



Ultra Cloud Core Common Execution Environment - Configuration and Administration Guide

First Published: 2020-11-12

Last Modified: 2026-04-23

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2020-2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE

About this Guide	xi
Conventions Used	xi

CHAPTER 1

Ultra Cloud Core Subscriber Microservices Infrastructure - Overview	1
Ultra Cloud Core Subscriber Microservices Infrastructure Overview	1
SMI on Bare Metal - Overview	2
Subscriber Microservices Infrastructure Architecture	2
SMI Bare Metal - Architecture	4
SMI Bare Metal Deployment Architecture	5
K8s Cluster Manager	6
K8s Resource Management	7
Common Execution Environment	8
Monitoring and Debugging	8
Tracing	8
Ops Center	8
Service Mesh	9
Common Data Layer	10
SMI VM Quantities and Sizing	11
SMI Bare Metal Hardware Requirements	11
Redundancy	12
Security	13
TLS Support for TACACS+	13
Secure TACACS+ Communication with TLS	13
Key benefits	14
Prerequisites for TLS integration	14
How TLS enhances security	15

Enable TLS for TACACS protocol communication 15
 Troubleshooting common issues 16

CHAPTER 2

Common Execution Environment 19

Overview 19
 CEE Installation 21
 Prerequisites 21
 Requirements 21
 Components Used 21
 Installing CEE 21
 CEE Pods 23
 Accessing CEE Ops Center 25
 Upgrading CEE 25
 Upgrading CEE Ops Center 26
 Upgrading CEE Products 27
 Configuring CEE 28
 Configuring Alerts 28
 Requirements 28
 Configuring Alert Rules 28
 Configuring Alerts for ETCD Nodes 30
 Helm Deployment Alert Rule 31
 Viewing Alert Logger 31
 Enabling SNMP Traps 33
 Disabling SNMP Traps 34
 Configuring Bulk Statistics 34
 Retrieving Bulk Statistics 35
 Grafana 38
 Accessing Grafana 39
 Using Dashboards 39
 User Management in Grafana 42
 Configuring Ingress for Prometheus 42
 Provisioning Local Users 42
 User Management 43
 Adding a User 43

Creating Unprivileged Users with SSH Key	44
Deleting a User	45
Modifying the Password	45
Modifying the Password for Other Users	46
Updating the Password Length	48
Group Management	48
Adding a User Group	48
Deleting a User Group	49
Assigning an User to an User Group	49
Unassigning a User from a User Group	50
Log Forwarding	50
Prerequisites	51
Requirements	51
Enabling Log Forwarding	51
Enabling Log Forwarding on StarOS	52
Configuring CEE Ops Center as a Listener	53
Configuring Fluent-D to Support Splunk	53
Configuring Fluent-Bit to Support Splunk	53
Configuring Fluent-Bit to Support Remote Forwarding	54
Configuring Fluent-Bit to Support Remote Listener	54
Configuring Fluent-Bit to Support Grafana Cloud	55
Configuring Fluent-Bit to Support Syslog	56
Configuring Fluent Worker to Drop and Retain Logs	56
Viewing the Logs in Loki	57
Verifying Log Forwarding	57
Troubleshooting	57
Log Rate Limiting	58
Rate Limiting Log Messages	58
Gather TAC	58
Debugging Data in CEE	59
Debugging Data	59
Log Monitoring	62
Cluster Monitoring	63
Configuring the Remote Cluster	64

- Configuring the Cluster Manager to Collect the Metrics from Remote Clusters 64
- Cluster Alerting 65
 - CIMC Alerts Exporter 65
 - Configuring the Remote Cluster 65
 - Configuring the Cluster Manager to Collect the Alerts from Remote Clusters 66
 - Configuring CIMC 66
 - Configuring Email Notification for Alerts 68
- UCS Server Status Alerts 68
 - Feature Description 68
 - Monitoring CIMC Reachability 68
- Node Problem Detector 69
 - Benefits of NPD 70
 - How NPD works 70
 - Enable the Node Problem Detector 70
 - Metrics for node issues 74
 - Best practices for using NPD 75
- Thanos ecosystem for metrix handling 75
 - Architecture 76
 - Push metrics data to an S3-compatible object storage using the Thanos ecosystem 78
 - How it Works 78
 - Configure Thanos components 80
 - Configure Object Storage 81
 - Manage Secret 81
 - Configure Thanos Receive 82
 - Configure Thanos Store Gateway 83
 - Configure Thanos Compact 83
 - Configure Thanos Query 85
 - Configure Thanos Query Frontend 85
 - Configure Thanos Ruler 85
 - Sending alerts by Prometheus to remote Alert Manager 86
 - Configure Remote AlertManager 86
 - Configure Object Storage for Thanos Sidecar 87
 - Configure AlertManager Ingress Exposure 88
- Sending Prometheus Server Metrics to Grafana Cloud 89

Feature Description	89
Remote Write Configuration	89
K8s Certificates Auto-Renewal	91
Certificate Management with Kubeadm	91
How it Works	91
OnDemand LDAP Connectivity Check	92
Feature Summary and Revision History	92
Summary Data	92
Revision History	93
Feature Description	93
How it Works	93

PART I
CEE Config Mode Command Reference 95

CHAPTER 3
Alerts Operation Config Mode Command Reference 97

alerts active	97
alerts add-silence	98
alerts add-silence matchers	99
alerts delete-silence	100
alerts history	100
alerts silence-by-id	100
alerts silences	101

CHAPTER 4
Bulk Statistics Config Mode Command Reference 103

bulk-stats	103
bulk-stats current	104
bulk-stats pod-query	105
bulk-stats query	106
bulk-stats vnf-alias	107

CHAPTER 5
CIMC Config Mode Command Reference 109

cimc	109
cimc cluster	109
cimc cluster default	110

cimc cluster server 110
 node-problem-detector agent 111

CHAPTER 6 Cluster Exec Mode Command Reference 113

cluster 113
 cluster configmaps 113
 cluster configmaps detail 114
 cluster connect 114
 cluster ingresses 115
 cluster ingresses detail 115
 cluster namespaces 115
 cluster nodes 116
 cluster nodes detail 117
 cluster persistent-volume-claims 117
 cluster persistent-volumes 118
 cluster pods 118
 cluster pods delete 119
 cluster pods detail 120
 cluster services 120
 cluster services detail 120

CHAPTER 7 Debug Exec Mode Command Reference 123

tac-debug-pkg create 123
 tac-debug-pkg create cores-filter 125
 tac-debug-pkg create logs-filter 125
 tac-debug-pkg delete 125
 tac-debug-pkg merge 126
 tac-debug-pkg status 126

CHAPTER 8 Grafana Config Mode Command Reference 127

grafana 127
 grafana dashboards 127
 grafana enable-basic-auth 128

CHAPTER 9	Logging Config Mode Command Reference	129
	logging fluent	129
	logging fluent tls	130
	logging fluentd	130
	logging listener	131
	logging loki	132
	logging splunk	132
	logging syslog	133
	logging worker	134

CHAPTER 10	NPD Config Mode Command Reference	137
	node-problem-detector agent	137
	node-problem-detector agent exporters	137
	node-problem-detector agent monitors	138
	node-problem-detector agent exporters k8s	139
	node-problem-detector agent exporters prometheus	139

CHAPTER 11	Prometheus Config Mode Command Reference	141
	prometheus	141
	prometheus federation	142
	prometheus federation exported-query-nodes	142
	prometheus federation remote-cluster-certs	143
	prometheus kvm-metrics defaults	144
	prometheus kvm-metrics monitor-server	144
	prometheus prometheus-operator	145
	prometheus pushgateway	145
	prometheus pushgateway port	146
	prometheus recording-rules group	146
	prometheus recording-rules group rule	146
	prometheus recording-rules group rule label	147
	prometheus server-settings	147

CHAPTER 12	SNMP Config Mode Command Reference	149
-------------------	---	------------

snmp-trapper	149
snmp-trapper source-ip-routes	150
snmp-trapper source-ip-routes source-external-vips	150
snmp-trapper v2c-target	151
snmp-trapper v3-target	151

CHAPTER 13	VES Adapter Config Mode Command Reference	153
	ves-adapter	153
	ves-adapter measurement-group	154
	ves-adapter measurement-group measurement	154



About this Guide



Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. While any existing biased terms are being substituted, exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

This preface describes the *Common Execution Environment* component of the Cisco *Subscriber Microservices Infrastructure (SMI)*, how it is organized and its document conventions.

This guide describes the Common Execution Environment (CEE) and includes infrastructure and interfaces, feature descriptions, specification compliance, session flows, configuration instructions, and CLI commands for monitoring and troubleshooting the system.

- [Conventions Used, on page xi](#)

Conventions Used

The following tables describe the conventions used throughout this documentation.

Notice Type	Description
Information Note	Provides information about important features or instructions.
Caution	Alerts you of potential damage to a program, device, or system.
Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.

Typeface Conventions	Description
Text represented as a <code>screen display</code>	This typeface represents displays that appear on your terminal screen, for example: <code>Login:</code>
Text represented as commands	This typeface represents commands that you enter, for example: show ip access-list This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.
Text represented as a command variable	This typeface represents a variable that is part of a command, for example: show card <i>slot_number</i> <i>slot_number</i> is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the File menu, then click New



CHAPTER 1

Ultra Cloud Core Subscriber Microservices Infrastructure - Overview

- [Ultra Cloud Core Subscriber Microservices Infrastructure Overview](#), on page 1
- [Subscriber Microservices Infrastructure Architecture](#), on page 2
- [Redundancy](#), on page 12
- [Security](#), on page 13

Ultra Cloud Core Subscriber Microservices Infrastructure Overview

The Ultra Cloud Core Subscriber Microservices Infrastructure (SMI) provides a run time environment for deploying and managing Cisco's cloud-native network functions (cNFs), also referred to as applications.

It is built around open source projects like Kubernetes (K8s), Docker, Helm, etcd, confd, and gRPC and provides a common set of services used by deployed cNFs including:

- **Protocol Load Balancing:** These microservices provide the external NF interfaces (HTTP, Diameter, GTP, LDAP, etc.) and load balance requests to the application microservices. They normalize internal communications and allow application evolution independent of the interface evolution. Each protocol type is usually implemented as a separate microservice. gRPC is used for internal communication with the application microservices
- **Database Service:** The database service provides a normalized gRPC interface to the application microservices. The database service can interface to different databases allowing the use of different back-end databases depending on the application requirements while maintaining the same interface.
- **Cisco Service Mesh:** This service provides rule-based control over load balancing decisions across different application containers. Through this service, SMI supports and automates operations such as canary upgrades, new service roll-outs, and in-service upgrades.
- **Telemetry Service:** Telemetry functionality is provided through a common set of microservices which collect real-time statistics, alarms, logs from various deployed application components, and translates and streams them to external functions.
- **Dashboard Service:** The dashboard service works with the telemetry service to provide operational overview data for application containers such as state, utilization, and key performance indicators (KPIs).

Cisco's cNFs are implemented as a set of microservices that make use of the common platform services offered by SMI. Refer to the NF's documentation for additional details.

SMI supports Cisco Smart Licensing, a cloud-based licensing model that simplifies the purchase, deployment, and management of Cisco software assets. For more information, refer to the [Smart Licensing](#) collection page.

SMI on Bare Metal - Overview

The SMI extends the deployment of Virtual Network Functions (VNF) and Cloud-Native Network Functions (CNFs) to bare metal servers (Cisco UCS-C servers) with the current release. Also, the SMI supports vertically integrated deployment on bare metal servers.

The following are some of the significant features deploying SMI on Bare Metal servers:

- Utilizes KubeVirt as the native virtualization layer
- Zero touch deployment for both VNF and CNF based applications
- Automated infrastructure upgrades
- Exposed API for deployment, configuration, and management to enable automation.
- Addresses edge deployment
 - Provides single compute user plane to run at remote sites
- Scales out without any additional overhead
- Ground up API (NETCONF, REST) driven design and architecture
 - All the interfaces are compliant with northbound NFVO (for instance, NSO).
- Simplification and remote management
- Removes shared storage from the architecture
- Single monitoring endpoint for both server and application health



Note The SMI has the ability to run virtual machines for legacy applications (CUPS CP and UPF).

Subscriber Microservices Infrastructure Architecture

The Ultra Cloud Core Subscriber Microservices Infrastructure (SMI) is a layered stack of cloud technologies that enable the rapid deployment of, and seamless life cycle operations for microservices-based applications.

The SMI stack consists of the following:

- **SMI Cluster Manager** — Creates the Kubernetes (K8s) cluster, creates the software repository, and provides ongoing Life Cycle Management (LCM) for the cluster including deployment, upgrades, and expansion.



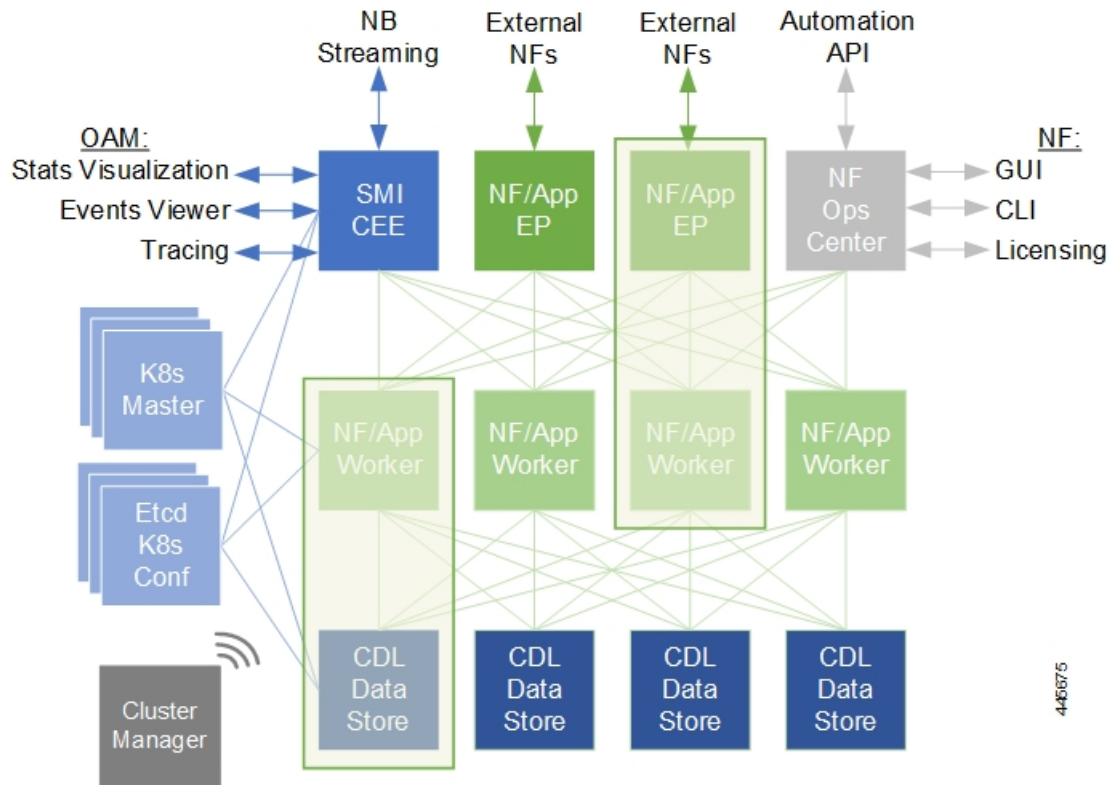
Note The SMI Cluster Manager can install all SMI based applications (including the SMI Cluster Manager) in a Day-0 manner. For Day-1 configurations, you can utilize the deployed application Ops Center.

The SMI Cluster Manager supports the following platforms:

- **VMware** — The Cluster Manager deploys the base images using the vSphere APIs.
- **Bare Metal** — The Cluster Manager configures:
 - UCS-C server based hosts using Cisco Integrated Management Controller (CIMC) APIs.
- **Manual** — The Cluster Manager allows other systems (NSO/ESC) to provision the base image and configure the K8s Cluster.
- **Kubernetes Management** — Includes the K8s control plane and etcd functions which provide LCM for the cNF applications deployed in the cluster as well as provides cluster health monitoring and resources scheduling.
- **Common Execution Environment (CEE)** — Provides common utilities and OAM functionalities for Cisco cNFs and applications, including licensing and entitlement functions, configuration management, telemetry and alarm visualization, logging management, and troubleshooting utilities. Additionally, it provides consistent interaction and experience for all customer touch points and integration points in relation to these tools and deployed applications.
- **Common Data Layer (CDL)** — Provides a high performance, low latency, stateful data store, designed specifically for 5G and subscriber applications. This next generation data store offers HA in local or geo-redundant deployments.
- **Service Mesh** — Provides sophisticated message routing between application containers, enabling managed interconnectivity, additional security, and the ability to deploy new code and new configurations in low risk manner.
- **NF/Application Worker nodes** — The containers that comprise an NF application pod.
- **NF/Application Endpoints (EPs)** – The NF's/application's interfaces to other entities on the network.
- **Application Programming Interfaces (APIs)** — SMI provides various APIs for deployment, configuration, and management automation.
- **Ops Center** — The SMI run time environment, as well as each Cisco cloud native application, includes an innovative management interface called Ops Center. This Netconf/Restconf interface, based on Yang schema, enables all configurations for SMI and Cisco cloud native applications, to be automated or managed directly through a CLI.

Figure 1 depicts how these components interconnect to comprise a microservice-based NF/application.

Figure 1: SMI Components



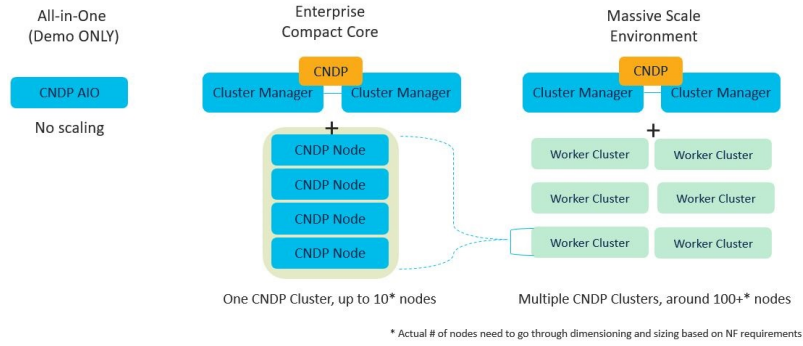
SMI Bare Metal - Architecture

The SMI enables the deployment of Cluster Manager on Bare Metal servers. The following are some of the salient features of SMI Bare Metal architecture:

- Enables all the application containers to run on the bare metal servers with enough resource isolation
- Provides a migration path for SMI on VM to SMI on bare metal
- Automated bring up at the Data Center
- Hardware agnostic architecture

The following figure depicts the high-level SMI Bare Metal Architecture:

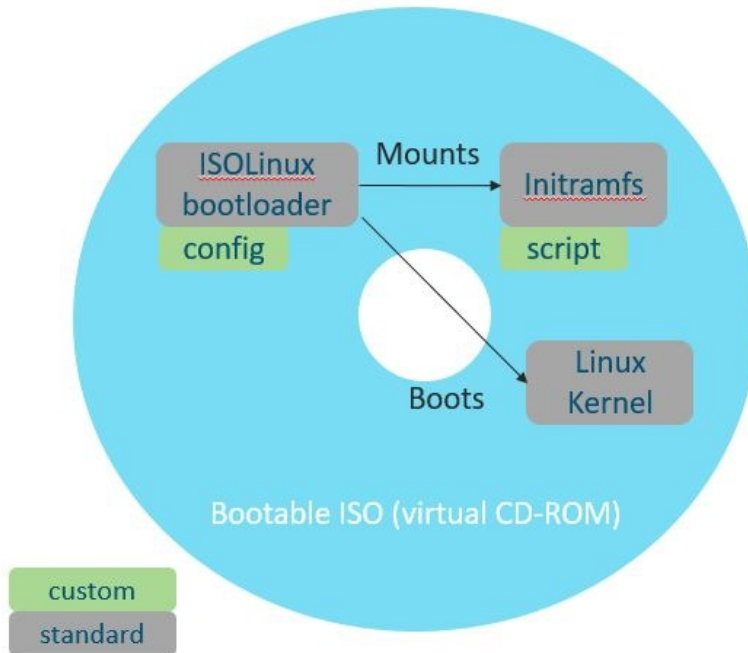
Figure 2: SMI Bare Metal High-level Architecture



With the help of a Bootable ISO, the SMI Cluster Manager boots the Linux Kernel from the base image. This allows compatibility with most of the standard hardware platforms. A customized script downloads and writes the HD image using the Initial RAM File System. Also, the Bootable ISOs smaller size - 23 Mega Bytes (MB) - reduces latency.

The following figure depicts the operations of the Bootable ISO:

Figure 3: Bootable ISO



SMI Bare Metal Deployment Architecture

The SMI Bare Metal deployment architecture comprises of a two node Management Cluster. The two node cluster comprises of a SMI CEE (for monitoring) and SMI Cluster Manager running on it. Also, the two node cluster is responsible for:

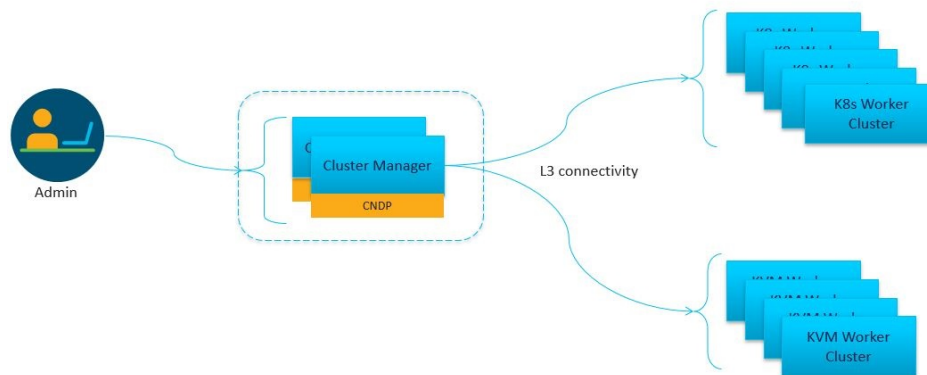
- installing and upgrading the BIOS, host OS, Kubernetes, KVM.

- installing and upgrading Kubernetes based NFs.
- adding the day-0 configuration to installed NFs.
- installing StarOS NFs (UPF).
- installing VPC-DI and VPC-SI.
- monitoring and alerting.

The SMI Cluster Manager provisions and manages the Life Cycle Management (LCM) of each worker node for both the K8s and Kernel based Virtual Machine (KVM) infrastructure.

The following figure depicts the high-level architecture of SMI Bare Metal Deployment architecture:

Figure 4: SMI Bare Metal Deployment Architecture



K8s Cluster Manager

SMI operational components and microservices are deployed on VMs. (Refer to *SMI VM Quantities and Sizing* for details.)

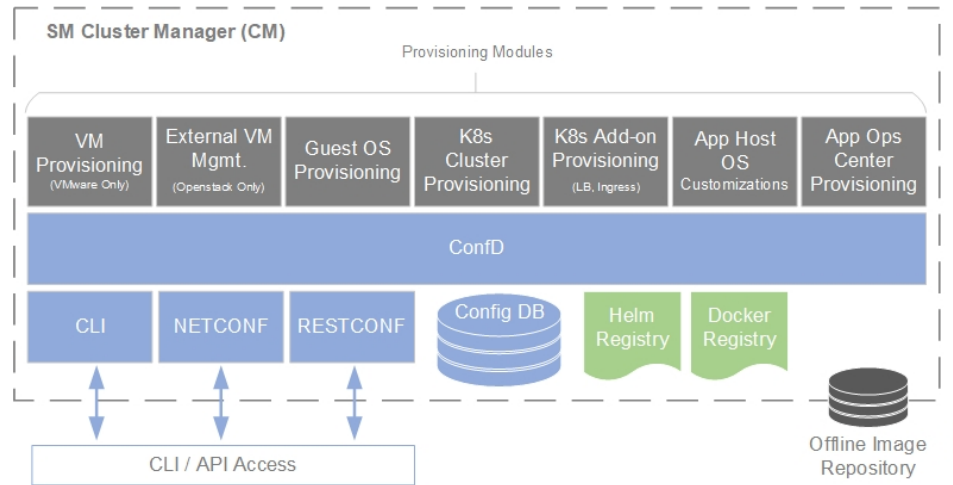
The SMI Cluster Manager (CM) is also deployed as a VM and is used to bootstrap the deployment of other components and applications.

The CM works with the Virtual Infrastructure Manager (VIM) to instantiate the required VMs. In VMware environments, the CM instantiates the virtual machines (VMs) required for the cluster. In OpenStack environments, the CM makes an API call to an orchestrator or Virtual Network Function Manager (VNFM) to instantiate VMs.

The VMs are deployed with a guest OS that is provided with SMI. Once instantiated, the CM provisions the OS, and deploys or provisions the SMI microservices (for example, K8s control plane, K8s etcd, and so on.).

Once all VMs and K8s components are built, the CM can deploy 5G application Ops Centers, which enable NETCONF/RESTCONF interfaces for application configuration and management. All of these actions are API driven and all can be automated and orchestrated.

Figure 5: SMI Cluster Manager Functionality



Scheduling rules such as affinity and anti-affinity help guide K8s for proper node placement, as well as adding node taint and tolerances. Because K8s uses a declarative method of deployment, operators simply need to update the desired number of services and K8s manages scheduling and maintains the correct number of services, even during failure scenarios.

K8s Resource Management

SMI leverages the native resource reservation controls in K8s.

K8s provides a framework to intelligently place pods on the correct server, VM, and/or node, and assign the appropriate system resources, including:

- Service taints, tolerances, affinity, and anti-affinity rules
 - Provides rules for pod placement across available hardware
 - Prevents resource "hotspots" by separating pods with similar resource profiles
 - Provides high availability (HA) by ensuring secondary instances through pod separation
- CPU reservation
 - Allows applications to specify CPUs/CPU requirements (similar to CPU pinning)
 - Prevents negative impacts from context switching, or noisy/grabby neighbors
- Pod quality of service (QoS) definition (e.g. the quality and range of resources available to the Pods)
 - Guaranteed (resource requests = resource limits)
 - Burstable (resource requests > resource limits)
 - Best effort (no resource requests nor limits)

DSCP is implemented at the network level to manage the quality of service and ensure critical traffic is prioritized.

Common Execution Environment

SMI's Common Execution Environment (CEE) provides OAM capabilities for deployed NFs.

The CEE captures information (key metrics) from the NFs in a centralized way for engineers to debug and troubleshoot the overall solution.

There is only one CEE available per K8s cluster, which provides the common set of tools for all deployed NFs. CEE life cycle is independent of NF and it comes equipped with a dedicated Ops Center, which provides the user interface (CLI) and APIs for managing the monitoring tools.

Monitoring and Debugging

The SMI platform provides multiple layers of health checking:

- **Deployment health checks** — These confirm that the infrastructure meets the application requirements.
NOTE: Some deployment health checks (input/output operations per second (IOPS) validation and network throughput) may impact performance and should only be executed during the deployment phase.
- **Run time health checks** — These checks are constantly running in the background to verify that logging and tracing are set to the lowest levels, and to check error rates and alarms.
- **Pod health checks** — These confirm that the pod is alive and service availability. If the pod fails the health check, it is killed and re-scheduled onto another available node.
- **Performance checks** — The checks provide such data as transactions per second (TPS), number of records (sessions), CPU and memory utilization, errors, etc.

Statistics are available for viewing through Grafana, as well as for streaming using Prometheus. They are also available in bulkstat format. The granularity of statistics can be as small as 1 second. Statistics are stored for up to 3 days using Thanos to compress and compact the data.

Logging utilizes journald and rsyslog to collect and distribute logs northbound to a fully featured logging platform. SMI also includes logging utilities to collect snapshots for troubleshooting and uploading to Cisco TAC support centers. Logging verbosity and detail levels are set via API, and can be set to Critical, Error, Warning, Informational, or Debug.

Application and platform events can be forwarded northbound using Prometheus plugins such as VES and/or SNMP.

Tracing

Cisco's cloud native based applications are designed to tag messages in a method compatible with OpenTracing project guidelines.

SMI provides tooling and centralized storage for continuous tracings of cNFs even as they may span across multiple nodes.

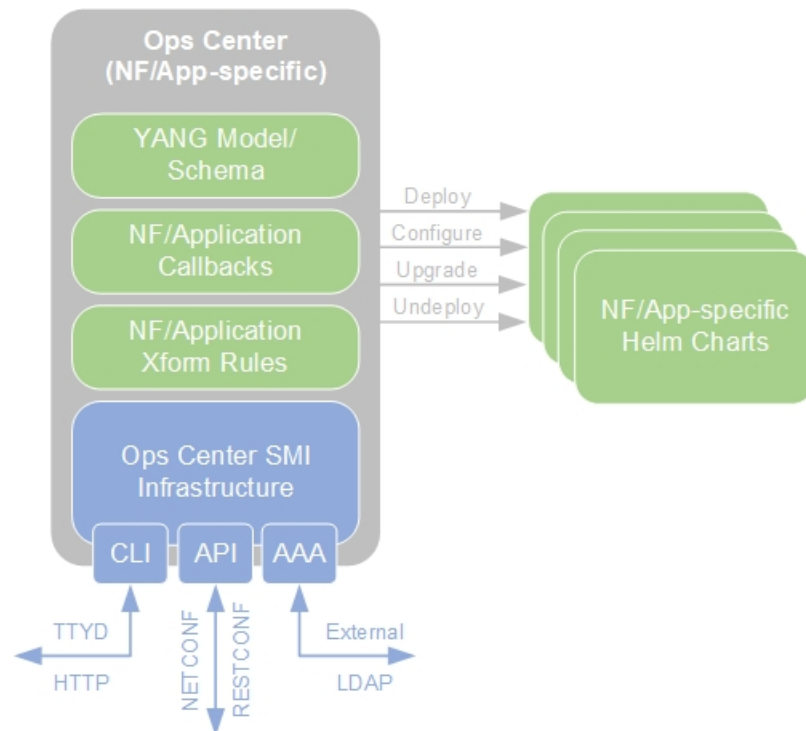
This tracing shows all "message spans" from platform ingress to platform egress as well as how long each unit of work takes.

Ops Center

Cisco's cNFs consist of Helm charts (applications and charts) and Docker files (images).

To simplify and establish consistent operations across the various charts and images that comprise each NF, each NF is designed with an Ops Center. Ops Centers provide a common, stable CLI/API for operators to deploy and manage the NF in a holistic way.

Figure 6: NF/Application Ops Center



SMI provides the following functionality in relation to NF Ops Centers:

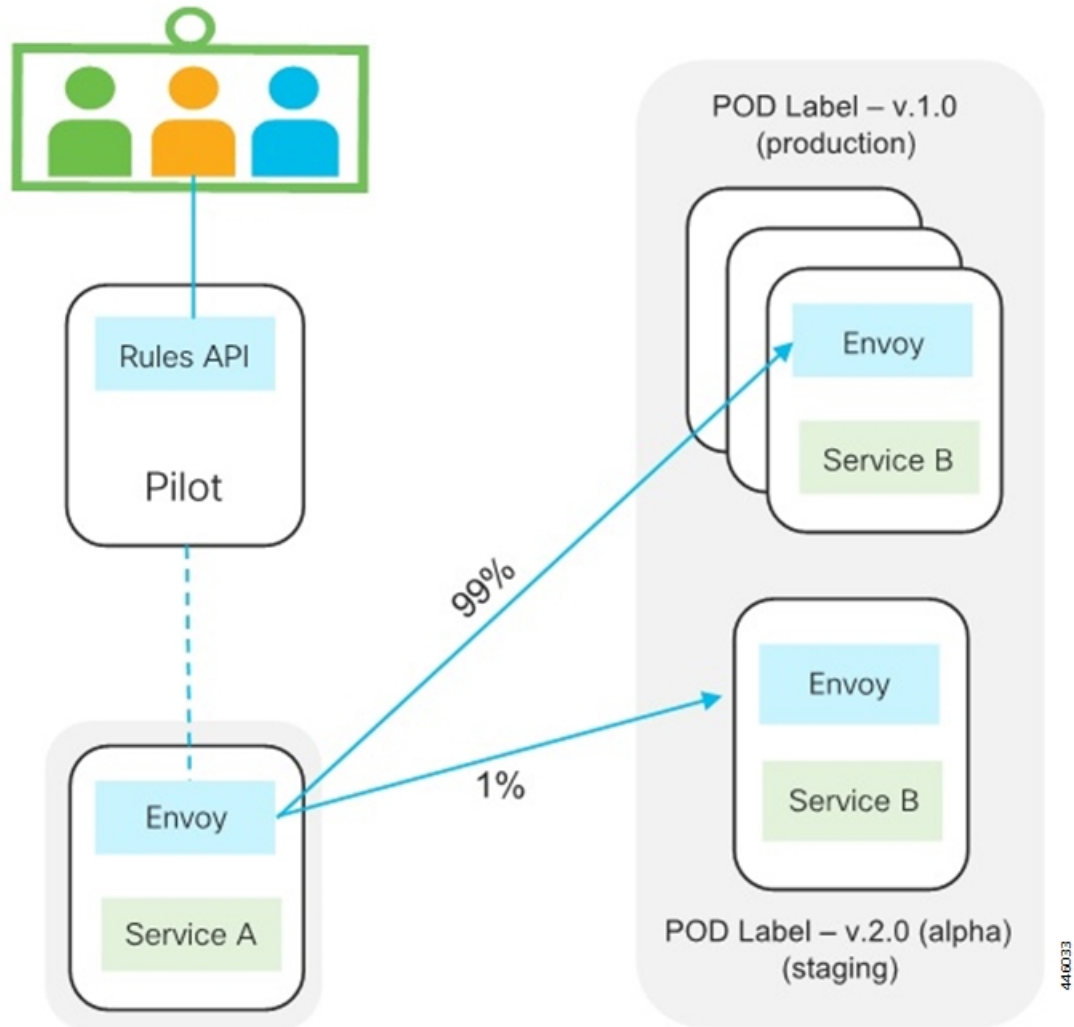
- Common NETCONF, RESTCONF, and CLI interfaces, which allows for integration network orchestrators such as Cisco's Network Services Orchestrator (NSO) without need for a custom network element driver (NED)
- A YANG model for the application
- Audit logging and configuration validation
- Lightweight Directory Access Protocol (LDAP) interface directory information services — for example, Active Directory (AD) — to ensure all applications use a common set of user accounts
- Cisco Smart Licensing integration
- Callbacks into the application to execute operational commands
- NETCONF Access Control (NACM) security model

Service Mesh

The Service Mesh enabled through SMI connects and manages messages between all pods and services in the cluster. Using this service mesh, traffic is steered within the cluster to finely control which NF components are part of the traffic flow.

Granular controls such as traffic percentage, or application-based traffic characteristics — for example, access point name (APN), subscription permanent identifier (SUPI), or other layer 7 attribute value pairs (AVPs) — are used to control traffic within the cluster. This control enables selective and precise upgrades, such as "canary upgrades". This limits risk and impact when deploying changes in-service and in production. It also affords the ability to selectively drain or decommission NFs.

Figure 7: SMI Service Mesh



Besides traffic management applications, the service mesh aids in tracking the flow of traffic between services and nodes, providing security to prevent unauthorized service access and isolating rogue services.

Common Data Layer

The Common Data Layer (CDL) component enabled through SMI provides the clean separation of stateful (also known as backing services) and stateless services (e.g. application services).

CDL provides services for efficiently managing stateful subscriber and identity information across all deployed Cisco NFs. The CDL is an in-memory database designed specifically for high performance carrier grade requirements and subscriber data. Separating stateful services in this way allows for the stateless application services to be autonomous, lightweight, upgradable, recoverable, and rapidly scalable.

Stateful services must address the availability, consistency, and portability of state. These typically require replication across one or more containers while maintaining state consistency.

As such, CDL redundancy is achieved by local and remote replication of session data. In addition, a background process scans the data store for inconsistencies, stale data, and corruption, and corrects them both locally and remotely.

SMI VM Quantities and Sizing

Table 1 and Table 2 provide SMI VM quantity and sizing recommendations.

NOTE: Individual NFs are deployed as K8s workers through SMI. They each have their own VM recommendations. Refer to the NF documentation for details.

Table 1: SMI VM Function and Quantities

VM Purpose	Redundancy	# VMs
SMI Cluster Manager	NA	1
K8s Control Plane	3	3
K8s EtcD	3	3
OAM	3	3

Table 2: SMI VM Sizing Recommendations

VM Function	vCPUs	NUMA per VM (Single/Double)	CPU Pinned	RAM (GB)	Boot Volume Size (GB)	Data Volume Size (GB)
SMI Cluster Manager	2	1	Yes	16	40	100
K8s Control Plane	2	1	Yes	16	100	20
K8s EtcD (CDL)	2	1	Yes	16	100	20
OAM	12	1	Yes	112	100	200

SMI Bare Metal Hardware Requirements

The following table lists the minimum Bare Metal requirements for deploying SMI Cluster Manager.

Table 3: SMI Bare Metal Hardware Requirements (UCS-C Series)

Item	Requirements
Server	Cisco UCS M5/M6/M7/M8

Item	Requirements
Networking	<ul style="list-style-type: none"> • Cisco Catalyst 3850 Switches • Cisco Nexus 9000 Series Switches
Storage	SSD Note For Disk drives, you must use SSDs to improve the read/write access speed.

Cisco UCS M7 Server Deployment Limitations

The Cisco UCS M7 server deployment has these limitations:

- Only two interfaces are created by CNDP on the deployed virtual machine (VM).
- The two ports are mapped to NUMA0, even when the cluster configuration uses the "full" flavor.
- When deploying a KVM-based SI-CP (using the "upf" keyword), there is no validation to check whether the physical port is up or down. As a result, virtual functions (VFs) attached to the VM may belong to ports that are down on the host. All four ports need to be connected, or at least the port whose VF is attached to the VM must be connected.
- The cluster configuration does not provide an option to specify which physical ports (PFs) or VFs are attached to the VM.



Note The Bare Metal requirements listed in the table for deploying SMI Cluster Manager are for reference only. For specific requirements, contact your Cisco account representative.

Redundancy

SMI enables redundancy at multiple levels:

- **Network** — This is provided by the infrastructure and hardware with dual networking paths, dual NICs, and interface bonding. It is also provided by the SMI platform through the use of virtual IP addresses (VIPs), load balancers (LBs), and through the use of Cisco's Service Mesh.
- **K8s cluster** — The K8s cluster leverages a multiple control plane design.

In order to avoid potential conflicts if two components modify the same objects, K8s implements a leader/follower pattern for the controller manager and the scheduler. Each group elects one leader, then the other group members assume follower roles. At any point in time, only the leader is active, and the followers are passive.

K8s configuration (etcd) also uses a consensus-based leader/follower election process. Storage includes Storage Area Network/Network Area Storage (SAN/NAS) for persistence during server or VM failure. On leader failure, a new election takes place to determine a new leader. When the old leader recovers, it comes back as follower. Nothing happens on follower failure.

- **OAM services** — OAM services are deployed in large VMs on two or more nodes. Storage includes SAN/NAS for persistence during VM failure. Services are designed to reserve 50%+ capacity per server in order to allow K8s to reschedule services to next available OAM nodes without impact during a failure.
- **NF applications** — Cisco's stateless applications support N+1 redundancy and rely on K8s to monitor and reschedule when necessary. Application components are distributed across servers for HA purposes.

Security

SMI provides several secure methods for accessing, managing, and configuring the system, all based on APIs, including the Ops Center CLI, and NETCONF/RESTCONF interfaces.

Monitoring interfaces such as Grafana also integrate security and authentication using LDAP Systems Security Services Daemon (SSSD) and Secure Architecture for the Networked Enterprise (SANE).

Access and any configuration changes using the provided CLI and/or API are securely logged.

TLS Support for TACACS+

Table 4: Feature history

Feature name	Release	Description
TLS support for TACACS+	2025.04.1	<p>This feature introduces robust Transport Layer Security (TLS) support for TACACS+ communications, enabling secure, encrypted transport of all TACACS+ traffic between clients and servers.</p> <p>It leverages industry-standard TLS 1.3 (with optional TLS 1.2 for interoperability) and supports mutual TLS (mTLS) for enhanced peer authentication.</p>

Secure TACACS+ Communication with TLS

TLS Support for TACACS+ is a security feature that allows you to establish robust, encrypted communication channels between your device and TACACS+ servers. This enhancement fundamentally strengthens your network's authentication and authorization infrastructure by integrating industry-standard Transport Layer Security (TLS) protocols. As a result, all interactions are protected against modern cyber threats through advanced encryption and authentication mechanisms, ensuring the confidentiality and integrity of your network administration activities.

Key benefits

- **Uncompromised data confidentiality:** All TACACS+ traffic, including sensitive authentication credentials and authorization requests, is encrypted using strong TLS 1.3 protocols. This prevents unauthorized access and eavesdropping, safeguarding critical operational data.
- **Assured data integrity:** TLS guarantees that data exchanged between your device and the TACACS+ server remains unaltered during transit. Any attempt at tampering is immediately detected, maintaining the reliability of your authentication and authorization decisions.
- **Enhanced peer authentication:** Move beyond less secure pre-shared keys (PSKs).
- **Operational resilience through mTLS:** With Mutual TLS (mTLS), both ends of the connection authenticate each other using digital certificates. This significantly reduces the risk of impersonation and ensures that only trusted entities can participate in the TACACS+ exchange.
- **Simplified and automated certificate management:** Critical digital certificates are securely stored and managed within the platform's secret store. This facilitates automatic certificate rotation and refresh without requiring service restarts, ensuring continuous security without operational disruption.
- **Adherence to industry standards:** This feature implementation aligns with RFC 8907 (The TACACS+ Protocol) and the IETF Internet-Draft "TACACS+ over TLS 1.3", guaranteeing interoperability and compliance with leading security practices.

Prerequisites for TLS integration

For ISE server:

- Device Admin Service must be enabled.
- TLS 1.3 must be allowed (TLS 1.2 permitted only if absolutely necessary).
- TACACS over TLS listener enabled (default port: 6049).



Note Review network firewalls or ACLs to allow traffic on port 6049.

- A system certificate with TACACS usage enabled.
- Root and intermediate CAs must be imported and trusted with "Trust for Client Authentication and Syslog" enabled.



Note Ensure all certificates and CAs are current and not expired.

- Server certificate should include Subject Alternative Names (SANs) with acceptable SAN types as ipAddress (IPv4/IPv6), dNSName, directoryName.

For TACACS client:

- Use IPv4 or IPv6 literals; DNS resolution is not supported.
- Ensure outbound reachability to the TACACS server's TLS port (default: 6049).

- mTLS is enforced; both client and server must authenticate.
- Must possess a valid client certificate and key.
- The certificate chain must terminate at a CA trusted by TACACS client.
- Client certificate must include SANs that satisfy the client's policy.
- Access to secret store for CA trust and, if needed, client certificates.
- RBAC (Role-Based Access Control) must permit reading of referenced secrets.
- Rotation of secrets should automatically trigger refresh without restarts.

How TLS enhances security

When TLS is activated for a TACACS+ server, your device initiates a secure TLS session before any TACACS+ data is exchanged. Within this encrypted tunnel, TACACS+ messages are transmitted in their native format, relying entirely on the robust cryptographic protection provided by TLS. This design eliminates the need for legacy per-packet obfuscation, streamlining the protocol while leveraging the proven security framework of TLS.

Enable TLS for TACACS protocol communication

To enable and configure TLS for your TACACS+ servers, you will interact with the system's configuration interface. The following parameters are configured on a per-server basis, allowing granular control over your security settings.

Follow these steps to enable TLS for TACACS servers:

Procedure

Step 1 Connect to your cluster deployer and navigate to the specific Ops-center security configuration using the following command:

```
tacacs-security serverserver_name
```

Example:

```
tacacs-security server1
```

Step 2 Enable TLS encryption for the specific TACACS+ server.

- Use the command **tls-enabled true** to enable TLS for your specific TACACS+ server. This option is false by default.

When **tls-enabled** is set to **true**, the port will default to 6049 if not explicitly specified. When **tls-enabled** is set to **true**, the port automatically defaults to 6049 (the standard port for TACACS+ over TLS) if not explicitly set. When **tls-enabled** is **false**, the legacy port 49 is used.

To disable, set **tls-enabled** to **false** for the specific TACACS+ server using the command **tacacs-security server <server_name> tls-enabled false** and run the system's apply command again. The remote-ca-secret and local-cert-secret values will no longer be active for TLS.

Sample configuration for mTLS

- b. Specify the remote-ca-secret (CA certificate secret name) and local-cert-secret (client certificate secret name) for secure communication. These parameters are crucial for establishing a secure TLS connection.

remote-ca-secret tacacs-ca

local-cert-secret tacacs-local-certs

Step 3 Save and commit the configuration.

Step 4 (Optional) After configuration, you can verify the TLS connection by attempting to authenticate through the configured TACACS+ server and checking system logs for successful TLS handshakes or connection details.

Sample configuration for mTLS

This example demonstrates how to configure a TACACS+ server (server 1) to use TLS with mutual authentication.

```
// 1. Define the client certificate and private key in the secret store
secrets tls tacacs-local-certs
  private-key
"$8$RUZDw8Pvj60IKmw96QzoHesO9/wG6BerKJNcaGFxfIYUnhqo6Py98VWFFQdcmn2H4DMai2L..." // Encrypted
  private key content
  certificate "-----BEGIN CERTIFICATE-----..." // Client certificate content
exit

// 2. Define the CA certificate(s) to trust the TACACS+ server
secrets ca-cert tacacs-ca
  certificate "-----BEGIN CERTIFICATE-----..." // CA certificate content
exit

// 3. Configure the TACACS+ server to use TLS and mTLS
tacacs-security service system
tacacs-security server 1
  address 10.71.232.71 // IP address of the TACACS+ server
  port 6049 // Explicitly set or defaults when tls-enabled is true
  tls-enabled true // Enable TLS for this server
  remote-ca-secret tacacs-ca // Reference to the trusted CA for server authentication
  local-cert-secret tacacs-local-certs // Reference to the client certificate for mTLS
```

Troubleshooting common issues

This table provides guidance for diagnosing and resolving common issues encountered when configuring or operating TACACS+ with TLS.

Issue/Error	Cause	Resolution
Certificate verify failed	The TACACS+ server's certificate could not be validated. This might be due to an untrusted CA, an expired certificate, an incorrect hostname/IP in the certificate's Subject Alternative Name (SAN), or a corrupted certificate chain.	Verify that the remote-ca-secret points to the correct and trusted CA certificate(s). Ensure the server's certificate is valid and not expired, and that its SAN matches the configured server address.

Issue/Error	Cause	Resolution
mTLS handshake alert / unknown CA	The TACACS+ server did not accept your device's client certificate during mutual authentication. This typically means the local-cert-secret is missing, incorrect, or the server does not trust the CA that issued your device's certificate.	Confirm that local-cert-secret is correctly configured and points to a valid client certificate and key. Ensure the TACACS+ server's trust store includes the CA that issued your device's client certificate.
Connection refused/timeout	The device could not establish a network connection to the TACACS+ server.	Verify the TACACS+ server's IP address and port (6049 for TLS) are correct. Check network connectivity, firewall rules, and routing to ensure the device can reach the server on the specified port.
Authentication works in plaintext but fails with TLS	The TACACS+ server might not be correctly configured to accept "TACACS over TLS" connections, or there's a mismatch in TLS versions or security policies.	Confirm that the TACACS+ server is configured to listen for TACACS+ over TLS on port 6049 and accepts cleartext TACACS+ bodies within the TLS session. Verify that TLS 1.3 (or 1.2 if configured) is supported and enabled on both ends, and check for any SAN validation policies on the server.



CHAPTER 2

Common Execution Environment

- [Overview, on page 19](#)
- [CEE Installation, on page 21](#)
- [CEE Pods, on page 23](#)
- [Accessing CEE Ops Center, on page 25](#)
- [Upgrading CEE, on page 25](#)
- [Configuring CEE, on page 28](#)
- [Configuring Alerts, on page 28](#)
- [Configuring Bulk Statistics, on page 34](#)
- [Retrieving Bulk Statistics, on page 35](#)
- [Grafana, on page 38](#)
- [Provisioning Local Users, on page 42](#)
- [Log Forwarding, on page 50](#)
- [Gather TAC, on page 58](#)
- [Log Monitoring, on page 62](#)
- [Cluster Monitoring, on page 63](#)
- [Cluster Alerting, on page 65](#)
- [UCS Server Status Alerts, on page 68](#)
- [Node Problem Detector, on page 69](#)
- [Thanos ecosystem for metrix handling, on page 75](#)
- [Sending Prometheus Server Metrics to Grafana Cloud, on page 89](#)
- [K8s Certificates Auto-Renewal, on page 91](#)
- [OnDemand LDAP Connectivity Check, on page 92](#)

Overview

The Common Execution Environment (CEE) is a software solution developed for monitoring mobile and cable applications that are deployed on the Subscriber Microservices Infrastructure (SMI). The CEE captures information (key metrics) from the applications in a centralized way for engineers to debug and troubleshoot.

The CEE is the common set of tools that are installed for all the applications. It comes equipped with a dedicated Ops Center, which provides the user interface (Command Line Interface) and APIs for managing the monitoring tools. There is only one CEE available for each cluster.

The CEE includes the following components:

- **CEE Ops Center** - The CEE Ops Center allows users to configure and install the CEE. The CEE Ops Center contains the following components:
 - **Metrics Collection** - It includes functions such as reporting from *Prometheus*, alerting and Bulk statistics and so on.
 - **Metrics Visualization** - The metrics are displayed to the end users through a *Grafana* dashboard. The dashboard displays the key metrics such as CPU usage, memory, and disk input and output (I/O) utilization of each application deployed on the SMI. Use cases include:
 - Import custom Grafana dashboard from a GIT repository.

For more information, refer [Grafana](#) section.

- **Bulk Statistics** - Configures application specific statistics, which are collected through the Gather TAC feature. The Bulk Statistics are automatically generated based on the user requirements at repeated intervals. Use cases include:
 - Generate query for current PDU per 4G session.
 - Generate query for current PDU per 4G IPv6 session.
 - View bulk statistics.

For more information, refer to the [Configuring Bulk Statistics](#) section.

- **Metrics Global Query** - *Thanos* - a set of software components for metric system - provides the ability to perform global queries across multiple clusters. Use cases include:
 - In cable environment with multiple Kubernetes clusters, where instances of Prometheus collect metrics specific to cluster, a global Prometheus instance (set up as a part of an application Ops Center) is used as focal point to gather data and respond to queries for metrics from all Prometheus pods.

For more information, refer *Cluster Monitoring* section.

- **Alerting** - Enables you to monitor applications, containers or nodes by setting up alert rules. The CEE uses the *Prometheus Alert Manager* for generating alerts. Use cases include:
 - Monitor the success rate of SMF session creation by configuring Prometheus alert rule to report if session creation is less than threshold.
 - Configure Prometheus alert rule to report if pod has restarted.
 - Alerts addon: If *snmp-trapper* is configured, alert is also sent as SNMP Trap to the receiving agent.
 - View active alerts.
 - View alerts history.

For more information, refer to the [Configuring Alerts](#) section.

- **Log Monitoring** - The *Kubetail* utility in the CEE Ops Center allows end users to monitor the logs of an application in real time.
- **Log Forwarding** - The Log Forwarding function collects and forwards all the logs to any of the third-party applications present in the customer infrastructure. Use cases include:

- Configure log forwarding to an external Splunk server.
- Configure log forwarding to an external Fluent-D or Fluent-Bit instance, where logs can be streamed to supporting application such as ElasticSearch.

For more information, refer [Log Forwarding](#) section.

- **Gather TAC** - The Gather TAC function is used for creating log files at specified intervals of time. The logs are collected based on the pods that are deployed on the Kubernetes cluster. Use cases include:
 - When a Network Function (NF) exhibits some issues, the log collection can be configured to include data and statistics for the system and pods in a specific namespace within the last few hours.

For more information, refer [Gather TAC](#) section.

CEE Installation

This section describes the procedures involved in installing the CEE using the Ops Center.

Prerequisites

The prerequisites for installing the CEE are:

1. Installing the SMI Cluster Manager.
2. Storing the CEE and associated product tarballs in the local repository.
3. Applying the necessary cluster configuration for bringing the Kubernetes Cluster on the target nodes.

Requirements

All the versions of CEE.

Components Used

The following components are used for installing the CEE:

1. The SMI Cluster Manager.
2. The SMI CEE.

Installing CEE

You can install the CEE using the SMI Cluster Manager CLI. To install CEE, use the following configurations:

1. Login to the SMI Cluster Manager CLI (using the ingress URL) and enter the configuration mode.

```
https://cli.smi-cluster-manager.<IP_address>.<customer_specific_domain_name>
```

- Use the following configuration to install the CEE in offline mode.

```
configure
  software cnf software_name
    url HTTP_HTTPS_File_URL
    user username
    password password
    sha256 sha256_hash
  exit
```



Note For offline installation, you must download the CNF software package from the repository.

Use the following configuration to install the CEE in online mode.

```
configure
  repository repo_url
  username username
  password password
  sha256 sha256_hash
  exit
```

- Link the CEE into the desired cluster in the **ops-centers**.

```
configure
  clusters cluster_name ops-center app_name instance_name
  repository repo_url
  username username
  password password

  secrets docker-registry <docker_secret_registry>
  docker-server docker_server_name
  docker-username docker_username
  docker-password docker_password
  docker-email <email_id@domain.com>
  namespace <namespace>
  exit
  sync-default-repository true
  netconf-ip <ipv4address>
  netconf-port <portnumber>
  ssh-ip <ipv4address>
  ssh-port <portnumber>
  ingress-hostname <ipv4address>.nip.io
  initial-boot-parameters use-volume-claims true
  initial-boot-parameters first-boot-password password
  initial-boot-parameters auto-deploy true
  initial-boot-parameters single-node true
  initial-boot-parameters image-pull-secrets <secret_name>
  exit
exit
```

4. Run the cluster synchronization to deploy the CEE Ops Center and wait for the synchronization to complete.

```
clusters cluster_name actions sync run
```

5. Verify the cluster synchronization through cluster sync status or log commands.

```
clusters cluster_name actions sync status
```

```
clusters cluster_name actions sync logs
```

NOTES:

- *customer_specific_domain_name* - Specifies the customer's domain name.
- **software cnf** *software_name* - Specifies the Cisco's Cloud Native software. *software_name* is the name of the Cloud Native software.
 - **url** *HTTP_HTTPS_File_URL* - Specifies the repository URL.
 - **user** *username* - Specifies the username for HTTP/HTTPS authentication.
 - **password** *password* - Specifies the password used for downloading the software package.
 - **sha256** *sha256_hash* - Specifies the SHA256 hash of the software download.
- **repository** *repo_url* - Specifies the CNF repository.
- **clusters** *cluster_name* **actions sync run** - Synchronizes the committed changes to the cluster.
- **clusters** *cluster_name* **actions sync status** - Displays the status of the cluster synchronization.
- **clusters** *cluster_name* **actions sync logs** - Displays the logs generated during the cluster synchronization process.

CEE Pods

A pod is a process that runs on your Kubernetes cluster. Pod encapsulates a granular unit that is known as a container. A pod contains one or multiple containers.

Kubernetes deploys one or multiple pods on a single node which can be a physical or virtual machine. Each pod has a discrete identity with an internal IP address and Port space. However, the containers within a pod can share the storage and network resources.

The following table lists the Common Execution Environment (CEE) pod names and their descriptions.

Table 5: CEE Pods

Pod Name	Description
alert-logger	Stores and maintains historical alerts that are received from the Alert manager. These alerts are available to user through the CEE ops-center.
alert-router	Provides routing support for the alert manager to pass alerts to its receivers.

Pod Name	Description
alertmanager	Process alerts from Prometheus and route them to its receivers through alert-router. It also provides a list of active alerts available to the user in CEE ops-center and Grafana.
blackbox-exporter	Enables Prometheus blackbox probing of endpoints over HTTP, TCP, and ICMP.
bulk-stats	Provides summary of statistics that are collected by Prometheus service and create periodic snapshots of statistics on each node in the form of CSV files.
cee-product-documentation	CEE Product documentation page provides an overview of CEE functions.
cimc-alerts-exporter	Scrapes and exports CIMC alerts to be viewable in Grafana.
core-retriever	Assists in retrieving the core dumps.
documentation	Contains the documentation (metrics and usage).
fluentbit	Collects the logs from journalD or systemd and forwards to the external applications like splunk or another remote fluent instance.
grafana-dashboard-metrics	Assists in collating Grafana metrics on the dashboard.
fluentbit-listener	Collects the logs from remote fluent instances and forward these logs to external collectors like Splunk.
grafana	Provides visualization tool and host-level dashboards to examine metrics and alerts.
grafana-dashboard-metrics	Supports the internal file server for Grafana dashboards.
kube-state-metrics	Assists in generating metrics about the state of Kubernetes objects: node status, node capacity (CPU and memory), and so on.
loki	Provides support to visualize the logs that are provided by the locally installed fluentBit pods.
logs-retriever	Assists in retrieving Kernel, Kubelet, and Container level logs through output to the JournalD driver.
logs-forwarder	Support pods logs forwarding to external server through Fluent-bit.
metrics-proxy-group	Create tunnels to enable Prometheus to scrape KPIs from the node-exporters on KVM nodes.
node-exporter	Exports the node metrics to Prometheus and to be viewable on the Grafana dashboard in Host details and summary dashboards.
ops-center-cee-ops-center	Supports user management, authentication, configuration, and show commands for CEE features, which run on pods inside the cluster.

Pod Name	Description
path-provisioner	Provisions the local storage volume along with pv-provisioner.
pgpool	Manage the Postgres resource pool for connection, replication, load balance, and so on. <i>Pgpool</i> is a middleware that works between <i>PostgreSQL</i> servers and a <i>PostgreSQL</i> database.
postgres	Supports SQL database with redundancy to store alerts and Grafana dashboards.
prometheus-hi-res	Enables monitoring and alerting for the Kubernetes cluster, both local and remote. It scrapes alerts, metrics, kubernetes resources exported by pods and nodes information.
prometheus-rules	Contains the default alerting rules and recording rules for Prometheus.
prometheus-scrapeconfigs-synch	Synchronizes the Prometheus scrape configuration.
pv-manager	Monitors the state of nodes and manages persistent volume and associated pods.
pv-provisioner	Enables the application pods to automatically provision the persistent volumes.
restart-kubelet	Monitors the pod ready status and resets the kubelet if the state is in not-ready even though pod is ready.
show-tac-manager	Supports the Tac-Debug feature to collect coredump, logs, metrics, statistics, and ops-center configuration. It also maintains and provides HTTPS access to files storage in the internal Apache server.
smart-agent-cee-global-ops-center	Manages and enforces the Cisco Smart licensing feature per agreement. The the CEE ops-center provides the configuration facility.
thanos-query-hi-res	Runs the Thanos application to support the Prometheus query, data storage, and remote cluster monitoring.

Accessing CEE Ops Center

You can access the CEE Ops Center CLI through the ingress URL. For example:

```
https://cli.cee-global-ops-center.<ip_address>.<customer_specific_domain_name>
```

Upgrading CEE

This section describes the procedure involved in upgrading the CEE Ops Center and CEE products.

Upgrading CEE Ops Center

To upgrade the CEE Ops Center, use the following configurations:

1. Use the following configuration to modify the CEE Ops Center to point it to the new tarball.

```
configure
  cluster cluster_name
    ops-centers app_name instance_name
    repository repo_url
    username username
    password password
    initial-boot-parameters auto-deploy true
  exit
commit
```

2. Run the cluster synchronization to upgrade the CEE Ops Center.



Note Ensure that you enable auto deploy for the CEE products that are being updated.

```
clusters cluster_name actions sync run
```

3. Verify whether the helm charts have been updated through the CEE Ops Center.

```
show helm charts
```

A sample output is shown below:

CHART	INSTANCE	NAMESPACE	STATUS	VERSION	REVISION	RELEASE
cee-ops-center	cee-global-ops-center		deployed	2023.02.1.d249	1	
0.7.0-2023-02-1-0513-230331051211-dec612f	cee-global					
cnat-monitoring	cee-global-cnat-monitoring		deployed	2023.02.1.d249	1	
0.7.0-2023-02-1-0031-230331183330-58ec41c	cee-global					
product-documentation	cee-global-product-documentation		deployed	2023.02.1.d249	1	
0.8.0-2023-02-1-0131-230321085503-2699cb5	cee-global					
pv-manager	cee-global-pv-manager		deployed	2023.02.1.d249	1	
0.3.0-2023-02-1-0029-230320155437-e484272	cee-global					
smi-autoheal	cee-global-smi-autoheal		deployed	2023.02.1.d249	1	
0.2.0-2023-02-1-0030-230330084451-99684bf	cee-global					
smi-show-tac	cee-global-smi-show-tac		deployed	2023.02.1.d249	1	
0.4.0-2023-02-1-0189-230331050005-81130f1	cee-global					
storage-provisioner	cee-global-storage-provisioner		deployed	2023.02.1.d249	1	
0.3.0-2023-02-1-0120-230320160505-1597fdb	cee-global					
telegraf-monitoring	cee-global-telegraf-monitoring		deployed	2023.02.1.d249	1	
0.1.0-2023-02-1-0048-230330084426-9b02da0	cee-global					

4. Verify the status of the system.

```
show system status
```

A sample output is shown below:

```
system status deployed true
system status percent-ready 91.3
```

NOTES:

- **cluster** *cluster_name* - Specifies the name of the cluster. For example, *aio*.
- **ops-centers** *app_name instance_name* - Specifies the installation of the Ops Center. *app_name* is the name of the application. For example, *cee*. The *instance_name* is the name of the instance. For example, *global*.
- **username** *username* - Specifies the username used for logging in to the repository.
- **password** *password* - Specifies the password used for logging into the repository.
- **repository** *repo_url* - Specifies the product chart repository URL.
- **initial-boot-parameters auto-deploy true** – Deploys the product chart automatically.
- **commit** - Commits the configuration changes.
- **show helm status** - Displays the status of the system.
- **clusters cluster_name actions sync run** - Synchronizes the committed changes to the cluster.

Upgrading CEE Products

To upgrade the CEE products, use the following configurations:

1. Access the CEE Ops Center through the ingress URL.

```
https://cli.cee-global-ops-center.<ipv4_address>.<customer_specific_domain_name>
```

NOTES:

- *customer_specific_domain_name* - Specifies the name of the domain specific to the customer.

2. Use the following configuration to update the CEE products chart URL.

configure

```
helm default-repository repo_name
helm repository repo_name
url cee_product_chart_url
username username
password password
exit
commit
```

NOTES:

- *customer_specific_domain_name* - Specifies the name of the domain specific to the customer.
- **helm default-repository** *repo_name* - Specifies the default helm repository name.
- **helm repository** *repo_name* - Specifies the name of the helm repository to update.
- **url** *cee_product_chart_url* - Specifies the product chart URL. For example, *http://charts.<ipv4address>.<domain_name>/cee-2019-09-13/*
- **username** *username* - Specifies the user name.

- **password** *password* - Specifies the password.
- **commit** - Commits the configuration changes.

Configuring CEE

The subsequent sections provide more information about the CEE configuration procedures.

Configuring Alerts

When an anomaly is detected, the system generates a notification called an alert. Based on the statistics pegged by the system, alerts are fired. You can configure an expression to fire an alert when the expression becomes true.

The CEE uses the *Prometheus Alert Manager* for alerting operations. The CEE YANG model - either through CLI or API - allows users to view the active alerts, silenced alerts and alert history. A predefined set of alerting rules are added whenever the application is installed or updated. Also, the applications can call the alert API directly to add or clear alerts. The *Prometheus Alert Manager* API (v2) is the standard API used.

The *Prometheus Alerts Manager* includes the following options:

- **Defining Alert Rules** – This option defines the Alert Manager on what to alert. You can define the alerts using the *Prometheus Query Language (PromQL)*.
- **Defining Alert Routing** – This option defines the *Prometheus Alert Manager* on what to do with the received alerts. At present, the SNMP Trapper is supported as the outbound alerting. Also, the CEE provides an Alert Logger for storing the generated alerts.

Requirements

For configuring alerts:

1. The CEE product must be installed and running.
2. The CEE Ops Center must be accessible.

Configuring Alert Rules

Use the following configuration to configure the alert rules.

```

configure
  alerts rules group alert_group_name
  interval-seconds seconds
  rule rule_name
    expression promql_expression
    duration duration
    severity severity_level
    type alert_type

```

```

annotation annotation_name
value annotation_value
exit

```

exit

NOTES:

- **alerts rules** – Specifies the Prometheus alerting rules.
- **interval-seconds** *seconds* – Specifies the evaluation interval of the rule group in seconds.
- **group** *alert_group_name* – Specifies the Prometheus alerting rule group. One alert group can have multiple list of rules. *alert_group_name* is the name of the alert group. The alert-group-name must be a string in the range of 0 through 64 characters.
- **rule** *rule_name* – Specifies the alerting rule definition. *rule_name* is the name of the rule.
- **expression** *promql_expression* – Specifies the PromQL alerting rule expression. *promql_expression* is the alert rule query expressed in PromQL syntax. The *promql_expression* must be a string.
- **duration** *duration* – Specifies the duration of a true condition before it is considered true. *duration* is the time interval before the alert is fired.
- **severity** *severity_level* – Specifies the relative level of urgency for the operator's attention. *severity_level* is the severity level of the alert. The severity levels are: critical, major, minor and warning.
- **type** *alert_type* – Specifies the type of the alert. *alert_type* is the user-defined alert types. For example, Communications Alarm, Environmental Alarm, Equipment Alarm, Indeterminate Integrity Violation, Operational Violation, Physical Violation, Processing Error Alarm, Quality of Service Alarm, Security Service, Mechanism Violation, or Time Domain Violation.
- **annotation** *annotation_name* – Specifies the annotation to attach to the alerts. *annotation_name* is the name of the annotation.
- **value** *annotation_value* – Specifies the annotation value. *annotation_value* is the value of the annotation.

The following example monitors the success rate of SMF session creation by configuring Prometheus alert rule to report if session creation is less than threshold.

Example:

```

cee# configure terminal
  alerts rules group SMFProcStatus
    interval-seconds 300
    rule PDNSessCreate
    expression
"sum(increase(smf_service_stats{app_name=\"SMF\",procedure_type=\"pdn_sess_create\",status=\"success\"}[5m]))
/
sum(increase(smf_service_stats{app_name=\"SMF\",procedure_type=\"pdn_sess_create\",status=\"attempted\"}[5m]))
< 0.95"
    severity    major
    type        "Communications Alarm"
    annotation  summary
    value      "This alert is fired when the success percentage of pdn_sess_create procedure is
lesser threshold"
    exit

```

In the following example, a alert is sent as SNMP Trap to receiving agent when a snmp-trapper is configured.

Example:

```
cee# configure terminal
  snmp-trapper enable true v2c-target 172.16.181.41 community public port 161
  exit
```

The following example configures an alert, which is fired when the percentage of UDM responses is less than the specified threshold limit.

Example:

```
cee# configure terminal
  alerts rules group SMFUDMchk_incr
  interval-seconds 300
  rule SMFUDMchk_incr
  expression "sum(increase(smf_restep_http_msg_total{nf_type=\"udm\",
message_direction=\"outbound\", response_status=~\"2..\"}[3m])) /
sum(increase(smf_restep_http_msg_total{nf_type=\"udm\", message_direction=\"outbound\"}[3m]))
< 0.95"
  severity major
  type "Communications Alarm"
  annotation summary
  value "This alert is fired when the percentage of UDM responses is less than threshold"
  exit
exit
exit
```

You can view the configured alert using the **show running-config alerts** command.

Example:

The following example displays the alerts configured in the running configuration:

```
cee# show running-config alerts
  interval-seconds 300
  rule SMFUDMchk_incr
  expression "sum(increase(smf_restep_http_msg_total{nf_type=\"udm\",
message_direction=\"outbound\", response_status=~\"2..\"}[3m])) /
sum(increase(smf_restep_http_msg_total{nf_type=\"udm\", message_direction=\"outbound\"}[3m]))
< 0.95"
  severity major
  type "Communications Alarm"
  annotation summary
  value "This alert is fired when the percentage of UDM responses is less than
threshold"
  exit
  exit
exit
```

Configuring Alerts for ETCD Nodes

The ETCD runs on separate nodes in a multi-node environment as opposed to a container within Kubernetes environment. A [Node-Exporter](#) runs on each of the ETCD nodes to obtain host level metrics. Also, the CEE Prometheus starts scraping the metrics automatically after deployment.

You can create alerting rules based on the ETCD Node-Exporter metrics. To configure alerting rules based on ETCD Node-Exporter metrics, use the *ETCD Node IP* as the instance label instead of the Pod name in the expression.



Important The Node-Exporter on ETCD is not running as a Kubernetes Pod.

The following examples configure alerting rules based on ETCD Node-Exporter metrics.

Example:

The following expression configures alerts based on the availability of host memory (less than 30%):

```
((node_memory_MemAvailable_bytes{{{instance="<ETCD-Node-IP>:9100"}}} /
node_memory_MemTotal_bytes{{{instance="<ETCD-Node-IP>:9100"}}}) < 30
```

The following expression configures alerts based on the average CPU usage for five minutes. (greater than 70%):

```
sum(avg without
(cpu)(irate(node_cpu_seconds_total{instance="<ETCD-Node-IP>:9100",mode!="idle"}[5m]))) *
100 > 70
```

Helm Deployment Alert Rule

The CEE Ops Center comes equipped with a built-in alert rule - *helm_deploy_failure* - to indicate the failure status of helm chart deployment. This alert rule comes by default as a Prometheus alerting rule during CEE deployment.

The following is an alert rule definition for *helm_deploy_failure* alert in Prometheus:

```
- alert: helm-deploy-failure
  annotations:
    type: Processing Error Alarm
    description: 'Helm chart {{{labels.chart}}}/{{{labels.namespace}}} deployment failed'
    summary: 'Helm chart failed to deploy for 5 minutes'
  expr: |
    helm_chart_deploy_success < 1
  labels:
    severity: critical
  for: 5m
```

The following example shows an alert generated when helm chart deployment fails.

```
alerts active helm-deploy-failure 3edde79a3f86
state active
severity critical
type "Processing Error Alarm"
startsAt 2020-04-17T17:55:57.084Z
source tfchan-dev
labels [ "chart: smi-show-tac" "chartVersion: 0.1.0-helmfail-0108-200310183805-6888120"
"component: ops-center" "exported_release: cee-smi-show-tac" "instance: 192.168.190.28:8082"
"job: kubernetes-pods" "namespace: cee" "pod: ops-center-cee-ops-center-5ccddd5d9f-6rffw"
"pod_template_hash: 5ccddd5d9f" "release: cee-ops-center" ]
annotations [ "description: Helm chart smi-show-tac/cee deployment failed" "summary: Helm
chart failed to deploy for 5 minutes" ]
```



Note If SNMP Trapper is configured, this alert goes to the external SNMP receiver as an SNMP trap. For instance, when there is already a conflict of resources, the Helm deployment fails.

Viewing Alert Logger

The Alert Logger stores all the generated alerts by default. You can view the stored alerts using the following **show** commands.

```
show alert history { detail | summary }
```

show alert active { detail | summary }

You can narrow down the result using the following filtering options:

- **annotations** – Specifies the annotations of the alert.
- **endsAt** – Specifies the end time of the alert.
- **labels** – Specifies the additional labels of the alert.
- **severity** – Specifies the severity of the alert.
- **source** – Specifies the source of the alert.
- **startsAt** – Specifies the start time of the alert.
- **type** – Specifies the type of the alert.

You can view the history of configured alerts using **show alerts history** command.

The following examples displays the history of the alerts configured in the system:

Example:

```
cee# show alerts history summary
NAME UID SEVERITY STARTS AT DURATION SOURCE SUMMARY
-----
k8s-pod-crashing-loop 13218bfedfb7 critical 11-02T19:42:40 3m50s upf-cm-tb16-2-cm1 Pod
cee-global/alert-logger-56f85f54df-wdppb (alert-logger) is restarting 1.01 times / 5 minutes.
k8s-pod-crashing-loop bf8f6b0e167c critical 11-02T19:42:40 3m50s upf-cm-tb16-2-cm1 Pod
cee-global/pgpool-5cc9d4b44f-4kklz (pgpool) is restarting 1.01 times / 5 minutes.
k8s-pod-crashing-loop 840f362e970e critical 11-02T19:42:40 3m50s upf-cm-tb16-2-cm1 Pod
cee-global/grafana-5b9779c7d6-hmptk (grafana) is restarting 1.01 times / 5 minutes.
k8s-pod-crashing-loop 40f4de09d667 critical 11-02T19:42:30 3m50s upf-cm-tb16-2-cm1 Pod
cee-global/pgpool-5cc9d4b44f-gwdpp (pgpool) is restarting 1.01 times / 5 minutes.
k8s-pod-not-ready 3adel1624bfa8 critical 11-02T19:40:40 40s postgres-0 Pod
cee-global/postgres-0 has been in a non-ready state for longer than 1 minute.
```

The following examples displays a detailed history of the alerts configured in the system:

```
cee# show alerts history detail
alerts history detail k8s-pod-crashing-loop 13218bfedfb7
severity critical
type "Processing Error Alarm"
startsAt 2020-11-02T19:42:40.400Z
endsAt 2020-11-02T19:46:30.400Z
source upf-cm-tb16-2-cm1
summary "Pod cee-global/alert-logger-56f85f54df-wdppb (alert-logger) is restarting 1.01
times / 5 minutes."
labels [ "alertname: k8s-pod-crashing-loop" "cluster: upf-cm_cee-global" "component:
kubernetes-metrics" "container: alert-logger"
"hostname: upf-cm-tb16-2-cm1" "instance: 192.168.211.203:8080" "job: kubernetes-pods"
"monitor: prometheus"
"namespace: cee-global" "pod: alert-logger-56f85f54df-wdppb" "pod_template_hash: db7bf9f7"
"release: cee-global-cnat-monitoring" "replica: upf-cm_cee-global" "severity: critical" ]
annotations [ "summary: Pod cee-global/alert-logger-56f85f54df-wdppb (alert-logger) is
restarting 1.01 times / 5 minutes."
"type: Processing Error Alarm" ]
```

You can view the active using the **show alerts active** command.

Example:

```
show alerts active summary
NAME UID SEVERITY STARTS AT SOURCE SUMMARY
-----
server-alert 02232d49cccd minor 10-29T06:09:04 upf-4 PS_RDNDNT_MODE: Power Supply redundancy
is lost or non-redundant: Check Redundancy Policy or resear/replace Power Supply
server-alert f97ec27bc318 minor 10-29T06:09:04 cm-2 PS_RDNDNT_MODE: Power Supply redundancy
is lost or non-redundant: Check Redundancy Policy or resear/replace Power Supply
watchdog 0dbfe73527ad minor 10-29T06:07:58 System This is an alert meant to ensure that the
entire alerting pipeline is functional. This alert is always firing, therefore it should
always be firing...
```

Example:

```
show alerts active detail
alerts active detail server-alert 359fe8fd1dd8
severity warning
type "Equipment Alarm"
startsAt 2020-10-29T06:09:04.243Z
source cm-2
summary "Storage Virtual Drive 0 Degraded: please check the storage controller, or resear
the storage drive"
labels [ "alertname: server-alert" "cluster: tb16-2" "description: Storage Virtual Drive 0
Degraded:
please check the storage controller, or resear the storage drive" "fault_id:
sys/rack-unit-1/board/
storage-SAS-MRAID/vd-0/fault-F1008" "id: 3523411968" "monitor: prometheus" "replica: tb16-2"
"server: cm-2" "severity: warning" ]
annotations [ "dn:
tb16-2/cm-2/sys/rack-unit-1/board/storage-SAS-MRAID/vd-0/fault-F1008/3523411968"
"summary: Storage Virtual Drive 0 Degraded: please check the storage controller, or resear
the
storage drive" "type: Equipment Alarm" ]
```

Enabling SNMP Traps

Use the following configuration to enable the SNMP Traps.

```
configure
  snmp-trapper enable true
  snmp-trapper { v2c-target target | v3-target target | v3-engine-id
source_engine_id }
    community [ community_string ]
    port [ port ]
  exit
  snmp-trapper source-ip-routes [ vip_options ]
  exit
```

NOTES:

- **snmp-trapper enable true** – Enables the snmp-trapper parameters
- **v2c-target|v3-target [target]** – Specifies the list of SNMP v2c and v3 trap receivers.
- **community [community_string]** – Specifies the SNMP Trap receiver community.
- **v3-engine-id source_engine_id** – Specifies the source engine ID for the v3 traps. *source_engine_id* must be an hexagonal string. For instance, 80004f.

- **port** [*port*] – Specifies the SNMP Trap receiver port. *port* must be an integer in the range of 0 through 65535. The default value is 162.
- **source-ip-routes** [*vip_options*] – Enables binding to source IP for SNMP routing. *vip* specifies the virtual IP (VIP) address. The different options for virtual IP addresses include:
 - **default-external-vip** – Specifies the default external VIP for source IP routing.
 - **internal-vip** – Specifies the internal VIP for source IP routing.
 - **source-external-vips** -Specifies the external VIP per namespace.

Disabling SNMP Traps

Use the following configuration to disable SNMP Traps.

```
configure
  no snmp-trapper enable
  exit
```

NOTES:

- **no snmp-trapper enable** - Disables SNMP Traps.

Configuring Bulk Statistics

Bulk statistics provide a mechanism to view the summary of the CEE metrics. You can configure bulk statistics to pull the CEE metrics periodically. Also, you can download the metrics in Comma-Separated Value (CSV) format through Secure File Transfer Protocol (SFTP).

Use the following configuration to configure bulk statistics in CEE Ops Center.

```
configure
  bulk-stats enable [ true ]
  bulk-stats external-ip [ ipv4_address ]
  bulk-stats external-port [ port ]
  bulk-stats interval-minutes [ interval ]
  bulk-stats pod-query [ pod_query ] default-value value
  bulk-stats prune-interval-days [ prune_interval ]
  bulk-stats query [ query ]
  bulk-stats user [ user ]
  bulk-stats vnf-name [ vnf ]
  bulk-stats global-default-value [ default_value ]
  bulk-stats vnf-alias [ vnf_alias ]
  exit
```

NOTES:

- **bulk-stats enable** [*true*] – Enables the bulk statistics.
- **global-default-value** [*default_value*] – Specifies the default value used in bulk-stats, if **pod-query** or **query** fails to return any value.

- **external-ip** [*ipv4_address*] – Specifies the external IP for downloading the bulk statistics over SFTP.
- **external-port** [*port*] – Specifies the external port for downloading the bulk statistics over SFTP.
- **interval-minutes** [*interval*] – Specifies the time interval (in minutes) to create the bulk statistics.
- **pod-query** [*pod_query*] **default-value** *value* – Specifies the query to execute for retrieving the bulk statistics data.] **default-value** *value* is the default value used in bulk-stats, if the configured **pod-query** fails to return any value. *value* will override the **global-default-value**
- **prune-interval-days** [*prune_interval*] – Prunes the interval (in days) to remove the bulk statistics.
- **query** [*query*] – Specifies the query to execute for retrieving the bulk statistics.
- **user** [*user*] – Specifies the user authorized to download the bulk statistics files.
- **vnf-name** [*vnf*] – Specifies the VNF name (namespace) to add in the bulk statistics CSV file.
- **vnf-alias** [*vnf_alias*] – Specifies the VNF alias for a specified namespace.

The following example generates query for current PDU per 4G session.

Example:

```
cee# configure terminal
  bulk-stats enable true
  bulk-stats user admin
  bulk-stats external-ip 172.16.181.41
  bulk-stats external-port 2222
  bulk-stats vnf-name lbucs009
  bulk-stats query 4G_current_pdu_sessions
  expression "sum(smf_up_session_counters{app_name=\"SMF\",rat_type=\"EUTRA\"})"
  label      4G_current_pdu_sessions
  exit
```

The following example generates query for current PDU per 4G IPv6 session.

Example:

```
cee# configure terminal
  bulk-stats enable true
  bulk-stats user admin
  bulk-stats external-ip 172.16.181.41
  bulk-stats external-port 2222
  bulk-stats vnf-name lbucs009
  bulk-stats query 4G_current_pdu_sessions_IPv6
  expression
  "sum(smf_up_session_counters{app_name=\"SMF\",rat_type=\"EUTRA\",pdu_type=\"ipv6\"})"
  label      4G_current_pdu_sessions_IPv6
  exit
```

Retrieving Bulk Statistics

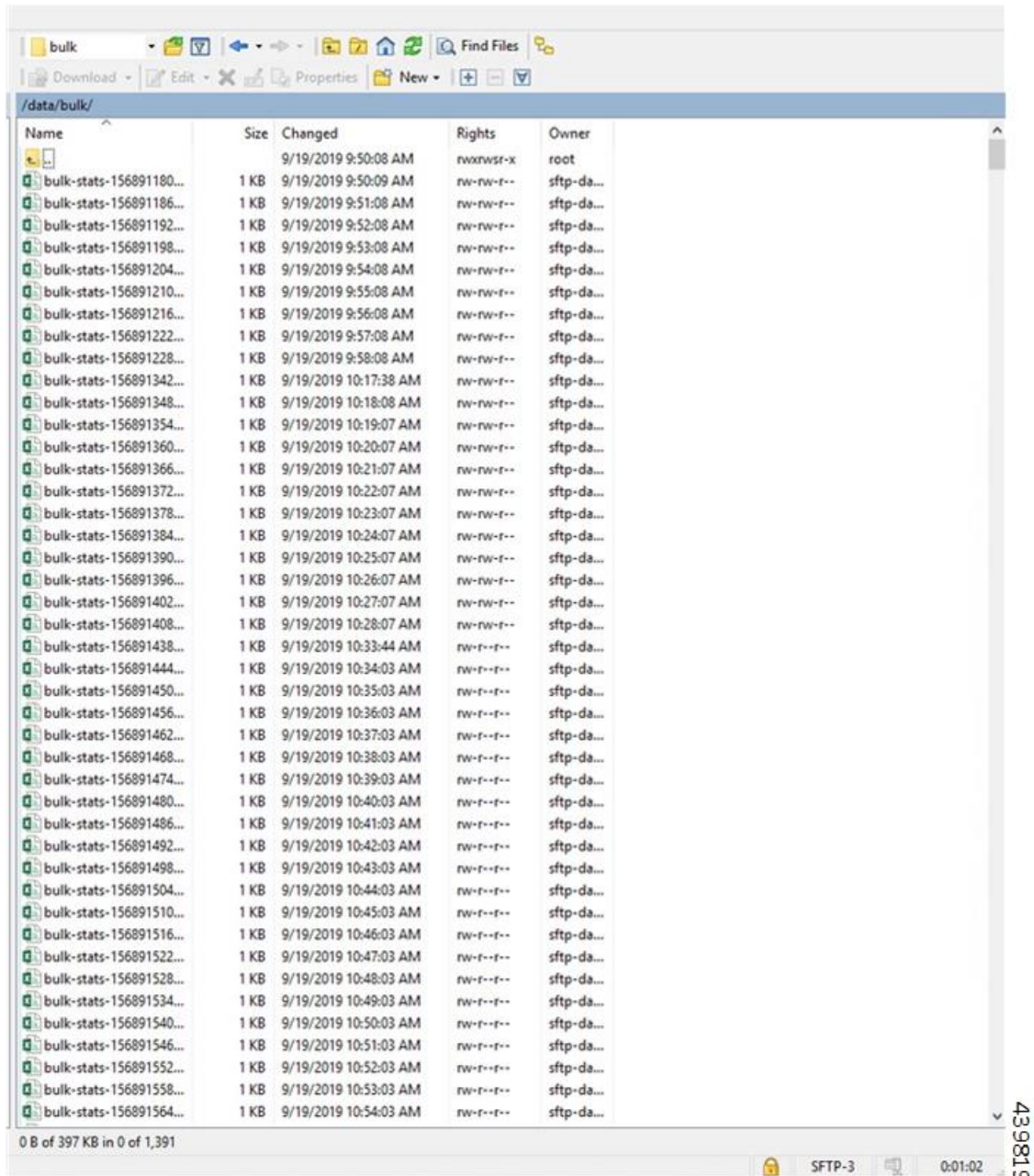
The CEE stores the Bulk statistics in the CSV format. You can download the stored files from the external host. You must configure the external IP and port to download the bulk statistics.

Use the following command from the external host, which is accessible to the CEE cluster, to download or retrieve the bulk statistics in CSV format.

```
scp -P [external-port] [user]@[external-ip]:/data/[bulk | rate]/[filename].csv [local-folder]
```

Also, you can use any of the Graphical User Interface (GUI) SFTP tool to browse and download the CSV files. A SFTP tool displaying the directory where bulk statistics are stored in CSV format is shown below:

Figure 8: Bulk Statistics - GUI



The following example displays the various parameters in bulk statistics.

Example:

UID	NAMESPACE	METRIC		LABELS
		ALIAS	VALUE	
439bd4f3d7c8	*	active-alerts	1.0	[alertname=watchdog]
50113c94b989	cee-global	configuration-change-total	3.0	[source=System]
5ada6437a102	cee-global	cpu-core-count	48.0	[hostname=ott-bm2-cm-cm-1]
9a918f9f153a	*	cpu-idle	95.793	[hostname=ott-bm2-cm-cm-1]
3cac0a6ad9ee	*	cpu-iowait	0.003	[hostname=ott-bm2-cm-cm-1]
52b70483c10e	*	cpu-softirq	0.229	[hostname=ott-bm2-cm-cm-1]
88f3c5d2cc32	*	cpu-steal	0.0	[hostname=ott-bm2-cm-cm-1]
2c2354f17788	*	cpu-system	1.485	[hostname=ott-bm2-cm-cm-1]
137b898a8afe	*	cpu-user	2.205	[hostname=ott-bm2-cm-cm-1]
76d3a2158b50	cee-global	daemonset-ready-percent	100.0	[daemonset=blackbox-exporter]
44d0bfe7d92d	kube-system	daemonset-ready-percent	100.0	[daemonset=calico-node]
d2e91d076768	cee-global	daemonset-ready-percent	100.0	[daemonset=core-retriever]
ec70bdc6dbaf	kube-system	daemonset-ready-percent	100.0	[daemonset=journald-adapter]
e13a31621bbc	smi-vips	daemonset-ready-percent	100.0	[daemonset=keepalived]
3583e73ab8c8	kube-system	daemonset-ready-percent	100.0	[daemonset=kube-proxy]
a78d2ca5a7c4	cee-global	daemonset-ready-percent	100.0	[daemonset=logs-retriever]
04d9a0c4691d	kube-system	daemonset-ready-percent	100.0	[daemonset=maintainer]
376fbe4611bd	cee-global	daemonset-ready-percent	100.0	[daemonset=node-exporter]
d109bf9be31d	cee-global	daemonset-ready-percent	100.0	[daemonset=path-provisioner]
11090fd5e91f	cee-global	daemonset-ready-percent	100.0	[daemonset=restart-kubelet]
d770ae176453	smi-secure-access	daemonset-ready-percent	100.0	[daemonset=secure-access-controller]
b0344050b3d5	kube-system	daemonset-ready-percent	100.0	[daemonset=user-password-monitor]
48ce4437eb7b	cee-global	deployment-ready-percent	100.0	[deployment=alert-logger]
8f59873fff50	cee-global	deployment-ready-percent	100.0	[deployment=alert-router]
6119200c32be	cee-global	deployment-ready-percent	100.0	[deployment=alertmanager-config-sync]
28fb43ce4d90	kube-system	deployment-ready-percent	100.0	[deployment=calico-kube-controllers]
2e57b5973770	cee-global	deployment-ready-percent	100.0	[deployment=cee-global-product-documentation]
69bcc641b4b	kube-system	deployment-ready-percent	100.0	[deployment=cluster-cert-maintainer]
2803753e1298	smi-cm	deployment-ready-percent	100.0	[deployment=cluster-files-offline-smi-cluster-deployer]
948a96222d29	kube-system	deployment-ready-percent	100.0	[deployment=coredns]
c5006862911f	cee-global	deployment-ready-percent	100.0	[deployment=grafana]

```

100.0
346b7b8c0b54 cee-global deployment-ready-percent [
deployment=grafana-dashboard-metrics ] 100.0
e2bece200bd8 cee-global deployment-ready-percent [ deployment=kube-state-metrics
] 100.0
2b99fde0f918 nginx-ingress deployment-ready-percent [
deployment=nginx-ingress-ingress-nginx-controller ] 100.0
a01523e2af8d nginx-ingress deployment-ready-percent [
deployment=nginx-ingress-ingress-nginx-defaultbackend ] 100.0
cc8a64825b3e cee-global deployment-ready-percent [
deployment=ops-center-cee-global-ops-center ] 100.0
5e74886b3429 smi-cm deployment-ready-percent [
deployment=ops-center-smi-cluster-deployer ] 100.0
72f4818bff4f smi-ops-control deployment-ready-percent [
deployment=opscenter-controller ] 100.0
23a868c32ce9 cee-global deployment-ready-percent [ deployment=pgpool ]
100.0
2c2d372c36d5 cee-global deployment-ready-percent [ deployment=prometheus-rules
] 100.0
3950d7cbde90 cee-global deployment-ready-percent [
deployment=prometheus-scrapeconfigs-synch ] 100.0
a7bdb748677a cee-global deployment-ready-percent [ deployment=pv-manager ]
100.0
161b4d128721 cee-global deployment-ready-percent [ deployment=pv-provisioner
] 100.0
2c4aa52f6c98 cee-global deployment-ready-percent [ deployment=show-tac-manager
] 100.0
70b960ace2f0 cee-global deployment-ready-percent [
deployment=smart-agent-cee-global-ops-center ] 100.0
0a3fb053bbec smi-certs deployment-ready-percent [ deployment=ss-cert-provisioner
] 100.0
7fd1e489a7e5 cee-global deployment-ready-percent [
deployment=thanos-query-frontend-hi-res ] 100.0
72587dc5987f cee-global deployment-ready-percent [ deployment=thanos-query-hi-res
] 100.0
b8140482f112 * entitlement-status [ tag=System ]
0.0
a3e2bc7a1b71 * filesystem-data-avail-bytes [ hostname=ott-bm2-cm-cm-1 ]
626088103936.0
29dced1e7b92 * filesystem-root-avail-bytes [ hostname=ott-bm2-cm-cm-1 ]
6002253824.0
1eb34dc3b330 * k8s-pods-status [ phase=Failed ]
0.0
8d23272d645a * k8s-pods-status [ phase=Pending ]
0.0
65d9342f3c90 * k8s-pods-status [ phase=Running ]
56.0
74c9de9ac37e * k8s-pods-status [ phase=Succeeded ]
0.0
8a12153befc2 * k8s-pods-status [ phase=Unknown ]
0.0
4ba6cc59b00c * kubelet-node-status [ condition=DiskPressure ]
0.0
e343dc31dfcf * kubelet-node-status [ condition=MemoryPressure ]
0.0

```

Grafana

Grafana is an open source data visualization tool used for displaying application metrics in interactive dashboards.

Accessing Grafana

In the Cisco Common Execution Environment (CEE), you can define and deploy Grafana dashboards using:

- Git server pods, and
- ConfigMap

managed through a combination of Kubernetes Custom Resource Definitions (CRDs) and Git repositories. Here's a summary of the process

You can access Grafana login page through Ingress using any of the standard web browsers. For instance, using Google Chrome navigate to the Grafana login page with the following Ingress URL:

https://grafana.<ipv4_address>.<customer_specific_domain_name>

NOTES:

- *customer_specific_domain_name* - Specifies the customer's domain name.

Figure 9: Grafana – Login Page



Important Authentication to Grafana happens through the CEE Ops Center since Grafana is associated with it. If the CEE Ops Center is configured with Lightweight Directory Access Protocol (LDAP), Grafana authenticates through LDAP.



Important Third-Party Software Vulnerability - The Content Security Policy support in Grafana uses the *unsafe-eval* version of **script-src** because AngularJS is not fully migrated in Grafana.

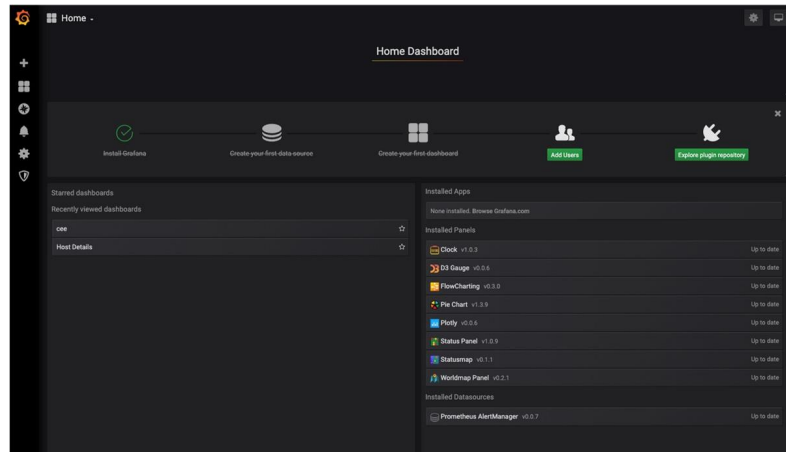
Using Dashboards


The Grafana home page lists the dashboards bundled with CEE. The dashboards provide an overall status of the system.

To view the dashboards, perform the following steps:

1. Navigate to the Grafana Login page using any standard web browser.
2. Login to Grafana to view the home page.

Figure 10: Grafana – Home Page



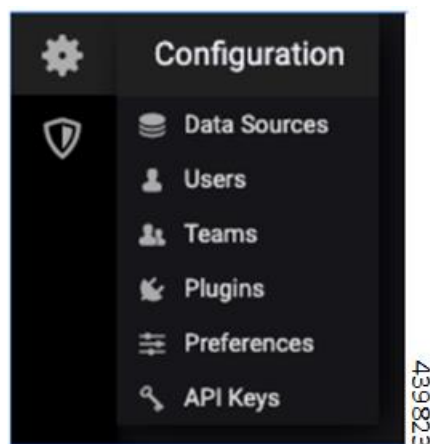
3. Click  icon on the left pane.
4. Click **Dashboards**.
 - a. By default, you can view the following dashboards.
 1. **Host Summary** – This dashboard provides an overview of CPU, Memory, Disk I/O Utilization, Filesystem Fullness and Filesystem Fill UP time. You can choose the machine from **Machine** drop-down list to view the host summary of individual machines available in the clusters.
 2. **Host Details** – This dashboard provides a detailed view on each of the following categories:
 - Basic CPU/ Mem / Disk Gauge
 - Basic CPU/ Mem / Disk Info
 - Basic CPU / Mem Graph
 - Basic Net / Disk Info
 - CPU Memory Net Disk
 - Memory Detail Meminfo
 - Memory Detail Vmstat
 - Memory Detail Vmstat Counters
 - System Detail
 - Disk Detail
 - Filesystem Detail
 - Network Traffic Detail
 - Network Sockstat

- Network Netstat
- Network Netstat TCP
- Network Netstat TCP Linux MIPs
- Network Netstat UDP
- Network Netstat ICMP
- Node Exporter



You can create new Dashboards or Data Sources in Grafana through settings  tab.

Figure 11: Settings Tab



Important The bundled dashboards are provisioned statically and cannot be modified. However, you can copy or clone the dashboards before saving the changes.

The following example imports custom Grafana dashboard from a Git repository.

Example:

```
cee# configure terminal
grafana dashboards sample
git-url https://wwin-github.cisco.com/mobile-cnat-sample/sample-dashboards.git
exit
```

Persistent Grafana Dashboards

The custom Grafana dashboards are made persistent so that the dashboards are not lost during pod restart, node shutdown or any upgrade. The Grafana dashboards are stored in the */mnt/stateful_partition/data/cee-global/data-postgres-x* directory.

The Grafana pod has built-in APIs to manipulate the dashboard resources. These APIs will allow the user to perform the following functions:

- Export or backup all Grafana dashboards as files
- Restore Grafana dashboards from backup
- Create a new Grafana dashboard
- Modify the existing Grafana dashboard
- Authenticate API using basic authentication

The valid users of the dashboards are either local users or TACACS users who have access to CEE Ops Center. You can configure the **grafana enable-basic-auth { true | false }** CLI in CEE Ops Center to enable or disable basic authentication. When enabled, you can perform all CRUD operations on the dashboards with the existing Grafana HTTP API.

User Management in Grafana

You can create and manage the local users and user groups through the CEE Ops Center as described in the [Provisioning Local Users, on page 42](#) section.

Configuring Ingress for Prometheus

SMI allows a Kubernetes ingress resource with basic authentication to be added to Prometheus using Grafana. This allows SMI metrics to be used as a data source by an external Grafana instance.

1. Login to Grafana User Interface.
2. Select **Configuration Data Sources**.
3. Click **Add data source**.
4. Enter a name for the data source in the **Name** field.
5. Select **Type Prometheus**.
6. Enter the new ingress URL (e.g. prometheus-xxx) in the **URL**.
7. Enable and configure **Basic Auth**
8. Provide the same credentials as configured via the SMI Ops Center in the **User** and **Password** fields under **Basic Auth Details**.
9. Enable **Skip TLS Verify** and keep the remaining options as it is.
10. Click **Save & Test**. The result should be successful and the data source is ready to use.

Provisioning Local Users

A new YANG model is introduced in SMI to support user management in compliance with Cisco Secure Development Life-cycle (CSDL) requirements.



Note This new YANG model is applicable to SMI Cluster Manager and all other product Ops Centers.

User Management

This chapter describes how to create and manage local users using the Ops Center CLI (for both the products and SMI Cluster Manager Ops Center).



Important Users with administrator privileges can add, modify, and delete other users and groups. All the other users only have privileges to change their own password.

Adding a User

To add a new user, use the following configurations:

configure

```
smiuser add-user username username password password
exit
```

NOTES:

- **smiuser add-user:** Adds a new local user.
- **username** *username*: Specifies the name of the user.
username must be alphanumeric string.
- **password** *password*: Specifies the password. The password must meet the following criteria:
 - Minimum 8 characters in length.
 - Contain at least one lowercase character.
 - Contain at least one uppercase character.
 - Contain at least one numeric character.
 - Contain at least one special character, which includes the following:
 - ['~', '@', '#', '%', '^', '&', '*', '(', ')', '_', '+', '!', '-', '=', '[', ']', ':', '"', ';', '\', '|', '<', '>', '?', ',', '!', '/', '\$']



Important Your password must not contain the '{' and '}' special characters.

- Password must not start with '\$'.
 - Password must not be simple or based on dictionary word.
 - Do not reuse passwords.
- Use the following command to configure the number of passwords to keep in history:
- ```
password requisite pam_pwhistory.so debug enforce_for_root remember=12
```

- Minimum number of days that are allowed between password changes is seven.

The following example adds a new user called 'user1' and assigns the password for the new user.

```
cee# configure terminal
 smiuser add-user username user1 password Cisco@123
message User added
```

The following example adds a new user called 'user2' and assigns the password for the new user.

```
cee# configure terminal
 smiuser add-user username user2 password Cisco@12345
message User added
```

In the following example, when an existing user name (user2) is added as a new user, the Ops Center displays an error message.

```
cee# configure terminal
 smiuser add-user username user2 password Cisco@12345
message User already exists
```

## Creating Unprivileged Users with SSH Key

The SMI Cluster Manager allows creating unprivileged users on cluster nodes with SSH key access. These users will remain even after the SMI Cluster Manager is upgraded. Also, the SMI Cluster Manager considers the users created with the comment *smi.user* to be managed by the Cluster Manager. If an existing user, who is not an *smi.user*, is added to the configuration, the SMI Cluster Manager throws an error during cluster synchronization to prevent damaging or blocking communication to the system.

To add a SSH key and password to an user on all the nodes, use the following configuration:

```
configure
node-defaults os users username
 password password
 authorized-keys key_name
 algorithm ssh_algorithm
 key-data key_data
 exit
authorized-keys key_name
 algorithm ssh_algorithm
 key-data key_data
 exit
exit
```

To add a SSH key and password to an user on a specific node, use the following configuration:

```
configure
node node_name os users username
 password password
 authorized-keys key_name
 algorithm ssh_algorithm
 key-data key_data
 exit
authorized-keys key_name
 algorithm ssh_algorithm
 key-data key_data
```

```
exit
exit
```

**NOTES:**

- **node-defaults os users** *username* - Specifies the default value applicable to all the nodes for the selected user. *username* is the name of the user to be created.
- **node** *node\_name* **os users** *username* - Specifies the default value applicable to the specific node for the selected user. *node\_name* is the name of the specific node. *username* is the name of the user to be created.
- **password** *password* - Specifies the password used for authentication.
- **authorized-keys** *key\_name* - Specifies the name of the SSH key.
- **algorithm** *ssh\_algorithm* - Specifies the SSH algorithm used for generating the SSH key. For example, SSH-RSA or SSH-Ed25519 algorithm.
- **key-data** *key\_data* - Specifies the generated SSH key.

## Deleting a User

To delete a user, use the following configuration:

```
configure
 smiuser delete-user username username
exit
```

**Note**

- **smiuser delete-user** - Deletes a local user.
- **username** *username* - Specifies the name of the user.  
*username* must be alphanumeric string.

The following example deletes a user called 'user2'.

```
cee# configure terminal
 smiuser delete-user username user2
message User deleted
```

In the following example, when a non-existing user is deleted, the Ops Center displays an error message.

```
cee# configure terminal
 smiuser delete-user username user2
message User does not exist
```

## Modifying the Password

To modify the password (for self), use the following configuration:

```
configure
 smiuser change-self-password current_password current_password new_password
 new_password
 confirm_password new_password password_expire_days number_of_days
exit
```

**Note**

- **smiuser change-password** - Modifies the password for an user.
- **current\_password** *current\_password* - Specifies the current password for an user.
- **new\_password** *new\_password* - Assign a new password for the user. For information on password policy, see [Adding a User](#) section.
- **confirm\_password** *new\_password* - Enter the newly assigned password one more time.
- **password\_expire\_days** *number\_of\_days* - (Optional) Specifies the expiry date of the password. The default value is 180 days.

The following example updates the password for the current user.

```
cee# configure terminal
 smiuser change-self-password current_password Cisco@123 new_password Cisco@345
 confirm_password Cisco@345 password_expire_days 180
message Password updated successfully
```

The following example updates the password for the user called 'user1' without assigning the password expiry date.

```
cee# configure terminal
 smiuser change-self-password current_password Cisco@123 new_password Cisco@345
 confirm_password Cisco@345
message Password updated successfully
```

## Reset the Administrator Password

You can reset the administrator password if you have access to the K8s Cluster through **kubect1** command-line utility.

To reset the administrator password:

1. Enter the Ops Center Pod's EXEC mode.
2. Use the following command to reset the administrator password.

```
kubect1 exec -it <pod_name> -n <pod_namespace> /usr/local/bin/reset-admin
```

3. Enter the new password when prompted.

**NOTES:**

- **kubect1 exec -it** - Executes a command inside a container. **-it** passes the standard input stream to the container or TTY.
- **<pod\_name> -n** - Specifies the name of the Pod. **-n** specifies the namespace scope for this CLI request.
- **<pod\_namespace>** - Specifies the namespace of the Pod.
- **/usr/local/bin/reset-admin** - Resets the administrator password.

## Modifying the Password for Other Users

You can modify the password for other users using the following configuration:

```

configure
 smiuser change-password username username current_password current_password
 new_password new_password
 confirm_password new_password password_expire_days number_of_days
 exit

```



- Note**
- **smiuser change-password** - Modifies the password for an user.
  - **username *username*** - Specifies the name of the user.  
*username* must be alphanumeric string.
  - **current\_password *current\_password*** - Specifies the current password for an user.
  - **new\_password *new\_password*** - Assign a new password for the user. For information on password policy, see [Adding a User](#) section.
  - **confirm\_password *new\_password*** - Enter the newly assigned password one more time.
  - **password\_expire\_days *number\_of\_days*** - (Optional) Specifies the expiry date of the password. The default value is 180 days.

The following example updates the password for the user called 'user1'.

```

cee# configure terminal
 smiuser change-password username user1 current_password Cisco@123 new_password Cisco@345
 confirm_password Cisco@345 password_expire_days 180
message Password updated successfully

```

The following example updates the password for the user called 'user1' without assigning the optional password expiry date.

```

cee# configure terminal
 smiuser change-password username user1 current_password Cisco@123 new_password Cisco@345
 confirm_password Cisco@345
message Password updated successfully

```

The following example updates the password for the user called 'user1' without assigning the password expiry date.

```

cee# configure terminal
 smiuser change-password username user1 current_password Cisco@123 new_password Cisco@345
 confirm_password Cisco@345
message Password updated successfully

```

The following example updates the password for the user called 'user1' with an existing password.

```

cee# configure terminal
 smiuser change-password username user1 current_password Cisco@345 new_password Cisco@345
 confirm_password Cisco@345
message Password has been already used

```

The following example updates the password for the user called 'user1' with different values for new password and confirm password parameters.

```

cee# configure terminal
 smiuser change-password username user1 current_password Cisco@345 new_password Cisco@345
 confirm_password Cisco@567
message Passwords do not match

```

## Updating the Password Length

To update the length of the password, use the following configuration:

```
configure
smiuser update-password-length length number_of_characters
exit
```



- 
- Note**
- **smiuser update-password-length** - Updates the length of the password.
  - **length *number\_of\_characters*** - Specifies the length of the password. *number\_of\_characters* must be a numeric value.
- 

The following example updates the minimum length of the password to 10 characters.

```
cee# configure terminal
 smiuser update-password-length length 10
message Password updated successfully
```

## Group Management

This chapter describes how to create and manage user groups using the Ops Center CLI (of both the products and SMI Cluster Manager).

### Adding a User Group

To add a user group, use the following configuration:

```
configure
smiuser add-group groupname group_name
exit
```



- 
- Note**
- **smiuser add-group** - Adds a new user group.
  - **groupname *group\_name*** - Specifies the name of the user group. *group\_name* must be a alphanumeric value.
- 

The following example adds a new user group called 'group1'.

```
cee# configure terminal
 smiuser add-group groupname group1
message Group added
```

In the following example, when a user group that already exists is added, the Ops Center displays an error message.

```
cee# configure terminal
 smiuser add-group groupname group1
message Group already exists
```

## Deleting a User Group

To delete a user group, use the following configuration:

```
configure
smiuser delete-group groupname group_name
exit
```



- 
- Note**
- **smiuser delete-group** - Deletes a user group.
  - **groupname** *group\_name* - Specifies the name of the user group. *group\_name* must be a alphanumeric value.
- 

The following example deletes a new user group called 'group2'.

```
cee# configure terminal
 smiuser delete-group groupname group2
message Group deleted
```

In the following example, when a user group that does not exist is deleted, the Ops Center displays an error message.

```
cee# configure terminal
 smiuser delete-group groupname group2
message Group does not exist
```

## Assigning an User to an User Group

To assign an user to an user group, use the following configuration:

```
configure
smiuser assign-user-group username username group group_name
exit
```



- 
- Note**
- **smiuser assign-user-group** - Assigns an user to a user group.
  - **username** *username* - Specifies the name of the user. *username* must be alphanumeric value.
  - **groupname** *group\_name* - Specifies the name of the user group. *group\_name* must be a alphanumeric value.
- 

The following example assigns an user called 'user1' to a group called 'group1'.

```
cee# configure terminal
 smiuser assign-user-group username user1 group group1
message User assigned to group successfully
```

The following example assigns a non-existing user to an existing group.

```
cee# configure terminal
 smiuser assign-user-group username user20 group group1
message User does not exist
```

The following example assigns a non-existing group to an existing user.

```
cee# configure terminal
 smiuser assign-user-group username user1 group group10
message Group does not exist
```

## Unassigning a User from a User Group

To unassign a user from a user group, use the following configuration:

```
configure
 smiuser unassign-user-group username username group group_name
exit
```



- 
- Note**
- **smiuser unassign-user-group** - Removes an user from a user group.
  - **username** *username* - Specifies the name of the user. *username* must be alphanumeric value.
  - **groupname** *group\_name* - Specifies the name of the user group. *group\_name* must be a alphanumeric value.
- 

The following example removes an user from a group.

```
cee# configure terminal
 smiuser unassign-user-group username user1 group group1
message User un-assigned from group successfully
```

The following example removes a non-existing user from a group.

```
cee# configure terminal
 smiuser unassign-user-group username user10 group group1
message User is not a member of this group
```

The following example removes an user from an non-existing group.

```
cee# configure terminal
 smiuser unassign-user-group username user1 group group10
message Group does not exist
```

## Log Forwarding

Log Forwarding allows you to forward the log entries (including the host and container-level log entries) stored in JournalD to the external collectors. SMI supports target hosts such as Fluent-x, Splunk, Loki and Grafana Cloud for log forwarding.

To stream data, Fluent-x uses the Forward protocol and Splunk uses HTTPS. Fluent Bit sends logs to Grafana Cloud by providing the appropriate URL and ensuring that TLS is enabled.



- 
- Note** SMI enables only one target host of Grafana Cloud type for logs forwarding. However, Splunk, Fluent-bit, and Loki can be enabled in parallel.
- 

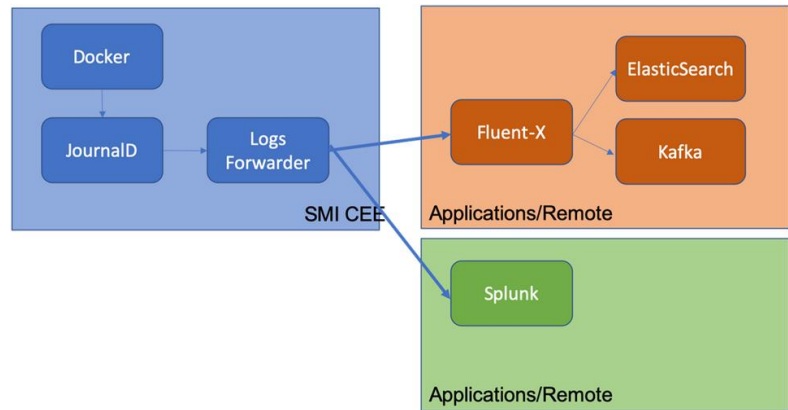
The CEE utilizes FluentD for buffering and persistent connection support. FluentD is an open-source data collection and consumption software. Using FluentD, you can collect logging events from various sources and unify it for better usage and understanding. For more information of FluentD, see <https://docs.fluentd.org/>.

By default, FluentD is configured with the following parameters to support buffering and keepalive:

```
total_limit_size 1GB
chunk_limit_size 8MB
compress text
flush_mode interval
flush_interval 5s
overflow_action drop_oldest_chunk
retry_timeout 1h
```

The following figure depicts the high-level Log Forwarding architecture:

**Figure 12: Log Forwarding Architecture**



## Prerequisites

You can enable Log Forwarding in CEE to forward log entries to the external collectors. You must ensure that the K8s cluster is installed in the CEE before enabling Log Forwarding.

## Requirements

The following are the requirements for enabling Log Forwarding:

### Fluent-x

1. The target endpoint must be a Fluentbit or FluentD instance or cluster with the Forward protocol input plugin enabled.
2. The endpoint must be hosted within the Kubernetes clusters or a remote system with network reachability.

## Enabling Log Forwarding

This section describes the procedure involved in enabling Log Forwarding on Fluent-x and Splunk.

This section describes the procedure involved in enabling Log Forwarding on Fluent-x.

### Enabling Log Forwarding on Fluent-x

Use the following configuration to enable Log Forwarding on Fluent-x.

```
configure
logging fluent host fluentbit/fluentd_endpoint_fqdn/ipv4_address port endpoint_port
```

**NOTES:**

- **logging fluent** – Specifies the Fluent forwarding parameters.
- **host *fluentbit/fluentd\_endpoint\_fqdn/ipv4\_address*** – Specifies the Fluentbit or Fluentd instance host information.
- **port *endpoint\_port*** – Specifies the Fluentbit or Fluentd instance port information.

The log forwarding to an external Fluent-D or Fluent-Bit instance, where logs can be streamed to supporting application such as ElasticSearch.

**Example:**

```
cee# configure terminal
logging fluent host 172.16.181.41 port 8001
exit
```

**Enabling Log Forwarding on Splunk**

Use the following configuration to enable Log Forwarding on Splunk.

```
configure
logging splunk host splunk_endpoint_fqdn/ipv4_address port hec_port auth-token
splunk_configured_token
```

**NOTES:**

- **logging splunk** – Specifies the Splunk endpoint.
- **host *splunk\_endpoint\_fqdn/ipv4\_address*** – Specifies the Splunk host information.
- **port *hec\_port*** – Specifies the Splunk port information.
- **auth-token *splunk\_configured\_token*** – Specifies the Splunk Authentication Token for the HTTP Event Collector interface.

The following example configures log forwarding to an external Splunk server.

**Example:**

```
cee# configure terminal
logging splunk host 172.16.181.41 port 8001
exit
```

## Enabling Log Forwarding on StarOS

Use the following configuration to enable Log Forwarding from StarOS.

```
configure
logging
syslog cee_ops_center_listener_ip_address
facility local5
msg-format rfc5424
commit
```

**NOTES:**

- **syslog** - Specifies the syslog messages.
- **cee\_ops\_center\_listener\_ip\_address** - Specifies the CEE Ops Center Listener IP address.
- **facility local5** - Specifies the syslog facility values.
- **msg-format rfc5424** - Specifies the syslog message format.

## Configuring CEE Ops Center as a Listener

You can configure the CEE Ops Center to listen to the logs from StarOS.

Use the following configuration to configure CEE Ops Center as listener:

```
configure
logging
 listener enable
 external-ip cee_ops_center_listener_ip_address
commit
```

**NOTES:**

- **listener enable** - Enables the CEE Ops Center to listen to the logs from StarOS.
- **external-ip***cee\_ops\_center\_listener\_ip\_address* - Specifies the CEE Ops Center Listener IP address.

## Configuring Fluent-D to Support Splunk

You can configure Fluent-Bit to send logs to Fluent-D. When the Fluent-D receives the logs, it forwards the received logs to Splunk.

To configure Fluent-D to support Splunk, use the following configuration:

```
configure
logging splunk host splunk_host
logging splunk port splunk_port
logging splunk auth-token auth_token
```

**NOTES:**

- **logging splunk host** *splunk\_host*—Specify the Splunk host information.
- **logging splunk port** *splunk\_port*—Specify the Splunk port information.
- **logging splunk auth-token** *auth\_token*—Specify Splunk Authentication Token for the HTTP Event Collector interface.

## Configuring Fluent-Bit to Support Splunk

You can configure Fluent-Bit to send logs to Splunk. This configuration is applicable only when you configure the local cluster as the as the Listener and the remote cluster in remote forwarding mode.

When you configure Fluent-Bit to support Splunk, the local logs are sent to Splunk using Fluent-Bit and the remote logs are sent to the fluent listener (Fluent-Bit). The Fluent-Bit in turn forwards the remote logs to Splunk.

To configure Fluent-Bit to support Splunk, use the following configuration:

```

configure
 logging splunk listener enable
 logging splunk listener external-ip external_vip_ip
 logging splunk host splunk_host
 logging splunk port splunk_port
 logging splunk auth-token auth_token

```

#### NOTES:

- **logging splunk listener *enable***—Enable Fluent-Bit to send logs to Splunk.
- **logging splunk listener **external-ip** *external\_vip\_ip***—Specify the external virtual IP address of the local cluster.
- **logging splunk host *splunk\_host***—Specify the Splunk host information.
- **logging splunk port *splunk\_port***—Specify the Splunk port information.
- **logging splunk **auth-token** *auth\_token***—Specify the Splunk Authentication Token for the HTTP Event Collector interface.

## Configuring Fluent-Bit to Support Remote Forwarding

You can configure Fluent-Bit to send logs to the remote cluster.

To configure Fluent-Bit to support remote forwarding, use the following configuration:

```

configure
 logging fluent host remote_cluster_ip
 logging fluent port remote_cluster_port
 logging fluent protocol forward

```

#### NOTES:

- **logging fluent host *remote-cluster-ip***—Specify the Fluent-Bit host information.
- **logging fluent port *remote-cluster-port***—Specify the Fluent-Bit port information.
- **logging fluent **protocol** *outbound\_protocol***—Specify the outbound protocol.

## Configuring Fluent-Bit to Support Remote Listener

You can configure Fluent-Bit to receive logs from the remote cluster.

To configure Fluent-Bit to support remote listening, use the following configuration:

```

configure
 logging splunk listner enable
 logging splunk listner external-ip external_vip_ip

```

#### NOTES:

- **logging splunk listner enable**—Enable Fluent-Bit to support remote listening.
- **logging splunk listner external-ip external\_vip\_ip**—Specify the external virtual IP address of the remote cluster

## Configuring Fluent-Bit to Support Grafana Cloud

You can configure Fluent-Bit to send logs to Grafana Cloud.

To configure Fluent-Bit to enable Grafana Cloud log forwarding, use the following configuration:

### configure

```
logging grafana-cloud host grafana_cloud_host
logging grafana-cloud port grafana_cloud_port
logging grafana-cloud http-user http_user
logging grafana-cloud http-password http_password
```

To configure Fluent HTTP proxy, use the following configuration:

### configure

```
logging proxy http-proxy proxy_url
logging proxy https-proxy proxy_url
logging proxy no-proxy comma_seperated_url
```

### NOTES:

- **logging grafana-cloud host grafana\_cloud\_host**—Specify the host logs.
- **logging grafana-cloud port grafana\_cloud\_port**—Specify the host port. The default port is set to 443.
- **logging grafana-cloud http-user http\_user**—Specify the HTTP user information.
- **logging grafana-cloud http-password http\_password**—Specify the HTTP user password.
- **logging proxy http-proxy proxy\_url**—Specify the HTTP proxy URL.
- **logging proxy https-proxy proxy\_url**—Specify the HTTPS proxy URL.
- **logging proxy no-proxy comma\_seperated\_url**—Specify the comma-separated domain name.

### Labels and Label Keys

To configure the label, use the following configuration:

### configure

```
logging grafana-cloud labels key value
exit
```

To configure the label keys, use the following configuration:

### configure

```
logging grafana-cloud labels-keys [$KEY1,$KEY2]
```

### NOTES:

- By default, the labels for the stream are set to job=fluent-bit, log\_source=cndp, hostname={nodeName}.

- You can configure K8s label keys for the log stream such as container name (`$k8s_container_name`) and namespace (`$k8s_namespace_name`). The label keys must start with `$`.

## Configuring Fluent-Bit to Support Syslog

You can configure Fluent-Bit to send logs to the Syslog server. The Syslog messages are sent over UDP, TCP, or TLS in RFC3164 or RFC5424 format.

To configure log forwarding to the Syslog server, use the following configuration:

```
config
 logging syslog host server_host
 logging syslog mode server_mode
 logging syslog port server_port
 logging syslog syslog_format syslog_format
 logging syslog_maxsize syslog_maxsize
end
```

### NOTES:

- **logging syslog host *server\_host***—Specify the domain or IP address of the remote Syslog server.
- **logging syslog mode *server\_mode***—Specify the TCP, TLS or UDP transport type.
- **logging syslog port *server\_mode***—Specify the TCP, TLS or UDP port of the remote Syslog server.
- **logging syslog syslog\_format *syslog\_format***—Specify the *rfc3164* or *rfc5424* Syslog protocol format to use.
- **logging syslog\_maxsize *syslog\_maxsize***—Specify the maximum size allowed per message. The value must be an integer representing the number of bytes allowed.

## Configuring Fluent Worker to Drop and Retain Logs

To enable CEE log forwarding for Fluent worker pods, use the following configuration.

The filters on Fluent worker pods that intake the logs from each node reduce the volume of logs being forwarded.

```
configure
 logging worker drop-namespace-logs namespace_names
 logging worker drop-pod-logs pod_names
 logging worker exclude-logs-with-annotation true
 logging worker keep-pod-logs pod_names
 logging worker keep-namespace-logs namespace_names
 logging worker drop-os-service-logs service_names
 logging worker remove-keys [keys]
```

### NOTES:

- **logging worker drop-namespace-logs *namespace\_names***—Specify to drop logs by namespaces. *namespace\_names* must be a regex string with selected namespace names inside double quotes.
- **logging worker drop-pod-logs *pod\_names***—Specify to drop logs by pods. *pod\_names* must be a regex string with selected pod names inside double quotes.

- **logging worker exclude-logs-with-annotation true**—Specify to exclude logs from selected pods using annotation.



**Note** After adding or removing annotation from any pod, it is required to restart the fluent-worker pod for the changes to take effect.

- **logging worker keep-pod-logs *pod\_names***—Specify to retain logs by pods. *pod\_names* must be a regex string with selected pod names inside double quotes.
- **logging worker keep-namespace-logs *namespace\_names***—Specify to retain logs by namespaces. *namespace\_names* must be a regex string with selected namespace names inside double quotes.
- **logging worker drop-os-service-logs *service\_names***—Specify to drop logs from selected OS services. The currently supported values for *services\_names* are audit, kernel, or kubelet.
- **logging worker remove-keys [ *keys* ]**—Specify to remove keys from log entries. The log entry keys to be dropped are case sensitive.

## Viewing the Logs in Loki

You must enable the Loki (Grafana) to view all the logs the CEE Ops Center was listening.

Use the following configuration to enable the Loki:

```
configure
 logging
 loki enable
 retention-period retention_period_in_hours
 commit
```

### NOTES:

- **loki enable** - Enables Loki to view to the logs.
- **retention-period***retention\_period\_in\_hours* - Specifies the retention period of the logs in hours.

## Verifying Log Forwarding

You can verify the external collectors for the log entries received. The logs are specific to the external collector. For example, you can use Kibana to verify the entries in ELK stack.

## Troubleshooting

This section provides information on the common issues encountered while enabling log forwarding.

To resolve the issues related to Log Forwarding, verify if:

- The configured endpoint IP/FQDN and port number are correct.
- The external endpoint is reachable from the all Kubernetes nodes (both control plane and worker).

- The Forward protocol plugin is enabled at the endpoint.
- The logs are generated from any of the nodes.
- The external endpoint is configured to dump the logs into a file for verifying the incoming entries.

## Log Rate Limiting

This section describes the basic principle used in rate limiting log messages in SMI Logging functionality.

### Rate Limiting Log Messages

The SMI uses the *systemd-journald* service—a Linux system service for collecting and storing log data—for storing Kubernetes system and pods level log messages to files on the disk. You can configure Rate Limiting to reduce the number of messages logged. Also, Rate Limiting discards some log messages while limiting others. You can apply Rate Limiting to all the messages in the system based on the service so that logs from the services do not interfere with each others limit.

You can configure Rate Limit by defining the *RateLimitIntervalSec* and *RateLimitBurst* parameter in */etc/systemd/journald.conf* file. If the messages exceed the specified value defined in the *RateLimitBurst* parameter within the specified interval defined in the *RateLimitIntervalSec* parameter, the log messages are dropped until the interval period is over.

In the following example, the log messages are dropped, if it exceeds 10000 messages within an interval of one second.

```
RateLimitIntervalSec=1s
RateLimitBurst=10000
```

The disk usage reserved for journal log affects the *RateLimitBurst* parameter. The value defined in the *RateLimitBurst* parameter is multiplied by a factor based on the disk usage reserved for the journal logs. More messages are dropped within interval when less disk space is available.

You can run the following command to find out if the log messages are dropped:

```
sudo systemctl status systemd-journald
```




---

**Note** Using this command, you can verify the number of suppressed messages as well.

---

The following example shows the number of suppressed messages from the *docker.service*:

```
Sep 02 21:09:58 tb15-ultram-cnat-cnat-core-protocol-data1 systemd-journald[3791]:
Suppressed 12229 messages from docker.service
```

## Gather TAC

Gather TAC is the primary mechanism through which the application debug files are extracted from a cluster. Whenever a debug package is required, a user can trigger the Gather TAC through the CLI or API. The user can specify a start and end time to download the index files (for that specific time period) of all the artifacts from the system. The user can extract the following data using Gather TAC:

- A tar ball of system and K8s pod logs.

- A tar ball of all bulk statistics produced within the specified time period.
- A tar ball of the current configuration, last 100 commits, and all audit information.
- A tar ball of the Prometheus data covering the time period.
- A list of all core files covering the time period

## Debugging Data in CEE

Using the Gather TAC function, you can collect logs from the coredump, Kubernetes, Kernel, Kubelet, and container logs. The collected files are compressed and stored in an internal Apache server.

### Debugging Data

- The following commands are used for requesting the TAC debug information.

`tac debug pkg create and delete`

#### New command

```
tac-debug-pkg create last<time_to_now>
tac-debug-pkg delete last<time_to_now>
```

#### Old command

```
tac-debug-pkg create {from start_time | to end_time } {logs-filter namespace
namespace | pod_name pod_name
{cores-filter { process process_name }} {{ cfg | cores | logs | metrics
| stats } {false | true}}
tac-debug-pkg delete tac-id tac_id
```

Previously, the command syntax required a user to specify a time period by entering *from* and *to* criteria.

The *new* syntax for **tac-debug-pkg create** and **tac-debug-pkg delete** commands now allows users to specify the duration relative to the current time using the last keyword:

```
tac-debug-pkg create last<time_to_now>
tac-debug-pkg delete last<time_to_now>
```

<time\_to\_now> specifies the time to now in terms of the number of:

**Days** - Expressed as "D", "d", or "day"; for example "5D"

**Hours** - Expressed as "H", "h", or "hour"; for example "3h"

**Minutes** - Expressed as "M", "m", "min", or "minute"; for example "18minute"

**Seconds** - Expressed as "S", "s", "sec", or "second"; for example "3600sec"

Additionally, omitting the *to* keyword from the *from* parameter instructs the system to collect the TAC package from the specified time until *now*:

```
tac-debug-pkg create from <time_to_now>
```

The *from* keyword no longer requires the use of the *to* keyword if you are creating the TAC package from a specific time until now.

Table 6: tac-debug-pkg usage examples

| User Intention                                             | Command                                              |
|------------------------------------------------------------|------------------------------------------------------|
| collect tac-debug-package for last 50 seconds              | <b>tac-debug-pkg create last 50s</b>                 |
| collect tac-debug-package for last 10 minutes              | <b>tac-debug-pkg create last 10min</b>               |
| collect tac-debug-package for last 3 hours                 | <b>tac-debug-pkg create last 3H</b>                  |
| collect tac-debug-package for last 7 days                  | <b>tac-debug-pkg create last 7day</b>                |
| delete all collected tac-debug-package for the past 2 days | <b>tac-debug-pkg delete last 2D</b>                  |
| collect tac-debug-package from 2019-08-09_01:00:00 to now  | <b>tac-debug-pkg create from 2019-08-09_01:00:00</b> |

Other tac debug pkg commands

```
tac-debug-pkg merge tac-id tac_id
tac-debug-pkg status
tac-debug-pkg list
```

- Access the Apache server through the Ops Center.

URL: `https://show-tac-manager.smi-show-tac.{IP address}.<domain_name>`

- A directory is created based on the *tac-id*: `/tac/[tac-id]/`
- A manifest file is created for each of the *tac-debug-pkg* to the store metadata. A sample *manifest.json* file is shown below:

```
{
 tac-id:"1554868784",
 from:"2019-04-08_00:00:00",
 to:"2019-04-10_00:00:00",
 cores:[{node:"node-01",
file:"/cores/node-01/core.test.0.2f4afbe0dc494e879d3f42429fed1c38.20130.1554770483000000.xz"},
 {node:"node-01",
file:"/cores/node-01/core.test.0.2f4afbe0dc494e879d3f42429fed1c38.18448.1554770577000000.xz"}],
 config:[{node:"node-01",
file:"/tac/1554868784/config/<ipv4address>_configuration.tar.gz.base64"}],
 stats:[{node:"node-01",
file:"/tac/1554868784/stats/Stats_2019-04-08_00-00-00_2019-04-10_00-00-00.tar.gz"}],
 logs:[{node:"node-01",
file:"/tac/1554868784/logs/Logs_2019-04-10_04-00-17.tar.gz"}],
 metrics:[{node:"node-01",
file:"/tac/1554868784/metrics/Metrics_2019-04-08_00-00-00_2019-04-10_00-00-00.tar.gz"}]}
}
```



**Important** Authentication to the Apache server is enabled by default.

- The following services collect and retrieve logs, data chunks, and bulk statistics.
  - **Core retrieving service** - This service retrieves the list of coredump based on the time duration. The systemd coredump service compresses the core files. The configuration parameters in the core files determines the name of the core file. Due to the core file large size, they are not copied on the disk. You can access it through the proxy from its original location.

```
file location:
./cores/{node name}/core.xxx...
Sample file location on server:

cores/node-1/core.test.0.99775297099c489ea08052d533206b66.10213.1554504010000000.xz
```

- **Logs retrieving service** - This service collects Kernel, System, Containers level logs using JournalD service. In return, the sender receives a tar file which contains logs files based on the time duration. The files are created with following naming convention:

```
./tmp/logs/{random string}/{namespace}/{pod}/{container.log}
```

A sample file (Tar) format with the timestamp embedded in the file name is shown below:

```
./tmp/logs/{random string/Logs_{yyyy-mm-dd_hh-mm-ss}.tar.gz
```

- **Prometheus data retrieving service** - This service retrieves snapshot of data chunks saved by the Prometheus service. You can specify the time duration for saving a snapshot. A sample file and directory structure for the data snapshot is shown below:

```
directory: data/snapshots/20190405T175611Z-7ee562389bd9ab66/01D7N0QVNBXRF5MRVFQB5MQQCW

files:
./chunk/0001
./index
./meta.json
./tombstones
```

- **Bulk statistics retrieving service** - This service retrieves statistics saved by the Prometheus service. You can specify the time duration for saving the statistics. A tar file is stored onto the Apache server for review. A sample file location on the server is shown below:

```
tac/0123456789/stats/Stats_2019-4-04_00-00-00_2019-04-04_18-00-00.tar.gz
```

The following example collect logs for pods in cdl-global namespace for CDL application.

#### Example:

```
cee# tac-debug-pkg create from 2019-12-18_00:00:00 to 2019-12-18_20:00:00 logs-filter {
namespace cdl-global }
response : Tue Dec 18 18:40:55 UTC 2019 tac-debug pkg ID : 157660805
```

# Log Monitoring

For real time monitoring of application logs, the CEE Ops Center uses the Kubectl utility. The Kubectl utility allows:

- Tailing multiple pods in a single stream.
- Tailing all containers within the Pods.
- Using regular expression to match or find Pod names.
- Color coding the output of each pod.

To monitor applications logs using the Kubectl utility in the CEE Ops Center, use the following command:

```
cluster logs kubetail_options
```

### Example:

```
my-pod-v1
my-pod-v1 -c my-container
my-pod-v1 -t intl-context -c my-container
'(service|consumer|thing)' -e regex
-l service=my-service
--selector service=my-service --since 10m
--tail 1
```

### NOTES:

- **cluster logs** - Tails a set of pods.
- *kubetail\_options* - Specifies the following options to tail Kubernetes pods:
  - *[-h]*, *--help* - Displays the help text.
  - *[-c]*, *--container* - Specifies the name of the container to tail in the pod. You can use this option multiple times. By default, this option specifies all the containers in the pod.
  - *[-n]*, *--namespace* - Specifies the Kubernetes namespace where the pods are located.
  - *[-t]*, *--context* - Specifies the Kubernetes context. For example, *intl-context*. It relies on the *~/.kube/config* file for the context.
  - *[-l]*, *--selector* - Specifies the Label selector. You can ignore the pod name if this option is used.
  - *[-d]*, *--dry-run* - Prints the names of the matched pods and containers.
  - *[-p]*, *--previous* - Returns the logs for the previous instances of the pods, if the pods are available. Returns either *true* or *false*. Default value is *false*.
  - *[-f]*, *--follow* - Specifies whether the logs must be streamed. Returns either *true* or *false*. Default value is *true*.
  - *[-s]*, *--since* - Displays the logs that are newer than a relative duration. For example, 5 seconds, 2 minutes, or 3 hours. Default value is 10 seconds.
  - *[-b]*, *--line-buffered* - Specify this flag to use a line-buffered. Default value is *false*.
  - *[-e]*, *--regex* - Specifies a matching name to use (*regex* or *substring*).

- `[-j]`, `--jq` - Parse a *json* output using this option. For example, `--jq ".logger + \" \" + .message"`.
- `[-k]`, `--colored-ouput` - Displays a colored output. The options include:
  - `pod` - Display the name of the pod in color.
  - `line` - Display a entire line in color.
  - `false` - Displays the output without color.

The default value is *false*.

- `[-z]`, `--skip-colors` - Specifies the comma-separated list of colors which is not used in the output. If you have green foreground on black, this option will skips dark grey and green colors. For example, `-z 2,8,10`. Default value is 7,8.
  - `--timestamps` - Displays the timestamps for each log line.
  - `--tail` - Displays the lines of the recent log files. Default value is `-1`.
- `[-v]`, `--version` - Prints the Kubetail utility version.
- `[-r]`, `--cluster` - Specifies the name of the Kubeconfig cluster to use.
- `[-i]`, `--show-color-index` - Displays the color index before the pod name prefix shown before each log line. Normally only the pod name is added as a prefix before each line, for example `[app-5b7ff6cbcd-bjv8n]`. If this option is selected, then the color index is added as well: `[1:app-5b7ff6cbcd-bjv8n]`. This is useful if you have color blindness or if you want to know which colors to exclude (see "`--skip-colors`"). Default value is *false*.

### Init Logs

The cluster deployer supports debug logging in **kubeadm init** for errors such as cluster synchronization failure due to misconfiguration.

A debug message captures the error logs in `/var/tmp/kubeadm_out.log` to access the setup and retrieve the kubeadm init logs. You can view the error messages during cluster sync.

## Cluster Monitoring

The monitoring module in CEE monitors the local and remote clusters. The monitoring module is based on Prometheus, Thanos and Node Exporter open source projects. It provides an overall centralized metrics view for the entire cluster. Prometheus is configured to scrape the local Kubernetes resources and the Node Exporter. The Node Exporter provides all the system level information, while Thanos collects these metrics and export them to Grafana. You can visualize the monitored metrics using Grafana.




---

**Note** The SMI Cluster Manager acts as a Central Monitoring System.

---

Also, you can configure the Thanos to collect the metrics from the remote cluster. It is possible to configure any number of remote clusters using the CLI. The connections to the remote cluster is secured through Transport Layer Security (TLS) protocol. For monitoring the clusters, the local Thanos acts as client and the remote

cluster act as server. Therefore, you must configure the local system with client certificates, and remote clusters with server certificates.

## Configuring the Remote Cluster

You can configure the Cluster Manager to monitor the remote clusters for alerts. The remote clusters act as a server.

To configure the Cluster Manager for monitoring the remote clusters, use the following configuration:

```
configure
prometheus query-mode server
 prometheus server-settings external-ip external_vip_ip
 server-settings ssl-key ssl_eky
 server-settings ssl-crt ssl_crt
 server-settings ssl-ca ssl_ca
```

### NOTES:

- **prometheus query-mode server**—Configure the Cluster Manager to monitor the remote clusters for alerts.
- **prometheus server-settings external-ip external\_vip\_ip**—Configure the server settings for the specified remote cluster.
- **server-settings ssl-key ssl\_eky**—Specify the SSL key.
- **server-settings ssl-crt ssl\_crt**—Specify the SSL certificate.
- **server-settings ssl-ca ssl\_ca**—Specify the SSL certificate authority.

## Configuring the Cluster Manager to Collect the Metrics from Remote Clusters

You can configure the Cluster Manager to monitor the remote clusters. The Cluster Manager acts as a client in this scenario.

To configure the Cluster Manager to collect the alerts from remote clusters:

1. Configure the remote cluster.

```
configure
prometheus query-mode client
 federation remote-cluster-certs remote_cluster_IP
```

### Example:

```
cee# config terminal
cee(config)# prometheus query-mode client federation remote-cluster-certs 10.84.114.218

name bxbpod

SSL multiline raw certificates
ssl-key "ssl-key"
ssl-crt "ssl-crt"
ssl-ca "ssl-ca"
```

2. Add all the remote clusters to the federation.

```

configure
 prometheus federation subordinates remote_cluster_IPs

```

**NOTES:**

- **prometheus query-mode client** - Configures the Cluster Manager to monitor the remote clusters.
- **federation remote-cluster-certs** *remote\_cluster\_IP* - Configures the specifies remote cluster with SSL certificates.
- **prometheus federation subordinates***remote\_cluster\_IPs* - Add all the remote clusters to the federation.

## Cluster Alerting

In CEE, the Alerting module is responsible for gathering the alerts from local and remote clusters. The Alerting module is based on the [Prometheus](#) and [Alert Manager](#) Open Source projects. It provides a centralized alerts view of the entire cluster. You can visualize the alerts using Grafana. The Prometheus scrapes the local cluster metrics.

You can configure the Prometheus with the alert rules to generate the alerts when the specified alert criteria is met. Also, you can configure Prometheus and Alert Manager to process the alerts by other modules like Alert logger, Alert router, and SNMP Trapper etc. The Alert Manager is configured with multiple webhooks to hand over to the alerts to these modules.

## CIMC Alerts Exporter

It is possible to configure a cluster with a cluster of CIMC devices. This enables the CIMC devices to receive alerts from the configured CIMC cluster. The CIMC exporter periodically polls the configured CIMC clusters and exports the received alerts through Prometheus.

## Configuring the Remote Cluster

You can configure the Cluster Manager to monitor the remote clusters for alerts. The remote clusters act as a server.

To configure the Cluster Manager for monitoring the remote clusters, use the following configuration:

```

configure
 prometheus query-mode server
 prometheus server-settings external-ip external_vip_ip
 server-settings ssl-key ssl_eky
 server-settings ssl-crt ssl_crt
 server-settings ssl-ca ssl_ca

```

**NOTES:**

- **prometheus query-mode server**—Configure the Cluster Manager to monitor the remote clusters for alerts.
- **prometheus server-settings external-ip** *external\_vip\_ip*—Configure the server settings for the specified remote cluster.
- **server-settings ssl-key** *ssl\_key*—Specify the SSL key.

- **server-settings ssl-crt *ssl\_cert***—Specify the SSL certificate.
- **server-settings ssl-ca *ssl\_ca***—Specify the SSL certificate authority.

## Configuring the Cluster Manager to Collect the Alerts from Remote Clusters

You can configure the Cluster Manager to monitor the remote clusters. The Cluster Manager acts as a client in this scenario.

To configure the Cluster Manager to collect the alerts from remote clusters:

1. Configure the remote cluster.

```
configure
 prometheus query-mode client
 federation remote-cluster-certs remote_cluster_ip
```

### Example:

```
cee# config terminal
cee(config)# prometheus query-mode client federation remote-cluster-certs 10.84.114.218

name bxbpod

SSL multiline raw certificates
ssl-key "ssl-key"
ssl-crt "ssl-crt"
ssl-ca "ssl-ca"
```

2. Configure the alert port to receive the alerts:

```
configure
 prometheus federation remote-cluster-certs alert-rx-port port_number
```

### Example:

```
cee# config terminal
cee(config)# prometheus federation remote-cluster-certs alert-rx-port 8701
```

3. Add all the remote clusters to the federation.

```
configure
 prometheus federation subordinates remote_cluster_ip
```

### NOTES:

- **prometheus query-mode client**—Configure the Cluster Manager to monitor the remote clusters.
- **federation remote-cluster-certs *remote\_cluster\_ip***—Configure the specifies remote cluster with SSL certificates.
- **prometheus federation remote-cluster-certs alert-rx-port *port\_number***—Configure the alert port to receive the alerts.
- **prometheus federation subordinates *remote\_cluster\_ip***—Add all the remote clusters to the federation.

## Configuring CIMC

Use the following configuration to configure the CIMC cluster:

```

configure
 cimc enabled
 cluster cluster_name
 default username username
 default password password
 server IPv4address
 name cimc_server_name
 server IPv4address
 name alert_name
 username username
 password password

```

## NOTES:

- **cimc enabled** - Enables the CIMC cluster.
- **cluster** *cluster\_name* - Specifies the CIMC cluster name.
- **default username** *username* - Specifies the default user name of the CIMC cluster.
- **default password** *password* - Specifies the default password of the CIMC cluster.
- **server** *IPv4address* - Specifies the CIMC server's IPv4 address.
- **name** *cimc\_server\_name* - Specifies the CIMC server name.
- **name** *alert\_name* - Specifies the alert name.
- **username** *username* - Specifies the user name for authentication.
- **password** *password* - Specifies the password for authentication.

To view the active alerts use the following command:

```

cee# show alerts active summary
NAME UID SEVERITY STARTS AT SOURCE SUMMARY

k8s-pod-not-ready 0239ce185c88 critical 07-10T18:58:59 makoruko-aio-control-plane
Pod / has been in a non-ready state for longer than 1 minute.
k8s-deployment-replic d048b003fce0 critical 07-10T17:23:29 makoruko-aio-control-plane
Deployment cdl/documentation has not matched the expected number of replicas for longer
than 2 minutes.
k8s-pod-not-ready 93b83787d3b9 critical 07-10T17:22:29 makoruko-aio-control-plane
Pod / has been in a non-ready state for longer than 1 minute.
k8s-pod-not-ready 1c9e6f3a4abd critical 07-10T17:22:29 makoruko-aio-control-plane
Pod / has been in a non-ready state for longer than 1 minute.
k8s-deployment-replic 3a170f244c17 critical 07-10T17:23:29 makoruko-aio-control-plane
Deployment cdl/api-cdl-ops-center has not matched the expected number of replicas for
longer than 2 minutes.
k8s-pod-not-ready 9859a350e6bc critical 07-10T17:22:29 makoruko-aio-control-plane
Pod / has been in a non-ready state for longer than 1 minute.
k8s-deployment-replic 113f35cc5f71 critical 07-10T17:23:29 makoruko-aio-control-plane
Deployment smi/deployer-ui-smi-cluster-deployer-deployer-console has not matched the
expected number of repl...

```

```
k8s-pod-not-ready 9e623b582dc4 critical 07-10T17:22:29 System
Pod / has been in a non-ready state for longer than 1 minute.
```

## Configuring Email Notification for Alerts

You can configure the Ops Center to send the email notifications to a maximum of 10 recipients for the generated alerts. To configure email notifications for the alerts, use the following configuration:

```
configure
 smtp enabled
 smtp recipients recipient_name
 email email_id
exit
```

### NOTES:

- **smtp enabled** - Enables sending email notification for the generated alerts.
- **smtp recipients** *recipient\_name* - Specifies the name of the recipient.
- **email** *email\_id* - Specifies the email address of the recipient.

## UCS Server Status Alerts

### Feature Description

If the UCS server is powered down or non-accessible, an alert will be set up to report and notify the UCS server availability status.

The SMI metrics track and report faults on the UCS server. The **cimc\_server\_not\_reachable\_alert** metric tracks the availability status of the UCS server. To establish an HTTP connection during login, this metric is set to 1 or 0 based on success (response) or failure.

### Monitoring CIMC Reachability

To monitor CIMC reachability, log on to the CEE CLI Ops Center. You can enable CIMC, define a cluster, and add configuration for server IP and credentials using the commands in the [Configuring CIMC, on page 66](#) section.

When CIMC is not reachable, the value for the **cimc\_server\_not\_reachable\_alert** metric will be set to 1 and exposed for Prometheus. These values can be tracked in the Grafana dashboard.

After sometime, the **server-not-reachable-alert** alert will be created in CEE. If the CIMC becomes reachable, the exposed metric will be deleted from the Prometheus client to prevent it from firing any longer, and the alert will be moved to history.

# Node Problem Detector

Table 7: Feature History

| Feature Name                      | Release   | Feature Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Support for Node Problem Detector | 2025.02.1 | <p>This feature supports onboarding of Node Problem Detector (NPD) through CEE Ops center.</p> <p>The NPD enhances hardware resiliency by detecting and addressing issues at the OS, hardware, and platform levels. It prevents the impact of node problems on Kubernetes pods, thereby maintaining application performance and reliability.</p> <p>Note that the NPD complements existing metrics, alerting, and log forwarding solutions.</p> <p>Command introduced:</p> <p><b>node-problem-detector agent enabled</b></p> |

The Node Problem Detector (NPD) is a CNDP monitoring add-on designed to enhance cluster resilience by identifying and addressing node-level issues across operating system (OS), hardware, and platform layers. Its primary objective is to detect potential problems early, preventing them from affecting Kubernetes pods and ensuring optimal application performance and reliability.

## Core capabilities of NPD

The key capabilities of NPD are:

- **Detection of node issues:** With proper configuration, it is possible to identify and report problems that may impact node performance, including:
  - **Kernel issues:** Such as kernel deadlocks and corrupted file systems.
  - **Container runtime issues:** For example, unresponsive runtime daemons.
  - **Infrastructure daemon issues:** Such as failures in critical services like NTP.
  - **Hardware issues:** Detects faulty CPUs, memory, or disks.
- **Monitoring types:**
  - **Log-based monitoring:** Continuously scans system logs for predefined patterns of known issues.
  - **Custom-script-based monitoring:** Enables users to define and execute custom monitoring rules for unique use cases.

- **Exporter types:**
  - Prometheus exporter: Used for prometheus-formatted metrics
  - K8s exporter: Used for reporting k8s events/conditions to K8s api-server.

## Benefits of NPD

- **Proactive Issue Detection:** Quickly identifies issues at the node level, reducing the risk of application disruptions.
- **Enhanced Reliability:** Prevents node-level problems from affecting Kubernetes pods, ensuring consistent application performance.
- **Resource Efficiency:** Operates with minimal system overhead, enabling smooth cluster operations.
- **Extensibility:** Supports custom rules and scripts, allowing users to tailor monitoring to their specific needs.

## How NPD works

The Node Problem Detector follows a systematic process to monitor and manage node health:

1. **Log Parsing:** Monitors system logs to detect errors and patterns associated with node issues.
2. **Custom Rules:** Executes predefined scripts and rules for tailored issue detection.
3. **Issue Reporting:** Detects problems and reports them to the Kubernetes API server.
4. **Metrics Generation:** Exports metrics to monitoring tools such as Prometheus and Grafana.

## Enable the Node Problem Detector

To enable the Node Problem Detector in your Kubernetes environment, perform the following steps:

### Procedure

---

**Step 1** Execute the following CLI command from CEE to enable the NPD.

#### Example:

**node-problem-detector agent enabled**

This command enables NPD with default configurations and built-in monitoring rules for detecting well-known issues.

The following is an example configuration with the default rules that are loaded automatically when the NPD feature is enabled.

```
day0-kernel-monitor.json: |-
 {
 "plugin": "kmsg",
 "logPath": "/dev/kmsg",
 "lookback": "5m",
```

```

"bufferSize": 10,
"source": "day0-kernel-monitor",
"metricsReporting": true,
"conditions": [],
"rules": [
 {
 "type": "temporary",
 "reason": "OOMKilling",
 "pattern": "Killed process \\d+ (.+) total-vm:\\d+kB, anon-rss:\\d+kB, file-rss:\\d+kB.*"
 },
 {
 "type": "temporary",
 "reason": "TaskHung",
 "pattern": "task [\\S]+:\\w+ blocked for more than \\w+ seconds\\."
 },
 {
 "type": "temporary",
 "reason": "UnregisterNetDevice",
 "pattern": "unregister_netdevice: waiting for \\w+ to become free. Usage count = \\d+"
 },
 {
 "type": "temporary",
 "reason": "KernelOops",
 "pattern": "BUG: unable to handle kernel NULL pointer dereference at .*"
 },
 {
 "type": "temporary",
 "reason": "KernelOops",
 "pattern": "divide error: 0000 \\[#\\d+\\] SMP"
 },
 {
 "type": "temporary",
 "reason": "Ext4Error",
 "pattern": "EXT4-fs error .*"
 },
 {
 "type": "temporary",
 "reason": "Ext4Warning",
 "pattern": "EXT4-fs warning .*"
 },
 {
 "type": "temporary",
 "reason": "IOError",
 "pattern": "Buffer I/O error .*"
 },
 {
 "type": "temporary",
 "reason": "MemoryReadError",
 "pattern": "CE memory read error .*"
 },
 {
 "type": "temporary",
 "reason": "ReadOnlyFileSystem",
 "pattern": "Remounting filesystem read-only"
 },
 {
 "type": "temporary",
 "reason": "CPUSoftLockup",
 "pattern": "BUG: soft lockup - CPU#\\d+ stuck for \\d+s!.*"
 },
 {
 "type": "temporary",
 "reason": "CPUHighTemperature",
 "pattern": "CPU\\d+: \\w+ temperature above threshold, cpu clock throttled \\(total events

```

```

= \\d+\\)"
 },
 {
 "type": "temporary",
 "reason": "IncorrectCalbleReconn",
 "pattern": "\\w+: \\([\\S]+\\): Enslaving as a backup interface with a down link"
 }
]
}
day0-systemd-service-monitor.json: |-
{
 "plugin": "journal",
 "pluginConfig": {
 "source": "systemd"
 },
 "logPath": "/var/log/journal",
 "lookback": "5m",
 "bufferSize": 10,
 "source": "day0-systemd-service-monitor",
 "metricsReporting": true,
 "conditions": [],
 "rules": [
 {
 "type": "temporary",
 "reason": "SystemdNetworkdStart",
 "pattern": "Starting Network Service..."
 }
]
}
day0-systemd-networkd-monitor.json: |-
{
 "plugin": "journal",
 "pluginConfig": {
 "source": "systemd-networkd"
 },
 "logPath": "/var/log/journal",
 "lookback": "5m",
 "bufferSize": 10,
 "source": "day0-systemd-networkd-monitor",
 "metricsReporting": true,
 "conditions": [],
 "rules": [
 {
 "type": "temporary",
 "reason": "BondInterfaceConfigured",
 "pattern": "bd\\d+: Configured"
 }
]
}
day0-ucs-monitor.json: |-
{
 "plugin": "journal",
 "pluginConfig": {
 "source": "otelcol"
 },
 "logPath": "/var/log/journal",
 "lookback": "5m",
 "bufferSize": 10,
 "source": "day0-ucs-monitor",
 "metricsReporting": true,
 "conditions": [],
 "rules": [
 {
 "type": "temporary",

```

```

 "reason": "CorrectableDIMMErrors",
 "pattern": ".*\\| read \\d+ correctable ECC errors on CPU\\d+ DIMM [A-Z]\\d+\\s+\\| Asserted"
 },
 {
 "type": "temporary",
 "reason": "RAIDCtrlReset",
 "pattern": ".*SLOT-HBA:Controller encountered a fatal error and was reset"
 },
 {
 "type": "temporary",
 "reason": "RAIDCtrlFIFOFull",
 "pattern": ".*I2C Controller \\d+ software FIFO is full.*"
 },
 {
 "type": "temporary",
 "reason": "RAIDCtrlI2cRWEError",
 "pattern": ".*Restarting storage as we hit an I2C read/write error = 0[xX][0-9a-fA-F]+"
 }
]
}
day0-clock-sync-monitor.json: |-
{
 "plugin": "custom",
 "pluginConfig": {
 "invoke_interval": "60s",
 "timeout": "10s"
 },
 "source": "day0-clock-sync-monitor",
 "metricsReporting": true,
 "conditions": [],
 "rules": [
 {
 "type": "temporary",
 "reason": "ClockOutOfSync",
 "path": "/npd/plugin/check_timedatectl_property.sh",
 "args": ["NTPSynchronized", "yes"],
 "timeout": "5s"
 }
]
}
}

```

**Step 2** *[Optional]* If additional monitoring rules are required, configure them using the CEE CLI. Custom rules can address specific issues unique to your environment.

The following is an example configuration for additional rules.

```

node-problem-detector agent enabled
node-problem-detector agent monitors docker-corrupt-overlay-monitor
journalctl source dockerd
journalctl log-path /var/log/journal
journalctl enable-report-metrics true
journalctl default-conditions CorruptDockerOverlay2
reason NoCorruptDockerOverlay2
message "docker overlay2 is functioning properly"
exit
journalctl rules a-rules
pattern "Error trying v2 registry: failed to register layer: rename /var/lib/docker/image/(.+)/var/lib/docker/image/(.+): directory not empty.*"
type temporary
reason CorruptDockerImage
exit
journalctl rules b-rules
pattern "returned error: readlink /var/lib/docker/overlay2.*: invalid argument.*"

```

```

 type permanent
 condition CorruptDockerOverlay2
 reason CorruptDockerOverlay2
 exit
exit
node-problem-detector agent monitors ntp-monitor
 custom invoke-interval 60s
 custom timeout 5s
 custom default-conditions NTPProblem
 reason NTPIsUp
 message "NTP service is up"
 exit
 custom rules rule1
 type permanent
 condition NTPProblem
 reason NTPIsDown
 path /npd/plugin/check_systemd_service.sh
 args [chrony.service]
 timeout 3s
 exit
exit
node-problem-detector agent monitors readonlyfs-monitor
 kmsg log-path /dev/kmsg
 kmsg look-back 5m
 kmsg default-conditions ReadonlyFilesystem
 reason FilesystemIsNotReadOnly
 message "Filesystem is not read-only"
 exit
 kmsg rules ruleee
 pattern "Remounting filesystem read-only"
 type permanent
 condition ReadonlyFilesystem
 reason FilesystemIsReadOnly
 exit
exit
node-problem-detector agent monitors syslog-bad-disk-monitor
 filelog timestamp-regex ^.{15}
 filelog message-regex "(?i)Currently unreadable.*sectors|(?)Offline uncorrectable sectors"
 filelog timestamp-format-regex "Jan _2 15:04:05"
 filelog log-path /var/log/messages
 filelog skip-list [" audit:" " audit["]
 filelog default-conditions DiskBadBlock
 reason DiskBadBlock
 message "Disk has no Bad Block"
 exit
 filelog rules rule1
 pattern "Currently unreadable.*sectors|Offline uncorrectable sectors"
 type permanent
 condition DiskBadBlock
 reason DiskBadBlock
 exit
exit

```

**Step 3** Confirm that NPD is running and actively monitoring nodes.

**Step 4** Use monitoring tools such as Prometheus or Grafana to visualize metrics generated by NPD.

## Metrics for node issues

When the node problems are detected, the SMI generates the following key metrics for monitoring:

| Metric Name                                                                                                                                                          | Description                                                                                         | Value Type |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------|
| problem_counter_total<br><b>Note</b><br>Temporary issue counts may increase, while permanent issues are either 0 or 1 (since permanent issues require a node reset). | Displays the total number of occurrences of a specific type of problem.                             | Counter    |
| problem_gauge                                                                                                                                                        | Indicates whether a specific type of problem is currently affecting the node (1 for yes, 0 for no). | Gauge      |
| service_status                                                                                                                                                       | Displays the status of an internal service (1 if the service is up, 0 if it is down).               | Gauge      |

These metrics provide valuable insights into the health and performance of nodes, enabling administrators to take corrective actions.

## Best practices for using NPD

To maximize the benefits of NPD, follow these best practices:

- **Customize for specific needs:** Add custom monitoring rules to address unique issues in your environment.
- **Integrate with monitoring tools:** Leverage tools such as Prometheus and Grafana to visualize and act on alerts.
- **Regularly update rules:** Keep detection rules updated to reflect changes in your infrastructure or application stack.
- **Monitor system overhead:** Ensure that NPD operates efficiently with minimal impact on cluster resources.

## Thanos ecosystem for metrix handling

The Thanos ecosystem is an open-source project that significantly enhances Prometheus's capabilities, transforming it into a highly available, globally queryable, and long-term metrics storage solution. While Prometheus excels at real-time monitoring within a single instance, it has limitations regarding horizontal scalability, long-term data retention, and providing a unified view across multiple Prometheus servers.

Thanos addresses these challenges by introducing several components that work together to extend Prometheus. Its core benefits include:

- **Global Query View:** Thanos allows users to query metrics from multiple Prometheus instances, across different clusters or regions, as if they were a single, unified data source. This simplifies monitoring distributed environments.
- **Long-Term Storage:** It enables cost-efficient storage of historical metric data in object storage (like S3, MinIO), overcoming Prometheus's inherent short-term retention limits.
- **High Availability:** Thanos provides mechanisms to ensure that metrics remain available even if a Prometheus instance or other component fails, often by integrating with redundant Prometheus setups and deduplicating data.

The Thanos ecosystem achieves this through various specialized components, such as the Sidecar, Receive, Querier, Store Gateway, Compactor, and Ruler, each designed to handle specific aspects of metric collection, storage, and querying.

## Architecture

This section describes the high-level architecture for the two deployment models namely, Thanos Sidecar and Thanos Receive with AWS S3 compatible Object Storage.

### Thanos Sidecar Deployment

The Thanos Sidecar architecture integrates directly with existing Prometheus deployments, extending their capabilities for long-term storage and global querying.

- **Deployment Model:**

- A Thanos Sidecar runs alongside each Prometheus server.
- Typically co-located within the same Kubernetes pod or on the same machine as its respective Prometheus instance.

- **Key Functions & Data Flow:**

- **Store API Exposure:** The Sidecar acts as an intermediary, exposing the Prometheus instance's live and historical Time Series Database (TSDB) data via the Thanos Store API.
- **Query Access:** This allows a central Thanos Querier to directly access metrics from that specific Prometheus.
- **Block Upload:** Periodically, the Sidecar reads completed TSDB blocks from Prometheus's data directory.
- **Long-Term Storage:** It uploads these blocks to a shared, centralized object storage bucket (e.g., S3).

- **Benefits & Use Cases:**

- Extends existing Prometheus deployments with minimal disruption.
- Enables Prometheus instances to maintain shorter local retention policies.
- Ensures data durability and global accessibility for long-term analysis.
- Primarily a pull-based mechanism for data collection and archival.

### Thanos Receive Deployment

The Thanos Receive architecture provides a scalable and highly available push-based metrics ingestion endpoint for Prometheus.

- **Deployment Model:**

- Thanos Receive operates as a standalone component, separate from individual Prometheus instances.
- Often deployed as a horizontally scalable cluster (e.g., a Kubernetes StatefulSet) for high availability and scalability.

- **Key Functions & Data Flow:**

- **Remote Write Endpoint:** It acts as an endpoint for the Prometheus Remote Write API.
- **Metrics Ingestion:** Prometheus instances are configured to push their metrics to the Thanos Receive cluster.
- **Local Storage:** Receive ingests these incoming metrics and stores them in its own internal TSDB.
- **Load Balancing/HA:** Receive instances often participate in a hashring to distribute incoming series across multiple replicas, ensuring high availability and scalability.
- **Store API Exposure:** It exposes the Thanos Store API, making the ingested data immediately queryable by a Thanos Querier.
- **Block Upload:** Similar to the Sidecar, Receive periodically uploads its completed TSDB blocks to a centralized object storage bucket for long-term retention.

- **Benefits & Use Cases:**

- Ideal for scenarios requiring multi-tenancy (isolating metrics from different sources).
- Suitable for environments where Prometheus instances cannot be directly accessed by a Sidecar (e.g., air-gapped networks or egress-only setups).
- Provides a push-based model for data transfer.
- Offers scalable and highly available real-time data ingestion.

## Components

This section describes the main components for the architecture of the two deployment models.

- **Thanos Sidecar:** This connects to Prometheus. It reads Prometheus data for queries. It also uploads data to S3-compatible cloud storage.
- **Thanos Receive:** This acts as a central ingestion point. It receives metrics from remote Prometheus instances using the remote-write protocol. It stores these metrics in S3-compatible object storage.
- **Thanos Store Gateway:** This provides access to historical metrics. It retrieves data stored in S3-compatible object storage. It also implements caching for improved performance.
- **Thanos Compact:** This optimizes long-term storage. It compacts small blocks of metrics into larger ones. It also creates 5-minute and 1-hour downsampled data.
- **Thanos Query:** This provides a global query interface. It aggregates data from multiple sources. It also deduplicates metrics across replicas.
- **Thanos Query Frontend:** This improves query performance. It splits large queries into smaller chunks. It caches query results and compresses responses.
- **Thanos Ruler:** This evaluates recording and alerting rules. It applies these rules against the global metrics view. It stores rule results in S3-compatible object storage.

# Push metrics data to an S3-compatible object storage using the Thanos ecosystem

Table 8: Feature History

| Feature Name               | Release Information | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Support for Thanos Receive | 2025.04.1           | Thanos Receive was introduced as an incremental feature to provide greater flexibility in metrics ingestion, specifically enabling a push-based approach via Prometheus's remote write API. It serves as an alternative to the Sidecar for scenarios where direct access to Prometheus instances for Sidecar deployment is not feasible, such as in multi-tenant environments or air-gapped networks where metrics must be actively pushed out. This component allows for horizontally scalable and highly available ingestion of real-time data, complementing the Sidecar's pull-based block storage and expanding Thanos's ability to handle diverse operational requirements. |

In this feature, the CEE provides you the option to send metrics data stored in Prometheus to a remote object storage, for example, Amazon Web Services (AWS) S3 or MinIO, by using Thanos.

This feature provides the following two deployment models:

- **Thanos Sidecar**

The Thanos Sidecar is deployed alongside each Prometheus instance in the same pod. Its primary functions are to implement Thanos's Store API, allowing Thanos Queriers to access Prometheus's data without directly interacting with its native APIs, and optionally to upload Prometheus's Time Series Database (TSDB) blocks to an object storage bucket (like S3 or MinIO) every two hours. This enables long-term storage of historical metrics and allows Prometheus instances to operate with shorter retention periods.

- **Thanos Receive**

The Thanos Receive component acts as a remote write endpoint for Prometheus, accepting metrics via the Prometheus Remote Write API. It stores these incoming metrics in its local TSDB and can also upload these blocks to object storage at regular intervals, similar to the Sidecar, contributing to long-term storage. Thanos Receive is particularly useful for multi-tenancy, as it can manage separate TSDB instances for different tenants, and for environments where metrics can only be pushed into Thanos, such as air-gapped setups. It also exposes the StoreAPI for real-time querying by Thanos Queriers.

## How it Works

### Thanos Sidecar

This section describes how to configure the Sidecar deployment with AWS S3 or MinIO.

### Prerequisites

- S3 bucket in AWS or MinIO



---

**Note** For more information about how to create an AWS S3 or MinIO bucket, refer to the original product documentation.

---

Integrating directly with existing Prometheus deployments, the Thanos Sidecar extends their capabilities for long-term storage and global querying.

1. **Prometheus Scrapes & Stores:** A Prometheus instance scrapes targets and stores metrics locally in its Time Series Database (TSDB).
2. **Sidecar Connects:** Runs alongside this Prometheus instance, connecting to its local storage.
3. **Real-time Querying:** Exposes Prometheus's live and historical data via the Thanos Store API, making it immediately queryable by a Thanos Querier.
4. **Block Upload:** Periodically (e.g., every two hours), the Sidecar uploads completed TSDB blocks from Prometheus's data directory to a configured object storage bucket for long-term retention.

### Thanos Receive

This section describes how to configure the Remote-write target including the Receiver URL and enable TLS support for the same using the CEE Ops-Center for the Receive deployment with AWS S3 or MinIO.

#### Prerequisites

- S3 bucket in AWS or MinIO
- Deploy Thanos Recieve



---

**Note** For more information about how to create an AWS S3 or MinIO bucket, refer to the original product documentation.

---

For scenarios requiring a push-based metrics ingestion model, Thanos Receive provides a scalable and highly available endpoint.

1. **Prometheus Remote Write:** Prometheus instances are configured to remote write their metrics to the Thanos Receive component.
2. **Receive Ingestion & Storage:** Ingests these incoming metrics, storing them in its own internal TSDB.
3. **Real-time Querying:** Exposes its ingested data via the Thanos Store API, allowing a Thanos Querier to access real-time metrics pushed to it.
4. **Block Upload:** Similar to the Sidecar, Thanos Receive periodically uploads its completed TSDB blocks to object storage, ensuring data durability and long-term storage for the pushed metrics.

### Thanos Store Gateway

Bridging the gap between object storage and query engines, the Thanos Store Gateway makes historical metrics readily accessible.

1. **Object Storage Connection:** Connects to the configured object storage bucket where Prometheus TSDB blocks (uploaded by Sidecars or Receive components) are stored.
2. **Index Creation/Loading:** It builds or loads indices for the blocks in object storage, allowing for efficient lookup of time series data.
3. **Store API Exposure:** Exposes the Thanos Store API, enabling the Thanos Querier to request and retrieve historical metric data from object storage.
4. **Data Retrieval:** Upon receiving a query, it fetches the relevant blocks from object storage and serves the requested data to the Querier.

### Thanos Compactor

Optimizing long-term storage and query performance, the Thanos Compactor manages and refines data blocks in object storage.

1. **Block Discovery:** The Compactor continuously scans the object storage bucket for new or existing TSDB blocks.
2. **Compaction:** It merges smaller, adjacent blocks into larger ones to reduce the number of files and improve query efficiency.
3. **Downsampling:** For older data, the Compactor creates downsampled versions (e.g., 5-minute or 1-hour averages) to further reduce storage footprint and speed up long-range queries.
4. **Retention Enforcement:** It applies retention policies, deleting blocks that have exceeded their configured retention period from object storage.

### Thanos Ruler

Extending Prometheus's rule evaluation capabilities to long-term and global data, the Thanos Ruler provides centralized alerting and recording.

1. **Querying Data:** The Ruler uses the Thanos Querier to fetch metric data from across the entire Thanos system (live Prometheus instances via Sidecars/Receive, and historical data via Store Gateways).
2. **Rule Evaluation:** It evaluates Prometheus-compatible recording rules and alerting rules against this aggregated data.
3. **Recording Rules Output:** For recording rules, it writes the resulting new time series back into the Thanos system, often via a Thanos Receive component, for long-term storage and further querying.
4. **Alerting:** For alerting rules, if conditions are met, it sends alerts to an Alertmanager instance, just like a standard Prometheus server.

## Configure Thanos components

You configure Thanos components and related settings using ConfD CLI commands. All YANG configuration parameters are documented in a tabular format.

## Configure Object Storage

All Thanos components that need object storage reference a shared configuration for S3-compatible storage. This is the recommended method for configuring object storage for all Thanos components.

### Procedure

---

**Step 1** Enter the configuration mode.

**Example:**

```
[user/global] cee# config
```

**Step 2** Create an object store configuration name.

**Example:**

```
[[user/global] cee(config)# thanos object-stores object-store my-store
```

**Step 3** Set the S3 bucket name.

**Example:**

```
[[user/global] cee(config)# bucket name "thanos-metrics"
```

**Step 4** Set the S3 endpoint URL (e.g., for MinIO).

**Example:**

```
[[user/global] cee(config)# bucket endpoint "minio.example.com:9000" // Example for MinIO endpoint
```

**Step 5** Specify the name of the secret containing S3 access keys.

**Example:**

```
[[user/global] cee(config)# bucket credentials-secret-name "s3-credentials"
```

**Step 6** Disable insecure connections (set to **false** for production)

**Example:**

```
[[user/global] cee(config)# bucket insecure false
```

**Step 7** Specify the name of the CA certificate secret name.

**Example:**

```
[[user/global] cee(config)# bucket ca-cert-secret-name minio-ca
```

---

## Manage Secret

This section provides tasks for configuring essential secrets that Thanos components and other services use. These secrets include S3 credentials, CA certificates, and basic authentication details.

The system automatically creates Kubernetes secrets from ConfD parameters.

## Procedure

---

**Step 1** Configure S3 credentials for object storage.

**Example:**

```
secrets generic minio-cred namespaces cee-global
data access-key value 8C2kQeVHdaR52lANxAp7iSq1C9QWWIwvkxsJVdnlosbs=
exit
data secret-key value 8zwsuvwFjJVTPOF5KgqcE3yrdvp3l6WsQkzbuVBizDg=
exit
```

**Step 2** Configure a CA certificate.

**Example:**

```
secrets ca-cert minio-ca
certificate "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----\n"
namespaces cee-global
exit
```

**Step 3** Configure basic authentication credentials.

**Example:**

```
secrets generic minio-cred
data username value $8$49gsgkNZKyp1OxY924NDDEGseHeQyt9pKvPO/mNmPsm=
exit
data password value 8oSnwQvzCD1zv5iBduVTF/eICHTXEAauZ+waPOz3igCE=
exit
```

---

## Configure Thanos Receive

This component receives metrics and uploads them to S3-compatible object storage.

### Procedure

---

**Step 1** Enable Thanos Receive.

**Example:**

```
[[user/global] cee(config)# thanos receive enable true
```

By default it is set to *false*.

**Step 2** Set the volume size for data storage.

**Example:**

```
[[user/global] cee(config)# thanos receive volume 100
```

By default it is set to *100GB*.

**Step 3** Reference an object store configuration.

**Example:**

```
[[user/global] cee(config)# thanos receive object-store-ref my-store
```

**Step 4** Set memory limits

**Example:**

```
[[user/global] cee(config)# thanos receive limits memory 32Gi
```

Valid memory values are 16Gi , 32Gi , 64Gi , and 128Gi. By default it is set to 16Gi.

**Step 5** Configure Prometheus to send its metrics to Thanos Receive using **remote-write** configuration.

For more information on **remote-write** configuration, refer to [Remote Write Configuration](#).

---

## Configure Thanos Store Gateway

This component provides access to historical metrics stored in S3-compatible object storage.

### Procedure

---

**Step 1** Enable Thanos Store.

**Example:**

```
[[user/global] cee(config)# thanos store enable true
```

By default it is set to *false*.

**Step 2** Set the cache volume size.

**Example:**

```
[[user/global] cee(config)# thanos store volume 200
```

By default it is set to *100*.

**Step 3** Reference an object store configuration.

**Example:**

```
[[user/global] cee(config)# thanos store object-store-ref my-store
```

**Step 4** Set memory limits.

**Example:**

```
[[user/global] cee(config)# thanos store limits memory 32G
```

Valid memory values are 16Gi , 32Gi , 64Gi , and 128Gi. By default it is set to 16Gi.

---

## Configure Thanos Compact

This component compacts and downsamples metrics in S3-compatible object storage.

## Procedure

---

**Step 1** Enable Thanos Compact.

**Example:**

```
[[user/global] cee(config)# thanos compact enable true
```

By default it is set to *false*.

**Step 2** Set the working volume size.

**Example:**

```
[[user/global] cee(config)# thanos compact volume 200
```

By default it is set to *100*.

**Step 3** Configure retention periods for raw, 5-minute, and 1-hour downsampled data

**Example:**

```
[[user/global] cee(config)# thanos compact retention-resolution-raw 60d
[[user/global] cee(config)# thanos compact retention-resolution-5m 180d
[[user/global] cee(config)# thanos compact retention-resolution-1h 2y
```

By default

- **retention-resolution-raw** is set to *30d*
- **retention-resolution-5m** is set to *90d*
- **retention-resolution-1h** is set to *1y*

**Step 4** Disable downsampling process.

**Example:**

```
[[user/global] cee(config)# thanos compact downsampling-disable false
```

By default it is set to *false*.

**Step 5** Set compaction and downsampling concurrency.

**Example:**

```
[[user/global] cee(config)# thanos compact compact-concurrency 2
[[user/global] cee(config)# thanos compact downsample-concurrency 2
```

By default it is set to *1*.

**Step 6** Set memory limits.

**Example:**

```
[[user/global] cee(config)# thanos compact limits memory 32Gi
```

Valid memory values are 16Gi , 32Gi , 64Gi , and 128Gi. By default it is set to 16Gi.

---

## Configure Thanos Query

This component aggregates data from multiple sources.

### Procedure

---

Set memory limits.

#### Example:

```
[[user/global] cee(config)# thanos query limits memory 32Gi.
```

Valid memory values are 16Gi , 32Gi , 64Gi , and 128Gi. By default it is set to 16Gi.

---

## Configure Thanos Query Frontend

This component optimizes and caches queries

### Procedure

---

Set memory limits.

#### Example:

```
[[user/global] cee(config)# thanos query-frontend limits memory 32Gi
```

Valid memory values are 16Gi , 32Gi , 64Gi , and 128Gi. By default it is set to 16Gi.

---

## Configure Thanos Ruler

This component evaluates recording and alerting rules. It stores rule results in S3-compatible object storage.

### Procedure

---

**Step 1** Enable Thanos Ruler.

#### Example:

```
[[user/global] cee(config)# thanos ruler enable true
```

By default it is set to *false*.

**Step 2** Set the storage volume size for rule evaluation.

#### Example:

```
[[user/global] cee(config)# thanos ruler volume 150
```

By default it is set to *100*.

**Step 3** Configure the rule evaluation interval.

**Example:**

```
[[user/global] cee(config)# thanos ruler evaluation-interval 30s
```

By default it is set to *1m*.

**Step 4** Reference an object store configuration.

**Example:**

```
[[user/global] cee(config)# thanos ruler object-store-ref my-store
```

**Step 5** Set memory limits

**Example:**

```
[[user/global] cee(config)# thanos ruler limits memory 32Gi
```

Valid memory values are 16Gi , 32Gi , 64Gi , and 128Gi. By default it is set to 16Gi.

## Sending alerts by Prometheus to remote Alert Manager

CNDP supports sending alerts by Prometheus to remote Alert Manager. This involves setting up Prometheus to dispatch alerts to external AlertManager instances and controlling the external accessibility of local AlertManager deployments for both inbound alert reception and outbound monitoring.

### Configure Remote AlertManager

Prometheus can send alerts to remote AlertManagers.

#### Procedure

**Step 1** Create a remote **AlertManager** configuration.

**Example:**

```
[[user/global] cee(config)# prometheus remote-alert-managers name
```

```
[[user/global] cee(config)# prometheus remote-alert-managers production-cluster
```

**Step 2** Enable the configuration.

**Example:**

```
[[user/global] cee(config)# enable true
```

By default it is set to *true*.

**Step 3** Provide the URL path prefix of the remote AlertManager.

**Example:**

```
[[user/global] cee(config)# path-prefix
```

**Step 4** Provide the endpoint URL of the remote AlertManager.

**Example:**

```
[[user/global] cee(config)# endpoint https://alertmanager.prod.example.com:9093
```

**Step 5** Configure the timeout.

**Example:**

```
[[user/global] cee(config)# timeout 30s
```

**Step 6** Configure maximum number of retries whens ending alerts.

**Example:**

```
[[user/global] cee(config)# max-retries 2
```

By default it is set to 3.

**Step 7** Configure minimum and maximum backoff time between retries.

**Example:**

```
[[user/global] cee(config)# min-backoff 1s
[[user/global] cee(config)# max-backoff 2m
```

By default minimum backoff time is set to 1s and maximum backoff time is set to 5m.

**Step 8** Enable retry on HTTP 429.

**Example:**

```
[[user/global] cee(config)# retry-on-http-429 true
```

By default it is set to *true*.

**Step 9** Configure TLS settings and basic authentication.

**Example:**

```
[[user/global] cee(config)# tls-config ca-cert-secret-name alertmanager-ca
[[user/global] cee(config)# tls-config insecure-skip-verify false
[[user/global] cee(config)# basic-auth-secret-name alertmanager-auth
[[user/global] cee(config)# exit
```

---

## Configure Object Storage for Thanos Sidecar

Configure Object Storage that can be used by thanos sidecar to push metrics data periodically.

### Procedure

---

**Step 1** (Optional) Set the S3 bucket name for the Thanos sidecar.

**Example:**

```
[[user/global] cee(config)# prometheus thanos-s3-object-store bucket metrics-bucket
```

**Step 2** Set the S3 endpoint URL for the Thanos sidecar (e.g., for MinIO).

**Example:**

```
[[user/global] cee(config)# prometheus thanos-s3-object-store endpoint minio.example.com:9000
```

**Step 3** (Recommended) Set the storage access credentials that contains the access key and secret-key.

**Example:**

```
[[user/global] cee(config)# prometheus thanos-s3-object-store credentials-secret-name 138-minio-cred
```

**Step 4** (Optional) Set the S3 access key for the Thanos sidecar.

**Example:**

```
[[user/global] cee(config)# prometheus thanos-s3-object-store access-key minioadmin
```

**Note**

It is recommended to use **credentials-secret-name** instead of S3 access key.

**Step 5** (Optional) Set the S3 secret key for the Thanos sidecar (use an encrypted value).

**Example:**

```
[[user/global] cee(config)# prometheus thanos-s3-object-store secret-key 8encrypted-secret-value
```

**Note**

It is recommended to use **credentials-secret-name** instead of S3 secret key.

**Step 6** (Optional) Disable insecure connections for the Thanos sidecar (set to **false** for production)

**Example:**

```
[[user/global] cee(config)# prometheus thanos-s3-object-store insecure false
```

**Step 7** (Optional) Use signature version 2 authentication.

**Example:**

```
[[user/global] cee(config)# prometheus thanos-s3-object-store signature-version2 false
```

**Step 8** (Optional) Specify the name of the secret containing the CA certificate for TLS verification.

**Example:**

```
[[user/global] cee(config)# prometheus thanos-s3-object-store ca-cert-secret-name minio-ca
```

## Configure AlertManager Ingress Exposure

Local AlertManager can be exposed externally via ingress to receive alerts from remote Prometheus instances.

### Procedure

Enable AlertManager ingress exposure.

This provides **alerts-in** for receiving external alerts and **alerts-out** for monitoring.

**Example:**

```
[[user/global] cee(config)# alertManagerIngress enable true
```

By default it is set to *false*.

# Sending Prometheus Server Metrics to Grafana Cloud

## Feature Description

The CEE leverages the existing remote-write feature to support the following functionalities:

- Push the Prometheus server metrics to Grafana Cloud
- Enable the following Prometheus parameters for CNDP Grafana Cloud integration:

- Remote Timeout

The **remote-timeout-seconds** command sets the timeout for requests to the remote write endpoint, in seconds. Default: 30 seconds.

- Queue Configuration

The **queue-config** command configures the queue used to write to remote storage.

- Relabel Configuration

The **relabel-configs** command defines a list of relabel configurations before the metrics are written to remote storage. The relabeling feature in Prometheus rewrites the label set of a target dynamically.

## Remote Write Configuration

### Configuring Remote Write to Push Prometheus Metrics

To push the Prometheus metrics to Grafana Cloud using remote-write, if you have configured TLS, use the following sample configuration:

```
prometheus remote-write target demo
 url https://prometheus-us-centrall1.grafana.net/api/prom/push
 basic-auth-secret-name remote-basic-auth
 tls-config skip-verify false
 tls-config ca-cert-secret-name ca-cert
exit
```

To push the Prometheus metrics to Grafana Cloud using remote-write, use the following sample configuration:

```
prometheus remote-write target demo
 url https://prometheus-us-centrall1.grafana.net/api/prom/push
 basic-auth username 725569
 basic-auth password 8ntCDRl2FkMD1m8mj9FohYwTuy/jo+7Cka0msfP2qW3Y=
 proxy-url http://proxy-wsa.esl.cisco.com:80
exit
```

### NOTES:

- **url**—Specify the target URL of Grafana Cloud.
- **basic-auth-secret-name** — Specify the name of the secret that contains user name and password.
- **tls-config skip-verify false** — Configure the TLS verification.
- **tls-config ca-cert-secret-name** — Specify the name of the CA Certificate.

- **basic-auth username** — Specify the username in Confd.
- **basic-auth password** — Specify the password in Confd. The password is encrypted in Confd and passed to the metrics helm chart.
- **proxy-url**—Specify the optional proxy URL to access Grafana Cloud in Confd.

### Configuring Prometheus Parameters

To configure the Prometheus parameters to Grafana Cloud using remote-write, use the following sample configuration:

- Remote Timeout—The **remote-timeout-seconds** command sets the timeout for requests to the remote write endpoint, in seconds. Default: 30 seconds.

The following is a sample configuration:

```
prometheus remote-write target demo
 remote-timeout-seconds 60
 exit
```

- Queue Configuration—The **queue-config** command configures the queue used to write to remote storage.

The following is a sample configuration:

```
prometheus remote-write target demo
 ...
 queue-config capacity 500
 queue-config max-shards 100
 queue-config min-shards 2
 queue-config max-samples-per-send 300
 queue-config batch-send-deadline-seconds 10
 exit
```

#### NOTES:

- **queue-config capacity**: Specify the number of samples to buffer per shard. Default: 2500.  
It is recommended to have adequate capacity in each shard to buffer several requests. The adequate capacity can maintain the throughput while processing occasional slow remote requests.
- **queue-config max-shards**: Specify the maximum number of shards. Default: 200.
- **queue-config min-shards**: Specify the minimum number of shards. Default: 1.
- **queue-config max-samples-per-send**: Specify the maximum number of samples per send. Default: 500.
- **queue-config batch-send-deadline-seconds**: Specify the maximum time in seconds that a sample will wait in buffer. Default: 5 seconds.
- Relabel Configuration—The **relabel-configs** command defines a list of relabel configurations before the metrics are written to remote storage. The relabeling feature in Prometheus rewrites the label set of a target dynamically.

The following is a sample configuration:

```
prometheus remote-write target demo
 ...
 relabel-configs test1
 target-label test1_label
 regex (.)?(.+)
 exit
```

```

replacement ${1}@${2}
action replace
source-labels container
source-labels pod
exit
exit

```

**NOTES:**

- **target-label:** Specify the label to which the resulting value is written in a replace action.
- **regex:** Specify the regular expression against which the extracted value is matched.  
Default = (.\*)
- **replacement:** Specify the replacement value against which a regex replace is performed if the regular expression matches.  
Default = \$1
- **action:** Specify the replace, keep, or drop action to perform based on regex matching.  
Default = replace
- **source-labels:** Specify the source label to select values from existing labels.
- Multiple relabeling steps can be configured per scrape configuration. The steps are applied to the label set of each target in order of appearance in the configuration file.
- Note that Prometheus will drop any label with empty value, hence use the labels with caution.

# K8s Certificates Auto-Renewal

## Certificate Management with Kubeadm

In kubeadm v1.21.0, client certificates generated by kubeadm expire after 1 year. The root certificates expires in 10 years. This feature enables monitoring and automatic renewal of kubeadm certificates before the expiry date from the CM or CEE. The CEE triggers an alert to notify the user of any certificate that is going to expire in 30 days.

The smi-cluster-maintainer pod monitors the k8s certificates and automate the renewal process, regardless of the cluster sync.

## How it Works

This section describes the sequence of operation for the feature.

1. The certificates in CM managed K8s clusters, control planes, workers, and external ETCD nodes is checked every 12 hours.
2. If any certificate is expiring in 60 days on the nodes, then the auto-renew process is triggered.
  - If the renewal is successful, then the following checks shows all the certificates as valid.

- If the renewal is unsuccessful, then the auto-renew process is re-initiated for the next cycle or iteration of validating the certificates.
3. If any certificate is expiring in 30 days on the nodes, then the auto-renew process is triggered along with sending an alert to the user.

In such cases, a manual intervention might be required to renew the certificates, which are nearing their expiry date.

The kubernetes certificate expiry alert is show below.

**Rules:**

- **Alert:** kube\_certificate\_expiring
  - **Annotations:**
    - **Type:** Kubernetes Certificate Expiring Alarm
    - **Summary:** "Kubernetes certificate {{ \$labels.cert\_path }} on host: {{ \$labels.node\_name }} is expiring in {{ \$labels.days\_to\_expiry }} days."
  - **Expression:**

```
|
 kube_certificate_expiring != 0
```
  - **Labels:**
    - **Severity:** critical




---

**Note** The certificate auto-renewal process must restart the api-server. You might experience a temporary k8s API downtime during the certificate auto-renewal process.

---

## OnDemand LDAP Connectivity Check

### Feature Summary and Revision History

#### Summary Data

|                                           |                                                                                      |
|-------------------------------------------|--------------------------------------------------------------------------------------|
| Applicable Product (s) or Functional Area | KVM-based application deployment support<br>K8s-based application deployment support |
| Applicable Platforms                      | Bare Metal, OpenStack, VMware                                                        |
| Feature Default Setting                   | Disabled – Configuration Required                                                    |

|                                 |                                                                                          |
|---------------------------------|------------------------------------------------------------------------------------------|
| Related Changes in this Release | Not Applicable                                                                           |
| Related Documentation           | <i>UCC CEE Configuration and Administration Guide</i><br><i>UCC SMI Operations Guide</i> |

## Revision History

| Revision Details  | Release   |
|-------------------|-----------|
| First introduced. | 2022.02.1 |

## Feature Description

The SMI Ops Center provides an external authentication using LDAP support. The LDAP configuration can be configured in the SMI Ops Center using CLI or the RESTCONF APIs.

This feature enables you to validate a new LDAP configuration before adding it to the system or an existing LDAP configuration.

## How it Works

This section describes how the feature works.

### How to Validate a New Configuration

The steps to validate a new LDAP configuration are as follows.

1. Login to the SMI Ops Center.
2. Provide the LDAP new configuration inputs to validate (see the following example ).

```
[pv/global] cee# smldap validate-security-config validate-new-security-config { ?
Possible completions:
base-dn LDAP Base DN
bind-dn LDAP Bind DN
group-attr Group attribute
group-mapping LDAP group to application security mapping
ldap-filter LDAP Filter - use %s to sub username
ldap-server-url LDAP Server URL (https://tools.ietf.org/html/rfc2255)
ldap-username-domain LDAP Username Domain
password Password
username Existing User name in LDAP server
```

3. Validate the LDAP new configuration (see the following example configuration).

```
cee(config)# smldap validate-security-config validate-new-security-config
{ base-dn dc=smi-lab,dc=com bind-dn cn=%s,ou=people,dc=smi-lab,dc=com group-attr
memberOf group-mapping { group admin ldap-group group1 } username user5 password
Passwd@123 ldap-filter cn=%s ldap-server-url ldap://209.165.200.224 }
Mon Jun 20 05:02:24.635 UTC+00:00
message accept "admin" external-user-group 1117 1117 /tmp
```

### How to Validate an Existing LDAP Configuration

Use the following example configuration to validate an existing LDAP configuration.

```
cee# smldap validate-security-config validate-current-security-config
Mon Jun 20 05:07:41.765 UTC+00:00
Value for 'username' (<string>): user5
Value for 'password' (<string>): *****
message accept "admin" external-user-group 1117 1117 /tmp
```



## PART I

# CEE Config Mode Command Reference

- [Alerts Operation Config Mode Command Reference, on page 97](#)
- [Bulk Statistics Config Mode Command Reference, on page 103](#)
- [CIMC Config Mode Command Reference, on page 109](#)
- [Cluster Exec Mode Command Reference, on page 113](#)
- [Debug Exec Mode Command Reference, on page 123](#)
- [Grafana Config Mode Command Reference, on page 127](#)
- [Logging Config Mode Command Reference, on page 129](#)
- [NPD Config Mode Command Reference, on page 137](#)
- [Prometheus Config Mode Command Reference, on page 141](#)
- [SNMP Config Mode Command Reference, on page 149](#)
- [VES Adapter Config Mode Command Reference, on page 153](#)





## CHAPTER 3

# Alerts Operation Config Mode Command Reference

---

- [alerts active, on page 97](#)
- [alerts add-silence, on page 98](#)
- [alerts add-silence matchers, on page 99](#)
- [alerts delete-silence, on page 100](#)
- [alerts history, on page 100](#)
- [alerts silence-by-id, on page 100](#)
- [alerts silences, on page 101](#)

## alerts active

Displays filtered list of active alerts.

---

### Command Modes

Exec > Global Configuration

---

### Syntax Description

```
active filter alerts_filter active { false | true } silenced { false | true }
inhibited { false | true } unprocessed { false | true } receiver regex
```

#### **active { false | true }**

Specify whether to display active alerts.

Must be one of the following:

- **false**
- **true**

Default Value: true.

#### **filter *alerts\_filter***

Specify the alerts filter in the format *label expr value or regex*.

Must be a string.

**inhibited { false | true}**

Specify whether to display inhibited alerts.

Must be one of the following:

- **false**
- **true**

Default Value: false.

**receiver *regex***

Specify a regex matching receivers to filter alerts.

Must be a string.

**silenced { false | true}**

Specify whether to display silenced alerts.

Must be one of the following:

- **false**
- **true**

Default Value: false.

**unprocessed { false | true}**

Specify whether to display unprocessed alerts.

Must be one of the following:

- **false**
- **true**

Default Value: false.

**Usage Guidelines**

Use this command to view filtered list of active alerts.

## alerts add-silence

Adds a silence.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

```
add-silence id silence_id startsAt silence_start_time endsAt silence_end_time
createdBy silence_creator_id comment additional_info
```

**comment *additional\_info***

Specify additional information for the silence.

Must be a string.

**createdBy *silence\_creator\_id***

Specify the silence creator identity.

Must be a string.

**endsAt *silence\_end\_time***

Specify the silence end time.

Must be a string in the date-and-time pattern. For information on the date-and-time pattern, see the Input Pattern Types section.

**id *silence\_id***

Specify the silence ID to add.

Must be a string in the uuid pattern. For information on the uuid pattern, see the Input Pattern Types section.

**startsAt *silence\_start\_time***

Specify the silence start time.

Must be a string in the date-and-time pattern. For information on the date-and-time pattern, see the Input Pattern Types section.

**Usage Guidelines**

Use this command to add a silence.

## alerts add-silence matchers

Configures the list of label values to filter alerts.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

**matchers** *alert\_label\_name* **value** *filter\_value* **isRegex** { **false** | **true**}

**isRegex { false | true }**

Specify whether the value is a regular expression or not.

Must be one of the following:

- **false**
- **true**

**value *filter\_value***

Specify the filter value.

Must be a string.

***alert\_label\_name***

Specify the alert label name.

Must be a string.

**Usage Guidelines** Use this command to configure the list of label values to filter alerts.

## alerts delete-silence

Deletes specified silence.

**Command Modes** Exec > Global Configuration

**Syntax Description** **delete-silence id** *silence\_id\_to\_delete*

***id silence\_id\_to\_delete***

Specify the ID of the silence to delete.

Must be a string in the uuid pattern. For information on the uuid pattern, see the Input Pattern Types section.

**Usage Guidelines** Use this command to delete specified silence.

## alerts history

Displays alerts history.

**Command Modes** Exec > Global Configuration

**Syntax Description** **history filter** *alerts\_filter*

***filter alerts\_filter***

Specify the comma-separated values to filter alerts.

Must be a string.

**Usage Guidelines** Use this command to view alerts history.

## alerts silence-by-id

Displays information about a specific silence.

**Command Modes** Exec > Global Configuration

**Syntax Description** **silence-by-id id** *silence\_id*

**id *silence\_id***

Specify the silence ID.

Must be a string in the uuid pattern. For information on the uuid pattern, see the Input Pattern Types section.

---

**Usage Guidelines** Use this command to view information about a specific silence.

## alerts silences

Displays filtered list of silences.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** **silences filter** *alerts\_filter*

**filter *alerts\_filter***

Specify the alerts filter in the format *label expr value or regex*.

Must be a string.

---

**Usage Guidelines** Use this command to view filtered list of silences.





## CHAPTER 4

# Bulk Statistics Config Mode Command Reference

- [bulk-stats](#), on page 103
- [bulk-stats current](#), on page 104
- [bulk-stats pod-query](#), on page 105
- [bulk-stats query](#), on page 106
- [bulk-stats vnf-alias](#), on page 107

## bulk-stats

Configures bulk statistics parameters.

### Command Modes

Exec > Global Configuration

### Syntax Description

```
bulk-stats { enable { false | true } | user user_name | external-ip ip_address | external-port port_number | interval-minutes create_interval | prune-interval-days prune_interval | vnf-name vnf_name | global-default-value global_default_value | global-default-namespace global_default_namespace }
```

### **enable { false | true }**

Specify to enable or disable bulk statistics.

Must be one of the following:

- **false**
- **true**

Default Value: true.

### **external-ip *ip\_address***

Specify the external IP address for downloading the bulk statistics over SFTP.

Must be an IPv4 address.

-Or-

Must be an IPv6 address.

**external-port *port\_number***

Specify the external port number for downloading the bulk statistics over SFTP.

Must be an integer.

Default Value: 2222.

**global-default-namespace *global\_default\_namespace***

Specify the namespace used in the bulk statistics file if the query did not return any value.

Must be a string.

**global-default-value *global\_default\_value***

Specify the value used in the bulk statistics file if the query did not return any value.

Must be a string.

**interval-minutes *create\_interval***

Specify the interval for creating the bulk statistics in minutes.

Must be an integer.

Default Value: 1.

**prune-interval-days *prune\_interval***

Specify the time interval, in number of days, to remove the bulk statistics.

Must be an integer.

Default Value: 1.

**user *user\_name***

Specify the user authorized to download the bulk statistics.

Must be a string.

Default Value: admin.

**vnf-name *vnf\_name***

Specify the VNF name to be added to the bulk statistics CSV file.

Must be a string.

Default Value: default.

---

**Usage Guidelines**

Use this command to configure bulk statistics parameters.

## bulk-stats current

Displays the list of current bulk statistics.

**Command Modes**

Exec &gt; CEE

**Syntax Description**

```
show bulk-stats current [uid unique_id | namespace bulk_statistics_namespace |
key bulk_statistics_key | label bulk_statistics_label | value bulk_statistics_value]
```

**alias** *bulkstats\_alias*

Specify the bulkstats alias.

Must be a string.

**labels** *bulkstat\_labels*

Specify multiple bulkstat labels.

Must be a string.

**metric** *bulkstats\_metric*

Specify the bulkstats metric name.

Must be a string.

**namespace** *bulk\_statistics\_namespace*

Specify the bulk statistics namespace.

Must be a string.

**uid** *unique\_id*

Specify the unique identifier.

Must be a string.

**value** *bulk\_statistics\_value*

Specify the value of the bulk statistics.

Must be of type decimal64, with 3 fraction digits.

**Usage Guidelines**

Use this command to view the list of current bulk statistics.

## bulk-stats pod-query

Configures the queries for retrieving the bulk statistics.

**Command Modes**

Exec &gt; Global Configuration

**Syntax Description**

```
bulk-stats pod-query field_name { query query_name | rate-query rate_query_name
| default-value default_value}
```

**default-value** *default\_value*

Specify the value used in the bulk statistics file if the query did not return any value.

Must be a string.

**query *query\_name***

Specify the query to be executed in Prometheus Query Language (PromQL). The query must be grouped by namespace and pod\_name or pod.

Must be a string.

**rate-query *rate\_query\_name***

Specify the rate query to be executed in PromQL format. The rate query must use rate (not irate) and include group by namespace and pod\_name (or pod). Use \$INTERVAL for interval.

Must be a string.

**field\_name**

Specify the name of the field to add to the pod statistics.

Must be a string.

**Usage Guidelines**

Use this command to configure the queries for retrieving the bulk statistics.

## bulk-stats query

Configures the query to retrieve bulk statistics data.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

```
bulk-stats query query_name { expression expression | label label | default-value
default_value | default-namespace default_namespace}
```

**alias *bulkstats\_alias***

Specify the bulkstats alias.

Must be a string.

**default-namespace *default\_namespace***

Specify the namespace used in bulk statistics file if the query did not return any value.

Must be a string.

**default-value *default\_value***

Specify the value used in bulk statistics file if the query did not return any value.

Must be a string.

**expression *expression***

Specify the query to execute in PromQL format.

Must be a string.

**label *bulkstat\_label***

Specify a single bulkstat label. Stats will populate the label in AVP format.

Must be a string.

**labels *bulkstat\_labels***

Specify multiple bulkstat labels. Stats will populate the label in AVP format separated by semicolon (;).

Must be a string.

**query\_name**

Specify the query name.

Must be a string.

---

**Usage Guidelines** Use this command to configure the query to execute to retrieve the bulk statistics data.

## bulk-stats vnf-alias

Configures the VNF alias for a given namespace.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** **bulk-stats vnf-alias** *namespace* **alias** *alias*

**alias *alias***

Specify the alias to apply.

Must be a string.

**namespace**

Specify the namespace to apply the alias.

Must be a string.

---

**Usage Guidelines** Use this command to configure the VNF alias for a given namespace.





## CHAPTER 5

# CIMC Config Mode Command Reference

- [cimc](#), on page 109
- [cimc cluster](#), on page 109
- [cimc cluster default](#), on page 110
- [cimc cluster server](#), on page 110
- [node-problem-detector agent](#), on page 111

## cimc

Configures the CIMC Alerts Exporter configuration.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** `cimc enabled { false | true }`

**enabled { false | true }**

Specify to enable or disable the CIMC Alerts Exporter.

Must be one of the following:

- **false**
- **true**

Default Value: false.

---

**Usage Guidelines** Use this command to configure the CIMC Alerts Exporter configuration.

## cimc cluster

Configures the list of CIMC clusters to be monitored.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** `cluster cluster_name`

***cluster\_name***

Specify the cluster name, used for label.

Must be a string in the pattern [a-z0-9][a-z0-9\.\-]\*[a-z0-9].

**Usage Guidelines**

Use this command to configure the list of CIMC clusters to be monitored.

## cimc cluster default

Configures the default values to be used on all CIMC connections.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

**default** **username** *user\_name* **password** *password*

**password** *password*

Specify the default password used to connect CIMC.

Must be a string.

**username** *user\_name*

Specify the default user name used to connect CIMC.

Must be a string.

**Usage Guidelines**

Use this command to configure the default values to be used on all CIMC connections.

## cimc cluster server

Configures the CIMC server parameters.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

**server** *cimc\_server\_name* **ip** *ip\_address/host\_name* **username** *user\_name* **password** *password*

**ip** *ip\_address/host\_name*

Specify the CIMC server IP address or host name.

Must be a string.

**password** *password*

Specify the password to connect CIMC.

Must be a string.

**username** *user\_name*

Specify the user name to connect CIMC.

Must be a string.

***cimc\_server\_name***

Specify the CIMC server name to use in alerts.

Must be a string.

---

**Usage Guidelines** Use this command to configure the default values to be used on all CIMC connections.

## node-problem-detector agent

Enable Node Problem Detector (NPD), the diagnostic tool, to identify and report problems that may impact node performance.

---

**Command Modes** Exec

---

**Syntax Description** `node-problem-detector agent enabled`

---

**Usage Guidelines** This command enables NPD with default configurations and built-in monitoring rules for detecting well-known issues. By default NPD is disabled.

When the NPD is enabled, it is easy to monitor and manage the health of all nodes on the cluster.

node-problem-detector agent



## CHAPTER 6

# Cluster Exec Mode Command Reference

---

- [cluster](#), on page 113
- [cluster configmaps](#), on page 113
- [cluster configmaps detail](#), on page 114
- [cluster connect](#), on page 114
- [cluster ingresses](#), on page 115
- [cluster ingresses detail](#), on page 115
- [cluster namespaces](#), on page 115
- [cluster nodes](#), on page 116
- [cluster nodes detail](#), on page 117
- [cluster persistent-volume-claims](#), on page 117
- [cluster persistent-volumes](#), on page 118
- [cluster pods](#), on page 118
- [cluster pods delete](#), on page 119
- [cluster pods detail](#), on page 120
- [cluster services](#), on page 120
- [cluster services detail](#), on page 120

## cluster

Displays the cluster details.

---

**Command Modes** Exec

---

**Syntax Description** `show cluster`

---

**Usage Guidelines** Use this command to view the cluster details.

## cluster configmaps

Displays the current configuration maps.

---

**Command Modes** Exec

---

**Syntax Description** `show cluster configmaps namespace configmap_name detail`

***configmap\_name***

Specify the configuration map name.

Must be a string.

***namespace***

Specify the configmap namespace.

Must be a string.

---

**Usage Guidelines** Use this command to view the current configuration maps.

## cluster configmaps detail

Displays configmap details.

---

**Command Modes** Exec

---

**Syntax Description** `cluster configmaps namespace configmap_name details`

---

**Usage Guidelines** Use this command to view configmap details.

## cluster connect

Connect to and debug K8s pods and containers.

---

**Command Modes** Exec

---

**Syntax Description** `cluster connect namespace namespace pod-name pod_name container container_name`

***namespace***

Specify the node namespace.

Must be a string.

***pod\_name***

Specify the pod name.

Must be a string.

***container\_name***

Specify the container name.

Must be a string.

**Usage Guidelines** Use this command to connect to and debug the K8s pods and containers.

## cluster ingresses

Displays the current ingresses.

**Command Modes** Exec

**Syntax Description** `show cluster ingresses namespace ingress-name ingress_name host-name host_name`

**host-name** *host\_name*

Specify the host name.

Must be a string.

**ingress-name** *ingress\_name*

Specify the ingress name.

Must be a string.

**namespace**

Specify the ingress namespace.

Must be a string.

**Usage Guidelines** Use this command to view the current ingresses.

## cluster ingresses detail

Displays ingress details.

**Command Modes** Exec

**Syntax Description** `cluster ingresses namespace ingress-name ingress_name host-name host_name details`

**Usage Guidelines** Use this command to view ingress details.

## cluster namespaces

Configures the current namespace.

**Command Modes** Exec

**Syntax Description** `show cluster namespaces namespace smi-application smi_application_name istio-enabled { true | false}`

**istio-enabled { true | false}**

Enables or disables Istio.

Must be one of the following:

- **false**
- **true**

**smi-application *smi\_application\_name***

Specify the name of the SMI application.

Must be a string.

**namespace**

Specify the cluster's namespace.

Must be a string.

**Usage Guidelines**

Use this command to configure the current namespace.

## cluster nodes

Displays the current nodes in the cluster.

**Command Modes**

Exec

**Syntax Description**

```
show cluster nodes node_name status status version version ip ip_address os-image
os_image kernel-version kernel_version container-runtime container_runtime
```

**container-runtime *container\_runtime***

Specify the container runtime.

Must be a string.

**ip *ip\_address***

Specify the node's IP address.

Must be a string.

**kernel-version *kernel\_version***

Specify the Kernel version.

Must be a string.

**os-image *os\_image***

Specify the OS image.

Must be a string.

**status *status***

Specify the status of the nodes.

Must be a string.

**version *version***

Specify the K8s version of the nodes.

Must be a string.

***name***

Specify the name of the nodes.

Must be a string.

---

**Usage Guidelines** Use this command to view the current nodes in the cluster.

## cluster nodes detail

Displays node details.

---

**Command Modes** Exec

---

**Syntax Description** `cluster nodes node_name detail`

---

**Usage Guidelines** Use this command to view node details.

## cluster persistent-volume-claims

Displays the current persistent volume claims.

---

**Command Modes** Exec

---

**Syntax Description** `show cluster persistent-volume-claims namespace pvc_name status volume volume capacity capacity storageclass storage_class`

**capacity *capacity***

Specify the volume capacity.

Must be a string.

**pvc-name *pvc\_name***

Specify the persistent volume class name.

Must be a string.

**status**

Specify the status of the persistent volume claims.

Must be a string.

**storageclass *storage\_class***

Specify the storage class.

Must be a string.

**volume *volume***

Specify the volume.

Must be a string.

**namespace**

Specify the namespace of the volume persistent claims.

Must be a string.

---

**Usage Guidelines** Use this command to view the current persistent volume claims.

## cluster persistent-volumes

Displays the current persistent volume details.

---

**Command Modes** Exec

---

**Syntax Description** `show cluster persistent-volumes pvc_name status`

***pvc\_name***

Specify the persistent volume class (PVC) name.

Must be a string.

***status***

Specify the status of the persistent volume.

Must be a string.

---

**Usage Guidelines** Use this command to view details of the current persistent volumes.

## cluster pods

Displays the current pod details.

---

**Command Modes** Exec

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax Description</b> | <pre><b>show cluster pods</b> <i>namespace</i> <b>pod-name</b> <i>pod_name</i> <b>ready</b> <i>readiness</i> <b>status</b> <b>restarts</b> <b>start-time</b> <i>start_time</i></pre> <p><b>pod-name</b> <i>pod_name</i></p> <p>Specify the name of the pod.<br/>Must be a string.</p> <p><b>ready</b> <i>readiness</i></p> <p>Specify the container readiness.<br/>Must be a string.</p> <p><b>restarts</b></p> <p>Restarts the pods.<br/>Must be an integer.</p> <p><b>start-time</b> <i>start_time</i></p> <p>Specify the start time.<br/>Must be a string.</p> <p><b>status</b></p> <p>Specify the status of the pods.<br/>Must be a string.</p> <p><b>namespace</b></p> <p>Specify the namespace of the cluster services.<br/>Must be a string.</p> |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Usage Guidelines** Use this command to view the current pod details.

## cluster pods delete

Deletes pods.

**Command Modes** Exec

**Syntax Description**

```
cluster pods namespace pod_name delete
```

**Usage Guidelines** Use this command to delete pods.

## cluster pods detail

Displays pod details.

---

**Command Modes** Exec

---

**Syntax Description** `cluster pods namespace pod_name detail`

---

**Usage Guidelines** Use this command to view pod details.

## cluster services

Displays the current services in the cluster.

---

**Command Modes** Exec

---

**Syntax Description** `show cluster services namespace service-name service_name cluster-ip cluster_ip external-ips external_ips`

**cluster-ip cluster\_ip**

Specify the IP address of the cluster.

Must be a string.

**external-ips external\_ips**

Specify the external IPs.

Must be a string.

**service-name service\_name**

Specify the service name.

Must be a string.

**namespace**

Specify the namespace of the cluster services.

Must be a string.

---

**Usage Guidelines** Use this command to view the current services in the cluster.

## cluster services detail

Displays cluster service details.

---

**Command Modes** Exec

---

**Syntax Description** `cluster services namespace service_name detail`

---

**Usage Guidelines** Use this command to view cluster service details.





## CHAPTER 7

# Debug Exec Mode Command Reference

- [tac-debug-pkg create](#), on page 123
- [tac-debug-pkg create cores-filter](#), on page 125
- [tac-debug-pkg create logs-filter](#), on page 125
- [tac-debug-pkg delete](#), on page 125
- [tac-debug-pkg merge](#), on page 126
- [tac-debug-pkg status](#), on page 126

## tac-debug-pkg create

Creates TAC debug information.

**Command Modes** Exec > Global Configuration

**Syntax Description** `tac-debug-pkg create { from yyyy-mm-dd_hh:mm:ss | to yyyy-mm-dd_hh:mm:ss | last time_to_now | cfg { false | true } | cores { false | true } | logs { false | true } | metrics { false | true } | stats { false | true } }`

### **cfg { false | true }**

Specify to enable or disable Ops Center configuration collection. To enable, set to true.

Must be one of the following:

- false
- true

Default Value: true.

### **cores { false | true }**

Specify to enable or disable core files collection. To enable, set to true.

Must be one of the following:

- false
- true

Default Value: true.

**from *yyyy-mm-dd\_hh:mm:ss***

Specify the start time in *yyyy-mm-dd\_hh:mm:ss* format.

Must be a string.

**last *time\_to\_now***

Specify the time to now in number of days, hours, minutes, or seconds.

Must be a string.

**logs { false | true }**

Specify to enable or disable logs collection. To enable, set to true.

Must be one of the following:

- false
- true

Default Value: true.

**metrics { false | true }**

Specify to enable or disable metrics collection. To enable, set to true.

Must be one of the following:

- false
- true

Default Value: true.

**stats { false | true }**

Specify to enable or disable bulk statistics collection. To enable, set to true.

Must be one of the following:

- false
- true

Default Value: true.

**to *yyyy-mm-dd\_hh:mm:ss***

Specify the End time: *yyyy-mm-dd\_hh:mm:ss* format.

Must be a string.

---

**Usage Guidelines**

Use this command to create TAC debug information.

## tac-debug-pkg create cores-filter

Configures the filter for gathering cores.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** `tac-debug-pkg create cores-filter process process_name`

**process *process\_name***

Specify the name of the process with coredump.

Must be a string.

---

**Usage Guidelines** Use this command to configure the filter for gathering cores.

## tac-debug-pkg create logs-filter

Configures filter for gathering logs.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** `tac-debug-pkg create logs-filter { pod_name pod_name | namespace namespace }`

**namespace *namespace***

Specify the namespace.

Must be a string.

**pod\_name *pod\_name***

Specify the pod name.

Must be a string.

---

**Usage Guidelines** Use this command to configure the filter for gathering logs. Filters are combined. Namespace overrides pod\_name to include all pods.

## tac-debug-pkg delete

Removes TAC debug information.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** `tac-debug-pkg delete { tac-id tac_id | last days }`

**last *days***

Specify the packages created in last nth days.

Must be an integer.

**tac-id *tac\_id***

Specify the TAC debug package ID.

Must be a string.

---

**Usage Guidelines** Use this command to remove debug information.

## tac-debug-pkg merge

Creates single data collection.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** **merge**

**tac-id *tac\_id***

Specify the TAC debug package ID.

Must be a string.

---

**Usage Guidelines** Use this command to create single data collection.

## tac-debug-pkg status

Displays the status of the current TAC debug gathering session.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** **tac-debug-pkg status**

---

**Usage Guidelines** Use this command to view status of the current TAC debug gathering session.



## CHAPTER 8

# Grafana Config Mode Command Reference

- [grafana](#), on page 127
- [grafana dashboards](#), on page 127
- [grafana enable-basic-auth](#), on page 128

## grafana

Configures Grafana parameters.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** `grafana enable { false | true }`

**enable { false | true }**

Specify to enable or disable Grafana and dashboard.

Must be one of the following:

- **false**
- **true**

Default Value: true.

---

**Usage Guidelines** Use this command to configure Grafana parameters.

## grafana dashboards

Configures Git repositories containing dashboards.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** `grafana dashboards [ git_repo_name ] git-url git_url`

**git-url *git\_url***

Specify the Git URL.

Must be a string.

**git\_repo\_name**

Specify name of the Git repository.

Must be a string.

---

**Usage Guidelines** Use this command to configure Git repositories containing dashboards.

## grafana enable-basic-auth

Configure basic authentication in CEE Ops Center.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** `grafana enable-basic-auth { false | true }`

**enable-basic-auth { false | true }**

Specify to enable or disable basic authentication in CEE Ops Center.

Must be one of the following:

- **false**
- **true**

Default value: **true**

---

**Usage Guidelines** Use this command to configure basic authentication in CEE Ops Center for custom Grafana dashboards.



## CHAPTER 9

# Logging Config Mode Command Reference

- [logging fluent](#), on page 129
- [logging fluent tls](#), on page 130
- [logging fluentd](#), on page 130
- [logging listener](#), on page 131
- [logging loki](#), on page 132
- [logging splunk](#), on page 132
- [logging syslog](#), on page 133
- [logging worker](#), on page 134

## logging fluent

Configures Fluent Forwarding parameters.

### Command Modes

Exec > Global Configuration

### Syntax Description

```
logging fluent { host host_info | port port_number | protocol outbound_protocol |
disable-tls { false | true } | disable-tls-verification { false | true
} | flush-interval flush_interval | storage-limit storage_limit }
```

#### **host** *host\_info*

Specify the Fluentbit or Fluentd instance host information.

Must be a string.

#### **port** *port\_number*

Specify the Fluentbit or Fluentd instance port number.

Must be an integer.

#### **protocol** *outbound\_protocol*

Specify the outbound protocol.

Must be one of the following:

- **forward**

- **http**

Default Value: **http**

---

**Usage Guidelines**

Use this command to configure Fluent Forwarding parameters to enable log forwarding to Fluent endpoint.

## logging fluent tls

Configures TLS communication with Splunk endpoint and TLS certification verification parameters.

---

**Command Modes**

Exec > Global Configuration

---

**Syntax Description**

```
tls { disable-tls { false | true } | disable-tls-verification { false | true } }
```

**disable-tls { false | true }**

Specify to enable or disable TLS communication with Splunk endpoint. To enable, set to false.

Must be one of the following:

- **false**
- **true**

Default Value: false.

**disable-tls-verification { false | true }**

Specify to enable or disable TLS certification verification. To enable, set to false.

Must be one of the following:

- **false**
- **true**

Default Value: false.

---

**Usage Guidelines**

Use this command to configure TLS communication with Splunk endpoint and TLS certification verification parameters.

## logging fluentd

Configures FluentD parameters.

---

**Command Modes**

Exec > Global Configuration

---

**Syntax Description**

```
fluentd workers { number_of_workers | buffer-total-limit-size buffer_size_limit | buffer-chunk-limit-size chunk_size_limit | flush-interval flush_interval }
```

**buffer-chunk-limit-size *chunk\_size\_limit***

Specify the maximum size of each chunk in MB.

Must be an integer in the range of 1-10.

Default Value: 8

**buffer-total-limit-size *buffer\_size\_limit***

Specify the size limitation of the buffer in GB.

Must be an integer in the range of 1-3.

Default Value: 1

**flush-interval *flush\_interval***

Specify the flush interval in seconds.

Must be an integer in the range of 1-10.

Default Value: 5.

**workers *number\_of\_workers***

Specify the number of workers.

Must be an integer in the range of 1-5.

Default Value: 2

---

**Usage Guidelines** Use this command to configure FluentD parameters.

## logging listener

Enables the Logs Listener for incoming logs.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** **listener enable external-ip** *ip\_address* **udp-port** *port\_number* **buffer-max-size** *buffer\_max\_size* **buffer-chunk-size** *buffer\_chunk\_max\_size*

**enable**

Specify to enable Logs Listener.

**external-ip *ip\_address***

Specify the exposed IP endpoint for incoming logs.

Must be an IPv4 address.

-Or-

Must be an IPv6 address.

**udp-port *port\_number***

Specify the Listener UDP port number.

Must be an integer.

Default Value: 514.

**Usage Guidelines** Use this command to enable the Logs Listener for incoming logs.

## logging loki

Configures the Grafana Loki parameters.

**Command Modes** Exec > Global Configuration

**Syntax Description** `logging loki [ enable | retention-period retention_period ]`

**enable**

Specify to enable Grafana Loki Logging Visualization.

**retention-period *retention\_period***

Specify the retention period.

Must be a string.

**Usage Guidelines** Use this command to configure Grafana Loki parameters.

## logging splunk

Configures Splunk endpoint.

**Command Modes** Exec > Global Configuration

**Syntax Description** `logging splunk { host host_info | port port_number | auth-token auth_token }`

**auth-token *auth\_token***

Specify the Splunk Authentication Token for the HTTP Event Collector interface.

Must be a string.

**disable-tls { false | true }**

Specify to enable or disable TLS communication with Splunk endpoint. To enable, set to false.

Must be one of the following:

- **false**
- **true**

Default Value: false.

**disable-tls-verification { false | true }**

Specify to enable or disable TLS certification verification. To enable, set to false.

Must be one of the following:

- false
- true

Default Value: false.

**host *host\_info***

Specify the Splunk host information.

Must be a string.

**port *port\_number***

Specify the Splunk port number.

Must be an integer.

**Usage Guidelines**

Use this command to configure Splunk endpoint to enable log forwarding to Splunk endpoint using HTTP Event Collector interface.

## logging syslog

Configure log forwarding to the Syslog server.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

```
logging syslog { host server_host | mode server_mode | port server_port |
syslog_format syslog_format | syslog_maxsize syslog_maxsize }
```

**host *server\_host***

Specify the domain or IP address of the remote Syslog server.

**mode *server\_mode***

Specify the TCP, TLS, or UDP transport type.

**port *server\_port***

Specify the TCP, TLS, or UDP port of the remote Syslog server.

**syslog\_format *syslog\_format***

Specify the *rfc3164* or *rfc5424* Syslog protocol format to use.

**syslog\_maxsize *syslog\_maxsize***

Specify the maximum size allowed per message. The value must be an integer representing the number of bytes allowed.

**Usage Guidelines**

Use this command to configure Fluent-bit to enable log forwarding to the Syslog server.

## logging worker

Enables CEE log forwarding for Fluent Worker pods.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

```
logging worker [drop-namespace-logs namespace_names | drop-pod-logs pod_names
| exclude-logs-with-annotation true | keep-pod-logs pod_names |
keep-namespace-logs namespace_names | drop-os-service-logs [service_names |
remove-keys [keys]]
```

**drop-namespace-logs *namespace\_names***

Specify to drop logs by namespaces. *namespace\_names* must be a regex string with selected namespace names inside double quotes.

**drop-pod-logs *pod\_names***

Specify to drop logs by pods. *pod\_names* must be a regex string with selected pod names inside double quotes.

**exclude-logs-with-annotation true**

Specify to exclude logs from selected pods using annotation.



**Note** After adding or removing annotation from any pod, it is required to restart the fluent-worker pod for the changes to take effect.

**keep-namespace-logs *namespace\_names***

Specify to retain logs by namespaces. *namespace\_names* must be a regex string with selected namespace names inside double quotes.

**keep-pod-logs *pod\_names***

Specify to retain logs by pods. *pod\_names* must be a regex string with selected pod names inside double quotes.

**drop-os-service-logs [ *service\_names* ]**

Specify to drop logs from selected OS services. The currently supported values for *services\_names* are audit, kernel, or kubelet.

**remove-keys [ *keys* ]**

Specify to remove keys from log entries. The log entry keys to be dropped are case sensitive.

---

**Usage Guidelines**

Use this command to enable CEE log forwarding for Fluent Worker pods. The filters on Fluent worker pods that intake the logs from each node reduce the volume of logs being forwarded.

■ logging worker



# CHAPTER 10

## NPD Config Mode Command Reference

- [node-problem-detector agent](#), on page 137
- [node-problem-detector agent exporters](#), on page 137
- [node-problem-detector agent monitors](#), on page 138
- [node-problem-detector agent exporters k8s](#), on page 139
- [node-problem-detector agent exporters prometheus](#), on page 139

### node-problem-detector agent

Enable Node Problem Detector (NPD), the diagnostic tool, to identify and report problems that may impact node performance.

#### Command Modes

Exec

#### Syntax Description

```
node-problem-detector agent enabled
```

#### Usage Guidelines

This command enables NPD with default configurations and built-in monitoring rules for detecting well-known issues. By default NPD is disabled.

When the NPD is enabled, it is easy to monitor and manage the health of all nodes on the cluster.

### node-problem-detector agent exporters

Configure the Node Problem Detector (NPD) exporter definitions.

```
node-problem-detector agent exporters { k8s apiserver-uri-override string | prometheus port port_number }
```

```
no node-problem-detector agent exporters { k8s apiserver-uri-override string | prometheus port port_number }
```

#### Syntax Description

**k8s apiserver-uri-override** *string* Configure the custom URI that is used to connect to the Kubernetes API server. The custom URI can be an alphanumeric string.

**prometheus port** *port\_number* Specify the port number of Prometheus exporter. Port number is an integer from 0 to 65535.

**Command Default** NPD functionality is disabled.

**Command Modes** Global configuration

| Command History | Release   | Modification                 |
|-----------------|-----------|------------------------------|
|                 | 2025.02.1 | This command was introduced. |

**Usage Guidelines** Use this command to enable the NPD tool to monitor the health of nodes within the cluster.

### Example

This example shows how to configure the NPD in the Kubernetes environment.

```
node-problem-detector agent exporters k8s apiserver-uri-override http://example:8443
```

## node-problem-detector agent monitors

Configure the Node Problem Detector (NPD) monitor definitions.

**node-problem-detector agent monitors** *monitor\_name* { **custom** | **filelog** | **journald** | **ksmg** }

**no node-problem-detector agent monitors**

| Syntax Description | monitors <i>monitor_name</i> | Configure the monitor name that is used as the source value for reporting the node problems. |
|--------------------|------------------------------|----------------------------------------------------------------------------------------------|
|                    | <b>custom</b>                | Specify to use custom plugin.                                                                |
|                    | filelog                      | Specify to use filelog plugin.                                                               |
|                    | journald                     | Specify to use journald plugin.                                                              |
|                    | ksmg                         | Specify to use ksmg plugin.                                                                  |

**Command Default** NPD functionality is disabled.

**Command Modes** Global configuration

| Command History | Release   | Modification                 |
|-----------------|-----------|------------------------------|
|                 | 2025.02.1 | This command was introduced. |

**Usage Guidelines** Use this command to enable the NPD tool to monitor the health of nodes within the cluster.

**Example**

This example shows how to configure the NPD monitor in the Kubernetes environment.

```
node-problem-detector agent monitors test
```

## node-problem-detector agent exporters k8s

Configure the K8s exporter to enable communication with the Kubernetes API server for monitoring resources.

**node-problem-detector agent exporters k8s apiserver-uri-override** *uri\_string*

**Syntax Description**

**apiserver-uri-override***uri\_string* This keyword allows you to specify a custom URI for connecting to the API server.

This URI can include the IP address, port, and other options needed for the connection.

If you are managing multiple clusters and need NPD to export problems to a centralized API server, this override is essential.

**Command Default**

None

**Command Modes**

Global configuration mode

**Command History**

| Release           | Modification                 |
|-------------------|------------------------------|
| Release 2025.02.1 | This command was introduced. |

**Usage Guidelines**

This configuration must be performed to enable communication with the Kubernetes API server.

**Usage Guidelines****Example**

The following example shows how to configure Kubernetes API server IP address and port for K8s to communicate with.

```
node-problem-detector agent exporters k8s apiserver-override=https://209.165.200.227:8080
```

## node-problem-detector agent exporters prometheus

Configure the Node Problem Detector to act as an agent that exports node health metrics to Prometheus.

**node-problem-detector agent exporters prometheus port** *port\_number*

---

**Syntax Description**     `port`*port\_number* Specify the port to bind the Prometheus scrape endpoint.  
The default value of port number is 20257. Use 0 to disable.

---

**Command Default**     None

**Command Modes**     Global configuration mode

---

| Command History | Release           | Modification                 |
|-----------------|-------------------|------------------------------|
|                 | Release 2025.02.1 | This command was introduced. |

---

**Usage Guidelines**     This command is used to set up Node Problem Detector to work as an agent that collects node health data and exports it to Prometheus for monitoring and alerting purposes. It is an essential part of maintaining the health and reliability of Kubernetes clusters in production environments.

### Example

The following example shows the configuration of port to bind the Prometheus scrape endpoint.

```
node-problem-detector agent exporters prometheus port 8080
```



# CHAPTER 11

## Prometheus Config Mode Command Reference

- [prometheus](#), on page 141
- [prometheus federation](#), on page 142
- [prometheus federation exported-query-nodes](#), on page 142
- [prometheus federation remote-cluster-certs](#), on page 143
- [prometheus kvm-metrics defaults](#), on page 144
- [prometheus kvm-metrics monitor-server](#), on page 144
- [prometheus prometheus-operator](#), on page 145
- [prometheus pushgateway](#), on page 145
- [prometheus pushgateway port](#), on page 146
- [prometheus recording-rules group](#), on page 146
- [prometheus recording-rules group rule](#), on page 146
- [prometheus recording-rules group rule label](#), on page 147
- [prometheus server-settings](#), on page 147

### prometheus

Configures Prometheus-related parameters.

**Command Modes** Exec > Global Configuration (config)

**Syntax Description** `prometheus { scrape-interval scrape_interval | volume volume_size | query-mode query_mode }`

#### **query-mode *query\_mode***

Specify the query mode.

Must be one of the following:

- **client**
- **server**

#### **scrape-interval *scrape\_interval\_frequency***

Specify the frequency at which Prometheus fetches metrics in seconds.

Must be an integer in the range of 10-900.

Default Value: 10.

**volume *volume\_size***

Specify the volume size in GB to be used if useVolumeClaims is true.

Must be an integer in the range of 20-1024.

Default Value: 100.

**Usage Guidelines** Use this command to configure Prometheus-related parameters.

## prometheus federation

Configures scraping metrics from other cluster.

**Command Modes** Exec > Global Configuration (config)

**Syntax Description** `prometheus federation [ subordinates ip_address/host_name ]`

**subordinates *ip\_address/host\_name***

Specify the other cluster metrics server in the format *ip\_address/host\_name*.

Must be a string.

**Usage Guidelines** Use this command to scrape metrics from other cluster.

## prometheus federation exported-query-nodes

Configures the exported query nodes.

**Command Modes** Exec > Global Configuration (config)

**Syntax Description** `prometheus federation exported-query-nodes [ address ip_address | port port_number ]`

**address *ip\_address***

Specify the IP address.

Must be an IPv4 address.

-Or-

Must be an IPv6 address.

**port *port\_number***

Specify the port number.

Must be an integer in the range of 0-65535.

---

**Usage Guidelines**

Use this command to configure the exported query nodes.

## prometheus federation remote-cluster-certs

Configures cluster-specific TLS/SSL certificate configuration.

---

**Command Modes**

Exec > Global Configuration (config)

---

**Syntax Description**

```
remote-cluster-certs remote_cluster_name { address remote_cluster_ip_address |
ssl-key ssl_key_certificate | ssl-crt ssl_cert_certificate | ssl-ca
ssl_certificate_authority | alert-rx-port port_number}
```

**address** *remote\_cluster\_ip\_address*

Specify the remote cluster's IP address.

Must be an IPv4 address.

-Or-

Must be an IPv6 address.

**alert-rx-port** *port\_number*

Specify the web port number to receive the alerts.

Must be an integer in the range of 8700-8750.

**ssl-ca** *ssl\_certificate\_authority*

Specify the SSL certificate authority.

Must be a string.

**ssl-crt** *ssl\_cert\_certificate*

Specify the SSL certificate.

Must be a string.

**ssl-key** *ssl\_key\_certificate*

Specify the SSL Key certificate.

Must be a string.

**remote\_cluster\_name**

Specify the remote cluster's name.

Must be a string.

---

**Usage Guidelines**

Use this command to configure cluster-specific TLS/SSL certificate configuration.

You can configure a maximum of 10 elements with this command.

## prometheus kvm-metrics defaults

Configures default values used by all connections.

**Command Modes** Exec > Global Configuration (config)

**Syntax Description** `kvm-metrics defaults private-key default_private_key user default_user`

**private-key default\_private\_key**

Specify the default private key for connections.

Must be a string.

**user default\_user**

Specify the default user for connections.

Must be a string.

**Usage Guidelines** Use this command to enable monitoring of KVM-only machines, and to configure default values used by all connections.

## prometheus kvm-metrics monitor-server

Configures monitor server targets.

**Command Modes** Exec > Global Configuration (config)

**Syntax Description** `kvm-metrics monitor-server { address ip_address | hostname host_name | user user | private-key private_key }`

**address ip\_address**

Specify the IP address to monitor.

Must be an IPv4 address.

-Or-

Must be an IPv6 address.

**hostname host\_name**

Specify the host name.

Must be a string.

**private-key private\_key**

Specify the private key for connections.

Must be a string.

**user *user***

Specify the user for connections.

Must be a string.

**Usage Guidelines** Use this command to enable monitoring of KVM-only machines, and configure monitor server targets.

## prometheus prometheus-operator

Configure Prometheus installation using Prometheus Operator.

The Prometheus Operator provides Kubernetes native deployment and management of Prometheus and related monitoring components. This operator simplifies and automate the configuration of a Prometheus based monitoring stack for Kubernetes clusters.

**Command Modes** Exec > Global Configuration (config)

**Syntax Description** `prometheus prometheus-operator { enabled | disabled}`

**prometheus-operator { enabled | disabled}**

Specify to enable or disable the Prometheus operator.

Default value: Disabled

**Usage Guidelines** Use this command to install Prometheus using the Prometheus operator.

## prometheus pushgateway

Configure Prometheus Pushgateway support.

The Pushgateway is an intermediary service which allows you to push metrics from jobs which cannot be scraped.

**Command Modes** Exec > Global Configuration (config)

**Syntax Description** `prometheus pushgateway { enabled | disabled}`

**pushgateway { enabled | disabled}**

Specify to enable or disable the Prometheus Pushgateway.

Default value: Disabled

**Usage Guidelines** Use this command to enable Prometheus Pushgateway.

## prometheus pushgateway port

Specify the Prometheus Pushgateway port.

The Pushgateway is an intermediary service which allows you to push metrics from jobs which cannot be scraped.

---

**Command Modes** Exec > Global Configuration (config)

---

**Syntax Description** `prometheus pushgateway port port_number`

**pushgateway port *port\_number***

Specify the port number of the Pushgateway port. *port\_number* must be an integer in the range of 0 to 65535.

Default value: 9091

---

**Usage Guidelines** Use this command to enable Prometheus Pushgateway.

## prometheus recording-rules group

Configures Prometheus record rule group.

---

**Command Modes** Exec > Global Configuration (config)

---

**Syntax Description** `prometheus recording-rules group record_rule_group_name { interval-seconds evaluation_interval }`

**interval-seconds *evaluation\_interval***

Specify the evaluation interval of the rule group in seconds.

Must be an integer.

***record\_rule\_group\_name***

Specify name of the record rule group.

Must be a string.

---

**Usage Guidelines** Use this command to configure Prometheus record rule group.

## prometheus recording-rules group rule

Configures record rule definition.

---

**Command Modes** Exec > Global Configuration (config)

---

**Syntax Description** `prometheus recording-rules group record_rule_group_name rule record_name { expression rule_expression }`

**expression *rule\_expression***

Specify PromQL record rule expression.

Must be a string.

**record *record\_name***

Specify the record name.

Must be a string.

**Usage Guidelines** Use this command to configure record rule definition.

## prometheus recording-rules group rule label

Configures labels to attach to the record rule time series.

**Command Modes** Exec > Global Configuration (config)

**Syntax Description** **prometheus recording-rules group** *record\_rule\_group\_name* **rule** *record\_name* { **label** *label\_name* **value** *label\_value* }

**value *label\_value***

Specify the label value.

Must be a string.

***label\_name***

Specify the label name.

Must be a string.

**Usage Guidelines** Use this command to configure labels to attach to the record rule time series.

## prometheus server-settings

Configures local cluster settings to enable scraping from remote manager.

**Command Modes** Exec > Global Configuration (config)

**Syntax Description** **server-settings** { **external-ip** *ip\_address* | **ssl-key** *ssl\_key\_certificate* | **ssl-crt** *ssl crt\_certificate* | **ssl-ca** *ssl\_certificate\_authority* }

**external-ip *ip\_address***

Specify the external IP address to expose this cluster to remote manager.

Must be an IPv4 address.

-Or-

Must be an IPv6 address.

***ssl-ca ssl\_certificate\_authority***

Specify the SSL certificate authority.

Must be a string.

***ssl-crt ssl crt\_certificate***

Specify the SSL CRT certificate.

Must be a string.

***ssl-key ssl\_key\_certificate***

Specify the SSL key certificate.

Must be a string.

---

**Usage Guidelines**

Use this command to configure local cluster settings to enable scraping from remote manager.



## CHAPTER 12

# SNMP Config Mode Command Reference

- [snmp-trapper](#), on page 149
- [snmp-trapper source-ip-routes](#), on page 150
- [snmp-trapper source-ip-routes source-external-vips](#), on page 150
- [snmp-trapper v2c-target](#), on page 151
- [snmp-trapper v3-target](#), on page 151

## snmp-trapper

Configures the SNMP trapper.

### Command Modes

Exec > Global Configuration

### Syntax Description

```
snmp-trapper enable { false | true } v3-engine-id v3_engine_id
```

#### **enable { false | true }**

Specify to enable or disable the SNMP trapper.

Must be one of the following:

- **false**
- **true**

Default Value: false.

#### **v3-engine-id v3\_engine\_id**

Specify the source engine ID for v3 traps as hex string. For example, 80004f.

Must be a string of 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 50, 52, 54, 56, 58, 60, 62, or 64 characters in the pattern [0-9a-fA-F]\*.

### Usage Guidelines

Use this command to configure the SNMP trapper.

## snmp-trapper source-ip-routes

Enables the binding to source IP for SNMP routing.

**Command Modes** Exec > Global Configuration

**Syntax Description** `snmp-trapper enable { false | true } source-ip-routes { internal-vip vip_address | default-external-vip vip_address }`

### **default-external-vip vip\_address**

Specify the default external Virtual IP address for source IP routing.

Must be a string in the ipv4-address pattern. For information on the ipv4-address pattern, see the Input Pattern Types section.

### **internal-vip vip\_address**

Specify the internal Virtual IP address for source IP routing.

Must be a string in the ipv4-address pattern. For information on the ipv4-address pattern, see the Input Pattern Types section.

**Usage Guidelines** Use this command to enable the binding to source IP for SNMP routing.

## snmp-trapper source-ip-routes source-external-vips

Configures the source external VIP routing by namespace.

**Command Modes** Exec > Global Configuration

**Syntax Description** `snmp-trapper enable { false | true } source-ip-routes source-external-vips { namespace external-vip vip_address }`

### **external-vip vip\_address**

Specify the external Virtual IP address for source IP routing.

Must be a string in the ipv4-address pattern. For information on the ipv4-address pattern, see the Input Pattern Types section.

### **namespace**

Specify the namespace for routing.

Must be a string.

**Usage Guidelines** Use this command to configure the source external Virtual IP address routing by namespace.

## snmp-trapper v2c-target

Configures the list of SNMP v2c trap receivers.

**Command Modes** Exec > Global Configuration

**Syntax Description** `snmp-trapper enable { true } v2c-target { url | community snmp_trap_community | port port_number }`

**community snmp\_trap\_community**

Specify the SNMP trap receiver community.

Must be a string.

Default Value: public.

**port\_number**

Specify the port number of the SNMP trap receiver port.

Must be an integer in the range of 0-65535.

Default Value: 162.

**url**

Specify the SNMP trap receiver hostname or IP address.

Must be a string.

**Usage Guidelines** Use this command to configure the list of SNMP v2c trap receivers.

## snmp-trapper v3-target

Configures the list of SNMP v3 trap receivers.

**Command Modes** Exec > Global Configuration

**Syntax Description** `snmp-trapper enable { true } v3-target { url port port_number user-name user_name auth authentication_protocol auth-key authentication_key priv privacy_protocol priv-key privacy_key }`

**auth-key authentication\_key**

Specify the key to the authentication protocol.

Must be a string of 8-maximum characters.

**auth authentication\_protocol**

Specify the authentication protocol to be used.

Must be one of the following:

- **md5**: HMAC-MD5-96 authentication protocol is used.
- **none**: No authentication is used.
- **sha**: HMAC-SHA-96 authentication protocol is used.

Default Value: none.

**priv-key *privacy\_key***

Specify the privacy key.

Must be a string of 8-maximum characters.

**priv *privacy\_protocol***

Specify the privacy protocol to be used.

Must be one of the following:

- **aes192**: AES-CFB (192 bits) protocol is used.
- **aes256**: AES-CFB (256 bits) protocol is used.
- **aes**: AES-CFB (128 bits) protocol is used.
- **des**: CBC-DES protocol is used.
- **none**: No privacy is used.

Default Value: none.

**user-name *user\_name***

Specify the SNMP trap receiver user name.

Must be a string.

***port\_number***

Specify the port number of the SNMP trap receiver port.

Must be an integer in the range of 0-65535.

Default Value: 162.

***url***

Specify the SNMP trap receiver hostname or IP address.

Must be a string.

---

**Usage Guidelines**

Use this command to configure list of SNMP v3 trap receivers.



# CHAPTER 13

## VES Adapter Config Mode Command Reference

- [ves-adapter](#), on page 153
- [ves-adapter measurement-group](#), on page 154
- [ves-adapter measurement-group measurement](#), on page 154

### ves-adapter

Configures VES Adapter parameters.

#### Command Modes

Exec > Global Configuration

#### Syntax Description

```
ves-adapter enable { false | true } [[[url ves_listener_url | user-name ves_listener_user_name | password ves_listener_password] | measurement-interval measurement_interval] | [measurement-group [name group_name | measurement [name measurement_name] query measurement_query]] | measurement-group measurement_group_name [measurement measurement_name | query measurement_query]]
```

#### **enable { false | true }**

Specify to enable or disable the VES Adapter.

Must be one of the following:

- **false**
- **true**

Default Value: false.

#### **measurement-interval** *measurement\_interval*

Specify the interval to fetch measurements in seconds.

Must be an integer in the range of 1-86400.

Default Value: 300.

#### **password** *ves\_listener\_password*

Specify the VES Listener password.

Must be a string.

**url** *ves\_listener\_url*

Specify the VES Listener URL with path.

Must be a string.

**user-name** *ves\_listener\_user\_name*

Specify the VES Listener user name.

Must be a string.

**Usage Guidelines** Use this command to configure VES Adapter parameters.

## ves-adapter measurement-group

Configures the list of additional measurement groups.

**Command Modes** Exec > Global Configuration

**Syntax Description** `ves-adapter enable { true } measurement-group group_name`

***group\_name***

Specify the additional measurement group name.

Must be a string.

**Usage Guidelines** Use this command to configure the list of additional measurement groups.

## ves-adapter measurement-group measurement

Configures the list of additional measurement queries.

**Command Modes** Exec > Global Configuration

**Syntax Description** `ves-adapter enable { true } measurement-group group_name { measurement measurement_name query measurement_query }`

***query measurement\_query***

Specify the query to execute in Promo QL format.

Must be a string.

***measurement\_name***

Specify the measurement name.

Must be a string.

---

**Usage Guidelines**

Use this command to configure the list of additional measurement queries.

ves-adapter measurement-group measurement