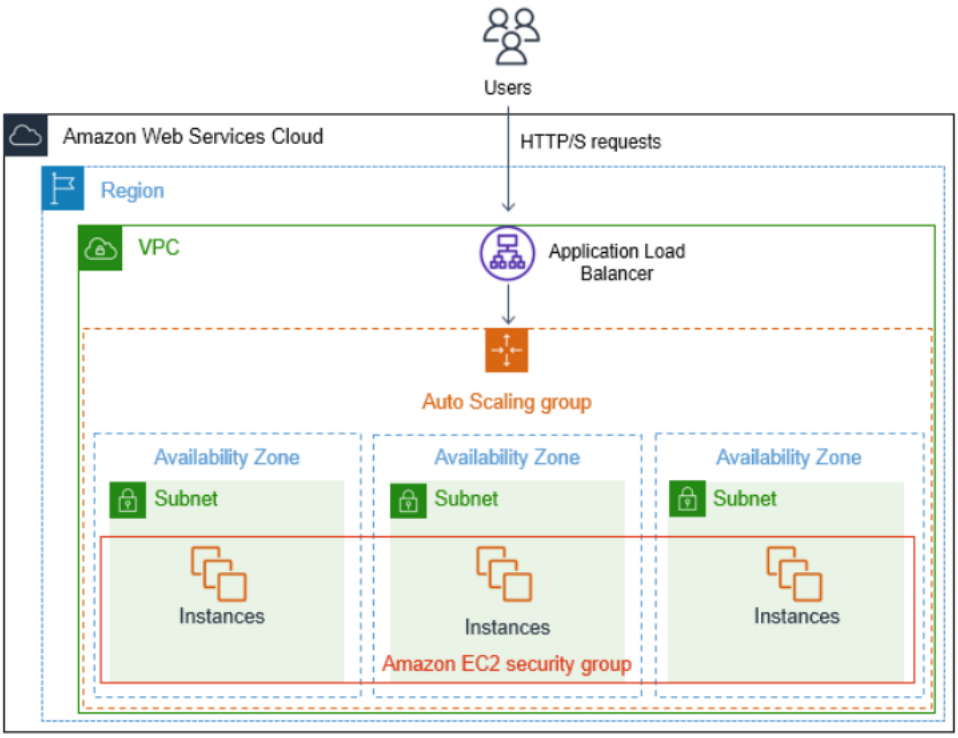
# PRODUCTION ENVIRONMENT SETUP WITH AWS CLOUD

**Architecture:**




**Prerequisites:**

**-VPC**

**(virtual private cloud this vpc creation includes the private & public subnets, security groups, route table & Internet gateway)**

**- Auto Scaling Group**

**( First we configure the create launch template & attach the template to Auto Scaling group)**

**-Load Balancer**

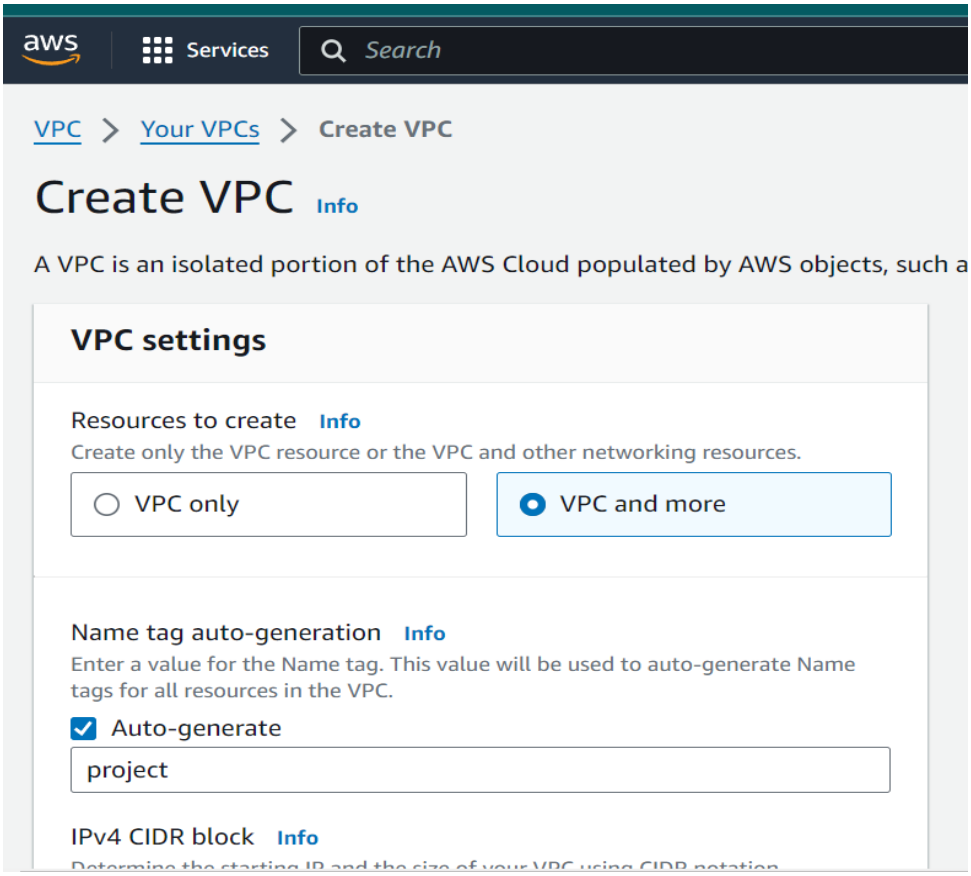
**(First we need to configure the target group then configure target group in load balancer)**

**-Application**

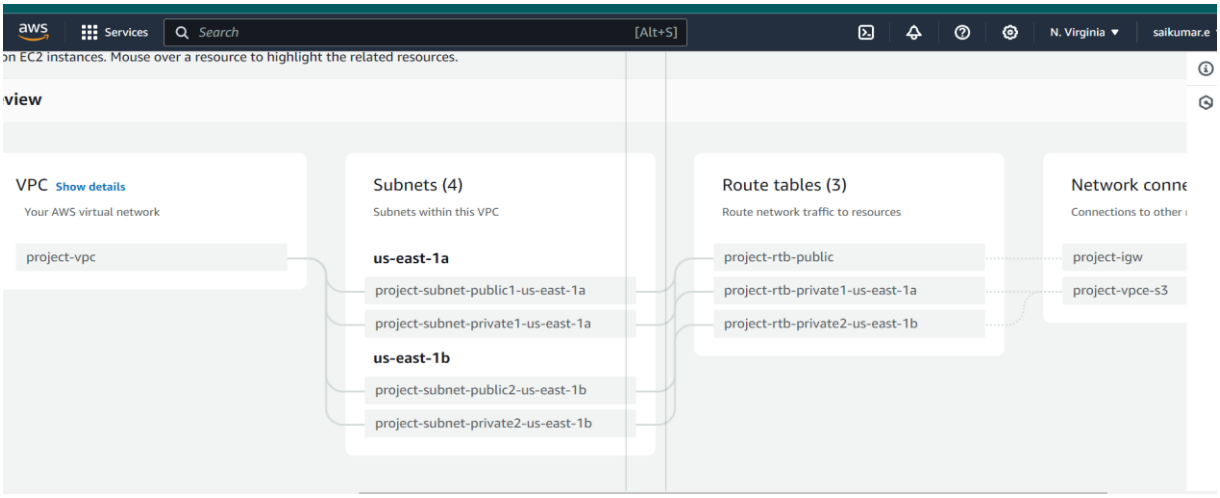
**(we need to deploy the application in two different subnets & two different servers it will accessible through application load balancer based on user requests auto-scaling will increase the servers based on the configuration)**

**STEP 1: Create the VPC(virtual private cloud) then the subnets, security groups automatically setup created in AWS.**

1. **Create a VPC then the below requirement will created automatically.**



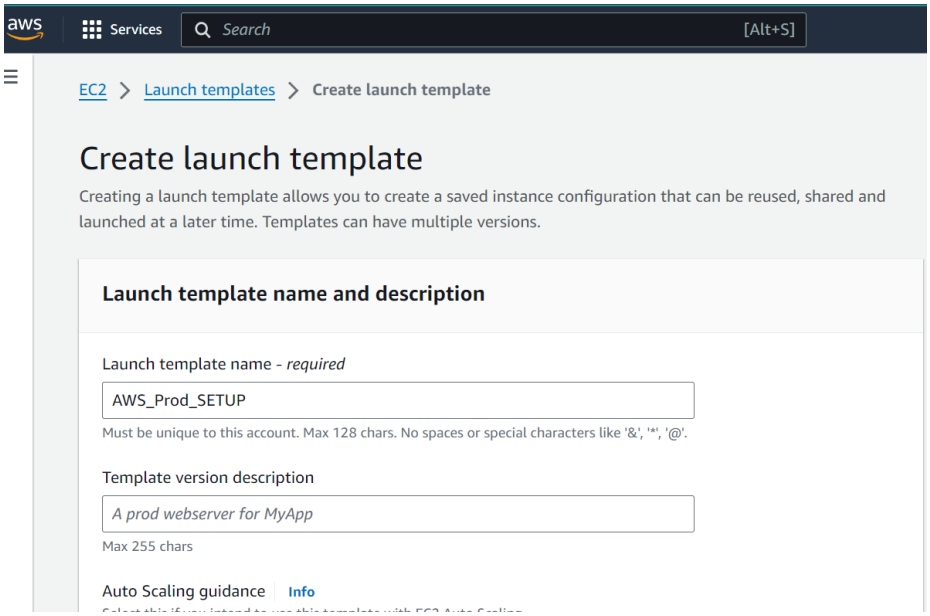

2. **Public Subnet & Private Subnet created automatically.**
3. **Route table will be allocated automatically.**
4. **Security groups created.**
5. **Attached the Internet gateway.**



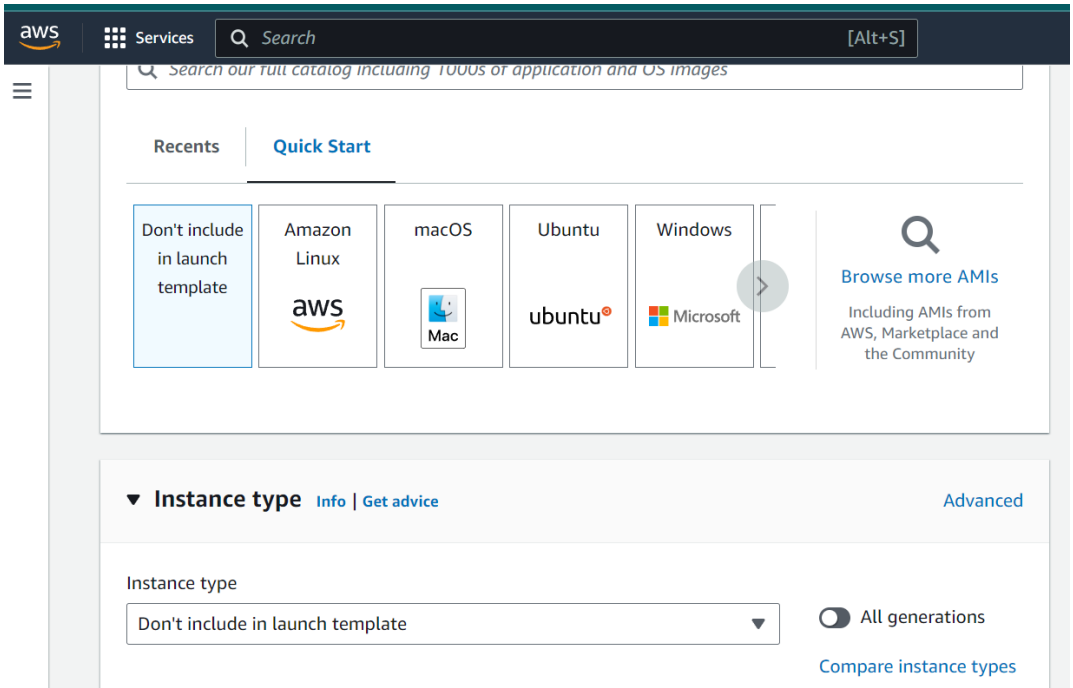

**The above image indicate the how the Subnets, Route tables, Internet gateway created automatically when we create VPC above methods.**


**STEP 2: Auto Scaling group**

1. **Auto scaling mainly used to manage the application performance based on customer usage.**
2. **When the customers accessing the application based upon configurations autoscaling service will create another application nodes to manage the traffic. When customer accessing very less autoscaling will dismantle the extra nodes based on configuration.**
3. **First need to create a launch template.**

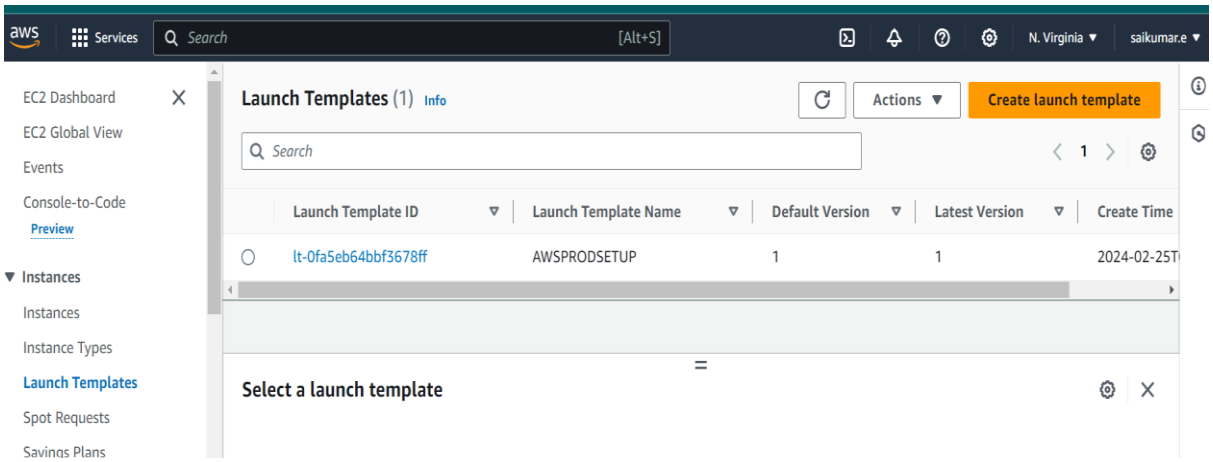



4. **Then create the servers which flavour required like ubuntu, cent-os etc..**

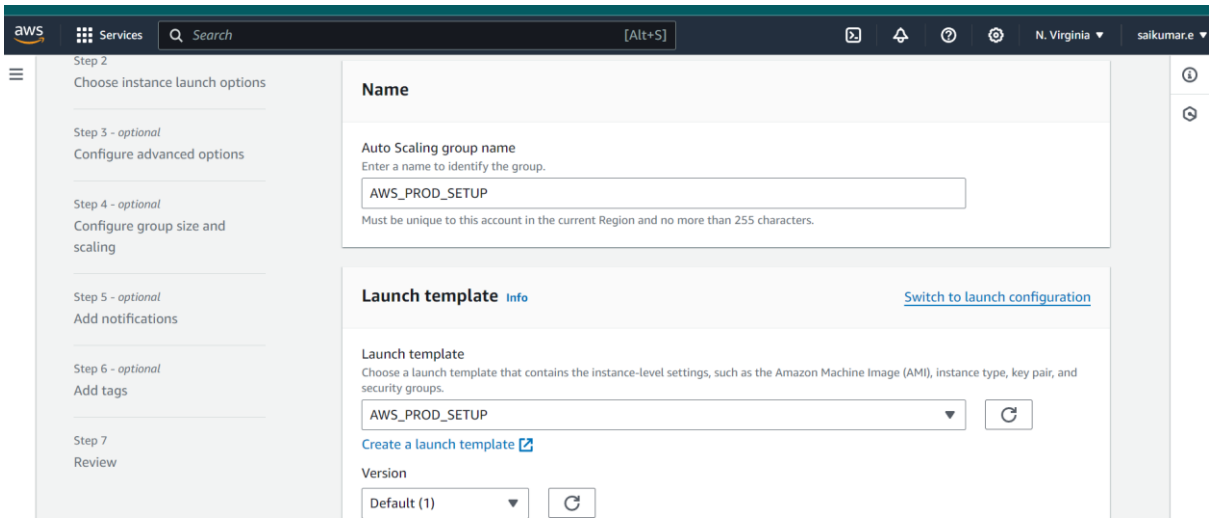



5. **Based upon you requirement you can create the KeyPairs ( If you created the created the server in public subnet then download ppk file or If you create the private subnet download the pem file to connect the server with ssh connectivity with help of jumphost.**

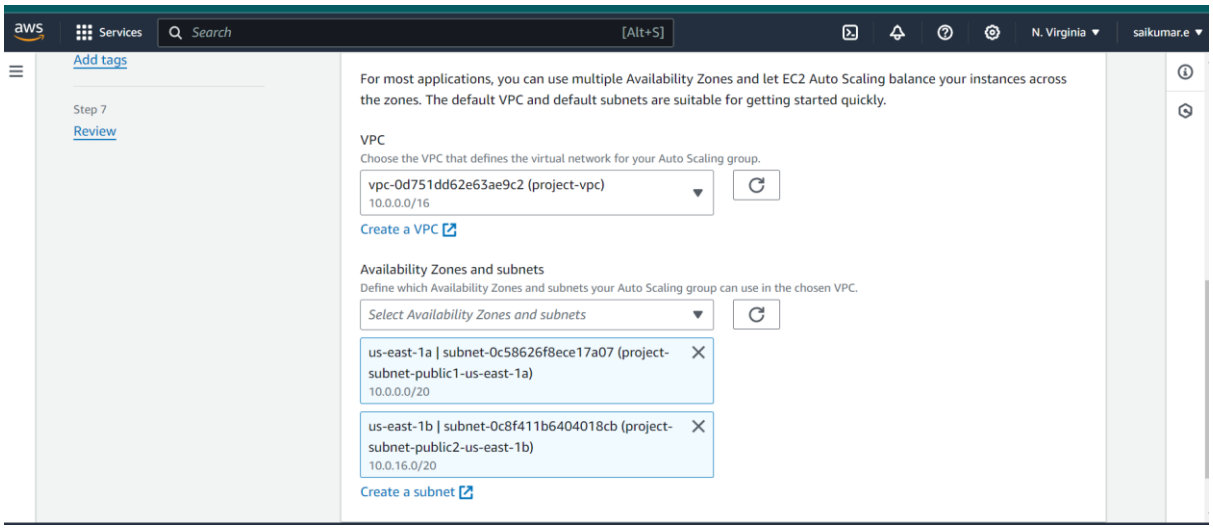
6. **Lauch template created successfully. (you can in the below Image).**



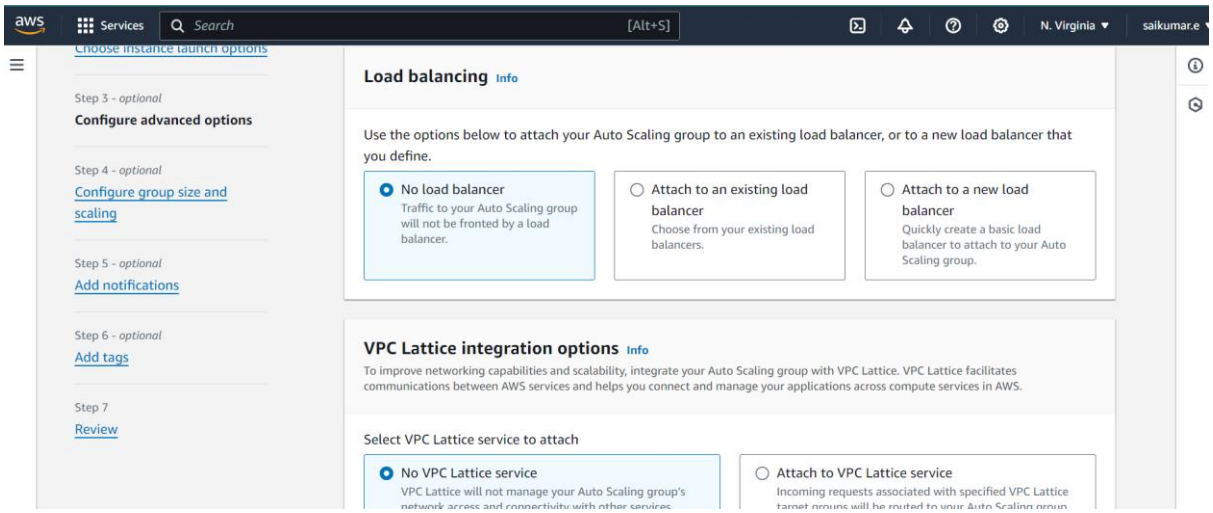

7. **Then go to auto scaling configuration & use the launch temple to create auto-scaling group.**




8. **Add the VPC & select the private or public subnets where we need to host the application based on our requirement.**



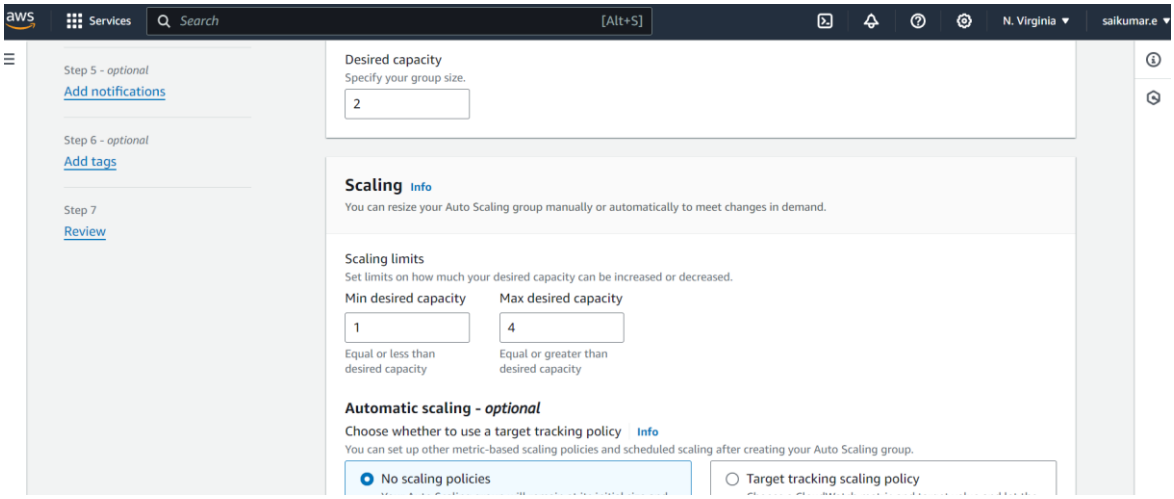
9. **Select the load balancer based on requirement.**



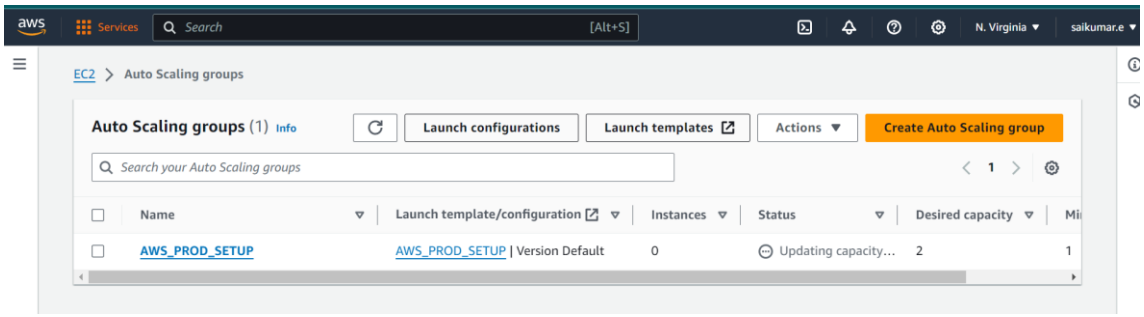

10. **Select the scaling limits based on your application traffic.**

    **Desired capacity indicates the size of scaling group(initially it will launch the two servers)**

    **Max desired capacity indicated when the application usage increase automatically auto scaling enable the extra application nodes up to Max desired capacity( max limit 4 servers on our requirement)**

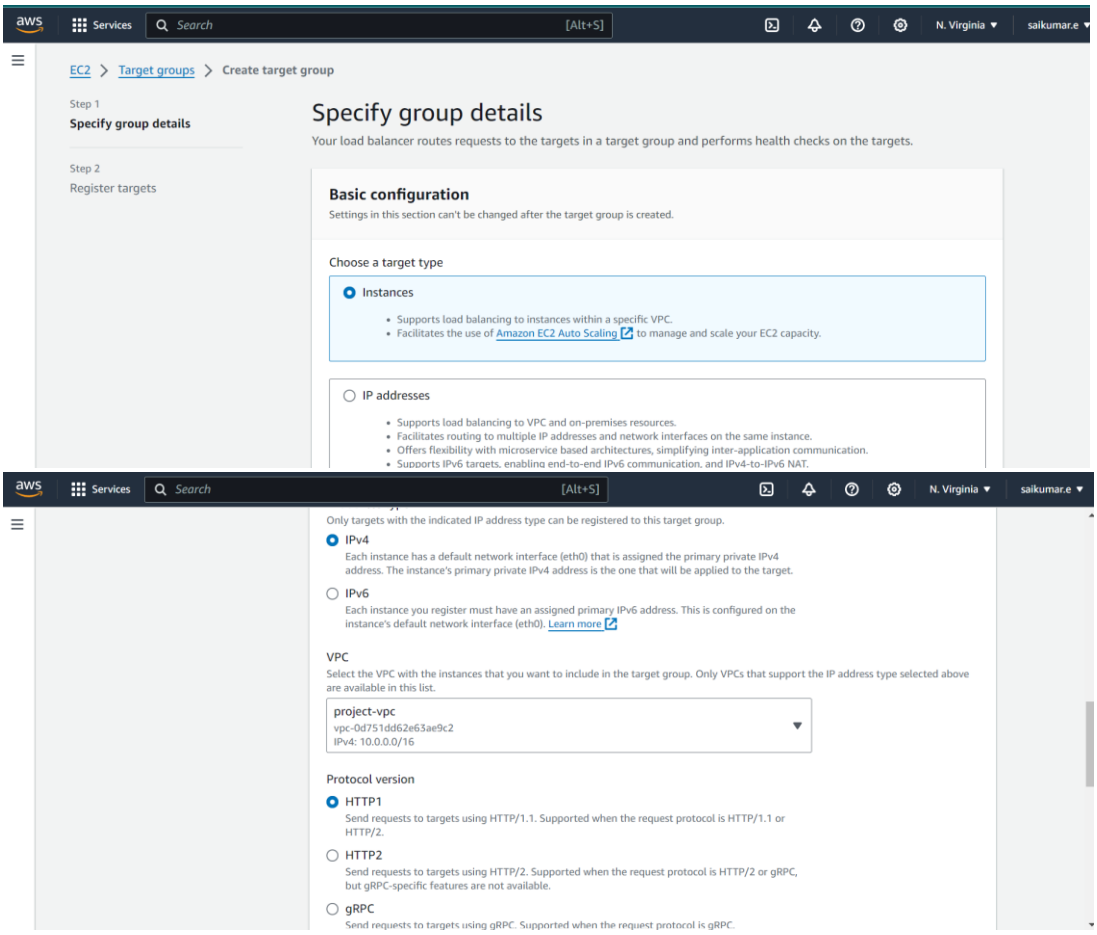



11. **Auto-scaling group created successfully.**

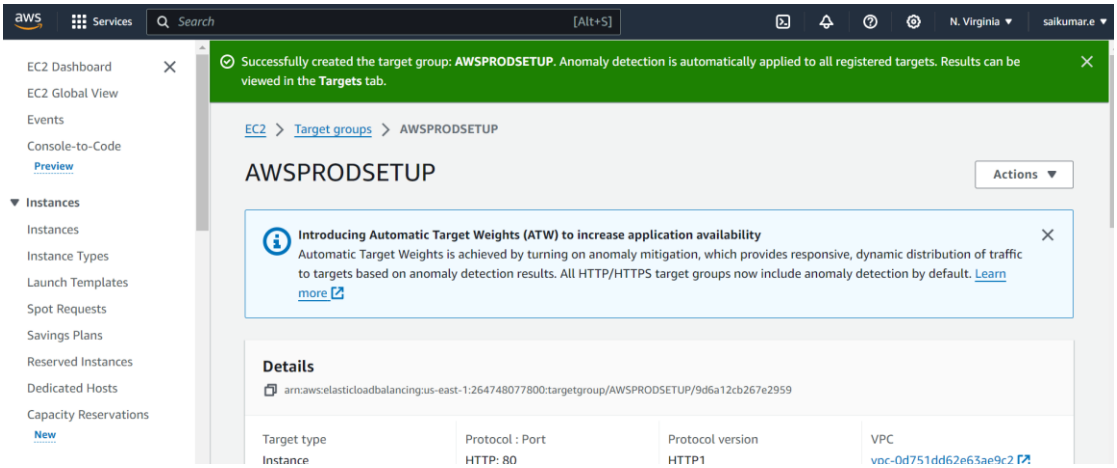


## STEP 3: LOAD BALANCER

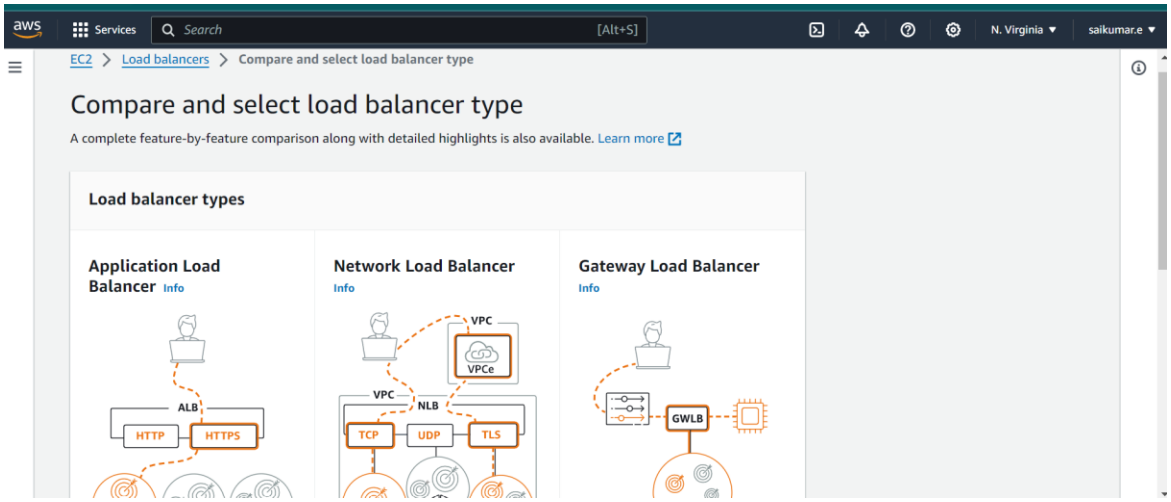
1. **Load balancer is used to balance the application traffic based on customer usage.**
2. **Load balancer share the customer requests for application nodes.(Node 1 & Node2)**
3. **First need to create Target group & add the vpc.**

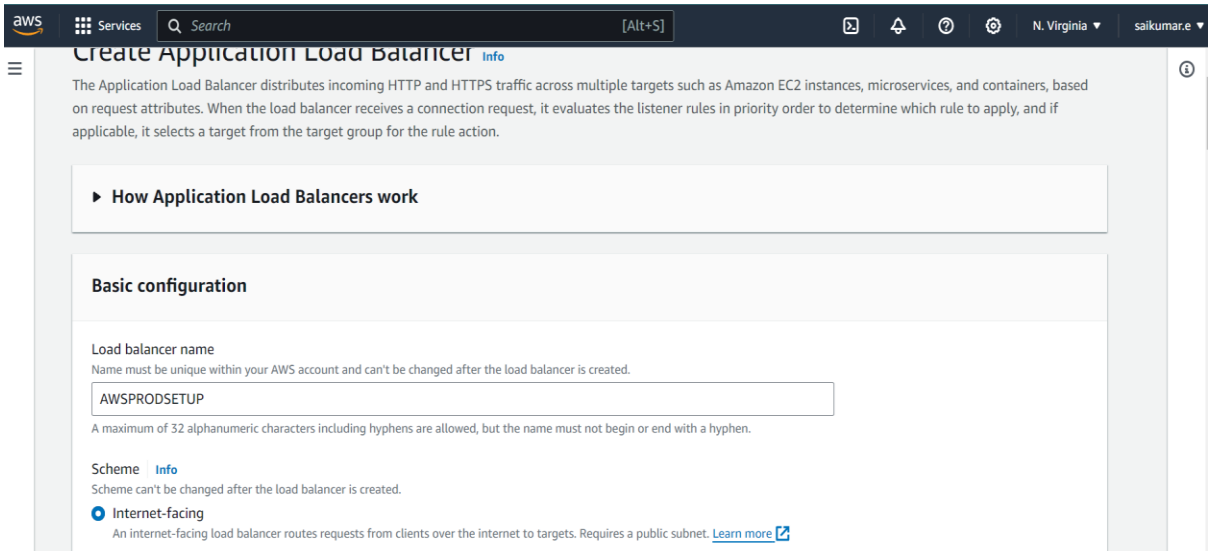



4. **Target group created successfully.**

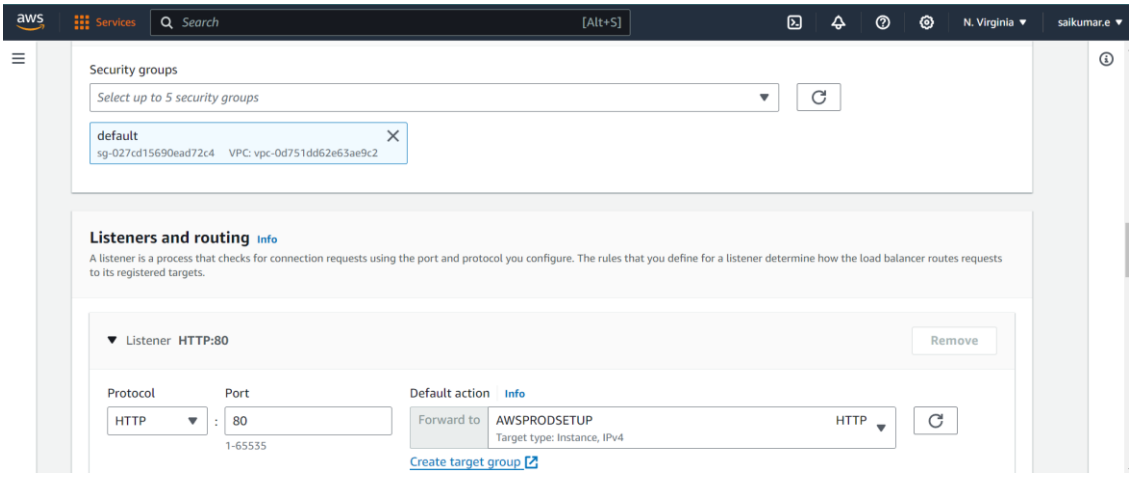


5. **Add the same target group in Load balancer configuration & Select the application load balancer.**

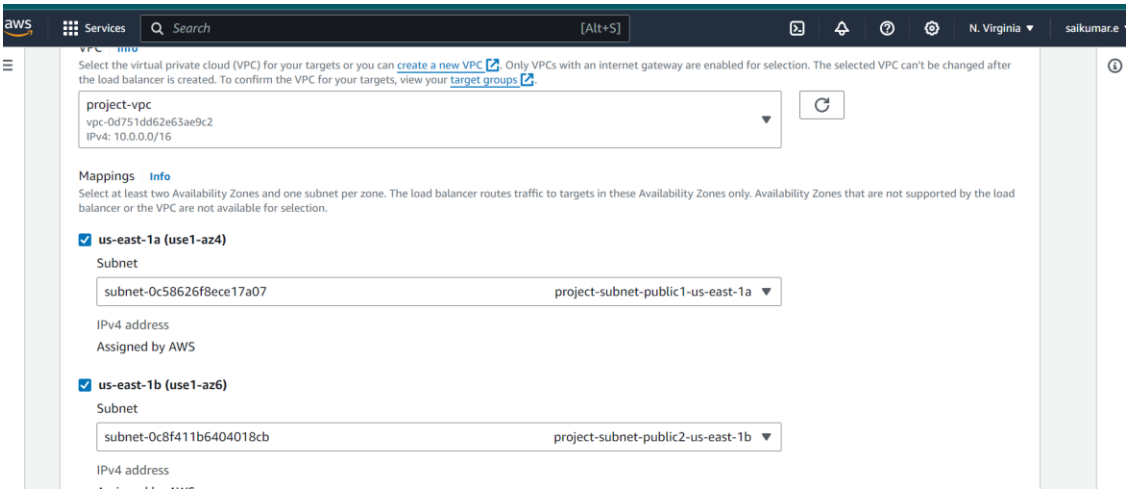


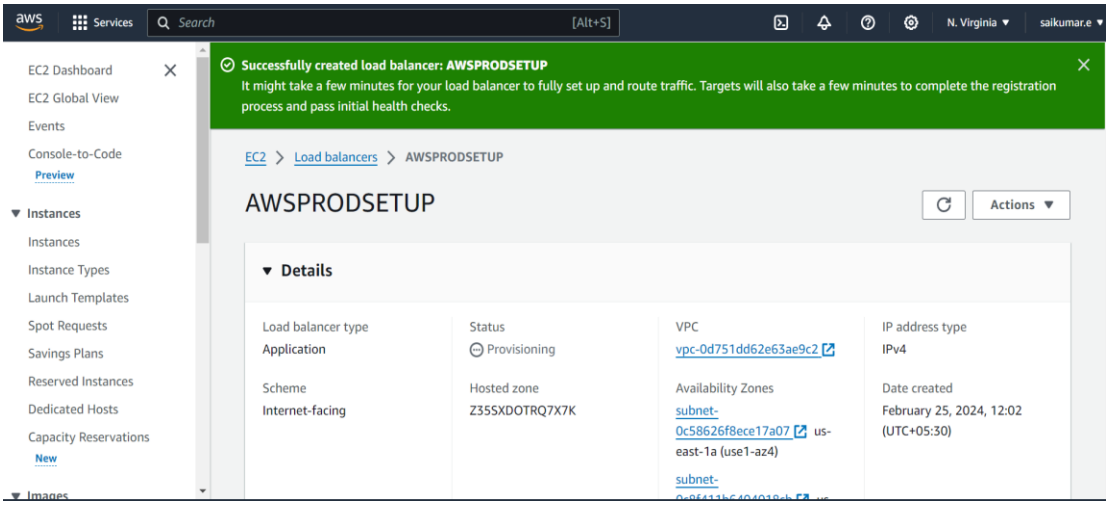

6. **Add the security group , Listeners & routing.**



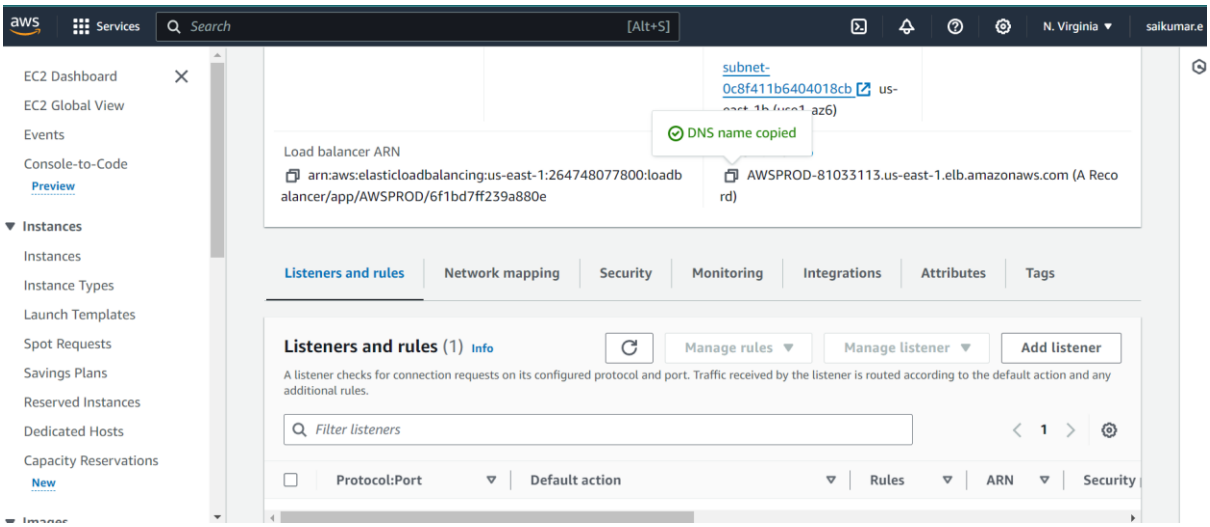
7. **Add the VPC & select the subnets (public or private ) based on you requirement.**




8. **Successfully created the Application load balancer.**




9. **We can use the DNS name as domain for the application & to manipulate the Application Ip addresses to Domain Name from Application load balancer.**




**STEP 4: Application Deployment.**

1. **Host your application in application EC2 instances like a small python script with port. So that Ip & port configured in load balancer due to this Application IP & Port manipulate into domain name (DNS name).**
2. **Customer access the application with domain name only & so the requests reaches to Application load balancer.**
3. **Based on load balancer configuration (IP & Port ) customer gets the output in User Interface.**