Panoramica delle integrazioni di storage NetApp

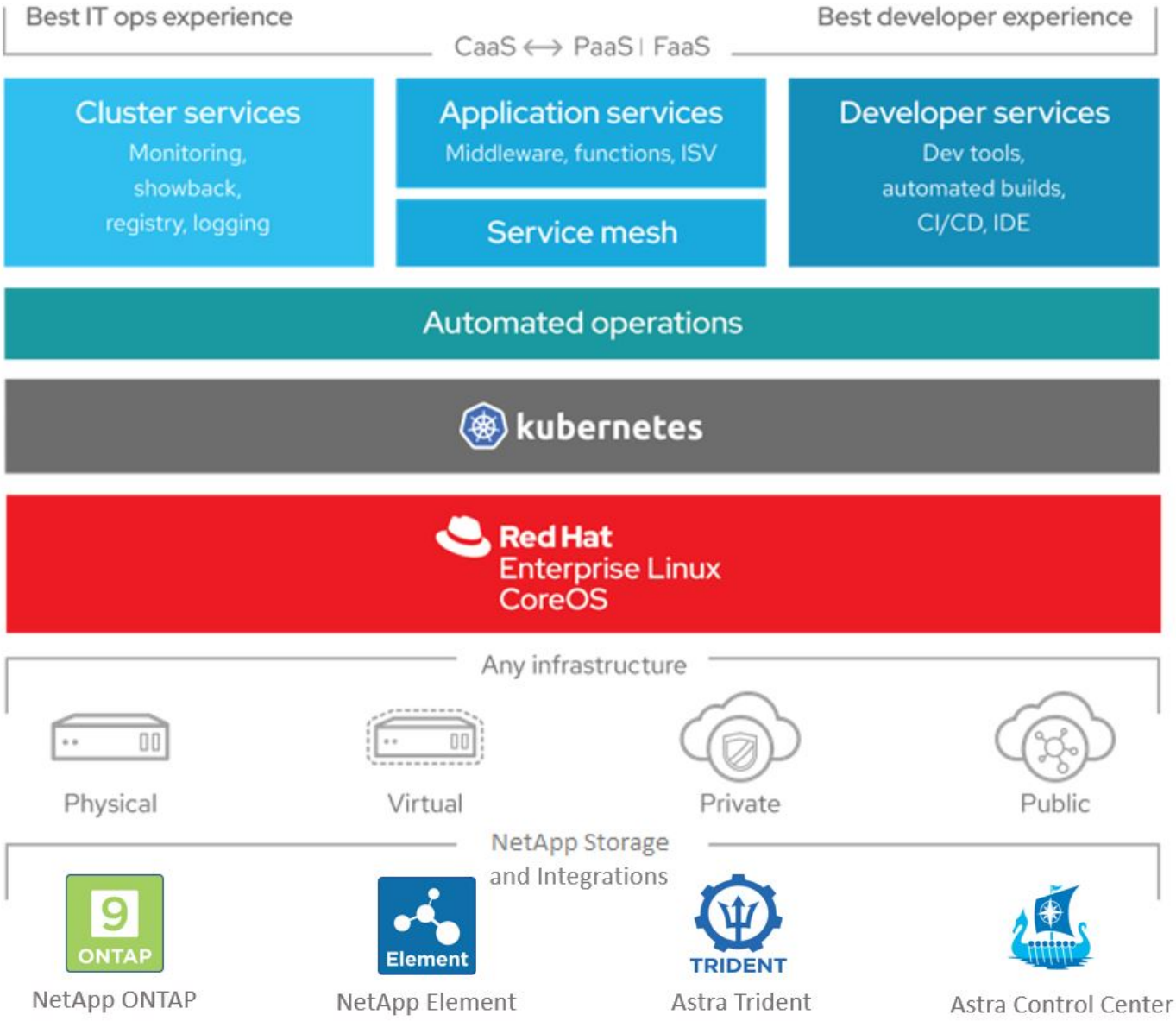
NetApp Solutions

NetApp
April 26, 2024

Sommario

Panoramica sull’integrazione dello storage NetApp

NetApp offre una serie di prodotti per aiutarvi nell’orchestrazione e nella gestione dei dati persistenti in ambienti basati su container, come Red Hat OpenShift.



NetApp Astra Control offre un set completo di servizi di gestione dei dati application-aware e storage per carichi di lavoro Kubernetes stateful, basati sulla tecnologia di protezione dei dati di NetApp. Astra Control Service è disponibile per supportare carichi di lavoro stateful nelle implementazioni Kubernetes native nel cloud. Astra Control Center è disponibile per supportare carichi di lavoro stateful in implementazioni on-premise, come Red Hat OpenShift. Per ulteriori informazioni, visita il sito Web di NetApp Astra Control "qui".

NetApp Astra Trident è un orchestrator di storage open-source e completamente supportato per container e distribuzioni Kubernetes, tra cui Red Hat OpenShift. Per ulteriori informazioni, visita il sito web di Astra Trident "qui".

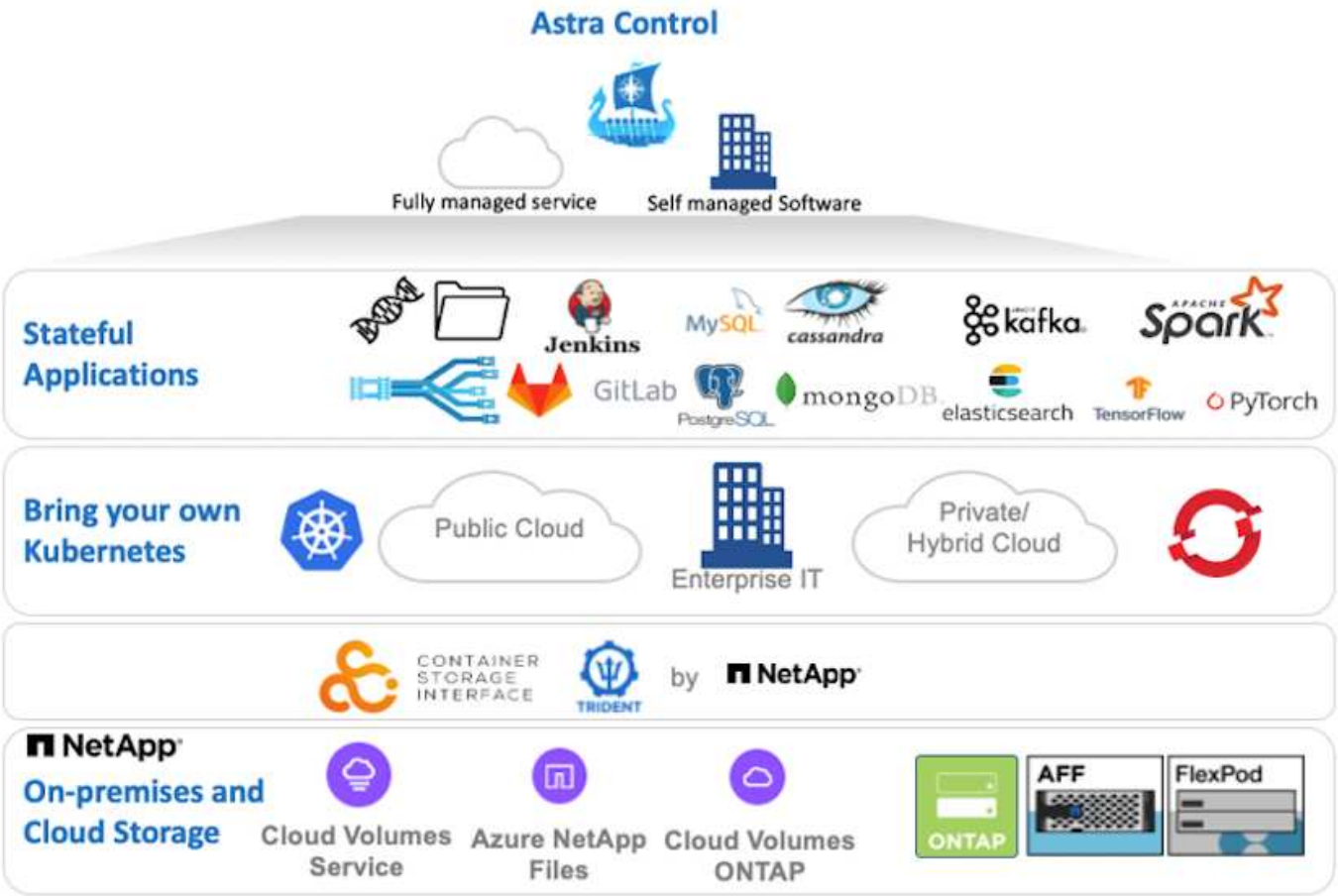
Le pagine seguenti contengono informazioni aggiuntive sui prodotti NetApp validati per la gestione delle

applicazioni e dello storage persistente nella soluzione Red Hat OpenShift con NetApp:

- "NetApp Astra Control Center"
- "NetApp Astra Trident"

Panoramica di NetApp Astra Control Center

NetApp Astra Control Center offre un'ampia gamma di servizi di gestione dei dati basati su applicazioni e storage per carichi di lavoro Kubernetes stateful implementati in un ambiente on-premise e basati sulla tecnologia di protezione dei dati di NetApp.



È possibile installare NetApp Astra Control Center su un cluster Red Hat OpenShift che dispone di Astra Trident Storage orchestrator implementato e configurato con classi di storage e backend di storage per i sistemi storage NetApp ONTAP.

Per l'installazione e la configurazione di Astra Trident per il supporto di Astra Control Center, vedere "questo documento qui".

In un ambiente connesso al cloud, il centro di controllo Astra utilizza Cloud Insights per fornire monitoraggio avanzato e telemetria. In assenza di una connessione Cloud Insights, sono disponibili funzioni limitate di monitoraggio e telemetria (7 giorni di metriche) ed esportate negli strumenti di monitoraggio nativi di Kubernetes (Prometheus e Grafana) attraverso endpoint di metriche aperte.

Il centro di controllo Astra è completamente integrato nell'ecosistema NetApp AutoSupport e Active IQ per fornire supporto agli utenti, fornire assistenza per la risoluzione dei problemi e visualizzare le statistiche di utilizzo.

Oltre alla versione a pagamento di Astra Control Center, è disponibile una licenza di valutazione di 90 giorni. La versione di valutazione è supportata tramite e-mail e community (canale slack). I clienti hanno accesso a questi e ad altri articoli della Knowledge base e alla documentazione disponibile nella dashboard di supporto dei prodotti.

Per iniziare a utilizzare NetApp Astra Control Center, visita il "Sito web Astra".

Prerequisiti per l'installazione di Astra Control Center

1. Uno o più cluster Red Hat OpenShift. Le versioni 4.6 EUS e 4.7 sono attualmente supportate.
2. Astra Trident deve essere già installato e configurato su ogni cluster Red Hat OpenShift.
3. Uno o più sistemi storage NetApp ONTAP con ONTAP 9.5 o superiore.

   > Per ogni installazione di OpenShift in un sito è consigliabile disporre di una SVM dedicata per lo storage persistente. Le implementazioni multi-sito richiedono sistemi storage aggiuntivi.

4. È necessario configurare un backend di storage Trident su ciascun cluster OpenShift con una SVM supportata da un cluster ONTAP.
5. StorageClass predefinita configurata su ciascun cluster OpenShift con Astra Trident come storage provisioning.
6. È necessario installare e configurare un bilanciamento del carico su ciascun cluster OpenShift per il bilanciamento del carico e l'esposizione dei servizi OpenShift.

   > Vedere il link "qui" per informazioni sui bilanciatori di carico validati per questo scopo.

7. È necessario configurare un registro di immagini privato per ospitare le immagini di NetApp Astra Control Center.

   > Vedere il link "qui" Per installare e configurare un registro privato OpenShift a tale scopo.

8. È necessario disporre dell'accesso Cluster Admin al cluster Red Hat OpenShift.
9. È necessario disporre dell'accesso come amministratore ai cluster NetApp ONTAP.
10. Una workstation di amministrazione con i tool docker o podman, tridentctl e oc o kubectl installati e aggiunti al percorso dei dollari.

    > Le installazioni di Docker devono avere una versione di Docker superiore alla 20.10 e le installazioni di Podman devono avere una versione di podman superiore alla 3.0.

Installare Astra Control Center

Utilizzo di OperatorHub

1. Accedere al NetApp Support Site e scaricare l'ultima versione di NetApp Astra Control Center. Per farlo, è necessaria una licenza allegata al tuo account NetApp. Dopo aver scaricato il tarball, trasferirlo sulla workstation di amministrazione.

   > Per iniziare a utilizzare una licenza di prova per Astra Control, visitare il sito "Sito di registrazione Astra".

2. Disimballare il tar ball e modificare la directory di lavoro nella cartella risultante.

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

3. Prima di iniziare l'installazione, trasferire le immagini di Astra Control Center in un registro di immagini. Puoi scegliere di farlo con Docker o Podman; in questo passaggio vengono fornite le istruzioni per entrambi.

Podman

a. Esportare ‘reFQDN del Registro di sistema con il nome dell’organizzazione/namespace/progetto come variabile di ambiente 'gistry'.

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

b. Accedere al Registro di sistema.

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

> Se si utilizza `kubeadmin` utente per accedere al registro privato, quindi utilizzare il token invece della password - `podman login -u ocp-user -p token --tls-verify=false astra-registry.apps.ocp-vmw.cie.netapp.com`.

> In alternativa, è possibile creare un account di servizio, assegnare un ruolo di editor del Registro di sistema e/o di visualizzatore del Registro di sistema (a seconda che si richieda l’accesso push/pull) e accedere al Registro di sistema utilizzando il token dell’account di servizio.

c. Creare un file script della shell e incollarne il contenuto seguente.

```
[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
```

> Se si utilizzano certificati non attendibili per il Registro di sistema, modificare lo script della shell e utilizzare `--tls-verify=false` per il comando podman push `podman push $REGISTRY/$(echo $astraImage | sed 's/[\/]\+\///') --tls-verify=false`.

d. Rendere il file eseguibile.

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

e. Eseguire lo script della shell.

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

Docker

a. Esportare ‘reFQDN del Registro di sistema con il nome dell’organizzazione/namespace/progetto come variabile di ambiente 'gistry'.

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

b. Accedere al Registro di sistema.

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

> Se si utilizza `kubeadmin` utente per accedere al registro privato, quindi utilizzare il token invece della password - `docker login -u ocp-user -p token astra-registry.apps.ocp-vmw.cie.netapp.com`.

> In alternativa, è possibile creare un account di servizio, assegnare un ruolo di editor del Registro di sistema e/o di visualizzatore del Registro di sistema (a seconda che si richieda l’accesso push/pull) e accedere al Registro di sistema utilizzando il token dell’account di servizio.

c. Creare un file script della shell e incollarne il contenuto seguente.

```
[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. Rendere il file eseguibile.

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

e. Eseguire lo script della shell.

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

4. Quando si utilizzano registri di immagini private non pubblicamente attendibili, caricare i certificati TLS del registro di immagini nei nodi OpenShift. A tale scopo, creare una configurazione nello spazio dei nomi openshift-config utilizzando i certificati TLS e applicarla alla configurazione dell'immagine del cluster per rendere attendibile il certificato.

```
[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
```

> Se si utilizza un registro interno di OpenShift con certificati TLS predefiniti dall'operatore di ingresso con un percorso, è comunque necessario seguire la procedura precedente per applicare la patch ai certificati con il nome host del percorso. Per estrarre i certificati dall'operatore di ingresso, è possibile utilizzare il comando `oc extract secret/router-ca --keys=tls.crt -n openshift-ingress-operator`.

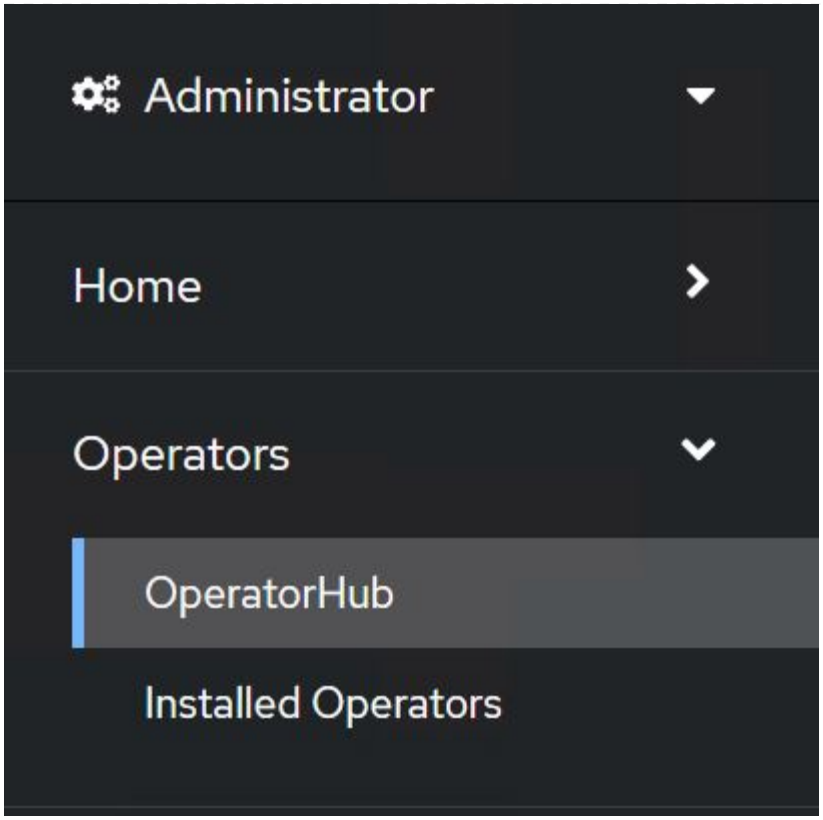
5. Creare uno spazio dei nomi `netapp-acc-operator` Per Astra Control Center.

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

namespace/netapp-acc-operator created
```

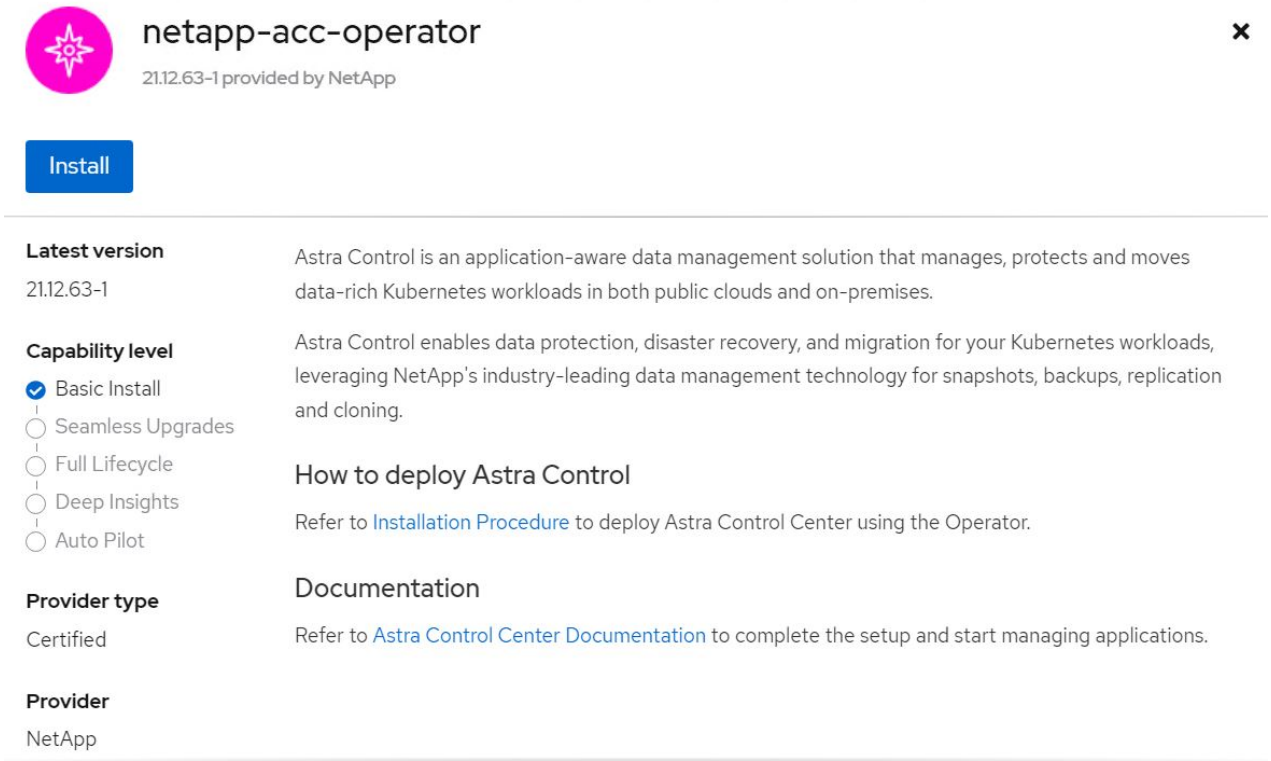
6. Creare un segreto con le credenziali per accedere al registro delle immagini in `netapp-acc-operator` namespace.

```
[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
```

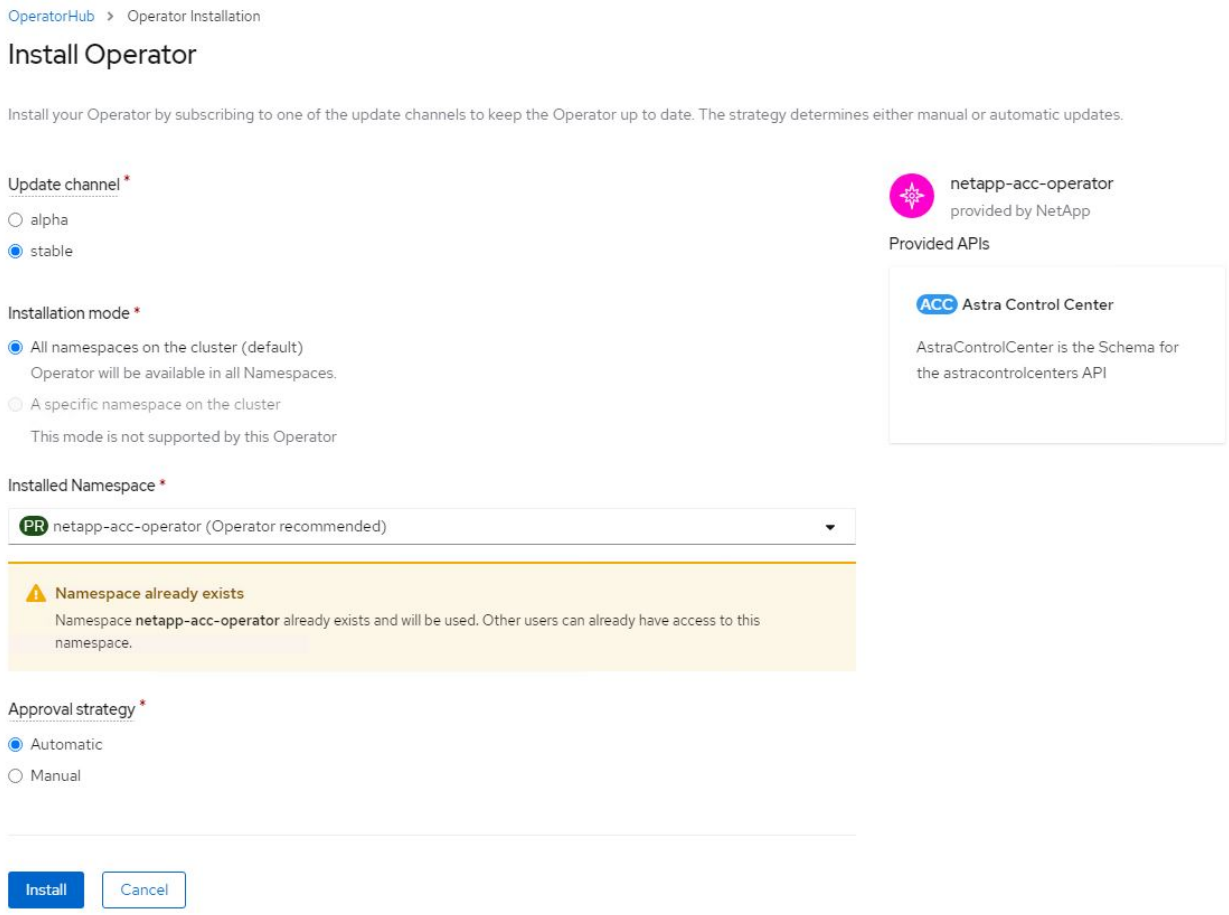
7. Accedi alla console GUI di Red Hat OpenShift con accesso cluster-admin.
8. Selezionare Administrator (Amministratore) dal menu a discesa Perspective (prospettiva).
9. Accedere a Operator > OperatorHub e cercare Astra.



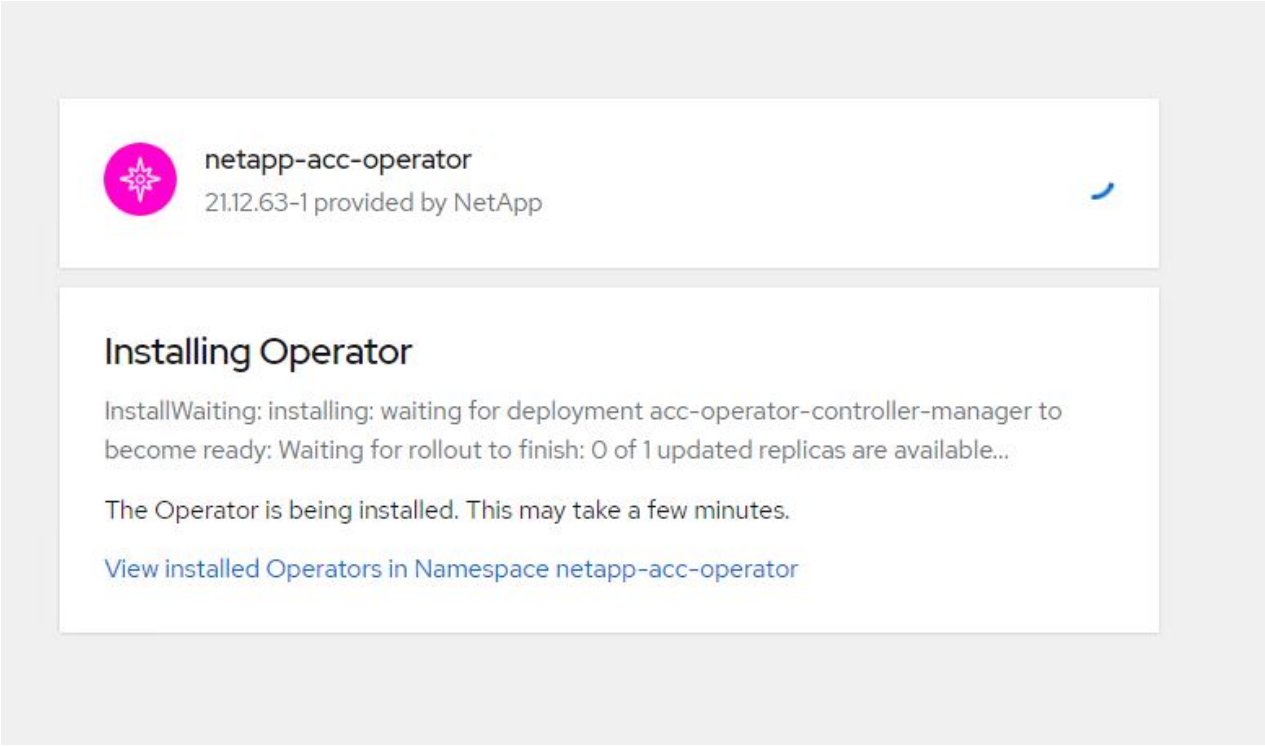
10. Selezionare `netapp-acc-operator` affiancare e fare clic su `Install`.



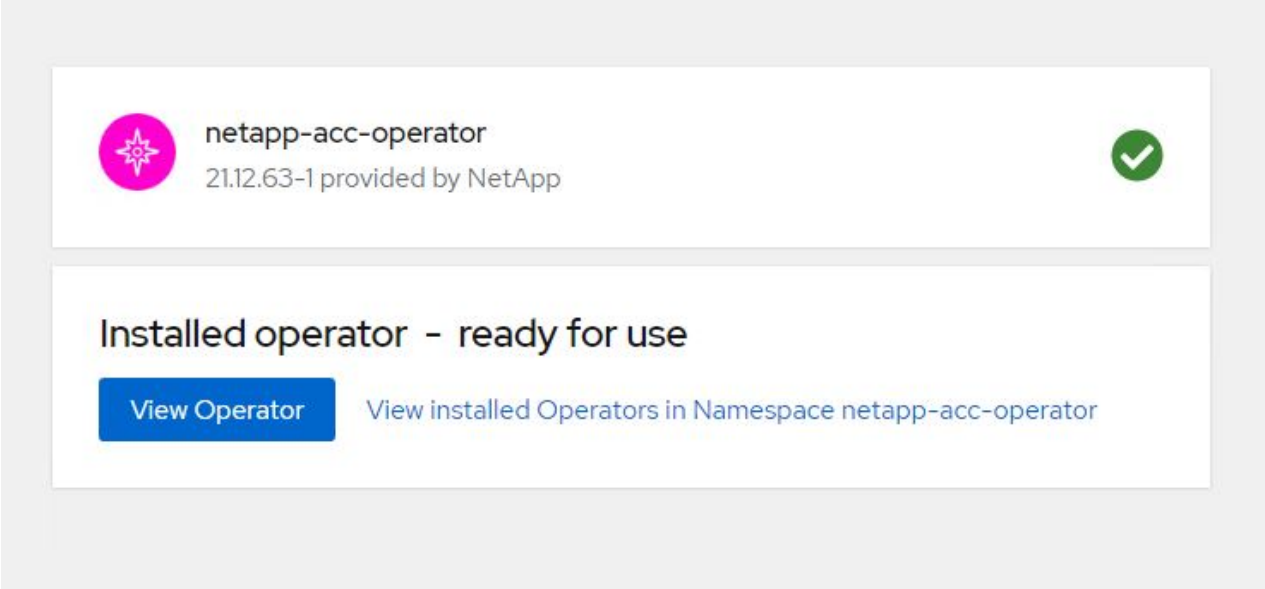
11. Nella schermata Install Operator (Installa operatore), accettare tutti i parametri predefiniti e fare clic su `Install`.



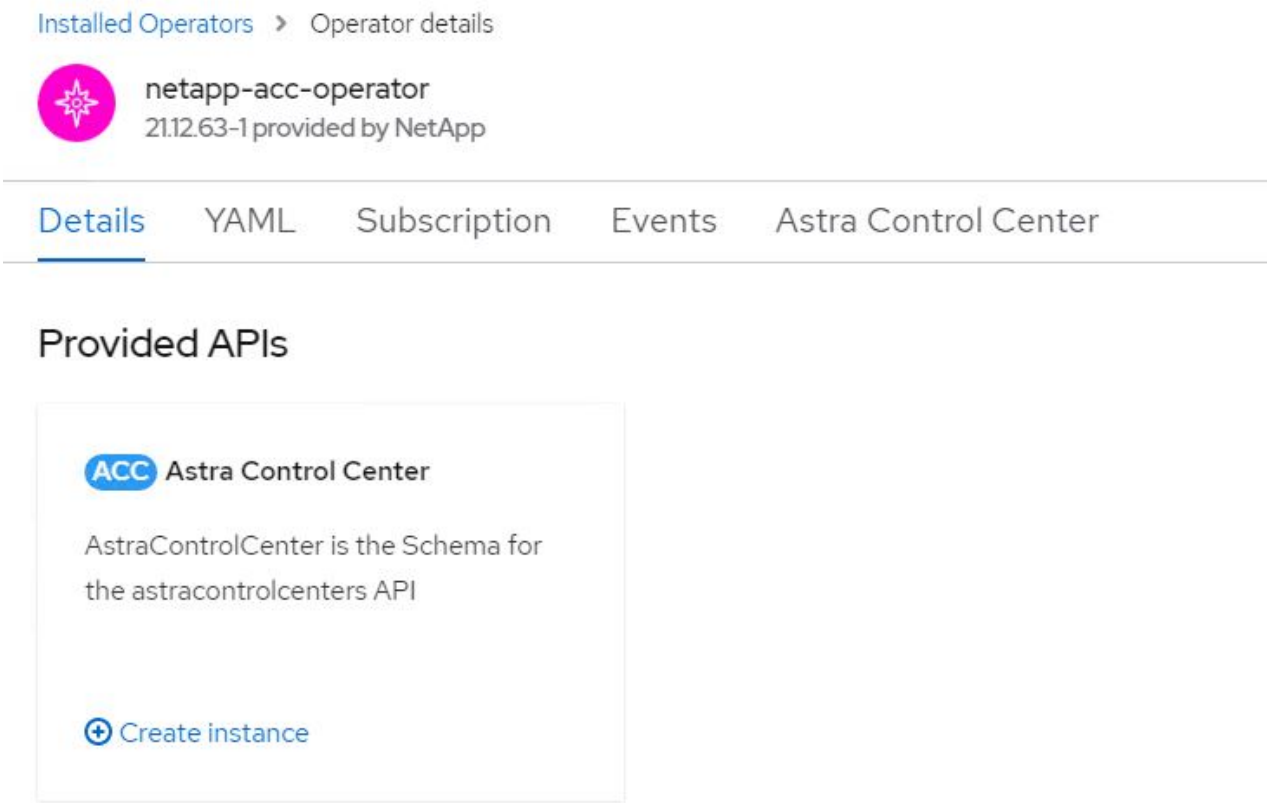
12. Attendere il completamento dell'installazione da parte dell'operatore.



13. Una volta completata l'installazione dell'operatore, selezionare per fare clic su `View Operator`.



14. Quindi fare clic su `Create Instance` Nel riquadro Astra Control Center dell'operatore.



15. Riempire `Create AstraControlCenter` campi del modulo e fare clic su `Create`.

    a. Se si desidera, modificare il nome dell'istanza di Astra Control Center.
    b. Se si desidera, attivare o disattivare il supporto automatico. Si consiglia di mantenere la funzionalità di supporto automatico.
    c. Inserire il nome FQDN per Astra Control Center.
    d. Inserire la versione di Astra Control Center; per impostazione predefinita viene visualizzata la

versione più recente.

e. Inserisci un nome account per Astra Control Center e i dettagli dell'amministratore come nome, cognome e indirizzo e-mail.

f. Inserire il criterio di recupero del volume, l'impostazione predefinita è Mantieni.

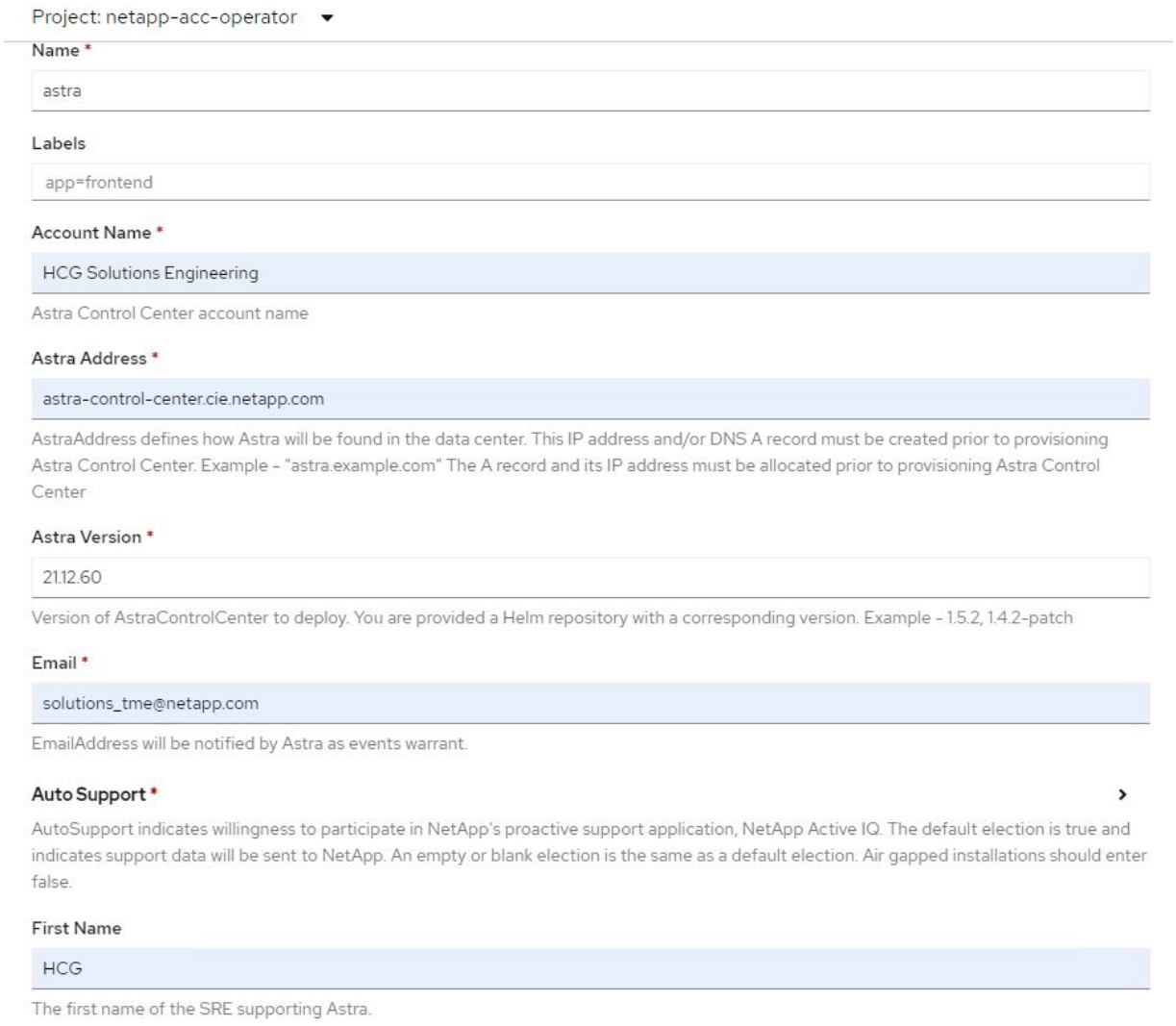
g. In Image Registry (Registro immagini), immettere l'FQDN del registro insieme al nome dell'organizzazione assegnato durante l'invio delle immagini al registro (in questo esempio, `astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra`)

h. Se si utilizza un registro che richiede l'autenticazione, inserire il nome segreto nella sezione Registro immagini.

i. Configurare le opzioni di scalabilità per i limiti delle risorse di Astra Control Center.

j. Inserire il nome della classe di storage se si desidera inserire PVC in una classe di storage non predefinita.

k. Definire le preferenze di gestione CRD.



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

Automatizzato [Ansible]

1. Per utilizzare i playbook Ansible per implementare Astra Control Center, è necessaria una macchina Ubuntu/RHEL con Ansible installato. Seguire le procedure "qui" Per Ubuntu e RHEL.
2. Clonare il repository GitHub che ospita il contenuto Ansible.

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

3. Accedi al sito NetApp Support e scarica l'ultima versione di NetApp Astra Control Center. Per farlo, è necessaria una licenza allegata al tuo account NetApp. Dopo aver scaricato il tarball, trasferirlo sulla workstation.

> Per iniziare a utilizzare una licenza di prova per Astra Control, visitare il sito "Sito di registrazione Astra".

4. Creare o ottenere il file kubeconfig con accesso amministratore al cluster OpenShift su cui deve essere installato Astra Control Center.

5. Modificare la directory in na_astra_control_suite.

```
cd na_astra_control_suite
```

6. Modificare il `vars/vars.yml` e inserire le variabili con le informazioni richieste.

```
#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. Esegui il manuale per implementare Astra Control Center. Il playbook richiede privilegi root per alcune configurazioni.

   Se l'utente che esegue il playbook è root o ha configurato sudo senza password, eseguire il seguente comando per eseguire il playbook.

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

   Se l'utente ha configurato l'accesso sudo basato su password, eseguire il seguente comando per eseguire il manuale, quindi inserire la password sudo.

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

Fasi successive all'installazione

1. Il completamento dell'installazione potrebbe richiedere alcuni minuti. Verificare che tutti i pod e i servizi in `netapp-astra-cc` namespace in esecuzione.

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

2. Controllare `acc-operator-controller-manager` registri per garantire che l'installazione sia completata.

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

Il seguente messaggio indica la corretta installazione di Astra Control Center.

```
{"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. Il nome utente per l'accesso ad Astra Control Center è l'indirizzo e-mail dell'amministratore fornito nel file CRD e la password è una stringa `ACC-` Aggiunto all'UUID di Astra Control Center. Eseguire il seguente comando:

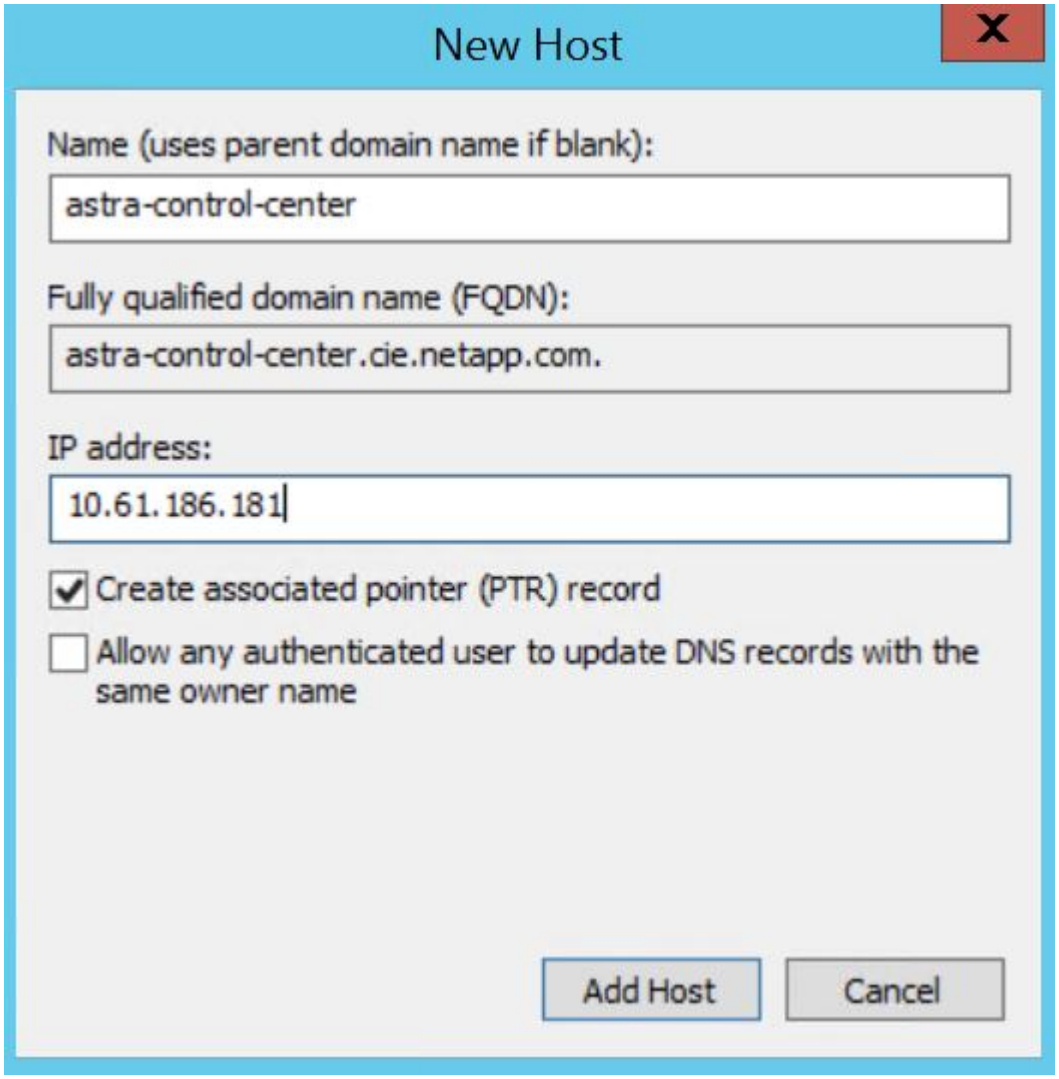
```
[netapp-user@rhel7 ~]$ oc get astracontrolcenters -n netapp-astra-cc
NAME    UUID
astra   345c55a5-bf2e-21f0-84b8-b6f2bce5e95f
```

In questo esempio, la password è `ACC-345c55a5-bf2e-21f0-84b8-b6f2bce5e95f`.

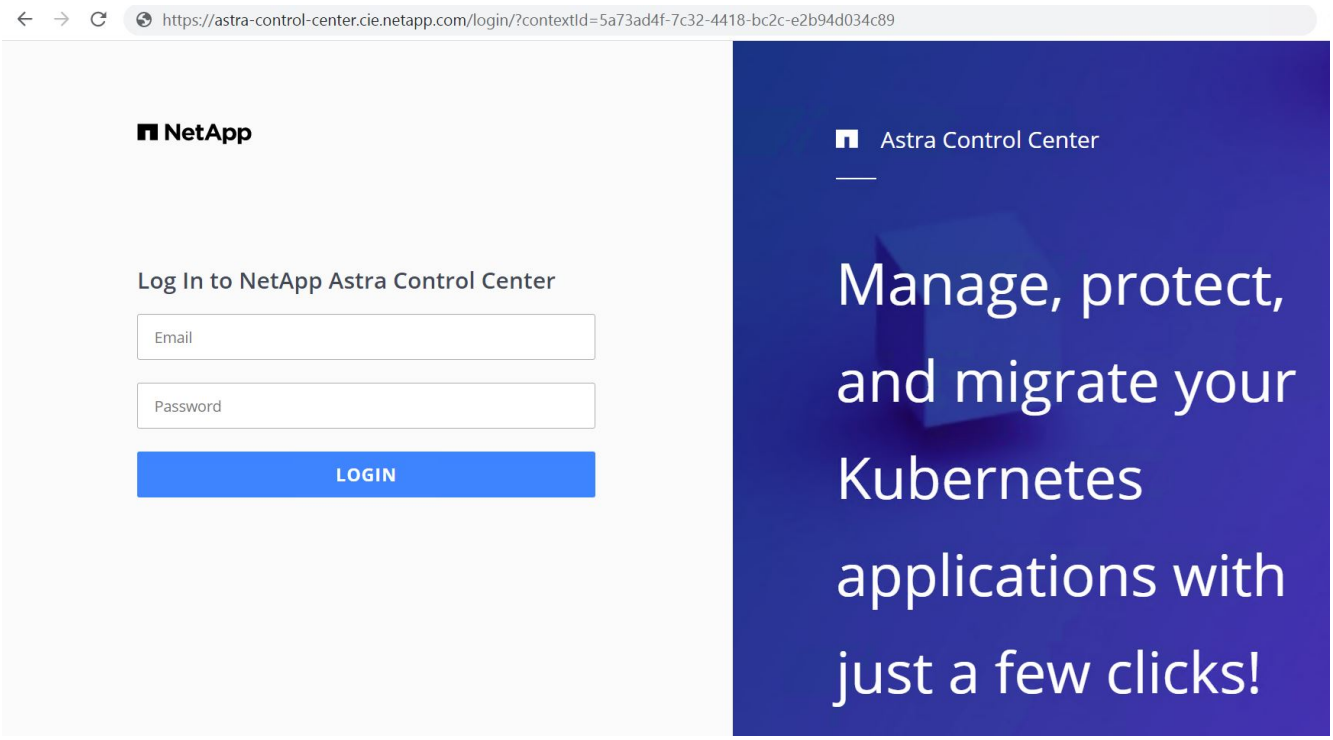
4. Ottieni l'IP del bilanciamento del carico del servizio traefik.

```
[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
```

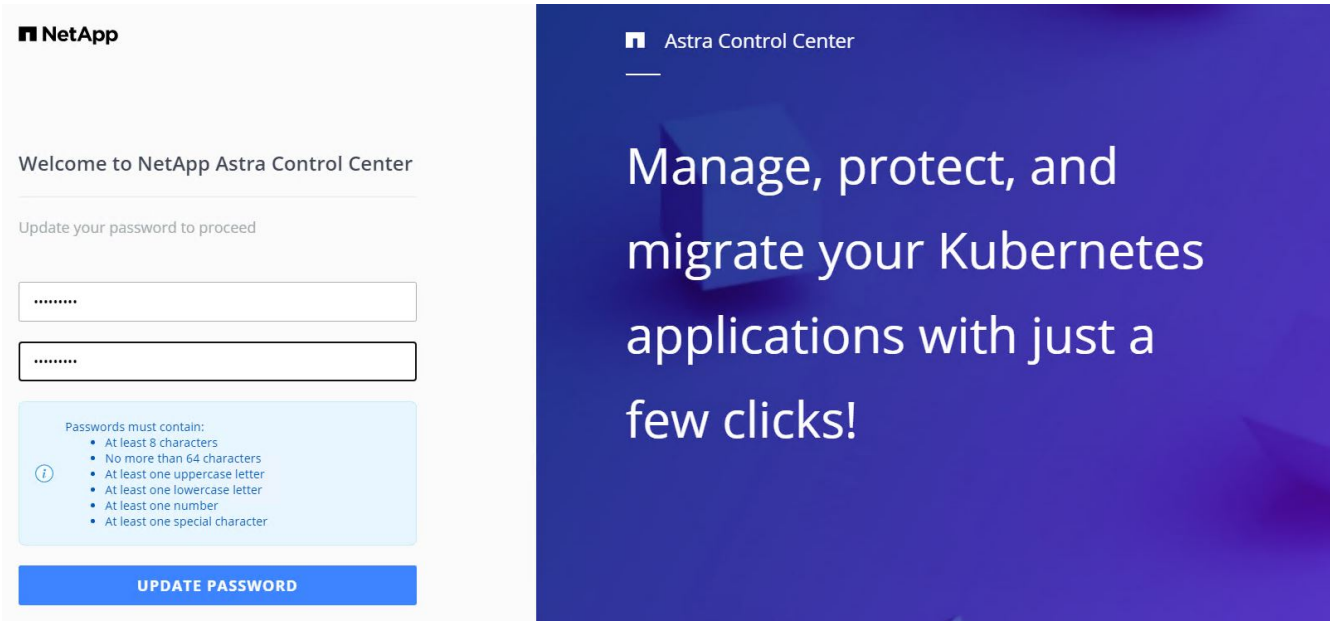
5. Aggiungere una voce nel server DNS che punta all'FQDN fornito nel file CRD di Astra Control Center `EXTERNAL-IP` del servizio traefik.



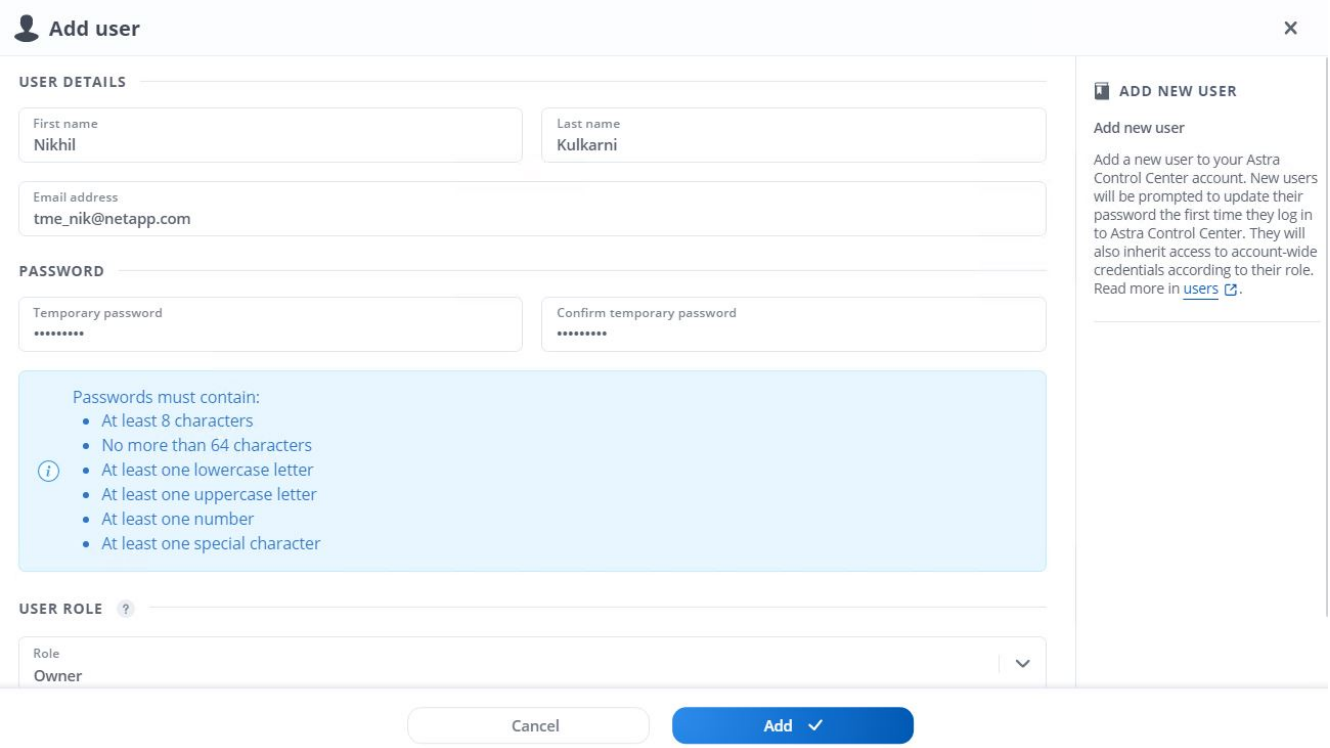
6. Accedere alla GUI di Astra Control Center esplorando il relativo FQDN.



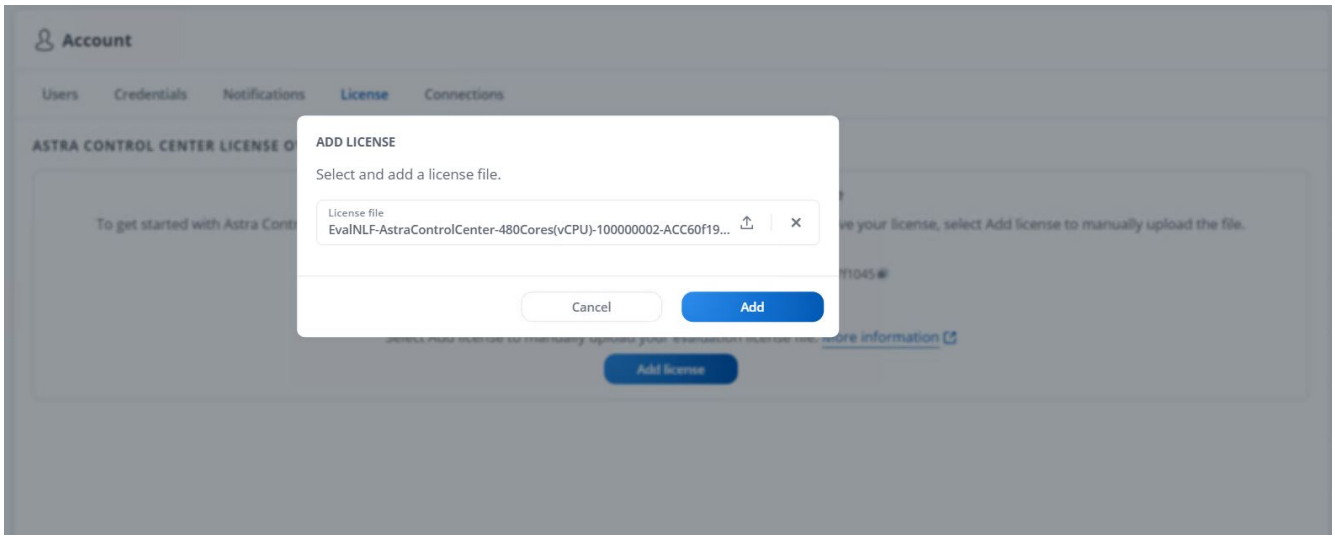
7. Quando si accede all'interfaccia grafica di Astra Control Center per la prima volta utilizzando l'indirizzo email admin fornito in CRD, è necessario modificare la password.



8. Se si desidera aggiungere un utente ad Astra Control Center, accedere a account > Users (account > utenti), fare clic su Add (Aggiungi), inserire i dettagli dell'utente e fare clic su Add (Aggiungi).



9. Astra Control Center richiede una licenza per il funzionamento di tutte le funzionalità IT. Per aggiungere una licenza, accedere a account > License (account > licenza), fare clic su Add License (Aggiungi licenza) e caricare il file di licenza.



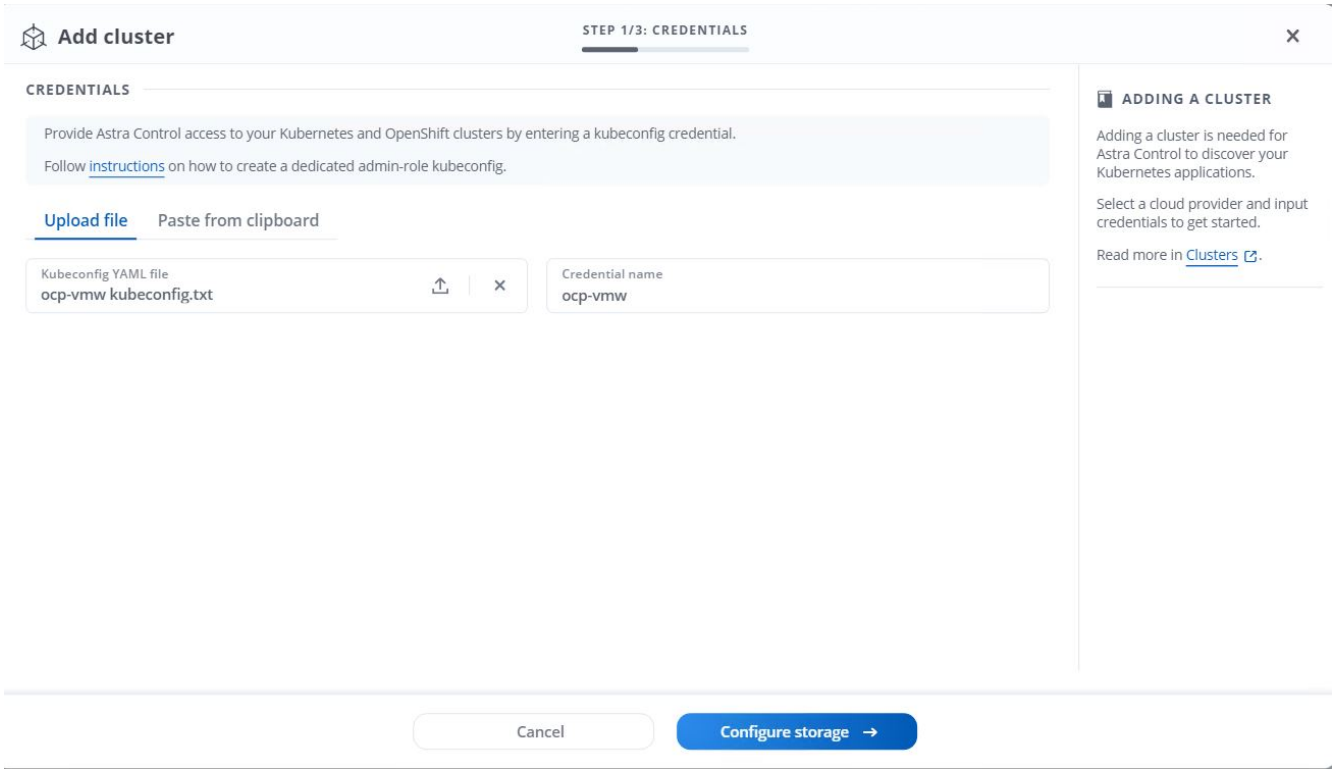
In caso di problemi con l'installazione o la configurazione di NetApp Astra Control Center, è disponibile la knowledge base dei problemi noti "qui".

Registra i tuoi Red Hat OpenShift Clusters con Astra Control Center

Per consentire ad Astra Control Center di gestire i carichi di lavoro, devi prima registrare il cluster Red Hat OpenShift.

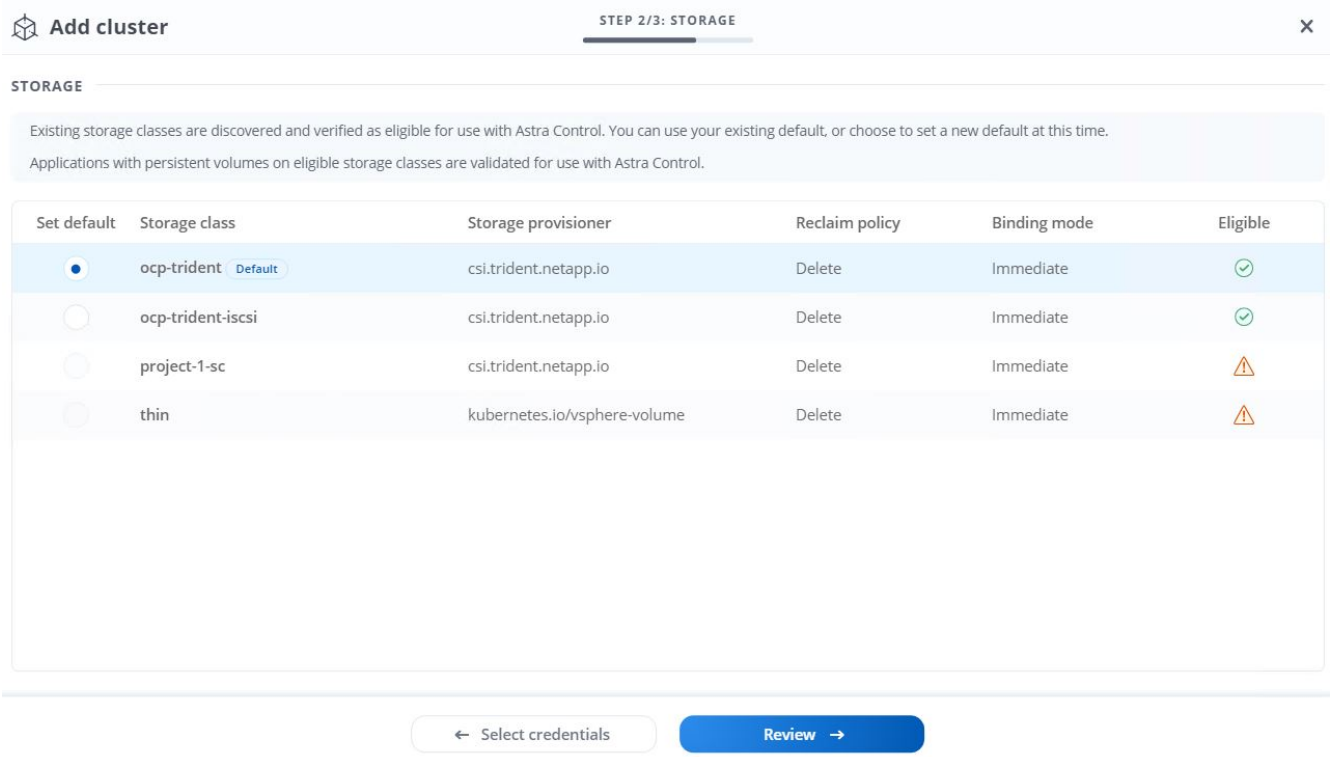
Registra i cluster Red Hat OpenShift

1. Il primo passo consiste nell'aggiungere i cluster OpenShift all'Astra Control Center e gestirli. Accedere a Clusters e fare clic su Add a Cluster (Aggiungi cluster), caricare il file kubeconfig per il cluster OpenShift e fare clic su Select Storage (Seleziona storage).

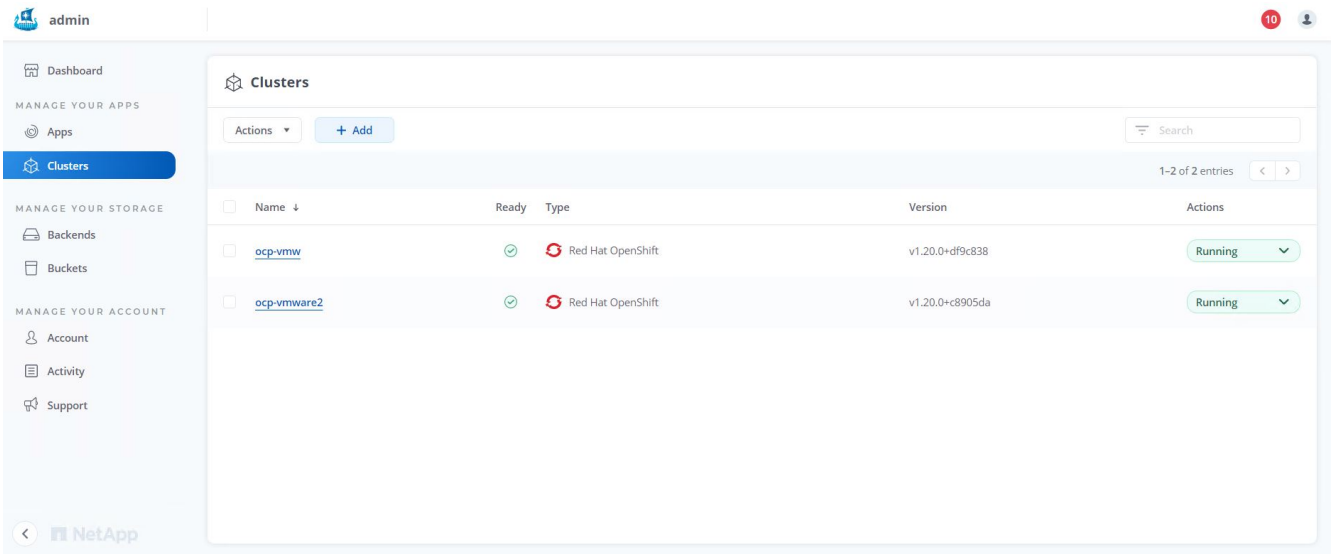


> Il file kubeconfig può essere generato per l'autenticazione con un nome utente e una password o un token. I token scadono dopo un periodo di tempo limitato e potrebbero non essere raggiungibili dal cluster registrato. NetApp consiglia di utilizzare un file kubeconfig con nome utente e password per registrare i cluster OpenShift su Astra Control Center.

2. Astra Control Center rileva le classi di storage idonee. Selezionare ora il modo in cui lo storageclass effettua il provisioning dei volumi utilizzando Trident supportato da una SVM su NetApp ONTAP e fare clic su Review (esamina). Nel riquadro successivo, verificare i dettagli e fare clic su Add Cluster (Aggiungi cluster).

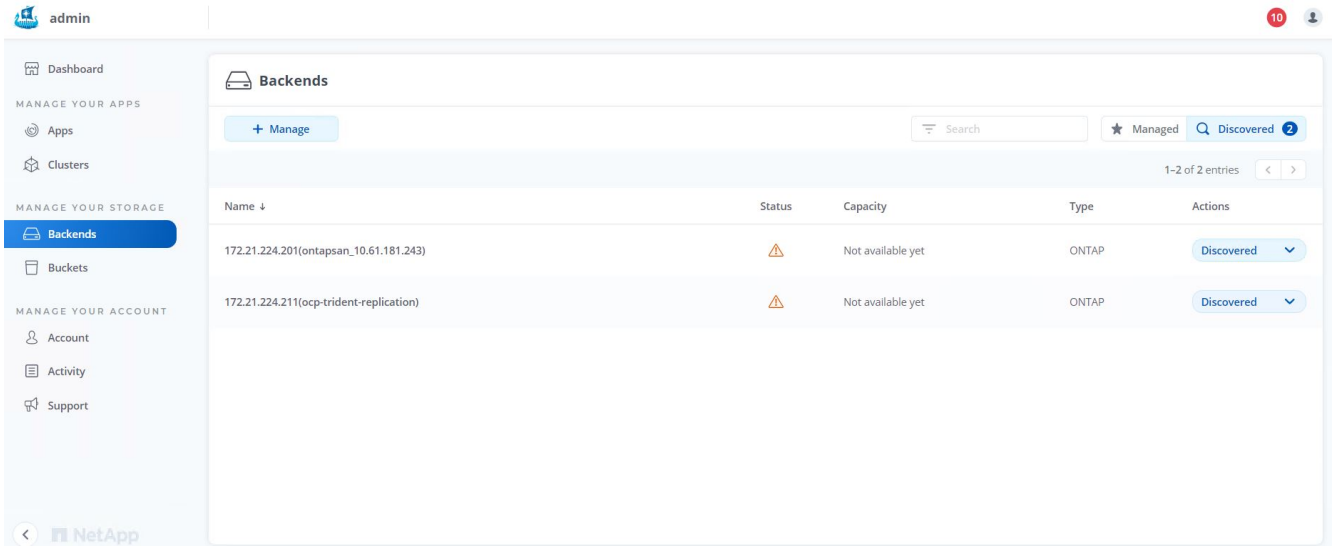


3. Registrare entrambi i cluster OpenShift come descritto al punto 1. Una volta aggiunti, i cluster passano allo stato di rilevamento mentre Astra Control Center li ispeziona e installa gli agenti necessari. Lo stato del cluster diventa in esecuzione dopo che sono stati registrati correttamente.

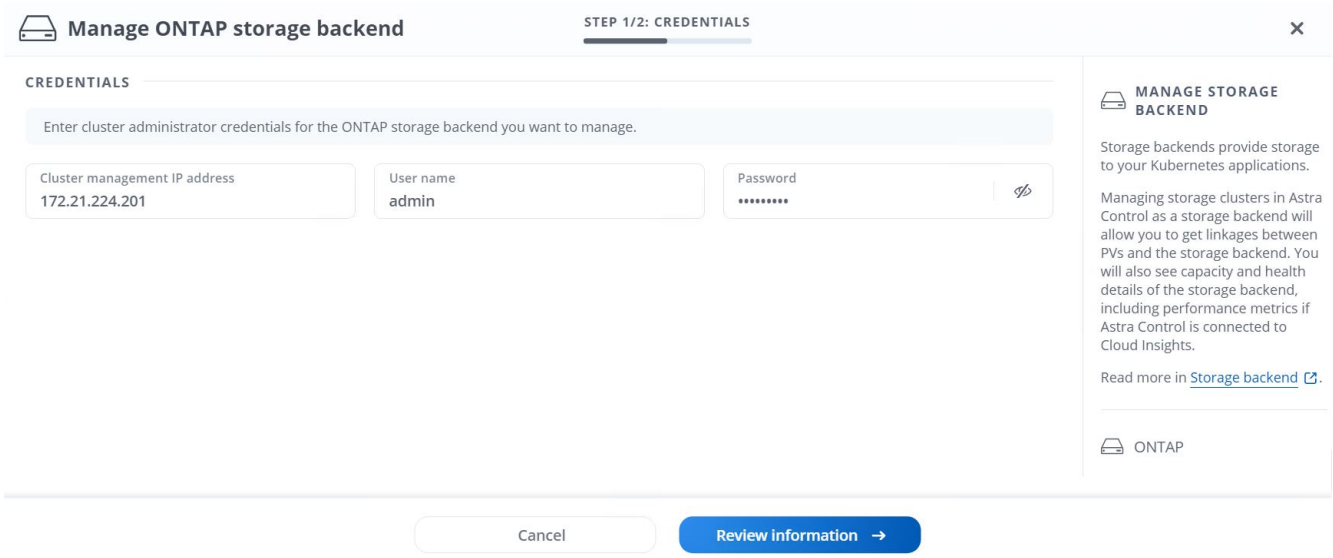


> Tutti i cluster Red Hat OpenShift che devono essere gestiti da Astra Control Center devono avere accesso al registro delle immagini utilizzato per l'installazione, poiché gli agenti installati sui cluster gestiti estraggono le immagini da tale registro.

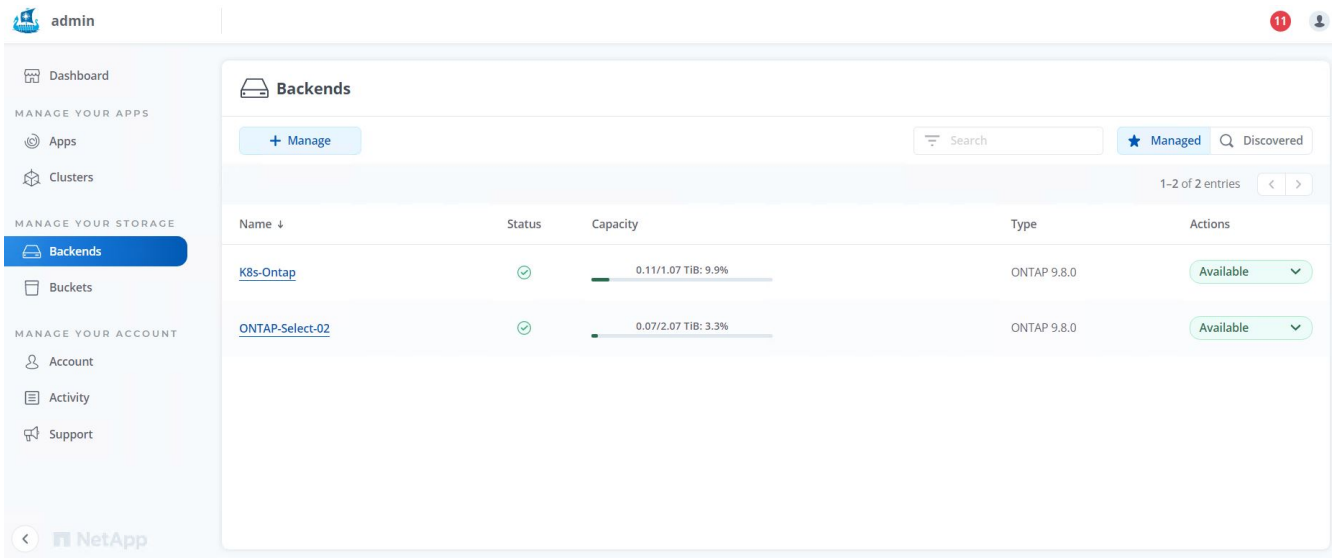
4. Importa i cluster ONTAP come risorse storage da gestire come back-end dal centro di controllo Astra. Quando i cluster OpenShift vengono aggiunti ad Astra e viene configurato uno storageclass, il cluster ONTAP viene automaticamente ispezionato e ispezionato per il backup dello storageclass, ma non viene importato nel centro di controllo Astra da gestire.



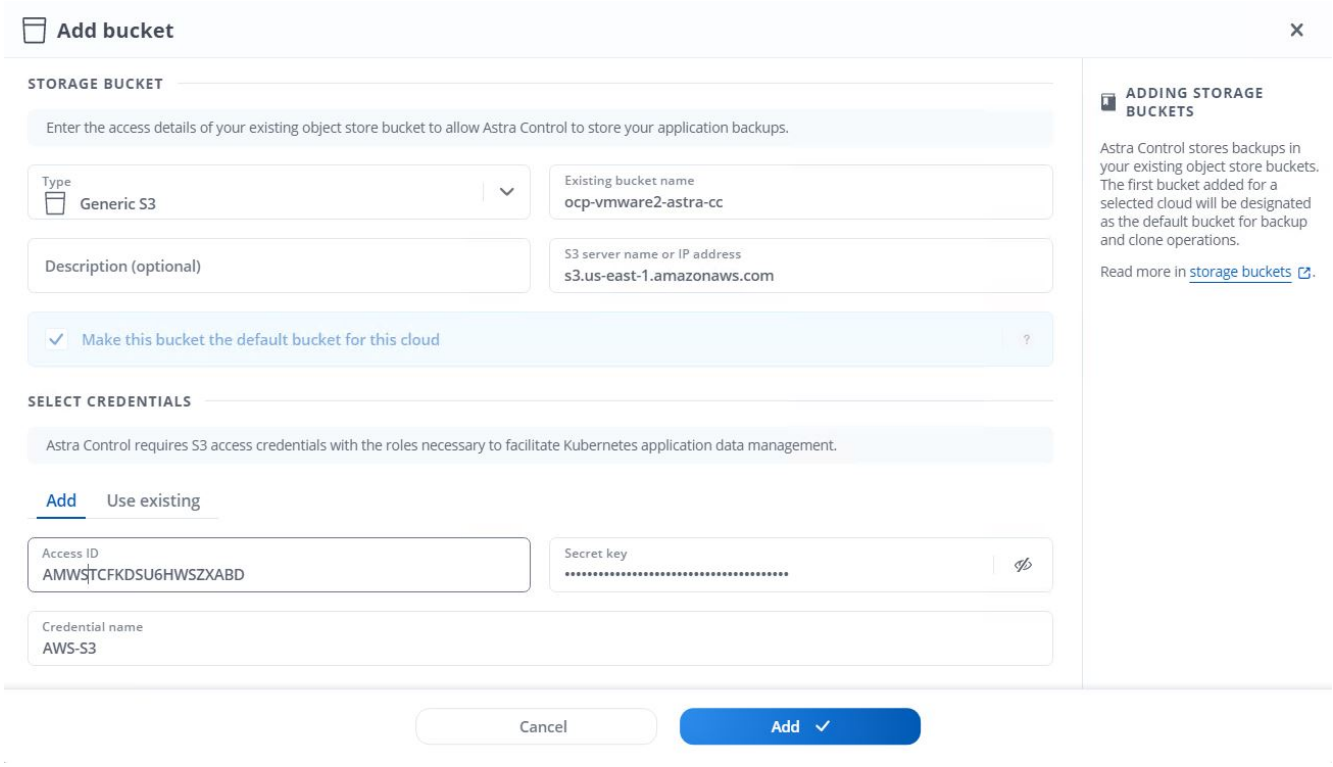
5. Per importare i cluster ONTAP, accedere a Backend, fare clic sul menu a discesa e selezionare Manage (Gestisci) accanto al cluster ONTAP da gestire. Immettere le credenziali del cluster ONTAP, fare clic su informazioni di revisione, quindi fare clic su Importa backend storage.



6. Una volta aggiunti i backend, lo stato diventa disponibile. Questi backend ora dispongono delle informazioni sui volumi persistenti nel cluster OpenShift e sui volumi corrispondenti nel sistema ONTAP.



7. Per il backup e il ripristino tra cluster OpenShift utilizzando Astra Control Center, è necessario eseguire il provisioning di un bucket di storage a oggetti che supporti il protocollo S3. Le opzioni attualmente supportate sono ONTAP S3, StorageGRID e AWS S3. Ai fini di questa installazione, configureremo un bucket AWS S3. Accedere a Bucket, fare clic su Add bucket (Aggiungi bucket) e selezionare Generic S3. Inserisci i dettagli sul bucket S3 e le credenziali per accedervi, fai clic sulla casella di controllo "Rendi questo bucket il bucket predefinito per il cloud", quindi fai clic su Aggiungi.

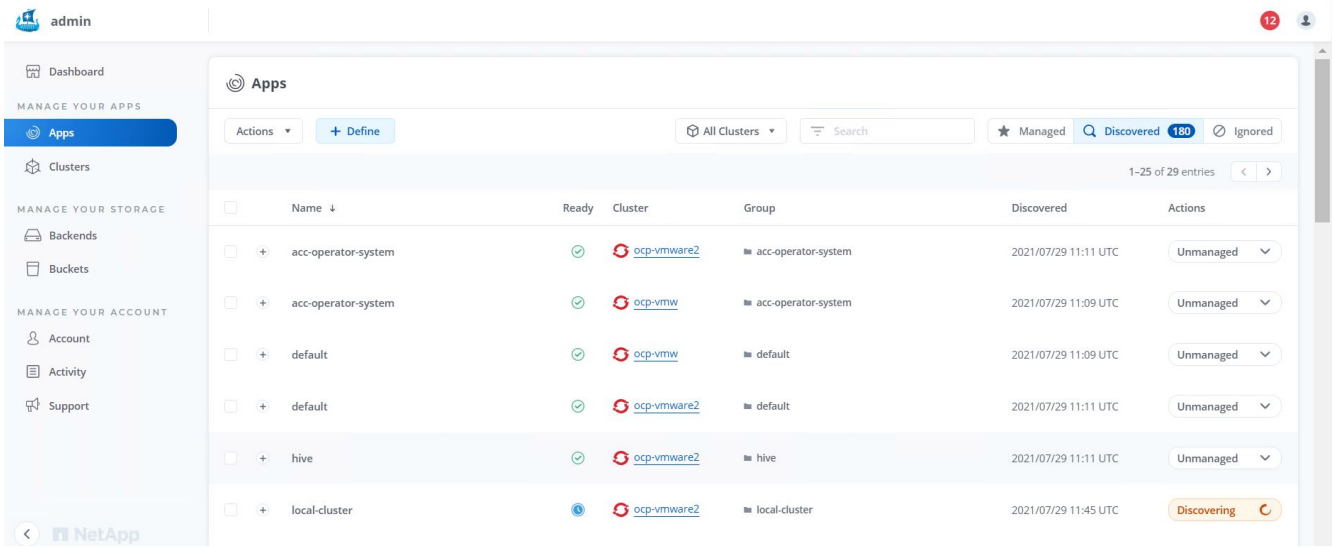


Scegliere le applicazioni da proteggere

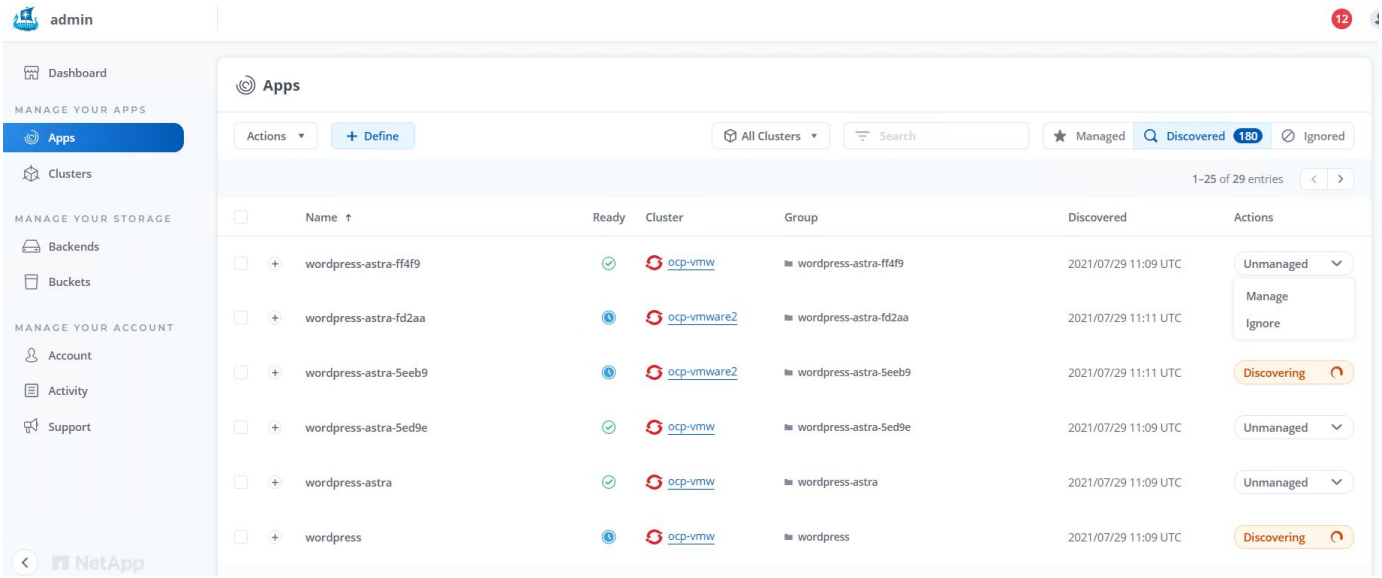
Dopo aver registrato i cluster Red Hat OpenShift, è possibile individuare le applicazioni implementate e gestirle tramite Astra Control Center.

Gestire le applicazioni

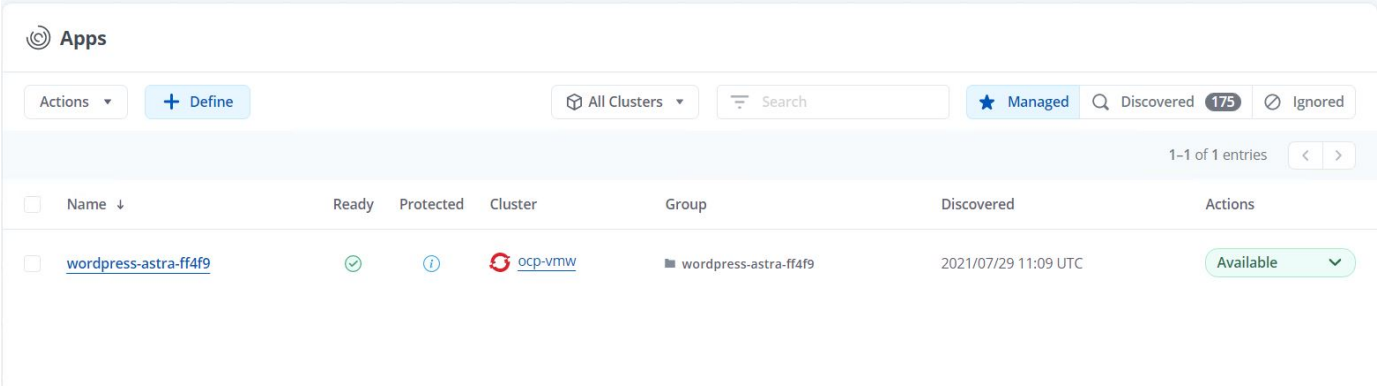
1. Una volta registrati i cluster OpenShift e i backend ONTAP con il centro di controllo Astra, il centro di controllo inizia automaticamente a rilevare le applicazioni in tutti gli spazi dei nomi che utilizzano lo storageclass configurato con il backend ONTAP specificato.



2. Accedere a Apps > Dovered (applicazioni > rilevate) e fare clic sul menu a discesa accanto all'applicazione che si desidera gestire utilizzando Astra. Quindi fare clic su Manage (Gestisci)



1. L'applicazione entra nello stato Available (disponibile) e può essere visualizzata nella scheda Managed (gestito) nella sezione Apps (applicazioni).



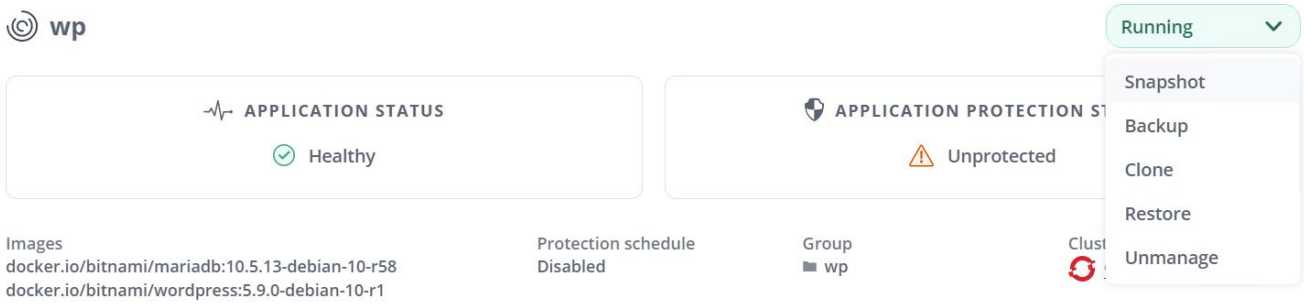
Proteggi le tue applicazioni

Una volta gestiti i carichi di lavoro delle applicazioni da Astra Control Center, è possibile configurare le impostazioni di protezione per tali carichi di lavoro.

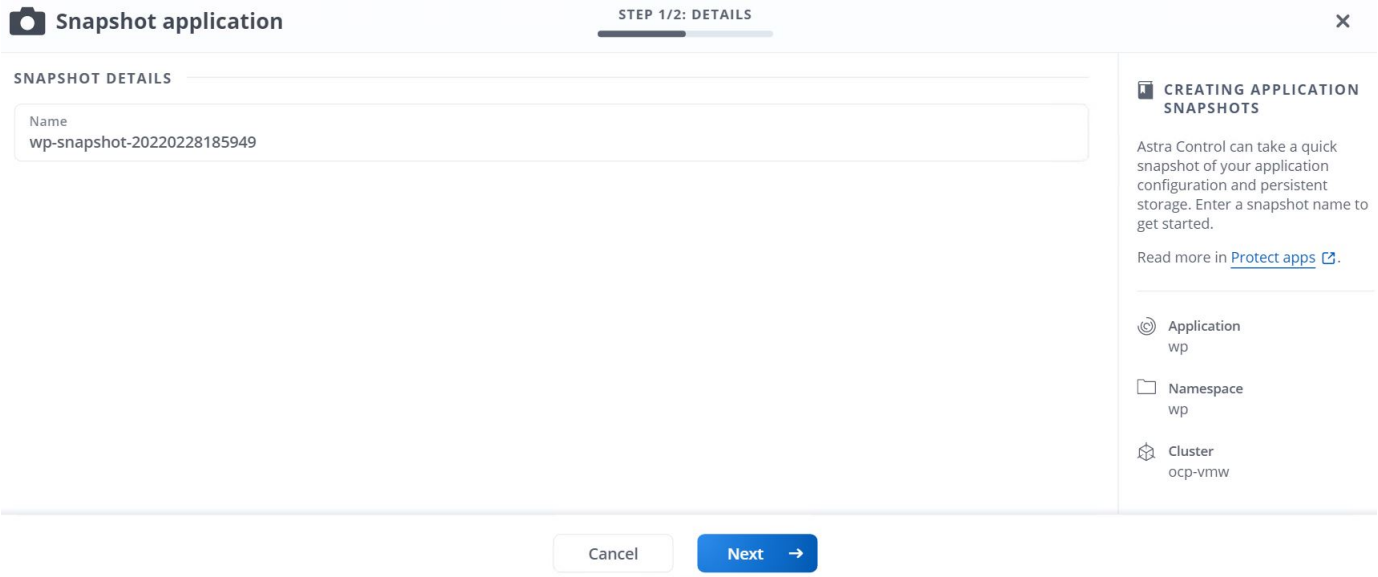
Creazione di un'istantanea dell'applicazione

Un'istantanea di un'applicazione crea una copia Snapshot di ONTAP che può essere utilizzata per ripristinare o clonare l'applicazione in un momento specifico in base a tale copia Snapshot.

1. Per creare un'istantanea dell'applicazione, accedere alla scheda applicazioni > gestite e fare clic sull'applicazione di cui si desidera creare una copia Snapshot. Fare clic sul menu a discesa accanto al nome dell'applicazione e fare clic su Snapshot.



2. Inserire i dettagli dell'istantanea, fare clic su Next (Avanti), quindi su Snapshot (istantanea). La creazione dello snapshot richiede circa un minuto e lo stato diventa disponibile dopo la creazione dello snapshot.



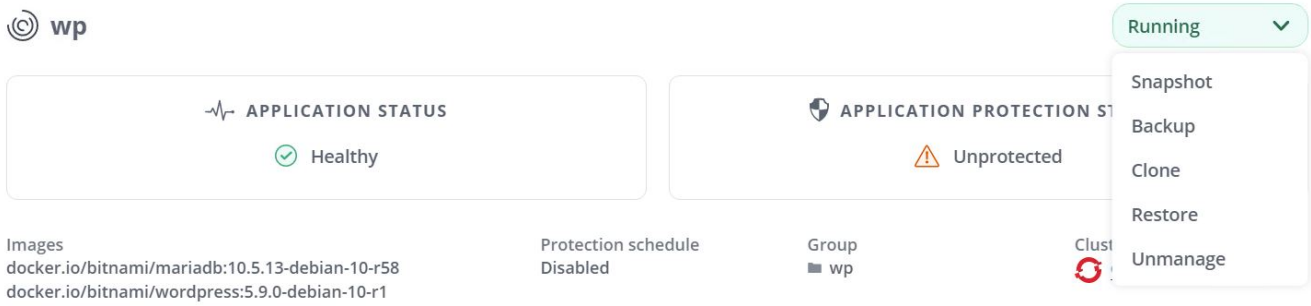
Creazione di un backup dell'applicazione

Un backup di un'applicazione acquisisce lo stato attivo dell'applicazione e la configurazione delle risorse IT, le taglia in file e le memorizza in un bucket di storage a oggetti remoto.

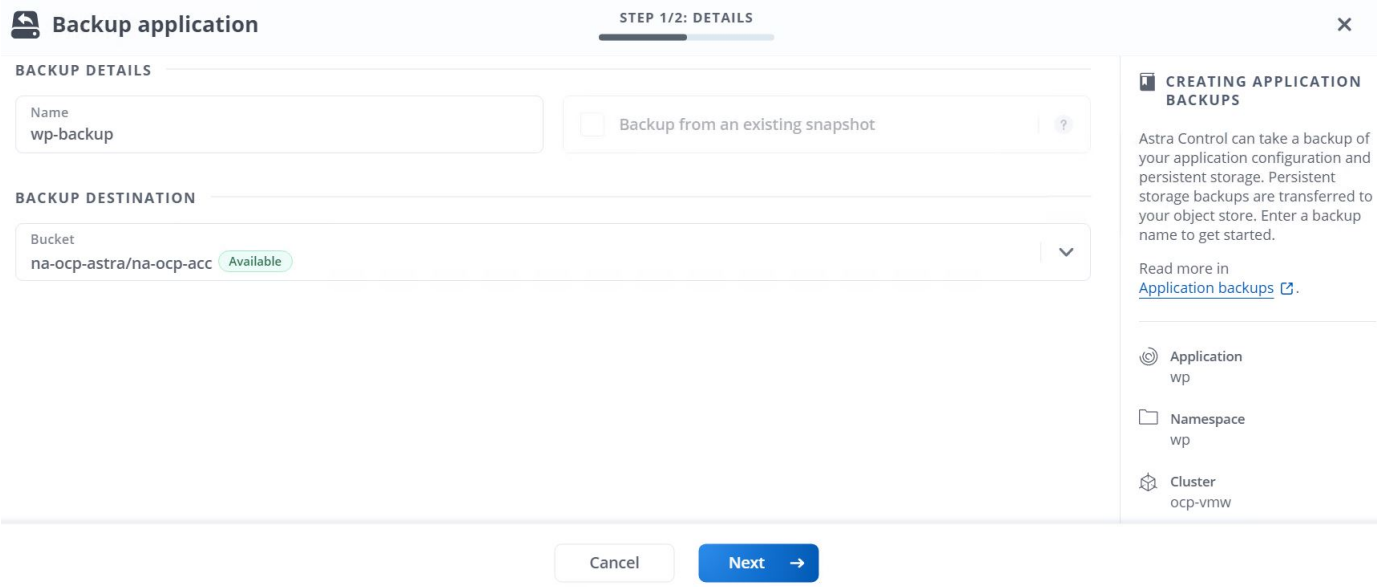
Per il backup e il ripristino delle applicazioni gestite nel centro di controllo Astra, è necessario configurare le impostazioni del superutente per i sistemi ONTAP di backup come prerequisito. A tale scopo, immettere i seguenti comandi.

```
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. Per creare un backup dell'applicazione gestita in Astra Control Center, accedere alla scheda Apps (applicazioni) > Managed (gestite) e fare clic sull'applicazione di cui si desidera eseguire il backup. Fare clic sul menu a discesa accanto al nome dell'applicazione e fare clic su Backup.



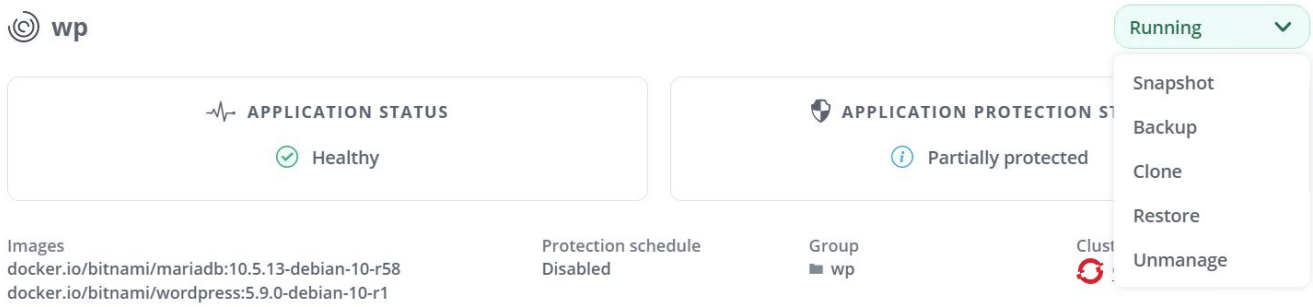
2. Inserire i dettagli del backup, selezionare il bucket di storage a oggetti in cui memorizzare i file di backup, fare clic su Next (Avanti) e, dopo aver esaminato i dettagli, fare clic su Backup (Backup). A seconda delle dimensioni dell'applicazione e dei dati, il backup può richiedere alcuni minuti e lo stato del backup diventa disponibile una volta completato correttamente il backup.



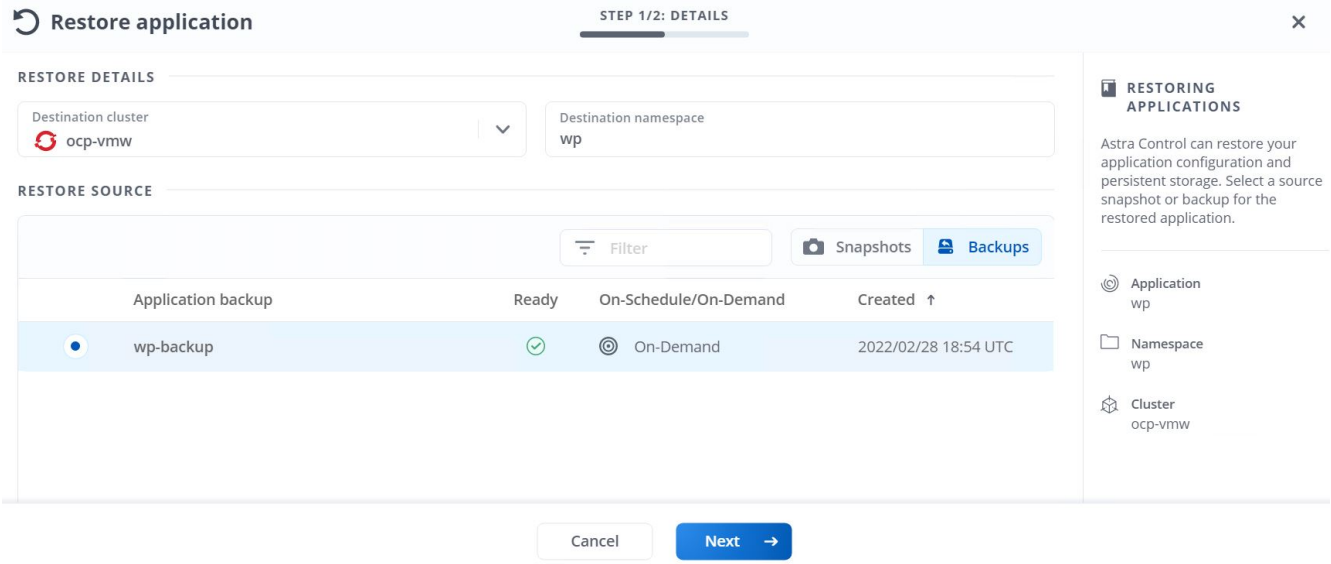
Ripristino di un’applicazione

Con la semplice pressione di un pulsante, è possibile ripristinare un’applicazione nello spazio dei nomi di origine nello stesso cluster o in un cluster remoto per la protezione delle applicazioni e il disaster recovery.

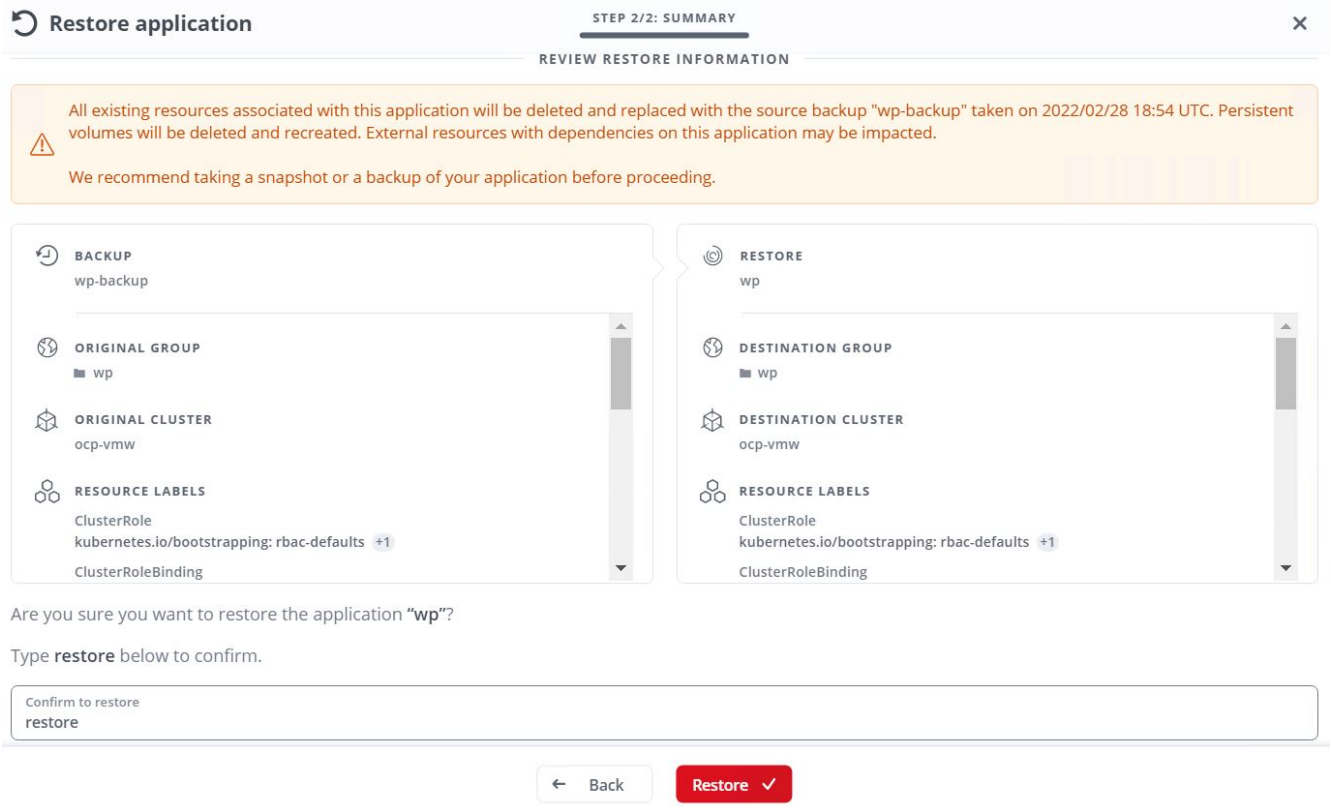
1. Per ripristinare un’applicazione, selezionare Apps (applicazioni) > Managed Tab (scheda gestita) e fare clic sull’applicazione in questione. Fare clic sul menu a discesa accanto al nome dell’applicazione e fare clic su `Restore`.



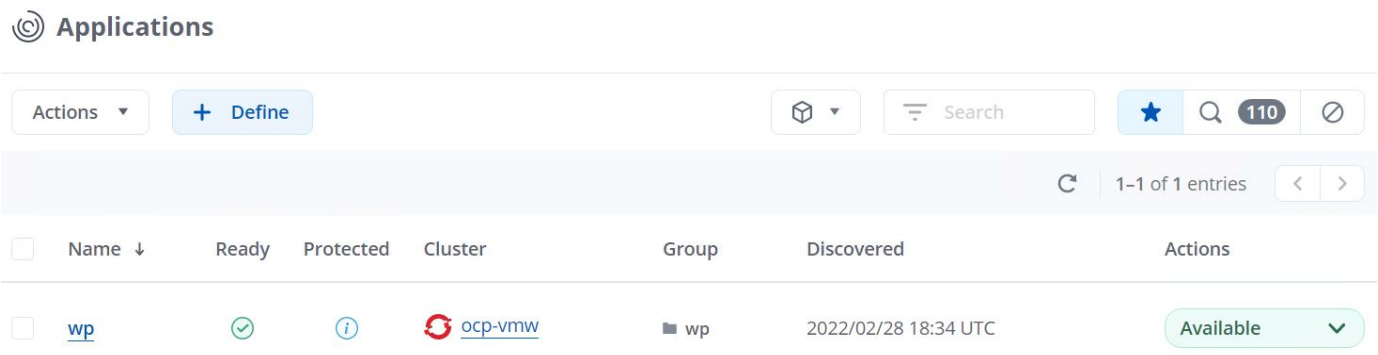
2. Immettere il nome dello spazio dei nomi di ripristino, selezionare il cluster in cui si desidera ripristinarlo e scegliere se si desidera ripristinarlo da uno snapshot esistente o da un backup dell’applicazione. Fare clic su Avanti.



3. Nel riquadro di revisione, immettere `restore` E fare clic su Restore (Ripristina) dopo aver esaminato i dettagli.



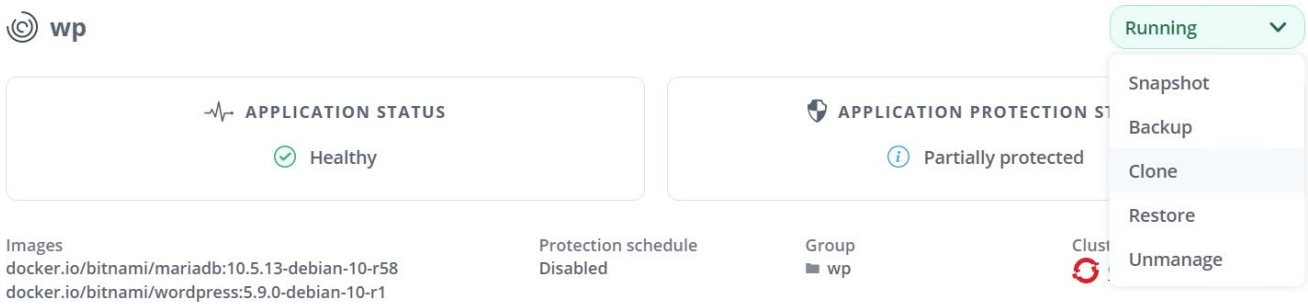
4. La nuova applicazione passa allo stato di ripristino mentre Astra Control Center ripristina l'applicazione sul cluster selezionato. Una volta installate e rilevate tutte le risorse dell'applicazione da Astra, l'applicazione passa allo stato Available (disponibile).



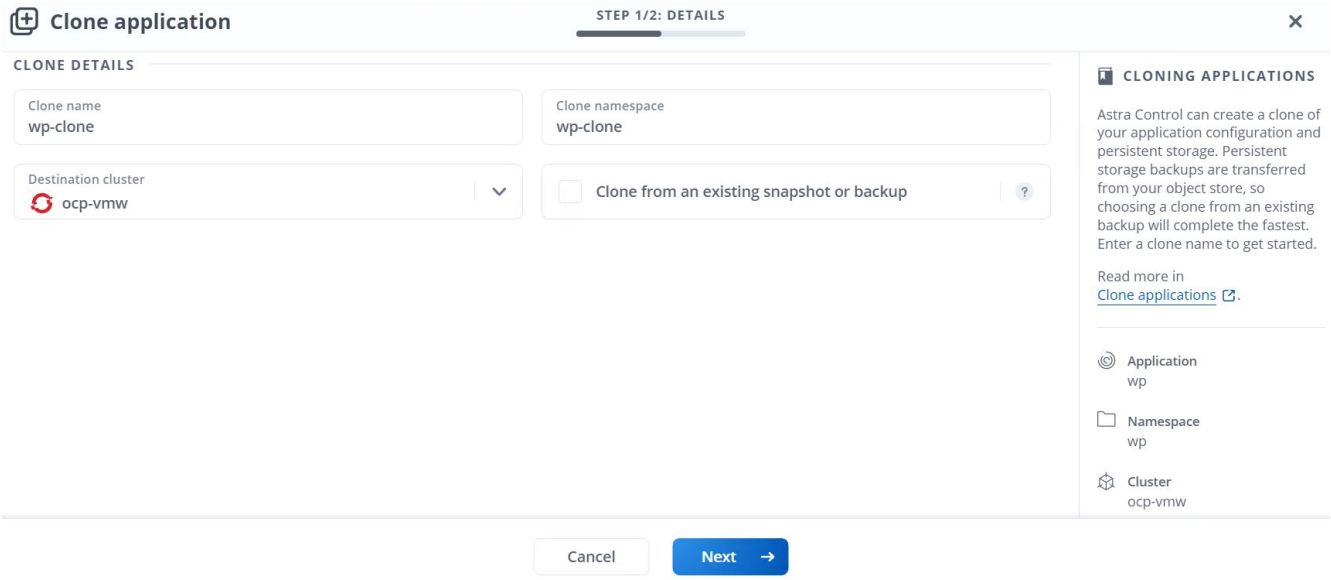
Clonare un'applicazione

È possibile clonare un'applicazione nel cluster di origine o in un cluster remoto per scopi di sviluppo/test o protezione dell'applicazione e disaster recovery. La clonazione di un'applicazione all'interno dello stesso cluster sullo stesso backend di storage utilizza la tecnologia NetApp FlexClone, che clona i PVC all'istante e consente di risparmiare spazio di storage.

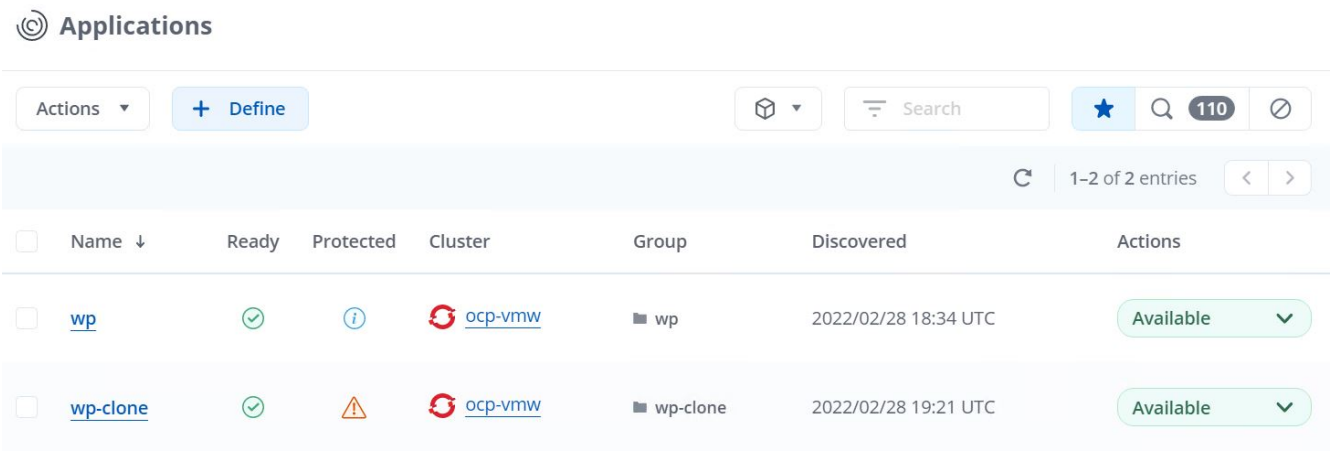
1. Per clonare un'applicazione, accedere alla scheda applicazioni > gestite e fare clic sull'applicazione in questione. Fare clic sul menu a discesa accanto al nome dell'applicazione e fare clic su Clone (Clona).



2. Immettere i dettagli del nuovo spazio dei nomi, selezionare il cluster in cui si desidera clonarlo e scegliere se clonarlo da uno snapshot esistente o da un backup o dallo stato corrente dell'applicazione. Quindi, fare clic su Next (Avanti) e su Clone on review pane (Clona sul pannello di revisione) dopo aver esaminato i dettagli.



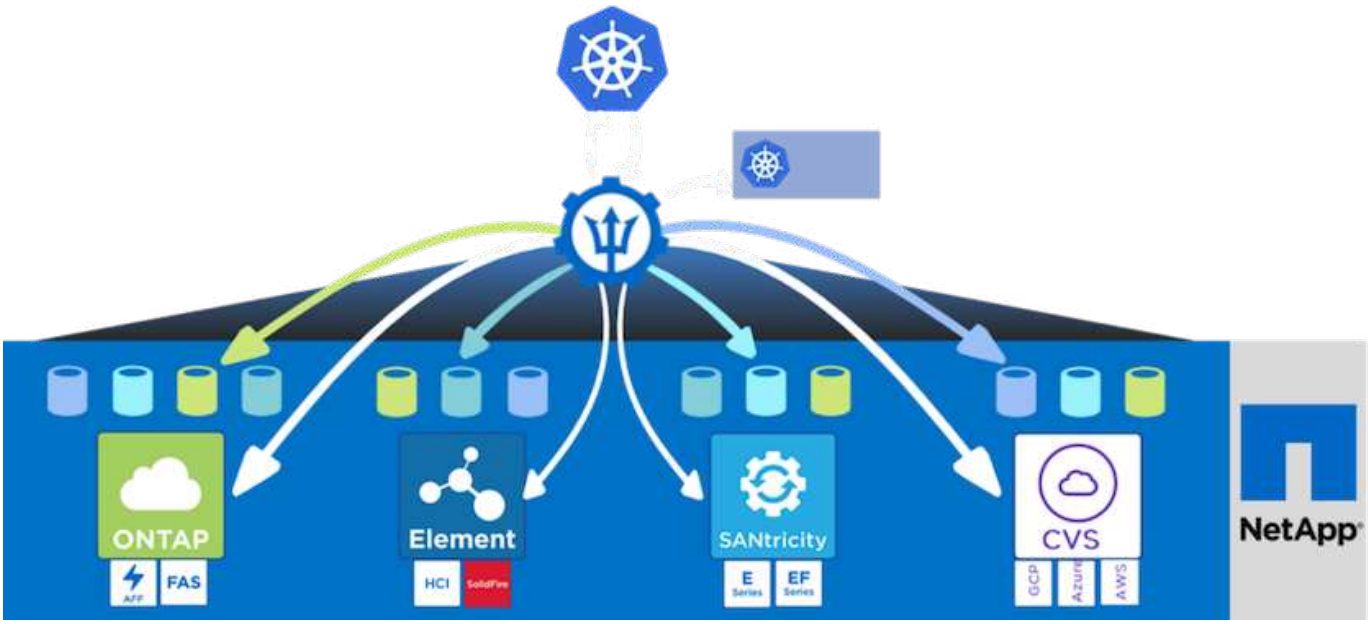
3. La nuova applicazione passa allo stato di rilevamento mentre Astra Control Center crea l'applicazione sul cluster selezionato. Una volta installate e rilevate tutte le risorse dell'applicazione da Astra, l'applicazione passa allo stato Available (disponibile).



Panoramica di Astra Trident

Astra Trident è un orchestrator di storage open-source e completamente supportato per container e distribuzioni Kubernetes, incluso Red Hat OpenShift. Trident lavora con l'intero portfolio di storage NetApp, inclusi i sistemi storage NetApp ONTAP ed Element, e supporta anche connessioni NFS e iSCSI. Trident accelera il workflow DevOps consentendo agli utenti finali di eseguire il provisioning e gestire lo storage dai sistemi storage NetApp senza richiedere l'intervento di un amministratore dello storage.

Un amministratore può configurare una serie di backend di storage in base alle esigenze di progetto e ai modelli di sistemi di storage che consentono funzionalità di storage avanzate, tra cui compressione, tipi di dischi specifici o livelli di QoS che garantiscono un certo livello di performance. Una volta definiti, questi backend possono essere utilizzati dagli sviluppatori nei loro progetti per creare dichiarazioni di volume persistenti (PVC) e per collegare storage persistente ai propri container on-demand.



Astra Trident ha un rapido ciclo di sviluppo e, proprio come Kubernetes, viene rilasciato quattro volte all'anno.

L'ultima versione di Astra Trident è la 22.01 rilasciata a gennaio 2022. Matrice di supporto per quale versione di Trident è stata testata con la quale è possibile trovare la distribuzione Kubernetes "qui".

A partire dalla versione 20.04, l'impostazione di Trident viene eseguita dall'operatore Trident. L'operatore semplifica le implementazioni su larga scala e fornisce supporto aggiuntivo, inclusa la riparazione automatica dei pod implementati nell'installazione di Trident.

Con la versione 21.01, è stato reso disponibile un grafico Helm per facilitare l'installazione dell'operatore Trident.

Scarica Astra Trident

Per installare Trident sul cluster di utenti implementato ed eseguire il provisioning di un volume persistente, attenersi alla seguente procedura:

1. Scaricare l'archivio di installazione sulla workstation di amministrazione ed estrarre il contenuto. La versione corrente di Trident è la 22.01, che può essere scaricata "qui".

```
[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. Estrarre l'installazione di Trident dal bundle scaricato.

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

Installare l'operatore Trident con Helm

1. Innanzitutto, impostare la posizione del cluster utente `kubeconfig` File come variabile di ambiente in modo da non doverla fare riferimento, perché Trident non ha alcuna opzione per passare questo file.

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

2. Eseguire il comando Helm per installare l'operatore Trident dal tarball nella directory helm durante la creazione dello spazio dei nomi Trident nel cluster di utenti.

```
[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. È possibile verificare che Trident sia installato correttamente controllando i pod in esecuzione nello spazio dei nomi o utilizzando il binario tridentctl per controllare la versione installata.

```
[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          |
+----------------+----------------+
```

In alcuni casi, gli ambienti dei clienti potrebbero richiedere la personalizzazione dell'implementazione di Trident. In questi casi, è anche possibile installare manualmente l'operatore Trident e aggiornare i manifesti inclusi per personalizzare l'implementazione.

Installare manualmente l'operatore Trident

1. Innanzitutto, impostare la posizione del cluster utente `kubeconfig` File come variabile di ambiente in modo da non doverla fare riferimento, perché Trident non ha alcuna opzione per passare questo file.

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

2. Il `trident-installer` la directory contiene i manifesti per la definizione di tutte le risorse richieste. Utilizzando i manifesti appropriati, creare `TridentOrchestrator` definizione personalizzata delle risorse.

```
[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. Se non ne esiste uno, creare uno spazio dei nomi Trident nel cluster utilizzando il manifesto fornito.

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

4. Creare le risorse necessarie per l'implementazione dell'operatore Trident, ad esempio un `ServiceAccount` per l'operatore, un `ClusterRole` e. `ClusterRoleBinding` al `ServiceAccount`, un

dedicato `PodSecurityPolicy`o l'operatore stesso.

```
[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. È possibile controllare lo stato dell'operatore dopo l'implementazione con i seguenti comandi:

```
[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. Con l'implementazione dell'operatore, ora possiamo utilizzarlo per installare Trident. Per eseguire questa operazione, è necessario creare un `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. È possibile verificare che Trident sia installato correttamente controllando i pod in esecuzione nello spazio dei nomi o utilizzando il binario tridentctl per controllare la versione installata.

```
[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          |
+----------------+----------------+
```

Preparare i nodi di lavoro per lo storage

NFS

La maggior parte delle distribuzioni Kubernetes viene fornita con i pacchetti e le utility per montare i backend NFS installati di default, incluso Red Hat OpenShift.

Tuttavia, per NFSv3, non esiste alcun meccanismo per negoziare la concorrenza tra il client e il server. Pertanto, il numero massimo di voci della tabella degli slot sunrpc lato client deve essere sincronizzato manualmente con il valore supportato sul server per garantire le migliori prestazioni per la connessione NFS

senza che il server debba ridurre le dimensioni della finestra della connessione.

Per ONTAP, il numero massimo supportato di voci della tabella degli slot sunrpc è 128, ovvero ONTAP può gestire 128 richieste NFS simultanee alla volta. Tuttavia, per impostazione predefinita, Red Hat CoreOS/Red Hat Enterprise Linux ha un massimo di 65,536 voci della tabella degli slot sunrpc per connessione. È necessario impostare questo valore su 128 e questo può essere fatto usando Machine Config Operator (MCO) in OpenShift.

Per modificare il numero massimo di voci della tabella degli slot sunrpc nei nodi di lavoro OpenShift, attenersi alla seguente procedura:

1. Accedere alla console Web di OCP e selezionare Compute > Machine Configs (calcolo > configurazioni macchina). Fare clic su Create Machine Config. Copiare e incollare il file YAML e fare clic su Create (Crea).

```
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. Dopo aver creato l'MCO, la configurazione deve essere applicata a tutti i nodi di lavoro e riavviata uno alla volta. L'intero processo richiede da 20 a 30 minuti circa. Verificare se la configurazione del computer viene applicata utilizzando `oc get mcp` e assicurarsi che il pool di configurazione del computer per i lavoratori sia aggiornato.

```
[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

Per preparare i nodi di lavoro per consentire la mappatura dei volumi di storage a blocchi tramite il protocollo iSCSI, è necessario installare i pacchetti necessari per supportare tale funzionalità.

In Red Hat OpenShift, questo viene gestito applicando un MCO (Machine Config Operator) al cluster dopo averlo implementato.

Per configurare i nodi di lavoro per l'esecuzione dei servizi iSCSI, attenersi alla seguente procedura:

1. Accedere alla console Web di OCP e selezionare Compute > Machine Configs (calcolo > configurazioni macchina). Fare clic su Create Machine Config. Copiare e incollare il file YAML e fare clic su Create (Crea).

   Quando non si utilizza il multipathing:

   ```
   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: ""
   ```

   Quando si utilizza il multipathing:

```
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. Una volta creata la configurazione, sono necessari circa 20 - 30 minuti per applicarla ai nodi di lavoro e ricaricarla. Verificare se la configurazione del computer viene applicata utilizzando `oc get mcp` e assicurarsi che il pool di configurazione del computer per i lavoratori sia aggiornato. È inoltre possibile accedere ai nodi di lavoro per confermare che il servizio iscsid è in esecuzione (e il servizio multipath è in esecuzione se si utilizza il multipathing).

```
[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
```

È inoltre possibile confermare che MachineConfig sia stato applicato correttamente e che i servizi siano stati avviati come previsto eseguendo il `oc debug` con i flag appropriati.

Creazione di backend per il sistema storage

Dopo aver completato l'installazione di Astra Trident Operator, è necessario configurare il backend per la piattaforma di storage NetApp specifica in uso. Seguire i collegamenti riportati di seguito per continuare

l'installazione e la configurazione di Astra Trident.

- "NetApp ONTAP NFS"
- "ISCSI NetApp ONTAP"
- "ISCSI NetApp Element"

Configurazione NFS di NetApp ONTAP

Per consentire l'integrazione di Trident con il sistema storage NetApp ONTAP, è necessario creare un backend che consenta la comunicazione con il sistema storage.

1. Nell'archivio di installazione scaricato in sono disponibili file backend di esempio `sample-input` gerarchia di cartelle. Per i sistemi NetApp ONTAP che servono NFS, copiare il `backend-ontap-nas.json` nella directory di lavoro e modificare il file.

```
[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. Modificare il backendName, managementLIF, dataLIF, svm, nome utente, e password in questo file.

```
{
    "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"
}
```

> È consigliabile definire il valore backendName personalizzato come combinazione di storageDriverName e dataLIF che fornisce NFS per una facile identificazione.

3. Una volta creato questo file di back-end, eseguire il comando seguente per creare il primo backend.

```
[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. Una volta creato il backend, è necessario creare una classe di storage. Come per il backend, esiste un file di esempio della classe di storage che può essere modificato per l'ambiente disponibile nella cartella di input di esempio. Copiarlo nella directory di lavoro e apportare le modifiche necessarie per riflettere il backend creato.

```
[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. L'unica modifica che deve essere apportata a questo file è definire `backendType` valore al nome del driver di storage dal backend appena creato. Annotare anche il valore del campo nome, a cui si deve fare riferimento in un passaggio successivo.

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

> Esiste un campo opzionale chiamato `fsType` definito in questo file. Questa riga può essere eliminata nei backend NFS.

6. Eseguire `oc` per creare la classe di storage.

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

7. Una volta creata la classe di storage, è necessario creare la prima dichiarazione di volume persistente (PVC). C'è un esempio `pvc-basic.yaml` file che può essere utilizzato per eseguire questa azione, disponibile anche in input di esempio.

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

8. L'unica modifica che deve essere apportata a questo file è garantire che il `storageClassName` il campo corrisponde a quello appena creato. La definizione PVC può essere ulteriormente personalizzata in base alle esigenze del carico di lavoro da fornire.

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

9. Creare il PVC emettendo il `oc` comando. La creazione può richiedere del tempo a seconda delle dimensioni del volume di backup da creare, in modo da poter guardare il processo mentre viene completato.

```
[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
```

Configurazione iSCSI di NetApp ONTAP

Per consentire l'integrazione di Trident con il sistema storage NetApp ONTAP, è necessario creare un backend che consenta la comunicazione con il sistema storage.

1. Nell'archivio di installazione scaricato in sono disponibili file backend di esempio `sample-input` gerarchia di cartelle. Per i sistemi NetApp ONTAP che utilizzano iSCSI, copiare il `backend-ontap-san.json` nella directory di lavoro e modificare il file.

```
[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. Modificare i valori di gestione LIF, dataLIF, svm, nome utente e password in questo file.

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

3. Una volta creato questo file di back-end, eseguire il comando seguente per creare il primo backend.

```
[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. Una volta creato il backend, è necessario creare una classe di storage. Come per il backend, esiste un file di esempio della classe di storage che può essere modificato per l'ambiente disponibile nella cartella di input di esempio. Copiarlo nella directory di lavoro e apportare le modifiche necessarie per riflettere il backend creato.

```
[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. L'unica modifica che deve essere apportata a questo file è definire `backendType` valore al nome del driver di storage dal backend appena creato. Annotare anche il valore del campo nome, a cui si deve fare riferimento in un passaggio successivo.

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

> Esiste un campo opzionale chiamato `fsType` definito in questo file. Nei backend iSCSI, questo valore può essere impostato su un tipo di filesystem Linux specifico (XFS, ext4, ecc.) o può essere cancellato per consentire a OpenShift di decidere quale filesystem usare.

6. Eseguire `oc` per creare la classe di storage.

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

7. Una volta creata la classe di storage, è necessario creare la prima dichiarazione di volume persistente (PVC). C'è un esempio `pvc-basic.yaml` file che può essere utilizzato per eseguire questa azione, disponibile anche in input di esempio.

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

8. L'unica modifica che deve essere apportata a questo file è garantire che il `storageClassName` il campo corrisponde a quello appena creato. La definizione PVC può essere ulteriormente personalizzata in base alle esigenze del carico di lavoro da fornire.

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

9. Creare il PVC emettendo il `oc` comando. La creazione può richiedere del tempo a seconda delle

dimensioni del volume di backup da creare, in modo da poter guardare il processo mentre viene completato.

```
[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
```

Configurazione iSCSI NetApp Element

Per abilitare l'integrazione di Trident con il sistema storage NetApp Element, è necessario creare un backend che consenta la comunicazione con il sistema storage utilizzando il protocollo iSCSI.

1. Nell'archivio di installazione scaricato in sono disponibili file backend di esempio `sample-input` gerarchia di cartelle. Per i sistemi NetApp Element che utilizzano iSCSI, copiare `backend-solidfire.json` nella directory di lavoro e modificare il file.

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

   a. Modificare i valori di utente, password e MVIP su `EndPoint` linea.
   b. Modificare il `SVIP` valore.

```
 {
    "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. Una volta creato questo file back-end, eseguire il seguente comando per creare il primo backend.

```
[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. Una volta creato il backend, è necessario creare una classe di storage. Come per il backend, esiste un file di esempio della classe di storage che può essere modificato per l'ambiente disponibile nella cartella di input di esempio. Copiarlo nella directory di lavoro e apportare le modifiche necessarie per riflettere il backend creato.

```
[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. L'unica modifica che deve essere apportata a questo file è definire `backendType` valore al nome del driver di storage dal backend appena creato. Annotare anche il valore del campo nome, a cui si deve fare riferimento in un passaggio successivo.

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

> Esiste un campo opzionale chiamato `fsType` definito in questo file. Nei backend iSCSI, questo valore può essere impostato su un tipo di filesystem Linux specifico (XFS, ext4 e così via), oppure può essere cancellato per consentire a OpenShift di decidere quale filesystem usare.

5. Eseguire `oc` per creare la classe di storage.

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

6. Una volta creata la classe di storage, è necessario creare la prima dichiarazione di volume persistente (PVC). C'è un esempio `pvc-basic.yaml` file che può essere utilizzato per eseguire questa azione, disponibile anche in input di esempio.

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

7. L'unica modifica che deve essere apportata a questo file è garantire che il `storageClassName` il campo corrisponde a quello appena creato. La definizione PVC può essere ulteriormente personalizzata in base alle esigenze del carico di lavoro da fornire.

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

8. Creare il PVC emettendo il `oc` comando. La creazione può richiedere del tempo a seconda delle dimensioni del volume di backup da creare, in modo da poter guardare il processo mentre viene completato.

```
[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
```