Technical Report

# Secure Multitenancy in ONTAP

## Overview and Design Considerations

Dan Tulledge, NetApp
January 2021 | TR-4160

## Abstract

This report discusses the implementation of secure multitenancy using storage virtual machines (SVMs) in NetApp® ONTAP®, covering design considerations and best practices.

## TABLE OF CONTENTS



### LIST OF TABLES



### LIST OF FIGURES



# Introduction

## Purpose and scope

This technical report describes the implementation of secure multitenancy by using ONTAP. It describes common storage virtual machine (SVM) deployment scenarios and provides SVM best practice recommendations. This report is intended as a reference guide only and is not a replacement for product documentation, ONTAP technical reports, or end-to-end ONTAP operational recommendations. Where possible, these documents are referenced.

## Intended audience

This document is intended for storage architects and storage administrators who want to understand secure multitenancy concepts in ONTAP, different SVM deployment scenarios, and best practices. This document assumes that the reader has a fundamental knowledge of ONTAP architecture, as covered in ONTAP Concepts.

# Overview

## Secure multitenancy (SMT)

### What Is secure multitenancy?

Traditionally, storage consumers who wanted to securely store their data did so by purchasing and deploying one or more physical storage arrays. The physical size of the array could vary based on the needs for capacity and throughput, but typically an entire array of some sort was required in order to provide the secure data and performance isolation that was required.

Secure multitenancy is the use of secure virtual partitions within a shared physical storage environment for the purpose of sharing the physical environment among multiple distinct tenants. For instance, a storage service provider might configure a storage array in such a way that each of three different customers is provisioned a certain portion of the array's disk capacity and network resources. In a secure multitenant environment, each customer would have access only to the resources explicitly provisioned to that customer. The customer would not have access to other customers' data or even be aware of the existence of the other customers or the fact that they share a common physical array. Secure multitenant environments should also provide a means to make sure that no single tenant consumes so much of the shared performance capability so as to affect the other tenants.

### Benefits of secure multitenancy

Traditional siloed models are inefficient. Deploying separate physical hardware stacks for each independent workload is costly and time consuming. Typically, low resource utilization rates in these siloed environments translate directly to wasted resources. In contrast, multitenant environments enjoy the following benefits:

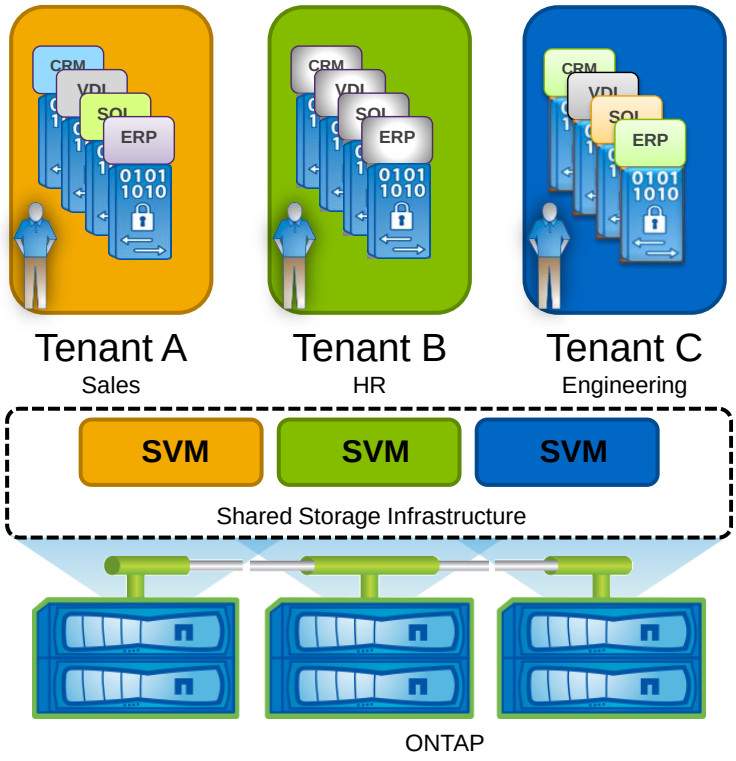
- **Reduced cost**. Dedicating a secure logical storage partition per tenant allows for economies of scale and is more cost effective than dedicated hardware.
- **Faster deployment time.** Logical partitions can be created in a fraction of the time required to rack, cable, install, and configure a separate physical array.
- **Improved resource utilization.** Sharing physical hardware increases resource utilization rates and eliminates the need to overprovision individual workloads in order to account for workload variability.
- **Secure isolation.** Secure multitenancy allows businesses to consolidate tenants onto shared resources, while being assured that tenants will not have access to resources not explicitly assigned

to them. Tenants sharing the same physical hardware can operate independently and with the expectation that no single tenant will consume resources unfairly. For example, production and dev/test can run on the same system without the risk of dev/test affecting production workloads.

### SMT in ONTAP

ONTAP is an inherently multitenant storage operating system. ONTAP is architected in such a way that all data access is done through secure virtual storage partitions. It is possible to have a single partition that represents the resources of the entire cluster or multiple partitions that are assigned specific subsets of cluster resources. These secure virtual storage partitions are known as storage virtual machines, or SVMs.

**Figure 1) SVMs.**




## Storage virtual machines

### Introduction to SVMs

The secure logical storage partition through which data is accessed in ONTAP is known as an SVM. A cluster serves data through at least one and possibly multiple SVMs. An SVM is a logical abstraction that represents a set of physical resources of the cluster. Data volumes and logical network interfaces (LIFs) are created and assigned to an SVM and might reside on any node in the cluster to which the SVM has been given access. An SVM might own resources on multiple nodes concurrently, and those resources can be moved nondisruptively from one node to another. For example, a flexible volume might be nondisruptively moved to a new node and aggregate, or a data LIF could be transparently reassigned to a different physical network port. In this manner, the SVM abstracts the cluster hardware and is not tied to specific physical hardware.

An SVM is capable of supporting multiple data protocols concurrently. Volumes within the SVM can be junctioned together to form a single NAS namespace, which makes all of an SVM's available through a

single share or mount point to NFS and CIFS clients. To illustrate this example to the extreme, imagine a 24-node cluster licensed for UNIX and Windows file services with a single SVM configured with thousands of volumes and accessed from a single LIF on one of the nodes. SVMs also support block-based protocols, and LUNs can be created and exported using iSCSI, FC, or FCoE. Any or all of these data protocols might be configured for use within a given SVM.

Because it is a secure entity, an SVM is only aware of the resources that have been assigned to it and has no knowledge of other SVMs and their respective resources. Each SVM operates as a separate and distinct entity with its own security domain. Tenants can manage the resources allocated to them through a delegated SVM administration account. Each SVM can connect to unique authentication zones such as Active Directory, LDAP, or NIS.

An SVM is effectively isolated from other SVMs that share the same physical hardware.

From a performance perspective, maximums IOPS and throughput levels can be set per SVM by using quality of service (QoS) policy groups. This allows the cluster administrator to quantify the performance capabilities allocated to each SVM.

ONTAP is highly scalable, and additional storage controllers and disks can be easily added to existing clusters in order to scale capacity and performance to meet rising demands. Because virtual storage servers within the cluster, SVMs are also highly scalable. As new nodes or aggregates are added to the cluster, the SVM can be nondisruptively configured to use them. In this way, new disk, cache, and network resources can be made available to the SVM to create new data volumes or migrate existing workloads to these new resources in order to balance performance.

This scalability also enables the SVM to be highly resilient. SVMs are no longer tied to the lifecycle of a given storage controller. As new hardware is introduced to replace hardware that is to be retired, SVM resources can be nondisruptively moved from the old controllers to the new controllers. At this point the old controllers can be retired from service while the SVM is still online and available to serve data.

**Note:** Internal to ONTAP, an SVM is also referred to as a Vserver. In the command examples presented in this document, the `vserver` command is used to perform actions relating to SVMs.

## Components of an SVM

### LIFs

All SVM networking is done through LIFs that are created within the SVM. As logical constructs, LIFs are abstracted from the physical networking ports on which they reside. LIFs are described in further detail in section 3.2.

### Flexible volumes

A flexible volume is the basic unit of storage for an SVM. An SVM has a root volume and can have one or more data volumes. Data volumes can be created in any aggregate that has been delegated by the cluster administrator for use by the SVM. Depending on the data protocols used by the SVM, volumes can contain either LUNs for use with block protocols, files for use with NAS protocols, or both concurrently. For access using NAS protocols, the volume must be added to the SVM namespace through the creation of a client-visible directory called a junction.

### Namespace

Each SVM has a distinct namespace through which all of the NAS data shared from that SVM can be accessed. This namespace can be thought of as a map to all of the junctioned volumes for the SVM, no matter on which node or aggregate they might physically reside. Volumes may be junctioned at the root of the namespace or beneath other volumes that are part of the namespace hierarchy. Namespaces are discussed in section 3 and in further detail in TR-4129: Namespaces in Clustered Data ONTAP.

#### NetApp ONTAP FlexGroup volume

A FlexGroup volume is a single namespace that is made up of multiple constituent/member volumes. It is managed by storage administrators, and it acts like a NetApp FlexVol® volume. Files in a FlexGroup volume are allocated to individual member volumes and are not striped across volumes or nodes. There can be multiple SVMs with FlexGroup Volumes on a given cluster, and they can coexist on the same cluster that also contains SVMs with flexible volumes. For further information about FlexGroup volumes, refer to TR-4037: NetApp ONTAP FlexGroup Volumes.

# SVM design considerations

## SVM layouts

### Single SVM clusters

For many clusters, a single SVM that utilizes all of the physical resources available within the cluster could be the most logical and flexible option. This option is often the simplest to configure and maintain when there is a single tenant who will be consuming all of the available resources of the cluster. However, there are many instances, especially in multi-tenant environments, in which a cluster with multiple SVMs would be the most preferable. These use cases are discussed in the next section. A cluster initially configured with a single SVM can have additional SVMs defined as requirements or needs change.

### Multiple SVM clusters

Although it is possible to have only a single SVM for each cluster, there are also many scenarios where multiple SVMs would be required or offer advantages. Multiple SVMs could be created in any of the following scenarios.

#### Workload separation

A benefit of creating an SVM for individual workloads on the cluster is that it enables the delegation of data management to the IT groups that are directly responsible for the data being stored. Application owners can be given the autonomy to control the datasets belonging to their specific application, while not having administrative rights to SVMs hosting other application workloads or the responsibility of overall cluster administration. When workloads are split off into isolated SVMs, QoS policies can be put into place to provide performance isolation at the SVM/workload level. For workloads with different data replication requirements, splitting into separate SVMs can be an effective way to logically group the volumes and LUNs that compose the workload.

#### NAS/SAN separation

SVMs are, like ONTAP itself, inherently multiprotocol and can serve NAS (CIFS, NFS) and SAN (iSCSI, FCP, FCoE) workloads concurrently. There is no technical requirement that NAS and SAN workloads run in different SVMs. However, there are a few reasons why separating NAS and SAN workloads into separate SVMs might be a good idea.

In many enterprises, there is a division of labor and responsibility when it comes to NAS and SAN. If there is a dedicated storage management team, it is common for that team to administer the storage hardware, including storage arrays and switches. The team also typically handles the creation and masking of LUNs along with creating the Fibre Channel zones required to securely assign LUNs to hosts. It is very likely that this team would be responsible for overall management of the ONTAP cluster. It is a logical extension of the team's storage administration responsibilities to also manage the SVMs that will be providing block data services.

NAS, in contrast, tends to be very tightly coupled with the server teams that consume these storage services. In Windows environments, the server administrators have a vested interest in managing the CIFS shares. Likewise, UNIX administrators tend to be very familiar with managing NFS exports.

With this in mind, enterprises can create NAS SVMs and delegate administrative control to the respective server administrators, without granting them administrative access to the SAN SVMs.

### File server consolidation

When consolidating multiple file servers into a single ONTAP cluster, there are some that will migrate easily into a single SVM, whereas others might require that additional SVMs be created. In general, if file servers to be consolidated share the same authentication services, belong to the same security zones, are managed by the same administrative team, and have no unresolvable overlap in share names, then a single SVM might suffice. Otherwise, creating additional SVMs will allow for file servers being transitioned to ONTAP to be physically consolidated while maintaining the necessary secure logical separation. In general, the recommendation is to consolidate file servers into the smallest number of SVMs that is technically possible while still meeting authentication and performance requirements in order to streamline administration.

### Infrastructure service providers

SVMs allow service providers to securely allocate storage resources to a tenant and delegate management of those resources without dedicating physical hardware to each tenant or exposing multiple tenants and their data to one another. Service providers can create tiers of service based on the types of cluster resources that will be made available to the tenant SVM, such as SSD storage, high-performance nodes with NetApp Flash Cache, Gigabit Ethernet (GbE) versus 10GbE versus 100GbE interfaces, and so on. You can nondisruptively reconfigure volumes and LIFs to use these resources, allowing service providers to maintain high availability for their customers. QoS also allows providers to control the data throughput allocated to each tenant. With QoS policies in place, tenants can share the same physical nodes of a cluster without one tenant consuming an unfair share of the node's resources.

## SVM limits

A minimum of one SVM is required to access data in ONTAP; however, it is possible to create and use many more. The best practice for number of SVMs per cluster is arrived at by considering several factors along with the application and use case environments as discussed earlier:

- The number of nodes in the cluster
- The maximum number of IP and FCP LIFs per node
- The maximum number of IP, iSCSI, and FCP LIFs per port
- Keeping port capacity available to accommodate partner LIFs in the event of HA failover
- Whether SVM management is done using a dedicated management LIF or a combined data/management LIF

For the maximum number of IP LIFS per node, FCP LIFS per node, IP LIFS per port, ISCSI LIFS per port, and FCP LIFS per port, see the NetApp Hardware Universe.

**Note:** An iSCSI LIF is defined as an IP LIF that has the iSCSI protocol enabled.

**Note:** Achieving the maximum iSCSI or FCP LIFs per node will require an adequate number of ports.

### NAS SVMs

For NAS SVMs, the choice to have separate management LIFs or to combine them with data LIFs will affect the number of SVMs it is possible to create. The following configuration options are recommended:

- **Combined data/management LIFs.** In this configuration each SVM requires one active IP LIF, which will be used for combined data and management access.
- **Dedicated management LIF.** In this configuration each SVM requires two IP LIFs: one active management LIF and one active data LIF.

**Note:** In an HA pair, you should take care to make sure that enough LIFs are available on each node's partner so that HA failover is possible. Although the maximum IP LIFs per node is 256, for example, in practical terms this means 128 active LIFs and 128 that would only be instantiated in the case of failover.

For the maximum number of NAS SVMs per cluster, see the NetApp Hardware Universe.

**Note:** It is possible to add more than the minimum required number of LIFs to an SVM. Doing so will reduce the maximum SVMs it is possible to create per cluster.

### SAN-enabled SVMs

For SAN-enabled SVMs, the following configuration options are recommended:

- **FC/FCoE data LIFs with dedicated IP management LIFs.** In this configuration, each SVM requires two FC/FCoE LIFs on each node of the cluster. For management, an IP LIF should be created and dedicated for management use.
- **iSCSI data LIF with dedicated IP management LIF.** In this configuration, each SVM requires one iSCSI LIF on each node of the cluster, which will be used solely for iSCSI data traffic. It is recommended that the LIFs are created on top of interface groups, which combine multiple physical ports into a single virtual port for redundancy. An IP LIF is also needed for management per SVM.

For the maximum number of SAN-enabled SVMs per cluster, see the NetApp Hardware Universe.

**Note:** It is possible to add more than the minimum required number of LIFs to an SVM. Doing so will reduce the maximum SVMs it is possible to create per cluster.

### Mixed SAN and NAS SVMs

Clusters can contain a mixture of NAS-only and SAN-enabled SVMs. The exact number of SVMs that can exist in a single cluster will vary depending on how many of each type are created. The prime factor in determining exactly how many SVMs can be created in a cluster is the number of LIFs that the cluster nodes will support. In general, as long as the cluster has available capacity for the creation of required data and management LIFs while allowing for HA failover, an SVM can be created.

## Tenant tiering

Although SVMs have the potential to use any resource available within the cluster, cluster administrators also have the ability to control exactly to which resources—and which class of resources—a tenant has access. This allows the cluster administrator to implement a tiering strategy whereby different business units, workloads, or customers can be assigned different classes of resources. A small cluster might have a small number of potential tiers; however, a large cluster with multiple controller and disk types can support many tiers.

You can create aggregates of various types: SAS aggregates, SATA aggregates, SSD aggregates, and Flash Pool aggregates, for example. You can provision tenant volumes in the appropriate aggregate based on requirements in place at the time of initial creation. If those needs or requirements change at a later time, cluster administrators can nondisruptively relocate the volumes to another aggregate of a different tier.

Aggregates can be located on nodes of differing capability as well, adding further potential to create a differentiation in tier. You can move workloads between nodes of differing memory and CPU potential, as well as differing amounts of flash-based cache.

For NAS workloads, you can also be nondisruptively migrate LIFs to interfaces of differing capability. For instance, network access can be tiered by bandwidth (GbE versus 10GbE versus 100GbE) or degree of redundancy, such as a single physical port or an interface group composed of multiple physical interfaces.

You can nondisruptively move SVM resources between tiers based on performance and capacity requirements or service-level agreements. You can move datasets with changing requirements throughout their lifecycle among tiers, as required. For example, you can create volumes on high-performance tiers for intense data processing and later moved to cost-effective tiers for archival.

**Note:** It is possible to reassign a volume to a different SVM by using the `volume rehost` command.

## Language considerations

Each SVM has a language setting that is specified at the time of SVM creation. This setting determines the default character set that will be used for the data stored in all volumes within the SVM. This SVM language setting can be changed at a later time if necessary. You cannot change the language for volumes that have been created within the SVM, however. Therefore, it is important to plan language choices ahead of time and be mindful to set them correctly.

One very important factor to consider when choosing an SVM language is whether or not the data will be replicated using SnapMirror data protection mirrors. When replicating volumes from one SVM to another, the source and destination volumes must have the same language setting. It is currently possible to replicate between SVMs whose language settings differ, as long as the volume language settings match. Where possible, it is advisable to choose an appropriate language setting based on technical requirements and standardize the use of this setting for every SVM and volume.

**Best practice**

Unless there are technical requirements that mandate a specific SVM language, consider standardizing SVM deployments using the C.UTF-8 (POSIX with UTF-8) language.

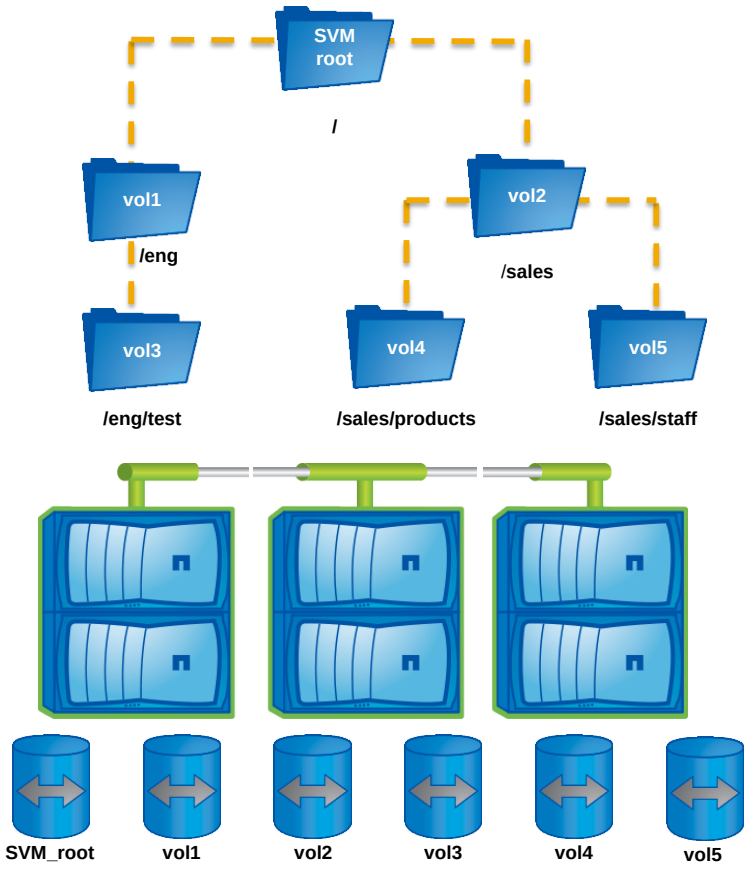
## SVM namespace considerations

### Namespace

For NAS protocols, SVM data is accessed using a single hierarchical directory structure known as a namespace. Each SVM has a namespace that is separate and distinct from the namespaces of any other SVMs in the cluster. FlexVol® volumes created within a given SVM can only be mapped to that SVM's namespace.

Volumes are mapped into the namespace hierarchy through junctions. Each volume can have one junction path at a time, which designates where in the namespace the volume will be placed. A volume's junction path does not have to correspond with the volume name; however, doing so can make it easier to understand where in the namespace a particular volume is junctioned. Volumes can be junctioned at the root of the namespace or beneath another volume that is already junctioned. Figure 2 shows a sample namespace.

**Figure 2) Volumes junctioned into an SVM namespace.**




**Note:** Volumes are the only storage objects that can be junctioned into a namespace; qtrees, volume subdirectories, and individual files cannot. A volume can be junctioned directly off of another volume or off of a user-created subdirectory or qtree.

## Export policies

Access to each volume in the namespace is determined through the creation of an export policy. Export policies dictate which volumes are accessible from which hosts. A volume can have only one export policy, which applies to all data within the volume, including qtrees; however, the same export policy can be applied to many volumes. Each export policy can have multiple rules that specify a host or range of hosts and the type of access they are granted.

An example export policy is as follows:

```
             Policy          Rule    Access   Client                RO
Vserver      Name            Index   Protocol Match                 Rule
------------ --------------- ------  -------- --------------------- ---------
vs1          vsphere         1       any      192.168.1.1           any
vs1          vsphere         2       any      192.168.2.1           any
vs1          vsphere         3       any      192.168.3.0/24        any
```

Volumes that are assigned this export policy will be exported to the two hosts explicitly defined in the first two rules and to the entire subnet defined in the third rule.

Export policies are always created at the volume level. Although it is possible to have qtrees in ONTAP, data in qtrees is exported by creating an export policy for the volume that contains the qtree. A qtree

inherits the export policy of its containing volume. For a FlexClone® volume, which is a space-efficient copy of an existing FlexVol volume, the export policy does not have to match the export policy of the parent FlexVol volume.

For a detailed discussion of namespaces and export policies, see TR-4129: Namespaces in Clustered Data ONTAP.

## SVM networking

### Types of network objects in ONTAP

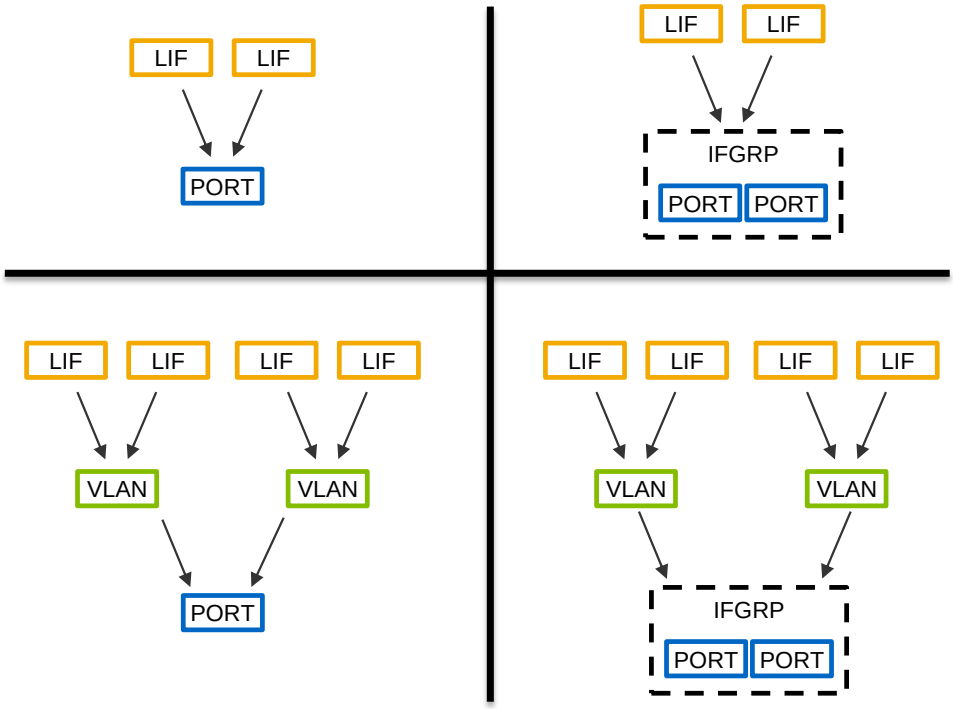
There are three types of network objects in ONTAP: physical, virtual, and logical.

The first type is a simple physical port, such as a GbE port or 10GbE or 100GbE port.

The second type is a virtual network interface such as an interface group or a virtual LAN (VLAN). Interface groups aggregate multiple physical ports into a single virtual interface for redundancy. VLAN interfaces partition a single physical port or interface group into multiple isolated broadcast domains.

Lastly, LIFs are created as an abstraction on top of the physical or virtual interface layer. IP-based LIFs for NAS or iSCSI are assigned IP addresses, and FC-based LIFs are assigned WWPNs. LIFs are the only interface type that gets assigned directly to an SVM. Management LIFs and data LIFs used for NAS protocols are mobile and can be nondisruptively reassigned or failed over to a new physical port, interface group, or VLAN. Data LIFs used for SAN protocols do not require the same mobility and do not migrate or fail over to new physical ports. This is because MPIO and ALUA in ONTAP automatically choose the most optimal port for host traffic. However, it is possible for SAN LIFs to be first brought offline and then reassigned to a new home port. This provides a means to accommodate new hardware being added to the cluster and the retirement of existing hardware. Note that bringing a SAN LIF offline does not mean that a host will lose access to mapped LUNs as LUNs are exported through multiple SAN LIFs concurrently, and LUN pathing is managed through multipath I/O software on the host.

**Figure 3) Types of LIF configurations in ONTAP.**



## LIFs

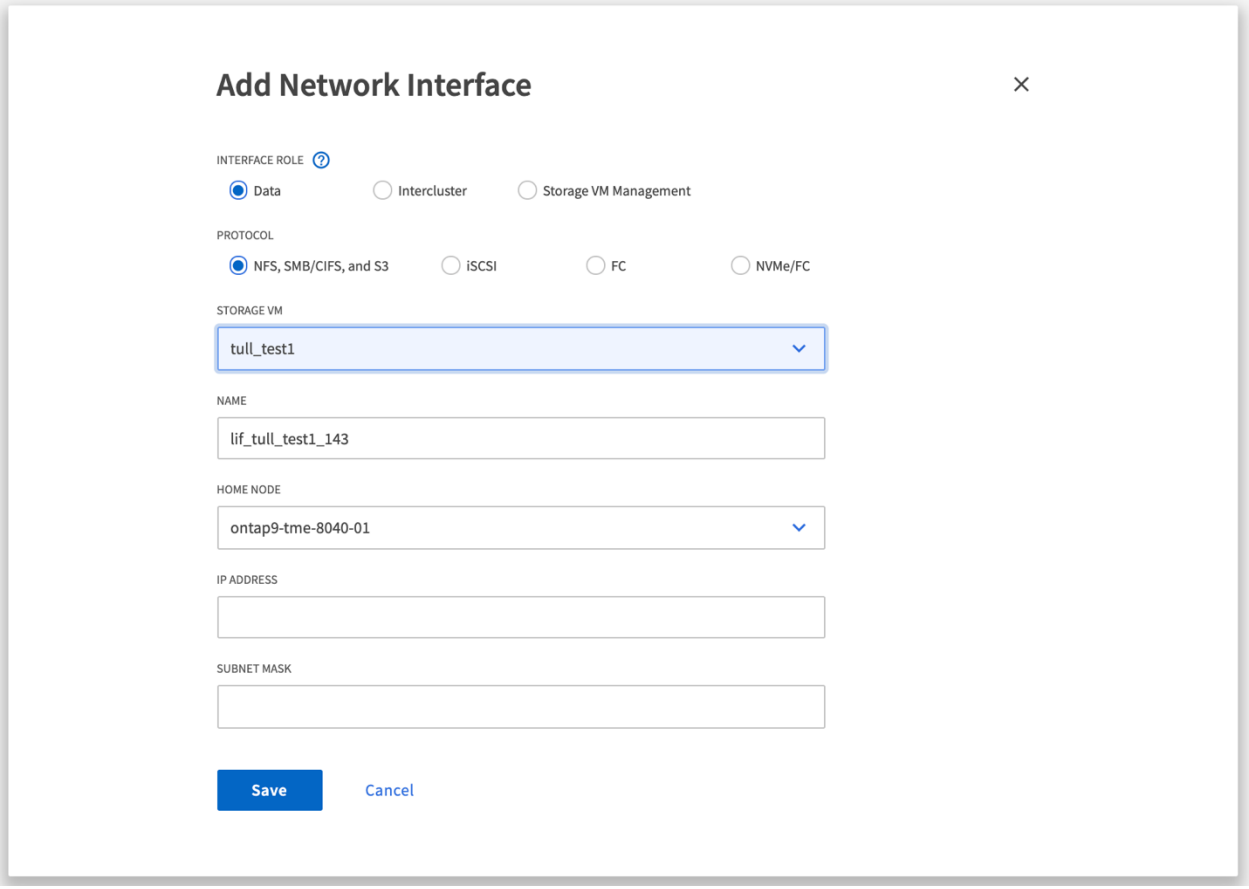
There are several types of LIF in ONTAP, and each is used for a distinct purpose. Table 4 lists the LIF types and their functions.

**Table 1) Types of LIFs.**

| LIF type | Function | Minimum required | Maximum allowed |
|---|---|---|---|
| Node management | Used for system maintenance of a specific node, SNMP, NTP, and ASUP™ | 1 per node | 1 per port/subnet |
| Cluster management | Management interface for the entire cluster | 1 per cluster | N/A |
| Cluster | Used for intracluster traffic | 2 per node | 2 per node |
| Data | Associated with an SVM and used for data protocols and protocol services (NIS, LDAP, Active Directory, WINS, DNS) | 1 per SVM | 128 per node in HA config<br>256 per node in non-HA |
| Intercluster | Used for intercluster communication, such as setting up cluster peers and SnapMirror traffic | 1 per node if cluster peering is enabled | N/A |

You can use data LIFs for data protocol access, SVM management access, or both, as shown in Figure 4.

**Figure 4) LIF creation options as seen in ONTAP System Manager.**




## SVM resource placement

One of the key features of SVMs in ONTAP is that each is a logical entity that exists on the cluster, not bound to any single controller or HA pair. Because of this, SVMs can contain resources from any node in the cluster and from multiple nodes concurrently. This empowers administrators with a great amount of flexibility. For example, data volumes for an SVM can reside on a single aggregate, or they can be distributed across multiple aggregates on multiple nodes. Using the data mobility features of ONTAP, these volumes can be relocated to different aggregates nondisruptively, even if the new aggregate is on a different node. Likewise, data LIFs are logical and can be moved nondisruptively to new physical ports, VLANs, or interface groups. These ports can theoretically be on any node of the cluster; however, care must be taken to make sure that the LIF is moved to a physical port that is connected to an appropriate physical network. NAS clients can connect to shares or exports using an SVM's data LIF on any node and access all of the SVM's data volumes regardless of the nodes and aggregates in which those volumes are contained. This allows for unprecedented flexibility at the physical level to introduce new resources to the cluster, retire resources from the cluster, and balance workload and capacity across the cluster.

## SAN LIF considerations

Unlike LIFs used for NAS data access, LIFs that will be used for iSCSI, FCP, or FCoE access do not migrate from their assigned home physical port. Additionally, LIFs used for iSCSI cannot be used for any other protocol, such as NFS or CIFS, and must be dedicated for use with iSCSI. Consequently, a SAN LIF of the appropriate type is typically created on each node of the cluster.

It is expected that hosts will use multipath I/O drivers and determine the optimum LIFs to use for LUN access through the use of Asymmetric Logical Unit Access (ALUA).

For host operating systems that have upper limits on the number of paths per LUN, this could be a challenge in large clusters with many nodes and in smaller clusters with many available paths per node. To reduce the number of paths available to a host, consider using portsets. Portsets can limit the number of paths available per node or the number of nodes through which the LUN is available.

For further discussion of SAN best practices in ONTAP, see TR-4080: Best Practices for Modern SAN

**Best practice**

For SVMs using the iSCSI protocol, consider creating a LIF dedicated to SVM management with no data protocol access and placing that LIF in an appropriately configured failover group. Since iSCSI data LIFs do not migrate, this configuration promotes high availability for the management interface.

## Network isolation of tenants

An important aspect of multitenancy is securing network traffic so that tenants can be securely isolated from one another. Although it might be desirable for SVMs in an enterprise context to share a common IP network, SVMs used by individual tenants within a shared infrastructure environment should not share IP networks and should remain separated. Both of these goals can be accomplished through the use of routing groups.

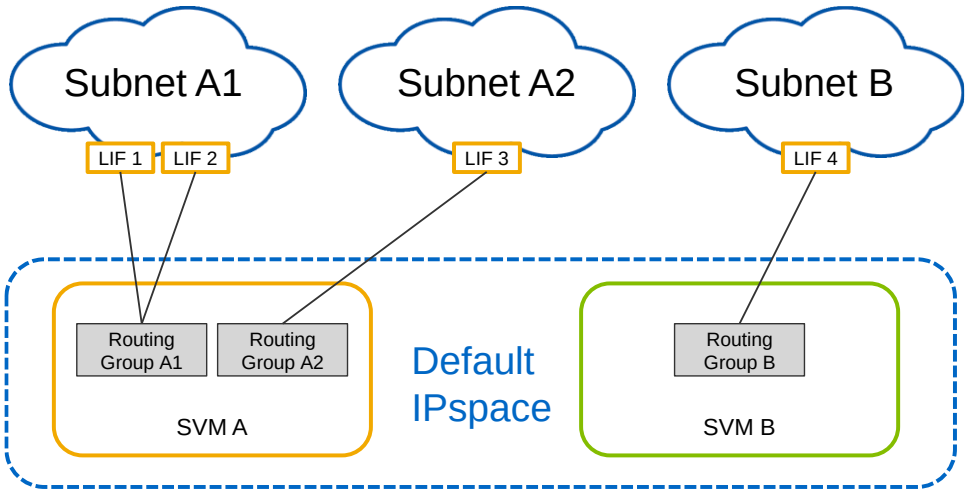
### Routing groups

Routing groups are used with SVMs to control outbound network traffic for the LIFs belonging to the SVM. Each routing group is a separate and distinct routing table. It is possible for an SVM to have more than one routing group, but routing groups are never shared between SVMs. Each LIF belonging to an SVM is associated with one and only one routing group. Multiple LIFs in the same SVM can share a common routing group and must be on the same IP subnet. Routing groups provide secure, segregated traffic forwarding and SVM-scoped network administration and control.

### Single IP network

A common deployment scenario for the enterprise is to provide multitenancy at the workload, departmental, or administrative level. In this use case, a single enterprise-wide IP network is deployed. Here, routing groups scoped to a specific SVM provide the required control and separation for packet forwarding and network administration.

**Figure 5) Single enterprise-wide IP network.**

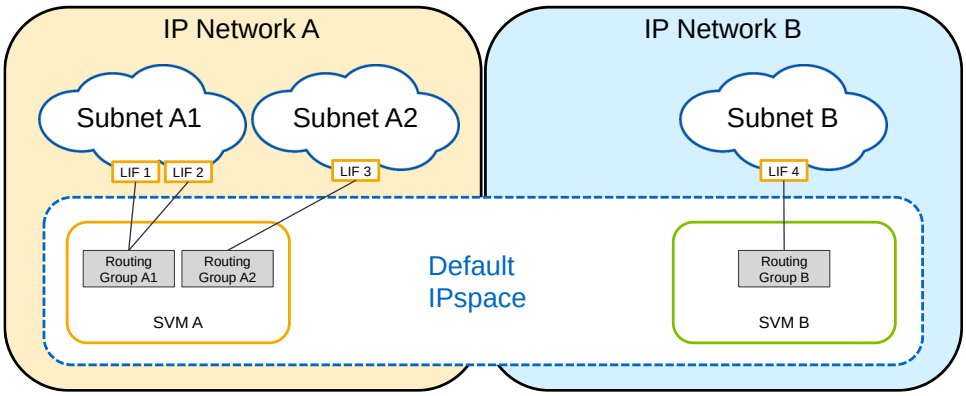


## Multiple IP networks

Another use case for secure multi-tenant network configuration is that of an enterprise or service provider whose tenants require segregated IP networks, each with a unique nonoverlapping IP address range. Examples of this could be an enterprise with test/dev SVMs on a separate network from production or that host data in a DMZ. In this instance, each SVM is confined to its own IP network, consisting of one or more subnets. The SVM LIFs would be created on top of a VLAN interface in order to securely scale the physical resources of the cluster and allow for many secure SVMs to be created. Because each routing group represents a distinct routing table, traffic is not routed between SVMs.

**Note:** The subnets depicted here could be connected to different networks entirely, such as a separate VLAN. It is possible through the use of multiple LIFs for the same SVM to be connected to multiple VLANs.

**Figure 6) Multiple nonoverlapping IP networks within the same enterprise.**




Configurations that require multiple IP networks and where identical IP address ranges are reused among tenants are facilitated by using IPspaces.

## Administrative networking control

Delegated SVM administrators using the predefined SVM administrative role have access only to a subset of the networking commands available to a cluster administrator. Only cluster administrators have the ability to move LIFs to different physical interfaces or VLANs. This is inherently secure from an SVM admin perspective, because the SVM admin does not have the potential to mistakenly reassign the LIF to an invalid port. It is also possible to create cluster administrator roles that would allow for administration of the cluster but restrict the ability to directly move LIFs or modify LIF failover and routing groups. The following is an example of such a custom role.

```
cluster::> security login role show -role NoNetworkAccess
  (security login role show)
           Role          Command/                                    Access
Vserver    Name          Directory                             Query Level
---------- ------------- --------- ----------------------------------- --------
cluster  NoNetworkAccess
                         DEFAULT                                       all
cluster  NoNetworkAccess
                         network interface failover-groups             none
cluster  NoNetworkAccess
                         network interface migrate                     none
cluster  NoNetworkAccess
                         network interface migrate-all                 none
cluster  NoNetworkAccess
                         network routing-groups                        none
```

Roles are covered in further detail in the following section.

For an in-depth discussion of networking best practices, see TR-4847: Best Practices for Clustered Data ONTAP Network Configurations.

# SVM security

## Users and roles

### Administrative users

There are default administrative user accounts within ONTAP as well as a robust means to create users with a customized set of privileges. The default cluster administrator is the `admin` user. Cluster administrators have the ability to administer the entire cluster and all of its resources. For SVMs, the default administrator is the `vsadmin` user. Although the `vsadmin` user is created with every SVM, it must be explicitly enabled in order to delegate administration of the SVM. SVM administrators may only administer their respective SVMs.

### User roles

A role in ONTAP is a collection of access control rules that specify what type of access a user will have to a given command directory, command subdirectory, or command.

There are predefined roles within ONTAP for both the cluster and SVM contexts.

**Table 2) Default cluster user roles.**

| Role | Access level | Capabilities |
|---|---|---|
| admin | All | All |
| readonly | Readonly | Read-only |
| none | None | None |

**Table 3) Default SVM user roles.**

| Role | Default capabilities |
|---|---|
| vsadmin | • Manage your own user account local password and public key<br>• Manage volumes, quotas, qtrees, Snapshot™ copies, FlexCache® files, and files<br>• Perform NetApp SnapLock® operations, except privileged delete<br>• Manage LUNs<br>• Manage data protection mirrors<br>• Configure protocols<br>• Configure services<br>• Monitor jobs<br>• Monitor network connections and network interface<br>• Monitor the health of an SVM |
| vsadmin-volume | • Manage your own user account local password and public key<br>• Manage volumes, quotas, qtrees, Snapshot copies, FlexCache files, and files<br>• Manage LUNs<br>• Configure protocols<br>• Configure services<br>• Monitor network interfaces |

| Role | Default capabilities |
|---|---|
| | • Monitor the health of an SVM |
| vsadmin-protocol | • Manage your own user account local password and public key<br>• Configure protocols<br>• Configure services<br>• Manage LUNs<br>• Monitor network interfaces<br>• Monitor the health of an SVM |
| vsadmin-backup | • Manage your own user account local password and public key<br>• Manage NDMP operations<br>• Create a restored volume read/write<br>• Manage NetApp SnapMirror® relationships and Snapshot copies<br>• View volumes and network information |
| vsadmin-snaplock | • Manage your own user account local password and public key<br>• Manage volumes except volume moves<br>• Manage quotas, qtrees, Snapshot copies, and files<br>• Perform SnapLock operations, including privileged delete<br>• Configure protocols<br>• Configure services<br>• Monitor jobs<br>• Monitor network connections and network interface |
| vsadmin-readonly | • Manage your own user account local password and public key<br>• Monitor the health of an SVM<br>• Monitor network interface<br>• View volumes and LUNs<br>• View services and protocols |

## Custom users and roles

In addition to the default users and roles, it is possible to create additional cluster and SVM users and to define custom roles that specify to what commands those users will have access.

**Table 7) Role-based access control (RBAC) definitions.**

| Capability | Access | Description |
|---|---|---|
| Directory/subdirectory | All | Permits access to the directory and all subdirectories and commands contained within |
| | Readonly | Permits read-only access to the directory and subdirectories. |
| | None | Denies access to the directory and all subdirectories and commands contained within |
| Command | All | Permits execution of the command |

| Capability | Access | Description |
|---|---|---|
| | None | Denies execution of the command |

Roles are defined in a hierarchical manner from general to specific. Rules defined for specific commands or subdirectories will override rules that are defined for their parent directory.

### Creating users with the CLI

To create a new user, use the `security login create` command. In the following example the user is created locally on the ONTAP cluster and logs in using a password using Secure Shell (SSH). It is also possible to specify a domain or LDAP user or to specify the use of a public key in lieu of a password.

```
cluster::> security login create -username newuser -application ssh -authmethod password -role
customrole -vserver vs1

Please enter a password for user 'newuser':
Please enter it again:

cluster::> security login show -vserver vs1 -username newuser

Vserver: vs1
                            Authentication                  Acct
UserName         Application Method         Role Name        Locked
---------------- ----------- -------------- ---------------- ------
newuser          ssh         password       customrole       no
```

### Creating roles with the CLI

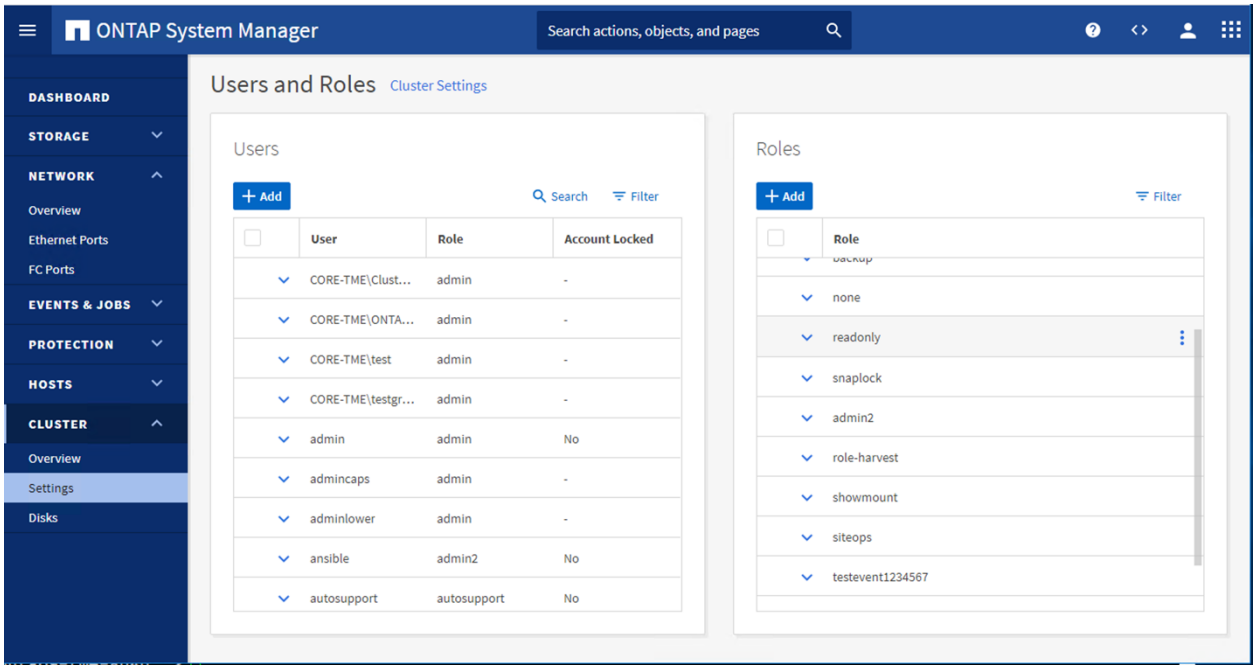
Roles are created in the CLI using the `security login role create` command. The role is created one rule at a time. The following series of commands creates the custom role discussed in the administrative networking control section earlier.

```
role create -vserver cluster -role NoNetworkAccess -cmddirname "network interface failover-
groups" -access none

role create -vserver cluster -role NoNetworkAccess -cmddirname "network interface migrate" -
access none

role create -vserver cluster -role NoNetworkAccess -cmddirname "network interface migrate-all" -
access none

role create -vserver cluster -role NoNetworkAccess -cmddirname "network routing-groups" -access
none

role modify –vserver cluster –role NoNetworkAccess –cmddirname DEFAULT –access all
```

### Creating users and roles with ONTAP System Manager

To create and edit custom roles with ONTAP System Manager, navigate to Cluster -> Settings -> Users and Roles.

**Figure 7) Users and Roles in ONTAP System Manager.**



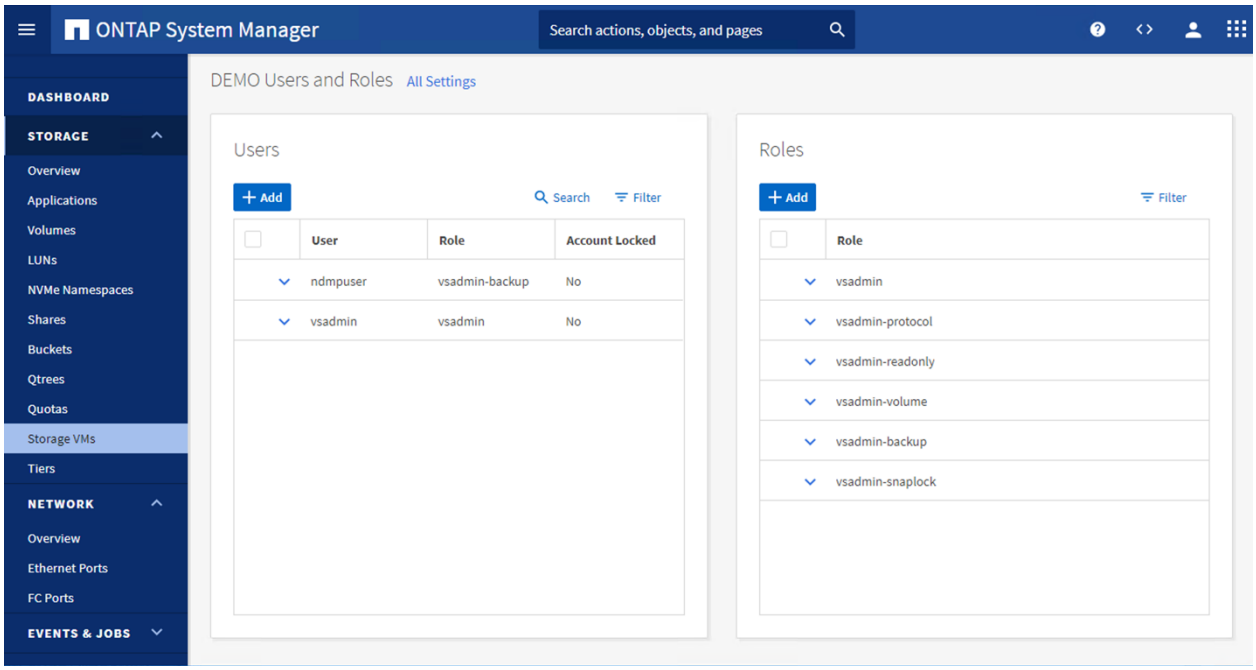

## SVM administrator delegation

In addition to users who have administrative capabilities at the cluster level, it is possible for each SVM to have accounts enabled that have administrative rights only for that specific SVM. Each SVM has a `vsadmin` account created by default, but it must explicitly be enabled. To enable the `vsadmin` account, assign a password and then unlock the account.

```
cluster::> security login password -username vsadmin -vserver vs1

Enter a new password:
Enter it again:

cluster::> security login unlock -username vsadmin -vserver vs1

cluster::> security login show -username vsadmin -vserver vs1

Vserver: vs1
                              Authentication                  Acct
UserName         Application  Method         Role Name        Locked
---------------- -----------  -------------- ---------------- ------
vsadmin          ontapi       password       vsadmin          no
vsadmin          ssh          password       vsadmin          no
2 entries were displayed.
```

It is also possible to enable the SVM administrator through System Manager; navigate to Storage -> Storage VMs -> <SVM name> -> Settings -> Users and Roles.

**Figure 8) Delegating SVM administration.**




## Aggregate delegation

When delegating administrative access to an SVM admin, it is important to also delegate the aggregates that the SVM can use for provisioning new flexible volumes. A cluster administrator can use any aggregate in the cluster to create a new volume for an SVM or to relocate an existing flexible volume. An SVM admin, however, might only create new volumes in the aggregates that have been delegated to that SVM. The delegated aggregates are enumerated in the `aggr_list` option for the SVM.

In the CLI, a cluster admin can delegate aggregates by using the `vserver modify` command.

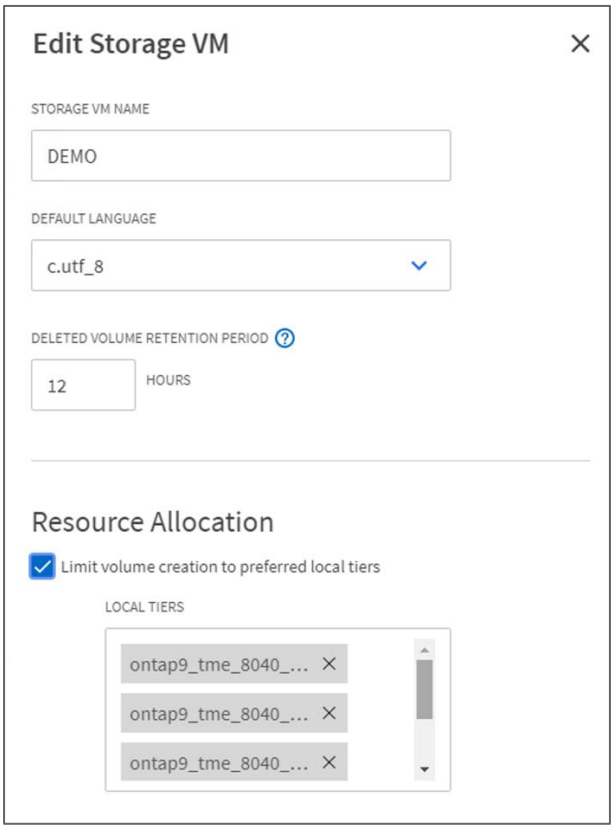
```
cluster::> vserver modify -vserver vs1 -aggr-list aggr_data1,aggr_data2
```

The SVM show command can be used to check which aggregates have been assigned.

```
cluster::> vserver show -vserver vs1 -fields aggr-list
vserver aggr-list
------- --------------------
Vs1     aggr_data1,aggr_data2
```

It is also possible to delegate aggregates through System Manager; navigate to Storage -> Storage VMs -> <SVM name> -> Overview -> Edit.

**Figure 9) Delegating aggregates for volume creation.**




**Note:** In System Manager 9.8, aggregates are called local tiers.

## Firewall rules

ONTAP includes functionality to restrict access to management service protocols available on the system. A number of system-defined firewall policies are included, and custom policies can be created using the `system services firewall policy` command directory.

The following example displays the firewall rules in place for the system-defined `mgmt` policy.

```
system services firewall policy show -policy mgmt -vserver  demo
Vserver Policy       Service    Allowed
------- ------------ ---------- -------------------
demo
        mgmt
                     dns        0.0.0.0/0, ::/0
                     http       0.0.0.0/0, ::/0
                     https      0.0.0.0/0, ::/0
                     ndmp       0.0.0.0/0, ::/0
                     ndmps      0.0.0.0/0, ::/0
                     ntp        0.0.0.0/0, ::/0
                     portmap    0.0.0.0/0
                     snmp       0.0.0.0/0, ::/0
                     ssh        0.0.0.0/0, ::/0
```

To create a custom firewall policy using the `mgmt` policy as a base, the policy can be cloned using the `system services firewall policy clone` command.

```
system services firewall policy clone -policy mgmt -destination-policy mgmt-custom -vserver demo
-destination-vserver demo2
```

To customize the newly cloned firewall policy, use the system services firewall policy modify command. For example, to allow SSH access only from the 192.168.1.0/24 subnet, use the following:

```
system services firewall policy modify -policy mgmt-custom -service ssh -allow-list
192.168.1.0/24 -vserver demo2
system services firewall policy show mgmt-custom
Vserver Policy       Service    Allowed
------- ------------ ---------- -------------------
demo2
        mgmt-custom
                     dns        0.0.0.0/0, ::/0
                     http       0.0.0.0/0, ::/0
                     https      0.0.0.0/0, ::/0
                     ndmp       0.0.0.0/0, ::/0
                     ndmps      0.0.0.0/0, ::/0
                     ntp        0.0.0.0/0, ::/0
                     portmap    0.0.0.0/0
                     snmp       0.0.0.0/0, ::/0
                     ssh        192.168.1.0/24
9 entries were displayed.
```

To assign the firewall policy to a LIF, modify the `-firewall-policy` attribute of the desired LIF.

```
cluster::> network interface modify -vserver vs1 -lif vs1_mgmt -firewall-policy mgmt-custom

cluster::> network interface show -vserver vs1 -lif vs1_mgmt

                    Vserver Name: vs1
          Logical Interface Name: vs1_mgmt
               (Deprecated)-Role: data
                   Data Protocol: nfs
               Network Address: 192.168.1.1
                       Netmask: 255.255.255.0
            Bits in the Netmask: 24
                      Home Node: cluster-01
                      Home Port: e0a
                   Current Node: cluster-01
                   Current Port: e0a
              Operational Status: up
                 Extended Status: -
                      Numeric ID: 1025
                         Is Home: true
           Administrative Status: up
                 Failover Policy: nextavail
                 Firewall Policy: mgmt-custom
                     Auto Revert: false
   Fully Qualified DNS Zone Name: none
         DNS Query Listen Enable: false
             Failover Group Name: system-defined
                        FCP WWPN: -
                  Address family: ipv4
                         Comment: -
```

## SVM-scoped external KMIP servers for NetApp Volume Encryption (NVE)

Starting with ONTAP 9.6, you can use an SVM scope to configure external key management for a named SVM in the cluster. This is best for multitenant environments in which each tenant uses a different SVM (or set of SVMs) to serve data. Only the SVM administrator for a given tenant has access to the keys for that tenant. SVM-scoped external key managers can coexist with cluster-scoped external key managers. You can configure onboard key management at the cluster scope and external key management at the SVM scope. You can use the `security key-manager key migrate` command to migrate keys from onboard key management at the cluster scope to external key managers at the SVM scope.

For more details on configuring SVM-scoped external KMIP servers for NVE, see the NetApp Encryption Power Guide.

## SVM performance monitoring and isolation

### Using storage quality of service

ONTAP includes a storage quality of service (QoS) feature that allows cluster administrators to manage system performance by setting maximum throughput thresholds for a variety of storage objects. By assigning storage QoS policies to a storage object, the administrator can make sure that the object does not consume more cluster resources than is expected.

#### Storage objects

The following storage objects can have storage QoS policies applied to them:

- SVM
- FlexVol volume
- LUN
- File

For purposes of this paper, we will focus on applying storage QoS policies at the SVM layer.

#### Policy groups

Policy groups are used to define throughput limits for a given SVM. It is also possible to create and assign a policy group that does not have a throughput limit set. This allows for monitoring of SVM throughput without enforcing limits. A maximum of 12,000 policy groups may be created per cluster, and up to 40,000 storage objects can be assigned to those groups.

Policy groups are created by using the `qos policy-group create` command. Existing groups can be modified with the `qos policy-group modify` command. Throughput limits can be defined using IOPS or MB/s as metrics.

```
cluster::> qos policy-group create -policy-group pg2 -vserver vs2 -max-throughput 1000iops

cluster::> qos policy-group modify -policy-group pg1 -max-throughput 1000MB/S
```

Policy groups can be displayed using the `qos policy-group show` command.

```
cluster::> qos policy-group show
Name             Vserver     Class        Wklds Throughput   Is Shared
---------------- ----------- ------------ ----- ------------ ---------
pg1              vs1         user-defined 0     0-1000MB/S.  true
pg2              vs2         user-defined -     0-1000IOPS.  true
2 entries were displayed.
```

#### Assigning storage QoS policy groups to SVMs

After a policy group is defined, it can be assigned to an SVM by modifying the `-qos-policy-group` attribute of the SVM.

Per SVM, only one type of storage object can be assigned to a QoS policy group. If the entire SVM is added to a policy group, then one cannot add specific volumes and LUNs to policy groups as well.

**Best practice**

Standardizing on the use of storage QoS policies at the SVM level allows cluster administrators to apply per-tenant throughput limits. Since SVM admins can create volumes and LUNs but cannot create or assign storage QoS policies to them, assigning policies to the SVM is a means for the cluster

administrator to make sure that each storage object within a given SVM is covered under a storage QoS policy after being initially assigned.

The following is an example of applying a QoS policy at the SVM level:

```
cluster::> vserver modify -vserver vs1 -qos-policy-group pg1

cluster::> vserver show -vserver vs1 -fields qos-policy-group
vserver qos-policy-group
------- ----------------
vs1     pg1
```

### Monitoring SVM performance with QoS

When a storage object is assigned to a QoS policy group, this is said to define a QoS workload. QoS can provide a vast amount of detailed information regarding workload performance using the `qos statistics` command and its various options, including workload characteristics, latency, disk utilization, and CPU utilization. A complete discussion is beyond the scope of this document; however, you can find more information in the ONTAP QoS statistics documentation.

## Data protection

ONTAP provides a means to replicate data within the same cluster or to a peered cluster. Both load-sharing mirrors and data protection mirrors are supported. For detailed information on peering and mirroring, see TR-4015: SnapMirror Configuration and Best Practices Guide for ONTAP 9.

### Cluster and SVM peering

It is possible to create intracluster volume mirrors for load sharing and data protection. It is also possible to create intercluster mirrors if special intercluster LIFs are set up on both the source and destination cluster and the clusters are connected in what's known as a peer relationship. Replication traffic between the two clusters will occur over the intercluster LIFs.

ONTAP includes the concept of SVM peering in addition to cluster peering. For both intracluster mirroring and intercluster mirroring between peered clusters, the SVMs containing the source and destination volumes must also be in a peer relationship. This allows for a higher level of granularity in the control of data protection mirrors and provides a foundation for delegating the control of data protection mirrors to SVM administrators.

**Note:** SVMs can only be peered if they are within the same cluster or if their containing clusters are also peered.

SVM peers can be created and shown with the `vserver peer` command. The following example sets up an SVM peer relationship between two SVMs on the same cluster:

```
cluster::> vserver peer create -vserver vs1 -peer-vserver vs2 -applications snapmirror

Info: 'vserver peer create' command is successful.

cluster::> vserver peer show-all
            Peer        Peer                                  Peering         Remote
Vserver     Vserver     State        Peer Cluster             Applications    Vserver
----------- ----------- ------------ ------------------------ --------------- ---------------
vs1         vs2         peered       cluster                  snapmirror      vs2
vs2         vs1         peered       cluster                  snapmirror      vs1
2 entries were displayed.
```

Because the two SVMs to be peered are in the same cluster, no further action is required after issuing the peer create command. If, however, the SVMs to be peered are in separate clusters that are in a cluster peer relationship, then the SVM peer relationship will be in a pending state until the cluster administrator of the remote cluster accepts the peer relationship.

```
cluster2::> vserver peer show

            Peer        Peer
Vserver     Vserver     State
----------- ----------- ------------
vs1          vs3         pending
```

To accept the SVM peer request, the remote cluster administrator should issue the `vserver peer accept` command.

```
cluster2::> vserver peer accept -vserver vs1 -peervserver vs3
```

### Unique SVM naming requirements

In order to peer two SVMs, their names must be unique on both the source and destination clusters. For instance, for an SVM called vs1 on cluster1 to be peered with SVM vs2 on cluster2, cluster1 cannot have an SVM called vs2, and cluster2 cannot have an SVM called vs1.

**Best practice**

Adopt a naming scheme that will make sure that SVM names are unique across peered clusters. One means to accomplish this is to name each SVM using a fully qualified domain name.

## Language considerations

Volumes can have a different language than the SVM in which they are contained. By default, a volume will inherit the language of the containing SVM if no language is specified when the volume is created. It is possible to specify a different language for the volume at creation time. It is possible to change the language of an SVM; however, volume language cannot be changed. SnapMirror copying between SVMs of different language types can occur, but the source and destination volumes must have the same language.

The default language for SVMs in ONTAP 8.2 is C.UTF-8. This setting provides a neutral, non–country specific encoding. Unless technical requirements dictate the use of a different language encoding, consider using C.UTF-8.

UTF-8 encoding is variable length. The presence or absence of certain bits in a byte of UTF-8 indicates the number of bytes that make up the encoded character. Therefore, the use of certain special characters in [language] will cause errors if [language].UTF-8 parsing is used.

Use these qualifying questions to determine the best option for the SVM language encoding:

- Are there any older CIFS clients (Windows® 95/95/ME)? If yes, match the SVM language encoding to the [language] client locale.
- Are there any NFSv2/3 clients not using UTF-8? If yes, match the SVM language encoding to the [language] client locale.
- Are all the CIFS clients post Windows 95/98/ME and all the NFS clients using a UTF-8 locale? If so, set the SVM language encoding to C.UTF-8. If some clients are not using UTF-8, set the SVM language encoding to the [language] client locale.

Table 7 describes how to specify an SVM's language. en_US is used in each example but can be substituted with the appropriate client locale.

**Table 4) SVM language recommendations.**

| Client protocol | Client encoding type | SVM language recommendation |
|---|---|---|
| CIFS (Windows 95/98/ME) | ISO 8859-1 | Use [language]. Example: "en_US." |

| Client protocol | Client encoding type | SVM language recommendation |
|---|---|---|
| CIFS (Windows NT® 3.1+) | UCS-2 | • If all clients use UTF-8, use C.UTF-8.<br>If any client does not use UTF-8, use [language]. Example: “en_US.” |
| NFSv2/3 | Non-UTF-8 client locale | Use [language]. Example: “en_US.” |
| NFSv2/3 | UTF-8 client locale | Use C.UTF-8. |
| NFSv4 | UTF-8 | Use C.UTF-8. |
| FC or iSCSI | N/A | C.UTF-8 preferred; C/POSIX is acceptable. |

In deployments with mixed non-Unicode client language encodings, the SVM language encoding should match the highest priority client language encoding.

Incorrectly displayed characters might not be remediable in situations where different clients use the same file access protocol (for example, NFSv3), but with different language encodings. The same is true when clients have a mix of client locales.

The following best practices should also be considered when selecting an SVM’s language setting:

- For environments using both CIFS and NFS clients, match the NFS client language encoding.
- Do not use non-ASCII characters in your file names, including “smart quotes” and currency symbols (for example, € for euro, £ for UK pound).
- UTF-8 language encoding is preferred over non-UTF-8 encodings. However, there are situations in which this is not recommended. For example, when NFS clients are using non-UTF-8 encodings, the SVM’s language should be set to the same non-UTF-8 encoding.
- When sharing files between CIFS and NFS, only use characters that are legal for both and are in the NFS character set.
- For customers planning to move from NFSv3 to NFSv4, the SVM’s language should match the NFSv3 client language.
- If a volume inside an SVM will be a SnapMirror destination, choose the same language used on the volume that will be the SnapMirror source.

## Load-sharing mirrors

Load-sharing (LS) mirrors are a special type of SnapMirror that can increase performance and availability of volumes accessed by using CIFS or NFSv3.You can use LS mirrors to distribute reads across multiple nodes for datasets that are read-only. You can also use LS mirrors to increase the availability of an SVM’s namespace when used to create multiple instances of the SVM root volume.

Creating LS mirrors of an SVM root volume on each node of the cluster provides multiple redundant copies that can be used should the primary volume become unavailable. If the root volume should become temporarily unavailable, read access to the volume is provided through the LS mirrors, and the namespace remains available. If the root volume is permanently unavailable, one of the mirrors can be promoted to make write access available.

There are some trade-offs that must be made in order to achieve this degree of availability. Because the SVM root volume is the location of the SVM namespace root, any new volumes that are junctioned into the namespace at the root are not visible until the LS mirror set has been updated. This update can only be done by a cluster administrator, because the command required (`snapmirror update-ls-set`) is not available at the SVM level. The cluster administrator can schedule these updates to occur on a regular basis. SVM admins should be mindful that updates to the namespace will only be visible after the

mirror set is updated. Any automated workflows that operate under cluster credentials and update the namespace should also update the LS mirror set.

**Best practice**

For added resiliency and namespace availability, consider creating a load-sharing mirror set for SVM root volumes.

### Data protection mirrors

ONTAP provides the ability to asynchronously mirror volumes from one SVM to another SVM on the same or a different cluster by creating a SnapMirror data protection mirror. These data protection mirrors are used to maintain local backup copies or to provide a remotely replicated copy, which can be used for disaster recovery and business continuance.

### Vaulting

The NetApp SnapVault® feature allows for asymmetric Snapshot retention between source and destination volumes. Local Snapshot copies could be retained for short-term backup and recovery, while SnapVault copies are retained for long-term archival storage.

## Management tools

### CLI

The ONTAP command line interface is available to cluster administrators as well as SVM administrators. Cluster administrators can use SSH to connect to the cluster management LIF in order to access the CLI in a cluster context. From this context, the admin can manage items related to the cluster as a whole, as well as each SVM on the cluster.

SVM administrators can use SSH to connect directly to the CLI in an SVM context. From this context they are able to manage items directly pertaining to the SVM to which they have been delegated access. SVM admins are able to manage data protocols and services; storage objects such as volumes, LUNs, and qtrees; and Snapshot copies of volumes and monitor the overall health of the SVM.

Cluster admins, once logged in to the CLI, can switch to an SVM context using the `vserver context` command.

The command line interface is currently the only interactive management tool available to SVM administrators.

**Note:** There is a maximum of 250 concurrent SSH sessions in ONTAP.

### NetApp Manageability SDK

ONTAP includes a rich API set available through the NetApp Manageability SDK. The SDK provides APIs and sample code to develop custom management tools using C, C++, Java®, Perl, C#, VB.NET, Windows PowerShell™, Python, and Ruby. You can use these APIs to integrate ONTAP management with existing orchestration tools or create custom management portals.

### ONTAP PowerShell Toolkit

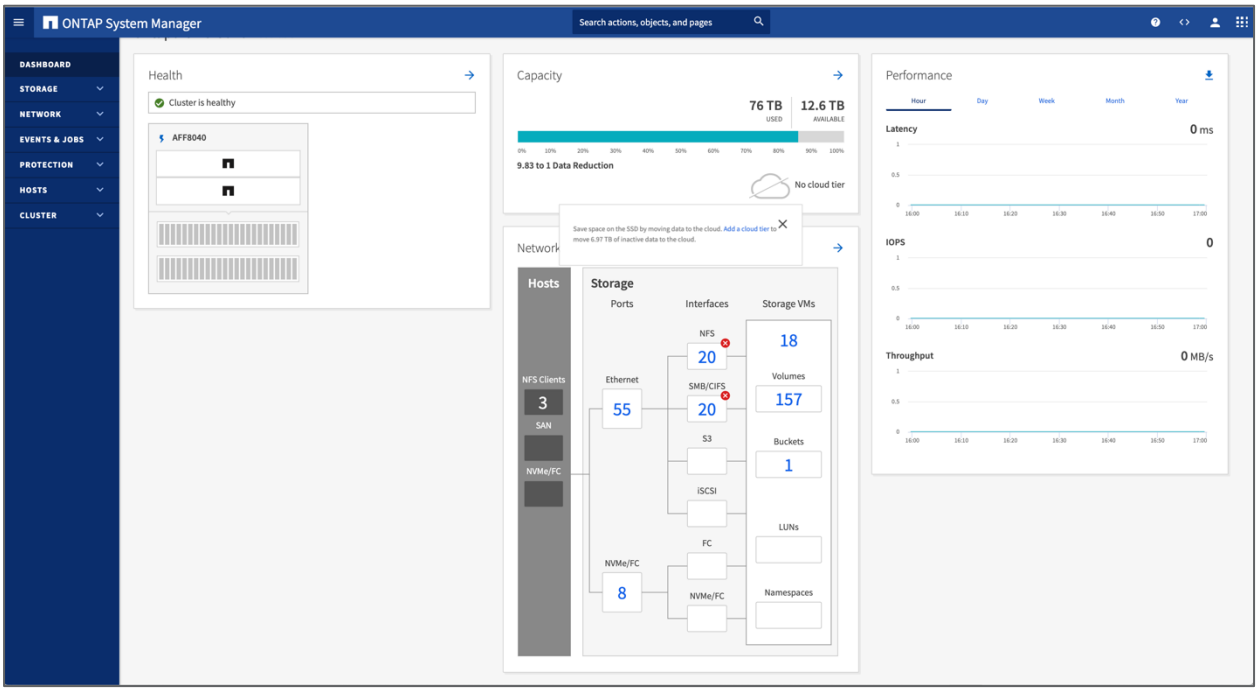
For those who prefer to manage and automate through the use of Windows PowerShell, NetApp offers a powerful set of cmdlets for use with ONTAP. Available through the NetApp communities portal, the ONTAP PowerShell Toolkit is a powerful and versatile means to manage clusters and SVMs alike. For more information, see Making the Most of Data ONTAP PowerShell Toolkit.

## ONTAP System Manager

NetApp ONTAP System Manager is a graphical web-based management tool. It provides GUI-based management of storage systems and storage objects, including the ability to configure data protocols, provision storage objects, and create and manage SVMs.

**Note:** System Manager login is only available to cluster administrators.

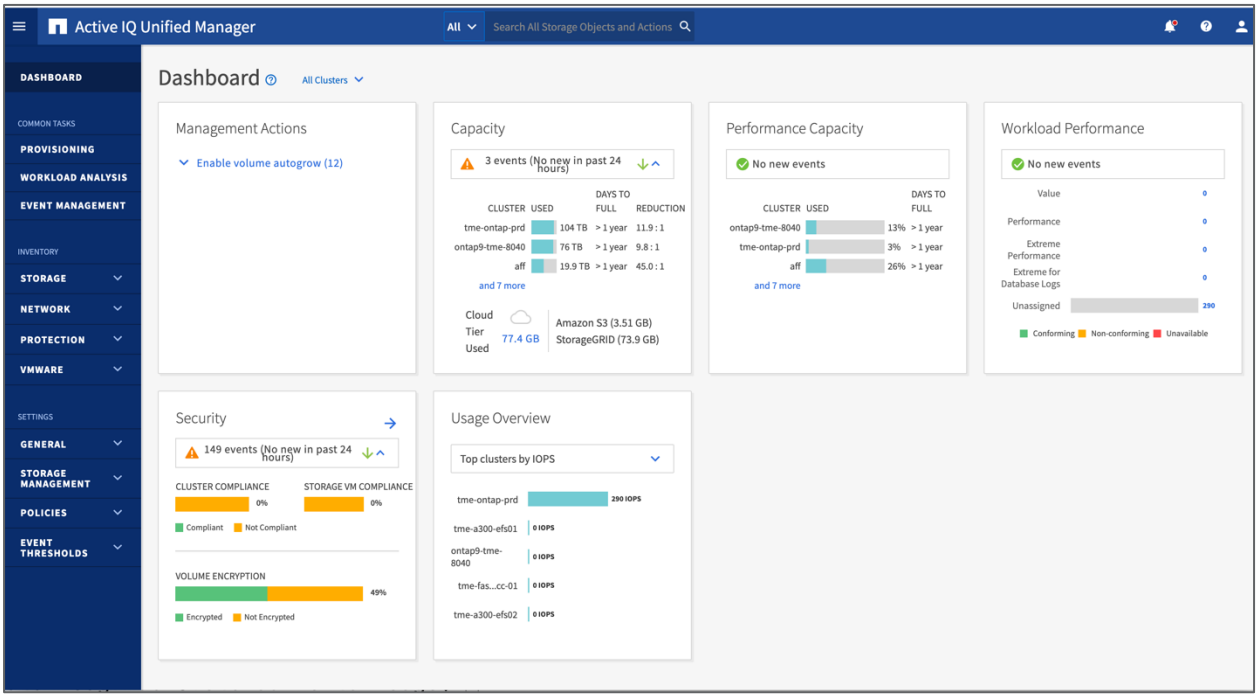
**Figure 10) ONTAP System Manager.**




## Active IQ Unified Manager

Active IQ Unified Manager is a centralized interface for managing multiple clusters across multiple environments. It includes capabilities for monitoring, alerting, and reporting on storage infrastructure at scale. You can find more information about Active IQ Unified Manager on the NetApp documentation site.

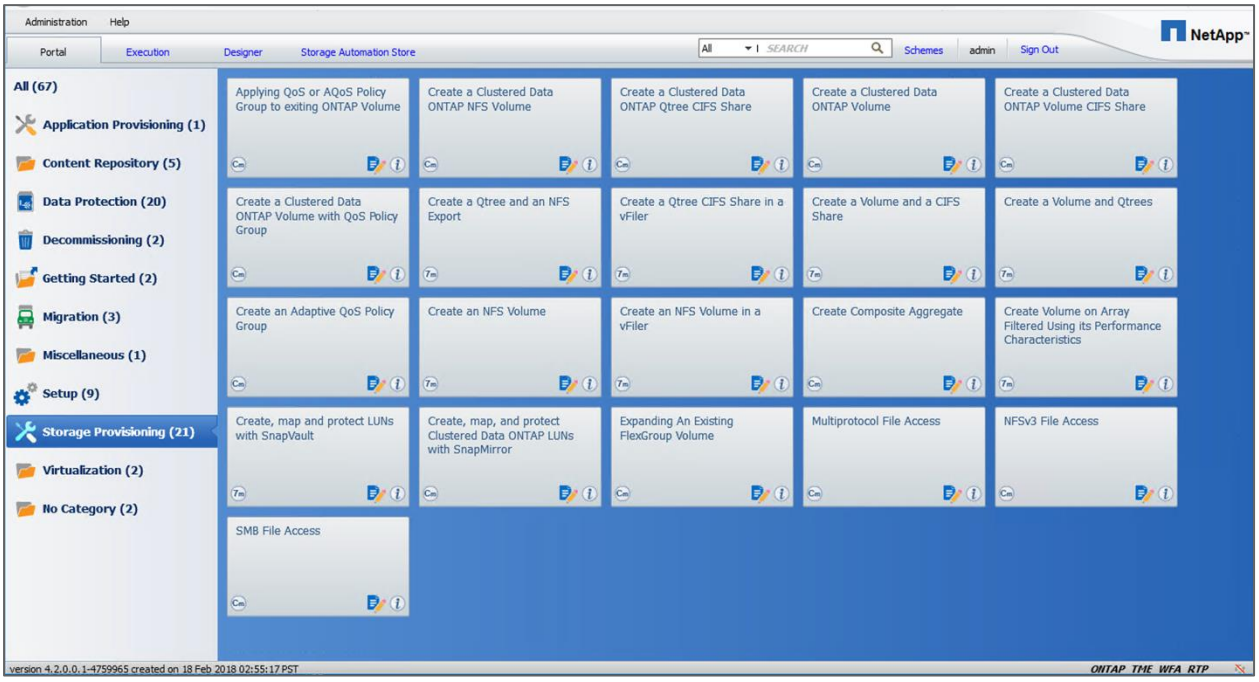
**Figure 11) Active IQ Unified Manager.**




## OnCommand Workflow Automation

OnCommand Workflow Automation (OnCommand WFA) is a powerful tool for storage automation. OnCommand WFA allows storage administrators to create, test, and publish custom workflows for an endless variety of storage functions and tasks. For further information about OnCommand WFA, see the OnCommand Workflow Automation Documentation Resources page.

**Figure 12) OnCommand Workflow Automation portal.**



# Where to find additional information

To learn more about the information described in this document, refer to the following documents and/or websites:

- ONTAP 9 Documentation Center
  http://docs.netapp.com/ontap-9/index.jsp

# Version history

| Version | Date | Document Version History |
|---|---|---|
| Version 1.0 | July 2013 | Pre ONTAP 9 release |
| Version 1.1 | December 2020 | Updated for ONTAP 9 |
| Version 1.2 | January 2021 | IPspaces update |

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.



TR-4160-0121

NetApp