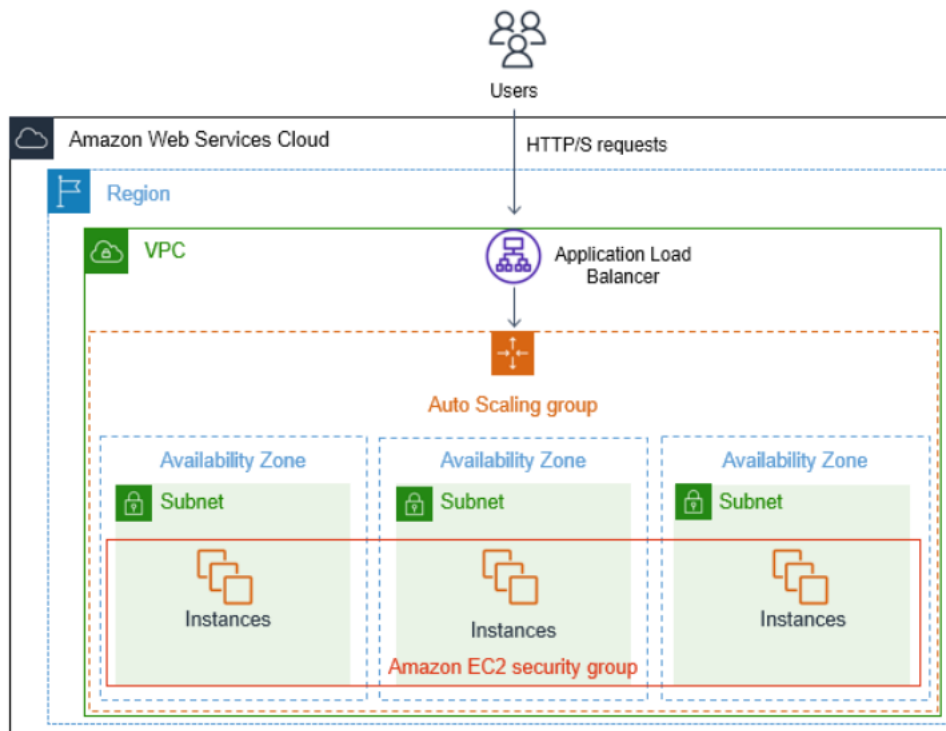




PRODUCTION ENVIRONMENT SETUP WITH AWS CLOUD

Architecture:



Prerequisites:

-VPC

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

- Auto Scaling Group

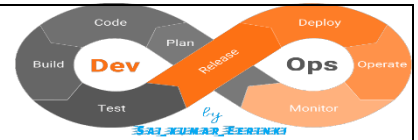
(First we configure the create launch template & attach the template to Auto Scaling group)

-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.

Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

☐ VPC only ☒ VPC and more

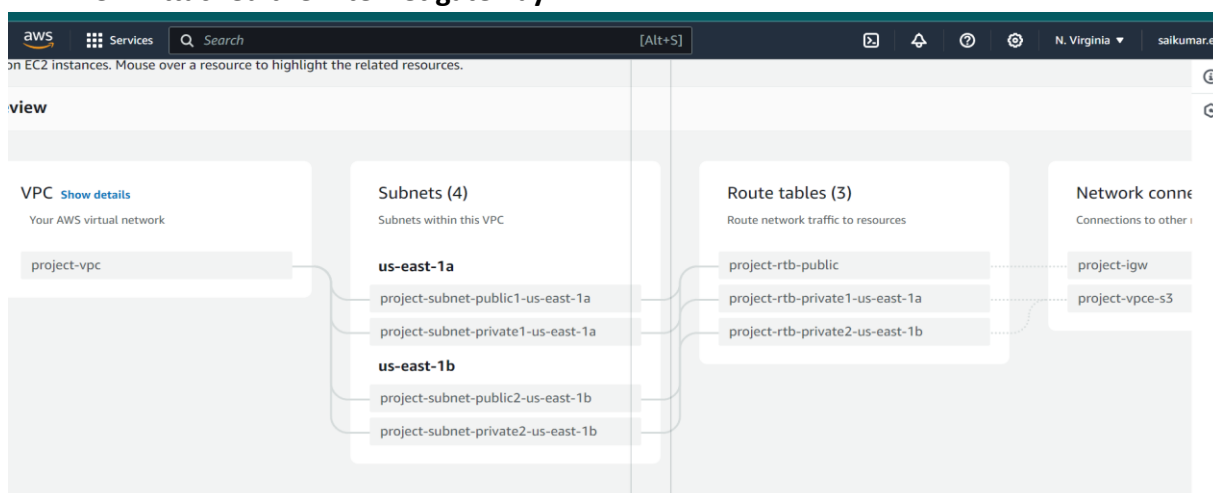
Name tag auto-generation [Info](#)
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

☒ Auto-generate

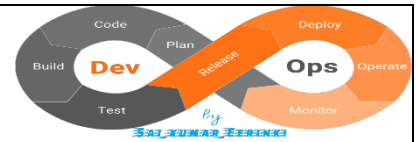
project

IPv4 CIDR block [Info](#)
Determine the starting IP and the size of your VPC using CIDR notation.

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.

aws Services Search [Alt+S]

EC2 > Launch templates > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

AWS_Prod_SETUP

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

A prod webserver for MyApp

Max 255 chars

[Auto Scaling guidance](#) [Info](#)

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

aws Services Search [Alt+S]

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

Recents Quick Start

Don't include in launch template Amazon Linux macOS Ubuntu Windows

aws ubuntu Microsoft

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

▼ Instance type Info | Get advice Advanced

Instance type

Don't include in launch template

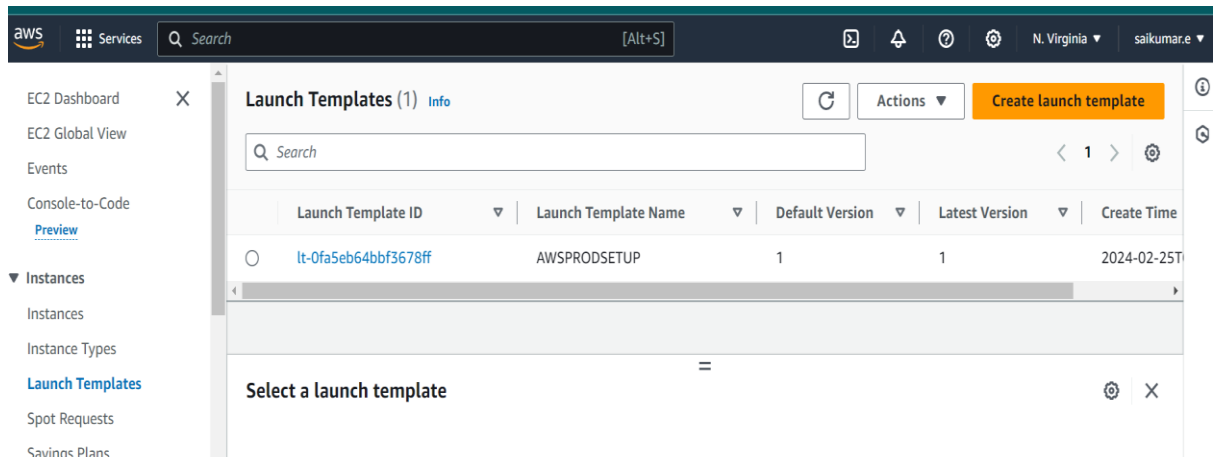
All generations

Compare instance types

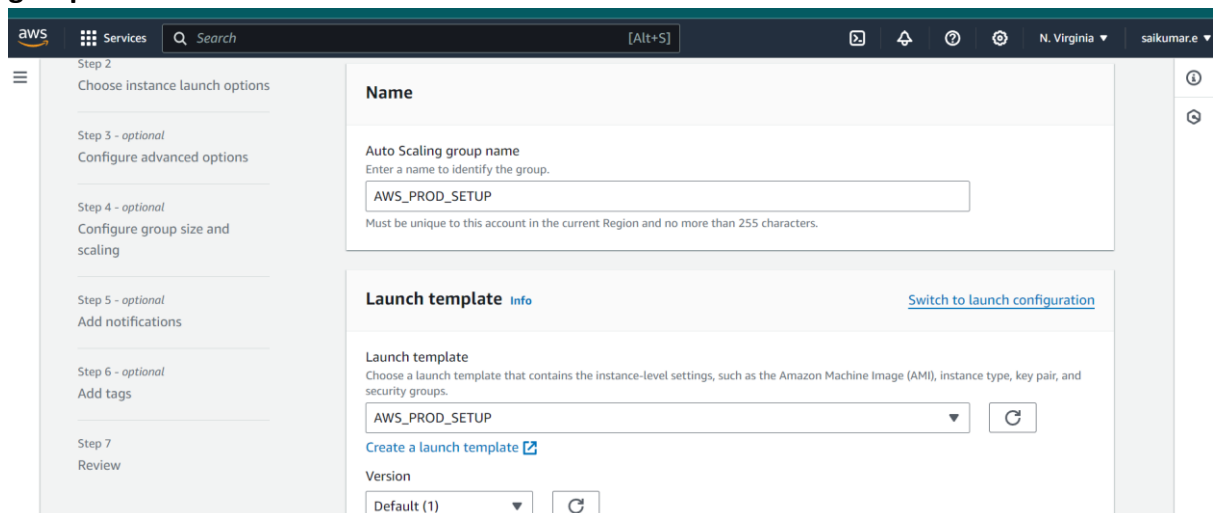
5. Based upon you requirement you can create the KeyPairs (If you created the created the server in public subnet then download ppk file or If you create the private subnet download the pem file to connect the server with ssh connectivity with help of jump host.



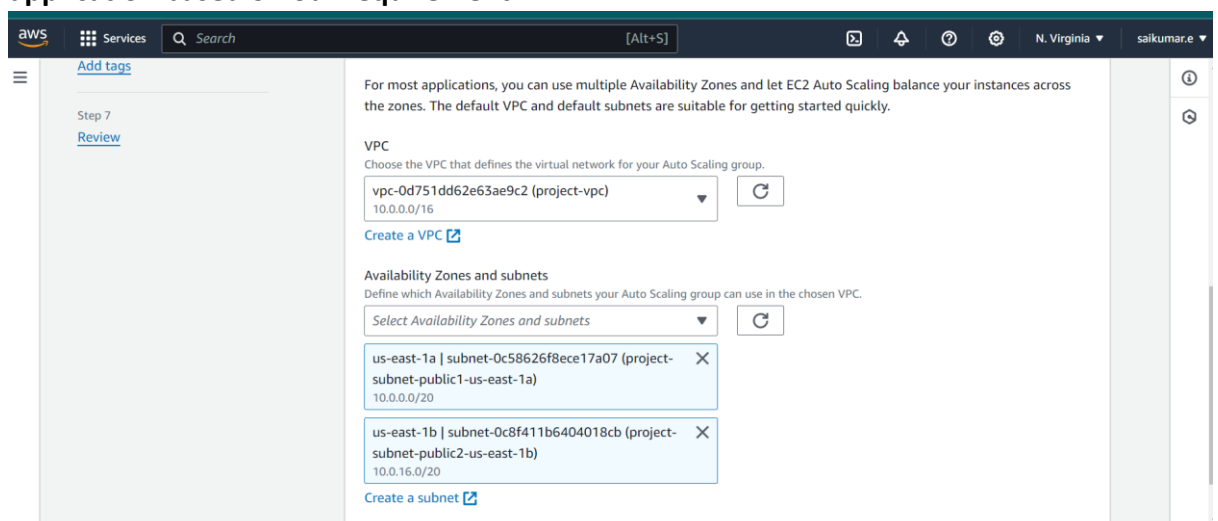
6. Launch template created successfully. (you can in the below Image).



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.

	Name	Launch template/configuration	Instances	Status	Desired capacity	Min
<input type="checkbox"/>	AWS_PROD_SETUP	AWS_PROD_SETUP Version Default	0	Updating capacity...	2	1



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.

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

☒ **Instances**

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

☐ **IP addresses**

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

Only targets with the indicated IP address type can be registered to this target group.

☒ **IPv4**

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

☐ **IPv6**

Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

project-vpc
vpc-0d751dd62e63ae9c2
IPv4: 10.0.0.0/16

Protocol version

☒ **HTTP1**

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

☐ **HTTP2**

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

☐ **gRPC**

Send requests to targets using gRPC. Supported when the request protocol is gRPC.

4. Target group created successfully.

Successfully created the target group: AWSPRODSETUP. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab.

AWSPRODSETUP

Introducing Automatic Target Weights (ATW) to increase application availability

Automatic Target Weights is achieved by turning on anomaly mitigation, which provides responsive, dynamic distribution of traffic to targets based on anomaly detection results. All HTTP/HTTPS target groups now include anomaly detection by default. [Learn more](#)

Details

arn:aws:elasticloadbalancing:us-east-1:264748077800:targetgroup/AWSPRODSETUP/9d6a12cb267e2959

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-0d751dd62e63ae9c2



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

aws Services Search [Alt+S] N. Virginia saikumar.e

EC2 > Load balancers > Compare and select load balancer type

Compare and 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

Network Load Balancer

Gateway Load Balancer

aws Services Search [Alt+S] N. Virginia saikumar.e

Create Application Load Balancer

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 can't be changed after the load balancer is created.

AWSPRODSETUP

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

Scheme [Info](#)
Scheme can't 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](#)

6. Add the security group , Listeners & routing.

aws Services Search [Alt+S] N. Virginia saikumar.e

Security groups

Select up to 5 security groups

default
sg-027cd15690ead72c4 VPC: vpc-0d751dd62e63ae9c2

Listeners and routing

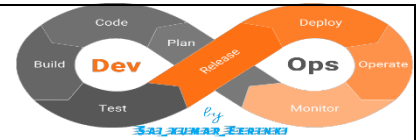
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80 Remove

Protocol HTTP Port 80 Default action Forward to AWSPRODSETUP Target type: Instance, IPv4 HTTP

1-65535

[Create target group](#)



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

VPC info
Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

project-vpc
vpc-0d751dd62e63ae9c2
IPv4: 10.0.0.0/16

Mappings info
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

☒ **us-east-1a (use1-az4)**
Subnet
subnet-0c58626f8e17a07 project-subnet-public1-us-east-1a
IPv4 address
Assigned by AWS

☒ **us-east-1b (use1-az6)**
Subnet
subnet-0c8f411b6404018cb project-subnet-public2-us-east-1b
IPv4 address

8. Successfully created the Application load balancer.

Successfully created load balancer: AWSPRODSETUP
It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

AWSPRODSETUP

Details

Load balancer type Application	Status Provisioning	VPC vpc-0d751dd62e63ae9c2	IP address type IPv4
Scheme Internet-facing	Hosted zone Z35XDOTRQ7X7K	Availability Zones subnet-0c58626f8e17a07 us-east-1a (use1-az4) subnet-0c8f411b6404018cb us-east-1b (use1-az6)	Date created February 25, 2024, 12:02 (UTC+05:30)

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.

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:264748077800:loadbalancer/app/AWSPROD/6f1bd7ff239a880e

AWSPROD-81033113.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (1) info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Filter listeners

Protocol:Port	Default action	Rules	ARN	Security
HTTP:80	Default			

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.