Install Trident

NetApp Solutions

NetApp
August 18, 2021

Table of Contents

Install Trident

Previous: Peer AKS VNet and Azure NetApp Files VNet.

To install Trident using Helm, complete the following steps:

1. Install Helm (for installation instructions, visit the source).
2. Download and extract the Trident 20.01.1 installer.

```
$wget
$tar -xf trident-installer-21.01.1.tar.gz
```

3. Change the directory to `trident-installer`.

```
$cd trident-installer
```

4. Copy `tridentctl` to a directory in your system `$PATH`.

```
$sudo cp ./tridentctl /usr/local/bin
```

5. Install Trident on the Kubernetes (K8s) cluster with Helm (source):
   a. Change the directory to the `helm` directory.

```
$cd helm
```

   b. Install Trident.

```
$helm install trident trident-operator-21.01.1.tgz --namespace
trident --create-namespace
```

   c. Check the status of Trident pods.

```
$kubectl -n trident get pods
```

   If all the pods are up and running, then Trident is installed and you can move forward.

6. Set up the Azure NetApp Files backend and storage class for AKS.
   a. Create an Azure Service Principle.

   The service principal is how Trident communicates with Azure to manipulate your Azure NetApp Files resources.

```
$az ad sp create-for-rbac --name ""
```

The output should look like the following example:

```
{
"appId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
"displayName": "netapptrident",
"name": "",
"password": "xxxxxxxxxxxxxxx.xxxxxxxxxxxxxx",
"tenant": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx"
}
```

7. Create a Trident backend json file, example name `anf-backend.json`.
8. Using your preferred text editor, complete the following fields inside the `anf-backend.json` file:

```
{
    "version": 1,
    "storageDriverName": "azure-netapp-files",
    "subscriptionID": "fakec765-4774-fake-ae98-a721add4fake",
    "tenantID": "fakef836-edc1-fake-bff9-b2d865eefake",
    "clientID": "fake0f63-bf8e-fake-8076-8de91e57fake",
    "clientSecret": "SECRET",
    "location": "westeurope",
    "serviceLevel": "Standard",
    "virtualNetwork": "anf-vnet",
    "subnet": "default",
    "nfsMountOptions": "vers=3,proto=tcp",
    "limitVolumeSize": "500Gi",
    "defaults": {
    "exportRule": "0.0.0.0/0",
    "size": "200Gi"
}
```

9. Substitute the following fields:
    - `subscriptionID`. Your Azure subscription ID.
    - `tenantID`. Your Azure Tenant ID from the output of `az ad sp` in the previous step.
    - `clientID`. Your appID from the output of `az ad sp` in the previous step.
    - `clientSecret`. Your password from the output of `az ad sp` in the previous step.
10. Instruct Trident to create the Azure NetApp Files backend in the `trident` namespace using `anf-backend.json` as the configuration file:

```
$tridentctl create backend -f anf-backend.json -n trident
```

```
+------------------------+-------------------+--------------------------------------+--------+---------+
|          NAME          |   STORAGE DRIVER  |                 UUID                 | STATE  | VOLUMES |
+------------------------+-------------------+--------------------------------------+--------+---------+
| azurenetappfiles_86181 | azure-netapp-files | 2ca85462-59ac-4946-be05-c03f5575a2ad | online |       0 |
+------------------------+-------------------+--------------------------------------+--------+---------+
```

11. Create a storage class. Kubernetes users provision volumes by using PVCs that specify a storage class by name. Instruct K8s to create a storage class `azurenetappfiles` that references the Trident backend created in the previous step.
12. Create a YAML (`anf-storage-class.yaml`) file for storage class and copy.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
name: azurenetappfiles
provisioner: netapp.io/trident
parameters:
backendType: "azure-netapp-files"
$kubectl create -f anf-storage-class.yaml
```

13. Verify that the storage class was created.

```
kubectl get sc azurenetappfiles
```

```
NAME              PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
azurenetappfiles  csi.trident.netapp.io   Delete          Immediate           false                  98s
```

Next: Set up Dask with RAPIDS deployment on AKS using Helm.