Convalide dei casi d'utilizzo

NetApp Solutions

NetApp
April 26, 2024

Sommario

Validazione dei casi d’utilizzo: DevOps con NetApp Astra

I seguenti casi di utilizzo sono stati validati per DevOps con NetApp Astra:

- "Integra la protezione nelle pipeline ci/CD con NetApp Astra Control"
- "Sfrutta Astra Control per facilitare l’analisi post-mortem e ripristinare l’applicazione"
- "Accelerazione dello sviluppo software con NetApp FlexClones"

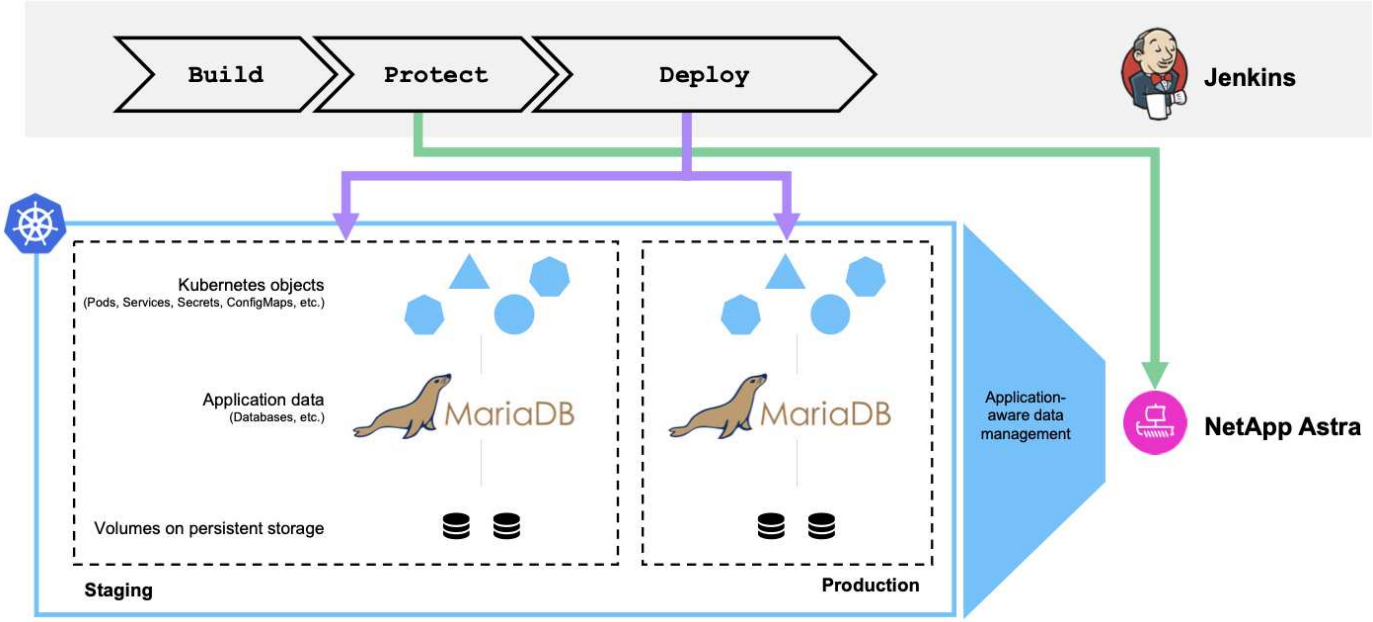
Integra la protezione nelle pipeline ci/CD con NetApp Astra Control

Panoramica

Uno degli utilizzi più comuni dei flussi di lavoro DevOps è l’integrazione continua e le pipeline di implementazione continua (ci/CD) che creano, integrano ed eseguono suite di test automatizzate sulle applicazioni man mano che gli sviluppatori commettono nuovo codice. Gli ingegneri DevOps e gli SREs (Site-Reliability Engineer) dispongono di pipeline dedicate ai vari flussi di lavoro per lo sviluppo di nuove funzionalità, il test di regressione, la correzione di bug, la progettazione della qualità e altre funzioni del processo di sviluppo.

Man mano che i team aumentano il loro livello di automazione, il ritmo del cambiamento per le applicazioni in-produzione può sembrare schiacciante. Pertanto, alcuni team preferiscono proteggere le applicazioni o i servizi in-produzione. Oltre a proteggere il codice e le immagini dei container, desiderano proteggere lo stato dell’applicazione, i dati di configurazione (come gli oggetti Kubernetes e le risorse associate all’applicazione) e i dati persistenti di un’applicazione.

In questo caso di utilizzo, analizziamo più da vicino una pipeline di promozione-produzione che implementa una nuova versione di un’applicazione: Prima in un ambiente di staging e poi in un ambiente di produzione. Questo esempio si applica allo stesso modo ai principali cloud pubblici e anche a un ambiente on-premise. Anche se mostriamo l’implementazione di una versione dell’applicazione, la pipeline può essere utilizzata anche con altre strategie, come l’implementazione blu/verde o canary. Come parte della pipeline ci/CD, proteggeremo l’applicazione creando un backup completo dell’applicazione. Un backup applicativo-aware dell’applicazione in-production e dei relativi dati, stato e configurazione può essere utile per numerosi flussi di lavoro DevOps.



L'applicazione utilizzata per la convalida di questo caso di utilizzo è stata "Magento", Una soluzione di e-commerce con front-end basato su web; un'istanza di Elasticsearch per le funzionalità di ricerca e analisi; e un database MariaDB che tiene traccia di tutti i dettagli dell'inventario di acquisto e delle transazioni. Questa applicazione containerizzata è stata installata in un cluster Red Hat OpenShift. Ogni pod dell'applicazione utilizzava volumi persistenti per memorizzare i dati. I volumi persistenti sono stati creati automaticamente da NetApp Astra Trident, lo storage orchestrator per Kubernetes conforme a Container Storage Interface, che consente il provisioning dello storage sui sistemi storage NetApp. Inoltre, per utilizzare le funzionalità di protezione delle applicazioni di Astra Control Center, l'applicazione in questione è stata gestita da Astra Control, che è stata utilizzata per attivare i backup delle applicazioni che memorizzavano lo stato dell'applicazione insieme ai dati contenuti nei volumi persistenti. Abbiamo utilizzato "SDK NetApp Astra Control Python" Per automatizzare il processo di attivazione dei backup delle applicazioni, che è stato poi introdotto in una pipeline ci/CD. Questa pipeline è stata creata ed eseguita utilizzando un popolare tool ci/CD chiamato ["Jenkins"] per automatizzare il flusso di creazione, protezione e implementazione dell'applicazione.

Analizziamo i prerequisiti e la procedura per introdurre la protezione in una pipeline ci/CD.

Prerequisiti per la convalida del caso d'utilizzo

I seguenti strumenti o piattaforme sono stati implementati e configurati come prerequisiti:

1. Red Hat OpenShift Container Platform
2. NetApp Astra Trident installato su OpenShift con un sistema ONTAP di back-end configurato
3. Uno storageclass predefinito configurato, che punta a un backend NetApp ONTAP
4. NetApp Astra Control Center installato su un cluster OpenShift
5. Cluster OpenShift aggiunto come cluster gestito ad Astra Control Center
6. Jenkins è stato installato su un cluster OpenShift e configurato con un nodo Agent su cui è installato un motore Docker

Installazione dell'applicazione

Iniziamo con l'installazione iniziale dell'applicazione negli ambienti di staging e produzione. Ai fini di questo caso d'utilizzo, questo passaggio è un prerequisito, quindi viene eseguito manualmente. La pipeline ci/CD

viene utilizzata per i flussi di lavoro di creazione e implementazione successivi come risultato delle nuove versioni dell'applicazione.

L'ambiente di produzione in questo caso di utilizzo è uno spazio dei nomi chiamato `magento-prod`e il corrispondente ambiente di staging è uno spazio dei nomi chiamato `magento-staging` Configurato sul cluster Red Hat OpenShift. Per installare l'applicazione, attenersi alla seguente procedura:

1. Installare l'applicazione Magento utilizzando bitnami Helm Chart nell'ambiente di produzione. Utilizziamo RWX PVS per i pod Magento e MariaDB.

```
[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 Helm Chart richiede un servizio LoadBalancer per esporre il servizio Magento GUI. Abbiamo utilizzato "MetalLB" per fornire un servizio di bilanciamento del carico on-premise in questo esempio.

2. Dopo alcuni minuti, verificare che tutti i pod e i servizi siano in esecuzione.

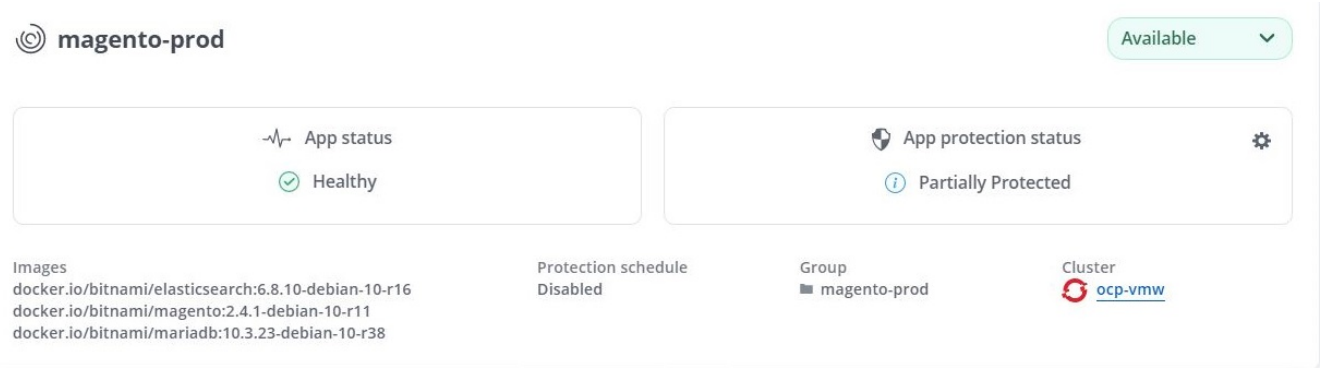
```
[netapp-user@rhel7 na_astra_control_suite]$ oc get pods -n magento-prod
NAME                                                    READY   STATUS
RESTARTS   AGE
magento-9d658fd96-qrxmt                                 1/1     Running
0          49m
magento-elasticsearch-coordinating-only-69869cc5-768rm   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. Ripetere la stessa procedura per l'ambiente di staging.

Gestire l'applicazione Magento in Astra Control Center

1. Accedere ad applicazioni e selezionare la scheda applicazioni rilevate.
2. Fare clic sui puntini di sospensione dell'applicazione Magento nell'ambiente di produzione (`magento-prod`), quindi fare clic su Manage (Gestisci).
3. L'applicazione Magento è ora gestita da Astra Control Center. Tutte le operazioni supportate da Astra

Control possono essere eseguite sull'applicazione. Prendere nota anche della versione dell'applicazione.

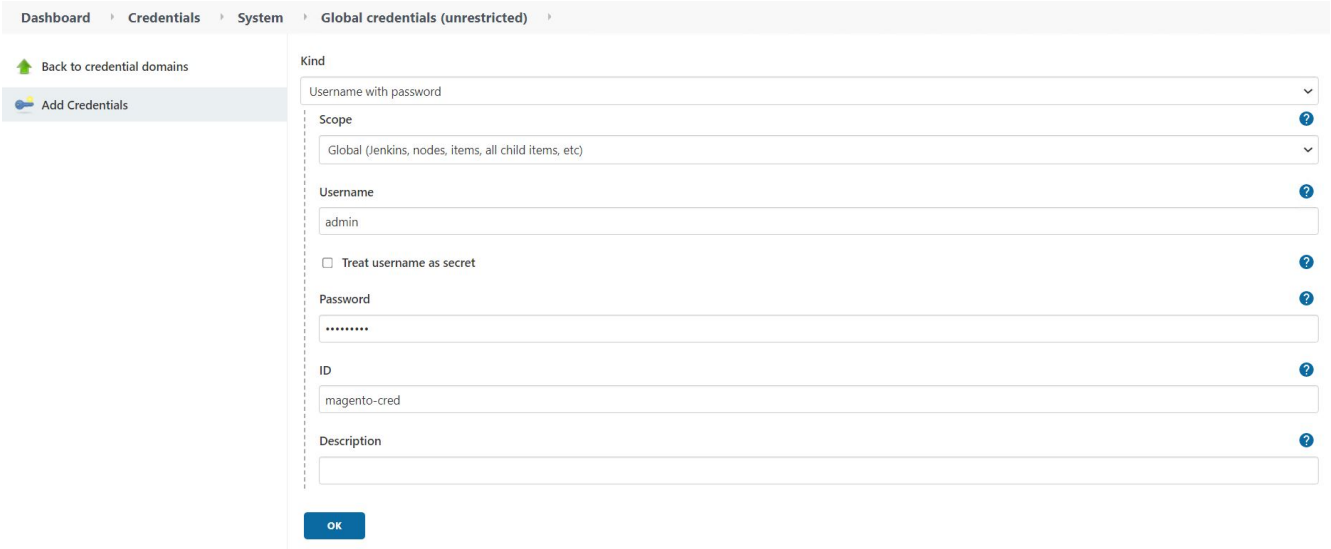


4. Ripetere i passaggi per la gestione dell'applicazione Magento nell'ambiente di staging (`magento-staging`).

Pipeline ci/CD con protezione integrata

Quando lavoriamo con le nuove versioni delle applicazioni, utilizziamo una pipeline ci/CD per creare l'immagine container, eseguire backup degli ambienti di staging e produzione, implementare la nuova versione dell'applicazione nell'ambiente di staging, attendere l'approvazione per la promozione in produzione, quindi, implementare la nuova versione dell'applicazione nell'ambiente di produzione. Per utilizzare una pipeline ci/CD, attenersi alla seguente procedura:

1. Accedi a Jenkins e crea le credenziali richieste: Una per Magento creds, una per MariaDB admin creds e la terza per MariaDB root creds.
2. Accedere a Manage Jenkins > Manage Credentials (Gestisci Jenkins > Gestisci credenziali) e fare clic sul dominio appropriato.
3. Fare clic su Add Credentials (Aggiungi credenziali) e impostare il tipo su Username (Nome utente) con password e ambito impostati su Global (Globale). Immettere il nome utente, la password e un ID per le credenziali, quindi fare clic su OK.

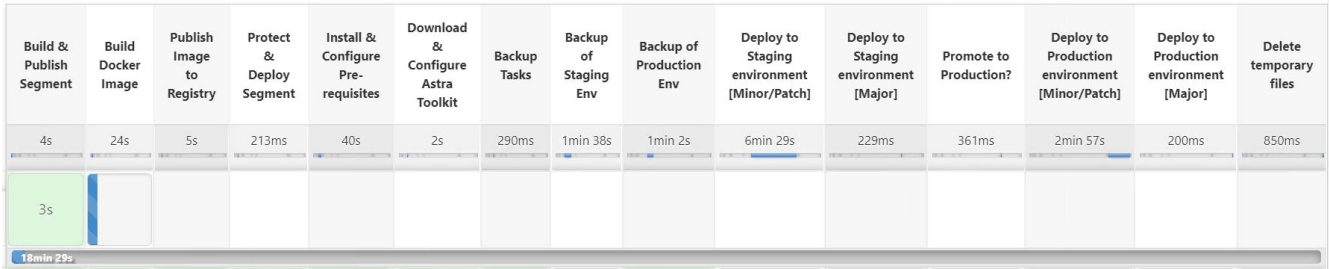


4. Ripetere la stessa procedura per le altre due credenziali.
5. Tornare alla dashboard, creare una pipeline facendo clic su New Item (nuovo elemento), quindi fare clic su Pipeline (pipeline).

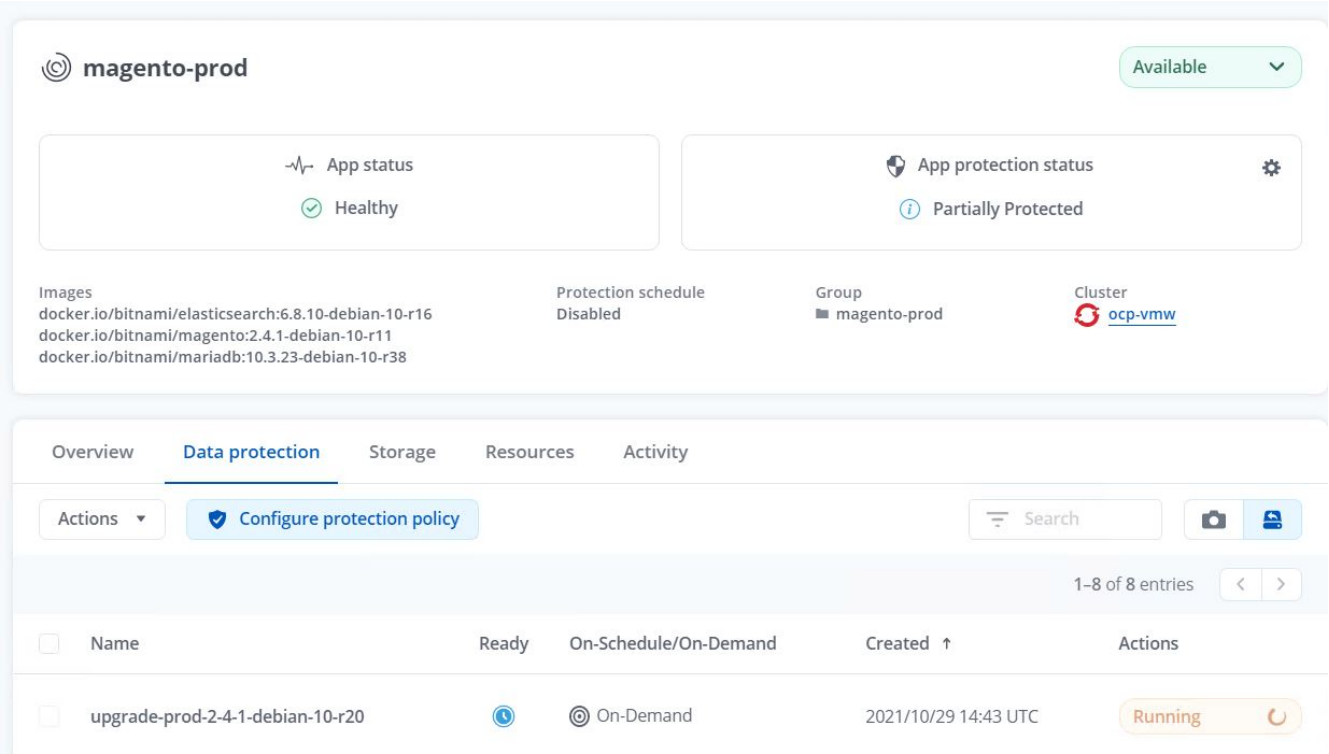
6. Copiare la pipeline dal file Jenkinsfile "qui".
7. Incollare la pipeline nella sezione della pipeline Jenkins, quindi fare clic su Save (Salva).
8. Compilare i parametri della pipeline Jenkins con i relativi dettagli, tra cui la versione del grafico Helm, la versione dell'applicazione Magento a cui si desidera eseguire l'aggiornamento, la versione del toolkit Astra, l'FQDN di Astra Control Center, il token API e il relativo ID istanza. Specificare il registro del docker, lo spazio dei nomi e l'IP Magento degli ambienti di produzione e di staging e specificare anche gli ID delle credenziali create.

```
MAGENTO_VERSION = '2.4.1-debian-10-r14'
CHART_VERSION = '14'
RELEASE_TYPE = 'MINOR'
ASTRA_TOOLKIT_VERSION = '2.0.2'
ASTRA_API_TOKEN = 'xxxxxxxx'
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')
```

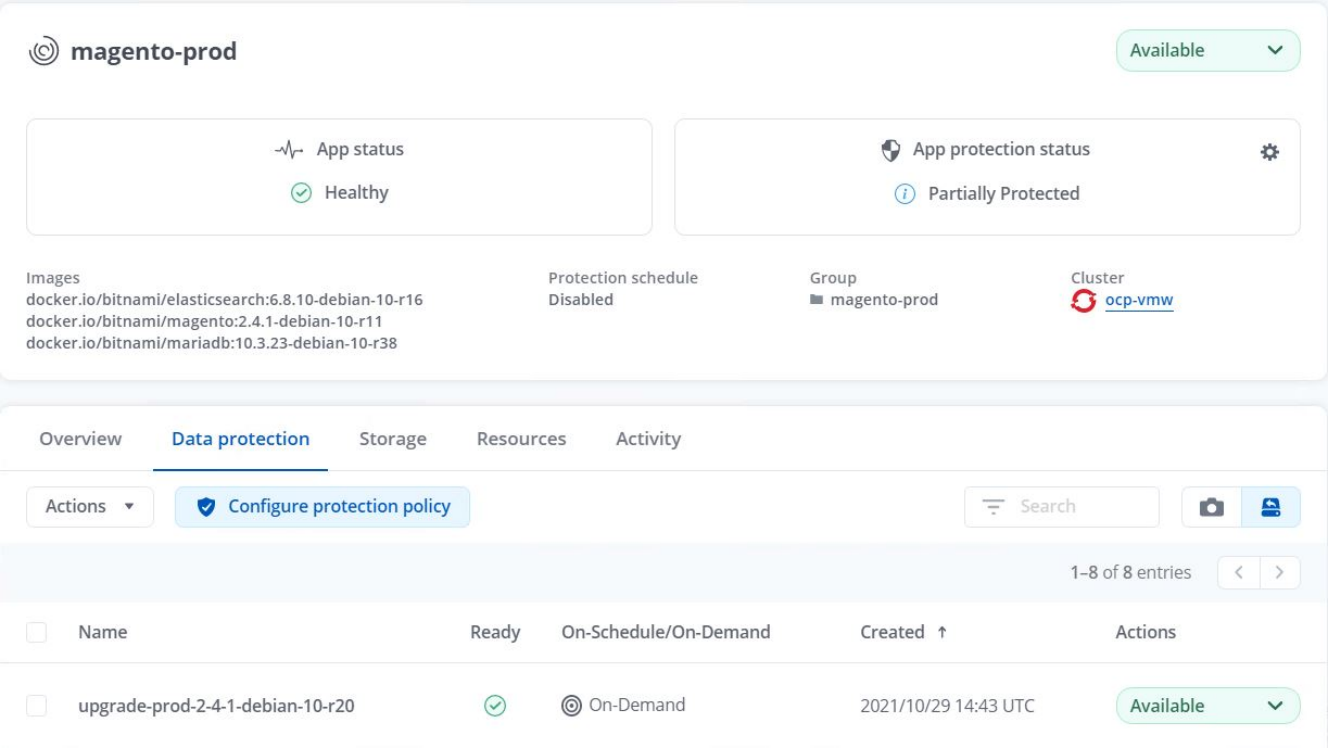
9. Fare clic su Crea ora. La pipeline inizia a essere eseguita e procede attraverso le fasi. L'immagine dell'applicazione viene creata e caricata nel registro del container.



10. I backup dell'applicazione vengono avviati tramite Astra Control.



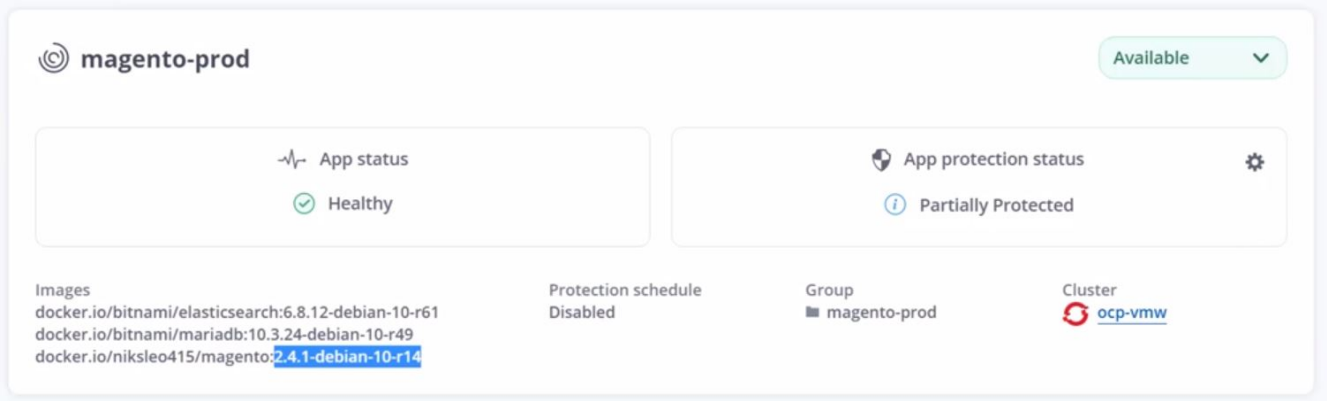
11. Una volta completate le fasi di backup, verificare i backup da Astra Control Center.



12. La nuova versione dell’applicazione viene quindi distribuita nell’ambiente di staging.

13. Al termine di questa fase, il programma attende che l'utente approvi la distribuzione in produzione. In questa fase, supponiamo che il team di QA esegua alcuni test manuali e approvi la produzione. Fare clic su Approve (approva) per distribuire la nuova versione dell'applicazione nell'ambiente di produzione.

14. Verificare che anche l'applicazione di produzione sia aggiornata alla versione desiderata.



Come parte della pipeline ci/CD, abbiamo dimostrato la capacità di proteggere l'applicazione creando un backup completo e integrato con l'applicazione. Poiché il backup dell'intera applicazione è stato eseguito nell'ambito della pipeline di promozione-produzione, puoi sentirti più sicuro delle implementazioni altamente automatizzate delle applicazioni. Questo backup integrato con l'applicazione contenente i dati, lo stato e la configurazione dell'applicazione può essere utile per numerosi flussi di lavoro DevOps. Un importante flusso di lavoro potrebbe essere il ripristino della versione precedente dell'applicazione in caso di problemi imprevisti.

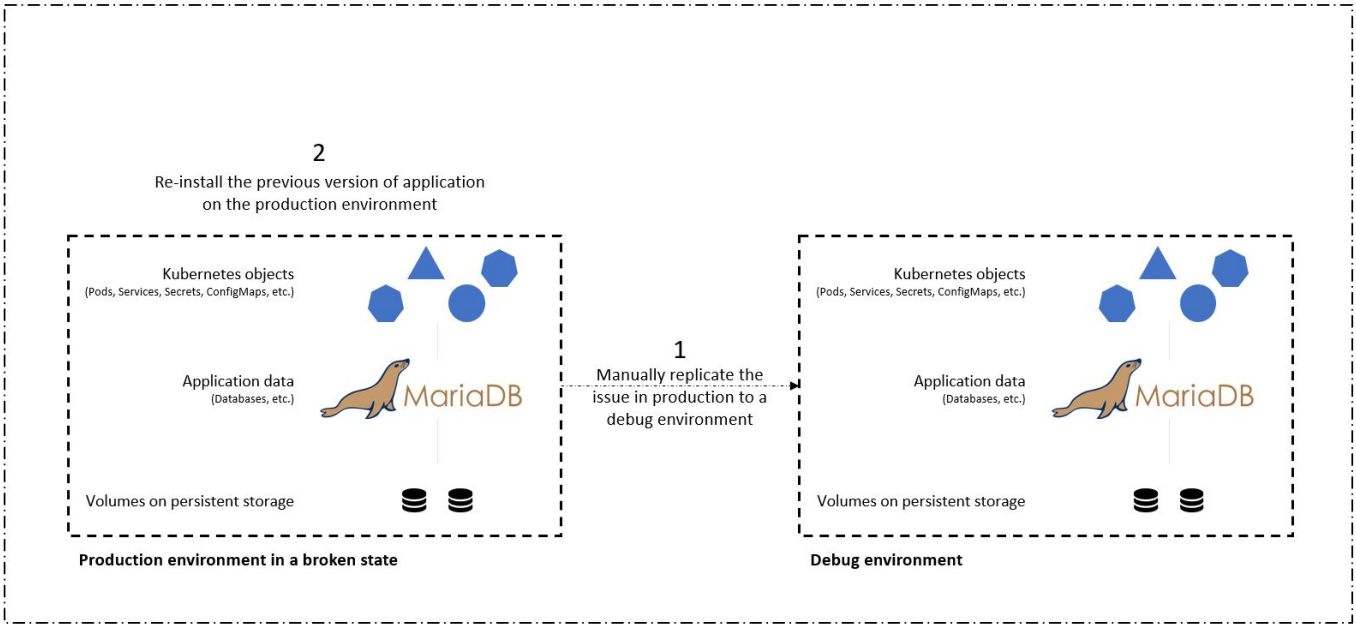
Anche se abbiamo dimostrato un workflow ci/CD attraverso lo strumento Jenkins, il concetto può essere estrapolato in modo semplice ed efficiente a diversi strumenti e strategie. Per vedere questo caso d'utilizzo in azione, guarda il video "qui".

Utilizza Astra Control per facilitare l'analisi post-mortem e ripristinare l'applicazione

Panoramica

In "primo caso di utilizzo", Abbiamo dimostrato come utilizzare NetApp Astra Control Center per proteggere le tue applicazioni in Kubernetes. In questa sezione viene descritto come integrare i backup delle applicazioni tramite Astra Control direttamente nel flusso di lavoro di sviluppo utilizzando l'SDK Python del toolkit NetApp Astra. Questo approccio consente la protezione degli ambienti di sviluppo e produzione automatizzando i backup on-demand durante il processo di integrazione continua e implementazione continua (ci/CD). Con questo ulteriore livello di protezione dei dati coerente con le applicazioni aggiunto alla pipeline ci/CD e alle applicazioni di produzione, i processi di sviluppo sono sicuri se qualcosa va storto nel processo, il che promuove buone pratiche di business-continuity.

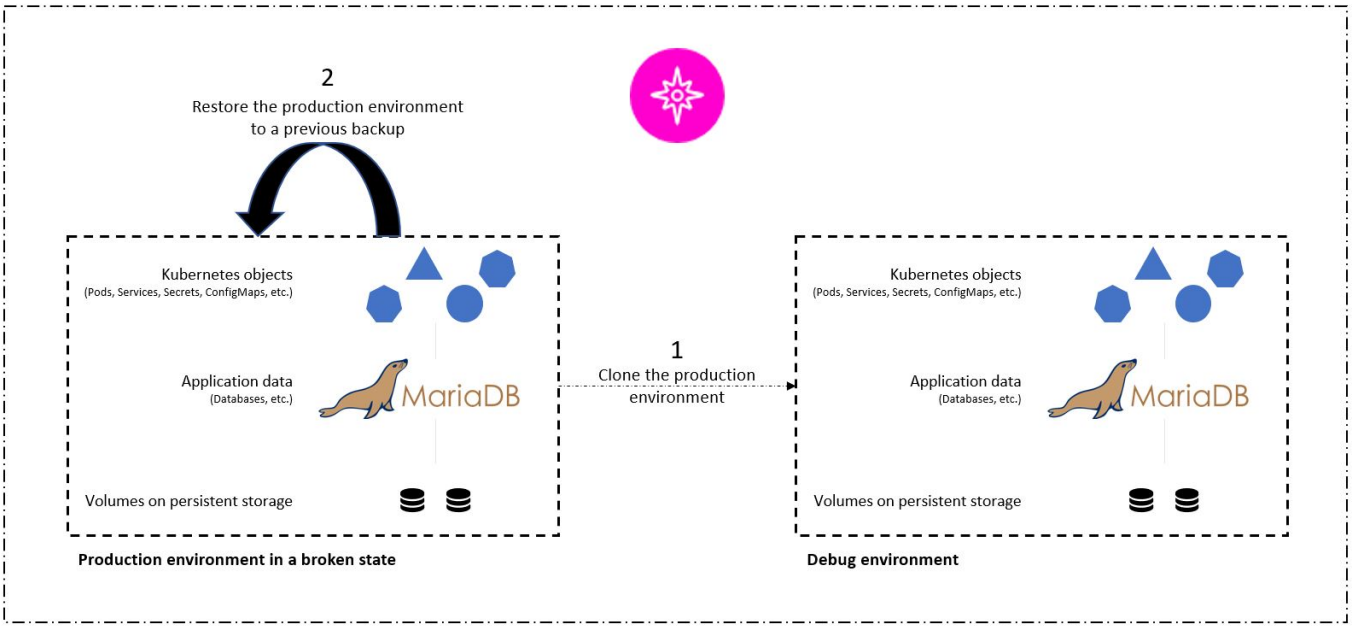
In un workflow tradizionale, dopo aver riscontrato un errore quando l'applicazione viene aggiornata a una nuova versione, il team di sviluppo tentava di risolvere il problema in tempo reale in base ai report di bug forniti dai clienti. In alternativa, al primo segnale di problemi, il team potrebbe tentare di ridistribuire l'applicazione in un ambiente di debug parallelo per portare il processo offline. Potevano ridistribuire una base di codice precedente da una versione precedente in produzione, ripristinando l'applicazione in ordine di lavoro.



Anche se questo approccio funziona, il team dovrebbe assicurarsi che lo stato dell'applicazione di produzione guasta corrisponda a quello della versione vista in produzione quando si sono verificati i problemi. Inoltre, dovrebbero dedicare del tempo alla promozione della build sicuramente funzionante in produzione, recuperando il codice dal repository e ridistribuendo le immagini della macchina per ripristinare l'applicazione a un buon stato di esecuzione. Inoltre, in questo scenario, non abbiamo considerato se il database di produzione stesso fosse corrotto dal codice difettoso. Idealmente, esistono processi di backup separati per i dati del database, ma dobbiamo presumere che siano coerenti con lo stato dell'applicazione così come è stata pubblicata? È qui che i benefici di backup, ripristini e cloni stateful e coerenti con l'applicazione con Astra Control dimostrano davvero il loro valore.

Innanzitutto, possiamo utilizzare Astra Control per facilitare l'analisi post-mortem sullo stato dell'applicazione. Per farlo, cloniamo la versione di produzione buggy in un ambiente di test parallelo in modo coerente con l'applicazione. La messa da parte di questo ambiente nello stato di bug-ridden ci consente di risolvere il problema in tempo reale.

Inoltre, Astra Control supporta la funzionalità di ripristino in-place che consente di ripristinare l'applicazione di produzione a un ultimo backup accettabile (che ha preceduto la versione del codice interessata). La versione ripristinata assume la posizione dell'applicazione di produzione precedente, in modo coerente e stateful con l'applicazione, incluso l'IP di ingresso precedentemente assegnato. Di conseguenza, i clienti che accedono al front-end non sarebbero a conoscenza della transizione alla versione di backup.



Prerequisiti per la convalida del caso d'utilizzo

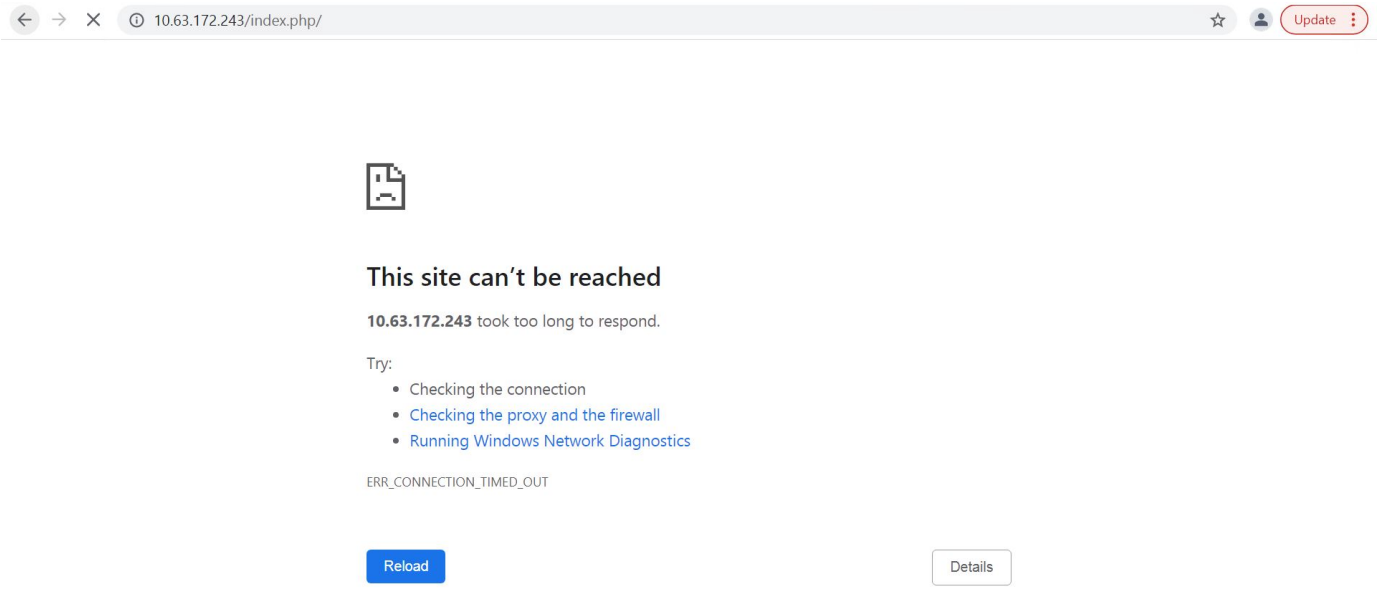
I seguenti strumenti o piattaforme sono stati implementati e configurati come prerequisiti:

- Red Hat OpenShift Container Platform.
- NetApp Astra Trident installato su OpenShift con un backend configurato su un sistema NetApp ONTAP.
- Uno storageclass predefinito configurato, che punta a un backend NetApp ONTAP.
- NetApp Astra Control Center installato su un cluster OpenShift.
- Cluster OpenShift aggiunto come cluster gestito ad Astra Control Center.
- Jenkins installato su un cluster OpenShift.
- Applicazione Magento installata nell'ambiente di produzione. In questo caso di utilizzo, l'ambiente di produzione è uno spazio dei nomi chiamato "lento-prod" in un cluster Red Hat OpenShift.
- Applicazione di produzione gestita da Astra Control Center.
- Backup sicuramente funzionanti dell'applicazione di produzione acquisita con Astra Control.

Clonare e ripristinare la pipeline

Considerando che l'applicazione è stata aggiornata a una nuova versione, l'applicazione nell'ambiente di produzione (`magento-prod`) non si comporta come previsto dopo l'aggiornamento. Supponiamo che i dati restituiti dalle query front-end non corrispondano alla richiesta o che il database sia stato effettivamente

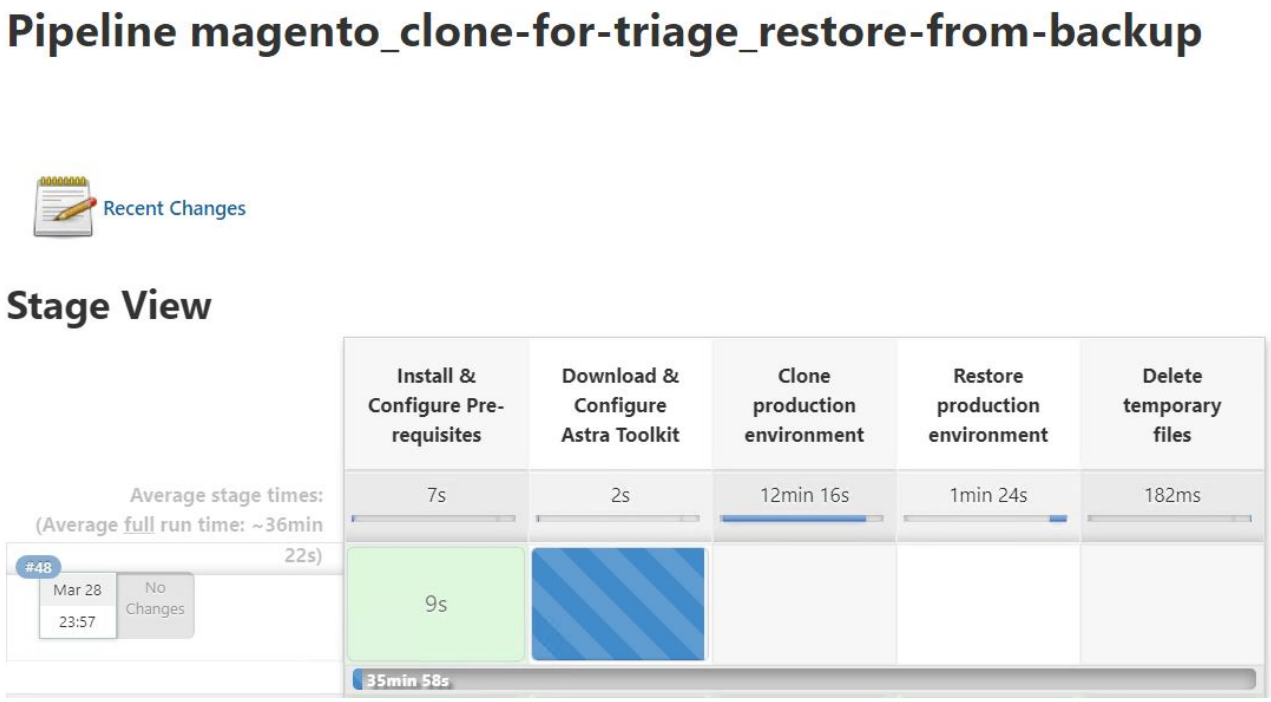
danneggiato. Per clonare e ripristinare la pipeline, attenersi alla seguente procedura:



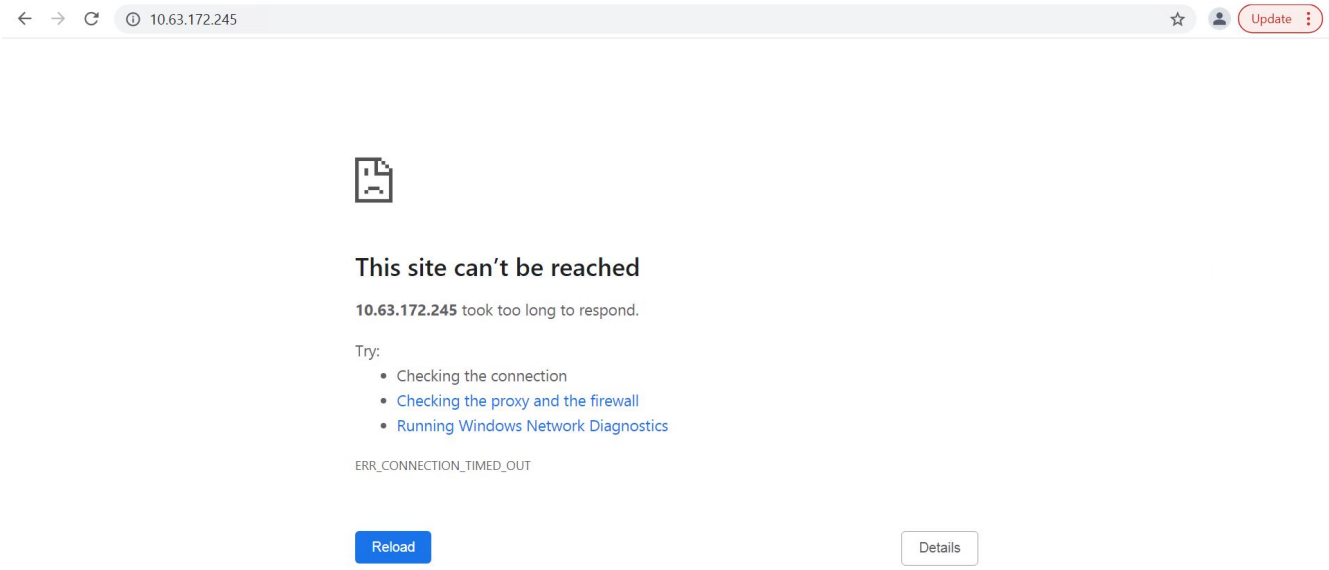
1. Accedere a Jenkins e creare una pipeline facendo clic su New Item (nuovo elemento), quindi su Pipeline (pipeline).
2. Copiare la pipeline dal file Jenkinsfile "qui".
3. Incollare la pipeline nella sezione della pipeline Jenkins, quindi fare clic su Save (Salva).
4. Compilare i parametri della pipeline Jenkins con i relativi dettagli, come la versione corrente dell'applicazione Magento in produzione, l'FQDN di Astra Control Center, il token API, l'ID dell'istanza e il nome dell'applicazione o lo spazio dei nomi degli ambienti di produzione e debug, nonché i nomi dei cluster di origine e destinazione. Ai fini di questo caso d'utilizzo, l'ambiente di produzione è uno spazio dei nomi chiamato 'lento-prod' e l'ambiente di debug è uno spazio dei nomi chiamato 'lento-debug' configurato su un cluster Red Hat OpenShift.

```
MAGENTO_VERSION = '2.4.1-debian-10-r14'
ASTRA_TOOLKIT_VERSION = '2.0.2'
ASTRA_API_TOKEN = 'xxxxx'
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'
```

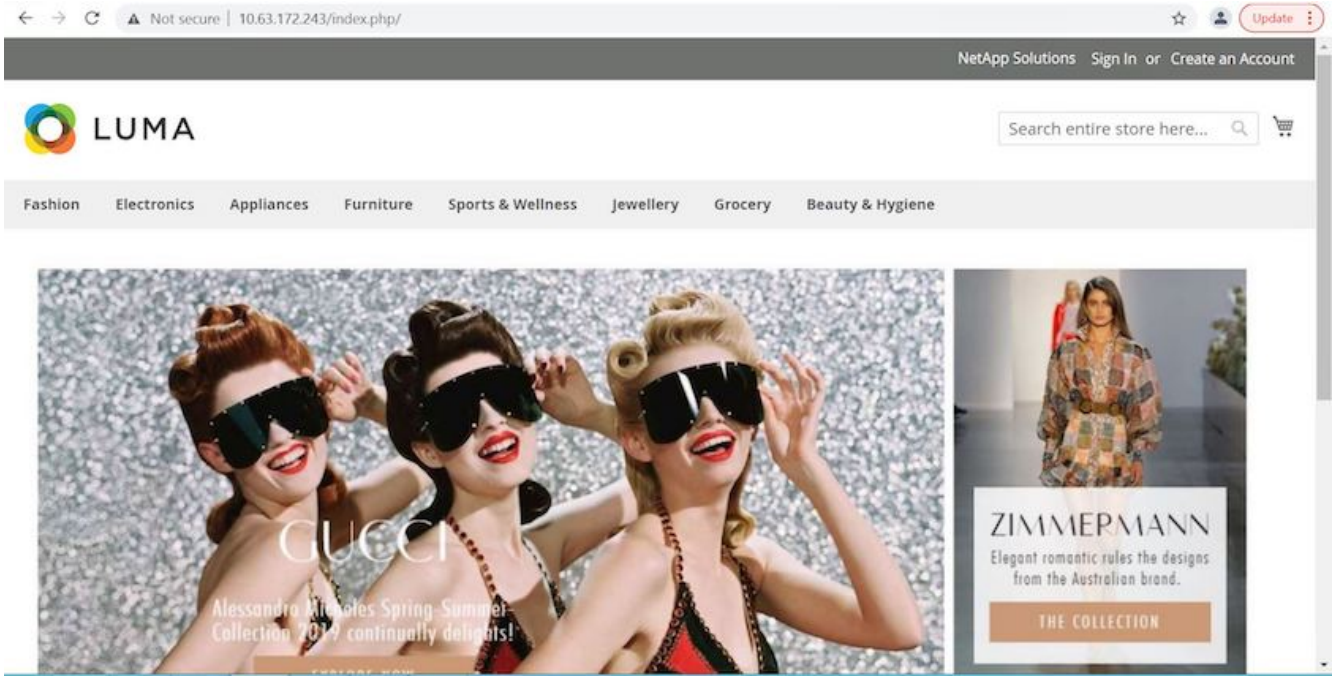
5. Fare clic su Crea ora. La pipeline inizia a essere eseguita e procede attraverso le fasi. L'applicazione viene prima clonata nello stato corrente in un ambiente di debug e quindi ripristinata al backup funzionante.



6. Verificare che l’applicazione clonata sia la versione contenente bug.



7. Verificare che l’ambiente di produzione sia ripristinato a un backup funzionante e che l’applicazione in produzione funzioni come previsto.



Queste due operazioni in tandem accelerano il ritorno alle normali operazioni di business. Per vedere questo caso d'utilizzo in azione, guarda il video "qui".

Accelerazione dello sviluppo software con la tecnologia FlexClone di NetApp

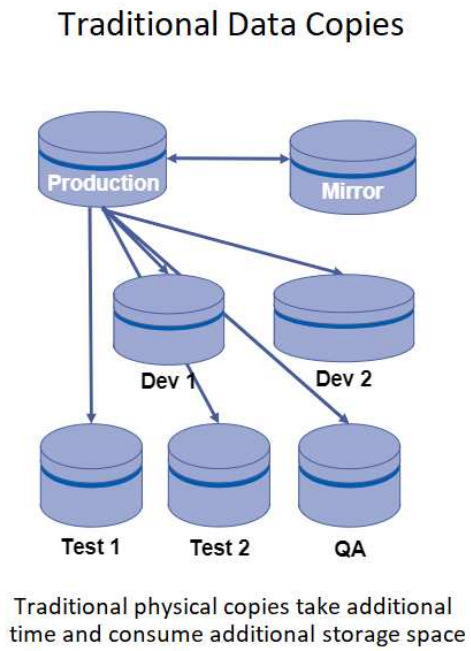
Panoramica

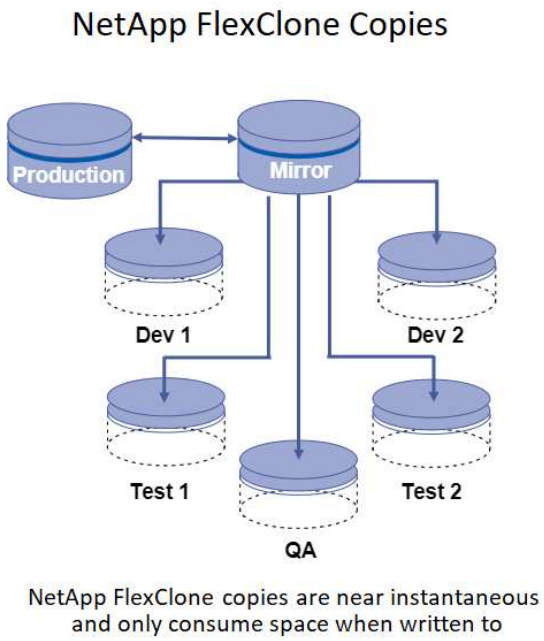
La clonazione di un'applicazione distribuita in un cluster Kubernetes è uno strumento molto utile per gli sviluppatori che desiderano accelerare i propri flussi di lavoro condividendo ambienti con i partner o testando nuove versioni di codice in un ambiente di sviluppo senza interferire con la versione su cui stanno attualmente lavorando. La clonazione stateful e coerente con l'applicazione di un'applicazione Kubernetes è una delle funzionalità principali incluse in NetApp Astra Control, insieme al backup e ripristino delle applicazioni. Come bonus, se un'applicazione viene clonata all'interno dello stesso cluster Kubernetes utilizzando lo stesso backend di storage, Astra Control utilizza per impostazione predefinita la tecnologia NetApp FlexClone per la duplicazione di volumi di dati persistenti, accelerando notevolmente il processo. Accelerando questo processo, l'ambiente clonato viene fornito e disponibile per l'utilizzo in pochi istanti, consentendo agli sviluppatori di riprendere il proprio lavoro con una breve pausa rispetto alla ridistribuzione dell'ambiente di test o sviluppo. Come ulteriore comodità, tutte le funzioni disponibili in NetApp Astra Control possono essere chiamate con un'API, che consente una facile integrazione in framework di automazione come Ansible. Pertanto, gli ambienti possono essere gestiti in tempi ancora più rapidi, perché sono necessarie solo modifiche di lieve entità in un manuale o in un ruolo per iniziare la procedura di cloning.

Che cos'è la tecnologia FlexClone di NetApp?

La tecnologia NetApp FlexClone è una copia scrivibile e point-in-time basata su snapshot di un NetApp FlexVol. Vengono forniti quasi istantaneamente, contengono tutti i dati del volume di origine e non consumano spazio di storage aggiuntivo fino a quando i dati nel nuovo volume non iniziano a divergere dall'origine. Vengono spesso utilizzati in ambienti basati su modelli o di sviluppo quando più copie di dati sono utili per scopi di staging e i sistemi storage dispongono di risorse limitate per il provisioning di questi volumi. Rispetto a un sistema storage tradizionale in cui i dati devono essere copiati più volte, con un conseguente consumo di tempo e spazio di storage significativo, la tecnologia NetApp FlexClone accelera le attività dipendenti dallo

storage.





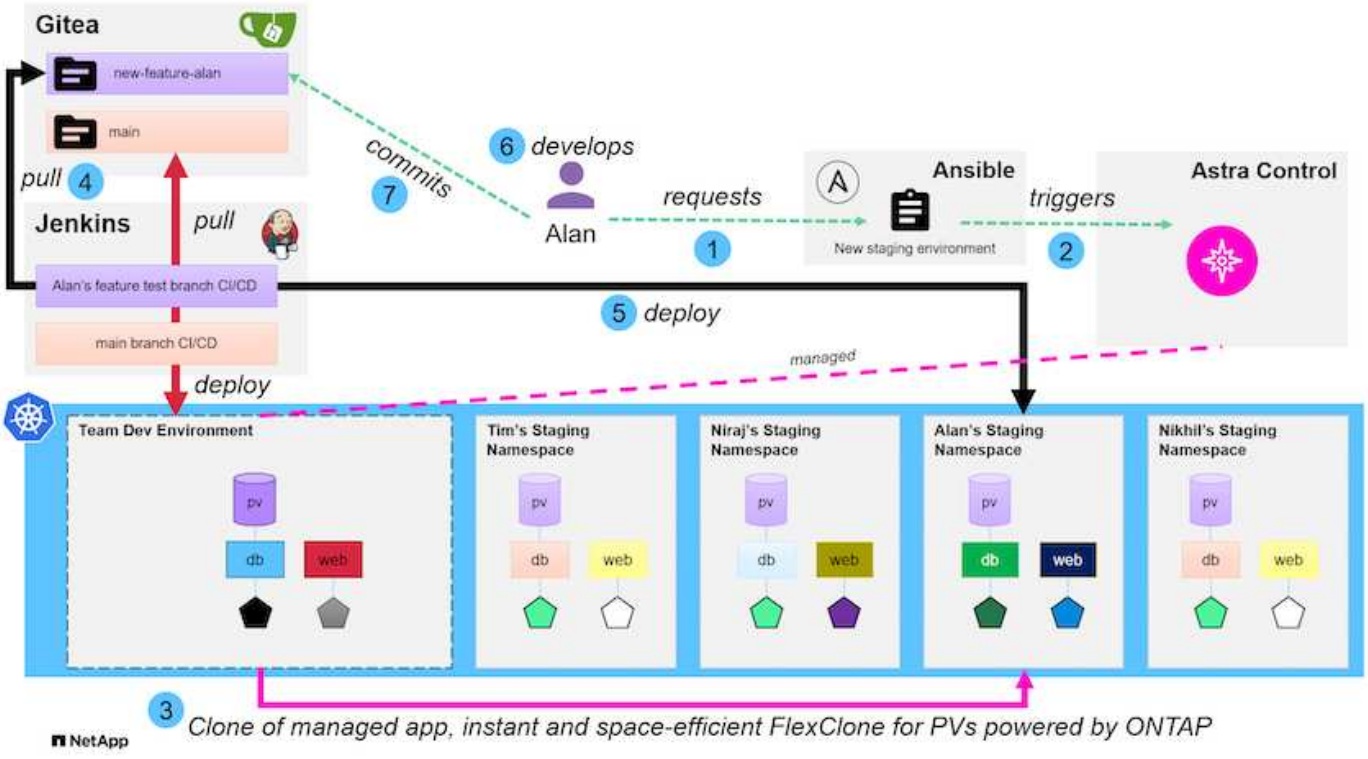
Per ulteriori informazioni sulla tecnologia FlexClone di NetApp, visita la pagina all'indirizzo "Documenti NetApp".

Prerequisiti

1. Una distribuzione Kubernetes supportata, come Red Hat OpenShift 4.6.8+, Rancher 2.5+ o Kubernetes 1.19+.
2. NetApp Astra Control Center 21.12+.
3. Un sistema NetApp ONTAP con un backend di storage configurato tramite NetApp Astra Trident.
4. Ansible 2.9+.
5. Modelli per gli ambienti che si desidera clonare come applicazioni gestite in NetApp Astra Control.

Introduzione al caso d'utilizzo

In questo caso di utilizzo, viene visualizzato un aspetto simile al seguente flusso di lavoro:



1. Un utente esegue il playbook ansible per creare un nuovo ambiente di staging.
2. Ansible utilizza il modulo URI-API per richiamare Astra Control per eseguire l'operazione di cloning.
3. Astra Control esegue un'operazione di cloning su un ambiente modello con provisioning anticipato, creando così una nuova applicazione gestita.

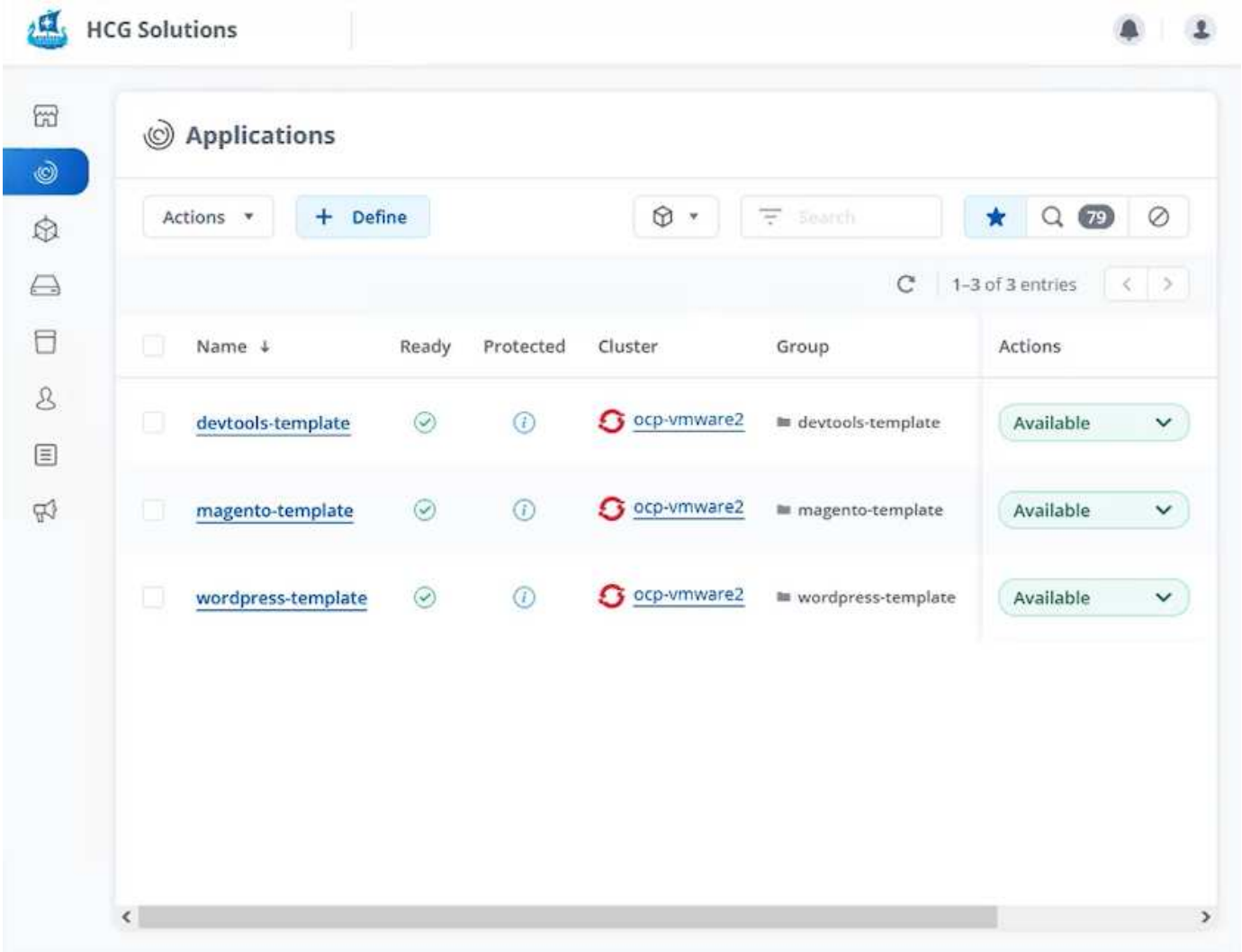
> Questo ambiente può essere una singola applicazione standalone in fase di sviluppo o un intero ambiente di sviluppo come una pipeline Jenkins ci/CD.

4. L'utente quindi estrae una versione del proprio codice nell'ambiente di sviluppo clonato da un repository online come Gitea.
5. La nuova versione dell'applicazione viene implementata e gestita da NetApp Astra Control.

> Entrambi questi processi possono essere automatizzati.

6. L'utente può sviluppare nuovo codice in questo ambiente clonato.
7. Quando l'utente è soddisfatto dei propri sforzi di sviluppo, può reinviare il codice al repository ospitato.

Il caso d'utilizzo qui presentato dipende dall'esistenza di modelli Golden per gli ambienti o le applicazioni che si desidera clonare. Nel nostro ambiente abbiamo creato tre modelli di questo tipo, uno per un'implementazione di WordPress, uno per un'implementazione di Magento e uno per un ambiente ci/CD di Jenkins con Gitea che abbiamo chiamato DevTools.



Ciascuno di questi ambienti è gestito da NetApp Astra Control, con volumi persistenti attualmente memorizzati su un sistema di storage NetApp ONTAP con un backend NFS fornito da NetApp Astra Trident.

Validazione del caso d'utilizzo

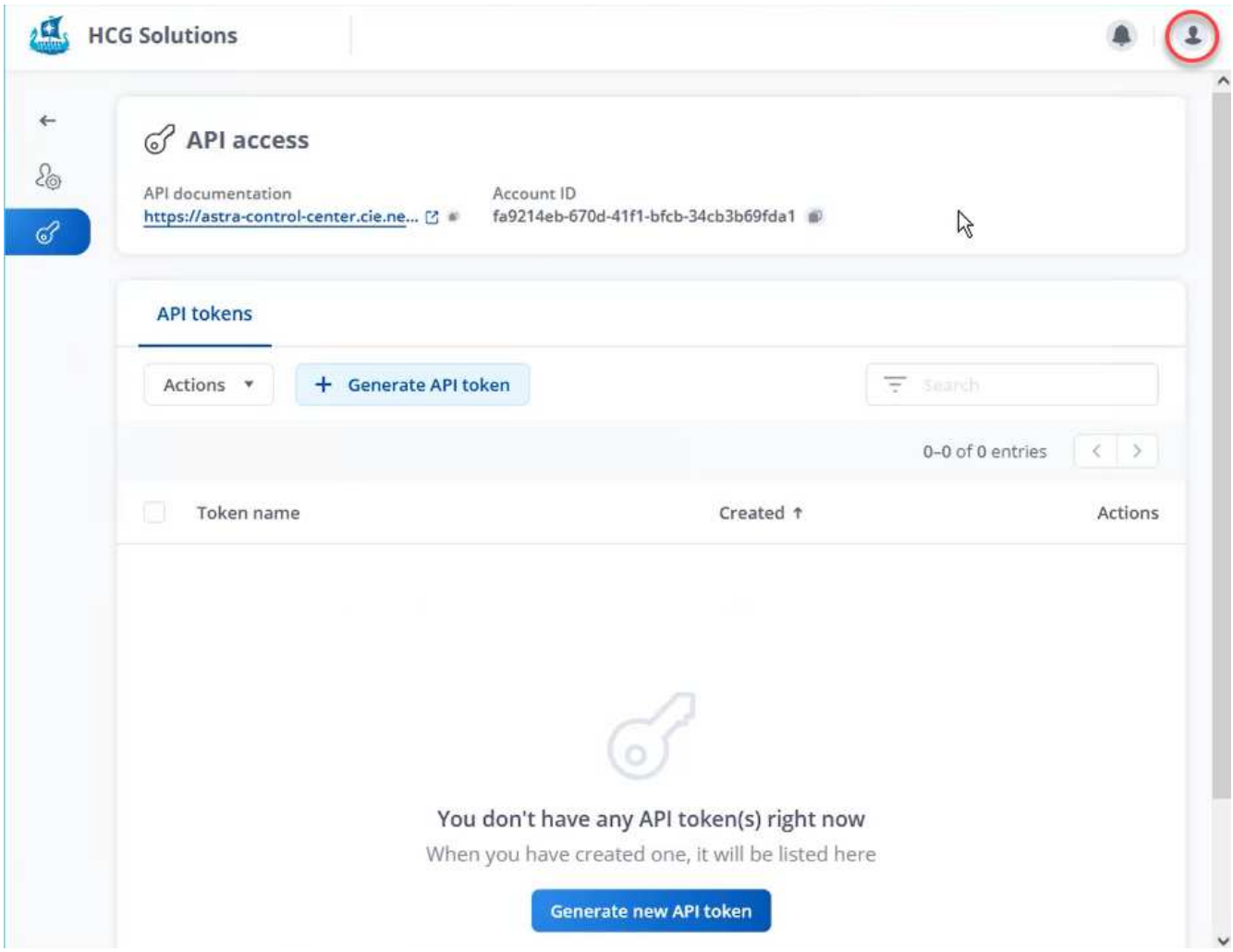
1. Clonare il toolkit ansible fornito dal team NetApp Solutions Engineering, che include il ruolo di cloning e il manuale di aggiornamento dell'applicazione.

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

2. Modifica `vars/clone_vars.yml` E compila i valori globali che si adattano al tuo ambiente Astra Control.

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

> I valori dell'ambiente globale da compilare sono disponibili sotto l'icona del profilo utente in NetApp Astra Control nel menu API Access.



3. Una volta completate le variabili globali, è possibile scegliere i valori per l'applicazione specifica che si desidera clonare. Per clonare l'ambiente devtools in un ambiente personale chiamato `alan-devtools`, eseguire le seguenti operazioni:

```
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
```

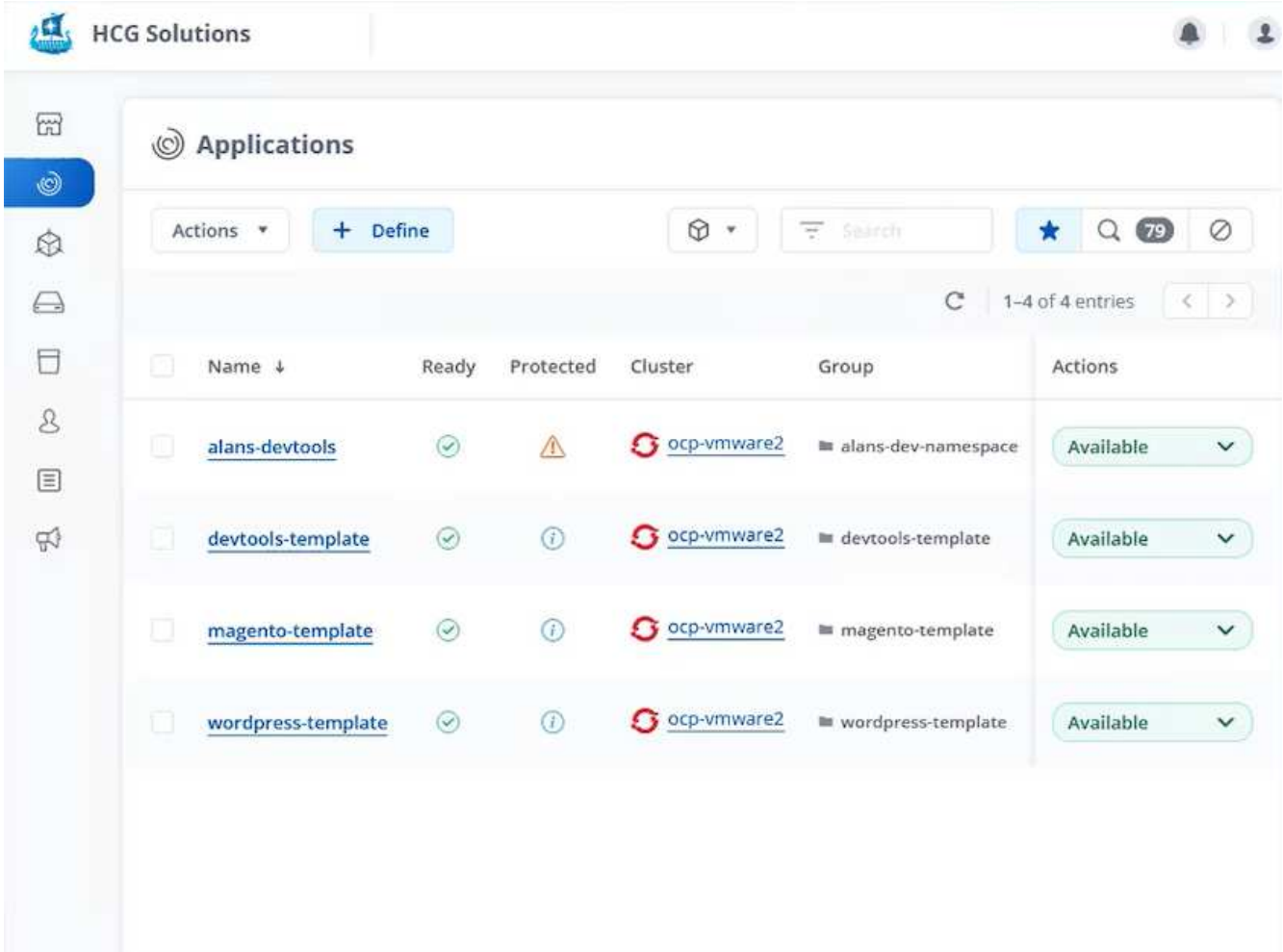
> Per sfruttare la tecnologia FlexClone di NetApp nel processo di cloning, `src-cluster` e. `dest-cluster` deve essere lo stesso.

4. È ora possibile eseguire il manuale per clonare l'applicazione.

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

> Il playbook così come è stato scritto deve essere eseguito dall'utente root o da un utente che può eseguire l'escalation attraverso il processo sudo passando l'argomento "-K".

5. Quando il playbook completa la sua esecuzione, l'applicazione clonata viene visualizzata come disponibile nella console di Astra Control Center.



6. Un utente può quindi accedere all'ambiente Kubernetes in cui è stata implementata l'applicazione, verificare che l'applicazione sia esposta con un nuovo indirizzo IP e iniziare il lavoro di sviluppo.

Per una dimostrazione di questo caso di utilizzo e un esempio di aggiornamento di un'applicazione, vedere "qui".