**Hybrid Multi Cloud Computing Task 2**

**Given By Mr. Vimal Daga Sir !!**

Aws vpc subnets gateway wordpress Terraform



**Task Description :-**

AWS task 2

Create a terraform code to

1. Create a VPC

2. In that VPC create 2 subnet

a) public subnet

b) private subnet

3. Create a internet gateway and connect it to the public subnet.

4. Create a routing table for Internet gateway ,update and associate it with public subnet.

5. Launch an ec2 instance using wordpress AMI having the security group allowing port 80 in public subnet. Also attach the key to login into it.

6. Launch an ec2 instance using MYSQL AMI with security group allowing port 3306 in private subnet. Also attach the key.

( no need to give the public IP to MySQL instance).

Pre-requisites :-

- Account On Aws

- A IAM User with Admin Access.

- Any Base Os Installed in ur System in my case its windows 10.

- Aws CLI Installed and configure profile

- Terraform Installed in Base Os

- Knowledge of Aws vpc , security groups , subnets and all

I will try to explain AlmosT EveryThing !!



**Terraform :-**

**Terraform** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software" \o "Open-source software) [infrastructure as code](https://en.wikipedia.org/wiki/Infrastructure_as_code" \o "Infrastructure as code) software tool created by [HashiCorp](https://en.wikipedia.org/wiki/HashiCorp" \o "HashiCorp). It enables users to define and provision a datacenter infrastructure using a high-level configuration language known as Hashicorp Configuration Language (HCL), or optionally [JSON](https://en.wikipedia.org/wiki/JSON" \o "JSON). Terraform supports a number of cloud infrastructure providers such as [Amazon Web Services](https://en.wikipedia.org/wiki/Amazon_Web_Services" \o "Amazon Web Services), IBM Cloud (formerly [Bluemix](https://en.wikipedia.org/wiki/Bluemix" \o "Bluemix)), [Google Cloud Platform](https://en.wikipedia.org/wiki/Google_Cloud_Platform" \o "Google Cloud Platform), [DigitalOcean](https://en.wikipedia.org/wiki/DigitalOcean" \o "DigitalOcean),[[4]](https://en.wikipedia.org/wiki/Terraform_(software)" \l "cite_note-4) [Linode](https://en.wikipedia.org/wiki/Linode" \o "Linode), [Microsoft Azure](https://en.wikipedia.org/wiki/Microsoft_Azure" \o "Microsoft Azure), [Oracle Cloud Infrastructure](https://en.wikipedia.org/wiki/Oracle_Cloud" \o "Oracle Cloud), [OVH](https://en.wikipedia.org/wiki/OVH" \o "OVH), [Scaleway](https://en.wikipedia.org/wiki/Scaleway" \o "Scaleway), [VMware vSphere](https://en.wikipedia.org/wiki/VMware_vSphere" \o "VMware vSphere) or [Open Telekom Cloud](https://open-telekom-cloud.com/) as well as [OpenNebula](https://en.wikipedia.org/wiki/OpenNebula" \o "OpenNebula)[[12]](https://en.wikipedia.org/wiki/Terraform_(software)" \l "cite_note-12) and [OpenStack](https://en.wikipedia.org/wiki/OpenStack" \o "OpenStack)

sO Lets Start

This is WhaT I Have Done Using Terraform Code -

- Created key and download it by using terraform For Furthur Use

- Created VPC with instance tenancy as default and dns\_hostname true

- Created Security Group in same vpc by allowing port no 22 for ssh , 80 for weberver and 3306 for mysql .

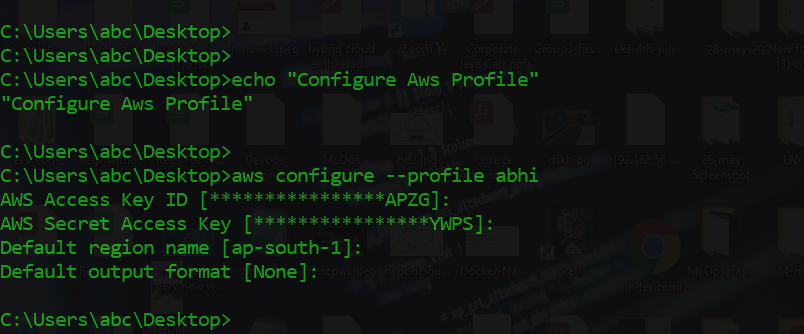
- Then I have created 2 subnets first one is public subnet and second one is private

- Internet Gateway

- Route Table and edit for route table association

- And Finally launched our instance using our own setup means on our own vpc , subnets and all .

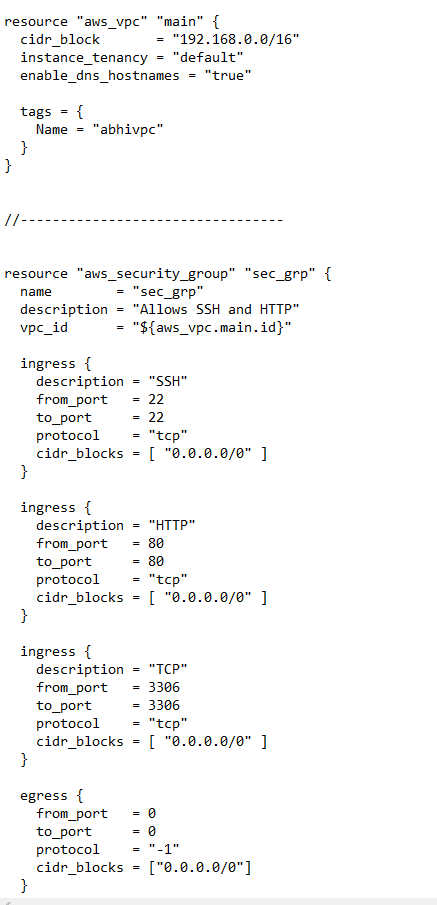
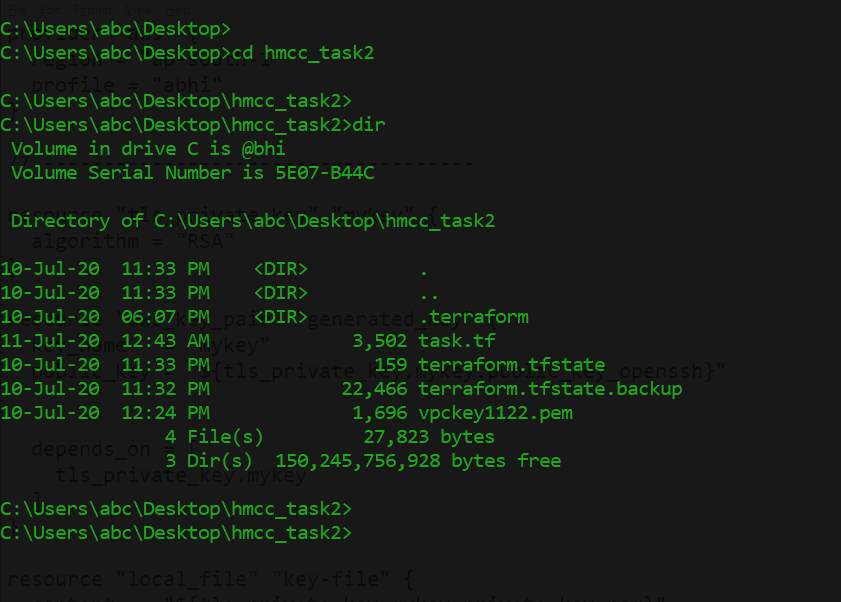
So First of all we need to configure our aws profile but before that we need a aws IAM user with Admin Power BuT same I had showed u multiple times in my other task so go through those task if u want to learn that how to create a Aws IAM User with admin access . u can go through my latest that I shared was lauching owncloud on eks ….

So this is how u can configure a profile 

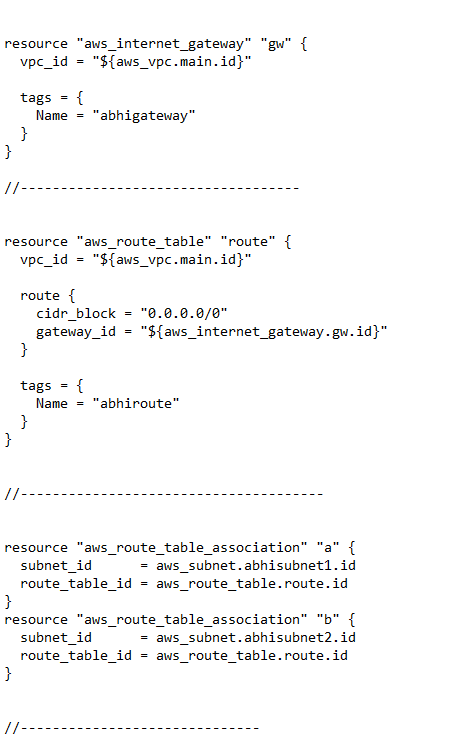
Then we need to give this profile name in our terraform code also so that we can do furthur things and this will be given ub provider section

This is how we have to give profile name

This is the location where we have our files of terraform to automate this task !



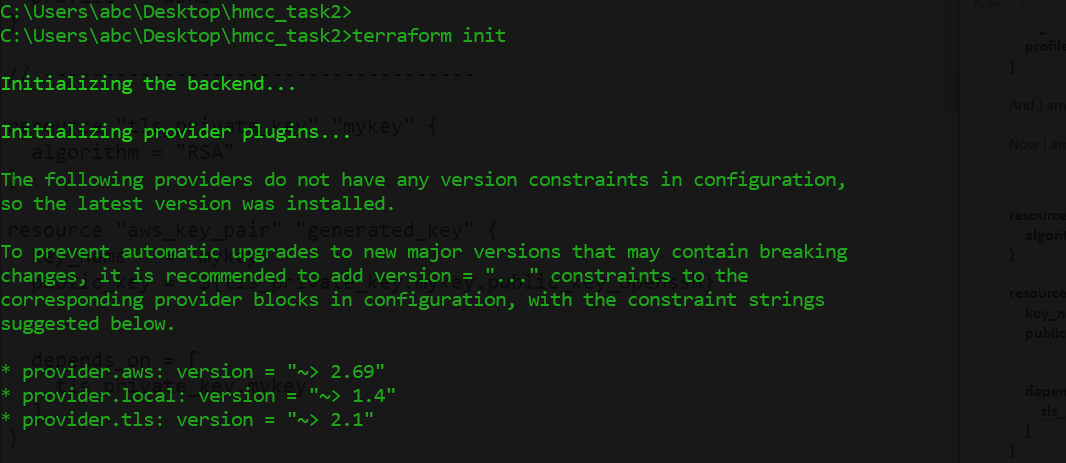




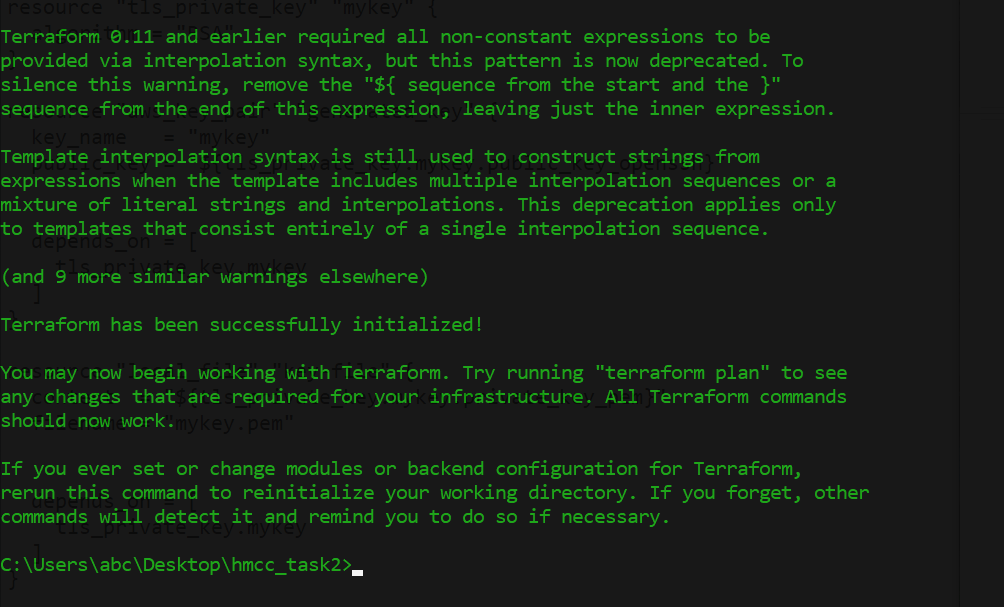


There are some command that we need to run for using terraform I.e

This command will install some plugin required according to our code

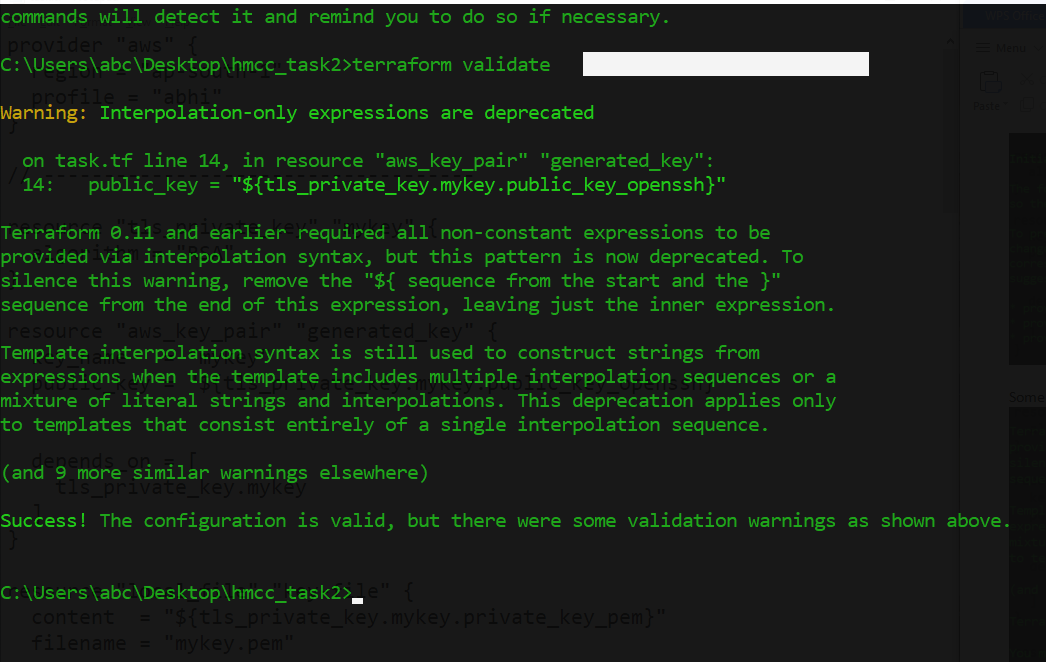
**terrafrorm init**

Some plugin that we need are installed now successfully



Then for testing our we can run terraform validate command

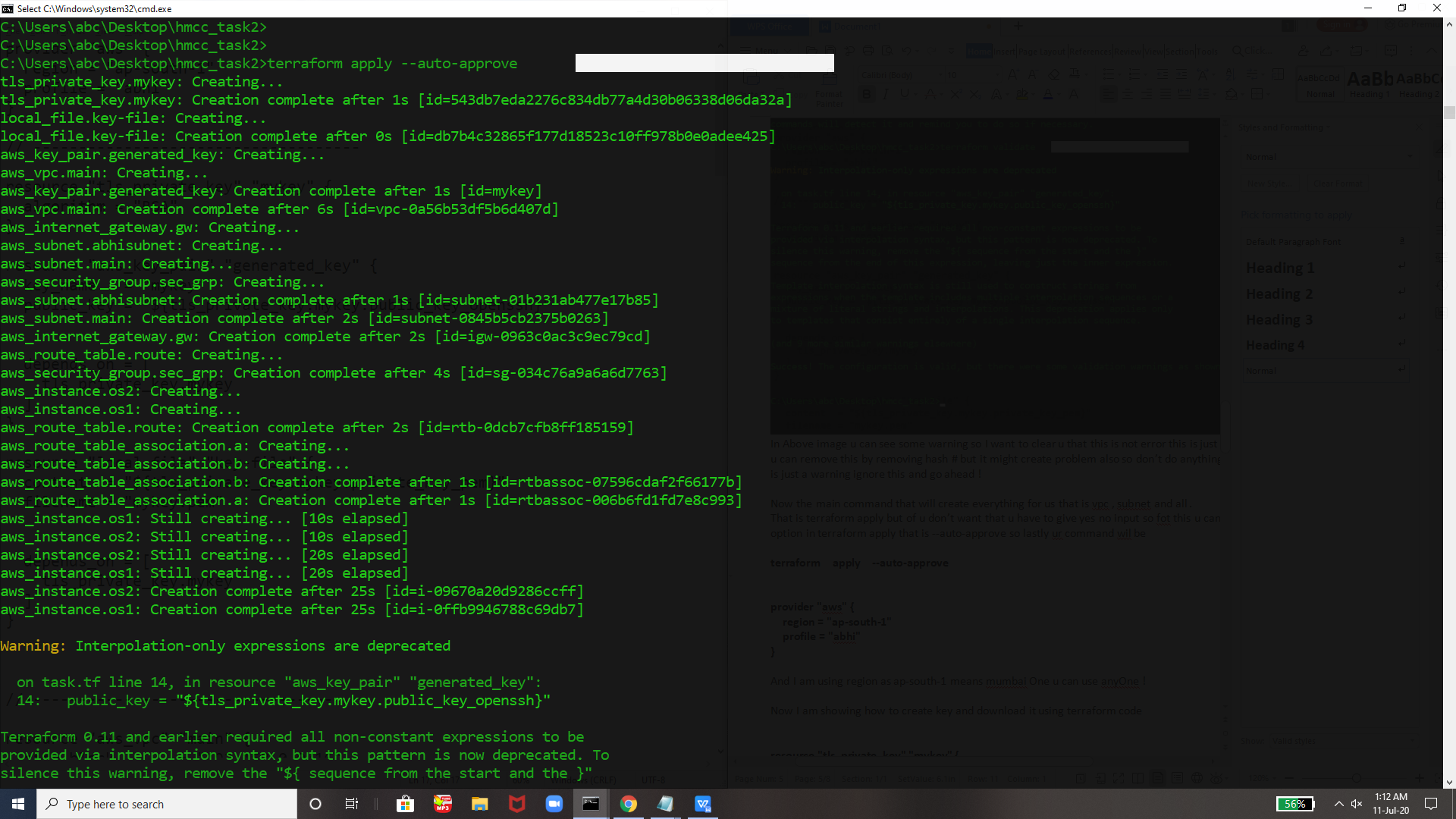
**terraform validate**

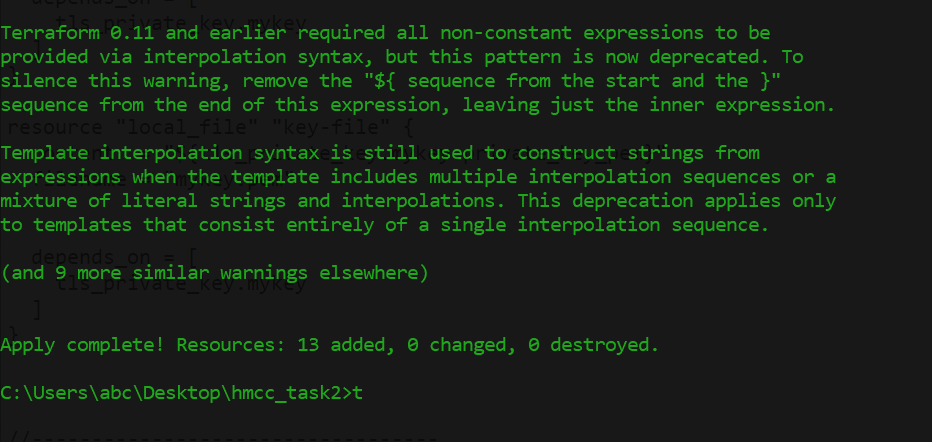


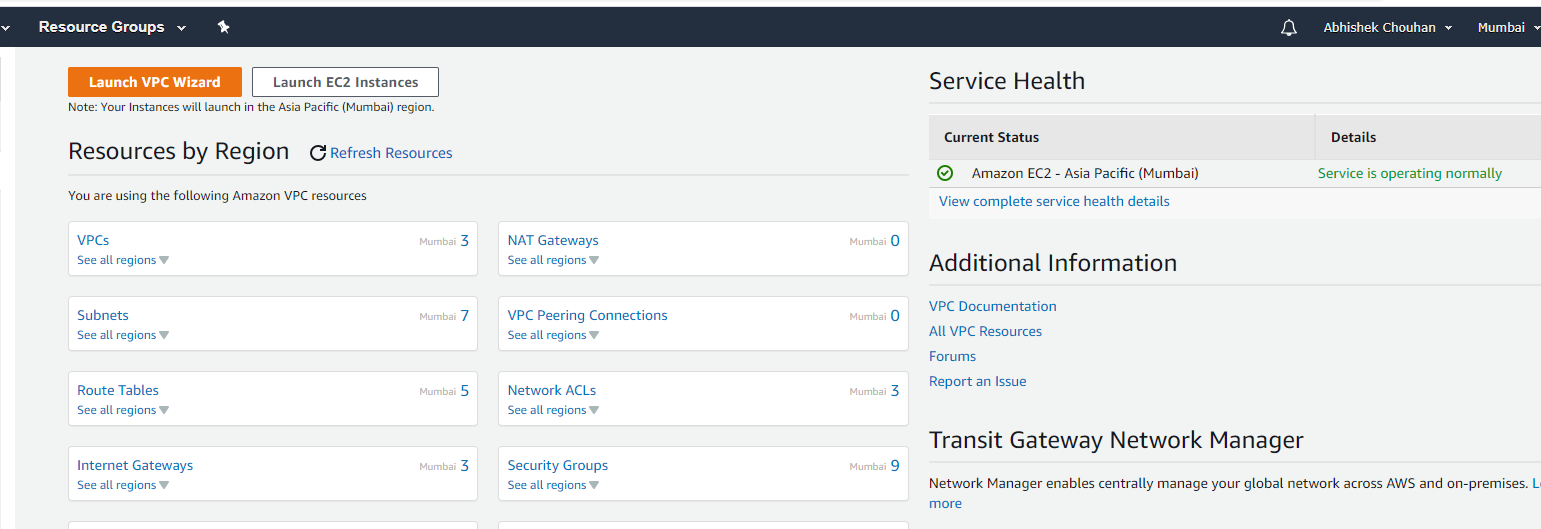
In Above image u can see some warning so I want to clear u that this is not error this is just a warning u can remove this by removing hash # but it might create problem also so don’t do anything and this is just a warning ignore this and go ahead !

Now the main command that will create everything for us that is vpc , subnet and all .That is terraform apply but of u don’t want that u have to give yes no input so fot this u can use one option in terraform apply that is --auto-approve so lastly ur command wil be

**terraform apply --auto-approve**



Again u can see some Interpolation-only expressions are deprecated this kind of word there but this are just warnings so ignore them and go ahead 

Now Everything is created we can see by going to aws console also . that is what I am doing

**provider "aws" {**

**region = "ap-south-1"**

**profile = "abhi"**

**}**

And I am using region as ap-south-1 means mumbai One u can use anyOne !

Now I am showing how to create key and download it using terraform code

**resource "tls\_private\_key" "mykey" {**

**algorithm = "RSA"**

**}**

**resource "aws\_key\_pair" "generated\_key" {**

**key\_name = "mykey"**

**public\_key = "${tls\_private\_key.mykey.public\_key\_openssh}"**

**depends\_on = [**

**tls\_private\_key.mykey**

**]**

**}**

**resource "local\_file" "key-file" {**

**content = "${tls\_private\_key.mykey.private\_key\_pem}"**

**filename = "mykey.pem"**

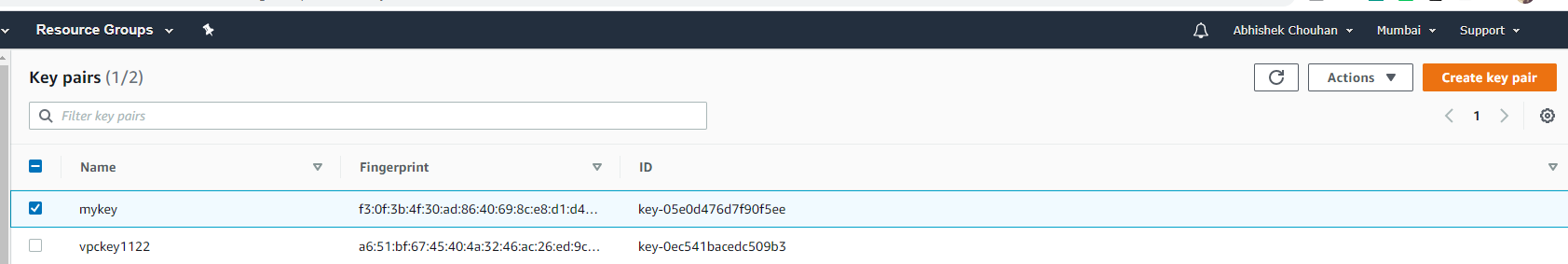
**depends\_on = [**

**tls\_private\_key.mykey**

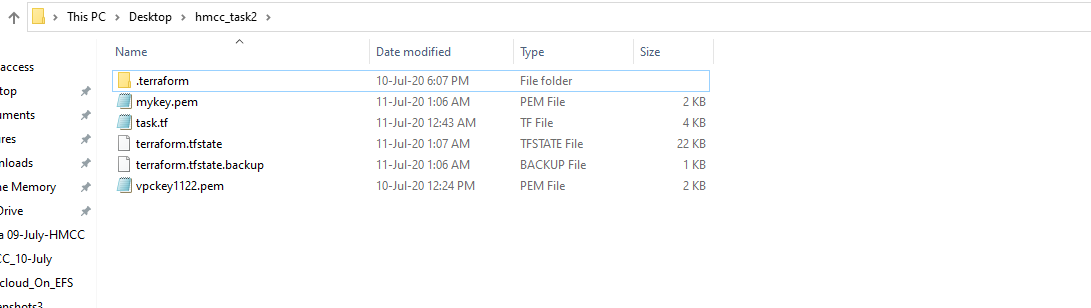
**]**

**}**

U can see that key is created



And its also downloaded in the folder automatically



Th is above code will create new key by the name of mykey and will download also in same folder

Creating vpc with cidt\_block=192.168.0.0/16 and dns\_hostname as true .

**resource "aws\_vpc" "main" {**

**cidr\_block = "192.168.0.0/16"**

**instance\_tenancy = "default"**

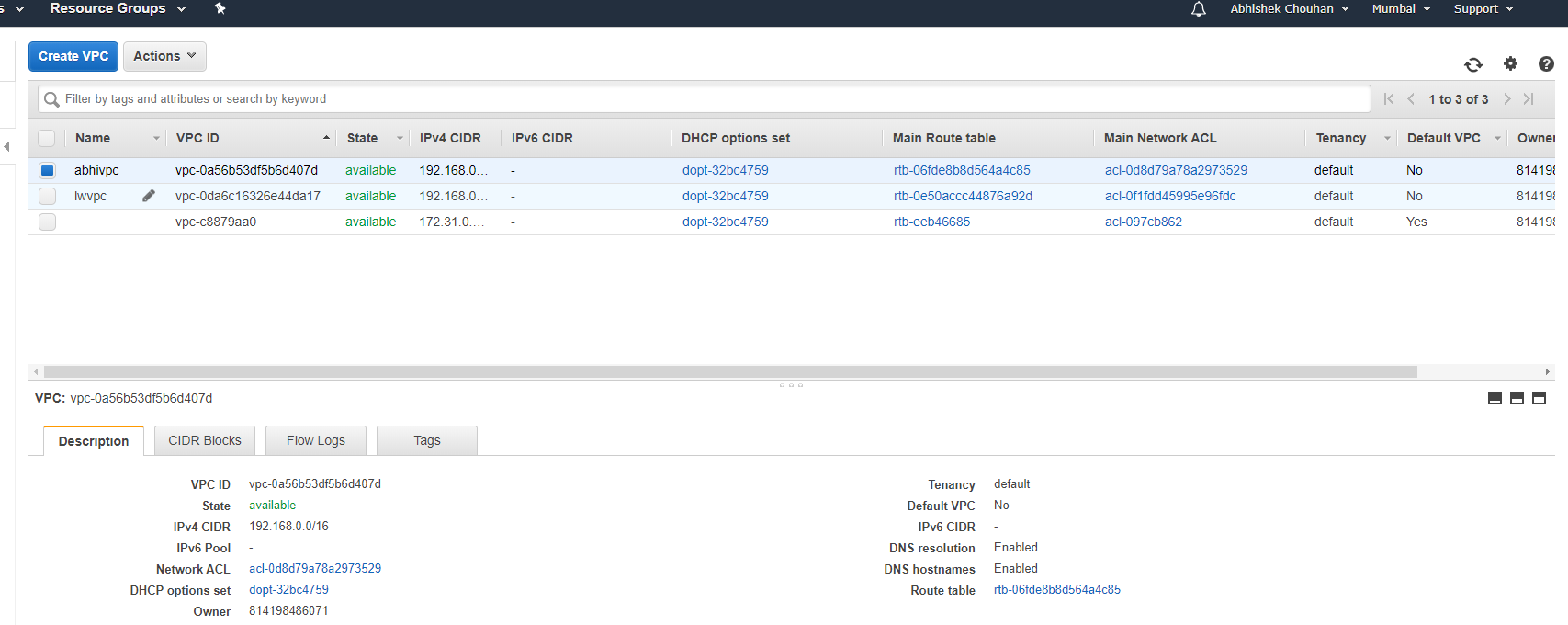
**enable\_dns\_hostnames = "true"**

**tags = {**

**Name = "abhivpc"**

**}**

**}**



Now Creating Security Group For our Instance and furthur use

**resource "aws\_security\_group" "sec\_grp" {**

**name = "sec\_grp"**

**description = "Allows SSH and HTTP"**

**vpc\_id = "${aws\_vpc.main.id}"**

**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" ]**

**}**

**ingress {**

**description = "TCP"**

**from\_port = 3306**

**to\_port = 3306**

**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"]**

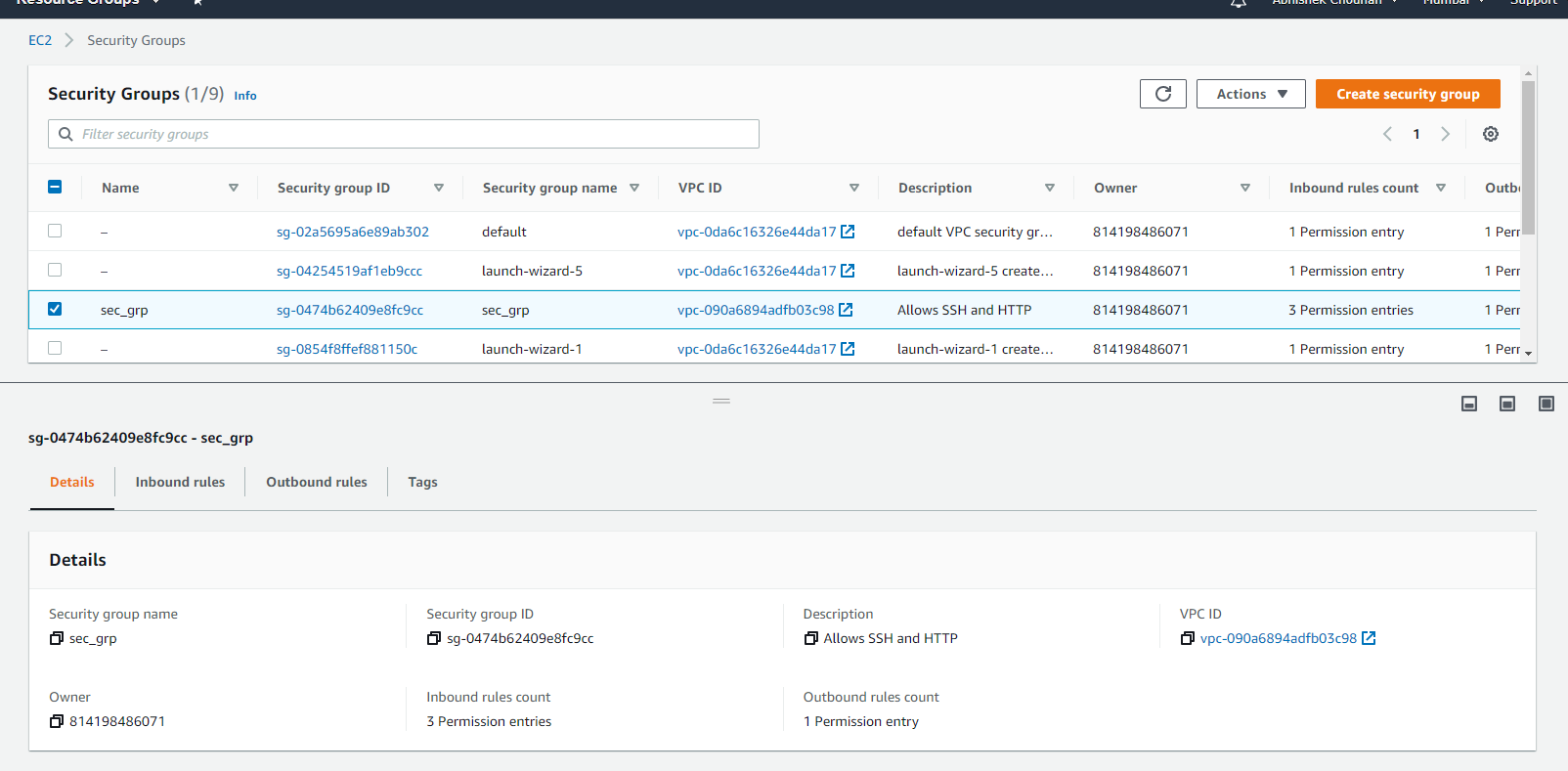
**}**

**tags = {**

**Name = "sec\_grp"**

**}**

**}**



Creating subnets one is public and private for furthur use

**a.) Public Subnet**

**resource "aws\_subnet" "abhisubnet1" {**

**vpc\_id = "${aws\_vpc.main.id}"**

**cidr\_block = "192.168.0.0/24"**

**availability\_zone = "ap-south-1a"**

**map\_public\_ip\_on\_launch = "true"**

**tags = {**

**Name = "abhisubnet1"**

**}**

**}**

**b.) Private Subnet**

**resource "aws\_subnet" "abhisubnet2" {**

**vpc\_id = "${aws\_vpc.main.id}"**

**cidr\_block = "192.168.1.0/24"**

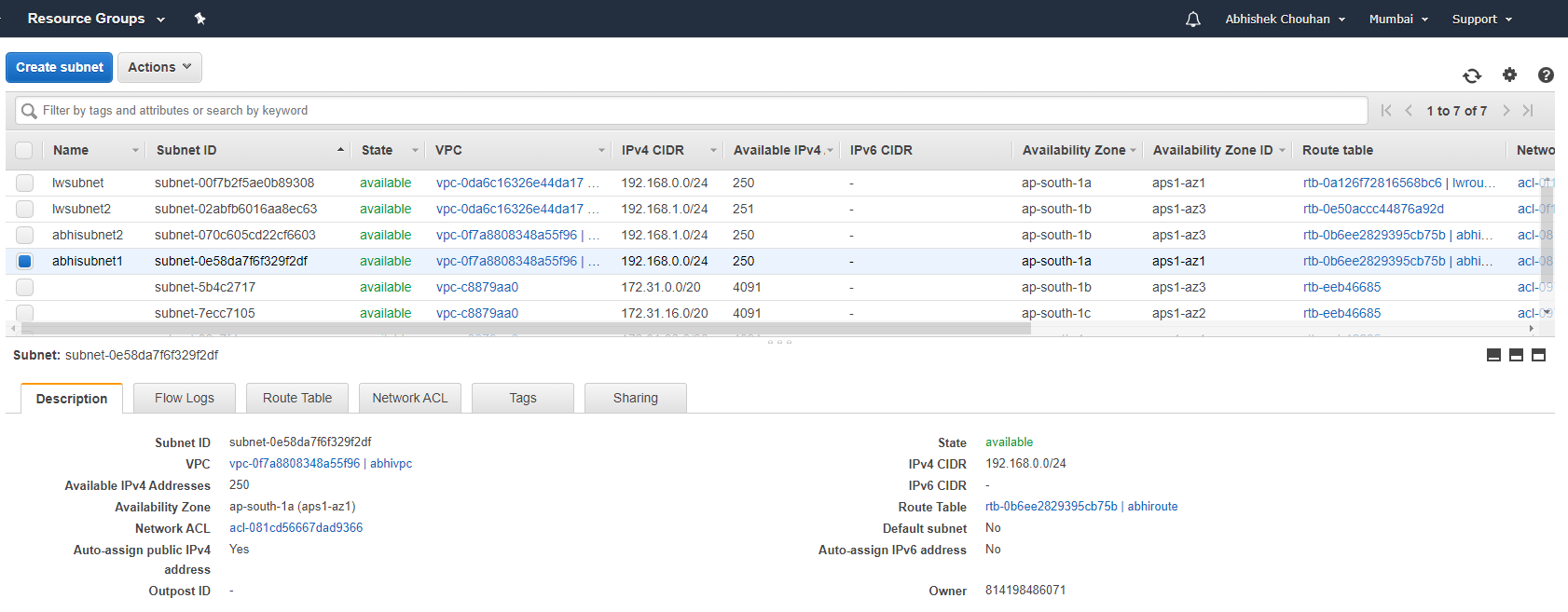
**availability\_zone = "ap-south-1b"**

**tags = {**

**Name = "abhisubnet2"**

**}**

**}**



Creating Gateway

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

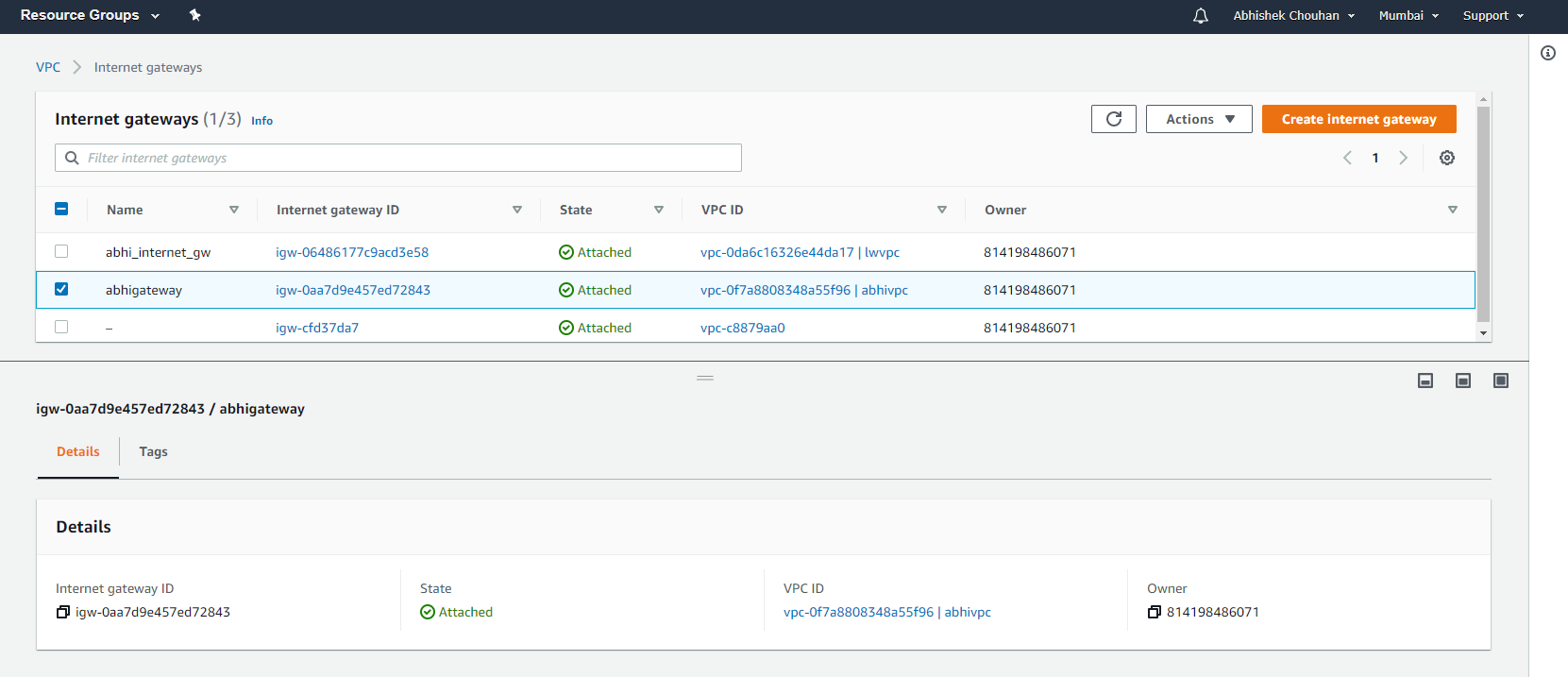
**vpc\_id = "${aws\_vpc.main.id}"**

**tags = {**

**Name = "abhigateway"**

**}**

**}**



Now Creating Route Table -

**resource "aws\_route\_table" "route" {**

**vpc\_id = "${aws\_vpc.main.id}"**

**route {**

**cidr\_block = "0.0.0.0/0"**

**gateway\_id = "${aws\_internet\_gateway.gw.id}"**

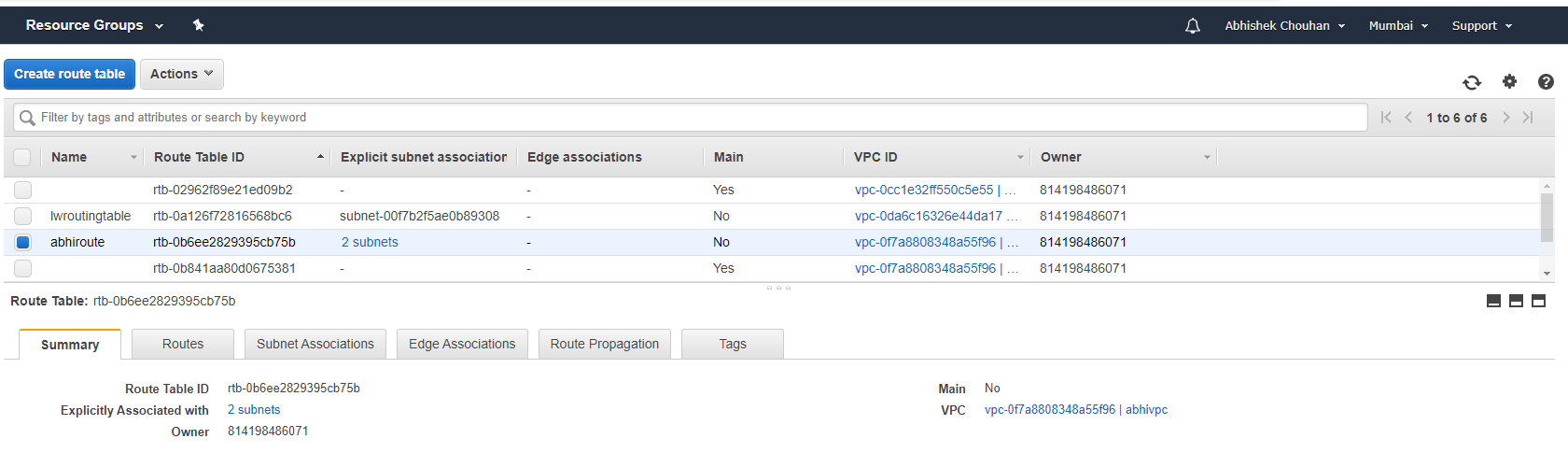
**}**

**tags = {**

**Name = "abhiroute"**

**}**

**}**



Editing the route table association -

**resource "aws\_route\_table\_association" "a" {**

**subnet\_id = aws\_subnet.abhisubnet1.id**

**route\_table\_id = aws\_route\_table.route.id**

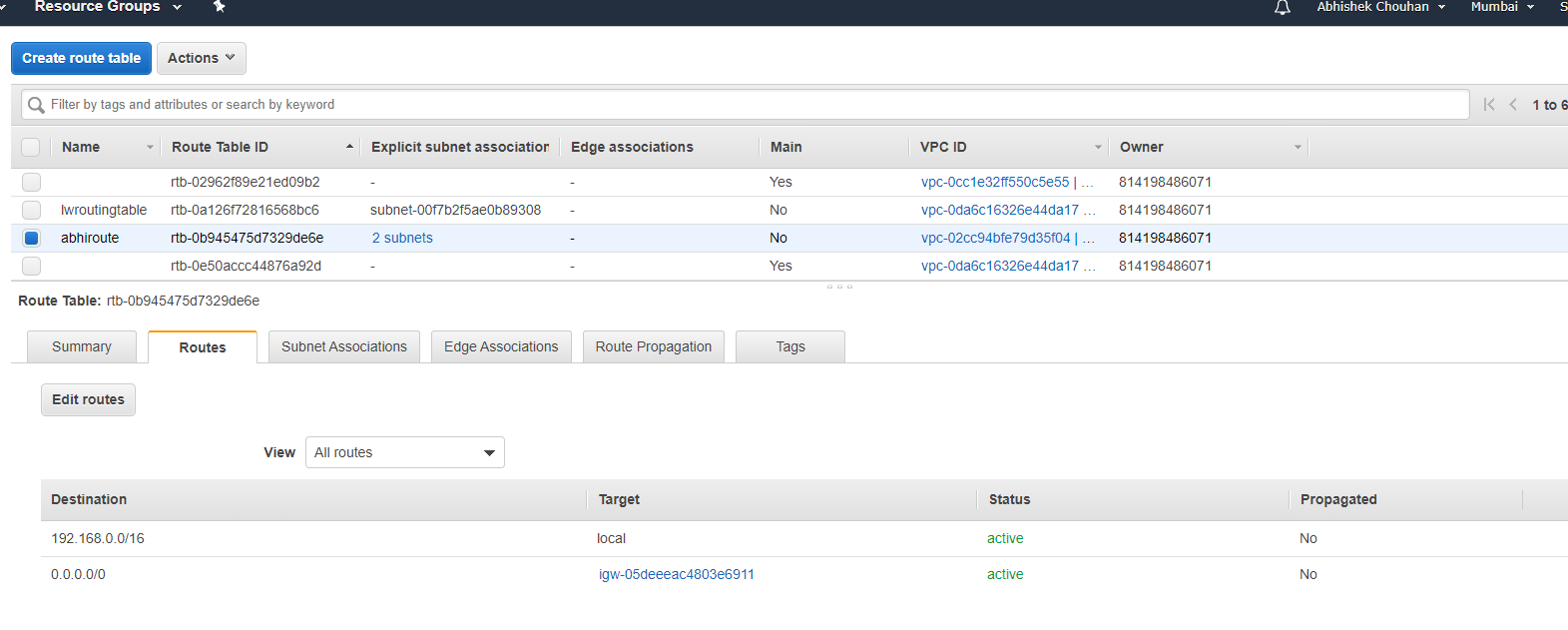
**}**

**resource "aws\_route\_table\_association" "b" {**

**subnet\_id = aws\_subnet.abhisubnet2.id**

**route\_table\_id = aws\_route\_table.route.id**

**}**



Finally Launching Insatnce First one is For WordPress and SecondOne is For MySql -

**resource "aws\_instance" "os1"**

**ami = "ami-7e257211"**

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

**key\_name = aws\_key\_pair.generated\_key.key\_name**

**vpc\_security\_group\_ids = [aws\_security\_group.sec\_grp.id]**

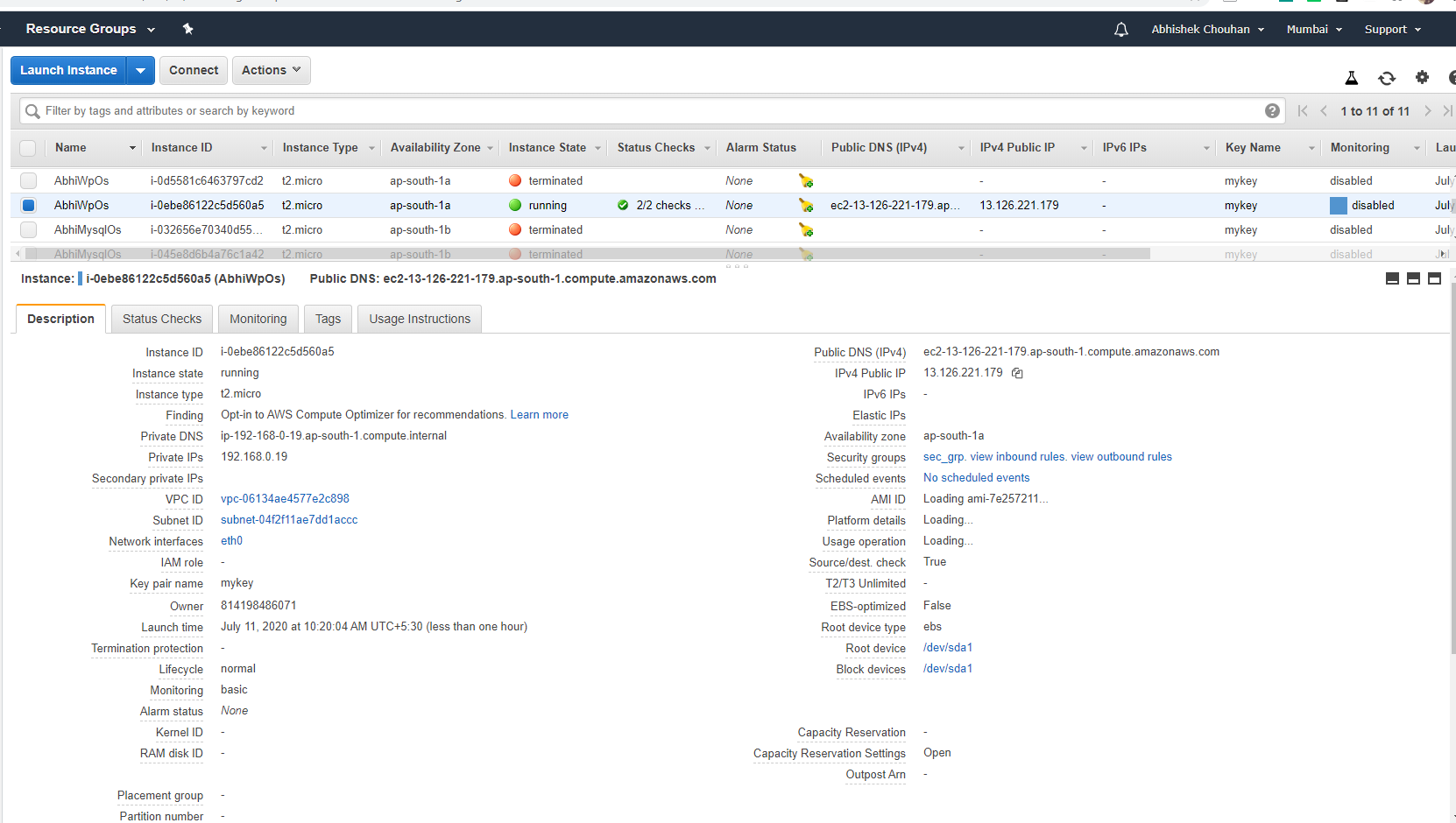
**subnet\_id = "${aws\_subnet.abhisubnet1.id}"**

**tags = {**

**Name = "AbhiWpOs"**

**}**

**}**



1. MySql Instance

**resource "aws\_instance" "os2" {**

**ami = "ami-08706cb5f68222d09"**

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

**key\_name = aws\_key\_pair.generated\_key.key\_name**

**vpc\_security\_group\_ids = [aws\_security\_group.sec\_grp.id]**

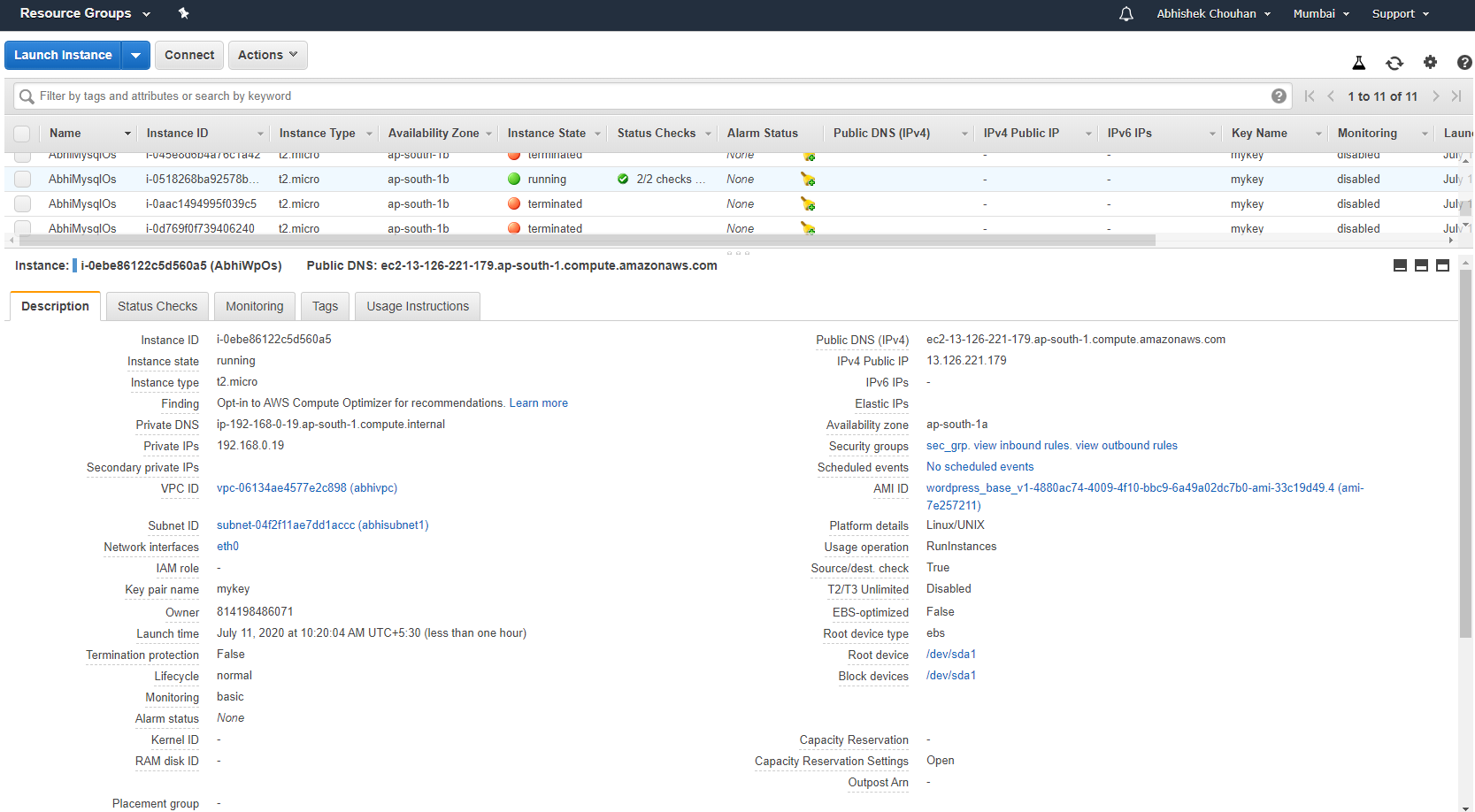
**subnet\_id = "${aws\_subnet.abhisubnet2.id}"**

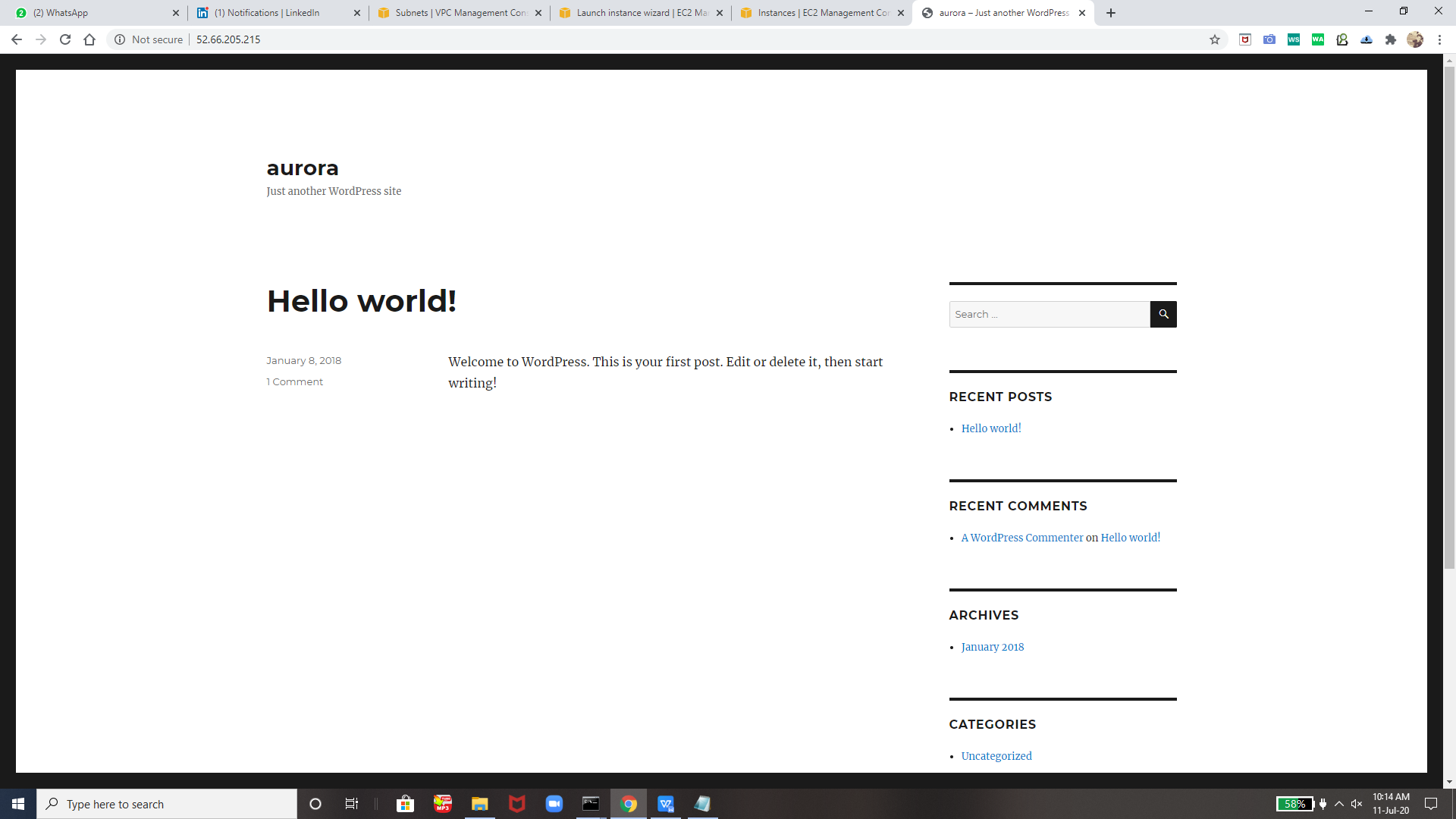
**tags = {**

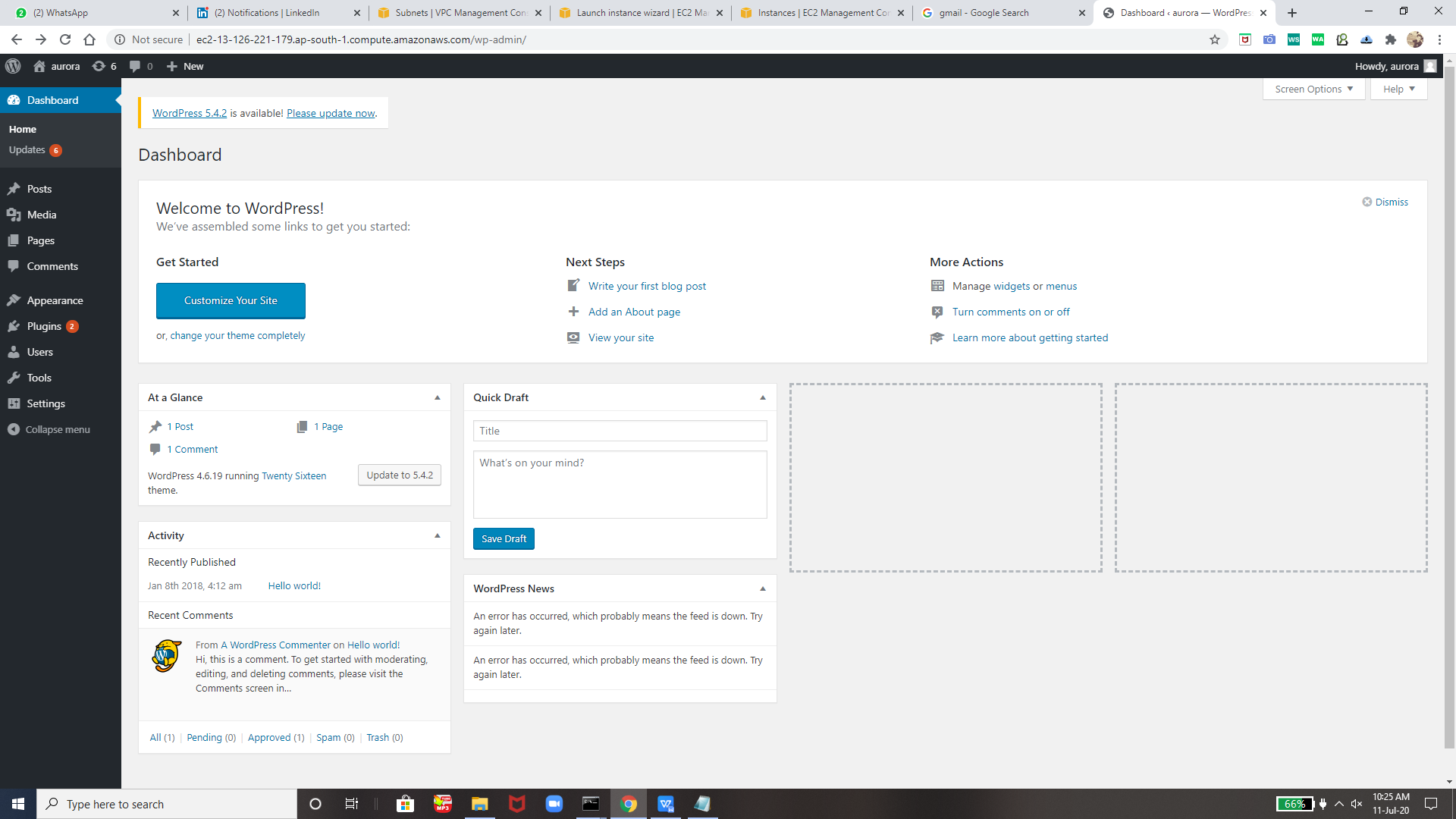
**Name = "AbhiMysqlOs"**

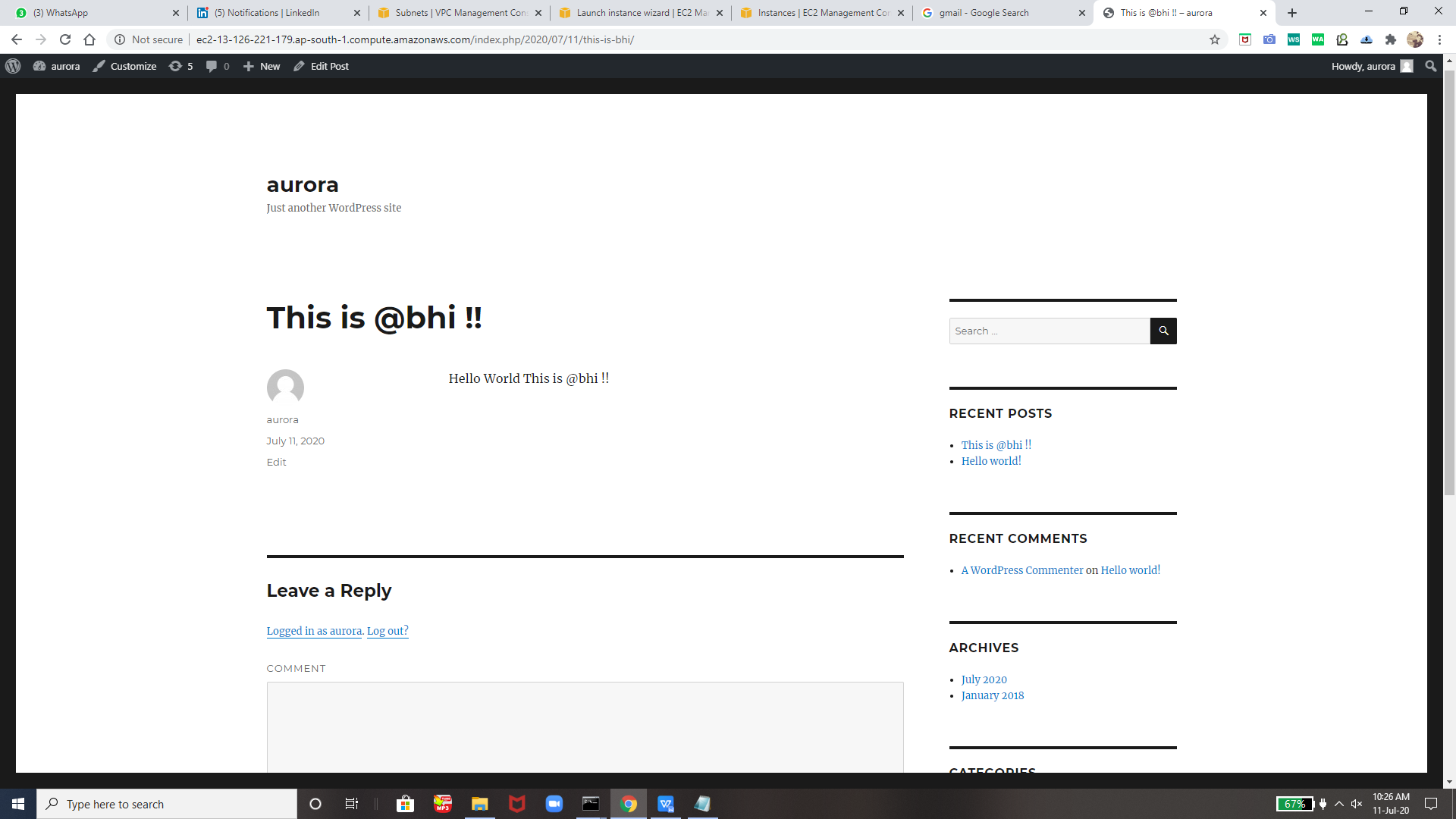
**}**

**}**

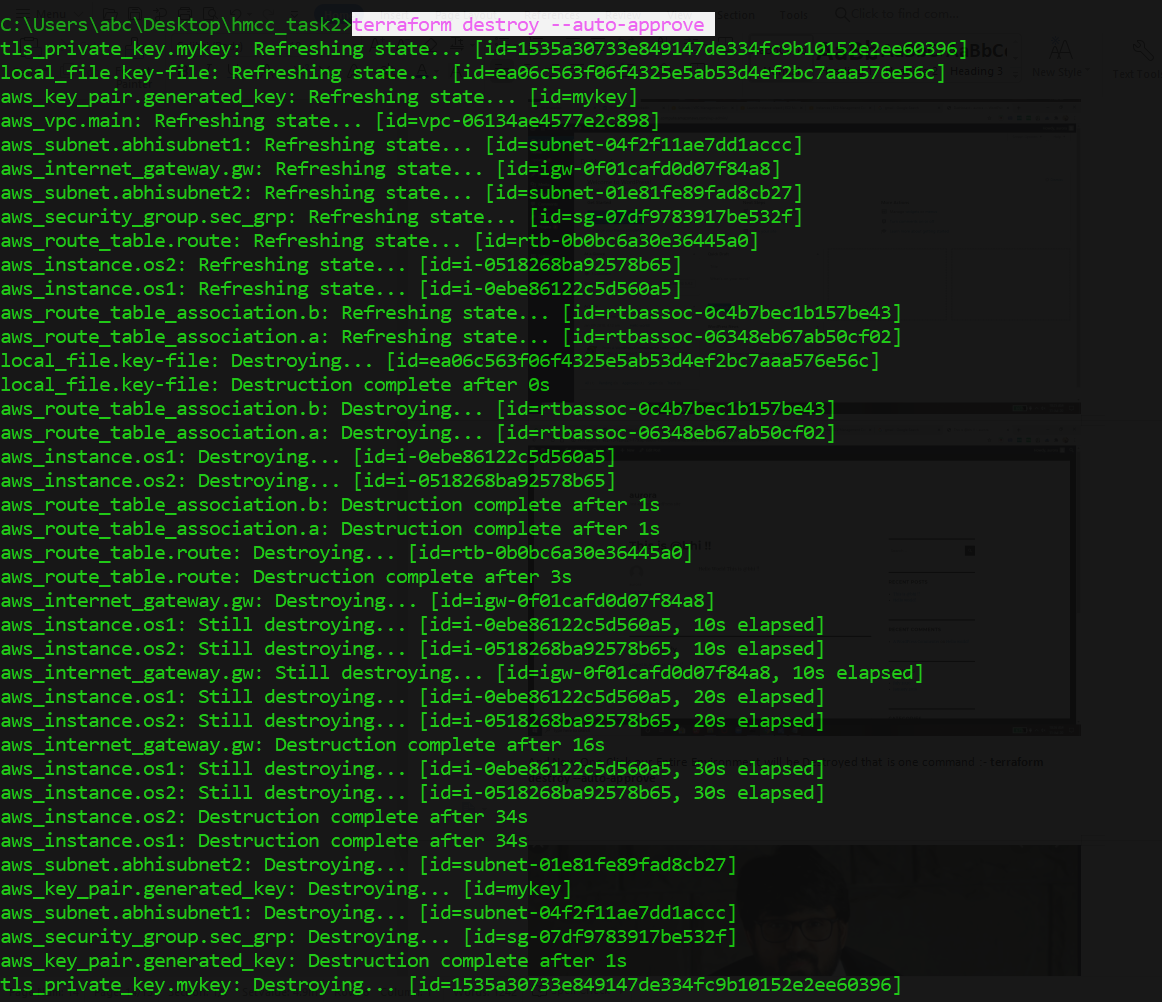






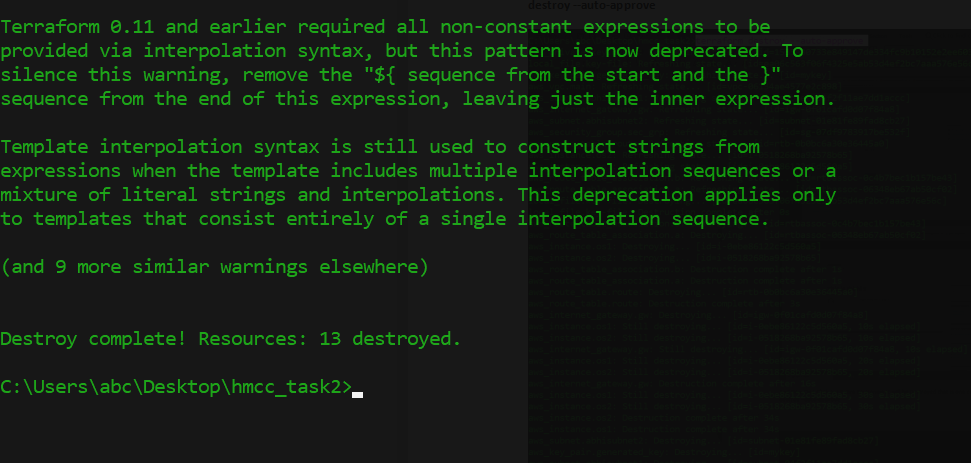


And Now One Click our Entire Environment will be Destroyed that is one command :- **terraform destroy --auto-approve**



Destroying EveryThing!

Destroyed Sucessfully !!





**THank yOu Vimal Sir For Teaching Terraform , Aws and its internal concepts of vpc , subnets , route table , gateway and many more and now we can create the terraform code to automate the things !!**

**Thank you EveryOne For Reading !!**