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

[Logos for OpenShift, Docker, and Kubernetes]
OSP
Red Hat OpenStack Platform
OCP on OSP - Benefits

Containers, Virtual Machines, and Bare-metal

- OpenShift
- Kubernetes
- Container
- Container
- Service
- VM
- VM
- KVM
- Ironic

OpenStack shared services
- compute
- networking
- storage

standard hardware
OSP - Configuration
OCP - Architecture

- DNS
- LDAP
- Load Balancer
- Bastion Server
- Ansible
- Neutron
- OpenShift Masters (x3)
- OpenStack Container Traffic
- Control Network
- Floating IP Pool
- IP Router
- Floating IP
- Load Balanced Traffic
- Tenant Network
- Cinder
- Cinder Storage
- OpenShift_sdnet (Flannel)
- Infrastructure Nodes (x2)
- Registry (x1)
- Registry Storage
- Docker Storage
- App Nodes (x2+)

Version 0.4
6 Oct 2016
Mark Lamourne
<mlamourne@redhat.com>
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

<table>
<thead>
<tr>
<th>Type</th>
<th>Domain Name</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>developers.ocp.mydomain.com</td>
<td>62.124.XXX.137</td>
</tr>
<tr>
<td>A</td>
<td>*.apps.ocp.mydomain.com</td>
<td>62.124.XXX.137</td>
</tr>
</tbody>
</table>
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

```bash
$ 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
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"]
ssh_key_name="osoos"
additional_servers=["lb-router", "lb-master"]
subs_username="qa@redhat.com"
subs_password="redhatqa"
subs_server_url="subscription.rhsm.stage.redhat.com:443/subscription"
subs_base_url="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_subnet"
Preparing the environment

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

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

- 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
Preparing the environment

<table>
<thead>
<tr>
<th>Security Groups</th>
<th>Key Pairs</th>
<th>Floating IPs</th>
<th>API Access</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>Default security group</td>
</tr>
<tr>
<td>test1</td>
<td></td>
</tr>
<tr>
<td>os00s</td>
<td></td>
</tr>
</tbody>
</table>
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

### Network Topology

<table>
<thead>
<tr>
<th>Name</th>
<th>Subnets Associated</th>
</tr>
</thead>
<tbody>
<tr>
<td>external_net</td>
<td>ext_sub1 10.11.152.0/21</td>
</tr>
<tr>
<td>bianet2</td>
<td>biasubnet2 192.168.1.0/24</td>
</tr>
<tr>
<td>cluster_net</td>
<td>cls_sub1 172.17.0.0/16</td>
</tr>
<tr>
<td>storage_net</td>
<td>st_sub1 172.18.0.0/16</td>
</tr>
<tr>
<td>bianetwork</td>
<td>subnet-bla 10.35.2.0/23</td>
</tr>
</tbody>
</table>
OSP disk

- name: Create volume
  os_volume: state=present availability_zone={{ availability_zone }}
  split(':' ) .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 + '.' + item.item.1|default('') ]"
## OSP disk

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Size</th>
<th>Status</th>
<th>Type</th>
<th>Attached To</th>
</tr>
</thead>
<tbody>
<tr>
<td>lb-master-vda</td>
<td>-</td>
<td>20GB</td>
<td>In-use</td>
<td>-</td>
<td>Attached to lb-master.testuser:</td>
</tr>
<tr>
<td>lb-router-vda</td>
<td>-</td>
<td>20GB</td>
<td>In-use</td>
<td>-</td>
<td>Attached to lb-router.testuser:</td>
</tr>
<tr>
<td>n063-vda</td>
<td>-</td>
<td>20GB</td>
<td>In-use</td>
<td>-</td>
<td>Attached to n063.testuser:</td>
</tr>
<tr>
<td>n063-vdb</td>
<td>-</td>
<td>50GB</td>
<td>In-use</td>
<td>-</td>
<td>Attached to n063.testuser:</td>
</tr>
</tbody>
</table>
OSP instances

- 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 for server create: 1
FAILED - RETRYING: TASK: create_nodes : Wait for server create: 2
FAILED - RETRYING: TASK: create_nodes : Wait for server create: 3
FAILED - RETRYING: TASK: create_nodes : Wait for server create: 4
FAILED - RETRYING: TASK: create_nodes : Wait for server create: 5
FAILED - RETRYING: TASK: create_nodes : Wait for server create: 6
FAILED - RETRYING: TASK: create_nodes : Wait for server create: 7
FAILED - RETRYING: TASK: create_nodes : Wait for server create: 8
FAILED - RETRYING: TASK: create_nodes : Wait for server create: 9
FAILED - RETRYING: TASK: create_nodes : Wait for server create: 10
FAILED - RETRYING: TASK: create_nodes : Wait for server create: 11
### OSP instances

<table>
<thead>
<tr>
<th>Instance Name</th>
<th>Image Name</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>lb-master.testuser.osepool.centralci.eng.bos.redhat.com</td>
<td>-</td>
<td>10.11.152.100 Floating IPs: 10.19.235.221</td>
</tr>
<tr>
<td>lb-router.testuser.osepool.centralci.eng.bos.redhat.com</td>
<td>-</td>
<td>10.11.152.101 Floating IPs: 10.19.235.222</td>
</tr>
<tr>
<td>n063.testuser.osepool.centralci.eng.bos.redhat.com</td>
<td>storage_net 172.18.0.63 external_net</td>
<td>10.11.152.63 Floating IPs: 10.19.235.223</td>
</tr>
</tbody>
</table>
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 }}"

- name: Delete hosts file
  file:
    state: absent
    path: "/tmp/{{ r }}-hosts"
High availability

- `set_fact`: node_cluster_name={{ ansible_hostname }}.{% openshift_cluster_name %}
- `set_fact`: openshift_master_public_mask={{ hostvars[node_cluster_name].openshift_cluster_name }.openshift_master_public_mask}
- `set_fact`: openshift_master_cluster_public_vip={{ hostvars[node_cluster_name].openshift_cluster_name }.openshift_master_cluster_public_vip}
- `set_fact`: openshift_master_cluster_mask={{ hostvars[node_cluster_name].openshift_cluster_name }.openshift_master_cluster_mask}
- `set_fact`: openshift_master_cluster_vip={{ hostvars[node_cluster_name].openshift_cluster_name }.open_shift_master_cluster_vip}
- `set_fact`: node_cluster_ip={{ hostvars[node_cluster_name].open_shift_cluster_name }.open_shift_cluster_name}.open_shift_cluster_ip

- `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"
Installation complete

```json
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

Log in with...

- htpasswd_auth
- rht_idap_provider
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.
OSP cleanup

[ose2@fdsa 05005]$ 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