Deploy NetApp Trident (Automated Deployment)

NetApp Solutions

Dorian Henderson, Kevin Hoke
February 19, 2021

Table of Contents

Deploy NetApp Trident (Automated Deployment)

NetApp Trident is deployed by using an Ansible playbook that is available with NVIDIA DeepOps. Follow these steps to set up NetApp Trident:

1. From the Deployment Jump VM, navigate to the DeepOps directory and open a VI editor to `config/group_vars/netapp-trident.yml`. The file from DeepOps lists two backends and two storage classes. In this solution only one backend and storage class are used.

   Use the following template to update the file and its parameters (highlighted in yellow) to match your environment.

```
---
# vars file for netapp-trident playbook
# URL of the Trident installer package that you wish to download and use
trident_version: "20.07.0"# Version of Trident desired
trident_installer_url:
"https://github.com/NetApp/trident/releases/download/v{{ trident_version
}}/trident-installer-{{ trident_version }}.tar.gz"
# Kubernetes version
# Note: Do not include patch version, e.g. provide value of 1.16, not
1.16.7.
# Note: Versions 1.14 and above are supported when deploying Trident
with DeepOps.
#   If you are using an earlier version, you must deploy Trident
manually.
k8s_version: 1.17.9# Version of Kubernetes running
# Denotes whether or not to create new backends after deploying trident
# For more info, refer to: https://netapp-
trident.readthedocs.io/en/stable-v20.04/kubernetes/operator-
install.html#creating-a-trident-backend
create_backends: true
# List of backends to create
# For more info on parameter values, refer to: https://netapp-
trident.readthedocs.io/en/stable-
v20.04/kubernetes/operations/tasks/backends/ontap.html
# Note: Parameters other than those listed below are not avaible when
creating a backend via DeepOps
#   If you wish to use other parameter values, you must create your
backend manually.
backends_to_create:
  - backendName: ontap-flexvol
    storageDriverName: ontap-nas # only 'ontap-nas' and 'ontap-nas-
flexgroup' are supported when creating a backend via DeepOps
    managementLIF: 172.21.232.118# Cluster Management IP or SVM Mgmt LIF
IP
    dataLIF: 172.21.235.119# NFS LIF IP
```

```
    svm: infra-NFS-hci-ai# Name of SVM
    username: admin# Username to connect to the ONTAP cluster
    password:  P@ssw0rd# Password to login
    storagePrefix: trident
    limitAggregateUsage: ""
    limitVolumeSize: ""
    nfsMountOptions: ""
    defaults:
      spaceReserve: none
      snapshotPolicy: none
      snapshotReserve: 0
      splitOnClone: false
      encryption: false
      unixPermissions: 777
      snapshotDir: false
      exportPolicy: default
      securityStyle: unix
      tieringPolicy: none
# Add additional backends as needed
# Denotes whether or not to create new StorageClasses for your NetApp
storage
# For more info, refer to: https://netapp-
trident.readthedocs.io/en/stable-v20.04/kubernetes/operator-
install.html#creating-a-storage-class
create_StorageClasses: true
# List of StorageClasses to create
# Note: Each item in the list should be an actual K8s StorageClass
definition in yaml format
# For more info on StorageClass definitions, refer to https://netapp-
trident.readthedocs.io/en/stable-
v20.04/kubernetes/concepts/objects.html#kubernetes-storageclass-objects.
storageClasses_to_create:
  - apiVersion: storage.k8s.io/v1
    kind: StorageClass
    metadata:
      name: ontap-flexvol
      annotations:
        storageclass.kubernetes.io/is-default-class: "true"
    provisioner: csi.trident.netapp.io
    parameters:
      backendType: "ontap-nas"
# Add additional StorageClasses as needed
# Denotes whether or not to copy tridenctl binary to localhost
copy_tridentctl_to_localhost: true
# Directory that tridentctl will be copied to on localhost
tridentctl_copy_to_directory: ../ # will be copied to 'deepops/'
```

```
directory
```

2. Setup NetApp Trident by using the Ansible playbook.

```
ansible-playbook -l k8s-cluster playbooks/netapp-trident.yml
```

3. Verify that Trident is running.

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

The expected output is as follows:

```
rarvind@deployment-jump:~/deepops$ ./tridentctl -n trident version
+----------------+----------------+
| SERVER VERSION | CLIENT VERSION |
+----------------+----------------+
| 20.07.0        | 20.07.0        |
+----------------+----------------+
```

Next: Deploy NVIDIA Triton Inference Server (Automated Deployment)