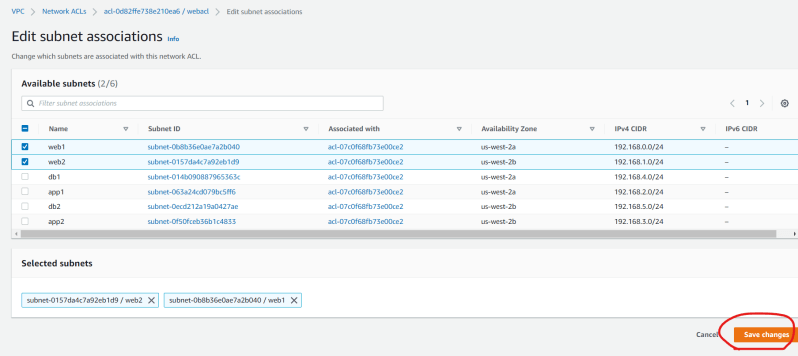
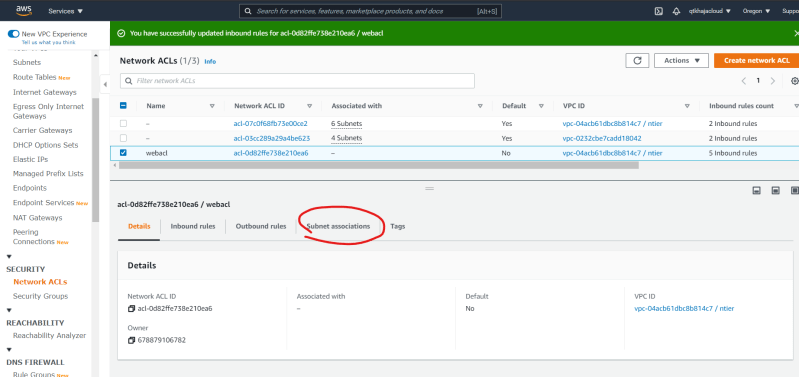
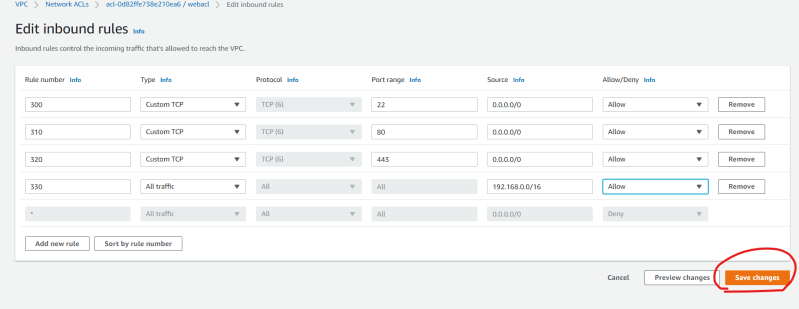
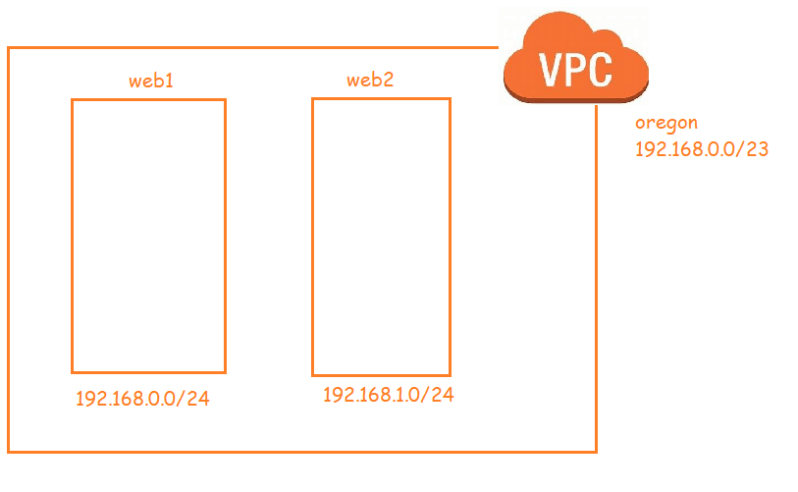
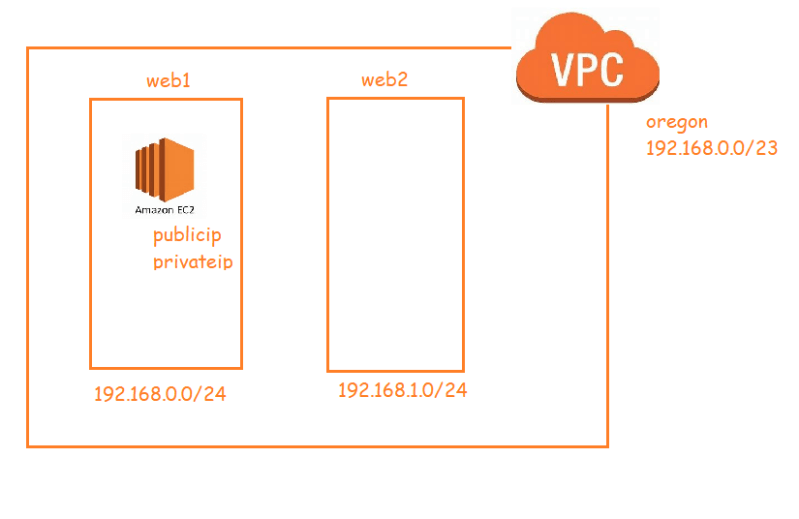
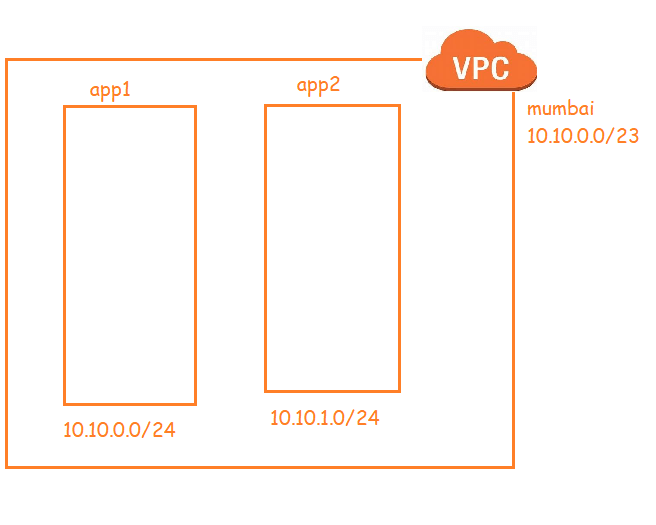
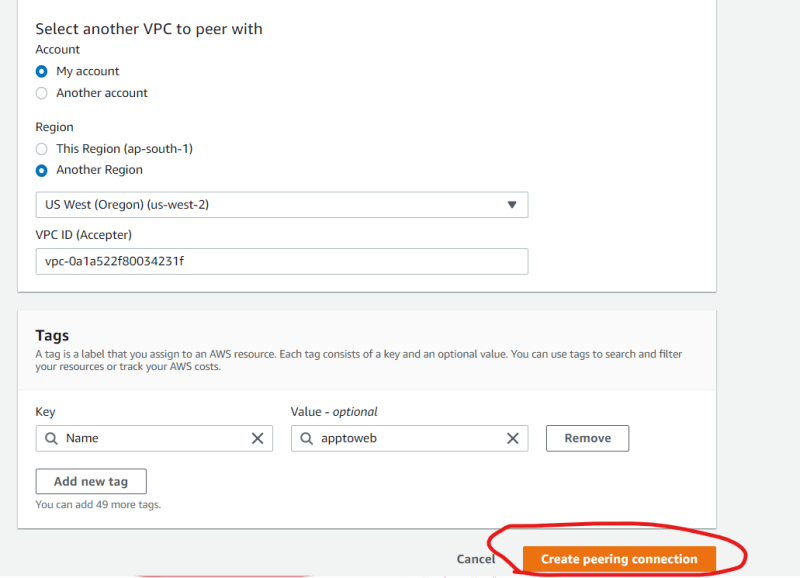
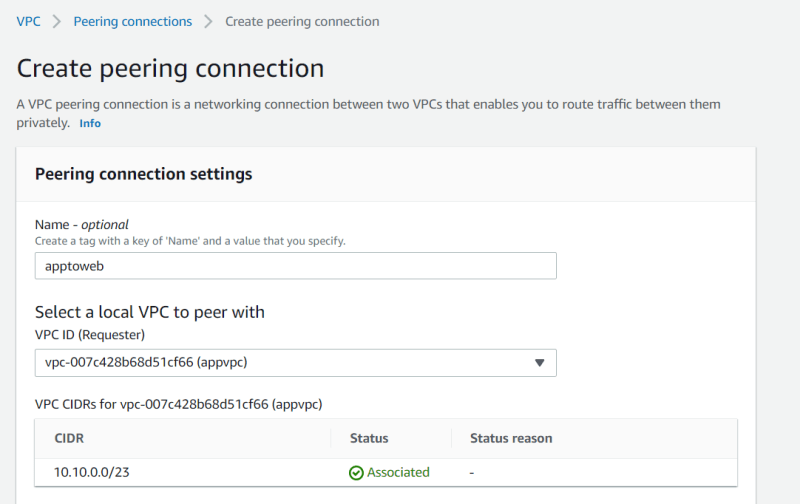
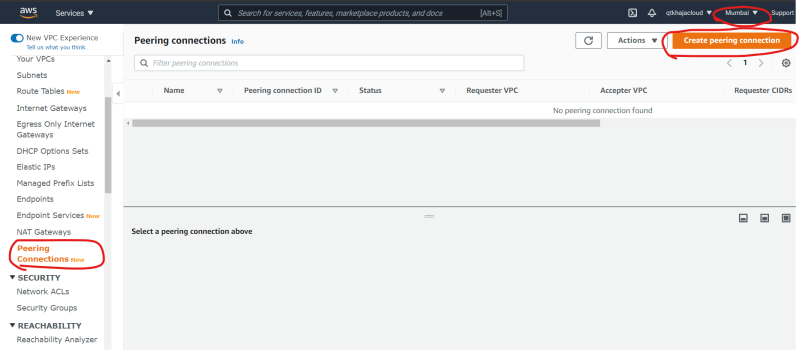
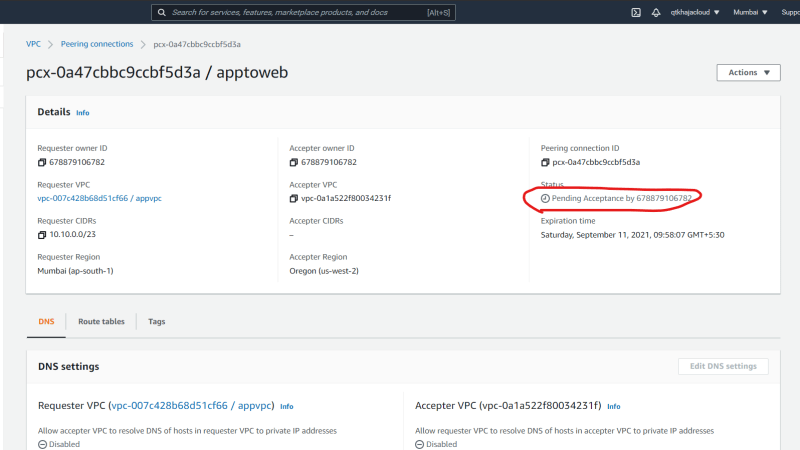
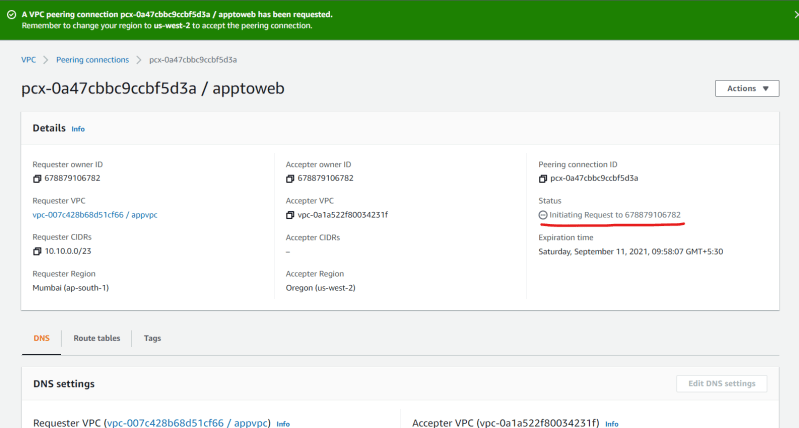
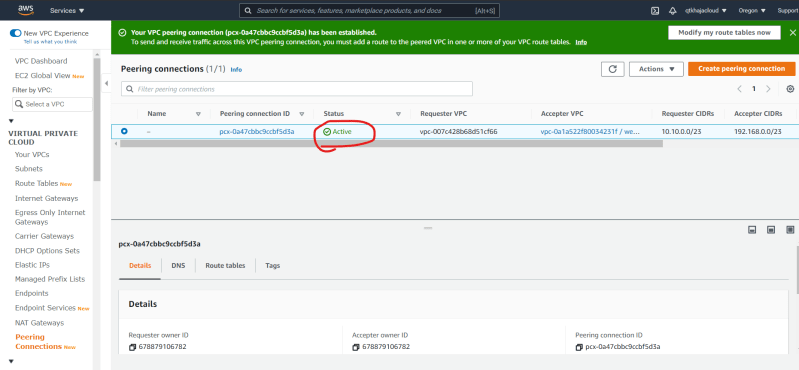
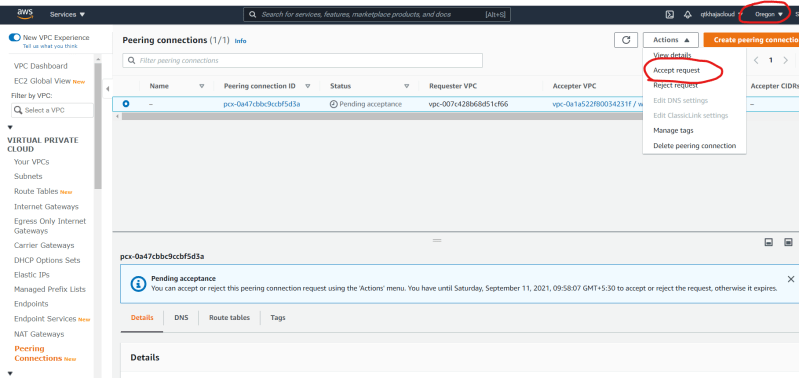
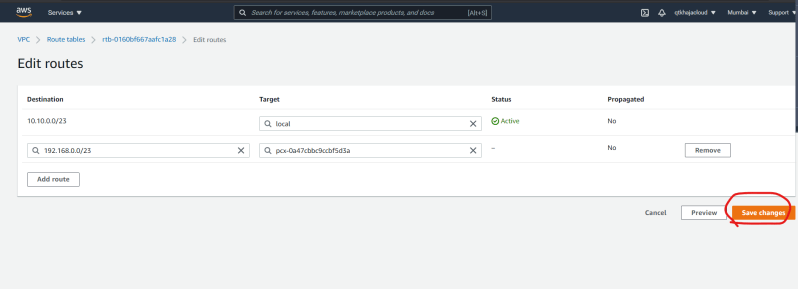
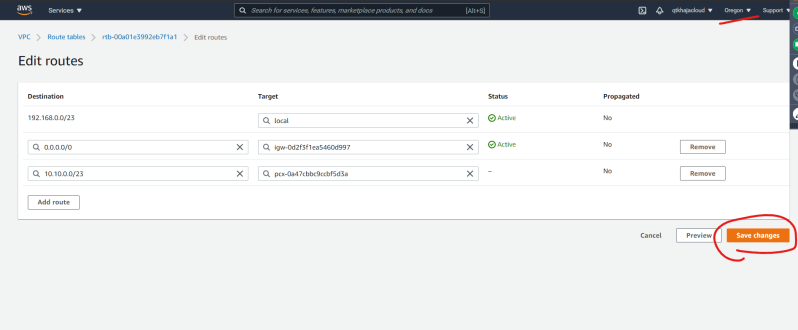
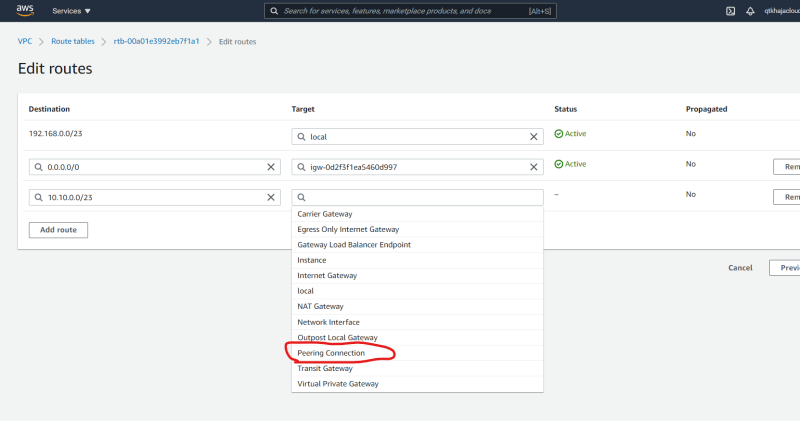


* Network ACL allows us to write allow and deny rules and can be associated with subnet
* Each Network ACL rule will have
  + priority Number: lower the number higher the priority
  + protocol:
  + source/destination address
  + source/destination port
  + Action: Allow/ Deny
* Let’s try to create a NACL rule for
  + web subnet which allows 22,80 port from anywhere
  + all connections from vpc will be allowed
  + the other traffic is denied
* When we write NACL rules, don’t use consecutive numbers for priority number 
* Now let’s try to create a NACL for private subnets which
  + allow all communication from vpc and denies everything else
* Let’s try to create the following vpc in us-west-2 (oregon) 
* Ensure both the subnets are public
* Subnet should allow 22, 80, 443 connections from anywhere
* Now create an ec2 instance with public ip in web1/web2 
* Now lets create the following vpc in ap-south-1 (mumbai) 
* Ensure both the subnets are private.
* We cannot establish the connection between two ec2 instances in two different vpcs. it is possible only if they have public ip address
* In many case we would want connectivity b/w ec2 instances in different vpcs but privately.
* AWS supports peering connection
* So lets create a peering connection b/w Mumbai vpc and oregon vpc 
* VPC peering can be created b/w any two vpcs where cidr’s donot collide 
* Now the vpc in oregon need to accept the peering request 
* Now we should configure route tables to use the peering connection 
* Now we should be able to establish private connectivity b/w ec2 instances 