**AWS PROJECT**

**VPC with public private subnets in production**

**About the project: -** This example demonstrates how to create a VPC that you can use for servers in a production environment to improve resiliency, you deploy the servers in two availability zones, by using an auto scaling group and an application load balancer. For additional security, you deploy the servers in private subnets. The servers receive requests through the load balancer. The servers can connect to the internet by using a NAT gateway. To improve resiliency, you deploy the NAT gateway in both the availability zones.

**Overview: -**

1. The VPC has public subnets and private subnets in two availability zones.
2. Each public subnet contains a NAT gateway and a load balancer node.
3. The servers run in the private subnets, are launched and terminated by using an auto scaling group and receive traffic from the load balancer
4. The servers can connect to the internet by using the NAT gateway.

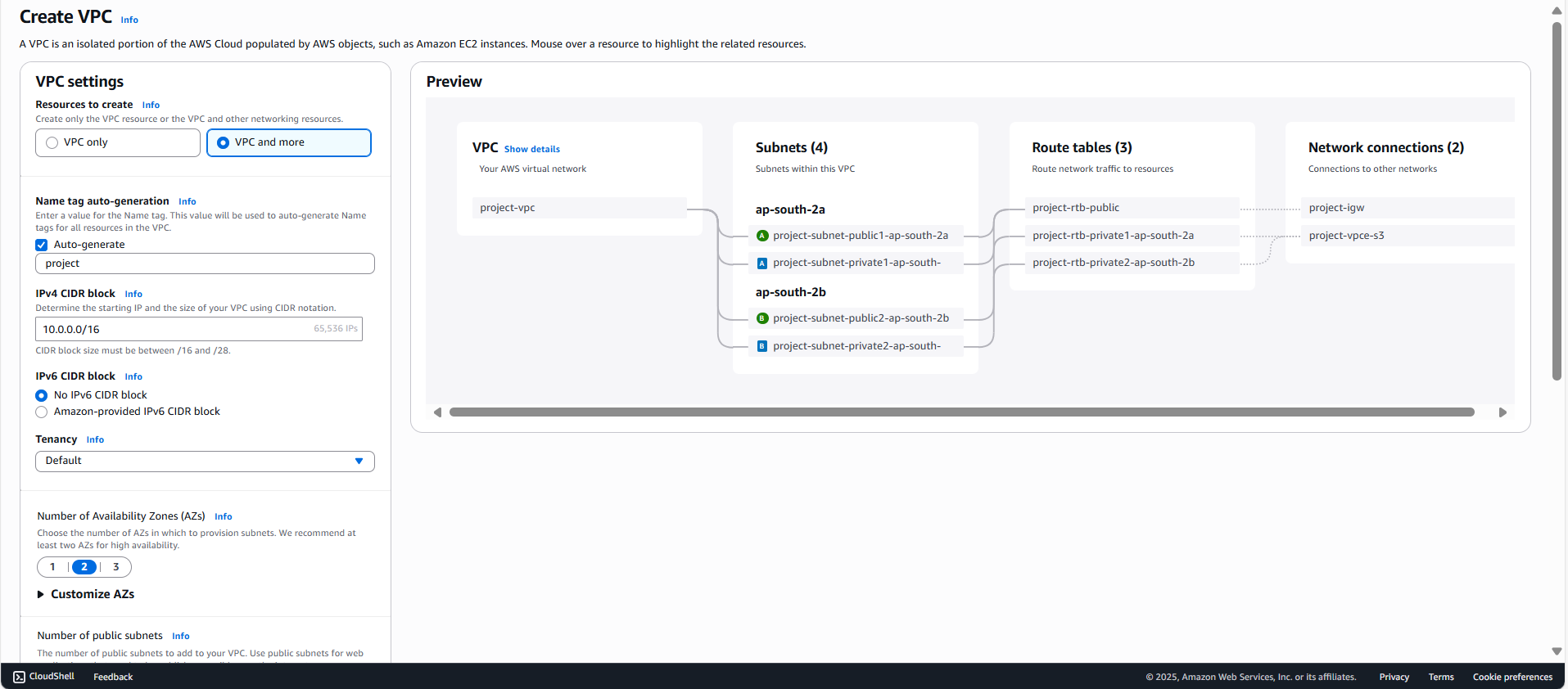
**Before we start: -**

1. Auto scaling group
2. Load balancer
3. Target group
4. Bastion host or Jump server

**Creation of VPC: -**

**Step 1.** Login to AWS console and search for VPC and click on create VPC

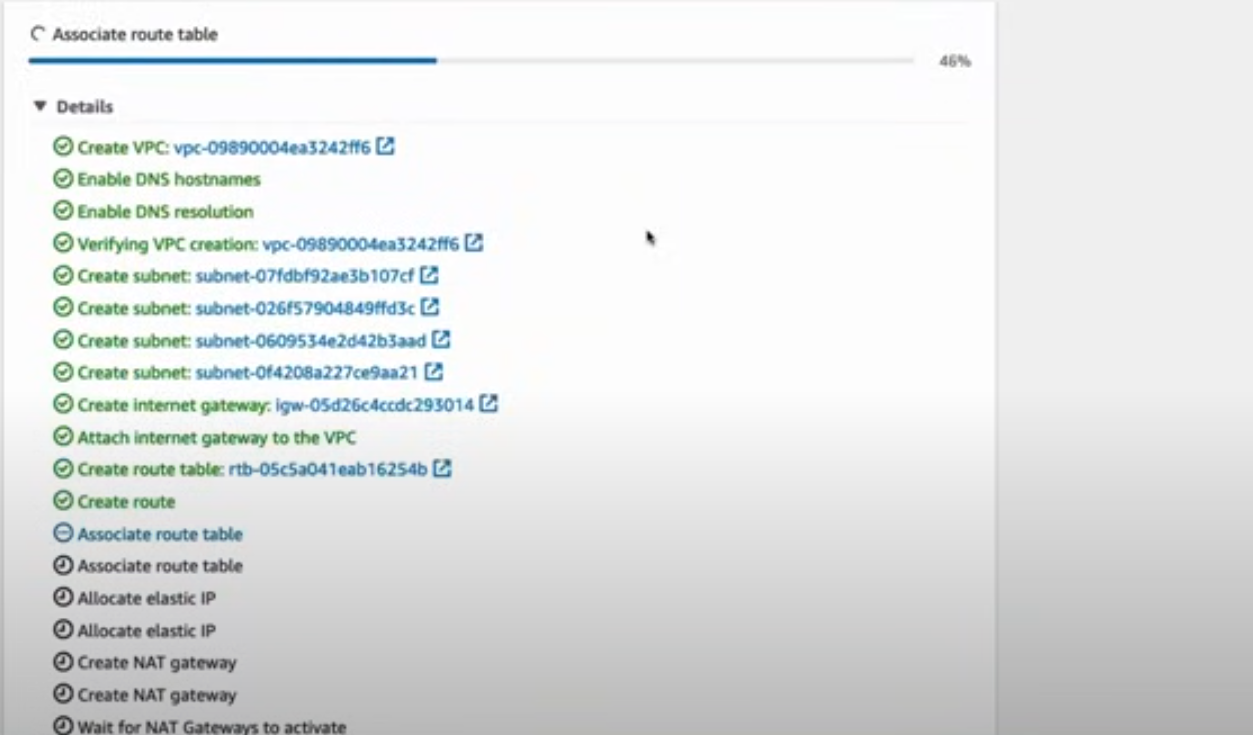
**Step 2.** Select VPC and more to create



**Step 3.** Select no. of availability zones, public subnets and private subnets as **“2”** and “**1”** NAT

gateway per availability zoneand VPC endpoints as **“NONE”**

**Step 4.** We can see the bunch of services which are creating along with VPC creation



**Step 5.** Then create autoscaling groups by searching ec2 and select auto scaling group. Then

click on create auto scaling group and click on Create a launch template and select the required fields and click on create auto scaling group

**Step 6.** Again, search for auto scaling groups and select the launch template and click on next

and select the VPC and availability zones and subnets as private subnets as the ec2 should be in private subnet and click on next. Then select the group size as 2 with minimum as 1 and maximum as 3 or 4 or 5 which needs to be auto created and at the last page click on create auto scaling group

**Step 7.** Then check whether the 2 ec2 instances in 2 availability zones are created or not

along with the auto scaling group

**Step 8.** Inorder to install the application inside the server go to ec2 instance and try to login

to it but we cannot find the public IP as it is so secure that everyone cannot able to login, here the bastion host comes into place which is the mediator between the private subnet and the external user

**Step 9.** Now create a bastion host by searching ec2 and click on create instance and select

the VPC which is created but not the default VPC and enable the auto assign public IP and click on launch instance

**Step 10.** Now login to the bastion host ec2 instance by using **SCP 🡪 secure copy** command

**Scp -i path-of-.pem-file path-of-.pem-file ubuntu@public-ip-of-bastion-host/home/ubuntu**

**Step 11.**  Login to the bastion host by using below command and check whether the pem file

Is available or not by using ls command.

ssh -i .pem-file ubuntu@public-ip

**Step 12.** Now connect to the other ec2 instances which are created by auto scaling group by

Using

ssh -i pem-file ubuntu@private-ip-of-the- other-ec2 -instance

**Step 13.** Create a html file using vim index.html and run the python server by using

python3 -m http.server 8000

**Step 14.** Now search for ec2 and click on load balancer and select application load balancer.

It should be in internet facing and select the VPC that was created and select the 2 availability zones for the public subnets with the Created VPC and select the security group

**Step 15.** Then click on create a target group and select the ec2 instances and the port as

same as the one which is using to host and click on include as pending and click on create target group

**Step 16.** And select the target group in load balancer which is just created and click on create

load balancer

**Step 17.** Copy the DNS name which is in ALB and paste it in the browser to host the

application