DAY 2 OPS FOR RED HAT OPENSTACK PLATFORM

By chrisj

FEB 26 2018

4

Red Hat OpenStack 12 is out, so it's time to re-test Day2 opstools.
Red Hat Cloudforms has been providing day2 operations and monitoring services for OpenStack for as long as I remember. However starting with Red Hat OSP 10 some new tools have been added to ease up operations. With Red Hat OSP 11 we ended up with three new agents - fluentd, sensu and collecd.

I am not going to focus on installing and integrating Red Hat Cloudforms. I will leave that piece for another blog post in the future. There is a lot of great documentation available on Red Hat website for OpenStack + Cloudforms integration. This is a good start:
https://access.redhat.com/documentation/en-us/red_hat_cloudforms/4.5/html/installing_red_hat_cloudforms_on_red_hat_enterprise_linux_openstack_platform/ (https://access.redhat.com/documentation/en-us/red_hat_cloudforms/4.5/html/installing_red_hat_cloudforms_on_red_hat_enterprise_linux_openstack_platform/)

Instead Cloudforms, I will describe installation and configuration of new third party agents and integrating them with third party dashboards.

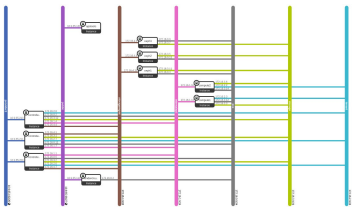
Agents and what they do:

Fluentd – open source data collector for logging
Integrates with: **Elasticsearch, Kibana**

Sensu - Monitor servers, services, application health
Integrates with: **Uchiwa**

Collectd - gathers metrics from various sources - operating system, applications, logfiles and external devices
Integrates with: **Grafana**

Architecture:
For this effort, I have built a quick reference architecture lab:



Please note 1 undercloud node, 3 controllers, 2 computes and 3 ceph nodes connected to standard TripleO networks
At the top a new node – opstools – connected just to public network, running vanilla RHEL 7.4 and connected to OSP12 repositories.

Installation Opstools server:

On pre-installed RHEL7 node:

```
[root@opstools ~]# yum install git ansible

[root@opstools ~]# git clone https://github.com/centos-opstools
/opstools-ansible.git

[root@opstools ~]# cd opstools-ansible/

[root@opstools opstools-ansible]# ssh-copy-id root@localhost
```

Two files need to be defined before executing the playbook – hosts inventory file and config.yaml that defined password, ports, network settings, security, etc.

```
[root@opstools opstools-ansible]# vi inventory/hosts
opstools ansible_host=localhost ansible_user=root ansible_become=true

[am_hosts]
opstools

[logging_hosts]
opstools

[pm_hosts]
opstools
```

```
[root@opstools opstools-ansible]# vi config.yml
grafana_username: admin
grafana_password: changme
uchiwa_credentials:
  - username: 'uchiwa'
    password: 'changeme'
kibana_credentials:
  - username: 'kibana'
    password: 'changeme'
data_storage: graphite
```

All the settings are described in here:
https://github.com/centos-opstools/opstools-ansible (https://github.com/centos-opstools/opstools-ansible)

Install all the dashboards with a single playbook

```
[root@opstools opstools-ansible]# ansible-playbook playbook.yml -e
@config.yml
```

The playbook is decent, but it's being modified constantly in true CI/CD fashion, so it's not unusual to hit a small bug. It is usually very easy to correct these bugs. Simply run the playbook with -vvvv to better identify failing component. Fix and re-run again. Most of the issues I hit are due to missing repository or typo in the package name.

The successfully deployed opstools server will result in following message:

```
PLAY RECAP
*********************************************************************************************************************************************************************
opstools                   : ok=187  changed=39   unreachable=0
failed=0
```

After playbooks complete, you can verify the functionality of 3 dashboards by opening them in your webbrowser:
https://<opstools-ip-or-host>/kibana

https://<opstools-ip-or-host>/uchiva
https://<opstools-ip-or-host>/grafana

Installation OpenStack:

First let me start with steps that are specific to OSP12. These steps will not apply to OSP11 or OSP10 (but will probably apply to OSP13 and above).
Since OSP12 introduced containerization of overcloud services, we need to ensure that we provide containers to fluentd, sensu and collectd.

When preparing container images I have included yaml configuration files for all 3 agents:

[osp12 specific]
```
(undercloud) [stack@chrisj-undercloud ~]$ openstack overcloud container
image prepare \
--namespace docker-registry.engineering.redhat.com/rhosp12 \
--set ceph_namespace=docker-registry.engineering.redhat.com/ceph \
--environment-file /usr/share/openstack-tripleo-heat-templates
/environments/ceph-ansible/ceph-ansible.yaml \
--environment-file /usr/share/openstack-tripleo-heat-templates
/environments/logging-environment.yaml \
--environment-file /usr/share/openstack-tripleo-heat-templates
/environments/monitoring-environment.yaml \
--environment-file /usr/share/openstack-tripleo-heat-templates
/environments/collectd-environment.yaml
```

[osp12 specific]
```
(undercloud) [stack@chrisj-undercloud ~]$ openstack overcloud container
image prepare \
--images-file ~/container-images.yaml \
--namespace docker-registry.engineering.redhat.com/rhosp12 \
--tag 12.0-20180124.1 \
--set ceph_namespace=docker-registry.engineering.redhat.com/ceph \
--environment-file /usr/share/openstack-tripleo-heat-templates
/environments/ceph-ansible/ceph-ansible.yaml \
--environment-file /usr/share/openstack-tripleo-heat-templates
/environments/logging-environment.yaml \
--environment-file /usr/share/openstack-tripleo-heat-templates
/environments/monitoring-environment.yaml \
--environment-file /usr/share/openstack-tripleo-heat-templates
/environments/collectd-environment.yaml
```

[osp12 specific]
```
(undercloud) [stack@chrisj-undercloud ~]$ openstack overcloud container
image prepare --env-file ~/templates/docker-registry.yaml --namespace
172.16.0.11:8787/rhosp12 --tag 12.0-20180124.1 --set
ceph_namespace=172.16.0.11:8787/ceph --environment-file /usr/share
/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
ansible.yaml \
--environment-file /usr/share/openstack-tripleo-heat-templates
/environments/logging-environment.yaml \
--environment-file /usr/share/openstack-tripleo-heat-templates
/environments/monitoring-environment.yaml \
--environment-file /usr/share/openstack-tripleo-heat-templates
/environments/collectd-environment.yaml
```

Continue with standard overcloud preparation
Steps below are valid for all OSP versions.

Next copy default opstools configuration yaml files to you local templates directory:
```
(undercloud) [stack@chrisj-undercloud ~]$ cp /usr/share/openstack-
tripleo-heat-templates/environments/logging-environment.yaml templates/
(undercloud) [stack@chrisj-undercloud ~]$ cp /usr/share/openstack-
tripleo-heat-templates/environments/monitoring-environment.yaml
templates/
(undercloud) [stack@chrisj-undercloud ~]$ cp /usr/share/openstack-
tripleo-heat-templates/environments/collectd-environment.yaml
templates/
```

Edit the files to include information about the opstools server and metrics that needs to be tracked

```
(undercloud) [stack@chrisj-undercloud templates]$ vi logging-
environment.yaml
## A Heat environment file which can be used to set up
## logging agents

resource_registry:
  OS::TripleO::Services::FluentdClient: /usr/share/openstack-tripleo-
heat-templates/docker/services/fluentd-client.yaml

parameter_defaults:

## Simple configuration
#
 LoggingServers:
   - host: 10.9.65.120
     port: 24224
#   - host: log1.example.com
#     port: 24224
#
## Example SSL configuration
## (note the use of port 24284 for ssl connections)
#
# LoggingServers:
#   - host: 192.168.24.11
#     port: 24284
# LoggingUsesSSL: true
# LoggingSharedKey: secret
# LoggingSSLCertificate: |
#   -----BEGIN CERTIFICATE-----
#   ...certificate data here...
#   -----END CERTIFICATE-----
```

```
(undercloud) [stack@chrisj-undercloud templates]$ vi monitoring-
environment.yaml
## A Heat environment file which can be used to set up monitoring
agents

resource_registry:
  OS::TripleO::Services::SensuClient: /usr/share/openstack-tripleo-
heat-templates/docker/services/sensu-client.yaml

parameter_defaults:
  MonitoringRabbitHost: 10.9.65.120
  MonitoringRabbitPort: 5672
  MonitoringRabbitUserName: sensu
  MonitoringRabbitPassword: sensu
#  MonitoringRabbitUseSSL: false
#  MonitoringRabbitVhost: "/sensu"
#  SensuClientCustomConfig:
#    api:
#      warning: 10
#      critical: 20
```

```
(undercloud) [stack@chrisj-undercloud templates]$ vi collectd-
environment.yaml
resource_registry:
  OS::TripleO::Services::Collectd: /usr/share/openstack-tripleo-heat-
templates/docker/services/collectd.yaml

parameter_defaults:
#
## Collectd server configuration
   CollectdServer: 10.9.65.120
#
###############
#### Other config parameters, the values shown here are the defaults
###############
#
#   CollectdServerPort: 25826
#   CollectdSecurityLevel: None
#
###############
#### If CollectdSecurityLevel is set to Encrypt or Sign
#### the following parameters are also needed
###############
#
#   CollectdUsername: user
#   CollectdPassword: password
#
## CollectdDefaultPlugins, These are the default plugins used by
collectd
```

```
#
   CollectdDefaultPlugins:
     - disk
     - interface
     - load
     - memory
     - processes
     - tcpconns
#
## Extra plugins can be enabled by the CollectdExtraPlugins parameter:
## All the plugins availables are:
#
   CollectdExtraPlugins:
     - disk
     - df
     - cpu
#
## You can use ExtraConfig (or one of the related *ExtraConfig keys)
## to configure collectd.  See the documentation for puppet-collectd at
## https://github.com/voxpupuli/puppet-collectd for details.
#
   ExtraConfig:
     collectd::plugin::disk::disks:
       - "/^[vhs]d[a-f][0-9]?$/"
     collectd::plugin::df::mountpoints:
       - "/"
     collectd::plugin::df::ignoreselected: false
     collectd::plugin::cpu::valuespercentage: true
```

Please note Resource registry section with absolute path change. Additional information on setting up these files can be found in here:
https://docs.openstack.org/tripleo-docs/latest/install/advanced_deployment/ops_tools.html

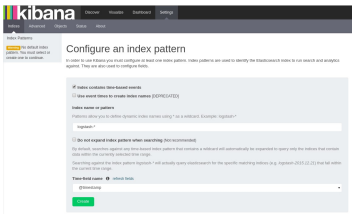
Finally make sure to include the newly create files in your deploy command. Example:

```
(undercloud) [stack@chrisj-undercloud ~]$ cat deploy.sh
#!/bin/bash

source ~/stackrc
cd ~/
time openstack overcloud deploy --templates --stack chrisj \
    --ntp-server 10.9.71.7 \
    -e templates/network-environment.yaml \
    -e templates/node-info.yaml \
    -e templates/docker-registry.yaml \
    -e templates/host-memory.yaml \
    -e /usr/share/openstack-tripleo-heat-templates/environments
/network-isolation.yaml \
    -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-
ansible/ceph-ansible.yaml \
    -e templates/ceph-custom-config.yaml \
    -e templates/logging-environment.yaml \
    -e templates/monitoring-environment.yaml \
    -e templates/collectd-environment.yaml
```

How to use Dashboards:

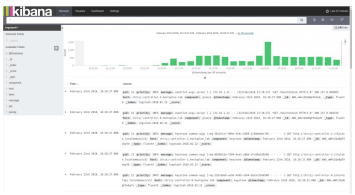
1. Kibana – logging
After accessing dashboard (https://<ops-tools-ip/kibana) for the first time, you will be greated by the following screen:



simply select 'Create' button

Going to Discover tab will show you all the logs displayed from all the overcloud nodes. You can search for specific message or filter them out in any specific way desired.

There is also way to visualize the data.



2. Uchiwa – monitoring

Unfortunately out-of-the-box Openstack health checks no longer apply to OSP12 services. This is due to containerization of all services and inability to verify health with systemd. BZ have been already raised:

https://bugzilla.redhat.com/show_bug.cgi?id=1498360 (https://bugzilla.redhat.com/show_bug.cgi?id=1498360)

https://bugzilla.redhat.com/show_bug.cgi?id=1510408 (https://bugzilla.redhat.com/show_bug.cgi?id=1510408)

and potential fix:

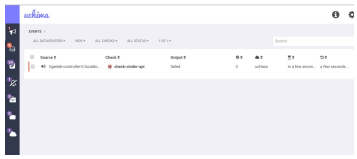
https://review.rdoproject.org/r/#/c/10731/ (https://review.rdoproject.org/r/#/c/10731/)

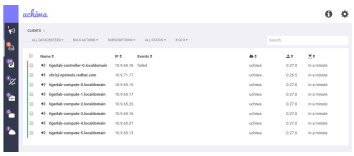
In OSP11 and OSP10 however (and hopefully in future releases) you could take advantage of the healthchecks.

There are checks for most of the OpenStack standard service. The general idea is, if something goes down or have operational issues, the alert will appear in event section (main page).

If you get an alert that doesn't apply to your environment, you can simply silence it or even remove.

Here is the example of RH OSP11 in a healthy state:



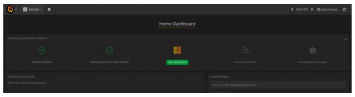


The one failed check is dues to openstack-cinder-api service moving to httpd in OSP11, so it can be silenced

3. Grafana – performance

Grafana dashboard (https://<ops-tools-ip/grafana) requires little more tuning before being able to display any data.

The initial screen will ask you to create new dashboard:



Select dashboard and start playing with it:

- select 'graph'

- select 'panel title' at the top and 'Edit'

- in general tab change title to 'CPU Load - Controllers'

- in metrics tab under 'data source' select Graphite or default

- then 'select metric' → collectd

- 'select metric' and type → **"controller"**

- 'select metric' → *

- 'select metric' → Load

- 'select metric' → shortterm

The end result should look something like this:

it's a nice graph for tracking CPU load.

There is tons of more options that could be measured. I also found this quick 10 minutes tutorial in here:

https://youtu.be/sKNZMtoSHN4 (https://youtu.be/sKNZMtoSHN4)

This concludes the installation procedure of the opstools for Red Hat OpenStack. Happy hacking!

Tags:
collectd (/?q=taxonomy/term/1) grafana (/?q=taxonomy/term/2) sensu (/?q=taxonomy/term/3)
uchiwa (/?q=taxonomy/term/4) fluentd (/?q=taxonomy/term/5) kibana (/?q=taxonomy/term/6)
logstash (/?q=taxonomy/term/7) elasticsearch (/?q=taxonomy/term/8) openstack (/?q=taxonomy/term/9)
osp12 (/?q=taxonomy/term/10)

ADD NEW COMMENT (/?Q=NODE/5#COMMENT-FORM)

THERE ARE 4 COMMENTS

ONE SPECIFIC DEPENDENCY I HAD (/?Q=COMMENT/5#COMMENT-5)

Submitted by Chris Paquin on Thu, 03/29/2018 - 15:29

One specific dependency I had to install on Centos 7 was python-netaddr.noarch

reply (/?q=comment/reply/5/5)

I HAVEN'T SEEN THIS ON RHEL7, (/?Q=COMMENT/7#COMMENT-7)

Submitted by chrisj on Sat, 04/21/2018 - 02:11

I haven't seen this on RHEL7, thanks for the input.

reply (/?q=comment/reply/5/7)

I'D SUGGEST TO CLONE THE GIT (/?Q=COMMENT/6#COMMENT-6)

Submitted by Matthias Runge on Mon, 04/02/2018 - 15:03

I'd suggest to clone the git repo to a laptop and to run ansible from there. I would not recommend to log into any machine as root via ssh, since most machines should have root logins disabled.

reply (/?q=comment/reply/5/6)

HEY MATTHIAS, (/?Q=COMMENT/8#COMMENT-8)

Submitted by chrisj on Sat, 04/21/2018 - 02:13

Hey Matthias,

Thanks for your comment. Definetly a valid statement. I have taken some shortcuts here.

Also, I am not encrypting any of the connections for the agents, which is probably not a best practice.

reply (/?q=comment/reply/5/8)

ADD NEW COMMENT

Your name

Comment *

- No HTML tags allowed.
- Web page addresses and e-mail addresses turn into links automatically.
- Lines and paragraphs break automatically.

More information about text formats (/?q=filter/tips)

SAVE PREVIEW