MOSK Operations Guide

version latest

Contents

Copyright notice

Preface

- About this documentation set
- Intended audience
- Technology Preview support scope
- Documentation history
- Conventions

About this documentation set

This documentation provides information on how to deploy and operate a Mirantis OpenStack on Kubernetes (MOSK) environment. The documentation is intended to help operators to understand the core concepts of the product. The documentation provides sufficient information to deploy and operate the solution.

The information provided in this documentation set is being constantly improved and amended based on the feedback and kind requests from the consumers of MOSK.

The following table lists the guides included in the documentation set you are reading:

Guides list

Guide	Purpose
MOSK Reference Architecture	Learn the fundamentals of MOSK reference architecture to appropriately plan your deployment
MOSK Deployment Guide	Deploy an MOSK environment of a preferred configuration using supported deployment profiles tailored to the demands of specific business cases
MOSK Operations Guide	Operate your MOSK environment
MOSK Release notes	Learn about new features and bug fixes in the current MOSK version

The MOSK documentation home page contains references to all guides included in this documentation set. For your convenience, we provide all guides in HTML (default), single-page HTML, PDF, and ePUB formats. To use the preferred format of a guide, select the required option from the Formats menu next to the guide title.

Intended audience

This documentation is intended for engineers who have the basic knowledge of Linux, virtualization and containerization technologies, Kubernetes API and CLI, Helm and Helm charts, Mirantis Kubernetes Engine (MKE), and OpenStack.

Technology Preview support scope

This documentation set includes description of the Technology Preview features. A Technology Preview feature provide early access to upcoming product innovations, allowing customers to experience the functionality and provide feedback during the development process. Technology Preview features may be privately or publicly available and neither are intended for production use. While Mirantis will provide support for such features through official channels, normal Service Level Agreements do not apply. Customers may be supported by Mirantis Customer Support or Mirantis Field Support.

As Mirantis considers making future iterations of Technology Preview features generally available, we will attempt to resolve any issues that customers experience when using these features.

During the development of a Technology Preview feature, additional components may become available to the public for testing. Because Technology Preview features are being under development, Mirantis cannot guarantee the stability of such features. As a result, if you are using Technology Preview features, you may not be able to seamlessly upgrade to subsequent releases of that feature. Mirantis makes no guarantees that Technology Preview features will be graduated to a generally available product release.

The Mirantis Customer Success Organization may create bug reports on behalf of support cases filed by customers. These bug reports will then be forwarded to the Mirantis Product team for possible inclusion in a future release.

Documentation history

The following table contains the released revision of the documentation set you are reading:

Release date	Description
November 05, 2020	Mirantis OpenStack on Kubernetes (MOSK) GA release

Conventions

This documentation set uses the following conventions in the HTML format:

Documentation conventions

Convention	Description
boldface font	Inline CLI tools and commands, titles of the procedures and system response examples, table titles
monospaced font	Files names and paths, Helm charts parameters and their values, names of packages, nodes names and labels, and so on
italic font	Information that distinguishes some concept or term
Links	External links and cross-references, footnotes
Main menu > menu item	GUI elements that include any part of interactive user interface and menu navigation

Superscript	Some extra, brief information
Note The Note block	Messages of a generic meaning that may be useful for the user
Caution! The Caution block	Information that prevents a user from mistakes and undesirable consequences when following the procedures
Warning The Warning block	Messages that include details that can be easily missed, but should not be ignored by the user and are valuable before proceeding
Seealso The See also block	List of references that may be helpful for understanding of some related tools, concepts, and so on
Learn more The Learn more block	Used in the Release Notes to wrap a list of internal references to the reference architecture, deployment and operation procedures specific to a newly implemented product feature

Introduction

This guide outlines the post-deployment Day-2 operations for a Mirantis OpenStack on Kubernetes environment. It describes how to configure and manage the MOSK components, perform different types of cloud verification, and enable additional features depending on your cloud needs. The guide also contains day-to-day maintenance procedures such as how to back up and restore, update and upgrade, or troubleshoot your MOSK cluster.

OpenStack operations

The section covers the management aspects of an OpenStack cluster deployed on Kubernetes.

Add a compute node

This section describes how to add a new compute node to your existing Mirantis OpenStack on Kubernetes deployment.

To add a compute node:

1. Add a bare metal host to the managed cluster with MOSK as described in Mirantis Container Cloud Operations Guide: Add a bare metal host.
2. Create a Kubernetes machine in your cluster as described in Mitantis Container Cloud Operations Guide: Add a machine.

   When adding the machine, specify the node labels as required for an OpenStack compute node:

OpenStack node roles

Node role	Description	Kubernetes labels	Minimal count
OpenStack control plane	Hosts the OpenStack control plane services such as database, messaging, API, schedulers, conductors, L3 and L2 agents.	openstack-control-plane=enabled openstack-gateway=enabled openvswitch=enabled	3
OpenStack compute	Hosts the OpenStack compute services such as libvirt and L2 agents.	openstack-compute-node=enabled openvswitch=enabled (for a deployment with Open vSwitch as a back end for networking)	Varies

Delete a compute node

This section describes how to delete an OpenStack compute node from your MOSK deployment.

To delete a compute node:

1. Migrate all workloads from the node. For more information, follow Nova official documentation: Migrate instances.
2. Ensure that there are no pods running on the node to delete by draining the node as instructed in the Kubernetes official documentation: Safely drain node.
3. Delete the node through the Mirantis Container Cloud web UI as described in Mirantis Container Cloud Operations Guide: Delete a machine.

Add a controller node

This section describes how to add a new control plane node to the existing MOSK deployment.

To add an OpenStack controller node:

1. Add a bare metal host to the managed cluster with MOSK as described in Mirantis Container Cloud Operations Guide: Add a bare metal host using CLI.

   When adding the bare metal host YAML file, specify the following OpenStack control plane node labels for the OpenStack control plane services such as database, messaging, API, schedulers, conductors, L3 and L2 agents:

   - openstack-control-plane=enabled
   - openstack-gateway=enabled
   - openvswitch=enabled
2. Create a Kubernetes machine in your cluster as described in Mirantis Container Cloud Operations Guide: Add a machine using CLI.

   When adding the machine, verify that OpenStack control plane node has the following labels:

   - openstack-control-plane=enabled
   - openstack-gateway=enabled
   - openvswitch=enabled

   > Note
   >
   > Depending on the applications that were colocated on the failed controller node, you may need to specify some additional labels, for example, ceph_role_mgr=true and ceph_role_mon=true . To successfuly replace a failed mon and mgr node, refer to Mirantis Container Cloud Operations Guide: Manage Ceph.

3. Verify that the node is in the Ready state through the Kubernetes API:

   ```
   kubectl get node <NODE-NAME> -o wide | grep Ready
   ```

4. Verify that the node has all required labels described in the previous steps:

   ```
   kubectl get nodes --show-labels
   ```

5. Configure new Octavia health manager resources:

   1. Rerun the octavia-create-resources job:

      ```
      kubectl -n osh-system exec -t <OS-CONTROLLER-POD> -c osdpl osctl-job-rerun octavia-create-resources openstack
      ```

   2. Wait until the Octavia health manager pod on the newly added control plane node appears in the Running state:

      ```
      kubectl -n openstack get pods -o wide | grep <NODE_ID> | grep octavia-health-manager
      ```

Note

If the pod is in the crashloopbackoff state, remove it:

```
kubectl -n openstack delete pod <OCTAVIA-HEALTH-MANAGER-POD-NAME>
```

3. Verify that an OpenStack port for the node has been created and the node is in the Active state:

```
kubectl -n openstack exec -t <KEYSTONE-CLIENT-POD-NAME> openstack port show octavia-health-manager-listen-port-<NODE-NAME>
```

Replace a failed controller node

This section describes how to replace a failed control plane node in your MOSK deployment. The procedure applies to the control plane nodes that are, for example, permanently failed due to a hardware failure and appear in the NotReady state:

```
kubectl get nodes <CONTAINER-CLOUD-NODE-NAME>
```

Example of system response:

```
NAME                        STATUS     ROLES    AGE   VERSION
<CONTAINER-CLOUD-NODE-NAME>   NotReady   <none>   10d   v1.18.8-mirantis-1
```

To replace a failed controller node:

1. Remove the Kubernetes labels from the failed node by editing the .metadata.labels node object:

   ```
   kubectl edit node <CONTAINER-CLOUD-NODE-NAME>
   ```

2. Add the control plane node to your deployment as described in Add a controller node.
3. Identify all stateful applications present on the failed node:

   ```
   node=<CONTAINER-CLOUD-NODE-NAME>
   claims=$(kubectl -n openstack get pv -o jsonpath="{.items[?(@.spec.nodeAffinity.required.nodeSelectorTerms[0].matchExpressions[0].values[0] == '${node}')].spec.claimRef.name}")
   for i in $claims; do echo $i done
   ```

   Example of system response:

   ```
   mysql-data-mariadb-server-2
   openstack-operator-bind-mounts-rfr-openstack-redis-1
   etcd-data-etcd-etcd-0
   ```

4. Reschedule stateful applications pods to healthy controller nodes as described in Reschedule stateful applications.

5. Remove the OpenStack port related to the Octavia health manager pod of the failed node:

```
kubectl -n openstack exec -t <KEYSTONE-CLIENT-POD-NAME> openstack port delete octavia-health-manager-listen-port-<NODE-NAME>
```

Reschedule stateful applications

The rescheduling of stateful applications may be required when replacing a permanently failed node, decomissioning a node, migrating applications to nodes with a more suitable set of hardware, and in several other use cases.

MOSK deployment profiles include the following stateful applications:

- OpenStack database (MariaDB)
- OpenStack coordination (etcd)
- OpenStack Time Series Database back end (Redis)

Each stateful application from the list above has a persistent volume claim (PVC) based on a local persistent volume per pod. Each of control plane nodes has a set of local volumes available. To migrate an application pod to another node, recreate a PVC with the persistent volume from the target node.

> Caution!
>
> A stateful application pod can only be migrated to a node that does not contain other pods of this application.

> Caution!
>
> When a PVC is removed, all data present in the related persistent volume is removed from the node as well.

To reschedule pods to another control plane node:

1. Recreate a PVC on another control plane node:

   > Note
   >
   > Skip this step for etcd as the etcd Helm chart does not support joining of a pod with a clean volume to the cluster.

   1. Select one of the persistent volumes available on the node:

> **Caution!**
>
> A stateful application pod can only be migrated to the node that does not contain other pods of this application.

```
NODE_NAME=<NODE-NAME>
STORAGE_CLASS=$(kubectl -n openstack get osdpl <OSDPL_OBJECT_NAME> -o jsonpath='{.spec.local_volume_storage_class}')
kubectl -n openstack get pv -o json | jq --arg NODE_NAME $NODE_NAME --arg STORAGE_CLASS $STORAGE_CLASS -r '.items[] | select(.spec.nodeAffinity.required.nodeSelectorTerms[0].matchExpressions[0].values[0] == $NODE_NAME and .spec.storageClassName == $STORAGE_CLASS and .status.phase == "Available") | .metadata.name'
```

2. As the new PVC should contain the same parameters as the deleted one except for volumeName, save the old PVC configuration in YAML:

```
kubectl -n <NAMESPACE> get pvc <PVC-NAME> -o yaml > <OLD-PVC>.yaml
```

> Note
>
> <NAMESPACE> is a Kubernetes namespace where the PVC is created. For Redis, specify openstack-redis, for other applications specify openstack.

3. Delete the old PVC:

```
kubectl -n <NAMESPACE> delete pvc <PVC-NAME>
```

> Note
>
> If a PVC has stuck in the terminating state, run kubectl -n openstack edit pvc <PVC-NAME> and remove the finalizers section from metadata of the PVC.

4. Create the PVC with a new persistent volume:

```
cat <<EOF | kubectl apply -f -
    apiVersion: v1
    kind: PersistentVolumeClaim
    metadata:
      name: <PVC-NAME>
      namespace: <NAMESPACE>
    spec:
      accessModes:
      - ReadWriteOnce
      resources:
        requests:
```

```
    storage: <STORAGE-SIZE>
  storageClassName: <STORAGE-CLASS>
  volumeMode: Filesystem
  volumeName: <PV-NAME>
EOF
```

Caution!

<STORAGE-SIZE>, <STORAGE-CLASS>, and <NAMESPACE> should correspond to the storage, storageClassName, and namespace values from the <OLD-PVC>.yaml file with the old PVC configuration.

2. Reschedule the pods:

- For MariaDB:

  1. Remove the pod:

     ```
     kubectl -n openstack delete pod <STATEFULSET-NAME>-<NUMBER>
     ```

  2. Wait until the pod appears in the Ready state.

     When the rescheduling is finalized, the mariadb-server-2 pod joins back to the Galera cluster with a clean MySQL data directory and requests the Galera state transfer from the available nodes.

- For etcd:

  1. Scale down the etcd StatefulSet to 0 replicas:

     ```
     kubectl -n openstack scale sts etcd-etcd --replicas=0
     ```

  2. Remove all PVCs as the etcd Helm chart does not support joining of a pod with a clean volume to the cluster:

     Note

     During this operation no important data can be lost as etcd is used only for coordination.

     ```
     kubectl -n openstack delete pvc -l application=etcd,component=server
     ```

3. Scale the etcd StatefulSet to the initial number of replicas:

```
kubectl -n openstack scale sts etcd-etcd --replicas=<NUMBER-OF-REPLICAS>
```

4. Wait until all etcd pods appear in the Ready state.

- For Redis:

1. Remove the pod:

```
kubectl -n openstack-redis delete pod <STATEFULSET-NAME>-<NUMBER>
```

2. Wait until the pod appears in the Ready state.

Run Tempest tests

The OpenStack Integration Test Suite (Tempest), is a set of integration tests to be run against a live OpenStack cluster. This section instructs you on how to verify the workability of your OpenStack deployment using Tempest.

To verify an OpenStack deployment using Tempest:

1. Configure Tempest as required.

   To change the Tempest run parameters, use the following structure in the OsDpl CR:

```
spec:
  services:
    tempest:
      tempest:
        values:
          conf:
            script: |
              tempest run --config-file /etc/tempest/tempest.conf \\
              --concurrency 4 --blacklist-file /etc/tempest/test-blacklist --smoke
```

   The following example structure from the OsDpl CR will set image:build_timeout to 600 in the tempest.conf file:

```
spec:
  services:
    tempest:
      tempest:
        values:
          conf:
            tempest:
              image:
                build_timeout: 600
```

2. Run Tempest. The OpenStack Tempest is deployed like other OpenStack services in a dedicated openstack-tempest Helm release by adding tempest to spec:features:services in the OSDPL resource.

```
spec:
  features:
    services:
      - tempest
```

3. Wait until Tempest is ready. The Tempest tests are launched by the openstack-tempest-run-tests job. To keep track of the tests execution, run:

```
kubectl -n openstack logs -l application=tempest,component=run-tests
```

4. Get the Tempest results. The Tempest results can be stored in a pvc-tempest PersistentVolumeClaim (PVC). To get them from a PVC, use:

```
# Run pod and mount pvc to it
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: tempest-test-results-pod
  namespace: openstack
spec:
  nodeSelector:
    openstack-control-plane: enabled
  volumes:
    - name: tempest-pvc-storage
      persistentVolumeClaim:
        claimName: pvc-tempest
  containers:
    - name: tempest-pvc-container
      image: ubuntu
      command: ['sh', '-c', 'sleep infinity']
      volumeMounts:
        - mountPath: "/var/lib/tempest/data"
          name: tempest-pvc-storage
EOF
```

5. If required, copy the results locally:

```
kubectl -n openstack cp tempest-test-results-pod:/var/lib/tempest/data/report_file.xml .
```

6. Remove the Tempest test results pod:

```
kubectl -n openstack delete pod tempest-test-results-pod
```

7. To rerun Tempest:
    1. Remove Tempest from the list of enabled services.
    2. Wait until Tempest jobs are removed.
    3. Add Tempest back to the list of the enabled services.

> Seealso
>
> MOSK Reference Architecture: OpenStackDeployment custom resource

Update OpenStack

The update of the OpenStack components is performed during the MOSK managed cluster release update through the Container Cloud web UI as described in Mirantis Container Cloud Operations Guide: Update a managed cluster.

Calculate a maintenance window duration

This section provides the background information on the approximate time spent on operations for pods of different purposes, possible data plane impact during these operations, and the possibility of a parallel pods update. Such data helps the cloud administrators to correctly estimate maintenance windows and impacts on the workloads for your OpenStack deployment.

Maintenance window calculation

Pod name	Pod description	Kubernetes kind	Readiness time	Data plane impact	Parallel update
[*]-api	Contains API services of OpenStack components. Horizontally well scalable.	Deployment	<30s	NO	YES (batches 10% of overall count)
[*]-conductor	Contains proxy service between OpenStack and database.	Deployment	<30s	NO	YES (batches 10% of overall count)
[*]-scheduler	Spreads OpenStack resources between nodes.	Deployment	<30s	NO	YES (batches 10% of overall count)

[*]-worker [*]-engine [*]-volume [*]-backup [*]	Process user requests.	Deployment	<30s	NO	YES (batches 10% of overall count)
nova-compute	Processes user requests, interacts with the data plane services.	DaemonSet	<120s	NO	YES (batches 10% of overall count)
neutron-l3-agent	Creates virtual routers (spawns keepalived processes for the HA routers).	DaemonSet	10-15m (for 100 routers)	YES	NO (one by one)
neutron-openvswitch-agent	Configures tunnels between nodes.	DaemonSet	<120s	NO	YES (batches 10% of overall count)
neutron-dhcp-agent	Configures the DHCP server for the networking service.	DaemonSet	<30s	Partially (only if the downtime exceeds the lease time out.	YES (batches 10% of overall count)
neutron-metadata-agent	Provides metadata information to user workloads (VMs).	DaemonSet	<30s	NO	YES (batches 10% of overall count)
libvirt	Starts the libvirtd communication daemon.	DaemonSet	<30s	NO	YES (batches 10% of overall count)
openvswitch-[*]	Sets up the Open vSwitch datapaths and then operates the switching across each bridge.	DaemonSet	<30s	YES	NO (one by one)
mariadb-[*]	Contains persistent storage (database) for OpenStack deployment.	StatefulSet	<180s	NO	NO (one by one)
memcached-[*]	Contains the memory object caching system.	Deployment	<30s	NO	NO (one by one)

[*]-rabbitmq-[*]	Contains the messaging service for OpenStack.	Stateful Set	<30s	NO	NO (one by one)

Ceph operations

To manage a running Ceph cluster, for example, to add or remove a Ceph OSD, remove a Ceph node, replace a failed physical disk with a Ceph node, or update your Ceph cluster, refer to Mirantis Container Cloud Operations Guide: Manage Ceph.

Before you proceed with any reading or writing operation, first check the cluster status using the ceph tool as described in Mirantis Container Cloud Operations Guide: Verify the Ceph core services.

StackLight operations

The section covers the StackLight management aspects.

View Grafana dashboards

Using the Grafana web UI, you can view the visual representation of the metric graphs based on the time series databases.

This section describes only the OpenStack-related Grafana dashboards. For other dashboards, including the system, Kubernetes, Ceph, and StackLight dashboards, see Mirantis Container Cloud Operations Guide: View Grafana dashboards.

To view the Grafana dashboards:

1. Log in to the Grafana web UI as described in Mirantis Container Cloud Operations Guide: Access StackLight web UIs.
2. From the drop-down list, select the required dashboard to inspect the status and statistics of the corresponding service deployed in Mirantis OpenStack on Kubernetes:

Dashboard	Description
OpenStack - Overview	Priovides general information on OpenStack services resources consumption, API errors, deployed OpenStack compute nodes and block storage usage.
KPI - Downtime	Provides uptime statistics for OpenStack compute instances, including graphs on created VMs status and availability.
KPI - Provisioning	Provides provisioning statistics for OpenStack compute instances, including graphs on VM creation results by day.
Cinder	Provides graphs on the OpenStack Block Storage service health, HTTP API availability, pool capacity and utilization, number of created volumes and snapshots.
Glance	Provides graphs on the OpenStack Image service health, HTTP API availability, number of created images and snapshots.
Gnocchi	Provides panels and graphs on the Gnocchi health and HTTP API availability.
Heat	Provides graphs on the OpenStack Orchestration service health, HTTP API availability and usage.
Ironic	Provides graphs on the OpenStack Bare Metal Provisioning service health, HTTP API availability, provisioned nodes by state and installed ironic-conductor back-end drivers.
Keystone	Provides graphs on the OpenStack Identity service health, HTTP API availability, number of tenants and users by state.
Neutron	Provides graphs on the OpenStack networking service health, HTTP API availability, agents status and usage of Neutron L2 and L3 resources.

NGINX Ingress controller	Monitors the number of requests, response times and statuses, as well as the number of Ingress SSL certificates including expiration time and resources usage.
Nova - Availability Zones	Provides detailed graphs on the OpenStack availability zones and hypervisor usage.
Nova - Hypervisor Overview	Provides a set of single-stat panels presenting resources usage by host.
Nova - Instances	Provides graphs on libvirt Prometheus exporter health and resources usage. Monitors the number of running instances and tasks and allows sorting the metrics by top instances.
Nova - Overview	Provides graphs on the OpenStack compute services (nova-scheduler, nova-conductor, and nova-compute) health, as well as HTTP API availability.
Nova - Tenants	Provides graphs on CPU, RAM, disk throughput, IOPS, and space usage and allocation and allows sorting the metrics by top tenants.
Nova - Users	Provides graphs on CPU, RAM, disk throughput, IOPS, and space usage and allocation and allows sorting the metrics by top users.
Nova - Utilization	Provides detailed graphs on Nova hypervisor resources capacity and consumption.
Memcached	Memcached Prometheus exporter dashboard. Monitors Kubernetes Memcached pods and displays memory usage, hit rate, evicts and reclaims rate, items in cache, network statistics, and commands rate.
MySQL	MySQL Prometheus exporter dashboard. Monitors Kubernetes MySQL pods, resources usage and provides details on current connections and database performance.
RabbitMQ	RabbitMQ Prometheus exporter dashboard. Monitors Kubernetes RabbitMQ pods, resources usage and provides details on cluster utilization and performance.

View Kibana dashboards

Using the Kibana web UI, you can view the visual representation of your OpenStack deployment notifications.

This section describes only the Openstack-related Kibana dashboards. For other dashboards, including the logs and Kubernetes events dashboards, see Mirantis Container Cloud Operations Guide: View Kibana dashboards.

To view the Kibana dashboards:

1. Log in to the Kibana web UI as described in Mirantis Container Cloud Operations Guide: Access StackLight web UIs.
2. Click the required dashboard to inspect the visualizations or perform a search:

Dashboard	Description
Notifications	Provides visualizations on the number of notifications over time per source and severity, host, and breakdowns. Includes search.
Audit	Provides search on audit notifications.

Available StackLight alerts

This section provides an overview of the available predefined OpenStack-related StackLight alerts. To view the alerts, use the Prometheus web UI. To view the firing alerts, use Alertmanager or Alerta web UI.

For other alerts, including the node, Kubernetes, Ceph, and StackLight alerts, see Mirantis Container Cloud Operations Guide: Available StackLight alerts.

Core services

This section describes the alerts available for the core services.

libvirt

This section lists the alerts for the libvirt service.

- LibvirtDown

LibvirtDown

Severity	Critical
Summary	Failure to gather libvirt metrics.
Description	The libvirt metric exporter fails to gather metrics on the {{ $labels.host }} host for 2 minutes.

MariaDB

This section lists the alerts for the MariaDB service.

- MariadbGaleraDonorFallingBehind
- MariadbGaleraNotReady
- MariadbGaleraOutOfSync
- MariadbInnodbLogWaits
- MariadbInnodbReplicationFallenBehind
- MariadbTableLockWaitHigh

MariadbGaleraDonorFallingBehind

Severity	Warning
Summary	MariaDB cluster donor node is falling behind.
Description	The MariaDB cluster node is falling behind (the queue size is {{ $value }}).

MariadbGaleraNotReady

Severity	Major
Summary	MariaDB cluster is not ready.
Description	The MariaDB cluster is not ready to accept queries.

MariadbGaleraOutOfSync

Severity	Minor
Summary	MariaDB cluster node is out of sync.
Description	The MariaDB cluster node is not in sync ({{ $value }} != 4).

MariadbInnodbLogWaits

Severity	Warning
Summary	MariaDB InnoDB log writes are stalling.
Description	The MariaDB InnoDB logs are waiting for the disk at a rate of {{ $value }} per second.

MariadbInnodbReplicationFallenBehind

Severity	Warning
Summary	MariaDB InnoDB replication is lagging.
Description	The MariaDB InnoDB replication has fallen behind and is not recovering.

MariadbTableLockWaitHigh

Severity	Minor
Summary	MariaDB table lock waits are high.
Description	MariaDB has {{ $value }}% of table lock waits.

Memcached

This section lists the alerts for the Memcached service.

- MemcachedServiceDown
- MemcachedConnectionsNoneMinor
- MemcachedConnectionsNoneMajor
- MemcachedEvictionsLimit

MemcachedServiceDown

Severity	Minor
Summary	The Memcached service is down.
Description	The Memcached service in the {{ $labels.namespace }} namespace is down.

MemcachedConnectionsNoneMinor

Severity	Minor
Summary	Memcached has no open connections.
Description	The Memcached database in the {{ $labels.namespace }} namespace has no open connections.

MemcachedConnectionsNoneMajor

Severity	Major
Summary	Memcached has no open connections on all nodes.
Description	The Memcached database has no open connections on all nodes.

MemcachedEvictionsLimit

Severity	Warning
Summary	Memcached has 10 evictions.
Description	The average of {{ $value }} evictions in the Memcached database occurred in the {{ $labels.namespace }} namespace during the last minute.

SSL certificates

This section describes the alerts for the OpenStack SSL certificates.

- OpenstackSSLCertExpirationMajor
- OpenstackSSLCertExpirationWarning
- OpenstackSSLProbesFailing

OpenstackSSLCertExpirationMajor

Severity	Major

Summary	SSL certificate for an OpenStack service expires in 10 days.
Description	The SSL certificate for the Openstack {{ $labels.service }} service endpoint {{ $labels.instance }} expires in 10 days.

OpenstackSSLCertExpirationWarning

Severity	Warning
Summary	SSL certificate for an OpenStack service expires in 30 days.
Description	The SSL certificate for the OpenStack {{ $labels.service }} service endpoint {{ $labels.instance }} expires in 30 days.

OpenstackSSLProbesFailing

Severity	Critical
Summary	SSL certificate probes for an OpenStack service are failing.
Description	The SSL certificate probes for the OpenStack {{ $labels.service }} service endpoint {{ $labels.instance }} are failing.

RabbitMQ

This section lists the alerts for the RabbitMQ service.

- RabbitMQNetworkPartitionsDetected
- RabbitMQDown
- RabbitMQFileDescriptorUsageWarning
- RabbitMQNodeDiskFreeAlarm
- RabbitMQNodeMemoryAlarm

RabbitMQNetworkPartitionsDetected

Severity	Warning

Summary	RabbitMQ network partitions detected.
Description	The RabbitMQ server {{ $labels.node }} in the {{ $labels.namespace }} namespace detected {{ $value }} network partitions.

RabbitMQDown

Severity	Critical
Summary	RabbitMQ is down.
Description	The RabbitMQ server monitored by {{ $labels.service }} in the {{ $labels.namespace }} namespace is down for the last 2 minutes.

RabbitMQFileDescriptorUsageWarning

Severity	Warning
Summary	RabbitMQ file descriptors consumption is high for the last 10 minutes.
Description	The RabbitMQ server {{ $labels.node }} in the {{ $labels.namespace }} namespace uses {{ $value }}% of file descriptors.

RabbitMQNodeDiskFreeAlarm

Severity	Warning
Summary	RabbitMQ disk space consumption is high.
Description	The RabbitMQ server {{ $labels.node }} in the {{ $labels.namespace }} namespace has low free disk space available.

RabbitMQNodeMemoryAlarm

Severity	Minor
Summary	RabbitMQ memory consumption is high.
Description	The RabbitMQ server {{ $labels.node }} in the {{ $labels.namespace }} namespace has low free memory.

OpenStack

This section describes the alerts available for the OpenStack services.

OpenStack services API

This section describes the alerts for the OpenStack services API.

- OpenstackServiceApiDown
- OpenstackServiceApiOutage

OpenstackServiceApiDown

Severity	Critical
Summary	The {{ $labels.service_name }} endpoint is not accessible.
Description	The OpenStack service {{ $labels.service_name }} {{ $labels.interface }} API is not accessible for the {{ $labels.endpoint_id }} endpoint.

OpenstackServiceApiOutage

Severity	Critical
Summary	OpenStack service {{ $labels.service_name }} API outage.
Description	The OpenStack service API is not accessible for all available {{ $labels.service_name }} endpoints in the OpenStack service catalog.

Cinder

This section lists the alerts for Cinder.

- CinderServiceDown
- CinderServiceOutage

- CinderServicesDownMajor
- CinderServicesDownMinor

CinderServiceDown

Severity	Minor
Summary	The {{ $labels.binary }} service is down.
Description	The {{ $labels.binary }} service on the {{ $labels.hostname }} node is down.

CinderServiceOutage

Severity	Critical
Summary	The {{ $labels.binary }} service outage.
Description	All {{ $labels.binary }} services are down.

CinderServicesDownMajor

Severity	Major
Summary	60% of {{ $labels.binary }} services are down.
Description	{{ $value }} {{ $labels.binary }} services (>= 60%) are down.

CinderServicesDownMinor

Severity	Minor

Summary	30% of {{ $labels.binary }} services are down.
Description	{{ $value }} {{ $labels.binary }} services (>= 30%) are down.

Ironic

This section lists the alerts for Ironic.

- IronicDriverMissing

IronicDriverMissing

Severity	Major
Summary	The ironic-conductor {{ $labels.driver }} back-end driver is missing.
Description	The {{ $labels.driver }} back-end driver of the ironic-conductor service is missing on the {{ $value }} node(s).

Neutron

This section lists the alerts for Neutron.

- NeutronAgentDown
- NeutronAgentsDownMajor
- NeutronAgentsDownMinor
- NeutronAgentsOutage

NeutronAgentDown

Severity	Minor
Summary	The {{ $labels.binary }} agent is down.
Description	The {{ $labels.binary }} agent on the {{ $labels.hostname }} node is down.

NeutronAgentsDownMajor

Severity	Major
Summary	60% of {{ $labels.binary }} agents are down.
Description	{{ $value }} {{ $labels.binary }} agents (>= 60%) are down.

NeutronAgentsDownMinor

Severity	Minor
Summary	30% of {{ $labels.binary }} agents are down.
Description	{{ $value }} {{ $labels.binary }} agents (>= 30%) are down.

NeutronAgentsOutage

Severity	Critical
Summary	{{ $labels.binary }} agents outage.
Description	All {{ $labels.binary }} agents are down.

Nova

This section lists the alerts for Nova.

- NovaComputeServicesDownMajor
- NovaComputeServicesDownMinor
- NovaServiceDown
- NovaServiceOutage
- NovaServicesDownMajor
- NovaServicesDownMinor

NovaComputeServicesDownMajor

Severity	Major

Summary	More than 50% of nova-compute services are down.
Description	More than 50% of nova-compute services are down.

NovaComputeServicesDownMinor

Severity	Minor
Summary	More than 25% of nova-compute services are down.
Description	More than 25% of nova-compute services are down.

NovaServiceDown

Severity	Minor
Summary	The {{ $labels.binary }} service is down.
Description	The {{ $labels.binary }} service on the {{ $labels.hostname }} node is down.

NovaServiceOutage

Severity	Critical
Summary	The {{ $labels.binary }} service outage.
Description	All {{ $labels.binary }} services are down.

NovaServicesDownMajor

Severity	Major
Summary	More than 60% of {{ $labels.binary }} services are down.
Description	More than 60% of {{ $labels.binary }} services are down.

NovaServicesDownMinor

Severity	Minor
Summary	30% of {{ $labels.binary }} services are down.
Description	More than 30% of {{ $labels.binary }} services are down.

Configure StackLight

This section describes how to configure StackLight in your Mirantis OpenStack on Kubernetes deployment and includes the description of OpenStack-related StackLight parameters and their verification. For other available configuration keys and their configuration verification, see Mirantis Container Cloud Operations Guide: Manage StackLight.

Configure StackLight

This section describes the StackLight configuration workflow.

To configure StackLight:

1. Obtain kubeconfig of the Mirantis Container Cloud management cluster and open the Cluster object manifest of the managed cluster for editing as described in the steps 1-7 in Mirantis Container Cloud Operations Guide: Configure StackLight.

2. In the following section of the opened manifest, configure the StackLight parameters as required:

   - For the OpenStack-based parameters, see StackLight configuration parameters.
   - For other parameters, see Mirantis Container Cloud Operations Guide: StackLight configuration parameters.

```
spec:
  providerSpec:
    value:
      helmReleases:
      - name: stacklight
        values:
```

3. Verify StackLight configuration depending on the modified parameters:

   - For OpenStack-related parameters, see Verify StackLight after configuration.
   - For other parameters, see Mirantis Container Cloud Operations Guide: Verify StackLight after configuration.

StackLight configuration parameters

This section describes the OpenStack-related StackLight configuration keys that you can specify in the values section to change StackLight settings as required. For other available configuration keys, see Mirantis Container Cloud Operations Guide: StackLight configuration parameters.

Prior to making any changes to StackLight configuration, perform the steps described in Configure StackLight. After changing StackLight configuration, verify the changes as described in Verify StackLight after configuration.

- OpenStack
- Ironic
- RabbitMQ
- Telegraf
- SSL certificates

OpenStack

Key	Description	Example values

openstack.enabled (bool)	Enables OpenStack monitoring. Set to true by default.	true or false
openstack.namespace (string)	Defines the namespace within which the OpenStack virtualized control plane is installed. Set to openstack by default.	openstack

---

Ironic

Key	Description	Example values

openstack.ironic.enabled (bool)	Enables Ironic monitoring. Set to false by default.	true or false

RabbitMQ

Key	Description	Example values

openstack.rabbitmq.credentialsConfig (map)	Defines the RabbitMQ credentials to use if credentials discovery is disabled or some required parameters were not found during the discovery.	credentialsConfig:
		  username: "stacklight"
		  password: "stacklight"
		  host: "rabbitmq.openstack.svc"
		  queue: "notifications"
		  vhost: "openstack"

openstack.rabbitmq.credentialsDiscovery (map)	Enables the credentials discovery to obtain the username and password from the secret object.	credentialsDiscovery:
		  enabled: true
		  namespace: openstack
		  secretName: os-rabbitmq-user-credentials

Telegraf

Key	Description	Example values

openstack.telegraf.credentialsConfig (map)	Specifies the OpenStack credentials to use if the credentials discovery is disabled or some required parameters were not found during the discovery.	credentialsConfig: identityEndpoint: "" # "http://keystone-api.openstack.svc:5000/v3" domain: "" # "default" password: "" # "workshop" project: "" # "admin" region: "" # "RegionOne" username: "" # "admin"

openstack.telegraf.credentialsDiscovery (map)	Enables the credentials discovery to obtain all required parameters from the secret object.	credentialsDiscovery:
		  enabled: true
		  namespace: openstack
		  secretName: keystone-keystone-admin

openstack.telegraf.interval (string)	Specifies the interval of metrics gathering from the OpenStack API. Set to 1m by default.	1m, 3m

SSL certificates

Key	Description	Example values

openstack.externalFQDN (string)	External FQDN used to communicate with OpenStack services. Used for certificates monitoring.	https://os.ssl.mirantis.net/

Verify StackLight after configuration

This section describes how to verify StackLight after configuring its OpenStack-related parameters as described in Configure StackLight and StackLight configuration parameters. Perform the verification procedure for a particular modified StackLight key.

To verify StackLight after configuring other parameters, see Mirantis Container Cloud Operations Guide: Verify StackLight after configuration.

To verify StackLight after configuration:

Key	Verification procedure
• openstack.enabled • openstack.namespace	1. In the Grafana web UI, verify that the OpenStack dashboards are present and not empty. 2. In the Prometheus web UI, click Alerts and verify that the OpenStack alerts are present in the list of alerts.
openstack.ironic.enabled	1. In the Grafana web UI, verify that the Ironic dashboard is present and not empty. 2. In the Prometheus web UI, click Alerts and verify that the Ironic* alerts are present in the list of alerts.
• openstack.rabbitmq.credentialsConfig • openstack.rabbitmq.credentialsDiscovery	In the Kibana web UI, click Discover and verify that the audit-* and notifications-* indexes contain documents.

• openstack.telegraf.credentialsConfig • openstack.telegraf.credentialsDiscovery • openstack.telegraf.interval	In the Grafana web UI, verify that the OpenStack dashboards are present and not empty.
openstack.externalFQDN	1. In the Prometheus web UI, navigate to Status > Targets. 2. Verify that the blackbox-external-endpoint target contains the configured domains (URLs).

Tungsten Fabric operations

The section covers the management aspects of a Tungsten Fabric cluster deployed on Kubernetes.

> Caution!
>
> Before you proceed with the Tungsten Fabric management, read through MOSK Reference Architecture: Tungsten Fabric known limitations

Update Tungsten Fabric

The Tungsten Fabric cluster update is performed during the MOSK managed cluster release update through the Container Cloud web UI as described in Mirantis Container Cloud Operations Guide: Update a managed cluster.

Replace a failed TF controller node

If one of the Tungsten Fabric (TF) controller nodes has failed, follow this procedure to replace it with a new node.

To replace a TF controller node:

> Note
>
> Pods that belong to the failed node can stay in the Terminating state.

1. Delete the failed TF controller node from the Kubernetes cluster:

   ```
   kubectl delete node <FAILED-TF-CONTROLLER-NODE-NAME>
   ```

   > Note
   >
   > Once the failed node has been removed from the cluster, all pods that hanged in the Terminating state should be removed.

2. Assign the TF labels for the new control plane node as per the table below using the following command:

   ```
   kubectl label node <NODE-NAME> <LABEL-KEY=LABEL-VALUE> ...
   ```

Tungsten Fabric (TF) node roles

Node role	Description	Kubernetes labels	Minimal count
TF control plane	Hosts the TF control plane services such as database, messaging, api, svc, config.	tfconfig=enabled tfcontrol=enabled tfwebui=enabled tfconfigdb=enabled	3
TF analytics	Hosts the TF analytics services.	tfanalytics=enabled tfanalyticsdb=enabled	3
TF vRouter	Hosts the TF vRouter module and vRouter agent.	tfvrouter=enabled	Varies

Note

TF supports only Kubernetes OpenStack workloads. Therefore, you should label OpenStack compute nodes with the tfvrouter=enabled label.

Note

Do not specify the openstack-gateway=enabled and openvswitch=enabled labels for the MOSK deployments with TF as a networking back end for OpenStack.

3. Once you label the new Kubernetes node, new pods start scheduling on the node. Though, pods that use Persistent Volume Claims are stuck in the Pending state as their volume claims stay bounded to the local volumes from the deleted node. To resolve the issue:

    1. Delete the PersistentVolumeClaim (PVC) bounded to the local volume from the failed node:

        ```
        kubectl -n tf delete pvc <PVC-BOUNDED-TO-NON-EXISTING-VOLUME>
        ```

        Note

        Clustered services that use PVC, such as Cassandra, Kafka, and ZooKeeper, start the replication process when new pods move to the Ready state.

    2. Delete the pod that is using the removed PVC:

```
kubectl -n tf delete pod <POD-NAME>
```

4. Verify that the pods have successfully started on the replaced controller node and stay in the Ready state.

Limitations

The section covers the limitations of Mirantis OpenStack on Kubernetes.

[3544] Due to a community issue, Kubernetes pods may occasionally not be rescheduled on the nodes that are in the NotReady state. As a workaround, manually reschedule the pods from the node in the NotReady state using the kubectl drain --ignore-daemonsets --force <node-uuid> command.