

3-Tier Architecture: Deployment of Java-Springboot-App on MySQL Database With Load Balancer and Auto Scalling Goup

Project Overview:

Project Name: Healthcare

Language use: Java

Database: MySQL

Architecture: 3 Tier

Domain: Health care

Modules: Patient, Doctor, Admin

Patient - Registration, Login, Online Appointment, Cancel Appointment, manage profile

Doctor: Login, view appointment, manage profile

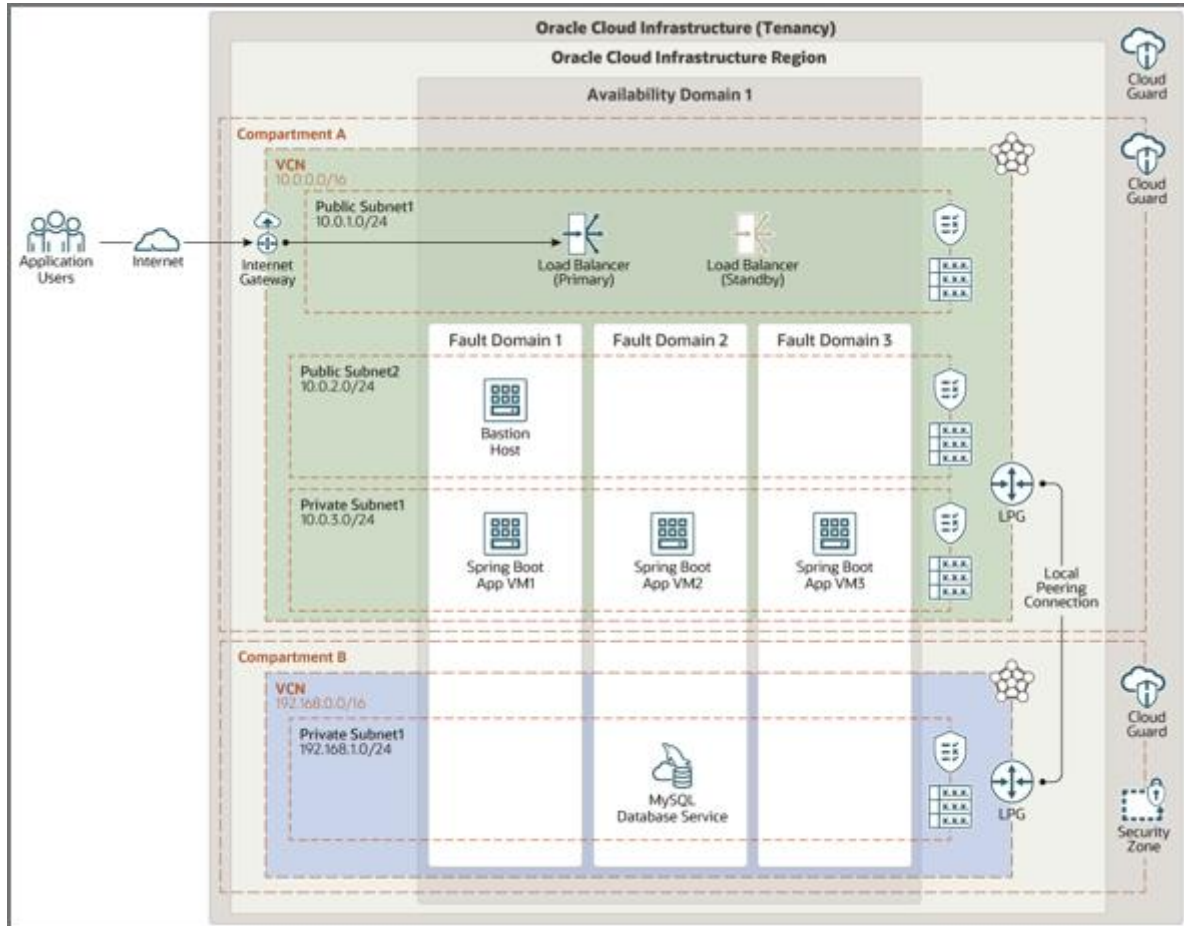
Admin: Add Doctor, Department Management, Reports

Resources Needs for infrastructure

1. VPC
2. SUBNET (PUBLIC & PRIVATE)
3. INTERNET GATEWAY
4. ROUT TABLE (CONNECT WITH IGW)
5. NAT GATEWAY
6. ROUT TABLE (CONNECT WITH NAT)
7. NACL (INBOUND & OUTBOUND RULE)
8. SECURITY GROUP (INBOUND & OUTBOUND)
9. EC2 INSTANCES (APP-SERVER, JUMPBOX, DB-SERVER)
10. Load Balancer

11. Target Group
12. Auto Scaling Group

Architecture:



Detailed Steps:

1. Creating VPC:

- **VPC:** VPC stands for Virtual Private Cloud. It's a service provided by cloud computing platforms like Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure that allows users to create a logically isolated section of the cloud where they can launch resources like virtual machines, databases, and storage networks.

AWS Account -> VPC -> Create VPC(healthcare-vpc)->

VPC > Your VPCs > Create VPC

Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

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 - optional
Creates a tag with a key of 'Name' and a value that you specify.

healthcare-vpc

IPv4 CIDR block [Info](#)
☒ IPv4 CIDR manual input
☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
192.168.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)
☒ No IPv6 CIDR block
☐ IPAM-allocated IPv6 CIDR block
☐ Amazon-provided IPv6 CIDR block
☐ IPv6 CIDR owned by me

Tenancy [Info](#)
Default

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	
Q Name X	Q healthcare-vpc X	Remove tag

Add tag

You can add 49 more tags.

Cancel Create VPC

2. Creating Subnets:

AWS subnets are associated with a specific availability zone within a region. Each subnet is tied to a particular availability zone, providing a way to distribute resources across different zones for fault tolerance and high availability.

Requirement: Create 2 Subnets one is for Application servers and another is for Database. According to our requirement the application servers to be created at public subnet as these servers can be accessed by public people. The DB server to be created at private subnet as it contains data.

Subnet1(public): AWS Account→ Subnet→ Create Subnet(healthcare-public-subnet-1)→



Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block [Info](#)
Choose the IPv4 VPC CIDR block to create a subnet in.

IPv4 subnet CIDR block
 256 IPs
< > ^ v

▼ **Tags - optional**

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="healthcare-public-subnet-1"/>	<input type="button" value="Remove"/>
<input type="button" value="Add new tag"/>		
You can add 49 more tags.		
<input type="button" value="Remove"/>		
<input type="button" value="Add new subnet"/>		

Subnet2(private): healthcare-private-subnet-2
Subnet1(private): AWS Account→ Subnet→ Create Subnet(healthcare-private-subnet-2)→



Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

healthcare-private-subnet-2

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

Asia Pacific (Singapore) / ap-southeast-1b

IPv4 VPC CIDR block [Info](#)
Choose the IPv4 VPC CIDR block to create a subnet in.

192.168.0.0/16

IPv4 subnet CIDR block

192.168.2.0/24256 IPs

<>^v

▼ **Tags - optional**

Key	Value - optional	
<div>Q NameX</div>	<div>Q healthcare-private-subnet-2X</div>	<div>Remove</div>
<div>Add new tag</div> <p>You can add 49 more tags.</p> <div>Remove</div>		
<div>Add new subnet</div>		

CancelCreate subnet

Subnet2(private): healthcare-public-subnet-2

Subnet1(public): AWS Account→ Subnet→ Create Subnet(healthcare-public-subnet-3)→



VPC

VPC ID
Create subnets in this VPC.
vpc-04f3e5986c952522a (healthcare-vpc)

Associated VPC CIDRs

IPv4 CIDRs
192.168.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
healthcare-public-subnet3
The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Asia Pacific (Singapore) / ap-southeast-1c

IPv4 VPC CIDR block [Info](#)
Choose the IPv4 VPC CIDR block to create a subnet in.
192.168.0.0/16

IPv4 subnet CIDR block
190.168.3.0/24 256 IPs

▼ Tags - optional

Key	Value - optional	
Name	healthcare-public-subnet3	Remove

Add new tag
You can add 49 more tags.

Remove

Add new subnet

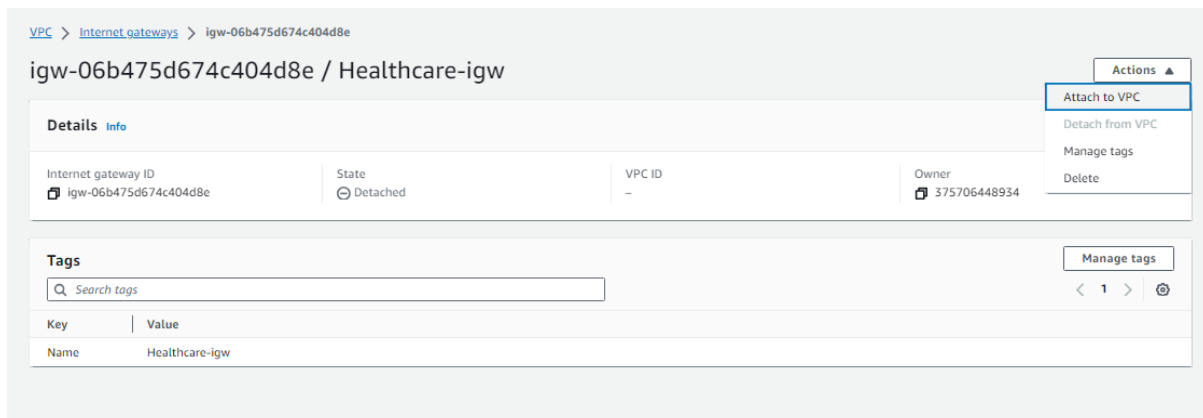
Cancel Create subnet

Create Internet-Gateways:

In Amazon Web Services (AWS), an internet gateway is a crucial component that allows communication between instances in your VPC (Virtual Private Cloud) and the internet. It acts as a gateway or entry/exit point for network traffic between your VPC and the internet.

Requirement : We need to attach Internet Gateway with the public subnet of VPC to have internet access between resources present in public subnet and Internet.

Create IGW: Healthcare-igw



VPC > Internet gateways > igw-06b475d674c404d8e

igw-06b475d674c404d8e / Healthcare-igw

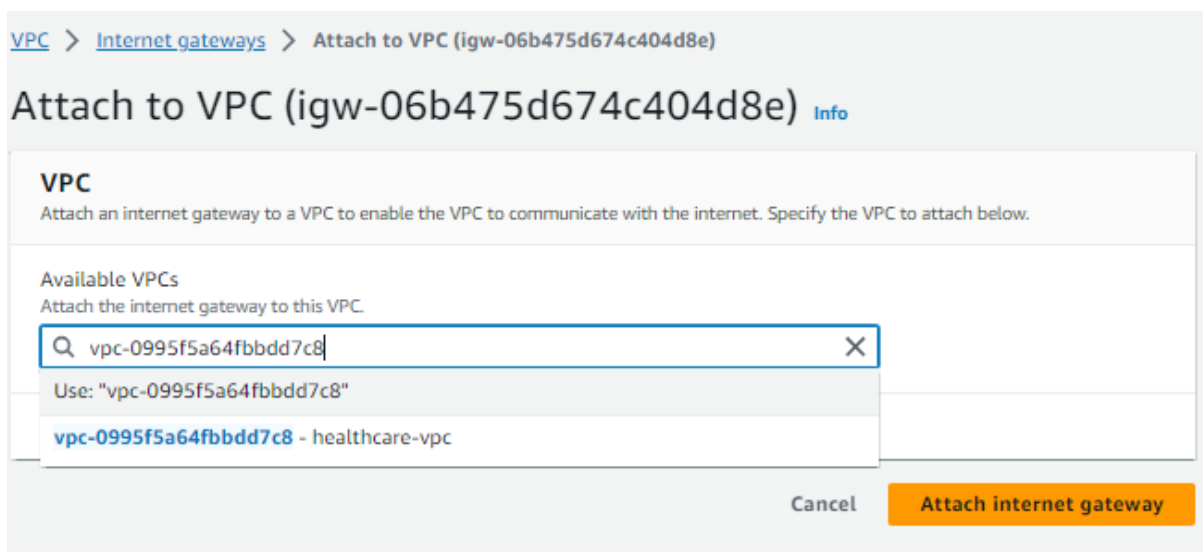
Details [Info](#)

Internet gateway ID igw-06b475d674c404d8e	State Detached	VPC ID -	Owner 375706448934
--	-------------------	-------------	-----------------------

Tags [Manage tags](#)

Search tags

Key	Value
Name	Healthcare-igw



VPC > Internet gateways > Attach to VPC (igw-06b475d674c404d8e)

Attach to VPC (igw-06b475d674c404d8e) [Info](#)

VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

Search: vpc-0995f5a64fbbdd7c8

Use: "vpc-0995f5a64fbbdd7c8"

vpc-0995f5a64fbbdd7c8 - healthcare-vpc

Cancel **Attach internet gateway**

Create Route Tables:

In AWS (Amazon Web Services), a route table is a key component of the Virtual Private Cloud (VPC) networking setup. It's essentially a set of rules, or a table, that defines how network traffic should be directed within a VPC.

Purpose Of Rout Table:

Route tables define the paths for network traffic within a subnet or Virtual Private Cloud (VPC), determining how data is directed. They facilitate communication between subnets, control outbound traffic, and are crucial for effective network routing in cloud environments.

AWS Dashboard→Search Bar (search Rout Table)→Create Route Table

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.

vpc-0995f5a64fbbdd7c8 (healthcare-vpc) ▼

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="healthcare-rtb"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

Associate Subnets and Add Rules:

After creating Rout Table, go to the subnet associations and associate with required subnet.

Route Table→Subnet Association→Associate healthcare-public-subnet-1



rtb-0d2016f2a09ba0f95 / healthcare-rtb Actions

Details [info](#)

Route table ID
rtb-0d2016f2a09ba0f95

VPC
vpc-0995f5a64fbdd7c8 | healthcare-vpc

Main
No

Owner ID
375706448934

Explicit subnet associations
[subnet-0b6ae0238d5e3c692](#) / [healthcare-public-subnet-1](#)

Edge associations
-

[Routes](#) | [Subnet associations](#) | [Edge associations](#) | [Route propagation](#) | [Tags](#)

Routes (1)
Both Edit routes

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)

< 1 >

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input type="checkbox"/>	healthcare-public-subnet-1	subnet-0b6ae0238d5e3c692	192.168.1.0/24	-	rtb-0d2016f2a09ba0f95 / healthcare-rtb
<input checked="" type="checkbox"/>	healthcare-private-subnet-2	subnet-0804b591d8a664eb0	192.168.2.0/24	-	Main (rtb-0fa2d834586f4745b)

Selected subnets

[subnet-0804b591d8a664eb0 / healthcare-private-subnet-2](#)

Cancel Save associations

Route Table→Edit Route→Add Rout (0.0.0.0/0)→Connect Igw

Edit routes

Destination	Target	Status	Propagated
192.168.0.0/16	<div>local</div> <div>local</div>	Active	No
<input type="text" value="0.0.0.0/0"/>	<div>Internet Gateway</div> <div> <input type="text" value="igw-06b475d674c404d8e"/> <div> Use: "igw-06b475d674c404d8e" igw-06b475d674c404d8e (Healthcare-igw) </div> </div>	-	No

Add route

Cancel Preview Save changes

Create NAT Gateway:

A Network Address Translation (NAT) Gateway is a managed AWS service that allows private subnet resources to initiate outbound internet traffic while remaining hidden from the public internet.

Create NAT gateway and connect with healthcare-public-subnet-1 and allocate elastic ip

Nat gateway→Subnet(healthcare-public-subnet-1)→Allocate Elastic ip→Create NAT gateway

Create NAT gateway [Info](#)

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the Internet.

NAT gateway settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.

Connectivity type
Select a connectivity type for the NAT gateway.
☒ Public
☐ Private

Elastic IP allocation ID [Info](#)
Assign an Elastic IP address to the NAT gateway.
 [Allocate Elastic IP](#)

[Additional settings](#) [Info](#)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="healthcare-natgw"/>	Remove

[Add new tag](#)
You can add 49 more tags.

[Cancel](#) [Create NAT gateway](#)

Create Route Table for NAT Gateway:

Here we create a route table and associate with healthcare-private-subnet-2 after association add rout (0.0.0.0/0) with Nat gateway.

Route Table→Name rtb→Connect VPC



Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key

Value - optional

Remove

Add new tag

You can add 49 more tags.

Cancel

Create route table

Route Table→Subnet Association→Associate All Private-Subnet

Change which subnets are associated with this route table.

Available subnets (2/3)

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input type="checkbox"/>	healthcare-public-subnet3	subnet-0ec1590c32e6a2390	192.168.3.0/24	-	rtb-0bbcae65a24ab73ad / health
<input checked="" type="checkbox"/>	healthcare-private-subnet2	subnet-0448c0579ceb8b587	192.168.2.0/24	-	rtb-0bbcae65a24ab73ad / health
<input checked="" type="checkbox"/>	healthcare-private-subnet1	subnet-0743d548632a6fc46	192.168.1.0/24	-	rtb-0bbcae65a24ab73ad / health

Selected subnets

Cancel

Save associations

Route Table→Add route (0.0.0.0/0)→Target (Nat gateway)

Edit routes

Destination	Target	Status	Propagated
192.168.0.0/16	<input type="text" value="local"/> <input type="text" value="local"/>	Active	No
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="NAT Gateway"/> <input type="text" value="nat-077591051791d7fa8"/> <p>Use: "nat-077591051791d7fa8"</p> <input type="text" value="nat-077591051791d7fa8 (healthcare-natgw)"/>	-	No

Add route

Remove

Cancel

Preview

Save changes

After Configuring the required resources under VPC (Subnets, Internet-gateways, NAT-gateways, Route Tables). We need to create 4 EC2-Instances under the VPC and deploy the Spring-boot Applications.

Create Security Group:

It performs the function of a virtual firewall, managing the inbound and outbound traffic for one or more Amazon EC2 instances or other AWS services within a VPC. Security group have two rule i.e. ,

1. Inbound Rule: These outline the types of traffic that are permitted to use the resources. It serves as a virtual firewall, controlling the traffic going in and coming out of a VPC for one or more Amazon EC2 instances or other AWS services.

2. Outbound Rule: These regulate the traffic that is permitted to depart from the resources. The destination for incoming traffic is dealt with by outbound rules. They may be forwarded to an alternative Security Group, a CIDR block, a single IPv4 or IPv6 address, or all three.

Four Security Groups need to be created i.e app-server-Sg, Jumpbox-Sg, Database-Sg, Loadbalancer-Sg.

Create security group

A security group acts as a virtual firewall for your instances to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name:

Description:

VPC:

Inbound rules

Type	Protocol	Port range	Source	Action	Description - optional
SSH	TCP	22	myip-01	allow	
Custom TCP	TCP	8080	myip-01	allow	

Outbound rules

Type	Protocol	Port range	Destination	Action	Description - optional
All traffic	All	All	Custom	allow	

Tags - optional

No tags associated with this resource.

Same process for all Security Group only difference in rule(inbound and outbound)

Create EC-2 instances:

EC2 offers a variety of instance types optimized for different use cases, including compute-optimized, memory-optimized, and storage-optimized instances. Users can choose instances with the right balance of CPU, memory, storage, and networking capabilities.

Using Three EC-2 Instances (On Public-Subnets):

Here 3 EC2 Instances

1. App-server-vm1

App-server-vm1 is a virtual machine instance, likely representing an application server, within a cloud-based infrastructure.

2. Jumpbox-vm

A jumpbox virtual machine, also known as a bastion host, provides a secure entry point for accessing and managing other machines within a network.

3. Date-base-vm

database-vm1 is a virtual machine instance designed to host and manage database services within a cloud environment.

- Open AWS dashboard the go to search bar and search EC2 then click on EC2 service then create EC2 instance.

Instance→Launch Instance→ Name→ Application and OS Images→Instance type→Key Pair→Network Setting(select vps, Select subnet, Auto-assign public IP, Security group name – required)→Launch



Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

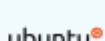
▼ Application and OS Images (Amazon Machine Image) [Info](#)


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

Quick Start


Amazon
Linux
aws

macOS

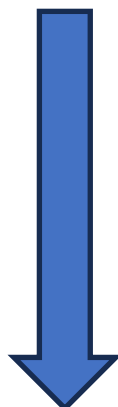

Ubuntu


Windows


Red Hat


SUSE L



[Browse more AMIs](#)
Including AMIs from



Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-0fa377108253bf620 (64-bit (x86)) / ami-05f8c2ee58e71f8e6 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-12-07

Architecture

AMI ID

64-bit (x86)

ami-0fa377108253bf620

Verified provider

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0146 USD per Hour

On-Demand Windows base pricing: 0.0192 USD per Hour

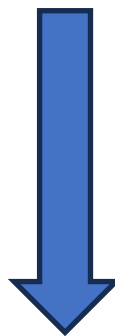
On-Demand RHEL base pricing: 0.0746 USD per Hour

On-Demand SUSE base pricing: 0.0146 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 - *required*

vm2-key

▼

↻

Create new key pair

▼

Network settings

Info

VPC - *required* | Info

vpc-0995f5a64fbbdd7c8 (healthcare-vpc)

▼

192.168.0.0/16

↻

Subnet | Info

subnet-0b6ae0238d5e3c692

healthcare-public-subnet-1

▼

VPC: vpc-0995f5a64fbbdd7c8

Owner: 375706448934

Availability Zone: ap-southeast-1a

IP addresses available: 250

CIDR: 192.168.1.0/24

↻

Create new subnet

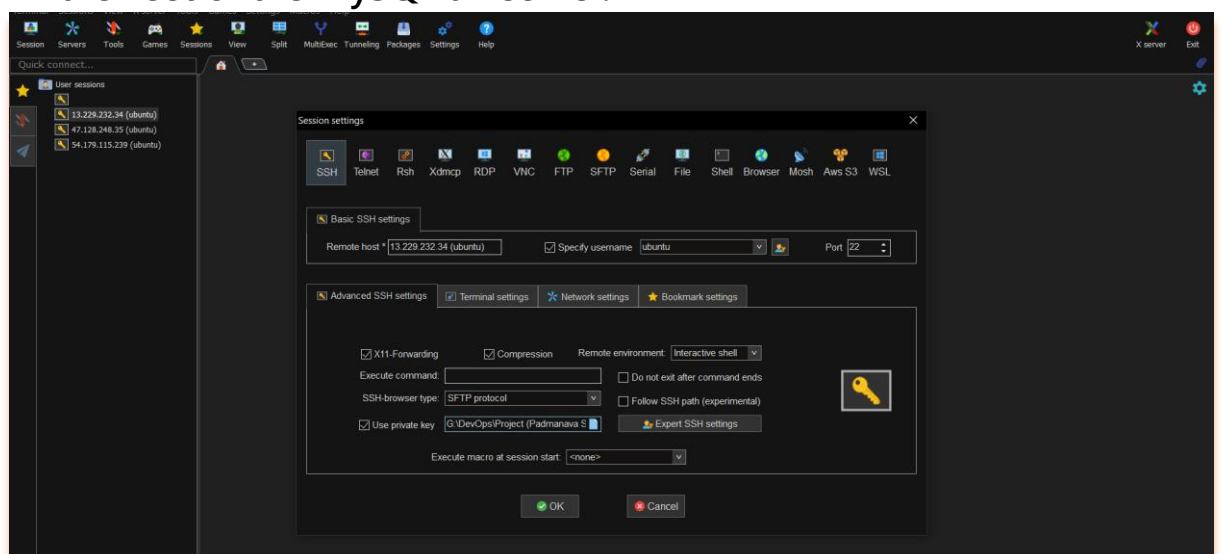
🔗

Auto-assign public IP | Info

Same procedure applies on rest of three instances.

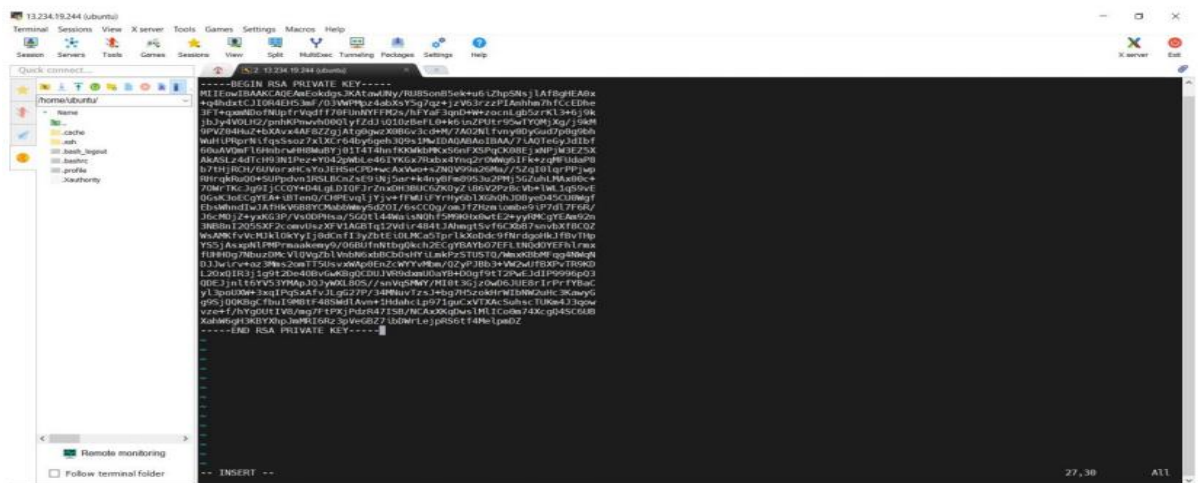
Lunch vm in the Terminal:

After creating instance then copy the public Ip of jump box vm and open the terminal like mobaXterm & connect with database-vm then set of the MySQL dB server.



Procedure To Connection of Database-vm And Set Up of MYSQL Database:

- ➔ Open GitHub spring-boot-db. repo then go to src→resources→application. properties
(spring.datasource.url=jdbc: mysql://mysqladb:3306/mydb)→put your private ip of database-vm
- ➔ Open jumpbox-vm
- ➔ Go to root user (sudo su -)
- ➔ Update machine (apt-get update)
- ➔ Open vi key.pem



- ➔ Open key on word pad copy & paste on vi editor
- ➔ Give permission (chmod 400 vm2-key.pem)
- ➔ Write ssh command to connect to database machine (ssh -i "vm2-key.pem" [ubuntu@192.168.2.96](https://192.168.2.96))
- ➔ Install mysql(apt-get install mysql-server)
- ➔ Install mysql client (apt-get install mysql-client)
- ➔ Enter into mysql server (mysql -u root -p) & give the password root
- ➔ Create database
- ➔ After enter into MySQL server then apply MySQL commands
 - show databases;
 - create database mydb;
 - use mydb;
- ➔ After creation of database then exit form MySQL and configure the bind_Address.

```

#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket             = /var/run/mysqld/mysqld.sock
# port               = 3306
# datadir            = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir             = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address        = 127.0.0.1
bind-address         = 0.0.0.0
mysqlx-bind-address  = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size      = 16M
-- INSERT --

```

- ➔ After configure the bind_adress then enter into MySQL server
 - ➔ Restart the services (service mysql restart & systemctl restart MySQL)
 - ➔ Create user (MySQL> CREATE USER 'root'@'%' IDENTIFIED BY 'root')
 - ➔ Grant all privileges (GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION;)
 - ➔ After grant privileges then flush privileges(FLUSH PRIVILEGES;)
- This is procedure to set up the MySQL database.

Procedure To Set Up App-server-vm1 & App-server-vm2:

- ➔ Open App-server vm as same as jumpbox vm
- ➔ Go to root user (sudo su -)
- ➔ Update the system (apt-get update)
- ➔ Install java (apt-get install openjdk-8-jdk)
- ➔ Go to opt (cd /opt)
- ➔ Install maven (wget <https://dlcdn.apache.org/maven/maven-3/3.9.6/binaries/apache-maven-3.9.6-bin.tar.gz>)
- ➔ You will get a tar file & untar this file (tar -xvzf tarfile)

- ➔ Check list(ls) then rename untar file (mv untar file maven)
 - ➔ Go to your GitHub repo copy the java-springboot-db url
 - ➔ Clone it in your machine (git clone url)
 - ➔ Go to your java-springboot-db folder
 - ➔ Create Artifact (/opt/maven/bin/mvn clean package)
 - ➔ Check list (ls) here we get a .war or .jar file
 - ➔ Give the java command (java -jar .war) file
 - ➔ After that take public ip of app-server then check in your browser with port 8080
- After completion of all app-server-vm set up then we will use this app sever as service.

Procedure to making app-server-vm as service:

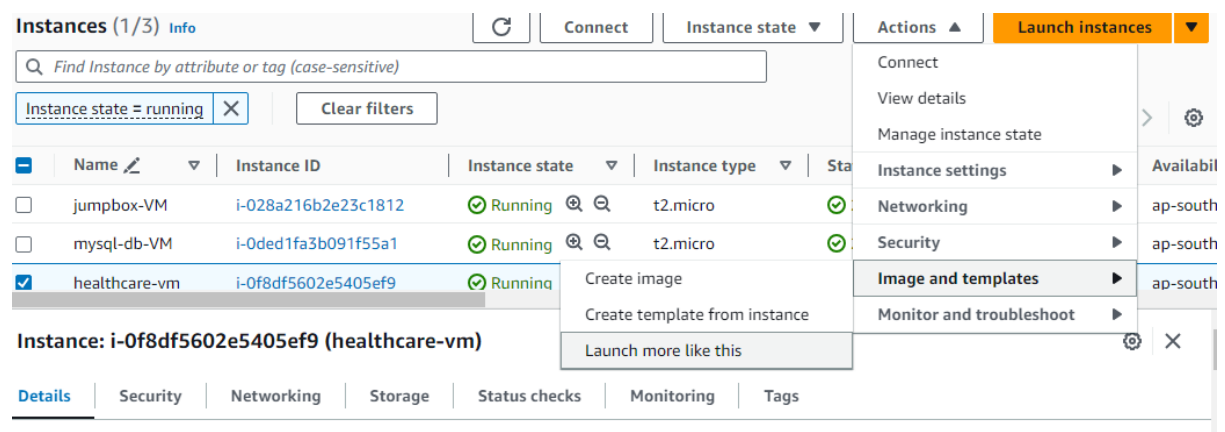
- ➔ Stop the spring boot app which is running in foreground.
- ➔ After Stop this app then go to root user and then enter to this (cd /etc/systemd/system/)
- ➔ Then create a file i. e.; vi (mybootapp.service) and wriet inside this file
- ➔ [Unit]
- ➔ Description=My Spring Boot Application
- ➔ After=syslog.target
- ➔ [Service]
- ➔ User=ubuntu
- ➔ ExecStart=java -jar /opt/ [aws-project](#)/target/springboot-app-1.0.war
- ➔ SuccessExitStatus=143
- ➔ [Install]
- ➔ WantedBy=multi-user.target
- ➔ After write this script we should run this four command to make spring-boot-app as a servise i. e.;
- ➔ (systemctl daemon-reload
- ➔ systemctl start mybootapp
- ➔ systemctl enable mybootap
- ➔ systemctl status mybootapp)

Create AMI (Amazon Machine Image):

Amazon Machine Image (AMI) is a pre-configured template that contains the software configuration (operating system, application server, applications, and related configurations) required to launch an instance in Amazon Elastic Compute Cloud (EC2). It's like a snapshot of a virtual machine that can be used to create multiple instances with the same configuration.

Before we are set the auto scaling group first of all we create a Ami of app-server-vm.

App-server-vm→Action→image & templates→create image→image name→Image description(optional)→create image.

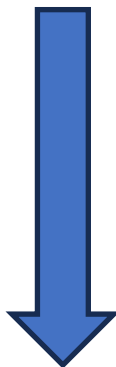


The screenshot shows the AWS Management Console 'Instances' page. At the top, there's a search bar and filters. Below, a table lists three instances: 'jumpbox-VM', 'mysql-db-VM', and 'healthcare-vm'. The 'healthcare-vm' instance is selected, and its context menu is open, showing options like 'Connect', 'View details', 'Manage instance state', 'Instance settings', 'Networking', 'Security', 'Image and templates', and 'Monitor and troubleshoot'. The 'Image and templates' option is highlighted. Below the table, there's a section for the selected instance, 'healthcare-vm', with tabs for 'Details', 'Security', 'Networking', 'Storage', 'Status checks', 'Monitoring', and 'Tags'.

	Name	Instance ID	Instance state	Instance type	Status
<input type="checkbox"/>	jumpbox-VM	i-028a216b2e23c1812	Running	t2.micro	✓
<input type="checkbox"/>	mysql-db-VM	i-0ded1fa3b091f55a1	Running	t2.micro	✓
<input checked="" type="checkbox"/>	healthcare-vm	i-0f8df5602e5405ef9	Running		✓

Instance: i-0f8df5602e5405ef9 (healthcare-vm)

Actions menu options: Connect, View details, Manage instance state, Instance settings, Networking, Security, Image and templates, Monitor and troubleshoot.



Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 Instance. You can create an image from the configuration of an existing instance.

Instance ID
i-0F8dF5602e5405ef9 (healthcare-vm)

Image name
healthcare-AMI
Maximum 127 characters. Can't be modified after creation.

Image description - optional
healthcare-AMI
Maximum 255 characters

No reboot
☐ Enable

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev...	Create new snapshot f...	8	EBS General Purpose S...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

[Add volume](#)

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

☒ Tag image and snapshots together
Tag the image and the snapshots with the same tag.

☐ Tag image and snapshots separately
Tag the image and the snapshots with different tags.

Key
Q Name X

Value - optional
Q healthcare-AMI X Remove
Use "healthcare-AMI"

[Add new tag](#)
You can add up to 49 more tags.

[Cancel](#) [Create image](#)

Create Launch Template:

Launch Templates reduce the number of steps required to create an instance by capturing all launch parameters within one resource. This makes the process easy to reproduce.

After creation of AMI then we proceed to create launch template using this AMI.

EC2 Dashboard → Launch Template → LT name → AMI → Instance type → Key pair → Network setting (only SG) → Create



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*

healthcare-launch-teplate

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

Template version description

healthcare-launch-teplate

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

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ 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

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

Recents

My AMIs

Quick Start

☐ Don't include in launch template

☒ Owned by me

☐ Shared with me



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

healthcare-AMI

ami-0892f4aa51a0be810

2023-12-29T19:13:29.000Z

Virtualization: hvm

ENA enabled: true

Root device type: ebs

Description

healthcare-AMI

Architecture

x86_64

AMI ID

ami-0892f4aa51a0be810



▼ 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: 0.0146 USD per Hour
On-Demand Windows base pricing: 0.0192 USD per Hour
On-Demand RHEL base pricing: 0.0746 USD per Hour
On-Demand SUSE base pricing: 0.0146 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

vm2-key

[Create new key pair](#)

▼ Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template

[Create new subnet](#)

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

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

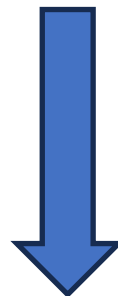
Security groups [Info](#)

Select security groups

[Compare security group rules](#)

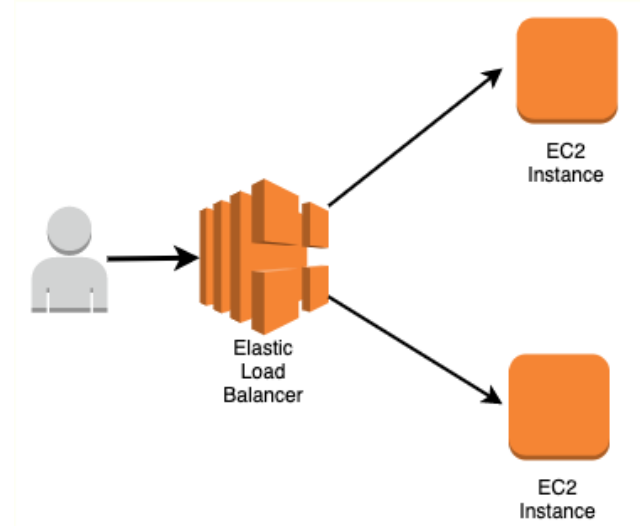
healthcare-SG sg-0cb50dc12c1eb8f93 ✕
VPC: vpc-04f3e5986c952522a

► [Advanced network configuration](#)



Create Load Balancer:

In AWS, a load balancer helps distribute incoming network traffic across multiple targets, such as EC2 instances, containers, IP addresses, or Lambda functions. This distribution ensures no single resource gets overwhelmed, thereby enhancing the fault tolerance and availability of your applications.



In load balancer there are 4 main resources are present e.g.,

1. **Fault Tolerance:** A resilient load balancer is designed to tolerate faults, errors, or failures without compromising its core functionality.
2. **Redundancy:** Resilient load balancers often employ redundancy by having multiple instances or nodes. If one node fails, another can take over to ensure uninterrupted load balancing. Redundancy can be implemented at both hardware and software levels.
3. **Health Checking:** Load balancers continuously monitor the health of the backend servers to which they distribute traffic. Health checking involves periodically verifying the availability and responsiveness of each server.
4. **Auto-Scaling Integration:** When there is a sudden increase in traffic, auto-scaling can dynamically add more backend servers to handle the load. The load balancer then adjusts its distribution accordingly.

- ❖ Before creating load balancer first, we create the Target group.

Create Target Group:

In Amazon Web Services (AWS), a "target group" refers to a component used in the Elastic Load Balancing (ELB) service. Target groups are essentially groups of resources or instances—such as EC2 instances or IP addresses—where incoming traffic is routed based on the rules and conditions defined in the associated load balancer.

Aws (search bar)→Target group→Basic configuration→Register your target

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 IPv6-to-IPv6 NAT.

☐ **Lambda function**

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

☐ **Application Load Balancer**

- 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

healthcare-Tg

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

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation.

HTTP 80

1-65535



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.

healthcare-vpc
vpc-0995f5a64fbbdd7c8
IPv4: 192.168.0.0/16

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.

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP

Health check path

Use the default path of "/" to ping the root, or specify a custom path if preferred.

/department

Up to 1024 characters allowed.

Advanced health check settings

Attributes

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

Tags - optional

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

Cancel

Next

Available instances (5)

Filter instances

<input type="checkbox"/>	Instance ID	Name	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
<input type="checkbox"/>	i-0f8f5602a5425e9f9	healthcare-vm	Running	healthcare-SG	ap-southeast-1a	192.168.1.106	subnet-0743d548632a6b146	December 28, 2023, 21:15 (UTC+05:30)
<input type="checkbox"/>	i-0dd1153b091155a1	mysql-db-VM	Running	mysql-db-SG	ap-southeast-1a	192.168.1.176	subnet-0743d548632a6b146	December 28, 2023, 19:21 (UTC+05:30)
<input type="checkbox"/>	i-02ba216b3c73c1812	jumpbox-VM	Running	jumpbox-SG	ap-southeast-1c	192.168.3.33	subnet-0e1590c32a6a2390	December 28, 2023, 19:20 (UTC+05:30)

0 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

8080

+0530 (separate multiple ports with comma)

Include as pending below

Review targets

Targets (0)

Filter targets

Show only pending

Remove all pending

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
No instances added yet										

Specify instances above, or leave the group empty if you prefer to add targets later.

0 pending

Cancel

Previous

Create target group




Create Load Balancer:

After creating Target Group then go to Load Balancer.

AWS (search bar)→Load balancer→Create Load balancer→load balancer type (choose application load balancer)→Basic Configuration→Scheme (internet facing)→Network mapping→Security Group→Listeners and routing→

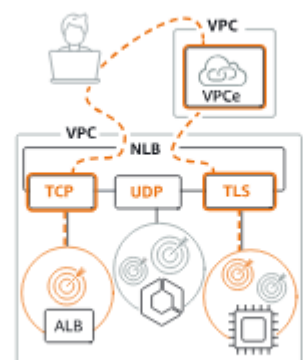
Application Load Balancer [Info](#)



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

► Classic Load Balancer - *previous generation*

Close



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.

healthcare-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 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.

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.

After creating Load Balancer then copy the dns address then paste it in your browser and health check on your Target Group

▼ Details

Load balancer type Application	Status ✔ Active	VPC vpc-0995f5a64fbbdd7c8 ↗	IP address type IPv4
Scheme Internet-facing	Hosted zone Z1LMS91P8CMLE5	Availability Zones subnet-0b6ae0238d5e3c692 ↗ ap-southeast-1a (apse1-az2) subnet-0804b591d8a664eb0 ↗ ap-southeast-1b (apse1-az1)	Date created December 26, 2023, 23:21 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:ap-southeast-1:375706448934:loadbalancer/app/healthcare-lb/78dd92b5260be03a		DNS name Info healthcare-lb-529757922.ap-southeast-1.elb.amazonaws.com (A Record)	



EC2 > Target groups

Target groups (1/1) [Info](#)

[Refresh](#) [Actions](#) [Create target group](#)

<input checked="" type="checkbox"/>	Name	ARN	Port	Protocol	Target type
<input checked="" type="checkbox"/>	healthcare-tg	arn:aws:elasticloadbalanci...	80	HTTP	Instance

Target group: healthcare-tg

2 Total targets	2 Healthy 0 Anomalous	0 Unhealthy	0 Unused	0 Initial	0 Draining
--------------------	-----------------------------	----------------	-------------	--------------	---------------

► **Distribution of targets by Availability Zone (AZ)**
Select values in this table to see corresponding filters applied to the Registered targets table below.

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](#).

healthcare-vpc
vpc-0995f5a64fbbdd7c8
IPv4: 192.168.0.0/16

[Refresh](#)

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.

☒ ap-southeast-1a (apse1-az2)

Subnet

subnet-0b6ae0238d5e3c692 healthcare-public-subnet-1

IPv4 address

Assigned by AWS

☒ ap-southeast-1b (apse1-az1)

Subnet

subnet-0804b591d8a664eb0 healthcare-private-subnet-2

⚠ The selected subnet does not have a route to an internet gateway. This means that your load balancer will not receive internet traffic.
You can proceed with this selection; however, for internet traffic to reach your load balancer, you must update the subnet's route table in the [VPC console](#).

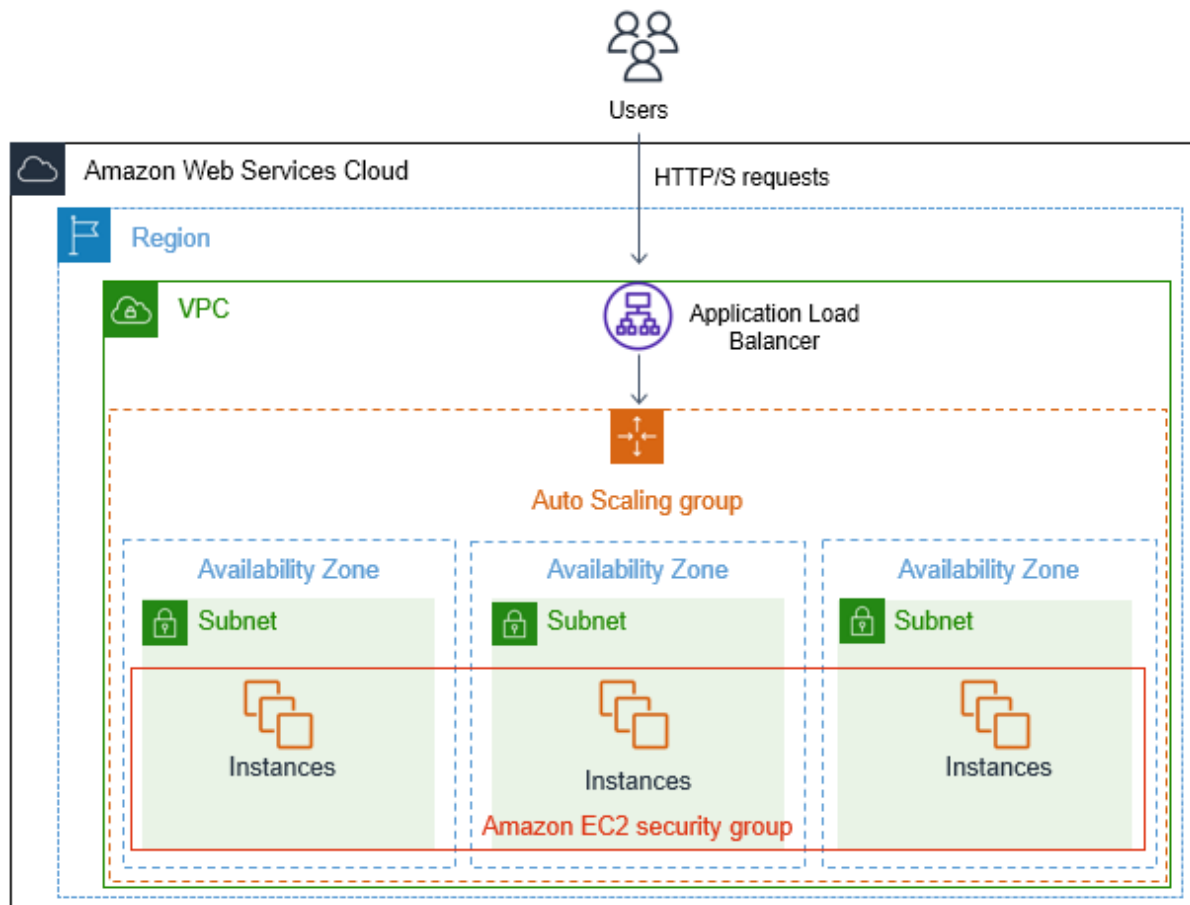
IPv4 address

Assigned by AWS

Create Autoscaling Group:

In Amazon Web Services (AWS), a "target group" refers to a component used in the Elastic Load Balancing (ELB) service. Target groups are essentially groups of resources or instances—such as EC2 instances or IP addresses—where incoming traffic is routed based on the rules and conditions defined in the associated load balancer.





Using Of ASG:

The most common way to scale is based on CPU usage, is by defining a threshold to specify when to add new instance and when to terminate, however, you can configure other criteria, for example, memory usage, disk IO, and even you can define your custom metric like a number of the webserver requests. so it depends on where you see the bottleneck on your environment on high load.

Go to EC-2 dashboard then go to search bar to search auto scaling group

Create ASG → Choose launch template → Name → launch template(customize) → Choose instance launch options → Configure advanced options – optional → Configure group size and scaling – optional → Review → Create ASG



Name

Auto Scaling group name

Enter a name to identify the group.

healthcare-ASG

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

Launch template Info

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

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.

healthcare-launch-teplate

Create a launch template

Version

Default (1)

Create a launch template version

Description

healthcare-launch-teplate

AMI ID

ami-0892f4aa51a0be810

Key pair name

vm2-key

Launch template

healthcare-launch-teplate

lt-0450f3e772229f85f

Security groups

-

Security group IDs

sg-0cb50dc12c1eb8f93

Instance type

t2.micro

Request Spot Instances

No

Additional details

Storage (volumes)

-

Date created

Sat Dec 30 2023 00:52:46 GMT+0530 (India Standard Time)

Cancel

Next

[Alt+S]

Choose instance launch options Info

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

Instance type requirements Info

Override launch template

Launch template

healthcare-launch-teplate

lt-0450f3e772229f85f

Version

Default

Description

healthcare-launch-teplate

Instance type

t2.micro

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-04f3e5986c952522a (healthcare-vpc)

192.168.0.0/16

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

ap-southeast-1a | subnet-0743d548632a6fc46

(healthcare-private-subnet1)

192.168.1.0/24

ap-southeast-1b | subnet-0448c0579ceb8b587

(healthcare-private-subnet2)

192.168.2.0/24

ap-southeast-1c | subnet-0ec1590c32e6a2390

(healthcare-public-subnet3)

192.168.3.0/24

Create a subnet

Cancel

Skip to review

Previous

Next

Configure advanced options - optional

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
You can create a new load balancer or attach an existing load balancer to your Auto Scaling group.

☒ Attach to an existing load balancer
Choose from your existing load balancers.

☐ Attach to a new load balancer
Choose from your new load balancers.

Attach to an existing load balancer

Select the load balancer that you want to attach to your Auto Scaling group.

☒ Choose from your existing load balancers
☐ Choose from your new load balancers

Existing load balancer target group

Select the target group that you want to attach to your Auto Scaling group.

VPCLattice integration options

Select the VPC Lattice service that you want to attach to your Auto Scaling group.

☒ No VPC Lattice service
☐ Attach to a VPC Lattice service

Health checks

EC2 health checks

☒ Enable EC2 health checks
EC2 health checks are used to monitor the health of your EC2 instances. You can choose to enable EC2 health checks or not.

Additional health check options - optional

☒ Enable EC2 health checks
EC2 health checks are used to monitor the health of your EC2 instances. You can choose to enable EC2 health checks or not.

Health check options

☒ Enable EC2 health checks
EC2 health checks are used to monitor the health of your EC2 instances. You can choose to enable EC2 health checks or not.

Additional settings

Monitoring

☒ Enable monitoring
Enable monitoring for your Auto Scaling group.

Default instance profile

☒ Enable default instance profile
Enable default instance profile for your Auto Scaling group.

Configure group size and scaling - optional

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size

Desired capacity

Choose the number of instances that you want to run in your Auto Scaling group.

Scaling

Scaling limits

Choose the minimum and maximum number of instances that you want to run in your Auto Scaling group.

Automatic scaling - optional

☒ Enable automatic scaling
Enable automatic scaling for your Auto Scaling group.

Instance maintenance policy - new

Control availability and size during replacement events

Choose the replacement policy that you want to use for your Auto Scaling group.

☒ Replace all instances
☐ Replace some instances

Instance scale-in protection

☒ Enable instance scale-in protection
Enable instance scale-in protection for your Auto Scaling group.

After creating the Auto Scaling Group then go to your load balancer and copy the dns link and paste in your browser and check your output. After that you go to your Target Group check the health of your server. This is the all set up of the load balancer with Auto Scaling Group