vSphere VMFS Datastore - iSCSI Storage backend with ONTAP

NetApp Solutions

Dorian Henderson, Josh Powell, Suresh Thoppay
July 20, 2021

Table of Contents

vSphere VMFS Datastore - iSCSI Storage backend with ONTAP

About this task

This section covers the creation of a VMFS datastore with ONTAP iSCSI storage.

For automated provisioning, use one of these scripts: [PowerShell], Ansible Playbook, or [Terraform].

What you need

- The basic skills necessary to manage a vSphere environment and ONTAP.
- An ONTAP storage system (FAS/AFF/CVO/ONTAP Select/ASA) running ONTAP 9.8 or later
- ONTAP credentials (SVM name, userID, and password)
- ONTAP network port, SVM, and LUN information for iSCSI
- A completed iSCSI configuration worksheet
- vCenter Server credentials
- vSphere host(s) information
  - vSphere 7.0 or later
- iSCSI VMKernel adapter IP informattion
- Network switch(es)
  - With ONTAP system network data ports and connected vSphere hosts
  - VLAN(s) configured for iSCSI
  - (Optional) link aggregation configured for ONTAP network data ports
- ONTAP Tool for VMware vSphere deployed, configured, and ready to consume

Steps

1. Check compatibility with the Interoperability Matrix Tool (IMT).
2. Verify that the iSCSI configuration is supported.
3. Complete the following ONTAP and vSphere tasks.

ONTAP tasks

1. Verify the ONTAP license for iSCSI.
   a. Use the `system license show` command to check if iSCSI is listed.
   b. Use `license add -license-code <license code>` to add the license.
2. Verify that the iSCSI protocol is enabled on the SVM.
3. Verify that iSCSI network logical interfaces are available on the SVM.

When an SVM is created using the GUI, iSCSI network interfaces are also created.

4. Use the `Network interface` command to view or make changes to the network interface.

Two iSCSI network interfaces per node are recommended.

5. Create an iSCSI network interface. You can use the default-data-blocks service policy.
6. Verify that the data-iscsi service is included in the service policy. You can use `network interface service-policy show` to verify.
7. Verify that jumbo frames are enabled.
8. Create and map the LUN. Skip this step if you are using ONTAP tools for VMware vSphere. Repeat this step for each LUN.

VMware vSphere tasks

1. Verify that at least one NIC is available for the iSCSI VLAN. Two NICs are preferred for better performance and fault tolerance.
2. Identify the number of physical NICs available on the vSphere host.
3. Configure the iSCSI initiator. A typical use case is a software iSCSI initiator.
4. Verify that the TCPIP stack for iSCSI is available.
5. Verify that iSCSI portgroups are available.
   - We typically use a single virtual switch with multiple uplink ports.
   - Use 1:1 adapter mapping.
6. Verify that iSCSI VMKernel adapters are enabled to match the number of NICs and that IPs are assigned.
7. Bind the iSCSI software adapter to the iSCSI VMKernel adapter(s).
8. Provision the VMFS datastore with ONTAP Tools. Repeat this step for all datastores.
9. Verify hardware acceleration support.

What's next?

After these the tasks are completed, the VMFS datastore is ready to consume for provisioning virtual machines.

Ansible Playbook

```
## Disclaimer: Sample script for reference purpose only.

- hosts: '{{ vsphere_host }}'
  name: Play for vSphere iSCSI Configuration
  connection: local
  gather_facts: false
  tasks:
    # Generate Session ID for vCenter
    - name: Generate a Session ID for vCenter
```

```
      uri:
        url: "https://{{ vcenter_hostname }}/rest/com/vmware/cis/session"
        validate_certs: false
        method: POST
        user: "{{ vcenter_username }}"
       password: "{{ vcenter_password }}"
        force_basic_auth: yes
        return_content: yes
      register: vclogin

    # Generate Session ID for ONTAP tools with vCenter
    - name: Generate a Session ID for ONTAP tools with vCenter
      uri:
        url: "https://{{ ontap_tools_ip
}}:8143/api/rest/2.0/security/user/login"
        validate_certs: false
        method: POST
        return_content: yes
        body_format: json
        body:
          vcenterUserName: "{{ vcenter_username }}"
          vcenterPassword: "{{ vcenter_password }}"
      register: login

    # Get existing registered ONTAP Cluster info with ONTAP tools
    - name: Get ONTAP Cluster info from ONTAP tools
      uri:
        url: "https://{{ ontap_tools_ip
}}:8143/api/rest/2.0/storage/clusters"
        validate_certs: false
        method: Get
        return_content: yes
        headers:
          vmware-api-session-id: "{{ login.json.vmwareApiSessionId }}"
      register: clusterinfo

    - name: Get ONTAP Cluster ID
      set_fact:
        ontap_cluster_id: "{{ clusterinfo.json |
json_query(clusteridquery) }}"
      vars:
        clusteridquery: "records[?ipAddress == '{{ netapp_hostname }}' &&
type=='Cluster'].id | [0]"

    - name: Get ONTAP SVM ID
      set_fact:
```

```
        ontap_svm_id: "{{ clusterinfo.json | json_query(svmidquery) }}"
      vars:
        svmidquery: "records[?ipAddress == '{{ netapp_hostname }}' &&
type=='SVM' && name == '{{ svm_name }}'].id | [0]"

    - name: Get Aggregate detail
      uri:
        url: "https://{{ ontap_tools_ip
}}:8143/api/rest/2.0/storage/clusters/{{ ontap_svm_id }}/aggregates"
        validate_certs: false
        method: GET
        return_content: yes
        headers:
          vmware-api-session-id: "{{ login.json.vmwareApiSessionId }}"
          cluster-id: "{{ ontap_svm_id }}"
      when: ontap_svm_id != ''
      register: aggrinfo

    - name: Select Aggregate with max free capacity
      set_fact:
        aggr_name: "{{ aggrinfo.json | json_query(aggrquery) }}"
      vars:
        aggrquery: "max_by(records, &freeCapacity).name"

    - name: Convert datastore size in MB
      set_fact:
        datastoreSizeInMB: "{{ iscsi_datastore_size |
human_to_bytes/1024/1024 | int }}"

    - name: Get vSphere Cluster Info
      uri:
        url: "https://{{ vcenter_hostname }}/api/vcenter/cluster?names={{
vsphere_cluster }}"
        validate_certs: false
        method: GET
        return_content: yes
        body_format: json
        headers:
          vmware-api-session-id: "{{ vclogin.json.value }}"
      when: vsphere_cluster != ''
      register: vcenterclusterid

    - name: Create iSCSI VMFS-6 Datastore with ONTAP tools
      uri:
        url: "https://{{ ontap_tools_ip
}}:8143/api/rest/3.0/admin/datastore"
```

```
        validate_certs: false
        method: POST
        return_content: yes
        status_code: [200]
        body_format: json
        body:
          traditionalDatastoreRequest:
            name: "{{ iscsi_datastore_name }}"
            datastoreType: VMFS
            protocol: ISCSI
            spaceReserve: Thin
            clusterID:  "{{ ontap_cluster_id }}"
            svmID: "{{ ontap_svm_id }}"
            targetMoref: ClusterComputeResource:{{
vcenterclusterid.json[0].cluster }}
            datastoreSizeInMB: "{{ datastoreSizeInMB | int }}"
            vmfsFileSystem: VMFS6
            aggrName: "{{ aggr_name }}"
            existingFlexVolName: ""
            volumeStyle: FLEXVOL
            datastoreClusterMoref: ""
        headers:
          vmware-api-session-id: "{{ login.json.vmwareApiSessionId }}"
      when: ontap_cluster_id != '' and ontap_svm_id != '' and aggr_name !=
''
      register: result
      changed_when: result.status == 200
```