

AWS Project-VPC with public-private subnet in Production

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 Availability Zones.

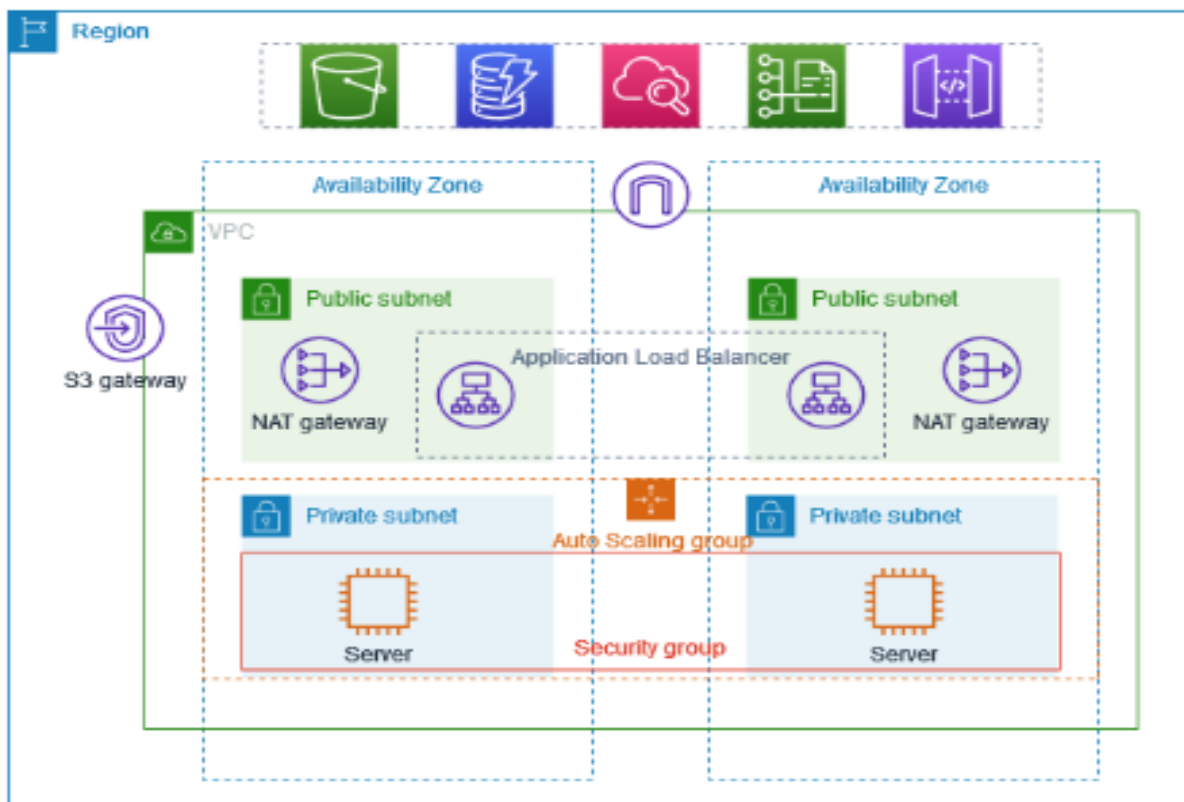
Contents

- [Overview](#)
- [1. Create the VPC](#)
- [2. Deploy your application](#)
- [3. Test your configuration](#)
- [4. Clean up](#)

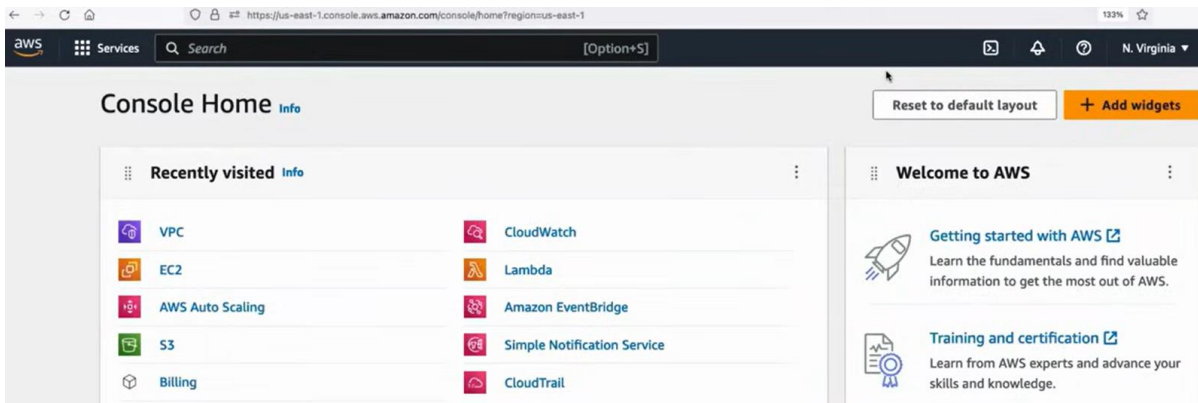
Overview:

The VPC has public subnets and private subnets in two Availability zones. Each public subnet contains a NAT gateway and a load balancer node. The servers run in the private subnets, are launched and terminated by using an Auto Scaling group and receive traffic from the load balancer. The servers can connect to the internet by using NAT gateway.

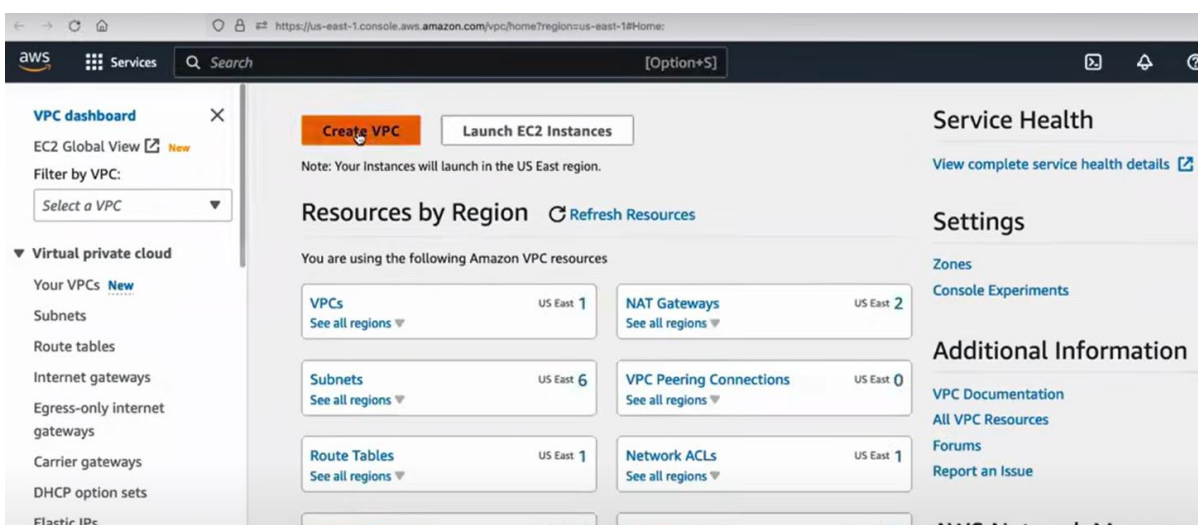
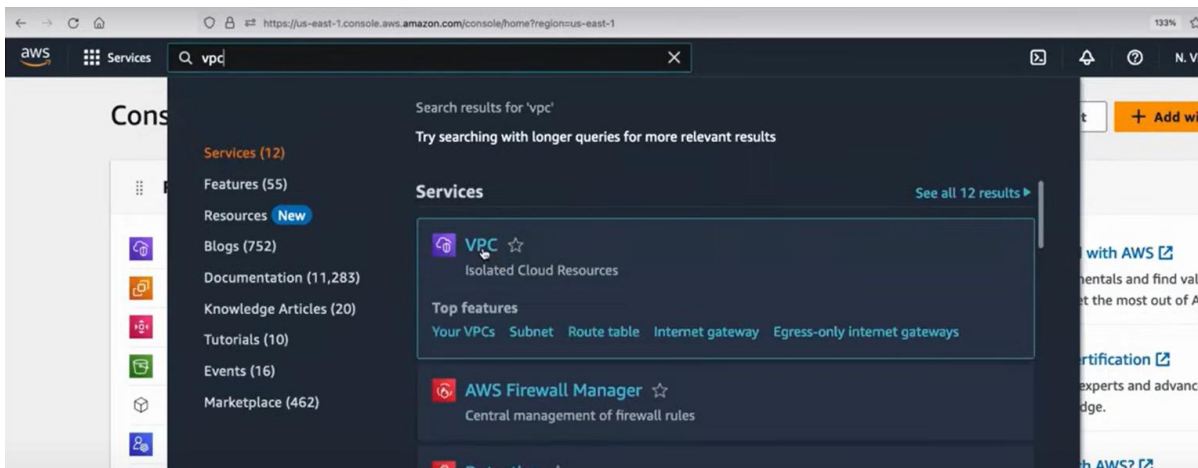
The following diagram provides an overview of the resources included in this example. The VPC has public subnets and private subnets in two Availability Zones. Each public subnet contains a NAT gateway and a load balancer node. The servers run in the private subnets, are launched and terminated by using an Auto Scaling group, and receive traffic from the load balancer. The servers can connect to the internet by using the NAT gateway. The servers can connect to Amazon S3 by using a gateway VPC endpoint.



1. Go to Aws console and search for VPC



2. Select the VPC and create VPC.



3. Select VPC and more in VPC settings and 1 Per AZ in NAT gateway, leave remaining as default and create VPC.

Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

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.

10.0.0.0/16 65,536 IPs

IPv6 CIDR block [Info](#)

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

VPC endpoints [Info](#)
Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None S3 Gateway

DNS options [Info](#)

☒ Enable DNS hostnames
☒ Enable DNS resolution

► **Additional tags**

Preview

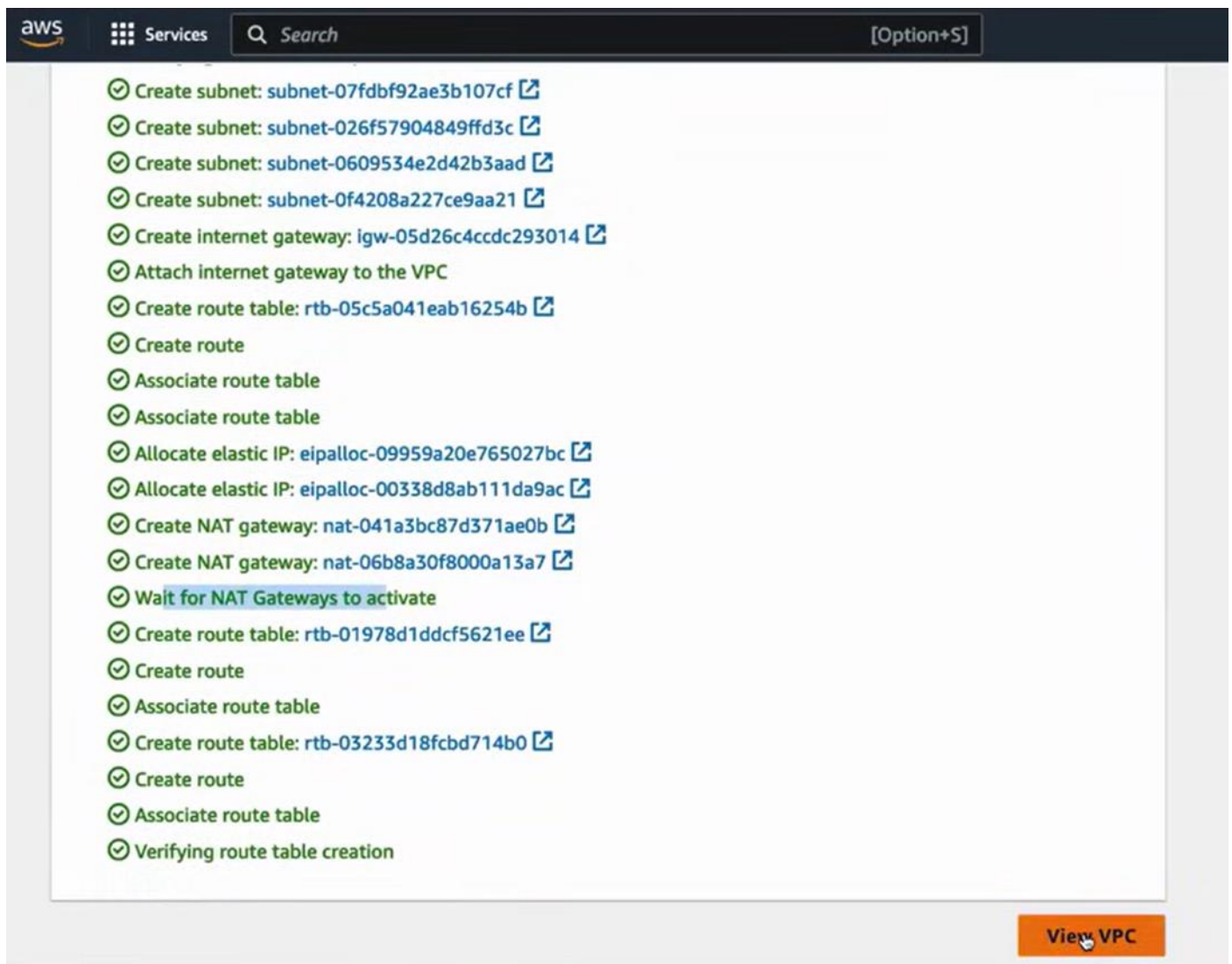
VPC [Show details](#)
Your AWS virtual network

project-vpc

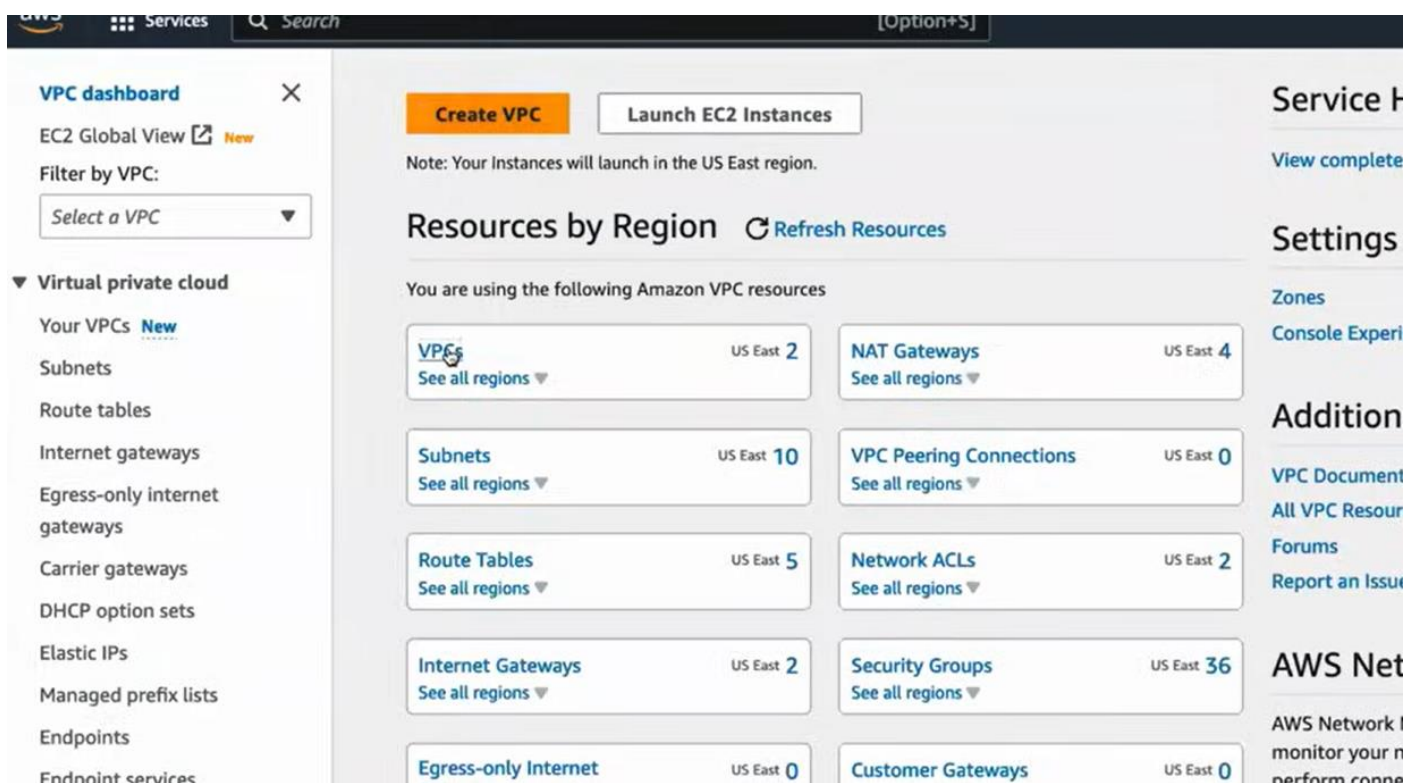
Subnets (4)
Subnets within this VPC

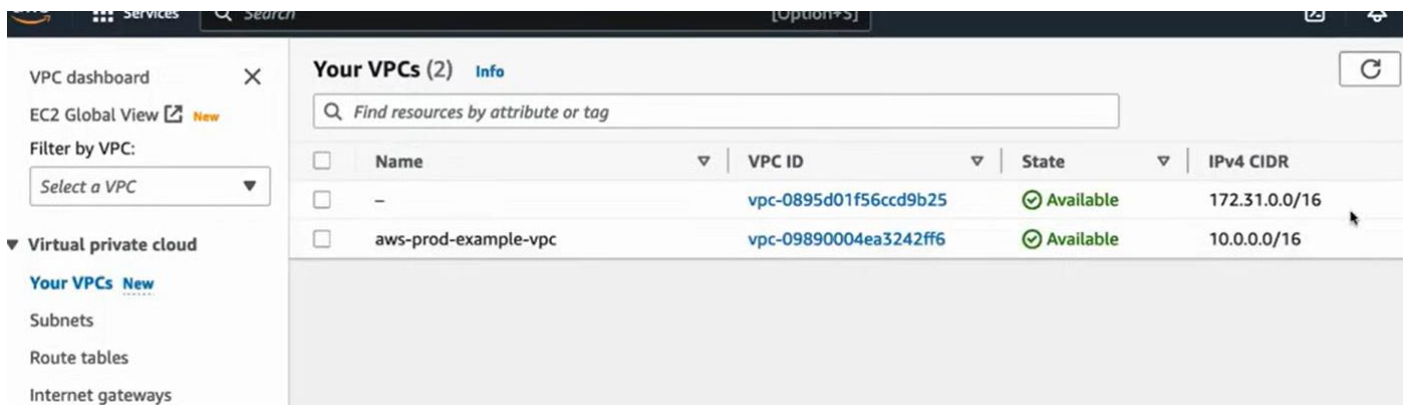
us-east-1a
project-subnet-public1-us-east-1a
project-subnet-private1-us-east-1a
us-east-1b
project-subnet-public2-us-east-1b
project-subnet-private2-us-east-1b

Cancel **Create VPC**

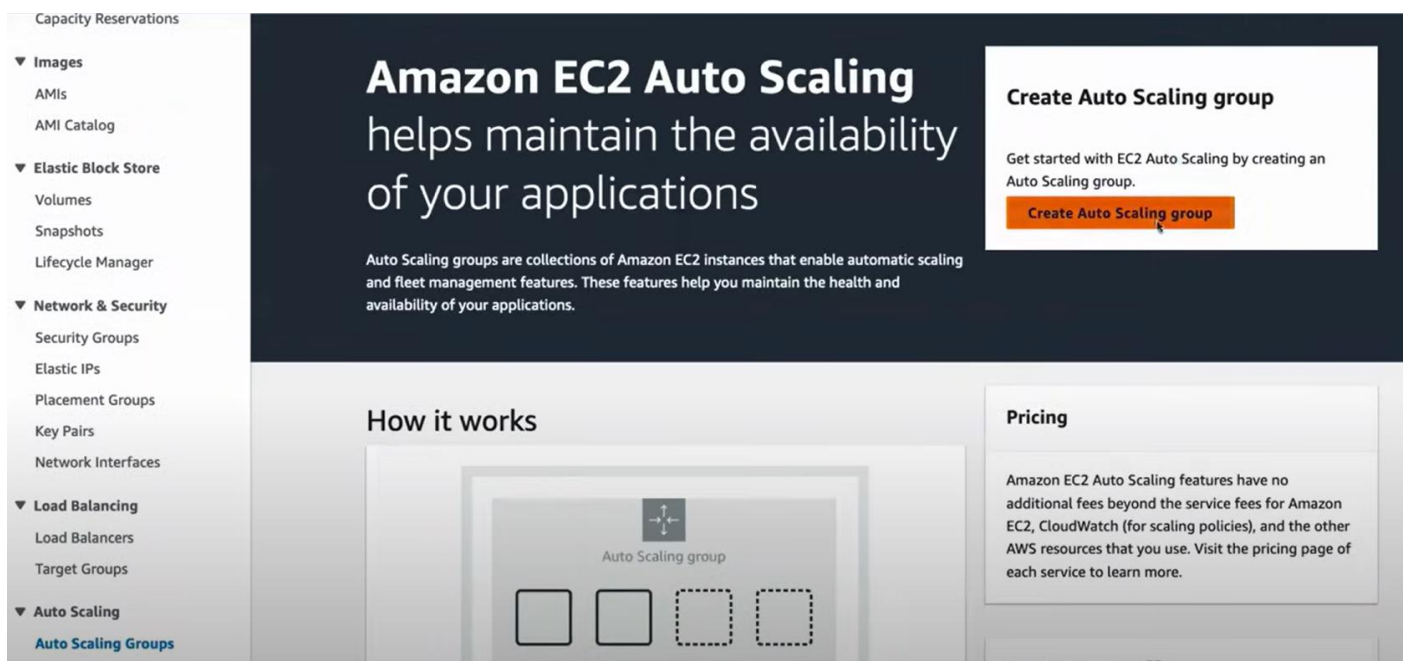
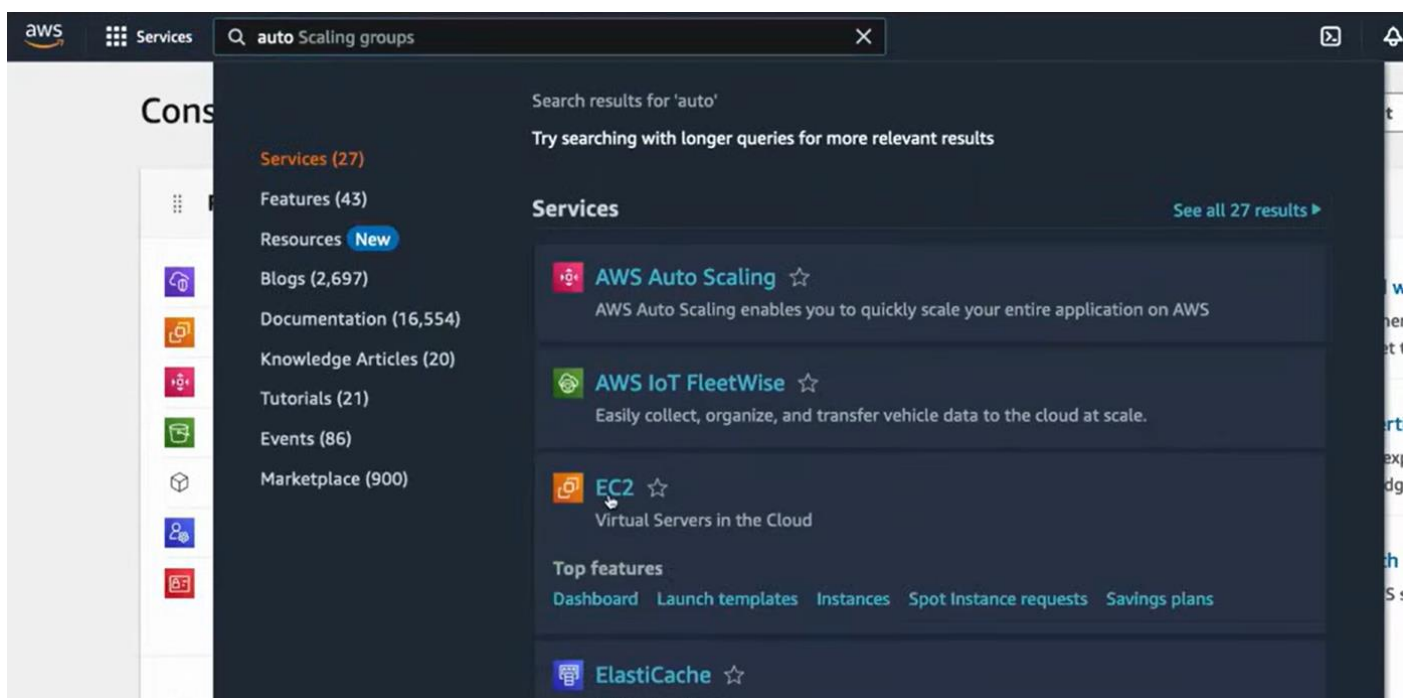


4. Click on view VPC you can see your created VPC





5. Now create a Auto Scaling as shown



EC2 / Auto Scaling groups / Create Auto Scaling group

Step 1

Choose launch template or configuration

Step 2

Choose instance launch options

Step 3 - optional

Configure advanced options

Step 4 - optional

Configure group size and scaling policies

Step 5 - optional

Add notifications

Step 6 - optional

Add tags

Step 7

Review

Choose launch template or configuration Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name

Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

Switch to launch configuration

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Select a launch template

Create a launch template

Cancel

Next

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

MyTemplate

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

Select this if you intend to use this template with EC2 Auto Scaling

☒ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Template tags

Source template

Summary

Software Image (AMI)

-

Virtual server type (instance type)

-

Firewall (security group)

-

Storage (volumes)

-

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of

Cancel

Create launch template

Launch template name and description

Launch template name - required

aws-prod-example

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

Template version description

proof of concept for app deploy in aws private subnet

Max 255 chars

Auto Scaling guidance Info

Select this if you intend to use this template with EC2 Auto Scaling

☒ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Template tags

Source template

Summary

Software Image (AMI)

-

Virtual server type (instance type)

-

Firewall (security group)

-

Storage (volumes)

-

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS

▼ 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

RecentsQuick Start

Recently launched

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230516

ami-053b0d53c279acc90

2023-05-16T03:38:03.000Z architecture: 64-bit (x86) Virtualization: hvm ENA enabled: true Root device type: ebs boot mode: legacy-bios

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-05-16

ArchitectureAMI ID

▼ Summary

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)

ami-053b0d53c279acc90

Virtual server type (instance type)

-

Firewall (security group)

-

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage on t2.micro, t3.micro, and t3a.micro instances.

Cancel

Create launch template

6. Add your key pair to this, if not create a new one.

▼ 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

Don't include in launch template

Create new key pair

▼ Network settings [Info](#)

Subnet Info

7. In Network settings select create security group, give VPC that is created previously and create template as shown below.

▼ Network settings Info

Subnet Info

Don't include in launch template ▼

When you specify a subnet, a network interface is automatically added to your template.

Create new subnet

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Select existing security group

☒ Create security group

Security group name - required

MyWebServerGroup

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and .-:/[]+=&:()!\$*

Description - required Info

Allows SSH access to developers

VPC - required Info

vpc-0895d01f56ccd9b25
172.31.0.0/16

(default) ▼

Inbound Security Group Rules

No security group rules are currently included in this template. Add a new rule to include it in the launch template.

▼ Summary

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more
ami-053b0d53c279acc90

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS

Cancel

Create launch template

8. Click on Inbound Security Groups Rules and configure as follow.

allow ssh access

VPC - required Info

vpc-09890004ea3242ff6 (aws-prod-example-vpc)
10.0.0.0/16

▼

Inbound Security Group Rules

No security group rules are currently included in this template. Add a new rule to include it in the launch template.

Add security group rule

► Advanced network configuration

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) Remove

Type Info	Protocol Info	Port range Info
ssh ▼	TCP	22
Source type Info	Source Info	Description - optional Info
Anywhere ▼	<input type="text" value="Add CIDR, prefix list or security"/> <input type="text" value="0.0.0.0/0 X"/>	e.g. SSH for admin desktop

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X

Add security group rule

► Advanced network configuration

Type Info	Protocol Info	Port range Info
Custom TCP ▼	TCP	8000
Source type Info	Source Info	Description - optional Info
Anywhere ▼	<input type="text" value="Add CIDR, prefix list or security"/> <input type="text" value="0.0.0.0/0 X"/>	e.g. SSH for admin desktop

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X

Add security group rule

9. Click Create template

Creating template

Please wait while we create your template.
Do not close your browser while this is loading.

○ Creating security group rules

► Details

10. Now Name Auto Scaling Group and add this created template to Auto Scaling Group. Click on next.

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template or configuration

Step 2
Choose instance launch options

Step 3 - optional
Configure advanced options

Step 4 - optional
Configure group size and scaling policies

Step 5 - optional
Add notifications

Step 6 - optional
Add tags

Step 7
Review

Choose launch template or configuration [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name
Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

Launch template [Info](#) [Switch to launch configuration](#)

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Select a launch template

[Create a launch template](#)

[Cancel](#) [Next](#)

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

aws-prod-example

[Create a launch template](#)

Version
Default (1)

[Create a launch template version](#)

Description proof of concept for app deploy in aws private subnet	Launch template aws-prod-example lt-0eebdb51bee1684b2	Instance type t2.micro
AMI ID ami-053b0d53c279acc90	Security groups -	Request Spot Instances No
Key pair name aws_login	Security group IDs sg-0eb19d86be3c7500e	

Additional details

Storage (volumes) -	Date created Mon Jul 03 2023 19:07:35 GMT+0530 (India Standard Time)
-------------------------------	---

11. Select the created VPC that is created and also the Availability Zones and subnets. Click on next and create Auto Scaling Group.

Choose instance launch options [Info](#)

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Network [Info](#)

For **most applications**, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-09890004ea3242ff6 (aws-prod-example-vpc) ▼
10.0.0.0/16



[Create a VPC](#)

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets ▼



[Create a subnet](#)

Configure advanced options - *optional* [Info](#)

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over 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.

dynamically scale the number of instances in the group.

ich options

options

e and

Group size - optional [Info](#)

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity

1

Minimum capacity

1

Maximum capacity

1

Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

☐ Target tracking scaling policy

☒ None

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](#)

Cancel [Previous](#) [Create Auto Scaling group](#)

12. Check and verify the Two Instances are created in desired region.
13. Now create a new EC2 instance and launch as shown below.

following the simple steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [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

Recents

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Li

Browse more AMIs

▼ Summary

Number of instances [Info](#)

Software Image (AMI)

Amazon Linux 2023 AMI 2023.1.2...[read more](#)
ami-06b09bfcae1453cb

Virtual server type (instance type)

-

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Cancel

Launch instance

[Review commands](#)

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group
 ☐ Select existing security group

We'll create a new security group called 'launch-wizard-34' with the following rules:

☒ Allow SSH traffic from
Helps you connect to your instance

Anywhere
0.0.0.0/0
Anywhere

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
 ☐ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

▼ Network Settings [Info](#)

Edit

14. Make sure that the Bastion VPC is created in same VPC .

Auto-assign public IP [Info](#)

Firewall (security groups) [Info](#)

Launching instance

Please wait while we launch your instance.

Do not close your browser while this is loading.

○ Creating security group rules

23%

► Details

15. You can see instances are created.

<input type="checkbox"/>	Name	Instance ID
<input type="checkbox"/>	-	i-0e619
<input type="checkbox"/>	-	i-0bd77
<input type="checkbox"/>	bastion-host	i-01b11

16. Use your Terminal to login and give the SSH key to it using Scp command to the Bastion instance using Ip address of it.

17. Use the below command with your bastion Ip.

18. Do ls command you will see your Aws pem file. **ssh -i your pem file ubuntu@your ip address**

19. Now we want to install our application in any one of our two instances, login to any one instance using private ip address using this command **ssh -i your pem file ubuntu@your ip address**

20. Now you are login in a instance using private ip. Install/add your application here within this instance.

21. I am using some basic html page and python3 with server.

```
vim index.html
python3 -m http.server 8000
```

22. Now go to EC2 Load balancers.

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

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

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 performance.

23. Select Application Load Balancer and 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 Elastic Load Balancing works

Basic configuration

Load balancer name

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

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](#)

☐ Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

Network mapping [Info](#)

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

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](#).

-

↺

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, subnets, or the VPC are not available for selection.

☐ us-east-1a (use1-az4)

☐ us-east-1b (use1-az6)

24. Provide the VPC that is Created now and pick up the both Mappings change to public subnet.

25. Now select security groups already created or you can create custom one also.

Listeners and routing [Info](#)

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

Port

Default action [Info](#)

HTTP ▼

:

80

1-65535

Forward to

Select a target group ▼

↻

Create target group [↗](#)

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

You can add up to 50 more tags.

26. Create target group using port 8000

EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

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.

☐ Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

aws-prod-example

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

Protocol

Port

HTTP ▼

:

80

1-65535

VPC

Select the VPC with the instances that you want to include in the target group.

aws-prod-example-vpc

27. Now select the both instances and create and add this to the load balancer.

Listeners and routing [Info](#)

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	Port	Default action	Info
HTTP ▼	80 1-65535	Forward to	aws-prod-example Target type: Instance, IPv4 Create target group
			HTTP ▼ ↻

Listener tags - *optional*

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)

You can add up to 50 more tags.

28. Now create the load balancer.

The screenshot shows the AWS Management Console interface. On the left, there's a sidebar with navigation links: 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Limits', and 'Instances'. The 'Instances' section is expanded. The main content area is titled 'EC2 > Load balancers'. It shows 'Load balancers (1/1)' and a description: 'Elastic Load Balancing scales your load balancer capacity a'. Below this is a search bar with the placeholder text 'Find resources by attribute or tag'. A table lists the load balancers with columns for 'Name' and 'DNS name'. The first entry is 'aws-prod-example' with a DNS name 'aws-p'.

29. Click on the load balancer and copy DNS name in details section and search in your browser, you can see your application.

The screenshot shows a web browser window. The address bar contains the URL 'aws-prod-example-564969239.us-east-1.elb.amazonaws.com'. The page title is 'My First AWS PROJECT to demonstrate apps in private subnet'. The page content is mostly blank, suggesting the application is not yet fully loaded or is in a state where only the title is visible.

30. The above one is my basic application which I used. The project is done.

NOTE: For more information please refer to this Amazon official doc.

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-example-private-subnets-nat.html>