**Terraform Related notes**

1)first you have to visit [**https://www.terraform.io/downloads.html**](https://www.terraform.io/downloads.html)

**2)Download terraform which is your compatible os type.**

**3)after downloading complete extract terraform setup then install terraform using sudo mv terraform /usr/bin/ command.**

**4)after that you want to check its installed or not enter terraform command.**

**5)terraform –-vsersion to check which version installed in your system.**

**Afetr that enter one by one command which is listed below.**

**first create one user in aws console with administrator access and download csv file.**

**1)mkdir mytraining**

**2)cd mytraining/**

**3)vim aws.tf (note atfer enter this command showing some error regarding vim not installed then you want to install through this command sudo apt install vim )**

**4)vim aws.tf**

provider "aws" {

access\_key = "AKIAJ7JWTKN7IO5RBLUQ"

secret\_key = "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"

region = "ap-south-1"

}

**5)cat aws.tf (for view purpose)**

**6)terraform init**

**7)vim ec2.tf**

resource "aws\_instance" "my\_ec2" {

ami = "ami code which is available in aws console”

tags {

Name ="My terraform-ec2"

**}**

**instance\_type = "t2.micro"**

**}**

**8)cat ec2.tf**

**9)terraform validate**

**10)terraform plan**

**11)terraform apply**

**12)terraform destroy(this command use only for terminate your changes in aws console)**

**After that we are use variables**

**13)vim variable.tf**

variable "ami"{

default = "ami-531a4c3c"

type ="string"

}

variable "type"{

default ="t2.micro"

}

**14)cat variable.tf**

**15) vim ec2.tf (make changes in ec2 for call variable in this file which is declare in variable file.)**

resource "aws\_instance" "my\_ec2" {// i declare variable

ami = "${var.ami}"

tags {

Name ="My terraform-ec2"

}

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

}

**16)terraform validate**

**17)terraform plan**

**18)terraform apply**

**ec2 modification has been done using variable declaration and calling.After that i want to create one VPC through terraform with 2 subnet mask,internet gateway,route table.**

**19)vim vpc.tf**

resource "aws\_vpc" "Terraform\_VPC" {

cidr\_block = "10.0.0.0/16"

tags {

Name ="Rupesh\_vpc"

}

}

**20)terraform validate**

**21) terraform plan**

**22) terraform apply (after that your VPC has been crated in aws console)**

**after that you want to declare variable for subnet in variable.tf file this very mendatory.**

**23)Vim variable.tf**

variable "vpc\_id"{

default ="vpc-ce6cc5a6"

}

**24)vim subnet1.tf**

resource "aws\_subnet" "subnet1" {

vpc\_id = "${var.vpc\_id}"

cidr\_block = "10.0.1.0/24"

tags {

Name = "subnet1"

}

}

**need to declare variable in vaeriable.tf file (not required)**

variable "subnet1"{

default ="10.0.1.0/24"

}

**25) terraform validate**

**26)terraform plan**

**27)terraform apply(after that your 1st subnet has been created in aws console)**

**28)vim subnet2.tf**

resource "aws\_subnet" "subnet2" {

vpc\_id = "${var.vpc\_id}"

cidr\_block = "10.0.2.0/24"

tags {

Name = "subnet2"

}

}

**need to declare variable (not required)**

variable "subnet2"{

default ="10.0.2.0/24"

}

**29) terraform validate**

**30)terraform plan**

**31)terraform apply(after that your 2nd subnet has been created in aws console)**

**32)vim gw.tf(gateway)**

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

vpc\_id = "${var.vpc\_id}"

tags {

Name = "terraform\_gw"

}

}

**33)terraform validate**

**34) terraform plan**

**35) terraform apply (after that your gate way has been create in aws console)**

**37)vim variable.tf (you want to declare variable for route table)**

variable "gw\_id"{

default ="igw-ebeddc82"

}

**38)vim routetable.tf**

resource "aws\_route\_table" "main" {

vpc\_id = "${var.vpc\_id}"

route {

cidr\_block = "0.0.0.0/0"

gateway\_id = "${var.gw\_id}"

}

tags {

Name = "route table"

} }

**39)terraform validate**

**40)terraform plan**

**41)terraform apply**

**42)terraform destroy**