Technical Report

FlexPod Security Hardening with Red Hat OpenShift

Abhinav Singh, NetApp
February 2026 | TR-5019

In partnership with

Abstract

This technical report outlines best practices for securing FlexPod® infrastructure integrated with RedHat® OpenShift® and NetApp® Trident™ software. It provides a comprehensive framework for hardening compute, network, storage, and container orchestration and virtual machine to meet enterprise security and compliance requirements. The guide includes actionable recommendations for identity management, data protection, network segmentation, workload security and audit readiness.

TABLE OF CONTENTS

Solution Overview **5**

Introduction 5

Audience 5

FlexPod overview 5

FlexPod components 7

FlexPod security hardening **14**

FlexPod topology 14

Hardware and software 15

General considerations 15

Network and traffic segmentation 17

Network access restriction 18

Login authentication 18

Role-based access control 27

Login banners 32

Login session timeout and limits 34

Time synchronization 36

Remote logging 39

Configuration backup 43

FIPS 140 compliance 46

UEFI secure boot 50

Image validation 51

NFS access authorization 53

ONTAP data-at-rest encryption 54

ONTAP data-in-flight encryption 55

ONTAP ransomware protection 56

Secure NX-OS with Cisco Live Protect 58

OpenShift overview and installation **60**

OpenShift and Red Hat Linux CoreOS overview 60

OpenShift – installation and configuration 62

Post installation configuration 72

OpenShift network configuration 74

NetApp Trident **75**

Trident overview 75
Best Practice for Trident 75
ONTAP backend drivers 77
Trident Protect 84
OpenShift security hardening 84
Red Hat Enterprise Linux CoreOS Security 84
Identity and access management security 88
Kubernetes security 95
Network security 97
Container security 103
Encryption and secret management 109
Auditing and logging 112
Configuring OpenShift backup 117
Security Profiles Operator 119
Compliance Operator on OpenShift 121
OpenShift virtualization setup and hardening 125
Red Hat OpenShift Virtualization 125
OpenShift Virtualization VM Networking 125
Grouping and separating VMs with namespaces or Projects 127
Storage components 128
Platform configuration 129
Multi-tenant configuration 131
Multitenancy within OpenShift cluster and storage 131
Support Information and Advisory 134
Support information 134
Security advisories 140
Conclusion 147
Where to find additional information 148
Version history 149

LIST OF TABLES

Table 1) NetApp AFF / AFX /ASA / FAS product family systems technical documentation. 13
Table 2 Hardware and Software used in the solution 15

Table 3) Configured VLANs and their usage. 17
Table 4) Supported System-defined roles in Intersight....28
Table 5) Example predefined user roles for Cisco NX-OS software. 29
Table 6) Example predefined roles for ONTAP cluster administrators. 30
Table 7) Predefined roles for ONTAP SVM administrators. 31
Table 8) Supported host key type alogrithms for ONTAP SSH connections 49
Table 9) Hostnames and IP addresses for OpenShift cluster 63
Table 10) Types of users 89
Table 11) Default groups in OpenShift 90
Table 12) Predefined OpenShift roles and description. 91
Table 13) Identity providers available with OpenShift 93

LIST OF FIGURES

Figure 1) FlexPod solution components 5
Figure 2) Cisco UCS C-Series M7 and M8 rack servers, UCS XE9305 and 9508 chassis with X-Series compute nodes 7
Figure 3) Cisco UCS 9108 100G and 9108 25G IFM for the 9508 chassis. 8
Figure 4) Cisco UCS 6th and 5th generation Fabric Interconnects. 8
Figure 5) Cisco UCS VICs for C-Series servers and X-Series compute nodes. 9
Figure 6) UCS management with Intersight 10
Figure 7) Example Cisco Nexus 9K switches. 11
Figure 8) Example Cisco MDS 9100 / 9300 Series switches. 12
Figure 9 13
Figure 10) FlexPod topology with UCS X-Series chassis and AFF A90 storage system 14
Figure 11) vNICs configuration for OpenShift worker nodes 18
Figure 12) UCS Secure Boot Process 50
Figure 13) Node installation workflow 65
Figure 14) OpenShift user, group and role 91
Figure 15) OpenShift network policy 98
Figure 16) Encrypted and non-encrypted traffic 103
Figure 17) Audit in OpenShift 113
Figure 18) OpenShift Virtualization VM attachment to the network 126
Figure 19) VM grouping and separation using namespaces 128
Figure 20) Multi-tenant in OpenShift, NetApp ONTAP and Trident 132

Solution Overview

Introduction

With the rise of high-profile security breaches and ransomware attacks, enterprises face growing risks of operational disruption, financial loss, and theft of sensitive customer data—often resulting in significant reputational damage. As both the frequency and impact of these attacks increase, organizations must not only strengthen the security of their existing systems but also adopt modern platforms and architectures that are secure by design.

FlexPod solutions provide the foundational infrastructure for enterprises and businesses around the world. Today, FlexPod environments not only support traditional workloads but also modern cloudnative platforms such as Red Hat OpenShift, which enables enterprises to securely deploy and operate containers and virtual machines at scale. With OpenShift running on FlexPod, organizations gain a unified, security focused platform for virtualized, containerized, and hybrid cloud workloads. -native platforms such as Red Hat OpenShift, which enables enterprises to securely deploy and operate containers and virtual machines at scale. With OpenShift running on FlexPod, organizations gain a unified, security-focused platform for virtualized, containerized, and hybrid cloud workloads.

Given the critical role FlexPod plays in hosting diverse business applications—from legacy systems to containerized microservices and VM based workloads—implementing security hardening best practices is essential. This technical report provides insights into the tools, capabilities, and integrated security technologies within the FlexPod stack that help organizations protect their data, prevent breaches, and recover quickly from security incidents. The content is designed to help enterprises strengthen the security posture of both their infrastructure and the modern applications that run on it.-based workloads—implementing security hardening best practices is essential. This technical report provides insights into the tools, capabilities, and integrated security technologies within the FlexPod stack that help organizations protect their data, prevent breaches, and recover quickly from security incidents. The content is designed to help enterprises strengthen the security posture of both their infrastructure and the modern applications that run on it.

Audience

This document is intended for a broad range of technical and strategic stakeholders, including IT architects, security practitioners, Kubernetes and platform administrators, sales engineers, field consultants, professional services teams, IT managers, partner engineering, and customers seeking to leverage a secure, efficient, and innovation ready infrastructure ready.-ready

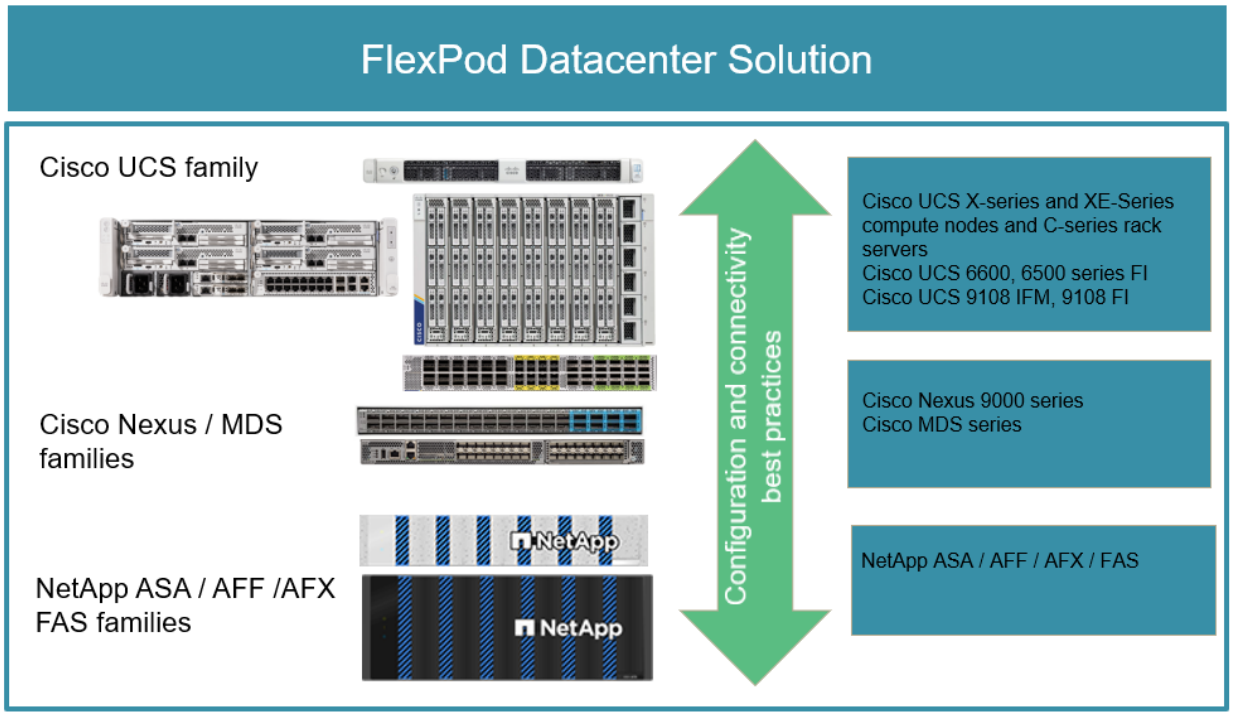
FlexPod overview

FlexPod is the best practice converged infrastructure data center architecture that includes the following components from Cisco® and NetApp:

- Cisco Unified Computing System (Cisco UCS)
- Cisco Nexus and MDS families of switches
- NetApp Fabric Attached Storage (FAS), All Flash FAS (AFF), AFX, All SAN Array (ASA) systems

Shown in **Error! Reference source not found.** are some of the components utilized for creating FlexPod solutions. These components are connected and configured according to the best practices of both Cisco and NetApp to provide an ideal platform for running a variety of enterprise workloads with confidence.

Figure 1) FlexPod solution components



A comprehensive portfolio of Cisco Validated Designs (CVDs) and NetApp Verified Architectures (NVAs) is available to support FlexPod deployments. Developed through ongoing collaboration and innovation between NetApp and Cisco, these designs cover a wide range of data center workloads. Each CVD and NVA undergoes extensive testing and validation to ensure reliability, providing detailed reference architectures and step-by-step deployment guidance that help customers and partners confidently design, deploy, and adopt FlexPod solutions.

By leveraging these proven design guides, organizations can minimize risk, reduce downtime, and enhance the availability, scalability, flexibility, and security of their FlexPod environments. The FlexPod architecture incorporates modular technologies across Cisco UCS compute, Cisco Nexus and MDS networking, and NetApp storage, each offering multiple platform and resource options.

FlexPod benefits

The FlexPod Datacenter solution provides several strategic customer advantages:

- **High availability** across all infrastructure layers, eliminating single points of failure.
- **Scalable design** that enables independent expansion of compute, storage performance and capacity, or network bandwidth.
- **Hybrid-cloud readiness** with a policy-driven, modular architecture that grows with evolving business needs.
- **Flexible infrastructure options** that support components beyond validated designs through Cisco and NetApp interoperability matrix guidance.
- **Integrated ecosystem support**, including monitoring, automation, orchestration, and workload optimization.
- **Cooperative support model**, enhanced by Cisco Solution Support, for streamlined issue resolution.

FlexPod components

Cisco compute components

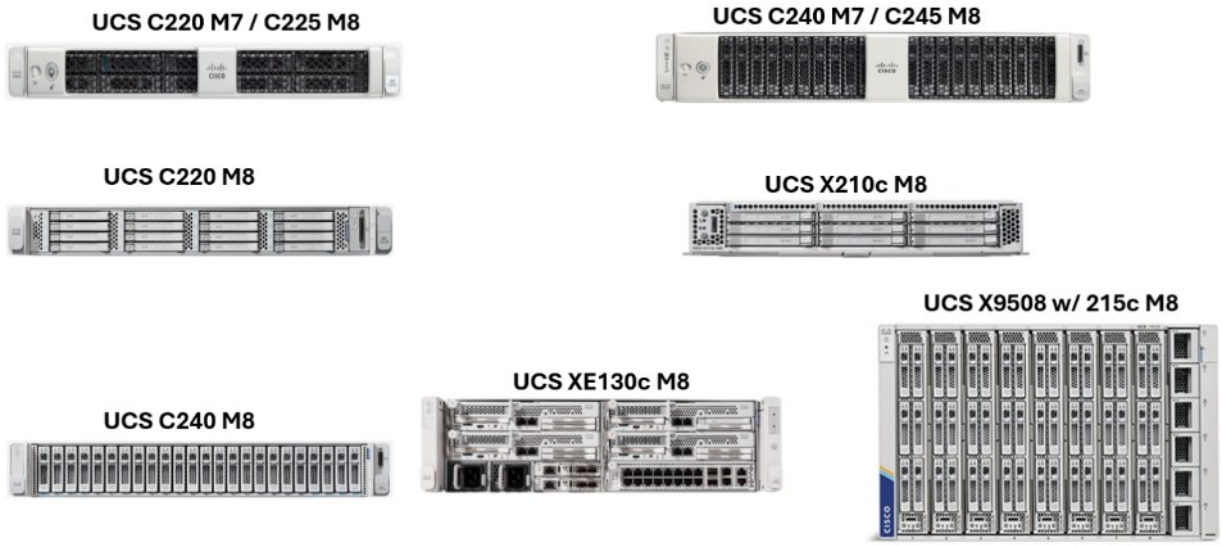
The Cisco Unified Computing System (UCS) is an integrated computing infrastructure to provide unified computing resources, unified fabric, and unified management. It enables companies to automate and accelerate deployment of applications, including virtualization and bare-metal workloads.

The Cisco UCS supports a wide range of deployment use cases including remote and branch locations, data centers, and hybrid cloud use cases. Depending on the specific solution requirements, the FlexPod implementation can utilize a variety of Cisco compute components and at different scales. The following subsections provide additional information on some of the UCS components.

UCS server / compute node

Shown in Figure 2 are some examples of the UCS server components, including UCS C-Series rack servers, UCS X9508 chassis with X-Series compute nodes and the new UCS XE9305 chassis with XE-Series compute nodes. The Cisco UCS C-Series rack servers are available in one and two rack-unit (RU) form factors, Intel and AMD CPU based models, and with various CPU speed/cores, memory, and I/O options. The Cisco X-Series and XE Series compute nodes are also available with various CPU, memory, and I/O options and they are all supported in the FlexPod architecture to meet the diverse business requirements.

Figure 2) Cisco UCS C-Series M7 and M8 rack servers, UCS XE9305 and 9508 chassis with X-Series compute nodes.



In addition to the latest generation C220 / C225 / C240 M8 rack servers, C240 M7 rack server, X210c / X215c M8 and XE130c M8 compute nodes shown in the figure, prior generation rack server and blade server variants can also be utilized while they are still supported.

Intelligent Fabric Module

Intelligent Fabric Module (IFM) provide unified fabric connectivity for the Cisco UCS X9508 X-Series chassis.

The UCSX 9108 100G IFM has eight 100-G unified Ethernet ports for connecting the blade servers in the UCS X9508 chassis with FIs. Each 9108 has one 100-G connection, or four 25-G lanes (depending on the VIC installed in the server), towards each UCS X210c compute node in the X9108 chassis. The

UCSX-9108 25G I/O Fabric Module (IFM) provides eight 25-Gb unified Ethernet uplinks that connect the blade servers housed in the UCS X9508 chassis to the Fabric Interconnects (FIs). Each UCSX-9108 IFM delivers a 25-Gb connection—implemented as a single 25-G lane or aggregated lanes depending on the VIC installed in each server. The 9108 IFM also works in concert with the FI to manage the chassis environment.

Shown in Figure 3 are the Cisco UCS 9108 25G and 100G IFMs for the X9508 chassis.

Figure 3) Cisco UCS 9108 100G and 9108 25G IFM for the 9508 chassis.

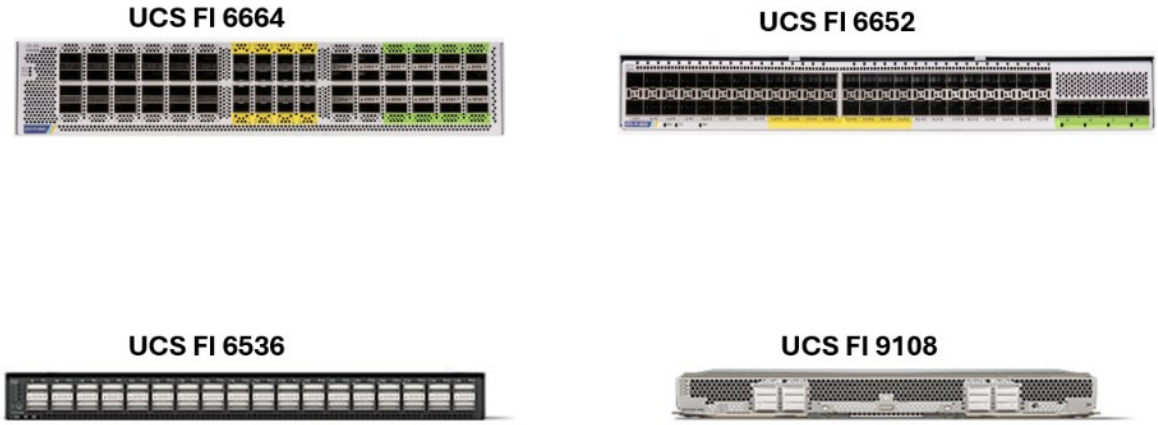
UCS Fabric Interconnects

The Cisco UCS Fabric Interconnects (FIs) provide connectivity and management for the entire Cisco UCS. Typically deployed as an active/active pair, the system's FIs integrates all components into a single, highly available management domain controlled by the Cisco UCS Manager or Cisco Intersight. Cisco UCS FIs provide a single unified fabric for the system, with low-latency, lossless, cut-through switching that supports LAN, SAN, and management traffic using a single set of cables.

The latest 6th-generation Cisco UCS Fabric Interconnects (UCS 6652 and UCS 6664) support line-rate, low-latency, lossless 10/25/40/50/100/400-Gbps Ethernet, and 16/32/64-Gbps Fibre Channel, in addition to NVMe-over-Fabric and FCoE capabilities.

The 5th generation Cisco UCS FI 6536 supports 100/40/25/10 Gbps Ethernet ports and 32Gbps Fibre Channel ports using breakout cables. Figure 4) Cisco UCS 6th and 5th generation Fabric Interconnects. shows the Cisco UCS 6th and 5th generation of Fabric Interconnects.Figure 4) Cisco UCS 6th and 5th generation Fabric Interconnects.

Figure 4) Cisco UCS 6th and 5th generation Fabric Interconnects.



Edge Chassis Management Controller (eCMC)

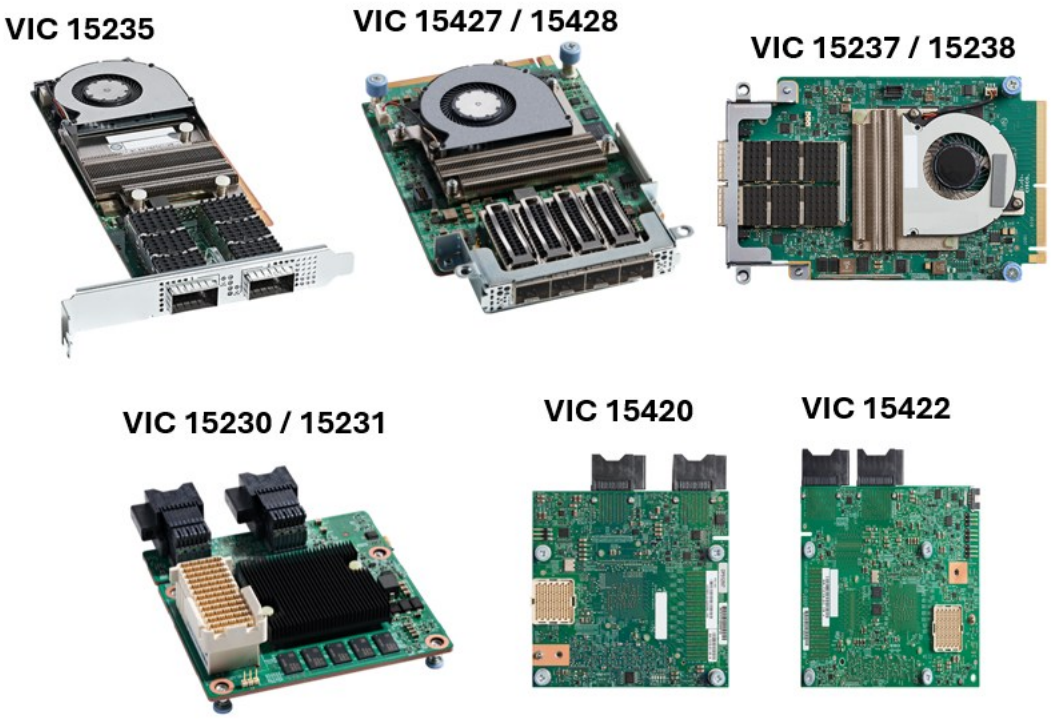
Cisco Unified Edge (Cisco UCS XE9305) is an all-in-one chassis that seamlessly integrates compute, networking, storage, and security capabilities into a unified, high-performance platform featuring compute nodes with Intel Xeon 6 processors. The Cisco Unified Edge solution features a specialized management component known as the Cisco Edge Chassis Management Controller (eCMC), designed specifically for the Cisco UCS XE9305 Chassis.

Cisco Edge Chassis Management Controllers (eCMCs) form a unified fabric that provides connectivity between all nodes within the chassis and with upstream networks and provides local chassis management and a secure control plane connection with Cisco Intersight. For more information, see Cisco UCS XE9305 Chassis Data Sheet.

UCS Virtual Interface Cards

Cisco UCS Virtual Interface Cards (VICs) unify system management, LAN, and SAN connectivity for rack and blade servers. It supports up to 512 virtual devices, either virtual Network Interface Cards (vNICs) or virtual Host Bus Adapters (vHBAs) using the Cisco SingleConnect technology. As a result of virtualization, the VIC cards greatly simplify the network connectivity and reduces the number of network adapters, cables, and switch ports needed for solution deployment. Shown in Figure 5 are some of the Cisco UCS VICs available for the C-Series servers and the X-Series compute nodes.

Figure 5) Cisco UCS VICs for C-Series servers and X-Series compute nodes.



The different adapter models support different blade and rack servers with a variety of port count, port speed, and form factors of modular LAN on Motherboard (mLOM), mezzanine card, and PCIe interface. The adapters can support some combinations of 10/25/40/100/200-G Ethernet and Fibre Channel over Ethernet (FCoE). They incorporate Cisco's Converged Network Adapter (CNA) technology and support a comprehensive feature set and simplify adapter management and application deployment. With a combination of Cisco VIC in mLOM, mezzanine, and port expander / bridge card configurations, you can fully take advantage of the bandwidth and connectivity available to the blade and rack mount servers.

Note: VIC is supported on the B-Series, C-Series, and X-Series platforms, but not on the XE-Series.

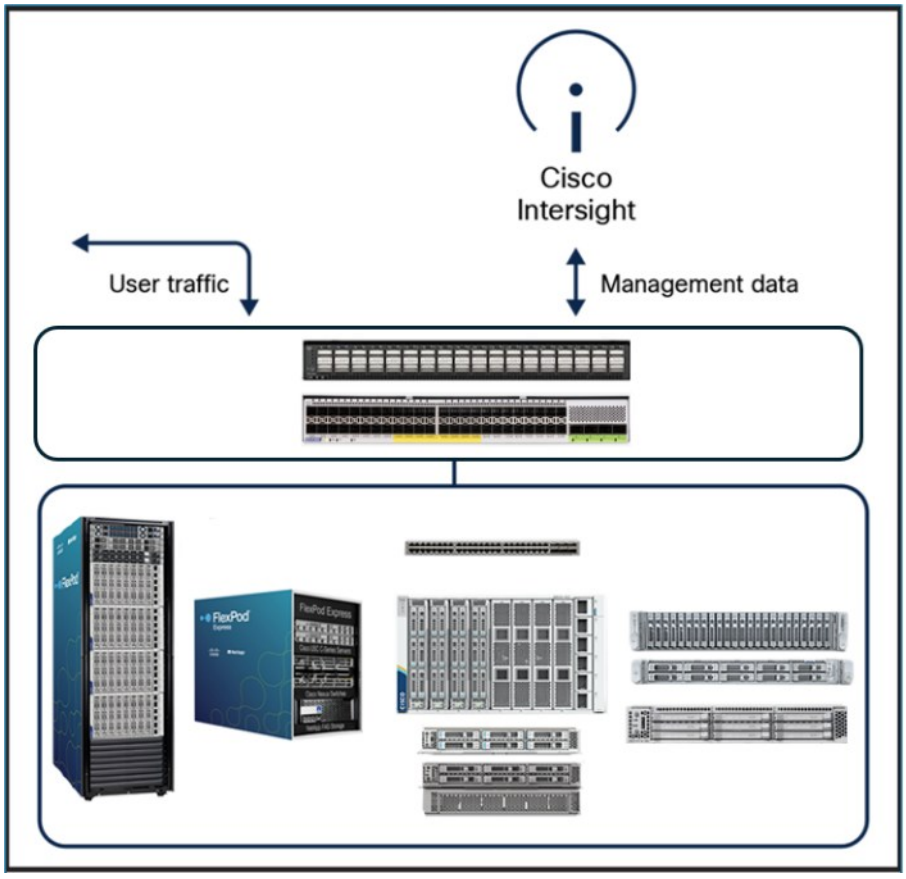
For details on the Cisco UCS product families, technical specifications, and documentation, please refer to the Cisco UCS web site for information.

Cisco UCS management options

Cisco UCS management options include the native, embedded Cisco UCS Manager (UCSM) for on-premise, fabric-interconnect-based management, and Cisco Intersight for cloud-based, SaaS or appliance driven management, which allows for either native or Intersight Managed Mode (IMM). Both provide centralized, policy-driven control of compute, network, and storage, including firmware updates, service profiles, and diagnostics.

A Cisco UCS compute system is available in many blade or rack-mount configurations. With Intersight, systems that come with Fabric Interconnects (FIs) are configured during build time to run with Intersight's cloud-management services (Intersight Managed Mode, IMM). Systems without FIs will run in standalone mode and can be claimed by Intersight using a two-factor authentication mechanism that requires access to the system. Figure 6) UCS management with Intersightshows the user and management traffic separation in Intersight.

Figure 6) UCS management with Intersight



Note: The management mode of the Unified Edge is always set to Intersight, meaning the device is fully managed through the Cisco Intersight platform. Other management modes (such as Standalone or UCS Manager) are not applicable to Unified Edge

Connecting to Intersight

There are three methods of running Intersight Managed Mode (IMM). These depend on the type of deployment the end user needs. For example, if there is a requirement for an air-gapped environment, but IMM is needed, an on-premises version of Intersight can be used.

Cisco Intersight also provides an API that enables automation with platforms such as Ansible and Terraform, allowing automated workflows to be used across all three Intersight deployment models.

Here are the types of Intersight deployments:

- Cloud-based Cisco UCS management (Intersight SaaS)
  - This cloud-native approach provides a centralized, web-based interface accessible from anywhere. It simplifies IT management by eliminating the need for on-premises hardware and software, making it an ideal choice for organizations seeking agility, scalability, and easy access to the latest features.

  Note: This solution covers Intersight SaaS.
- Connected virtual appliance (Intersight CVA)
  - For businesses that prefer an on-premises solution while still benefiting from cloud connectivity, the connected virtual appliance is the answer. It offers the flexibility to run the Intersight virtual appliance within the data center while maintaining seamless connections to the Intersight cloud for updates and technical support.
- Private virtual appliance (Intersight PVA)
  - If security and isolation are important factors, the private virtual appliance delivers an on-premises, air-gapped option. It operates in complete isolation from the Intersight cloud and the internet, ensuring that the infrastructure remains secure while still taking advantage of Intersight's management capabilities.

Cisco switching components

Nexus switches

The Cisco Nexus 9000 Series Switches offer both modular and fixed 1/10/25/40/100 Gigabit Ethernet switch configurations with scalability up to 60 Tbps of nonblocking performance with less than five-microsecond latency, wire speed VXLAN gateway, bridging, and routing support.

FlexPod utilizes Cisco Nexus Series switches to provide Ethernet switching fabric for communications between the Cisco UCS and NetApp storage controllers. When selecting a switch model for FlexPod deployment, there are many factors to consider, such as performance, port speed, port density, switching latency, and technology such as ACI and VXLAN support, for your design objectives as well as the switches' support timespan.

The validation for many recent FlexPod CVDs utilizes the Cisco Nexus 9000 series switches, such as the Nexus 93600CD-GX which deliver high performance 10/25/40/50/100/200/400 ports, low latency, and exceptional power efficiency in a compact 1U form factor. Figure 7 shows a few Cisco Nexus 9k switches

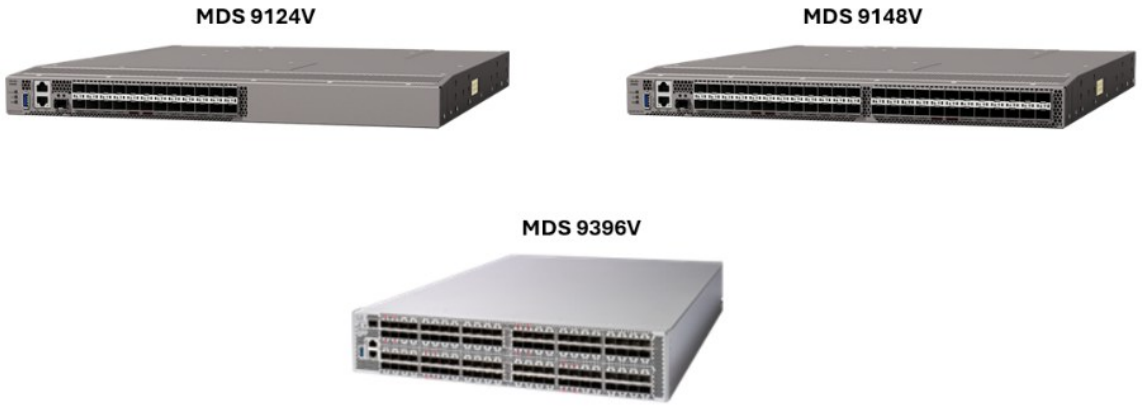
Figure 7) Example Cisco Nexus 9K switches.

Please refer to Cisco Data Center Switches for more information on the available Nexus switches and their specifications and documentation.

MDS switches

The Cisco MDS 9000 Series Fabric switches are an optional component in the FlexPod architecture. These switches are highly reliable, highly flexible, secure, and can provide visibility into the traffic flow in the fabric. Figure 8 shows some examples of fixed MDS switches that can be utilized to build redundant FC SAN fabrics for a FlexPod solution to meet application and business requirements.

Figure 8) Example Cisco MDS 9100 / 9300 Series switches.



The latest Cisco MDS 9396V, MDS 9148V, and MDS 9124V 64-Gbps Fibre Channel switches deliver state-of-the-art performance with high-bandwidth, low-latency SAN connectivity purpose-built for modern all-flash and NVMe/FC storage environments. The Cisco MDS 9396V, introduced as one of the newest 64G 2RU fabric switches, provides up to ninety-six line-rate Fibre Channel ports, offering exceptional scalability and high-density deployment options for enterprise data centers.

Complementing this, the next-generation MDS 9148V and MDS 9124V switches bring 48-port and 24-port 64G Fibre Channel capabilities, respectively, with support for 8/16/24/32/64G FC speeds, enabling flexible, high-performance connectivity for all-flash arrays and latency-sensitive workloads. These new hardware platforms also incorporate advanced SAN analytics and real-time telemetry—features designed to enhance visibility and ensure proactive operations—allowing frame-header inspection and telemetry streaming to analytics platforms such as Nexus Dashboard (ND) further strengthening operational insight and troubleshooting workflows. Together, these latest MDS switches form a reliable, scalable, and forward-looking SAN infrastructure that maintains full interoperability within the broader Cisco MDS 9000 family, ensuring seamless integration into existing environments while supporting next-generation storage architectures.

Please refer to Cisco MDS Fabric Switches for more information on the available MDS Fabric switches and see the NetApp IMT and Cisco Hardware and Software Compatibility List for a complete list of supported SAN switches.

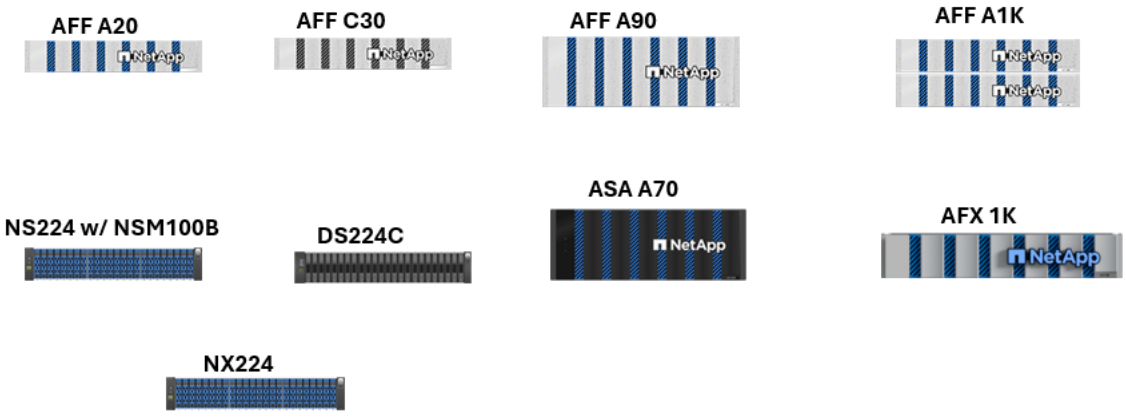
NetApp storage components

Redundant NetApp Fabric Attached Storage (FAS), All Flash FAS (AFF), All SAN Array (ASA), and **AFX** systems running the latest ONTAP® software releases are recommended for creating FlexPod solutions. Running the latest ONTAP release ensures access to ongoing ONTAP innovations, performance enhancements, quality improvements, bug fixes, and security vulnerability patches.

NetApp storage components for FlexPod solutions now focus on supported platforms running the latest ONTAP® software. Current offerings include **NetApp FAS50, FAS70, and FAS90** for balanced performance and capacity; **AFF A-Series systems** (A20, A30, A50, A70, A90, and A1K) and **ASA A-Series systems** delivering industry-leading performance with full NVMe support; **AFF C-Series and ASA C-Series** featuring high-density QLC flash optimized for capacity-centric workloads; **AFX platforms** delivers all the performance benefits of parallel file systems with an disaggregated architecture; and **ASA platforms** offering SAN-optimized, symmetric active-active architecture with a 99.9999% availability SLA. These systems deliver seamless NetApp Data Fabric capabilities—from edge to core to cloud—enhanced by ONTAP innovations, cyber-resilience capabilities (including the Ransomware Recovery Guarantee), and cost-effective scalability for modern data centers.

NetApp offers a variety of storage systems and disk shelves to meet your performance and capacity requirements. NetApp storage systems and NetApp Ransomware Recovery Guarantee offer a cost-effective approach to cyber resilience with more protection and security. Please see Figure 9 for some example systems from the NetApp storage system portfolio. Please see Table 1) NetApp AFF / AFX /ASA / FAS product family systems technical documentation. for links to product pages for detailed information about NetApp AFF, ASA, AFX and FAS storage systems' capabilities and specifications.

Figure 9



Note: The above picture only shows a small subset of the available NetApp storage controllers.

Table 1) NetApp AFF / AFX /ASA / FAS product family systems technical documentation.

Product family	Technical documentation
AFF A-series systems	AFF A-series documentation
AFF C-series systems	AFF C-series documentation
AFX system	AFX system documentation
ASA A-series systems	ASA A-series documentation
ASA C-series systems	ASA C-series documentation
FAS systems	FAS systems documentation

Note: Please consult the NetApp disk shelves and storage media documentation and NetApp Hardware Universe for details on the disk shelves and the supported disk shelves for each storage controller model.

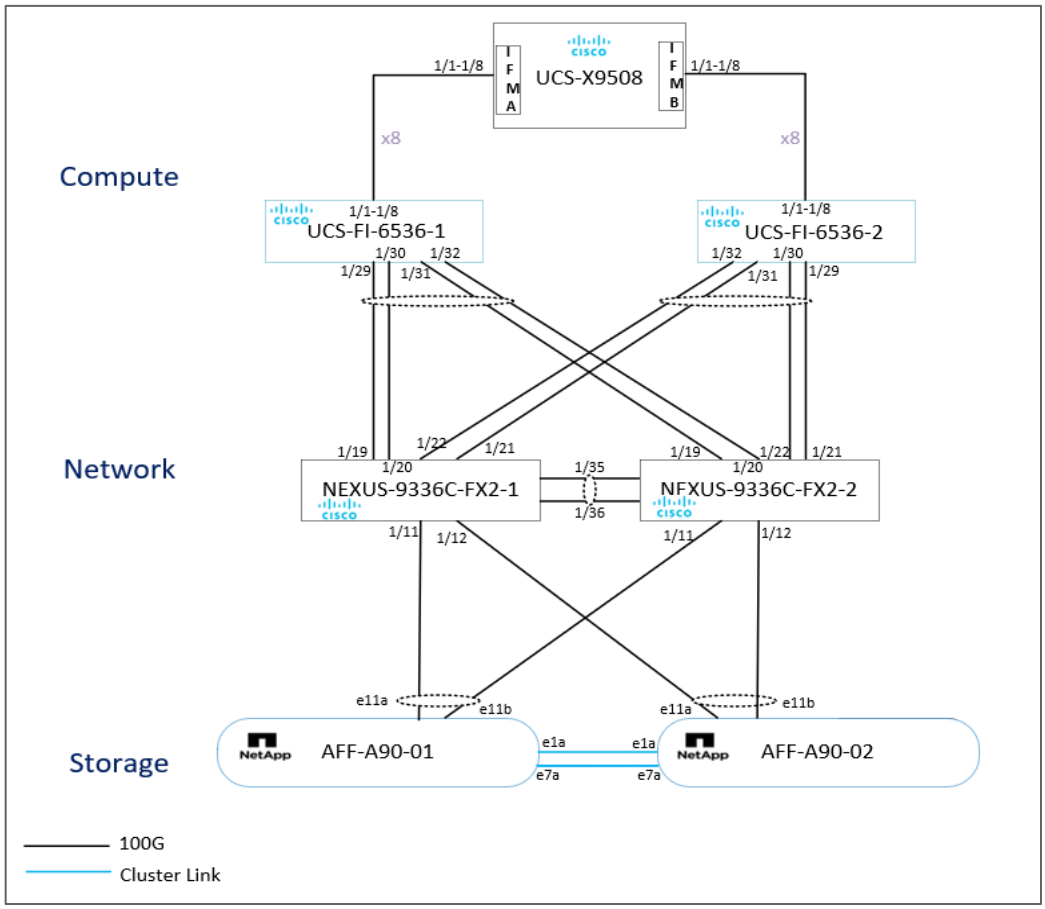
FlexPod security hardening

Typical FlexPod solutions include compute, network, storage, and virtualization layers. Therefore, we examined the various aspects of the solutions and aimed to highlight and bring awareness to the available tools and technologies for securely implementing your business solutions. In some cases, we also provide concrete examples for you to adopt or provide guidance for you to consider. Some of the discussion topics span all layers of the component, while others might be specific features for achieving certain security objectives. Additional resources and information from NetApp, Cisco, and Red Hat should also be reviewed and adopted as appropriate to continually improve your overall security posture.

FlexPod topology

For the validation of the FlexPod security hardening solution, supported technology components from NetApp, Cisco, and Red Hat are utilized. The solution features latest NetApp AFF A90 single HA pair running ONTAP 9.17.1, a pair of Cisco Nexus 9336C-FX2 switches, a pair of Cisco UCS 6536 FIs, and four Cisco UCS X210c M7 servers and 1 UCS X210cM6 server running Red Hat Enterprise Linux CoreOS (RHCOS). Figure 10) FlexPod topology with UCS X-Series chassis and AFF A90 storage system shows the FlexPod topology used in this solution.

Figure 10) FlexPod topology with UCS X-Series chassis and AFF A90 storage system



The reference hardware configuration includes:

- Two Cisco Nexus 9336C-FX2 Switches in Cisco NX-OS mode provide the switching fabric.

- Two Cisco UCS 6536 Fabric Interconnects (FI) provide chassis connectivity. Two 100 Gigabit Ethernet ports from each FI, configured as a Port-Channel, are connected to each Nexus 9336C-FX2.
- One Cisco UCS X9508 Chassis connects to fabric interconnects using Cisco UCS UCSX-I-9108-100G IFMs, where eight 100 Gigabit Ethernet ports are used on each IOM to connect to the appropriate FI.
- One NetApp AFF A90 HA pair connects to the Cisco Nexus 9336C-FX2 Switches using two 100 GE ports from each controller configured as a Port-Channel.

This reference configuration consists of 6 Cisco UCS X210c M7 servers and 1 additional Cisco UCS X210c M6 server.

Hardware and software

Table 2 Hardware and Software used in the solutionlists the hardware and software used for the solution validation. It is important to note that Cisco, NetApp, and Red Hat have interoperability matrixes that should be consulted to determine support for any specific implementation of FlexPod. Click the following links for more information:

- NetApp Interoperability Matrix Tool: http://support.netapp.com/matrix/
- Cisco UCS Hardware and Software Interoperability Tool: http://www.cisco.com/web/techdoc/ucs/interoperability/matrix/matrix.html
- Red Hat supported hardware Guide: https://catalog.redhat.com/en/hardware

Table 2 Hardware and Software used in the solution

Category	Component	Software version
Compute	Cisco UCS Fabric Interconnect 6536	6.0(1.250198)
	Cisco UCS X210c M7 servers	6.0(1.250120)
	Cisco UCS X210c M6 servers	6.0(1.250120)
	Cisco UCSX-I-9108-100G	6.0(1b)
	Cisco VIC 15231 (PID: UCSX-ML-V5D200G)	5.4(1.30)
	Cisco VIC 15231 ENIC Driver	5.14.0-570.45.1.el9_6.x86_64 enic
Network	Cisco Nexus 9336C-FX2	10.6.1(F)
Storage	NetApp AFF A90	9.17.1
	NetApp System Manager	9.17.1
Software	RedHat OpenShift	4.19.13
	Red Hat OpenShift Virtualization	4.19.15
	RHCOS	9.6
	Kubernetes	v1.32.8
	NetApp Trident	25.10.0

General considerations

Management plane, control plane, and data plane

When implementing a security hardening strategy for your FlexPod solution, it is important to understand what is being protected. This can typically be broken down into three areas: management plane, control plane, and data plane.

- The management plane contains the traffic that supports provisioning, maintenance, and monitoring functions for the device. Example - HTTP/HTTPS, SSH, Simple Network Management Protocol (SNMP), Syslog, DNS, etc. This also includes management access to all the FlexPod components.
- The control plane refers to the aspects that affect the network and communication such as switching, signaling, Link Layer Discovery Protocol (LLDP), Fiber Channel over Ethernet (FCoE), Internet Small Computer Systems Interface (iSCSI), Address Resolution Protocol (ARP), and Layer 2 keepalive.
- The data plane refers to the actual information, such as the IO data between the RHCOS hosts and ONTAP storage using iSCSI or NFS protocols, and the data stored on the physical storage shelves within the FlexPod system.

Protection at the management plane layer can be provided by hardening strategies such as limiting access to the devices, implementing login authentication and role-based access control that is appropriate for the solution. Protection at the control plane layer can involve utilizing appropriate security features for authentication and access and implementing access control list. On the other hand, protection at the data plane layer can be provided by strategies such as encrypting data in transit and at rest and ensuring that the cryptographic modules in use meet the security standards and requirements.

Bastion Host

A bastion host, or a jump host, is typically a virtual machine which is hardened to provide secure access to your private FlexPod solution environment. You might implement SSH based access, or Remote Desktop Protocol (RDP) based access to the bastion host. Once you are on the bastion host, you can access your FlexPod solution environment.

Internet proxy server

Instead of giving the devices in the FlexPod solution direct internet access, which can open them up for cyber-attacks from the internet, you can place an internet proxy server between your FlexPod solution and the internet.

Disable unused services

As a best security practice, administrators should disable any services that are not required for their environment. Cisco Intersight-managed infrastructure, Cisco Nexus switches running NX-OS, and NetApp ONTAP all ship with most nonessential services disabled by default. Any service that needs to be enabled must be explicitly configured by the administrator through the appropriate Intersight policies, NX-OS configuration commands, or ONTAP administrative settings.

Use secure protocols

FlexPod solution management communications can contain sensitive data. As a result, secure protocols should be used whenever possible. Examples of choosing secure protocols include the use of SSH instead of Telnet and the use of HTTPS instead of HTTP so that both authentication data and management information are encrypted.

To take a step further, disable unsecure protocols after verifying that a secure alternative is available and accessible. For example, you can confirm if Telnet service is disabled on a device by trying to telnet into the device and getting refused for the connection attempt.

```
[admin@sec-rhel-9 ~]$ telnet fpsa-9336-u1419
Trying 172.22.14.253...
telnet: connect to address 172.22.14.253: Connection refused
```

Network and traffic segmentation

VLANs

A VLAN is a logical segment in a switched network by function or application. Each VLAN is considered a logical network, and packets must be forwarded through a router for destinations that do not belong to the VLAN.

In FlexPod solutions, VLANs are utilized for network and traffic segmentations. Depending on the specific FlexPod solution, there are various VLANs being utilized, including out-of-band and in-band management networks, NFS, iSCSI protocol networks, and VM traffic. Table 3 lists VLANs configured.

Table 3) Configured VLANs and their usage.

Name	VLAN ID	Usage
Native-VLAN	2	VLAN 2 used as native VLAN instead of default VLAN (1)
OOB-MGMT-VLAN	2214	Out-of-band management VLAN for devices
OCP-MGMT-VLAN	178	Management VLAN for OCP Nodes
NFS-VLAN	2216	NFS VLAN persistent storage
iSCSI-A-VLAN	411	iSCSI-A path for persistent storage
iSCSI-B-VLAN	412	iSCSI-B path for persistent storage
OCP-VM-VLAN	177	VLAN for virtual machine front end interfaces

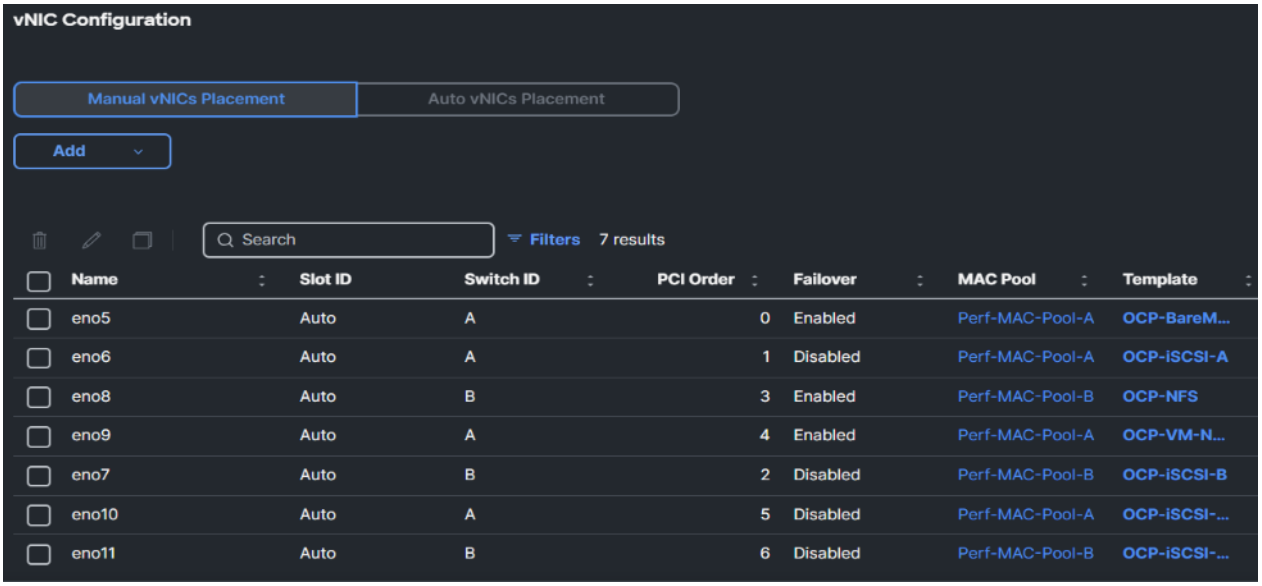
Some of the key highlights of VLAN usage are as follows:

- OOB-MGMT-VLAN allows customers to manage and access out-of-band management interfaces of various devices and is brought into the infrastructure to allow access to the Cisco UCS servers, Cisco Nexus switches, and NetApp ONTAP storage cluster.
- OCP-MGMT-VLAN is used for in-band management of OpenShift nodes, API and ingress services.
- NFS-VLAN provides persistent volumes to containers.
- A pair of iSCSI VLANs (iSCSI-A-VLAN and iSCSI-B-VLAN) are also configured to provide persistent volumes to containers.
- OCP-VM-VLAN provides connectivity to the front-end interfaces of virtual machines running on OpenShift Virtualization.

Note: If your deployment includes NVMe/TCP protocol, please refer to FlexPod Datacenter with Red Hat OpenShift Bare Metal Manual Configuration with Cisco UCS X-Series Direct CVD which utilize that protocol for additional details.

The vNICs were configured and mapped to vNIC templates in Cisco Intersight for the OpenShift worker nodes. The required VLANs were assigned to the appropriate vNICs to support both OpenShift Bare Metal and OpenShift Virtualization. Figure 11) vNICs configuration for OpenShift worker nodes illustrates the vNIC configuration on the OpenShift worker nodes.

Figure 11) vNICs configuration for OpenShift worker nodes



ONTAP IPspaces

You can use an IPspace to create a distinct IP address space for each SVM in an ONTAP storage cluster. IPspace enables clients in administratively separate network domains to access cluster data while using overlapping IP addresses from the same IP address subnet range. IP addresses defined for an IPspace are applicable only within that IPspace. A distinct routing table is maintained for each IPspace and no cross-IPspace traffic routing happens. To utilize IPspaces in ONTAP, the traffic from different clients with overlapping IP addresses will need proper networking and computer layer separations.

Network access restriction

Beyond segregating network traffic and access by function, organizations can further restrict a network segment (such as a specific VLAN) by permitting access only to authorized devices

Cisco Nexus IP ACL configuration

On Cisco Nexus switches, you can configure and apply Access Control List (ACL) to filter traffic. Nexus switches support ACLs based on IPs (IPv4 and IPv6) or MAC addresses. An ACL is an ordered set of rules, where each rule specifies a set of conditions for a packet to match. The first matching rule determines whether the packet is permitted or denied. When there is no match, the applicable implicit rule is applied. The implicit rule ensures that the switch denies unmatched IP or MAC traffic.

IPv4, IPv6, and MAC ACLs also allow you to identify traffic by protocol. You can specify any protocol by number and some also by name. The created ACL filters and rules can be applied to interfaces such as ports, port channels, and VLANs. For more information on configuring IP ACL and VLAN ACL, refer to Configuring IP ACLs and Configuring VLAN ACLs documentation.

Login authentication

Password strengths and policies

Using a strong password to access and secure your FlexPod components is very important. The following is a list of guidelines for creating a strong password.

- Use a password which is at least eight characters long.
- Use a password which contains lower case letters, upper case letters, digits, and special characters.
- Use a password which does not contain many consecutive characters or numbers.
- Use a password which does not contain many repeating characters or numbers.
- Use a password which does not contain dictionary words or proper names.
- Use a password which is not identical to the username or the reverse of the username.
- Check documentation for the specific password characters which are not allowed by the specific FlexPod components.

In addition to using a strong password, it is a best practice to change passwords regularly and to use password policies to enforce the required password changes. Please consult documentation for details on implementing password policies.

Lightweight Directory Access Protocol authentication

The Lightweight Directory Access Protocol (LDAP) provides centralized validation of users attempting to gain access to a device. For a FlexPod environment with a lot of users, using LDAP is recommended to simplify and centralize login authentication. In addition to authentication, LDAP also supports authorization for role-based access control.

Cisco Intersight login authentication

Cisco Intersight login authentication is based on secure identity management and role-based access control.

Cisco Intersight uses secure identity management, modern cloud-based authentication mechanisms, and Role-Based Access Control (RBAC) to control user access to managed infrastructure. Authentication in Intersight is integrated with cloud-native identity models and, unlike UCS Manager, does not require on-prem AAA provider configuration.

User Management

Intersight provides granular RBAC by assigning roles and permissions to users or user groups. Roles define the operations a user can perform, and access can be scoped further using Organizations and Resource Groups, enabling fine-grained segmentation of managed assets.

Users in Intersight are authenticated through Cisco Intersight accounts, SSO providers, or external identity providers, depending on the customer setup.

Remote / External Authentication

Unlike UCSM (which requires explicit RADIUS/TACACS/LDAP provider setup), Cisco Intersight integrates remote authentication through Single Sign-On (SSO) solutions. Supported SSO mechanisms include:

- SAML 2.0 identity providers (IdPs) such as Azure AD, Okta, Ping, etc.
- Cisco Secure Access / Duo, enabling MFA for cloud authentication.

Intersight *does not* require configuring local AAA providers like RADIUS or TACACS+ within the Intersight interface. These are instead configured on the upstream SSO/IdP.

(This source verifies Cisco cloud products commonly integrate with SAML, OAuth2, and modern IdPs used by other Cisco platforms.) [cisco.com]

Two-Factor / Multi-Factor Authentication

Multi-factor authentication (MFA) is provided through:

- SSO provider policies

- Cisco Duo MFA
- Cloud identity providers with MFA frameworks

Because Intersight offloads authentication to IdPs, MFA is fully supported and centrally enforced at the identity provider level.

Role-Based Access Control (RBAC)

RBAC in Cisco Intersight applies at multiple levels:

- **Global Organization Level** – High-level access control
- **Resource Groups** – Granular control of specific sets of assets
- **Predefined and Custom Roles** – Fine-tuned privilege sets

RBAC is strictly enforced within Intersight itself and does not rely on external AAA attributes (unlike UCSM, which requires LDAP/RADIUS/TACACS attribute mapping for role and locale assignment).

Note: Cisco Intersight login authentication is based on secure cloud-identity management with strong integration to modern identity providers and role-based access control (RBAC).

Cisco Nexus login authentication

Cisco NX-OS devices support local authentication and remote authentication using one or more RADIUS or TACACS+ servers. It is part of the Authentication, Authorization, and Accounting (AAA) feature for user identity verification, access permission, and activity reporting. The password-strength checking is enabled by default on NX-OS to prevent weak passwords. It uses SHA256 as the hashing algorithm for password encryption.

Local authentication

On the Cisco NX-OS devices, you can create and manage users accounts and assign roles that limit access to operations. Up to a maximum of 256 user accounts can be created. In addition, you can configure the expire option which determines the date when the user account is disabled.

RADIUS authentication

RADIUS client which runs on the NX-OS devices can send authentication requests to a RADIUS server that contains user authentication information. When the transmission to a RADIUS server timed out, the authentication is reverted to the local authentication.

On the Cisco NX-OS devices, you can also enable RADIUS server feature for it to act as a central RADIUS server for RADIUS client login authentication.

TACACS+ authentication

TACACS+ security protocol can provide centralized authentication for users to access a NX-OS device. The TACACS+ client/server protocol uses TCP port 49. In addition, TACACS+ provides separate facilities for authorization and accounting. While the RADIUS protocol only encrypts passwords, the TACACS+ encrypts the entire protocol payload between the switch and the AAA server. You must configure a TACACS+ server before configuring the TACACS+ features on your NX-OS device.

LDAP authentication

LDAP provides centralized validation of users attempting to gain access to a device. LDAP can provide support for both authentication and authorization. The LDAP client/server protocol uses TCP port 389. You must have configured an LDAP server before configuring the LDAP features on your Nexus device.

Password encryption

NX-OS supports strong, reversible 128-bit Advanced Encryption Standard (AES) password encryption, also known as type-6 encryption. AES password encryption is not enabled by default. The following example shows how to configure a master encryption key, which is used to encrypt and decrypt passwords, and enable the AES password encryption.

```
fpsa-9336-u1419# key config-key ascii
New Master Key:
Retype Master Key:

fpsa-9336-u1419# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
fpsa-9336-u1419 (config)# feature password encryption aes
fpsa-9336-u1419 (config)# show encryption service stat
Encryption service enabled
Master Encryption Key configured
Type-6 encryption is  being used
fpsa-9336-u1419 (config)# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.
fpsa-9336-u1419 (config)# exit
fpsa-9336-u1419#
```

Note: The Master-key length should be in between 16 to 64 chars(inclusive).

Note: AES password encryption is supported by the RADIUS and TACACS+ applications.

Certificate-based authentication

Cisco NX-OS supports certificate-based authentication for ssh access. The following highlights the configuration steps needed to enable certificate-based authentication for the default admin user.

Use OpenSSL on a Linux machine to generate a certificate.

```
[admin@sec-rhel-9 ~]$ openssl req -x509 -nodes -days 2190 -newkey rsa:2048 -keyout admin.key -out
admin.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=admin"
Generating a RSA private key
.....................................................................................................
.........+++++
.............+++++
writing new private key to 'admin.key'
-----
```

Note: The above command generates a public certificate named admin.pem and a private key named admin.key. The common name (CN) corresponds to the NX-OS user ID admin.

Log in to the switch and use the SCP protocol to copy the public certificate into the switch's bootflash.

```
fpsa-9336-u1419# copy scp://admin@10.61.176.196//home/admin/admin.pem bootflash:admin.pem
Enter vrf (If no input, current vrf 'default' is considered): management
The authenticity of host '10.61.176.196 (10.61.176.196)' can't be established.
ECDSA key fingerprint is SHA256:u4xLZk54XMLyTbb11viwlO8YE35A2wzfhkpyHjSf7kE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.61.176.196' (ECDSA) to the list of known hosts.
admin@10.61.176.196's password:
admin.pem
100% 1257     1.2KB/s   00:00
Copy complete, now saving to disk (please wait)...
Copy complete..
```

Enter global configuration mode, configure the public certificate file in bootflash as the sshkey, check the user account configuration, and save the configuration changes.

```
fpsa-9336-u1419# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
fpsa-9336-u1419(config)# username admin sshkey file bootflash:admin.pem
```

```
fpsa-9336-u1419(config)# show user-account admin
user:admin
        this user account has no expiry date
        roles:network-admin
        ssh public key: ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDzTzVVOImUraegHIIGvIusGEJToizufzJbqH4jyiBhRFZLuOBCdv+u9ynOLehHxcjDM
tm3IKjA8HZyNqj4gqJiEiuak6Nvv
C3VAEg3k+mYMiMtZFhd5XedvmAUHBQ72yn0wRYi1g9+nwfbvfkh3phYWtT4wpcbNGH6cBFXBgzmC7u2qDwjb8/Vv5kfuEhJuw
kh36LjMx/dGatcOyqWVtLaesTSBTRRUg//9fzh/uVqBDoJBkG9U14B/urr85
NaFKcrHKQuV41/XeTcv/WMM1WUH17L8vy8n66Mvr9ocUe0auYLtAiKXyAT2waRyoukF6ZIAGAWWhb/7TGhMroZAaVd

fpsa-9336-u1419(config)# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.

fpsa-9336-u1419(config)# exit
```

Now that the certificate-based authentication is configured on the switch for the admin user. The admin user can use ssh to interact with the switch by using certificate-based authentication and without specifying a password. See below for an example of gathering the system uptime information from the switch by using this authentication approach.

```
[admin@sec-rhel-9 ~]$ ssh -i admin.key admin@fpsa-9336-u1419 "show system uptime"
Access restricted to authorized users ONLY!
System start time:          Mon Jan  5 10:35:34 2026
System uptime:              14 days, 14 hours, 33 minutes, 41 seconds
Kernel uptime:              14 days, 14 hours, 35 minutes, 53 seconds
```

Instead of specifying the private key with the -i identify file flag, you can also copy the content of the user's private key file and save it as the user's ssh identify file .ssh/id_rsa. Afterwards, the ssh command can locate the identify file information automatically from the default location without needing to specify it as a command line parameter.

```
[admin@sec-rhel-9 ~]$ cp admin.key ~/.ssh/id_rsa

[admin@sec-rhel-9 ~]$ ssh admin@fpsa-9336-u1419 "show system uptime"
Access restricted to authorized users ONLY!
System start time:          Mon Jan  5 10:35:34 2026
System uptime:              14 days, 14 hours, 36 minutes, 1 seconds
Kernel uptime:              14 days, 14 hours, 38 minutes, 13 seconds
```

NetApp ONTAP login authentication

For ONTAP storage, you can enable local or remote cluster and Storage Virtual Machine (SVM) administrator accounts. A local account is one in which the account information, public key, or security certificate resides on the storage system. For remote accounts, the account information is stored remotely. For example, AD account information is stored on a domain controller, and LDAP and NIS accounts reside on LDAP and NIS servers.

A cluster administrator accesses the admin SVM for the cluster. The admin SVM and a cluster administrator with the reserved name admin are automatically created when the cluster is set up. A cluster administrator with the default admin role can administer the entire cluster and its resources. The cluster administrator can create additional cluster administrators with different roles as needed. An SVM administrator accesses data SVM. The cluster administrator creates data SVMs and SVM administrators as needed.

Local authentication

You can configure ONTAP to create administrator accounts to access the admin SVM or data SVMs with a password. You are prompted for the password after you enter the account creation command. During account creation, you can optionally assign an access control role to the account. If you are unsure of the

access control role that you want to assign to a login account, you can use the `security login modify` command to add/update the role for the account later.

Beginning with ONTAP 9.13.1, you can associate an X.509 certificate with the public key that you associate with the administrator account. This gives you the added security of certificate expiration or revocation checks upon SSH login for that account.

Active directory authentication

You can configure ONTAP to enable Active Directory (AD) users or group accounts to access the admin SVM or data SVMs. Any user in the AD group can access the SVM with the role that is assigned to the group.

You must configure AD domain controller access to the cluster or SVM before an AD account can access the SVM. To use AD domain controller for authentication, the ONTAP cluster time must be synchronized to within five minutes of the time on the AD domain controller.

You can use an SSH public key as either your primary or secondary authentication method with an AD user password. However, if you choose to use an SSH public key as your primary authentication, no AD authentication takes place.

LDAP or NIS authentication

You can configure ONTAP to enable LDAP or NIS user accounts to access the admin SVM or data SVMs. You must configure LDAP or NIS server access to the SVM before the account can access the SVM. Multifactor authentication is also supported for remote users using LDAP or NIS servers for authentication.

SAML authentication

You can configure Security Assertion Markup Language (SAML) authentication for web services. When SAML authentication is configured and enabled, users are authenticated by an external Identity Provider (IdP) instead of the directory service providers such as Active Directory and LDAP.

SAML authentication applies only to http and ontapi applications, which are used by the Service Processor Infrastructure, ONTAP APIs, or System Manager web services. In addition, SAML authentication is applicable only for accessing the admin SVM.

Application methods

The "`security login create`" command creates a login method for the management utility. A login method consists of a username, an application (access method), and an authentication method. It can optionally include an access-control role name.

The application method specifies the access type of the login method. Possible values include amqp, console, http, ontapi, rsh, snmp, service-processor, ssh, and telnet. A username can be associated with multiple applications. However, application access should be limited only to what the user requires.

Setting this parameter to service-processor grants the user access to the Service Processor. When this parameter is set to service-processor, the `-authentication-method` parameter must be set to password because the Service Processor only supports password authentication. SVM user accounts cannot access the Service Processor. Therefore, operators and administrators cannot use the `-vserver` parameter when this parameter is set to service-processor.

For security reasons, Telnet and Remote Shell (RSH) are disabled by default because NetApp recommends Secure Shell (SSH) for secure remote access. If there is a requirement or unique need for Telnet or RSH, they can be enabled.

Multi-factor authentication

Multifactor authentication (MFA) allows you to enhance security by requiring users to provide two authentication methods to log in to an admin or data SVM. Depending upon your version of ONTAP, you can use a combination of an SSH public key, user password, and time-based one-time password (TOTP) to set up multifactor authentication.

As an example, let's create a new administrator admin2 who is allowed to ssh into the cluster and configure a combination of password and public key for two-factor authentication.

Use the `ssh-keygen` command in Linux to create a public/private key pair for the admin2 user. The key type used here is `ecdsa-sha2-nistp256` which is supported even when the more secure Federal Information Processing Standard (FIPS) 140 compliant mode is enabled for the cluster. Enter a passphrase when prompted.

Note: Please see the FIPS 140 compliance section for information on FIPS mode.

```
[admin2@sec-rhel-9 ~]# ssh-keygen -t ecdsa-sha2-nistp256
Generating public/private ecdsa-sha2-nistp256 key pair.
Enter file in which to save the key (/home/admin2/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/admin2/.ssh/id_ecdsa
Your public key has been saved in /home/admin2/.ssh/id_ecdsa.pub
The key fingerprint is:
SHA256:kb6dIM6Jka9/z6Lxz2RrHO2f1oOda0UcIe9+6hAKG3k admin2@sec-rhel-9.5
The key's randomart image is:
+---[ECDSA 256]---+
|             . ..|
|          .    o. |
|        o      o.|
|     . . o    . o|
|    o . S E..  o |
|     * o O.o... .|
|    . * o.*o. oo+|
|     . oo=o..o.Bo|
|    ..oo.=*  +*.o|
+----[SHA256]-----+

[admin2@sec-rhel-9 ~]# cat //home/admin2/.ssh/id_ecdsa.pub
ecdsa-sha2-nistp256
*******************************lzdHAyNTYAAABBBN3VNirdrcDQS4PGs5H9AgoCdDSDzgNQdjt0SmUra6TnA85UY2X
fhTvduW0pjeJecdGhnktbJhs/B930TWA4pgc= admin2@sec-rhel-9.5
```

Create the new admin2 user in ONTAP with the "security login create" command, specifying *ssh* as the application and enabling both password and publickey authentication. Use the "security login show" command to verify the user's login configuration.

```
fpsa-a90::> security login create -user-or-group-name admin2 -application ssh -authentication-
method password -second-authentication-method publickey

Please enter a password for user 'admin2':
Please enter it again:
Warning: Public key authentication is being setup for user "admin2". This requires creating a
public key for the user. After this command completes, use the "security login publickey create"
command to create a public key for user "admin2".

fpsa-a90::> security login show -user-or-group-name admin2

Vserver: fpsa-a90
                                                                  Second
User/Group                Authentication                 Acct   Authentication
Name           Application Method        Role Name        Locked Method
-------------- ----------- ------------ ---------------- ------ --------------
admin2         ssh         password     admin            no     publickey
```

To create the public key in ONTAP for the admin2 user, use the `security login publickey create` command and provide the associated parameters, including the public key from the id_ecdsa.pub file generated above.

```
fpsa-a90::> security login publickey create -username admin2 -index 0 -publickey "ecdsa-sha2-
nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbml**************NirdrcDQS4PGs5H9AgoCdDSDzgNQdjt0SmUra6TnA85
UY2XfhTvduW0pjeJecdGhnktbJhs/B930TWA4pgc= admin2@sec-rhel-9.5" -vserver fpsa-a90
```

After the two-factor authentication is configured, the admin2 user can login to the ONTAP cluster with the needed SSH key in place and the correct password as shown below.

```
[admin2@sec-rhel-9 ~]# ssh fpsa-a90
The authenticity of host 'fpsa-a90.fpmc.sa (172.22.14.50)' can't be established.
ECDSA key fingerprint is SHA256:xcpQVQIP+MXabKKfcOvqg0Qd2cxBPBGSDe1ee9Hlkj8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'fpsa-a90.fpmc.sa,172.22.14.50' (ECDSA) to the list of known hosts.
Access restricted to authorized users ONLY!
Password:

Last login time: 11/11/2025 13:18:35
fpsa-a90::>
```

With ONTAP 9.13.1, you can configure password or SSH public key as the first authentication method and use time-based one-time password (TOTP) as the secondary password. You must configure your TOTP app to work with your smartphone and create your TOTP secret key. TOTP is supported by various authenticator apps such as Google Authenticator. With ONTAP 9.12.1 and later, you can use Yubikey hardware authentication devices for SSH client MFA using the FIDO2 (Fast IDentity Online) or Personal Identity Verification (PIV) authentication standards.

Enhanced SHA-512 password hash

Secure Hash Algorithm 2 (SHA-2) is a set of cryptographic hash functions. SHA-2 was first published by the National Institute of Standards and Technology (NIST) as a U.S. federal standard. The algorithms of the SHA-2 family are named after their digest lengths in bits, e.g., SHA-256 and SHA-512.

ONTAP 9 supports the SHA-2 password hash function and defaults to using SHA-512 for hashing newly created or changed passwords. We can check whether the newly created admin2 account is using the SHA-512 hash with the `security login show` command in advanced privilege mode as shown in the example below.

```
fpsa-a90::> set advanced

Warning: These advanced commands are potentially dangerous; use them only when directed to do so
by NetApp personnel.
Do you want to continue? {y|n}: y

fpsa-a90::*> security login show -user-or-group-name admin2 -fields hash-function
vserver   user-or-group-name application authentication-method remote-switch-ipaddress hash-
function
--------------  ------------------ -----------  --------------------  ----------------------  -----
--------
fpsa-a90 admin2         ssh         password             -                       sha512

fpsa-a90::*> set admin

fpsa-a90::>
```

Note: Administrator accounts created prior to ONTAP 9.0 continue to use MD5 password hash after the upgrade until the passwords are manually changed. By running the `security login show -fields hash-function` command in advanced mode, you can check for user passwords which are not using the sha512 has-function. If there are, you can expire the passwords for those user accounts to force a password change during their next login. Please see ONTAP documentation for additional details.

Certificate-based authentication

When using the NetApp Manageability SDK API access to ONTAP, you must use certificate-based authentication instead of the user ID and password authentication. The following highlights the step of generating and using certificate-based access for ONTAP REST API.

Use OpenSSL to generate a certificate by running the following command.

```
[admin2@sec-rhel-9 ~]#  openssl req -x509 -nodes -days 2190 -newkey rsa:2048 -keyout admin2.key -
out admin2.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=admin2"
....+..+.+..+...+......+.+........+....+..+.........+......+.........+.....+....+..++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++*.............+.+......++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++*....+......+....+...+..+......+.........+..+...+....
+...+...+..+....+.....................+...+......+......+........+.+.........+......+...+..+...+.
...........+....+.....+.......+...+..........+...+......+.+...+...+.........+........+.....+.+.
.....+.....+....+.....+.+........+.......+...+...........+..+...........+...+.+.........+......
.....+....+..+...+.+.............+...+.........+.........+.......+...........+..+.+..+.+......+
..........................+...+.........+...........+.+..............+..+...+....+.....+......+
...+......+......++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
....+.+...+.....++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+....+.....+..
.+...+.......+..+......+...+...........++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++*..............................+....+..+.......+...+........+.+...+..+.........+.+.....+.
...+..+....+...+...........+....+.............+..+.+..+.............+...+.......+............
+.........+......+......+........+....+..+.+.....+.........+..................+..+....+.........
+..............+.+.....+.+.........+.............+.+.........+........+.......+...+...+.....+...
...+............+........+...+.......+.....+................+.+......+.............+......
.+.........+.........+.............+..+...+...++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++
-----
```

Note: The above command generates a public certificate named admin2.pem and a private key named admin2.key. The common name (CN) corresponds to the ONTAP user ID admin2.

Install the contents of the public certificate in privacy enhanced mail (pem) format in ONTAP by running the following command and pasting the certificate's contents when prompted.

```
fpsa-a90::> security certificate install -type client-ca -vserver fpsa-a90
Enter certificate: Press <Enter> when done
-----BEGIN CERTIFICATE-----
MIIDdTCCAl2gAwIBAgIUHnQC1Ld/PdcihqRNJ3S53ZLeJgUwDQYJKoZIhvcNAQEL
BQAwSjELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAk5DMQwwCgYDVQQHDANSVFAxDzAN
BgNVBAoMBk5ldEFwcDEPMA0GA1UEAwwGYWRtaW4yMB4XDTI1MTIwMjA5MDEzM1oX
DTMxMTIwMTA5MDEzM1owSjELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAk5DMQwwCgYD
VQQHDANSVFAxDzANBgNVBAoMBk5ldEFwcDEPMA0GA1UEAwwGYWRtaW4yMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwl7lVEUJ/6oCYg1ulO+F54Hiw0Kq
************************************************************
Xg5dd0SE1g8Af/1srrbiMQD/g4+zDd0OTqhqvZlnl+CcSrWyrBWeViBcbYfMzkwg
xSmp4U6LkTFui8yLrSQNdrFWsmYn9xewUKdWUjHaHw938hLWYg71OnjTbKTDAohP
E4NeCEfEH1dBmOMipnDeE1rjYINTJ+7PXdPJyjIMl4uDmXttLvkua/vHfbIKhJRC
Ycce+MUYXejHxnI5JdVo3QCCFuwtdBd17yNqaLsI579qS7Lbnu/YCJiAaQIDAQAB
o1MwUTAdBgNVHQ4EFgQUa3n/DXQRwDb6cFsI+eluvWgi9/wwHwYDVR0jBBgwFoAU
a3n/DXQRwDb6cFsI+eluvWgi9/wwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0B
AQsFAAOCAQEAamqEYto5fq4tbdOeo5WLPMOco+R8KIApz6E+2/wjjJE921CK9Kw2
uRF2DXE+MvUFeaArdrynsjiTVbocRe27W9XUBz++7xq+/8mULDfuFVJxYf5xlrib
QUfhiyWgbK5AkW8K+GE3W2V8Nn4JNEkgHzPfb+020RkUv8hiwlSyO4/4MpNCTRHf
N+PNyypK3rR4hJ2xcgdTRO0nWu/kkf5guzoupIK+zkKXZFFskCOqZHJcccg9qFP3
F+sInL1fshVrNUuP7G3ynuNa2eB1IEpbr0hZB5SengO//nSuuoDWgjUAhV44smKP
qVSydqo9Ss2/Ba7htGB1u5zMYf8Sh1lVFw==
-----END CERTIFICATE-----


You should keep a copy of the CA-signed digital certificate for future reference.

The installed certificate's CA and serial number for reference:
CA: admin2
serial: 1E7402D4B77F3DD72286A44D2774B9DD92DE2605

The certificate's generated name for reference: admin2
```

Note: The entire certificate, including the BEGIN CERTIFICATE and END CERTIFICATE lines, from the public certificate file should be pasted when prompted. Some lines from the certificate file have been omitted in the example above.

Configure ONTAP to allow client access through SSL and add the http application for the ONTAP REST API access for the admin2 user as shown in the example below.

```
fpsa-a90::> security ssl modify -vserver fpsa-a90 -client-enabled true

fpsa-a90::> security login create -user-or-group-name admin2 -application http -authentication-
method cert -role admin -vserver fpsa-a90
```

Issue the following command on the Linux client to query the ONTAP version to confirm using certificate for REST API access to ONTAP.

```
[admin2@sec-rhel-9 ~]# curl -k --cert-type PEM --cert ./admin2.pem --key-type PEM --key
./admin2.key -X GET "https://fpsa-a90.fpmc.sa/api/cluster?fields=version"
{
  "version": {
    "full": "NetApp Release 9.17.1: Fri Sep 05 00:50:54 UTC 2025",
    "generation": 9,
    "major": 17,
    "minor": 1
  },
  "_links": {
    "self": {
      "href": "/api/cluster"
    }
  }
}
```

Role-based access control

With role-based access control (RBAC), users can be configured to have access to only the systems and privileges required for their job roles and functions. The FlexPod component software include predefined user roles and privileges that can be used for various needs. Customizations are also available to define custom roles using various privileges to meet specific needs.

Cisco Intersight role-based access control

Intersight provides Role-Based Access Control (RBAC) to authorize or restrict system access to a user, based on user roles and privileges. A user role in Intersight represents a collection of the privileges a user must perform a set of operations and provides granular access to resources. Intersight provides role-based access to individual users or a set of users under Groups.

A User represents an entity that can log in to a specific role in Intersight with a Cisco ID or a Single Sign-on credential configured on Intersight. A user is granted write or read-only access to the required system resources only if the specific role grants the privileges access to the role. A Privilege comprises of a set of actions a user can perform. For example, a user with a Server Administrator role can update server configurations, create, and deploy a server profile, or perform server actions on the managed servers.

A Group represents a collection of users with a specific role, permission, and privileges to manage targets within one or more Organizations. You can create multiple user groups to assign common roles and privileges to a set of users. If you make changes to a role or a privilege in a group, all users in a group inherit the same privileges.

A System-defined role is created by default in an account. Each system-defined role has multiple privilege associated with it and allows a user to perform operations as permitted by the privileges. These roles cannot be modified. For example, a Device Technician can only claim a target but not perform any other operation.

A User-defined role is created to assign multiple privileges to a user to perform different operations. These roles can be modified by an administrator as required. For example, you can create a user-defined role named device user and assign both Device Administrator and Server Administrator privileges. The user can now perform both target claim and server operations. To create a User Defined role, use the Create Role.

Note: Only users with Account Administrator or User Access Administrator privileges can manage a user-defined role.

All user roles in Intersight are tied to a set of privileges and enable a user to perform operations specific to the role. Based on your user role, you can view or perform an action related to a Server, Fabric Interconnect, or manage user access. You can assign privileges to a user using the system-defined roles or user-defined roles which can be created using the Create Role wizard.

An Intersight Account Administrator or a User Access Administrator can add a user to an Intersight account and assign one or more roles to a user. You can view a list of users with the corresponding details of their roles in the Users table view, accessible from Settings > Users.

An Intersight Account Administrator or a User Access Administrator can view the Roles and Privileges for an account. You can view information about the User's Name, Account Name, Email, and Role, accessible from System > Access Details. The Access table lists a mapping of account to privileges when logged in with an account-specific access or a mapping of Organization, its description and privileges mapped to the Organization when logged in with an Organization-specific access.

For more information on understanding role-based access control in Intersight, refer here.

Table 4) Supported System-defined roles in Intersight

System-defined roles	Permissions
Account Administrator	• Create resource groups, Create Organizations, create user-defined roles, assign Organizations to roles, delete Organizations, and delete roles. • Perform all management and administration tasks in Intersight. • Claim a target, create and deploy a server, create cluster profiles and policies, and cross launch management interfaces. • Perform server actions including firmware upgrade, power management, launch KVM, install an OS, open a TAC case, and secure erase. • Create Users and Groups, generate, and manage API keys and more. • Enable IP Access Management, add Trusted IP addresses, and unlock access for accounts that are denied access to Intersight. • Create and maintain metrics explorations. • Create and maintain metrics explorations. • Create Traffic Mirroring or Switched Port Analyzer (SPAN) sessions.
Read-only—View services and resources and perform a limited set of actions	• View all dashboard widgets, table, and detail views of Servers, Clusters, Fabric Interconnects, Profiles, and Policies. • Access metrics explorations.
Device Technician—Claim a target	• Claim a target and view the target details. • View the license and account details.
Device Administrator—Claim and manage targets	• Claim and unclaim (delete) a target. • View the target workflow tasks, search for targets in an account, license, and account details.

Server Administrator—Manage Servers	• View/add all server-related widgets on the dashboard. • Perform all operations related to a server including power management, firmware upgrade of a server, creation and deployment of server profiles and policies, installation of an operating system, and secure erase. • View claimed targets in an account, launch management interfaces and the CLI, create server profiles and policies and associate them with servers. • View license status and account details. • Create and maintain metrics explorations.
User Access Administrator— Manage users and groups in Intersight	• Create user-defined roles, assign Organizations to roles, view resource groups, view Organizations, and delete roles. • Add a User or a Group, set up third-party Identity Providers, and set up Single Sign-On. • View license status, audit logs and sessions, and account details. • Enable IP Access Management, add Trusted IP addresses, and unlock access for accounts that are denied access to Intersight.
Workload Administrator—Access workload definitions and deployments	• Access workload definitions and workload deployments. • Create new versions of workload definitions.
Workload Operator—Access workload deployments	• Access workload deployments. • View workload definitions. • View the list of workload definition versions.

Cisco Nexus role-based access control

In Cisco Nexus, rule is the basic element of a role. The rules contained in a user role define the allowed operations for the user who is assigned the role. Each user can have multiple roles and each user role can contain multiple rules. Table 5 lists the Cisco NX-OS software user roles.

Table 5) Example predefined user roles for Cisco NX-OS software.

User role	Capabilities
network-admin	• Predefined network admin role has access to all commands on the switch.
network-operator	• Predefined network operator role has access to all read commands on the switch.
vdc-admin	• Predefined vdc admin role has access to all commands within a VDC instance.
vdc-operator	• Predefined vdc operator role has access to all read commands within a VDC instance.
dev-ops	• Predefined system role for devops access. This role cannot be modified.

Note: The Cisco Nexus 9000 Series switches support a single virtual device context (VDC). As a result, the vdc-admin has the same privileges and limitations as the network-admin and the vdc-operator has the same privileges and limitations as the network-operator.

You can apply rules for commands, features, feature groups, and object identifiers. Access restrictions can be assigned for specific virtual routing and forwarding instances (VRFs), VLANs, and interfaces. See below for a few limitations of using RBAC and refer to the Cisco Nexus NX-OS documentation for complete guidelines and restriction details.

- A user account can have up to 64 user roles.
- A user role can have up to 256 rules.
- You cannot remove the default user roles from the default admin user accounts.

See below for an example of creating a read-only user with network-operator role which has complete read access to the device.

```
fpsa-9336-u1419# conf t
Enter configuration commands, one per line. End with CNTL/Z.
fpsa-9336-u1419(config)# username fpmonitor password A1s2D4f5! role network-operator
fpsa-9336-u1419(config)# exit
fpsa-9336-u1419# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.
```

For the next example, we create another admin user with username admin2, who is assigned the network-admin role and has complete control of the device.

```
fpsa-9336-u1419# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
fpsa-9336-u1419(config)# username admin2 password A1s2D4f5! role network-admin
fpsa-9336-u1419(config)# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.
fpsa-9336-u1419(config)# exit
fpsa-9336-u1419#
```

NetApp ONTAP role-based access control

The RBAC solution in ONTAP limits users' administrative access to the level granted for their defined role, which allows administrators to manage users by assigned role. ONTAP provides several predefined roles, and you can also create custom roles as necessary.

By default, a cluster administrator is assigned the predefined admin role. For the administration of the created Storage Virtual Machine (SVM), the default vsadmin account can be enabled for SVM specific management. There are also predefined SVM administrator roles that can be utilized. Table 66 and Table 77 list the example predefined roles for ONTAP cluster administrators and ONTAP SVM administrators.

Table 6) Example predefined roles for ONTAP cluster administrators.

Cluster role	Level of access	Capabilities
admin	all	All command directories (`DEFAULT`)
autosupport	all	• `set system node` • `autosupport`
	none	All other command directories (`DEFAULT`)
backup	all	`vserver services ndmp`
	readonly	volume
	none	All other command directories (`DEFAULT`)
readonly	all	Evokes all `show` commands and resets its own password

	readonly	• `security login password` For managing own user account local password and key information only • `set`
	none	All other command directories (`DEFAULT`)
none	none	All command directories (`DEFAULT`)

Note: The `autosupport` role is assigned to the predefined `autosupport` account, which is used by AutoSupport® OnDemand. ONTAP prevents you from modifying or deleting the `autosupport` account. ONTAP also prevents you from assigning the `autosupport` role to other user accounts.

Table 7) Predefined roles for ONTAP SVM administrators.

Cluster role	Capabilities
vsadmin	• Managing own user account local password and key information • Managing volumes, except volume moves • Managing quotas, qtrees, Snapshot® copies, and files • Managing LUNs • Performing SnapLock operations, except privileged delete • Configuring protocols: NFS, SMB, iSCSI, and FC. including FCoE • Configuring services: DNS, LDAP, and NIS • Monitoring jobs • Monitoring network connections and network interface • Monitoring the health of the SVM
vsadmin-volume	• Managing own user account local password and key information • Managing volumes, except volume moves • Managing quotas, qtrees, Snapshot copies, and files • Managing LUNs • Configuring protocols: NFS, SMB, iSCSI, and FC, including FCoE • Configuring services: DNS, LDAP, and NIS • Monitoring network interface • Monitoring the health of the SVM
vsadmin-protocol	• Managing own user account local password and key information • Configuring protocols: NFS, SMB, iSCSI, and FC, including FCoE • Configuring services: DNS, LDAP, and NIS • Managing LUNs • Monitoring network interface • Monitoring the health of the SVM
vsadmin-backup	• Managing own user account local password and key information • Managing NDMP operations • Making a restored volume read/write

	• Managing SnapMirror relationships and Snapshot copies • Viewing volumes and network information
vsadmin-snaplock	• Managing own user account local password and key information • Managing volumes, except volume moves • Managing quotas, qtrees, Snapshot copies, and files • Performing SnapLock operations, including privileged delete • Configuring protocols: NFS and SMB • Configuring services: DNS, LDAP, and NIS • Monitoring jobs • Monitoring network connections and network interface
vsadmin-readonly	• Managing own user account local password and key information • Monitoring the health of the SVM • Monitoring network interface • Viewing volumes and LUNs • Viewing services and protocols

See below for an example of creating a read-only user for monitoring the health of the storage cluster using ssh application with password authentication. The command prompts for a password and password confirmation.

```
fpsa-a90::> security login create -vserver fpsa-a90 -user-or-group-name fpmonitor -application
ssh -authentication-method password -role readonly

Please enter a password for user 'fpmonitor':
Please enter it again:
```

In addition to using RBAC for access control and privileges management, beginning with ONTAP 9.11.1, you can use multi-admin verification (MAV) to ensure that certain protected operations, such as deleting volumes or Snapshot® copies, can be executed only after approvals from designated administrators. Please see **Error! Reference source not found.** example for MAV configuration and usage information.

Login banners

Login banners allow an organization to present any operators, administrators, and even miscreants with terms and conditions of acceptable use, and they indicate who is permitted access to the system. This approach is helpful for establishing expectations for access and use of the system.

How the banners are configured varies between the FlexPod components. For this security hardening efforts, we are focusing on the banner message which shows up during authentication when a user tries to connect to the device and before the user is presented with the password prompt.

Cisco Nexus login banners

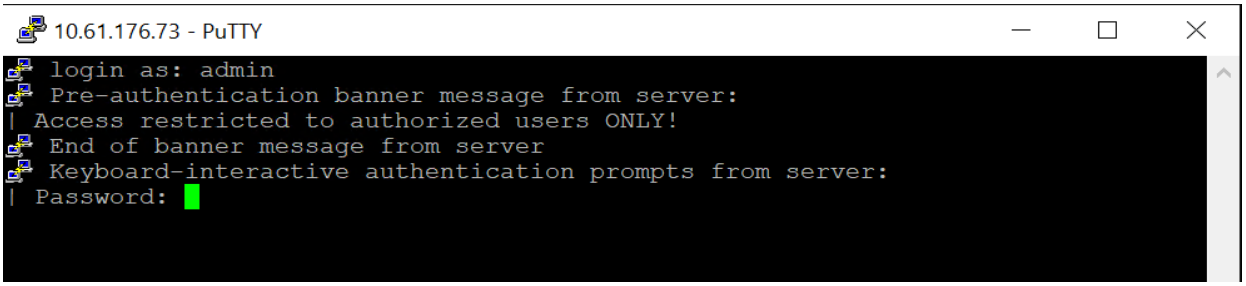
On the Cisco Nexus 9k switches, you can configure message-of-the-day (MOTD) banner to display a login banner during authentication process before the password prompt is presented to the user. The MOTD banner can be up to 40 lines and up to 80 characters per line.

To create MOTD banner, enter global configuration mode, invoke the `banner motd` command and provide the banner message inline. In the following example, (#) is used as the message delimiter. After receiving a line feed character, the (>) prompt is displayed. Repeat the delimiter (#) to complete the message for the command.

```
fpsa-9336-u1419# conf t
Enter configuration commands, one per line. End with CNTL/Z.
```

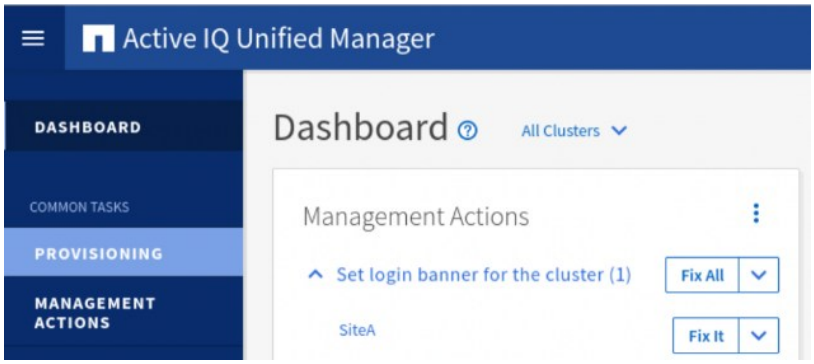
```
fpsa-9336-u1419(config)# banner motd #Access restricted to authorized users ONLY!
> #
fpsa-9336-u1419(config)# exit
fpsa-9336-u1419# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.
```

After the MOTD banner is created, log out and then login again to check the banner message.



NetApp ONTAP login banners

NetApp Active IQ Unified Manager (AIQUM) checks for the presence of the cluster login banner. If a login banner has not been configured in the cluster, AIQUM reports it as a management action on its Dashboard as shown in the screenshot below.



The Fix it button on the AIQUM Dashboard can be used to automatically configure the cluster with a login banner of “Access restricted to authorized users”.

Alternatively, the ONTAP `security login banner modify` command can be used to manually modify the login banner. The banner text must be in double quotes (“ ”), as shown in the following example.

```
fpsa-a90::> security login banner modify -vserver fpsa-a90 -message "Access restricted to
authorized users ONLY!"
```

The login banner is displayed just before the authentication step during the SSH and console device login process.

```
$ ssh admin@ fpsa-a90.fpmc.sa
Access restricted to authorized users ONLY!
Password:
```

A login banner can also be created for storage virtual machines (SVM). The follow lists how the cluster-level and SVM-level login banners interact.

- The banner configured for the cluster is also used for all SVMs that do not have a banner message defined.
- An SVM-level banner can be configured for each SVM.
- If a cluster-level banner has been configured, it is overridden by the SVM-level banner for the given SVM.

In addition to the login banner, you can also create a cluster-level and/or an SVM-level message of the day (MOTD) banner to communicate information to the users after login. Please see the `security login motd modify` command in ONTAP documentation for details on its configuration.

Login session timeout and limits

The login session timeout and limits are important to prevent stale sessions and session piggybacking and to reduce attack surfaces.

Cisco Intersight login session timeout and limits

Cisco Intersight web session refresh request and timeout

The default Cisco Intersight session timeout is 16 hours (57600 seconds), and you can configure it from 350 seconds to 1 year. Intersight also has an idle session timeout, which defaults to 1800 seconds (30 minutes). While there are no explicit session limits in terms of the number of sessions, the platform does have a configurable maximum number of sessions allowed per account and per user.

The following task provides details on how to configure account settings in Intersight.

1. Log into Intersight as a user with account administrator role.
2. Navigate to the System > Account Details.

You can view the details of the existing account settings.

3. Click Configure.

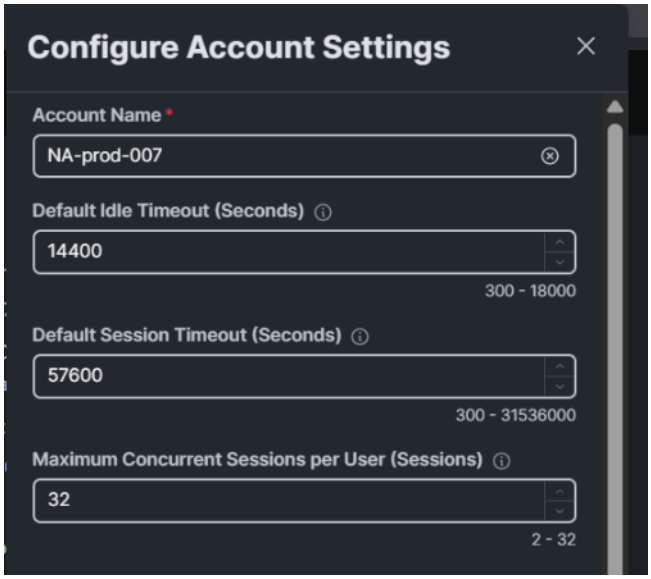
The Configure Account Settings window is displayed.

4. Update the following fields as needed.
    - Account Name—Name of the account.

Note: If you have sessions opened with Account Name URL, and you modify the Account Name, the sessions opened with the old Account Name URL are terminated.

- Default Idle Timeout (Seconds)—Provide the idle timeout interval for the web session in seconds. The system default value is 18,000 seconds (5 hours).
- Default Session Timeout (Seconds)—Provide the session expiry duration in seconds. The system default is 57,600 (16 hours).
- Maximum Concurrent Sessions per User (Sessions)—Provide the maximum number of concurrent sessions allowed per user. The system default as well as the maximum number of concurrent sessions is 32.

The example below shows setting the Default Idle Timeout, Default Session Timeout and Maximum Concurrent Sessions.



Cisco Nexus login session timeout and limits

On Cisco Nexus switches, the default inactive session timeout is 30 minutes, and the default session limit is 32. The following is an example of entering global configuration mode, updating the inactive session timeout to 15 minutes, reducing the session limit to 4, saving the configuration, and then checking for those updated settings.

```
fpsa-9336-u1419-a# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
fpsa-9336-u1419-a(config)# line vty
fpsa-9336-u1419-a config-line)# exec-timeout 15
fpsa-9336-u1419-a(config-line)# session-limit 4
fpsa-9336-u1419-a(config-line)# exit
fpsa-9336-u1419-a(config)# exit

fpsa-9336-u1419-a# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.

fpsa-9336-u1419-a# show running-config all | begin vty
line vty
  session-limit 4
  exec-timeout 15
  logout-warning 20
  absolute-timeout 0
...
```

NetApp ONTAP login session timeout and limits

The default ONTAP CLI session timeout is 30 minutes. To see the session timeout setting or to modify the timeout setting, invoke the `system timeout` commands as shown in the example below.

```
fpsa-a90::> system timeout show
CLI session timeout: 30 minutes

fpsa-a90::> system timeout modify -timeout 15

fpsa-a90::> system timeout show
CLI session timeout: 15 minutes
```

The ONTAP management session limits can also be modified based on the access interfaces (cli / ontapi / rest) and categories (application / location / request / vserver). The following example demonstrates how

to show the default management session limits, modify the defaults for the maximum active CLI sessions, and show the updated settings.

```
fpsa-a90::> security session limit show
Interface Category    Max-Active
--------- ----------- ----------
cli       application         64
cli       location            32
cli       vserver             16
ontapi    application         20
ontapi    location            20
ontapi    request             20
ontapi    vserver             10
rest      application         20
rest      location            20
rest      request             20
rest      vserver             10
11 entries were displayed.

fpsa-a90::> security session limit modify -interface cli -category * -max-active-limit 8
3 entries were modified.

fpsa-a90::> security session limit show
Interface Category    Max-Active
--------- ----------- ----------
cli       application          8
cli       location             8
cli       vserver              8
ontapi    application         20
ontapi    location            20
ontapi    request             20
ontapi    vserver             10
rest      application         20
rest      location            20
rest      request             20
rest      vserver             10
11 entries were displayed..
```

Time synchronization

The network time protocol (NTP) is used to synchronize the time between FlexPod components. Inadequate time synchronization makes troubleshooting issue logs from the various solution components of the FlexPod solution challenging.

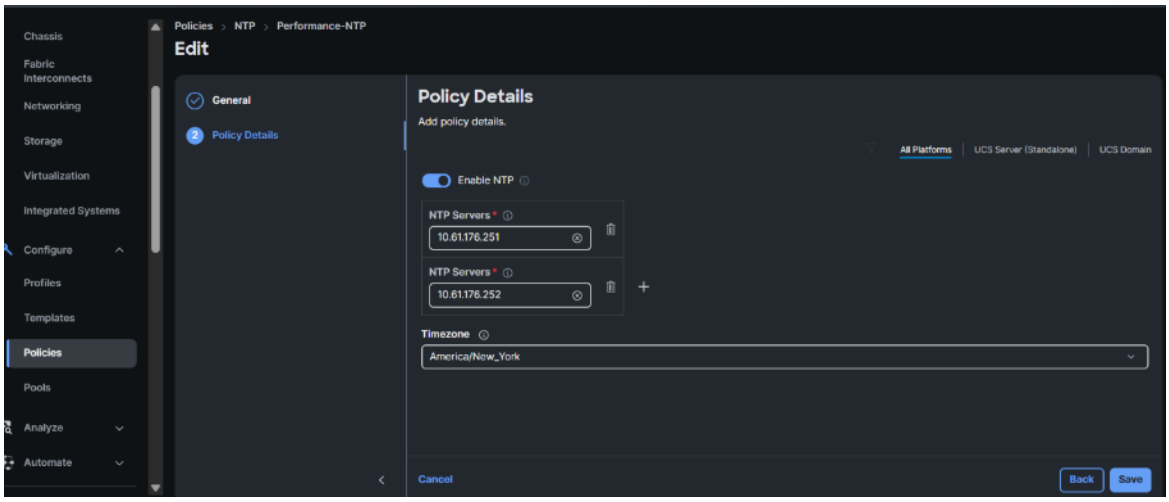
For a clustered solution like the ONTAP storage cluster, problems can occur when the time on the nodes are not accurately synchronized. As a result, it is best practice to configure and use NTP servers to synchronize the time of the FlexPod components.

Precision Time Protocol (PTP) is defined in IEEE 1588 to improve the clock synchronization of network measurements and control systems. While the accuracy expectations of NTP range from a few microseconds to tens of milliseconds depending on the environment, the accuracy expectation of PTP is in the order of 100 nanoseconds depending on the resolution and accuracy of the hardware. The improved timing precision enhances network monitoring accuracy and troubleshooting ability.

PTP is supported by Cisco Nexus and RHCOS (via PTP Operator on bare-metal) hosts. However, Cisco Intersight and NetApp ONTAP currently support NTP for time synchronization. The following sections discuss utilizing NTP for time synchronization of the FlexPod components.

Cisco UCS time synchronization

To configure NTP server in the Intersight, Go to Policies under Configure, click Create Policy, select NTP and click Start, Select the Organization, Enter the Name and click Next. Then Add NTP server details and the Timezone, then click Create.



Cisco Nexus time synchronization

NTP client

You can configure the Cisco NX-OS switch to synchronize time with an external NTP server during the initial device initialization. You also have the option to configure it from the global configuration mode and check the configuration as the following example shows.

```
fpsa-9336-u1419-a# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
fpsa-9336-u1419-a(config)# ntp server 10.61.176.251 use-vrf management
fpsa-9336-u1419-a(config)# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.
fpsa-9336-u1419-a(config)# exit

fpsa-9336-u1419(config)# show ntp peers
-------------------------------------------------
  Peer IP Address               Serv/Peer
-------------------------------------------------
  10.61.178.254                 Peer (configured)
  10.61.176.252                 Server (configured)
  10.61.176.251                 Server (configured)
```

NTP distribution

The Cisco NX-OS device can use NTP to distribute time, so other devices can use it as a time server. It can also act as an authoritative NTP server to distribute time to other devices even when it is not synchronized to an outside time source.

FlexPod CVDs and NVAs typically configure the FlexPod switches to distribute time for in-band management (IB-MGMT) network. The following is an example for configuring two NX-OS switches as NTP peers for the IB-MGMT network time distribution.

Switch A:

```
fpsa-9336-u1419# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
fpsa-9336-u1419(config)# vrf context ocp-sec
fpsa-9336-u1419(config-vrf)# ip route 0.0.0.0/0 10.61.178.1
fpsa-9336-u1419(config-vrf)# exit
fpsa-9336-u1419(config)# interface vlan178
fpsa-9336-u1419(config-if)# vrf member ocp-sec
```

```
fpsa-9336-u1419(config-if)# ip address 10.61.178.253/24
fpsa-9336-u1419(config-if)# no shutdown
fpsa-9336-u1419(config-if)# exit
fpsa-9336-u1419(config)# ntp source-interface vlan178
fpsa-9336-u1419(config)# ntp peer 10.61.178.254 use-vrf ocp-sec
fpsa-9336-u1419(config)# exit
fpsa-9336-u1419# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.
```

Switch B:

```
fpsa-9336-u1417# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
fpsa-9336-u1417(config)# vrf context ocp-sec
fpsa-9336-u1417(config-vrf)# ip route 0.0.0.0/0 10.61.178.1
fpsa-9336-u1417(config-vrf)# exit
fpsa-9336-u1417(config)# interface vlan178
fpsa-9336-u1417(config-if)# vrf member ocp-sec
fpsa-9336-u1417(config-if)# ip address 10.61.178.254/24
fpsa-9336-u1417(config-if)# no shutdown
fpsa-9336-u1417(config-if)# exit
fpsa-9336-u1417(config)# ntp source-interface vlan178
fpsa-9336-u1417(config)# ntp peer 10.61.178.253 use-vrf ocp-sec
fpsa-9336-u1417(config)# exit
fpsa-9336-u1417# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.
```

You can also configure the device to authenticate the time sources to which the local clock is synchronized. When you enable NTP authentication, the device synchronizes to a time source only if the source carries one of the authentication keys specified by the ntp trusted-key command. The device drops any packets that fail the authentication check and prevents them from updating the local clock.

Note: NTP authentication is disabled by default.

NetApp ONTAP time synchronization

Although ONTAP enables you to manually set the time zone, date, and time on the cluster, you must configure the Network Time Protocol (NTP) servers to synchronize the cluster time with external NTP servers.

You can associate a maximum of 10 external NTP servers by using the `cluster time-service ntp server create` command. For redundancy and quality of time service, you should associate at least three external NTP servers with the cluster.

Please see below for an example of using CLI to add two additional external NTP servers to ONTAP cluster time service when there is one NTP server already configured.

```
fpsa-a90::> cluster time-service ntp server show
                                     Is
                                     Authentication
Server                       Version  Enabled        Key ID
---------------------------- -------  -------------- ------
10.61.178.253                 auto     false          -
10.61.178.254                 auto     false          -
2 entries were displayed.

fpsa-a90::> cluster time-service ntp server create -server 10.61.176.251

fpsa-a90::> cluster time-service ntp server show
                                     Is
                                     Authentication
Server                       Version  Enabled        Key ID
---------------------------- -------  -------------- ------
```

```
10.61.176.251                  auto      false          -
10.61.178.253                  auto      false          -
10.61.178.254                  auto      false          -
3 entries were displayed.
```

Remote logging

Prior to setting up remote logging on the FlexPod components, a remote syslog server must already exist. If your environment does not already contain a syslog server, you must first create one. If your environment already contains a syslog server for logging events from other systems, then you might want to use that one for important event notifications as well.

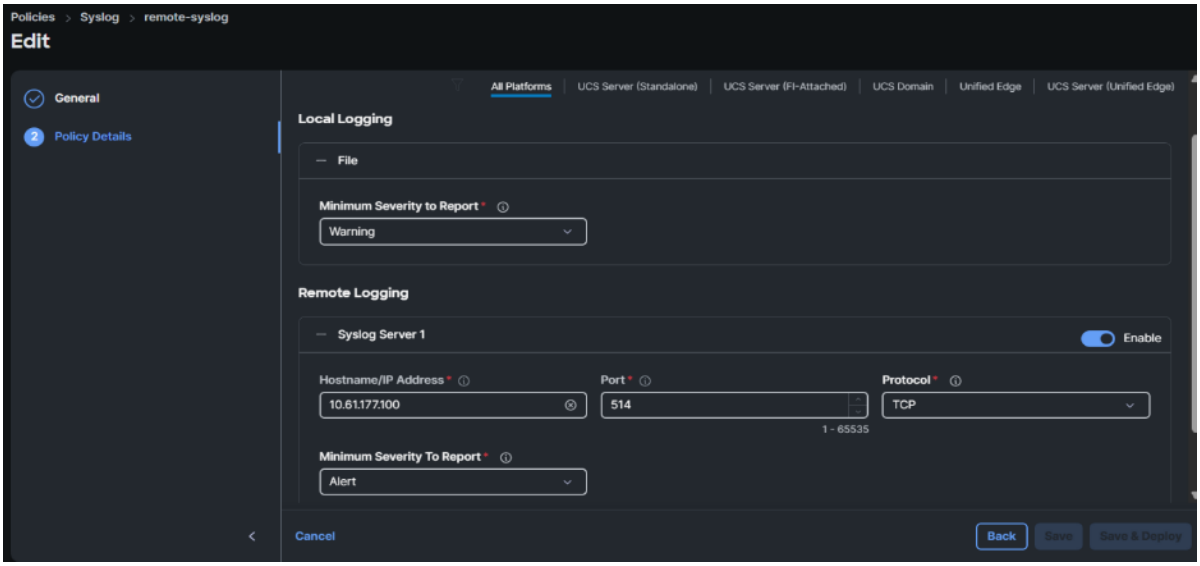
Cisco Intersight remote logging

Cisco UCS-based systems generate log messages that capture routine operations, system events, failures, and critical conditions. Cisco Intersight supports remote logging by allowing administrators to forward system, audit, and operational logs from Intersight-managed infrastructure—including UCS servers, Fabric Interconnects, and chassis—to external syslog servers for monitoring, security, and compliance.

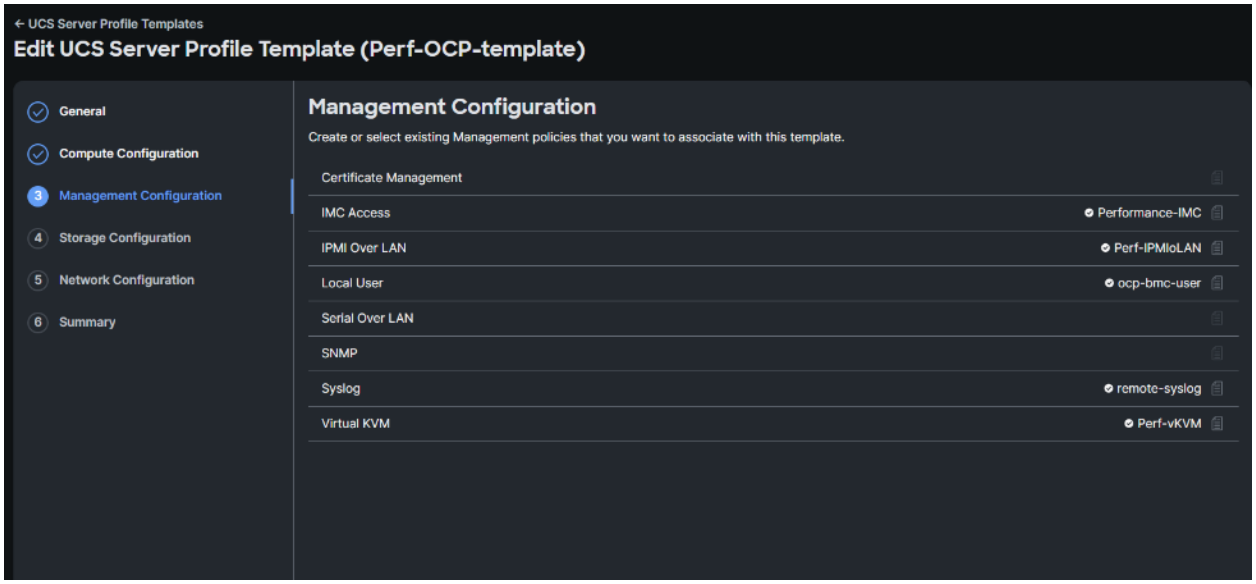
Remote syslog forwarding is configured through Intersight Syslog Policies, which apply to Fabric Interconnects and UCS servers

Steps to configure remote logging.

1. Log in to Cisco Intersight with Account Administrator or Domain Administrator role.
2. Choose Configure > Policies, and then select Create Policy.
3. Select Syslog and click Start.
4. In the General page, configure the following parameters.
    - Organization
    - Name
    - Set Tags (Optional)
    - Description
5. On the Policy Details page, configure Remote Logging with Minimum Severity To Report and click Save.



6. Attach the created remote logging to the Server Profile Template.



Cisco Nexus remote logging

System message logging controls the destination and filters the severity level of messages that system processes generate. By default, the device outputs messages to terminal sessions and logs system messages to a log file. You can also configure logging to syslog servers on remote systems.

Cisco recommends configuring the syslog server to use the management virtual routing and forwarding (VRF) instance. You can configure up to eight syslog servers to log system messages remotely.

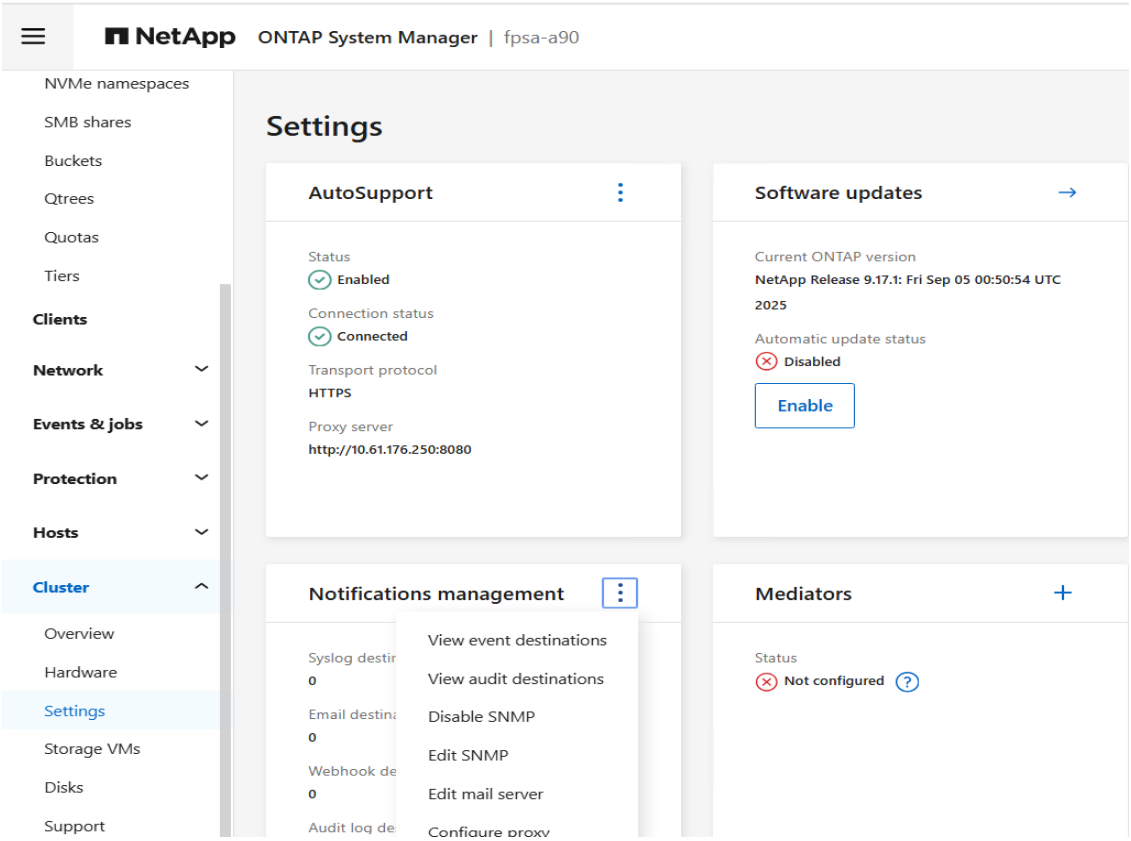
The following is an example for configuring a remote syslog server for logging messages using the management VRF.

```
fpsa-9336-u1419# conf t
Enter configuration commands, one per line. End with CNTL/Z.

fpsa-9336-u1419(config)# logging server 10.61.177.100 5 use-vrf management

fpsa-9336-u1419(config)# show logging server
Logging server:                 enabled
{10.61.177.100}
        server status:          Configured
        server severity:        notifications
        server facility:        local7
        server VRF:             management
        server port:            514

fpsa-9336-u1419(config)# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.
```

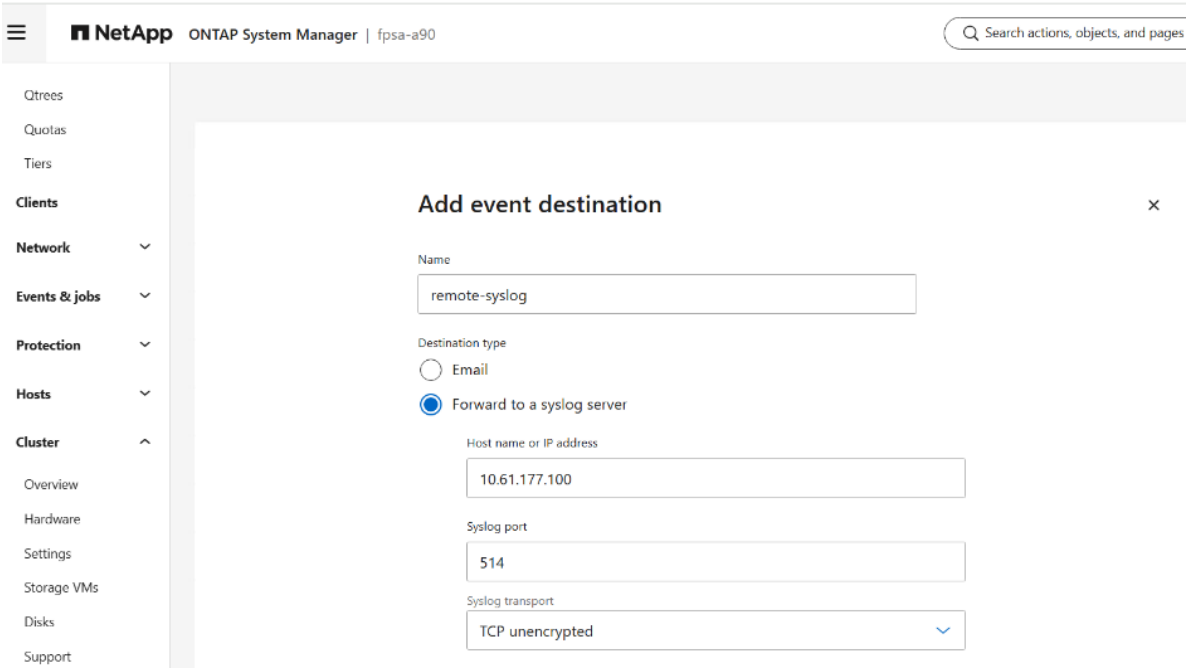
NetApp ONTAP remote logging

You can configure the ONTAP Event Management System (EMS) to forward notifications for events to a syslog server. Perform the following steps in ONTAP System Manager to configure syslog server information.

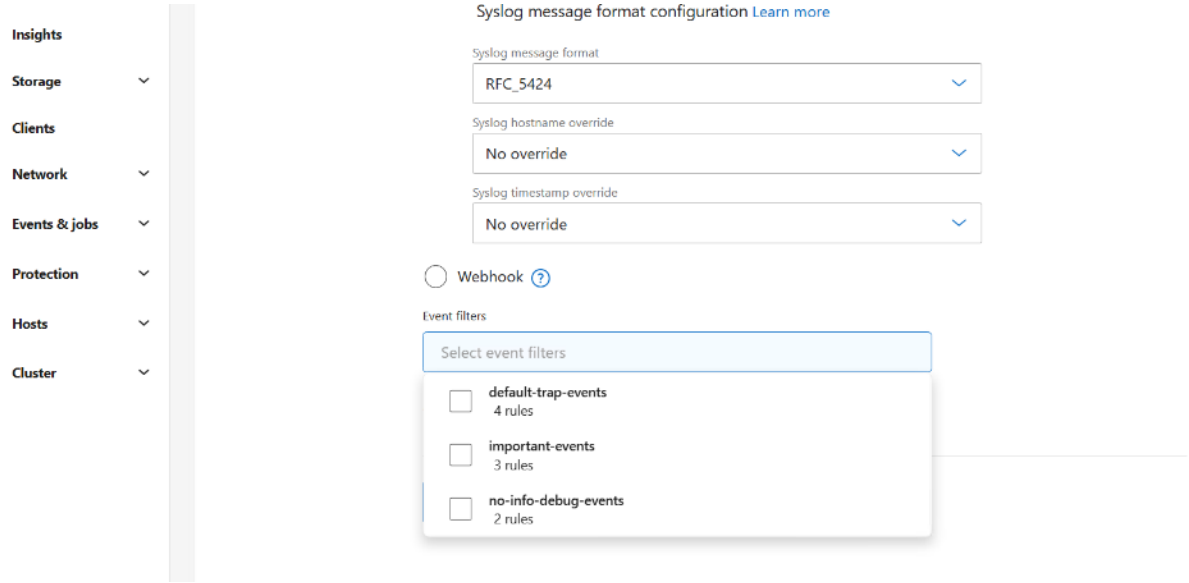
1. Click Cluster > Settings, then click the ellipses in the Notifications Management section and select View Event Destinations.



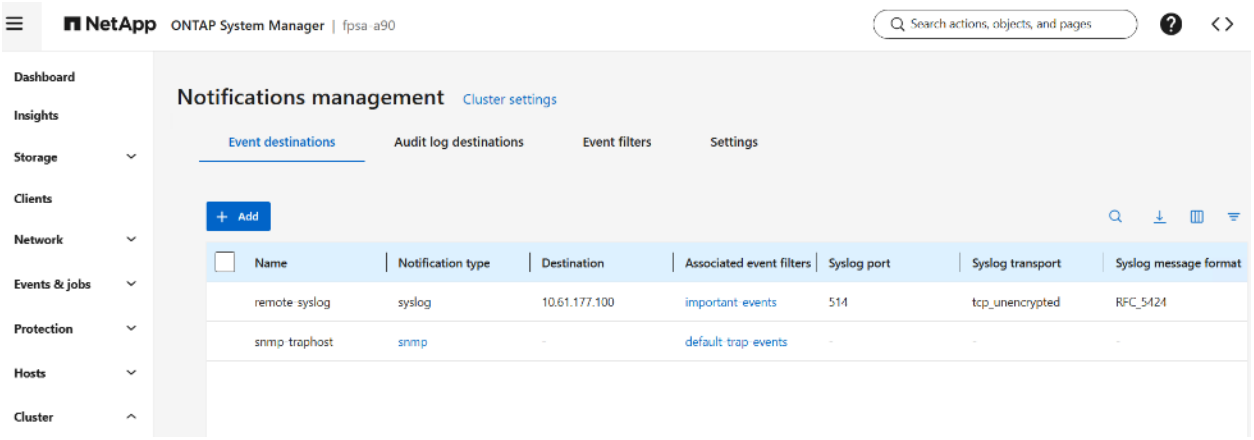
2. Select Events Destinations tab on the Notification Management page, click +Add to add a new destination.
3. In the Add event destination dialog, provide a name for the destination, select Forward to a syslog server, and enter the syslog server hostname or IP, syslog port, and syslog transport.



4. In the EVENT FILTERS field, click Select Event Filters to select an existing event filter, such as Important-events, or select Add a new event filter below it. Click Save when done and check the Event Destinations tab to confirm the created syslog destination.



5. You can also configure ONTAP to send important Event Management System (EMS) event notifications by using CLI. For example, the following command creates an event notification destination to send notifications to a syslog server. To resolve the syslog server name, DNS server



must be configured on the cluster.

```
fpsa-a90::> event notification destination create -name remote-syslog -syslog 10.61.177.100
```

You can create an event filter by using the `event filter create` command to build a filter that meets your specific event criteria. You can use the already defined event filters. You can check the event filters defined on the cluster by using the `event filter show` command.

```
fpsa-a90::> event filter show
Filter      Rule Rule                                    SNMP Trap
Name        Posn Type     Message Name     Severity      Type       Parameters
----------- ---- -------- ---------------- ------------- ---------- -----------
default-trap-events
            1    include  *                EMERGENCY, ALERT
                                                         *          *=*
            2    include  callhome.*       ERROR         *          *=*
            3    include  *                *             Standard,  Built-in
                                                                    *=*
```

```
            4    exclude  *                *              *          *=*
important-events
            1    include  *                EMERGENCY, ALERT
                                                          *          *=*
            2    include  callhome.*       ERROR          *          *=*
            3    exclude  *                *              *          *=*
no-info-debug-events
            1    include  *                EMERGENCY, ALERT, ERROR, NOTICE
                                                          *          *=*
            2    exclude  *                *              *          *=*
9 entries were displayed.
```

The following example shows how you can create event notification with the `event notification create` command to forward `important-events` to the syslog server and check the event notification configuration with the `event notification show` command.

```
fpsa-a90::> event notification create -filter-name important-events -destinations syslog-ems

fpsa-a90::> event notification show
ID   Filter Name                     Destinations
---- ------------------------------  -----------------
1    default-trap-events             snmp-traphost
2    important-events                remote-syslog
2 entries were displayed.
```

Configuration backup

It is best practice to setup scheduled backup of FlexPod component configurations. A backup server must be already set up and configured such that the FlexPod components have access to it using a protocol supported by their configuration backup.

Cisco Nexus configuration backup

The configurations of the Cisco Nexus 9000 switches can be backed up manually at any time with the copy command and automated scheduled backups can be enabled by using the NX-OS scheduler feature.

For the scheduled configuration backup to use scp protocol without specifying password each time, we need to create SSH keys for the user and transfer the public key into the SSH server where the configuration backup will be copied over.

The following shows an example of generating a pair of ssh public key and private key in the global configuration mode for the admin user and showing the generated public key information.

```
fpsa-9336-u1419 # configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
fpsa-9336-u1419(config)# username admin keypair generate rsa 2048 force

fpsa-9336-u1419(config)# show username admin keypair
**********************************

rsa Keys generated:Mon Jan  5 10:42:41 2026
ssh-rsa ******Ommited***********

bitcount:2048
fingerprint:
SHA256:3HT4hqqQxpUEH2/23xSxZ2+byUTFrH/0Vz/GdxqU4Oo**********************************

could not retrieve dsa key information
**********************************

could not retrieve ecdsa key information
**********************************
fpsa-9336-u1419(config)#
```

To transfer the generated public key into the SSH server, we need to export the previously generated keys. The export command shown in the example below requested for a passphrase to be entered and then created the key_rsa and key_rsa.pub files in the bootflash.

```
fpsa-9336-u1419(config)# username admin keypair export bootflash:key_rsa rsa force
Enter Passphrase:
fpsa-9336-u1419(config)#
```

The following shows an example of copying the public key file in the bootflash over to the SSH server.

```
fpsa-9336-u1419(config)# copy bootflash:key_rsa.pub
scp://admin@10.61.177.100/home/admin/key_rsa.pub
Enter vrf (If no input, current vrf 'default' is considered): management
admin@10.61.177.100's password:
key_rsa.pub
100%  402     0.4KB/s   00:00
Copy complete, now saving to disk (please wait)...
Copy complete.
fpsa-9336-u1419(config)#
```

On the backup server, append the public key stored in the key_rsa.pub file to the authorized_keys file.

```
[admin@sec-rhel ~]$ cat key_rsa.pub >> $HOME/.ssh/authorized_keys

[admin@sec-rhel ~]$ chmod 0700 $HOME/.ssh
[admin@sec-rhel ~]$ chmod 0600 $HOME/.ssh/authorized_keys
```

Note: If this is the first time a key is being added to the authorized_keys file, the newly created authorized_keys file may not have the right permission for the SSH server to consume. The chmod commands above updated the directory and file permissions for them to be compliant.

After the public key is in place, we can manually backup the switch configuration using scp without needing to specify the password.

```
fpsa-9336-u1419(config)# copy running-config
scp://admin@10.61.177.100/home/admin/backup/Nexus/Switch-A vrf management
fpsa-9336-u1419-running-config
100%   24KB  23.5KB/s   00:00
Copy complete, now saving to disk (please wait)...
Copy complete. .
```

Note: The directory /home/admin/backup/Nexus had already been created previously.

The following is an example of configuration backup scheduling for one of the Nexus switches.

```
fpsa-9336-u1419# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
fpsa-9336-u1419(config)# feature scheduler
fpsa-9336-u1419(config)# scheduler logfile size 1024
fpsa-9336-u1419(config)# scheduler job name backup-cfg
fpsa-9336-u1419(config-job)# copy run
scp://admin@10.61.177.100/home/admin/backup/Nexus/$(SWITCHNAME)-cfg.$(TIMESTAMP) vrf management
fpsa-9336-u1419(config-job)# exit
fpsa-9336-u1419(config)# scheduler schedule name daily
fpsa-9336-u1419(config-schedule)# job name backup-cfg
fpsa-9336-u1419(config-schedule)# time daily 12:00
fpsa-9336-u1419(config-schedule)# end
```

Note: Using “vrf management” in the copy command is only needed when Mgmt0 interface is part of VRF management.

The `show schedule job` and `show scheduler schedule` commands can be used to verify that the backup job and schedule have been correctly set up. Save the configuration changes by copying the running-config to startup-config.

```
fpsa-9336-u1419# show scheduler job
```

```
Job Name: backup-cfg
--------------------
  copy running-config scp://admin@10.61.177.100/home/admin/backup/Nexus/$(SWITCHNAME)-
cfg.$(TIMESTAMP) vrf management

==============================================================================

fpsa-9336-u1419# show scheduler schedule
Schedule Name       : daily
---------------------------
User Name           : admin
Schedule Type       : Run every day at 12 Hrs 0 Mins
Last Execution Time : Yet to be executed
-----------------------------------------------
     Job Name            Last Execution Status
-----------------------------------------------
backup-cfg                              -NA-
==============================================================================

fpsa-9336-u1419# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.
```

After the scheduled backup time, you can check the backup server for the existence of the switch configuration backup files.

```
[admin@sec-rhel ~]$ ls -l /home/admin/backup/Nexus/
total 72
-rw-r--r--. 1 admin admin 24359 Jan  5 12:00 fpsa-9336-u1419-cfg.2026-01-05-12.00.00
-rw-r--r--. 1 admin admin 24359 Jan  6 12:00 fpsa-9336-u1419-cfg.2026-01-06-12.00.00
-rw-r--r--. 1 admin admin 24359 Jan  7 12:00 fpsa-9336-u1419-cfg.2026-01-07-12.00.01
```

NetApp ONTAP configuration backup

The configuration backup files of the NetApp ONTAP cluster and nodes are automatically created according to the following schedules:

- Every 8 hours
- Daily
- Weekly

At each of these times, a node configuration backup file is created on each healthy node in the cluster. All these node configuration backup files are then collected in a single cluster configuration backup file along with the replicated cluster configuration and saved on one or more nodes in the cluster.

An example of viewing the ONTAP cluster configuration backup files in advanced privilege mode is shown below:

```
fpsa-a90::> set advanced

Warning: These advanced commands are potentially dangerous; use them only when directed to do so
by NetApp personnel.
Do you want to continue? {y|n}: y

fpsa-a90-::*> system configuration backup show
Node       Backup Name                                Time                Size
---------  ------------------------------------------ ------------------- -----
fpsa-a90-01  fpsa-a90.8hour.2025-12-03.18_15_01.7z 12/03 18:15:01 52.17MB
fpsa-a90-01  fpsa-a90.8hour.2025-12-04.02_15_00.7z 12/04 02:15:00 52.72MB
fpsa-a90-01  fpsa-a90.daily.2025-12-03.00_10_01.7z 12/03 00:10:01 51.37MB
fpsa-a90-01  fpsa-a90.daily.2025-12-04.00_10_00.7z 12/04 00:10:00 52.95MB
fpsa-a90-01  fpsa-a90.weekly.2025-11-23.00_15_01.7z 11/23 00:15:01 54.62MB
fpsa-a90-01  fpsa-a90.weekly.2025-11-30.00_15_01.7z 11/30 00:15:01 55.43MB
fpsa-a90-02  fpsa-a90.8hour.2025-12-03.18_15_01.7z 12/03 18:15:01 52.17MB
fpsa-a90-02  fpsa-a90.8hour.2025-12-04.02_15_00.7z 12/04 02:15:00 52.72MB
```

```
fpsa-a90-02  fpsa-a90.daily.2025-12-03.00_10_01.7z 12/03 00:10:01 51.37MB
fpsa-a90-02  fpsa-a90.daily.2025-12-04.00_10_00.7z 12/04 00:10:00 52.95MB
fpsa-a90-02  fpsa-a90.weekly.2025-11-23.00_15_01.7z 11/23 00:15:01 54.62MB
fpsa-a90-02  fpsa-a90.weekly.2025-11-30.00_15_01.7z 11/30 00:15:01 55.43MB
12 entries were displayed.

fpsa-a90::*> set admin
fpsa-a90::>
```

You can use `system configuration backup settings` commands in advanced mode to manage configuration backup schedules and specify a remote URL (HTTP, HTTPS, FTP, FTPS, or TFTP) where the configuration backup files will be uploaded in addition to the default locations in the cluster.

An example of setting up an automated ONTAP cluster configuration backup upload destination, username, and password is shown below:

```
fpsa-a90::> set advanced

Warning: These advanced commands are potentially dangerous; use them only when directed to do so
by NetApp personnel.
Do you want to continue? {y|n}: y

fpsa-a90::*> system configuration backup settings modify -destination
ftp://10.61.177.100/home/admin/backup/ONTAP -username admin

fpsa-a90::*> system configuration backup settings show
Backup Destination URL                            Username
------------------------------------------------- -------------
ftp://10.61.177.100/home/admin/backup/ONTAP     admin

fpsa-a90::*> system configuration backup settings set-password

Enter the password:
Confirm the password:

fpsa-a90::*>
```

FIPS 140 compliance

The Federal Information Processing Standard 140 (FIPS 140) is a U.S. government standard that sets security requirements for cryptographic modules in hardware, software, and firmware to protect sensitive information. Compliance with the standard is mandated for use by U.S. government agencies, and it is also often used in industries such as financial services and healthcare.

Under the FIPS 140 standard, both the algorithm and the module are evaluated for compliance, using programs that are jointly developed by the U.S. National Institute of Standards and Technology (NIST) and the Canadian Centre for Cyber Security (CCCS).

The Cryptographic Module Validation Program (CMVP) is the accreditation program for cryptographic module security. The Cryptographic Algorithm Validation Program (CAVP) provides guidelines for validating the effectiveness of FIPS-approved and NIST-recommended cryptographic algorithms.

FIPS 140 security requirements encompass 11 areas related to the design, strength, and operation of a cryptographic module. In each of the 11 areas, there are four security levels. Level 1 is the least restrictive, specifying the lowest level of security, and Level 4 specifies the highest level.

Note: Please see links in the reference section for more FIPS compliance information available from NetApp, Cisco, and Red Hat.

Cisco UCS FIPS 140 compliance

Cisco UCS is compliant with FIPS140-2 level 1 through direct implementation of the FIPS-compliant Cisco SSL crypto module. The module, once implemented, is vetted by a third party that is federally certified to ascertain compliance status.

- Utilizes Cisco SSL module
  - Already FIPS compliant
  - SSH-approved cipher list
  - SSL/TLS implementation
  - Eliminates weak or compromised components
- Regularly updated
- Lab validates that the module is incorporated correctly
  - Build logs
  - Source-access-identifying calls to the module
  - All admin access points to the cluster are covered here
- SSH for CLI
- HTTPS for UI

Cisco Intersight

Cisco Intersight integrates FIPS 140-2 validated cryptographic modules to ensure secure operation of its TLS and SSH services. It incorporates the Cisco FIPS Object Module v6.2 for SSH and v7.2 (for TLS, providing approved algorithms for session establishment, key derivation, hashing, and symmetric encryption. Intersight operates in FIPS mode by default at build time, requiring no additional configuration.

Note: FIPS 140-2 level 2 on Intersight Private Virtual Appliance / Connected Virtual Appliance is only applicable to single-node appliance.

Cisco Nexus FIPS 140 compliance

Enable FIPS mode

Cisco NX-OS devices support FIPS mode. It has the following prerequisites, configuration guidelines, and limitations.

- Disable Telnet. Users should log in using Secure Shell (SSH) only.
- Disable SNMPv1 and v2. Any existing user accounts on the device that have been configured for SNMPv3 should be configured only with SHA for authentication and AES/3DES for privacy.
- Delete all SSH server RSA1 key-pairs.
- Enable HMAC-SHA1 message integrity checking (MIC) for use during the Cisco TrustSec Security Association Protocol (SAP) negotiation. To do so, enter the sap hash-algorithm HMAC-SHA-1 command from the cts-manual or cts-dot1x mode.
- The user authentication mechanisms supported for SSH are usernames and passwords, public keys, and X.509 certificates.
- Your passwords should have a minimum of eight alphanumeric characters.
- Disable Radius and TACACS when FIPS mode is on. This is enforced due to OpenSSL in FIPS mode.

By default, FIPS mode is disabled on the Cisco NX-OS devices. To enable FIPS mode on the Cisco NX-OS devices, perform the following steps on all FlexPod switches.

Re-generate RSA key with 2048 key bits from the console, if the switch was initialized with an RSA key which was not using 2048 key bits. In the example output below, the actual key information has been removed as noted.

```
fpsa-9336-u1419# conf terminal
Enter configuration commands, one per line. End with CNTL/Z.

fpsa-9336-u1419(config)# no feature ssh
XML interface to system may become unavailable since ssh is disabled

fpsa-9336-u1419(config)# no ssh key rsa

fpsa-9336-u1419(config)# ssh key rsa 2048

fpsa-9336-u1419(config)# show ssh key
**************************************
rsa Keys generated:Mon Jan  5 15:19:10 2026

ssh-rsa **********

bitcount:2048
fingerprint:
SHA256:k5zc69mJ4whMe9+2KMKFYoBFQLBLhdk5YIOtckLYwJM
**************************************
could not retrieve dsa key information
**************************************
could not retrieve ecdsa key information
**************************************

fpsa-9336-u1419(config)# feature ssh

fpsa-9336-u1419(config)# exit

fpsa-9336-u1419# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.
```

Enter global configuration mode to enable FIPS mode. After FIPS mode is enabled, reboot the switch for the FIPS mode to take effect as instructed.

```
fpsa-9336-u1419# conf terminal
Enter configuration commands, one per line. End with CNTL/Z.

fpsa-9336-u1419(config)# fips mode enable

FIPS mode is enabled
System reboot is required after saving the configuration for the system to be in FIPS modeFpsa-
9336-u1419(config)# exit

fpsa-9336-u1419# copy running-config startup-config
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.

fpsa-9336-u1419# reload
This command will reboot the system. (y/n)?  [n] y
```

Check FIPS status once the switch is up.

```
fpsa-9336-u1419# show fips status
 FIPS Status: enabled
Switch Mode: FIPS
----------------------
LC          STATUS
```

```
----------------------
```

NetApp ONTAP FIPS 140 compliance

Enable FIPS mode

NetApp ONTAP data management software has a FIPS mode configuration that instantiates an added level of security for the customer. This FIPS mode applies to the control plan and secures all control Interfaces of ONTAP.

When FIPS mode is enabled, there are related security practices that will be enforced.

- Transport Layer Security v1.1 (TLSv1.1) is disabled, and only TLS v1.2 and TLS v1.3 remain enabled.
- An SNMP users or SNMP traphosts that are non-compliant to FIPS will be deleted automatically.
- An SNMPv1 user, SNMPv2c user or SNMPv3 user (with none or MD5 as authentication protocol or none or DES as encryption protocol or both) is non-compliant to FIPS.
- An SNMPv1 traphost or SNMPv3 traphost (configured with an SNMPv3 user non-compliant to FIPS) is non-compliant to FIPS.

In addition, before you enable FIPS mode on your cluster, existing SSH public key accounts without the supported key algorithms must be reconfigured with a supported key type. The accounts should be reconfigured before you enable FIPS or the administrator authentication will fail. Table 8 lists host key type algorithms that are supported for ONTAP SSH connections.

Table 8) Supported host key type alogrithms for ONTAP SSH connections

ONTAP release	Key types supported in FIPS mode	Key types supported in non-FIPS mode
9.11.1 and later	ecdsa-sha2-nistp256	ecdsa-sha2-nistp256 rsa-sha2-512 rsa-sha2-256 ssh-ed25519 ssh-dss ssh-rsa
9.10.1 and earlier	ecdsa-sha2-nistp256 ssh-ed25519	ecdsa-sha2-nistp256 ssh-ed25519 ssh-dss ssh-rsa

By default, the FIPS 140-2 mode is disabled on the ONTAP cluster. To enable FIPS mode on the ONTAP cluster, elevate privilege from admin to advanced mode and use the `security config modify` command to enable it.

```
fpsa-a90::> set advanced

Warning: These advanced commands are potentially dangerous; use them only when directed to do so
by NetApp personnel.
Do you want to continue? {y|n}: y

fpsa-a90::*> security config modify * -is-fips-enabled true
1 entry was modified.
```

The protocols and cipher suites supported by ONTAP can be retrieved by using the `security config show` command. The following provides a truncated output from that command. Please note that the supported protocols and cipher suites for ONTAP can be further restricted by using the `security config modify` command and providing a list of protocols and cipher suites with `-supported-protocols` and `-supported-cipher-suites` options.

```
fpsa-a90::*> security config show
Cluster    Supported
FIPS Mode  Protocols Supported Cipher Suites
---------- --------- ------------------------------------------------------------
true       TLSv1.3,  TLS_RSA_WITH_AES_128_CCM, TLS_RSA_WITH_AES_128_CCM_8,
           TLSv1.2   TLS_RSA_WITH_AES_128_GCM_SHA256,
                     TLS_RSA_WITH_AES_128_CBC_SHA,
                     TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CCM,
                     TLS_RSA_WITH_AES_256_CCM_8,
                     TLS_RSA_WITH_AES_256_GCM_SHA384,
                     TLS_RSA_WITH_AES_256_CBC_SHA,
                     TLS_RSA_WITH_AES_256_CBC_SHA256,
                     TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,
                     TLS_DHE_DSS_WITH_AES_128_CBC_SHA,
                     ...
```

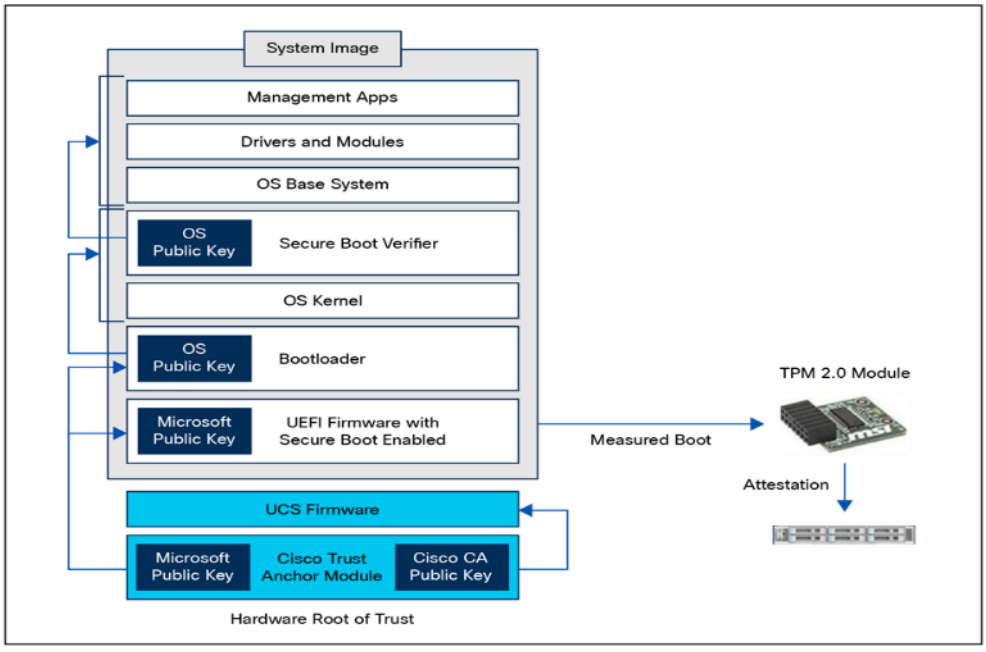
UEFI secure boot

Cisco Secure Boot helps to ensure that the code that executes on Cisco hardware platforms is authentic and unmodified. Cisco hardware-anchored secure boot protects the microloader (the first piece of code that boots) in tamper-resistant hardware, establishing a root of trust that helps prevent Cisco network devices from executing tainted network software. Subsequent boot of the installed operating system is verified and attested with the Trusted Platform Module (TPM).

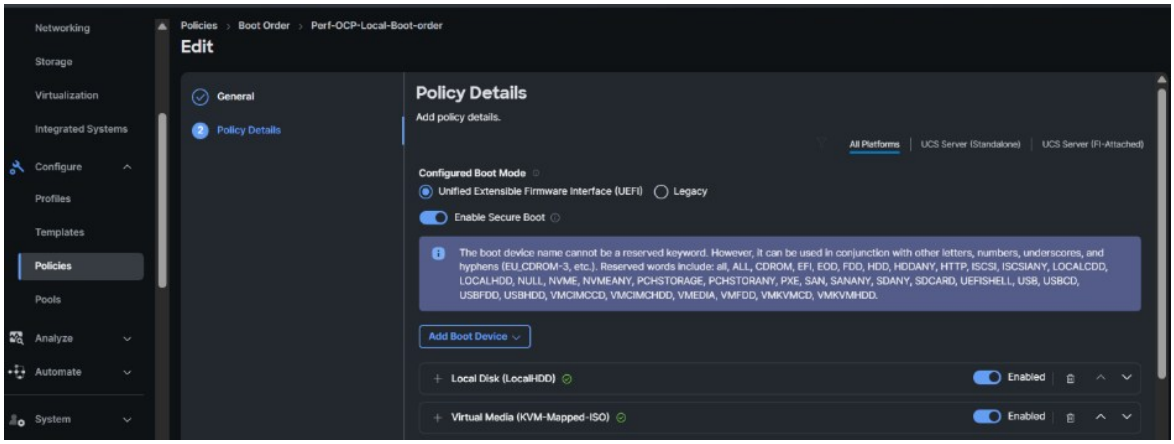
Cisco Secure Boot helps ensure that the code that executes on Cisco hardware platforms is genuine and untampered. A typical UEFI-based boot process starts at the UEFI firmware and works up to the boot loader and the operating system. A tampered UEFI firmware can result in the entire boot process being compromised.

Using a hardware-anchored root of trust, digitally signed software images, and a unique device identity, Cisco hardware-anchored secure boot establishes a chain of trust that boots the system securely and validates the integrity of the software. The root of trust (a.k.a. the microloader), which is protected by tamper-resistant hardware, first performs a self-check, and then verifies the UEFI firmware, and thus kicks off the chain of trust leading to integrity verification of the entire operating system. Figure 12) UCS Secure Boot Process illustrates the secure boot process in UCS.

Figure 12) UCS Secure Boot Process



In a FlexPod configuration, UEFI secure boot is enabled in the Cisco Intersight boot order policy configuration. In the policy, you will select **UEFI** as the boot mode and Enable Secure Boot and then add the desired boot devices (like local disks, virtual media, or SAN boot options) in the correct order.



Since Secure Boot is already enabled in the Cisco UCS BIOS, it will also be active on the OpenShift RHCOS layer. You can confirm this by logging in to any RHCOS node and checking its Secure Boot status.

```
[admin@sec-rhel-9 ~]$ ssh core@10.61.178.101
The authenticity of host '10.61.178.101 (10.61.178.101)' can't be established.
ECDSA key fingerprint is SHA256:TdbzN/Qlz6QONeZ45p/cZvMOe79/ylMWDOa/djIzTQs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.61.178.101' (ECDSA) to the list of known hosts.
Red Hat Enterprise Linux CoreOS 9.6.20250916-0
  Part of OpenShift 4.19, RHCOS is a Kubernetes-native operating system
  managed by the Machine Config Operator (`clusteroperator/machine-config`).
WARNING: Direct SSH access to machines is not recommended; instead,
make configuration changes via `machineconfig` objects:
  https://docs.openshift.com/container-platform/4.19/architecture/architecture-rhcos.html

---
Last login: Thu Oct 30 15:56:02 2025 from 10.61.176.196
[systemd]
Failed Units: 1
  NetworkManager-wait-online.service

[core@control-1 ~]$ mokutil --sb-state
       SecureBoot enabled
```

Note: When UEFI Secure Boot is enabled, the NVIDIA GPU driver installation will fail. To install the NVIDIA GPU driver, Secure Boot must be disabled.

Image validation

To secure a FlexPod solution, updating software regularly to apply bug fixes and address security vulnerabilities is important part of the FlexPod best practices. To ensure image integrity of the software you install on your FlexPod solution components, download those images from the official software download sites from Cisco, NetApp, and Red Hat as shown below.

- Cisco software download: https://software.cisco.com/download/home
- NetApp software download: https://mysupport.netapp.com/site/downloads
- Red Hat software download: https://access.redhat.com/downloads

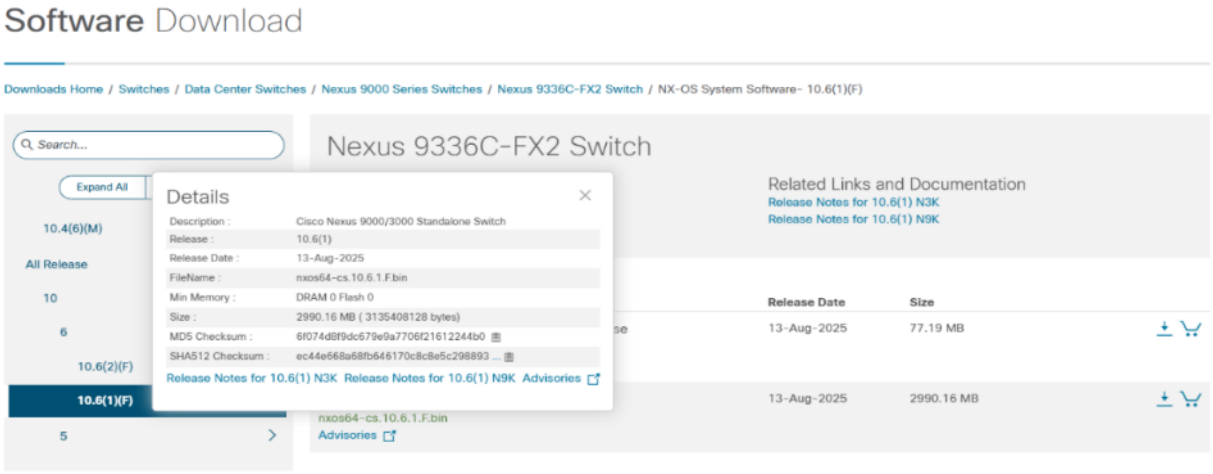
After downloading the images, you can utilize the cryptographic hashes, e.g., sha-1, sha-256, or md5, that are provided for the images to confirm the integrity of the downloaded images.

To confirm the integrity of a downloaded image, use a sha-1, sha-256, and/or a md5 utility, such as sha1sum, sha256sum, and md5sum on a Linux distribution, to calculate your own hash for the downloaded image file. If the calculated hash matches the message digest provided on the official download site, you can be assured of the integrity of the downloaded image. An example of such image validation is provided in the ONTAP image validation subsection.

Cisco Nexus image validation

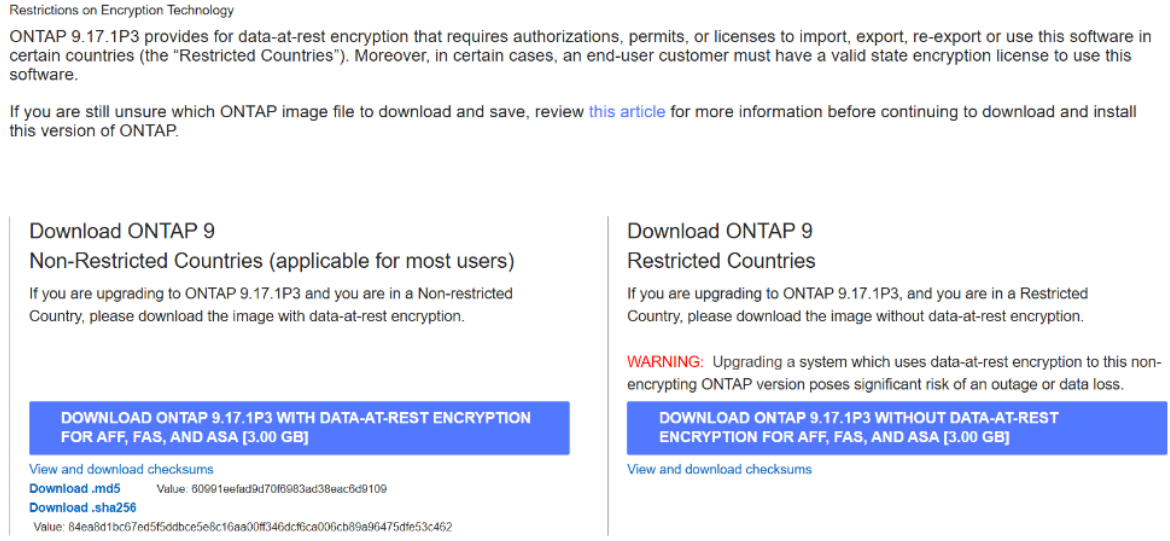
For Cisco Nexus NX-OS images, you can search for the NX-OS software and then narrow down the search by selecting Switches > Data Center Switches > Nexus 9000 Series Switches and then pick the switch model you are using and the firmware version you would like to download.

In the example screenshot below, Nexus 9336C-FX2 switch, and NX-OS software type were selected along with the 10.6.1.F firmware version. Once you hover your mouse on top of the Cisco Nexus 9000/3000 Standalone Switch location, the Details for the firmware information pop up. You can use the copy icon to the right of the MDS and SHA512 checksums to copy those checksums to validate your downloaded image.



NetApp image validation

For NetApp ONTAP software download, there are two different images, one with NetApp Volume Encryption included and the other without it. Be sure to review the Restrictions on Encryption Technology information on the page and select the correct one for your situation.



Please also note that you can click on View and download checksums link to view and download the md5 and sha256 checksums to validate the downloaded image. An example of comparing the ONTAP software image md5 and sha256 checksums against the downloaded checksums using a Linux system is shown below.

```
[admin@sec-rhel-9 ocp-sec]$ md5sum 9171P3_q_image.tgz
60991eefad9d70f6983ad38eac6d9109  9171P3_q_image.tgz

[admin@sec-rhel-9 ocp-sec]$ cat 9171P3_q_image.tgz.md5
60991eefad9d70f6983ad38eac6d9109 9171P3_q_image.tgz

[admin@sec-rhel-9 ocp-sec]$ sha256sum 9171P3_q_image.tgz
84ea8d1bc67ed5f5ddbce5e8c16aa00ff346dcf6ca006cb89a96475dfe53c462  9171P3_q_image.tgz

[admin@sec-rhel-9 ocp-sec]$ cat 9171P3_q_image.tgz.sha256
84ea8d1bc67ed5f5ddbce5e8c16aa00ff346dcf6ca006cb89a96475dfe53c462 9171P3_q_image.tgz
```

NFS access authorization

Layered access configuration overview

There are several layers of configurations that will be needed within the various components of a FlexPod virtual infrastructure solution to provide NFS protocol data services. Here are highlights of some of the configurations needed.

- NFS VLAN will need to be properly configured in all Nexus switches, in Intersight, and for ONTAP storage data ports.
- The ONTAP storage virtual machine (SVM) providing the NFS protocol service must have the NFS protocol enabled and NFS LIFs configured.
- If switch ACL is in use for the NFS VLAN, then the IPs of the OpenShift worker nodes NFS interface and the NFS LIFs of the ONTAP storage cluster must be allowed to access the NFS VLAN.
- The ONTAP SVM must have the NFS volume created and exported, and the SVM export policy must allow access from the OpenShift worker nodes NFS interfaces.

ONTAP export policy

ONTAP uses export policy to permit client access to NFS volumes. An export policy works very much like the access control list on the Nexus switches. The ONTAP NFS export policy contains one or more export rules, and the policy can be associated with a volume to configure client access to the volume. The result of this rule-checking process determines whether a client is granted or denied (with a permission-denied message) access to the volume.

When a new storage virtual machine (SVM) is created, a default export policy (called default) is created automatically. You must create one or more rules for the default export policy before clients can access data on the SVM. You should verify that access is open to all NFS clients in the default export policy and later restrict access to individual volumes by creating custom export policies for individual volumes as necessary.

The following example illustrates the commands for showing the export-policy of the infrastructure SVM, creating an export-policy rule for NFS clients using a subnet matching approach, showing the export-policy rule created, and assigning the export-policy to the root volume of the SVM.

```
fpsa-a90::*> vserver export-policy show -vserver ocp-svm.fpmc.sa -policyname default

    Vserver: ocp-svm.fpmc.sa
Policy Name: default

fpsa-a90::*> vserver export-policy rule create -vserver ocp-svm.fpmc.sa  -policyname default -
ruleindex 1 -protocol nfs -clientmatch 172.22.16.0/24 -rorule sys -rwrule sys -superuser sys -
allow-suid true
```

```
fpsa-a90::*> vserver export-policy rule show -vserver ocp-svm.fpmc.sa -policyname default
             Policy          Rule    Access   Client                RO
Vserver      Name            Index   Protocol Match                 Rule
------------ --------------- ------  -------- --------------------- ---------
ocp-svm.fpmc.sa
             default         1       nfs      172.22.16.0/24        sys
```

Note: Using the entire NFS subnet for the export-policy rule simplifies NFS client access configuration. However, if individual NFS client IP match is desirable due to security consideration, then IP address of the individual client should be provided to the `-clientmatch` parameter. Additional export-policy rules will need to be added to allow additional NFS clients to access the volume.

Kerberos considerations

If Kerberos is used in your environment for strong authentication, you need to work with your Kerberos administrator to determine requirements and appropriate storage system configurations and then enable the SVM as a Kerberos client. In addition, the environment requires DNS, NTP, and a secure directory service such as Active Directory, or OpenLDAP configured to use LDAP over SSL/TLS.

Also, if Kerberos authentication is required, select NFSv4 or later version to fully realize the security benefits of Kerberos in ONTAP. When Kerberos is enabled on the SVM, be sure to enable Kerberos on several data LIFs on multiple nodes for redundancy and use one of the three available security methods: krb5 (Kerberos 5 protocol), krb5i (Kerberos v5 protocol with integrity checking using checksums), or krb5p (Kerberos v5 protocol with privacy service).

ONTAP data-at-rest encryption

Data-at-rest encryption is important to protect sensitive data in the event a storage system is stolen, repurposed, or returned. ONTAP 9 has three FIPS 140-2 compliant data-at-rest encryption solutions:

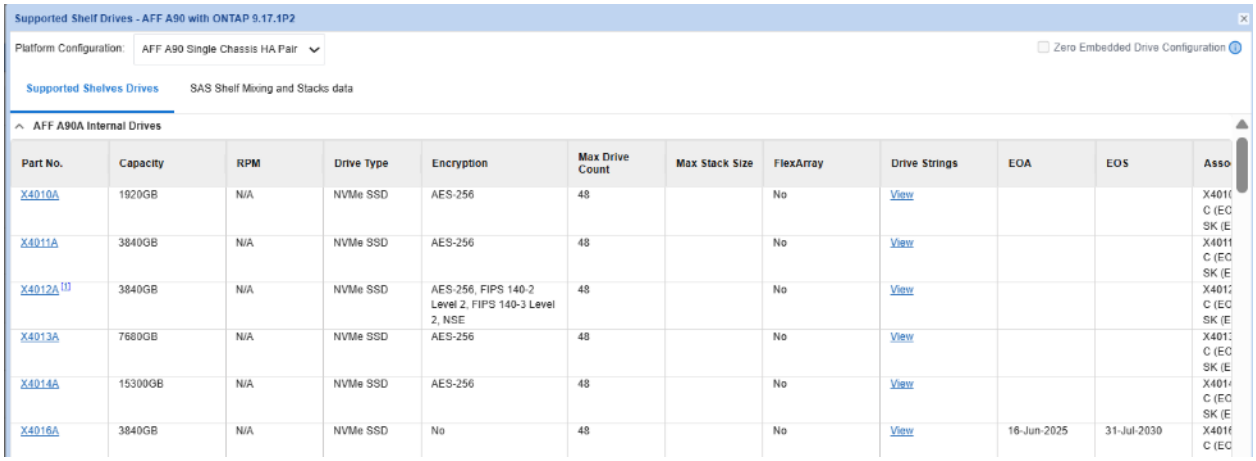
- NetApp Storage Encryption (NSE) is a hardware solution that uses self-encrypting drives (SEDs).
- NetApp Aggregate Encryption (NAE) is a software solution that enables encryption of any data volume on any drive type where it is enabled with unique keys for each aggregate.
- NetApp Volume Encryption (NVE) is a software solution that enables encryption of any data volume on any drive type where it is enabled with a unique key for each volume.

With NSE, full disk encryption is available with FIPS 140-2 level 2 SEDs. Full disk encryption is also available for NVMe SEDs that do not have FIPS 140-2 certification.

NSE, NAE, and NVE can use either external key management or the onboard key manager (OKM). Use of NSE, NAE, and NVE does not affect ONTAP storage efficiency features. NAE volumes participate in and benefit from aggregate deduplication. However, NVE volumes are excluded from aggregate de-duplication.

If you need to segregate access to data and make sure that data is protected all the time, NSE SEDs can be combined with network- or fabric-level encryption. NSE SEDs act like a backstop if an administrator forgets to configure or misconfigures higher-level encryption. By using both software (NAE or NVE) and hardware (NSE or NVMe SED) you can achieve double encryption at rest.

Note: You can check on NetApp Hardware Universe, https://hwu.netapp.com, for the drive support details for your platform configuration and the running ONTAP release. The screenshot below shows the supported drives, including self-encrypting drives, for the AFF A90 storage system.



Part No.	Capacity	RPM	Drive Type	Encryption	Max Drive Count	Max Stack Size	FlexArray	Drive Strings	EOA	EOS	Asso
X4010A	1920GB	N/A	NVMe SSD	AES-256	48		No	View			X401(C (EC SK (E
X4011A	3840GB	N/A	NVMe SSD	AES-256	48		No	View			X4011 C (EC SK (E
X4012A [1]	3840GB	N/A	NVMe SSD	AES-256, FIPS 140-2 Level 2, FIPS 140-3 Level 2, NSE	48		No	View			X401 C (EC SK (E
X4013A	7680GB	N/A	NVMe SSD	AES-256	48		No	View			X401 C (EC SK (E
X4014A	15300GB	N/A	NVMe SSD	AES-256	48		No	View			X401 C (EC SK (E
X4016A	3840GB	N/A	NVMe SSD	No	48		No	View	16-Jun-2025	31-Jul-2030	X401(C (EC

ONTAP data-in-flight encryption

ONTAP IPsec data-in-flight encryption

Internet Protocol Security (IPsec) is an IETF standard. ONTAP uses IPsec in transport mode to ensure data is continuously secure and encrypted, even while in transit. ONTAP IPsec provides end-to-end encryption support for all IP traffic between a client and an ONTAP SVM. IPsec data encryption for all IP traffic includes support for NFS, iSCSI, and SMB/CIFS protocols. After IPsec is configured, network traffic between the client and ONTAP is protected with preventive measures to combat replay and man-in-the-middle (MITM) attacks.

Providing NFS encryption over the wire is one of the main use cases for IPsec. Prior to ONTAP 9.8, NFS over-the-wire encryption required the setup and configuration of Kerberos to utilize krb5p to encrypt NFS data in flight. This is not always simple or easy to accomplish in every customer's environment.

You can check if IPsec is enabled on the cluster and enable it to ensure data is continuously secure and encrypted, even while in transit, as shown in the following example.

```
fpsa-a90::> security ipsec config show
     IPsec Enabled: false
   IPsec Log Level: 2
Replay Window Size: 0
   Offload Enabled: false

fpsa-a90::> security ipsec config modify -is-enabled true

fpsa-a90::> security ipsec config show
     IPsec Enabled: true
   IPsec Log Level: 2
Replay Window Size: 0
   Offload Enabled: false
```

Although the IPsec capability must be enabled on the cluster, it applies to individual SVM IP addresses using a Security Policy Database (SPD) entry. The policy (SPD) entry contains the client IP address (remote IP subnet), SVM IP address (local IP subnet), the encryption cipher suite to use, and the pre-shared key (PSK). In addition to the IPsec policy entry, the client must be configured with the same information (local and remote IP, PSK, and cipher suite) before traffic can flow over the IPsec connection.

Note: For NetApp SnapMirror and cluster peering traffic encryption, cluster peering encryption (CPE) is still recommended over IPsec for secure in-transit over the wire. CPE performs better for these workloads than IPsec. You do not need a license for IPsec, and there are no import or export restrictions.

Note: See the ONTAP 9 Network Management documentation for more information about IPsec.

ONTAP cluster peering encryption

You can protect your data by replicating it to a remote cluster for backup, restore, migration, and disaster recovery purposes using features such as NetApp SnapMirror® and SnapVault®. For leveraging SnapMirror the ONTAP clusters must be in a peer relationship so that they can communicate with each other and perform the data mirroring. The source cluster and destination cluster use inter-cluster network interfaces to communicate with each other and to exchange data.

New cluster-peer relationships established with ONTAP 9.6 and later have cluster peering encryption (CPE) enabled by default. CPE uses a PSK and the Transport Layer Security (TLS) to secure cross-cluster peering communications. Cluster peering encrypts all data between the cluster peers. For example, when using SnapMirror, all peering information as well as all SnapMirror relationships between the source and destination cluster peer are encrypted. This adds an additional layer of security between the peered clusters.

To check if CPE is enabled, issue the `cluster peer show` command to examine the encryption setting. As shown in the example below, tls-psk is the encryption protocol used for inter-cluster communication.

```
fpsa-a90::> cluster peer show -instance


                      Peer Cluster Name: nb-d11-a400
          Remote Intercluster Addresses: 10.195.14.81, 10.195.14.82
     Availability of the Remote Cluster: Available
                   Remote Cluster Name: nb-d11-a400
                   Active IP Addresses: 10.195.14.82, 10.195.14.81
                  Cluster Serial Number: 1-80-000011
                   Remote Cluster Nodes: nb-d11-a400-01, nb-d11-a400-02
                  Remote Cluster Health: true
                Unreachable Local Nodes: -
         Address Family of Relationship: ipv4
    Authentication Status Administrative: use-authentication
       Authentication Status Operational: ok
                        Last Update Time: 1/14/2026 14:56:40
            IPspace for the Relationship: Default
Proposed Setting for Encryption of Inter-Cluster Communication: -
Encryption Protocol For Inter-Cluster Communication: tls-psk
  Algorithm By Which the PSK Was Derived: akep2
```

Note: To enable encryption on cluster peer relationships that were created before ONTAP 9.6, you must first upgrade the source and destination cluster to 9.6 and later and then use the `cluster peer modify` command to change both the source and destination cluster peers to use cluster peering encryption. For more information about cluster peering encryption, see the Cluster and SVM peering information in the ONTAP 9 documentation.

ONTAP ransomware protection

Ransomware attacks continue to be big threats to enterprises as attacks can potentially lead to business disruptions, monetary and data losses, and business reputation impacts. NetApp provides a suite of tools and solutions that can be utilized to help detect and recover from ransomware attacks quickly to minimize business impacts.

The ONTAP FPolicy framework is used to monitor and manage file access. ONTAP volume Snapshot is a point-in-time copy of the volume data for quick recovery. Autonomous ransomware protection (ARP) automates detection and provides protection and alerts for ransomware attacks. Additional external tools such as NetApp Cloud Insights can help make the orchestration and protection against ransomware simple.

FPolicy

NetApp FPolicy is a file access notification framework that is used to monitor and manage file access events on storage virtual machines (SVMs) through native or partner solutions. There are two parts to an FPolicy solution. The first part of the solution is the ONTAP framework which contains and maintains the FPolicy configuration, monitors file events, and sends notifications to native or external FPolicy servers. The second part of the solutions is the native or external FPolicy servers which process notifications sent by ONTAP FPolicy to fulfill customer use cases such as data governance, compliance, and ransomware protection.

FPolicy determines how the storage system handles requests from individual client systems for operations such as create, open, rename, and delete. There are two basic FPolicy configuration types. One configuration uses external FPolicy servers, such as NetApp Cloud Insights, to process and act upon notifications. The other configuration uses the ONTAP native FPolicy server for simple file blocking based on extensions.

For example, an administrator can configure FPolicy to scan for files based on the following to avoid storing mp3 file types on the system:

- File extensions (natively supported in ONTAP): For example, block all files matching *.mp3. This is a fast but less reliable approach. Native support for file blocking based on file extensions does not require a connection to any external FPolicy server.
- File signatures (requires external FPolicy server): For example, block all files with signature matching mp3 format. This approach is slower as the FPolicy server needs to access the data in the file. However, this approach is more accurate as signature matching is performed.

The administrator should enable events for CREATE, OPEN, CLOSE and RENAME requests. When the FPolicy server is notified of these event triggers, it can run checks based on either of the two mechanisms (file extension or file signature) and DENY requests if a match is found.

The external FPolicy notifications are sent either in synchronous or asynchronous mode. With asynchronous notifications, the node does not wait for a response. Asynchronous notification is suitable for applications where the FPolicy server does not require that actions be taken because of notification evaluation such as for the monitoring and auditing of the file access activities. With synchronous notifications, the FPolicy server must acknowledge every notification. Synchronous notification is used when an action is required based on the results of notification evaluation such as deciding whether to allow or deny requests based on criteria specified on the external FPolicy server.

Volume Snapshot and policy

A Snapshot copy is a read-only, point-in-time image of a volume. A Snaphot copy consumes minimal storage space and incurs negligible performance overhead. You can use a Snapshot copy to restore the entire contents of a volume, or to recover individual files or LUNs. Snapshot copies are stored in the snapshot directory on the volume.

A Snapshot policy defines how the system creates Snapshot copies. The policy specifies when to create Snapshot copies, how many copies to retain, and how to name them. The default policy for a volume automatically creates Snapshot copies based on the following schedule, with the oldest Snapshot copies deleted to make room for newer copies:

- A maximum of six hourly Snapshot copies taken five minutes past the hour.
- A maximum of two daily Snapshot copies taken Monday through Saturday at 10 minutes after midnight.
- A maximum of two weekly Snapshot copies taken every Sunday at 15 minutes after midnight.

You can examine the default snapshot policy as shown in the example below.

```
fpsa-a90::> snapshot policy show -policy default
```

```
Vserver: fpsa-a90
                                 Number of Is
Policy Name                      Schedules Enabled Comment
-------------------------------- --------- ------- ---------------------------------
default                                  3 true    Default policy with hourly, daily & weekly schedules.
    Schedule       Count  Prefix        SnapMirror Label    Retention Period
    -------------- -----  ------------- ------------------  ------------------
    hourly             6  hourly        -                   0 seconds
    daily              2  daily         daily               0 seconds
    weekly             2  weekly        weekly              0 seconds
```

Unless you specify a Snapshot policy when you create a volume, the volume inherits the Snapshot policy associated with its containing storage virtual machine (SVM).

With ONTAP release 9.12.1 and later, you can also lock a Snapshot copy on a non- SnapLock® volume. Locking Snapshot copies ensures that they can't be deleted accidentally or maliciously.

Autonomous ransomware protection

In addition to ransomware detection and prevention using external FPolicy user behavioral analytics (UBA) with NetApp Cloud Insights and the NetApp FPolicy partner ecosystem, ONTAP 9.10.1 introduces autonomous ransomware protection. ONTAP anti-ransomware uses a built-in on-box machine learning (ML) capability that looks at NAS (NFS and SMB) volume workload activity plus data entropy to automatically detect ransomware. It monitors for activity that is different from UBA so that it can detect attacks that UBA does not.

When you enable autonomous ransomware protection (ARP), it runs in learning mode. In learning mode, the ONTAP system develops an alert profile based on the analytic areas: entropy, file extension types, and file IOPS. After running ARP in learning mode for enough time to assess workload characteristics, you can switch to active mode and start protecting your data. Once ARP has switched to active mode, ONTAP will create ARP snapshots to protect the data if a threat is detected.

Although you can switch from learning mode to active mode at any time, it is recommended that you leave ARP in learning mode for a minimum of 30 days. Switching to active mode too early might lead to too many false positives. Beginning with ONTAP 9.13.1, ARP automatically determines the optimal learning period interval and automates the switch, which may occur before 30 days.

SnapLock protection / compliance

ONTAP SnapLock® is a high-performance compliance solution that provides capability of data retention and WORM protection for retained data. You can use SnapLock to create non-modifiable and non-erasable volumes to prevent files from being altered or deleted until a set retention date.

SnapLock allows this retention to be performed at the file level through standard open file protocols such as CIFS and NFS. The supported open file protocols for SnapLock are NFS (versions 2, 3, and 4) and CIFS (SMB 1.0, 2.0, and 3.0).

SnapLock WORM storage uses NetApp Snapshot technology and can leverage SnapMirror replication, and SnapVault backups as the base technology for providing backup recovery protection for data.

You use the SnapLock compliance clock feature to lock Snapshot copies for a specified period so that they cannot be deleted until the expiration time is reached. Locking Snapshot copies makes them tamperproof and protects them from ransomware threats. You can use locked Snapshot copies to recover data if a volume is compromised by a ransomware attack.

Secure NX-OS with Cisco Live Protect

Cisco Live Protect is a security feature that

- provides real-time, kernel-level security using Extended Berkeley Packet Filter (eBPF) technology through the Tetragon agent embedded in NX-OS.

- enables Common Vulnerabilities and Exposures (CVE) compensating controls and zero-day attack mitigation without software upgrades or reboots, and
- uses NXSecure to manage and activate the feature, configuring Tetragon for monitoring, detection, and logging of security events for enhanced threat visibility and telemetry integration.

While managing data center network infrastructure, security and uptime are non-negotiable. Patch updates for CVEs lead to operational disruptions and unacceptable downtime for critical systems. With Cisco Live Protect feature, you can address emerging vulnerabilities immediately by deploying real-time shields that mitigate CVE exploitation. This proactive approach eliminates the need for disruptive patching, emergency maintenance, or urgent code upgrades in your data center. Thus, the Live Protect feature allows you to maintain continuous protection and operational stability.

Release-specific information

- Beginning with NX-OS Release 10.6(1)F, only monitoring mode is supported.
- Beginning with NX-OS Release 10.6(2)F, enforce mode is also supported.

To enable NXSecure feature for Cisco Live Protect we must enable nxsecure feature on the switch.

By default, the feature is disabled.

```
show feature | grep nxsecure
nxsecure              1          disabled
```

Run the command below to enable.

```
fpsa-9336-u1419(config)# feature nxsecure
fpsa-9336-u1419(config)# show feature | grep nxsecure
nxsecure              1          enabled

fpsa-9336-u1419(config)# show nxsecure status
Tetragon Agent Status: Running
```

Verify the current set of generated log files.

```
fpsa-9336-u1419(config)# show nxsecure logfiles
  tetragon.log
  tetragon-2026-01-23T10-35-20.293.log
  tetragon-2026-01-23T10-35-08.261.log
  tetragon-2026-01-23T10-34-47.192.log
  alert-rule-1-all-execution-by-admin.log
  alert-rule-1-all-execution-by-admin-2026-01-23T09-37-08.725.log
  alert-rule-3-vsh-exec-by-non-root.log
  alert-rule-2-privilege-execution-by-admin.log
  alert-rule-3-vsh-exec-by-non-root-2026-01-23T09-37-08.725.log
  alert-rule-2-privilege-execution-by-admin-2026-01-23T09-37-08.725.log
  alert-rule-3-vsh-exec-by-non-root-2026-01-23T07-37-08.724.log
  alert-rule-1-all-execution-by-admin-2026-01-23T07-37-08.724.log
  alert-rule-2-privilege-execution-by-admin-2026-01-23T07-37-08.724.log
  alert-rule-3-vsh-exec-by-non-root-2026-01-23T05-37-08.723.log
  alert-rule-2-privilege-execution-by-admin-2026-01-23T05-37-08.723.log
  alert-rule-1-all-execution-by-admin-2026-01-23T05-37-08.723.log
```

Further, you can configure telemetry for collecting data for analyzing and troubleshooting. For more information about the telemetry, refer to the NX-OS programmability guide.

You can utilize Visore, which is the built-in browser for the NX-OS DME (Data Management Engine) model. Telemetry on NX-OS (gRPC/GPB, JSON/HTTP, NX-API) requires “sensor paths,” which define what the switch should stream. These sensor paths come from the DME object tree, not from CLI commands.

OpenShift overview and installation

OpenShift and Red Hat Linux CoreOS overview

Red Hat Enterprise Linux CoreOS (RHCOS) is the operating system base for OpenShift Container Platform (OCP). As a lightweight and purpose-built operating system, it is based on Red Hat Enterprise Linux 9 and uses the same kernel, code, open-source development process, and ships with a specific subset of RHEL software packages.

RHCOS is built and supported for use in OpenShift 4 clusters. Its primary goal is to provide a secure operating system platform for running Kubernetes, OpenShift services, and the containerized workloads running on the aggregated platform.

RHCOS Design

RHCOS represents the next generation of single-purpose container operating system technology. Created by the same development teams that created Red Hat Enterprise Linux Atomic Host and CoreOS Container Linux, RHCOS combines the quality standards of Red Hat Enterprise Linux (RHEL) with automated, remote upgrade features from Container Linux.

RHCOS is supported only as a component of OpenShift Container Platform 4 for all OpenShift Container Platform machines. It is the only supported operating system for the OpenShift Container Platform control plane or master machines. While RHCOS is the default operating system for all cluster machines, some of the cluster is composed of compute nodes, which are also known as worker nodes. These may require certain flexibility in design to enable certain workload types. To accommodate such scenarios, worker nodes can be created that use RHEL as their operating system, instead of RHCOS.

There are two general ways RHCOS is deployed in OpenShift Container Platform 4:

- If the cluster is installed on infrastructure that the cluster provisions, RHCOS images are downloaded to the target platform during installation, and suitable Ignition config files, which control the RHCOS configuration, are used to deploy the machines. This approach represents the installer-provisioned installation model.
- If the cluster is installed on a local infrastructure, follow the installation documentation to obtain the RHCOS images, generate Ignition config files, and use the Ignition config files to provision the machines. This model is the approach associated with user-provisioned infrastructure.

Key RHCOS Features

The following list describes key features of the RHCOS operating system:

- **Based on RHEL**- The underlying operating system consists primarily of RHEL components. The same quality, security, and control measures that support RHEL also support RHCOS. For example, RHCOS software is in RPM packages, and each RHCOS system starts up with a RHEL kernel and a set of services that are managed by the systemd init system.
- **Controlled immutability**- Although it contains RHEL components, RHCOS is designed to be managed more tightly and indirectly than a default RHEL installation. Management is performed remotely from the OpenShift Container Platform cluster. On set up, RHCOS machines have only a few system settings which can be modified. This controlled immutability allows an OpenShift Container Platform to store the latest state of RHCOS systems in the cluster, so it is always able to create additional machines and perform updates based on the latest RHCOS configurations.
- **CRI-O container runtime-** RHCOS contains features for running the OCI- and libcontainer formed containers that Docker requires. It does this by incorporating the CRI-O container engine instead of the Docker container engine. CRI-O focuses on features needed by Kubernetes platforms. In this approach, CRI-O can offer specific compatibility with different Kubernetes versions. CRI-O also offers a smaller footprint and reduced attack surface than is possible with

container engines that offer a superset beyond Kubernetes-centric features. Since the OpenShift Container Platform is really Kubernetes, it benefits from these features as well. Now, CRI-O is only available as a container engine within OpenShift Container Platform clusters.

- **Set of command line container tools** For tasks such as building, copying, and otherwise managing containers, RHCOS replaces Docker with a compatible set of container tools. The podman command supports many container runtime features, such as running, starting, stopping, listing, and removing containers and container images. The skopeo command can copy, authenticate, and sign images. The crictl command can be used to work with containers and pods from the CRI-O container engine. While direct use of these tools in RHCOS is discouraged, they can be used for debugging purposes.
- **rpm-ostree upgrades** RHCOS features transactional upgrades using the rpm-ostree system. Updates are delivered by means of container images and are part of the OpenShift Container Platform update process. When deployed, the container image is pulled, extracted, and written to disk, then the bootloader is modified to boot into the new version. The machine will reboot into the update in a rolling manner to ensure cluster capacity is minimally impacted.
- **Updated through Machine Config Operator** In OpenShift Container Platform, the Machine Config Operator handles operating system upgrades. Instead of upgrading individual packages, as is done with Yum upgrades, the rpm-ostree command delivers upgrades of the OS as an atomic unit. The new OS deployment is staged during upgrades and goes into effect on the next reboot. If something goes wrong with the upgrade, a single rollback and reboot returns the system to the previous state. RHCOS upgrades in OpenShift Container Platform are performed during cluster updates.

Everything in a Container

For RHCOS systems the layout of the rpm-ostree file system has the following characteristics : rpm-ostree upgrades RHCOS features transactional upgrades using the rpm-ostree system. Updates are delivered by means of container images and are part of the OpenShift Container Platform update process. When deployed, the container image is pulled, extracted, and written to disk, then the bootloader is modified to boot into the new version. The machine will reboot into the update in a rolling manner to ensure cluster capacity is minimally impacted.

RHCOS is not intended to be run outside of an OpenShift Cluster. Conceptually, RHCOS is the base layer of OpenShift Container Platform 4. As a member of the cluster, RHCOS is managed by and upgraded through the cluster. To run software, admins for RHCOS should use containers.

Machine Config Operator

OpenShift 4 is an operator-driven platform. This approach allows specific operators such as the Machine Config Operator (MCO) to take a declarative approach to cluster component life cycle management. This effectively allows the automated management of cluster updates that range from the kernel to services higher in the stack. The MCO is the component that joins RHCOS and the OpenShift Cluster together. MachineConfig resources provide the interface to thread OS configuration through the cluster from bootstrap to cluster upgrades. Some specific configuration options are abstracted into higher-level knobs within MachineConfigs such as kernelType to enable the real-time kernel or kernelArguments. Configuration for the container runtime and the kubelet can be handled with their respective precursor objects, ContainerRuntimeConfigs and KubeletConfigs, which are translated into MachineConfigs by dedicated controllers.

Ignition

The first boot agent driving RHCOS is Ignition. From Ignition’s Github page : Ignition is the utility used by CoreOS Container Linux, Fedora CoreOS, and RHEL CoreOS to manipulate disks during the initramfs. This includes partitioning disks, formatting partitions, writing files (i.e. regular files, systemd units), and configuring users. On first boot, Ignition reads its configuration from a source of truth (remote URL, network metadata service, hypervisor bridge, etc.) and applies the configuration. During installation, the

installer dynamically creates Ignition profiles to setup and install RHCOS and OpenShift. While Ignition configuration files can be used to change the behavior of the bootstrap, it should only be done when necessary, such as to enable hardware or set targeted settings.

OpenShift – installation and configuration

For the installation of OpenShift cluster we have used agent-based installation. The Agent-based installation method provides flexibility to boot your on-premises servers in any way that you choose. It combines the ease of use of the Assisted Installation service with the ability to run offline, including in air-gapped environments. Agent-based installation is a subcommand of the OpenShift Container Platform installer. It generates a bootable ISO image containing all the information required to deploy an OpenShift Container Platform cluster, with an available release image.

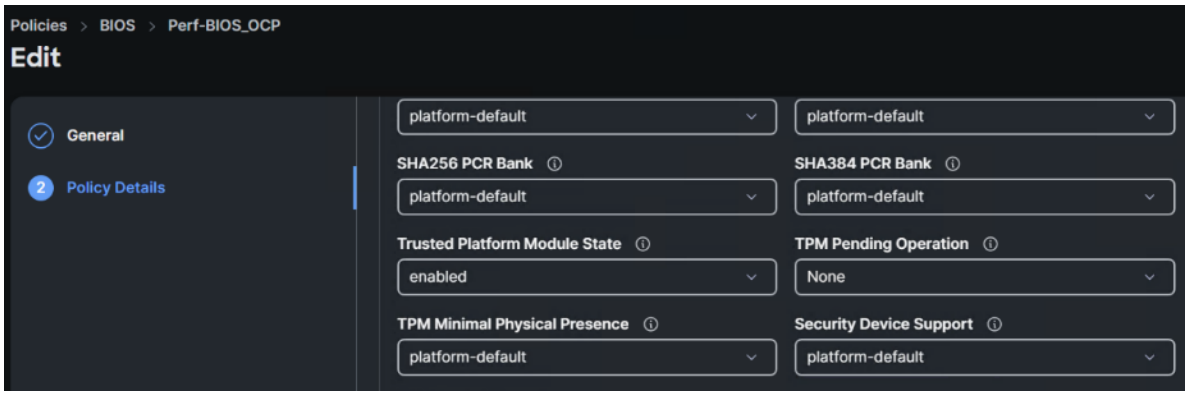
Prerequisites

The FlexPod for OpenShift solution utilizes the Agent-Based Installer (ABI) for deploying OpenShift. As a result, when provisioning and managing the FlexPod infrastructure, you must supply all supporting cluster infrastructure components and resources. These include the installer VM or host, networking, storage, and the individual cluster machines required for deployment.

The following supporting cluster resources are required for installation:

- Control plane and compute machines that form the OpenShift cluster
- Cluster networking, including required VLANs, IP addressing, and connectivity
- Storage for both cluster infrastructure components and application workloads
- Installer VM or host, configured with FIPS mode enabled

Additionally, **TPM** must be enabled at the BIOS level on each server. In Cisco Intersight, this is achieved by configuring the BIOS policy to enable TPM.



Note: If there are any leftover TPM encryption keys from a previous operation system on the server, the cluster deployment can get stuck. To avoid this situation, use TPMClear to reset TPM state. Once the state has been reset following a reboot, change the setting to 'None' to avoid resetting the TPM in future reboots.

Network Requirements

The following infrastructure services need to be deployed to support the OpenShift cluster, during the validation of this solution we have provided VMs on your hypervisor of choice to run the required services. You can use existing DNS and DHCP services available in the data center.

There are various infrastructure services prerequisites for deploying OpenShift 4.19. These prerequisites are as follows:

- DNS and DHCP services – these services were configured on Microsoft Windows Server VMs
- NTP Distribution was done with the Cisco Nexus switches
- Specific DNS entries for deploying OpenShift – added to the DNS server
    - Base Domain: fpmc.sa
    - OpenShift Cluster Name: flexpod-ocp
- A FIPS enabled Linux VM/Physical server for generating installer ISO and cluster management – a Rocky Linux 9 / RHEL 9 VM with appropriate packages

The DNS domain name for the OpenShift cluster should be the cluster name followed by the base domain, for example, flexpod-ocp.fpmc.sa. Table number 9 lists the information for fully qualified domain names used during validation.

Table 9) Hostnames and IP addresses for OpenShift cluster

Usage	Hostname	IP Address
API	api.flexpod-ocp.fpmc.sa	10.61.178.110
Ingress LB (apps)	*. apps.flexpod-ocp.fpmc.sa	10.61.178.111
control-1	control-1.flexpod-ocp.fpmc.sa	10.61.178.101
control-2	control-2.flexpod-ocp.fpmc.sa	10.61.178.102
control-3	control-3.flexpod-ocp.fpmc.sa	10.61.178.103
compute-1	compute-1.flexpod-ocp.fpmc.sa	10.61.178.104
compute-2	compute-2.flexpod-ocp.fpmc.sa	10.61.178.105

Note: Obtain the MAC addresses of the bare metal Interfaces from the UCS Server Profile for each node to be used in the DHCP configuration to assign reserved IP addresses (reservations) to the nodes. The KVM IP address also needs to be gathered for the control-plane and worker nodes from the server profiles.

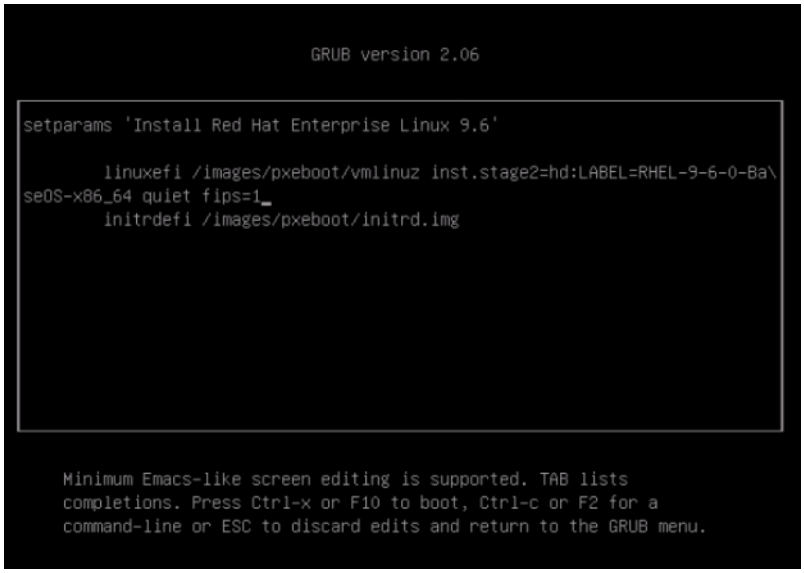
Installer VM

To enable FIPS mode for your OpenShift cluster, you must run the installation program from a Red Hat Enterprise Linux (RHEL) computer configured to operate in FIPS mode. For more information about configuring FIPS mode on RHEL, see Switching RHEL to FIPS mode.

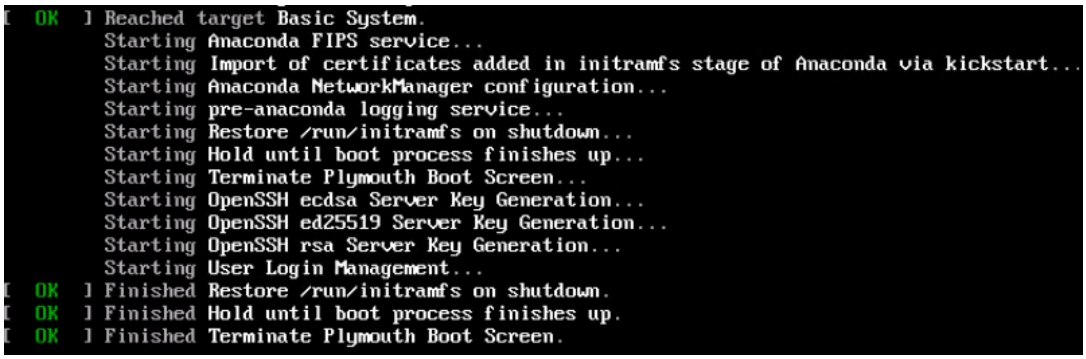
When running Red Hat Enterprise Linux (RHEL) or Red Hat Enterprise Linux CoreOS (RHCOS) booted in FIPS mode, OpenShift Container Platform core components use the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation on only the x86_64, ppc64le, and s390x architectures.

Add Kernel Argument During Installation

1. Boot the RHEL installer (via ISO or PXE).
2. At the **boot menu**, press e to edit the kernel boot parameters.
3. Add fips=1 to the end of the line starting with Linuxefi



#FIPS service started.



#Verify FIPS is enabled on the VM.

```
[root@provisioner ~]# fips-mode-setup --check
FIPS mode is enabled.
[root@provisioner ~]#
```

During a cluster deployment, the Federal Information Processing Standards (FIPS) change is applied when the Red Hat Enterprise Linux CoreOS (RHCOS) machines are deployed in your cluster. To obtain compliance, FIPS mode must be enabled before cluster nodes first boot and should not be disabled once enabled. FIPS cannot be enabled after installation or turned off. Systems built in a non-FIPS state are unsafe for use if FIPS mode is enabled after installation. Such systems should be prevented from entering or otherwise discarded from the cluster.

Note: OpenShift Container Platform requires the use of a FIPS-capable installation binary to install a cluster in FIPS mode.

During the first boot of RHCOS, the system performs several cryptographic operations. Per the FIPS standard, any cryptographic material must be only created and used with FIPS-validated modules. If a non-FIPS system is bootstrapped and then FIPS is enabled, any previously generated cryptographic material must be re-generated which would disrupt the functioning of the cluster.

Agent-based installation is a subcommand of the OpenShift Container Platform installer. It generates a bootable ISO image containing all the information required to deploy an OpenShift Container Platform cluster, with an available release image.

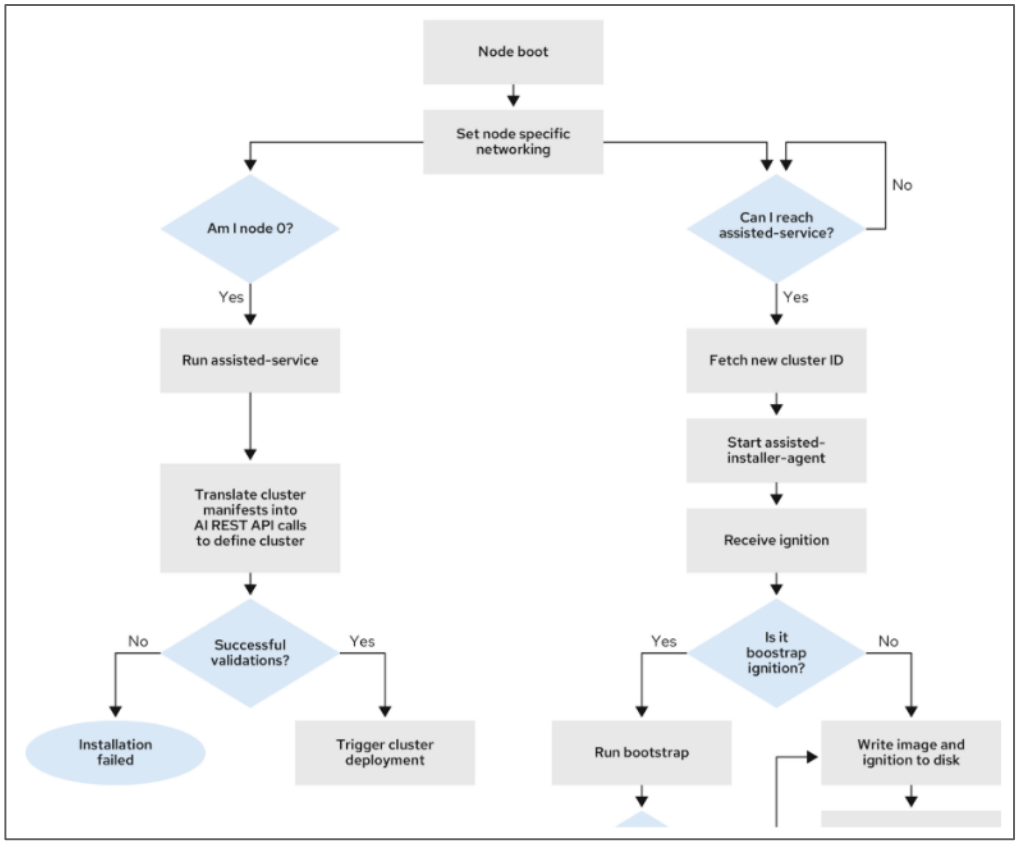
As an OpenShift Container Platform user, you can leverage the advantages of the Assisted Installer hosted service in disconnected environments.

The Agent-based installation comprises a bootable ISO that contains the Assisted discovery agent and the Assisted Service. Both are required to perform the cluster installation, but the latter runs on only one of the hosts. The openshift-install-fips agent create image subcommand generates an ephemeral ISO based on the inputs that you provide. Figure 13) Node installation workflow illustrates the OpenShift nodes installation workflow.

You can choose to provide inputs through the following manifests:

- Install-config.yaml
- Agent-config.yaml

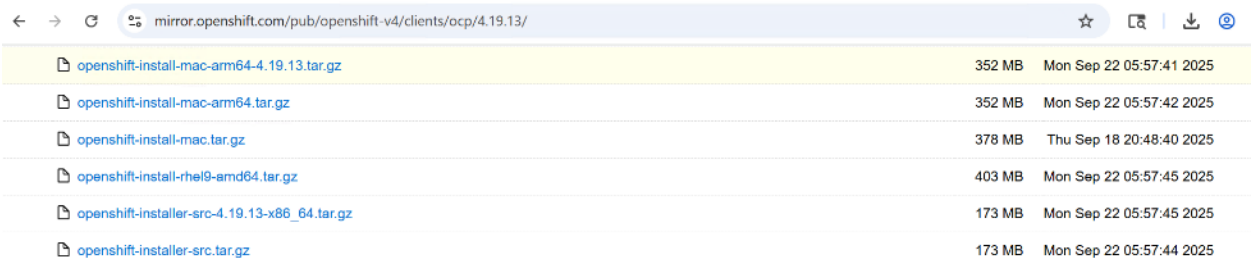
Figure 13) Node installation workflow



As we discussed, OpenShift Container Platform requires the use of a FIPS-capable installation binary to install a cluster in FIPS mode, you can obtain this binary by downloading it from the public OpenShift mirror.

1. Go to OpenShift mirror registry to download OpenShift FIPS based installer.

https://mirror.openshift.com/pub/openshift-v4/clients/ocp/4.19.13/



2. Download openshift-install-rhel9-amd64.tar.g and extract it

```
[admin@sec-rhel-9 ocp-sec]$ tar -xvzf openshift-install-rhel9-amd64.tar.gz


[admin@sec-rhel-9 ocp-sec]$ cp openshift-install-fips /usr/local/bin/openshift-install-fips
```

3. Create Install-config.yaml file.

```
apiVersion: v1
baseDomain: fpmc.sa
metadata:
  name: flexpod-ocp

compute:
- name: worker
  replicas: 2
  platform:
    baremetal: {}

controlPlane:
  name: master
  replicas: 3
  platform:
    baremetal: {}

platform:
  baremetal:
    apiVIP: 10.61.178.110
    ingressVIP: 10.61.178.111
    hosts:
      - name: control-1
        role: master
        bmc:
          address: redfish+http://172.22.14.225
          username: ocp-bmc
          password: ********
        bootMACAddress: 00:25:b5:02:1a:22
        rootDeviceHints:
          deviceName: /dev/sda
      - name: control-2
        role: master
        bmc:
          address: redfish+http://172.22.14.228
          username: ocp-bmc
          password: ********
        bootMACAddress: 00:25:b5:02:1a:2e
        rootDeviceHints:
          deviceName: /dev/sda
      - name: control-3
        role: master
        bmc:
          address: redfish+http://172.22.14.227
          username: ocp-bmc
          password: ********
```

```
        bootMACAddress: 00:25:b5:02:1a:2b
        rootDeviceHints:
          deviceName: /dev/sda
      - name: compute-1
        role: worker
        bmc:
          address: redfish+http://172.22.14.226
          username: ocp-bmc
          password: ********
        bootMACAddress: 00:25:b5:02:1a:31
        rootDeviceHints:
          deviceName: /dev/sda
      - name: compute-2
        role: worker
        bmc:
          address: redfish+http://172.22.14.229
          username: ocp-bmc
          password: ********
        bootMACAddress: 00:25:b5:02:1a:34
        rootDeviceHints:
          deviceName: /dev/sda

networking:
  clusterNetwork:
  - cidr: 10.128.0.0/16
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.61.178.0/24
  networkType: OVNKubernetes
  serviceNetwork:
  - 172.30.0.0/16

fips: true  # Enable FIPS mode

pullSecret:
'{"auths":{"cloud.openshift.com":{"auth**********************************************************
**********************************************************************************************
**********************************************************************************************
**********************************************************************************************
**********************************************************************************************
**********************************"email":"************@netapp.com"}}}'

sshKey: ecdsa-sha2-nistp256
**************************************/+Uy0nFdJSg4GD2imvYcKmjaZ1tFSBuhPFa+HQecAb+ILoY5ytK1yxd
uFRRzKfhldw+yfs7t4yVY10Y0rtuU= admin@sec-rhel-9.5
```

4. Create agent-config.yaml, you must specify rendezvous IP.

```
apiVersion: v1alpha1
kind: AgentConfig
metadata:
  name: flexpod-ocp
rendezvousIP: 10.61.178.101  #control-1 node IP
```

Note: The **rendezvous IP** must be known at the time of generating the agent ISO, so that during the initial boot all the hosts can check in to the assisted service. If the IP addresses are assigned using a Dynamic Host Configuration Protocol (DHCP) server, then the rendezvousIP field must be set to an IP address of one of the hosts that will become part of the deployed control plane. In an environment without a DHCP server, you can define IP addresses statically.

5. Generate ZTP cluster manifests.

```
[admin@sec-rhel-9 ocp-sec]$ openshift-install-fips agent create cluster-manifests --dir flexpod-
ocp/

INFO Configuration has 3 master replicas and 2 worker replicas
INFO Consuming Install Config from target directory
INFO Consuming Agent Config from target directory
INFO Cluster-Manifests created in: flexpod-ocp/cluster-manifests and flexpod-ocp/mirror
```

Note: The install-config.yaml and agent-config.yaml files are deleted and replaced by the cluster manifests generated through the above command. Any configurations made to the install-config.yaml and agent-config.yaml files are imported to the ZTP cluster manifests when you run the openshift-install agent create cluster-manifests command.

6. Navigate to cluster-manifest directory and add the highlighted section to the agent-cluster-install.yaml file.

```
[admin@sec-rhel-9 cluster-manifests]$ vi agent-cluster-install.yaml
  creationTimestamp: null
  name: flexpod-ocp
spec:
  apiVIP: 10.61.178.110
  apiVIPs:
  - 10.61.178.110
  clusterDeploymentRef:
    name: flexpod-ocp
  imageSetRef:
    name: openshift-4.19.13
  ingressVIP: 10.61.178.111
  ingressVIPs:
  - 10.61.178.111
  networking:
    clusterNetwork:
    - cidr: 10.128.0.0/16
      hostPrefix: 23
    machineNetwork:
    - cidr: 10.61.178.0/24
    networkType: OVNKubernetes
    serviceNetwork:
    - 172.30.0.0/16
    userManagedNetworking: false
  platformType: BareMetal
  provisionRequirements:
    controlPlaneAgents: 3
    workerAgents: 2
  diskEncryption:
    enableOn: all
    mode: tpmv2
  sshPublicKey: ecdsa-sha2-nistp256
****************************************/+Uy0nFdJSg4GD2imvYcKmjaZ1tFSBuhPFa+HQecAb+ILoY5ytK1yxd
uFRRzKfhldw+yfs7t4yVY10Y0rtuU= admin@sec-rhel-9.5
status:
  debugInfo:
    eventsURL: ""
    logsURL: ""
  progress:
    totalPercentage: 0
```

Note: Specify which disk encryption mode to use. Valid values are tpmv2 and tang. If you are using tang, specify the Tang servers.

Note: In addition to enabling FIPS mode, TPM 2.0–based disk encryption is applied to the M.2 boot drives. This ensures that the boot media is encrypted for enhanced security. Red Hat CoreOS (RHCOS) was booted from the server's local M.2 drives

Note: In OpenShift Container Platform, you can specify a requirement for more than one Tang server. You can also configure the TPM v2 and Tang encryption modes simultaneously. This enables boot disk data decryption only if the TPM secure cryptoprocessor is present and the Tang servers are accessible over a secure network.

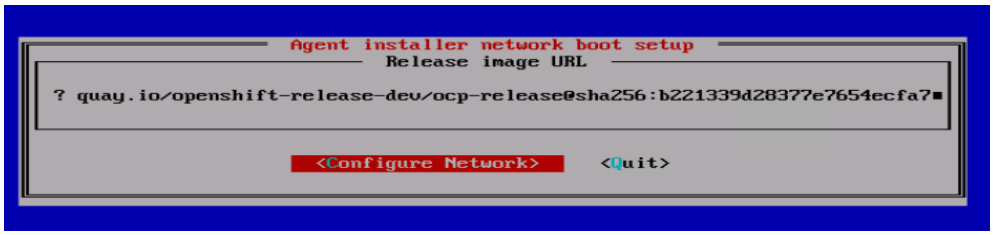
7. Now create the agent ISO using the cluster manifest.

```
admin@sec-rhel-9 ocp-sec]$ openshift-install-fips --dir flexpod-ocp/ agent create image
INFO The rendezvous host IP (node0 IP) is 10.61.178.101
INFO Extracting base ISO from release payload
INFO Verifying cached file
INFO Using cached Base ISO /root/.cache/agent/image_cache/coreos-x86_64.iso
INFO Consuming ClusterDeployment Config from target directory
```

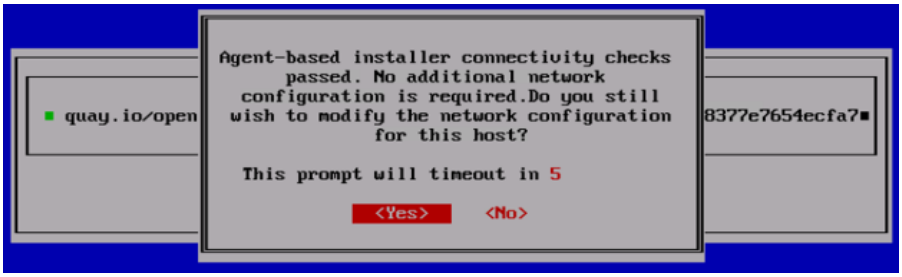
```
INFO Consuming Agent PullSecret from target directory
INFO Consuming Mirror Registries Certificate File from target directory
INFO Consuming Mirror Registries Config from target directory
INFO Consuming InfraEnv Config from target directory
INFO Consuming ClusterImageSet Config from target directory
INFO Consuming AgentClusterInstall Config from target directory
INFO Generated ISO at agent.x86_64.iso.INFO
```

Note: Red Hat Enterprise Linux CoreOS (RHCOS) supports multipathing on the primary disk, allowing stronger resilience to hardware failure to achieve higher host availability. Multipathing is enabled by default in the agent ISO image, with a default /etc/multipath.conf configuration.

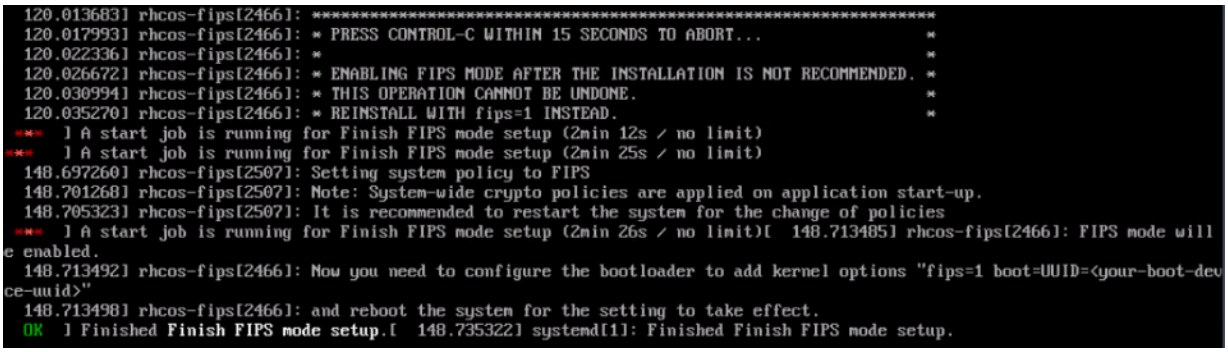
8. After the agent ISO is generated, use the ISO to boot RHCOS on all compute and control nodes.



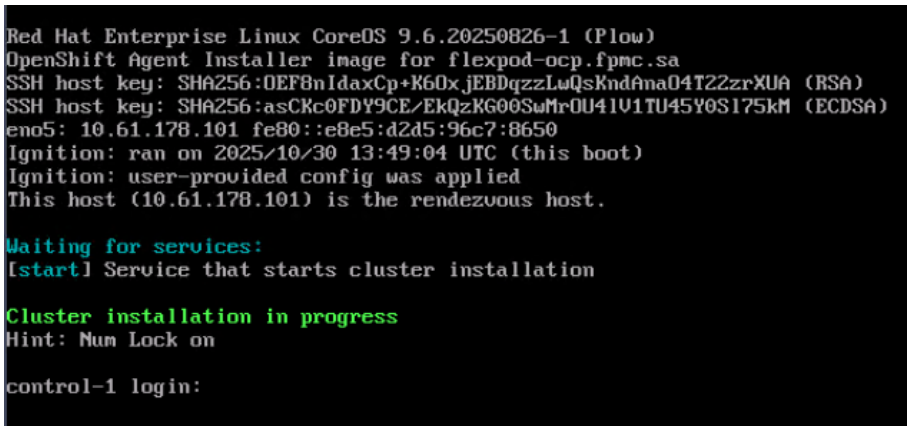
9. The installer will do a network connectivity check.



10. FIPS mode setup will be done on all the nodes during RHCOS installation.



11. OCP cluster setup process will start automatically.



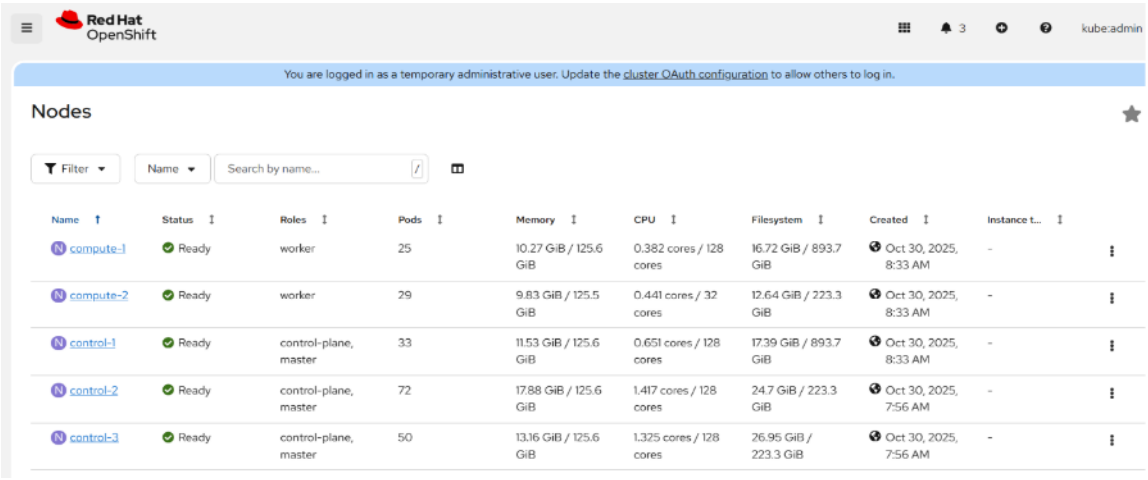
12. Verify the installation progress.

```
[admin@sec-rhel-9 ~]$ openshift-install-fips --dir flexpod-ocp/ agent wait-for bootstrap-complete
DEBUG OpenShift Installer 4.19.13
DEBUG Built from commit 7e30b7d6b421087ee6b6aaa639e40392c22ce52b
DEBUG asset directory: flexpod-ocp
DEBUG Loading Agent Config...
DEBUG Using Agent Config loaded from state file
DEBUG Loading Agent Manifests...
DEBUG   Loading Agent PullSecret...
DEBUG     Loading Agent Workflow...
DEBUG     Using Agent Workflow loaded from state file
DEBUG     Loading Agent Installer ClusterInfo...
DEBUG       Loading Agent Workflow...
DEBUG       Loading AddNodes Config...

*****OutPut Ommited*****
DEBUG RendezvousIP from the AgentConfig 10.61.178.101
DEBUG Loading Agent Installer API Auth Config...
DEBUG   Loading Agent Workflow...
DEBUG   Using Agent Workflow loaded from state file
DEBUG   Loading AddNodes Config...
DEBUG   Using AddNodes Config loaded from state file
DEBUG Using Agent Installer API Auth Config loaded from state file
INFO Bootstrap Kube API Initialized
INFO Bootstrap configMap status is complete
INFO Bootstrap is complete
INFO cluster bootstrap is complete
```

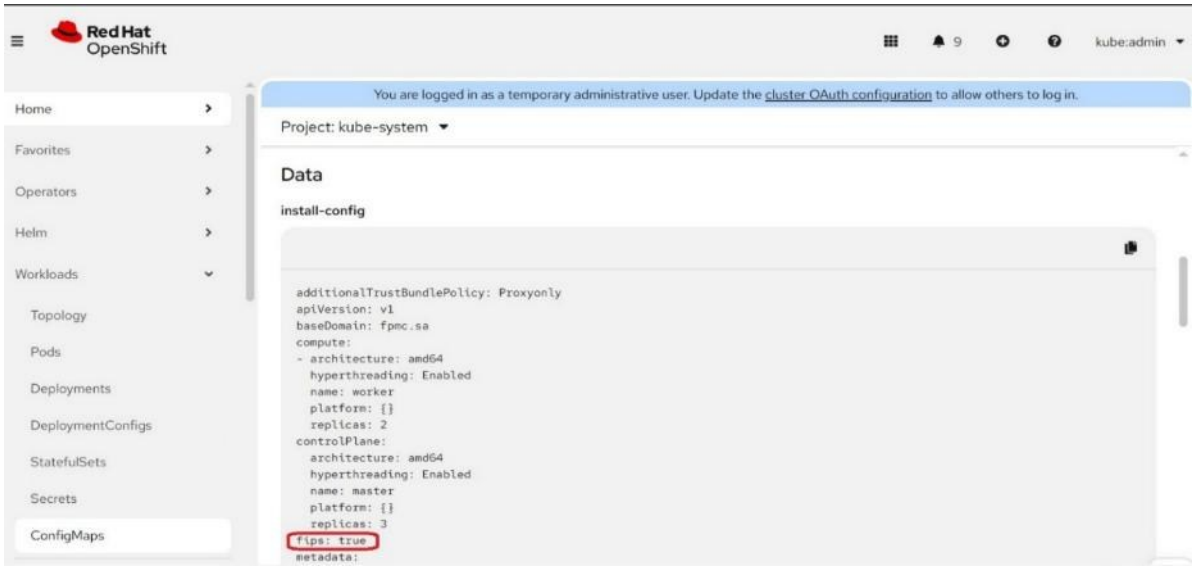
13. Wait for some time and re-verify the installation status.

```
[admin@sec-rhel-9 ~$ openshift-install-fips --dir ocp-sec agent wait-for install-complete

INFO Bootstrap Kube API Initialized
INFO Bootstrap configMap status is complete
INFO Bootstrap is complete
INFO cluster bootstrap is complete

******OUTPUT Ommited*******
INFO Cluster is installed
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run
INFO     export KUBECONFIG=/admin/ocp-sec/flexpod-ocp/auth/kubeconfig
INFO Access the OpenShift web-console here: https://console-openshift-console.apps.flexpod-
ocp.fpmc.sa/
```

14. Login to OpenShift web console using the kubeadmin credential placed in auth directory under flexpod-ocp directory.



15. Verify if FIPS is enabled on the cluster.



16. Log into one of the nodes and verify if boot disk encryption is enabled.

```
[core@compute-1 ~]$ sudo cryptsetup status root
/dev/mapper/root is active and is in use.
  type:    LUKS2
  cipher:  aes-cbc-essiv:sha256
  keysize: 256 bits
  key location: keyring
  device:  /dev/sda4
  sector size:  512
  offset:  32768 sectors
  size:    1874301576 sectors
  mode:    read/write
```

Note: The encryption format. When the TPM v2 or Tang encryption modes are enabled, the RHCOS boot disks are encrypted using the LUKS2 format.

Note: The encryption algorithm used to encrypt the LUKS2 volume. The aes-cbc-essiv:sha256 cipher is used if FIPS mode is enabled.

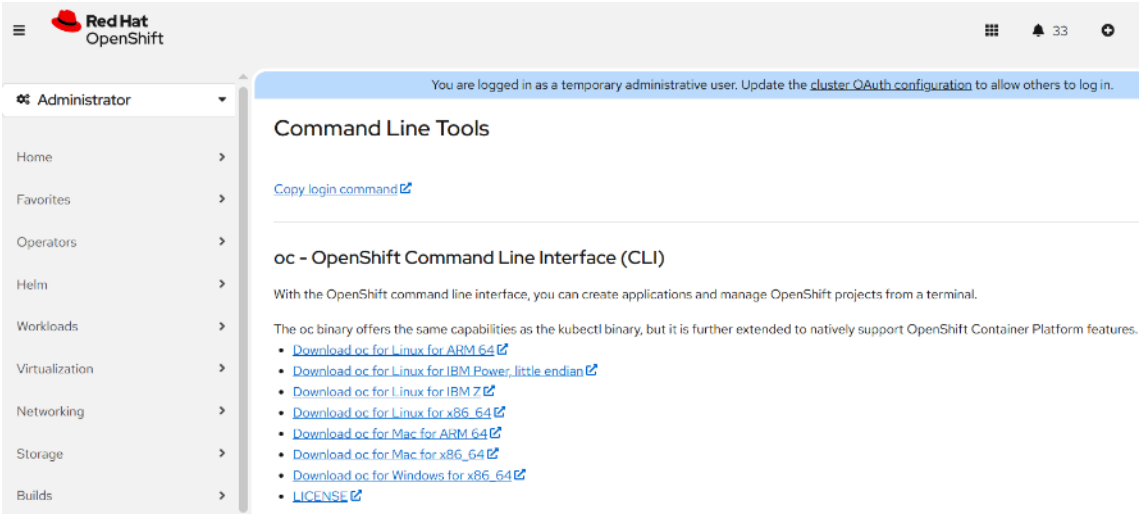
17. Also verify TPM based encryption.

```
[core@control-1 ~]$ sudo clevis luks list -d /dev/sda4
1: sss '{"t":1,"pins":{"tpm2":[{"hash":"sha256","key":"ecc"}]}}'
[core@control-1 ~]$
[core@control-1 ~]$
[core@control-1 ~]$ sudo tpm2_getcap properties-fixed | grep -A2 TPM2_PT_FAMILY_INDICATOR
TPM2_PT_FAMILY_INDICATOR:
  raw: 0x322E3000
  value: "2.0"
[core@control-1 ~]$
```

Post installation configuration

Download oc for Linux for x86_64.

Download OpenShift command line interface tool on your workstation vm.



NTP configuration

It is essential to configure NTP on OpenShift control-plane and compute nodes.

1. Follow the steps below to assign time server to the OpenShift cluster using MachineConfig files for both control and compute nodes.

```
curl https://mirror.openshift.com/pub/openshift-v4/clients/butane/latest/butane --output butane
chmod +x butane
```

2. Build the following files.

```
cat 99-control-plane-chrony-conf-override.bu
variant: openshift
version: 4.19.0
metadata:
  name: 99-control-plane-chrony-conf-override
  labels:
    machineconfiguration.openshift.io/role: master
storage:
  files:
    - path: /etc/chrony.conf
      mode: 0644
      overwrite: true
      contents:
        inline: |
          driftfile /var/lib/chrony/drift
          makestep 1.0 3
```

```
          rtcsync
          logdir /var/log/chrony
          server 10.61.176.251 iburst
          server 10.61.176.252 iburst

---
cat 99-worker-chrony-conf-override.bu
variant: openshift
version: 4.19.0
metadata:
  name: 99-worker-chrony-conf-override
  labels:
    machineconfiguration.openshift.io/role: worker
storage:
  files:
    - path: /etc/chrony.conf
      mode: 0644
      overwrite: true
      contents:
        inline: |
          driftfile /var/lib/chrony/drift
          makestep 1.0 3
          rtcsync
          logdir /var/log/chrony
          server 10.61.176.251 iburst
          server 10.61.176.252 iburst
```

3. Create .yaml files from the butane files with butane, then load the configurations into OpenShift.

```
./butane 99-control-plane-chrony-conf-override.bu -o ./99-control-plane-chrony-conf-override.yaml
./butane 99-worker-chrony-conf-override.bu -o ./99-worker-chrony-conf-override.yaml


oc create -f 99-control-plane-chrony-conf-override.yaml
oc create -f 99-worker-chrony-conf-override.yaml
```

RedHat OpenShift login banners

1. Create a Custom HTML File

```
POD=$(oc get pods -n openshift-authentication -o name | head -n 1)

oc exec -n openshift-authentication "$POD" -- cat /var/config/system/secrets/v4-0-config-system-
ocp-branding-template/login.html > login.html
```

2. Edit login.html, In the body of the document, under **div class="pf-v6-c-login__main-body**, add below message and save.

```
<p>
   This system is for use by authorized users only. Unauthorized access is prohibited.
</p>
```

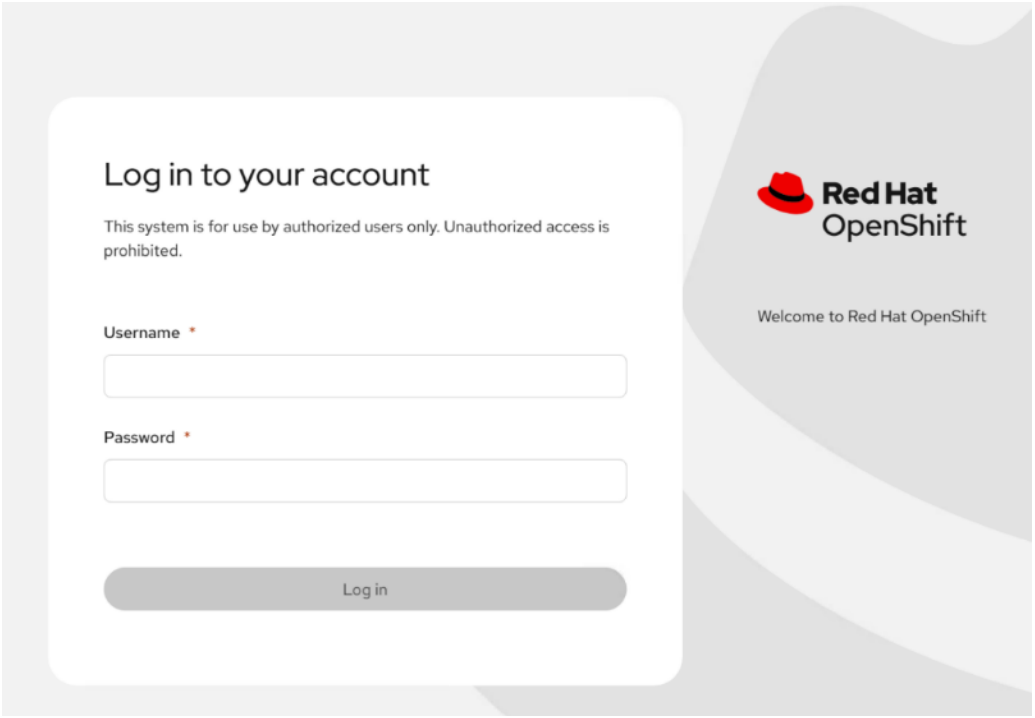
3. To apply the customizations to OpenShift, create a secret in openshift-config namespace

```
oc create secret generic login-template --from-file=login.html -n openshift-config
```

4. Then patch the OAuth Resource to Use the Secret

```
oc patch oauths cluster --type=json -p='[{
 "op": "add",
 "path": "/spec/templates",
 "value": {
   "login": {"name": "login-template"}
 }
}]'``
```

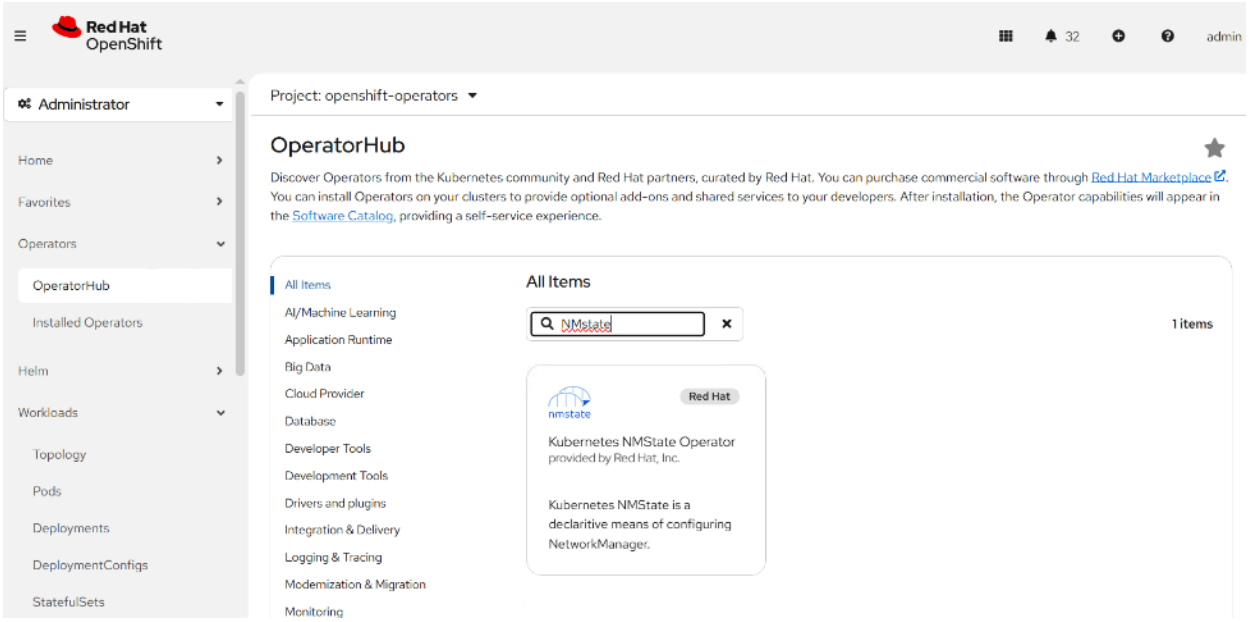
5. Logout and login again. The login banner is displayed just before the authentication step.



OpenShift network configuration

In the Red Hat OpenShift Bare Metal environment, the NMState Operator was used to set up and maintain OpenShift server networking. At later stage with OpenShift Virtualization, NMState was also used to create network bridges on vNICs added to the UCS Server Profiles for OpenShift Virtualization. Network Attachment Definitions (NADs) with specified VLAN tags and MTUs are then created on top of the bridges and attach VMs to the network.

In the OpenShift Console, go to Operators > OperatorHub. In the search box, enter NMState and Kubernetes NMState Operator should appear. Click Kubernetes NMState Operator.



NetApp Trident

Trident overview

Trident is a fully supported open-source project maintained by NetApp. It has been designed to help you meet your containerized application's persistence demands using industry-standard interfaces, such as the Container Storage Interface (CSI).

NetApp Trident enables consumption and management of storage resources across all popular NetApp storage platforms, in the public cloud or on premises, including on-premises ONTAP clusters (AFF, FAS, AFX and ASA), ONTAP Select, Cloud Volumes ONTAP, Element software, Azure NetApp Files, Amazon FSx for NetApp ONTAP, and Google Cloud NetApp Volumes.

Trident is a Container Storage Interface (CSI) compliant dynamic storage orchestrator that natively integrates with Kubernetes. Trident runs as a single Controller Pod plus a Node Pod on each worker node in the cluster. Refer to Trident architecture for details.

To ensure Trident can be installed in a wide variety of environments and organizations, NetApp offers multiple installation options. You can install Trident using the Trident operator (manually or using Helm) or with tridentctl. In Red Hat OpenShift, you can install NetApp Trident using the Red Hat certified operator. In this solution we installed Trident from the Red Hat OpenShift Container Platform.

Best Practice for Trident

Deploy to a dedicated namespace

It is important to prevent applications, application administrators, users, and management applications from accessing Trident object definitions or the pods to ensure reliable storage and block potential malicious activity.

To separate the other applications and users from Trident, always install Trident in its own Kubernetes namespace (trident). Putting Trident in its own namespace assures that only the Kubernetes administrative personnel have access to the Trident pod and the artifacts (such as backend and CHAP secrets if applicable) stored in the namespaced CRD objects.
You should ensure that you allow only administrators access to the Trident namespace and thus access to the tridentctl application.

Limit storage resource access to OpenShift cluster members

Limiting access to the NFS volumes, iSCSI LUNs, and FC LUNs created by Trident is a critical component of the security posture for your Kubernetes deployment. Doing so prevents hosts that are not a part of the Kubernetes cluster from accessing the volumes and potentially modifying data unexpectedly.

It's important to understand that namespaces are the logical boundary for resources in Kubernetes. The assumption is that resources in the same namespace are able to be shared, however, importantly, there is no cross-namespace capability. This means that even though PVs are global objects, when bound to a PVC they are only accessible by pods which are in the same namespace. It is critical to ensure that namespaces are used to provide separation when appropriate.

The primary concern for most organizations regarding data security in a Kubernetes context is that a process in a container can access storage mounted to the host, but which is not intended for the container. Namespaces are designed to prevent this type of compromise. However, there is one exception: privileged containers.

A privileged container is one that is run with substantially more host-level permissions than normal. These are not denied by default, so ensure that you disable the capability by using pod security policies.

Use quotas and range limits to control storage consumption

Kubernetes has two features which, when combined, provide a powerful mechanism for limiting the resource consumption by applications. The storage quota mechanism enables the administrator to implement global, and storage class specific, capacity and object count consumption limits on a per-namespace basis. Further, using a range limit ensures that the PVC requests are within both a minimum and maximum value before the request is forwarded to the provisioner.

These values are defined on a per-namespace basis, which means that each namespace should have values defined which fall in line with their resource requirements

Use a dedicated export policy

You should ensure that an export policy exists for each backend that only allows access to the nodes present in the Kubernetes cluster. Trident can automatically create and manage export policies. This way, Trident limits access to the volumes it provisions to the nodes in the Kubernetes cluster and simplifies the addition/deletion of nodes.

Alternatively, you can also create an export policy manually and populate it with one or more export rules that process each node access request:

- Use the vserver export-policy create ONTAP CLI command to create the export policy.
- Add rules to the export policy by using the vserver export-policy rule create ONTAP CLI command.

Running these commands enables you to restrict which Kubernetes nodes have access to the data.

Use CHAP authentication with ONTAP SAN backends

Trident supports CHAP-based authentication for ONTAP SAN workloads (using the ontap-san and ontap-san-economy drivers). NetApp recommends using bidirectional CHAP with Trident for authentication between a host and the storage backend.

For ONTAP backends that use the SAN storage drivers, Trident can set up bidirectional CHAP and manage CHAP usernames and secrets through tridentctl.

Note: MD5 algo is not support on FIPS enabled OpenShift cluster.

Use Trident with NVE and NAE

NetApp ONTAP provides data-at-rest encryption to protect sensitive data in the event a disk is stolen, returned, or repurposed. For details, refer to Configure NetApp Volume Encryption overview.

- If NAE is enabled on the backend, any volume provisioned in Trident will be NAE-enabled.
    - You can set the NVE encryption flag to "" to create NAE-enabled volumes.
- If NAE is not enabled on the backend, any volume provisioned in Trident will be NVE-enabled unless the NVE encryption flag is set to false (the default value) in the backend configuration.
- You can manually create an NVE volume in Trident by explicitly setting the NVE encryption flag to true.

Note: Volumes created in Trident on an NAE-enabled backend must be NVE or NAE encrypted.

Note: You can set the NVE encryption flag to true in the Trident backend configuration to override the NAE encryption and use a specific encryption key on a per volume basis.

Note: Setting the NVE encryption flag to false on an NAE-enabled backend creates an NAE-enabled volume. You cannot disable NAE encryption by setting the NVE encryption flag to false

Linux Unified Key Setup (LUKS)

You can enable Linux Unified Key Setup (LUKS) to encrypt ONTAP SAN and ONTAP SAN ECONOMY volumes on Trident. Trident supports passphrase rotation and volume expansion for LUKS-encrypted volumes.

In Trident, LUKS-encrypted volumes use the aes-xts-plain64 cypher and mode, as recommended by NIST. For more info about LUKS, refer here.

Kerberos in-flight encryption

Using Kerberos in-flight encryption, you can improve data access security by enabling encryption for the traffic between your managed cluster and the storage backend.

Trident supports Kerberos encryption for ONTAP as a storage backend:

- **On-premise ONTAP** - Trident supports Kerberos encryption over NFSv3 and NFSv4 connections from Red Hat OpenShift and upstream Kubernetes clusters to on-premise ONTAP volumes.

You can create, delete, resize, snapshot, clone, read-only clone, and import volumes that use NFS encryption.

Note: Kerberos encryption for NFS traffic with on-premise ONTAP storage backends is only supported using the ontap-nas storage driver.

For more information on Kerberos, refer here.

ONTAP backend drivers

ONTAP backend drivers are differentiated by the protocol used and how the volumes are provisioned on the storage system. Therefore, give careful consideration when deciding which driver to deploy.

At a higher level, if your application has components which need shared storage (multiple pods accessing the same PVC), NAS-based drivers would be the default choice, while the block-based iSCSI drivers meet the needs of non-shared storage. Choose the protocol based on the requirements of the application and the comfort level of the storage and infrastructure teams. There is little difference between them for most applications, so often the decision is based upon whether or not shared storage (where more than one pod will need simultaneous access) is needed.

The available ONTAP backend drivers are:

- ontap-nas: Each PV provisioned is a full ONTAP FlexVolume.
- ontap-nas-economy: Each PV provisioned is a qtree, with a configurable number of qtrees per FlexVolume (default is 200).
- ontap-nas-flexgroup: Each PV provisioned as a full ONTAP FlexGroup, and all aggregates assigned to a SVM are used.
- ontap-san: Each PV provisioned is a LUN within its own FlexVolume.
- ontap-san-economy: Each PV provisioned is a LUN, with a configurable number of LUNs per FlexVolume (default is 100).

In this solution we will configure ontap-nas and ontap-san backend. Manually Trident backend can be configured using tridentctl tool or by defining TridentBackendConfig Custom Resource Definition (CRD) using oc or kubectl.

Configure Backend

Trident offers two modes of authenticating an ONTAP backend.

- Credential-based: This mode requires sufficient permissions to the ONTAP backend. It is recommended to use an account associated with a pre-defined security login role, such as admin or vsadmin to ensure maximum compatibility with ONTAP versions.
- Certificate-based: This mode requires a certificate installed on the backend for Trident to communicate with an ONTAP cluster. Here, the backend definition must contain Base64-encoded values of the client certificate, key, and the trusted CA certificate if used (recommended).

You can update existing backends to move between credential-based and certificate-based methods. However, only one authentication method is supported at a time. To switch to a different authentication method, you must remove the existing method from the backend configuration.

ONTAP SAN Backend

Prepare to configure a backend with ONTAP SAN drivers. We will use TridentBackendConfig(TBC) to define and manage a storage backend using YAML. It is a Kubernetes Custom Resource Definition (CRD) used by Trident.

Sample iSCSI backend configuration with CHAP authentication.

1. Create a secret using Base64 encoded credentials.

```
cat encoded-secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: ocp-iscsi-secret
  namespace: trident
type: Opaque
data:
  username: YWRtaW4=
  password: TmV0QXBwITIz
  chapInitiatorSecret: Y2w5cXhJbTM2REt5YXd4eQ==
  chapTargetInitiatorSecret: cnF4aWdYZ2tlc0lwd3h5eg==
  chapTargetUsername: aUpGNGhlQlJUMFRDd3h5eg==
  chapUsername: dWgyYU5DTFNkNmNOd3h5eg==
```

Note: It is recommended to create ONTAP cluster role with minimum privileges so that you do not have to use the ONTAP admin role to perform operations in Trident.

Note: When you include the username in a Trident backend configuration, Trident uses the ONTAP cluster role you created to perform the operations.

2. Apply the secret yaml

```
oc apply -f ontap-san-secret.yaml
```

3. Create TridentBackendConfig (TBC) file.

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: ocp-iscsi-backend-tbc
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-san
  backendName: ocp-iscsi-backend

  # Management connectivity to the SVM
  managementLIF: 172.22.14.50
  svm: ocp-svm.fpmc.sa

  sanType: iscsi
  useREST: true

  # Pull ONTAP credentials from the Secret (no plaintext here)
  credentials:
    name: ocp-iscsi-secret
  useCHAP: true
  defaults:
    encryption: "true"     # ENABLES NVE on ONTAP
    nameTemplate:
"{{.config.StoragePrefix}}_{{.config.BackendName}}_{{.volume.Namespace}}_{{.volume.RequestName}}"
```

4. Apply TBC yaml file.

```
[admin@sec-rhel-9 iscsi-test]$ oc apply -f tbc-iscsi-backend.yaml
tridentbackendconfig.trident.netapp.io/ocp-iscsi-backend-tbc created
```

5. Check the backend status.

```
[admin@sec-rhel-9 iscsi-test]$ oc -n trident get tbc ocp-iscsi-backend-tbc -o yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
{"apiVersion":"trident.netapp.io/v1","kind":"TridentBackendConfig","metadata":{"annotations":{},"
name":"ocp-iscsi-backend-tbc","namespace":"trident"},"spec":{"backendName":"ocp-iscsi-
backend","credentials":{"name":"ocp-iscsi-
secret"},"managementLIF":"172.22.14.50","sanType":"iscsi","storageDriverName":"ontap-
san","svm":"ocp-svm.fpmc.sa","useCHAP":true,"useREST":true,"version":1}}
  creationTimestamp: "2026-01-13T12:34:54Z"
  finalizers:
  - trident.netapp.io
  generation: 1
  name: ocp-iscsi-backend-tbc
  namespace: trident
  resourceVersion: "5139776"
  uid: 960a742d-080b-4c90-a3e7-fb9367ac9382
spec:
  backendName: ocp-iscsi-backend
  credentials:
    name: ocp-iscsi-secret
  managementLIF: 172.22.14.50
  sanType: iscsi
  storageDriverName: ontap-san
  svm: ocp-svm.fpmc.sa
  useCHAP: true
  useREST: true
  version: 1
status:
  backendInfo:
    backendName: ocp-iscsi-backend
    backendUUID: 3c4828bd-2409-40da-9bd1-eea52057425a
  deletionPolicy: delete
  lastOperationStatus: Success
  message: Backend 'ocp-iscsi-backend' created
  phase: Bound
```

6. Verify the backend is visible.

```
[admin@sec-rhel-9 iscsi-test]$ oc -n trident get tbe
NAME        BACKEND             BACKEND UUID
tbe-pqvcm   ocp-iscsi-backend   3c4828bd-2409-40da-9bd1-eea52057425a
```

Note: When FIPS is enabled on OpenShift cluster, MD5 algorithm for CHAP will not work.

```
[core@compute-1 ~]# sudo cat /etc/iscsi/iscsid.conf | grep MD5
# Valid values are MD5, SHA1, SHA256, and SHA3-256.
# The default is MD5.
#node.session.auth.chap_algs = SHA3-256,SHA256,SHA1,MD5
```

7. Login to each compute node and enable SHA256 as CHAP algorithm.

```
[core@compute-1 ~]# sudo vi /etc/iscsi/iscsid.conf
---------------------------------------------
# *************
# CHAP Settings
# *************

# To enable CHAP authentication set node.session.auth.authmethod
# to CHAP. The default is None.
#node.session.auth.authmethod = CHAP

# To configure which CHAP algorithms to enable, set
# node.session.auth.chap_algs to a comma separated list.
# The algorithms should be listed in order of decreasing
# preference - in particular, with the most preferred algorithm first.
# Valid values are MD5, SHA1, SHA256, and SHA3-256.
# The default is MD5.
node.session.auth.chap_algs = SHA256
```

Note: Make sure to restart iSCSI service.

#Verify if CHAP is enabled.

```
[admin@sec-rhel-9 ocp-sec]$ oc get tbe tbe-pqvcm -n trident -o yaml | grep useCHAP
    useCHAP: true
[admin@sec-rhel-9 ocp-sec]$
```

#Also verify in ONTAP.

```
fpsa-a90-u1516::> vserver iscsi security show -vserver ocp-svm.fpmc.sa
                                 Auth   Auth CHAP Inbound CHAP  Outbound CHAP
Vserver    Initiator Name        Type   Policy    User Name     User Name
---------- --------------------- ------ --------- ------------ -------------
ocp-svm.fpmc.sa
           default               CHAP   local     uh2aNCLSd6cNwxyz
                                                              iJF4heBRT0TCwxyz
```

Note: When updating the CHAP secrets for a backend, you must use TBC secret to update the backend. Do not update the credentials on the storage cluster using the ONTAP CLI or ONTAP System Manager as Trident will not be able to pick up these changes.

8. Create storage class for iSCSI backend.

```
cat sc-ontap-iscsi.yaml
apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

  name: ontap-iscsi

parameters:

  backendType: "ontap-san"

  sanType: "iscsi"

  provisioningType: "thin"

  snapshots: "true"

allowVolumeExpansion: true

provisioner: csi.trident.netapp.io

mountOptions:
  - discard
```

9. Create a PVC based on iSCSI storage class.

10. Verify the encryption in ONTAP.

```
fpsa-a90::*> vol show -vserver ocp-svm.fpmc.sa -volume
trident_ocp_iscsi_backend_default_test_pvc_iscsi_1bd55 -fields encrypt,encryption-state
vserver         volume                                                  encrypt encryption-state
--------------- ------------------------------------------------------- ------- ----------------
```

```
ocp-svm.fpmc.sa trident_ocp_iscsi_backend_default_test_pvc_iscsi_1bd55 true    full
```

11. Deploy a pod to test iSCSI connection.

```
apiVersion: v1
kind: Pod
metadata:
  name: iscsi-pod
spec:
  containers:
    - name: iscsi-container
      image: registry.access.redhat.com/ubi8/ubi
      command: ["/bin/sh", "-c", "sleep 3600"]     # Keeps the pod running
      volumeMounts:
        - mountPath: /mnt/iscsi
          name: iscsi-volume
  volumes:
    - name: iscsi-volume
      persistentVolumeClaim:
        claimName: test-pvc-iscsi
```

12. Wait for the pod to get status running.

```
[admin@sec-rhel-9 ocp-sec]$ oc get pod -o wide
NAME        READY   STATUS    RESTARTS   AGE   IP             NODE        NOMINATED NODE
READINESS GATES
iscsi-pod   1/1     Running   0          20m   10.128.8.27    compute-2   <none>           <none>
```

13. Now ssh to compute-2 to verify the iSCSI session, also the CHAP should be mentioned.

```
[core@compute-2 ~]$ sudo iscsiadm -m session -P 3
iSCSI Transport Class version 2.0-870
version 6.2.1.9
Target: iqn.1992-08.com.netapp:sn.082972cda33d11f09f85d039eac0cb5f:vs.4
        Current Portal: 192.168.111.52:3260,1048
        Persistent Portal: 192.168.111.52:3260,1048
                **********
                Interface:
                **********
                Iface Name: default
                Iface Transport: tcp
                Iface Initiatorname: iqn.1994-05.com.redhat:9b48bd7596d
                Iface IPaddress: 192.168.111.105
                Iface HWaddress: default
                Iface Netdev: default
                SID: 1
                iSCSI Connection State: LOGGED IN
                iSCSI Session State: LOGGED_IN
                Internal iscsid Session State: NO CHANGE
                *********
                Timeouts:
                *********
                Recovery Timeout: 5
                Target Reset Timeout: 30
                LUN Reset Timeout: 30
                Abort Timeout: 15
                *****
                CHAP:
                *****
                username: uh2aNCLSd6cNwxyz
                password: ********
                username_in: iJF4heBRT0TCwxyz
                password_in: ********
                ************************
                Negotiated iSCSI params:
                ************************
                HeaderDigest: None
                DataDigest: None
                MaxRecvDataSegmentLength: 262144
                MaxXmitDataSegmentLength: 65536
                FirstBurstLength: 65536
                MaxBurstLength: 1048576
```

ONTAP NAS Backend

New and existing backends can use a certificate and communicate with the ONTAP backend. Three parameters are required in the backend definition.

- clientCertificate: Base64-encoded value of client certificate.

- clientPrivateKey: Base64-encoded value of associated private key.
- trustedCACertificate: Base64-encoded value of trusted CA certificate. If using a trusted CA, this parameter must be provided. This can be ignored if no trusted CA is used.

Note: Trusted CA was not used while configuring the backend.

1. Generate a client certificate and key.

```
openssl req -x509 -nodes -days 2190 -newkey rsa:4096 -keyout trident.key -out trident.pem -subj
"/C=US/ST=NC/L=RTP/O=NetApp/CN=admin"
```

2. Install the client certificate and key on the ONTAP cluster.

```
fpsa-a90::> security certificate install -type client-ca -cert-name trident-user -vserver ocp-
svm.fpmc.sa

Enter certificate: Press <Enter> when done

fpsa-a90::> security ssl modify -vserver ocp-svm.fpmc.sa  -client-enabled true
```

3. Create cert-based authentication method.

```
fpsa-a90::> security login create -user-or-group-name trident-user -application ontapi -
authentication-method cert

fpsa-a90::> security login create -user-or-group-name trident-user -application http -
authentication-method cert
```

4. Encode the certificate and key with Base64..

```
base64 -w 0 trident.crt >> cert_base64

base64 -w 0 trident.key >> key_base64
```

5. Create a backend file.

```
cat nfs-cert-backend.yaml
version: 1

storageDriverName: ontap-nas

backendName: ocp-nfs-backend

managementLIF: 172.22.14.50

dataLIF: 172.22.16.51

svm: ocp-svm.fpmc.sa

clientCertificate: LS0tLS1C…….S0tLS0K

clientPrivateKey: LS0tLS……S0tLS0K

useREST: true

defaults:

  spaceReserve: none

  exportPolicy: default

  snapshotPolicy: default

  snapshotReserve: '5'

  encryption: "true"     # ENABLES NVE on ONTAP
```

```
  nameTemplate:
"{{.config.StoragePrefix}}_{{.config.BackendName}}_{{.volume.Namespace}}_{{.volume.RequestName}}"
```

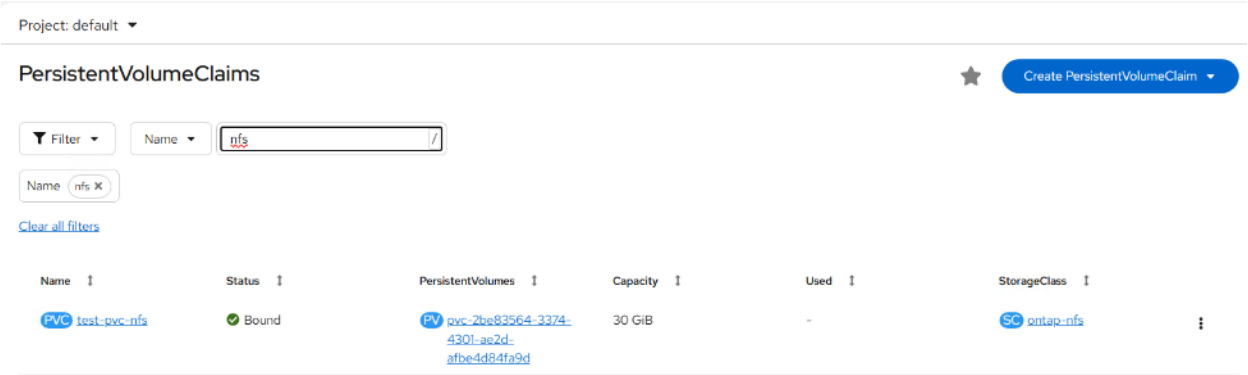
6. Create backend using tridentctl.

```
[admin@sec-rhel-9 ocp-sec]$ tridentctl create backend -f nfs-cert-backend.yaml -n trident
+-------------------+----------------+--------------------------------------+--------+-----------
-+---------+
|       NAME        | STORAGE DRIVER |                 UUID                 | STATE  | USER-STATE
| VOLUMES |
+-------------------+----------------+--------------------------------------+--------+-----------
+---------+
| ocp-nfs-backend   | ontap-nas      | 3b73e35c-bd76-4b2e-8649-2b83449c7698 | online | normal
|       0 |
+-------------------+----------------+--------------------------------------+--------+-----------
+---------+
```

7. Create storageclass based on below yaml file.

```
cat nfs-sc.yaml
apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

  name: ontap-nfs

  annotations:

    storageclass.kubernetes.io/is-default-class: "true"

provisioner: csi.trident.netapp.io

parameters:

  backendType: "ontap-nas"

  provisioningType: "thin"

  snapshots: "true"

allowVolumeExpansion: true
```

8. Create a test nfs pvc.



9. Verify the encryption on the newly created nfs volume.

```
fpsa-a90::> vol show -vserver ocp-svm.fpmc.sa -volume
trident_ocp_nfs_backend_default_test_pvc_nfs_2be83 -fields encrypt,encryption-state
vserver         volume                                              encrypt encryption-state
--------------- --------------------------------------------------- ------- ----------------
ocp-svm.fpmc.sa trident_ocp_nfs_backend_default_test_pvc_nfs_2be83 true    full
```

Trident Protect

You can protect your containerized applications, their associated data, and any virtual machines running on OpenShift by leveraging NetApp Trident Protect. Trident Protect delivers advanced application-aware data management, enhancing the resiliency, availability, and operational efficiency of stateful Kubernetes workloads that rely on NetApp ONTAP storage and the Trident CSI provisioner. It streamlines the end-to-end process of managing, safeguarding, and moving containerized workloads across on-premises and public cloud environments, while also providing robust automation capabilities through its API and CLI.

Trident Protect also provides filesystem freeze and unfreeze capabilities for KubeVirt virtual machines during data protection operations to ensure data consistency. The configuration method and default behavior for VM freeze operations varies across Trident Protect versions, with newer releases offering simplified configuration through Helm chart parameters.

Using Trident Protect, you can use the asynchronous replication capabilities of NetApp SnapMirror technology to replicate data and application changes from one storage backend to another, on the same cluster or between different clusters.

For more information about NetApp Trident Protect and it's advanced application data management capabilities, refer to Trident Protect Documentation.

OpenShift security hardening

Red Hat Enterprise Linux CoreOS Security

Red Hat Enterprise Linux CoreOS (RHCOS) is the operating system base for OpenShift Container Platform (OCP). As a lightweight and purpose-built operating system, it is based on Red Hat Enterprise Linux and uses the same kernel, code, open-source development process, and ships with a specific subset of RHEL software packages.

RHCOS is built and supported for use in OpenShift 4 clusters. Its primary goal is to provide a secure operating system platform for running Kubernetes, OpenShift services, and the containerized workloads running on the aggregated platform.

Add a kernel argument to RHCOS nodes

Some security features need to be done by passing arguments to the kernel when an RHCOS node boots. There are kernel arguments that enable tradeoffs between security and system availability. One example of a kernel-related argument that can have an impact on RHCOS security is pti. Turning on the kernel page-table isolation kernel argument (pti=on) hardens the kernel to prevent bypassing kernel address space layout randomization (KASLR), while mitigating the Meltdown security vulnerability.

Kernel arguments can also be used to disable physical ports, such as Universal Serial Bus (USB) and Firewire (IEEE 1394). Input/output (I/O) devices include, for example, Compact Disk (CD) and Digital Video Disk (DVD) drives. Physically disabling or removing such connection ports and I/O devices help prevent exfiltration of information from information systems and the introduction of malicious code into systems from those ports/devices. The following procedure enables kernel page-table isolation symmetric multi-threading on all worker nodes in a cluster. Substituting master for worker would apply that change to master nodes instead. Once done, the kernel argument is appended to the end of the existing kernel arguments.

To see the current MachineConfigs, type:

```
[admin@sec-rhel-9 ocp-sec]$ oc get machineconfig
NAME                                               GENERATEDBYCONTROLLER
IGNITIONVERSION   AGE
```

```
00-master                                          37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
00-worker                                          37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
01-master-container-runtime                        37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
01-master-kubelet                                  37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
01-worker-container-runtime                        37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
01-worker-kubelet                                  37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
50-masters-chrony-configuration
3.1.0             38d
50-workers-chrony-configuration
3.1.0             38d
97-master-generated-kubelet                        37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
97-worker-generated-kubelet                        37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
98-master-generated-kubelet                        37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
98-worker-generated-kubelet                        37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
99-assisted-installer-master-ssh
3.1.0             38d
99-master-fips
3.2.0             38d
99-master-generated-registries                     37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
99-master-ssh
3.2.0             38d
99-worker-fips
3.2.0             38d
99-worker-generated-registries                     37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
99-worker-ssh
3.5.0             38d
rendered-master-bf60f07134d397ec055a4cdf5119ebd9   37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
rendered-worker-03f3a4402368c0c922fa12aa25454d2b   37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             38d
rendered-worker-5d7cd42fbecc79e4f70081e33802a981   37499cebe4e30aea812eb2a0e0cb5ed9ee961c5f
3.5.0             12s
```

Run below command to verify if pti is enabled or not.

```
[core@compute-2 ~]$ cat /proc/cmdline
BOOT_IMAGE=(hd0,gpt3)/boot/ostree/rhcos-
0507a2ecc42ce08f9f494c33d2099f6418508b795280daace3ce724c8d3b4bdd/vmlinuz-5.14.0-
570.45.1.el9_6.x86_64 ignition.platform.id=metal ignition.firstboot ip=eno5:dhcp
systemd.unified_cgroup_hierarchy=1 cgroup_no_v1=all psi=0
ostree=/ostree/boot.0/rhcos/0507a2ecc42ce08f9f494c33d2099f6418508b795280daace3ce724c8d3b4bdd/0
fips=1 boot=LABEL=boot
```

To enable KPTI, apply the following MachineConfig to add pti=on to the kernel arguments and force it to be enabled. Note that the following example is for nodes with the worker role:

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 20-worker-kargs-force-pti-on
spec:
  config:
    ignition:
      version: 3.5.0
  kernelArguments:
```

```
    - pti=on
```

Apply the machine config.

```
[admin@sec-rhel-9 ocp-sec]$ oc apply -f 20-worker-force-pti-on.yaml
machineconfig.machineconfiguration.openshift.io/99-worker-kargs-force-pti-on created
```

Verify MCP.

```
[admin@sec-rhel-9 ocp-sec]$ oc get mcp
NAME     CONFIG                                             UPDATED   UPDATING   DEGRADED
MACHINECOUNT   READYMACHINECOUNT   UPDATEDMACHINECOUNT   DEGRADEDMACHINECOUNT   AGE
master   rendered-master-bf60f07134d397ec055a4cdf5119ebd9   True      False      False      3
3                  3                   0                      38d
worker   rendered-worker-03f3a4402368c0c922fa12aa25454d2b   False     True       False      2
0                  0                   0                      38d
```

After applying the MachineConfig and waiting for the nodes to reboot, confirm the argument was appended and the feature was enabled:

```
[core@compute-1 ~]$ cat /proc/cmdline

BOOT_IMAGE=(hd0,gpt3)/boot/ostree/rhcos-
0507a2ecc42ce08f9f494c33d2099f6418508b795280daace3ce724c8d3b4bdd/vmlinuz-5.14.0-
570.45.1.el9_6.x86_64 ignition.platform.id=metal ip=eno5:dhcp
ostree=/ostree/boot.0/rhcos/0507a2ecc42ce08f9f494c33d2099f6418508b795280daace3ce724c8d3b4bdd/0
fips=1 boot=LABEL=boot root=UUID=7893794f-89c4-4c9d-8dec-e1f31c04d788 rw rootflags=prjquota
systemd.unified_cgroup_hierarchy=1 cgroup_no_v1=all psi=0 pti=on

[core@compute-1 ~]$ journalctl --no-pager | grep -i "page tables isolation"
Dec 08 12:56:39 localhost kernel: Kernel/User page tables isolation: force enabled on command
line.
Dec 08 12:56:39 localhost kernel: Kernel/User page tables isolation: enabled
```

Check MCP status on the worker nodes.

```
[admin@sec-rhel-9 ocp-sec]$ oc get mcp
NAME     CONFIG                                             UPDATED   UPDATING   DEGRADED
MACHINECOUNT   READYMACHINECOUNT   UPDATEDMACHINECOUNT   DEGRADEDMACHINECOUNT   AGE
master   rendered-master-bf60f07134d397ec055a4cdf5119ebd9   True      False      False      3
3                  3                   0                      39d
worker   rendered-worker-5d7cd42fbecc79e4f70081e33802a981   True      False      False      2
2                  2                   0                      39d
```

File Integrity Operator

The File Integrity Operator is an OpenShift Container Platform Operator that continually runs file integrity checks on the cluster nodes. It deploys a daemon set that initializes and runs privileged advanced intrusion detection environment (AIDE) containers on each node, providing a status object with a log of files that are modified during the initial run of the daemon set pods.

You can install File Integrity operator through either web console or CLI. We installed using web console Operator Hub.

```
[admin@sec-rhel-9 ~]$ oc get deploy -n openshift-file-integrity

NAME                      READY   UP-TO-DATE   AVAILABLE   AGE
file-integrity-operator   1/1     1            1           91s
```

To enable scans on worker nodes, create a FileIntegrity CR using the below yaml file.

```
apiVersion: fileintegrity.openshift.io/v1alpha1
kind: FileIntegrity
metadata:
  name: worker-fileintegrity
  namespace: openshift-file-integrity
```

```
spec:
  nodeSelector:
      node-role.kubernetes.io/worker: ""
  tolerations:
  - key: "myNode"
    operator: "Exists"
    effect: "NoSchedule"
  config:
    name: "myconfig"
    namespace: "openshift-file-integrity"
    key: "config"
    gracePeriod: 20
    maxBackups: 5
    initialDelay: 60
  debug: false
status:
  phase: Active
```

Apply the YAML file to the openshift-file-integrity namespace.

```
oc apply -f worker-fileintegrity.yaml -n openshift-file-integrity
```

Verify the status.

```
[admin@sec-rhel-9 ~]$ oc get fileintegrities -n openshift-file-integrity

NAME                   AGE
worker-fileintegrity   89s
```

Verify the pod status.

```
[admin@sec-rhel-9 ~]$ oc get pod  -n openshift-file-integrity
NAME                                          READY   STATUS    RESTARTS        AGE
aide-worker-fileintegrity-bk6pc               1/1     Running   0               2m29s
aide-worker-fileintegrity-r7x2p               1/1     Running   0               2m29s
file-integrity-operator-7888959c67-mfxzm      1/1     Running   0               9m43s
```

The scan results of the FileIntegrity CR are reported in another object called FileIntegrityNodeStatuses. Verify the node status.

```
[admin@sec-rhel-9 ~]$ oc get fileintegritynodestatuses -n openshift-file-integrity
NAME                              NODE        STATUS
worker-fileintegrity-compute-1    compute-1   Succeeded
worker-fileintegrity-compute-2    compute-2   Succeeded
```

Verify the FileIntergrity custom resources status.

```
[admin@sec-rhel-9 ~]$ oc get fileintegrities/worker-fileintegrity  -o jsonpath="{ .status.phase
}" -n openshift-file-integrity
Active
```

To trigger a failure scenario, alter one of the files monitored by AIDE. For instance, you can modify /etc/resolv.conf on any of the worker nodes

```
[admin@sec-rhel-9 ~]$ oc debug node/compute-2
Starting pod/compute-2-debug-xp5jr...
To use host binaries, run `chroot /host`
Pod IP: 10.61.178.105
If you don't see a command prompt, try pressing enter.
sh-5.1# echo "# integrity test" >> /host/etc/resolv.conf
sh-5.1# exit
exit

Removing debug pod ...
```

After some time, the Failed condition appears in the results array of the corresponding FileIntegrityNodeStatus object. The previous Succeeded condition is still preserved, enabling you to identify exactly when the check started failing.

```
[admin@sec-rhel-9]$ oc get fileintegritynodestatuses.fileintegrity.openshift.io/worker-
fileintegrity-compute-2 -n openshift-file-integrity -ojsonpath='{.results}' | jq -r
[
  {
    "condition": "Succeeded",
    "lastProbeTime": "2026-01-15T12:01:53Z"
  },
  {
    "condition": "Failed",
    "filesChanged": 1,
    "lastProbeTime": "2026-01-15T12:07:43Z",
    "resultConfigMapName": "aide-worker-fileintegrity-compute-2-failed",
    "resultConfigMapNamespace": "openshift-file-integrity"
  }
]
```

The Failed condition points to a config map that gives more details about what exactly failed.

```
[admin@sec-rhel-9]$ oc get cm -n openshift-file-integrity
NAME                                          DATA   AGE
aide-pause                                    1      50m
aide-reinit                                   1      50m
aide-worker-fileintegrity-compute-2-failed    1      8m40s
kube-root-ca.crt                              1      57m
openshift-service-ca.crt                      1      57m
worker-fileintegrity                          1      50m
```

Run the command to find the reason for failed config map.

```
[root@sec-rhel-9 ocp-sec]# oc describe cm aide-worker-fileintegrity-compute-2-failed -n openshift-file-integrity
Name:         aide-worker-fileintegrity-compute-2-failed
Namespace:    openshift-file-integrity
Labels:       file-integrity.openshift.io/node=compute-2
              file-integrity.openshift.io/owner=worker-fileintegrity
              file-integrity.openshift.io/result-log=
Annotations:  file-integrity.openshift.io/files-added: 0
              file-integrity.openshift.io/files-changed: 1
              file-integrity.openshift.io/files-removed: 0

Data
====
integritylog:
----
gcry_md_enable 1 failed
Start timestamp: 2026-01-15 12:05:34 +0000 (AIDE 0.16)
AIDE found differences between database and filesystem!!

Summary:
  Total number of entries:  38803
  Added entries:            0
  Removed entries:          0
  Changed entries:          1

---------------------------------------------------
Changed entries:
---------------------------------------------------

f   ...    .C... : /hostroot/etc/resolv.conf

---------------------------------------------------
Detailed information about changes:
---------------------------------------------------

File: /hostroot/etc/resolv.conf
  SHA512   : 4s0QGeKSOjTDOSEslx2roqNyuqOzUSPZ | sQq3rjGLsAC+Boy85+JALO0GjgpFPW9P
             hudPrcQaVDnkp4PdMqzAfAvUjr3r8xxa | yhznrul03TM9W3YiExAxI7NFoUBtr11h
             m59xDOQuhPEpsUtAU08GiQ==         | CZDV7AR/wrfm0n5CmLe3BQ==
```

Identity and access management security

To control access to an OpenShift Container Platform cluster, a cluster administrator can configure user authentication and ensure only approved users access the cluster.

To interact with an OpenShift Container Platform cluster, users must first authenticate to the OpenShift Container Platform API in some way. You can authenticate by providing an OAuth access token or an X.509 client certificate in your requests to the OpenShift Container Platform API.

An administrator can configure authentication through the following tasks:

- Configuring an identity provider: You can define any supported identity provider in OpenShift Container Platform and add it to your cluster.
- Configuring the internal OAuth server: The OpenShift Container Platform control plane includes a built-in OAuth server that determines the user's identity from the configured identity provider and creates an access token. You can configure the token duration and inactivity timeout, and customize the internal OAuth server URL.

Note: Users can view and manage OAuth tokens owned by them.

- Registering an OAuth client: OpenShift Container Platform includes several default OAuth clients. You can register and configure additional OAuth clients.

Note: When users send a request for an OAuth token, they must specify either a default or custom OAuth client that receives and uses the token.

- Managing cloud provider credentials using the Cloud Credentials Operator: Cluster components use cloud provider credentials to get permissions required to perform cluster-related tasks.
- Impersonating a system admin user: You can grant cluster administrator permissions to a user by impersonating a system admin user.

Users

When it comes to direct use and management of an OpenShift cluster, there are regular users (who typically run workloads and may do some administration) and system users (who can interact with the API). The kubeadmin user is the first user created on an OpenShift cluster and requires special attention since it holds superuser privileges. User accounts that function behind the scenes include service accounts and virtual system users.

Table 10) Types of users

User Type	Description
Regular users	This is the way most interactive OpenShift Container Platform users are represented. Regular users are created automatically in the system upon first login or can be created via the API. Regular users are represented with the User object.
System users	Many of these are created automatically when the infrastructure is defined, mainly for the purpose of enabling the infrastructure to interact with the API securely. They include a cluster administrator (with access to everything), a per-node user, users for use by routers and registries, and various others. Finally, there is an anonymous system user that is used by default for unauthenticated requests. Examples: system:admin system:openshift-registry system:node:node1.example.com
Service accounts	These are special system users associated with projects; some are created automatically when the project is first created, while project administrators can create more for the purpose of defining access to the contents of each project. Service accounts are represented with

	the ServiceAccount object. Examples: system:serviceaccount:default:deployer syste m:serviceaccount:foo:builder

Groups

A user can be assigned to one or more groups, each of which represents a certain set of users. Groups are useful when managing authorization policies to grant permissions to multiple users at once, for example allowing access to objects within a project, versus granting them to users individually. In addition to explicitly defined groups, there are also system groups, or virtual groups, that are automatically provisioned by the cluster.

The following default virtual groups are most important:

Table 11) Default groups in OpenShift

Virtual group	Description
system:authenticated	Automatically associated with all authenticated users.
system:authenticated:oauth	Automatically associated with all users authenticated with an OAuth access token.
system:unauthenticated	Automatically associated with all unauthenticated users.

API authentication

Requests to the OpenShift Container Platform API are authenticated using the following methods:

OAuth access tokens

- Obtained from the OpenShift Container Platform OAuth server using the <*namespace_route*>/oauth/authorize and <*namespace_route*>/oauth/token endpoints.
- Sent as an Authorization: Bearer… header.
- Sent as a websocket subprotocol header in the form base64url.bearer.authorization.k8s.io.<base64url-encoded-token> for websocket requests.

X.509 client certificates

- Requires an HTTPS connection to the API server.
- Verified by the API server against a trusted certificate authority bundle.
- The API server creates and distributes certificates to controllers to authenticate themselves.

Any request with an invalid access token or an invalid certificate is rejected by the authentication layer with a 401 error.

If no access token or certificate is presented, the authentication layer assigns the system:anonymous virtual user and the system:unauthenticated virtual group to the request. This allows the authorization layer to determine which requests, if any, an anonymous user is allowed to make.

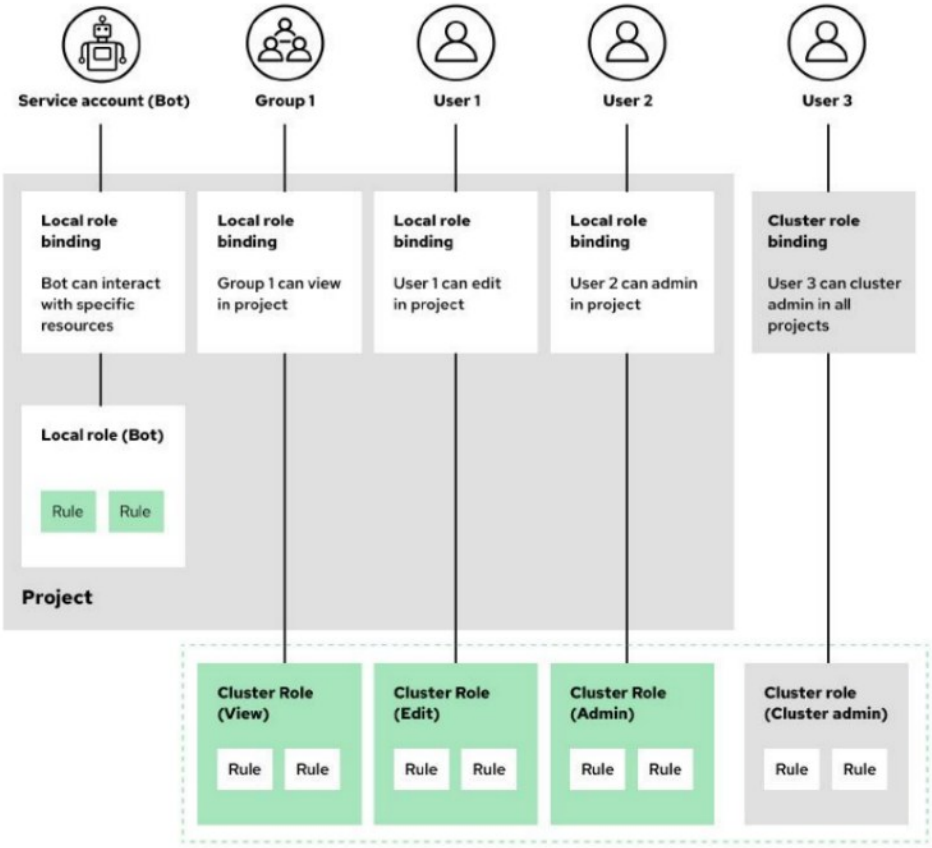
OpenShift role-based access control

The concept of identity is used to uniquely identify an actor in an OpenShift cluster. It defines who the actor is. Identity is represented as a user but, from a security standpoint, there are different types of users defined at different levels to take into consideration.

The actions a user is allowed to perform in the cluster depends on their unique identification determined via methods such as role-based access control (RBAC). An identity can also be used to audit what actions a user performed at some point in the past.

Users or groups can be given access to projects by cluster administrators or project administrators but may also be delegated to create their own projects. Users or groups are only allowed to see the content in the projects to which they are assigned and are given specific roles within projects. These roles are referred to as local role bindings and determine what actions are allowed within those projects. Figure 14) OpenShift user, group and role illustrates the relationship between projects and local role binding.

Figure 14) OpenShift user, group and role



See Table 12 for information on the predefined OpenShift roles. To meet the needs of your environment, you can create custom roles by cloning an existing role and modifying the cloned copy for your environment.

Table 12) Predefined OpenShift roles and description.

Role Name	Scope	Description
cluster-admin	Cluster-wide	• Full administrative access to the entire cluster can perform any action.
cluster-status	Cluster-wide	• Allow requesting cluster status information.
self-provisioner	Cluster-wide	• Permits creating new projects (enabled by default for all users).
admin	Project	• Full control over a project; can manage resources and roles (except quotas).
edit	Project	• Can create and modify application resources; cannot manage roles or quotas.
view	Project	• Read-only access to view resources in a project.

Basic-user	Project	• Basic read access to projects and limited self-information.
system:admin	System	• Full cluster admin privileges for system components.
system:node-admin	System	• Node-level administrative access.
system:master	System	• Master-level system access.
system:openshift-registry	System	• Access for the internal image registry.
system:image-builder	System	• Allows building images.
system:image-puller	System	• Allows pulling images from the internal registry.
system:build-strategy-docker	System	• Allows using Docker build strategy.
system:build-strategy-source	System	• Allows using Source-to-Image (S2I) build strategy.

OpenShift Container Platform OAuth server

The OpenShift Container Platform control plane includes an integrated OAuth server that issues OAuth access tokens for API authentication. When a user requests a new token, the OAuth server consults the configured identity provider to verify the user's identity. It then maps that identity to a corresponding user account, generates an access token, and returns it for use.

The internal OAuth server generates two kinds of tokens:

- Access tokens: Longer-lived tokens that grant access to the API.
- Authorize codes: Short-lived tokens whose only use is to be exchanged for an access token.

You can set default lifetimes for both token types, and if needed, override the access token duration by defining an OAuthClient object.

You can configure default options for the internal OAuth server's token duration.

Note: By default, tokens are only valid for 24 hours. Existing sessions expire after this time elapses

To change the default value based on the requirement, create a configuration file that contains the token duration options. Add **accessTokenMaxAgeSeconds** and **accessTokenInactivityTimeout** field and set desired value:

Sample config file for setting token duration of 4 hours.

```
apiVersion: config.openshift.io/v1
kind: OAuth
metadata:
  name: cluster
spec:
  tokenConfig:
    accessTokenMaxAgeSeconds: 14400
```

Save the file to apply the changes and Check that the OAuth server pods have restarted:

```
[admin@sec-rhel-9 ocp-sec]$ oc get clusteroperators authentication
NAME             VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE   MESSAGE
authentication   4.19.13   True        True          False      33d
OAuthServerDeploymentProgressing: deployment/oauth-openshift.openshift-authentication: observed
generation is 13, desired generation is 14.
```

Check that a new revision of the Kubernetes API server pods has rolled out. This will take several minutes.

```
[admin@sec-rhel-9 ]$ oc get clusteroperators kube-apiserver
NAME             VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE   MESSAGE
kube-apiserver   4.19.13   True        False         False      33d
```

You can also configure OAuth tokens to expire after a specified period of inactivity. By default, no inactivity timeout is applied.

```
oc edit oauth cluster
```

Now add the spec.tokenConfig.accessTokenInactivityTimeout field and set your timeout value.

```
apiVersion: config.openshift.io/v1
kind: OAuth
metadata:
...
spec:
  tokenConfig:
    accessTokenInactivityTimeout: 400s
```

Note: The minimum allowed timeout value is 300s.

You can customize the internal oAuth server URL by setting the custom hostname and TLS certificate in the spec.componentRoutes field of the cluster Ingress configuration. For more information refer here.

Understanding OpenShift Identity Provider

The OpenShift Container Platform master includes an integrated OAuth server that issues OAuth access tokens for API authentication by developers and administrators.

After installing your cluster, administrators can configure OAuth to define an identity provider. By default, only a kubeadmin user exists on your cluster. To specify an identity provider, you must create a custom resource (CR) that describes that identity provider and add it to the cluster.The following types of identity providers are available with OpenShift:

Table 13) Identity providers available with OpenShift

Identity provider	Description
httpassd	Configure the htpasswd identity provider to validate user names and passwords against a flat file generated using htpasswd.
Keystone	Configure the keystone identity provider to integrate your OpenShift Container Platform cluster with Keystone to enable shared authentication with an OpenStack Keystone v3 server configured to store users in an internal database.
LDAP	Configure the ldap identity provider to validate user names and passwords against an LDAPv3 server, using simple bind authentication.
Basic authentication	Configure a basic-authentication identity provider for users to log in to OpenShift Container Platform with credentials validated against a remote identity provider. Basic authentication is a generic backend integration mechanism.
Request header	Configure a request-header identity provider to identify users from request header values, such as X-Remote-User. It is typically used in combination with an authenticating proxy, which sets the request header value.
GitHub or GitHub Enterprise	Configure a github identity provider to validate user names and passwords against GitHub or GitHub Enterprise's OAuth authentication server.
GitLab	Configure a gitlab identity provider to use GitLab.com or any other GitLab instance as an identity provider.
Google	Configure a google identity provider using Google's OpenID Connect integration.

OpenID Connect	Configure an oidc identity provider to integrate with an OpenID Connect identity provider using an Authorization Code Flow.

Best Practices for RBAC Management

Follow the Principle of Least Privilege: While it is possible to use wildcards to grant access to all resources in an apiGroup or allow the role to use all verbs, it is always advisable to enumerate only the needed verbs and resources, especially if the role only needs read access.

Use ClusterRoles for Namespaced Resources with Caution: Roles scoped at a cluster level, i.e. ClusterRoles, should be used with extreme caution. Instead, using local roles to provide access at a namespace level offers more granular access. It is better to grant access to an explicit list of namespaces as opposed to the entire cluster. Otherwise, the ClusterRole might inadvertently open up access to resources in a namespace that the user does not need.

Make Use of the Predefined ClusterRoles: OpenShift 4 provides a set of predefined ClusterRoles. Defining custom roles instead of using the default ones can result in unexpected behavior as the predefined ClusterRoles are maintained and curated by the OpenShift team, with frequent patches and role updates. When defining a ClusterRole, it might be easy to forget about some resource or a verb, which might be difficult to debug or worse, it might be tempting to create a ClusterRole with more permission than is needed.

Don’t Use the Default ServiceAccount, Use a Specific One Instead: While each namespace comes with a stock default serviceAccount, it is preferable to add a specific serviceAccount for a pod. This is both more explicit and enables better documentation and traceability back to the specific pod’s serviceAccount. More importantly, the default service account is used unless an explicit serviceAccount is specified for a pod — in practice, extending the permissions for the default serviceAccount means that all pods that don’t explicitly set a specific serviceAccount would inherit these extended permissions. This could result in unintended access being granted.

Manage Bindings via Groups as Much as Possible: When binding multiple subjects to a Role through a RoleBinding or a ClusterRoleBinding, it is preferable to include all the subjects in a group and refer to the group in the binding over individually listing the users. Having the set of users centrally defined as a group allows for easier review and management of the group, which is especially important if it’s needed to remove a user’s binding.

Integrating with External Identity Providers

OpenShift supports a variety of OAuth identity providers, each providing a different degree of assured trust. The choice of identity provider depends on factors such as what is available in the environment and the desired level of security. See the OpenShift documentation for a full list of supported identity providers.

Where possible, choose to use an OAuth or OIDC-based identity provider (GitLab, OIDC) over the HTPasswd, LDAP or BasicAuthentication, as they provide stronger authentication methods. Alternatively, the RequestHeader identity provider can be used to tie into SAML and Microsoft SSPI systems and can be configured to operate against custom identity solutions by using an Apache reverse proxy and custom Apache modules. For examples of this type of identity provider configuration, see Configuring a request header identity provider.

The OpenShift CLI (oc) supports the Security Support Provider Interface (SSPI) to allow for SSO flows on Microsoft Windows. If you use the request header identity provider with a GSSAPI-enabled proxy to connect an Active Directory server to OpenShift Container Platform, users can automatically authenticate to OpenShift Container Platform by using the oc command line interface from a domain-joined Microsoft Windows computer.

Removing the kubeadmin user

After you define an identity provider and create a new cluster-admin user, you can remove the kubeadmin user to improve cluster security.

```
oc delete secrets kubeadmin -n kube-system
```

Note: Running this procedure before another user has cluster-admin rights will require a full reinstallation of OpenShift Container Platform. This command cannot be undone.

Kubernetes security

Securing Platform Services

Described below are the individual Kubernetes platform services that need to be managed by OpenShift.

Access to the Cluster For users to interact with the OpenShift Container Platform, they must first authenticate to the cluster. OpenShift includes an integrated OAuth server for token-based authentication. The authentication layer identifies the user or service associated with requests to the OpenShift Container Platform API. The authorization layer then uses information about the requesting user or service to determine if the request is allowed. Internal connections to the API server are authenticated by X.509 certificates. External access to the API server is managed by the ingress controller. Best practice is to separate access to the API server from access to workloads running on the cluster. This can be achieved by configuring separate ingress controllers for each type of access.

Control Plane The control plane is composed of master nodes. OpenShift services, such as the API Server, etcd, Controller, Scheduler, etc., run only on these master nodes. These control plane services manage workloads on the compute nodes, which are also known as worker nodes. A default OpenShift 4 cluster must include three master nodes to ensure that a quorum for etcd is maintained. Most of the control plane components are deployed as static pods. Static pods are managed by the kubelet and are always bound to one kubelet on a specific node. The kubelet automatically creates a mirror Pod on the API server for each static pod. This means that the pods running on a node are visible on the API server but cannot be controlled from there which minimizes the attack surface. Security secrets for control plane components such as the Kubernetes apiserver, etcd, the controller manager, and the scheduler are stored with their respective static pod configurations in the /etc/kubernetes/static-pod-resources/*/secrets directory on its host. Secrets for the control plane components are automatically managed and rotated by OpenShift.

API Server

The OpenShift API server is managed by the apiserver operator. The API server is served over HTTPS with authentication and authorization, and the secure API endpoint is bound to 0.0.0.0 :6443. The API server cannot be configured to listen on any other port. If needed, configure the load balancer in front of the API to listen on any custom port and redirect requests to the expected port 6443. The default API server uses the certificate from the Ingress Controller. Clients outside of the cluster will not be able to verify the API server’s certificate by default. This certificate can be replaced by one that is issued by a CA that clients trust. When deployed in a public cloud, the API server can and should be configured to only be accessed from a private zone.

OAuth Server

OpenShift includes an embedded OAuth server for authentication and Role-Based Access Control (RBAC) for authorization.

Etcd

etcd stores the persistent master state while other components watch etcd for changes to bring themselves into the specified state. etcd also stores kubernetes secrets. Given its importance to the functioning of the cluster, security for the etcd datastore is built into OpenShift. The cluster etcd is

managed by the cluster etcd operator. etcd is automatically deployed on each of the 3 master nodes. Its pod specification file is created on control plane nodes at /etc/kubernetes/manifests/etcdmember.yaml. The kubeconfig file for system:admin (admin.conf) is stored in /etc/kubernetes/kubeconfig.

OpenShift uses X.509 certificates to provide secure communication to etcd. OpenShift configures these automatically. OpenShift does not use the etcd-certfile or etcd-keyfile flags.

OpenShift supports data at rest encryption of the etcd datastore but it is up to the customer to configure. The AES-CBC (Cipher Block Chaining) cipher is used with the keys stored and automatically rotated on the filesystem of the master. Encryption of the etcd datastore can be enabled postinstallation against a running system. For more information on etcd encryption please see the section in Chapter 8 entitled Etcd Datastore Encryption.

Scheduler

Pod scheduling is an internal process that determines placement of new pods onto nodes within the cluster. The scheduler watches new pods as they get created and identifies the most suitable node to host them. It then creates bindings (pod to node bindings) for the pods using the master API. The default scheduling behavior is managed with the OpenShift kube-scheduler-operator.

Cluster administrators may wish to influence the behavior of the scheduler in order to simplify auditing of regulatory requirements. For example, if an organization's policy requires that applications with certain types of sensitive 138 data only be deployed to specific physical nodes. This can be done by making use of the following advanced pod scheduling methods. Each one addresses a different use case so it’s important to understand what behavior is desired. It is important to note that a combination of methods may be required in order to achieve the desired outcome.

Note: The scheduling operator’s default behavior can be changed by creating or editing the scheduler policy ConfigMap in the openshift-config project. However, it is not recommended. The techniques listed below are preferred.

Node Selectors

Node selectors are used when there are specific resources that must be scheduled on specific nodes. With node selectors, a label is applied to the node and all services that need to be scheduled on those nodes need to be configured with the corresponding node selector. Node selectors can be configured on a cluster level, project level or pod level for desired granularity.

If a node selector is specified in conjunction with a resource and there are no available nodes with that selector label, those resources will not be schedulable. On the contrary, if a node selector is not specified within a resource, those resources can still be scheduled on the nodes with a selector label.

Taints and Tolerations

Taints and tolerations are used when the desired outcome is to prevent resources from being scheduled on specific nodes by default unless otherwise necessary. When a taint is applied to a node, all resources will be 139 repelled from that node unless it is configured with a toleration that allows it to be scheduled there.

Controller Manager

The Controller Manager Server watches etcd for changes to objects such as replication, namespace, and service account controller objects, and then uses the API to enforce the specified state. The kube-controller-manager is managed with the cluster Controller Manager Operator. In other words, the controller manager, in combination with the other cluster operators ensures that the declared state for objects in the cluster is maintained.

Ingress Controller

The Ingress Controller and wild card DNS are managed with the ingress operator. An Ingress Controller is configured to accept external requests and proxy them based on the configured routes. This is limited to HTTP, HTTPS using SNI, and TLS using SNI, which is enough for web applications and services that work over TLS with SNI. By default, the OpenShift Container Platform uses the Ingress Operator to create

an internal CA and issue a wildcard certificate that is valid for applications under the .apps subdomain. The default ingress certificate can be replaced. After replacing the certificate, all applications, including the web console and CLI, will have encryption provided by a specified certificate. See the Replacing default ingress certificate documentation. More information about the Ingress Operator can be found in Chapter 6: Network Security.

Console

The OpenShift Container Platform web console is a user interface accessible from a web browser. Administrators can use the web console to manage and monitor the status of the cluster. Developers can use the webconsole to visualize, browse, and manage the contents of projects. Modify the OpenShift Container Platform web console to set a logout redirect URL or disable the console.

Kubelet

The kubelet runs on each node in the cluster and registers each node with the API server. The kubelet is installed as part of RHEL CoreOS and runs as a systemd service. OpenShift automatically generates and rotates the certificates for the kubelet to serve HTTPS traffic. CRI-O Container Runtime OpenShift uses CRI-O as the container runtime. CRI-O is run as a systemd service on each node in the cluster and is installed as part of RHEL CoreOS. CRI-O is a lightweight, Kubernetes-specific runtime with a reduced attack surface. CRI-O is versioned with Kubernetes.

Network security

In this section we will discuss ssecuring network traffic and enforcing network policies in OpenShift Container Platform. The network is a major threat vector and is involved in most Information security attacks. Typically, such network attacks exploit system vulnerabilities or misconfiguration.

The adoption of cloud computing challenges the traditional perimeter security model that assumed relatively static and controlled workloads. The concept of zero trust security has emerged to address new security challenges of cloud native architecture:

- The cloud infrastructure is shared among workloads with different levels of trust
- Applications are decomposed into interconnected containerized microservices increasing the attack surface
- Continuous deployment of new software versions potentially changes the communication patterns

Controlling the traffic between pods is an essential part of securing the applications. Traditional workloads use firewalls and specific routing rules to provide isolation via partitioning, traffic restrictions, and port blocking. In the container world, container runtimes and orchestrators handle this functionality through defined network security policies.

One of OpenShift's strengths is the ability to comprehensively manage the wide variety of control types in logical collections such as pods and their related controls. Network policy is defined using both cluster-scoped and namespace-scoped network policy APIs. By defining network policy across these different levels, you can create sophisticated network security configurations for your clusters, including full multi-tenant isolation.

Network policies

Cluster-scoped network policy

Cluster and network administrators can use the AdminNetworkPolicy to define network policy at the cluster level. The AdminNetworkPolicy feature consists of two APIs: the AdminNetworkPolicy API and BaselineAdminNetworkPolicy API. These APIs are used to set rules that can be applied to the entire cluster or delegated to the namespace-scoped NetworkPolicy.

Policies defined using the AdminNetworkPolicy API take precedence over all other policy types when set to "Allow" or "Deny". However, administrators can also use "Pass" to delegate responsibility for a given

policy to the namespace-scoped NetworkPolicy to allow application developers and namespace tenants to control specific aspects of network security for their projects.

Policies defined using the BaselineAdminNetworkPolicy API apply only when no other network policy overrides them. When you use the AdminNetworkPolicy API to delegate an aspect of network policy to the namespace-scoped NetworkPolicy, you should also define a sensible minimum restriction in the BaselineAdminNetworkPolicy. This ensures a baseline level of network security at the cluster level in case the NetworkPolicy for a namespace does not provide sufficient protection.

Namespace-scoped network policy

Application developers and namespace tenants can use the NetworkPolicy API to define network policy rules for a specific namespace. Rules in the NetworkPolicy for a namespace take precedence over cluster-wide rules configured using the BaselineAdminNetworkPolicy API, or for a cluster-wide rule that has been delegated or "passed" from the cluster-wide AdminNetworkPolicy API.

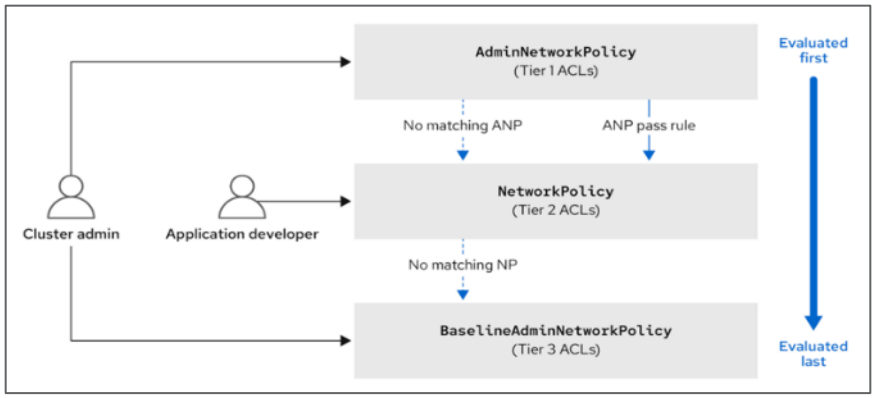
Understanding network policy

When a network connection is established, the network provider (default: OVN-Kubernetes) checks the connection details against network policy rules to determine how to handle the connection.

OVN-Kubernetes evaluates connections against network policy objects in the following order:

1. Check for matches in the AdminNetworkPolicy tier.
    i. If a connection matches an Allow or Deny rule, follow that rule and stop evaluating.
    ii. If a connection matches a Pass rule, move to the NetworkPolicy tier.
2. Check for matches in the NetworkPolicy tier.
    i. If a connection matches a rule, follow that rule and stop evaluating.
    ii. If no match is found, move to the BaselineAdminNetworkPolicy tier.
3. Follow a matching rule in the BaselineAdminNetworkPolicy tier.

Figure 15) OpenShift network policy



AdminNetworkPolicy

An AdminNetworkPolicy (ANP) is a cluster-scoped custom resource definition (CRD). As a OpenShift Container Platform administrator, you can use ANP to secure your network by creating network policies before creating namespaces. Additionally, you can create network policies on a cluster-scoped level that is non-overridable by NetworkPolicy objects.

The key difference between AdminNetworkPolicy and NetworkPolicy objects are that the former is for administrators and is cluster scoped while the latter is for tenant owners and is namespace scoped.

An ANP allows administrators to specify the following:

- A priority value that determines the order of its evaluation. The lower the value the higher the precedence.
- A set of pods that consists of a set of namespaces or namespace on which the policy is applied.
- A list of ingress rules to be applied for all ingress traffic towards the subject.
- A list of egress rules to be applied for all egress traffic from the subject.

BaselineAdminNetworkPolicy

BaselineAdminNetworkPolicy (BANP) is a cluster-scoped custom resource definition (CRD). As a OpenShift Container Platform administrator, you can use BANP to setup and enforce optional baseline network policy rules that are overridable by users using NetworkPolicy objects if need be. Rule actions for BANP are allow or deny.

The BaselineAdminNetworkPolicy resource is a cluster singleton object that can be used as a guardrail policy in case a passed traffic policy does not match any NetworkPolicy objects in the cluster. A BANP can also be used as a default security model that provides guardrails that intra-cluster traffic is blocked by default and a user will need to use NetworkPolicy objects to allow known traffic. You must use default as the name when creating a BANP resource.

A BANP allows administrators to specify:

- A subject that consists of a set of namespaces or namespace.
- A list of ingress rules to be applied for all ingress traffic towards the subject.
- A list of egress rules to be applied for all egress traffic from the subject.

NetworkPolicy

By default, all pods in a project are accessible from other pods and network endpoints. To isolate one or more pods in a project, you can create NetworkPolicy objects in that project to indicate the allowed incoming connections. Project administrators can create and delete NetworkPolicy objects within their own project.

If a pod is matched by selectors in one or more NetworkPolicy objects, then the pod will accept only connections that are allowed by at least one of those NetworkPolicy objects. A pod that is not selected by any NetworkPolicy objects is fully accessible.

A network policy applies to only the Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), and Stream Control Transmission Protocol (SCTP) protocols. Other protocols are not affected.

Note: A network policy does not apply to the host network namespace. Pods with host networking enabled are unaffected by network policy rules. However, pods connecting to the host-networked pods might be affected by the network policy rules.

Let’s examine an example of how a network policy is applied. In this scenario, two pods have been deployed within the same namespace.

```
[admin@sec-rhel-9 ocp-sec]$ oc get pod  -n  my-namespace -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP             NODE
NOMINATED NODE   READINESS GATES
test-deployment-84ccb5677f-mq4jf   1/1     Running   0          81s   10.128.4.188   compute-1
<none>           <none>
test-deployment-84ccb5677f-scxw6   1/1     Running   0          81s   10.128.1.17    compute-2
<none>           <none>
```

By default, both pods can communicate with each other.

```
[root@test-deployment-84ccb5677f-mq4jf /]# ping 10.128.1.17
PING 10.128.1.17 (10.128.1.17) 56(84) bytes of data.
64 bytes from 10.128.1.17: icmp_seq=1 ttl=62 time=2.21 ms
64 bytes from 10.128.1.17: icmp_seq=2 ttl=62 time=1.71 ms
64 bytes from 10.128.1.17: icmp_seq=3 ttl=62 time=0.142 ms
64 bytes from 10.128.1.17: icmp_seq=4 ttl=62 time=0.215 ms
64 bytes from 10.128.1.17: icmp_seq=5 ttl=62 time=0.223 ms
```

Create a NetworkPolicy and apply to make a project deny by default.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: deny-by-default
  namespace: my-namespace
spec:
  podSelector: {}
  ingress: []
```

Ping again.

```
[root@test-deployment-84ccb5677f-mq4jf /]# ping 10.128.1.17
PING 10.128.1.17 (10.128.1.17) 56(84) bytes of data.
```

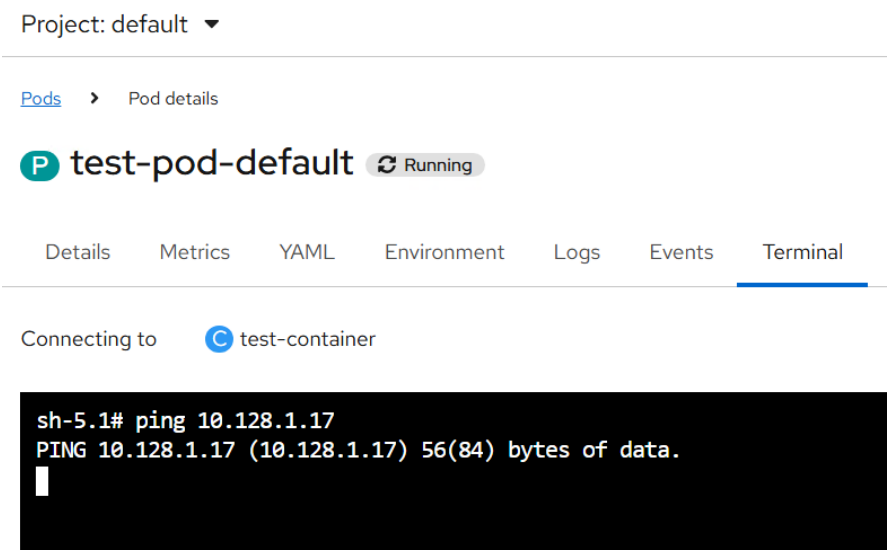
Similarly, we can create a NetworkPolicy to accept connections from other pods in the same project but reject all other connections from pods in other projects.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: allow-same-namespace
  namespace: my-namespace
spec:
  podSelector: {}
  ingress:
  - from:
    - podSelector: {}
```

Test ping.

```
[root@test-deployment-84ccb5677f-mq4jf /]# ping 10.128.1.17
PING 10.128.1.17 (10.128.1.17) 56(84) bytes of data.
64 bytes from 10.128.1.17: icmp_seq=1 ttl=62 time=2.92 ms
64 bytes from 10.128.1.17: icmp_seq=2 ttl=62 time=1.83 ms
64 bytes from 10.128.1.17: icmp_seq=3 ttl=62 time=0.255 ms
64 bytes from 10.128.1.17: icmp_seq=4 ttl=62 time=0.238 ms
64 bytes from 10.128.1.17: icmp_seq=5 ttl=62 time=0.137 ms
64 bytes from 10.128.1.17: icmp_seq=6 ttl=62 time=0.137 ms
64 bytes from 10.128.1.17: icmp_seq=7 ttl=62 time=0.107 ms
^C
--- 10.128.1.17 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6132ms
rtt min/avg/max/mdev = 0.107/0.802/2.916/1.035 ms
```

Now try to ping from a pod from another namespace.



For more information about network policies, refer to the Red Hat OpenShift documentation.

Egress Firewall

As a cluster administrator, you can create an egress firewall for a project that restrict egress traffic leaving your OpenShift Container Platform cluster. An egress firewall supports the following scenarios:

- A pod can only connect to internal hosts and cannot initiate connections to the public internet.
- A pod can only connect to the public internet and cannot initiate connections to internal hosts that are outside the OpenShift Container Platform cluster.
- A pod cannot reach specified internal subnets or hosts outside the OpenShift Container Platform cluster.
- A pod can only connect to specific external hosts.

For example, you can allow one project access to a specified IP range but deny the same access to a different project. Or, you can restrict application developers from updating from Python pip mirrors, and force updates to come only from approved sources.

You configure an egress firewall policy by creating an EgressFirewall custom resource (CR). The egress firewall matches network traffic that meets any of the following criteria:

- An IP address range in CIDR format
- A DNS name that resolves to an IP address
- A port number
- A protocol that is one of the following protocols: TCP, UDP, and SCTP

The following example outlines the egress firewall rules necessary for API server connectivity and restricting access to another subnet.

```
apiVersion: k8s.ovn.org/v1
kind: EgressFirewall
metadata:
  name: default
  namespace: test-egress
spec:
  egress:
  - to:
      cidrSelector: 10.61.178.0/24
```

```
    type: Allow
# ...
  - to:
      cidrSelector: 10.61.177.0/24
    type: Deny
```

#Validation.

We deployed a sample pod in the test-egress namespace and will try to ping the API servers and other subnet.

```
[admin@sec-rhel-9 ocp-sec]# oc get pod -n test-egress
NAME       READY   STATUS    RESTARTS   AGE
test-pod   1/1     Running   0          17m
```

From inside the pod, ping API server and Google.

```
[root@test-pod /]# ping 10.61.178.101
PING 10.61.178.101 (10.61.178.101) 56(84) bytes of data.
64 bytes from 10.61.178.101: icmp_seq=1 ttl=62 time=2.57 ms
64 bytes from 10.61.178.101: icmp_seq=2 ttl=62 time=0.739 ms
64 bytes from 10.61.178.101: icmp_seq=3 ttl=62 time=0.172 ms
64 bytes from 10.61.178.101: icmp_seq=4 ttl=62 time=0.201 ms
64 bytes from 10.61.178.101: icmp_seq=5 ttl=62 time=0.185 ms
^C
--- 10.61.178.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4107ms
```

```
[root@test-pod /]# ping www.google.com
PING www.google.com (142.251.153.119) 56(84) bytes of data.
64 bytes from 142.251.153.119 (142.251.153.119): icmp_seq=1 ttl=106 time=9.80 ms
64 bytes from 142.251.153.119 (142.251.153.119): icmp_seq=2 ttl=106 time=8.94 ms
64 bytes from 142.251.153.119 (142.251.153.119): icmp_seq=3 ttl=106 time=8.33 ms
64 bytes from 142.251.153.119 (142.251.153.119): icmp_seq=4 ttl=106 time=8.23 ms
64 bytes from 142.251.153.119 (142.251.153.119): icmp_seq=5 ttl=106 time=8.25 ms
```

Ping any endpoint on the restricted network which is set as Deny.

```
[root@test-pod /]# ping 10.61.177.104 -c 10
PING 10.61.177.104 (10.61.177.104) 56(84) bytes of data.

--- 10.61.177.104 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9228ms
```

Configuring IPsec encryption

By enabling IPsec, you can encrypt both internal pod-to-pod cluster traffic between nodes and external traffic between pods and IPsec endpoints external to your cluster. All pod-to-pod network traffic between nodes on the OVN-Kubernetes cluster network is encrypted with IPsec in *Transport mode*.

Note: IPsec is disabled by default. You can enable IPsec either during or after installing the cluster.

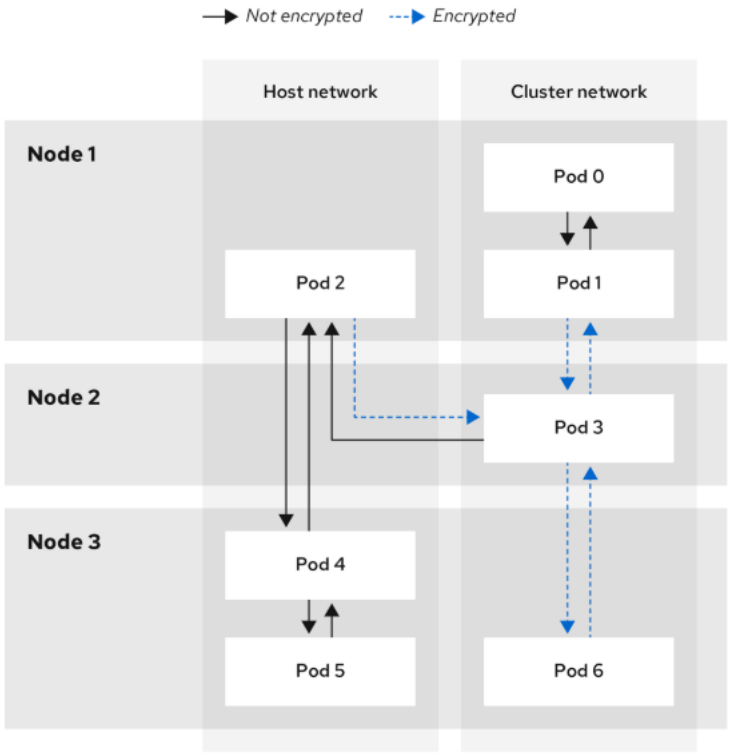
With IPsec enabled, only the following network traffic flows between pods are encrypted:

- Traffic between pods on different nodes on the cluster network
- Traffic from a pod on the host network to a pod on the cluster network

The following traffic flows are not encrypted:

- Traffic between pods on the same node on the cluster network
- Traffic between pods on the host network
- Traffic from a pod on the cluster network to a pod on the host network

Figure 16) Encrypted and non-encrypted traffic



The encrypt cipher used is AES-GCM-16-256. The integrity check value (ICV) is 16 bytes. The key length is 256 bits.

The IPsec mode used is *Transport mode*, a mode that encrypts end-to-end communication by adding an Encapsulated Security Payload (ESP) header to the IP header of the original packet and encrypts the packet data. OpenShift Container Platform does not currently use or support IPsec *Tunnel mode* for pod-to-pod communication.

OpenShift Container Platform supports the use of IPsec to encrypt traffic destined for external hosts, ensuring confidentiality and integrity of data in transit. This feature relies on X.509 certificates that you must supply.

As a cluster administrator you can enable pod-to-pod IPsec encryption between the cluster and external IPsec endpoints.

You can configure IPsec in either of the following modes:

- Full: Encryption for pod-to-pod and external traffic
- External: Encryption for external traffic

Container security

Isolating containers with multitenancy

Multitenancy allows applications on an OpenShift Container Platform cluster that are owned by multiple users, and run across multiple hosts and namespaces, to remain isolated from each other and from outside attacks. You obtain multitenancy by applying role-based access control (RBAC) to Kubernetes namespaces.

In Kubernetes, namespaces are areas where applications can run in ways that are separate from other applications. OpenShift Container Platform uses and extends namespaces by adding extra annotations, including MCS labeling in SELinux, and identifying these extended namespaces as projects. Within the scope of a project, users can maintain their own cluster resources, including service accounts, policies, constraints, and various other objects.

RBAC objects are assigned to projects to authorize selected users to have access to those projects. That authorization takes the form of rules, roles, and bindings:

- Rules define what a user can create or access in a project.
- Roles are collections of rules that you can bind to selected users or groups.
- Bindings define the association between users or groups and roles.

Local RBAC roles and bindings attach a user or group to a particular project. Cluster RBAC can attach cluster-wide roles and bindings to all projects in a cluster. There are default cluster roles that can be assigned to provide admin, basic-user, cluster-admin, and cluster-status access.

Protecting control plane with admission plugins

While RBAC controls access rules between users and groups and available projects, admission plugins define access to the OpenShift Container Platform master API. Admission plugins form a chain of rules that consist of:

Default admissions plugins: These implement a default set of policies and resources limits that are applied to components of the OpenShift Container Platform control plane.

Mutating admission plugins: These plugins dynamically extend the admission chain. They call out to a webhook server and can both authenticate a request and modify the selected resource.

Validating admission plugins: These validate requests for a selected resource and can both validate the request and ensure that the resource does not change again.

API requests go through admissions plugins in a chain, with any failure along the way causing the request to be rejected. Each admission plugin is associated with resources and only responds to requests for those resources. For mor information on admission plugins, refer here.

Security context constraints (SCCs)

You can use security context constraints (SCCs) to define a set of conditions that a pod must run with to be accepted into the system.

Some aspects that can be managed by SCCs include:

- Running of privileged containers
- Capabilities a container can request to be added
- Use of host directories as volumes
- SELinux context of the container
- Container user ID

If you have the required permissions, you can adjust the default SCC policies to be more permissive, if required.

SCCs are OpenShift resources that can be listed using the command below.

```
[admin@sec-rhel-9 ocp-sec]$ oc describe scc restricted
Name:                                           restricted
Priority:                                       <none>
Access:
  Users:                                        <none>
```

```
  Groups:                                       <none>
Settings:
  Allow Privileged:                             false
  Allow Privilege Escalation:                   true
  Default Add Capabilities:                     <none>
  Required Drop Capabilities:                   KILL,MKNOD,SETUID,SETGID
  Allowed Capabilities:                         <none>
  Allowed Seccomp Profiles:                     <none>
  Allowed Volume Types:
configMap,csi,downwardAPI,emptyDir,ephemeral,persistentVolumeClaim,projected,secret
  Allowed Flexvolumes:                          <all>
  Allowed Unsafe Sysctls:                       <none>
  Forbidden Sysctls:                            <none>
  Allow Host Network:                           false
  Allow Host Ports:                             false
  Allow Host PID:                               false
  Allow Host IPC:                               false
  Read Only Root Filesystem:                    false
  Run As User Strategy: MustRunAsRange
    UID:                                        <none>
    UID Range Min:                              <none>
    UID Range Max:                              <none>
  SELinux Context Strategy: MustRunAs
    User:                                       <none>
    Role:                                       <none>
    Type:                                       <none>
    Level:                                      <none>
  FSGroup Strategy: MustRunAs
    Ranges:                                     <none>
  Supplemental Groups Strategy: RunAsAny
    Ranges:                                     <none>
```

SELinux contexts can be managed for the container's main process by using SCCs. The SELinux Context setting allows management of a strategy; the restricted SCC defines a strategy of MustRunAs, which forces the pods of the project to define an SELinux policy but does not define any values for SELinux contexts, which means that the project must define the options, such as user, role, type, and level. Failure to do so prevents pods from being created.

Example of creating a custom SCC.

```
apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  name: custom-scc
allowPrivilegedContainer: false
requiredDropCapabilities:
- KILL
- MKNOD
- SETUID
- SETGID
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: MustRunAs
fsGroup:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
```

Seccomp Profiles

Seccomp (Secure Computing Mode) profiles are used to restrict the system calls that a container can make, thereby reducing the attack surface. Seccomp security profiles list the syscalls a process can make. Permissions are broader than SELinux, enabling users to restrict operations system-wide, such as write

- Default Profiles: OpenShift provides default profiles that can be used to secure containers.

- Custom Profiles: You can create custom seccomp profiles to meet specific security requirements.
- Applying Seccomp Profiles: Apply seccomp profiles to your pods by specifying them in the pod's security context.

Example of applying a seccomp profile:

```
apiVersion: v1
kind: Pod
metadata:
  name: seccomp-pod
spec:
  containers:
  - name: my-container
    image: my-image
    securityContext:
      seccompProfile:
        type: Localhost
        localhostProfile: my-custom-profile.json
```

In the later section we will cover Security Profiles Operator (SPO) that provides a way to define secure computing profiles and SELinux profiles as custom resources, synchronizing profiles to every node ina given namespace. The SPO manages SELinux policies and seccomp profiles for namespaced workloads. For more information, see Enabling the Security Profiles Operator.

You can create seccomp and SELinux profiles, bind policies to pods, record workloads, and synchronize all worker nodes in a namespace.

Granting roles to service accounts

You can assign roles to service accounts, in the same way that users are assigned role-based access. There are three default service accounts created for each project.

A service account:

- is limited in scope to a particular project
- derives its name from its project
- is automatically assigned an API token and credentials to access the OpenShift Container Registry

Service accounts associated with platform components automatically have their keys rotated.

Controlling access using OAuth

You can use API access control via authentication and authorization for securing your container platform. The OpenShift Container Platform master includes a built-in OAuth server. Users can obtain OAuth access tokens to authenticate themselves to the API.

As an administrator, you can configure OAuth to authenticate using an identity provider, such as LDAP, GitHub, or Google. The identity provider is used by default for new OpenShift Container Platform deployments, but you can configure this at initial installation time or post installation.

API access control and management

Applications can have multiple, independent API services which have different endpoints that require management. OpenShift Container Platform includes a containerized version of the 3scale API gateway so that you can manage your APIs and control access.

3scale gives you a variety of standard options for API authentication and security, which can be used alone or in combination to issue credentials and control access: standard API keys, application ID and key pair, and OAuth 2.0.

You can restrict access to specific endpoints, methods, and services and apply access policy for groups of users. Application plans allow you to set rate limits for API usage and control traffic flow for groups of developers.

For a tutorial on using APIcast v2, the containerized 3scale API Gateway, see Running APIcast on Red Hat OpenShift in the 3scale documentation.

Red Hat Single Sign-On

The Red Hat Single Sign-On server enables you to secure your applications by providing web single sign-on capabilities based on standards, including SAML 2.0, OpenID Connect, and OAuth 2.0. The server can act as a SAML or OpenID Connect–based identity provider (IdP), mediating with your enterprise user directory or third-party identity provider for identity information and your applications using standards-based tokens. You can integrate Red Hat Single Sign-On with LDAP-based directory services including Microsoft Active Directory and Red Hat Enterprise Linux Identity Management.

Secure self-service web console

OpenShift Container Platform provides a self-service web console to ensure that teams do not access other environments without authorization. OpenShift Container Platform ensures a secure multitenant master by providing the following:

- Access to the master uses Transport Layer Security (TLS)
- Access to the API Server uses X.509 certificates or OAuth access tokens
- Project quota limits the damage that a rogue token could do
- The etcd service is not exposed directly to the cluster

Managing certificates for the platform

OpenShift Container Platform has multiple components within its framework that use REST-based HTTPS communication leveraging encryption via TLS certificates. OpenShift Container Platform's installer configures these certificates during installation. There are some primary components that generate this traffic:

- masters (API server and controllers)
- etcd
- nodes
- registry
- router

Note: You can configure custom serving certificates for the public hostnames of the API server and web console during initial installation or when redeploying certificates. You can also use custom CA.

Container image signatures

Red Hat delivers signatures for the images in the Red Hat Container Registries. Those signatures can be automatically verified when being pulled to OpenShift Container Platform 4 clusters by using the Machine Config Operator (MCO).

Quay.io serves most of the images that make up OpenShift Container Platform, and only the release image is signed. Release images refer to the approved OpenShift Container Platform images, offering a degree of protection against supply chain attacks. However, some extensions to OpenShift Container Platform, such as logging, monitoring, and service mesh, are shipped as Operators from the Operator Lifecycle Manager (OLM). Those images ship from the Red Hat Ecosystem Catalog Container images registry.

To verify the integrity of those images between Red Hat registries and your infrastructure, enable signature verification.

Default policy

```
[core@compute-1 ~]$ cat /etc/containers/policy.json
{
    "default": [
        {
            "type": "insecureAcceptAnything"
        }
    ],
    "transports":
        {
            "docker-daemon":
                {
                    "": [{"type":"insecureAcceptAnything"}]
                }
        }
```

After applying the yaml file.

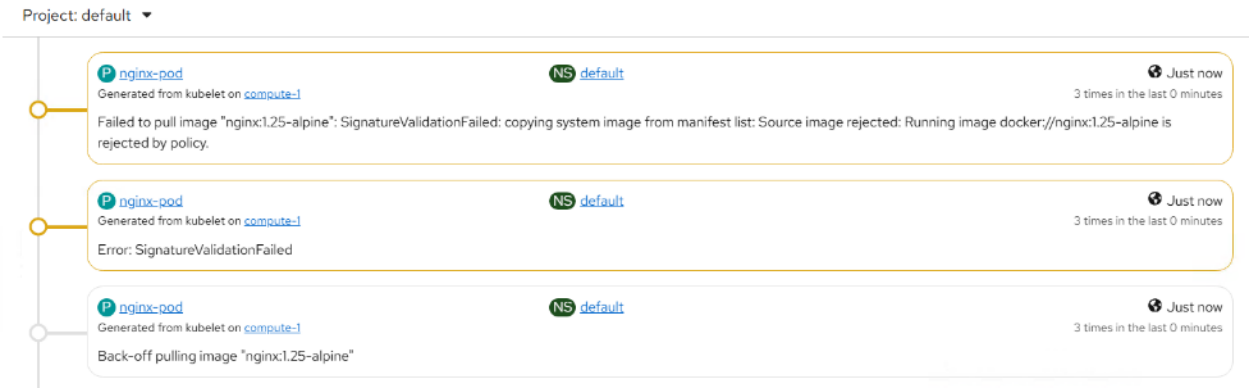
```
[core@compute-1 ~]$ cat /etc/containers/policy.json
{
  "default": [
    {
      "type": "reject"
    }
  ],
  "transports": {
    "docker": {
      "registry.access.redhat.com": [
        {
          "type": "signedBy",
          "keyType": "GPGKeys",
          "keyPath": "/etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release"
        }
      ],
      "registry.redhat.io": [
        {
          "type": "signedBy",
          "keyType": "GPGKeys",
          "keyPath": "/etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release"
        }
      ]
    },
    "docker-daemon": {
      "": [
        {
          "type": "reject"
        }
      ]
    }
  }
}
```

Let’s test by deploying a pod using generic image.

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
    - name: nginx
      image: nginx:1.25-alpine
      ports:
```

```
      - containerPort: 80
```

As expected, the signature validation failed, hence the image was not pulled.



For the restricted environment we can add only our private registries to the trust registry list.

Encryption and secret management

ETCD Encryption

Etcd is a distributed key-value datastore used by Kubernetes to store configuration data. Etcd is only available to OpenShift administrators. The data in etcd represents the entire state of the Kubernetes cluster. The datastore is continually monitored for changes which are then applied to the running state of the system. By default, secret objects are stored in etcd and retrieved as needed via the API. In OpenShift, all traffic between the API server and the worker nodes is encrypted, ensuring that secrets stored in etcd and transmitted to pods are encrypted in-transit.

By default, etcd data is not encrypted in OpenShift Container Platform. You can enable etcd encryption for your cluster to provide an additional layer of data security. For example, it can help protect the loss of sensitive data if an etcd backup is exposed to the incorrect parties. ETCD encryption can also be enable during the OpenShift installation along with FIPS. In this guide we have enabled encrypted post-installation.

When you enable etcd encryption, the following OpenShift API server and Kubernetes API server resources are encrypted:

- Secrets
- Config maps
- Routes
- OAuth access tokens
- OAuth authorize tokens

Supported encryption types

The following encryption types are supported for encrypting etcd data in OpenShift Container Platform:

- AES-CBC- Uses AES-CBC with PKCS#7 padding and a 32-byte key to perform the encryption. The encryption keys are rotated weekly.
- AES-GCM- Uses AES-GCM with a random nonce and a 32-byte key to perform the encryption. The encryption keys are rotated weekly.

Note: On the newer OpenShift cluster it is recommended to use AES-GCM.

```
[admin@sec-rhel-9 ocp-sec]$ oc get openshiftapiserver -o=jsonpath='{range
.items[0].status.conditions[?(@.type=="Encrypted")]}{.reason}{"\n"}{.message}{"\n"}'
EncryptionDisabled
Encryption is not enabled
```

Run the following command to

```
oc patch apiserver cluster -p='{"spec": {"encryption": {"type": "aesgcm"}}}' --type=merge
apiserver.config.openshift.io/cluster patched
```

Verify the progress and completion.

```
[admin@sec-rhel-9 ocp-sec]$ oc get openshiftapiserver -o=jsonpath='{range
.items[0].status.conditions[?(@.type=="Encrypted")]}{.reason}{"\n"}{.message}{"\n"}'
EncryptionInProgress
Resource routes.route.openshift.io is not encrypted

[admin@sec-rhel-9 ocp-sec]$ oc get kubeapiserver -o=jsonpath='{range
.items[0].status.conditions[?(@.type=="Encrypted")]}{.reason}{"\n"}{.message}{"\n"}'
EncryptionInProgress
Resource secrets is not encrypted


[admin@sec-rhel-9 ocp-sec]$ oc get openshiftapiserver -o=jsonpath='{range
.items[0].status.conditions[?(@.type=="Encrypted")]}{.reason}{"\n"}{.message}{"\n"}'
EncryptionCompleted
All resources encrypted: routes.route.openshift.io

[admin@sec-rhel-9 ocp-sec]$ oc get kubeapiserver -o=jsonpath='{range
.items[0].status.conditions[?(@.type=="Encrypted")]}{.reason}{"\n"}{.message}{"\n"}'
EncryptionCompleted
All resources encrypted: secrets, configmaps
```

Entropy

In computing, entropy is a measure of unpredictability or randomness. The operating system kernel maintains a pool of this randomness, often called the "entropy pool." Think of it like a constantly shuffled deck of cards. Every time a program needs a truly random number, it draws a card from this deck. If the deck isn't shuffled well or often enough (low entropy), the cards become predictable.

An OpenShift cluster is a massive consumer of cryptographic functions, all of which depend on a high-quality source of randomness. Without sufficient entropy, these operations can become slow or even insecure.

Key areas in OpenShift that rely heavily on entropy include:

- **TLS/SSL Certificates:** Generating private keys for certificates that secure communication between components, routes, and users.
- **Session Keys:** Creating unique, unpredictable keys for encrypted communication sessions (e.g., TLS handshakes).
- **Authentication Tokens:** Generating secure OAuth tokens, session cookies, and other secrets.
- **Data Encryption:** Any application-level or storage-level encryption (like etcd encryption) requires random keys.
- **Container Security:** Generating random IDs, secrets, and other values needed for secure container operation.

rngd (the RNG Daemon) is a critical component for entropy management on RHCOS and OpenShift nodes. It’s a lightweight service from the rng-tools package and feeds entropy from hardware RNG

devices (like TPM, virtio-rng, or CPU RNG instructions) into the kernel's entropy pool. It also ensures cryptographic operations (TLS, key generation, FIPS checks) have sufficient randomness. We will deploy rng test container to the cluster on one of the control nodes.

Below is the yaml file used to deploy a pod.

```
apiVersion: v1
kind: Pod
metadata:
  name: rng-test-pod
spec:
  nodeName: control-2 # IMPORTANT: Change this to your master node's name!
  containers:
  - name: rng-test-container
    # This is a standard Red Hat support image. If your cluster is disconnected,
    # you may need to replace this with an image from your internal registry.
    image: registry.redhat.io/rhel9/support-tools:9:6
    securityContext:
      privileged: true
    volumeMounts:
    - name: host-root
      mountPath: /host
    # This command just keeps the pod running so we can connect to it.
    command: ["/bin/bash", "-c", "sleep infinity"]
  volumes:
  - name: host-root
    hostPath:
      path: /
  restartPolicy: Never
  tolerations:
  - effect: NoSchedule
    key: node-role.kubernetes.io/master
    operator: Exists
```

Get a shell inside the running pod.

```
oc exec -it rng-test-pod -- /bin/bash
```

Install and run 'rngtest'.

```
dnf install -y rng-tools
```

Verify rng is installed.

```
[root@rng-test-pod /]# ps ax | grep rng
    244 pts/0    S+     0:00 grep --color=auto rng
[root@rng-test-pod /]# cat /proc/cpuinfo | grep -o rand | uniq
rand
root@rng-test-pod /]# cat /etc/os-release | grep PRETTY
PRETTY_NAME="Red Hat Enterprise Linux 9.6 (Plow)"
```

All 1000 samples passed the suite of FIPS tests. That means the random stream meets FIPS 140-2 statistical criteria.

```
[root@rng-test-pod /]#  cat /dev/random | rngtest -c 1000
rngtest 6.16
Copyright (c) 2004 by Henrique de Moraes Holschuh
This is free software; see the source for copying conditions.  There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PART
ICULAR PURPOSE.

rngtest: starting FIPS tests...
rngtest: bits received from input: 20000032
rngtest: FIPS 140-2 successes: 1000
rngtest: FIPS 140-2 failures: 0
rngtest: FIPS 140-2(2001-10-10) Monobit: 0
rngtest: FIPS 140-2(2001-10-10) Poker: 0
rngtest: FIPS 140-2(2001-10-10) Runs: 0
rngtest: FIPS 140-2(2001-10-10) Long run: 0
rngtest: FIPS 140-2(2001-10-10) Continuous run: 0
rngtest: input channel speed: (min=96.331; avg=6629.644; max=9536.743)Mibits/s
rngtest: FIPS tests speed: (min=165.856; avg=210.424; max=221.785)Mibits/s
rngtest: Program run time: 96759 microseconds
[root@rng-test-pod /]#
```

Public Key Certificates

Strong certificate use within the platform is critical to modern application security. The only way for public key infrastructure (PKI) to scale for a container orchestration platform is by increasing the use and reach of automation. OpenShift provides integrated management of X.509 certificates for internal cluster components. Containerized applications are responsible for managing their own certificates signed by organizational CAs or may make use of the OpenShift Service CA if they wish.

The platform includes multiple certificate authorities (CAs) providing independent chains of trust, increasing the security posture of the cluster. These internal self-signing CAs enable automation because the key is known to the cluster. The certificates generated by each CA are used to identify a particular OpenShift platform component to another OpenShift platform component. CA bundles are used when more than one communication path needs to be authenticated.

The OpenShift CAs are managed by the cluster and are only used within the cluster. Which means:

- Each cluster CA can only issue certificates for its own purpose within its own cluster.
- CAs for one OpenShift cluster cannot be used for a different OpenShift cluster, thus avoiding cross-cluster interference.
- Cluster CAs cannot be used by an external CA that the cluster does not control.

Like all secrets, long-term certificates are a point of vulnerability. OpenShift automatically manages rotation of certificates generated by the internal CAs. To increase security for external access points, custom certificates from an external CA can and should be installed for the public host names of the OpenShift Container Platform API and web console. This confines the use of internal CA to the cluster components.

For more information on configuring certificates, refer here.

To understand more about certificate types and descriptions, refer here.

Protecting Cluster Data on Disk

Organizational policies may require system configuration data or other operational data to be encrypted at rest. When installing OpenShift Container Platform, protection of data at rest can be achieved by enabling full-disk encryption that is managed by utilities on the operating system.

RHEL CoreOS supports full-disk encryption for both Network Bound Disk and TPM2 backed encryption modes. Currently, RHEL CoreOS does not support key-cycling for full-disk encryption. Full disk encryption is implemented through LUKS which uses passphrases to unlock the actual key. The passphrase for unlocking the key is discerned by Clevis through meta-data in the LUKS header and the backend (either a TPM2 or a Tang server). If the passphrase needs to be cycled, administrators must do a rolling-replacement of nodes using the updated configuration.

Note: As noted earlier in the OpenShift installation section, we have enabled disk encryption using the TPM2-backed encryption mode

#Output from one of the control node.

```
[core@control-1 ~]$ lsblk -o NAME,FSTYPE,TYPE | grep crypt
└─sda4   crypto_LUKS part
  └─root xfs         crypt
```

Auditing and logging

OpenShift is architected in a way that leverages the utility of containerizing its own applications and subsystems. This means that many traditional Linux audit functions are architected to fit OpenShift's placement in an overall solution set.

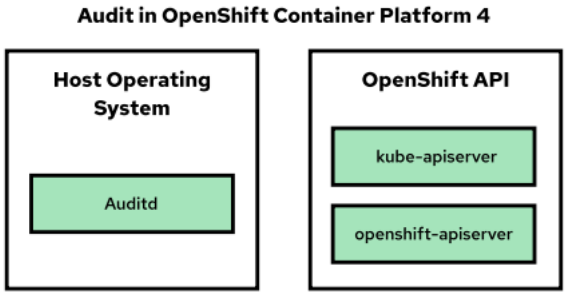
OpenShift Container Platform auditing provides a security-relevant chronological set of records documenting the sequence of activities that have affected the system by individual users, administrators, or other components of the system.

This may require a bit of rethinking on role separation, since an Information Security program may need to separate the duties of the function creator (performed by an application role), and the activity review (performed by an auditor role). This is because practitioners may arrive here with a certain viewpoint regarding responsibility for conducting audits and may perceive certain practices presented herein as a change of responsibility.

Since audit logs contain a record of security-relevant events, it is important to ensure they are not tampered with. This involves configuring auditing services to securely collect and store audit logs, and to protect those audit logs from unauthorized access. In many cases audit logs are securely forwarded to a log storage component or Security Information and Event Management (SIEM) system that is managed by a separate operations or security team. Information systems include this type of secure forwarding configuration to maintain non-repudiation of audit data, which is often an organizational requirement.

The auditd service in RHEL and RHCOS meets the requirements for ensuring non-repudiation for the audit logs it processes. However, it should be noted that general log collectors such as those found in the OpenShift cluster logging stack are best effort. OpenShift leverages its cluster logging components to aggregate audit, so the best-effort qualification is applied to audit collection and forwarding.

Figure 17) Audit in OpenShift



OpenShift Container Platform auditing provides a security-relevant chronological set of records documenting the sequence of activities that have affected the system by individual users, administrators, or other components of the system.

List the OpenShift API server audit logs that are available for each control plane node:

```
oc adm node-logs --role=master --path=openshift-apiserver/
```

View a specific OpenShift API server audit log by providing the node name and the log name

```
oc adm node-logs <node_name> --path=openshift-apiserver/<log_name>
```

List the Kubernetes API server audit logs that are available for each control plane node

```
oc adm node-logs --role=master --path=kube-apiserver/
```

View a specific Kubernetes API server audit log by providing the node name and the log name.

```
oc adm node-logs <node_name> --path=kube-apiserver/<log_name>
```

List the OpenShift OAuth API server audit logs that are available for each control plane node.

```
oc adm node-logs --role=master --path=oauth-apiserver/
```

View a specific OpenShift OAuth API server audit log by providing the node name and the log name

```
oc adm node-logs --role=master --path=oauth-server/
```

Note: For filtering audit logs, you can use tools like **jq**.

You can gather audit logs by running the command below.

```
oc adm must-gather -- /usr/bin/gather_audit_logs
```

Configuring the audit log policy

You can adjust how much detail the API server records by selecting an appropriate audit log policy profile. OpenShift Container Platform uses the **Default** profile unless you choose otherwise. Although you can switch to a more detailed profile that captures full request bodies, keep in mind that doing so will increase resource consumption, including CPU, memory, and I/O.

Note: Sensitive resources, such as Secret, Route, and OAuthClient objects, are only logged at the metadata level. OpenShift OAuth server events are only logged at the metadata level.

Edit the APIServer resource.

```
oc edit apiserver cluster
```

Add the profile.

```
apiVersion: config.openshift.io/v1
kind: APIServer
metadata:
...
spec:
  audit:
    profile: WriteRequestBodies
```

Verify that a new revision of the Kubernetes API server pods is rolled out. It can take several minutes for all nodes to update to the new revision.

```
oc get kubeapiserver -o=jsonpath='{range
.items[0].status.conditions[?(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"
}'
```

Review the NodeInstallerProgressing status condition for the Kubernetes API server to verify that all nodes are at the latest revision. The output shows AllNodesAtLatestRevision upon successful update.

Configuring audit log policy with custom rules

You can configure an audit log policy that defines custom rules. You can specify multiple groups and define which profile to use for that group. These custom rules take precedence over the top-level profile field. The custom rules are evaluated from top to bottom, and the first that matches is applied.

A hardened audit policy focuses on maximizing security visibility while minimizing unnecessary log noise. It provides full audit coverage for high-risk actors such as cluster administrators, CI/CD service accounts, and automation bots, ensuring that all their actions can be traced and reviewed. It enables full request-body logging for mutating operations—including POST, PUT, PATCH, and DELETE—to capture the exact changes made to cluster resources. At the same time, it applies selective logging for read operations to avoid overwhelming the system with excessive, low-value logs.

The policy ensures strong visibility into authentication events, RBAC activity, token usage, and OAuth flows, all of which are critical for detecting misconfigurations or malicious behavior. Routine system components generate minimal noise, with detailed logging only when they perform mutating actions. Finally, all unmatched or low-risk traffic falls back to Default logging, providing a secure and balanced baseline without unnecessary overhead.

#Sample custom policy:

```
apiVersion: config.openshift.io/v1
kind: APIServer
metadata:
  name: cluster
spec:
  audit:
    customRules:

      # 1) Cluster administrators (including kube:admin)
      # Full body logging for all reads/writes — highest visibility
      - group: system:masters
        profile: AllRequestBodies

      # 2) SDN & Controller service accounts performing infra changes
      # These mutate networking, nodes, security objects at scale.
      - group: system:serviceaccounts:openshift-network-operator
        profile: WriteRequestBodies

      - group: system:serviceaccounts:openshift-machine-config-operator
        profile: WriteRequestBodies

      # 3) Default serviceaccounts across all namespaces
      # Many workloads use the default SA; track mutating actions here.
      - group: system:serviceaccounts
        profile: WriteRequestBodies

      # 4) OAuth-authenticated users (human users)
      # Log request bodies on write verbs (POST, PUT, PATCH, DELETE)
      - group: system:authenticated:oauth
        profile: WriteRequestBodies

      # 5) Generic authenticated clients (e.g., API tokens, CI clients)
      # Read bodies are often noisy, so restrict to write bodies only.
      - group: system:authenticated
        profile: WriteRequestBodies

    # Default audit profile for everything else (unauthenticated or unmatched)
    profile: Default
```

Configure OpenShift remote logging

As a cluster administrator, you can deploy logging on your OpenShift Container Platform cluster and use it to collect and aggregate node system audit logs, application container logs, and infrastructure logs to a remote server.

You can use logging to perform the following tasks:

- Forward logs to your chosen log outputs, including on-cluster, Red Hat managed log storage.
- Visualize your log data in the OpenShift Container Platform web console.

To get started with logging, you must install the following Operators:

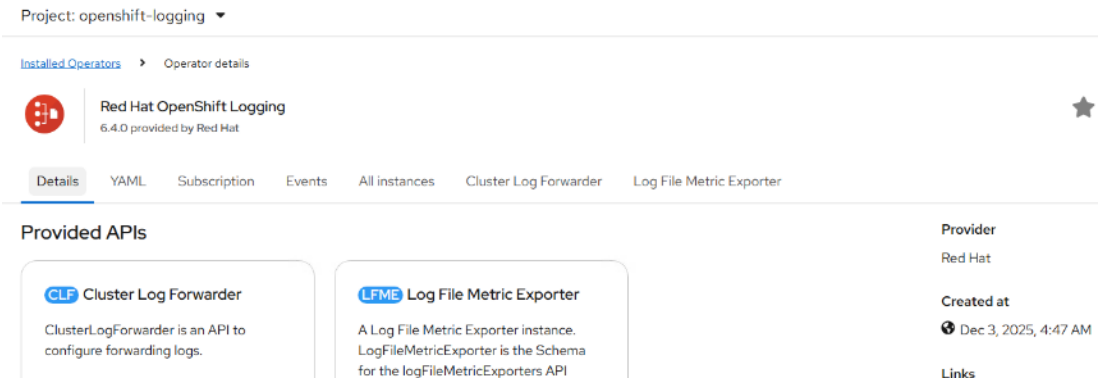
- Loki Operator to manage your log store.
- Red Hat OpenShift Logging Operator to manage log collection and forwarding.
- Cluster Observability Operator (COO) to manage visualization.

You can use either the OpenShift Container Platform web console or the OpenShift Container Platform CLI to install or configure logging.

Create a namespace

```
Oc create ns openshift-logging
```

Install Red Hat OpenShift Logging operator in the openshift-logging namespace.



Create a secret

```
oc create secret generic syslog-secret \
  --from-literal=username=admin \
  --from-literal=password=<syslog_password> \
  -n openshift-logging
```

Create ClusterLogForwarder CR object.

```
apiVersion: observability.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: rsyslog-ocp
  namespace: openshift-logging
spec:
  outputs:
    - name: remote-syslog
      type: syslog
      syslog:
        facility: local0
        rfc: RFC5424
        severity: informational
        url: tcp://10.61.177.100:514
      secret:
        name: syslog-secret
  pipelines:
  - inputRefs:
    - application
    - infrastructure
    - audit
    name: syslog-pipeline
    outputRefs:
    - remote-syslog
  serviceAccount:
    name: logcollector false
```

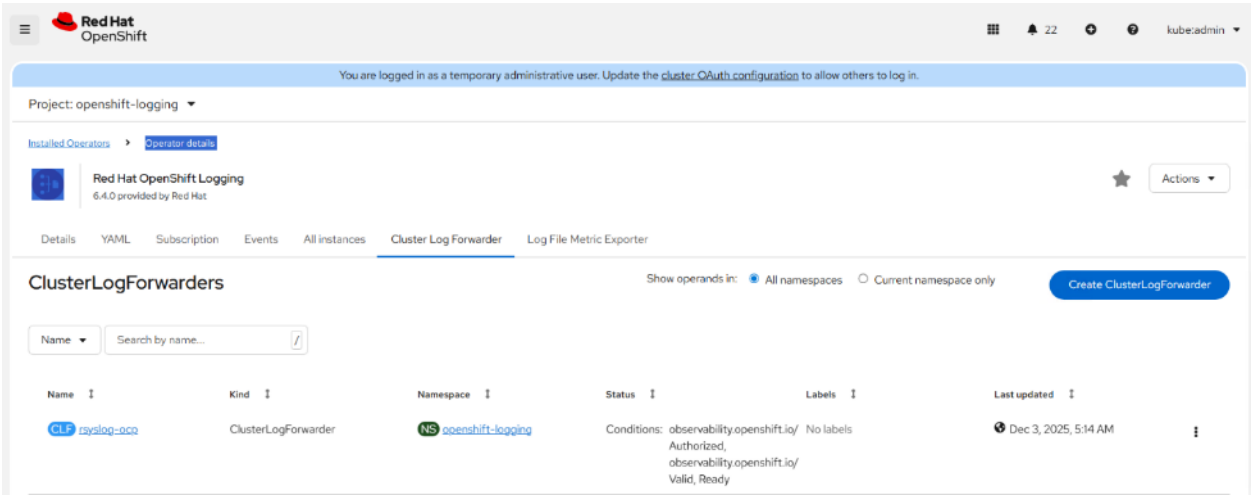
Note: You can add namespace_name, pod_name, and container_name elements to the message field of the record by adding the enrichment field to your ClusterLogForwarder custom resource (CR).

Best practice

If your remote syslog endpoint supports TLS (e.g., tcp+tls), provide a secret with the CA (and optionally client cert/key) and switch the URL send the logs.

Create the CR object.

```
oc create -f <filename.yaml
```



Example syslog message output on the syslog server.


```
Dec  3 15:49:49 control-2 openshift-cluster-version_cluster-version-operator-859689dfb7-
6ddpr_cluster-version-operator[398e7384-09ac-4980-b1aa-fd6bb8a52a89] namespace_name=openshift-
cluster-version, container_name=cluster-version-operator, pod_name=cluster-version-operator-
859689dfb7-6ddpr, message={"@timestamp":"2025-12-03T15:49:49.088589909Z","hostname":"control-
2","kubernetes":{"annotations":{"openshift.io/required-
scc":"hostaccess","openshift.io/scc":"hostaccess"},"container_id":"cri-
o://42a6db560d9383aed539ddb61d089ace3e8bc494d0df87df7bd7e3775793968d","container_image":"quay.io/
openshift-release-dev/ocp-
release@sha256:b221339d28377e7654ecfa76debf7cd11eccc4e45516cca393df6a5ca4dbc736","container_iostr
eam":"stderr","container_name":"cluster-version-operator","labels":{"k8s-app":"cluster-version-
operator","pod-template-hash":"859689dfb7"},"namespace_id":"1f3e725b-7718-4fa9-8888-
bfb9a4318dd7","namespace_labels":{"kubernetes_io_metadata_name":"openshift-cluster-
version","name":"openshift-cluster-version","olm_operatorgroup_uid_734f3f8d-17a3-40ad-be1e-
370c7032f148":"","openshift_io_cluster-monitoring":"true","openshift_io_run-level":"","pod-
security_kubernetes_io_audit":"privileged","pod-
security_kubernetes_io_enforce":"privileged","pod-
security_kubernetes_io_warn":"privileged"},"namespace_name":"openshift-cluster-
version","pod_id":"398e7384-09ac-4980-b1aa-
fd6bb8a52a89","pod_ip":"10.61.178.102","pod_name":"cluster-version-operator-859689dfb7-
6ddpr","pod_owner":"ReplicaSet/cluster-version-operator-
859689dfb7"},"level":"info","log_source":"container","log_type":"infrastructure","message":"I1203
15:49:49.088565       1 sync_worker.go:1056] Done syncing for configmap \"openshift-machine-
config-operator/machine-config-osimageurl\" (795 of 924)","openshift":{"cluster_id":"8ed9044f-
f02a-4f02-b28e-d551a40568d7","sequence":1764776989104065473},"timestamp":"2025-12-
03T15:49:49.088589909Z"}
```


Note: Logging releases on a different cadence from OpenShift Container Platform, the logging documentation is available as a separate documentation set at Red Hat OpenShift Logging.

Configuring OpenShift backup

Etcd is the key-value store for OpenShift Container Platform, which persists the state of all resource objects.

Back up your cluster's etcd data regularly and store in a secure location ideally outside the OpenShift Container Platform environment. Do not take an etcd backup before the first certificate rotation completes, which occurs 24 hours after installation, otherwise the backup will contain expired certificates. It is also recommended to take etcd backups during non-peak usage hours because the etcd snapshot has a high I/O cost.

Be sure to take an etcd backup before you update your cluster. Taking a backup before you update is important because when you restore your cluster, you must use an etcd backup that was taken from the

same z-stream release. For example, an OpenShift Container Platform 4.17.5 cluster must use an etcd backup that was taken from 4.17.5.

To start the etcd data backup, follow the below steps.

```
[admin@sec-rhel-9 ocp-sec]$ oc debug --as-root node/control-1
Starting pod/control-1-debug-bt77r ...
To use host binaries, run `chroot /host`
Pod IP: 10.61.178.101
If you don't see a command prompt, try pressing enter.
sh-5.1#
sh-5.1#
sh-5.1# chroot /host
```

(OPTIONAL) If the cluster-wide proxy is enabled, export the NO_PROXY, HTTP_PROXY, and HTTPS_PROXY environment variables by running the following commands:

```
export HTTP_PROXY=http://<your_proxy.example.com>:8080

export HTTPS_PROXY=https://<your_proxy.example.com>:8080

export NO_PROXY=<example.com>
```

Run the cluster-backup.sh script in the debug shell and pass in the location to save the backup.

Note: The cluster-backup.sh script is maintained as a component of the etcd Cluster Operator and is a wrapper around the etcdctl snapshot save command.

```
sh-5.1# /usr/local/bin/cluster-backup.sh /home/core/assets/backup
Certificate /etc/kubernetes/static-pod-certs/configmaps/etcd-all-bundles/server-ca-bundle.crt is
missing. Checking in different directory
Certificate /etc/kubernetes/static-pod-resources/etcd-certs/configmaps/etcd-all-bundles/server-
ca-bundle.crt found!
found latest kube-apiserver: /etc/kubernetes/static-pod-resources/kube-apiserver-pod-17
found latest kube-controller-manager: /etc/kubernetes/static-pod-resources/kube-controller-
manager-pod-6
found latest kube-scheduler: /etc/kubernetes/static-pod-resources/kube-scheduler-pod-6
found latest etcd: /etc/kubernetes/static-pod-resources/etcd-pod-10
19c91c8c3d1cc75929eac296079a1c07dfb7927ff695a9807687da8d98da3105
etcdctl version: 3.5.21
API version: 3.5
{"level":"info","ts":"2025-12-
03T04:52:54.240916Z","caller":"snapshot/v3_snapshot.go:65","msg":"created temporary db
file","path":"/home/core/assets/backup/snapshot_2025-12-03_045252.db.part"}
{"level":"info","ts":"2025-12-
03T04:52:54.258348Z","logger":"client","caller":"v3@v3.5.21/maintenance.go:212","msg":"opened
snapshot stream; downloading"}
{"level":"info","ts":"2025-12-
03T04:52:54.258379Z","caller":"snapshot/v3_snapshot.go:73","msg":"fetching
snapshot","endpoint":"https://10.61.178.101:2379"}
{"level":"info","ts":"2025-12-
03T04:52:54.559674Z","logger":"client","caller":"v3@v3.5.21/maintenance.go:220","msg":"completed
snapshot read; closing"}
{"level":"info","ts":"2025-12-
03T04:52:54.768773Z","caller":"snapshot/v3_snapshot.go:88","msg":"fetched
snapshot","endpoint":"https://10.61.178.101:2379","size":"98 MB","took":"now"}
{"level":"info","ts":"2025-12-
03T04:52:54.768842Z","caller":"snapshot/v3_snapshot.go:97","msg":"saved","path":"/home/core/asset
s/backup/snapshot_2025-12-03_045252.db"}
Snapshot saved at /home/core/assets/backup/snapshot_2025-12-03_045252.db
{"hash":1587407813,"revision":10565323,"totalKey":7169,"totalSize":98230272}
snapshot db and kube resources are successfully saved to /home/core/assets/backup
```

Two files are created in the /home/core/assets/backup directory.

```
sh-5.1# ls -lr /home/core/assets/backup/
total 96016
-rw-------. 1 root root    83696 Dec  3 04:52 static_kuberesources_2025-12-03_045252.tar.gz
```

```
-rw-------. 1 root root 98230304 Dec  3 04:52 snapshot_2025-12-03_045252.db
```

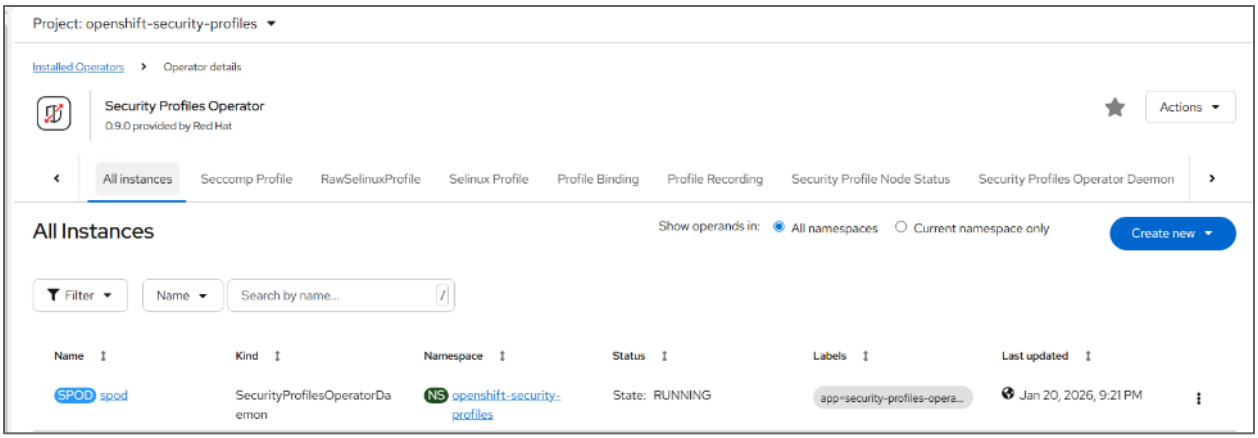
Security Profiles Operator

OpenShift Container Platform Security Profiles Operator (SPO) provides a way to define secure computing (seccomp) profiles and SELinux profiles as custom resources, synchronizing profiles to every node in a given namespace. For the latest updates, see the release notes.

The SPO can distribute custom resources to each node while a reconciliation loop ensures that the profiles stay up-to-date. OpenShift Container Platform administrators can use the Security Profiles Operator to define increased security measures in clusters.

Note: The Security Profiles Operator supports only Red Hat Enterprise Linux CoreOS (RHCOS) worker nodes. Red Hat Enterprise Linux (RHEL) nodes are not supported.

The installation of Security Profile Operator was done through OpenShift web console. You can also install it using CLI.



To enable enhanced logging verbosity, patch the spod configuration and adjust the value by running the command below.

```
oc -n openshift-security-profiles patch spod \
    spod --type=merge -p '{"spec":{"verbosity":1}}'
```

Note: The Security Profiles Operator supports the default logging verbosity of 0 and an enhanced verbosity of 1.

Sample SecompProfiles.

- audit.json
- violation.json
- fine-grained.json

Create a SecompProfile object for audit.

```
apiVersion: security-profiles-operator.x-k8s.io/v1beta1
kind: SeccompProfile
metadata:
  name: profile1
spec:
  defaultAction: SCMP_ACT_LOG
```

Verify the profile path.

```
[admin@sec-rhel-9 ocp-sec]$ oc get seccompprofile profile1 --output wide
```

```
NAME        STATUS      AGE   LOCALHOSTPROFILE
profile1    Installed   13m   operator/profile1.json
```

Create a patch.yaml file using the above info.

```
spec:
  template:
    spec:
      securityContext:
        seccompProfile:
          type: Localhost
          localhostProfile: operator/profile1.json
```

Create a pod and apply the newly created SecompProfile.

```
apiVersion: v1
kind: Pod
metadata:
  name: test-pod
spec:
  securityContext:
    runAsNonRoot: true
    seccompProfile:
      type: Localhost
      localhostProfile: operator/profile1.json
  containers:
    - name: test-container
      image: quay.io/security-profiles-operator/test-nginx-unprivileged:1.21
      securityContext:
        allowPrivilegeEscalation: false
        capabilities:
          drop: [ALL]
```

You can also apply the SecompProfile to a deployment.

Create a new namespace.

```
oc create ns my-namespace
```

Apply the deployment yaml.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test-nginx-deployment
  namespace: my-namespace
spec:
  replicas: 1
  selector:
    matchLabels:
      app: test-nginx
  template:
    metadata:
      labels:
        app: test-nginx
    spec:
      securityContext:
        runAsNonRoot: true
      containers:
        - name: test-container
          image: quay.io/security-profiles-operator/test-nginx-unprivileged:1.21
          securityContext:
            allowPrivilegeEscalation: false
            capabilities:
              drop: ["ALL"]
          ports:
            - containerPort: 80
```

Verify the deployment doesn’t have SecompProfile applied.

```
[admin@sec-rhel-9 ocp-sec]$ oc -n my-namespace get deployment test-nginx-deployment --
output=jsonpath='{.spec.template.spec.securityContext}' | jq .
{
  "runAsNonRoot": true
}
```

Apply the profile to the deployment using the pathc.yaml file.

```
 oc -n my-namespace patch deployment test-nginx-deployment --patch-file patch.yaml --type=merge
```

Verify the deployment now.

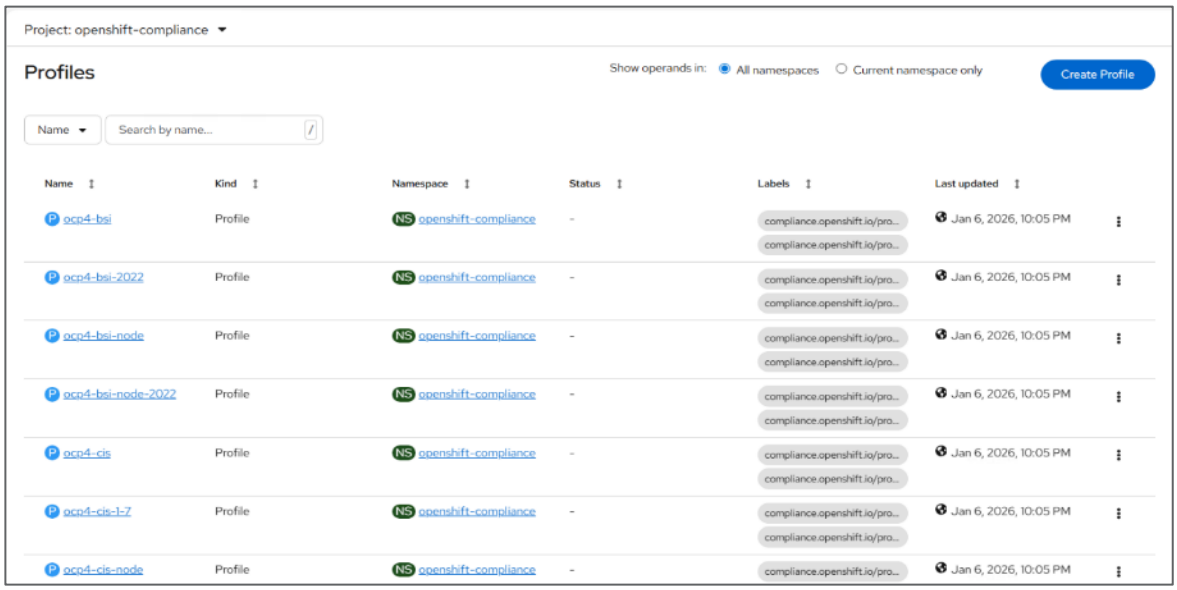
```
[admin@sec-rhel-9 ocp-sec]$ oc -n my-namespace get deployment test-nginx-deployment --
output=jsonpath='{.spec.template.spec.securityContext}' | jq .
{
  "runAsNonRoot": true,
  "seccompProfile": {
    "localhostProfile": "operator/profile1.json",
    "type": "Localhost"
  }
}
```

For more information about Security Profile Operator, refer here.

Compliance Operator on OpenShift

The Compliance Operator is designed for FIPS environments and enables OpenShift Container Platform administrators to define the desired compliance state of a cluster. It provides a clear overview of any gaps and actionable remediation steps. This operator evaluates compliance across both Kubernetes API resources within OpenShift and the nodes that power the cluster. Leveraging OpenSCAP—a NIST-certified tool—the Compliance Operator scans and enforces security policies based on provided content, ensuring adherence to stringent regulatory standards.

The compliance operator has multiple profile tabs. Each profile corresponds to a specific compliance benchmark and includes a prefix indicating the product it applies to. For example, ocp4-e8 applies the Essential 8 benchmark to OpenShift Container Platform, while rhcos4-e8 applies the same benchmark to

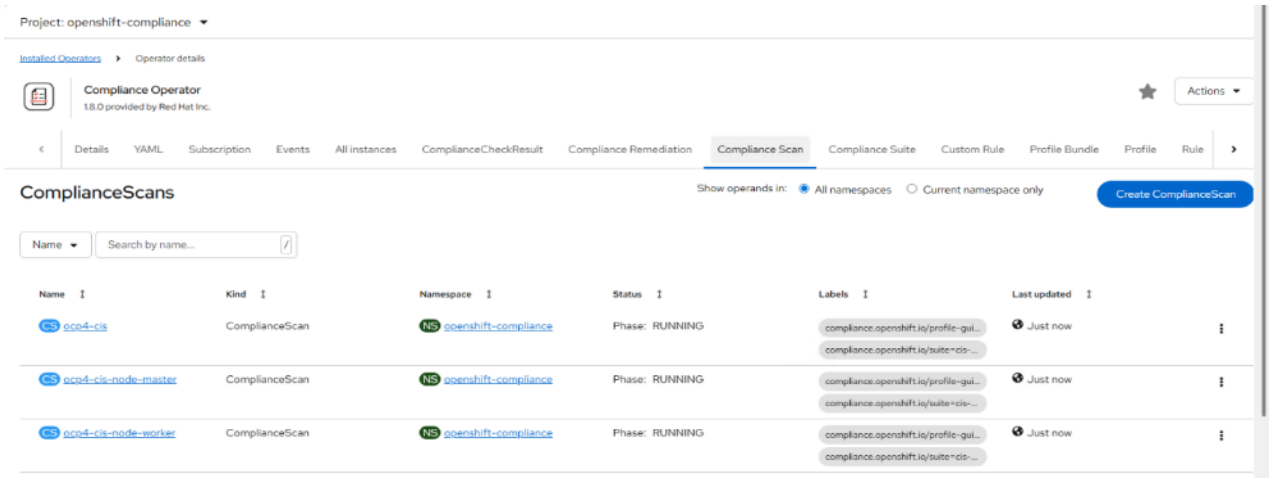


Red Hat Enterprise Linux CoreOS (RHCOS). Similarly, ocp4-cis represents the CIS Benchmark for OpenShift Container Platform v4.x, and ocp4-cis-node applies the CIS Benchmark to the operating system of the nodes running OpenShift Container Platform v4.x.

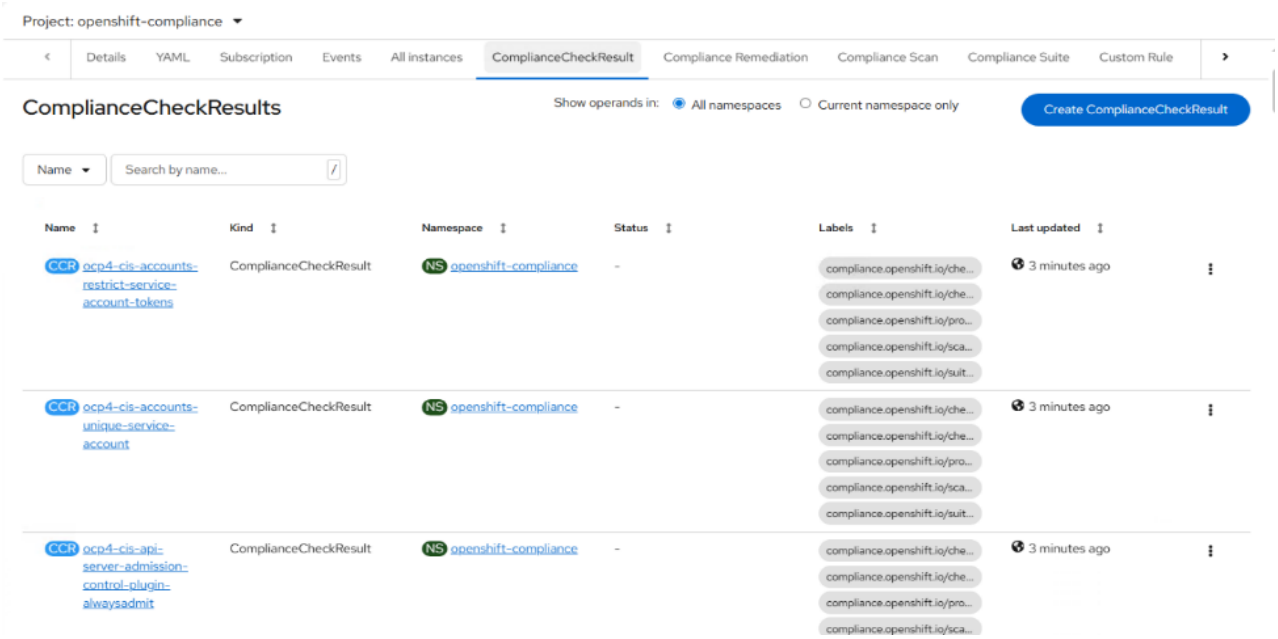
To run the scan, create Scan Binding Setting.

```
apiVersion: compliance.openshift.io/v1alpha1
kind: ScanSettingBinding
metadata:
  name: cis-compliance
  namespace: openshift-compliance
profiles:
  - name: ocp4-cis-node
    kind: Profile
    apiGroup: compliance.openshift.io/v1alpha1
  - name: ocp4-cis
    kind: Profile
    apiGroup: compliance.openshift.io/v1alpha1
settingsRef:
  name: default-auto-apply
  kind: ScanSetting
  apiGroup: compliance.openshift.io/v1alpha1
```

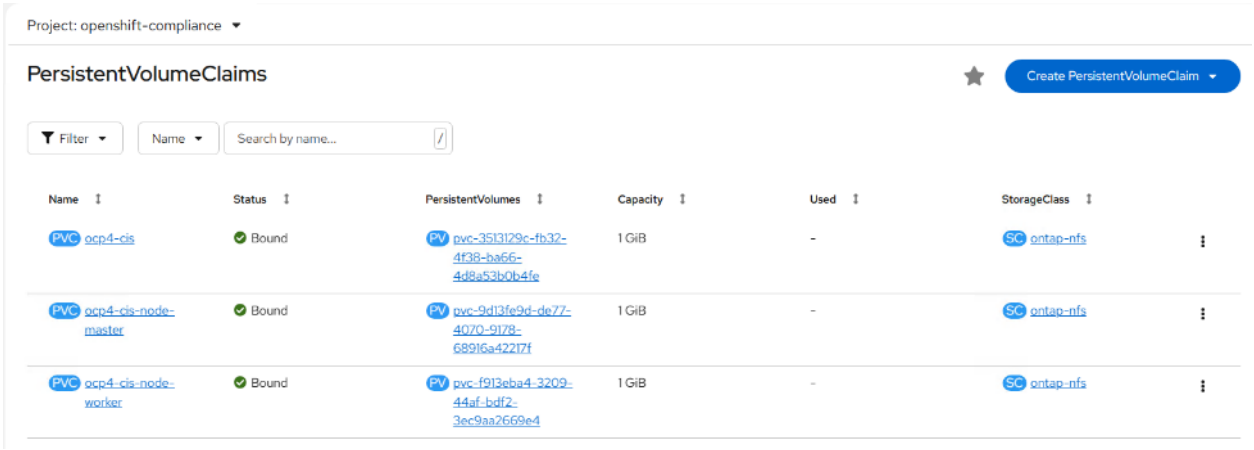
Compliance scan will start running.



Go to ComplianceCheckResults to verify the result.



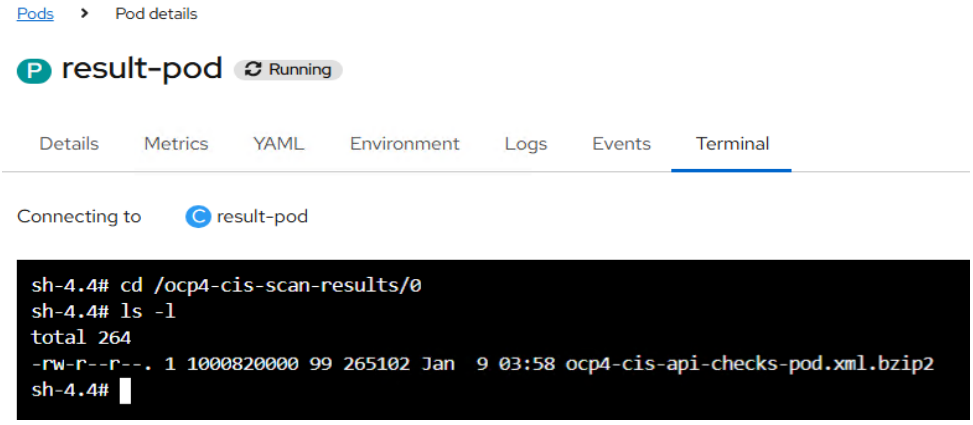
The compliance Operator creates 3 PVC in the openshift-compliance namespace, and the results will be stored in ocp4-cis PVC. We will create a test pod using ocp4-cis pvc to see the result file.



Below is the yaml file used to deploy the pod.

```
apiVersion: v1
kind: Pod
metadata:
  name: result-pod
spec:
  containers:
    - name: result-pod
      image: registry.access.redhat.com/ubi8/ubi
      command: ["sleep", "3000"]
      volumeMounts:
        - mountPath: "/ocp4-cis-scan-results"
          name: ocp4-cis-scan-vol
  volumes:
    - name: ocp4-cis-scan-vol
      persistentVolumeClaim:
        claimName: ocp4-cis
```

Go to the pod terminal and verify the cis file in directory ocp4-cis-scan-results/0.



Copy the result file to the management VM.

```
[admin@sec-rhel-9 compliance]$ oc cp result-pod:/ocp4-cis-scan-results/0 . -n openshift-
compliance
tar: Removing leading `/' from member names
[admin@sec-rhel-9 compliance]$
[admin@sec-rhel-9 compliance]$ ls
ocp4-cis-api-checks-pod.xml.bzip2
```

Install OpenSCAP tool to generate compliance report and bzip2 tool to decompress the result file.

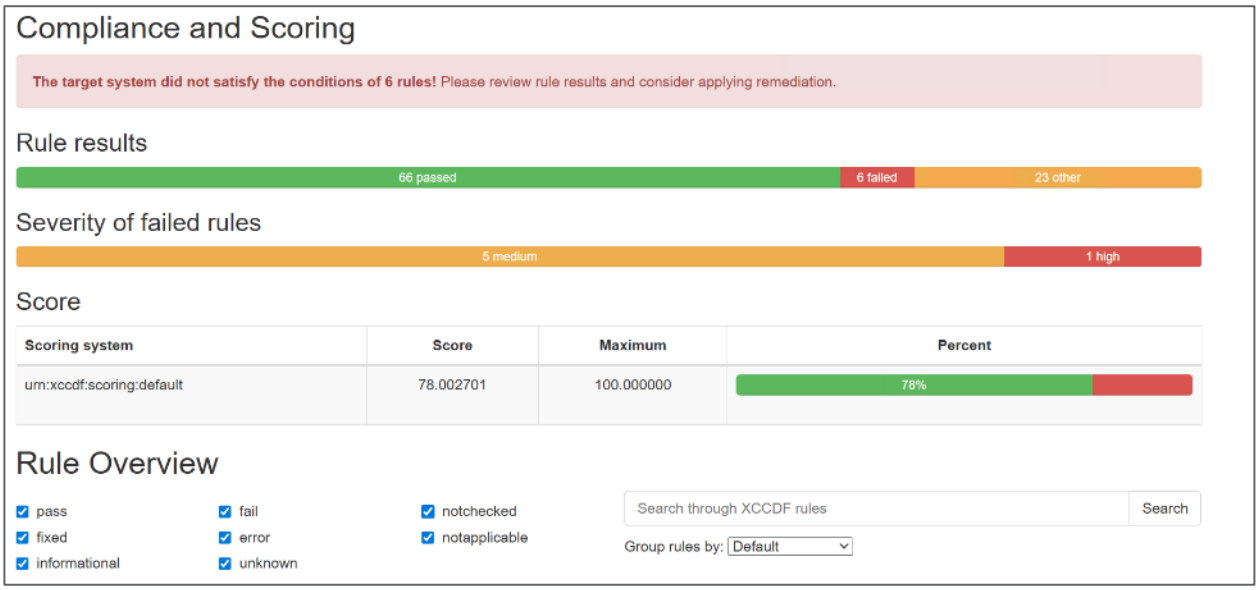
```
[admin@sec-rhel-9 compliance]$ yum install openscap-scanner -y
[admin@sec-rhel-9 compliance]$ yum install bzip2 -y
```

Now decompress the file and convert xml to html format.

```
[admin@sec-rhel-9 compliance]$ bunzip2 ocp4-cis-api-checks-pod.xml.bzip2
bunzip2: Can't guess original name for ocp4-cis-api-checks-pod.xml.bzip2 -- using ocp4-cis-api-
checks-pod.xml.bzip2.out
[admin@sec-rhel-9 compliance]$
[admin@sec-rhel-9 compliance]$ mv ocp4-cis-api-checks-pod.xml.bzip2.out ocp4-cis-api-checks-
pod.xml
[admin@sec-rhel-9 compliance]$
[admin@sec-rhel-9 compliance]$ oscap xccdf generate report ocp4-cis-api-checks-pod.xml > ocp4-
cis-api-checks-pod.html
[admin@sec-rhel-9 compliance]$
[admin@sec-rhel-9 compliance]$
[admin@sec-rhel-9 compliance]$ ls
ocp4-cis-api-checks-pod.html  ocp4-cis-api-checks-pod.xml
[admin@sec-rhel-9 compliance]$
```

Now download the HTML file and see the compliance report. Based on the report you can plan the remediation of the cluster.

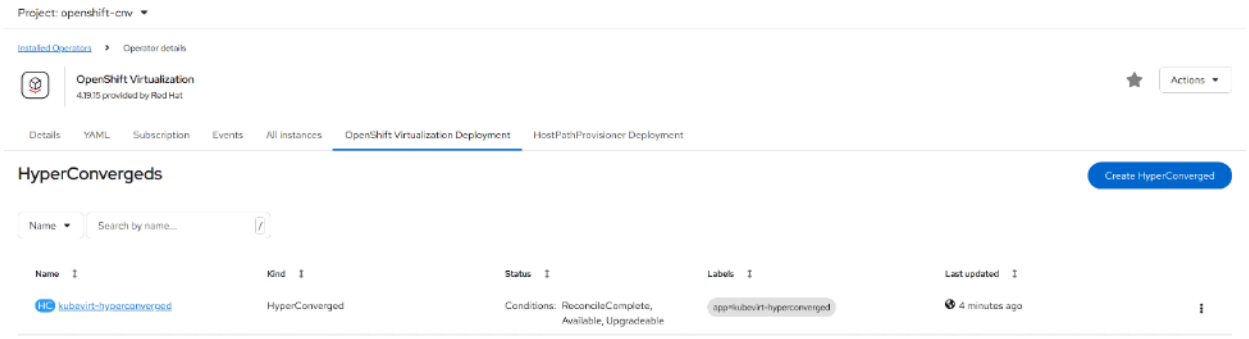
#Sample compliance report.



OpenShift virtualization setup and hardening

Red Hat OpenShift Virtualization

Red Hat OpenShift Virtualization is layered onto a Red Hat OpenShift Bare Metal environment by installing and configuring the OpenShift Virtualization Operator and a HyperConverged Deployment as shown below. By default, the OpenShift Virtualization Operator is deployed in the openshift-cnv namespace and initial VMs can be configured there, but before VMs can be configured, VM networking and a place to store VMs will need to be set up. With Red Hat OpenShift Virtualization, three Cisco virtual network interfaces (vNICs) are added, along with at least one VLAN for VM management. This VLAN is added to the Nexus switches, to the NetApp storage and to the UCS Domain Profile and worker node Server Profile.



This section provides prescriptive instructions for creating a more security-focused, standard configuration baseline for Red Hat® OpenShift® Virtualization. Prior to implementation, Cluster administrators should make sure that Red Hat OpenShift is properly hardened. Compliance Operator provides guidance on supported profiles for hardening both Red Hat OpenShift and Red Hat Enterprise Linux® CoreOS.

The scope of this section is limited to OpenShift Virtualization as a Red Hat OpenShift extension installed on top of the platform.

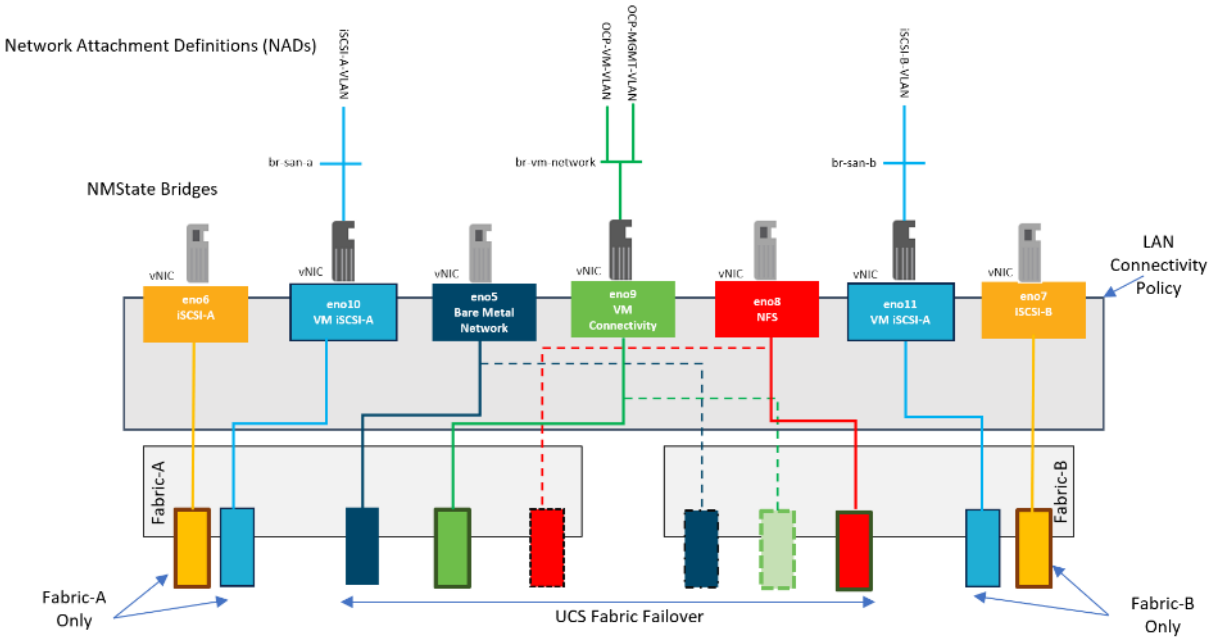
OpenShift Virtualization VM Networking

In the earlier Red Hat OpenShift Bare Metal setup, the NMState Operator was responsible for configuring and managing the networking on OpenShift nodes. In this validation with OpenShift Virtualization, NMState is similarly used, but now it provisions network bridges on the vNICs added to the UCS Server Profiles for virtualization workloads. On top of these bridges, Network Attachment Definitions (NADs) are created with the required VLAN IDs and MTU settings, enabling virtual machines to connect to the appropriate networks.

Figure 18) OpenShift Virtualization VM attachment to the network illustrates that 3 vNICs (eno9, eno10, and eno11) are added to the Worker Server Profile Template in addition to the 4 vNICs that were used for OpenShift on Bare Metal. vNIC eno9 is used for VM front end connectivity and multiple VLANs can be configured on that vNIC. It is required that those VLANs are configured in the Cisco Nexus switches and in the Cisco UCS Domain Profile VLAN policy and in the Ethernet Network Group policy attached to the vNIC in the LAN Connectivity policy. This vNIC is configured with Cisco UCS Fabric Failover which will fail the vNIC over to the other FI in case of an FI failure or reboot. vNICs eno10 and eno11 are identical to vNICs eno6 and eno7 and provide VM in-guest iSCSI connectivity. These vNICs do not have Fabric Failover configured. Additional iSCSI can be added to these vNICs in the same way VLANs are added to vNIC eno9. NMState Bridges are configured on eno9, eno10, and eno11 using Node Network Configuration Policies (NNCPs). Then, Network Attachment Definitions (NADs) are added, specifying the VLAN tag and the MTU (if 9000). VM NICs then reference the NAD and are attached to the correct VLAN.

All three of the added vNICs use the Ethernet Adapter policy with 16 RX queues and Receive Side Scaling (RSS) enabled to allow the RX queues to be serviced by different CPU cores.

Figure 18) OpenShift Virtualization VM attachment to the network



Note: If you intend to use NVMe-TCP, then NVMe-TCP VLANs can be added to the vNIC eno6, eno7, eno10 and eno11 along with iSCSI based on Fabric A/B topology and respective Bridge interfaces can be added in the NADs.

Note: NADs created in the default namespace are accessible cluster-wide to all namespaces, whereas NADs created in dedicated namespaces are restricted for use only within that specific namespace.

Dedicated VLANs to segment network traffic

To ensure that virtual machines (VMs) cannot access each other's network interfaces, it is essential to enforce complete network segmentation using dedicated VLANs. Assigning VMs to logically isolated VLANs provides strict Layer 2 (L2) separation, preventing unauthorized or unintended communication between workloads and reducing the risk of data leakage or security breaches.

In an OpenShift Virtualization environment, this segmentation is implemented not only at the VLAN level but also through NADs. Each NAD defines how a VM connects to a specific secondary network—typically backed by a bridge or OVN logical network—and includes parameters such as VLAN ID and MTU. By creating separate NADs for each VLAN, administrators ensure that VMs can only attach to the networks explicitly allowed for their namespace or workload type. This enforces consistent segmentation at the Kubernetes layer in addition to the underlying L2 boundary.

Verify if VLANs are configured for open virtual networking (OVN) Kubernetes and for bridge container network interface (CNI).

```
[admin@sec-rhel-9 ocp-sec]$ oc get network-attachment-definitions.k8s.cni.cncf.io -A -o json | jq
-r '.items[].spec.config'
{

    "cniVersion": "0.3.1",
    "name": "vlan-178",
    "type": "ovn-k8s-cni-overlay",
```

```
    "netAttachDefName": "default/vlan-178",
    "topology": "layer2",
    "vlanID": 178
}
{
    "cniVersion": "0.3.1",
    "name": "vlan-177",
    "type": "bridge",
    "bridge": "br-vm-network",
    "ipam": {},
    "macspoofchk": false,
    "preserveDefaultVlan": false,
    "vlan": 177
}
```

Note: Use dedicated VLANs as required.

Enable MAC spoof filtering

MAC spoof filtering is a security feature designed to prevent manipulator-in-the-middle attacks by validating the authenticity of Media Access Control (MAC) addresses in network traffic. By inspecting the MAC address of incoming packets, this mechanism ensures that only authorized devices can communicate with a specific device or network, reducing the risk of unauthorized access

Verify whether the NAD has MAC spoofing enabled.

```
[admin@sec-rhel-9 ocp-sec]$
oc get network-attachment-definitions.k8s.cni.cncf.io -A -o json \
| jq -r '
  .items[]
  | select(((.spec.config | fromjson).macspoofchk // false) == false)
  | "\(.metadata.namespace)/\(.metadata.name)"
'
default/vlan-178
openshift-cnv/vlan-177
test-vm/vlan-177
```

To activate MAC spoofing on VLAN 177, execute the command shown below.

```
oc patch network-attachment-definition vlan-177 -n test-vm \
--type='merge' \
-p '{"spec":{"config":"{\"cniVersion\":\"0.3.1\",\"name\":\"vlan-
177\",\"type\":\"bridge\",\"bridge\":\"br-vm-
network\",\"ipam\":{},\"macspoofchk\":true,\"preserveDefaultVlan\":false,\"vlan\":177}"}}'
```

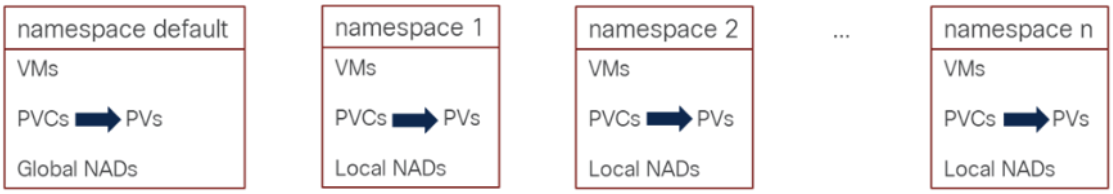
Verify after patching.

```
[admin@sec-rhel-9 ocp-sec]$ oc get network-attachment-definitions.k8s.cni.cncf.io -A -o json | jq
-r '                                                              .items[]
  | select((.spec.config | fromjson).macspoofchk == true)
  | "\(.metadata.namespace)/\(.metadata.name)"
'
test-vm/vlan-177
```

Grouping and separating VMs with namespaces or Projects

VMs can be created in the default openshift-cnv namespace or project, or in any other namespace. Using namespaces in this way allows VMs to be both grouped together and separated from other VMs. Figure 18 shows this grouping. With respect to NADs, as described earlier, NADs created within the default namespace are globally visible to VMs, where NADs created within other namespaces are only visible to VMs within that namespace. Storage persistent volume claims (PVCs) for VM disks are also contained within the namespace, but it is important to note that the corresponding persistent volumes (PVs) are tracked at the cluster level and not within namespaces.

Figure 19) VM grouping and separation using namespaces



Storage components

Proper storage configuration plays a critical role in maintaining security in virtualized environments. It enables organizations to define and enforce policies that govern how data is stored and accessed. By implementing these configurations, organizations can reduce the risk of unauthorized access to sensitive information and ensure data is managed in a security-conscious manner. Furthermore, modern storage platforms often provide built-in security features such as encryption and access controls, which can be leveraged to strengthen protection against potential threats.

Restrict access to cross-namespace dataVolumes cloning

Cross-namespace DataVolume cloning in OpenShift allows copying persistent data from one namespace to another. While this feature is useful for workflows like image distribution or backup, it introduces significant security risks if not properly controlled. Unrestricted cloning can expose sensitive data from one project or tenant to another, violating isolation principles. If a user in one namespace can clone from another without proper authorization, they effectively gain access to resources outside their scope.

Recommendation

- Do not create ClusterRoles that allow datavolumes/source globally.
- If cloning is required, grant access only to specific users or service accounts via RoleBindings in the source namespace.
- Regularly review RoleBindings and ClusterRoles for permissions on datavolumes/source.
- Remove any rolebinding resources that grant unintended access across namespaces.

Creating RBAC resources for cloning data volumes only between 2 namespaces.

Create a ClusterRole manifest using below yaml.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: datavolume-cloner
rules:
- apiGroups: ["cdi.kubevirt.io"]
  resources: ["datavolumes/source"]
  verbs: ["*"]
```

Create RoleBinding using below yaml.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: can-clone
  namespace: dev1
subjects:
- kind: ServiceAccount
  name: default
  namespace: dev2
roleRef:
  kind: ClusterRole
```

```
  name: datavolume-cloner
  apiGroup: rbac.authorization.k8s.io
```

Disable shareable disks

Sharable disks are shared between VMs, and they can be used only if a file system installed on top of the VM is a cluster file system, or the application using the device is distributed and cloud aware. The incorrect usage of shareable disks might cause data corruption. Sharing system resources increases the risk of unauthorized access to data, manipulation of data flow restrictions, and could lead to corruption and data loss.

You can verify the VM volumes to make sure the shareable flag is not set to true by using the command below.

```
oc get vm -A -o json \
| jq -r '
  .items[]
  | select(any(.spec.template.spec.domain.devices.disks[]?; .shareable == true))
  | .metadata.namespace + "/" + .metadata.name
'
```

Note: Shareable disks are not enabled by default

Input/Output (I/O) error policy

Error policies for disks control how input/output (I/O) errors are handled. If a read or write operation fails on the storage, an I/O error occurs. Setting the error policy to “ignore” is not recommended because it makes it difficult to determine the root cause of any I/O issues, which can lead to data loss if the data is not correctly written to the disk.

You can verify the policy by using the command below.

```
oc get vm -A -o json \
| jq -r '
  .items[]
  | select(any(.spec.template.spec.domain.devices.disks[]?; .errorPolicy == "ignore"))
  | .metadata.namespace + "/" + .metadata.name
```

Note: By default, the errorPolicy is not set on any disk and the default value is stop.

Platform configuration

Restrict GPU and USB pass-through

The ability to pass through devices provides the capability to offload tasks from the central processing unit (CPU) to the device itself. Restricting pass-through to approved devices reduces the risk of unauthorized or unintended data sharing or transmission introduced by allowing graphics processing unit (GPU) and USB connectivity.

To list the host devices available to virtualization workloads and verify the host devices.

```
oc get hyperconverged kubevirt-hyperconverged -n openshift-cnv \
-ojsonpath='{.spec.permittedHostDevices}'
```

Note: By default, OpenShift Virtualization doesn’t configure any pass-through devices.

To remove all permitted devices, run the following.

```
oc patch hyperconverged kubevirt-hyperconverged -n openshift-cnv \
  --type='json' \
  -p='[
    { "op": "remove", "path": "/spec/permittedHostDevices" }
  ]'
```

Disable persistentReservations feature gate

Persistent reservations are used to reserve shared logical unit numbers (LUN) among multiple VMs. This feature is required when the Windows guest makes use of the Windows Shared Cluster Filesystem. Enabling persistent reservations introduces additional complexity and overhead as it requires calculated management of resource allocation and monitoring to make sure reservations are not overcommitted and do not influence the overall performance of the system.

```
[admin@sec-rhel-9 ocp-sec]$ oc get hyperconverged kubevirt-hyperconverged -n openshift-cnv -o
jsonpath='{.spec.featureGates.persistentReservation}'
false
[admin@sec-rhel-9 ocp-sec]$
```

Note: By default, persistentReservations is disabled and set to false. It is recommended to be enabled only when planning to use Windows Shared Cluster Filesystem or similar applications

Disable downwardMetrics feature gate

The downwardMetrics feature allows users to collect and monitor additional metrics related to host and VM performance. Rationale: Enabling the “downwardMetrics” feature introduces the risk of unauthorized or unintended sharing of information, as well as manipulation of information flow restrictions. The additional metrics include sensitive performance data, which could aid in the reconnaissance activities of a malicious actor.

Verify that the downwardMetrics feature gate is disabled.

```
[admin@sec-rhel-9 ocp-sec]$ oc get hyperconverged kubevirt-hyperconverged -n openshift-cnv -o
jsonpath='{.spec.featureGates.downwardMetrics}'
false
```

Require the use of trusted registries secured with TLS

Transport Layer Security (TLS) is a cryptographic protocol used in this context to protect data in transit. Restricting operations to the use of trusted registries ensures the use of approved container images. By only pulling container images from trusted registries, organizations can reduce the risk of introducing unknown vulnerabilities or malicious software into their systems. This helps make sure applications and systems remain secure and stable.

Command to view any insecure registries.

```
oc get hyperconverged kubevirt-hyperconverged -n openshift-cnv -o
jsonpath='{.spec.storageImport.insecureRegistries}'
```

Note: TLS is enabled by default. No insecure registries are configured or enabled by default.

Disable exec access to the pods

The ability to exec commands in a pod allows for arbitrary execution by users. This includes administrative functions which normally require elevation of privileges by an approved administrator and could lead to unauthorized use of both security- and non-security-related administrative functions.

To verify who can exec commands in pods, us ethe command below.

```
oc adm policy who-can exec pod
```

Use RBAC to limit exec permissions, avoid granting pods/exec to broad roles like developer or view and restrict trusted admins only.

Restrict VNC access to cluster workloads

Access to each workload’s virtual screen is provided via virtual network computing (VNC). Permission to use VNC is granted via a role, which means users with this role assigned can access the VNC console of all workloads in a namespace. Access to use VNC should be carefully considered as it is usually preferable to access workloads via other means which support a stronger encryption and authentication mechanism.

```
oc get rolebinding -A -o json \
| jq -c '.items[] | select(.roleRef.name | contains("token.kubevirt.io:generate"))'
```

No RoleBindings currently reference a role with token.kubevirt.io:generate in its name.
This implies that no users have been granted VNC token generation permissions.

Multi-tenant configuration

The multi-tenant architecture built on a base FlexPod foundation delivers a secure, scalable, and flexible platform capable of hosting multiple isolated tenants. Red Hat OpenShift can be deployed on this FlexPod infrastructure as one of the tenants alongside other supported hypervisors and operating systems. Additional multi-tenancy can then be implemented natively within OpenShift.

In a FlexPod multi-tenant configuration, the initial FlexPod deployment effectively becomes the first infrastructure tenant. For guidance on this baseline setup, refer to the FlexPod Datacenter with Red Hat OCP Bare Metal Manual Configuration with Cisco UCS X-Series Direct design. This solution uses the same model: first deploying the base FlexPod, then installing Red Hat OpenShift Container Platform on bare metal, and eventually adding the OpenShift Virtualization layer on top of the OpenShift environment.

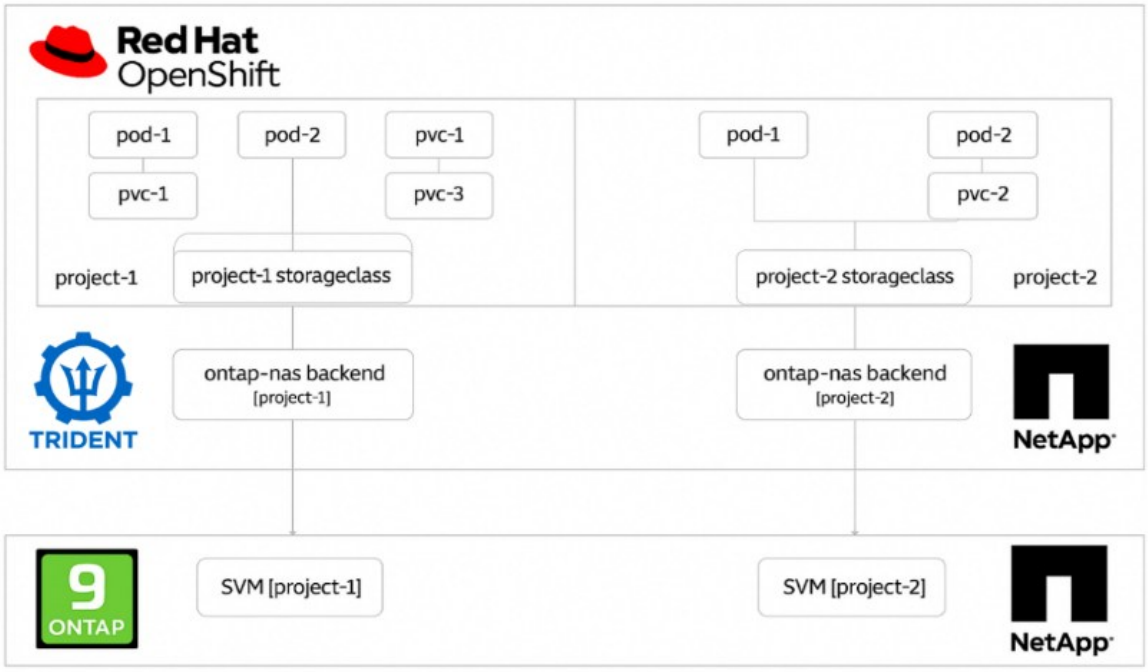
OpenShift provides built-in, Kubernetes-native mechanisms for multi-tenancy that allow multiple teams, applications, or customers to securely share a single cluster while maintaining strong isolation, security, and operational efficiency. Multitenancy is an architecture that allows multiple tenants to coexist on the same cluster with proper isolation of resources, security, and so on. In this context, a tenant can be viewed as a subset of the cluster resources that are configured to be used by a particular group of users for an exclusive purpose. Configuring multitenancy on a Red Hat OpenShift cluster provides the following advantages:

- A reduction in CapEx and OpEx by allowing cluster resources to be shared
- Securing the workloads from cross-contamination of security breaches
- Protection of workloads from unexpected performance degradation due to resource contention

Multitenancy within OpenShift cluster and storage

fully enable multitenancy in an OpenShift cluster, administrators must define quotas and enforce restrictions across key resource domains—including compute, storage, networking, and security—so that each tenant receives properly isolated and governed resources; while this solution touches on several of these areas, its primary focus is on best practices for securing and isolating data consumed or produced by multiple workloads within the same Red Hat OpenShift cluster through multitenant storage configurations dynamically provisioned by Trident and backed by NetApp ONTAP. Figure 20) Multi-tenant in OpenShift, NetApp ONTAP and Trident illustrates the multitenancy in Red Hat OpenShift, NetApp ONTAP and Trident.

Figure 20) Multi-tenant in OpenShift, NetApp ONTAP and Trident



Red Hat OpenShift cluster resources

From the perspective of a Red Hat OpenShift cluster, the foundational construct for implementing multitenancy is the project. A project effectively partitions the cluster into multiple virtual workspaces, providing the initial layer of isolation required for tenant separation.

Building on this, administrators must configure RBAC to control access within and across these projects. A recommended best practice is to group all developers associated with a specific workload or project into a corresponding user group within the organization's Identity Provider (IdP). OpenShift's integration with external IdPs enables seamless synchronization of users and groups, allowing these identities to be imported directly into the cluster. This approach enables administrators to assign project-specific access only to the appropriate user groups, ensuring secure separation of resources and preventing unauthorized access to workloads belonging to other tenants.

Red Hat OpenShift virtualization networking

NADs play a central role in enabling advanced networking for OpenShift Virtualization while also reinforcing multitenancy boundaries within the cluster. NADs are namespace-scoped, each OpenShift project (tenant) maintains its own set of network definitions, ensuring that networks are isolated by default and cannot be shared or viewed across namespaces, aligning naturally with OpenShift's project-based multitenancy model. While NADs define and attach these additional interfaces, NetworkPolicies continue to govern how workloads communicate on the default cluster network, providing critical east-west traffic isolation between tenants even when multiple NICs are present. Together, NADs and NetworkPolicies enable OpenShift to deliver a robust multitenant virtualized environment—where each tenant's VMs can operate independently, securely, and with tailored network performance characteristics—while maintaining strict boundaries that prevent unauthorized cross-tenant communication.

NetApp ONTAP

To ensure strong isolation of shared persistent storage in a Red Hat OpenShift cluster, it is essential that volumes created for each project appear to the hosts as though they originate from entirely separate storage environments. The recommended approach is to provision a dedicated Storage Virtual Machine (SVM) in NetApp ONTAP for each project or workload and put them in a separate IPSpace, thereby creating independent storage domains that cleanly separate tenant data and operations. Also, create Trident user for each SVM with only appropriate role and a dedicated export policy for each SVM.

NetApp Trident

Once separate SVMs have been created in NetApp ONTAP for each OpenShift project, each SVM must be associated with its own Trident backend. The Trident backend configuration defines how persistent storage is allocated to cluster resources and requires the SVM details—along with at least the appropriate protocol driver—to be specified. Additionally, the backend configuration can include optional parameters that control volume-provisioning behaviour, such as volume size limits, aggregate usage constraints, and other storage policies. Detailed guidance on defining Trident backends is available in the Trident documentation.

Below is a sample Trident backend for ontap-nas driver created for project-1 namespace.

```
# cat tbc-nas-project-1.yaml

apiVersion: v1
kind: Secret
metadata:
  name: tbc-nas-secret
  namespace: trident
type: Opaque
data:
  username: <base64-encoded-svm-username>
  password: <base64-encoded-svm-password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-nas-project1
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: <mgmt-lif>
  dataLIF: <nfs-lif>
  backendName: ontap-nas-project-1
  svm: svm-project-1
  defaults:
    exportPolicy: project-1-worker
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
  credentials:
    name: tbc-nas-secret
  labels:
    tenant: “project-1”
```

After configuring the Trident backends, the next step is to configure StorageClasses. Configure as many storage classes as there are backends, providing each storage class access to spin up volumes only on one backend. We can map the StorageClass to a particular Trident backend by using the labels parameter while defining the storage class. Thus, there is a one-to-one mapping from StorageClass to Trident backend which points back to one SVM. This ensures that all storage claims via the StorageClass assigned to that project are served by the SVM dedicated to that project only.

Configure Kubernetes StorageClass object and create the storage class to instruct Trident how to provision volumes for each tenant. Kubernetes StorageClass objects are specified by name in PersistentVolumeClaims to provision storage with a set of properties. The storage class itself identifies

the provisioner to be used and defines that set of properties in terms the provisioner understands. It is one of two basic objects that need to be created and managed by the administrator. The other is the Trident backend object.

Below is a sample Kubernetes StorageClass object for project-1 that Trident uses. Here the StorageClass has selector that points to the project-1 backend which was created in the previous steps.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-1-sc
  annotations:
    # Uncomment the next line if you want this to be the default class
    # storageclass.kubernetes.io/is-default-class: "true"
provisioner: csi.trident.netapp.io

# Allow PVCs to be grown after creation (supported by Trident/ONTAP)
allowVolumeExpansion: true

# Bind PVs only after a Pod is scheduled (helps multi-zone correctness)
volumeBindingMode: WaitForFirstConsumer

# Optional NFS tuning; adjust to your environment
mountOptions:
  - vers=4.2

parameters:
  # Trident driver family to use (e.g., ontap-nas, ontap-nas-flexgroup, ontap-san)
  backendType: "ontap-nas"

  # Ask Trident to create thin-provisioned volumes on ONTAP
  provisioningType: "thin"
  selector: "tenant=project-1"   # matches the backend labels above
```

Similarly, you can create Trident backend using SVM and export policy details and StorageClass for the other project. To understand more about Kubernetes StorageClass object refer to the Trident documentation.

Support Information and Advisory

Support information

Cisco Intersight Email Notifications

Intersight provides fault monitoring capabilities to track and set up alarms for all managed targets. An alarm alerts you about a failure in the endpoint (a fault) or a threshold that has been crossed. An alarm in Intersight includes information about the operational state of the affected object at the time the fault was raised.

Manually checking new alarms can be a time-consuming and disruptive task, distracting you from current work. Determining which alarms require attention is also a demanding task. Email notifications automatically poll for recent alarms, determine their severity, and direct especially concerning ones to a user's email address based on a rule you create.

You will receive alerts when an alarm transitions between the Cleared severity level and any Active severitylevel—such as Critical, Warning, or Informational, according to the rule you have configured.

Notifications are based on a rule that are set for the incoming alarms. You can define specific filter conditions for email notifications, ensuring that the appropriate team members are notified about issues requiring their attention. By setting up these filters, you can tailor notifications to specific criteria.

Using filter conditions, you can:

- Customize alarm notifications based on specific parameters, such as host type, domain, or server name.
- Ensure alarms are sent to designated individuals or teams only when predefined conditions are met.

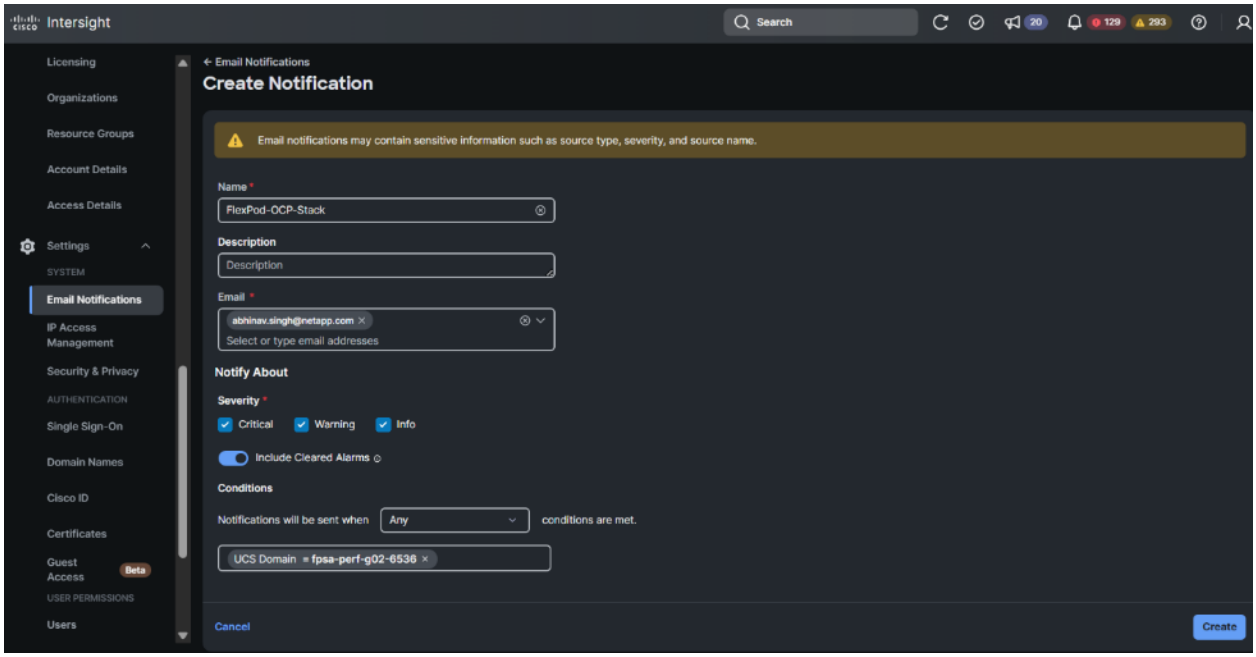
Log in to Cisco Intersight, Expand Settings > select Email Notifications.

Enter all the information required.

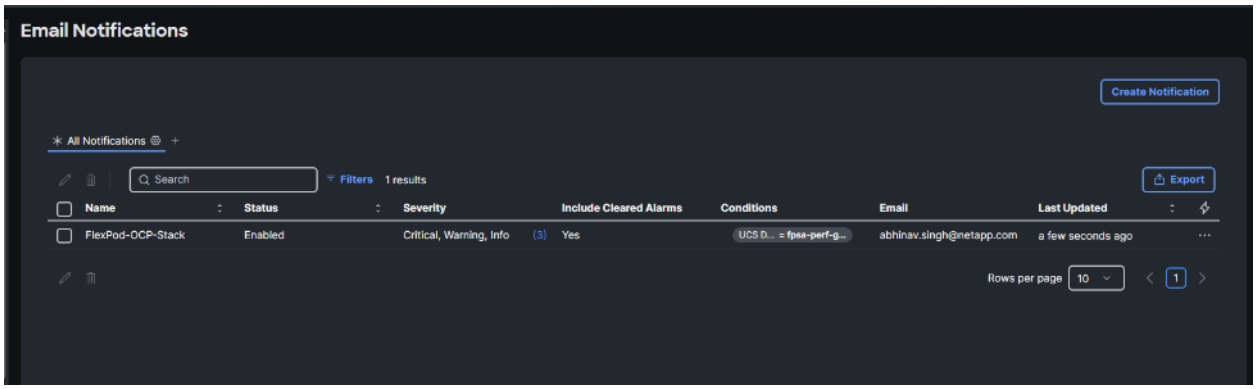
- Name of the rule
- Description (optional)
- Email- The email address(es) to which notifications will be sent.
- Severity
    - Critical
    - Warning
    - Informational (info)
- Include Cleared Alarms - A setting of Yes indicates that notifications will be sent when alarms are resolved and marked as cleared. A setting of No indicates that notifications will not be sent for cleared alarms.
- Conditions- The filter condition(s) to which the notification rule applies.

  If multiple severity levels are selected, the least severity level among them will trigger the notification email when it is reached.

For more information on severity levels, see the Alarms in Intersight section in **Cisco Intersight Alarms Reference Guide**.



Note: Including alarms that have transitioned to a cleared state provides visibility into recently resolved issues. This can help with auditing, troubleshooting, or monitoring patterns.



Cisco Nexus Smart Call Home

Cisco Nexus Smart Call Home provides an email-based notification for critical system policies. You can use the Call Home feature to email a network operations center, page a support engineer, or use Cisco Smart Call Home services to open a case with the Technical Assistance Center (TAC).

Smart Call Home provides secure message transport from devices, continuous device health monitoring and real-time diagnostics alerts, Smart Call Home message analysis. It facilitates automatic execution and attachment of relevant CLI command output to speed up issue resolution. It supports multiple message formats, such as short text, full text, and XML formats to serve different communication needs.

To register for Smart Call Home, you will need your SMARTnet contract and your contact information. Please refer to the Cisco Nexus Smart Call Home documentation for details on the pre-requisites, configuration details, and the supported alert group and command output collection information.

Cisco Connected TAC

Connected TAC is Cisco's proactive, automated technical support capability that securely connects your devices to Cisco's cloud (typically via Cisco Intersight) to provide:

- Automated tech-support file collection
- Proactive issue detection and case creation
- Faster diagnostics using Cisco's digitized expertise
- Telemetry-based health insights and recommendations

Connected TAC is a service that uses secure, bidirectional connectivity to analyze device data and identify potential problems before they impact the network. Connected TAC is also included as part of the Intersight ecosystem for connected data-center devices. Devices are connected to the Intersight portal through a NXDC that is embedded in the Cisco NX-OS image of each system.

Beginning with Cisco NX-OS Release 10.2(3)F on Nexus switches and Cisco MDS 9000 NX-OS Release 9.3(2) on MDS platforms, the Device Connector capability is supported. Devices are connected to the Intersight portal through a Nexus Switch Device Connector (NXDC) that is embedded in the Cisco NX-OS image of each system. This feature enables a secure, bidirectional communication channel through which connected devices transmit telemetry data and receive operational directives from the Cisco Intersight cloud service over an encrypted Internet connection.

Furthermore, standalone Cisco Nexus switches, Cisco MDS switches, and Cisco Nexus Dashboard (ND) systems can all establish connectivity with Cisco Intersight to activate Connected TAC functionality, thereby enabling automated diagnostic data collection, proactive issue detection, and enhanced technical assistance workflows.

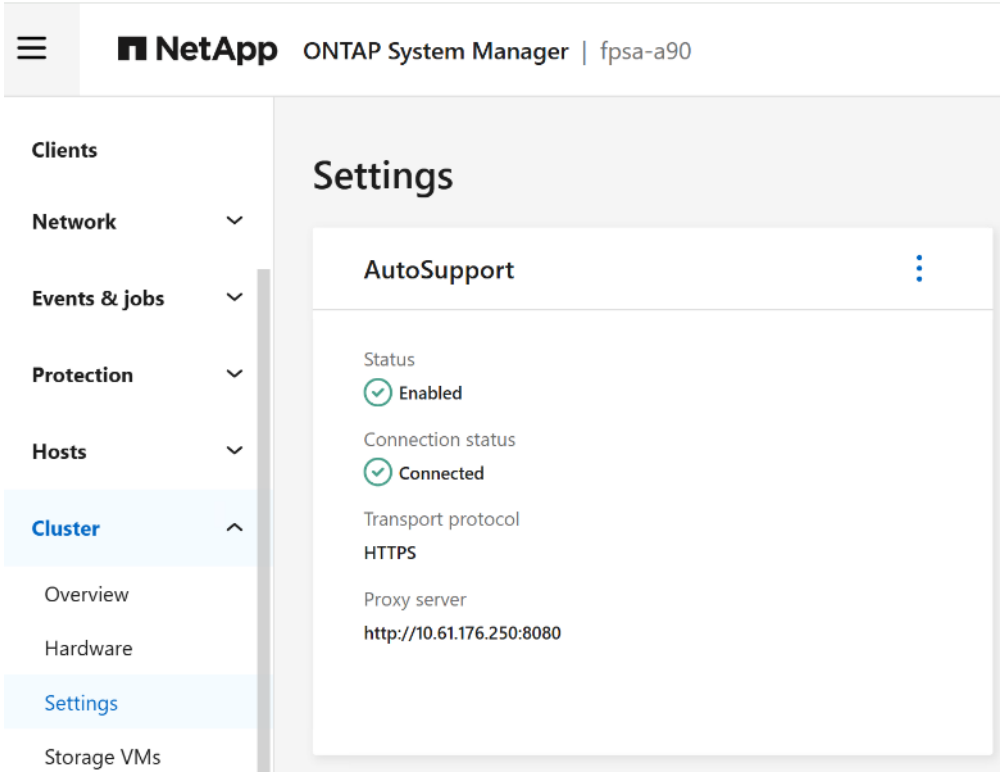
NetApp ONTAP support information

ONTAP AutoSupport® is a mechanism that proactively monitors the health of your ONTAP system and automatically sends messages to NetApp technical support, your internal support organization, or a support partner.

The AutoSupport component of ONTAP collects telemetry data and sends it for analysis. Active IQ Digital Advisor analyzes the data from AutoSupport and provides proactive care and optimization. Using artificial intelligence, Active IQ can identify potential problems and help you resolve them before they impact your business. By default, the system collects AutoSupport information and stores it locally, even if you disable AutoSupport. While you can disable AutoSupport at any time, you should leave it enabled.

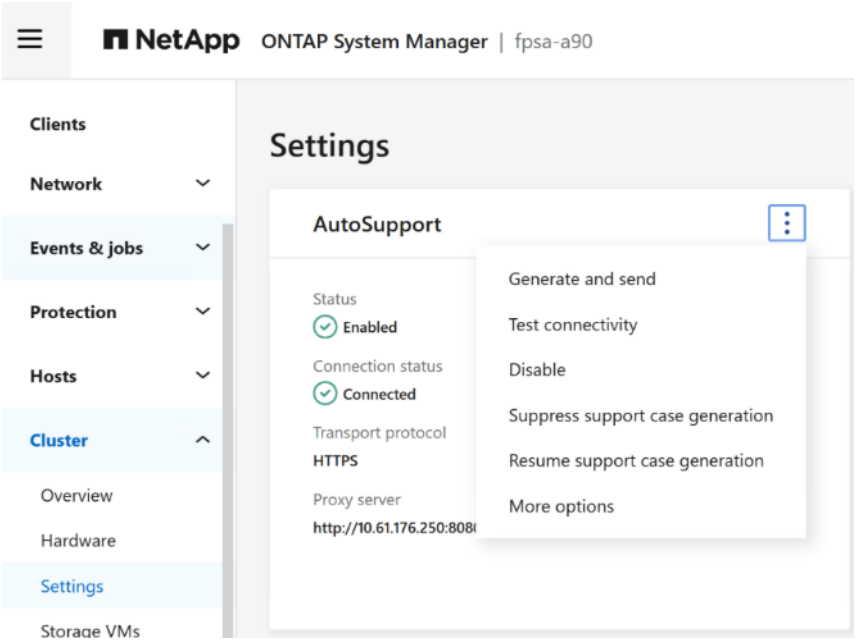
While AutoSupport messages to technical support are enabled by default, you must set the correct options and have a valid mail host to have messages sent to your internal support organization. Only the cluster administrator can perform AutoSupport management. The storage virtual machine (SVM) administrator has no access to AutoSupport.

To configure AutoSupport from the ONTAP System Manager, go to Cluster > Settings, click on the ellipsis in the AutoSupport tile, select More options to configure the settings for AutoSupport, such as transport protocol, proxy server, mail host, email sender and recipients.



Note: The HTTPS transport protocol is the default and the recommended protocol for sending AutoSupport.

In addition to automatically generated AutoSupport, you can also generate AutoSupport on demand by clicking on the ellipsis and selecting Generate and Send action.



OpenShift support information

OpenShift Container Platform collects telemetry and configuration data about your cluster and reports it to Red Hat by using the Telemeter Client and the Insights Operator. Red Hat uses this data to understand and resolve issues in a connected cluster. Similar to connected clusters, you can Use remote health monitoring in a restricted network. OpenShift Container Platform collects data and monitors health using the following:

- **Telemetry**: The Telemetry Client gathers and uploads the metrics values to Red Hat every four minutes and thirty seconds. Red Hat uses this data to:
    - Monitor the clusters.
    - Roll out OpenShift Container Platform upgrades.
    - Improve the upgrade experience.
- **Insights Operator**: By default, OpenShift Container Platform installs and enables the Insights Operator, which reports configuration and component failure status every two hours. The Insights Operator helps to:
    - Identify potential cluster issues proactively.
    - Provide a solution and preventive action in Red Hat OpenShift Cluster Manager.

You can review telemetry information.

If you have enabled remote health reporting, Use Insights to identify issues with your cluster. You can optionally disable remote health reporting.

Gathering data about your OpenShift cluster

When opening a support case, it is helpful to provide debugging information about your cluster to Red Hat Support. The **oc adm must-gather** CLI command collects the information from your cluster that is most likely needed for debugging issues.

For more information on gathering data about OpenShift Cluster, refer to https://docs.redhat.com/en/documentation/openshift_container_platform/4.19/html/support/gathering-cluster-data#support_gathering_data_gathering-cluster-data

```
[admin@sec-rhel-9 ocp-sec]$ oc adm must-gather

**********OutPut Omited*************
must-gather-cgbfw] POD 2025-11-27T04:22:31.952926170Z INFO: Image with low level tools to use:
quay.io/openshift-release-dev/ocp-v4.0-art-
dev@sha256:1e08254e19e6ff79986f34b6c9382856a25abaab8ec4d6086914f70021c25e89
[must-gather-cgbfw] POD 2025-11-27T04:22:32.011059334Z INFO: Getting status/tsdb from prometheus-
k8s-1
[must-gather-cgbfw] POD 2025-11-27T04:22:32.044250148Z daemonset.apps/perf-node-gather-daemonset
created
[must-gather-cgbfw] POD 2025-11-27T04:22:32.058991252Z volume usage percentage 0
[must-gather-cgbfw] POD 2025-11-27T04:22:32.136713608Z Waiting for performance profile collector
pods to become ready: 1
[must-gather-cgbfw] POD 2025-11-27T04:22:32.344239542Z INFO: Getting status from alertmanager-
main-0
[must-gather-cgbfw] POD 2025-11-27T04:22:32.518086322Z Gathering data for ns/assisted-
installer...
[must-gather-cgbfw] POD 2025-11-27T04:22:32.570952945Z INFO: Worker host service log collection
to complete.
[must-gather-cgbfw] POD 2025-11-27T04:22:32.665110257Z Gathering data for ns/openshift-machine-
api...
[must-gather-cgbfw] POD 2025-11-27T04:22:33.229841464Z Waiting for performance profile collector
pods to become ready: 2
[must-gather-cgbfw] POD 2025-11-27T04:22:33.686828308Z Wrote inspect data to must-gather.
[must-gather-cgbfw] POD 2025-11-27T04:22:34.322152409Z Waiting for performance profile collector
pods to become ready: 3
[must-gather-cgbfw] POD 2025-11-27T04:22:35.414896490Z Waiting for performance profile collector
pods to become ready: 4
[must-gather-cgbfw] POD 2025-11-27T04:22:36.514484680Z Waiting for performance profile collector
pods to become ready: 5
[must-gather-cgbfw] POD 2025-11-27T04:22:37.067741282Z volume usage percentage 0
[must-gather-cgbfw] POD 2025-11-27T04:22:37.606554649Z Waiting for performance profile collector
pods to become ready: 6
[must-gather-cgbfw] POD 2025-11-27T04:22:38.693733677Z Waiting for performance profile collector
pods to become ready: 7
[must-gather-cgbfw] POD 2025-11-27T04:22:39.788747476Z Waiting for performance profile collector
pods to become ready: 8
[must-gather-cgbfw] POD 2025-11-27T04:22:40.881747375Z Daemonset perf-node-gather-daemonset ready
5 out of 5
[must-gather-cgbfw] POD 2025-11-27T04:22:41.093961793Z Collecting performance related data for
node compute-1
[must-gather-cgbfw] POD 2025-11-27T04:22:41.098946172Z Collecting performance related data for
node control-3
[must-gather-cgbfw] POD 2025-11-27T04:22:41.103439166Z Collecting performance related data for
node compute-2
[must-gather-cgbfw] POD 2025-11-27T04:22:41.108515650Z Collecting performance related data for
node control-2
[must-gather-cgbfw] POD 2025-11-27T04:22:41.113198193Z Collecting performance related data for
node control-1
[must-gather-cgbfw] POD 2025-11-27T04:22:41.349790329Z Gathering data for ns/openshift-cloud-
controller-manager-operator...
[must-gather-cgbfw] POD 2025-11-27T04:22:42.038840268Z Gathering data for ns/openshift-cloud-
controller-manager...
[must-gather-cgbfw] POD 2025-11-27T04:22:42.077497094Z volume usage percentage 0
[must-gather-cgbfw] POD 2025-11-27T04:22:42.253934436Z Gathering data for ns/openshift-cloud-
credential-operator...
[must-gather-cgbfw] POD 2025-11-27T04:22:43.165093143Z Gathering data for ns/openshift-config-
operator...
[must-gather-cgbfw] POD 2025-11-27T04:22:43.360428624Z Gathering data for ns/openshift-console-
operator...
[must-gather-cgbfw] POD 2025-11-27T04:22:43.574640813Z Gathering data for ns/openshift-console...
[must-gather-cgbfw] POD 2025-11-27T04:22:44.242647209Z Gathering data for ns/openshift-cluster-
storage-operator...
[must-gather-cgbfw] POD 2025-11-27T04:22:44.422130283Z Gathering data for ns/openshift-dns-
operator...
[must-gather-cgbfw] POD 2025-11-27T04:22:44.578482305Z Gathering data for ns/openshift-dns...
[must-gather-cgbfw] POD 2025-11-27T04:22:44.842360772Z Gathering data for ns/openshift-etcd-
operator...
[must-gather-cgbfw] POD 2025-11-27T04:22:45.445244365Z Gathering data for ns/openshift-etcd...
[must-gather-cgbfw] POD 2025-11-27T04:22:47.088360163Z volume usage percentage 0
```

```
[must-gather-cgbfw] POD 2025-11-27T04:22:48.795779915Z Gathering data for ns/openshift-image-
registry...
[must-gather-cgbfw] POD 2025-11-27T04:22:49.571319096Z Gathering data for ns/openshift-ingress-
operator...
[must-gather-cgbfw] POD 2025-11-27T04:22:49.831455762Z Gathering data for ns/openshift-ingress-
canary...
[must-gather-cgbfw] POD 2025-11-27T04:22:50.243538556Z Gathering data for ns/openshift-
insights...
[must-gather-cgbfw] POD 2025-11-27T04:22:51.664116030Z Gathering data for ns/openshift-
monitoring...
.tgz'

[admin@sec-rhel-9 ocp-sec]$ ls -lr | grep must-gather
drwxr-xr-x. 3 root    root          4096 Nov 26 23:24 must-gather.local.6303451730326235937
drwxr-xr-x. 3 root    root          4096 Nov 26 23:18 must-gather.local.324078142430953747
# Create a compressed file from the must-gather directory!
[root@sec-rhel-9 ocp-sec]$ tar cvaf must-gather-`date +"%m-%d-%Y-%H-%M-%S^C-8ed9044f-f02a-4f02-
b28e-d551a40568d7.tar.gz .


#Verify the compressed file
[admin@sec-rhel-9 ocp-sec]$ ls -lr | grep must-gather
drwxr-xr-x. 3 root    root          4096 Nov 26 23:24 must-gather.local.6303451730326235937
drwxr-xr-x. 3 root    root          4096 Nov 26 23:18 must-gather.local.324078142430953747
-rw-r--r--. 1 root    root    3674566498 Nov 26 23:31 must-gather-11-26-2025-23-27-56-8ed9044f-
f02a-4f02-b28e-d551a40568d7.tar.gz
```

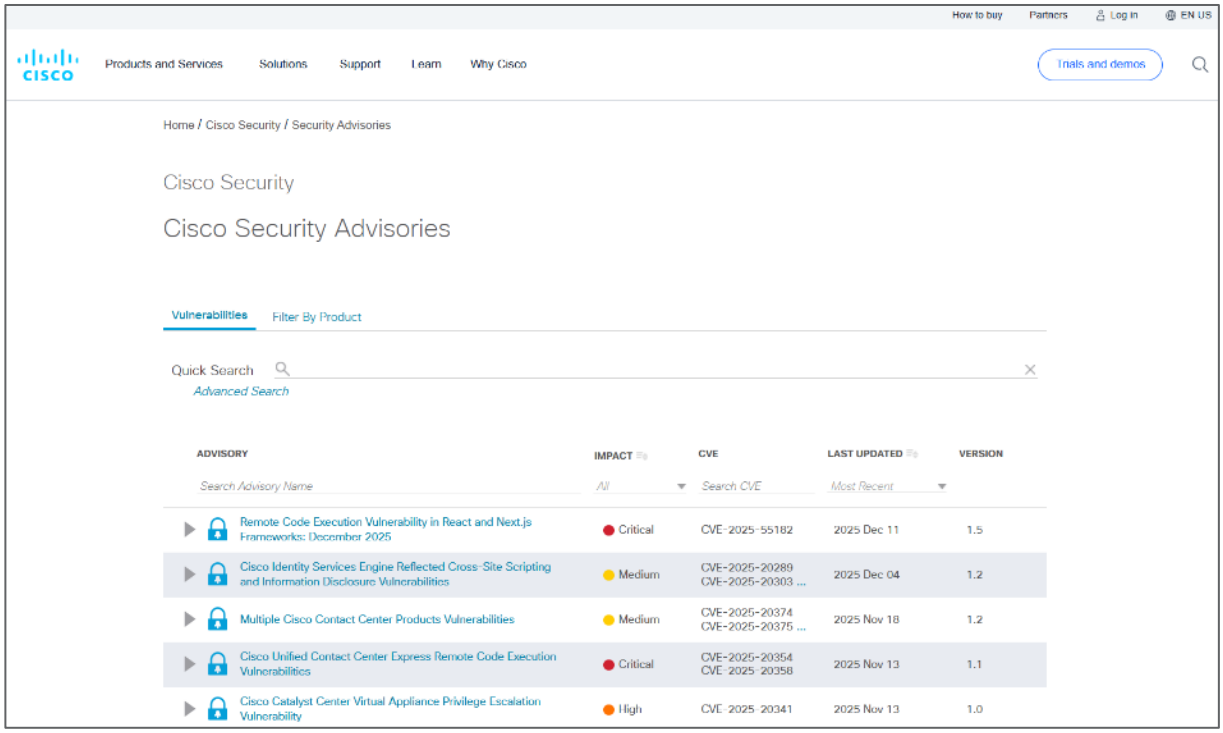
The compressed file can be attached to your support case on the Customer Support page of the Red Hat Customer Portal

Security advisories

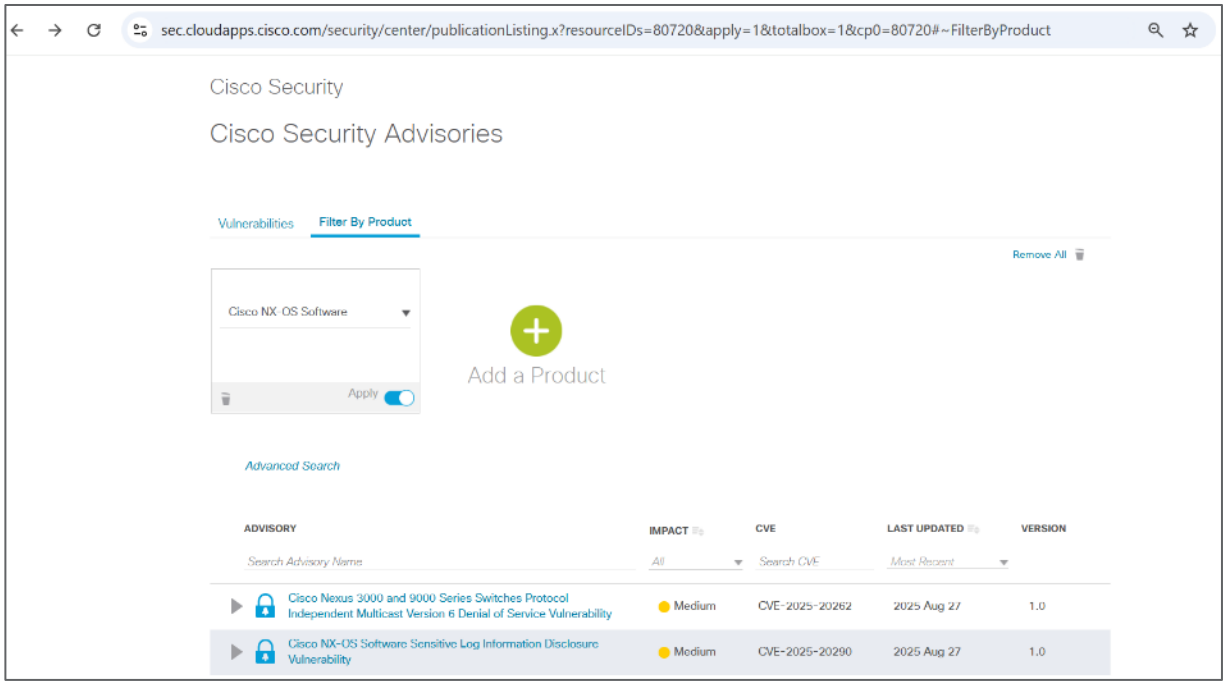
FlexPod converged infrastructure integrates solutions from NetApp, Cisco, and Red Hat. To provide a secure environment for your solutions, you need to follow security best practices for the respective components when deploying the solution. In addition, solution security is not a one-and-done effort. As new vulnerabilities are detected, companies make assessments on their products and provide security advisories to inform their customers on those issues as well as documenting fixes and workarounds. As an on-going security hardening effort, it is important to subscribe to security advisories from NetApp, Cisco, and Red Hat, evaluate those vulnerabilities, and apply remediations as appropriate to keep your FlexPod solution secure.

Cisco security advisories

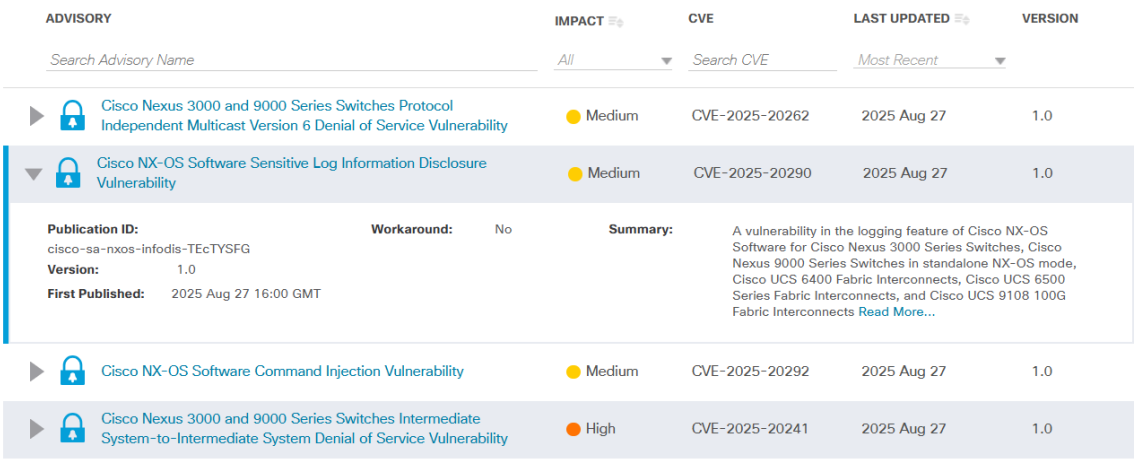
Cisco Security Advisories are available on Cisco security information site, https://sec.cloudapps.cisco.com/security/center/home.x. On the Security Advisories page, you can review a list of vulnerability announcements and remediation instructions published by Cisco.



To find security advisories for a particular Cisco product of interest, e.g., Cisco Unified Computing System (Management Software) and Cisco NX-OS Software, click on the Filter By Product tab, select the products, and set Apply to filter on those products as shown below.

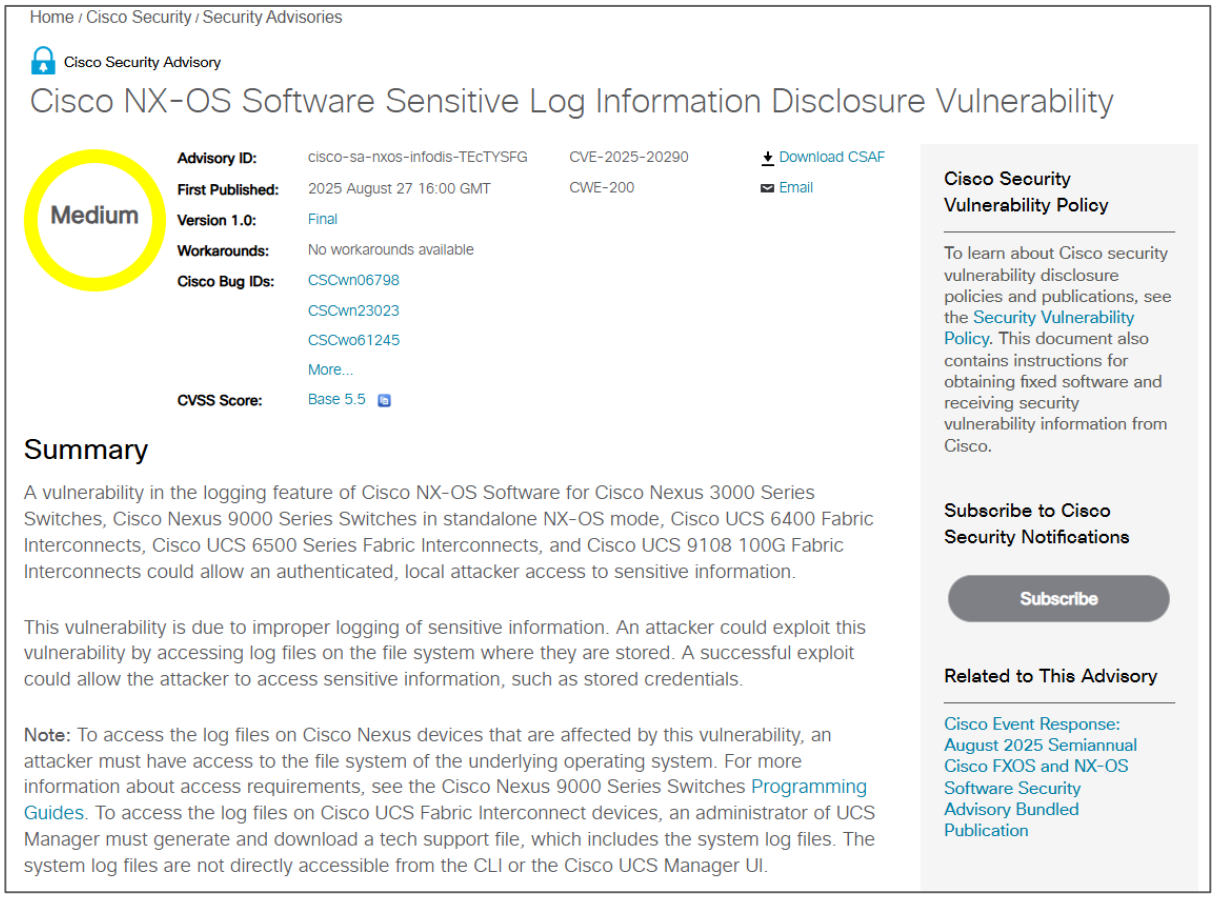


As illustrated in the screenshots below, you can click on the arrow to the left of the Advisory Name to quickly see a summary of the security advisory, whether a workaround is available, or the publishing information. You can click on the Advisory Name to see a full description of the security advisory, affected products, workarounds, fixed software, and find additional resources and information.



ADVISORY	IMPACT	CVE	LAST UPDATED	VERSION
Search Advisory Name	All	Search CVE	Most Recent	
Cisco Nexus 3000 and 9000 Series Switches Protocol Independent Multicast Version 6 Denial of Service Vulnerability	Medium	CVE-2025-20262	2025 Aug 27	1.0
Cisco NX-OS Software Sensitive Log Information Disclosure Vulnerability	Medium	CVE-2025-20290	2025 Aug 27	1.0
Cisco NX-OS Software Command Injection Vulnerability	Medium	CVE-2025-20292	2025 Aug 27	1.0
Cisco Nexus 3000 and 9000 Series Switches Intermediate System-to-Intermediate System Denial of Service Vulnerability	High	CVE-2025-20241	2025 Aug 27	1.0

Publication ID: cisco-sa-nxos-infodis-TEcTYSFG
Version: 1.0
First Published: 2025 Aug 27 16:00 GMT
Workaround: No
Summary: A vulnerability in the logging feature of Cisco NX-OS Software for Cisco Nexus 3000 Series Switches, Cisco Nexus 9000 Series Switches in standalone NX-OS mode, Cisco UCS 6400 Fabric Interconnects, Cisco UCS 6500 Series Fabric Interconnects, and Cisco UCS 9108 100G Fabric Interconnects Read More...



Home / Cisco Security / Security Advisories

Cisco Security Advisory

Cisco NX-OS Software Sensitive Log Information Disclosure Vulnerability

Medium

Advisory ID: cisco-sa-nxos-infodis-TEcTYSFG CVE-2025-20290 Download CSAF
First Published: 2025 August 27 16:00 GMT CWE-200 Email
Version 1.0: Final
Workarounds: No workarounds available
Cisco Bug IDs: CSCwn06798, CSCwn23023, CSCwo61245
More...
CVSS Score: Base 5.5

Summary

A vulnerability in the logging feature of Cisco NX-OS Software for Cisco Nexus 3000 Series Switches, Cisco Nexus 9000 Series Switches in standalone NX-OS mode, Cisco UCS 6400 Fabric Interconnects, Cisco UCS 6500 Series Fabric Interconnects, and Cisco UCS 9108 100G Fabric Interconnects could allow an authenticated, local attacker access to sensitive information.

This vulnerability is due to improper logging of sensitive information. An attacker could exploit this vulnerability by accessing log files on the file system where they are stored. A successful exploit could allow the attacker to access sensitive information, such as stored credentials.

Note: To access the log files on Cisco Nexus devices that are affected by this vulnerability, an attacker must have access to the file system of the underlying operating system. For more information about access requirements, see the Cisco Nexus 9000 Series Switches Programming Guides. To access the log files on Cisco UCS Fabric Interconnect devices, an administrator of UCS Manager must generate and download a tech support file, which includes the system log files. The system log files are not directly accessible from the CLI or the Cisco UCS Manager UI.

Cisco Security Vulnerability Policy

To learn about Cisco security vulnerability disclosure policies and publications, see the Security Vulnerability Policy. This document also contains instructions for obtaining fixed software and receiving security vulnerability information from Cisco.

Subscribe to Cisco Security Notifications

Subscribe

Related to This Advisory

Cisco Event Response: August 2025 Semiannual Cisco FXOS and NX-OS Software Security Advisory Bundled Publication

Under the Cisco Security Center Home page, https://sec.cloudapps.cisco.com/security/center/home.x , you can find additional security related information including Cisco policies and processes, Cisco security solutions, experts' blog, tactical resources, and notification registration for Cisco Security RSS feeds, Cisco Security Blog, and customized email notifications.

Cisco Intersight security advisories

Advisories, vulnerabilities, and incident responses

CERT advisories

Cisco's Computer Emergency Response Team (CERT) advisories are transmitted when new vulnerabilities are identified. Cisco's internal CERT team monitors and alerts product groups to potential issues that might affect their respective components. When these items are identified by CERT or are otherwise indicated by vendor partners (Red Hat, etc.), patches are either developed or acquired from the respective vendors. Cisco has heavily invested to protect customers by creating this team, which constantly monitors threats and builds a centralized solution to remediate these issues and vulnerabilities.

Additional vulnerability testing measures

Cisco also utilizes an internal tool for threat modeling called Threat-builder. This tool is used to explicitly map out application components and services and to identify potential attack surfaces and develop line items for direct evaluation. This information, along with industry tools, is used for vulnerability and exploit testing by Cisco's ASIG (Advanced Security Initiatives Group). ASIG also uses fuzzing and manual testing as part of its suite of tools.

Running vulnerability scans against PVA/CVA

PVA and CVA have an abstracted shell (such as IOS®, HXCLI, etc.). You cannot run a credentialed root scan against this shell. The backing, embedded operating system is currently CentOS, but that will soon change to Alma. You will not be able to enter the development debug shell.

Incident response

The Cisco Product Security Incident Response Team (PSIRT) is responsible for responding to Cisco product security incidents. The Cisco PSIRT is a dedicated, global team that manages the receipt, investigation, and public reporting of information about security vulnerabilities and issues related to Cisco products and services. Cisco defines a security vulnerability as a weakness in the computational logic (for example, code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability. Cisco reserves the right to deviate from this definition based on specific circumstances. The Cisco PSIRT adheres to ISO/IEC 29147:2018, which is a set of guidelines for disclosure of potential vulnerabilities established by the International Organization for Standardization.

The Cisco PSIRT is on call and works 24 hours a day with Cisco customers, independent security researchers, consultants, industry organizations, and other vendors to identify possible security vulnerabilities and issues with Cisco products and networks.

All vulnerabilities disclosed in Cisco Security Advisories are assigned a Common Vulnerability and Exposures (CVE) identifier and a Common Vulnerability Scoring System (CVSS score) to aid in identification. Additionally, all vulnerabilities are classified based on a Security Impact Rating (SIR).

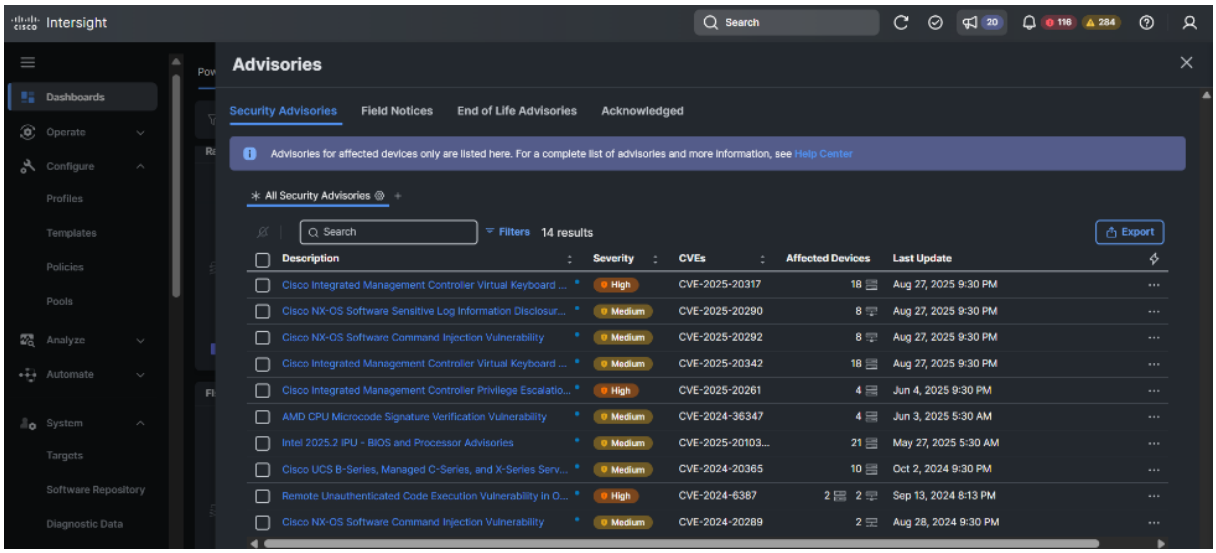
Cisco uses version 3.1 of CVSS as part of its standard process of evaluating reported potential vulnerabilities in Cisco products. The CVSS model uses three distinct measurements or scores that include base, temporal, and environmental calculations. Cisco provides an evaluation of the base vulnerability score and, in some instances, a temporal vulnerability score. End users are encouraged to compute the environmental score based on their network parameters.

In addition, Cisco uses the Security Impact Rating (SIR) to categorize vulnerability severity in a simpler manner. The SIR is based on the CVSS base score, adjusted by PSIRT to account for variables specific to Cisco, and is included in every Cisco Security Advisory.

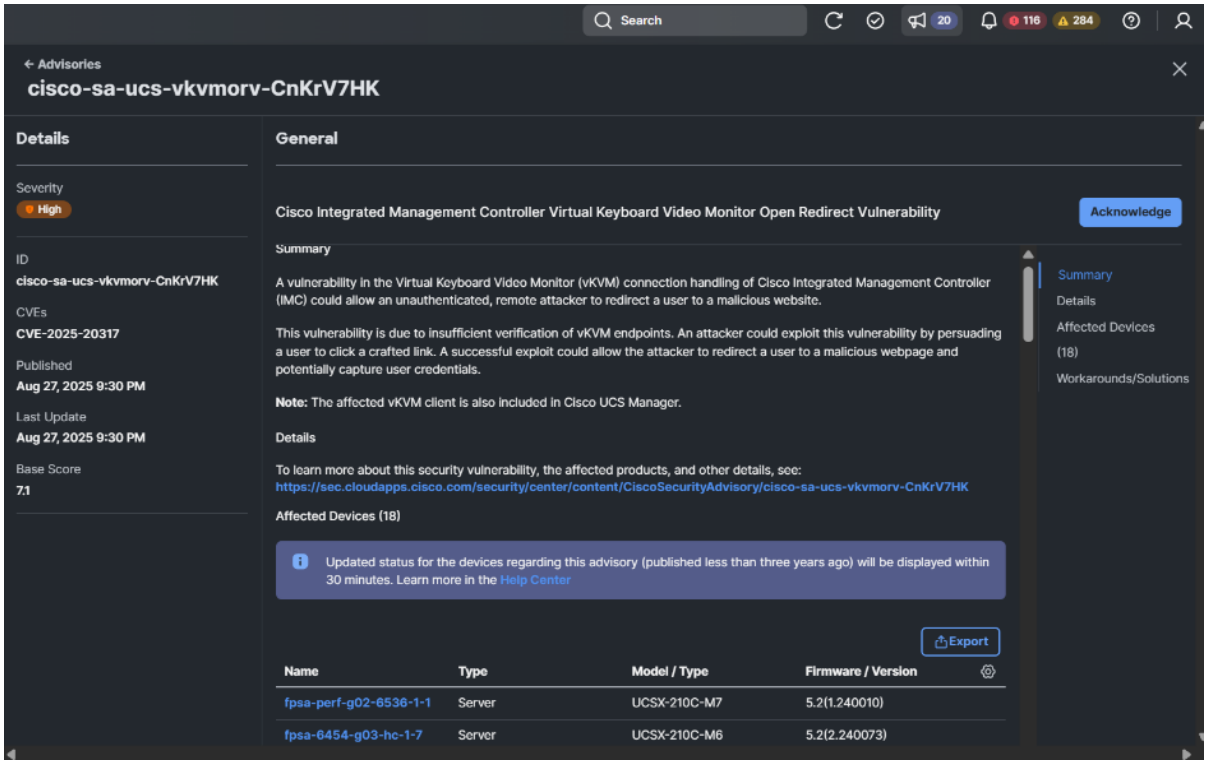
Cisco PSIRT assigns a Common Vulnerabilities and Exposures Identifier (CVE ID) to any vulnerability that is found in Cisco products and that qualifies to receive this identifier. Usually, all vulnerabilities with medium, high, or severe SIRs — that is, a CVSS score of 4.0 or greater — will qualify for a CVE ID.

For the on-going security monitoring, Intersight alerts users about the endpoint devices that are impacted by supported security advisories based on the versions running in the environment. You can click on the Advisories icon (loudspeaker icon) in the top menu bar in the dashboard to display the relevant Cisco Security Advisories.

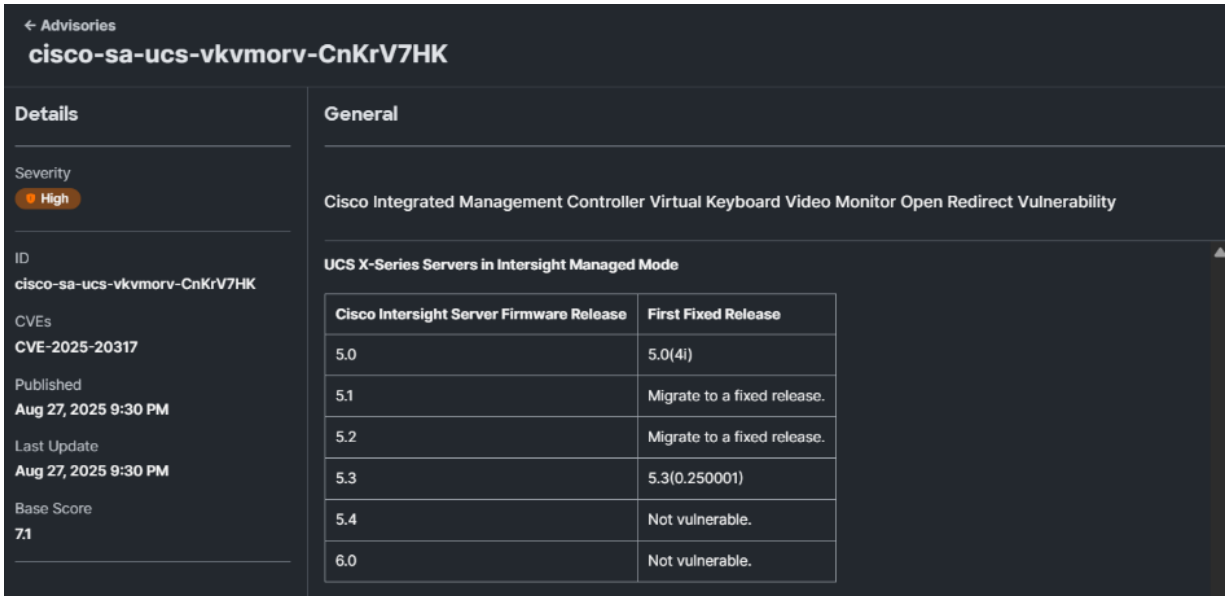
The following is a partial screenshot of Advisories for the devices that have been claimed and discovered in the account. In the security advisories table, it provides description of the advisory, severity level, Common Vulnerabilities and Exposures (CVE) ID numbers, total number of affected devices, and the date and time of the update.



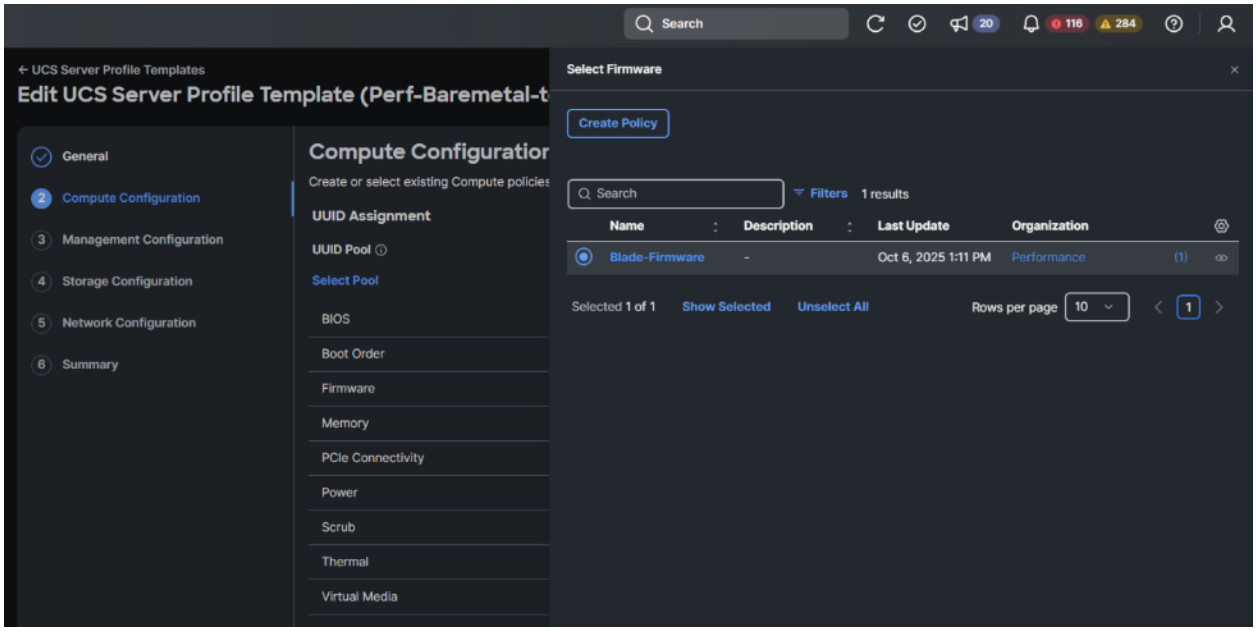
To see the details of a particular advisory, you can simply click on the description link for it, such as the link for the `Cisco Integrated Management Controller Virtual Keyboard Video Monitor Open Redirect Vulnerability.` The advisory page contains a summary of the advisory, links to the details on the Cisco security advisory portal, a list of the affected endpoint devices, workarounds, and the first fixed releases for the various server platform models. In addition, you can also acknowledge the advisory from the page by clicking on the Acknowledge button shown near the upper right-hand corner in the screenshot below.



In the advisory page above, it shows that X-210C M6 and M7 servers running Firmware version 5.2 are affected. Also, if we scroll down, we can see all affected UCS servers. For example, the below figure displays X-Series server firmware affected by this CVE.



To address this issue, we added the Fixed release firmware version to the server profile template and upgraded the blades.

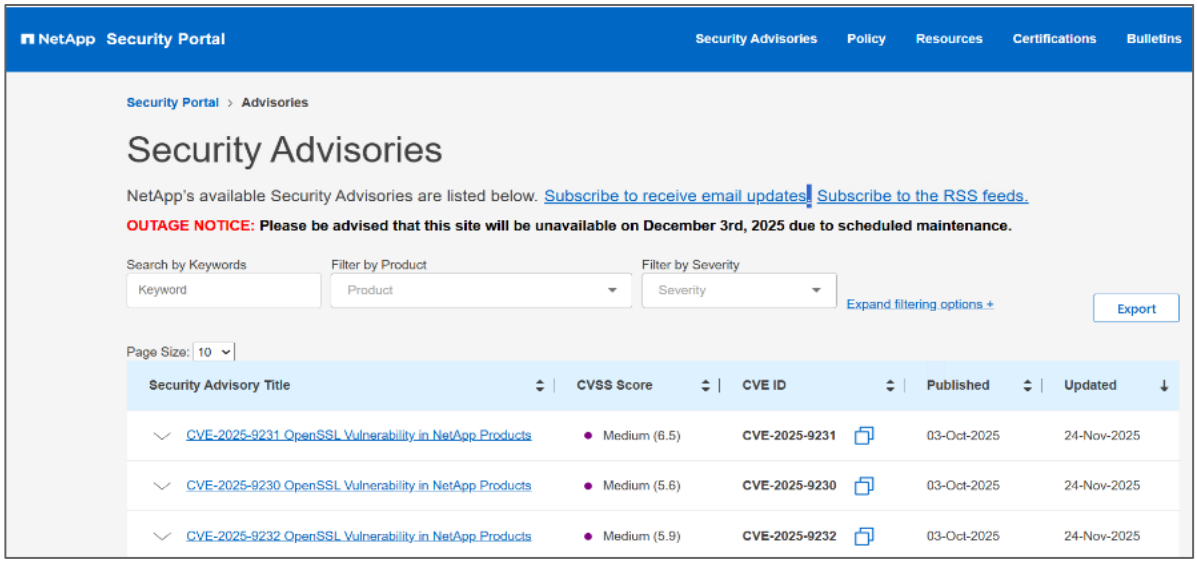


Note: Pease refer to the FlexPod CVDs in the reference section and Intersight Help Center for more information on using Intersight for FlexPod management and additional Intersight features and functionalities.

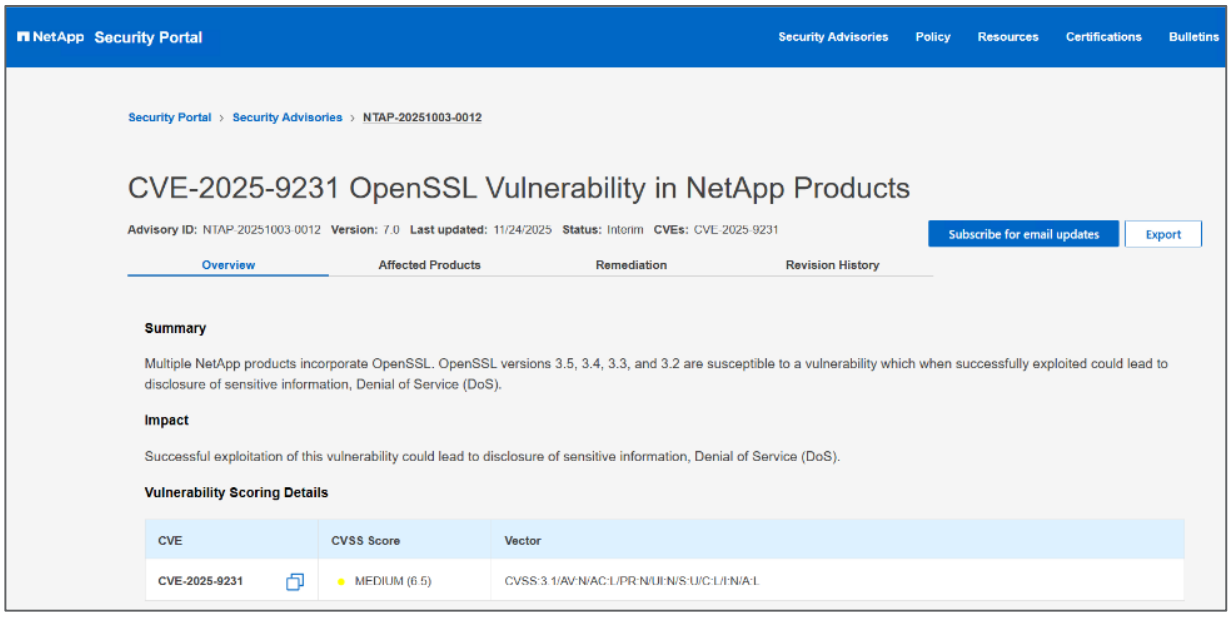
NetApp security advisories

NetApp Security Advisories are available on NetApp security information site, https://security.netapp.com. On the Advisories page, https://security.netapp.com/advisory, you can review a list of vulnerability announcements and remediation instructions published by the NetApp Product Security Incident Response Team (PSIRT). You can click into the vulnerability ID to get detailed information on the

vulnerability and remediation. You can also click on the Subscribe to receive email updates button to sign up for email notifications.



In the following example, the OpenSSL Vulnerability (CVE-2025-9231) in NetApp products is documented. The information includes summary and impact of the vulnerability, the scoring details, the affected products, software versions and fixes, workarounds, and where to obtain the software fixes.

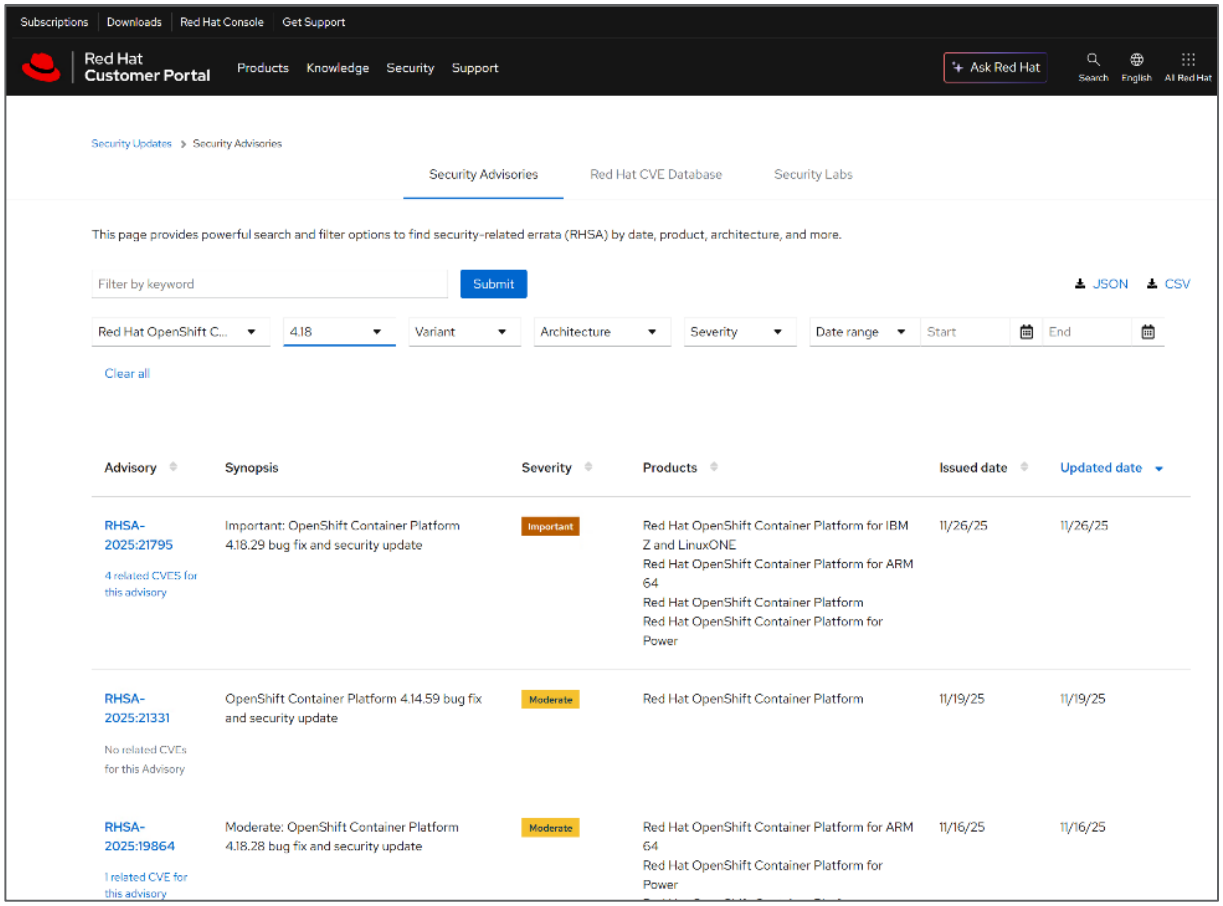


In addition to providing the product security advisories, the NetApp security web site, https://security.netapp.com, is where customers can learn NetApp security policies, certifications, and find resources to help them maintain the confidentiality, integrity, and availability of their data.

Red Hat Security Advisory (RHSA)

Red Hat Security Advisories are available on Red Hat security advisories information site, which documents remediations for the reported Red Hat product security vulnerabilities. You can search for security advisories for a particular Red Hat product and specify a severity level, All / Critical / Important /

Moderate / Low, for your search. You can also sign up to receive the latest security advisories and updates on the web site, https://access.redhat.com/security/security-updates/security-advisories.



As technologies evolve and knowledge about the various attacks and vulnerabilities become public and are documented, it is also important to be up to date on those incidents and security advisories so you can better evaluate the vulnerabilities of your solutions. Keeping the software and firmware updated can help mitigate any vulnerabilities which have been addressed and adopting new security functionalities can help improve the overall security of your solution.

Conclusion

This technical report has outlined a streamlined and actionable security hardening approach for deploying FlexPod with Red Hat OpenShift, covering all foundational components—Cisco UCS, Cisco Nexus, NetApp ONTAP, NetApp Trident, Red Hat OpenShift container platform and OpenShift Virtualization. Together, these technologies form a tightly integrated, defense-in-depth architecture that supports modern, container-driven workloads with strong baseline security.

Across the compute layer, Cisco UCS combined with Cisco Intersight enhances platform security through built-in capabilities such as policy-driven server profiles, firmware compliance, secure boot enforcement, hardware-level integrity validation, and centralized lifecycle governance. Intersight's consistent configuration management and automated drift detection provide an additional layer of assurance, reducing risks associated with misconfigurations and operational variance.

On the data layer, NetApp ONTAP contributes robust, enterprise-grade security features—including data-at-rest encryption, role-based access control, secure multitenancy, immutable snapshots, and

integrated ransomware protection. These capabilities ensure that storage services consumed through NetApp Trident inherit a secure, consistent, and compliant foundation for both containerized and virtualized applications.

OpenShift further strengthens this architecture through built-in cluster security, including strong authentication and authorization, workload isolation, configuration integrity, and continuous monitoring. OpenShift Virtualization extends these protections to VM workloads, enabling unified policy enforcement across both virtual machines and containers.

By applying the hardening recommendations outlined in this report, organizations can deploy a secure, resilient, and operationally consistent FlexPod platform that benefits from the native security strengths of Cisco Intersight, Nexus and NetApp ONTAP—while leveraging the powerful application orchestration and workload protections provided by Red Hat OpenShift.

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

FlexPod

- FlexPod Home Page
  https://www.flexpod.com
- Cisco Validated Design and deployment guides for FlexPod:
  https://www.cisco.com/c/en/us/solutions/design-zone/data-center-design-guides/flexpod-design-guides.html

Cisco UCS

- Cisco Servers - Unified Computing System (UCS)
  https://www.cisco.com/c/en/us/products/servers-unified-computing/index.html
- FIPS 140-2 Level 1 Security Policy for Cisco Secure ACS FIPS Module
  https://csrc.nist.gov/csrc/media/projects/cryptographic-module-validation-program/documents/security-policies/140sp948.pdf
- Cisco Unified Edge
  https://www.cisco.com/site/us/en/products/computing/unified-edge/index.html

Cisco Nexus

- Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 10.2(x)
  https://www.cisco.com/c/en/us/td/docs/dcn/nx-os/nexus9000/102x/configuration/Security/cisco-nexus-9000-nx-os-security-configuration-guide-102x.html

Cisco Intersight

- Intersight help center
  https://intersight.com/help/appliance

NetApp ONTAP

- ONTAP product documentation
  https://docs.netapp.com/us-en/ontap-family/
- NetApp Trident
  https://docs.netapp.com/us-en/trident/index.html
- Data protection and disaster recovery

https://docs.netapp.com/us-en/ontap/data-protection-disaster-recovery/index.html

- NetApp Trident Protect
- https://docs.netapp.com/us-en/trident/trident-protect/learn-about-trident-protect.html
- TR-4569: Security hardening guide for NetApp ONTAP
  https://www.netapp.com/media/10674-tr4569.pdf
- TR-4572: The NetApp solution for ransomware
  https://www.netapp.com/media/7334-tr4572.pdf

Red Hat OpenShift

- OpenShift container platform
  https://docs.redhat.com/en/documentation/openshift_container_platform/4.19
- OpenShift security and compliance
  https://docs.redhat.com/en/documentation/openshift_container_platform/4.19/html/security_and_compliance/index
- OpenShift Virtualization
  https://docs.redhat.com/en/documentation/openshift_container_platform/4.19/html/virtualization/index

Version history

As an option, use the NetApp Table style to create a Version History table. Do not add a table number or caption.

Version	Date	Document version history
Version 1.0	Feb 2026	

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.

NetApp