Tripleo/Director Debugging

Table of Contents

Undercloud 1
- Undercloud install Recommendations 1
- Undercloud install Debug 2
- Enrolling overcloud Nodes 2

Introspection 3
- Introspection Recommendations 3
- Introspection Debug 4

Deploy/Provision phase 10
- Deploy phase recommendations 10
- Deploy phase Debug 11

Overcloud Deploy 16
- Overcloud templates 16
- Mistral Plans Debug 16
- Overcloud Deploy Debug 22

Undercloud

Undercloud install Recommendations

Undercloud recommended specs

- CPUs: 2x Physical cores, 24 threads
- Disk: (single ssd or 2x drives 7200RPM) Root disk / raid1 Disk for swift
- Memory: 64G
- Network : 10G for provisioning

undercloud.conf options

```
 undercloud_debug = false  -->  only enable debug if needed
enable_telemetry = false  -->  disable telemetry if its not needed
automated_clean = true    --> (is osp11 as clean_nodes=true) runs wipefs --force --all
before making node available
enable_tempest = true     --> if you want to certifie your overcloud after deploy
enable_ui = false         --> tripleo UI
enable_validations = true --> Overcloud Ansible Validations
enabled_drivers = pxe_ipmitool,pxe_drac --> if you need to enable a specific driver
```

You can also override hieradata of the undercloud install,undercloud.conf

```
hieradata_override = /home/stack/hieradata-override.yaml
```

And in the override file for example

```
 cat > /home/stack/hieradata-override.yaml << __EOF__
neutron::dns_domain: example.com
nova::network::neutron::dhcp_domain: example.com
__EOF__
```

Undercloud tuning

Keystone worker counts: increase the worker count to available cpus/2 if you have enough ram

```
 [root@undercloud ~]# cat /etc/httpd/conf.d/10-keystone_wsgi_admin.conf | grep
processes
  WSGIDaemonProcess keystone_admin display-name=keystone-admin group=keystone
processes=4 threads=2 user=keystone
[root@undercloud ~]# cat /etc/httpd/conf.d/10-keystone_wsgi_main.conf  | grep
processes
  WSGIDaemonProcess keystone_main display-name=keystone-main group=keystone
processes=4 threads=2 user=keystone
```

Heat RPC response timeout, increase to 600 to be on the safe side

```
 [root@undercloud ~]# cat /etc/heat/heat.conf | grep ^rpc_response_timeout
rpc_response_timeout = 600
```

Undercloud install Debug

TODO!!

Enrolling overcloud Nodes

When enrolling overcloud nodes with the import command, the ironic driver tests the connectivity with the Ilo/drac

```
[stack@undercloud ~]$ openstack baremetal import --json ~/instackenv.json
```

If the command takes a long time to finish and the nodes are in enroll state

```
 [stack@undercloud conf.d]$ openstack baremetal list
This command is deprecated. Instead, use 'openstack baremetal node list'.
+--------------------------------------+------+--------------+-------------
+--------------------+-------------+
| UUID                                 | Name | Instance UUID | Power State |
Provisioning State | Maintenance |
+--------------------------------------+------+--------------+-------------
+--------------------+-------------+
| 149a4a65-739c-4599-823c-cf0c6dbcd37c | None | None          | None        | enroll
| False       |
| 82079e5a-b02e-443c-bf6b-51b4cb6b149f | None | None          | None        | enroll
| False       |
| ac0b308f-15bd-43f2-8aaa-fc9c872c0343 | None | None          | None        | enroll
| False       |
| 6a05da8a-5963-4a19-a419-e71ef684e3e9 | None | None          | None        | enroll
| False       |
+--------------------------------------+------+--------------+-------------
+--------------------+-------------+
```

Check the ironic-conductor.log for problems communicating the IPMI/ilo/drac BMC platforms:

```
 2017-06-11 10:27:14.977 10009 ERROR ironic.conductor.manager [req-afc9a72c-58bf-4fd0-
a23d-d8d43905bbde - - - - -] Failed to validate power driver interface for node
149a4a65-739c-4599-823c-cf0c6dbcd37c. Error: SSH connection cannot be established:
Failed to establish SSH connection to host 192.168.101.1.
2017-06-11 10:27:14.984 10009 ERROR ironic.conductor.manager [req-f614ec7f-e7f0-4231-
97b7-65ee3f248110 - - - - -] Failed to validate power driver interface for node
6a05da8a-5963-4a19-a419-e71ef684e3e9. Error: SSH connection cannot be established:
Failed to establish SSH connection to host 192.168.101.1.
2017-06-11 10:27:15.227 10009 ERROR ironic.conductor.manager [req-bde5bf5c-e89b-4cc5-
89f8-c857f8392e41 - - - - -] Failed to validate power driver interface for node
ac0b308f-15bd-43f2-8aaa-fc9c872c0343. Error: SSH connection cannot be established:
Failed to establish SSH connection to host 192.168.101.1.
2017-06-11 10:27:15.232 10009 ERROR ironic.conductor.manager [req-987d2ef5-129d-401d-
9c8c-ff3a856b9fb1 - - - - -] Failed to validate power driver interface for node
82079e5a-b02e-443c-bf6b-51b4cb6b149f. Error: SSH connection cannot be established:
Failed to establish SSH connection to host 192.168.101.1.
```

Fix the problem with the BMC connection, and rerun the baremetal import command again

Introspection

Introspection Recommendations

- Try small instrospection bundles first.

- Do not introspect more nodes than your DHCP range allows.
- After introspection, wait for about two minutes to let DHCP leases to expire.
- Watch for any additional DHCP servers on the provisioning network

Introspection Debug

Logs

```
 [root@undercloud ironic]# ls -l /var/log/ironic-inspector/*.log
-rw-r--r--. 1 ironic-inspector ironic-inspector 32080 Jun  1 10:37 /var/log/ironic-
inspector/ironic-inspector.log
[root@undercloud ironic]# ls -l /var/log/ironic/*.log
-rw-r--r--. 1 ironic ironic 2238295 Jun  3 08:26 /var/log/ironic/ironic-api.log
-rw-r--r--. 1 ironic ironic  235499 Jun  2 07:54 /var/log/ironic/ironic-conductor.log
root@undercloud ~]# journalctl -u openstack-ironic-inspector -u openstack-ironic-
inspector-dnsmasq
Jun 07 09:07:57 undercloud.localdomain ironic-inspector[1236]: 2017-06-07 09:07:57.353
1236 INFO ironic_inspector.node_cache [-] [node: ba4db2bf-c327-4fca-a37e-83cc17c1561a
state processing] Updating node state: processing --> finished
Jun 07 09:07:57 undercloud.localdomain ironic-inspector[1236]: 2017-06-07 09:07:57.347
1236 INFO ironic_inspector.process [-] [node: ba4db2bf-c327-4fca-a37e-83cc17c1561a MAC
aa:bb:cc:dd:ee:02] Introspection finished successfully
Jun 07 09:07:57 undercloud.localdomain ironic-inspector[1236]: 2017-06-07 09:07:57.315
1236 INFO ironic_inspector.process [-] [node: ba4db2bf-c327-4fca-a37e-83cc17c1561a MAC
aa:bb:cc:dd:ee:02] Node powered-off
```

Accessing logs from the ram disk run

If something fails during the introspection ramdisk run, ironic-inspector stores the ramdisk logs in /var/log/ironic-inspector/ramdisk/ as gz-compressed tar files. File names contain date, time and IPMI address of the node if it was detected (only for bare metal).

If you want to inspect the ram disk run even if it didn't fail we have to modify the always_store_ramdisk_logs option in /etc/ironic-inspector/inspector.conf and restart the service

```
 [root@undercloud ~]# sed -ibck
's/^.*always_store_ramdisk_logs.*$/always_store_ramdisk_logs = true/g' /etc/ironic-
inspector/inspector.conf
[root@undercloud ~]#  openstack baremetal introspection start ba4db2bf-c327-4fca-a37e-
83cc17c1561a
[root@undercloud ~]# systemctl restart openstack-ironic-inspector.service
[root@undercloud ~]# ls -l /var/log/ironic-inspector/ramdisk/
total 20
-rw-r--r--. 1 ironic-inspector ironic-inspector 16820 Jun  7 09:16 ba4db2bf-c327-4fca-
a37e-83cc17c1561a_20170607-131619.504127.tar.gz
```

Log into the ramdisk for debugging via ssh/console

Use ssl to create a hash for a password

```
 [root@undercloud ~]# openssl passwd -1
Password:
Verifying - Password:
$1$ZqPeffYv$CvGO/oS8b28YRdMMS2WCF1
```

Edit /httpboot/inspector.ipxe manually. Find the line starting with “kernel” and append rootpwd=”HASH” to it, also disable selinux with the selinux=0 option

```
[root@undercloud ~]# cat /httpboot/inspector.ipxe
#!ipxe
:retry_boot
imgfree
kernel --timeout 60000 http://10.0.0.10:8088/agent.kernel ipa-inspection-callback-
url=http://10.0.0.10:5050/v1/continue ipa-inspection-collectors=default,extra-
hardware,logs systemd.journald.forward_to_console=yes BOOTIF=${mac} ipa-inspection-
dhcp-all-interfaces=1 ipa-collect-lldp=1 initrd=agent.ramdisk
rootpwd="$1$UQ/HlKRP$pXaAJKgSS7z7SPqOTH0FV/" selinux=0 || goto retry_boot
initrd --timeout 60000 http://10.0.0.10:8088/agent.ramdisk || goto retry_boot
boot
```

We can see the options we added to the cmdline being loaded on the next ram disk run

```
ramdisk/journal:Jun 07 09:14:38 localhost.localdomain kernel: Command line: ipa-
inspection-callback-url=http://10.0.0.10:5050/v1/continue ipa-inspection-
collectors=default,extra-hardware,logs systemd.journald.forward_to_console=yes
BOOTIF=aa:bb:cc:dd:ee:02 ipa-inspection-dhcp-all-interfaces=1 ipa-collect-lldp=1
initrd=agent.ramdisk rootpwd="$1$UQ/HlKRP$pXaAJKgSS7z7SPqOTH0FV/" selinux=0
```

Modify ramdisk image

If you need to modify the ramdisk image to fix some issue you can follow these steps:

```
[root@undercloud httpboot]# cp /httpboot/agent.ramdisk /root/agent.ramdisk
[root@undercloud ~]# mkdir agent.ramdisk.dir
[root@undercloud ~]# cd agent.ramdisk.dir
[root@undercloud agent.ramdisk.dir]# gzip -dc ../agent.ramdisk | cpio --extract
1872427 blocks
[root@undercloud agent.ramdisk.dir]# ls
bin  boot  dev  etc  home  init  lib  lib64  lost+found  media  mnt  opt  proc  root
run  sbin  srv  sys  tmp  usr  var
```

After modifying the image, we zip it again and move it into the httpboot dir

```
[root@undercloud httpboot]# find . | cpio -oc | gzip -c -9>| ~/agent.ramdisk.root-test
[root@undercloud httpboot]# cp ~/agent.ramdisk.root-test /httpboot/agent.ramdisk
```

Check selinux permissions are ok:

```
[root@undercloud httpboot]#chcon system_u:object_r:httpd_user_content_t:s0
agent.ramdisk
```

Check the Ironic driver you are using is enabled and available

```
[root@undercloud ~]# openstack baremetal driver list
+---------------------+------------------------+
| Supported driver(s) | Active host(s)         |
+---------------------+------------------------+
| ipmi                | undercloud.localdomain |
| pxe_drac            | undercloud.localdomain |
| pxe_ilo             | undercloud.localdomain |
| pxe_ipmitool        | undercloud.localdomain |
| pxe_ssh             | undercloud.localdomain |
+---------------------+------------------------+
```

If you have power on/power off timeout problems in Ironic(https://access.redhat.com/solutions/2332151), increase the power_retry and power_wait parameters

```
# Options defined in ironic.drivers.modules.ilo.power
#

# Number of times a power operation needs to be retried
# (integer value)
power_retry=6

# Amount of time in seconds to wait in between power
# operations (integer value)
power_wait=20
```

If using IPMI driver, and errors are seen check:

- Ipmitool is installed.
- The IPMI controller on your bare metal server is turned on.
- The IPMI controller credentials passed in the command are right.
- The conductor node has a route to the IPMI controller. This can be checked by just pinging the IPMI controller IP from the conductor node.

Also test that you can access the status of the BMC using the ipmitool command several times

```
ipmitool -I lanplus -H <ip-address> -U <username> -P <password> chassis power status
```

slow or unresponsive BMCs in the environment

the retry_timeout configuration option in the [ipmi] section may need to be increased. The default is fairly conservative, as setting this timeout too low can cause older BMCs to crash and require a hard-reset.

```
[root@undercloud ~]# more /etc/ironic/ironic.conf | grep ^retry_timeout
retry_timeout = 15
```

Check Bios config "Legacy" or "UEFI"

Out-of-the-box install of Red Hat OpenStack Platform 10 sets ironic.conf on the undercloud in default_boot_mode = bios

The default state of the ironic.conf file on the undercloud is:

```
default_boot_mode = bios
```

If uefi boot is needed in introspection and deploy, change it to:

```
default_boot_mode = uefi
```

DHCP info and debug for introspection

The introspection dhcp service uses dnsmasq:

```
[root@undercloud ~]# ps -ef | grep -i ironic-inspector | grep dns
nobody    1310     1  0 Jun01 ?        00:00:00 /sbin/dnsmasq --conf-file=/etc/ironic
-inspector/dnsmasq.conf
```

The conf file is in /etc/ironic-inspector/dnsmasq.conf, you can check interface and dhcp-range:

```
[root@undercloud ~]# cat /etc/ironic-inspector/dnsmasq.conf
port=0
interface=br-ctlplane
bind-interfaces
dhcp-range=10.0.0.100,10.0.0.120,29
dhcp-sequential-ip
dhcp-match=ipxe,175
dhcp-match=set:efi,option:client-arch,7
# Client is running iPXE; move to next stage of chainloading
dhcp-boot=tag:ipxe,http://10.0.0.10:8088/inspector.ipxe
# Client is running PXE over EFI; send EFI version of iPXE chainloader
dhcp-boot=tag:efi,ipxe.efi
# Client is running PXE over BIOS; send BIOS version of iPXE chainloader
dhcp-boot=undionly.kpxe,localhost.localdomain,10.0.0.10
```

Use tcpdump on the provisioning interface:

```
[root@undercloud ~]# tcpdump -i <network-interface> port 67 or port 68 or port 69 -e
-n
```

Filter matching the client Mac address:

```
tcpdump -i br0 -vvv -s 1500 '((port 67 or port 68) and (udp[38:4] = 0x3e0ccf08))'
```

Tcpdump filter to capture packets sent by the client (DISCOVER, REQUEST, INFORM):

```
tcpdump -i br0 -vvv -s 1500 '((port 67 or port 68) and (udp[8:1] = 0x1))'
```

If the Dhcp client isn't able to comunicate with the Dhcp server

Check that the client Mac is getting whitelisted in Iptables, We can look for the mac address of the overcloud node that we are going to introspect, in this case the node uid is **50fd27b6-7af7-425e-94b2-f6fb0f9f5bfa** with mac **aa:bb:cc:dd:ee:01**

```
 [root@undercloud ~]# openstack baremetal port list
+--------------------------------------+-------------------+
| UUID                                 | Address           |
+--------------------------------------+-------------------+
| 5da9df9d-adea-4d89-ade2-f083478cdfbf | aa:bb:cc:dd:ee:01 |
| 3a2687cf-318b-4c80-bc18-01c20b49ed29 | aa:bb:cc:dd:ee:02 |
| 04cf3e6a-c27f-4ce4-9719-93dfe82e48db | aa:bb:cc:dd:ee:03 |
| 24b39ae8-1009-4d21-9269-683f010790bf | aa:bb:cc:dd:ee:04 |
+--------------------------------------+-------------------+
[root@undercloud ~]# openstack baremetal port show 5da9df9d-adea-4d89-ade2-
f083478cdfbf
+------------+--------------------------------------+
| Field      | Value                                |
+------------+--------------------------------------+
| address    | aa:bb:cc:dd:ee:01                    |
| created_at | 2017-06-01T13:57:08+00:00            |
| extra      | {}                                   |
| node_uuid  | 50fd27b6-7af7-425e-94b2-f6fb0f9f5bfa |
| updated_at | 2017-06-02T11:53:59+00:00            |
| uuid       | 5da9df9d-adea-4d89-ade2-f083478cdfbf |
+------------+--------------------------------------+
```

We now are going to run the introspection on that node and check the iptable rules, our mac is going to get white listed and removed from the macs with the DROP target

```
 [root@undercloud ~]# iptables -nL | grep -A 2 "Chain ironic-inspector"
Chain ironic-inspector (1 references)
target     prot opt source               destination
REJECT     all  --  0.0.0.0/0            0.0.0.0/0            reject-with icmp-port-
unreachable
[root@undercloud ~]# openstack  baremetal introspection start 50fd27b6-7af7-425e-94b2-
f6fb0f9f5bfa
[root@undercloud ~]# iptables -nL | grep -B 2 -A 2 -i AA:BB
Chain ironic-inspector (1 references)
target     prot opt source               destination
DROP       all  --  0.0.0.0/0            0.0.0.0/0            MAC AA:BB:CC:DD:EE:04
DROP       all  --  0.0.0.0/0            0.0.0.0/0            MAC AA:BB:CC:DD:EE:03
DROP       all  --  0.0.0.0/0            0.0.0.0/0            MAC AA:BB:CC:DD:EE:02
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
```

We can also check in the logs the same process in the introspection logs

```
 [root@undercloud ironic-inspector]# grep -i 'aa:bb:cc:dd:ee:01' ironic-inspector.log
2017-06-07 16:48:49.196 7354 INFO ironic_inspector.introspect [-] [node: 50fd27b6-
7af7-425e-94b2-f6fb0f9f5bfa state starting] Whitelisting MAC's [u'aa:bb:cc:dd:ee:01']
on the firewall
2017-06-07 16:48:51.688 7354 INFO ironic_inspector.introspect [-] [node: 50fd27b6-
7af7-425e-94b2-f6fb0f9f5bfa state starting] The following attributes will be used for
look up: {u'mac': [u'aa:bb:cc:dd:ee:01']}
```

Introspection Failure, Kill it

Sometimes introspection may fail. Unfortunately there is no good way of ending the process. To cancel introspection follow the following steps

```
 [stack@undercloud ~]$ ironic node-set-power-state [NODE UUID] off
[stack@undercloud ~]$ sudo rm /var/lib/ironic-discoverd/discoverd.sqlite
[stack@undercloud ~]$ sudo systemctl restart openstack-ironic-discoverd
```

Deploy/Provision phase

Deploy phase recommendations

Try small scale deployment first

Try deployment with the smallest number of nodes possible,Single Controller, single Compute, single CephStorage, etc.

Fixed sized Deployment batches

We recommend not deploying 32 nodes at a time. 32 is the typical amount you can fit within a 42 RU rack. Deploying 32 at a time also minimizes the debugging necessary to diagnose issues with the deployment.

Configure even unused NICs

Specify NIC configurations for network interfaces unused by OpenStack Define interfaces in YAML files in the nic-configs directory:

```
Set use_dhcp: false and defroute: false
```

Power off unused nodes

Before redeployment, verify that unused nodes are OFF Use DRAC/iLO to check power state Do not relay on Ironic, We have seen cases where nodes from previous deployments which are now in maintenance, are left hanging around in a powered on state causing problems with ongoing

deployments.

Deploy phase Debug

In this context we consider deploy the Phase during which the overcloud image is copied to the local overcloud nodes, this is done booting the undercloud nodes via ipxe to load a ramdisk, then exporting the local root disk as a iscsi target to the undercloud node, and then the overcloud image gets copied via dd

Deploy Logs

If the deploy fails during deploy the ironic-conductor.log should have the error

```
[root@undercloud ironic]# ls -l /var/log/ironic/ironic-conductor.log
-rw-r--r--. 1 ironic ironic 288970 Jun  7 09:16 /var/log/ironic/ironic-conductor.log
```

If you need to check the ramdisk run log output, it is saved in /var/log/ironic/deploy/ dir

```
 [root@undercloud ironic]# ls -ld  /var/log/ironic/deploy/
drwxr-xr-x. 2 ironic ironic 4096 Jun  5 02:48 /var/log/ironic/deploy/
```

Inside the dir you can find a tarball named after the UID of the overcloud node, the journal file contains the output of the run

```
 [root@undercloud ironic]# ls -l  /var/log/ironic/deploy/*tar.gz
-rw-r--r--. 1 ironic ironic 17573 Jun  5 02:48 /var/log/ironic/deploy/d8ee5249-a016-
444f-bd02-77e422332381_32068670-ef4e-4699-911f-6c0ed4da541c_2017-06-05-06:48:24.tar.gz
-rw-r--r--. 1 ironic ironic 17806 Jun  5 02:48 /var/log/ironic/deploy/e4cf9855-392c-
40be-92b6-9b10674373d0_d92dc0f2-d03c-4cf4-9d1e-36772055520a_2017-06-05-06:48:23.tar.gz
[root@undercloud ironic]# tar -zxvf /var/log/ironic/deploy/d8ee5249-a016-444f-bd02-
77e422332381_32068670-ef4e-4699-911f-6c0ed4da541c_2017-06-05-06:48:24.tar.gz && tail
journal
Jun 05 02:48:20 host-10-0-0-62 logger[2251]: 83haiku: debug: /dev/vda1 is not a BeFS
partition: exiting
Jun 05 02:48:20 host-10-0-0-62 logger[2251]: 50mounted-tests: debug: running subtest
/usr/libexec/os-probes/mounted/90linux-distro
Jun 05 02:48:20 host-10-0-0-62 logger[2251]: 50mounted-tests: debug: running subtest
/usr/libexec/os-probes/mounted/90solaris
Jun 05 02:48:20 host-10-0-0-62 logger[2251]: 50mounted-tests: debug: running subtest
/usr/libexec/os-probes/mounted/efi
Jun 05 02:48:21 host-10-0-0-62 ironic-python-agent[525]: 2017-06-05 02:48:21.149 525
INFO ironic_python_agent.extensions.image [-] GRUB2 successfully installed on /dev/vda
Jun 05 02:48:21 host-10-0-0-62 kernel: XFS (vda2): Unmounting Filesystem
Jun 05 02:48:21 host-10-0-0-62 ironic-python-agent[525]: 2017-06-05 02:48:21.285 525
INFO root [-] Command image.install_bootloader completed: Command name:
install_bootloader, params: {u'efi_system_part_uuid': None, u'root_uuid': u'2a59886b-
5268-41e5-b37f-c92611eabd96'}, status: SUCCEEDED, result: None.
Jun 05 02:48:21 host-10-0-0-62 ironic-python-agent[525]: ::ffff:10.0.0.10 - -
[05/Jun/2017 02:48:21] "POST /v1/commands?wait=true HTTP/1.1" 200 265
Jun 05 02:48:23 host-10-0-0-62 NetworkManager[186]: <warn>  [1496645303.8756] dhcp4
(eth2): request timed out
```

PXE booting in the Deployment Phase

The dhcp deployment service is different from the introspection dhcp service, the deployment dnsmasq processes is running inside a namespace created by neutron, when the Ipxe boot starts you will see with ironic node-list the node in wait for callback state, if the node hangs in the wait for callback state the undercloud node is having issues booting ipxe using dhcp, a review of the boot proccess from the system Ilo/console is needed.

```
 [root@undercloud ~]# ps -ef | grep -i dhcp-hostsfile
nobody    4243     1  0 Jun01 ?        00:00:00 dnsmasq --no-hosts --no-resolv
--strict-order --except-interface=lo --pid-file=/var/lib/neutron/dhcp/0ba89bb7-dccd
-4001-bb12-5b53ed82c594/pid --dhcp-hostsfile=/var/lib/neutron/dhcp/0ba89bb7-dccd-4001
-bb12-5b53ed82c594/host --addn-hosts=/var/lib/neutron/dhcp/0ba89bb7-dccd-4001-bb12
-5b53ed82c594/addn_hosts --dhcp-optsfile=/var/lib/neutron/dhcp/0ba89bb7-dccd-4001-bb12
-5b53ed82c594/opts --dhcp-leasefile=/var/lib/neutron/dhcp/0ba89bb7-dccd-4001-bb12
-5b53ed82c594/leases --dhcp-match=set:ipxe,175 --bind-interfaces
--interface=tap9fa1fa7b-14 --dhcp-range=set:tag0,10.0.0.0,static,86400s --dhcp-option
-force=option:mtu,1500 --dhcp-lease-max=256 --conf-file=/etc/dnsmasq-ironic.conf
```

```
 [root@undercloud ~]# ip netns
qdhcp-0ba89bb7-dccd-4001-bb12-5b53ed82c594
[root@undercloud ~]# ip netns exec qdhcp-0ba89bb7-dccd-4001-bb12-5b53ed82c594 ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
9: tap9fa1fa7b-14: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UNKNOWN qlen 1000
    link/ether fa:16:3e:ae:bf:50 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.50/24 brd 10.0.0.255 scope global tap9fa1fa7b-14
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:feae:bf50/64 scope link
       valid_lft forever preferred_lft forever
```

So if we want to Tcpdump is better if we do it inside the namespace using the tap device:

```
[root@undercloud ~]# ip netns exec qdhcp-0ba89bb7-dccd-4001-bb12-5b53ed82c594 tcpdump
-i tap9fa1fa7b-14 -vvv -s 1500 '((port 67 or port 68))'
```

When the undercloud is a virtual machine running on VMware ESXi, DHCP during Introspection is successful, but it fails during deployment. DHCP requests are being received on the tap device, but the offers are not received by the nodes Forged transmit has to be set to Accept so ESXi does not compare source and effective MAC addresses. (https://access.redhat.com/solutions/1980283)

Populating the overcloud local disks with the overcloud image

When we run the overcloud deploy, and the ipxe boot has loaded the ramdisk, a iscsi target is created for the local root disk, from the journal of the deploy ram disk run for and overcloud node:

```
 Jun 03 09:07:00 host-10-0-0-55 ironic-python-agent[528]: 2017-06-03 09:07:00.668 528
INFO ironic_python_agent.extensions.iscsi [-] Created iSCSI target with iqn iqn.2008-
10.org.openstack:e4cf9855-392c-40be-92b6-9b10674373d0, portal port 3260, on device
/dev/vda using linux-io
Jun 03 09:07:00 host-10-0-0-55 ironic-python-agent[528]: 2017-06-03 09:07:00.669 528
INFO root [-] Command iscsi.start_iscsi_target completed: Command name:
start_iscsi_target, params: {u'wipe_disk_metadata': True, u'iqn': u'iqn.2008-
10.org.openstack:e4cf9855-392c-40be-92b6-9b10674373d0', u'portal_port': 3260}, status:
SUCCEEDED, result: {'iscsi_target_iqn': u'iqn.2008-10.org.openstack:e4cf9855-392c-
40be-92b6-9b10674373d0'}.
Jun 03 09:07:00 host-10-0-0-55 ironic-python-agent[528]: ::ffff:10.0.0.10 - -
[03/Jun/2017 09:07:00] "POST /v1/commands?wait=true HTTP/1.1" 200 386
```

Now we can see the isci initiator in the undercloud node is connected to the target:

```
 [root@undercloud ~]# iscsiadm -m session -P 2
Target: iqn.2008-10.org.openstack:d8ee5249-a016-444f-bd02-77e422332381 (non-flash)
    Current Portal: 10.0.0.52:3260,1
    Persistent Portal: 10.0.0.52:3260,1
        **********
        Interface:
        **********
        Iface Name: default
        Iface Transport: tcp
        Iface Initiatorname: iqn.1994-05.com.redhat:10481c98aef
        Iface IPaddress: 10.0.0.10
        Iface HWaddress: <empty>
        Iface Netdev: <empty>
        SID: 10
        iSCSI Connection State: LOGGED IN
        iSCSI Session State: LOGGED_IN
```

After this the copy of the undercloud image starts, if the dd process hangs and doesn't finish, load a live usb/iso on the overcloud node that is failing and test the root disk, do a local dd, check for PFAs in the SMART stats,etc.

```
 [root@undercloud ~]# ps -ef | grep -i dd
root      6715  1306  0 05:18 ?        00:00:00 sudo ironic-rootwrap
/etc/ironic/rootwrap.conf dd if=/var/lib/ironic/images/e4cf9855-392c-40be-92b6-
9b10674373d0/disk of=/dev/disk/by-path/ip-10.0.0.60:3260-iscsi-iqn.2008-
10.org.openstack:e4cf9855-392c-40be-92b6-9b10674373d0-lun-1-part2 bs=1M oflag=direct
root      6717  6715  0 05:18 ?        00:00:00 /usr/bin/python2 /usr/bin/ironic-
rootwrap /etc/ironic/rootwrap.conf dd if=/var/lib/ironic/images/e4cf9855-392c-40be-
92b6-9b10674373d0/disk of=/dev/disk/by-path/ip-10.0.0.60:3260-iscsi-iqn.2008-
10.org.openstack:e4cf9855-392c-40be-92b6-9b10674373d0-lun-1-part2 bs=1M oflag=direct
root      6719  6717  2 05:18 ?        00:00:00 /bin/dd
if=/var/lib/ironic/images/e4cf9855-392c-40be-92b6-9b10674373d0/disk of=/dev/disk/by-
path/ip-10.0.0.60:3260-iscsi-iqn.2008-10.org.openstack:e4cf9855-392c-40be-92b6-
9b10674373d0-lun-1-part2 bs=1M oflag=direct
```

During this time, Nova is constantly polling Ironic to check whether the node has successfully deployed the overcloud image. It does this by checking the provision state of the node. Once it's 'active', the deployment has been successful:

```
 [root@undercloud ~]# grep -m 1 "Still waiting for ironic node" /var/log/nova/nova-
compute.log
2016-09-22 09:12:08.287 11893 DEBUG nova.virt.ironic.driver [-] [instance: 9fefdc3e-
0247-4732-bad1-0a87c26a4251] Still waiting for ironic node 875921b3-9864-4aef-8d53-
e6a69f2b1fde to unprovision: power_state="power on", target_power_state=None,
provision_state="deleting", target_provision_state="available" _log_ironic_polling
/usr/lib/python2.7/site-packages/nova/virt/ironic/driver.py:120
```

Overview of the deploy stages trough Ironic node states

When we first run the overcloud deploy command the ironic nodes go from active to deploying state

```
2017-06-08 05:17:13.045 1306 INFO ironic.conductor.task_manager [req-a874331f-e318-
40f3-8136-27acd0b0eb38 2d9df91b629246fbaf68145784156541
a0776ed8e5fb4c3e96647dca76ecfd4b - - -] Node e4cf9855-392c-40be-92b6-9b10674373d0
moved to provision state "deploying" from state "available"; target provision state is
"active"
```

Once the power state of the nodes goes from poweroff to poweron and the iPXE boot starts it changes to wait call-back

```
2017-06-08 05:17:26.283 1306 INFO ironic.conductor.task_manager [req-a874331f-e318-
40f3-8136-27acd0b0eb38 2d9df91b629246fbaf68145784156541
a0776ed8e5fb4c3e96647dca76ecfd4b - - -] Node e4cf9855-392c-40be-92b6-9b10674373d0
moved to provision state "wait call-back" from state "deploying"; target provision
state is "active"
```

When the ramdisk is booted correctly the state changes again to deploying

```
2017-06-08 05:18:04.809 1306 INFO ironic.conductor.task_manager [req-0526985b-6d49-
4684-8d3f-ff446c399ce0 - - - - -] Node e4cf9855-392c-40be-92b6-9b10674373d0 moved to
provision state "deploying" from state "wait call-back"; target provision state is
"active"
```

At this point the ram disk has booted and the iscsi target gets mapped to the undercloud, the dd proccess starts

```
 2017-06-08 05:18:07.979 1306 INFO ironic_lib.disk_utils [req-0526985b-6d49-4684-8d3f-
ff446c399ce0 - - - - -] Disk metadata on /dev/disk/by-path/ip-10.0.0.60:3260-iscsi-
iqn.2008-10.org.openstack:e4cf9855-392c-40be-92b6-9b10674373d0-lun-1 successfully
destroyed for node e4cf9855-392c-40be-92b6-9b10674373d0
2017-06-08 05:18:08.840 1306 INFO ironic_lib.disk_utils [req-0526985b-6d49-4684-8d3f-
ff446c399ce0 - - - - -] Successfully completed the disk device /dev/disk/by-path/ip-
10.0.0.60:3260-iscsi-iqn.2008-10.org.openstack:e4cf9855-392c-40be-92b6-9b10674373d0-
lun-1 partitioning for node e4cf9855-392c-40be-92b6-9b10674373d0
2017-06-08 05:18:09.435 1306 INFO ironic_lib.disk_utils [req-0526985b-6d49-4684-8d3f-
ff446c399ce0 - - - - -] Configdrive for node e4cf9855-392c-40be-92b6-9b10674373d0
successfully copied onto partition /dev/disk/by-path/ip-10.0.0.60:3260-iscsi-iqn.2008-
10.org.openstack:e4cf9855-392c-40be-92b6-9b10674373d0-lun-1-part1
2017-06-08 05:19:37.241 1306 INFO ironic_lib.disk_utils [req-cd605464-c12c-4c58-9b8e-
d33c109bd3ad - - - - -] Image for d8ee5249-a016-444f-bd02-77e422332381 successfully
populated
```

When the dd copy has finished and the overcloud image has been successfully populated, the server goes into active state in ironic, and the deploy phase finishes

```
 2017-06-08 05:19:59.857 1306 INFO ironic.conductor.task_manager [req-0526985b-6d49-
4684-8d3f-ff446c399ce0 - - - - -] Node e4cf9855-392c-40be-92b6-9b10674373d0 moved to
provision state "active" from state "deploying"; target provision state is "None"
2017-06-08 05:19:59.859 1306 INFO ironic.drivers.modules.agent_base_vendor [req-
0526985b-6d49-4684-8d3f-ff446c399ce0 - - - - -] Deployment to node e4cf9855-392c-40be-
92b6-9b10674373d0 done
```

Check root device hints are working

Overcloud Deploy

TODO

Overcloud templates

os-net-config github

The os-net-config github has a sample dir with all the combinations that are supported by os-net-config and you can use those examples directly in your net-config/ yaml templates http://git.openstack.org/cgit/openstack/os-net-config/tree/etc/os-net-config/samples

Overcloud Domain Name in the templates

To specify the domain name for the overcloud nodes we have to add to our templates the CloudDomain: localexample.com , we can also add the fqdn of our vip services/ips, for example:

```
[stack@under templates]$ cat network-environment.yaml | grep -i Cloud
 CloudName: openstack.localexample.com
 CloudNameInternal: overcloud.internalapi.localexample.com
 CloudNameCtlplane: overcloud.ctlplane.localexample.com
 CloudDomain: localexample.com
```

The problem is that currently cloud-init overrides the CloudDomain info, so until there is a final solution as a workaround we have to set the dhcp_domain to match CloudDomain in nova.conf

```
https://bugs.launchpad.net/tripleo/+bug/1581472
```

Mistral Plans Debug

The Mistral log is very detailed and useful for in depth debugging

```
tail -f /var/log/mistral/engine.log | grep "ERROR\|tripleo_common";
```

Checking Mitral with cli to spot Errors

```
 [root@undercloud ~]# mistral execution-list | grep "ERROR";
# Grab the execution ID from above.
[root@undercloud ~]# mistral execution-get $EXECUTION_ID
[root@undercloud ~]# mistral execution-get-output $EXECUTION_ID
# Also look at the actions
[root@undercloud ~]# mistral action-execution-list
[root@undercloud ~]# mistral action-execution-get-output $ACTION_ID
```

Here is and example, we can check the execution-list

```
 [root@undercloud mistral]#  mistral execution-list
+------------------------------+------------------------------
+------------------------------+-----------------------
+------------------------------+---------+------------------------------
+---------------------+---------------------+
| ID                           | Workflow ID                  | Workflow name
| Description            | Task Execution ID            | State   | State info
| Created at          | Updated at          |
+------------------------------+------------------------------
+------------------------------+-----------------------
+------------------------------+---------+------------------------------
+---------------------+---------------------+
| 8aadba6c-a46d-4b87-aa0d-     | 1ab1c5d1-7408-40e3-b02d-     |
tripleo.baremetal.v1.introsp |                         | <none>                       |
SUCCESS | None                         | 2017-06-07 08:18:16 | 2017-06-07 09:19:20 |
| d755981f2836                 | 35367fbb809e                 | ect_manageable_nodes
|                         |                              |         |
|                      |                     |
| 726a96c4-36f7-45ff-9795-b58a | e433d0eb-99cd-450a-a6ae-     |
tripleo.baremetal.v1.introsp | sub-workflow execution | 9830243f-a701-48b1-8fcc-     |
SUCCESS | None                         | 2017-06-07 08:18:18 | 2017-06-07 09:19:18 |
| bcb93f17                     | 3d525401e368                 | ect
|                         | 89c3fa2ac501                 |         |
|                      |                     |
```

And check the output using the Execution ID

```
 [root@undercloud mistral]# mistral execution-get-output 726a96c4-36f7-45ff-9795-
b58abcb93f17
{
    "status": "SUCCESS",
    "message": "Successfully introspected nodes.",
    "introspected_nodes": {
        "ba4db2bf-c327-4fca-a37e-83cc17c1561a": {
            "uuid": "ba4db2bf-c327-4fca-a37e-83cc17c1561a",
```

Manually execute a workflow

We can check all the triple workbooks

```
 [root@undercloud mistral]# mistral workbook-list
+---------------------------+--------+---------------------+------------+
| Name                      | Tags   | Created at          | Updated at |
+---------------------------+--------+---------------------+------------+
| tripleo.baremetal.v1      | <none> | 2017-06-01 13:54:18 | None       |
| tripleo.deployment.v1     | <none> | 2017-06-01 13:54:21 | None       |
| tripleo.package_update.v1 | <none> | 2017-06-01 13:54:23 | None       |
| tripleo.plan_management.v1 | <none> | 2017-06-01 13:54:27 | None       |
| tripleo.scale.v1          | <none> | 2017-06-01 13:54:28 | None       |
| tripleo.stack.v1          | <none> | 2017-06-01 13:54:30 | None       |
| tripleo.validations.v1    | <none> | 2017-06-01 13:54:34 | None       |
+---------------------------+--------+---------------------+------------+
```

With workflow list, we can check all the workflows for a workbook

```
[root@undercloud mistral]#  mistral workflow-list | grep "tripleo.baremetal.v1"  | awk
-F "|" '{ print $3 }'
tripleo.baremetal.v1.introspect_manageable_nodes
tripleo.baremetal.v1.set_node_state
tripleo.baremetal.v1.manage
tripleo.baremetal.v1.create_raid_configuration
tripleo.baremetal.v1.manual_cleaning
tripleo.baremetal.v1.tag_nodes
tripleo.baremetal.v1.provide
tripleo.baremetal.v1.register_or_update
tripleo.baremetal.v1.cellv2_discovery
tripleo.baremetal.v1.configure_manageable_nodes
tripleo.baremetal.v1.tag_node
tripleo.baremetal.v1.set_power_state
tripleo.baremetal.v1.provide_manageable_nodes
tripleo.baremetal.v1.configure
tripleo.baremetal.v1.introspect
```

We can then use the mistral workflow-get command to list the input parameters needed by the

workflow, we are going to use the tripleo.baremetal.v1.provide_manageable_nodes , this simply list ironic nodes in manageable state

```
 [root@undercloud mistral]# mistral workflow-get
tripleo.baremetal.v1.provide_manageable_nodes
+------------+--------------------------------------------+
| Field      | Value                                      |
+------------+--------------------------------------------+
| ID         | b9977a5c-2f04-40cf-9104-ea1733bb0d45       |
| Name       | tripleo.baremetal.v1.provide_manageable_nodes |
| Project ID | 143c2579d1d749308f92033209fc79c5           |
| Tags       | <none>                                     |
| Input      | queue_name=tripleo                         |
| Created at | 2017-06-01 13:54:18                        |
| Updated at | None                                       |
+------------+--------------------------------------------+
```

And you can execute the workflow

```
 [root@undercloud mistral]# mistral execution-create
tripleo.baremetal.v1.provide_manageable_nodes '{"queue_name": "tripleo"}'
+-------------------+--------------------------------------------+
| Field             | Value                                      |
+-------------------+--------------------------------------------+
| ID                | c6212165-fb42-4a83-8074-05496866bb8e       |
| Workflow ID       | b9977a5c-2f04-40cf-9104-ea1733bb0d45       |
| Workflow name     | tripleo.baremetal.v1.provide_manageable_nodes |
| Description       |                                            |
| Task Execution ID | <none>                                     |
| State             | RUNNING                                    |
| State info        | None                                       |
| Created at        | 2017-06-08 21:25:41                        |
| Updated at        | 2017-06-08 21:25:41                        |
+-------------------+--------------------------------------------+
```

With the ID we can see the execution list and get the output

```
 [root@undercloud mistral]# mistral execution-list | grep c6212165-fb42-4a83-8074-
05496866bb8e
| c6212165-fb42-4a83-8074-05496866bb8e | b9977a5c-2f04-40cf-9104-ea1733bb0d45 |
tripleo.baremetal.v1.provide_manageable_nodes           |                       |
<none>                               | SUCCESS | None                        | 2017-
06-08 21:25:41 | 2017-06-08 21:26:05 |
[root@undercloud mistral]# mistral execution-get-output c6212165-fb42-4a83-8074-
05496866bb8e
{
    "managed_nodes": [
        "50fd27b6-7af7-425e-94b2-f6fb0f9f5bfa",
        "ba4db2bf-c327-4fca-a37e-83cc17c1561a"
    ],
    "status": "SUCCESS"
}
```

You can re-run failed mistral tasks using

```
 [root@undercloud mistral]#  mistral task-list;
# Find the ID for the failed task.
[root@undercloud mistral]#  mistral task-rerun $ID;
```

Get your overcloud passwords from mistral

A plan is the combination of a as a Swift container + Mistral environment.

In swift we have the templates we provided:

```
 [root@undercloud mistral]# swift list overcloud | grep net-config
ci/common/net-config-multinode-os-net-config.yaml
ci/common/net-config-multinode.yaml
firstboot/os-net-config-mappings.yaml
net-config-bond.yaml
net-config-bridge.yaml
net-config-linux-bridge.yaml
net-config-noop.yaml
net-config-static-bridge-with-external-dhcp.yaml
net-config-static-bridge.yaml
net-config-static.yaml
net-config-undercloud.yaml
network/scripts/run-os-net-config.sh
```

In the mistral enviroment we can find variables includinf the default passwords:

```
 [root@undercloud mistral]# mistral environment-get overcloud
+-------------
+-----------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------------------
-----------------------------------------------------+
| Field       | Value
|
+-------------
+--------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
-----------------------------------------------------+
| Name        | overcloud
|
| Description | <none>
|
| Variables   | {
|
|             |     "parameter_defaults": {
|
|             |         "ControllerCount": 1,
|
|             |         "NovaComputeLibvirtType": "qemu",
|
|             |         "NtpServer": "95.81.173.74",
|
|             |         "ComputeCount": 1,
|
|             |         "StackAction": "CREATE",
|
|             |         "OvercloudComputeFlavor": "compute",
|
|             |         "OvercloudControlFlavor": "control",
|
|             |         "UpdateIdentifier": ""
|
|             |     },
|
|             |     "passwords": {
|
|             |         "KeystoneFernetKey1": "cbTCTW5nGP-
8u7x9jw6ti9_t65YUsByy8EBdw3XkNao=",
|
|             |         "KeystoneFernetKey0": "xbwAGJ_-Y-
LVbIcH9PlNiCVb4NT_5EyfG0rRoVXOkDM=",
|
|             |         "HAProxyStatsPassword": "EdDx72ybkwDw4b2KM3tkdN7bT",
|
|             |         "HeatPassword": "dUR9C9Qy4eBxgD82usZhvGFsD",
|
|             |         "KeystoneCredential1": "h7AQ-
HYvOFzpU7Lxb2bFywYkEkK3l9dJDRXFw1Kl7xI=",
|
|             |         "CongressPassword": "pKNBHNWrxsRpjVpRJYgcjMwgp",
|
```

```
|             |         "NeutronPassword": "YPDgACsfmN4uGHcnzebc3rEJv",
|
|             |         "SnmpdReadonlyUserPassword":
"bcc0f245d947ea6b62eea3aa63bdbfb3380ff0b5",
|
|             |         "GlancePassword": "Nwy9PbgccFv7K2AFpjfQs9tG8",
|
|             |         "AdminPassword": "YjrqYfGvhfpmatkzpHD9R9myj",
|
|             |         "IronicPassword": "KhKes3scq9tEd9B3GPf37jRTr",
|
|             |         "HeatStackDomainAdminPassword": "VkE48BwexxcCZXM9DR4putAPT",
|
|             |         "ZaqarPassword": "F86nYgsqmHbedfMfReUJXpVra",
|
```

Mistral By Example: Full overcloud install using Mistral

http://tripleo.org/mistral-api/mistral-api.html

Overcloud Deploy Debug

Follow the error in heat

Following an error on the overcloud stack list, we first check that the overcloud stack has failed

```
 [stack@undercloud ~]$ heat stack-list
WARNING (shell) "heat stack-list" is deprecated, please use "openstack stack list"
instead
+--------------------------------------+------------+--------------
+----------------------+--------------+
| id                                   | stack_name | stack_status  | creation_time
| updated_time |
+--------------------------------------+------------+--------------
+----------------------+--------------+
| a7681ece-034d-4fba-8874-7803601b9507 | overcloud  | CREATE_FAILED | 2017-06-
12T13:34:13Z | None         |
+--------------------------------------+------------+--------------
+----------------------+--------------+
```

We can then list the resources of the stack, the compute resource group has failed

```
 [stack@undercloud ~]$  openstack stack resource list overcloud | grep -i Failed
| Compute                                | 6f30d573-685b-468d-b225-fdba504f2533
| OS::Heat::ResourceGroup                      | CREATE_FAILED   | 2017-06-
12T13:34:13Z |
```

We can dig in deeper, going into the nested stacks, we can see that the node that has failed is

compute node 0

```
 [stack@undercloud ~]$ openstack stack resource list 6f30d573-685b-468d-b225-
fdba504f2533 | grep FAILED
| 0             | f9c1c76e-f7a7-4770-9bd0-a4c49e480d84 | OS::TripleO::Compute |
CREATE_FAILED   | 2017-06-12T13:34:53Z |
```

If we check in the compute 0 stack, we can see we have to failed SoftwareDeployment resources

```
 [stack@undercloud ~]$ openstack stack resource list  f9c1c76e-f7a7-4770-9bd0-
a4c49e480d84 | grep FAILED
| NetworkDeployment     | 3fcbbf55-1238-4d36-9539-53e40ef2e136 |
OS::TripleO::SoftwareDeployment              | CREATE_FAILED   | 2017-06-12T13:34:54Z
|
| UpdateDeployment      | d7fc1305-9b55-4afb-8612-db2917b11505 |
OS::Heat::SoftwareDeployment                 | CREATE_FAILED   | 2017-06-12T13:34:54Z
|
```

We can get this info in 1 command using the nested depth option on the cli

```
 [stack@undercloud ~]$ openstack stack resource list -n5 overcloud | grep FAILED
| Compute                                  | 6f30d573-685b-468d-b225-fdba504f2533
| OS::Heat::ResourceGroup
| CREATE_FAILED   | 2017-06-12T13:34:13Z | overcloud
|
| 0                                        | f9c1c76e-f7a7-4770-9bd0-a4c49e480d84
| OS::TripleO::Compute
| CREATE_FAILED   | 2017-06-12T13:34:53Z | overcloud-Compute-f7gr76kpxqaz
|
| UpdateDeployment                         | d7fc1305-9b55-4afb-8612-db2917b11505
| OS::Heat::SoftwareDeployment
| CREATE_FAILED   | 2017-06-12T13:34:54Z | overcloud-Compute-f7gr76kpxqaz-0-
h42tisowbbuu                                                                    |
| NetworkDeployment                        | 3fcbbf55-1238-4d36-9539-53e40ef2e136
| OS::TripleO::SoftwareDeployment
| CREATE_FAILED   | 2017-06-12T13:34:54Z | overcloud-Compute-f7gr76kpxqaz-0-
h42tisowbbuu                                                                  |
```

Once we know where the error is, we can use resource show to check the attributes

```
 [stack@undercloud ~]$ openstack stack resource list overcloud | grep -i failed
| Compute                                | 6f30d573-685b-468d-b225-fdba504f2533
| OS::Heat::ResourceGroup                      | CREATE_FAILED   | 2017-06-
12T13:34:13Z |
[stack@undercloud ~]$ openstack stack resource list 6f30d573-685b-468d-b225-
fdba504f2533
+---------------+--------------------------------------+---------------------
+-----------------+----------------------+
| resource_name | physical_resource_id                 | resource_type        |
resource_status | updated_time         |
+---------------+--------------------------------------+---------------------
+-----------------+----------------------+
| 0             | f9c1c76e-f7a7-4770-9bd0-a4c49e480d84 | OS::TripleO::Compute |
CREATE_FAILED   | 2017-06-12T13:34:53Z |
+---------------+--------------------------------------+---------------------
+-----------------+----------------------+
```

With resource show we have to specify the stack(can be nested) and the resource name, in this example the stack id is the compute resource-id that we get from the previous command

```
 [stack@undercloud ~]$ openstack stack resource show 6f30d573-685b-468d-b225-
fdba504f2533 0
+------------------------
+-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
----------------------------------------+
| Field                  | Value
|
+------------------------
+-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
----------------------------------------+
| attributes             | {u'storage_mgmt_ip_address': u'10.0.0.59',
u'nova_server_resource': u'c05fc792-3807-4bd0-becd-d7fc5035474b', u'hostname':
u'overcloud-compute-0', u'tenant_ip_address': u'192.168.3.154',
u'external_ip_address':  |
|                        | u'10.0.0.59', u'known_hosts_entry':
u'192.168.2.154,overcloud-compute-0.liquid.zz,overcloud-compute-0,10.0.0.59,overcloud-
compute-0.external.liquid.zz,overcloud-compute-0.external,192.168.2.154,overcloud-
|
|                        | compute-0.internalapi.liquid.zz,overcloud-compute-
0.internalapi,172.16.1.12,overcloud-compute-0.storage.liquid.zz,overcloud-compute-
0.storage,10.0.0.59,overcloud-compute-0.storagemgmt.liquid.zz,overcloud-       |
|                        | compute-0.storagemgmt,192.168.3.154,overcloud-compute-
0.tenant.liquid.zz,overcloud-compute-0.tenant,10.0.0.59,overcloud-compute-
0.management.liquid.zz,overcloud-compute-0.management,10.0.0.59,overcloud-         |
|                        | compute-0.ctlplane.liquid.zz,overcloud-compute-0.ctlplane
', u'hosts_entry': u'192.168.2.154 overcloud-compute-0.liquid.zz overcloud-compute-
0\n10.0.0.59 overcloud-compute-0.external.liquid.zz overcloud-        |
```

```
|                        | compute-0.external\n192.168.2.154 overcloud-compute-
0.internalapi.liquid.zz overcloud-compute-0.internalapi\n172.16.1.12 overcloud-
compute-0.storage.liquid.zz overcloud-compute-0.storage\n10.0.0.59 overcloud-   |
|                        | compute-0.storagemgmt.liquid.zz overcloud-compute-
0.storagemgmt\n192.168.3.154 overcloud-compute-0.tenant.liquid.zz overcloud-compute-
0.tenant\n10.0.0.59 overcloud-compute-0.management.liquid.zz overcloud-      |
|                        | compute-0.management\n10.0.0.59 overcloud-compute-
0.ctlplane.liquid.zz overcloud-compute-0.ctlplane\n', u'storage_ip_address':
u'172.16.1.12', u'hostname_map': {u'management': u'overcloud-                      |
|                        | compute-0.management.liquid.zz', u'storage': u'overcloud-
compute-0.storage.liquid.zz', u'ctlplane': u'overcloud-compute-0.ctlplane.liquid.zz',
u'external': u'overcloud-compute-0.external.liquid.zz',             |
|                        | u'internal_api': u'overcloud-compute-
0.internalapi.liquid.zz', u'storage_mgmt': u'overcloud-compute-
0.storagemgmt.liquid.zz', u'tenant': u'overcloud-compute-0.tenant.liquid.zz'},
u'internal_api_ip_address':     |
|                        | u'192.168.2.154', u'ip_address': u'10.0.0.59',
u'management_ip_address': u'10.0.0.59'}
|
| creation_time          | 2017-06-12T13:34:53Z
|
| description            |
|
| links                  | [{u'href':
u'http://10.0.0.10:8004/v1/e6128b61cf7a4169a8f61cdf41a27344/stacks/overcloud-Compute-
f7gr76kpxqaz/6f30d573-685b-468d-b225-fdba504f2533/resources/0', u'rel': u'self'},
{u'href':                          |
|                        |
u'http://10.0.0.10:8004/v1/e6128b61cf7a4169a8f61cdf41a27344/stacks/overcloud-Compute-
f7gr76kpxqaz/6f30d573-685b-468d-b225-fdba504f2533', u'rel': u'stack'}, {u'href':
|
|                        |
u'http://10.0.0.10:8004/v1/e6128b61cf7a4169a8f61cdf41a27344/stacks/overcloud-Compute-
f7gr76kpxqaz-0-h42tisowbbuu/f9c1c76e-f7a7-4770-9bd0-a4c49e480d84', u'rel': u'nested'}]
|
| logical_resource_id    | 0
|
| parent_resource        | Compute
|
| physical_resource_id   | f9c1c76e-f7a7-4770-9bd0-a4c49e480d84
|
| required_by            | []
|
| resource_name          | 0
|
| resource_status        | CREATE_FAILED
|
| resource_status_reason | CREATE aborted
|
| resource_type          | OS::TripleO::Compute
|
```

```
| updated_time           | 2017-06-12T13:34:53Z
|
+------------------------
+-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
--------------------------------------+
```

Looking at the attributes of the previous command we can see something strange is going on, we have several ip definitions with the same IP: storage_mgmt_ip_address': u'10.0.0.59, external_ip_address: u'10.0.0.59'

We can still try and get more info from the resources in the last stack:

```
 [stack@undercloud ~]$ openstack stack resource list f9c1c76e-f7a7-4770-9bd0-
a4c49e480d84 | grep FAILED
+----------------------+-------------------------------------
+-----------------------------------------+-----------------
+----------------------+
| resource_name         | physical_resource_id                | resource_type
| resource_status | updated_time         |
+----------------------+-------------------------------------
+-----------------------------------------+-----------------
+----------------------+
| NetworkDeployment     | 3fcbbf55-1238-4d36-9539-53e40ef2e136 |
OS::TripleO::SoftwareDeployment              | CREATE_FAILED   | 2017-06-12T13:34:54Z
|
| UpdateDeployment      | d7fc1305-9b55-4afb-8612-db2917b11505 |
OS::Heat::SoftwareDeployment                 | CREATE_FAILED   | 2017-06-12T13:34:54Z
|
+----------------------+-------------------------------------
+-----------------------------------------+-----------------
+----------------------+
[stack@undercloud ~]$ openstack stack resource show f9c1c76e-f7a7-4770-9bd0-
a4c49e480d84 NetworkDeployment
+------------------------
+-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
--------------------------------------+
| Field                  | Value
|
+------------------------
+-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
--------------------------------------+
| attributes             | {u'deploy_stdout': None, u'deploy_stderr': None,
u'deploy_status_code': None}
|
| creation_time          | 2017-06-12T13:34:54Z
|
| description            |
```

```
|
| links                  | [{u'href':
u'http://10.0.0.10:8004/v1/e6128b61cf7a4169a8f61cdf41a27344/stacks/overcloud-Compute-
f7gr76kpxqaz-0-h42tisowbbuu/f9c1c76e-f7a7-4770-9bd0-
a4c49e480d84/resources/NetworkDeployment', u'rel': u'self'},   |
|                        | {u'href':
u'http://10.0.0.10:8004/v1/e6128b61cf7a4169a8f61cdf41a27344/stacks/overcloud-Compute-
f7gr76kpxqaz-0-h42tisowbbuu/f9c1c76e-f7a7-4770-9bd0-a4c49e480d84', u'rel': u'stack'}]
|
| logical_resource_id    | NetworkDeployment
|
| parent_resource        | 0
|
| physical_resource_id   | 3fcbbf55-1238-4d36-9539-53e40ef2e136
|
| required_by            | [u'NovaComputeDeployment']
|
| resource_name          | NetworkDeployment
|
| resource_status        | CREATE_FAILED
|
| resource_status_reason | CREATE aborted
|
| resource_type          | OS::TripleO::SoftwareDeployment
|
| updated_time           | 2017-06-12T13:34:54Z
|
+------------------------
+-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
----------------------------------------+
[stack@undercloud ~]$ openstack stack resource show f9c1c76e-f7a7-4770-9bd0-
a4c49e480d84 UpdateDeployment
+------------------------
+-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
----------------------------------------+
| Field                  | Value
|
+------------------------
+-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
----------------------------------------+
| attributes             | {u'deploy_stdout': None, u'deploy_stderr': None,
u'deploy_status_code': None}
|
| creation_time          | 2017-06-12T13:34:54Z
|
| description            |
|
| links                  | [{u'href':
```

```
u'http://10.0.0.10:8004/v1/e6128b61cf7a4169a8f61cdf41a27344/stacks/overcloud-Compute-
f7gr76kpxqaz-0-h42tisowbbuu/f9c1c76e-f7a7-4770-9bd0-
a4c49e480d84/resources/UpdateDeployment', u'rel': u'self'},    |
|                          | {u'href':
u'http://10.0.0.10:8004/v1/e6128b61cf7a4169a8f61cdf41a27344/stacks/overcloud-Compute-
f7gr76kpxqaz-0-h42tisowbbuu/f9c1c76e-f7a7-4770-9bd0-a4c49e480d84', u'rel': u'stack'}]
|
| logical_resource_id    | UpdateDeployment
|
| parent_resource        | 0
|
| physical_resource_id   | d7fc1305-9b55-4afb-8612-db2917b11505
|
| required_by            | [u'NovaComputeDeployment']
|
| resource_name          | UpdateDeployment
|
| resource_status        | CREATE_FAILED
|
| resource_status_reason | CREATE aborted
|
| resource_type          | OS::Heat::SoftwareDeployment
|
| updated_time           | 2017-06-12T13:34:54Z
|
```

We can also use the deploy show to get output of the failed command, here we can see as input values the configuration of the br-ex bridge with nic1

```
 [stack@undercloud ~]$ heat resource-list -n5 overcloud | grep SoftwareDeployment |
grep CREATE_FAILED
WARNING (shell) "heat resource-list" is deprecated, please use "openstack stack
resource list" instead
| NetworkDeployment                         | 3fcbbf55-1238-4d36-9539-53e40ef2e136
| OS::TripleO::SoftwareDeployment
| CREATE_FAILED   | 2017-06-12T13:34:54Z | overcloud-Compute-f7gr76kpxqaz-0-
h42tisowbbuu                                                                    |
| UpdateDeployment                          | d7fc1305-9b55-4afb-8612-db2917b11505
| OS::Heat::SoftwareDeployment
| CREATE_FAILED   | 2017-06-12T13:34:54Z | overcloud-Compute-f7gr76kpxqaz-0-
h42tisowbbuu                                                                    |
[stack@undercloud ~]$ heat deployment-show 3fcbbf55-1238-4d36-9539-53e40ef2e136
WARNING (shell) "heat deployment-show" is deprecated, please use "openstack software
deployment show" instead
{
  "status": "IN_PROGRESS",
  "server_id": "c05fc792-3807-4bd0-becd-d7fc5035474b",
  "config_id": "43d20c1e-94c4-41de-b657-b522edbb5c0c",
  "output_values": null,
  "creation_time": "2017-06-12T13:40:14Z",
  "input_values": {
    "interface_name": "nic1",
    "bridge_name": "br-ex"
  },
  "action": "CREATE",
  "status_reason": "Deploy data available",
  "id": "3fcbbf55-1238-4d36-9539-53e40ef2e136"
}
[stack@undercloud ~]$ heat deployment-show d7fc1305-9b55-4afb-8612-db2917b11505
WARNING (shell) "heat deployment-show" is deprecated, please use "openstack software
deployment show" instead
{
  "status": "IN_PROGRESS",
  "server_id": "c05fc792-3807-4bd0-becd-d7fc5035474b",
  "config_id": "d0134c5e-3aae-4a6a-98f4-5c577a41d24c",
  "output_values": null,
  "creation_time": "2017-06-12T13:40:08Z",
  "input_values": {
    "update_identifier": ""
  },
  "action": "CREATE",
  "status_reason": "Deploy data available",
  "id": "d7fc1305-9b55-4afb-8612-db2917b11505"
}
```

To get more details on these deployments we can use the heat config-list/config-show commands

We first get the deployment id:

```
 [stack@undercloud ~]$ heat resource-list -n5 overcloud | grep SoftwareDeployment |
grep CREATE_FAILED
WARNING (shell) "heat resource-list" is deprecated, please use "openstack stack
resource list" instead
| NetworkDeployment                          | 3fcbbf55-1238-4d36-9539-53e40ef2e136
| OS::TripleO::SoftwareDeployment
| CREATE_FAILED   | 2017-06-12T13:34:54Z | overcloud-Compute-f7gr76kpxqaz-0-
h42tisowbbuu                                                                    |
| UpdateDeployment                           | d7fc1305-9b55-4afb-8612-db2917b11505
| OS::Heat::SoftwareDeployment
| CREATE_FAILED   | 2017-06-12T13:34:54Z | overcloud-Compute-f7gr76kpxqaz-0-
h42tisowbbuu                                                                    |
```

With the deployment id, we grep it to get the config id

```
 [stack@undercloud ~]$  heat deployment-list | grep 3fcbbf55-1238-4d36-9539-
53e40ef2e136
WARNING (shell) "heat deployment-list" is deprecated, please use "openstack software
deployment list" instead
| 3fcbbf55-1238-4d36-9539-53e40ef2e136 | 43d20c1e-94c4-41de-b657-b522edbb5c0c |
c05fc792-3807-4bd0-becd-d7fc5035474b | CREATE | IN_PROGRESS | 2017-06-12T13:40:14Z |
Deploy data available |
```

The second column is the config id, once we have it we can do a config show and see what inputs have been passed to os-net-config on the compute 0 host:

```
 [stack@undercloud ~]$ heat config-show 43d20c1e-94c4-41de-b657-b522edbb5c0c
WARNING (shell) "heat config-show" is deprecated, please use "openstack software
config show" instead
{
  "inputs": [
    {
      "type": "String",
      "name": "interface_name",
      "value": "nic1"
    },
    {
      "type": "String",
      "name": "bridge_name",
      "value": "br-ex"
    },
    {
      "type": "String",
      "name": "deploy_server_id",
      "value": "c05fc792-3807-4bd0-becd-d7fc5035474b",
      "description": "ID of the server being deployed to"
    },
    {
```

```
      "type": "String",
      "name": "deploy_action",
      "value": "CREATE",
      "description": "Name of the current action being deployed"
    },
    {
      "type": "String",
      "name": "deploy_stack_id",
      "value": "overcloud-Compute-f7gr76kpxqaz-0-h42tisowbbuu/f9c1c76e-f7a7-4770-9bd0-
a4c49e480d84",
      "description": "ID of the stack this deployment belongs to"
    },
    {
      "type": "String",
      "name": "deploy_resource_name",
      "value": "NetworkDeployment",
      "description": "Name of this deployment resource in the stack"
    },
    {
      "type": "String",
      "name": "deploy_signal_transport",
      "value": "CFN_SIGNAL",
      "description": "How the server should signal to heat with the deployment output
values."
    },
    {
      "type": "String",
      "name": "deploy_signal_id",
      "value":
"http://10.0.0.10:8000/v1/signal/arn%3Aopenstack%3Aheat%3A%3Ae6128b61cf7a4169a8f61cdf4
1a27344%3Astacks%2Fovercloud-Compute-f7gr76kpxqaz-0-h42tisowbbuu%2Ff9c1c76e-f7a7-4770-
9bd0-a4c49e480d84%2Fresources%2FNetworkDeployment?Timestamp=2017-06-
12T13%3A34%3A54Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=38be6b4a84d24b429a6b6fcf434
535e5&SignatureVersion=2&Signature=XBnVw%2BUhMBj56vdkjJcNfV5kjylCkJLQ%2FGHec61iLDQ%3D"
,
      "description": "ID of signal to use for signaling output values"
    },
    {
      "type": "String",
      "name": "deploy_signal_verb",
      "value": "POST",
      "description": "HTTP verb to use for signaling outputvalues"
    }
  ],
  "group": "os-apply-config",
  "name": "NetworkDeployment",
  "outputs": [],
  "creation_time": "2017-06-12T13:40:13Z",
  "options": {},
  "config": {
    "os_net_config": {
```

```
      "network_config": [
        {
          "addresses": [
            {
              "ip_netmask": "10.0.0.59/24"
            }
          ],
          "bonding_options": "mode=1 miimon=150",
          "members": [
            {
              "type": "interface",
              "name": "eth0",
              "primary": true
            },
            {
              "type": "interface",
              "name": "eth1"
            }
          ],
          "routes": [
            {
              "ip_netmask": "169.254.169.254/32",
              "next_hop": "10.0.0.10"
            }
          ],
          "use_dhcp": false,
          "type": "linux_bond",
          "name": "bond0"
        },
        {
          "dns_servers": [
            "192.168.101.1",
            "8.8.8.8"
          ],
          "type": "ovs_bridge",
          "name": "br-tenant",
          "members": [
            {
              "type": "linux_bond",
              "bonding_options": "mode=1 miimon=150",
              "members": [
                {
                  "type": "interface",
                  "name": "eth2",
                  "primary": true
                },
                {
                  "type": "interface",
                  "name": "eth3"
                }
              ],
```

```
            "name": "bond1"
          },
          {
            "device": "bond1",
            "use_dhcp": false,
            "type": "vlan",
            "addresses": [
              {
                "ip_netmask": "192.168.3.154/24"
              }
            ],
            "vlan_id": 300
          },
          {
            "addresses": [
              {
                "ip_netmask": "10.0.0.59/24"
              }
            ],
            "routes": [
              {
                "default": true,
                "next_hop": "192.168.101.1"
              }
            ],
            "device": "bond1",
            "use_dhcp": false,
            "type": "vlan",
            "vlan_id": 101
          }
        ]
      },
      {
        "dns_servers": [
          "192.168.101.1",
          "8.8.8.8"
        ],
        "type": "ovs_bridge",
        "name": "br-api",
        "members": [
          {
            "use_dhcp": false,
            "type": "linux_bond",
            "bonding_options": "mode=1 miimon=150",
            "members": [
              {
                "type": "interface",
                "name": "eth4",
                "primary": true
              },
              {
```

```
              "type": "interface",
              "name": "eth5"
            }
          ],
          "name": "bond2"
        },
        {
          "device": "bond2",
          "use_dhcp": false,
          "type": "vlan",
          "addresses": [
            {
              "ip_netmask": "192.168.2.154/24"
            }
          ],
          "vlan_id": 200
        }
      ]
    },
    {
      "dns_servers": [
        "192.168.101.1",
        "8.8.8.8"
      ],
      "type": "ovs_bridge",
      "name": "br-storage",
      "members": [
        {
          "type": "linux_bond",
          "bonding_options": "mode=1 miimon=150",
          "members": [
            {
              "type": "interface",
              "name": "eth6",
              "primary": true
            },
            {
              "type": "interface",
              "name": "eth7"
            }
          ],
          "name": "bond3"
        },
        {
          "device": "bond3",
          "use_dhcp": false,
          "type": "vlan",
          "addresses": [
            {
              "ip_netmask": "10.0.0.59/24"
            }
```

```
              ],
              "vlan_id": 100
            }
          ]
        }
      ]
    }
  },
  "id": "43d20c1e-94c4-41de-b657-b522edbb5c0c"
}
```

Checking Errors from the Overcloud Node

Once the nodes are deployed we can access them via ILO if no network has been configured, or via SSH using the heat-admin user from the undercloud node. To be able to access the overcloud node via the ILO we need to have previously modified the overcloud image to add a root passwd

```
#virt-customize -a overcloud-full.qcow2 --root-password password:test
```

In this example we check the stack resource list and see that the controller nodes have been stuck a long time in network deployment

```
 2017-06-13 14:38:05Z [overcloud.Controller.0.NetworkDeployment]: CREATE_IN_PROGRESS
state changed
2017-06-13 14:38:06Z [overcloud.Controller.1.NetworkDeployment]: CREATE_IN_PROGRESS
state changed
2017-06-13 14:38:07Z [overcloud.Controller.2.NetworkDeployment]: CREATE_IN_PROGRESS
state changed
```

We are going to connect via ipmi to check whats happening.

```
 root@overcloud-controller-0 log]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether aa:bb:cc:dd:ee:01 brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 52:54:00:7a:67:df brd ff:ff:ff:ff:ff:ff
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 52:54:00:8f:3d:8a brd ff:ff:ff:ff:ff:ff
5: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 52:54:00:d3:8f:e4 brd ff:ff:ff:ff:ff:ff
6: eth4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 52:54:00:4c:cc:8d brd ff:ff:ff:ff:ff:ff
7: eth5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 52:54:00:14:6e:eb brd ff:ff:ff:ff:ff:ff
8: eth6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 52:54:00:c6:09:20 brd ff:ff:ff:ff:ff:ff
9: eth7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 52:54:00:45:64:c7 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::5054:ff:fe45:64c7/64 scope link
       valid_lft forever preferred_lft forever
```

Network is not configured, lets check if the needed processes are running:

```
 [root@overcloud-controller-0 os-net-config]# ps ax | grep -e os- -e heat
ps ax | grep -e os- -e heat
 2966 ?        Ss     0:01 /usr/bin/python /usr/bin/os-collect-config
11155 ttyS0    S+     0:00 grep --color=auto -e os- -e heat
```

We can see that os-collect-config is running, but we don't have the heat client, nor the os-refresh-config process.

Also /var/lib/heat-config dir is empty, so no config files have been copied from the undercloud

```
 [root@overcloud-controller-0 os-net-config]# ls /var/lib/heat-config
deployed
```

We can check the system logs and see, that we can't access the metadata server:

```
Jun 13 10:44:44 overcloud-controller-0 os-collect-config:
HTTPConnectionPool(host='169.254.169.254', port=80): Max retries exceeded with url:
/latest/meta-data/ (Caused by
NewConnectionError('<requests.packages.urllib3.connection.HTTPConnection object at
0x3acfbd0>: Failed to establish a new connection: [Errno 101] Network is
unreachable',))
```

If we check for os-net-config in the messages file, we can look for errors in the network config

```
 [root@overcloud-controller-1 ~]# cat /var/log/messages | grep -A 80 "os-net-config
-c"
ig -c"ar/log/messages | grep -A 80 "os-net-conf
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: + os-net-config -c /etc/os-
net-config/config.json -v --detailed-exit-codes
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] Using config file at: /etc/os-net-config/config.json
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] Using mapping file at: /etc/os-net-config/mapping.yaml
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] Ifcfg net config provider created.
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] nic8 mapped to: eth7
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] nic7 mapped to: eth6
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] nic6 mapped to: eth5
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] nic5 mapped to: eth4
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] nic4 mapped to: eth3
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] nic3 mapped to: eth2
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] nic2 mapped to: eth1
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] nic1 mapped to: eth0
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding linux bond: bond0
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding custom route for interface: bond0
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding interface: eth0
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
```

```
[INFO] adding interface: eth1
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding bridge: br-tenant
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding interface: eth2
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding vlan: vlan300
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding bridge: br-api
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding linux bond: bond2
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding interface: eth4
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding interface: eth5
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding vlan: vlan200
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding bridge: br-storage
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding linux bond: bond3
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding interface: eth6
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding interface: eth7
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: [2017/06/13 10:38:09 AM]
[INFO] adding vlan: vlan100
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: Traceback (most recent call
last):
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: File "/usr/bin/os-net-
config", line 10, in <module>
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: sys.exit(main())
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: File
"/usr/lib/python2.7/site-packages/os_net_config/cli.py", line 184, in main
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: obj =
objects.object_from_json(iface_json)
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: File
"/usr/lib/python2.7/site-packages/os_net_config/objects.py", line 38, in
object_from_json
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: return
Interface.from_json(json)
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: File
"/usr/lib/python2.7/site-packages/os_net_config/objects.py", line 299, in from_json
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: opts =
_BaseOpts.base_opts_from_json(json)
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: File
"/usr/lib/python2.7/site-packages/os_net_config/objects.py", line 253, in
base_opts_from_json
Jun 13 10:38:09 overcloud-controller-1 os-collect-config:
addresses.append(Address.from_json(address))
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: File
```

```
"/usr/lib/python2.7/site-packages/os_net_config/objects.py", line 171, in from_json
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: ip_netmask =
_get_required_field(json, 'ip_netmask', 'Address')
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: File
"/usr/lib/python2.7/site-packages/os_net_config/objects.py", line 78, in
_get_required_field
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: raise
InvalidConfigException(msg)
Jun 13 10:38:09 overcloud-controller-1 os-collect-config:
os_net_config.objects.InvalidConfigException: Address JSON objects require
'ip_netmask' to be configured.
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: + RETVAL=1
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: + [[ 1 == 2 ]]
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: + [[ 1 != 0 ]]
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: + echo 'ERROR: os-net-config
configuration failed.'
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: ERROR: os-net-config
configuration failed.
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: + exit 1
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: + configure_safe_defaults
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: + [[ 1 == 0 ]]
Jun 13 10:38:09 overcloud-controller-1 os-collect-config: + cat
```

We can see that os-net-config says there is no ip_netmask configure for vlan100, we can take at the config file being user by os-net-config

```
 [root@overcloud-controller-1 ~]# cat /etc/os-net-config/config.json | jq .
cat /etc/o s-net-config/config.json | jq .
{
  "network_config": [
    {
      "name": "bond0",
      "type": "linux_bond",
      "use_dhcp": false,
      "routes": [
        {
          "next_hop": "10.0.0.10",
          "ip_netmask": "169.254.169.254/32"
        }
      ],
      "members": [
        {
          "primary": true,
          "name": "eth0",
          "type": "interface"
        },
        {
          "name": "eth1",
          "type": "interface"
        }
```

```
      ],
      "bonding_options": "mode=1 miimon=150",
      "addresses": [
        {
          "ip_netmask": "10.0.0.58/24"
        }
      ]
    },
    {
      "members": [
        {
          "primary": true,
          "name": "eth2",
          "type": "interface"
        },
        {
          "vlan_id": 300,
          "addresses": [
            {
              "ip_netmask": "192.168.3.151/24"
            }
          ],
          "type": "vlan",
          "use_dhcp": false
        }
      ],
      "name": "br-tenant",
      "type": "ovs_bridge",
      "dns_servers": [
        "192.168.101.1",
        "8.8.8.8"
      ]
    },
    {
      "members": [
        {
          "name": "bond2",
          "members": [
            {
              "primary": true,
              "name": "eth4",
              "type": "interface"
            },
            {
              "name": "eth5",
              "type": "interface"
            }
          ],
          "bonding_options": "mode=1 miimon=150",
          "type": "linux_bond"
        },
```

```
        {
          "vlan_id": 200,
          "addresses": [
            {
              "ip_netmask": "192.168.2.151/24"
            }
          ],
          "type": "vlan",
          "device": "bond2"
        }
      ],
      "name": "br-api",
      "type": "ovs_bridge",
      "dns_servers": [
        "192.168.101.1",
        "8.8.8.8"
      ]
    },
    {
      "members": [
        {
          "name": "bond3",
          "members": [
            {
              "primary": true,
              "name": "eth6",
              "type": "interface"
            },
            {
              "name": "eth7",
              "type": "interface"
            }
          ],
          "bonding_options": "mode=1 miimon=150",
          "type": "linux_bond"
        },
        {
          "vlan_id": 100,
          "addresses": [
            {
              "ip_netmask": "192.168.1.151/24"
            }
          ],
          "type": "vlan",
          "device": "bond3"
        }
      ],
      "name": "br-storage",
      "type": "ovs_bridge",
      "dns_servers": [
        "192.168.101.1",
```

```
        "8.8.8.8"
      ]
    },
    {
      "name": "eth3",
      "vlan_id": 101,
      "addresses": [
        {
          "ip_netmask": "192.168.101.151/24",
          "routes": null
        },
        {
          "next_hop": "192.168.101.1",
          "default": true
        }
      ],
      "type": "interface",
      "use_dhcp": false
    }
  ]
}
```

Looks like something is wrong with eth3, the first problem is that the eth3 device doesn't have vlans, is a access port, to do tests for a valid config we can modify the file in /etc/os-net-config

```
 root@overcloud-controller-1 os-net-config]# pwd
/etc/os-net-config
[root@overcloud-controller-1 os-net-config]# ls
config.json  dhcp_all_interfaces.yaml
```

We stop the collect service, so it doesn't interfere with the manual execution

```
 [root@overcloud-controller-1 os-net-config]# systemctl stop os-collect-config.service
rvicemctl stop os-collect-config.se
```

I created another filed called config.json and remvoved the eth3 configuration and run the os-net-config command, and all works fine

```
 [root@overcloud-controller-1 os-net-config]# os-net-config -c /etc/os-net-
config/config.json.test -v --detailed-exit-codes
/config.json.test -v --detailed-exit-codes
[2017/06/13 03:37:27 PM] [INFO] Using config file at: /etc/os-net-
config/config.json.test
[2017/06/13 03:37:27 PM] [INFO] Using mapping file at: /etc/os-net-config/mapping.yaml
[2017/06/13 03:37:27 PM] [INFO] Ifcfg net config provider created.
[2017/06/13 03:37:27 PM] [INFO] nic2 mapped to: eth7
[2017/06/13 03:37:27 PM] [INFO] nic1 mapped to: eth3
```

```
[2017/06/13 03:37:27 PM] [INFO] adding linux bond: bond0
[2017/06/13 03:37:27 PM] [INFO] adding custom route for interface: bond0
[2017/06/13 03:37:27 PM] [INFO] adding interface: eth0
[2017/06/13 03:37:27 PM] [INFO] adding interface: eth1
[2017/06/13 03:37:27 PM] [INFO] adding bridge: br-tenant
[2017/06/13 03:37:27 PM] [INFO] adding interface: eth2
[2017/06/13 03:37:27 PM] [INFO] adding vlan: vlan300
[2017/06/13 03:37:27 PM] [INFO] adding bridge: br-api
[2017/06/13 03:37:27 PM] [INFO] adding linux bond: bond2
[2017/06/13 03:37:27 PM] [INFO] adding interface: eth4
[2017/06/13 03:37:27 PM] [INFO] adding interface: eth5
[2017/06/13 03:37:27 PM] [INFO] adding vlan: vlan200
[2017/06/13 03:37:27 PM] [INFO] adding bridge: br-storage
[2017/06/13 03:37:27 PM] [INFO] adding linux bond: bond3
[2017/06/13 03:37:27 PM] [INFO] adding interface: eth6
[2017/06/13 03:37:27 PM] [INFO] adding interface: eth7
[2017/06/13 03:37:27 PM] [INFO] adding vlan: vlan100
[2017/06/13 03:37:27 PM] [INFO] applying network configs...
[2017/06/13 03:37:31 PM] [INFO] running ifup on bridge: br-tenant
[2017/06/13 03:37:31 PM] [INFO] running ifup on bridge: br-storage
[2017/06/13 03:37:32 PM] [INFO] running ifup on bridge: br-api
[2017/06/13 03:37:32 PM] [INFO] running ifup on interface: eth7
[2017/06/13 03:37:32 PM] [INFO] running ifup on interface: eth6
[2017/06/13 03:37:32 PM] [INFO] running ifup on interface: eth5
[2017/06/13 03:37:32 PM] [INFO] running ifup on interface: eth4
[2017/06/13 03:37:32 PM] [INFO] running ifup on interface: eth2
[2017/06/13 03:37:33 PM] [INFO] running ifup on interface: eth1
[2017/06/13 03:37:33 PM] [INFO] running ifup on interface: eth0
[2017/06/13 03:37:33 PM] [INFO] running ifup on interface: vlan300
[2017/06/13 03:37:38 PM] [INFO] running ifup on interface: vlan100
[2017/06/13 03:37:42 PM] [INFO] running ifup on interface: bond3
[2017/06/13 03:37:43 PM] [INFO] running ifup on interface: vlan200
[2017/06/13 03:37:47 PM] [INFO] running ifup on interface: bond2
[2017/06/13 03:37:48 PM] [INFO] running ifup on interface: eth1
[2017/06/13 03:37:48 PM] [INFO] running ifup on interface: eth0
[2017/06/13 03:37:48 PM] [INFO] running ifup on interface: eth5
[2017/06/13 03:37:48 PM] [INFO] running ifup on interface: eth4
[2017/06/13 03:37:48 PM] [INFO] running ifup on interface: eth7
[2017/06/13 03:37:48 PM] [INFO] running ifup on interface: eth6
[2017/06/13 03:37:48 PM] [INFO] running ifup on interface: bond0
[2017/06/13 03:37:53 PM] [INFO] running ifup on interface: bond2
[2017/06/13 03:37:53 PM] [INFO] running ifup on interface: bond3
[2017/06/13 03:37:54 PM] [INFO] running ifup on interface: vlan200
[2017/06/13 03:37:54 PM] [INFO] running ifup on interface: vlan300
[2017/06/13 03:37:55 PM] [INFO] running ifup on interface: vlan100
```

After checking the controller nic-config templates, there was and indentation error in the controller.yml file, with the vlan line in eth3 configuration, once the files where modified and the stack rerun the installation continued ok.

Checking info in /var/lib/heat-config

One dir that has a lot of deploy information is /var/lib/heat-config

```
 [root@overcloud-controller-0 ~]# ls /var/lib/heat-config
deployed  heat-config-puppet  heat-config-script
```

The deployed dir has all the software deployment json files and the values returned to heat , there are 2 tipes of files, the normal .json file that has the software configuration deployment info and the notify.json files that have the information that is sent back to heat once the software configuration is run:

```
 [root@overcloud-controller-0 ~]# ls /var/lib/heat-config/deployed/
0a79cff6-4a38-4c5a-9fa3-f1616e8b149c.json         a8b511c6-1674-4591-b0be-
10635a461b5d.json         d46a530b-7b16-4948-bc2f-6b28774af5d8.json         f5cf6214-
ad7a-474f-a1a2-b8bb46928447.json
0a79cff6-4a38-4c5a-9fa3-f1616e8b149c.notify.json  a8b511c6-1674-4591-b0be-
10635a461b5d.notify.json  d46a530b-7b16-4948-bc2f-6b28774af5d8.notify.json  f6ab440a-
2ba5-461e-8d88-3b11ec51c68c.json
13fa3272-df78-4496-8cee-0d31205697d5.json         b346901b-2da7-48a4-b71a-
46efa69cc40c.json         e5107f50-dc0c-4235-b01c-02a97408f570.json         f6ab440a-
2ba5-461e-8d88-3b11ec51c68c.notify.json
13fa3272-df78-4496-8cee-0d31205697d5.notify.json  b346901b-2da7-48a4-b71a-
46efa69cc40c.notify.json  e5107f50-dc0c-4235-b01c-02a97408f570.notify.json
```

Here is and example:

```
 [root@overcloud-controller-0 deployed]# cat a8b511c6-1674-4591-b0be-10635a461b5d.json
| grep config
  "config": "#!/bin/bash\nset -e\n\nfunction ping_retry() {\n  local IP_ADDR=$1\n
local TIMES=${2:-'10'}\n  local COUNT=0\n  local PING_CMD=ping\n  if [[ $IP_ADDR =~
\":\" ]]; then\n    PING_CMD=ping6\n  fi\n  until [ $COUNT -ge $TIMES ]; do\n    if
$PING_CMD -w 300 -c 1 $IP_ADDR &> /dev/null; then\n      echo \"Ping to $IP_ADDR
succeeded.\"\n      return 0\n    fi\n    echo \"Ping to $IP_ADDR failed.
Retrying...\"\n    COUNT=$(($COUNT + 1))\n  done\n  return 1\n}\n\n# For each unique
remote IP (specified via Heat) we check to\n# see if one of the locally configured
networks matches and if so we\n# attempt a ping test the remote network IP.\nfunction
ping_controller_ips() {\n  local REMOTE_IPS=$1\n  for REMOTE_IP in $(echo $REMOTE_IPS
| sed -e \"s| |\\n|g\" | sort -u); do\n    if [[ $REMOTE_IP =~ \":\" ]]; then\n
networks=$(ip -6 r | grep -v default | cut -d \" \" -f 1 | grep -v \"unreachable\")\n
else\n      networks=$(ip r | grep -v default | cut -d \" \" -f 1)\n    fi\n    for
LOCAL_NETWORK in $networks; do\n      in_network=$(python -c \"import ipaddr;
net=ipaddr.IPNetwork('$LOCAL_NETWORK'); addr=ipaddr.IPAddress('$REMOTE_IP');
print(addr in net)\")\n      if [[ $in_network == \"True\" ]]; then\n        echo
\"Trying to ping $REMOTE_IP for local network ${LOCAL_NETWORK}.\"\n        set +e\n
if ! ping_retry $REMOTE_IP; then\n          echo \"FAILURE\"\n          echo
\"$REMOTE_IP is not pingable. Local Network: $LOCAL_NETWORK\" >&2\n          exit 1\n
fi\n        set -e\n        echo \"SUCCESS\"\n      fi\n    done\n  done\n}\n\n# Ping
all default gateways. There should only be one\n# if using upstream t-h-t network
templates but we test\n# all of them should some manual network config have\n#
multiple gateways.\nfunction ping_default_gateways() {\n  DEFAULT_GW=$(ip r | grep
^default | cut -d \" \" -f 3)\n  set +e\n  for GW in $DEFAULT_GW; do\n    echo -n
\"Trying to ping default gateway ${GW}...\"\n    if ! ping_retry $GW; then\n      echo
\"FAILURE\"\n      echo \"$GW is not pingable.\"\n      exit 1\n    fi\n  done\n  set
-e\n  echo \"SUCCESS\"\n}\n\nping_controller_ips
\"$ping_test_ips\"\nping_default_gateways\n",
[root@overcloud-controller-0 deployed]# cat a8b511c6-1674-4591-b0be-
10635a461b5d.notify.json
{
  "deploy_stdout": "Trying to ping 10.0.0.58 for local network 10.0.0.0/24.\nPing to
10.0.0.58 succeeded.\nSUCCESS\nTrying to ping 192.168.101.150 for local network
192.168.101.0/24.\nPing to 192.168.101.150 succeeded.\nSUCCESS\nTrying to ping
192.168.1.151 for local network 192.168.1.0/24.\nPing to 192.168.1.151
succeeded.\nSUCCESS\nTrying to ping 192.168.2.152 for local network
192.168.2.0/24.\nPing to 192.168.2.152 succeeded.\nSUCCESS\nTrying to ping
192.168.3.161 for local network 192.168.3.0/24.\nPing to 192.168.3.161
succeeded.\nSUCCESS\nTrying to ping default gateway 192.168.101.1...Ping to
192.168.101.1 succeeded.\nSUCCESS\n",
  "deploy_stderr": "",
  "deploy_status_code": 0
```

We can also check with the UUID the info of the software config on heat:

```
 [stack@undercloud ~]$ heat config-list | grep -i a8b511c6-1674-4591-b0be-10635a461b5d
WARNING (shell) "heat config-list" is deprecated, please use "openstack software
config list" instead
| a8b511c6-1674-4591-b0be-10635a461b5d | ControllerAllNodesValidationDeployment
| script          | 2017-06-15T13:55:32Z |
[stack@undercloud ~]$ heat config-show a8b511c6-1674-4591-b0be-10635a461b5d
WARNING (shell) "heat config-show" is deprecated, please use "openstack software
config show" instead
{
  "inputs": [
    {
      "default": "192.168.101.150 192.168.2.152 192.168.4.161 192.168.1.151
192.168.3.161 10.0.0.58",
      "type": "String",
      "name": "ping_test_ips",
      "value": "192.168.101.150 192.168.2.152 192.168.4.161 192.168.1.151
192.168.3.161 10.0.0.58",
      "description": ""
    },
```

os-collect-config

This is a tool that starts at boot time, and continues to run via systemd. It's primary responsibility is to monitor and download metadata from the Heat API. If the data has changed, it automatically calls os-refresh-config. Its configured at boot time by cloud-init that populates /etc/os-collect-config.conf with credentials and the metadata url of the undercloud.

```
 [root@ctrl02 ~]# cat /etc/os-collect-config.conf
 [DEFAULT]
command = os-refresh-config --timeout 14400
collectors = ec2
collectors = request
collectors = local
 [request]
metadata_url = http://10.0.0.10:8080/v1/AUTH_e6128b61cf7a4169a8f61cdf41a27344/ov-
74kpxh4ucep-1-4sgqxt3uvuen-Controller-upu7hcpbqzlx/d5c2ea57-1c1c-4366-82f8-
01acdded95e7?temp_url_sig=f64f4ea5483aca3caff9e3cf6a7ce2539cf352ef&temp_url_expires=21
47483586
```

os-refresh-config

os-collet-config calls another tool call os-refresh-config, among them you can see os-net-config, os-refresh-config doesn't have a way to signal errors to heat, that's why failures like the ones in the network config don't get reported back to heat, and you will have to wait for a timeout

The important steps that os-refresh-config take are as follows:

- Apply systemctl configurables

- Run os-apply-config (see below)
- Configure the networking for the host
- Download and set the hieradata files for puppet parameters
- Configure /etc/hosts
- Deploy software configuration with Heat

It is within these scripts that the metadata downloaded by os-collect-config will be acted upon, these are this scripts that run

```
 [root@ctrl01 ~]# ls -l /usr/libexec/os-refresh-config/configure.d/
total 32
-rwxr-xr-x. 1 root root   42 May  4 12:13 20-os-apply-config
-rwxr-xr-x. 1 root root 3547 May  4 12:13 20-os-net-config
-rwxr-xr-x. 1 root root  629 May  4 12:13 25-set-network-gateway
-rwxr-xr-x. 1 root root 2830 May  4 12:13 40-hiera-datafiles
-rwxr-xr-x. 1 root root  394 May  4 12:13 40-truncate-nova-config
-rwxr-xr-x. 1 root root 1387 May  4 12:13 51-hosts
-rwxr-xr-x. 1 root root 6575 Sep  8  2016 55-heat-config
```

If needed you can run the scripts by hand to get the output on screen for debugging purposes. Heat seeds the SoftwareDeployment to the nodes in the form of metadata, and relies on os-collect-config to download it, and os-refresh-config to apply it.

Checking puppet in the overcloud nodes

The hieradata files are populated from metadata in heat by the 40-hiera-datafiles script that is run by os-refresh-config, it dumps thehieradata info in /etc/puppet/hieradata, it can be very useful to check variable data in this dir, so for example we are wondering were this ip came from

```
 [root@ctrl01 hieradata]# pcs status | grep 192.168.2.190
ip-192.168.2.190       (ocf::heartbeat:IPaddr2):       Started ctrl01`
```

We can check in the hieradata files

```
 root@ctrl01 hieradata]# grep 192.168.2.190 /etc/puppet/hieradata/vip_data.yaml
redis_vip: 192.168.2.190
tripleo::keepalived::redis_virtual_ip: 192.168.2.190
```

Also service conf files variables, for example the endpoint on this host for the tenant vxlan tunnel

```
 root@ctrl01 hieradata]# grep local_ip /etc/puppet/hieradata/service_configs.yaml
neutron::agents::ml2::ovs::local_ip: 192.168.3.150
```

Puppet code is run from dir "", if puppet is getting at some point we can kill the puppet process and

run it by hand

Then, make it so that we deny everyone from removing the stack, even if you're an admin. This means that the policy would have to be modified again for it to be deleted in the future (which may ensure that a higher privilege level is required for security and business continuity):

```
 undercloud$ sudo grep -m1 stacks:delete /etc/heat/policy.json
    "stacks:delete": "rule:deny_stack_user",
undercloud$ sudo sed -i /stacks:delete/{s/rule:.*/'rule:deny_everybody",'/}
/etc/heat/policy.json
undercloud$ sudo grep -m1 stacks:delete /etc/heat/policy.json
    "stacks:delete": "rule:deny_everybody",
```

Then try to delete the overcloud stack:

```
 undercloud$ source ~/stackrc
undercloud$ openstack stack delete overcloud
Are you sure you want to delete this stack(s) [y/N]? y
Forbidden: overcloud
Unable to delete 1 of the 1 stacks.
```

As you can see heat prevents us from deleting the stac

Deleting and cleaning up the overcloud

As the stack user connect to the undercloud, and start by deleting the stack, using the --wait option is a good choice

```
 stack@undercloud advanced]$ openstack stack delete overcloud --wait
Are you sure you want to delete this stack(s) [y/N]? y
2017-06-18 21:04:19Z [overcloud]: DELETE_IN_PROGRESS  Stack DELETE started
2017-06-18 21:04:20Z [overcloud.BlockStorageSshKnownHostsDeployment]:
DELETE_IN_PROGRESS  state changed
```

With the deletion of the stack, it should have also freed up our "baremetal" nodes (our virtual machine based infrastructure for our overcloud) by powering them down via the Ironic pxe_ssh driver. When a node in ironic has been deleted, it reverts back to an 'available' and 'powered off' state.

```
 [stack@undercloud advanced]$ openstack baremetal list
This command is deprecated. Instead, use 'openstack baremetal node list'.
+--------------------------------------+-----------------+--------------
+-------------+--------------------+-------------+
| UUID                                 | Name            | Instance UUID | Power State
| Provisioning State | Maintenance |
+--------------------------------------+-----------------+--------------
+-------------+--------------------+-------------+
| 4ad48228-68b4-4d61-9abd-e414d7148621 | tricontroller01 | None          | power off
| available          | False       |
| 04006063-8d62-4532-bad7-14913a518f4a | tricontroller02 | None          | power off
| available          | False       |
| 00c212e2-1dd8-4518-9648-8150b8e327d2 | tricontroller03 | None          | power off
| available          | False       |
| be734001-85c5-4e3c-a98a-5bb67d348522 | tricompute01    | None          | power off
| available          | False       |
+--------------------------------------+-----------------+--------------
+-------------+--------------------+-------------+
```

Delete the Mistral plan associated with the stack.

```
 [stack@undercloud advanced]$ openstack overcloud plan delete overcloud
Deleting plan overcloud...
```

Delete the supporting files related to the previous installation.

```
[stack@undercloud advanced]$ rm overcloudrc overcloud-env.json
```

With that you have removed your overcloud if you need to go further and delete the ironic nodes you can do

```
 for uuid in $(ironic node-list | grep -v UUID | awk -F'|' '{ print $2}' | xargs`) ;
do
    ironic node-set-maintenance $uuid false
    ironic node-delete $uuid
done
```

ironic node-list shows Instance UUID even after deleting the stack

```
 [stack@instack ~]$ heat stack-list
+----+------------+--------------+--------------+--------------+
| id | stack_name | stack_status | creation_time | updated_time |
+----+------------+--------------+--------------+--------------+
+----+------------+--------------+--------------+--------------+
[stack@instack ~]$ nova list
+----+------+--------+------------+-------------+----------+
| ID | Name | Status | Task State | Power State | Networks |
+----+------+--------+------------+-------------+----------+
+----+------+--------+------------+-------------+----------+
[stack@instack ~]$ ironic node-list
+--------------------------------------+------+--------------------------------------
+-------------+--------------------+-------------+
| UUID                                 | Name | Instance UUID                        |
Power State | Provisioning State | Maintenance |
+--------------------------------------+------+--------------------------------------
+-------------+--------------------+-------------+
| 9e57d620-3ec5-4b5e-96b1-bf56cce43411 | None | 1b7a6e50-3c15-4228-85d4-1f666a200ad5 |
power off   | available          | False       |
| 88b73085-1c8e-4b6d-bd0b-b876060e2e81 | None | 31196811-ee42-4df7-b8e2-6c83a716f5d9 |
power off   | available          | False       |
| d3ac9b50-bfe4-435b-a6f8-05545cd4a629 | None | 2b962287-6e1f-4f75-8991-46b3fa01e942 |
power off   | available          | False       |
+--------------------------------------+------+--------------------------------------
+-------------+--------------------+-------------+
```

We can use ironic node-update <node_uuid> remove instance_uuid

```
ironic node-update 9e57d620-3ec5-4b5e-96b1-bf56cce43411 remove instance_uuid
```

There are also times in OSP10 when nodes remain in active state after the stack has been deleted(bz: 1263186)