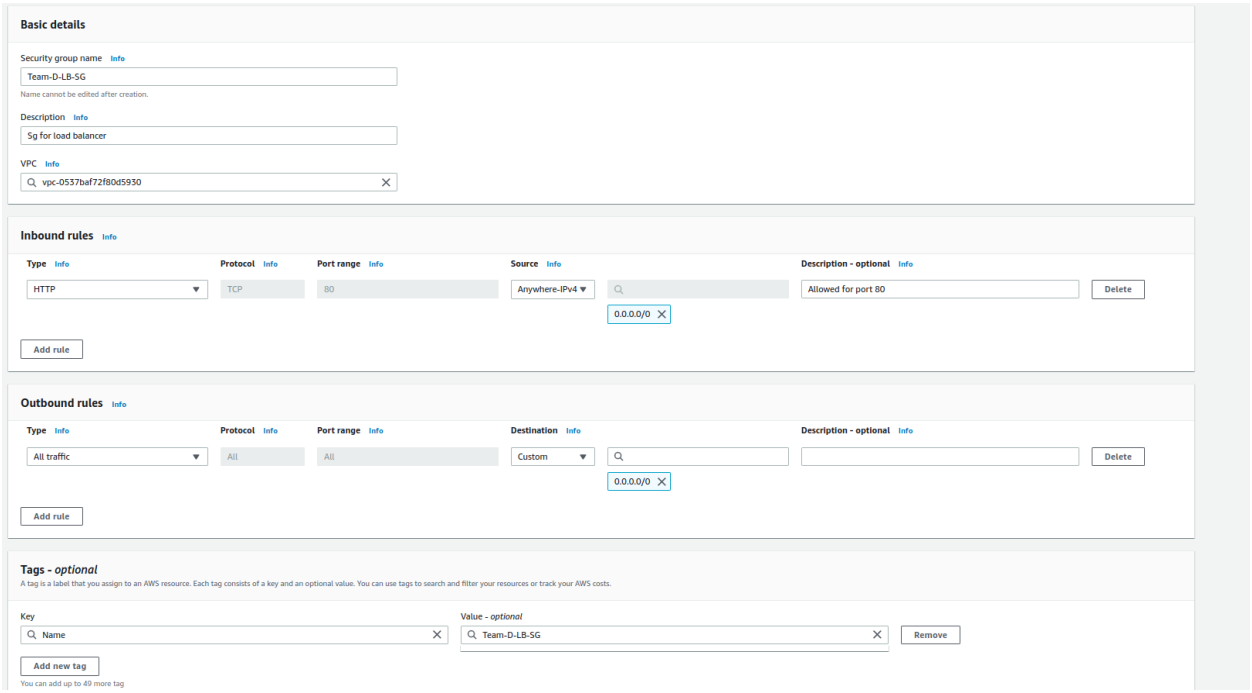
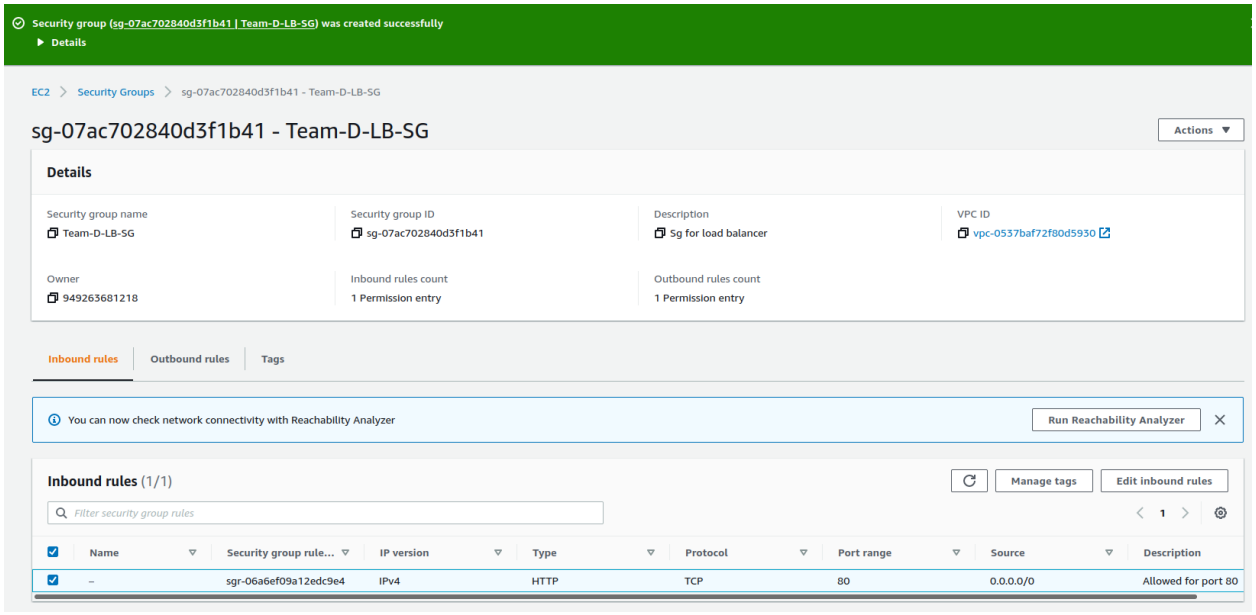
Create Application Load balancer

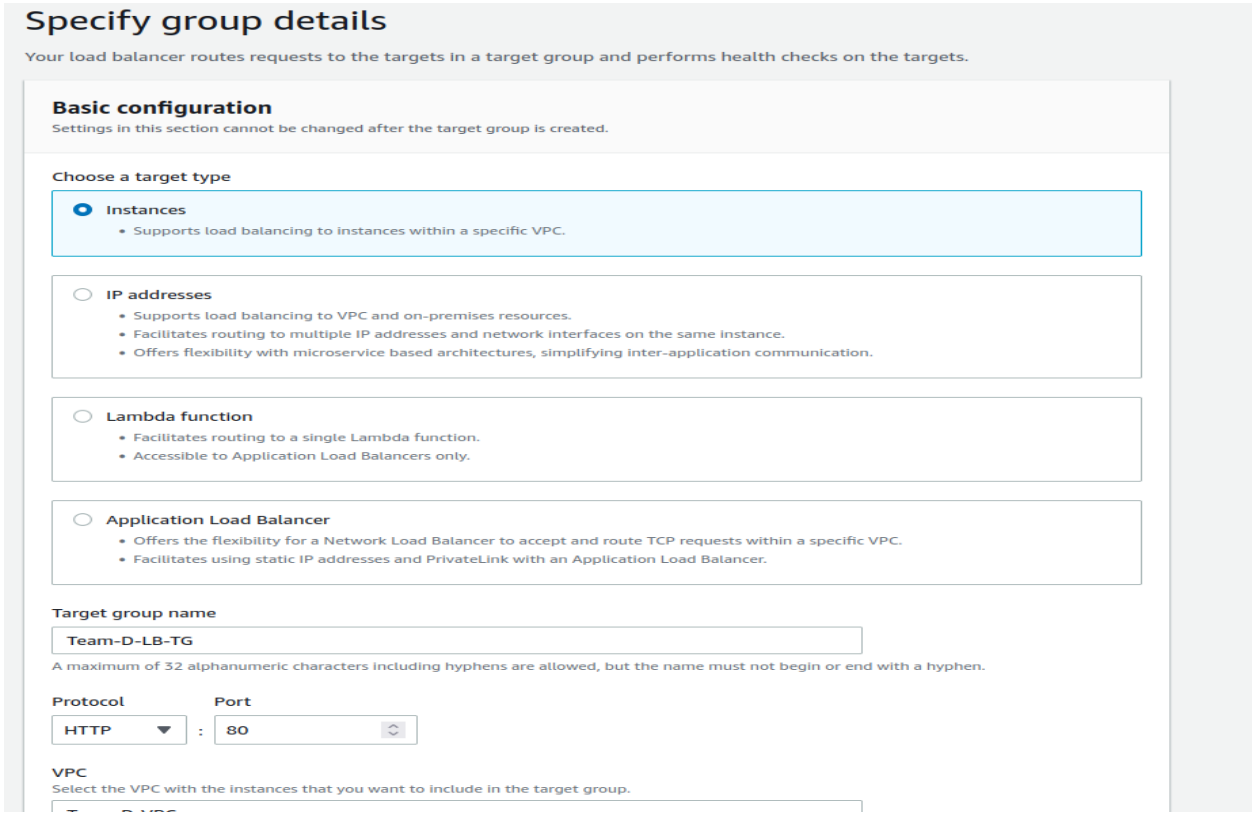
- Start a simple server in private EC2 8880
- Target ALB to serve that server
- ALB should be accessible through port 80 listener
- Health Check
- Register healthy on 3 success
- Register unhealthy on 5 success
- Timeout 5 Seconds
- Interval 45 Seconds
- Access the server via ALB publicly using ALB's DNS name.

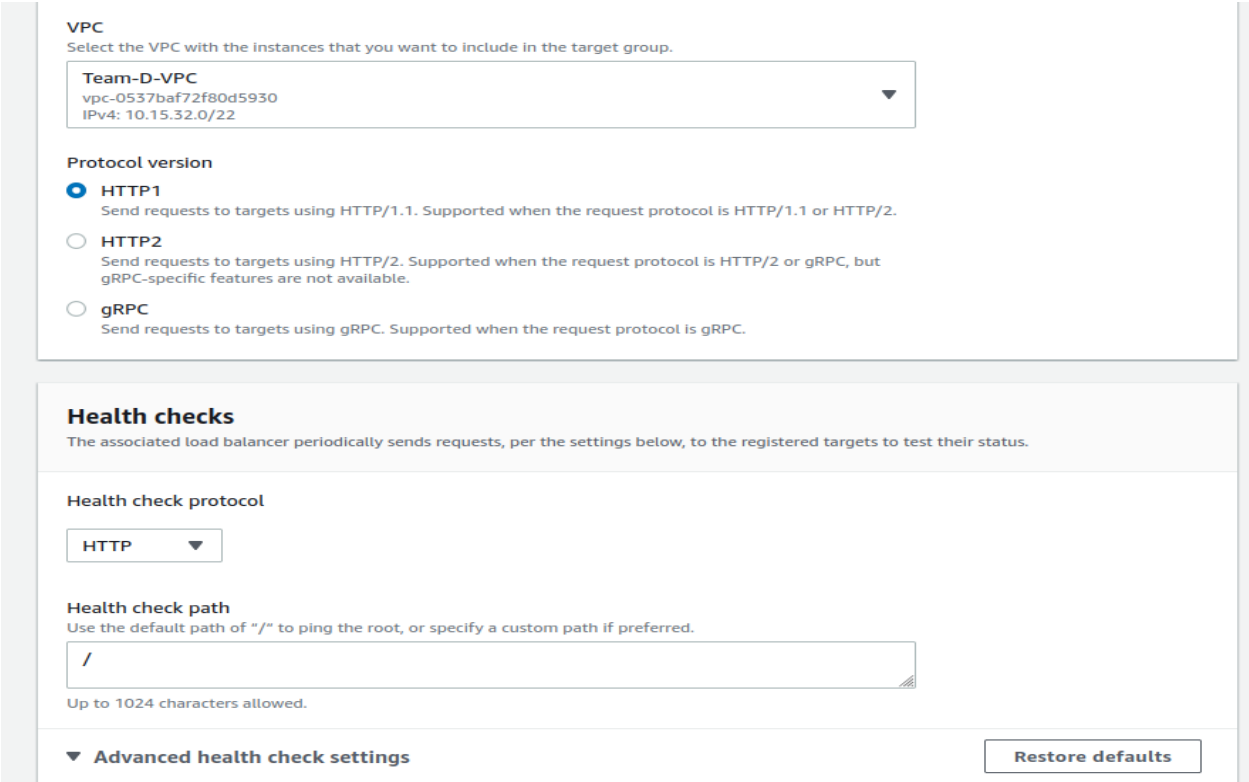
Security Group Create for our Load balancer first as;

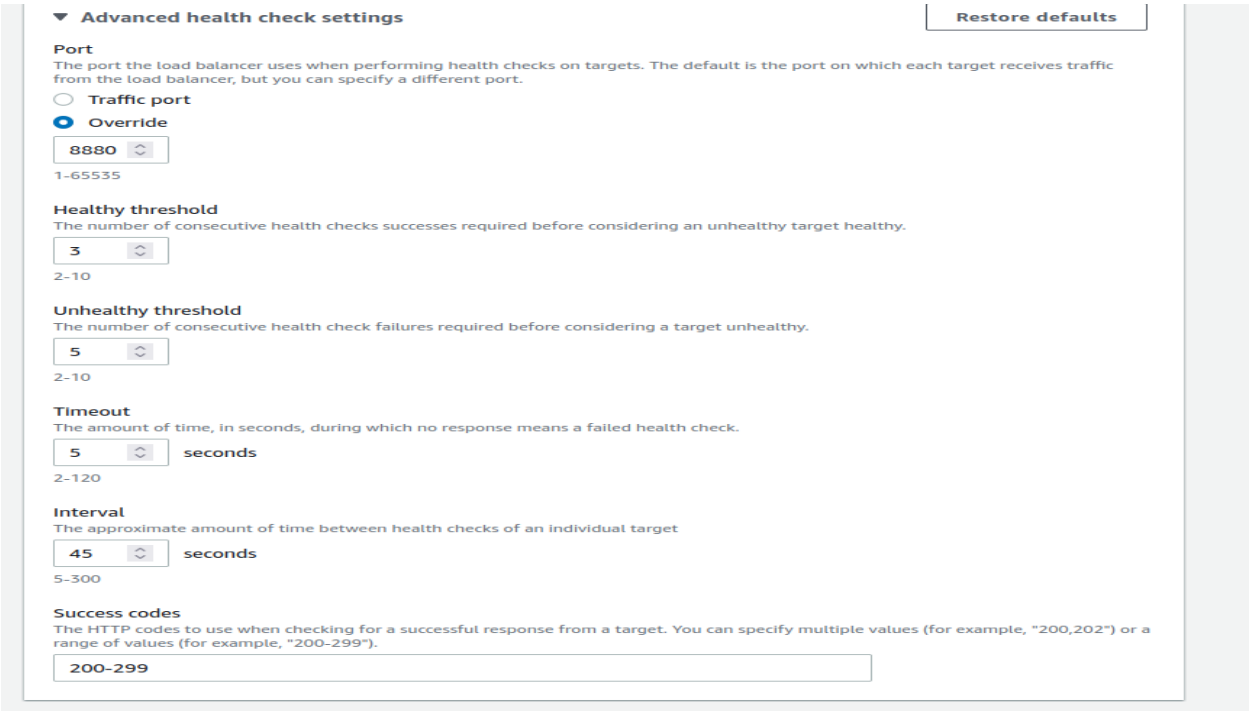




Creating Target Group :

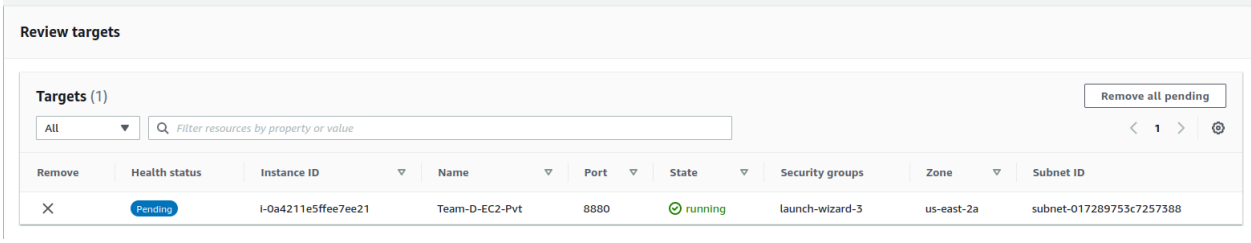


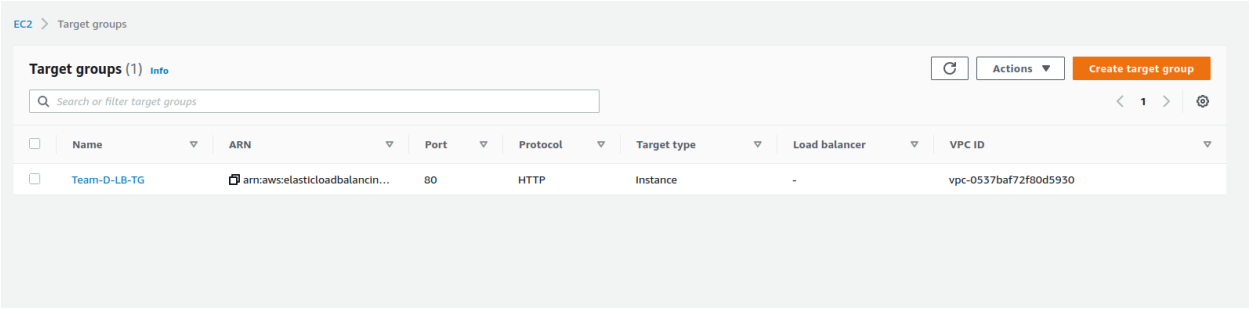




Note that the port is in **Override** and has port **8880** and other health check settings as per question.

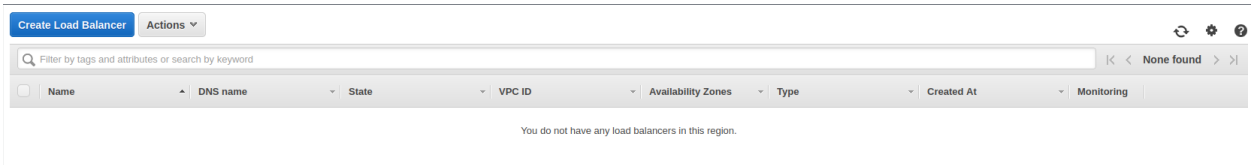
Register the target:





Now we create load balancer as:

Load Balancer Home



Load Balancer Creation

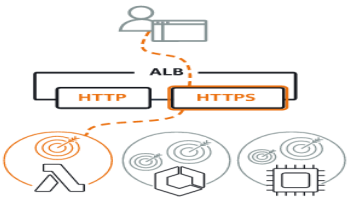
Select the application load balancer from below:

Select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. Learn more

Load balancer types

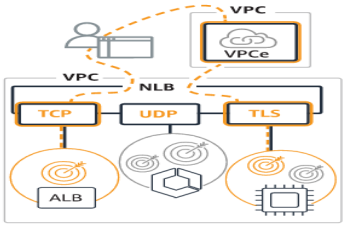
Application Load Balancer Info



Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create

Network Load Balancer Info



Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

Gateway Load Balancer Info

Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▸ **How Application Load Balancers work**

Basic configuration

Load balancer name

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

Team-D-LB

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info

Scheme cannot be changed after the load balancer is created.

- (•) Internet-facing
  An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more
- ( ) Internal
  An internal load balancer routes requests from clients to targets using private IP addresses.

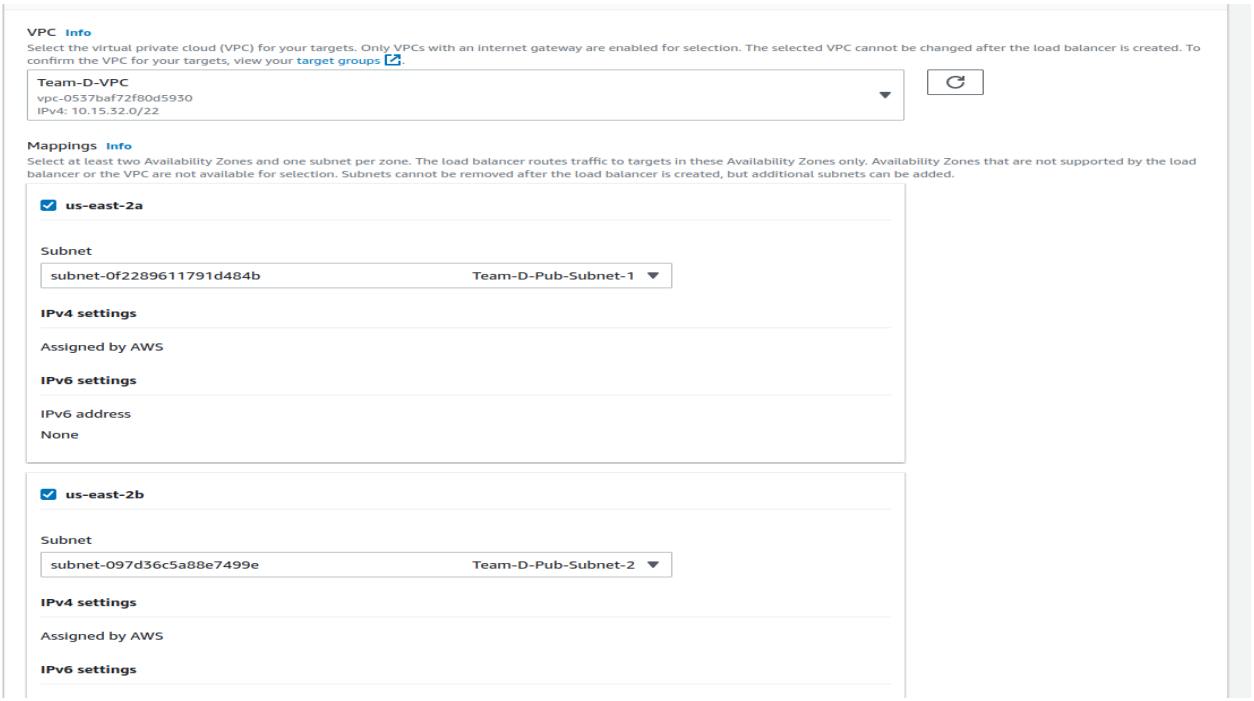
IP address type Info

Select the type of IP addresses that your subnets use.

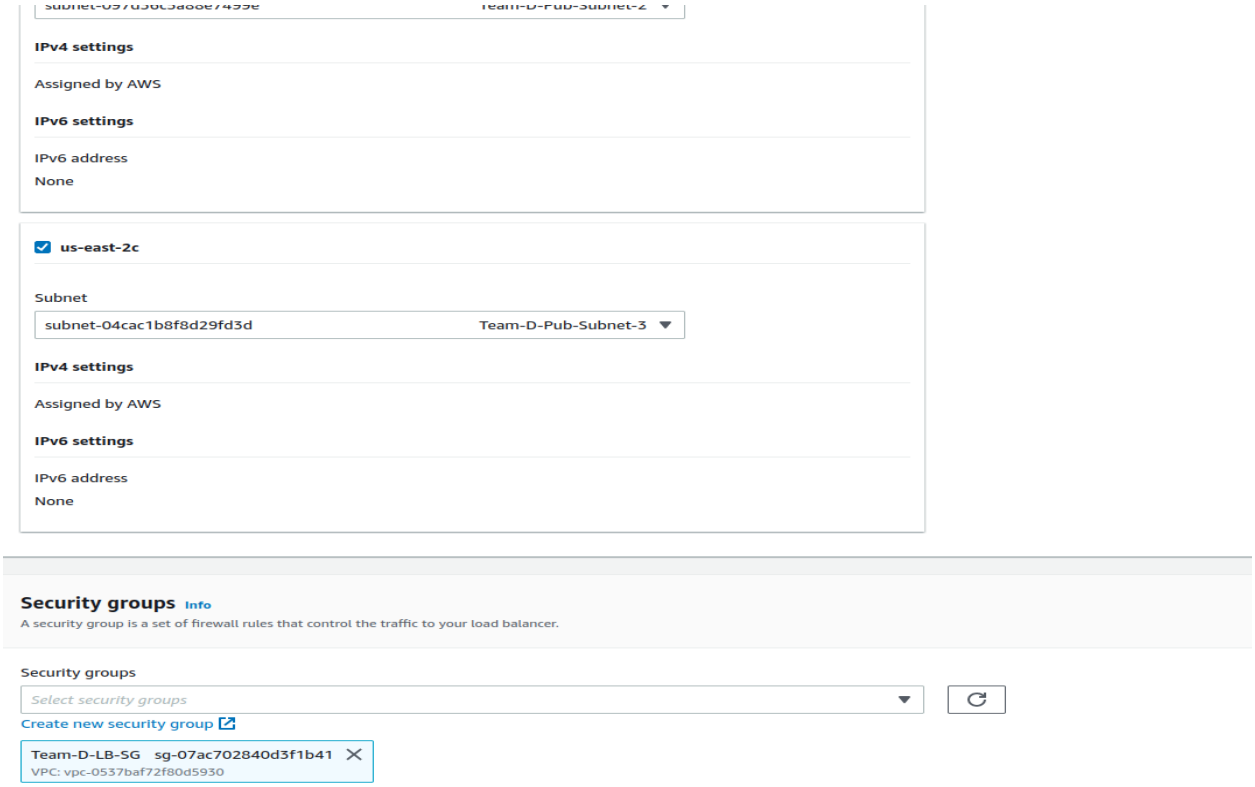
- (•) IPv4
  Recommended for internal load balancers.
- ( ) Dualstack
  Includes IPv4 and IPv6 addresses.

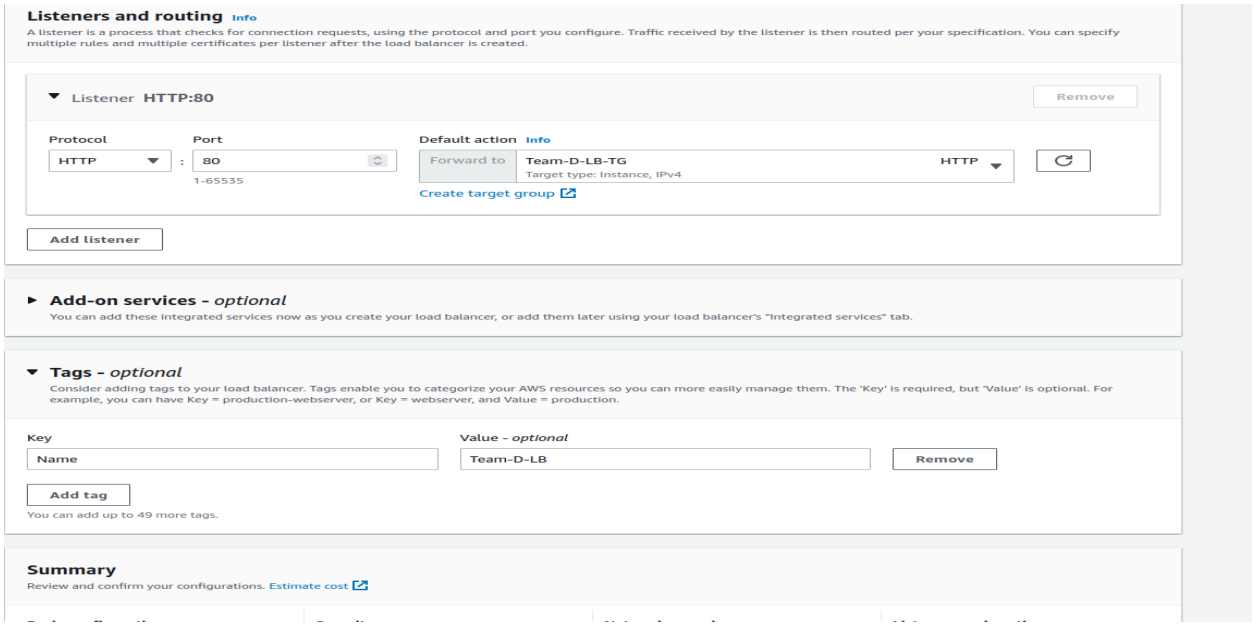
Network mapping Info

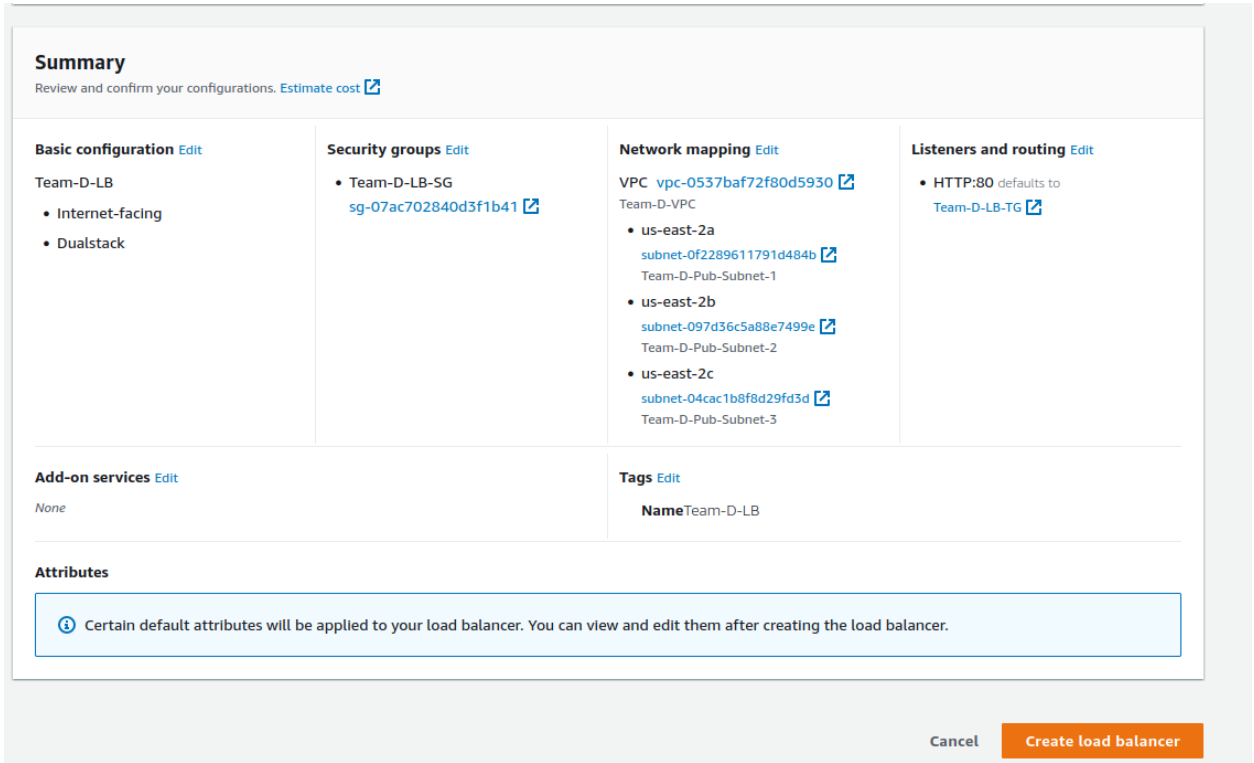
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.



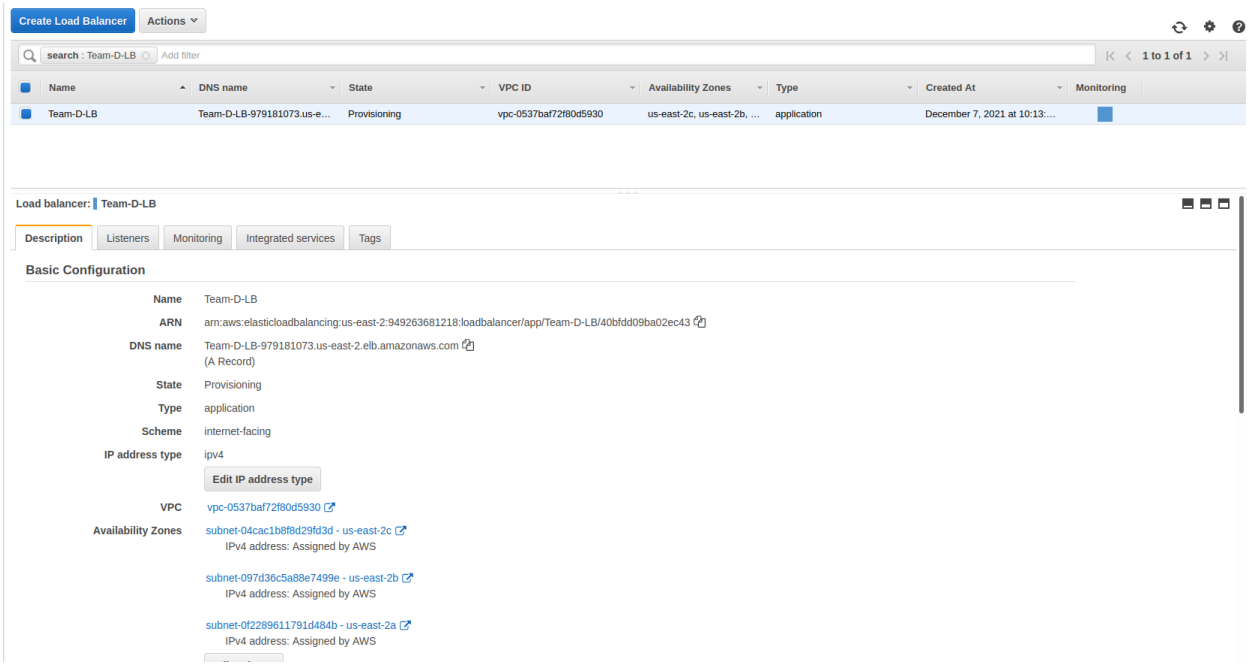
Select our created security group and target group:



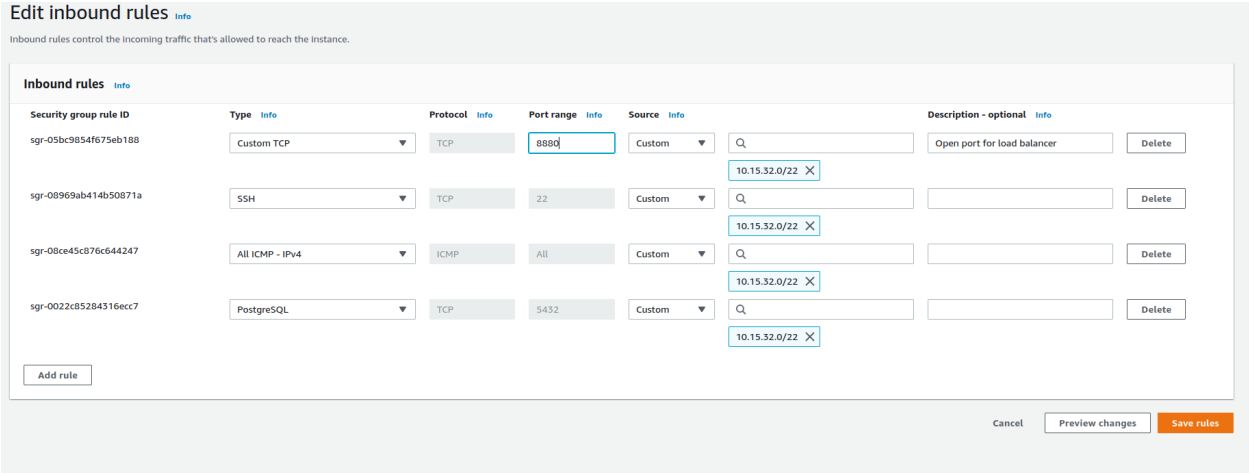

Listeners and routing Info
A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.
Listener HTTP:80
Remove
Protocol
HTTP
Port
80
1-65535
Default action Info
Forward to
Team-D-LB-TG
Target type: Instance, IPv4
HTTP
Create target group
Add listener
Add-on services - optional
You can add these integrated services now as you create your load balancer, or add them later using your load balancer's "Integrated services" tab.
Tags - optional
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.
Key
Name
Value - optional
Team-D-LB
Remove
Add tag
You can add up to 49 more tags.
Summary
Review and confirm your configurations. Estimate cost


Summary
Review and confirm your configurations. Estimate cost
Basic configuration Edit
Team-D-LB
Internet-facing
Dualstack
Security groups Edit
Team-D-LB-SG
sg-07ac702840d3f1b41
Network mapping Edit
VPC vpc-0537baf72f80d5930
Team-D-VPC
us-east-2a
subnet-0f2289611791d484b
Team-D-Pub-Subnet-1
us-east-2b
subnet-097d36c5a88e7499e
Team-D-Pub-Subnet-2
us-east-2c
subnet-04cac1b8f8d29fd3d
Team-D-Pub-Subnet-3
Listeners and routing Edit
HTTP:80 defaults to
Team-D-LB-TG
Add-on services Edit
None
Tags Edit
NameTeam-D-LB
Attributes
Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.
Cancel
Create load balancer

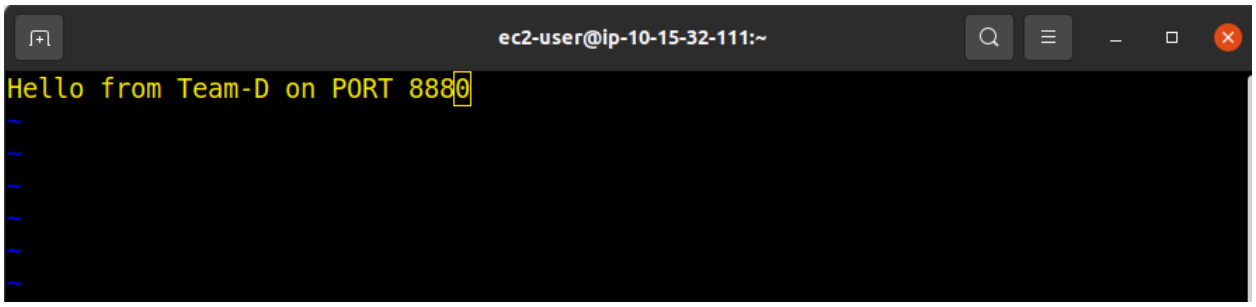
Created Load Balancer



Open Port 8880



We have our index page as below:



Run the python script to see it in the web from our private ec2-instance as :

*here, **10.15.32.111** is our private ip of the private ec2 instance.*

We can finally see the result through the ALB DNS Name publicly as,