export AWS\_ACCESS\_KEY\_ID=AKIA5SZ27L24DPTFYTVZ

export AWS\_SECRET\_ACCESS\_KEY=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

export AWS\_DEFAULT\_REGION=us-east-2

https://drive.google.com/drive/folders/19dIqBf1SExSylWGOlXsv94erAQzb8Oom

1)cat provider.tf

provider "aws" {

region = "us-east-2"

access\_key = "AKIA5SZ27L24DPTFYTVZ"

secret\_key = "9r6teM0Cu1HUjK+7dWYb92V0LyFXOluau01lRS/u"

}

**2) root@terraform string]# vi main.tf**

[root@terraform string]# cat main.tf

resource "aws\_instance" "ec2\_example" {

ami = "ami-0568773882d492fc8"

instance\_type = var.instance\_type

tags ={

Name = "Terraform Ec@"

}

}

variable "instance\_type" {

description = "Type of Instance"

type = string

default = "t2.micro"

}

3) r**oot@terraform string]# cat main1.tf**  (number )

resource "aws\_instance" "ec2\_example1" {

ami = "ami-0568773882d492fc8"

instance\_type = var.instance\_type

count = var.instance\_count

tags ={

Name = "Terraform Ec@"

}

}

variable "instance\_count" {

description = "Type of Instance"

type = number

default = 2

}

**4) vi list.tf**

[root@terraform list]# cat list.tf

resource "aws\_instance" "ec2\_example" {

ami = "ami-0568773882d492fc8"

instance\_type = "t2.micro"

}

resource "aws\_iam\_user" "example" {

count = length(var.user\_name)

name = var.user\_name[count.index]

}

variable "user\_name" {

description = "IAM USERNAME"

type = list(string)

default = ["user1", "user2". "user3"]

}

**5) cat map.tf**

resource "aws\_instance" "ec222\_example" {

ami = "ami-0568773882d492fc8"

instance\_type = "t2.micro"

tags = var.project\_environment

}

variable "project\_environment" {

description = "project name & env"

type = map(string)

default = {

project = "project-alpha",

environment = "dev"

}

}

**6) cat main.tf (using external var file )**

resource "aws\_instance" "ec2\_example1" {

ami = "ami-0568773882d492fc8"

instance\_type = var.instance\_type

tags ={

Name = var.environment\_name

}

}

**[root@terraform variable]# cat variable.tf**

variable "instance\_name" {

}

variable "environment\_name" {

}

**[root@terraform variable]# cat stage.tfvars**

instance\_type="t2.micro"

environment\_name = "stage"

**[root@terraform variable]# cat prod.tfvars**

instance\_type="t2.micro"

environment\_name = "prod"

**7)vi vpc.tf**

resource "aws\_vpc" "app\_vpc" {

cidr\_block = var.vpc\_cidr

tags = {

Name = "app-vpc"

}

}

resource "aws\_internet\_gateway" "igw" {

vpc\_id = aws\_vpc.app\_vpc.id

tags = {

Name = "vpc\_igw"

}

}

resource "aws\_subnet" "public\_subnet" {

vpc\_id = aws\_vpc.app\_vpc.id

cidr\_block = var.public\_subnet\_cidr

map\_public\_ip\_on\_launch = true

availability\_zone = "us-east-2a"

tags = {

Name = "public-subnet"

}

}

resource "aws\_route\_table" "public\_rt" {

vpc\_id = aws\_vpc.app\_vpc.id

route {

cidr\_block = "0.0.0.0/0"

gateway\_id = aws\_internet\_gateway.igw.id

}

tags = {

Name = "public\_rt"

}

}

resource "aws\_route\_table\_association" "public\_rt\_asso" {

subnet\_id = aws\_subnet.public\_subnet.id

route\_table\_id = aws\_route\_table.public\_rt.id

}

**vi variables.tf**

variable "vpc\_cidr" {

default = "178.0.0.0/16"

}

variable "public\_subnet\_cidr" {

default = "178.0.10.0/24"

}

**8)[root@terraform out]# cat out.tf**

resource "aws\_instance" "ec2\_example" {

instance\_type = "t2.micro"

ami ="ami-0568773882d492fc8"

tags ={

Name = "output\_instance"

}

}

output "my\_console\_output" {

value = aws\_instance.ec2\_example.public\_ip

}

**9)root@terraform local]# cat local.tf**

locals {

staging\_env = "staging"

}

resource "aws\_vpc" "staging-vpc" {

cidr\_block = "10.0.0.0/16"

tags = {

Name = "${local.staging\_env}--tag"

}

}

#subnet

#tags

#ami

**10)# cat sec-grp.tf**

resource "aws\_instance" "web" {

ami = "ami-0568773882d492fc8"

instance\_type = "t2.micro"

security\_groups = [aws\_security\_group.global.name]

tags = {

Name = "instance"

}

}

resource "aws\_security\_group" "global" {

name ="Allow httpssh"

ingress {

protocol = "TCP"

cidr\_blocks = ["0.0.0.0/0"]

from\_port = 443

to\_port = 443

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

ipv6\_cidr\_blocks = ["::/0"]

}

}

**11)vi local.tf**

locals {

staging\_env = "staging"

}

resource "aws\_vpc" "vpc" {

cidr\_block = "10.0.0.0/16"

tags = {

Name = "${local.staging\_env}--vpc-tag"

}

}

resource "aws\_subnet" "subnet" {

vpc\_id = aws\_vpc.vpc.id

cidr\_block = "10.0.1.0/24"

tags = {

Name = "${local.staging\_env}--subnet-tag"

}

}

resource "aws\_internet\_gateway" "gw" {

vpc\_id = aws\_vpc.vpc.id

tags = {

Name = "${local.staging\_env}--gw-tag"

}

}

resource "aws\_route\_table" "rt" {

vpc\_id = aws\_vpc.vpc.id

route {

cidr\_block = "0.0.0.0/0"

gateway\_id = aws\_internet\_gateway.gw.id

}

tags = {

Name = "${local.staging\_env}--rt-tag"

}

}

resource "aws\_instance" "web" {

ami = "ami-0568773882d492fc8"

instance\_type = "t2.micro"

subnet\_id = aws\_subnet.subnet.id

tags = {

Name = "${local.staging\_env}--instance-tag"

}

}

output "my\_console\_output" {

value = aws\_instance.web.public\_ip

}

**12)vi all-ec2.tf**

| variable "ingressrules" { |  |
| --- | --- |
|  | type = list(number) |
|  | default = [80,443] |
|  | } |
|  |  |
|  |  |
|  | variable "egressrules" { |
|  | type = list(number) |
|  | default = [80,443,25,3306,53,8080] |
|  | } |
|  |  |
|  | resource "aws\_instance" "web" { |
|  | ami = "ami-0f1fb91a596abf28d" |
|  | instance\_type = "t2.micro" |
|  | security\_groups = [aws\_security\_group.global.name] |
|  | tags = { |
|  | Name = "With-VPC" |
|  | } |
|  | } |
|  |  |
|  | resource "aws\_security\_group" "global" { |
|  | name = "Allow HTTPsh2021" |
|  |  |
|  | dynamic "ingress" { |
|  | iterator = port |
|  | for\_each = var.ingressrules |
|  | content { |
|  | from\_port = port.value |
|  | to\_port = port.value |
|  | protocol = "TCP" |
|  | cidr\_blocks = ["0.0.0.0/0"] |
|  | } |
|  | } |
|  |  |
|  | dynamic "egress" { |
|  | iterator = port |
|  | for\_each = var.egressrules |
|  | content { |
|  | from\_port = port.value |
|  | to\_port = port.value |
|  | protocol = "TCP" |
|  | cidr\_blocks = ["0.0.0.0/0"] |
|  | } |
|  | } |
|  | } |

**13)# cat s3.tf**

resource "aws\_s3\_bucket" "bu" {

bucket = "my-test-bucket655rhgfhgci656jhg"

acl = "public-read"

versioning {

enabled = true

}

}

resource "aws\_s3\_bucket\_object" "object" {

bucket = aws\_s3\_bucket.bu.id

key = "sample.txt"

source = "/root/s3/sample.txt"

}

**14)[root@terraform for-each-global]# cat loop.tf**

variable "user\_names" {

type = set(string)

default =["user1", "user2", "user3"]

}

resource "aws\_iam\_user" "example"{

for\_each = var.user\_names

name = each.value

}

**15)# cat loop1.tf**

variable "name" {

type = list(string)

default =["apple", "mango", "orange"]

}

output "upper\_name"{

value = [for name in var.name : upper(name)]

}

**16) cat rds-database.tf**

resource "aws\_db\_instance" "myrds" {

allocated\_storage = 10

engine = "mysql"

engine\_version = "8.0.27"

instance\_class = "db.t2.micro"

name = "mydb"

username = "foo"

password = "database"

# parameter\_group\_name = "default.mysql8.0.27"

storage\_type = "gp2"

identifier = "rdstf"

publicly\_accessible = true

skip\_final\_snapshot = true

tags = {

Name = "MyRDS"

}

}

17)