



Implementazione con l'operatore Trident

Astra Trident

NetApp

November 20, 2023

This PDF was generated from <https://docs.netapp.com/it-it/trident-2107/trident-get-started/kubernetes-customize-deploy.html> on November 20, 2023. Always check docs.netapp.com for the latest.

Sommario

- Implementazione con l'operatore Trident 1
 - Implementare l'operatore Trident utilizzando Helm 1
 - Implementare l'operatore Trident manualmente 2
 - Personalizzare l'implementazione dell'operatore Trident. 7

Implementazione con l'operatore Trident

Puoi implementare Astra Trident con l'operatore Trident. È possibile implementare l'operatore Trident manualmente o utilizzando Helm.



Se non si è ancora familiarizzato con il ["concetti di base"](#), è il momento ideale per farlo.

Di cosa hai bisogno

Per implementare Astra Trident, devono essere soddisfatti i seguenti prerequisiti:

- Si dispone dei privilegi completi per un cluster Kubernetes supportato che esegue Kubernetes 1.14 e versioni successive.
- Hai accesso a un sistema storage NetApp supportato.
- È possibile montare volumi da tutti i nodi di lavoro Kubernetes.
- Hai un host Linux con `kubectl` (o. oc, Se si utilizza OpenShift) installato e configurato per gestire il cluster Kubernetes che si desidera utilizzare.
- È stato impostato il `KUBECONFIG` Variabile di ambiente che punta alla configurazione del cluster Kubernetes.
- È stata attivata la ["Porte caratteristiche richieste da Astra Trident"](#).
- Se utilizzi Kubernetes con Docker Enterprise, ["Seguire la procedura per abilitare l'accesso CLI"](#).

Hai tutto questo? Fantastico! Iniziamo.

Implementare l'operatore Trident utilizzando Helm

Eseguire i passaggi elencati per implementare l'operatore Trident utilizzando Helm.

Di cosa hai bisogno

Oltre ai prerequisiti elencati in precedenza, per implementare l'operatore Trident utilizzando Helm, è necessario disporre di quanto segue:

- Kubernetes 1.17 e versioni successive
- Helm versione 3

Fasi

1. Scaricare il pacchetto di installazione dal ["Trident GitHub"](#) pagina. Il pacchetto di installazione include il grafico Helm in `/helm` directory.
2. Utilizzare `helm install` e specificare un nome per la distribuzione. Vedere il seguente esempio:

```
helm install <name> trident-operator-21.07.1.tgz --namespace <namespace you want to use for Trident>
```

Esistono due modi per passare i dati di configurazione durante l'installazione:

- `--values` (o. `-f`): Specificare un file YAML con override. Questo valore può essere specificato più volte e il file più a destra avrà la precedenza.

- `--set`: Specificare le sostituzioni sulla riga di comando.

Ad esempio, per modificare il valore predefinito di debug, eseguire quanto segue `--set` comando:

```
$ helm install <name> trident-operator-21.07.1.tgz --set tridentDebug=true
```

Il `values.yaml` file, che fa parte del grafico Helm, fornisce l'elenco delle chiavi e i relativi valori predefiniti.

`helm list` mostra i dettagli dell'installazione, ad esempio nome, spazio dei nomi, grafico, stato, versione dell'applicazione, numero di revisione e così via.

Implementare l'operatore Trident manualmente

Eseguire i passaggi elencati per implementare manualmente l'operatore Trident.

Fase 1: Qualificare il cluster Kubernetes

La prima cosa da fare è accedere all'host Linux e verificare che stia gestendo un ["Cluster Kubernetes supportato"](#) disporre dei privilegi necessari per.



Con OpenShift, utilizzare `oc` invece di `kubectl` in tutti gli esempi che seguono, accedere come **system:admin** eseguendo `oc login -u system:admin` oppure `oc login -u kube-admin`.

Per verificare se la versione di Kubernetes è successiva alla 1.14, eseguire il seguente comando:

```
kubectl version
```

Per verificare se si dispone dei privilegi di amministratore del cluster Kubernetes, eseguire il seguente comando:

```
kubectl auth can-i '*' '*' --all-namespaces
```

Per verificare se è possibile avviare un pod che utilizza un'immagine da Docker Hub e raggiungere il sistema di storage sulla rete pod, eseguire il seguente comando:

```
kubectl run -i --tty ping --image=busybox --restart=Never --rm -- \
ping <management IP>
```

Fase 2: Scaricare e configurare l'operatore



A partire da 21.01, l'operatore Trident ha un ambito cluster. L'utilizzo dell'operatore Trident per installare Trident richiede la creazione di `TridentOrchestrator` Definizione personalizzata delle risorse (CRD) e definizione di altre risorse. Prima di installare Astra Trident, eseguire questa procedura per configurare l'operatore.

1. Scaricare l'ultima versione di "[Pacchetto di installazione Trident](#)" Dalla sezione *Downloads* ed estrarla.

```
wget https://github.com/NetApp/trident/releases/download/v21.04/trident-
installer-21.04.tar.gz
tar -xf trident-installer-21.04.tar.gz
cd trident-installer
```

2. Utilizzare il manifesto CRD appropriato per creare `TridentOrchestrator` CRD. Quindi, creare un `TridentOrchestrator` Custom Resource in seguito per creare un'installazione da parte dell'operatore.

Eseguire il seguente comando:

```
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

3. Dopo il `TridentOrchestrator` Viene creato un CRD, creare le seguenti risorse necessarie per l'implementazione dell'operatore:

- Un account di servizio per l'operatore
- Un `ClusterRole` e `ClusterRoleBinding` al `ServiceAccount`
- Una policy `PodSecurityPolicy` dedicata
- L'operatore stesso

Il programma di installazione di Trident contiene i manifesti per la definizione di queste risorse. Per impostazione predefinita, l'operatore viene implementato in `trident` namespace. Se il `trident` namespace non esiste, utilizzare il seguente manifesto per crearne uno.

```
$ kubectl apply -f deploy/namespace.yaml
```

4. Per implementare l'operatore in uno spazio dei nomi diverso da quello predefinito `trident` namespace, è necessario aggiornare `serviceaccount.yaml`, `clusterrolebinding.yaml` e `operator.yaml` manifesta e genera il tuo `bundle.yaml`.

Eseguire il comando seguente per aggiornare i manifesti YAML e generare il `bundle.yaml` utilizzando il `kustomization.yaml`:

```
kubectl kustomize deploy/ > deploy/bundle.yaml
```

Eseguire il seguente comando per creare le risorse e implementare l'operatore:

```
kubectl create -f deploy/bundle.yaml
```

5. Per verificare lo stato dell'operatore dopo l'implementazione, procedere come segue:

```
$ kubectl get deployment -n <operator-namespace>
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
trident-operator    1/1      1              1            3m

$ kubectl get pods -n <operator-namespace>
NAME                                READY    STATUS      RESTARTS
AGE
trident-operator-54cb664d-lnjxh    1/1      Running      0
3m
```

L'implementazione dell'operatore crea correttamente un pod in esecuzione su uno dei nodi di lavoro nel cluster.



In un cluster Kubernetes dovrebbe esserci solo **un'istanza** dell'operatore. Non creare implementazioni multiple dell'operatore Trident.

Fase 3: Creazione `TridentOrchestrator` E installare Trident

Ora sei pronto per installare Astra Trident usando l'operatore! Per questo è necessario creare `TridentOrchestrator`. Il programma di installazione di Trident include definizioni di esempio per la creazione `TridentOrchestrator`. In questo modo viene eseguita un'installazione in `trident namespace`.

```

$ kubectl create -f deploy/crds/tridentorchestrator_cr.yaml
tridentorchestrator.trident.netapp.io/trident created

$ kubectl describe torc trident
Name:          trident
Namespace:
Labels:        <none>
Annotations:   <none>
API Version:   trident.netapp.io/v1
Kind:          TridentOrchestrator
...
Spec:
  Debug:      true
  Namespace:  trident
Status:
  Current Installation Params:
    IPv6:                false
    Autosupport Hostname:
    Autosupport Image:    netapp/trident-autosupport:21.04
    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:21.04.0
  Message:               Trident installed Namespace:
trident
  Status:                 Installed
  Version:                v21.04.0
Events:
  Type Reason Age From Message ----
  Installing 74s trident-operator.netapp.io Installing Trident Normal
  Installed 67s trident-operator.netapp.io Trident installed

```

L'operatore Trident consente di personalizzare il modo in cui Astra Trident viene installato utilizzando gli attributi in `TridentOrchestrator spec`. Vedere ["Personalizza la tua implementazione Trident"](#).

Lo Stato di `TridentOrchestrator` Indica se l'installazione ha avuto esito positivo e visualizza la versione di Trident installata.

Stato	Descrizione
Installazione in corso	L'operatore sta installando Astra Trident TridentOrchestrator CR.
Installato	Astra Trident è stato installato correttamente.
Disinstallazione in corso	L'operatore sta disinstallando Astra Trident, perché <code>spec.uninstall=true</code> .
Disinstallato	Astra Trident disinstallato.
Non riuscito	L'operatore non ha potuto installare, applicare patch, aggiornare o disinstallare Astra Trident; l'operatore tenterà automaticamente di eseguire il ripristino da questo stato. Se lo stato persiste, è necessario eseguire la risoluzione dei problemi.
Aggiornamento in corso	L'operatore sta aggiornando un'installazione esistente.
Errore	Il TridentOrchestrator non viene utilizzato. Un'altra esiste già.

Durante l'installazione, lo stato di `TridentOrchestrator` modificherà da `Installing` a `Installed`. Se si osserva `Failed` e l'operatore non è in grado di eseguire il ripristino da solo, è necessario controllare i registri dell'operatore. Vedere ["risoluzione dei problemi"](#) sezione.

Puoi verificare se l'installazione di Astra Trident è stata completata dando un'occhiata ai pod creati:

```
$ kubectl get pod -n trident
```

NAME	READY	STATUS	RESTARTS	AGE
trident-csi-7d466bf5c7-v4cpw	5/5	Running	0	1m
trident-csi-mr6zc	2/2	Running	0	1m
trident-csi-xrp7w	2/2	Running	0	1m
trident-csi-zh2jt	2/2	Running	0	1m
trident-operator-766f7b8658-ldzsv	1/1	Running	0	3m

È anche possibile utilizzare `tridentctl` Per verificare la versione di Astra Trident installata.

```
$ ./tridentctl -n trident version
```

+-----+	
SERVER VERSION	CLIENT VERSION
+-----+	
21.04.0	21.04.0
+-----+	

Ora puoi continuare a creare un back-end. Vedere ["attività post-implementazione"](#).



Per la risoluzione dei problemi durante l'implementazione, consultare ["risoluzione dei problemi"](#) sezione.

Personalizzare l'implementazione dell'operatore Trident

L'operatore Trident consente di personalizzare il modo in cui Astra Trident viene installato utilizzando gli attributi in `TridentOrchestrator spec`.

Per un elenco degli attributi, consultare la tabella seguente:

Parametro	Descrizione	Predefinito
<code>namespace</code>	Spazio dei nomi in cui installare Astra Trident	"predefinito"
<code>debug</code>	Attiva il debug per Astra Trident	falso
<code>IPv6</code>	Installare Astra Trident su IPv6	falso
<code>k8sTimeout</code>	Timeout per le operazioni Kubernetes	30 sec
<code>silenceAutosupport</code>	Non inviare pacchetti AutoSupport automaticamente a NetApp	falso
<code>enableNodePrep</code>	Gestire automaticamente le dipendenze dei nodi di lavoro (BETA)	falso
<code>autosupportImage</code>	L'immagine del contenitore per la telemetria AutoSupport	"netapp/trident-autosupport:21.04.0"
<code>autosupportProxy</code>	Indirizzo/porta di un proxy per l'invio di telemetria AutoSupport	"http://proxy.example.com:8888"
<code>uninstall</code>	Flag utilizzato per disinstallare Astra Trident	falso
<code>logFormat</code>	Formato di registrazione Astra Trident da utilizzare [text,json]	"testo"
<code>tridentImage</code>	Immagine Astra Trident da installare	"netapp/trident:21.04"
<code>imageRegistry</code>	Percorso al registro interno, del formato <registry FQDN>[:port] [/subpath]	"k8s.gcr.io/sig-storage (k8s 1.17+) o quay.io/k8scsi"
<code>kubeletDir</code>	Percorso della directory del kubelet sull'host	"/var/lib/kubelet"
<code>wipeout</code>	Un elenco di risorse da eliminare per eseguire una rimozione completa di Astra Trident	

Parametro	Descrizione	Predefinito
imagePullSecrets	Secrets (segreti) per estrarre immagini da un registro interno	



`spec.namespace` è specificato in `TridentOrchestrator` Per indicare in quale spazio dei nomi Astra Trident è installato. Questo parametro **non può essere aggiornato dopo l'installazione di Astra Trident**. Il tentativo di eseguire questa operazione causa lo stato di `TridentOrchestrator` per passare a `Failed`. Astra Trident non deve essere migrato tra spazi dei nomi.



La preparazione automatica dei nodi di lavoro è una funzionalità * beta* che deve essere utilizzata solo in ambienti non di produzione.

È possibile utilizzare gli attributi menzionati in precedenza per la definizione `TridentOrchestrator` per personalizzare l'installazione. Ecco un esempio:

```
$ cat deploy/crds/tridentorchestrator_cr_imagepullsecrets.yaml
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  tridentImage: netapp/trident:21.04.0
  imagePullSecrets:
    - thisisasecret
```

Se si desidera personalizzare l'installazione oltre ciò che si desidera `TridentOrchestrator` gli argomenti lo consentono, dovresti considerare di utilizzare `tridentctl` Per generare manifesti YAML personalizzati che è possibile modificare in base alle esigenze.

Informazioni sul copyright

Copyright © 2023 NetApp, Inc. Tutti i diritti riservati. Stampato negli Stati Uniti d'America. Nessuna porzione di questo documento soggetta a copyright può essere riprodotta in qualsiasi formato o mezzo (grafico, elettronico o meccanico, inclusi fotocopie, registrazione, nastri o storage in un sistema elettronico) senza previo consenso scritto da parte del detentore del copyright.

Il software derivato dal materiale sottoposto a copyright di NetApp è soggetto alla seguente licenza e dichiarazione di non responsabilità:

IL PRESENTE SOFTWARE VIENE FORNITO DA NETAPP "COSÌ COM'È" E SENZA QUALSIVOGLIA TIPO DI GARANZIA IMPLICITA O ESPRESSA FRA CUI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ E IDONEITÀ PER UNO SCOPO SPECIFICO, CHE VENGONO DECLINATE DAL PRESENTE DOCUMENTO. NETAPP NON VERRÀ CONSIDERATA RESPONSABILE IN ALCUN CASO PER QUALSIVOGLIA DANNO DIRETTO, INDIRETTO, ACCIDENTALE, SPECIALE, ESEMPLARE E CONSEGUENZIALE (COMPRESI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, PROCUREMENT O SOSTITUZIONE DI MERCI O SERVIZI, IMPOSSIBILITÀ DI UTILIZZO O PERDITA DI DATI O PROFITTI OPPURE INTERRUZIONE DELL'ATTIVITÀ AZIENDALE) CAUSATO IN QUALSIVOGLIA MODO O IN RELAZIONE A QUALUNQUE TEORIA DI RESPONSABILITÀ, SIA ESSA CONTRATTUALE, RIGOROSA O DOVUTA A INSOLVENZA (COMPRESA LA NEGLIGENZA O ALTRO) INSORTA IN QUALSIASI MODO ATTRAVERSO L'UTILIZZO DEL PRESENTE SOFTWARE ANCHE IN PRESENZA DI UN PREAVVISO CIRCA L'EVENTUALITÀ DI QUESTO TIPO DI DANNI.

NetApp si riserva il diritto di modificare in qualsiasi momento qualunque prodotto descritto nel presente documento senza fornire alcun preavviso. NetApp non si assume alcuna responsabilità circa l'utilizzo dei prodotti o materiali descritti nel presente documento, con l'eccezione di quanto concordato espressamente e per iscritto da NetApp. L'utilizzo o l'acquisto del presente prodotto non comporta il rilascio di una licenza nell'ambito di un qualche diritto di brevetto, marchio commerciale o altro diritto di proprietà intellettuale di NetApp.

Il prodotto descritto in questa guida può essere protetto da uno o più brevetti degli Stati Uniti, esteri o in attesa di approvazione.

LEGENDA PER I DIRITTI SOTTOPOSTI A LIMITAZIONE: l'utilizzo, la duplicazione o la divulgazione da parte degli enti governativi sono soggetti alle limitazioni indicate nel sottoparagrafo (b)(3) della clausola Rights in Technical Data and Computer Software del DFARS 252.227-7013 (FEB 2014) e FAR 52.227-19 (DIC 2007).

I dati contenuti nel presente documento riguardano un articolo commerciale (secondo la definizione data in FAR 2.101) e sono di proprietà di NetApp, Inc. Tutti i dati tecnici e il software NetApp forniti secondo i termini del presente Contratto sono articoli aventi natura commerciale, sviluppati con finanziamenti esclusivamente privati. Il governo statunitense ha una licenza irrevocabile limitata, non esclusiva, non trasferibile, non cedibile, mondiale, per l'utilizzo dei Dati esclusivamente in connessione con e a supporto di un contratto governativo statunitense in base al quale i Dati sono distribuiti. Con la sola esclusione di quanto indicato nel presente documento, i Dati non possono essere utilizzati, divulgati, riprodotti, modificati, visualizzati o mostrati senza la previa approvazione scritta di NetApp, Inc. I diritti di licenza del governo degli Stati Uniti per il Dipartimento della Difesa sono limitati ai diritti identificati nella clausola DFARS 252.227-7015(b) (FEB 2014).

Informazioni sul marchio commerciale

NETAPP, il logo NETAPP e i marchi elencati alla pagina <http://www.netapp.com/TM> sono marchi di NetApp, Inc. Gli altri nomi di aziende e prodotti potrebbero essere marchi dei rispettivi proprietari.