MCP Q2`18 Release Notes

version q2-18
Copyright notice

2020 Mirantis, Inc. All rights reserved.

This product is protected by U.S. and international copyright and intellectual property laws. No part of this publication may be reproduced in any written, electronic, recording, or photocopying form without written permission of Mirantis, Inc.

Mirantis, Inc. reserves the right to modify the content of this document at any time without prior notice. Functionality described in the document may not be available at the moment. The document contains the latest information at the time of publication.

Mirantis, Inc. and the Mirantis Logo are trademarks of Mirantis, Inc. and/or its affiliates in the United States and other countries. Third party trademarks, service marks, and names mentioned in this document are the properties of their respective owners.
What’s new
The latest MCP versions introduce the following features and enhancements.

MCP DriveTrain

- Updates flow
- Saltstack v2017.7
- Offline mirror VM management with Salt
- Cloud verification pipelines
- Standard cluster configurations for OpenStack in Model Designer

Updates flow
Implemented an easy update flow of the MCP to a certain MCP release version through DriveTrain using the Deploy - update cloud pipeline. The pipeline enables you to update software release artifacts of an MCP deployment, such as packages, images, and Git repositories to higher minor versions.

Learn more
- Upgrade MCP to a newer release version

Saltstack v2017.7
Upgraded SaltStack to version 2017.7 and introduced the capability to upgrade SaltStack from version 2016.3 to 2017.7. An automated upgrade procedure is available for the MCP release versions starting from the Build ID 2018.8.0.
If your MCP cluster Build ID is 2018.4.0 or earlier, a manual SaltStack upgrade procedure is available within the scope of the MCP release version update. If the version of your MCP cluster is not tagged with a Build ID, contact Mirantis Support for further details.

Learn more
- Upgrade MCP to a newer release version

Offline mirror VM management with Salt
OFFLINE DEPLOYMENT
Introduced the ability to connect and manage the offline mirror APT node through the Salt Master node as any other infrastructure VM on your MCP deployment. The new functionality enables you to comply with the infrastructure management security best practices.

You can connect the offline mirror VM to Salt by configuring the Reclass model of your deployment as described in our documentation. The Salt minion on the offline image VM will be adjusted automatically on boot during the APT VM provisioning.

Learn more

- MCP Deployment Guide: Enable the management of the APT node through the Salt Master node

Cloud verification pipelines

Implemented the Cloud Verification Pipelines (CVP) tooling that enables the deployment, QA, support engineers, and cloud operators to perform functional as well as non-functional types of testing of cloud environments through DriveTrain.

CVP can be applied to the newly built deployments to verify the functionality, performance, and fault tolerance of cloud environments as well as to the existing deployments to verify their functionality and performance before and after any changes.

The CVP include the following pipelines:

- CVP - Sanity checks
  Sanity testing to verify the cloud platform infrastructure components

- CVP - Functional tests
  OpenStack functional integration testing (Tempest)

- CVP - Performance tests
  OpenStack Rally-based baseline performance testing

- CVP - HA tests
  OpenStack high availability testing

- CVP - StackLight tests
  StackLight LMA basic verification

Learn more

- MCP Operations Guide: Cloud verification

Standard cluster configurations for OpenStack in Model Designer
Introduced the capability to create predefined recommended setups for the OpenStack-based MCP clouds of different sizes using the Model Designer, from minimal and compact clouds to the large ones (200+ compute nodes). It reduces manual intervention from deployment engineers and makes the deployment process faster. The resulting cluster models may still require manual changes to meet customer requirements and the specifics of an environment.

Learn more

- MCP Deployment Guide: Create a deployment metadata model using the Model Designer UI

## Security improvements

- Audit system integration
- Tunable SSL configuration for Apache
- Tunable SSL configuration for NGINX
- TLS encryption and authentication for libvirtd
- Security warning on logon to the MCP VCP nodes and Horizon

### Audit system integration

**MCP INFRASTRUCTURE**

Implemented the audit system integration to MCP. The Linux Audit system enables the system administrator to track security-relevant events by creating an audit trail, which is a log for every action on the server. More specifically, based on the pre-configured rules, the audit system creates log entries that record system calls. By monitoring the events happening on your system, you can reveal violations of system security policies and adjust the set of audit rules to prevent further misuse or unauthorized activities within the system.

Examples of what you can track with the audit system:

- Access to files or directories
- Changing of time settings
- Mount and umount operations
- Loading and unloading kernel modules
- Changing and setting of hostnames
- Changing permissions on a file or directory
- Network activity
- Others
The pre-configured list of the audit rules is conformed with the CIS recommendations. Generally, the configuration of the auditd service in compliance with CIS in your MCP deployment is as simple as:

1. Include the system.auditd.server.ciscat class into your Reclass model.
2. Apply changes using the auditd formula.

Learn more
MCP Deployment guide: Enable the audit system

Tunable SSL configuration for Apache

Extended security settings to configure Apache. More specifically, the implemented enhancements include:

- **EXTENDED** The Normal SSL configuration mode
  In addition to generic SSL parameters like SSLEngine, SSLCertificateFile, SSLCertificateKeyFile, SSLCertificateChainFile, the following parameters are implemented:
  - SSLCipherSuite
  - SSLProtocol

- **EXTENDED** The Secure SSL configuration mode
  In addition to the Normal mode, the Secure mode has a wider list of values for SSLCipherSuite parameter and additional SSLHonorCipherOrder parameter.

- **NEW** The Manual SSL configuration mode
  The Manual mode includes all generic SSL parameters listed for the Normal and Secure modes. All the rest of available Apache parameters are added to the apache Salt formula. Most of the parameters are applied on the vhost level, however, some of them are defined on the server level only. Once defined in a pillar, parameters can be applied to all virtual sites simultaneously or per site.

Learn more
- Apache Formula readme file for the sample pillar configuration
- Apache Module mod_ssl official documentation for the full list of the Apache SSL parameters
Tunable SSL configuration for NGINX

Extended security settings to configure NGINX. More specifically, the implemented enhancements include:

- **EXTENDED** The Normal SSL configuration mode
  
  In addition to generic SSL parameters like ssi, ssl, ssl_session_cache, ssl_session_timeout, ssl_certificate_key, and ssl_certificate, the following parameters are implemented:
  
  - ssl_protocols
  - ssl_ciphers
  - ssl_prefer_server_ciphers
  - ssl_ecdh_curve

- **EXTENDED** The Secure SSL configuration mode
  
  In addition to the Normal mode, some parameters of the Secure mode have a wider list of values and the following new parameters are introduced:
  
  - ssl_dhparam
  - ssl_stapling
  - ssl_stapling_verify

- **NEW** Manual SSL configuration mode

  The Manual mode includes all generic SSL parameters listed for the Normal and Secure modes. All the rest of available NGINX parameters are added to the nginx Salt formula. Once defined in a pillar, parameters can be applied to all virtual sites simultaneously or per site.

Learn more

- [NGINX Formula readme file](#) for the sample pillar configuration
- [NGINX Module ngx_http_ssl_module official documentation](#) for the full list of the NGINX SSL parameters

TLS encryption and authentication for libvirtd

Implemented libvirt control channel and live migration over TLS. By protecting libvirt with TLS, you prevent your cloud workloads from security compromise. The attacker without an appropriate TLS certificate will not be able to connect to libvirtd and affect its operation.
The general workflow of the enablement of libvirt control channel and live migration over TLS in your MCP deployment is as follows:

1. Include the system.nova.compute.libvirt.ssl class into your Reclass model.
2. Apply changes using the nova state.

Security warning on logon to the MCP VCP nodes and Horizon

**MCP OPENSTACK**

Added a disclaimer on SSH and interactive logon to the MCP VCP nodes and logon to Horizon configured with the Mirantis Horizon theme. The disclaimer states that an unauthorized access to or misuse of a computer system is prohibited under the Computer Misuse Act 1990. The act is designed to protect computer users against wilful attacks and theft of information. The act makes it an offense to access or even attempt to access a computer system without the appropriate authorization. Therefore, if a hacker makes even unsuccessful attempts to get into a system, they can be prosecuted using this law.

**StackLight LMA**

- Alerta
- Gainsight integration
- OpenContrail
- Prometheus long-term storage for Grafana
- Alerts rationalization
- Dashboards rationalization

**Alerta**

Implemented the Alerta service as part of the monitoring system in StackLight LMA. Alerta receives the alerts from Alertmanager, combines, deduplicates, and then represents them
through a web UI. Using the Alerta web UI, you can manage the received alerts, including filtering and viewing the most recent ones.

Learn more

- Official Alerta documentation
- MCP Reference Architecture: StackLight LMA components
- MCP Operations Guide: Install Alerta on an existing StackLight LMA deployment

Gainsight integration

Added the possibility to integrate StackLight LMA with Gainsight, a customer relationship management (CRM) tool and extension for Salesforce. Gainsight integration service enables you to query Prometheus for particular metrics data on a selected time frame and send the data in a single CSV file to Gainsight. Mirantis uses the collected data for further analysis to improve the quality of customer support.

Learn more

- MCP Reference Architecture: StackLight LMA components
- MCP Operations Guide: Enable Gainsight integration on an existing StackLight LMA deployment
- Known issue

OpenContrail

Updated StackLight LMA to support OpenContrail version 4.0, rationalized the OpenContrail alerts and dashboards.

Learn more

- MCP Operations Guide: StackLight LMA alerts
- MCP Operations Guide: OpenContrail dashboards

Prometheus long-term storage for Grafana

Enabled Grafana to use the data source of the long-term storage Prometheus to provide access to 180 days of historical data.
Learn more

- MCP Reference Architecture: StackLight LMA components
- MCP Operations Guide: Enable Grafana to use the Prometheus long-term storage data source

Alerts rationalization

- Rationalized StackLight LMA alerts by validating the alerts usability and definitions as well as by standardizing the alerts descriptions and severities.
- Extended StackLight LMA by implementing the monitoring of additional services, SSH and cron.
For a comprehensive list of available StackLight LMA alerts, see MCP Operations Guide: StackLight LMA alerts.

Dashboards rationalization

Rationalized StackLight LMA Grafana dashboards by improving the dashboards usability, adding, splitting and removing the unnecessary ones.

Learn more

MCP Operations Guide: View Grafana dashboards

Kubernetes

- Kubernetes 1.10.4
- Kubernetes 1.10 with OpenContrail 4.0
- Kubernetes version upgrade
- Virtlet 1.1.2
- MetalLB support
- Edge Cloud MVP

Kubernetes 1.10.4

Added support for the upstream Kubernetes version 1.10.4. For the list of enhancements and bug fixes, see the Kubernetes 1.10.4 release notes.
Kubernetes 1.10 with OpenContrail 4.0

Added the capability to deploy a Kubernetes 1.10 cluster with the containerized OpenContrail version 4.0. OpenContrail integrates with Kubernetes by providing different isolation modes for virtual machines, pods, and bare metal workloads. The OpenContrail cluster runs in pods on the Kubernetes Master nodes and the vRouter service runs on the Kubernetes Nodes.

Caution!

OpenContrail with Virtlet is still available as technical preview. Use such configuration for testing and evaluation purposes only.

Learn more

- MCP Reference Architecture: OpenContrail 4.0 cluster overview
- MCP Deployment Guide: Deploy OpenContrail 4.0 for Kubernetes

Kubernetes version upgrade

Implemented the Kubernetes upgrade procedure for an easy upgrade or update of your existing Kubernetes cluster to the latest supported version. You can use either the dedicated Jenkins pipeline to upgrade automatically or perform the manual steps as required. The Kubernetes update procedure is the same as the upgrade one.

Learn more

MCP Operations Guide: Update or upgrade Kubernetes

Virtlet 1.1.2

Upgraded Virtlet to version 1.1.2. For the list of enhancements and bug fixes as compared to version 1.0.0 supported in the previous MCP release, see: the Virtlet releases page.

Caution!

Virtlet with OpenContrail is still available as technical preview. Use such configuration for testing and evaluation purposes only.
MetalLB support

Implemented support for the MetalLB Kubernetes add-on that provides a network load balancer for bare metal Calico-based Kubernetes clusters using standard routing protocols. In MCP, MetalLB supports only the layer-2 mode.

Learn more

- MCP Reference Architecture: MetalLB support
- MCP Deployment Guide: Enable the MetalLB support

Edge Cloud MVP

TECHNICAL PREVIEW

Implemented a new experimental deployment model for the Edge Cloud minimum viable product (MVP) reference architecture. Edge Cloud provides a reduced footprint deployment with a minimal three controller nodes configuration.

Edge Cloud is based on the Kubernetes with Calico architecture that includes Virtlet and the CNI Genie plugin that enables the Flannel CNI plugin support.

Networking in Edge cloud differs from the default Kubernetes configuration to accommodate necessary edge cloud networking requirements.

This feature is available as technical preview only.

Learn more

- MCP Deployment Guide: Deploy Edge Cloud MVP
- MCP Reference Architecture: Plan a Kubernetes cluster

OpenContrail 4.0

MCP introduces the OpenContrail version 4.0 that has several OpenContrail subsystems delivered as containers to reduce the complexity of the OpenContrail deployment and to group the related OpenContrail components.

In the OpenStack deployments, the OpenContrail services run as the analytics, analyticsdb, and controller fat Docker containers managed by docker-compose with the exception of contrail-vrouter-agent running on the compute nodes as a non-containerized service. In the Kubernetes deployments, these containers run as pods managed by the Kubernetes API.
You can deploy OpenContrail 4.0 with OpenStack Pike using the Jenkins pipeline or manually. You can also upgrade OpenContrail 3.2 to 4.0 with Pike for your existing Ocata-based OpenStack deployments.

In Build ID 2018.8.0, OpenContrail 4.0 for Kubernetes is available as technical preview. Use such configuration for testing and evaluation purposes only. For production deployments, use Build ID 2018.8.0-milestone1 where the OpenContrail 4.0 with Kubernetes 1.10 functionality is fully supported.

Caution!
OpenContrail with Virtlet is still available as technical preview. Use such configuration for testing and evaluation purposes only.

Learn more

- MCP Reference Architecture: OpenContrail 4.0 cluster overview
- MCP Deployment Guide: Deploy OpenContrail
- MCP Operations Guide: Upgrade OpenContrail from 3.2 to 4.0

Keycloak identity and authorization

Added support for Keycloak, the identity and access management solution that provides a single entry point for MCP deployments. Using the Keycloak single sign-on functionality you can authenticate to MCP services with Keycloak rather than with individual services, including the ones that do not have a built-in authentication mechanism. Keycloak supports SAML/OAuth2 and OpenID Connect authentication methods.

Warning
Keycloak identity and access management solution is available as technical preview only. Use such configuration for testing and evaluation purposes only.

Note
Currently, the integration with OpenStack and StackLight LMA can only be performed manually. Contact Mirantis Support for further details.
## Learn more

- [About Keycloak](#)
- [MCP Reference Architecture: Plan Keycloak identity and authorization](#)
- [MCP Deployment Guide: Deploy Keycloak](#)
- [MCP Operations Guide: Keycloak operations](#)
Release artifacts

The milestone MCP release artifacts are tagged with the 2018.8.0-milestone1 release version tag including APT repository snapshots, Git repository tags, and Docker image versions. As compared to the GA 2018.8.0 version, the release artifacts tagged with the 2018.8.0-milestone1 Build ID add support for the OpenContrail 4.0 with Kubernetes 1.10.

The combination of versions of MCP components that can be installed using the artifacts tagged with the 2018.8.0 or 2018.8.0-milestone1 Release Version tags are listed in Major components versions. These versions combinations within one Build ID have passed integration testing and are considered stable and working, with the known issues captured in Known issues.

Note

To view the list of software packages used in MCP and their respective license information, download MCP 2018.8.0 Encryption and Licensing. This list is the same for 2018.8.0 and 2018.8.0-milestone1 Build IDs, since only Salt formulas and Reclass model were updated in 2018.8.0-milestone1 as compared to 2018.8.0.

MCP release artifacts

<table>
<thead>
<tr>
<th>Type</th>
<th>Artifact</th>
<th>Path for Build ID 2018.8.0</th>
<th>Path for Build ID 2018.8.0-milestone1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td>Source URL</td>
<td>Source URL</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>QCoW images</td>
<td><a href="http://images.mirantis.com/cfg01-day01-2018.8.0.qcow2">http://images.mirantis.com/cfg01-day01-2018.8.0.qcow2</a></td>
<td><a href="http://images.mirantis.com/cfg01-day01-2018.8.0-milestone1.qcow2.md5">http://images.mirantis.com/cfg01-day01-2018.8.0-milestone1.qcow2.md5</a></td>
<td></td>
</tr>
<tr>
<td>MCP apt01 offline</td>
<td><a href="http://images.mirantis.com/mcp-offline-image-2018.8.0.qcow2">http://images.mirantis.com/mcp-offline-image-2018.8.0.qcow2</a></td>
<td>n/a (The milestone release does not provide for the offline deployment use case.)</td>
<td></td>
</tr>
<tr>
<td>Package</td>
<td>Deb Source</td>
<td>Deb Source</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td></td>
</tr>
</tbody>
</table>

©2020, Mirantis Inc.
| **Upstream Ubuntu system packages** | deb https://mirror.mirantis.com/2018.8.0/ubuntu/ xenial  
deb https://mirror.mirantis.com/2018.8.0/ubuntu/ xenial-updates  
deb https://mirror.mirantis.com/2018.8.0-milestone1/ubuntu/ xenial-updates  
| **MCP Git repositories** | Jenkins pipeline library for MCP operations https://github.com/Mirantis/mk-pipelines/ tag 2018.8.0  
https://github.com/Mirantis/pipeline-library/ tag 2018.8.0-milestone1 |
| **MCP offline image model** | https://github.com/Mirantis/mcp-offline-model tag 2018.8.0 | n/a (The milestone release does not provide for the offline deployment use case.) |
| **Docker images** | alerta-web docker-prod-local.artifactory.mirantis.com/mirantis/external/alerta-web:2018.8.0  
|  | alertmanager docker-prod-local.artifactory.mirantis.com/openstack-docker/alertmanager:2018.8.0  
|  | aptly docker-prod-local.artifactory.mirantis.com/mirantis/cicd/aptly:2018.8.0  
|  | aptly-api docker-prod-local.artifactory.mirantis.com/mirantis/cicd/aptly-api:2018.8.0  
<table>
<thead>
<tr>
<th>Package</th>
<th>Docker URI Path</th>
<th>Milestone Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>ctl</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/projectcalico/ctl:v1.6.4</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/projectcalico/ctl:v1.6.4</td>
</tr>
<tr>
<td>cni</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/projectcalico/cni:v1.11.6</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/projectcalico/cni:v1.11.6</td>
</tr>
<tr>
<td>Component</td>
<td>Image</td>
<td>Image</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>hyperkube-amd64</td>
<td>docker-prod-local.artifactory.mirantis/kubernetes/hyperkube-amd64:v1.10.4-4</td>
<td>docker-prod-local.artifactory.mirantis/kubernetes/hyperkube-amd64:v1.10.4-4-milestone1</td>
</tr>
<tr>
<td>k8s-dns-kube-dns-amd64</td>
<td>gcr.io/google_containers/k8s-dns-kube-dns-amd64:1.14.5</td>
<td>gcr.io/google_containers/k8s-dns-kube-dns-amd64:1.14.5</td>
</tr>
<tr>
<td>k8s-dns-dnsmasq-amd64</td>
<td>gcr.io/google_containers/k8s-dns-dnsmasq-amd64:1.14.5</td>
<td>gcr.io/google_containers/k8s-dns-dnsmasq-amd64:1.14.5</td>
</tr>
<tr>
<td>k8s-dns-sidecar-amd64</td>
<td>gcr.io/google_containers/k8s-dns-sidecar-amd64:1.14.5</td>
<td>gcr.io/google_containers/k8s-dns-sidecar-amd64:1.14.5</td>
</tr>
<tr>
<td>k8s-netchecker-agent</td>
<td>docker-prod-local.artifactory.mirantis/k8s-netchecker-agent:v1.2.2</td>
<td>docker-prod-local.artifactory.mirantis/k8s-netchecker-agent:v1.2.2</td>
</tr>
<tr>
<td>k8s-netchecker-server</td>
<td>docker-prod-local.artifactory.mirantis/k8s-netchecker-server:v1.2.2</td>
<td>docker-prod-local.artifactory.mirantis/k8s-netchecker-server:v1.2.2</td>
</tr>
<tr>
<td>kubernetes-dashboard-amd64</td>
<td>n/a</td>
<td>k8s.gcr.io/kubernetes-dashboard-amd64:v1.8.3</td>
</tr>
<tr>
<td>Component</td>
<td>Docker Image URL</td>
<td>Docker Image URL</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>node</td>
<td>docker-prod-local.artifactory.mirantis/projectcalico/calico/node:v2.6.10</td>
<td>docker-prod-local.artifactory.mirantis/projectcalico/calico/node:v2.6.10</td>
</tr>
<tr>
<td>Package</td>
<td>MCP Offline Image</td>
<td>MCP Release Milestone</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------</td>
<td>-----------------------</td>
</tr>
</tbody>
</table>

1(1, 2, 3) Available in the MCP offline image. For details, see: MCP Reference Architecture: Mirror image content.
Major components versions

The following tables list the MCP components of the Q2`18 Release Versions with Build IDs 2018.8.0 and 2018.8.0-milestone1. These components are initially installed by default depending on the cluster deployment model. The Build ID 2018.8.0-milestone1 is the Q2`18 milestone Release Version that adds support for OpenContrail 4.0 with Kubernetes 1.10.

For comparison purposes, the tables also list the MCP components versions of the previous GA MCP version, which is 2018.4.0.

The tables divide the following types of the MCP components and their respective subcomponents:

- Software components and Release Versions from the Mirantis repositories Includes components from the Mirantis repositories that are built and managed by Mirantis. Some packages from these repositories substitute the same packages from mirrored repositories using combinations of package versions and repositories priorities of an MCP cluster.

- Software components and Release Versions from mirrored repositories Includes components from mirrored repositories that are redistributed by Mirantis but unmodified with regard to their upstream versions.

### Software components and Release Versions from the Mirantis repositories

<table>
<thead>
<tr>
<th>Component</th>
<th>Application/service</th>
<th>2018.4.0</th>
<th>2018.8.0 (current release)</th>
<th>2018.8.0-milestone1 (current release)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed storage</td>
<td>Ceph</td>
<td>12.2.4-1xen ial</td>
<td>12.2.4-1xen ial+mir3</td>
<td>12.2.4-1xen ial+mir3</td>
<td>Luminous v12.2.4 Release Notes</td>
</tr>
<tr>
<td>DriveTrain</td>
<td>Aptly</td>
<td>1.2.0</td>
<td>1.3.0</td>
<td>1.3.0</td>
<td>Resource type - Docker image</td>
</tr>
<tr>
<td></td>
<td>aptly-publisher</td>
<td>0.12.11</td>
<td>0.12.11</td>
<td>0.12.11</td>
<td>Resource type - Docker image</td>
</tr>
<tr>
<td></td>
<td>Gerrit</td>
<td>2.13.6</td>
<td>2.13.6</td>
<td>2.13.6</td>
<td>Resource type - Docker image</td>
</tr>
<tr>
<td></td>
<td>jenkins-master</td>
<td>2.100</td>
<td>2.121.2, 2.100</td>
<td>2.121.2, 2.100</td>
<td>Resource type - Docker image.</td>
</tr>
<tr>
<td></td>
<td>Jenkins pipe</td>
<td>2018.4.0</td>
<td>2018.8.0</td>
<td>2018.8.0-milestone1</td>
<td>Resource type - Git repository</td>
</tr>
<tr>
<td></td>
<td>line-library</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reclass</td>
<td>1.4.1</td>
<td>1.4.1-1tcp4</td>
<td>1.4.1-1tcp4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reclass model</td>
<td>2018.4.0</td>
<td>2018.8.0</td>
<td>2018.8.0-milestone1</td>
<td>Resource type - Git repository</td>
</tr>
<tr>
<td>Resource Type</td>
<td>Version</td>
<td>Notes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Salt Formulas</td>
<td>2018.4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hypervisors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>libvirt</td>
<td>4.0.0-1.7&lt;br&gt;16.04+mcp1</td>
<td>4.0.0-1.7&lt;br&gt;16.04+mcp2</td>
<td>4.0.0-1.7&lt;br&gt;16.04+mcp2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>qemu</td>
<td>2.11+dfsg-1&lt;br&gt;.1~u16.04+&lt;br&gt;mcp1</td>
<td>2.11+dfsg-1&lt;br&gt;.4~u16.04+&lt;br&gt;mcp2</td>
<td>2.11+dfsg-1&lt;br&gt;.2~u16.04+&lt;br&gt;mcp2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kubernetes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.8.11-9</td>
<td>1.10.4-4</td>
<td>1.10.4-4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vortlet</td>
<td>1.0.0</td>
<td>1.1.2</td>
<td>1.1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenContrail Networking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cassandra</td>
<td>2.1.20</td>
<td>2.2.12, 2.1.20</td>
<td>2.2.12, 2.1.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kafka</td>
<td>2.9.2-0.8.2&lt;br&gt;-0-contrail0</td>
<td>2.11-0.9.0.1&lt;br&gt;-0-contrail1, 2.9.2-0.0-&lt;br&gt;contrail0</td>
<td>2.11-0.9.0.1&lt;br&gt;-0-contrail1, 2.9.2-0-&lt;br&gt;contrail0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenContrail</td>
<td>3.2.3.x, 3.2.10.0</td>
<td>4.0.4.0, 3.2.11.0</td>
<td>4.0.4.0, 3.2.11.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OpenContrail 3.2.x does not support OpenStack Pike. OpenContrail 3.2.x is not supported for Kubernetes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zookeeper</td>
<td>3.4.5</td>
<td>3.4.8, 3.4.5</td>
<td>3.4.8, 3.4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Version 3.4.8-1 for OpenContrail 4.0.4.0, 3.4.5 for OpenContrail 3.2.11.0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenStack Pike²</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Barbican</td>
<td>5.0.x</td>
<td>5.0.0-3~u16.04+mcp0</td>
<td>5.0.0-3~u16.04+mcp0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cinder</td>
<td>11.1.x</td>
<td>11.1.1-2~u16.04+mcp77</td>
<td>11.1.1-2~u16.04+mcp77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Designate</td>
<td>5.0.x</td>
<td>5.0.2-2~u16.04+mcp6</td>
<td>5.0.2-2~u16.04+mcp6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

¹ OpenContrail is not supported for Kubernetes.

² OpenStack Pike supports OpenStack Queens.
<table>
<thead>
<tr>
<th></th>
<th>15.0.x</th>
<th>15.0.1-1~u1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6.04+mcp1</td>
<td>6.04+mcp1</td>
</tr>
<tr>
<td>Glance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15.0.1-1~u1</td>
<td>6.04+mcp1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heat</td>
<td>9.0.x</td>
<td>9.0.4-1~u16</td>
</tr>
<tr>
<td></td>
<td>0.04+mcp31</td>
<td>0.04+mcp31</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Horizon</td>
<td>12.0.x</td>
<td>12.0.3-2~u1</td>
</tr>
<tr>
<td></td>
<td>6.04+mcp48</td>
<td>6.04+mcp48</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ironic</td>
<td>9.1.x</td>
<td>9.1.4-1~u16</td>
</tr>
<tr>
<td></td>
<td>0.04+mcp29</td>
<td>0.04+mcp29</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keystone</td>
<td>12.0.x</td>
<td>12.0.1-4~u1</td>
</tr>
<tr>
<td></td>
<td>6.04+mcp9</td>
<td>6.04+mcp9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manila</td>
<td>5.0.x</td>
<td>5.0.1-1~u16</td>
</tr>
<tr>
<td></td>
<td>0.04+mcp61</td>
<td>0.04+mcp61</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nova</td>
<td>16.1.x</td>
<td>16.1.4-3~u1</td>
</tr>
<tr>
<td></td>
<td>6.04+mcp132</td>
<td>6.04+mcp132</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenStack</td>
<td>BGP VPN</td>
<td>7.0.x</td>
</tr>
<tr>
<td>Networking</td>
<td></td>
<td>7.0.0-2~u16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.04+mcp2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.0.0-2~u16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.04+mcp2</td>
</tr>
<tr>
<td>L2 Gateway</td>
<td>11.0.x</td>
<td>11.0.0-1~u1</td>
</tr>
<tr>
<td></td>
<td>6.04+mcp6</td>
<td>6.04+mcp6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neutron</td>
<td>11.0.x</td>
<td>11.0.5-1~u1</td>
</tr>
<tr>
<td></td>
<td>6.04+mcp33</td>
<td>6.04+mcp33</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenStack</td>
<td>Aodh</td>
<td>5.x</td>
</tr>
<tr>
<td>Telemetry</td>
<td></td>
<td>5.1.0-3~u16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.04+mcp10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.1.0-3~u16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.04+mcp10</td>
</tr>
<tr>
<td>Ceilometer</td>
<td>9.0.5</td>
<td>9.0.5-2~u16</td>
</tr>
<tr>
<td></td>
<td>0.04+mcp16</td>
<td>9.0.5-2~u16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.04+mcp16</td>
</tr>
<tr>
<td>Panko</td>
<td>3.1.x</td>
<td>3.1.0-1~u16</td>
</tr>
<tr>
<td></td>
<td>0.04+mcp11</td>
<td>3.1.0-1~u16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.04+mcp11</td>
</tr>
<tr>
<td>Gnocchi</td>
<td>4.0.4</td>
<td>4.0.5-2~u16</td>
</tr>
<tr>
<td></td>
<td>0.04+mcp2</td>
<td>4.0.5-2~u16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Telemetry</td>
</tr>
<tr>
<td>StackLight</td>
<td>jmx-exporte</td>
<td>0.10~20170</td>
</tr>
<tr>
<td>LMA</td>
<td>r</td>
<td>912072743</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.10~20180</td>
</tr>
<tr>
<td></td>
<td></td>
<td>716100330</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.10~20180</td>
</tr>
<tr>
<td></td>
<td></td>
<td>716100330</td>
</tr>
<tr>
<td>libvirt-expor</td>
<td>0.1~201803</td>
<td>0.1~201803</td>
</tr>
<tr>
<td>ter</td>
<td>19200151</td>
<td>19200151</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1~201803</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19200151</td>
</tr>
<tr>
<td>Telegraf</td>
<td>1.5.3</td>
<td>1.5.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.5.3</td>
</tr>
</tbody>
</table>

©2020, Mirantis Inc.
<table>
<thead>
<tr>
<th>Software components and Release Versions from mirrored repositories</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Component</strong></td>
</tr>
<tr>
<td>DriveTrain</td>
</tr>
<tr>
<td>SaltStack</td>
</tr>
<tr>
<td>Kubernetes</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>OpenContrail networking</td>
</tr>
<tr>
<td>StackLight LMA</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Software</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>Fluentd</td>
</tr>
<tr>
<td>Grafana</td>
</tr>
<tr>
<td>Heka</td>
</tr>
<tr>
<td>InfluxDB</td>
</tr>
<tr>
<td>Kibana</td>
</tr>
<tr>
<td>Prometheus</td>
</tr>
<tr>
<td>Pushgateway</td>
</tr>
<tr>
<td>System</td>
</tr>
<tr>
<td>Docker</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Git</td>
</tr>
<tr>
<td>GlusterFS</td>
</tr>
<tr>
<td>HAProxy</td>
</tr>
<tr>
<td>NGINX</td>
</tr>
<tr>
<td>OpenLDAP</td>
</tr>
<tr>
<td>Galera</td>
</tr>
<tr>
<td>MySQL</td>
</tr>
<tr>
<td>Base OS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>OS for HW nodes</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

2 For the OpenStack releases support schedule, see [MCP OpenStack Releases](#).
3 End of software development (EoSD)
4 When newer versions of Ubuntu packages are available in the Mirantis repositories, MCP installs them instead of the versions available in the Ubuntu repositories.

©2020, Mirantis Inc.
Addressed issues

This section provides the list of the addressed issues in the current MCP release version.

**DriveTrain**

- Fixed the salt-pepper package to be properly installed on the cfg01 node by default to resolve the issue with failures of the Deploy - OpenStack Jenkins deployment pipeline in case of an offline deployment.
- Fixed the issue that caused the Deploy - update cloud pipeline fail to deploy the ctl nodes in case of the workload server running in the ERROR state.
- Fixed the issue with the updating of the dbs nodes using the Deploy - update cloud Jenkins pipeline. The issue caused the pipeline to get stuck in a loop verifying if Galera is working properly.
- Fixed the issue with Salt minions failing to start after reboot on the proxy nodes.
- Fixed the issue with the salt.control state that could lead to data loss in case of non-running libvirt.
- Fixed the issue with Xtrabackup included to the rsync of the MySQL backup using Backupninja.
- Fixed the issue with the inability to generate an MCP cluster model for a large OpenStack architecture with nine KVM nodes and six controller nodes using the existing Cookiecutter templates.
- Fixed the issue with the debmirror Salt Formula that did not allow providing a proxy. This prevented from using debmirror through the Salt Formula in environments where external communication is allowed only through a proxy.
- Enabled the home_dir_mode permissions in the linux.system formula.
- Fixed the dependency for DPDK bond interfaces that could cause the linux.network state failure when using the DPDK bond. The issue occurred if one of the DPDK links was not ready and available at that moment.

**OpenStack**

- Fixed the issue with Horizon ignoring the LOGIN_REDIRECT_URL setting. The issue affected the OpenStack Pike release.
- Fixed the issue with the test_autoscaling Tempest test failing for OpenStack Telemetry.
- Fixed the issue that prevented booting from a Virtio-SCSI RBD volume with a configuration drive.
- Fixed the issue leading to evacuation failure of a VM with SR-IOV in case the target host has already had a VM with the same PCI address as the original VM.
- Fixed the issue with Nova failing to create live snapshots of an instance if the original VM image changed its accessibility.
• Fixed the issue with the tempest.api.compute.admin.test_live_migration Tempest test failures.

• Fixed the Watchdog restart failure that caused the Deploy - OpenStack Jenkins pipeline failures.

• Fixed the race condition that might occur when running highstate.

• Fixed a Manila deployment issue by increasing the default HAProxy server response timeout. The initial server response timeout set in HAProxy was too short for Manila API to return the response.

• Fixed the issue with cinder-volume logging.

• Fixed the issue that caused Python services on controller nodes VMs to consume /dev/random entropy leading to connection failures when using SSL for Galera.

• Fixed the issue that could cause failure of an OpenStack environment deployment using the Deploy - OpenStack Jenkins pipeline during the deployment of LDAP for Keystone.

• Fixed logging for nova-placement under the Web Server Gateway Interface (WSGI).

• Fixed the issue with the CIS audit benchmark that caused Keystone having no access to files on the ctl node.

• Fixed the issue with python-barbican dependencies leading to failure of Keystone.

• Fixed the issue with Nova API using the Keystone public endpoint for project ID verification.

OpenContrail

• Fixed the issue with the policy-create Neutron OpenContrail extension not working properly.

• Fixed the issue with some Neutron extensions missing after upgrading OpenContrail to version 3.2.

• Fixed the issue with the desynchronization of the irond ifmap-server used by OpenContrail by enabling an internal OpenContrail IF-MAP implementation as an alternative to irond. Previously, when ifmap and contrail-api could not contact each other, the new instances failed to reach the metadata service causing desynchronization across all network nodes.

• Fixed the issue with the OpenContrail DPDK-based vRouter failing to start with 1 GB hugepages enabled.

• Fixed the issue with failure to create a new network through Horizon for the OpenStack environments with OpenContrail.

• Fixed the issue with the OpenContrail web UI dashboard showing no alarms for the vRouter process in the DOWN state.

• Fixed in Build ID 2018.8.0-milestone1 Fixed the issue with the vrouter-agent failure on the OpenContrail 4.0 with the DPDK-accelerated dataplane deployments by enhancing the logic of the vRouter services restart.

• Fixed in Build ID 2018.8.0-milestone1 Fixed the issue with no connectivity between the VM instances on the OpenContrail 4.0 with the DPDK-accelerated dataplane deployments due to the incorrect MTU parameters.
• Fixed in Build ID 2018.8.0-milestone1 Fixed the issue with the broken HAProxy configuration file on the OpenContrail 4.0 deployments.

**StackLight LMA**

• Fixed the issue with the Prometheus image tags that could lead to the downgrade of the Prometheus major version during the upgrade to the 2018.4.0 release version.
• Fixed the issue with Fluentd failing to parse the neutron-server logs on the ctl nodes and causing no OpenStack API availability data being displayed in Grafana. The issue affected only the OpenStack Pike release.
• Fixed the issue with the Cinder and Nova Grafana dashboards displaying an incorrect number of volumes and volumes in use.
• Fixed the issue with the Nova Grafana dashboard displaying wrong values in Virtual Disks.
• Fixed the issue with the PrometheusTargetDown alert missing the host field.
• Fixed the issue with the inability to use multiple notification channels when using Alertmanager for the StackLight LMA notifications.
• Fixed the issue with the Jenkins Grafana dashboard showing no metrics.
• Fixed the issue with the Prometheus and Alertmanager StackLight LMA services being accessed through HTTP instead of HTTPS.

**Networking**

• Fixed the issue with Neutron failing to attach multiple networks to a DVR router in case of requests coming in parallel.
• Added the capability to configure ovsdb_interface for Neutron agents to fix the issue with the DHCP agent failing to add its tap port into br-int and causing VM creation failures in case of OpenDaylight used as an SDN.
• Fixed the issue with duplicated entries creation in /etc/network/interfaces and /etc/network/interfaces.d/ifcfg-br-prv in case of using DPDK.
• Fixed the issue with the HAProxy service being installed and started with default configurations once the metadata agent is installed. Now, the HAProxy service is disabled after installation and starts only in a particular namespace by the request of the neutron-metadata-agent.
• Fixed the issue with the mechanism_drivers list by ordering the mechanism driver entry points.
• Fixed the issue with the inability to set a custom firewall driver in Neutron.
• Fixed the issue with the Neutron ML2 dhcp_lease_duration being not parametrized.
• Fixed the issue with the inability to create multiple sets of HugePages.
• Fixed the issue with the inability to specify the exact list of flat networks or disable flat networks in Neutron.
• Fixed the issue with the inability to explicitly specify the list of physical networks in Neutron.
Mirantis Technical Bulletins

Mirantis constantly focuses on the product quality and stability. Therefore, aside from the fixes of the security and critical flaws for the current MCP version affecting Mirantis products and services, we provide resolution for the customer deployments on top of the previous MCP versions, which can be affected, in the form of technical bulletins. Each technical bulletin includes the detailed issue description, possible impact, steps to determine whether a deployment is affected with the issue, procedure to resolve the issue, and revert the fix if required.

Such security and critical issue advisories are also proactively e-mailed to the customers with active service contracts.

For the full list of the Mirantis Technical Bulletins, refer to the Mirantis OpenStack Technical Bulletins page at the Mirantis official website.
Known issues
This section lists the MCP known issues and workarounds.

DriveTrain
This section lists the DriveTrain-related issues with available workarounds.

• The Salt Master CA does not provide the Certificate Revocation List (CRL) and index files to identify the revoked or expired certificates.
   Workaround:
   To list all currently issued certificates, follow the step 3 of the Replace the Salt Master CA certificates procedure.

• The definition is missing for Gerrit SSH on port 29418, and only Gerrit HTTP is exposed through prx0[1-2].
   Workaround:
   Use the ngx_stream_proxy_module that supports both HTTP and HTTPS.

• If the Gerrit container is recreated, the Jenkins update service config build fails with the WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! warning during the git fetch step for obtaining the upstream libraries. The issue occurs because of the old entries in the /var/jenkins_home/.ssh/known_hosts file.
   Workaround:
   1. Log in to the Jenkins Master container.
   2. Delete the old entry of ssh-keygen in /var/jenkins_home/.ssh/known_hosts.

• The restoration of the MySQL database using Xtrabackup fails in the offline mode with the following error message: tar: qpress-11-linux-x64.tar: Cannot open: No such file or directory.
   Workaround:
   1. Log in to the apt01 offline node.
   2. Download the qpress-11-linux-x64.tar file.
   3. Move the downloaded file to the /srv/http/ directory.

• The Deploy - OpenStack job may occasionally fail with the SaltInvocationError: ca_server did not respond salt master must permit peers to call the sign_remote_certificate function error message.
   Workaround:
1. Examine the output or log to determine the failed state and the target node.
2. Log in to the Salt Master node.
3. Rerun the failed state for the target node:
   ```
salt -C '<target_nodes>' state.sls '<state>'
   
   For example:
   ```
   ```
   salt -C '@salt:minion' state.sls 'salt.minion.cert'
   ```
4. Rerun the Deploy - OpenStack job.

   The Deploy - OpenStack job fails on the `linux.system.file` state with the 404 error when trying to fetch the qcow2 images from http://images.mirantis.com/.

   **Workaround:**
   1. Open your Git project repository with the Reclass model on the cluster level.
   2. In `/infra/init.yml`:
      1. Find the following strings:
         ```
         salt_control_trusty_image: http://images.mirantis.com/ubuntu-14-04-x64-mcp2018.8.0.qcow2
         salt_control_xenial_image: http://images.mirantis.com/ubuntu-16-04-x64-mcp2018.8.0.qcow2
         ```
      2. Replace the strings with the following ones:
         ```
         salt_control_trusty_image: http://images.mirantis.com/ubuntu-14-04-x64-mcp2018.8.0.qcow2
         salt_control_xenial_image: http://images.mirantis.com/ubuntu-16-04-x64-mcp2018.8.0.qcow2
         ```

**OpenStack**

   The maximum size of a Glance image is limited to 30 GB on the system level of the Reclass model in `/nginx/server/proxy/openstack/glance.yml`.

   **Workaround:**
   1. Open your Git project repository with the Reclass model on the cluster level.
   2. In `openstack/proxy.yml`, add the following parameters under `nginx:server:site`:
      ```
      nginx_proxy_openstack_api_glance:
        proxy:
        request_buffer: false
        size: 100000m
      ```
Note

The request_buffer: false parameter is added for requests to be sent to Glance directly, without cache. Otherwise, the prx node may have failures.

• On an MCP cluster containing 1000 compute nodes, the RabbitMQ three-node cluster stops receiving connection after approximately five minutes of processing messages from services and a huge number of agents.

  Workaround:

  Divide the compute nodes into cells with 200-300 nodes per each cell and dedicate one three-node msg and dbs cell per each compute cell. For details, see the OpenStack documentation:

  • Multiple Cells
  • Setup of Cells V2

• Since the Ceilometer API is deprecated starting OpenStack Ocata, Ceilometer API encounters the database connection failures when deployed with OpenStack Pike. Therefore, use Gnocchi API, Aodh API, and Panko to access the Telemetry data.

  Warning

  Telemetry for OpenStack Pike does not support integration with StackLight LMA.

• When the Xtrabackup cron job is running for MySQL, the database is not available for other MCP cluster activities.

  Workaround:

  1. Open your Git project repository with the Reclass model on the cluster level.
  2. In /openstack/database_init.yml, disable the automatic backup for Xtrabackup:

```yaml
...  
parameters:
  xtrabackup:
    client:
    cron: False
```

  3. Apply changes by performing the steps 5-10 of the Xtrabackup configuration procedure.
The OpenRC v2/v3 file located in Horizon may not be valid for OpenStack Pike. To verify if you are affected, run `openstack catalog list` from the ctl node. The Keystone catalog must have an unversioned identity endpoint. For example:

```
| keystone | identity       | RegionOne                             |
|          |               | admin: http://172.30.121.15:35357     |
|          |               | RegionOne                             |
|          | public: https://125.22.218.33:5000     |
|          |               | RegionOne                             |
|          | internal: http://172.30.121.15:5000    |
```

If the identity endpoint includes a version, proceed to the workaround below.

**Workaround:**

1. Log in to the Salt Master node.
2. In the `openstack/control_init.yml` and `openstack/control.yml` files, add the following parameters:
   ```yaml
   parameters:
     _param:
       cluster_public_protocol: https
       keystone_service_protocol: http
       keystone_public_path: "/
       keystone_internal_path: "/
       keystone_admin_path: "/
   ```
3. Apply the following state:
   ```bash
   salt -C 'I@keystone:client' state.apply keystone.client
   ```
4. On the prx nodes:
   1. Verify that the `openstack/proxy.yml` file does not include the `_param:horizon_identity_version: 2` parameter or its value is 3.
   2. Verify that the `/etc/openstack-dashboard/local_settings.py` file includes:
      ```python
      OPENSTACK_API_VERSIONS = { "identity": 3 }
      ```

On the OpenStack environments with OpenContrail and Barbican, if you use a non-default Keystone domain, the LBaaS VIP cannot be created. LBaaS cannot download a secret created by the Barbican user in any project other than the project where opencontrail_barbican_user has admin privileges.

**Workaround:**
1. Apply the Salt formula patch to your Barbican Salt formula.
2. On every OpenStack controller node where Barbican API is installed, add the following configuration to /etc/barbican/policy.json:

```
barbican:
  server:
    policy:
      all_domains_reader: 'user:<user_ID> and project:<project_ID>'
      secret_acl_read: '"read":%(target.secret.read)s or rule:all_domains_reader"
      container_acl_read: '"read":%(target.container.read)s or rule:all_domains_reader"
```

Replace <user_ID> and <project_ID> with the corresponding OpenStack environment values.

3. Log in to the Salt Master node.
4. Apply the following state:

```
salt -C 'I@barbican:server' state.apply barbican
```

This configuration adds appropriate rights to read the secrets and containers from Barbican.

**Kubernetes**

- The Kubernetes Nodes (cmp) autogeneration supports up to 155 nodes only.
  
  Workaround:

  While generating your Kubernetes-based MCP cluster model using the Model Designer UI, set refs/changes/87/23387/6 in the Cookiecutter template branch parameter field.
• Fixed in Build ID 2018.8.0-milestone1 The Deploy - OpenStack Jenkins pipeline job fails during the StackLight LMA deployment stage when deploying a Kubernetes cluster with StackLight LMA.

Workaround:

Choose from the following options:

• Prior to the deployment:
  1. Open your Git project repository with the Reclass model on the cluster level.
  2. In the /cluster/<cluster_name>/kubernetes/control.yml file, add the system.kubernetes.control.roles.fluentd-view class.

• During the deployment:
  1. Run the following command:

```bash
salt -C 'I@kubernetes:master' state.sls kubernetes exclude=kubernetes.master.setup
```
  2. Rerun the Deploy - OpenStack Jenkins pipeline job.

• If you are deploying a Kubernetes cluster without access to Mirantis internal network using the Deploy - OpenStack Jenkins pipeline job, you may encounter an issue with Artifactory availability.

Workaround:

1. Open your Git project repository with the Reclass model on the cluster level.
2. In the kubernetes/control.yml and kubernetes/compute.yml files, add the following parameters:

```yaml
parameters:
  param:
  kubernetes_calico_calicoctl_repo: docker-prod-local.artifactory.mirantis.com/mirantis/projectcalico/calico
  kubernetes_calico_repo: docker-prod-local.artifactory.mirantis.com/mirantis/projectcalico/calico
  kubernetes_calico_cni_repo: docker-prod-local.artifactory.mirantis.com/mirantis/projectcalico/calico
  kubernetes_hyperkube_repo: docker-prod-local.artifactory.mirantis.com/mirantis/kubernetes
  kubernetes_contrail_cni_repo: docker-prod-local.artifactory.mirantis.com/mirantis/kubernetes/contrail-integration
  kubernetes_sriov_repo: https://docker-prod-local.artifactory.mirantis.com/artifactory/binary-prod-local/mirantis/kubernetes/sriov-cni
```
3. Rerun the Deploy - OpenStack Jenkins pipeline job.

• Once you deploy a Kubernetes cluster, the Kubernetes dashboard is not accessible because the role-based access control (RBAC) is not supported in the current version of the Kubernetes dashboard by default.

Workaround:

1. Log in to the Salt Master node.
2. Create an empty k8s directory.
3. Change directory to k8s and initialize an empty git repository using git init.
4. Clone the Kubernetes Salt Formula repository:
5. Obtain the patch:

```bash
git checkout 0637cd6b68f5ec93096ef0e93656501671e5119
```

6. Copy the new files to the Kubernetes Salt Formula directory:

```bash
cp -r ~/k8s/kubernetes /usr/share/salt-formulas/env/
```

7. Open your Git project repository with the Reclass model on the cluster level.

8. In the `kubernetes/init.yml` file, add the following parameters:

```yaml
parameters:
_param:
  kubernetes_dashboard_image: gcr.io/google_containers/kubernetes-dashboard-amd64:v1.8.3
```

9. Apply the following state:

```bash
salt -C 'I@kubernetes:master' state.sls kubernetes.master.kube-addons
```

## OpenContrail

- Fixed in Build ID 2018.8.0-milestone1 The OpenContrail analytics database is overloaded with a big number of exported flows from all vRouters. The issue usually occurs on large deployments with high traffic.

  **Workaround:**
  1. Log in to the OpenContrail web UI.
  2. In Configure > Infrastructure > Global Config, open the Forwarding Options tab.
  3. Click the Edit forwarding options icon in the upper right corner of the tab.
  4. Set the Flow Export Rate option to 0.

- Fixed in Build ID 2018.8.0-milestone1 A Pike-based MCP cluster contains the following OpenContrail 4.0 region-related issues:

  - The `contrail-keystone-auth.conf` file does not have control over region. And the `openstack_region` pillars are missing in `classes/system/contrail/control/control4_0.yml` and `classes/system/contrail/compute/cluster4_0.yml`.

  - In `contrail-svc-monitor.conf`, the `region_name` parameter is hardcoded.

  - In the OpenContrail web UI, the region parameter can not be changed.

  **Workaround:**
1. Apply the Salt formula patches to your OpenContrail Salt formula.
2. Open your project Git repository on the cluster level.
3. Add the following pillar to /classes/cluster/<cluster_name>/opencontrail/control.yml:

   ```yaml
   ...
   opencontrail:
     web:
       identity:
         region: ${_param:openstack_region}
   ...
   ```

4. Add the following pillars to /classes/cluster/<cluster_name>/opencontrail/compute.yml:

   ```yaml
   ...
   opencontrail:
     common:
       identity:
         region: ${_param:openstack_region}
   ...
   ```

5. Choose from the following options:

   - For new deployments, proceed with the cluster configuration as required. The fix will be applied during your cluster deployment.
   - For existing deployments, apply the following state from the Salt Master node:

     ```bash
     salt -C 'I@opencontrail:control' state.apply opencontrail
     ```

   **Caution!**

   After upgrading your MCP cluster to the next release version, for example, to the Build ID 2018.8.0-milestone1, revert the changes made in the OpenContrail control.yml and compute.yml files.

   - During the upgrade of OpenContrail from v3.2 to v4.0, the Deploy - upgrade Opencontrail to 4.x pipeline fails due to ZooKeeper failing to connect to Kafka on all analytics nodes. This causes some services stuck in the initializing state, for example:

     ```
     contrail-collector: initializing
     contrail-analytics-api: initializing (UvePartitions:UVE-Aggregation[None], Collector connection down)
     contrail-alarm-gen: initializing (Collector, Zookeeper:AlarmGenerator[Connection time-out] connection down)
     ```

     **Workaround:**

     1. Restart ZooKeeper on all analyticsdb containers.
2. Once ZooKeeper is up and running, restart the services that are stuck.
3. Rerun the Deploy - upgrade Opencontrail to 4.x pipeline.

- During the upgrade of OpenContrail from v3.2 to v4.0, the Deploy - upgrade Opencontrail to 4.x pipeline fails with the Command `docker-compose up -d` failed error message due to several configuration templates for analyticsdb containers created as directories instead of files. This causes docker-compose fail to initialize the opencontrail_analytics_1 and opencontrail_analyticsdb_1 containers.
  Workaround:
  1. Run the following command from the Salt Master node:

```
    salt -C 'nal*' cmd.run "rm -rf /etc/cassandra/cassandra_analytics.yaml \
    && rm -rf /etc/cassandra/cassandra-env-analytics.sh/ \n    && rm -rf /etc/zookeeper/conf/zoo_analytics.cfg"
```

  2. Rerun the Deploy - upgrade Opencontrail to 4.x pipeline.

- Fixed in Build ID 2018.8.0-milestone1 Creating a network in Horizon on an OpenStack-based MCP deployment with OpenContrail v4.0 may fail with the Failed to create network “test_net”: Unrecognized attribute(s) ‘contrail:policys’ error message. The issue also affects the upgrade of OpenContrail from v3.2 to v4.0.
  Workaround:
  1. Log in to the Salt Master node.
  2. In the classes/cluster/<cluster_name>/openstack/proxy.yml file, move the cluster.{{ cookiecutter.cluster_name }}.infra class to the bottom of the classes list.
  3. Refresh Salt pillars:

```
salt "prx*" saltutil.refresh_pillar
```

  4. Verify that a proper OpenContrail version is set in the pillar:

```
salt "prx*" pillar.data | grep opencontrail_version -A1
```

  The expected value for OpenContrail v4.0 is 4.0.
  5. Rerun the horizon state on the prx nodes:

```
salt "prx*" salt.sls horizon
```
After deploying OpenContrail v4.0 or upgrading it from v3.2 to v4.0 on an OpenStack-based MCP cluster without DPDK, the Contrail vRouter uses the DPDK package.

Workaround:

1. SSH to an OpenStack compute node.

2. Verify if the contrail-vrouter-dpdk package is present on the node:

   ```
   dpkg -l | grep contrail-vrouter-dpdk
   ```

3. If the package is present, uninstall it:

   ```
   apt purge -y contrail-vrouter-dpdk
   ```

4. Reboot the OpenStack compute node:

   ```
   reboot
   ```

5. SSH to the OpenStack compute node.

6. Verify that the contrail-vrouter-dpdk is not present:

   ```
   dpkg -l | grep contrail-vrouter-dpdk
   ```

7. Verify that the OpenContrail services are in the active state:

   ```
   contrail-status
   ```

   Example:

   ```
   root@cmp002:~# contrail-status
   == Contrail vRouter ==
   contrail-vrouter-agent:       active
   contrail-vrouter-nodemgr:     active
   ```

8. Perform the steps 1-7 on the remaining OpenStack compute nodes.

After deploying OpenContrail v4.0, some OpenContrail services on one of the analytics nodes may be inactive.

Workaround:

1. Log in to the Salt Master node.

2. Restart all analytics containers:

   ```
   salt -C '@opencontrail:collector' cmd.run 'docker restart opencontrail-analytics_1'
   ```
• After the upgrade of OpenContrail from v3.2 to v4.0, the old package versions remain on the ntw and nal nodes. Once you restart one of the ntw or nal nodes, the services of OpenContrail v3.2 start on the host and services of OpenContrail v4.0 start in containers. This causes both versions of the services trying to open the same network ports.

Workaround:

1. Log in to the Salt Master node.
2. Disable the services of OpenContrail v3.2:

   ```
salt -C 'ntw*' cmd.run "for service_name in supervisor-config supervisor-control supervisor-database contrail-webui-webserver contrail-webui-jobserver zookeeper ifmap-server; do echo 'manual' | sudo tee /etc/init/$service_name.override; done; for service_name in contrail-control redis-server supervisor; do update-rc.d $service_name disable; done"
   
salt -C 'nal*' cmd.run "for service_name in supervisor-analytics supervisor-database zookeeper; do echo 'manual' | sudo tee /etc/init/$service_name.override; done; for service_name in redis-server supervisor; do update-rc.d $service_name disable; done"
   ```

• In the OpenContrail-based Kubernetes MCP clusters, the conformance tests from the StatefulSetBasic set may fail if the concurrency is more than 2. Inside pods, the logs may contain the network outage messages. Mirantis recommends running all tests from the StatefulSetBasic set of the conformance tests consequentially, with concurrency=1.

• During the deployment of Kubernetes 1.10 with OpenContrail 4.0 using the Build ID 2018.8.0-milestone1, the Kubernetes Nodes have the apt_mk_version=stable MCP version instead of 2018.8.0-milestone1.

Workaround:

1. In your Git project repository with the Reclass model on the cluster level:
   1. In classes/cluster/<cluster_name>/infra/config.yml, remove the following class from the kubernetes_compute section:

   ```
   - cluster.<cluster_name>.opencontrail.compute
   ```
   2. In classes/cluster/<cluster_name>/kubernetes/compute.yml, add the following class before cluster.<cluster_name>.infra:

   ```
   - cluster.<cluster_name>.opencontrail.compute
   ```
   3. In classes/cluster/<cluster_name>/kubernetes/control.yml, add the following class before cluster.<cluster_name>.infra:

   ```
   - cluster.<cluster_name>.opencontrail.control
   ```
   2. In your Git project repository with the Reclass model on the system level, remove all occurrences of the following class in the classes/system/reclass/storage/system/kubernetes_contrail_cluster.yml file:
3. Log in to the Salt Master node.
4. Apply the following state:

```
salt '*' state.sls linux.system.repo
```

**StackLight LMA**

- The Gainsight integration service may not operate properly due to an issue with the Gainsight Docker image. The workaround is to use the newer available image.

**Workaround:**

1. Open your Git project repository with the Reclass model on the cluster level.
2. In stacklight/client.yml, specify the following parameter:

```
parameters:
  _param:
    ...
    docker_image_prometheus_gainsight: "${_param:mcp_docker_registry}/openstack-docker/gainsight:2019.2.0"
```
3. Log in to the Salt Master node.
4. Refresh Salt pillars:

```
salt -C '*' saltutil.refresh_pillar -t 2
```
5. Apply the new Docker image for Gainsight:

```
salt -C 'l@docker:swarm:role:master' state.sls docker.client
```