MCP Q1`18 Release Notes

version q1-18
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.
Release artifacts

The latest MCP release artifacts are tagged with the 2018.4.0 release version tag including APT repository snapshots, Git repository tags, and Docker image versions.

The combination of versions of MCP components that can be installed using the artifacts tagged with the 2018.4.0 release version tag are listed in Major components versions. This versions combination has passed integration testing and is 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 2018.4 Encryption and Licensing.

<table>
<thead>
<tr>
<th>Type</th>
<th>Artifact</th>
<th>Path</th>
</tr>
</thead>
</table>
| Mirantis apt/deb packages | OpenContrail packages       | deb http://apt.mirantis.com/xenial/ 2018.4.0 oc32  
|                    |                                | deb http://apt.mirantis.com/trusty/ 2018.4.0 oc32 |
|                    | OpenStack packages             | deb http://apt.mirantis.com/trust//openstack/mitaka/ 2018.4.0 main  
|                    |                                | deb http://apt.mirantis.com/xenial/openstack/ocata/ 2018.4.0 main  
|                    |                                | deb http://apt.mirantis.com/xenial/openstack/pike/ 2018.4.0 main  
|                    |                                | deb http://apt.mirantis.com/trusty/ 2018.4.0 mitaka  
|                    |                                | deb http://apt.mirantis.com/xenial/ 2018.4.0 ocata |
|                    | Extra packages                 | deb http://apt.mirantis.com/xenial/ 2018.4.0 extra |
|                    | Salt formulas packages         | deb http://apt.mirantis.com/xenial/ 2018.4.0 salt |
| QOW images         | MCP cfg01 day01 image          | http://images.mirantis.com/cfg01-day01-2018.4.0.qcow2  
|                    |                                | http://images.mirantis.com/cfg01-day01-2018.4.0.qcow2.md5 |
|                    | MCP apt01 offline image        | http://images.mirantis.com/mcp-offline-image-2018.4.0.qcow2  
<p>|                    |                                | <a href="http://images.mirantis.com/mcp-offline-image-2018.4.0.qcow2.md5">http://images.mirantis.com/mcp-offline-image-2018.4.0.qcow2.md5</a> |</p>
<table>
<thead>
<tr>
<th><strong>VCP Ubuntu 16.04 image</strong></th>
<th><strong><a href="http://images.mirantis.com/ubuntu-16-04-x64-mcp2018.4.0.qcow2">http://images.mirantis.com/ubuntu-16-04-x64-mcp2018.4.0.qcow2</a></strong>&lt;br&gt;<strong><a href="http://images.mirantis.com/ubuntu-16-04-x64-mcp2018.4.0.qcow2.md5">http://images.mirantis.com/ubuntu-16-04-x64-mcp2018.4.0.qcow2.md5</a></strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Upstream mirrors</strong></td>
<td><strong>aptly</strong>&lt;br&gt;<strong>deb <a href="http://mirror.mirantis.com/2018.4.0/aptly/xenial">http://mirror.mirantis.com/2018.4.0/aptly/xenial</a> squeeze main</strong></td>
</tr>
<tr>
<td><strong>Cassandra</strong></td>
<td><strong>deb <a href="http://mirror.mirantis.com/2018.4.0/cassandra/trusty">http://mirror.mirantis.com/2018.4.0/cassandra/trusty</a> 21x main</strong>&lt;br&gt;<strong>deb <a href="http://mirror.mirantis.com/2018.4.0/cassandra/xenial">http://mirror.mirantis.com/2018.4.0/cassandra/xenial</a> 21x main</strong></td>
</tr>
<tr>
<td><strong>Ceph</strong></td>
<td><strong>deb <a href="http://apt.mirantis.com/xenial">http://apt.mirantis.com/xenial</a> 2018.4.0 ceph-jewel</strong>&lt;br&gt;<strong>deb <a href="http://apt.mirantis.com/xenial">http://apt.mirantis.com/xenial</a> 2018.4.0 ceph-luminous</strong></td>
</tr>
<tr>
<td><strong>Docker</strong></td>
<td><strong>deb <a href="http://mirror.mirantis.com/2018.4.0/docker/xenial">http://mirror.mirantis.com/2018.4.0/docker/xenial</a> xenial stable</strong>&lt;br&gt;<strong>deb <a href="http://mirror.mirantis.com/2018.4.0/docker-1.x/xenial">http://mirror.mirantis.com/2018.4.0/docker-1.x/xenial</a> ubuntu-xenial main</strong></td>
</tr>
<tr>
<td><strong>Fluentd</strong></td>
<td><strong>deb <a href="http://mirror.mirantis.com/2018.4.0/td-agent/xenial">http://mirror.mirantis.com/2018.4.0/td-agent/xenial</a> xenial contrib</strong></td>
</tr>
<tr>
<td><strong>GlusterFS</strong></td>
<td><strong>deb <a href="http://mirror.mirantis.com/2018.4.0/glusterfs-3.8/xenial">http://mirror.mirantis.com/2018.4.0/glusterfs-3.8/xenial</a> xenial main</strong></td>
</tr>
<tr>
<td><strong>InfluxDB</strong></td>
<td><strong>deb <a href="http://mirror.mirantis.com/2018.4.0/influxdb/xenial">http://mirror.mirantis.com/2018.4.0/influxdb/xenial</a> xenial stable</strong></td>
</tr>
<tr>
<td><strong>Kibana</strong></td>
<td><strong>deb <a href="http://mirror.mirantis.com/2018.4.0/kibana-4.6/xenial">http://mirror.mirantis.com/2018.4.0/kibana-4.6/xenial</a> stable main</strong></td>
</tr>
<tr>
<td><strong>MAAS</strong></td>
<td><strong>deb <a href="http://mirror.mirantis.com/2018.4.0/maas/xenial">http://mirror.mirantis.com/2018.4.0/maas/xenial</a> xenial main</strong></td>
</tr>
<tr>
<td><strong>SaltStack packages</strong></td>
<td><strong>deb <a href="http://mirror.mirantis.com/2018.4.0/saltstack-2016.3/trusty">http://mirror.mirantis.com/2018.4.0/saltstack-2016.3/trusty</a> trusty main</strong>&lt;br&gt;<strong>deb <a href="http://mirror.mirantis.com/2018.4.0/saltstack-2016.3/xenial">http://mirror.mirantis.com/2018.4.0/saltstack-2016.3/xenial</a> xenial main</strong></td>
</tr>
</tbody>
</table>
| Upstream Ubuntu system packages | deb https://mirror.mirantis.com/2018.4.0/ubuntu/ xenial
deb https://mirror.mirantis.com/2018.4.0/ubuntu/xenial-updates
deb https://mirror.mirantis.com/2018.4.0/ubuntu/xenial-security |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MCP Git repositories</td>
<td>Jenkins pipeline library for MCP operations</td>
</tr>
<tr>
<td>General Jenkins pipeline library</td>
<td><a href="https://github.com/Mirantis/pipeline-library">https://github.com/Mirantis/pipeline-library</a> tag 2018.4.0</td>
</tr>
<tr>
<td>Reclass system level</td>
<td><a href="https://github.com/Mirantis/reclass-system-salt-model">https://github.com/Mirantis/reclass-system-salt-model</a> tag 2018.4.0</td>
</tr>
<tr>
<td>MCP common scripts</td>
<td><a href="https://github.com/Mirantis/mcp-common-scripts/">https://github.com/Mirantis/mcp-common-scripts/</a> tag 2018.4.0</td>
</tr>
<tr>
<td>MCP offline image model</td>
<td><a href="https://github.com/Mirantis/mcp-offline-model">https://github.com/Mirantis/mcp-offline-model</a> tag 2018.4.0</td>
</tr>
</tbody>
</table>

Docker images

| Alertmanager | docker-prod-local.artifactory.mirantis.com/openstack-doc
ker/alertmanager:2018.4.0 |
| Apty | docker-prod-local.artifactory.mirantis.com/mirantis/cicd/ap
ty:2018.4.0 |
| Apty-api | docker-prod-local.artifactory.mirantis.com/mirantis/cicd/ap
ty-api:2018.4.0 |
| Apty-public | docker-prod-local.artifactory.mirantis.com/mirantis/cicd/ap
ty-public:2018.4.0 |
| Apty-publisher | docker-prod-local.artifactory.mirantis.com/mirantis/cicd/ap
ty-publisher:2018.4.0 |
| Cluster-proportional-autoscaler-amd64 | docker-prod-local.artifactory.mirantis.com/mirantis/extern
al/cluster-proportional-autoscaler-amd64:2018.4.0 |
| Compose | docker-prod-local.artifactory.mirantis.com/mirantis/extern
al/compose:2018.4.0 |
| Contrail-cni | docker-prod-local.artifactory.mirantis.com/mirantis/kuber
etes/contrail-integration/contrail-cni:2018.4.0 |
| Contrail-network-controller | docker-prod-local.artifactory.mirantis.com/mirantis/kuber
etes/contrail-integration/contrail-network-controller:201
8.4.0 |
| Ctl | docker-prod-local.artifactory.mirantis.com/mirantis/projec
tcalico/calico/ctl:2018.4.0 |
| Cni | docker-prod-local.artifactory.mirantis.com/mirantis/projec
tcalico/calico/cni:2018.4.0 |
<table>
<thead>
<tr>
<th>Package</th>
<th>Repository Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>devops-portal</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/oss/devops-portal:2018.4.0</td>
</tr>
<tr>
<td>elasticsearch</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/elasticsearch:2018.4.0</td>
</tr>
<tr>
<td>etcd</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/etcd:2018.4.0</td>
</tr>
<tr>
<td>etcd-operator</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/etcd-operator:2018.4.0</td>
</tr>
<tr>
<td>gerrit</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/cicd/gerrit:2018.4.0</td>
</tr>
<tr>
<td>grafana</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/grafana:2018.4.0</td>
</tr>
<tr>
<td>heka</td>
<td>docker-prod-local.artifactory.mirantis.com/openstack-doc docker/heka:2018.4.0</td>
</tr>
<tr>
<td>hyperkube-amd64</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/kubernetes/hyperkube-amd64:2018.4.0</td>
</tr>
<tr>
<td>janitor-monkey</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/oss/janitor-monkey:2018.4.0</td>
</tr>
<tr>
<td>jenkins</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/cicd/jenkins:2018.4.0</td>
</tr>
<tr>
<td>jnlp-slave</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/cicd/jnlp-slave:2018.4.0</td>
</tr>
<tr>
<td>k8s-dns-kube-dns-amd64</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/k8s-dns-kube-dns-amd64:2018.4.0</td>
</tr>
<tr>
<td>k8s-dns-dnsmasq-amd64</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/k8s-dns-dnsmasq-amd64:2018.4.0</td>
</tr>
<tr>
<td>k8s-dns-sidecar-amd64</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/k8s-dns-sidecar-amd64:2018.4.0</td>
</tr>
<tr>
<td>k8s-netchecker-agent</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/kubernetes/k8s-netchecker-agent:2018.4.0</td>
</tr>
<tr>
<td>k8s-netchecker-server</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/kubernetes/k8s-netchecker-server:2018.4.0</td>
</tr>
<tr>
<td>kube-policy-controller</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/kube-policy-controller:2018.4.0</td>
</tr>
<tr>
<td>kubernetes-dashboard-amd64</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/kubernetes-dashboard-amd64:2018.4.0</td>
</tr>
<tr>
<td>mongo</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/mongo:2018.4.0</td>
</tr>
<tr>
<td>mysql</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/cicd/mysql:2018.4.0</td>
</tr>
<tr>
<td>Image</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>node</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/projectcalico/calico/node:2018.4.0</td>
</tr>
<tr>
<td>openldap</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/openldap:2018.4.0</td>
</tr>
<tr>
<td>pause</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/pause:2018.4.0</td>
</tr>
<tr>
<td>phpldapadmin</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/cicd/phpldapadmin:2018.4.0</td>
</tr>
<tr>
<td>postgres</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/postgres:2018.4.0</td>
</tr>
<tr>
<td>prometheus</td>
<td>docker-prod-local.artifactory.mirantis.com/openstack-docker/prometheus:2018.4.0</td>
</tr>
<tr>
<td>prometheus_relay</td>
<td>docker-prod-local.artifactory.mirantis.com/openstack-docker/prometheus_relay:2018.4.0</td>
</tr>
<tr>
<td>pushgateway</td>
<td>docker-prod-local.artifactory.mirantis.com/openstack-docker/pushgateway:2018.4.0</td>
</tr>
<tr>
<td>pushkin</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/oss/pushkin:2018.4.0</td>
</tr>
<tr>
<td>qa-tools</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/oss/qa-tools:2018.4.0</td>
</tr>
<tr>
<td>registry</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/registry:2018.4.0</td>
</tr>
<tr>
<td>remote_storage_adapter</td>
<td>docker-prod-local.artifactory.mirantis.com/openstack-docker/remote_storage_adapter:2018.4.0</td>
</tr>
<tr>
<td>rundeck</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/oss/rundeck:2018.4.0</td>
</tr>
<tr>
<td>security-monkey-api</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/oss/security-monkey-api:2018.4.0</td>
</tr>
<tr>
<td>security-monkey-scheduler</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/oss/security-monkey-scheduler:2018.4.0</td>
</tr>
<tr>
<td>telegraf</td>
<td>docker-prod-local.artifactory.mirantis.com/openstack-docker/telegraf:2018.4.0</td>
</tr>
<tr>
<td>tiller</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/tiller:2018.4.0</td>
</tr>
<tr>
<td>virtlet</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/kubernetes/virtlet:2018.4.0</td>
</tr>
<tr>
<td>visualizer</td>
<td>docker-prod-local.artifactory.mirantis.com/mirantis/external/visualizer:2018.4.0</td>
</tr>
</tbody>
</table>
## Major components versions

The following table lists the MCP components of the Q1`18 Release Version with Build ID 2018.4.0 that are initially installed by default.

### Software components and versions - Build ID 2018.4.0

<table>
<thead>
<tr>
<th>Component</th>
<th>Application/service</th>
<th>Version</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed storage</td>
<td>Ceph</td>
<td>12.2.4-1xenial</td>
<td>Luminous</td>
</tr>
<tr>
<td>DriveTrain</td>
<td>Aptly</td>
<td>1.2.0</td>
<td>Resource type - Docker image</td>
</tr>
<tr>
<td></td>
<td>aptly-publisher</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>Resource type - Docker image</td>
</tr>
<tr>
<td></td>
<td>jenkins-master</td>
<td>2.100</td>
<td>Resource type - Docker image</td>
</tr>
<tr>
<td></td>
<td>Jenkins pipeline-library</td>
<td>2018.4.0</td>
<td>Resource type - Git repository</td>
</tr>
<tr>
<td></td>
<td>Reclass</td>
<td>1.4.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reclass model</td>
<td>2018.4.0</td>
<td>Resource type - Git repository</td>
</tr>
<tr>
<td></td>
<td>Salt</td>
<td>2016.3.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Salt formulas</td>
<td>2018.4.0</td>
<td>Resource type - binary repository only</td>
</tr>
<tr>
<td>Hypervisors</td>
<td>libvirt</td>
<td>4.0.0-1.7~u16.04+mcp1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>qemu</td>
<td>1:2.11+dfsg-1.1~u16.04+mcp1</td>
<td>qemu-kvm</td>
</tr>
<tr>
<td>Kubernetes</td>
<td>Calico</td>
<td>2.6.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Calico CNI</td>
<td>1.11.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>etcd</td>
<td>3.3.x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kubernetes</td>
<td>1.8.11-9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Virtlet</td>
<td>1.0.0</td>
<td></td>
</tr>
<tr>
<td>OpenContrail networking</td>
<td>Cassandra</td>
<td>2.1.20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kafka</td>
<td>2.9.2-0.8.2.0-0contrail0</td>
<td></td>
</tr>
<tr>
<td>Component</td>
<td>Version</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>OpenContrail</td>
<td>3.2.3.x, 3.2.10.0</td>
<td>OpenContrail 3.2.x does not support the OpenStack Pike release.</td>
<td></td>
</tr>
<tr>
<td>Redis</td>
<td>2.8.4-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zookeeper</td>
<td>3.4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenStack Pike</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Barbican</td>
<td>5.0.x ¹</td>
<td>MCP officially supports Pike, Ocata, and Mitaka OpenStack releases.</td>
<td></td>
</tr>
<tr>
<td>Cinder</td>
<td>11.1.x ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Designate</td>
<td>5.0.1 ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glance</td>
<td>15.0.x ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heat</td>
<td>9.0.x ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Horizon</td>
<td>12.0.x ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ironic</td>
<td>9.1.x ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keystone</td>
<td>12.0.x ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manila</td>
<td>5.0.1 ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nova</td>
<td>16.1.x ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenStack Networking</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGP VPN</td>
<td>7.0.x ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2 Gateway</td>
<td>11.0.x ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Networking ODL</td>
<td>11.0.x ¹</td>
<td>neutron-plugin-ml2</td>
<td></td>
</tr>
<tr>
<td>ML2 plugin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neutron</td>
<td>11.0.x ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenStack Telemetry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aodh</td>
<td>5.x ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ceilometer</td>
<td>9.0.5 ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Panko</td>
<td>3.1.x ¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gnocchi</td>
<td>4.0.4 ¹</td>
<td>For Telemetry</td>
<td></td>
</tr>
<tr>
<td>StackLight LMA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alertmanager</td>
<td>0.11.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elasticsearch</td>
<td>5.6.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fluentd</td>
<td>1.0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grafana</td>
<td>4.5.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heka</td>
<td>0.10.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>InfluxDB</td>
<td>1.5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td>Version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jmx-exporter</td>
<td>0.10~20170912072743</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kibana</td>
<td>5.6.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>libvirt-exporter</td>
<td>0.1~20180319200151</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prometheus</td>
<td>2.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pushgateway</td>
<td>0.4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Telegraf</td>
<td>1.5.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StackLight OSS</td>
<td>Janitor Monkey 2.5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MongoDB</td>
<td>3.4.x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pushkin</td>
<td>0.1.6.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rundeck</td>
<td>2.7.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Monkey</td>
<td>0.9.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Monkey OpenStack</td>
<td>0.1.dev0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System</td>
<td>Docker 17.12.x, 1.12.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>17.12.x on StackLight and CI/CD nodes, 1.12.6 on Kubernetes nodes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Git 2.7.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GlusterFS 3.8.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HAProxy 1.6.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NGINX 1.10.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OpenLDAP 2.4.40</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
|                          | Open vSwitch 2.8.0-1~u16.04+mc
                          |   p2 openvswitch-common, openvswitch-switch, python-openvswitch |
|                          | Galera 25.3.14-1             |
|                          | MySQL 5.6.35                 |
|                          | RabbitMQ rabbitmq-server 3.6
                          |   15-2~u16.04+mcp1           |
|                          | Base OS Ubuntu Xenial - 4.4
                          |   0.116 linux-image-virtual-lts-xen
                          |   ial,linux-image-generic-lts-
                          |   xenial                      |
|                          | OS for HW nodes Ubuntu Xenial - 4.4
                          |   0.116 linux-image-generic-lts-xenial |

1(1, 2, 3, 4, 5), The minor version of the component has been updated in 2018.4.1. See 6, 7, 8, 9, 10, Updated packages for details. 11, 12, 13, 14, 15, 16, 17, 18)
What’s new

The latest Mirantis Cloud Platform (MCP) release continues to focus on Day-2 operations with support for the OpenStack Pike version along with the Keystone API v3, Telemetry, Manila, TLS encryption, and a number of OpenStack Neutron plugins. In addition, MCP provides support for newer versions of Kubernetes and Virtlet as well as enhancements for the Day-1 operations and StackLight LMA.

More specifically, the latest MCP introduces the following features and enhancements.

DriveTrain

The latest MCP release contains the following enhancements for the DriveTrain life-cycle management toolchain.

Offline image optimization

Optimized user experience by reducing the size of the offline image by approximately 140 GB. The latest offline image contains the reduced number of mirrors for the deb packages repositories and does not include the mirrors for Python and debug packages and all duplicated mirrors that are concurrently provided by Mirantis.

MCP upgrade

Implemented the Deploy - update cloud Jenkins pipeline allowing for the upgrade of your MCP deployment to a certain MCP release version, which is a stable and product-ready combination of versions of MCP components tagged with a specific Build ID. The same pipeline can be used to roll back an MCP deployment to a previous MCP release version.

Host operating system with LDAP integration

Added the capability to integrate your external LDAP server with Keystone and a host operating system. This configuration is supported by Reclass as an optional setup.
Learn more

MCP Deployment Guide: Configure LDAP integration with MCP

Disk partitioning
Added support for disk partitioning and software RAID configuration to the MAAS Salt formula. Now, you can design the disk layout as well as partitioning schema per node depending on the needs of your deployment.

Learn more

MCP Deployment Guide: Add a custom disk layout per node in the MCP model

Watchdog
Added the possibility to enable watchdog that detects and recovers servers from serious malfunctions. This functionality can be implemented through either a watchdog timer, which is a hardware device, or a software-only softdog driver.

Learn more

MCP Deployment Guide: Enable a watchdog

Physical servers scaling
Added the ability to scale the number of physical servers in the MCP infrastructure including KVM nodes hosting the VCP, compute, and gateway nodes. Also, documented the procedure of moving the OpenStack controller nodes to other physical hosts.

Learn more

- MCP Operations Guide: Manage controller nodes
- MCP Operations Guide: Manage compute nodes
- MCP Operations Guide: Manage gateway nodes
- MCP Operations Guide: Move a VCP node to another host
PXE booting over UEFI

Documented the procedure of configuring the Preboot Execution Environment (PXE) to boot a hardware server from the network over Unified Extensible Firmware Interface (UEFI). The related section instructs on how to configure a physical server not yet loaded with an operating system to use UEFI on boot time during the remote provisioning by MAAS.

Learn more

MCP Deployment Guide: Configure PXE booting over UEFI

OpenStack Pike support

Added support for the OpenStack Pike release including enhancements and bug fixes for the world’s leading OpenStack infrastructure deployment and management technologies along with the Keystone API v3 and StackLight LMA support.

For the users who prefer older releases of OpenStack, MCP can also accommodate the Mitaka and Ocata OpenStack releases. For the support schedule, see MCP OpenStack Releases.

OpenStack

Mirantis Cloud Platform now supports the OpenStack Pike release with OpenStack Telemetry and Manila services as well as includes a number of Neutron plugins, TLS encryption, and NTP authentication support.

Telemetry

Onboarded the OpenStack Telemetry into MCP with OpenStack Pike deployments including the following components:

- Aodh - the alarming service
- Ceilometer - the data collection service
- Panko - the metadata indexing, event storage service
- Gnocchi - the time-series database, resource indexing service

Starting from OpenStack Pike, deprecated Telemetry that uses the legacy StackLight LMA back ends.

Warning

Telemetry for OpenStack Pike does not support integration with StackLight LMA.
Manila
Added support for Manila, also known as the OpenStack Shared File Systems service, to provide coordinated access to shared or distributed file systems that a compute instance can consume.

Learn more
MCP Deployment Guide: Deploy Manila

Multi-node mode for Dogtag
Added the capability to deploy Dogtag for Barbican in a multi-node mode to provide high availability.

Learn more
MCP Deployment Guide: Deploy Dogtag

Neutron BGP VPN
Added support for the Neutron BGP VPN Interconnection service that allows triggering the establishment of connectivity between VMs and your existing BGP IP VPN (a set of external sites) setup outside the data center.
Also, added support for the BaGPipe driver that is a lightweight implementation of the BGP-based VPNs used as a reference back end for the Neutron BGP VPN Interconnection service.

Learn more
MCP Operations Guide: Configure BGP VPN

L2 Gateway plugin
Added support for the L2 Gateway (L2GW) plugin for the Neutron service that enables interconnection between a given tenant network and a VLAN on a physical switch.

Learn more

*MCP Operations Guide: Enable L2 Gateway support*

### Networking OpenDaylight ML2 plugin

Added support for the Neutron Networking OpenDaylight (NW-ODL) ML2 plugin that allows for OpenStack integration with OpenDaylight.

Learn more

*MCP Operations Guide: Enable the Networking NW-ODL ML2 plugin*

### Transport Layer Security (TLS)

Added support for the Transport Layer Security (TLS) protocol that assures the confidentiality and integrity of network traffic inside an OpenStack Pike deployment.

More specifically, the implemented enhancements include:

- Enabled the TLS cryptographic protocol for the client-server OpenStack applications as well as for the RabbitMQ and MySQL back ends.
- Implemented the ability to encrypt the internal OpenStack API HTTP with TLS.
- Added the explicit definition of the TLS usage for the Push Notification service (StackLight OSS). This option allows working correctly with custom SMTP servers that use a non-TLS connection.

To enable the option, set the `pushkin_smtp_use_tls` variable on the cluster level of the Reclass model:

```yaml
parameters:
  _param:
    pushkin_smtp_use_tls: true
```
NTP authentication

Added the ability to secure the Network Time Protocol (NTP) service by enabling authentication for it. NTP allows for proper clock synchronization of services among OpenStack nodes.

StackLight LMA

StackLight LMA introduces a set of enhancements and changes.

Fluentd service

Implemented the Fluentd service as part of the logging system in StackLight LMA. Fluentd collects logs from various data sources, sends them to Elasticsearch, generates metrics based on the analysis of incoming log entries, and exposes these metrics to Prometheus. Typically, Fluentd implements the same log collecting functionality as Heka but provides a number of benefits such as dynamic development, support for all modern software, flexibility, and ease of use.

To migrate your Prometheus-based StackLight LMA installation to Fluentd, see MCP Operations Guide: Migrate from Heka to Fluentd.

The enhancements of the Fluentd service also include:

- Added the capability to specify the log rotation size for Fluentd td-agent using the `td_agent_log_rotate_size` parameter. This prevents a single log file from taking too much disk space. This parameter is set to 10 MB by default.
- Improved the Prometheus web UI by implementing the host label for Fluentd, Telegraf, and exporters targets. Such approach simplifies the identification of issues.

Learn more

- MCP Reference Architecture: StackLight components
**Prometheus long-term storage**

Implemented the StackLight long-term storage based on Prometheus and added the capability to change the long-term storage type from InfluxDB to the Prometheus-based one.

**Learn more**

- MCP Reference Architecture: StackLight LMA overview
- MCP Reference Architecture: StackLight LMA components
- MCP Operations Guide: Change the long-term storage type

**Kubernetes**

The latest MCP release provides the following improvements for Kubernetes:

- Added support for the Kubernetes v1.8.11-9 that provides enhancements and bug fixes of the community Kubernetes version. For a list of new features, see [Kubernetes 1.8 release notes](#).
- Upgraded Virtlet to version 1.0.0. For details, see [Virtlet releases page](#).
Resolved issues
This section provides the list of the addressed issues in the current MCP release version.

DriveTrain

• Fixed the network issue in Docker Swarm that caused the Jenkins slaves fail when deploying the Jenkins stack on the cid nodes.
• Fixed the issues with misconfiguration of the Cassandra and Zookeeper backups.
• Fixed the Ceph versions issue that caused the OpenStack compute nodes to be deployed with Ceph Jewel instead of Ceph Luminous.
• Fixed the issue with the NGINX Salt formula failing to apply new certificate files if the prx nodes already contain certificate files with the same names. Now, if the source files change, the certificate chain file will be recreated.
• Fixed the debconf dependency for the Pluggable Authentication Module (PAM) in LDAP by installing debconf-utils.
• Fixed the issue with Jenkins ignoring the reachability of the minions and attempting to continue the deployment while the minions are unreachable.
• Fixed the issue with cloud-init failing to run during the Salt Master node bootstrapping in case of an offline deployment.
• Fixed the issue with the prebuilt mirror image missing the virtinst packages. However, consider using XML definitions supported by default instead of virtinst.

OpenStack

• Fixed the issue with assigning a security group to an interface failing with the ‘Port’ object has no attribute ‘security_groups’ error message. The issue affected only the OpenStack Ocata version.
• Fixed the validation failure for Heat templates that have three or more nested levels.
• Fixed the issue with the horizon-mirantis-theme and openstack-dashboard-contrail-panels packages breaking the installation of the openstack-dashboard package.
• Fixed the security issue with the VNC console listening on all networks of a compute node without any access restrictions in iptables. This vulnerability allowed accessing a remote VM console by an unauthorized user.
• Fixed the issue with Horizon making duplicate and extra requests.
• Fixed the issue with the Keystone catalog containing endpoints for the deprecated EC2 API service by removing these endpoints.

OpenContrail

• Fixed the issue with the “NoSuchProcess: process name:cassandra pid:XXXX” error in the OpenContrail logs.
• Fixed the security issue in the OpenContrail API not requiring authentication or authorization to create, modify, or delete any network resource.

• Fixed the incompatibility issue of the python-docker v1.9 package with the upstream Ubuntu python-backports.ssl-match-hostname v3.4 package that prevented the creation of namespaces for SNAT and the loadbalancer service instances in OpenContrail.

**StackLight LMA**

• Fixed the issue with the false appearance of the NovaLibvirtDown alert caused by the exporter failing to collect the data about at least one instance from the node.

• Reduced the number of metrics that Telegraf exposes to Prometheus to fix the issue with a mon node failing because of the Ceph metrics consuming all the available memory.

• Fixed the issue with Ceilometer failing to collect the data from `metadata.user_metadata.stack`.

• Fixed the high availability issue by implementing a check for HAProxy in Keepalived. Previously, if the HAProxy process stopped, the VIP could still be assigned to a node with the stopped HAProxy service. This resulted in all resources available through HAProxy being unavailable through the VIP. Now, Keepalived checks if the HAProxy process runs on a particular node.

• Fixed the issue with false positive errors in GlusterFS logs. Previously, the Telegraf GlusterFS input plugin, installed on every GlusterFS node, was querying all GlusterFS nodes at the same time, which led to locked transactions. Now, StackLight LMA monitors the GlusterFS cluster using remote_agent and sends the queries to the VIP managed by Keepalived.

• Fixed the issue with the StackLight LMA deployment failure caused by the notification service connecting to the wrong Elasticsearch IP address.

• Fixed the issue with the wrong name of the Elasticsearch Curator package in the Elasticsearch Salt formula.

**DevOps Portal**

• Fixed the issue with the Heatmaps and DriveTrain dashboards not working in the DevOps Portal.

• Fixed the issue with Rundeck failing to start because of the misconfiguration of the GlusterFS volume.

• Fixed the issue with Grafana Compute Dashboard, deployed together with the DevOps Portal, displaying incorrect statistics in the Network RX/TX panel.

• Fixed the issue with Alertmanager failing to send some alerts to Salesforce with the Resource Case Not Found error message recorded in the Push Notification service logs.

• Fixed the issue with Heka (log_collector) failing to read logs from the DevOps Portal and reporting a number of permission denied error messages in syslog.
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 MCP DriveTrain-related issues with available workarounds.

- Occasionally, due to default misconfiguration of watchdog, the salt state may fail to restart the watchdog service.

  **Workaround:**

  1. Stop the watchdog service:

     ```bash
     salt '*' cmd.run 'service watchdog stop'
     ```

  2. Reapply the watchdog.server:

     ```bash
     salt '*' state.sls watchdog.server -l debug
     ```

- Due to the default Docker Swarm CA configuration in the offline APT image, the Docker system with services, including Aptly and Docker registry, may be unable to start. The issue reproduces only if the apt01 node is powered off for more than 90 days.

  The workaround is to apply the **patch** for the apt01 bootstrap script.

  **See also**

  MCP Deployment Guide: Generate configuration drives manually

**OpenStack**

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

- Occasionally, the deployment of Designate may fail. The issue affects automatic OpenStack deployments through the Jenkins web UI as well as manual OpenStack deployments.

  **Workaround:**

  - For a DriveTrain Designate deployment, rerun the Deploy - OpenStack Jenkins deployment pipeline.
  - For a manual Designate deployment, reapply the following state:

    ```bash
    salt -C '@designate:server' state.sls designate -b 1
    ```
• The Designate quotas can be set on non-existing projects or on the same project for the project ID and name independently, but only the quotas set with a project ID are actually enforced on projects.
  Workaround:
  Create and list active Designate quotas using a project ID only.

• The deployment of an OpenStack environment using the Deploy - OpenStack Jenkins pipeline may fail during the deployment of LDAP for Keystone.
  Workaround:
  1. Deploy an OpenStack environment without LDAP enabled.
  2. Enable LDAP as described in MCP Deployment Guide: Configure LDAP with Keystone server.

• Occasionally, when Keystone rotates the fernet tokens in GlusterFS, the storage token may fail to be created. The issue affects only the existing deployments.
  Workaround:
  • Run the following command:

    ```bash
    keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone
    ```

  • To prevent failures, disable the Keystone rotation on the cluster level by adding the following snippet to cluster/<cluster_name>/openstack/control_init.yml:

    ```yaml
    linux:
      system:
        job:
          keystone_job_rotate:
            enabled: False
    ```

• In case of a DPDK deployment, the linux.network state fails due to the incorrect ovs-vswitchd priority. Generic ovs-vswitchd has higher priority than ovs-vswitchd-dpdk. The following error will occur:

    ```bash
    ovs-vswitchd: ovs|00007|dpdk|ERR|DPDK not supported in this copy of Open vSwitch.
    ```

  Workaround:
  Perform the following steps before applying the linux.network state:
  1. Install the Open vSwitch DPDK packages:

    ```bash
    salt -C 'I@nova:compute' pkg.install openvswitch-switch-dpdk,openvswitch-switch,bridge-utils,vlan
    ```

  2. Switch alternatives for Open vSwitch:

    ```bash
    salt -C 'I@nova:compute' cmd.run 'update-alternatives --set ovs-vswitchd /usr/lib/openvswitch-switch-dpdk/ovs-vswitchd-dpdk'
    ```

  3. Stop generic Open vSwitch on the DPDK compute nodes:
4. Remove old Open vSwitch database on the DPDK compute nodes:
```bash
salt -C 'l@nova:compute' cmd.run 'rm -f /etc/openvswitch/conf.db /etc/openvswitch/.conf.db.~lock~'
```

5. Start the DPDK Open vSwitch on the DPDK compute nodes:
```bash
salt -C 'l@nova:compute' service.start openvswitch-switch
```

- In case of a DPDK deployment, the `linux.network` state may fail when using the DPDK bond. The issue occurs if one of the DPDK links is not ready and available at the moment.
  
  **Workaround:**
  Reapply the `linux.network` state on the DPDK compute nodes.

**OpenContrail**

This section lists the OpenContrail-related issues with available workarounds if any.

- Setting up a host router prevents an instance to receive the default gateway.
  
  **Workaround:**
  Configure your vRouter to automatically choose what next hop must be used:
  1. Log in to the OpenContrail web UI.
  2. Go to Configure -> Networking -> Networks -> Edit network.
  3. Choose Host Route(s).
  4. Use the + - button to add and configure your host routes as required.

- When deleting multiple instances in a row, some of them cannot be deleted returning the Internal Error 500 from OpenContrail.
  
  **Workaround:**
  Repeat the deletion procedure until succeeded.

- The `contrail-vrouter-agent` service crashes under load with no specific errors. The workaround is not required since the service automatically restarts after a crash.

- OpenContrail treats the SNAT port as a regular port applying the default security group that blocks a VM ingress traffic from the external network by default.
  
  **Workaround:**
  Since the default security group does not have a rule to allow ingress traffic from anywhere, modify it for SNAT to work properly. Since the default security group is only associated with the router interface, it is safe to allow all ingress traffic from the 0.0.0.0 source IP. This way, the incoming traffic will not be dropped and the VMs on the tenant
network will have desired external connectivity (NAT). The egress ports 80 and 443 can be then allowed on a dedicated (non-default) group that should be associated as usual on VMs.

**Caution!**

The default security group must not be associated with any other ports and instances but only with the router SNAT port, which is the default option in OpenContrail.

- The OpenContrail DPDK-based vRouter with 1 GB hugepages enabled may fail to start with the VROUTER: Error initializing flow table: Cannot allocate memory. error due to the default mount point /dev/hugepages created by systemd.

**Workaround:**

1. Log in to any OpenStack compute node.
2. Unmount the default mount point:
   
   ```bash
   umount /dev/hugepages
   ```
3. Restart the supervisor-vrouter service:
   
   ```bash
   service supervisor-vrouter restart
   ```
4. (Optional) To force these actions after any reboot of the OpenStack compute node, add umount /dev/hugepages; service supervisor-vrouter restart to the /etc/rc.local file before the exit 0 record. For example:

   ```bash
   #!/bin/sh -e
   # rc.local
   # This script is executed at the end of each multiuser runlevel.
   # Make sure that the script will "" on success or any other
   # value on error.
   # In order to enable or disable this script just change the execution
   # bits.
   # By default this script does nothing.
   umount /dev/hugepages; service supervisor-vrouter restart
   exit 0
   ```
Maintenance updates

Mirantis is releasing the maintenance updates for the Q1’18 MCP release.

Q1’18 maintenance updates summary

<table>
<thead>
<tr>
<th>Maintenance update</th>
<th>Release date</th>
<th>Summary</th>
</tr>
</thead>
</table>
| 2018.4.1           | April 17, 2019   | • Bug fixes for the Nova service of OpenStack Pike and/or Ocata
|                    |                  | • Bug fix for Ceph                           |

2018.4.1

The MCP 2018.4.1 update contains bug fixes for the OpenStack Nova and Ceph MCP components. The update is available starting from April 17, 2019.

To apply the MCP 2018.4.1 update to the MCP 2018.4.0 deployment, follow the guidelines available in the How to apply the update section.

Addressed issues

MCP 2018.4.1 update contains fixes for the OpenStack Nova and Ceph MCP components.

Nova

• [1-28] [Ocata] [Pike] Fixed the community issue that caused a compute node trying to delete the instance after a failed evacuation.
• [19639] [Ocata] Fixed the issue with OpenStack VMs creating two ports on the same network.
• [28148] [Pike] Fixed the issue with erroneous compute resource allocation in the placement service causing lack of free resources in an OpenStack environment for primary operations, such as spawning, resizing, or migrating an instance.

Ceph

[28408] Added the capability to modify the `mon_max_pg_per_osd` parameter in `/etc/ceph/ceph.conf` to avoid the Ceph cluster outage issues.

Updated packages

MCP 2019.4.1 update includes the following changes in the minor versions of the packages. All other versions of the major MCP components remain the same as in the MCP Q1’18 GA release and can be found in Major components versions. All 2018.4.1 packages are available at [http://mirror.mirantis.com/update/2018.4.0/](http://mirror.mirantis.com/update/2018.4.0/).

Updated major software components from the Mirantis repositories
<table>
<thead>
<tr>
<th>Component</th>
<th>Application/service</th>
<th>2018.4.0</th>
<th>2018.4.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenStack Pike</td>
<td>Barbican</td>
<td>1:5.0.0.1-1~u16.04+m cp11</td>
<td>1:5.0.0.2-1~u16.04+m cp15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:11.1.0-1~u16.04+m cp59</td>
<td>2:11.1.1-1~u16.04+m cp113</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:5.0.1-2~u16.04+m cp1</td>
<td>1:5.0.2-2~u16.04+m cp15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:15.0.1-1~u16.04+m cp3</td>
<td>2:15.0.1-1~u16.04+m cp15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:9.0.3-1~u16.04+m cp23</td>
<td>1:9.0.5-1~u16.04+m cp40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3:12.0.2-1~u16.04+m cp79</td>
<td>3:12.0.3-3~u16.04+m cp54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:9.1.3-1~u16.04+m cp23</td>
<td>1:9.1.5-1~u16.04+m cp27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:12.0.0-2~u16.04+m cp22</td>
<td>2:12.0.1-4~u16.04+m cp13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:5.0.1-1~u16.04+m cp50</td>
<td>1:5.0.1-1~u16.04+m cp76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:16.1.0-1~u16.04+m cp134</td>
<td>2:16.1.5-3~u16.04+m cp127</td>
</tr>
<tr>
<td>OpenStack Networking Pike</td>
<td>BGP VPN</td>
<td>7.0.0-2~u16.04+mcp1</td>
<td>7.0.0-2~u16.04+mcp16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:11.0.0-1~u16.04+m cp4</td>
<td>1:11.0.0-1~u16.04+m cp6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:11.0.3-1~u16.04+m cp42</td>
<td>2:11.0.5-2~u16.04+m cp179</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:11.0.3-1~u16.04+m cp42</td>
<td>2:11.0.5-2~u16.04+m cp179</td>
</tr>
<tr>
<td>OpenStack Telemetry Pike</td>
<td>Aodh</td>
<td>5.1.0-2~u16.04+mcp6</td>
<td>5.1.0-3~u16.04+mcp10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:9.0.5-2~u16.04+m cp10</td>
<td>1:9.0.6-2~u16.04+m cp15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.1.0-1~u16.04+mcp9</td>
<td>3.1.0-1~u16.04+mcp14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.0.4-2~u16.04+mcp1</td>
<td>4.0.5-2~u16.04+mcp2</td>
</tr>
<tr>
<td>OpenStack Ocata</td>
<td>Barbican</td>
<td>1:4.0.0-1~u16.04+m cp11</td>
<td>1:4.0.0-2~u16.04+m cp13</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td>Cinder</td>
<td>2:10.0.6-1~u16.04+mcp12</td>
<td>2:10.0.8-1~u16.04+mcp5</td>
<td></td>
</tr>
<tr>
<td>Designate</td>
<td>1:4.0.0-2~u16.04+mcp17</td>
<td>1:4.0.0-2~u16.04+mcp19</td>
<td></td>
</tr>
<tr>
<td>Heat</td>
<td>1:8.0.6-1~u16.04+mcp7</td>
<td>1:8.0.7-1~u16.04+mcp9</td>
<td></td>
</tr>
<tr>
<td>Horizon</td>
<td>3:11.0.4-1~u16.04+mcp56</td>
<td>3:11.0.4-2~u16.04+mcp60</td>
<td></td>
</tr>
<tr>
<td>Ironic</td>
<td>1:7.0.4-1~u16.04+mos3</td>
<td>1:7.0.7-1~u16.04+mos4</td>
<td></td>
</tr>
<tr>
<td>Keystone</td>
<td>2:11.0.3-1~u16.04+mcp6</td>
<td>2:11.0.4-1~u16.04+mcp5</td>
<td></td>
</tr>
<tr>
<td>Manila</td>
<td>1.14.0-1~u16.04+mcp2</td>
<td>1:4.0.1-1.0~u16.04+mcp9</td>
<td></td>
</tr>
<tr>
<td>Nova</td>
<td>2:15.1.0-1~u16.04+mcp35</td>
<td>2:15.1.5-1~u16.04+mcp20</td>
<td></td>
</tr>
<tr>
<td>OpenStack Networking Ocata</td>
<td>Networking ODL ML2 plugin</td>
<td>2:10.0.4-1~u16.04+mcp13</td>
<td>2:10.0.7-1~u16.04+mcp31</td>
</tr>
<tr>
<td>Neutron</td>
<td>2:10.0.4-1~u16.04+mcp13</td>
<td>2:10.0.7-1~u16.04+mcp31</td>
<td></td>
</tr>
</tbody>
</table>

How to apply the update

Enable the update repository

This section describes how to prepare your MCP cluster repository for the MCP 2018.4.1 maintenance update on the non-offline OpenStack Ocata- and Pike-based MCP clusters with the Build ID 2018.4.0. The procedure covers switching of all MCP-related repositories from apt.mirantis.com to mirror.mirantis.com.

Caution!

The following procedure mainly implies a non-customized cluster. It covers only a few basic customization cases and must be adjusted by a support engineer on a per-cloud basis.

To enable the update repository:

1. Back up the following MCP cluster items:
   - The cluster and system level of your Reclass model
• The salt-formula-linux Salt formula
• The /etc/apt/sources.list and /etc/apt/sources.list.d/* files from every node of the cluster

2. Update the 2018.4.0 branch of your Reclass model on the system level:

   Note
   Since this procedure implies a non-customized cluster, pulling of the MCP 2018.4.1 maintenance update changes should not cause any conflicts.

1. Log in to the Salt Master node.

2. Open your Git project repository with the Reclass model on the system level:

   cd /srv/salt/reclass/classes/system

3. Identify the ID of the last commit in your system:

   git log --name-status HEAD^..HEAD

4. Verify that your last commit ID matches the one in the system response of the following command:

   git show 2018.4.0

   Warning
   If the IDs do not match, do not proceed with the next steps. Contact Mirantis support for details.

5. Pull the latest changes from the 2018.4.0 branch:

   git pull origin release/2018.4.0

6. If you have merge conflicts, fix them using the following exemplary steps:

   Caution!
   This step describes an example action plan for a customized cluster. The final procedure must be prepared by a support engineer for a particular cloud.
1. Move the corresponding conflicting changes to the cluster level of the Reclass model.

Warning
Mirantis does not recommend modifying the system and service levels of your Reclass model.

2. Use the following set of commands to pull the latest changes:

   cd /srv/salt/reclass/classes/system
   git stash
   git pull origin release/2018.4.0
   git stash pop

3. Switch to the mirror.mirantis.com repository from apt.mirantis.com:

   1. Log in to the Salt Master node.
   2. Open your Git project repository with the Reclass model on the cluster level:

   ```bash
   cd /srv/salt/reclass/classes/cluster/<CLUSTER_NAME>/
   ```

3. In each configuration YAML file of each MCP cluster component, rename the following classes where present:

<table>
<thead>
<tr>
<th>Old class name</th>
<th>New class name</th>
</tr>
</thead>
<tbody>
<tr>
<td>system.linux.system.repo.mcp.apt_mirantis.saltstack_2016_3</td>
<td>system.linux.system.repo.mcp.apt_mirantis.saltstack</td>
</tr>
<tr>
<td>system.linux.system.repo.mcp.extra</td>
<td>system.linux.system.repo.mcp.apt_mirantis.extra</td>
</tr>
<tr>
<td>system.linux.system.repo.mcp.salt</td>
<td>system.linux.system.repo.mcp.apt_mirantis.salt-formulas</td>
</tr>
<tr>
<td></td>
<td>For the Salt Master node configuration file located in infra/config/init.yml by default</td>
</tr>
</tbody>
</table>

Use the following sed command as an example to rename the classes:

```bash
egrep -lRZ "system\.linux\.system\.repo\.mcp\.apt_mirantis\.saltstack_2016_3" . | xargs -0 -l sed -i "s/system\.linux\.system\.repo\.mcp\.apt_mirantis\.saltstack_2016_3/system\.linux\.system\.repo\.mcp\.apt_mirantis\.saltstack/g"
```

4. In infra/init.yml:
   - Add system.defaults as the first class in the classes list
• Add the mcp_version: ${_param:apt_mk_version} parameter

```
classes:
- system.defaults
...

parameters:
  _param:
    apt_mk_version: 2018.4.0
    mcp_version: ${_param:apt_mk_version}
```

5. For the OpenContrail-based MCP clusters:

1. Rename the `system.linux.system.repo.mcp.contrail` class to `system.linux.system.repo.mcp.apt_mirantis.contrail` using the following `sed` command as an example:

   ```bash
   egrep -lRZ "system\.linux\.system\.repo\.mcp\.contrail" \.
   \sargs -0 -i sed -i \n   's/system\.linux\.system\.repo\.mcp\.contrail/system.linux.system.repo.mcp.apt_mirantis.contrail/g'
   ```

2. Add the `system.linux.system.repo.mcp.apt_mirantis.contrail_ocata` class to the following files:
   - `opencontrail/compute.yml`
   - `openstack/proxy.yml`
   - `openstack/control.yml`

   Use the following `sed` command as an example:

   ```bash
   sed -i 's/classes:/classes:
   - system.linux.system.repo.mcp.apt_mirantis.contrail_ocata/g' \n   /srv/salt/reclass/classes/cluster/<CLUSTER_NAME>/opencontrail/compute.yml
   ```

3. In `opencontrail/compute.yml`, change the `linux_system_codename_contrail` parameter value to use Xenial for the compute nodes instead of Trusty. If this parameter is absent, add it explicitly.

   ```bash
   linux_system_codename_contrail: ${_param:linux_system_codename}
   ```

   Use the following `sed` command as an example:

   ```bash
   sed -i 's/  _param:/  _param:
   _param:linux_system_codename_contrail: ${_param:linux_system_codename}/g' \n   /srv/salt/reclass/classes/cluster/<CLUSTER_NAME>/opencontrail/compute.yml
   ```

4. In `opencontrail/init.yml`, add `linux_repo_contrail_version: 3.2`:

   ```bash
   sed -i 's/  _param:/  _param:
   _param:linux_repo_contrail_version: 3.2/g' \n   /srv/salt/reclass/classes/cluster/<CLUSTER_NAME>/opencontrail/init.yml
   ```
6. If you have dedicated MAAS nodes, in `infra/maas.yml`, change the location of the region parameter as follows:

```yaml
maas:
  ...
  cluster:
    {# MAINTENANCE: allow to disable maas #}
    enabled: {{ cookiecutter.maas_enabled }}
    region:
      ...
```

4. Add the **update repository** classes:

1. Log in to the Salt Master node.
2. Open your Git project repository with the Reclass model on the cluster level:

   ```bash
cd /srv/salt/reclass/classes/cluster/<CLUSTER_NAME>/
```
3. Add the `system.linux.system.repo.mcp.apt_mirantis.update.<component_name>` class under each MCP component:

   - `system.linux.system.repo.mcp.apt_mirantis.ubuntu`
   - `system.linux.system.repo.mcp.apt_mirantis.update.ubuntu`
   - `system.linux.system.repo.mcp.apt_mirantis.extra`
   - `system.linux.system.repo.mcp.apt_mirantis.update.extra`

   **Note**
   Ordering is not mandatory but helps to track and verify changes.

Use the following `sed` command as an example:

```
egrep -lRZ "system\llinux\lsystem\rrepomcp\lapt\lapt_mirantis\lupdate\l\(ceph\|contrail\|contrail\l\rcat\|docker\|elastic\|extra\l\rre\l\rstack\l\lstack\l\rformulas\l\rstack\l\lrid\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lupdate\l\lup
6. Refresh pillars:

```
salt '*' saltutil.clear_cache && salt '*' saltutil.refresh_pillar
```

7. Verify the Reclass model data compilation for each node:

```
salt '*' saltutil.sync_all
reclass -i
```

8. Update salt-formula-linux to enable a proper setup of the mirror.mirantis.com mirrors and APT preferences:

1. Verify that the salt-formula-linux files version matches the version for the MCP Build ID 2018.4.0 that is 2017.4.1+201804181255.bb0708d~xenial1_all:

```
dpkg -l | grep salt-formula-linux
```

2. Inspect salt-formula-linux for customizations:

```
debsums -s salt-formula-linux
```

The system response must be empty.

- If system response of the command is empty, proceed to the next step.
- If any of the files does not match:
  1. Manually apply the salt-formula-linux patch according to the customizations of your MCP cluster.
  2. Proceed to the next step.

3. From the Salt Master node, update salt-formula-linux to the latest version available at mirror.mirantis.com. For example:

```
```

```
dpkg -i salt-formula-linux.deb
```

9. Verify that you have successfully backed up all /etc/apt/ files from each node of the cluster.

10. Perform the force cleanup of the /etc/apt/ directory and apply the linux.system.repo state:

```
salt '*' cmd.run 'rm -v /etc/apt/preferences.d/*'
salt '*' cmd.run 'rm -v /etc/apt/preferences || true'
salt '*' cmd.run 'rm -v /etc/apt/sources.list /etc/apt/sources.list.d/* || true'
salt '*' saltutil.sync_all
salt '*' state.sls linux.system.repo
```
If any of these states fail:

1. Revert the changes made in the steps 2-4 of this procedure.
2. Reapply the states specified in the steps 5-6 of this procedure.
3. Investigate the root cause of the issue. Usually, failures are caused by a custom or manually added repositories configuration. Contact Mirantis support for details.

11 Update all remaining Salt formulas:

   1. Verify that the Salt formulas do not contain any customizations:

       ```bash
       apt-get install debsums -y
       for i in $(dpkg-query -W -f='${Package}
       | sed "s/ //g" |grep 'salt-formula-');
       do debsums -s ${i}; done
       ```

       The system response must be empty.

       **Warning**

       Do not proceed if the system response is not empty. Otherwise, your MCP cluster customizations will be lost. Contact Mirantis support for details.

   2. Update the Salt formulas:

       ```bash
       salt -C I@salt:master state.apply salt.master
       salt '*' saltutil.sync_all
       ```

12 In Drivetrain jenkins trigger git-mirror-downstream-mk-pipelines and git-mirror-downstream-pipeline-library jobs with BRANCHES parameter set to release/2018.4.0.

13 Update jenkins configuration and add the new Deploy - Maintenance Update pipeline job to Jenkins:

   ```bash
   salt 'cid01*' state.sls jenkins.client.job
   ```

14 Commit new changes to your Reclass cluster model. For example:

   ```bash
   cd /srv/salt/reclass/
   git add -A
   git commit -m "Repositories update for 2018.4.1"
   ```
15 Proceed to Prepare the MCP cluster for the update.

Prepare the MCP cluster for the update

This section includes the obligatory preparation procedure that you should perform before you proceed with the MCP update.

To prepare the MCP cluster for the maintenance update:

1. Verify that you have completed the steps described in Enable the update repository.
2. Verify that Nova is up and running:
   1. From the Salt Master node, verify that the following state is applied without errors:

```
ssh ctl01 "source keystonecv3; nova service-list"
```

   2. From any OpenStack controller node, create a test instance:

```
openstack server create --flavor <FLAVOR_ID> --image <IMAGE_ID> --nic <NIC_ID> --net-id <NETWORK_ID> <TEST_INSTANCE_NAME>
```

3. If your MCP cluster runs Ceph, verify that your Ceph cluster is healthy and the Ceph formula states can be applied without errors:
   1. From the Salt Master node, apply:

```
salt "$\{CEPH\_MON\}" cmd.run 'ceph -s'
```

   Example of system response:

```cmn01.domain.com:
   cluster e0b75d1b-544c-4e5d-98ac-cfbaf29387ca
   health HEALTH_OK
   monmap e3: 3 mons at {cmn01=192.168.16.14:6789/0,cmn02=192.168.16.15:6789/0,\cmn03=192.168.16.16:6789/0}
   election epoch 42, quorum 0,1,2 cmn01,cmn02,cmn03
   osdmap e102: 6 osds: 6 up, 6 in
   flags sortbitwise,require_jewel_osds
   pgmap v41138: 384 pgs, 6 pools, 45056 kB data, 19 objects
   798 MB used, 60575 MB / 61373 MB avail
   384 active+clean```

   2. From the Salt Master node, apply:

```
salt -C '@ceph:mon' state.sls ceph.mon test=true
```
3. From one of the Ceph OSD nodes, apply:

```bash
salt-call state.sls ceph.osd test=true
```

4. From the Salt Master node, apply:

```bash
salt -C 'I@ceph:osd' state.sls ceph.osd test=true
```

**Warning**

If the system response of any state applied in the steps 3.2 - 3.4 contains comments about configuration changes to be applied, do not proceed with the next steps.

**Example of system response:**

```
----------
ID: common_config
Function: file.managed
   Name: /etc/ceph/ceph.conf
   Result: None
   Comment: The file /etc/ceph/ceph.conf is set to be changed
   Started: 09:22:42.187794
   Duration: 101.473 ms
   Changes:
       diff:
       ...
       +++
       @@ -1,7 +1,7 @@
       [global]
       mon initial members = cmn01,cmn02,cmn03
       mon host = 10.11.0.66:6789,10.11.0.67:6789,10.11.0.68:6789
       -cluster network = 10.11.0.0/16
       +cluster network = 10.12.0.0/16
```

Contact Mirantis support for further details.

4. Optional. Perform an online database synchronization since this task can take a significant amount of time based on the cloud size.

5. Verify that all nodes of your MCP cluster are available and all OpenStack-related services are up and running.

6. Schedule a maintenance window for each update phase depending on your cluster size.
Now, you can proceed with either Apply the MCP 2018.4.1 maintenance update through Jenkins or Apply the MCP 2018.4.1 maintenance update manually. The automated update procedure is recommended for the following configurations:

- MCP 2018.4.0 OpenStack Ocata with OpenContrail 3.2 and Ceph Luminous
- MCP 2018.4.0 OpenStack Pike with OVS and Ceph Luminous

For the customized MCP deployments, we recommend using the manual update procedure.

**Apply the MCP 2018.4.1 maintenance update through Jenkins**

This section describes how to apply the MCP 2018.4.1 maintenance update using the Deploy - Maintenance Update Jenkins pipeline job that updates the Nova and Ceph packages.

The recommended MCP configurations to perform the automated update to MCP 2018.4.1 include:

- MCP 2018.4.0 OpenStack Ocata with OpenContrail 3.2 and Ceph Luminous
- MCP 2018.4.0 OpenStack Pike with OVS and Ceph Luminous

Otherwise, we recommend that you perform the update manually as described in Apply the MCP 2018.4.1 maintenance update manually.

To apply the MCP 2018.4.1 maintenance update:

1. Verify that you have completed the steps described in Prepare the MCP cluster for the update.
2. Log in to the Salt Master node.
3. In cluster/<CLUSTER_NAME>/cicd/control/leader.yml of your Git project repository with the Reclass model, verify that the following class is present:

   ```yaml
   classes:
   - system.jenkins.client.job.deploy.update
   ```

4. Update pillar on the Jenkins nodes:

   ```bash
   salt -C 'I@jenkins:client' saltutil.pillar_refresh
   ```

5. Apply the following state:

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

6. Choose from the following options:

   - If your cluster runs Ceph, run the Deploy - Maintenance Update Jenkins pipeline job with the COMPONENTS = nova,ceph parameter specified and the TARGET_SERVERS parameter unset.
• If your cluster does not run Ceph, run the Deploy - Maintenance Update Jenkins pipeline job with the COMPONENTS = nova parameter specified and the TARGET_SERVERS parameter unset.

Note
If TARGET_SERVERS is set to any custom value, the pipeline job executes the steps above on all hosts that match the target value.

Setting of TARGET_SERVERS to a custom value is useful, for example, for a parallel execution of the pipeline job on all OpenStack compute nodes with TARGET_SERVERS = ‘l@nova:compute:enabled:true’. You can also specify several targets, for example, TARGET_SERVERS = ctl01.domain.tst,ctl02.domain.tst,ctl03.domain.tst.

However, Mirantis recommends sticking to the default value with TARGET_SERVERS unset. Also, Mirantis recommends updating the OpenStack controller nodes before updating the OpenStack compute nodes.

The pipeline workflow if COMPONENTS = nova and TARGET_SERVERS is unset:

1. The following actions are performed on each OpenStack controller node one by one:

   1. Stop apache2 and all Nova services.

   Note
   Since the OpenStack controller node runs other OpenStack components services along with nova-api, all API services will be unavailable during the node update. The apache2 service will start during the execution of the step 3.

   2. Stop all Nova services running on a node.

   3. Run the apt install nova* python-nova* --only-upgrade command to update the existing Nova packages.

   4. Apply the Nova Salt state.

   5. Verify that all Nova services from the output of the nova service-list command located on the node being updated are in the up state.

   Note
   If any of the steps is unsuccessful, the pipeline job raises a corresponding error and stops.
2. The steps 1.2 - 1.5 are performed on each OpenStack compute node one by one. Only the nova-compute service is stopped during this stage.

The pipeline workflow if COMPONENTS = ceph and TARGET_SERVERS is unset:
1. Detect the Ceph nodes: mon, rgw, osd, and Ceph clients.
2. Apply the ceph.common state to update the mon_max_pg_per_osd variable.
3. Restart the Ceph services on the first Ceph node.
4. Wait for the Ceph cluster to become healthy on the first Ceph node.
5. Repeat the steps 3-4 on the remaining Ceph nodes one by one.

The pipeline workflow if COMPONENTS = ceph,nova and TARGET_SERVERS is unset:
1. The Ceph components are updated following the stages described above for Ceph.
2. The Nova components are updated following the stages described above for Nova.

Apply the MCP 2018.4.1 maintenance update manually
This section describes how to apply the MCP 2018.4.1 maintenance update manually. The procedure will update the Nova and Ceph components. You can update only the Nova component if Ceph is not installed on your cluster.

The steps included in the procedure below are aligned with the Deploy - Maintenance Update Jenkins pipeline workflow. We provide the manual update procedure for the use cases where the manual update is preferred over the automated one. More specifically, these use cases include:

• Customized cluster model configuration
• Customized Nova service configuration
• Customized code in openstack-nova and ceph packages

To apply the MCP 2018.4.1 maintenance update manually:

1. Verify that you have completed the steps described in Prepare the MCP cluster for the update.
2. Update the Nova component:
   1. On each OpenStack controller node one by one, perform the following steps:
      1. Stop apache2:

Note
Since the OpenStack controller node runs other OpenStack components services along with nova-api, all API services will be unavailable during the node update. The apache2 service will start during the execution of the step 4.
2. Stop all Nova services:

```
systemctl stop nova*
```

3. Update the existing Nova packages:

```
apt install nova* python-nova* --only-upgrade
```

4. Apply the Nova Salt state.

5. Verify the system output of the nova service-list command. All Nova services located on the node being updated should be in the up state. For example:

```
nova service-list
```

```
+---------------------------+------------------+-------+----------+---------+-------+-------------+-----------------+-------------+
| Id                        | Binary           | Host  | Zone     | Status  | State | Updated_at  | Disabled Reason | Forced down |
+---------------------------+------------------+-------+----------+---------+-------+-------------+-----------------+-------------+
| 49db9d9d-...-fa9445323d49 | nova-scheduler   | ctl01 | internal | enabled | up    | 2019-04-... | -               | False       |
| 7b5a0192-...-d2bc4ba6d5d | nova-consoleauth | ctl01 | internal | enabled | up    | 2019-04-... | -               | False       |
| 98c944ec-...-7c3122d02ac2 | nova-conductor   | ctl01 | internal | enabled | up    | 2019-04-... | -               | False       |
| ae0ff51a-...-1dcdaf523b9b | nova-compute     | cmp1  | nova     | enabled | up    | 2019-04-... | -               | False       |
+---------------------------+------------------+-------+----------+---------+-------+-------------+-----------------+-------------+
```

6. Verify that apache2 is running:

```
systemctl status apache2
```

2. On each OpenStack compute node, perform the following steps:

1. Stop all Nova services:

```
systemctl stop nova*
```

2. Update the existing Nova packages:

```
apt install nova* python-nova* --only-upgrade
```

3. Apply the Nova Salt state.

4. Verify the system output of the nova service-list command. All Nova services located on the node being updated should be in the up state. For example:

```
nova service-list
```

```
+---------------------------+------------------+-------+----------+---------+-------+-------------+-----------------+-------------+
| Id                        | Binary           | Host  | Zone     | Status  | State | Updated_at  | Disabled Reason | Forced down |
+---------------------------+------------------+-------+----------+---------+-------+-------------+-----------------+-------------+
| 49db9d9d-...-fa9445323d49 | nova-scheduler   | ctl01 | internal | enabled | up    | 2019-04-... | -               | False       |
| 7b5a0192-...-d2bc4ba6d5d | nova-consoleauth | ctl01 | internal | enabled | up    | 2019-04-... | -               | False       |
| 98c944ec-...-7c3122d02ac2 | nova-conductor   | ctl01 | internal | enabled | up    | 2019-04-... | -               | False       |
| ae0ff51a-...-1dcdaf523b9b | nova-compute     | cmp1  | nova     | enabled | up    | 2019-04-... | -               | False       |
+---------------------------+------------------+-------+----------+---------+-------+-------------+-----------------+-------------+
```

3. Update the Ceph component:
1. Verify that the Ceph nodes, that are mon, rgw, osd, and Ceph clients are responding:

```
salt -C "I@ceph:common" test.ping
```

2. On the first Ceph monitor node:
   1. Apply the ceph.common state to update the mon_max_pg_per_osd variable:

```
salt -C "I@ceph:mon and I@ceph:common:keyring:admin" state.sls ceph.common
```
   2. Restart the Ceph services:

```
salt -C "I@ceph:mon and I@ceph:common:keyring:admin" cmd.run "systemctl restart ceph-mon.target"
```
   3. Wait for the Ceph cluster to become healthy on the first Ceph monitor node:

```
salt -C "I@ceph:mon and I@ceph:common:keyring:admin" cmd.run "ceph -s"
```

3. Obtain the list of all Ceph Monitor nodes:

```
salt -C "I@ceph:mon" test.ping
```

4. On each of the remaining Ceph Monitor nodes (one by one):
   1. Apply the ceph.common state to update the mon_max_pg_per_osd variable:

```
salt `<HOST_NAME>` state.sls ceph.common
```
   2. Restart the Ceph services:

```
salt `<HOST_NAME>` cmd.run "systemctl restart ceph-mon.target"
```
   3. Wait for the Ceph cluster to become healthy on the Ceph Monitor node:

```
salt `<HOST_NAME>` cmd.run "ceph -s"
```

5. Obtain the list of the Ceph RadosGW nodes:

```
salt -C "I@ceph:radosgw" test.ping
```

6. On each of the Ceph RadosGW nodes (one by one):
   1. Apply the ceph.common state to update the mon_max_pg_per_osd variable:

```
salt `<HOST_NAME>` state.sls ceph.common
```
   2. Restart the Ceph services:

```
salt `<HOST_NAME>` cmd.run "systemctl restart ceph-radosgw.target"
```
   3. Wait for the Ceph cluster to become healthy on the Ceph RadosGW node:
7. Obtain the list of the Ceph OSD nodes and set the noout flag:

```
salt -C "I@ceph:osd" test.ping
salt -C "I@ceph:mon and I@ceph:common:keyring:admin" cmd.run 'ceph osd set noout'
```

8. On each Ceph OSD node (one by one):

   1. Restart the Ceph OSD node:

```
salt `<HOST_NAME>` cmd.run "systemctl restart ceph-osd.target"
```

   2. Monitor the cluster health after each restart:

```
ceph osd status 2>&1 | grep <HOST_NAME>
```

9. Unset the noout flag:

```
salt -C "I@ceph:mon and I@ceph:common:keyring:admin" cmd.run 'ceph osd unset noout'
```

10. Apply the updated configuration to all Ceph clients:

```
salt -C "I@ceph:common" state.sls ceph.common
```