MCP Feature Update 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.
Preface

The MCP Feature Update Notes are intended to continually provide quick updates on new functionality and improvements of the product.

Intended audience

These update notes are intended for developers, deployment and QA engineers, system administrators, and operators. They assume that the reader is already familiar with network and cloud concepts.

Content specification

This documentation is structured to reflect the development deliverables over time. Each calendar quarter has a dedicated section and includes sprint-based sub-sections. Each sprint-related page contains feature updates delivered during the sprint and filters the notes by areas where deployment takes place.

A feature update note can be included in one of the following lists:

- **New features**
  Record the summary of newly implemented features and how to apply the update to an existing deployment if applicable

- **Upgrade notes**
  Record the summary of improvements to existing functionality

- **Deprecation notes**
  Record deprecated features

- **Bug fixes**
  Record the summary of bug fixes for blocker, critical, and/or customer-found issues

- **Other notes**
  Record any additional notes that are not related to any of the above lists

Documentation history

These update notes are released biweekly, on a sprint basis. The reader can view the sprint schedule and availability dates of the update notes by visiting the Sprint schedule page that belongs to the calendar quarter in question.
Q3`17
This section includes the sprint updates for the third calendar quarter of the year 2017.

Q3`17 sprint schedule
The table below contains the sprint schedule for the third calendar quarter of the year 2017.

<table>
<thead>
<tr>
<th>Sprint #</th>
<th>Start</th>
<th>End</th>
<th>Notes availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Sep7</td>
<td>Sep20</td>
<td>Sprint 13</td>
</tr>
<tr>
<td>14</td>
<td>Sep21</td>
<td>Oct4</td>
<td>Sprint 14</td>
</tr>
</tbody>
</table>

Sprint 14
Reclass model

Summary

- Added system metadata to set upstream proxy for the MAAS, Gerrit, Jenkins, and Aptly services
- Added possibility to define NGINX proxy timeouts for the OSS service
- Added support for the Push Notifications service to Salesforce
- Improved the OpenStack cloud credentials defining process for the OSS services
- Moved Prometheus-based Stacklight and OSS services to monitoring nodes

New features

- Added system metadata to set upstream proxy for the MAAS, Gerrit, Jenkins, and Aptly services.

To apply the change to an existing cicd cluster deployment:

1. Set the http_proxy parameter in cluster.<CLUSTER_NAME>.cicd.control.init.yml:

   ```parameters:
   _param:
   http_proxy: 'http://<<upstream_proxy_address>>:<<upstream_proxy_port>>'
   ```

2. Apply the docker.client state:

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

To apply the change to an existing maas deployment:

1. Configure upstream_proxy for MAAS:
2. On the Salt Master node, update the MaaS Salt formula to latest:

```
salt-call state.sls salt.master
```

3. On the MAAS node, apply the maas.region state:

```
salt-call state.sls maas.region
reboot
```

- Added possibility to define NGINX proxy timeouts for DevOps Portal through Salt metadata. This helps you to avoid Gateway Timeout errors that occur due to huge amount of data, connection, and other issues.

To configure NGINX proxy timeouts:

1. Define following parameters on a system level for each service in your Docker stack.

```yaml
parameters:
    devops_portal:
        config:
            service:
                your_service:
                    configure_proxy: true
                    resolve_hostname: true

# Proxy parameters, default value equals 300 ms
proxy_connect_timeout: 300
proxy_send_timeout: 300
proxy_read_timeout: 300
send_timeout: 300
```

Note

The following is the example of the above configuration location:

```
/srv/salt/reclass/classes/system/docker/swarm/stack/YOUR_SERVICE.yml
```

2. Recreate devops-portal stack:
docker stack rm devops-portal
salt -C 'I@devops_portal:config' state.sls devops_portal.config
salt -C 'I@docker:swarm:role:master' state.sls docker.client

The NGINX configuration is updated during the Devops Portal Salt formula application.

• Added support for the Push Notification service to Salesforce. To configure the Push Notification service for a Salesforce account, define the following variables on the cluster level metadata:

  sfdc_auth_url: https://login.salesforce.com/services/oauth2/token
  sfdc_username: user@example.net
  sfdc_password: secret
  sfdc_consumer_key: example_consumer_key
  sfdc_consumer_secret: example_consumer_secret
  sfdc_organization_id: example_organization_id
  sfdc_sandbox_enabled: True/False

Upgrade notes

• Improved the OpenStack cloud credentials defining process for the OSS services. The changes affect cluster and system levels of an MCP deployment and include the following:

  • cacert_path and cafile parameters for the Security Audit and Cleanup services have changed to:

    security_monkey_openstack:
    source_credentials: source/path/for/os/credentials/on/env
    service_credentials: path/to/os/credentials/in/service/container

    janitor_monkey_openstack:
    source_credentials: source/path/for/os/credentials/on/env
    service_credentials: path/to/os/credentials/in/service/container

    Now, you do not need to specify the certificate file but only the directory where it is located. The certificate file is named cert.pem for all services by default; and the source_credentials value equals to the value of the oss_openstack_credentials_path parameter.

• oss_openstack_cert_path has changed as follows:

    oss_openstack_credentials_path: source/path/for/os/credentials/on/env

Bug fixes
For stability and diversification of resources, Prometheus-based Stacklight and OSS services were moved to monitoring nodes. Now, on spawning new monitoring nodes, different Docker swarm is set up on them to deliver Stacklight and/or OSS services.

To deploy Stacklight with OSS services on monitoring nodes, generate a new deployment model using the Model Designer UI.

Cluster model templates

Summary

- Enabled Virtlet for Kubernetes
- Fixed the issue with the system.influxdb.database.ceilometer class missing

New features

- Enabled Virtlet for Kubernetes for an easier configuration and deployment of Virtlet.

To enable Virtlet during an initial deployment of an MCP Kubernetes cluster:

1. When creating a Kubernetes deployment metadata model using the ModelDesigner UI, enable the Virtlet enabled check box in the Kubernetes Product parameters section.
2. After you generate the model and publish it to your project repository, modify the hosts parameter in classes/cluster/<cluster-name/kubernetes/control.yml to define the number of compute nodes on which you want to enable Virtlet. For example:

   ```yaml
   virtlet:
     enabled: true
     namespace: kube-system
     hosts:
     - cmp01
     - cmp02
     image: mirantis/virtlet:latest
   kube_network_manager: enabled: true
   ```

Bug fixes

- Added the missing system.influxdb.database.ceilometer class that allows creating an InfluxDB database for Tenant Telemetry in OpenStack environments.

StackLight

Summary

- Added a Grafana dashboard for the Prometheus Alertmanager service
- Added a Grafana dashboard for the Prometheus Pushgateway service
- Implemented new alerts
• Improved the Nova dashboard in Grafana

New features

• Added a Grafana dashboard for the Prometheus Alertmanager service. The dashboard provides Alertmanager performance metrics, such as the overall health status, the number of alerts, the rate of successful and failed notifications, and so on.

• Added a Grafana dashboard for the Prometheus Pushgateway service. The dashboard provides Pushgateway performance metrics, such as the overall health status and the CPU and memory usage.

• Implemented new alerts that include:
  • Alerts that fire when the Salt Master node’s or Salt Minion node’s services are down on particular nodes.
  • An alert that fires when the local API service is down.
  • An alert that fires when the rate of out-of-memory errors is too high on a node.

• Improved the Nova dashboard in Grafana. Now, the dashboard displays the average Nova instance creation time.

DevOps Portal

Summary

• Implemented new DriveTrain page in the DevOps Portal UI with custom Jenkins interface

New features

• Implemented new DriveTrain page in the DevOps portal UI with custom Jenkins interface. Now, the DevOps Portal user can perform the following operations through the DevOps Portal UI:
  • View the list of Jenkins jobs by views with job names and last build statuses and descriptions
  • View specific Jenkins job information including list of builds with their statuses and descriptions as well as the stages for the last five builds
  • Analyze a specific build information including stages, console output, and artifact list on a build information page
  • View a job console output in real time and manage the build flow
  • Execute Jenkins jobs with custom parameters and re-run builds

Kubernetes

Summary

• Added ExternalDNS for Kubernetes
• Added support for Calico to Virtlet

New features

• Added ExternalDNS for Kubernetes. ExternalDNS enables you to set 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. For details, see ExternalDNS for Kubernetes

• Added support for Calico to Virtlet that now allows running virtual machines on both types of MCP Kubernetes deployments, with Mirantis OpenContrail and Calico-based networking systems.

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.

Sprint 13

Reclass model

Summary

• Added the Prometheus Relay service
• Spawned two replicas of Prometheus to provide HA
• Added SSL support for cloud-monitoring services
• Removed GlusterFS requirements for Prometheus

New features

• Added the Prometheus Relay service. The Prometheus Relay service is responsible for getting PromQL queries from external components, such as Grafana, passing them to all discovered Prometheus servers, merging the results and returning the data. Prometheus Relay can be used to handle Prometheus high availability or sharding.

• Spawned two replicas of Prometheus inside Docker Swarm. These two instances have the same configuration file and scrape the same endpoints. Therefore, they can be treated as one-to-one copies. Due to a limitation, when connecting to the Prometheus web UI it is not possible to choose to which of the existing Prometheus servers to connect and those servers may contain slightly different results for the same queries. Therefore, we suggest that you use Grafana to visualize the data.

Upgrade notes

• Added SSL support for the following cloud-monitoring services:
  • Rundeck CIS Collectors
To provide SSL support for CIS, set up `cert` and `ssl_cert_file` on a cluster level metadata:

```plaintext
rundeck_cis_openstack:
  auth_url: ${_param:oss_openstack_auth_url}/auth/tokens
  username: ${_param:oss_openstack_username}
  password: ${_param:oss_openstack_password}
  cert:
      -----BEGIN CERTIFICATE-----
      MIIE0DCCA7igAwIBAgIBBzANBgkqhkiG9w0BAQsFADCBgzELMAkGA1UEBhMCVVMx
      -----END CERTIFICATE-----
  ssl_cert_file: cert.pem
```

If all parameters are defined properly, Rundeck enables the SSL support automatically.

- **Cleanup Service**

  To provide SSL support for Cleanup Service, specify the `cert` path and set the `ssl_verify` variable to True on a cluster level metadata:

  ```plaintext
  janitor_monkey_openstack:
    username: ${_param:oss_openstack_username}
    password: ${_param:oss_openstack_password}
    auth_url: ${_param:oss_openstack_auth_url}
    ssl_verify: True
    cacert_path: ${_param:oss_openstack_cert_path}
  ```

- **Security Audit Service**

  To provide SSL support for Security Audit Service, provide `cert` path, set the `ssl_verify` variable to True, and select the endpoint type for cloud connections on a cluster level metadata:

  ```plaintext
  security_monkey_openstack:
    username: ${_param:oss_openstack_username}
    password: ${_param:oss_openstack_password}
    auth_url: ${_param:oss_openstack_auth_url}
    ssl_verify: True
    endpoint_type: public
    cacert_path: ${_param:oss_openstack_cert_path}
  ```

**Note**

By default, the `cacert_path` variable is defined as follows:

```plaintext
oss_openstack_cert_path: /srv/volumes/rundeck/storage/content/keys/cis/openstack/cert.pem
```
Deprecation notes

• Removed GlusterFS from Prometheus requirements. Prometheus and Alertmanager do not require shared storage. Now, the configurations for Prometheus and Alertmanager will be generated on every Docker Swarm node.

Cluster model templates

Summary

• Added the Prometheus Relay service
• Removed GlusterFS requirements for Prometheus

New features

• Added the Prometheus Relay service. The Prometheus Relay service is responsible for getting PromQL queries from external components, such as Grafana, passing them to all discovered Prometheus servers, merging the results and returning the data. Prometheus Relay can be used to handle Prometheus high availability or sharding.

Deprecation notes

• Removed GlusterFS from Prometheus requirements. Prometheus and Alertmanager do not require shared storage. Now, the configurations for Prometheus and Alertmanager will be generated on every Docker Swarm node.

Jenkins pipelines

Summary

• Implemented MCP DriveTrain update pipelines

New features

• Added pipelines for the MCP DriveTrain automated update. Now, you can 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:

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

2. Apply the change:

   ```bash
   salt -C 'I@jenkins:client' state.apply jenkins.client
   ```
See also
Update MCP DriveTrain

StackLight

Summary

• Implemented monitoring for the InfluxDB Relay service

Upgrade notes

• Added a Grafana dashboard for the InfluxDB Relay service. The dashboard provides such metrics as the total amount of queries to the InfluxDB Relay instance, the number of error queries, the amount of data sent to a particular back end, the amount of data in the buffer, and so on.
• Added alerts that fire when InfluxDB Relay holds too much data in its buffer.