

CAPSTONE PROJECT

Designing And Maintaining A Scalable Web Application in AWS



OBJECTIVES



01

Design and deploy a VPC

- ✓ Set up VPC and subnets.
- ✓ Attach Internet and NAT Gateways.
- ✓ Enforce security through Security Groups and Network ACLs.

02

Implement Auto Scaling and load balancing

- ✓ Configure an Auto Scaling Group with EC2 instances across multiple availability zones.
- ✓ Configure the ALB to span across multiple availability zones, improving fault tolerance..
- ✓ Define scaling policies to automatically add or remove instances based on CPU utilization or other custom metrics.

03

Configuring a DynamoDB Table in AWS

- ✓ Create and configure a DynamoDB table using the AWS Management Console.
- ✓ Enable interaction with DynamoDB using the Boto3 SDK in Python.
- ✓ Add and manage records (items) using attributes via the console.

BASIC TERMINOLOGIES

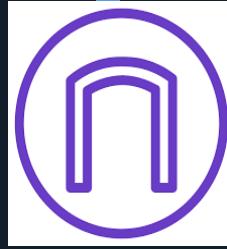
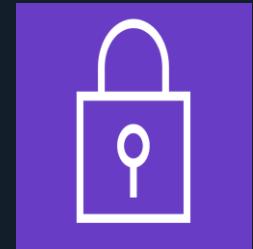
VPC

A logically isolated section of the public cloud.



Subnet

A segment of the VPC for isolating resources and controlling access within VPC.

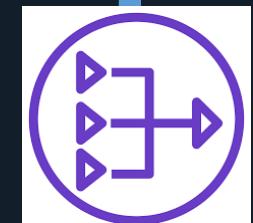


Internet Gateway

Gateway that enables communication between your VPC and the internet.

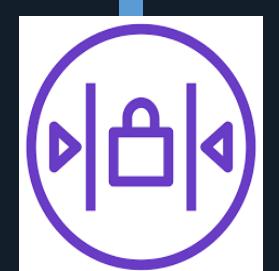
NAT Gateway

Allows instances in a private subnet to connect to the internet without exposing them to inbound connection from the internet.



Security Group

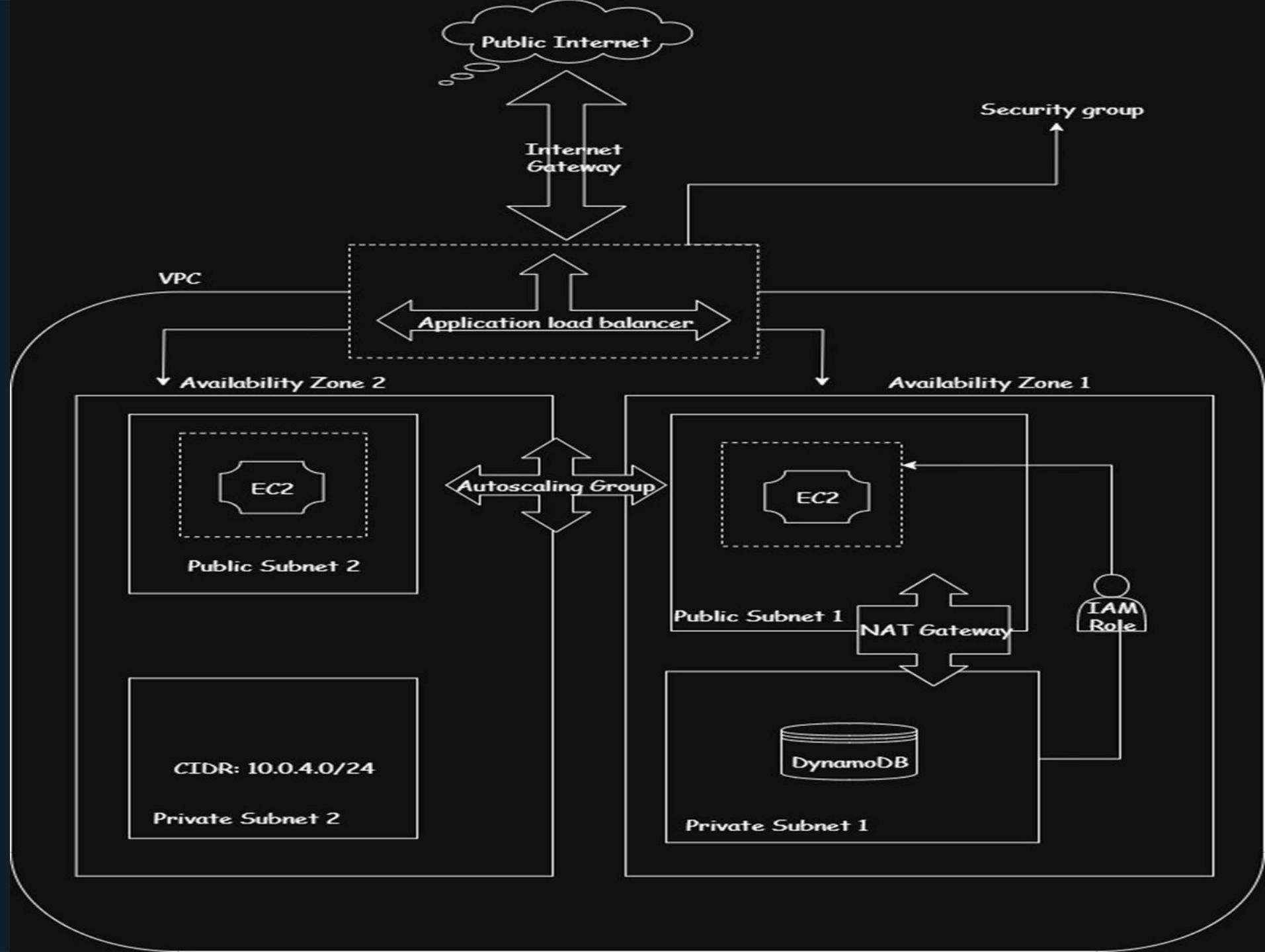
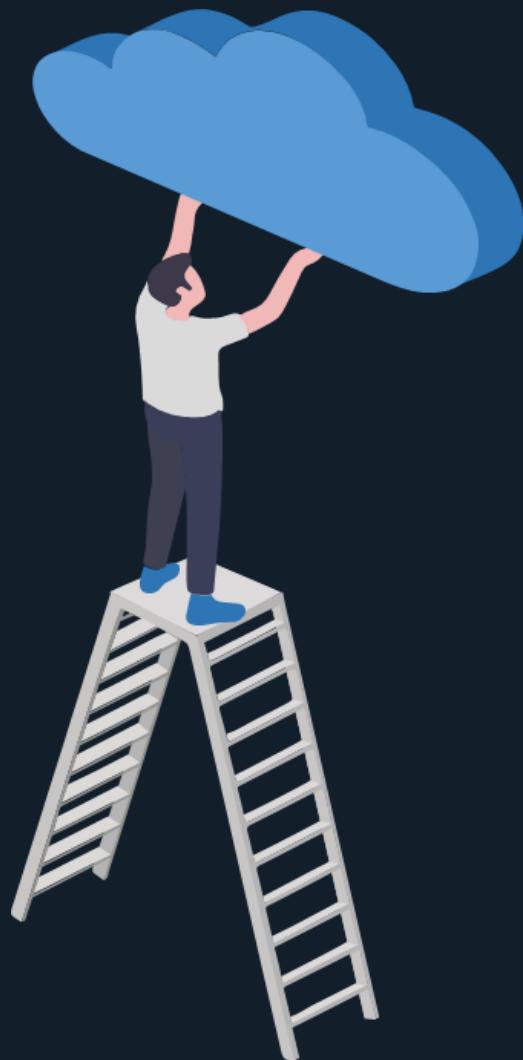
a virtual firewall that controls inbound and outbound traffic for your EC2 instances.



NACL

a stateless firewall that acts as an optional layer of security for subnets within a VPC, controlling inbound and outbound traffic.

THE ARCHITECTURE



Step1: Create a VPC

- ❖ Log in to AWS Console
 - ❖ Navigate to the **VPC Dashboard** by searching for "VPC" in the **AWS Services** search bar.
 - ❖ Click "**VPCs**" on the left panel.
 - ❖ Click the "**Create VPC**" button.
 - ❖ Choose "**VPC only**" or "VPC with subnets" if you want AWS to auto-generate other components.
 - ❖ Configure VPC Settings
 - ❖ Click "**Create VPC**"
- NOTES:**
- ❖ A well-chosen CIDR block, like 10.0.0.0/16, ensures sufficient IP addresses for all subnets and future scalability.
 - ❖ Defines whether instances launched in your VPC will run on shared hardware (multitenancy) or a dedicated hardware

VPC Created

aws | Search [Alt+S] | United States (N. Virginia) | Genius_Genie @ 2885-1884-1637 | VPC | Your VPCs | vpc-0eacf2424a1a33c78 | Actions X

You successfully created vpc-0eacf2424a1a33c78 / Room9-Capstone-VPC

vpc-0eacf2424a1a33c78 / Room9-Capstone-VPC

Details Info

VPC ID vpc-0eacf2424a1a33c78	State Available	Block Public Access Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-02cf66b22bed203ba	Main route table rtb-0efd0bc301bfc711a
Main network ACL acl-022ef68558e0b435d	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -
IPv6 CIDR (Network border group) -	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 288518841637

Resource map | CIDRs | Flow logs | Tags | Integrations

Resource map

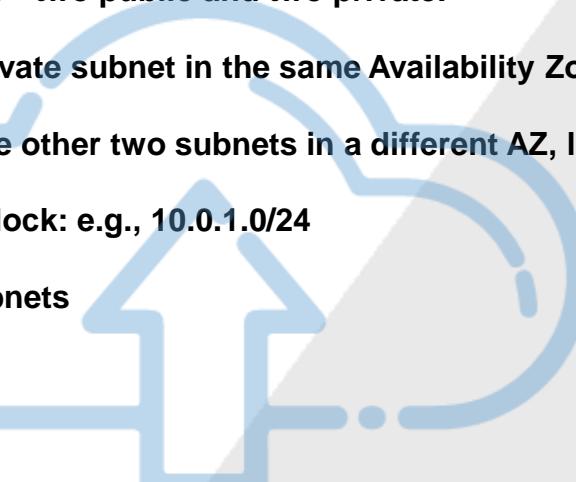
VPC Show details Your AWS virtual network Room9-Capstone-VPC	Subnets (0) Subnets within this VPC	Route tables (1) Route network traffic to resources rtb-0efd0bc301bfc711a	Network connections (0) Connections to other networks
---	---	--	---

Next : Select Subnets in the left panel (marked red)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step2: Create 4 subnets

- ❖ Navigate to **Subnets**
- ❖ **Click Create subnet**
- ❖ **Create four subnets in total—two public and two private.**
- ❖ **Pair one public and one private subnet in the same Availability Zone (AZ), such as us-east-1a. Do the same for the other two subnets in a different AZ, like us-east-1b.**
- ❖ **Select distinct IPv4 CIDR block: e.g., 10.0.1.0/24**
- ❖ **Repeat to create all four subnets**



Private subnet in AZ 1

- ✓ To create all subnets at a goal, use the add new subnet button
- ✓ Subnets are assigned distinct CIDR blocks (10.0.1.0/24, 10.0.2.0/24...) to segment the VPC's IP range and avoid overlap.

The screenshot shows the AWS VPC Subnets creation interface. The subnet is being configured with the following details:

- Subnet name:** Room9-Private-Subnet1
- Availability Zone:** United States (N. Virginia) / us-east-1a
- IPv4 VPC CIDR block:** 10.0.0.0/16
- IPv4 subnet CIDR block:** 10.0.2.0/24 (256 IPs)
- Tags:** Name: Room9-Private-Subnet1

A red box highlights the "Add new subnet" button at the bottom left of the form. The "Create subnet" button is located at the bottom right.

Subnets Created

aws | Search [Alt+S] | United States (N. Virginia) | Genius_Genie @ 2885-1884-1637

VPC Subnets

VPC dashboard < **Subnets (4)** Info Last updated less than a minute ago Actions Create subnet

Find subnets by attribute or tag

Subnet ID : subnet-0139ada187f04957d X Subnet ID : subnet-0566708cabd4bd4d1 X Subnet ID : subnet-006982ee98411768f X (+1) Clear filters

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR
Room9-Public-Subnet2	subnet-006982ee98411768f	Available	vpc-0eacf2424a1a33c78 Room9	Off	10.0.3.0/24	-	-
Room9-Public-Subnet1	subnet-0139ada187f04957d	Available	vpc-0eacf2424a1a33c78 Room9	Off	10.0.1.0/24	-	-
Room9-Private-Subnet1	subnet-0566708cabd4bd4d1	Available	vpc-0eacf2424a1a33c78 Room9	Off	10.0.2.0/24	-	-
Room9-Private-Subnet2	subnet-0a04939e31599fd91	Available	vpc-0eacf2424a1a33c78 Room9	Off	10.0.4.0/24	-	-

Select a subnet

✓ Next, create the internet Gateway, marked red in the left pane.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step3: Create Internet Gateway

- ❖ Go to **Internet Gateways**
- ❖ **Click Create internet gateway**
- ❖ **Give it a name and click on create VPC**
- ❖ Select the created VPC, click on actions
- ❖ Select attach to VPC and choose the VPC created

NOTES

- ❖ **Attaching the IGW to the VPC allows public resources, like load balancers, to be accessible globally.**

Internet Gateway Created

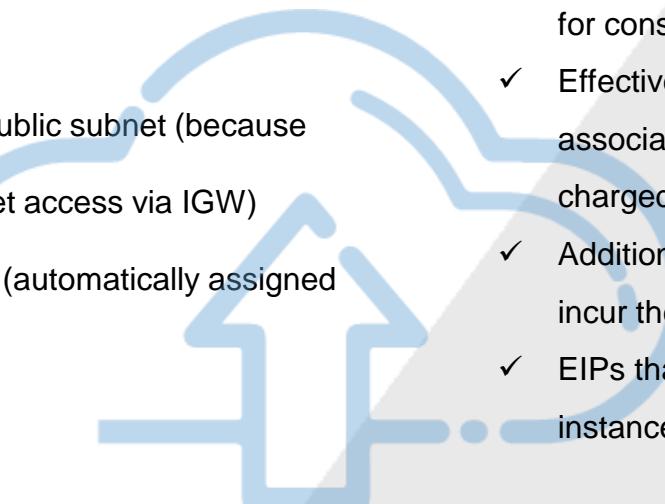
- ✓ Notice that a VPC id has been added in the Details section of your IGW

The screenshot shows the AWS VPC Internet Gateways page. A green success message at the top states: "Internet gateway igw-0c8e73e7d3406a9ca successfully attached to vpc-0eacf2424a1a33c78". Below this, the Internet gateway details are shown: ID igw-0c8e73e7d3406a9ca, State Attached, and VPC ID vpc-0eacf2424a1a33c78 | Room9-Capstone-VPC. The VPC ID is highlighted with a red box and an arrow points from the yellow notice text to it. The left sidebar shows the navigation path: VPC > Internet gateways > igw-0c8e73e7d3406a9ca. The "NAT gateways" link in the sidebar is circled in red.

✓ Next, create the NAT Gateway (marked red in the left pane)

Step4: Create NAT Gateway

- ❖ Go to **NAT Gateways** in the left pane
- ❖ Click **Create NAT gateway**
- ❖ **Configure NAT Gateway Settings (Name, subnet, etc)**
- ❖ For the subnet, Choose a public subnet (because NAT Gateway needs internet access via IGW)
- ❖ Click “**Allocate Elastic IP**” (automatically assigned by AWS)
- ❖ Click Create NAT Gateway



What to know about Elastic IPs

- ✓ An Elastic IP (EIP) is allocated to the NAT Gateway to provide a static public IP for consistent outbound communication.
- ✓ Effective **February 1, 2024**, EIP associated with a running instance is charged **(\$0.005) hourly**.
- ✓ Additional EIPs on the same instance will incur the same charge.
- ✓ EIPs that are not associated with any instance will also incur a charge.

NAT Gateway Created

aws | Search [Alt+S] | United States (N. Virginia) | Genius_Genie @ 2885-1884-1637 | VPC NAT gateways nat-09e89de516528bfc8 | Actions X

NAT gateway nat-09e89de516528bfc8 | Room9-NAT-Gateway was created successfully.

nat-09e89de516528bfc8 / Room9-NAT-Gateway

Details	Connectivity type	State	State message
NAT gateway ID nat-09e89de516528bfc8	Public	Pending	-
NAT gateway ARN arn:aws:ec2:us-east-1:288518841637:natgateway/nat-09e89de516528bfc8	Primary public IPv4 address -	Primary private IPv4 address -	Primary network interface ID -
VPC vpc-0eacf2424a1a33c78 / Room9-Capstone-VPC	Subnet subnet-0139ada187f04957d / Room9-Public-Subnet1	Created Tuesday 24 June 2025 at 19:06:52 GMT	Deleted -

Secondary IPv4 addresses | Monitoring | Tags

Secondary IPv4 addresses

Search

Private IPv4 address	Network interface ID	Status	Failure message
Secondary IPv4 addresses are not available for this nat gateway.			

C Edit secondary IPv4 address associations < 1 > ⚙️

✓ Next, Go to Route Tables to create route tables (marked red)

VPC dashboard EC2 Global View Filter by VPC: Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only Internet gateways Carrier gateways DHCP option sets Elastic IPs Managed prefix lists NAT gateways Peering connections Route servers New Security Network ACLs Security groups PrivateLink and Lattice Getting started Updated Endpoints Updated Endpoint services Service networks Updated Lattice services Resource configurations New CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step5: Create Route Tables

- ❖ Go to **Route Tables**
- ❖ **Click Create route table (create 2 route table: Public and Private)**
- ❖ Select the VPC created after giving it a name
- ❖ **Click Edit Routes** to add route
 - ❖ Select the IGW created to allow internet traffic (for public route)
 - ❖ Select the NAT Gateway created for the private route
- ❖ **Click edit subnet associations In the Subnet Associations tab**
- ❖ Associate your **public subnet with the public route**
- ❖ **Do the same for the private route (private subnet association)**

Route Table Created (Private Route Table)

AWS | Search [Alt+S] | United States (N. Virginia) | Genius_Genie @ 2885-1884-1637

VPC > Route tables > rtb-010481213f991d683

You have successfully updated subnet associations for rtb-010481213f991d683 / Room9-Private-RT.

rtb-010481213f991d683 / Room9-Private-RT

Actions

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-010481213f991d683	<input type="checkbox"/> No	2 subnets	-
VPC	Owner ID	-	
vpc-0eacf2424a1a33c78 Room9-Capstone-VPC	288518841637		

Routes **Subnet associations** **Edge associations** **Route propagation** **Tags**

Explicit subnet associations (2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Room9-Private-Subnet1	subnet-0566708cabd4bd4d1	10.0.2.0/24	-
Room9-Private-Subnet2	subnet-0a04939e31599fd91	10.0.4.0/24	-

Edit subnet associations

Subnets without explicit associations (0)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
		No subnets without explicit associations	All your subnets are associated with a route table.

Edit subnet associations

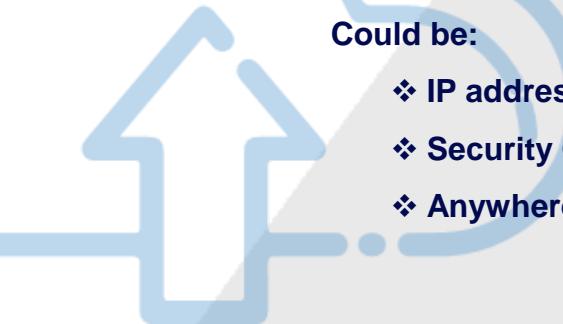
Next, create the security groups

Step6: Create and Configure Security Groups

- ❖ On the left menu, select **Security Groups**
- ❖ Click Create security group
- ❖ Set Up Basic Details (name, description, VPC)
- ❖ Add inbound and outbound rules
- ❖ Click create security groups

NOTES

- ❖ **Protocol:** Defines the type of network communication allowed (e.g., TCP, UDP).
- ❖ **Port Range:** Specific ports or port ranges allowed for the selected protocol. E.g., HTTP = Port 80.
- ❖ **Port:** Specifies the origin of the incoming traffic.
Could be:
 - ❖ IP address or range
 - ❖ Security Group
 - ❖ Anywhere (0.0.0.0/0)



Step 6: Inbound and Outbound Rules

AWS | Search [Alt+S] | United States (N. Virginia) | Genius_Genie @ 2885-1884-1637 | VPC > Security Groups > Create security group

Inbound Rules

Type	Protocol	Port Range	Source	Description	Action
SSH	TCP	22	Anywhere	Allow traffic from anywhere	Delete
HTTP	TCP	80	Anywhere	Allow traffic from anywhere	Delete
HTTPS	TCP	443	Anywhere	Allow traffic from anywhere	Delete

Add rule

Outbound Rules

Type	Protocol	Port Range	Destination	Description - optional	Action
All traffic	All	All	Custom	0.0.0.0/0	Delete

Add rule

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.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags

Create security group

Security Group Created

AWS | Search [Alt+S] | United States (N. Virginia) | Genius_Genie @ 2885-1884-1637 | VPC dashboard < | EC2 Global View | Filter by VPC: | Virtual private cloud | Your VPCs | Subnets | Route tables | Internet gateways | Egress-only Internet gateways | Carrier gateways | DHCP option sets | Elastic IPs | Managed prefix lists | NAT gateways | Peering connections | Route servers New | Security | Network ACLs | Security groups | PrivateLink and Lattice | Getting started Updated | Endpoints Updated | Endpoint services | Service networks Updated | Lattice services | Resource configurations New | CloudShell | Feedback | © 2025, Amazon Web Services, Inc. or its affiliates. | Privacy | Terms | Cookie preferences

Security group (sg-0267dc9ba608ad1b5 | Room9-VPC-SG) was created successfully
► Details

sg-0267dc9ba608ad1b5 - Room9-VPC-SG

Actions

Details

Security group name	Room9-VPC-SG	Security group ID	sg-0267dc9ba608ad1b5	Description	Allow traffic from specific ports	VPC ID	vpc-0eacf2424a1a33c78
Owner	288518841637	Inbound rules count	3 Permission entries	Outbound rules count	1 Permission entry		

Inbound rules | **Outbound rules** | **Sharing - new** | **VPC associations - new** | **Tags**

Inbound rules (3)

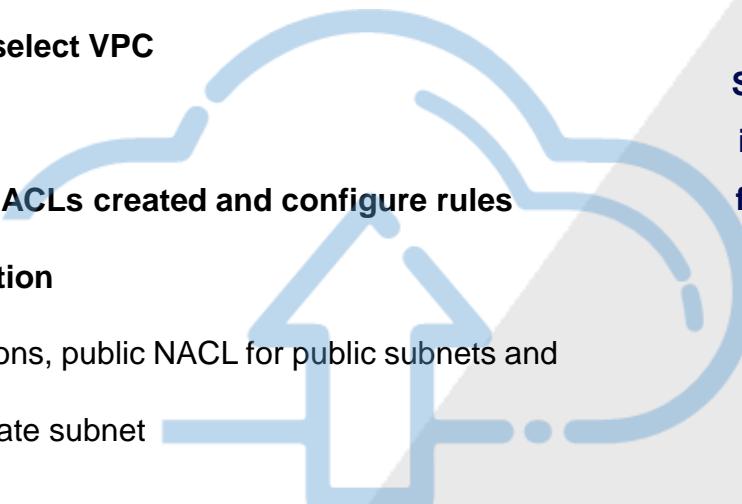
<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-0c5d38b5edf337915	IPv4	HTTPS	TCP	443	0.0.0.0/0	SSH from anywhere
<input type="checkbox"/>	-	sgr-0871fa7a293eb5573	IPv4	HTTP	TCP	80	0.0.0.0/0	Allow traffic from anyw...
<input type="checkbox"/>	-	sgr-02ccff74dd3c1d49c	IPv4	SSH	TCP	22	0.0.0.0/0	Allow traffic from anyw...

Step7: Create and configure NACLs

- ❖ On the left menu, select **Network ACLs**
- ❖ Click “**Create network ACL**” (**Create two, one private and one public**)
- ❖ **Give it a name and select VPC**
- ❖ **Click create**
- ❖ **Select each of the NACLs created and configure rules and subnet association**
- ❖ For subnet associations, public NACL for public subnets and Private NACL for private subnet

NOTES

See next slides for exemplary inbound and outbound rules for public and private NACLs



Step 12: Inbound Rules for Public NACL

VPC > Network ACLs > acl-01bf0186f275aa1fb / Room9-Public-NACL > Edit inbound rules



Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to enter the VPC.

Rule number <small>Info</small>	Type Info	Protocol Info	Port range Info	Destination Info	Allow/Deny Info
100	HTTP (80) ▾	TCP (6) ▾	80	0.0.0.0/0	Allow ▾ <button>Remove</button>
110	HTTPS (443) ▾	TCP (6) ▾	443	0.0.0.0/0	Allow ▾ <button>Remove</button>
120	SSH (22) ▾	TCP (6) ▾	22	0.0.0.0/0	Allow ▾ <button>Remove</button>
*	All traffic ▾	All ▾	All	0.0.0.0/0	Deny ▾

[Add new rule](#) [Sort by rule number](#)

Cancel

[Preview changes](#)

[Save changes](#)

Step 12: Configure Inbound Rules for Private NACL

AWS CloudShell | Search [Alt+S] | United States (N. Virginia) | Genius_Genie @ 2885-1884-1637 | VPC Network ACLs acl-02c3537caeb31c076 / Room9-Private-NACL Edit inbound rules

Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the VPC.

Rule number <small>Info</small>	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Allow/Deny <small>Info</small>
100	SSH (22)	TCP (6)	22	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

[Add new rule](#) [Sort by rule number](#)

[Cancel](#) [Preview changes](#) [Save changes](#)

Step 12: Configure Outbound Rules for Private NACL

Edit outbound rules Info

Outbound rules control the outgoing traffic that's allowed to leave the VPC.

Rule number <small>Info</small>	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Destination <small>Info</small>	Allow/Deny <small>Info</small>	
100	HTTP (80) ▾	TCP (6) ▾	80	0.0.0.0/0	Allow ▾	Remove
110	HTTPS (443) ▾	TCP (6) ▾	443	0.0.0.0/0	Allow ▾	Remove
120	SSH (22) ▾	TCP (6) ▾	22	0.0.0.0/0	Allow ▾	Remove
*	All traffic ▾	All ▾	All	0.0.0.0/0	Deny ▾	

Add new rule

Sort by rule number

Cancel

Preview changes

Save changes

Step8: Set up extra security groups

- ❖ Follow the steps for creating a security to set up two extra security groups. (one for an application load balancer (ALB) and the other for your instances)
- ❖ See next slides for inbound and outbound rules for the ALB security group and EC2 instances security groups.
- ❖ **Note that the source for traffic for the EC2 security group is the ALB security group.**
- ❖ **This is because we want to allow traffic only from the application load balancer**



Step8: Set up security groups (Application Load Balancer)

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

Outbound rules Info

Type [Info](#) Protocol [Info](#) Port range [Info](#) Destination [Info](#) Description - optional [Info](#)

Step8: Set up security groups (EC2 Instances)

aws | X | United States (N. Virginia) ▾ | Genius_Genie @ 2885-1884-1637 ▾ | i ? ! g !

VPC > Security Groups > Create security group

Inbound rules Info

Type	Protocol	Port range	Source	Description - optional	Action
SSH	TCP	22	Custom	sg-070d5ab4b8c4a39c5	Delete
HTTP	TCP	80	Custom	sg-070d5ab4b8c4a39c5	Delete
HTTPS	TCP	443	Custom	sg-070d5ab4b8c4a39c5	Delete

Add rule

Outbound rules Info

Type	Protocol	Port range	Destination	Description - optional	Action
All traffic	All	All	Custom	sg-070d5ab4b8c4a39c5	Delete

Add rule

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.

No tags associated with the resource.

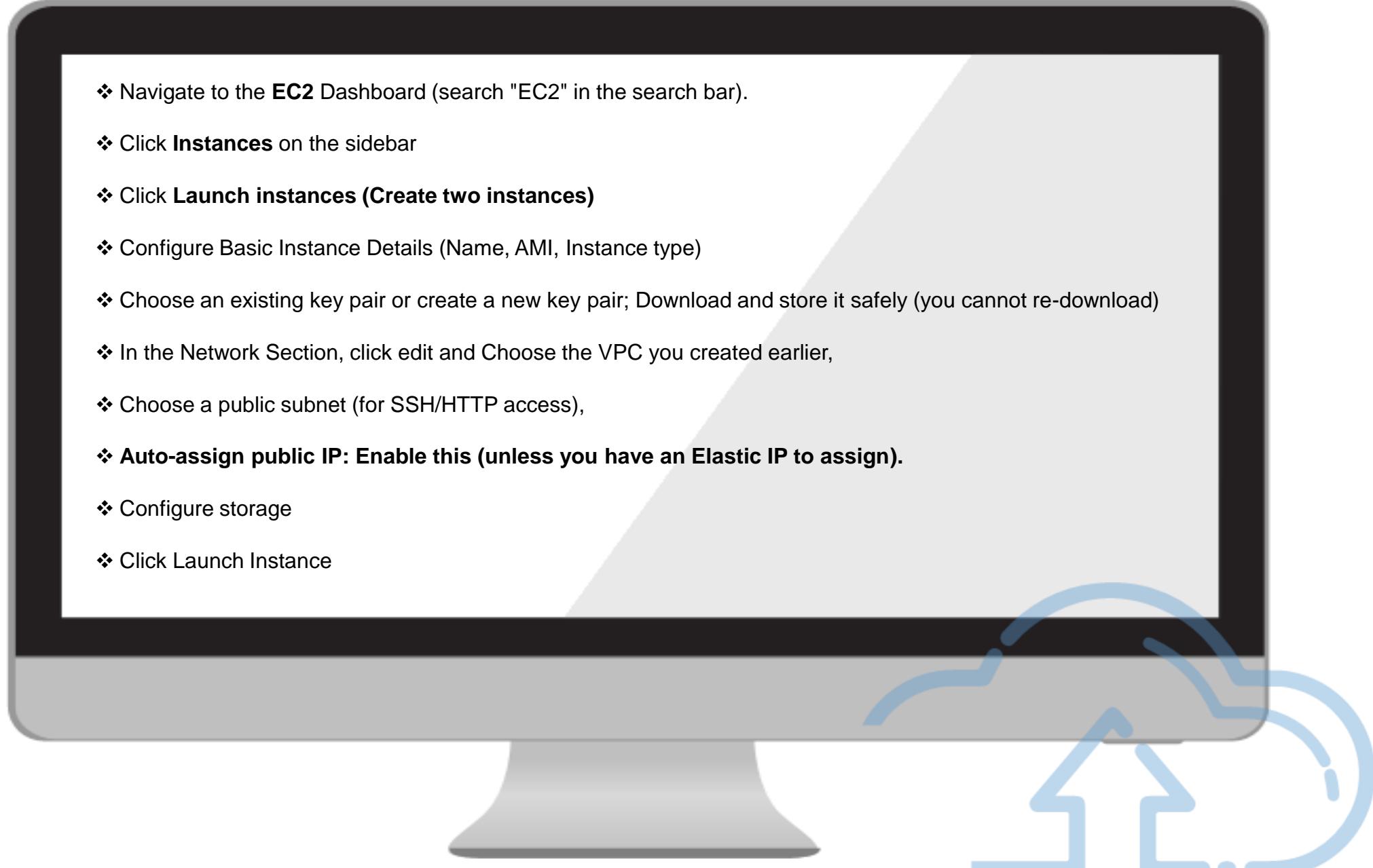
Add new tag

You can add up to 50 more tags

Cancel Create security group

Step9: Launch an EC2 Instance

- ❖ Navigate to the **EC2** Dashboard (search "EC2" in the search bar).
- ❖ Click **Instances** on the sidebar
- ❖ Click **Launch instances (Create two instances)**
- ❖ Configure Basic Instance Details (Name, AMI, Instance type)
- ❖ Choose an existing key pair or create a new key pair; Download and store it safely (you cannot re-download)
- ❖ In the Network Section, click edit and Choose the VPC you created earlier,
- ❖ Choose a public subnet (for SSH/HTTP access),
- ❖ **Auto-assign public IP: Enable this (unless you have an Elastic IP to assign).**
- ❖ Configure storage
- ❖ Click Launch Instance



Step9: Create and launch an instance

aws | [Alt+S] | United States (N. Virginia) | Genius_Genie @ 2885-1884-1637 | [CloudShell](#) | [Feedback](#)

EC2 > Instances > Launch an instance

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
Room9 Web Server [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 Linux  Debian 

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI
ami-09e6f87a47903347c (64-bit (x86), uefi-preferred) / ami-0db36bccb6bf68b98 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs [Free tier eligible](#)

Description
Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.7.20250609.0 x86_64 HVM kernel-6.1

Architecture **Boot mode** **AMI ID** **Publish Date** **Username** [i](#)

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2... [read more](#)
ami-09e6f87a47903347c

Virtual server type (instance type)
t2.micro

Firewall (security group)
Room9-VPC-SG

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. [X](#)

[Cancel](#) [Launch instance](#) [Preview code](#)

Step9: Instance launched

- ✓ To connect to your Instance, click the connect button as shown in this image

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like EC2, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The main content area displays an "Instance summary for i-063a903fbcb79bef8 (Room9 Web Server)". The summary includes details such as Instance ID (i-063a903fbcb79bef8), Public IPv4 address (34.200.239.201), Instance state (Running), and VPC ID (vpc-0eacf2424a1a33c78). At the top right of the summary card, there's a "Connect" button, which is highlighted with a red oval. Below the summary, there are tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under the Details tab, there's a section for Instance details showing AMI ID (ami-09e6f87a47903347c) and AMI name (al2023-ami-2023.7.20250609.0-kernel-6.1-x86_64). The status bar at the bottom indicates the instance was launched on Tuesday, June 24, 2025, at 21:58:52 GMT+0000.

Instance summary for i-063a903fbcb79bef8 (Room9 Web Server)

Updated less than a minute ago

Instance ID: i-063a903fbcb79bef8

Public IPv4 address: 34.200.239.201 | open address

Instance state: Running

Private IP DNS name (IPv4 only): ip-10-0-1-139.ec2.internal

Instance type: t2.micro

VPC ID: vpc-0eacf2424a1a33c78 (Room9-Capstone-VPC)

Subnet ID: subnet-0139ada187f04957d (Room9-Public-Subnet1)

Instance ARN: arn:aws:ec2:us-east-1:288518841637:instance/i-063a903fbcb79bef8

Details Status and alarms Monitoring Security Networking Storage Tags

AMI ID: ami-09e6f87a47903347c

AMI name: al2023-ami-2023.7.20250609.0-kernel-6.1-x86_64

Stop protection: Disabled

Monitoring: disabled

Allowed image: -

Launch time: Tue Jun 24 2025 21:58:52 GMT+0000 (Greenwich Mean Time) (1 minute)

Platform details: Linux/UNIX

Termination protection: Disabled

AMI location: amazon/al2023-ami-2023.7.20250609.0-kernel-6.1-x86_64

CloudShell Feedback

Search [Alt+S]

United States (N. Virginia) Genius_Genie @ 2885-1884-1637

Step9: Instance launched

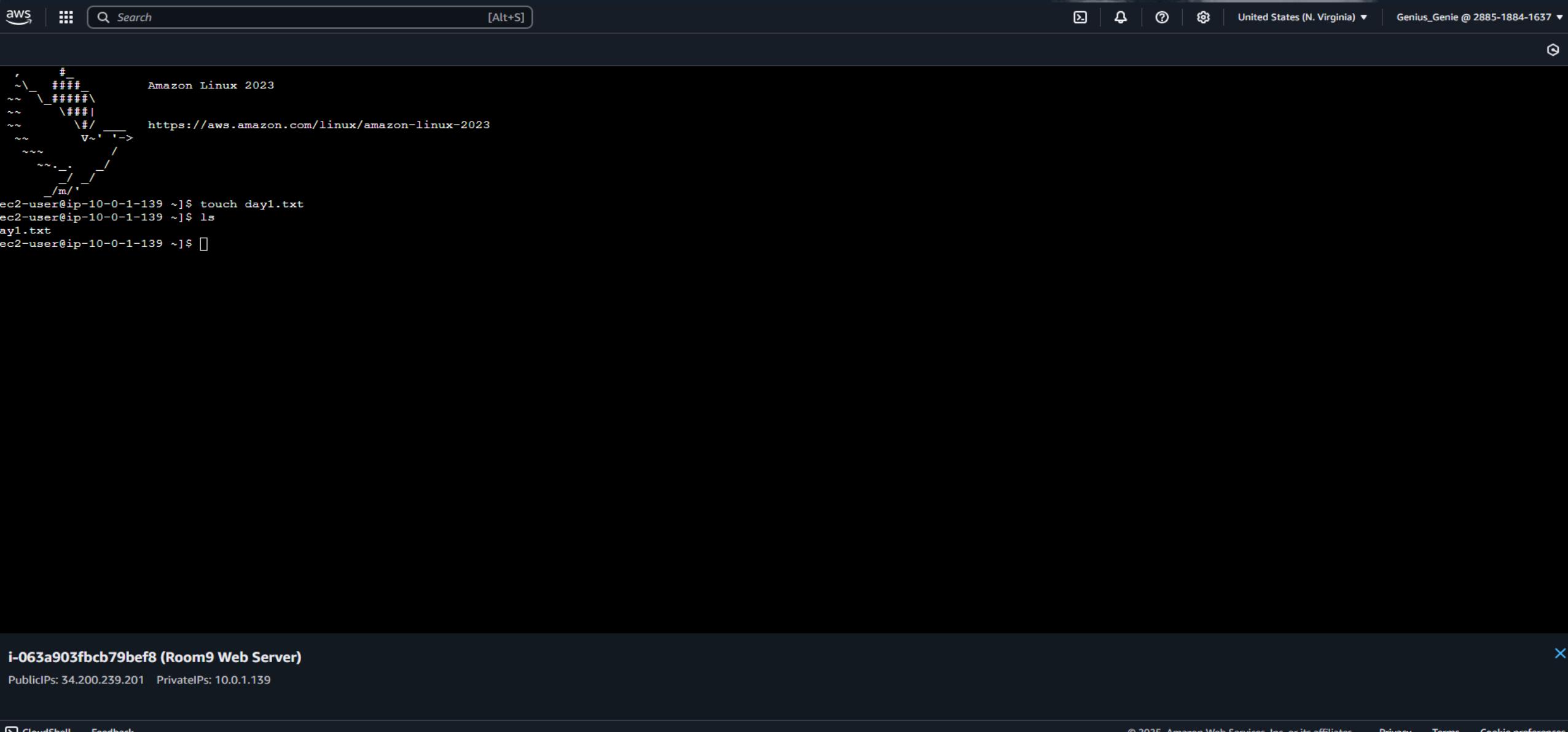
The screenshot shows the AWS EC2 Connect interface for launching an instance. The top navigation bar includes the AWS logo, search bar, and navigation links for EC2, Instances, and the specific instance ID i-09138ac7e1bf1a33a. The sub-navigation bar shows 'Connect to instance'. The main section is titled 'Connect' with a 'Info' link. Below it, a sub-section says 'Connect to an instance using the browser-based client.' There are four tabs: 'EC2 Instance Connect', 'Session Manager', 'SSH client' (which is highlighted with a red oval), and 'EC2 serial console'. The 'Instance ID' is listed as i-09138ac7e1bf1a33a (Room9 Application Server). Under 'Connect using a Public IP', the 'Public IPv4 address' 44.204.215.151 is selected. The 'Username' field contains 'ec2-user'. A note at the bottom states: 'Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' At the bottom right are 'Cancel' and 'Connect' buttons.

NOTES

- ✓ Connect to instance in the browser OR
- ✓ Connect via preferred CLI with the downloaded .pem file created.
- ✓ Copy the command for ssh via the SSH Client tab marked red

Step9: Connected Instance

- ✓ Next, install an application or webserver on the instances created via the CLI you have “ssh’ed” into
- ✓ The commands are provided in the next slide



```
aws | [CloudShell] Search [Alt+S] | [?] | [?] | [?] | United States (N. Virginia) | Genius_Genie @ 2885-1884-1637 | [X]
```

```
~\### Amazon Linux 2023
~~\###\ https://aws.amazon.com/linux/amazon-linux-2023
~~\###\ V~'-->
~~\###\ /m/
ec2-user@ip-10-0-1-139 ~]$ touch day1.txt
ec2-user@ip-10-0-1-139 ~]$ ls
ay1.txt
ec2-user@ip-10-0-1-139 ~]$ [ ]
```

i-063a903fbcb79bef8 (Room9 Web Server)
PublicIPs: 34.200.239.201 PrivateIPs: 10.0.1.139

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

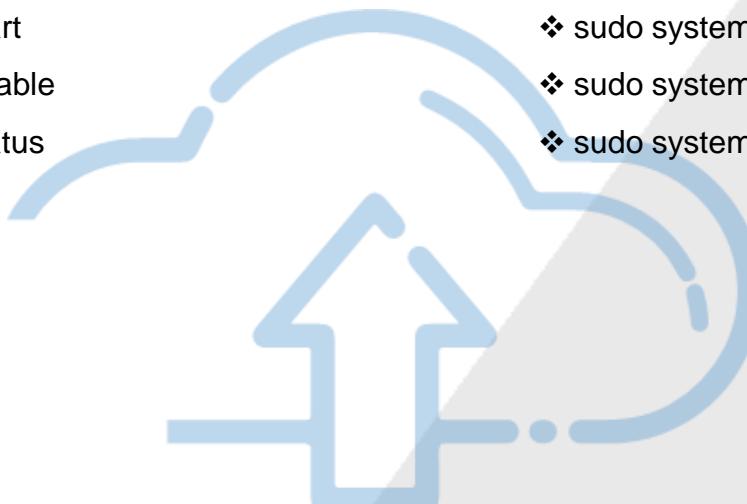
Step10: Install Apache Or Nginx Servers on instances

APACHE COMMANDS

- ❖ sudo yum update
- ❖ sudo yum install httpd -y
- ❖ sudo systemctl start
- ❖ sudo systemctl enable
- ❖ sudo systemctl status

NGINX COMMANDS

- ❖ sudo yum update
- ❖ sudo yum install nginx -y
- ❖ sudo systemctl start
- ❖ sudo systemctl enable
- ❖ sudo systemctl status



Step10: Installed Apache Server

```
Verifying : httpd-filesystem-2.4.62-1.amzn2023.noarch 7/12
Verifying : httpd-tools-2.4.62-1.amzn2023.x86_64 8/12
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 9/12
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 10/12
Verifying : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 11/12
Verifying : mod_lua-2.4.62-1.amzn2023.x86_64 12/12
```

WARNING:

A newer release of "Amazon Linux" is available.

Available Versions:

Version 2023.7.20250623:

Run the following command to upgrade to 2023.7.20250623:

```
dnf upgrade --releasever=2023.7.20250623
```

✓ Install Apache server on instance one.

Release notes:

<https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.7.20250623.html>

Installed:

apr-1.7.5-1.amzn2023.0.4.x86_64	apr-util-1.6.3-1.amzn2023.0.1.x86_64	apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch	httpd-2.4.62-1.amzn2023.x86_64	httpd-core-2.4.62-1.amzn2023.x86_64
httpd-filesystem-2.4.62-1.amzn2023.noarch	httpd-tools-2.4.62-1.amzn2023.x86_64	libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch	mod_http2-2.0.27-1.amzn2023.0.3.x86_64	mod_lua-2.4.62-1.amzn2023.x86_64

Complete!

```
[ec2-user@ip-10-0-1-100 ~]$ sudo systemctl start httpd
[ec2-user@ip-10-0-1-100 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-10-0-1-100 ~]$
```

i-0fa52ea8b5bfa31d4 (Room9-EC2-Instance)

PublicIPs: 100.25.98.17 PrivateIPs: 10.0.1.100

Step10: Installed Nginx on instance 2.



A newer release of "Amazon Linux" is available.

Available Versions:

Version 2023.7.20250623:

Run the following command to upgrade to 2023.7.20250623:

```
dnf upgrade --releasever=2023.7.20250623
```

Release notes:

<https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.7.20250623.html>

Installed:

```
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 libunwind-1.4.0-5.amzn2023.0.2.x86_64
nginx-1:1.26.3-1.amzn2023.0.1.x86_64          nginx-core-1:1.26.3-1.amzn2023.0.1.x86_64    nginx-filesystem-1:1.26.3-1.amzn2023.0.1.noarch
nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
```

Complete!

```
[ec2-user@ip-10-0-3-113 ~]$ sudo systemctl start nginx
[ec2-user@ip-10-0-3-113 ~]$ sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-10-0-3-113 ~]$ █
```



i-0e36aed23e0be0f65 (Room9-EC2-Instance2)

PublicIPs: 44.201.131.109 PrivateIPs: 10.0.3.113

Step11: Create and configure an AMI

- ❖ Select the instance you want to create the AMI from.
- ❖ Fill in the details: Name and description
- ❖ **No reboot:** Leave unchecked (or check it if you don't want the instance rebooted)
- ❖ Click **Create image**
- ❖ Go to **AMIs** under **Images** in the EC2 sidebar
- ❖ Find your newly created AMI

NOTES

An **Amazon Machine Image (AMI)** is a snapshot of an EC2 instance's configuration — including its OS, installed software, settings, and data — that you can **reuse to launch new EC2 instances** with the exact same setup.

Step11: Created and configured AMI

EC2 > AMIs > ami-02f8aa797b4a80c64

EC2

Image summary for ami-02f8aa797b4a80c64

Actions ▾ **Launch instance from AMI**

AMI ID	ami-02f8aa797b4a80c64	Image type	machine	Platform details	Linux/UNIX	Root device type	EBS
AMI name	Room9-Apache-Webserver-AMI	Owner account ID	579111114754	Architecture	x86_64	Usage operation	RunInstances
Root device name	/dev/xvda	Status	Available	Source	579111114754/Room9-Apache-Webserver-AMI	Virtualization type	hvm
Boot mode	uefi-preferred	State reason	-	Creation date	2025-06-25T00:29:26.000Z	Kernel ID	-
Description	Amazon machine image	Product codes	-	RAM disk ID	-	Deprecation time	-
Last launched time	-	Block devices	/dev/xvda=snap-0c046b1a753c4afc0:8:true:g p3	Deregistration protection	Disabled	Allowed image	-
Source AMI ID	ami-09e6f87a47903347c	Source AMI Region	us-east-1				

Permissions **Storage** **Tags**

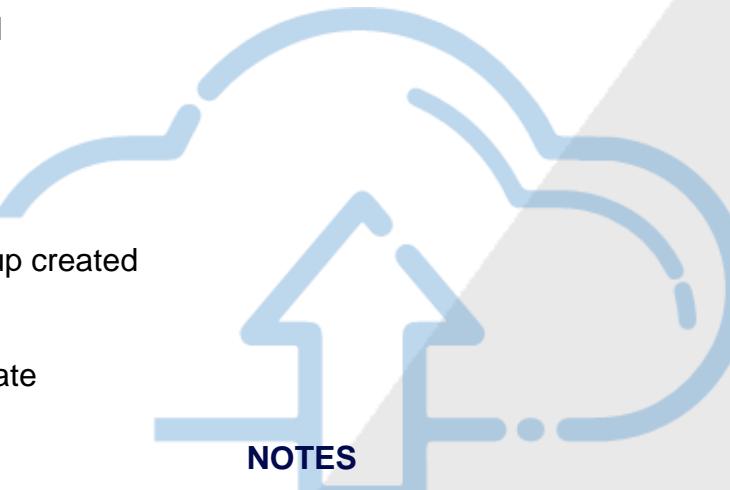
Image share permission
Private
This image is only shared with account IDs, organizations, or OUs that you have specified.

Next, click on Launch Templates, circled red in the EC2 pane

① Restrictions for sharing images publicly are managed using Block public access for AMIs setting under Data protection and security.

Step12: Create and launch template

- ❖ In the sidebar, under **Instances**, click **Launch Templates**
- ❖ Click Create launch template
- ❖ Fill in Launch Template Details (name, version description, source template)
- ❖ Choose the **AMI you created**
- ❖ Choose the instance type
- ❖ Select the key pair
- ❖ Configure Network settings
- ❖ Choose the ALB security group created
- ❖ Configure storage
- ❖ Review settings and click create



A **Launch Template** allows you to define EC2 instance configuration settings **once** and reuse them

Step 12: Create launch templates

EC2 > Launch templates > Create launch template

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

Recents My AMIs Quick Start

Owned by me Shared with me

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Room9-Apache-Webserver-AMI
ami-02f8aa797b4a80c64
2025-06-25T00:29:26.000Z Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred

Description

Amazon machine image

Architecture

x86_64

✓ **NB:** Select the AMI you created for your launch templates

AMI ID

ami-02f8aa797b4a80c64

Summary

Software Image (AMI)
Amazon machine image
ami-02f8aa797b4a80c64

Virtual server type (instance type)

Firewall (security group)

Storage (volumes)
1 volume(s) - 8 GiB

i Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available).

Cancel Create launch template

Step13: Create Target groups for the Application Load Balancer

- ❖ In the sidebar, scroll to **Load Balancing**
- ❖ Click **Target Groups**
- ❖ Choose Target Type (Instances in this case)
- ❖ Configure Basic Settings (Name, Port, VPC)
- ❖ Register targets
- ❖ Click create target groups



Step 13: Create target group for application load balancer

Step 1

Specify group details

Step 2

Register targets

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2)

Filter instances

< 1 >



<input type="checkbox"/> Instance ID	Name	State	Security groups
<input type="checkbox"/> i-0e36aed23e0be0f65	Room9-EC2-Instance2	Running	Room9-ALB-SG
<input type="checkbox"/> i-0fa52ea8b5bfa31d4	Room9-EC2-Instance	Running	Room9-ALB-SG

0 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

Include as pending below

Step 13: Created target for load balancer

EC2 > Target groups > Room9-ApacheTarget

EC2

Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Successfully created the target group: **Room9-ApacheTarget**. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the [Targets](#) tab.

Room9-ApacheTarget

Actions ▾

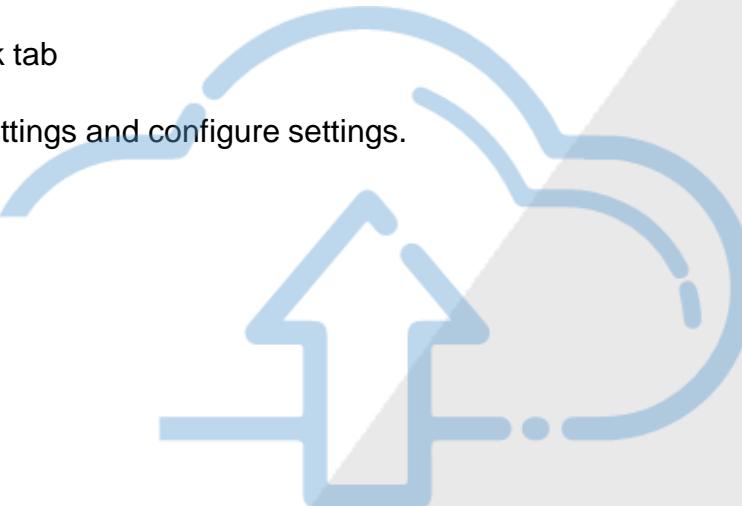
Details

arn:aws:elasticloadbalancing:us-east-1:579111114754:targetgroup/Room9-ApacheTarget/d8062315a87f193a

Target type	Protocol : Port	Protocol version	VPC		
Instance	HTTP: 80	HTTP1	vpc-0ae41a8e94ad10bd2		
IP address type	Load balancer	None associated			
IPv4					
Total targets	0	0	2		
	Healthy	Unhealthy	Unused		
	0 Anomalous				
► Distribution of targets by Availability Zone (AZ)					
Select values in this table to see corresponding filters applied to the Registered targets table below.					

Step14: Configure Health Checks for Target Group

- ❖ Click on the target group to be configured.
- ❖ Navigate to the Health check tab
- ❖ Click on edit health check settings and configure settings.
- ❖ Click on save changes



Step 14: Set up health checks

- ✓ Set up health checks to monitor the status of instances and route traffic only to healthy instances

The screenshot shows the 'Edit health check settings' page for a target group named 'Room9-ApacheTarget'. The page is divided into several sections:

- Health checks**: A general description of how the load balancer tests targets.
- Health check protocol**: Set to 'HTTP'.
- Health check path**: Set to '/'.
- Advanced health check settings**: A collapsed section containing:
 - Health check port**: Set to 'Traffic port' (radio button selected).
 - Healthy threshold**: Set to 2.
 - Unhealthy threshold**: Set to 2.
 - Timeout**: Set to 5 seconds.
 - Interval**: Set to 30 seconds.
 - Success codes**: Set to 200.
- Restore defaults** button (located in the 'Advanced health check settings' section).
- Cancel** and **Save changes** buttons at the bottom right.

Step15: Create the application load balancer

- ❖ On the left menu, under Load Balancing, click Load Balancers
- ❖ Choose Application Load Balancer
- ❖ Configure Load Balancer Basics e.g., Name, Scheme...
- ❖ Select Availability Zones and VPC
- ❖ Configure Security Groups
- ❖ Register Targets (Select target created)
- ❖ Review and Create



Step15: Create your application load balancer

EC2 > Load balancers > Room9-ApacheApp-ALB

EC2 Global View ▾ Events Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations Images AMIs AMI Catalog Elastic Block Store Volumes Snapshots Lifecycle Manager Network & Security Security Groups Elastic IPs Placement Groups Key Pairs

SuccessFully created load balancer: Room9-ApacheApp-ALB 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.

Application Load Balancers now support public IPv4 IP Address Management (IPAM) You can get started with this feature by configuring IP pools in the Network mapping section. Edit IP pools X

Room9-ApacheApp-ALB

C Actions ▾

Details		VPC	Load balancer IP address type
Load balancer type Application	Status Provisioning	vpc-0ae41a8e94ad10bd2 [?]	IPv4
Scheme Internet-facing	Hosted zone Z355XDOTRQ7X7K	Availability Zones subnet-009cf39efec2c2e81 us-east-1b (use1-az2) subnet-0320309a7b7b4d497 us-east-1a (use1-az1)	Date created June 25, 2025, 01:33 (UTC+00:00)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:579111114754:loadbalancer/app/Room9-ApacheApp-ALB/697483660bc1fd0a	DNS name info Room9-ApacheApp-ALB-2023750619.us-east-1.elb.amazonaws.com (A Record)		

Listeners and rules Network mapping Resource map Security Monitoring Integrations Attributes Capacity Tags

Listeners and rules (1) Info Manage rules ▾ Manage listener ▾ Add listener

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step16: Create the auto scaling group

- ❖ In the sidebar of the EC2 Dashboard, under **Auto Scaling**, click **Auto Scaling Groups**
- ❖ Click “**Create Auto Scaling group**”
- ❖ Configure Auto Scaling Group Basics
- ❖ **Choose the launch template created earlier**
- ❖ **Choose your VPC and two subnets for high availability**
- ❖ **Attach to the ALB and select the Target group created earlier**
- ❖ Configure Health Checks (Set **grace period** (e.g., 300 seconds) to allow instances time to boot)
- ❖ Configure Group Size and Scaling Policies; e.g. **Desired capacity (2), min capacity (1), max capacity (4)**
- ❖ Add Scaling Policies; **Target tracking policy** (recommended) e.g. Example: **CPU Utilization at 60%**
- ❖ Add Notifications (to receive alerts on scaling performance)
- ❖ **Review and create**

Step 16: Create your auto scaling group

- ✓ Subscribe to a notification service to receive updates on your scaling

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template

Step 2
Choose instance launch options

Step 3 - optional
Integrate with other services

Step 4 - optional
Configure group size and scaling

Step 5 - optional
Add notifications

Step 6 - optional
Add tags

Step 7
Review

Add notifications - optional Info

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

Notification 1 Remove

Send a notification to
Room9-SNS-Topic

With these recipients
[REDACTED]@gmail.com

Use existing topic

Event types
Notify subscribers whenever instances

Launch
 Terminate
 Fail to launch
 Fail to terminate

Add notification

Step 16: Created auto scaling group

EC2 > Auto Scaling groups > Room9-AutoScaling

Room9-AutoScaling

Room9-AutoScaling Capacity overview

arn:aws:autoscaling:us-east-1:579111114754:autoScalingGroup:8a9ec2d1-fd2b-42c9-aa71-a7cc27bb5e1e:autoScalingGroupName/Room9-AutoScaling

Desired capacity: 1 **Scaling limits (Min - Max)**: 1 - 4 **Desired capacity type**: Units (number of instances) **Status**: -

Date created: Wed Jun 25 2025 09:11:44 GMT+0000 (Greenwich Mean Time)

Edit

Details | **Integrations - new** | **Automatic scaling** | **Instance management** | **Instance refresh** | **Activity** | **Monitoring**

Launch template

Launch template: lt-0d848ace977036cb7 (Room9-ApacheApp-LaunchTemplate)

AMI ID: ami-02f8aa797b4a80c64

Instance type: t2.micro

Owner: arn:aws:iam::579111114754:root

Version: Default

Security groups: -

Security group IDs: sg-0225167ba30447267

Create time: Wed Jun 25 2025 00:49:08 GMT+0000 (Greenwich Mean Time)

Description: Launch template for Apache Auto Scaling Group

Storage (volumes): -

Key pair name: Room9

Request Spot Instances: No

Edit

View details in the launch template console

Network

Availability Zones: us-east-1a (us-east-1b), us-east-1a (us-east-1a)

Subnet ID: subnet-009cf39efec2c2e81, subnet-0320309a7b7b4d497

Availability Zone distribution: Balanced best effort

Edit

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 21: Confirmed notification settings



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

arn:aws:sns:us-east-1:57911114754:Room9-SNS-Topic:03dd79c9-4b3b-431f-868e-6324db05284d

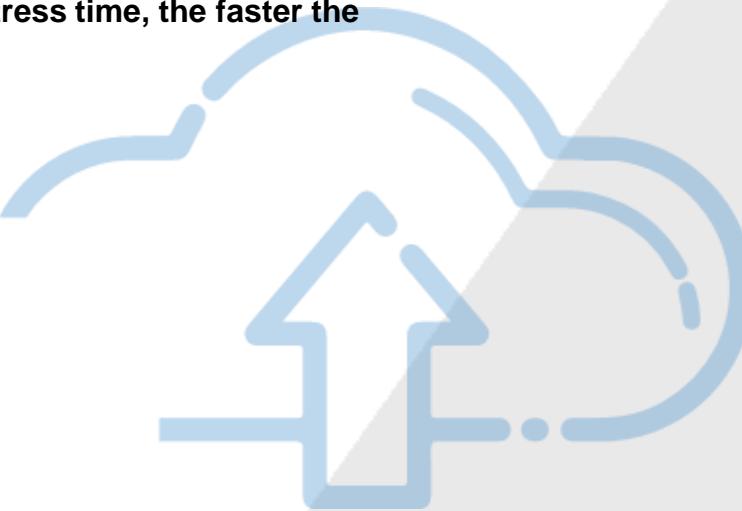
If it was not your intention to subscribe, [click here to unsubscribe](#).

Step17: Monitoring, Testing and Optimization

Simulate CPU overload with the stress command

- ✓ `sudo yum install -y stress`
- ✓ `stress --cpu 2 --timeout 60`

NB: The longer the stress time, the faster the results



Step 17: Monitoring, Testing And Optimization

- ✓ Simulate CPU-Intensive workload with the stress command

```
'~\_\#\#\#_          Amazon Linux 2023
~~ \_\#\#\#\#\_
~~ \#\#\#
~~ \#/   __  https://aws.amazon.com/linux/amazon-linux-2023
~~ V~' '-->
~~~ /
~~ .-./_/
~/m/'

Last login: Wed Jun 25 09:43:49 2025 from 18.206.107.28
[ec2-user@ip-10-0-3-113 ~]$ stress --cpu 2 --timeout 60
stress: info: [43762] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [43762] successful run completed in 60s
[ec2-user@ip-10-0-3-113 ~]$ stress --cpu 2 --timeout 60
stress: info: [43822] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [43822] successful run completed in 60s
[ec2-user@ip-10-0-3-113 ~]$ stress --cpu 2 --timeout 120
stress: info: [43886] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [43886] successful run completed in 120s
[ec2-user@ip-10-0-3-113 ~]$ █
```

i-0e36aed23e0be0f65 (Room9-EC2-Instance2)

Public IPs: 44.201.131.109 Private IPs: 10.0.3.113



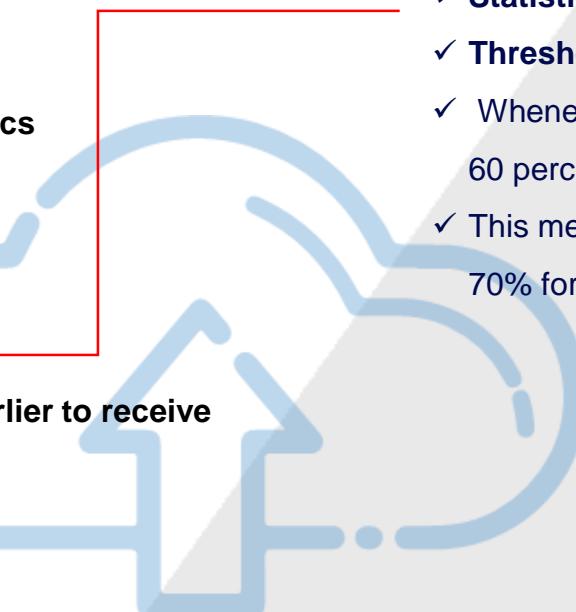
Step17: Monitoring, Testing and Optimization

SET UP CLOUD WATCH ALARMS

- ❖ In the search bar, type and open “**CloudWatch**”
- ❖ In the left sidebar, click “**Alarms**”
- ❖ **Click create alarm**
- ❖ Click “**Select metric**”
- ❖ Choose **EC2 → Per-Instance Metrics**
- ❖ Select your **Instance ID**
- ❖ **Check the CPUUtilization box**
- ❖ **Click Select metric**
- ❖ Configure the Alarm
- ❖ **Choose the SNS topic created earlier to receive email alerts**
- ❖ **Name and create the alarm**

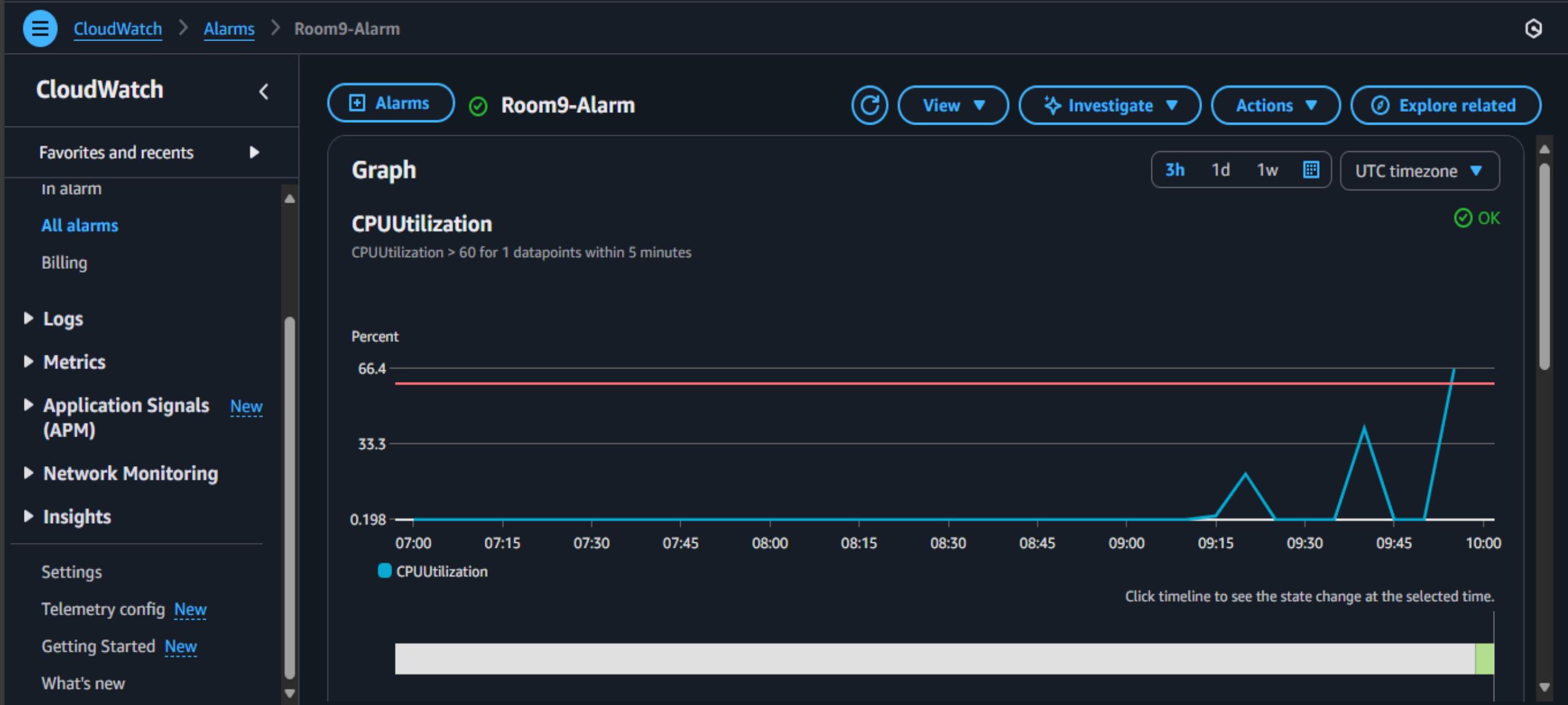
ALARM CONFIGURATION

- ✓ **Period:** 1 minute (for near real-time)
- ✓ **Statistic:** Average
- ✓ **Threshold type:** Static
- ✓ Whenever CPU utilization is greater than 60 percent for 1 out of 1 data points
- ✓ This means the alarm will trigger if CPU > 70% for one minute



Step17: Testing And Optimization

- ✓ Real time monitoring in cloud watch



Step17: Testing And Optimization

- ✓ Traffic simulation triggered auto-scaling events, resulting in new instances being launched and later terminated in line with defined scaling policies.

EC2 > Auto Scaling groups > Room9-AutoScaling			
Snapshots	Successful	instance: i-0a098cc35563060cc	At 2025-06-25T10:02:18Z an instance was launched in response to an unhealthy instance needing to be replaced.
Lifecycle Manager	In progress	Terminating EC2 instance: i-0c677f2cc1dae217f	At 2025-06-25T10:02:18Z an instance was taken out of service in response to an ELB system health check failure.
Network & Security	Successful	Launching a new EC2 instance: i-0c677f2cc1dae217f	At 2025-06-25T10:00:13Z an instance was launched in response to an unhealthy instance needing to be replaced.
Security Groups	Successful	Terminating EC2 instance: i-0f0939bbdc624104f	At 2025-06-25T10:00:13Z an instance was taken out of service in response to an ELB system health check failure.
Elastic IPs	Successful	Launching a new EC2 instance: i-0f0939bbdc624104f	At 2025-06-25T09:58:06Z an instance was launched in response to an unhealthy instance needing to be replaced.
Placement Groups	Successful	Terminating EC2 instance: i-0a6e281f54adc4de9	At 2025-06-25T09:58:06Z an instance was taken out of service in response to an ELB system health check failure.
Key Pairs			
Network Interfaces			
Load Balancing			
Load Balancers			
Target Groups			
Trust Stores			
Auto Scaling			
Auto Scaling Groups			
Settings			

Step17: Testing And Optimization

- ✓ Email Notification : Additionally, the integrated SNS configuration successfully sent email notifications, confirming each event in real time

 AWS Notifications <no-reply@sns.amazonaws.com> 10:08 AM (1 minute ago)   
to me ▾

Service: AWS Auto Scaling
Time: 2025-06-25T10:08:21.070Z
RequestId: 23665e9a-2735-16c4-fc26-9efe0ab53028
Event: autoscaling:EC2_INSTANCE_TERMINATE
AccountId: 579111114754
AutoScalingGroupName: Room9-AutoScaling
AutoScalingGroupARN: arn:aws:autoscaling:us-east-1:579111114754:autoScalingGroup:8a9ec2d1-fd2b-42c9-aa71-a7cc27bb5e1e:autoScalingGroupName/Room9-AutoScaling
ActivityId: 23665e9a-2735-16c4-fc26-9efe0ab53028
Description: Terminating EC2 instance: i-0c677f2cc1dae217f
Cause: At 2025-06-25T10:02:18Z an instance was taken out of service in response to an ELB system health check failure.
StartTime: 2025-06-25T10:02:18.309Z
EndTime: 2025-06-25T10:08:21.070Z
StatusCode: InProgress
StatusMessage:
Progress: 50
EC2InstanceId: i-0c677f2cc1dae217f

Step18: Create and configure DynamoDB

- ❖ Search for and open **DynamoDB** in the search bar
- ❖ In the sidebar, select Tables and click **Create table**
- ❖ Set up table details: **name, partition key,...**
- ❖ In most cases, leave the table settings as default
- ❖ Click **Create table**

NOTES

- ❖ **Partition Key:** A required attribute that determines how data is distributed and uniquely identifies each item if no sort key is used
- ❖ **Sort Key:** An optional attribute that, combined with the partition key, allows multiple items with the same partition key to be uniquely identified and sorted.

Step18: Create and configure DynamoDB

Create table

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

Students

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.)

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

StudentID

String



1 to 255 characters and case sensitive.

Sort key - optional

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Enter the sort key name

String



1 to 255 characters and case sensitive.

Table settings

Default settings

The fastest way to create your table. You can modify most of these settings after your table has been created. To modify these settings now, choose 'Customize settings'.

Customize settings

Use these advanced features to make DynamoDB work better for your needs.

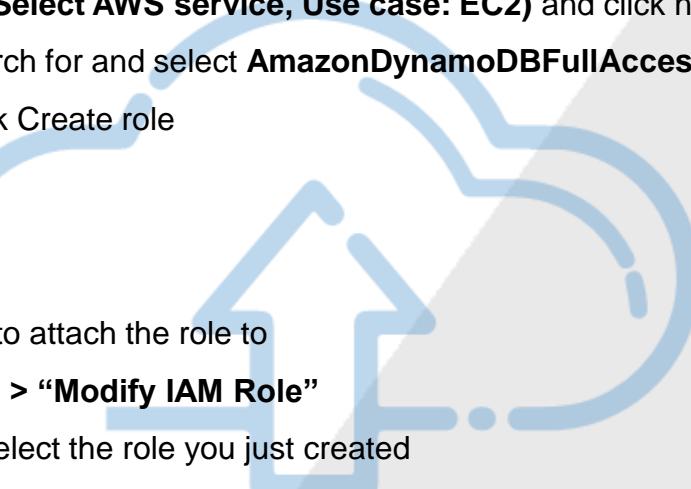
Step19: Create an IAM role and attach to instance

To create the IAM Role

- ❖ Go to the **IAM Dashboard**
- ❖ In the sidebar, select **Roles** and click on **Create role**
- ❖ Choose Trusted Entity Type (**Select AWS service, Use case: EC2**) and click next
- ❖ Attach permission policy; search for and select **AmazonDynamoDBFullAccess** and click next
- ❖ Give the role a name and click **Create role**

To attach to the instance:

- ❖ Go to EC2 Dashboard
- ❖ Select the instance you want to attach the role to
- ❖ Click “**Actions**” > “**Security**” > “**Modify IAM Role**”
- ❖ In the **IAM role dropdown**, select the role you just created
- ❖ Click “**Update IAM role**”



Step19: Create IAM Role

✓ Grant a full DynamoDB access policy for the role

✓ Next, attach the IAM role to an EC2 instance

The screenshot shows the 'Create role' wizard in the AWS IAM console. The navigation bar at the top indicates the path: IAM > Roles > Create role. The left sidebar shows the steps: Step 1 (Select trusted entity), Step 2 (Add permissions), Step 3 (Name, review, and create), and the current step, Step 3 (highlighted in blue). The main area is titled 'Name, review, and create' and contains 'Role details'. The 'Role name' field is filled with 'Room9DynamoEC2'. Below it, a note says 'Maximum 64 characters. Use alphanumeric and '+,-,@,_' characters.' The 'Description' field contains the text 'Allows EC2 instances to call AWS services (DynamoDB) on your behalf.' Below it, another note says 'Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/[\{\}]#\$%^*()~`-''. At the bottom of this section is a button labeled 'Edit'. The next section, 'Step 1: Select trusted entities', shows a 'Trust policy' editor with a JSON code block:

```
1: {  
2:   "Version": "2012-10-17",  
3:   "Statement": [  
4:     {  
5:       "Effect": "Allow",  
6:       "Action": [  
7:         "sts:AssumeRole"  
8:       ],  
9:       "Principal": [],  
10:      "Service": [  
11:        "ec2.amazonaws.com"  
12:      ]  
13:    }  
14:  ]  
15: }  
16: }
```

Below this is another 'Edit' button. The final section, 'Step 2: Add permissions', is partially visible at the bottom.

Step19: Attach IAM Role to your EC2 instance

- ✓ Attach the IAM role to your EC2 instance to grant access to your DB.

≡ EC2 > Instances > [i-0b6d89dc0af729a2c](#) > Modify IAM role



Modify IAM role Info

Attach an IAM role to your instance.

Instance ID

i-0b6d89dc0af729a2c (Room9-EC2-Dynamo)

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

Room9DynamoEC2



Create new IAM role

Cancel

Update IAM role

- ✓ Next, ssh into your instance to create tables

Step19: Create tables for your DB

To create tables

- ❖ Verify DynamoDB access role by running a simple command. e.g: **aws dynamodb list-tables**
- ❖ Ensure that python3 is installed on your instance by running **python3 --version or python --version**
- ❖ Install **Boto3 (AWS SDK for Python)**
- ❖ Create a python script by running **nano student_logger.py**
- ❖ Type your code to create table
- ❖ Save and exit file
- ❖ Next, run the python script with the command **python3 student_logger.py**
- ❖ Go to your DynamoDB table in AWS Console

Installing python and boto3

- ✓ Install python: **sudo yum install python3 -y**
- ✓ Check if pip is available: **pip3 --version**
- ✓ If not available , install pip: **sudo yum install python3-pip -y**
- ✓ install boto3 using pip: **pip3 install boto3**
- ✓ Verify installation by running : **python3**
- ✓ in the python shell run:
- ✓ **import boto3**
- ✓ **print("Boto3 is working!")**
- ✓ **exit()**

Step19: Create tables for your DB

- ✓ Set up Boto3 (the AWS SDK for Python) on the EC2 instance to enable seamless interaction with AWS services through Python scripts.

The screenshot shows a terminal session in an AWS CloudShell window. The terminal output is as follows:

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Wed Jun 25 11:08:20 2025 from 129.224.201.79
[ec2-user@ip-10-0-3-203 ~]$ sudo yum update
Amazon Linux 2023 Kernel Livepatch repository
Last metadata expiration check: 0:00:01 ago on Wed Jun 25 11:12:37 2025.

WARNING:
A newer release of "Amazon Linux" is available.

Available Versions:
Version 2023.7.20250623:
Run the following command to upgrade to 2023.7.20250623:
dnf upgrade --releasever=2023.7.20250623

Release notes:
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.7.20250623.html

Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-3-203 ~]$ sudo yum install python3 -y
Last metadata expiration check: 0:01:58 ago on Wed Jun 25 11:12:37 2025.
Package python3-3.9.22-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-3-203 ~]$ python3 --version
Python 3.9.22
[ec2-user@ip-10-0-3-203 ~]$
```

At the bottom of the terminal, the identifier "i-0502ca591120bdbff (Room9-Dynamo-EC2)" and network information "Public IPs: 54.152.23.128 Private IPs: 10.0.3.203" are displayed. The AWS CloudShell interface includes standard navigation and search bars at the top.

Step19: Create tables for your DB

- ✓ Use a python script to create a table in your DB

```
import boto3

# Create a DynamoDB resource using the us-east-1 region
dynamodb= boto3.resource('dynamodb', region_name='us-east-1')

# Reference your DynamoDB table
table= dynamodb.Table('Students')

# List of students to add
students = [
    {'StudentID': '001', 'name': 'Harrison Adom', 'course': 'AWS Cloud Fundamentals'},
    {'StudentID': '002', 'name': 'Fadilatu Suhuyini Mahamud', 'course': 'Linux Basics'},
    {'StudentID': '003', 'name': 'Richmond Nyarko', 'course': 'AMZ CloudFront'},
    {'StudentID': '004', 'name': 'Genevieve Kankam Mensah', 'course': 'Quantum Engineering'},
    {'StudentID': '005', 'name': 'Clifford Mills', 'course': 'Statistics'},]

# Loop through and insert each student
for student in students:
```

File ^K Cut ^T Execute 0 line ^C Location M-U Undo M-A Set Mark M-] T
Edit ^R Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^B Wh

i-0502ca591120bdbff (Room9-Dynamo-EC2)

Public IPs: 54.152.23.128 Private IPs: 10.0.3.203



Step19: Created tables

```
[ec2-user@ip-10-0-3-203 ~]$ ^C
[ec2-user@ip-10-0-3-203 ~]$ nano student_logger.py
[ec2-user@ip-10-0-3-203 ~]$ > student_logger.py
[ec2-user@ip-10-0-3-203 ~]$ nano student_logger.py
[ec2-user@ip-10-0-3-203 ~]$ > student_logger.py
[ec2-user@ip-10-0-3-203 ~]$ nano student_logger.py
[ec2-user@ip-10-0-3-203 ~]$ python3 student_logger.py
Inserted: Harrison Adom | Status: 200
Inserted: Fadilatu Suhuyini Mahamud | Status: 200
Inserted: Richmond Nyarko | Status: 200
Inserted: Genevieve Kankam Mensah | Status: 200
Inserted: Clifford Mills | Status: 200
[ec2-user@ip-10-0-3-203 ~]$ nano student_logger.py
[ec2-user@ip-10-0-3-203 ~]$ python3 student_logger.py
Inserted: Harrison Adom | Status: 200
Inserted: Fadilatu Suhuyini Mahamud | Status: 200
Inserted: Richmond Nyarko | Status: 200
Inserted: Genevieve Kankam Mensah | Status: 200
Inserted: Clifford Mills | Status: 200
[ec2-user@ip-10-0-3-203 ~]$
```

i-0502ca591120bdbff (Room9-Dynamo-EC2)

Public IPs: 54.152.23.128 Private IPs: 10.0.3.203



Step20: View Tables in DynamoDB

The screenshot shows the AWS DynamoDB console interface. On the left, there's a navigation sidebar with 'DynamoDB' selected. The main area is titled 'Students' and contains a 'Scan or query items' section. Under 'Select a table or index', 'Table - Students' is chosen. In the 'Select attribute projection' dropdown, 'Specific attributes' is selected. The 'Specific attributes to project' field contains 'StudentID'. Below this, there's a 'Filters - optional' section with 'Run' and 'Reset' buttons. A green status bar at the bottom indicates a successful scan: 'Completed - Items returned: 5 - Items scanned: 5 - Efficiency: 100% - RCU consumed: 2'. The main table area is titled 'Table: Students - Items returned (5)' and shows the following data:

StudentID	course	name
001	AWS Cloud ...	Harrison Adom
003	AMZ Cloud...	Richmond Nyarko
002	Linux Basics	Fadilatu Suhuyini Mahamud
004	Quantum E...	Genevieve Kankam Mensah
005	Statistics	Clifford Mills

On the far right of the table, there are 'Actions' and 'Create item' buttons, with the 'Create item' button circled in red.

- ✓ Use the create item button(circled red) to add more item via the console

Step21: Add more records via the AWS Console

- ✓ Manually add new records to the DynamoDB ‘students’ table using the AWS Console by entering attribute names and values through the Items view.

The screenshot shows the AWS DynamoDB 'Create item' interface. At the top, there are navigation links: 'DynamoDB' > 'Explore items: Students' > 'Create item'. To the right are 'Form' and 'JSON view' buttons. Below this, the title 'Create item' is displayed, followed by a sub-instruction: 'You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep.' A 'Learn more' link is also present. The main area is titled 'Attributes' and contains three entries:

Attribute name	Value	Type	Action
StudentID - Partition key	007	String	
course	Law	String	Remove
name	Florentia Teye	String	Remove

At the bottom right are 'Cancel' and 'Create item' buttons.

Added records via the AWS Console

- ✓ NB: Florentia, Ajara, and Esther have been added to the students table.

DynamoDB > Explore items > Students

Table: Students - Items returned (8)

Scan started on June 25, 2025, 12:59:51

	StudentID (String)	AWS Cloud ...	Harrison Adom
<input type="checkbox"/>	003		
<input type="checkbox"/>	001	Quantum E...	Genevieve Kankam Mensah
<input type="checkbox"/>	004	Law	Florentia Teye
<input type="checkbox"/>	007	Linux Basics	Fadilatu Suhuyini Mahamud
<input type="checkbox"/>	002	Biomedical ...	Esther Awudu
<input type="checkbox"/>	008	Statistics	Clifford Mills
<input type="checkbox"/>	005	Medicine	Ajara Amadu
<input type="checkbox"/>	006		

Actions

Items 1 / 8

Navigation ▲ ▼

Actions <input type="button" value="Duplicate item

CONCLUSION

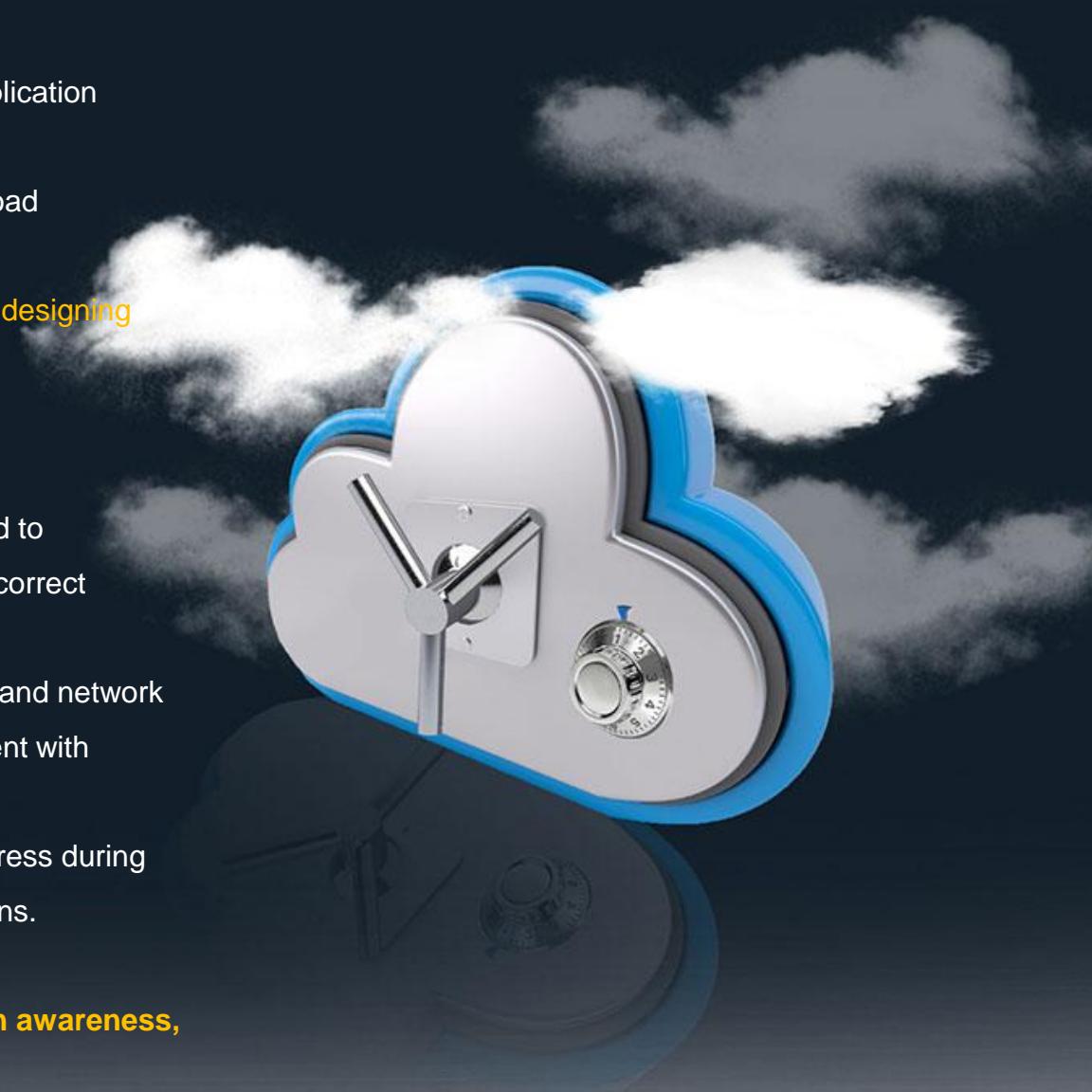
Project summary

- ✓ Architected and deployed a secure, scalable, and highly available web application infrastructure on AWS.
- ✓ Leveraged services such as EC2, VPC, DynamoDB, IAM, Auto Scaling, Load Balancing, and CloudWatch.
- ✓ Simulate a production-grade environment and gain practical experience in **designing** and securing cloud-native systems.

Challenges

- ✓ A region mismatch between the EC2 instance and the DynamoDB table led to connection failures, which was resolved by recreating the database in the correct region.
- ✓ Additionally, misconfigured inbound and outbound rules in security groups and network ACLs initially blocked essential traffic, requiring careful review and alignment with application needs.
- ✓ CloudWatch alarms failed to trigger as expected due to insufficient CPU stress during testing, highlighting the need for properly calibrated performance simulations.

These issues reinforced the importance of precise configuration, region awareness, and effective monitoring in cloud infrastructure design.





DESIGNING AND MAINTAINING A SCALABLE, WEB APPLICATION IN AWS