增 IPAM IP 範圍**選項，使用用於定義第三方 IPAM 提供者位址資訊的選項。

如需如何建立外部 IPAM 整合點的相關資訊，請參閱在 [vRealize Automation 中設定外部 IPAM 整合點](#)。如需如何為特定 IPAM 廠商建立 IPAM 整合點的範例，請參閱[提供者特定的外部 IPAM 整合使用案例](#)。

## 在 vRealize Automation Cloud Assembly 中使用網路和網路設定檔

您可以在 vRealize Automation Cloud Assembly 中使用網路和網路設定檔以協助定義部署的網路佈建行為。

在 vRealize Automation Cloud Assembly 中，您可以定義特定於雲端的網路設定檔。請參閱[進一步瞭解 vRealize Automation Cloud Assembly 中的網路設定檔](#)。

### 網路類型

藍圖中的網路元件定義為下列 `networkType` 類型之一。

網路類型	定義
<code>existing</code>	選取在基礎雲端提供者上設定的現有網路，例如 vCenter、Amazon Web Services 和 Microsoft Azure。 <code>outbound</code> 隨選網路需要現有網路。 您可以在現有網路上定義靜態 IP 位址的範圍。
<code>public</code>	公用網路上的機器可從網際網路進行存取。IT 管理員可定義這些網路。對於公用網路中允許出現網路流量的網路， <code>public</code> 網路的定義與 <code>existing</code> 網路的定義相同。
<code>private</code>	隨選網路類型。 將網路流量限制為僅在已部署網路上的資源之間產生。它可防止輸入和輸出流量。在 NSX 中，它可以等同於隨選 NAT 一對多。

網路類型	定義
outbound	<p>隨選網路類型。</p> <p>將網路流量限制為在部署中的計算資源之間產生，但也允許單向輸出網路流量。在 NSX 中，它可以等同於含外部 IP 的隨選 NAT 一對多。</p>
routed	<p>隨選網路類型。</p> <p>路由網路包含連結在一起的可用子網路之間分割的可路由 IP 空間。使用路由網路佈建且具有相同路由網路設定檔的虛擬機器可彼此通訊以及與現有網路通訊。</p> <p>路由網路是可用於 NSX-V 和 NSX-T 網路的隨選網路類型。Microsoft Azure 和 Amazon Web Services 依預設會提供此連線。</p> <p>routed 網路僅適用於 Cloud.NSX.Network 網路元件中的藍圖規格。</p>

如需包含網路元件資料的已填入藍圖的範例，請參閱 [vRealize Automation Cloud Assembly 藍圖中的網路、安全性和負載平衡器範例](#)。

## 網路案例

當您部署使用下列網路設定檔組態的藍圖時，以下是預期行為。

表 4-1. 網路案例

網路類型或案例	沒有網路設定檔可用於雲端區域	多個網路設定檔可用於雲端區域
無網路	<p>如果在藍圖中未指定任何網路，將從計算資源的相同佈建區域中選取一個隨機網路。</p> <p>喜好設定指定為標記為預設值的網路。</p> <p>如果可用的佈建區域中不存在任何網路，佈建會失敗。</p>	<p>從相符的網路設定檔中選取網路。</p> <p>喜好設定指定為標記為預設值的網路。</p> <p>如果沒有網路設定檔符合準則，佈建會失敗。</p>
現有網路	<p>如果藍圖中的網路元件包含限制標籤，這些限制會用來篩選可用網路清單。藍圖的網路元件中的限制標籤與網路標籤和網路設定檔限制標籤 (如果可用) 相符。</p> <p>從篩選的網路清單中，從計算資源的相同佈建區域中選取單一網路。</p> <p>喜好設定指定為標記為預設值的網路。</p> <p>根據限制進行篩選後，如果佈建區域中沒有任何網路，佈建會失敗。</p>	<p>從相符的網路設定檔中選取網路。</p> <p>喜好設定指定為標記為預設值的網路。</p> <p>如果沒有網路設定檔符合準則，佈建會失敗。</p> <p>可以使用網路限制根據預先指派的標籤篩選設定檔中的現有網路。</p>
公用網路	<p>如果網路有限制，會使用這些限制來篩選設定了 supports public IP 屬性的可用網路清單。</p> <p>從篩選的網路清單中，從與計算資源相同的佈建區域中選取隨機網路。</p> <p>喜好設定指定為標記為預設值的網路。</p> <p>根據限制進行篩選後，如果佈建區域中沒有任何公用網路，佈建會失敗。</p>	<p>從相符的網路設定檔中選取具有 supports public IP 屬性的網路。</p> <p>喜好設定指定為標記為預設值的網路。</p> <p>可以使用網路限制根據預先指派的標籤篩選設定檔中的現有公用網路。</p>

表 4-1. 網路案例 (續)

網路類型或案例	沒有網路設定檔可用於雲端區域	多個網路設定檔可用於雲端區域
私人網路	佈建會失敗，因為私人網路需要來自網路設定檔的資訊。	會根據相符網路設定檔中的設定建立新網路或新安全群組。 可以使用網路限制標籤來篩選網路設定檔和網路。
輸出網路	佈建會失敗，因為輸出網路需要來自網路設定檔的資訊。	會根據相符網路設定檔中的設定建立新網路或新安全群組。 可以使用網路限制標籤來篩選網路設定檔和網路。
隨選路由網路	佈建會失敗，因為路由網路需要來自網路設定檔的資訊。	對於 NSX-V，需要選取 DLR (分散式邏輯路由器)。 對於 NSX-T 和 VMware Cloud on AWS，需要與私人網路和輸出網路類似的隨選設定。
使用現有網路或公用網路的 Wordpress 使用案例範例	依照現有網路或公用網路中所述進行佈建。	如需現有網路和公用網路行為，請參閱上述說明。 請參閱 <a href="#">WordPress 使用案例</a> 。
使用現有/公用網路和私人/輸出網路的 Wordpress 使用案例範例	佈建會失敗，因為網路需要來自網路設定檔的資訊。	如需私人網路和輸出網路，請參閱上述說明。 請參閱 <a href="#">WordPress 使用案例</a> 。
使用負載平衡器的 Wordpress 使用案例範例	佈建會失敗，因為負載平衡器需要來自網路設定檔的資訊。 如果存在現有的負載平衡器，則可以進行佈建。	根據網路設定檔組態建立新的負載平衡器。 您可以指定網路設定檔中已啟用的現有負載平衡器。 如果您申請現有負載平衡器，但網路設定檔中的所有負載平衡器均不滿足限制，佈建將會失敗。 請參閱 <a href="#">WordPress 使用案例</a> 。

## 在 vRealize Automation Cloud Assembly 中的網路設定檔和藍圖設計中使用負載平衡器設定

您可以根據 NSX-V 或 NSX-T 網路設定檔組態，在 vRealize Automation Cloud Assembly 藍圖中設定負載平衡器。

負載平衡器選項取決於藍圖的網路設定檔中的設定，以及 NSX-T 或 NSX-V 來源應用程式中的設定。

### NSX-T 網路和負載平衡器選項

NSX-T 對每個第 1 層邏輯路由器支援一個負載平衡器服務。負載平衡器選項取決於負載平衡器元件在藍圖中與其相關聯的網路。

#### ■ 隨選輸出網路

如果負載平衡器計算資源連結到隨選輸出網路，將為隨選網路的第 1 層路由器建立負載平衡器。

#### ■ 隨選私人網路



隨選網路 VIP 必須與來源外部網路的 VIP 相關聯。

如果負載平衡器計算資源連結到隨選私人網路，將建立新的第 1 層路由器，並將其連結到網路設定檔中指定的第 0 層路由器。負載平衡器隨後會連結到第 1 層路由器。如果 VIP 位於外部網路中，則會啟用第 1 層路由器 VIP 通告。

#### ■ 現有網路

如果負載平衡器連結到現有網路，則會使用現有網路的第 1 層路由器建立負載平衡器。僅當沒有任何負載平衡器服務連結到第 1 層路由器時，才會建立新的小型負載平衡器服務。如果負載平衡器服務已存在，則新的虛擬伺服器會連結到該服務。如果現有網路未連結到第 1 層路由器，則會建立新的第 1 層路由器並將其連結到網路設定檔中定義的第 0 層路由器，並且第 1 層路由器 VIP 通告未啟用。

#### ■ 具有新安全群組的隨選網路

由於機器已連結至現有網路並使用安全群組建立隔離，此選項類似於在現有網路中建立的負載平衡器。不同點為：若要啟用資料路徑，會將第 1 層上行連接埠 IP 新增至隔離安全群組。

如果不再使用，將銷毀為隨選網路建立的負載平衡器服務。銷毀負載平衡器時，會銷毀監視器、集區、應用程式設定檔和 VIP。不會銷毀為外部網路建立的負載平衡器服務。

您可以使用藍圖元件面板中可用的非雲端負載平衡器元件在藍圖中指定 NSX-T 負載平衡器設定。

若要進一步瞭解在 NSX-T 網路部署中使用負載平衡器，請參閱 VMware 部落格文章 [vRA Cloud Assembly 負載平衡器與 NSX-T 搭配使用深入探究](#)。

## NSX-V 網路和負載平衡器選項

您可以根據是使用新的網路子網路還是安全群組建立隔離的網路，將負載平衡器設定為雙肩或單肩。

您可以使用藍圖元件面板的 NSX 區段中可用的負載平衡器元件，在藍圖中指定 NSX-V 負載平衡器設定。

用於建立雙肩負載平衡器的工作流程是：

- 1 建立服務 Edge。
- 2 將服務 Edge 的上行介面連結到公用網路。
- 3 將下行介面連結到隔離 (輸出) 網路。
- 4 為負載平衡器配置網路設定檔靜態 IP 範圍內的靜態 IP 位址。
- 5 設定負載平衡器。
- 6 設定防火牆。
- 7 設定預設閘道。

用於建立旁接 (one-arm) 負載平衡器的工作流程是：

- 1 建立服務 Edge。
- 2 做為上行連結從保留區中選取的網路。
- 3 為負載平衡器配置網路設定檔靜態 IP 範圍內的靜態 IP 位址。
- 4 設定負載平衡器。

5 設定防火牆

6 設定預設閘道。

## 如何新增因應不同需求的 vRealize Automation Cloud Assembly 儲存區設定檔

vRealize Automation Cloud Assembly 儲存區設定檔說明了要部署的儲存區種類。

儲存區通常根據服務層級或成本、效能或目的 (例如備份) 等特性進行剖析。

選取**基礎結構 > 設定 > 儲存區設定檔**，然後按一下**新增儲存區設定檔**。

### 進一步瞭解 vRealize Automation Cloud Assembly 儲存區設定檔

雲端帳戶區域包含儲存區設定檔，雲端管理員可以使用儲存區設定檔為區域定義儲存區。

儲存區原則包括磁碟自訂，以及依功能標籤識別儲存區類型的方法。隨後標籤與佈建服務申請限制進行比對，以便在部署時建立所需的儲存區。

儲存區設定檔在特定於雲端的區域下進行組織整理。一個雲端帳戶可能擁有多個區域，每個區域中可能有多個儲存區設定檔。

可以進行獨立於廠商的放置。例如，視覺化三個不同的廠商帳戶，每個帳戶中有一個區域。每個區域包含具有 fast 功能標籤的儲存區設定檔。在佈建時，包含硬快速限制標籤的申請會尋找相符的快速功能，而不管哪一個廠商雲端在提供資源。然後，相符項會在建立已部署的儲存區項目期間套用相關聯儲存區設定檔中的設定。

---

**備註** 不同的雲端儲存區可能具有不同的效能特性，但進行標記的管理員仍將其視為快速供應項目。

---

新增至儲存區設定檔的功能標籤應無法識別實際資源目標，而是說明儲存區類型。如需有關實際資源的詳細資訊，請參閱[儲存資源](#)。

## 如何使用標籤來管理 vRealize Automation Cloud Assembly 資源和部署

標籤是 vRealize Automation Cloud Assembly 的一個重要元件，其透過比對功能和限制來驅動部署放置。您必須瞭解並有效實作標籤，才能最佳使用 vRealize Automation Cloud Assembly。

從根本上說，標籤是您新增到 vRealize Automation Cloud Assembly 項目的標籤。您可以建立適合您組織和實作的任何標籤。標籤的作用比標籤 (label) 更多，因為它們會控制 vRealize Automation Cloud Assembly 如何以及在何處使用資源和基礎結構來建置可部署的服務。標籤也支援在 Cloud Assembly 中進行管理。

### 標籤結構

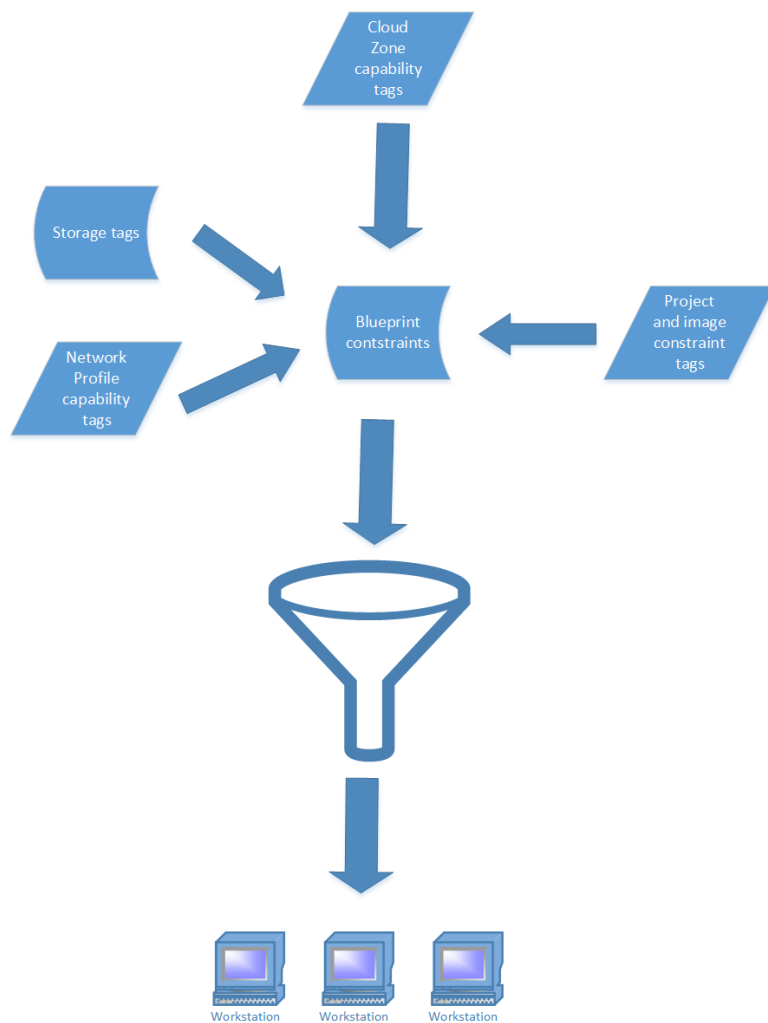
在結構上，標籤必須遵循 `name:value` 的配對慣例，但在其他方面其結構在很大程度上是自由格式。在 vRealize Automation Cloud Assembly 中，所有標籤都會以相同方式顯示，並且標籤功能由內容決定。

例如，基礎結構資源上的標籤主要用作功能標籤，因為 vRealize Automation Cloud Assembly 使用它們將資源與部署進行比對。其次，它們還會識別資源。

## 標籤功能

vRealize Automation Cloud Assembly 中標籤的主要功能是使用功能和限制設定部署。在雲端區域、網路和儲存區設定檔，以及個別基礎結構資源上放置的功能標籤定義了部署所需的功能。雲端管理員在專案上放置的限制標籤使其能夠對這些專案執行一種形式的管理。這些限制標籤將新增到藍圖內所示的其他限制中。

佈建期間，vRealize Automation Cloud Assembly 比對藍圖中的這些功能和限制 (也表示為標籤)，以定義部署組態。此以標籤為基礎的功能和限制功能做為 vRealize Automation Cloud Assembly 中部署組態的基礎。例如，您可以使用標籤使基礎結構僅在特定區域中的 PCI 資源上可用。



在次要層級，標籤還便於搜尋和識別儲存區及網路項目，以及其他基礎結構資源。

例如，假設您要設定雲端區域，並且有許多計算資源可用。如果您已為計算資源適當加上標籤，則可以使用 [雲端區域] 頁面中 [計算] 索引標籤上的搜尋功能，來篩選與該特定雲端區域相關聯的資源。

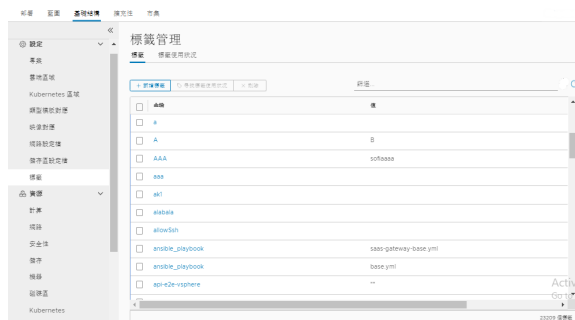
此外，[管理標籤] 頁面和資源組態頁面包含可讓您依標籤名稱找到項目的搜尋功能。為這些項目使用邏輯與易讀的標籤對促進此搜尋和識別功能很重要。

## 外部標籤

vRealize Automation Cloud Assembly 可能也包含外部標籤。將從與 vRealize Automation Cloud Assembly 執行個體相關聯的雲端帳戶自動匯入這些標籤。這些標籤可能會從 vSphere、AWS、Azure 或其他外部軟體產品匯入。匯入時，這些標籤的使用方法與使用者所建立的標籤相同。

## 管理標籤

您可以使用 vRealize Automation Cloud Assembly 中的 [管理標籤] 頁面來監控和管理標籤程式庫。您也可以在此頁面上建立標籤。此外，[管理標籤] 頁面是您可以檢視和識別外部標籤的唯一頁面。



## 標籤策略

為了盡可能減少混淆，在 vRealize Automation Cloud Assembly 中建立標籤之前，請設計適當的標籤策略和標籤慣例，以便建立和使用標籤的所有使用者都瞭解標籤的意思及其使用方式。請參閱[建立標記策略](#)。

## 建立標記策略

必須根據貴組織的 IT 結構及目標謹慎規劃和實作適當的標記策略，以最大化 Cloud Assembly 功能並盡量減少潛在混淆。

標記雖有數個一般用途，但您的標記策略必須根據您的組織需求、結構和目標而量身打造。

## 標記的最佳做法

有效標籤策略的一些一般特性：

- 設計和實作與業務結構相關的標記的連貫計劃，並將此計劃傳達給所有適用的使用者。計劃必須支援您的部署需求，使用清晰的人類可讀語言，並且所有適用的使用者均可理解。
- 對標籤使用簡單、明確和有意義的名稱。例如，儲存區和網路項目的標籤名稱應清晰且連貫，讓使用者可以輕易地瞭解他們為部署的資源選取或檢閱什麼樣的標籤指派。
- 雖然您可以使用不含值的名稱建立標籤，但最佳做法是為每個標籤名稱建立適用的值，這種做法更為合適，因為這可讓其他使用者清除地瞭解標籤使用狀況。

## 標記實作

繪製基本標記策略的主要考量事項。下列清單顯示了在對應策略時需考量的一般考量事項。請注意，這些考量事項具有代表性，而不是最終確定的。您可能需要與使用案例密切相關的其他考量事項。特定策略必須適用於特定使用案例。

- 您將部署至多少個不同的環境。通常，您會建立代表每個環境的標籤。
- 計算資源是如何建立結構並用來支援部署的。
- 您將部署至多少個不同的區域或位置。通常，您將在設定檔層級建立標籤來代表每個不同的區域或位置。
- 多少不同的儲存區選項可供用於部署，您想要如何說明它們的特性。這些選項應以標籤表示。
- 對您的網路選項進行分類，並建立標籤以容納所有適用的選項。
- 一般部署變數。例如，將部署至多少個不同的環境。通常，許多組織至少具有測試、開發和生產環境。您會想要建立及協調相符的藍圖限制標籤和雲端區域功能標籤，以便您可以輕鬆地將部署設定到一或多個此類環境中。
- 協調網路和儲存資源上的標籤，以便在使用這些標籤的網路和儲存區設定檔環境中具有邏輯意義。資源標籤可以對資源部署進行更為精細的控制。
- 將雲端區域和網路設定檔功能標籤以及其他功能標籤與藍圖限制標籤進行協調。一般而言，您的管理員將首先建立雲端區域和網路設定檔的功能標籤，然後其他使用者可以設計具有與這些功能標籤相符的限制的藍圖。

瞭解您組織的重要考量事項後，您可以規劃適當的標籤名稱，以邏輯方式處理下列考量事項。然後，建立策略的大綱，並使其可供具有建立或編輯標籤之權限的所有使用者使用。

您可以個別標記所有運算基礎結構資源以開始，這是一種有用的實作方法。如下所述，針對與特定資源相關聯的標籤名稱使用邏輯類別。對於執行個體，您可以將儲存資源標記為 tier1、tier2 等。此外，還可以根據其作業系統標記計算資源，例如 Windows、Linux 等。

標記資源後，您可以考慮採用最符合您需求的方法為雲端區域和儲存區及網路設定檔建立標籤。

## 在 vRealize Automation Cloud Assembly 中使用功能標籤

在 vRealize Automation Cloud Assembly 中，您可以使用功能標籤定義用於基礎結構元件部署的放置邏輯。功能標籤功能更強大且簡單，可對此類放置進行硬式編碼。

您可以在計算資源、雲端區域、映像和映像對應，以及網路和網路設定檔上建立功能標籤。用於建立這些資源的頁面包含用於建立功能標籤的選項。或者，您可以使用 vRealize Automation Cloud Assembly 中的 [管理標籤] 頁面建立功能標籤。雲端區域及網路設定檔上的功能標籤會影響這些區域或設定檔中的所有資源。儲存區或網路元件上的功能標籤只會影響套用了這些標籤的元件。

通常，功能標籤可定義諸如計算資源的位置、網路的介面卡類型或儲存資源的層層級等內容。此外，它們還可以定義環境位置或類型，以及任何其他業務考量事項。與整體標記策略一樣，您應該以邏輯方式組織整理功能標籤。

vRealize Automation Cloud Assembly 在部署期間將功能標籤與雲端區域和藍圖的限制進行比對。因此，建立和使用功能標籤時，您必須瞭解並計劃建立適當的藍圖限制，以便按預期進行比對。

例如，在 Wordpress 範例的 [新增雲端區域] 主題中，您為 OurCo-AWS-US-East 和 OurCo-AWS-US-West 區域分別建立開發和測試標籤。這表示 OurCo-AWS-US-East 區域是一個開發環境，OurCo-AWS-US-West 區域是一個測試環境。透過將這些功能標籤與適當的限制標籤配對，您可以將部署導向到所需的環境。

## 在 vRealize Automation Cloud Assembly 中使用限制標籤

您可以在 vRealize Automation Cloud Assembly 中向藍圖與其他各種元件新增限制標籤，以與資源、雲端區域和設定檔中定義的功能相符來產生適當的部署。

限制標籤在 vRealize Automation Cloud Assembly 中的兩個主要區域適用。第一個區域位於專案和映像中的組態端。第二個區域位於藍圖中的部署端。同時在兩個區域中套用的限制會合併至藍圖，以形成一組部署需求。

### 限制標籤在專案中的運作方式

設定 Cloud Assembly 時，雲端管理員可在專案和映像對應中套用限制標籤。如此一來，雲端管理員可以直接在專案層級套用管理限制。在此層級新增的所有限制會套用到為相應專案申請的每個藍圖。

如果專案中的標籤與藍圖中的標籤衝突，則會優先套用專案標籤，讓雲端管理員可以強制執行管理規則。

例如，如果雲端管理員在專案中建立 `location:london` 標籤，但開發人員在藍圖中放置

`location:boston` 標籤，則會優先套用前一個標籤，並且資源會部署到包含 `location:london` 標籤的基礎結構。

最多可在專案中套用三項限制。專案限制可為硬性或軟性。依預設，專案限制為硬性。使用硬限制可以嚴格強制執行部署限制。如果不符合一或多項硬限制，部署會失敗。軟限制提供了一種方法來表示將選取的喜好設定 (如果可用)，但不符合軟限制時，部署不會失敗。

### 限制標籤在藍圖中的運作方式

在藍圖中，您可以將限制標籤做為 YAML 代碼新增到資源，以與雲端管理員在資源、雲端區域和儲存區設定檔及網路設定檔中建立的相應功能標籤相符。此外，還有其他更複雜的實作限制標籤的選項。例如，您可以使用變數填入申請中的一或多個標籤。這可讓您在申請時指定一或多個標籤。

使用藍圖 YAML 程式碼中的 `tag` 標籤建立限制標籤。專案中的限制標籤將新增至藍圖中建立的限制標籤。

vRealize Automation Cloud Assembly 支援簡單的字串格式化，以便在 YAML 檔案中更輕鬆地使用限制：

```
[!tag_key[:tag_value][:hard|:soft]
```

依預設，vRealize Automation Cloud Assembly 會透過強制執行建立正向限制。儘管建議提供標籤值，但此值是選擇性的 (如應用程式的其餘部分一樣)。

以下具有 MySQL 的 WordPress 範例顯示 YAML 限制標籤，這些標籤指定了計算資源的位置資訊。

```
name: "wordpressWithMySQL"
components:
  mysql:
    type: "Compute"
    data:
```

```

    name: "mysql"
    # ... skipped lines ...
wordpress:
  type: "Compute"
  data:
    name: "wordpress"
    instanceType: small
    imageType: "ubuntu-server-1604"
    constraints:
      - tag: "!location:eu:hard"
      - tag: "location:us:soft"
      - tag: "!pci"
    # ... skipped lines ...

```

如需有關如何使用藍圖的詳細資訊，請參閱 [WordPress 使用案例：建立和擴充藍圖](#)。

## 硬限制和軟限制在專案和藍圖中的運作方式

專案和藍圖中的限制可為硬性或軟性。上述程式碼片段顯示了硬限制和軟限制的範例。依預設，所有限制均為硬性。使用硬限制可以嚴格強制執行部署限制。如果不符合一或多項硬限制，部署會失敗。軟限制用來表示將套用的喜好設定 (如果可用)，但不符合軟限制時，不會失敗。

如果特定資源類型存在一系列硬限制和軟限制，則軟限制也可以做為連接短路器。也就是說，如果多個資源滿足硬限制，會使用軟限制來選取在部署中使用的實際資源。

例如，最多可使用網路、儲存區和擴充性項目的任意組合對專案指定三項限制。此外，還可以選取每項限制為硬性還是軟性。假設您建立具有 `location:boston` 標籤的硬性儲存區限制。如果專案中的儲存區均不符合該限制，則任何相關的部署都會失敗。

**備註** 在專案和藍圖中，`failOnConstraintMergeConflict` 旗標可以修改限制的行為。將此旗標設定為 `true` 時，如果專案限制和藍圖限制之間存在衝突，申請將失敗。如果該旗標不存在或設定為 `false`，則專案限制優先於藍圖限制。

## 標準標籤

vRealize Automation Cloud Assembly 將標準標籤套用至部分部署，以支援對已部署的資源進行分析、監控和分組。

標準標籤在 vRealize Automation Cloud Assembly 中具有唯一性。與其他標籤不同，使用者在設定部署期間不使用標準標籤，並且不套用任何限制。這些標籤會在佈建期間自動套用至 AWS、Azure 和 vSphere 部署。這些標籤儲存為系統自訂內容，並且會在佈建後新增至部署。

下面顯示了標準標籤的清單。

表 4-2. 標準標籤

說明	標籤
組織	<code>org: orgID</code>
專案	<code>project: projectID</code>
申請者	<code>requester: username</code>



表 4-2. 標準標籤 (續)

說明	標籤
部署	deployment: deploymentID
藍圖參考 (如果適用)	blueprint: blueprintID
藍圖中的元件名稱	blueprintResourceName: CloudMachine_1
放置限制：在藍圖或申請參數中套用，或透過 IT 原則套用	constraints: key:value:soft
雲端帳戶	cloudAccount: accountID
區域或設定檔 (如果適用)	zone: zoneID, networkProfile: profileID, storageProfile: profileID

## vRealize Automation Cloud Assembly 如何處理標籤

在 vRealize Automation Cloud Assembly 中，標籤表示功能和限制，以決定在佈建期間如何以及在何處將資源配置給已佈建的部署。

vRealize Automation Cloud Assembly 使用解析標籤時的特定順序和階層來建立已佈建的部署。瞭解此程序的基本知識將有助於您有效地實作標籤，以建立可預測的部署。

以下清單概述了高層級作業和功能與限制標記的處理順序：

- 雲端區域會經過數個準則篩選，包括可用性和設定檔；此時符合區域所屬的區域設定檔中的標籤。
- 區域和計算功能標記用於依照硬限制篩選其餘雲端區域。
- 在篩選的區域以外，優先順序用來選取雲端區域。如果有多個雲端區域具有相同的優先順序，會使用雲端區域和計算功能的組合，依相符的軟限制將其排序。
- 選取雲端區域後，會透過比對一系列篩選器來選取主機，包括藍圖中所示的硬限制和軟限制。

## 如何設定簡單的標記結構

本主題說明了用於邏輯 vRealize Automation Cloud Assembly 標記策略的基本方法和選項。您可以使用這些範例做為實際部署的起點，或者可以設計更好地符合您需要的不同策略。

通常，雲端管理員是負責建立和維護標籤的主要人員。

本主題指的是 vRealize Automation Cloud Assembly 說明文件中其他位置所述的 WordPress 使用案例，以說明可以如何將標籤新增至部分關鍵項目。它還說明 WordPress 使用案例中出現的標記的範例可能的替代方案和延伸。

請參閱 [WordPress 使用案例](#)，以取得有關 WordPress 使用案例的詳細資訊。

WordPress 使用案例說明如何將標籤放置在雲端區域和儲存區，以及網路設定檔。這些設定檔就如同組織整理的資源套件。在設定檔上放置的標籤會套用至設定檔中的所有項目。您也可以建立標籤並將其放置在儲存區資源和個別網路項目上，以及計算資源上，但這些標籤只會套用到其放置所在的特定資源。設定標籤時，通常最好從標記計算資源開始，稍後將標籤新增至設定檔和雲端區域。此外，您可以使用這些標籤篩選雲端區域的計算資源的清單。



例如，如此使用案例中所示，您可以將標籤放置在儲存區設定檔上，也可以將標籤放置在個別儲存區原則、資料存放區和儲存區帳戶上。這些資源上的標籤可讓您更好地控制如何部署儲存資源。在準備部署時處理期間，這些標記會解析為在設定檔標記之後的下一個層級處理。

做為您可能如何設定一般客戶案例的範例，您可以將標記 `region: eastern` 放置在網路設定檔上。此標籤會套用到該設定檔中的所有資源。然後，您可以將標記 `networktype: pci` 放置在設定檔中的 PCI 網路資源上。具有東部和 PCI 限制的藍圖會建立將此 PCI 網路用於東部地區的部署。

## 程序

### 1 以邏輯和適當的方式標記計算基礎結構資源。

以邏輯方式標記計算資源特別重要，以便您可以使用 [建立雲端區域] 頁面的 [計算] 索引標籤上的搜尋功能找到它們。您可以使用此搜尋功能快速篩選與雲端區域相關聯的計算資源。如果您在設定檔層級標記儲存區和網路，可能不需要標記個別儲存區和網路資源。

- a 選取**資源 > 計算**，以檢視已為 vRealize Automation Cloud Assembly 執行個體匯入的計算資源。
- b 視情況選取每個計算資源，然後按一下**標籤**以將標籤新增至資源。如果適用，您可以將多個標籤新增至每個資源。
- c 視情況為儲存區和網路資源重複先前的步驟。

### 2 建立雲端區域和網路設定檔功能標記。

您可以為雲端區域和網路設定檔使用相同的標籤，也可以為每個項目建立唯一的標籤 (如果這對您的實作更有意義)。

在網路設定檔中，您可以將標記放置在整個設定檔上，以及設定檔中的子網路上。在設定檔層級套用的標籤將套用至所有元件，例如該設定檔中的子網路。子網路上的標籤僅套用至其放置所在的特定子網路。標籤處理期間，設定檔層級標籤優先於子網路層級標籤。

請參閱 [WordPress 使用案例：新增雲端區域](#) [WordPress 使用案例：新增網路設定檔](#)，以取得有關將標籤新增至雲端區域或網路設定檔的資訊。

在此範例中，我們建立三個簡單的標記，其會出現在 vRealize Automation Cloud Assembly 雲端區域和網路設定檔標記的整個使用案例說明文件中。這些標籤識別設定檔元件的環境。

- `zone:test`
- `zone:dev`
- `zone:prod`

### 3 為儲存區元件建立儲存區設定檔標籤。

通常，儲存區標籤識別儲存區項目的效能層級 (例如第 1 層或第 2 層)，或識別儲存區項目的特性 (例如 PCI)。

請參閱 [WordPress 使用案例：新增儲存區設定檔](#)，以取得有關將標籤新增至儲存區設定檔的資訊。

- `usage:general`
- `usage:fast`

## 結果

建立基本標記結構後，您可以開始使用它，以及視情況新增或編輯標籤，以縮小和擴充標記功能。

## 如何使用 vRealize Automation Cloud Assembly 中的資源

雲端管理員可以檢閱透過資料收集公開的 vRealize Automation Cloud Assembly 資源。雲端管理員可使用功能標籤標記資源，以影響 vRealize Automation Cloud Assembly 藍圖的部署位置。

### 計算資源

雲端管理員可以檢閱透過資料收集公開的計算資源。雲端管理員可以選擇將標籤直接套用到資源，以便在 vRealize Automation Cloud Assembly 佈建中標示功能用於比對。

### 網路資源

在 vRealize Automation Cloud Assembly 中，雲端管理員可以檢視和編輯已從專案的雲端帳戶和整合中進行資料收集的網路資源。

新增雲端帳戶後，資料收集會探索雲端帳戶的網路和安全性資訊，並讓該資訊可用於網路設定檔和其他選項中。

網路是可用網路網域或傳輸區域的 IP 特定元件。如果您是 Amazon Web Services 或 Microsoft Azure 使用者，請將網路視為子網路。

vRealize Automation Cloud Assembly **網路**頁面包含如下的資訊：

- 在您的雲端帳戶網路網域外部 (例如，在 vCenter、NSX-V 或 Amazon Web Services 中) 定義的網路和負載平衡器。
- 已由雲端管理員部署的網路和負載平衡器。
- 已由雲端管理員定義或修改的 IP 範圍和其他網路特性。
- 提供者特定的外部 IPAM 整合中特定位址空間的外部 IPAM 提供者 IP 範圍。

如需有關網路的詳細資訊，請參閱下列資訊、**網路**頁面上各種設定的路標說明和[進一步瞭解 vRealize Automation Cloud Assembly 中的網路設定檔](#)。

### 網路

您可以檢視和編輯網路及其特性，例如新增標籤或移除對公用 IP 存取的支援。您也可以可在可用的網路中定義新的 IP 範圍並管理現有的 IP 範圍。

對於現有網路，您可以透過選取網路的核取方塊，然後選取**管理 IP 範圍**或**標籤**來變更 IP 範圍和標籤設定。否則，您可以選取網路本身以編輯其資訊。

標籤提供了一種將相應網路 (或網路設定檔) 與藍圖中的網路元件相符的方法。網路標籤會套用到該網路的每個執行個體，無論網路可能存在於哪個網路設定檔。可以將網路設為任意數目網路設定檔的執行個體。無論網路設定檔在何處，且無論在何處使用網路，網路標籤均與該網路相關聯。藍圖已與一或多個網路設定檔相符後，藍圖中的其他元件會發生網路標籤比對。

## IP 範圍

選取 IP 範圍，以定義或變更組織中特定網路的起始和結束 IP 位址。

預設閘道不可包含在 IP 範圍內。子網路 IP 範圍不可包含子網路閘道值。

如果您針對特定的 IPAM 提供者使用外部 IPAM 整合，則可以新增 IPAM IP 範圍。此程序會在 [在 vRealize Automation Cloud Assembly 中將網路和網路設定檔設定為使用 IPAM 提供者值](#) 的整體外部 IPAM 整合工作流程的環境中進行說明。

## IP 位址

顯示組織所使用的已定義 IP 位址的狀態，例如可用或已配置。

## 負載平衡器

顯示組織中帳戶/區域雲端帳戶的可用負載平衡器的相關資訊。您可以針對每個可用的負載平衡器開啟和顯示已進行的設定。您也可以為負載平衡器新增和移除標籤。

## 安全資源

新增雲端帳戶後，資料收集會探索雲端帳戶的網路和安全性資訊，並讓該資訊可用於網路設定檔和其他選項中。

安全群組和防火牆規則支援網路隔離。

## 安全群組

您可以檢視已在 vRealize Automation Cloud Assembly 中建立的隨選安全群組以及在來源應用程式中建立的現有安全群組，例如，透過資料收集公開的 NSX-T 和 Amazon Web Services。

您可以檢視可用的安全群組，以及新增或移除所選安全群組的標籤。您也可以編輯已建立的隨選安全群組。雲端管理員可以開啟和編輯安全群組。藍圖作者可將一或多個安全群組指派給機器 NIC，以控制藍圖部署的網路規則與其他安全性層面。

在安全群組元件的藍圖代碼中使用防火牆規則時，

來自基礎雲端帳戶端點的現有安全群組 (例如 NSX-V、NSX-T 或 Amazon Web Services 應用程式) 由 vRealize Automation Cloud Assembly 收集資料。顯示現有安全群組並將其**在來源資料行中分類為 Discovered**。顯示您在 vRealize Automation Cloud Assembly 的藍圖或網路設定檔中建立的隨選安全群組，並將其**在來源資料行中分類為 Managed by Cloud Assembly**。只能看到可在組織中使用的安全群組。

如果您直接在來源應用程式中 (例如，在來源 NSX 應用程式中，而不是在 vRealize Automation Cloud Assembly 中) 編輯現有安全群組，則在從 vRealize Automation Cloud Assembly 中對相關聯雲端帳戶或整合點執行資料收集之前，這些更新不會顯示在 vRealize Automation Cloud Assembly 中。

雲端管理員可將一或多個標籤指派給現有安全群組，以允許在藍圖中使用該標籤。安全群組必須至少有一個標籤，否則無法在藍圖中使用。藍圖作者可以在藍圖中使用 `Cloud.SecurityGroup` 元件，透過使用標籤限制來配置現有的安全群組。在藍圖中，必須將 `Cloud.SecurityGroup` 安全群組元件套用於機器 NIC，以控制藍圖部署中的網路規則和其他安全性層面。

如需管理員如何將標籤新增到現有安全群組，或如何從來源應用程式中的安全群組收集標籤資料的相關資訊，請參閱[進一步瞭解 vRealize Automation Cloud Assembly 中的網路設定檔](#)。

如需藍圖作者如何在藍圖安全性元件中使用標籤的相關範例，請參閱 [vRealize Automation Cloud Assembly 藍圖中的網路、安全性和負載平衡器範例](#)。

若要取得藍圖程式碼範本，請參閱 [vRealize Automation Cloud Assembly 藍圖中的網路、安全性和負載平衡器範例](#)中的已將限制標籤套用至機器 NIC 的現有安全群組範例。

如需有關在網路設定檔中使用安全群組的詳細資訊，請參閱[進一步瞭解 vRealize Automation Cloud Assembly 中的網路設定檔](#)。

## 在網路設定檔和藍圖中定義安全群組的方式

您可以透過下列任一方式來利用安全群組功能：

- 網路設定檔中指定的現有安全群組

您可以將現有安全群組新增到網路設定檔。當藍圖使用該網路設定檔時，其機器會做為安全群組的成員分組在一起。此方法不需要您將安全群組元件新增到藍圖。

- 與藍圖中的機器元件相關聯的安全群組元件

您可以將安全群組元件拖放至藍圖，並透過在藍圖中的現有安全群組元件與已收集資料之資源中的現有安全群組上使用限制標籤，將安全群組元件繫結至機器 NIC。

- NSX-T 應用程式中指定的 NSX-T 標籤

- 您可以使用在藍圖中指定做為網路元件限制的 NSX-T 標籤，其中網路元件將連線至藍圖中的機器 NIC。透過 NSX-T 標籤，您可以使用預先定義的 NSX-T 標籤 (該標籤是從 NSX-T 來源端點進行資料收集的) 來動態分組機器。在 NSX-T 中建立 NSX-T 標籤時，請使用邏輯連接埠。

## 儲存資源

雲端管理員可以使用儲存資源及其功能，這些內容可透過相關聯雲端帳戶中進行的 vRealize Automation Cloud Assembly 資料收集發現。

儲存資源功能透過通常源自於來源雲端帳戶的標籤進行公開。雲端管理員可以選擇使用 vRealize Automation Cloud Assembly 將其他標籤直接套用至儲存資源。其他標籤可能會在佈建時標示特定的功能以用於比對。

儲存資源的功能在定義 vRealize Automation Cloud Assembly 儲存區設定檔的過程中變為可見。請參閱[進一步瞭解 vRealize Automation Cloud Assembly 儲存區設定檔](#)。

## 機器資源

在 vRealize Automation 中，所有使用者都可以檢閱透過資料收集公開的機器資源。

專案中的所有機器將會顯示在機器清單中。

與專案中的雲端帳戶相關聯的未受管理的機器會像受管理機器一樣顯示在此清單中。[來源] 資料行指示了機器狀態。

- 已發現 - 尚未上線的機器。

- 已部署 - 已從 vRealize Automation 上線或佈建的機器。

您可以使用工作負載上線計劃將未受管理的機器引入 vRealize Automation 管理。

如需使用上線計劃將未受管理的機器引入 vRealize Automation 管理的相關資訊，請參閱 [vRealize Automation Cloud Assembly 中的上線計劃是什麼](#)。

## 磁碟區資源

在 vRealize Automation Cloud Assembly 中，所有使用者均可檢閱磁碟區資源。

vRealize Automation Cloud Assembly 會顯示源自兩個來源的磁碟區或邏輯磁碟機：

- 透過從來源雲端帳戶收集資料探索到的磁碟區
- 與 vRealize Automation Cloud Assembly 佈建的工作負載相關聯的磁碟區

您可以根據磁碟區或邏輯磁碟機來檢閱容量和功能。清單也會公開源自來源雲端帳戶或 vRealize Automation Cloud Assembly 本身中新增的功能標籤。

## 進一步瞭解 vRealize Automation Cloud Assembly 中的資源

vRealize Automation Cloud Assembly 可以公開有關已進行資料收集的資源的其他資訊，例如成本。

### 如何在 vRealize Automation vRealize Automation Cloud Assembly 中進行資料收集

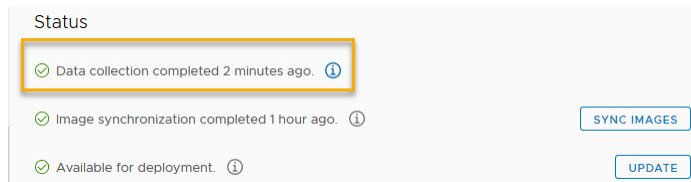
初始資料收集後，資源資料收集會每 10 分鐘自動執行一次。不可設定資料收集時間間隔，且無法手動起始資料收集。

您可以在現有雲端帳戶頁面的 [狀態] 區段中，探索該帳戶的資源資料收集和映像同步的相關資訊。若要執行此操作，請選取 **基礎結構** > **連線** > **雲端帳戶**，然後在您選擇的現有雲端帳戶上按一下 **開啟**。

您可以開啟現有的雲端帳戶，並在其頁面的 **狀態** 區段中查看其相關聯的端點版本。如果相關聯的端點已升級，則會在資料收集期間探索到新的端點版本，並反映在雲端帳戶頁面上的 **狀態** 區段中。

### 資源資料收集

資料收集會每 10 分鐘執行一次。每個雲端帳戶顯示最後一次完成其資料收集的時間。



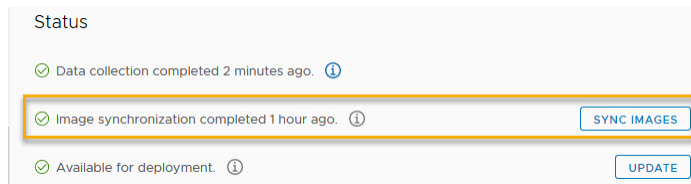


## 映像資料收集

映像同步會每 24 小時執行一次。您可以針對某些雲端帳戶類型起始映像同步。若要起始映像同步，請開啟雲端帳戶 (**基礎結構 > 雲端帳戶**，選取並開啟現有的雲端帳戶)，然後按一下 **同步映像** 按鈕。NSXCloud 帳戶沒有映像同步選項。

**備註** 映像在其內部分類為公用或私人。公用映像是共用的，並非專屬於特定的雲端訂閱或組織。私人映像不是共用的，且專屬於某個特定訂閱。公用和私人映像會每 24 小時自動同步一次。[雲端帳戶] 頁面上的選項可讓您觸發私人映像的同步。

[雲端帳戶] 頁面將顯示上次完成映像同步的時間。



為了促進部署中的容錯和高可用性，每個 NSX-T 資料中心端點均代表由三個 NSX Manager 組成的叢集。如需相關資訊，請參閱在 [vRealize Automation Cloud Assembly 中建立 NSX-T 雲端帳戶](#)。

## 雲端帳戶和上線計劃

建立雲端帳戶時，將為與此雲端帳戶相關聯的所有機器收集資料，然後在 **基礎結構 > 資源 > 機器** 頁面中顯示這些機器。如果雲端帳戶包含在 vRealize Automation Cloud Assembly 外部部署的機器，您可以使用上線計劃以允許 vRealize Automation Cloud Assembly 管理機器部署。

如需新增雲端帳戶的相關資訊，請參閱 [將雲端帳戶新增至 vRealize Automation Cloud Assembly](#)。

如需將未受管理的機器上線的相關資訊，請參閱 [vRealize Automation Cloud Assembly 中的上線計劃是什麼](#)。

## 什麼是 vRealize Automation Cloud Assembly 部署成本

專案管理員和雲端管理員通常負責管理成本。vRealize Automation Cloud Assembly 成本管理可協助您瞭解個別部署的貨幣影響，以便協助您管理資源。

您必須在 vRealize Operations 中設定並啟用成本以便與 vRealize Automation 搭配使用，才能檢視成本計算。設定 vRealize Operations 與 vRealize Automation 時，請確保兩個應用程式會設定為相同的時區。若要在 vRealize Operations 中設定時區，請啟用 SSH 並登入每個 vRealize Operations 節點，編輯 `$ALIVE_Base/user/conf/analytics/advanced.properties` 檔案，然後新增

```
timeZoneUseInMeteringCalculation = <time zone>。
```

一段時間內的部署成本在部署卡上顯示為月初至今成本，每月月初會重設為零。部署詳細資料中提供了元件成本明細。在部署層級提供此資訊可告知雲端管理員相關資訊，但它還有助於成員瞭解其工作可能對預算和長期開發的影響。

由於類似的業務原因，您可能需要瞭解整個專案的總成本。專案層級的成本反映了專案團隊所有部署的全部成本。若要檢視完整專案成本，請選取**基礎結構 > 專案 > 選取專案 > 成本**。結果會依部署顯示彙總成本和成本明細。

**備註** 專案成本僅包含私有雲工作負載的成本。如果專案包含屬於公有雲的部署，則這些部署的成本不會包括在專案成本內。

### 如何計算成本

您在部署層級所看到的計算和儲存資源的初始成本是基於業界標準基準費率，並且根據時間計算。將成本費率套用到主機，服務會計算 CPU 和記憶體費率。伺服器每 24 小時重新計算一次成本。

您也可以隨時在 vROPs 端點頁面的**基礎結構 > 整合 > vROPs 端點 > 中**手動重新整理成本伺服器。在 vCenter Server 區段中，按一下**同步**。使用**同步**選項手動重新整理成本伺服器時，會重新計算組織中所有專案的成本。根據組織有多少專案，此程序可能需要耗用大量資源，而且需要花費一定的時間。

如需支援資源的清單，請參閱 [vRealize Automation Cloud Assembly 中的計費元件類型清單](#)。

### 如何自訂計算的費率

使用一段時間內的基準值後，雲端管理員可能會發現您的實際成本已進行折算。您可以調整用於計算成本的費率，以便更好地反映您的業務實務。

如果進行調整，則計算變更將在下次服務執行計算時反映出來。伺服器每 24 小時重新計算這些值。

### vRealize Automation Cloud Assembly 中的計費元件類型清單





vRealize Automation Cloud Assembly 提供下列藍圖元件類型的基準成本資訊。

表 4-3. 計費元件類型

藍圖元件類型	服務名稱/物件類型	藍圖資源類型	註解
非雲端	機器	Cloud.Machine	如果為無關機器設定 vSphere，您可以檢視部署成本。
	磁碟	Cloud.Volume	如果將無關磁碟連結至設定了 vSphere 的虛擬機器，您可以檢視部署成本。
vSphere	vSphere 機器	Cloud.vSphere.Machine	使用特定於雲端的藍圖的部署。
	vSphere 磁碟	Cloud.vSphere.Disk	已使用連結至虛擬機器的雲端特定藍圖進行部署。

### 如何估計部署成本

部署目錄項目之前，您可以使用先期成本進行部署成本估計。

Daily Cost Estimate		×
	vSphere machine with disk	\$2.14
	Cloud_vSphere_Machine_1	\$2.11
	Compute	\$1.97
	Storage	\$0.03
	Additional charges	\$0.11
	Cloud_vSphere_Disk_1	\$0.03
	Storage	\$0.03
	Cloud_vSphere_Network_1	
	<i>Cost for this resource is not supported.</i>	
		<b>CLOSE</b>

部署的先期成本是在部署指定目錄項目之前，根據資源配置為該項目計算的每日成本估計。部署目錄項目之後，您可以在**部署和基礎結構 > 專案**索引標籤上，以先期成本的彙總形式來檢視月初至今的成本。對私有雲資源 (例如 vSphere 機器和 vSphere 磁碟、Cloud Assembly 目錄項目，以及已針對私有雲設定 vCenter 的非雲端項目) 支援先期成本。

**備註** 公有雲資源、非 vSphere 機器或磁碟私有雲資源不支援先期成本。

#### 必要條件

若要在 vRealize Automation Cloud Assembly 中檢視先期成本，您必須設定 vRealize Operations 整合端點為其啟用成本計算並預設貨幣。

#### 程序

- 1 從目錄中選取目錄項目，然後按一下**申請**。

Daily Cost Estimate	0.00
<b>CALCULATE</b>	DETAILS

- 2 輸入目錄項目申請的詳細資料，然後按一下**計算**。

Daily Cost Estimate	\$2.14
UPDATE	<b>DETAILS</b>

- 3 (選用) 按一下**詳細資料**，以在 [每日成本估計] 視窗中檢視成本明細。



## 後續步驟

如果每日成本估計可接受，請按一下**提交**以繼續部署申請。

## 如何確定部署看起來價格昂貴的原因

管理成本時，專案管理員和雲端管理員可能會發現部署價格昂貴。

部署的成本依據其資源進行計算。有時，單一部署中的資源組合會增加成本，從而使部署對於您的業務需求來說價格昂貴。因此，如果您的專案包含一或多個價格昂貴的部署，彙總專案成本也會增加。

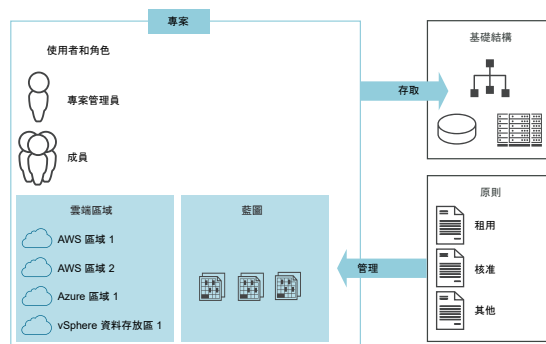
確定您的部署是否昂貴，這是您自己的業務需求所特有的。您可以透過開啟部署的詳細資料並按一下**成本**，以檢視部署的成本明細。若要修改部署成本，您必須編輯其藍圖的資源組成並重新部署。

# 新增和管理 vRealize Automation Cloud Assembly 專案

# 5

專案控制誰有權存取 vRealize Automation Cloud Assembly 藍圖和藍圖的部署位置。您可以使用專案來組織整理和管理您的使用者可以執行什麼，以及他們可以在雲端基礎結構中的哪些雲端區域中部署藍圖。

雲端管理員設定專案，他們可以向其中新增使用者和雲端區域。建立和部署藍圖的任何人均必須是至少一個專案的成員。



本章節討論下列主題：

- [如何為我的 vRealize Automation Cloud Assembly 開發團隊新增專案](#)
- [進一步瞭解 vRealize Automation Cloud Assembly 專案](#)

## 如何為我的 vRealize Automation Cloud Assembly 開發團隊新增專案

您可以建立一個專案，向其新增成員及雲端區域，以便專案成員可將其藍圖部署到相關聯的區域。做為 vRealize Automation Cloud Assembly 管理員，您可以為開發團隊建立一個專案。然後，您可以指派專案管理員，也可以專案管理員身分進行操作。

建立藍圖時，先選取與其相關聯的專案。此專案必須存在，您才能建立藍圖。

確保您的專案支援開發團隊的業務需求。

- 專案是否會提供支援團隊目標的資源。如需有關基礎結構資源和專案如何支援藍圖的範例，請參閱 [WordPress 使用案例](#)。

此程序基於建立僅包含基本組態的初始專案。當您的開發團隊建立和部署其藍圖時，您可以修改專案。您可以新增限制、自訂內容和其他選項，以提升部署效率。請參閱[進一步瞭解 vRealize Automation Cloud Assembly 專案](#)中的文章。

### 必要條件

- 確認您已設定雲端區域。請參閱第 4 章 建置 vRealize Automation Cloud Assembly 資源基礎結構。
- 針對包含做為此專案的雲端區域的區域，確認您為其設定對應和設定檔。請參閱第 4 章 建置 vRealize Automation Cloud Assembly 資源基礎結構。
- 確認您擁有執行此工作的必要權限。請參閱什麼是 vRealize Automation Cloud Assembly 使用者角色。
- 確定將指定為專案管理員的使用者。若要瞭解專案管理員可以在 vRealize Automation Cloud Assembly 中執行的動作，請參閱 什麼是 vRealize Automation Cloud Assembly 使用者角色。
- 如果您要新增 Active Directory 群組至專案，請確認已為您的組織設定了 Active Directory 群組。請參閱《管理 vRealize Automation》中的〈在 vRealize Automation 中編輯群組角色指派〉。如果這些群組未同步，則當您嘗試將其新增至專案時將無法使用。

### 程序

- 1 選取**基礎結構 > 設定 > 專案**，然後按一下**新增專案**。
- 2 輸入專案名稱。
- 3 按一下**使用者索引**標籤。
  - a 若要使專案成員進行的部署僅可供擁有者存取，請關閉**部署共用**。
  - b 新增具有已指派角色的使用者。
- 4 按一下**佈建**索引標籤，然後新增一或多個雲端區域。
 

雲端區域必須包含支援使用者所部署藍圖的資源。
- 5 按一下**建立**。
- 6 若要使用專案雲端區域測試您的專案，請按一下 [專案] 頁面上的**測試組態**。
 

模擬會針對專案雲端區域資源執行標準化假設部署測試。如果失敗，您可以檢閱詳細資料並更正資源組態。

### 後續步驟

開始使用藍圖。請參閱第 6 章 設計 vRealize Automation Cloud Assembly 部署。

## 進一步瞭解 vRealize Automation Cloud Assembly 專案

專案是藍圖和資源之間的連接器。您對它們的運作方式以及如何讓它們為您工作瞭解得越多，您的 vRealize Automation Cloud Assembly 開發和部署程序就會越有效。

### 使用 vRealize Automation Cloud Assembly 專案標籤和自訂內容

做為管理員，當專案的需求與 vRealize Automation Cloud Assembly 藍圖不同時，您可以新增專案層級的管理限制或自訂內容。除了限制標籤，您還可以新增在佈建程序期間新增到已部署資源的資源標籤，以便您可以管理這些資源。

## 什麼是專案資源標籤

專案資源標籤用作標準化識別標籤，可用於管理已部署的資源並確保符合性。

在專案中定義的資源標籤會新增至部署做為該專案一部分的所有元件資源。然後，您可以使用標準標記功能透過其他應用程式管理資源。

例如，做為雲端管理員，您要使用 CloudHealth 之類的應用程式來管理成本。您可以新增 `costCenter:eu-cc-1234` 標籤至專用於開發歐盟人力資源工具的專案。當專案團隊從此專案部署時，則會將標籤新增至已部署的資源。然後，設定成本計算工具以識別並管理包含此標籤的資源。其他成本中心的其他專案將具有與金鑰相符的備用值。

## 什麼是專案限制標籤

專案限制做為管理定義執行。專案限制是一個 `key:value` 標籤，用於定義部署請求在專案雲端區域中使用或避免的資源。

部署程序會尋找與專案限制相符的網路和儲存區的標籤，然後根據比對標籤進行部署。

擴充性限制用來指定要用於擴充性工作流程的 vRealize Orchestrator 整合執行個體。

設定專案限制時，可考慮下列格式。

- **key:value** 和 **key:value:hard**。如果必須在有相符功能標籤的資源上佈建藍圖，請使用此標籤，任一格式皆可。如果找不到相符標籤，部署程序會失敗。例如，由專案成員部署的藍圖必須在符合 PCI 標準的網路上佈建。您將使用 `security:pci`。如果在專案雲端區域中找不到任何網路，部署會失敗，以確保沒有任何不安全的部署。
- **key:value:soft**。如果您偏好相符資源但想要部署程序繼續而不會失敗，並且可以接受標籤不相符的資源，請使用此標籤。例如，您希望專案成員將其藍圖部署到成本更低的儲存區，但不想儲存區可用性干擾專案成員的部署能力。您將使用 `tier:silver:soft`。如果專案雲端區域中不存在加上 `tier:silver` 標籤的儲存區，藍圖仍會部署到其他儲存資源。
- **!key:value**。如果您想要避免部署到具有相符標籤的資源，請使用此標籤，指定為硬性或軟性皆可。

重要的是，專案限制標籤的優先順序高於藍圖限制標籤，並且專案限制標籤會在部署時覆寫藍圖限制標籤。如果您的藍圖不允許出現這種情況，您可以在藍圖中使用 `failOnConstraintMergeConflict:true`。例如，如果您的專案具有 `loc:london` 網路限制，而藍圖具有 `loc:mumbai` 網路限制，您想要部署失敗並顯示限制衝突訊息，而不希望專案位置優先，則新增類似以下範例的內容。

```
constraints:
  - tag: 'loc:mumbai'
failOnConstraintMergeConflict:true
```

## 如何使用專案自訂內容

您可以將專案自訂內容用於報告，以觸發和填入擴充性動作與工作流程以及覆寫藍圖層級內容。

透過將自訂內容新增至部署，您可以使用使用者介面中的值，或使用 API 擷取內容來產生報告。

擴充性也可以將自訂內容用於擴充性訂閱。

您可能想要針對某個專案變更藍圖的特定內容值。您可以提供替代名稱和值做為自訂內容。

## vRealize Automation Cloud Assembly 專案在部署時的運作方式

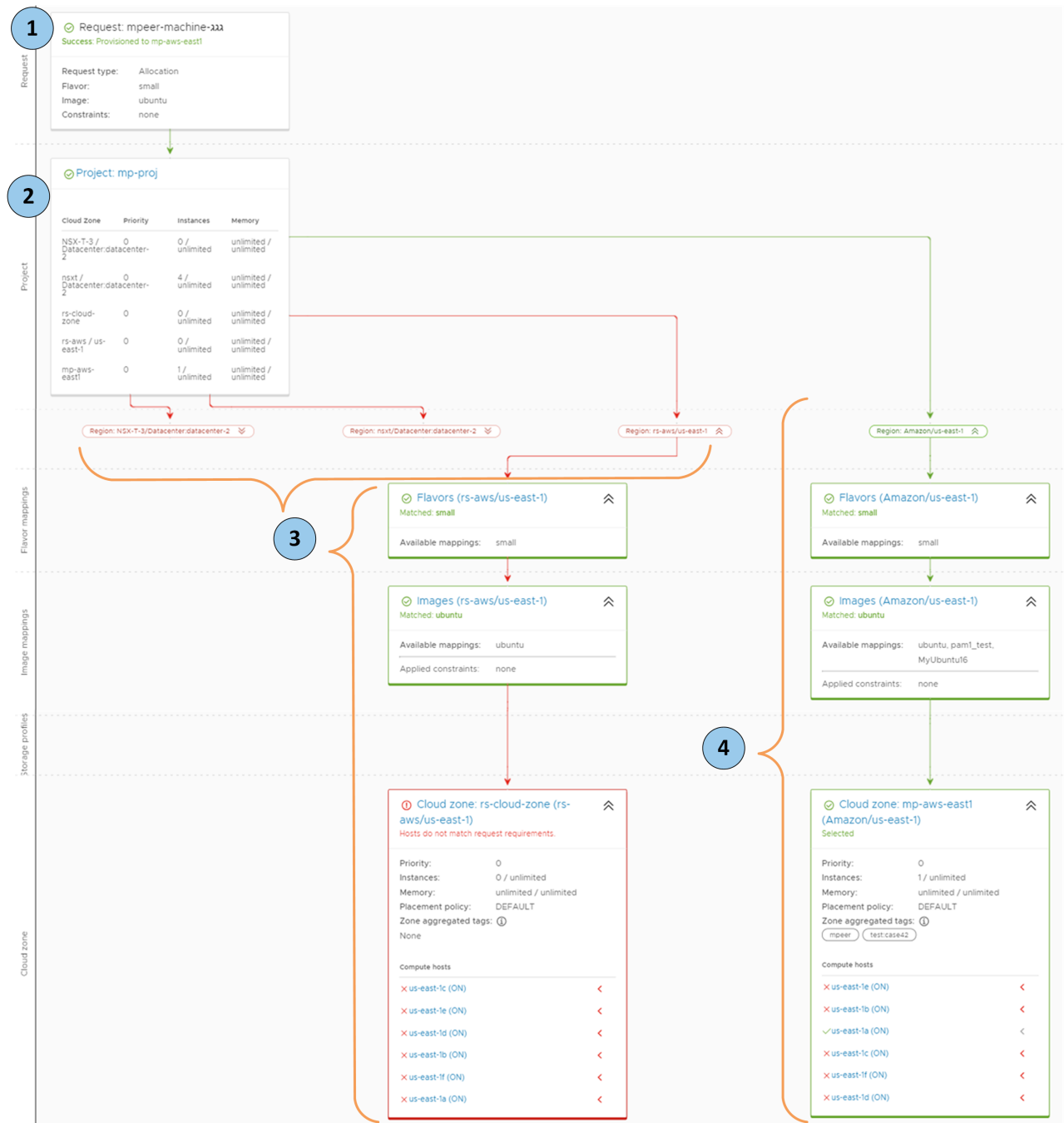
專案控制使用者對雲端區域的存取權和已佈建資源的使用者擁有權。無論您是雲端管理員還是藍圖開發人員，均必須瞭解在部署期間專案的運作方式，以便管理您的部署和疑難排解任何問題。

做為設定不同團隊的專案的雲端管理員，您必須瞭解專案如何判斷部署藍圖元件的位置。這可協助您建立支援藍圖開發人員的專案，以及疑難排解失敗的部署。

您建立藍圖時，首先將其與專案建立關聯。在部署時，會針對專案雲端區域評估藍圖需求，以找到最佳部署位置。

以下工作流程說明了相關程序。

- 1 您可以提交藍圖部署申請。
- 2 此專案會評估藍圖和專案需求，例如類型模板、映像和限制標籤。將需求與專案雲端區域進行比較，以找到支援這些需求的區域。
- 3 這些區域沒有支援此申請的資源。
- 4 此雲端區域支援申請需求，並且藍圖會部署到此雲端區域帳戶區域。



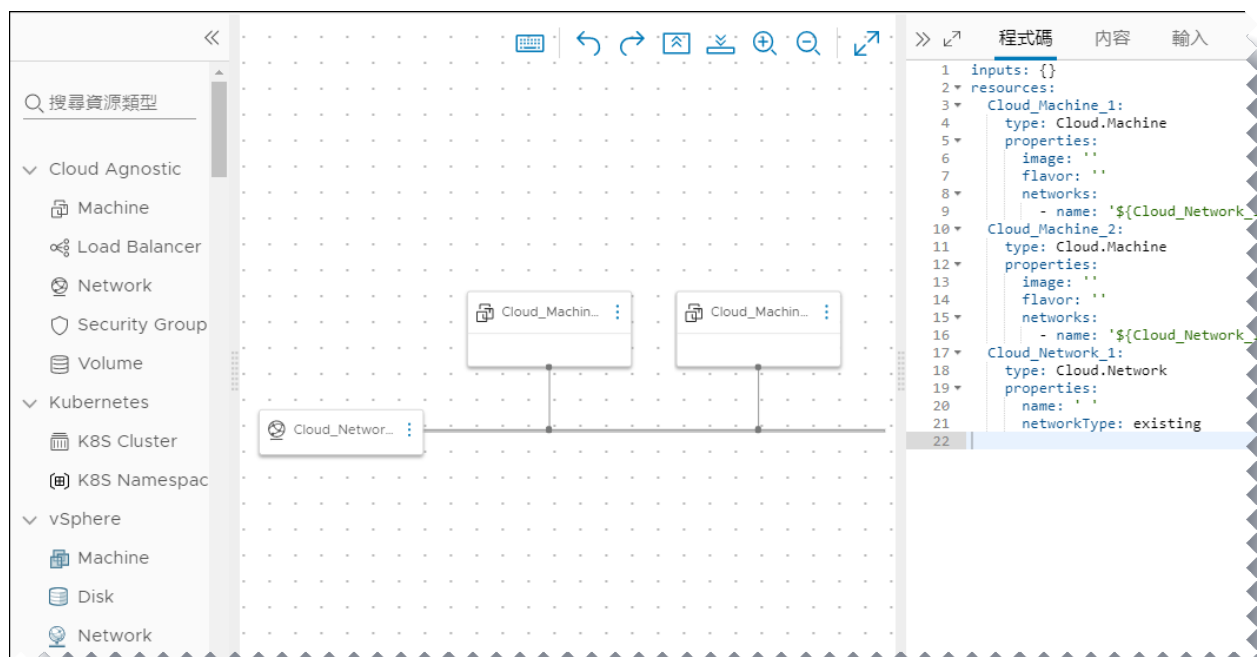
# 設計 vRealize Automation Cloud Assembly 部署

## 6

部署從藍圖開始，這些規格定義了透過 vRealize Automation Cloud Assembly 在雲端資源上建立的機器、應用程式和服務。

做為藍圖開發人員，您可以設計目標為特定雲端廠商的藍圖，或將藍圖設定為非雲端。指派給專案的雲端區域決定了您可以採取的方法。請洽詢雲端管理員，以確保您瞭解構成雲端區域的資源類型。

請注意，vRealize Automation Cloud Assembly 藍圖建立是一個基礎結構即代碼的程序。您可以在設計畫布中新增並連線元件以開始。然後，使用畫布右側的代碼編輯器填寫詳細資料。代碼編輯器可讓您直接輸入代碼或將內容值輸入表單中。



本章節討論下列主題：

- 在建立藍圖之前
- 建立藍圖的方式
- 如何從頭建立簡單 vRealize Automation Cloud Assembly 藍圖
- 如何強化簡單 vRealize Automation Cloud Assembly 藍圖

- [如何儲存不同版本的 vRealize Automation Cloud Assembly 藍圖](#)
- [如何使用 vRealize Automation Cloud Assembly 自訂已部署資源的名稱](#)
- [什麼是 vRealize Automation 資源內容](#)
- [可使用的一些藍圖代碼範例](#)
- [如何使用 vRealize Automation Cloud Assembly 市集](#)
- [如何使用擴充性延長和自動化應用程式生命週期](#)

## 在建立藍圖之前

您可以隨時建立 vRealize Automation Cloud Assembly 藍圖，但必須先定義雲端資源基礎結構才能部署藍圖。

- [第 4 章 建置 vRealize Automation Cloud Assembly 資源基礎結構](#)

此外，您必須建立包含這些基礎結構資源做為雲端區域的 vRealize Automation Cloud Assembly 專案。

- [使用 vRealize Automation Cloud Assembly 專案標籤和自訂內容](#)

## 建立藍圖的方式

vRealize Automation Cloud Assembly 建立藍圖並將其儲存為代碼，這可讓您輕鬆地設計和重複使用藍圖。

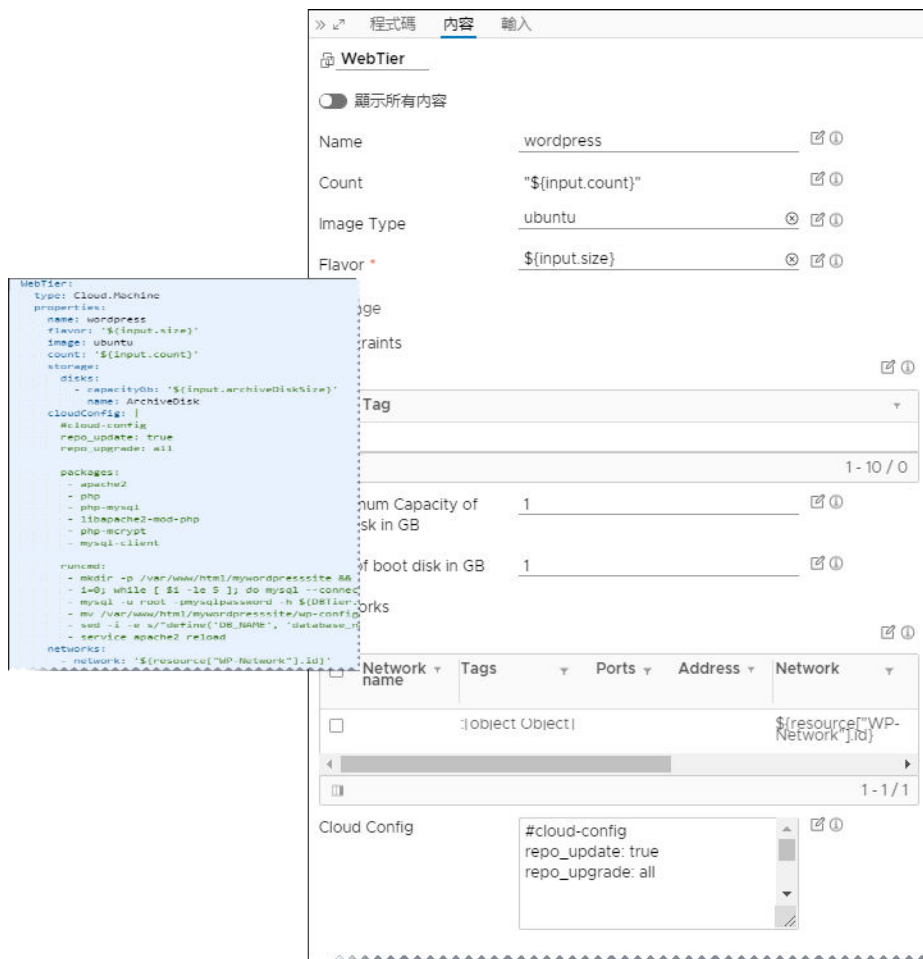
您可以從空白畫布建置藍圖，也可以利用現有代碼。

### vRealize Automation Cloud Assembly 藍圖設計頁面

若要從頭建立藍圖，請移至[藍圖](#)，然後按一下**新增**。將元件拖曳至畫布，連線這些元件，然後在代碼編輯器中完成元件設定。

透過代碼編輯器，您可以直接輸入、剪下、複製和貼上代碼。如果不願意編輯代碼，您可以在設計畫布中選取資源，按一下代碼編輯器的**內容**索引標籤，然後在此輸入值。您輸入的內容值會出現在代碼中，如同您直接將其輸入一樣。





請注意，您可以從一個藍圖複製代碼並將其貼到另一個藍圖。

## 藍圖複製

若要複製藍圖，請移至藍圖，選取來源，然後按一下複製。您可以複製藍圖以根據來源建立一個複本，然後將複製品指派給新專案或將其做為新應用程式的起始程式碼。

## 上傳和下載

vRealize Automation Cloud Assembly 市集提供完成的藍圖，以快速啟動您的工作。請參閱[如何使用 vRealize Automation Cloud Assembly 市集](#)。

此外，您也可以透過對您的站台有效的任何方式來上傳、下載和共用藍圖 YAML 代碼。您甚至可以使用外部編輯器和開發環境修改藍圖代碼。

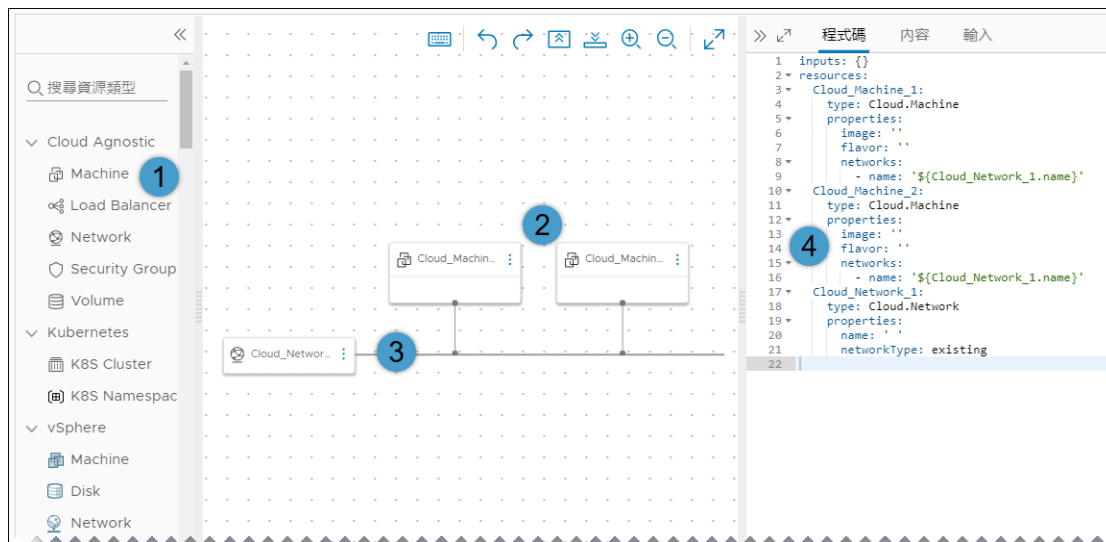
**備註** 驗證共用藍圖代碼的一般方式是，在藍圖設計頁面上的 vRealize Automation Cloud Assembly 代碼編輯器中對其進行檢查。

藍圖 249 個項目						
+ 新增 + 上傳 同步存儲庫 複製 部署 下載 刪除						
Q 篩選...						
<input type="checkbox"/>	名稱	源組權控制	專案	上次更新時間	更新者	發行的版本
<input checked="" type="checkbox"/>	vSphere-With-Disk-Attached		test-AD-project	2020年1月21日 下午12:48:57	sestervii@vmware.com	0 共 0
<input type="checkbox"/>	DB		0709-AWS-w2聯家表ホA中E6體傳0道Ü8äüH	2020年1月21日 上午11:34:36	pmartini@vmware.com	0 共 0
<input type="checkbox"/>	WordPress-BP		0709-AWS-w2聯家表ホA中E6體傳0道Ü8äüH	2020年1月20日 下午3:25:36	cani@vmware.com	0 共 0
<input type="checkbox"/>	git BPP		Azure Project1聯家表ホA中E6體傳0道Ü8äüH	2020年1月20日 下午3:18:40	cani@vmware.com	0 共 0
<input type="checkbox"/>	mp-nxsv		0717VSPHERE_PROJECTVSPHERE聯家表ホA中E6體傳0道Ü8äüH	2020年1月20日 上午10:03:51	pmartini@vmware.com	0 共 0
<input type="checkbox"/>	> WP - PORT		wordpress project	2020年1月19日 下午7:12:37	pmartini@vmware.com	2 共 2

## 如何從頭建立簡單 vRealize Automation Cloud Assembly 藍圖

您可以使用設計頁面為要佈建的機器或應用程式建立 vRealize Automation Cloud Assembly 藍圖規格。

- 1 找到元件。
- 2 將元件拖曳至畫布。
- 3 連線元件。
- 4 透過編輯藍圖代碼來設定元件。



從設計頁面中，您也可以變更藍圖名稱、版本或還原為版本、複製或部署藍圖。

## 如何選取 vRealize Automation Cloud Assembly 元件並將其新增至藍圖

vRealize Automation Cloud Assembly 元件是藍圖建置區塊。在設計頁面中，您可以使用非雲端元件或特定於雲端廠商的元件。

元件會顯示在設計頁面的左側供您選取。

## 非雲端元件

您可以將與雲端無關的元件部署到任何雲端廠商。在佈建時，部署會使用相符的雲端特定資源。例如，如果您希望藍圖同時部署至 AWS 雲端區域和 vSphere 雲端區域，請使用非雲端元件。

## 雲端廠商元件

Amazon Web Services、Microsoft Azure 和 VMware vSphere 元件僅可以部署到相符的 AWS、Azure 或 vSphere 雲端區域。

您可以將與非雲端元件新增至包含特定廠商之特定於雲端的元件的藍圖。請注意，專案雲端區域在廠商方面提供的支援。

## 組態管理元件

組態管理元件取決於您的整合式應用程式。例如，Puppet 元件可監控和強制執行其他元件的組態。

## 如何在 vRealize Automation Cloud Assembly 中連線藍圖資源

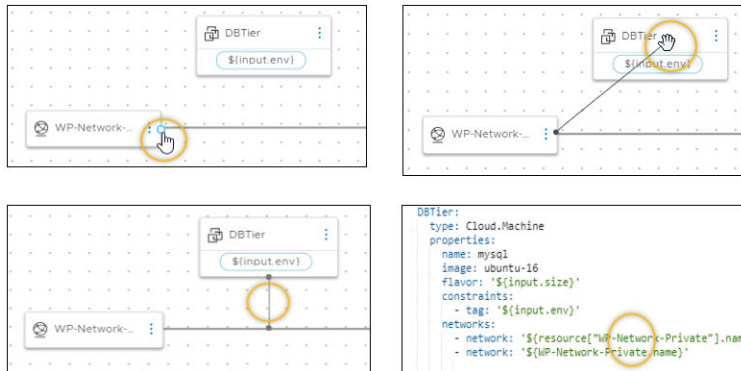
使用圖形設計畫布連線 vRealize Automation Cloud Assembly 藍圖資源。

如果資源符合連線的要求，您可以連線這些資源。例如：

- 將負載平衡器連線至機器叢集。
- 將機器連線到網路。
- 將外部儲存區連線至機器。

若要連線，請將游標暫留在資源邊緣上以顯示連線泡泡。按一下該泡泡並將其拖曳到目標資源，然後放開滑鼠按鈕。

在代碼編輯器中，來源資源的其他代碼會顯示在目標資源代碼中。



資源之間的實線表示這些資源必須最終處於相同的位置。雖然您可以在畫布上新增連線，但如果存在衝突的放置限制標籤，則部署會失敗。例如，如果連線的資源中有一個設定硬限制為測試 us-west-1 雲端區域，而另一個設定硬限制為生產 us-east-1 雲端區域，則部署會失敗。

## 如何建立有效的 vRealize Automation Cloud Assembly 藍圖代碼

在畫布中新增並連線 vRealize Automation Cloud Assembly 元件時，系統僅會建立起始程式碼。若要完全設定元件，請編輯代碼。

代碼編輯器可讓您直接輸入代碼或將內容值輸入表單中。為了協助直接建立代碼，vRealize Automation Cloud Assembly 編輯器包含語法完成和錯誤檢查功能。

**編輯器提示**
**範例**

可用的值

```

10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: ''
15     networks:
16       - name: small flavor
17       - name: large flavor
18 Cloud_Network_1:
19   type: Cloud.Network
20   properties:
21     name: ''
22     networkType: existing
  
```

允許的內容

```

10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: ''
15     tags: array
16     storage: object
17     remoteAccess: object
18     name: string
19     imageRef: string
20     count: integer
21     constraints: array
22     cloudConfig: string
  
```

子系內容

```

10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: ''
15     constraints:
16       - tag: string
17 Cloud_Network_1:
18   type: Cloud.Network
  
```

語法錯誤

Please correct errors in YAML editor before editing in canvas: row: 14, column: 17

```

10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: 'small'
15     constraints:
16     networks:
17       - name: '${Cloud_Network_1.name}'
18 Cloud_Network_1:
19   type: Cloud.Network
20   properties:
21     name: ''
22     networkType: existing
  
```

## 編輯器提示 範例

按 Ctrl+F  
可執行搜尋

```

1 inputs: {}
2 resources:
3   Cloud_Machine_1:
4     type: Cloud.Machine
5     properties:
6       image: ''
7       flavor: ''
8     networks:
9       - name: '${Cloud_Network_1.name}'
10  Cloud_Machine_2:
11    type: Cloud.Machine
12    properties:
13      image: ''
14      flavor: 'small'
15    constraints:
16    networks:
17      - name: '${Cloud_Network_1.name}'

```

## 選擇性參數

## 插入可選參數

- + attachedDisks
- + autoScaleConfiguration
- + cloudConfig
- + cloudConfigSettings

```

1 inputs: {}
2 resources:
3   Cloud_Machine_1:
4     type: Cloud.Machine
5     properties:
6       image: ''
7       flavor: ''
8     networks:
9       - name: '${Cloud_Network_1.name}'
10  Cloud_Machine_2:
11    type: Cloud.Machine
12    properties:
13      image: ''
14      flavor: 'small'
15    constraints:
16    networks:
17      - name: '${Cloud_Network_1.name}'

```

## 架構說明

對於所有自訂內容，還可以參閱 VMware [網站上的合併資源架構](#)。

**cloudConfig**

類型  
string

When provisioning an instance, machine cloud-init startup instructions from user data fields. Sample cloud config instructions:

```
#cloud-config
repo_update: true
repo_upgrade: all
packages:
- httpd
- mariadb-server

runcmd:
- [ sh, -c, 'amazon-linux-extras install -y
- systemctl start httpd
- sudo systemctl enable httpd
```

```

Tier:
type: Cloud.Machine
properties:
  name: mysql
  image: ubuntu-16
  flavor: '${input.size}'
  constraints:
    - tag: '${input.env}'
  networks:
    - name: '${resource["WP-Network-Private"]
    - name: '${WP-Network-Private.name}'
  remoteAccess:
    authentication: usernamePassword
    username: '${input.username}'
    password: '${input.userpassword}'
  cloudConfig:
    #cloud-config
    repo_update: true
    repo_upgrade: all

    packages:
      - mysql-server

    runcmd:
      - sed -e '/bind-address/ s/^#/#/' -i
      - service mysql restart
      - mysql -e "GRANT ALL PRIVILEGES ON *.
      - mysql -e "FLUSH PRIVILEGES;"
  attachedDisks: []
bTier:
type: Cloud.Machine

```

## 如何強化簡單 vRealize Automation Cloud Assembly 藍圖

存在進階 vRealize Automation Cloud Assembly 藍圖代碼的可能性，可使簡單的藍圖到達一個新層級。

此處說明的技術需要對基礎結構代碼有一定的瞭解。幸運的是，vRealize Automation Cloud Assembly 代碼一般人都看得懂，而且非常容易遵循。

## 使用者輸入如何自訂 vRealize Automation Cloud Assembly 藍圖

作為藍圖開發人員，您可以使用輸入參數，以便使用者可以在請求時進行自訂選取。

使用者提供輸入時，您不再需要儲存多個僅有細微差別的藍圖複本。此外，輸入可以為第 2 天作業準備藍圖，因為目錄使用者在更新部署時，系統會提示進行新輸入。

**注意** 某些內容變更會導致重新建立資源。例如，變更 `Cloud.Service.Azure.App.Service` 下的 `connection_string.name` 會刪除現有資源並建立新的資源。

設計輸入以支援第 2 天變更時，請決定是否允許刪除和重新建立資源的輸入。若要瞭解哪些內容會重新建立資源，請參閱 [VMware {code} 上 vRealize Automation 藍圖架構](#) 中的合併資源架構。

下列輸入顯示如何為 MySQL 資料庫伺服器建立一個藍圖，使用者可以在其中將該藍圖部署到不同的雲端資源環境中，並且每次都套用不同的容量和認證。

Environment	env.dev	▼ ⓘ
Database Tier Size *	small	▼ ⓘ
Database Username *	ouradmin	
Database Password *	••••••	
MySQL Data Disk Size	4	⬆️ ⬆️ ⓘ

## 如何定義藍圖輸入參數

在藍圖代碼中新增 `inputs` 區段，您可以在其中設定可選取的值。

在下列範例中，您可以選取機器大小、作業系統和叢集伺服器數目。

```
inputs:
  wp-size:
    type: string
    enum:
      - small
      - medium
    description: Size of Nodes
    title: Node Size
  wp-image:
    type: string
    enum:
      - coreos
      - ubuntu
```

```

    title: Select Image/OS
  wp-count:
    type: integer
    default: 2
    maximum: 5
    minimum: 2
    title: Wordpress Cluster Size
    description: Wordpress Cluster Size (Number of nodes)

```

如果您不滿意編輯程式碼，可以按一下程式碼編輯器輸入索引標籤，然後在其中輸入設定。下列範例顯示前面提到的 MySQL 資料庫的一些輸入。

The screenshot shows the 'Inputs' tab in the vRealize Automation Cloud Assembly interface. It displays a table of blueprint inputs. An 'Edit Blueprint Input: size' dialog is open, showing the following fields:

Name	Title	Type	Default Value
size	Tier Machine Size	string	
username	Database Username	string	
userpassword	Database Password	string	****
databaseDiskSize	MySQL Data Disk Size	number	4

The 'Edit Blueprint Input: size' dialog shows the following fields:

- Name: size
- Title: Tier Machine Size
- Description: Size of Nodes
- Type: string
- Encrypted: ☐

## 如何參考藍圖輸入參數

然後，在 `resources` 區段中，使用 `${input.property-name}` 語法參考輸入參數。

如果內容名稱中包含空格，請使用方括弧和雙引號進行分隔，而非使用點標記法：`${input["property name"]}`

**重要** 您在藍圖代碼中無法使用 `input` 一詞，除非指示輸入參數。

```

resources:
  WebTier:
    type: Cloud.Machine
    properties:

```

```
name: wordpress
flavor: '${input.wp-size}'
image: '${input.wp-image}'
count: '${input.wp-count}'
```

## 輸入內容清單

內容	說明
const	與 oneOf 搭配使用。實際值與易記標題相關聯。
預設值	輸入的預先填入值。 預設值必須為正確的類型。請勿輸入單字做為整數的預設值。
說明	輸入的使用者說明文字。
已加密	是否對使用者輸入的輸入進行加密 (true 或 false)。 密碼通常會加密。
列舉	允許值的下拉式功能表。 使用下列範例做為格式指南。 <pre>enum:   - value 1   - value 2</pre>
format	設定輸入的預期格式。例如, (25/04/19) 支援日期-時間。 允許在 vRealize Automation Service Broker 自訂表單中使用日期選擇器。
items	宣告陣列中的項目。支援數字、整數、字串、布林值或物件。
maxItems	陣列中可選取的項目數目上限。
maxLength	字串允許的字元數目上限。 例如, 若要限制欄位為 25 個字元, 請輸入 maxLength: 25。
上限	數字或整數的最大允許值。
minItems	陣列中可選取的項目數目下限。
minLength	字串允許的字元數目下限。
下限	數字或整數的最小允許值。
oneOf	允許使用者輸入表單為不太易記的值 (const) 顯示易記名稱 (title)。如果設定預設值, 則設定 const 而非 title。 適用於類型字串、整數和數字。
模式	規則運算式語法中的字串輸入允許的字元。 例如, '[a-z]+' 或 '[a-z0-9A-Z@#\$]+'
properties	宣告物件的 key:value 內容區塊。
readOnly	僅用來提供表單標籤。



內容	說明
title	與 oneOf 搭配使用。const 值的易記名稱。在部署時，標題顯示在使用者輸入表單上。
類型	數字、整數、字串、布林值或物件的資料類型。
writeOnly	在表單中隱藏星號後面的按鍵輸入。無法與 enum 搭配使用。顯示為 vRealize Automation Service Broker 自訂表單中的密碼欄位。

## 其他範例

### 包含列舉的字串

```
image:
  type: string
  title: Operating System
  description: The operating system version to use.
  enum:
    - ubuntu 16.04
    - ubuntu 18.04
  default: ubuntu 16.04

shell:
  type: string
  title: Default shell
  description: The default shell that will be configured for the created user.
  enum:
    - /bin/bash
    - /bin/sh
```

### 包含最小值和最大值的整數

```
count:
  type: integer
  title: Machine Count
  description: The number of machines that you want to deploy.
  maximum: 5
  minimum: 1
  default: 1
```

### 物件陣列

```
tags:
  type: array
  title: Tags
  description: Tags that you want applied to the machines.
  items:
    type: object
    properties:
```

```

key:
  type: string
  title: Key
value:
  type: string
  title: Value

```

### 包含易記名稱的字串

```

platform:
  type: string
  oneOf:
    - title: AWS
      const: platform:aws
    - title: Azure
      const: platform:azure
    - title: vSphere
      const: platform:vsphere
  default: platform:aws

```

### 包含模式驗證的字串

```

username:
  type: string
  title: Username
  description: The name for the user that will be created when the machine is provisioned.
  pattern: ^[a-zA-Z]+$

```

### 做為密碼的字串

```

password:
  type: string
  title: Password
  description: The initial password that will be required to logon to the machine.
  Configured to reset on first login.
  writeOnly: true

```

### 做為文字區域的字串

```

ssh_public_key:
  type: string
  title: SSH public key
  maxLength: 256

```

## 布林值

```
public_ip:
  type: boolean
  title: Assign public IP address
  description: Choose whether your machine should be internet facing.
  default: false
```

## 如何在 vRealize Automation Cloud Assembly 中設定元件部署順序

部署 vRealize Automation Cloud Assembly 藍圖時，一個元件可能需要先提供另一個元件。

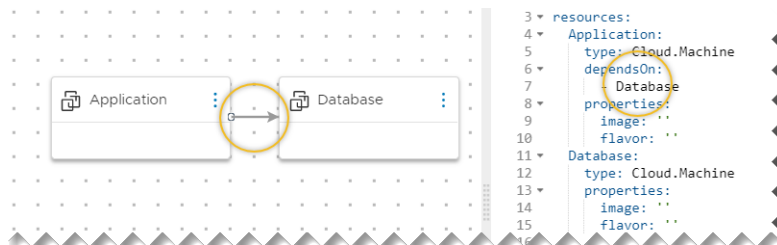
## 如何建立明確相依性

有時，一個元件需要先部署另一個元件。例如，可能需要先存在資料庫伺服器，才能建立應用程式伺服器並將其設定為存取資料庫伺服器。

明確相依性會設定部署時的建置順序，或用於縮小或擴充動作。您可以使用圖形設計畫布或代碼編輯器新增明確相依性。

- 設計畫布選項 - 從相依元件開始繪製連線，然後在要先部署的元件處結束。
- 代碼編輯器選項 - 將 `dependsOn` 內容新增至相依元件，並識別要先部署的元件。

明確相依性會在畫布中建立一個實心箭頭。



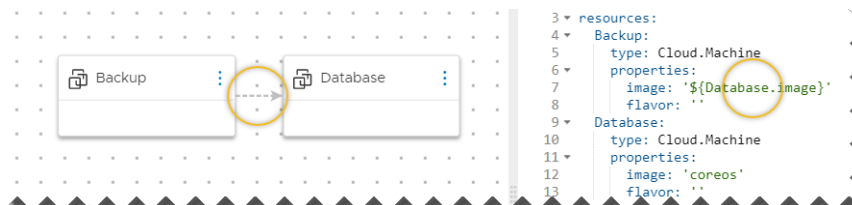
## 如何建立隱含相依性或內容繫結

有時，一個元件內容需要在另一個元件的內容中找到值。例如，備份伺服器可能需要正在備份的資料庫伺服器的作業系統映像，因此資料庫伺服器必須存在。

隱含相依性也稱為內容繫結，它會透過等到所需內容可用後再部署相依元件來控制建置順序。您可以使用代碼編輯器新增隱含相依性。

- 編輯相依元件，以新增識別該元件的內容和必須先存在的內容。

隱含相依性或內容繫結會在畫布中建立一個虛線箭頭。



## 如何使用運算式使 vRealize Automation Cloud Assembly 藍圖代碼更加多樣化

若要提高彈性，您可以向 vRealize Automation Cloud Assembly 藍圖代碼新增運算式。

運算式使用 `${expression}` 建構，如下列範例所示。

這些範例已經過剪除，僅顯示重要行。整個未編輯的藍圖會出現在結尾處。

### 範例

在部署時，允許使用者在遠端存取所需的加密金鑰中貼上：

```
inputs:
  sshKey:
    type: string
    maxLength: 500
resources:
  frontend:
    type: Cloud.Machine
    properties:
      remoteAccess:
        authentication: publicPrivateKey
        sshKey: '${input.sshKey}'
```

若要部署到 VMware Cloud on AWS，請將資料夾名稱設定為工作負載所需名稱：

```
inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere
resources:
  frontend:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'
```

在部署時，請使用符合所選環境的全小寫 env 標籤標記機器：

```
inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
```

```

    default: vSphere
resources:
  frontend:
    type: Cloud.Machine
    properties:
      constraints:
        - tag: '${"env:" + to_lower(input.environment)}'

```

將前端叢集中的機器數目設定為 1 (小型) 或 2 (大型)。請注意，大型叢集是透過清除程序設定的：

```

inputs:
  envsize:
    type: string
    enum:
      - Small
      - Large
resources:
  frontend:
    type: Cloud.Machine
    properties:
      count: '${input.envsize == "Small" ? 1 : 2}'

```

透過繫結到網路資源中找到的內容，將機器連結至相同的預設網路：

```

resources:
  frontend:
    type: Cloud.Machine
    properties:
      networks:
        - name: '${resource.Cloud_Network_1.name}'
  apitier:
    type: Cloud.Machine
    properties:
      networks:
        - name: '${resource.Cloud_Network_1.name}'
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      name: Default
      networkType: existing

```

加密提交至 API 的存取認證：

```

resources:
  apitier:
    type: Cloud.Machine
    properties:
      cloudConfig: |
        #cloud-config
        runcmd:
          - export apikey=${base64_encode(input.username:input.password)}
          - curl -i -H 'Accept:application/json' -H 'Authorization:Basic :$apikey' http://
example.com

```

探索 API 機器的位址：

```
resources:
  frontend:
    type: Cloud.Machine
    properties:
      cloudConfig: |
        runcmd:
          - echo ${resource.apitier.networks[0].address}
  apitier:
    type: Cloud.Machine
    properties:
      networks:
        - name: '${resource.Cloud_Network_1.name}'
```

## 完成藍圖

```
inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere
  sshKey:
    type: string
    maxLength: 500
  envsize:
    type: string
    enum:
      - Small
      - Large
resources:
  frontend:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'
      image: ubuntu
      flavor: medium
      count: '${input.envsize == "Small" ? 1 : 2}'
      remoteAccess:
        authentication: publicPrivateKey
        sshKey: '${input.sshKey}'
      cloudConfig: |
        packages:
          - nginx
        runcmd:
          - echo ${resource.apitier.networks[0].address}
    constraints:
      - tag: '${"env:" + to_lower(input.environment)}'
    networks:
```

```

    - name: '${resource.Cloud_Network_1.name}'
  apitier:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'
      image: ubuntu
      flavor: small
      cloudConfig: |
        #cloud-config
        runcmd:
          - export apikey=${base64_encode(input.username:input.password)}
          - curl -i -H 'Accept:application/json' -H 'Authorization:Basic :$apikey' http://
example.com
      remoteAccess:
        authentication: publicPrivateKey
        sshKey: '${input.sshKey}'
      constraints:
        - tag: '${"env:" + to_lower(input.environment)}'
      networks:
        - name: '${resource.Cloud_Network_1.name}'
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      name: Default
      networkType: existing
      constraints:
        - tag: '${"env:" + to_lower(input.environment)}'

```

## vRealize Automation Cloud Assembly 藍圖運算式語法

vRealize Automation Cloud Assembly 藍圖運算式語法會公開運算式的所有功能。

僅在[如何使用運算式使 vRealize Automation Cloud Assembly 藍圖代碼更加多樣化](#)顯示的範例中部分表示語法。

### 文字

支援下列文字：

- 布林值 (true 或 false)
- 整數
- 浮點
- 字串

反斜線逸出雙引號、單引號和反斜線本身：

" 逸出為 \"

' 逸出為 \'

\ 逸出為 \\

只需要在用相同類型的引號括住的字串內逸出引號，如下列範例所示。

```
"I am a \"double quoted\" string inside \"double quotes\"."
```

#### ■ Null

### 環境變數

環境名稱：

- orgId
- projectId
- projectName
- deploymentId
- deploymentName
- blueprintId
- blueprintVersion
- blueprintName
- requestedBy (使用者)
- requestedAt (時間)

語法：

```
env.ENV_NAME
```

範例：

```
${env.blueprintId}
```

### 資源變數

使用資源變數可繫結至來自其他資源的資源內容。

語法：

```
resource.RESOURCE_NAME.PROPERTY_NAME
```

範例：

- \${resource.db.id}
- \${resource.db.networks[0].address}
- \${resource.app.id} (傳回非叢集化資源的字串，未指定計數。傳回叢集化資源的陣列)。
- \${resource.app[0].id} (傳回叢集化資源的第一個項目)。

### 資源本身變數

僅允許對支援配置階段的資源使用資源本身變數。只有在配置階段完成後，才可使用資源本身變數 (或僅具有值集)。



語法：

```
self.property_name
```

範例：

```
${self.address} (傳回配置階段期間指派的位址)。
```

請注意，對於名為 `resource_x`、`self.property_name` 和 `resource.resource_x.property_name` 的資源是相同的，且同時視為自我參考。

## 叢集計數索引

語法：

```
count.index
```

範例：

```
${count.index == 0 ? "primary" : "secondary"} (傳回叢集化資源的節點類型)。
```

限制：

不支援將 `count.index` 用於資源配置。例如，下列容量運算式參考在輸入時建立的磁碟陣列中的位置時會失敗。

```
inputs:
  disks:
    type: array
    minItems: 0
    maxItems: 12
    items:
      type: object
      properties:
        size:
          type: integer
          title: Size (GB)
          minSize: 1
          maxSize: 2048
resources:
  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: '${input.disks[count.index].size}'
      count: '${length(input.disks)}'
```

## 條件

語法：

- 等號比較運算子為 `==` 和 `!=`。
- 關係運算子為 `<` `>` `<=` 和 `>=`。
- 邏輯運算子為 `&&` `||` 和 `!`。
- 條件式使用模式：

*condition-expression ? true-expression : false-expression*

範例：

```
${input.count < 5 && input.size == 'small'}
```

```
${input.count < 2 ? "small" : "large"}
```

## 算術運算子

語法：

運算子為 `+` `-` `/` `*` 和 `%`。

範例：

```
${(input.count + 5) * 2}
```

## 字串串連

語法：

`${'ABC' + 'DEF'}` 評估為 `ABCDEF`。

## 運算子 `[]` 和 `.`

在統一處理 `[]` 和 `.` 運算子方面，運算式遵循 ECMAScript。

因此，`expr.identifier` 相當於 `expr["identifier"]`。識別碼用來建構其值為 `identifier` 的文字，然後將 `[]` 運算子與該值搭配使用。

範例：

```
${resource.app.networks[0].address}
```

此外，如果內容中包含空格，請使用方括弧和雙引號進行分隔，而非使用點標記法。

不正確：

```
input.operating system
```

正確：

```
input["operating system"]
```

## 對應建構

語法：

```
${{'key1':'value1', 'key2':input.key2}}
```

## 陣列建構

語法：

```
${['key1','key2']}
```

範例：

```
${[1,2,3]}
```

## 函數

語法：

```
${function(arguments...)}
```

範例：

```
${to_lower(resource.app.name)}
```

表 6-1. 函數

函數	說明
abs(number)	絕對數值
floor(number)	傳回小於或等於引數且等於數學整數的最大 (最接近正無限大) 值
ceil(number)	傳回大於或等於引數且等於數學整數的最小 (最接近無限大的負數) 值
to_lower(str)	將字串轉換為小寫
to_upper(str)	將字串轉換為大寫
contains(array, value)	檢查陣列是否包含值
contains(string, value)	檢查字串是否包含值
join(array, delim)	加入含分隔符號的字串陣列，並傳回字串
split(string, delim)	分割含分隔符號的字串，並傳回字串陣列
slice(array, begin, end)	傳回從開始索引到結束索引的陣列配置
reverse(array)	反向陣列項目
starts_with(subject, prefix)	檢查主體字串是否以前置詞字串開頭
ends_with(subject, suffix)	檢查主體字串是否以後置詞字串結尾
replace(string, target, replacement)	將包含目標字串的字串取代為目標字串
substring(string, begin, end)	傳回從開始索引到結束索引的字串的子字串
format(format, values...)	傳回使用 Java 類 <a href="#">Formatter</a> 格式的格式化字串和值。
keys(map)	傳回對應的索引鍵
values(map)	傳回對應的值
merge(map, map)	傳回合併的對應
length(string)	傳回字串長度
length(array)	傳回陣列長度
max(array)	傳回數字陣列中的最大值
min(array)	傳回數字陣列中的最小值

表 6-1. 函數 (續)

函數	說明
sum(array)	傳回數字陣列中所有值的總和
avg(array)	傳回數字陣列中所有值的平均值
digest(value, type)	傳回使用支援類型 (md5、sha1、sha256、sha384、sha512) 的值的摘要
to_string(value)	傳回值的字串表示法
to_number(string)	以數字剖析字串
not_null(array)	返回不為 Null 的第一個項目
base64_encode(string)	傳回 base64 編碼值
base64_decode(string)	傳回解碼的 base64 值
now()	以 ISO-8601 格式傳回目前時間
uuid()	傳回隨機產生的 UUID
from_json(string)	剖析 json 字串
to_json(value)	將值序列化為 json 字串
json_path(value, path)	使用 <a href="#">XPath for JSON</a> 根據值評估路徑。
matches(string, regex)	檢查字串是否符合 regex 運算式
url_encode(string)	使用 url 編碼規格對字串進行編碼
trim(string)	移除前置空格和尾端空格

## 如何在 vRealize Automation Cloud Assembly 藍圖中自動初始化機器

您可以透過 vSphere 自訂規格或直接執行命令，在 vRealize Automation Cloud Assembly 中套用機器初始化。

藍圖代碼中的內容可依名稱參考 vSphere 自訂規格。或者，您可以將 cloudConfig 區段新增至藍圖，並在其中內嵌特定命令。

如果您嘗試合併內嵌的命令和自訂規格初始化，請務必謹慎。它們未正式相容，因此一起使用時可能會產生不一致或不必要的結果。

如需自訂規格如何影響 cloudConfig 區段中命令的範例，請參閱[如何部署具有靜態 IP 位址的 Linux 機器](#)。

## vRealize Automation Cloud Assembly 藍圖中的 vSphere 自訂規格

自訂規格可讓您在部署時 (部署至基於 vSphere 的雲端區域時)，套用客體作業系統設定。

自訂規格必須存在於 vSphere 中您部署到的目標上。

直接編輯藍圖代碼。下列範例指向 vSphere 上 WordPress 主機的 `cloud-assembly-linux` 自訂規格。

```
resources:
  WebTier:
    type: Cloud.vSphere.Machine
    properties:
      name: wordpress
      cpuCount: 2
      totalMemoryMB: 1024
      imageRef: 'Template: ubuntu-18.04'
      customizationSpec: 'cloud-assembly-linux'
      resourceGroupName: '/Datacenters/Datacenter/vm/deployments'
```

### 自訂規格與初始化命令

如果您希望佈建體驗與您目前在 vSphere 中執行的動作相符，則繼續使用自訂規格可能是最佳方式。但是，若要擴充到混合或多雲端佈建，則在藍圖的 `cloudConfig` 區段中內嵌初始化命令是較為中性的方法。

如需有關藍圖中 `cloudConfig` 區段的詳細資訊，請參閱 [vRealize Automation Cloud Assembly 藍圖中的組態命令](#)。

### vRealize Automation Cloud Assembly 藍圖中的組態命令

您可以向 vRealize Automation Cloud Assembly 藍圖代碼新增 `cloudConfig` 區段，並在該區段中新增您在部署時執行的機器初始化命令。

- Linux - 初始化命令遵循開放原始碼 [cloud-init](#) 標準。
- Windows - 初始化命令使用 [Cloudbase-init](#)。

---

**備註** Linux [cloud-init](#) 和 Windows [Cloudbase-init](#) 不共用相同的語法。一個作業系統的 `cloudConfig` 區段在另一個作業系統的機器映像中不起作用。

---

您可以使用初始化命令在建立執行個體時自動套用資料或設定，這樣可以自訂使用者、權限、安裝或任何其他以命令為基礎的作業。範例包括：

- 設定主機名稱
- 產生並設定 SSH 私密金鑰
- 安裝套件

在 vRealize Automation Cloud Assembly 中，您也可以設定基礎結構時預先將初始化命令新增至機器映像。參考來源映像的所有藍圖都會進行相同的初始化。

**重要** 您可能有一個映像對應和一個藍圖，且兩者皆包含一個初始化命令。在部署時，兩個命令將合併，並且 vRealize Automation Cloud Assembly 會執行合併的命令。

當兩個命令中出現相同的命令但包含不同的參數時，系統僅執行映像對應命令。

如需其他詳細資料，請參閱[進一步瞭解 vRealize Automation Cloud Assembly 中的映像對應](#)。

下列範例 cloudConfig 區段取自適用於 Linux 系統的 MySQL 伺服器的 [WordPress 使用案例：建立基本藍圖](#) 藍圖代碼。

若要確保正確解譯命令，請一律包括管道字元 cloudConfig: |，如下所示。

如果 cloud-init 指令碼行為異常，請在疑難排解時檢查 /var/log/cloud-init-output.log 中已擷取的主控制台輸出。如需有關 cloud-init 的詳細資訊，請參閱 [cloud-init 說明文件](#)。

```
cloudConfig: |
    #cloud-config
    repo_update: true
    repo_upgrade: all
    packages:
      - apache2
      - php
      - php-mysql
      - libapache2-mod-php
      - php-mcrypt
      - mysql-client
    runcmd:
      - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://
wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
mywordpresssite --strip-components 1
      - i=0; while [ $i -le 5 ]; do mysql --connect-timeout=3 -h $
{DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;
i=$((i+1)); done
      - mysql -u root -pmysqlpassword -h ${DBTier.networks[0].address} -e "create database
wordpress_blog;"
      - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
mywordpresssite/wp-config.php
      - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME',
'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e
s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER', 'root' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_PASSWORD',
'password_here' );"/"define( 'DB_PASSWORD', 'mysqlpassword' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_HOST',
'localhost' );"/"define( 'DB_HOST', '${DBTier.networks[0].address}' );"/ /var/www/html/
mywordpresssite/wp-config.php
      - service apache2 reload
```

## 如何部署具有靜態 IP 位址的 Linux 機器

部署到 vSphere 時，靜態 IP 位址需要 vRealize Automation Cloud Assembly 產生 vSphere 自訂規格，這可能會影響 cloud-init 命令。

### 問題

- vRealize Automation Cloud Assembly 藍圖包含 `assignment: static`，可將靜態 IP 位址套用至 vSphere 虛擬機器。
- 此藍圖還包含 cloudConfig 區段，其中包括使用 cloud-init 執行的初始化命令。
- 若要為虛擬機器提供靜態 IP，vRealize Automation Cloud Assembly 會動態產生要套用的 vSphere 自訂規格。
- 每當套用自訂規格時，最後一個作業將會重新啟動虛擬機器。
- 自訂規格不知道 cloud-init 命令正在執行中，因此，重新啟動操作會中斷這些命令。
- cloud-init 命令僅在首次開機時執行，並且在中斷後不會自動復原。
- 產生的虛擬機器僅保持部分設定狀態。

### 因應措施

建立包含定時停用 cloud-init 的機器範本。然後，根據範本部署機器，以便自訂規格和重新啟動能夠先於 cloud-init 執行。

### 範例程序 - Ubuntu 18.04

下列步驟適用於 Ubuntu 18.04。您可能需要進行調整，使此處顯示的構想適用於其他 Linux 版本或產品。

- 1 建立虛擬機器，並使用您所需的版本更新和套件使其保持最新狀態。

請注意，其他 Linux 產品可能並未預先安裝 cloud-init，但 Ubuntu 18.04 已預先安裝。

- 2 重新設定 cloud-init，將資料來源設定為 OVF。

```
sudo dpkg-reconfigure cloud-init
```

- 3 編輯以下檔案。

```
/etc/cloud/cloud.cfg
```

- a 新增以下行以啟用傳統客體作業系統自訂 (GOSC)。

```
disable_vmware_customization: true
```

- b 請確保網路組態已啟用。刪除停用設定或為其添加註解 (如果存在)。

```
network:
  # config: disabled
```

或者，檢查下列目錄中的所有組態檔。

```
/etc/cloud/cloud.cfg.d/*
```

刪除包含 `network: {config: disabled}` 設定的任何檔案。

## 4 編輯以下檔案。

```
/usr/lib/tmpfiles.d/tmp.conf
```

- 透過為設定添加註解，防止清除暫存目錄。

```
# D /tmp 1777 root root -
```

## 5 編輯以下檔案。

```
/lib/systemd/system/open-vmtools.service
```

- 透過在 [Unit] 區段下新增以下行，將 open-vmtools 設定為在 dbus.service 之後啟動。

```
After=dbus.service
```

## 6 建立新的空白檔案，該檔案將停用 cloud-init。

```
sudo touch /etc/cloud/cloud-init.disabled
```

## 7 建立 re\_init.sh 指令碼。在針對自訂規格暫停的 cron 工作延遲後，指令碼會重新啟用並初始化 cloud-init。

```
sudo rm -rf /etc/cloud/cloud-init.disabled
sudo cloud-init init
sleep 20
sudo cloud-init modules --mode config
sleep 20
sudo cloud-init modules --mode final
```

## 8 為指令碼新增執行權限。

```
sudo chmod +x re_init.sh
```

## 9 建立將在啟動時睡眠 90 秒後執行的 cron 工作。輸入 crontab -e，然後輸入下列內容：

```
@reboot ( sleep 90 ; sh /script_path/delay_init.sh )
```

如果完成自訂規格和重新啟動需要更長的時間，則可以套用 90 秒以上。

## 10 建立用於清理範本的 cleaner.sh 指令碼。將 cloudadmin 取代為在作業系統安裝期間設定的您自己的使用者。

此範例指令碼是特定於 Ubuntu 的。若要為其他 Linux 產品建立指令碼，請務必包含反白顯示的必要區段。

```
#!/bin/bash

# Add usernames to add to /etc/sudoers for passwordless sudo
users=("ubuntu" "cloudadmin")

for user in "${users[@]}"
do
cat /etc/sudoers | grep ^$user
RC=$?
if [ $RC != 0 ]; then
bash -c "echo \"$user ALL=(ALL) NOPASSWD:ALL\" >> /etc/sudoers"
```



```

fi
done

#grab Ubuntu Codename
codename="$(lsb_release -c | awk {'print $2'})"

#Stop services for cleanup
service rsyslog stop

#clear audit logs
if [ -f /var/log/audit/audit.log ]; then
cat /dev/null > /var/log/audit/audit.log
fi
if [ -f /var/log/wtmp ]; then
cat /dev/null > /var/log/wtmp
fi
if [ -f /var/log/lastlog ]; then
cat /dev/null > /var/log/lastlog
fi

#cleanup persistent udev rules
if [ -f /etc/udev/rules.d/70-persistent-net.rules ]; then
rm /etc/udev/rules.d/70-persistent-net.rules
fi

#cleanup /tmp directories
rm -rf /tmp/*
rm -rf /var/tmp/*

#cleanup current ssh keys
#rm -f /etc/ssh/ssh_host_*

#cat /dev/null > /etc/hostname

#cleanup apt
apt-get clean

#Clean Machine ID

truncate -s 0 /etc/machine-id
rm /var/lib/dbus/machine-id
ln -s /etc/machine-id /var/lib/dbus/machine-id

#Clean Cloud-init
cloud-init clean --logs --seed

#cleanup shell history
history -w
history -c

```

## 11 為範本清理指令碼新增執行權限。

```
sudo chmod +x cleaner.sh
```

- 12 在 Ubuntu 18.04 中，清理指令碼需要 root 權限。編輯以下檔案。

```
/etc/ssh/sshd_config
```

- a 請確保可以切換為 root 使用者。

```
PermitRootLogin yes
```

- b 為根使用者設定密碼。

```
sudo passwd root
```

- 13 執行清理指令碼。

```
sudo ./script_path/cleaner.sh
```

- 14 (選用) 出於安全性考慮，請還原步驟 12 以防進一步的 root 登入。

- 15 關閉虛擬機器，然後使用 vSphere 將其轉換為範本。

### 範本更新

cron 工作會在您每次更新範本時執行。如果更新所需的時間超過延遲時間 (例如 90 秒)，則需要重新新增 `/etc/cloud/cloud-init.disabled` 檔案，並在關閉範本前重新執行清理指令碼。否則，在首次開機時將不會停用 cloud-init，且自訂規格重新啟動會恢復以中斷 cloud-init 命令。

### 疑難排解

如果您懷疑 vSphere 自訂規格會阻止 cloud-init 完成，請暫時停用自訂規格，並確定 cloud-init 是否可以按預期完成。若要暫時停用自訂規格，請使用 `customizeGuestOS: false` 內容。

```
properties:
  image: ubuntu
  cpuCount: 1
  totalMemoryMB: 8192
  customizeGuestOS: false
```

## 如何使 vRealize Automation Cloud Assembly 部署等待初始化

有時，在繼續進行 vRealize Automation Cloud Assembly 部署之前，需要完全啟動虛擬機器。

例如，如果部署仍在安裝套件和啟動 Web 伺服器的機器，則可能會導致快速使用者嘗試在應用程式可用之前存取該應用程式的情況。

使用此功能時，請注意下列考量事項。

- 此功能使用 `cloud-init phone_home` 模組，並在部署 Linux 機器時可用。
- 由於 `Cloudbase-init` 限制，`phone_home` 不適用於 Windows。
- `phone_home` 可以像明確相依性一樣影響部署順序，但在計時和處理選項方面具有更大的彈性。  
請參閱 [如何在 vRealize Automation Cloud Assembly 中設定元件部署順序](#)。
- `phone_home` 需要藍圖中的 `cloudConfig` 命令。

- 您的創造力是一個因素。cloudConfig 區段可能包括作業之間的內嵌等待時間，可與 phone\_home 搭配使用。
- 如果機器範本中已包含 phone\_home 模組設定，則以藍圖為基礎的 phone\_home 將無法正常運作。
- 機器必須具有返回 vRealize Automation Cloud Assembly 的輸出通訊存取權限。

若要在 vRealize Automation Cloud Assembly 中使用 phone\_home 等待機器初始化，請將 cloudConfigSettings 區段新增至藍圖：

```
cloudConfigSettings:
  phoneHomeShouldWait: true
  phoneHomeTimeoutSeconds: 600
  phoneHomeFailOnTimeout: true
```

內容	說明
phoneHomeShouldWait	是否等待初始化 (true 或 false)。
phoneHomeTimeoutSeconds	何時決定是否要繼續部署，即使初始化仍在執行中。預設值為 10 分鐘。
phoneHomeFailOnTimeout	是否在逾時後繼續部署 (true 或 false)。請注意，即使繼續進行，部署仍可能因個別原因而失敗。

## 如何在 vRealize Automation Cloud Assembly 藍圖中啟用遠端存取

若要遠端存取 vRealize Automation Cloud Assembly 已部署的機器，請在部署之前，將內容新增至該機器的藍圖。

針對遠端存取，您可以設定下列其中一個驗證選項。

**備註** 在需要複製金鑰的情況下，您也可以在此藍圖中建立一個 cloudConfig 區段，以便在佈建時自動複製金鑰。此處沒有詳細指示，但[如何在 vRealize Automation Cloud Assembly 藍圖中自動初始化機器](#)提供了有關 cloudConfig 的一般資訊。

## 在 vRealize Automation Cloud Assembly 佈建時產生金鑰配對

如果您沒有自己的公開金鑰-私密金鑰配對可用於遠端存取驗證，則可以讓 vRealize Automation Cloud Assembly 產生金鑰配對。

使用下列程式碼做為準則。

- 1 在 vRealize Automation Cloud Assembly 中，將 remoteAccess 內容新增至藍圖，然後再進行佈建，如範例所示。

username 是可選的。如果遺漏，系統會產生一個隨機識別碼做為 username。

範例：

```
type: Cloud.Machine
properties:
  name: our-vm2
  image: Linux18
  flavor: small
  remoteAccess: authentication: generatedPublicPrivatekey username: testuser
```

- 2 在 vRealize Automation Cloud Assembly 中，從藍圖佈建機器，並使其進入啟動狀態。  
佈建程序會產生金鑰。

- 3 在部署 > 拓撲內容中找到金鑰名稱。
- 4 使用雲端提供者介面 (例如 vSphere Client) 來存取已佈建的機器命令列。
- 5 授與私密金鑰的讀取權限。

```
chmod 600 key-name
```

- 6 移至 vRealize Automation Cloud Assembly 部署，選取機器，然後按一下動作 > 取得私密金鑰。
- 7 將私密金鑰檔案複製到本機電腦。

一般的本機檔案路徑是 `/home/username/.ssh/key-name`。

- 8 開啟遠端 SSH 工作階段，然後連線至已佈建的機器。

```
ssh -i key-name user-name@machine-ip
```

## 為 vRealize Automation Cloud Assembly 提供您自己的公開金鑰-私密金鑰配對

許多公司會建立和發佈自己的公開金鑰-私密金鑰配對以進行驗證。

使用下列程式碼做為準則。

- 1 在本機環境中，取得或建立您的公開金鑰-私密金鑰配對。  
現在，只需在本機產生並儲存金鑰。
- 2 在 vRealize Automation Cloud Assembly 中，將 `remoteAccess` 內容新增至藍圖，然後再進行佈建，如範例所示。

`sshKey` 包括在公開金鑰檔案 `key-name.pub` 中找到的長英數字元。

`username` 是可選的，建立後可用來登入。如果遺漏，系統會產生一個隨機識別碼做為 `username`。

範例：

```
type: Cloud.Machine
properties:
  name: our-vm1
  image: Linux18
  flavor: small
  remoteAccess:
```

```
authentication: publicPrivateKey
sshKey: ssh-rsa Iq+5aQgBP3ZNT4o1baP5Ii+dstIcowRRkyobbfpA1mj9tslf
qGxvU66PX9IeZax5hZvNWFgJw6ag+Z1zndOLhVdVoW49f274/mIRild7Uuw...
username: testuser
```

- 3 在 vRealize Automation Cloud Assembly 中，從藍圖佈建機器，並使其進入啟動狀態。
- 4 使用雲端廠商用戶端存取已佈建的機器。
- 5 將公開金鑰檔案新增至機器上的主資料夾。使用您在 `remoteAccess.sshKey` 中指定的金鑰。
- 6 確認您的本機機器上存在對應的私密金鑰檔案。

金鑰通常是不含 `.pub` 副檔名的 `/home/username/.ssh/key-name`。

- 7 開啟遠端 SSH 工作階段，然後連線至已佈建的機器。

```
ssh -i key-name user-name@machine-ip
```

## 為 vRealize Automation Cloud Assembly 提供 AWS 金鑰配對

透過將 AWS 金鑰配對名稱新增至藍圖，您可以從遠端存取 vRealize Automation Cloud Assembly 部署至 AWS 的機器。

請注意，AWS 金鑰配對是特定於區域的。如果您將工作負載佈建到 `us-east-1`，金鑰配對皆必須存在於 `us-east-1` 中。

使用下列程式碼做為準則。此選項僅適用於 AWS 雲端區域。

```
type: Cloud.Machine
properties:
  image: Ubuntu
  flavor: small
  remoteAccess: authentication: keyPairName keyPair: cas-test
constraints:
  - tag: 'cloud:aws'
```

## 向 vRealize Automation Cloud Assembly 提供使用者名稱和密碼

透過將使用者名稱和密碼新增至藍圖，您可以對 vRealize Automation Cloud Assembly 部署的機器進行簡單的遠端存取。

雖然安全性較低，但透過使用者名稱和密碼遠端登入可能就是您的情況所需的全部。請注意，某些雲端廠商或組態可能不支援此安全性較低的選項。

- 1 在 vRealize Automation Cloud Assembly 中，將 `remoteAccess` 內容新增至藍圖，然後再進行佈建，如範例所示。

將使用者名稱和密碼設定為您想要用來登入的帳戶。

範例：

```
type: Cloud.Machine
properties:
  name: our-vm3
  image: Linux18
  flavor: small
  remoteAccess: authentication: usernamePassword username: testuser password: admin123
```

- 2 在 vRealize Automation Cloud Assembly 中，從藍圖佈建機器，並使其進入啟動狀態。
- 3 移至雲端廠商的介面，並存取已佈建的機器。
- 4 在已佈建的機器上，建立或啟用該帳戶。
- 5 從本機器上，開啟至已佈建機器的 IP 位址或 FQDN 的遠端工作階段，並像往常一樣透過使用者名稱和密碼登入。

## 如何儲存不同版本的 vRealize Automation Cloud Assembly 藍圖

做為藍圖開發人員，您可以安全地擷取工作藍圖的快照，然後再冒著風險進行進一步變更。

在部署時，您可以選取任何要部署的版本。

### 如何擷取藍圖版本

從設計頁面中，按一下**版本**，並提供名稱。

此名稱必須為英數字元，不能有空白，並且僅允許使用句點、連字號和底線做為特殊字元。

### 如何還原較舊版本

在設計頁面中，按一下**版本歷程記錄**。

在左側選取較舊版本，以在畫布和代碼編輯器中對其進行檢查。找到所需的版本時，按一下**還原**。還原將覆寫目前的草稿，而不會移除任何具名版本。

The screenshot displays the vRealize Automation Cloud Assembly interface. On the left, a sidebar shows the version history with two entries: '1.1' and '1'. The main canvas shows a blueprint design with components like 'WP-Network', 'DBTier', and 'WebTier'. On the right, a code editor shows the blueprint's JSON configuration, including details like 'name: UIBasicUK2', 'iteration: 1', and various input parameters for environment, size, and database tier.

## 如何將版本發佈至 vRealize Automation Service Broker 使用者

在設計頁面中，按一下**版本歷程記錄**。

在左側選取一個版本並按一下**發佈**。對目前草稿進行版本化之前，您無法發佈草稿。

如果發佈了多個版本的藍圖，vRealize Automation Service Broker 將使用最新版本。

## 如何比較藍圖版本

變更和版本累積時，您可能要確定它們之間的差異。

從 [版本歷程記錄] 視圖中，選取一個版本，並按一下**差異**。然後，從**差異比較**下拉式清單中選取要比較的另一個版本。

請注意，您可以在檢閱代碼差異或視覺拓撲差異之間切換。

圖 6-1. 代碼差異

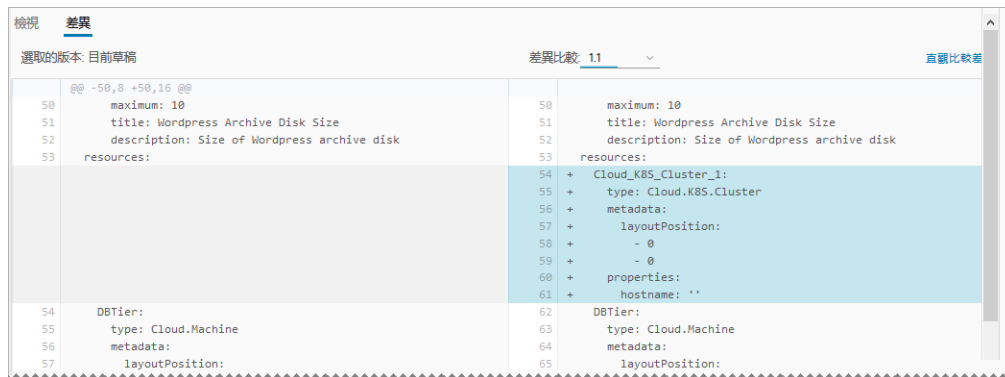
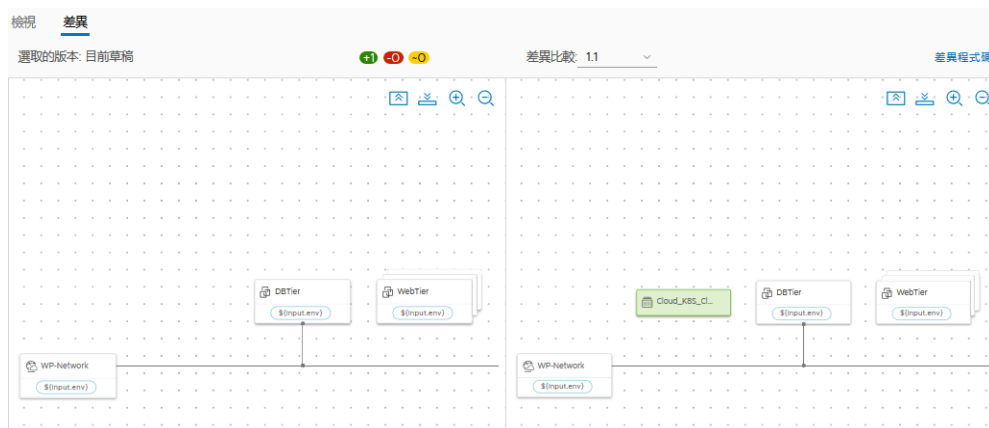


圖 6-2. 視覺拓撲差異



## 如何複製藍圖

雖然與儲存版本不同，但從設計頁面中選取**動作 > 複製**會建立目前藍圖的複本，以用於替代開發。

## 如何使用 vRealize Automation Cloud Assembly 自訂已部署資源的名稱

做為雲端或專案管理員，您對環境中的資源有一個指定的命名慣例，並且您希望已部署的資源遵循這些慣例，而無需使用者互動。您可以為 vRealize Automation Cloud Assembly 專案中的所有部署建立命名範本。

例如，您的主機命名慣例是為資源加上 *projectname-sitecode-costcenter-whereDeployed-identifier* 前置詞。您可以為每個專案的機器設定自訂命名範本。部分範本變數會在部署時從系統中提取，其他範本變數則以專案自訂內容為基礎。

所有資源名稱必須是唯一的。使用增量數字內容以確保唯一性。包括由 vRealize Automation Cloud Assembly 命名的部署在內的所有部署的數字將會遞增。隨著系統變得更強大，編號可能會看起來是隨機的，但它們仍可確保唯一性。

除了此處提供的範例之外，還可以新增使用者名稱、已使用的映像、其他內建選項，以及簡單字串。當您建置範本時，會提供有關可能選項的提示。

請注意，您看到的部分值只是使用案例範例而已。不能在您的環境中按原樣使用這些值。請考量您可以在何處從範例值替換或推斷，以符合您自己的雲端基礎結構和部署管理需求。

### 必要條件

- 確認您知道要用於專案中的部署的命名慣例。
- 此程序假設您具有或可以建立一個簡易藍圖，用於測試您的自訂主機前置詞命名。

### 程序

- 1 選取**基礎結構 > 專案**。
- 2 選取現有專案或建立新的專案。
- 3 在**佈建索引標籤**上，找到 [自訂內容] 區段，並建立站台代碼的內容和成本中心值。

您可以在此處將此處顯示的值取代為與您的環境相關的值。

自訂內容

指定應新增到此專案中的所有請求的自訂內容。 ①

定義自訂內容	名稱	值
	siteCode	BGL
	costCenter	IT-research

自訂命名

指定要用於在此專案中佈建的機器、網路、安全群組和磁碟的命名範本。

範本

`$(project.name)-$(resource.siteCode)-$(resource` ①

Hint: Avoid conflicting names by generating digits in names. `$(#####)`

- a 使用名稱 **siteCode** 和值 **BGL** 建立自訂內容。
- b 使用名稱 **costCenter** 和值 **IT-research** 新增另一個自訂內容。



#### 4 找到 [自訂命名] 區段並新增以下範本。

```
${project.name}-${resource.siteCode}-${resource.costCenter}-${endpoint.name}-${#####}
```

您可以在字串中進行複製，但如果這是您第一個命名範本，請考慮在建置範本時使用提示文字和快速選取。

#### 5 部署與專案相關聯的藍圖，以確認自訂名稱是否會套用至資源。

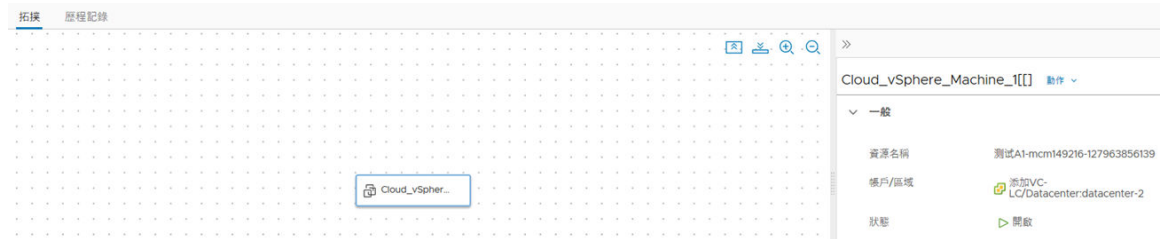
a 按一下**藍圖**索引標籤，然後按一下與專案相關聯的藍圖。

b 部署藍圖。

**部署**索引標籤隨即開啟，顯示您的部署正在進行中。

c 部署完成後，按一下部署名稱。

d 請注意，在**拓撲**索引標籤上，您的自訂名稱是右窗格中的資源名稱。



#### 6 如果部署了測試藍圖以驗證命名慣例，則可以刪除部署。

##### 後續步驟

為其他專案建立自訂命名範本。

## 什麼是 vRealize Automation 資源內容

在 vRealize Automation 基礎結構即代碼編輯器中，您可以按一下滑鼠或將游標暫留以取得語法和代碼完成說明。但是，若要檢視完整的藍圖資源內容集 (有時稱為自訂內容)，請參閱合併資源架構。

可以從 VMware {code} 網站取得架構。按一下連結，然後按一下**模型**，即可列出適用於藍圖的資源物件。

##### ■ VMware {code} 上的 vRealize Automation 資源類型架構

## 可使用的一些藍圖代碼範例

vRealize Automation Cloud Assembly 中的藍圖代碼在組合與應用方面幾乎無限制。

通常，成功程式碼的範例是進一步開發的最佳起點。遵循範例時，請進行取代，以在資源名稱、值等方面套用您的站台設定。

## vRealize Automation Cloud Assembly 藍圖中的 vSphere 機器範例

下列基本範例定義 vRealize Automation Cloud Assembly 藍圖中的 vSphere 資源。

資源	範例藍圖
具有 CPU、記憶體和作業系統的 vSphere 虛擬機器	<pre>resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       name: demo-machine       cpuCount: 1       totalMemoryMB: 1024       image: ubuntu</pre>
具有資料存放區資源的 vSphere 機器	<pre>resources:   demo-vsphere-disk-001:     type: Cloud.vSphere.Disk     properties:       name: DISK_001       type: 'HDD'       capacityGb: 10       dataStore: 'datastore-01'       provisioningType: thick</pre>
具有連結磁碟的 vSphere 機器	<pre>resources:   demo-vsphere-disk-001:     type: Cloud.vSphere.Disk     properties:       name: DISK_001       type: HDD       capacityGb: 10       dataStore: 'datastore-01'       provisioningType: thin   demo-machine:     type: Cloud.vSphere.Machine     properties:       name: demo-machine       cpuCount: 2       totalMemoryMB: 2048       imageRef: &gt;-         https://bintray.com/vmware/photon/ download_file?file_path=2.0%2FRC%2Fova%2Fphoton- custom-hw11-2.0-31bb961.ova       attachedDisks:         - source: '\${demo-vsphere-disk-001.id}'</pre>
連結複製映像中的 vSphere 機器 (附加正斜線和快照名稱)	<pre>resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       imageRef: 'demo-machine/snapshot-01'       cpuCount: 1       totalMemoryMB: 1024</pre>

資源	範例藍圖
vCenter 中位於特定資料夾的 vSphere 機器	<pre> resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       name: demo-machine       cpuCount: 2       totalMemoryMB: 1024       imageRef: ubuntu       resourceGroupName: 'myFolder' </pre>
具有多個 NIC 的 vSphere 機器	<pre> resources:   demo-machine:     type: Cloud.Machine     properties:       image: ubuntu       flavor: small       networks:         - name: '\${network-01.name}'           deviceIndex: 0         - name: '\${network-02.name}'           deviceIndex: 1     network-01:       type: Cloud.vSphere.Network       properties:         name: network-01     network-02:       type: Cloud.vSphere.Network       properties:         name: network-02 </pre>
來自快照映像的 vSphere 機器	<pre> resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       imageRef: 'demo-machine/snapshot-01'       cpuCount: 1       totalMemoryMB: 1024 </pre>
vCenter 中附加了標籤的 vSphere 機器	<pre> resources:   demo-machine:     type: Cloud.Machine     properties:       flavor: small       image: ubuntu       tags:         - key: env           value: demo </pre>
具有自訂規格的 vSphere 機器	<pre> resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       name: demo-machine       image: ubuntu       flavor: small       customizationSpec: Linux </pre>

資源	範例藍圖
具有 vSphere 網路元件和靜態 IP 位址的 vSphere 機器	<pre> resources:   demo-network:     type: Cloud.vSphere.Network     properties:       name: demo-network   demo-machine:     type: Cloud.vSphere.Machine     properties:       image: ubuntu       flavor: small       networks:         - name: demo-network           assignment: static </pre>
具有遠端存取權的 vSphere 機器	<pre> inputs:   username:     type: string     title: Username     description: Username     default: testUser   password:     type: string     title: Password     default: VMware@123     encrypted: true     description: Password for the given username resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       flavor: small       imageRef: &gt;-         https://cloud-images.ubuntu.com/         releases/16.04/release-20170307/ubuntu-16.04-         server-cloudimg-amd64.ova       cloudConfig:           ssh_pwauth: yes         chpasswd:           list:               \${input.username}:\${input.password}           expire: false       users:         - default         - name: \${input.username}           lock_passwd: false           sudo: ['ALL=(ALL) NOPASSWD:ALL']           groups: [wheel, sudo, admin]           shell: '/bin/bash'       runcmd:         - echo "Defaults:\${input.username} ! requiretty" &gt;&gt; /etc/sudoers.d/\${input.username} </pre>

## 記錄的 vRealize Automation Cloud Assembly 藍圖範例

透過包括一組完整的註解，此範例可讓您檢閱 vRealize Automation Cloud Assembly 藍圖中各個區段的結構和用途。

```
# *****
#
# This WordPress blueprint is enhanced with comments to explain its
# parameters.
#
# Try cloning it and experimenting with its YAML code. If you're new to
# YAML, visit yaml.org for general information.
#
# The blueprint deploys a minimum of 3 virtual machines and runs scripts
# to install packages.
#
# *****
#
# -----
# Blueprints need a descriptive name and version if
# source controlled in git.
# -----
name: WordPress Blueprint with Comments
formatVersion: 1
version: 1
#
# -----
# Inputs create user selections that appear at deployment time. Inputs
# can set placement decisions and configurations, and are referenced
# later, by the resources section.
# -----
inputs:
#
# -----
# Choose a cloud endpoint. 'Title' is the visible
# option text (oneOf allows for the friendly title). 'Const' is the
# tag that identifies the endpoint, which was set up earlier, under the
# Cloud Assembly Infrastructure tab.
# -----
platform:
  type: string
  title: Deploy to
  oneOf:
    - title: AWS
      const: aws
    - title: Azure
      const: azure
    - title: vSphere
      const: vsphere
  default: vsphere
#
# -----
# Choose the operating system. Note that the Cloud Assembly
# Infrastructure must also have an AWS, Azure, and vSphere Ubuntu image
```

```

# mapped. In this case, enum sets the option that you see, meaning there's
# no friendly title feature this time. Also, only Ubuntu is available
# here, but having this input stubbed in lets you add more operating
# systems later.
# -----
osimage:
  type: string
  title: Operating System
  description: Which OS to use
  enum:
    - Ubuntu
#
# -----
# Set the number of machines in the database cluster. Small and large
# correspond to 1 or 2 machines, respectively, which you see later,
# down in the resources section.
# -----
dbenvsize:
  type: string
  title: Database cluster size
  enum:
    - Small
    - Large
#
# -----
# Dynamically tag the machines that will be created. The
# 'array' of objects means you can create as many key-value pairs as
# needed. To see how array input looks when it's collected,
# open the blueprint and click TEST.
# -----
Mtags:
  type: array
  title: Tags
  description: Tags to apply to machines
  items:
    type: object
    properties:
      key:
        type: string
        title: Key
      value:
        type: string
        title: Value
#
# -----
# Create machine credentials. These credentials are needed in
# remote access configuration later, in the resources section.
# -----
username:
  type: string
  minLength: 4
  maxLength: 20
  pattern: '[a-z]+'
  title: Database Username
  description: Database Username

```

```

userpassword:
  type: string
  pattern: '[a-z0-9A-Z@#\$]+'
  encrypted: true
  title: Database Password
  description: Database Password
#
# -----
# Set the database storage disk size.
# -----
databaseDiskSize:
  type: number
  default: 4
  maximum: 10
  title: MySQL Data Disk Size
  description: Size of database disk
#
# -----
# Set the number of machines in the web cluster. Small, medium, and large
# correspond to 2, 3, and 4 machines, respectively, which you see later,
# in the WebTier part of the resources section.
# -----
clusterSize:
  type: string
  enum:
    - small
    - medium
    - large
  title: Wordpress Cluster Size
  description: Wordpress Cluster Size
#
# -----
# Set the archive storage disk size.
# -----
archiveDiskSize:
  type: number
  default: 4
  maximum: 10
  title: Wordpress Archive Disk Size
  description: Size of Wordpress archive disk
#
# -----
# The resources section configures the deployment of machines, disks,
# networks, and other objects. In several places, the code pulls from
# the preceding interactive user inputs.
# -----
resources:
#
# -----
# Create the database server. Choose a cloud agnostic machine 'type' so
# that it can deploy to AWS, Azure, or vSphere. Then enter its property
# settings.
# -----
DBTier:
  type: Cloud.Machine

```

```

    properties:
#
# -----
# Descriptive name for the virtual machine. Does not become the hostname
# upon deployment.
# -----
        name: mysql
#
# -----
# Hard-coded operating system image to use. To pull from user input above,
# enter the following instead.
# image: '${input.osimage}'
# -----
        image: Ubuntu
#
# -----
# Hard-coded capacity to use. Note that the Cloud Assembly
# Infrastructure must also have AWS, Azure, and vSphere flavors
# such as small, medium, and large mapped.
# -----
        flavor: small
#
# -----
# Tag the database machine to deploy to the cloud vendor chosen from the
# user input. Tags are case-sensitive, so 'to_lower' forces the tag to
# lowercase to ensure a match with a site's tagging convention. It's
# important if platform input were to contain any upper case characters.
# -----
        constraints:
            - tag: '${"env:" + to_lower(input.platform)}'
#
# -----
# Also tag the database machine with any free-form tags that were created
# during user input.
# -----
        tags: '${input.Mtags}'
#
# -----
# Set the database cluster size by referencing the dbenvsize user
# input. Small is one machine, and large defaults to two.
# -----
        count: '${input.dbenvsize == "Small" ? 1 : 2}'
#
# -----
# Add a variable to connect the machine to a network resource based on
# a property binding to another resource. In this case, it's the
# 'WP_Network' network that gets defined further below.
# -----
        networks:
            - name: '${resource.WP_Network.name}'
              network: '${resource.WP_Network.id}'
#
# -----
# Enable remote access to the database server. Reference the credentials
# from the user input.

```



```

# -----
remoteAccess:
  authentication: usernamePassword
  username: '${input.username}'
  password: '${input.userpassword}'
#
# -----
# You are free to add custom properties, which might be used to initiate
# an extensibility subscription, for example.
# -----
ABC-Company-ID: 9393
#
# -----
# Run OS commands or scripts to further configure the database machine,
# via operations such as setting a hostname, generating SSH private keys,
# or installing packages.
# -----
cloudConfig: |
  #cloud-config
  repo_update: true
  repo_upgrade: all
  packages:
    - mysql-server
  runcmd:
    - sed -e '/bind-address/ s/^#*#/' -i /etc/mysql/mysql.conf.d/mysqld.cnf
    - service mysql restart
    - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
    - mysql -e "FLUSH PRIVILEGES;"
  attachedDisks: []
#
# -----
# Create the web server. Choose a cloud agnostic machine 'type' so that it
# can deploy to AWS, Azure, or vSphere. Then enter its property settings.
# -----
WebTier:
  type: Cloud.Machine
  properties:
#
# -----
# Descriptive name for the virtual machine. Does not become the hostname
# upon deployment.
# -----
name: wordpress
#
# -----
# Hard-coded operating system image to use. To pull from user input above,
# enter the following instead:
# image: '${input.osimage}'
# -----
image: Ubuntu
#
# -----
# Hard-coded capacity to use. Note that the Cloud Assembly
# Infrastructure must also have AWS, Azure, and vSphere flavors
# such as small, medium, and large mapped.

```

```

# -----
#     flavor: small
#
# -----
# Set the web server cluster size by referencing the clusterSize user
# input. Small is 2 machines, medium is 3, and large defaults to 4.
# -----
#     count: '${input.clusterSize== "small" ? 2 : (input.clusterSize == "medium" ? 3 : 4)}'
#
# -----
# Set an environment variable to display object information under the
# Properties tab, post-deployment. Another example might be
# {env.blueprintID}
# -----
#     tags:
#       - key: cas.requestedBy
#         value: '${env.requestedBy}'
#
# -----
# You are free to add custom properties, which might be used to initiate
# an extensiblity subscription, for example.
# -----
#     ABC-Company-ID: 9393
#
# -----
# Tag the web server to deploy to the cloud vendor chosen from the
# user input. Tags are case-sensitive, so 'to_lower' forces the tag to
# lowercase to ensure a match with your site's tagging convention. It's
# important if platform input were to contain any upper case characters.
# -----
#     constraints:
#       - tag: '${"env:" + to_lower(input.platform)}'
#
# -----
# Add a variable to connect the machine to a network resource based on
# a property binding to another resource. In this case, it's the
# 'WP_Network' network that gets defined further below.
# -----
#     networks:
#       - name: '${resource.WP_Network.name}'
#         network: '${resource.WP_Network.id}'
#
# -----
# Run OS commands or scripts to further configure the web server,
# with operations such as setting a hostname, generating SSH private keys,
# or installing packages.
# -----
#     cloudConfig: |
#       #cloud-config
#       repo_update: true
#       repo_upgrade: all
#       packages:
#         - apache2
#         - php
#         - php-mysql

```

```

- libapache2-mod-php
- php-mcrypt
- mysql-client
runcmd:
- mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://
wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
mywordpresssite --strip-components 1
- i=0; while [ $i -le 5 ]; do mysql --connect-timeout=3 -h $
{DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;
i=$((i+1)); done
- mysql -u root -pmysqlpassword -h ${resource.DBTier.networks[0].address} -e
"create database wordpress_blog;"
- mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
mywordpresssite/wp-config.php
- sed -i -e s/"define('DB_NAME', 'database_name_here');"/"define('DB_NAME',
'wordpress_blog');"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e
s/"define('DB_USER', 'username_here');"/"define('DB_USER', 'root');"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define('DB_PASSWORD',
'password_here');"/"define('DB_PASSWORD', 'mysqlpassword');"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define('DB_HOST',
'localhost');"/"define('DB_HOST', '${resource.DBTier.networks[0].address}');"/ /var/www/html/
mywordpresssite/wp-config.php
- service apache2 reload
#
# -----
# Create the network that the database and web servers connect to.
# Choose a cloud agnostic network 'type' so that it can deploy to AWS,
# Azure, or vSphere. Then enter its property settings.
# -----
WP_Network:
  type: Cloud.Network
  properties:
#
# -----
# Descriptive name for the network. Does not become the network name
# upon deployment.
# -----
name: WP_Network
#
# -----
# Set the networkType to an existing network. You could also use a
# constraint tag to target a specific, like-tagged network.
# The other network types are private or public.
# -----
networkType: existing
#
# *****
#
# VMware hopes that you found this commented blueprint useful. Note that
# you can also access an API to create blueprints, or query for input
# schema that you intend to request. See the following Swagger
# documentation.

```

```
#
# www.mgmt.cloud.vmware.com/blueprint/api/swagger/swagger-ui.html
#
# *****
```

## vRealize Automation Cloud Assembly 藍圖中的網路、安全性和負載平衡器範例

這些藍圖程式碼範例說明了一些基本網路、安全性和負載平衡器組態。

如需詳細的網路和安全性實作案例的範例，請參閱 VMware 部落格，例如：

- [vRealize Automation Cloud Assembly 負載平衡器與 NSX-T 搭配使用深入探究](#)
- [使用 Cloud Assembly 和 NSX 實現網路自動化 – 第 1 部分](#) (包括使用 NSX-T 和 vCenter 雲端帳戶與網路 CIDR)
- [使用 Cloud Assembly 和 NSX 實現網路自動化 – 第 2 部分](#) (包括使用現有和輸出網路類型)
- [使用 Cloud Assembly 和 NSX 實現網路自動化 – 第 3 部分](#) (包括使用現有和隨選安全群組)
- [使用 Cloud Assembly 和 NSX 實現網路自動化 – 第 4 部分](#) (包括使用現有和隨選負載平衡器)

如需所有藍圖架構選項的完整摘要，請參閱〈[vRealize Automation 資源類型架構](#)〉。

如需有關網路類型的詳細資訊，請參閱在 [vRealize Automation Cloud Assembly 中使用網路和網路設定檔](#)。

資源案例	範例藍圖程式碼
具有多個 NIC 的 vSphere 機器	<pre>resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       image: ubuntu       flavor: small       networks:         - name: '\${network-01.id}'           deviceIndex: 0         - name: '\${network-02.id}'           deviceIndex: 1   Cloud_vSphere_Network_1:     type: Cloud.vSphere.Network     properties:       networkType: existing       name: network-01   Cloud_vSphere_Network_2:     type: Cloud.NSX.Network     properties:       networkType: existing       name: network-02</pre>
使用內部 IP 而非公用 IP 的公有雲機器	<pre>resources:   wf_proxy:     type: Cloud.Machine     properties:       image: ubuntu 16.04       flavor: small       constraints:         - tag: 'platform:vsphere'       networks:         - name: '\${resource.wf_net.id}'           assignPublicIpAddress: false</pre>
使用 NSX 網路元件類型的 NSX-V 或 NSX-T 的路由網路	<pre>Cloud_NSX_Network_1:   type: Cloud.NSX.Network   properties:     networkType: routed</pre>
為輸出網路標記 NSX 邏輯交換器 如需有關此案例的詳細資訊，請參閱社群部落格文章 <a href="#">使用 Cloud Assembly 在 NSX 中建立標籤</a> 。	<pre>Cloud_NSX_Network_1:   type: Cloud.NSX.Network   properties:     networkType: outbound     tags:       - key: app         value: opencart</pre>

## 資源案例

## 範例藍圖程式碼

已將限制標籤套用至機器 NIC 的現有安全群組

若要使用現有的安全群組，請為 Cloud.SecurityGroup 元件中的 securityGroupType 內容輸入 *existing*。若要建立隨選安全群組，請為

Cloud.SecurityGroup 元件中的 securityGroupType 內容輸入 *new*。

您可以使用標籤限制將標籤指派給 Cloud.SecurityGroup 元件，以配置現有的安全群組。不含標籤的安全群組無法在藍圖中使用。

```
formatVersion: 1
inputs: {}
resources:
  allowSsh_sg:
    type: Cloud.SecurityGroup
    properties:
      securityGroupType : existing
      constraints:
        - tag: allowSsh
  compute:
    type: Cloud.Machine
    properties:
      image: centos
      flavor: small
      networks:
        - network: '${resource.prod-net.id}'
          securityGroups:
            - '${resource.allowSsh_sg.id}'
  prod-net:
    type: Cloud.Network
    properties:
      networkType: existing
```

**資源案例****範例藍圖程式碼**

具有單肩負載平衡器的隨選網路

```

inputs: {}
resources:
  mp-existing:
    type: Cloud.Network
    properties:
      name: mp-existing
      networkType: existing
  mp-wordpress:
    type: Cloud.vSphere.Machine
    properties:
      name: wordpress
      count: 2
      flavor: small
      image: tiny
      customizationSpec: Linux
      networks:
        - network: '${resource["mp-private"].id}'
  mp-private:
    type: Cloud.NSX.Network
    properties:
      name: mp-private
      networkType: private
      constraints:
        - tag: nsxt
  mp-wordpress-lb:
    type: Cloud.LoadBalancer
    properties:
      name: wordpress-lb
      internetFacing: false
      network: '${resource.mp-existing.id}'
      instances: '${resource["mp-wordpress"].id}'
      routes:
        - protocol: HTTP
          port: '80'
          instanceProtocol: HTTP
          instancePort: '80'
          healthCheckConfiguration:
            protocol: HTTP
            port: '80'
            urlPath: /index.pl
            intervalSeconds: 60
            timeoutSeconds: 30
            unhealthyThreshold: 5
            healthyThreshold: 2

```

具有負載平衡器的現有網路

```

formatVersion: 1
inputs:
  count:
    type: integer
    default: 1
resources:
  ubuntu-vm:
    type: Cloud.Machine
    properties:
      name: ubuntu
      flavor: small
      image: tiny
      count: '${input.count}'
      networks:

```

## 資源案例

## 範例藍圖程式碼

```

- network: '$
{resource.Cloud_NSX_Network_1.id}'
Provider_LoadBalancer_1:
  type: Cloud.LoadBalancer
  properties:
    name: OC-LB
    routes:
      - protocol: HTTP
        port: '80'
        instanceProtocol: HTTP
        instancePort: '80'
        healthCheckConfiguration:
          protocol: HTTP
          port: '80'
          urlPath: /index.html
          intervalSeconds: 60
          timeoutSeconds: 5
          unhealthyThreshold: 5
          healthyThreshold: 2
        network: '$
{resource.Cloud_NSX_Network_1.id}'
        internetFacing: false
        instances: '${resource["ubuntu-vm"].id}'
Cloud_NSX_Network_1:
  type: Cloud.NSX.Network
  properties:
    networkType: existing
  constraints:
    - tag: nsxt24prod

```

## 啟用 Puppet 的具有使用者名稱和密碼存取權的藍圖

在此範例中，將 Puppet 組態管理新增至在具有使用者名稱和密碼存取權的 vCenter 計算資源上部署的藍圖。

此程序顯示您可以如何建立需要使用者名稱和密碼驗證的啟用 Puppet 的可部署資源的範例。使用者名稱和密碼存取權表示，使用者必須從計算資源手動登入 Puppet 主要機器，才能叫用 Puppet 組態管理。

您可以選擇性地設定可在藍圖中設定組態管理的遠端存取驗證，以便計算資源可使用 Puppet 主要機器處理驗證。啟用遠端存取後，計算資源會自動產生金鑰，以滿足密碼驗證。仍然需要有效的使用者名稱。

請參閱 [AWS Puppet 組態管理藍圖範例](#)和 [vCenter Puppet 組態藍圖範例](#)，以取得可以在 vRealize Automation Cloud Assembly 藍圖中如何設定不同 Puppet 案例的更多範例。

### 必要條件

- 在有效的網路上設定 Puppet Enterprise 執行個體。
- 使用整合功能將 Puppet Enterprise 執行個體新增至 vRealize Automation Cloud Assembly。請參閱在 [vRealize Automation Cloud Assembly 中設定 Puppet Enterprise 整合](#)
- 設定 vSphere 帳戶和 vCenter 計算資源。



## 程序

1 將 Puppet 組態管理元件新增至所需藍圖的畫布上的 vSphere 計算資源。

- 選取**基礎結構 > 管理 > 整合**。
- 按一下**新增整合**，然後選取 Puppet。
- 在 [Puppet 組態] 頁面上，輸入適當的資訊。

組態	說明	範例值
主機名稱	Puppet 主要機器的主機名稱或 IP 位址	Puppet-Ubuntu
SSH 連接埠	用於 vRealize Automation Cloud Assembly 和 Puppet 主要機器之間通訊的 SSH 連接埠。(選用)	NA
自動簽署密碼	在 Puppet 主要機器上設定的共用密碼，節點應提供以支援自動簽署憑證要求。	使用者特定
位置	指示 Puppet 主要機器是在私有雲還是公有雲上。  <b>備註</b> 僅在部署計算資源與 Puppet 主要機器之間存在連線時，才支援跨雲端部署。	
Cloud proxy	公有雲帳戶不需要，例如 Microsoft Azure 或 Amazon Web Services。如果您正使用以 vCenter 為基礎的雲端帳戶，請為您的帳戶選取適當的 cloud proxy。	NA
使用者名稱	Puppet 主要機器的 SSH 和 RBAC 使用者名稱。	使用者特定。YAML 值為 '{input.username}'
密碼	Puppet 主要機器的 SSH 和 RBAC 密碼。	使用者特定的 YAML 值為 '{input.password}'
為此使用者使用 sudo 命令	選取以將 sudo 命令用於 procidd。	true
名稱	Puppet 主要機器名稱。	PEMasterOnPrem
說明		

2 將使用者名稱和密碼內容新增至 Puppet YAML，如以下範例所示。

3 確保將 Puppet YAML 中 remoteAccess 內容的值設為 authentication: username and password，如以下範例所示。

### 範例：vCenter 使用者名稱和密碼 YAML 程式碼

以下範例顯示用於在 vCenter 計算資源上新增使用者名稱和密碼驗證的代表性的 YAML 程式碼。

```
inputs:
  username:
    type: string
    title: Username
    description: Username to use to install Puppet agent
```

```

    default: puppet
  password:
    type: string
    title: Password
    default: VMware@123
    encrypted: true
    description: Password for the given username to install Puppet agent
  resources:
    Puppet-Ubuntu:
      type: Cloud.vSphere.Machine
      properties:
        flavor: small
        imageRef: >-
          https://cloud-images.ubuntu.com/releases/16.04/release-20170307/ubuntu-16.04-server-
cloudimg-amd64.ova
        remoteAccess:
          authentication: usernamePassword
          username: '${input.username}'
          password: '${input.password}'
    Puppet_Agent:
      type: Cloud.Puppet
      properties:
        provider: PEMasterOnPrem
        environment: production
        role: 'role::linux_webserver'
        username: '${input.username}'
        password: '${input.password}'
        host: '${Puppet-Ubuntu.*}'
        useSudo: true
        agentConfiguration:
          certName: '${Puppet-Ubuntu.address}'

```

## AWS Puppet 組態管理藍圖範例

有多個選項可用於設定藍圖，以支援 AWS 計算資源上以 Puppet 為基礎的組態管理。

## 具有使用者名稱和密碼的 AWS 上的 Puppet 管理

範例...	範例藍圖 YAML
在任何支援的 Amazon Machine Image 上進行雲端組態驗證。	<pre> inputs:   username:     type: string     title: Username     default: puppet   password:     type: string     title: Password     encrypted: true     default: VMware@123 resources:   Webserver:     type: Cloud.AWS.EC2.Instance     properties:       flavor: small       image: centos       cloudConfig:           #cloud-config         ssh_pwauth: yes         chpasswd:           list:               \${input.username}:\${input.password}           expire: false         users:           - default           - name: \${input.username}             lock_passwd: false             sudo: ['ALL=(ALL) NOPASSWD:ALL']             groups: [wheel, sudo, admin]             shell: '/bin/bash'             ssh-authorized-keys:               - ssh-rsa                 AAAAB3NzaC1yc2EAAAADAQABAAQDytVL+Q6/+vGbmKXoRpX                 dmettem@dmettem-m01.vmware.com           runcmd:             - echo "Defaults:\${input.username} !requiretty"             &gt;&gt; /etc/sudoers.d/\${input.username}   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEOonAWS       environment: production       role: 'role::linux_webserver'       host: '\${Webserver.*}'       osType: linux       username: '\${input.username}'       password: '\${input.password}'       useSudo: true </pre>
在自訂 Amazon Machine Image 上對現有使用者進行雲端組態驗證。	<pre> inputs:   username:     type: string     title: Username     default: puppet   password:     type: string     title: Password     encrypted: true     default: VMware@123 </pre>

範例...	範例藍圖 YAML
	<pre> resources:   Webserver:     type: Cloud.AWS.EC2.Instance     properties:       flavor: small       image: centos       cloudConfig:           #cloud-config       runcmd:         - sudo sed -e 's/.*PasswordAuthentication no.*/ PasswordAuthentication yes/' -i /etc/ssh/sshd_config         - sudo service sshd restart   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEOAWS       environment: production       role: 'role::linux_webserver'       host: '\${Webserver.*}'       osType: linux       username: '\${input.username}'       password: '\${input.password}'       useSudo: true </pre>

## 具有已產生 PublicPrivateKey 的 AWS 上的 Puppet 管理

範例...	範例藍圖 YAML
具有 generatedPublicPrivateKey access 的 AWS 上的 remoteAccess.authentication 驗 證。	<pre> inputs: {} resources:   Machine:     type: Cloud.AWS.EC2.Instance     properties:       flavor: small       imageRef: ami-a4dc46db       remoteAccess:         authentication: generatedPublicPrivateKey   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: puppet-BlueprintProvisioningITSuite       environment: production       role: 'role::linux_webserver'       host: '\${Machine.*}'       osType: linux       username: ubuntu       useSudo: true       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'       useSudo: true </pre>

## vCenter Puppet 組態藍圖範例

有多種選項用於設定藍圖，以支援在 vCenter 計算資源上以 Puppet 為基礎的組態管理。

## vSphere 上的 Puppet，具有使用者名稱和密碼驗證

下列範例顯示了具有使用者名稱和密碼驗證的 vSphere OVA 上 Puppet 的 YAML 程式碼範例。

表 6-2.

範例...	範例藍圖 YAML
<p>具有使用者名稱和密碼驗證的 vSphere OVA 上 Puppet 的 YAML 程式碼。</p>	<pre>inputs:   username:     type: string     title: Username     default: puppet   password:     type: string     title: Password     encrypted: true     default: VMware@123 resources:   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEonAWS       environment: dev       role: 'role::linux_webserver'       username: '\${input.username}'       password: '\${input.password}'       useSudo: true       host: '\${Webserver.*}'       osType: linux       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'   Webserver:     type: Cloud.vSphere.Machine     properties:       cpuCount: 1       totalMemoryMB: 1024       imageRef: &gt;- https://cloud-images.ubuntu.com/releases/16.04/ release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       cloudConfig:           #cloud-config         ssh_pwauth: yes         chpasswd:           list:               \${input.username}:\${input.password}           expire: false         users:           - default           - name: \${input.username}             lock_passwd: false             sudo: ['ALL=(ALL) NOPASSWD:ALL']             groups: [wheel, sudo, admin]             shell: '/bin/bash'             ssh-authorized-keys:               - ssh-rsa                 AAAAB3NzaC1yc2EAAAADAQABAAQDytVL+Q6+vGbmKXoRpX                 dmettem@dmettem-m01.vmware.com             runcmd:               - echo "Defaults:\${input.username}"</pre>
<p>vSphere OVA 上 Puppet 的 YAML 程式碼，在計算資源上具有使用者名稱和密碼驗證。</p>	<pre>inputs:   username:     type: string     title: Username     default: puppet</pre>

表 6-2. (續)

範例...	範例藍圖 YAML
	<pre> password:   type: string   title: Password   encrypted: true   default: VMware@123 resources:   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEonAWS       environment: dev       role: 'role::linux_webserver'       username: '\${input.username}'       password: '\${input.password}'       useSudo: true       host: '\${Webserver.*}'       osType: linux       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'   Webserver:     type: Cloud.vSphere.Machine     properties:       cpuCount: 1       totalMemoryMB: 1024       imageRef: &gt;- https://cloud-images.ubuntu.com/releases/16.04/ release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       cloudConfig:           #cloud-config         ssh_pwauth: yes         chpasswd:           list:               \${input.username}:\${input.password}           expire: false         users:           - default           - name: \${input.username}             lock_passwd: false             sudo: ['ALL=(ALL) NOPASSWD:ALL']             groups: [wheel, sudo, admin]             shell: '/bin/bash'             ssh-authorized-keys:               - ssh-rsa                 AAAAB3NzaC1yc2EAAAADAQABAAQDytVL+Q6+vGbmKXoRpX                 dmettem@dmettem-m01.vmware.com       runcmd:         - echo "Defaults:\${input.username} </pre>
<p>vCenter 上 Puppet 的 YAML 程式碼， 在計算資源上具有已啟用遠端存取的密碼 驗證。</p>	<pre> inputs:   username:     type: string     title: Username     description: Username to use to install Puppet agent     default: puppet   password:     type: string     title: Password     default: VMware@123     encrypted: true </pre>

表 6-2. (續)

範例...	範例藍圖 YAML
	<pre> description: Password for the given username to install Puppet agent resources:   Puppet-Ubuntu:     type: Cloud.vSphere.Machine     properties:       flavor: small       imageRef: &gt;-         https://cloud-images.ubuntu.com/releases/16.04/         release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       remoteAccess:         authentication: usernamePassword         username: '\${input.username}'         password: '\${input.password}'   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEMasterOnPrem       environment: production       role: 'role::linux_webserver'       username: '\${input.username}'       password: '\${input.password}'       host: '\${Puppet-Ubuntu.*}'       useSudo: true       agentConfiguration:         certName: '\${Puppet-Ubuntu.address}' </pre>

## vSphere 上的 Puppet，具有產生的 PublicPrivateKey 驗證

表 6-3.

範例...	範例藍圖 YAML
vSphere OVA 上 Puppet 的 YAML 代碼，在計算資源上具有產生的 PublicPrivateKey 驗證。	<pre> inputs: {} resources:   Machine:     type: Cloud.vSphere.Machine     properties:       flavor: small       imageRef: &gt;-         https://cloud-images.ubuntu.com/releases/16.04/         release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       remoteAccess:         authentication: generatedPublicPrivateKey   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: puppet-BlueprintProvisioningITSuite       environment: production       role: 'role::linux_webserver'       host: '\${Machine.*}'       osType: linux       username: ubuntu       useSudo: true       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'         - echo "Defaults:\${input.username}" </pre>



## 如何使用 vRealize Automation Cloud Assembly 市集

若要快速啟動資源庫，請從 vRealize Automation Cloud Assembly 市集下載檔案。

市集提供了在 [VMware Solution Exchange](#) 上管理的已完成藍圖和開放虛擬化映像。標記有 `cloud assembly` 的 Solution Exchange 檔案會顯示在 vRealize Automation Cloud Assembly [市集] 索引標籤下。

### 如何存取市集

在 vRealize Automation Cloud Assembly 中，選取**基礎結構 > 連線 > 整合**。按一下**新增整合**，按一下**My VMware**，並提供您的 My VMware 帳戶認證。

### 如何下載及使用市集藍圖檔案

在**市集**索引標籤中，按一下**取得**，並接受藍圖使用者授權合約。然後，您可以將藍圖新增至 vRealize Automation Cloud Assembly 專案，或僅下載它。您可以在**藍圖**索引標籤中上傳藍圖。

對於基於專案的範例，假設您是巨量資料工作的專案管理員。為了協助您的團隊，請找到新增至團隊專案的市集 Hadoop 藍圖。然後，針對您的資源環境自訂藍圖並將其發佈。之後，將藍圖匯入 vRealize Automation Service Broker 目錄，以便您的團隊可以進行部署。

### 如何下載及使用市集映像檔

在**市集**索引標籤中，按一下**取得**，並接受 OVF 或 OVA 映像使用者授權合約。之後，您可以下載 OVF 或 OVA 映像，並在藍圖代碼中參考該映像。

繼續上一個範例，您的團隊可能需要存取 Hadoop 本身的版本。您可以下載 Hadoop OVF，並將其新增到雲端帳戶資源，例如 vCenter Server 內容程式庫。然後，更新任何需要指向 OVF 映像的藍圖代碼。

## 如何使用擴充性延長和自動化應用程式生命週期

您可以將擴充性動作或 vRealize Orchestrator 工作流程與擴充性訂閱搭配使用，來延長應用程式生命週期。

藉由 vRealize Automation Cloud Assembly 擴充性，您可以使用訂閱為事件指派擴充性動作或 vRealize Orchestrator 工作流程。發生指定的事件時，訂閱會啟動動作或工作流程以使其執行，並通知所有訂閱者。

### 擴充性動作

擴充性動作是小型的輕量型程式碼指令碼，用於指定動作和如何執行該動作。可以直接從預先定義的 vRealize Automation Cloud Assembly 動作範本匯入擴充性動作，也可以從 ZIP 檔案匯入。您也可以使用動作編輯器建立擴充性動作的自訂指令碼。在一個指令碼中將多個動作指令碼連結在一起時，即建立了一個動作流程。使用動作流程可以建立一系列動作。如需使用動作流程的相關資訊，請參閱[什麼是動作流程](#)。

## vRealize Orchestrator 工作流程

透過將 vRealize Automation Cloud Assembly 與現有的 vRealize Orchestrator 環境整合，您可以在擴充性訂閱中使用工作流程。

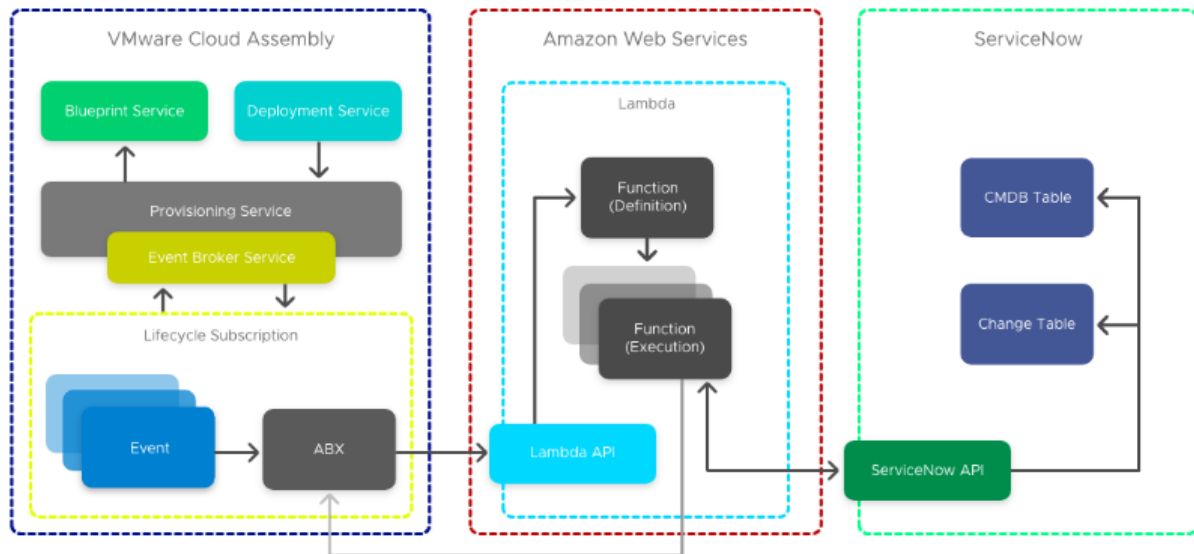
### 擴充性動作訂閱

您可以為 vRealize Automation Cloud Assembly 訂閱指派擴充性動作，以延長應用程式生命週期。

**備註** 下列訂閱只是一些使用案例範例，未涵蓋所有擴充性動作功能。

### 如何使用擴充性動作將 Cloud Assembly 與 ServiceNow 整合

使用擴充性動作，可以將 vRealize Automation Cloud Assembly 與企業 ITSM (例如 ServiceNow) 整合。

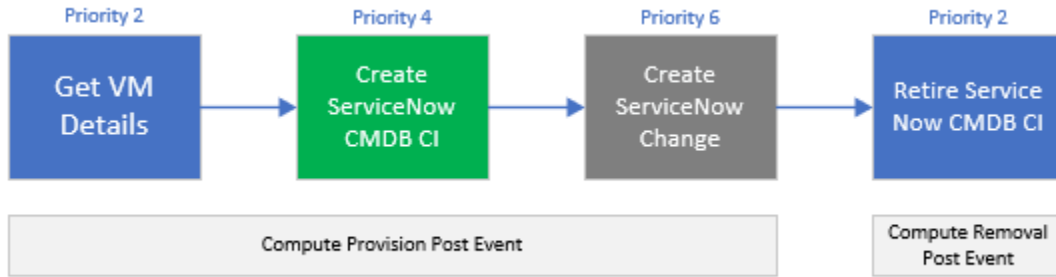


企業使用者通常會將其雲端管理平台與 IT 服務管理 (ITSM) 和設定管理資料庫 (CMDB) 平台整合以實現符合性。按照此範例，您可以使用擴充性動作指令碼將 vRealize Automation Cloud Assembly 與適用於 CMDB 和 ITSM 的 ServiceNow 整合。

**備註** 您也可以使用 vRealize Orchestrator 工作流程將 ServiceNow 與 vRealize Automation Cloud Assembly 整合。如需有關使用工作流程整合 ServiceNow 的資訊，請參閱[如何使用 vRealize Orchestrator 工作流程將適用於 ITSM 的 Cloud Assembly 與 ServiceNow 整合](#)。

若要建立此整合，請使用四個擴充性動作指令碼。在佈建期間，前三個指令碼在發生計算資源佈建後事件時按順序起始。第四個指令碼在發生計算資源移除後事件時觸發。

如需有關事件主題的詳細資訊，請參閱[vRealize Automation Cloud Assembly 隨附的事件主題](#)。



## 取得虛擬機器詳細資料

「取得虛擬機器詳細資料」指令碼可取得建立 CI 所需的其他裝載詳細資料，以及儲存在 Amazon Web Services Systems Manager 參數存放區 (SSM) 中的身分識別 Token。此外，此指令碼還可以使用其他內容更新 `customProperties` 以供日後使用。

## 建立 ServiceNow CMDB CI

「建立 ServiceNow CMDB CI」指令碼將 ServiceNow 執行個體 URL 做為輸入傳遞，並將執行個體儲存在 SSM 中以滿足安全性需求。此指令碼還會讀取 ServiceNow CMDB 唯一記錄識別碼回應 (`sys_id`)。此指令碼將回應做為輸出傳遞，並在建立期間寫入自訂內容 `serviceNowSysId`。此值用於在銷毀執行個體時將 CI 標記為已淘汰。

**備註** 可能需要向 vRealize Automation services Amazon Web Services 角色配置其他權限，以允許 Lambda 存取 SSM 參數存放區。

## 建立 ServiceNow 變更

此指令碼將 ServiceNow 執行個體 URL 做為輸入傳遞，並將 ServiceNow 認證儲存在 SSM 中以滿足安全性需求，從而完成 ITSM 整合。

## 建立 ServiceNow 變更

「淘汰 ServiceNow CMDB CI」指令碼會根據在建立指令碼中建立的自訂內容 `serviceNowSysId`，來提示 ServiceNow 停止 CI 並將其標記為已淘汰。

### 必要條件

- 設定此整合之前，篩選具有條件藍圖內容的所有事件訂閱：  
`event.data["customProperties"]["enable_servicenow"] === "true"`

**備註** 此內容存在於需要 ServiceNow 整合的藍圖上。

- 已安裝 Python 應用程式。

如需有關篩選訂閱的詳細資訊，請參閱[建立擴充性訂閱](#)。

### 程序

- 1 從虛擬機器開啟命令列提示字元。

## 2 執行「取得虛擬機器詳細資料」指令碼。

```

from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):
    baseUri = inputs['url']
    casToken = client.get_parameter(Name="casToken",WithDecryption=True)

    url = baseUri + "/iaas/login"
    headers = {"Accept":"application/json","Content-Type":"application/json"}
    payload = {"refreshToken":casToken['Parameter']['Value']}

    results = requests.post(url,json=payload,headers=headers)

    bearer = "Bearer "
    bearer = bearer + results.json()["token"]

    deploymentId = inputs['deploymentId']
    resourceId = inputs['resourceIds'][0]

    print("deploymentId: "+ deploymentId)
    print("resourceId:" + resourceId)

    machineUri = baseUri + "/iaas/machines/" + resourceId
    headers = {"Accept":"application/json","Content-Type":"application/json",
"Authorization":bearer }
    resultMachine = requests.get(machineUri,headers=headers)
    print("machine: " + resultMachine.text)

    print( "serviceNowCPUCount: "+ json.loads(resultMachine.text)["customProperties"]
["cpuCount"] )
    print( "serviceNowMemoryInMB: "+ json.loads(resultMachine.text)["customProperties"]
["memoryInMB"] )

    #update customProperties
    outputs = {}
    outputs['customProperties'] = inputs['customProperties']
    outputs['customProperties']['serviceNowCPUCount'] = int(json.loads(resultMachine.text)
["customProperties"]["cpuCount"])
    outputs['customProperties']['serviceNowMemoryInMB'] = json.loads(resultMachine.text)
["customProperties"]["memoryInMB"]
    return outputs

```

## 3 執行 CMDB 組態項目建立動作。

```

from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):

```

```

snowUser = client.get_parameter(Name="serviceNowUserName",WithDecryption=False)
snowPass = client.get_parameter(Name="serviceNowPassword",WithDecryption=True)
table_name = "cmdb_ci_vmware_instance"
url = "https://" + inputs['instanceUrl'] + "/api/now/table/{0}".format(table_name)
headers = {'Content-type': 'application/json', 'Accept': 'application/json'}
payload = {
    'name': inputs['customProperties']['serviceNowHostname'],
    'cpus': int(inputs['customProperties']['serviceNowCPUCount']),
    'memory': inputs['customProperties']['serviceNowMemoryInMB'],
    'correlation_id': inputs['deploymentId'],
    'disks_size': int(inputs['customProperties']['provisionGB']),
    'location': "Sydney",
    'vcenter_uuid': inputs['customProperties']['vcUuid'],
    'state': 'On',
    'sys_created_by': inputs['__metadata']['userName'],
    'owned_by': inputs['__metadata']['userName']
}
results = requests.post(
    url,
    json=payload,
    headers=headers,
    auth=(snowUser['Parameter']['Value'], snowPass['Parameter']['Value'])
)
print(results.text)

#parse response for the sys_id of CMDB CI reference
if json.loads(results.text)['result']:
    serviceNowResponse = json.loads(results.text)['result']
    serviceNowSysId = serviceNowResponse['sys_id']
    print(serviceNowSysId)

#update the serviceNowSysId customProperty
outputs = {}
outputs['customProperties'] = inputs['customProperties']
outputs['customProperties']['serviceNowSysId'] = serviceNowSysId;
return outputs

```

#### 4 執行建立動作指令碼。

```

from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):
    snowUser = client.get_parameter(Name="serviceNowUserName",WithDecryption=False)
    snowPass = client.get_parameter(Name="serviceNowPassword",WithDecryption=True)
    table_name = "change_request"
    url = "https://" + inputs['instanceUrl'] + "/api/now/table/{0}".format(table_name)
    headers = {'Content-type': 'application/json', 'Accept': 'application/json'}
    payload = {
        'short_description': 'Provision CAS VM Instance'
    }
    results = requests.post(

```

```

    url,
    json=payload,
    headers=headers,
    auth=(snowUser['Parameter']['Value'], snowPass['Parameter']['Value'])
)
print(results.text)

```

## 結果

vRealize Automation Cloud Assembly 已成功與 ITSM ServiceNow 整合。

## 後續步驟

如果需要，您可以使用 CMDB 組態淘汰動作將 CI 淘汰：

```

from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):
    snowUser = client.get_parameter(Name="serviceNowUserName",WithDecryption=False)
    snowPass = client.get_parameter(Name="serviceNowPassword",WithDecryption=True)
    tableName = "cmdb_ci_vmware_instance"
    sys_id =inputs['customProperties']['serviceNowSysId']
    url = "https://" + inputs['instanceUrl'] + "/api/now/"+tableName+"/"+{0}".format(sys_id)
    headers = {'Content-type': 'application/json', 'Accept': 'application/json'}
    payload = {
        'state': 'Retired'
    }

    results = requests.put(
        url,
        json=payload,
        headers=headers,
        auth=(inputs['username'], inputs['password'])
    )
    print(results.text)

```

如需有關如何使用擴充性動作在 vRealize Automation Cloud Assembly 中整合 ServiceNow 的詳細資訊，請參閱 [〈為 ServiceNow 整合使用基於動作的擴充性延伸 Cloud Assembly〉](#)。

## 如何在佈建期間使用擴充性動作標記虛擬機器

您可以將擴充性動作與訂閱搭配使用，以自動化和簡化標記虛擬機器操作。

做為雲端管理員，您可以使用擴充性動作和擴充性訂閱來建立自動以指定的輸入和輸出進行標記的部署。針對包含「標記虛擬機器」訂閱的專案建立新部署時，部署事件會觸發執行「標記虛擬機器」指令碼，並會自動套用標籤。這樣可以節省時間並提高效率，同時也可以簡化部署管理。

### 必要條件

- 對雲端管理員認證的存取權。

- 用於 Lambda 函數的 Amazon Web Services 角色。

## 程序

- 1 導覽至**擴充性 > 程式庫 > 動作 > 新增動作**，然後使用下列參數建立動作。

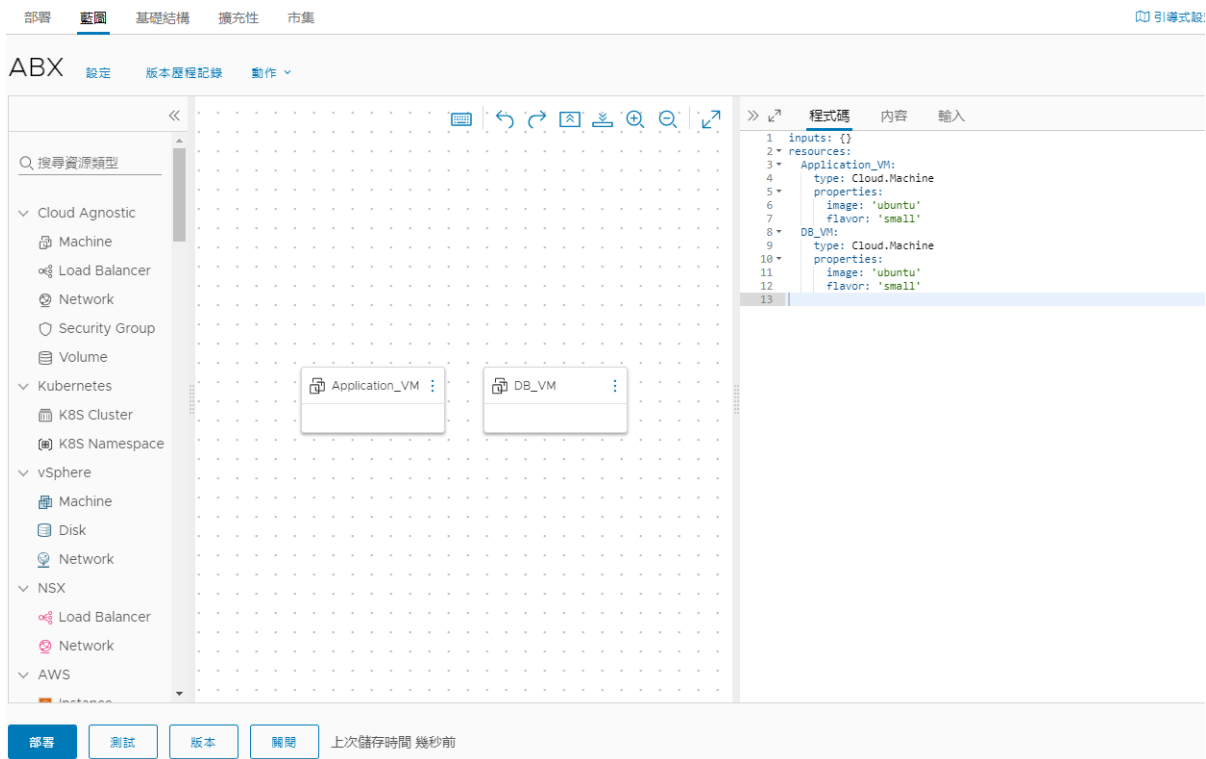
參數	說明
動作名稱	擴充性動作名稱，最好含「TagVM」做為前置詞或後置詞。
專案	用於測試擴充性動作的專案。
動作範本	標記虛擬機器
執行階段	Python
指令碼來源	寫入指令碼

- 2 輸入 **Handler** 做為**主要函數**。
- 3 新增標記輸入，用於測試擴充性動作。  
例如，resourceNames = ["DB\_VM"]，target = world。
- 4 若要儲存動作，請按一下**儲存**。
- 5 若要驗證動作，請按一下**測試**。
- 6 若要結束動作編輯器，請按一下**關閉**。
- 7 導覽至**擴充性 > 訂閱**。
- 8 按一下**新增訂閱**。
- 9 輸入下列訂閱詳細資料。

詳細資料	設定
事件主題	選取虛擬機器的標記階段的相關事件主題。例如，計算配置。 <b>備註</b> 標籤必須是訂閱架構的一部分。
封鎖	將訂閱逾時設定為 1 分鐘。
可執行的項目類型	選取擴充性動作可執行的類型。
可執行的識別碼	選取您的自訂擴充性動作。

- 10 若要儲存自訂擴充性動作訂閱，請按一下**建立**。

## 11 建立包含兩個虛擬機器的新藍圖：應用程式虛擬機器和資料庫虛擬機器。



12 若要部署虛擬機器，請按一下部署。

13 在部署期間，確認會起始事件且執行擴充性動作。

14 若要確認標籤是否正確套用，請導覽至基礎結構 > 資源 > 機器。

## 進一步瞭解擴充性動作

以動作為基礎的擴充性使用 vRealize Automation Cloud Assembly 中簡化的程式碼指令碼來自動執行擴充性動作。

以動作為基礎的擴充性提供一個輕量型且彈性的執行階段引擎介面，您可在其中定義小型可編寫指令碼動作以及設定它們以在事件代理服務 (EBS) 提供的特定事件發生時起始。

您可以在 vRealize Automation Cloud Assembly 中建立程式碼的這些擴充性動作指令碼，並將其指派給訂閱。與工作流程類似，當擴充性訂閱中包括的事件發生時，擴充性動作指令碼會觸發。這些擴充性動作指令碼用於更輕量且簡單的工作和步驟自動化。相對於使用 vRealize Orchestrator 伺服器在內部部署主控的工作流程，它們也主控於雲端。如需有關整合 vRealize Automation Cloud Assembly 與 vRealize Orchestrator 伺服器的詳細資訊，請參閱在 [Cloud Assembly 中設定 vRealize Orchestrator 整合](#)。

以動作為基礎的擴充性具有下列優點：

- vRealize Orchestrator 工作流程的一種替代方案，針對輕量型整合和自訂，使用小型且可重複使用的可編寫指令碼動作。
- 一種重複使用動作範本的方式，包含可重複使用的參數化動作。



透過撰寫使用者定義的動作指令碼程式碼或匯入預先定義的指令碼程式碼做為 .ZIP 套件，您可以建立擴充性動作。以動作為基礎的擴充性支援 Node.js 和 Python 執行階段環境，並依賴於 Amazon Web Services Lambda。因此，您必須具有包含 Amazon Web Services 身分識別與存取管理 (IAM) 的有效訂閱，並將 Amazon Web Services 設為 vRealize Automation Cloud Assembly 中的端點。如需 Amazon Web Services Lambda 入門的相關資訊，請參閱 [ABX : Cloud Assembly 服務的無伺服器擴充性](#)。

---

**備註** 擴充性動作是專案特定的。

---

### 如何建立擴充性動作

透過 vRealize Automation Cloud Assembly，您可以建立擴充性動作以用於擴充性訂閱中。

擴充性動作是高度可自訂、輕量型，且彈性的方式，用來使用使用者定義的指令碼和動作範本延伸應用程式生命週期。動作範本包含可協助您設定擴充性動作的基礎的預先定義參數。

有兩種方式可建立擴充性動作：

- 為擴充性動作指令碼撰寫使用者定義的程式碼。

---

**備註** 在擴充性動作編輯器中寫入使用者定義的代碼時，可能需要作用中的網際網路連線。

---

- 為擴充性動作匯入 ZIP 套件格式的部署套件。如需為擴充性動作建立 ZIP 套件的相關資訊，請參閱 [為 Python 執行階段擴充性動作建立 ZIP 套件](#) 或 [為 Node.js 執行階段擴充性動作建立 ZIP 套件](#)。

下列步驟說明了建立擴充性動作以將 Amazon Web Services 用作 FaaS 提供者的程序。

#### 必要條件

- 具有作用中且有效專案中的成員資格。
- 為 Lambda 函數設定的 Amazon Web Services 角色。例如，AWSLambdaBasicExecutionRole。
- 雲端管理員角色或已啟用的 iam:PassRole 權限。

#### 程序

- 1 選取 **擴充性 > 程式庫 > 動作**。
- 2 按一下 **新增動作**。
- 3 輸入動作的名稱，然後選取專案。
- 4 按 **下一步**。
- 5 搜尋並選取動作範本。

---

**備註** 若要在不使用動作範本的情況下建立自訂動作，請選取 **自訂指令碼**。

---

會顯示新的可設定參數。

- 6 選取 **撰寫指令碼或匯入套件**。
- 7 選取 **動作執行階段**。

## 8 針對動作的進入點輸入**主要函數**名稱。

**備註** 對於從 ZIP 套件匯入的動作，主要函數還必須包括含進入點的指令碼檔案的名稱。例如，如果主要指令碼檔案標題為 `main.py`，且輸入點為 `handler (context, inputs)`，則主要函數的名稱必須是 `main.handler`。

## 9 定義動作的輸入和輸出參數。

## 10 (選擇性) 將應用程式相依性新增至動作。

**備註** 對於從 ZIP 套件匯入的動作，會自動新增應用程式相依性。

## 11 若要定義逾時和記憶體限制，請啟用**設定自訂逾時和限制**選項。

## 12 若要測試動作，請按一下**儲存**，然後按一下**測試**。

### 後續步驟

建立並驗證擴充性動作之後，您可以將其指派給訂閱。

**備註** 擴充性訂閱使用擴充性動作的最新發行版本。建立新版本的動作之後，請按一下編輯器視窗右上方的**版本**。若要發行您要在訂閱中使用的動作版本，請按一下**發行**。

### 匯出和匯入擴充性動作

透過 vRealize Automation Cloud Assembly，您可以匯出和匯入要在不同專案中使用的擴充性動作。

#### 必要條件

現有的擴充性動作。

#### 程序

### 1 匯出擴充性動作。

- a 導覽至**擴充性 > 程式庫 > 動作**。
- b 選取擴充性動作，然後按一下**匯出**。

動作指令碼及其相依性將做為 ZIP 檔案儲存在您的本機環境中。

### 2 匯入擴充性動作。

- a 導覽至**擴充性 > 程式庫 > 動作**。
- b 按一下**匯入**。
- c 選取匯出的擴充性動作，然後將其指派給專案。
- d 按一下**匯入**。

**備註** 如果匯入的擴充性動作已指派給指定的專案，系統會提示您選取衝突解決原則。

**備用方案** 還可以直接從動作編輯器選取**匯入套件**選項來匯入動作指令碼。

## 為 Python 執行階段擴充性動作建立 ZIP 套件

您可以建立包含 vRealize Automation Cloud Assembly 擴充性動作所使用的 Python 指令碼和相依性的 ZIP 套件。

有兩種方式可為擴充性動作建立指令碼：

- 在 vRealize Automation Cloud Assembly 中，直接在擴充性動作編輯器中撰寫指令碼。
- 在本機環境中建立您的指令碼，並將其與任何相關相依性一起新增至 ZIP 套件。

透過使用 ZIP 套件，您可以建立包含動作指令碼和相依性的自訂預先設定範本，可將其匯入至 vRealize Automation Cloud Assembly 以便在擴充性動作中使用。

此外，您也可以在本機環境中建立您的指令碼，並將其與任何相關相依性一起新增至 ZIP 套件。例如，您的環境缺少網際網路存取權時，使用 ZIP 套件。

還可以使用 ZIP 套件來建立包含多個 Python 指令碼檔案的擴充性動作。使用多個指令碼檔案有助於組織整理擴充性動作代碼的結構。

### 必要條件

如果您使用的是 Python 3.3 或更早版本，請下載並設定 PIP 套件安裝程式。請參閱 [〈Python 套件索引〉](#)。

### 程序

- 1 在本機電腦上，為動作指令碼和相依性建立資料夾。  
例如，`/home/user1/zip-action`。
- 2 將一或多個主要 Python 動作指令碼新增至該資料夾。  
例如，`/home/user1/zip-action/main.py`。
- 3 (選擇性) 將 Python 指令碼的任何相依性新增至該資料夾。
  - a 建立包含相依性的 `requirements.txt` 檔案。請參閱 [〈需求檔案〉](#)。
  - b 開啟 Linux shell。

---

**備註** vRealize Automation Cloud Assembly 中以動作為基礎的擴充性執行階段是以 Linux 為基礎。因此，在 Windows 環境中編譯的任何 Python 相依性可能會使產生的 ZIP 套件無法用於建立擴充性動作。因此，您必須使用 Linux shell。

---

- c 透過執行下列命令，在指令碼資料夾中安裝 `requirements.txt` 檔案：

```
pip install -r requirements.txt --target=home/user1/zip-action
```

- 4 在已指派的資料夾中，選取您的指令碼元素以及 (如果適用) `requirements.txt` 檔案，並將其壓縮為 ZIP 套件。

**備註** 您的指令碼和相依性元素必須儲存在 ZIP 套件的根層級。在 Linux 環境中建立 ZIP 套件時，您可能會遇到問題，即套件內容未儲存在根層級。如果您遇到此問題，請在命令列 shell 中執行 `zip -r` 命令以建立套件。

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

## 後續步驟

使用 ZIP 套件建立擴充性動作指令碼。請參閱[如何建立擴充性動作](#)。

### 為 Node.js 執行階段擴充性動作建立 ZIP 套件

您可以建立包含 vRealize Automation Cloud Assembly 擴充性動作所使用的 Node.js 指令碼和相依性的 ZIP 套件。

有兩種方式可為擴充性動作建立指令碼：

- 在 vRealize Automation Cloud Assembly 中，直接在擴充性動作編輯器中撰寫指令碼。
- 在本機環境中建立您的指令碼，並將其與任何相關相依性一起新增至 ZIP 套件。

透過使用 ZIP 套件，您可以建立包含動作指令碼和相依性的自訂預先設定範本，可將其匯入至 vRealize Automation Cloud Assembly 以便在擴充性動作中使用。

此外，您也可以在 vRealize Automation Cloud Assembly 服務無法解析與動作指令碼中的相依性相關聯的模組時 (例如，您的環境缺少網際網路存取權) 使用 ZIP 套件。

此外，還可以使用套件建立包含多個 Node.js 指令碼檔案的擴充性動作。使用多個指令碼檔案有助於組織整理擴充性動作代碼的結構。

## 程序

- 1 在本機電腦上，為動作指令碼和相依性建立資料夾。

例如，`/home/user1/zip-action`。

- 2 將一或多個主要 Node.js 動作指令碼新增至該資料夾。

例如，`/home/user1/zip-action/main.js`。

- 3 (選擇性) 將 Node.js 指令碼的任何相依性新增至該資料夾。

- a 在指令碼資料夾中建立包含相依性的 `package.json` 檔案。請參閱[建立 package.json 檔案](#)和[在 package.json 檔案中指定 dependencies 和 devDependencies](#)。
- b 開啟命令列 shell。

- c 導覽至針對動作指令碼和相依性建立的資料夾。

```
cd /home/user1/zip-action
```

- d 透過執行下列命令，在指令碼資料夾中安裝 `package.json` 檔案：

```
npm install --production
```

**備註** 此命令會在您的資料夾中建立 `node_modules` 目錄。

- 4 在已指派的資料夾中，選取您的指令碼元素以及 (如果適用) `node_modules` 目錄，並將其壓縮為 ZIP 套件。

**備註** 您的指令碼和相依性元素必須儲存在 ZIP 套件的根層級。在 Linux 環境中建立 ZIP 套件時，您可能會遇到問題，即套件內容未儲存在根層級。如果您遇到此問題，請在命令列 shell 中執行 `zip -r` 命令以建立套件。

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

#### 後續步驟

使用 ZIP 套件建立擴充性動作指令碼。請參閱[如何建立擴充性動作](#)。

#### 設定特定於雲端的擴充性動作

您可以設定擴充性動作以用於雲端帳戶。

建立擴充性動作時，您可以設定此動作並將其連結至各個雲端帳戶：

- Microsoft Azure
- Amazon Web Services

#### 必要條件

需要有效的雲端帳戶。

#### 程序

- 1 選取**擴充性** > **程式庫** > **動作**。
- 2 按一下**新增動作**。
- 3 視需要輸入動作參數。
- 4 在 **FaaS 提供者** 下拉式功能表中，選取雲端帳戶提供者或選取**自動**。

**備註** 如果選取**自動**，動作會自動定義 FaaS 提供者。

- 5 按一下**儲存**。

## 結果

擴充性動作已連結，可用於設定的雲端帳戶。

### 設定內部部署擴充性動作

您可以將擴充性動作設定為使用內部部署 FaaS 提供者，而非 Amazon Web Services 或 Microsoft Azure 雲端帳戶。

透過針對擴充性動作使用內部部署 FaaS 提供者，您可以在 vRealize Automation Cloud Assembly 擴充性訂閱中使用內部部署服務，例如 LDAP、CMDB 或 vCenter 資料中心。

## 程序

- 1 選取**擴充性 > 程式庫 > 動作**。
- 2 按一下**新增動作**。
- 3 輸入擴充性動作的名稱和專案。
- 4 (選擇性) 輸入擴充性動作的說明。
- 5 按下一步。
- 6 建立或匯入擴充性動作指令碼。
- 7 按一下 **FaaS 提供者** 下拉式功能表，然後選取**內部部署**。
- 8 若要儲存新的擴充性動作，請按一下**儲存**。

## 後續步驟

在 vRealize Automation Cloud Assembly 擴充性訂閱中使用已建立的擴充性動作。

## 什麼是動作流程

動作流程是一組擴充性動作指令碼，用於延長生命週期和進一步自動化。

所有動作流程均以 `flow_start` 開頭且以 `flow_end` 結尾。您可以使用下列動作流程元素將多個擴充性動作指令碼連結在一起：

- **連續動作流程** - 多個擴充性動作指令碼會依序執行。
- **分支動作流程** - 分割路徑以提供相同輸出的多個擴充性動作指令碼或流程。
- **加入動作流程** - 一起加入並提供相同輸出的多個擴充性動作指令碼或流程。
- **條件式動作流程** - 滿足條件後執行的多個擴充性動作指令碼或流程。

### 連續動作流程

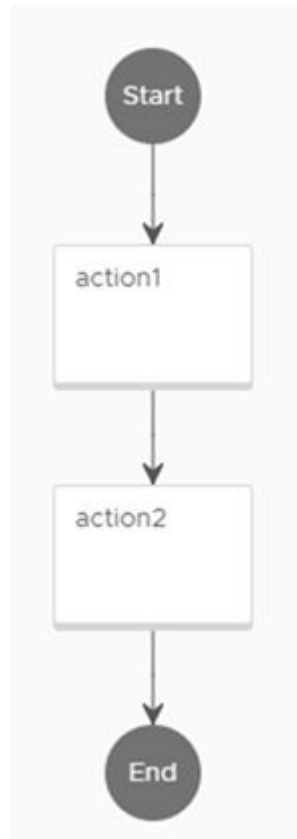
多個擴充性動作指令碼會依序執行。

```

version: "1"
flow:
  flow_start:
    next: action1
  action1:
    action: <action_name>
    next: action2
  action2:
    action: <action_name>
    next: flow_end

```

**備註** 您可以透過將先前動作指派為 `next: 動作`，來循環回先前動作。例如，在此範例中，您可以不執行 `next: flow_end`，而輸入 `next: action1` 重新執行 `action1` 並重新啟動動作序列。



## 分支動作流程

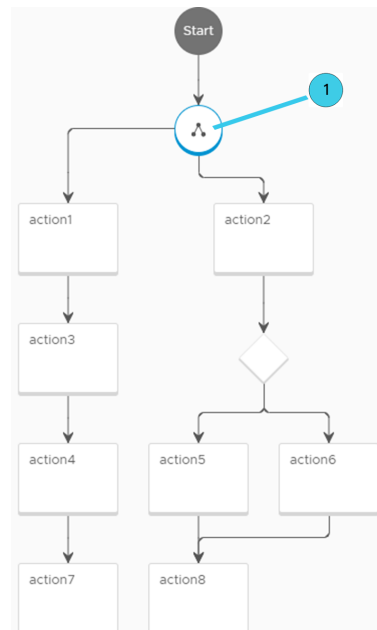
分割路徑以提供相同輸出的多個擴充性動作指令碼或流程。

```

version: "1"
flow:
  flow_start:
    next: forkAction
  forkAction:
    fork:
      next: [action1, action2]
  action1:
    action: <action_name>
    next: action3
  action3:
    action: <action_name>
    next: action4
  action4:
    action: <action_name>
    next: action7
  action7:
    action: <action_name>
  action2:
    action: <action_name>

```

**備註** 您可以透過將先前動作指派為 `next: 動作`，來循環回先前動作。例如，您可以不執行 `next: flow_end` 來結束動作流程，而輸入 `next: action1` 重新執行 `action1` 並重新啟動動作序列。



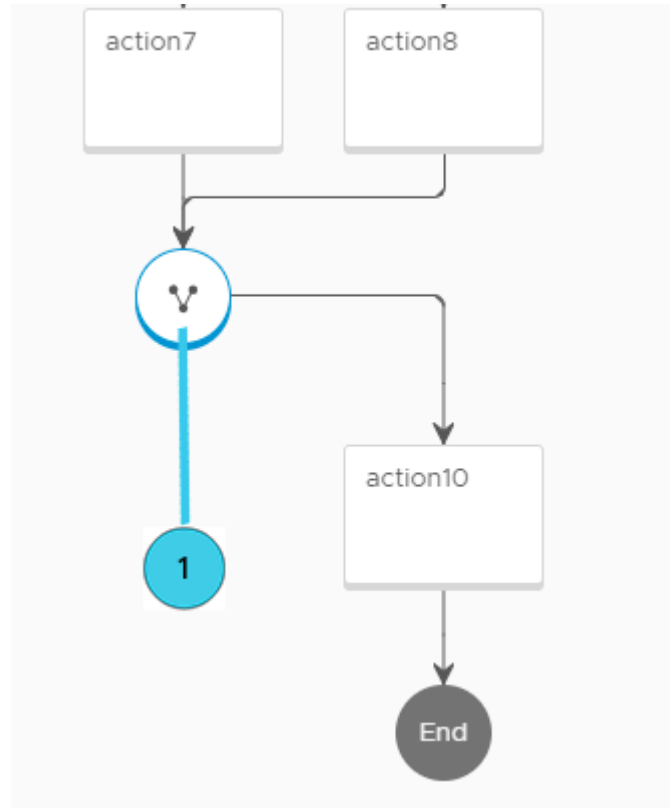
① 分支元素

## 加入動作流程

同時加入路徑並提供相同輸出的多個擴充性動作指令碼或流程。

```
version: "1"
action7:
  action: <action_name>
  next: joinElement
action8:
  action: <action_name>
  next: joinElement
joinElement:
  join:
    type: all
    next: action10
action10:
  action: <action_name>
  next: flow_end
```

**備註** 您可以透過將先前動作指派為 `next: 動作`，來循環回先前動作。例如，在此範例中，您可以不執行 `next: flow_end`，而輸入 `next: action1` 重新執行 `action1` 並重新啟動動作序列。



1 加入元素

## 條件式動作流程

使用切換元素滿足條件時執行的多個擴充性動作指令碼或流程。

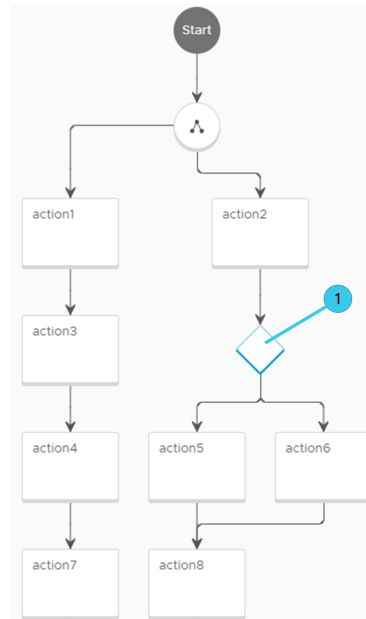
在某些情況下，條件必須等於 `true` 動作才能執行。在其他情況下，如此範例所示，必須滿足參數值，動作才能執行。如果不滿足任一條件，動作流程將會失敗。



```

version: 1
id: 1234
name: Test
inputs: ...
outputs: ...
flow:
  flow_start:
    next: forkAction
  forkAction:
    fork:
      next: [action1, action2]
  action1:
    action: <action_name>
    next: action3
  action3:
    action: <action_name>
    next: action4
  action4:
    action: <action_name>
    next: action7
  action7:
    action: <action_name>
    next: joinElement
  action2:
    action: <action_name>
    next: switchAction
  switchAction:
    switch:
      "${1 == 1}": action5
      "${1 != 1}": action6
  action5:
    action: <action_name>
    next: action8
  action6:
    action: <action_name>
    next: action8
  action8:
    action: <action_name>

```



① 切換元素

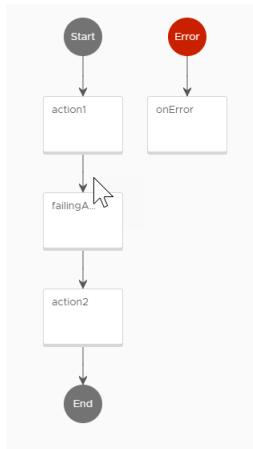
**備註** 您可以透過將先前動作指派為 `next: 動作`，來循環回先前動作。例如，您可以不執行 `next: flow_end` 來結束動作流程，而輸入 `next: action1` 重新執行 `action1` 並重新啟動動作序列。

### 如何將錯誤處理常式與動作流程搭配使用

您可以設定動作流程，以在流程的指定階段使用錯誤處理常式元素發出差錯。

錯誤處理常式元素需要下列兩個輸入：

- 為失敗動作指定的錯誤訊息。
- 動作流程輸入。



如果流程中的動作失敗且動作流程包含錯誤處理常式元素，則會發出錯誤訊息以警示您動作失敗。錯誤處理常式本身是一項動作。下列指令碼是您可在動作流程中使用的錯誤處理常式的範例。

```
def handler(context, inputs):

    errorMsg = inputs["errorMsg"]
    flowInputs = inputs["flowInputs"]

    print("Flow execution failed with error {0}".format(errorMsg))
    print("Flow inputs were: {0}".format(flowInputs))

    outputs = {
        "errorMsg": errorMsg,
        "flowInputs": flowInputs
    }

    return outputs
```

您可以在 [動作執行] 視窗中檢視成功和失敗的執行。

狀態	動作	動作識別碼
✓ 已完成	AWS-ABX	8a769ecc6df809c7016e01a83fe204e6
❌ 失敗	AWS-ABX	8a769ecc6df809c7016e01a83fe204e6
✓ 已完成	AWS-ABX	8a769ecc6df809c7016e01a83fe204e6
✓ 已完成	AWS-ABX	8a769ecc6df809c7016e01a83fe204e6

在此範例中，包含錯誤處理常式元素的 flow-with-handler 動作流程已成功執行。但是，此流程的其中一個動作失敗，並啟動了錯誤處理常式以發出錯誤。

## 如何追蹤動作執行

此動作執行索引標籤會為您顯示訂閱觸發的擴充性動作及其狀態的記錄。

您可以使用**擴充性 > 活動 > 動作執行**來檢視動作執行記錄。此外，您還可以一次依一或多個內容篩選動作執行清單。若要檢視個別動作執行的其他詳細資料，請按一下 [執行識別碼]。

## 對失敗的擴充性動作執行進行疑難排解

如果您的擴充性動作執行失敗，可以執行疑難排解步驟來更正它。

當動作執行失敗時，您可能會收到錯誤訊息、失敗的狀態，以及失敗的記錄。如果動作執行失敗，可能是由於部署或程式碼失敗。

問題	解決方案
部署失敗	這些失敗是與雲端帳戶組態、動作部署或可防止動作進行部署的其他相依性相關的問題所致。確保您使用的專案是在設定的雲端帳戶中定義，且擁有執行功能的權限。再次起始動作之前，您可以針對動作詳細資料頁面中的特定專案測試動作。
程式碼失敗	這些失敗是無效的指令碼或程式碼所致。使用動作執行記錄疑難排解和更正無效的指令碼。

## 擴充性工作流程訂閱

您可以將 vRealize Orchestrator 主控的工作流程與 vRealize Automation Cloud Assembly 搭配使用來延長應用程式生命週期。

## 如何使用 vRealize Orchestrator 工作流程訂閱修改虛擬機器內容

您可以使用現有的 vRealize Orchestrator 工作流程，來修改虛擬機器內容，以及將虛擬機器新增至 Active Directory。

訂閱架構會定義事件代理服務 (EBS) 訊息的裝載格式。若要在工作流程中接收和使用 EBS 訊息裝載，您必須定義「inputProperties」工作流程輸入參數。

### 必要條件

- 雲端管理員使用者角色
- 現有的 vRealize Orchestrator 內部部署工作流程。
- 與 vRealize Orchestrator 用戶端伺服器成功整合並連線。

### 程序

- 1 選取**擴充性 > 訂閱**。
- 2 按一下**新增訂閱**。

### 3 使用下列參數建立訂閱：

參數	值
名稱	RenameVM
事件主題	選取適用於所需 vRealize Orchestrator 整合的事件主題。例如，計算配置。
封鎖/非封鎖	非封鎖
可執行的項目	選取 vRealize Orchestrator 可執行的類型。
可執行的識別碼	選取所需工作流程。例如，設定虛擬機器名稱。

### 4 若要儲存訂閱，請按一下**建立**。

### 5 透過建立藍圖或部署現有藍圖來指派並啟用您的訂閱。

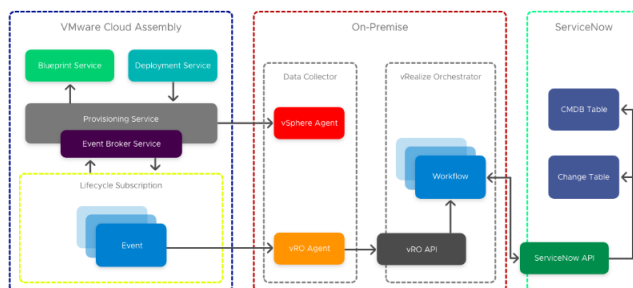
#### 後續步驟

透過以下方法之一確認已成功起始工作流程：

- 確認工作流程執行記錄，**擴充性 > 活動 > 工作流程執行**。
- 開啟 vRealize Orchestrator 用戶端，並透過導覽至工作流程並確認狀態，或透過開啟特定記錄索引標籤，來檢查工作流程狀態。

## 如何使用 vRealize Orchestrator 工作流程將適用於 ITSM 的 Cloud Assembly 與 ServiceNow 整合

使用 vRealize Orchestrator 主控的工作流程，可以將 vRealize Automation Cloud Assembly 與 ServiceNow 整合以實現 ITSM 符合性。



企業使用者通常會將其雲端管理平台與 IT 服務管理 (ITSM) 和設定管理資料庫 (CMDB) 平台整合以實現符合性。按照此範例，您可以使用 vRealize Orchestrator 主控的工作流程將 vRealize Automation Cloud Assembly 與適用於 CMDB 和 ITSM 的 ServiceNow 整合。使用 vRealize Orchestrator 整合和工作流程時，如果您有不同環境的多個執行個體，則功能標籤特別有用。如需有關功能標籤的詳細資訊，請參閱在 [vRealize Automation Cloud Assembly 中使用功能標籤](#)。

**備註** 還可以使用擴充性動作指令碼將 ServiceNow 與 vRealize Automation Cloud Assembly 整合。如需使用擴充性動作指令碼整合 ServiceNow 的相關資訊，請參閱 [如何使用擴充性動作將 Cloud Assembly 與 ServiceNow 整合](#)。

在此範例中，ServiceNow 整合由三個頂層工作流程組成。每個工作流程都有自己的訂閱，以便您可以單獨更新和逐一查看每個元件。

- 事件訂閱進入點 - 基本記錄，識別申請使用者和 vCenter 虛擬機器 (如果適用)。
- 整合工作流程 - 分隔技術工作流程的物件和摘要輸入，處理記錄、內容和輸出更新。
- 技術工作流程 - 下游系統整合，可供 ServiceNow API 建立具有除裝載以外的其他虛擬機器內容的 CMDB CI、CR 和 CAS IaaS API。

#### 必要條件

- 獨立或叢集化 vRealize Orchestrator 環境。
- vRealize Automation Cloud Assembly 中的 vRealize Orchestrator 整合。如需將獨立 vRealize Orchestrator 與 vRealize Automation Cloud Assembly 整合的相關資訊，請參閱在 [Cloud Assembly 中設定 vRealize Orchestrator 整合](#)。

#### 程序

- 1 在 vRealize Orchestrator 中建立和儲存包含多個工作流程中常用的組態的組態檔。
- 2 將 CAS API Token 儲存到與步驟 1 中的組態檔相同的位置。

---

**備註** CAS API Token 會到期。

---

- 3 在 vRealize Orchestrator 中建立具有提供的指令碼元素的工作流程。此指令碼將參考並尋找 REST 主機。此指令碼還會標準化使用 Token 的選擇性參數的 REST 動作，該參數新增為額外的授權標頭。

```
var configPath = "CS"
var configName = "environmentConfig"
var attributeName = "CASRestHost"

//get REST Host from configuration element
var restHost =
System.getModule("au.com.cs.example").getRestHostFromConfig(configPath,configName,attribute
Name)

var ConfigurationElement =
System.getModule("au.com.cs.example").getConfigurationElementByName(configName,configPath);
System.debug("ConfigurationElement:" + ConfigurationElement);
var casToken = ConfigurationElement.getAttributeWithKey("CASToken")["value"]
if(!casToken){
    throw "no CAS Token";
}
//REST Template
var opName = "casLogin";
var opTemplate = "/iaas/login";
var opMethod = "POST";

// create the REST operation:
var opLogin =
System.getModule("au.com.cs.example").createOp(restHost,opName,opMethod,opTemplate);
```

```
//cas API Token
var contentObject = {"refreshToken":casToken}
postContent = JSON.stringify(contentObject);

var loginResponse =
System.getModule("au.com.cs.example").executeOp(opLogin,null,postContent,null) ;

try{
    var tokenResponse = JSON.parse(loginResponse)['token']
    System.debug("token: " + tokenResponse);
} catch (ex) {
    throw ex + " No valid token";
}

//REST Template Machine Details
var opName = "machineDetails";
var opTemplate = "/iaas/machines/" + resourceId;
var opMethod = "GET";

var bearer = "Bearer " + tokenResponse;

var opMachine =
System.getModule("au.com.cs.example").createOp(restHost,opName,opMethod,opTemplate);

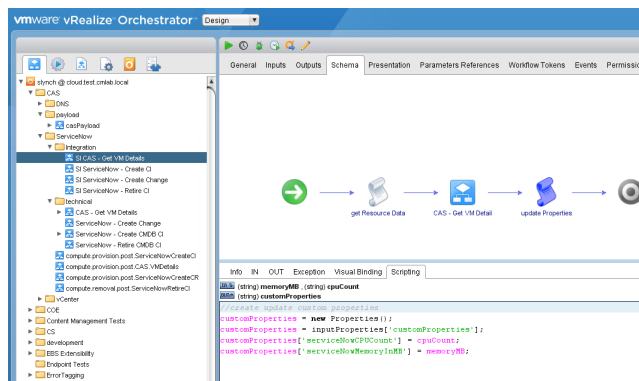
// (Rest Operation, Params, Content, Auth Token)
var vmResponse =
System.getModule("au.com.cs.example").executeOp(opMachine,null,"",bearer) ;

try{
    var vm = JSON.parse(vmResponse);
} catch (ex) {
    throw ex + " failed to parse vm details"
}

System.log("cpuCount: " + vm["customProperties"]["cpuCount"]);
System.log("memoryInMB: " + vm["customProperties"]["memoryInMB"]);

cpuCount = vm["customProperties"]["cpuCount"];
memoryMB = vm["customProperties"]["memoryInMB"];
```

此指令碼將輸出 `cpuCount` 和 `memoryMB` 傳送至父系工作流程，並更新現有 `customProperties` 內容。建立 CMDB 時，可以在後續工作流程中使用這些值。



- 4 將「建立 ServiceNow CMDB CI」指令碼元素新增至工作流程。此元素使用組態項目尋找 ServiceNow REST 主機，為 cmdb\_ci\_vmware\_instance 資料表建立 REST 作業，根據 post 資料的工作流程輸入建立一系列內容物件，並輸出傳回的 sys\_id。

```

var configPath = "CS"
var configName = "environmentConfig"
var attributeName = "serviceNowRestHost"
var tableName = "cmdb_ci_vmware_instance"

//get REST Host from configuration element
var restHost =
System.getModule("au.com.cs.example").getRestHostFromConfig(configPath,configName,attribute
Name)

//REST Template
var opName = "serviceNowCreatCI";
var opTemplate = "/api/now/table/" + tableName;
var opMethod = "POST";

// create the REST operation:
var opCI =
System.getModule("au.com.cs.example").createOp(restHost,opName,opMethod,opTemplate);

//cmdb_ci_vm_vmware table content to post;
var contentObject = {};
contentObject["name"] = hostname;
contentObject["cpus"] = cpuTotalCount;
contentObject["memory"] = MemoryInMB;
contentObject["correlation_id"]= deploymentId
contentObject["disks_size"]= diskProvisionGB
contentObject["location"] = "Sydney";
contentObject["vcenter_uuid"] = vcUuid;
contentObject["state"] = "On";
contentObject["owned_by"] = owner;

postContent = JSON.stringify(contentObject);
System.log("JSON: " + postContent);

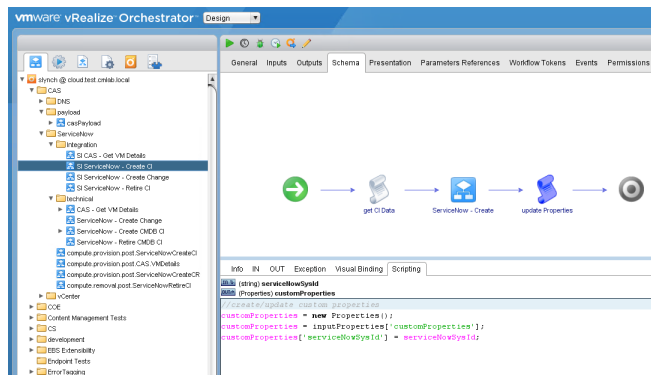
// (Rest Operation, Params, Content, Auth Token)
var ciResponse =
System.getModule("au.com.cs.example").executeOp(opCI,null,postContent,null) ;

try{
    var cmdbCI = JSON.parse(ciResponse);
} catch (ex) {
    throw ex + " failed to parse ServiceNow CMDB response";
}

serviceNowSysId = cmdbCI['result']['sys_id'];

```

- 5 使用來自子系工作流程的輸出，透過現有 `customProperties` 建立一個內容物件並使用來自 ServiceNow 的值覆寫 `serviceNowSysId` 內容。此唯一識別碼在 CMDB 中用於將執行個體標記為銷毀時淘汰。



## 結果

vRealize Automation Cloud Assembly 已成功與 ITSM ServiceNow 整合。如需有關如何使用工作流程在 vRealize Automation Cloud Assembly 中整合 ServiceNow 的詳細資訊，請參閱〈[為 ServiceNow 整合使用 vRealize Orchestrator 延伸 Cloud Assembly](#)〉。

## 進一步瞭解工作流程訂閱

透過使用 vRealize Orchestrator 與 vRealize Automation Cloud Assembly 的整合，您可以透過工作流程延伸應用程式的生命週期。

vRealize Automation Cloud Assembly 包括 vRealize Orchestrator 整合，可用來匯入並將現有的內部部署工作流程連結至訂閱。這些工作流程會保留在 vRealize Orchestrator 伺服器上。僅可以使用 vRealize Orchestrator 用戶端建立、修改和刪除工作流程。

### 建立 vRealize Orchestrator 工作流程的最佳做法

工作流程訂閱是以特定主題架構為基礎。若要確保訂閱可啟動 vRealize Orchestrator 工作流程，您必須為其設定正確的輸入參數，以便它們能夠使用事件資料。

#### 工作流程輸入參數

您的自訂工作流程可包含所有參數或耗用裝載中所有資料的單一參數。

若要使用單一參數，請為某個參數設定 `Properties` 類型和名稱 `inputProperties`。

#### 工作流程輸出參數

您的自訂工作流程可包含與回覆事件主題類型所必需的後續事件相關的輸出參數。

如果事件主題預期回覆，工作流程輸出參數必須符合回覆架構。

#### 如何追蹤工作流程執行

工作流程執行索引標籤會為您顯示訂閱觸發的工作流程及其狀態的記錄。

您可以使用 **擴充性 > 活動 > 工作流程執行** 來檢視工作流程執行記錄。



## 對失敗的工作流程訂閱進行疑難排解

如果您的工作流程訂閱失敗，可以執行疑難排解步驟來更正它。

失敗的工作流程執行可能會導致工作流程訂閱無法成功啟動或完成。工作流程執行失敗可能是由幾個常見問題導致的。

問題	原因	解決方案
vRealize Orchestrator 工作流程訂閱未成功啟動或完成。	將工作流程訂閱設定為在收到事件訊息時執行自訂工作流程，但該工作流程並未成功執行或完成。	<ol style="list-style-type: none"> <li>1 確認工作流程訂閱已正確儲存。</li> <li>2 確認工作流程訂閱條件已正確設定。</li> <li>3 確認 vRealize Orchestrator 包含指定的工作流程。</li> <li>4 確認已在 vRealize Orchestrator 中正確設定工作流程。</li> </ol>
您的核准要求 vRealize Orchestrator 工作流程訂閱未執行。	您已將核准前或核准後工作流程訂閱設定為執行 vRealize Orchestrator 工作流程。在服務目錄中要求符合已定義準則的機器時，該工作流程不會執行。	<p>若要成功執行核准工作流程訂閱，您必須確認所有元件已正確設定。</p> <ol style="list-style-type: none"> <li>1 確認核准原則處於作用中，並正確套用。</li> <li>2 確認您的工作流程訂閱已正確設定和儲存。</li> <li>3 檢閱與核准相關的訊息事件記錄。</li> </ol>
您的核准要求 vRealize Orchestrator 工作流程訂閱遭到拒絕。	<p>已設定執行指定 vRealize Orchestrator 工作流程的核准前或核准後工作流程訂閱，但此要求在外部核准層級遭到拒絕。</p> <p>一個可能的原因是 vRealize Orchestrator 中出現內部工作流程執行錯誤。例如，工作流程遺失或 vRealize Orchestrator 伺服器不在執行中。</p>	<ol style="list-style-type: none"> <li>1 檢閱與核准相關的訊息記錄。</li> <li>2 確認 vRealize Orchestrator 伺服器正在執行。</li> <li>3 確認 vRealize Orchestrator 包含指定的工作流程。</li> </ol>

## 進一步瞭解擴充性訂閱

您可以將擴充性動作或 vRealize Orchestrator 主控的工作流程與擴充性訂閱搭配使用，來延長應用程式生命週期。

當環境中發生觸發事件時，會啟動訂閱，並執行指定的工作流程或擴充性動作。您可以在事件記錄中檢視系統事件，在 [工作流程執行] 視窗中檢視工作流程執行，以及在 [動作執行] 視窗中檢視動作執行。訂閱是特定於專案的，這表示訂閱透過指定的專案連結到藍圖和部署。

## 擴充性術語

當您在 vRealize Automation Cloud Assembly 中使用擴充性和訂閱時，可能會遇到一些特定於訂閱和事件代理服務的術語。

表 6-4. 擴充性術語

術語	說明
事件主題	說明一組具有相同邏輯意圖和相同結構的事件。每個事件是事件主題的執行個體。 您可以將封鎖的參數指派給某些事件主題。如需詳細資訊，請參閱 <a href="#">封鎖事件主題</a> 。
事件	指示產生者中的狀態變更或任何受其管理的實體。事件是記錄事件發生相關資訊的實體。
事件代理服務	將產生者發佈的訊息分派給已訂閱的取用者的服務。
裝載	事件資料，其中包含與此事件主題相關的所有內容。
訂閱	指示訂閱者有興趣接收事件相關通知，因而訂閱了事件主題並定義了觸發通知的準則。訂閱將擴充性動作或工作流程連結至觸發的事件，用於自動執行應用程式生命週期的一些部分。
訂閱者	根據訂閱定義發佈至事件代理服務的事件通知的使用者。訂閱者也稱為取用者。
系統管理員	具有可使用 vRealize Automation Cloud Assembly 建立、讀取、更新和刪除承租人工作流程訂閱與系統工作流程訂閱之權限的使用者。
工作流程訂閱	指定觸發 vRealize Orchestrator 工作流程的事件主題和條件。
動作訂閱	指定觸發擴充性動作使其執行的事件主題和條件。
工作流程	整合在 vRealize Automation Cloud Assembly 中的 vRealize Orchestrator 工作流程。您可以將這些工作流程連結至訂閱中的事件。
擴充性動作	可以在訂閱中的觸發事件後執行的簡化的指令碼。擴充性動作與工作流程類似，但更為輕量。擴充性動作可從 vRealize Automation Cloud Assembly 中進行自訂。
動作執行	可透過 <b>動作執行</b> 索引標籤存取。動作執行是為了回應觸發的事件所執行的擴充性動作的詳細記錄。

## 封鎖事件主題

部分事件主題支援封鎖事件。擴充性訂閱的行為取決於主題是否支援這些事件類型以及您設定訂閱的方式。

vRealize Automation Cloud Assembly 擴充性訂閱可以使用兩大類型的事件主題：非封鎖和封鎖事件主題。事件主題類型會定義擴充性訂閱的行為。

### 非封鎖事件主題

非封鎖事件主題僅允許您建立非封鎖訂閱。系統以非同步方式觸發非封鎖訂閱，您不能依賴訂閱的觸發順序。

### 封鎖事件主題

部分事件主題支援封鎖。如果訂閱標記為封鎖，則任何具有相符條件的其他訂閱都不會收到符合所設定條件的所有訊息，直到執行封鎖訂閱的可執行項目為止。

封鎖訂閱將按優先順序執行。最高優先順序值為 0 (零)。如果同一事件主題具有優先順序層級相同的多個封鎖訂閱，這些訂閱會根據訂閱名稱按相反的字母順序執行。處理完所有封鎖訂閱之後，會將訊息同時傳送給所有非封鎖訂閱。由於封鎖訂閱以同步方式執行，因此，變更的事件裝載包括通知後續訂閱時更新的事件。

您可以使用封鎖事件主題來管理彼此相依的多個訂閱。

例如，您具有兩個佈建工作流程訂閱，其中第二個訂閱取決於第一個訂閱的結果。第一個訂閱在佈建期間變更內容，而第二個訂閱將新內容 (例如機器名稱) 記錄在檔案系統中。ChangeProperty 訂閱的優先順序為 0，而 RecordProperty 的優先順序為 1，因為第二個訂閱使用第一個訂閱的結果。佈建機器後，ChangeProperty 訂閱便開始執行。由於 RecordProperty 訂閱條件以佈建後條件為基礎，因此事件會觸發 RecordProperty 訂閱。但是，由於 ChangeProperty 工作流程為封鎖工作流程，因此在其完成前不會收到事件。當機器名稱發生變更並且第一個工作流程訂閱完成後，第二個工作流程訂閱便會執行，同時將機器名稱記錄在檔案系統中。

### 復原可執行的項目

對於封鎖事件主題，您可以將復原可執行項目新增至訂閱。如果主要可執行項目失敗，則訂閱中的復原可執行項目將會執行。例如，您可以建立一個工作流程訂閱，其中主要可執行項目是在 CMDB 系統 (例如 ServiceNow) 中建立記錄的工作流程。即使工作流程訂閱失敗，也可能會在 CMDB 系統中建立某些記錄。在此案例中，可使用復原可執行項目清理失敗的可執行項目在 CMDB 系統中留下的記錄。

對於包含多個彼此相依的訂閱的使用案例，您可以將 `ebs.recover.continuation` 內容新增至復原可執行項目。藉由此內容，可以指示擴充性服務在目前訂閱失敗的情況下是否必須繼續進行鏈結中的下一個訂閱。

## vRealize Automation Cloud Assembly 隨附的事件主題

vRealize Automation Cloud Assembly 包括預先定義的事件主題。

### 事件主題

事件主題是將類似事件分組在一起的類別。指派至訂閱時，事件主題會定義由哪個事件觸發訂閱。依預設，vRealize Automation Cloud Assembly 會提供下列事件主題。所有主題均可用於新增或更新自訂內容或資源的標籤。如果 vRealize Orchestrator 工作流程或擴充性動作失敗，對應的工作也會失敗。

表 6-5. Cloud Assembly 事件主題

事件主題	可封鎖	說明
Blueprint.configuration	否	在建立或刪除藍圖之類的藍圖設定事件發生時發出，可用於向外部系統通知此類事件。
Blueprint.version.configuration	否	在新藍圖版本設定事件發生時發出，例如建立、發行、取消發行或還原版本。此事件主題對於整合第三方版本控制系統很有用。
Compute allocation	是	配置 <code>resourcenames</code> 和 <code>hostselections</code> 之前所發出的事件。可在此階段修改這兩個內容。
Compute post provision	是	成功佈建資源後所發出的事件。

表 6-5. Cloud Assembly 事件主題 (續)

事件主題	可封鎖	說明
Compute post removal	是	移除計算資源後發出的事件。
Compute provision	是	在 Hypervisor 佈建資源之前發生的事件。 <b>備註</b> 您可以變更已配置的 IP 位址。
Compute removal	是	移除資源之前所發出的事件。
Compute reservation	是	保留時所發出的事件。 <b>備註</b> 您可以變更放置順序。
Deployment action completed	是	部署動作完成後發出。
Deployment action requested	是	部署動作完成前發出。
Deployment completed	是	部署藍圖或目錄請求後發出。
Deployment onboarded	否	將新部署上線時發出。
Deployment requested	是	部署藍圖或目錄請求前發出。
Deployment resource action completed	是	部署資源動作後發出。
Deployment resource action requested	是	部署資源動作前發出。
Deployment resource completed	是	佈建部署資源後發出。
Deployment resource requested	是	佈建部署資源前發出。
Disk allocation	是	針對預先配置磁碟資源發出。
Disk post removal	是	刪除磁碟資源後發出。
Disk post resize	是	重新調整磁碟資源大小後發出。
EventLog	是	記錄相關事件。
Kubernetes cluster allocation	是	針對預先配置 Kubernetes 叢集的資源發出。
Kubernetes cluster post provision	是	佈建 Kubernetes 叢集後發出。
Kubernetes cluster post removal	是	刪除 Kubernetes 叢集後發出。
Kubernetes cluster provision	是	佈建 Kubernetes 叢集前發出。
Kubernetes cluster removal	是	起始刪除 Kubernetes 叢集的程序前發出。
Load balancer post provision	是	佈建負載平衡器後發出。

表 6-5. Cloud Assembly 事件主題 (續)

事件主題	可封鎖	說明
Network Configure	是	在計算資源配置期間設定網路時發出的事件。 <b>備註</b> [設定網路] 主題支援多個 IP 位址/NIC。
Project Lifecycle	否	建立、更新或刪除專案時發出的事件。

## 事件架構

新增事件主題後，您可以檢視事件架構。此架構定義了事件裝載或 `inputProperties` 的結構。

## 擴充性事件記錄

[擴充性事件] 索引標籤會顯示您環境中已發生的所有事件的清單。

您可以使用 **擴充性 > 事件** 檢視擴充性事件記錄。此外，您可以一次依照一或多個內容篩選事件清單。若要檢視個別事件的其他詳細資料，請按一下事件的識別碼。

事件 3442 個項目

識別碼	時間戳記	事件主題	使用者名稱	目標識別碼	說明
d4b40f20-9792-4c48-aec0-0a16ce7eaabe	20/01/14 下午6:14	Blueprint configuration	pmartini@vmware.com		
1d86cdf8-9555-d312-1836-00d8b2a262e6	20/01/14 下午4:46	Project Lifecycle Event Topic	project-service	1d0b482d-8396-46a1-a493-54ed2d299ec8	update
dfcf544c-d263-c30c-3b58-23b31be25f6a	20/01/14 下午4:31	EventLog	vro-gateway	77ef8b8b-3180-44d0-a69a-503f27edc42c	Workflow run [77ef8b8b-3180-44d0-a69a-503f27edc42c] status changed to [FAILED]. workflowid [7246b7b5-fe89-4da5-a640-36ffc6874069] eventid: [33c01131-c9f9-3255-a686-a67671f44bcb]
16e0c6ed-40e4-4cb8-9be6-f7543b9c6a7c	20/01/14 下午3:55	Deployment completed	pmartini@vmware.com		
b754bc46-f255-4869-b789-252924b32c4d	20/01/14 下午3:55	Deployment requested	pmartini@vmware.com		

## 建立擴充性訂閱

透過將 vRealize Orchestrator 整合或擴充性動作與 vRealize Automation Cloud Assembly 搭配使用，您可以建立訂閱以延伸應用程式。

使用擴充性訂閱，您可以透過在發生特定生命週期事件時觸發工作流程或動作來延伸應用程式。也可以對訂閱套用篩選器來為指定的事件設定布林值條件。例如，僅當布林運算式為「true」時，事件與工作流程/動作才會觸發。如果您要控制何時觸發事件和動作，這十分有用。

**提示** 在 [依主題篩選事件] 文字方塊中，使用「Alt + 空格鍵」(在 Windows 上) 或「Option + 空格鍵」(在 Mac 上) 可顯示篩選選項。

## 必要條件

- 雲端管理員使用者角色
- 如果使用 vRealize Orchestrator 工作流程：
  - 內嵌式 vRealize Orchestrator 用戶端的程式庫或任何整合式外部 vRealize Orchestrator 執行個體的程式庫。
- 如果使用擴充性動作：
  - 現有的擴充性動作指令碼。如需詳細資訊，請參閱[如何建立擴充性動作](#)。

## 程序

- 1 選取**擴充性 > 訂閱**。
- 2 按一下**新增整合**。
- 3 輸入訂閱的詳細資料。
- 4 選取**事件主題**。
- 5 (選擇性) 設定事件主題的條件。
- 6 (選擇性) 如果適用，請為事件主題設定封鎖行為。
- 7 按一下**可執行的項目**，然後從下拉式功能表中選取 **vRO 工作流程**或 **ABX 動作**。
- 8 選取您要在訂閱中執行的工作流程或擴充性動作。
- 9 (選擇性) 若要定義擴充性訂閱的專案範圍，請停用**任何專案**，然後按一下**新增專案**。
- 10 按一下**建立儲存您的訂閱**。

## 結果

將建立您的訂閱。當事件 (依所選事件主題分類) 發生時，會起始連結的 vRealize Orchestrator 工作流程或擴充性動作，並通知所有訂閱者。

## 後續步驟

建立訂閱後，您可以建立或部署藍圖，以連結和使用訂閱。此外，您還可以在 vRealize Automation Cloud Assembly 的**擴充性**索引標籤中驗證工作流程執行的狀態。對於包含 vRealize Orchestrator 工作流程的訂閱，也可以從 vRealize Orchestrator 用戶端監控執行與工作流程狀態。

## 疑難排解擴充性訂閱

對擴充性訂閱失敗進行疑難排解。

訂閱失敗時，通常是工作流程或擴充性動作指令碼有錯誤所致。

## 檢視主題參數和裝載

您可以使用傾印訂閱主題參數指令碼，來檢視虛擬機器在任何指定事件階段的特定參數和裝載。

此指令碼主要用於偵錯和檢查 vRealize Orchestrator 工作流程的可用輸入。若要檢視虛擬機器的所有參數，請將下列指令碼與工作流程搭配使用：

```
function dumpProperties(props, lvl) {
    var keys = props.keys;
    var prefix = ""
    for (var i=0; i<lvl; i++){
        prefix = prefix + "";
    }
    for (k in keys){
        var key = keys[k];
        var value = props.get(keys[k])
        if ("Properties" == System.getObjectType(value)){
            System.log(Prefix + key + "[")
            dumpProperties(value, (lvl+2));
            System.log(prefix+ "]")
        } else{
            System.log( prefix + key + ":" + value)
        }
    }
}

dumpProperties(inputProperties, 0)

customProps = inputProperties.get("customProperties")
```

## 訂閱版本歷程記錄

如果您的訂閱失敗，您可以檢視版本歷程記錄。

### 檢視訂閱版本歷程記錄

[版本歷程記錄] 索引標籤可以顯示您的訂閱的變更歷程記錄，以及變更的使用者和日期。如果訂閱失敗或運作不正確，則版本歷程記錄可協助確定原因。

The screenshot shows the vRealize Automation Cloud Assembly interface. The navigation sidebar on the left has tabs for '部署' (Deploy), '藍圖' (Blueprints), '基礎結構' (Infrastructure), '擴充性' (Extensibility), and '市集' (Marketplace). The '訂閱' (Subscriptions) tab is selected, and the '版本歷程記錄' (Version History) sub-tab is active. The main content area displays a list of subscription events with columns for time, user, and details. A JSON object is shown on the right, representing the subscription data.

Navigation tabs: 部署, 藍圖, 基礎結構, 擴充性, 市集

Left sidebar: 事件, 訂閱, 程式庫, 事件主題, 動作, 工作流程, 活動, 動作執行, 工作流程執行

Header: Test subscription - 版本歷程記錄

Subscription events table:

Time	User	Details
20/01/13 下午3:09 - 目前	rishi@vmware.com	
20/01/13 下午3:09	rishi@vmware.com	
20/01/13 下午3:08	rishi@vmware.com	

Subscription details (JSON):

```
{
  "id": "sub_1578899317099",
  "type": "RUNNABLE",
  "eventTopicId": "kubernetes.cluster.allocation.pre",
  "name": "Test subscription",
  "orgId": "9d9648a7-115a-4a06-a613-a0c4077469f7",
  "ownerId": "rishi@vmware.com",
  "subscriberId": "abx-svc",
  "blocking": false,
  "description": "",
  "criteria": "",
  "constraints": {
    "projectId": null
  },
  "timeout": 0,
  "broadcast": false,
}
```

從**訂閱索引標籤**開啟您的訂閱。

2

若要檢視版本歷程記錄，請按一下**版本歷程記錄**。

3

您可以按一下每個變更項目，以檢視與變更相關聯的訂閱代碼。



# 管理 vRealize Automation Cloud Assembly 部署

7

做為 vRealize Automation Cloud Assembly 藍圖開發人員，您可以使用 [部署] 索引標籤來管理您的部署。您可以對失敗的佈建程序進行疑難排解，進行變更，並銷毀未使用的部署。

部署是已佈建的藍圖執行個體。[部署] 索引標籤會顯示成功及失敗的部署。您可以使用此頁面管理成功的部署，或開始對任何失敗的要求進行疑難排解。

## 使用部署卡

您可以使用卡清單找到並管理您的部署。您可以篩選或搜尋特定部署，然後在這些部署上執行動作。

- 1 根據屬性篩選您的申請。
- 2 根據關鍵字或申請者搜尋部署。
- 3 按時間或名稱對清單進行排序。
- 4 在部署上執行部署層級的動作，包括刪除未使用的部署以回收資源。

您也可以查看部署成本、到期日期和狀態。

部署 10 個項目，共 43 個

1 2 3 4

名稱	狀態	專案	請求者	資源	建立於	到期	動作
测试部署ADmac...	無說明	專案	测试添加AD canl@vmw...	1 資源	建立於 1 小時前	到期 10 天内	動作
WP-POR1	無說明	專案	wordpress... pmartini@...	0 資源	建立於 2 小時前	到期 10 天内	動作
111	無說明	專案	wordpress... pmartini@...	0 資源	建立於 2 小時前	到期 10 天内	動作
部署AD机	無說明	專案	测试添加AD canl@vmw...	0 資源	建立於 2 小時前	到期 10 天内	動作
部署AD机器	無說明	專案	测试添加AD canl@vmw...	1 資源	建立於 4 小時前	到期 10 天内	動作

本章節討論下列主題：

- 如何監控 vRealize Automation Cloud Assembly 中的作用中部署
- 如果 vRealize Automation Cloud Assembly 部署失敗，該如何解決
- 如何管理已完成 vRealize Automation Cloud Assembly 部署的生命週期
- 您可以對 vRealize Automation Cloud Assembly 部署執行哪些動作

## 如何監控 vRealize Automation Cloud Assembly 中的作用中部署

部署 vRealize Automation Cloud Assembly 藍圖後，您可以監控要求，以確保資源已佈建且正在執行。從部署卡開始，您可以驗證資源的佈建。接著，您可以檢查部署詳細資料。

### 程序

- 1 如果需要，請按一下**部署**，並使用篩選和搜尋找到進行中的部署卡。
- 2 檢閱卡狀態。

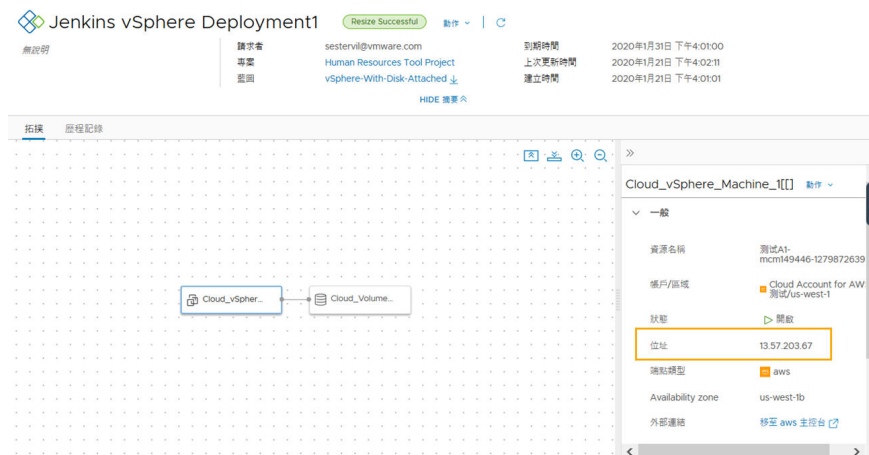
如果您的部署正在進行中，則程序列會指示剩餘工作數目。如果部署成功完成，則卡會顯示有關部署的



基本詳細資料。

- 3 若要判定資源的部署位置，請按一下部署名稱，然後在 [拓撲] 頁面上檢閱詳細資料。

您可能需要主要元件的 IP 位址。按一下每個元件時，請注意所提供的資訊是專門針對該元件的。在此範例中，會反白顯示 IP 位址。



外部連結的可用性取決於雲端提供者。如果可用，您必須具有該提供者的認證才能存取元件。

### 後續步驟

- 您可以對部署進行變更。請參閱[如何管理已完成 vRealize Automation Cloud Assembly 部署的生命週期](#)。
- 如果部署失敗，請參閱[如果 vRealize Automation Cloud Assembly 部署失敗，該如何解決](#)。

## 如果 vRealize Automation Cloud Assembly 部署失敗，該如何解決

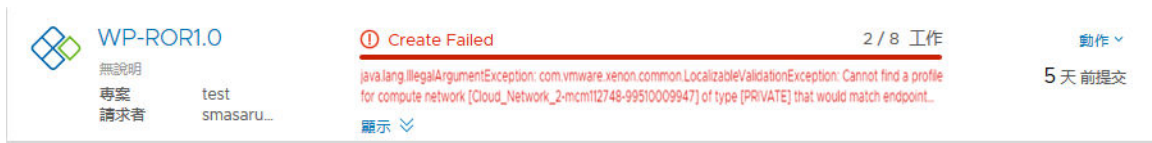
您的部署要求可能會因多種原因失敗。這可能是由於網路流量、目標雲端提供者上的資源不足或部署規格有缺陷所造成的。或者，部署成功，但似乎不起作用。您可以使用 vRealize Automation Cloud Assembly 來檢查您的部署、檢閱任何錯誤訊息，以及判定問題是否為環境、要求的工作負載規格或其他。

您可以使用此工作流程開始調查。該程序可能會表明失敗是由暫時性環境問題所致。在確認條件已經改善後重新部署申請可解決此類問題。在其他情況下，調查可能需要您詳細檢查其他區域。

您能夠以專案成員身分檢閱 vRealize Automation Cloud Assembly 中的要求詳細資料。

### 程序

- 若要判斷您的要求是否失敗，請按一下**部署索引**標籤，然後找到部署卡。



該卡指示了失敗的部署。

- 檢閱錯誤訊息。
- 如需詳細資訊，請按一下部署名稱以瞭解部署詳細資料。

- 在 [部署詳細資料] 頁面上，按一下**歷程記錄索引**標籤。



- 檢閱事件樹狀結構以查看佈建程序失敗的位置。在修改部署但變更失敗時，此樹狀結構非常有用。此樹狀結構也會顯示執行部署動作的時間。您可以使用此樹狀結構對失敗的變更進行疑難排解。
- 詳細資料**提供了更詳細的錯誤訊息版本。
- 如果要求的项目是 vRealize Automation Cloud Assembly 藍圖，則訊息右側的連結會開啟 vRealize Automation Cloud Assembly，以便您可以查看**要求詳細資料**。

### 3 要求詳細資料提供失敗元件的佈建工作流程，以便您可以調查問題。



- 檢閱錯誤訊息。
- 您可以開啟**開發模式**，在簡單的佈建工作流程和更詳細的流程圖之間切換。
- 按一下該卡以檢閱部署指令碼。

### 4 解決錯誤並重新部署藍圖。

這些錯誤可能存在於藍圖建構中，也可能與基礎結構的設定方式相關。

#### 後續步驟

解決錯誤並部署藍圖後，您可以在 [要求詳細資料] 中看到類似下列範例的資訊。若要查看要求詳細資料，請選取**基礎結構 > 活動 > 要求**。



## 如何管理已完成 vRealize Automation Cloud Assembly 部署的生命週期

佈建並執行部署後，您可以執行多個動作來管理部署。生命週期管理可包括開啟電源或關閉電源、調整部署大小，以及刪除部署。您也可以針對個別元件執行各種動作，以對其進行管理。

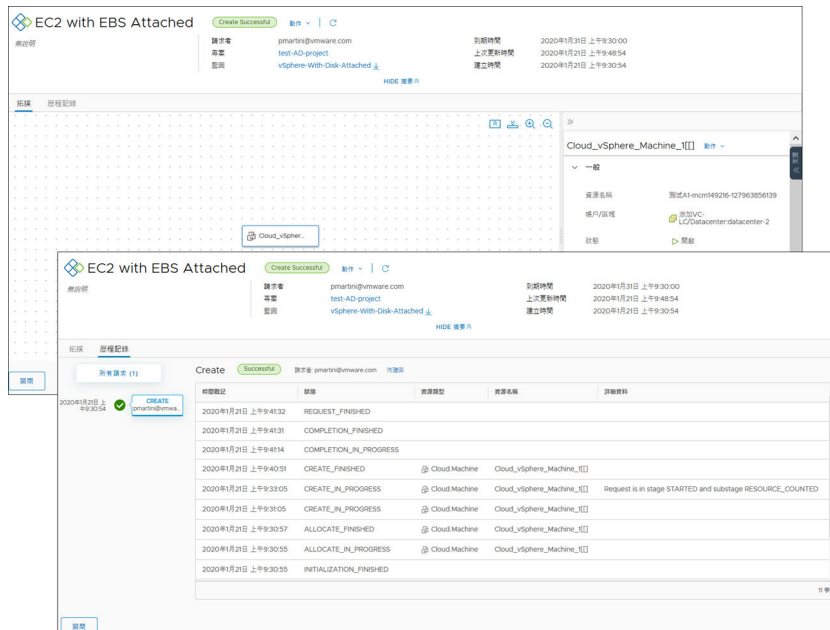
### 程序

- 1 按一下**部署**，並找到您的部署。
- 2 若要存取部署詳細資料，請按一下部署名稱。

您可以使用 [拓撲] 索引標籤來視覺化部署結構和資源。

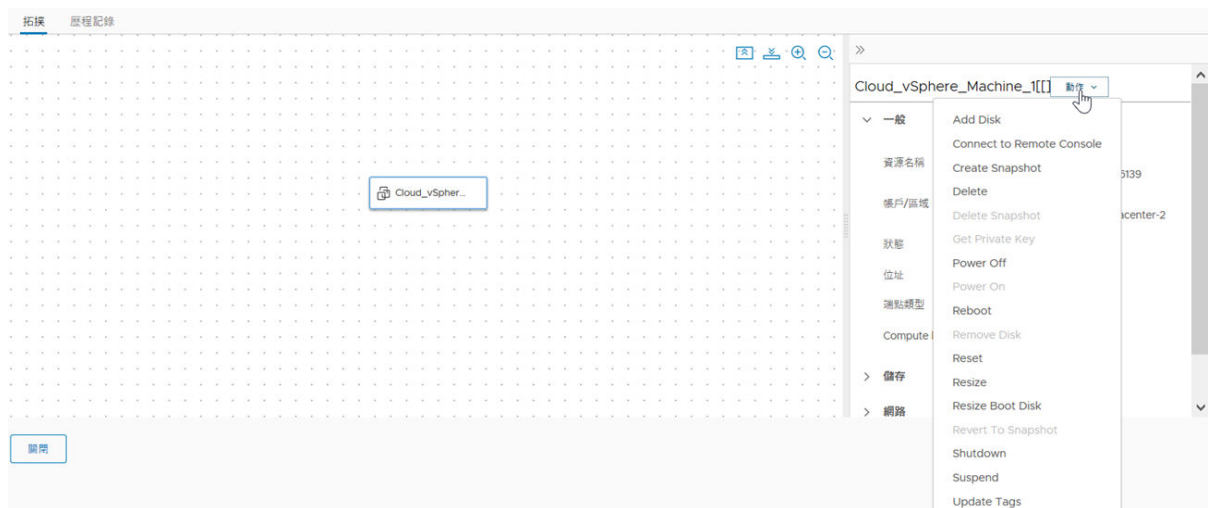
[歷程記錄] 索引標籤包括所有佈建事件，以及與部署要求項目後執行的動作相關的所有事件。如果佈建程序存在任何問題，[歷程記錄] 索引標籤事件將協助您疑難排解故障。

[成本] 索引標籤提供部分元件自部署以來的目前成本。



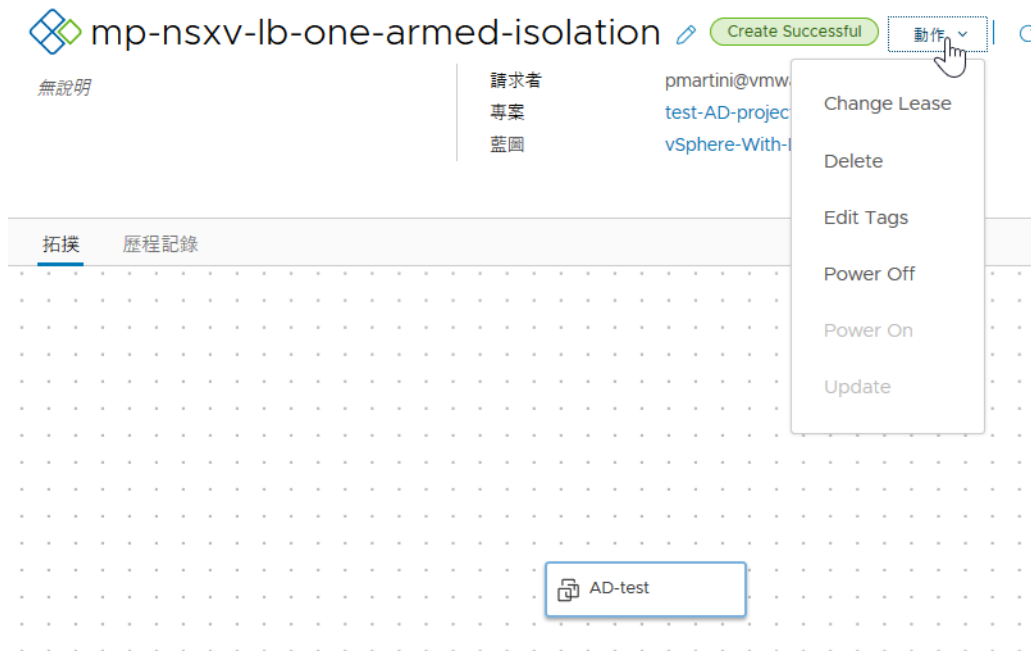
- 3 如果您判定部署的目前組態成本過高，並且想要調整元件大小，您可以在 [拓撲] 頁面上選取此元件，然後在 [元件] 頁面上選取動作 > 調整大小。

可用的動作取決於元件、雲端帳戶和您的權限。



- 4 做為開發生命週期的一部分，不再需要您的其中一個部署。若要移除部署並回收資源，請選取動作 > 刪除。

可用的動作取決於部署的狀態。



### 後續步驟

若要進一步瞭解可能的動作，請參閱[您可以對 vRealize Automation Cloud Assembly 部署執行哪些動作](#)。

## 您可以對 vRealize Automation Cloud Assembly 部署執行哪些動作

部署藍圖後，您可以在 vRealize Automation Cloud Assembly 中執行動作來管理資源。可用的動作取決於資源類型，以及動作在特定雲端帳戶或整合平台上是否受支援。

可用的動作也取決於您的管理員授權您執行的動作。

做為管理員或專案管理員，您可以在 vRealize Automation Service Broker 中設定第 2 天動作原則。請參閱 [〈如何授權取用者使用 Service Broker 第 2 天動作原則〉](#)

表 7-1. 可能動作的清單

動作	套用到這些資源類型	適用於這些雲端帳戶或整合	說明
新增磁碟	機器	<ul style="list-style-type: none"> <li>Amazon Web Services</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	將其他磁碟新增至現有虛擬機器。
變更租用	部署	<ul style="list-style-type: none"> <li>Amazon Web Services</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>變更租用到期日期和時間。</p> <p>租用到期時，系統會銷毀部署並回收資源。</p> <p>租用原則已在 vRealize Automation Service Broker 中設定。</p>

表 7-1. 可能動作的清單 (續)

動作	套用到這些資源類型	適用於這些雲端帳戶或整合	說明
連線至遠端主控台	機器	<ul style="list-style-type: none"> <li>VMware vSphere</li> </ul>	<p>在選取的機器上開啟遠端工作階段。</p> <p>檢閱成功連線的下列需求。</p> <ul style="list-style-type: none"> <li>做為部署取用者，請確認佈建的機器已開啟電源。</li> </ul>
建立快照	機器	<ul style="list-style-type: none"> <li>Google Cloud Platform</li> <li>VMware vSphere</li> </ul>	<p>建立虛擬機器的快照。</p> <p>如果 vSphere 中僅允許使用兩個快照並且已有兩個，則此命令僅在刪除一個快照後才可用。</p>
刪除	部署	<ul style="list-style-type: none"> <li>Amazon Web Services</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>銷毀部署。</p> <p>將刪除並回收所有資源。</p> <p>如果刪除失敗，您可以在部署上再次執行刪除動作。在第二次嘗試期間，您可以選取<b>忽略刪除失敗</b>。如果您選取此選項，則會刪除部署，但可能不會回收資源。您應檢查已佈建部署的系統，以確保移除所有資源。如果尚未全部移除，則必須手動刪除這些系統上的剩餘資源。</p>
	機器和負載平衡器	<ul style="list-style-type: none"> <li>Amazon Web Services</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>從部署中刪除機器或負載平衡器。此動作可能會導致部署無法使用。</p>
刪除快照	機器	<ul style="list-style-type: none"> <li>VMware vSphere</li> <li>Google Cloud Platform</li> </ul>	<p>刪除虛擬機器的快照。</p>
編輯標籤	部署	<ul style="list-style-type: none"> <li>Amazon Web Services</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>新增或修改套用至個別部署資源的資源標籤。</p>
關閉電源	部署	<ul style="list-style-type: none"> <li>Amazon Web Services</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>關閉部署的電源，但不關閉客體作業系統。</p>
	機器	<ul style="list-style-type: none"> <li>Amazon Web Services</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>關閉機器的電源，但不關閉客體作業系統。</p>
開啟電源	部署	<ul style="list-style-type: none"> <li>Amazon Web Services</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>開啟部署的電源。如果資源暫停，則一般作業會從機器暫停的時間繼續。</p>
	機器	<ul style="list-style-type: none"> <li>Amazon Web Services</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>開啟機器的電源。如果機器暫停，則一般作業會從機器暫停的時間點繼續。</p>
重新開機	機器	<ul style="list-style-type: none"> <li>Amazon Web Services</li> <li>VMware vSphere</li> </ul>	<p>將虛擬機器上的客體作業系統重新開機。</p> <p>對於 vSphere 機器，必須在機器上安裝 VMware Tools，才能使用此動作。</p>



表 7-1. 可能動作的清單 (續)

動作	套用到這些資源類型	適用於這些雲端帳戶或整合	說明
重新設定	負載平衡器	<ul style="list-style-type: none"> <li>■ Amazon Web Services</li> <li>■ Microsoft Azure</li> <li>■ Google Cloud Platform</li> <li>■ VMware vSphere</li> </ul>	變更負載平衡器通訊協定、連接埠、健全狀況組態和成員集區設定。
移除磁碟	機器	<ul style="list-style-type: none"> <li>■ Amazon Web Services</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	從現有虛擬機器中移除磁碟。
重設	機器	<ul style="list-style-type: none"> <li>■ Amazon Web Services</li> <li>■ Google Cloud Platform</li> <li>■ VMware vSphere</li> </ul>	強制虛擬機器重新啟動，但不關閉客體作業系統。
調整大小	機器	<ul style="list-style-type: none"> <li>■ Amazon Web Services</li> <li>■ Microsoft Azure</li> <li>■ Google Cloud Platform</li> <li>■ VMware vSphere</li> </ul>	增加或減少虛擬機器的 CPU 和記憶體。
調整開機磁碟大小	機器	<ul style="list-style-type: none"> <li>■ Amazon Web Services</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	增加或減少開機磁碟媒體的大小。
調整磁碟大小	儲存區磁碟	<ul style="list-style-type: none"> <li>■ Amazon Web Services</li> <li>■ Google Cloud Platform</li> </ul>	增加儲存區磁碟的容量。
重新啟動	機器	<ul style="list-style-type: none"> <li>■ Microsoft Azure</li> </ul>	關閉後重新啟動執行中的機器。
還原為快照	機器	<ul style="list-style-type: none"> <li>■ Google Cloud Platform</li> <li>■ VMware vSphere</li> </ul>	還原到機器的先前快照。 您必須擁有現有快照，才能使用此動作。
執行 Puppet 工作	受管理資源	<ul style="list-style-type: none"> <li>■ Puppet Enterprise</li> </ul>	在部署中的機器上執行所選工作。 工作在 Puppet 執行個體中定義。您必須能夠識別工作並提供輸入參數。
關閉	機器	<ul style="list-style-type: none"> <li>■ VMware vSphere</li> </ul>	關閉客體作業系統，然後關閉機器電源。必須在機器上安裝 VMware Tools，才能使用此動作。
暫停	機器	<ul style="list-style-type: none"> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	暫停機器以使其無法使用，並且不會耗用其正在使用之儲存區以外的任何系統資源。
更新	部署	<ul style="list-style-type: none"> <li>■ Amazon Web Services</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	根據輸入參數變更部署。
更新標籤	機器和磁碟	<ul style="list-style-type: none"> <li>■ Amazon Web Services</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	新增、修改或刪除套用到個別資源的標籤。