Red Hat OpenShift with NetApp

NetApp Solutions

NetApp
April 12, 2024

目錄

NVA-1160：採用NetApp的Red Hat OpenShift

NetApp公司Alan Cowles和NIKhil M Kulkarni

本參考文件提供Red Hat OpenShift解決方案的部署驗證、此解決方案是透過安裝程式佈建基礎架構（IPI）、部署於多種不同的資料中心環境、並已通過NetApp驗證。它也詳細說明了與NetApp儲存系統的儲存整合、運用Astra Trident儲存協調程式來管理持續儲存設備。最後、我們會探索並記錄許多解決方案驗證和實際使用案例。

使用案例

Red Hat OpenShift with NetApp解決方案的架構設計、可為下列使用案例的客戶提供卓越價值：

- 使用裸機、Red Hat OpenStack平台、Red Hat虛擬化及VMware vSphere上的IPI（安裝程式佈建基礎架構）、輕鬆部署及管理部署的Red Hat OpenShift。
- 結合企業容器和虛擬化工作負載的強大功能、搭配幾乎部署在OSP、RHV或vSphere上的Red Hat OpenShift、或是以OpenShift虛擬化技術部署在裸機上。
- 真正的組態與使用案例、強調Red Hat OpenShift的功能、適用於Kubernetes的開放原始碼儲存協調程式NetApp儲存設備和Astra Trident。

商業價值

企業逐漸採用DevOps實務來建立新產品、縮短發行週期、並快速新增新功能。由於容器和微服務的本質天生敏捷、因此在支援DevOps實務做法上扮演著重要角色。然而、在企業環境中以正式作業規模實作DevOps、卻帶來了自身的挑戰、並對基礎架構提出特定要求、例如：

- 堆疊中所有層級的高可用度
- 易於部署的程序
- 不中斷營運與升級
- API導向且可程式化的基礎架構、可跟上微服務敏捷度的腳步
- 多租戶共享、效能保證
- 能夠同時執行虛擬化與容器化的工作負載
- 能夠根據工作負載需求獨立擴充基礎架構

Red Hat OpenShift with NetApp瞭解這些挑戰、並提供解決方案、在客戶選擇的資料中心環境中實作完整自動化的RedHat OpenShift IPI部署、協助解決每個疑慮。

技術總覽

Red Hat OpenShift with NetApp解決方案包含下列主要元件：

Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform是完全受支援的企業Kubernetes平台。Red Hat針對開放原始碼Kubernetes進行了多項增強功能、以提供一個應用程式平台、其中所有元件均已完全整合、可用來建置、部

署及管理容器化應用程式。

如需詳細資訊、請造訪OpenShift網站 "請按這裡"。

NetApp儲存系統

NetApp擁有多種儲存系統、最適合用於企業資料中心和混合雲部署。NetApp產品組合包括NetApp ONTAP的NetApp功能、NetApp Element 功能與NetApp E系列儲存系統、所有這些系統都能為容器化應用程式提供持續儲存。

如需詳細資訊、請參閱NetApp網站 "請按這裡"。

NetApp儲存整合

NetApp Astra Control Center提供一組豐富的儲存設備與應用程式感知資料管理服務、可處理有狀態的Kubernetes工作負載、部署於內部環境、並採用值得信賴的NetApp資料保護技術。

如需詳細資訊、請造訪NetApp Astra網站 "請按這裡"。

Astra Trident是開放原始碼且完全支援的儲存協調工具、適用於容器和Kubernetes配送、包括Red Hat OpenShift。

如需詳細資訊、請造訪Astra Trident網站 "請按這裡"。

進階組態選項

本節專供實際使用者在將此解決方案部署至正式作業環境時可能需要執行的自訂作業、例如建立專用的私有映像登錄或部署自訂負載平衡器執行個體。

已驗證版本的目前支援對照表

技術	目的	軟體版本
NetApp ONTAP	儲存設備	9.8、9.9.1
NetApp Element	儲存設備	12.3.
NetApp Astra控制中心	應用程式感知資料管理	212.60
NetApp Astra Trident	儲存協調	22.01.0
Red Hat OpenShift	容器協調	4.6 EUS、4.7、4.8
Red Hat OpenStack平台	私有雲基礎架構	16.1
Red Hat虛擬化	資料中心虛擬化	4.4
VMware vSphere	資料中心虛擬化	6.7U3

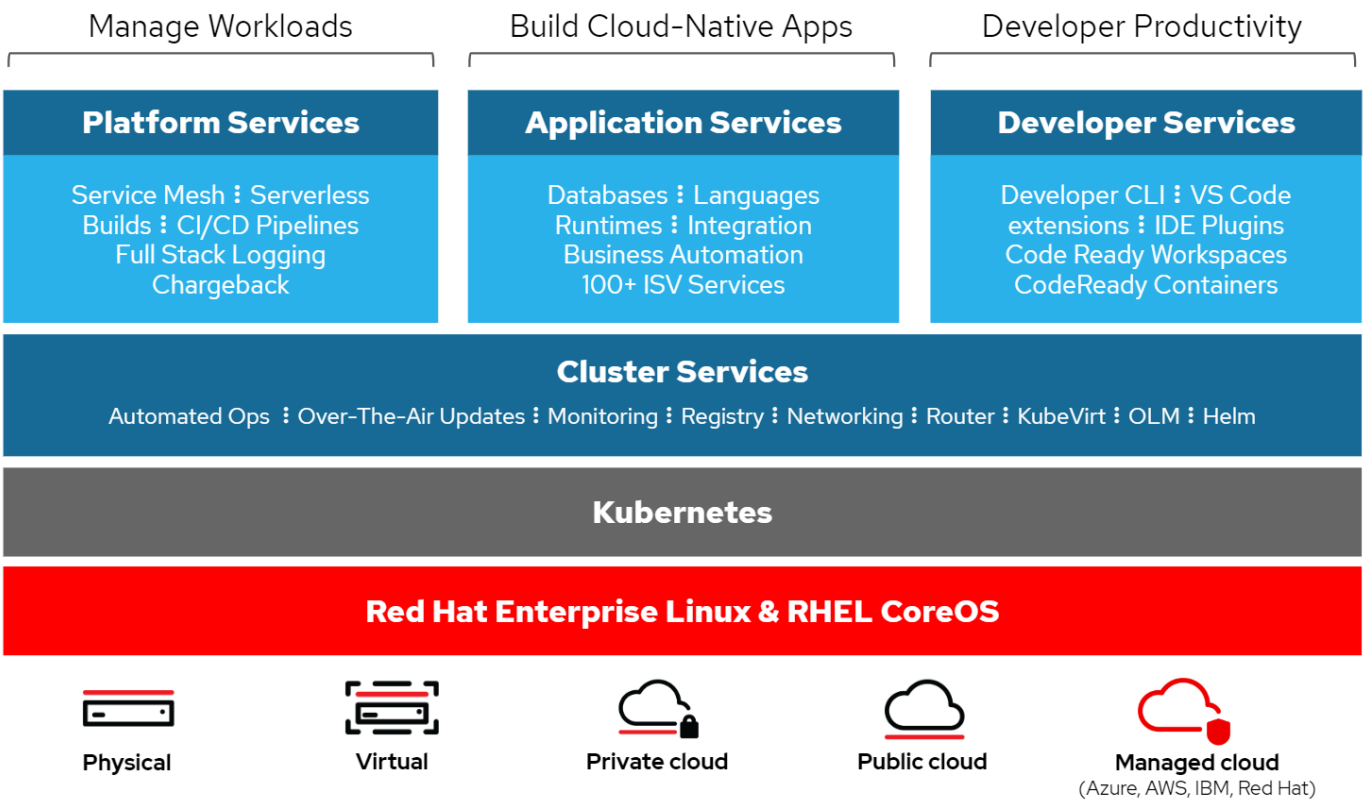
OpenShift總覽

Red Hat OpenShift Container Platform將開發與IT作業整合在單一平台上、可在內部部署與混合雲基礎架構之間一致地建置、部署及管理應用程式。Red Hat OpenShift以開放原始

碼創新技術和產業標準為基礎、包括Kubernetes和Red Hat Enterprise Linux CoreOS、這是全球領先業界的企業級Linux套裝作業系統、專為以容器為基礎的工作負載所設計。OpenShift是Cloud Native Computing Foundation（CNCF）認證Kubernetes方案的一部分、提供容器工作負載的可攜性與互通性。

Red Hat OpenShift提供下列功能：

- * 自助服務資源配置 * 開發人員可以利用他們最常用的工具、快速輕鬆地根據需求建立應用程式、同時讓作業完全掌控整個環境。
- * 持續儲存 * OpenShift Container Platform 支援持續儲存、可讓您同時執行有狀態的應用程式和雲端原生無狀態應用程式。
- * 持續整合與持續開發（ CI/CD ） * 此原始碼平台可大規模管理建置與部署映像。
- * 開放原始碼標準 * 除了其他開放原始碼技術之外、這些標準還納入開放容器計畫（ OCI ）和 Kubernetes 、以進行容器協調。您不受限於特定廠商的技術或業務藍圖。
- * CI/CD 管線 * OpenShift 提供立即可用的 CI/CD 管線支援、讓開發團隊能夠自動化應用程式交付程序的每個步驟、並確保在對應用程式程式碼或組態所做的每一項變更上執行。
- * 角色型存取控制（ RBAC ） * 此功能提供團隊與使用者追蹤功能、協助組織大型開發人員群組。
- * 自動化建置與部署 * OpenShift 可讓開發人員選擇建置容器化應用程式、或是讓平台從應用程式原始碼或甚至是二進位檔建置容器。然後、平台會根據應用程式定義的特性、在基礎架構上自動部署這些應用程式。例如、應該配置的資源數量、以及應該部署在基礎架構上的何處、以符合第三方授權的要求。
- * 一致的環境 * OpenShift 可確保為開發人員及整個應用程式生命週期所配置的環境、無論是作業系統、程式庫、執行階段版本（例如 Java 執行時期）、 甚至是使用中的應用程式執行階段（例如、 tomcat ）、以移除不一致環境所產生的風險。
- * 組態管理 * 平台內建組態與敏感資料管理功能、無論使用哪些技術來建置應用程式、或是使用何種環境、都能確保應用程式的應用程式組態一致且不受環境限制
已部署。
- * 應用程式記錄和指標。 * 快速意見反應是應用程式開發的重要層面。OpenShift整合式監控與記錄管理功能可立即提供指標給開發人員、讓他們瞭解應用程式在變更過程中的運作方式、並能在應用程式生命週期中盡快修正問題。
- * 安全性與容器目錄 * OpenShift 提供多租戶服務、並使用安全增強型 Linux （ SELinux ）、 cgroups 和安全運算模式（ seccomp ）來隔離及保護容器、保護使用者免於執行有害程式碼。它也透過TLS憑證為各種子系統提供加密功能、並可存取經過掃描和評分的Red Hat認證容器（access.redhat.com/containers）、特別強調安全性、為終端使用者提供認證、信任和安全的應用程式容器。



Red Hat OpenShift的部署方法

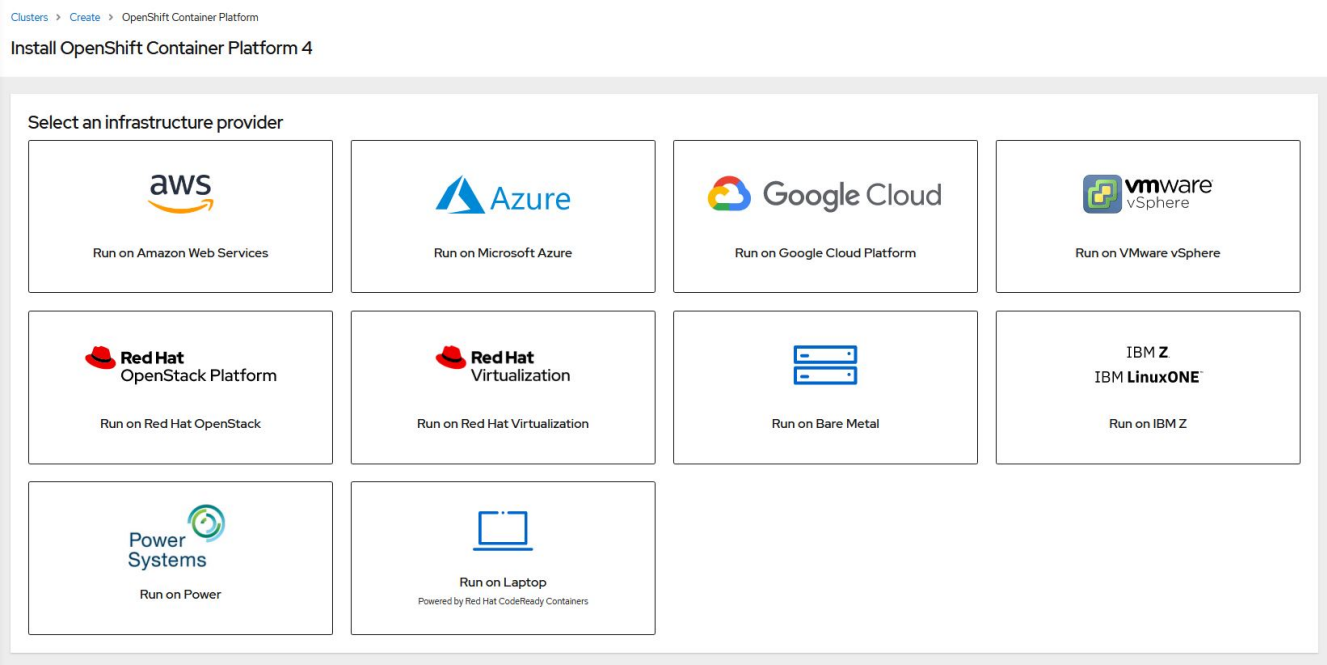
從Red Hat OpenShift 4開始、OpenShift的部署方法包括使用使用者資源配置基礎架構（UPI）進行手動部署、以進行高度自訂的部署、或使用安裝程式資源配置基礎架構（IPI）進行全自動部署。

在大多數情況下、 IPI 安裝方法是首選的方法、因為它允許快速部署 OpenShift 叢集以用於開發、測試和正式作業環境。

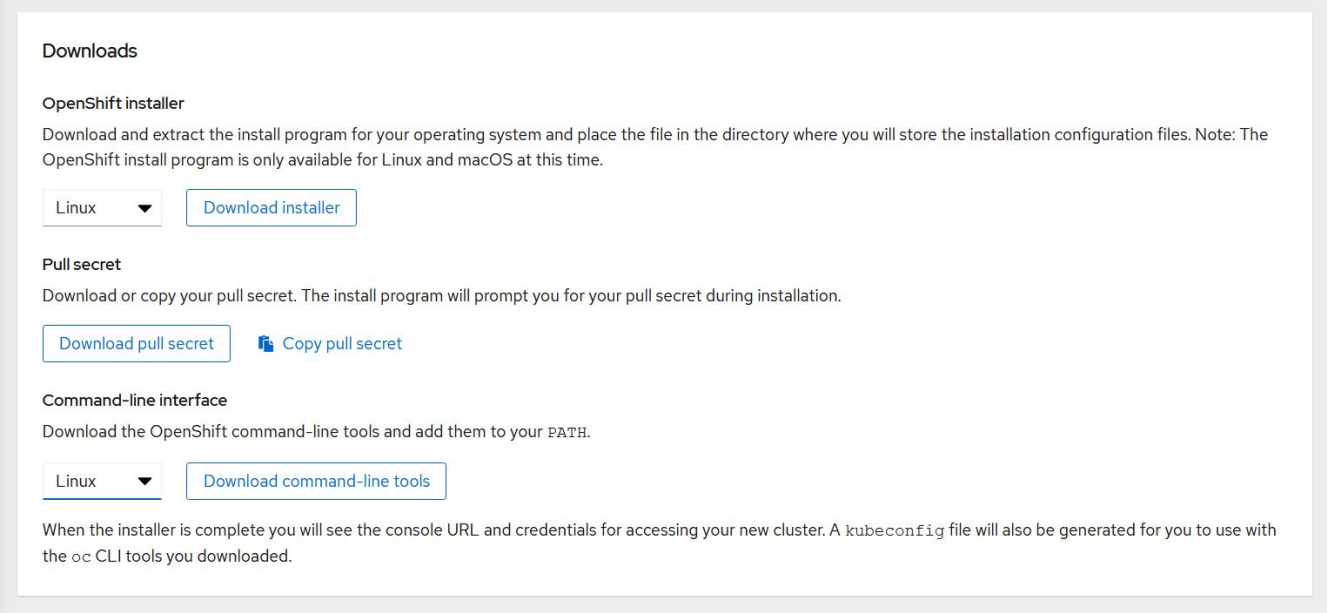
Red Hat OpenShift的IPI安裝

安裝程式佈建的OpenShift基礎架構（IPI）部署包括下列高層級步驟：

1. 請造訪Red Hat OpenShift "網站" 並使用SSO認證登入。
2. 選取您要部署Red Hat OpenShift的環境。



3. 在下一個畫面下載安裝程式、獨特的Pull Secret和用於管理的CLI工具。



4. 請依照 "安裝說明" 由Red Hat提供、可部署至您選擇的環境。

NetApp已驗證OpenShift部署

NetApp已在實驗室中使用安裝程式佈建基礎架構（IPI）部署方法、在下列每個資料中心環境中測試並驗證Red Hat OpenShift的部署：

- "裸機上的OpenShift"
- "Red Hat OpenStack平台上的OpenShift"
- "Red Hat虛擬化的OpenShift"
- "VMware vSphere上的OpenShift"

裸機上的**OpenShift**

裸機上的OpenShift可在市售伺服器上自動部署OpenShift Container Platform。

裸機上的OpenShift類似於OpenShift的虛擬部署、可輕鬆部署、快速配置及擴充OpenShift叢集、同時支援尚未準備好容器化的應用程式虛擬化工作負載。在裸機上部署之後、除了OpenShift環境之外、您不需要額外的管理成本來管理主機Hypervisor環境。透過直接部署在裸機伺服器上、您也可以減少主機與OpenShift環境之間共用資源的實體負荷限制。

裸機上的**OpenShift**提供下列功能：

- * IPI 或輔助安裝程式部署 * 透過安裝程式佈建基礎架構（ IPI ）在裸機伺服器上部署的 OpenShift 叢集、客戶可以直接在市售伺服器上部署功能廣泛、易於擴充的 OpenShift 環境、而無需管理 Hypervisor 層。
- * 精簡型叢集設計 * 為了將硬體需求降至最低、裸機上的 OpenShift 可讓使用者僅部署 3 個節點的叢集、讓 OpenShift 控制平面節點也能做為工作節點和主機容器。
- * OpenShift 虛擬化 * OpenShift 可以使用 OpenShift 虛擬化在容器內執行虛擬機器。此容器原生虛擬化可在容器內執行KVM Hypervisor、並附加持續磁碟區以供VM儲存。
- * AI / ML 最佳化基礎架構 * 將 Kubeflow 等應用程式整合至 OpenShift 環境、並運用 OpenShift 進階排程技術、為機器學習應用程式部署 Kubeflflow 。

網路設計

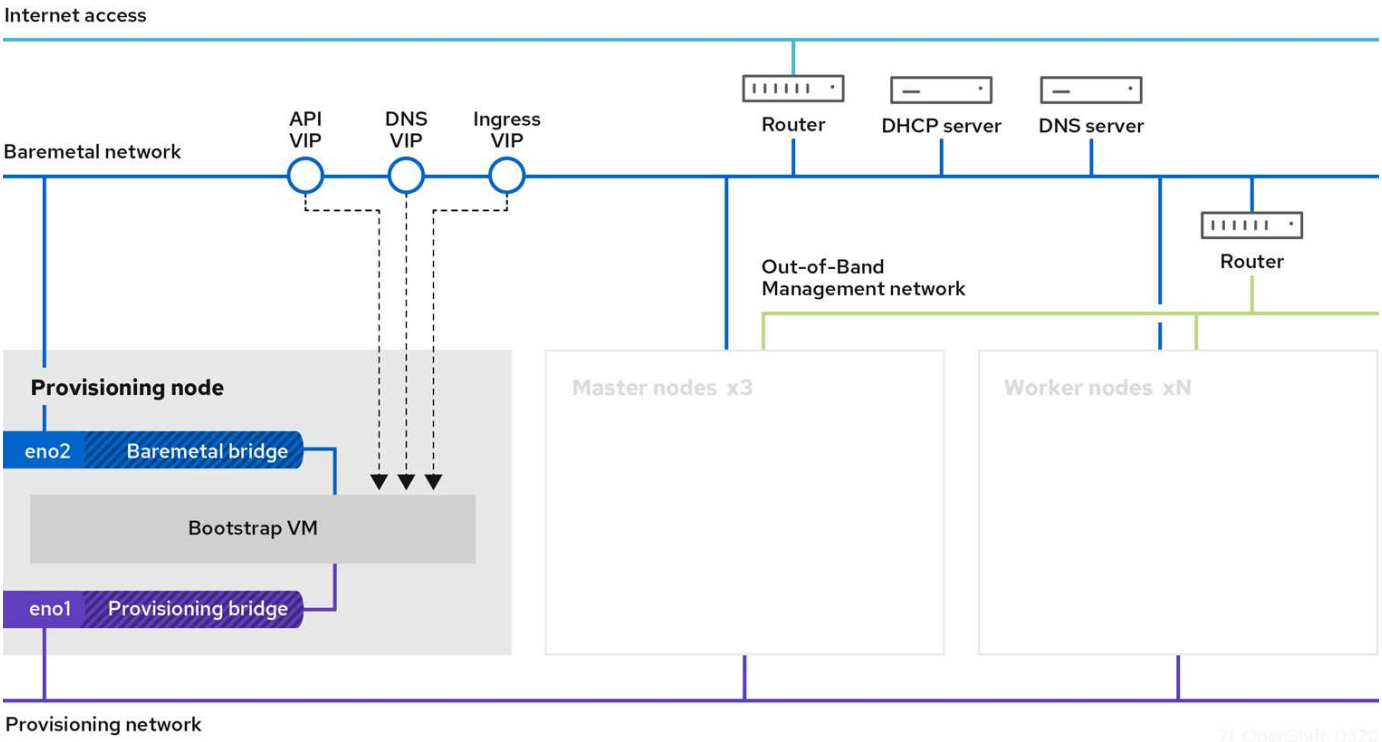
Red Hat OpenShift on NetApp解決方案使用兩個資料交換器、以25Gbps的速率提供主要資料連線能力。它也使用兩個管理交換器、以1Gbps的速率提供連線能力、用於儲存節點的頻內管理、以及IPMI功能的頻外管理。

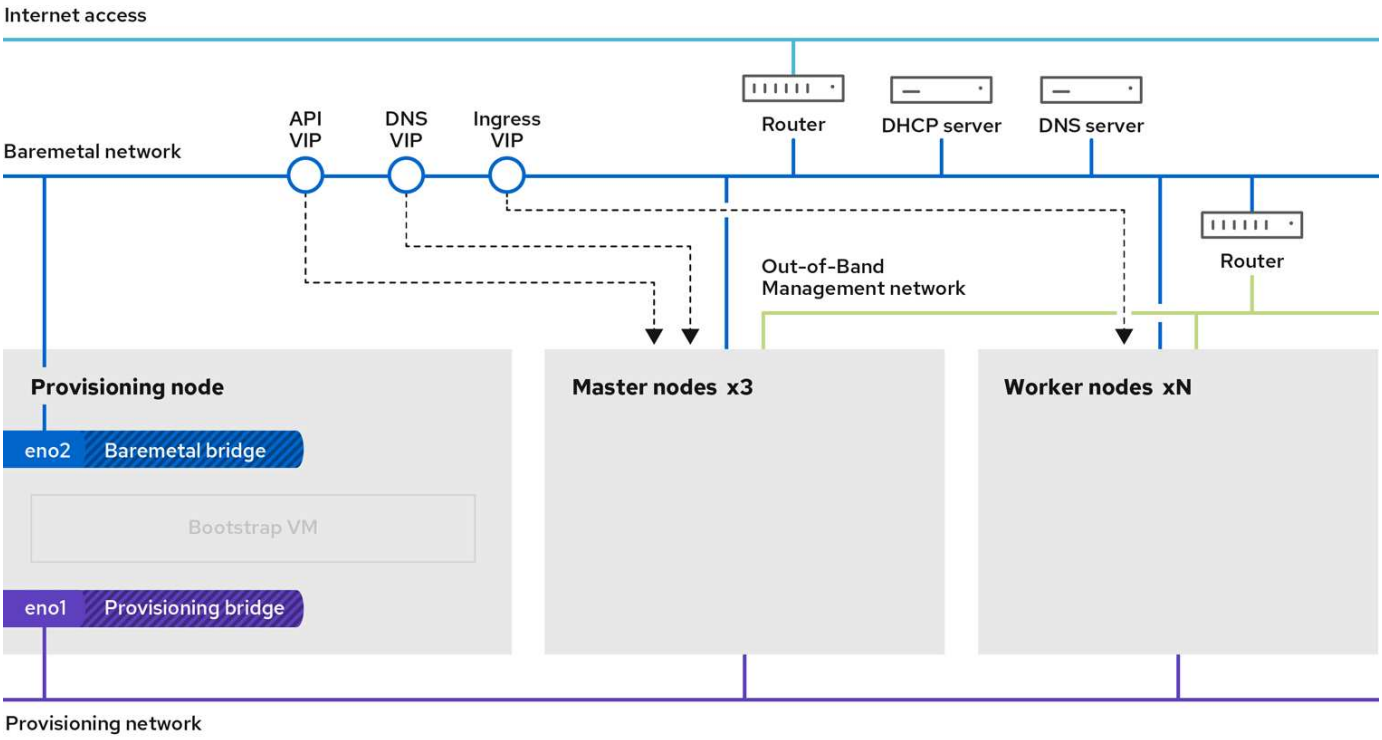
對於OpenShift裸機IPI部署、您必須建立資源配置程式節點、這是一台Red Hat Enterprise Linux 8機器、必須將網路介面連接至不同的網路。

- * 資源配置網路 * 此網路用於開機裸機節點、並安裝必要的映像和套件以部署 OpenShift 叢集。
- * 裸機網路 * 此網路用於叢集部署後的公開通訊。

為了設定資源配置工具節點、客戶建立橋接介面、讓流量能在節點本身和為部署目的而配置的Bootstrap VM上正確路由傳送。部署叢集之後、API和入口VIP位址會從啟動節點移轉至新部署的叢集。

下列影像說明IPI部署期間和部署完成後的環境。





VLAN需求

Red Hat OpenShift with NetApp解決方案的設計是使用虛擬區域網路（VLAN）、以邏輯方式將網路流量分隔為不同用途。

VLAN	目的	VLAN ID
頻外管理網路	裸機節點和IPMI的管理	16
裸機網路	叢集可用後、OpenShift服務的網路	181
資源配置網路	透過IPI開機及安裝裸機節點的網路	3485

> 雖然這些網路中的每個都是以VLAN分隔、但每個實體連接埠都必須在存取模式中設定、並指派主要VLAN、因為在執行PXE開機順序時、無法傳遞VLAN標記。

網路基礎架構支援資源

在部署OpenShift Container平台之前、應先準備好下列基礎架構：

- 至少有一部DNS伺服器提供完整的主機名稱解析、可從頻內管理網路和VM網路存取。
- 至少有一部NTP伺服器可從頻內管理網路和VM網路存取。
- （可選）用於帶內管理網路和VM網路的傳出網際網路連線。

Red Hat OpenStack平台上的OpenShift

Red Hat OpenStack平台提供整合式基礎、可建立、部署及擴充安全可靠的私有OpenStack雲端。

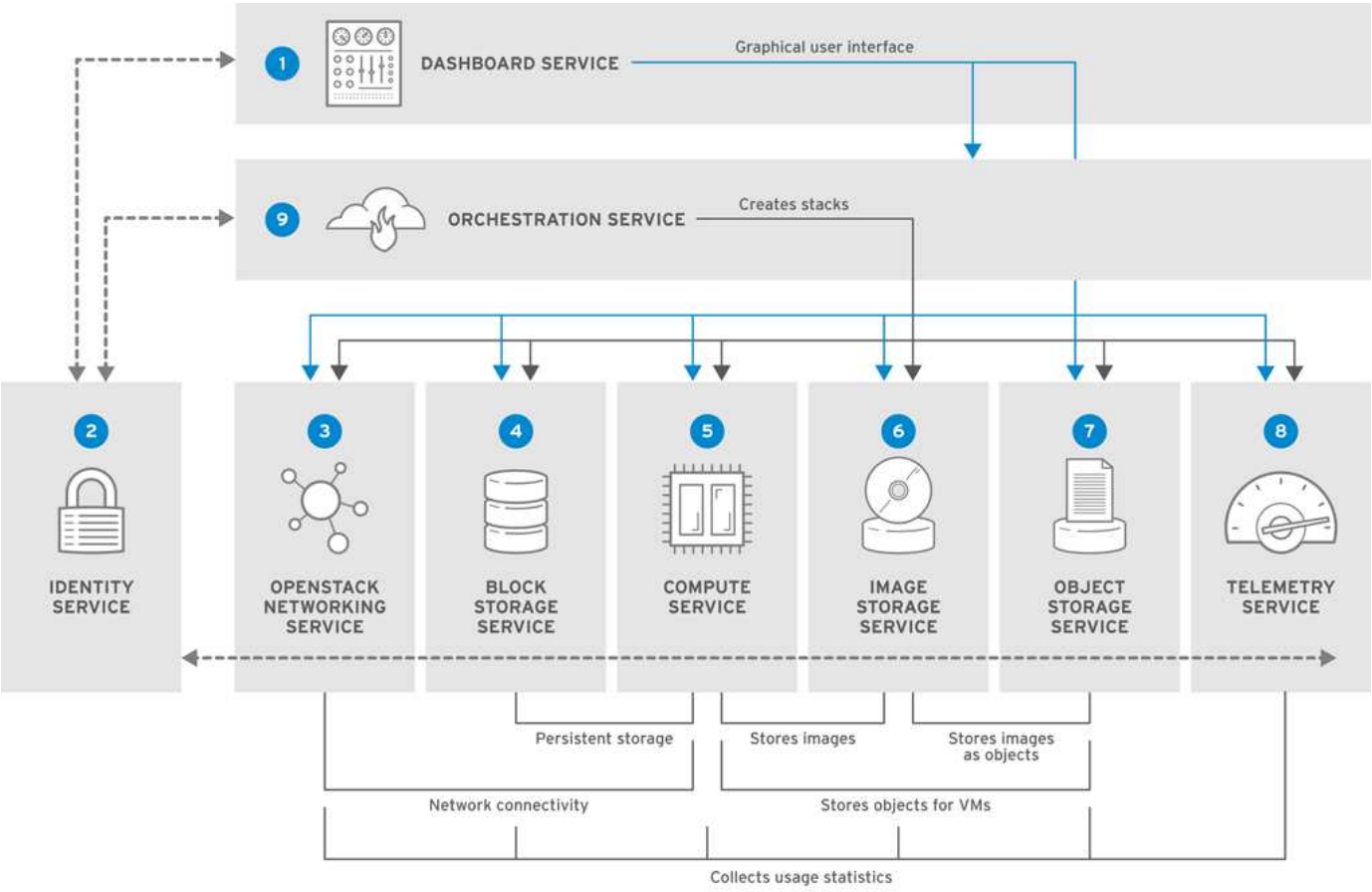
OSP是一種基礎架構即服務（IaaS）雲端、由管理運算、儲存和網路資源的控制服務集合來實作。此環境是使用網路介面來管理、讓系統管理員和使用者能夠控制、配置及自動化OpenStack資源。此外、OpenStack基礎架構也可透過廣泛的命令列介面和API、為系統管理員和終端使用者提供完整的自動化功能。

OpenStack專案是快速開發的社群專案、每六個月提供更新版本。Red Hat OpenStack Platform一開始會隨著每個上游版本一起發佈新版本、並為每個第三版提供長期支援、以跟上這個發行週期。最近、隨著OSP 16.0版（以OpenStack火車為基礎）的推出、Red Hat選擇不跟上發行版本的腳步、而是將新功能支援至次發行版本。最新版本為Red Hat OpenStack Platform 16.1、其中包含使用實例和上游版本的支援進階功能。

如需OSP的詳細資訊、請參閱 "Red Hat OpenStack Platform網站"。

OpenStack服務

OpenStack平台服務部署為容器、可將服務彼此隔離、並可輕鬆升級。OpenStack平台使用一套以Kolla建置及管理的容器。服務的部署是從Red Hat Custom Portal擷取容器映像來執行。這些服務容器是使用Podman命令來管理、並透過Red Hat OpenStack Director進行部署、設定及維護。



服務	專案名稱	說明
儀表板	Horizon	您用來管理OpenStack服務的網頁瀏覽器型儀表板。
身分識別	基礎概念	集中式服務、用於驗證和授權OpenStack服務、以及管理使用者、專案和角色。
OpenStack網路	中和	提供OpenStack服務介面之間的連線功能。
區塊儲存	資料廢止者	管理虛擬機器（VM）的持續區塊儲存Volume。
運算	Nova	管理並配置執行於運算節點上的VM。
映像	概覽	用於儲存VM映像和Volume快照等資源的登錄服務。
物件儲存	Swift	可讓使用者儲存及擷取檔案及任意資料。
遙測	Ceilometer	提供雲端資源使用量的測量結果。
協調	熱能	以範本為基礎的協調引擎、可支援自動建立資源堆疊。

網路設計

Red Hat OpenShift with NetApp解決方案使用兩個資料交換器、以25Gbps的速率提供主要資料連線能力。此外、它還使用兩個額外的管理交換器、以1Gbps的連線能力提供儲存節點的頻內管理、以及IPMI功能的頻外管理。

Red Hat OpenStack Director需要IPMI功能、才能使用具有諷刺意味的裸機資源配置服務來部署Red Hat OpenStack平台。

VLAN需求

Red Hat OpenShift with NetApp的設計、是為了使用虛擬區域網路（VLAN）、以邏輯方式將網路流量分隔為不同用途。此組態可擴充以滿足客戶需求、或進一步隔離特定的網路服務。下表列出在NetApp驗證解決方案時實作解決方案所需的VLAN。

VLAN	目的	VLAN ID
頻外管理網路	用於管理實體節點的網路、以及用於諷刺的IPMI服務。	16
儲存基礎架構	用於控制器節點的網路、可直接對應磁碟區以支援Swift等基礎架構服務。	201.
儲存資料夾	用於將區塊磁碟區直接對應及附加到環境中部署的虛擬執行個體的網路。	202.02
內部API	使用API通訊、RPC訊息和資料庫通訊、在OpenStack服務之間進行通訊所用的網路。	301.01
租戶	中子透過VXLAN的通道、為每個租戶提供自己的網路。網路流量會隔離在每個租戶網路內。每個租戶網路都有與其相關的IP子網路、而網路命名空間則表示多個租戶網路可以使用相同的位址範圍、而不會造成衝突。	302
儲存管理	OpenStack物件儲存設備（Swift）使用此網路、在參與的複本節點之間同步資料物件。Proxy服務是使用者要求與基礎儲存層之間的中介介面。Proxy會接收傳入要求、並找出必要的複本以擷取要求的資料。	303
PXE-	OpenStack Director提供的PXE開機是諷刺裸機資源配置服務的一部分、可協調OSP OverCloud的安裝。	3484.
外部	以開放式網路為基礎、裝載OpenStack儀表板（Horizon）進行圖形化管理、並允許公開API呼叫來管理OpenStack服務。	3485
頻內管理網路	提供系統管理功能的存取、例如SSH存取、DNS流量和網路時間傳輸協定（NTP）流量。此網路也可做為非控制器節點的閘道。	3486

網路基礎架構支援資源

在部署OpenShift Container Platform之前、應先準備好下列基礎架構：

- 至少有一部DNS伺服器提供完整的主機名稱解析。
- 至少三部NTP伺服器、可讓解決方案中的伺服器保持時間同步。
- （選用）OpenShift環境的傳出網際網路連線功能。

正式作業部署的最佳實務做法

本節列出組織在將此解決方案部署至正式作業環境之前、應考慮的幾項最佳實務做法。

將OpenShift部署至至少有三個運算節點的OSP私有雲

本文件所述的驗證架構、是部署三個OSP控制器節點和兩個OSP運算節點、提供最小的硬體部署、適合HA作業。此架構可確保容錯組態、讓兩個運算節點都能啟動虛擬執行個體、而已部署的VM則可在兩個Hypervisor之間移轉。

由於Red Hat OpenShift一開始會部署三個主節點、因此雙節點組態可能會導致至少兩個主節點佔用同一個節點、因此如果該特定節點無法使用、可能會導致OpenShift中斷。因此、部署至少三個OSP運算節點是Red Hat的最佳實務做法、如此一來、OpenShift主節點就能平均分散、解決方案就能獲得更高程度的容錯能力。

設定虛擬機器/主機關聯性

啟用VM/主機關聯性、即可在多個Hypervisor節點之間散佈OpenShift主機。

關聯性是一種定義一組VM和/或主機規則的方法、可決定VM是在同一主機上一起執行、還是在群組中的主機上執行、或是在不同的主機上執行。它會透過建立關聯群組來套用至VM、這些群組由一組相同的參數和條件的VM和/或主機組成。根據關聯群組中的VM是在同一主機或群組中的主機上執行、還是分別在不同主機上執行、關聯群組的參數可以定義正關聯性或負關聯性。在Red Hat OpenStack平台中、可以建立和強制執行主機關聯性和反關聯性規則、方法是建立伺服器群組並設定篩選器、以便Nova在伺服器群組中部署的執行個體部署在不同的運算節點上。

伺服器群組預設最多可管理10個虛擬執行個體的放置位置。您可以更新Nova的預設配額來修改此設定。

> OSP伺服器群組有特定的硬關聯性/反關聯性限制；如果資源不足、無法部署在個別節點上、或資源不足、無法共用節點、則VM將無法開機。

若要設定關聯群組、請參閱 "如何設定OpenStack執行個體的關聯性和反關聯性？"。

使用自訂安裝檔案進行**OpenShift**部署

IPI可透過本文稍早討論的互動式精靈、輕鬆部署OpenShift叢集。不過、您可能需要在叢集部署中變更某些預設值。

在這些執行個體中、您無需立即部署叢集、即可執行及執行wizard;而是建立組態檔、以便日後部署叢集。如果您需要變更任何IPI預設值、或是想要在環境中部署多個相同的叢集以供其他用途（例如多租戶）、這項功能就非常實用。如需建立OpenShift自訂安裝組態的詳細資訊、請參閱 "Red Hat OpenShift使用自訂功能在OpenStack上安裝叢集"。

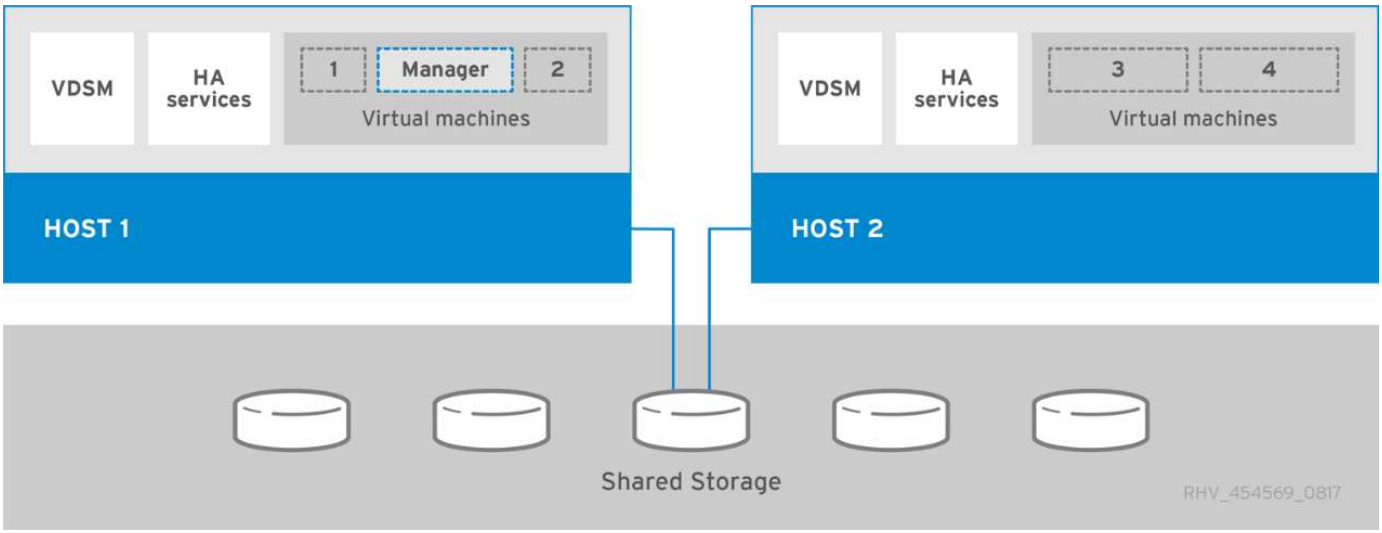
Red Hat虛擬化的OpenShift

Red Hat虛擬化（RHV）是企業虛擬資料中心平台、可在Red Hat Enterprise Linux（RHEL）上執行、並使用KVM Hypervisor。

如需更多有關RHV的資訊、請參閱 "Red Hat虛擬化網站"。

RHV提供下列功能：

- * 集中管理 VM 和主機 * RHE 管理程式在部署中以實體或虛擬機器（ VM ）的形式執行、並提供 Web 型 GUI 、以便從中央介面管理解決方案。
- * 自我代管引擎 * 為了將硬體需求降至最低、 RHE 可讓 RHE Manager （ RHV-M ）在執行來賓 VM 的同一部主機上部署為 VM 。
- * 高可用度 * 為了避免在主機故障時發生中斷、 RHE 可讓 VM 設定為高可用度。高可用度的VM是使用恢復原則在叢集層級加以控制。
- * 高擴充性 * 單一 RHE 叢集可擁有多達 200 部 Hypervisor 主機、讓 IT 能夠支援大型 VM 的需求、以裝載資源密集的企業級工作負載。
- RHE 採用從 RHE 繼承的增強安全性 * 、安全虛擬化（ sVirt ）和安全性增強 Linux （ SELinux ）技術、以提升主機和 VM 的安全性和強化。這些功能的主要優勢在於邏輯隔離VM及其相關資源。



網路設計

Red Hat OpenShift on NetApp解決方案使用兩個資料交換器、以25Gbps的速率提供主要資料連線能力。它也使用兩個額外的管理交換器、以1Gbps的速率提供連線能力、以進行儲存節點的頻內管理、以及IPMI功能的頻外管理。OCP使用RHV上的虛擬機器邏輯網路進行叢集管理。本節說明解決方案中所使用的每個虛擬網路區段的安排和用途、並概述部署解決方案的先決條件。

VLAN需求

RHV上的Red Hat OpenShift設計用於使用虛擬區域網路（VLAN）、以邏輯方式分隔不同用途的網路流量。此組態可擴充以滿足客戶需求、或進一步隔離特定的網路服務。下表列出在NetApp驗證解決方案時實作解決方案所需的VLAN。

VLAN	目的	**VLAN ID**
頻外管理網路	管理實體節點和IPMI	16
VM網路	虛擬訪客網路存取	1172
頻內管理網路	管理RHV-H節點、RHV-Manager和Ovirtmgmt網路	3343.
儲存網路	適用於iSCSI的儲存網路NetApp Element	3344
移轉網路	用於虛擬來賓移轉的網路	3345

網路基礎架構支援資源

在部署OpenShift Container Platform之前、應先準備好下列基礎架構：

- 至少有一部DNS伺服器提供完整的主機名稱解析、可從頻內管理網路和VM網路存取。
- 至少有一部NTP伺服器可從頻內管理網路和VM網路存取。
- （可選）用於帶內管理網路和VM網路的傳出網際網路連線。

正式作業部署的最佳實務做法

本節列出組織在將此解決方案部署至正式作業環境之前、應考慮的幾項最佳實務做法。

將OpenShift部署至至少三個節點的RHV叢集

本文件所述的驗證架構、提供最小的硬體部署、適用於HA作業、方法是部署兩個RHV-H Hypervisor節點、並確保具備容錯功能的組態、讓兩個主機都能管理託管引擎和已部署的VM、在兩個Hypervisor之間移轉。

由於Red Hat OpenShift一開始會部署三個主節點、因此在雙節點組態中、至少有兩個主節點會佔用同一個節點、因此如果該特定節點無法使用、可能會導致OpenShift中斷。因此、Red Hat的最佳實務做法是將至少三個RHV-H Hypervisor節點部署為解決方案的一部分、以便能平均分散OpenShift主節點、並使解決方案獲得更高程度的容錯能力。

設定虛擬機器/主機關聯性

您可以啟用VM/主機關聯性、將OpenShift主控點分散到多個Hypervisor節點。

關聯性是一種定義一組VM和/或主機規則的方法、可決定VM是在同一主機上一起執行、還是在群組中的主機上執行、或是在不同的主機上執行。它會透過建立關聯群組來套用至VM、這些群組由一組相同的參數和條件的VM和/或主機組成。根據關聯群組中的VM是在同一主機或群組中的主機上執行、還是分別在不同主機上執行、關聯群組的參數可以定義正關聯性或負關聯性。

為參數定義的條件可以是強制或軟強制。強制強制性可確保關聯群組中的虛擬機器永遠嚴格遵循正面或負面關聯性、而不受任何外部條件影響。軟強制功能可確保關聯群組中的VM設定較高的喜好設定、以便在可行的情況下遵循正面或負面關聯性。在本文所述的兩或三個Hypervisor組態中、建議使用軟性關聯性。在較大型的叢集中、硬關聯性可以正確分散OpenShift節點。

若要設定關聯群組、請參閱 "Red Hat 6.11.關聯群組文件"。

使用自訂安裝檔案進行OpenShift部署

IPI可透過本文稍早討論的互動式精靈、輕鬆部署OpenShift叢集。不過、在叢集部署過程中、可能需要變更某些預設值。

在這些執行個體中、您無需立即部署叢集、即可執行及執行精靈工作。而是會建立組態檔、以便稍後部署叢集。如果您想要變更任何IPI預設值、或是想要在環境中部署多個相同的叢集以供其他用途（例如多租戶）、這項功能就非常實用。如需建立OpenShift自訂安裝組態的詳細資訊、請參閱 "Red Hat OpenShift使用自訂功能在RHV上安裝叢集"。

VMware vSphere上的OpenShift

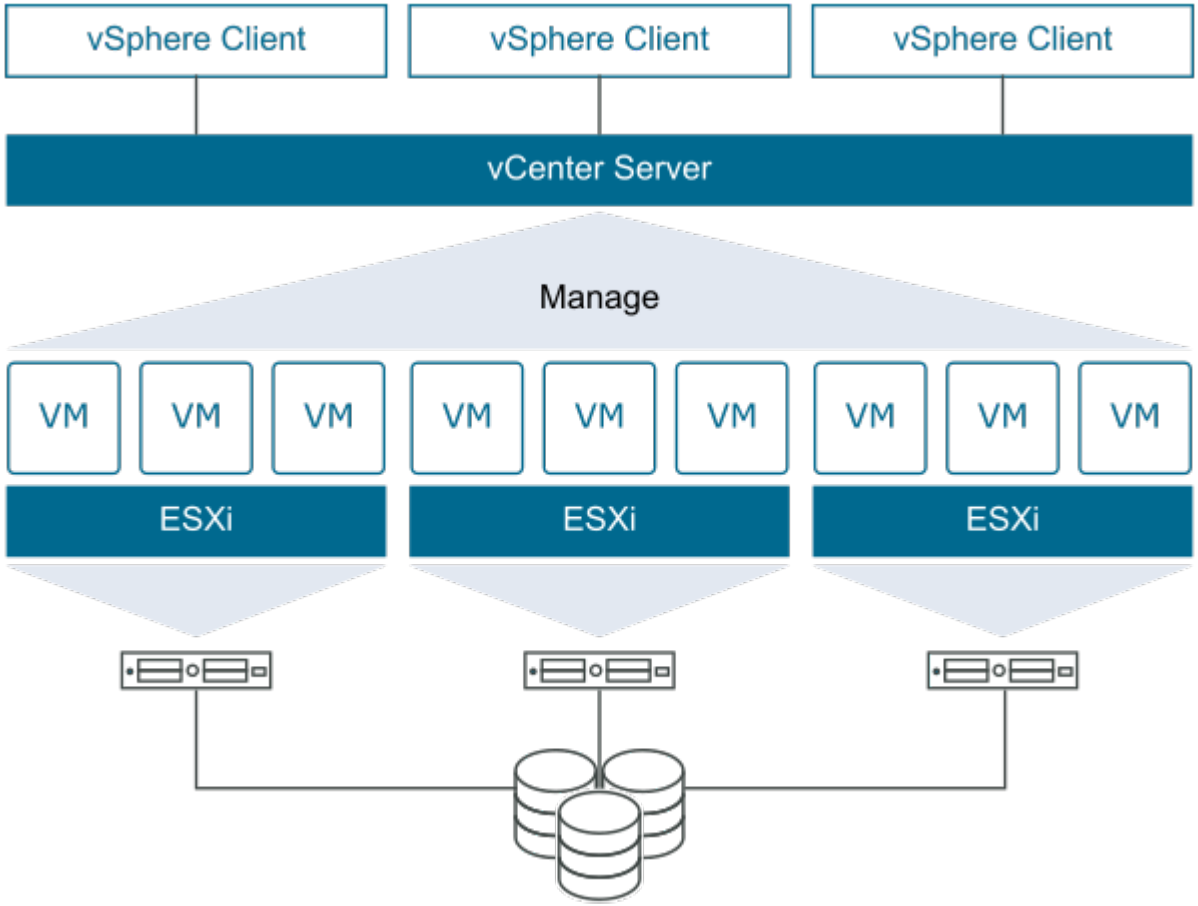
VMware vSphere是一套虛擬化平台、可集中管理ESXi Hypervisor上執行的大量虛擬化伺服器和網路。

如需VMware vSphere的詳細資訊、請參閱 "VMware vSphere網站"。

VMware vSphere提供下列功能：

- * VMware vCenter Server* VMware vCenter Server 可從單一主控台統一管理所有主機和虛擬機器、並彙總叢集、主機和虛擬機器的效能監控。
- * VMware vSphere VMotion* VMware vCenter 可讓您在要求時、以不中斷營運的方式、在叢集中的節點之間熱移轉 VM 。
- * vSphere High Availability * 為了避免在主機故障時造成中斷、 VMware vSphere 可讓主機進行叢集化並設定為高可用度。由於主機故障而中斷的VM會在叢集中的其他主機上、於近期重新開機、以還原服務。

- * 分散式資源排程器（DRS） * VMware vSphere 叢集可設定為負載平衡其所裝載虛擬機器的資源需求。具有資源爭用的VM可熱移轉至叢集中的其他節點、以確保有足夠的可用資源。



網路設計

Red Hat OpenShift on NetApp解決方案使用兩個資料交換器、以25Gbps的速率提供主要資料連線能力。此外、它還使用兩個額外的管理交換器、以1Gbps的連線能力提供儲存節點的頻內管理、以及IPMI功能的頻外管理。OCP使用VMware vSphere上的VM邏輯網路進行叢集管理。本節說明解決方案中使用的每個虛擬網路區段的安排和用途、並概述部署解決方案的先決條件。

VLAN需求

VMware vSphere上的Red Hat OpenShift設計用於使用虛擬區域網路（VLAN）、以邏輯方式分隔不同用途的網路流量。此組態可擴充以滿足客戶需求、或進一步隔離特定的網路服務。下表列出在NetApp驗證解決方案時實作解決方案所需的VLAN。

VLAN	目的	VLAN ID
頻外管理網路	管理實體節點和IPMI	16
VM網路	虛擬訪客網路存取	181
儲存網路	適用於不中斷NFS的儲存網路ONTAP	184..
儲存網路	適用於iSCSI的儲存網路ONTAP	185.
頻內管理網路	ESXi節點、vCenter Server ONTAP Select的管理功能	3480

VLAN	目的	VLAN ID
儲存網路	適用於iSCSI的儲存網路NetApp Element	3481
移轉網路	用於虛擬來賓移轉的網路	3482.34

網路基礎架構支援資源

在部署OpenShift Container Platform之前、應先準備好下列基礎架構：

- 至少有一部DNS伺服器提供完整的主機名稱解析、可從頻內管理網路和VM網路存取。
- 至少有一部NTP伺服器可從頻內管理網路和VM網路存取。
- （可選）用於帶內管理網路和VM網路的傳出網際網路連線。

正式作業部署的最佳實務做法

本節列出組織在將此解決方案部署至正式作業環境之前、應考慮的幾項最佳實務做法。

將**OpenShift**部署至至少三個節點的**ESXi**叢集

本文件所述的驗證架構、提供最小的硬體部署、可部署兩個ESXi Hypervisor節點、並透過啟用VMware vSphere HA和VMware VMotion來確保組態容錯。此組態可讓已部署的VM在兩個Hypervisor之間移轉、並在一部主機無法使用時重新開機。

由於Red Hat OpenShift一開始會部署三個主節點、因此在某些情況下、雙節點組態中至少有兩個主節點可以佔用同一個節點、因此如果該特定節點無法使用、可能會導致OpenShift中斷運作。因此、Red Hat最佳實務做法是至少部署三個ESXi Hypervisor節點、以便能平均分散OpenShift主節點、進而提供更高程度的容錯能力。

設定虛擬機器和主機關聯性

啟用VM和主機關聯性、可確保OpenShift主機能夠在多個Hypervisor節點之間散佈。

關聯性或反關聯性是一種定義一組VM和/或主機規則的方法、用以判斷VM是在同一主機或群組中的主機上一起執行、還是在不同的主機上執行。它會透過建立關聯群組來套用至VM、這些群組由一組相同的參數和條件的VM和/或主機組成。根據關聯群組中的VM是在同一主機或群組中的主機上執行、還是分別在不同主機上執行、關聯群組的參數可以定義正關聯性或負關聯性。

若要設定關聯群組、請參閱 "vSphere 6.7說明文件：使用DRS關聯性規則"。

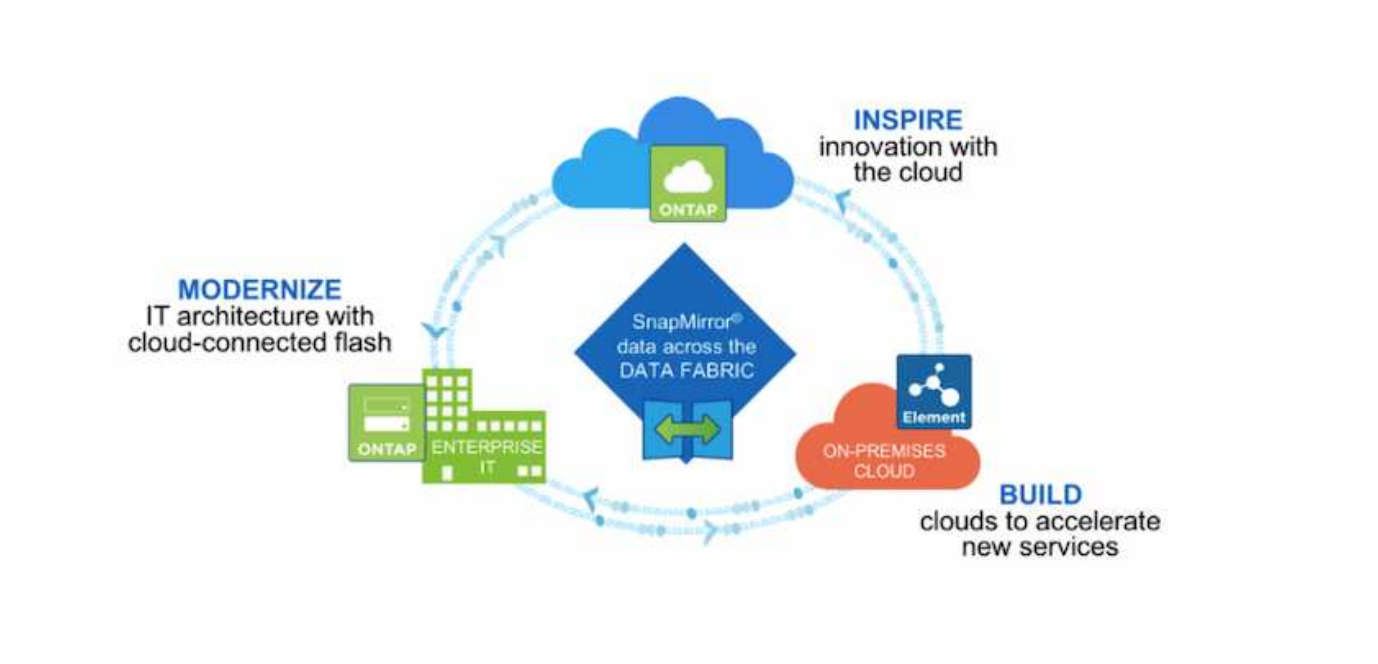
使用自訂安裝檔案進行**OpenShift**部署

IPI可透過本文稍早討論的互動式精靈、輕鬆部署OpenShift叢集。不過、您可能需要在叢集部署中變更某些預設值。

在這些執行個體中、您無需立即部署叢集、即可執行及執行精靈、但精靈會建立組態檔、以便日後部署叢集。如果您需要變更任何IPI預設值、或是想要在環境中部署多個相同的叢集以供其他用途（例如多租戶）、這項功能就非常實用。如需建立OpenShift自訂安裝組態的詳細資訊、請參閱 "Red Hat OpenShift使用自訂功能在vSphere上安裝叢集"。

NetApp儲存設備總覽

NetApp擁有數個符合Astra Trident Storage Orchestrator資格的儲存平台、可為部署在Red Hat OpenShift上的應用程式配置儲存設備。



- 支援以檔案為基礎（NFS）和區塊為基礎（iSCSI）的使用案例、可同時執行NetApp的支援功能和功能。AFF FAS ONTAP
- 在雲端和虛擬空間中、使用者可分別獲得相同的效益。Cloud Volumes ONTAP ONTAP Select
- NetApp Cloud Volumes Service 的功能（AWS/GCP）和Azure NetApp Files 功能豐富的功能、可在雲端提供檔案型儲存設備。
- 在可高度擴充的環境中、支援區塊型（iSCSI）使用案例。NetApp Element

> NetApp產品組合中的每個儲存系統都能簡化內部部署站台與雲端之間的資料管理與移動、確保您的資料是應用程式所在。

以下頁面提供更多有關在Red Hat OpenShift with NetApp解決方案中驗證的NetApp儲存系統的資訊：

- "NetApp ONTAP"
- "NetApp Element"

NetApp ONTAP

NetApp ONTAP 功能強大的儲存軟體工具、具備直覺式GUI、REST API與自動化整合、AI資訊預測分析與修正行動、不中斷營運的硬體升級、以及跨儲存設備匯入等功能。

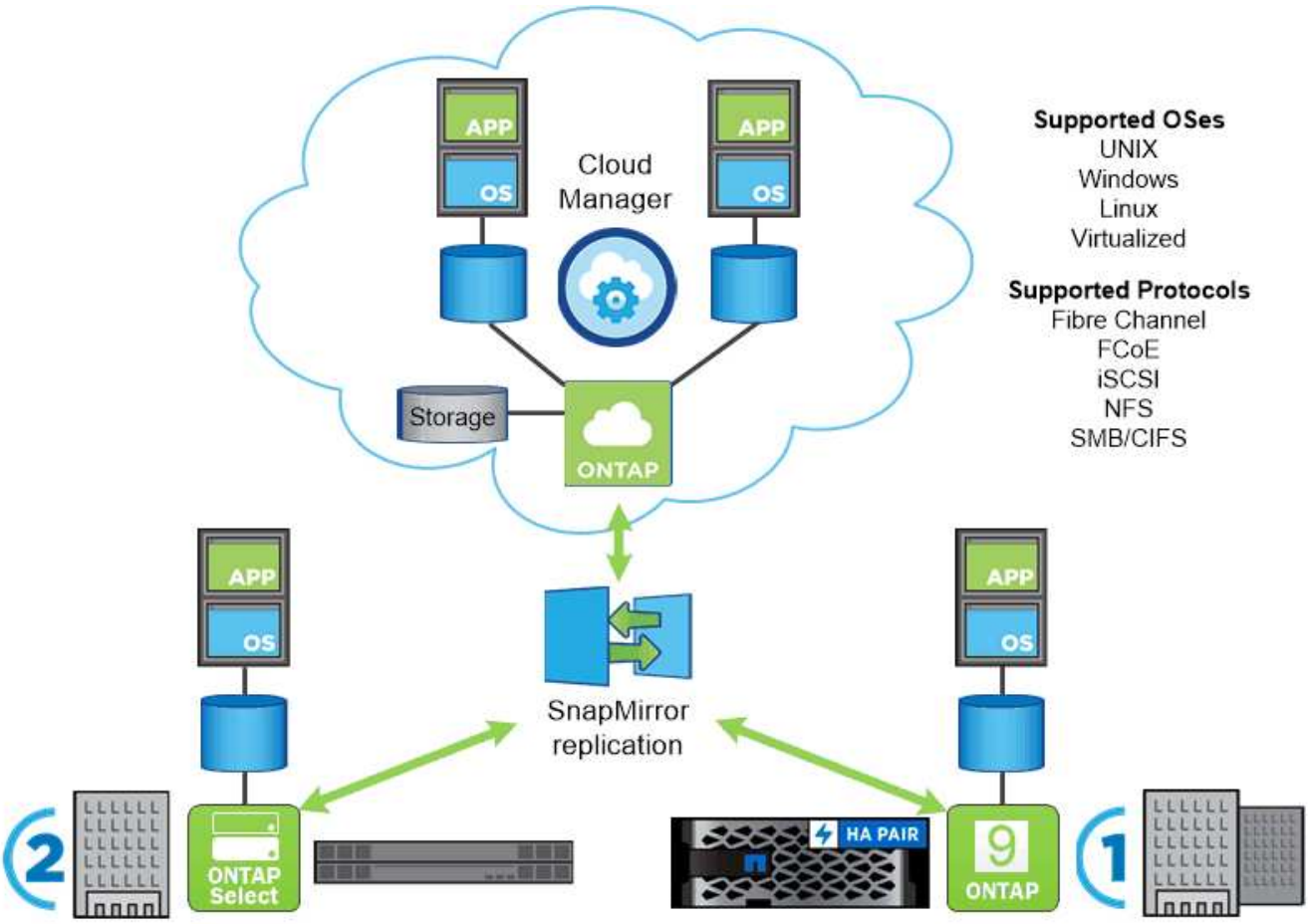
如需更多關於NetApp ONTAP NetApp NetApp資訊儲存系統的資訊、請造訪 "NetApp ONTAP 產品網站"。

支援下列功能：ONTAP

- 統一化儲存系統、可同時存取及管理NFS、CIFS、iSCSI、FC、FCoE、 和FC-NVMe傳輸協定。
- 不同的部署模式包括在All Flash、混合式和All HDD硬體組態上的內部部署、ONTAP Select 在支援的Hypervisor上的VM型儲存平台（例如：用作支援的Hypervisor）、以及在雲端上用Cloud Volumes ONTAP 作支援的
- 支援自動資料分層、即時資料壓縮、重複資料刪除及壓縮、可提升ONTAP 資料在支援功能完善的系統上的儲存效率。
- 工作負載型QoS控制儲存設備。
- 與公有雲無縫整合、以利資料分層和保護。此外、支援強大的資料保護功能、可在任何環境中脫穎而出：ONTAP
  - * NetApp Snapshot複本。*快速的時間點資料備份、使用最少的磁碟空間、不需額外的效能負荷。
  - * NetApp SnapMirror.*將資料的Snapshot複本從一個儲存系統鏡射到另一個儲存系統。支援將資料鏡射到其他實體平台、以及雲端原生服務。ONTAP
  - * NetApp SnapLock 功能*可將不可重複寫入的資料寫入無法在指定期間覆寫或清除的特殊磁碟區、以有效管理不可重複寫入的資料。
  - * NetApp SnapVault 功能*可將多個儲存系統的資料備份至中央Snapshot複本、作為所有指定系統的備份。
  - * NetApp SyncMirror Real-.*可將資料即時、RAID層級的鏡射、鏡射到實體連接至相同控制器的兩個不同磁碟叢。
  - * NetApp SnapRestore 功能*可根據需求、從Snapshot複本快速還原備份資料。
  - * NetApp FlexClone。*可根據Snapshot複本、即時提供NetApp磁碟區完整讀取且可寫入的複本。

如需ONTAP 更多關於效能的資訊、請參閱 "供應說明文件中心 ONTAP"。

NetApp ONTAP 產品可在內部部署、虛擬化或雲端上使用。



NetApp平台

NetApp AFF/FAS

NetApp提供強大的All Flash AFF （VMware）與橫向擴充混合式FAS （VMware）儲存平台、專為低延遲效能、整合式資料保護及多重傳輸協定支援而量身打造。

這兩種系統均採用ONTAP NetApp的NetApp支援資料管理軟體、這是業界最先進的資料管理軟體、可提供高可用度、雲端整合、簡化的儲存管理、為您的資料架構提供企業級的速度、效率和安全性。

如需NetApp AFF/FAS平台的詳細資訊、請按一下 "請按這裡"。

ONTAP Select

透過軟體定義部署的NetApp解決方案、可將其部署到您環境中的Hypervisor上。ONTAP Select ONTAP可安裝在VMware vSphere或KVM上、並提供硬體ONTAP 型的完整功能與體驗。

如需ONTAP Select 更多有關資訊、請按一下 "請按這裡"。

Cloud Volumes ONTAP

NetApp Cloud Volumes ONTAP 功能是雲端部署版本的NetApp ONTAP 功能、可部署在許多公有雲上、包括Amazon AWS、Microsoft Azure和Google Cloud。

如需Cloud Volumes ONTAP 更多有關資訊、請按一下 "請按這裡"。

包含NetApp的Red Hat OpenShift NetApp Element

藉助於支援環境的容量和處理量、每個儲存節點都能提供模組化且可擴充的效能。NetApp Element在單一叢集中、可將各個節點從4個擴充至100個、並提供許多進階的儲存管理功能。NetApp Element

如需NetApp Element 更多有關資料、請造訪 "NetApp SolidFire 產品網站"。

iSCSI登入重新導向與自我修復功能

利用iSCSI儲存傳輸協定（iSCSI儲存傳輸協定）、將SCSI命令封裝到傳統TCP/IP網路上的標準方法。NetApp Element當SCSI標準改變或乙太網路效能改善時、iSCSI儲存傳輸協定將不需進行任何變更。

雖然所有儲存節點都有管理IP和儲存IP、NetApp Element 但該軟件卻會針對叢集中的所有儲存流量、通告單一儲存虛擬IP位址（SVIP位址）。在iSCSI登入程序中、儲存設備會回應目標磁碟區已移至不同的位址、因此無法繼續協商程序。然後、主機會在不需要重新設定主機端的程序中、將登入要求重新發往新位址。此程序稱為iSCSI登入重新導向。

iSCSI登入重新導向是NetApp Element 支援不支援功能的軟體叢集的重要一環。收到主機登入要求時、節點會根據IOPS和磁碟區的容量需求、決定叢集的哪個成員應處理流量。如果NetApp Element 單一節點處理的磁碟區流量太多、或新增了節點、則會將磁碟區分散到整個過程中的整個軟件叢集、然後重新分配。在陣列中配置多個指定磁碟區複本。

如此一來、如果節點故障之後又發生磁碟區重新分佈、則登出後登入並重新導向至新位置後、主機連線能力將不會受到影響。利用iSCSI登入重新導向功能、NetApp Element 一套自我修復的橫向擴充架構、能夠不中斷升級與作業。

軟件叢集**QoS NetApp Element**

利用支援QoS的整套軟體叢集、可以根據每個Volume動態設定QoS。NetApp Element您可以使用每個Volume QoS設定、根據您定義的SLA來控制儲存效能。下列三個可設定的參數可定義QoS：

- 最低**IOPS**。 NetApp Element 此為整個磁碟區提供的持續IOPS下限。為磁碟區設定的最低IOPS是保證磁碟區效能的等級。每個磁碟區的效能不會低於此層級。
- 最大**IOPS**。 NetApp Element 此為指支援特定磁碟區的穩定IOPS上限。
- *爆發IOPS。*在短時間爆發案例中允許的IOPS上限。「連拍持續時間」設定可設定、預設值為1分鐘。如果某個Volume的執行量低於最高IOPS層級、則會累積大量資源。當效能等級變得非常高且受到推升時、磁碟區上的IOPS可在超過最大IOPS的情況下進行短暫的突發。

多租戶

安全的多租戶共享功能包括：

- *安全驗證。*挑戰握手驗證傳輸協定（CHAP）用於安全的Volume存取。輕量型目錄存取傳輸協定（LDAP）用於安全存取叢集、以進行管理和報告。
- * Volume存取群組（VAG）。*（選用）VAG可取代驗證、將任意數量的iSCSI啟動器特定iSCSI合格名稱（IQN）對應至一或多個磁碟區。若要存取VAG中的磁碟區、啟動器的IQN必須位於磁碟區群組的允許IQN清單中。
- 租戶虛擬**LAN**（**VLAN**）。*在網路層級、NetApp Element iSCSI啟動器和支援此功能的軟體叢集之間的端點對端點網路安全性、可透過使用VLAN來實現。針對任何為了隔離工作負載或租戶而建立的VLAN、NetApp Element 則由NetApp軟體建立獨立的iSCSI目標SVIP位址、只能透過特定的VLAN存取。
- 支援**VRF**的**VLANs**。 NetApp Element 為了進一步支援資料中心的安全性與擴充性、利用此軟體、您可以啟用任何租戶VLAN來執行類似VRF的功能。這項功能新增了這兩項主要功能：
  - * L3路由傳送至租戶SVIP位址。*此功能可讓您將iSCSI啟動器置於獨立的網路或VLAN上、而不受NetApp Element 支援於該軟體叢集的網路或VLAN上。
  - *重疊或重複的IP子網路。*此功能可讓您將範本新增至租戶環境、讓每個租戶VLAN都能從相同的IP子網路指派IP位址。這項功能可在需要擴充及保留IPspace的服務供應商環境中使用。

企業儲存效率

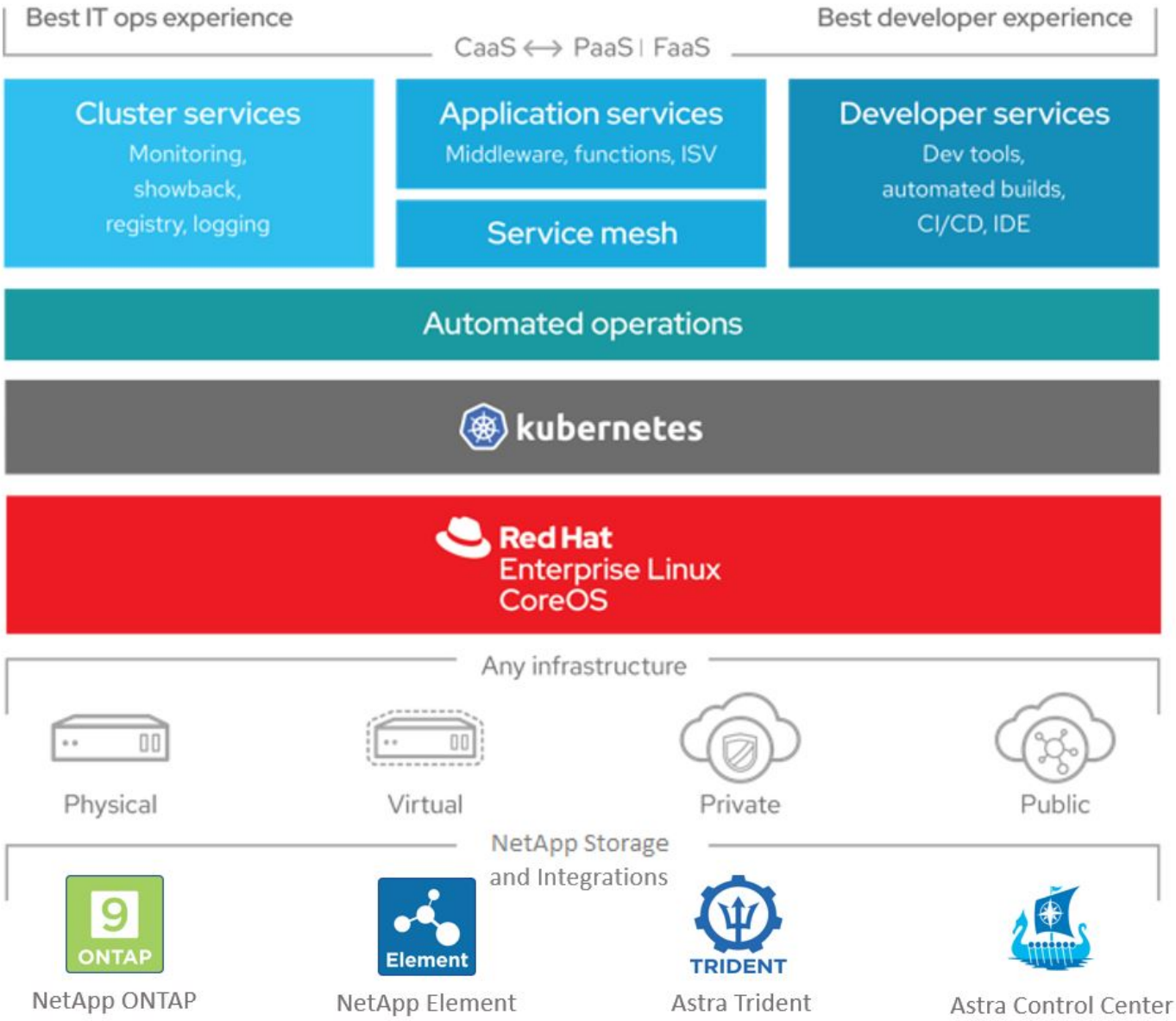
這個功能可提升整體儲存效率與效能。NetApp Element下列功能會即時執行、永遠開啟、而且使用者不需要手動設定：

- *重複資料刪除。*系統僅儲存獨特的4K區塊。任何重複的4K區塊都會自動與已儲存的資料版本相關聯。資料位於區塊磁碟機上、並使用NetApp Element 功能完善的Helix資料保護功能進行鏡射。此系統可大幅減少系統內的容量使用量和寫入作業。
- *壓縮。*壓縮是在資料寫入NVRAM之前內嵌執行的。資料會壓縮、儲存在4K區塊中、並在系統中保持壓縮狀態。此壓縮可大幅減少整個叢集的容量使用量、寫入作業和頻寬使用量。
- *精簡配置。*此功能可在您需要時提供適當數量的儲存設備、免除過度配置磁碟區或未充分利用磁碟區所造成的容量消耗。
- * Helix。*個別磁碟區的中繼資料儲存在中繼資料磁碟機上、並複寫到次要中繼資料磁碟機以供備援。

> 元件是專為自動化而設計。所有的儲存功能都可透過API取得。這些API是UI用來控制系統的唯一方法。

NetApp儲存整合概述

NetApp提供多種產品來協助您協調及管理以容器為基礎的環境中的持續資料、例如Red Hat OpenShift。



NetApp Astra Control提供豐富的儲存設備與應用程式感知資料管理服務、適用於狀態明確的Kubernetes工作負載、採用NetApp資料保護技術。Astra Control Service可支援雲端原生Kubernetes部署中的狀態工作負載。Astra Control Center可支援內部部署中的狀態工作負載、例如Red Hat OpenShift。如需詳細資訊、請參閱NetApp Astra Control網站 "請按這裡"。

NetApp Astra Trident是開放原始碼且完全支援的儲存協調工具、適用於容器和Kubernetes配送、包括Red Hat OpenShift。如需詳細資訊、請造訪Astra Trident網站 "請按這裡"。

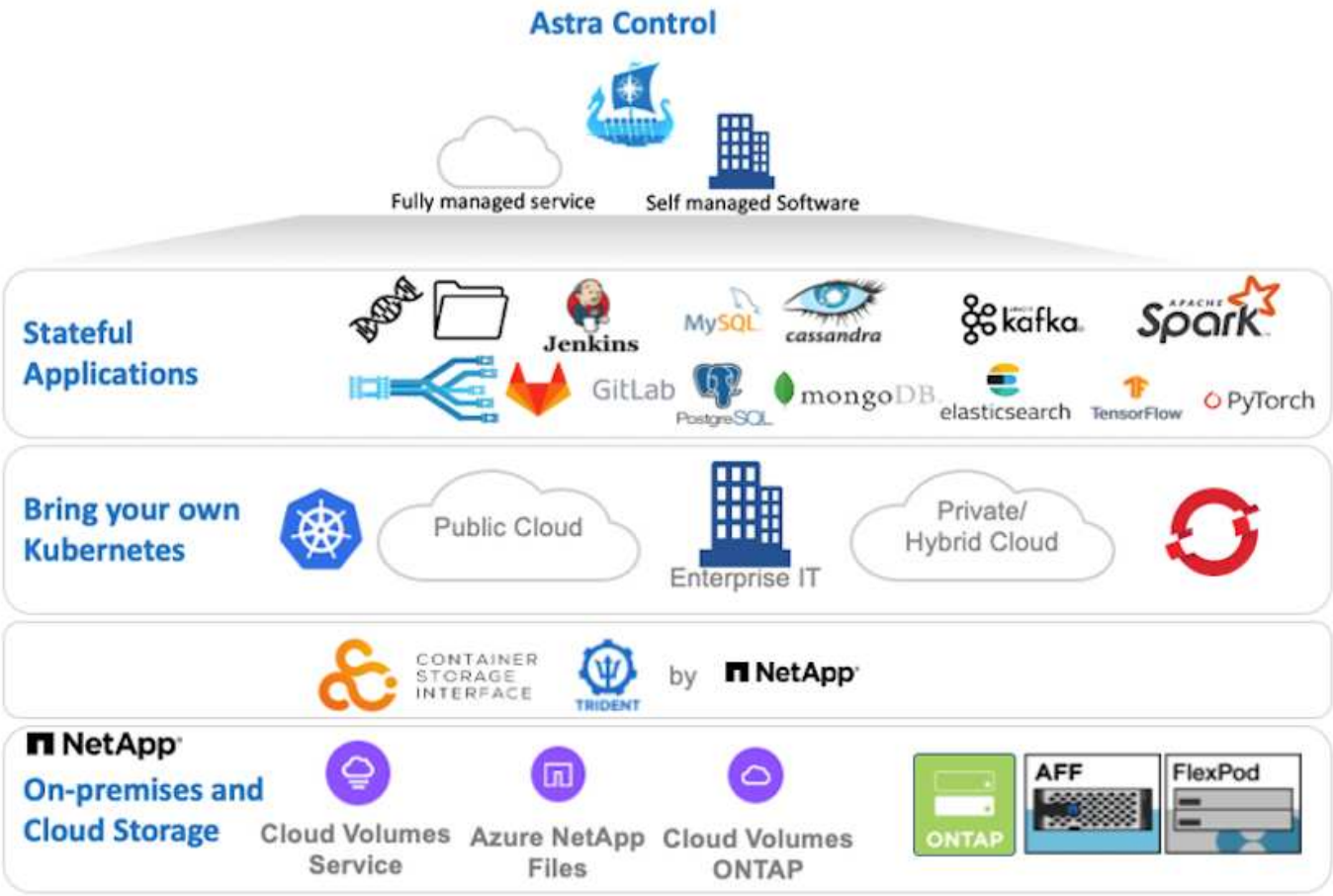
以下頁面提供更多有關NetApp產品的資訊、這些產品已通過Red Hat OpenShift with NetApp解決方案的應用程式與持續儲存管理驗證：

- "NetApp Astra控制中心"
- "NetApp Astra Trident"

NetApp Astra Control Center總覽

NetApp Astra Control Center提供豐富的儲存設備與應用程式感知資料管理服務、適用於部署在內部部署環境中

且採用NetApp資料保護技術的狀態式Kubernetes工作負載。



NetApp Astra Control Center可安裝在Red Hat OpenShift叢集上、該叢集已部署Astra Trident儲存Orchestrator、並已設定儲存類別和儲存後端、以供NetApp ONTAP orchstorage系統使用。

如需安裝及組態Astra Trident以支援Astra Control Center、請參閱 "本文檔"。

在雲端連線的環境中、Astra Control Center會使用Cloud Insights 效益技術來提供進階監控和遙測功能。如果沒有支援功能、則可透過開放式指標端點、將有限的監控和遙測（7天的數據價值）匯出至Kubernetes原生監控工具（Prometheus和Grafana）Cloud Insights 。

Astra Control Center已完全整合至NetApp AutoSupport 的整套功能和Active IQ 功能、可為使用者提供支援、協助疑難排解、以及顯示使用統計資料。

除了Astra Control Center的付費版本、我們也提供90天的評估授權。評估版本可透過電子郵件和社群（Slack通路）獲得支援。客戶可以存取這些和其他知識庫文章、以及產品內建支援儀表板所提供的文件。

若要開始使用NetApp Astra Control Center、請造訪 "Astra網站"。

Astra Control Center安裝先決條件

1. 一個或多個Red Hat OpenShift叢集。目前支援版本4.6 EUS和4.7。
2. 每個Red Hat OpenShift叢集上都必須已安裝並設定Astra Trident。
3. 一或多ONTAP 個執行ONTAP 不穩定9.5或更新版本的NetApp不穩定儲存系統。

> 最佳實務做法是在站台上安裝每個OpenShift、以擁有專屬的SVM來進行持續儲存。多站台部署需要額外的儲存系統。

4. 每個OpenShift叢集上都必須設定Trident儲存後端、並以SVM作為後盾ONTAP 、以供支援整個叢集。
5. 在每個OpenShift叢集上設定的預設StorageClass、以Astra Trident做為儲存資源配置程式。
6. 必須在每個OpenShift叢集上安裝及設定負載平衡器、才能平衡負載並公開OpenShift Services。

> 請參閱連結 "請按這裡" 以取得已針對此目的驗證的負載平衡器相關資訊。

7. 必須設定私有映像登錄來裝載NetApp Astra Control Center映像。

> 請參閱連結 "請按這裡" 若要安裝及設定OpenShift Private登錄以供此用途。

8. 您必須擁有Red Hat OpenShift叢集的叢集管理存取權。
9. 您必須擁有NetApp ONTAP 等群集的管理員存取權。
10. 安裝了Docker或podman、tridentctl、及occ或kubecl工具的管理工作站、並新增至$path。

> Docker安裝必須具有大於20.10的Docker版本、而Podman安裝的podman版本必須大於3.0。

安裝**Astra Control Center**

使用作業系統集線器

1. 登入NetApp支援網站、下載最新版本的NetApp Astra Control Center。若要這麼做、必須附上NetApp帳戶的授權。下載後、將其傳輸至管理工作站。

   若要開始使用Astra Control試用授權、請造訪 "Astra註冊網站"。

2. 解壓縮tar ball並將工作目錄變更為所產生的資料夾。

```
[netapp-user@rhel7 ~]$ tar -vxzf astra-control-center-
21.12.60.tar.gz
[netapp-user@rhel7 ~]$ cd astra-control-center-21.12.60
```

3. 開始安裝之前、請先將Astra Control Center映像推送到映像登錄。您可以選擇使用Docker或Podman來執行這項作業、本步驟提供這兩者的說明。

Podman

a. 將登錄FQDN與組織/命名空間/專案名稱匯出為環境變數「正式作業」。

```
[netapp-user@rhel7 ~]$ export REGISTRY=astra-
registry.apps.ocp-vmw.cie.netapp.com/netapp-astra
```

b. 登入登錄。

```
[netapp-user@rhel7 ~]$ podman login -u ocp-user -p password
--tls-verify=false astra-registry.apps.ocp-vmw.cie.netapp.com
```

> 如果您使用「kubeadmin」使用者登入私有登錄、請使用權杖、而非密碼「podman登入-u ocp-user-p權杖—tlS-VERIF=假astra-registry.apps.ocp-vmw.cie.netapp.com`」。

> 或者、您也可以建立服務帳戶、指派登錄編輯器和/或登錄檢視器角色（取決於您是否需要推入/拉取存取）、然後使用服務帳戶的權杖登入登錄。

c. 建立Shell指令碼檔案、然後將下列內容貼入其中。

```
[netapp-user@rhel7 ~]$ vi push-images-to-registry.sh

for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded
image trimming the 'Loaded images: '
  astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image(s): //')
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  podman tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  podman push ${REGISTRY}/${astraImage}
done
```

> 如果您的登錄使用不受信任的憑證、請編輯Shell指令碼、並針對podman推送命令「podman push $註 冊表/$（ECAECA$astraImage | sed 's/[\/]\+//'）-TLS -VERIFE=假」使用「-TLS-VERIFY」。

d. 將檔案設定為可執行檔。

```
[netapp-user@rhel7 ~]$ chmod +x push-images-to-registry.sh
```

e. 執行Shell指令碼。

```
[netapp-user@rhel7 ~]$ ./push-images-to-registry.sh
```

Docker

a. 將登錄FQDN與組織/命名空間/專案名稱匯出為環境變數「正式作業」。

```
[netapp-user@rhel7 ~]$ export REGISTRY=astra-
registry.apps.ocp-vmw.cie.netapp.com/netapp-astra
```

b. 登入登錄。

```
[netapp-user@rhel7 ~]$ docker login -u ocp-user -p password
astra-registry.apps.ocp-vmw.cie.netapp.com
```

> 如果您使用「kubeadmin」使用者登入私有登錄、請使用權杖而非密碼-「docker login-u ocp-user-p權杖astra-registry.apps.ocp-vmw.cie.netapp.com`」。

> 或者、您也可以建立服務帳戶、指派登錄編輯器和/或登錄檢視器角色（取決於您是否需要推入/拉取存取）、然後使用服務帳戶的權杖登入登錄。

c. 建立Shell指令碼檔案、然後將下列內容貼入其中。

```
[netapp-user@rhel7 ~]$ vi push-images-to-registry.sh

for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded
image trimming the 'Loaded images: '
  astraImage=$(docker load --input ${astraImageFile} | sed
's/Loaded image: //')
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  docker tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  docker push ${REGISTRY}/${astraImage}
done
```

d. 將檔案設定為可執行檔。

```
[netapp-user@rhel7 ~]$ chmod +x push-images-to-registry.sh
```

e. 執行Shell指令碼。

```
[netapp-user@rhel7 ~]$ ./push-images-to-registry.sh
```

4. 使用非公開信任的私有映像登錄時、請將映像登錄TLS憑證上傳至OpenShift節點。若要這麼做、請使用TLS憑證在openshift-config命名空間中建立組態對應、並將其修補至叢集映像組態、使憑證成為信任的憑證。

```
[netapp-user@rhel7 ~]$ oc create configmap default-ingress-ca -n
openshift-config --from-file=astra-registry.apps.ocp
-vmw.cie.netapp.com=tls.crt

[netapp-user@rhel7 ~]$ oc patch image.config.openshift.io/cluster
--patch '{"spec":{"additionalTrustedCA":{"name":"default-ingress-
ca"}}}' --type=merge
```

> 如果您使用OpenShift內部登錄搭配來自入口操作員的預設TLS憑證搭配路由、您仍需依照前一個步驟將憑證修補成路由主機名稱。若要從入口操作員擷取憑證、您可以使用命令「occExtract secret /路由器-ca --keys=ls.crt -n openshift-inemit-opoperers」。

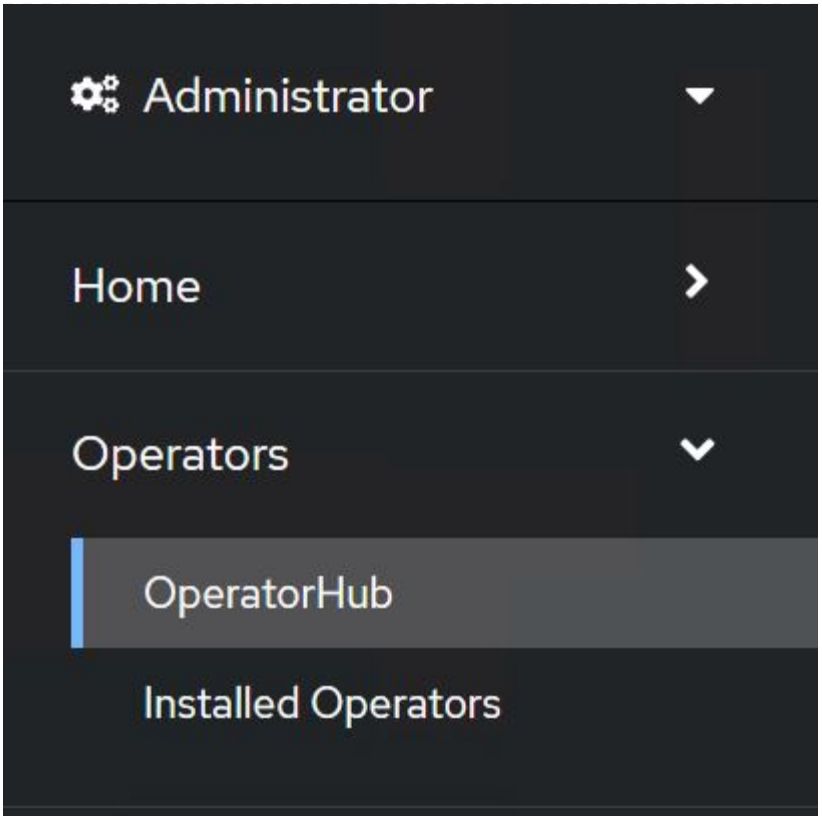
5. 為Astra Control Center建立命名空間「NetApp-acc operator」。

```
[netapp-user@rhel7 ~]$ oc create ns netapp-acc-operator

namespace/netapp-acc-operator created
```

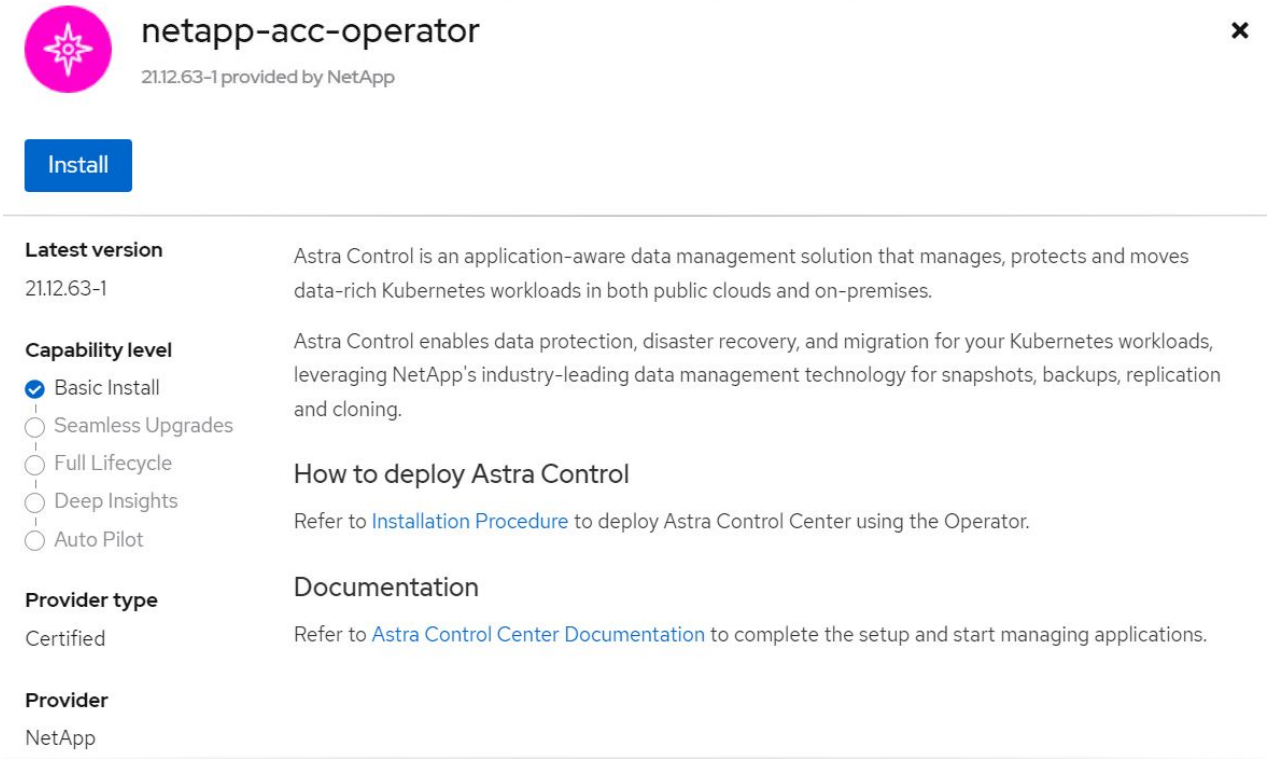
6. 使用認證資料建立秘密、以登入「NetApp-acc operator」命名空間中的映像登錄。

```
[netapp-user@rhel7 ~]$ oc create secret docker-registry astra-
registry-cred --docker-server=astra-registry.apps.ocp
-vmw.cie.netapp.com --docker-username=ocp-user --docker
-password=password -n netapp-acc-operator

secret/astra-registry-cred created
```

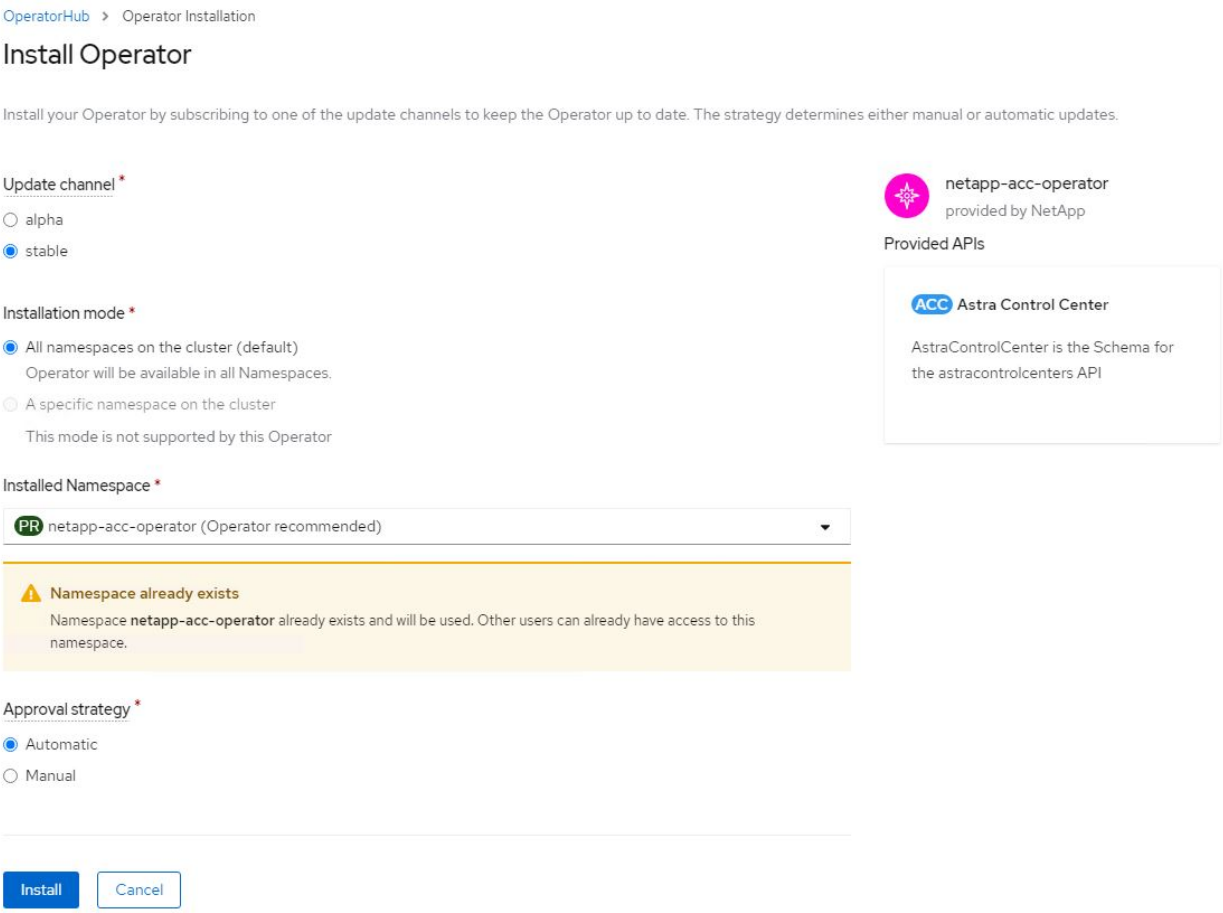
7. 使用叢集管理存取權登入Red Hat OpenShift GUI主控台。
8. 從Perspective（透視）下拉列表中選擇Administrator（管理員
9. 瀏覽至「運算子」>「運算子中樞」、然後搜尋Astra。



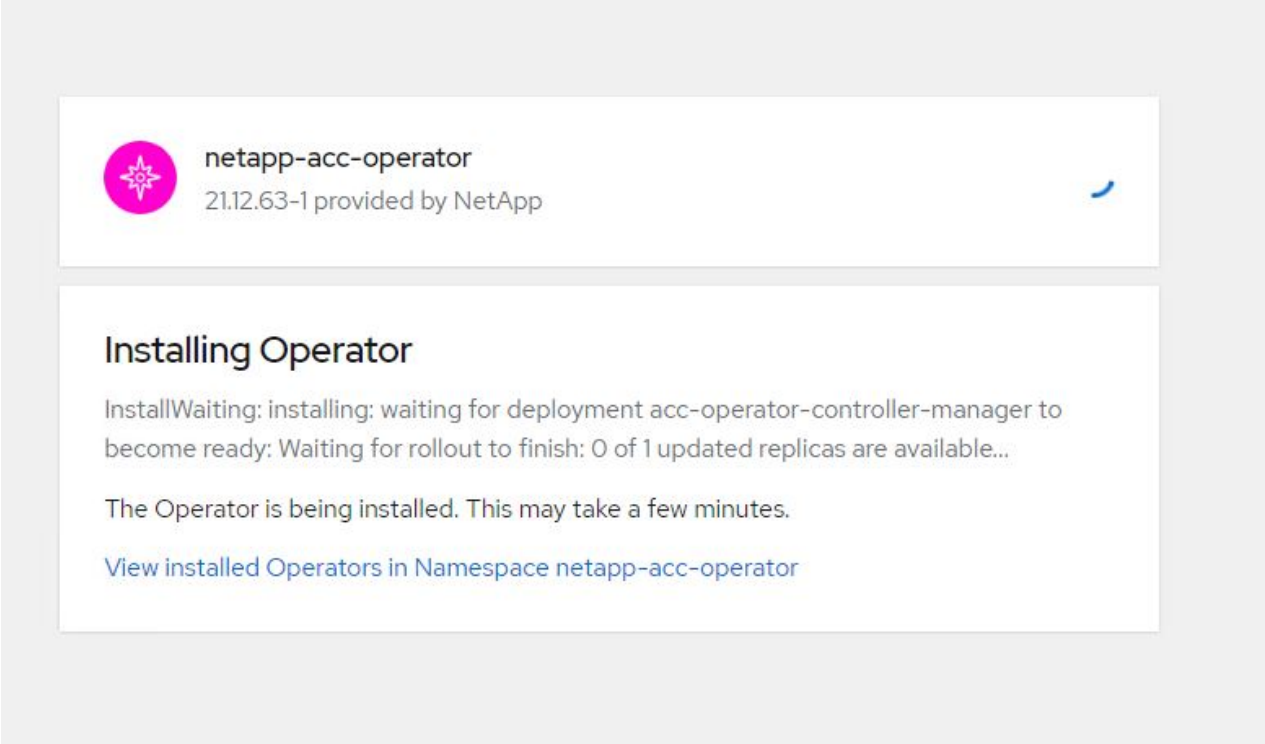
10. 選取「NetApp-acc operator」方塊、然後按一下「Install（安裝）」。



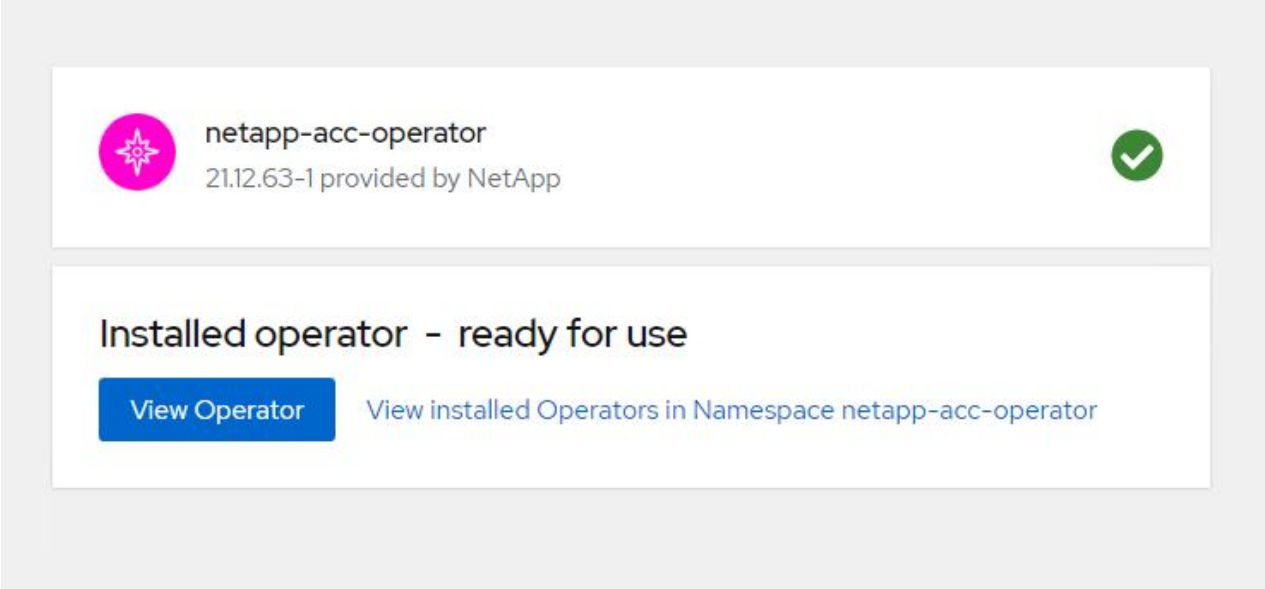
11. 在Install Operator（安裝操作員）畫面上、接受所有預設參數、然後按一下「Install（安裝）」。



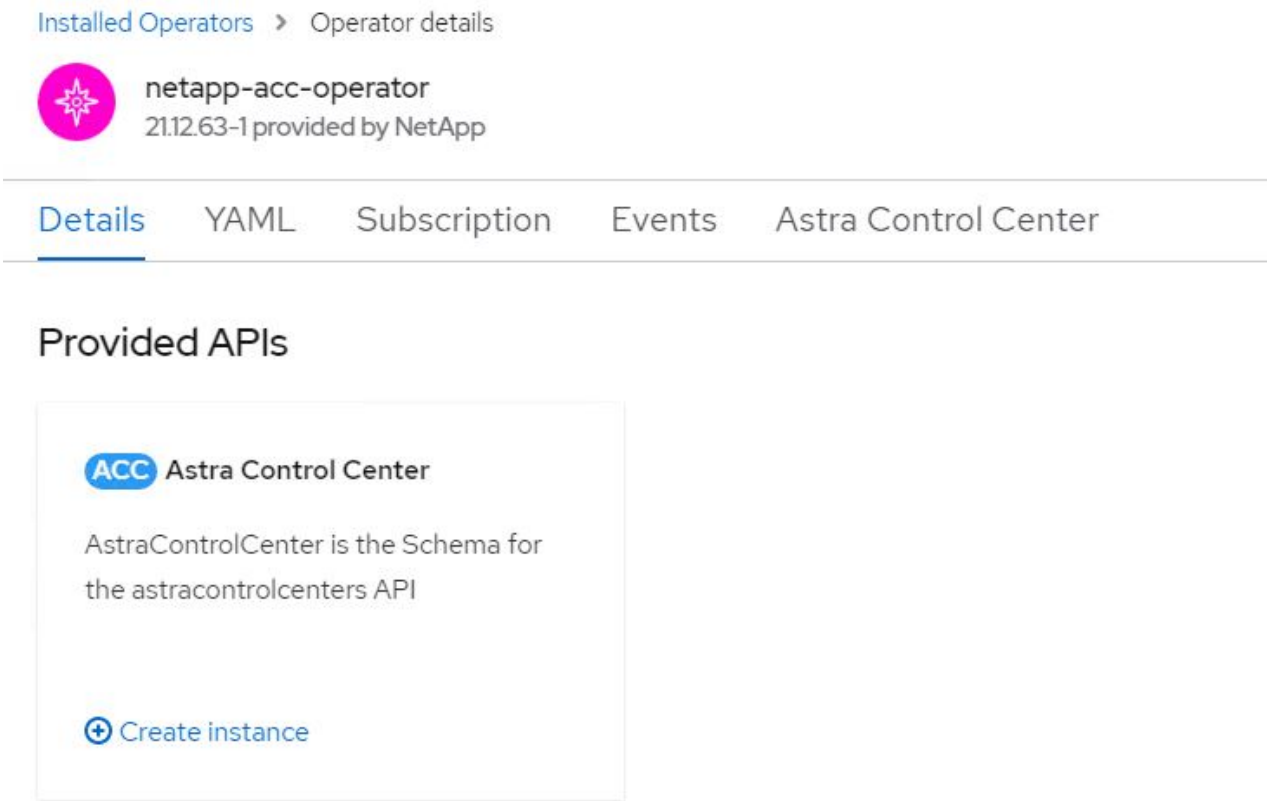
12. 等待操作員安裝完成。



13. 一旦操作員安裝成功、請瀏覽至「View operator」（檢視操作員）。



14. 然後按一下操作者中Astra Control Center的「Create Instance」（建立執行個體）。



15. 填寫「Create適用的」表單欄位、然後按一下「Create」（建立）。
    a. （可選）編輯Astra Control Center執行個體名稱。
    b. （可選）啟用或停用自動支援。建議保留「自動支援」功能。
    c. 輸入Astra Control Center的FQDN。
    d. 輸入Astra Control Center版本；預設會顯示最新版本。

e. 輸入Astra Control Center的帳戶名稱和管理員詳細資料、例如名字、姓氏和電子郵件地址。

f. 輸入Volume回收原則、預設為保留。

g. 在「Image登錄」中、輸入登錄的FQDN以及將映像推送到登錄時所提供的組織名稱（在此範例中為「astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra`」）。

h. 如果您使用需要驗證的登錄、請在「映像登錄」區段中輸入機密名稱。

i. 設定Astra Control Center資源限制的擴充選項。

j. 如果您要將PVCS放置在非預設儲存類別上、請輸入儲存類別名稱。

k. 定義客戶需求日處理偏好設定。

Project: netapp-acc-operator

Name *

astra

Labels

app=frontend

Account Name *

HCG Solutions Engineering

Astra Control Center account name

Astra Address *

astra-control-center.cie.netapp.com

AstraAddress defines how Astra will be found in the data center. This IP address and/or DNS A record must be created prior to provisioning Astra Control Center. Example - "astra.example.com" The A record and its IP address must be allocated prior to provisioning Astra Control Center

Astra Version *

21.12.60

Version of AstraControlCenter to deploy. You are provided a Helm repository with a corresponding version. Example - 1.5.2, 1.4.2-patch

Email *

solutions_tme@netapp.com

EmailAddress will be notified by Astra as events warrant.

Auto Support *

AutoSupport indicates willingness to participate in NetApp's proactive support application, NetApp Active IQ. The default election is true and indicates support data will be sent to NetApp. An empty or blank election is the same as a default election. Air gapped installations should enter false.

First Name

HCG

The first name of the SRE supporting Astra.

Last Name

Admin

The last name of the SRE supporting Astra.

Image Registry

The container image registry that is hosting the Astra application images, ACC Operator and ACC Helm Repository.

Name

astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra

The name of the image registry. For example "example.registry/astra". Do not prefix with protocol.

Secret

astra-registry-cred

The name of the Kubernetes secret that will authenticate with the image registry.

Volume Reclaim Policy

Retain

Reclaim policy to be set for persistent volumes

Astra Resources Scaler

Default

Scaling options for AstraControlCenter Resource limits.

Storage Class

The storage class to be used for PVCs. If not set, default storage class will be used.

Crds

Options for how ACC should handle CRDs.

Create Cancel

自動[可執行]

1. 若要使用Ansible教戰手冊來部署Astra Control Center、您需要安裝Ansible的Ubuntu / RHEL機器。請依照程序進行 "請按這裡" 適用於 Ubuntu 和 RHEL 。
2. 複製裝載可執行內容的GitHub儲存庫。

```
git clone https://github.com/NetApp-
Automation/na_astra_control_suite.git
```

3. 登入NetApp支援網站、下載最新版的NetApp Astra Control Center。若要這麼做、必須附上NetApp帳戶的授權。下載後、將其傳輸至工作站。

> 若要開始使用Astra Control試用授權、請造訪 "Astra註冊網站"。

4. 建立或取得具有OpenShift叢集管理員存取權的Kbeconfig檔案、以安裝Astra Control Center。
5. 將目錄變更為na_astra_control_suITE。

```
cd na_astra_control_suite
```

6. 編輯「vars/vars.yml」檔案、並在變數中填入所需資訊。

```
#Define whether or not to push the Astra Control Center images to
your private registry [Allowed values: yes, no]
push_images: yes

#The directory hosting the Astra Control Center installer
installer_directory: /home/admin/

#Specify the ingress type. Allowed values - "AccTraefik" or
"Generic"
#"AccTraefik" if you want the installer to create a LoadBalancer
type service to access ACC, requires MetalLB or similar.
#"Generic" if you want to create or configure ingress controller
yourself, installer just creates a ClusterIP service for traefik.
ingress_type: "AccTraefik"

#Name of the Astra Control Center installer (Do not include the
extension, just the name)
astra_tar_ball_name: astra-control-center-22.04.0

#The complete path to the kubeconfig file of the
kubernetes/openshift cluster Astra Control Center needs to be
installed to.
hosting_k8s_cluster_kubeconfig_path: /home/admin/cluster-
kubeconfig.yml

#Namespace in which Astra Control Center is to be installed
astra_namespace: netapp-astra-cc

#Astra Control Center Resources Scaler. Leave it blank if you want
to accept the Default setting.
astra_resources_scaler: Default

#Storageclass to be used for Astra Control Center PVCs, it must be
created before running the playbook [Leave it blank if you want the
PVCs to use default storageclass]
astra_trident_storageclass: basic

#Reclaim Policy for Astra Control Center Persistent Volumes [Allowed
values: Retain, Delete]
storageclass_reclaim_policy: Retain
```

```
#Private Registry Details
astra_registry_name: "docker.io"

#Whether the private registry requires credentials [Allowed values:
yes, no]
require_reg_creds: yes

#If require_reg_creds is yes, then define the container image
registry credentials
#Usually, the registry namespace and usernames are same for
individual users
astra_registry_namespace: "registry-user"
astra_registry_username: "registry-user"
astra_registry_password: "password"

#Kuberenets/OpenShift secret name for Astra Control Center
#This name will be assigned to the K8s secret created by the
playbook
astra_registry_secret_name: "astra-registry-credentials"

#Astra Control Center FQDN
acc_fqdn_address: astra-control-center.cie.netapp.com

#Name of the Astra Control Center instance
acc_account_name: ACC Account Name

#Administrator details for Astra Control Center
admin_email_address: admin@example.com
admin_first_name: Admin
admin_last_name: Admin
```

7. 執行教戰手冊以部署Astra Control Center。本方針要求特定組態具備root權限。

   如果執行方針的使用者是root或設定了無密碼Sudo、請執行下列命令來執行方針。

```
ansible-playbook install_acc_playbook.yml
```

   如果使用者已設定以密碼為基礎的Sudo存取、請執行下列命令來執行方針、然後輸入Sudo密碼。

```
ansible-playbook install_acc_playbook.yml -K
```

1. 安裝可能需要幾分鐘的時間才能完成。確認「NetApp-Astra -cc」命名空間中的所有Pod和服務均已啟動並正在執行。

```
[netapp-user@rhel7 ~]$ oc get all -n netapp-astra-cc
```

2. 檢查「acc oper-manager-manager」記錄、確保安裝完成。

```
[netapp-user@rhel7 ~]$ oc logs deploy/acc-operator-controller-manager -n
netapp-acc-operator -c manager -f
```

ⓘ 下列訊息表示Astra Control Center安裝成功。

```
{"level":"info","ts":1624054318.029971,"logger":"controllers.AstraContro
lCenter","msg":"Successfully Reconciled AstraControlCenter in
[seconds]s","AstraControlCenter":"netapp-astra-
cc/astra","ae.Version":"[21.12.60]"}
```

3. 登入Astra Control Center的使用者名稱是CRD檔案中所提供系統管理員的電子郵件地址、密碼是附加於Astra Control Center UUID的字串「ACC-」。執行下列命令：

```
[netapp-user@rhel7 ~]$ oc get astracontrolcenters -n netapp-astra-cc
NAME    UUID
astra   345c55a5-bf2e-21f0-84b8-b6f2bce5e95f
```

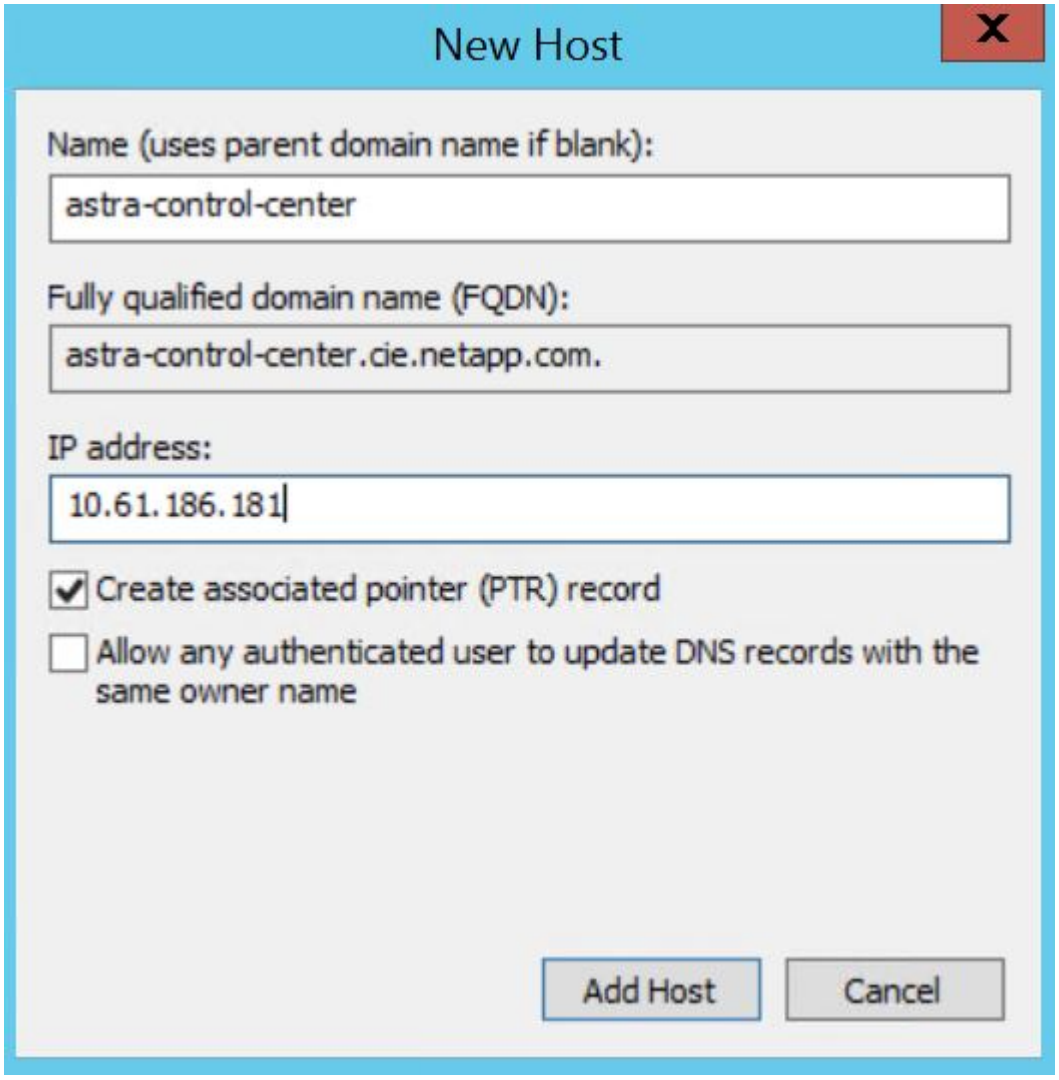
ⓘ 在此範例中、密碼為「ACC-345c55a5-bf2e-21f0-843b8-b6f2bce5e95f」。

4. 取得truefik服務負載平衡器IP。

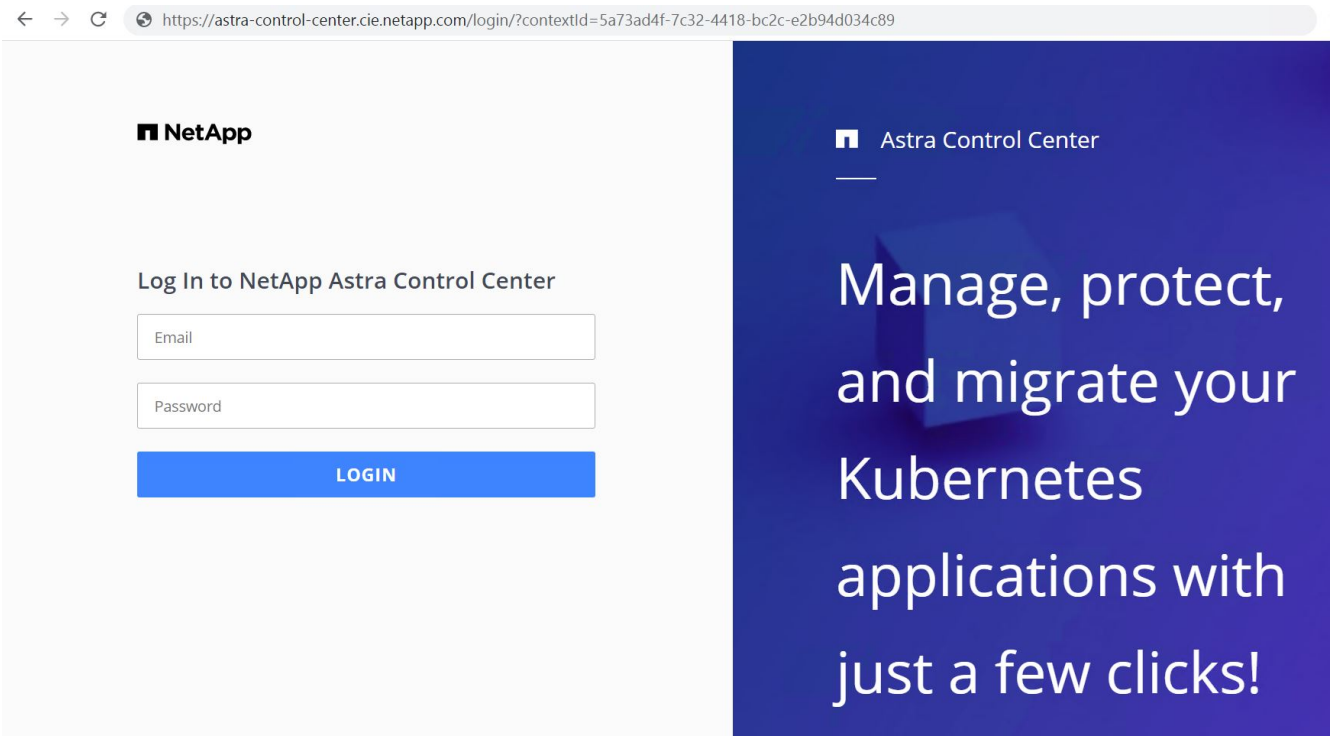
```
[netapp-user@rhel7 ~]$ oc get svc -n netapp-astra-cc | egrep
'EXTERNAL|traefik'

NAME                                          TYPE           CLUSTER-IP
EXTERNAL-IP     PORT(S)
AGE
traefik                                       LoadBalancer   172.30.99.142
10.61.186.181   80:30343/TCP,443:30060/TCP
16m
```

5. 在DNS伺服器中新增一個項目、將Astra Control Center CRD檔案中提供的FQDN指向raefik服務的「exter-

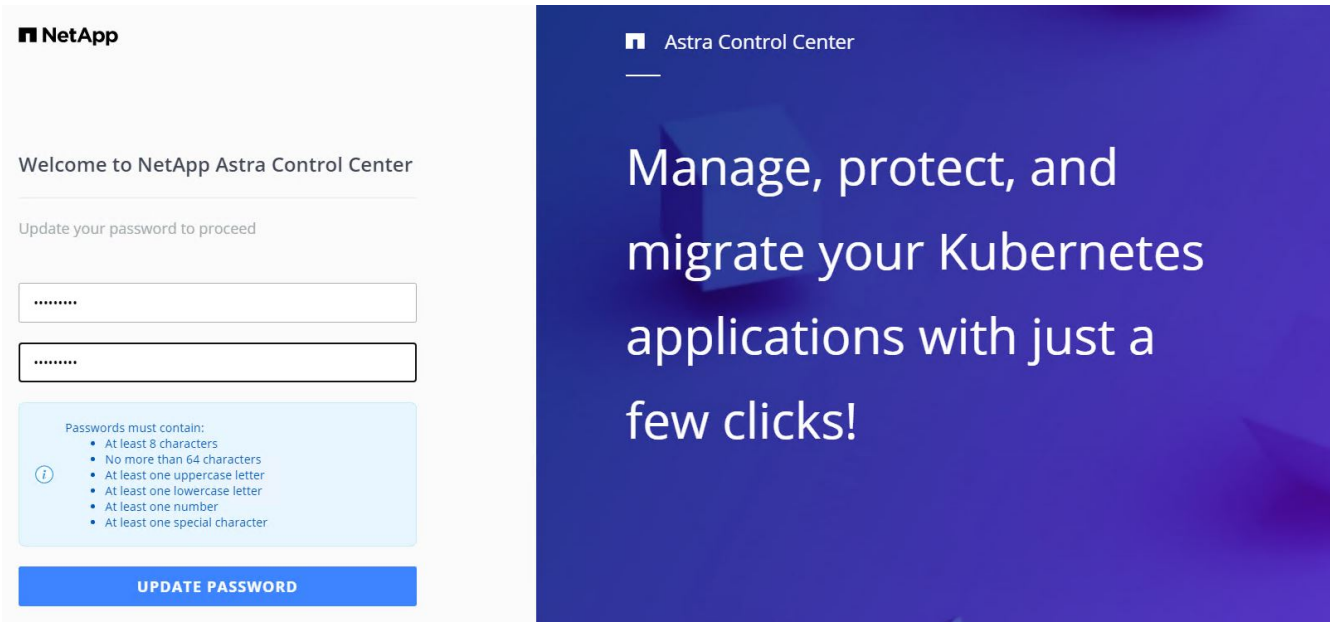
IP」。



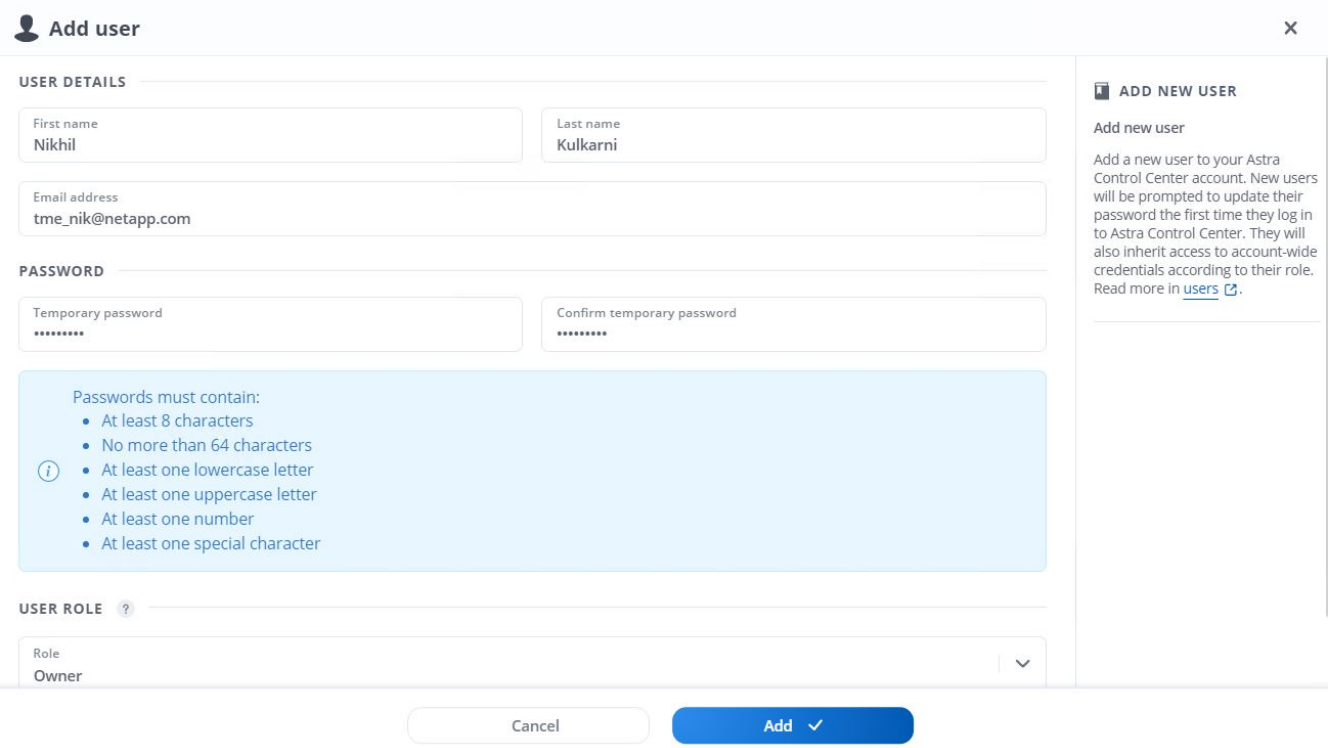
6. 瀏覽Astra Control Center GUI的FQDN即可登入。



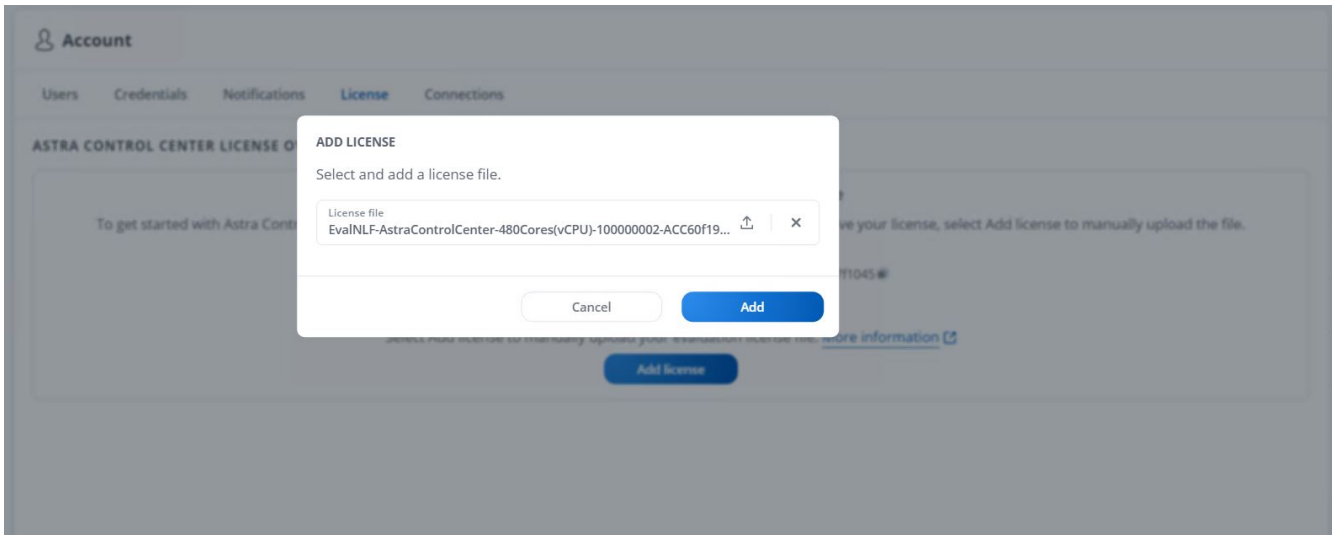
7. 第一次使用CRD提供的管理電子郵件地址登入Astra Control Center GUI時、您需要變更密碼。



8. 如果您想要新增使用者至Astra Control Center、請瀏覽至「帳戶」>「使用者」、按一下「新增」、輸入使用者的詳細資料、然後按一下「新增」。



9. Astra Control Center需要取得授權、才能讓所有IT功能正常運作。若要新增授權、請瀏覽至「帳戶」>「授權」、按一下「新增授權」、然後上傳授權檔案。



> 如果您在安裝或組態NetApp Astra Control Center時遇到問題、我們將提供已知問題的知識庫"請按這裡"。

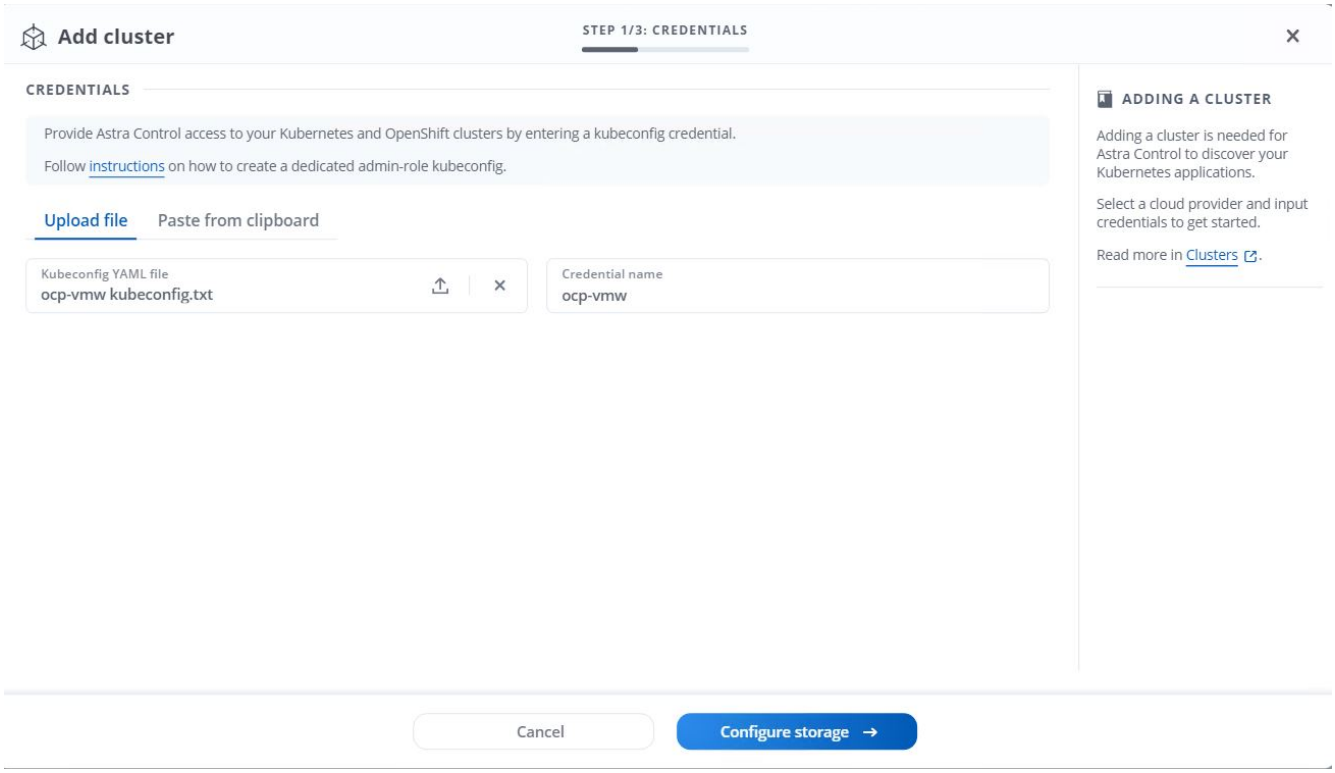
使用**Astra Control Center**註冊**Red Hat OpenShift**叢集

若要讓Astra Control Center管理工作負載、您必須先登錄Red Hat OpenShift叢集。

登錄**Red Hat OpenShift**叢集

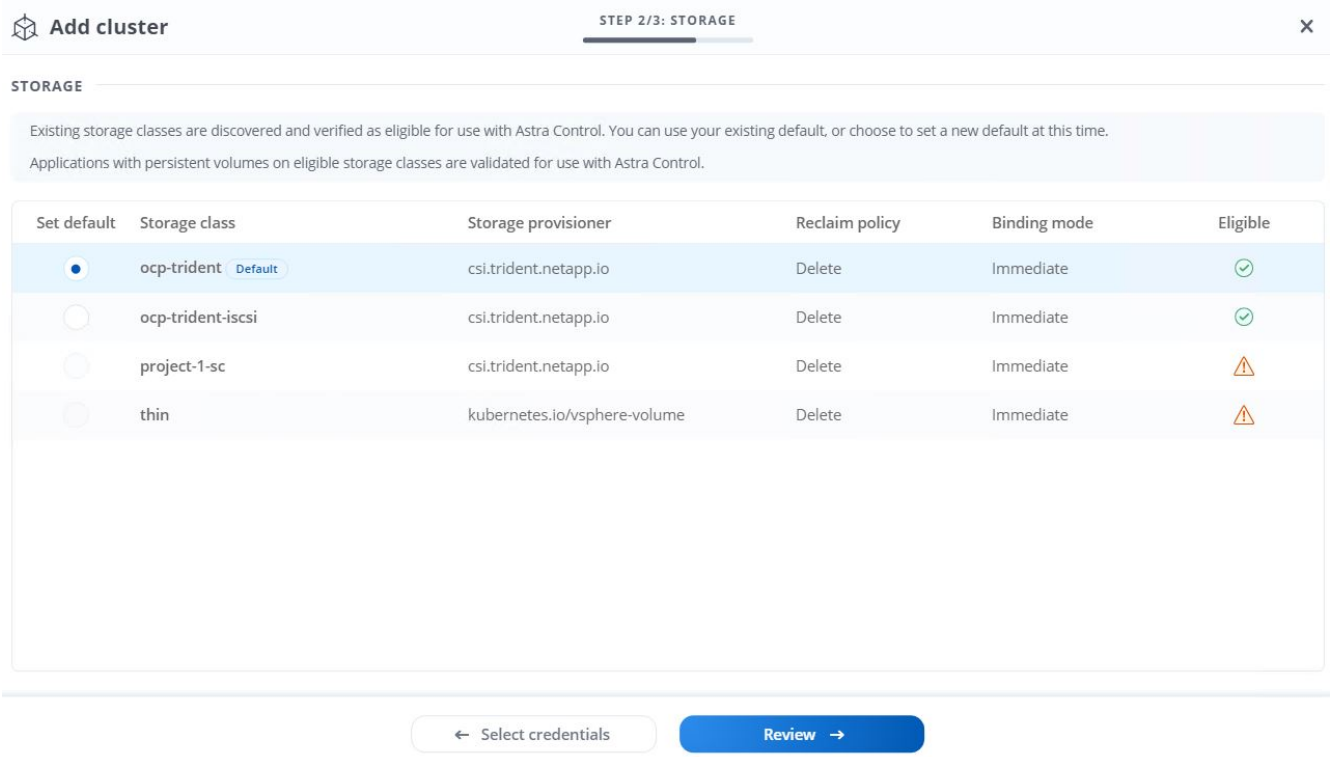
1. 第一步是將OpenShift叢集新增至Astra控制中心、並加以管理。移至「叢集」、然後按一下「新增叢集」、

上傳OpenShift叢集的Kbeconfig檔案、然後按一下「選取儲存設備」。

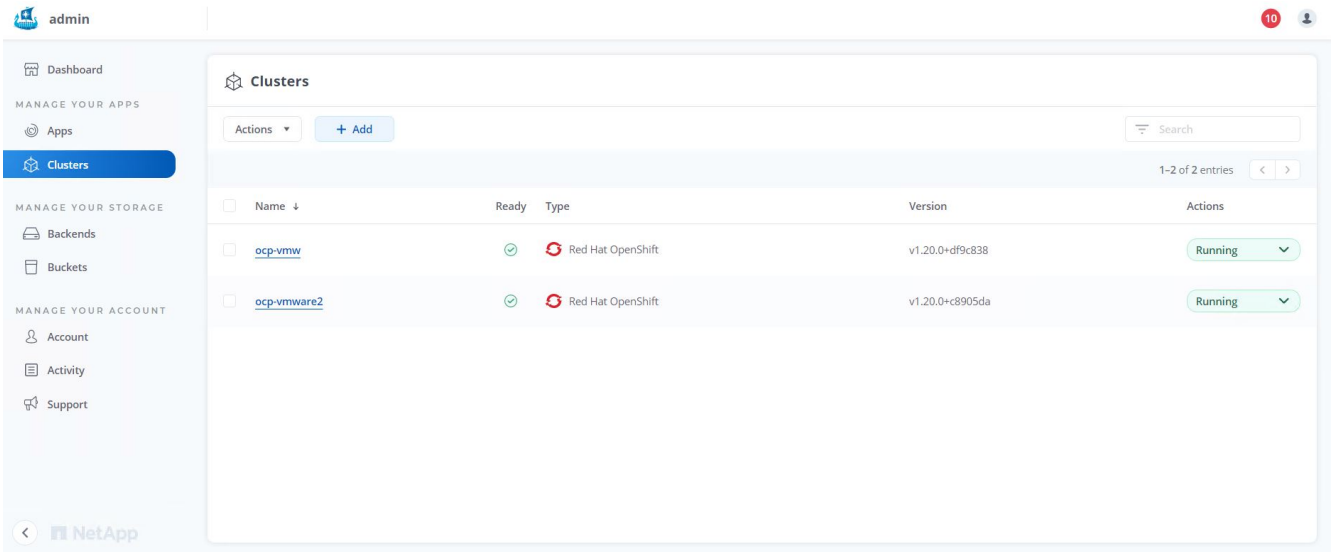


> 可以產生Kbeconfig檔案、以使用者名稱和密碼或權杖進行驗證。權杖會在有限時間後過期、並可能使登錄的叢集無法連線。NetApp建議使用用戶名和密碼的Kusbeconfig檔案、將OpenShift叢集登錄至Astra Control Center。

2. Astra Control Center會偵測合格的儲存類別。現在、請在NetApp ONTAP 上選擇使用Trident（以SVM為後盾）來配置Volume的方式、然後按一下「Review（檢閱）」。在下一個窗格中、確認詳細資料、然後按一下「Add Cluster（新增叢集）」。

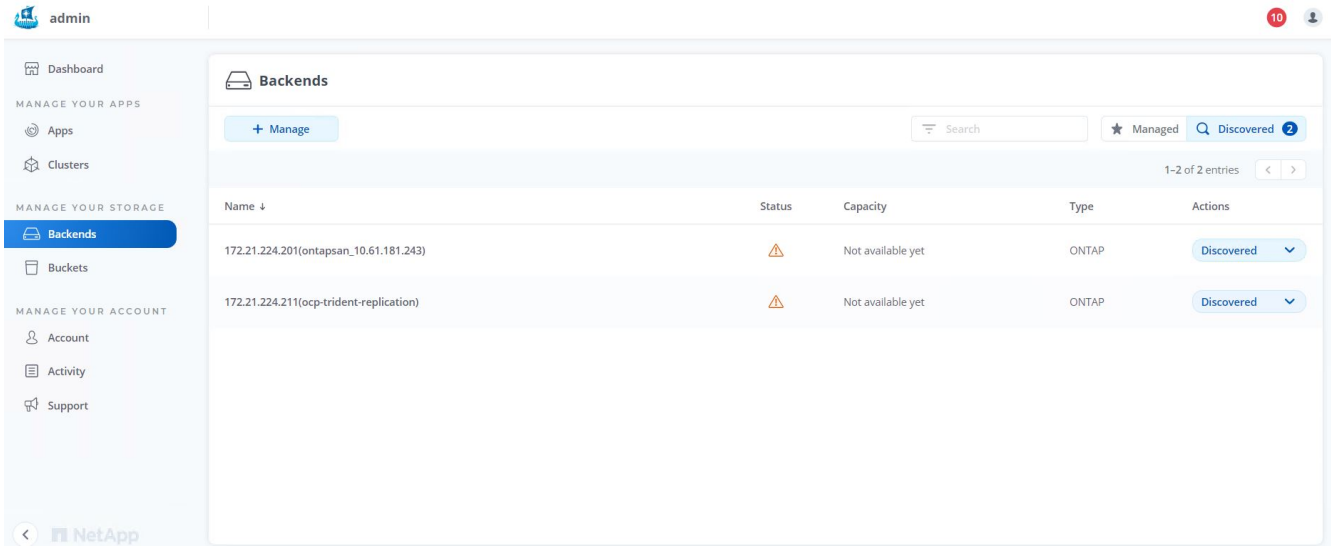


3. 如步驟1所述、登錄兩個OpenShift叢集。新增時、叢集會移至「Discovering」（探索）狀態、而Astra Control Center則會檢查並安裝必要的代理程式。叢集狀態會在成功登錄之後變更為執行中。

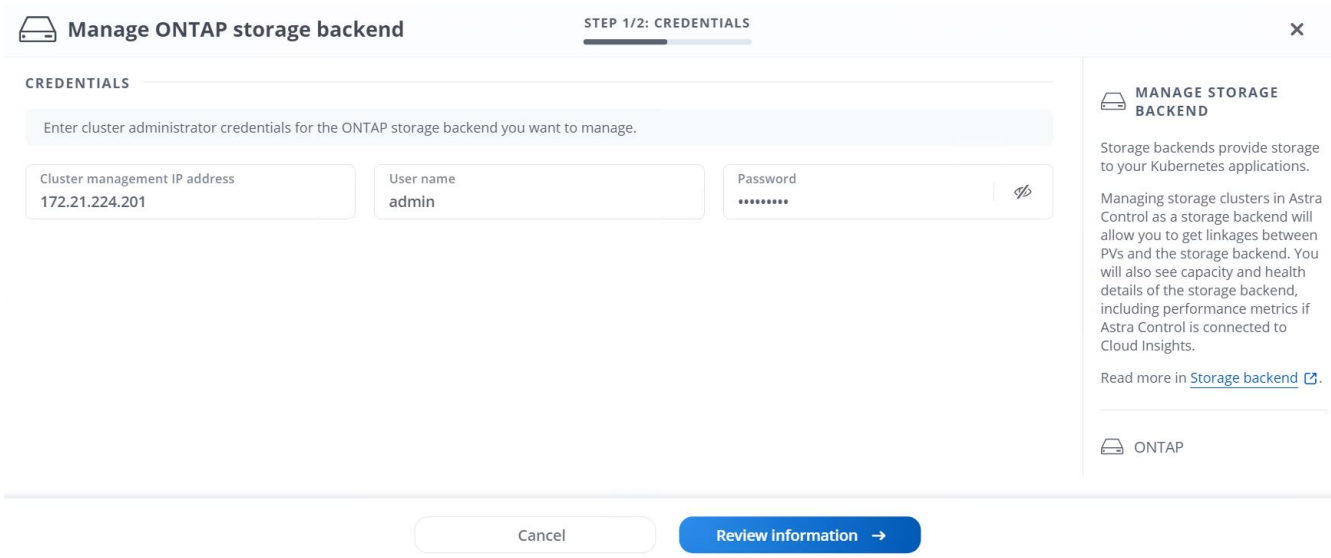


> 所有由Astra Control Center管理的Red Hat OpenShift叢集、都應該能夠存取安裝時所用的映像登錄、因為安裝在受管理叢集上的代理程式會從該登錄擷取映像。

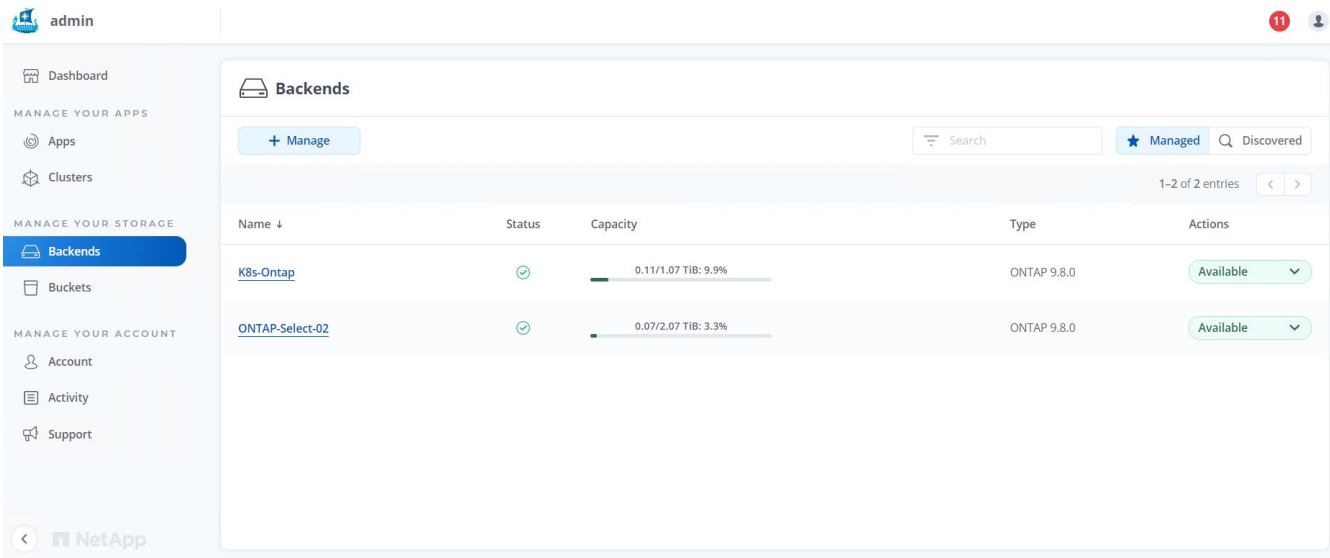
4. 將ONTAP 支援的物件叢集匯入為儲存資源、以由Astra Control Center作為後端管理。當OpenShift叢集新增至Astra且已設定儲存設備時、它會自動探索ONTAP 並檢查以儲存設備為後盾的不支援該叢集、但不會將其匯入要管理的Astra Control Center。



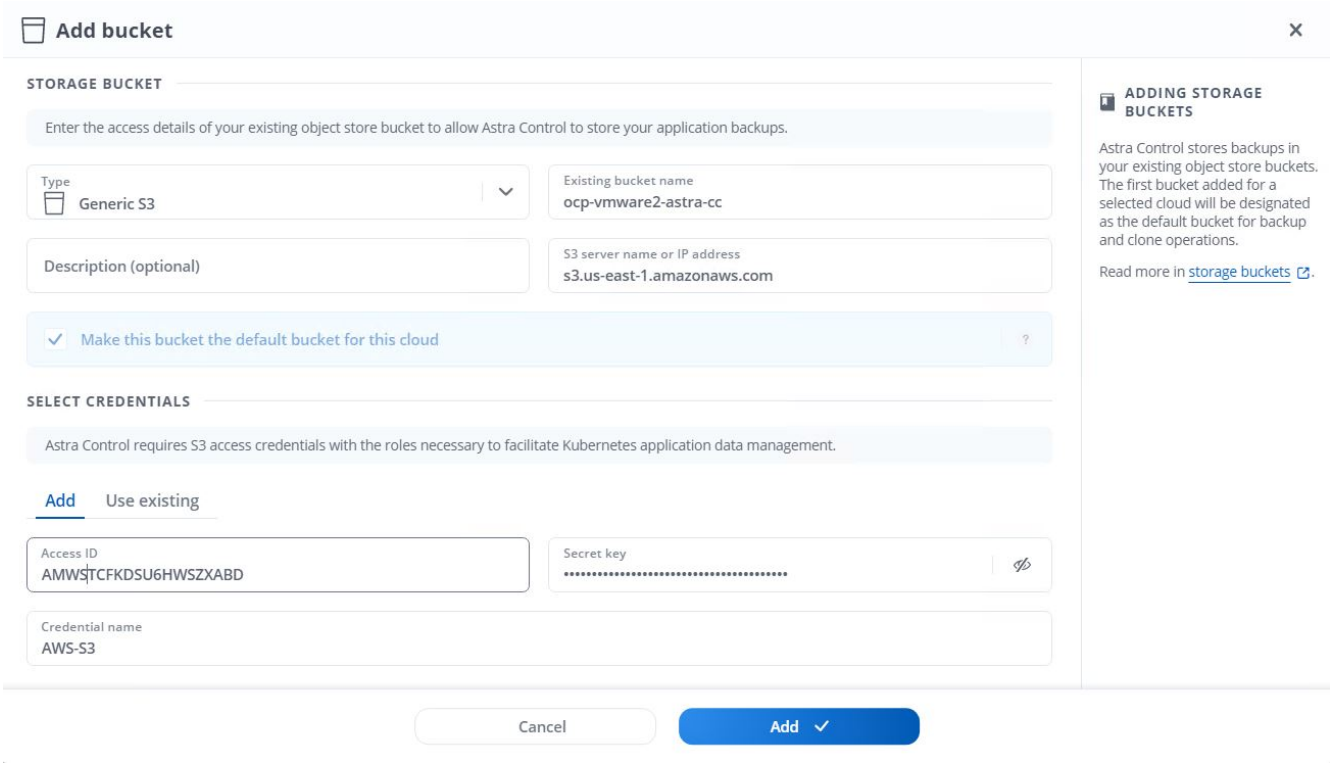
5. 若要匯入ONTAP 物件叢集、請前往後端、按一下下拉式清單、然後選取ONTAP 要管理之物件叢集旁的「Manage（管理）」。輸入ONTAP 「物件叢集認證」、按一下「檢閱資訊」、然後按一下「匯入儲存設備後端」。



6. 新增後端之後、狀態會變更為「可用」。這些後端現在有OpenShift叢集中持續磁碟區的相關資訊、ONTAP以及在整個系統上對應的磁碟區。



7. 若要使用Astra Control Center跨OpenShift叢集進行備份與還原、您必須配置支援S3傳輸協定的物件儲存桶。目前支援ONTAP 的選項包括不支援的S3、StorageGRID 不支援的功能、以及AWS S3。為了進行此安裝、我們將設定AWS S3儲存區。移至「Bucket」、按一下「Add Bucket」、然後選取「通用S3」。輸入S3儲存區及認證的詳細資料以進行存取、按一下「將此儲存區設為雲端的預設儲存區」核取方塊、然後按一下「新增」。



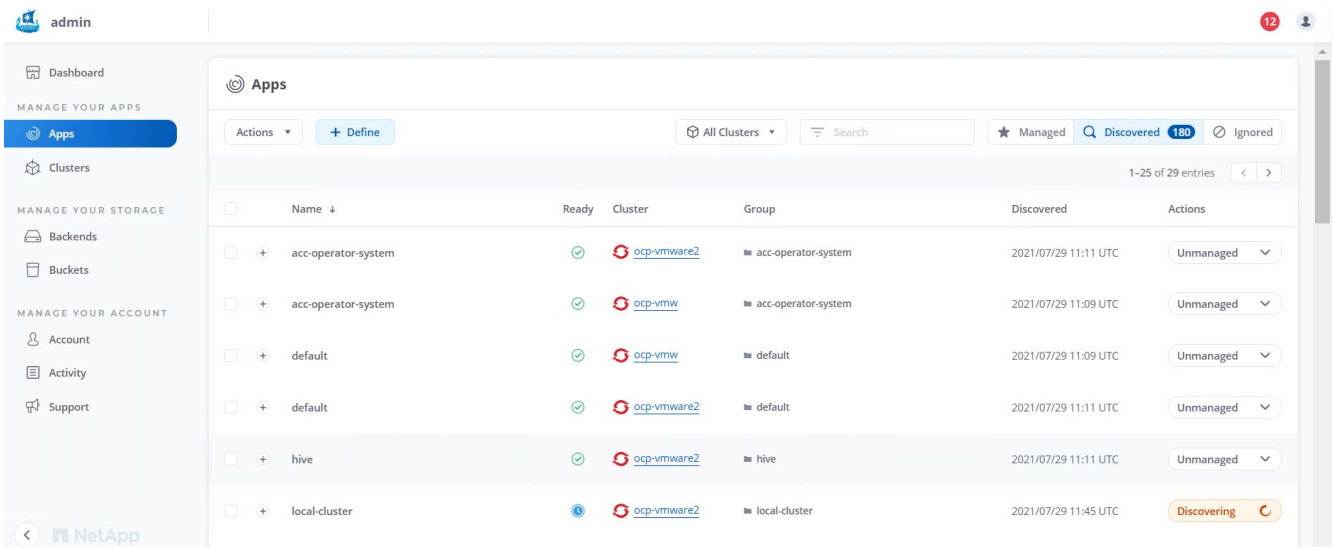
選擇要保護的應用程式

註冊Red Hat OpenShift叢集之後、您可以探索已部署的應用程式、並透過Astra Control Center進行管理。

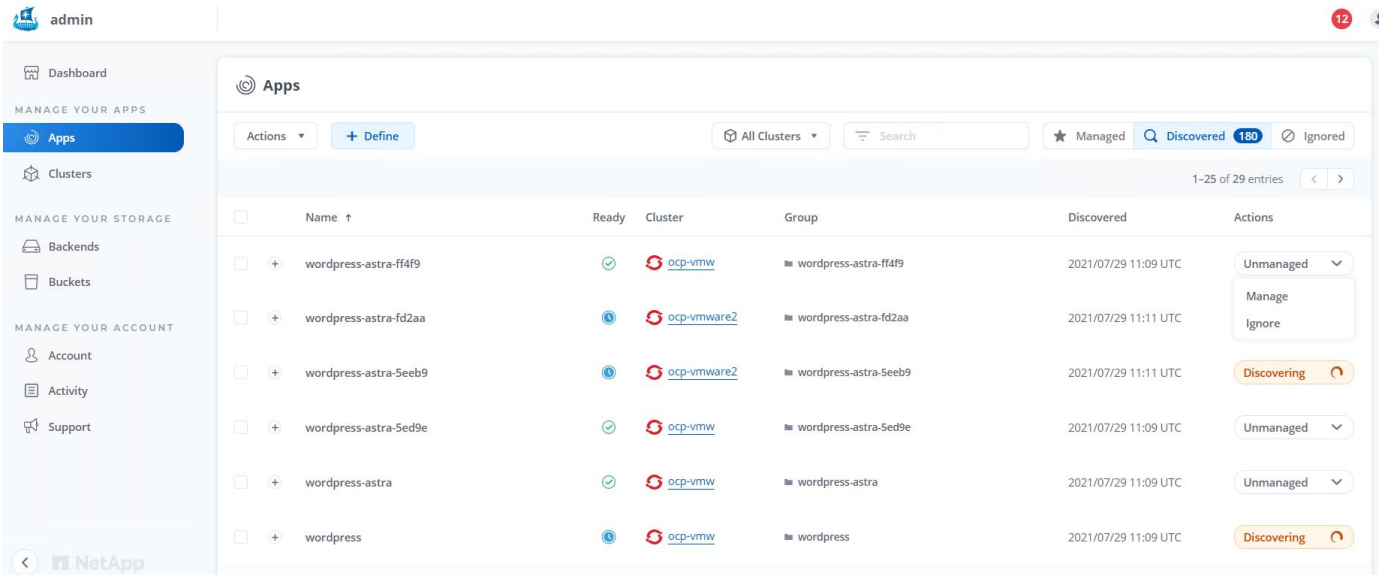
管理應用程式

1. 在ONTAP Astra Control Center登錄OpenShift叢集和還原後端之後、控制中心會自動開始探索所有命名空間

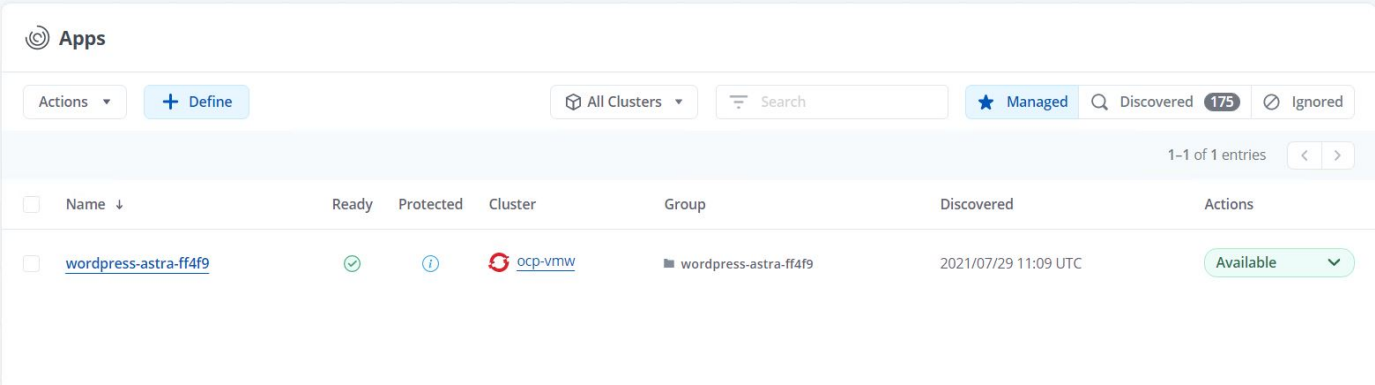
中的應用程式、這些命名空間使用以指定ONTAP 的支援功能後端設定的儲存機架。



2. 瀏覽至「應用程式>已探索」、然後按一下您要使用Astra管理的應用程式旁的下拉式功能表。然後按一下「管理」。



1. 應用程式會進入可用狀態、並可在「應用程式」區段的「受管理」索引標籤下檢視。



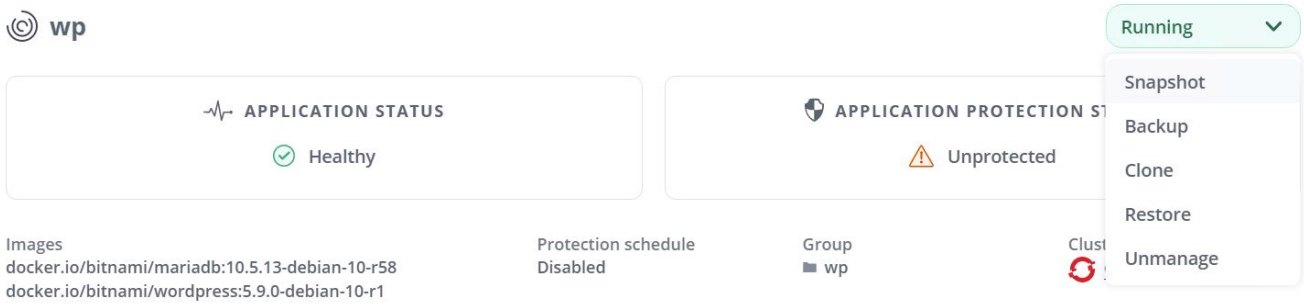
保護您的應用程式

由Astra Control Center管理應用程式工作負載之後、您可以設定這些工作負載的保護設定。

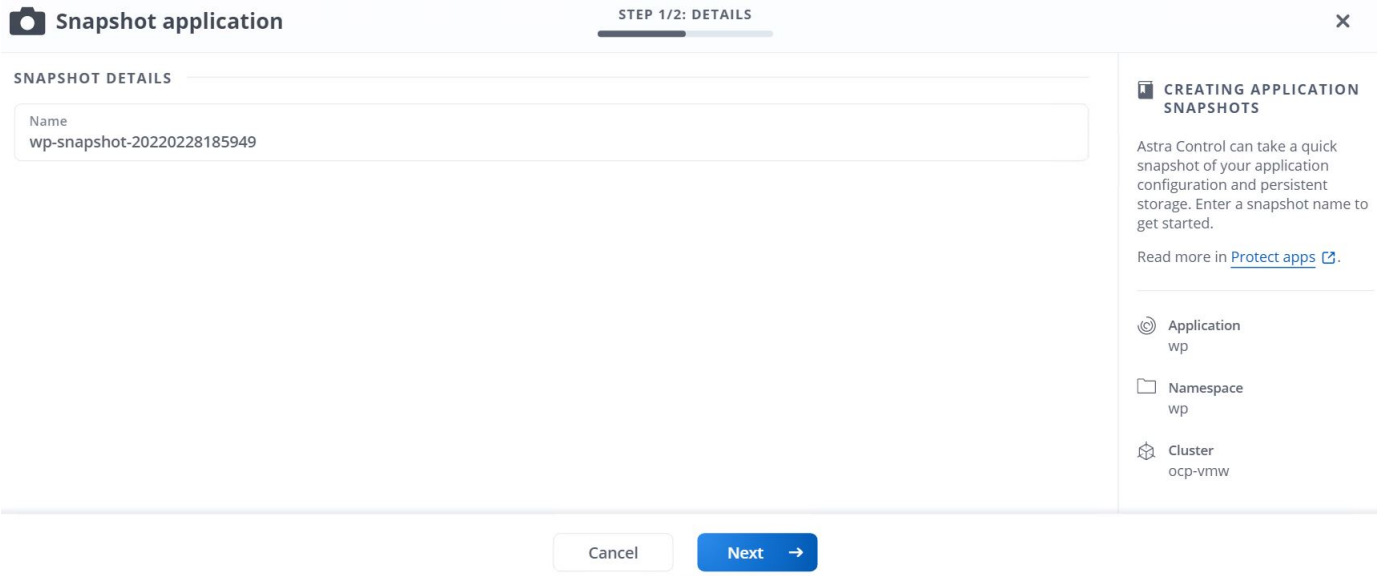
建立應用程式快照

應用程式的快照會建立ONTAP 一份「不含任何資料的Snapshot」複本、以便根據該Snapshot複本、將應用程式還原或複製到特定時間點。

1. 若要擷取應用程式的快照、請瀏覽至「應用程式」>「受管理的」索引標籤、然後按一下您要製作Snapshot複本的應用程式。按一下應用程式名稱旁的下拉式功能表、然後按一下「Snapshot（快照）」。



2. 輸入快照詳細資料、按一下「下一步」、然後按一下「Snapshot（快照）」。建立快照大約需要一分鐘、快照成功建立之後、狀態就會變成可用。



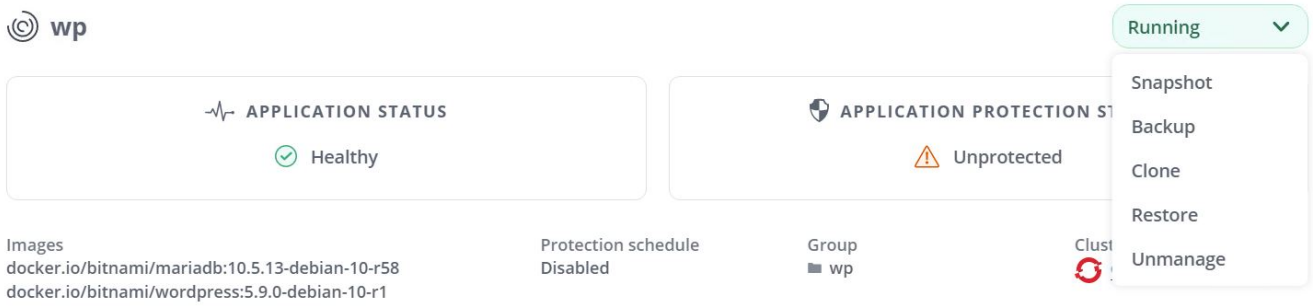
建立應用程式備份

應用程式的備份會擷取應用程式的作用中狀態及其資源組態、將其封裝到檔案中、並將其儲存在遠端物件儲存庫中。

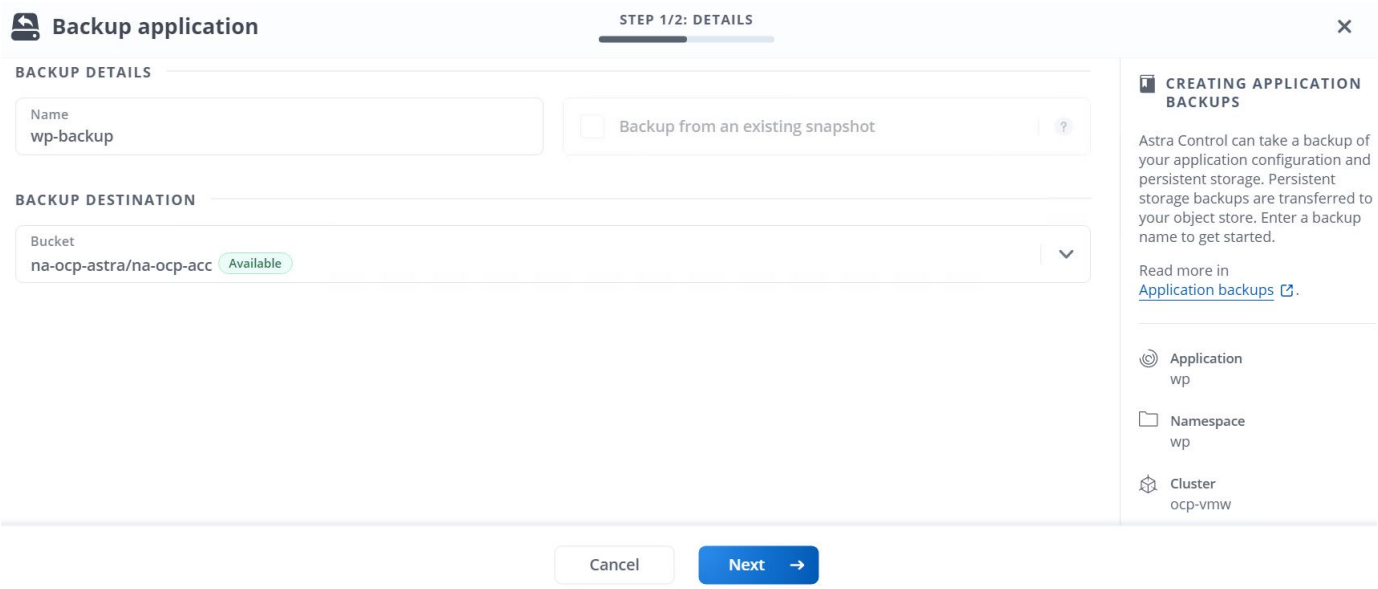
若要在Astra Control Center中備份及還原託管應用程式、您必須先設定支援ONTAP 的支援功能系統的超級使用者設定。若要這麼做、請輸入下列命令。

```
ONTAP::> export-policy rule modify -vserver ocp-trident -policyname
default -ruleindex 1 -superuser sys
ONTAP::> export-policy rule modify -policyname default -ruleindex 1 -anon
65534 -vserver ocp-trident
```

1. 若要在Astra Control Center中建立受管理應用程式的備份、請瀏覽至「應用程式」>「受管理的」索引標籤、然後按一下您要備份的應用程式。按一下應用程式名稱旁的下拉式功能表、然後按一下備份。



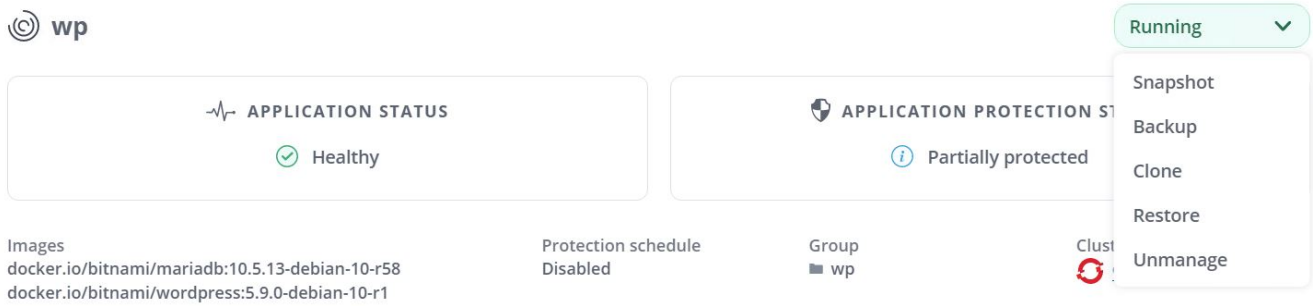
2. 輸入備份詳細資料、選取要保留備份檔案的物件儲存區、按一下「Next（下一步）」、然後在檢閱詳細資料之後、按一下「Backup（備份）」。視應用程式和資料的大小而定、備份可能需要數分鐘的時間、備份成功完成後、備份狀態就會變成可用狀態。



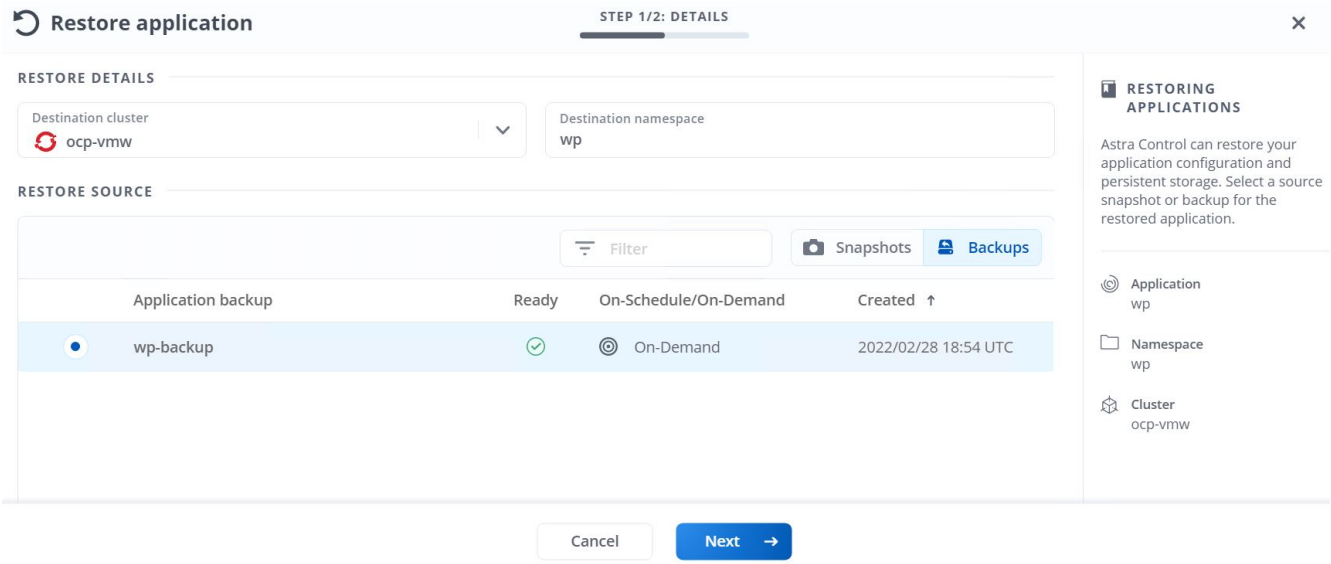
還原應用程式

只要按一下按鈕、就能將應用程式還原至同一個叢集中的原始命名空間、或還原至遠端叢集、以供應用程式保護和災難恢復之用。

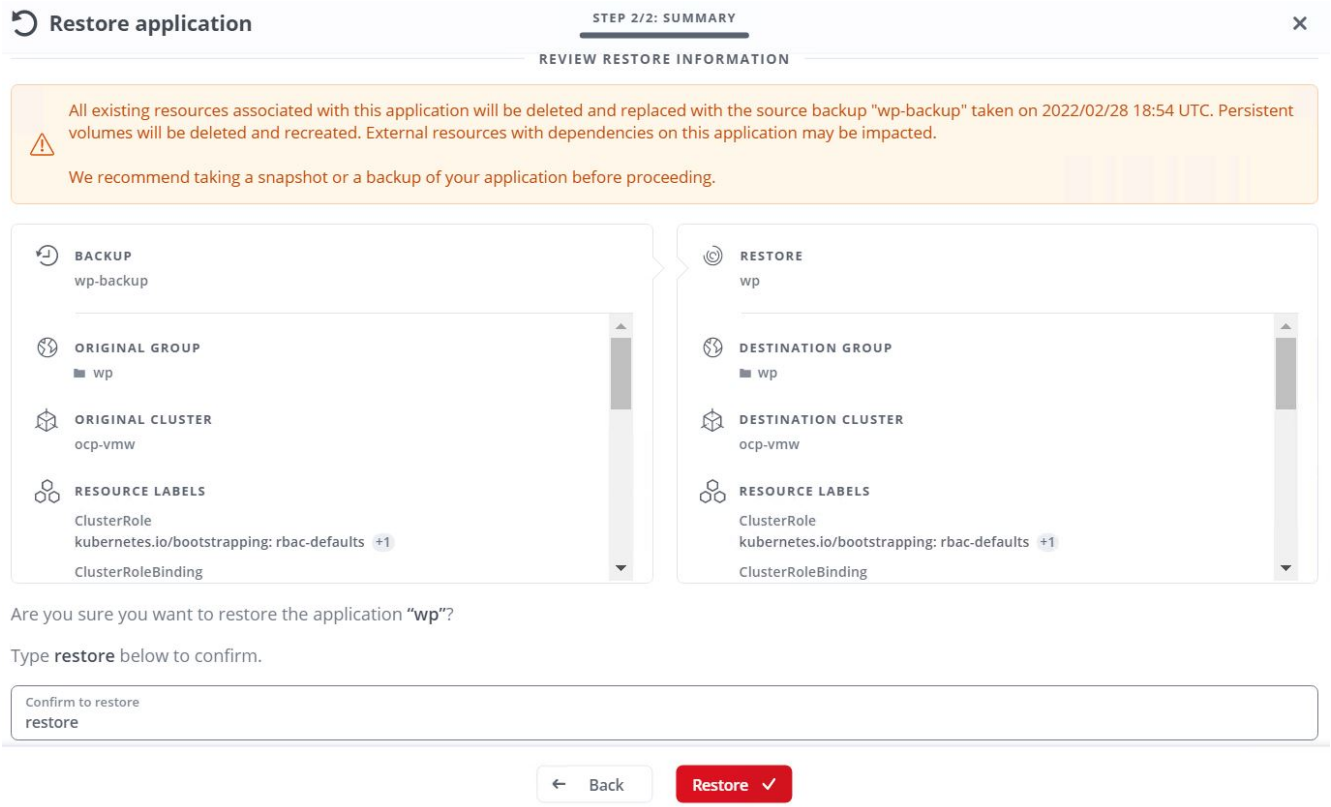
1. 若要還原應用程式、請瀏覽至「應用程式」>「受管理的」索引標籤、然後按一下有問題的應用程式。按一下應用程式名稱旁的下拉式功能表、然後按一下「還原」。



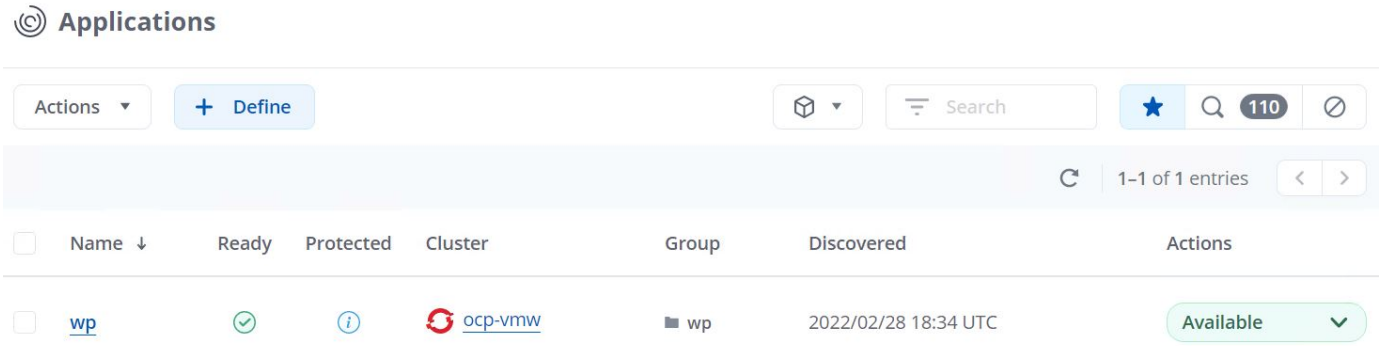
2. 輸入還原命名空間的名稱、選取您要還原的叢集、然後選擇是要從現有的快照或應用程式備份還原命名空間。按一下「下一步」



3. 在檢閱窗格中、輸入「重新儲存」、並在檢閱詳細資料後按一下「還原」。



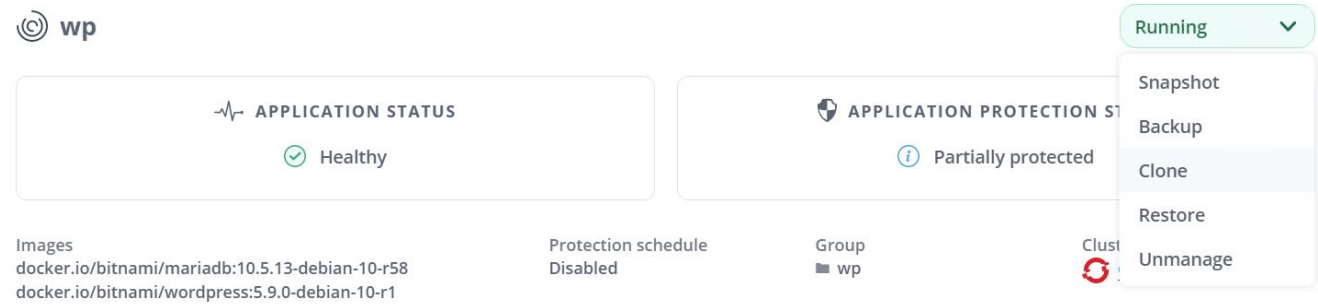
4. 新應用程式會在Astra Control Center還原所選叢集上的應用程式時、進入還原狀態。Astra安裝並偵測應用程式的所有資源之後、應用程式會進入可用狀態。



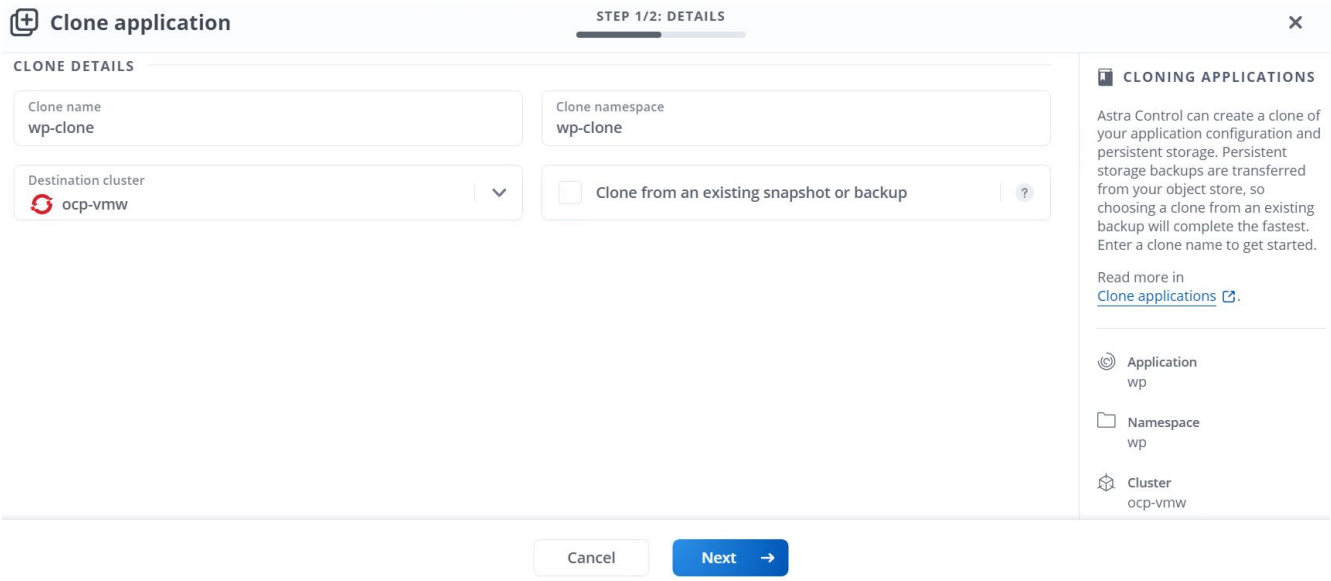
複製應用程式

您可以將應用程式複製到原始叢集或遠端叢集、以供開發/測試或應用程式保護及災難恢復之用。在同一個儲存後端的同一個叢集內複製應用程式時、會使用NetApp FlexClone技術來即時複製PVCS、並節省儲存空間。

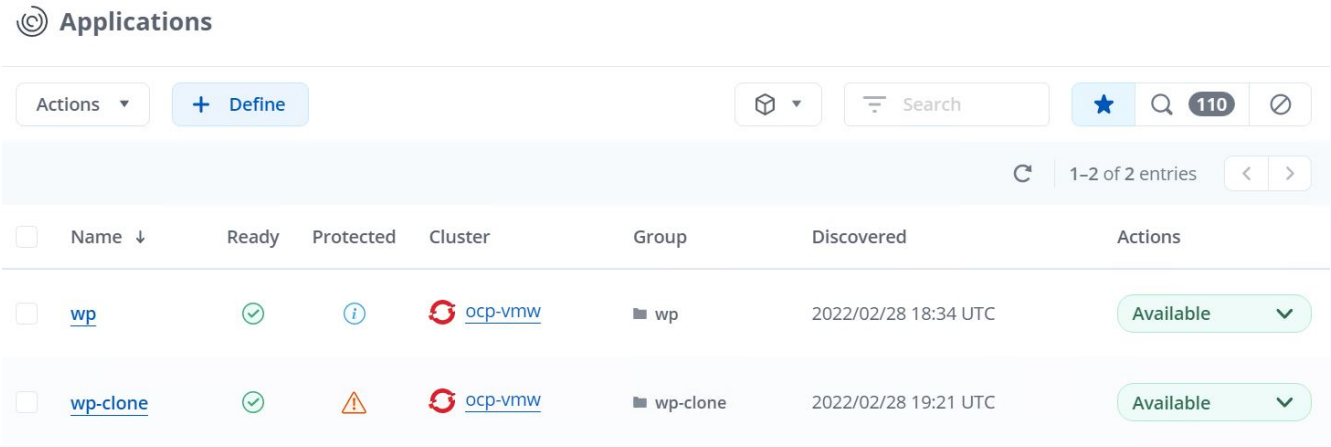
1. 若要複製應用程式、請瀏覽至「應用程式」>「受管理」索引標籤、然後按一下有問題的應用程式。按一下應用程式名稱旁的下拉式功能表、然後按一下Clone（複製）。



2. 輸入新命名空間的詳細資料、選取要複製到的叢集、然後選擇是否要從現有的快照或備份複製、或是從應用程式的目前狀態複製。檢閱詳細資料後、按一下「下一步」、然後按一下「檢閱窗格上的Clone（複製）」。



3. 當Astra Control Center在所選叢集上建立應用程式時、新的應用程式會進入「Discovering（探索）」狀態。Astra安裝並偵測應用程式的所有資源之後、應用程式會進入可用狀態。

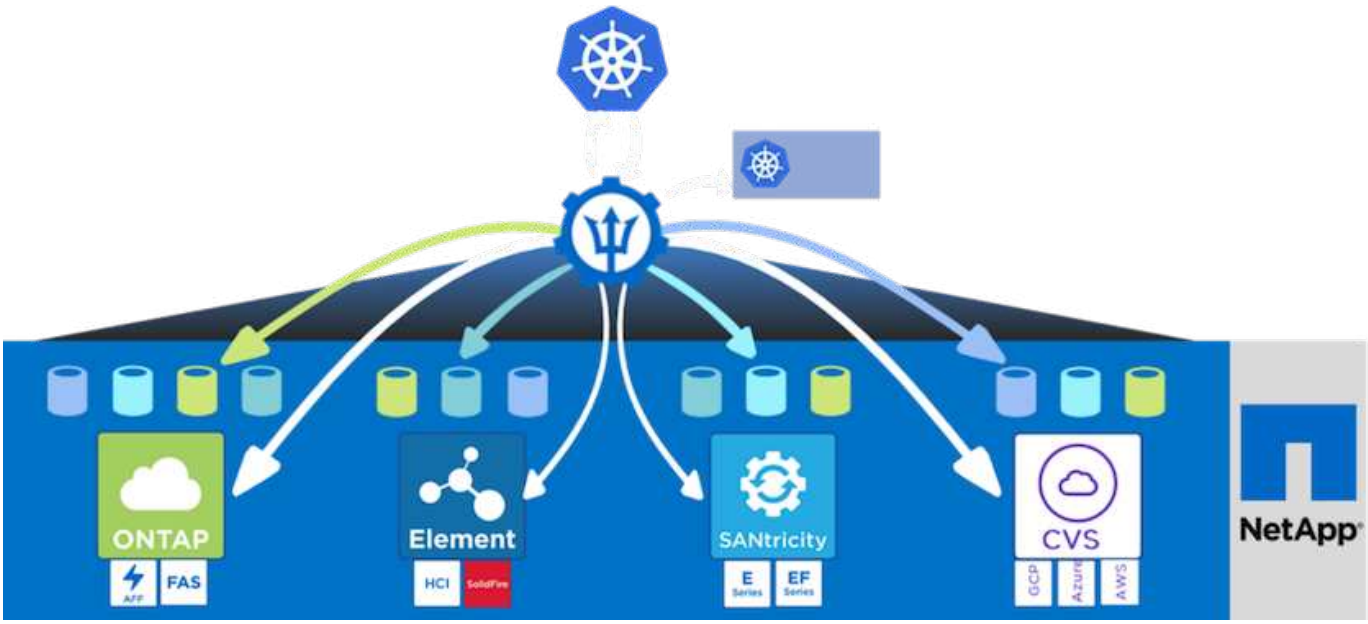


Astra Trident總覽

Astra Trident是開放原始碼且完全支援的儲存協調工具、適用於Container和Kubernetes配送、包括Red Hat OpenShift。Trident可搭配整個NetApp儲存產品組合（包括NetApp ONTAP 的整套和Element儲存系統）使用、

也支援NFS和iSCSI連線。Trident可讓終端使用者從NetApp儲存系統配置及管理儲存設備、而無需儲存管理員介入、進而加速DevOps工作流程。

系統管理員可根據專案需求和儲存系統模型來設定多個儲存後端、以啟用進階儲存功能、包括壓縮、特定磁碟類型或QoS層級、以保證特定層級的效能。定義後端後端之後、開發人員可在專案中使用這些後端來建立持續磁碟區宣告（PVCS）、並視需要將持續儲存附加至容器。



Astra Trident的開發週期很快、就像Kubernetes一樣、每年發行四次。

最新版的Astra Trident於2022年1月推出22.01版。支援對照表、顯示哪些版本的Trident已通過測試、可找到Kubernetes經銷產品 "請按這裡"。

從20.04版本開始、Trident設定由Trident操作員執行。營運者讓大規模部署變得更簡單、並提供額外支援、包括在Trident安裝過程中部署的Pod自我修復。

有了21.01版、我們提供了Helm圖表、方便您安裝Trident操作員。

下載**Astra Trident**

若要在已部署的使用者叢集上安裝Trident並佈建持續磁碟區、請完成下列步驟：

1. 將安裝歸檔檔案下載至管理工作站、並擷取內容。目前的Trident版本為22.01、可下載 "請按這裡"。

```
[netapp-user@rhel7 ~]$ wget
https://github.com/NetApp/trident/releases/download/v22.01.0/trident-
installer-22.01.0.tar.gz
--2021-05-06 15:17:30--
https://github.com/NetApp/trident/releases/download/v22.01.0/trident-
installer-22.01.0.tar.gz
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
```

```
Location: https://github-
releases.githubusercontent.com/77179634/a4fa9f00-a9f2-11eb-9053-
98e8e573d4ae?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210506%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20210506T191643Z&X-Amz-Expires=300&X-
Amz-
Signature=8a49a2a1e08c147d1ddd8149ce45a5714f9853fee19bb1c507989b9543eb36
30&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-
content-disposition=attachment%3B%20filename%3Dtrident-installer-
22.01.0.tar.gz&response-content-type=application%2Foctet-stream
[following]
--2021-05-06 15:17:30--  https://github-
releases.githubusercontent.com/77179634/a4fa9f00-a9f2-11eb-9053-
98e8e573d4ae?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210506%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20210506T191643Z&X-Amz-Expires=300&X-
Amz-
Signature=8a49a2a1e08c147d1ddd8149ce45a5714f9853fee19bb1c507989b9543eb36
30&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-
content-disposition=attachment%3B%20filename%3Dtrident-installer-
22.01.0.tar.gz&response-content-type=application%2Foctet-stream
Resolving github-releases.githubusercontent.com (github-
releases.githubusercontent.com)... 185.199.108.154, 185.199.109.154,
185.199.110.154, ...
Connecting to github-releases.githubusercontent.com (github-
releases.githubusercontent.com)|185.199.108.154|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38349341 (37M) [application/octet-stream]
Saving to: ‘trident-installer-22.01.0.tar.gz’

100%[=====================================================================
==============================================>] 38,349,341  88.5MB/s
in 0.4s

2021-05-06 15:17:30 (88.5 MB/s) - ‘trident-installer-22.01.0.tar.gz’
saved [38349341/38349341]
```

2. 從下載的套裝組合中擷取Trident安裝。

```
[netapp-user@rhel7 ~]$ tar -xzf trident-installer-22.01.0.tar.gz
[netapp-user@rhel7 ~]$ cd trident-installer/
[netapp-user@rhel7 trident-installer]$
```

使用**Helm**安裝**Trident**運算子

1. 首先將使用者叢集的「kubeconfig」檔案位置設定為環境變數、這樣您就不需要參考它、因為Trident沒有傳遞此檔案的選項。

```
[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/ocp-
install/auth/kubeconfig
```

2. 在使用者叢集中建立Trident命名空間時、請執行Helm命令、從Lm目錄的tar安裝Trident運算子。

```
[netapp-user@rhel7 trident-installer]$ helm install trident
helm/trident-operator-22.01.0.tgz --create-namespace --namespace trident
NAME: trident
LAST DEPLOYED: Fri May  7 12:54:25 2021
NAMESPACE: trident
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thank you for installing trident-operator, which will deploy and manage
NetApp's Trident CSI
storage provisioner for Kubernetes.

Your release is named 'trident' and is installed into the 'trident'
namespace.
Please note that there must be only one instance of Trident (and
trident-operator) in a Kubernetes cluster.

To configure Trident to manage storage resources, you will need a copy
of tridentctl, which is
available in pre-packaged Trident releases.  You may find all Trident
releases and source code
online at https://github.com/NetApp/trident.

To learn more about the release, try:

  $ helm status trident
  $ helm get all trident
```

3. 您可以檢查在命名空間中執行的Pod、或使用tridentctl二進位檔檢查安裝的版本、以驗證Trident是否已成功安裝。

```
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-5z45l                   1/2     Running   2          30s
trident-csi-696b685cf8-htdb2        6/6     Running   0          30s
trident-csi-b74p2                   2/2     Running   0          30s
trident-csi-lrw4n                   2/2     Running   0          30s
trident-operator-7c748d957-gr2gw    1/1     Running   0          36s

[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
+----------------+----------------+
| SERVER VERSION | CLIENT VERSION |
+----------------+----------------+
| 22.01.0          | 22.01.0          |
+----------------+----------------+
```

在某些情況下、客戶環境可能需要自訂Trident部署。在這些情況下、您也可以手動安裝Trident運算子、並更新隨附的資訊清單、以自訂部署。

手動安裝**Trident**運算子

1. 首先、將使用者叢集的「kubeconfig」檔案位置設定為環境變數、這樣您就不需要參考、因為Trident沒有傳遞此檔案的選項。

```
[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/ocp-
install/auth/kubeconfig
```

2. 「Trident安裝程式」目錄包含定義所有必要資源的資訊清單。使用適當的資訊清單、建立「TridentOrchestrator」自訂資源定義。

```
[netapp-user@rhel7 trident-installer]$ oc create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
customresourcedefinition.apiextensions.k8s.io/tridentorchestrators.tride
nt.netapp.io created
```

3. 如果不存在、請使用提供的資訊清單、在叢集中建立Trident命名空間。

```
[netapp-user@rhel7 trident-installer]$ oc apply -f deploy/namespace.yaml
namespace/trident created
```

4. 建立Trident營運者部署所需的資源、例如營運者的「服務帳戶」、專屬的「PodSecurity Policy」、或營運者本身的「ClusterROLTE」和「ClusterROLESTBinding」。

```
[netapp-user@rhel7 trident-installer]$ oc create -f deploy/bundle.yaml
serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created
```

5. 您可以使用下列命令來檢查部署後的操作員狀態：

```
[netapp-user@rhel7 trident-installer]$ oc get deployment -n trident
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
trident-operator    1/1     1            1           23s
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                 READY   STATUS    RESTARTS   AGE
trident-operator-66f48895cc-lzczk    1/1     Running   0          41s
```

6. 部署營運者之後、我們就可以使用它來安裝Trident。這需要建立「TridentOrchestrator」。

```
[netapp-user@rhel7 trident-installer]$ oc create -f
deploy/crds/tridentorchestrator_cr.yaml
tridentorchestrator.trident.netapp.io/trident created
[netapp-user@rhel7 trident-installer]$ oc describe torc trident
Name:         trident
Namespace:
Labels:       <none>
Annotations:  <none>
API Version:  trident.netapp.io/v1
Kind:         TridentOrchestrator
Metadata:
  Creation Timestamp:  2021-05-07T17:00:28Z
  Generation:          1
  Managed Fields:
    API Version:  trident.netapp.io/v1
    Fields Type:  FieldsV1
    fieldsV1:
      f:spec:
        .:
        f:debug:
        f:namespace:
    Manager:      kubectl-create
    Operation:    Update
    Time:         2021-05-07T17:00:28Z
    API Version:  trident.netapp.io/v1
```

```
    Fields Type:  FieldsV1
    fieldsV1:
      f:status:
        .:
        f:currentInstallationParams:
          .:
          f:IPv6:
          f:autosupportHostname:
          f:autosupportImage:
          f:autosupportProxy:
          f:autosupportSerialNumber:
          f:debug:
          f:enableNodePrep:
          f:imagePullSecrets:
          f:imageRegistry:
          f:k8sTimeout:
          f:kubeletDir:
          f:logFormat:
          f:silenceAutosupport:
          f:tridentImage:
        f:message:
        f:namespace:
        f:status:
        f:version:
    Manager:         trident-operator
    Operation:       Update
    Time:            2021-05-07T17:00:28Z
  Resource Version:  931421
  Self Link:
/apis/trident.netapp.io/v1/tridentorchestrators/trident
  UID:               8a26a7a6-dde8-4d55-9b66-a7126754d81f
Spec:
  Debug:      true
  Namespace:  trident
Status:
  Current Installation Params:
    IPv6:                       false
    Autosupport Hostname:
    Autosupport Image:          netapp/trident-autosupport:21.01
    Autosupport Proxy:
    Autosupport Serial Number:
    Debug:                      true
    Enable Node Prep:           false
    Image Pull Secrets:
    Image Registry:
    k8sTimeout:           30
```

```
    Kubelet Dir:          /var/lib/kubelet
    Log Format:           text
    Silence Autosupport:  false
    Trident Image:        netapp/trident:22.01.0
  Message:                Trident installed
  Namespace:              trident
  Status:                 Installed
  Version:                v22.01.0
Events:
  Type    Reason      Age   From                        Message
  ----    ------      ----  ----                        -------
  Normal  Installing  80s   trident-operator.netapp.io  Installing
Trident
  Normal  Installed   68s   trident-operator.netapp.io  Trident
installed
```

7. 您可以檢查在命名空間中執行的Pod、或使用tridentctl二進位檔檢查安裝的版本、以驗證Trident是否已成功安裝。

```
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-bb64c6cb4-lmd6h         6/6     Running   0          82s
trident-csi-gn59q                   2/2     Running   0          82s
trident-csi-m4szj                   2/2     Running   0          82s
trident-csi-sb9k9                   2/2     Running   0          82s
trident-operator-66f48895cc-lzczk   1/1     Running   0          2m39s

[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
+----------------+----------------+
| SERVER VERSION | CLIENT VERSION |
+----------------+----------------+
| 22.01.0          | 22.01.0          |
+----------------+----------------+
```

準備工作節點以供儲存

NFS

大多數Kubernetes發佈版本都隨附套件和公用程式、可在預設情況下安裝NFS後端、包括Red Hat OpenShift。

不過、對於NFSv3、用戶端與伺服器之間沒有協調並行的機制。因此、用戶端SUNRPC插槽表項目的最大數量必須以伺服器上支援的值手動同步、以確保NFS連線的最佳效能、而無需伺服器減少連線的視窗大小。

對於支援的SUNRPC插槽表項目數量上限為128、亦即、支援的每次可同時處理128個NFS要求。ONTAP ONTAP不過、根據預設、每個連線的Red Hat CoreOS/Red Hat Enterprise Linux最多可有65536個SUNRPC插槽表項目。我們需要將此值設為128、這可透過OpenShift中的機器組態操作員（MCO）來完成。

若要修改OpenShift工作節點中的最大社工PC插槽表格項目、請完成下列步驟：

1. 登入OCP網路主控台、然後瀏覽至「運算」>「機器組態」。按一下「Create Machine Config（建立機器組複製並貼上Yaml檔案、然後按一下「Create（建立）」。

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 98-worker-nfs-rpc-slot-tables
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
        - contents:
            source: data:text/plain;charset=utf-
8; base64,b3B0aW9ucyBzdW5ycGMgdGNwX21heF9zbG90X3RhYmxlX2VudHJpZXM9MTI4Cg=
=
          filesystem: root
          mode: 420
          path: /etc/modprobe.d/sunrpc.conf
```

2. 建立MCO之後、必須在所有工作節點上套用組態、然後逐一重新開機。整個程序約需20至30分鐘。使用「oce Get MCP」確認是否套用機器組態、並確認已更新員工的機器組態集區。

```
[netapp-user@rhel7 openshift-deploy]$ oc get mcp
NAME     CONFIG                                     UPDATED   UPDATING
DEGRADED
master   rendered-master-a520ae930e1d135e0dee7168   True      False
False
worker   rendered-worker-de321b36eeba62df41feb7bc   True      False
False
```

iSCSI

若要準備工作節點、以便透過iSCSI傳輸協定對應區塊儲存磁碟區、您必須安裝必要的套件、才能支援該功能。

在Red Hat OpenShift中、這是在叢集部署之後、將MCO（機器組態操作員）套用至叢集來處理。

若要設定工作節點以執行iSCSI服務、請完成下列步驟：

1. 登入OCP網路主控台、然後瀏覽至「運算」>「機器組態」。按一下「Create Machine Config（建立機器組複製並貼上Yaml檔案、然後按一下「Create（建立）」。

不使用多重路徑時：

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-worker-element-iscsi
spec:
  config:
    ignition:
      version: 3.2.0
    systemd:
      units:
        - name: iscsid.service
          enabled: true
          state: started
  osImageURL: ""
```

使用多重路徑時：

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 99-worker-ontap-iscsi
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
      - contents:
          source: data:text/plain;charset=utf-
8; base64,ZGVmYXVsdHMgewogICAgICAgIHVzZXJfZnJpZW5kbHlfbmFtZXMgbm8KICAgICA
gICBmaW5kX211bHRpcGF0aHMgbm8KfQoKYmxhY2tsaXN0X2V4Y2VwdGlvbnMgewogICAgICA
gIHByb3BlcnR5ICIoU0NTSV9JREVOVF98SURfV1dOKSIKfQoKYmxhY2tsaXN0IHsKfQoK
          verification: {}
        filesystem: root
        mode: 400
        path: /etc/multipath.conf
    systemd:
      units:
        - name: iscsid.service
          enabled: true
          state: started
        - name: multipathd.service
          enabled: true
          state: started
  osImageURL: ""
```

2. 建立組態之後、將組態套用至工作節點並重新載入大約需要20到30分鐘的時間。使用「oce Get MCP」確認是否套用機器組態、並確認已更新員工的機器組態集區。您也可以登入工作者節點、確認iscsid服務正在執行（如果使用多重路徑、則多路徑服務正在執行）。

```
[netapp-user@rhel7 openshift-deploy]$ oc get mcp
NAME     CONFIG                                     UPDATED   UPDATING
DEGRADED
master   rendered-master-a520ae930e1d135e0dee7168   True      False
False
worker   rendered-worker-de321b36eeba62df41feb7bc   True      False
False

[netapp-user@rhel7 openshift-deploy]$ ssh core@10.61.181.22 sudo
systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled;
vendor preset: disabled)
   Active: active (running) since Tue 2021-05-26 13:36:22 UTC; 3 min ago
     Docs: man:iscsid(8)
           man:iscsiadm(8)
 Main PID: 1242 (iscsid)
   Status: "Ready to process requests"
    Tasks: 1
   Memory: 4.9M
      CPU: 9ms
   CGroup: /system.slice/iscsid.service
           └─1242 /usr/sbin/iscsid -f

[netapp-user@rhel7 openshift-deploy]$ ssh core@10.61.181.22 sudo
systemctl status multipathd
 ● multipathd.service - Device-Mapper Multipath Device Controller
   Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled;
vendor preset: enabled)
   Active: active (running) since Tue 2021-05-26 13:36:22 UTC; 3 min ago
  Main PID: 918 (multipathd)
    Status: "up"
    Tasks: 7
    Memory: 13.7M
    CPU: 57ms
    CGroup: /system.slice/multipathd.service
            └─918 /sbin/multipathd -d -s
```

此外、您也可以使用適當的旗標來執行「occ偵錯」命令、確認機器組態已成功套用、服務已如預期般啟動。

建立儲存系統後端

完成Astra Trident操作員安裝之後、您必須為所使用的特定NetApp儲存平台設定後端。請依照下列連結繼續Astra Trident的設定與組態。

- "NetApp ONTAP 不適用於NFS"
- "NetApp ONTAP 支援iSCSI"
- "支援iSCSI NetApp Element"

NetApp ONTAP 不適用於NFS組態

若要與NetApp ONTAP 支援儲存系統進行Trident整合、您必須建立後端、以便與儲存系統進行通訊。

1. 下載的安裝歸檔文件中有「shame-INPUT」資料夾階層的範例後端檔案。對於ONTAP NetApp支援NFS的系統、請將「backender-ontap - nas.json」檔案複製到您的工作目錄、然後編輯檔案。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/ontap-nas/backend-ontap-nas.json ./
[netapp-user@rhel7 trident-installer]$ vi backend-ontap-nas.json
```

2. 編輯後端名稱、管理LIF、dataLIF、SVM、使用者名稱、 和密碼值。

```
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "ontap-nas+10.61.181.221",
    "managementLIF": "172.21.224.201",
    "dataLIF": "10.61.181.221",
    "svm": "trident_svm",
    "username": "cluster-admin",
    "password": "password"
}
```

> 最佳實務做法是將自訂的backendName值定義為storageDriverName和資料LIF的組合、以利NFS識別。

3. 在這個後端檔案就緒的情況下、執行下列命令來建立第一個後端。

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-nas.json
+------------------------+----------------
+--------------------------------------+--------+---------+
|          NAME          | STORAGE DRIVER |                  UUID
| STATE  | VOLUMES |
+------------------------+----------------
+--------------------------------------+--------+---------+
| ontap-nas+10.61.181.221 | ontap-nas      | be7a619d-c81d-445c-b80c-
5c87a73c5b1e | online |       0 |
+------------------------+----------------
+--------------------------------------+--------+---------+
```

4. 建立後端之後、您必須接著建立儲存類別。就像後端一樣、範例輸入資料夾中也有可供編輯的儲存類別檔案範例。將其複製到工作目錄、並進行必要的編輯、以反映所建立的後端。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

5. 唯一必須對此檔案進行的編輯、是從新建立的後端、將「backendType」值定義為儲存驅動程式名稱。另請注意名稱欄位值、此值必須在後續步驟中參考。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
```

> 在此檔案中定義了一個名為「FSType」的選用欄位。此行可在NFS後端刪除。

6. 執行「oc"命令以建立儲存類別。

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

7. 建立儲存類別之後、您必須建立第一個持續磁碟區宣告（PVc）。還有一個「PVC-base.yaml」檔案範例、也可在範例輸入中執行此動作。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-
basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

8. 唯一必須對此檔案進行的編輯、是確保「儲存類別名稱」欄位符合剛剛建立的欄位。您可以根據要配置的工作負載需求、進一步自訂PVc定義。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

9. 使用「oc"命令建立PVc。視所建立的備用磁碟區大小而定、建立作業可能需要一些時間、因此您可以在完成時觀看程序。

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME    STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
basic   Bound    pvc-b4370d37-0fa4-4c17-bd86-94f96c94b42d   1Gi
RWO            basic-csi      7s
```

NetApp ONTAP 支援iSCSI組態

若要與NetApp ONTAP 支援儲存系統進行Trident整合、您必須建立後端、以便與儲存系統進行通訊。

1. 下載的安裝歸檔文件中有「shame-INPUT」資料夾階層的範例後端檔案。對於ONTAP 供應iSCSI的NetApp支援系統、請將「backender-ontap - san・json」檔案複製到您的工作目錄、然後編輯該檔案。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/ontap-san/backend-ontap-san.json ./
[netapp-user@rhel7 trident-installer]$ vi backend-ontap-san.json
```

2. 編輯此檔案中的管理LIF、dataLIF、SVM、使用者名稱和密碼值。

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "172.21.224.201",
  "dataLIF": "10.61.181.240",
  "svm": "trident_svm",
  "username": "admin",
  "password": "password"
}
```

3. 在這個後端檔案就緒的情況下、執行下列命令來建立第一個後端。

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-san.json
+------------------------+----------------
+--------------------------------------+--------+---------+
|          NAME          | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+------------------------+----------------
+--------------------------------------+--------+---------+
| ontapsan_10.61.181.241 | ontap-san      | 6788533c-7fea-4a35-b797-
fb9bb3322b91 | online |       0 |
+------------------------+----------------
+--------------------------------------+--------+---------+
```

4. 建立後端之後、您必須接著建立儲存類別。就像後端一樣、範例輸入資料夾中也有可供編輯的儲存類別檔案範例。將其複製到工作目錄、並進行必要的編輯、以反映所建立的後端。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

5. 唯一必須對此檔案進行的編輯、是從新建立的後端、將「backendType」值定義為儲存驅動程式名稱。另請注意名稱欄位值、此值必須在後續步驟中參考。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
```

> 在此檔案中定義了一個名為「FSType」的選用欄位。在iSCSI後端中、此值可設定為特定的Linux檔案系統類型（XFS、ext4等）、也可刪除以允許OpenShift決定要使用的檔案系統。

6. 執行「oc"命令以建立儲存類別。

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

7. 建立儲存類別之後、您必須建立第一個持續磁碟區宣告（PVc）。還有一個「PVC-base.yaml」檔案範例、也可在範例輸入中執行此動作。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-
basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

8. 唯一必須對此檔案進行的編輯、是確保「儲存類別名稱」欄位符合剛剛建立的欄位。您可以根據要配置的工作負載需求、進一步自訂PVc定義。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

9. 使用「oc"命令建立PVc。視所建立的備用磁碟區大小而定、建立作業可能需要一些時間、因此您可以在完成時觀看程序。

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME    STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
basic   Bound    pvc-7ceac1ba-0189-43c7-8f98-094719f7956c   1Gi
RWO            basic-csi      3s
```

支援iSCSI組態NetApp Element

若要啟用Trident與NetApp Element 支援功能的整合、您必須建立後端、以便使用iSCSI傳輸協定與儲存系統進行通訊。

1. 下載的安裝歸檔文件中有「shame-INPUT」資料夾階層的範例後端檔案。若NetApp Element 為供應iSCSI的支援系統、請將「backend-solidfire.json」檔案複製到您的工作目錄、然後編輯檔案。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/solidfire/backend-solidfire.json ./
[netapp-user@rhel7 trident-installer]$ vi ./backend-solidfire.json
```

   a. 編輯「端點」行上的使用者、密碼和MVIP值。
   b. 編輯「VIP」值。

```
 {
    "version": 1,
    "storageDriverName": "solidfire-san",
    "Endpoint": "https://trident:password@172.21.224.150/json-
rpc/8.0",
    "SVIP": "10.61.180.200:3260",
    "TenantName": "trident",
    "Types": [{"Type": "Bronze", "Qos": {"minIOPS": 1000, "maxIOPS":
2000, "burstIOPS": 4000}},
              {"Type": "Silver", "Qos": {"minIOPS": 4000, "maxIOPS":
6000, "burstIOPS": 8000}},
              {"Type": "Gold", "Qos": {"minIOPS": 6000, "maxIOPS":
8000, "burstIOPS": 10000}}]
}
```

2. 在這個後端檔案就位的情況下、執行下列命令來建立您的第一個後端。

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-solidfire.json
+------------------------+----------------
+--------------------------------------+--------+---------+
|          NAME          | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+------------------------+----------------
+--------------------------------------+--------+---------+
| solidfire_10.61.180.200 | solidfire-san  | b90783ee-e0c9-49af-8d26-
3ea87ce2efdf | online |       0 |
+------------------------+----------------
+--------------------------------------+--------+---------+
```

3. 建立後端之後、您必須接著建立儲存類別。就像後端一樣、範例輸入資料夾中也有可供編輯的儲存類別檔案範例。將其複製到工作目錄、並進行必要的編輯、以反映所建立的後端。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

4. 唯一必須對此檔案進行的編輯、是從新建立的後端、將「backendType」值定義為儲存驅動程式名稱。另請注意名稱欄位值、此值必須在後續步驟中參考。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "solidfire-san"
```

> 在此檔案中定義了一個名為「FSType」的選用欄位。在iSCSI後端中、此值可設定為特定的Linux檔案系統類型（XFS、ext4等）、也可刪除此值、讓OpenShift決定要使用的檔案系統。

5. 執行「oc"命令以建立儲存類別。

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

6. 建立儲存類別之後、您必須建立第一個持續磁碟區宣告（PVc）。還有一個「PVC-base.yaml」檔案範例、也可在範例輸入中執行此動作。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-
basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

7. 唯一必須對此檔案進行的編輯、是確保「儲存類別名稱」欄位符合剛剛建立的欄位。您可以根據要配置的工作負載需求、進一步自訂PVc定義。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

8. 使用「oc"命令建立PVc。視所建立的備用磁碟區大小而定、建立作業可能需要一些時間、因此您可以在完成時觀看程序。

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME    STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
basic   Bound    pvc-3445b5cc-df24-453d-a1e6-b484e874349d   1Gi
RWO            basic-csi      5s
```

進階組態選項

探索負載平衡器選項：採用NetApp的Red Hat OpenShift

在大多數情況下、Red Hat OpenShift會透過路由、讓外部使用者能夠使用應用程式。提供可從外部存取的主機名稱、即可公開服務。OpenShift路由器可以使用定義的路由及其服務所識別的端點、以提供與外部用戶端的命名連線。

不過在某些情況下、應用程式需要部署和設定自訂的負載平衡器、才能提供適當的服務。其中一個例子是NetApp Astra Control Center。為了滿足這項需求、我們評估了許多自訂負載平衡器選項。本節將說明其安裝與組態。

以下頁面提供有關Red Hat OpenShift with NetApp解決方案中驗證的負載平衡器選項的其他資訊：

- "MetalLB."
- "F5 BIG-IP"

安裝**MetalLB**負載平衡器：**Red Hat OpenShift with NetApp**

本頁列出MetalLB負載平衡器的安裝與組態指示。

MetalLB是安裝在OpenShift叢集上的自我代管網路負載平衡器、可在未在雲端供應商上執行的叢集中、建立類型負載平衡器的OpenShift服務。MetalLB的兩項主要功能是位址分配和外部宣告、這些功能可搭配運作以支援負載平衡器服務。

MetalLB組態選項

根據MetalLB如何宣告指派給OpenShift叢集外部負載平衡器服務的IP位址、它以兩種模式運作：

- *第2層模式。*在此模式下、OpenShift叢集中的一個節點會取得服務的所有權、並回應該IP的ARP要求、以便在OpenShift叢集外部存取。因為只有節點會通告IP、所以它會有頻寬瓶頸和緩慢的容錯移轉限制。如需詳細資訊、請參閱文件 "請按這裡"。
- * BGP模式。*在此模式下、OpenShift叢集中的所有節點都會與路由器建立BGP對等工作階段、並通告路由以將流量轉送到服務IP。這項作業的先決條件是將MetalLB與該網路中的路由器整合。由於BGP中的雜湊機制、因此在變更服務的IP對節點對應時、會有一定的限制。如需詳細資訊、請參閱文件 "請按這裡"。

> 針對本文件、我們將在第2層模式中設定MetalLB。

安裝**MetalLB**負載平衡器

1. 下載MetalLB資源。

```
[netapp-user@rhel7 ~]$ wget
https://raw.githubusercontent.com/metallb/metallb/v0.10.2/manifests/name
space.yaml
[netapp-user@rhel7 ~]$ wget
https://raw.githubusercontent.com/metallb/metallb/v0.10.2/manifests/meta
llb.yaml
```

2. 編輯檔案「metallb.yaml」、並從「控制器部署」和「示範演講者」中移除「pec.template.spec.securityContext`」。

   要刪除的行數：

```
securityContext:
  runAsNonRoot: true
  runAsUser: 65534
```

3. 建立「metallb-system」命名空間。

```
[netapp-user@rhel7 ~]$ oc create -f namespace.yaml
namespace/metallb-system created
```

4. 建立MetalLB CR。

```
[netapp-user@rhel7 ~]$ oc create -f metallb.yaml
podsecuritypolicy.policy/controller created
podsecuritypolicy.policy/speaker created
serviceaccount/controller created
serviceaccount/speaker created
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
role.rbac.authorization.k8s.io/controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller
created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker
created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
rolebinding.rbac.authorization.k8s.io/controller created
daemonset.apps/speaker created
deployment.apps/controller created
```

5. 在設定MetalLB揚聲器之前、請先授予揚聲器示範設定提高權限、以便執行所需的網路組態、使負載平衡器正常運作。

```
[netapp-user@rhel7 ~]$ oc adm policy add-scc-to-user privileged -n
metallb-system -z speaker
clusterrole.rbac.authorization.k8s.io/system:openshift:scc:privileged
added: "speaker"
```

6. 在「metallb-system」命名空間中建立「ConfigMap」來設定MetalLB。

```
[netapp-user@rhel7 ~]$ vim metallb-config.yaml

apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
  name: config
data:
  config: |
    address-pools:
    - name: default
      protocol: layer2
      addresses:
      - 10.63.17.10-10.63.17.200

[netapp-user@rhel7 ~]$ oc create -f metallb-config.yaml
configmap/config created
```

7. 現在、當建立負載平衡器服務時、MetalLB會指派外部IP給服務、並回應ARP要求來通告IP位址。

> 如果您想要在BGP模式中設定MetalLB、請跳過上述步驟6、然後依照MetalLB文件中的程序進行 "請按這裡"。

安裝**F5 BIG-IP**負載平衡器

F5 BIG-IP是應用程式交付控制器（ADC）、提供一系列進階的正式作業級流量管理與安全服務、例如L4-L7負載平衡、SSL/TLS卸載、DNS、防火牆等。這些服務可大幅提升應用程式的可用度、安全性和效能。

您可以在專屬硬體、雲端或內部部署的虛擬應用裝置上、以各種方式部署和使用F5 BIG-IP。請參閱此處的文件、依照需求探索及部署F5 BIG-IP。

為有效整合使用Red Hat OpenShift的F5 BIG-IP服務、F5提供Big IP Container Ingress Service（CI）。CI是以控制器Pod的形式安裝、可針對特定的自訂資源定義（CRD）來觀看OpenShift API、並管理F5 BIG-IP系統組態。您可以在OpenShift中設定F5 BIG-IP CI、以控制服務類型負載平衡器和路由。

此外、若要自動分配IP位址以服務負載平衡器類型、您可以使用F5 IPAM控制器。將F5 IPAM控制器安裝為控制器Pod、會使用ipamLabel附註來監視負載平衡器服務的OpenShift API、以便從預先設定的集區分配IP位址。

本頁列出適用於F5 BIG-IP CI和IPAM控制器的安裝與組態指示。您必須部署並授權使用F5 BIG-IP系統、才能做為先決條件。也必須授權使用SDN服務、此服務預設隨附於Big IP VE基礎授權中。

> 可以在獨立或叢集模式中部署F5 BIG-IP。為了進行此驗證、在獨立模式下部署了F5 BIG-IP、但為了正式作業目的、最好使用一個BIG-IP叢集、以避免單點故障。

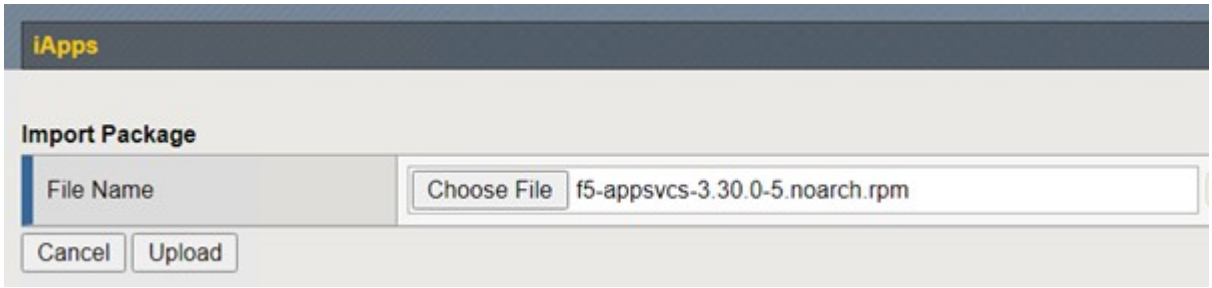
> 您可以在專屬硬體、雲端或內部部署的虛擬應用裝置上部署一個F5 BIG-IP系統、其版本超過12.x、以便與F5 CI整合。就本文件而言、以虛擬應用裝置（例如使用BIG-IP VE版本）的形式驗證的F5 BIG-IP系統。

已驗證的版本

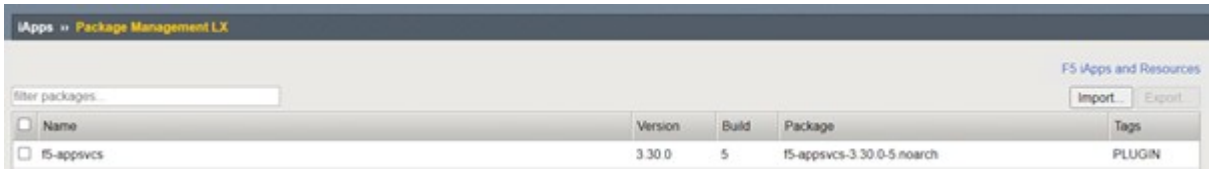
技術	軟體版本
Red Hat OpenShift	4.6 EUS、4.7
F5 BIG-IP VE版本	16.1.0
F5 Container Ingress服務	2.5.1
F5 IPAM控制器	0.1.4
F5 AS3	3.30.0

安裝

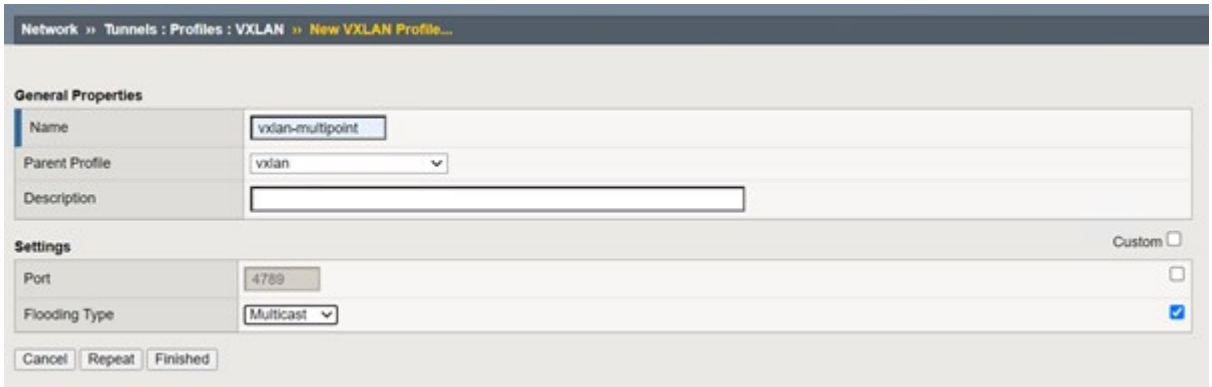
1. 安裝F5 Application Services 3擴充功能、讓BIG-IP系統接受Json中的組態、而非命令命令。前往 "F5 AS3 GitHub儲存庫"下載最新的RPM檔案。
2. 登入F5 BIG-IP系統、瀏覽至iApps >「套件管理Lx」、然後按一下「匯入」。
3. 按一下"選擇檔案"並選取下載的AS3 RPM檔案、按一下"確定"、然後按一下"上傳"。



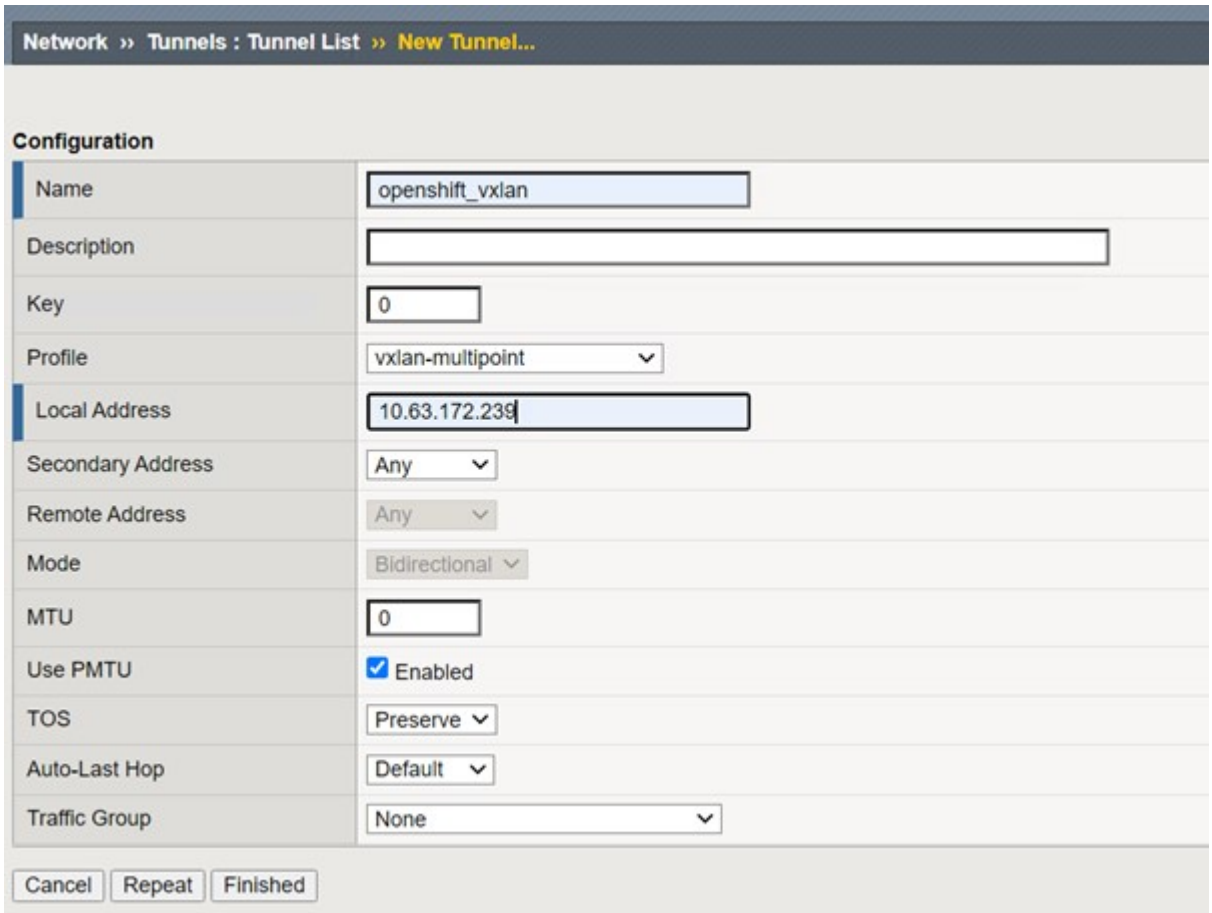
4. 確認已成功安裝AS3擴充功能。



5. 接下來、設定OpenShift與BIG-IP系統之間通訊所需的資源。首先在OpenShift和Big IP伺服器之間建立通道、方法是在適用於OpenShift SDN的Big IP系統上建立VXLAN通道介面。瀏覽至「Network（網路）」>「Tunnels（通道）」>「Profiles（設定檔）」、按一下「Create（建立）」、然後將「Parent Profile（父設定檔）」設定為VXLAN、「輸入設定檔的名稱、然後按一下「完成」。



6. 瀏覽至「網路」>「通道」>「通道清單」、按一下「建立」、然後輸入通道的名稱和本機IP位址。選取在上一個步驟中建立的通道設定檔、然後按一下「完成」。



7. 以叢集管理權限登入Red Hat OpenShift叢集。
8. 在OpenShift上為F5 BIG-IP伺服器建立主機子網路、將子網路從OpenShift叢集延伸至F5 BIG-IP伺服器。下載主機子網路Yaml定義。

```
wget https://github.com/F5Networks/k8s-bigip-
ctlr/blob/master/docs/config_examples/openshift/f5-kctlr-openshift-
hostsubnet.yaml
```

9. 編輯主機子網路檔案、並為OpenShift SDN新增BIG-IP VTEP（VXLAN通道）IP。

```
apiVersion: v1
kind: HostSubnet
metadata:
  name: f5-server
  annotations:
    pod.network.openshift.io/fixed-vnid-host: "0"
    pod.network.openshift.io/assign-subnet: "true"
# provide a name for the node that will serve as BIG-IP's entry into the
cluster
host: f5-server
# The hostIP address will be the BIG-IP interface address routable to
the
# OpenShift Origin nodes.
# This address is the BIG-IP VTEP in the SDN's VXLAN.
hostIP: 10.63.172.239
```

變更適用於您環境的主機IP和其他詳細資料。

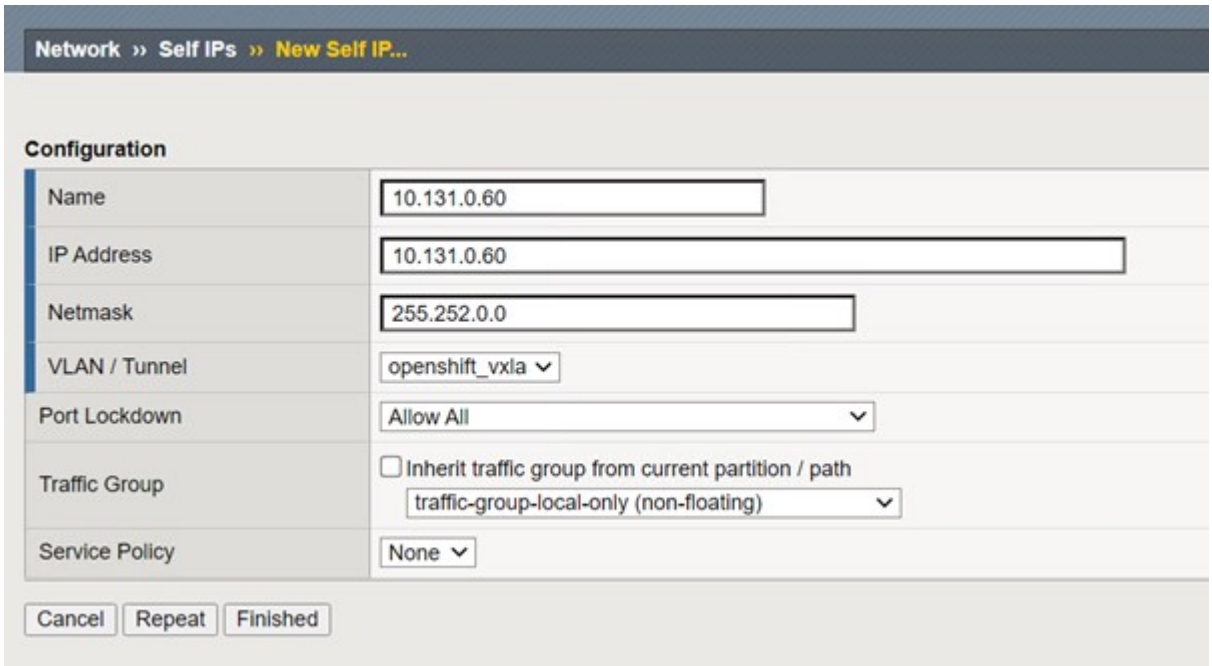
10. 建立主機子網路資源。

```
[admin@rhel-7 ~]$ oc create -f f5-kctlr-openshift-hostsubnet.yaml

hostsubnet.network.openshift.io/f5-server created
```

11. 取得為F5 BIG-IP伺服器所建立之主機子網路的叢集IP子網路範圍。

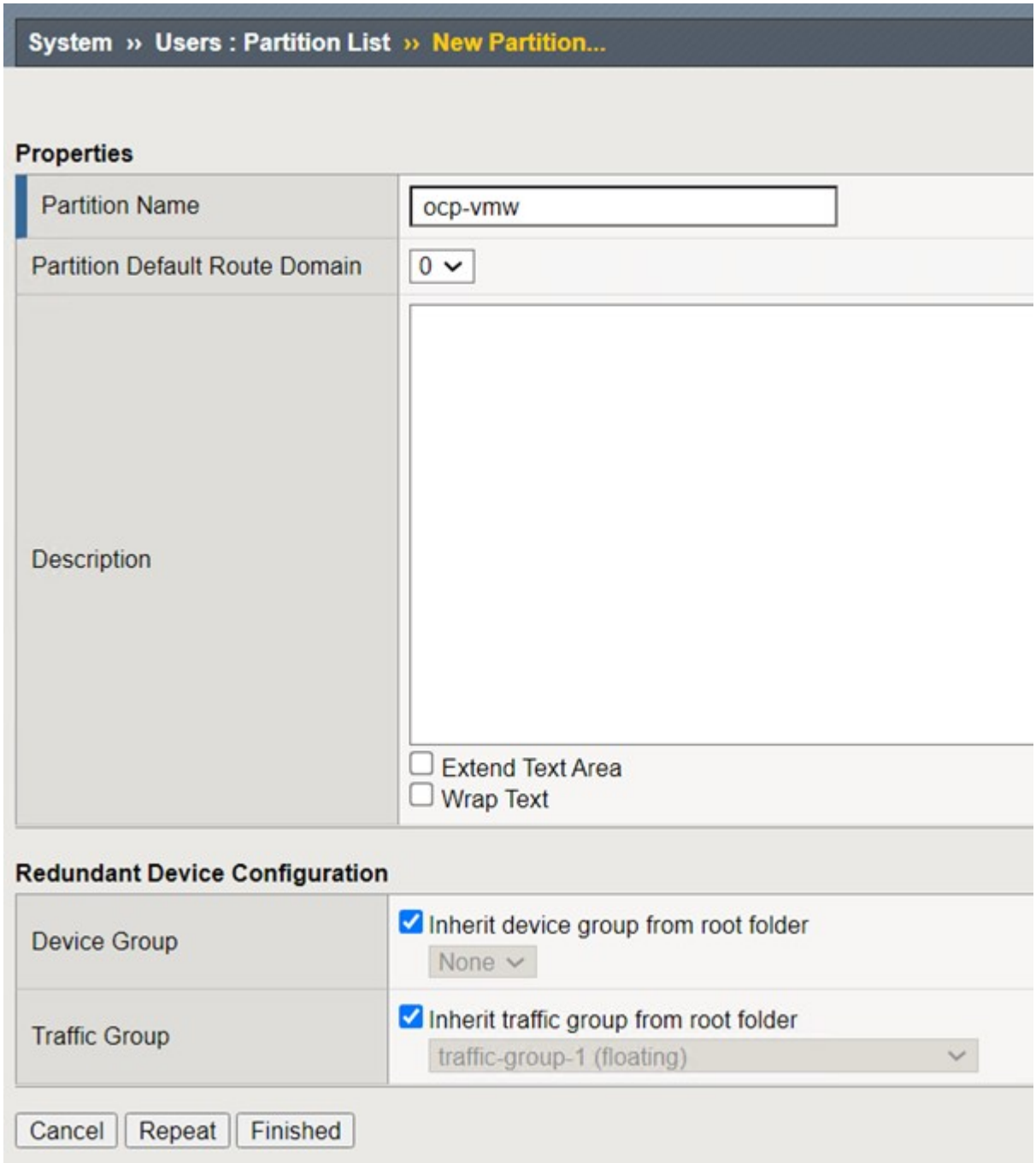
```
[admin@rhel-7 ~]$ oc get hostsubnet

NAME                          HOST                          HOST IP
SUBNET          EGRESS CIDRS   EGRESS IPS
f5-server                     f5-server                     10.63.172.239
10.131.0.0/23
ocp-vmw-nszws-master-0        ocp-vmw-nszws-master-0        10.63.172.44
10.128.0.0/23
ocp-vmw-nszws-master-1        ocp-vmw-nszws-master-1        10.63.172.47
10.130.0.0/23
ocp-vmw-nszws-master-2        ocp-vmw-nszws-master-2        10.63.172.48
10.129.0.0/23
ocp-vmw-nszws-worker-r8fh4    ocp-vmw-nszws-worker-r8fh4    10.63.172.7
10.130.2.0/23
ocp-vmw-nszws-worker-tvr46    ocp-vmw-nszws-worker-tvr46    10.63.172.11
10.129.2.0/23
ocp-vmw-nszws-worker-wdxhg    ocp-vmw-nszws-worker-wdxhg    10.63.172.24
10.128.2.0/23
ocp-vmw-nszws-worker-wg8r4    ocp-vmw-nszws-worker-wg8r4    10.63.172.15
10.131.2.0/23
ocp-vmw-nszws-worker-wtgfw    ocp-vmw-nszws-worker-wtgfw    10.63.172.17
10.128.4.0/23
```

12. 在OpenShift VXLAN上建立一個自有IP、並在OpenShift的主機子網路範圍中建立對應於F5 BIG-IP伺服器的IP。登入F5 BIG-IP系統、瀏覽至「網路」>「自助IP」、然後按一下「建立」。從為F5 BIG-IP主機子網路建立的叢集IP子網路輸入IP、選取VXLAN通道、然後輸入其他詳細資料。然後按一下「完成」。



13. 在要設定並搭配CI使用的F5 BIG-IP系統中建立分割區。瀏覽至「系統」>「使用者」>「分割清單」、按一

下「建立」、然後輸入詳細資料。然後按一下「完成」。



F5建議您不要在由CI管理的分割區上進行手動設定。

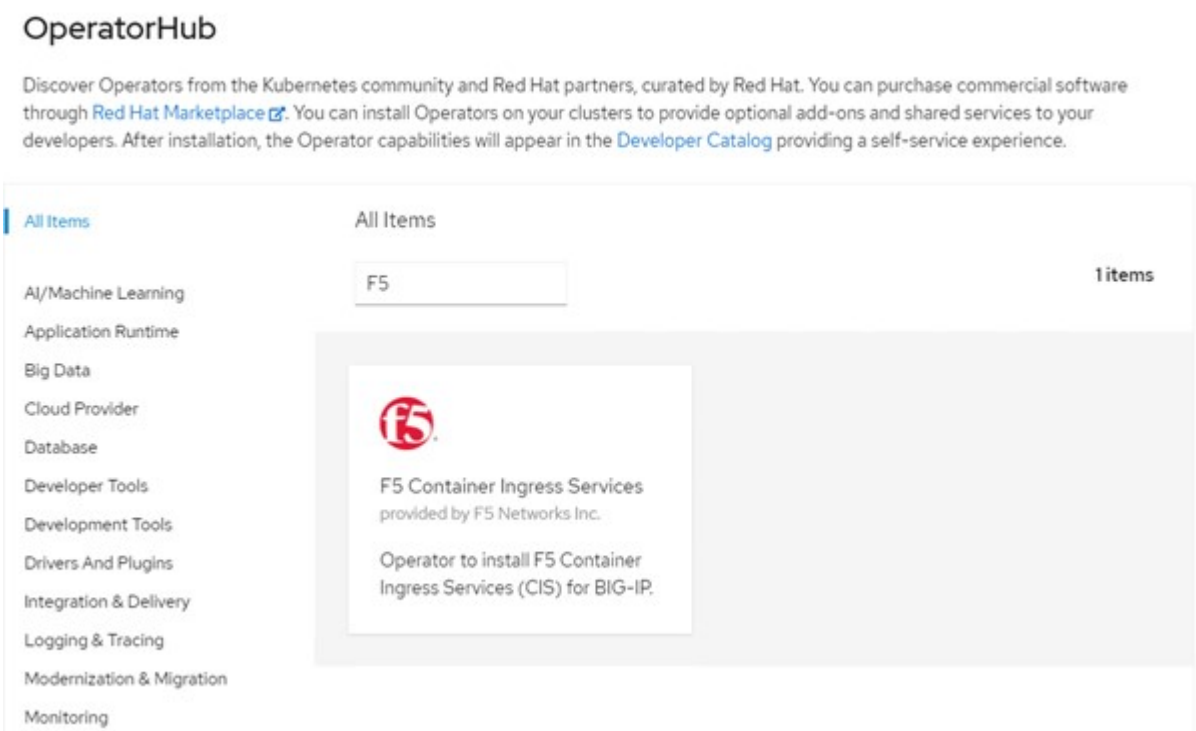
14. 使用來自作業系統集線器的操作員來安裝F5 BIG-IP CI。以叢集管理權限登入Red Hat OpenShift叢集、並使用F5 BIG-IP系統登入認證建立密碼、這是操作員的必要條件。

```
[admin@rhel-7 ~]$ oc create secret generic bigip-login -n kube-system
--from-literal=username=admin --from-literal=password=admin

secret/bigip-login created
```

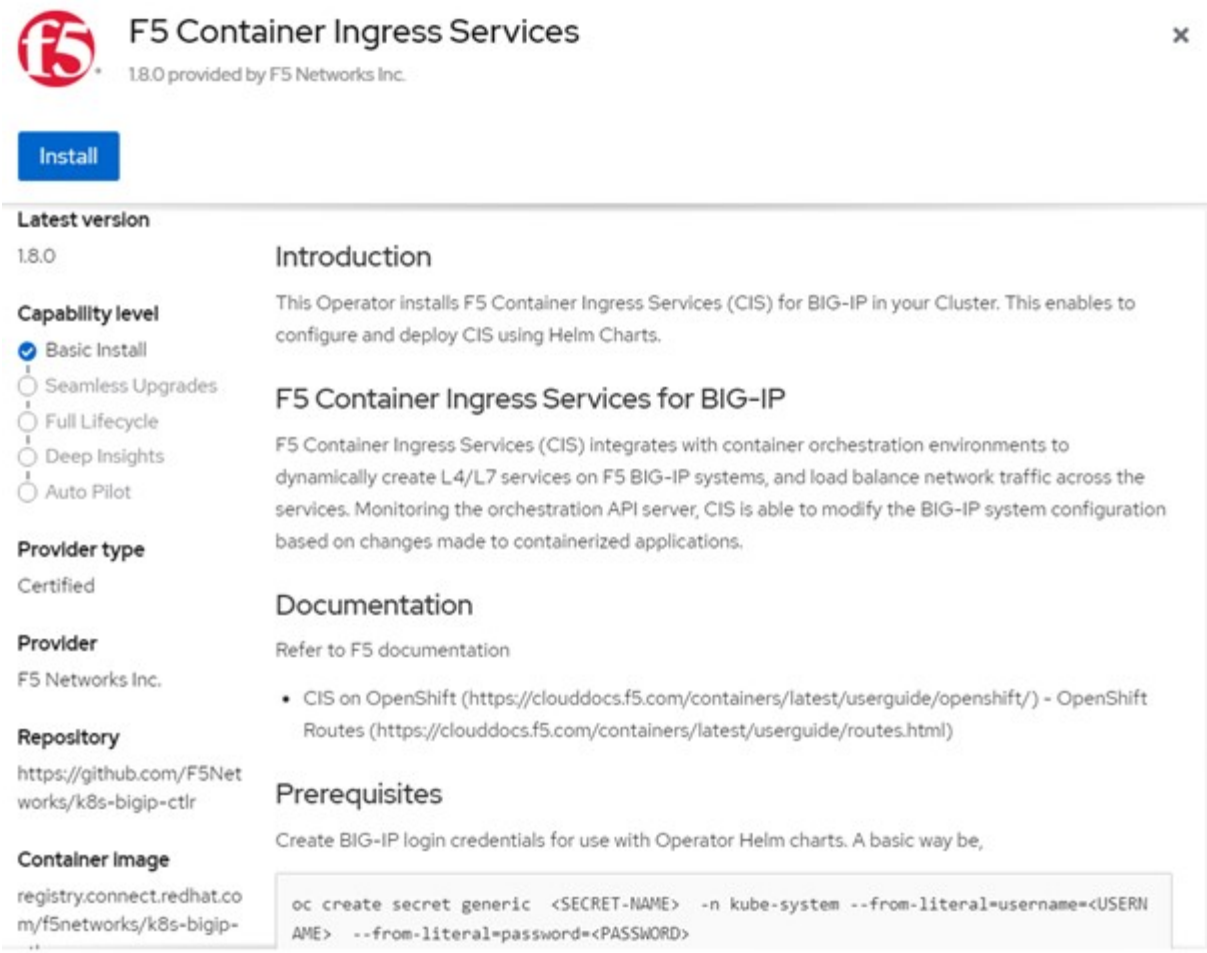
15. 安裝5個CI客戶需求日。

```
[admin@rhel-7 ~]$ oc apply -f
https://raw.githubusercontent.com/F5Networks/k8s-bigip-
ctlr/master/docs/config_examples/crd/Install/customresourcedefinitions.y
ml

customresourcedefinition.apiextensions.k8s.io/virtualservers.cis.f5.com
created
customresourcedefinition.apiextensions.k8s.io/tlsprofiles.cis.f5.com
created
customresourcedefinition.apiextensions.k8s.io/transportservers.cis.f5.co
m created
customresourcedefinition.apiextensions.k8s.io/externaldnss.cis.f5.com
created
customresourcedefinition.apiextensions.k8s.io/ingresslinks.cis.f5.com
created
```

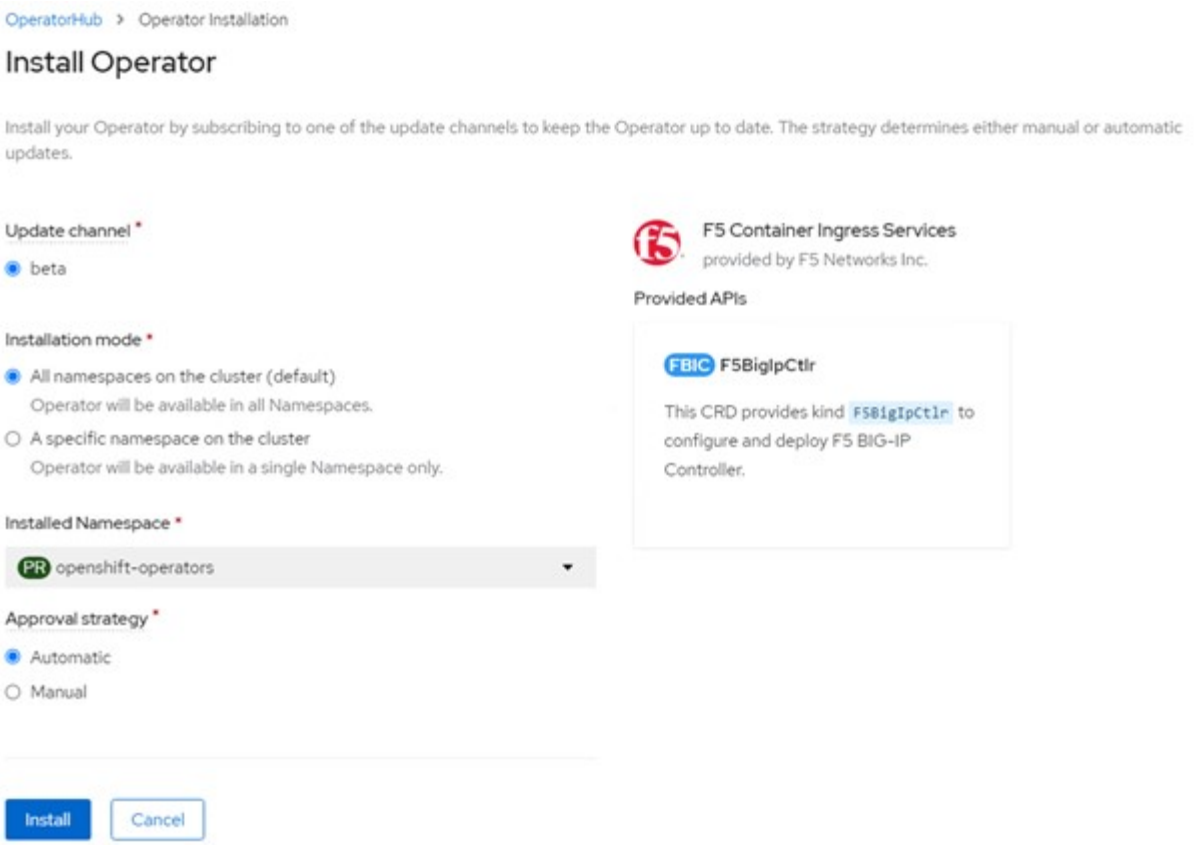
16. 瀏覽至「運算子」>「作業系統集線器」、搜尋關鍵字F5、然後按一下「F5 Container Ingress Service」方塊。



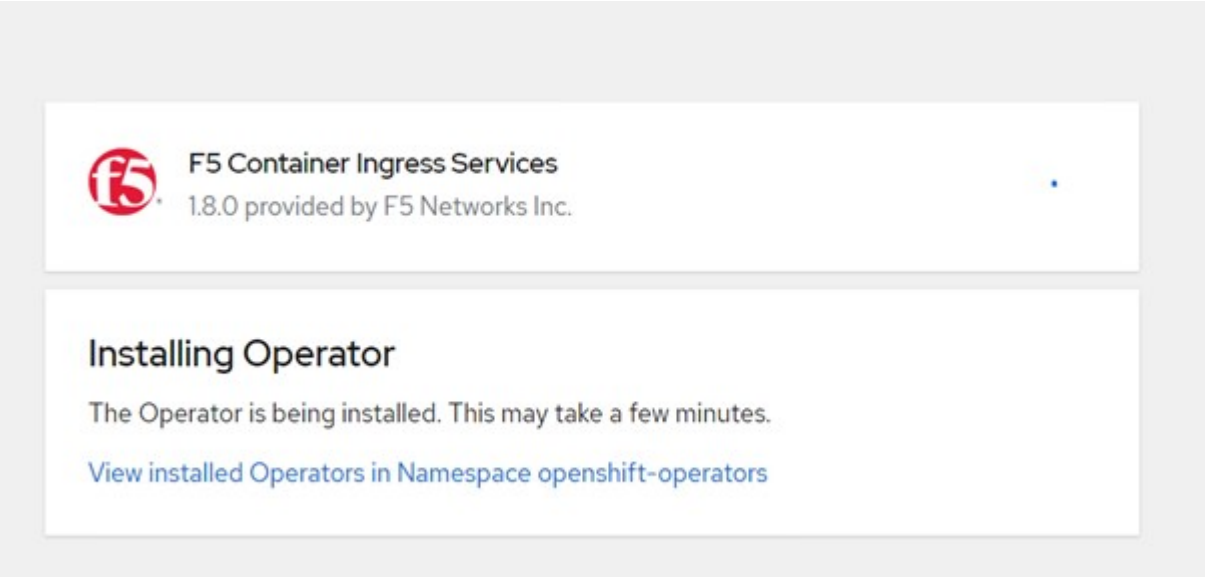
17. 閱讀操作員資訊、然後按一下「Install（安裝）」。



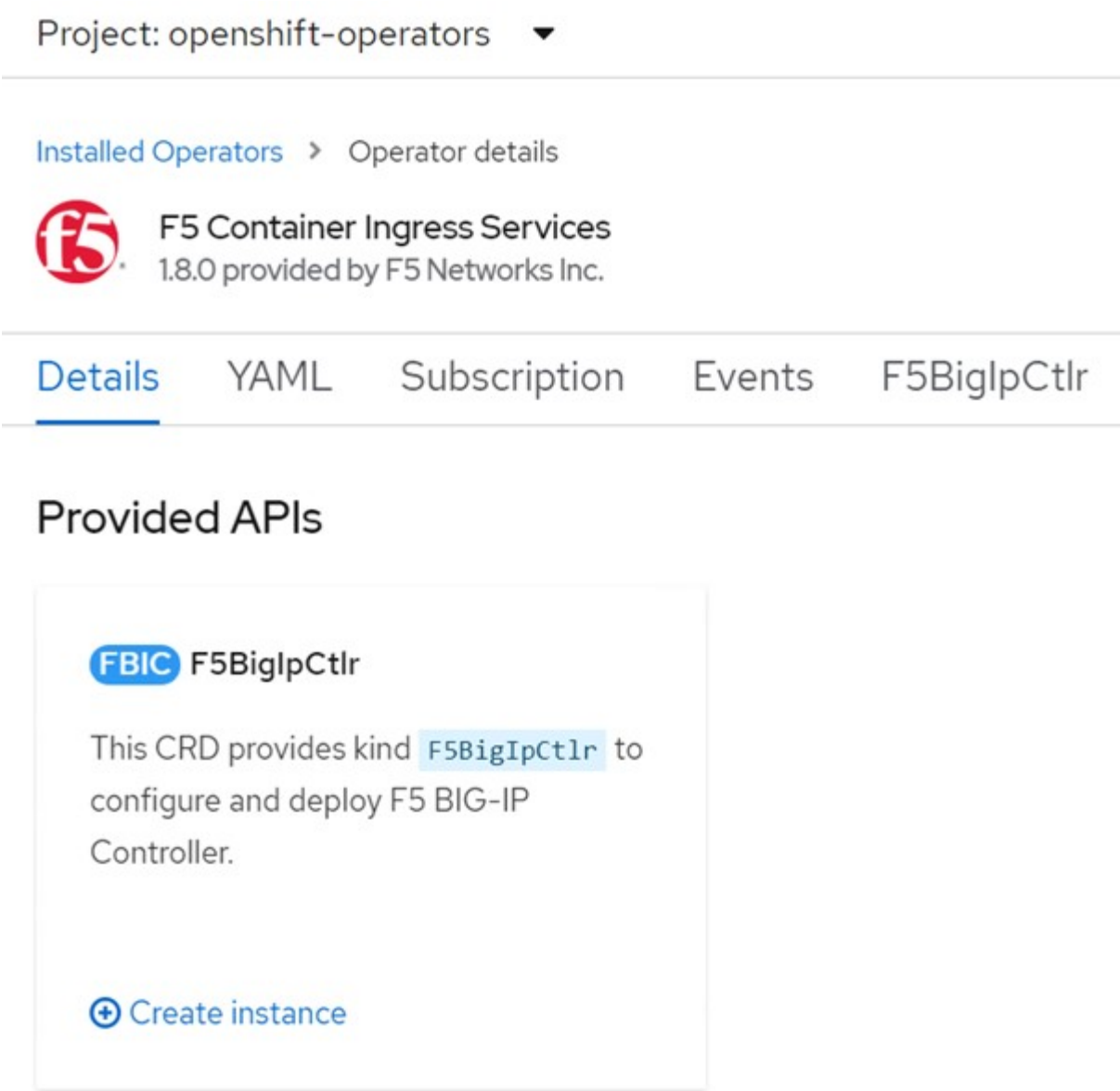
18. 在「Install（安裝）」操作員畫面上、保留所有預設參數、然後按一下「Install（安裝）」。



19. 安裝操作員需要一段時間。



20. 安裝操作員之後、會顯示安裝成功訊息。
21. 瀏覽至「運算子」>「安裝的運算子」、按一下「F5 Container Ingress Service」、然後按一下「F5Biglprcr」方塊下方的「Create Instance（建立執行個體）」。

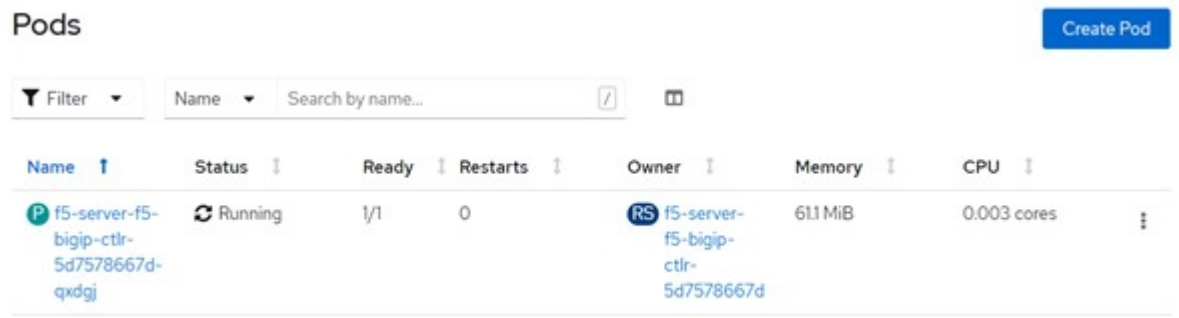


22. 按一下「Yaml View（Yaml檢視）」、然後在更新必要的參數後貼上下列內容。

> 請更新下列參數「bigip_partition'、「openshift_SDN_name'、「bigip_URL」和「bigip_login_secret」、以反映設定值、然後再複製內容。

```
apiVersion: cis.f5.com/v1
kind: F5BigIpCtlr
metadata:
  name: f5-server
  namespace: openshift-operators
spec:
  args:
    log_as3_response: true
    agent: as3
    log_level: DEBUG
    bigip_partition: ocp-vmw
    openshift_sdn_name: /Common/openshift_vxlan
    bigip_url: 10.61.181.19
    insecure: true
    pool-member-type: cluster
    custom_resource_mode: true
    as3_validation: true
    ipam: true
    manage_configmaps: true
  bigip_login_secret: bigip-login
  image:
    pullPolicy: Always
    repo: f5networks/cntr-ingress-svcs
    user: registry.connect.redhat.com
  namespace: kube-system
  rbac:
    create: true
  resources: {}
  serviceAccount:
    create: true
  version: latest
```

23. 貼上此內容之後、按一下「建立」。這會在K資料庫 系統命名空間中安裝CI Pod。



Red Hat OpenShift依預設提供一種方法、可透過L7負載平衡的路由來公開服務。內建的OpenShift路由器負責廣告和處理這些路由的流量。不過、您也可以設定F5 CI來支援透過外部的F5 BIG-IP系統的路由、以便作為輔助路由器執行、或取代自行代管的OpenShift路由器。CI會在Big IP系統中建立虛擬伺服器、做為OpenShift路由的路由器、而Big IP則負責通告和流量路由。如需啟用此功能的參數資訊、請參閱此處的文件。請注意、這些參數是針對APS/v1 API中的OpenShift部署資源所定義。因此、將這些項目搭配F5BigIprvtrr資源cis.f5.com/v1 API使用時、請將參數名稱的連字號（-）取代為底線（_）。

24. 傳遞給CI資源建立的引數包括「ipam: true」和「custom_resource_mode：true」。這些參數是啟用與IPAM控制器的CI整合所需的參數。建立F5 IPAM資源、確認CI已啟用IPAM整合。

```
[admin@rhel-7 ~]$ oc get f5ipam -n kube-system

NAMESPACE   NAME                                AGE
kube-system   ipam.10.61.181.19.ocp-vmw      43s
```

25. 建立F5 IPAM控制器所需的服務帳戶、角色和角色繫結。建立Yaml檔案並貼上下列內容。

```
[admin@rhel-7 ~]$ vi f5-ipam-rbac.yaml

kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: ipam-ctlr-clusterrole
rules:
  - apiGroups: ["fic.f5.com"]
    resources: ["ipams","ipams/status"]
    verbs: ["get", "list", "watch", "update", "patch"]
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: ipam-ctlr-clusterrole-binding
  namespace: kube-system
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: ipam-ctlr-clusterrole
subjects:
  - apiGroup: ""
    kind: ServiceAccount
    name: ipam-ctlr
    namespace: kube-system
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: ipam-ctlr
  namespace: kube-system
```

26. 建立資源。

```
[admin@rhel-7 ~]$ oc create -f f5-ipam-rbac.yaml

clusterrole.rbac.authorization.k8s.io/ipam-ctlr-clusterrole created
clusterrolebinding.rbac.authorization.k8s.io/ipam-ctlr-clusterrole-
binding created
serviceaccount/ipam-ctlr created
```

27. 建立Yaml檔案、然後貼上以下提供的F5 IPAM部署定義。

請更新下方spec.template.spec.contains[0].args中的IP範圍參數、以反映與您設定相對應的ipamLabel和IP位址範圍。

IPAM控制器的負載平衡器類型服務需要註釋ipamLabels ['range1'和'range2'、才能從定義的範圍偵測和指派IP位址。

```
[admin@rhel-7 ~]$ vi f5-ipam-deployment.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    name: f5-ipam-controller
  name: f5-ipam-controller
  namespace: kube-system
spec:
  replicas: 1
  selector:
    matchLabels:
      app: f5-ipam-controller
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: f5-ipam-controller
    spec:
      containers:
      - args:
        - --orchestration=openshift
        - --ip-range='{"range1":"10.63.172.242-10.63.172.249",
"range2":"10.63.170.111-10.63.170.129"}'
        - --log-level=DEBUG
        command:
        - /app/bin/f5-ipam-controller
        image: registry.connect.redhat.com/f5networks/f5-ipam-
controller:latest
        imagePullPolicy: IfNotPresent
        name: f5-ipam-controller
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      serviceAccount: ipam-ctlr
      serviceAccountName: ipam-ctlr
```

28. 建立F5 IPAM控制器部署。

```
[admin@rhel-7 ~]$ oc create -f f5-ipam-deployment.yaml

deployment/f5-ipam-controller created
```

29. 確認F5 IPAM控制器Pod正在執行。

```
[admin@rhel-7 ~]$ oc get pods -n kube-system

NAME                                        READY   STATUS    RESTARTS
AGE
f5-ipam-controller-5986cff5bd-2bvn6         1/1     Running   0
30s
f5-server-f5-bigip-ctlr-5d7578667d-qxdgj    1/1     Running   0
14m
```

30. 建立F5 IPAM架構。

```
[admin@rhel-7 ~]$ oc create -f
https://raw.githubusercontent.com/F5Networks/f5-ipam-
controller/main/docs/_static/schemas/ipam_schema.yaml

customresourcedefinition.apiextensions.k8s.io/ipams.fic.f5.com
```

驗證

1. 建立負載平衡器類型的服務

```
[admin@rhel-7 ~]$ vi example_svc.yaml

apiVersion: v1
kind: Service
metadata:
  annotations:
    cis.f5.com/ipamLabel: range1
  labels:
    app: f5-demo-test
  name: f5-demo-test
  namespace: default
spec:
  ports:
  - name: f5-demo-test
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: f5-demo-test
  sessionAffinity: None
  type: LoadBalancer
```

```
[admin@rhel-7 ~]$ oc create -f example_svc.yaml

service/f5-demo-test created
```

2. 檢查IPAM控制器是否指派外部IP給它。

```
[admin@rhel-7 ~]$ oc get svc

NAME           TYPE           CLUSTER-IP       EXTERNAL-IP
PORT(S)        AGE
f5-demo-test   LoadBalancer   172.30.210.108   10.63.172.242
80:32605/TCP   27s
```

3. 建立部署並使用所建立的負載平衡器服務。

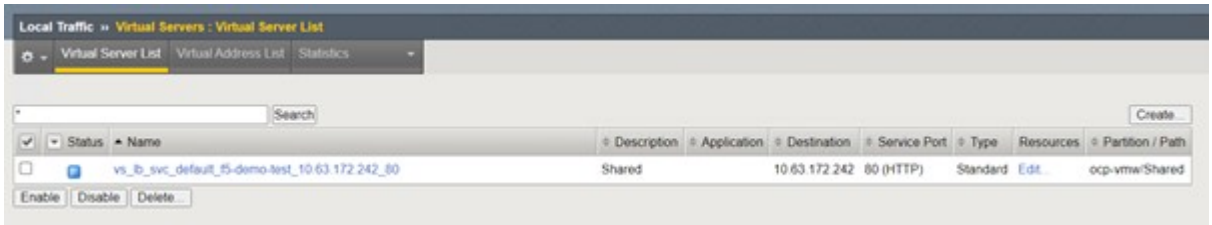
```
[admin@rhel-7 ~]$ vi example_deployment.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: f5-demo-test
  name: f5-demo-test
spec:
  replicas: 2
  selector:
    matchLabels:
      app: f5-demo-test
  template:
    metadata:
      labels:
        app: f5-demo-test
    spec:
      containers:
      - env:
        - name: service_name
          value: f5-demo-test
        image: nginx
        imagePullPolicy: Always
        name: f5-demo-test
        ports:
        - containerPort: 80
          protocol: TCP
```

```
[admin@rhel-7 ~]$ oc create -f example_deployment.yaml

deployment/f5-demo-test created
```

4. 檢查Pod是否正在執行。

```
[admin@rhel-7 ~]$ oc get pods

NAME                              READY   STATUS    RESTARTS   AGE
f5-demo-test-57c46f6f98-47wwp     1/1     Running   0          27s
f5-demo-test-57c46f6f98-cl2m8     1/1     Running   0          27s
```

5. 檢查OpenShift中是否針對負載平衡器類型的服務、在Big IP系統中建立對應的虛擬伺服器。瀏覽至本機流量>虛擬伺服器>虛擬伺服器清單。



建立私有映像登錄

對於大部分的Red Hat OpenShift部署、請使用類似的公用登錄 "Quay.IO" 或 "DockerHub" 滿足大多數客戶的需求。不過有時候客戶可能想要裝載自己的私有或自訂映像。

本程序說明如何建立私有映像登錄、並以Astra Trident和NetApp ONTAP 支援所提供的持續磁碟區作為後盾。

> Astra Control Center需要登錄來裝載Astra容器所需的映像。下節說明在Red Hat OpenShift叢集上設定私有登錄的步驟、以及推送支援Astra Control Center安裝所需的映像。

建立私有映像登錄

1. 移除目前預設儲存類別的預設註釋、並在OpenShift叢集的Trident備份儲存類別中註記為預設值。

```
[netapp-user@rhel7 ~]$ oc patch storageclass thin -p '{"metadata":
{"annotations": {"storageclass.kubernetes.io/is-default-class":
"false"}}}'
storageclass.storage.k8s.io/thin patched

[netapp-user@rhel7 ~]$ oc patch storageclass ocp-trident -p
'{"metadata": {"annotations": {"storageclass.kubernetes.io/is-default-
class": "true"}}}'
storageclass.storage.k8s.io/ocp-trident patched
```

2. 在「s pec」區段中輸入下列儲存參數、以編輯影像登錄操作員。

```
[netapp-user@rhel7 ~]$ oc edit
configs.imageregistry.operator.openshift.io

storage:
  pvc:
    claim:
```

3. 在「最新」區段中輸入下列參數、以建立具有自訂主機名稱的OpenShift路由。儲存並結束。

```
routes:
  - hostname: astra-registry.apps.ocp-vmw.cie.netapp.com
    name: netapp-astra-route
```

> 當您想要為路由建立自訂主機名稱時、會使用上述路由組態。如果希望OpenShift使用預設主機名稱來建立路由、您可以將下列參數新增至「預設路由：true」區段。

自訂TLS憑證

當您使用路由的自訂主機名稱時、預設會使用OpenShift Ingress運算子的預設TLS組態。不過、您可以將自訂TLS組態新增至路由。若要這麼做、請完成下列步驟。

a. 使用路由的TLS憑證和金鑰建立秘密。

```
[netapp-user@rhel7 ~]$ oc create secret tls astra-route-tls -n
openshift-image-registry –cert/home/admin/netapp-astra/tls.crt
--key=/home/admin/netapp-astra/tls.key
```

b. 編輯影像註冊運算子、並將下列參數新增至「s pec」區段。

```
[netapp-user@rhel7 ~]$ oc edit
configs.imageregistry.operator.openshift.io

routes:
  - hostname: astra-registry.apps.ocp-vmw.cie.netapp.com
    name: netapp-astra-route
    secretName: astra-route-tls
```

4. 再次編輯影像註冊業者、並將營運者的管理狀態變更為「老舊」狀態。儲存並結束。

```
oc edit configs.imageregistry/cluster

managementState: Managed
```

5. 如果滿足所有先決條件、就會為私有映像登錄建立PVCS、Pod和服務。幾分鐘後、登錄就會啟動。

```
[netapp-user@rhel7 ~]$oc get all -n openshift-image-registry

NAME                                                    READY   STATUS
RESTARTS   AGE
```

```
pod/cluster-image-registry-operator-74f6d954b6-rb7zr   1/1     Running
3          90d
pod/image-pruner-1627257600-f5cpj                       0/1     Completed
0          2d9h
pod/image-pruner-1627344000-swqx9                       0/1     Completed
0          33h
pod/image-pruner-1627430400-rv5nt                       0/1     Completed
0          9h
pod/image-registry-6758b547f-6pnj8                      1/1     Running
0          76m
pod/node-ca-bwb5r                                       1/1     Running
0          90d
pod/node-ca-f8w54                                       1/1     Running
0          90d
pod/node-ca-gjx7h                                       1/1     Running
0          90d
pod/node-ca-lcx4k                                       1/1     Running
0          33d
pod/node-ca-v7zmx                                       1/1     Running
0          7d21h
pod/node-ca-xpppp                                       1/1     Running
0          89d

NAME                                TYPE        CLUSTER-IP       EXTERNAL-
IP   PORT(S)     AGE
service/image-registry              ClusterIP   172.30.196.167   <none>
5000/TCP    15h
service/image-registry-operator     ClusterIP   None             <none>
60000/TCP   90d

NAME                       DESIRED   CURRENT   READY   UP-TO-DATE
AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/node-ca   6         6         6       6            6
kubernetes.io/os=linux   90d

NAME                                                 READY   UP-TO-DATE
AVAILABLE   AGE
deployment.apps/cluster-image-registry-operator   1/1     1            1
90d
deployment.apps/image-registry                    1/1     1            1
15h

NAME                                                            DESIRED
CURRENT   READY   AGE
replicaset.apps/cluster-image-registry-operator-74f6d954b6   1         1
1       90d
```

```
replicaset.apps/image-registry-6758b547f                    1         1
1       76m
replicaset.apps/image-registry-78bfbd7f59                   0         0
0       15h
replicaset.apps/image-registry-7fcc8d6cc8                   0         0
0       80m
replicaset.apps/image-registry-864f88f5b                    0         0
0       15h
replicaset.apps/image-registry-cb47fffb                     0         0
0       10h

NAME                                COMPLETIONS   DURATION   AGE
job.batch/image-pruner-1627257600   1/1           10s        2d9h
job.batch/image-pruner-1627344000   1/1           6s         33h
job.batch/image-pruner-1627430400   1/1           5s         9h

NAME                          SCHEDULE    SUSPEND   ACTIVE   LAST
SCHEDULE   AGE
cronjob.batch/image-pruner   0 0 * * *   False     0        9h
90d

NAME                                      HOST/PORT
PATH   SERVICES         PORT    TERMINATION   WILDCARD
route.route.openshift.io/public-routes   astra-registry.apps.ocp-
vmw.cie.netapp.com          image-registry   <all>   reencrypt     None
```

6. 如果您使用入口操作員OpenShift登錄路由的預設TLS憑證、則可以使用下列命令擷取TLS憑證。

```
[netapp-user@rhel7 ~]$ oc extract secret/router-ca --keys=tls.crt -n
openshift-ingress-operator
```

7. 若要允許OpenShift節點存取及從登錄中提取影像、請將憑證新增至OpenShift節點上的Docker用戶端。使用TLS憑證在「openshift-config」命名空間中建立組態對應、並將其修補至叢集映像組態、使憑證成為信任的憑證。

```
[netapp-user@rhel7 ~]$ oc create configmap astra-ca -n openshift-config
--from-file=astra-registry.apps.ocp-vmw.cie.netapp.com=tls.crt

[netapp-user@rhel7 ~]$ oc patch image.config.openshift.io/cluster
--patch '{"spec":{"additionalTrustedCA":{"name":"astra-ca"}}}'
--type=merge
```

8. OpenShift內部登錄是由驗證控制。所有OpenShift使用者都能存取OpenShift登錄、但登入使用者可以執行的作業取決於使用者權限。

a. 若要允許使用者或使用者群組從登錄擷取映像、使用者必須指派登錄檢視器角色。

```
[netapp-user@rhel7 ~]$ oc policy add-role-to-user registry-viewer
ocp-user

[netapp-user@rhel7 ~]$ oc policy add-role-to-group registry-viewer
ocp-user-group
```

b. 若要允許使用者或使用者群組寫入或推送映像、使用者必須指派登錄編輯器角色。

```
[netapp-user@rhel7 ~]$ oc policy add-role-to-user registry-editor
ocp-user

[netapp-user@rhel7 ~]$ oc policy add-role-to-group registry-editor
ocp-user-group
```

9. 若要讓OpenShift節點存取登錄並推送或拉出映像、您需要設定拉出密碼。

```
[netapp-user@rhel7 ~]$ oc create secret docker-registry astra-registry-
credentials --docker-server=astra-registry.apps.ocp-vmw.cie.netapp.com
--docker-username=ocp-user --docker-password=password
```

10. 這種拉出密碼可修補至服務帳戶、或在對應的Pod定義中參考。

a. 若要將IT修補為服務帳戶、請執行下列命令。

```
[netapp-user@rhel7 ~]$ oc secrets link <service_account_name> astra-
registry-credentials --for=pull
```

b. 若要參考Pod定義中的Pull機密、請將下列參數新增至「s pec」區段。

```
imagePullSecrets:
  - name: astra-registry-credentials
```

11. 若要從工作站推送或拉出OpenShift節點以外的映像、請完成下列步驟。

a. 將TLS憑證新增至Docker用戶端。

```
[netapp-user@rhel7 ~]$ sudo mkdir /etc/docker/certs.d/astra-
registry.apps.ocp-vmw.cie.netapp.com

[netapp-user@rhel7 ~]$ sudo cp /path/to/tls.crt
/etc/docker/certs.d/astra-registry.apps.ocp-vmw.cie.netapp.com
```

b. 使用occ登入命令登入OpenShift。

```
[netapp-user@rhel7 ~]$ oc login --token=sha256~D49SpB_lesSrJYwrM0LIO
-VRcjWHu0a27vKa0 --server=https://api.ocp-vmw.cie.netapp.com:6443
```

c. 使用podman/Docker命令、使用OpenShift使用者認證登入登錄。

podman

```
[netapp-user@rhel7 ~]$ podman login astra-registry.apps.ocp-
vmw.cie.netapp.com -u kubeadmin -p $(oc whoami -t) --tls
-verify=false
```

+附註：如果您使用「kubeadmin」使用者登入私有登錄、請使用權杖而非密碼。

Docker

```
[netapp-user@rhel7 ~]$ docker login astra-registry.apps.ocp-
vmw.cie.netapp.com -u kubeadmin -p $(oc whoami -t)
```

+附註：如果您使用「kubeadmin」使用者登入私有登錄、請使用權杖而非密碼。

d. 推或拉映像。

podman

```
[netapp-user@rhel7 ~]$ podman push astra-registry.apps.ocp-
vmw.cie.netapp.com/netapp-astra/vault-controller:latest
[netapp-user@rhel7 ~]$ podman pull astra-registry.apps.ocp-
vmw.cie.netapp.com/netapp-astra/vault-controller:latest
```

Docker

```
[netapp-user@rhel7 ~]$ docker push astra-registry.apps.ocp-
vmw.cie.netapp.com/netapp-astra/vault-controller:latest
[netapp-user@rhel7 ~]$ docker pull astra-registry.apps.ocp-
vmw.cie.netapp.com/netapp-astra/vault-controller:latest
```

解決方案驗證與使用案例：採用NetApp的Red Hat OpenShift

本頁所提供的範例包括採用NetApp的Red Hat OpenShift解決方案驗證和使用案例。

- "使用持續儲存設備部署Jenkins CI/CD管道"
- "使用NetApp在Red Hat OpenShift上設定多租戶共享"
- "Red Hat OpenShift虛擬化搭配NetApp ONTAP 產品"
- "採用NetApp的Red Hat OpenShift上的Kubernetes進階叢集管理"

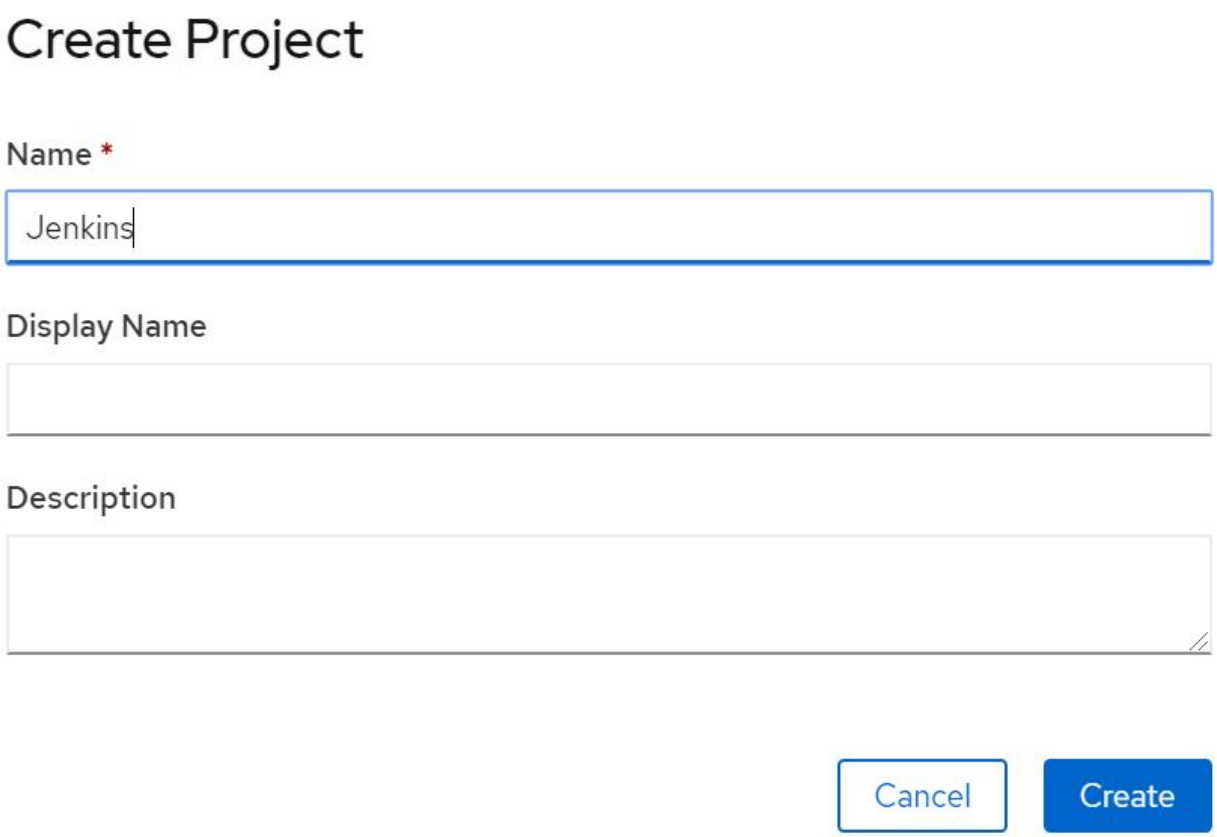
部署Jenkins CI/CD管道搭配持續儲存設備：Red Hat OpenShift with NetApp

本節提供與Jenkins部署持續整合/持續交付或部署（CI/CD）管線的步驟、以驗證解決方案的運作。

建立Jenkins部署所需的資源

若要建立部署Jenkins應用程式所需的資源、請完成下列步驟：

1. 建立名為Jenkins的新專案。



2. 在此範例中、我們部署Jenkins搭配持續儲存設備。若要支援Jenkins建置、請建立永久虛擬基礎架構。瀏覽至「儲存設備」>「持續磁碟區宣告」、然後按一下「建立持續磁碟區宣告」。選取建立的儲存類別、確定「持續Volume宣告名稱」為Jenkins、選取適當的大小和存取模式、然後按一下「建立」。

Project: jenkins

Create Persistent Volume Claim

Edit YAML

Storage Class

SC basic

Storage class for the new claim.

Persistent Volume Claim Name *

jenkins

A unique name for the storage claim within the project.

Access Mode *

Single User (RWO) Shared Access (RWX) Read Only (ROX)

Permissions to the mounted drive.

Size *

100 GiB

Desired storage capacity.

Use label selectors to request storage

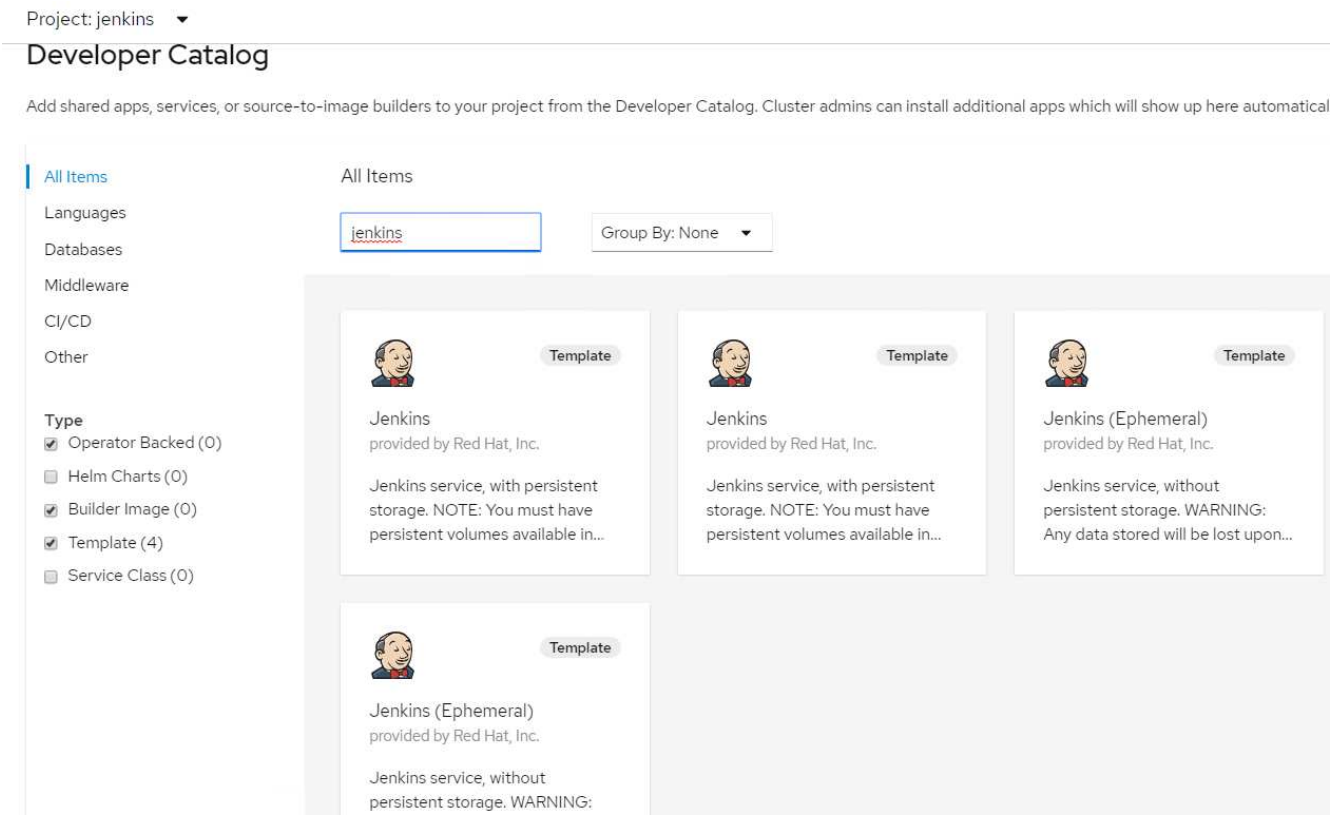
Use label selectors to define how storage is created.

Create Cancel

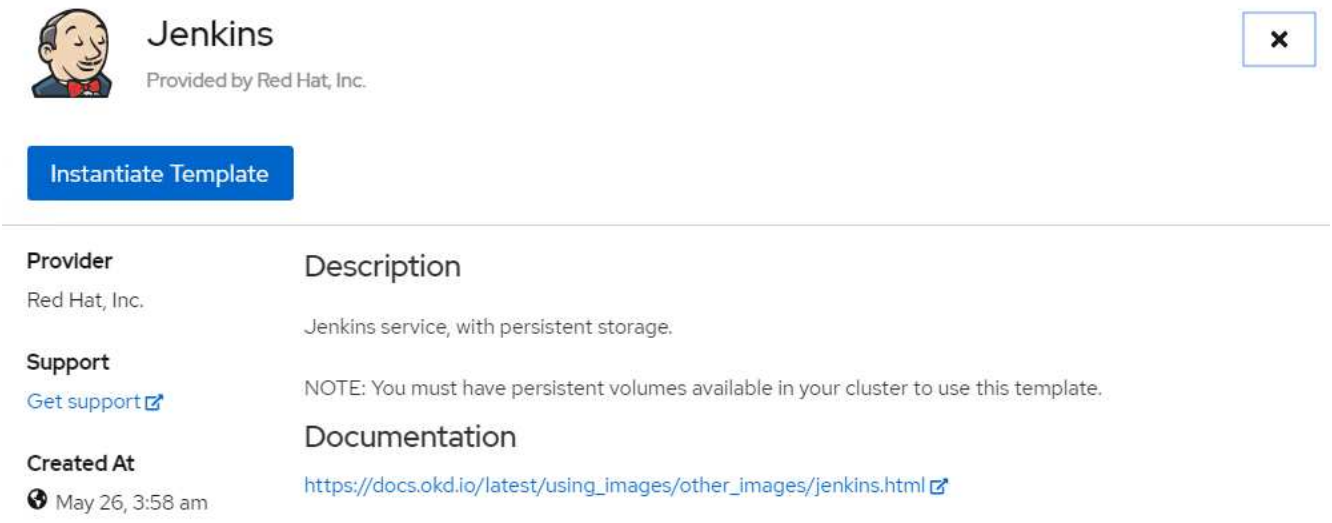
使用持續儲存設備部署**Jenkins**

若要使用持續儲存設備來部署Jenkins、請完成下列步驟：

1. 在左上角、將角色從「管理員」變更為「開發人員」。按一下「+新增」、然後從「目錄」中選取。在「依關鍵字篩選」列中、搜尋Jenkins。選取Jenkins Service with Persistent Storage。



2. 按一下「實體化範本」。



3. 根據預設、會填入Jenkins應用程式的詳細資料。根據您的需求修改參數、然後按一下「Create（建立）」。此程序可建立所有必要資源、以支援OpenShift上的Jenkins。

Instantiate Template

Namespace *

PR jenkins

Jenkins Service Name

jenkins

The name of the OpenShift Service exposed for the Jenkins container.

Jenkins JNLP Service Name

jenkins-jnlp

The name of the service used for master/slave communication.

Enable OAuth in Jenkins

true

Whether to enable OAuth OpenShift integration. If false, the static account 'admin' will be initialized with the password 'password'.

Memory Limit

1Gi

Maximum amount of memory the container can use.

Volume Capacity *

50Gi

Volume space available for data, e.g. 512Mi, 2Gi.

Jenkins ImageStream Namespace

openshift

The OpenShift Namespace where the Jenkins ImageStream resides.

Disable memory intensive administrative monitors

false

Whether to perform memory intensive, possibly slow, synchronization with the Jenkins Update Center on start. If true, the Jenkins core update monitor and site warnings monitor are disabled.

Jenkins ImageStreamTag

jenkins:2

Name of the ImageStreamTag to be used for the Jenkins image.

Fatal Error Log File

false

When a fatal error occurs, an error log is created with information and the state obtained at the time of the fatal error.

Allows use of Jenkins Update Center repository with invalid SSL certificate

false

Whether to allow use of a Jenkins Update Center that uses invalid certificate (self-signed, unknown CA). If any value other than 'false', certificate check is bypassed. By default, certificate check is enforced.

Create Cancel

Jenkins

INSTANT-APP JENKINS

View documentation Get support

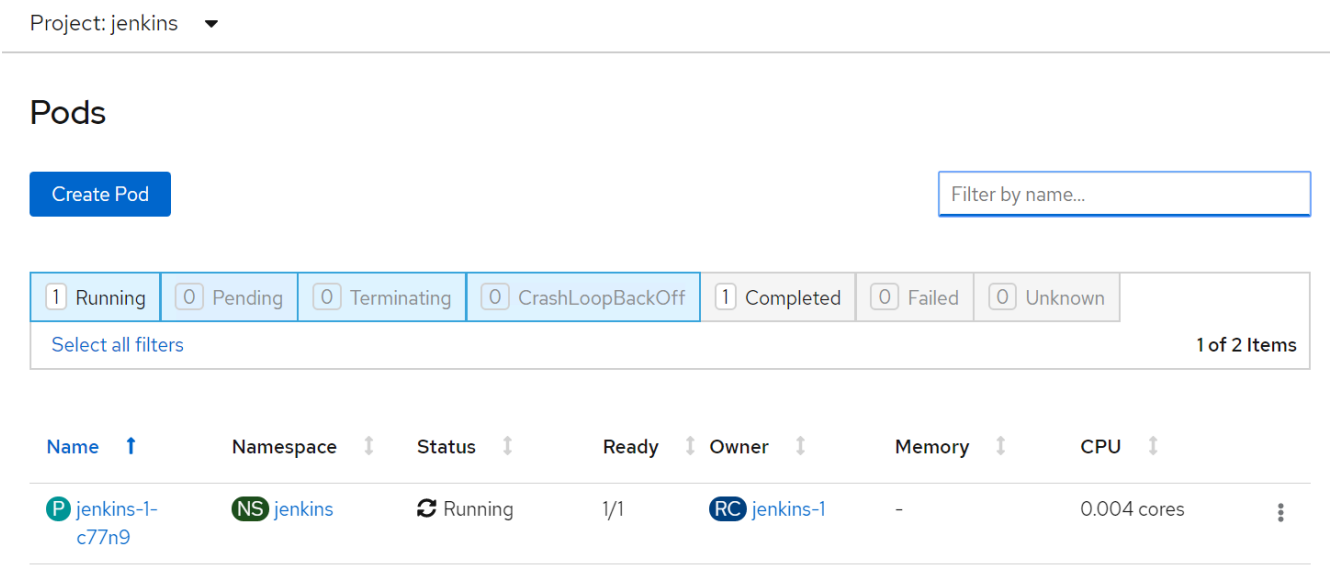
Jenkins service, with persistent storage.

NOTE: You must have persistent volumes available in your cluster to use this template.

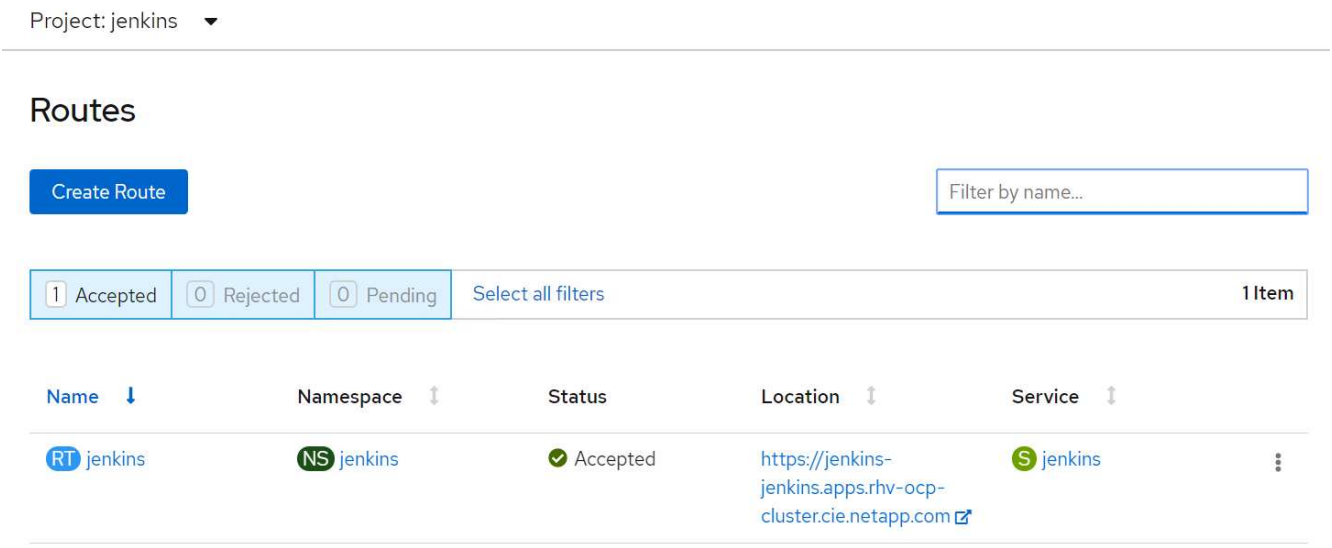
The following resources will be created:

- DeploymentConfig
- PersistentVolumeClaim
- RoleBinding
- Route
- Service
- ServiceAccount

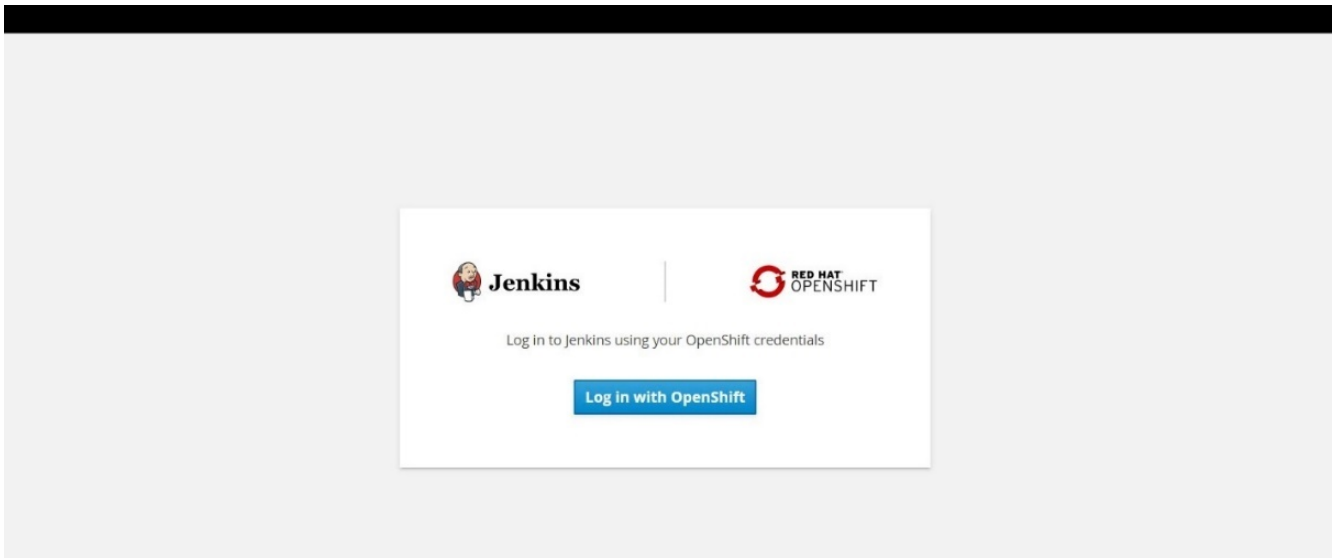
4. Jenkins Pod約需10至12分鐘才能進入就緒狀態。



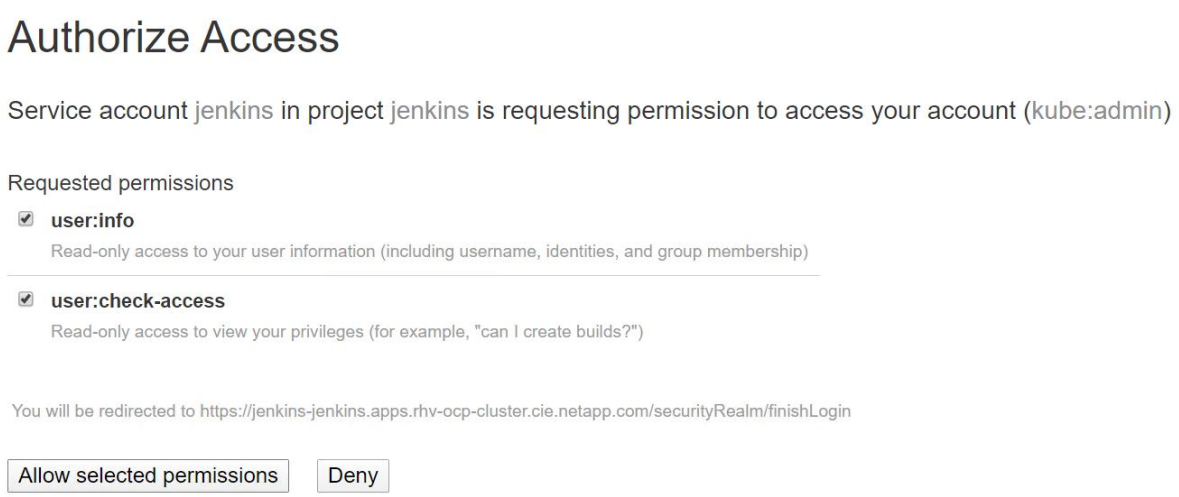
5. 建立Pod之後、請瀏覽至「Networking」（網路）>「Routes」（路由）。若要開啟Jenkins網頁、請按一下Jenkins路由提供的URL。



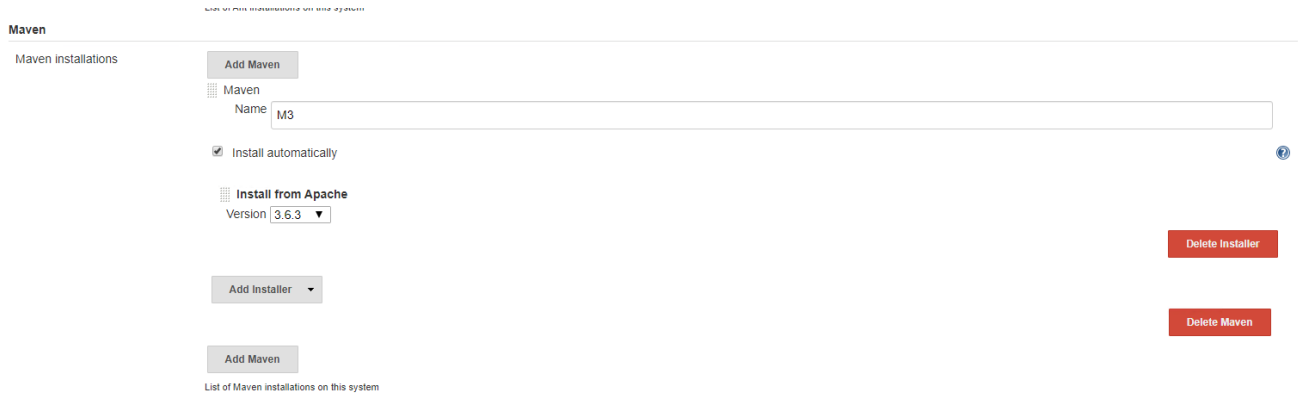
6. 由於在建立Jenkins應用程式時使用OpenShift Oauth、請按一下「使用OpenShift登入」。



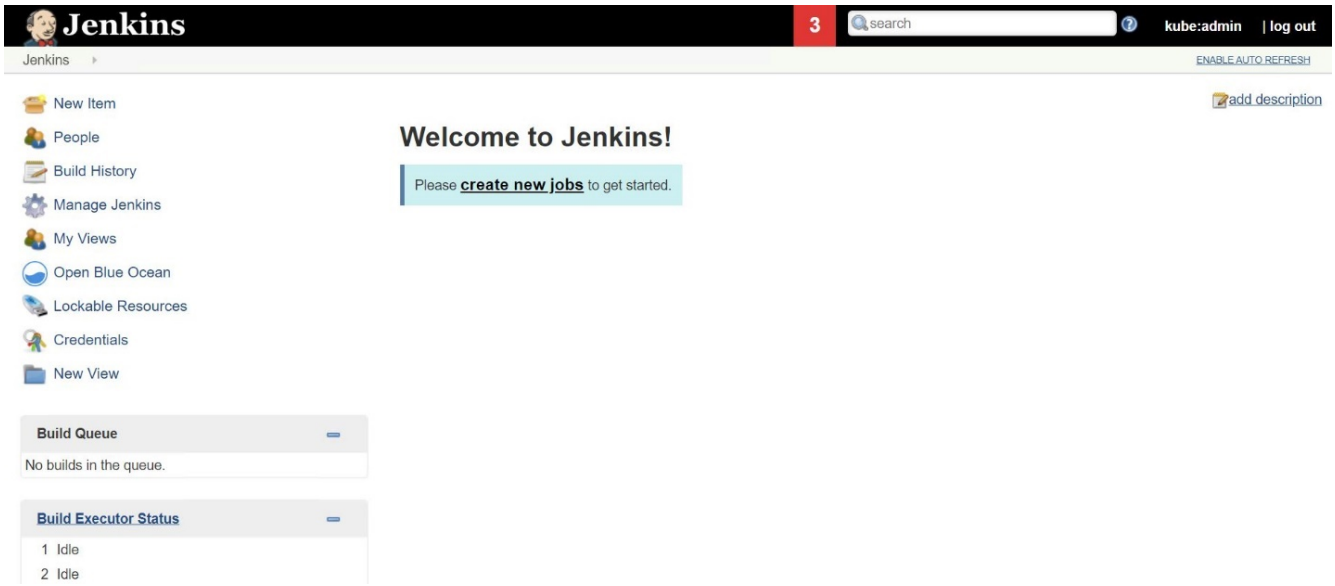
7. 授權Jenkins服務帳戶存取OpenShift使用者。



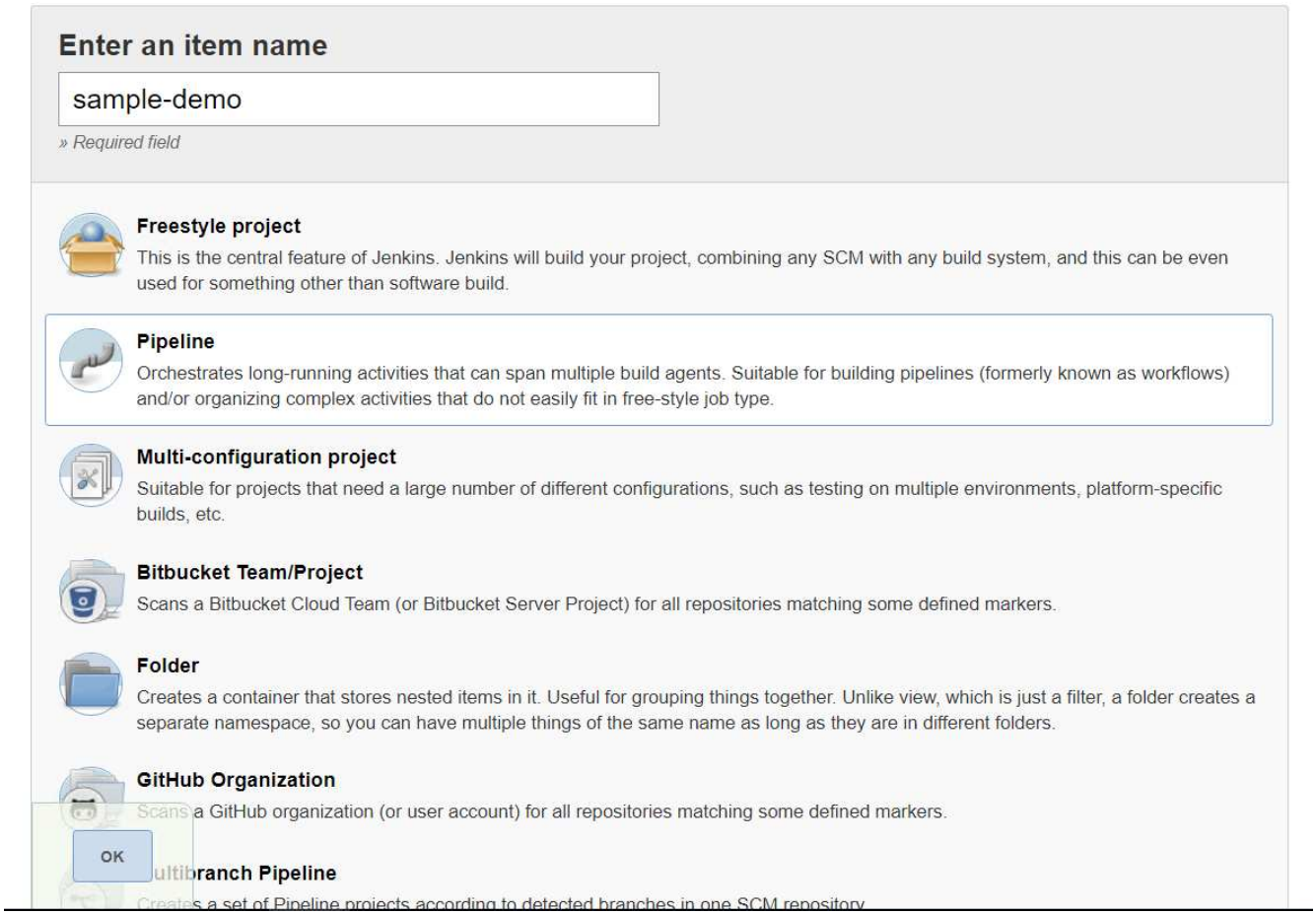
8. 隨即顯示Jenkins歡迎頁面。因為我們使用的是墨文建置、所以請先完成墨文安裝。瀏覽至「Manage Jenkins（管理Jenkins）」>「Global Tool Configuration（全域工具組態）」、然後按一下「Men（新增）」子標題中的「Add Maven（新增輸入您選擇的名稱、並確定已選取「自動安裝」選項。按一下儲存。



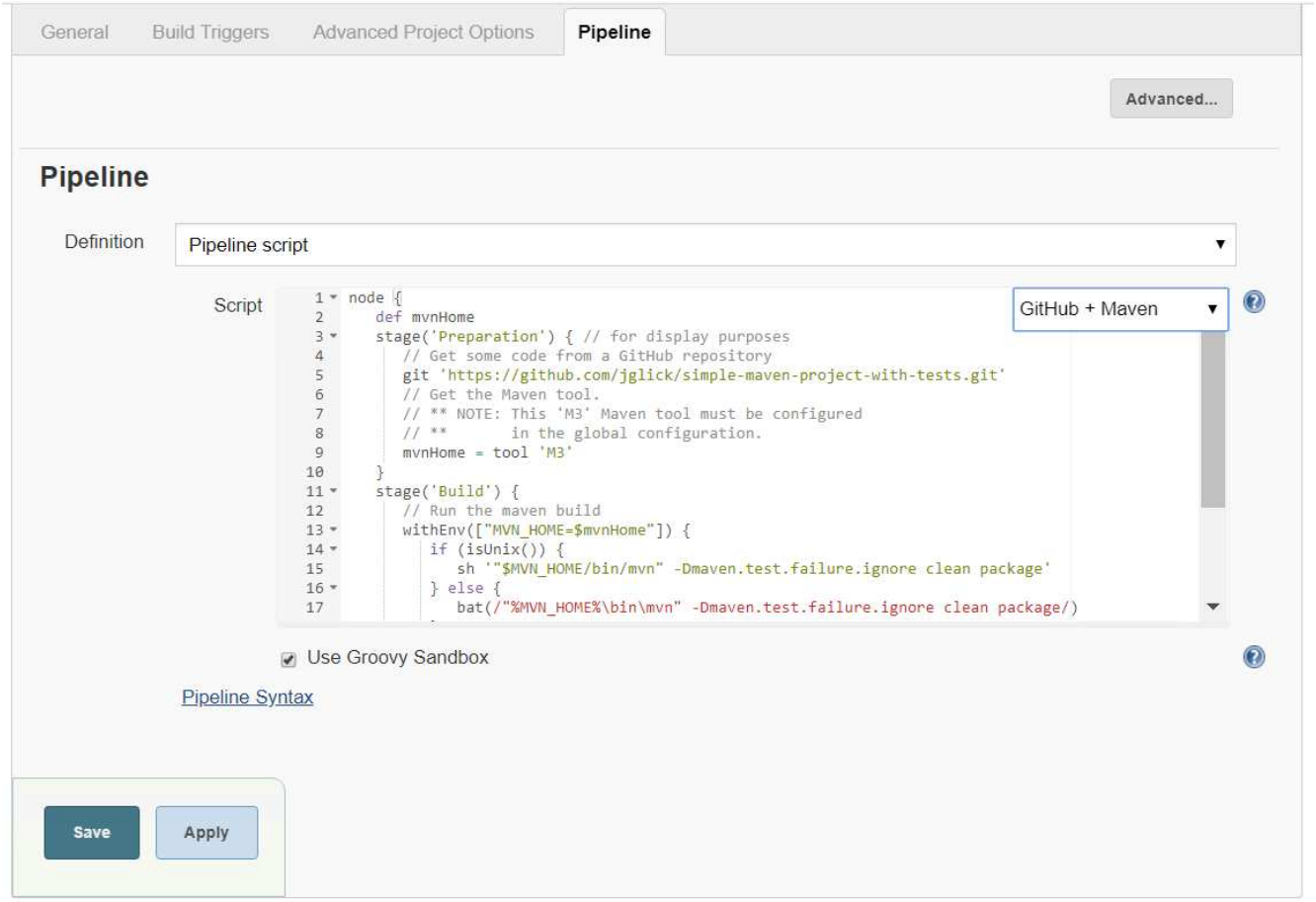
9. 您現在可以建立管道來示範CI/CD工作流程。在首頁上、按一下左側功能表中的「Create New Jobs（建立新工作）」或「New item（新項目）」。



10. 在「Create item（建立項目）」頁面上、輸入您選擇的名稱、選取「Pipeline（管道）」、然後按一下「OK（確定）」。



11. 選取Pipeline（管道）索引標籤。從「試用範例管道」下拉式功能表中、選取「Github + Maven」。程式碼會自動填入。按一下儲存。



12. 按一下「立即建置」、即可在準備、建置和測試階段觸發開發。完成整個建置程序並顯示建置結果可能需要幾分鐘的時間。

13. 只要有任何程式碼變更、就能重新建置管線、修補新版軟體、實現持續整合與持續交付。按一下「近期變更」以追蹤先前版本的變更。

使用**NetApp ONTAP** 功能在**Red Hat OpenShift**上設定多租戶共享

使用**NetApp**在**Red Hat OpenShift**上設定多租戶共享

許多在容器上執行多個應用程式或工作負載的組織、傾向於針對每個應用程式或工作負載部署一個Red Hat OpenShift叢集。這可讓他們針對應用程式或工作負載實作嚴格的隔離、最佳化效能並減少安全性弱點。不過、為每個應用程式部署個別的Red Hat OpenShift叢集、會產生自己的問題集。它會增加營運成本、必須自行監控及管理每個叢集、因為不同應用程式的專屬資源而增加成本、並阻礙有效的擴充性。

若要克服這些問題、您可以考慮在單一Red Hat OpenShift叢集中執行所有應用程式或工作負載。但是在這樣的架構中、資源隔離和應用程式安全性弱點是其中一項重大挑戰。某個工作負載中的任何安全弱點都可能自然延伸到另一個工作負載、進而增加影響區域。此外、任何應用程式突然無法控制的資源使用率、都可能影響另一個應用程式的效能、因為預設不會有資源配置原則。

因此、企業組織希望能在這兩個領域中找到最佳的解決方案、例如允許他們在單一叢集中執行所有工作負載、同時為每個工作負載提供專屬叢集的優點。

其中一個有效的解決方案是在Red Hat OpenShift上設定多租戶共享。多租戶共享是一種架構、可讓多個租戶在同一個叢集上共存、並適當隔離資源、安全性等。在這種情況下、租戶可視為叢集資源的子集、而這些資源已設定為供特定使用者群組專用。在Red Hat OpenShift叢集上設定多租戶共享可提供下列優點：

- 允許共用叢集資源、進而降低資本支出和營運成本
- 降低營運與管理成本
- 保護工作負載免於安全漏洞的交叉污染
- 保護工作負載、避免資源爭用造成非預期的效能降級

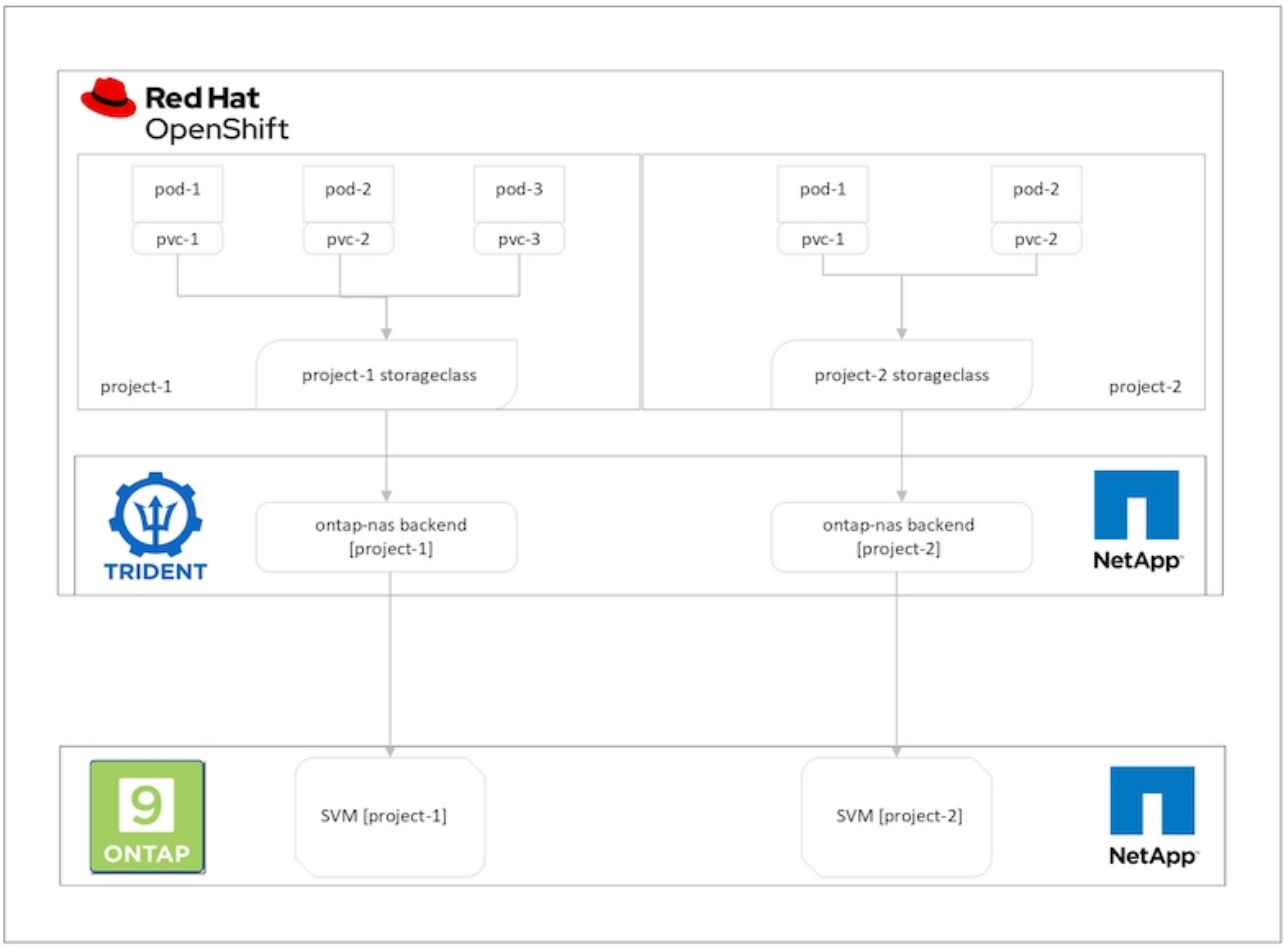
對於完全實現的多租戶OpenShift叢集、必須針對屬於不同資源桶的叢集資源設定配額和限制：運算、儲存、網路、安全性等。雖然我們涵蓋本解決方案中所有資源桶的某些層面、 我們將重點放在隔離和保護同一個Red Hat OpenShift叢集上多個工作負載所提供或使用的資料的最佳實務做法上、方法是在由Astra Trident以NetApp ONTAP S動地 配置的儲存資源上設定多租戶。

架構

雖然Red Hat OpenShift和Astra Trident以NetApp ONTAP 支援、但在預設情況下並未隔離不同的工作負載、但它們提供多種功能、可用來設定多租戶。為了更深入瞭ONTAP 解如何在採用NetApp®技術的Astra Trident的Red Hat OpenShift叢集上設計多租戶解決方案、請讓我們參考一組需求範例、並概述相關的組態。

假設某組織在Red Hat OpenShift叢集上執行兩項工作負載、這是兩個不同團隊正在進行的兩個專案的一部分。這些工作負載的資料位於PVCS上、由Astra Trident在NetApp ONTAP 的NAS後端動態配置。組織必須針對這兩項工作負載設計多租戶解決方案、並隔離用於這些專案的資源、以確保安全性和效能得以維持、主要著重於為這些應用程式提供服務的資料。

下圖說明Red Hat OpenShift叢集上的多租戶解決方案、其中Astra Trident以NetApp ONTAP 效益為後盾。



技術需求

1. NetApp ONTAP 解決方案儲存叢集
2. Red Hat OpenShift叢集
3. Astra Trident

Red Hat OpenShift–叢集資源

從Red Hat OpenShift叢集觀點來看、最重要的資源是專案。OpenShift專案可視為叢集資源、將整個OpenShift叢集分成多個虛擬叢集。因此、專案層級的隔離功能可提供設定多租戶的基礎。

接下來是在叢集中設定RBAC。最佳實務做法是讓所有開發人員在身分識別供應商（IDP）的單一使用者群組中、處理單一專案或工作負載。Red Hat OpenShift允許IDP整合和使用者群組同步、因此可將IDP中的使用者和群組匯入叢集。這有助於叢集管理員將專案專屬的叢集資源存取權、隔離給該專案的使用者群組、進而限制未獲授權存取任何叢集資源。若要深入瞭解IDP與Red Hat OpenShift的整合、請參閱文件 "請按這裡"。

NetApp ONTAP

將共享儲存設備隔離為Red Hat OpenShift叢集的持續儲存提供者非常重要、因為如此可確保在儲存設備上為每個專案建立的磁碟區、在主機看來就像是在不同的儲存設備上建立的磁碟區一樣。為達成此目標、請在NetApp ONTAP 支援上建立數量不限的SVM（儲存虛擬機器）、如同專案或工作負載一樣多、並將每個SVM專用於工作負載。

Astra Trident

在NetApp ONTAP 支援上建立不同專案的不同SVM之後、您必須將每個SVM對應到不同的Trident後端。Trident上的後端組態會將持續儲存設備分配給OpenShift叢集資源、而且需要將SVM的詳細資料對應至該資源。這至少應該是後端的傳輸協定驅動程式。您也可以選擇定義如何在儲存設備上配置磁碟區、以及設定磁碟區大小或集合體使用量等限制。如需有關Trident後端定義的詳細資料、請參閱 "請按這裡"。

Red Hat OpenShift–儲存資源

設定Trident後端之後、下一步是設定StorageClass。設定多個後端儲存類別、讓每個儲存類別都能存取、只在一個後端上增加磁碟區。我們可以在定義儲存類別時、使用storagePools參數、將StorageClass對應至特定的Trident後端。您可以找到定義儲存類別的詳細資料 "請按這裡"。因此、StorageClass與Trident後端之間有一對一對應關係、可指向一個SVM。如此可確保透過指派給該專案的StorageClass進行的所有儲存設備宣告、均由專屬該專案的SVM提供服務。

由於儲存類別並非命名資源、我們如何確保另一個命名空間或專案中的Pod對某個專案的儲存類別提出的要求遭到拒絕？答案是使用資源配額。資源配額是控制每個專案資源總使用量的物件。它可以限制專案中物件可耗用的資源數量和總容量。使用資源配額幾乎可以限制專案的所有資源、而且有效率地使用資源、有助於組織降低因資源過度配置或過度使用而造成的成本與中斷。請參閱文件 "請按這裡" 以取得更多資訊。

在此使用案例中、我們需要限制特定專案中的Pod、使其無法從非專屬專案的儲存類別中申請儲存設備。為達成此目標、我們必須將「<storage-class-name>.storageclass.storage.k8s.io/永久性 磁碟區」設為0、以限制其他儲存類別的持續磁碟區宣告。此外、叢集管理員必須確保專案中的開發人員不應擁有修改資源配額的存取權。

組態

對於任何多租戶解決方案、任何使用者都無法存取超過所需的叢集資源。因此、要設定為多租戶組態一部分的一整組資源、會分為叢集管理、儲存管理員和開發人員、分別負責每個專案。

下表概述不同使用者要執行的不同工作：

角色	工作
叢集管理	為不同的應用程式或工作負載建立專案
	為儲存管理員建立Cluster角色 和角色繫結
	建立角色與角色繫結、讓開發人員指派特定專案的存取權
	[選用]設定專案以排程特定節點上的Pod
儲存設備管理	在NetApp ONTAP 上建立SVM
	建立Trident後端
	建立StorageClass
	建立儲存資源配額
開發人員	驗證存取權限、以便在指派的專案中建立或修補PVCS或Pod
	驗證存取權限、以在另一個專案中建立或修補PVCS或Pod
	驗證存取權限、以檢視或編輯專案、資源配額和儲存類別

組態

先決條件

- NetApp ONTAP 產品叢集
- Red Hat OpenShift叢集
- 叢集上安裝的Trident
- 安裝了tridentctl和occ工具並新增至$path的管理工作站
- 管理員存取ONTAP 功能
- 叢集管理存取OpenShift叢集
- 叢集已與Identity Provider整合
- 身分識別供應商的設定可有效區分不同團隊中的使用者

組態：叢集管理工作

Red Hat OpenShift叢集管理會執行下列工作：

1. 以叢集管理的身分登入Red Hat OpenShift叢集。
2. 建立兩個對應於不同專案的專案。

```
oc create namespace project-1
oc create namespace project-2
```

3. 建立專案1的開發人員角色。

```
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-1
  name: developer-project-1
rules:
  - verbs:
      - '*'
    apiGroups:
      - apps
      - batch
      - autoscaling
      - extensions
      - networking.k8s.io
      - policy
      - apps.openshift.io
      - build.openshift.io
      - image.openshift.io
      - ingress.operator.openshift.io
      - route.openshift.io
      - snapshot.storage.k8s.io
      - template.openshift.io
    resources:
      - '*'
  - verbs:
      - '*'
    apiGroups:
      - ''
    resources:
      - bindings
      - configmaps
      - endpoints
      - events
      - persistentvolumeclaims
      - pods
      - pods/log
      - pods/attach
      - podtemplates
      - replicationcontrollers
      - services
      - limitranges
      - namespaces
      - componentstatuses
```

```
      - nodes
  - verbs:
      - '*'
    apiGroups:
      - trident.netapp.io
    resources:
      - tridentsnapshots
EOF
```

> 本節提供的角色定義只是一個範例。開發人員角色必須根據終端使用者需求加以定義。

1. 同樣地、請為專案2建立開發人員角色。
2. 所有OpenShift和NetApp儲存資源通常由儲存管理員管理。儲存管理員的存取權由安裝Trident時所建立的Trident操作員角色控制。此外、儲存管理員也需要存取資源配額、才能控制儲存設備的使用方式。
3. 在叢集中的所有專案中建立管理資源配額的角色、以將其附加至儲存設備管理員。

```
cat << EOF | oc create -f -
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: resource-quotas-role
rules:
  - verbs:
      - '*'
    apiGroups:
      - ''
    resources:
      - resourcequotas
  - verbs:
      - '*'
    apiGroups:
      - quota.openshift.io
    resources:
      - '*'
EOF
```

4. 請確定叢集已與組織的身分識別提供者整合、而且使用者群組已與叢集群組同步。下列範例顯示身分識別提供者已與叢集整合、並與使用者群組同步。

```
$ oc get groups
NAME                          USERS
ocp-netapp-storage-admins     ocp-netapp-storage-admin
ocp-project-1                 ocp-project-1-user
ocp-project-2                 ocp-project-2-user
```

1. 為儲存管理員設定Cluster勞力 綁定。

```
cat << EOF | oc create -f -
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-trident-operator
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-operator
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-resource-quotas-cr
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: resource-quotas-role
EOF
```

對於儲存管理員、必須綁定兩個角色：Trident運算子和資源配額。

1. 為開發人員建立角色連結、將開發人員專案1角色繫結至專案1中對應的群組（OCP專案-1）。

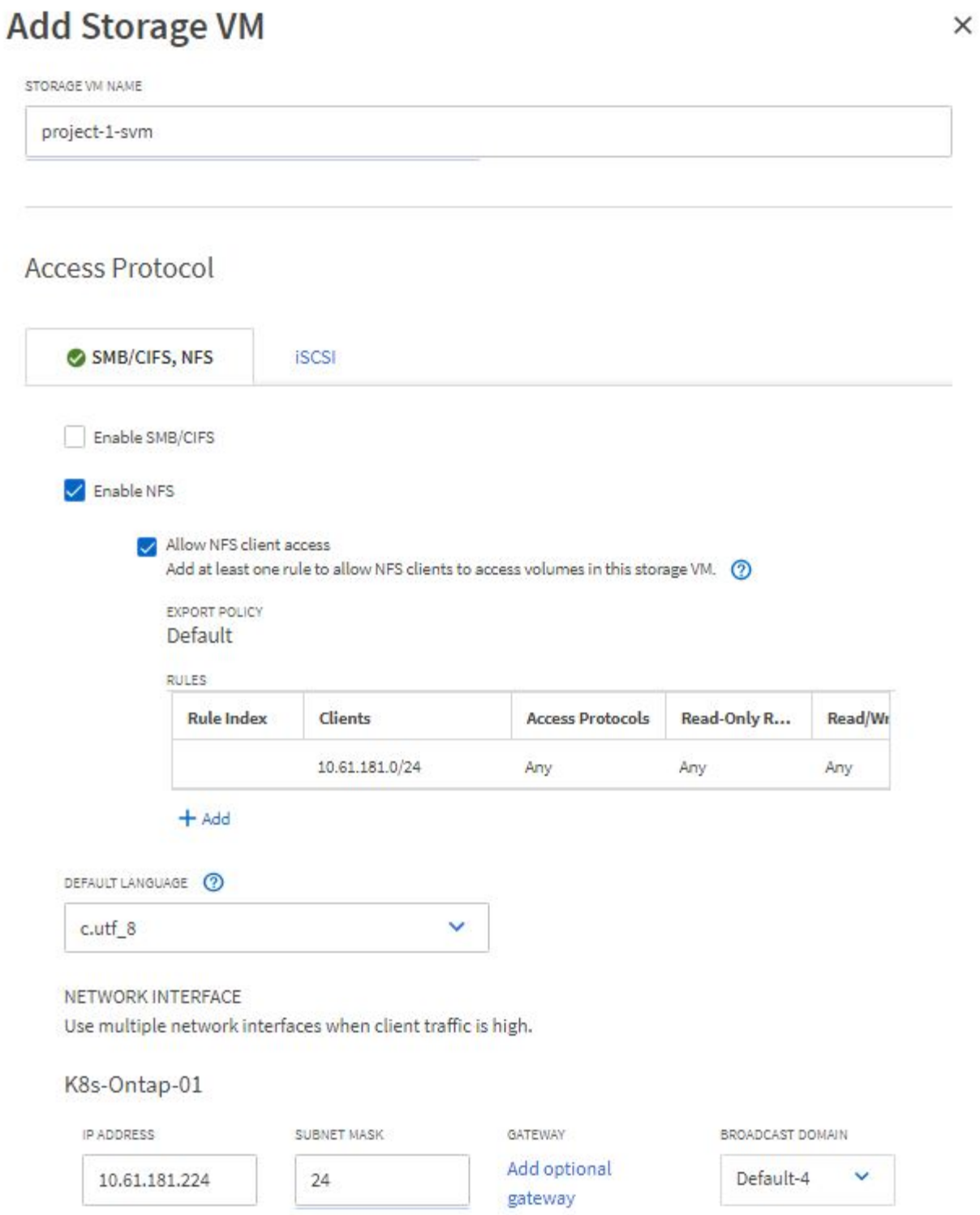
```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-1-developer
  namespace: project-1
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-1
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-1
EOF
```

2. 同樣地、請為開發人員建立角色連結至專案2中對應的使用者群組的角色連結。

組態：儲存設備管理工作

儲存管理員必須設定下列資源：

1. 以admin身分登入NetApp ONTAP 解決方案叢集。
2. 瀏覽至Storage（儲存設備）> Storage VM（儲存設備VM）、然後按一下Add提供所需的詳細資料、建立兩個SVM、一個用於專案1、另一個用於專案2。也可建立vsadmin帳戶來管理SVM及其資源。



1. 以儲存管理員身分登入Red Hat OpenShift叢集。
2. 建立專案1的後端、並將其對應至專案專用的SVM。NetApp建議使用SVM的vsadmin帳戶、將後端連線至SVM、而非ONTAP 使用該叢集管理員。

```
cat << EOF | tridentctl -n trident create backend -f
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "nfs_project_1",
    "managementLIF": "172.21.224.210",
    "dataLIF": "10.61.181.224",
    "svm": "project-1-svm",
    "username": "vsadmin",
    "password": "NetApp123"
}
EOF
```

本例使用的是ONTAP-NAS驅動程式。根據使用案例建立後端時、請使用適當的驅動程式。

我們假設Trident安裝在Trident專案中。

1. 同樣地、請為專案2建立Trident後端、並將其對應至專案2專用的SVM。
2. 接下來、建立儲存類別。建立專案1的儲存類別、並設定storagePools參數、以使用從專屬後端到專案1的儲存資源池。

```
cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-1-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_1:.*"
EOF
```

3. 同樣地、請為專案2建立儲存類別、並將其設定為使用從專屬後端到專案2的儲存資源池。
4. 建立資源配額、以限制專案1中的資源、要求儲存資源來自其他專案專用的儲存設備。

```
cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-1-sc-rq
  namespace: project-1
spec:
  hard:
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF
```

5. 同樣地、也可以建立資源配額、以限制專案2中的資源、要求儲存資源來自其他專案專用的儲存設備。

驗證

若要驗證先前步驟中設定的多租戶架構、請完成下列步驟：

驗證存取權、以在指派的專案中建立**PVCS**或**Pod**

1. 以專案1的開發人員OCP專案1使用者身分登入。
2. 檢查存取權限以建立新專案。

```
oc create ns sub-project-1
```

3. 在專案1中使用指派給專案1的storageclassed建立一個PVC.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-1
  namespace: project-1
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-1-sc
EOF
```

4. 檢查與室早相關的PV。

```
oc get pv
```

5. 驗證PV及其Volume是否是在專門用於NetApp ONTAP 上專案1的SVM中建立。

```
volume show -vserver project-1-svm
```

6. 在專案1中建立一個Pod、然後掛載上一步建立的永久虛擬儲存設備。

```
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  volumes:
    - name: test-pvc-project-1
      persistentVolumeClaim:
       claimName: test-pvc-project-1
  containers:
    - name: test-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: test-pvc-project-1
EOF
```

7. 檢查Pod是否正在執行、以及是否已掛載磁碟區。

```
oc describe pods test-pvc-pod -n project-1
```

驗證存取權限、以在另一個專案中建立**PVCS**或**Pod**、或使用其他專案專用的資源

1. 以專案1的開發人員OCP專案1使用者身分登入。
2. 使用指派給專案2的儲存裝置在專案1中建立一個PVC.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-1-sc-2
  namespace: project-1
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-2-sc
EOF
```

3. 在專案2中建立一個PVC.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-2-sc-1
  namespace: project-2
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-1-sc
EOF
```

4. 確保未建立PVCS「test-PVC-project - 1-sc-2」和「test-PVC-project - 2-sc-1」。

```
oc get pvc -n project-1
oc get pvc -n project-2
```

5. 在專案2中建立Pod。

```
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  containers:
    - name: test-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
EOF
```

驗證存取權限、以檢視及編輯專案、資源配額和儲存類別

1. 以專案1的開發人員OCP專案1使用者身分登入。
2. 檢查存取權限以建立新專案。

```
oc create ns sub-project-1
```

3. 驗證存取權限以檢視專案。

```
oc get ns
```

4. 檢查使用者是否可以在專案1中檢視或編輯資源配額。

```
oc get resourcequotas -n project-1
oc edit resourcequotas project-1-sc-rq -n project-1
```

5. 驗證使用者是否有權檢視儲存空間。

```
oc get sc
```

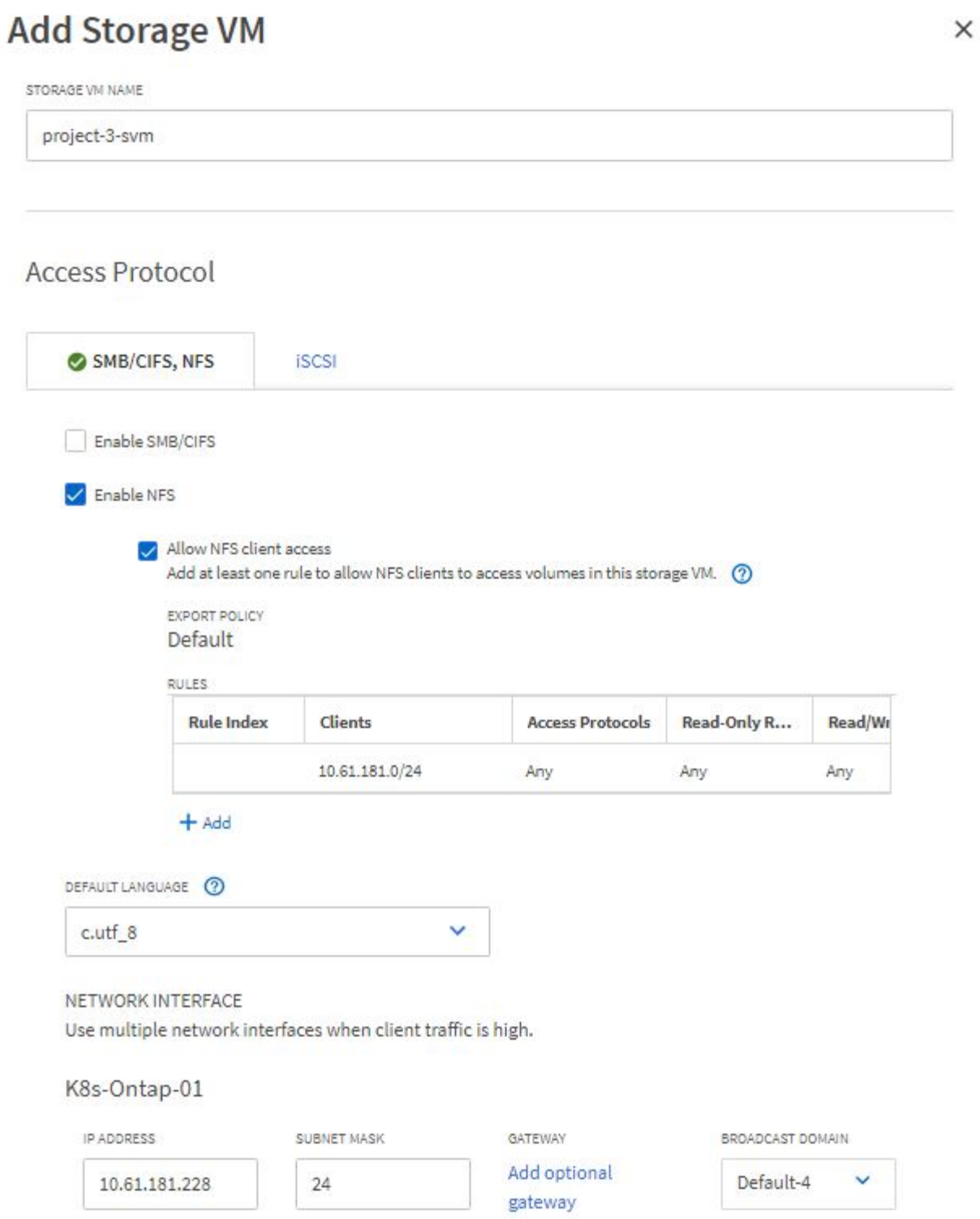
6. 檢查存取以描述儲存空間。
7. 驗證使用者的存取權、以編輯儲存空間。

```
oc edit sc project-1-sc
```

擴充：新增更多專案

在多租戶組態中、新增含有儲存資源的專案需要額外的組態、以確保不違反多租戶共享。若要在多租戶叢集中新增更多專案、請完成下列步驟：

1. 以儲存管理員身分登入NetApp ONTAP 解決方案叢集。
2. 瀏覽至「儲存虛擬機器」、然後按一下「Add（新增）」。建立專案3專用的新SVM。也可建立vsadmin帳戶來管理SVM及其資源。



1. 以叢集管理身分登入Red Hat OpenShift叢集。
2. 建立新專案。

```
oc create ns project-3
```

3. 確認專案3的使用者群組是在IDP上建立、並與OpenShift叢集同步。

```
oc get groups
```

4. 建立專案3的開發人員角色。

```
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-3
  name: developer-project-3
rules:
  - verbs:
      - '*'
    apiGroups:
      - apps
      - batch
      - autoscaling
      - extensions
      - networking.k8s.io
      - policy
      - apps.openshift.io
      - build.openshift.io
      - image.openshift.io
      - ingress.operator.openshift.io
      - route.openshift.io
      - snapshot.storage.k8s.io
      - template.openshift.io
    resources:
      - '*'
  - verbs:
      - '*'
    apiGroups:
      - ''
    resources:
      - bindings
      - configmaps
      - endpoints
      - events
      - persistentvolumeclaims
      - pods
      - pods/log
      - pods/attach
      - podtemplates
```

```
      - replicationcontrollers
      - services
      - limitranges
      - namespaces
      - componentstatuses
      - nodes
  - verbs:
      - '*'
    apiGroups:
      - trident.netapp.io
    resources:
      - tridentsnapshots
EOF
```

本節提供的角色定義只是一個範例。開發人員角色必須根據終端使用者需求加以定義。

1. 在Project 3中為開發人員建立角色繫結、將開發人員專案3角色繫結至專案3中對應的群組（OCP專案3）。

```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-3-developer
  namespace: project-3
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-3
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-3
EOF
```

2. 以儲存管理員身分登入Red Hat OpenShift叢集
3. 建立Trident後端、並將其對應至專案3專用的SVM。NetApp建議使用SVM的vsadmin帳戶、將後端連線至SVM、而非ONTAP 使用叢集管理員。

```
cat << EOF | tridentctl -n trident create backend -f
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "nfs_project_3",
    "managementLIF": "172.21.224.210",
    "dataLIF": "10.61.181.228",
    "svm": "project-3-svm",
    "username": "vsadmin",
    "password": "NetApp!23"
}
EOF
```

本例使用的是ONTAP-NAS驅動程式。根據使用案例、使用適當的驅動程式來建立後端。

我們假設Trident安裝在Trident專案中。

1. 建立專案3的儲存類別、並將其設定為使用從專案3專用後端的儲存資源池。

```
cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-3-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_3:.*"
EOF
```

2. 建立資源配額、以限制專案3中的資源、要求儲存資源來自其他專案專用的儲存設備。

```
cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-3-sc-rq
  namespace: project-3
spec:
  hard:
    project-1-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF
```

3. 修補其他專案中的資源配額、限制這些專案中的資源無法從專案3專用的儲存設備存取儲存設備。

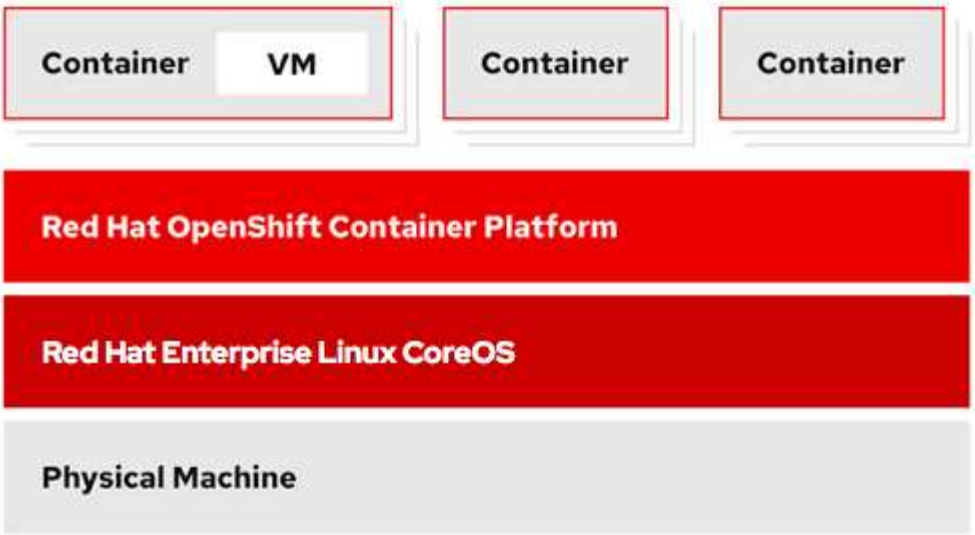
```
oc patch resourcequotas project-1-sc-rq -n project-1 --patch
'{"spec":{"hard":{ "project-3-
sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'
oc patch resourcequotas project-2-sc-rq -n project-2 --patch
'{"spec":{"hard":{ "project-3-
sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'
```

Red Hat OpenShift虛擬化搭配NetApp ONTAP 產品

Red Hat OpenShift虛擬化搭配NetApp ONTAP 產品

根據特定的使用案例、容器和虛擬機器（VM）都能做為不同應用程式類型的最佳平台。因此、許多組織在容器上執行部分工作負載、而在VM上執行部分工作負載。這通常會讓組織面臨更多挑戰、因為必須管理不同的平台：VM的Hypervisor和應用程式的Container Orchestrator。

為了因應這項挑戰、Red Hat從OpenShift版本4.6開始推出OpenShift虛擬化（先前稱為Container Native Virtualization）。OpenShift虛擬化功能可讓您在相同的OpenShift Container Platform安裝上、同時執行及管理虛擬機器與容器、提供混合式管理功能、以便透過操作員自動化VM的部署與管理。除了在OpenShift中建立VM之外、Red Hat還支援從VMware vSphere、Red Hat虛擬化及Red Hat OpenStack平台部署中匯入VM。



OpenShift虛擬化也支援即時VM移轉、VM磁碟複製、VM快照等特定功能、並在由NetApp ONTAP 支援的情況下、由Astra Trident提供協助。本文件稍後將在各自的章節中討論每個工作流程的範例。

若要深入瞭解Red Hat OpenShift虛擬化、請參閱文件 "請按這裡"。

部署 **OpenShift** 虛擬化

部署**Red Hat OpenShift**虛擬化技術搭配**NetApp ONTAP** 功能

先決條件

- Red Hat OpenShift叢集（高於版本4.6）安裝在裸機基礎架構上、並具有RHCOOS工作節點
- OpenShift叢集必須透過安裝程式提供的基礎架構（IPI）進行安裝
- 部署機器健全狀況檢查以維護VM的HA
- NetApp ONTAP 的叢集
- 安裝在OpenShift叢集上的Astra Trident
- Trident後端在ONTAP 叢集上設定SVM
- OpenShift叢集上設定的StorageClass、其中Astra Trident為資源配置程式
- 叢集管理存取Red Hat OpenShift叢集
- 管理員存取NetApp ONTAP 解決方案叢集
- 安裝了tridentctl和occ工具並新增至$path的管理工作站

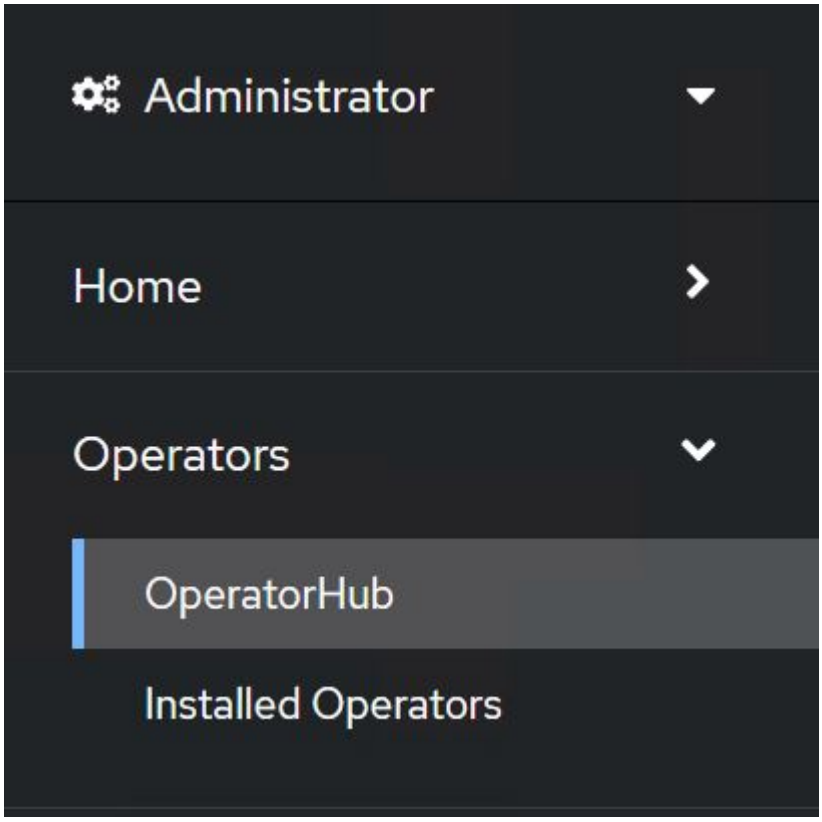
由於OpenShift虛擬化是由安裝在OpenShift叢集上的操作員所管理、因此會對記憶體、CPU和儲存設備產生額外的負荷、因此在規劃叢集的硬體需求時、必須將這些負荷列入考量。請參閱文件 "請按這裡" 以取得更多詳細資料。

或者、您也可以設定節點放置規則、以指定OpenShift叢集節點的子集來裝載OpenShift虛擬化操作員、控制器和VM。若要設定OpenShift虛擬化的節點放置規則、請遵循文件 "請按這裡"。

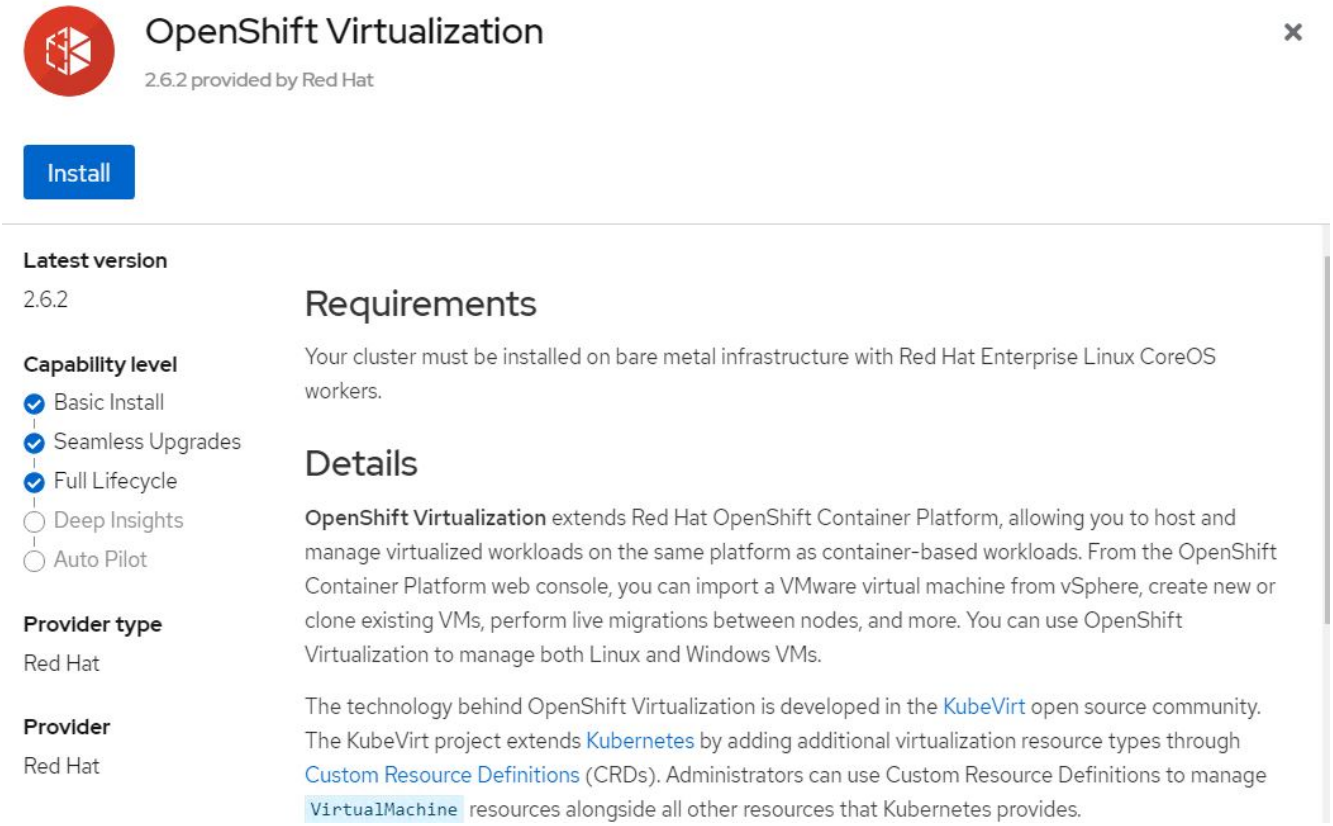
對於支援OpenShift虛擬化的儲存設備、NetApp建議使用專屬StorageClass、從特定Trident後端要求儲存設備、然後再由專屬SVM提供支援。這可維持多租戶層級、以處理OpenShift叢集上VM型工作負載所需的資料。

若要安裝OpenShift虛擬化、請完成下列步驟：

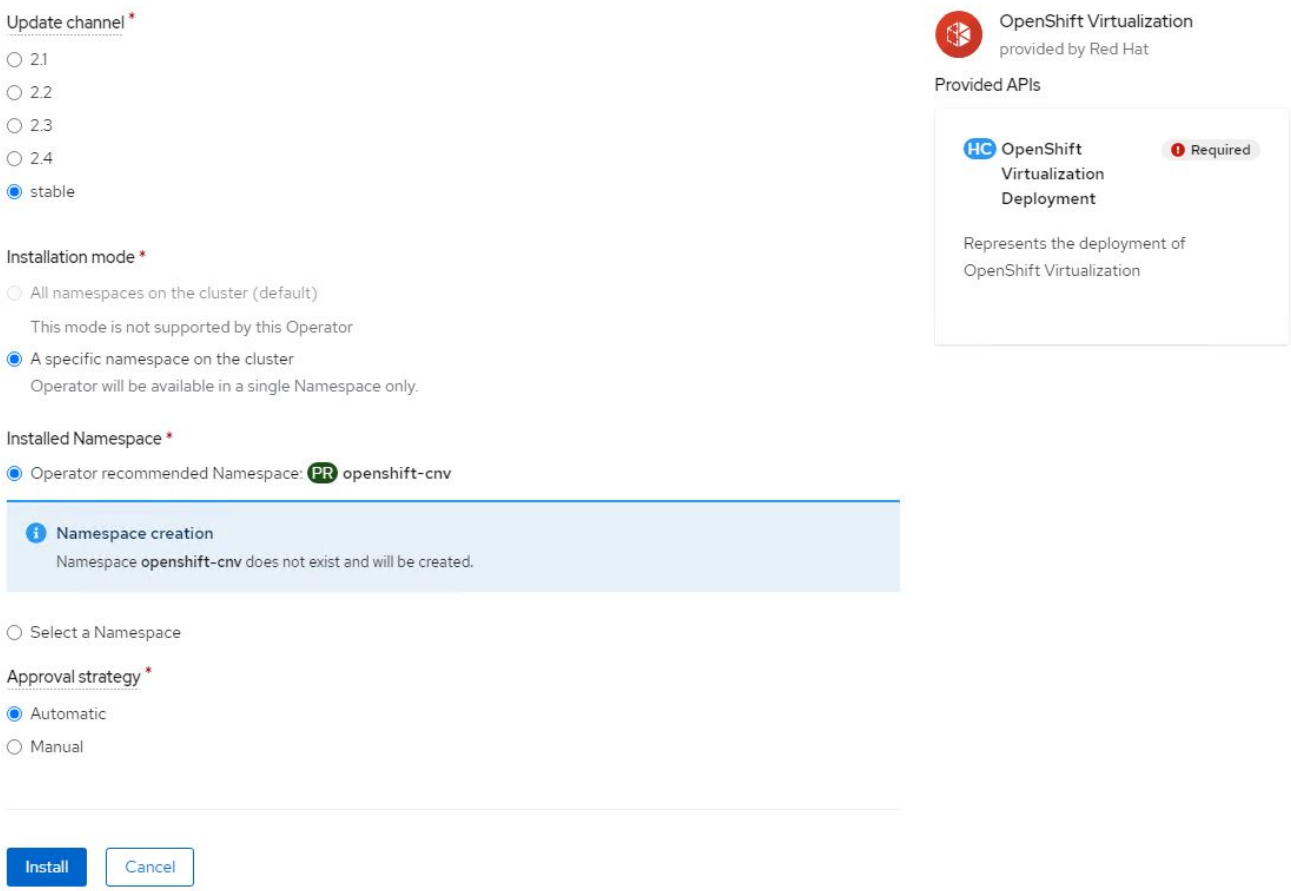
1. 以叢集管理存取權登入Red Hat OpenShift裸機叢集。
2. 從Perspective（透視）下拉列表中選擇Administrator（管理員
3. 瀏覽至「運算子」>「運算子中樞」、然後搜尋OpenShift虛擬化。



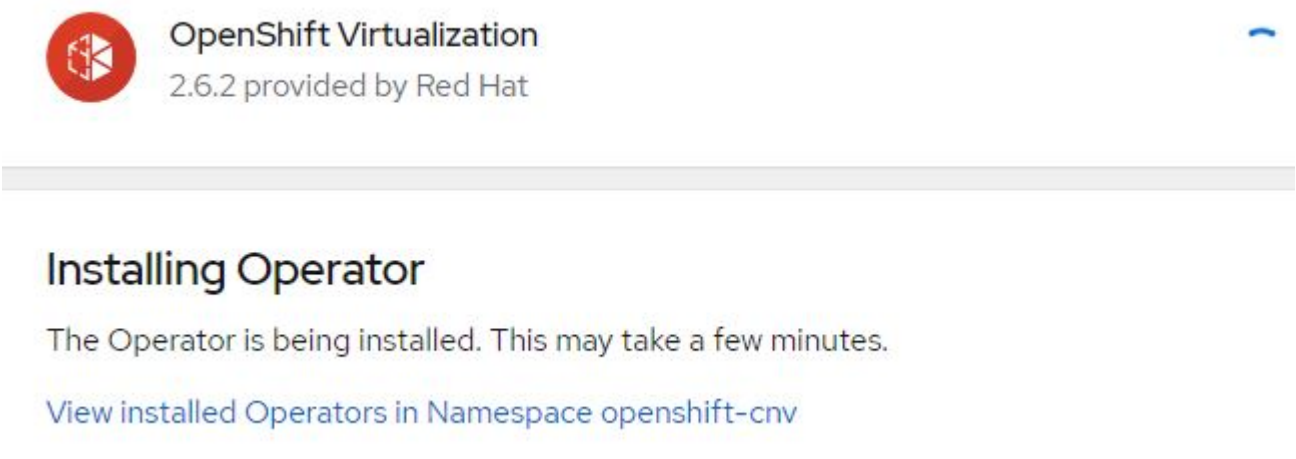
4. 選取OpenShift Virtualization動態磚、然後按一下Install（安裝）。



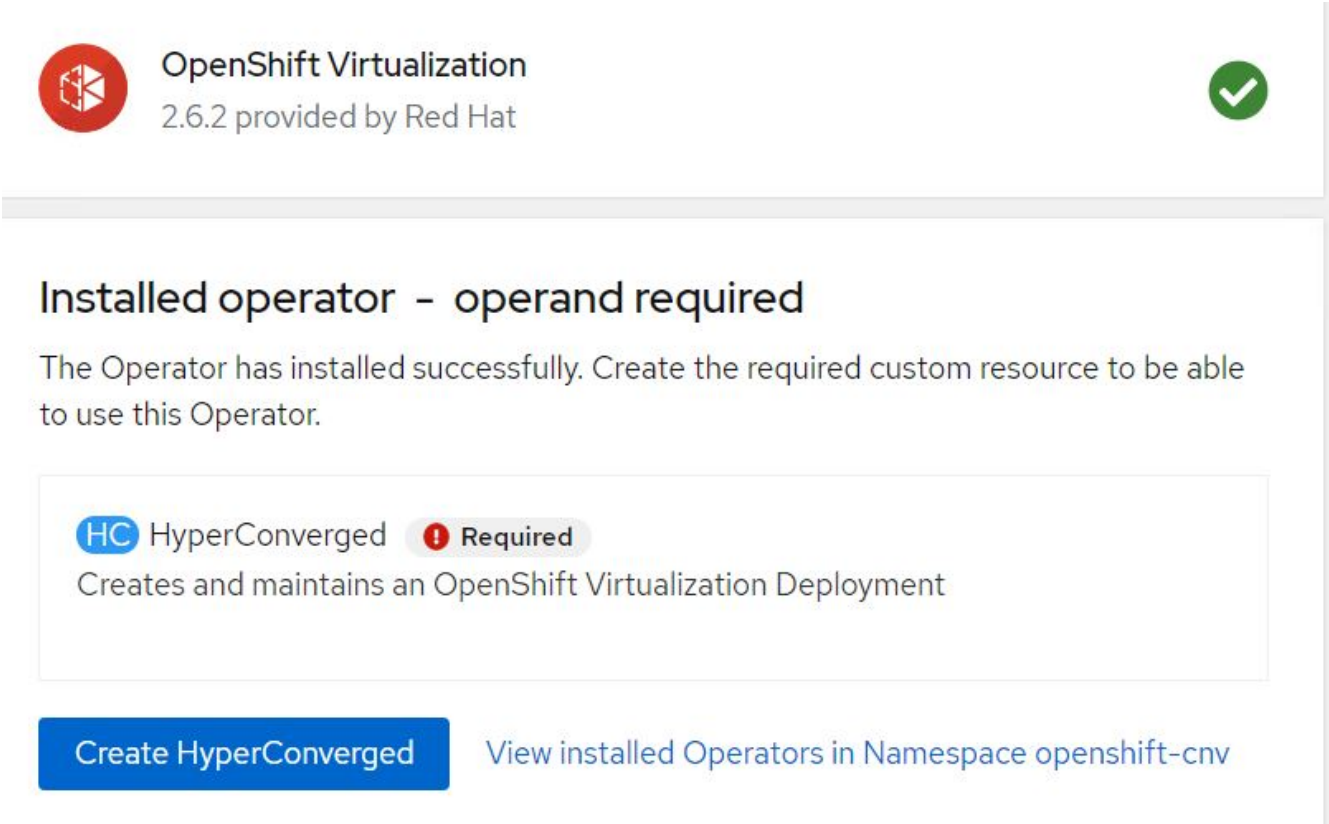
5. 在Install Operator（安裝操作員）畫面上、保留所有預設參數、然後按一下Install（安裝）。



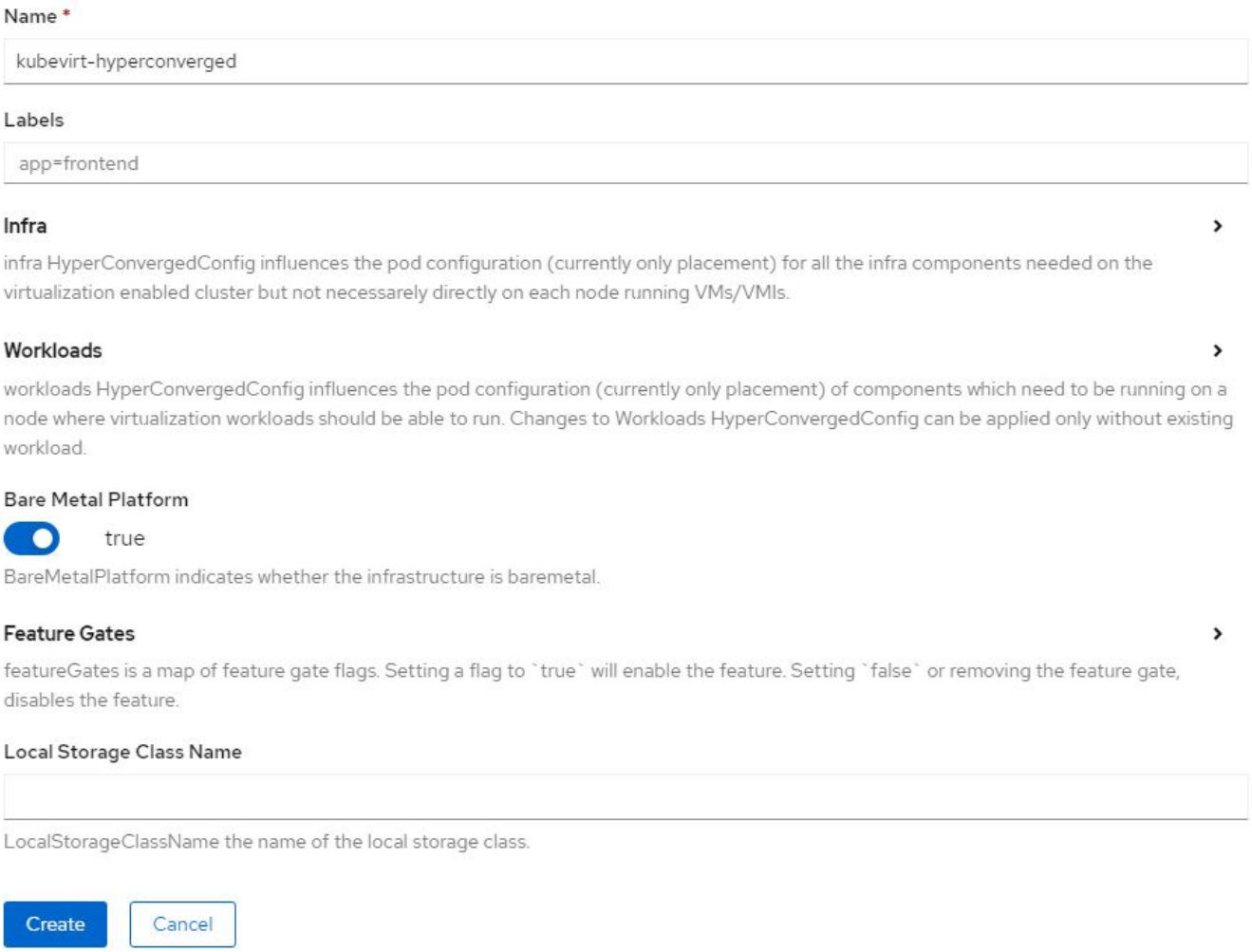
6. 等待操作員安裝完成。



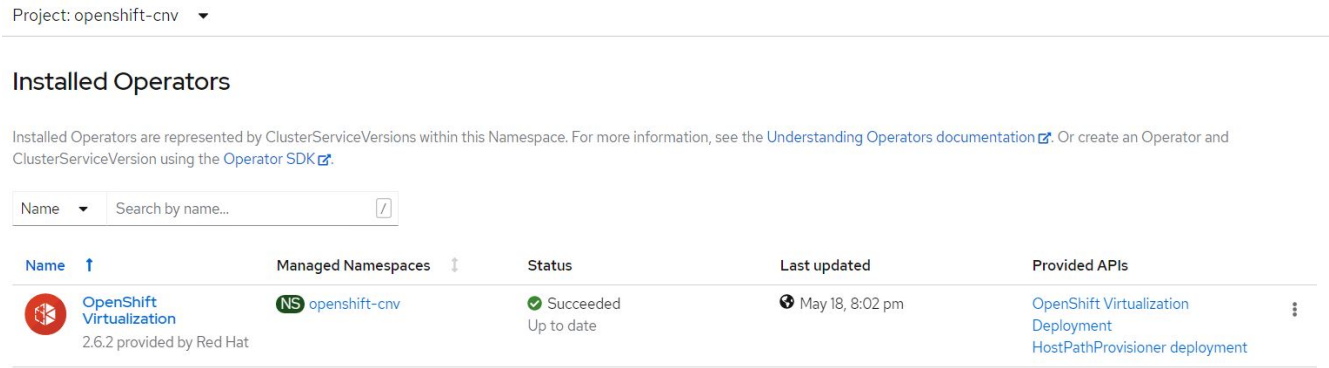
7. 安裝完操作員之後、按一下「Create hyperconverged（建立超融合式）」



8. 在Create hyperconverged（建立超融合式）畫面上、按一下Create（建立）、接受所有預設參數。此步驟會開始安裝OpenShift虛擬化。



9. 在openshift-cnv命名空間中的所有Pod移至執行狀態、且OpenShift虛擬化運算子處於「成功」狀態之後、即可開始使用運算子。現在可以在OpenShift叢集上建立VM。



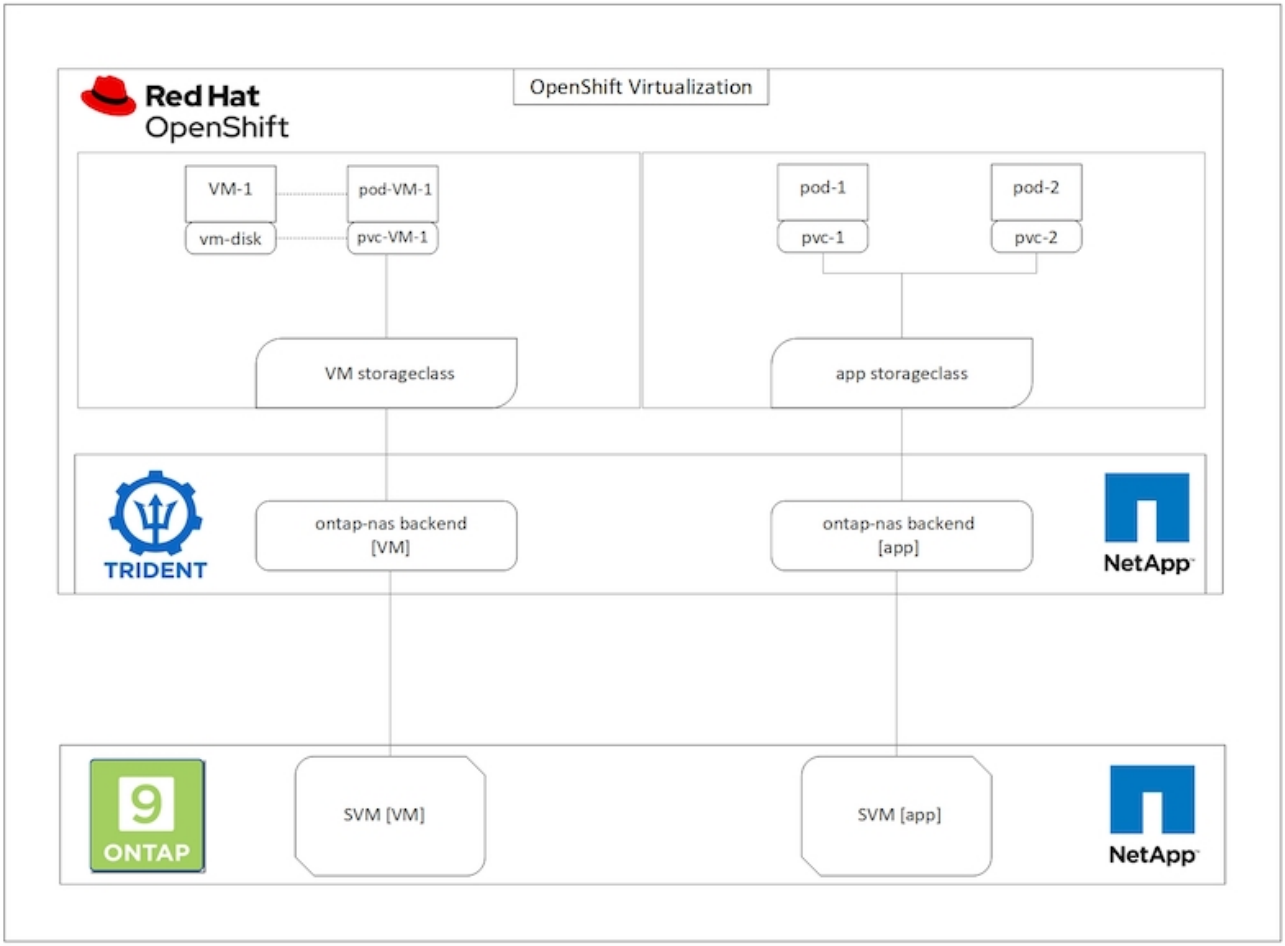
工作流程

工作流程：**Red Hat OpenShift**虛擬化搭配**NetApp ONTAP** 功能

建立**VM**

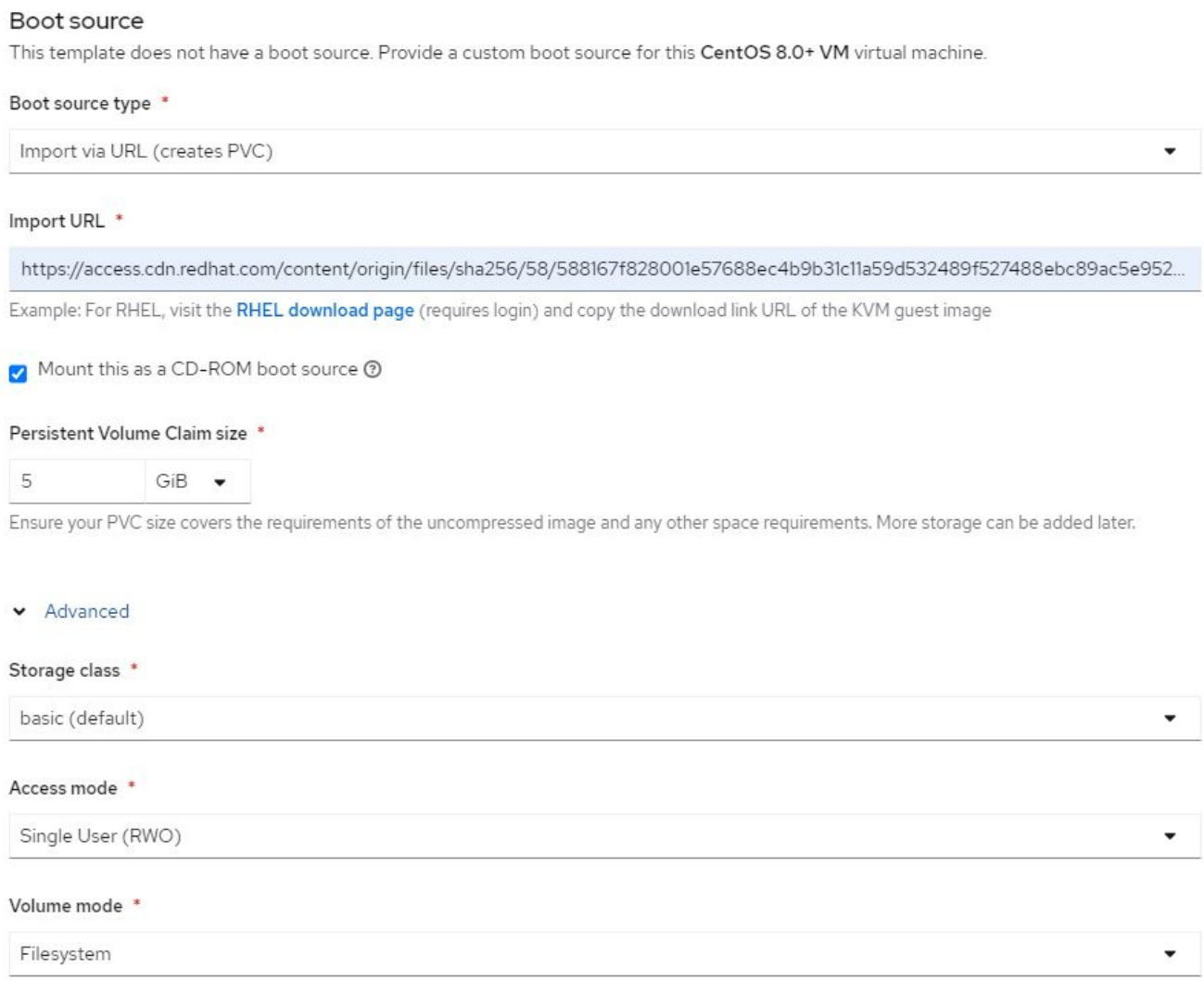
VM是有狀態的部署、需要磁碟區來裝載作業系統和資料。有了CNV、因為VM是以Pod形式執行、所以VM有NetApp ONTAP 透過Trident代管的PV作為後盾。這些磁碟區會附加為磁碟、並儲存整個檔案系統、包括VM

的開機來源。

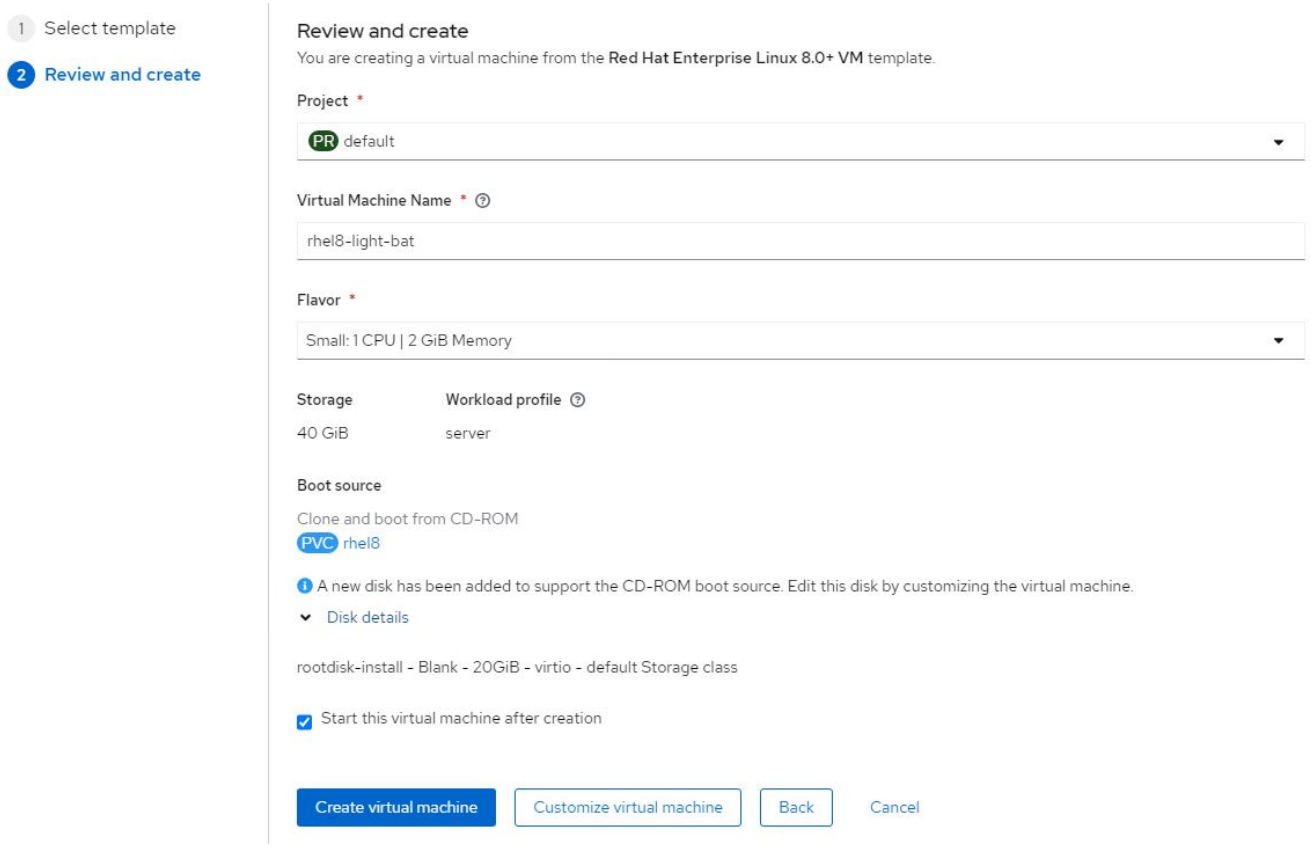


若要在OpenShift叢集上建立虛擬機器、請完成下列步驟：

1. 瀏覽至工作負載>虛擬化>虛擬機器、然後按一下建立>使用精靈。
2. 選取所需的作業系統、然後按「Next（下一步）」。
3. 如果選取的作業系統未設定開機來源、則必須加以設定。針對開機來源、選取是要從URL或登錄匯入OS映像、然後提供對應的詳細資料。展開「進階」、然後選取「Trident備份的StorageClass」。然後按「Next（下一步）」



4. 如果選取的作業系統已設定開機來源、則可以跳過上一個步驟。
5. 在「Review and Create（檢閱與建立）」窗格中、選取您要在其中建立VM的專案、並提供VM詳細資料。請確定已選取要複製的開機來源、並使用指派給所選作業系統的適當永久磁碟從CD-ROM開機。



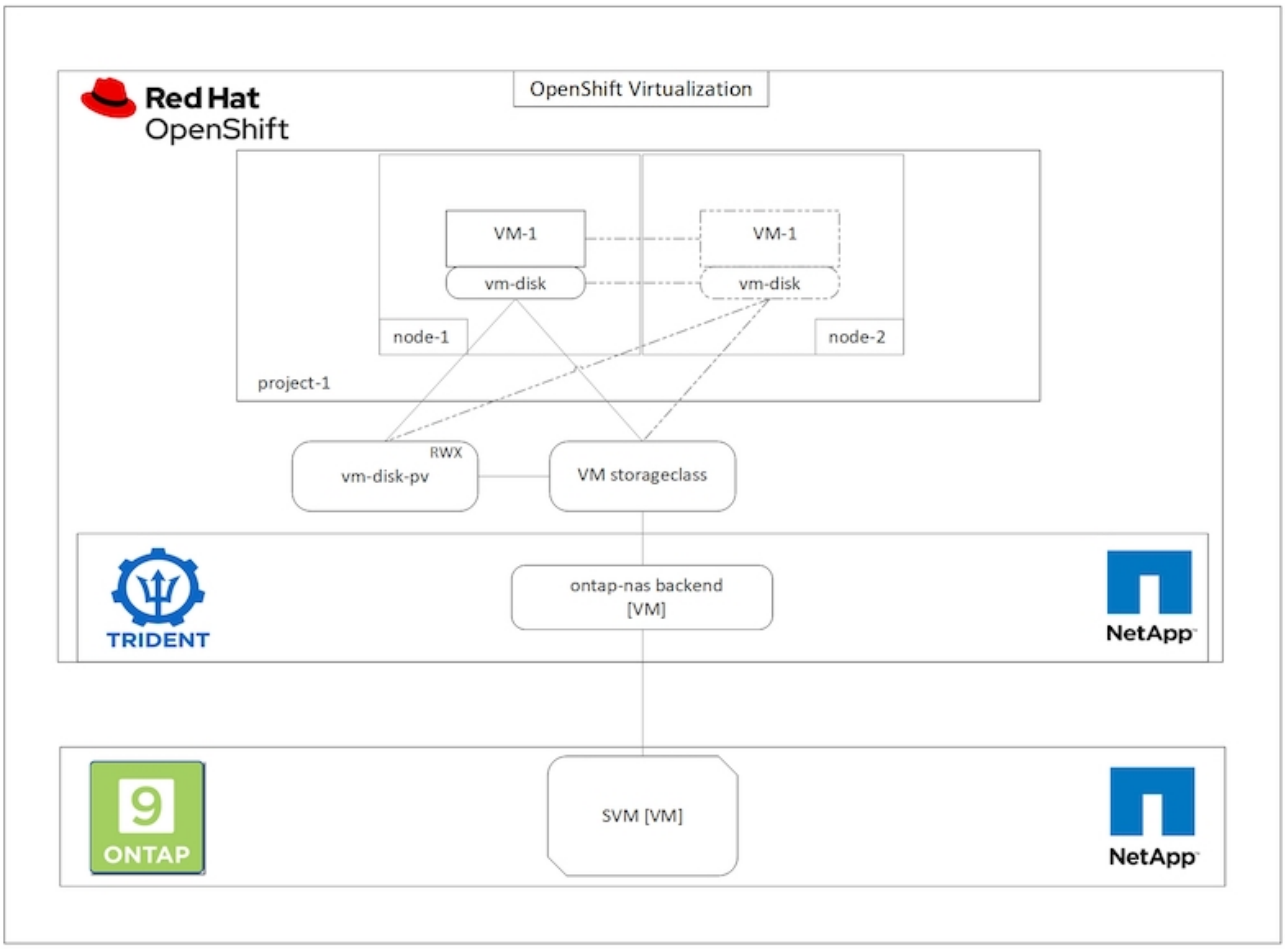
6. 如果您要自訂虛擬機器、請按一下「自訂虛擬機器」、然後修改所需的參數。
7. 按一下「Create Virtual Machine」（建立虛擬機器）以建立虛擬機器；這會使背景中的對應Pod旋轉。

當從URL或登錄設定範本或作業系統的開機來源時、會在「openshift-virtualization作業系統映像」專案中建立一個PVc、並將KVM來賓映像下載至PVc。您必須確定範本PVCS有足夠的資源配置空間、以容納對應作業系統的KVM來賓映像。這些PVCS隨後會以rootdisks的形式複製並附加至虛擬機器、並在任何專案中使用各自的範本建立。

工作流程：**Red Hat OpenShift**虛擬化搭配**NetApp ONTAP** 功能

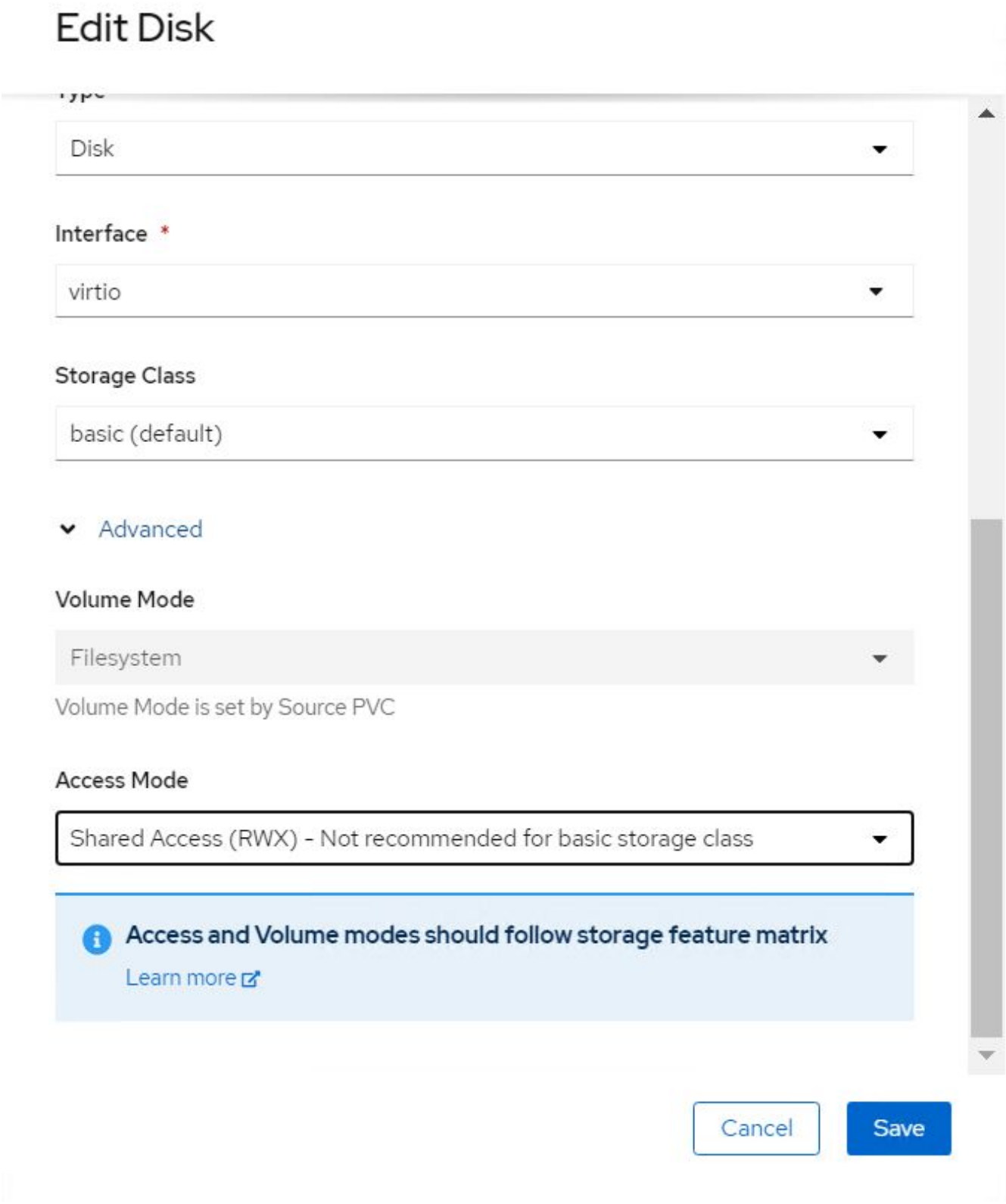
VM即時移轉

即時移轉是將VM執行個體從OpenShift叢集中的某個節點移轉到另一個節點的程序、不會造成停機。若要在OpenShift叢集中執行即時移轉、VM必須繫結至具有共用ReadWriteMany存取模式的PVCS。Astra Trident後端在啟用ONTAP NFS傳輸協定的NetApp支援叢集上設定SVM、可支援對PVCS進行共用的ReadWriteMany存取。因此、從採用NFS的SVM上的Trident配置的StorageClass所要求的具有PVCS的VM、可以在不中斷的情況下進行移轉。



若要建立連結至具有共用ReadWriteMany存取權之PVCS的VM：

1. 瀏覽至工作負載>虛擬化>虛擬機器、然後按一下建立>使用精靈。
2. 選取所需的作業系統、然後按「Next（下一步）」。讓我們假設所選的作業系統已經設定了開機來源。
3. 在「Review and Create（檢閱與建立）」窗格中、選取您要在其中建立VM的專案、並提供VM詳細資料。請確定已選取要複製的開機來源、並使用指派給所選作業系統的適當永久磁碟從CD-ROM開機。
4. 按一下[自訂虛擬機器]，然後按一下[儲存設備]
5. 按一下rootdisk旁的省略符號、並確定已選取使用Trident配置的儲存ageclasse。展開「進階」、然後選取「存取模式」的「共享存取（rwx）」。然後按一下「儲存」。



6. 按一下「Review（檢閱）」並確認、然後按一下「Create Virtual Machine（建立虛

若要將VM手動移轉至OpenShift叢集中的其他節點、請完成下列步驟。

1. 瀏覽至「工作負載」>「虛擬化」>「虛擬機器」。

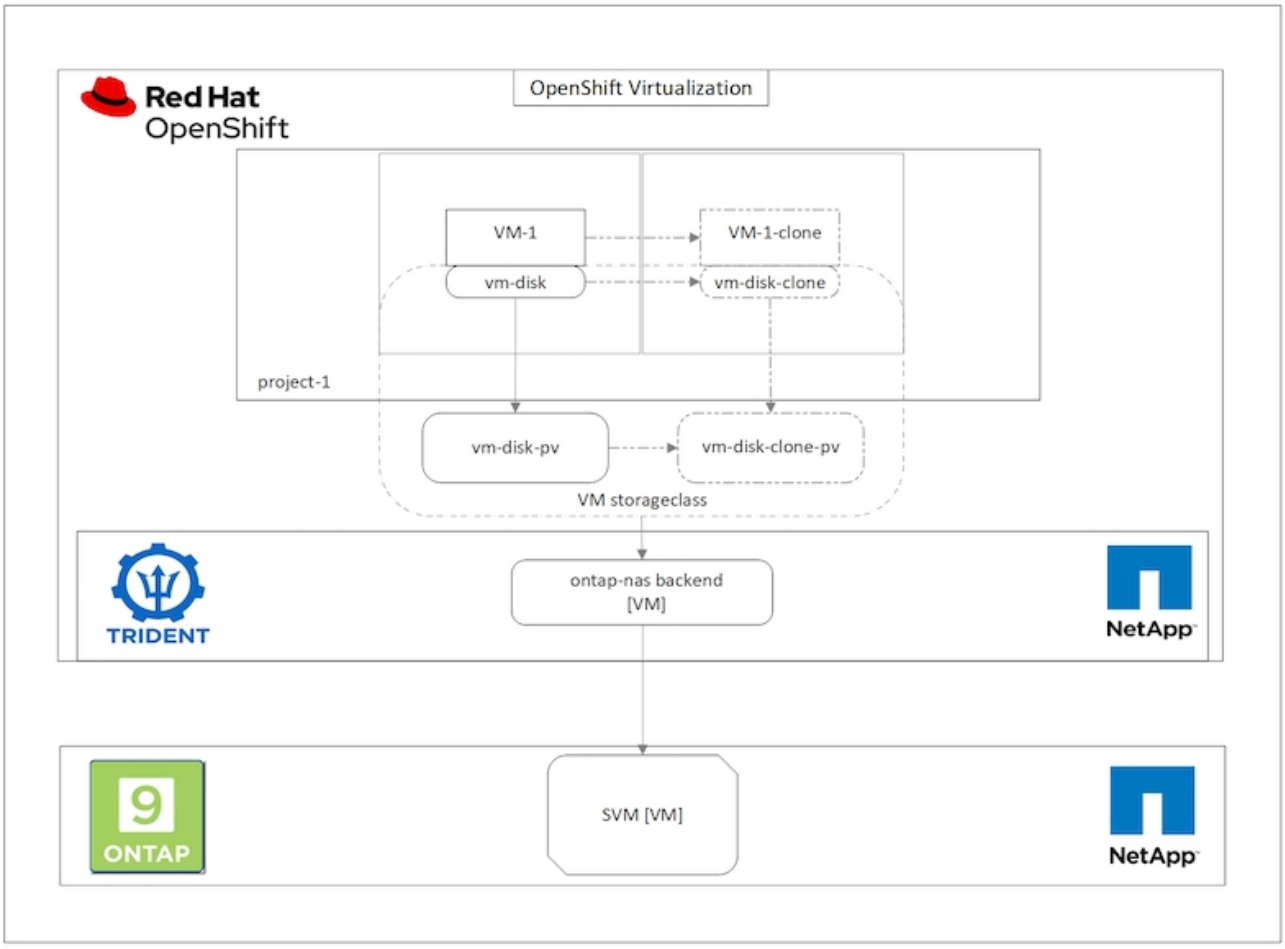
2. 針對您要移轉的VM、按一下省略符號、然後按一下「移轉虛擬機器」。
3. 當訊息快顯以確認時、請按一下「移轉」。

> 如果將設備策略設定為LiveMigrate、則當原始節點置於維護模式時、OpenShift叢集中的VM執行個體會自動移轉至其他節點。

工作流程：**Red Hat OpenShift**虛擬化搭配**NetApp ONTAP** 功能

虛擬機器複製

支援Astra Trident的Volume csi複製功能、即可在OpenShift中複製現有VM。透過複製現有的PV、可以使用現有的PVc作為資料來源來建立新的PVc。建立新的永久虛擬基礎架構之後、它會做為獨立實體運作、而且不會與來源永久虛擬基礎架構有任何連結或相依關係。



使用「csi Volume Cloning」時、必須考量下列限制：

1. 來源PVc和目的地PVc必須位於同一個專案中。
2. 同一儲存類別支援複製。
3. 只有在來源和目的地磁碟區使用相同的磁碟區模式設定時、才能執行複製；例如、區塊磁碟區只能複製到另一個區塊磁碟區。

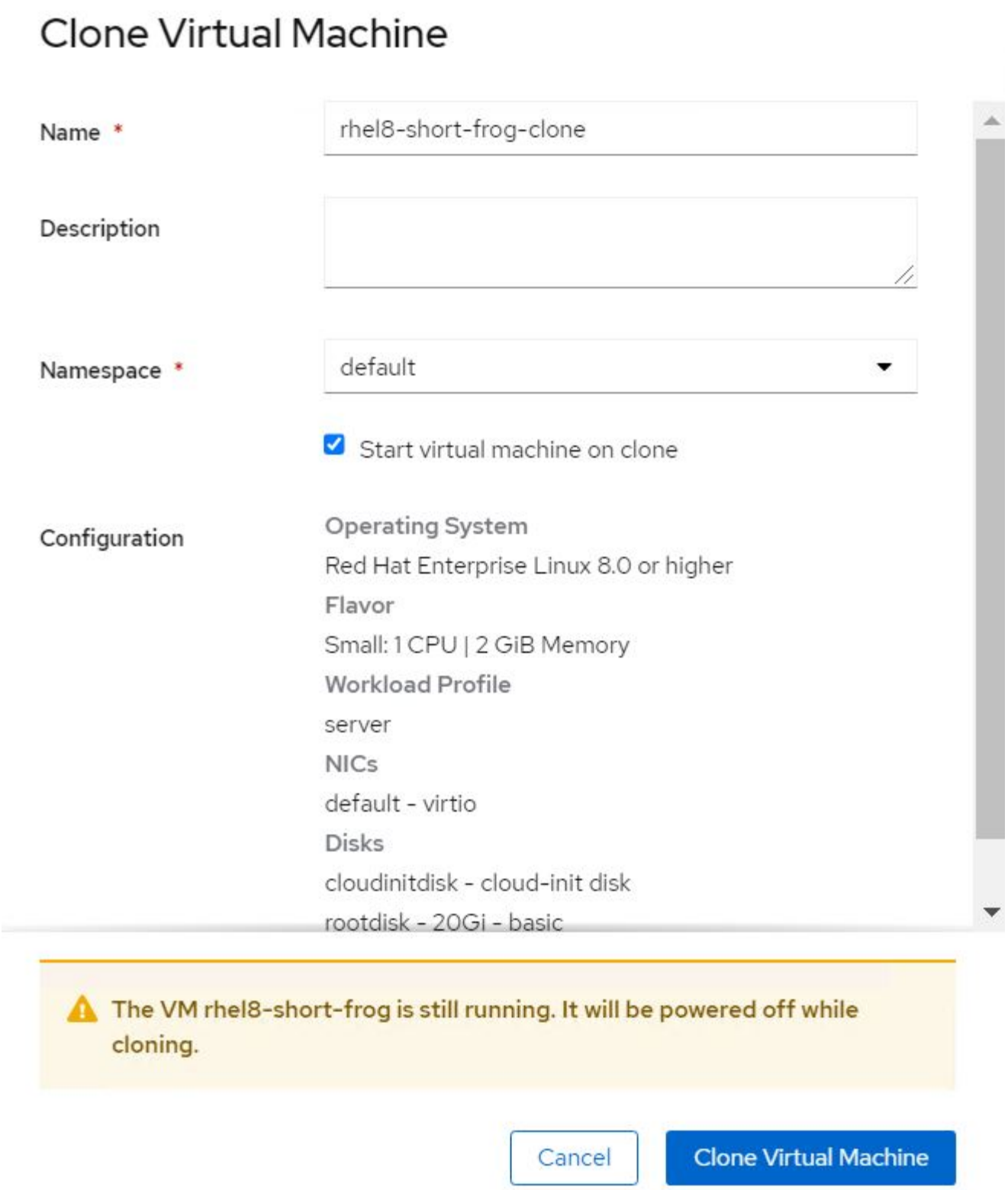
OpenShift叢集中的VM可透過兩種方式複製：

1. 關閉來源VM
2. 讓來源VM保持運作

關閉來源**VM**

關閉VM來複製現有VM是一項原生OpenShift功能、可透過Astra Trident的支援來實作。完成下列步驟以複製VM。

1. 瀏覽至「工作負載」>「虛擬化」>「虛擬機器」、然後按一下您要複製的虛擬機器旁的省略符號。
2. 按一下「Clone Virtual Machine（複製虛擬機器）」、並提供新VM的詳細資料。

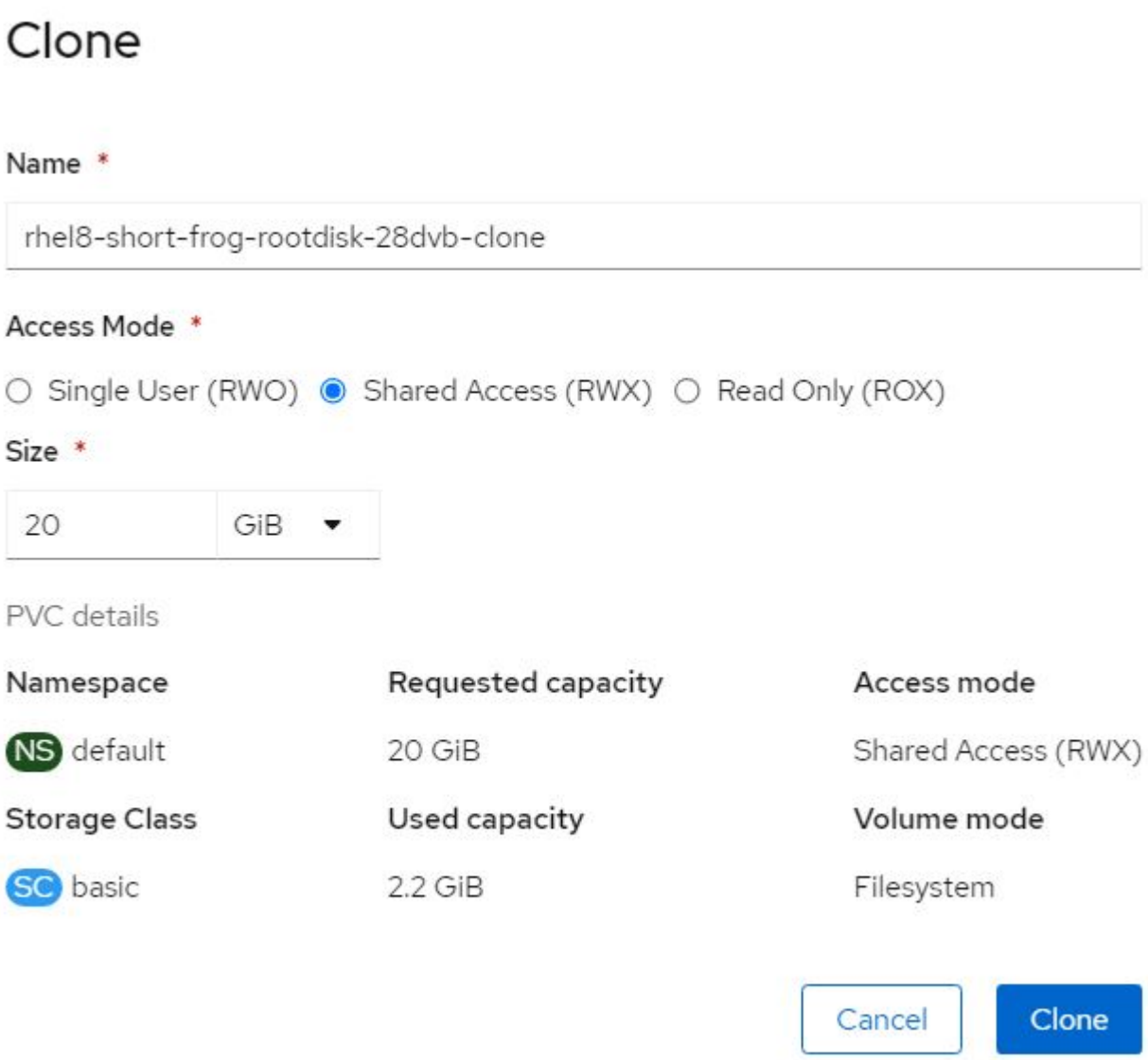


3. 按一下「Clone Virtual Machine（複製虛擬機器）」；這會關閉來源VM並開始建立複製VM。
4. 完成此步驟之後、您可以存取並驗證複製的VM內容。

讓來源**VM**保持運作

也可以複製現有VM、方法是複製來源VM的現有PVC,然後使用複製的PVC,建立新VM。此方法不需要關閉來源VM。完成下列步驟、即可在不關閉VM的情況下複製VM。

1. 瀏覽至「Storage（儲存設備）」>「PeristentVolume Claims（永久磁碟區宣告）」、然後按一下附加至來源VM的永久磁碟旁的省略號。
2. 按一下Clone PVc（複製PVc）、並提供新PVc的詳細資料。



3. 然後按一下Clone（複製）這會為新VM建立一個永久虛擬機器。
4. 瀏覽至「工作負載」>「虛擬化」>「虛擬機器」、然後按一下「建立」>「使用Yaml」
5. 在SPEC >範本> SPEC > Volume區段中、附加複製的PVc而非容器磁碟。請根據您的需求、提供新VM的所有其他詳細資料。

```
- name: rootdisk
  persistentVolumeClaim:
    claimName: rhel8-short-frog-rootdisk-28dvb-clone
```

6. 按一下「Create（建立）」以建立新的VM。
7. 成功建立VM之後、請存取並確認新VM是來源VM的複本。

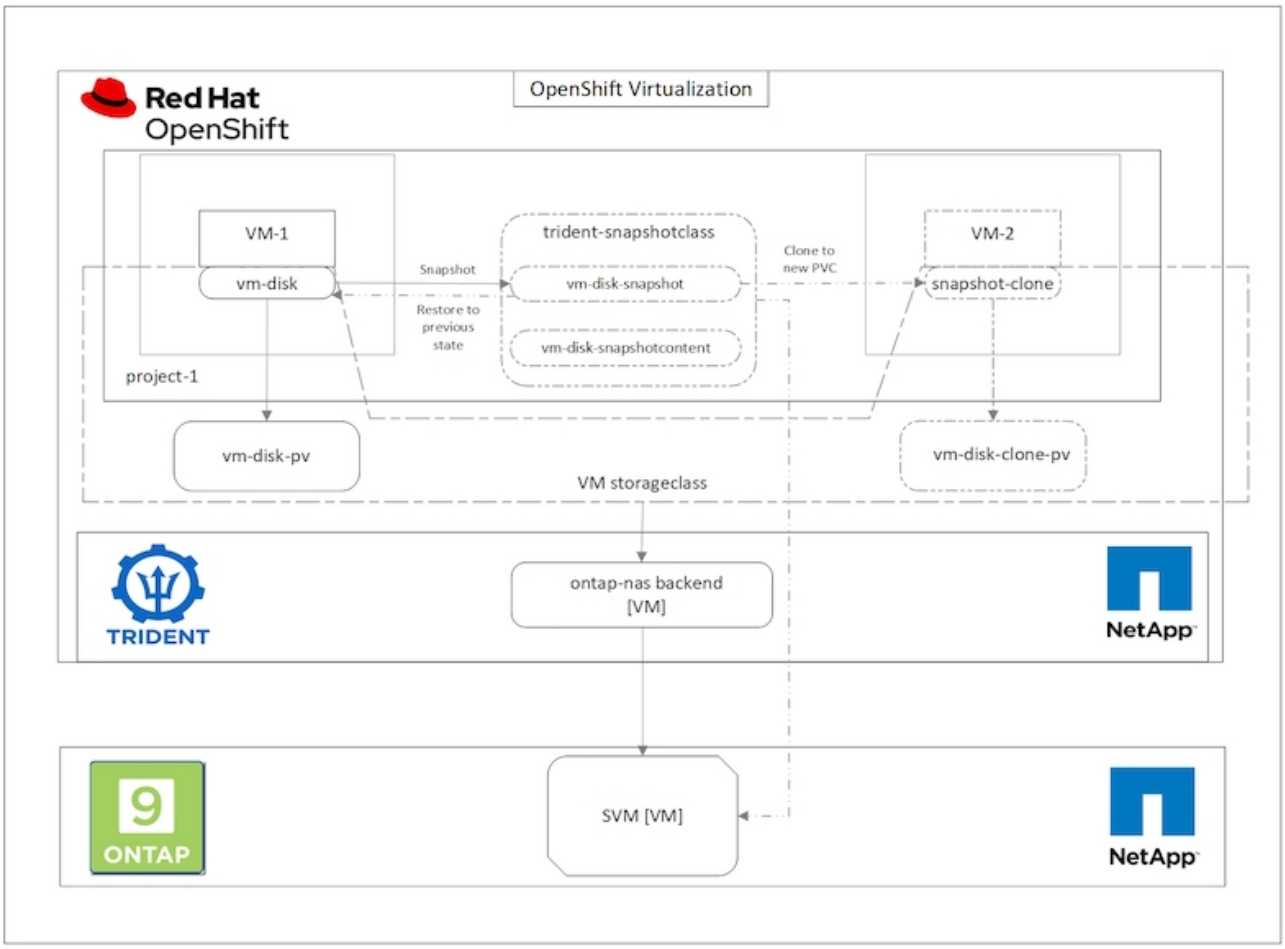
工作流程：**Red Hat OpenShift**虛擬化搭配**NetApp ONTAP** 功能

從**Snapshot**建立**VM**

有了Astra Trident和Red Hat OpenShift、使用者就能在IT資源配置的儲存類別上、對持續磁碟區進行快照。有了這項功能、使用者可以取得磁碟區的時間點複本、然後使用它來建立新的磁碟區、或將相同的磁碟區還原回先前的狀態。這可啟用或支援各種使用案例、從復原到複製到資料還原。

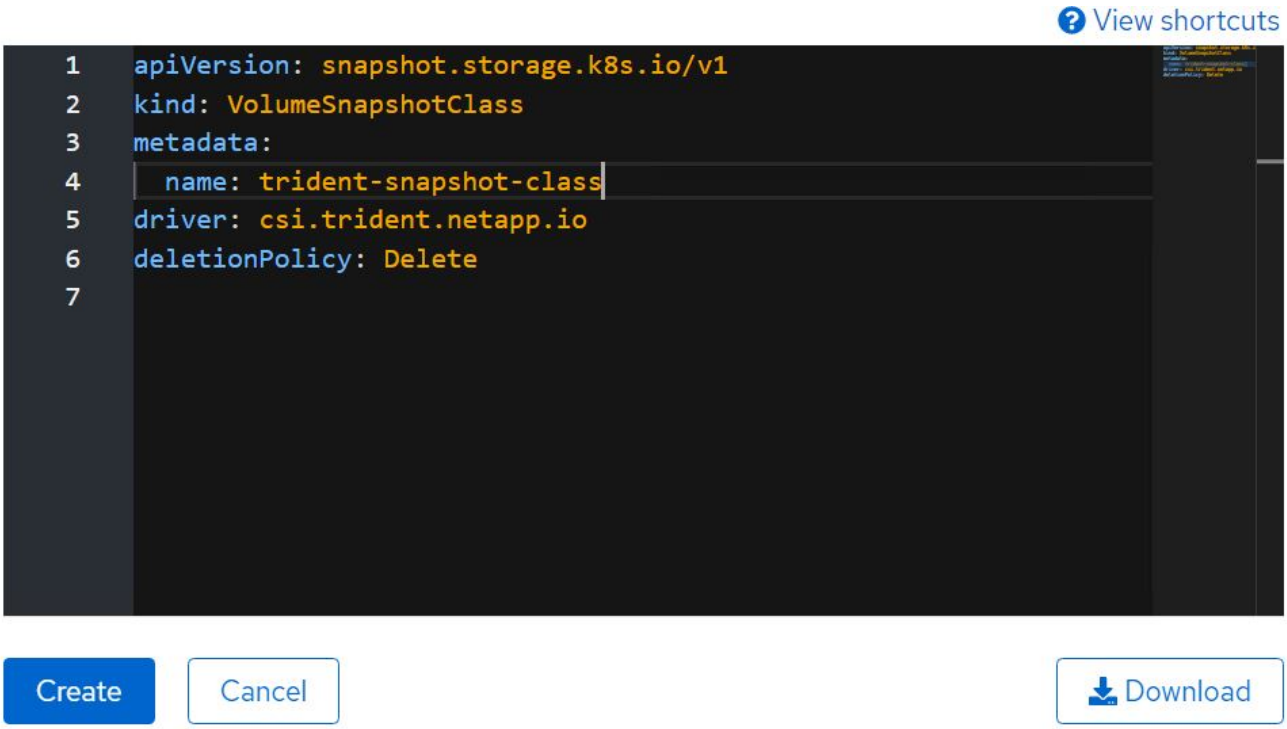
對於OpenShift中的Snapshot作業、必須定義Volume SnapshotClass、Volume Snapshot和Volume SnapshotContent等資源。

- Volume SnapshotContent是從叢集中的磁碟區擷取的實際快照。它是整個叢集的資源、類似於儲存的PersistentVolume。
- Volume Snapshot是建立Volume快照的要求。這類似於PeristentVolume Claim。
- Volume SnapshotClass可讓管理員為Volume Snapshot指定不同的屬性。它可讓您針對從相同磁碟區擷取的不同快照、擁有不同的屬性。

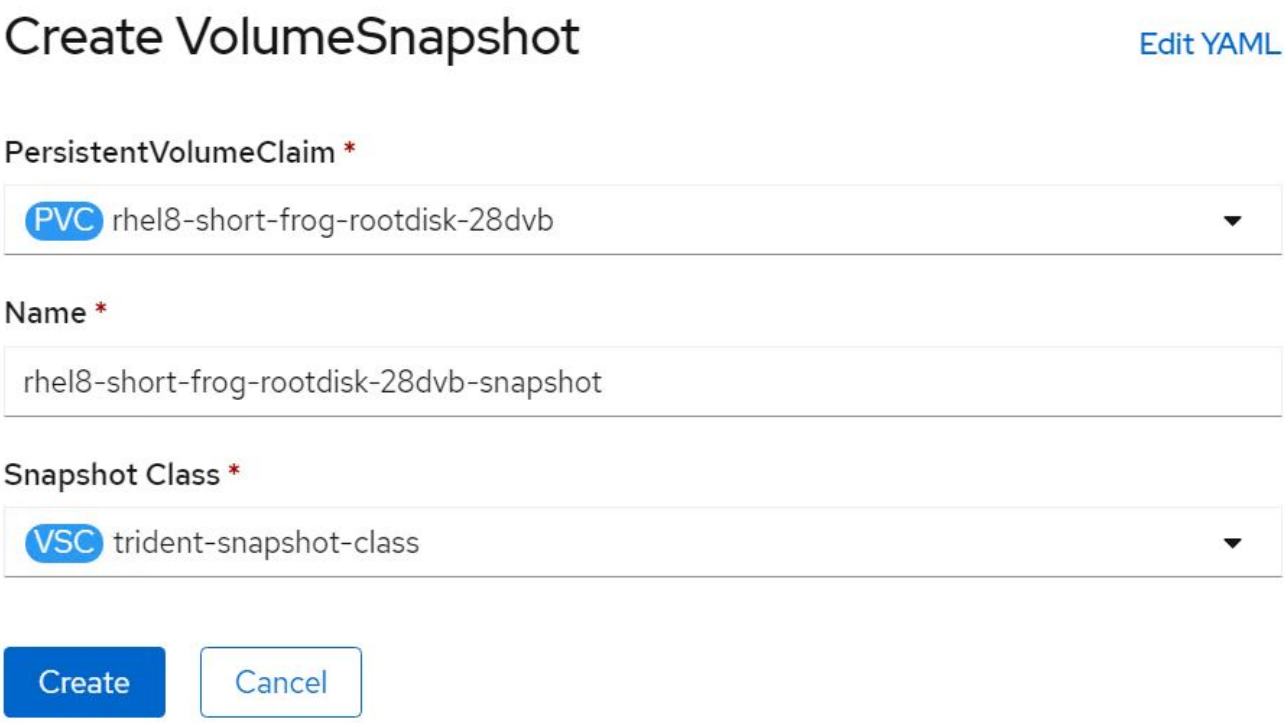


若要建立VM的Snapshot、請完成下列步驟：

1. 建立Volume SnapshotClass、然後使用該類別建立Volume Snapshot。瀏覽至「Storage（儲存設備）」>「Volume SnapshotClass（Volume SnapshotClass）」、然後按一下「Create Volume SnapshotClass
2. 輸入Snapshot Class的名稱、輸入驅動程式的csi.trident.netapp.io、然後按一下「Create（建立）」。



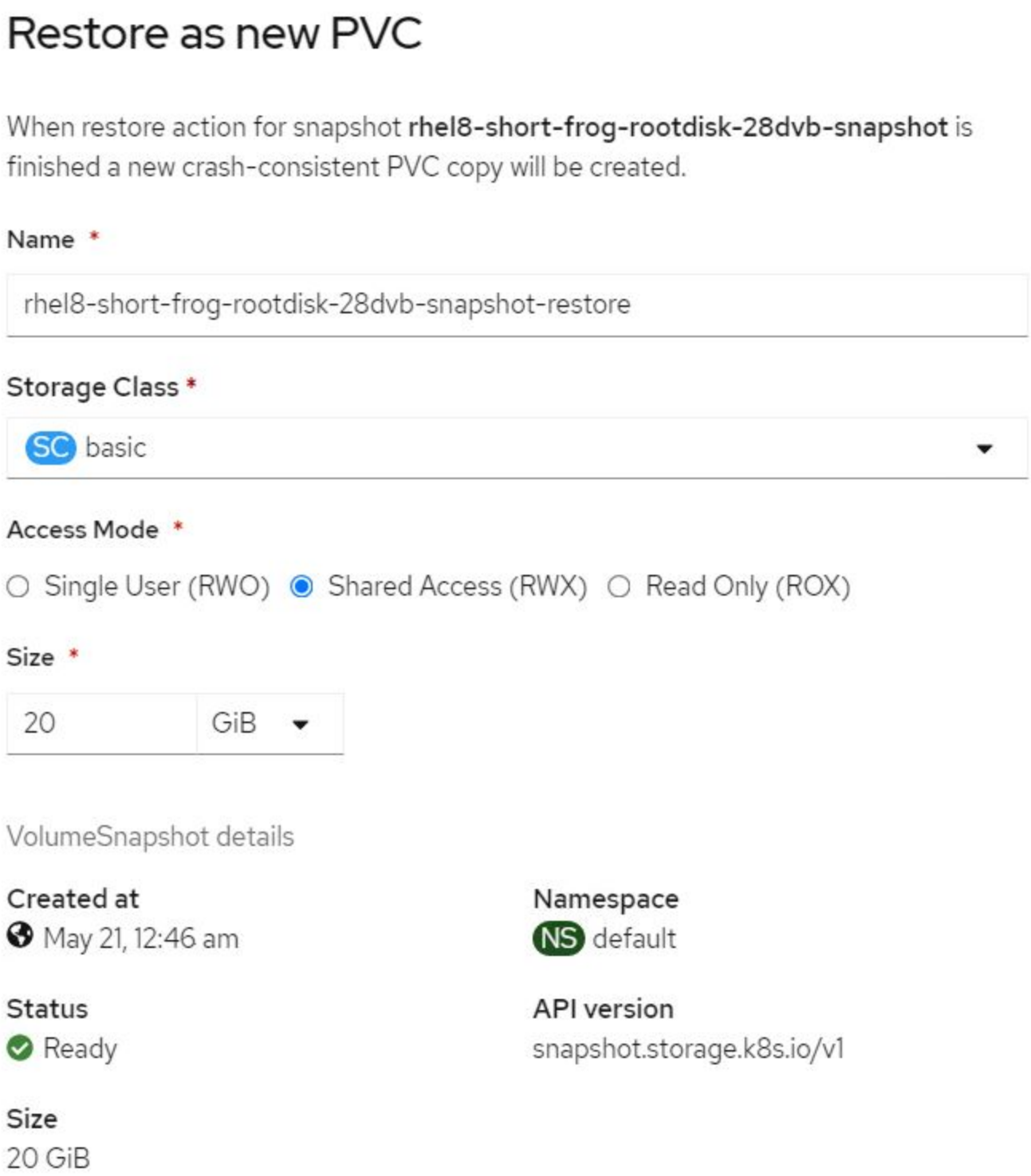
3. 識別附加至來源VM的PVc、然後建立該PVc的Snapshot。瀏覽至「儲存> Volume Snapshots」、然後按一下「Create Volume Snapshots（建立Volume Snapshot）」。
4. 選取您要建立Snapshot的永久虛擬磁碟、輸入Snapshot名稱或接受預設值、然後選取適當的Volume SnapshotClass。然後按一下「建立」。



5. 這會在該時間點建立永久虛擬資料快照。

從快照建立新的**VM**

1. 首先、將Snapshot還原成新的PVc。瀏覽至「Storage（儲存設備）」>「Volume Snapshots（Volume Snapshot）」、按一下您要還原的Snapshot旁邊的省略符號、然後按一下「Restore as new PVc（還原為新的PVc）」。
2. 輸入新的PVc詳細資料、然後按一下「還原」。這會產生新的PVC.



3. 接下來、從這個永久虛擬機器建立新的虛擬機器。瀏覽至「工作負載」>「虛擬化」>「虛擬機器」、然後按一下「建立」>「使用Yaml」

4. 在SPEC >範本> SPEC > Volume區段中、指定從Snapshot而非從Container磁碟建立的新永久虛擬磁碟。請根據您的需求、提供新VM的所有其他詳細資料。

```
- name: rootdisk
  persistentVolumeClaim:
    claimName: rhel8-short-frog-rootdisk-28dvb-snapshot-restore
```

5. 按一下「Create（建立）」以建立新的VM。
6. 成功建立虛擬機器之後、請存取並確認新虛擬機器的狀態與虛擬機器的狀態相同、而在建立快照時、虛擬機器的永久虛擬機器是用來建立快照的。

工作流程：**Red Hat OpenShift**虛擬化搭配**NetApp ONTAP** 功能

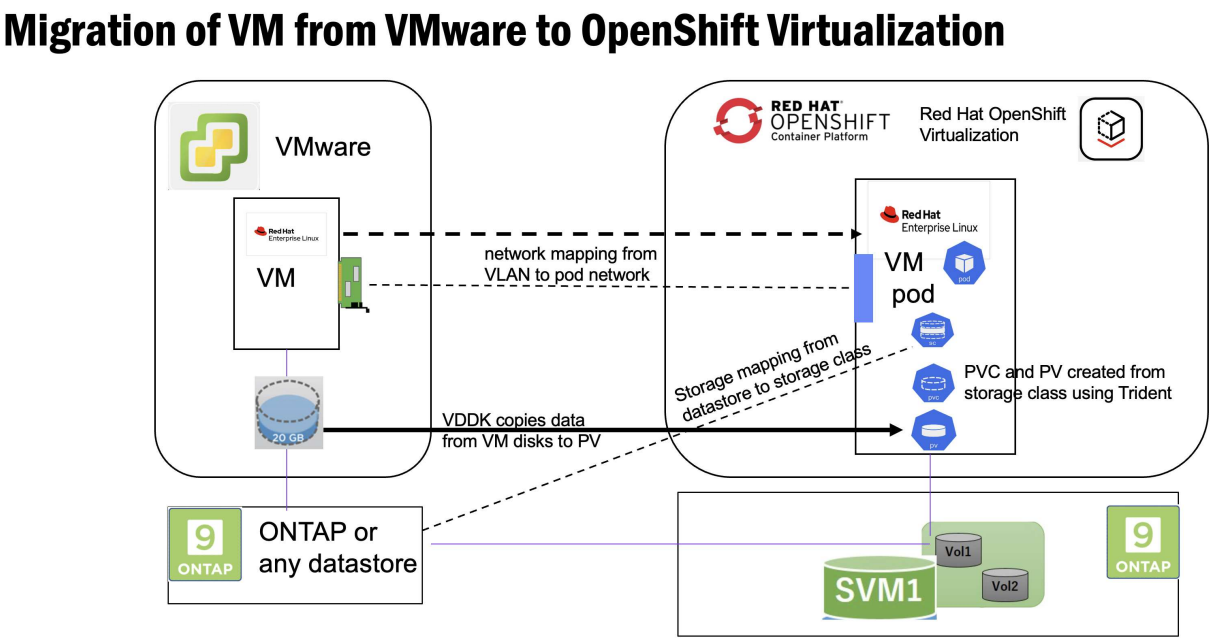
使用虛擬化移轉工具套件將 **VM** 從 **VMware** 移轉至 **OpenShift** 虛擬化

在本節中、我們將瞭解如何使用移轉工具套件來將虛擬機器從 VMware 移轉至 OpenShift Container 平台上執行的 OpenShift 虛擬化、並使用 Astra Trident 與 NetApp ONTAP 儲存設備整合。

以下影片展示 RHEL VM 從 VMware 移轉至 OpenShift 虛擬化的示範、使用 ONTAP SAN 進行持續儲存。

使用 Red Hat MTV 將 VM 移轉至使用 NetApp ONTAP 儲存設備的 OpenShift 虛擬化

下圖顯示虛擬機器從 VMware 移轉至 Red Hat OpenShift 虛擬化的高階檢視。



範例移轉的先決條件

關於 **VMware**

- 已安裝使用 RHEL 9.3 的 RHEL 9 VM 、並搭配下列組態：
  - CPU ： 2 、記憶體： 20 GB 、硬碟： 20 GB
  - 使用者認證： root 使用者和管理員使用者認證
- VM 準備就緒後、即安裝 PostgreSQL 伺服器。
  - PostgreSQL 伺服器已啟動、並可在開機時啟動

```
systemctl start postgresql.service`
systemctl enable postgresql.service
The above command ensures that the server can start in the VM in
OpenShift Virtualization after migration
```

  - 新增 2 個資料庫、 1 個資料表和 1 個資料列。請參閱 "請按這裡" 如需在 RHEL 上安裝 PostgreSQL 伺服器及建立資料庫和表格項目的指示。

> 請確定您啟動 PostgreSQL 伺服器、並讓服務在開機時啟動。

在 **OpenShift** 叢集上

下列安裝已在安裝 MTV 之前完成：

- OpenShift 叢集 4.13.34
- "Astra Trident 23.10"
- 叢集節點上啟用 iSCSI 的多重路徑（適用於 ONTAP - SAN 儲存類別）。請參閱提供的 yaml 、以建立在叢集中每個節點上啟用 iSCSI 的精靈集。
- Trident 後端和儲存類別、適用於使用 iSCSI 的 ONTAP SAN 。請參閱提供的 yaml 檔案、瞭解 Trident 後端和儲存類別。
- "OpenShift虛擬化"

若要在 OpenShift 叢集節點上安裝 iSCSI 和多重路徑、請使用以下提供的 yaml 檔案
準備 **iSCSI** 的叢集節點

```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  namespace: trident
  name: trident-iscsi-init
  labels:
    name: trident-iscsi-init
spec:
  selector:
    matchLabels:
      name: trident-iscsi-init
```

```
  template:
    metadata:
      labels:
        name: trident-iscsi-init
    spec:
      hostNetwork: true
      serviceAccount: trident-node-linux
      initContainers:
      - name: init-node
        command:
          - nsenter
          - --mount=/proc/1/ns/mnt
          - --
          - sh
          - -c
        args: ["$(STARTUP_SCRIPT)"]
        image: alpine:3.7
        env:
        - name: STARTUP_SCRIPT
          value: |
            #! /bin/bash
            sudo yum install -y lsscsi iscsi-initiator-utils sg3_utils
device-mapper-multipath
            rpm -q iscsi-initiator-utils
            sudo sed -i 's/^\(node.session.scan\).*/\1 = manual/'
/etc/iscsi/iscsid.conf
            cat /etc/iscsi/initiatorname.iscsi
            sudo mpathconf --enable --with_multipathd y --find_multipaths
n
            sudo systemctl enable --now iscsid multipathd
            sudo systemctl enable --now iscsi
        securityContext:
          privileged: true
      hostPID: true
      containers:
      - name: wait
        image: k8s.gcr.io/pause:3.1
      hostPID: true
      hostNetwork: true
      tolerations:
      - effect: NoSchedule
        key: node-role.kubernetes.io/master
  updateStrategy:
    type: RollingUpdate
```

使用下列 yaml 檔案建立 Trident 後端組態、以使用 ONTAP SAN 儲存設備

iSCSI 的 Trident 後端

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-san-secret
type: Opaque
stringData:
  username: <username>
  password: <password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: ontap-san
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <management LIF>
  backendName: ontap-san
  svm: <SVM name>
  credentials:
    name: backend-tbc-ontap-san-secret
```

使用下列 yaml 檔案建立 Trident 儲存類別組態、以使用 ONTAP SAN 儲存設備
iSCSI 的 Trident 儲存等級

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
allowVolumeExpansion: true
```

*** 安裝 MTV***

現在您可以安裝移轉工具套件（虛擬化）（MTV）。請參閱所提供的指示 "請按這裡" 取得安裝的說明。

移轉工具套件虛擬化（MTV）使用者介面已整合至 OpenShift 網路主控台。
您可以參閱 "請按這裡" 開始使用使用者介面執行各種工作。

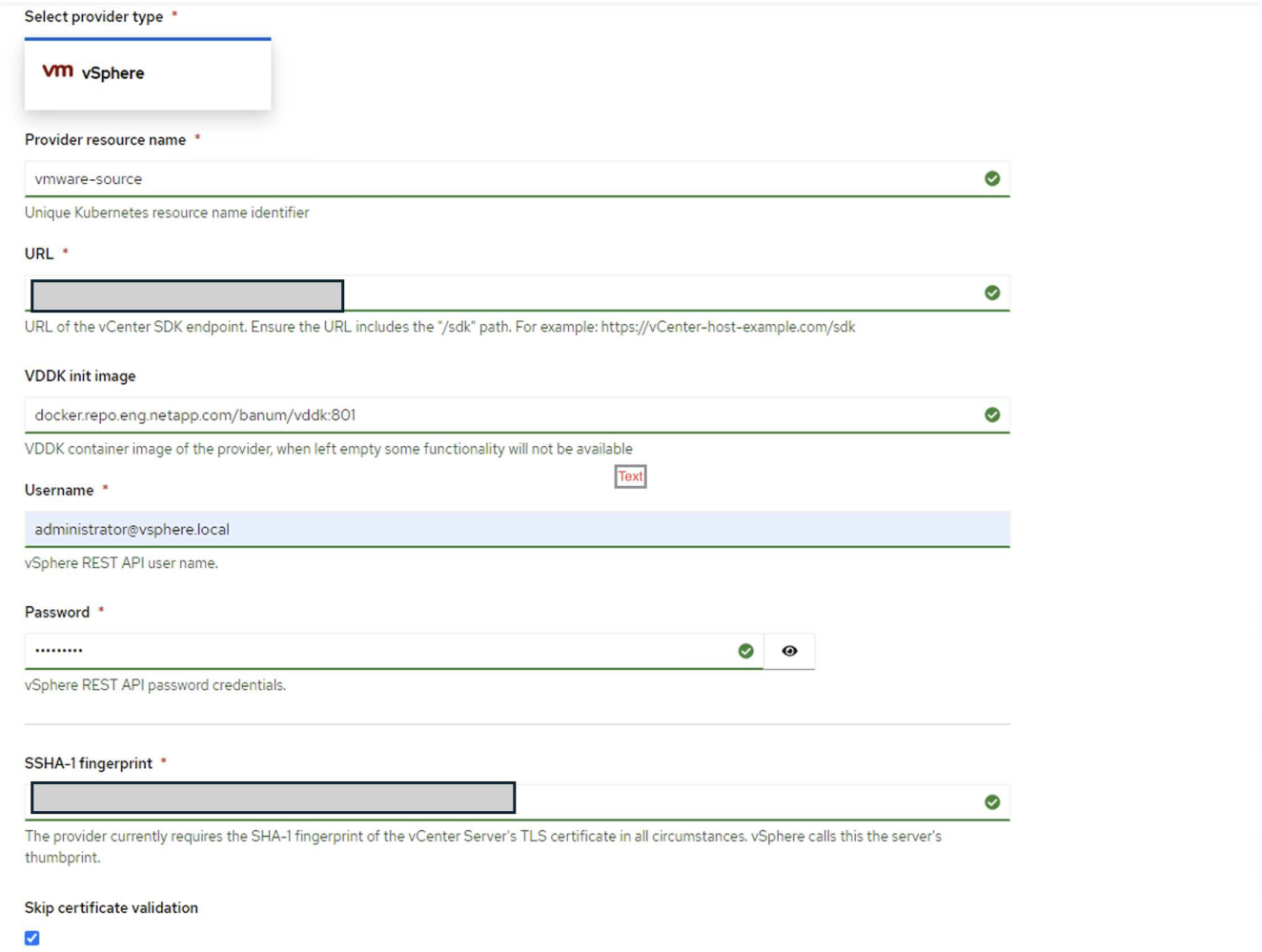
- 建立來源供應商 **

為了將 RHEL VM 從 VMware 移轉至 OpenShift 虛擬化、您必須先建立 VMware 的來源供應商。請參閱說明 "請按這裡" 以建立來源供應商。

您需要下列項目來建立 VMware 來源供應商：

- vCenter URL
- vCenter 認證
- vCenter 伺服器指紋
- 儲存庫中的 VDDK 映像

建立範例來源供應商：



> 虛擬化移轉工具套件（MTV）使用 VMware 虛擬磁碟開發套件（VDDK） SDK 來加速從 VMware vSphere 傳輸虛擬磁碟。因此、強烈建議您建立 VDDK 映像（雖然是選用的）。若要使用此功能、請下載 VMware 虛擬磁碟開發套件（VDDK）、建置 VDDK 映像、然後將 VDDK 映像推入映像登錄。

請遵循所提供的指示 "請按這裡" 建立 VDDK 映像、並將其推送至可從 OpenShift 叢集存取的登錄。

- 建立目的地供應商 **

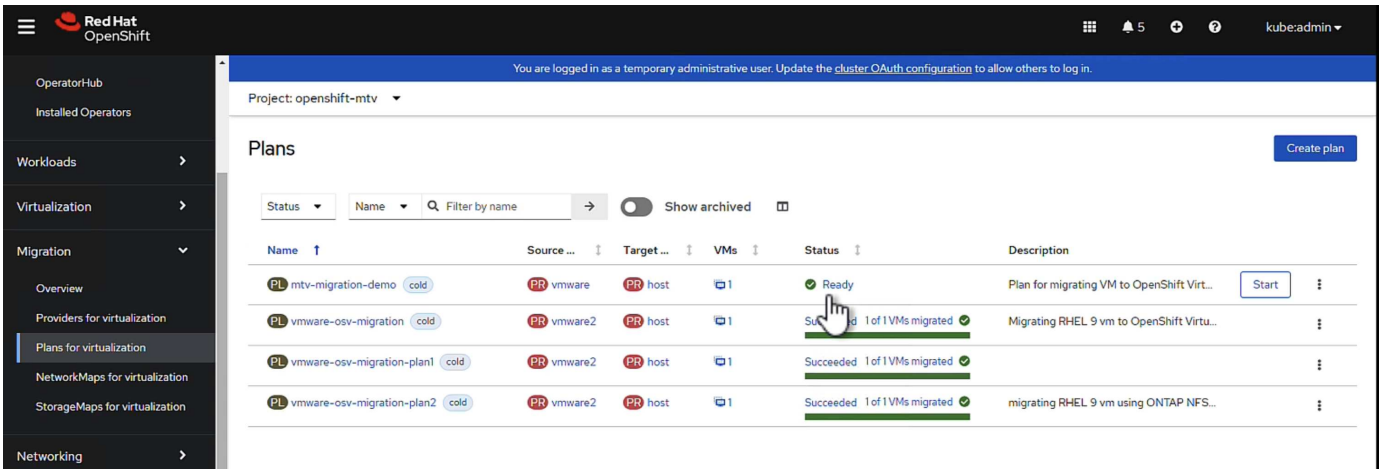
當 OpenShift 虛擬化供應商是來源供應商時、主機叢集會自動新增。

- 建立移轉計畫 **

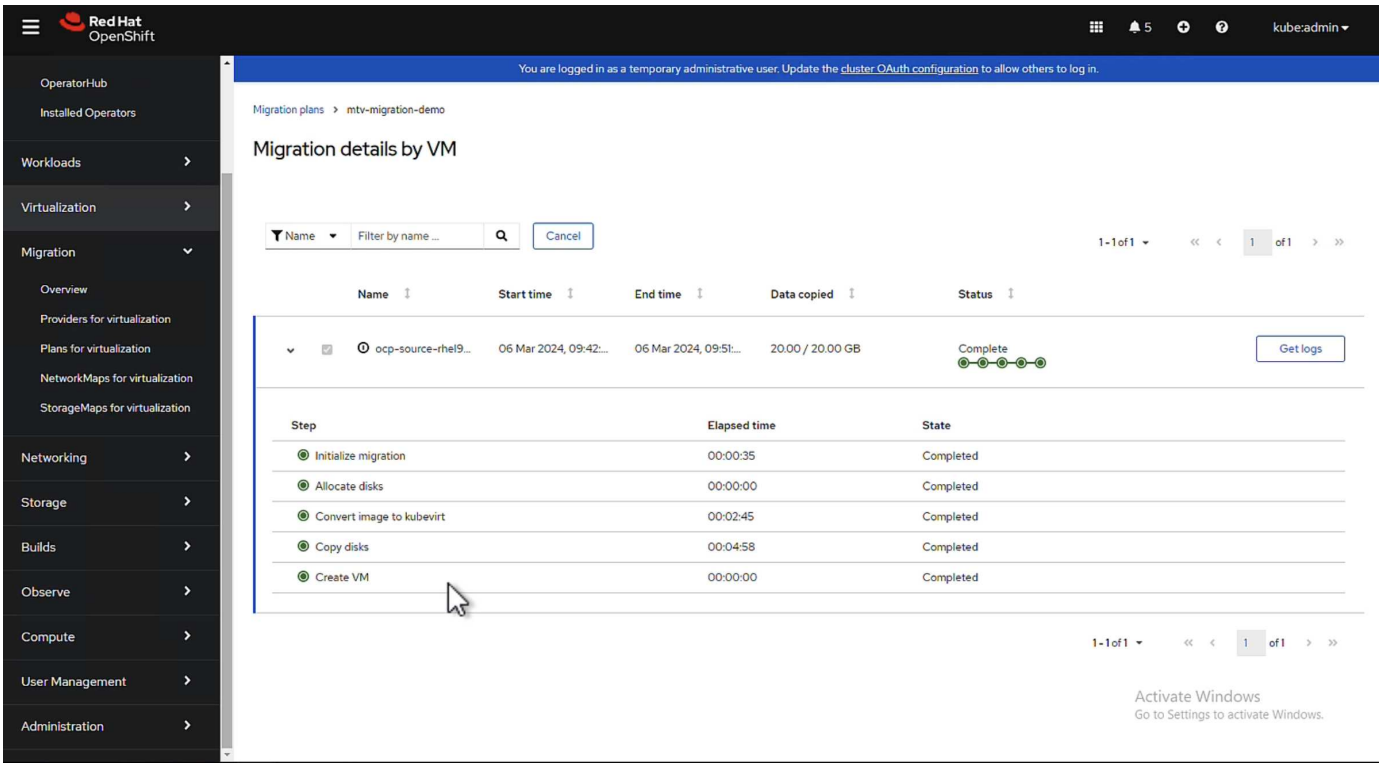
請遵循所提供的指示 "請按這裡" 以建立移轉計畫。

建立計畫時、如果尚未建立、則需要建立下列項目：

- 用於將來源網路對應至目標網路的網路對應。
- 將來源資料存放區對應至目標儲存類別的儲存對應。您可以選擇 ONTAP SAN 儲存類別。
一旦建立移轉計畫、計畫的狀態應該會顯示 * 就緒 * 、您現在應該可以 * 開始 * 計畫。



按一下 * 「開始」 * 將會執行一系列步驟、以完成虛擬機器的移轉。



完成所有步驟後、您可以按一下左側導覽功能表 * 虛擬化 * 下的 * 虛擬機器 * 來查看移轉的虛擬機器。提供存取虛擬機器的指示 "請按這裡"。

您可以登入虛擬機器並驗證 posgresql 資料庫的內容。資料表中的資料庫、資料表和項目應與在來源 VM 上建立的項目相同。

採用NetApp的Red Hat OpenShift上的Kubernetes進階叢集管理

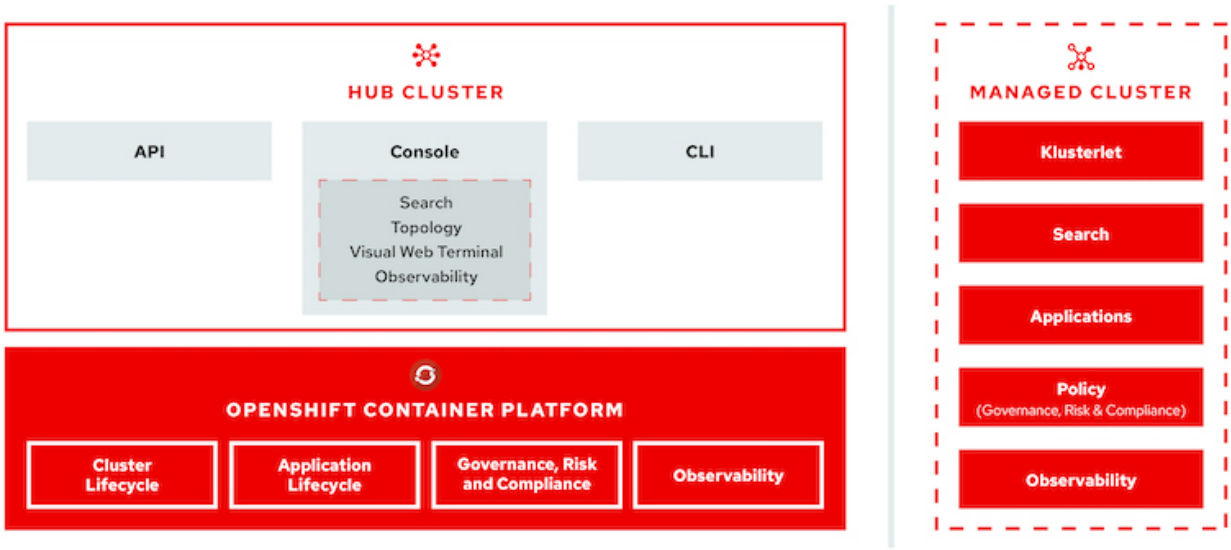
適用於**Kubernetes**的進階叢集管理：採用**NetApp**的**Red Hat OpenShift**

隨著容器化應用程式從開發移轉至正式作業、許多組織需要多個Red Hat OpenShift叢集來支援該應用程式的測試與部署。因此、組織通常會在OpenShift叢集上裝載多個應用程式或工作負載。因此、每個組織最終都必須管理一組叢集、因此OpenShift系統管理員必須面對新增的挑戰、即在橫跨多個內部部署資料中心和公有雲的各種環境中、管理及維護多個叢集。為了因應這些挑戰、Red Hat推出適用於Kubernetes的進階叢集管理。

適用於Kubernetes的Red Hat Advanced Cluster Management可讓您執行下列工作：

1. 跨資料中心和公有雲建立、匯入及管理多個叢集
2. 從單一主控台部署及管理多個叢集上的應用程式或工作負載
3. 監控及分析不同叢集資源的健全狀況與狀態
4. 監控並強制執行多個叢集的安全法規遵循

Red Hat Advanced Cluster Management for Kubernetes是以附加元件安裝至Red Hat OpenShift叢集的方式、它會將此叢集當作中央控制器來執行所有作業。此叢集稱為集線器叢集、會公開使用者連線至「進階叢集管理」的管理層面。透過進階叢集管理主控台匯入或建立的所有其他OpenShift叢集、均由中樞叢集管理、稱為託管叢集。它會在託管叢集上安裝名為Klusterlet的代理程式、將其連線至集線器叢集、並針對與叢集生命週期管理、應用程式生命週期管理、觀察性及安全性法規遵循相關的各種活動、提供服務要求。



如需詳細資訊、請參閱文件 "請按這裡"。

部署

為Kubernetes部署進階叢集管理

先決條件

1. 用於顯示中樞叢集的Red Hat OpenShift叢集（高於4.5版）
2. 適用於託管叢集的Red Hat OpenShift叢集（高於4.4.3版）
3. 叢集管理存取Red Hat OpenShift叢集
4. 適用於Kubernetes的進階叢集管理Red Hat訂購

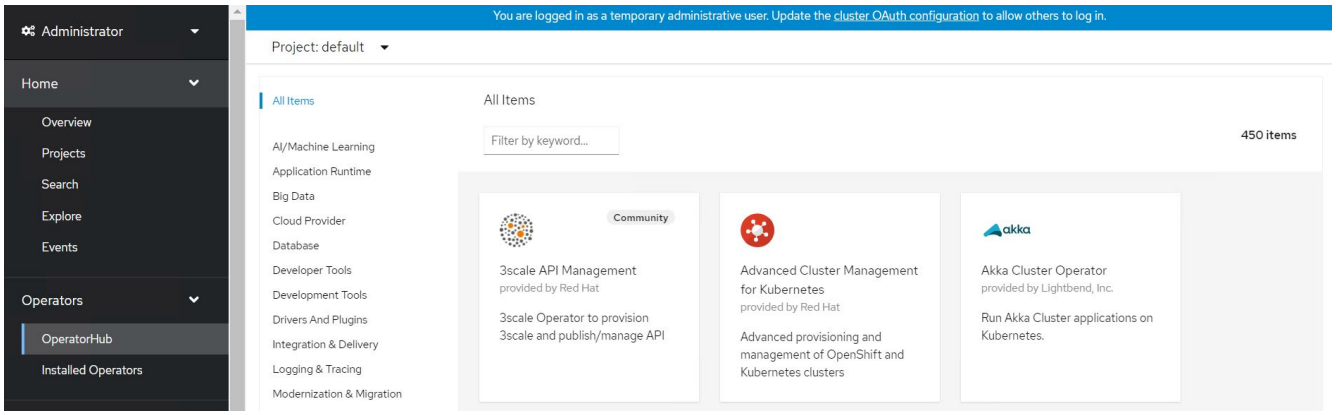
進階叢集管理是OpenShift叢集的附加元件、因此根據整個集線器和託管叢集所使用的功能、對硬體資源有特定的需求和限制。在調整叢集規模時、您必須將這些問題納入考量。請參閱文件 "請按這裡" 以取得更多詳細資料。

或者、如果集線器叢集有專屬節點來裝載基礎架構元件、而且您只想在這些節點上安裝「進階叢集管理」資源、則必須相應地將容許值和選取器新增至這些節點。如需詳細資料、請參閱文件 "請按這裡"。

為Kubernetes部署進階叢集管理

若要在OpenShift叢集上安裝適用於Kubernetes的進階叢集管理、請完成下列步驟：

1. 選擇OpenShift叢集做為中樞叢集、並以叢集管理權限登入。
2. 瀏覽至「運算子」>「運算子中樞」、然後搜尋Kubernetes的「進階叢集管理」。



3. 選取適用於Kubernetes的進階叢集管理、然後按一下安裝。

4. 在Install Operator（安裝操作員）畫面上、提供必要的詳細資料（NetApp建議保留預設參數）、然後按一下Install（安裝）。

OperatorHub > Operator Installation

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel *

- ○ release-2.0
- ○ release-2.1
- ◉ release-2.2

Installation mode *

- ○ All namespaces on the cluster (default)
  This mode is not supported by this Operator
- ◉ A specific namespace on the cluster
  Operator will be available in a single Namespace only.

Installed Namespace *

- ◉ Operator recommended Namespace: PR open-cluster-management

Namespace creation
Namespace **open-cluster-management** does not exist and will be created.

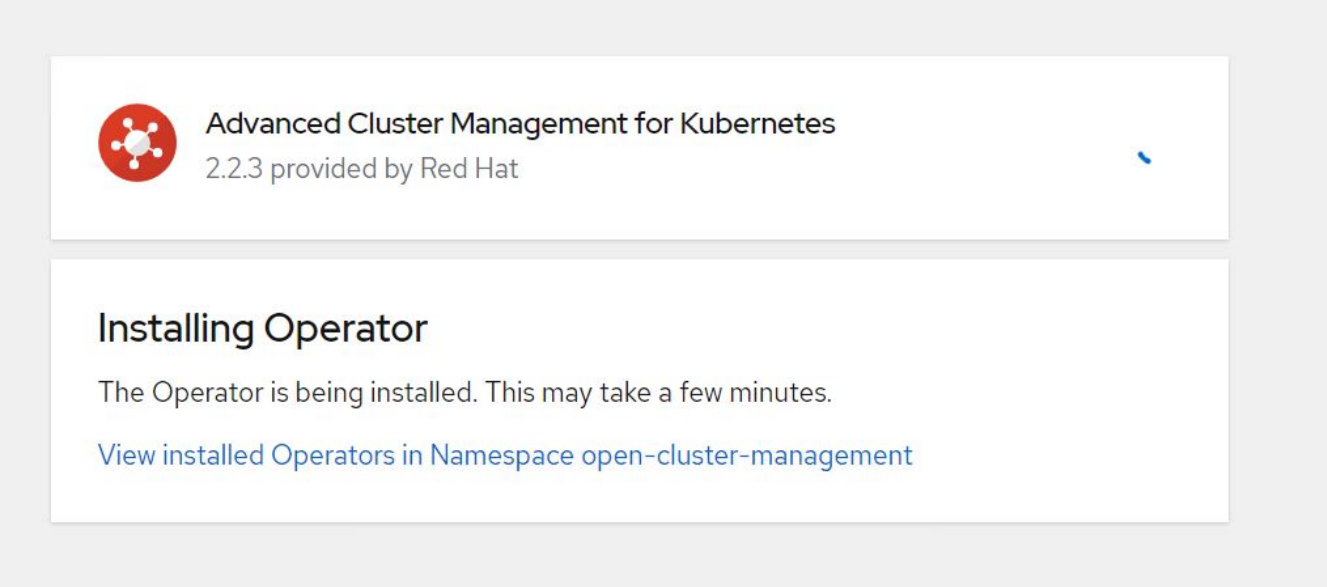
- ○ Select a Namespace

Approval strategy *

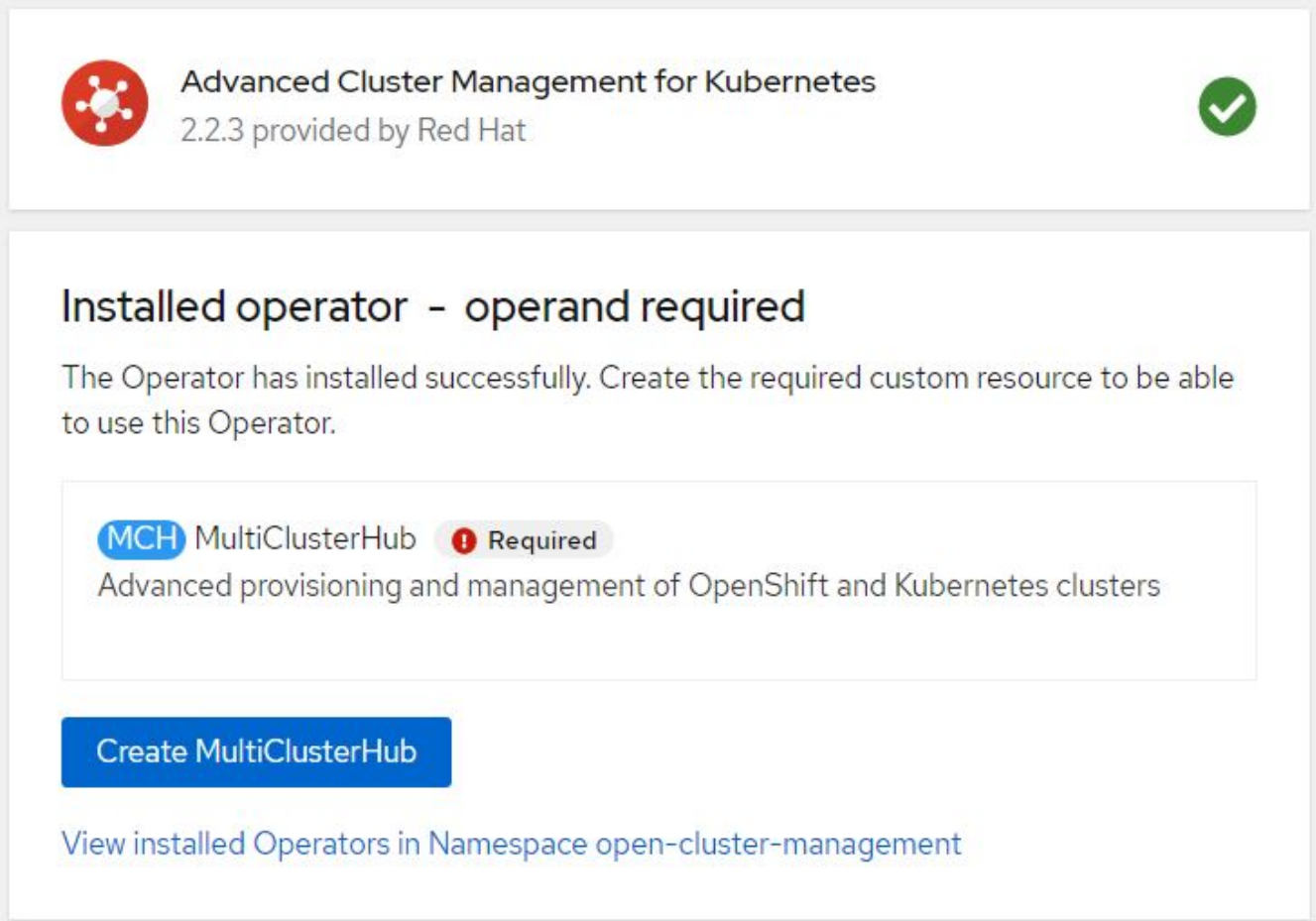
- ◉ Automatic
- ○ Manual

Install Cancel

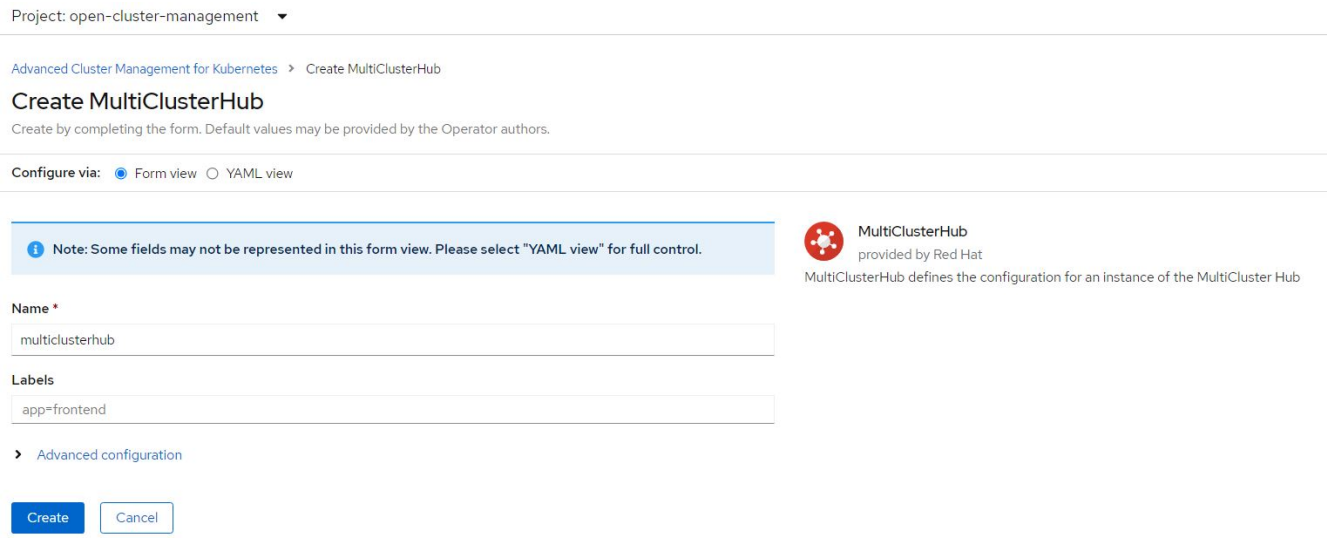
5. 等待操作員安裝完成。



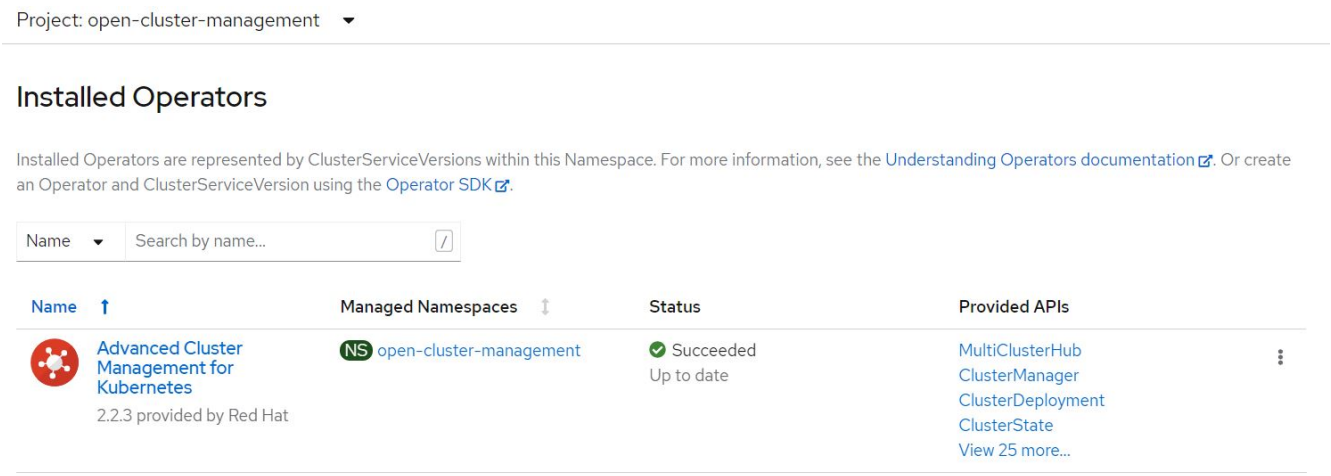
6. 安裝操作員之後、按一下「Create MultiClusterHub（建立MultiClusterHub）」。



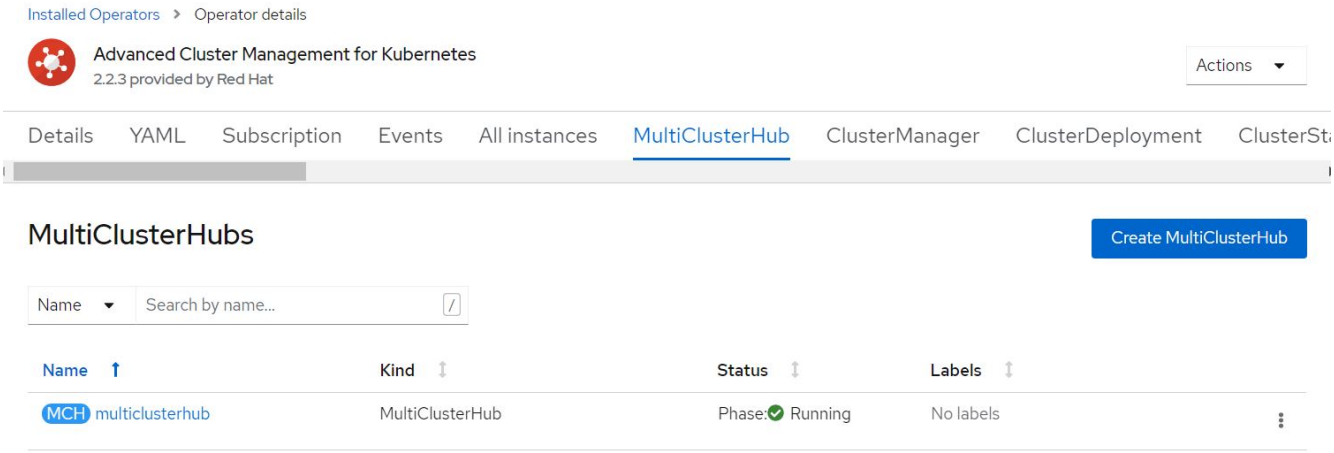
7. 在Create MultiClusterHub（建立MultiClusterHub）畫面上、按一下「建立」（在提供詳細資料之後）。這會啟動多叢集集集集線器的安裝。



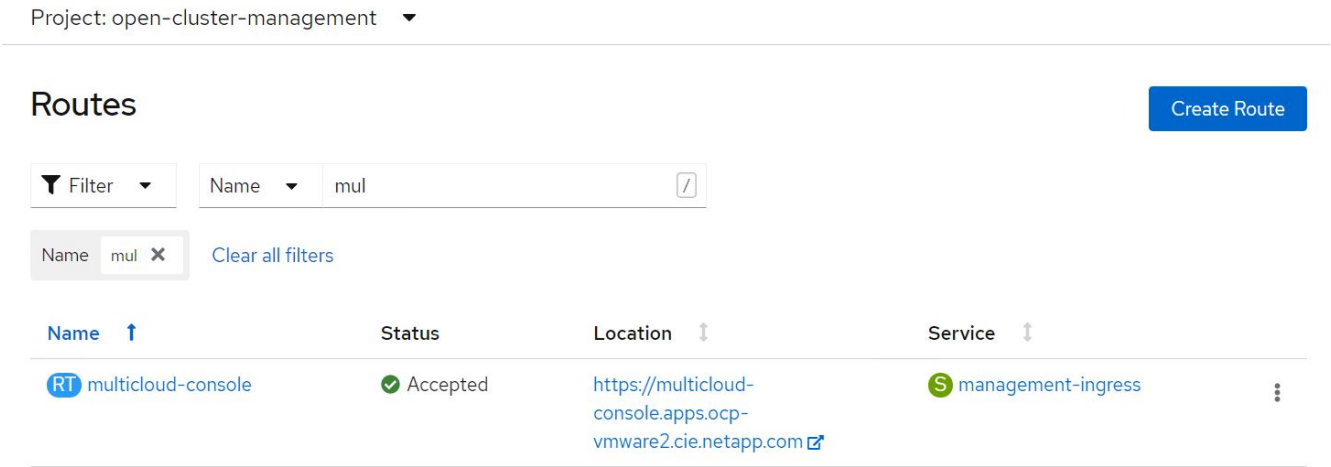
8. 在所有Pod移至開放式叢集管理命名空間中的執行狀態、且操作員移至成功狀態之後、就會安裝適用於Kubernetes的進階叢集管理。



9. 完成集線器安裝需要一些時間、完成之後、MultiCluster集線器會移至執行中狀態。



10. 它會在開放式叢集管理命名空間中建立路由。連線至路由中的URL、以存取進階叢集管理主控台。



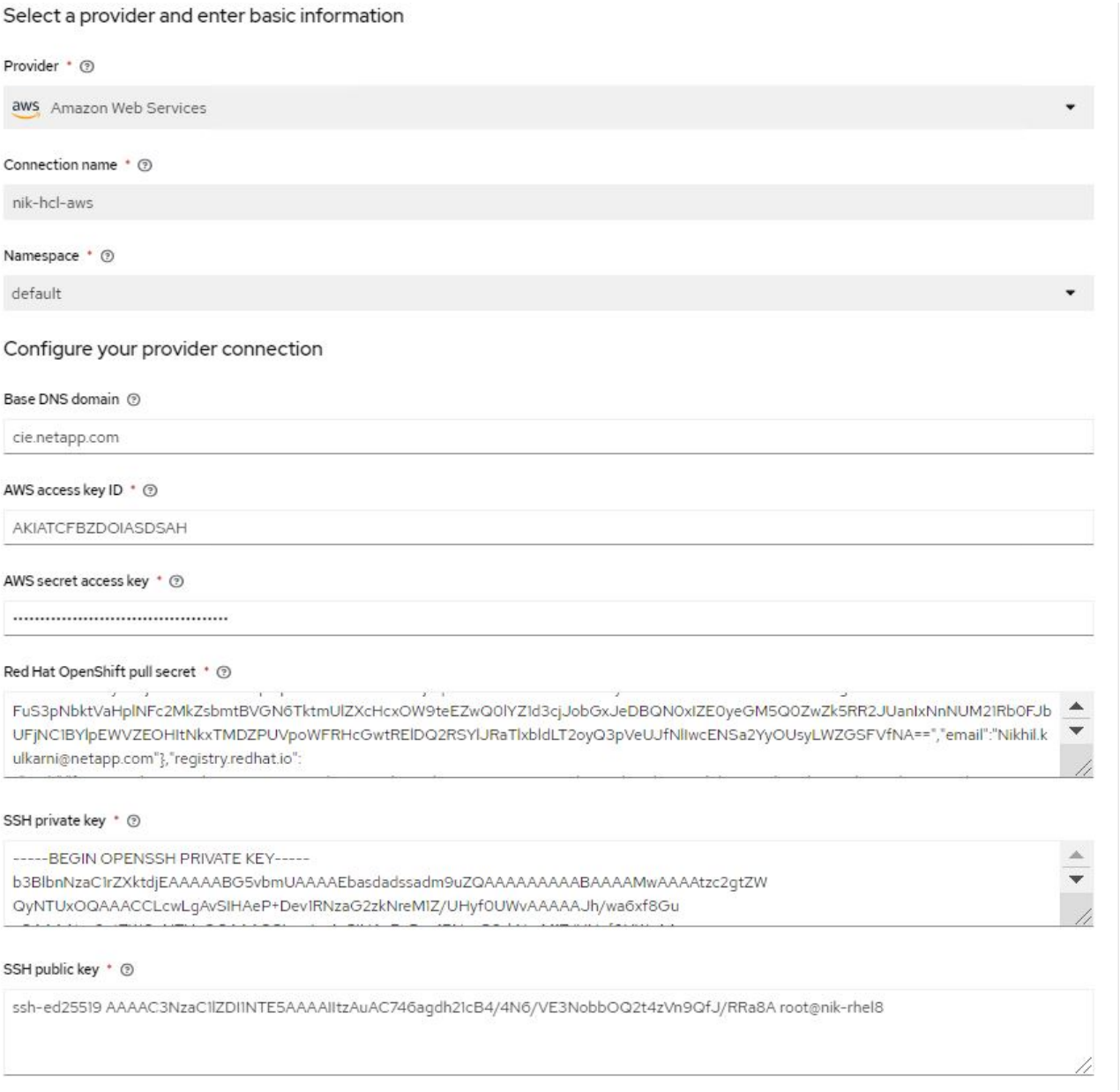
功能

特色：採用NetApp的Red Hat OpenShift上的Kubernetes進階叢集管理

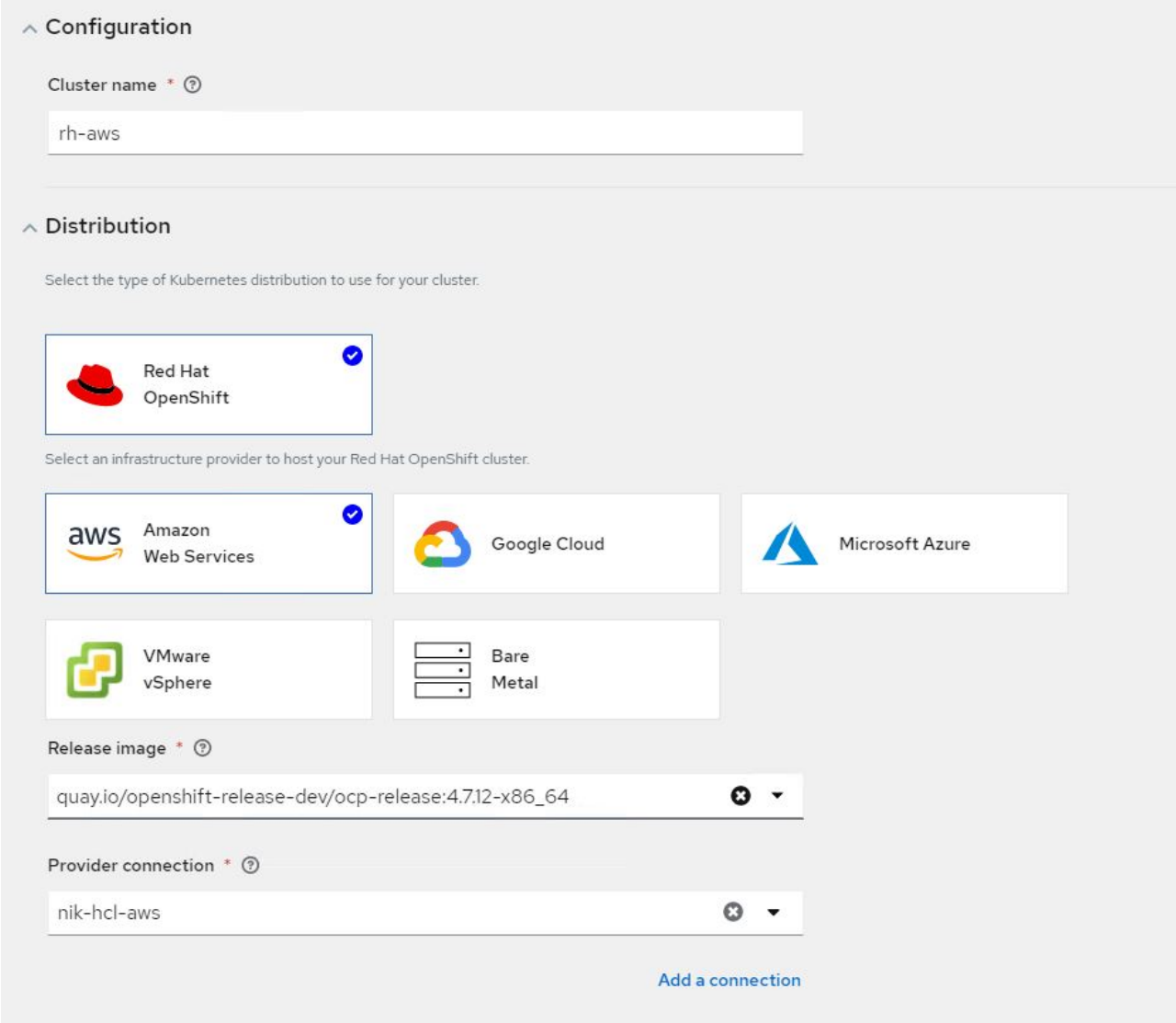
叢集生命週期管理

若要管理不同的OpenShift叢集、您可以建立或匯入至「進階叢集管理」。

1. 首先瀏覽至自動化基礎架構>叢集。
2. 若要建立新的OpenShift叢集、請完成下列步驟：
   a. 建立供應商連線：瀏覽至「提供者連線」、然後按一下「新增連線」、提供與所選提供者類型對應的所有詳細資料、然後按一下「新增」。



   b. 若要建立新叢集、請瀏覽至「叢集」、然後按一下「新增叢集」>「建立叢集」。提供叢集和對應供應商的詳細資料、然後按一下「Create（建立）」。



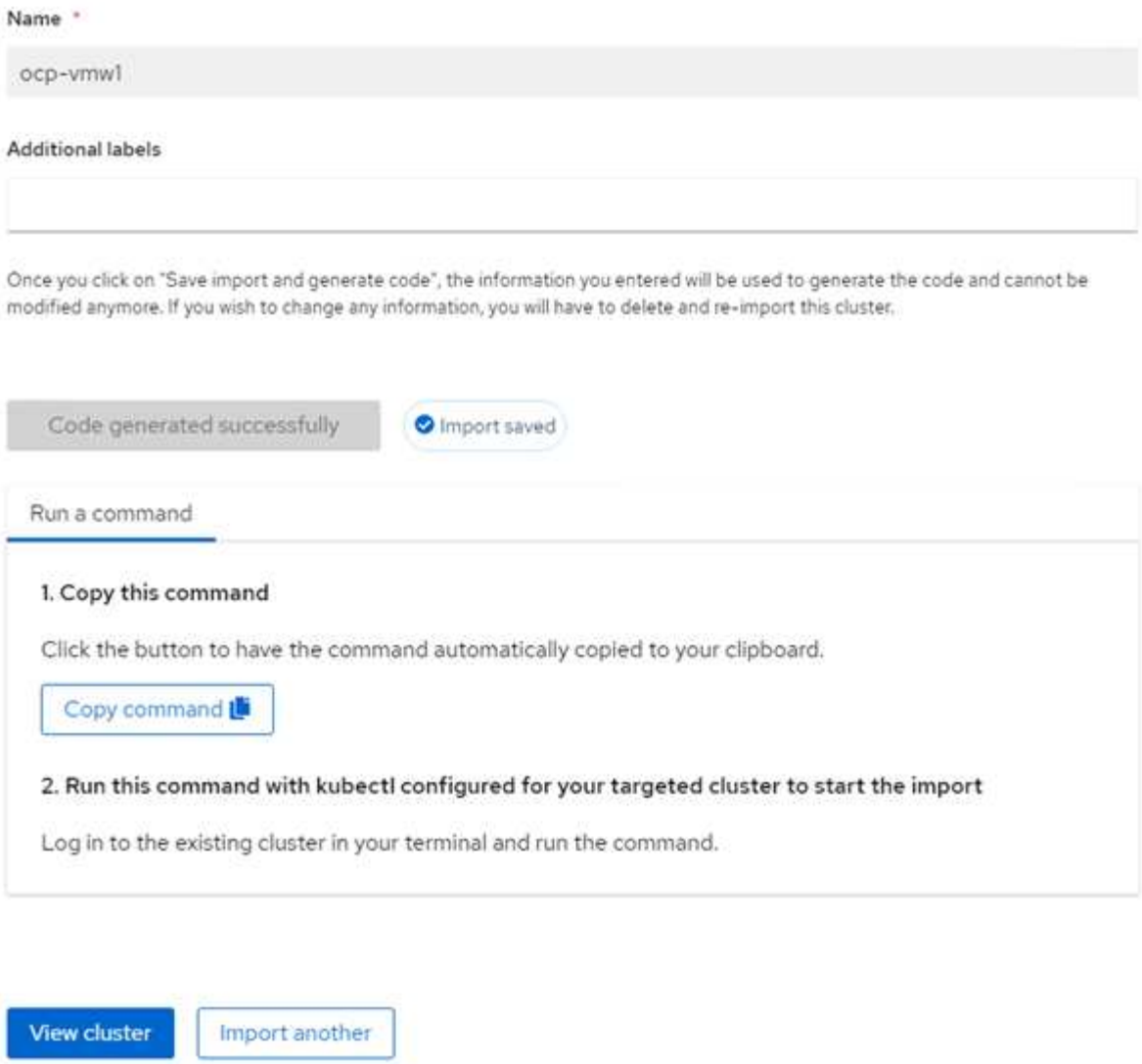
c. 建立叢集之後、它會顯示在叢集清單中、並顯示「Ready（就緒）」狀態。

3. 若要匯入現有的叢集、請完成下列步驟：

   a. 瀏覽至「叢集」、然後按一下「新增叢集」>「匯入現有的叢集」。

   b. 輸入叢集名稱、然後按一下「Save Import（儲存匯入）」和「Generate Code（產生程式碼）」。此時會顯示新增現有叢集的命令。

   c. 按一下「Copy Command（複製命令）」、然後在要新增至集線器叢集的叢集上執行命令。這會在叢集上啟動必要代理程式的安裝、完成此程序之後、叢集會顯示在叢集清單中、並顯示「Ready（就緒）」狀態。



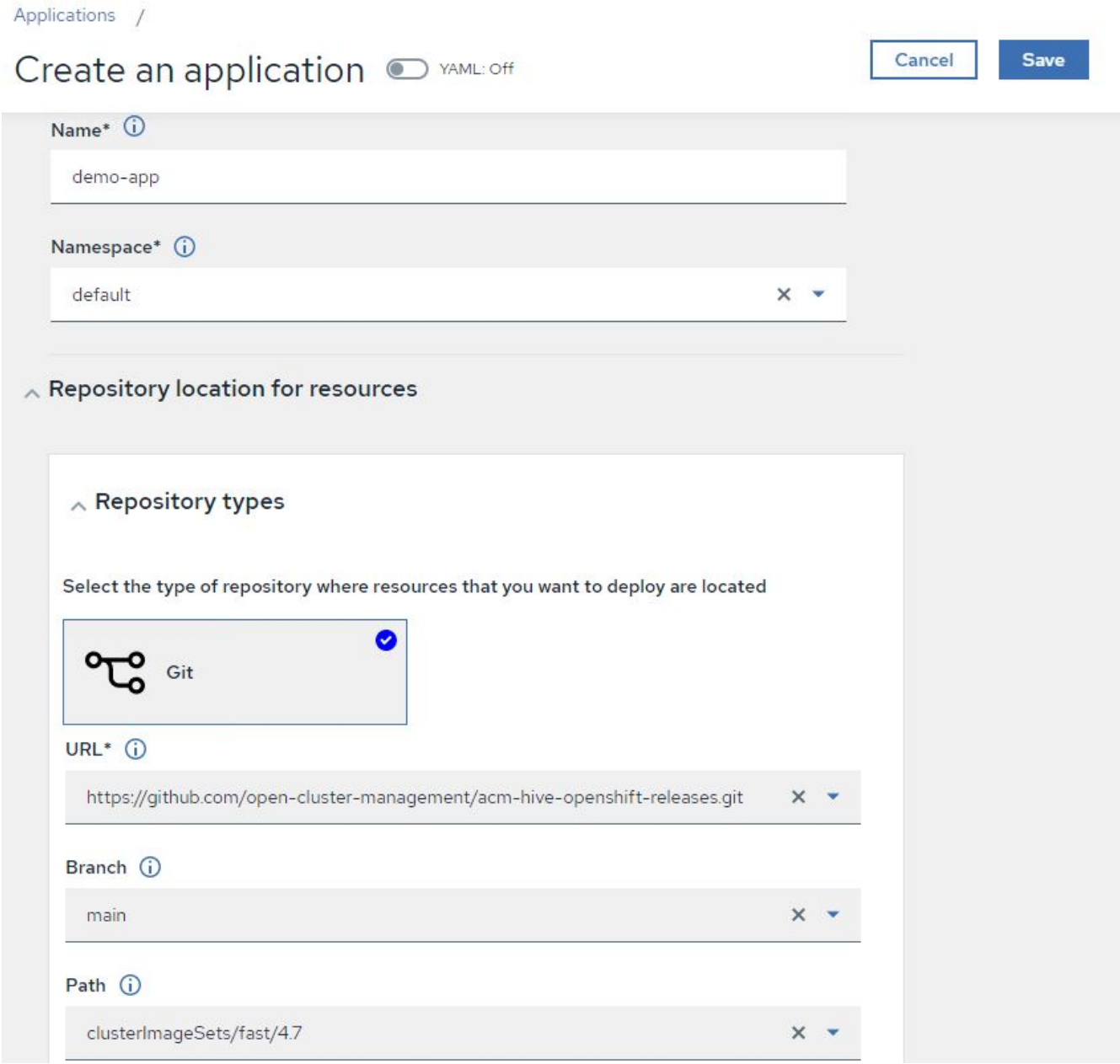
4. 建立及匯入多個叢集之後、您可以從單一主控台監控及管理這些叢集。

特色：採用**NetApp**的**Red Hat OpenShift**上的**Kubernetes**進階叢集管理

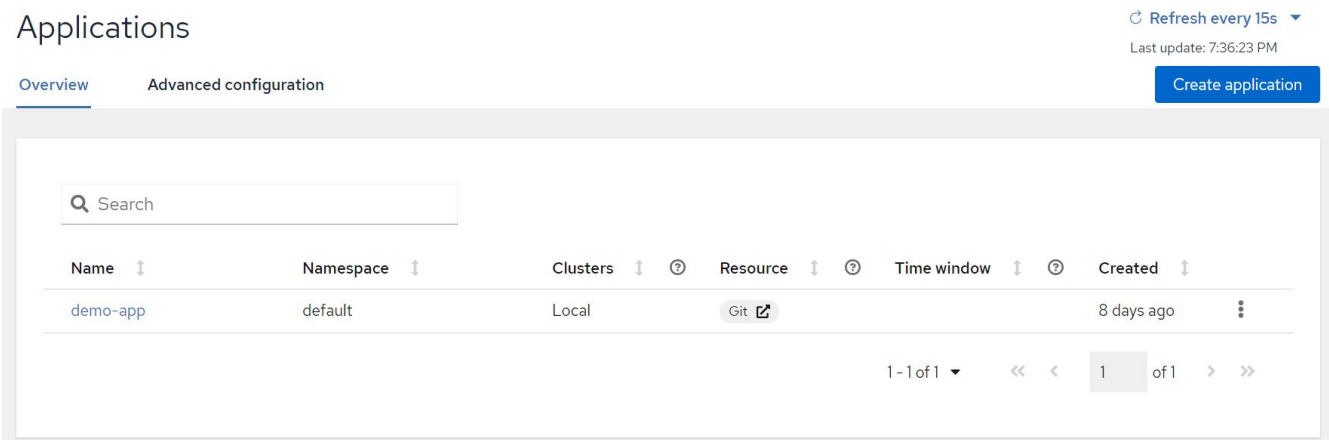
應用程式生命週期管理

若要在一組叢集之間建立應用程式並加以管理、

1. 從側邊列瀏覽至「管理應用程式」、然後按一下「建立應用程式」。提供您要建立的應用程式詳細資料、然後按一下「Save（儲存）」。



2. 安裝應用程式元件之後、應用程式會出現在清單中。



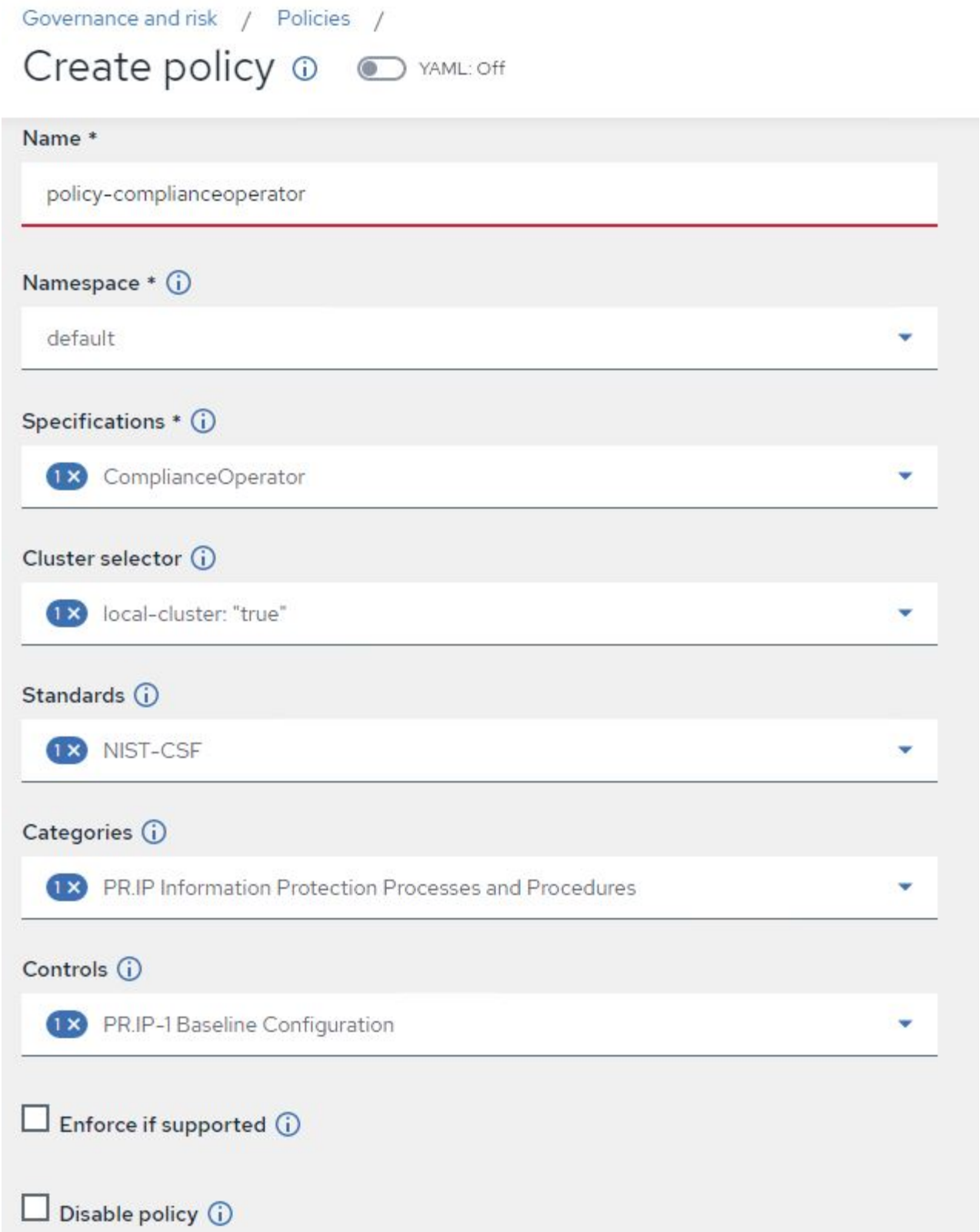
3. 現在可以從主控台監控及管理應用程式。

特色：採用**NetApp**的**Red Hat OpenShift**上的**Kubernetes**進階叢集管理

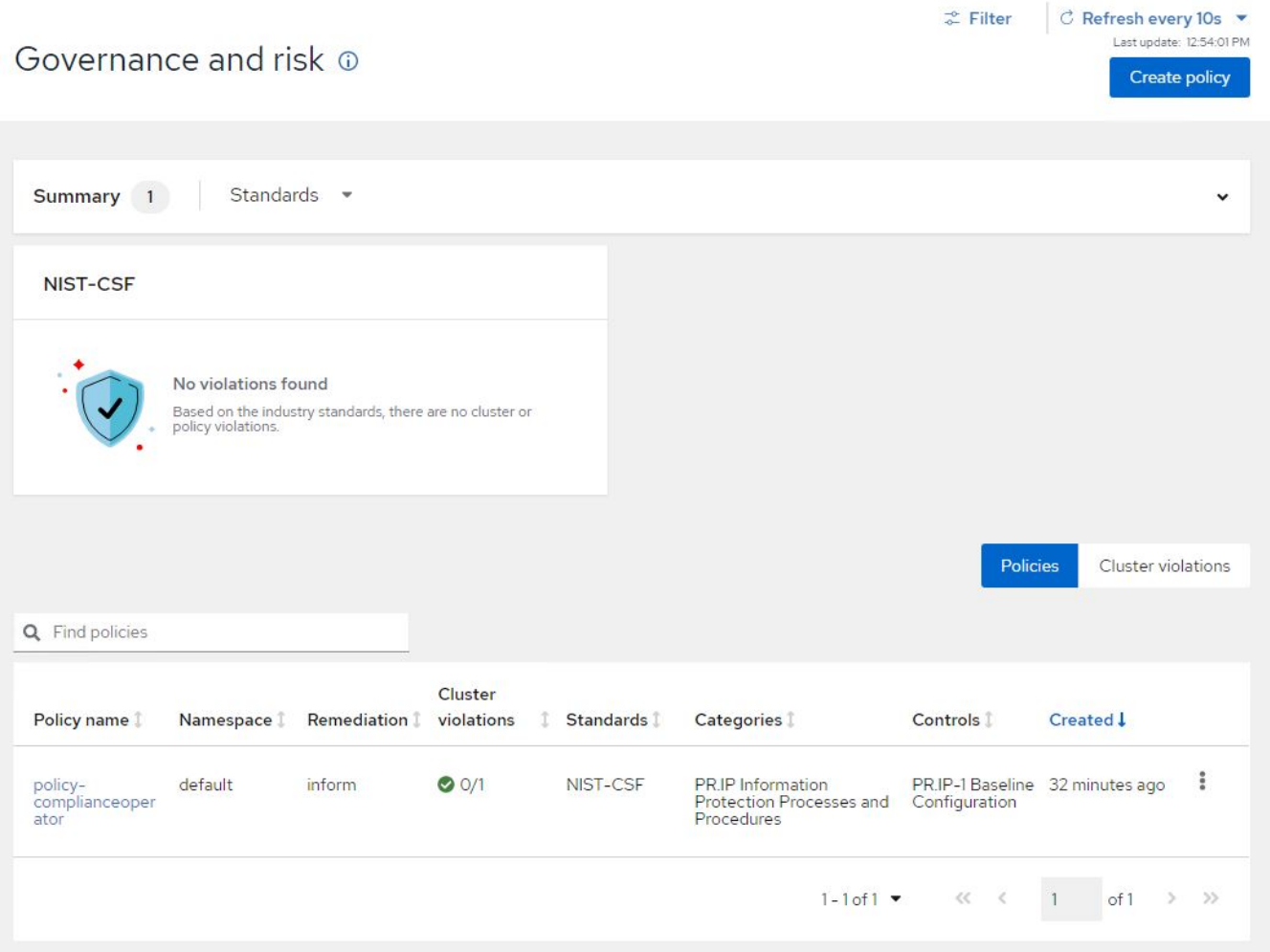
治理與風險

此功能可讓您針對不同的叢集定義法規遵循原則、並確保叢集符合此原則。您可以設定原則來通知或修正任何偏離或違反規則的情況。

1. 從側邊列導覽至「治理與風險」。
2. 若要建立規範原則、請按一下「建立原則」、輸入原則標準的詳細資料、然後選取應遵守此原則的叢集。如果您想要自動修正此原則的違規、請選取「強制執行（若有支援）」核取方塊、然後按一下「建立」。



3. 設定所有必要的原則之後、即可從「進階叢集管理」監控及修正任何原則或叢集違規。

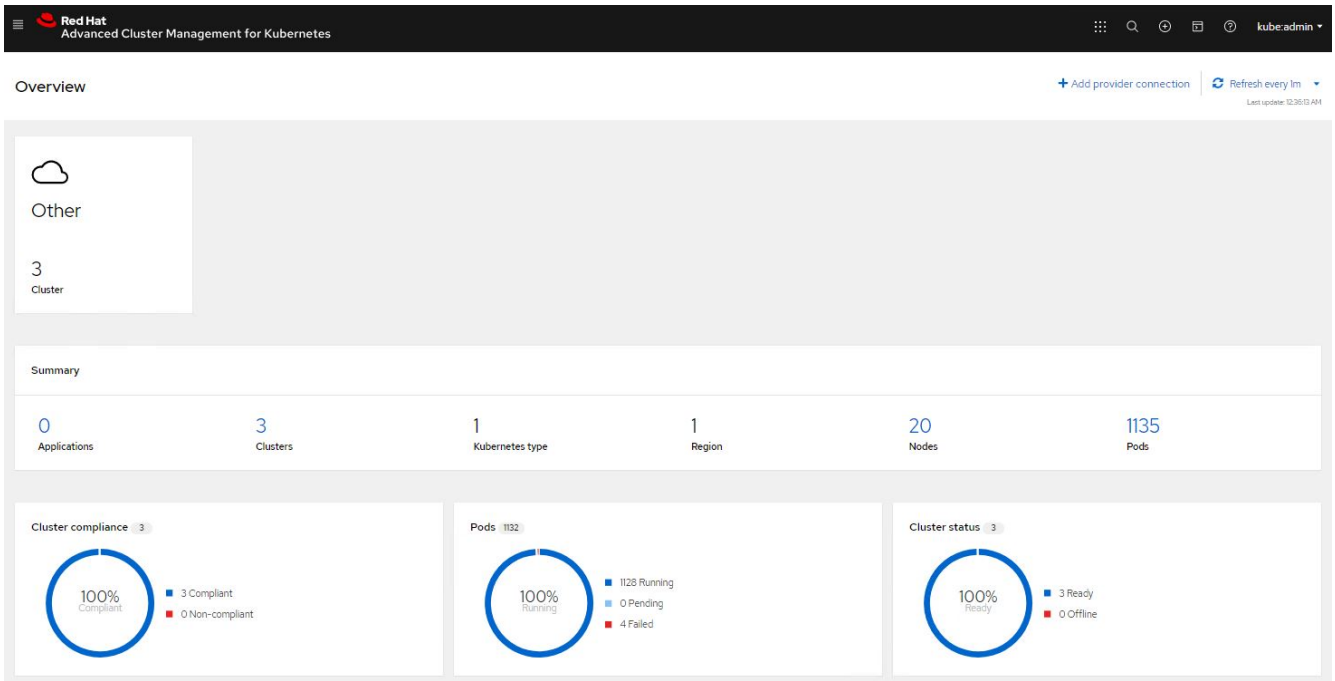


特色：採用**NetApp**的**Red Hat OpenShift**上的**Kubernetes**進階叢集管理

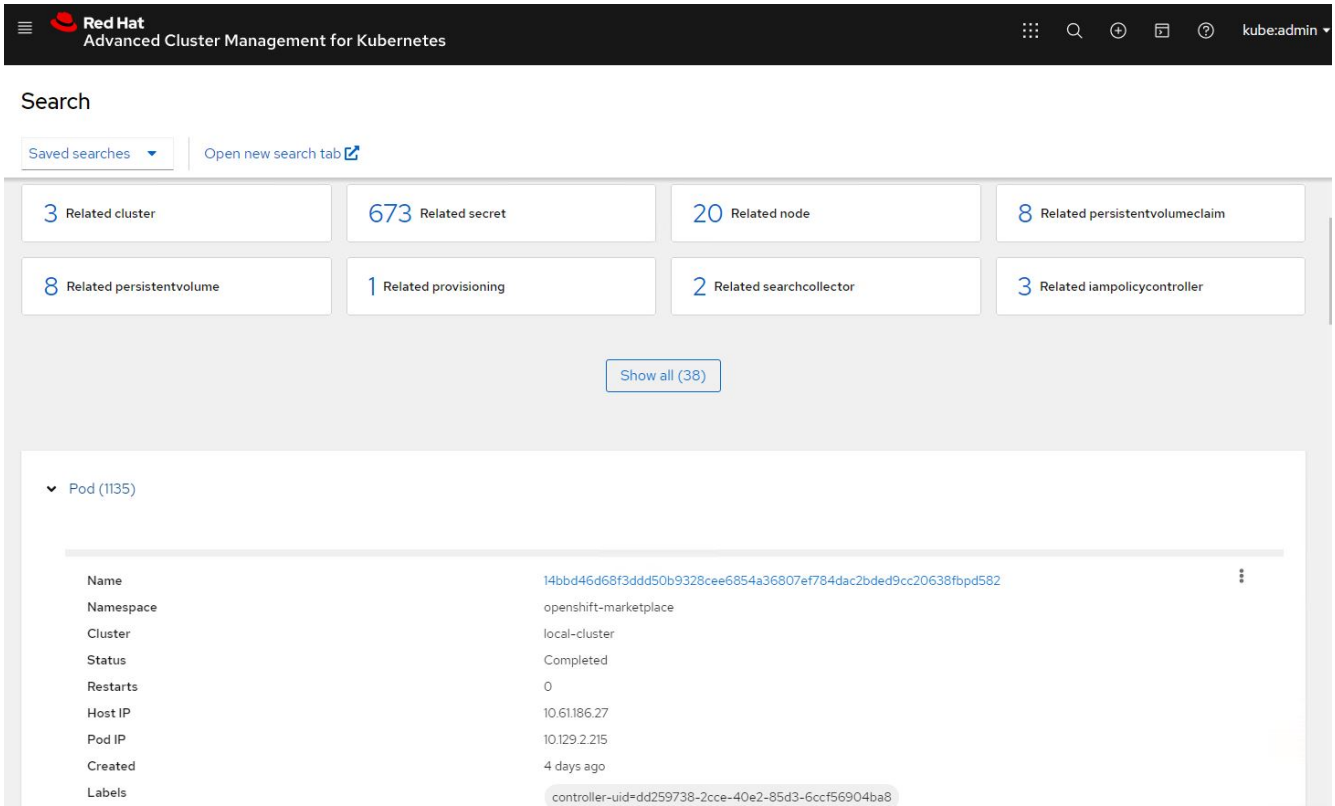
觀察能力

適用於Kubernetes的進階叢集管理提供一種方法、可監控所有叢集上的節點、Pod、應用程式和工作負載。

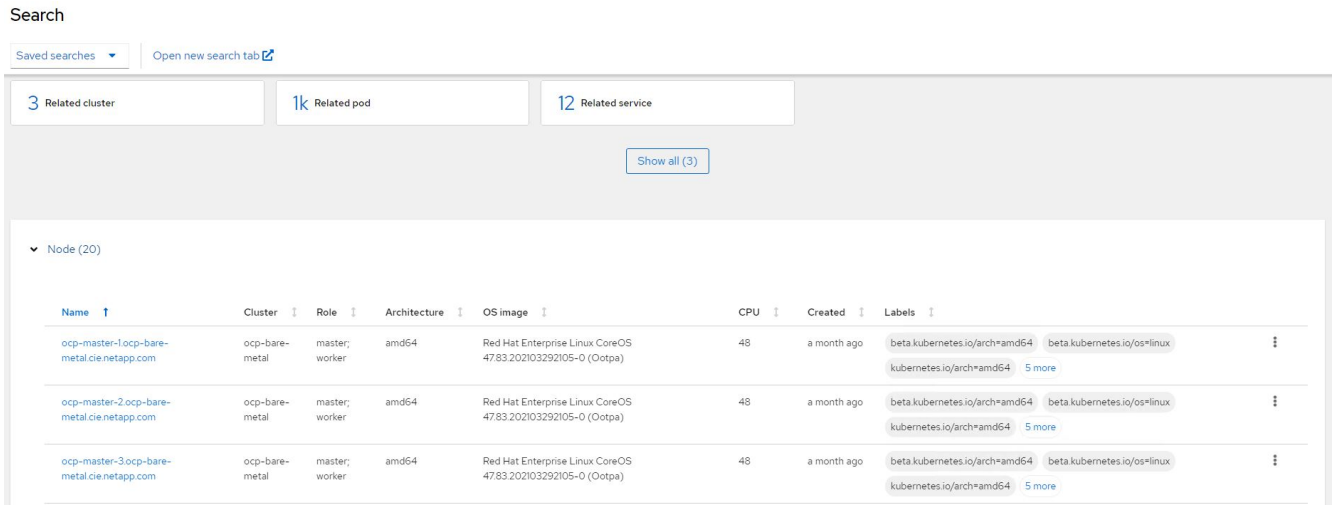
1. 瀏覽至「觀察環境」>「總覽」。



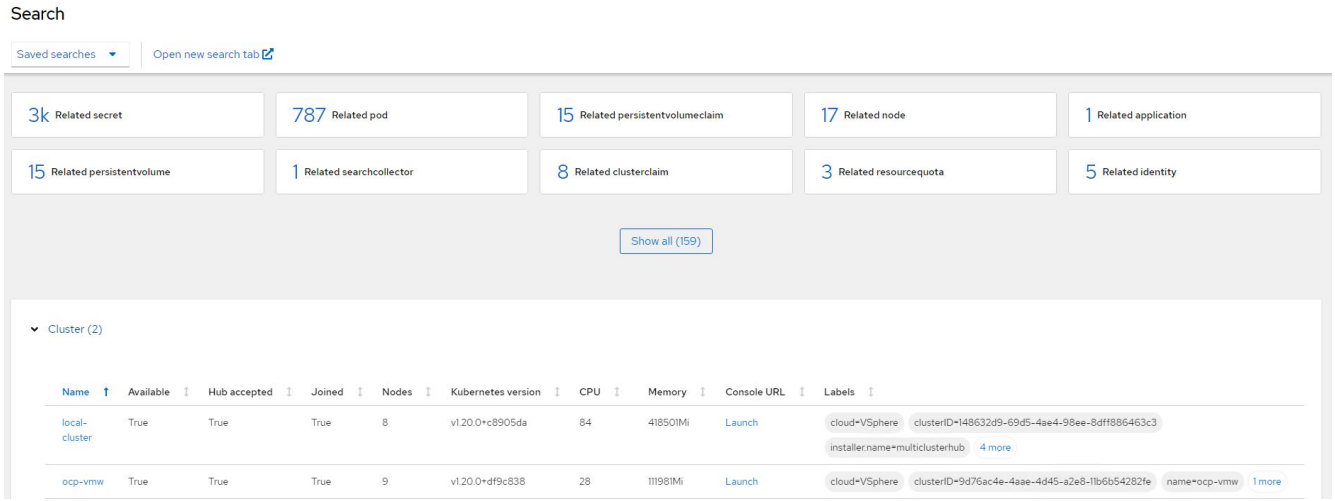
2. 所有叢集的所有Pod和工作負載都會根據各種篩選器進行監控和排序。按一下「Pod」以檢視對應的資料。



3. 叢集內的所有節點都會根據各種資料點進行監控與分析。按一下節點、深入瞭解對應的詳細資料。



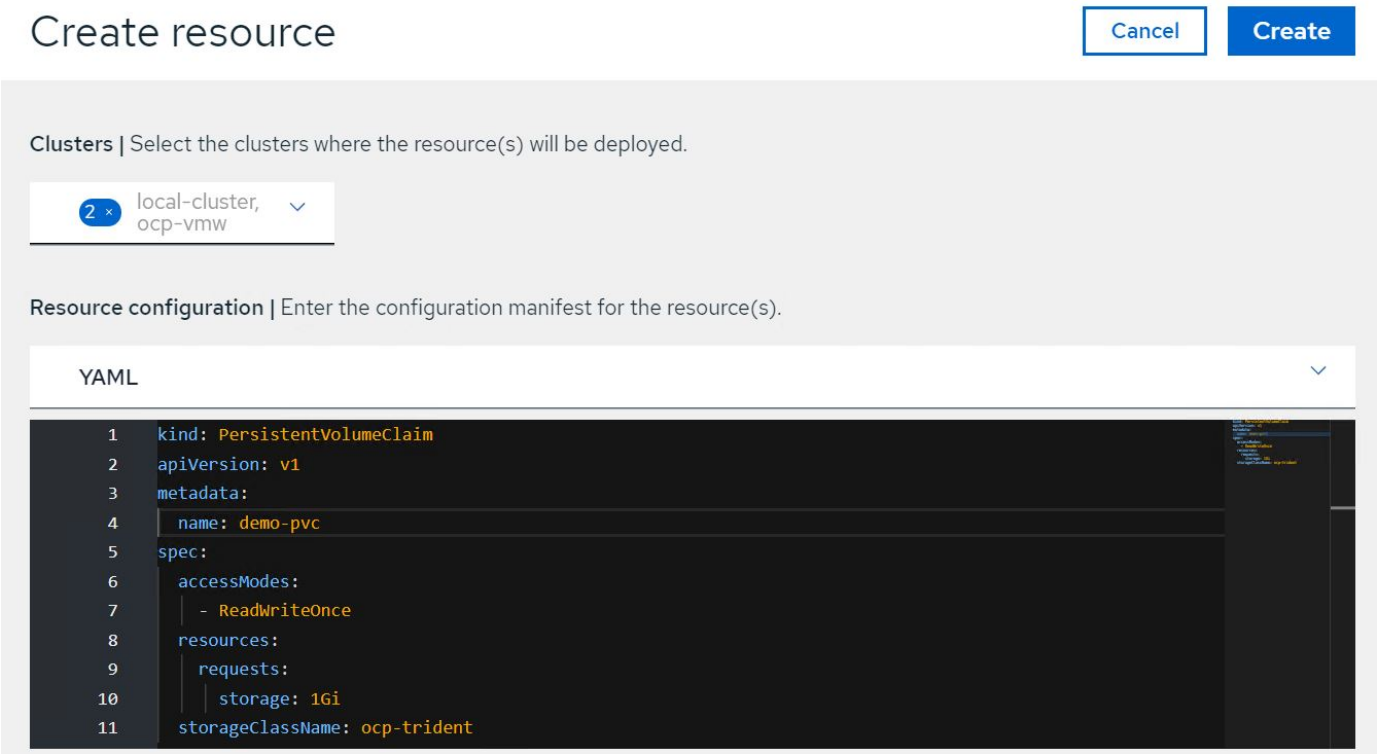
4. 所有叢集都會根據不同的叢集資源和參數進行監控和組織。按一下叢集以檢視叢集詳細資料。



特色：採用**NetApp**的**Red Hat OpenShift**上的**Kubernetes**進階叢集管理

在多個叢集上建立資源

Kubernetes的進階叢集管理功能可讓使用者從主控台同時在一或多個託管叢集上建立資源。舉例來ONTAP 說、如果您在不同站台有OpenShift叢集、並以不同的NetApp支援叢集做為後盾、而且想要在兩個站台上配置PVc、您可以按一下頂端列上的（+）符號。然後選取您要在其中建立永久虛擬基礎虛擬基礎網路的叢集、貼上資源Yaml、然後按一下「Create（建立）」。



影片與示範：Red Hat OpenShift with NetApp

下列影片示範本文件所述的部分功能：

使用 Red Hat MTV 將 VM 移轉至使用 NetApp ONTAP 儲存設備的 OpenShift 虛擬化

利用 Astra Control 和 NetApp FlexClone 技術加速軟體開發 - Red Hat OpenShift with NetApp

運用NetApp Astra Control執行事後分析及還原您的應用程式

Astra Control Center 提供 CI/CD 管線中的資料保護功能

使用 Astra Control Center 進行工作負載移轉： Red Hat OpenShift with NetApp

工作負載移轉：採用NetApp的Red Hat OpenShift

安裝OpenShift虛擬化：採用NetApp的Red Hat OpenShift

部署採用OpenShift虛擬化技術的虛擬機器-採用NetApp的Red Hat OpenShift

NetApp HCI for Red Hat OpenShift on Red Hat 虛擬化

其他資訊：Red Hat OpenShift with NetApp

若要深入瞭解本文件所述資訊、請檢閱下列網站：

- NetApp文件

  "https://docs.netapp.com/"

- Astra Trident文件

  "https://docs.netapp.com/us-en/trident/index.html"

- NetApp Astra Control Center文件

  "https://docs.netapp.com/us-en/astra-control-center/"

- Red Hat OpenShift文件

  "https://access.redhat.com/documentation/en-us/openshift_container_platform/4.7/"

- Red Hat OpenStack平台文件

  "https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/16.1/"

- Red Hat虚擬化文件

  "https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.4/"

- VMware vSphere文件

  "https://docs.vmware.com/"