



Public release

Matuzalem Muller authored 23 hours ago

34c313b0

README.md 6.01 KB

This documentation presents step by step instructions on how to set up a kubernetes cluster in Virtual Machines (VMs) from Google Cloud Platform (GCP). The cloud infrastructure is launched using Terraform, while the kubernetes cluster is remotely set up using Docker & Rancher. The instructions from this article are for OSX, but are also applicable to Linux hosts and possibly Windows devices.

- Instructions on how to use Terraform to launch VMs are based in: <https://medium.com/@josephbleroy/using-terraform-with-google-cloud-platform-part-1-n-6b5e4074c059>
- Below are the versions of each software used in this project:
 - Terraform v0.11.8
 - RKE v0.1.9
 - Docker v17.03.3
 - kubectl v1.11.0
 - Kubernetes v.v1.11.1

Table of contents

- [Create a project in GCP](#)
- [Set up Terraform in local device](#)
- [Download and install Google SDK](#)
- [Modify terraform-infrastructure.tf file to include correct account information and credentials](#)
- [Run Terraform and create the GCP infrastructure](#)
- [Install docker in all VMs created](#)
- [Install rke in local computer](#)
- [Deploy remote k8s cluster](#)
- [Move k8s local file created by rancher to k8s local configuration folder](#)
- [Install Helm in remote k8s cluster](#)

Create a project in GCP

<https://cloud.google.com/resource-manager/docs/creating-managing-projects>

Set up Terraform in local device

Install Terraform in local device:

```
mkdir /usr/local/terraform
wget https://releases.hashicorp.com/terraform/0.11.8/terraform_0.11.8_darwin_amd64.zip
unzip terraform_0.10.8_darwin_amd64.zip -d /usr/local/terraform
```

Set the PATH variable for Terraform:

```
vim ~/.bash_profile
. . .
# Required by Terraform
export PATH=$PATH:/usr/local/terraform
. . .
```

Restart the console/terminal and confirm that Terraform works:

```
terraform -v
```

Download and install Google SDK

Download and install Google SDK:

```
cd remote-setup
curl https://sdk.cloud.google.com | bash
```

Start the gcloud environment and connect to your account and project:

```
gcloud init
```

Modify terraform-infrastructure.tf file to include correct account information and credentials

Generate local SSH keys, which will be used to connect to the remote VMs. Save both keys (public and private) with default name (id_rsa) and place both keys inside the directory `remote-setup/keys` :

```
ssh-keygen -t rsa -b 4096 -C "email@domain.com"
```

Modify the file `remote-setup/terraform-infrastructure.tf` to include the user that will be created in the remote VMs and point to the project created in GCP (note that you should include the project ID):

```
(line 3) default = "user"
...
(line 18) project = "test-project-1234"
```

Download the JSON credential of the service account which will be used by Terraform to manage the GCP resources and save this file with the name "gcp.json" inside the "keys" directory: <https://console.cloud.google.com/apis/credentials/serviceaccountkey>.

Run Terraform and create the GCP infrastructure

Run the following commands within the `remote-setup` folder to see the changes that will be made by terraform, apply these changes and destroy them, respectively:

```
terraform plan
terraform apply
```

The following command can be used to destroy the changes:

```
terraform destroy
```

Install docker in all VMs created

<https://docs.docker.com/install/linux/docker-ce/debian/#install-from-a-package>
<https://download.docker.com/linux/debian/dists/stretch/pool/stable/amd64/>

Give permissions to the user created in the remote VMs to run docker:

```
sudo usermod -aG docker $USER
```

Install rke in local computer

<https://rancher.com/docs/rke/v0.1.x/en/installation/>

Deploy remote k8s cluster

Modify the file `remote-setup/cluster.yml` to include the correct IPs and usernames so Rancher can access the VMs and create the cluster. After changing the file, deploy the remote k8s cluster using rke. The following command should be run within the `remote-setup` folder:

```
rke up --config ./cluster.yml
```

- Note that when the VMs were started using terraform all the necessary firewall rules should have been setup already, but you may also need to change your settings to allow additional ports.
 - The docker version installed in the VM needs to be compatible with the rke version installed in the local computer. For example, at the time this project is being worked on (Q3 of 2018) the latest stable release of rke is 1.9.0, which supports docker-ce 17.03.x. However, this is not the latest version of Docker at this time.
-

Move k8s local file created by rancher to k8s local configuration folder

Move k8s configuration file created by rancher to the local configuration folder so k8s can locate the file and reach the nodes. Alternatively, you can also set the **KUBECONFIG** environmental variable to the path of **kube_config_cluster.yml**.

```
mv kube_config_cluster.yml config/ && cp config/kube_config_cluster.yml ~/.kube/config
- or -
export KUBECONFIG=$(pwd)/kube_config_cluster.yml
```

Install Helm in remote k8s cluster

See <https://github.com/helm/helm> for instructions on how to install Helm in your local computer.

Since RBAC is enabled in the cluster it's necessary to create a ServiceAccount and ClusterRoleBinding for the tiller service to manage charts.

```
kubectl --namespace kube-system create sa tiller
kubectl create clusterrolebinding tiller --clusterrole cluster-admin --serviceaccount=kube-system:tiller
helm init
kubectl --namespace kube-system patch deploy/tiller-deploy -p '{"spec": {"template": {"spec": {"serviceAccountName": "tiller"}}
```