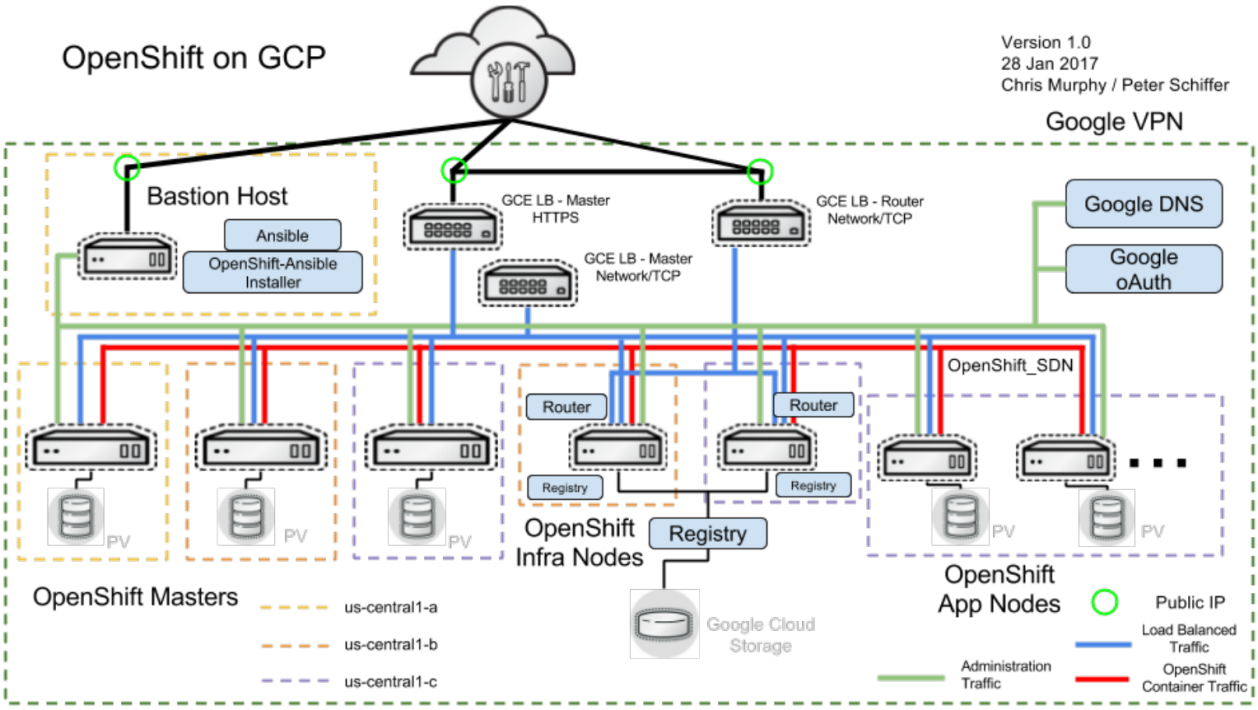
Deploying OpenShift Container Platform 3 on Google Cloud Platform

OpenShift is an open source hybrid cloud application Platform as a Service (PaaS) developed by Red Hat. Red Hat also offers a version of OpenShift for private clouds called OpenShift Enterprise.

OpenShift uses Red Hat Enterprise Linux (RHEL) and its SELinux(Security-Enhanced Linux) subsystem as its foundation. OpenShift supports multiple languages for ease of development, including Java, PHP, Python, Ruby, Perl and Node.js.



Cluster Requirement (min):

1. Master ----1
2. Infra-------1
3. Nodes----- 3

Pre-requisites We should know:

1. Google Computing services, load Balancers, DNS, Firewall rules
2. Intermediate knowledge about Linux and Ansible
3. Exposure on Networking and log analyse Skill

Pre-requisites We need :

1. Google Account with Enabled Project Billing
2. Quota's are needs to Changed based on Required Size of ClusterRequirement
3. Service Account

Cluster Hardware Requirements (min):

1. Master - n1-standard-1
2. infra - n1-standard-1
3. Node2 - n1-standard-32
4.Ansible VM - n1-standard-1

Cluster Software Requirements:

Google Cloud SDK 180.0.1 , Ansible 2.3.x, curl python which tar qemu-img openssl git python-pip pycrypto python-libcloud python2-jmespath java-1.8.0-openjdk-headless httpd-tools python2-passlib.

Step 1: Provision and Access ssh connection (ansible VM)

Ansible vm is used to make actions for the cluster setup.

//install essential packages

```
yum install curl python which tar qemu-img openssl git python-pip pycrypto python-libcloud python2-jmespath java-1.8.0-openjdk-headless httpd-tools python2-passlib && curl "https://bootstrap.pypa.io/get-pip.py" -o "get-pip.py && python get-pip.py

python get-pip.py          //installing pip
Pip -v                     // check pip version
```

Pip install ansible== //ansible versions

```
[root@ansi ch_hareeshrao4839]# pip install ansible==
Collecting ansible==
  Could not find a version that satisfies the requirement ansible== (from versions: 1.0, 1.1, 1.2, 1.2.1, 1.2.2, 1.2.3, 1.3.0, 1.3.1, 1.3.2, 1.3.3, 1.3.4, 1.4, 1.4.1, 1.4.2, 1.4.3, 1.4.4, 1.4.5, 1.5, 1.5.1, 1.5.2, 1.5.3, 1.5.4, 1.5.5, 1.6, 1.6.1, 1.6.2, 1.6.3, 1.6.4, 1.6.5, 1.6.6, 1.6.7, 1.6.8, 1.6.9, 1.6.10, 1.7, 1.7.1, 1.7.2, 1.8, 1.8.1, 1.8.2, 1.8.3, 1.8.4, 1.9.0.1, 1.9.1, 1.9.2, 1.9.3, 1.9.4, 1.9.5, 1.9.6, 2.0.0.0, 2.0.0.1, 2.0.0.2, 2.0.1.0, 2.0.2.0, 2.1.0.0, 2.1.1.0, 2.1.2.0, 2.1.3.0, 2.1.4.0, 2.1.5.0, 2.1.6.0, 2.2.0.0, 2.2.1.0, 2.2.2.0, 2.2.3.0, 2.3.0.0, 2.3.1.0, 2.3.2.0, 2.4.0.0, 2.4.1.0)
No matching distribution found for ansible==
```

pip install ansible==2.3.0.0 //install ansible 2.3.0

//Download openshift ansible Git repo

```
$ git clone https://github.com/openshift/openshift-ansible-contrib.git
$ cd openshift-ansible-contrib/reference-architecture/gcp
$ cp config.yaml.example config.yaml
```

Edit and modif y some changes in Config.yaml like domainName, GCP serviceID,key, Cluster node count and type, Openshift deployment type

$vi config.yaml

if you Want Openshift container platform Enterprise please mention RHSM Credential ,Other wise Cluster every VM launched on Centos and POC is Openshift Origin Cluster is created

```
# Username and password for Red Hat Customer Portal
rhsm_user: 'user@example.com'
rhsm_password: 'xxx'
# Pool name which shall be used to register the instances
rhsm_pool: 'OpenShift Enterprise Broker Infrastructure'
# Path to a RHEL image on local machine, downloaded from Red Hat Customer Portal
rhel_image_path: '~/Downloads/rhel-server-7.4-x86_64-kvm.qcow2'
# Choose to delete or retain the clean image during teardown
delete_image: false
```

Please mention GCP Project and which zone you wanna Deploy cluster

```
# Project ID and main zone settings for Google Cloud
```

```
gcloud_project: 'project-1'
gcloud_zone: 'us-central1-a'
```

Please mention Registered Domain name

```
# Public DNS domain which will be configured in Google Cloud DNS
public_hosted_zone: 'ocp.example.com'
# Public DNS name for the Master service
openshift_master_cluster_public_hostname: 'master.{{ public_hosted_zone }}'
# Internal DNS name for the Master service
openshift_master_cluster_hostname: 'internal-master.{{ public_hosted_zone }}'
# Domain name for the OpenShift applications
wildcard_zone: 'apps.{{ public_hosted_zone }}'
```

If you have any ssl Cert’s , please provide file path. Otherwise Automatically generated Selfsigned cert&attached to openshift Webconsole

```
# Paths on the local file system for the certificate files. If empty, self-signed
# certificate will be generated
master_https_key_file: ''
master_https_cert_file: ''
```

Required cluster vm’s count and type please mention

```
# How many instances should be created for each group
master_instance_group_size: 3
infra_node_instance_group_size: 3
node_instance_group_size: 3

# Machine types
bastion_machine_type: g1-small
master_machine_type: n1-standard-4
infra_machine_type: n1-standard-2
node_machine_type: n1-standard-2
```

```
```

If you provided RHSM credentials ,your deployment type Openshift-enterprise otherwise mention Origin

```
# OpenShift deployment type
# Use 'origin' to deploy OpenShift Origin on top of CentOS 7
openshift_deployment_type: openshift-enterprise
# Deploy OpenShift within containers instead of RPMs?
containerized: false

# OpenShift SDN selection
# options are 'redhat/openshift-ovs-multitenant', 'redhat/openshift-ovs-subnet'
os_sdn_network_plugin_name: 'redhat/openshift-ovs-multitenant'

# Deploy OpenShift Metrics
```

If you want by default metrics expose we need to change status from false to true

```
# Deploy OpenShift Metrics
openshift_hosted_metrics_deploy: false
```

```
openshift_hosted_metrics_storage_volume_size: 20Gi
```

Mention gcp account service ClientID and Secreat key.(we need to create Oauth account for that gcp Account)and If you any domain name people able access to your GUI mention Empty hostedDomain: ‘’ or you want specify particular domain people only loginable , mention mail domainName like hostDomain: ‘harrystuff.tech”

```
platform/latest/install_config/configuring_authentication.html#identity-providers-ansible
openshift_master_identity_providers:
- name: 'google'
  kind: 'GoogleIdentityProvider'
  login: true
  challenge: false
  mapping_method: 'claim'
  clientID: 'xxx-yyy.apps.googleusercontent.com'
  clientSecret: 'zzz'
  hostedDomain: 'example.com'
```

Host Disk Soze specification

```
# Disk sizes in GB
bastion_disk_size: 20
master_boot_disk_size: 45
master_docker_disk_size: 25
infra_boot_disk_size: 25
infra_docker_disk_size: 25
infra_openshift_disk_size: 50
node_boot_disk_size: 25
node_docker_disk_size: 25
node_openshift_disk_size: 50
```

Hosted on specific Vpc,and subnet

```
# Custom VPC Subnet, example value: '10.160.0.0/20'
# Default value is empty, when random subnet in form of 10.x.0.0/20 will be used
gce_vpc_custom_subnet_cidr: ''
```

Then Changed After , Create service Account on GCP

and run the Ansible Script for Cluster Provisioning .it takes more than one hour.

./ocp-on-gcp.sh -vvvv //v - verbrose

Once Script running is over , check status of the cluster like

1. Check Load Balancers Health Status
2. Dns Records
3.Firewall rules and tags

Openshift Cluster oprations:

initially we need check Where is admin.kubeconfig (its contains cluster info, certs info , secret key info and client Authentication and authentication keys info)

#######export kubeconfig file to variable######

```
export KUBECONFIG=/etc/origin/master/admin.kubeconfig
```

####set one user as a ClusterAdmin role####

```
oadm policy add-cluster-role-to-user cluster-admin hariesh4839@gmail.com      or


oadm policy add-cluster-role-to-user cluster-admin bsmith –config=/etc/origin/master/admin.kubeconfig
```

#####login to Cluster via CLI######
oc login https://master.harrystuff.tech –token=<hidden>

oc project <projectname> ###enter into the projects
oc projects ###check the all projects
oc stutus ###check cluster stutus
oc get nodes ###check Cluster nodes
oc get pods ###check all pods in current namespace
oc node <node name> ###To only list information about a single node
oc describe node <nodename> ###To get more detailed information about a specific node
oc delete node <node> ### Delete the node object

oc get pods –all-namespaces ###check all namespaces

oc get pods -n appdeployment ###check pods in specified namespace
oadm policy ###list of all available roles
oadm policy add-role-to-user <role> <user_name> ###Adding a Role to a User
oadm policy remove-role-from-user <role> <user_name>### Removing a Role from a User
oadm policy add-cluster-role-to-user <role> <user_name> ###adding cluster role
oadm policy remove-cluster-role-from-user <role> <user_name> ### removing cluster Role
oadm policy add-role-to-group <role> <groupname> ###Adding a Role to a Group
oadm policy remove-role-from-group <role> <groupname> ###Removing a Role from a Group
oadm policy add-cluster-role-to-group <role> <groupname> Adding a Cluster Role to a Group for All Prgs
oadm policy remove-cluster-role-from-group <role> <groupname> Removing a Cluster Role from a Group for All Projects
oadm policy add-role-to-user admin <user_name> -n <project_name>Creating an Administrator Within a Project
oadm policy add-cluster-role-to-user cluster-admin <user_name> Creating a Cluster Administrator

User Mangement:
oc get user ###To get the current list of users
oc get identity ###current list of identities
oc delete user ###To delete a user