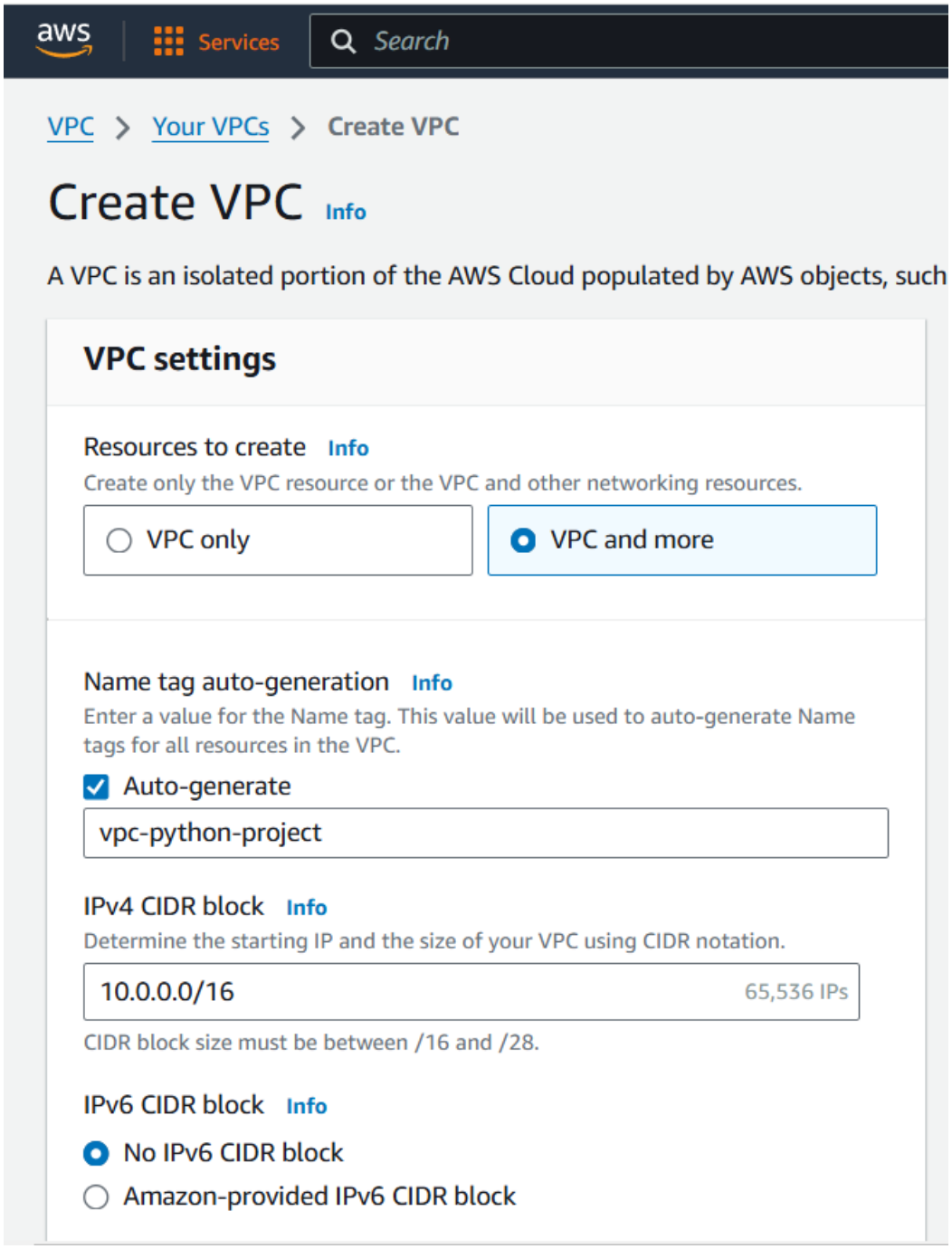
1. **Creating VPC**



Tenancy Info

Default

Number of Availability Zones (AZs) Info

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1	2	3

▶ Customize AZs

Number of public subnets Info

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0	2

Number of private subnets Info

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

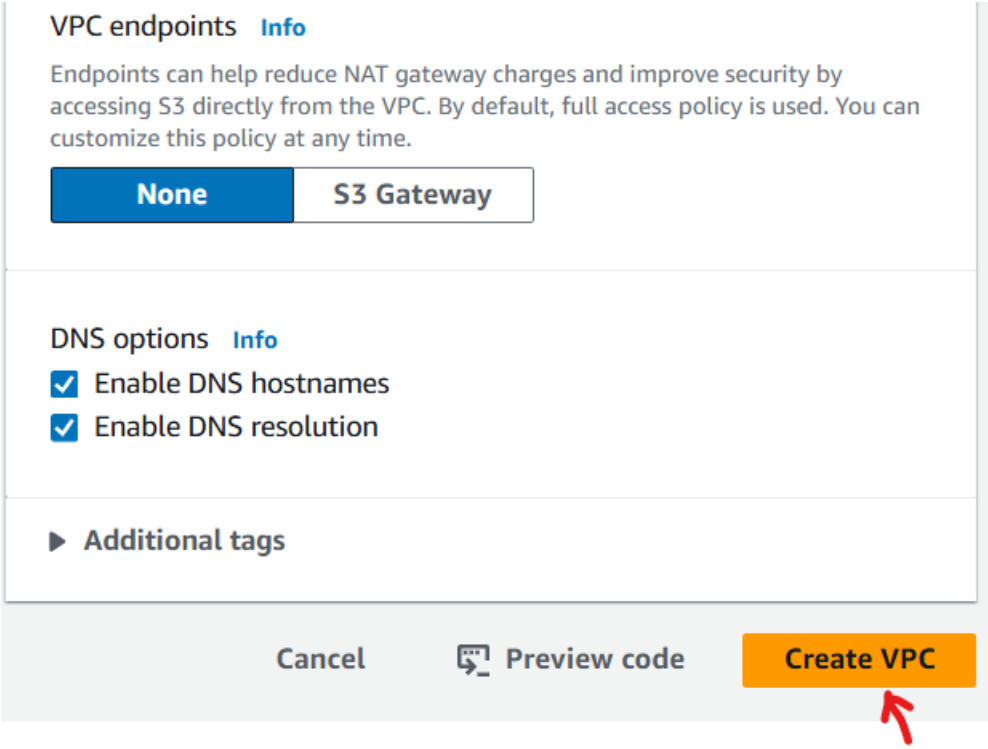
0	2	4

▶ Customize subnets CIDR blocks

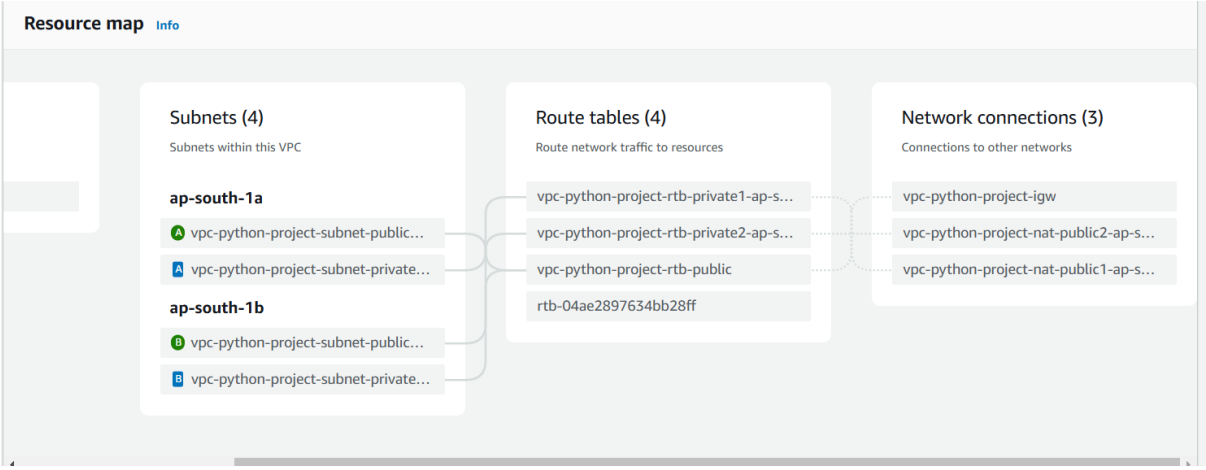
NAT gateways ($) Info

Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway

None	In 1 AZ	1 per AZ

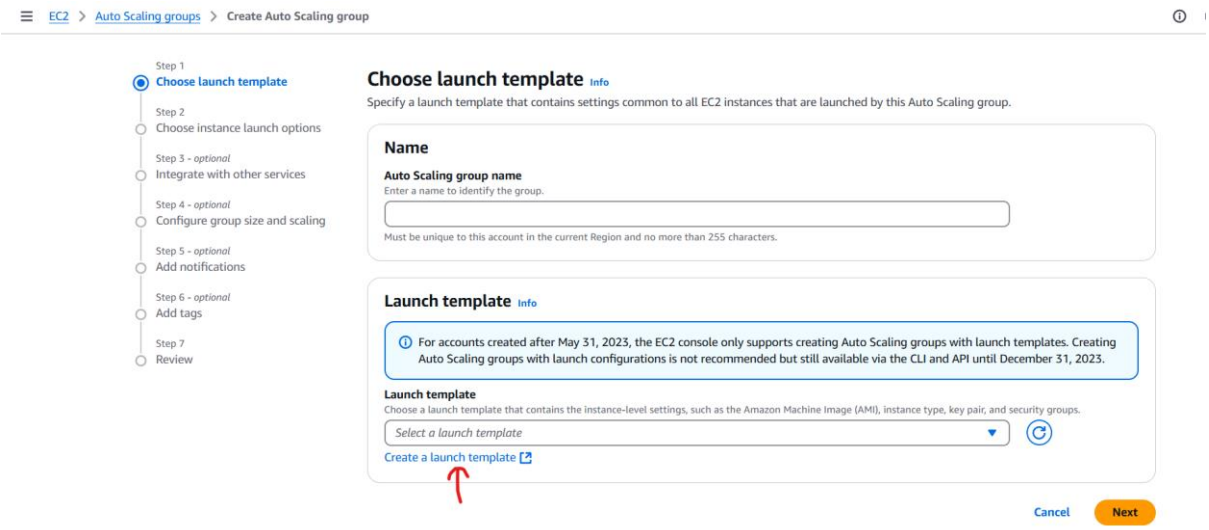


- **Resource map will be look like below post creation of VPC.**

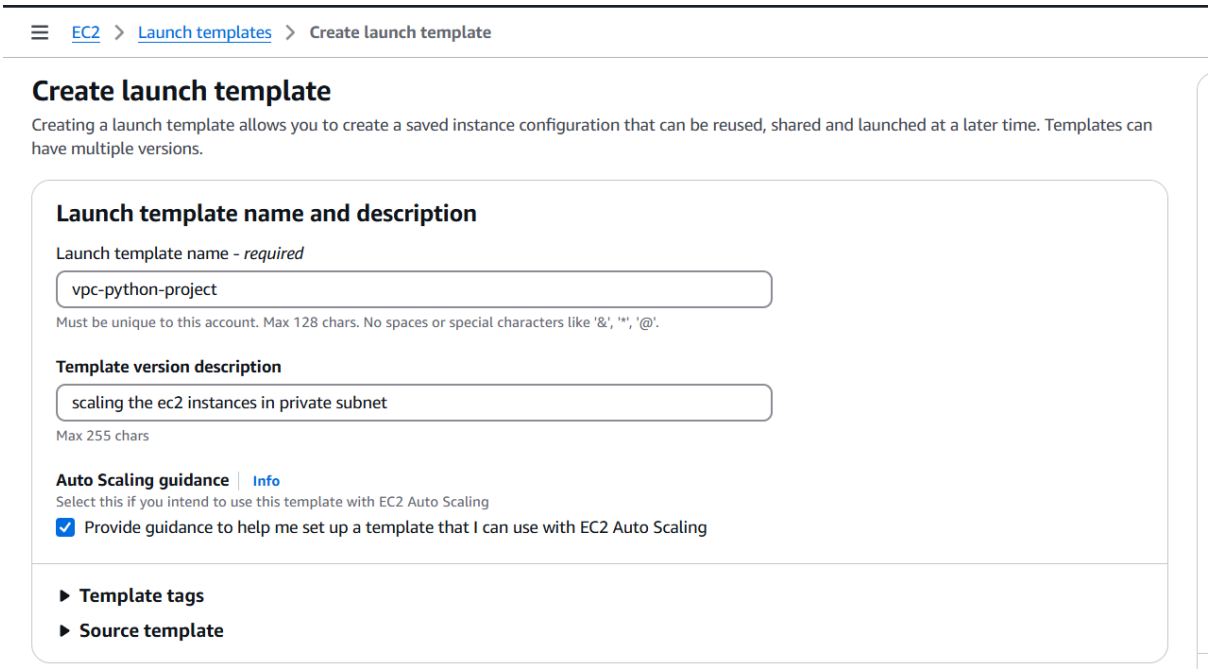


2. Creating launch template for auto scaling group

- **You need to create launch template first before creating the auto scaling group.**



- **Creating launch template**



Application and OS Images (Amazon Machine Image) - required Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | My AMIs | Quick Start

Recently launched

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20240801
ami-0522ab6e1ddcc7055
2024-08-01T15:04:23.000Z Architecture: 64-bit (x86) Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred

Description

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	
x86_64	ami-0522ab6e1ddcc7055	Verified provider

Instance type Info | Get advice

Advanced

Instance type

t2.micro Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

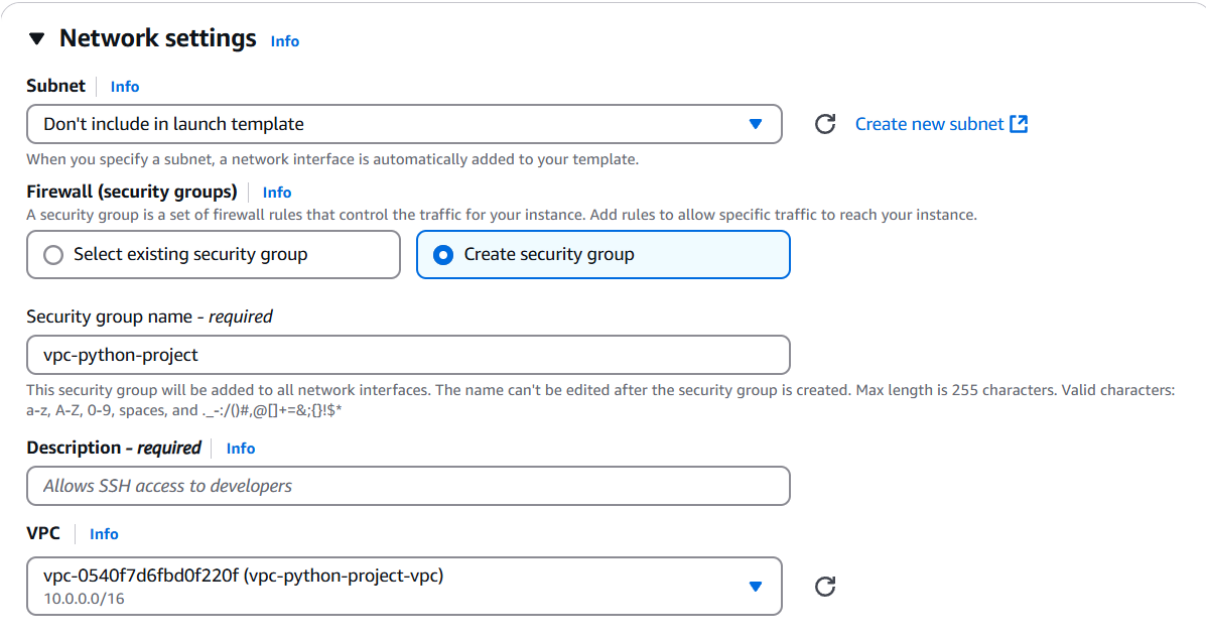
Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

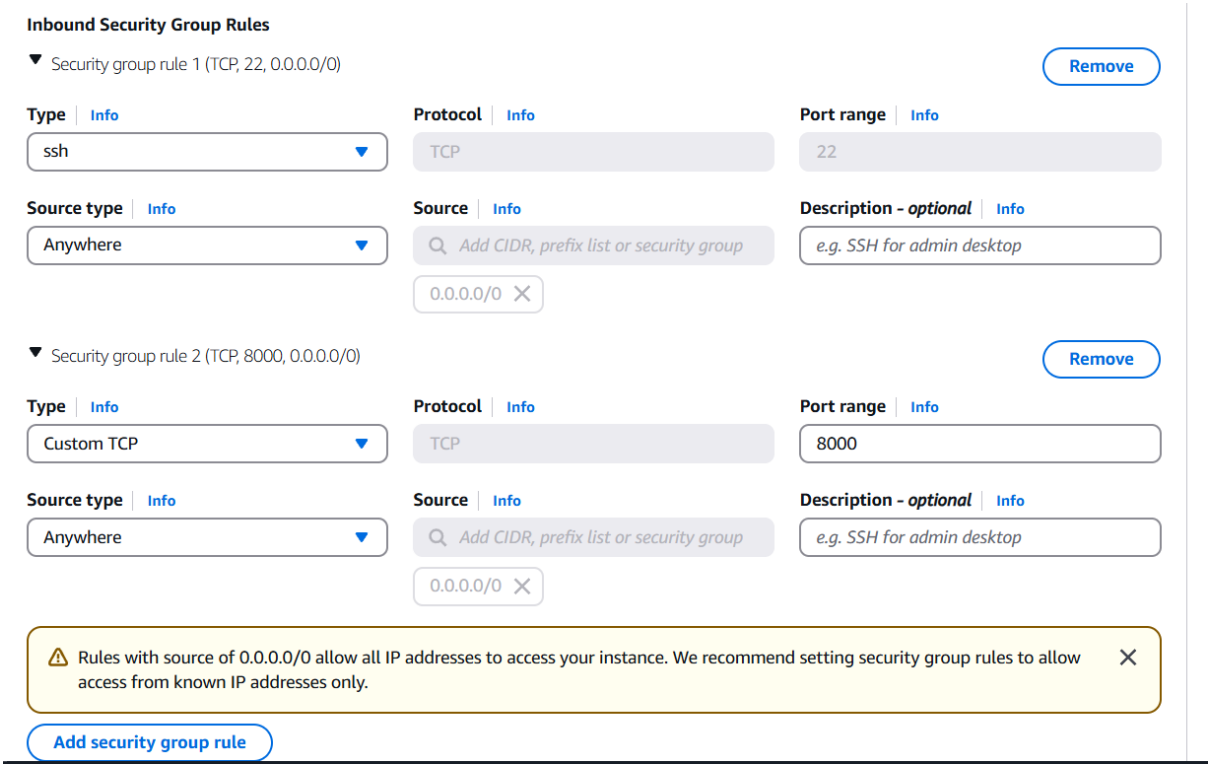
Key pair name

git_key

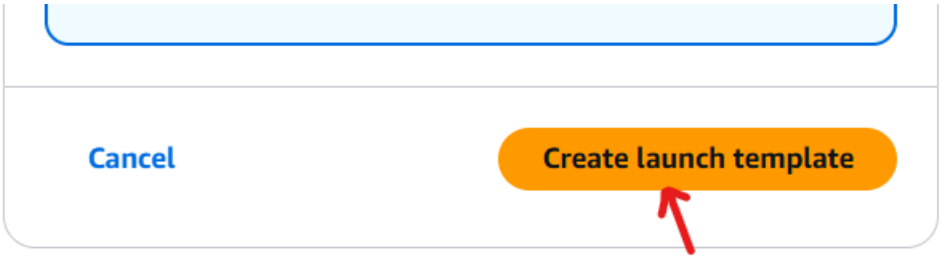
Create new key pair



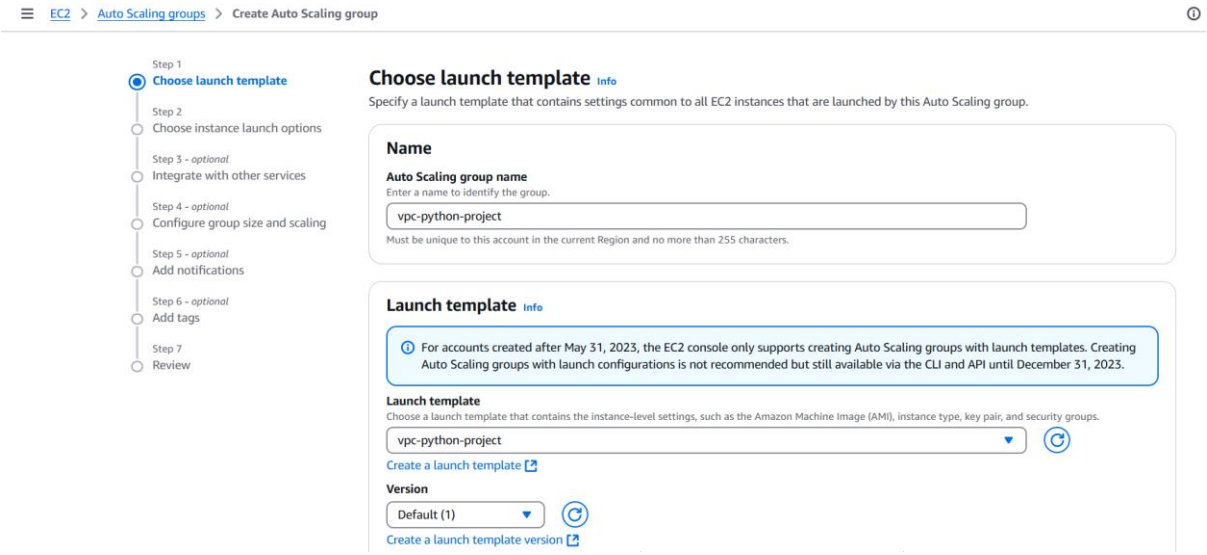
- **Creating inbound security group rule for ssh and application custom port 8000 which we are going to deploy on these instances.**



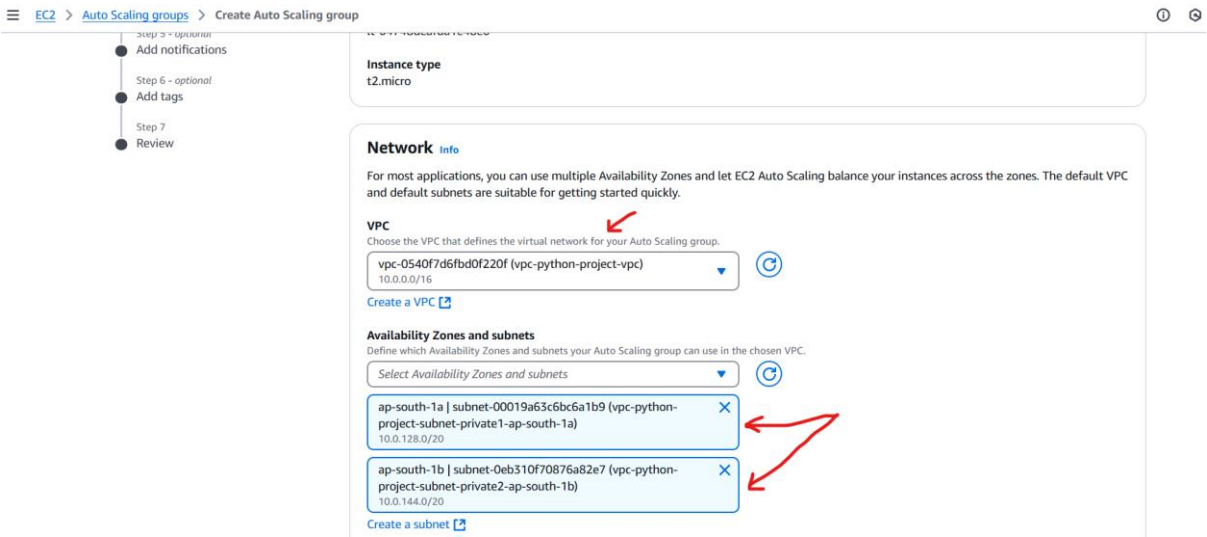
- **Click on create launch template**



3. Creating auto scaling group



- **Select your VPC with private subnet since we have to keep our application secure do not want to expose it over the internet.**



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - *optional*
Integrate with other services
Step 4 - *optional*
Configure group size and scaling
Step 5 - *optional*
Add notifications
Step 6 - *optional*
Add tags
Step 7
Review

Integrate with other services - *optional* Info

Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

Load balancing Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

- (●) No load balancer — Traffic to your Auto Scaling group will not be fronted by a load balancer.
- ( ) Attach to an existing load balancer — Choose from your existing load balancers.
- ( ) Attach to a new load balancer — Quickly create a basic load balancer to attach to your Auto Scaling group.

VPC Lattice integration options Info

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach

- (●) No VPC Lattice service — VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.
- ( ) Attach to VPC Lattice service — Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

Create new VPC Lattice service

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 4 - *optional*
Configure group size and scaling
Step 5 - *optional*
Add notifications
Step 6 - *optional*
Add tags
Step 7
Review

Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.
Units (number of instances)

Desired capacity
Specify your group size.
2

Scaling Info

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits
Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity	Max desired capacity
1	4
Equal or less than desired capacity	Equal or greater than desired capacity

Automatic scaling - *optional*
Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

- (●) No scaling policies — Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.
- ( ) Target tracking scaling policy — Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

EC2 > Auto Scaling groups > Create Auto Scaling group

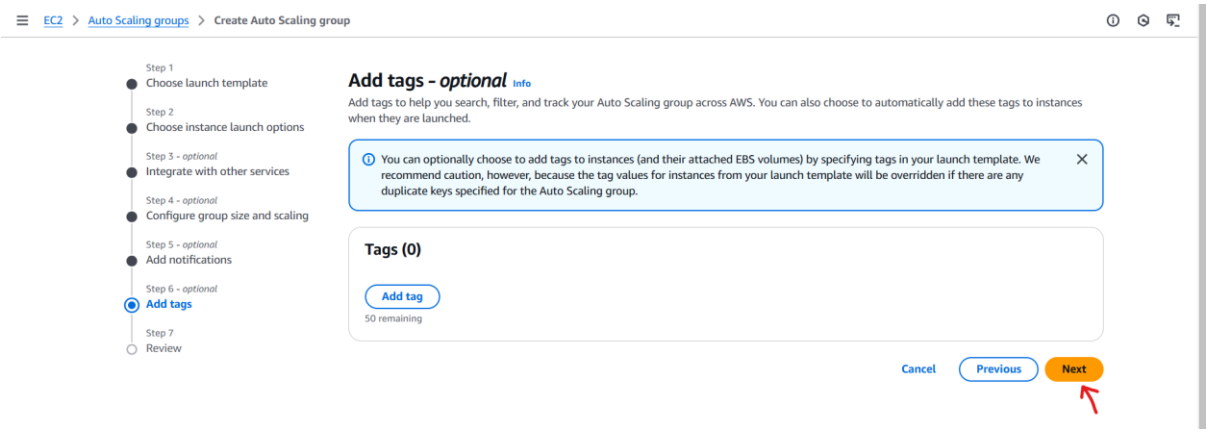
Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - *optional*
Integrate with other services
Step 4 - *optional*
Configure group size and scaling
Step 5 - *optional*
Add notifications
Step 6 - *optional*
Add tags
Step 7
Review

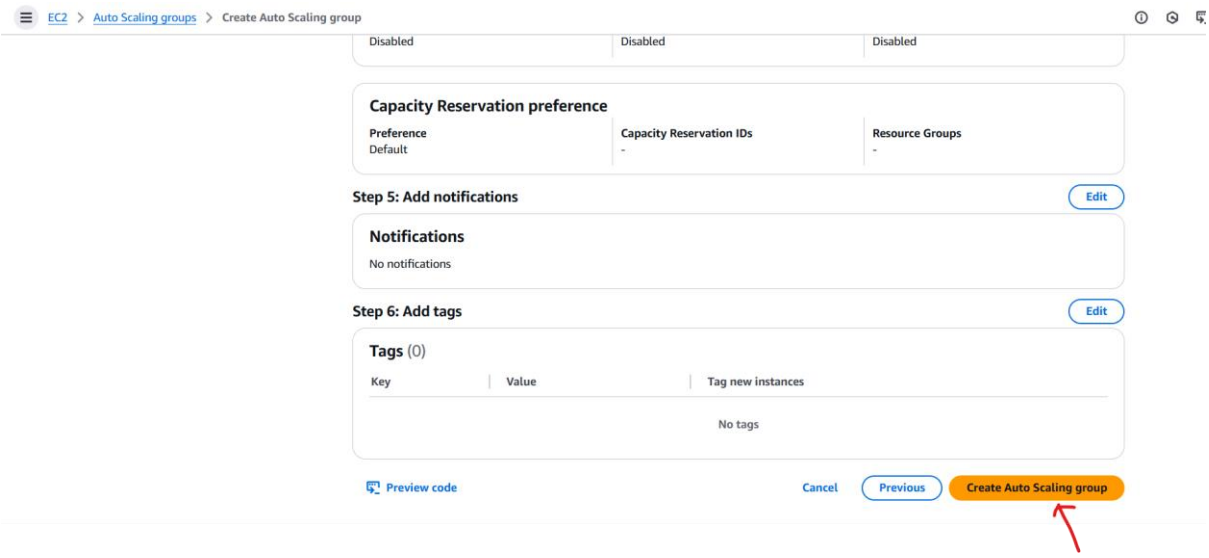
Add notifications - *optional* Info

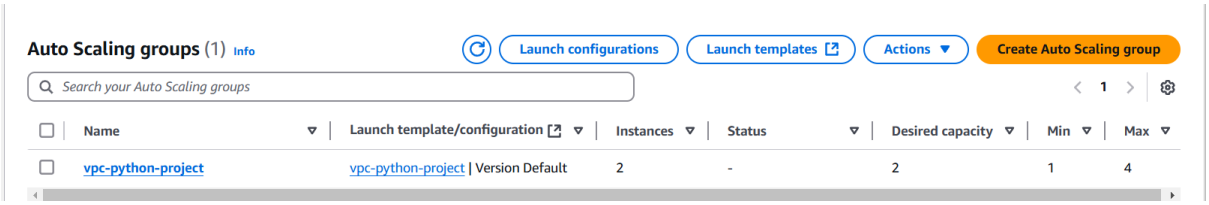
Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

Add notification

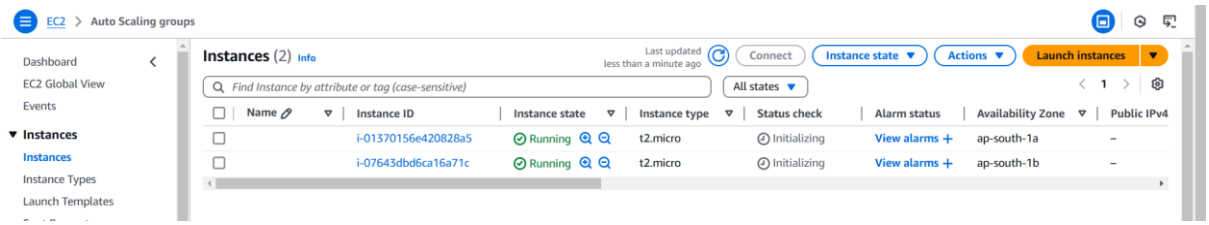
Cancel | Skip to review | Previous | Next



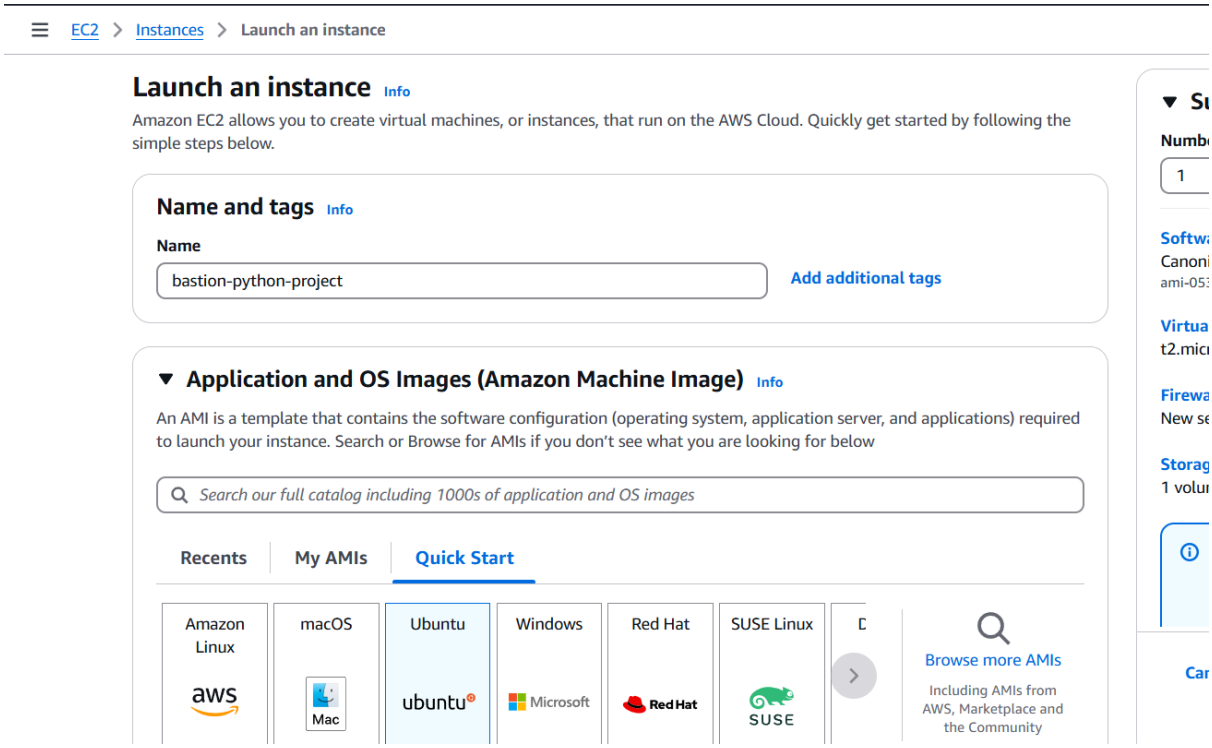




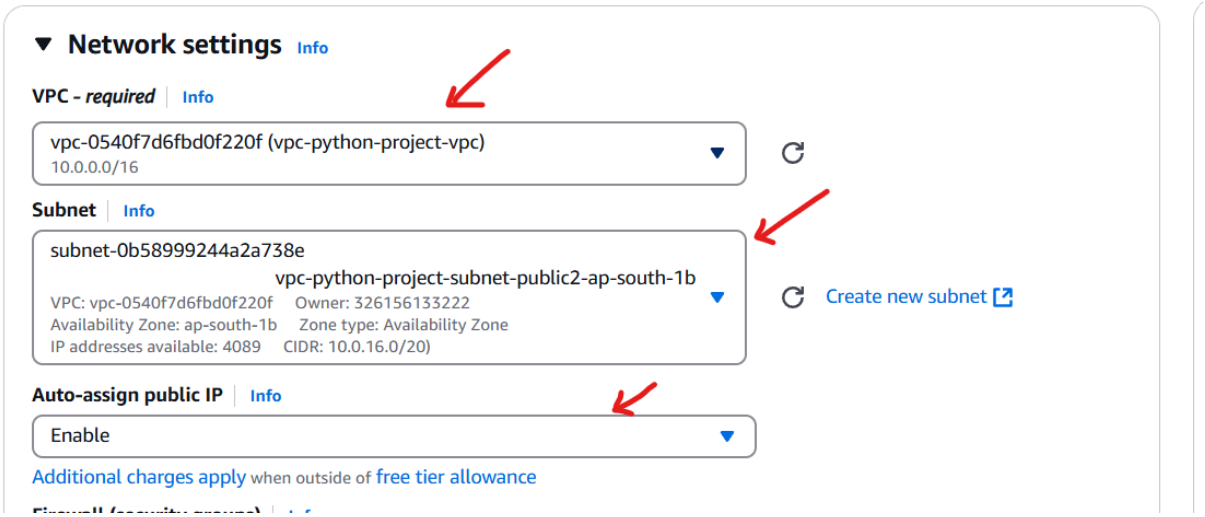
- **Now auto scaling group will create the instances as per desired capacity mentioned with private ips.**

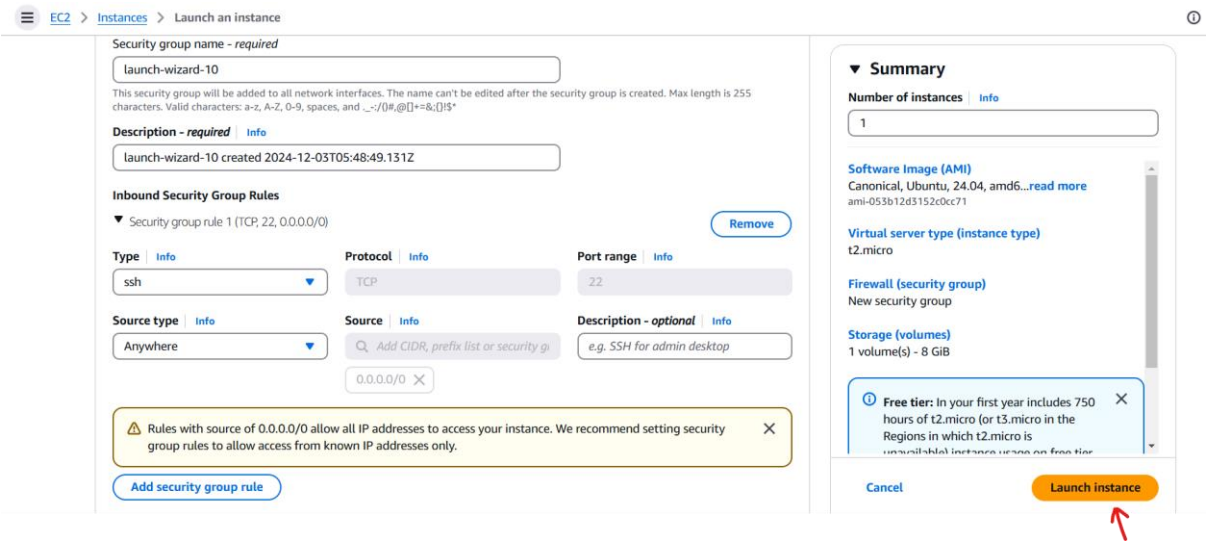


4. **Creating Bastion or jump host to access these instances since it do not have public ips. Bastion host is act as a mediator between private subnet and public subnet.**



➢ **Selecting our VPC with public subnet.**





- **Copying pem key from local machine to bastion host to access application instance from bastion host.**

```
C:\Users\swapnil\Downloads>scp -i git_key.pem git_key.pem ubuntu@65.2.57.9:/home/ubuntu/
```

- **Successfully logged in to one of the application instance from bastion host.**

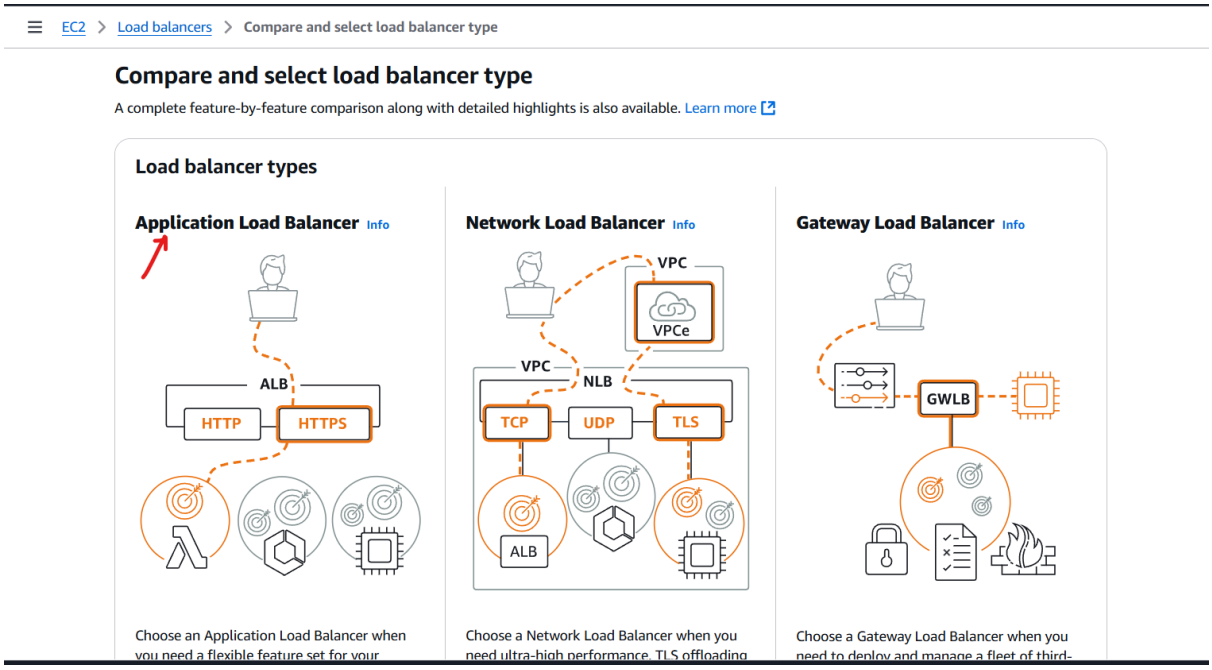
```
ubuntu@ip-10-0-18-164:~$ ssh -i git_key.pem ubuntu@10.0.143.109
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro
```

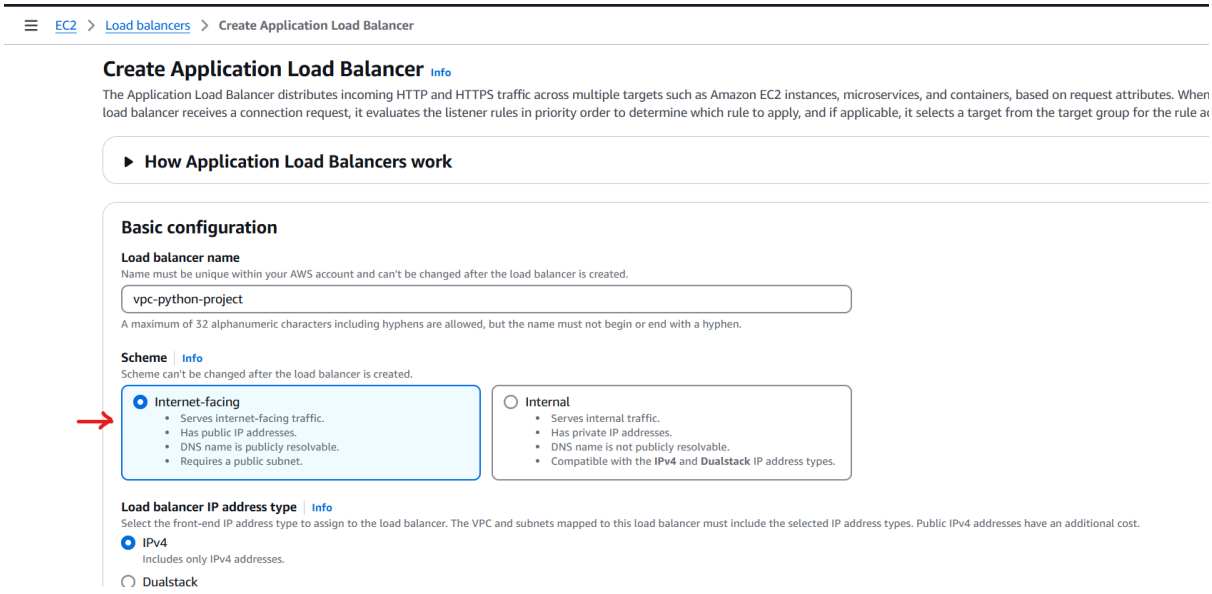
- **Running python server with sample index.html file on application instance.**

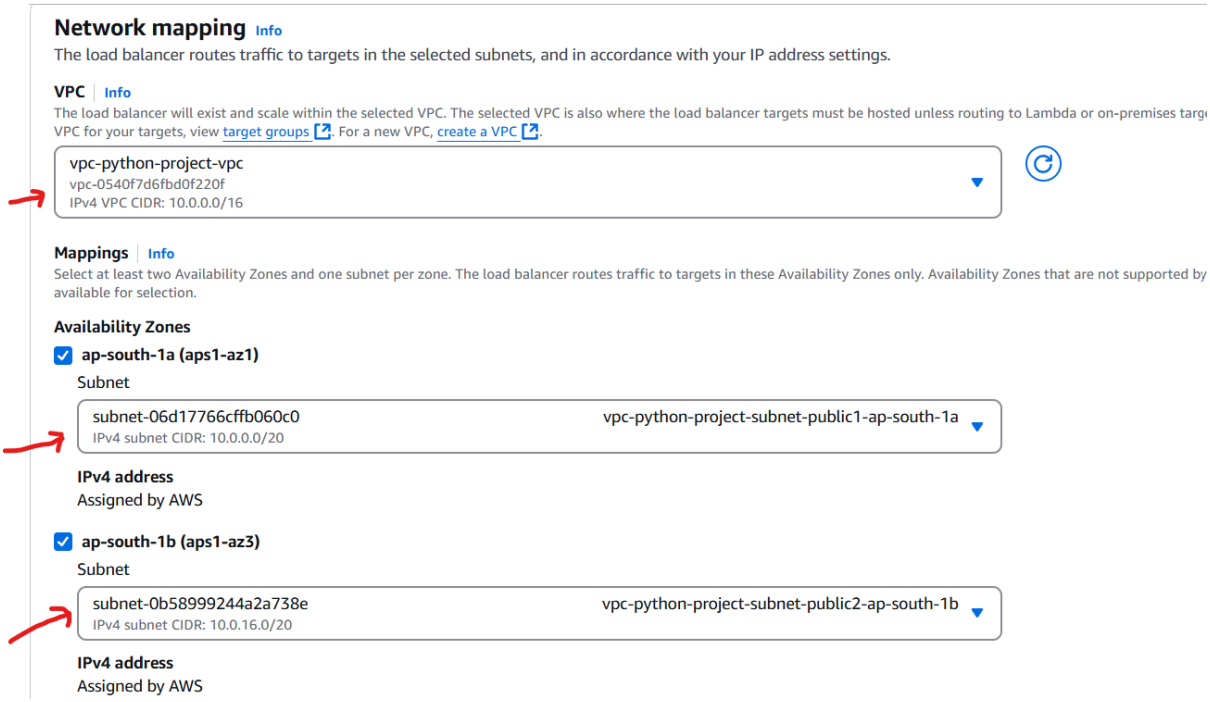
```
ubuntu@ip-10-0-143-109:~$ vim index.html
ubuntu@ip-10-0-143-109:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

5. Creating Application load balancer to incoming requests to your application

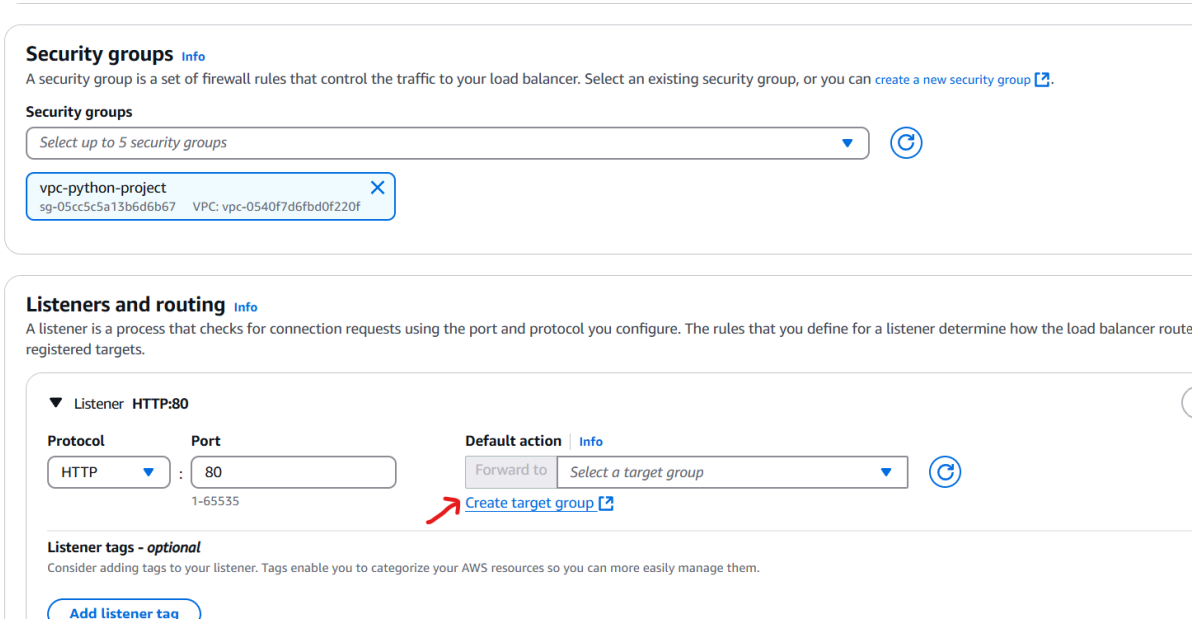


➢ **LB should be internet facing.**

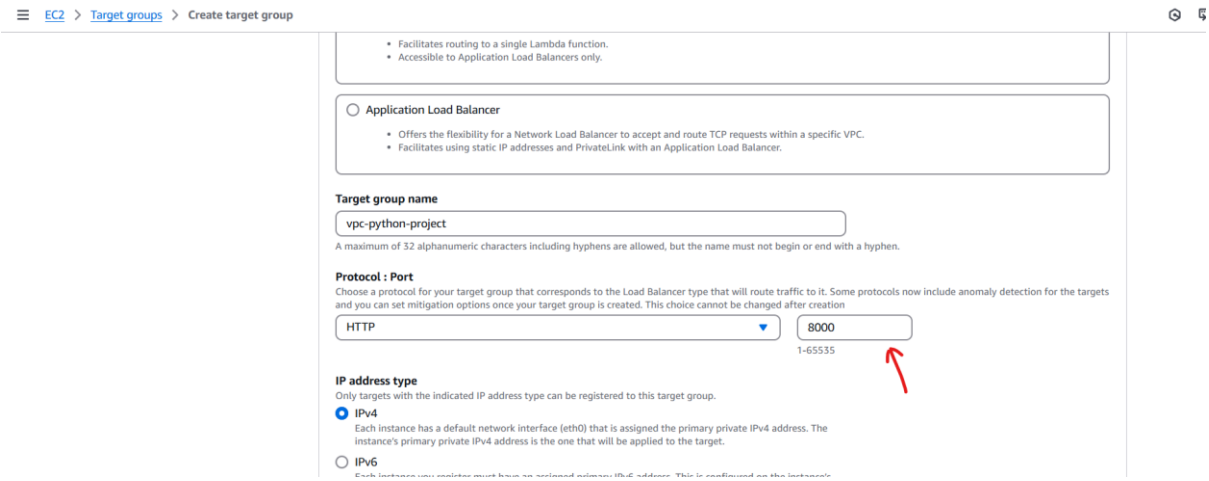


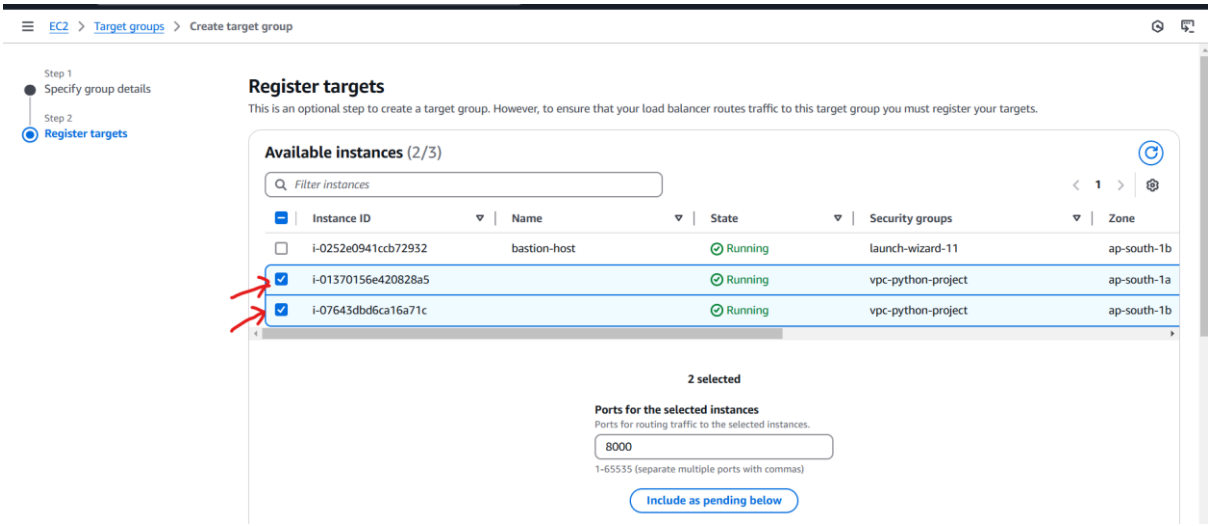


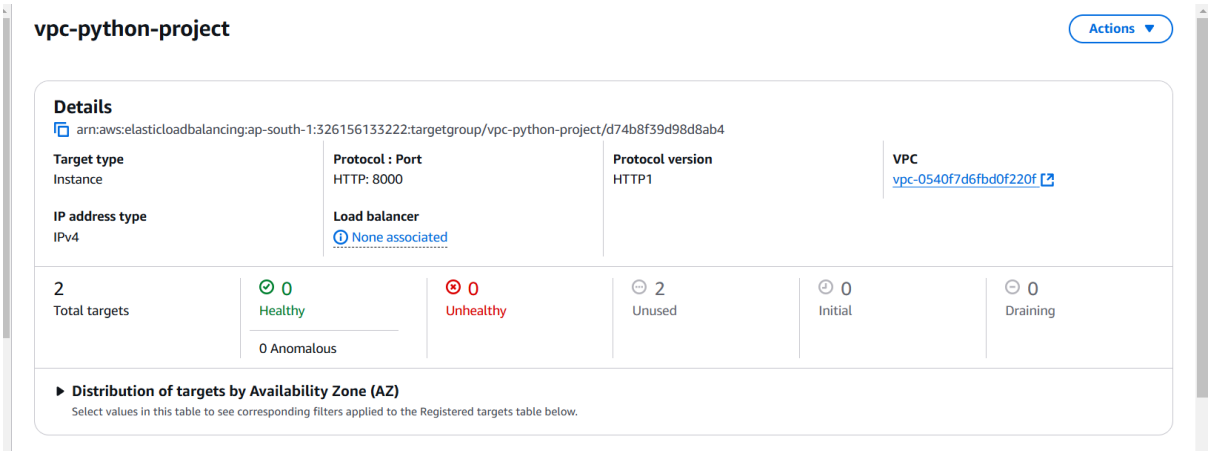
➢ Creating target group



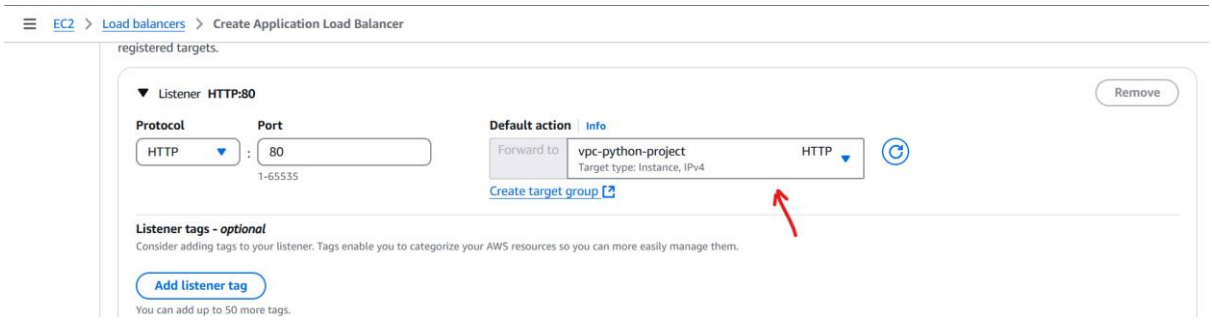
- **Selecting port 8000 since our application is running on same port.**

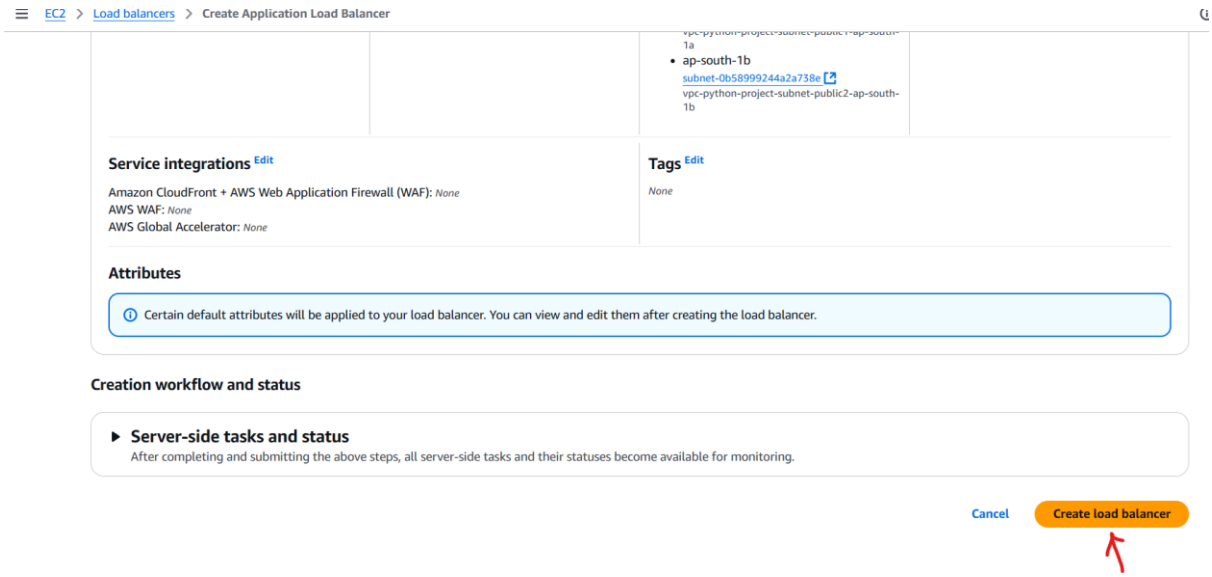




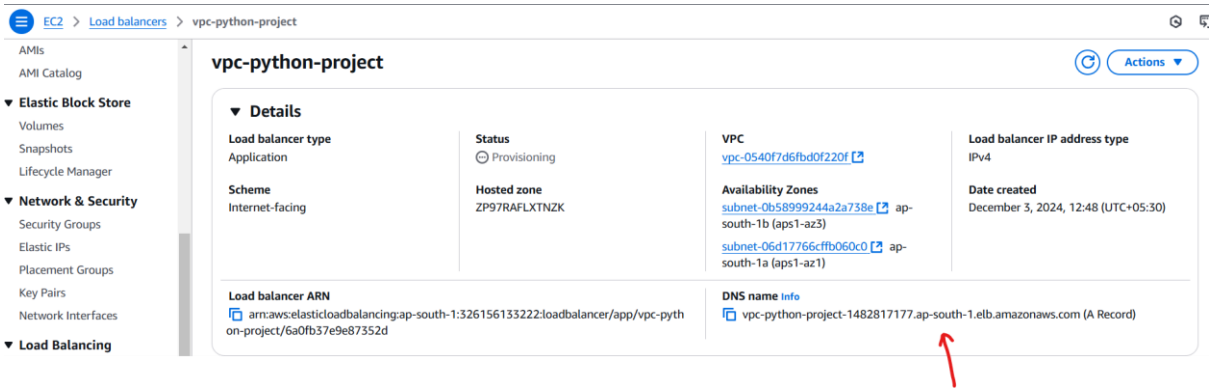


- **Back to the ALB configuration.**





- **Try to access your page by using DNS name present in LB.**



Not secure vpc-python-project-1482817177.ap-south-1.elb.amazonaws.com

This is a first VPC python project

Click me