MCP Q3`17 Release Notes

version q3-17
Copyright notice

2017 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
Mirantis Cloud Platform introduces a set of new features and enhancements.

MCP DriveTrain automated deployment
To reduce deployment time and eliminate possible human errors, MCP introduces an automated approach to the MCP DriveTrain deployment. This approach is based on the bootstrap automation of the Salt Master and MAAS nodes. On a Reclass model creation, a deployment-specific configuration drive is sent to the user’s email. This configuration drive is then used in cloud-init to set up a virtual machine with the Salt Master node, MAAS node, Jenkins server, and local Git server installed on it.

The high-level workflow of such deployment is as follows:
1. Manually deploy the Foundation physical node.
2. Create the deployment model using the Model Designer web UI.
3. Obtain the pre-built ISO configuration drive containing a system submodule on your email.
4. Bootstrap the Salt Master node and MAAS node.
5. Manually deploy the remaining bare metal servers through MAAS.
7. Switch from using the Jenkins service on the Salt Master node to a deployment-specific CI/CD.

See also
- Automated deployment of MCP DriveTrain

MCP components updates and upgrades
Mirantis Cloud Platform provides a number of update and upgrade procedures to deliver new features and enhancements to your existing MCP cluster.

MCP DriveTrain automated update
Mirantis Cloud Platform introduces pipelines for the MCP DriveTrain automated update. Now, you can easily update to the latest version of System Reclass model, Salt formulas, and Jenkins pipelines through CI/CD.

To add the update pipelines to your CI/CD:
1. Enable the update pipelines in the model by adding the following class to the master.yml file in the classes/cluster/${_param:cluster_name}/cicd/control/ directory:

   ```
   classes:
   - system.jenkins.client.job.deploy.update.utils
   ```
2. Apply the change:

```
salt -C 'I@jenkins:client' state.apply jenkins.client
```

See also

- Update MCP DriveTrain

**MCP cluster update**

MCP enables you to update your existing MCP cluster using the CI/CD update pipelines. The update procedure allows you to apply service configuration changes as well as update the available service packages.

See also

- Update an MCP cluster

**MCP cluster upgrade**

MCP enables you to upgrade your existing MCP cluster using the CI/CD upgrade pipelines. The upgrade procedure covers the following MCP cluster components:

- The Virtualized Control Plane (VCP) operating system upgrade from Ubuntu 14.04 to 16.04
- The VCP, OpenStack compute nodes, and OpenStack OVS nodes upgrade from Mitaka to Ocata
- The Mirantis OpenContrail nodes upgrade from the OpenContrail version 3.0 or 3.1 to 3.2

Note

The upgrade of Kubernetes from version 1.6 to 1.7 is not supported yet.

See also

- Upgrade an MCP cluster

**StackLight enhancements**
StackLight introduces a set of enhancements and changes related to monitoring and alerting:

- Implemented Prometheus, an open-source monitoring solution, as the core component of monitoring and alerting. Prometheus is responsible for scraping and storing the time series data. It replaces both Heka and InfluxDB. However, StackLight still uses Heka in the logging system and InfluxDB as a long-term storage for data analysis.
- Implemented Telegraf, which monitors nodes and collects metrics from the system. Telegraf replaces Collectd.
- Implemented Alertmanager, which handles alerts. Alertmanager replaces Nagios, Sensu, and Uchiwa.
- Integrated StackLight with the Notification Service to enable email or Salesforce notifications about alerts.

The visualization layer remains the same and is provided by Grafana. The legacy StackLight is still supported.

See also
- Logging, monitoring, and alerting planning

DevOps Portal enhancements

The DevOps Portal provides a consolidated view to ease MCP management (Day 2 operations) for cloud operators. The portal collects a comprehensive set of data about the cloud, offers visualization dashboards, and allows the operator to interact with a variety of tools.

More specifically, the DevOps Portal includes the following enhancements:

- Containerized the Push Notification and PostgreSQL services to deploy these services as part of the DevOps Portal deployment procedure.
- Enhanced client and server parts of the Runbooks Automation Salt formula to enable operators to configure the following resources: users and roles, projects, nodes, and jobs.
- Improved the Cleanup service operations:
  - Extended the service`s API to support retrieving the list of items scheduled for cleanup and the DevOps Portal to show this list.
  - Extended the service`s API and the DevOps Portal to support management of whitelist of items scheduled for cleanup.
  - Extended the service`s API and the DevOps Portal to support immediate item cleanup.
  - Extended the service`s API to support retrieving the list of rules that are currently active to clean up cloud items and the DevOps Portal to show this list.
  - Extended the service`s API and the DevOps Portal to support the rules` enabling and disabling.
• Enabled the deployment engineer to configure Cloud Intelligence Service (CIS) collectors into the Runbooks Automation service using a pre-configured repository with collectors’ jobs definitions through Reclass metadata. As a result, the cloud operator gets CIS collectors pre-configured in the Runbooks Automation service on the initial deployment.

• Added cloud SSL authorization support for the Runbooks Automation service, Cleanup service, and Security Audit service.

See also
- Deploy the DevOps Portal manually
- Use the Cleanup service
- Use the Runbooks Automation service

**OpenStack-related features**

Mirantis Cloud Platform now includes the OpenStack Ocata release as well as provides support for some additional OpenStack services.

**Support for OpenStack Ocata**

Mirantis Cloud Platform supports the OpenStack Ocata release and includes enhancements and bug fixes for the world’s leading OpenStack infrastructure deployment and management technologies.

For customers that prefer an older release of OpenStack, MCP can also accommodate the Newton and Mitaka OpenStack releases.

The following table lists the versions of OpenStack Ocata core components that MCP deploys:

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designate</td>
<td>4.0.x</td>
</tr>
<tr>
<td>Glance</td>
<td>14.0.x</td>
</tr>
<tr>
<td>Heat</td>
<td>8.0.x</td>
</tr>
<tr>
<td>Horizon</td>
<td>11.0.x</td>
</tr>
<tr>
<td>Ironic</td>
<td>7.0.x</td>
</tr>
<tr>
<td>Keystone</td>
<td>11.0.x</td>
</tr>
<tr>
<td>Neutron</td>
<td>10.0.x</td>
</tr>
<tr>
<td>Nova</td>
<td>15.0.x</td>
</tr>
</tbody>
</table>

**Support for Ironic**
Added support for Ironic that enables OpenStack bare metal provisioning as opposed to virtual machines provisioning. Ironic provides a unified interface to hardware servers allowing for easy management of physical machines in your MCP OpenStack environment.

Managing and using Ironic include creating suitable images, enrolling bare metal nodes into Ironic and configuring them appropriately, and adding compute flavors that correspond to the available bare metal nodes.

**Seealso**

- Ironic operations

**Support for OpenStack DNS as a Service**

MCP enables you to use the OpenStack Domain Name System as a Service component called Designate to provide DNS with integrated Keystone authentication for OpenStack environments. Designate uses the Galera MySQL cluster as the distributed database to provide a mapping of IP addresses to domain names and hosts to access the Internet.

Designate uses a RESTful API for creating, updating, and deleting DNS zones and records of the OpenStack environments. Designate integrates Nova and Neutron notifications for auto-generated records.

Designate supports Berkely Internet Name Domain (BIND9) and PowerDNS underlying DNS servers out of the box. You can use either a new or existing DNS server as a back end for Designate.

All components except the back end can run on the MCP Virtualized Control Plane (VCP) as a part of the OpenStack API.

**Seealso**

- Plan the Domain Name System
- Deploy Designate
- Designate operations

**Support for sudo policies**

Operating system users and groups can be configured to use sudo aliases. For the existing deployments, we recommend that you update your model with policies for trusted users and set the _parm:linux_system_user_sudo parameter to False afterward:

2. Assign the created sudo aliases to your team, represented by the system group. For example, as in the openssh/server/team/support.yml file of the same repository.
3. Set the \_param:linux_system_user_sudo: parameter in the cluster/CLUSTER_NAME/infra/init.yml file to False.

Note
For a production environment, the \_param:linux_system_user_sudo: parameter must be set to False. During deployment and for any other type of environment, you may not set it or set to True, meaning that any user will have unlimited access to root rights.

Kubernetes-related features
MCP leverages support for the latest Kubernetes 1.7 version that provides significant enhancements in multi-platform support, network policy, support for stateful applications, and extensibility features. MCP can now support up to 500 node scalability for bare metal Kubernetes clusters.

MCP enables you to deploy Kubernetes on bare metal as well as on top of AWS or OpenStack, with a possibility to enable OpenStack cloud provider functionality when deploying Kubernetes on top of OpenStack.

Note
Currently, DriveTrain does not provide an upgrade procedure for Kubernetes from version 1.6 to 1.7.

Virtlet for Kubernetes
MCP enables Kubernetes to run virtual machines using Virtlet. Virtlet is a Kubernetes Container Runtime Interface (CRI) implementation that is packaged as a Docker image and contains such components as libvirt daemon, QEMU/KVM wrapper, and so on. Virtlet is supported by both Calico and Mirantis OpenContrail networking systems.

Virtlet enables you to run unmodified QEMU/KVM virtual machines that do not include an additional Docker layer as in similar solutions in Kubernetes. Virtlet supports all standard Kubernetes objects, such as ReplicaSets, deployments, DaemonSets, and so on, as well as their operations. Virtlet also supports kubectl attach for VM pods that provides access to a VM serial console.

Seealso
- Transition to containers

Netchecker for Kubernetes
Netchecker is a default monitoring application for Kubernetes deployment with Calico networking. Netchecker monitors connectivity between nodes in a Kubernetes cluster.

The main features of Netchecker include:

- An easy to use REST API for verification of networking connectivity
- Providing of metrics to troubleshoot connectivity between the Kubernetes nodes
- Providing of latency probing between selected nodes and selected external hosts
- Testing of networking connectivity using custom payload size

See also

- Monitor connectivity between Kubernetes nodes using Netchecker
- Network checker overview

OpenStack cloud provider for Kubernetes

Enabled Kubernetes to deploy the OpenStack cloud provider that is built into Kubernetes itself. The OpenStack cloud provider allows you to leverage Cinder volumes and Neutron LBaaS to enhance the functionality of the Kubernetes cluster.

The two main functions provided by the OpenStack cloud provider are PersistentVolume for pods and LoadBalancer for services.

See also

- OpenStack cloud provider overview
- Enable OpenStack cloud provider in Kubernetes

ExternalDNS for Kubernetes

ExternalDNS allows setting up a DNS server for a Kubernetes cluster in order to control DNS records dynamically through Kubernetes resources and make Kubernetes resources discoverable through public DNS servers. ExternalDNS synchronizes exposed Kubernetes Services and Ingresses with DNS cloud providers, such as Designate, AWS Route 53, Google CloudDNS, and CoreDNS.

See also

- Deploy ExternalDNS for Kubernetes
Mirantis OpenContrail-related features

Mirantis Cloud Platform includes a number of features related to Mirantis OpenContrail.

Support for OpenContrail 3.2

Mirantis Cloud Platform supports the open source OpenContrail version 3.2 on new OpenStack Ocata MCP deployments to consume the latest innovation from the OpenContrail community and stay with the latest stable release.

New OpenStack Ocata MCP clusters have OpenContrail version 3.2 by default. You can also upgrade your existing Mitaka or Ocata-based MCP cluster with Mirantis OpenContrail version 3.0 or 3.1 to 3.2.

See also
- Upgrade an MCP cluster

Support of OpenStack Heat for OpenContrail

Added support of OpenStack Heat resources for OpenContrail to automate network security policy assignment and run workloads with network filtering protection. You are now able to set the OpenContrail variable using Heat.

By default, the OpenContrail Heat resources are enabled and installed on the OpenStack controller nodes.

OpenStack LBaaS v2 API HAProxy driver for OpenContrail

MCP enables you to deploy and configure OpenStack Load Balancing as a Service (LBaaS) with OpenContrail. Using load balancing allows you to minimize downtime (less than five seconds) during maintenance operations, such as a node removal and an automatic load balancing recovery.

The OpenContrail HAProxy driver provides the benefits of LBaaS v2 API along with listening of multiple ports for the same IP address by decoupling the virtual IP address from the physical port.

See also
- Mirantis OpenContrail HAProxy driver with LBaaSv2

TSN support for Mirantis OpenContrail

Added support for the top-of-rack services node (TSN) to the Mirantis OpenContrail cluster of MCP. Now, you can connect the Mirantis OpenContrail virtual network to a bare metal server through a top-of-rack (ToR) switch. Using this feature on large deployments enhances the
performance of the tenant-to-tenant networking and simplifies communication with the virtual instances that run on the OpenContrail cluster. TSN in MCP is enabled in HA mode and with SSL for the OVSDB configuration.

See also
- Enable TSN support for Mirantis OpenContrail

### NFV features support

Mirantis Cloud Platform supports NFV workloads running on an MCP cloud and includes a number of features related to NFV.

#### Neutron Quality of Service (QoS)

Enabled Neutron Quality of Service, or QoS, a Neutron feature that enables OpenStack administrators to limit and prioritize network traffic through a set of policies for better network bandwidth. By default, Neutron QoS is disabled. You can enable it before or after deploying an OpenStack environment.

See also
- Configure Neutron OVS

#### Pinning of RX queues to specific CPU cores for DPDK ports

Added the ability to map the port RX queues to specific CPU cores. Configuring port queue pinning manually may help to achieve maximum network performance through matching the ports that run specific workloads with specific CPU cores. Each port can process a certain number of Transmit and Receive (RX/TX) operations, therefore it is up to the Network Administrator to decide on the most efficient port mapping.

See also
- Enable OVS DPDK support

#### Network trunking

Enabled network port trunking that allows attaching a virtual machine to multiple Neutron networks using VLANs as a local encapsulation to differentiate traffic for each network as it goes in and out of a single virtual machine network interface (VIF).

Using network trunking is particularly beneficial in the following use cases:
Some applications require connection to hundreds of Neutron networks. To achieve this, you may want to use a single or a few VIFs and VLANs to differentiate traffic for each network rather than having hundreds of VIFs per VM.

Cloud workloads are often very dynamic. You may prefer to add or remove VLANs rather than to hotplug interfaces in a virtual machine.

Moving a virtual machine from one network to another without detaching the VIF from the virtual machine.

A virtual machine may run many containers. Each container may have requirements to be connected to different Neutron networks. Assigning a VLAN or other encapsulation ID for each container is more efficient and scalable than requiring a vNIC per container.

Some legacy applications that require VLANs to connect to multiple networks.

See also

- Enable network trunking
# Major components versions

The following table describes the versions of software that Mirantis Cloud Platform installs.

<table>
<thead>
<tr>
<th>Software</th>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container networking</td>
<td>Calico</td>
<td>2.3.0</td>
</tr>
<tr>
<td>Container platform</td>
<td>Kubernetes</td>
<td>• 1.6</td>
</tr>
<tr>
<td>Container platform configuration</td>
<td>etcd</td>
<td>3.2.2</td>
</tr>
<tr>
<td>Container runtime</td>
<td>Docker</td>
<td>1.13</td>
</tr>
<tr>
<td>Database</td>
<td>MySQL</td>
<td>5.6.23</td>
</tr>
<tr>
<td>Database</td>
<td>PostgreSQL</td>
<td>9.6.2</td>
</tr>
<tr>
<td>Database HA</td>
<td>Galera</td>
<td>25.3.10-1~u14.04+mcp1</td>
</tr>
<tr>
<td>Distributed storage</td>
<td>Ceph</td>
<td>10.2.9 Jewel</td>
</tr>
<tr>
<td>DriveTrain</td>
<td>GlusterFS</td>
<td>3.7</td>
</tr>
<tr>
<td>DriveTrain</td>
<td>aptly</td>
<td>0.9.7</td>
</tr>
<tr>
<td>DriveTrain</td>
<td>Gerrit</td>
<td>2.13.6</td>
</tr>
<tr>
<td>DriveTrain</td>
<td>Git</td>
<td>1:2.7.4-0ubuntu1</td>
</tr>
<tr>
<td>DriveTrain</td>
<td>Jenkins</td>
<td>2.52</td>
</tr>
<tr>
<td>DriveTrain</td>
<td>Jenkins Pipelines</td>
<td>latest</td>
</tr>
<tr>
<td>DriveTrain</td>
<td>Reclass</td>
<td>1.4.1</td>
</tr>
<tr>
<td>DriveTrain</td>
<td>Reclass models</td>
<td>latest</td>
</tr>
<tr>
<td>DriveTrain</td>
<td>Salt</td>
<td>2016.3</td>
</tr>
<tr>
<td>DriveTrain</td>
<td>Salt formulas</td>
<td>latest</td>
</tr>
<tr>
<td>Hypervisor</td>
<td>KVM</td>
<td>2.5</td>
</tr>
<tr>
<td>Hypervisor</td>
<td>libvirt</td>
<td>1.3.1</td>
</tr>
<tr>
<td>Hypervisor</td>
<td>QEMU</td>
<td>2.5</td>
</tr>
<tr>
<td>Load balancer</td>
<td>HAProxy</td>
<td>1.6.3-1~u14.04+mcp1</td>
</tr>
<tr>
<td>Load balancer</td>
<td>NGINX</td>
<td>1.10.0-0ubuntu0.16.0</td>
</tr>
<tr>
<td>Messaging queue service</td>
<td>RabbitMQ</td>
<td>3.6.6-1~u14.04+mcp1</td>
</tr>
<tr>
<td>Mirantis OpenContrail controller</td>
<td>Cassandra</td>
<td>3.0.13</td>
</tr>
<tr>
<td>Service</td>
<td>Version</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------</td>
<td></td>
</tr>
<tr>
<td>HDFS</td>
<td>2.7.2</td>
<td></td>
</tr>
<tr>
<td>Kafka</td>
<td>0.10.2.0</td>
<td></td>
</tr>
<tr>
<td>Redis</td>
<td>3.2</td>
<td></td>
</tr>
<tr>
<td>ZooKeeper</td>
<td>3.5</td>
<td></td>
</tr>
<tr>
<td>OpenStack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ceilometer (Telemetry service)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 8.0.1 (Ocata-based)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 6.1.5 (Mitaka-based)</td>
<td></td>
</tr>
<tr>
<td>Cinder (Block storage service)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 10.0.3 (Ocata-based)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 8.1.1 (Mitaka-based)</td>
<td></td>
</tr>
<tr>
<td>Glance (Image service)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 14.0.0 (Ocata-based)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 12.0.0 (Mitaka-based)</td>
<td></td>
</tr>
<tr>
<td>Heat (Orchestration service)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 8.0.1 (Ocata-based)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 6.1.2 (Mitaka-based)</td>
<td></td>
</tr>
<tr>
<td>Horizon (Dashboard service)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 11.0.3 (Ocata-based)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 9.1.2 (Mitaka-based)</td>
<td></td>
</tr>
<tr>
<td>Ironic (Bare metal service)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 7.0.2 (Ocata-based)</td>
<td></td>
</tr>
<tr>
<td>Keystone (Identity service)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 11.0.2 (Ocata-based)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 9.3.0 (Mitaka-based)</td>
<td></td>
</tr>
<tr>
<td>Neutron (Network service)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 10.0.2 (Ocata-based)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 8.4.0 (Mitaka-based)</td>
<td></td>
</tr>
<tr>
<td>Nova (Compute service)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 15.0.6 (Ocata-based)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 13.1.4 (Mitaka-based)</td>
<td></td>
</tr>
<tr>
<td>Operating system</td>
<td>Ubuntu</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Xenial Xerus (16.04)</td>
<td></td>
</tr>
<tr>
<td>StackLight</td>
<td>Alertmanager</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.6.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>collectd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.4, 5.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grafana</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.3.2</td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td>Version</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>Elasticsearch</td>
<td>2.4.5</td>
<td></td>
</tr>
<tr>
<td>Heka</td>
<td>0.10.1</td>
<td></td>
</tr>
<tr>
<td>InfluxDB</td>
<td>1.3.3-1</td>
<td></td>
</tr>
<tr>
<td>Kibana</td>
<td>4.6.4</td>
<td></td>
</tr>
<tr>
<td>Prometheus</td>
<td>1.6.3</td>
<td></td>
</tr>
<tr>
<td>Sensu</td>
<td>0.26</td>
<td></td>
</tr>
<tr>
<td>Telegraf</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td>Uchiwa</td>
<td>0.22</td>
<td></td>
</tr>
<tr>
<td>Virtual switching fabric</td>
<td>Mirantis OpenContrail</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 3.2.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 3.1.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Open vSwitch</td>
<td>2.6</td>
</tr>
<tr>
<td>Other</td>
<td>Spinnaker</td>
<td>1.1.2</td>
</tr>
</tbody>
</table>

1. See MCP Reference Architecture for exceptions.
2. Telegraf is based on version 1.2 but includes enhancements from version 1.3.
Resolved issues

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

• Fixed the issue with injection metadata disappearing after resizing an instance.
• Fixed the failure in evacuation of a virtual machine to another node.
• Allowed irrelevant, self-defined, specifications in ComputeCapacityFilter. LP1582589
• Fixed the failure caused by absence of the wait attribute in the NetworkInfo object when rescheduling a virtual machine. LP1636109
• Fixed the issue with starting instances using SR-IOV with a multi-provider, or multi-segment, network. LP1659467
• Fixed the issue with cpu_thread_policy=prefer misbehavior. LP1578155
• Fixed the NUMA scheduling error on evacuation when the evacuated VM is scheduled on a compute node where a vCIC is hosted. LP1658982
• Fixed the issue with throwing uninformative Unexpected Exception from neutronclient for an expired or invalid token. In case of the admin context, the Nova exception handler logs a message to inform operators that the Neutron admin credential are not well configured then return the 500 response code to user. In case of the user context, it returns the 401 response code. LP1571722
• Fixed the issue with instances evacuation failing with the following error message: Virtual Interface creation failed. LP1590490
• Fixed the issue caused by the BDM race in get_available_resource(). LP1602057
• Fixed the issue with numerous logging of the following error message: Migration for instance refers to another host’s instance! during resize or cold migration when migration is finished but not confirmed or reverted. LP1555320
• Fixed the failure in instance creation after deletion of a host aggregate. LP1605804
• Fixed the issue with dropping RARP packets by neutron-ovs-agent when QEMU is live-migrating a virtual machine. LP1414559
• Fixed the issue with getting public flavors to create instance when running OpenStack Dashboard in Internet Explorer 11.
• Fixed the issue in swiftclient failing with the following error message: Unable to get the Swift service info.
• Fixed the issue with creation a security group, network, router, and key pair in the OpenStack Dashboard service.
• Fixed the issue with mounting more than seven volumes when using virtio-scsi. LP1686116
• Fixed the issue with aodh-expirer default installation.
• Fixed the issue with the reconnection delay caused by oslo.messaging trying to close the channel belonging to a dropped connection. LP1689801
• Fixed the issue with pagination on the Instances page.
• Fixed the issue with Ceilometer installation caused by the outdated version of python-cotyledon.
OpenContrail

• Fixed the security issue with running the Cassandra database as root.
• Fixed the security issue with showing a Compute node address in the traceroute command's output.
SaltStack

- Fixed the idempotence issue with the Salt opencontrail state.
- Fixed the issue in the Salt master state failing with the following error message: Jinja error: ‘instancemethod’ object is not iterable.
- Fixed the endless recursion in salt-formula-collectd version a1c0470.
StackLight

- Fixed the issue with frequent polling of the GlusterFS daemon by StackLight. The polling interval for GlusterFS has been increased.
- Fixed the issue with generation of Sensu alarms when log_collector is hanging on a node.
- Fixed the Heka and Collectd formulas in the legacy version of StackLight.
- Fixed the issue with the time field in InfluxDB.
- Fixed the incompatibility issue with running Collectd built for Ubuntu Xenial in the Linux kernel version 4.8.0.
Others

• Fixed the CPU overconsumption issue after GlusterFS logs rotation.
• Fixed the issue with GlusterFS logs rotation.
• Updated MySQL to the version 5.6.36.
• Fixed the CI/CD issue with the incorrectly calculated pass rate by the openstack-component-checker job.

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.

Prometheus server crashes

• Description: Occasionally, the Prometheus server may crash with the panic: runtime error: makeslice: len out of range error message after remote_storage_adapter recovers after a failure. See Issue #2969.

• Workaround: No manual intervention is required. Docker Swarm automatically restarts Prometheus.

Connection timeouts between Prometheus and the remote storage adapter

• Description: Large clusters with about 400 nodes can experience connection timeouts between Prometheus and the remote storage adapter.

Syslog messages are not properly handled

• Description: Syslog messages are not properly parsed, which may lead to decoder errors on heka.log_collector. Therefore, such messages may not reach Elasticsearch.

Tenant renaming causes errors in the Contrail web UI

• Description: OpenContrail does not support tenant renaming due to architecture limitations.

The Contrail Kafka service may contain wrong replication factor

• Description: The Contrail Kafka service configuration may contain a wrong value for the ReplicationFactor: parameter. This can result in Kafka instances having limited HA. To verify whether your MCP clusters are affected:

```
cat /usr/share/kafka/config/server.properties | grep factordefault.replication.factor
/usr/share/kafka/bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic -uve-0
```

If the output of the above command contains ReplicationFactor:1, your clusters are affected.

• Workaround:

1. Update the OpenContrail formula to the latest version.
2. Delete all affected User-Visible Entities (UVEs):

```
/usr/share/kafka/bin/kafka-topics.sh --delete --zookeeper localhost:2181 --topic -uve-0
```
3. Verify that the replication factor changed for all affected UVEs:

```
/usr/share/kafka/bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic -uve-0
```

Example:

```
root@nal-02:~# /usr/share/kafka/bin/kafka-topics.sh --delete --zookeeper localhost:2181 --topic -uve-0
Topic -uve-0 is marked for deletion.
```
Note: This will have no impact if `delete.topic.enable` is not set to true.

root@nal-02:~# /usr/share/kafka/bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic -uve-0

Topic:-uve-0  PartitionCount:1  ReplicationFactor:2  Configs:
Topic: -uve-0  Partition: 0  Leader: 1  Replicas: 1,2  Isr: 1,2