Feb 9th

Thursday, February 1, 2018 8:38 AM

Title: Deploy Controllers with TripleO and deploy Compute nodes with <blank>

Summary:

Introductions: Mark Danielson (IT Architect for the College of Engineering).

What: Introduce and give an overview of the idea of using TripleO to deploy the control plane and using any other deployment strategy to deploy the Compute Nodes.

TripleO Diagram

Why: TripleO is a complex tool used to deploy a complex infrastructure.

Controller Components

However, the Compute nodes are not as complex once the control plane is deployed. Very Simple relatively speaking.

Compute Components

If you have the time to let your architect learn TripleO, then great! The more the merrier. However, if not, this method allows your architect to design their own deployment of the compute node leaving the control plane deployment to those with more time on their hands. <or leave said hands to do both>

Technical Reason:

1) The same golden image used to deploy the controllers is the same used to deploy compute. So when it comes time to patch you are patching dozens if not hundreds of bits on a compute node that you will never use but have to update.

2) *infrastructure as code* Risks of touching every machine when making changes. Touching Machines

How:

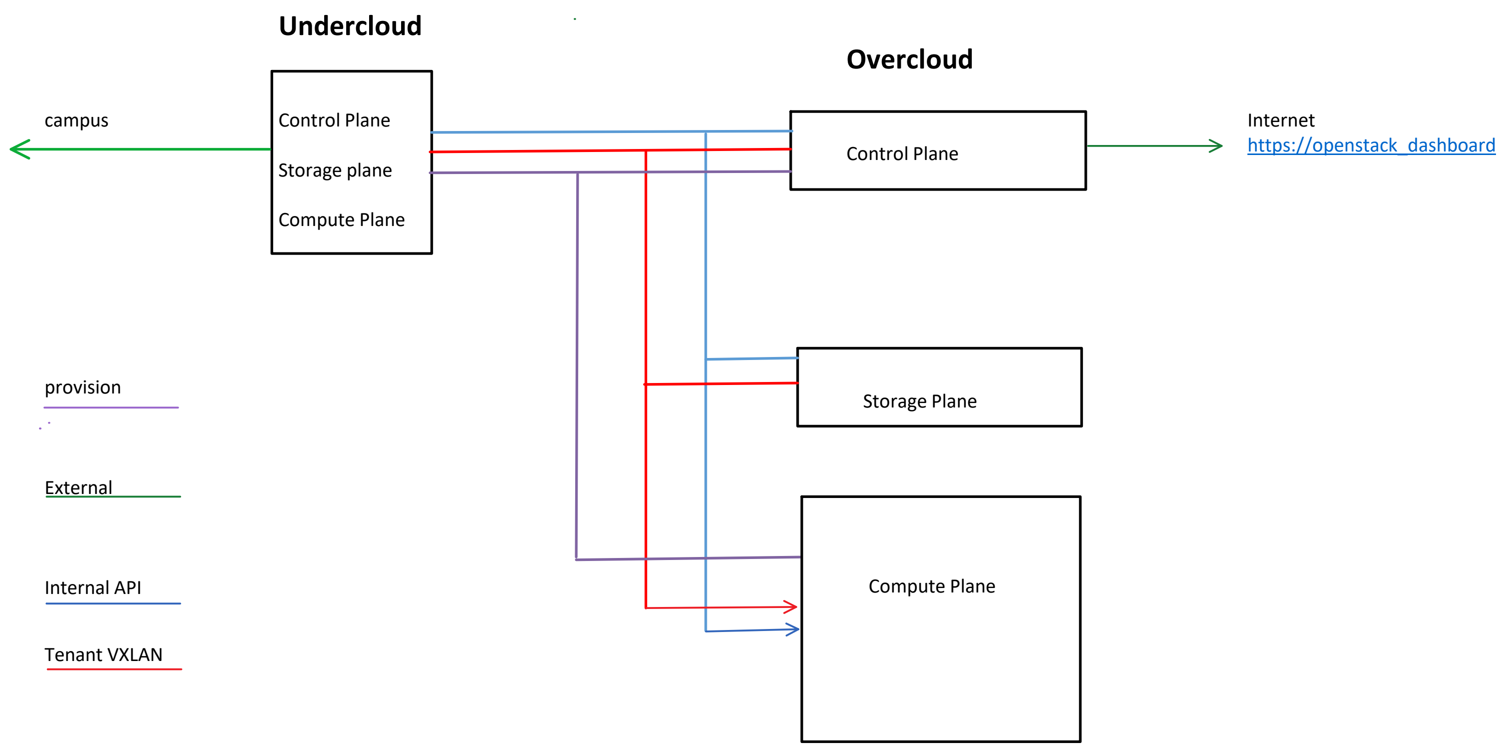
Salt Outline

Where:

- This notebook is available in the Teams channel File repository.
- Working on a more friendly git repo I can share
- mdanielson@gatech.edu
- Tech Tower 4th Floor Rm 438

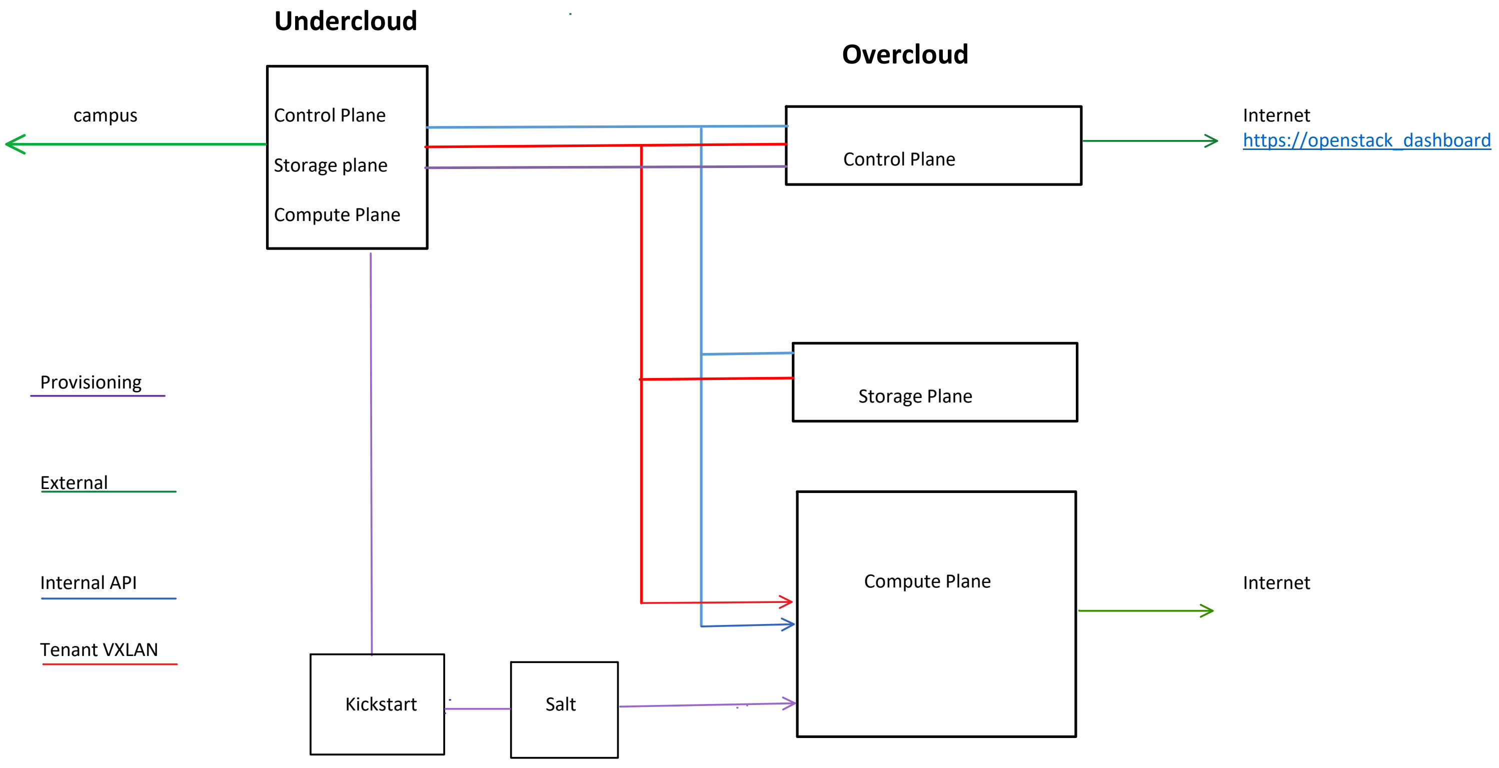
TripleO Diagram

Friday, February 2, 2018 9:08 AM



TripleO Diagram Improved(?)

Friday, February 2, 2018 9:08 AM



ToDo:

Friday, February 2, 2018 9:29 AM

- ~~Create list of Controller Components~~
- ~~Create list of Compute Components~~
- ~~Touching Every Machine??~~
- ~~SaltStack Outline~~

Controller Components

Wednesday, February 7, 2018 11:02 AM

- OpenStack Neutron DHCP Agent
- OpenStack Neutron Layer 3 Agent
- OpenStack Neutron Metadata Agent
- OpenStack Neutron Open vSwitch Agent
- OpenStack Neutron Open vSwitch Cleanup Utility
- OpenStack Neutron Server
- OpenStack Alarm evaluator service
- OpenStack Alarm listener service
- OpenStack Alarm notifier service
- OpenStack ceilometer central agent
- OpenStack ceilometer collection service
- OpenStack ceilometer notification agent
- OpenStack Cinder API Server
- OpenStack Cinder Backup Server
- OpenStack Cinder Scheduler Server
- Cluster Controlled openstack-cinder-volume
- OpenStack Image Service (code-named Glance) API
- OpenStack Image Service (code-named Glance) Regi
- OpenStack gnocchi metricd service
- OpenStack gnocchi statsd service
- Openstack Heat CFN-compatible API Service
- OpenStack Heat CloudWatch API Service
- OpenStack Heat API Service
- Openstack Heat Engine Service
- OpenStack Nova API Server
- OpenStack Nova Cert Server
- OpenStack Nova Conductor Server
- OpenStack Nova VNC console auth Server
- OpenStack Nova NoVNC Proxy Server
- OpenStack Nova Scheduler Server

- OpenStack Object Storage (swift) - Account Audit
- OpenStack Object Storage (swift) - Account Reape
- OpenStack Object Storage (swift) - Account Repli
- OpenStack Object Storage (swift) - Account Serve
- OpenStack Object Storage (swift) - Container Aud
- OpenStack Object Storage (swift) - Container Rep
- OpenStack Object Storage (swift) - Container Upd
- OpenStack Object Storage (swift) - Container Ser
- OpenStack Object Storage (swift) - Object Audito
- OpenStack Object Storage (swift) - Object Replic
- OpenStack Object Storage (swift) - Object Update
- OpenStack Object Storage (swift) - Object Server
- OpenStack Object Storage (swift) - Proxy Server
- Open vSwitch Internal Unit
- Open vSwitch
- MySQL Server
- RabbitMQ
- REDIS
- MongoDB

All Components are installed on three nodes configured via PaceMaker/Corosync and available via HAProxy

Compute Components

Wednesday, February 7, 2018 11:13 AM

Services:

- OpenStack Neutron Open vSwitch Agent
- OpenStack Neutron Open vSwitch Cleanup Utility
- OpenStack ceilometer compute agent
- OpenStack Nova Compute Server
- Open vSwitch

Salt state file:

```
include:
      - openstack/compute_dir/compute_packages
      - openstack/compute_dir/networks
      - openstack/compute_dir/ovs
      - openstack/compute_dir/iptables
      - openstack/compute_dir/nova_conf
      - openstack/compute_dir/neutron_conf
      - openstack/compute_dir/ceph_conf
      - openstack/compute_dir/ceilometer_conf
      - openstack/compute_dir/cinder_conf
```

Wednesday, February 7, 2018 11:39 AM

-
-

Salt Outline

Thursday, February 8, 2018 2:19 PM

Pre-Salt

Node Group Isolation:

#On Master:
Add to the Node Group (Compute, or Project Specific)

Pillars:
Add Pillars necessary for compute node role:

- Role: Compute
- Openstack Passwords and authorities
- Bonding Slaves (Not used at the moment
- Networks:
    - Need to be long to these Vlans:
        - Control_plane
        - Internal_API
        - Tenant
        - Storage
    - Sample Pillar file

State Files:

- Install packages
- Setup Network access to control plane
- Add OVS bridges br-ex,br-int,br-tun
- Add bond interface to external bridge (br-ex)
- Setup iptable templates *neutron*
- Installing and Configuring: *templates instead of modules atm*
    - Nova
    - Libvirt *unless*
    - Neutron *permissions*
    - Ceph *moving libvirt secrets here*
    - Ceilometer
    - Cinder
- Last minute check on important services

Pre-Salt

Thursday, February 8, 2018 2:32 PM

#Pre-salt

1. Partition Hard Disks
2. Set Root Password
3. Create sudo account
4. Add salt master RSA pub key sudo account
5. Disable NetworkManager
6. Enable network
7. Configure network access to salt master
8. Change Hostname
9. Configure Salt Yum Repository
10. Add salt master to Host file
11. Register to satellite.gatech.edu
12. Install salt minion software
13. Update OS
14. Reboot

#On Salt Master:

1. Configure ssh config to node
2. Generate salt key (if not already done)
3. Copy public key to master key directory
4. Copy keys to node
5. Config id name in minion config
6. Enable salt-minion service
7. Start salt-minion service
8. Test with ping module

Preseed

Friday, February 9, 2018 9:36 AM

On Master:
~/.ssh/config:

```
…
Host r3-r5-18
     IdentityFile ~/.ssh/fedora-atomic
     User rhel
     Hostname 172.31.16.231
…
```

```
#!/bin/bash
minion=$1
Host=$2
salt-key --gen-keys=$minion

cp $minion.pub /etc/salt/pki/master/minions/$minion
scp /etc/salt/pre_keys/$minion.* $Host:~/
ssh $Host 'sudo cp ~/*.pem /etc/salt/pki/minion.pem'
ssh $Host 'sudo cp ~/*.pub /etc/salt/pki/minion.pub'
ssh $Host "sudo sed -i -e 's|#id:|id: $minion|' /etc/salt/minion"
Ssh $Host "sudo systemctl enable salt-minion"
ssh $Host "sudo systemctl start salt-minion"
```

Nodegroups

Friday, February 9, 2018 9:38 AM

/etc/salt/master:

```
nodegroups:
   kube: 'G@environment:kube'
   ceph: 'I@environment:ceph-prod'
   asdl: 'r3-r5*'
   computes: 'compute*'
   controllers: 'controller*'
```

Top.sls

```
base:
  'I@roles:kube-master':
    - match: compound
    - kube/kube_master
  'I@roles:kube-node':
    - match: compound
    - kube/kube_nodes
  'I@roles:compute':
    - match: compound
    - openstack/compute
```

Sample Pillar file

Friday, February 9, 2018 9:44 AM

```
roles: ['compute']

include:
  - openstack/pass
bonding:
  slaves: ['eno1','eno2']
networks:
  Control_plane:
    ip: '172.31.23.154'
    vlan_tag: 'None'
  Internal_api:
    ip: '172.23.16.237'
    vlan_tag: '2516'
  Tenant:
    ip: '172.23.18.237'
    vlan_tag: '2518'
  Storage:
    ip: '172.31.17.237'
    vlan_tag: '2317'
```

Unless Libvirt

Friday, February 9, 2018 10:31 AM

```
- unless: "[ '{{ pillar['CephClientKey'] }}' = `virsh secret-get-value {{ pillar['rbd_secret'] }}` ]"
```