

These are my screenshots for Module 9 Challenge Lab - Creating a Scalable and Highly Available Environment for the Cafe

Create NAT gateway

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.

► 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 resources across your AWS account.

[CloudShell](#) [Feedback](#) [Language](#) Privacy Terms Cookie preferences © 2023, Amazon Web Services, Inc. or its affiliates.

EN_US access the internet.

8. Create a NAT gateway in the **Public Subnet** in the second Availability Zone.
9. Configure the network to send internet-bound traffic from instances in **Private Subnet 2** to the NAT gateway you just created.

Task 3: Creating a bastion host

Due No Due Date Points 100 Submitting an external tool

Submit Details AWS Start Lab End Lab 2:37 Instructions Grades Actions Files README Terminal Source EN_US

Previous Next

Edit routes

Edit routes

Destination	Target	Status
10.0.0.0/16	<input type="text" value="local"/> <input type="button" value="X"/>	<input checked="" type="radio"/> Active

Propagated
No

Edit routes

Destination	Target	Status
<input type="text" value="0.0.0.0/0"/> <input type="button" value="X"/>	<input type="text" value="nat-0b4a649baeb572ba9"/> <input type="button" value="X"/>	-

Propagated
No

EN_US access the internet.

8. Create a NAT gateway in the **Public Subnet** in the second Availability Zone.
9. Configure the network to send internet-bound traffic from instances in **Private Subnet 2** to the NAT gateway you just created.

Task 3: Creating a bastion host

Due No Due Date Points 100 Submitting an external tool

Submit Details AWS Start Lab End Lab 2:34 Instructions Grades Actions Files README Terminal Source EN_US

Previous Next

Task 4: Creating a launch template

Auto-assign public IP info

Enable

Firewall (security group) info

Create security group

SSH only SG

Inbound Security Group Rules

Type: ssh, Protocol: TCP, Port range: 22, Source type: My IP

Only allows the following traffic:

- Type: SSH
- Port: 22
- Source: Your IP address

Task 5: Creating an Auto Scaling group

Success

Successfully created HAnhTemp (it-0ad6cc56259ad57a57)

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 you 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

Screenshot 1: AWS Management Console - EC2 Management Console

The screenshot shows the AWS Management Console with the EC2 Management Console open. The left sidebar navigation includes Services, Search, [Alt+S], N. Vir, and vclabs/user/2555392-10417751@student.swin.edu.au. Under Instances, Launch Templates is selected. A specific launch template version is displayed in the main pane.

Launch template version details:

- Version: 1 (Default)
- Description: -
- Date created: 2023-07-23T07:52:51.000Z
- Created by: am:awssts:992902254046:assumed-role/vclabs/user/2555392-104177513@student.swin.edu.au
- AMI ID: ami-03a6215b7f0ca8f59
- Instance type: t2.micro
- Key pair name: keylab9
- Availability Zone: -
- Security groups: sg-0b1aa6ea2ed8a0ec0
- Security group IDs: -

Screenshot 2: AWS Academy - Module 9 Challenge Lab - Creating a Scalable and Highly Available Environment for the Cafe

The screenshot shows the AWS Academy challenge lab interface. The task is titled "Module 9 Challenge Lab - Creating a Scalable and Highly Available Environment for the Cafe".

Task 6: Creating a load balancer

Console output:

```
console. You should have two instances, both with the name that you configured as resource tags in the previous task.
```

Instructions:

Now that you web application server instances are deployed in private subnets, you need a way for the outside world to connect

Create Auto Scaling group | EC2 NAT gateways | VPC Manager

Health check type: EC2, Health check grace period: 300 seconds

Additional settings

Monitoring: Disabled	Default instance warmup: Disabled
----------------------	-----------------------------------

Step 4: Configure group size and scaling policies

Group size

Desired capacity: 2	Minimum capacity: 2
Maximum capacity: 6	

Scaling policy

Target tracking scaling	Policy type: Target tracking scaling	Scaling policy name: Target Tracking Policy
-------------------------	--------------------------------------	---

As required to maintain Average CPU

Review

Verify your settings for your Auto Scaling group, and edit the settings as needed. When you are satisfied with your settings, choose **Create Auto Scaling group**.

After Amazon EC2 Auto Scaling creates your Auto Scaling group, it immediately starts launching instances. The new instances appear in the list of instances on the console. After an instance is fully configured and passes the initial health checks, it is considered healthy by Amazon EC2 Auto Scaling and enters the **InService** state.

After creating your Auto Scaling group, you can open it in the console and configure other settings. Several additional features are available, such as:

- **Scheduled scaling**
- **Predictive scaling**
- **Termination policies**
- **Lifecycle hooks**

aws academy

Home, Modules, Announcements, Discussions, Grades

Module 9 Challenge Lab - Creating a Scalable and Highly Available Environment for the Cafe

Due, No Due Date, Points 100, Submitting, an external tool

Submit, Details, AWS, Start Lab, End Lab, 2:17, Instructions, Grades

Actions

Files, README, Terminal, Source

EN_US

console. You should have two instances, both with the name that you configured as resource tags in the previous task.

Task 6: Creating a load balancer

Now that you web application server instances are deployed in private subnets, you need a way for the outside world to connect

EC2 Management Console NAT gateways | VPC Manager

New EC2 Experience

Instances (4) info

Name	Instance ID	Instance state	Instance type
webserver	i-05eaaaf6cf8ba77c4	Running	t2.micro
CafeWebAppS...	i-0b94115510bd8a0fc	Running	t2.micro
Bastion Host	i-04b2d33a24fc53365	Running	t2.micro
webserver	i-02a5390be0800bbef	Running	t2.micro

Select an instance

aws academy

Home, Modules, Announcements, Discussions, Grades

Module 9 Challenge Lab - Creating a Scalable and Highly Available Environment for the Cafe

Due, No Due Date, Points 100, Submitting, an external tool

Submit, Details, AWS, Start Lab, End Lab, 2:16, Instructions, Grades

Actions

Files, README, Terminal, Source

EN_US

13. To verify that you created the Auto Scaling group correctly, go to the **Amazon EC2 console**. You should have two instances, both with the name that you configured as resource tags in the previous task.

Task 6: Creating a load balancer

Now that you web

The screenshot shows two windows side-by-side. On the left, the AWS Management Console displays the 'Network mapping info' for a load balancer. It shows a VPC selected (vpc-047d3477820192732) and two subnets assigned: subnet-0aff7e4e2781b0bd0 (Public Subnet 1) and subnet-0464e96954fa401cf (Public Subnet 2). Both are IPv4 addresses assigned by AWS. On the right, an AWS Academy assignment titled 'Module 9 Challenge Lab - Creating a Scalable and Highly Available Environment for the Cafe' is shown. The assignment details include a due date of No Due Date, 100 points, and the option to submit an external tool. The assignment content lists requirements: VPC (Uses the VPC configured for this lab), Subnets (Uses the two public subnets), Security group (Creates a new security group that allows HTTP traffic from anywhere), and Target group (Creates a new target group).

This screenshot shows the AWS Management Console on the left where a new security group is being created. The 'Basic details' section shows the security group name as 'ELB SG' and the VPC as 'vpc-047d3477820192732'. The 'Inbound rules' section contains one rule: Type: HTTP, Protocol: TCP, Port range: 80, Source type: Anywhere-IPv4, and Source IP: 0.0.0.0/0. The 'Outbound rules' section is empty. On the right, the same AWS Academy assignment page is displayed, listing the requirements for the challenge lab.

The screenshot shows the AWS Management Console with the EC2 Manager service selected. A modal window titled "Successfully created target group: HanhTarget" is open, indicating the creation of a new target group. The main EC2 Target groups page shows a single entry: "HanhTarget" with ARN "arn:aws:elasticloadbalancing:us-east-1:104177513@student.swin.edu.au:targetgroup/HanhTarget/80". The sidebar on the left lists various EC2 services like Instances, Images, and Network & Security.

The screenshot shows the AWS Management Console with the EC2 Manager service selected. A modal window titled "Successfully created load balancer: HanhELB" is open, indicating the creation of a new Application Load Balancer. The main EC2 Load balancers page shows a single entry: "HanhELB" with ARN "arn:aws:elasticloadbalancing:us-east-1:104177513@student.swin.edu.au:loadbalancer/app/HanhELB/104177513-HanhELB". The sidebar on the left lists various EC2 services like Instances, Images, and Network & Security.

Screenshot of the AWS Management Console showing the creation of an Auto Scaling group. The 'Instance type requirements' section is open, showing a launch template named 'HAnhTemp' with version 1 and an instance type of 't2.micro'. The 'Load balancing - optional' section shows a selected target group 'HAnhTarget | HTTP Application Load Balancer: HAnhELB'.

Screenshot of the AWS Management Console showing the creation of an Auto Scaling group. The 'Instance type requirements' section is open, showing a launch template named 'HAnhTemp' with version 1 and an instance type of 't2.micro'. The 'Load balancing - optional' section shows a selected target group 'HAnhTarget | HTTP Application Load Balancer: HAnhELB'.

Screenshot of a web browser displaying a simulated cafe website. The page features a header with the word 'Café' and a navigation bar with links like Home, About Us, Contact Us, Menu, and Order History. Below the navigation is a large image of pastries. A sidebar on the left says 'Frank bakes a rich variety of cookies. Try them all!' with images of cookies. Another sidebar on the right says 'Our tarts are always a customer favorite!' with images of tarts. The main content area describes the cafe's offerings.

The screenshot shows a dual-monitor setup. The left monitor displays the AWS EC2 Management console, specifically the Instances page. It lists four instances: 'webserver' (running, t2.micro), 'CafeWebAppS...' (running, t2.micro), 'Bastion Host' (running, t2.micro), and another 'webserver' (running, t2.micro). The right monitor displays an online course titled 'Module 9 Challenge Lab - Creating a Scalable and Highly Available Environment for the Café'. The course interface includes sections for Home, Modules, Announcements, Discussions, Grades, and Actions. It shows a total score of 36/36. A sidebar on the right contains links for [Answer 01] through [Answer 06].