**1. AWS Infra Setup**

Steps:

1] Write a Infrastructure as code using terraform, which automatically create a VPC.

2] In that VPC we have to create 2 subnets:

a] Public subnet [ Accessible for Public World ]

b] Private subnet [ Restricted for Public World ]

3] Create a public facing internet gateway for connect our VPC/Network to the internet world and attach this gateway to our VPC.

4] Create a routing table for Internet gateway so that instance can connect to outside world, update and associate it with public subnet.

5] Launch an ec2 instance which has Wordpress setup already having the security group allowing port 80 so that our client can connect to our wordpress site.

Also attach the key to instance for further login into it.

6] Launch an ec2 instance which has MYSQL setup already with security group allowing port 3306 in private subnet so that our wordpress vm can connect with the same.

Also attach the key with the same.

Note: Wordpress instance has to be part of public subnet so that our client can connect our site.

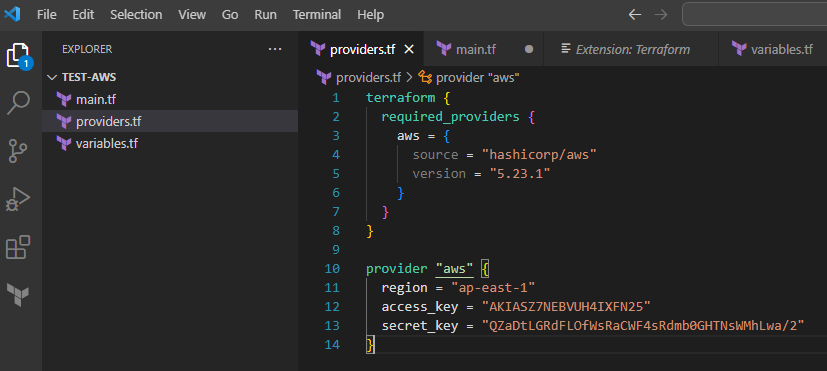
Mysql instance has to be part of private subnet so that outside world can’t connect to it.

Don’t forgot to add auto ip assign and auto dns name assignment option to be enabled.

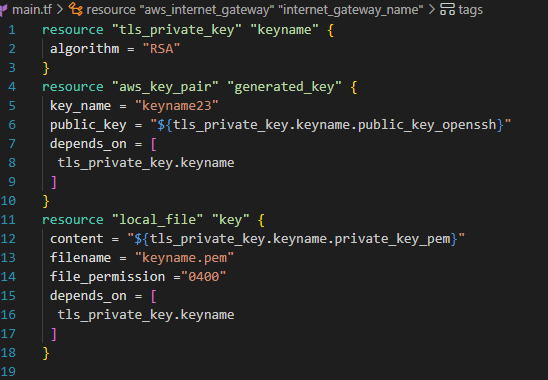
Procedure: 1

As Terraform is a pluggin based application first we have to install the required plugins ( in our case being AWS ) . This is done via HCL code

1. In Above diagram I have created file which is provider.tf using this we can install the required plugins for aws to intract with terraform using API.



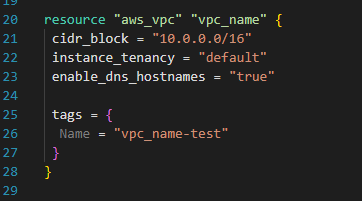
2. Next we can create a key pair for logging in to our instance when needed to.



3.Once you run it , a new Key-Pain is generated and stored in the current working directory with an .pem extension.

4. It’s time to create our VPC , in which we launch multiple subnets . Since each VPC is isolated from one and other also from the cloud as well. But we must specify the range of our IP for using a VN

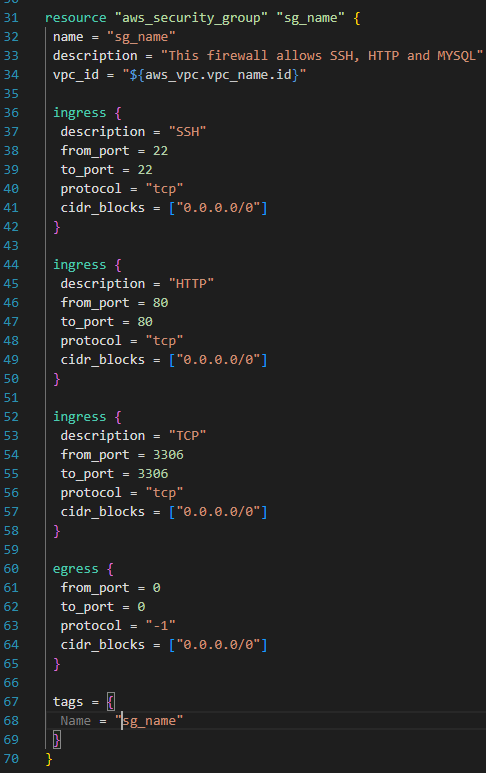
5. Here, CIDR stands for Classless Inter-Domain Routing. A VPC is always assigned with a network name ranging to at max of : 255.255.0.0255.255.0.0 and can use up to 65536 IPs.



6. Now we have to create Security Group for it .

7.An SG is essentially just a (virtual) firewall meant for the instance we create inorder to control the ingress and egress traffic. We have to set the inbound and outbound rules though , which are the types of traffics permitted to enter/exit from our instance

8. In that, we are creating an SG which can allow HTTP , SSH protocols and MySQL traffics .



9. In this step , we are creating subnets based on our VPC .One of them being Public and the other being Private.

10. Here, we are creating two subnets in different AZ’s .

11. In a network there can be 256 Ips, but we are presented with 251. The other 5 being reserved for :

i) Network name

ii) Router

iii) DHCP server

iv) Broadcasting

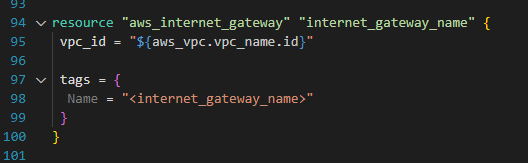
v) Future use

12. In my case, my subnet’s ip is “192.168.0.0” , with the subnet mask being 255.255.255.0



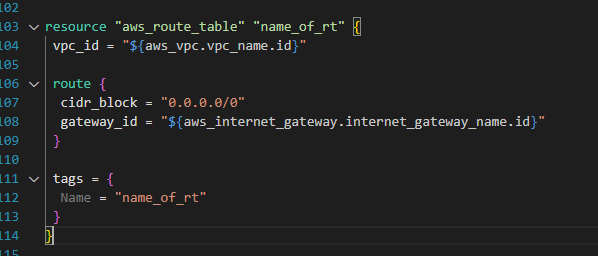
13. Internet Gateway

It’s a part of VPC which establishes communication between the VPC and Internet.



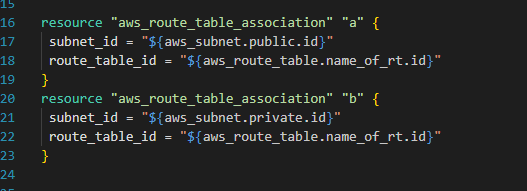
14. Creating a Route Table

15. A Route Table is meant to control where the traffic is diverted to . Every subnet in the VPC must associate with the Route Table, this allows it to control the routing for traffic into the subnets.



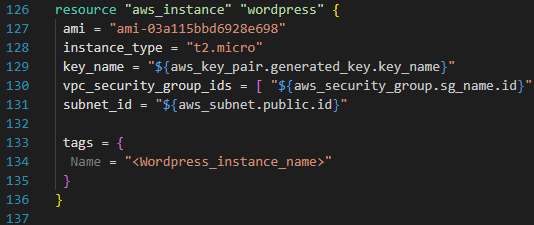
16.Here, we are creating a route table in our VPC, which will lead the traffic to the Internet Gateway.

17. Once that’s done we can Associate it to our subnets.

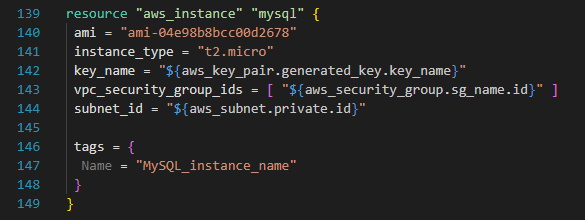


18. Next we are creating instances ( 2 to be exact )

i) WordPress Instance:



ii) MySQL Instance:



19.Now for the last few steps , which is initializing and applying .

Command for initializing :  
  
1. Terraform init - Initialize Terraform and providers.

2.Terraform Plan - Preview Terraform changes before applying.

3. Terraform Apply - Apply Terraform changes to infrastructure.