AWS EC2 VPC

By default, any subnet created, if not explicitly associated to any route table, is then assigned to the default route table.
Hence, the main route table is kept private. Because if it is kept public, every created subnet would become public be default.

We have to create a new route table, which is public.
Name : MYPublicRoute

If you see the type route tables, the one created by us is not Main, whereas the default one is Main.

To make our route table public, add am outer route to the Internet Gateway.
To do this, edit your own route table, which you created above - MYPublicRoute
Add the internet gateway route, source : all ips : 0.0.0.0/0, target : MYIGW (Internet gateway created by us before this)
Same for ipv6 , source -- ::/0 , target : MYIGW
This adds routes to our route table, for all source ips to the Internet gateway, making them public
Now, any subnet connected to this Route Table would automatically become public.

So, go to the Subnet associations tab of the Route table, edit them, and add the desired subnets which you want to keep public (like the subnets where you would keep your webservers)

So, from the 2 subnets we had, we assigned one to the public route table, and the other one remained assigned to the default route table which is private.

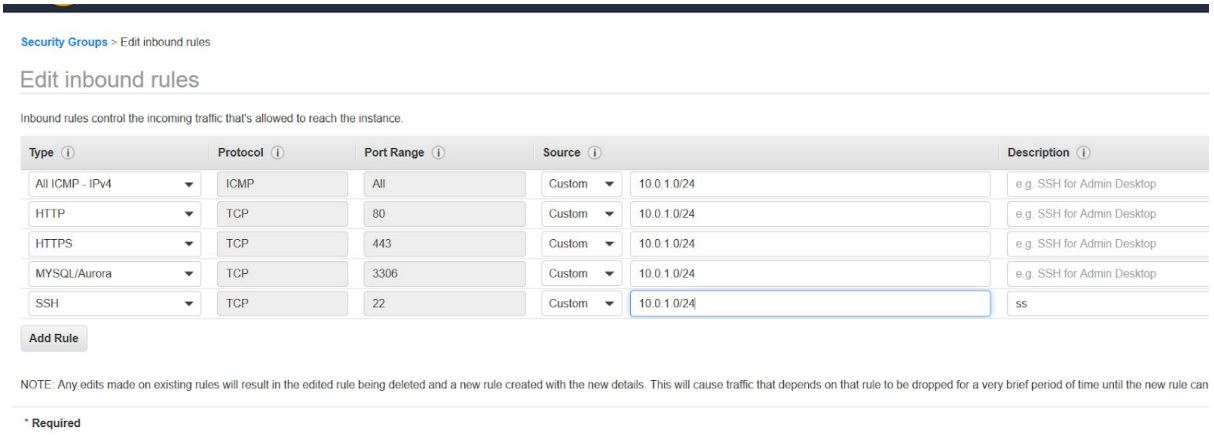
Creating instances in VPC :

Public instance :
Now, go over, launch EC2, assign MyVPCForaws to it.
assign the public subnet :
In security groups, we only see the groups for this VPC. Security groups do not span across VPCs.

While creating EC2, auto assign public IP option is disabled, when you select a private subnet.

When both instances are created, we can ssh into the public instance using the general putty shell.
But if we try to ssh into the DB server / private server frm the public instance, we cannot do so, because we assigned both of them to different security groups, and by default, differnt security groups do not allow each other.
So, go ahead and create a new security group for DB.
10.0.1.0/24

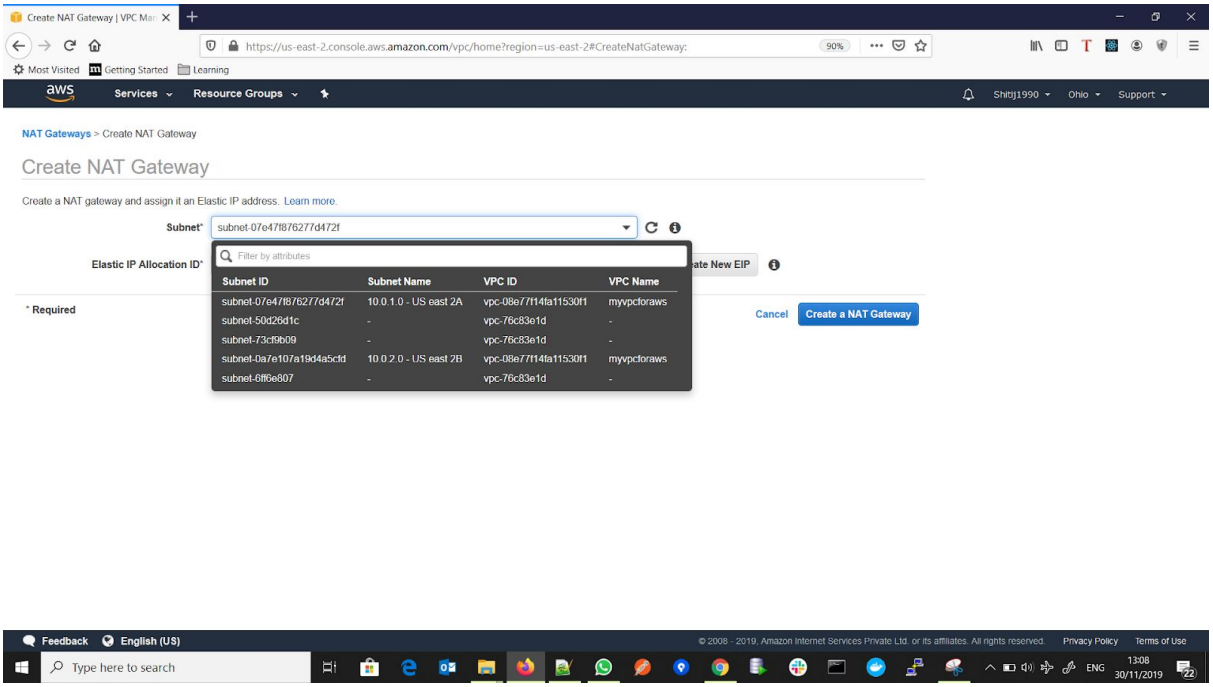
Add rules to the security group to allow traffic :



Assign the security group MyDBSecGrp t the DB server.
Try pinging the private server / DB server from the public webserver now, since you allowed it to be accessible via the sec grp rule, it should be able to ping it.

Create NAT Gateway to allow the private subnet / instance to access internet :

Create NAT gateway, assign it to our public subnet because it will point to internet gateway.



- Create elastic ip and create NAT gateway
- Edit route tables to point to this gateway . Since the private subnet is associated with this Route table, adding this entry into this route table :
    - An entry for mapping all traffic 0.0.0.0/0 to the NAT gateway would mean that we have provided a gateway to the private subnet via this route, to the NAT gateway, which is associated with the public subnet
- Now, we go the NAT geteway, after it is up we can ssh into the private db server, and try installing mysql or anything, it should be able to do so, because it is now able to access the internet.
- Private subnet is associated with the MAIN route table, which is private. Since it was not assigned to the new route table explicitly it remained associated with the main route table by default. When we have created the NAT gateway in the public subnet, we add a route in the MAIN route table (connected to private subnet), which binds all traffic to this route to the NAT gateway, In this way, the private subnets are connected to the NAT gateway, which is in the public subnet, and thus they get a gateway to the internet.

.