

REPORT
on
SKILL ORIENTED COURSE – I (Course Code: V20SOC01)
B.TECH III SEM (2021-2025 Batch)
Academic Year: 2022-23



SRI VASAVI ENGINEERING COLLEGE(Autonomous)
Pedatadepalli,TADEPALLIGUDEM
Department of Computer Science and Engineering
[Artificial Intelligence]

SRI VASAVI ENGINEERING COLLEGE (Autonomous)
PEDATADEPALLI, TADEPALLIGUDEM.



Certificate

*This is to certify Mr./Ms. _____ bearing
RollNo _____ of **CAI** Branch of **III Semester** submitted the report on
_____ as a part of **SKILL ORIENTED**
COURSE – I -AWS Academy Cloud Foundations (Course Code: V20SOC01) during the
academic year **2022-23**.*

Faculty Incharge

Head of the Department

External Examiner

INDEX

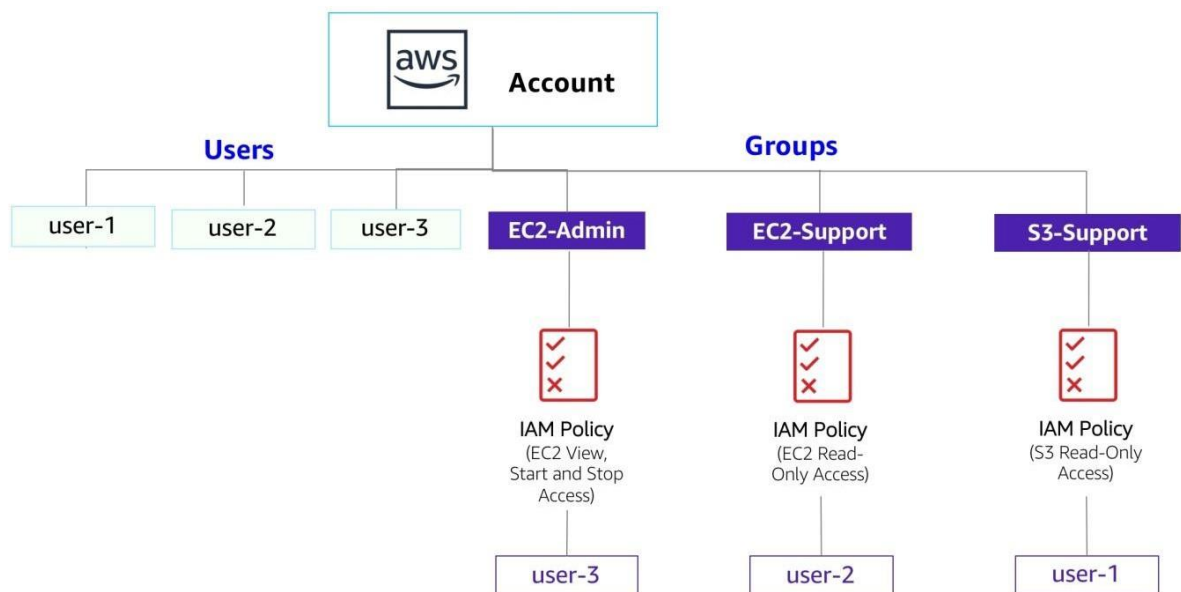
S.No	Name of the Topic	Page. No	Remarks
1	Introduction to AWS IAM	1-10	
2	Build your VPC and Launch a Web Server	11-18	
3	Introduction to Amazon EC2	19-29	
4	Working with EBS	30-39	
5	Build a Database Server	40-45	
6	Scale & Load Balance your Architecture	46-55	

Lab 1: Introduction to AWS IAM

AWS Identity and Access Management (IAM) is a web service that enables Amazon Web Services (AWS) customers to manage users and user permissions in AWS. With IAM, you can centrally manage **users**, **security credentials** such as access keys, and **permissions** that control which AWS resources users can access.

Topics covered

This lab will demonstrate:



- Exploring pre-created **IAM Users and Groups**
- Inspecting **IAM policies** as applied to the pre-created groups
- Following a **real-world scenario**, adding users to groups with specific capabilities enabled
- Locating and using the **IAM sign-in URL**
- **Experimenting** with the effects of policies on service access

Other AWS Services

During this lab, you may receive error messages when performing actions beyond the steps in this lab guide. These messages will not impact your ability to complete the lab.

AWS Identity and Access Management

AWS Identity and Access Management (IAM) can be used to:

- **Manage IAM Users and their access:** You can create Users and assign them individual security credentials (access keys, passwords, and multi-factor authentication devices). You can manage permissions to control which operations a User can perform.
- **Manage IAM Roles and their permissions:** An IAM Role is similar to a User, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a Role is intended to be *assumable* by anyone who needs it.
- **Manage federated users and their permissions:** You can enable *identity federation* to allow existing users in your enterprise to access the AWS Management Console, to call AWS APIs and to access resources, without the need to create an IAM User for each identity.

Duration

This lab takes approximately **40 minutes** to complete.

Accessing the AWS Management Console

1. At the top of these instructions, choose **Start Lab** to launch your lab.

A Start Lab panel opens displaying the lab status. In the **Start Lab** dialog box that opens, note the AWS Region, as you will need to refer to it later in this lab.

2. Wait until you see the message "**Lab status: ready**", then click the **X** to close the Start Lab panel.
3. At the top of these instructions, choose **AWS** This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Click on the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Explore the Users and Groups

In this task, you will explore the Users and Groups that have already been created for you in IAM.

5. In the **AWS Management Console**, on the **Services** menu, select **IAM**.
6. In the navigation pane on the left, choose **Users**.

The following IAM Users have been created for you:

- user-1
 - user-2
 - user-3
7. Choose **user-1**.

This will bring to a summary page for user-1. The **Permissions** tab will be displayed.

8. Notice that user-1 does not have any permissions.
9. Choose the **Groups** tab.

user-1 also is not a member of any groups.

10. Choose the **Security credentials** tab.

user-1 is assigned a **Console password**

11. In the navigation pane on the left, choose **User groups**.

The following groups have already been created for you:

- EC2-Admin
 - EC2-Support
 - S3-Support
12. Choose the **EC2-Support** group.

This will bring you to the summary page for the **EC2-Support** group.

13. Choose the **Permissions** tab.

This group has a Managed Policy associated with it, called **AmazonEC2ReadOnlyAccess**. Managed Policies are pre-built policies (built either by AWS or by your administrators) that can be attached to IAM Users and Groups. When the policy is updated, the changes to the policy are immediately apply against all Users and Groups that are attached to the policy.

14. Choose the plus (+) icon next to the AmazonEC2ReadOnlyAccess policy to view the policy details..

The basic structure of the statements in an IAM Policy is:

- **Effect** says whether to *Allow* or *Deny* the permissions.

- **Action** specifies the API calls that can be made against an AWS Service (eg *cloudwatch:ListMetrics*).
 - **Resource** defines the scope of entities covered by the policy rule (eg a specific Amazon S3 bucket or Amazon EC2 instance, or * which means *any resource*).
15. Choose the minus icon (-) to hide the policy details.
 16. In the navigation pane on the left, choose **User groups**.
 17. Choose the **S3-Support** group and then choose the **Permissions** tab. The S3-Support group has the **AmazonS3ReadOnlyAccess** policy attached.
 18. Choose the plus (+) icon to view the policy details.

This policy grants permissions to Get and List resources in Amazon S3.

19. Choose the minus icon (-) to hide the policy details.
20. In the navigation pane on the left, choose **User groups**.
21. Choose the **EC2-Admin** group and then choose the **Permissions** tab.

This Group is slightly different from the other two. Instead of a *Managed Policy*, it has an **Inline Policy**, which is a policy assigned to just one User or Group. Inline Policies are typically used to apply permissions for one-off situations.

22. Choose the plus (+) icon to view the policy details.

This policy grants permission to view (Describe) information about Amazon EC2 and also the ability to Start and Stop instances.

23. Choose the minus icon (-) to hide the policy details.

Business Scenario

For the remainder of this lab, you will work with these Users and Groups to enable permissions supporting the following business scenario:

Your company is growing its use of Amazon Web Services, and is using many Amazon EC2 instances and a great deal of Amazon S3 storage. You wish to give access to new staff depending upon their job function:

User	In Group	Permissions
user-1	S3-Support	Read-Only access to Amazon S3
user-2	EC2-Support	Read-Only access to Amazon EC2
user-3	EC2-Admin	View, Start and Stop Amazon EC2 instances

Task 2: Add Users to Groups

You have recently hired **user-1** into a role where they will provide support for Amazon S3. You will add them to the **S3-Support** group so that they inherit the necessary permissions via the attached *AmazonS3ReadOnlyAccess* policy.

You can ignore any "not authorized" errors that appear during this task. They are caused by your lab account having limited permissions and will not impact your ability to complete the lab.

Add user-1 to the S3-Support Group

24. In the left navigation pane, choose **User groups**.
25. Choose the **S3-Support** group.
26. Choose the **Users** tab.
27. In the **Users** tab, choose **Add users**.
28. In the **Add Users to S3-Support** window, configure the following:
 - Select **user-1**.
 - At the bottom of the screen, choose **Add Users**.

In the **Users** tab you will see that user-1 has been added to the group.

Add user-2 to the EC2-Support Group

You have hired **user-2** into a role where they will provide support for Amazon EC2.

29. Using similar steps to the ones above, add **user-2** to the **EC2-Support** group.

user2 should now be part of the **EC2-Support** group.

Add user-3 to the EC2-Admin Group

You have hired **user-3** as your Amazon EC2 administrator, who manage your EC2 instances.

30. Using similar steps to the ones above, add **user-3** to the **EC2-Admin** group.

user-3 should now be part of the **EC2-Admin** group.

31. In the navigation pane on the left, choose **User groups**.

Each Group should now have a **1** in the Users column for the number of Users in each Group.

If you do not have a **1** beside each group, revisit the above instructions above to ensure that each user is assigned to a User group, as shown in the table in the Business Scenario section.

Task 3: Sign-In and Test Users

In this task, you will test the permissions of each IAM User.

32. In the navigation pane on the left, choose **Dashboard**.

An **IAM users sign-in link** is displayed on the right. It will look similar to: *<https://123456789012.signin.aws.amazon.com/console>*

This link can be used to sign-in to the AWS Account you are currently using.

33. Copy the **Sign-in URL for IAM users in this account** to a text editor.
34. Open a private (Incognito) window.

Mozilla Firefox

- Choose the menu bars at the top-right of the screen
- Select **New private window**

Google Chrome

- Choose the ellipsis at the top-right of the screen
- Select **New Incognito Window**

Microsoft Edge

- Choose the ellipsis at the top-right of the screen
- Choose **New InPrivate window**

Microsoft Internet Explorer

- Choose the **Tools** menu option
 - Choose **InPrivate Browsing**
35. Paste the **IAM users sign-in** link into the address bar of your private browser session and press **Enter**.

Next, you will sign-in as **user-1**, who has been hired as your Amazon S3 storage support staff.

36. Sign-in with:

- **IAM user name:** user-1
- **Password:** Lab-Password1

37. In the **Services** menu, choose **S3**.

38. Choose the name of the bucket that exists in the account and browse the contents.

Since your user is part of the **S3-Support** Group in IAM, they have permission to view a list of Amazon S3 buckets and the contents.

Note: The bucket does not contain any objects.

Now, test whether they have access to Amazon EC2.

39. In the **Services** menu, choose **EC2**.

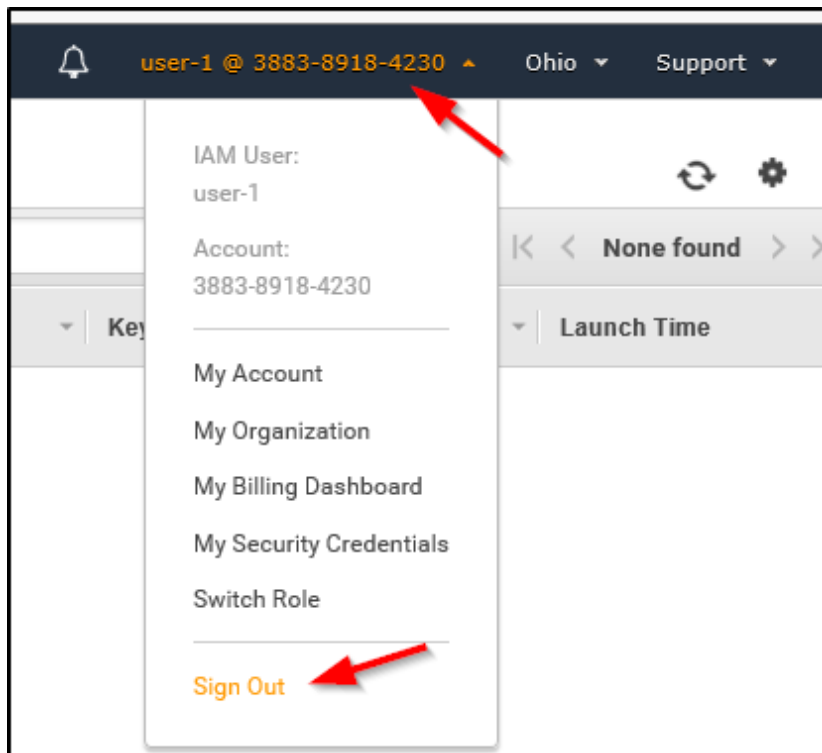
40. In the left navigation pane, choose **Instances**.

You cannot see any instances. Instead, you see a message that states *You are not authorized to perform this operation*. This is because this user has not been granted any permissions to access Amazon EC2.

You will now sign-in as **user-2**, who has been hired as your Amazon EC2 support person.

41. Sign user-1 out of the **AWS Management Console** by completing the following actions:

- At the top of the screen, choose **user-1**
- Choose **Sign Out**



42. Paste the **IAM users sign-in** link into your private browser tab's address bar and press **Enter**.

Note: This link should be in your text editor.

43. Sign-in with:

- **IAM user name:** user-2
- **Password:** Lab-Password2

44. In the **Services** menu, choose **EC2**.

45. In the navigation pane on the left, choose **Instances**.

You are now able to see an Amazon EC2 instance because you have Read Only permissions. However, you will not be able to make any changes to Amazon EC2 resources.

If you cannot see an Amazon EC2 instance, then your Region may be incorrect. In the top-right of the screen, pull-down the Region menu and select the region that you noted at the start of the lab (for example, **N. Virginia**).

- Select the instance named *LabHost*.

46. In the **Instance state** menu above, select **Stop instance**.

47. In the **Stop Instance** window, select **Stop**.

You will receive an error stating *You are not authorized to perform this operation*. This demonstrates that the policy only allows you to view information, without making changes.

48. Choose the X to close the *Failed to stop the instance* message.

Next, check if user-2 can access Amazon S3

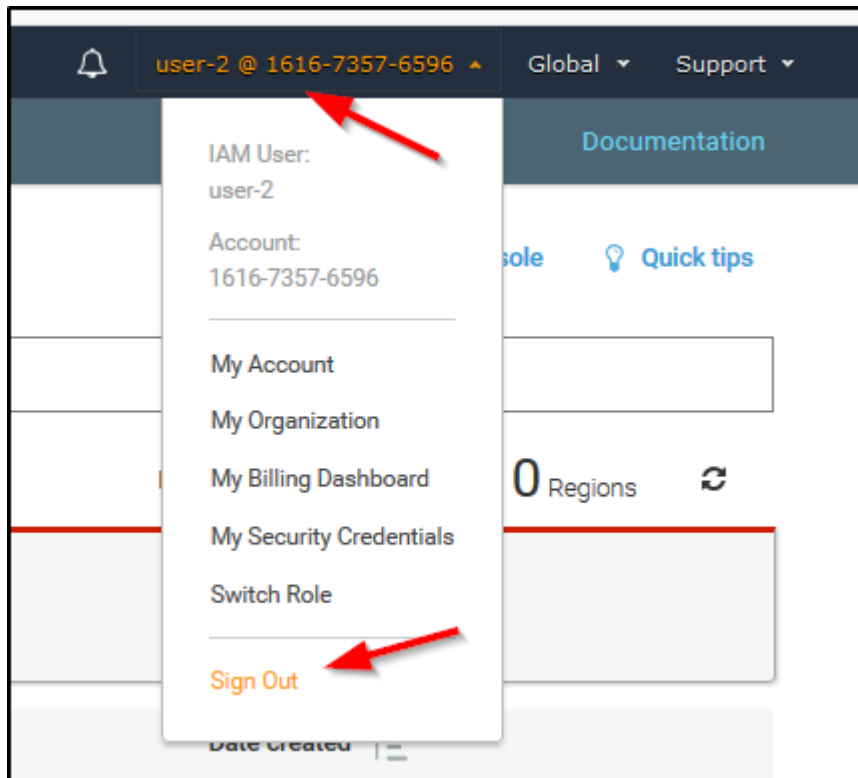
49. In the **Services**, choose **S3**.

You will see the message **You don't have permissions to list buckets** because user-2 does not have permission to access Amazon S3.

You will now sign-in as **user-3**, who has been hired as your Amazon EC2 administrator.

50. Sign user-2 out of the **AWS Management Console** by completing the following actions:

- At the top of the screen, choose **user-2**
- Choose **Sign Out**



51. Paste the **IAM users sign-in** link into your private window and press **Enter**.
52. Paste the sign-in link into the address bar of your private web browser tab again. If it is not in your clipboard, retrieve it from the text editor where you stored it earlier
53. Sign-in with:
 - **IAM user name:** user-3
 - **Password:** Lab-Password3
54. In the **Services** menu, choose **EC2**.
55. In the navigation pane on the left, choose **Instances**.

As an EC2 Administrator, you should now have permissions to Stop the Amazon EC2 instance.

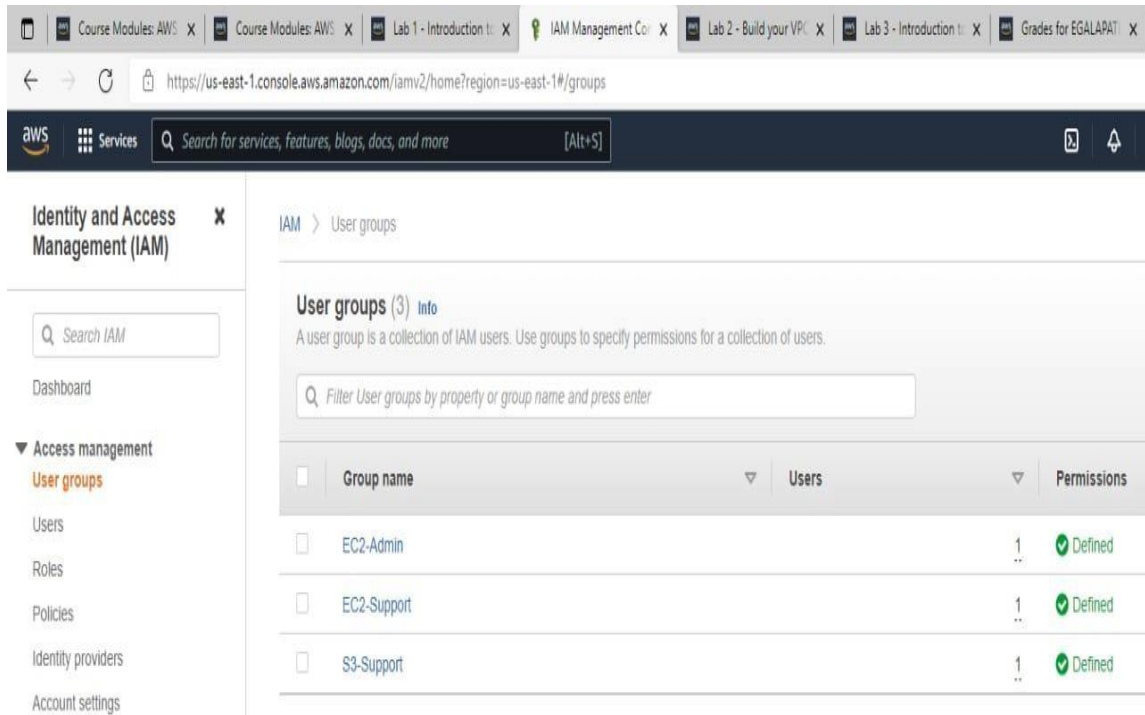
Select the instance named *LabHost*.

If you cannot see an Amazon EC2 instance, then your Region may be incorrect. In the top-right of the screen, pull-down the Region menu and select the region that you noted at the start of the lab (for example, **N. Virginia**).

56. In the **Instance state** menu, choose **Stop instance**.
57. In the **Stop instance** window, choose **Stop**. The instance will enter the *stopping* state and will shutdown.

58. Close your private browser window.
59. End Lab

Output:



The screenshot shows the AWS IAM console interface. The left sidebar contains the 'Identity and Access Management (IAM)' menu with options like Dashboard, Access management, User groups, Users, Roles, Policies, Identity providers, and Account settings. The main content area is titled 'User groups (3)' and includes a search bar. Below the search bar is a table listing the user groups.

<input type="checkbox"/>	Group name	Users	Permissions
<input type="checkbox"/>	EC2-Admin	1	Defined
<input type="checkbox"/>	EC2-Support	1	Defined
<input type="checkbox"/>	S3-Support	1	Defined

Lab 2: Build your VPC and Launch a Web Server

Lab overview and objectives

In this lab, you will use Amazon Virtual Private Cloud (VPC) to create your own VPC and add additional components to produce a customized network. You will also create security groups for your EC2 instance. You will then configure and customize an EC2 instance to run a web server and launch it into the VPC.

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you defined. This virtual network closely resembles a traditional network that you would operate in your own data center, with the benefits of using the scalable infrastructure of AWS. You can create a VPC that spans multiple Availability Zones.

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

- Create a VPC.
- Create subnets.
- Configure a security group.
- Launch an EC2 instance into a VPC.

Duration

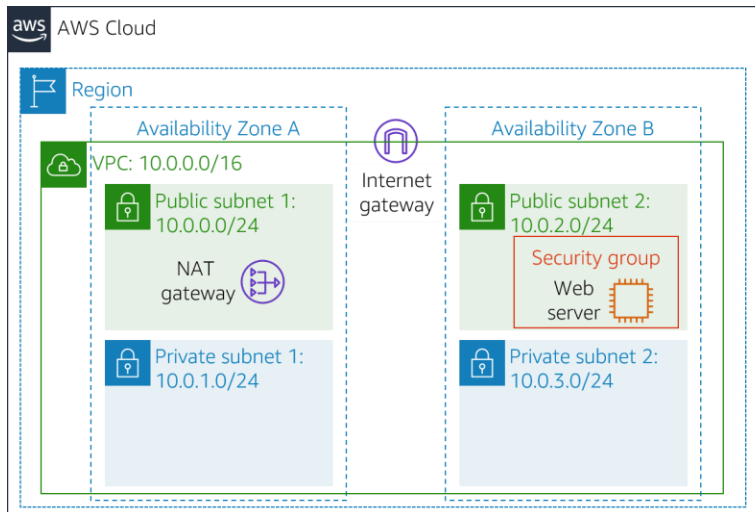
This lab takes approximately **30 minutes** to complete.

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

Scenario

In this lab you build the following infrastructure:



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway

Accessing the AWS Management Console

1. At the top of these instructions, choose [Start Lab](#) to launch your lab.

A Start Lab panel opens displaying the lab status.

2. Wait until you see the message "**Lab status: ready**", then choose the **X** to close the Start Lab panel
3. At the top of these instructions, choose [AWS](#)

This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Choose on the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Create Your VPC

In this task, you will use the VPC Wizard to create a VPC an Internet Gateway and two subnets in a single Availability Zone. An **Internet gateway (IGW)** is a VPC component that allows communication between instances in your VPC and the Internet.

After creating a VPC, you can add **subnets**. Each subnet resides entirely within one Availability Zone and cannot span zones. If a subnet's traffic is routed to an Internet Gateway, the subnet is known as a *public subnet*. If a subnet does not have a route to the Internet gateway, the subnet is known as a *private subnet*.

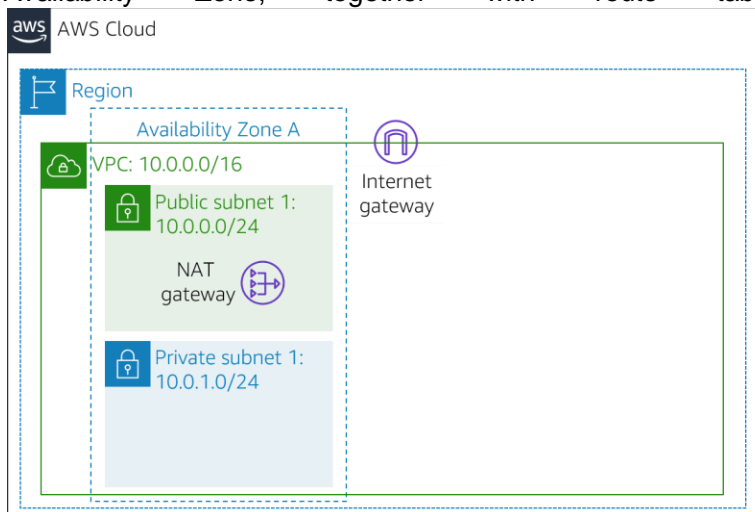
The wizard will also create a *NAT Gateway*, which is used to provide internet connectivity to EC2 instances in the private subnets.

5. In the search box to the right of **Services**, search for and choose **VPC** to open the VPC console.
6. Choose **Launch VPC Wizard**
7. In the left navigation pane, choose **VPC with Public and Private Subnets** (the second option).
8. Choose **Select** then configure:
 - **VPC name:** Lab VPC
 - **Availability Zone:** Select the *first* Availability Zone
 - **Public subnet name:** Public Subnet 1
 - **Availability Zone:** Select the *first* Availability Zone (the same as used above)
 - **Private subnet name:** Private Subnet 1
 - **Elastic IP Allocation ID:** Choose in the box and select the displayed IP address
9. Choose **Create VPC**

The wizard will create your VPC.

10. Once it is complete, choose **OK**

The wizard has provisioned a VPC with a public subnet and a private subnet in the same Availability Zone, together with route tables for each subnet:



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway

The Public Subnet has a CIDR of **10.0.0.0/24**, which means that it contains all IP addresses starting with **10.0.0.x**.

The Private Subnet has a CIDR of **10.0.1.0/24**, which means that it contains all IP addresses starting with **10.0.1.x**.

Task 2: Create Additional Subnets

In this task, you will create two additional subnets in a second Availability Zone. This is useful for creating resources in multiple Availability Zones to provide *High Availability*.

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

First, you will create a second Public Subnet.

12. Choose **Create subnet** then configure:
- **VPC ID:** Lab VPC (select from the menu).
 - **Subnet name:** Public Subnet 2
 - **Availability Zone:** Select the *second* Availability Zone
 - **IPv4 CIDR block:** 10.0.2.0/24

The subnet will have all IP addresses starting with **10.0.2.x**.

13. Choose **Create subnet**

The second *public* subnet was created. You will now create a second *private* subnet.

14. Choose **Create subnet** then configure:
- **VPC ID:** Lab VPC
 - **Subnet name:** Private Subnet 2
 - **Availability Zone:** Select the *second* Availability Zone
 - **CIDR block:** 10.0.3.0/24

The subnet will have all IP addresses starting with **10.0.3.x**.

15. Choose **Create subnet**

The second *private* subnet was created.

You will now configure the Private Subnets to route internet-bound traffic to the NAT Gateway so that resources in the Private Subnet are able to connect to the Internet, while still keeping the resources private. This is done by configuring a *Route Table*.

A *route table* contains a set of rules, called *routes*, that are used to determine where network traffic is directed. Each subnet in a VPC must be associated with a route table; the route table controls routing for the subnet.

16. In the left navigation pane, choose **Route Tables**.
17. Select the route table with **Main = Yes** and **VPC = Lab VPC**. (Expand the *VPC ID* column if necessary to view the VPC name.)

18. In the **Name** column for this route table, choose the pencil then type Private Route Table and choose **Save**
19. In the lower pane, choose the **Routes** tab.

Note that **Destination 0.0.0.0/0** is set to **Target nat-xxxxxxx**. This means that traffic destined for the internet (0.0.0.0/0) will be sent to the NAT Gateway. The NAT Gateway will then forward the traffic to the internet.

This route table is therefore being used to route traffic from Private Subnets.

20. In the lower pane, choose the **Subnet Associations** tab.

You will now associate this route table to the Private Subnets.

21. Choose **Edit subnet associations**

22. Select both **Private Subnet 1** and **Private Subnet 2**.

You can expand the *Subnet ID* column to view the Subnet names.

23. Choose **Save associations**

You will now configure the Route Table that is used by the Public Subnets.

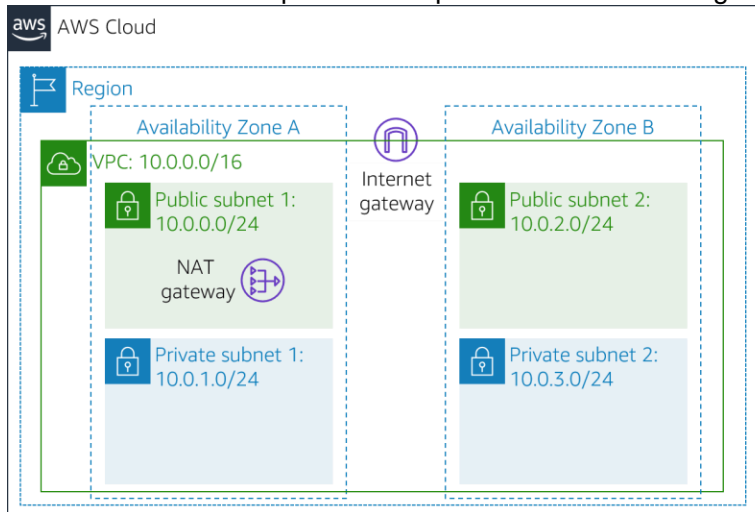
24. Select the route table with **Main = No** and **VPC = Lab VPC** (and deselect any other subnets).
25. In the **Name** column for this route table, choose the pencil then type Public Route Table, and choose **Save**
26. In the lower pane, choose the **Routes** tab.

Note that **Destination 0.0.0.0/0** is set to **Target igw-xxxxxxx**, which is the Internet Gateway. This means that internet-bound traffic will be sent straight to the internet via the Internet Gateway.

You will now associate this route table to the Public Subnets.

27. Choose the **Subnet associations** tab.
28. Choose **Edit subnet associations**
29. Select both **Public Subnet 1** and **Public Subnet 2**.
30. Choose **Save associations**

Your VPC now has public and private subnets configured in two Availability Zones:



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway

Task 3: Create a VPC Security Group

In this task, you will create a VPC security group, which acts as a virtual firewall. When you launch an instance, you associate one or more security groups with the instance. You can add rules to each security group that allow traffic to or from its associated instances.

31. In the left navigation pane, choose **Security Groups**.
32. Choose **Create security group** and then configure:
 - **Security group name:** Web Security Group
 - **Description:** Enable HTTP access
 - **VPC:** Lab VPC
33. In the **Inbound rules** pane, choose **Add rule**
34. Configure the following settings:
 - **Type:** HTTP
 - **Source:** Anywhere-IPv4
 - **Description:** Permit web request
35. Scroll to the bottom of the page and choose **Create security group**

You will use this security group in the next task when launching an Amazon EC2 instance.

Task 4: Launch a Web Server Instance

In this task, you will launch an Amazon EC2 instance into the new VPC. You will configure the instance to act as a web server.

36. In the search box to the right of **Services**, search for and choose **EC2** to open the EC2 console.
37. Choose **Launch instances**

38. Name the instance:

- Give it the name **Web Server 1**

When you name your instance, AWS creates a tag and associates it with the instance. A tag is a key value pair. The key for this pair is ***Name***, and the value is the name you enter for your EC2 instance.

39. Choose an AMI from which to create the instance:

- In the list of available *Quick Start* AMIs, keep the default **Amazon Linux** AMI selected.
- Also keep the default **Amazon Linux 2 AMI (HVM)** selected.

The type of *Amazon Machine Image (AMI)* you choose determines the Operating System that will run on the EC2 instance that you launch.

40. Choose an Instance type:

- In the *Instance type* panel, keep the default **t2.micro** selected.

The *Instance Type* defines the hardware resources assigned to the instance.

41. Select the key pair to associate with the instance:

- From the **Key pair name** menu, select **vockey**.

The vockey key pair you selected will allow you to connect to this instance via SSH after it has launched. Although you will not need to do that in this lab, it is still required to identify an existing key pair, or create a new one, when you launch an instance.

42. Configure the Network settings:

- Next to Network settings, choose **Edit**, then configure:
 - **Network:** *Lab VPC*
 - **Subnet:** *Public Subnet 2 (not Private!)*
 - **Auto-assign public IP:** *Enable*

This will ensure that the instance runs in Public Subnet 2 of the VPC that you created earlier in this lab.

- Next, you will configure the instance to use the *Web Security Group* that you created earlier.
 - Under Firewall (security groups), choose **Select an existing security group**.
 - For **Common security groups**, select **Web Security Group**.

This security group will permit HTTP access to the instance.

43. In the *Configure storage* section, keep the default settings.

44. Configure a script to run on the instance when it launches:

- Expand the **Advanced details** panel.
- Scroll to the bottom of the page and then copy and paste the code shown below into the `#!/bin/bash`

```
# Install Apache Web Server and PHP
yum install -y httpd mysql php
# Download Lab files
```

```
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-ACCLFO-2/2-lab2-
vpc/s3/lab-app.zip
unzip lab-app.zip -d /var/www/html/
# Turn on web server
chkconfig httpd on
service httpd start
```

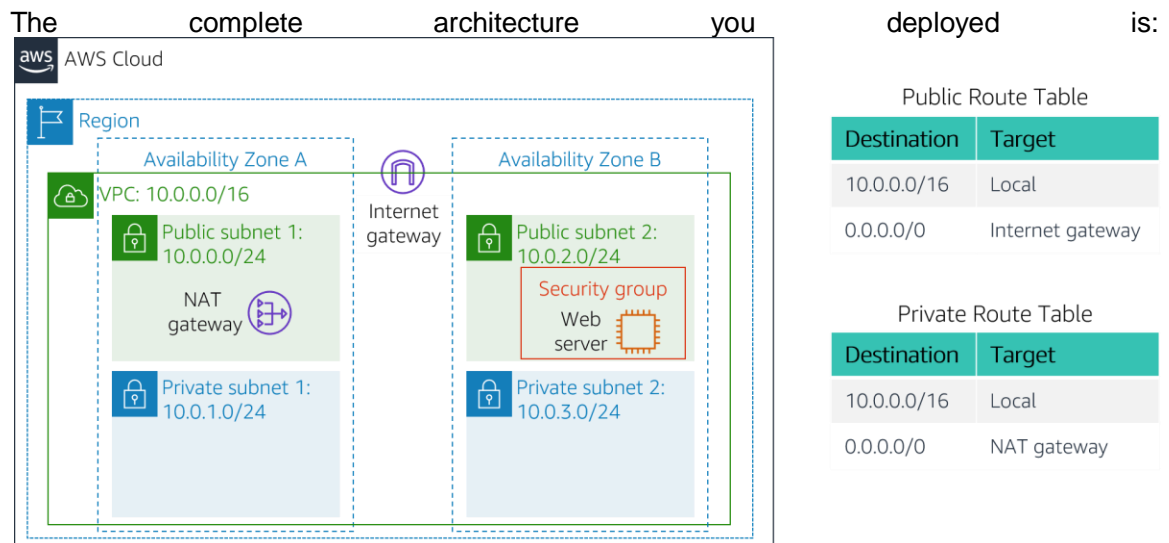
This script will run with root user permissions on the guest OS of the instance. It will run automatically when the instance launches for the first time. The script installs a web server, a database, and PHP libraries, and then it downloads and installs a PHP web application on the web server.

45. At the bottom of the **Summary** panel on the right side of the screen choose **Launch instance**

You will see a Success message.

46. Choose **View all instances**
47. Wait until **Web Server 1** shows *2/2 checks passed* in the **Status check** column. This may take a few minutes. Choose the refresh icon at the top of the page every 30 seconds or so to more quickly become aware of the latest status of the instance.
48. Select **Web Server 1**.
49. Copy the **Public IPv4 DNS** value shown in the **Details** tab at the bottom of the page.
50. Open a new web browser tab, paste the **Public DNS** value and press Enter.

You should see a web page displaying the AWS logo and instance meta-data values.



Lab 3: Introduction to Amazon EC2

Lab overview and objectives



This lab provides you with a basic overview of launching, resizing, managing, and monitoring an Amazon EC2 instance.

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

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

- Launch a web server with termination protection enabled
- Monitor Your EC2 instance
- Modify the security group that your web server is using to allow HTTP access
- Resize your Amazon EC2 instance to scale
- Explore EC2 limits
- Test termination protection
- Terminate your EC2 instance

Duration

This lab takes approximately **35 minutes** to complete.

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

Accessing the AWS Management Console

1. At the top of these instructions, choose **Start Lab** to launch your lab.

A Start Lab panel opens displaying the lab status.

2. Wait until you see the message "**Lab status: ready**", then choose the **X** to close the Start Lab panel.
3. At the top of these instructions, choose **AWS**

This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Choose on the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Launch Your Amazon EC2 Instance

In this task, you will launch an Amazon EC2 instance with *termination protection*. Termination protection prevents you from accidentally terminating an EC2 instance. You will deploy your instance with a User Data script that will allow you to deploy a simple web server

5. In the **AWS Management Console** in the search box to the right of **Services**, choose **Compute** and then choose **EC2**.

Note: Verify that your EC2 console is currently managing resources in the **N. Virginia** (us-east-1) region. You can verify this by looking at the drop down menu at the top of the screen, to the left of your username. If it does not already indicate N. Virginia, choose the N. Virginia region from the region menu before proceeding to the next step.

6. Choose **Launch instances**

Step 1: Name and tags

7. Give the instance the name `Web Server`.

The Name you give this instance will be stored as a tag. Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type — you can quickly identify a specific resource based on the tags you have assigned to it. Each tag consists of a Key and a Value, both of which you define. You can define multiple tags to associate with the instance if you want to.

In this case, the tag that will be created will consist of a *key* called `Name` with a *value* of `Web Server`

Step 2: Application and OS Images (Amazon Machine Image)

8. In the list of available *Quick Start* AMIs, keep the default **Amazon Linux** AMI selected.
9. Also keep the default **Amazon Linux 2 AMI (HVM)** selected.

An **Amazon Machine Image (AMI)** provides the information required to launch an instance, which is a virtual server in the cloud. An AMI includes:

- A template for the root volume for the instance (for example, an operating system or an application server with applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it is launched

The **Quick Start** list contains the most commonly-used AMIs. You can also create your own AMI or select an AMI from the AWS Marketplace, an online store where you can sell or buy software that runs on AWS.

Step 3: Instance type

10. In the *Instance type* panel, keep the default **t2.micro** selected.

Amazon EC2 provides a wide selection of *instance types* optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more *instance sizes*, allowing you to scale your resources to the requirements of your target workload.

The t2.micro instance type has 1 virtual CPU and 1 GiB of memory.

Note: You may be restricted from using other instance types in this lab.

Step 4: Key pair (login)

11. For **Key pair name - *required***, choose **vockey**.

Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To ensure you will be able to log in to the guest OS of the instance you create, you identify an existing key pair or create a new key pair when launching the instance. Amazon EC2 then installs the key on the guest OS when the instance is launched. That way, when you attempt to login to the instance and you provide the private key, you will be authorized to connect to the instance.

Note: In this lab you will not actually use the key pair you have specified to log into your instance.

Step 5: Network settings

12. Next to Network settings, choose **Edit**.
13. For **VPC**, select **Lab VPC**.

The Lab VPC was created using an AWS CloudFormation template during the setup process of your lab. This VPC includes two public subnets in two different Availability Zones.

Note: Keep the default subnet. This is the subnet in which the instance will run. Notice also that by default, the instance will be assigned a public IP address.

14. Under **Firewall (security groups)**, choose **Create security group** and configure:
 - **Security group name:** Web Server security group
 - **Description:** Security group for my web server

A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add *rules* to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group.

- Under **Inbound security group rules**, notice that one rule exists. **Remove** this rule.

Step 6: Configure storage

15. In the *Configure storage* section, keep the default settings.

Amazon EC2 stores data on a network-attached virtual disk called *Elastic Block Store*.

You will launch the Amazon EC2 instance using a default 8 GiB disk volume. This will be your root volume (also known as a 'boot' volume).

Step 7: Advanced details

16. Expand **Advanced details**.
17. For **Termination protection**, select **Enable**.

When an Amazon EC2 instance is no longer required, it can be *terminated*, which means that the instance is deleted and its resources are released. A terminated instance cannot be accessed again and the data that was on it cannot be recovered. If you want to prevent the instance from being accidentally terminated, you can enable *termination protection* for the instance, which prevents it from being terminated as long as this setting remains enabled.

18. Scroll to the bottom of the page and then copy and paste the code shown below into the **User data** box:

```
#!/bin/bash
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Hello From Your Web Server!</h1></html>' > /var/www/html/index.html
```

When you launch an instance, you can pass *user data* to the instance that can be used to perform automated installation and configuration tasks after the instance starts.

Your instance is running Amazon Linux 2. The *shell script* you have specified will run as the *root* guest OS user when the instance starts. The script will:

- Install an Apache web server (httpd)
- Configure the web server to automatically start on boot
- Run the Web server once it has finished installing
- Create a simple web page

Step 8: Launch the instance

19. At the bottom of the **Summary** panel on the right side of the screen choose **Launch instance**

You will see a Success message.

20. Choose **View all instances**

- Your **Web Server** should be selected.
- Review the information displayed in the **Details** tab. It includes information about the instance type, security settings and network settings.

The instance receives a *Public IPv4 DNS* that you can use to contact the instance from the Internet.

To view more information, drag the window divider upwards.

At first, the instance will appear in a *Pending* state, which means it is being launched. It will then change to *Initializing*, and finally to *Running*

21. Wait for your instance to display the following:

- **Instance State:** *Running*
- **Status Checks:** *2/2 checks passed*

Congratulations! You have successfully launched your first Amazon EC2 instance.

Task 2: Monitor Your Instance

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Elastic Compute Cloud (Amazon EC2) instances and your AWS solutions.

22. Choose the **Status Checks** tab.

With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues.

Notice that both the **System reachability** and **Instance reachability** checks have passed.

23. Choose the **Monitoring** tab.

This tab displays Amazon CloudWatch metrics for your instance. Currently, there are not many metrics to display because the instance was recently launched.

You can choose the three dots icon in any graph and select **Enlarge** to see an expanded view of the chosen metric.

Amazon EC2 sends metrics to Amazon CloudWatch for your EC2 instances. Basic (five-minute) monitoring is enabled by default. You can also enable detailed (one-minute) monitoring.

24. In the **Actions** menu towards the top of the console, select **Monitor and troubleshoot Get system log**.

The System Log displays the console output of the instance, which is a valuable tool for problem diagnosis. It is especially useful for troubleshooting kernel problems and service configuration issues that could cause an instance to terminate or become unreachable before its SSH daemon can be started. If you do not see a system log, wait a few minutes and then try again.

25. Scroll through the output and note that the HTTP package was installed from the **user data** that you added when you created the instance.
26. Choose **Cancel**.
27. Ensure **Web Server** is still selected. Then, in the **Actions** menu, select **Monitor and troubleshoot Get instance screenshot**.

This shows you what your Amazon EC2 instance console would look like if a screen were attached to it

If you are unable to reach your instance via SSH or RDP, you can capture a screenshot of your instance and view it as an image. This provides visibility as to the status of the instance, and allows for quicker troubleshooting.

28. Choose **Cancel**.

Congratulations! You have explored several ways to monitor your instance.

Task 3: Update Your Security Group and Access the Web Server

When you launched the EC2 instance, you provided a script that installed a web server and created a simple web page. In this task, you will access content from the web server.

29. Ensure **Web Server** is still selected. Choose the **Details** tab.
30. Copy the **Public IPv4 address** of your instance to your clipboard.
31. Open a new tab in your web browser, paste the IP address you just copied, then press **Enter**.

Question: Are you able to access your web server? Why not?

You are **not** currently able to access your web server because the *security group* is not permitting inbound traffic on port 80, which is used for HTTP web requests. This is a demonstration of using a security group as a firewall to restrict the network traffic that is

allowed in and out of an instance.

To correct this, you will now update the security group to permit web traffic on port 80.

32. Keep the browser tab open, but return to the **EC2 Console** tab.
33. In the left navigation pane, choose **Security Groups**.
34. Select **Web Server security group**.
35. Choose the **Inbound rules** tab.

The security group currently has no inbound rules

36. Choose **Edit inbound rules**, select **Add rule** and then configure:
 - **Type:** *HTTP*
 - **Source:** *Anywhere-IPv4*
 - Choose **Save rules**
37. Return to the web server tab that you previously opened and refresh the page.

You should see the message *Hello From Your Web Server!*

Congratulations! You have successfully modified your security group to permit HTTP traffic into your Amazon EC2 Instance.

Task 4: Resize Your Instance: Instance Type and EBS Volume

As your needs change, you might find that your instance is over-utilized (too small) or under-utilized (too large). If so, you can change the *instance type*. For example, if a *t2.micro* instance is too small for its workload, you can change it to an *m5.medium* instance. Similarly, you can change the size of a disk.

Stop Your Instance

Before you can resize an instance, you must *stop* it.

When you stop an instance, it is shut down. There is no runtime charge for a stopped EC2 instance, but the storage charge for attached Amazon EBS volumes remains.

38. On the **EC2 Management Console**, in the left navigation pane, choose **Instances**.

Web Server should already be selected.

39. In the **Instance State** menu, select **Stop instance**.
40. Choose **Stop**

Your instance will perform a normal shutdown and then will stop running.

41. Wait for the **Instance State** to display: *Stopped*.

Change The Instance Type

42. In the **Actions** menu, select **Instance settings Change instance type**, then configure:
 - **Instance Type:** *t2.small*
 - Choose **Apply**

When the instance is started again it will run as a *t2.small*, which has twice as much memory as a *t2.micro* instance. **NOTE:** You may be restricted from using other instance types in this lab.

Resize the EBS Volume

43. Choose the **Storage** tab, select the name of the Volume ID, then select the checkbox next to the volume that displays.
44. In the **Actions** menu, select **Modify volume**.

The disk volume currently has a size of 8 GiB. You will now increase the size of this disk.

45. Change the size to: 10 **NOTE:** You may be restricted from creating large Amazon EBS volumes in this lab.
46. Choose **Modify**
47. Choose **Modify** again to confirm and increase the size of the volume.

Start the Resized Instance

You will now start the instance again, which will now have more memory and more disk space.

49. In left navigation pane, choose **Instances**.
50. Select the **Web Server** instance.
51. In the **Instance state** menu, select **Start instance**.

Congratulations! You have successfully resized your Amazon EC2 Instance. In this task you changed your instance type from *t2.micro* to *t2.small*. You also modified your root disk volume from 8 GiB to 10 GiB.

Task 5: Explore EC2 Limits

Amazon EC2 provides different resources that you can use. These resources include images, instances, volumes, and snapshots. When you create an AWS account, there are default limits on these resources on a per-region basis.

52. In the left navigation pane, choose **Limits**.

Note: You may see some banner messages indicating that you cannot load some limits. You can safely ignore these messages.

53. From the **All limits** drop down list, choose **Running instances**.

Notice that there are limits on the number and types of instances that can run in a region. For example, there is a limit on the number of *Running On-Demand Standard...* instances that you can launch in this region. When launching instances, the request must not cause your usage to exceed the instance limits currently defined in that region.

You can request an increase for many of these limits.

Task 6: Test Termination Protection

You can delete your instance when you no longer need it. This is referred to as *terminating* your instance. You cannot connect to or restart an instance after it has been terminated. In this task, you will learn how to use *termination protection*.

54. In left navigation pane, choose **Instances**.
55. Select the **Web Server** instance and in the **Instance state** menu, select **Terminate instance**.
56. Then choose **Terminate**

Note that there is a message that says: *Failed to terminate the instance i-1234567xxx. The instance 'i-1234567xxx' may not be terminated. Modify its 'disableApiTermination' instance attribute and try again.*

This is a safeguard to prevent the accidental termination of an instance. If you really want to terminate the instance, you will need to disable the termination protection.

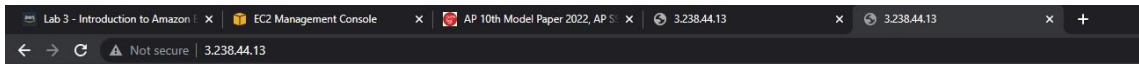
57. In the **Actions** menu, select **Instance settings Change termination protection**.
58. Remove the check next to **Enable**.
59. Choose **Save**

You can now terminate the instance.

60. Select the **Web Server** instance again and in the **Instance state** menu, select **Terminate instance**.
61. Choose **Terminate**

Congratulations! You have successfully tested termination protection and terminated your instance.

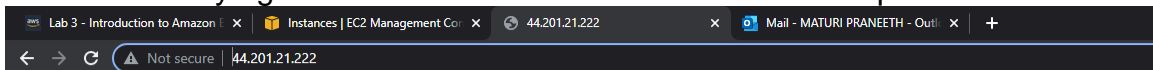
Output



Hello From Your Web Server!

Output II

After modifying the html code the output be like:



timetable

period	1	2	3	4	1:00PM 2:00Pm	5	6	7
Time/Day	9:30AM- 10:30AM	10:30AM- 11:20AM	11:20AM- 12:10PM	12:10PM- 1:00PM		2:00PM- 2:50PM	2:50PM- 3:40PM	3:40PM- 4:30PM
Mon	SE	DAA	SE	P&S	L B U R N E C A H K	DAA	DBMS	JAVA
TUE	P&S	JAVA	SE	DBMS		JAVA LAB		
WED	DBMS	DAA	DBMS	JAVA		P&S	PCS-2(apt)	
THU	DAA	SE	P&S	LIBRARY		PCS-2(ver)		DBMS
FRI	JAVA	SE	DAA	P&S		DBMS LAB		
SAT	DBMS	STAR R LAB				STAR R	JAVA	SPORTS

Lab 4: Working with EBS

Lab Overview



This lab focuses on Amazon Elastic Block Store (Amazon EBS), a key underlying storage mechanism for Amazon EC2 instances. In this lab, you will learn how to create an Amazon EBS volume, attach it to an instance, apply a file system to the volume, and then take a snapshot backup.

Topics covered

By the end of this lab, you will be able to:

- Create an Amazon EBS volume
- Attach and mount your volume to an EC2 instance
- Create a snapshot of your volume
- Create a new volume from your snapshot
- Attach and mount the new volume to your EC2 instance

Lab Pre-requisites

To successfully complete this lab, you should be familiar with basic Amazon EC2 usage and with basic Linux server administration. You should feel comfortable using the Linux command-line tools.

Duration

This lab will require approximately **30 minutes** to complete.

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

What is Amazon Elastic Block Store?

Amazon Elastic Block Store (Amazon EBS) offers persistent storage for Amazon EC2 instances. Amazon EBS volumes are network-attached and persist independently from the life of an instance. Amazon EBS volumes are highly available, highly reliable volumes that can be leveraged as an Amazon EC2 instances boot partition or attached to a running Amazon EC2 instance as a standard block device.

When used as a boot partition, Amazon EC2 instances can be stopped and subsequently restarted, enabling you to pay only for the storage resources used while maintaining your instance's state. Amazon EBS volumes offer greatly improved durability over local Amazon EC2 instance stores because Amazon EBS volumes are automatically replicated on the backend (in a single Availability Zone).

For those wanting even more durability, Amazon EBS provides the ability to create point-in-time consistent snapshots of your volumes that are then stored in Amazon Simple Storage Service (Amazon S3) and automatically replicated across multiple Availability Zones. These snapshots can be used as the starting point for new Amazon EBS volumes and can protect your data for long-term durability. You can also easily share these snapshots with co-workers and other AWS developers.

This lab guide explains basic concepts of Amazon EBS in a step-by-step fashion. However, it can only give a brief overview of Amazon EBS concepts. For further information, see the [Amazon EBS documentation](#).

Amazon EBS Volume Features

Amazon EBS volumes deliver the following features:

- **Persistent storage:** Volume lifetime is independent of any particular Amazon EC2 instance.
- **General purpose:** Amazon EBS volumes are raw, unformatted block devices that can be used from any operating system.
- **High performance:** Amazon EBS volumes are equal to or better than local Amazon EC2 drives.
- **High reliability:** Amazon EBS volumes have built-in redundancy within an Availability Zone.
- **Designed for resiliency:** The AFR (Annual Failure Rate) of Amazon EBS is between 0.1% and 1%.
- **Variable size:** Volume sizes range from 1 GB to 16 TB.

- **Easy to use:** Amazon EBS volumes can be easily created, attached, backed up, restored, and deleted.

Accessing the AWS Management Console

1. At the top of these instructions, click [Start Lab](#) to launch your lab.

A Start Lab panel opens displaying the lab status.

2. Wait until you see the message "**Lab status: ready**", then click the **X** to close the Start Lab panel.
3. At the top of these instructions, click [AWS](#)

This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Choose the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Create a New EBS Volume

In this task, you will create and attach an Amazon EBS volume to a new Amazon EC2 instance.

5. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
6. In the left navigation pane, choose **Instances**.

An Amazon EC2 instance named **Lab** has already been launched for your lab.

7. Note the **Availability Zone** of the instance. It will look similar to *us-east-1a*.
8. In the left navigation pane, choose **Volumes**.

You will see an existing volume that is being used by the Amazon EC2 instance. This volume has a size of 8 GiB, which makes it easy to distinguish from the volume you will create next, which will be 1 GiB in size.

9. Choose **Create volume** then configure:
 - **Volume Type:** *General Purpose SSD (gp2)*
 - **Size (GiB):** 1. **NOTE:** You may be restricted from creating large volumes.

- **Availability Zone:** Select the same availability zone as your EC2 instance.
- Choose **Add Tag**
- In the Tag Editor, enter:
 - **Key:** Name
 - **Value:** My Volume
- 10. Choose **Create Volume**

Your new volume will appear in the list, and will move from the *Creating* state to the *Available* state. You may need to choose **refresh** to see your new volume.

Task 2: Attach the Volume to an Instance

You can now attach your new volume to the Amazon EC2 instance.

11. Select **My Volume**.
12. In the **Actions** menu, choose **Attach volume**.
13. Choose the **Instance** field, then select the instance that appears (Lab).

Note that the **Device** field is set to `/dev/sdf`. You will use this device identifier in a later task.

14. Choose **Attach volume** The volume state is now *In-use*.

Task 3: Connect to Your Amazon EC2 Instance

Windows Users: Using SSH to Connect

These instructions are for Windows users only.

If you are using macOS or Linux, [skip to the next section](#).

15. Read through the three bullet points in this step before you start to complete the actions, because you will not be able to see these instructions when the Details panel is open.
 - Choose the **Details** drop down menu above these instructions you are currently reading, and then choose **Show**. A Credentials window will open.
 - Choose the **Download PPK** button and save the **labsuser.ppk** file. Typically your browser will save it to the Downloads directory.
 - Then exit the Details panel by choosing the **X**.
16. Download needed software.
 - You will use **PuTTY** to SSH to Amazon EC2 instances. If you do not have PuTTY installed on your computer, [download it here](#).
17. Open **putty.exe**
18. Configure PuTTY to not timeout:

- Choose **Connection**
- Set **Seconds between keepalives** to 30

This allows you to keep the PuTTY session open for a longer period of time.

19. Configure your PuTTY session:

- Choose **Session**
 - **Host Name (or IP address):** Copy and paste the **IPv4 Public IP address** for the instance. To find it, return to the EC2 Console and choose **Instances**. Check the box next to the instance and in the *Description* tab copy the **IPv4 Public IP** value.
 - Back in PuTTY, in the **Connection** list, expand **SSH**
 - Choose **Auth** (don't expand it)
 - Choose **Browse**
 - Browse to and select the labsuser.ppk file that you downloaded
 - Choose **Open** to select it
 - Choose **Open**
20. Choose **Yes**, to trust the host and connect to it.
21. When prompted **login as**, enter: ec2-user

This will connect you to the EC2 instance.

22. Windows Users: Choose here to skip ahead to the next task.

macOS and Linux Users

These instructions are for Mac/Linux users only. If you are a Windows user, [skip ahead to the next task.](#)

23. Read through all the instructions in this one step before you start to complete the actions, because you will not be able to see these instructions when the Details panel is open.
- Choose the **Details** drop down menu above these instructions you are currently reading, and then choose **Show**. A Credentials window will open.
 - Choose the **Download** button and save the **labsuser.pem** file.
 - Then exit the Details panel by choosing the **X**.
24. Open a terminal window, and change directory `cd` to the directory where the labsuser.pem file was downloaded.

For example, run this command, if it was saved to your Downloads directory:

```
cd ~/Downloads
```

25. Change the permissions on the key to be read only, by running this command:

```
chmod 400 labsuser.pem
```

26. Return to the AWS Management Console, and in the EC2 service, choose **Instances**.

The **Lab** instance should be selected.

27. In the *Details* tab, copy the **Public IPv4 address** value.
28. Return to the terminal window and run this command (replace **<public-ip>** with the actual public IP address you copied):

```
ssh -i labsuser.pem ec2-user@<public-ip>
```

29. Type **yes** when prompted to allow a first connection to this remote SSH server.

Because you are using a key pair for authentication, you will not be prompted for a password.

Task 4: Create and Configure Your File System

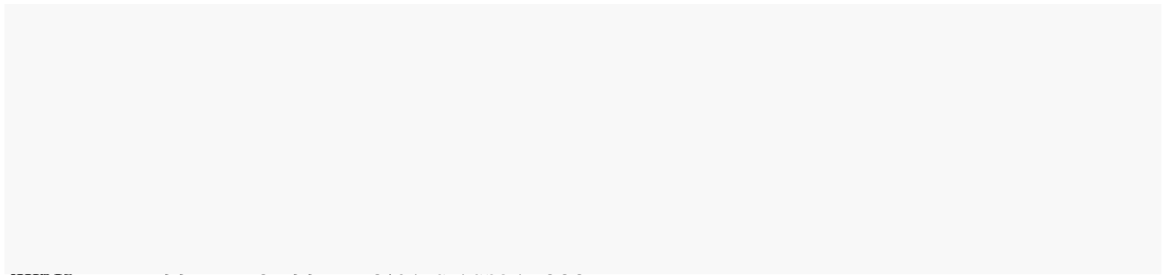
In this task, you will add the new volume to a Linux instance as an ext3 file system under the `/mnt/data-store` mount point.

If you are using PuTTY, you can paste text by right-clicking in the PuTTY window.

30. View the storage available on your instance:

```
df -h
```

You should see output similar to:



This is showing the original 8GB disk volume. Your new volume is not yet shown.

31. Create an ext3 file system on the new volume:

```
sudo mkfs -t ext3 /dev/sdf
```

32. Create a directory for mounting the new storage volume:


```
sudo mkdir /mnt/data-store
```

33. Mount the new volume:

```
sudo mount /dev/sdf /mnt/data-store
```

To configure the Linux instance to mount this volume whenever the instance is started, you will need to add a line to */etc/fstab*.

```
echo "/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab
```

34. View the configuration file to see the setting on the last line:

```
cat /etc/fstab
```

35. View the available storage again:

```
df -h
```

The output will now contain an additional line - */dev/xvdf*:

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	484M	0	484M	0%	/dev
tmpfs	492M	0	492M	0%	/dev/shm
tmpfs	492M	460K	491M	1%	/run
tmpfs	492M	0	492M	0%	/sys/fs/cgroup
/dev/xvda1	8.0G	1.5G	6.6G	19%	/
tmpfs	99M	0	99M	0%	/run/user/0
tmpfs	99M	0	99M	0%	/run/user/1000
/dev/xvdf	976M	1.3M	924M	1%	/mnt/data-store

36. On your mounted volume, create a file and add some text to it.

```
sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"
```

37. Verify that the text has been written to your volume.

```
cat /mnt/data-store/file.txt
```

Task 5: Create an Amazon EBS Snapshot

In this task, you will create a snapshot of your EBS volume.

You can create any number of point-in-time, consistent snapshots from Amazon EBS volumes at any time. Amazon EBS snapshots are stored in Amazon S3 with high durability. New Amazon EBS volumes can be created out of snapshots for cloning or restoring backups. Amazon EBS snapshots can also be easily shared among AWS users or copied over AWS regions.

38. In the **AWS Management Console**, choose **Volumes** and select **My Volume**.

39. In the **Actions** menu, select **Create snapshot**.

40. Choose **Add tag** then configure:

- **Key:** Name
- **Value:** My Snapshot
- Choose **Create snapshot**

41. In the left navigation pane, choose **Snapshots**.

Your snapshot is displayed. The status will first have a state of *Pending*, which means that the snapshot is being created. It will then change to a state of *Completed*.

Note: Only used storage blocks are copied to snapshots, so empty blocks do not occupy any snapshot storage space.

42. In your remote SSH session, delete the file that you created on your volume.

```
sudo rm /mnt/data-store/file.txt
```

43. Verify that the file has been deleted.

```
ls /mnt/data-store/
```

Your file has been deleted.

Task 6: Restore the Amazon EBS Snapshot

If you ever wish to retrieve data stored in a snapshot, you can **Restore** the snapshot to a new EBS volume.

Create a Volume Using Your Snapshot

44. In the **AWS Management Console**, select **My Snapshot**.
45. In the **Actions** menu, select **Create volume from snapshot**.
46. For **Availability Zone** Select the same availability zone that you used earlier.
47. Choose **Add tag** then configure:
 - **Key:** Name
 - **Value:** Restored Volume
 - Choose **Create volume**

Note: When restoring a snapshot to a new volume, you can also modify the configuration, such as changing the volume type, size or Availability Zone.

Attach the Restored Volume to Your EC2 Instance

48. In the left navigation pane, choose **Volumes**.
49. Select **Restored Volume**.
50. In the **Actions** menu, select **Attach volume**.
51. Choose the **Instance** field, then select the (Lab) instance that appears.

Note that the **Device** field is set to `/dev/sdg`. You will use this device identifier in a later task.

52. Choose **Attach volume**

The volume state is now *in-use*.

Mount the Restored Volume

53. Create a directory for mounting the new storage volume:

```
sudo mkdir /mnt/data-store2
```

54. Mount the new volume:

```
sudo mount /dev/sdg /mnt/data-store2
```

55. Verify that volume you mounted has the file that you created earlier.

```
ls /mnt/data-store2/
```

You should see file.txt.

Output

```
Amazon Linux 2
Kernel 5.10.109-104.500.amzn2.x86_64 on an x86_64

ip-172-31-5-164 login: [ 24.569154] xfs filesystem being remounted at /tmp supports timestamps until 2038 (0x7fffffff)
[ 24.592825] xfs filesystem being remounted at /var/tmp supports timestamps until 2038 (0x7fffffff)
_
```

Lab 5: Build Your DB Server and Interact With Your DB Using an App

Version 4.6.6 (TESS2)

This lab is designed to reinforce the concept of leveraging an AWS-managed database instance for solving relational database needs.

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, which allows you to focus on your applications and business. Amazon RDS provides you with six familiar database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB.

Objectives

After completing this lab, you can:

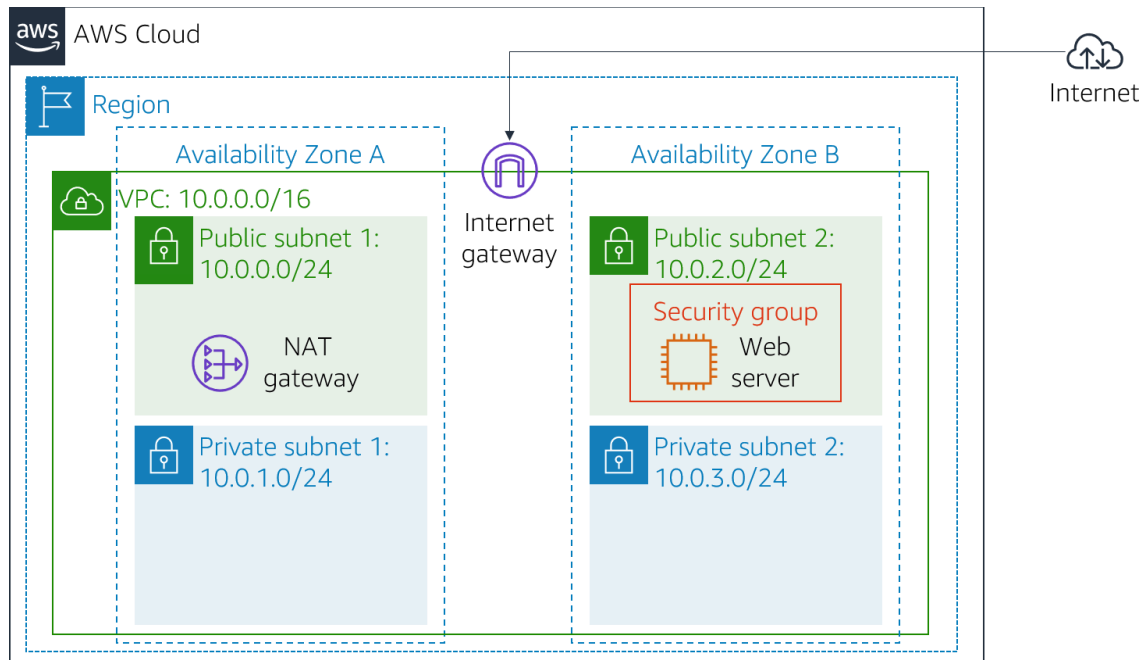
- Launch an Amazon RDS DB instance with high availability.
- Configure the DB instance to permit connections from your web server.
- Open a web application and interact with your database.

Duration

This lab takes approximately **30 minutes**.

Scenario You start with the following infrastructure:

At the end of the lab, this is the infrastructure:



Accessing the AWS Management Console

1. At the top of these instructions, choose Start Lab to launch your lab.

A Start Lab panel opens displaying the lab status.

2. Wait until you see the message "**Lab status: ready**", then choose the **X** to close the Start Lab panel.
3. At the top of these instructions, choose AWS

This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Choose on the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Create a Security Group for the RDS DB Instance

In this task, you will create a security group to allow your web server to access your RDS DB instance. The security group will be used when you launch the database instance.

5. In the **AWS Management Console**, on the Services menu, choose **VPC**.
6. In the left navigation pane, choose **Security Groups**.
7. Choose Create security group and then configure:
 - **Security group name:** DB Security Group
 - **Description:** Permit access from Web Security Group
 - **VPC:** *Lab VPC*

You will now add a rule to the security group to permit inbound database requests.

8. In the **Inbound rules** pane, choose Add rule

The security group currently has no rules. You will add a rule to permit access from the *Web Security Group*.

9. Configure the following settings:
 - **Type:** *MySQL/Aurora (3306)*
 - **CIDR, IP, Security Group or Prefix List:** Type *sg* and then select *Web Security Group*.

This configures the Database security group to permit inbound traffic on port 3306 from any EC2 instance that is associated with the *Web Security Group*.

10. Choose Create security group

You will use this security group when launching the Amazon RDS database.

Task 2: Create a DB Subnet Group

In this task, you will create a *DB subnet group* that is used to tell RDS which subnets can be used for the database. Each DB subnet group requires subnets in at least two Availability Zones.

11. On the Services menu, choose **RDS**.
12. In the left navigation pane, choose **Subnet groups**.

If the navigation pane is not visible, choose the menu icon in the top-left corner.

13. Choose Create DB Subnet Group then configure:
 - **Name:** DB-Subnet-Group
 - **Description:** DB Subnet Group
 - **VPC:** *Lab VPC*
14. Scroll down to the **Add Subnets** section.
15. Expand the list of values under **Availability Zones** and select the first two zones: **us-east-1a** and **us-east-1b**.
16. Expand the list of values under **Subnets** and select the subnets associated with the CIDR ranges **10.0.1.0/24** and **10.0.3.0/24**.

These subnets should now be shown in the **Subnets selected** table.

17. Choose Create

You will use this DB subnet group when creating the database in the next task.

Task 3: Create an Amazon RDS DB Instance

In this task, you will configure and launch a Multi-AZ Amazon RDS for MySQL database instance.

Amazon RDS **Multi-AZ** deployments provide enhanced availability and durability for Database (DB) instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB instance, Amazon RDS automatically creates a primary DB instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ).

18. In the left navigation pane, choose **Databases**.
19. Choose Create database

If you see **Switch to the new database creation flow** at the top of the screen, please choose it.

20. Select **MySQL**.
21. Under **Settings**, configure:
 - **DB instance identifier:** lab-db
 - **Master username:** main
 - **Master password:** lab-password
 - **Confirm password:** lab-password
22. Under **DB instance class**, configure:
 - Select **Burstable classes (includes t classes)**.

- Select *db.t3.micro*
- 23. Under **Storage**, configure:
 - **Storage type:** *General Purpose (SSD)*
 - **Allocated storage:** *20*
- 24. Under **Connectivity**, configure:
 - **Virtual Private Cloud (VPC):** *Lab VPC*
- 25. Under **Existing VPC security groups**, from the dropdown list:
 - Choose *DB Security Group*.
 - Deselect *default*.
- 26. Expand **Additional configuration**, then configure:
 - **Initial database name:** *lab*
 - Uncheck **Enable automatic backups**.
 - Uncheck **Enable encryption**
 - Uncheck **Enable Enhanced monitoring**.

This will turn off backups, which is not normally recommended, but will make the database deploy faster for this lab.

27. Choose Create database

Your database will now be launched.

If you receive an error that mentions "not authorized to perform: iam:CreateRole", make sure you unchecked *Enable Enhanced monitoring* in the previous step.

28. Choose **lab-db** (choose the link itself).

You will now need to wait **approximately 4 minutes** for the database to be available. The deployment process is deploying a database in two different Availability zones.

While you are waiting, you might want to review the [Amazon RDS FAQs](#) or grab a cup of coffee.

29. Wait until **Info** changes to **Modifying** or **Available**.

30. Scroll down to the **Connectivity & security** section and copy the **Endpoint** field.

It will look similar to: *lab-db.cgqg8lhnvnxv.us-west-2.rds.amazonaws.com*

31. Paste the Endpoint value into a text editor. You will use it later in the lab.

Task 4: Interact with Your Database

In this task, you will open a web application running on your web server and configure it to use the database.

32. To copy the **WebServer** IP address, choose on the Details drop down menu above these instructions, and then choose Show.
33. Open a new web browser tab, paste the *WebServer* IP address and press Enter.

The web application will be displayed, showing information about the EC2 instance.

34. Choose the **RDS** link at the top of the page.

You will now configure the application to connect to your database.

35. Configure the following settings:
 - **Endpoint:** Paste the Endpoint you copied to a text editor earlier
 - **Database:** lab
 - **Username:** main
 - **Password:** lab-password
 - Choose **Submit**

A message will appear explaining that the application is running a command to copy information to the database. After a few seconds the application will display an **Address Book**.

The Address Book application is using the RDS database to store information.

36. Test the web application by adding, editing and removing contacts.

The data is being persisted to the database and is automatically replicating to the second Availability Zone.

Output

Last name	First name	Phone	Email	Admin
Doe	Jane	010-110-1101	janed@someotheraddress.org	Edit Remove
Johnson	Roberto	123-456-7890	robertoj@someaddress.com	Edit Remove

Lab 6: Scale and Load Balance Your Architecture

Version 4.6.6 (TESS3) + custom change

This lab walks you through using the Elastic Load Balancing (ELB) and Auto Scaling services to load balance and automatically scale your infrastructure.

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve fault tolerance in your applications by seamlessly providing the required amount of load balancing capacity needed to route application traffic.

Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity out or in 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. Auto Scaling can also automatically increase the number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs. Auto Scaling is well suited to applications that have stable demand patterns or that experience hourly, daily, or weekly variability in usage.

Objectives

After completing this lab, you can:

- Create an Amazon Machine Image (AMI) from a running instance.
- Create a load balancer.
- Create a launch configuration and an Auto Scaling group.
- Automatically scale new instances within a private subnet
- Create Amazon CloudWatch alarms and monitor performance of your infrastructure.

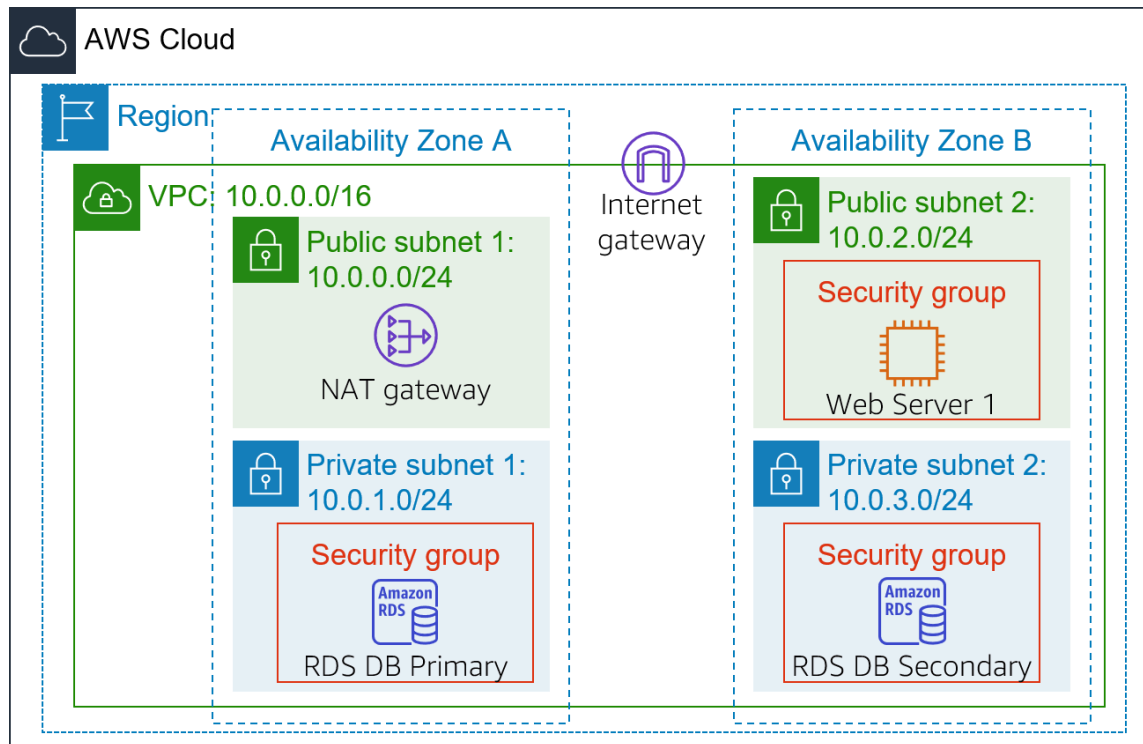
Duration

This lab takes approximately **30 minutes**.

Scenario

You start with the following infrastructure:

The final state of the infrastructure is:



Accessing the AWS Management Console

1. At the top of these instructions, click [Start Lab](#) to launch your lab.
A Start Lab panel opens displaying the lab status.
2. Wait until you see the message "**Lab status: in creation**", then click the **X** to close the Start Lab panel.

Note: It may take approximately 10 minutes or longer for the lab status to change to ready.

3. At the top of these instructions, click [AWS](#)

This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing

the site from opening pop-up windows. Click on the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Create an AMI for Auto Scaling

In this task, you will create an AMI from the existing *Web Server 1*. This will save the contents of the boot disk so that new instances can be launched with identical content.

5. In the **AWS Management Console**, on the **Services** menu, click **EC2**.

6. In the left navigation pane, click **Instances**.

First, you will confirm that the instance is running.

7. Wait until the **Status Checks** for **Web Server 1** displays *2/2 checks passed*. Click refresh to update.

You will now create an AMI based upon this instance.

8. Select **Web Server 1**.

9. In the **Actions** menu, click **Image and templates > Create image**, then configure:

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

10. Click **Create image**

A confirmation banner displays the **AMI ID** for your new AMI.

You will use this AMI when launching the Auto Scaling group later in the lab.

Task 2: Create a Load Balancer

In this task, you will create a load balancer that can balance traffic across multiple EC2 instances and Availability Zones.

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

Analysis: *Target Groups* define where to *send* traffic that comes into the Load Balancer. The Application Load Balancer can send traffic to multiple Target Groups based upon the URL of the incoming request, such as having requests from mobile apps going to a different set of servers. Your web application will use only one Target Group.

- Choose a target type: **Instances**
 - **Target group name**, enter: LabGroup
 - Select **Lab VPC** from the **VPC** drop-down menu.
12. Choose **Next**. The **Register targets** screen appears.

Note: *Targets* are the individual instances that will respond to requests from the Load Balancer.

You do not have any web application instances yet, so you can skip this step.

13. Review the settings and choose **Create target group**

14. In the left navigation pane, click **Load Balancers**.

15. At the top of the screen, choose **Create Load Balancer**.

Several different types of load balancer are displayed. You will be using an *Application Load Balancer* that operates at the request level (layer 7), routing traffic to targets — EC2 instances, containers, IP addresses and Lambda functions — based on the content of the request. For more information, see: Comparison of Load Balancers

16. Under **Application Load Balancer**, choose **Create**

17. Under **Load balancer name**, enter: LabELB

18. Scroll down to the **Network mapping** section, then:

- For **VPC**, select: **Lab VPC**

You will now specify which *subnets* the Load Balancer should use. The load balancer will be internet facing, so you will select both Public Subnets.

- Choose the **first** displayed Availability Zone, then select **Public Subnet 1** from the Subnet drop down menu that displays beneath it.
- Choose the **second** displayed Availability Zone, then select **Public Subnet 2** from the Subnet drop down menu that displays beneath it.

You should now have two subnets selected: **Public Subnet1** and **Public Subnet 2**. (If not, go back and try the configuration again.)

19. In the **Security groups** section:

- Choose the Security groups drop down menu and select **Web Security Group**
- Below the drop down menu, choose the **X** next to the default security group to remove it.

The **Web Security Group** security group should now be the only one that appears.

20. For the Listener HTTP:80 row, set the Default action to forward to **LabGroup**.

21. Scroll to the bottom and choose **Create load balancer**

The load balancer is successfully created.

- Choose **View load balancer**

The load balancer will show a state of *provisioning*. There is no need to wait until it is ready. Please continue with the next task.

Task 3: Create a Launch Configuration and an Auto Scaling Group

In this task, you will create a *launch configuration* for your Auto Scaling group. A launch configuration is a template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances such as the AMI, the instance type, a key pair, security group and disks.

22. In the left navigation pane, click **Launch Configurations**.

23. Click **Create launch configuration**

24. Configure these settings:

- **Launch configuration name:** LabConfig
- **Amazon Machine Image (AMI)** Choose *Web Server AMI*
- **Instance type:**
 - Choose **Choose instance type**
 - Select *t3.micro*
 - Choose **Choose**

Note: If you have launched the lab in the us-east-1 Region, select the **t2.micro** instance type. To find the Region, look in the upper right-hand corner of the Amazon EC2 console.

Note: If you receive the error message "Something went wrong. Please refresh and try again.", you may ignore it and continue with the exercise.

- **Additional configuration**

- **Monitoring:** Select *Enable EC2 instance detailed monitoring within CloudWatch*

This allows Auto Scaling to react quickly to changing utilization.

25. Under **Security groups**, you will configure the launch configuration to use the *Web Security Group* that has already been created for you.

- Choose **Select an existing security group**
- Select **Web Security Group**

26. Under **Key pair** configure:

- **Key pair options:** *Choose an existing key pair*
- **Existing key pair:** *vockey*
- Select **I acknowledge...**
- Click **Create launch configuration**

You will now create an Auto Scaling group that uses this Launch Configuration.

27. Select the checkbox for the *LabConfig* Launch Configuration.

28. From the **Actions** menu, choose *Create Auto Scaling group*

29. Enter Auto Scaling group name:

- **Name:** *Lab Auto Scaling Group*

30. Choose **Next**

31. On the **Network** page configure

- **Network:** *Lab VPC*

You can ignore the message regarding "No public IP address"

- **Subnet:** Select *Private Subnet 1 (10.0.1.0/24) and Private Subnet 2 (10.0.3.0/24)*

This will launch EC2 instances in private subnets across both Availability Zones.

32. Choose **Next**

33. In the **Load balancing - optional** pane, choose **Attach to an existing load balancer**

34. In the **Attach to an existing load balancer** pane, use the dropdown list to select *LabGroup*.

35. In the **Additional settings - optional** pane, select **Enable group metrics collection within CloudWatch**

This will capture metrics at 1-minute intervals, which allows Auto Scaling to react quickly to changing usage patterns.

36. Choose **Next**

37. Under **Group size**, configure:

- **Desired capacity:** 2
- **Minimum capacity:** 2
- **Maximum capacity:** 6

This will allow Auto Scaling to automatically add/remove instances, always keeping between 2 and 6 instances running.

38. Under **Scaling policies**, choose *Target tracking scaling policy* and configure:

- **Lab policy name:** LabScalingPolicy
- **Metric type:** *Average CPU Utilization*
- **Target value:** 60

This tells Auto Scaling to maintain an *average* CPU utilization *across all instances* at 60%. Auto Scaling will automatically add or remove capacity as required to keep the metric at, or close to, the specified target value. It adjusts to fluctuations in the metric due to a fluctuating load pattern.

39. Choose **Next**

Auto Scaling can send a notification when a scaling event takes place. You will use the default settings.

40. Choose **Next**

Tags applied to the Auto Scaling group will be automatically propagated to the instances that are launched.

41. Choose **Add tag** and Configure the following:

- **Key:** Name
- **Value:** Lab Instance

42. Click **Next**

43. Review the details of your Auto Scaling group, then click **Create Auto Scaling group**. If you encounter an error **Failed to create Auto Scaling group**, then click **Retry Failed Tasks**.

Your Auto Scaling group will initially show an instance count of zero, but new instances will be launched to reach the **Desired** count of 2 instances.

Task 4: Verify that Load Balancing is Working

In this task, you will verify that Load Balancing is working correctly.

44. In the left navigation pane, click **Instances**.

You should see two new instances named **Lab Instance**. These were launched by Auto Scaling.

If the instances or names are not displayed, wait 30 seconds and click **refresh** in the top-right.

First, you will confirm that the new instances have passed their Health Check.

45. In the left navigation pane, click **Target Groups** (in the *Load Balancing* section).

46. Choose *LabGroup*

47. Click the **Targets** tab.

Two **Lab Instance** targets should be listed for this target group.

48. Wait until the **Status** of both instances transitions to *healthy*. Click **Refresh** in the upper-right to check for updates.

Healthy indicates that an instance has passed the Load Balancer's health check. This means that the Load Balancer will send traffic to the instance.

You can now access the Auto Scaling group via the Load Balancer.

49. In the left navigation pane, click **Load Balancers**.

50. In the lower pane, copy the **DNS name** of the load balancer, making sure to omit "(A Record)".

It should look similar to: *LabELB-1998580470.us-west-2.elb.amazonaws.com*

51. Open a new web browser tab, paste the DNS Name you just copied, and press Enter.

The application should appear in your browser. This indicates that the Load Balancer received the request, sent it to one of the EC2 instances, then passed back the result.

Task 5: Test Auto Scaling

You created an Auto Scaling group with a minimum of two instances and a maximum of six instances. Currently two instances are running because the

minimum size is two and the group is currently not under any load. You will now increase the load to cause Auto Scaling to add additional instances.

52. Return to the AWS management console, but do not close the application tab — you will return to it soon.

53. On the **Services** menu, click **CloudWatch**.

54. In the left navigation pane, choose **All alarms**.

Two alarms will be displayed. These were created automatically by the Auto Scaling group. They will automatically keep the average CPU load close to 60% while also staying within the limitation of having two to six instances.

Note: Please follow these steps only if you do not see the alarms in 60 seconds.

- On the **Services** menu, click **EC2**.
- In the left navigation pane, choose **Auto Scaling Groups**.
- Select **Lab Auto Scaling Group**.
- In the bottom half of the page, choose the **Automatic Scaling** tab.
- Select **LabScalingPolicy**.
- Click **Actions** and **Edit**.
- Change the **Target Value** to 50.
- Click **Update**.
- On the **Services** menu, click **CloudWatch**.
- In the left navigation pane, click **All alarms** and verify you see two alarms.

55. Click the **OK** alarm, which has *AlarmHigh* in its name.

If no alarm is showing **OK**, wait a minute then click refresh in the top-right until the alarm status changes.

The **OK** indicates that the alarm has *not* been triggered. It is the alarm for **CPU Utilization > 60**, which will add instances when average CPU is high. The chart should show very low levels of CPU at the moment.

You will now tell the application to perform calculations that should raise the CPU level.

56. Return to the browser tab with the web application.

57. Click **Load Test** beside the AWS logo.

This will cause the application to generate high loads. The browser page will automatically refresh so that all instances in the Auto Scaling group will generate load. Do not close this tab.

58. Return to browser tab with the **CloudWatch** console.

In less than 5 minutes, the **AlarmLow** alarm should change to **OK** and the **AlarmHigh** alarm status should change to *In alarm*.

You can click Refresh in the top-right every 60 seconds to update the display.

You should see the **AlarmHigh** chart indicating an increasing CPU percentage. Once it crosses the 60% line for more than 3 minutes, it will trigger Auto Scaling to add additional instances.

59. Wait until the **AlarmHigh** alarm enters the *In alarm* state.

You can now view the additional instance(s) that were launched.

60. On the **Services** menu, click **EC2**.

61. In the left navigation pane, click **Instances**.

More than two instances labeled **Lab Instance** should now be running. The new instance(s) were created by Auto Scaling in response to the Alarm.

Task 6: Terminate Web Server 1

In this task, you will terminate *Web Server 1*. This instance was used to create the AMI used by your Auto Scaling group, but it is no longer needed.

62. Select **Web Server 1** (and ensure it is the only instance selected).

63. In the **Instance state** menu, click **Instance State > Terminate Instance**.

64. Choose **Terminate**

Output

The screenshot shows the AWS Management Console for the EC2 service. A green banner at the top of the console area states "Successfully terminated i-0bd924bbb269e957b". Below this, the "Instances (1/5)" section is visible, showing a table of EC2 instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP Address. The first instance, "Web Server 1" (ID: i-0bd924bbb269e957b), is in the "Terminated" state. The other four instances, including "Bastion Host" and three "Lab Instance"s, are in the "Running" state.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP Address
Web Server 1	i-0bd924bbb269e957b	Terminated	t2.micro	2/2 checks passed	No alarms	us-east-1a	-
Bastion Host	i-0a8cd00a2ab155cd9	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	-
Lab Instance	i-0d3c7435802a91f66	Running	t3.micro	2/2 checks passed	No alarms	us-east-1a	-
Lab Instance	i-054047cc3a7e59fa9	Running	t3.micro	2/2 checks passed	No alarms	us-east-1b	-
Lab Instance	i-0fa09337fcf151219	Running	t3.micro	2/2 checks passed	No alarms	us-east-1b	-