



NetApp®

Go further, faster

Developer Guide

Cluster Mode APIs

Amir Mukeri

August 6, 2012

Table of Contents

<u>1 OVERVIEW OF DATA ONTAP 8 CLUSTER-MODE</u>	3
1.1 BUSINESS CHALLENGES WITH TRADITIONAL STORAGE	3
1.2 DATA ONTAP 8.1 CLUSTER-MODE	3
1.3 ARCHITECTURE	4
1.3.1 IMPORTANT COMPONENTS OF CLUSTER-MODE	4
1.3.2 CLUSTER NAMESPACE	5
<u>2 CLUSTER-MODE APIS AND NMSDK</u>	7
2.1 DIFFERENCES BETWEEN CLUSTER AND VSERVER ADMINISTRATORS	7
2.2 MANAGING VSERVICES	7
2.3 UNDERSTANDING VSERVER ADMINISTRATION CAPABILITIES	8
2.4 VSERVER TUNNELING	8
2.5 HOW TO ADMINISTER CLUSTER USING NMSDK	8
2.6 HOW TO ADMINISTER VSERVER USING NMSDK	9
2.7 NEW API CATEGORIES IN CLUSTER-MODE COMPARED TO 7-MODE	9
2.8 MISSING API CATEGORIES IN CLUSTER-MODE COMPARED TO 7-MODE	9
2.9 COMMON API CATEGORIES IN CLUSTER---MODE AND 7---MODE	10
2.10 ITERATIVE APIS	11
2.10.1 EXAMPLE OF ITERATIVE APIS	11
2.11 ZEXPLORE DEVELOPMENT INTERFACE (ZEDI)	13
<u>3 HOW TO DO COMMON 7-MODE TASKS IN CLUSTER-MODE USING APIS</u>	14
3.1 AGGREGATE OPERATIONS	14
3.2 SOME CLUSTER ADMINISTRATOR TASKS	14
3.2.1 VSERVER OPERATIONS	14
3.2.2 NETWORK INTERFACES AND ROUTING GROUPS	15
3.2.3 ROLES AND USERS	15
3.3 SERVICE CONFIGURATION	15
3.3.1 NETWORK INFORMATION SERVICE (NIS)	15
3.3.2 HOST-NAME RESOLUTION	15
3.3.3 LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP)	16
3.4 PROTOCOL CONFIGURATION	16
3.5 VOLUME OPERATIONS	18
3.6 LUN OPERATIONS	18
3.7 QTREES AND QUOTAS	19
3.8 EXPORTS, SHARES AND NAME MAPPINGS	19
3.8.1 EXPORTS	19
3.8.2 SHARES	20
3.8.3 NAME MAPPINGS	20
3.9 SNAPSHOT OPERATIONS	21
3.10 SNAPMIRROR OPERATIONS	22
3.11 STORAGE EFFICIENCY	22
3.12 PERFORMANCE MONITORING	23

1 Overview of Data ONTAP 8 Cluster-Mode

1.1 Business challenges with traditional storage

- **Capacity Scaling:** Capacity expansion in traditional storage systems may require downtime, either during physical installation or when redistributing existing data across the newly installed capacity.
- **Performance scaling:** Standalone storage systems may lack the I/O throughput to meet the needs of large-scale enterprise applications.
- **Availability:** Traditional storage systems often have single points of failure that can affect data availability.
- **Right-sized SLAs:** Not all enterprise data requires the same level of service (performance, resiliency, and so on). Traditional storage systems support a single class of service, often resulting in poor utilization or unnecessary expense.
- **Cost:** With rapid data growth, storage is consuming a larger and larger portion of shrinking IT budgets.
- **Complicated management:** Discrete storage systems and their subsystems must be managed independently. Existing resource virtualization does not extend far enough in scope.

1.2 Data ONTAP 8.1 Cluster-Mode

NetApp Data ONTAP 8.1 Cluster-Mode helps to achieve results and get to market faster by providing the massive throughput and scalability you need to meet the demanding requirements of your high performance computing and digital media content applications. It also allows achieving high levels of performance, manageability, and reliability for your large Linux, UNIX, or Microsoft Windows clusters with Data ONTAP Cluster-Mode.

The Data ONTAP Cluster-Mode operating system provides:

- Scale-up, scale-out and Scale-down is possible with multiple nodes using a global namespace
- Storage virtualization with Vservers eliminates the physical boundary of a single controller.
- Immortality as it provides non-disruptive operations when you re-distribute load or capacity rebalance and network load balancing with-in the cluster or upgrade, expand the nodes in a cluster.
- All the storage efficiency like snapshots, thin provisioning, space efficient cloning, deduplication, data compression and RAID-DP are also available.

The solution for the above mentioned business challenges could be taken care of by the scale-out cluster-mode approach as follows:

- Grow capacity incrementally, on demand, through the nondisruptive addition of storage shelves and growth of storage containers (pools, LUNs, file systems). Support nondisruptive redistribution of existing data to the newly provisioned capacity as needed.
- Grow performance incrementally, on demand, through the addition of storage controllers in small, economical (pay-as-you-grow) units without disrupting applications.

- Leverage clustered controller configurations with appropriate failover support to ensure continuous data availability in the face of individual component faults.
- Support different levels of service and provide the ability to dynamically modify the service characteristics associated with stored data, by non-disruptively migrating data to slower, less costly disk, and/or by applying quality of service criteria.
- Control costs through the use of scale-out architectures that employ commodity components. Grow capacity and performance on an as-needed (pay-as-you-go) basis. Increase utilization through thin provisioning and data deduplication.
- Provide a single point of management across the aggregate set of components. Leverage policy-based management to streamline configuration, provisioning, replication, and backup. Provide a flexible monitoring and reporting structure implementing an exception based management model. Virtualize resources across multiple controllers so that volumes become simple-to manage logical entities that span storage controllers for performance and dynamic redistribution of data.

1.3 Architecture

1.3.1 Important components of Cluster-Mode

1.3.1.1 Virtual Server (Vserver)

A Vserver is a logical file system namespace capable of spanning beyond the boundaries of physical nodes in a cluster.

- Clients can access virtual servers from any node in the cluster, but only through the associated logical interfaces (LIFs)
- Each Vserver has a root volume under which additional volumes are mounted extending the namespace
- It can span multiple physical nodes
- It is associated with one or more logical interfaces; clients access the data on the virtual server through the logical interfaces

1.3.1.2 Logical Interface (LIF)

A logical interface is essentially one or more than one IP address with associated characteristics, such as a home port, a list of ports to failover to, a firewall policy, a routing group, and so on.

- Client network access is through logical interfaces (LIFs) dedicated to the Vserver
- A Vserver can have more than one LIF. You can have many clients mounting one LIF or one client mounting multiple LIFs
- In case a volume is moved from one node to another within a single namespace across nodes, it is recommended to migrate the LIF to the new location of the volume to provide more data locality and for load balancing
- This means IP addresses are no longer tied to a single physical interface

1.3.1.3 Aggregate

An aggregate is a RAID-level collection of disks; it could contain more than one RAID group.

- Serve as resources for Vservers, and are shared by all Vservers.

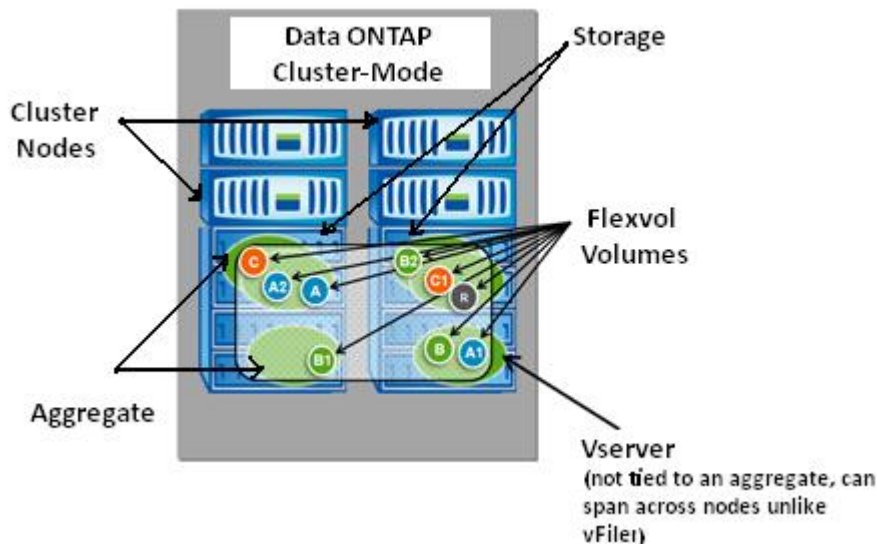
Note: Starting with Data ONTAP 8.1, newly created aggregates are 64-bit by default, and new systems are shipped with the root volume in a 64-bit aggregate.

1.3.1.4 FlexVol volumes

A volume is a logical unit of storage. The disk space that a volume occupies is provided by an aggregate.

- Each volume is associated (though not permanently) with one individual aggregate and as such one physical node
- In cluster mode the volumes are part of a Vserver and not tied an aggregate
- Can be moved from aggregate to aggregate with the Data Motion for volumes feature, without loss of access to the client. This means the volumes are no longer tied to aggregates as in the traditional 7g/7-mode implementation but are part of Vservers. This provides more flexibility to move the volumes within a single namespace to address issues like capacity management and load balancing.

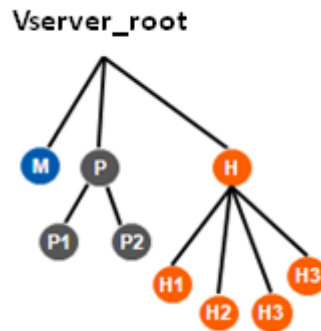
Note: Starting in Data ONTAP 8.0.1, a 64-bit volume can be used as the root volume



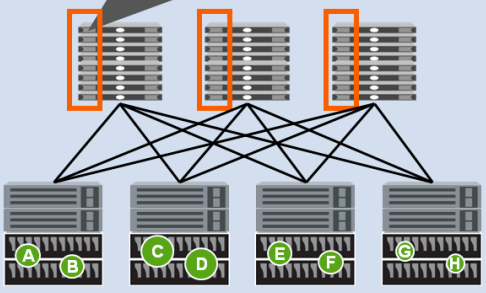
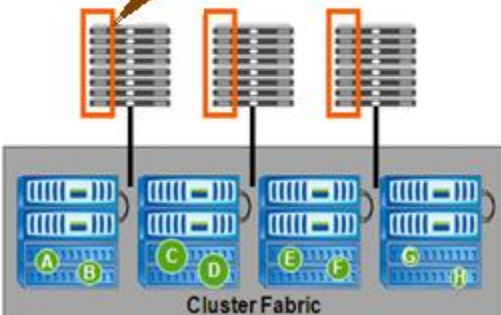
1.3.2 Cluster Namespace

A Cluster Namespace is a collection of filesystems hosted from different nodes in the cluster. Each Vserver has a file namespace that consist of a single root volume. The Vserver namespace consists of one or more volumes, linked via junctions that connect from a named junction inode in one volume to the root directory of another volume. A cluster can have more than one Vserver.

All the volumes belonging to the Vserver are linked into the global namespace in that cluster. The cluster namespace is mounted at a single point in the cluster. The top directory of the cluster namespace within a cluster is a synthetic directory, containing entries for the root directory of each Vserver namespace in the cluster.



1.3.2.1 Benefits of a Cluster Namespace

Without a Cluster Namespace	With a Cluster Namespace
<p>Many mount points per client:</p> <pre> /mount/box1/volA /mount/box2/volB ... /mount/box8/volH </pre> 	<p>Single mount point per client: /mount/vserver_root</p> 
<ul style="list-style-type: none"> • Change mapping for thousands of clients when moving or adding data. 	<ul style="list-style-type: none"> • Namespace unchanged as data moves.
<ul style="list-style-type: none"> • Difficult to manage 	<ul style="list-style-type: none"> • Much easier to manage
<ul style="list-style-type: none"> • Very complex to change 	<ul style="list-style-type: none"> • Much easier to change
<ul style="list-style-type: none"> • Doesn't scale 	<ul style="list-style-type: none"> • Seamlessly scales to petabytes

2 Cluster-Mode APIs and NMSDK

Due to the new cluster-mode infrastructure described above, APIs for cluster mode Data ONTAP 8.1 (and onwards) are divided into two sets:

- **Cluster:** to administer the entire cluster and its resources
- **Vserver:** to administer only their own cluster Vservers' storage and network resources

2.1 Differences between Cluster and Vserver administrators

Cluster administrators administer the entire cluster. Vserver administrators administer only their own cluster Vservers.

Cluster administrators can administer the entire cluster and its resources. They can also set up cluster Vservers and delegate Vserver administration to Vserver administrators. The specific capabilities that cluster administrators have depend on their access-control roles. By default, a cluster administrator with the "admin" account name or role name has all capabilities for managing the cluster and Vservers.

Vserver administrators, on the other hand, can administer only their own cluster Vservers' storage and network resources, such as volumes, protocols, LIFs, and services. The specific capabilities that Vserver administrators have depend on the access-control roles that are assigned by cluster administrators. For more information about Vserver administrator capabilities, see the Data ONTAP Cluster-Mode Vserver Administrator Capabilities Overview Guide (on [NetApp support site](#)).

2.2 Managing Vservers

Cluster administrators can manage and administer the Vservers within a cluster. However, the cluster administrator can delegate the administration of the Vservers to Vserver administrators. To manage and administer Vservers, you must understand what a Vserver is, its benefits, and the associated management tasks.

Only a cluster administrator can perform the following Vserver management tasks:

- Creating Vservers
 - There are two methods for creating Vservers:
 - The vserver setup command launches an easy-to-use wizard that allows you to quickly create a functional Vserver.
 - The vserver create command enables you to create a Vserver with the root volume and basic configuration, such as name service switch, name mapping switch, and root volume security style.
- Modifying Vservers
- Deleting Vservers
- Renaming Vservers
- Administering Vservers from the Vserver context
- Starting and stopping Vservers

Note: Both cluster administrators and Vserver administrators can view information about Vservers.

2.3 Understanding Vserver administration capabilities

A Vserver administrator can administer a Vserver and its resources, such as volumes, protocols, and services, depending on the capabilities assigned by the cluster administrator. A Vserver administrator cannot create, modify, or delete a Vserver. To administer a Vserver efficiently, you must understand what a Vserver is and its benefits.

A Vserver is a secure virtual storage server, which can have its own user and authentication domain. A Vserver provides benefits, such as nondisruptive operation, scalability, and unified storage.

You might have all or some of the following administration capabilities:

- **Data access protocol configuration:** You can configure data access protocols, such as NFS, CIFS, iSCSI, and FCP.
- **Services configuration:** You can configure services, such as LDAP, NIS, and DNS.
- **Storage management:** You can manage volumes, quotas, qtrees, and files.
- **LUN management:** You can manage LUNs in a SAN environment.
- **Snapshot copy management:** You can manage Snapshot copies of the volumes.
- **Monitoring Vserver:** You can monitor jobs, network connection, network interface, and Vserver health.

For troubleshooting or modifying Vserver configurations, you must contact the cluster administrator.

2.4 Vserver tunneling

The mechanism of accessing Vserver APIs through a cluster-management interface is called Vserver tunneling.

Data ONTAP responds to a tunneled API based on the tunnel destination, target interface and the API family, as described in the following cases:

- Data ONTAP Vserver APIs can be executed if they are sent through a cluster-management LIF to an Admin Vserver.
- Data ONTAP Vserver APIs can be executed if they are sent through a Vserver-management LIF to a Data Vserver.
- Data ONTAP Vserver APIs can be executed if they are sent through a cluster-management LIF to a Data Vserver.
- Data ONTAP C-Mode APIs can be executed if they are sent through a cluster-management LIF to a Node Vserver

2.5 How to administer Cluster using NMSDK

Cluster administrator can use all the APIs listed under Data ONTAP cluster mode APIs by connecting to cluster management IP of the cluster.

2.6 How to administer Vserver using NMSDK

Cluster administrator can administer Vservers from the Vserver context by using Vserver tunneling in NMSDK.

After setting up a functional Vserver or a Vserver with basic network configuration, you can optionally delegate the administration of the Vserver to a Vserver administrator. You can delegate Vserver administration by creating and assigning user accounts either with predefined roles or customized roles.

Vserver administrator can directly connect to Vserver IP using the appropriate username and password.

2.7 New API categories in Cluster-Mode compared to 7-Mode

- antivirus
- audit
- autosupport
- cluster
- cluster-peer
- config-backup
- dashboard
- exports (under nfs in 7-mode)
- flexcache
- gpo
- job
- kerberos
- ldap
- name-mapping (partially available in nameservice in 7-mode)
- qos
- security
- vserver

2.8 Missing API categories in Cluster-Mode compared to 7-Mode

- consistency
- copyoffload
- dfm
- flash
- fpolicy
- ic
- ipspace
- license
- lock
- nameservice (partially available in name-mapping)
- priority

- radius
- reallocate
- snaplock
- snapvault
- software
- useradmin
- vfiler
- vmservices

2.9 Common API categories in Cluster-- - Mode and 7-- - Mode

- aggr
- cf
- cifs
- clock
- clone
- diagnosis
- disk
- ems
- fc
- fcp
- fcport
- file
- igroup
- iscsi
- lun
- net
- nis
- options
- perf
- portset
- qtree
- quota
- rsh
- ses
- sis
- snapmirror
- snapshot
- snmp
- storage---adapter
- storage---array
- storage---disk
- storage---initiator
- system
- volume

- waf1

2.10 Iterative APIs

Iterative APIs allow you to operate on a large set of objects by dividing that set of objects into manageable sub-sets, repeatedly executing an API call.

To repeatedly call a Data ONTAP 7-Mode API, you can use `*-iter-start`, `*-iter-next`, and `*-iter-end`. In Cluster-Mode, you can use a single API `*-get-iter`.

`*-get-iter` API supports queries, which can be used to return all objects that match the query. The objects are sorted alphabetically (default) in ascending order of the keys. It can be sorted numerically too.

Following are the input elements of an iterative API:

- `max-records`—determines the maximum objects that can be sent as response, in a one API call.

Each iterative API, for an API category, has a set upper limit and a default value. This element is supported for all APIs that support queries.

- `tag`— specifies a value used by the server to identify the next set of records to return.

If `tag` value is not specified, the server starts from the first record of the query result.

- `query`—specifies the query and must be specified for each iteration of an API.

You can use wildcards and range in a query.

- `desired-attributes`—specifies the number of elements that can be returned for each record.

This is an optional value. If it is not specified, elements, which contain values, are returned.

Following are the output elements of an iterative API:

- `next-tag`—specifies the tag value passed to the tag input element in the next call to the same API.
- `num-records`—specifies the total number of records returned for the API call.
- `attributes-list`—specifies the actual list of records in the `*-get-iter` API call.

2.10.1 Example of Iterative APIs

This is an example of iterator-style API (`job-get-iter`), along with request and response packets obtained over 3 iterations. Please note that response objects are abbreviated (not all elements are shown).

Iteration 1:

Request:

```
<job-get-iter>
  <max-records>3</max-records>
</job-get-iter>
```

Response:

```
<results status="passed">
  <attributes-list>
    <job-info>
```

```

    <job-id>1</job-id>
    <job-name>Certificate Expiry Check</job-name>
    <job-priority>low</job-priority>
    <job-type>Security Certificate</job-type>
    <job-uuid>c983750f-9eec-11df-8add-123478563412</job-uuid>
  </job-info>
  <job-info>
    <job-id>2</job-id>
    <job-name>CLUSTER BACKUP AUTO 8hour</job-name>
    <job-priority>medium</job-priority>
    <job-type>CLUSTER BACKUP</job-type>
    <job-uuid>11118790-9eed-11df-8add-123478563412</job-uuid>
  </job-info>
  <job-info>
    <job-id>3</job-id>
    <job-name>CLUSTER BACKUP AUTO daily</job-name>
    <job-priority>medium</job-priority>
    <job-type>CLUSTER BACKUP</job-type>
    <job-uuid>119c8993-9eed-11df-8add-123478563412</job-uuid>
  </job-info>
</attributes-list>
<next-tag>&lt;job-get-iter-key-td&gt;
&lt;key-0&gt;3&lt;/key-0&gt;
&lt;/job-get-iter-key-td&gt;
</next-tag>
  <num-records>3</num-records>
</results>

```

Note: No tag is provided in the request for the first iteration. The response has the next-tag element, which is required in the next iteration. Also, note that the number of records (num-records) returned is equal to the maximum number of records (max-records) requested for. This indicates that there are more records present in the list.

Iteration 2:

Request:

```

  <job-get-iter>
    <max-records>2</max-records>
    <tag>
      <job-get-iter-key-td>
        <key-0>3</key-0>
      </job-get-iter-key-td>
    </tag>
  </job-get-iter>

```

Response:

```

<results status="passed">
  <attributes-list>
    <job-info>
      <job-id>4</job-id>
      <job-name>CLUSTER BACKUP AUTO weekly</job-name>
      <job-priority>medium</job-priority>
      <job-type>CLUSTER BACKUP</job-type>
      <job-uuid>119db730-9eed-11df-8add-123478563412</job-uuid>
    </job-info>
    <job-info>
      <job-id>5</job-id>
      <job-name>Vol Reaper</job-name>

```

```

        <job-priority>high</job-priority>
        <job-type>Vol Reaper</job-type>
        <job-uuid>79129213-9eed-11df-8add-123478563412</job-uuid>
    </job-info>
</attributes-list>
<next-tag>&lt;job-get-iter-key-td&gt;
    &lt;key-0>5</key-0&gt;
&lt;/job-get-iter-key-td&gt;
</next-tag>
    <num-records>2</num-records>
</results>

```

Note: The next-tag obtained in Iteration 1 is passed as the tag for Iteration 2.

Iteration 3:

Request:

```

<job-get-iter>
    <max-records>2</max-records>
    <tag>
        <job-get-iter-key-td>
            <key-0>5</key-0>
        </job-get-iter-key-td>
    </tag>
</job-get-iter>

```

Response:

```

<results status="passed">
    <attributes-list>
        <job-info>
            <job-id>6</job-id>
            <job-name>Licensing</job-name>
            <job-priority>low</job-priority>
            <job-type>Cluster Licenses</job-type>
            <job-uuid>794aab08-9eed-11df-8add-123478563412</job-uuid>
        </job-info>
    </attributes-list>
    <num-records>1</num-records>
</results>

```

Note: The response does not contain any next-tag element, which confirms that no more iteration is required (as no more records are left in the list). This is also indicated by num-records element, as it is less than max-records.

2.11 ZExplore Development Interface (ZEDI)

ZEDI is a utility with graphical user interface bundled with NetApp Manageability SDK (NMSDK). This utility enables you to test DATA ONTAP APIs and Data Fabric Manager server APIs from a dialog box. This utility allows you to generate raw XML request for any given API. You can supply necessary arguments in the XML request before invoking the API through HTTP or HTTPS and you can view the response in raw XML format or tree format. Additionally, for a given API, the utility can generate sample codes in Java, Perl, C, C#, Python and Ruby to demonstrate how the said API can be invoked using NMSDK Core APIs. You can choose to include comments (API documentation descriptions) and/or optional parameters while generating the XML request and sample codes. You can generate Workflows by sequencing multiple APIs in logical order. It also supports vFiler and Vserver tunneling, which enables a DATA ONTAP API to be invoked directly on a vFiler or a Vserver respectively. For more information about ZEDI, see the ZEDI User Guide or visit [ZEDI developer community page](#).

3 How to do common 7-Mode tasks in Cluster-Mode using APIs

Some common differences to keep in mind while using the cluster-mode APIs:

1. As mentioned [above](#), there are two sets of APIs in cluster-mode. You can look at the API documentation accordingly and select the right API for your task.
2. Data is served by Vservers and you need to create and configure it before you can provision storage.
3. Iterative APIs are new in cluster-mode, so take a look at the example [above](#) for better understanding of these APIs.
4. There are direct APIs in cluster-mode for which there are *options* in 7-Mode. No need to use *options-set* API to modify these options. You can look at specific examples below as we discuss the API categories in more detail.
5. In Cluster-mode, to create schedules (for tasks like snapshot, snapmirror etc), use the *job-schedule-cron-create* or *job-schedule-interval-create* APIs (you can look at *job* category of Cluster-Mode API documentation for further details). You can later attach these schedules with respective policies.

3.1 Aggregate operations

Aggregates serve as resources for all Vservers. An aggregate may span multiple nodes in cluster mode environment.

How is it different from 7-Mode?

No change in basic aggregate operations such as create, destroy, rename, online, offline etc. This is a cluster administrator task, so the APIs for these operations can be found under *aggr* category of cluster mode APIs.

3.2 Some Cluster Administrator tasks

Unlike 7-Mode, you need to do following operations before provisioning further storage (i.e. before volume, LUN, or qtree creation).

3.2.1 Vserver operations

- Vserver is a new concept in Cluster Mode as described in [Section 2.3.1.1](#)
- You need to create a Vserver to provision storage in cluster mode. Detailed capabilities of Cluster and Vserver are described in [Sections 3.1, 3.2](#) and [3.3](#).
- Create a Vserver using ***vserver-create*** API (Since this is a cluster administrator task, you can look at *vserver* category of Cluster-Mode API documentation for further details).
 - To create a Vserver you should create at least one non-root aggregate first.

3.2.2 Network interfaces and routing groups

Create a data LIF on Vserver using ***net-interface-create*** API so that clients can access data through this LIF.

- You can specify list of data protocols to be configured on the LIF.

Each LIF is associated with one routing group and uses only the routes of that group. Multiple LIFs can share a routing group.

- Create a new networking routing group route using ***net-routing-group-route-create*** API.

Where to look for APIs? Since both of above tasks are cluster administrator task, you can look at *net* category of Cluster-Mode API documentation for further details.

3.2.3 Roles and Users

A cluster administrator creates user roles and accounts. For example, cluster administrator can create Vserver administrator roles and user accounts and can delegate Vserver level tasks to those administrators.

- You can create new user roles using ***security-login-role-create*** API
- Create a new user account associated the specified application and authentication method using ***security-login-create*** API.

Where to look for APIs? You can look at *security* category of Cluster-Mode API documentation for further details.

3.3 Service configuration

3.3.1 Network Information Service (NIS)

If you specified NIS as a name service option during Vserver create, you must create a NIS domain configuration for the Vserver. You can use the ***nis-create*** API to create a NIS domain configuration.

You can create multiple NIS domains. However, you can only use one that is set to active.

Where to look for APIs? Since this is a Vserver level task, you can look at *nis* category of Vserver API documentation for further details

3.3.2 Host-name resolution

Cluster-Mode supports two methods for host-name resolution: DNS and hosts table. Cluster administrators can configure Domain Name Service (DNS) and hosts file naming services for host-name lookup in the cluster Vserver.

3.3.2.1 Hosts table

You can use the ***net-hosts-create*** API for configuring the hosts file that resides in the root volume of the admin Vserver. By default, the order of lookup for the admin Vserver is hosts file first and then DNS.

3.3.2.2 Domain Name Service (DNS)

If you want to configure DNS later, you should use the ***net-dns-create*** API.

Where to look for APIs? Since these are Vserver level tasks, you can look at *net* category of Vserver API documentation for further details

3.3.3 Lightweight Directory Access Protocol (LDAP)

An LDAP server enables you to centrally maintain user information.

If you store your user database on an LDAP server, you can configure your Vserver to look up user information in the LDAP database. For example, on your LDAP server, you can store logins and passwords for administrative users of the console and the rsh, telnet, http, https, and ssh protocols, making it possible for you to centrally manage them.

1. You can use the ***ldap-client-create*** API to create an LDAP client configuration (cluster administrator task). You must set up an LDAP client first to be able to use LDAP services.
2. To associate an LDAP client configuration with a Vserver, you must create an LDAP configuration and sets its *-client-config* parameter to the name of the LDAP client. You can use the ***ldap-config-create*** API to configure a Vserver to use an LDAP client.

Where to look for APIs? You can look at *ldap* category of Cluster-Mode and Vserver API documentation for further details.

How are these different from 7-Mode?

- Similar configuration can be done in 7-Mode as well but by modifying options using the ***options-set*** API. For example, you need to modify different *nis.**, *dns.**, *ldap.** options to configure these services.

3.4 Protocol Configuration

You can configure a Vserver with any combination of supported data access protocols, which are NFS, CIFS, iSCSI, and FCP, to serve data. Depending on your capabilities, you can manage data access protocols on a Vserver to control the data access by the clients.

- You can configure an NFS server on a Vserver using ***nfs-service-create*** API to provide data access to its NFS clients.
- You can enable and configure a CIFS server on a Vserver using ***cifs-server-create*** API to let CIFS clients access files on your storage system.
- You can configure FC protocol on a Vserver using ***fcp-service-create*** API to export LUNs and transfer block data to the FC initiator hosts.

- You can configure iSCSI protocol on a Vserver using ***iscsi-service-create*** API to export LUNs and transfer block data to the iSCSI initiator hosts.

Note: You can configure and manage only the protocols that are allowed on the Vserver by the cluster administrator.

These are Vserver level tasks, so you can look at *nfs*, *cifs*, *fcp* and *iscsi* categories of Vserver API documentation for further details. For more details on how to execute these APIs look at [Section 3.6](#).

How it is different from 7-Mode?

	7-Mode	Cluster-Mode
NFS	<ol style="list-style-type: none"> Use <i>nfs-enable</i> API to enable NFS on a storage system. Modify nfs options using <i>options-set</i> API to enable/disable different versions of NFS or to use features like ACLs and delegations. 	<ol style="list-style-type: none"> You need to first configure a NFS server on a Vserver using <i>nfs-service-create</i> API or <i>nfs-enable</i> API will create a NFS server with default options. You can enable different NFS versions and features you want to use while creating a NFS server or you can modify these options using <i>nfs-service-modify</i> API and NOT options-set API like in 7-Mode.
CIFS	Use <i>cifs-setup</i> API to configure CIFS services	Use <i>cifs-server-create</i> API to configure CIFS services on a Vserver
FCP	Use <i>fcp-service-start</i> API to start the FCP services.	<p>Since certain tasks can be done only by cluster administrator and not by Vserver administrator, FCP APIs are divided into Cluster and Vserver APIs:</p> <ul style="list-style-type: none"> Fiber Channel target adapter configuration APIs are at cluster level Other 7-Mode like APIs are at Vserver level <p>As in the case of other protocols, you need to first create FCP service at Vserver level using <i>fcp-service-create</i> API. You can start the FCP service at the time of service creation or later using <i>fcp-service-start</i> API.</p>
iSCSI	Use <i>iscsi-service-start</i> API to start the iSCSI services.	As in the case of other protocols, you need to first create iSCSI service at Vserver level using <i>iscsi-service-create</i> API. You can start the iSCSI service at the time of service creation or later using <i>iscsi-service-start</i> API.

3.5 Volume operations

A Vserver represents the logical layer of data storage. Depending on your capabilities, you can perform the following tasks to manage volumes on a Vserver:

- Creating, modifying, renaming, or deleting volumes (the number of volumes you can create on the Vserver is defined by the cluster administrator).
Note: It is best not to store user data in the root volume of a Vserver.
- Mounting or unmounting volumes
- Removing junctions from volumes
- Viewing volume status

Note: Vserver administrator cannot copy or move volumes between aggregates. Cluster administrator can perform these tasks.

How is it different from 7-Mode?

1. In Cluster-Mode Vserver serves data, so you need to create a Vserver (see [Vserver operations](#) for more details) first to create volumes.
2. You need to configure data LIF on a Vserver, using which you can mount volumes from client.
3. In Cluster-Mode we need to specify **junction-path** so that it can be mounted from a client using this path. You can provide junction-path while creating a volume or later using **volume-modify-iter** API (look at API documentation for more details).
4. In Data ONTAP 7-Mode, you can create a new flexible, traditional, or sparse volume whereas only flexible volume in Cluster-Mode.
5. Cluster administrator can transition a 7-Mode flexible volume to a cluster-mode volume using **volume-transition*** APIs.

3.6 LUN operations

In a SAN environment, you can provision storage by creating LUNs, igroups, and mapping the LUNs to the igroups. After you create LUNs, you can manage their availability, mapping, and accessibility. Depending on your capabilities, you can perform the following tasks to manage LUNs:

- Creating, modifying, renaming, or deleting LUNs
- Modifying LUN size
- Managing igroups and port sets
- Mapping LUNs to the initiators
- Unmapping LUNs
- Viewing list of LUNs

Where to look for APIs? Since this is a Vserver level task, you can look at *lun* category of Vserver API documentation for further details

How is it different from 7-Mode?

- No change in basic operations and APIs.

3.7 qtrees and quotas

You can manage volume qtrees and volume quotas by performing the following tasks:

- Creating, modifying, renaming, or deleting qtrees
- Viewing qtree status and statistics
- Creating, modifying, renaming, or deleting quota policy and policy rules
- Viewing quota policy and policy rules

Where to look for APIs? Since this is a Vserver level task, you can look at *qtree* and *quota* categories of Vserver API documentation for further details

How is it different from 7-Mode?

- The same export permissions will be applied to all the qtrees/sub-directories contained within that volume. Unlike 7-Mode, Cluster-Mode doesn't support sub-volume exports.

3.8 Exports, Shares and Name mappings

3.8.1 Exports

Each volume is associated with exactly one export policy. Each export policy is identified by a unique name and a unique numeric ID. A Data ONTAP, cluster can contain up to 1,024 export policies. Each Vserver has at least one export policy called default, which contains no rules. This export policy cannot be deleted, although it can be renamed or modified. Each volume on a Vserver by default is associated with the default export policy.

Export policies consist of individual export rules. An export policy can contain a large number of rules (approximately 4,000). Each rule specifies access permissions to volumes for one or more clients. The clients can be specified by hostname, IP address, or netgroup.

Steps to create export policies to restrict access to volumes to specific clients:

1. Create an export policy using *export-policy-create* API.
2. Add a rule to the export policy using *export-rule-create* API.
3. You can provide the export policy as one of the optional input arguments to *volume-create* API or you can later use *volume-modify-iter* API to specify the snapshot policy.

Where to look for APIs? Since this is a Vserver level task, you can look at *exports* category of Vserver API documentation for further details

How is it different from 7-Mode?

- In Data ONTAP 7-Mode, pathname must be directory name or file to export whereas in Data ONTAP Cluster-Mode, it must be a junction path of the volume to be exported. The same export permissions will be applied to all the qtrees/sub-directories contained within that volume. Unlike 7-Mode, Cluster-Mode doesn't support sub-volume exports.

- Even though the same 7-Mode *nfs-exports*-* APIs are present in cluster-mode, we recommend using *exports* category APIs as described above.
- In Data ONTAP Cluster-Mode, the export entries are always persistent.

3.8.2 Shares

1. If you want to use the CIFS home directory feature, you must add at least one home directory search path. You can add a home directory search path by using the Vserver *cifs-home-directory-search-path-add* API.
2. You must first create a CIFS share before you can share data on a Vserver with CIFS clients. You can use the Vserver *cifs-share-create* API to create a CIFS share.
3. You can use the Vserver *cifs-share-access-control-create* API to create an access control list for a CIFS share. This enables you to control the level of access to a share for users and groups.

Where to look for APIs? Since this is a Vserver level task, you can look at *cifs* category of Vserver API documentation for further details

How is it different from 7-Mode?

- No change in basic operations and but API names have changed.

7-Mode APIs	Cluster-Mode APIs
cifs-share-add	cifs-share-create
cifs-share-change	cifs-share-modify
cifs-share-ace-*	cifs-share-access-control-*
cifs-homedir-*	cifs-home-directory-*

3.8.3 Name mappings

Data ONTAP uses name mapping to map CIFS identities to UNIX identities, Kerberos identities to UNIX identities, and UNIX identities to CIFS identities. It needs this information to obtain user credentials and provide proper file access regardless of whether they are connecting from an NFS client or a CIFS client.

Name mapping is usually required due to the multi-protocol nature as Data ONTAP supports CIFS and NFS access to the same files, as well as NTFS and UNIX security styles on volumes.

- You can use the Vserver *name-mapping-create* API to create a name mapping. Data ONTAP supports up to 1024 name mappings for each direction.
- You can configure a default user to use if all other mapping attempts fail for a user, or if you do not want to map individual users between UNIX and Windows. Alternatively, if you want authentication of non-mapped users to fail, you should not configure a default user.
 - For CIFS, you can configure the default unix user by using *cifs-options-modify* API and modifying the default-unix-user argument

- For NFS, you can configure the default windows user by using *nfs-service-modify* API and modifying the default-windows-user argument
- You can set up UNIX user accounts on a Vserver (using *name-mapping-unix-user-create* API) to provide an authentication mechanism for NFS access.
- You can also set up local UNIX groups on a Vserver (using *name-mapping-unix-group-create* API) along with local UNIX users.
- You can use the *name-mapping-unix-group-add-user* API to add a user to a UNIX group that is local to a Vserver.

Where to look for APIs? Since this is a Vserver level task, you can look at *name-mapping* category of Vserver API documentation for further details

How is it different from 7-Mode?

- In Data ONTAP 7-mode:
 - You can specify how each Windows name maps to a UNIX name by entering information in the */etc/usermap.cfg* file. If you accept the default mapping, you do not need to enter this information. There is **no API** to perform this task.
 - For a CIFS user to have a UID and GIDs, you must create a UNIX account in the UNIX password database that corresponds to the user's UNIX name. Add an entry in the */etc/passwd* file for the UNIX name of each user. There is **no API** to perform this task.
 - For enabling/disabling default user, you need to modify *wafld.default_unix_user* or *wafld.default_nt_user* options.
 - There are some *nameservice* APIs to get mapping of uid, gid, and usernames.

3.9 Snapshot operations

A **Vserver** uses Snapshot technology to back up its data volumes. The Snapshot copies of the volumes reside within the Vserver. You can manage the Snapshot copies, and restore files from the Snapshot copies if data is corrupted. Depending on your capabilities, you can perform the following tasks to manage Snapshot copies of volumes of a Vserver:

- Creating, modifying, renaming, or deleting Snapshot copies
- Viewing Snapshot policies
- Computing reclaimable space for Snapshot copies
- Viewing the list of Snapshot copies
- Restoring files from Snapshot copies

Some Snapshot copy tasks are for the **cluster administrator** to perform and cannot be performed by the Vserver administrator.

- Creating, modifying, or deleting a snapshot policy
- Managing schedules within a snapshot policy

Snapshot policy specifies a schedule or schedules on which Snapshot copies are taken and the maximum number of Snapshot copies that each schedule can take. A Snapshot policy can include from one to five schedules.

How it is different from 7-Mode?

- In 7-Mode, there is no concept of snapshot policy and you can set snapshot schedules using *snapshot-set-schedule* API.
- In Cluster-mode, to create schedules that can be used in Snapshot policies, use the *job-schedule-cron-create* or *job-schedule-interval-create* APIs (you can look at *job* category of Cluster-Mode API documentation for further details).
- In Cluster-Mode, you can provide the snapshot policy as one of the optional input arguments to *volume-create* API or you can later use *volume-modify-iter* API to specify the snapshot policy.

3.10 SnapMirror operations

SnapMirror is a feature of Data ONTAP that enables you to replicate data. SnapMirror enables you to replicate data from specified source volumes to specified destination volumes.

1. Create a destination volume on the destination cluster that will become the data protection mirror using the *volume-create* API (See [Section 4.5](#) for how to create a volume).
2. Create a data protection mirror relationship using the *snapmirror-create* API. Data protection mirror relationship will be created, but the relationship is left in an uninitialized state.
3. Initialize the data protection mirror using the *snapmirror-initialize* API.

Where to look for APIs? This is a cluster administrator task, so the APIs for these operations can be found under *snapmirror* category of cluster mode APIs.

How is it different from 7-Mode?

- On Data ONTAP 7-Mode, the endpoint can be a volume or a qtree. On Data ONTAP Cluster-Mode the endpoint can only be a volume.
- On Data ONTAP Cluster-Mode, *snapmirror-initialize* API is usually used after the *snapmirror-create* API, but it can be used alone, that is, without the *snapmirror-create* API, to create and initially update a SnapMirror relationship.
- In 7-Mode, you can set snapmirror schedules using *snapmirror-set-schedule* API whereas in cluster-mode you need to set schedules as explained above (in the case of snapshots).

3.11 Storage Efficiency

Manage deduplication and compression on flexible volumes. In the following API descriptions, these space saving technologies are collectively referred to as "sis".

1. Enable storage efficiency on a volume using *sis-enable* API.
2. Create a new efficiency policy using *sis-policy-create* API. You need to specify Cron type job schedule name while creating a policy. Existing schedules can be queried using the *job-schedule-cron-get-iter* API.

3. Setup or modify sis policy, schedule or options for a volume using ***sis-set-config*** API.
4. Start a sis operation on a volume using ***sis-start*** API.

Where to look for APIs? Since this is a Vserver level task, you can look at *sis* category of Vserver API documentation for further details

How is it different from 7-Mode?

- In Data ONTAP Cluster-Mode while using *sis-set-config* API, policy-name and schedule must not be specified together in the same API call. If schedule is passed, any previous policy-name set on the volume is automatically reset.
- In Data ONTAP 7-Mode, there is no concept of storage efficiency policy and you can set schedules using *sis-set-config* API.

3.12 Performance Monitoring

The *perf* APIs enable monitoring of system performance. They provide a single interface to data across various subsystems in the appliance. Data are organized hierarchically, with objects, instances, and counters. An object generally represents a subsystem, an instance represents a single physical or logical entity within the subsystem, and a counter is a datum specific to the instance.

1. Call ***perf-object-list-info*** API to see the available objects and their priv level.
2. Call ***perf-object-instance-list-info-iter*** API to get the instances for the object name specified as input argument.
3. Call ***perf-object-counter-list-info*** API to get the counters for the object name specified as input argument.
4. Call ***perf-object-get-instances*** API with object name, priv level, instance name or uuid(from step 2 output) to get counter values.

Where to look for APIs? This is a cluster administrator task, so the APIs for these operations can be found under *perf* category of cluster mode APIs.

How is it different from 7-Mode?

No change.