**Terraform code Examples**



Terraform Module for provisioning a general purpose EC2 host.

**Format**

resource "aws\_instance" "Instance-1" {

ami = "………………………”   
 instance\_type = "………………………”  
 subnet\_id = "………………………”  
 vpc\_security\_group\_ids = ["………………………”]  
 key\_name = "………………………”

tags = {  
 Name = "………………………”

OS = "………………………”

App = "………………………”

Environment" = "………………………”  
 }

}

1. To create EC2 instance

# vi test1.tf

provider "aws" {

profile = "default"

region = "ap-south-1"

}

resource "aws\_instance" "instance-1" {

ami = "ami-04b1ddd35fd71475a"

instance\_type = "t2.micro"

count = 2

}

1. To create EC2 instance with tag name

# vi test1.tf

provider "aws" {

profile = "default"

region = "ap-south-1"

}

resource "aws\_instance" "instance-1" {

ami = "ami-04b1ddd35fd71475a"

instance\_type = "t2.micro"

count = 2

tags = {

Name = “webserver1”

}

}

1. To create multiple EC2 instances with different AMI

# vi test2.tf

provider "aws" {

profile = "default"

region = "ap-south-1"

}

resource "aws\_instance" "instance-1" {

ami = "ami-04b1ddd35fd71475a"

instance\_type = "t2.micro"

}

resource "aws\_instance" "instance-2" {

ami = "ami-0a9d27a9f4f5c0efc"

instance\_type = "t2.micro"

}

resource "aws\_instance" "instance-3" {

ami = "ami-0a4a70bd98c6d6441"

instance\_type = "t2.micro"

}

1. Create Security group and allow SSH and HTTP

# vi test4.tk

provider "aws" {

profile = "default"

region = "ap-south-1"

}

resource "aws\_security\_group" "SG\_1" {

name = "linux-sg-ssh-http"

description = "Allow HTTP and SSH traffic"

ingress {

description = "SSH"

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

ingress {

description = "HTTP"

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

}

1. **Creating Keypair**
2. First create private key in your system

* ssh-keygen

Enter key name: deepakawskey1

Then press 3 times enter

dir or ls

**Now create terraform file**

* Nano test1.tf

provider "aws" {

profile = "default"

region = "ap-south-1"

}

resource "aws\_key\_pair" "keypair-1" {

key\_name = "deepakawskey1"

public\_key = file("deepakawskey1.pub")

}

1. Create Instance with default SG and run apache web service script in new instance
2. Create scipt.sh file in current directory

#!/bin/bash

sudo su

yum install httpd -y

echo "<html> <h1> Welcome to India </h1> </html>" >> /var/www/html/index.html

sudo systemctl start httpd

sudo systemctl enable httpd

1. # vi test7.tf

provider "aws" {

profile = "default"

region = "ap-south-1"

}

resource "aws\_instance" "instance-1" {

ami = "ami-04b1ddd35fd71475a"

instance\_type = "t2.micro"

key\_name = "deepkey1"

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

}

1. Create Load Balancer

provider "aws" {

profile = "default"

region = "ap-south-1"

}

resource "aws\_elb" "elb1" {

name = "deepak-terraform-elb1"

availability\_zones = ["ap-south-1a", "ap-south-1b"]

security\_groups = [ "sg-06629a21f198dc9f2"]

instances = [ "i-0d94c426466f0f6a8", "i-0c15facb3d9260613"]

listener {

instance\_port = 80

instance\_protocol = "http"

lb\_port = 80

lb\_protocol = "http"

}

}

1. Create instance –Install docker –pull image and run container.
2. Create scipt.sh file in current directory

#!/bin/bash

#!/bin/bash

sudo su

yum install docker -y

systemctl start docker

systemctl enable docker

docker pull deepaksaidockerhub/tsd

docker run -it -p 82:80 -d deepaksaidockerhub/tsd

1. # vi test7.tf

provider "aws" {

profile = "default"

region = "ap-south-1"

}

resource "aws\_instance" "instance-1" {

ami = "ami-04b1ddd35fd71475a"

instance\_type = "t2.micro"

key\_name = "deepkey1"

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

}

Other way to launch instance and configure Apache web service

1. Create instance with –Instance with default SG and Apache web service installed.

# vi test6.tk

provider "aws" {

profile = "default"

region = "ap-south-1"

}

resource "aws\_instance" "instance-1" {

ami = "ami-04b1ddd35fd71475a"

instance\_type = "t2.micro"

key\_name = "deepakawskey1"

count = 1

connection {

type = "ssh"

host = self.public\_ip

private\_key = file("deepakawskey1")

user = "ec2-user"

}

provisioner "remote-exec" {

inline = [

"sudo yum install httpd -y",

"sudo systemctl start httpd",

]

}

}

1. Create instance with –Instance with default SG and Apache web service installed.

# vi test7.tk

provider "aws" {

profile = "default"

region = "ap-south-1"

}

resource "aws\_instance" "instance-1" {

ami = "ami-04b1ddd35fd71475a"

instance\_type = "t2.micro"

key\_name = "deepkey1"

user\_data = <<-EOF

#!/bin/bash

sudo su

yum install httpd -y

echo "<html> <h1> Welcome to India </h1> </html>" >> /var/www/html/index.html

sudo systemctl start httpd

sudo systemctl enable httpd

EOF