



## 使用案例驗證 NetApp Solutions

NetApp  
April 12, 2024

This PDF was generated from [https://docs.netapp.com/zh-tw/netapp-solutions/containers/dwn\\_use\\_case\\_integrated\\_data\\_protection.html](https://docs.netapp.com/zh-tw/netapp-solutions/containers/dwn_use_case_integrated_data_protection.html) on April 12, 2024. Always check docs.netapp.com for the latest.

# 目錄

使用案例驗證：採用NetApp Astra的DevOps .....	1
利用NetApp Astra Control將保護功能整合至CI/CD管路 .....	1
使用Astra Control來協助事後分析及還原應用程式 .....	7
利用NetApp FlexClone技術加速軟體開發 .....	12

# 使用案例驗證：採用NetApp Astra的DevOps

下列使用案例已通過NetApp Astra DevOps驗證：

- "利用NetApp Astra Control將保護功能整合至CI/CD管路"
- "運用Astra Control來協助事後分析及還原應用程式"
- "利用NetApp FlexClones加速軟體開發"

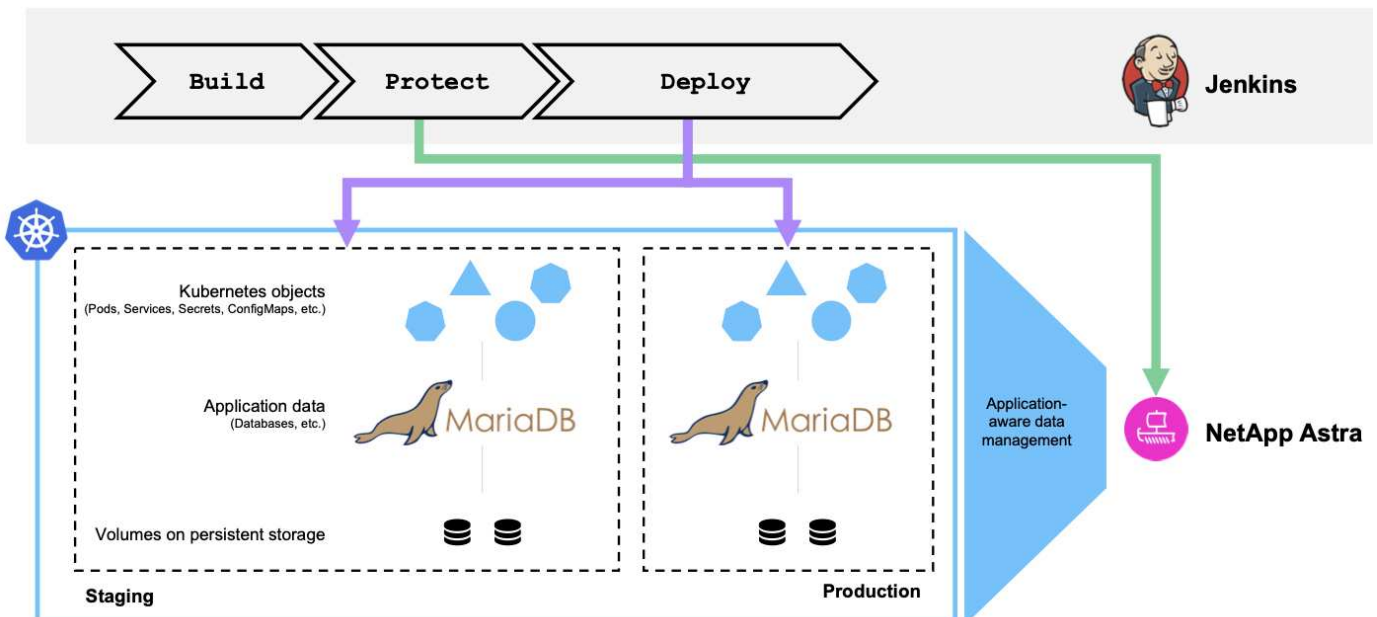
## 利用NetApp Astra Control將保護功能整合至CI/CD管路

### 總覽

DevOps工作流程最常見的用途之一是持續整合與持續部署（CI/CD）管線、可在開發人員提交新程式碼時、在應用程式上建置、整合及執行自動化測試套件。DevOps工程師與站台可靠性工程師（SRE）通常擁有專為開發新功能、回歸測試、錯誤修復、品質工程及開發流程中其他功能所設計的各種工作流程管道。

隨著團隊提高自動化程度、實際執行中應用程式的變更速度可能會令人感到難以抗拒。因此、有些團隊偏好保護正式作業中的應用程式或服務。除了保護程式碼和容器映像之外、他們也想要保護應用程式狀態、組態資料（例如Kubernetes物件和與應用程式相關的資源）、以及應用程式的持續資料。

在此使用案例中、我們將深入探討一條促銷到正式作業的傳輸途徑、這條途徑會部署新版本的應用程式：先進入接移環境、然後進入正式作業環境。此範例同樣適用於主要公有雲、也適用於內部部署環境。雖然我們展示了某個應用程式版本的部署、但管道也可搭配其他策略使用、例如藍色/綠色或金卡部署。在CI/CD管道中、我們將建立完整的應用程式備份來保護應用程式。對正式作業應用程式及其資料、狀態和組態的應用程式感知備份、對於許多DevOps工作流程而言、都很實用。



用於驗證此使用案例的應用程式為 "Magento" 這是一種電子商務解決方案、具備網路型前端、搜尋與分析功能的Elasticsearch執行個體、以及追蹤所有購物庫存與交易詳細資料的MariaDB資料庫。此容器化應用程式安裝於Red Hat OpenShift叢集。應用程式中的每個Pod都使用持續磁碟區來儲存資料。NetApp Astra Trident會自動建立持續性磁碟區、這是Container Storage Interface相容的Kubernetes儲存協調程式、可在NetApp儲存系統上配置儲存設備。此外、為了利用Astra Control Center的應用程式保護功能、相關應用程式是由Astra Control管

理、Astra Control隨後用於觸發應用程式備份、以儲存應用程式狀態及持續磁碟區中的資料。我們使用 "[NetApp Astra Control Python SDK](#)" 將觸發應用程式備份的程序自動化、然後將其導入CI/CD傳輸途徑。此管道是使用稱為的熱門CI/CD工具來建立和執行 ["[Jenkins](#)"]將建置、保護及部署應用程式的流程自動化。

讓我們來執行先決條件和程序、以便在CI/CD管道中引進保護措施。

## 使用案例驗證先決條件

下列工具或平台已部署並設定為先決條件：

1. Red Hat OpenShift Container Platform
2. NetApp Astra Trident安裝在OpenShift上、ONTAP 並已設定後端至NetApp的整套系統
3. 預設已設定的儲存設備、指向NetApp ONTAP 的後端
4. NetApp Astra Control Center安裝在OpenShift叢集上
5. OpenShift叢集已新增為Astra Control Center的受管理叢集
6. Jenkins安裝在OpenShift叢集上、並設定了安裝Docker引擎的代理程式節點

## 安裝應用程式

讓我們先從在接移和正式作業環境中初始安裝應用程式開始著手。就本使用案例而言、此步驟為必要步驟、因此必須手動執行。CI/CD傳輸途徑可用於應用程式新版本的後續建置與部署工作流程。

此使用案例中的正式作業環境稱為「magento prod」命名空間、而對應的接移環境則是在Red Hat OpenShift叢集上設定的命名空間「magento接移」。若要安裝所有應用程式、請完成下列步驟：

1. 在正式作業環境中使用位元名稱管理圖表來安裝Magento應用程式。我們將rwx PV用於Magento和MariaDB Pod。

```
[netapp-user@rhel7 na_astra_control_suite]$ helm install --version 14
magento bitnami/magento -n magento-prod --create-namespace --set
image.tag=2.4.1-debian-10-
r11,magentoHost=10.63.172.243,persistence.magento.accessMode=ReadWriteMa
ny,persistence.apache.accessMode=ReadWriteMany,mariadb.master.persistenc
e.accessModes[0]=ReadWriteMany
```



Magento bitnami船旗圖表需要負載平衡器服務才能公開Magento GUI服務。我們使用 "[MetalLB](#)." 以提供內部負載平衡器服務。

2. 幾分鐘後、請確認所有的Pod和服務都在執行中。

```
[netapp-user@rhel7 na_astra_control_suite]$ oc get pods -n magento-prod
NAME                                READY   STATUS
RESTARTS   AGE
magento-9d658fd96-qrxmt             1/1     Running
0         49m
magento-elasticsearch-coordinating-only-69869cc5-768rm  1/1     Running
0         49m
magento-elasticsearch-data-0        1/1     Running
0         49m
magento-elasticsearch-master-0      1/1     Running
0         49m
magento-mariadb-0                   1/1     Running
0         49m
```

3. 對接移環境重複相同的程序。

## 在Astra Control Center中管理Magento應用程式

1. 瀏覽至「應用程式」、然後選取「探索到的應用程式」索引標籤。
2. 在正式作業環境中、按一下Magento應用程式的省略符號（「magent-prod」）、然後按一下「Manage（管理）」。
3. Magento應用程式現在由Astra Control Center管理。Astra Control支援的所有作業都可在應用程式上執行。也請記下應用程式的版本。

**magento-prod** Available

App status: Healthy

App protection status: Partially Protected

Images:  
 docker.io/bitnami/elasticsearch:6.8.10-debian-10-r16  
 docker.io/bitnami/magento:2.4.1-debian-10-r11  
 docker.io/bitnami/mariadb:10.3.23-debian-10-r38

Protection schedule: Disabled

Group: magento-prod

Cluster: ocp-vmw

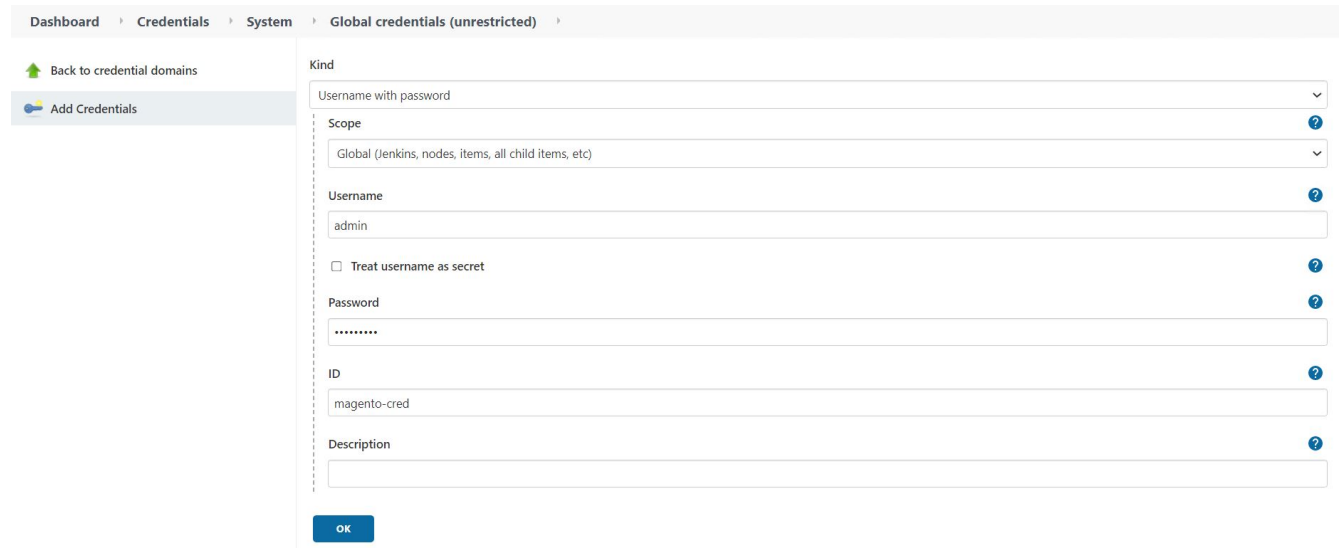
4. 在接移環境（「magento接移」）中重複管理Magento應用程式的步驟。

## 整合式保護功能的CI/CD傳輸途徑

當我們使用新版本的應用程式時、我們會使用CI/CD管道來建置容器映像、備份暫存和正式作業環境、將新版本的應用程式部署到暫存環境、等待核准升級至正式作業環境、然後將新版本的應用程式部署到正式作業環境。若要使用CI/CD管道、請完成下列步驟：

1. 登入Jenkins並建立所需的認證資料：一項用於Magento creds、一項用於MariaDB管理creds、第三項用於MariaDB根creds。
2. 瀏覽至「管理Jenkins」>「管理認證」、然後按一下適當的網域。

3. 按一下「Add Credentials（新增認證）」、然後將「種類」設為「使用者名稱」、並將「範圍」設為輸入認證的使用者名稱、密碼和ID、然後按一下「OK（確定）」。




4. 對其他兩個認證重複相同的程序。
5. 返回儀表板、按一下「New item（新項目）」建立管道、然後按一下「Pipeline（管道）」。
6. 從Jenkinsfile複製管路 ["請按這裡"](#)。
7. 將管線貼到Jenkins Pipeline區段、然後按一下「Save（儲存）」。
8. 在Jenkins管線的參數中填入相關詳細資料、包括船舵圖表版本、要升級至的Magento應用程式版本、Astra工具組版本、Astra Control Center FQDN、API權杖及其執行個體ID。指定正式作業和暫存環境的Docker登錄、命名空間和Magento IP、並指定所建立認證的認證ID。


```
MAGENTO_VERSION = '2.4.1-debian-10-r14'
CHART_VERSION = '14'
RELEASE_TYPE = 'MINOR'
ASTRA_TOOLKIT_VERSION = '2.0.2'
ASTRA_API_TOKEN = 'xxxxxxxxx'
ASTRA_INSTANCE_ID = 'xxx-xxx-xxx-xxx-xxx'
ASTRA_FQDN = 'netapp-astra-control-center.org.example.com'
DOCKER_REGISTRY = 'docker.io/netapp-solutions-cicd'
PROD_NAMESPACE = 'magento-prod'
PROD_MAGENTO_IP = 'x.x.x.x'
STAGING_NAMESPACE = 'magento-staging'
STAGING_MAGENTO_IP = 'x.x.x.x'
MAGENTO_CREDS = credentials('magento-cred')
MAGENTO_MARIADB_CREDS = credentials('magento-mariadb-cred')
MAGENTO_MARIADB_ROOT_CREDS = credentials('magento-mariadb-root-cred')
```


9. 按一下「立即建置」。管道會開始執行、並逐步完成各個步驟。應用程式映像會先建置並上傳至Container登錄。

Build & Publish Segment	Build Docker Image	Publish Image to Registry	Protect & Deploy Segment	Install & Configure Pre-requisites	Download & Configure Astra Toolkit	Backup Tasks	Backup of Staging Env	Backup of Production Env	Deploy to Staging environment [Minor/Patch]	Deploy to Staging environment [Major]	Promote to Production?	Deploy to Production environment [Minor/Patch]	Deploy to Production environment [Major]	Delete temporary files
4s	24s	5s	213ms	40s	2s	290ms	1min 38s	1min 2s	6min 29s	229ms	361ms	2min 57s	200ms	850ms
3s														
18min 29s														

10. 應用程式備份是透過Astra Control啟動。


**magento-prod**
Available

 App status  
Healthy

 App protection status  
Partially Protected

**Images**  
docker.io/bitnami/elasticsearch:6.8.10-debian-10-r16  
docker.io/bitnami/magento:2.4.1-debian-10-r11  
docker.io/bitnami/mariadb:10.3.23-debian-10-r38

**Protection schedule**  
Disabled




**Group**  
magento-prod

**Cluster**  
ocp-vmw


Overview
Data protection
Storage
Resources
Activity


Actions
Configure protection policy
Search


1-8 of 8 entries

<input type="checkbox"/>	Name	Ready	On-Schedule/On-Demand	Created ↑	Actions
<input type="checkbox"/>	upgrade-prod-2-4-1-debian-10-r20		 On-Demand	2021/10/29 14:43 UTC	Running 

11. 備份階段成功完成後、請從Astra Control Center驗證備份。


**magento-prod**
Available



**App status**  
Healthy


**App protection status**  
Partially Protected

**Images**  
 docker.io/bitnami/elasticsearch:6.8.10-debian-10-r16  
 docker.io/bitnami/magento:2.4.1-debian-10-r11  
 docker.io/bitnami/mariadb:10.3.23-debian-10-r38



**Protection schedule**  
 Disabled

**Group**  
 magento-prod

**Cluster**  
 ocp-vmw

Overview
**Data protection**
Storage
Resources
Activity

Actions
 Configure protection policy

Search
 


1-8 of 8 entries
 <
>

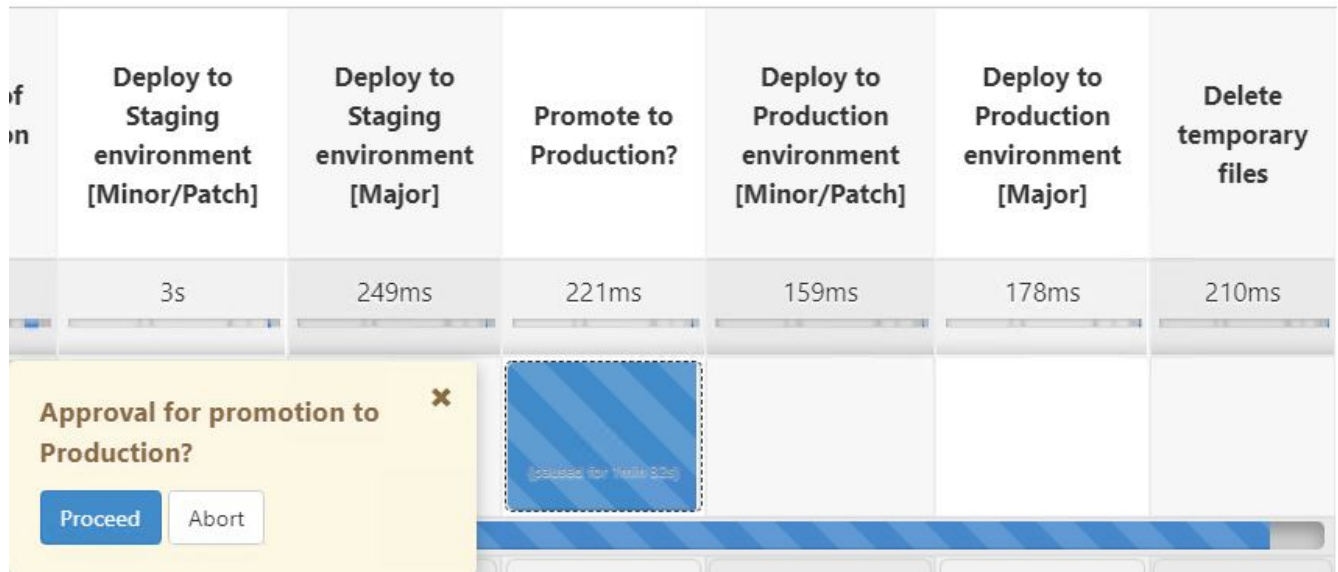
<input type="checkbox"/>	Name	Ready	On-Schedule/On-Demand	Created ↑	Actions
<input type="checkbox"/>	upgrade-prod-2-4-1-debian-10-r20	<span>Healthy</span>	On-Demand	2021/10/29 14:43 UTC	<span>Available</span>

12. 然後將新版本的應用程式部署至接移環境。

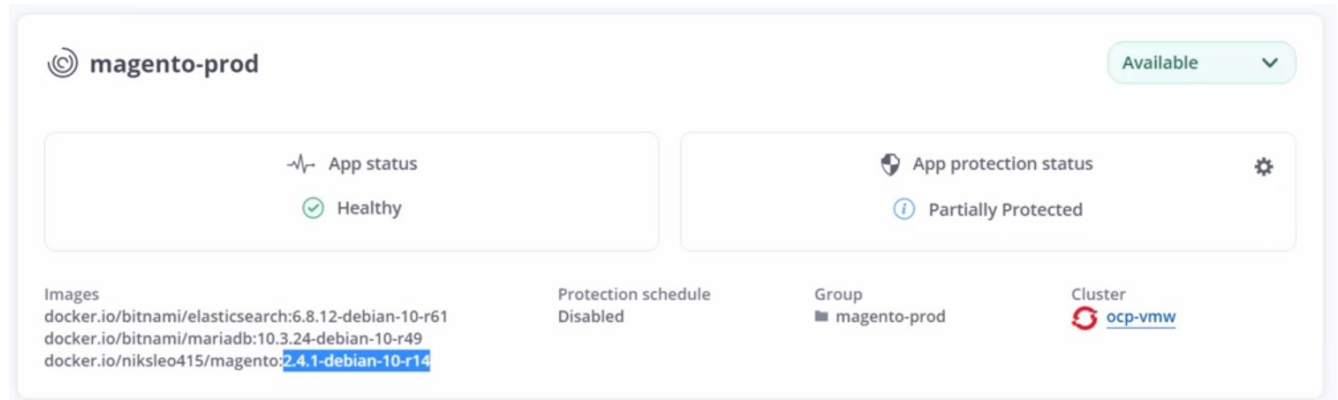
Build & Publish Segment	Build Docker Image	Publish Image to Registry	Protect & Deploy Segment	Install & Configure Pre-requisites	Download & Configure Astra Toolkit	Backup Tasks	Backup of Staging Env	Backup of Production Env	Deploy to Staging environment [Minor/Patch]	Deploy to Staging environment [Major]	Promote to Production?	Deploy to Production environment [Minor/Patch]	Deploy to Production environment [Major]	Delete temporary files
4s	47s	7s	238ms	1min 25s	2s	273ms	1min 53s	1min 18s	5min 20s	211ms	337ms	2min 39s	187ms	780ms
3s	4min 16s	30s	485ms	7s	3s	153ms	6min 9s	5min 9s						
7min 1s														

13. 完成此步驟之後、程式會等待使用者核准正式作業部署。在此階段、假設QA團隊執行一些手動測試並核准正式作業。然後、您可以按一下「核准」、將新版本的應用程式部署到正式作業環境。





14. 確認正式作業應用程式也已升級至所需的版本。



在CI/CD管道中、我們建立完整的應用程式感知備份、展現保護應用程式的能力。由於整個應用程式都已備份為促銷活動到正式作業的傳輸途徑之一、因此您對於高度自動化的應用程式部署更有信心。此應用程式感知備份包含應用程式的資料、狀態和組態、對於眾多DevOps工作流程而言非常實用。其中一項重要工作流程是在發生無法預期的問題時、將應用程式回復至舊版。

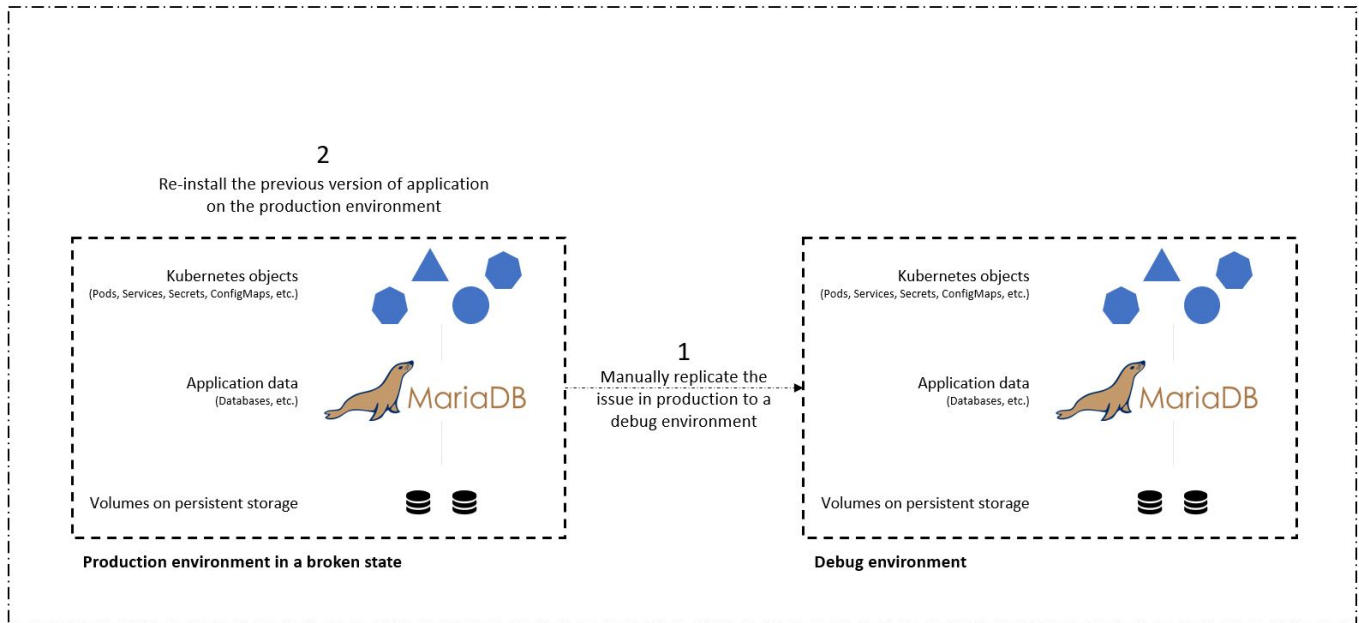
雖然我們透過Jenkins工具展示了CI/CD工作流程、但這項概念可以輕鬆且有效率地推斷到不同的工具和策略。若要查看實際使用案例、請觀看影片 ["請按這裡"](#)。

## 使用Astra Control來協助事後分析及還原應用程式

### 總覽

在中 ["第一次使用案例"](#)、我們示範如何使用NetApp Astra Control Center來保護Kubernetes中的應用程式。本節說明如何使用NetApp Astra工具套件中的Python SDK、透過Astra Control將應用程式備份直接整合至您的開發工作流程。此方法可在持續整合與持續部署（CI/CD）程序期間、將隨需備份自動化、藉此保護開發與正式作業環境。在CI/CD傳輸途徑和正式作業應用程式中加入這一層額外的應用程式一致資料保護功能、如果程序發生問題、開發程序就能安全無虞、進而促進良好的營運持續性實務做法。

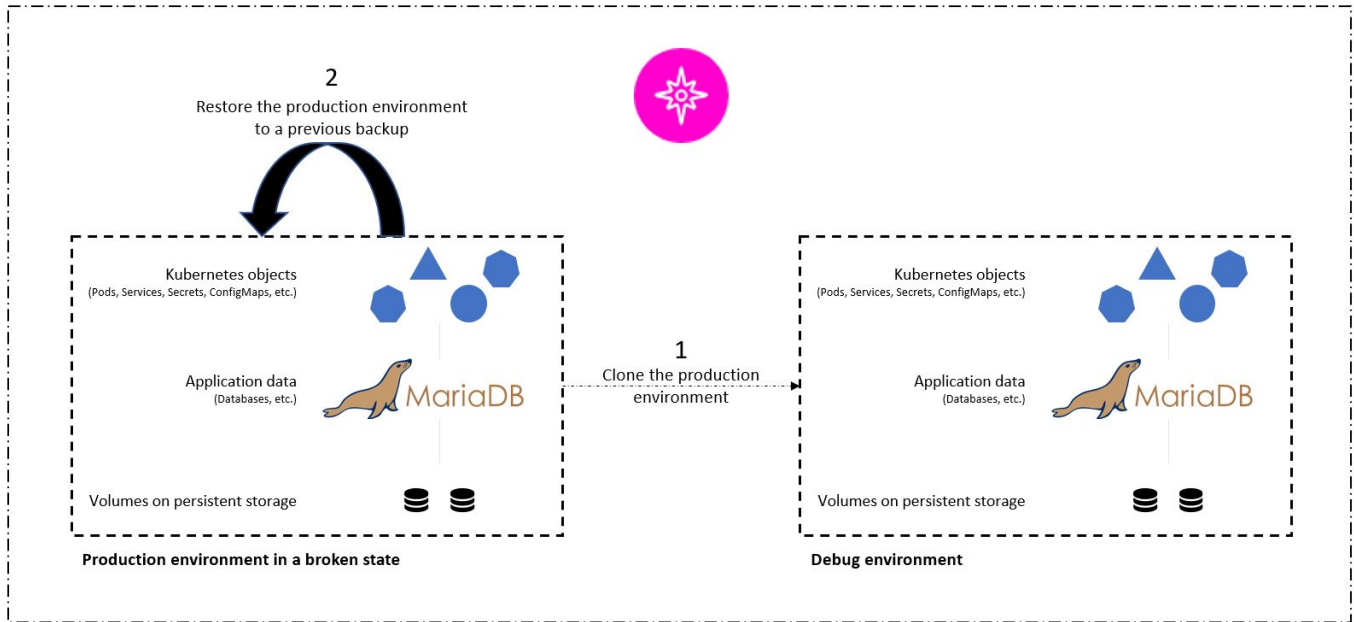
在傳統的工作流程中、當應用程式升級至新版本時遇到故障之後、開發團隊會根據客戶提供的錯誤報告、嘗試即時疑難排解問題。或者、在第一次發生問題時、團隊可能會嘗試將應用程式重新部署到平行的偵錯環境、使該程序離線。他們可以將舊版程式碼庫從舊版重新部署至正式作業環境、以便將應用程式還原至正常運作狀態。



雖然此方法可行、但團隊必須確保中斷的正式作業應用程式狀態與發生問題時在正式作業環境中看到的版本相符。他們也必須花時間將已知良好的建置作業提升至正式作業環境、方法是從儲存庫擷取程式碼、然後重新部署機器映像、將應用程式還原至良好的執行狀態。此外、在此案例中、我們並未考慮正式作業資料庫本身是否因為錯誤的程式碼而毀損。理想情況下、資料庫資料有個別的備份程序、但我們必須假設它們與應用程式發佈時的狀態一致嗎？使用Astra Control進行狀態化和應用程式一致的備份、還原和複製、在這種情況下、真正展現出其價值。

首先、我們可以使用Astra Control來協助驗證後分析應用程式狀態。我們會以應用程式一致的方式、將正式作業版本複製到平行測試環境、以達成此目標。將此環境保留在錯誤狀態下、讓我們能夠即時疑難排解問題。

此外、Astra Control還支援就地還原功能、可讓我們將正式作業應用程式還原至上次可接受的備份（在受影響的程式碼版本之前）。還原的版本會以應用程式一致且有狀態的方式（包括先前指派的入口IP）、假設先前的錯誤正式作業應用程式位置。因此、存取前端的客戶將不會察覺到移轉到備份版本。



## 使用案例驗證先決條件

下列工具或平台已部署並設定為先決條件：

- Red Hat OpenShift Container Platform。
- NetApp Astra Trident安裝在OpenShift上、後端設定為NetApp ONTAP 的整套系統。
- 預設已設定的儲存設備、指向NetApp ONTAP 的後端。
- NetApp Astra Control Center安裝在OpenShift叢集上。
- OpenShift叢集已新增為Astra Control Center的受管理叢集。
- Jenkins安裝在OpenShift叢集上。
- 安裝在正式作業環境中的Magento應用程式。此使用案例中的正式作業環境是Red Hat OpenShift叢集中稱為「magento prod」的命名空間。
- 由Astra Control Center管理的正式作業應用程式。
- 使用Astra Control擷取的正式作業應用程式已知良好的備份。

## 複製與還原管線

考慮到應用程式已升級至新版本、正式作業環境中的應用程式（「magento prod」）在升級後並未如預期運作。假設由前端查詢傳回的資料與要求不符、或是資料庫已毀損。若要複製及還原管線、請完成下列步驟：



## This site can't be reached

10.63.172.243 took too long to respond.

Try:

- Checking the connection
- [Checking the proxy and the firewall](#)
- [Running Windows Network Diagnostics](#)

ERR\_CONNECTION\_TIMED\_OUT

Reload

Details

1. 登入Jenkins並按一下「New item（新項目）」、然後按一下「Pipeline（管道）」、建立管道。
2. 從Jenkinsfile複製管路 "[請按這裡](#)"。
3. 將管線貼到Jenkins Pipeline區段、然後按一下「Save（儲存）」。
4. 在Jenkins管線的參數中填入相關的詳細資料、例如目前正式作業的Magento應用程式版本、Astra Control Center FQDN、API權杖、正式作業與偵錯環境的執行個體ID和應用程式名稱或命名空間、以及來源與目的地叢集名稱。就本使用案例而言、正式作業環境是稱為「magent-prod」的命名空間、而偵錯環境則是在Red Hat OpenShift叢集上設定的命名空間、稱為「magento除錯」。

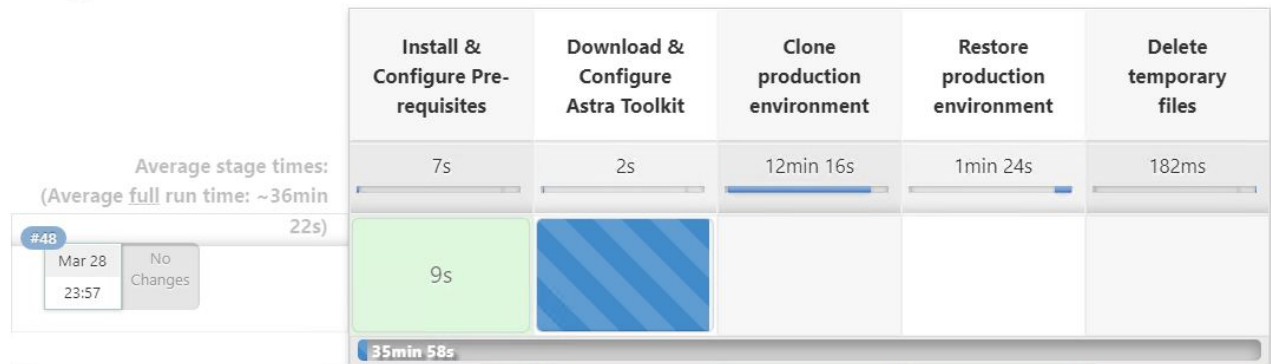
```
MAGENTO_VERSION = '2.4.1-debian-10-r14'
ASTRA_TOOLKIT_VERSION = '2.0.2'
ASTRA_API_TOKEN = 'xxxxxx'
ASTRA_INSTANCE_ID = 'xxx-xxx-xxx-xxx-xxx'
ASTRA_FQDN = 'netapp-astra-control-center.org.example.com'
PROD_APP_NAME = 'magento-prod'
DEBUG_APP_NAME = 'magento-debug'
DEBUG_NAMESPACE = 'magento-debug'
PROD_KUBERNETES_CLUSTER = 'ocp-vmw'
DEBUG_KUBERNETES_CLUSTER = 'ocp-vmw'
```

5. 按一下「立即建置」。管道會開始執行、並逐步完成各個步驟。應用程式會先在目前狀態下複製到偵錯環境、然後將應用程式還原至已知可運作的備份。

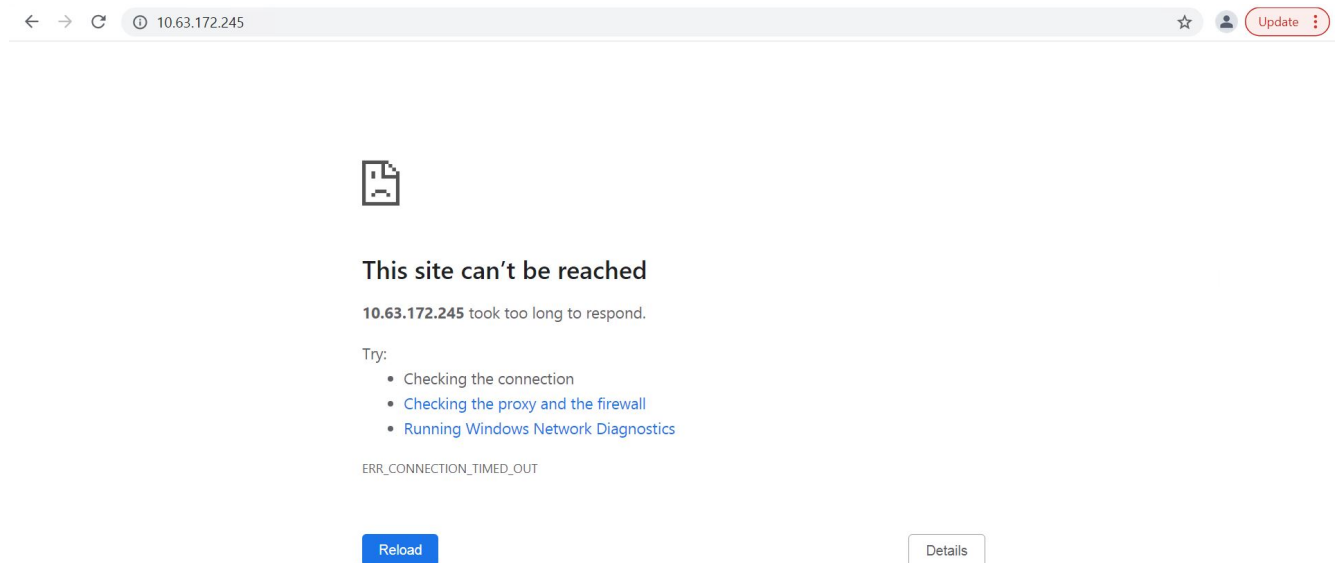
# Pipeline magento\_clone-for-triage\_restore-from-backup



## Stage View

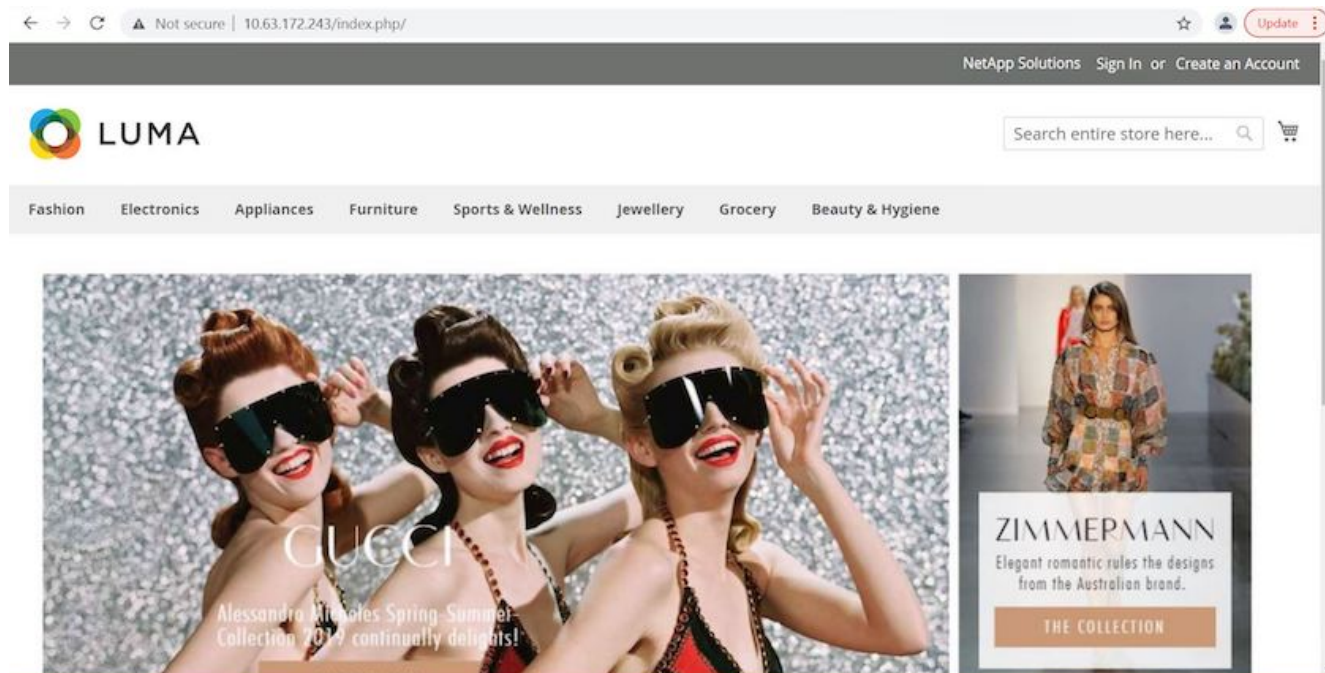


6. 驗證複製的應用程式是否為內含錯誤的版本。



7. 確認正式作業環境已還原為正常運作的備份、而且正式作業中的應用程式也能正常運作。





這兩項作業會同時加速恢復正常業務營運。若要查看實際使用案例、請觀看影片 ["請按這裡"](#)。

## 利用NetApp FlexClone技術加速軟體開發

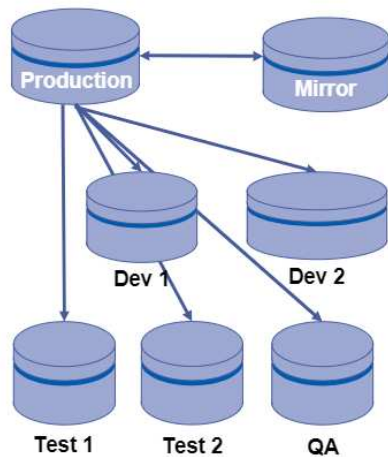
### 總覽

在Kubernetes叢集中複製已部署的應用程式、對於想要與合作夥伴共享環境、或在開發環境中測試新版本程式碼、而不干擾目前所使用版本的開發人員來說、是非常實用的工具。Kubernetes應用程式的狀態化及應用程式一致的複製、是NetApp Astra Control隨附的主要功能、以及應用程式的備份與還原。此外、如果應用程式是使用相同的儲存後端、在同一個Kubernetes叢集內複製、則Astra Control預設使用NetApp FlexClone技術來複製持續資料磁碟區、因此可大幅加速程序。藉由加速此程序、複製的環境會在幾分鐘內完成配置並可供使用、相較於重新部署測試或開發環境、開發人員只需稍作暫停即可恢復工作。此外、NetApp Astra Control提供的所有功能都可透過API來呼叫、以便輕鬆整合至Ansible等自動化架構。因此、環境可以更快速地分段、因為只需要在方針或角色中做些細微的變更、就能開始複製程序。

### 什麼是NetApp FlexClone技術？

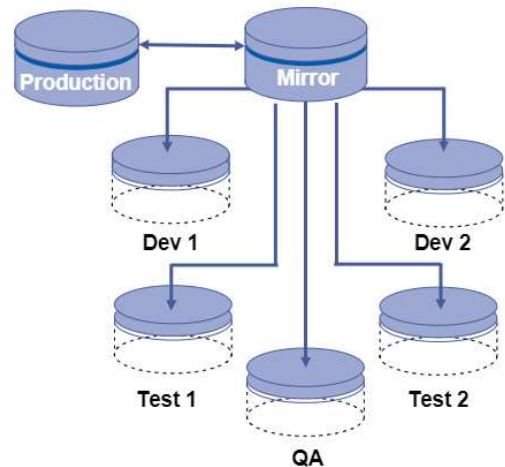
NetApp FlexClone技術是可寫入的時間點Snapshot型NetApp FlexVol 產品複本。這些資源幾乎會立即配置、包含來源磁碟區的所有資料、而且在新磁碟區中的資料開始從來源轉移之前、不會佔用額外的儲存空間。當多個資料複本可用於接移目的、而儲存系統的資源有限、無法配置這些磁碟區時、通常會在開發或範本型環境中使用這些資料複本。相較於必須多次複製資料以節省大量儲存空間與時間的傳統儲存系統、NetApp FlexClone技術可加速儲存相關工作。

## Traditional Data Copies



Traditional physical copies take additional time and consume additional storage space

## NetApp FlexClone Copies



NetApp FlexClone copies are near instantaneous and only consume space when written to

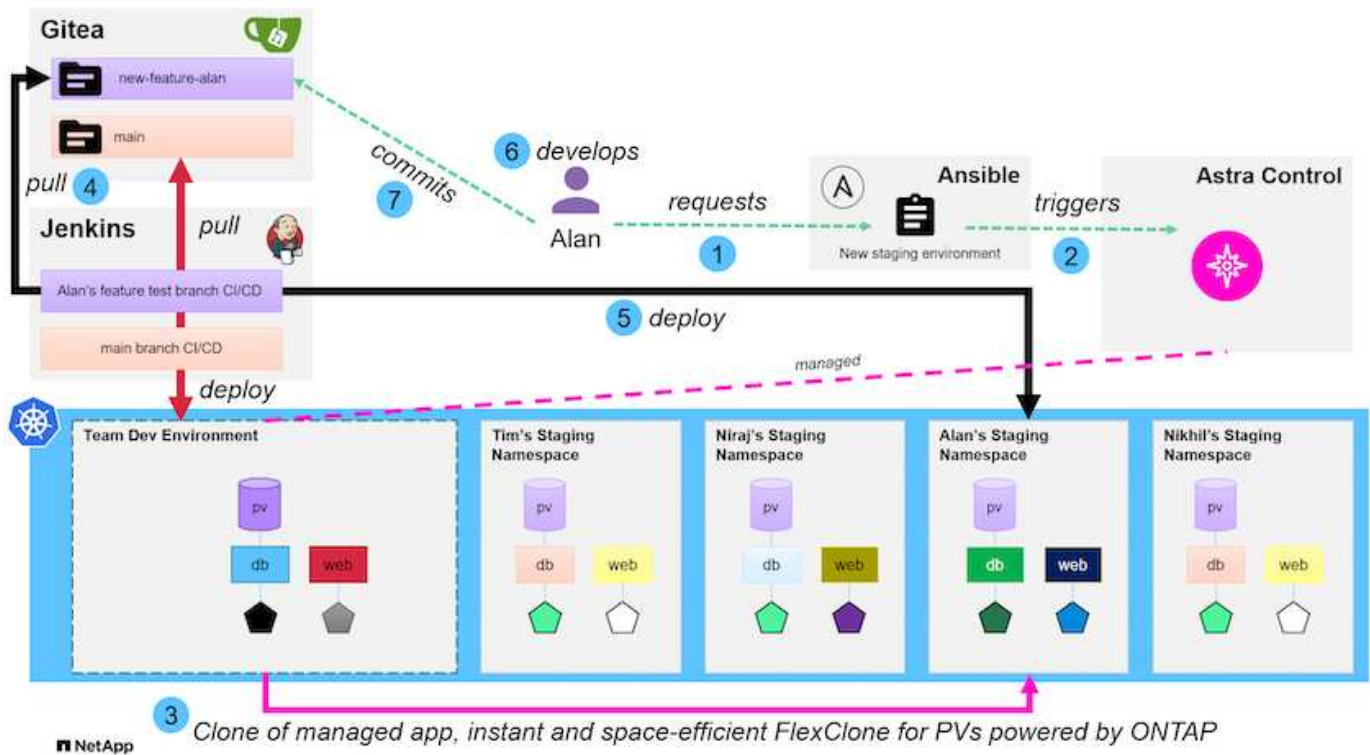
若要深入瞭解NetApp FlexClone技術、請造訪網頁：["NetApp文件"](#)。

## 先決條件

1. 支援的Kubernetes Distribution、例如Red Hat OpenShift 4.6.8+、Rancher 2.5+或Kubernetes 1.19+。
2. NetApp Astra Control Center 21：12+。
3. NetApp ONTAP 的NetApp功能系統、其儲存後端是透過NetApp Astra Trident設定的。
4. Ansible 2.9+。
5. 您想要在NetApp Astra Control中複製為託管應用程式的環境範本。

## 使用案例簡介

在此使用案例中、我們會將類似下列工作流程的內容視覺化：



1. 使用者執行Ansible教戰手冊、以建立新的接移環境。
2. Ansible使用URI-API模組呼叫Astra Control來執行複製作業。
3. Astra Control會在預先配置的範本環境中執行複製作業、進而建立新的託管應用程式。



此環境可以是開發中的單一獨立應用程式、也可以是Jenkins CI/CD管道等整個開發環境。

4. 然後、使用者會從Gitea等線上儲存庫、將其程式碼版本擷取到複製的開發環境中。
5. 新版應用程式由NetApp Astra Control部署及管理。

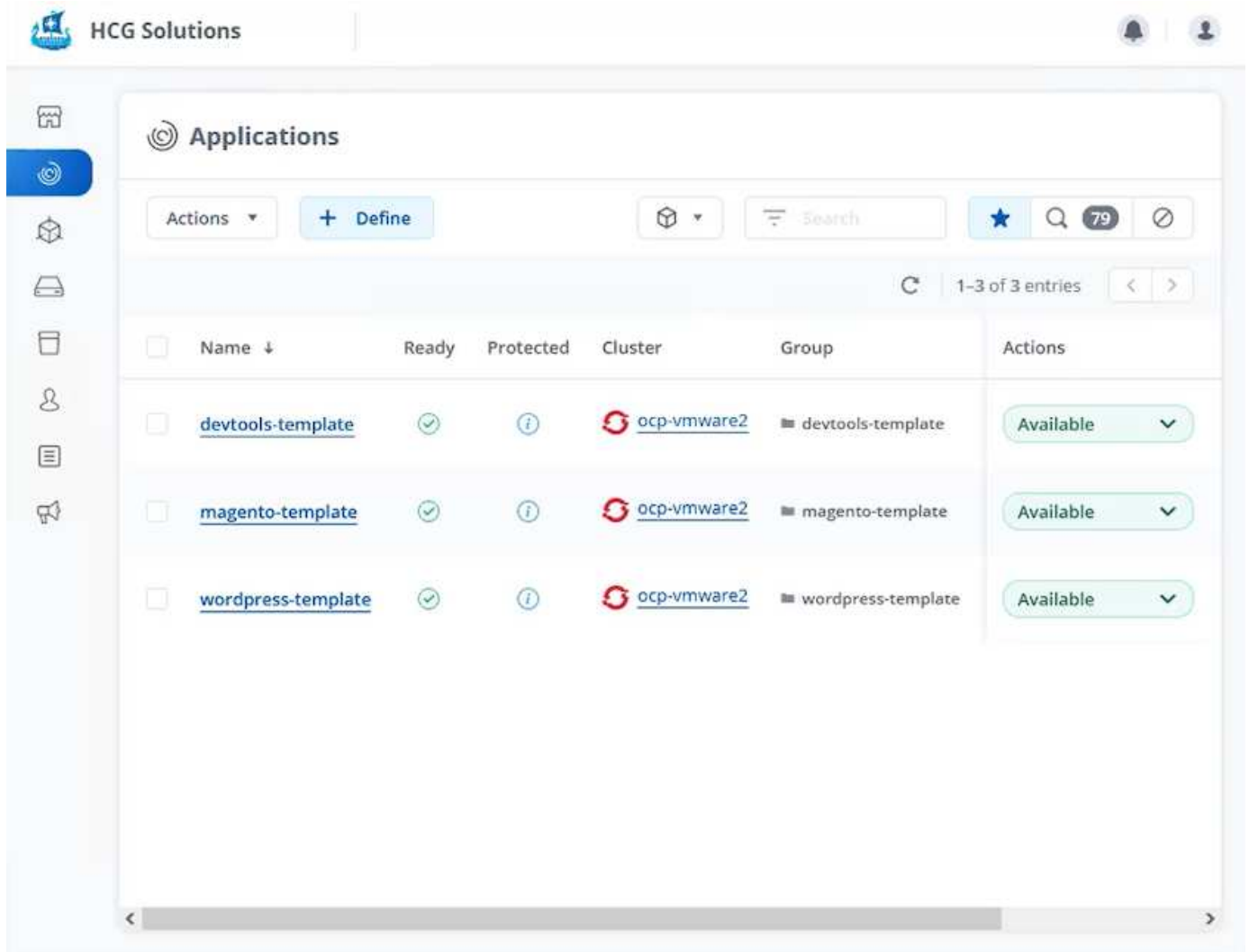


這兩種程序都可以自動化。

6. 使用者可在此複製環境中開發新程式碼。
7. 當使用者對開發工作感到滿意時、他們可以將程式碼推回託管儲存庫。

此處提供的使用案例取決於您想要複製的特定環境或應用程式是否有黃金範本。在我們的環境中、我們建立了三個範本、一個用於WordPress部署、一個用於Magento部署、另一個用於Jenkins CI/CD環境、以及我們標題為DevTools的Gitea。





每個環境都由NetApp Astra控制管理、其中持續的Volume目前儲存在NetApp ONTAP 的NetApp解決方案儲存系統上、並採用NetApp Astra Trident提供的NFS後端。

## 使用案例驗證

1. 複製NetApp解決方案工程團隊提供的Ansible工具組、其中包括複製角色和應用程式更新方針。

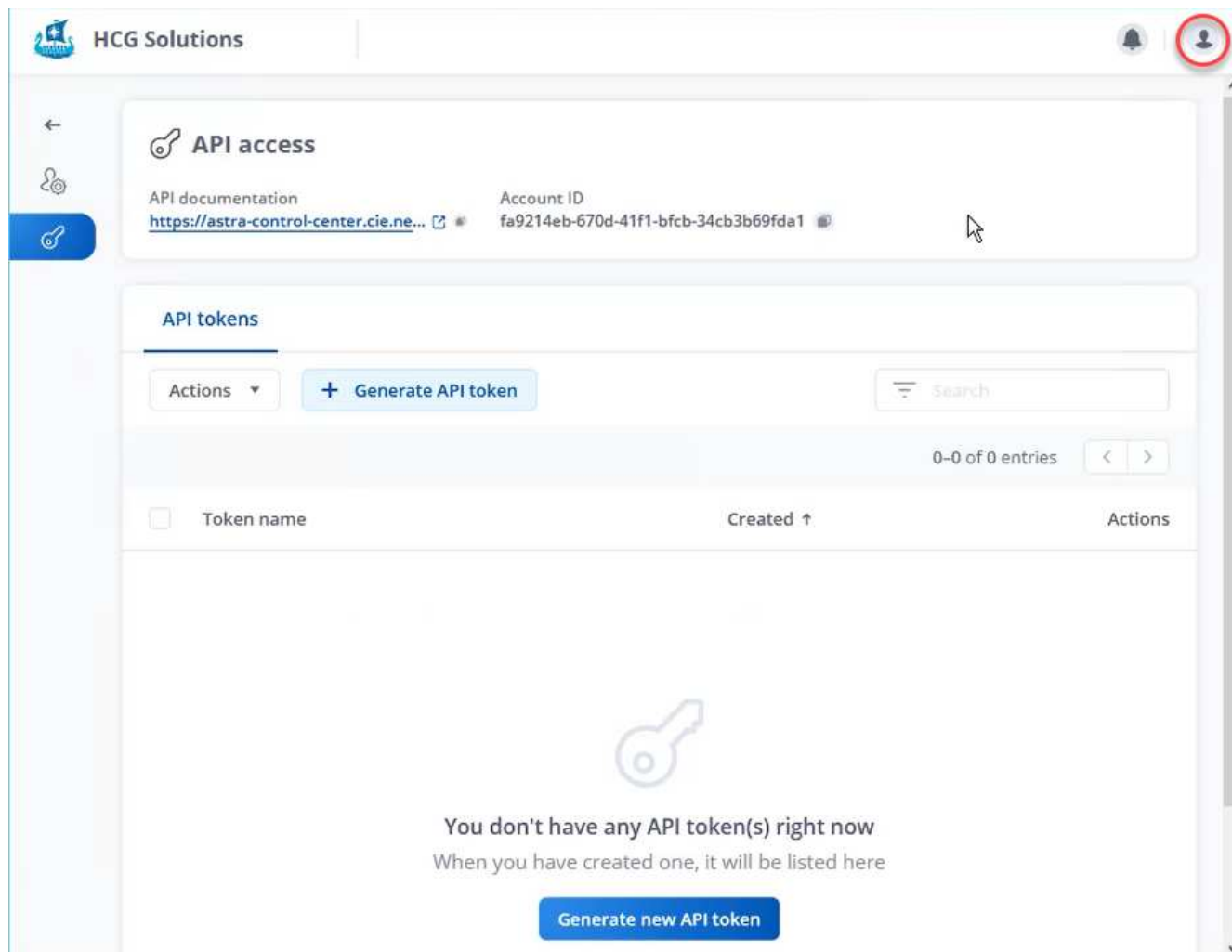
```
[netapp-user@rhel7 ~]$ git clone https://github.com/NetApp-Automation/na_astra_control_suite.git
[netapp-user@rhel7 ~]$ cd na_astra_control_suite
```

2. 編輯「vars/clone\_vars.ml」、並填入符合Astra Control環境的全域值。

```
astra_control_fqdn: astra-control-center.example.com
astra_control_account_id: "xxxx-xxxx-xxxx-xxxx-xxxx"
astra_control_api_token: "xxxxx"
```



您需要填寫的全域環境值可在「API存取」功能表下的NetApp Astra Control中的使用者設定檔圖示下找到。



3. 完成全域變數之後、您可以選擇要複製的特定應用程式值。若要將devtools環境複製到名為「Alan devtools/」的個人環境、您可以執行下列動作：

```
clone_details:
- clone_name: alan-devtools
  destination_namespace: alan-dev-namespace
  source_cluster_name: ocp-vmware2
  destination_cluster_name: ocp-vmware2
  source_application_name: devtools-template
```



若要在複製程序中善用NetApp FlexClone技術、「shrc叢集」和「最大叢集」必須相同。

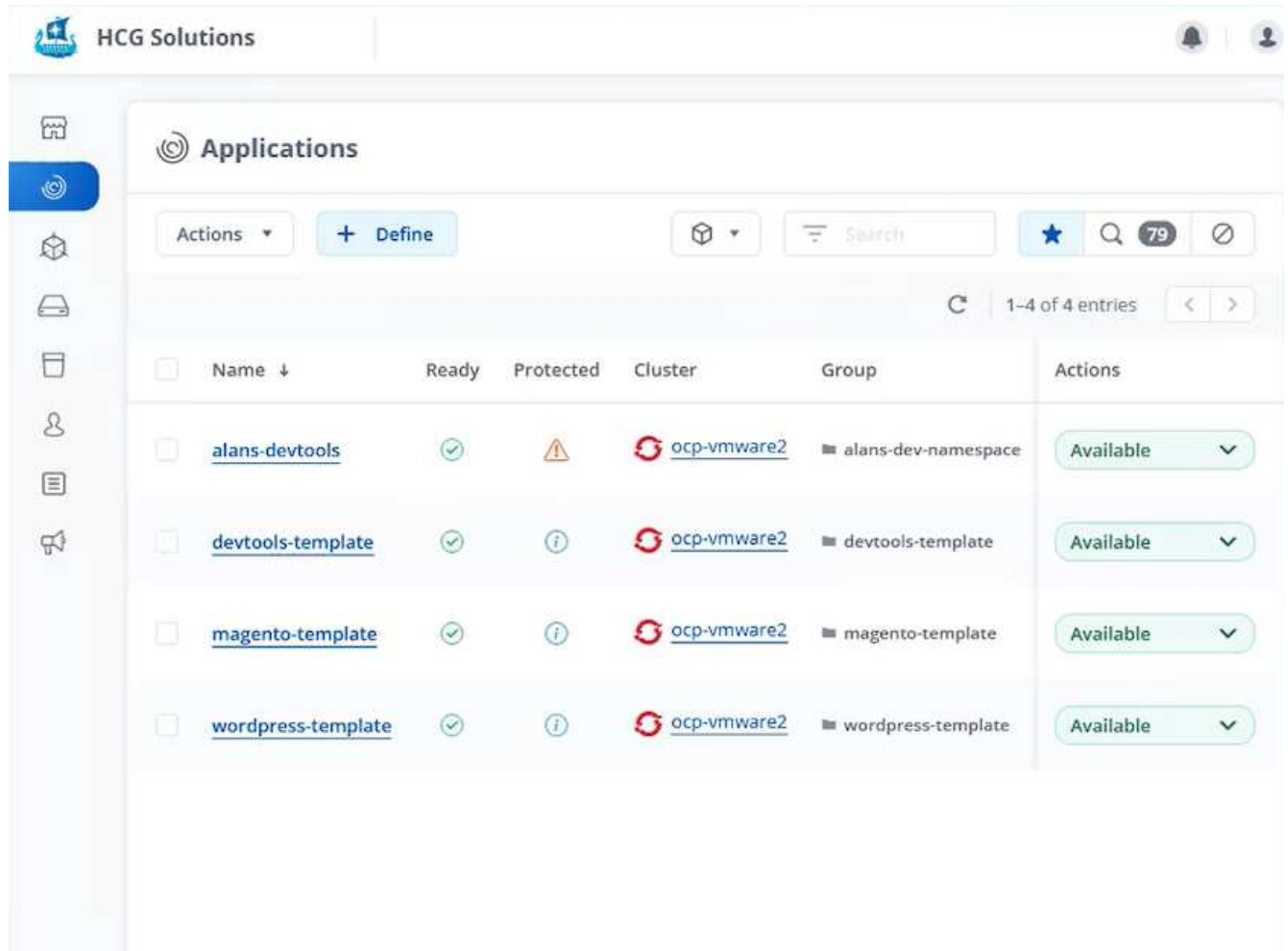
4. 您現在可以執行方針來複製應用程式。

```
[netapp-user@rhel7 na_astra_control_suite]$ ansible-playbook -K clone_app_playbook.yml]
```



撰寫的方針書必須由root使用者執行、或是由能夠透過Sudo程序傳遞「-K」引數而向上呈報的人執行。

5. 當實戰手冊完成執行時、複製的應用程式會顯示為Astra Control Center主控台可用的應用程式。



6. 然後使用者可以登入部署應用程式的Kubernetes環境、驗證應用程式是否有新的IP位址公開、然後開始開發工作。

如需此使用案例的示範、以及升級應用程式的範例、請參閱 ["請按這裡"](#)。

## 版權資訊

Copyright © 2024 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

## 商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。