**1.Provision Ubuntu server 14.04:-**

[root@192.168.0.1 ~]# cd terraform/

[root@192.168.0.1]# vi aws.tf

provider "aws" {

access\_key = "aws-user access-key"

secret\_key = "aws-user secret key"

region = "us-west-2"

}

data "aws\_ami" "ubuntu" {

most\_recent = true

filter {

name = "name"

values = ["ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-\*"]

}

filter {

name = "virtualization-type"

values = ["hvm"]

}

}

resource "aws\_instance" "web" {

ami = "${data.aws\_ami.ubuntu.id}"

instance\_type = "t2.micro"

tags = {

Name = "your instance name"

}

}

[root@192.168.0.1]# terraform plan

[root@192.168.0.1]# terraform apply

===============================================

Count:-variable to control multiple instances

resource "aws\_instance" "web" {

key\_name = "terraform-demo"

public\_key = "${file("terraform-demo.pub")}"

}

resource "aws\_instance" "my-instance" {

count = "${var.instance\_count}"

ami = "${lookup(var.ami,var.aws\_region)}"

instance\_type = "${var.instance\_type}"

key\_name = "${aws\_key\_pair.terraform-demo.key\_name}"

user\_data = "${file("install\_apache.sh")}"

tags = {

Name = "Terraform-${count.index + 1}"

Batch = "any time u mention for demo"

}

}

variable "ami" {

type = "map"

default = {

"us-east-1" = "ami-04169656fea786776"

"us-west-1" = "ami-006fce2a9625b177f"

}

}

variable "instance\_count" {

default = "2"

}

variable "instance\_type" {

default = "t2.nano"

}

variable "aws\_region" {

default = "us-east-1"

}

**[root@192.168.0.1]# terraform plan**

**[root@192.168.0.1]# terraform apply**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Setup nginx server by using terraform**

Create new terraform configuration file :-

vi www-1.tf

resource ""aws\_instance" "my-instance" {

image = "ubuntu-18-04-x64"

name = "www-1"

region = "us-west-2"

size = "s-1vcpu-1gb"

private\_networking = true

ssh\_keys = [

var.ssh\_fingerprint

]

connection {

host = self.ipv4\_address

user = "root"

type = "ssh"

private\_key = file(var.pvt\_key)

timeout = "2m"

}

provisioner "remote-exec" {

inline = [

"export PATH=$PATH:/usr/bin",

# install nginx

"sudo apt-get update",

"sudo apt-get -y install nginx"

]

}

}

**[root@192.168.0.1]# terraform plan**

**[root@192.168.0.1]# terraform apply**

­­­­­­­­­­­­­­­­­­­­­­­--------------------------------------------------------------------------------------

2.Deploy Nginx web service in kubernetes:-

kubectl get nodes

kubectl create deployment nginx --image=nginx

kubectl get deployments

kubectl describe deployment nginx

kubectl create service nodeport nginx --tcp=80:80

kubectl get svc

"Now you can verify that the Nginx page is reachable on all nodes using the curl command."

curl master-node:30386

curl node-1:30386

curl node-2:30386

"PublicIP:InternalPort>, that is 192.168.0.1.99:30386"

"“WELCOME TO NGINX!”"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Upgrade nginx deployment with zero downtime:-**

**application/deployment.yml**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: nginx-deployment**

**spec:**

**selector:**

**matchLabels:**

**app: nginx**

**replicas: 2**

**template:**

**metadata:**

**labels:**

**app: nginx**

**spec:**

**containers:**

**- name: nginx**

**image: nginx:1.16.1 # Update the version of nginx from 1.14.2 to 1.16.1**

**ports:**

**- containerPort: 80**

*#kubectl apply -f https://k8s.io/examples/application/deployment-update.yaml*

*#kubectl get pods -l app=nginx*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SSL-SELFSIGNED CERTIFICATE INSTALLED IN UBUNTU:-**

1.vim /etc/nginx/snippets/self-signed.conf

ssl\_certificate /etc/ssl/certs/example.com.pem;

ssl\_certificate\_key /etc/ssl/private/example.com.key;

wq!

2. vim /etc/nginx/snippets/ssl-params.conf

ssl\_protocols TLSv1 TLSv1.1 TLSv1.2;

ssl\_prefer\_server\_ciphers on;

ssl\_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH";

ssl\_ecdh\_curve secp384r1;

ssl\_session\_cache shared:SSL:10m;

ssl\_session\_tickets off;

ssl\_stapling off;

ssl\_stapling\_verify on;

resolver 8.8.8.8 8.8.4.4 valid=300s;

resolver\_timeout 5s;

add\_header Strict-Transport-Security "max-age=63072000; includeSubdomains";

add\_header X-Frame-Options DENY;

add\_header X-Content-Type-Options nosniff;

ssl\_dhparam /etc/ssl/certs/dhparam.pem;

wq!

3. vim /etc/nginx/sites-available/default

server {

listen 80 default\_server;

listen [::]:80 default\_server;

# SSL configuration

listen 443 ssl default\_server;

listen [::]:443 ssl default\_server;

include snippets/self-signed.conf;

include snippets/ssl-params.conf;

wq!

#systemctl restart nginx

\*\*\*\*test\*\*\*

https://Server\_IP or Hostname

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-

**docker image based on the php-apache image.**

FROM php:5-apache

ADD index.php /var/www/html/index.php

RUN chmod a+rx index.php

**index.php**

index.php

?php

$x = 0.0001;

for ($i = 0; $i <= 1000000; $i++) {

$x += sqrt($x);

}

echo "OK!";

?>

**application/php-apache.yml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: php-apache

spec:

selector:

matchLabels:

run: php-apache

replicas: 1

template:

metadata:

labels:

run: php-apache

spec:

containers:

- name: php-apache

image: k8s.gcr.io/hpa-example

ports:

- containerPort: 80

resources:

limits:

cpu: 500m

requests:

cpu: 200m

# kubectl apply -f https://k8s.io/examples/application/php-apache.yaml

#kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**playbook to upgrade nginx:-**

---

- hosts: all

tasks:

- name: upgrade nginx web server

apt: pkg=nginx state=upgrade update\_cache=true

notify:

- start nginx

handlers:

- name: start nginx

service: name=nginx state=started

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

--