



NetApp Astra Trident總覽

NetApp Solutions

NetApp
April 12, 2024

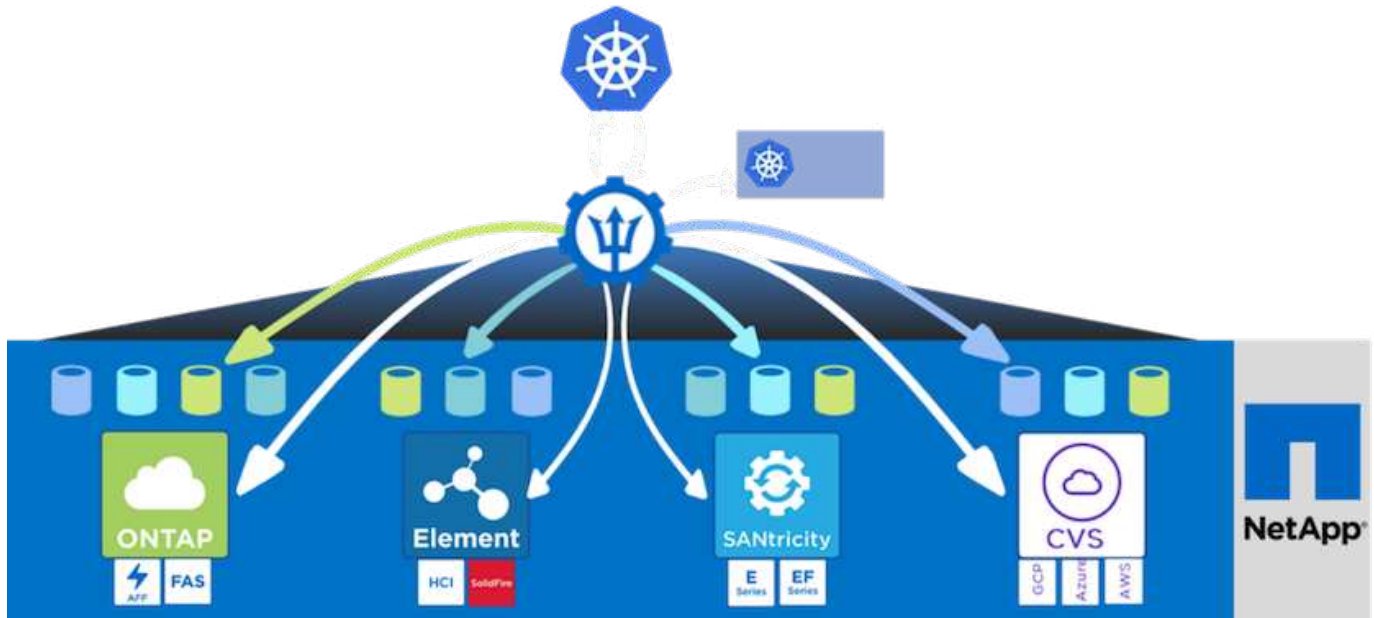
目錄

Astra Trident總覽	1
下載Astra Trident	1
使用Helm安裝Trident運算子	3
手動安裝Trident運算子	4
準備工作節點以供儲存	7
建立儲存系統後端	11
NetApp ONTAP 不適用於NFS組態	12
NetApp ONTAP 支援iSCSI組態	14
支援iSCSI組態NetApp Element	17

Astra Trident總覽

Astra Trident是開放原始碼且完全支援的儲存協調工具、適用於Container和Kubernetes配送、包括Red Hat OpenShift。Trident可搭配整個NetApp儲存產品組合（包括NetApp ONTAP 的整套和Element儲存系統）使用、也支援NFS和iSCSI連線。Trident可讓終端使用者從NetApp儲存系統配置及管理儲存設備、而無需儲存管理員介入、進而加速DevOps工作流程。

系統管理員可根據專案需求和儲存系統模型來設定多個儲存後端、以啟用進階儲存功能、包括壓縮、特定磁碟類型或QoS層級、以保證特定層級的效能。定義後端後端之後、開發人員可在專案中使用這些後端來建立持續磁碟區宣告（PVCS）、並視需要將持續儲存附加至容器。



Astra Trident的開發週期很快、就像Kubernetes一樣、每年發行四次。

最新版的Astra Trident於2022年1月推出22.01版。支援對照表、顯示哪些版本的Trident已通過測試、可找到Kubernetes經銷產品 ["請按這裡"](#)。

從20.04版本開始、Trident設定由Trident操作員執行。營運者讓大規模部署變得更簡單、並提供額外支援、包括在Trident安裝過程中部署的Pod自我修復。

有了21.01版、我們提供了Helm圖表、方便您安裝Trident操作員。

下載Astra Trident

若要在已部署的使用者叢集上安裝Trident並佈建持續磁碟區、請完成下列步驟：

1. 將安裝歸檔檔案下載至管理工作站、並擷取內容。目前的Trident版本為22.01、可下載 ["請按這裡"](#)。

```
[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. 從下載的套裝組合中擷取Trident安裝。

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

使用Helm安裝Trident運算子

1. 首先將使用者叢集的「kubeconfig」檔案位置設定為環境變數、這樣您就不需要參考它、因為Trident沒有傳遞此檔案的選項。

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

2. 在使用者叢集中建立Trident命名空間時、請執行Helm命令、從Lm目錄的tar安裝Trident運算子。

```
[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. 您可以檢查在命名空間中執行的Pod、或使用tridentctl二進位檔檢查安裝的版本、以驗證Trident是否已成功安裝。

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



在某些情況下、客戶環境可能需要自訂Trident部署。在這些情況下、您也可以手動安裝Trident運算子、並更新隨附的資訊清單、以自訂部署。

手動安裝Trident運算子

1. 首先、將使用者叢集的「kubeconfig」檔案位置設定為環境變數、這樣您就不需要參考、因為Trident沒有傳遞此檔案的選項。

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

2. 「Trident安裝程式」目錄包含定義所有必要資源的資訊清單。使用適當的資訊清單、建立「TridentOrchestrator」自訂資源定義。

```
[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. 如果不存在、請使用提供的資訊清單、在叢集中建立Trident命名空間。

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

4. 建立Trident營運者部署所需的資源、例如營運者的「服務帳戶」、專屬的「PodSecurity Policy」、或營運者本身的「ClusterROLTE」和「ClusterROLESTBinding」。

```
[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. 您可以使用下列命令來檢查部署後的操作員狀態：

```
[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. 部署營運者之後、我們就可以使用它來安裝Trident。這需要建立「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. 您可以檢查在命名空間中執行的Pod、或使用tridentctl二進位檔檢查安裝的版本、以驗證Trident是否已成功安裝。

```

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

```

準備工作節點以供儲存

NFS

大多數Kubernetes發佈版本都隨附套件和公用程式、可在預設情況下安裝NFS後端、包括Red Hat OpenShift。

不過、對於NFSv3、用戶端與伺服器之間沒有協調並行的機制。因此、用戶端SUNRPC插槽表項目的最大數量必須以伺服器上支援的值手動同步、以確保NFS連線的最佳效能、而無需伺服器減少連線的視窗大小。

對於支援的SUNRPC插槽表項目數量上限為128、亦即、支援的每次可同時處理128個NFS要求。ONTAP
ONTAP不過、根據預設、每個連線的Red Hat CoreOS/Red Hat Enterprise Linux最多可有65536個SUNRPC插
槽表項目。我們需要將此值設為128、這可透過OpenShift中的機器組態操作員（MCO）來完成。

若要修改OpenShift工作節點中的最大社工PC插槽表格項目、請完成下列步驟：

1. 登入OCP網路主控台、然後瀏覽至「運算」>「機器組態」。按一下「Create Machine Config（建立機器組
複製並貼上Yaml檔案、然後按一下「Create（建立）」。

```
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. 建立MCO之後、必須在所有工作節點上套用組態、然後逐一重新開機。整個程序約需20至30分鐘。使
用「oc Get MCP」確認是否套用機器組態、並確認已更新員工的機器組態集區。

```
[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

若要準備工作節點、以便透過iSCSI傳輸協定對應區塊儲存磁碟區、您必須安裝必要的套件、才能支援該功能。

在Red Hat OpenShift中、這是在叢集部署之後、將MCO（機器組態操作員）套用至叢集來處理。

若要設定工作節點以執行iSCSI服務、請完成下列步驟：

1. 登入OCP網路主控台、然後瀏覽至「運算」>「機器組態」。按一下「Create Machine Config（建立機器組複製並貼上Yaml檔案、然後按一下「Create（建立）」。

不使用多重路徑時：

```
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: ""
```

使用多重路徑時：

```

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,ZGVmYXVsdHMgewogICAgICAgIHVzZXJfZnJpZW5kbHlfbmFtZXNMgbm8KICAgICAgICBmaW5kX211bHRpcGF0aHMgbm8KfQoKYmxhY2tsaXN0X2V4Y2VwdGlvbnMgewogICAgICAgIHByb3BlcnR5ICIoU0NTSV9JREVOVF98SURfV1dOKSfQoKYmxhY2tsaXN0IHsKfQoK
          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. 建立組態之後、將組態套用至工作節點並重新載入大約需要20到30分鐘的時間。使用「ocean Get MCP」確認是否套用機器組態、並確認已更新員工的機器組態集區。您也可以登入工作者節點、確認iscsid服務正在執行（如果使用多重路徑、則多路徑服務正在執行）。

```
[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
```



此外、您也可以使用適當的旗標來執行「occ偵錯」命令、確認機器組態已成功套用、服務已如預期般啟動。

建立儲存系統後端

完成Astra Trident操作員安裝之後、您必須為所使用的特定NetApp儲存平台設定後端。請依照下列連結繼

續Astra Trident的設定與組態。

- ["NetApp ONTAP 不適用於NFS"](#)
- ["NetApp ONTAP 支援iSCSI"](#)
- ["支援iSCSI NetApp Element"](#)

NetApp ONTAP 不適用於NFS組態

若要與NetApp ONTAP 支援儲存系統進行Trident整合、您必須建立後端、以便與儲存系統進行通訊。

1. 下載的安裝歸檔文件中有「shame-INPUT」資料夾階層的範例後端檔案。對於ONTAP NetApp支援NFS的系統、請將「backend-ontap - nas.json」檔案複製到您的工作目錄、然後編輯檔案。

```
[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. 編輯後端名稱、管理LIF、dataLIF、SVM、使用者名稱、和密碼值。

```
{
  "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"
}
```



最佳實務做法是將自訂的backendName值定義為storageDriverName和資料LIF的組合、以利NFS識別。

3. 在這個後端檔案就緒的情況下、執行下列命令來建立第一個後端。

```
[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. 建立後端之後、您必須接著建立儲存類別。就像後端一樣、範例輸入資料夾中也有可供編輯的儲存類別檔案範例。將其複製到工作目錄、並進行必要的編輯、以反映所建立的後端。

```
[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. 唯一必須對此檔案進行的編輯、是從新建立的後端、將「backendType」值定義為儲存驅動程式名稱。另請注意名稱欄位值、此值必須在後續步驟中參考。

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



在此檔案中定義了一個名為「FSType」的選用欄位。此行可在NFS後端刪除。

6. 執行「oc」命令以建立儲存類別。

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

7. 建立儲存類別之後、您必須建立第一個持續磁碟區宣告 (PVC)。還有一個「PVC-base.yaml」檔案範例、也可在範例輸入中執行此動作。

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

8. 唯一必須對此檔案進行的編輯、是確保「儲存類別名稱」欄位符合剛剛建立的欄位。您可以根據要配置的工作負載需求、進一步自訂PVC定義。

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

9. 使用「oc」命令建立PVC。視所建立的備用磁碟區大小而定、建立作業可能需要一些時間、因此您可以在完成時觀看程序。

```
[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 支援iSCSI組態

若要與NetApp ONTAP 支援儲存系統進行Trident整合、您必須建立後端、以便與儲存系統進行通訊。

1. 下載的安裝歸檔文件中有「shame-INPUT」資料夾階層的範例後端檔案。對於ONTAP 供應iSCSI的NetApp 支援系統、請將「backender-ontap - san . json」檔案複製到您的工作目錄、然後編輯該檔案。

```
[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. 編輯此檔案中的管理LIF、dataLIF、SVM、使用者名稱和密碼值。

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

3. 在這個後端檔案就緒的情況下、執行下列命令來建立第一個後端。

```
[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. 建立後端之後、您必須接著建立儲存類別。就像後端一樣、範例輸入資料夾中也有可供編輯的儲存類別檔案範例。將其複製到工作目錄、並進行必要的編輯、以反映所建立的後端。

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

5. 唯一必須對此檔案進行的編輯、是從新建立的後端、將「backendType」值定義為儲存驅動程式名稱。另請注意名稱欄位值、此值必須在後續步驟中參考。

```

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

```



在此檔案中定義了一個名為「FSType」的選用欄位。在iSCSI後端中、此值可設定為特定的Linux檔案系統類型（XFS、ext4等）、也可刪除以允許OpenShift決定要使用的檔案系統。

6. 執行「oc」命令以建立儲存類別。

```

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

```

7. 建立儲存類別之後、您必須建立第一個持續磁碟區宣告（PVC）。還有一個「PVC-base.yaml」檔案範例、也可在範例輸入中執行此動作。

```

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

```

8. 唯一必須對此檔案進行的編輯、是確保「儲存類別名稱」欄位符合剛剛建立的欄位。您可以根據要配置的工作負載需求、進一步自訂PVC定義。

```

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

```

9. 使用「oc」命令建立PVC。視所建立的備用磁碟區大小而定、建立作業可能需要一些時間、因此您可以在完成時觀看程序。

```
[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

支援iSCSI組態NetApp Element

若要啟用Trident與NetApp Element 支援功能的整合、您必須建立後端、以便使用iSCSI傳輸協定與儲存系統進行通訊。

1. 下載的安裝歸檔文件中有「shame-INPUT」資料夾階層的範例後端檔案。若NetApp Element 為供應iSCSI的支援系統、請將「backend-solidfire.json」檔案複製到您的工作目錄、然後編輯檔案。

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

- a. 編輯「端點」行上的使用者、密碼和MVIP值。
- b. 編輯「VIP」值。

```
{
  "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. 在這個後端檔案就位的情況下、執行下列命令來建立您的第一個後端。

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

NAME	STATE	VOLUMES	STORAGE DRIVER	UUID
solidfire_10.61.180.200	online	0	solidfire-san	b90783ee-e0c9-49af-8d26-3ea87ce2efdf

3. 建立後端之後、您必須接著建立儲存類別。就像後端一樣、範例輸入資料夾中也有可供編輯的儲存類別檔案範例。將其複製到工作目錄、並進行必要的編輯、以反映所建立的後端。

```
[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. 唯一必須對此檔案進行的編輯、是從新建立的後端、將「backendType」值定義為儲存驅動程式名稱。另請注意名稱欄位值、此值必須在後續步驟中參考。

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



在此檔案中定義了一個名為「FSType」的選用欄位。在iSCSI後端中、此值可設定為特定的Linux檔案系統類型（XFS、ext4等）、也可刪除此值、讓OpenShift決定要使用的檔案系統。

5. 執行「oc」命令以建立儲存類別。

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

6. 建立儲存類別之後、您必須建立第一個持續磁碟區宣告（PVC）。還有一個「PVC-base.yaml」檔案範例、也可在範例輸入中執行此動作。

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

7. 唯一必須對此檔案進行的編輯、是確保「儲存類別名稱」欄位符合剛剛建立的欄位。您可以根據要配置的工作負載需求、進一步自訂PVC定義。

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

8. 使用「oc」命令建立PVC。視所建立的備用磁碟區大小而定、建立作業可能需要一些時間、因此您可以在完成時觀看程序。

```
[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
RWO		basic-csi	5s

版權資訊

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

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

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

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

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

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

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

商標資訊

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