GCP KUBO Deploying Guide

Abhilash S

㈜크로센트
2018. 06

CROSSENT

Agenda

1. GCP Infrastructure Setting
2. Deploying Bosh Director on GCP
3. Deploy LB in GCP
4. Deploying KUBO on GCP
5. KUBO Deployment Architecture on GCP

Note: This deployment Guide is targeted only Ubuntu 16.04 users.

1. GCP Infrastructure Setting (1/11)

- Install Pre-Requirements (1/4)

Note: Install and Configure below dependencies in your Notebook.

❖ Install gcloud cli on Ubuntu.

```
# export CLOUD_SDK_REPO="cloud-sdk-$(lsb_release -c -s)"
# echo "deb http://packages.cloud.google.com/apt $CLOUD_SDK_REPO main" | sudo tee -a /etc/apt/sour
ces.list.d/google-cloud-sdk.list
# curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
# sudo apt-get update && sudo apt-get install google-cloud-sdk
```

❖ Gcloud Login Credentials.

```
Email: abhisr.kubo@gmail.com
Password: Please Check your email
```

❖ Target your Gcloud Account.

```
$ gcloud init
```

Note: In Below Output Verfication code will change at the time of your login.

Note: Select kubernetes-207123 project as shown in below output.

1. GCP Infrastructure Setting (2/11)

- Install Pre-Requirements (2/4)

Output

```
root@DESKTOP-2R196QV:/home/abhisr/workspace# gcloud init
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic (1/1 checks) passed.

You must log in to continue. Would you like to log in (Y/n)?  y

Go to the following link in your browser:

    https://accounts.google.com/o/oauth2/auth?redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob&prompt=select_account&response_type=code&client_id=32555940559.apps.googleusercontent.com&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&access_type=offline


Enter verification code: 4/AABAV1h5B6oKl5sio-WoGvsngPjKUBzVCGTRTQVL1TGTs_f2cj39GDA
You are logged in as: [abhisr.kubo@gmail.com].

Pick cloud project to use:
 [1] kubernetes-207123
 [2] Create a new project
Please enter numeric choice or text value (must exactly match list
item):  1

Your current project has been set to: [kubernetes-207123].
```

❖ Select the Region.

```
Do you want to configure a default Compute Region and Zone? (Y/n)?  y

Which Google Compute Engine zone would you like to use as project
default?
If you do not specify a zone via a command line flag while working
with Compute Engine resources, the default is assumed.
 [32] asia-northeast1-b
 [33] asia-northeast1-c
 [34] asia-northeast1-a
```

Note: Select [34] asia-northeast1-a as a default zone.

```
Please enter numeric choice or text value (must exactly match list
item):  34

Your project default Compute Engine zone has been set to [asia-northeast1-a].
You can change it by running [gcloud config set compute/zone NAME].

Your project default Compute Engine region has been set to [asia-northeast1].
You can change it by running [gcloud config set compute/region NAME].

Created a default .boto configuration file at [/root/.boto]. See this file and
[https://cloud.google.com/storage/docs/gsutil/commands/config] for more
information about configuring Google Cloud Storage.
Your Google Cloud SDK is configured and ready to use!

* Commands that require authentication will use abhisr.kubo@gmail.com by default
* Commands will reference project `kubernetes-207123` by default
* Compute Engine commands will use region `asia-northeast1` by default
* Compute Engine commands will use zone `asia-northeast1-a` by default

Run `gcloud help config` to learn how to change individual settings

This gcloud configuration is called [default]. You can create additional configurations if you work with multiple accounts and/or projects.
Run `gcloud topic configurations` to learn more.

Some things to try next:

* Run `gcloud --help` to see the Cloud Platform services you can interact with. And run `gcloud help COMMAND` to get help on any gcloud command.
* Run `gcloud topic -h` to learn about advanced features of the SDK like arg files and output formatting
```

1. GCP Infrastructure Setting (3/11)

- Install Pre-Requirements (3/4)

❖ Install Terraform CLI.

```
$ wget https://releases.hashicorp.com/terraform/0.11.7/terraform_0.11.7_linux_amd64.zip
$ unzip terraform_0.11.7_linux_amd64.zip
$ chmod +x terraform
$ sudo mv terraform /usr/local/bin/terraform
```

❖ Cloning Kubo-Deployment.

```
$ cd ~/workspace && mkdir gcp && cd ~/workspace/gcp
$ git clone https://github.com/cloudfoundry-incubator/kubo-deployment.git -b v0.17.0
$ cd kubo-deployment/docs/terraform/gcp/platform
```

❖ Edit kubo-infrastructure.tf file to deploy inception vm.

```
$ vi kubo-infrastructure.tf
```

Before Editing

```
178  resource "google_compute_instance" "bosh-bastion" {
179    name         = "${var.prefix}-bosh-bastion"
180    machine_type = "n1-standard-1"
181    zone         = "${var.zone}"
```

After Editing

```
178  resource "google_compute_instance" "bosh-bastion" {
179    name         = "${var.prefix}-bosh-bastion"
180    machine_type = "g1-small"
181    zone         = "${var.zone}"
```

1. GCP Infrastructure Setting (4/11)

- Install Pre-Requirements (4/4)

❖ Edit kubo-infrastructure.tf file to deploy nat vm.

```
$ vi kubo-infrastructure.tf
```

Before Editing

```
314 resource "google_compute_instance" "nat-instance-private-with-nat-primary" {
315   name         = "${var.prefix}-nat-instance-primary"
316   machine_type = "n1-standard-1"
317   zone         = "${var.zone}"
```

After Editing

```
314 resource "google_compute_instance" "nat-instance-private-with-nat-primary" {
315   name         = "${var.prefix}-nat-instance-primary"
316   machine_type = "g1-small"
317   zone         = "${var.zone}"
```

❖ Install Docker on Ubuntu 16.04 .

```
$ sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add - $
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
$ sudo apt-get update
$ sudo apt-get install -y docker-ce
$ sudo docker ps -a
```

1. GCP Infrastructure Setting (5/11)

- Enable and Create API's & VPC & Service Accounts in GCP Portal (1/4)

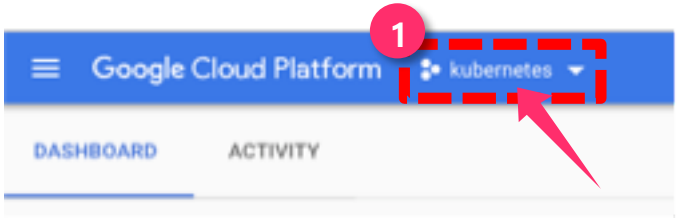
❖ GCP Portal URL.

$ https://cloud.google.com/

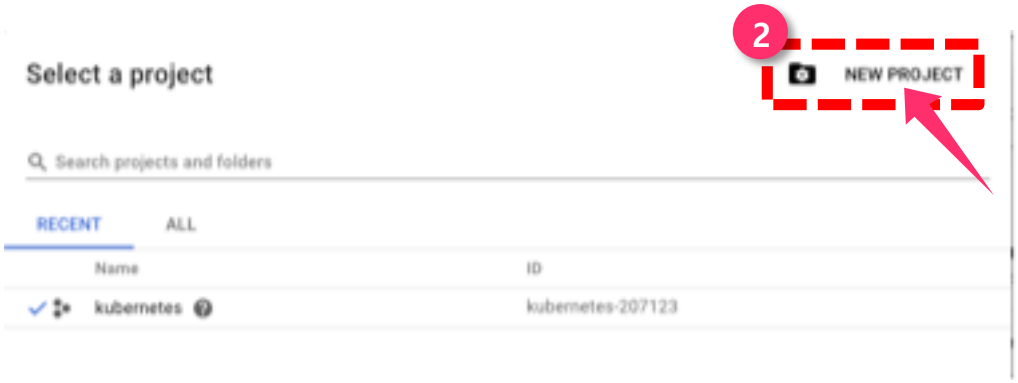
❖ Gcloud Login Credentials.

Email: abhisr.kubo@gmail.com
Password: Please Check your email

❖ Select You Project.



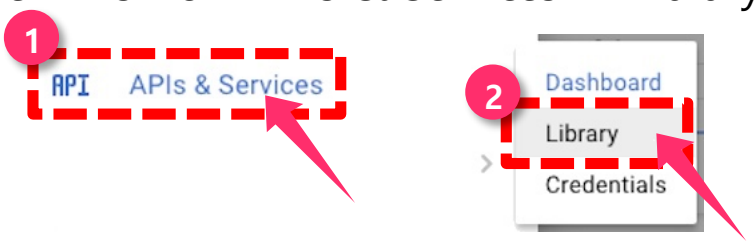
❖ To Create Your Project.



1. GCP Infrastructure Setting (6/11)

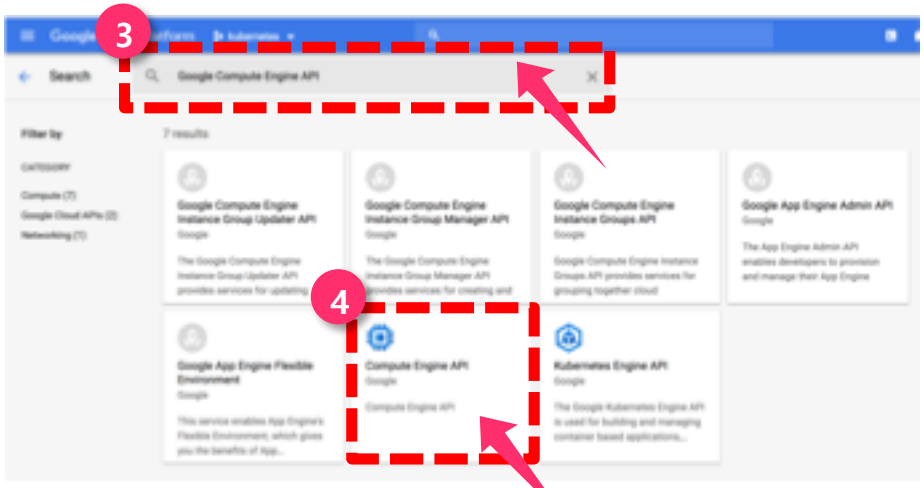
- Enable and Create API's & VPC & Service Accounts in GCP Portal (2/4)

❖ Enable API's from APIs & Services → Library .

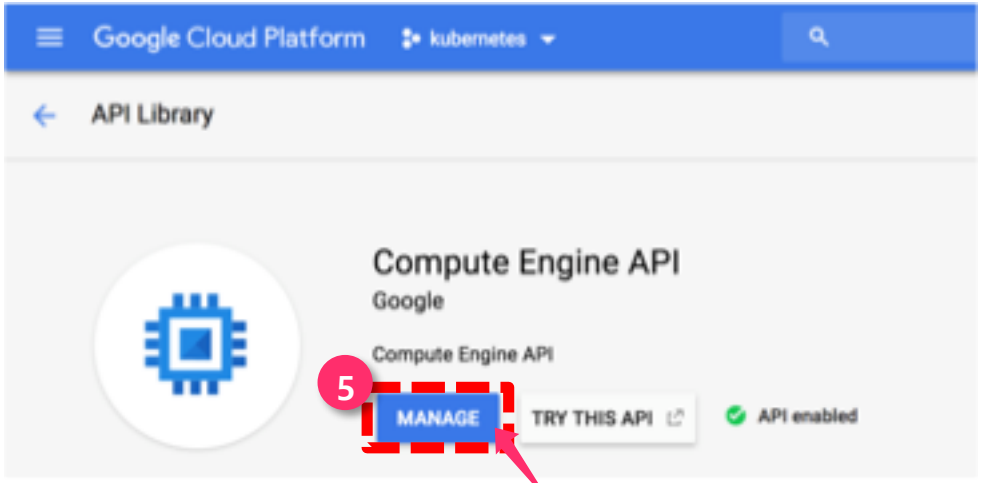


❖ Search and Enable below API's.

1) Google Compute Engine API
2) Google Cloud Resource Manager API
3) Google Cloud Resource Manager API V2
4) Google Identity and Access Management



Note: Enable Remaining three API's in same process.



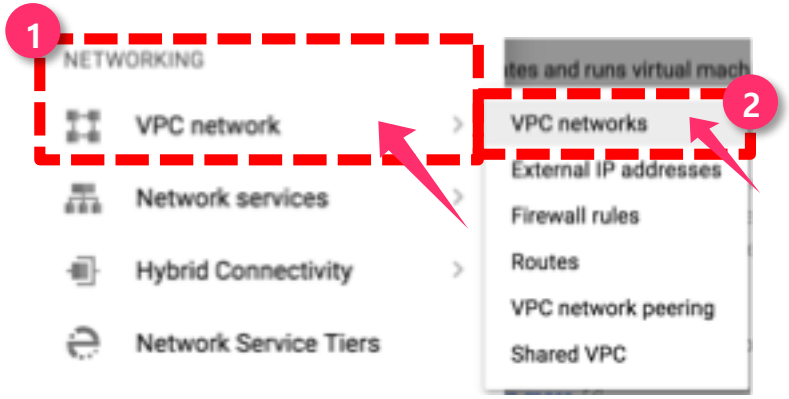
Note: If API is not Enable, You can see **Enable** Button instead of **Manage**.

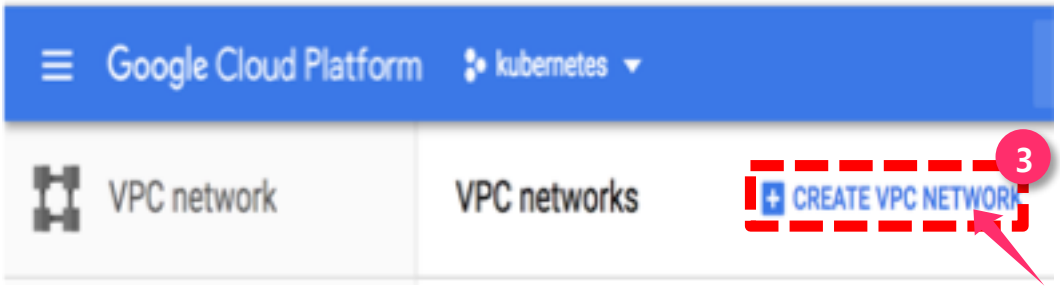
CROSSENT

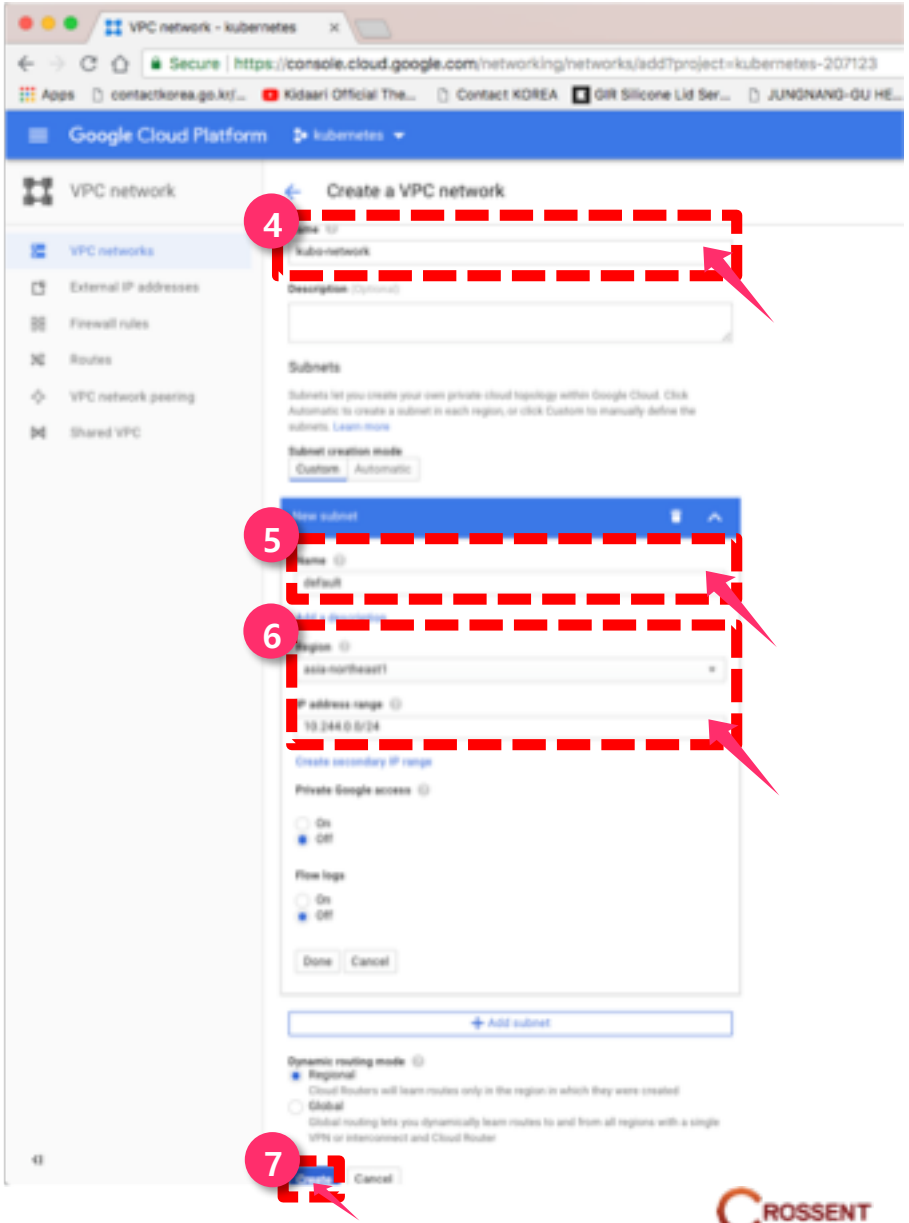
1. GCP Infrastructure Setting (7/11)

- Enable and Create API's & VPC & Service Accounts in GCP Portal (3/4)

❖ Create VPC in GCP Portal.



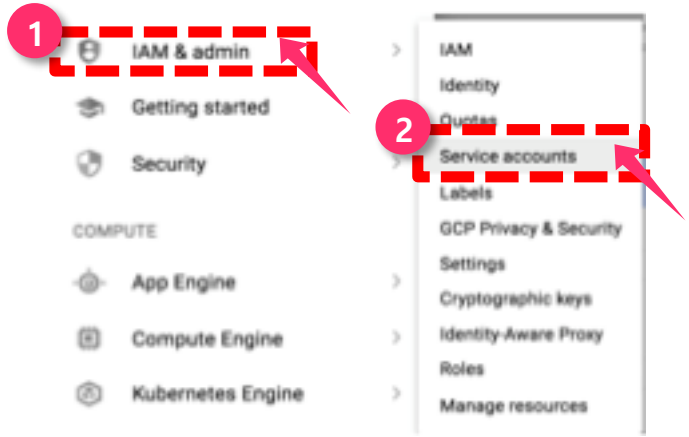


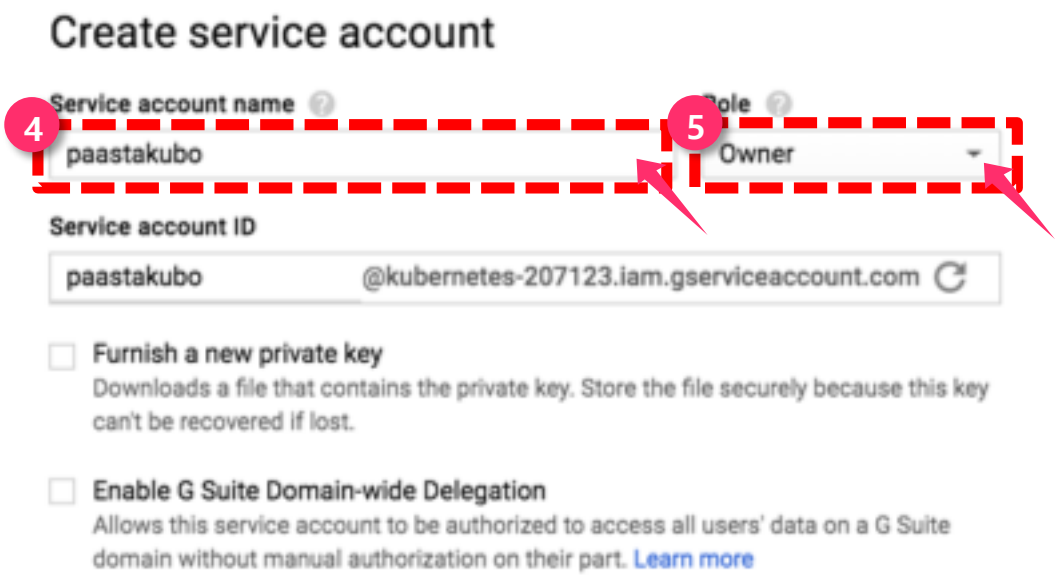


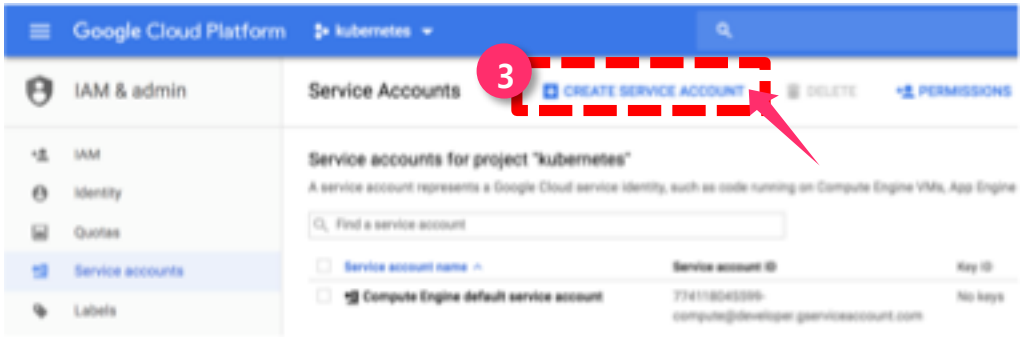
1. GCP Infrastructure Setting (8/11)

- Enable and Create API's & VPC & Service Accounts in GCP Portal (4/4)

❖ Create GCP Service Account.







1. GCP Infrastructure Setting (9/11)

- Deploying Inception & Nat Gateway VM's (1/3)

❖ Move to your kubo-deployment directory.

```
$ cd ~/workspace/gcp/kubo-deployment/docs/terraform/gcp/platform
```

❖ Create a service account key.

```
$ gcloud iam service-accounts keys create ~/workspace/gcp/cfcr-tf.key.json --iam-account paastakub
o@kubernetes-207123.iam.gserviceaccount.com
```

Note: Get your vpc name, project-id, service_account_email, region, zone created in previous sections.

❖ Export an environment variable to deploy GCP Resources.

```
$ export network=kubo-network
$ export subnet_ip_prefix="10.0.1"
$ export project_id=kubernetes-207123
$ export region=asia-northeast1
$ export zone=asia-northeast1-a
$ export service_account_email=paastakubo@kubernetes-207123.iam.gserviceaccount.com
$ gcloud config set compute/zone asia-northeast1-a
$ gcloud config set compute/region asia-northeast1
$ export GOOGLE_CREDENTIALS=$(cat ~/workspace/gcp/cfcr-tf.key.json)
$ export GOOGLE_APPLICATION_CREDENTIALS= ~/workspace/gcp/cfcr-tf.key.json
```

1. GCP Infrastructure Setting (10/11)

- Deploying Inception & Nat Gateway VM's (2/3)

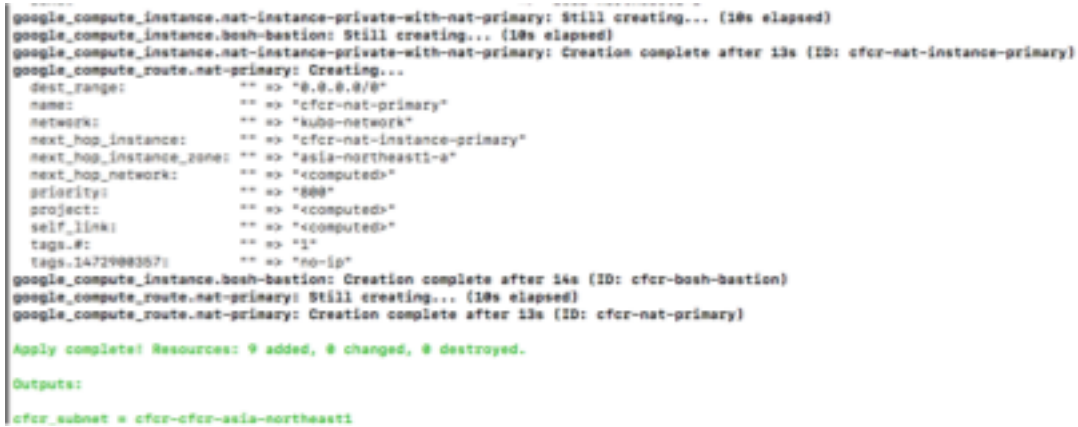
❖ Initialize the Terraform cloud provider.

```
$ sudo docker run -i -t -v $(pwd):/$(basename $(pwd)) -w /$(basename $(pwd)) hashicorp/terraform:light init
```

❖ Create the inception, nat gateway and other resources

```
$ sudo docker run -i -t -e CHECKPOINT_DISABLE=1 -e "GOOGLE_CREDENTIALS=${GOOGLE_CREDENTIALS}" -v $(pwd):/$(basename $(pwd)) -w /$(basename $(pwd)) hashicorp/terraform:light apply -var service_account_email=${service_account_email} -var projectid=${project_id} -var network=${network} -var region=${region} -var prefix=${prefix:-cfcr} -var zone=${zone} -var subnet_ip_prefix=${subnet_ip_prefix} -state /$(basename $(pwd))/${prefix:-cfcr}.tfstate
```

Output



```
google_compute_instance.nat-instance-private-with-nat-primary: Still creating... (10s elapsed)
google_compute_instance.bosh-bastion: Still creating... (10s elapsed)
google_compute_instance.nat-instance-private-with-nat-primary: Creation complete after 13s (ID: cfcr-nat-instance-primary)
google_compute_route.nat-primary: Creating...
  dest_range:             "" => "0.0.0.0/0"
  name:                   "" => "cfcr-nat-primary"
  network:                "" => "kubo-network"
  next_hop_instance:      "" => "cfcr-nat-instance-primary"
  next_hop_instance_zone: "" => "asia-northeast1-a"
  next_hop_network:       "" => "<computed>"
  priority:               "" => "800"
  project:                "" => "<computed>"
  self_link:              "" => "<computed>"
  tags.#:                 "" => "1"
  tags.1472908357:        "" => "no-ip"
google_compute_instance.bosh-bastion: Creation complete after 14s (ID: cfcr-bosh-bastion)
google_compute_route.nat-primary: Still creating... (10s elapsed)
google_compute_route.nat-primary: Creation complete after 13s (ID: cfcr-nat-primary)

Apply complete! Resources: 9 added, 0 changed, 0 destroyed.

Outputs:

cfcr_subnet = cfcr-cfcr-asia-northeast1
```

❖ Copy the service account key to the newly created inception VM

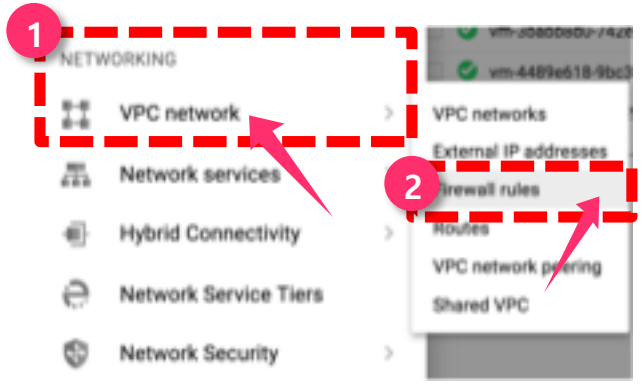
```
$ gcloud compute scp ~/cfcr-tf.key.json cfcr-bosh-bastion:~/cfcr-tf.key.json
```

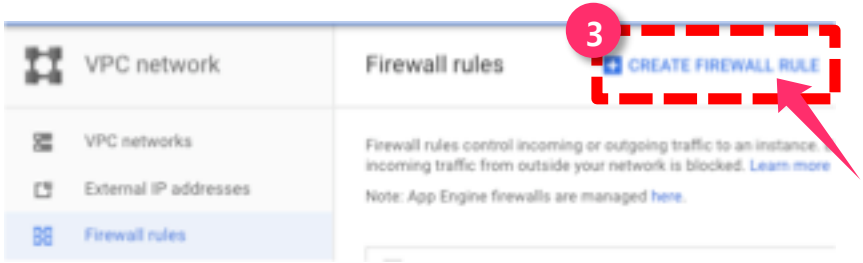
1. GCP Infrastructure Setting (11/11)

- Deploying Inception & Nat Gateway VM's (3/3)

Note: Perfrom below steps in GCP Portal.

❖ Create Firewall Rules in GCP Portal.





2. Deploying Bosh Director on GCP (1/5)

- Deploying Bosh-Director (1/5)

Note: Perfrom below steps in your Inception.

❖ ssh into inception

```
$ gcloud compute ssh cfcr-bosh-bastion
$ mkdir ~/workspace && cd ~/workspace
```

❖ Next clone the bosh-deployment and kubo-deployment repositories

```
$ git clone https://github.com/cloudfoundry/bosh-deployment.git
$ git clone https://github.com/cloudfoundry-incubator/kubo-deployment.git -b v0.17.0
```

❖ Edit GCP cpi.yml file to configure Nat VM.

```
$ vi ~/workspace/bosh-deployment/gcp/cpi.yml
```

Before Editing

```
- type: replace
  path: /networks/name=default/subnets/0/cloud_properties?
  value:
    network_name: ((network))
    subnetwork_name: ((subnetwork))
    ephemeral_external_ip: false
    tags: ((tags))
```

After Editing

```
- type: replace
  path: /networks/name=default/subnets/0/cloud_properties?
  value:
    network_name: ((network))
    subnetwork_name: ((subnetwork))
    ephemeral_external_ip: false
    tags:
    - no-ip
    - internal
```

Your Nat Gateway VM Tags

2. Deploying Bosh Director on GCP (2/5)

- Deploying Bosh-Director (2/5)

❖ Edit Bosh bosh.yml file to configure Nat VM.

```
$ vi ~/workspace/bosh-deployment/bosh.yml
```

Before Editing

```
disk_pools:
- name: disks
  disk_size: 65_536

networks:
- name: default
  type: manual
  subnets:
  - range: ((internal_cidr))
    gateway: ((internal_gw))
    static: [((internal_ip))]
    dns: [8.8.8.8]
```

After Editing

```
disk_pools:
- name: disks
  disk_size: 40_960

networks:
- name: default
  type: manual
  subnets:
  - range: ((internal_cidr))
    gateway: ((internal_gw))
    static: [((internal_ip))]
    dns: [10.0.1.1]
```

Your Subnet CIDR & Nat Gateway

❖ Create directory to store director deployment state and credentials.

```
$ cd ~/workspace/bosh-deployment/ && mkdir kubo && cd kubo
```

2. Deploying Bosh Director on GCP (3/5)

- Deploying Bosh-Director (3/5)

- Below command create bosh-init vm on virtualbox by means of bosh.yml as base manifest.
- Yaml files with –o command set variables director_name, internal_ip, internal_gw, internal_cidr, outbound_network_name using –v command.
- It also create state.json to record running state and creds.yml **(for certs & credentails)** in your ~/deployments/vbox directory
- Following Command Creates Director's vm and install it's jobs

```
$ bosh create-env ~/workspace/bosh-deployment/bosh.yml
  --state=~/workspace/bosh-deployment/kubo/state.json
  -o ~/workspace/bosh-deployment/gcp/cpi.yml
  -o ~/workspace/bosh-deployment/gcp/service-account.yml
  -o ~/workspace/bosh-deployment/jumpbox-user.yml
  -o ~/workspace/bosh-deployment/uaa.yml
  -o ~/workspace/bosh-deployment/credhub.yml
  -o ~/workspace/bosh-deployment/local-dns.yml
  --vars-store=~/workspace/bosh-deployment/kubo/creds.yml
  -v director_name="kubo"
  -v internal_ip=10.0.1.252
  -v internal_gw=10.0.1.1
  -v internal_cidr=10.0.1.0/24
  --var-file gcp_credentials_json=~/cfcr-tf.key.json
  -v project_id=kubernetes-207123
  -v zone=asia-northeast1-a
  -v network=kubo-vpc
  -v subnetwork=cfcr-cfcr-asia-northeast1
  -v service_account=paastakubo@kubernetes-207123.iam.gserviceaccount.com
```

Check subnet name in GCP Portal

2. Deploying Bosh Director on GCP (4/5)

- Deploying Bosh-Director (4/5)

❖ Log into director with your alias name.

```
$ bosh alias-env kubo -e 10.0.1.252 --ca-cert <(bosh int  ~/workspace/bosh-deployment/kubo/creds.yml --path /director_ssl/ca)

$ export BOSH_CLIENT=admin

$ export BOSH_CLIENT_SECRET=`bosh int ~/workspace/bosh-deployment/kubo/creds.yml --path /admin_password`
```

❖ Create directory to download releases and stemcells

```
$ mkdir ~/workspace/releases && cd ~/workspace/releases
```

❖ Download Upload GCP Stemcell

```
$ wget https://s3.amazonaws.com/bosh-core-stemcells/google/bosh-stemcell-3586.16-google-kvm-ubuntu-trusty-go_agent.tgz
$ bosh -e kubo upload-stemcell bosh-stemcell-3586.16-google-kvm-ubuntu-trusty-go_agent.tgz
```

2. Deploying Bosh Director on GCP (5/5)

- Deploying Bosh-Director (5/5)

❖ Target & Log into director credhub

```
$ export CREDHUB_CLIENT=credhub-admin
$ export CREDHUB_SECRET=$(bosh int --path /credhub_admin_client_secret ~/workspace/bosh-deployment/kubo/creds.yml)
$ export CREDHUB_CA_CERT=$(bosh int --path /credhub_tls/ca ~/workspace/bosh-deployment/kubo/creds.yml)
$ credhub login -s https://10.0.1.252:8844 --skip-tls-validation
```

❖ List out Certificates and Passwords in credhub

```
$ credhub find
```

❖ To Delete Certificates and Passwords in credhub

```
$ credhub delete -n /director_name/deployment-name/certificate-name
```

3. Deploy LB in GCP

- Deploy Master Load Balancer

Note: Perfrom below steps in your Inception.

❖ Install Load Balancer for Kubernetes Master

```
$ cd ~/workspace/kubo-deployment/docs/terraform/gcp/routing
$ terraform init
$ terraform apply -var network=kubo-network -var projectid=kubernetes-207123 -var region=asia-northeast1 -var ip_cidr_range="10.0.1.0/24" -state=~/workspace/terraform.tfstate
```

Output

```
e="${subnet_ip_prefix}.0/24" -state=${kubo_terraform_state}
google_compute_firewall.cfcr-tcp-public: Creating...
  allow.#:                  "" => "1"
  allow.1052118868.ports.#: "" => "1"
  allow.1052118868.ports.0: "" => "8443"
  allow.1052118868.protocol: "" => "tcp"
  name:                     "" => "cfcr-cfcr-tcp-public"
  network:                  "" => "kubo-network"
  project:                  "" => "<computed>"
  self_link:                "" => "<computed>"
  target_tags.#:            "" => "1"
  target_tags.755606486:    "" => "master"
google_compute_target_pool.cfcr-tcp-public: Creating...
  instances.#: "" => "<computed>"
  name:        "" => "cfcr-cfcr-tcp-public"
  project:     "" => "<computed>"
  region:      "" => "asia-northeast1"
  self_link:   "" => "<computed>"
google_compute_address.cfcr-tcp: Creating...
  address:   "" => "<computed>"
  name:      "" => "cfcr-cfcr"
  self_link: "" => "<computed>"
google_compute_firewall.cfcr-tcp-public: Still creating... (10s elapsed)
google_compute_target_pool.cfcr-tcp-public: Still creating... (10s elapsed)
google_compute_address.cfcr-tcp: Still creating... (10s elapsed)
google_compute_target_pool.cfcr-tcp-public: Creation complete
google_compute_address.cfcr-tcp: Creation complete
google_compute_forwarding_rule.cfcr-tcp: Creating...
  ip_address:   "" => "35.200.120.79"
  ip_protocol:  "" => "TCP"
  name:         "" => "cfcr-cfcr-tcp"
  port_range:   "" => "8443"
  project:      "" => "<computed>"
  region:       "" => "<computed>"
  self_link:    "" => "<computed>"
  target:       "" => "https://www.googleapis.com/compute/v1/projects/kubernetes-207123/regions/asia-northeast1/targetPools/cfcr-cfcr-tcp-public"
google_compute_firewall.cfcr-tcp-public: Creation complete
google_compute_forwarding_rule.cfcr-tcp: Still creating... (10s elapsed)
google_compute_forwarding_rule.cfcr-tcp: Creation complete

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

The state of your infrastructure has been saved to the path
below. This state is required to modify and destroy your
infrastructure, so keep it safe. To inspect the complete state
use the `terraform show` command.

State path: /root/kubo-env/kubo/terraform.tfstate

Outputs:

cfcr_master_target_pool = cfcr-cfcr-tcp-public
master_lb_ip_address = 35.200.120.79
```

4. Deploying KUBO on GCP (1/8)

- Deploying Kubo (1/5)

❖ Download kubo-release.

```
$ cd ~/workspace && mkdir releases && cd ~/workspace/releases
$ wget https://github.com/cloudfoundry-incubator/kubo-release/releases/download/v0.17.0/kubo-release-0.17.0.tgz
```

❖ Upload kubo-release

```
$ bosh -e kubo upload-release ~/workspace/releases/kubo-release-0.17.0.tgz
```

❖ Edit kubo-deployment cfcr.yml file for deploying kubernetes master on GCP

```
$ cd ~/workspace/kubo-deployment
$ vi ~/workspace/kubo-deployment/manifests/cfcr.yml
```

Before Editing

```
77  - name: master
78    instances: 3
79    networks:
80    - name: default
81    azs: [z1,z2,z3]
```

After Editing

```
- name: master
  instances: 1
  vm_extensions:
  - kubo
  networks:
  - name: default
  azs: [z1]
```

Your LB Extension Name in Cloud-config

4. Deploying KUBO on GCP (2/8)

- Deploying Kubo (2/5)

❖ Edit kubo-deployment cfcr.yml file for deploying Kubernetes worker-nodes GCP

```
$ vi ~/workspace/kubo-deployment/manifests/cfcr.yml
```

Before Editing

```
155   - name: worker
156     instances: 3
157     networks:
158     - name: default
159     azs: [z1,z2,z3]
```

After Editing

```
- name: worker
  instances: 2
  networks:
  - name: default
  azs: [z1]
```

❖ Edit kubo-deployment cfcr.yml file for deploying kubernetes master certificates

```
$ vi ~/workspace/kubo-deployment/manifests/cfcr.yml
```

Before Editing

```
237 - name: tls-kubernetes
238   type: certificate
239   options:
240     ca: kubo_ca
241     organization: "system:masters"
242     common_name: master.cfcr.internal
243     alternative_names:
244     - 10.100.200.1
245     - kubernetes
246     - kubernetes.default
247     - kubernetes.default.svc
248     - kubernetes.default.svc.cluster.local
249     - master.cfcr.internal
```

After Editing

```
- name: tls-kubernetes
  type: certificate
  options:
    ca: kubo_ca
    organization: "system:masters"
    common_name: 35.200.120.79
    alternative_names:
    - 10.100.200.1
    - kubernetes
    - kubernetes.default
    - kubernetes.default.svc
    - kubernetes.default.svc.cluster.local
    - master.cfcr.internal
    - 35.200.120.79
```

Note: Change 35.200.120.79 IP according to your LB IP.

4. Deploying KUBO on GCP (3/8)

- Deploying Kubo (3/5)

❖ Create and Save bosh director cloud-config

```
$ vi ~/workspace/kubo-deployment/manifests/cloud-config.yml
```

4. Deploying KUBO on GCP (4/8)

- Deploying Kubo (4/5)

❖ Update bosh director cloud-config

```
$ bosh -e kubo update-cloud-config ~/workspace/kubo-deployment/manifests/cloud-config.yml
```

❖ Check the desired releases and stemcell to deploy kubo on GCP bosh

```
$ bosh -e kubo releases
```

Output

```
Using environment '10.0.1.252' as client 'admin'

Name       Version  Commit Hash
bosh-dns   1.5.0*   f5a8d25
bpm        0.6.0*   b6f4675
cfcr-etcd  1.3*     6a62d8f
docker     32.0.0*  542c382
kubo       0.17.0*  ad9ef809

(*) Currently deployed
(+) Uncommitted changes

5 releases

Succeeded
```

```
$ bosh -e kubo stemcells
```

Output

```
Using environment '10.0.1.252' as client 'admin'

Name                                     Version   OS             CPI  CID
bosh-google-kvm-ubuntu-trusty-go_agent   3586.16*  ubuntu-trusty  -    stemcell-a74cd734-2bc8-4366-7ba8-b63a63493b4d

(*) Currently deployed

1 stemcells

Succeeded
```

4. Deploying KUBO on GCP (5/8)

- Deploying Kubo (5/5)

❖ Deploy Kubo on gcp

```
$ bosh -e kubo -d cfcr deploy /home/abhisr/workspace/kubo-deployment/manifests/cfcr.yml -o /ho
me/abhisr/workspace/kubo-deployment/manifests/ops-files/iaas/gcp/cloud-provider.yml -v project_i
d=kubernetes-207123 -v network=kubo-vpc -v director_name=kubo -v deployment_name=cfcr
```

❖ Deploy Kubernetes add-ons

```
$ bosh -e kubo -d cfcr run-errand apply-specs
```

❖ Check Kubernetes Deployment

```
$ bosh -e kubo -d cfcr vms
```

Output

```
Using environment '10.0.1.252' as client 'admin'

Task 141. Done

Deployment 'cfcr'

Instance                                     Process State  AZ  IPs         VM CID                                   VM Type        Active
master/528ffe75-5391-4313-aec6-6c1941a17685  running        z1  10.0.1.101  vm-befdca1f-ebdd-4018-44d8-feb6bc8ab462  small          true
worker/61b51092-6a00-4c74-8c10-e3e6b479487e  running        z1  10.0.1.102  vm-e4827bef-3225-4860-4bfa-efb283438085  small-highmem  true
worker/bd7caa22-39c1-40f9-bc8a-efe3471220ca  running        z1  10.0.1.103  vm-c8582161-1077-41f4-7058-22feba0b4840  small-highmem  true

3 vms

Succeeded
```

❖ ssh into master and work vms

```
$ bosh -e kubo -d cfcr ssh master
```

```
$ bosh -e kubo -d cfcr ssh worker/61b51092-6a00-4c74-8c10-e3e6b479487e
```

4. Deploying KUBO on GCP (6/8)

- Accessing Kubernetes (1/3)

❖ Download tls-kubernetes certificate from credhub

```
$ bosh -e kubo -d cfcr int <(credhub get -n "/kubo/cfcr/tls-kubernetes" --output-json) --path=/value
/ca > ~/workspace/kubo-deployment/kubo/kubernetes.crt
```

❖ Download kubo-admin-password password from credhub

```
$ bosh -e kubo -d cfcr int <(credhub get -n "/kubo/cfcr/kubo-admin-password" --output-json) --pat
h=/value >  ~/workspace/kubo-deployment/kubo/kubernetes_pwd.crt
```

❖ Configure Kubernetes Cluster by using above Certificate (kubernetes.crt)

```
$ kubectl config set-cluster "dev" --server https://you-lb-publib-ip:8443 --embed-certs=true --certific
ate-authority=~/workspace/kubo-deployment/kubo/kubernetes.crt
```

❖ Configure Kubernetes User and Context (Note: Use Kubernets User Password from above kubernetes_pwd.crt)

```
$ kubectl config set-credentials "dev" --token=your-kubernetes_pwd.crt-password
```

```
$ kubectl config set-context "dev" --cluster="dev" --user="dev
```

```
$ kubectl config use-context "dev"
```

4. Deploying KUBO on GCP (7/8)

- Accessing Kubernetes (2/3)

❖ Obtain kubernetes node Information

```
$ kubectl get node -o wide
```

Output

```
NAME                                   STATUS   ROLES    AGE   VERSION   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION      CONTAINER-RUNTIME
vm-c8582161-1077-41f4-7058-22feba084840   Ready    <none>   21h   v1.10.3               Ubuntu 14.04.5 LTS   4.4.0-127-generic   docker://17.12.1-ce
vm-e48278ef-3225-4860-4bfa-efb283430885   Ready    <none>   21h   v1.10.3               Ubuntu 14.04.5 LTS   4.4.0-127-generic   docker://17.12.1-ce
```

❖ Obtain kubernetes Cluster Information

```
$ kubectl cluster-info
```

Output

```
Kubernetes master is running at https://35.200.44.55:8443
Heapster is running at https://35.200.44.55:8443/api/v1/namespaces/kube-system/services/heapster/proxy
KubeDNS is running at https://35.200.44.55:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
monitoring-influxdb is running at https://35.200.44.55:8443/api/v1/namespaces/kube-system/services/monitoring-influxdb/proxy
```

❖ Get all information regarding your Namespace

```
$ kubectl get pods --namespace=kube-system
$ kubectl get all -n kube-system
```

❖ Check 8001 port is active on your notebook, if active kill the process

```
$ sudo lsof -PiTCP -sTCP:LISTEN
$ sudo kill -9 your-pid-number
```

❖ Accessing to Kubernetes Dashboard

```
$ kubectl proxy
  Starting to serve on 127.0.0.1:8001
$ http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/login
```

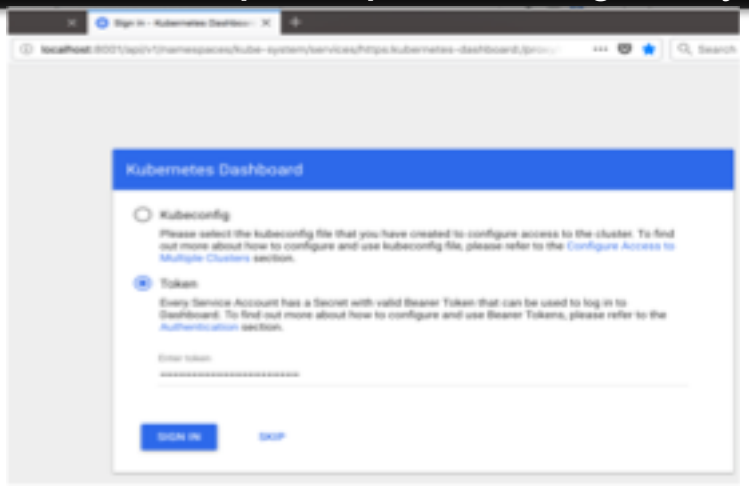
4. Deploying KUBO on GCP (8/8)

- Accessing Kubernetes (3/3)

❖ Accessing Kubernetes Dashboard in your Browser

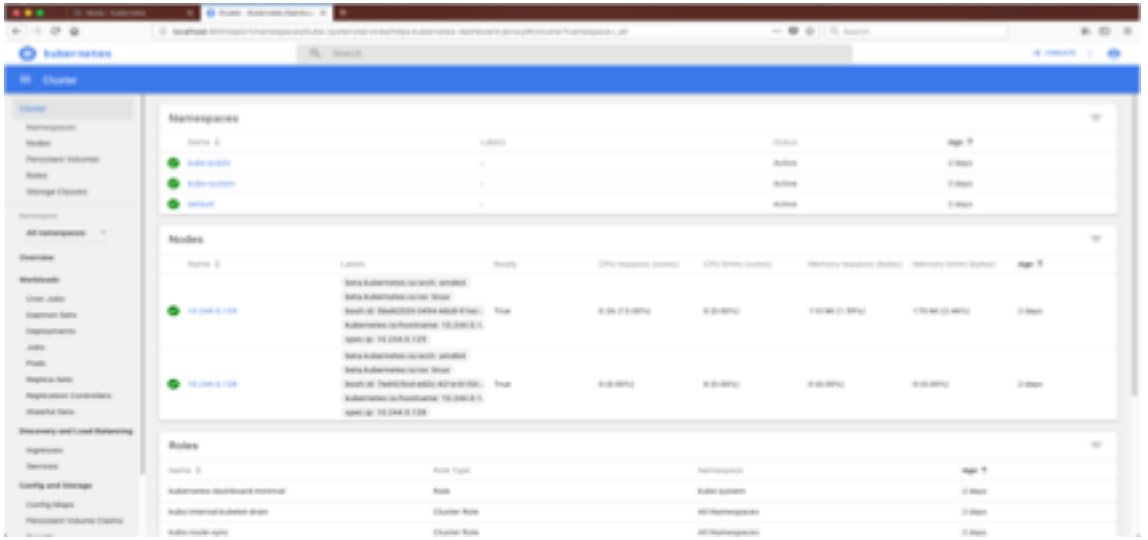
Select token option and paste above kubernetes_pwd.crt password to login into your kubernets cluster

Output



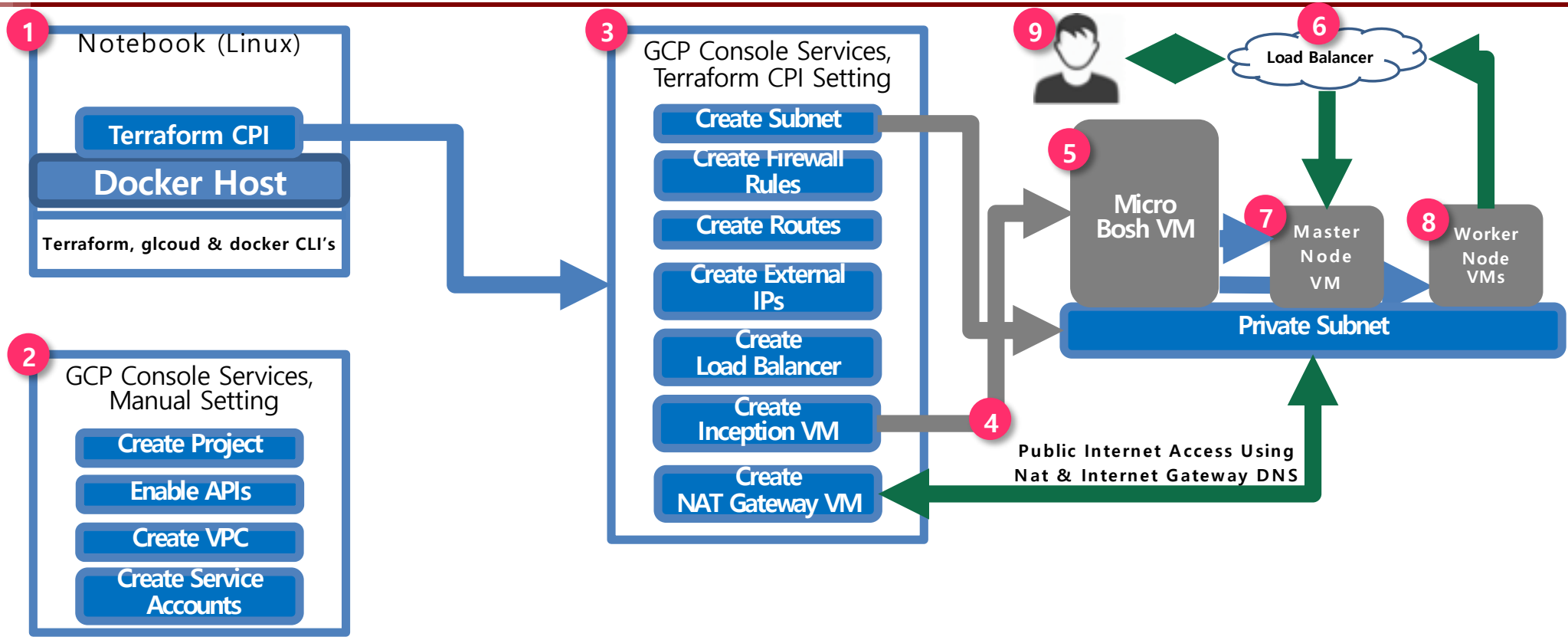
❖ Kubernetes Home Dashboard

Output



5. KUBO Deployment Architecture on GCP

- Deployment Architecture



THANK YOU