Provisioning OpenShift on OpenStack

Deploy & start using

Sim Zacks
Principal Engineer, Red Hat inc.

Sasha Segal
Senior Engineer, Red Hat inc.

Who we are

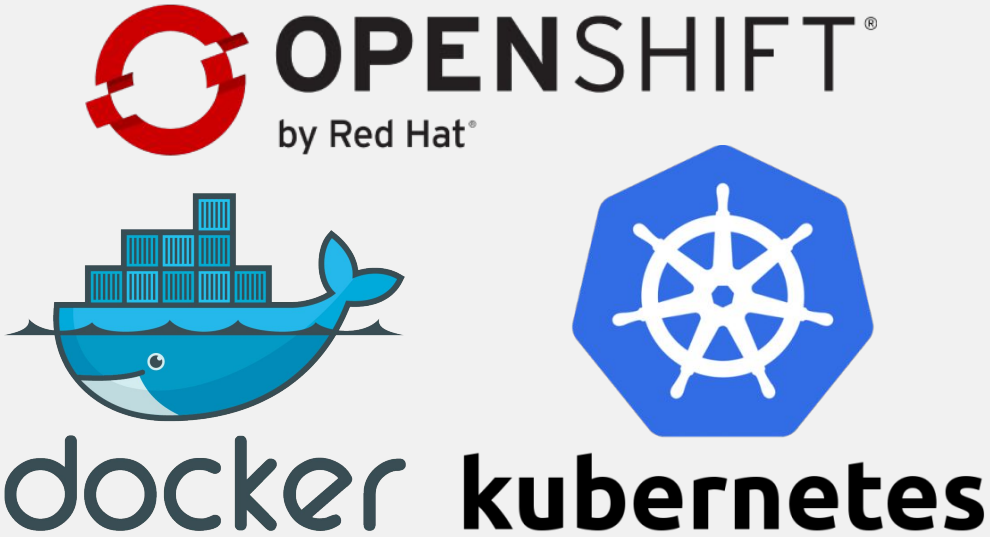
- DevOps Engineers - experts in software engineering and sys admin
- QE Ops - we build customer-like environments for testing products
- Central CI - we manage all the CI requirements for the testing env

Sim Zacks
Principal Engineer

Sasha Segal
Senior Engineer

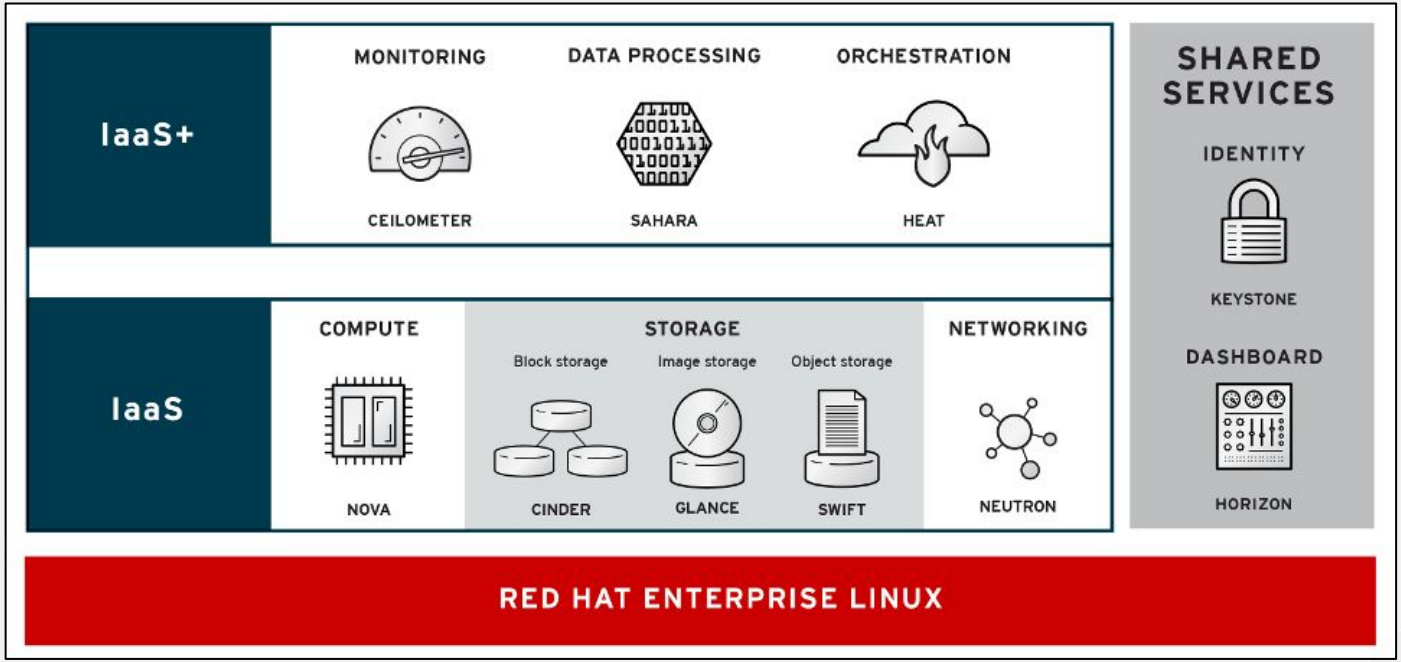
OCP

OpenShift Container Platform

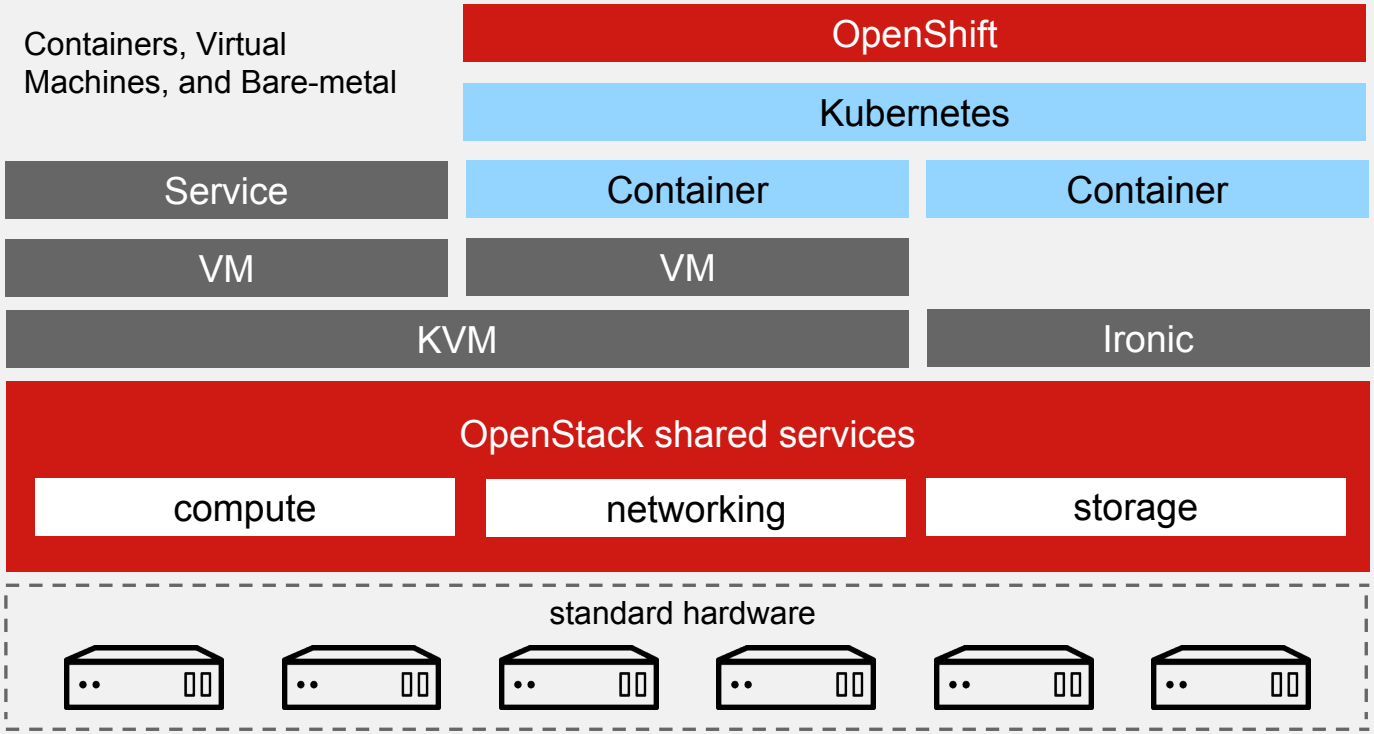


OSP

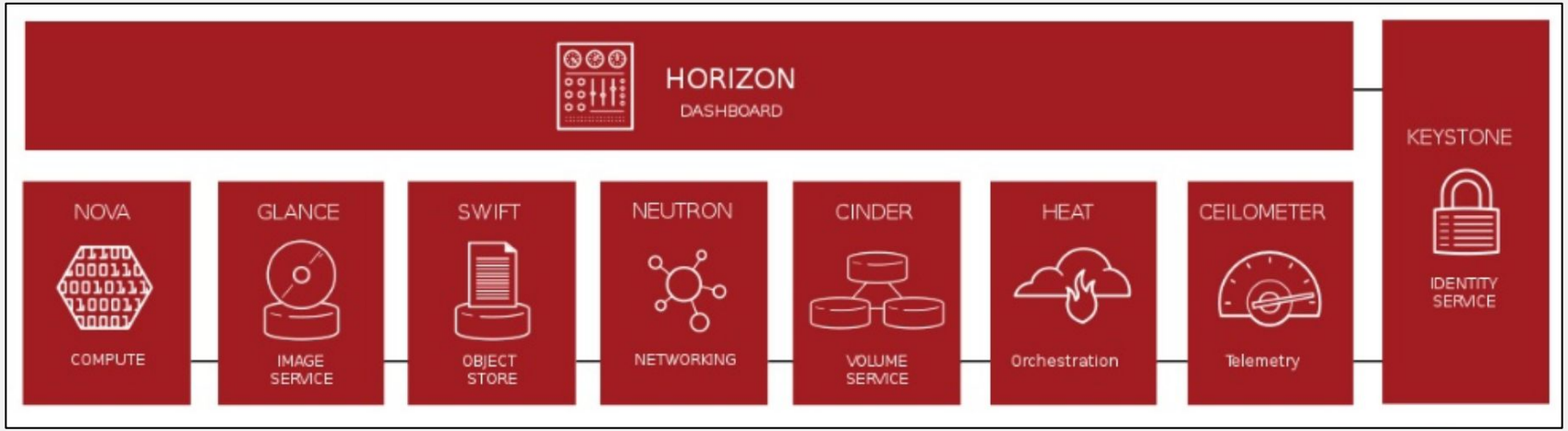
Red Hat OpenStack Platform



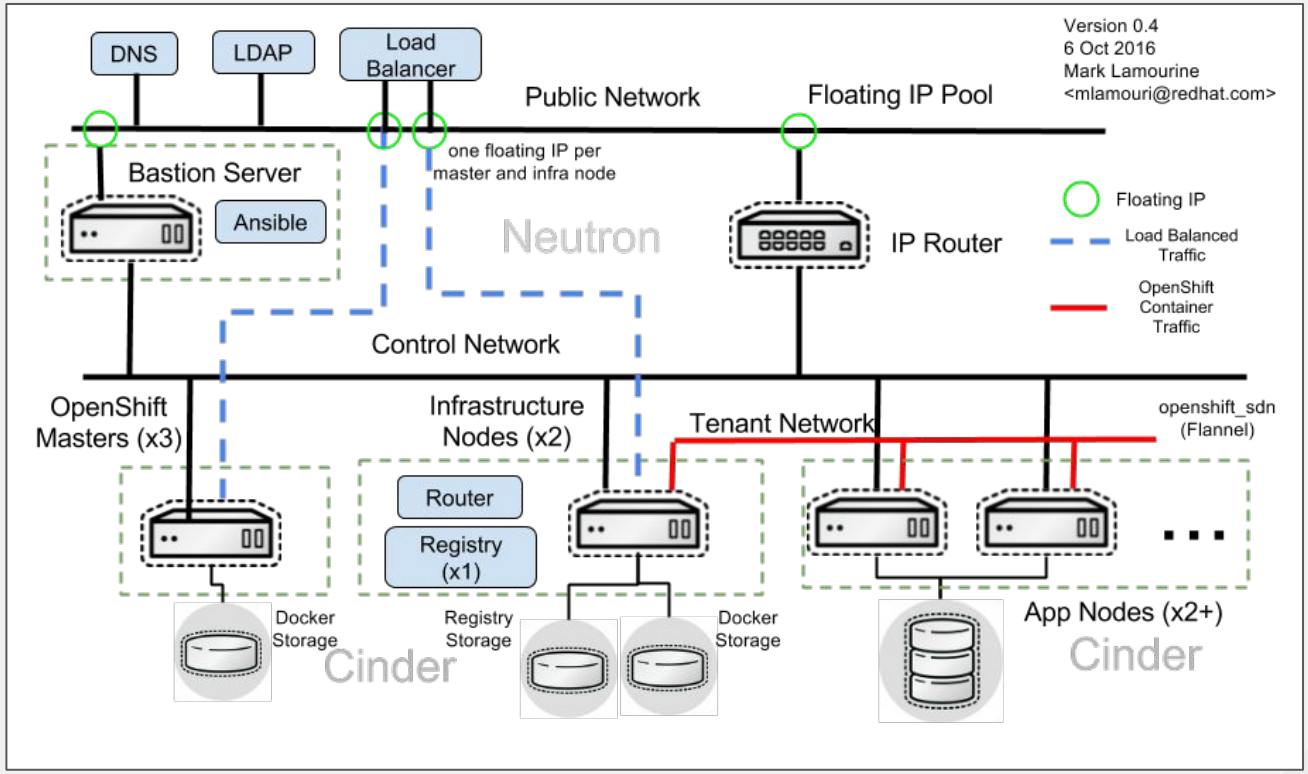
OCP on OSP - Benefits



OSP - Configuration

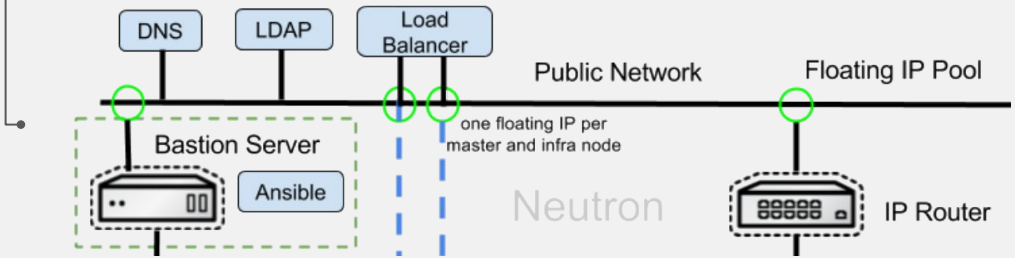


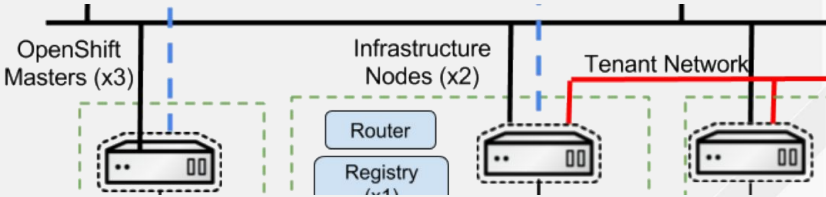
OCP - Architecture



OSP - Networks

- Tenant network - inter-container communication
- Control network - instance to instance communication
- External network - communication with the outside world





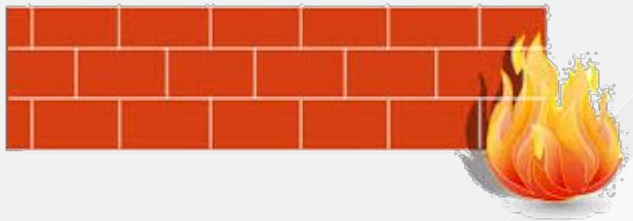
OCP - DNS

- Developers portal
- Wildcard for application domains
- Both pointing at the load balancer

Type	Domain Name	IP Address
A	developers.ocp.mydomain.com	62.124.XXX.137
A	*.apps.ocp.mydomain.com	62.124.XXX.137

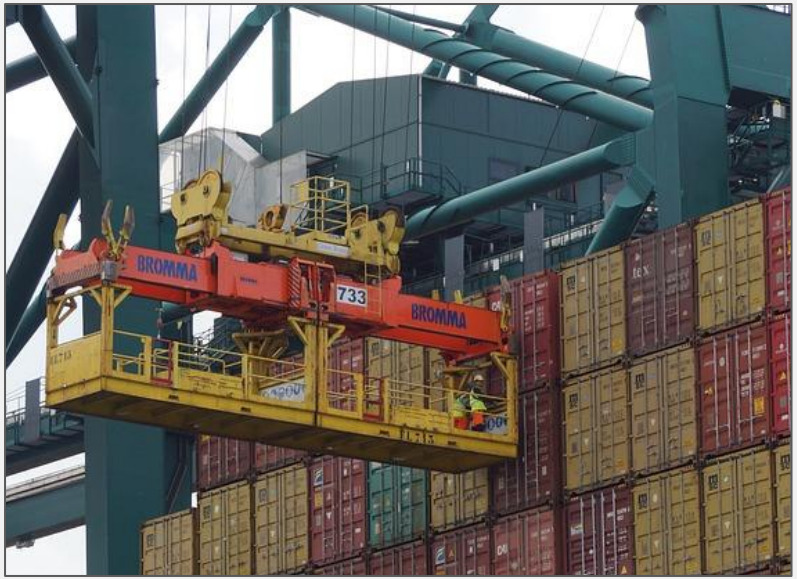
OSP - Security Groups

- All instances - SSH, ICMP, DNS
- Master - etcd, SDN, internal DNS, https, kubernetes and fluentd
- Infra nodes - http, https, SDN, http for the docker registry, kubernetes
- App nodes - SDN, Kubernetes



OSP - Provision Hosts

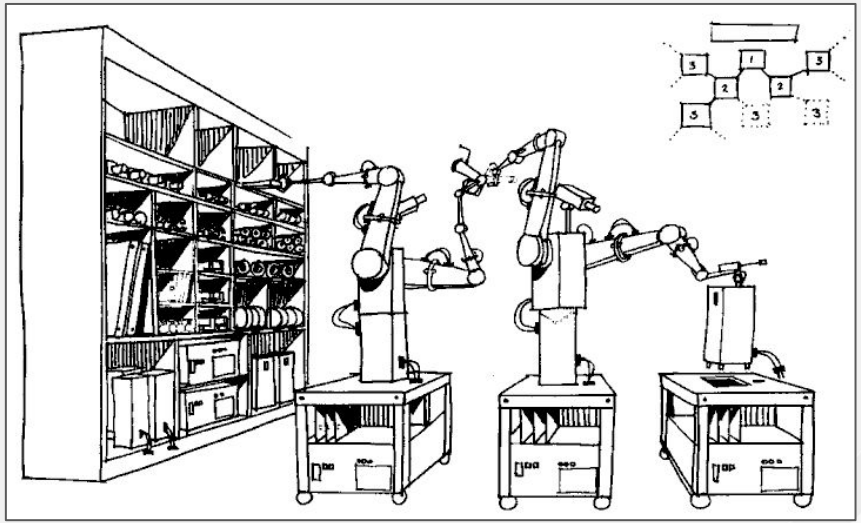
- RHEL 7
- Enable repos
- Choose security groups
- Attach networks
- Permanent storage for nodes
- Ansible on the bastion server
- Let ’er rip



OCP on OSP Implementations

OCP on OSP - Heat Orchestration

- Fully supported
- Define infrastructure in YAML templates
- Provision as needed
- Parameterized
- Use results to drive ansible OCP installation



OCP on OSP - Ansible Installer

https://github.com/redhat-openstack/openshift-on-openstack

OCP on OSP - Heat Process

```
$ heat stack-create my-openshift -t 180 \
      -e openshift_parameters.yaml \
      -P master_count=3 \
      -P infra_count=2 \
      -P deploy_router=true \
      -f openshift-on-openstack/openshift.yaml
```

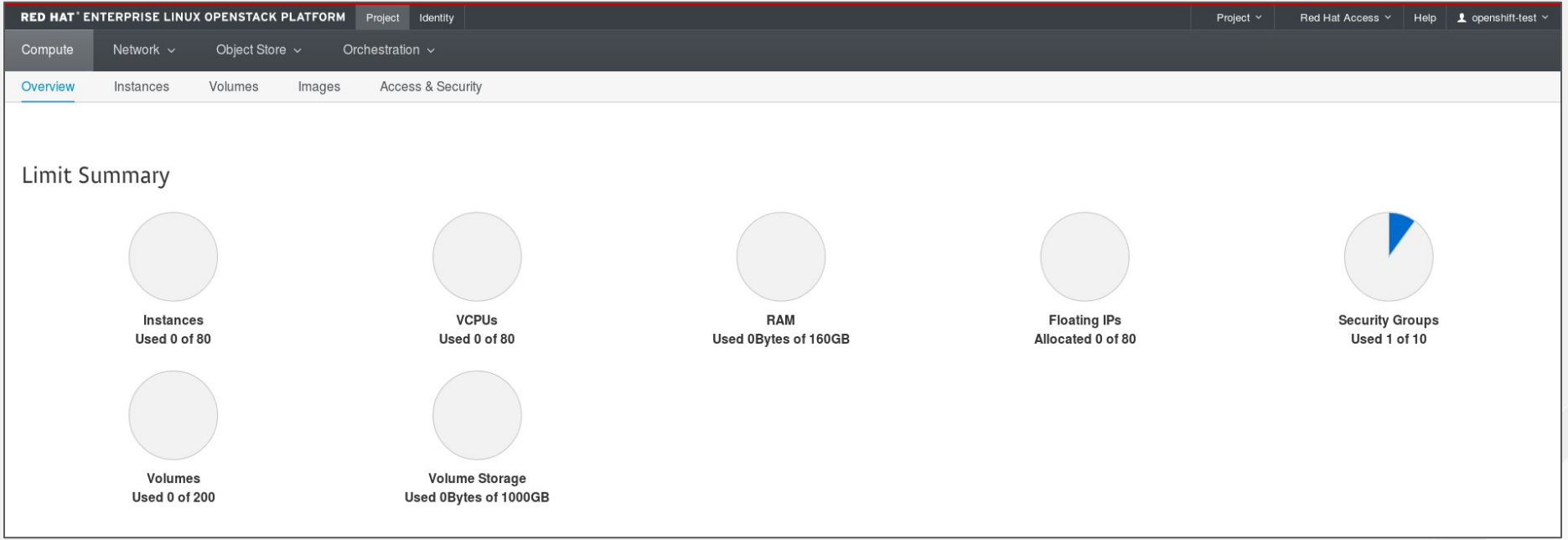
Pure Ansible PoC

OCP on OSP - Pure Ansible

- Testing environment - setup and tear down
- Fully automate with ansible, manage configuration drift
- One technology to be familiar with
- Parameterized playbook
- Ansible-openstack module

DEMO

Empty OSP Tenant

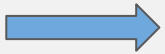


Configurable parameters

```
ose_nodes=["n057","n058","n059","n060","n061","n062","n063"]
primary_master="n058"
ose_masters=["n059","n060"]
ose_routers=["n061","n062"]
server_disk_layout=["/dev/vda:20","/dev/vdb:5"]
docker_disk="/dev/vdb"
flavor="m1.small"
security_groups="default"
availability_zone="nova"
dns_servers=["10.34.49.18","10.34.49.19"]
sshkey_name="osoos"
additional_servers=["lb-router","lb-master"]
subs_username="qa@redhat.com"
subs_password="redhatqa"
subs_serverurl="subscription.rhsm.stage.redhat.com:443/subscription"
subs_baseurl="https://cdn.redhat.com"
subs_pool="8a85f9823e3d5e43013e3ddd4e9509c4"
image="rhel-7.2-server-x86_64-updated"
haproxy_image="rhel-7.2-server-x86_64-released"
security_group="osoos"
external_net_name="external_net"
external_subnet_name="ext_sub1"
```

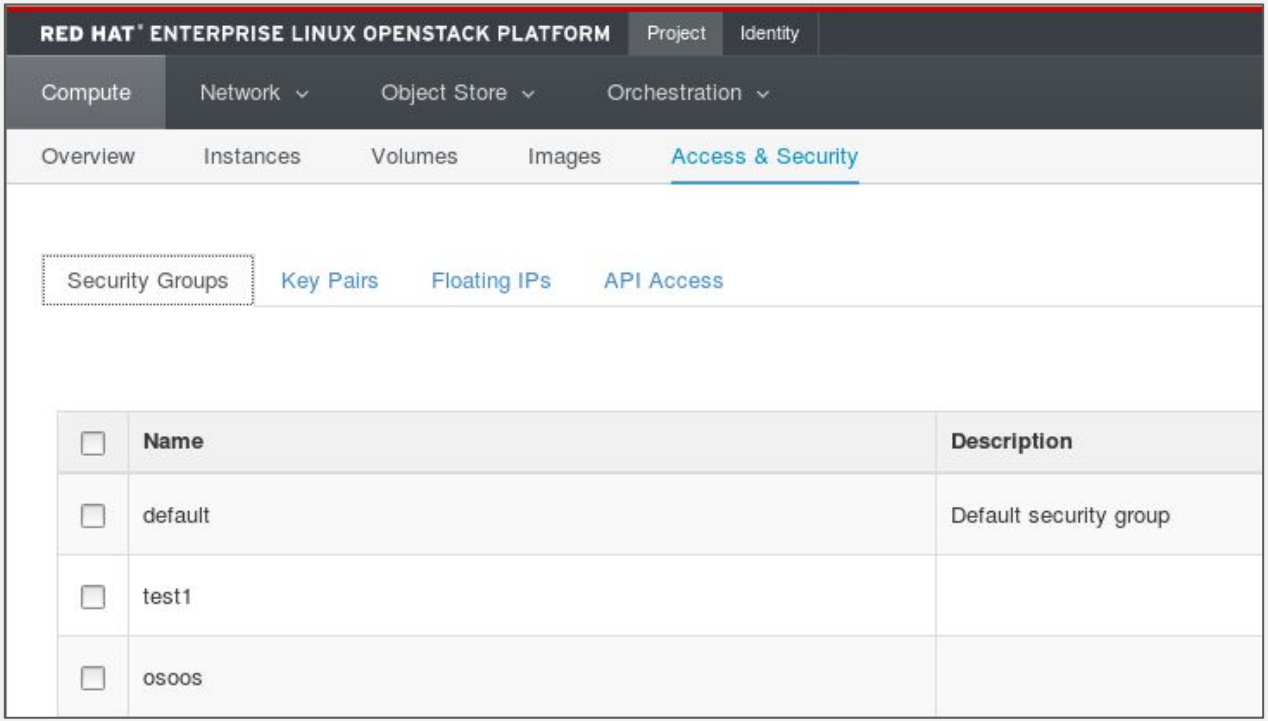
Preparing the environment

```
- name: Create security group
  os_security_group:
    name: "{{ security_group }}"
    state: present

- set_fact:
    structure: "{{ structure|default({}) | combine({'securi

- name: Store trail log
  copy: content="{{ structure }}" dest="{{ trail_file }}"

- name: Open all ingress traffic
  os_security_group_rule:
    security_group: "{{ security_group }}"
    direction: ingress
    remote_ip_prefix: 0.0.0.0/0
    state: present

- name: Open all egress
  os_security_group_rule:
    security_group: "{{ security_group }}"
    direction: egress
    remote_ip_prefix: 0.0.0.0/0
    state: present
```



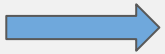
```
TASK [create_env : Log we completed previous stage] ***
changed: [localhost]

TASK [create_env : Create security group] *************
ok: [localhost]

TASK [create_env : set_fact] **************************
ok: [localhost]

TASK [create_env : Store trail log] *******************
changed: [localhost]

TASK [create_env : Open all ingress traffic] **********
ok: [localhost]

TASK [create_env : Open all egress] *******************
ok: [localhost]
```

Preparing the environment



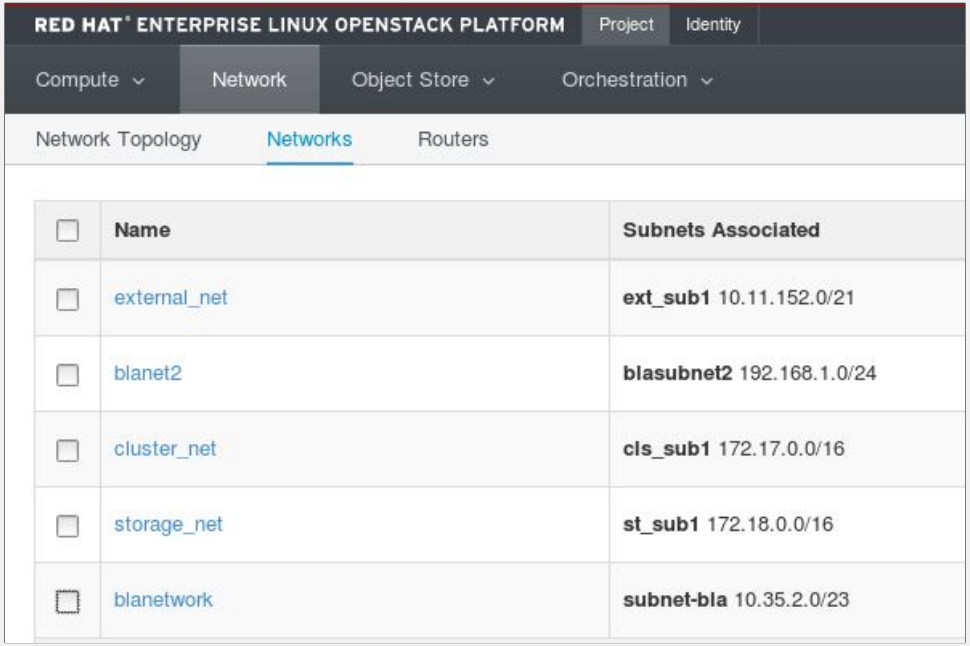
OSP network

```
- name: Create external network
  os_network:
    name: "{{ external_net_name }}"
    state: present
    external: false
  register: ext_net

- os_networks_facts: name="{{ external_net_name }}"

- name: Create subnet for external network
  os_subnet:
    name: "{{ external_subnet_name }}"
    state: present
    network_name: "{{ openstack_networks.0.name }}"
    cidr: "{{ external_subnet }}"

- name: Create ports for external network
  os_port:
    state: present
    name: "{{ item }}-ext"
    network: "{{ openstack_networks.0.name }}"
    fixed_ips:
      - ip_address: "{{ ports[item]['ext_ip'] }}"
    security_groups: "{{ security_group }}"
  with_items: "{{ ose_nodes + additional_servers }}"
  register: ext_ports
```



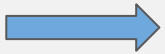
```
TASK [create_networks : Create external network] ***********
ok: [localhost]

TASK [create_networks : os_networks_facts] *****************
ok: [localhost]

TASK [create_networks : Create subnet for external network]
ok: [localhost]

TASK [create_networks : Create ports for external network] *
ok: [localhost] => (item=n057)
ok: [localhost] => (item=n058)
ok: [localhost] => (item=n059)
ok: [localhost] => (item=n060)
ok: [localhost] => (item=n061)
ok: [localhost] => (item=n062)
ok: [localhost] => (item=n063)
ok: [localhost] => (item=lb-router)
ok: [localhost] => (item=lb-master)

TASK [create_networks : Create cluster network] ************
ok: [localhost]

TASK [create_networks : Create subnet for cluster network] *
ok: [localhost]
```

OSP network



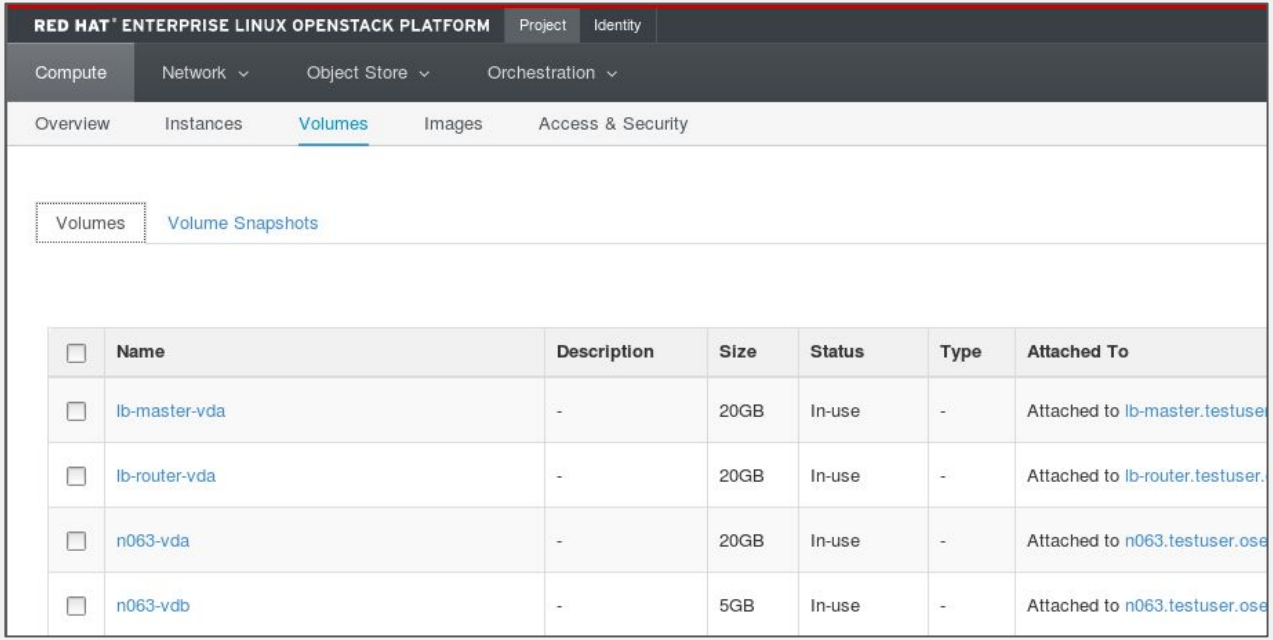
OSP disk

```
- name: Create volume
  os_volume: state=present availability_zone={{ availabili
.split(':').0.split('/')[-1] == 'vda' %}{{ image }}{% else
  with_nested:
    - "{{ ose_nodes }}"
    - "{{ server_disk_layout }}"
  register: disk_create
  async: 7200
  poll: 0

- name: Wait for disk create to complete
  async_status: jid={{ item.ansible_job_id }}
  register: disk_create_jobs
  until: disk_create_jobs.finished
  retries: 300
  with_items: "{{ disk_create.results }}"

- set_fact:
    disks: "{{ disks|default([]) + [ item.item.0 + '-' + i
  with_items: "{{ disk_create.results }}"
```



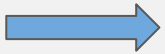
```
TASK [create_volumes : Create volume] *************
ok: [localhost] => (item=['n057', '/dev/vda:20'])
ok: [localhost] => (item=['n057', '/dev/vdb:5'])
ok: [localhost] => (item=['n058', '/dev/vda:20'])
ok: [localhost] => (item=['n058', '/dev/vdb:5'])
ok: [localhost] => (item=['n059', '/dev/vda:20'])
ok: [localhost] => (item=['n059', '/dev/vdb:5'])
ok: [localhost] => (item=['n060', '/dev/vda:20'])
ok: [localhost] => (item=['n060', '/dev/vdb:5'])
ok: [localhost] => (item=['n061', '/dev/vda:20'])
ok: [localhost] => (item=['n061', '/dev/vdb:5'])
ok: [localhost] => (item=['n062', '/dev/vda:20'])
ok: [localhost] => (item=['n062', '/dev/vdb:5'])
ok: [localhost] => (item=['n063', '/dev/vda:20'])
ok: [localhost] => (item=[u'n063', u'/dev/vdb:5'])
```

OSP disk



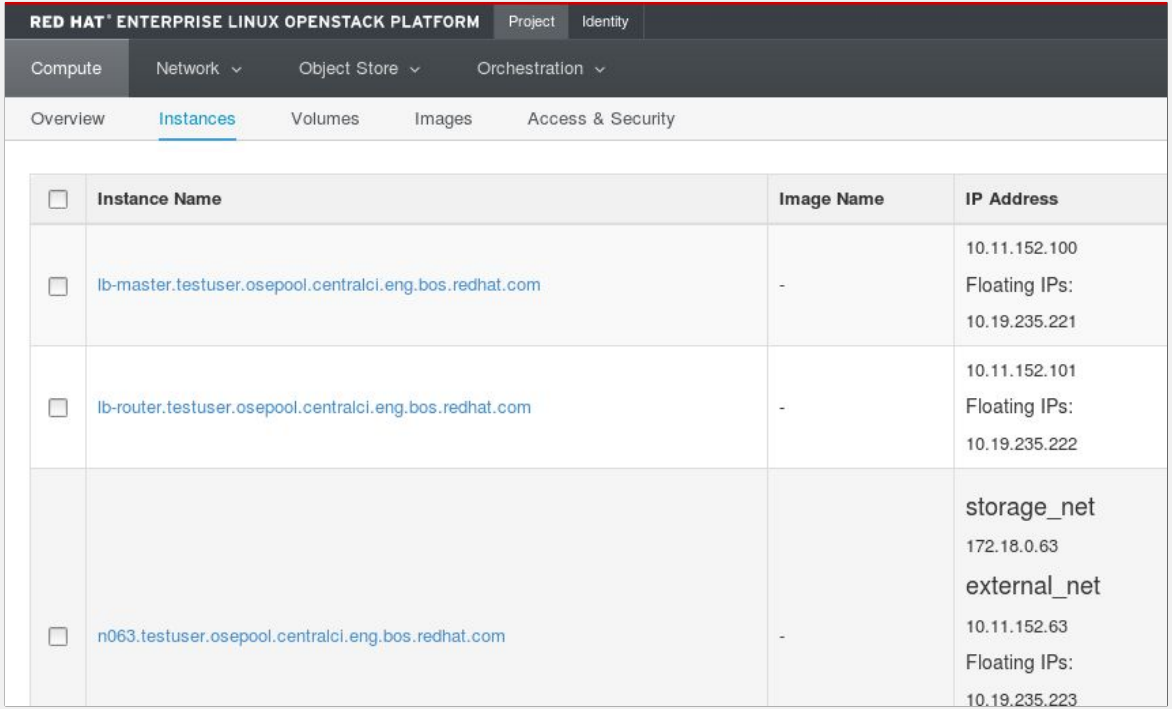
OSP instances

```yaml
- name: Create servers
  os_server:
    name: "{{ item }}.{{ project }}.{{ domain }}"
    state: present
    boot_volume: "{{ item }}-vda"
    flavor: "{{ flavor }}"
    key_name: "{{ sshkey_name }}"
    security_groups: "{{ security_group }}"
    availability_zone: "{{ availability_zone }}"
    auto_ip: no
    nics: [
      "net-id={{ external_net_id }},port-name={{ item }}-ext"]
  with_items:
    - "{{ all_servers }}"
  changed_when: no
  register: create_server
  async: 7200
  poll: 0

- name: Wait for server create
  async_status: jid={{ item.ansible_job_id }}
  register: create_server_jobs
  until: create_server_jobs.finished
  retries: 300
  with_items: "{{ create_server.results }}"
```



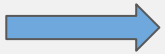
```
TASK [create_nodes : Create servers] **********
ok: [localhost] => (item=n057)
ok: [localhost] => (item=n058)
ok: [localhost] => (item=n059)
ok: [localhost] => (item=n060)
ok: [localhost] => (item=n061)
ok: [localhost] => (item=n062)
ok: [localhost] => (item=n063)
ok: [localhost] => (item=lb-router)
ok: [localhost] => (item=lb-master)

TASK [create_nodes : Wait for server create] **
FAILED - RETRYING: TASK: create_nodes : Wait fo
FAILED - RETRYING: TASK: create_nodes : Wait fo
FAILED - RETRYING: TASK: create_nodes : Wait fo
FAILED - RETRYING: TASK: create_nodes : Wait fo
changed: [localhost] => (item={u'_ansible_parse
 u'item': u'n057', u'finished': 0, u'results_fi
FAILED - RETRYING: TASK: create_nodes : Wait fo
changed: [localhost] => (item={u'_ansible_parse
'item': u'n058', u'finished': 0, u'results_file
changed: [localhost] => (item={u'_ansible_parse
```

OSP instances



Configuration

```
- name: Populate hosts file
  template: src=hosts.j2 dest=/tmp/{{ r }}-hosts

- name: Copy hosts file to nodes
  command: scp /tmp/{{ r }}-hosts {{ item }}:/etc/hosts
  with_items: "{{ ose_public.values() }}"
  register: populate_hosts
  async: 7200
  poll: 0

- name: Wait for hosts file copy
  async_status: jid={{ item.ansible_job_id }}
  register: populate_hosts_jobs
  until: populate_hosts_jobs.finished
  retries: 300
  with_items: "{{ populate_hosts.results }}"

- file:
    state: absent
    path: "/tmp/{{ r }}-hosts"
```



```
TASK [configure_nodes : Copy hosts file to nodes] ****
ok: [localhost] => (item=10.19.235.229)
ok: [localhost] => (item=10.19.235.227)
ok: [localhost] => (item=10.19.235.228)
ok: [localhost] => (item=10.19.235.225)
ok: [localhost] => (item=10.19.235.221)
ok: [localhost] => (item=10.19.235.226)
ok: [localhost] => (item=10.19.235.222)
ok: [localhost] => (item=10.19.235.224)
ok: [localhost] => (item=10.19.235.223)

TASK [configure_nodes : Wait for hosts file copy] ****
changed: [localhost] => (item={u'_ansible_parsed': Tru
5.229', u'finished': 0, u'results_file': u'/home/sasha
changed: [localhost] => (item={u'_ansible_parsed': Tru
35.227', u'finished': 0, u'results_file': u'/home/sash
changed: [localhost] => (item={u'_ansible_parsed': Tru
35.228', u'finished': 0, u'results_file': u'/home/sash
changed: [localhost] => (item={u'_ansible_parsed': Tru
35.225', u'finished': 0, u'results_file': u'/home/sash
changed: [localhost] => (item={u'_ansible_parsed': Tru
35.221', u'finished': 0, u'results_file': u'/home/sash
changed: [localhost] => (item={u'_ansible_parsed': Tru
35.226', u'finished': 0, u'results_file': u'/home/sash
```

ansible-openshift

```
Running step 5 on primary_master - /usr/share/ansible/openshift-ansible/playbooks/byo/config.yml

PLAY [Verify Ansible version is greater than or equal to 2.1.0.0] **************

TASK [Verify Ansible version is greater than or equal to 2.1.0.0] **************
skipping: [localhost]

PLAY [Create initial host groups for localhost] ********************************

TASK [include_vars] ************************************************************
ok: [localhost]

TASK [add_host] ****************************************************************
ok: [localhost] => (item=n059.o.internal)
ok: [localhost] => (item=n058.o.internal)
ok: [localhost] => (item=n063.o.internal)
ok: [localhost] => (item=n062.o.internal)
ok: [localhost] => (item=n061.o.internal)
ok: [localhost] => (item=n057.o.internal)
ok: [localhost] => (item=n060.o.internal)

PLAY [Create initial host groups for all hosts] ********************************

TASK [include_vars] ************************************************************
ok: [n058.o.internal]
ok: [n059.o.internal]
ok: [n060.o.internal]
ok: [n061.o.internal]
ok: [n063.o.internal]
ok: [n057.o.internal]
ok: [n062.o.internal]

PLAY [Populate config host groups] *********************************************
```

High availability

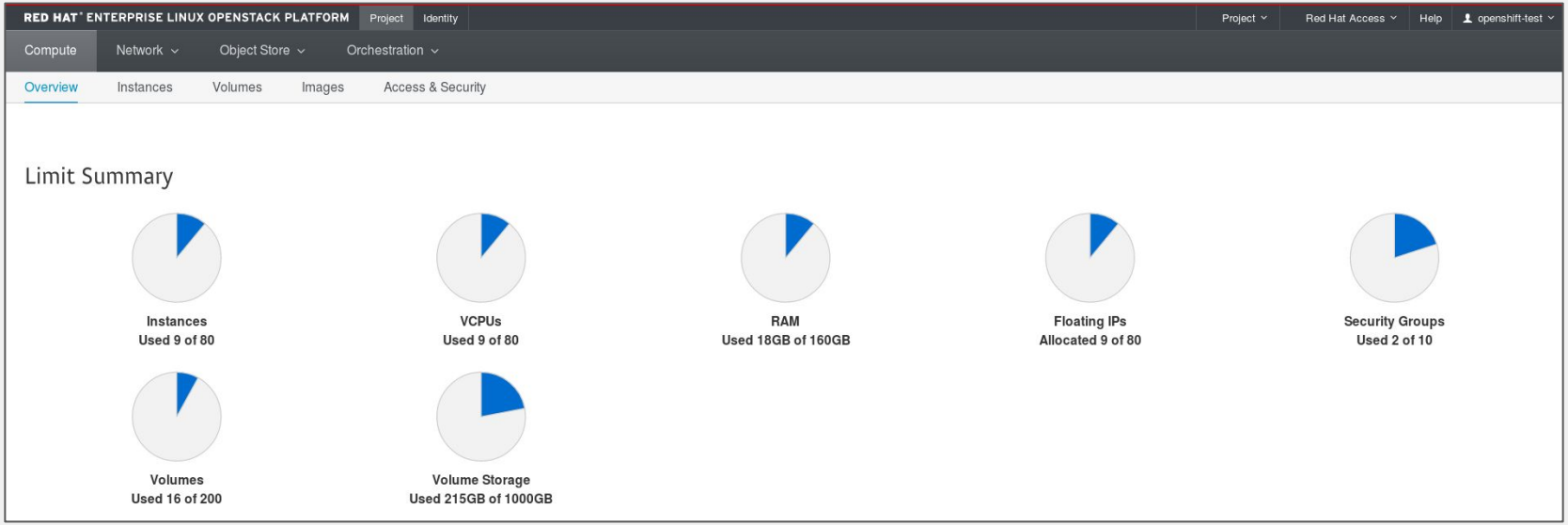
```
- set_fact: node_cluster_name={{ ansible_hostname }}.{{ openshift_clu
- set_fact: openshift_master_public_mask={{ hostvars[node_cluster_name
- set_fact: openshift_master_cluster_public_vip={{ hostvars[node_clust
- set_fact: openshift_master_cluster_mask={{ hostvars[node_cluster_nam
- set_fact: openshift_master_cluster_vip={{ hostvars[node_cluster_name
- set_fact: node_cluster_ip={{ hostvars[node_cluster_name].openshift_

- name: test keepalived multicast iptables exception on masters
  shell: iptables -L -v -n |grep '224.0.0.18'|
  register: iptables_keepalived_masters
  when: openshift_role == "master"
```

```
TASK [setup] ********************************
ok: [n059.testuser.osepool.centralci.eng.b
ok: [n060.testuser.osepool.centralci.eng.b
ok: [n057.testuser.osepool.centralci.eng.b
ok: [n058.testuser.osepool.centralci.eng.b
ok: [n061.testuser.osepool.centralci.eng.b
ok: [n062.testuser.osepool.centralci.eng.b
ok: [n063.testuser.osepool.centralci.eng.b

TASK [keepalived-masters : set_fact] *****
ok: [n057.testuser.osepool.centralci.eng.b
```

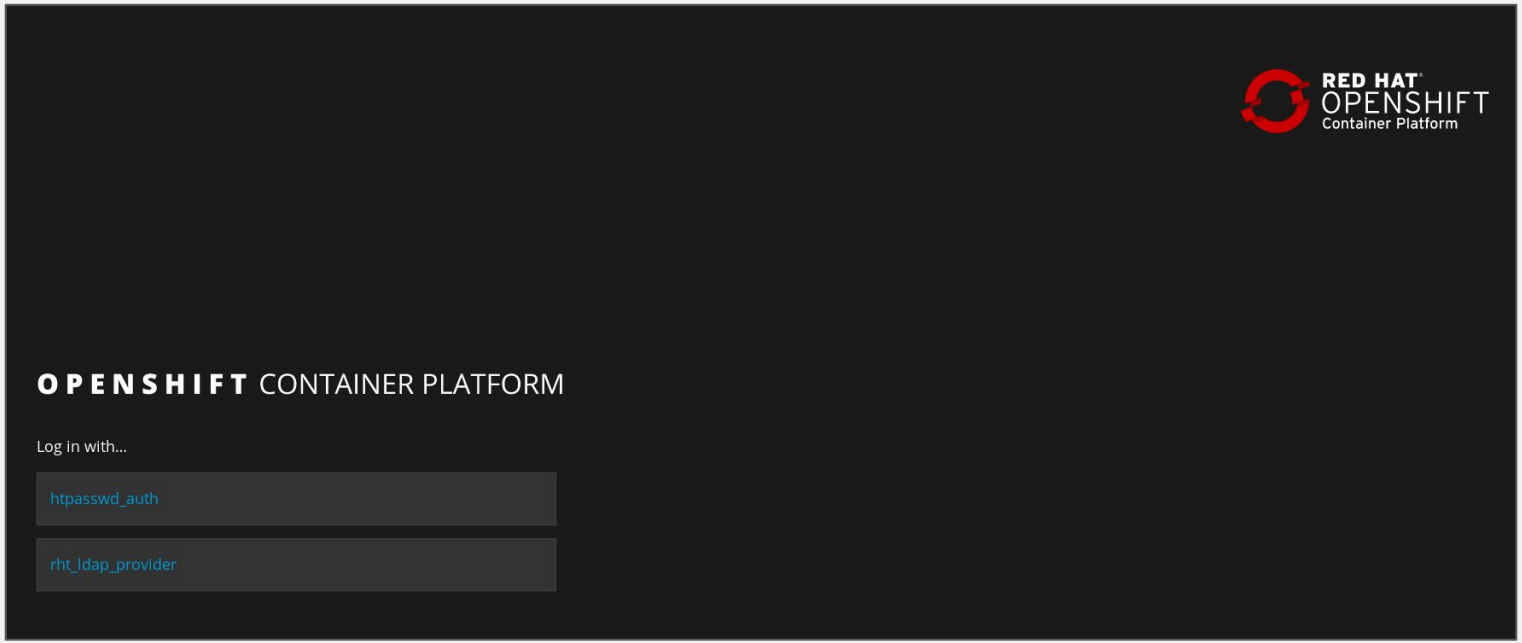
Installation complete

```
ok: [localhost] => {
    "msg": [
        "We successfully finished installation.",
        "The Openshift admin portal is accessible on this url:",
        "https://osemaster.globalci.osepool.centralci.eng.bos.redhat.com:8443/console/",
        "username: newadmin",
        "password: redhat",
        "You can access the cluster nodes directly via ssh:",
        "n057.globalci.osepool.centralci.eng.bos.redhat.com",
        "n058.globalci.osepool.centralci.eng.bos.redhat.com",
        "n059.globalci.osepool.centralci.eng.bos.redhat.com",
        "n060.globalci.osepool.centralci.eng.bos.redhat.com",
        "n061.globalci.osepool.centralci.eng.bos.redhat.com",
        "n062.globalci.osepool.centralci.eng.bos.redhat.com",
        "n063.globalci.osepool.centralci.eng.bos.redhat.com",
        ""
    ]
}
```

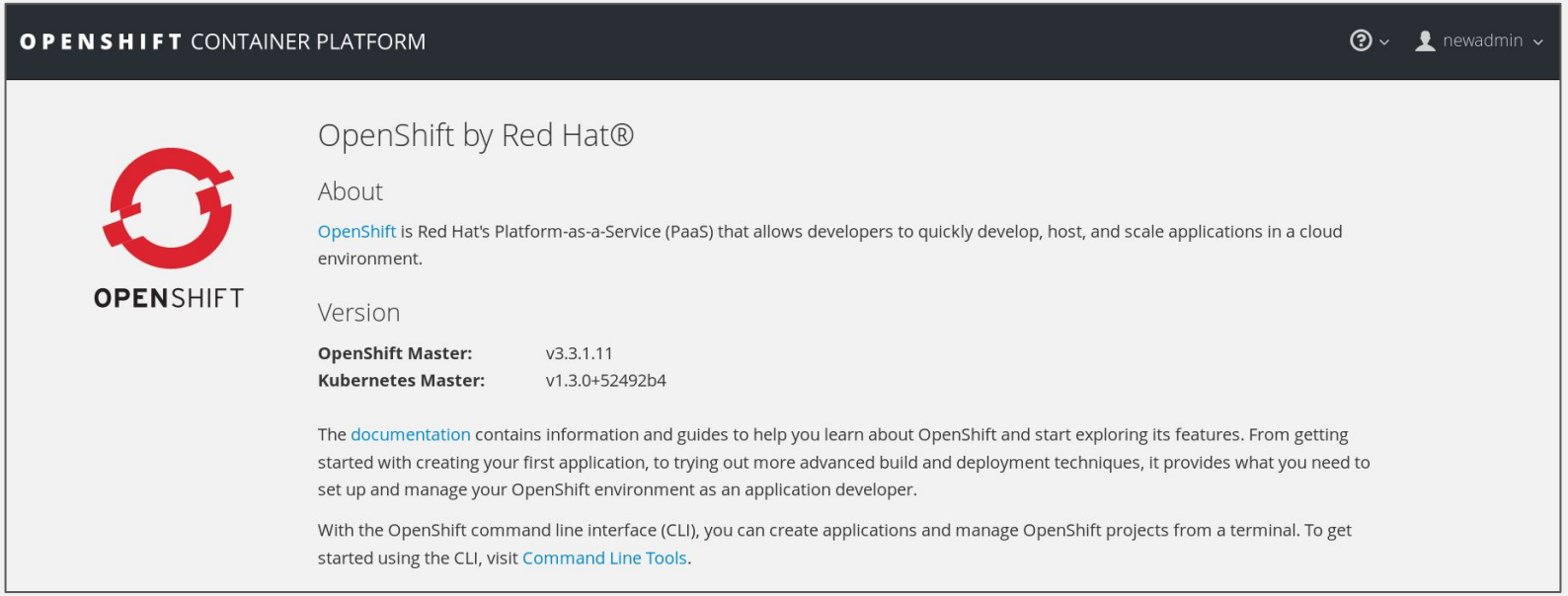
Installation complete



OCP


RED HAT
OPENSHIFT
Container Platform
OPENSHIFT CONTAINER PLATFORM
Log in with...
htpasswd_auth
rht_ldap_provider

OCP


OPENSHIFT CONTAINER PLATFORM
newadmin
OPENSHIFT
OpenShift by Red Hat®
About
OpenShift is Red Hat's Platform-as-a-Service (PaaS) that allows developers to quickly develop, host, and scale applications in a cloud environment.
Version
OpenShift Master: v3.3.1.11
Kubernetes Master: v1.3.0+52492b4
The documentation contains information and guides to help you learn about OpenShift and start exploring its features. From getting started with creating your first application, to trying out more advanced build and deployment techniques, it provides what you need to set up and manage your OpenShift environment as an application developer.
With the OpenShift command line interface (CLI), you can create applications and manage OpenShift projects from a terminal. To get started using the CLI, visit Command Line Tools.

OCP


OPENSHIFT CONTAINER PLATFORM
newadmin
Projects
New Project
default
kube-system
logging
management-infra
Management Infrastructure

OSP cleanup

```
[ose2@fdsa OSOOS]$ bash ansible-openshift/install.sh --clean

PLAY [admin] *******************************************************************

TASK [setup] *******************************************************************
ok: [localhost]

TASK [create_admin_station : Python pip version] *******************************
ok: [localhost]

TASK [create_admin_station : Install python-setuptools] ************************
skipping: [localhost]

TASK [create_admin_station : Download pip] *************************************
skipping: [localhost]

TASK [create_admin_station : Install pip] **************************************
skipping: [localhost]

TASK [create_admin_station : Upgrade pip] **************************************
ok: [localhost]

TASK [create_admin_station : Install shade with pip] ***************************
ok: [localhost]

TASK [create_admin_station : Install python client] ****************************
ok: [localhost]

TASK [clean_tenant : Get info about existing instances] ************************
ok: [localhost]

TASK [clean_tenant : Remove floating ip addresses] *****************************

TASK [clean_tenant : Remove instances] *****************************************

TASK [clean_tenant : Get info about existing volumes] **************************
ok: [localhost]

TASK [clean_tenant : Remove volumes] *******************************************

TASK [clean_tenant : Get info about existing network ports] ********************
```

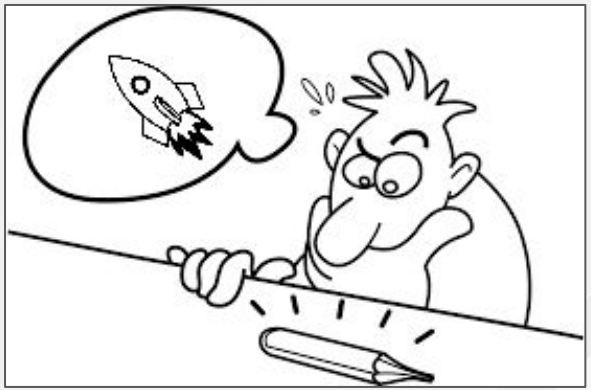
Challenges

- Floating ip addresses
- DNS resolving issue

Future Plans

OCP on OSP - Future Plans

- Work with engineering team
- Add Gluster as our permanent storage
- Make playbook more configurable
- Add support for more DNS providers



RED HAT SUMMIT

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos