

Lab 4: Configure High Availability in Your Amazon VPC

© 2025 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Lab overview

Amazon Web Services (AWS) provides services and infrastructure to build reliable, fault-tolerant, and highly available systems in the cloud. Fault tolerance is a system's ability to remain in operation even if some of the components used to build the system fail. High availability is not about preventing system failure but the ability of the system to recover quickly from it. As an AWS solutions architect, it is important to design your systems to be highly available and fault tolerant when needed. You must also understand the benefits and costs of those designs. In this lab, you integrate two powerful AWS services: Elastic Load Balancing and Auto Scaling groups. You create an Auto Scaling group of Amazon Elastic Compute Cloud (Amazon EC2) instances operating as application servers. You then configure an Application Load Balancer to load balance between the instances inside that Auto Scaling group. You continue to work with the Amazon Relational Database Service (Amazon RDS) by permitting Multi-AZ, creating a read replica, and promoting a read replica. With read replicas, you can write to the primary database and read from the read replica. Because a read replica can be promoted to be the primary database, it is a useful tool in high availability and disaster recovery.

The following image shows the final architecture:

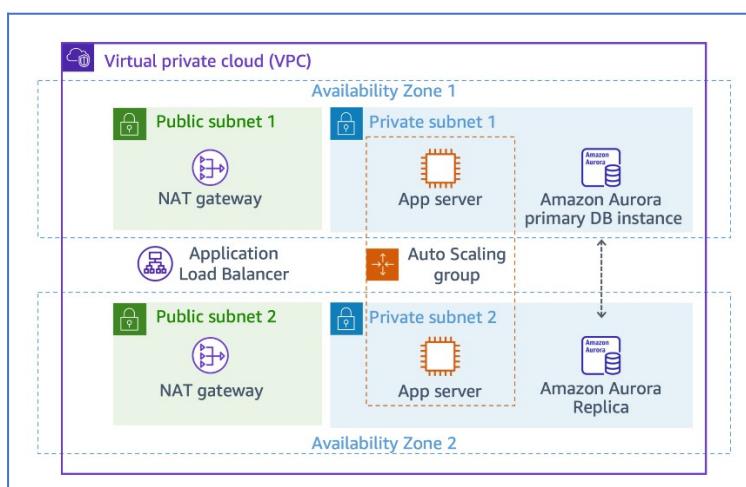


Image description: The preceding image depicts the main focus of this lab. This lab focuses on designing highly available and fault-tolerant systems in AWS by integrating Elastic Load Balancing with Auto Scaling groups for EC2 instances, while

also exploring Amazon RDS features such as Multi-AZ deployment, read replicas, and promotion capabilities to enhance database reliability and performance.

Objectives

After completing this lab, you should be able to do the following:

- Create an Amazon EC2 Auto Scaling group and register it with an Application Load Balancer spanning across multiple Availability Zones.
- Create a highly available Amazon Aurora database (DB) cluster.
- Modify an Aurora DB cluster to be highly available.
- Modify an Amazon Virtual Private Cloud (Amazon VPC) configuration to be highly available using redundant NAT gateways.
- Confirmed your database can perform a failover to a read replica instance.

Task 1: Inspect your existing lab environment

Review the configuration of the existing environment. The following resources have been provisioned for you through AWS CloudFormation:

- An Amazon VPC
- Public and private subnets in two Availability Zones
- An internet gateway (not shown in the diagram) associated with the public subnets
- A NAT gateway in one of the public subnets
- An Application Load Balancer deployed across the two public subnets to receive and forward incoming application traffic
- An EC2 instance in one of the private subnets, running a basic inventory tracking application
- An Aurora DB cluster containing a single DB instance in one of the private subnets to store inventory data

The following image shows the initial architecture:

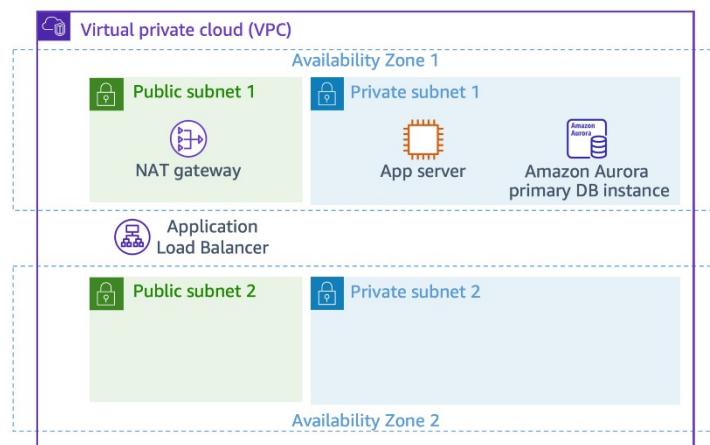


Image description: Preceding image depicts the initial architecture; it consists of a VPC to facilitate networking with Public and Private subnets. This initial architecture also depicts an Application Load Balancer (ALB) to be able to distribute the incoming traffic to multiple EC2 instances. In-order to tighten the security posture we have an EC2 instance (App server) and an instance of Amazon Aurora (relational database) provisioned in the Private subnet. To further enhance high-availability we will improvise this architecture with Auto-scaling groups and multi-AZ database cluster.

Task 1.1: Examine the network infrastructure

In this task, you review the network configuration details for the lab environment.

- At the top of the AWS Management Console, in the search bar, search for and choose **VPC**.

The screenshot shows the AWS VPC Dashboard. The left sidebar has sections for 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), Security (Network ACLs), and a New button. The main area displays 'Resources by Region' for the United States region. It shows counts for VPCs (2), Subnets (8), Route Tables (4), Internet Gateways (1), NAT Gateways (1), VPC Peering Connections (0), Network ACLs (2), Security Groups (6), Customer Gateways (0), and Egress-only Internet Gateways (0). Buttons for 'Create VPC' and 'Launch EC2 Instances' are at the top. A 'Service Health' box shows no issues. A 'Settings' box includes options for Block Public Access, Zones, and Console Experiments. An 'Additional Information' box links to VPC Documentation, All VPC Resources, Forums, and Report an Issue. An 'AWS Network Manager' box provides information about the service.

Note: The Lab VPC was created for you by the lab environment, and all of the application resources used by this lab exercise exist inside this VPC.

- In the left navigation pane, choose **Your VPCs**.

Your Lab VPC appears on the list along with the default VPC.

The screenshot shows the AWS VPC dashboard with the 'Your VPCs' section selected. It displays two VPCs: 'Lab VPC' (vpc-096d40eaacafdf19a4) and another unnamed VPC (vpc-096ebcf6e9ad464e9). The 'Lab VPC' details page is open, showing its configuration. Key details include:

VPC ID	Name	State	Block Public Access	DNS hostnames
vpc-096d40eaacafdf19a4	Lab VPC	Available	Off	Enabled
vpc-096ebcf6e9ad464e9	-	Available	Off	-

Details

VPC ID	Name	State	Block Public Access	DNS hostnames
vpc-096d40eaacafdf19a4	Lab VPC	Available	Off	Enabled
vpc-096ebcf6e9ad464e9	-	Available	Off	-

IPv4 CIDR (Network border group)

VPC ID	Name	State	Block Public Access	DNS hostnames
vpc-096d40eaacafdf19a4	Lab VPC	Available	Off	Enabled
vpc-096ebcf6e9ad464e9	-	Available	Off	-

- In the left navigation pane, choose **Subnets**.

The subnets that are part of the Lab VPC are displayed in a list. Examine the following details listed in the columns for Public Subnet 1:

- In the VPC column, you can identify which VPC this subnet is associated with. This subnet exists inside the Lab VPC.
- In the IPv4 Classless Inter-Domain Routing (CIDR) column, the value of 10.0.0.0/24 means this subnet includes the 256 IPs (five of which are reserved and unusable) between 10.0.0.0 and 10.0.0.255.
- In the Availability Zone column, you can identify the Availability Zone in which this subnet resides. This subnet resides in the Availability Zone ending with an “a”.

The screenshot shows the AWS VPC dashboard with the 'Subnets' section selected. It displays eight subnets under the 'Lab VPC'. The subnets are:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	Availability Zone
Public Subnet 2	subnet-0e04f6b8017c1b988	Available	vpc-096d40eaacafdf19a4 Lab VPC	Off	10.0.1.0/24	us-west-2b
Public Subnet 1	subnet-02365a081971d24c9	Available	vpc-096d40eaacafdf19a4 Lab VPC	Off	10.0.0.0/24	us-west-2a
Private Subnet 2	subnet-076b1cb82d312b51e	Available	vpc-096d40eaacafdf19a4 Lab VPC	Off	10.0.4.0/23	us-west-2b
Private Subnet 1	subnet-02ad7a020511da5c7	Available	vpc-096d40eaacafdf19a4 Lab VPC	Off	10.0.2.0/23	us-west-2a
-	subnet-088b184cad0617077	Available	vpc-096ebcf6e9ad464e9	Off	172.31.32.0/20	us-west-2b
-	subnet-04dc67d9f2e56607a	Available	vpc-096ebcf6e9ad464e9	Off	172.31.0.0/20	us-west-2c
-	subnet-0c6017a67646a4a16	Available	vpc-096ebcf6e9ad464e9	Off	172.31.48.0/20	us-west-2d
-	subnet-04f0f1a04e97492aa	Available	vpc-096ebcf6e9ad464e9	Off	172.31.16.0/20	us-west-2a

- To reveal more details at the bottom of the page, select **Public Subnet 1**.

Note: To expand the lower window pane, drag the divider up and down. Alternatively, to choose a preset size for the lower pane you can choose one of the three square icons.



Image description: Preceding image depicts the icons available which will help the user adjust the pane size when viewing details of VPC services inside the AWS Management Console.

The screenshot shows the AWS Management Console interface for the VPC dashboard. On the left, there's a sidebar with various VPC-related options like EC2 Global View, Virtual private cloud, Subnets, Route tables, Internet gateways, Egress-only Internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, and Route servers. The 'Subnets' section is currently selected. The main content area displays a table of subnets with columns for Name, Subnet ID, State, VPC, Block Public Access, IPv4 CIDR, and Availability Zone. One subnet, 'Public Subnet 1' (subnet-02365a081971d24c9), is selected and expanded to show its details. The 'Details' tab is active, showing information such as Subnet ID (subnet-02365a081971d24c9), Subnet ARN (arn:aws:e2:us-west-2:903564999125:subnet/subnet-02365a081971d24c9), State (Available), IPv4 CIDR (10.0.0.0/24), Available IPv4 addresses (248), Availability Zone ID (usw2-az1), and other settings like Block Public Access (Off), VPC (vpc-096d40eaacaf19a4 | Lab VPC), and Auto-assign public IPv4 address.

- On the lower half of the page, choose the **Route table** tab.

This tab displays details about the routing for this subnet:

- The first entry specifies that traffic destined within the VPC's CIDR range (10.0.0.0/20) is routed within the VPC (local).
- The second entry specifies that any traffic destined for the internet (0.0.0.0/0) is routed to the internet gateway (igw-0e8ecb29bf33d8535). This configuration makes it a *public* subnet.

Subnets (1/8) Info

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	Availability Zone
Public Subnet 2	subnet-0e04f6b8017c1b988	Available	vpc-096d40eaacafdf19a4 Lab ...	Off	10.0.1.0/24	us-west-2b
Public Subnet 1	subnet-02365a081971d24c9	Available	vpc-096d40eaacafdf19a4 Lab ...	Off	10.0.0.0/24	us-west-2a
Private Subnet 2	subnet-076b1cb2d312b51e	Available	vpc-096d40eaacafdf19a4 Lab ...	Off	10.0.4.0/23	us-west-2b
Private Subnet 1	subnet-02ad7a020511da5c7	Available	vpc-096d40eaacafdf19a4 Lab ...	Off	10.0.2.0/23	us-west-2a

subnet-02365a081971d24c9 / Public Subnet 1

Route table: rtb-0517ea1f76e617d66 / Public Route Table

Routes (2)

Destination	Target
10.0.0.0/20	local
0.0.0.0/0	igw-0e8ecb29bf33d8535

- Choose the **Network ACL** tab.

This tab displays the network access control list (ACL) associated with the subnet. The rules currently permit *all* traffic to flow in and out of the subnet. You can further restrict the traffic by modifying the network ACL rules or by using security groups.

Subnets (1/8) Info

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	Availability Zone	Availability Zone ID	Network border group	Route table	Network ACL
Public Subnet 1	subnet-02365a081971d24c9	Available	vpc-096d40eaacafdf19a4 Lab ...	Off	10.0.0.0/24	us-west-2a	usw2-az1	us-west-2	rtb-0517ea1f76e617d66 Pub...	ad-0bc72403
Private Subnet 2	subnet-076b1cb2d312b51e	Available	vpc-096d40eaacafdf19a4 Lab ...	Off	10.0.4.0/23	us-west-2b	usw2-az2	us-west-2	rtb-038ef02c47986d65 Privat...	ad-0bc72403
Private Subnet 1	subnet-02ad7a020511da5c7	Available	vpc-096d40eaacafdf19a4 Lab ...	Off	10.0.2.0/23	us-west-2a	usw2-az1	us-west-2	rtb-038ef02c47986d65 Privat...	ad-0bc72403
-	subnet-088a184cad0617077	Available	vpc-096ecb5fe5edadd4e4d9	Off	172.31.32.0/20	us-west-2b	usw2-az2	us-west-2	rtb-052cd8367b026311b	ad-0de6e001
-	subnet-04667679f92f566077	Available	vpc-096ecb5fe5edadd4e4d9	Off	172.31.0.0/20	us-west-2c	usw2-az3	us-west-2	rtb-052cd8367b026311b	ad-0de6e001
-	subnet-0d0172a67645a6116	Available	vpc-096ecb5fe5edadd4e4d9	Off	172.31.48.0/20	us-west-2d	usw2-az4	us-west-2	rtb-052cd8367b026311b	ad-0de6e001

subnet-02365a081971d24c9 / Public Subnet 1

Network ACL: ad-0bc72403b7a138de

Inbound rules (2)

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Outbound rules (2)

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

- In the left navigation pane, choose **Internet gateways**.

An internet gateway called *Lab IG* is already associated with the Lab VPC.

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. A table lists one internet gateway:

Name	Internet gateway ID	State	VPC ID	Owner
Lab IG	igw-0e8ecb29bf33d8535	Attached	vpc-096d40eaacaf19a4 Lab VPC	903564999125

Below the table, the details for 'igw-0e8ecb29bf33d8535 / Lab IG' are shown, including its state as 'Attached'.

- In the left navigation pane, choose **Security groups**.

The screenshot shows the AWS Security Groups page with a list of 6 security groups:

Name	Security group ID	Security group name	VPC ID	Description
Inventory-DB	sg-0169ff651629c4b78	Inventory-DB	vpc-096d40eaacaf19a4	Enable access to MySQL
Inventory-ALB	sg-0fc76437d0b1f195	Inventory-ALB	vpc-096d40eaacaf19a4	Enable access to App
Inventory-App	sg-0ac419973d26f91a6	Inventory-App	vpc-096d40eaacaf19a4	Enable access to App
-	sg-0783dc928ba59988f	GuardDutyManagedSecurityGroup-vpc-...	vpc-096d40eaacaf19a4	Associated with VPC-vpc-096d40eaacaf..
-	sg-09192748503f153f4	default	vpc-096ebcf6e9ad464e9	default VPC security group
-	sg-0f2c23dd87433a7d3	default	vpc-096d40eaacaf19a4	default VPC security group

Below the table, a section titled 'Select a security group' is visible.

- Select the **Inventory-ALB** security group.

This is the security group used to control incoming traffic to the Application Load Balancer.

The screenshot shows the AWS VPC console with the 'Security groups' section selected. A search bar at the top is empty. Below it is a table titled 'Security Groups (1/6) Info' with columns: Name, Security group ID, Security group name, VPC ID, and Description. There are six rows: 'Inventory-DB' (sg-0169ff651629c4b78), 'Inventory-App' (sg-0ac419973d26f91a6), 'Inventory-ALB' (sg-0fc76437d0b1f195, checked), and three others. The 'Inventory-ALB' row has a tooltip 'GuardDutyManagedSecurityGroup-vpc-...' over the VPC ID column. Below the table is a section for 'sg-0fc76437d0b1f195 - Inventory-ALB' with tabs for Details, Inbound rules, Outbound rules, Sharing - new, VPC associations - new, and Tags. The 'Details' tab is active, showing details like Security group name (Inventory-ALB), Security group ID (sg-0fc76437d0b1f195), Owner (903564999125), Inbound rules count (1 Permission entry), Description (Enable access to App), and VPC ID (vpc-096d40eaacaf19a4).

- On the lower half of the page, choose the **Inbound rules** tab.

The security group permits inbound web traffic (port 80) from everywhere (*0.0.0.0/0*).

The screenshot shows the same AWS VPC console and security group list. The 'Inbound rules' tab is now active for the 'Inventory-ALB' security group. It displays a table titled 'Inbound rules (1)' with columns: Name, Security group rule ID, IP version, Type, Protocol, Port range, and Source. One rule is listed: sgr-0479e861679d80416, IPv4, HTTP, TCP, port 80, source 0.0.0.0/0.

- Choose the **Outbound rules** tab.

By default, security groups allow all outbound traffic. However, you can modify these rules as necessary.

Security Groups (1/6) Info

Name	Security group ID	Security group name	VPC ID	Description
Inventory-DB	sg-0169ff651629c4b78	Inventory-DB	vpc-096d40eaacafdf19a4	Enable access to MySQL
Inventory-App	sg-0ac419973d26f91a6	Inventory-App	vpc-096d40eaacafdf19a4	Enable access to App
Inventory-ALB	sg-0fcf76437d0b1f195	Inventory-ALB	vpc-096d40eaacafdf19a4	Enable access to App
-	sg-0783dc928ba59988f	GuardDutyManagedSecurityGroup-vpc-...	vpc-096d40eaacafdf19a4	Associated with VPC-vpc-096d40eaacaf..

sg-0fcf76437d0b1f195 - Inventory-ALB

Outbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination
-	sgr-0b1ed693f7af7de18	IPv4	All traffic	All	All	0.0.0.0/0

- Select the **Inventory-App** security group. Ensure that it is the only security group selected.

This is the security group used to control incoming traffic to the *AppServer* EC2 instance.

Security Groups (1/6) Info

Name	Security group ID	Security group name	VPC ID	Description
Inventory-DB	sg-0169ff651629c4b78	Inventory-DB	vpc-096d40eaacafdf19a4	Enable access to MySQL
Inventory-App	sg-0ac419973d26f91a6	Inventory-App	vpc-096d40eaacafdf19a4	Enable access to App
Inventory-ALB	sg-0fcf76437d0b1f195	Inventory-ALB	vpc-096d40eaacafdf19a4	Enable access to App
-	sg-0783dc928ba59988f	GuardDutyManagedSecurityGroup-vpc-...	vpc-096d40eaacafdf19a4	Associated with VPC-vpc-096d40eaacaf..

sg-0ac419973d26f91a6 - Inventory-App

Details

Security group name Inventory-App	Security group ID sg-0ac419973d26f91a6	Description Enable access to App
Owner 903564999125	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry

- On the lower half of the page, choose the **Inbound rules** tab.

The security group only permits inbound web traffic (port 80) from the Application Load Balancer security group (*Inventory-ALB*).

The screenshot shows the AWS VPC Security Groups page. On the left, there's a navigation sidebar with options like Egress-only Internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers, Security (Network ACLs, Security groups), PrivateLink and Lattice, and more. The main area displays a table of security groups. One row is selected: "Inventory-App" (sg-0ac419973d26f91a6). Below the table, a sub-section titled "sg-0ac419973d26f91a6 - Inventory-App" shows tabs for Details, Inbound rules, Outbound rules, Sharing - new, VPC associations - new, and Tags. The "Inbound rules" tab is active, showing a table with one rule: "sgr-0e0ac82891b7534dc" (Type: HTTP, Protocol: TCP, Port range: 80, Source: sg-0fcf76437d0b1f195).

- Choose the **Outbound rules** tab.

By default, security groups allow all outbound traffic. As with the outbound rules for the Application Load Balancer security group, you can modify these rules as necessary.

This screenshot is identical to the previous one, but the "Outbound rules" tab is now active. The sub-section "sg-0ac419973d26f91a6 - Inventory-App" shows the same tabs: Details, Inbound rules, Outbound rules, Sharing - new, VPC associations - new, and Tags. The "Outbound rules" tab is active, showing a table with one rule: "sgr-0aca35584c80fb49" (IP version: IPv4, Type: All traffic, Protocol: All, Port range: All, Destination: 0.0.0.0/0).

- Select the **Inventory-DB** security group. Ensure that it is the only security group selected.

This is the security group used to control incoming traffic to the database.

The screenshot shows the AWS VPC Security Groups page. On the left, there's a navigation sidebar with sections like Egress-only Internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers, Security (Network ACLs, Security groups), PrivateLink and Lattice, and Service networks. The 'Security groups' section is expanded, showing 'Inventory-DB' selected. The main content area displays a table of security groups with columns for Name, Security group ID, Security group name, VPC ID, and Description. Below the table, a specific security group ('sg-0169ff651629c4b78 - Inventory-DB') is selected, and its details are shown in a modal. The modal includes tabs for Details, Inbound rules, Outbound rules, Sharing - new, VPC associations - new, and Tags. The 'Inbound rules' tab is selected, showing one rule: 'sg-0027fd7e5e0c5a0b3' allowing MySQL/Aurora traffic (TCP port 3306) from the 'Inventory-App' security group.

- On the lower half of the page, choose the **Inbound rules** tab.

The security group permits inbound MySQL/Aurora traffic (port 3306) from the application server security group (*Inventory-App*).

This screenshot is identical to the one above, showing the AWS VPC Security Groups page. The 'Inventory-DB' security group is selected, and its details are shown in a modal. The 'Inbound rules' tab is selected, displaying one rule: 'sg-0027fd7e5e0c5a0b3' allowing MySQL/Aurora traffic (TCP port 3306) from the 'Inventory-App' security group.

- Choose the **Outbound rules** tab.

By default, security groups allow all outbound traffic. As with the outbound rules for the previous security groups, you can modify these rules as necessary.

Security Groups (1/6) Info

Name	Security group ID	Security group name	VPC ID	Description
Inventory-DB	sg-0169ff651629c4b78	Inventory-DB	vpc-096d40eaacaf19a4	Enable access to MySQL
Inventory-App	sg-0ac419973d26f91a6	Inventory-App	vpc-096d40eaacaf19a4	Enable access to App
Inventory-ALB	sg-0fcfb764370b1f195	Inventory-ALB	vpc-096d40eaacaf19a4	Enable access to App
-	sg-0783d928ba59988f	GuardDutyManagedSecurityGroup-vpc-...	vpc-096d40eaacaf19a4	Associated with VPC-vpc-096d40eaacaf..

sg-0169ff651629c4b78 - Inventory-DB

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

Outbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination
-	sgr-00391f03737201a79	IPv4	All traffic	All	All	0.0.0.0/0

Task 1.2: Examine the EC2 instance

An EC2 instance has been provided for you. This instance runs a basic Hypertext Preprocessor (PHP) application that tracks inventory in a database. In this task, you inspect the instance details.

- At the top of the console, in the search bar, search for and choose **EC2**.

Resources

You are using the following Amazon EC2 resources in the United States (Oregon) Region:

Instances (running)	1	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	1	Instances	1
Key pairs	0	Load balancers	1	Placement groups	0
Security groups	6	Snapshots	0	Volumes	1

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Service health

Region: United States (Oregon)

Status: This service is operating normally.

Zones

Zone name	Zone ID
us-west-2a	usw2-az1

- In the left navigation pane, choose **Instances**.

Instances (1) Info

Last updated less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
AppServer	i-029a444b99138ac3c	Running	t3.micro	3/3 checks passed	View alarms +	us-west-2a	-	-

- Select the **AppServer** instance to reveal more details at the bottom of the page.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with links like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main content area displays the 'Instances (1/1) Info' section. A search bar at the top allows filtering by Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IP. The table lists one instance: 'AppServer' (i-029a444b99138ac3c), which is 'Running' (t3.micro), has 3/3 checks passed, and is in us-west-2a. Below the table, the instance details for 'i-029a444b99138ac3c (AppServer)' are shown under the 'Details' tab. The 'Instance summary' section includes fields for Instance ID (i-029a444b99138ac3c), Instance state (Running), Hostname type (IP name: ip-10-0-3-246.us-west-2.compute.internal), and Instance type (t3.micro). It also lists Public IPv4 address, Private IP DNS name (IPv4 only) (ip-10-0-3-246.us-west-2.compute.internal), and Private IPv4 addresses (10.0.3.246).

- After reviewing the instance details, choose the **Actions** dropdown menu, choose **Instance settings**, and then choose **Edit user data**.

The screenshot shows the same AWS EC2 Instances page as before, but with the Actions dropdown menu open on the right side of the instance table. The 'Edit user data' option is highlighted. The main content area shows the instance details for 'i-029a444b99138ac3c (AppServer)' under the 'Details' tab. The 'Edit user data' section contains the current user data:

```
#!/bin/bash
#yum -y update
dnf update -y
dnf install -y httpd wget php-fpm php-mysql php-json php php-devel

# Install and enable AWS Systems Manager Agent
```

Below this, there's a note: "To edit your instance's user data you first need to stop your instance." At the bottom right of the modal are 'Cancel' and 'Save' buttons.

- On the **Edit user data** page, choose **Copy user data**.
- Paste the user data you just copied into a text editor. You use it in a later task.

The Userdata copied is as follows:

```
#!/bin/bash
#yum -y update
dnf update -y      # Updates all installed packages on the system
without prompting for confirmation (using the dnf package manager).
dnf install -y httpd wget php-fpm php-mysqli php-json php php-devel #
Installs the Apache HTTP Server (httpd), the wget utility, PHP FastCGI
Process Manager (php-fpm), PHP MySQLi extension, PHP JSON extension,
the core PHP package, and PHP development headers without prompting for
confirmation.

# Install and enable AWS Systems Manager Agent
cd /tmp      # Changes the current directory to /tmp.
systemctl enable amazon-ssm-agent      # Enables the AWS Systems Manager
Agent service to start automatically on boot.
systemctl start amazon-ssm-agent      # Starts the AWS Systems Manager
Agent service immediately.

# Download Inventory App Lab files
wget https://us-west-2-tcprod.s3.us-west-2.amazonaws.com/courses/ILT-
TF-200-ARCHIT/v7.9.8.prod-d6e2cf0a/lab-4-HA/scripts/inventory-app.zip
# Downloads the inventory-app.zip file from the specified S3 bucket.
unzip inventory-app.zip -d /var/www/html/  # Unzips the contents of
inventory-app.zip into the /var/www/html/ directory.

# Download and install the AWS SDK for PHP
wget
https://github.com/aws/aws-sdk-php/releases/download/3.295.2/aws.zip
# Downloads the AWS SDK for PHP as a zip file.
unzip aws -d /var/www/html      # Unzips the contents of the 'aws' file
(which is likely a previous or partial unzip) into the /var/www/html
directory. This line might be redundant or incorrect if 'aws' itself is
the zip file.
unzip /var/www/html/aws.zip -d /var/www/html/      # Unzips the
contents of the aws.zip file located in /var/www/html/ into the
/var/www/html/ directory.

# Load Amazon Aurora DB connection details from AWS CloudFormation
un="dbadmin"          # Sets a shell variable 'un' (username) to
"dbadmin".
pw="lab-password"    # Sets a shell variable 'pw' (password) to "lab-
password".
```

```

ep="inventory-cluster.cluster-c6msorg80ejw.us-west-2.rds.amazonaws.com"
# Sets a shell variable 'ep' (endpoint) to the specified Amazon Aurora
cluster endpoint.

db="inventory"      # Sets a shell variable 'db' (database name) to
"inventory".

# Populate PHP app settings with DB info
sed -i "s/DBENDPOINT/$ep/g" /var/www/html/get-parameters.php      #
Uses the 'sed' command to replace all occurrences of "DBENDPOINT" in
the /var/www/html/get-parameters.php file with the value of the '$ep'
variable (the database endpoint). The '-i' option modifies the file in-
place.

sed -i "s/DBNAME/$db/g" /var/www/html/get-parameters.php      #
Uses the 'sed' command to replace all occurrences of "DBNAME" in the
/var/www/html/get-parameters.php file with the value of the '$db'
variable (the database name).

sed -i "s/DBUSERNAME/$un/g" /var/www/html/get-parameters.php      #
Uses the 'sed' command to replace all occurrences of "DBUSERNAME" in
the /var/www/html/get-parameters.php file with the value of the '$un'
variable (the database username).

sed -i "s/DBPASSWORD/$pw/g" /var/www/html/get-parameters.php      #
Uses the 'sed' command to replace all occurrences of "DBPASSWORD" in
the /var/www/html/get-parameters.php file with the value of the '$pw'
variable (the database password).

# Turn on web server
systemctl enable --now httpd      # Enables the Apache HTTP Server
service to start automatically on boot and starts it immediately.

```

This UserDatascript is like a quick recipe to build a simple website for tracking your stuff on a computer in the cloud. First, it tidies up the computer and gets all the latest tools. Then, it installs the main ingredients: a program to show websites (like a waiter), the special language the website uses (PHP), and some helpers so the website can talk to databases and other online services. It also installs a special helper from Amazon so you can manage the computer from afar. Next, it downloads the actual website files and unpacks them into the right place. It also gets some extra tools from Amazon that the website might need. Finally, it takes the secret keys and the address for where the information about your stuff is stored (the database) and writes them into the website's settings so it knows where to look. Lastly, it turns on the website server so you can open the inventory website in your internet browser and start using it.

Task 1.3: Examine the load balancer configuration

An Application Load Balancer and target group have been provided for you. In this task, you review their configuration.

- Expand the navigation menu by choosing the menu icon in the upper-left corner.

- In the left navigation pane, choose **Target Groups**.

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
Inventory-App	arn:aws:elasticloadbalancing:us-west-2:903564999125:targetgroup/Inventory-App/812824c85b954aa2	80	HTTP	Instance	Inventory-LB	vpc-096d40eaacaf19a4

0 target groups selected
Select a target group above.

- Select the **Inventory-App** target group to reveal more details at the bottom of the page.

Target type	Protocol	IP address type	Load balancer	VPC
Instance	HTTP: 80	IPv4	Inventory-LB	vpc-096d40eaacaf19a4

- On the lower half of the page, choose the **Targets** tab.

The Application Load Balancer forwards incoming requests to all targets on the list. The AppServer EC2 instance you examined earlier is already registered as a target.

The screenshot shows the AWS EC2 Target groups page. In the left navigation pane, under the 'Load Balancing' section, 'Target Groups' is selected. The main content area displays a table titled 'Target groups (1/1)'. A single row is shown for 'Inventory-App', which is associated with an ARN (arn:aws:elasticloadbalancing:us-west-2:591862542:targetgroup/Inventory-LB), port 80, protocol HTTP, target type Instance, and load balancer Inventory-LB. A VPC ID is also listed as vpc-096d40eaacafdf19a4.

- In the left navigation pane, choose **Load Balancers**.

The screenshot shows the AWS EC2 Load balancers page. In the left navigation pane, under the 'Load Balancing' section, 'Load Balancers' is selected. The main content area displays a table titled 'Load balancers (1)'. A single row is shown for 'Inventory-LB', which is active, has a VPC ID of vpc-096d40eaacafdf19a4, and is associated with two availability zones. The type is listed as application, and it was created on April 18, 2025, at 13:29 (UTC+0).

- Choose the **Inventory-LB** load balancer name to reveal more details.

The screenshot shows the AWS EC2 Load balancer details page for 'Inventory-LB'. In the left navigation pane, under the 'Load Balancing' section, 'Load Balancers' is selected. The main content area shows the 'Details' tab for 'Inventory-LB'. Key details include the load balancer type (Application), status (Active), scheme (Internet-facing), hosted zone (Z1H1FLSHABSF5), VPC (vpc-096d40eaacafdf19a4), and availability zones (us-west-2a and us-west-2b). The load balancer IP address type is IPv4, and it was created on April 18, 2025, at 13:29 (UTC+0).

Task 1.4: Open the PHP inventory application in a web browser

To confirm the inventory application is working correctly, you need to retrieve the URL for the inventory application settings page.

- Copy the **InventoryAppSettingsPageURL** (<http://Inventory-LB-591862542.us-west-2.elb.amazonaws.com/settings.php>)

Note: It should be similar to <http://Inventory-LB-xxxx.elb.amazonaws.com/settings.php>.

- Open a new web browser tab, paste the URL you copied in the previous step, and press **Enter**.

The settings page for the inventory application is displayed. The database endpoint, database name, and login details are already populated with the values for the Aurora database.

The screenshot shows a web browser window with the following details:

- Address Bar:** Not secure <http://inventory-lb-591862542.us-west-2.elb.amazonaws.com/settings.php>
- Page Title:** Inventory
- Section:** Settings
- Form Fields:**
 - Endpoint: inventory-cluster.cluster-c6msorg80ejw.us-west-2.rds.amazonaws.com
 - Database: inventory
 - Username: dbadmin
 - Password: lab-password
- Buttons:** Save

- Leave all the settings on the inventory app settings page as the default configurations.
- Choose **Save**.

After saving the settings, the inventory application redirects to the main page, and inventory for various items are displayed. You can add items to the inventory or modify the details of the existing inventory items. When you interact with this application, the load balancer forwards your requests to the previous AppServer in the load balancer's target group. The AppServer registers any inventory changes in the Aurora database. The bottom of the page displays the instance ID and the Availability Zone where the instance resides.

The screenshot shows a web browser window with the URL <http://inventory-lb-591862542.us-west-2.elb.amazonaws.com>. The page title is "Inventory". The main content is a table with columns "Store", "Item", and "Quantity". The data is as follows:

Store	Item	Quantity
Puerto Rico	Amazon Echo	12
Paris	Amazon Dot	3
Detroit	Amazon Tap	5

+ Add inventory

This page was generated by instance i-029a444b99138ac3c in Availability Zone us-west-2a.

Note: Leave this inventory application web browser tab open while working on the remaining lab tasks. You return to it in later tasks.

You have now finished inspecting all of the resources created for you in the lab environment and successfully accessed the provided inventory application. Next, you create a launch template to use with Amazon EC2 Auto Scaling to make the inventory application highly available.

Task 2: Create a launch template

Note: Before you can create an Auto Scaling group, you must create a launch template that includes the parameters required to launch an EC2 instance, such as the ID of the Amazon Machine Image (AMI) and an instance type.

In this task, you create a launch template.

- At the top of the console, in the search bar, search for and choose **EC2**.

The screenshot shows the AWS EC2 console dashboard. The left sidebar navigation includes EC2, Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main content area has several sections:

- Resources:** Shows counts for various EC2 resources across the United States (Oregon) Region.
- Launch instance:** A button to "Launch instance" and a note that instances will launch in the United States (Oregon) Region.
- Service health:** A link to the AWS Health Dashboard.
- Zones:** A table for managing zones with columns for Zone name and Zone ID.
- Account attributes:** A section for managing account settings.
- Additional information:** Links to Get started walkthroughs, Getting started guide, Documentation, All EC2 resources, Forums, Pricing, and Contact us.

- In the left navigation pane, below **Instances**, choose **Launch Templates**.

EC2

Compute

EC2 launch templates

Streamline, simplify and standardize instance launches

Use launch templates to automate instance launches, simplify permission policies, and enforce best practices across your organization. Save launch parameters in a template that can be used for on-demand launches and with managed services, including EC2 Auto Scaling and EC2 Fleet. Easily update your launch parameters by creating a new launch template version.

New launch template

Create launch template

Benefits and features

- Streamline provisioning**
Minimize steps to provision instances. With EC2 Auto Scaling, updates to a launch template can be automatically passed to an Auto Scaling group. [Learn more](#)
- Simplify permissions**
Create shorter, easier to manage IAM policies. [Learn more](#)

Documentation

Documentation [API reference](#)

- Choose **Create launch template**.
- In the **Launch template name and description** section, configure the following:
 - Launch template name:** Enter **Lab-launch-template**.
 - Template version description:** Enter **version 1**

aws | Search [Alt+S]

EC2 VPC S3 IAM Lambda

EC2 > Launch templates > Create launch template

Create launch template

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

Launch template name and description

Launch template name - *required*

Lab-launch-template

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

Template version description

version 1

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](#)

You must choose an *AMI*. An AMI is an image defining the root volume of the instance along with its operating system, applications, and related details. Without this information, your template would be unable to launch new instances.

AMIs are available for various operating systems (OSs). In this lab, you launch instances running the Amazon Linux 2023 OS.

- For **Application and OS Images (Amazon Machine Image) Info**, choose the **Quick Start** tab.
- Choose **Amazon Linux** as the OS.
- For **Amazon Machine Image**, choose **Amazon Linux 2023 AMI**.
- For **Instance type**, choose **t3.micro** from the dropdown menu.

The screenshot shows the AWS Management Console with the EC2 service selected. The navigation bar includes links for EC2, VPC, S3, IAM, and Lambda. Below the navigation bar, the path is shown as EC2 > Launch templates > Create launch template. The main content area is titled "Launch template contents" with a sub-section "Application and OS Images (Amazon Machine Image) Info". A search bar is present above a grid of operating system icons. The "Quick Start" tab is selected. A specific AMI, "Amazon Linux 2023 AMI", is highlighted in the grid. Below the grid, detailed information about the AMI is provided, including its ID, publish date, and a "Verified provider" badge. The "Instance type" section at the bottom shows the "t3.micro" instance type selected. Advanced options are available for this section.

When you launch an instance, the *instance type* determines the hardware allocated to your instance. Each instance type offers different compute, memory, and storage capabilities, and they are grouped in *instance families* based on these capabilities.

- In **Key pair (login)** section, for the **key pair name**, choose **Don't include in launch template**.
- In the **Network Settings** section, for **Security groups**, choose **Inventory-App**.

aws | Search [Alt+S] | [] [] [] [] []

EC2 VPC S3 IAM Lambda

EC2 > Launch templates > Create launch template

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

Don't include in launch template [Create new key pair](#)

▼ Network settings [Info](#)

Subnet [Info](#)

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](#)

Inventory-App sg-0ac419973d26f91a6 X
VPC: vpc-096d40eaaacafdf19a4

► Advanced network configuration

- Scroll down to the **Advanced details** section.

aws | Search [Alt+S] | [] [] [] [] []

EC2 VPC S3 IAM Lambda

EC2 > Launch templates > Create launch template

Inventory-App sg-0ac419973d26f91a6 X
VPC: vpc-096d40eaaacafdf19a4

► Advanced network configuration

▼ Storage (volumes) [Info](#)

EBS Volumes [Hide details](#)

Volume 1 (AMI Root) : 8 GiB, EBS, General purpose SSD (gp3), 3000 IOPS
AMI Volumes are not included in the template unless modified

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage [X](#)

Add new volume

▼ Resource tags [Info](#)

No resource tags are currently included in this template. Add a resource tag to include it in the launch template.

Add new tag

You can add up to 50 more tags.

► Advanced details [Info](#)

- Expand **Advanced details**.

The screenshot shows the 'Advanced details' section of the AWS EC2 Launch Templates configuration. It includes fields for IAM instance profile, Hostname type, DNS Hostname, Instance auto-recovery, Shutdown behavior, Stop - Hibernate behavior, Termination protection, Stop protection, and Detailed CloudWatch monitoring. Each field has a dropdown menu set to 'Don't include in launch template'. A 'Create new IAM profile' button is visible.

Advanced details

IAM instance profile: Don't include in launch template | Create new IAM profile

Hostname type: Don't include in launch template

DNS Hostname:
Enable resource-based IPv4 (A record) DNS requests
Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery: Don't include in launch template

Shutdown behavior: Don't include in launch template

Stop - Hibernate behavior: Don't include in launch template

Termination protection: Don't include in launch template

Stop protection: Don't include in launch template

Detailed CloudWatch monitoring: Don't include in launch template

- For **IAM instance profile**, choose **Inventory-App-Role**.

The screenshot shows the same 'Advanced details' configuration page as above, but with the 'IAM instance profile' dropdown set to 'Inventory-App-Role'. The other fields remain at their previous settings.

Advanced details

IAM instance profile: Inventory-App-Role | Create new IAM profile

Hostname type: Don't include in launch template

DNS Hostname:
Enable resource-based IPv4 (A record) DNS requests
Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery: Don't include in launch template

Shutdown behavior: Don't include in launch template

Stop - Hibernate behavior: Don't include in launch template

Termination protection: Don't include in launch template

Stop protection: Don't include in launch template

Screenshot of the AWS EC2 'Create launch template' configuration page. The interface shows various sections with dropdown menus and radio button options.

- Detailed CloudWatch monitoring**: Info dropdown, set to "Don't include in launch template".
- Credit specification**: Info dropdown, set to "Don't include in launch template".
- Placement group**: Info dropdown, set to "Don't include in launch template". To the right is a link to "Create new placement group".
- EBS-optimized instance**: Info dropdown, set to "Don't include in launch template".
- Instance bandwidth configuration**: Info dropdown, set to "Don't include in launch template".
- Purchasing option**: Info section with three radio button options:
 - Don't include in launch template
 - Capacity Blocks
Launch Instances for your active capacity blocks
 - Spot instances
Request Spot Instances at the Spot price, capped at the On-Demand price
- Capacity reservation**: Info dropdown, set to "Don't include in launch template".
- Tenancy**: Info dropdown, set to "Don't include in launch template".

Screenshot of the AWS EC2 'Create launch template' configuration page, showing different settings compared to the first screenshot.

- RAM disk ID**: Info dropdown, set to "Don't include in launch template".
- Kernel ID**: Info dropdown, set to "Don't include in launch template".
- Nitro Enclave**: Info dropdown, set to "Don't include in launch template".
- License configurations**: Info dropdown, set to "Don't include in launch template". To the right is a link to "Create new license configuration".
- CPU options**: Info section with a note: "Configure CPUs for your instance to optimize performance and save on licensing costs." and two radio button options:
 - Use default CPU options
 - Specify CPU options
- Default active vCPUs**: Set to 2.
- Total vCPUs**: Set to 2.

- For **Metadata version**, choose **V2 only (token required)**.

Metadata accessible | [Info](#)

Metadata IPv6 endpoint | [Info](#)

Metadata version | [Info](#)

⚠️ For V2 requests, you must include a session token in all instance metadata requests. Applications or agents that use V1 for instance metadata access will break.

Metadata response hop limit | [Info](#)

Allow tags in metadata | [Info](#)

User data - optional | [Info](#)
 Upload a file with your user data or enter it in the field.

```
#!/bin/bash
#yum -y update
dnf update -y
dnf install -y httpd wget php-fpm php-mysql php-json php-devel

# Install and enable AWS Systems Manager Agent
cd /tmp
systemctl enable amazon-ssm-agent
systemctl start amazon-ssm-agent

# Download Inventory App Lab files
wget https://us-west-2-tcprod.s3.us-west-2.amazonaws.com/courses/ILT-TF-200-ARCHIT/v7.9.8.prod-d6e2cf0a/lab-4-HA/scripts/inventory-app.zip
unzip inventory-app.zip -d /var/www/html/
```

- In the **User data** section, paste the user data you saved to your text editor during Task 1.2.

User data - optional | [Info](#)
 Upload a file with your user data or enter it in the field.

```
#!/bin/bash
#yum -y update
dnf update -y
dnf install -y httpd wget php-fpm php-mysql php-json php-devel

# Install and enable AWS Systems Manager Agent
cd /tmp
systemctl enable amazon-ssm-agent
systemctl start amazon-ssm-agent

# Download Inventory App Lab files
wget https://us-west-2-tcprod.s3.us-west-2.amazonaws.com/courses/ILT-TF-200-ARCHIT/v7.9.8.prod-d6e2cf0a/lab-4-HA/scripts/inventory-app.zip
unzip inventory-app.zip -d /var/www/html/
```

Software Image (AMI)
 Amazon Linux 2023 AMI 2023.7.2...[read more](#)
 ami-05572e592e80ae689

Virtual server type (instance type)
 t3.micro

Firewall (security group)
 Inventory-App

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 t2.micro when t2.micro isn't available)

User data has already been base64 encoded

[Cancel](#) [Create launch template](#)

- Choose **Create launch template**.

Success
Successfully created Lab-launch-template[lt-02cbefcf167cf7b].

Actions log

Next Steps

Launch an instance
With On-Demand Instances, you pay for compute capacity by the second (for Linux, with a minimum of 60 seconds) or by the hour (for all other operating systems) with no long-term commitments or upfront payments. Launch an On-Demand Instance from your launch template.

Launch instance from this template

Create an Auto Scaling group from your template
Amazon EC2 Auto Scaling helps maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.

Create Auto Scaling group

Create Spot Fleet
A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance type in each Availability Zone) is set by Amazon EC2, and adjusted gradually based on the long-term supply of and demand for Spot Instances. Spot instances are well-suited for data-analysis, batch jobs, background processing, and optional tasks.

Create Spot Fleet

View launch templates

- Choose **View launch templates**.

Launch Templates (1/1)

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By	Managed	Operator
lt-02cbefcf167cf7b	Lab-launch-template	1	1	2025-04-18T12:30:43.000Z	arn:aws:sts::903564999125:assumed-role/AWSLabsUser-hvREsEWSpL9xbV5tdgDNey/bf420498-e8ed-4bd3-8727-cd8a9792f173	false	-

Lab-launch-template (lt-02cbefcf167cf7b)

Launch template details

Launch template ID lt-02cbefcf167cf7b	Launch template name Lab-launch-template	Default version 1	Owner arn:aws:sts::903564999125:assumed-role/AWSLabsUser-hvREsEWSpL9xbV5tdgDNey/bf420498-e8ed-4bd3-8727-cd8a9792f173
--	---	----------------------	---

Details **Versions** **Template tags**

Launch template version details

Version 1 (Default)	Description version 1	Date created 2025-04-18T12:30:43.000Z	Created by arn:aws:sts::903564999125:assumed-role/AWSLabsUser-hvREsEWSpL9xbV5tdgDNey/bf420498-e8ed-4bd3-8727-cd8a9792f173
Instance details	Storage	Resource tags	Advanced details
AMI ID ami-05572c392e80ace09	Instance type t3.micro	Availability Zone -	Key pair name -
Security groups sg-0ac419973d26f91a6	Security group IDs sg-0ac419973d26f91a6		

You have successfully created the launch template.

Task 3: Create an Auto Scaling group

In this task, you create an Auto Scaling group that deploys EC2 instances across your *private subnets*. This is a security best practice when deploying applications because instances in a private subnet cannot be accessed from the internet. Instead, users send requests to the Application Load Balancer, which forwards the requests to the EC2 instances in the private subnets, as shown in the following diagram:

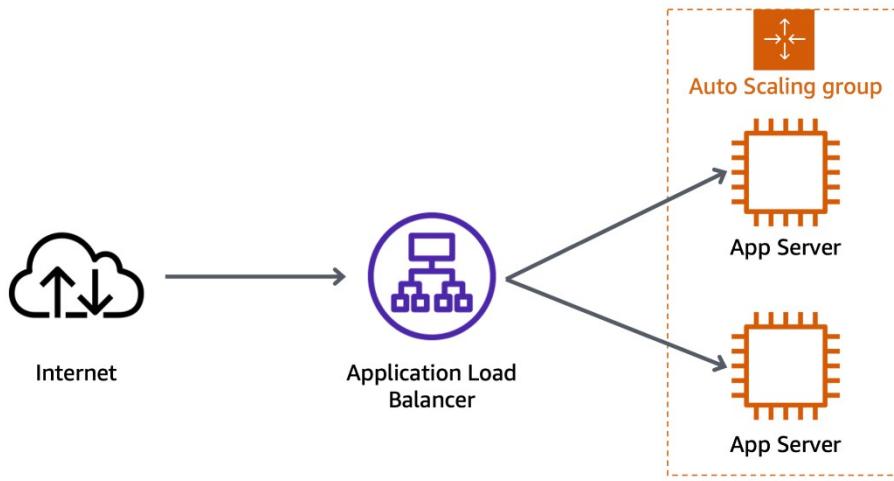


Image description: Preceding image depicts how traffic flowing from internet is distributed to multiple EC2 instances behind an Auto-Scaling-Group (ASG).

NOTE: Application Load Balancer (ALB)s have health checks which check the health of the EC2 instance before distributing the traffic to a particular EC2 instance - avoiding sending any traffic to an instance which is being de-commissioned later due to Auto-Scaling actions.

Learn more: Amazon EC2 Auto Scaling is a service designed to *launch* or *terminate* EC2 instances automatically based on user-defined policies, schedules, and health checks. The service also automatically distributes instances across multiple Availability Zones to make applications highly available. For more information, see [What is Amazon EC2 Auto Scaling?](#).

- In the left navigation pane, below Auto Scaling, choose **Auto Scaling Groups**.

The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar has a tree view with 'Auto Scaling Groups' expanded. The main content area is titled 'Amazon EC2 Auto Scaling' and describes it as helping to maintain application availability. It includes a diagram showing an 'Auto Scaling group' containing three instances, with labels for 'Minimum size', 'Desired capacity', 'Maximum size', and 'Scale out as needed'. To the right, there are sections for 'Pricing', 'Getting started', and 'FAQ'.

- Choose **Create Auto Scaling group** and configure the following:
 - Auto Scaling group name:** Enter **Inventory-ASG**
 - Launch template:** From the dropdown menu, select the launch template that you created earlier.

The screenshot shows the 'Create Auto Scaling group' wizard at Step 1: Choose launch template or configuration. The 'Auto Scaling group name' field contains 'Inventory-ASG'. The 'Launch template' dropdown shows 'Lab-launch-template' with version 1 selected. Other configuration options like AMI ID, Security groups, and Instance type (t3.micro) are also visible.

- Choose **Next**.

The **Choose instance launch options** page is displayed.

The screenshot shows the 'Choose instance launch options' page. The 'Launch template' dropdown shows 'Lab-launch-template' with version 1 selected. Under 'Network', the VPC 'vpc-096bcf69ad464c9' is chosen. In the 'Availability Zones and subnets' section, 'Select Availability Zones and subnets' is set to '172.31.0.0/16 - Default'. The 'Availability Zone distribution' dropdown is set to 'Balanced best effort'. Navigation buttons at the bottom include 'Cancel', 'Skip to review', 'Previous', and 'Next'.

- In the **Network** section, configure the following:
 - **VPC:** Select **Lab VPC** from the dropdown menu.
 - **Availability Zones and subnets:** Select **Private Subnet 1** and **Private Subnet 2** from the dropdown menu.

Choose instance launch options

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

Instance type requirements

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template	Version	Description
Lab-launch-template	Default	version 1

Instance type

t3.micro

Network

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-096d40eacafdf19a4 (Lab VPC)
10.0.0.0/20

Create a VPC

Availability Zones and subnets

Select Availability Zones and subnets

us-west-2a | subnet-02ad7a020511da5c7 (Private Subnet 1) X
10.0.2.0/23

us-west-2b | subnet-076b1cb82d512b051e (Private Subnet 2) X
10.0.4.0/23

Create a subnet

Availability Zone distribution - new

Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

Balanced best effort
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

Balanced only
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

Cancel **Skip to review** **Previous** **Next**

- Choose **Next**.

Integrate with other services - 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.

Load balancing

No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

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

Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

VPC Lattice integration options

No VPC Lattice service
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

Attach to VPC Lattice service
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

Application Recovery Controller (ARC) zonal shift - new

During an Availability Zone impairment, target instance launches towards other healthy Availability Zones.

Enable zonal shift
New instance launches will be retargeted towards healthy Availability Zones until the zonal shift is canceled.

Cancel **Skip to review** **Previous** **Next**

- On the **Integrate with other services - optional** page, configure the following:
 - Select **Attach to an existing load balancer**.
 - Select **Choose from your load balancer target groups**.
 - From the **Existing load balancer target groups** dropdown menu, select **Inventory-App | HTTP**.

Integrate with other services - optional Info

Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

Load balancing Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

- No load balancer Traffic to your Auto Scaling group will not be fronted by a load balancer.
- Attach to an existing load balancer Choose from your existing load balancers.
- Attach to a new load balancer Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer

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

- Choose from your load balancer target groups This option allows you to attach Application, Network, or Gateway Load Balancers.
- Choose from Classic Load Balancers

Existing load balancer target groups

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

Inventory-App | HTTP X
Application Load Balancer: Inventory-LB

This tells the Auto Scaling group to register new EC2 instances as part of the *Inventory-App* target group that you examined earlier. The load balancer sends traffic to instances that are in this target group.

○ Health check grace period: Enter 300

By default, the health check grace period is set to 300.

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks

Always enabled

Additional health check types - optional Info

- Turn on Elastic Load Balancing health checks Recommended Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.
- Turn on VPC Lattice health checks VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.
- Turn on Amazon EBS health checks EBS monitors whether an instance's root volume or attached volume stalls. When it reports an unhealthy volume, EC2 Auto Scaling can replace the instance on its next periodic health check.

Health check grace period Info

This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

300 seconds

Cancel **Skip to review** **Previous** **Next**

- Choose **Next**.

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

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Desired capacity

Specify your group size.

1

Scaling

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity

1

Max desired capacity

1

Equal or less than desired capacity Equal or greater than desired capacity

Automatic scaling - optional

Choose whether to use a target tracking policy

No scaling policies

Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy

Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

- On the **Configure group size and scaling - optional** page, configure the following:
 - Desired capacity:** Enter 2
 - Min desired capacity:** Enter 2
 - Max desired capacity:** Enter 2

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

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Desired capacity

Specify your group size.

2

Scaling

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity

2

Max desired capacity

2

Equal or less than desired capacity Equal or greater than desired capacity

Automatic scaling - optional

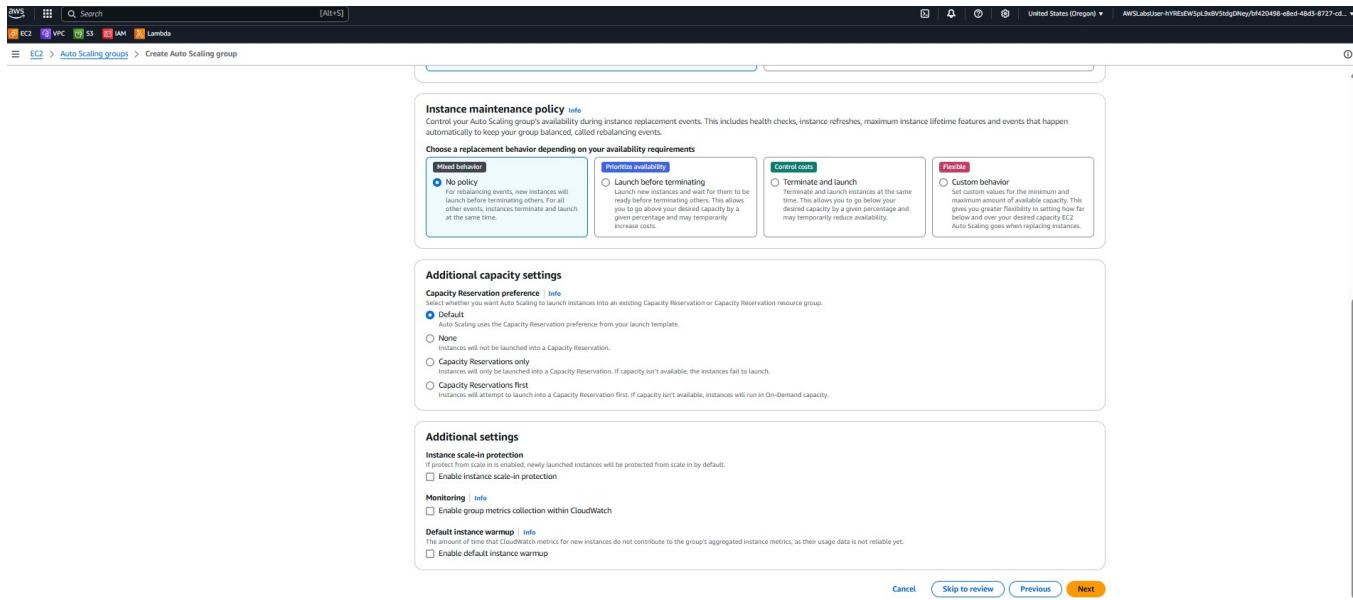
Choose whether to use a target tracking policy

No scaling policies

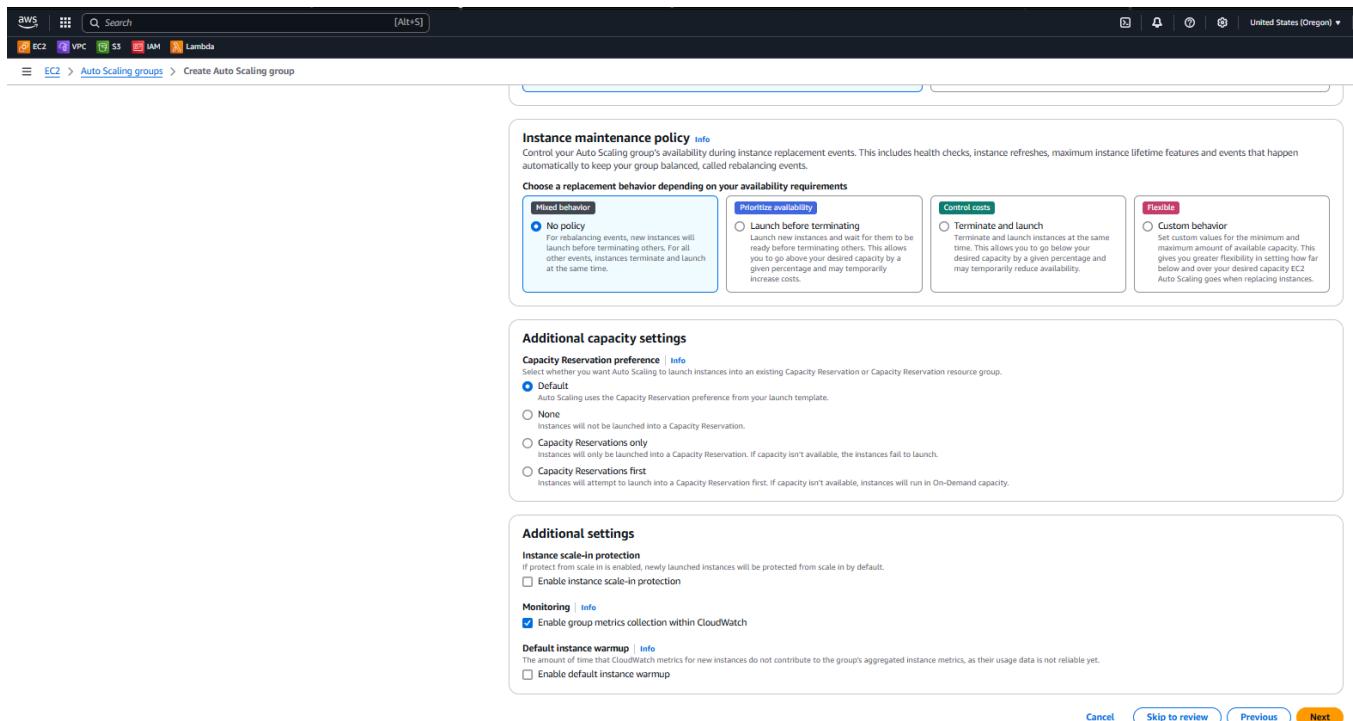
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy

Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.



- In the **Additional settings** section, choose **Enable group metrics collection within CloudWatch**.



- Choose **Next**.

For this lab, you always maintain two instances to ensure high availability. If the application is expected to receive varying loads of traffic, it is also possible to create *scaling policies* that define when to launch and terminate instances. However, this is not necessary for the Inventory application in this lab.

The screenshot shows the AWS EC2 Auto Scaling group creation wizard at Step 5: **Add notifications - optional**. The left sidebar lists steps 1 through 7. Step 6, "Add tags", is highlighted with a blue circle. The main content area has a heading "Add notifications - optional" with a "Info" link. A note says "Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group." Below this is a button labeled "Add notification". At the bottom right are "Cancel", "Skip to review", "Previous", and "Next" buttons.

- Step 1 Choose launch template or configuration
- 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.

Add notification

Cancel **Skip to review** **Previous** **Next**

- Choose **Next** until the **Add tags - optional** page is displayed.

The screenshot shows the AWS EC2 Auto Scaling group creation wizard at Step 6: **Add tags - optional**. The left sidebar lists steps 1 through 7. Step 6, "Add tags", is highlighted with a blue circle. The main content area has a heading "Add tags - optional" with a "Info" link. A note says "Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched." Below this is a callout box with the text: "You optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group." To the right is a "Tags (0)" section with an "Add tag" button and a note "50 remaining". At the bottom right are "Cancel", "Previous", and "Next" buttons.

- Step 1 Choose launch template or configuration
- 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
- Add tags**
- Step 7 Review

Add tags - optional Info
Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

(i) You optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group.

Tags (0)

Add tag
50 remaining

Cancel **Previous** **Next**

- Choose **Add tag** and then configure the following:
 - Key:** Enter **Name**
 - Value - optional:** Enter **Inventory-App**

This tags the Auto Scaling group with a name, which also applies to the EC2 instances launched by the Auto Scaling group. This helps you identify which EC2 instances are associated with which application or with business concepts, such as cost centers.

Screenshot of the AWS EC2 Auto Scaling Groups 'Create Auto Scaling group' wizard Step 6: Add tags - optional.

Step 1: Choose launch template or configuration

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 tags - optional

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

Tags (1)

Key	Value - optional	Tag new instances
Name	Inventory-App	<input checked="" type="checkbox"/>

Add tag

49 remaining

Cancel **Previous** **Next**

- Choose **Next**.

Screenshot of the AWS EC2 Auto Scaling Groups 'Create Auto Scaling group' wizard Step 7: Review.

Step 1: Choose launch template or configuration

Group details

Auto Scaling group name: Inventory-ASG

Launch template

Launch template	Version	Description
Lab-launch-template	Default	version 1

Step 2: Choose instance launch options

Network

VPC: vpc-096d40eaacaf19a4

Availability Zones and subnets

Availability Zone	Subnet	Subnet CIDR range
us-west-2a	subnet-02ad7a020511da5c7	10.0.2.0/23
us-west-2b	subnet-076b1cb82d312b51e	10.0.4.0/23

Screenshot of the AWS EC2 Auto Scaling Groups 'Create Auto Scaling group' wizard Step 7: Review.

Step 3: Integrate with other services

Load balancing

Load balancer 1

Name	Type	Target group
Inventory-LB	Application/HTTP	Inventory-App

VPC Lattice integration options

VPC Lattice target groups:

-

Application Recovery Controller (ARC) zonal shift

ARC zonal shift: Disabled

Screenshot of the AWS EC2 Auto Scaling group creation wizard, Step 4: Configure group size and scaling policies.

Health checks

Health check type EC2	Health check grace period 300 seconds
--------------------------	--

Step 4: Configure group size and scaling policies

Group size

Desired capacity 2	Desired capacity type Units (number of instances)
-----------------------	--

Scaling

Minimum desired capacity 2	Maximum desired capacity 2
Target tracking policy -	

Instance maintenance policy

Replacement behavior No policy	Min healthy percentage -	Max healthy percentage -
-----------------------------------	-----------------------------	-----------------------------

Screenshot of the AWS EC2 Auto Scaling group creation wizard, Step 5: Add notifications.

Additional settings

Instance scale-in protection Disabled	Monitoring Enabled	Default instance warmup Disabled
--	-----------------------	-------------------------------------

Capacity Reservation preference

Preference Default	Capacity Reservation IDs -	Resource Groups -
-----------------------	-------------------------------	----------------------

Step 5: Add notifications

Notifications

No notifications

Step 6: Add tags

Tags (1)

Key	Value	Tag new instances
Name	Inventory-App	Yes

Buttons at the bottom: Preview code, Cancel, Previous, Create Auto Scaling group.

- Review the Auto Scaling group configuration for accuracy, and then choose **Create Auto Scaling group**.

Screenshot of the AWS EC2 Auto Scaling groups page.

Auto Scaling groups (1) Info

Last updated	Launch configurations	Launch templates	Actions	Create Auto Scaling group																
less than a minute ago																				
<table border="1"> <thead> <tr> <th>Name</th> <th>Launch template/configuration</th> <th>Instances</th> <th>Status</th> <th>Desired capacity</th> <th>Min</th> <th>Max</th> <th>Availability Zones</th> </tr> </thead> <tbody> <tr> <td>Inventory-ASG</td> <td>Lab-launch-template</td> <td>2</td> <td>-</td> <td>2</td> <td>2</td> <td>2</td> <td>us-west-2a, us-west-2b</td> </tr> </tbody> </table>					Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones	Inventory-ASG	Lab-launch-template	2	-	2	2	2	us-west-2a, us-west-2b
Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones													
Inventory-ASG	Lab-launch-template	2	-	2	2	2	us-west-2a, us-west-2b													

0 Auto Scaling groups selected

Select an Auto Scaling group

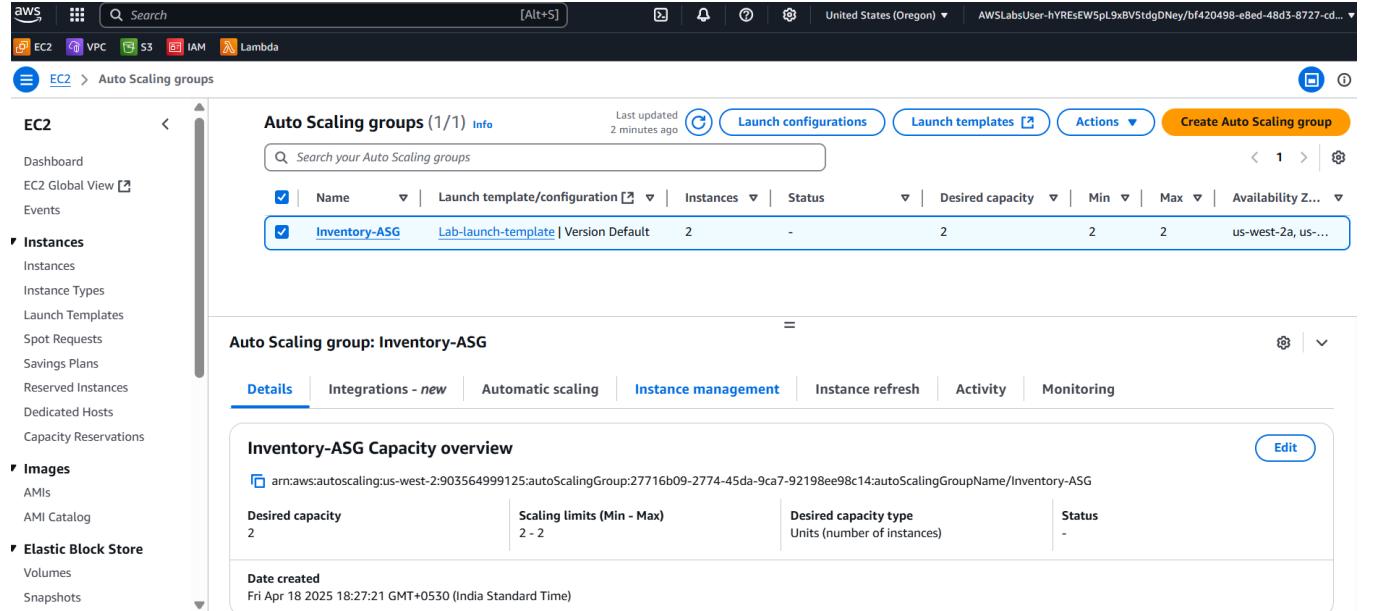
Left sidebar navigation:

- EC2
- Dashboard
- EC2 Global View
- Events
- Instances
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations
- Images
- AMIs
- AMI Catalog

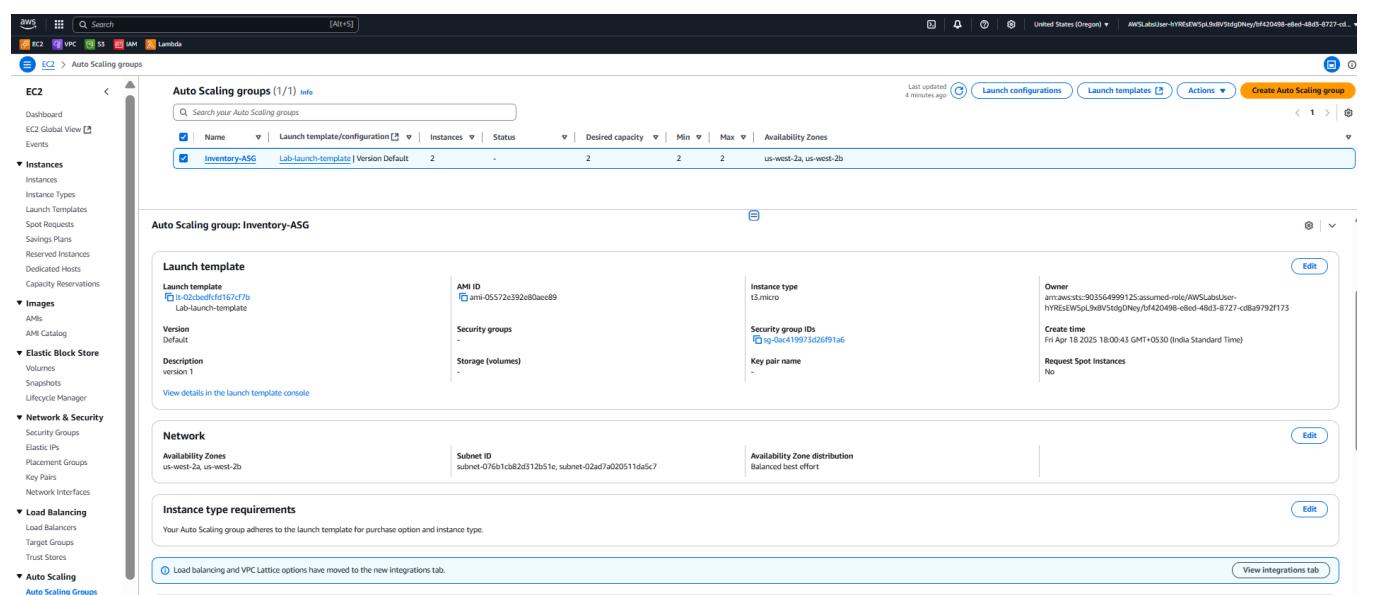
Your application will soon be running across two Availability Zones. Amazon EC2 Auto Scaling maintains the configuration even if an instance or Availability Zone fails.

Now that you have created your Auto Scaling group, you can verify that the group has launched your EC2 instances.

- Choose your Auto Scaling group.



The screenshot shows the AWS EC2 Auto Scaling Groups page. On the left, a sidebar lists various EC2 services: 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), and Elastic Block Store (Volumes, Snapshots). The main content area displays the 'Auto Scaling groups (1/1)' section. A table shows one group named 'Inventory-ASG' with a launch template 'Lab-launch-template'. The table includes columns for Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Availability Zones. The 'Inventory-ASG' row is selected. Below the table, the 'Auto Scaling group: Inventory-ASG' details are shown under the 'Instance management' tab. It includes sections for Capacity overview, Desired capacity (2), Scaling limits (Min - Max 2 - 2), Desired capacity type (Units (number of instances)), and Status (-). The Date created is Fri Apr 18 2025 18:27:21 GMT+0530 (India Standard Time).



This screenshot shows the same 'Auto Scaling group: Inventory-ASG' page but with more tabs visible: Details, Integrations - new, Automatic scaling, Instance management (selected), Instance refresh, Activity, and Monitoring. Under the 'Launch template' tab, it shows the launch template 'am-02cbedf167a7fb7b' with an AMI ID 'ami-05572ec392e03ace89', instance type 't3.micro', and security group 'sg-0ac419973d26f91a6'. The 'Network' tab shows availability zones 'us-west-2a, us-west-2b', subnet ID 'subnet-02ad7a020511da5c7', and availability zone distribution 'Balanced best effort'. The 'Instance type requirements' tab indicates that the group adheres to the launch template's purchase option and instance type. A note at the bottom states that load balancing and VPC Lattice options have moved to the new Integrations tab.

The screenshot shows the AWS Auto Scaling Groups page. On the left, there's a navigation sidebar with sections like EC2, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. Under Auto Scaling, 'Auto Scaling Groups' is selected. The main content area displays 'Auto Scaling groups (1/1)'. A table lists one group: 'Inventory-ASG' (Lab-launch-template | Version Default). The group has 2 instances, a desired capacity of 2, and is in the us-west-2a, us-west-2b availability zones. Below the table, the 'Auto Scaling group: Inventory-ASG' configuration is shown in tabs: 'Health check', 'Instance maintenance policy', 'Capacity Reservation preference', 'Advanced configurations', and 'Tags (1)'. The 'Activity' tab is not visible in this screenshot.

- Examine the **Group details** section to review information about the Auto Scaling group.
- Choose the **Activity** tab.

This screenshot is similar to the previous one, but the 'Activity' tab is now selected in the 'Auto Scaling group: Inventory-ASG' configuration section. The 'Activity notifications' and 'Activity history' sections are visible. The 'Activity history' section shows two entries:

Status	Description	Cause	Start time	End time
Successful	Launching a new EC2 instance: i-0f016c9c1a0bed4	At 2025-04-18T12:57:21Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2025-04-18T12:57:23Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2025 April 18, 06:27:25 PM +05:30	2025 April 18, 06:27:31 PM +05:30
Successful	Launching a new EC2 instance: i-09d3abf23dd808ad	At 2025-04-18T12:57:21Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2025-04-18T12:57:23Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2025 April 18, 06:27:24 PM +05:30	2025 April 18, 06:27:31 PM +05:30

The **Activity history** section maintains a record of events that have occurred in your Auto Scaling group. The Status column contains the status of your instances. When your instances are launching, the status column shows *PreInService*. After an instance is launched, the status changes to *Successful*.

- Choose the **Instance management** tab.

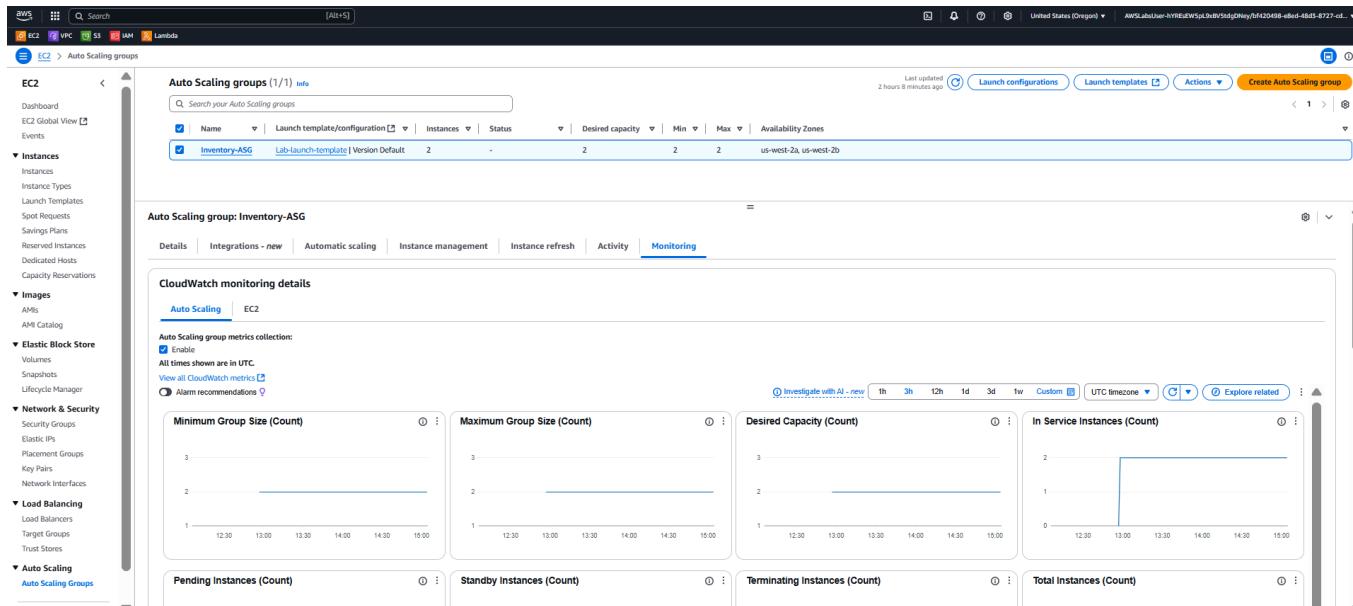
The screenshot shows the AWS EC2 Auto Scaling Groups page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, and Network & Security. The main area displays 'Auto Scaling groups (1/1) Info' for 'Inventory-ASG'. It lists one instance: 'i-030cb8bf23dd808ad' (InService, t3.micro, Lab-launch-template Version 1, us-west-2b, Healthy). Another instance, 'i-0f016e19cc7a0bed4', is also listed but is not visible in the screenshot.

Your Auto Scaling group has launched two EC2 instances, and they are in the *InService* lifecycle state. The Health status column shows the result of the EC2 instance health check on your instances.

Refresh: If your instances have not reached the *InService* state yet, you need to wait a few minutes. You can choose refresh to retrieve the current lifecycle state of your instances.

- Choose the **Monitoring** tab.

Here, you can review monitoring-related information for your Auto Scaling group and EC2 Instances.



The screenshot shows the AWS CloudWatch Metrics dashboard for an Auto Scaling group named "Inventory-ASG". The dashboard is divided into two main tabs: "Auto Scaling" and "EC2". The "Auto Scaling" tab is currently active, displaying CloudWatch monitoring details for the Auto Scaling group. The "EC2" tab displays metrics for the EC2 instances managed by the Auto Scaling group. The metrics shown include CPU Utilization (Percent), Disk Reads (Bytes), Disk Read Operations (Operations), Disk Write Operations (Operations), Network In (Bytes), Network Out (Bytes), and Status Check Failed (Any) (Count). The CPU Utilization chart shows a sharp spike from approximately 0.33% at 12:30 UTC to 24.4% at 13:00 UTC. The Disk Read Operations chart shows "No data available. Try adjusting the dashboard time range." The Disk Write Operations chart shows "No data available. Try adjusting the dashboard time range." The Network In chart shows 35.2M bytes. The Network Out chart shows 87.7k bytes. The Status Check Failed chart shows 1 failure.

Learn more: This page provides information about activity in your Auto Scaling group and the usage and health status of your instances. The Auto Scaling tab displays Amazon CloudWatch metrics about your Auto Scaling group, and the EC2 tab displays metrics for the EC2 instances managed by the Auto Scaling group. For more information, see [Monitor your Auto Scaling instances and groups](#).

You have now successfully created an Auto Scaling group, which maintains your application's availability and makes it resilient to instance or Availability Zone failures. Next, you test the high availability of the application.

Task 4: Test the application

In this task, you confirm that your web application is running and highly available.

- In the left navigation pane, choose **Target Groups**.
- Under **Name**, select **Inventory-App**.
- On the lower half of the page, choose the **Targets** tab.

The screenshot shows the AWS EC2 Target groups interface. The left sidebar includes options like AMIs, Volumes, Security Groups, Load Balancers, and Auto Scaling. The main area displays the 'Target groups (1/1)' section for 'Inventory-App'. It lists one target: 'arn:aws:elasticloadbalancing:us-west-2:9035649991...'. The 'Targets' tab is selected, showing a table of registered targets:

Instance ID	Name	Port	Zone	Health status	Health status details	Administr...	Override details
i-0f016e19cc7a0bed4	Inventory-App	80	us-west-2a (usw2-az1)	Healthy	-	No override	No override is currently active on t...
i-030cb8bf23dd808ad	Inventory-App	80	us-west-2b (usw2-az2)	Healthy	-	No override	No override is currently active on t...
i-029a444b99138ac3c	AppServer	80	us-west-2a (usw2-az1)	Healthy	-	No override	No override is currently active on t...

In the Registered targets section, there are three instances. This includes the two Auto Scaling instances named Inventory-App and the original instance you examined in Task 1, named AppServer.

The Health status column shows the results of the load balancer health check that you performed against the instances. In this task, you remove the original AppServer instance from the target group, leaving only the two instances managed by Amazon EC2 Auto Scaling.

- For the instance, select **AppServer**.

The screenshot shows the AWS EC2 Target groups interface, identical to the previous one but with a key difference: the 'AppServer' instance has been deregistered. The 'Targets' tab is still selected, and the table now shows only two registered targets:

Instance ID	Name	Port	Zone	Health status	Health status details	Administr...	Override details
i-0f016e19cc7a0bed4	Inventory-App	80	us-west-2a (usw2-az1)	Healthy	-	No override	No override is currently active on t...
i-030cb8bf23dd808ad	Inventory-App	80	us-west-2b (usw2-az2)	Healthy	-	No override	No override is currently active on t...

- To remove the instance from the load balancer's target group, choose **Deregister**.

A **Successfully deregistered 1 target.** message is displayed on top of the screen.

The screenshot shows the AWS EC2 Target groups page. A green success message at the top says "Successfully deregistered 1 target." Below it, the "Target groups (1/1)" section lists a single target group named "Inventory-App". The "Registered targets (3)" table shows three instances: "i-0f016e19cc7a0bed4" (Inventory-App, healthy), "i-030cb8bf23dd808ad" (Inventory-App, healthy), and "i-029a444b99138ac3c" (AppServer, draining). The "Health status details" column for the AppServer instance shows "Target deregistration is in progress".

The load balancer stops routing requests to a target as soon as it is deregistered. The Health status column for the AppServer instance displays a *draining* state, and the Health Status Details column displays *Target deregistration is in progress* until in-flight requests have completed. After a few minutes, the AppServer instance finishes deregistering, and only the two Auto Scaling instances remain on the list of registered targets.

Note: Deregistering the instance only detaches it from the load balancer. The AppServer instance continues to run indefinitely until you terminate it.

- Return to the Inventory Application tab in your web browser.

Note: If you closed the browser tab, you can reopen the inventory application by doing the following:

- In the left navigation pane, choose **Load Balancers**.
- Select the **Inventory-LB** load balancer.
- In the **Details** tab on the lower half of the page, copy the **DNS name** to your clipboard.

It should be similar to *Inventory-LB-xxxx.elb.amazonaws.com*.

- Open a new web browser tab, paste the DNS name from your clipboard, and press **Enter**.

The load balancer forwards your request to one of the EC2 instances. The bottom of the page displays the instance ID and Availability Zone.

Store	Item	Quantity
Puerto Rico	Amazon Echo	12
Paris	Amazon Dot	3
Detroit	Amazon Tap	5

+ Add Inventory

This page was generated by instance **i-0f016e19cc7a0bed4** in Availability Zone **us-west-2a**.

- **Refresh:** Refresh the page in your web browser a few times. The instance ID and Availability Zone sometimes change between the two instances.

Store	Item	Quantity
Puerto Rico	Amazon Echo	12
Paris	Amazon Dot	3
Detroit	Amazon Tap	5

+ Add Inventory

This page was generated by instance **i-0f016e19cc7a0bed4** in Availability Zone **us-west-2a**.

Store	Item	Quantity
Puerto Rico	Amazon Echo	12
Paris	Amazon Dot	3
Detroit	Amazon Tap	5

+ Add Inventory

This page was generated by instance **i-030cb8bf23dd808ad** in Availability Zone **us-west-2b**.

Note: The flow of information is as follows:

- You send the request to the Application Load Balancer, which resides in the *public* subnets. The public subnets are connected to the internet.
- The Application Load Balancer chooses one of the EC2 instances that reside in the *private* subnets and forwards the request to the instance.
- The EC2 instance then returns the web page to the Application Load Balancer, which returns the page to your web browser.

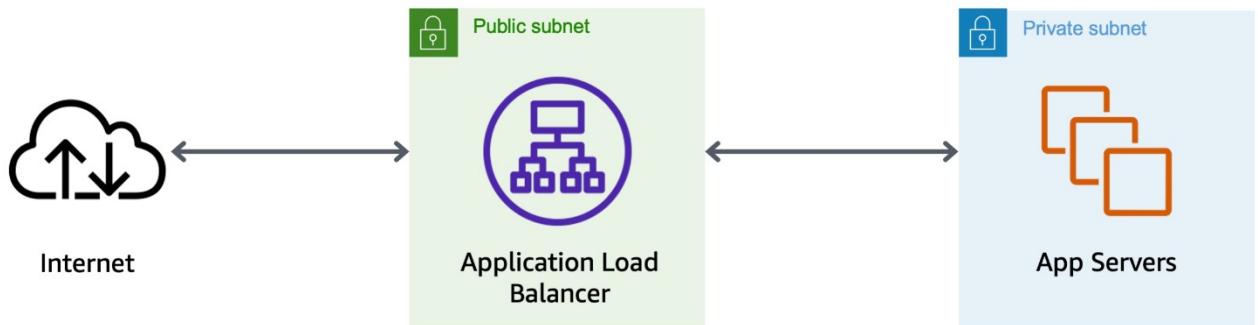


Image description: The preceding image depicts the flow of information for this web application

You have now confirmed that Amazon EC2 Auto Scaling successfully launched two new Inventory-App instances across two Availability Zones, and you deregistered the original AppServer instance from the load balancer. The Auto Scaling group maintains high availability for your application in the event of failure. Next, you simulate a failure by terminating one of the Inventory-App instances managed by Amazon EC2 Auto Scaling.

Task 5: Test high availability of the application tier

In this task, you test the high availability configuration of your application by terminating one of the EC2 instances.

- Return to the **EC2 Management Console**, but do not close the application tab. You return to it in later tasks.

The screenshot shows the AWS EC2 Management Console dashboard. On the left, there's a navigation pane with links for Dashboard, Instances, Images, and Elastic Block Store. The main area displays various resources: Instances (running) 1, Auto Scaling Groups 0, Capacity Reservations 0, Dedicated Hosts 0, Elastic IPs 1, Instances 1, Key pairs 0, Load balancers 1, Placement groups 0, Security groups 6, Snapshots 0, and Volumes 1. Below this, there's a 'Launch instance' section with a 'Launch instance' button, a 'Service health' section showing 'Status' as 'operating normally', and an 'Explore AWS' section with various promotional links.

- In the left navigation pane, choose **Instances**.

The screenshot shows the 'Instances' page in the AWS EC2 Management Console. The left navigation pane is expanded to show the 'Instances' section. The main table lists three instances: 'Inventory-App' (two entries, both running), and 'AppServer' (one entry, running). The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IP. A 'Select an instance' dropdown menu is open at the bottom of the table.

Now, terminate one of the web application instances to simulate a failure.

- Choose one of the Inventory-App instances. (It does not matter which one you choose.)
- Choose **Instance State** and then choose **Terminate (delete) instance**.

Screenshot of the AWS EC2 Instances page showing three running instances: Inventory-App (t3.micro), Inventory-App (t3.micro), and AppServer (t3.micro). The instance details for 'Inventory-App' are shown, including its Public IPv4 address (10.0.5.164) and Private IP DNS name (ip-10-0-5-164.us-west-2.compute.internal).

A context menu is open over the first 'Inventory-App' instance, with the 'Terminate (delete) instance' option highlighted.

The 'Terminate (delete)' confirmation dialog is displayed, asking if you're sure you want to terminate the instance. It notes that on an EBS-backed instance, the default action is to delete the root EBS volume. A warning message states: "⚠️ On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost."

- Choose **Terminate (delete)**.

Screenshot of the AWS EC2 Instances page showing the termination process. The instance 'Inventory-App' (i-030cb8bf23dd808ad) is listed with its status as 'Shutting-down'. A success message at the top indicates: 'Successfully initiated termination (deletion) of i-030cb8bf23dd808ad'.

The instance details for 'Inventory-App' are shown again, now indicating it is 'Shutting-down'.

Instances (1/3) Info								
Find Instance by attribute or tag (case-sensitive)		Last updated less than a minute ago	Connect	Instance state	Actions	Launch instances		
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
<input checked="" type="checkbox"/> Inventory-App	i-030cb8bf23dd808ad	Terminated	t3.micro	-	View alarms +	us-west-2b	-	-
<input type="checkbox"/> Inventory-App	i-0f016e19cc7a0bed4	Running	t3.micro	3/3 checks passed	View alarms +	us-west-2a	-	-
<input type="checkbox"/> AppServer	i-029a444b99138ac3c	Running	t3.micro	3/3 checks passed	View alarms +	us-west-2a	-	-

After a short period of time, the load balancer health checks will notice that the instance is not responding and automatically route all incoming requests to the remaining instance.

- Leaving the console open, switch to the Inventory Application tab in your web browser and refresh the page several times.

The Availability Zone shown at the bottom of the page stays the same. Even though an instance has failed, your application remains available.

Store	Item	Quantity
Puerto Rico	Amazon Echo	12
Paris	Amazon Dot	3
Detroit	Amazon Tap	5

[+ Add Inventory](#)

This page was generated by instance i-0f016e19cc7a0bed4 in Availability Zone us-west-2a.

After a few minutes, Amazon EC2 Auto Scaling also detects the instance failure. You configured Amazon EC2 Auto Scaling to keep two instances running, so Amazon EC2 Auto Scaling automatically launches a replacement instance.

- Refresh:** Return to the **EC2 Management Console**. Reload the list of instances using the refresh button every 30 seconds until a new EC2 instance named *Inventory-App* appears.

Instances (1/4) Info								
Find Instance by attribute or tag (case-sensitive)		Last updated 4 minutes ago	Connect	Instance state	Actions	Launch instances		
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
<input checked="" type="checkbox"/> Inventory-App	i-030cb8bf23dd808ad	Terminated	t3.micro	-	View alarms +	us-west-2b	-	-
<input type="checkbox"/> Inventory-App	i-09f1f8085d7288771	Initializing	t3.micro	-	View alarms +	us-west-2b	-	-
<input type="checkbox"/> Inventory-App	i-0f016e19cc7a0bed4	Running	t3.micro	3/3 checks passed	View alarms +	us-west-2a	-	-
<input type="checkbox"/> AppServer	i-029a444b99138ac3c	Running	t3.micro	3/3 checks passed	View alarms +	us-west-2a	-	-

The newly launched instance displays *Initializing* under the Status check column. After a few minutes, the health check for the new instance should become *healthy*, and the load balancer resumes distributing traffic between two Availability Zones.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Inventory-App	i-030cb8bf23dd808ad	Running	t3.micro		View alarms +	us-west-2b	-
Inventory-App	i-09f1f8085d7288771	Running	t3.micro		View alarms +	us-west-2b	-
Inventory-App	i-0f016e19cc7a0bed4	Running	t3.micro		View alarms +	us-west-2a	-
AppServer	i-029a444b9913ac3c	Running	t3.micro		View alarms +	us-west-2a	-

- **Refresh:** Return to the Inventory Application tab and refresh the page several times. The instance ID and Availability Zone change as you refresh the page.

Store	Item	Quantity
Puerto Rico	Amazon Echo	12
Paris	Amazon Dot	3
Detroit	Amazon Tap	5

+ Add Inventory

Store	Item	Quantity
Puerto Rico	Amazon Echo	12
Paris	Amazon Dot	3
Detroit	Amazon Tap	5

+ Add Inventory

This page was generated by instance **i-0f016e19cc7a0bed4** in Availability Zone **us-west-2a**.

This page was generated by instance **i-09f1f8085d7288771** in Availability Zone **us-west-2b**.

This demonstrates that your application is now highly available.

You have successfully verified that your application is highly available.

Task 6: Configure high availability of the database tier

You verified that the application tier was highly available in the previous task. However, the Aurora database is still operating from only one database instance.

Task 6.1: Configure the database to run across multiple Availability Zones

In this task, you make the Aurora database highly available by configuring it to run across multiple Availability Zones.

- At the top of the console, in the search bar, search for and choose **RDS**.

The screenshot shows the AWS RDS Dashboard. On the left, the navigation pane includes links for Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, and Event subscriptions. The main panel displays 'Resources' with sections for DB Instances (1/40), DB Clusters (1/40), Reserved instances (0/40), Snapshots (0), Recent events (3), and Event subscriptions (0/20). It also lists Parameter groups (4), Option groups (2), Subnet groups (1/50), and Supported platforms (VPC). A 'Create database' button is present. To the right, there are 'Recommended services' like Amazon Augmented AI, Amazon AppFlow, AWS Firewall Manager, Cloud9, and AWS Well-Architected Tool, and a 'Recommended for you' section for testing DR strategies.

- In the left navigation pane, choose **Databases**.

The screenshot shows the 'Databases' page under the RDS service. The left navigation pane remains the same. The main area displays a table titled 'Databases (2)' with columns for DB identifier, Status, Role, Engine, Region & AZ, Size, Recommendations, CPU, and Current. Two rows are listed: 'inventory-cluster' (Regional cluster, Aurora MySQL, us-west-2, 1 instance) and 'inventory-primary' (Writer instance, Aurora MySQL, us-west-2b, db.t3.medium). A 'Create database' button is located at the bottom right of the table.

- Locate the row that contains the **inventory-primary** value.
- In the fifth column, labelled **Region & AZ**, note in which Availability Zone the primary is located.

Aurora and RDS > Databases

Databases (2)

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations	CPU	Current
inventory-cluster	Available	Regional cluster	Aurora MySQL	us-west-2	1 instance	-	-	-
inventory-primary	Available	Writer instance	Aurora MySQL	us-west-2b	db.t3.medium	9.63%		

Caution: In the following steps you create an additional instance for the database cluster. For true high-availability architecture, the second instance must be located in an Availability Zone that is *different* from that of the primary instance.

- Select the **inventory-cluster** radio button associated with your Aurora database cluster.
- Choose **Actions** and then choose **Add reader**.

Aurora and RDS > Databases

Databases (2)

DB identifier	Status	Role	Engine	Region & AZ	Size
inventory-cluster	Available	Regional cluster	Aurora MySQL	us-west-2	1 inst
inventory-primary	Available	Writer instance	Aurora MySQL	us-west-2b	db.t3

Actions

- Stop temporarily
- Delete
- Set up EC2 connection
- Set up Lambda connection
- Migrate data from EC2 database - new
- Add AWS Region
- Add reader**
- Create cross-Region read replica
- Create blue/green deployment
- Create clone
- Promote
- Take snapshot
- Restore to point in time
- Backtrack
- Export to Amazon S3
- Add replica auto scaling
- Create zero-ETL integration
- Create RDS Proxy
- Create ElastiCache cluster

Add reader

You are creating a replica DB instance from a source DB instance. This new DB instance will have the source DB instance's DB security groups and DB parameter groups.

Settings

Aurora replica source
Source DB cluster identifier: **inventory-primary** (Role: Writer instance Parent: inventory-cluster)

DB instance identifier
This is the unique key that identifies a DB instance. This parameter is stored as a lowercase string (for example, mydbinstance).

Instance configuration
The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class | [Info](#)

Hide filters

Include previous generation classes

Serverless v2

Memory optimized classes (includes r classes)

Burstable classes (includes t classes)

db.t3.medium

- In the **Settings** section, configure the following:
 - DB instance identifier:** Enter `inventory-replica`

The screenshot shows the 'Add reader' configuration page for an Aurora replica. In the 'Aurora replica source' section, 'inventory-primary' is selected as the source DB cluster identifier. In the 'DB instance identifier' section, 'inventory-replica' is entered.

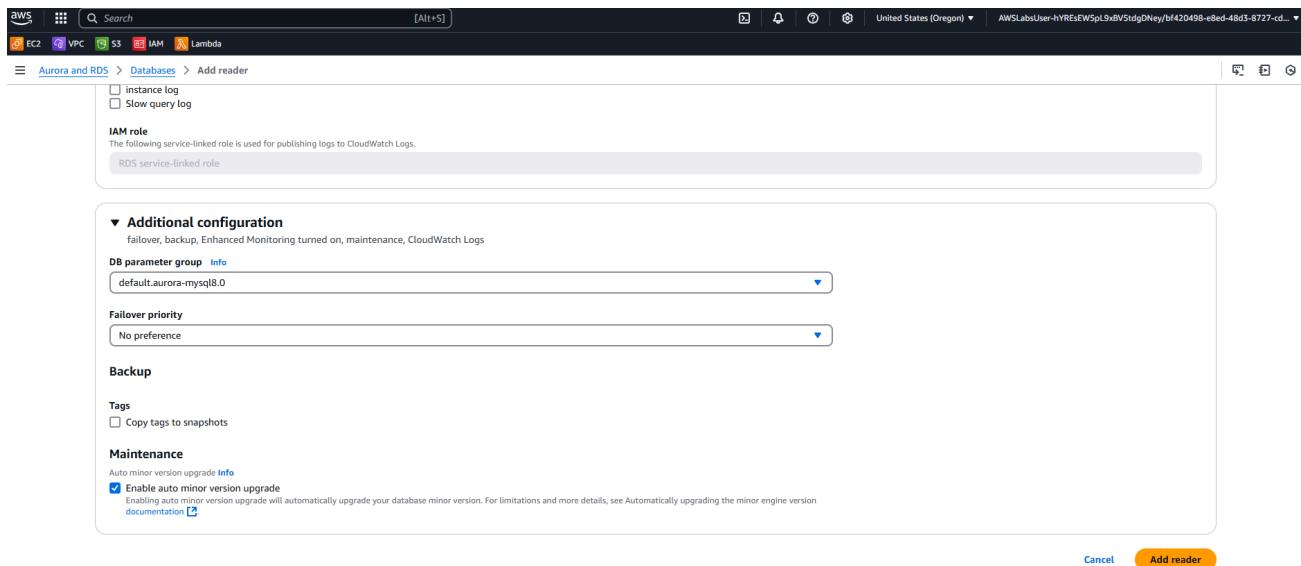
- In the **Connectivity** section, under **Availability Zone**, select a *different Availability Zone* from the one you noted above where the inventory-primary is located. (**choose us-west-2a**, as *us-west-2b* is the availability for inventory-primary)

The screenshot shows the 'Connectivity' section of the 'Add reader' configuration page. 'Not publicly accessible' is selected for public access. 'us-west-2a' is selected as the availability zone.

- In the **Monitoring** section under **Enhanced Monitoring**, uncheck **Enable Enhanced monitoring**

The screenshot shows the 'Monitoring' section of the 'Add reader' configuration page. Under 'Enhanced Monitoring', the 'Enable Enhanced monitoring' checkbox is unchecked.

- At the bottom of the page, choose **Add reader**.



A new DB identifier named *inventory-replica* appears on the list, and its status is *Creating*. This is your Aurora Replica instance. You can continue to the next task without waiting.

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations	CPU	Current activity	Mail
inventory-cluster	Available	Regional c...	Aurora My...	us-west-2	2 instances	-	-	-	non-
inventory-primary	Available	Writer ins...	Aurora My...	us-west-2b	db.t3.medium	10.26%	2 Selects/Sec	non-	
inventory-replica	Creating	Reader ins...	Aurora My...	us-west-2a	db.t3.medium	-	-	-	non-

Learn more: When your Aurora Replica finishes launching, your database is deployed in a highly available configuration across multiple Availability Zones. This does not mean that the database is *distributed* across multiple instances. Although both the primary DB instance and the Aurora Replica access the same shared storage, only the primary DB instance can be used for writes. Aurora Replicas have two main purposes. You can issue queries to them to scale the read operations for your application. You typically do so by connecting to the reader endpoint of the cluster. That way, Aurora can spread the load for read-only connections across as many Aurora Replicas as you have in the cluster. Aurora Replicas also help to increase availability. If the writer instance in a cluster becomes unavailable, Aurora automatically promotes one of the reader instances to take its place as the new writer. For more information, see [Replication with Amazon Aurora](#).

While the Aurora Replica launches, continue to the next task to configure high availability for the NAT gateway, and then return to the Amazon RDS console in the final task to confirm high availability of the database after the creation of the replica is complete.

You have successfully configured high availability for the database tier.

Task 7: Make the NAT gateway highly available

In this task, you make the NAT gateway highly available by launching another NAT gateway in the second Availability Zone.

The Inventory-App servers are deployed in private subnets across two Availability Zones. If they need to access the internet (for example, to download data), the requests must be redirected through a *NAT gateway* (located in a public subnet). The current architecture has only one NAT gateway in Public Subnet 1, and all of the Inventory-App servers use this NAT gateway to reach the internet. This means that if Availability Zone 1 failed, none of the application servers would be able to communicate with the internet. Adding a second NAT gateway in Availability Zone 2 ensures that resources in private subnets can still reach the internet even if Availability Zone 1 fails.

The resulting architecture shown in the following diagram is highly available:

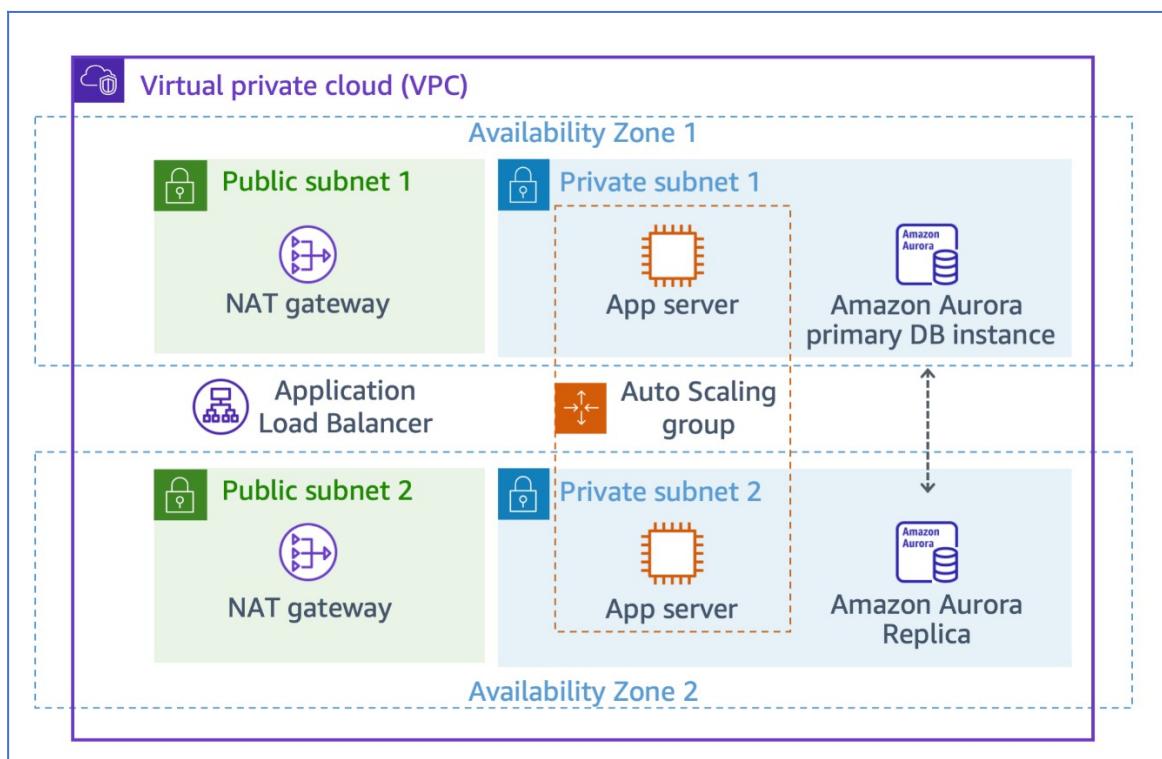


Image description: The preceding image depicts, how to enhance fault tolerance and ensure continuous internet access for Inventory-App servers in private subnets across two Availability Zones, it is recommended to add a second NAT gateway in Availability Zone 2, complementing the existing one in Public Subnet 1.

Task 7.1: Create a second NAT gateway

- At the top of the console, in the search bar, search for and choose **VPC**.

The screenshot shows the AWS VPC dashboard. In the left navigation pane, under 'Virtual private cloud', 'NAT gateways' is selected. The main area displays 'Resources by Region' with sections for VPCs, Subnets, Route Tables, Internet Gateways, and NAT Gateways. Each section shows the number of resources and a link to 'See all regions'. On the right side, there are three boxes: 'Service Health' (with a link to 'View complete service health details'), 'Settings' (with links to 'Block Public Access', 'Zones', and 'Console Experiments'), and 'Additional Information' (with links to 'VPC Documentation', 'All VPC Resources', 'Forums', and 'Report an Issue').

- In the left navigation pane, choose **NAT gateways**.

The existing NAT gateway is displayed. Now create one for the other Availability Zone.

The screenshot shows the 'NAT gateways (1/1)' page. The table lists one existing NAT gateway: 'nat-0f5d0cd41046370b6', which is 'Public' and 'Available'. The 'Actions' button is highlighted in orange. Below the table, the 'Details' tab is selected, showing the gateway's ID, ARN, VPC, and subnet information. Other tabs include 'Secondary IPv4 addresses', 'Monitoring', and 'Tags'.

- Choose **Create NAT gateway** and configure the following:
 - Name - optional:** Enter **my-nat-gateway**.
 - Subnet:** Select **Public Subnet 2** from the dropdown menu.

Screenshot of the AWS VPC NAT gateways creation page:

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.

- Choose **Allocate Elastic IP**.

Screenshot of the AWS VPC NAT gateways creation page after allocation:

Elastic IP address 100.21.38.30 (eipalloc-04d5892ff77a63e36) allocated.

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.

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="my-nat-gateway"/> <input type="button" value="Remove"/>
<input type="button" value="Add new tag"/>	

You can add 49 more tags.

- Choose **Create NAT gateway**.

A **NAT gateway nat-xxxxxxxx | my-nat-gateway was created successfully.** message is displayed on top of the screen.

The screenshot shows the AWS VPC dashboard with the 'NAT gateways' section selected. A success message at the top states: 'NAT gateway nat-01262bc27e14dd82e | my-nat-gateway was created successfully.' Below this, the 'nat-01262bc27e14dd82e / my-nat-gateway' details are displayed. The 'Details' section includes:

- NAT gateway ID:** nat-01262bc27e14dd82e
- Connectivity type:** Public
- State:** Pending
- NAT gateway ARN:** arn:aws:ec2:us-west-2:903564999125:natgateway/nat-01262bc27e14dd82e
- Primary public IPv4 address:** -
- Primary private IPv4 address:** 10.0.1.241
- Subnet:** subnet-0e04f6b8017c1b988 / Public Subnet 2
- Created:** Saturday 19 April 2025 at 00:12:36 GMT+5:30
- State message:** Info -
- Primary network interface ID:** eni-060d2f8d62b3bd841
- Deleted:** -

The 'Secondary IPv4 addresses' tab is selected, showing one entry: Private IP address 10.0.1.241 associated with Network interface ID eni-060d2f8d62b3bd841.

Task 7.2: Create and configure a new route table

Now, create a new route table for Private Subnet 2 that redirects traffic to the new NAT gateway.

- In the left navigation pane, choose **Route tables**.

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. The 'Route tables (4)' table is displayed, showing the following data:

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
-	rtb-052cd8367b026311b	-	-	Yes	vpc-096ebcf
Private Route Table 1	rtb-038ef0f2c47986da6	2 subnets	-	No	vpc-096d40e
-	rtb-009c9cc9e823aa642	-	-	Yes	vpc-096d40e
Public Route Table	rtb-0517ea1f76e617d66	2 subnets	-	No	vpc-096d40e

A 'Create route table' button is visible in the top right corner.

- Choose **Create route table** and configure the following:
 - Name - optional:** Enter **Private Route Table 2**.
 - VPC:** Select **Lab VPC** from the dropdown menu.

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.

Private Route Table 2

VPC
The VPC to use for this route table.

vpc-096d40eaacaf19a4 (Lab 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 Remove

Add new tag

You can add 49 more tags.

Cancel **Create route table**

- Choose **Create route table**.

A **Route table rtb-xxxxxxx | Private Route Table 2 was created successfully.** message is displayed on top of the screen.

Route table rtb-0c51a86478523ad8f | Private Route Table 2 was created successfully.

rtb-0c51a86478523ad8f / Private Route Table 2

Details Info

Route table ID rtb-0c51a86478523ad8f	Main <input checked="" type="checkbox"/> No	Explicit subnet associations —	Edge associations —
VPC vpc-096d40eaacaf19a4 Lab VPC	Owner ID 903564999125		

Routes (1) **Edit routes**

Destination	Target	Status	Propagated
10.0.0.20	local	Active	No

Details for the newly created route table are displayed. There is currently one route, which directs all traffic *locally*. Now, add a route to send internet-bound traffic through the new NAT gateway.

- Choose **Edit routes**.

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/20	local	Active	No

Add route Cancel Preview Save changes

- Choose **Add route** and configure the following:
 - Destination:** Enter **0.0.0.0/0**
 - Target:** Choose **NAT Gateway > my-nat-gateway**.

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/20	local	Active	No
0.0.0.0/0	NAT Gateway	-	No

Add route Remove Cancel Preview Save changes

- Choose **Save changes**.

A **Updated routes for rtb-xxxxxxxxxxxx / Private Route Table 2 successfully.** message is displayed on top of the screen.

VPC dashboard

rtb-0c51a86478523ad8f / Private Route Table 2

Updated routes for rtb-0c51a86478523ad8f / Private Route Table 2 successfully

Details

Route table ID rtb-0c51a86478523ad8f	Main No	Explicit subnet associations -	Edge associations -
VPC vpc-096d40eaacaf19a4 Lab VPC	Owner ID 903564999125		

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	nat-01262bc27e14dd82e	Active	No
10.0.0.0/20	local	Active	No

Both Edit routes

You have created the route table and configured it to route internet-bound traffic through the new NAT gateway. Next, associate the route table with Private Subnet 2.

Task 7.3: Configure routing for Private Subnet 2

- Choose the **Subnet associations** tab

The screenshot shows the AWS VPC Route Tables page. The left sidebar is expanded to show the 'Virtual private cloud' section, specifically the 'Route tables' subsection. The main content area displays the details for 'rtb-0c51a86478523ad8f / Private Route Table 2'. The 'Subnet associations' tab is selected. Below it, the 'Explicit subnet associations' section is shown, which currently contains no associations. A button labeled 'Edit subnet associations' is visible at the top right of this section.

- Choose **Edit subnet associations**.

The screenshot shows the 'Edit subnet associations' page for the route table 'rtb-0c51a86478523ad8f'. The top navigation bar includes the VPC icon, Route tables link, and the specific route table ID. The main content area is titled 'Edit subnet associations' and contains the sub-header 'Change which subnets are associated with this route table.' Below this is a table titled 'Available subnets (4)'. The table lists four subnets: 'Private Subnet 2' (subnet-076b1cb82d312b51e, 10.0.4.0/23), 'Public Subnet 1' (subnet-02365a081971d24c9, 10.0.0.0/24), 'Private Subnet 1' (subnet-02ad7a020511da5c7, 10.0.2.0/23), and 'Public Subnet 2' (subnet-0e04fb8017c1b988, 10.0.1.0/24). The 'Route table ID' column shows the route table for each subnet. At the bottom right of the table are 'Cancel' and 'Save associations' buttons.

- Select **Private Subnet 2**.

Available subnets (1/4)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
Private Subnet 2	subnet-076b1cb82d312b51e	10.0.4.0/23	-	rtb-038ef0f2c47986da6 / Private Rout...
Public Subnet 1	subnet-02365a081971d24c9	10.0.0.0/24	-	rtb-0517ea1f76e617d66 / Public Rout...
Private Subnet 1	subnet-02adfa020511da5c7	10.0.2.0/23	-	rtb-038ef0f2c47986da6 / Private Rout...
Public Subnet 2	subnet-0e4ff6b8017c1b988	10.0.1.0/24	-	rtb-0517ea1f76e617d66 / Public Rout...

Selected subnets

- subnet-076b1cb82d312b51e / Private Subnet 2 X

Actions

Save associations

- Choose **Save associations**.

A Updated routes for rtb-xxxxxxxxxxxx / Private Route Table 2 successfully.

rtb-0c51a86478523ad8f / Private Route Table 2

Details

Route table ID rtb-0c51a86478523ad8f	Main <input type="checkbox"/> No	Explicit subnet associations subnet-076b1cb82d312b51e / Private Subnet 2
VPC vpc-096d40eaacafdf19a4 Lab VPC	Owner ID 903564999125	Edge associations -

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	nat-01262bc27e14dd82e	Active	No
10.0.0.0/20	local	Active	No

Internet-bound traffic from Private Subnet 2 is now sent to the NAT gateway in the same Availability Zone.

Your NAT gateways are now highly available. A failure in one Availability Zone does not impact traffic in the other Availability Zone.

You have successfully verified that your NAT gateways are highly available.

Task 8: Force a failover of the Aurora database

In this task, to demonstrate that your database is capable or performing a failover, you force the cluster to perform a failover to the Aurora Read Replica instance you created in an earlier task.

- At the top of the console, in the search bar, search for and choose **RDS**.

The screenshot shows the Aurora and RDS dashboard. On the left, a navigation pane lists options like Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, and Event subscriptions. The main area displays 'Resources' for the US West (Oregon) region, showing DB Instances (2/40), Parameter groups (4), DB Clusters (1/40), Reserved instances (0/40), Snapshots (0), and Events (5). A 'Create database' button is available. To the right, 'Recommended services' include Amazon Augmented AI, Amazon AppFlow, AWS Firewall Manager, Cloud9, and AWS Well-Architected Tool.

- In the left navigation pane, choose **Databases**.

The screenshot shows the 'Databases' page with three entries: 'inventory-cluster' (Regional cluster, Aurora MySQL, us-west-2, 2 instances), 'inventory-primary' (Writer instance, Aurora MySQL, us-west-2b, db.t3.medium), and 'inventory-replica' (Reader instance, Aurora MySQL, us-west-2a, db.t3.medium). The 'inventory-primary' instance is highlighted with a blue border.

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations	CPU	Current activity
inventory-cluster	Available	Regional cluster	Aurora MySQL	us-west-2	2 instances	-	-	-
inventory-primary	Available	Writer instance	Aurora MySQL	us-west-2b	db.t3.medium	11.29%	2 Selects	11.24%
inventory-replica	Available	Reader instance	Aurora MySQL	us-west-2a	db.t3.medium	11.24%	2 Selects	11.24%

Caution: Verify that the *inventory-replica* DB instance status is changed to *Available* before continuing to the next step.

- For the DB identifier, select the **inventory-primary** DB identifier associated with your Aurora primary DB instance.

The screenshot shows the 'Databases' page with the 'inventory-primary' instance selected, indicated by a blue border around its row. The other two instances are shown in grey.

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations	CPU	Current activity
inventory-cluster	Available	Regional cluster	Aurora MySQL	us-west-2	2 instances	-	-	-
inventory-primary	Available	Writer instance	Aurora MySQL	us-west-2b	db.t3.medium	11.29%	2 Selects	11.24%
inventory-replica	Available	Reader instance	Aurora MySQL	us-west-2a	db.t3.medium	11.24%	2 Selects	11.24%

Note: The primary DB instance with DB identifier **inventory-primary** currently displays *Writer* under the Role column. This is the only database node in the cluster that can currently be used for writes.

- Choose **Actions**.

Aurora and RDS > Databases

Databases (3)

DB identifier	Status	Role	Engine	Region ...	Size
inventory-cluster	Available	Regional cluster	Aurora MySQL	us-west-2	2 instances
inventory-primary	Available	Writer instance	Aurora MySQL	us-west-2b	db.t3.medium
inventory-replica	Available	Reader instance	Aurora MySQL	us-west-2a	db.t3.medium

Actions ▾

Create database ▾

Reboot
Delete
Failover
Set up EC2 connection
Set up Lambda connection
Migrate data from EC2 database - new
Take snapshot
Create ElastiCache cluster

Current activity

11.29% 2 Selects

11.24% 2 Selects

- Choose **Failover**.

The RDS console displays the **Failover DB Cluster** page.

Aurora and RDS > Databases

Databases (3)

Failover DB Cluster

Are you sure you want to failover inventory-cluster?

Cancel Failover

- Choose **Failover**.

The *inventory-cluster* status is now **Failing over**.

Aurora and RDS > Databases

Failing over DB cluster inventory-cluster.

Databases (3)

DB identifier	Status	Role	Engine	Region ...	Size
inventory-cluster	Available	Regional cluster	Aurora MySQL	us-west-2	2 instances
inventory-primary	Available	Writer instance	Aurora MySQL	us-west-2b	db.t3.medium
inventory-replica	Available	Reader instance	Aurora MySQL	us-west-2a	db.t3.medium

Aurora and RDS

Databases (3)

DB identifier	Status	Role	Engine	Region	Size	Recommendations	CPU
inventory-cluster	Available	Regional cluster	Aurora MySQL	us-west-2	2 instances	-	
inventory-primary	Available	Reader instance	Aurora MySQL	us-west-2b	db.t3.medium	11.29%	
inventory-replica	Available	Writer instance	Aurora MySQL	us-west-2a	db.t3.medium	11.24%	

- From the navigation menu on the left, choose **Events**.

Events (11)

Source	Type	Time	Message
inventory-cluster	Clusters	April 19, 2025, 00:46 (UTC+05:30)	Completed failover to DB instance: inventory-replica
inventory-primary	Instances	April 19, 2025, 00:45 (UTC+05:30)	DB instance restarted
inventory-replica	Instances	April 19, 2025, 00:45 (UTC+05:30)	DB instance restarted
inventory-primary	Instances	April 19, 2025, 00:45 (UTC+05:30)	A new writer was promoted. Restarting database as a reader.
inventory-replica	Instances	April 19, 2025, 00:45 (UTC+05:30)	DB instance shutdown
inventory-cluster	Clusters	April 19, 2025, 00:45 (UTC+05:30)	Started cross AZ failover to DB instance: inventory-replica
inventory-replica	Instances	April 18, 2025, 23:57 (UTC+05:30)	DB instance created
inventory-replica	Instances	April 18, 2025, 23:53 (UTC+05:30)	Cluster topology is updated.
inventory-primary	Instances	April 18, 2025, 13:37 (UTC+05:30)	DB instance created
inventory-primary	Instances	April 18, 2025, 13:34 (UTC+05:30)	Cluster topology is updated.
inventory-cluster	Clusters	April 18, 2025, 13:32 (UTC+05:30)	DB cluster created

- Review the logs as the failover is occurring. Notice that the Read replica instance is shutdown, promoted to the writer and then rebooted. When the reboot of the read replica is completed, then the *inventory-primary* is rebooted.

Observe that the application continues to function correctly after the failover.

You have successfully verified that your database can successfully complete a failover and is highly available.

Conclusion

You now have successfully completed the following:

- Created an Amazon EC2 Auto Scaling group and registered it with an Application Load Balancer spanning across multiple Availability Zones.
- Created a highly available Aurora DB cluster.
- Modified an Aurora DB cluster to be highly available.
- Modified an Amazon VPC configuration to be highly available using redundant NAT gateways.
- Confirmed your database can perform a failover to a read replica instance.