



ACA Module 9 Lab: Creating a Highly Available Environment

COS 20019- Cloud Computing Architecture

Nguyen Manh Dung

30/6/2023

So this all my step to finish ACA Module 9 Lab with explanation with all the screenshots included a requirement of the project

Task 1: Inspecting your VPC

Review the configuration of the VPC that was created for this lab.

The screenshot shows two browser tabs side-by-side. The left tab is from AWS Academy, displaying a lab assignment titled "Module 9 Guided Lab - Creating a Highly Available Environment". It includes instructions for arranging the AWS Management Console and a list of components for the lab. The right tab is the "VPC Management Console" showing the "Subnets (1/4) Info" page. It lists four subnets under a single VPC, each with its CIDR range and other details. Below the subnet table is a "Details" section for a specific subnet.

Task2: Creating an Application Load Balancer

Create Security Group with below configuration

Security group name: Inventory-LB

Description: Enable web access to load balancer

VPC: Remove the default VPC by choosing the X to the right of it. Then select **Lab VPC**.

Under **Inbound rules**, choose Add rule and configure as described:

Type: HTTP

Source: Anywhere-IPv4

Still under Inbound rules, choose Add rule again and configure:

Type: HTTPS

Source: Anywhere-IPv4

ACAv2EN... > Assignments
Module 9 Guided Lab - Creating a Highly Available Environment

Due No Due Date Points 100 Submitting an external tool

Submit Details AWS Start Lab End Lab 2:15 Instructions Grades Actions

EN_US

25. In the **Security groups** section, select the **Create a new security group** hyperlink. This opens a new browser tab. Configure the new security group settings:

- o **Security group name:** Inventory-LB
- o **Description:** Enable web access to load balancer
- o **VPC:** Remove the default VPC by choosing the X to the right of it. Then select **Lab VPC**.

26. Under **Inbound rules**, choose **Add rule** and configure as described:

- o **Type:** HTTP
- o **Source:** Anywhere-IPv4

27. Still under **Inbound rules**, choose **Add rule** again and configure:

- o **Type:** HTTPS
- o **Source:** Anywhere-IPv4

Load balancers | EC2 Management

Inventory-LB

Name cannot be edited after creation.

Description: Enable web access to load balancer

VPC: vpc-035808e571e80413f

Inbound rules

Inbound rule 1

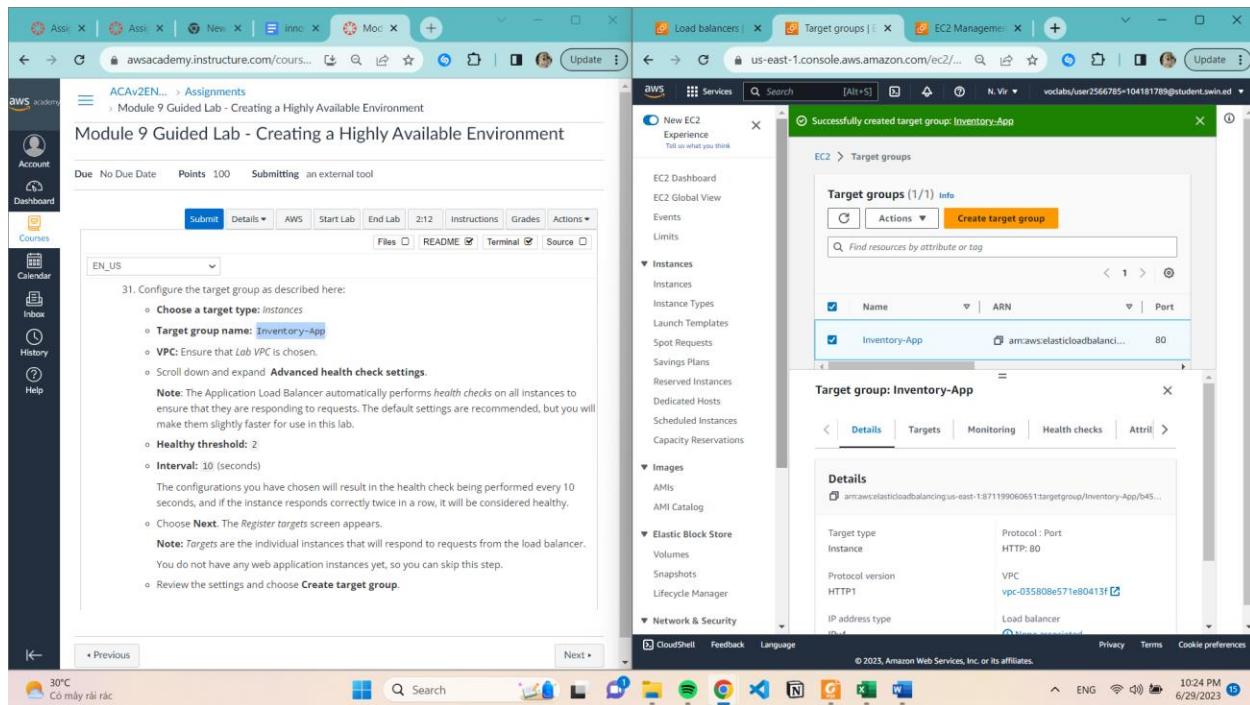
Type	Protocol	Port range
HTTP	TCP	80
Source type	Source	Description - optional
Anywhere-IPv4		0.0.0.0/0

Inbound rule 2

Type	Protocol	Port range
HTTPS	TCP	443
Source type	Source	Description - optional
Anywhere-IPv4		0.0.0.0/0

Create Target Group

- Configure the target group as described here:
- Choose a target type:** Instances
- Target group name:** Inventory-App
- VPC:** Ensure that Lab VPC is chosen.
- Scroll down and expand **Advanced health check settings.**
- Healthy threshold:** 2
- Interval:** 10 (seconds)



Successful create load balancer

The screenshot shows two browser tabs side-by-side. The left tab is from 'awsacademy.instructure.com' titled 'Module 9 Guided Lab - Creating a Highly Available Environment'. It contains several steps for creating a load balancer, including instructions to return to the browser tab where the load balancer was defined, choose the refresh icon in the 'Listeners and routing' section, set the 'Default action' to forward to the 'Inventory-App' target group, and scroll to the bottom to choose 'Create load balancer'. Step 35 lists two options: 'The load balancer is successfully created.' and 'Choose View load balancer.' Below this is a section titled 'Task 3: Creating an Auto Scaling group' with a note about Amazon EC2 Auto Scaling.

The right tab is from 'us-east-1.console.aws.amazon.com' titled 'Load balancers'. It shows a list of 'Load balancers (1/1)' with one entry named 'Inventory-LB'. The details for 'Inventory-LB' are shown in a modal window, including its status as 'Provisioning', hosted zone 'Z355XDOTRQ7X7K', and IP address type 'IPv4'. The modal also includes tabs for 'Details', 'Listeners', 'Network mapping', 'Security', and 'Monitoring'.

Task 3: Create Auto Scaling group

Create image

- **Image name:** Web Server AMI
- **Image description:** Lab AMI for Web Server

The left screenshot shows a browser window for 'awsacademy.instructure.com' displaying a module assignment titled 'Module 9 Guided Lab - Creating a Highly Available Environment'. It includes steps 39 and 40, which mention creating an AMI with 'Image name: web Server AMI' and 'Image description: Lab AMI for Web Server'. The right screenshot shows the AWS EC2 'Create Image' configuration page. It shows the instance ID 'i-0242e4f8af99b409e' and the selected 'Image name' 'Web Server AMI'. The 'Image description - optional' field contains 'Lab AMI for Web Server'. Under 'Instance volumes', there is one EBS volume of size 8 GiB and type 'EBS General Purpose S...'. A note at the bottom states: 'During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.'

Create a Launch Template and an Auto Scaling Group

Launch template name: Inventory-LT

Under **Auto Scaling guidance**, select *Provide guidance to help me set up a template that I can use with EC2 Auto Scaling*

In the Application and OS Images (Amazon Machine Image) area, choose My AMIs.

Amazon Machine Image (AMI): choose Web Server AMI

Instance type: choose t2.micro

Key pair name: choose vockey

Firewall (security groups): choose **Select existing security group**

Security groups: Inventory-App

IAM instance profile: Inventory-App-Role

Scroll down to the **Detailed CloudWatch monitoring setting**. Select **Enable**

The screenshot shows two browser windows side-by-side. The left window is an AWS Academy assignment titled 'Module 9 Guided Lab - Creating a Highly Available Environment'. It contains several steps for creating a launch template, including selecting 'Inventory-LT' as the launch template name and choosing 'Web Server AMI' as the AMI. Step 43 asks to 'Choose Create launch template' and step 44 asks to 'Configure the launch template settings and create it'. The right window is the AWS EC2 'Launch templates' page, showing a single entry for 'Inventory-LT' with its ID 'lt-03e63be38e95bf4d4'. The 'Details' tab of the launch template is open, displaying the configuration: Launch template ID 'lt-03e63be38e95bf4d4', Launch template name 'Inventory-LT', Default version '1', and Owner 'arn:aws:sts::871199060651:assumed-role/vockeyls/user2566785-104181789@student.swin.edu.au'. The status bar at the bottom of the screen shows the date and time as '6/29/2023 10:33 PM'.

Create Auto Scaling Group

Step 1

- **Auto Scaling group name:** Inventory-ASG
- **Launch template:** Inventory-LT template you just created is selected.

Step 2

- **VPC:** choose Lab VPC
- **Availability Zones and subnets:** Choose Private Subnet 1 and then choose Private Subnet 2.

Step 3

- In the **Load balancing** panel:
- Choose **Attach to an existing load balancer**
- **Existing load balancer target groups:** select Inventory-App.
- In the **Health checks** panel:
- **Health check grace period:** 90 seconds
- In the **Additional settings** panel:
- Select **Enable group metrics collection within CloudWatch**

The screenshot shows two overlapping windows. The left window is an AWS Academy assignment titled 'Module 9 Guided Lab - Creating a Highly Available Environment'. It displays step 49, which asks to configure advanced options for Auto Scaling. Step 49 includes instructions to attach to an existing load balancer (Inventory-LB), set a 90-second health check grace period, and enable CloudWatch Metrics collection. Step 50 asks to configure group size and scaling policies. The right window is a detailed configuration screen for 'Step 3: Configure advanced options' in the AWS CloudWatch Metrics collection interface. It shows the 'Load balancing' section with 'Inventory-LB' selected as the target group and 'Application/HTTP' as the type. The 'VPC Lattice integration options' section is empty. The 'Health checks' section shows 'EC2' as the health check type and '90 seconds' as the grace period. The 'Additional settings' section has 'Monitoring Enabled' and 'Default instance warmup Disabled'.

Step 4

- Under **Group size**, configure:
- **Desired capacity: 2**
- **Minimum capacity: 2**
- **Maximum capacity: 2**
- Under **Scaling policies**, choose None

awsacademy.instructure.com/courses/.../assignments

Create Auto Scale

Step 4: Configure group size and scaling policies

Group size

Desired capacity	2	Minimum capacity	2
Maximum capacity	2		

Scaling policy

No scaling policy

Instance scale-in protection

Instance scale-in protection

Enable instance protection from scale in

Step 5: Add notifications

Notifications

No notifications

Step 6: Add tags

Step 5,6

Choose Add tag and Configure the following:

Key: Name

Value: Inventory-App

The screenshot shows two browser windows side-by-side. The left window is an AWS Academy assignment titled 'Module 9 Guided Lab - Creating a Highly Available Environment'. It displays Step 52, which asks to 'Configure the details in Step 6 (Add tags - optional)'. Under 'Key: Name', it lists 'Inventory-App'. Step 53, 'Configure the details in Step 6 (Review)', is also visible. The right window is the AWS CloudFormation 'Create Auto Scaling' page. It shows the 'Scaling policy' section with 'No scaling policy' selected. In the 'Step 6: Add tags' section, a table shows a single tag: 'Name' (Inventory-App) and 'Value' (Yes). At the bottom right of the CloudFormation window, there is a button labeled 'Create Auto Scaling group'.

This screenshot shows the same two browser windows as the previous one. The left window now shows Step 53, 'Configure the details in Step 6 (Review)', with the review results indicating that the group has no instances but is currently updating. The right window shows the AWS CloudFormation 'Auto Scaling groups' page. A green banner at the top says 'Inventory-ASG created successfully. Group metrics collection is enabled.' Below this, the 'Auto Scaling groups (1/1) info' table shows one entry: 'Inventory-ASG' with 'Inventory-LT' as the launch template. The 'Auto Scaling group: Inventory-ASG' configuration page is open, showing settings like subnet (0545039b05eb62610), instance type requirements, load balancing (target group 'Inventory-App'), and VPC Lattice integration options.

Task 4: Updating security group

Application security group

The screenshot shows two browser windows side-by-side. The left window is an AWS Academy assignment titled 'Module 9 Guided Lab - Creating a Highly Available Environment'. It displays steps 57 and 58, which involve editing inbound rules. Step 57 asks to choose 'Edit inbound rules'. Step 58 provides instructions for doing so, mentioning 'Type: HTTP', 'Source' (with sub-instructions like choosing 'Custom', deleting current contents, entering 'sg', and selecting 'Inventory-LB'), and a 'Description' field ('Traffic from load balancer'). Step 58 also notes that application servers can now receive traffic from the load balancer. The right window is the 'EC2 Management' console, specifically the 'Edit inbound rules' page for a security group named 'sg-0e3323e48b2901aad'. It shows an existing rule: 'Inbound rule 1' with 'Security group rule ID' as 'sg-0e3323e48b2901aad', 'Type' as 'HTTP', 'Protocol' as 'TCP', 'Port range' as '80', 'Source type' as 'Custom', and 'Source' as 'sg-0e9f3edda035c1ce'. A description 'Traffic from load balancer' is present. Buttons for 'Add rule', 'Cancel', 'Preview changes', and 'Save rules' are visible.

Database security group (Create inbound rules)

This screenshot shows the same setup as the previous one, but for a 'Database security group'. The assignment step 59 instructs to choose 'inventory-us' from the 'security groups' list. Step 60 details how to edit inbound rules, including deleting the existing rule, adding a new one for 'MySQL/Aurora' on port 3306 with a 'Custom' source, and saving it with a 'Description' of 'Traffic from application servers'. The EC2 Management console window shows the updated rule: 'Inbound rule 1' with 'Security group rule ID' as 'sg-0501bc13d448e2510', 'Type' as 'MYSQL/Aurora', 'Protocol' as 'TCP', 'Port range' as '3306', 'Source type' as 'Custom', and 'Source' as 'sg-03323e48b2901aad'. A description 'Traffic from application servers' is present. Buttons for 'Add rule', 'Cancel', 'Preview changes', and 'Save rules' are visible.

Task 5: Testing the application

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

62. Select **Inventory-App**.

63. In the lower half of the page, choose the **Targets** tab.

This tab should show two *registered targets*. The **Health status** column shows the results of the load balancer health check that is performed against the instances.

64. In the **Registered targets** area, occasionally choose the refresh icon until the **Status** for both instances appears as *healthy*.

If the status does not eventually change to *healthy*, ask your educator for help with diagnosing the configuration.

You will test the application by connecting to the load balancer, which will then send your request to one of the EC2 instances. You will first need to retrieve the Domain Name System (DNS) name of the load balancer.

The load balancer forwarded your request to one of the EC2 instances. The instance ID and Availability Zone are shown at the bottom of the webpage.

68. Reload the page in your web browser. You should notice that the instance ID and Availability Zone sometimes toggles between the two instances.

When this web application displays, the flow of data over the network is:

- You sent the request to the **load balancer**, which resides in the **public subnets** that are connected to the internet.
- The load balancer chose one of the **EC2 instances** that reside in the **private subnets** and forwarded the request to it.
- The EC2 instance then returned the webpage content to the load balancer, which

Task 6: Testing high availability

The screenshot shows two windows side-by-side. On the left is a web browser displaying the 'Module 9 Guided Lab - Creating a Highly Available Environment' from awsacademy.instructure.com. The task 71 is visible: 'Select one of the Inventory-App instances (it does not matter which one you select)'. On the right is the AWS CloudWatch Metrics console showing metrics for 'Inventory System' over time.

The screenshot shows two windows side-by-side. On the left is a web browser displaying the 'Module 9 Guided Lab - Creating a Highly Available Environment' from awsacademy.instructure.com. Task 74 is visible: 'Return to the web application tab in your web browser and reload the page several times.' On the right is the AWS EC2 Instances console showing a list of instances. A modal window titled 'Terminate instance?' is open, asking if the user wants to terminate instance i-01b75127985ffe23f. The status of this instance is shown as 'Terminated' in the list.

Optional task 1: Making the database highly available

The screenshot shows two browser windows side-by-side. The left window is from AWS Academy, displaying a guided lab titled 'Module 9 Guided Lab - Creating a Highly Available Environment'. It contains numbered steps 76 through 80, which describe creating an Amazon RDS MySQL instance and setting it up as a multi-AZ standby. Step 80 specifically mentions selecting 'Create a standby instance'. The right window is from the Amazon RDS console, showing the 'Databases' section. It lists a single database named 'inventory-db' with the status 'Available'. A tooltip suggests creating a Blue/Green deployment. The bottom status bar indicates the date as 6/29/2023 and the time as 11:01 PM.

Optional task 2: Configuring a highly available NAT gateway

Create NAT gateway

This screenshot also features two browser windows. The left window is the same AWS Academy module as before, with steps 86 through 89. Step 86 asks to choose VPC, step 87 shows the existing NAT gateway, and steps 88 and 89 guide the user through creating a new NAT gateway in a private subnet. The right window is the VPC Management console, specifically the 'NAT gateways' section. It shows a new NAT gateway being created with the name 'natgate' and assigned to the public subnet 'subnet-03303b930fcfaef6d'. The status bar at the bottom shows the date as 6/29/2023 and the time as 11:05 PM.

Create route table

The image shows two side-by-side browser windows. The left window is from AWS Academy, displaying a guided lab titled 'Module 9 Guided Lab - Creating a Highly Available Environment'. It shows step 90: 'Choose Create route table and configure these settings' with options for Name (Private Route Table 2), VPC (Lab VPC), and Create route table. Step 91: 'Observe the settings in the Routes tab.' Step 92: 'Choose Edit routes and then configure these settings' with options for Add route, Destination (0.0.0.0/0), and Target (Select NAT Gateway). A tip at the bottom says 'Tip: To discover which nat. entry is NOT the one to select, choose the Details button above these instructions and then select Show, the name of NATGateway1 is displayed. Return to the VPC console and select the Target that is NOT NATGateway1.' The right window is from the AWS VPC Manager, specifically the 'Create route table' page. It shows a 'Route table settings' section with 'Name' set to 'Private Route Table 2' and 'VPC' set to 'vpc-035808e571e80413f (Lab VPC)'. Below it is a 'Tags' section where a tag 'Name: Private Route Table 2' is added. At the bottom right is a 'Create route table' button.

Edit routes

The image shows two side-by-side browser windows. The left window is from AWS Academy, displaying the same guided lab as before. It shows step 91: 'Observe the settings in the Routes tab.' Step 92: 'Choose Edit routes and then configure these settings' with options for Add route, Destination (0.0.0.0/0), and Target (Select NAT Gateway). A tip at the bottom says 'Tip: To discover which nat. entry is NOT the one to select, choose the Details button above these instructions and then select Show, the name of NATGateway1 is displayed. Return to the VPC console and select the Target that is NOT NATGateway1.' The right window is from the AWS VPC Manager, specifically the 'Edit routes' page for a route table named 'rtb-0d7d8aefbf1b116bed'. It shows two routes: one for destination 10.0.0.0/16 targetting 'local' (Status: Active) and another for destination 0.0.0.0/0 targetting 'nat-057b43a557475a24' (Status: In Progress). There are 'Edit routes' and 'Add route' buttons at the bottom.

Save subnet association

Module 9 Guided Lab - Creating a Highly Available Environment

Due No Due Date Points 100 Submitting an external tool

EN_US

above these instructions and then select **Show**, the name of NATGateway1 is displayed. Return to the VPC console and select the Target that is *NOT* NATGateway1.

- o Choose **Save changes**

93. Choose the **Subnet associations** tab.

94. Choose Edit subnet associations

95. Select **Private Subnet 2**.

96. Choose Save associations

This action now sends internet-bound traffic from Private Subnet 2 to the NAT gateway that is in the same Availability Zone.

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

Available subnets (1/4)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table
<input checked="" type="checkbox"/> Private Subnet 2	subnet-04639e3ee8f67...	10.0.4.0/23	-	rtb-02baa99
<input type="checkbox"/> Public Subnet 2	subnet-03303b930fcfa...	10.0.1.0/24	-	rtb-065358b
<input type="checkbox"/> Private Subnet 1	subnet-0545039b05eb...	10.0.2.0/23	-	rtb-02baa99
<input type="checkbox"/> Public Subnet 1	subnet-0fe7cfe2affe20...	10.0.0.0/24	-	rtb-065358b

Selected subnets

subnet-04639e3ee8f67fc3c / Private Subnet 2 X

Cancel **Save associations**