



NetApp Astra Trident – Überblick

NetApp Solutions

NetApp
April 26, 2024

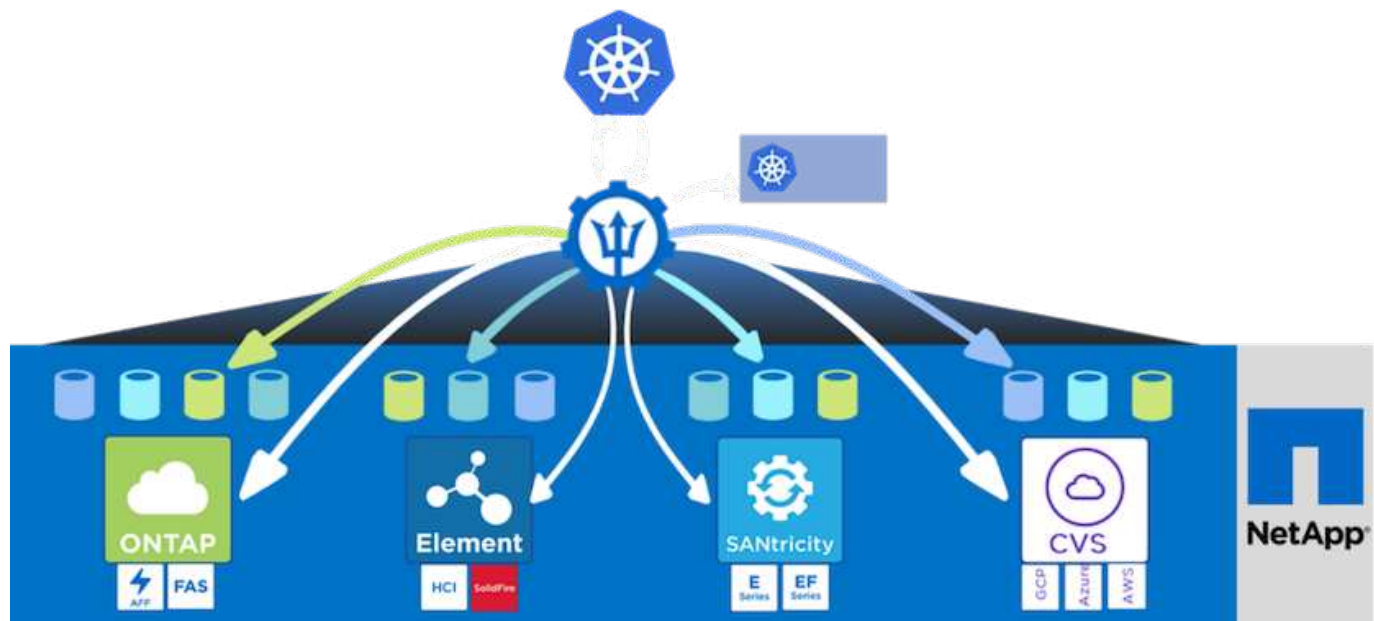
Inhalt

- Astra Trident – Überblick 1
 - Laden Sie Astra Trident Herunter 1
 - Installieren Sie den Trident Operator mit Helm 3
 - Trident Operator kann manuell installiert werden 5
 - Worker-Nodes für Storage vorbereiten 8
 - Erstellen von Storage-System-Back-Ends 13
 - Konfiguration von NetApp ONTAP NFS 13
 - NetApp ONTAP iSCSI-Konfiguration 15
 - iSCSI-Konfiguration von NetApp Element 18

Astra Trident – Überblick

Astra Trident ist ein Open-Source- und vollständig unterstützter Storage-Orchestrator für Container und Kubernetes-Distributionen, einschließlich Red hat OpenShift. Trident kann mit dem gesamten NetApp Storage-Portfolio eingesetzt werden, einschließlich NetApp ONTAP und Element Storage-Systeme. Es unterstützt auch NFS- und iSCSI-Verbindungen. Trident beschleunigt den DevOps-Workflow, da Endbenutzer Storage über ihre NetApp Storage-Systeme bereitstellen und managen können, ohne dass ein Storage-Administrator eingreifen muss.

Ein Administrator kann verschiedene Storage-Back-Ends basierend auf den Projektanforderungen und Storage-Systemmodellen konfigurieren, die erweiterte Storage-Funktionen wie Komprimierung, bestimmte Festplattentypen oder QoS-Level ermöglichen, die eine bestimmte Performance garantieren. Nach ihrer Definition können diese Back-Ends von Entwicklern in ihren Projekten verwendet werden, um persistente Volume Claims (PVCs) zu erstellen und persistenten Storage nach Bedarf an ihre Container anzubinden.



Astra Trident verfügt über einen schnellen Entwicklungszyklus, und genau wie Kubernetes bereits viermal im Jahr veröffentlicht.

Die neueste Version von Astra Trident ist die 22.01 Version von Januar 2022. Eine Support-Matrix, in der die Version von Trident getestet wurde, mit der Kubernetes Distribution zu finden ist ["Hier"](#).

Ab Version 20.04 wird die Trident-Einrichtung vom Trident Operator durchgeführt. Der Operator vereinfacht umfangreiche Implementierungen und bietet zusätzlichen Support einschließlich Selbstreparatur für Pods, die im Rahmen der Trident-Installation bereitgestellt werden.

In der Version 21.01 wurde ein Helm Chart zur Erleichterung der Installation des Trident Operators zur Verfügung gestellt.

Laden Sie Astra Trident Herunter

Um Trident auf dem implementierten Benutzer-Cluster zu installieren und ein persistentes Volume bereitzustellen, gehen Sie die folgenden Schritte durch:

1. Laden Sie das Installationsarchiv auf die Admin-Arbeitsstation herunter und extrahieren Sie den Inhalt. Die

aktuelle Version von Trident ist 22.01, die heruntergeladen werden kann ["Hier"](#).

```
[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. Extrahieren Sie die Trident Installation aus dem heruntergeladenen Paket.

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

Installieren Sie den Trident Operator mit Helm

1. Legen Sie zunächst den Speicherort des Benutzer-Clusters fest `kubeconfig` Datei als Umgebungsvariable, damit Sie nicht darauf verweisen müssen, weil Trident keine Option hat, diese Datei zu übergeben.

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

2. Führen Sie den Helm-Befehl aus, um den Trident-Operator aus dem Tarball im Steuerverzeichnis zu installieren, während der Dreizack-Namespace in Ihrem Benutzer-Cluster erstellt wird.

```
[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. Sie können überprüfen, ob Trident erfolgreich installiert wurde, indem Sie die Pods prüfen, die im Namespace ausgeführt werden, oder die tridentctl-Binärdatei verwenden, um die installierte Version zu überprüfen.

```
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
```

NAME	READY	STATUS	RESTARTS	AGE
trident-csi-5z451	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 einigen Fällen müssen die Kundenumgebungen möglicherweise die Anpassungen der Trident Implementierung erfordern. In diesen Fällen kann der Trident-Operator manuell installiert und die enthaltenen Manifeste aktualisiert werden, um die Implementierung anzupassen.

Trident Operator kann manuell installiert werden

1. Legen Sie zunächst den Speicherort des Benutzer-Clusters fest kubeconfig Datei als Umgebungsvariable, damit Sie nicht darauf verweisen müssen, weil Trident keine Option hat, diese Datei zu übergeben.

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

2. Der trident-installer Das Verzeichnis enthält Manifeste für die Definition aller erforderlichen Ressourcen. Erstellen Sie mithilfe der entsprechenden Manifeste das TridentOrchestrator Benutzerdefinierte Ressourcendefinition.

```
[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. Wenn nicht vorhanden ist, erstellen Sie mithilfe des angegebenen Manifests einen Trident Namespace im Cluster.

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

4. Erstellen Sie die Ressourcen, die für die Trident-Operator, wie z. B. ein, erforderlich sind ServiceAccount Für den Operator A ClusterRole Und ClusterRoleBinding Bis zum ServiceAccount, Eine engagierte PodSecurityPolicy, Oder der Operator selbst.

```
[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. Sie können den Status des Bedieners überprüfen, nachdem er mit den folgenden Befehlen bereitgestellt wurde:

```
[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. Mit dem implementierten Operator können wir nun Trident installieren. Dazu muss ein erstellt werden 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:      kubect1-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. Sie können überprüfen, ob Trident erfolgreich installiert wurde, indem Sie die Pods prüfen, die im Namespace ausgeführt werden, oder die tridentctl-Binärdatei verwenden, um die installierte Version zu überprüfen.

```

[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        |
+-----+

```

Worker-Nodes für Storage vorbereiten

NFS

Bei den meisten Kubernetes-Distributionen kommen Pakete und Utilities zur standardmäßig installierten NFS-Back-Ends einschließlich Red hat OpenShift zum Einsatz.

Bei NFSv3 gibt es jedoch keinen Mechanismus, um die Parallelität zwischen dem Client und dem Server auszuhandeln. Daher muss die maximale Anzahl der clientseitigen sunrpc-Slot-Tabelleneinträge manuell mit dem unterstützten Wert auf dem Server synchronisiert werden, um die beste Leistung für die NFS-Verbindung zu gewährleisten, ohne dass der Server die Fenstergröße der Verbindung verringern muss.

Bei ONTAP ist die unterstützte maximale Anzahl von sunrpc-Slot-Tabelleneinträgen 128, d.h. ONTAP kann 128 gleichzeitige NFS-Anfragen gleichzeitig verarbeiten. Standardmäßig hat Red hat CoreOS/Red hat Enterprise Linux jedoch maximal 65,536 Sunrpc Slot-Tabelleneinträge pro Verbindung. Dieser Wert muss auf 128 gesetzt werden. Dies kann mit Machine Config Operator (MCO) in OpenShift geschehen.

Gehen Sie wie folgt vor, um die maximalen Einträge in den OpenShift Worker Nodes zu ändern:

1. Melden Sie sich bei der OCP-Webkonsole an, und navigieren Sie zu „Compute“ > „Machine Configs“. Klicken Sie Auf Maschinenkonfiguration Erstellen. Kopieren Sie die YAML-Datei und fügen Sie sie ein, und klicken Sie auf Erstellen.

```
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. Nach der Erstellung des MCO muss die Konfiguration auf alle Arbeitsknoten angewendet und nacheinander neu gestartet werden. Der gesamte Vorgang dauert etwa 20 bis 30 Minuten. Überprüfen Sie, ob die Maschinenkonfiguration mit angewendet wird `oc get mcp` Und stellen Sie sicher, dass der Konfigurationspool für die Maschinenkonfiguration für die Arbeitnehmer aktualisiert wird.

```
[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

Um die Worker-Knoten vorzubereiten, die die Zuordnung von Block-Speicher-Volumes über das iSCSI-Protokoll ermöglichen, müssen Sie die erforderlichen Pakete installieren, um diese Funktionalität zu unterstützen.

In Red hat OpenShift wird dieser Vorgang durch Anwendung eines MCO (Machine Config Operator) auf das Cluster durchgeführt, nachdem es bereitgestellt wurde.

Führen Sie die folgenden Schritte aus, um die Worker-Knoten für die Ausführung von iSCSI-Diensten zu konfigurieren:

1. Melden Sie sich bei der OCP-Webkonsole an, und navigieren Sie zu „Compute“ > „Machine Configs“. Klicken Sie Auf Maschinenkonfiguration Erstellen. Kopieren Sie die YAML-Datei und fügen Sie sie ein, und klicken Sie auf Erstellen.

Wenn Sie kein Multipathing verwenden:

```
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: ""
```

Bei Verwendung von 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,ZGVmYXVsdHMgewogICAgICAgIHVzZXJfZnJpZW5kbHlfbmFtZXMgYm8KICAgICAgICBmaW5kX211bHRpcGF0aHMGbm8KfQoKYmxhY2tsaXN0X2V4Y2VwdGlvbnMGewogICAgICAgIHByb3BlcnR5ICIoU0NTSV9JREV0VF98SURfV1dOKSIKfQoKYmxhY2tsaXN0IHsKfQoK
          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. Nach der Erstellung der Konfiguration dauert es etwa 20 bis 30 Minuten, die Konfiguration auf die Worker-Nodes anzuwenden und erneut zu laden. Überprüfen Sie, ob die Maschinenkonfiguration mit angewendet wird `oc get mcp` Und stellen Sie sicher, dass der Konfigurationspool für die Maschinenkonfiguration für die Arbeitnehmer aktualisiert wird. Sie können sich auch bei den Worker-Nodes anmelden, um zu bestätigen, dass der iscsid-Service ausgeführt wird (und der Multipathd-Dienst wird ausgeführt, wenn Multipathing verwendet wird).

```
[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
```



Es ist auch möglich zu bestätigen, dass die MachineConfig erfolgreich angewendet wurde und die Dienste wie erwartet durch Ausführen der gestartet wurden `oc debug` Befehl mit den entsprechenden Flags.

Erstellen von Storage-System-Back-Ends

Nach Abschluss der Installation des Astra Trident Operator müssen Sie das Backend für die spezifische NetApp Storage-Plattform konfigurieren, die Sie verwenden. Folgen Sie den Links unten, um mit der Einrichtung und Konfiguration von Astra Trident fortzufahren.

- ["NetApp ONTAP NFS"](#)
- ["NetApp ONTAP iSCSI"](#)
- ["NetApp Element iSCSI"](#)

Konfiguration von NetApp ONTAP NFS

Um die Trident Integration mit dem NetApp ONTAP Storage-System zu aktivieren, müssen Sie ein Back-End erstellen, das die Kommunikation mit dem Storage-System ermöglicht.

1. Im heruntergeladenen Installationsarchiv stehen Beispiele für Backend-Dateien zur Verfügung `sample-input` Ordnerhierarchie. Kopieren Sie für NetApp ONTAP-Systeme, die NFS bereitstellen, den `backend-ontap-nas.json` Datei in Ihr Arbeitsverzeichnis und bearbeiten Sie die Datei.

```
[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. `BackendName`, `Management LIF`, `Daten LIF`, `svm`, `Benutzername`, bearbeiten Und `Kennwort`werte in dieser Datei.

```
{
  "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"
}
```



Als Best Practice empfiehlt es sich, den benutzerdefinierten `BackendName`-Wert als Kombination aus `storageDriverName` und der `DatenLIF` zu definieren, die NFS bedienen, um die einfache Identifizierung zu erleichtern.

3. Führen Sie mit dieser Backend-Datei den folgenden Befehl aus, um Ihr erstes Backend zu erstellen.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-nas.json
```

NAME	STATE	VOLUMES	STORAGE DRIVER	UUID
ontap-nas+10.61.181.221	online	0	ontap-nas	be7a619d-c81d-445c-b80c-5c87a73c5b1e

4. Wenn das Back-End erstellt wird, müssen Sie als nächstes eine Storage-Klasse erstellen. Wie beim Backend gibt es auch eine Beispiel-Speicherklassendatei, die für die im Ordner Sample-Inputs verfügbare Umgebung bearbeitet werden kann. Kopieren Sie ihn in das Arbeitsverzeichnis und nehmen Sie die erforderlichen Änderungen an dem erstellten Backend vor.

```
[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. Die einzige Bearbeitung, die zu dieser Datei gemacht werden muss, ist das zu definieren `backendType` Wert für den Namen des Speichertreibers aus dem neu erstellten Back-End. Notieren Sie auch den Wert des Namensfelds, auf den in einem späteren Schritt verwiesen werden muss.

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



Es gibt ein optionales Feld mit dem Namen `fsType` Das ist in dieser Datei definiert. Diese Zeile kann in NFS-Back-Ends gelöscht werden.

6. Führen Sie die aus `oc` Befehl zum Erstellen der Storage-Klasse.

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


7. Nach Erstellung der Storage-Klasse müssen Sie dann die erste Forderung für ein persistentes Volume (PVC) erstellen. Es gibt ein Beispiel `pvc-basic.yaml` Datei, mit der diese Aktion ausgeführt werden kann, die sich auch in Sample-Inputs befindet.

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

8. Die einzige Bearbeitung, die zu dieser Datei gemacht werden muss, ist sicherzustellen, dass die `storageClassName` Feld entspricht dem gerade erstellten. Die PVC-Definition kann je nach Bedarf des bereitzustellenden Workloads weiter angepasst werden.

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

9. Erstellen Sie das PVC, indem Sie die ausstellen `oc` Befehl. Die Erstellung kann je nach Größe des erstellten Sicherungsvolumens einige Zeit in Anspruch nehmen, sodass Sie den Prozess nach Abschluss beobachten können.

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

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

NetApp ONTAP iSCSI-Konfiguration

Um die Trident Integration mit dem NetApp ONTAP Storage-System zu aktivieren, müssen Sie ein Back-End erstellen, das die Kommunikation mit dem Storage-System ermöglicht.

1. Im heruntergeladenen Installationsarchiv stehen Beispiele für Backend-Dateien zur Verfügung `sample-input` Ordnerhierarchie. Kopieren Sie bei NetApp ONTAP-Systemen, die iSCSI bereitstellen, das

backend-ontap-san.json Datei in Ihr Arbeitsverzeichnis und bearbeiten Sie die Datei.

```
[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. Bearbeiten Sie die Werte ManagementLIF, dataLIF, svm, Benutzername und Passwort in dieser Datei.

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

3. Führen Sie mit dieser Backend-Datei den folgenden Befehl aus, um Ihr erstes Backend zu erstellen.

```
[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. Wenn das Back-End erstellt wird, müssen Sie als nächstes eine Storage-Klasse erstellen. Wie beim Backend gibt es auch eine Beispiel-Speicherklassendatei, die für die im Ordner Sample-Inputs verfügbare Umgebung bearbeitet werden kann. Kopieren Sie ihn in das Arbeitsverzeichnis und nehmen Sie die erforderlichen Änderungen an dem erstellten Backend vor.

```
[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. Die einzige Bearbeitung, die zu dieser Datei gemacht werden muss, ist das zu definieren backendType

Wert für den Namen des Speichertreibers aus dem neu erstellten Back-End. Notieren Sie auch den Wert des Namensfelds, auf den in einem späteren Schritt verwiesen werden muss.

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



Es gibt ein optionales Feld mit dem Namen `fsType`. Das ist in dieser Datei definiert. In iSCSI-Back-Ends kann dieser Wert auf einen bestimmten Linux-Dateisystem-Typ (XFS, ext4 usw.) gesetzt oder gelöscht werden, damit OpenShift entscheiden kann, welches Dateisystem verwendet werden soll.

6. Führen Sie die aus `oc` Befehl zum Erstellen der Storage-Klasse.

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

7. Nach Erstellung der Storage-Klasse müssen Sie dann die erste Forderung für ein persistentes Volume (PVC) erstellen. Es gibt ein Beispiel `pvc-basic.yaml` Datei, mit der diese Aktion ausgeführt werden kann, die sich auch in `Sample-Inputs` befindet.

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

8. Die einzige Bearbeitung, die zu dieser Datei gemacht werden muss, ist sicherzustellen, dass die `storageClassName` Feld entspricht dem gerade erstellten. Die PVC-Definition kann je nach Bedarf des bereitzustellenden Workloads weiter angepasst werden.

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

9. Erstellen Sie das PVC, indem Sie die ausstellen `oc` Befehl. Die Erstellung kann je nach Größe des erstellten Sicherungsvolumens einige Zeit in Anspruch nehmen, sodass Sie den Prozess nach Abschluss beobachten können.

```
[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
basic	Bound	pvc-7ceac1ba-0189-43c7-8f98-094719f7956c	1Gi

```
ACCESS MODES   STORAGECLASS  AGE
basic          basic-csi     3s
```

ISCSI-Konfiguration von NetApp Element

Um die Trident Integration mit dem NetApp Element Storage-System zu aktivieren, müssen Sie ein Backend erstellen, das die Kommunikation mit dem Storage-System über das iSCSI-Protokoll ermöglicht.

1. Im heruntergeladenen Installationsarchiv stehen Beispiele für Backend-Dateien zur Verfügung `sample-input` Ordnerhierarchie. Kopieren Sie für NetApp Element-Systeme, die iSCSI-Server bereitstellen, das `backend-solidfire.json` Datei in Ihr Arbeitsverzeichnis und bearbeiten Sie die Datei.

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

- a. Bearbeiten Sie den Benutzer-, das Kennwort und den MVIP-Wert auf dem `EndPoint` Linie.
- b. Bearbeiten Sie das `SVIP` Wert:

```
{
  "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. Führen Sie mit dieser Back-End-Datei den folgenden Befehl aus, um Ihr erstes Backend zu erstellen.

```
[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. Wenn das Back-End erstellt wird, müssen Sie als nächstes eine Storage-Klasse erstellen. Wie beim Backend gibt es auch eine Beispiel-Speicherklassendatei, die für die im Ordner Sample-Inputs verfügbare Umgebung bearbeitet werden kann. Kopieren Sie ihn in das Arbeitsverzeichnis und nehmen Sie die erforderlichen Änderungen an dem erstellten Backend vor.

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

4. Die einzige Bearbeitung, die zu dieser Datei gemacht werden muss, ist das zu definieren backendType Wert für den Namen des Speichertreibers aus dem neu erstellten Back-End. Notieren Sie auch den Wert des Namensfelds, auf den in einem späteren Schritt verwiesen werden muss.

```

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

```



Es gibt ein optionales Feld mit dem Namen `fsType`. Das ist in dieser Datei definiert. In iSCSI-Back-Ends kann dieser Wert auf einen bestimmten Linux-Dateisystem-Typ (XFS, ext4 usw.) gesetzt werden. OpenShift kann auch gelöscht werden, damit OpenShift entscheiden kann, welches Dateisystem verwendet werden soll.

5. Führen Sie die aus `oc` Befehl zum Erstellen der Storage-Klasse.

```

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

```

6. Nach Erstellung der Storage-Klasse müssen Sie dann die erste Forderung für ein persistentes Volume (PVC) erstellen. Es gibt ein Beispiel `pvc-basic.yaml` Datei, mit der diese Aktion ausgeführt werden kann, die sich auch in `Sample-Inputs` befindet.

```

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

```

7. Die einzige Bearbeitung, die zu dieser Datei gemacht werden muss, ist sicherzustellen, dass die `storageClassName` Feld entspricht dem gerade erstellten. Die PVC-Definition kann je nach Bedarf des bereitzustellenden Workloads weiter angepasst werden.

```

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

```

8. Erstellen Sie das PVC, indem Sie die ausstellen `oc` Befehl. Die Erstellung kann je nach Größe des erstellten Sicherungsvolumens einige Zeit in Anspruch nehmen, sodass Sie den Prozess nach Abschluss beobachten können.

```
[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
basic	Bound	pvc-3445b5cc-df24-453d-a1e6-b484e874349d	1Gi
		basic-csi	5s

Copyright-Informationen

Copyright © 2024 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFTE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.