

MASTERING AWS CLOUD FROM BEGINNER TO PRO



Table Of Content

Introduction to Data Center Structure

What is Cloud Computing?

Benefits of Cloud Computing

Types of Cloud Computing

Introduction to AWS Free Tier

How to Create an AWS Free Tier Account

How to Create an IAM User in AWS

How to Set Up CloudWatch in AWS

Introduction to AWS Regions and Availability Zones

Introduction to Availability Zones

Introduction to AWS EC2

How to Create an EC2 Instance

How to Create a Key Pair

How to Create a Security Group

Deploying a Web Server on an EC2 Instance

Introduction to Elastic IP

Introduction to AWS CLI

Introduction to Elastic Block Store (EBS)

How to Create an EBS Volume

Introduction to EBS Snapshots

Introduction to AWS ELB

How to Set Up AWS ELB

Introduction to AWS CloudWatch

Introduction to AWS EFS

Introduction to AWS Auto Scaling

How to Create an AWS Auto Scaling Group

Introduction to Amazon S3

How to Use AWS S3

Introduction to AWS RDS

Introduction to Data Center Structure

Imagine a data center with a number of computers and servers.



Hundreds and thousands of servers run side by side to provide compute resources to an organization and its branches.

Imagine all these servers are virtualized and a hypervisor is installed on each of them, so that the virtualization team can create virtual machines on them. If one of the employees in the company needs a virtual machine or a computer to do their work, they have to contact the virtualization team to get the required compute resources.

The Virtualization Team works in the data center, and most large companies require more than one administrator to manage their virtualization platform.

This is virtualization — something that existed before cloud computing.

Now you see similar virtualization platforms, but instead of contacting the virtualization team to get compute resources, you have a self-help portal, a website, or often a command-line interface to access the virtualized platform.

So if you need a virtual machine, virtual storage, or any other virtual service, you simply need to log in and create it yourself — this is called **cloud computing**, where access to your virtual resources is through the network. This means you can connect to your cloud portal via API from anywhere and at any time, and create, manage, or maintain your virtual resources.

Now, if this setup is for a single organization, it is known as a **private cloud**. But if it is publicly available so that anyone can sign up with a cloud provider to access it, it is known as a **public cloud provider**.

AWS, Azure, and Google Cloud are major names in the public cloud environment.

In this course, we use AWS Cloud Computing.

What is Cloud Computing?

It is the on-demand delivery of IT resources over the internet, where you pay only for what you use. Therefore, there is no need to purchase hardware.

You can access any computing power, storage, or database from a cloud provider like AWS.

In today's world, more than 90% of companies use cloud computing — and they do so because of the many benefits it offers.

Benefits of Cloud Computing

Some of the most important benefits of cloud computing include:

-Agility

You can easily access a wide range of cloud technologies and quickly build anything you can imagine.

You just need to create an account in AWS and start building your compute resources.

-Elasticity

You can easily scale your resources up or down based on your organization's needs.

You can access resources whenever you need and in whatever amount you require. However, since you grow or shrink your resources, you must keep control over your costs — something most people tend to forget.

-Cost Saving

Since you pay based on usage and consumption, it helps you save on costs.

-Deploy Globally in Minute

You can make your services global in less than a minute. For example, AWS infrastructure is spread across the world, and with just a few clicks, you can launch your application in multiple physical locations.

Placing your application closer to the end user can reduce latency and enhance the user experience.

Types of Cloud Computing

Cloud computing consists of the following three main types:

-Infrastructure as a Service

It is a virtual machine, and you can manage the virtual machine's operating system — this is what AWS's EC2 service provides.

-Platform as a Service

In this service, you don't have to worry about the virtual platform — you simply choose the platform you need. For example, if you need an Oracle Database, there's no need to create a virtual machine.

In this case, you can use the AWS RDS service, and AWS will provision and set up everything needed to run the Oracle Database — so you don't have to deal with the setup process yourself.

-Software as a Service

It is one of the cloud-based services that is easily accessible and usable — you just need to subscribe to it and start using it.

Introduction to AWS Free Tier

AWS is essentially an infrastructure that is spread across the globe, and you can build your own infrastructure on top of it. To do this, you need an AWS Free Tier account.

AWS is one of the largest cloud providers in the world today, holding nearly 46% of the market share.

In this section, we intend to create an AWS Free Tier account, then modify its settings. Next, we will create an IAM user for accessing AWS services, enable MFA (Multi-Factor Authentication) on our AWS accounts, and configure a billing alarm. This alarm will notify us if usage exceeds the allowed limit — meaning if you create a service and keep it running without deleting it, charges will apply. However, by setting up a billing alarm, you'll receive a notification once your usage reaches the defined threshold.

Finally, you can configure a certificate for your domain within AWS using the **ACM (AWS Certificate Manager)** service. This allows you to use **HTTPS** for secure access to your web services.

How to Create an AWS Free Tier Account

Search for “AWS Free Tier” on Google and then enter the Amazon website from the search results.

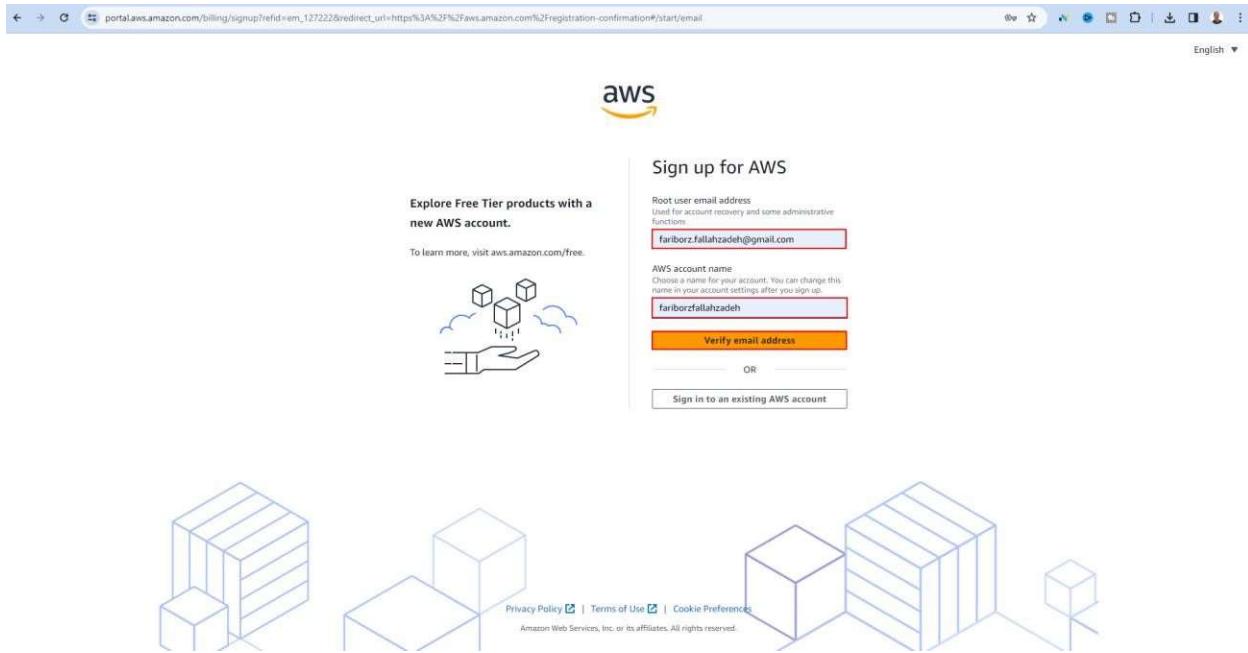
Google search results for "aws free tier". The top result is a link to the AWS Free Cloud Computing Services page, which is highlighted with a red box. Below the search bar, there's a "People also ask" section with several collapsed questions:

- Is the AWS free tier really free?
- Is AWS free for 1 year?
- What happens after 12 months of AWS free tier?
- How many hours is AWS free tier free?

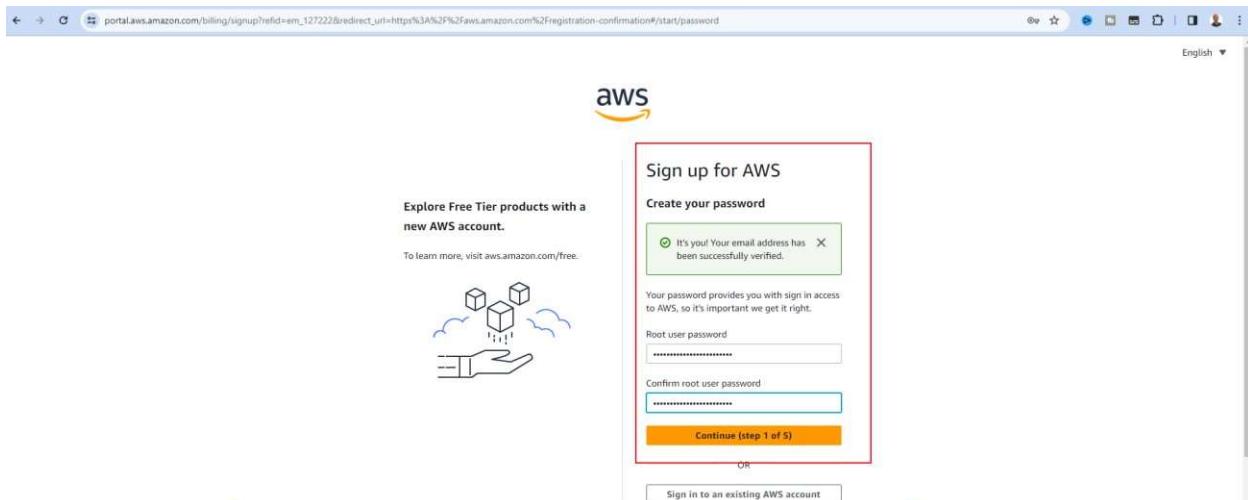
At this stage, click on the **Create Free Account** button.

AWS Free Tier landing page. The "Create a Free Account" button is highlighted with a red box.

At this stage, you need to provide a **Root User Email Address** and an **Account Name** to access the AWS Management Console.



At this stage, your email address needs to be verified. An **Email Verification Code** will be sent to your email, which you must enter in the verification field, then click the **Verify** button. After that, you will need to set a **Root Password**.



Next, you need to enter your **billing information** and **credit or debit card details**.

After completing your bank account and card information, you need to select a **Support Plan**. Since you are going to use this account for testing and learning purposes, it's best to choose the **Basic Support** plan, which is free.

Select a support plan

Choose a support plan for your business or personal account. [Compare plans and pricing examples](#)
 You can change your plan anytime in the AWS Management Console.

<input checked="" type="radio"/> Basic support - Free <ul style="list-style-type: none">Recommended for new users just getting started with AWS24x7 self-service access to AWS resourcesFor account and billing issues onlyAccess to Personal Health Dashboard & Trusted Advisor 	<input type="radio"/> Developer support - From \$29/month <ul style="list-style-type: none">Recommended for developers experimenting with AWSEmail access to AWS Support during business hours12 (business)-hour response times 	<input type="radio"/> Business support - From \$100/month <ul style="list-style-type: none">Recommended for running production workloads on AWS24x7 tech support via email, phone, and chat1-hour response timesFull set of Trusted Advisor best-practice recommendations 
--	--	---

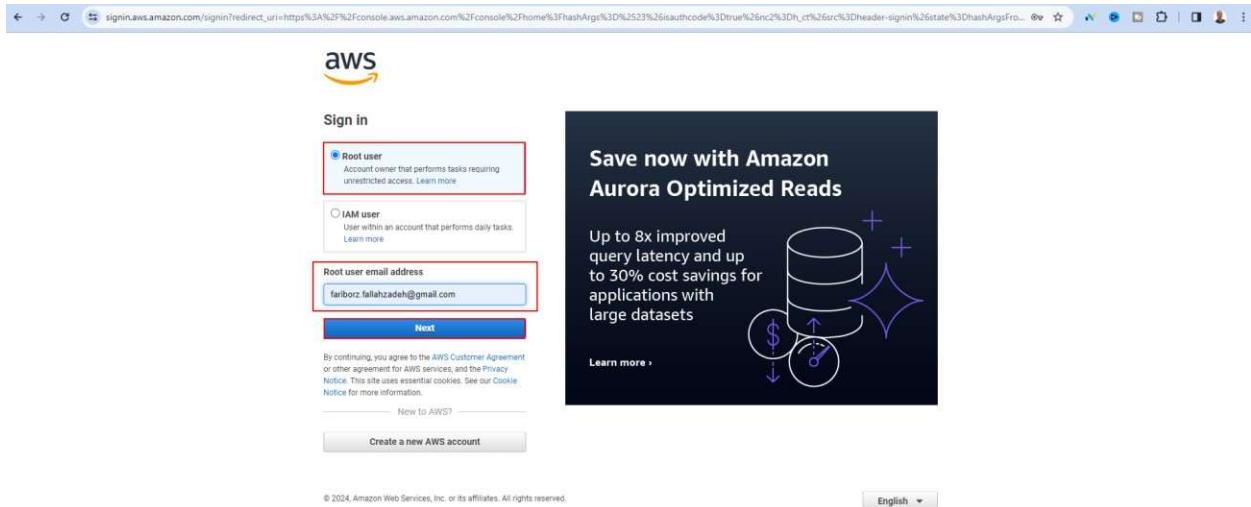
Finally, click on the **Complete Sign Up** button to finish the account creation process.

After completing the account creation steps, you should see the following message, which means your account has been successfully created.



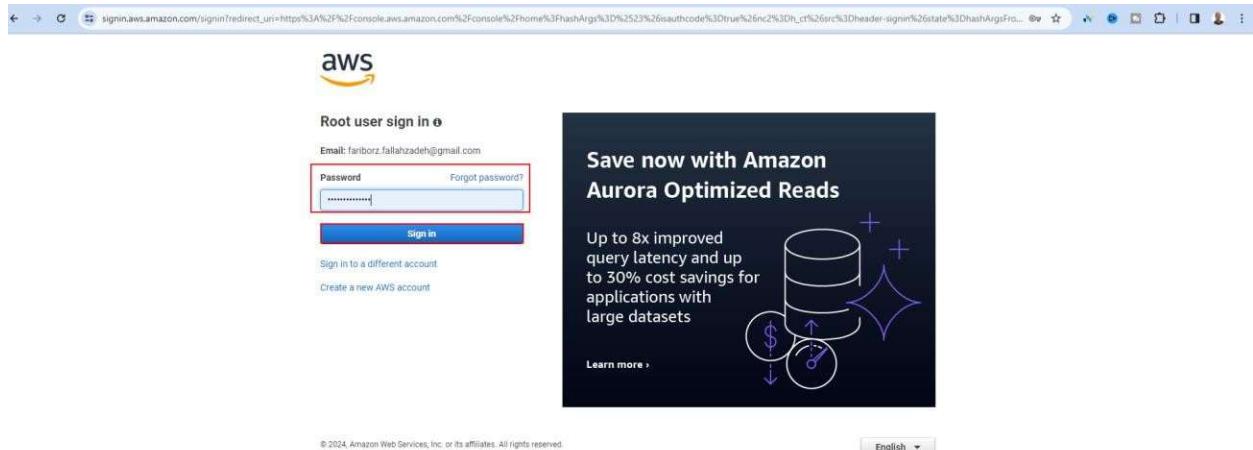
Then, click the button at the top to enter the **AWS Management Console**.

At this stage, you need to log in to your **AWS Management Console** using the **Root User**. The root user account has the highest level of access to the AWS Management Console.

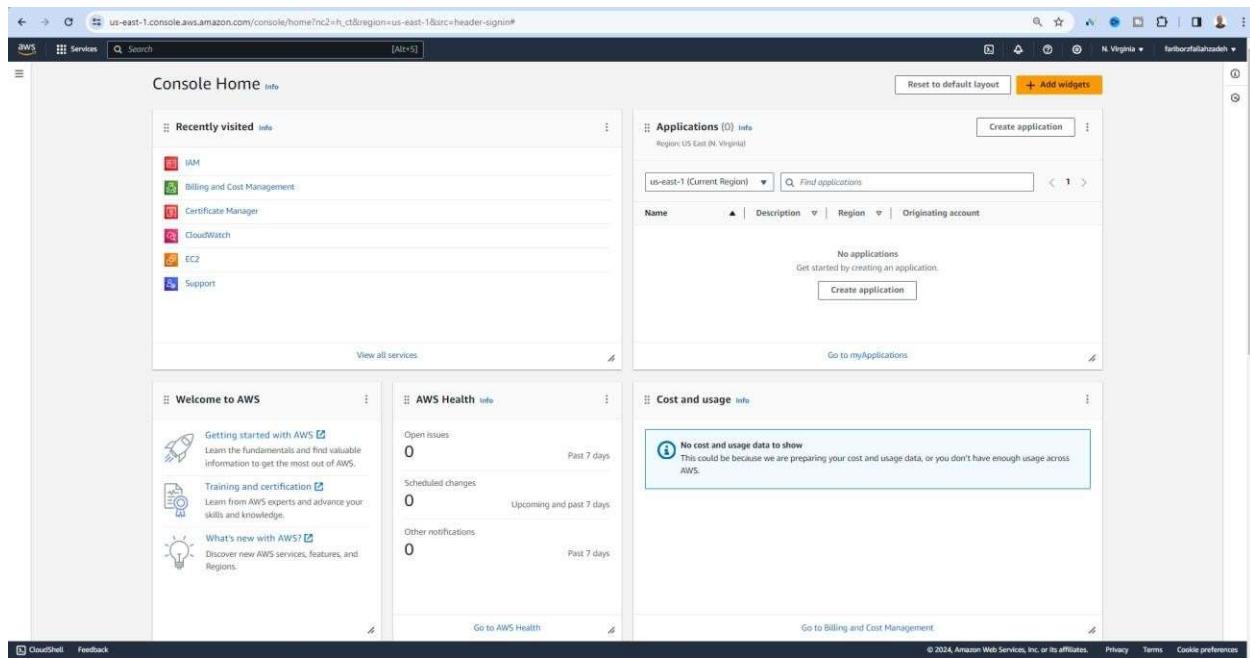


After entering the **Root Username**, click the **Next** button.

At this stage, you need to enter your **Root Password** and then click the **Sign in** button to access the AWS Management Console.

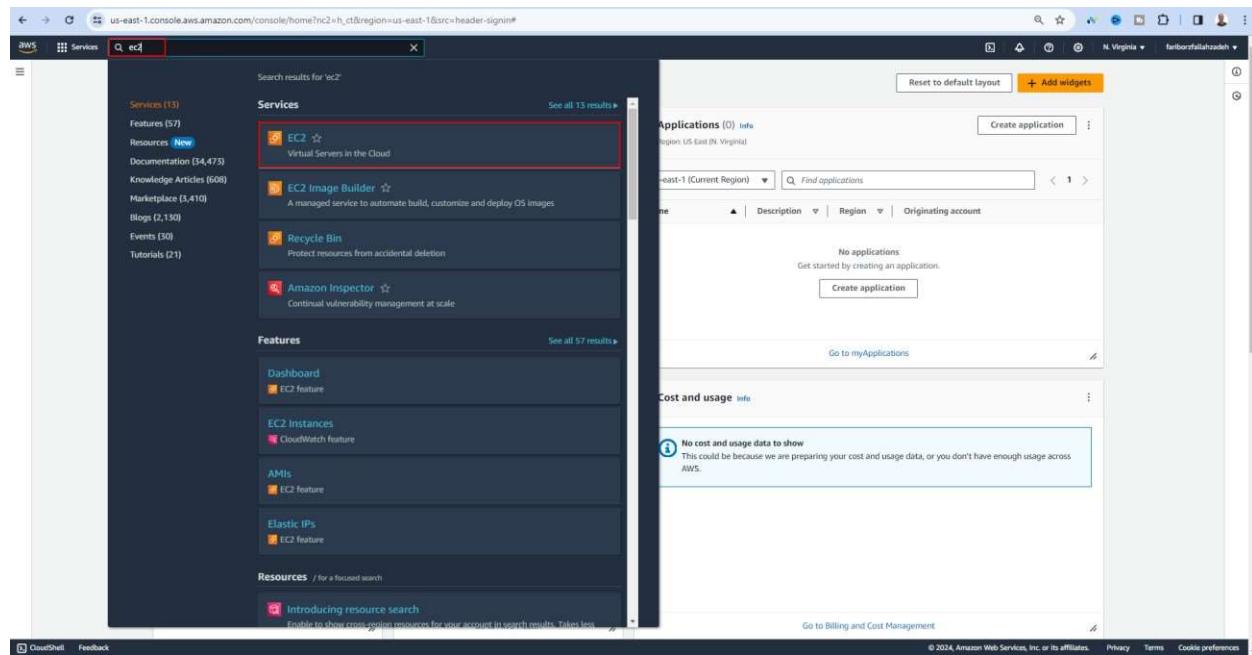


After logging in, as you can see, you are now inside the **AWS Management Console**.



To make sure your account is activated, simply check one of the available services in your account — for example, the **EC2** service, which is used to create virtual machines.

At this stage, type **EC2** in the search bar and then click on it to enter the **EC2 Instances** page.



The screenshot shows the AWS Management Console search results for the term "ec2". The search bar at the top contains "ec2". The results are categorized into Services, Features, and Resources.

- Services:**
 - EC2** (Virtual Servers In the Cloud) - This item is highlighted with a red box.
 - EC2 Image Builder
 - Recycle Bin
 - Amazon Inspector
- Features:**
 - Dashboard (EC2 feature)
 - EC2 Instances (CloudWatch feature)
 - AMIs (EC2 feature)
 - Elastic IPs (EC2 feature)
- Resources:**
 - Introducing resource search

On the right side of the screen, there is a separate window titled "Applications (0) Info" showing the "No applications" status. Below it is the "Cost and usage" section with a message indicating "No cost and usage data to show".

If all the values on this page are zero, it means your account is **active** and **enabled**.

The screenshot shows the AWS EC2 Home page for the US East (N. Virginia) Region. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and more. The main content area is titled 'Resources' and displays various Amazon EC2 metrics. A large red box highlights the following resource counts:

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

Below the resources section, there's a 'Launch instance' button and a 'Service health' panel. The 'Service health' panel shows the region as 'US East (N. Virginia)' with no issues. The 'Zones' section lists available zones: us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, and us-east-1f. The 'Account attributes' section shows the default VPC (vpc-0281d25ad35f8e6cf). The 'Explore AWS' section includes a link to 'Enable Best Price-Performance with AWS Graviton2'. The bottom right corner includes copyright information: © 2024, Amazon Web Services, Inc. or its affiliates.

If you see an error at the top or your account is not activated, you can submit a **support ticket** to AWS Support and explain the issue so it can be resolved as quickly as possible.

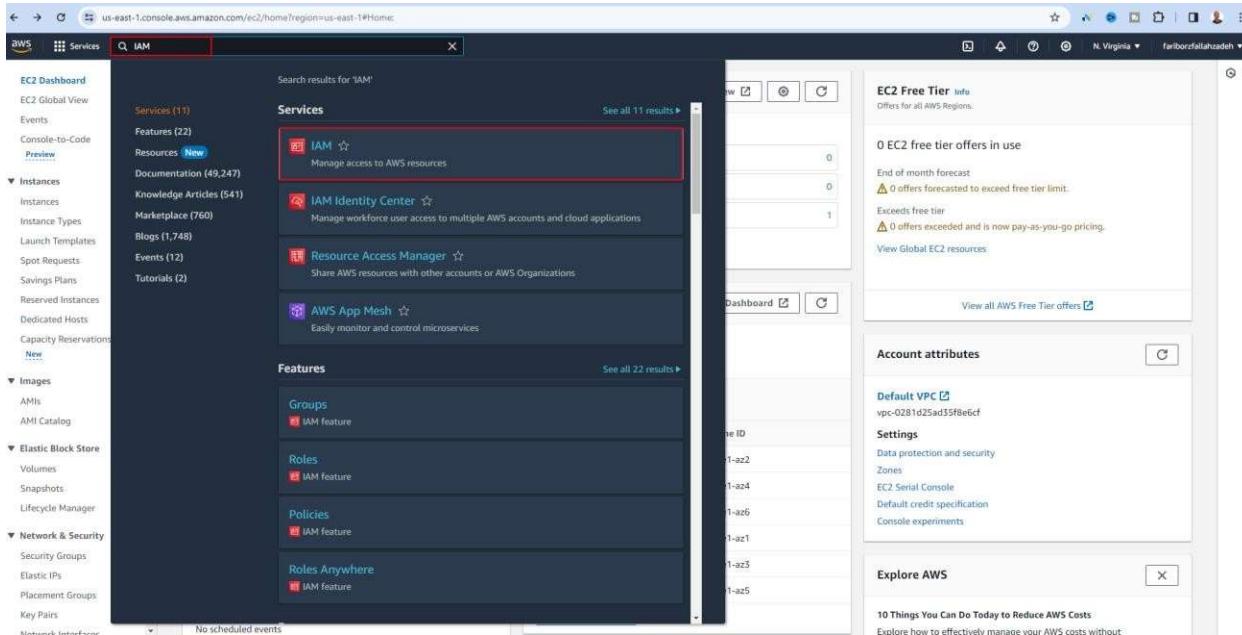
In the section below, you can access the **AWS Support Center** on the website.

The screenshot shows the AWS EC2 Dashboard for the US East (N. Virginia) Region. On the right side, a context menu is open over the 'EC2 Home' link, with the 'Support Center' option highlighted. The 'Support Center' menu includes links for 'Expert Help', 'Documentation', 'Training', 'Getting Started Resource Center', and 'Send feedback'. The main dashboard displays various EC2 resource statistics and management options like 'Launch instance', 'Migrate a server', and 'Scheduled events'.

It is recommended not to use the **Root User** to manage AWS services. Instead, it's better to use another user called an **AWS IAM User** for managing services.

How to Create an IAM User in AWS

To create an IAM user, type **IAM** in the search bar and then click on it.



The first thing you need to do before creating an IAM user is to **enable MFA (Multi-Factor Authentication)** for the Root User. You should install an app called **Google Authenticator** on your phone. This app provides a 6-digit code used for authentication.

Each time you enter your password, you must also enter this code for authentication. This mechanism uses **OTP (One-Time Password)**, which means a unique numeric password is generated for you at each login attempt.

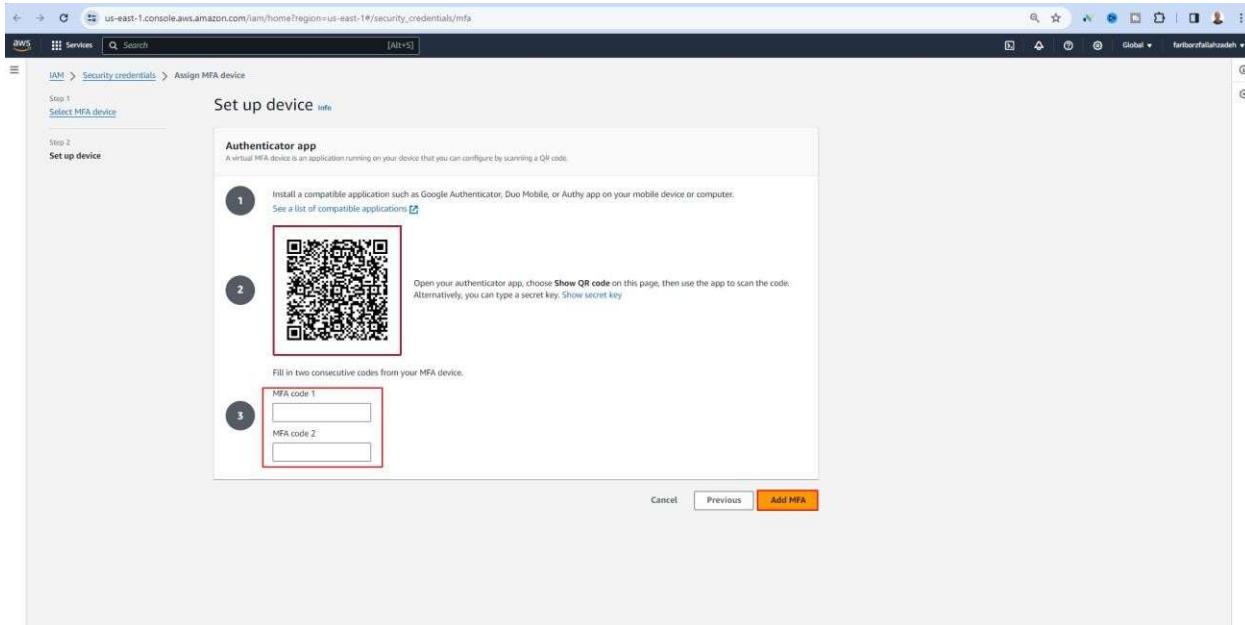
First, click on the **Add MFA** button.

The screenshot shows the AWS IAM Dashboard. On the left sidebar, under 'Access management', there is a red box around the 'Add MFA for root user' section. This section contains a warning icon and the text: 'Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.' To the right of this text is a red-bordered 'Add MFA' button. The rest of the dashboard includes sections for IAM resources (User groups: 0, Users: 0, Roles: 2, Policies: 0, Identity providers: 0), What's new (with a link to updates), and various links for AWS Account, Quick Links, Tools, and Additional information.

In the **Assign MFA Device** section, click to proceed and select an MFA device. Here, you should use your **phone** and assign it a name. Choose the **Authenticator app** option, then click the **Next** button.

The screenshot shows the 'Select MFA device' step of the 'Assign MFA device' wizard. It is Step 1 of 2. The 'MFA device name' field is filled with 'Myphone'. The 'MFA device' section shows three options: 'Authenticator app' (selected, highlighted with a red border), 'Security Key', and 'Hardware TOTP token'. At the bottom right are 'Cancel' and 'Next' buttons.

At this stage, open the **Google Authenticator** app on your mobile device, tap the + button, and in AWS click on **Show QR Code**. Scan the QR code with Google Authenticator, then enter the **two MFA codes** generated by the app into the fields provided. Finally, click on the **Add MFA** button.



Then, in the **IAM User** section, click on the **Dashboard** link and press the **Refresh** button. As you can see, **MFA has now been added** to the Root User.

The screenshot shows the AWS IAM Dashboard. On the left, the navigation menu includes 'Identity and Access Management (IAM)', 'Dashboard', 'Access management' (with 'User groups', 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings'), 'Access reports' (with 'Access Analyzer', 'External access', 'Unused access', 'Analyzer settings', 'Credential report', 'Organization activity', 'Service control policies (SCPs)'), and 'Related consoles' (with 'IAM Identity Center' and 'AWS Organizations'). The main content area is titled 'IAM Dashboard' and contains sections for 'Security recommendations', 'IAM resources', and 'What's new'. The 'Security recommendations' section highlights that 'Root user has MFA' (Having multi-factor authentication (MFA) for the root user improves security for this account). The 'IAM resources' section shows 0 User groups, 0 Users, 2 Roles, 0 Policies, and 0 Identity providers. The 'What's new' section lists recent changes: 'IAM Access Analyzer now simplifies inspecting unused access to guide you toward least privilege.', 'IAM Access Analyzer introduces custom policy checks powered by automated reasoning.', 'Announcing AWS IAM Identity Center APIs for visibility into workforce access to AWS.', and 'New organization-wide IAM condition keys to restrict AWS service-to-service requests.' On the right, there are panels for 'AWS Account' (Account ID: 093418366137, Account Alias: Create, Sign-in URL: https://093418366137.siginin.aws.amazon.com/console), 'Quick Links' (My security credentials, Tools), 'Tools' (Policy simulator), and 'Additional information' (Security best practices in IAM, IAM documentation, Videos, blog posts, and additional resources).

At this stage, to create an IAM user, click on the **Users** link and then click the **Add Users** button.

The screenshot shows the AWS IAM Users page. The left navigation menu is identical to the previous dashboard screenshot. The main content area is titled 'Users (0) Info' and describes what an IAM user is. It features a search bar and a table with columns: User name, Path, Group, Last activity, MFA, Password age, Console last sign-in, Access key ID, Active key age, and Access key last use. A note below the table states 'No resources to display'. At the top right of the table area, there is a 'Create user' button. On the right side of the page, there are 'Quick Links' (My security credentials, Tools), 'Tools' (Policy simulator), and 'Additional information' (Security best practices in IAM, IAM documentation, Videos, blog posts, and additional resources).

At this stage, choose a **username** for the IAM user, then select the option "**Provide user access to the AWS Management Console**" so the user can access the AWS Management Console, which is browser-based.

Next, select "**I want to create an IAM user**", then choose "**Autogenerated password**", and also check the box "**User must create a new password at next sign-in**".

Finally, click the **Next** button.

The screenshot shows the 'Specify user details' step of the IAM user creation wizard. On the left, a sidebar lists steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main area is titled 'Specify user details' and contains a 'User details' section. In the 'User name' field, 'itadmin' is entered. Below it, a note states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + - (hyphen)'. A checkbox labeled 'Provide user access to the AWS Management Console - optional' is checked. A tooltip for this checkbox says: 'If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center'. Under 'User type', the 'I want to create an IAM user' radio button is selected, with a note: 'We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.' In the 'Create password' section, the 'Autogenerated password' radio button is selected, with a note: 'You can view the password after you create the user.' Below it, a 'Custom password' input field is shown with placeholder text 'Enter a custom password for the user.' and a note: 'Must be at least 8 characters long. Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols (@ # \$ % ^ & * () _ + - (hyphen) = { }) !'. A 'Show password' checkbox is available. A tooltip for the password requirement says: 'Users must create a new password at next sign-in - Recommended. Users automatically get the IAMUserChangePassword policy to allow them to change their own password.' At the bottom, a note for programmatic access says: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more.' A 'Cancel' button and a prominent orange 'Next Step' button are at the bottom right.

By default, this user has no permissions. To assign permissions, you can select the **Attach policies directly** option and then grant **AdministratorAccess** to the user.

As you can see in this section, you can apply various types of **policies** to the IAM user. Once selected, click the **Next** button.

The screenshot shows the AWS IAM 'Create user' wizard at Step 2: Set permissions. The 'Attach policies directly' option is selected, and the 'AdministratorAccess' policy is checked. The table below lists other available policies:

Policy name	Type	Attached entities
AccessAnalyzerServiceRolePolicy	AWS managed	0
AdministratorAccess <input checked="" type="checkbox"/>	AWS managed - job function	0
AdministratorAccess-Amplify	AWS managed	0
AdministratorAccess-AWSLambdaBurst	AWS managed	0
AlexaForBusinessDeviceSetup	AWS managed	0
AlexaForBusinessFullAccess	AWS managed	0
AlexaForBusinessGatewayExecution	AWS managed	0
AlexaForBusinessLifecycleDelegatedAccessPolicy	AWS managed	0
AlexaForBusinessNetworkProfileServicePolicy	AWS managed	0
AlexaForBusinessPolicyDelegatedAccessPolicy	AWS managed	0
AlexaForBusinessReadOnlyAccess	AWS managed	0

At this stage, click on the **Create User** button to create the IAM user.

The screenshot shows the 'Review and create' step of the IAM user creation wizard. The left sidebar lists steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main area displays the user details and permissions summary.

User details:

User name:	itadmin	Console password type:	Autogenerated	Require password reset:	Yes
------------	---------	------------------------	---------------	-------------------------	-----

Permissions summary:

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy
IAMUserChangePassword	AWS managed	Permissions policy

Tags - optional:

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

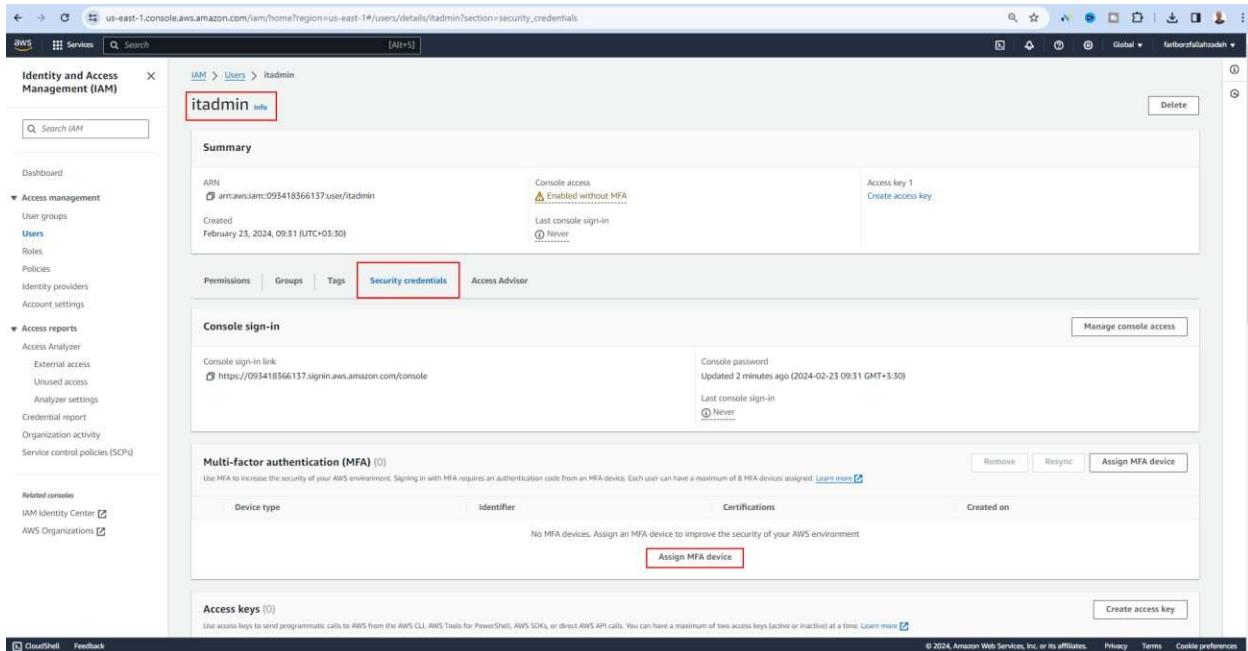
Buttons at the bottom: Cancel, Previous, **Create user**.

To view the IAM user, simply click on the **Users** link, and you will be able to see the IAM user you created.

To enhance the security of the IAM user, it is recommended to add MFA to the user. To do this, simply click on the **User** link.

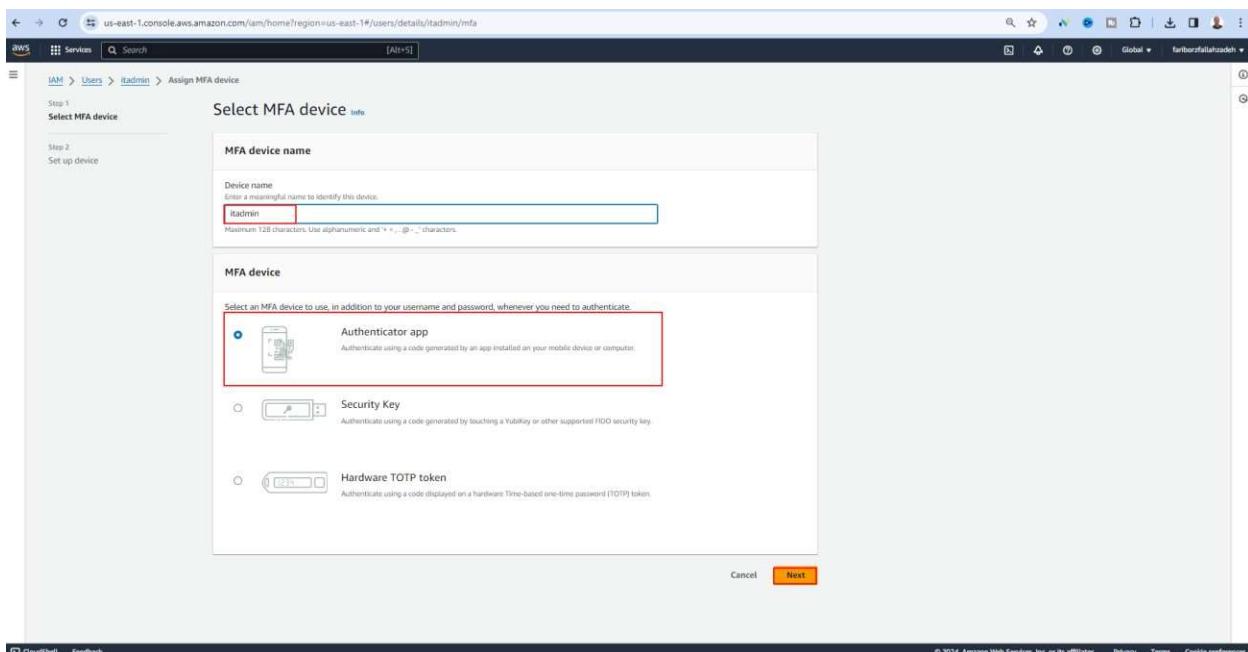
The screenshot shows the AWS IAM service in a web browser. The URL is `us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users`. The left sidebar has sections for Identity and Access Management (IAM), Access management, Access reports, and Related consoles. The main content area shows a success message: "User created successfully" with the note "You can view and download the user's password and email instructions for signing in to the AWS Management Console." Below this, the "Users (1) Info" section displays a table with one row for "itadmin". The table columns are: User name, Path, Group, Last activity, MFA, Password age, Console last sign-in, Access key ID, Active key age, and Access key last used. The "User name" column shows "itadmin" with a red box around it. The "Path" column shows "/". The "Group" column shows "0". The "Last activity" column shows a timestamp. The "MFA" column shows "None". The "Password age" column shows "Never". The "Console last sign-in" column shows "Never". The "Access key ID" column shows "None". The "Active key age" column shows "Never". The "Access key last used" column shows "Never". There are "View user" and "Create user" buttons at the top right of the table. At the bottom of the page, there are links for CloudShell, Feedback, and a footer with copyright information and links for Privacy, Terms, and Cookie preferences.

Next, click on the **Security Credentials** section and then click on the **Assign MFA Device** button.



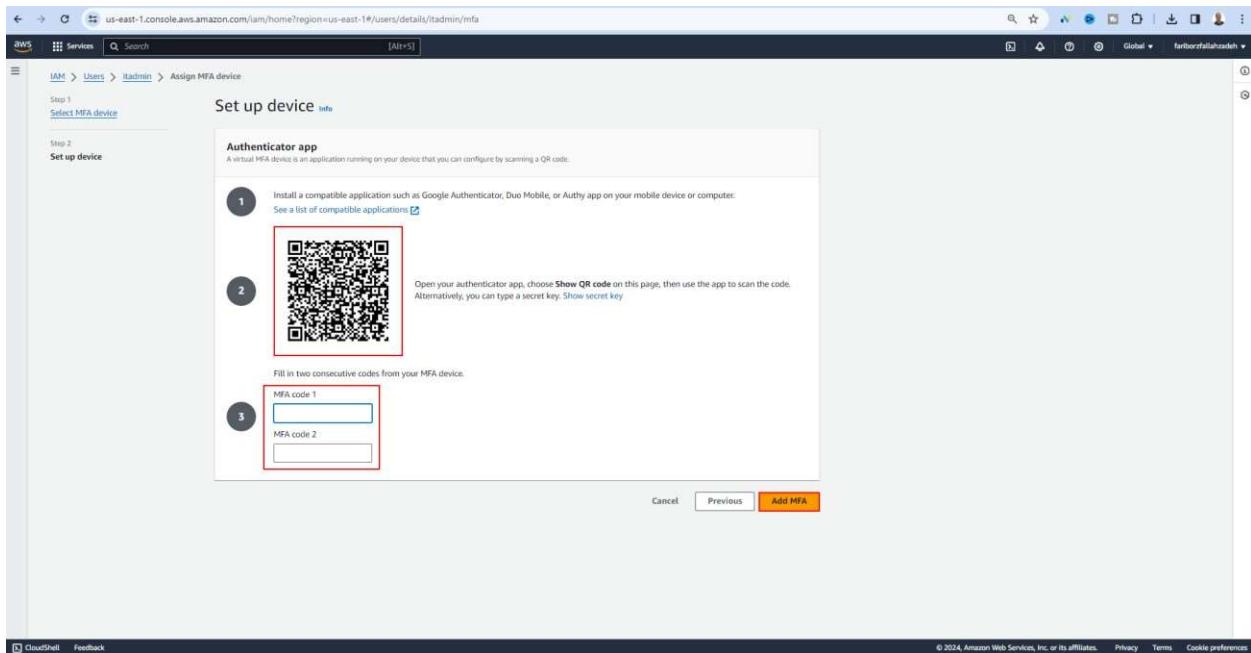
The screenshot shows the AWS IAM User Details page for a user named 'itadmin'. The 'Security credentials' tab is selected. Under the 'Multi-factor authentication (MFA)' section, there is a table with one row and a single button labeled 'Assign MFA device' which is highlighted with a red box.

At this stage, you need to specify a name for the **MFA Device**, select the **Authenticator App** as the MFA device type, and then click the **Next** button.



The screenshot shows the 'Select MFA device' step in the AWS IAM User Details process. In the 'MFA device' section, the 'Authenticator app' option is selected and highlighted with a red box. The 'Next' button at the bottom right is also highlighted with a red box.

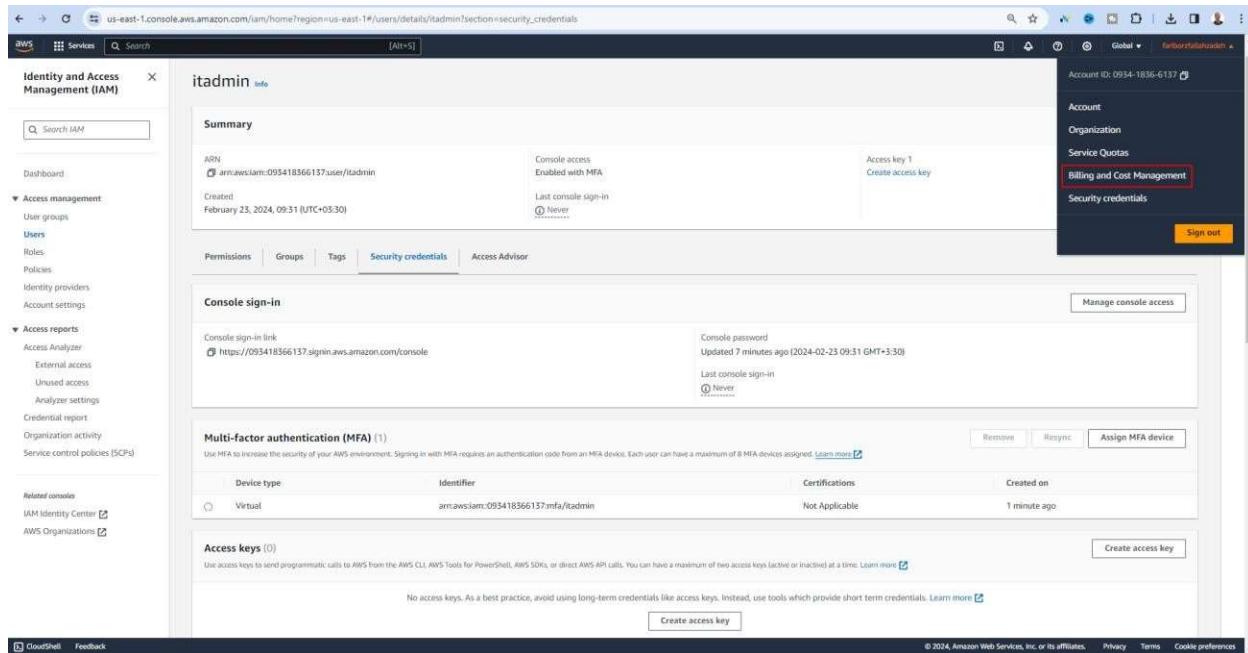
Then, go to your phone, open the **Google Authenticator** app, tap the + button, and scan the **QR code** provided by AWS. Enter the two **MFA codes** generated by Google Authenticator in the respective fields, and finally, click the **Add MFA** button.



How to Set Up CloudWatch in AWS

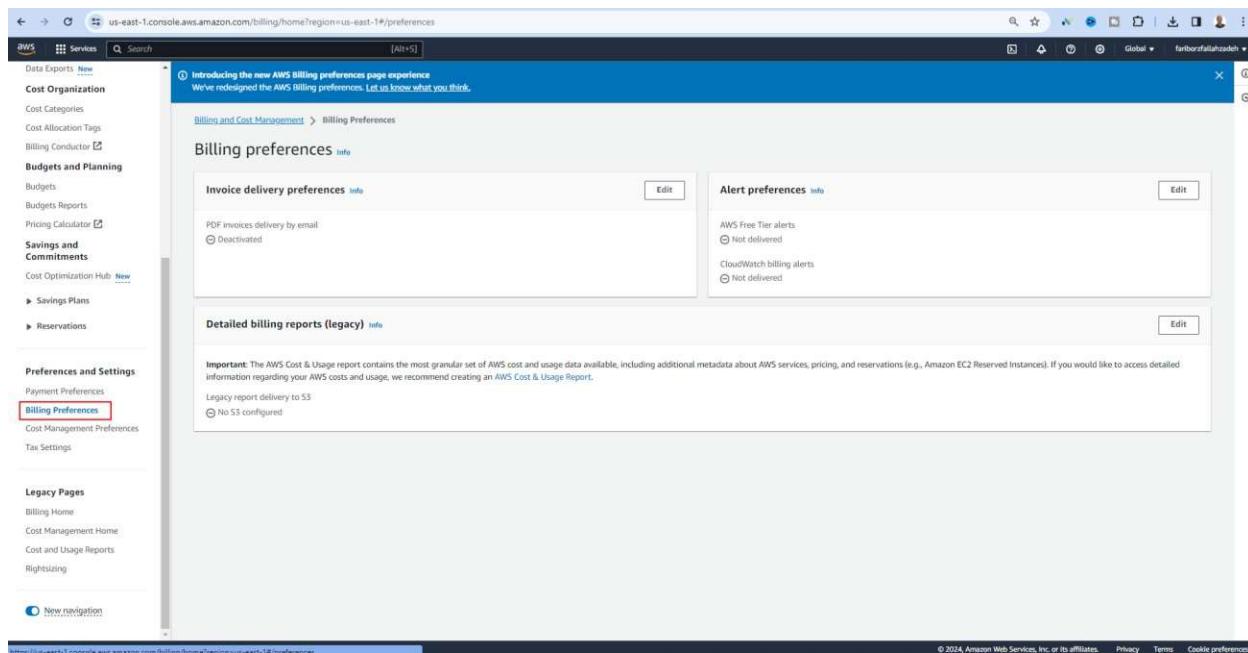
To monitor the usage of services in AWS and control costs, you need to configure a few options under the **Billing and Cost Management** section. This will allow you to set up a **Billing Alarm**, which will notify you when usage exceeds the specified threshold. Additionally, you should delete any created resources to avoid unnecessary charges.

To configure the billing settings, you need to go to the **Billing and Cost Management** section.



The screenshot shows the AWS IAM user details page for 'itadmin'. The 'Security credentials' section is highlighted with a red box, and the 'Billing and Cost Management' link is also highlighted with a red box. Other visible sections include 'Summary', 'Console sign-in', 'Multi-factor authentication (MFA)', and 'Access keys'.

At this stage, click on the **Billing Preferences** link.



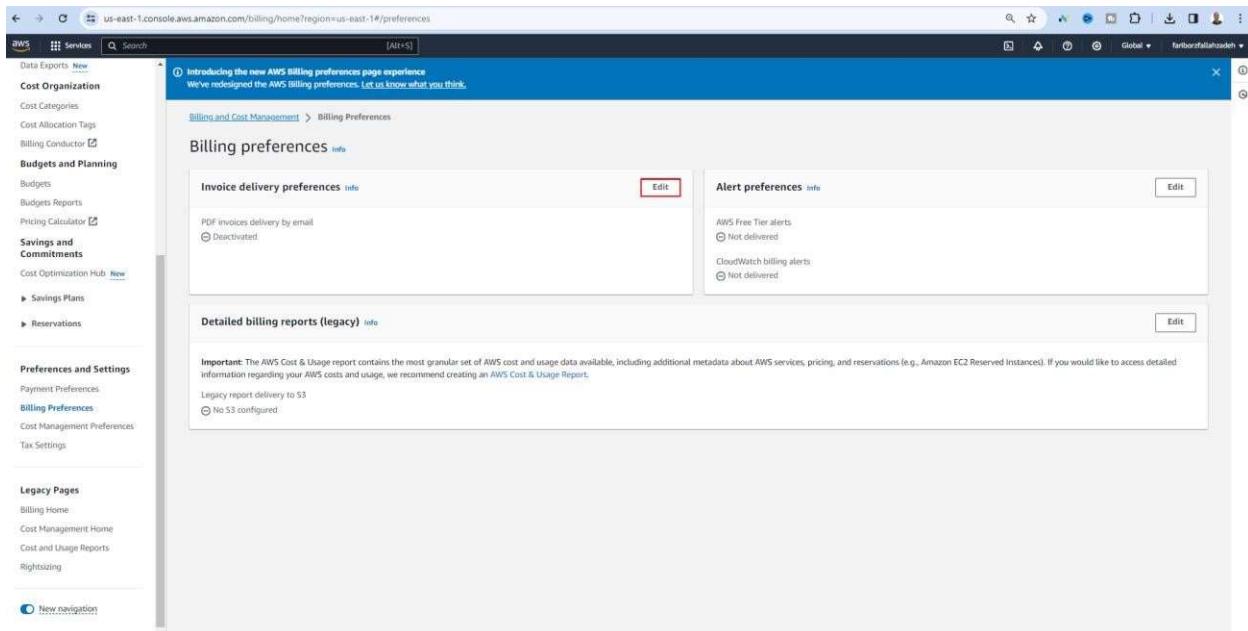
The screenshot shows the AWS Billing preferences page. The 'Billing Preferences' link in the left sidebar is highlighted with a red box. The main content area displays 'Invoice delivery preferences', 'Alert preferences', and 'Detailed billing reports (legacy)'.

When using a Free Tier account, it's important to know that there are limits on the usage of resources. This means that if you exceed the allowed limits for services, you will be charged.

One of the things you should do is check the **Bill** section on the AWS website every night before going to bed. Review your bill to see the charges applied for using services, and if necessary, delete any resources each night to avoid incurring additional costs.

In the **Billing Preferences** section, it is recommended to configure the following options:

First, under **Invoice Delivery Preferences**, select the **PDF Invoices delivered by email** option, then click the **Update** button.



First, under **Invoice Delivery Preferences**, select the **PDF Invoices delivered by email** option, then click the **Update** button.

By selecting this option, your invoices will be sent to you via email as a PDF file.

The screenshot shows the AWS Billing Preferences page. On the left, there's a sidebar with navigation links like Data Exports, Cost Organization, Cost Categories, Cost Allocation Tags, Billing Conductor, Budgets and Planning, Preferences and Settings, and Legacy Pages. The main content area has a header "Introducing the new AWS Billing preferences page experience" and "Billing and Cost Management > Billing Preferences".

The "Invoice delivery preferences" section contains a checkbox labeled "PDF invoices delivered by email" which is checked. Below it are "Update" and "Cancel" buttons. To the right, under "Alert preferences", there are sections for "AWS Free Tier alerts" (radio button selected) and "CloudWatch billing alerts" (radio button selected). At the bottom, there's a note about the AWS Cost & Usage report and a "Detailed billing reports (legacy)" section with a note about legacy report delivery to S3.

Next, configure the settings for **Alert Preferences**.

The screenshot shows the AWS Billing preferences page. On the left, there's a sidebar with navigation links like Data Exports, Cost Organization, Budgets and Planning, Preferences and Settings, and Legacy Pages. The main content area has a banner at the top stating "Introducing the new AWS Billing preferences page experience" and "Your invoice delivery preferences were updated successfully." Below this, the "Billing preferences" section is shown with two tabs: "Invoice delivery preferences" and "Alert preferences". Under "Invoice delivery preferences", it says "PDF invoices delivery by email" with a status of "Activated". Under "Alert preferences", there are two options: "AWS Free Tier alerts" (radio button selected) and "CloudWatch billing alerts" (radio button selected). A note below states: "Important: The AWS Cost & Usage report contains the most granular set of AWS cost and usage data available, including additional metadata about AWS services, pricing, and reservations (e.g., Amazon EC2 Reserved Instances). If you would like to access detailed information regarding your AWS costs and usage, we recommend creating an AWS Cost & Usage Report." At the bottom right of the "Alert preferences" tab, there's an "Edit" button.

At this stage, by selecting the following options, you will receive an email via AWS CloudWatch when 85% of the services in your Free Tier account are used.

This screenshot is similar to the previous one but shows the "Alert preferences" tab expanded. In the "Alert preferences" section, two checkboxes are highlighted with a red border: "Receive AWS Free Tier alerts" and "Receive CloudWatch billing alerts". The "Receive AWS Free Tier alerts" checkbox is checked, and the "Email address to receive Free Tier usage alerts - optional:" field contains the email "fariborz.fallahzadeh@gmail.com". The "Receive CloudWatch billing alerts" checkbox is also checked. At the bottom of the "Alert preferences" tab, there are "Update" and "Cancel" buttons.

At this stage, you need to configure the settings for **CloudWatch**. CloudWatch is a monitoring service, and we need this service to monitor your billing.

We need to be in a region called **North Virginia**. The pricing for services may vary across different regions, and this region is suitable for a Free Tier account.

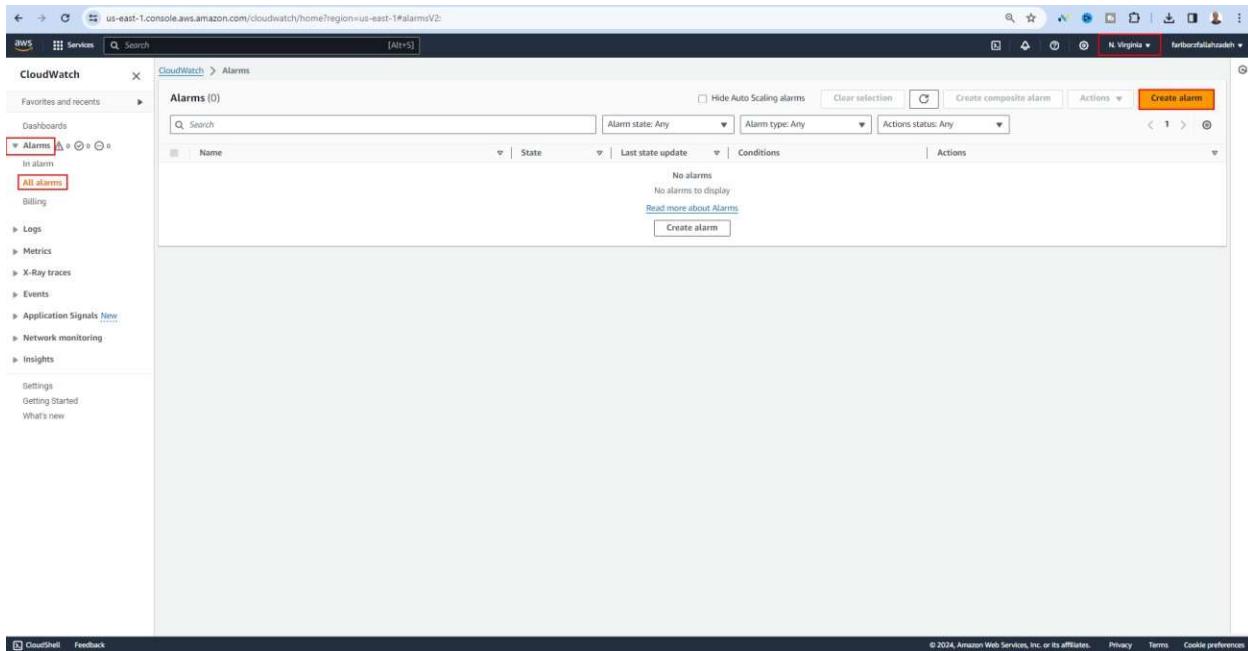
The screenshot shows the AWS Billing Home console at us-east-1.console.aws.amazon.com/billing/home?region=us-east-1#/preferences. A search bar at the top contains the query "cloudwatch". The search results are displayed under three main categories: Services, Features, and Resources.

- Services:** CloudWatch (Monitor Resources and Applications), Athena (Serverless interactive analytics service), Amazon EventBridge (Serverless service for building event-driven applications).
- Features:** CloudWatch dashboard (Systems Manager feature), Create a SFTP server (AWS Transfer Family feature), Data sources (Athena feature), Encryption Configuration (CloudWatch feature).
- Resources:** Introducing resource search (Enable to show cross-region resources for your account in search results. Takes less than 5 minutes to set up.)

A modal window titled "Alert preferences" is open on the right side of the screen. It lists two alert types:

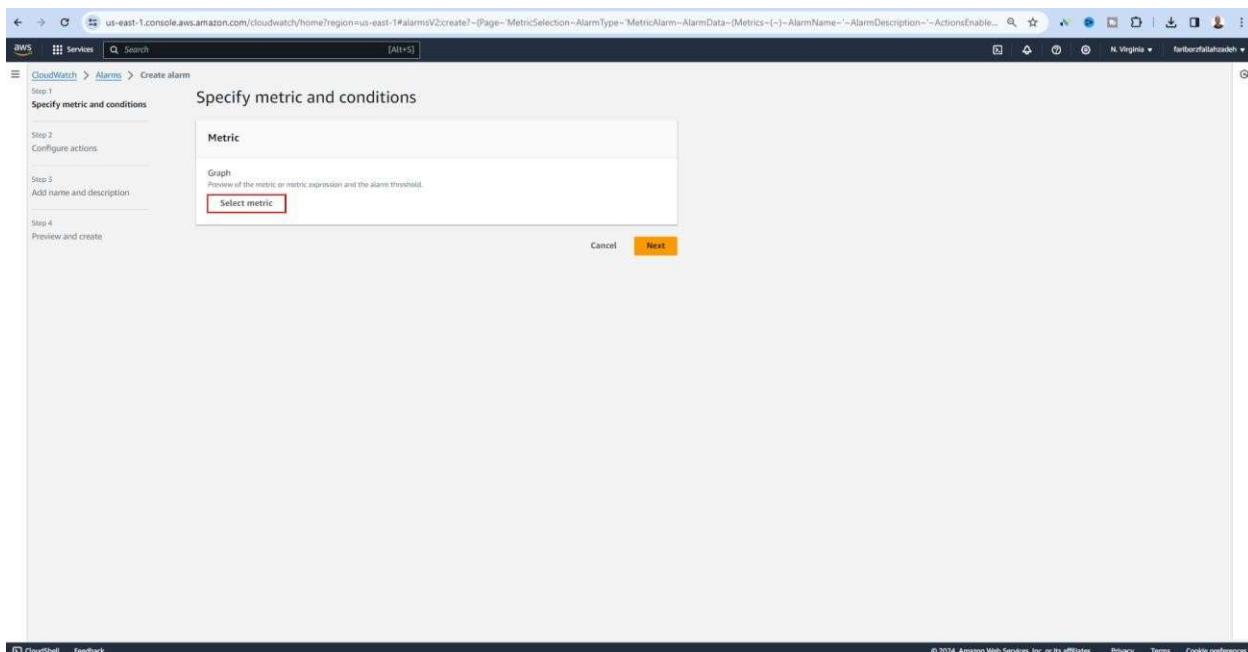
- AWS Free Tier alerts: Delivered to farborz.fallahzadeh@gmail.com
- CloudWatch billing alerts: Delivered

At this stage, click on the **All Alarms** link, then click the **Create Alarm** button.



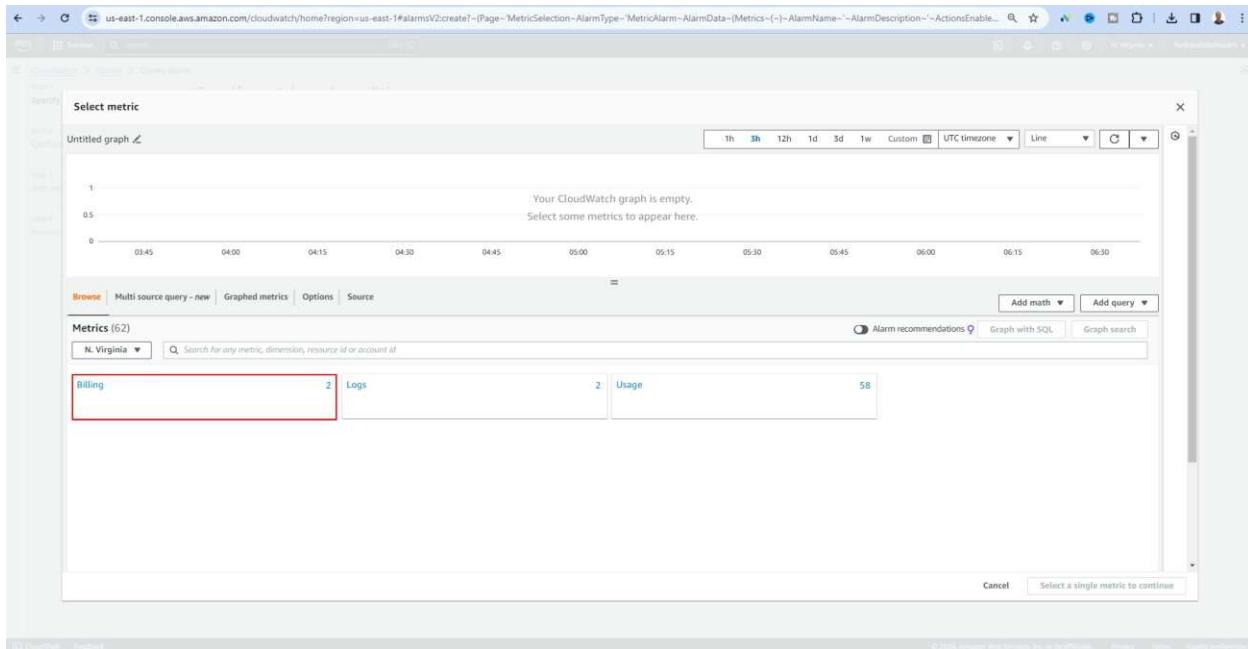
The screenshot shows the AWS CloudWatch Alarms page. On the left, there's a navigation sidebar with various links like Dashboards, Alarms (which is currently selected and highlighted in red), In alarm, All alarms (also highlighted in red), Billing, Logs, Metrics, X-Ray traces, Events, Application Signals, Network monitoring, and Insights. The main content area is titled 'Alarms (0)' and contains a search bar and filters for Alarm state, Alarm type, Actions status, and Last state update. A large orange 'Create alarm' button is located at the top right of this section. Below it, there's a message 'No alarms to display' and a 'Create alarm' button. At the bottom of the page, there are links for CloudShell, Feedback, and copyright information.

At this stage, click on the **Select Metric** button.

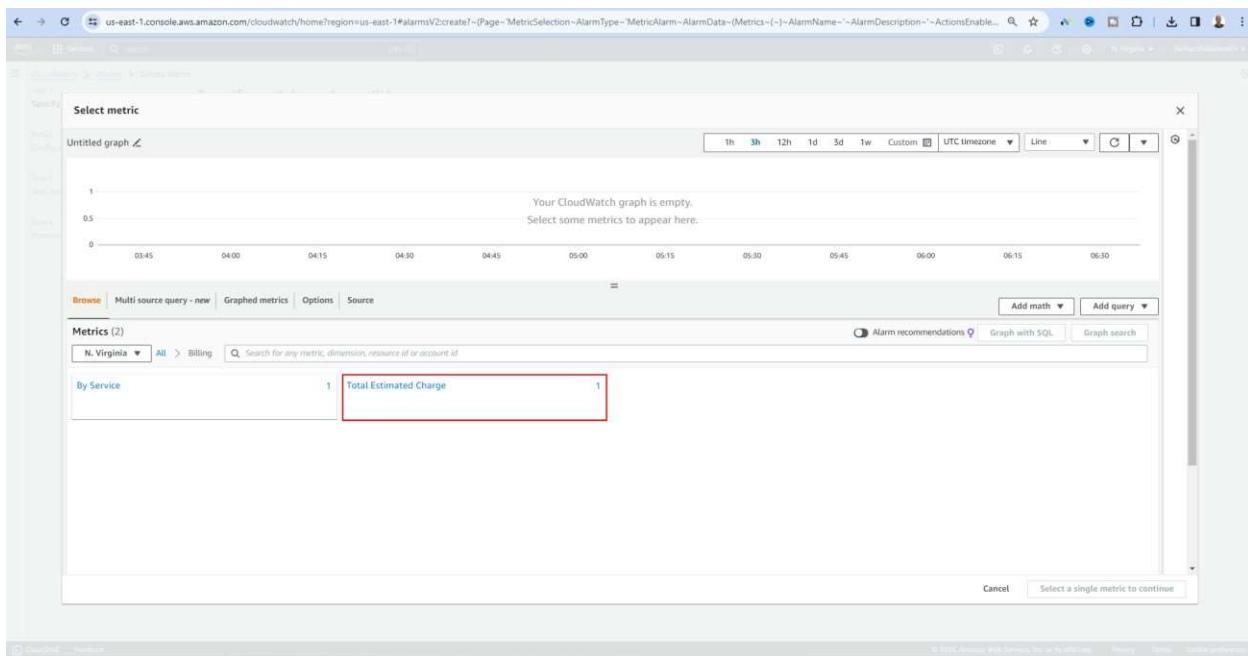


The screenshot shows the 'Specify metric and conditions' step of the 'Create alarm' wizard. The left sidebar lists steps: Step 1 (Specify metric and conditions, highlighted in red), Step 2 (Configure actions), Step 3 (Add name and description), and Step 4 (Preview and create). The main area has a 'Metric' section with a 'Graph' preview and a 'Select metric' button. At the bottom, there are 'Cancel' and 'Next' buttons. The footer includes CloudShell, Feedback, and copyright information.

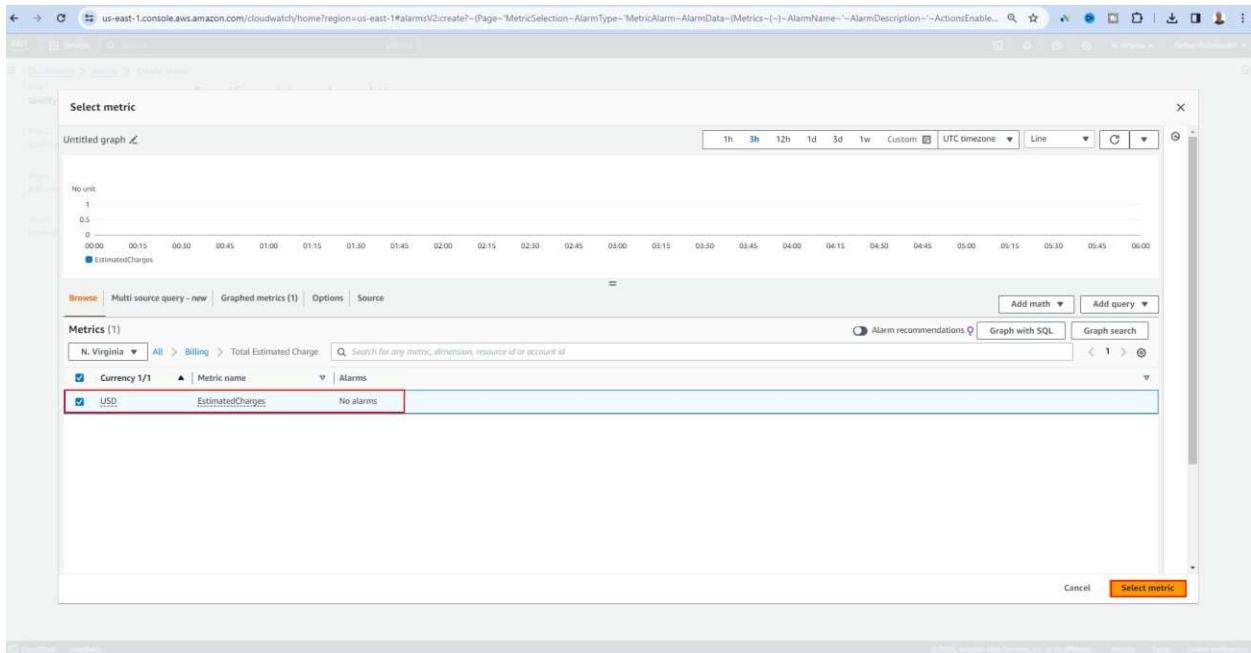
Click on the **Billing** link.



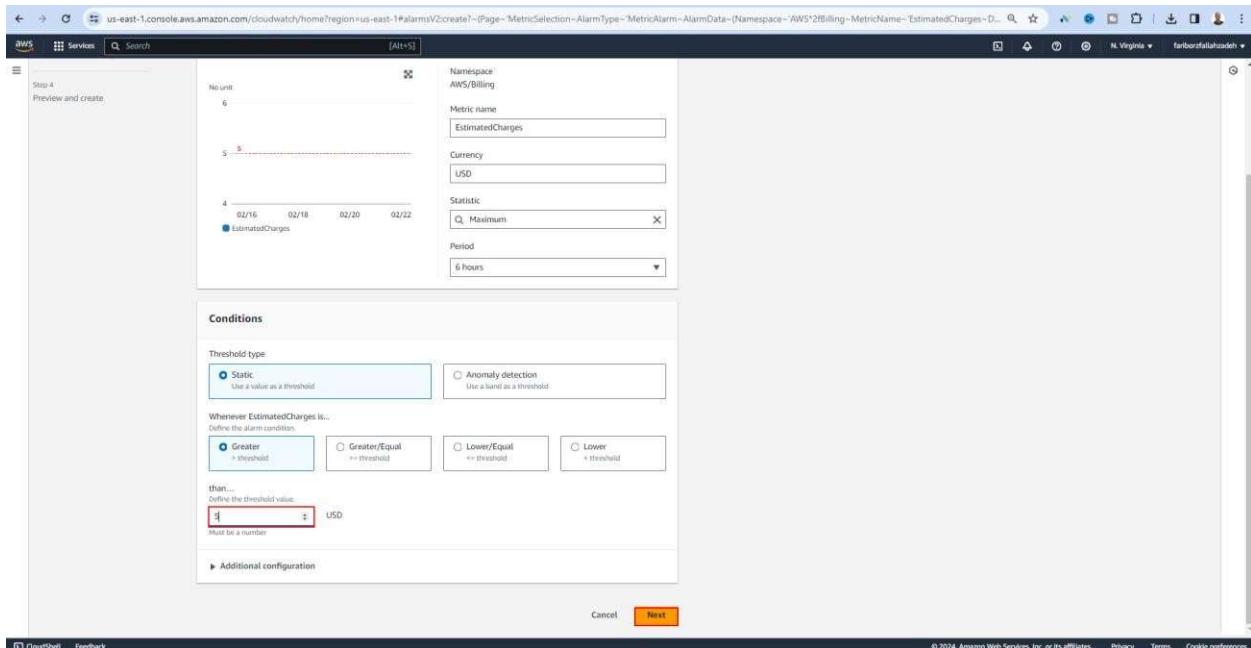
At this stage, click on the **Total Estimated Charge** link.



At this stage, select **USD Currency** and then click the **Select Metric** button.



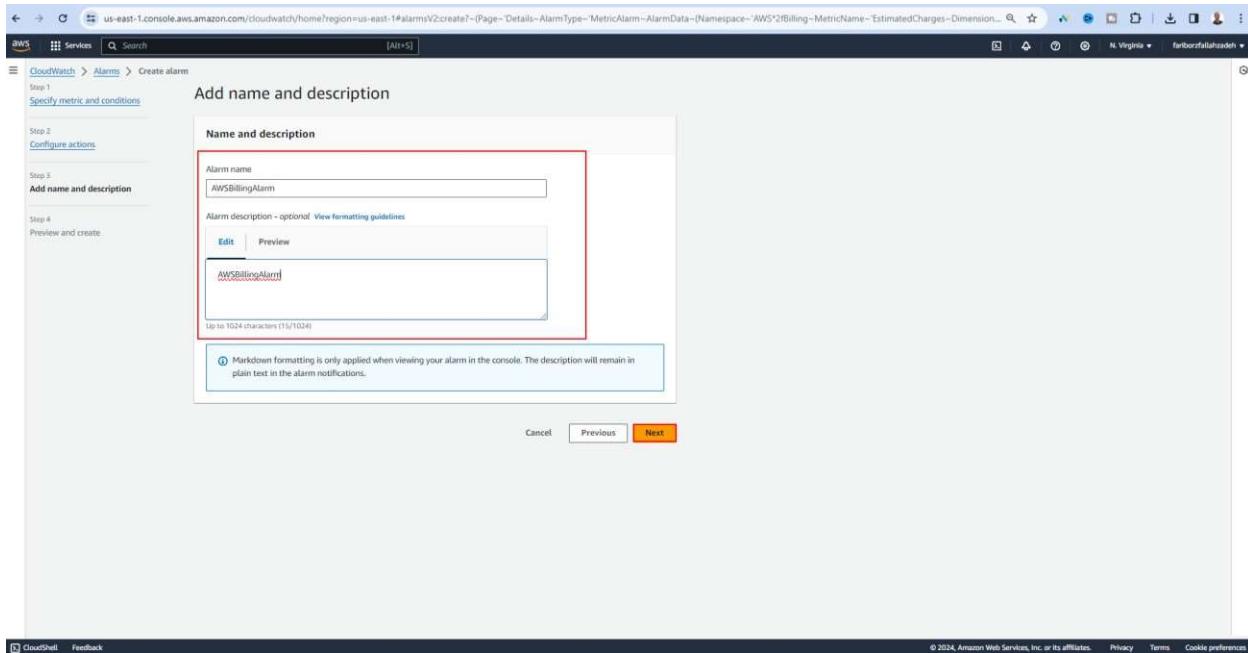
In the **Threshold value** section, set the value to **5 dollars** so that if the charges exceed 5 dollars, you will receive a notification via email. Then, click the **Next** button.



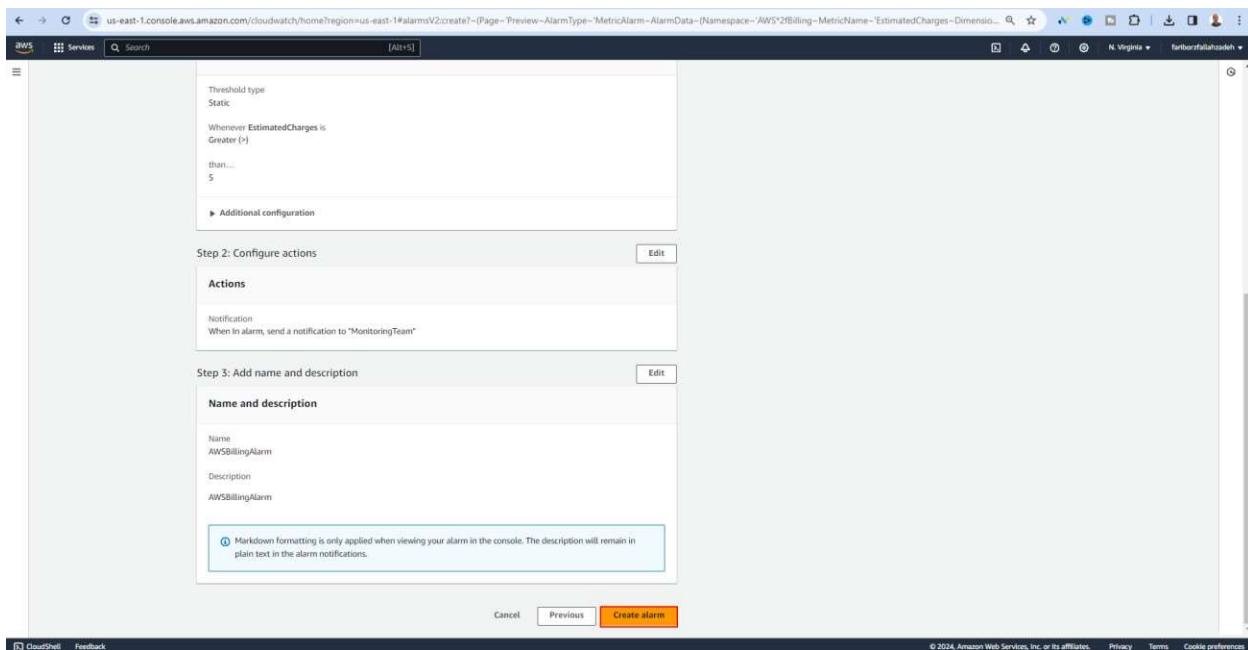
At this stage, you need to use a service called **SNS (Simple Notification Service)**. In this section, you will define an **SNS Topic**, which means you need to associate an email address with a topic. Define a name for the topic, then click the **Create Topic** button. Finally, click the **Next** button.

The screenshot shows the AWS CloudWatch Metrics Alarm creation process at Step 2: Configure actions. The 'Notification' section is active, displaying three trigger options: 'In alarm' (selected), 'OK', and 'Insufficient data'. Below these is a section for sending notifications to an SNS topic, with 'Create new topic' selected. A red box highlights the 'Create new topic...' input field containing 'MonitoringTeam'. Further down, an 'Email endpoints that will receive the notification...' section lists 'farborc.fallahzadeh@gmail.com' and 'user1@example.com, user2@example.com', also enclosed in a red box. At the bottom of the 'Notification' panel are 'Create topic' and 'Add notification' buttons. To the right, collapsed sections for 'Lambda action' and 'Auto Scaling action' are shown with their respective 'Add' buttons.

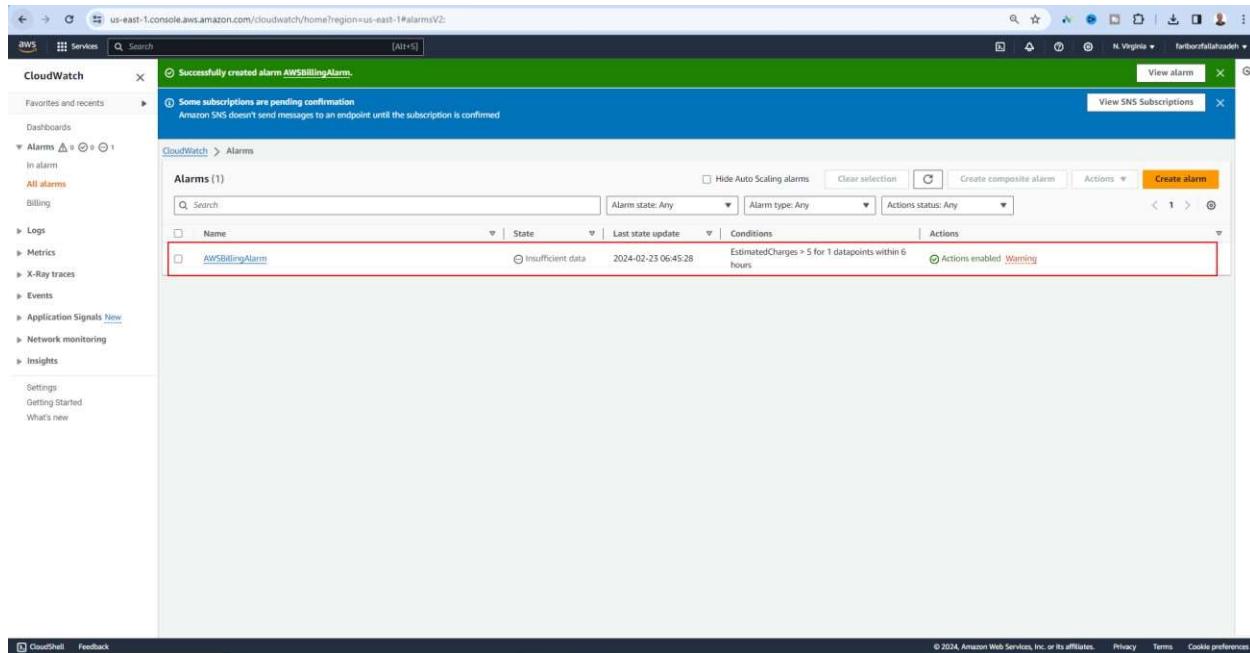
At this stage, you need to define an **Alarm Name** and an **Alarm Description**, then click the **Next** button.



At this stage, click the **Create Alarm** button.



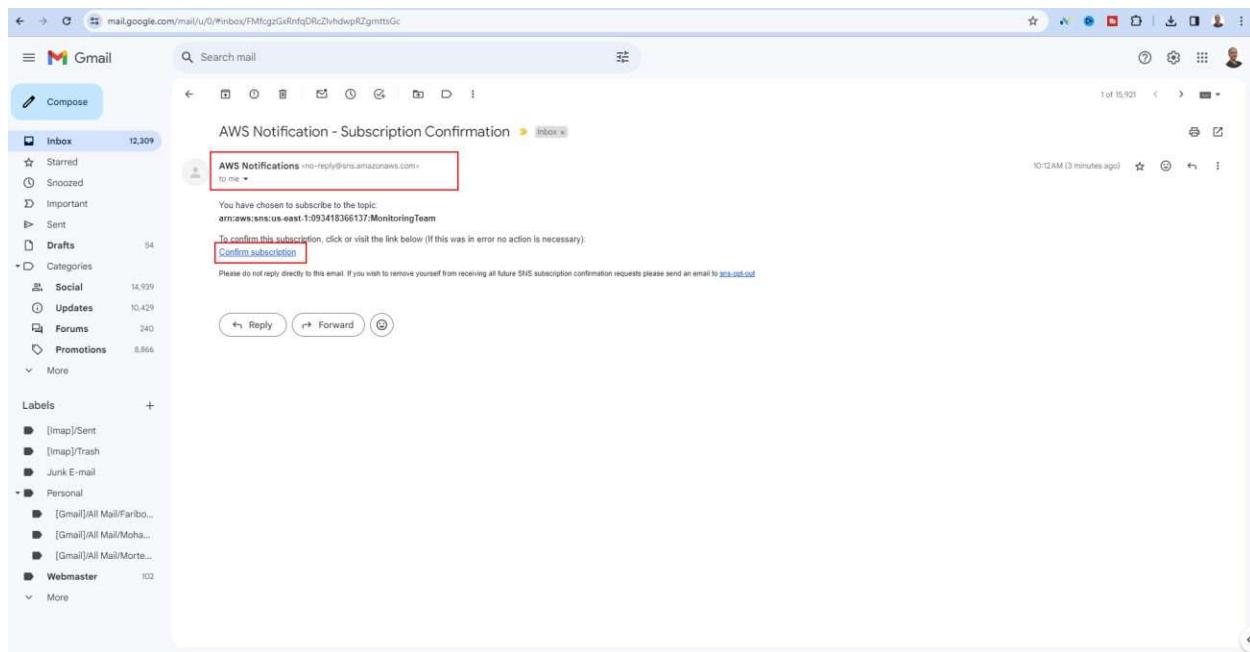
At this stage, we need to verify your **email address** to receive the alarm notifications.



The screenshot shows the AWS CloudWatch Alarms page. At the top, a green banner indicates "Successfully created alarm AWSBillingAlarm." Below this, a message states "Some subscriptions are pending confirmation" and "Amazon SNS doesn't send messages to an endpoint until the subscription is confirmed." The main table lists one alarm:

Name	State	Last state update	Conditions	Actions
AWSBillingAlarm	Insufficient data	2024-02-25 06:45:28	EstimatedCharges > 5 for 1 datapoints within 6 hours	Actions enabled Warning

Go to your email address and click on the **Confirm Subscription** link.



The screenshot shows a Gmail inbox with 12,309 unread emails. An incoming email from "AWS Notifications" is highlighted, titled "AWS Notification - Subscription Confirmation". The email body contains the following text:

You have chosen to subscribe to the topic
arn:aws:sns:us-east-1:09341368137:MonitoringTeam
To confirm this subscription, click or visit the link below (If this was in error no action is necessary)
[Confirm subscription](#)

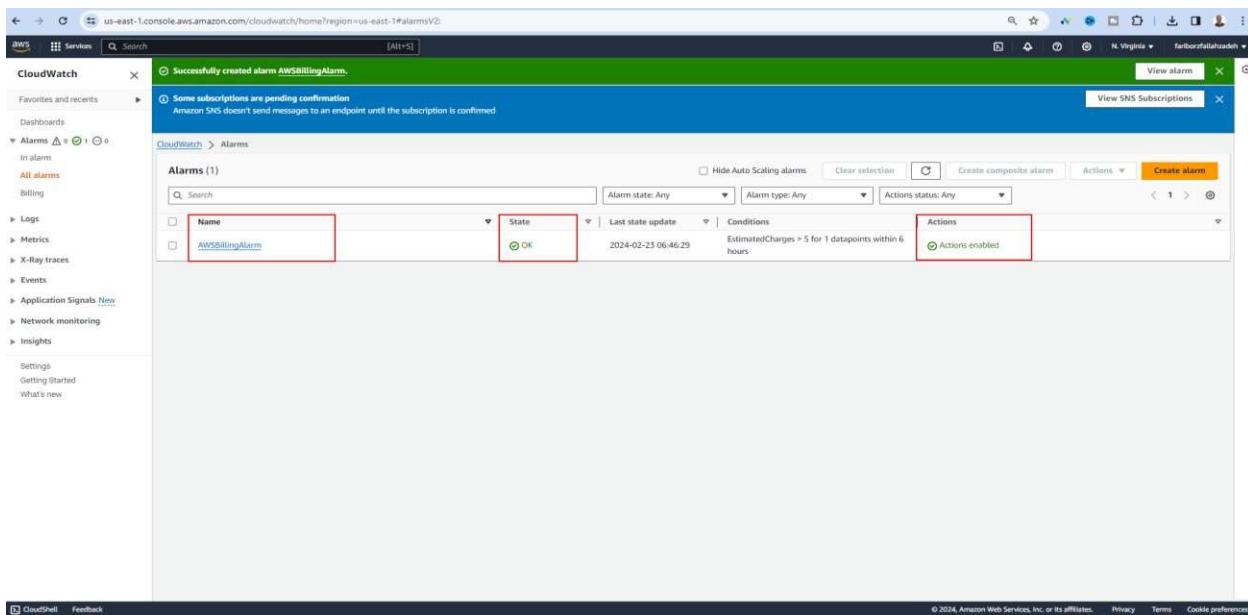
Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns:unsub](#).

As shown in the image below, the email address has been successfully verified.



Next, return to the **CloudWatch Alarm** page and refresh the page. The alarm status should show as **OK**, and its action should be marked as **Action Enabled**.

Currently, since the service usage is less than 5 dollars, the alarm status is shown as **OK**.



All the AWS setup steps have been completed, and now you can log in to the **AWS Management Console** using the **IAM User** through the link shown in the image. You can also set up an alias for this link.

The screenshot shows the AWS IAM Dashboard. On the left, there's a sidebar with navigation links like Identity and Access Management (IAM), Dashboard, Access management, IAM resources, and Tools. The main content area has sections for Security recommendations, IAM resources (with counts of 0 User groups, 1 User, 2 Roles, 0 Policies, and 0 Identity providers), and What's new (listing recent changes like IAM Access Analyzer simplifications and custom policy checks). To the right, there are panels for AWS Account (Account ID: 093418366137, Account Alias: Create, Sign-in URL: https://093418366137.signin.aws.amazon.com/console), Quick Links (My security credentials, Policy simulator, Additional information), and Tools (Security best practices in IAM, IAM documentation, Videos, blog posts, and additional resources). The bottom right corner includes copyright information: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

At this stage, define an Alias.

The screenshot shows the AWS IAM Dashboard. A modal window titled "Create alias for AWS account 093418366137" is open. In the "Preferred alias" field, the value "dsinfo" is entered. Below the field, a note states: "Must be no more than 63 characters. Valid characters are a-z, 0-9, and - (hyphen)." Under "New sign-in URL:", the URL "https://dsinfo.sigin.aws.amazon.com/console" is displayed. A note below it says: "IAM users will still be able to use the default URL containing the AWS account ID." At the bottom right of the modal are "Cancel" and "Create alias" buttons. The background of the dashboard shows various IAM resources like User groups, Users, and Roles, along with a "What's new" section and a "Security recommendations" section.

As shown in the image below, the alias for the link has been set up.

The screenshot shows the AWS IAM Dashboard after the alias creation. A green banner at the top center reads "Alias dsinfo created for this account." The main interface remains largely the same, with the "AWS Account" sidebar showing the account ID "093418366137" and the "Sign-in URL" "https://dsinfo.sigin.aws.amazon.com/console". The "Quick Links" and "Tools" sections are also visible on the right side of the dashboard.

At this stage, click on the **Users** section, and then click on **IAM User**.

The screenshot shows the AWS Identity and Access Management (IAM) service interface. On the left, there's a navigation sidebar with options like Dashboard, User groups, Roles, Policies, Identity providers, Account settings, Access reports, and Related consoles. Under 'Access management', the 'Users' option is selected and highlighted with a red box. The main content area is titled 'Users (1) info' and contains a table with one row for 'itadmin'. The table columns include User name, Path, Group, Last activity, MFA, Password age, Console last sign-in, Access key ID, Active key age, and Access key last used. The 'itadmin' row has a red box around it. At the top right of the main content area, there are 'Delete' and 'Create user' buttons.

In the **Security Credentials** section, click on the **Manage Console Access** button.

This screenshot shows the detailed view for the 'itadmin' user. The left sidebar remains the same as the previous screenshot. The main content area is titled 'itadmin info'. It features a 'Summary' section with details like ARN, Created date, and Last console sign-in. Below this is a 'Console sign-in' section with a 'Manage console access' button. The most prominent part is the 'Security credentials' tab, which is highlighted with a red box. This tab contains sections for Multi-factor authentication (MFA), Access keys, and a note about best practices for access keys. At the bottom of the page, there are links for CloudShell, Feedback, and standard footer links for Privacy, Terms, and Cookie preferences.

At this stage, select the following options and then click the **Apply** button.

The screenshot shows the AWS IAM User Details page for the user 'itadmin'. A modal window titled 'Manage console access' is open over the main content. The modal contains the following settings:

- Console access:** The 'Enable' radio button is selected.
- Set password:** The 'Autogenerated password' radio button is selected.
- Checkboxes:** The 'User must create new password at next sign-in' checkbox is checked, and the 'Users can automatically get the UserAccessChangePassword policy to allow them to change their own password' checkbox is also checked.

At the bottom right of the modal, there are 'Cancel' and 'Apply' buttons. The 'Apply' button is highlighted with a red border.

At this stage, download the **CSV file**. This file contains the **URL**, **IAM User**, and **IAM Password**.

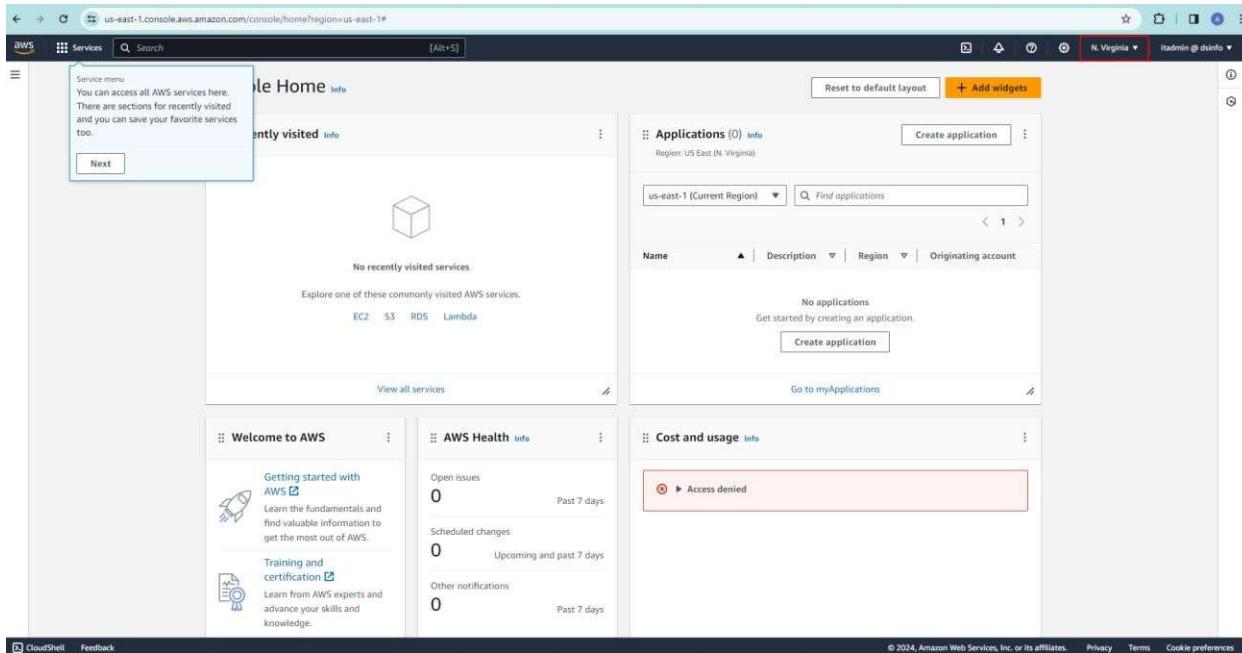
The screenshot shows the AWS Identity and Access Management (IAM) console. A modal window titled 'Console password' is open, displaying a success message: 'You have successfully updated the user's new password.' It also shows the URL for the sign-in page: 'https://signin.aws.amazon.com/console'. Below this, it lists the 'User name' as 'itadmin' and the 'Console password' as '*****'. There are 'Show' and 'Close' buttons at the bottom of the modal. The background shows the IAM user details for 'itadmin', including a summary tab, access keys, and security credentials.

Then, go to the **AWS Management Console** link and sign in using the **IAM User** and **IAM Password** from the CSV file.

The screenshot shows the AWS Management Console sign-in page. The URL is 'eu-north-1.signin.aws.amazon.com'. The form is titled 'Sign in as IAM user' and requires 'Account ID (12 digits) or account alias', 'IAM user name', 'Password', and a 'Remember this account' checkbox. Below the form are links for 'Sign in using root user email' and 'Forgot password?'. To the right of the form is an advertisement for 'Amazon Lightsail' featuring a cartoon character and the text 'Lightsail is the easiest way to get started on AWS'. At the bottom of the page are language and legal links.

Since we have enabled MFA for the IAM user, you will need to enter the code generated by the **Google Authenticator** app in this section, and then click the **Submit** button.

As you can see, we have successfully logged in to the **AWS Management Console**.



Introduction to AWS Regions and Availability Zones

AWS is spread across the world and is present in most countries, continuously expanding. These countries are referred to as **Regions**.

Within each Region, there are several **Availability Zones**. Each Zone is like a **Data Center**, and an Availability Zone consists of multiple **Data Centers**.

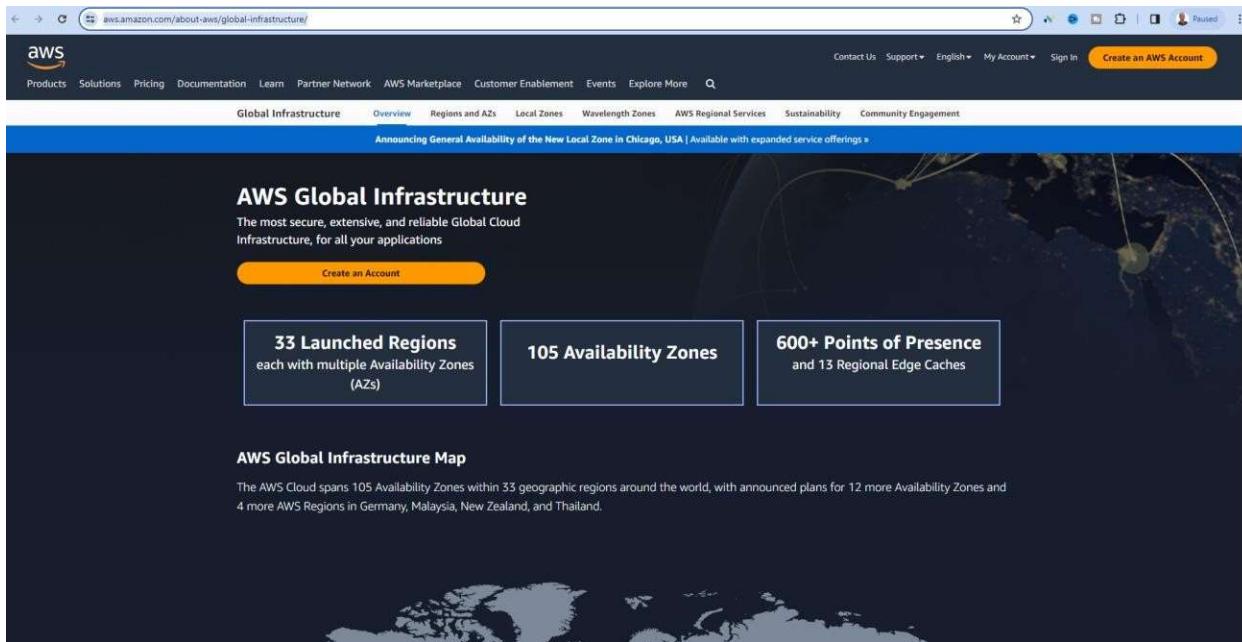
Each Region must have at least two **Availability Zones**.

AWS claims to have high and extensive security, and it is a reliable cloud platform. It offers more than **175 services** through its data centers, and these services are continuously increasing.

Currently, AWS has more than **105 Availability Zones** across **33 Geographic Regions** worldwide.

To understand the AWS Global Infrastructure, you can view the following link:

<https://aws.amazon.com/about-aws/global-infrastructure/>



By using these Availability Zones, you can achieve **High Availability** for your infrastructure. For example, if you have four web servers, you can place two web servers in one Availability Zone and the other two in another Availability Zone.

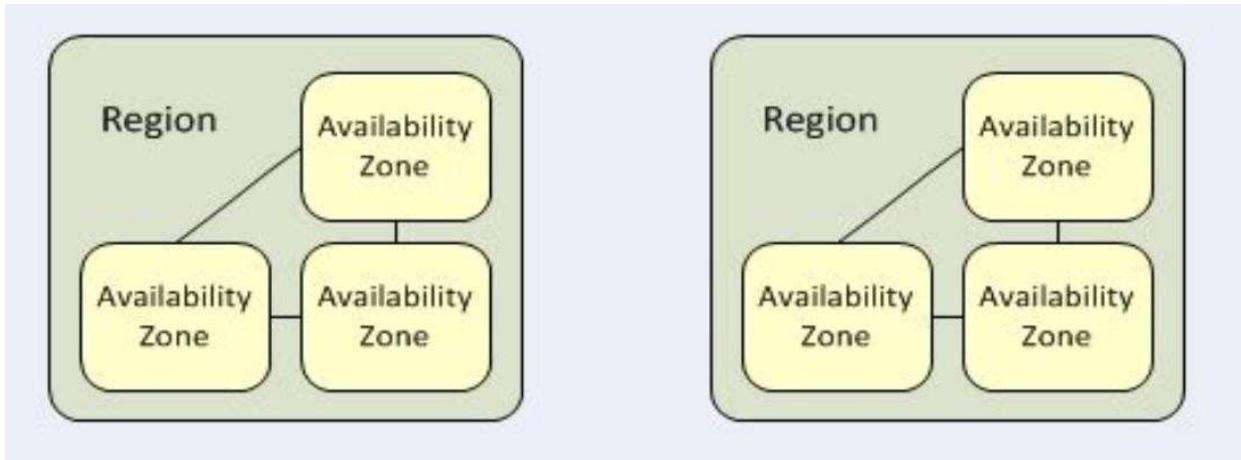
Therefore, if one of your Availability Zones goes down or becomes slow, your web server in the other Availability Zone will remain **up** and operational.

Benefits of Using AWS Infrastructure

- **High Availability** through multiple Availability Zones.
- **Improved Continuity** through replication across Regions.
- **Data Compliance** and required **Residency**.
- **Geographic Expansion** to leverage global infrastructure.

Introduction to Availability Zones

An **Availability Zone** means that you can have multiple zones, and you can distribute your infrastructure across several Availability Zones within your region to achieve **High Availability**.



For example, if you launch a **Virtual Machine**, which is known as an **Instance**, you need a place for it to reside, and that place is the **Availability Zone**. Alternatively, you can allow AWS to decide which Availability Zone your instance will be placed in.

When designing an application, for your **infrastructure layout**, you can choose multiple Availability Zones within a region, and your infrastructure and traffic will be distributed across these zones.

When you log in to the **AWS Management Console** using an **IAM User**, you can view all the services provided by AWS.

For example, if you are a **DevOps** specialist or a **SysAdmin**, you may need a virtual machine. You can use the **EC2** service from AWS to fulfill this need.

In this course, our focus will be on services like **EC2**, **Beanstalk**, and storage services such as **S3** or **EFS**, along with database services like **RDS** and **Elasticache**, which are relevant to **DevOps**, **SysAdmin**, and **Developer** roles.

In the **AWS Management Console**, you can select the **Region** for your project.

When you select a region, you can use the Availability Zones within that region, so your data will be stored across those zones.

When using AWS Free Tier, it's better to choose a **US-based Region** because other regions may have higher costs. Although the Free Tier account is free, it has limitations, and if you exceed those limits, you will incur charges.

Introduction to AWS EC2

EC2 (Elastic Compute Cloud) is one of the most popular AWS services. It provides you with virtual machines and related services.

Introduction to EC2 Features

-Web Service API

EC2 provides you with a **Web Service API** to manage, provision, and deprovision virtual machines within the cloud.

-Ease Scale Up/Down

You can easily perform **Scale Up** or **Scale Down** operations on your resources in EC2.

For example, if you have 8GB of RAM, you can easily upgrade it to 16GB (Scale Up), or you can reduce it to 4GB (Scale Down).

You can scale up or scale down all the resources of a virtual machine, such as **CPU**, **network**, **storage**, and more.

The ability to scale up or scale down resources is due to the **elasticity** of the EC2 service.

-Pay Only For What you User

You pay for what you use, and how much you consume is calculated using different methods.

-Can Be Integrated into Several Other Services

The EC2 service easily integrates with various other services like **S3**, **EFS**, and more.

Introduction to EC2 Pricing

The cost of using EC2 is categorized into the following four methods:

-On Demand

It is calculated on a per-second or per-hour basis, and you don't need to pay for the entire duration — it depends on the amount of time you use the service.

-Reserved

You can reserve capacity for **1 to 3 years** and take advantage of discounts on the service.

-Spot

Amazon EC2 **Spot Instances** are a type of compute instance in Amazon EC2 cloud services that allow users to access computing resources (such as CPU and memory) at a lower cost compared to reserved or default EC2 instances. Essentially, Spot Instances enable users to take advantage of **unused capacity** in Amazon's data centers, often referred to as "empty space," in a cost-effective manner.

These EC2 instances are offered through an auction system, meaning that users specify their **bid price** for each compute resource. If the current price set by Amazon matches the user's bid, the Spot instance becomes available. However, if the user's bid price exceeds the current price, the Spot instance will be terminated and then reactivated when the price matches the user's bid or when other compute resources become available.

-Dedicated Host

In this case, you have a **Dedicated Host**, but it comes at a higher cost.

In this course, we will be using **On Demand** and **Free Tier** accounts.

Introduction to EC2 Components

An EC2 service consists of the following components:

-AMI

When you want to launch an EC2 instance, you need an **AMI** (Amazon Machine Image). AWS provides a large number of AMIs for creating EC2 instances.

-Instance Type

When you launch an EC2 instance, the **Instance Type** specifies the hardware configuration that the EC2 instance will use.

For example, it defines the amount of **CPU**, **RAM**, and **network** resources required for the EC2 instance.

-Amazon Elastic Block Store

EBS, which stands for **Elastic Block Storage**, is a type of storage in AWS. Each AMI comes with a storage volume — for example, it could be **8GB** for a Linux machine or **30GB** for a Windows operating system.

Therefore, there are different types of **AMIs** that can use **EBS storage**.

You can specify the **storage size** or attach your own **storage** to the EC2 instance.

The **EBS service** is a **virtual hard disk** that can be used to store an operating system or data.

-TAG

For each resource in AWS, you have a **TAG**, which is used in a **Key:Value** format. A tag can be something like the **EC2 Name**, **Project**, or the **customer** who will use your service. You can define and use multiple tags as needed.

Defining **tags** can be very useful for **filtering** and **billing** purposes.

-Security Group

A **Security Group** acts as a **virtual firewall** used to control traffic for one or more instances.

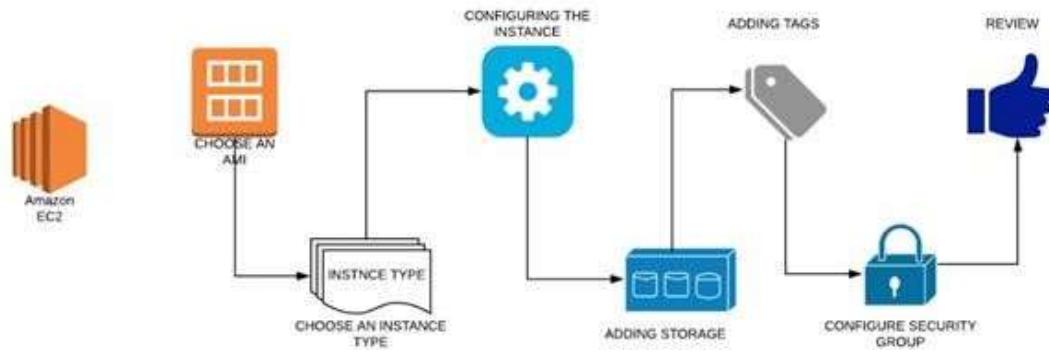
-EC2 Public Key

An **EC2 instance** uses **public key cryptography** to encrypt or decrypt login information.

Introduction to EC2 Instance Creation Steps

Creating an EC2 instance is very simple and **wizard-based**. You can select the **AMI**, choose the **Instance Type**, and configure other settings as needed.

In the image below, you can see the **steps for creating an EC2 instance**.



How to Create an EC2 Instance

In this section, we will learn **how to create an EC2 instance**.

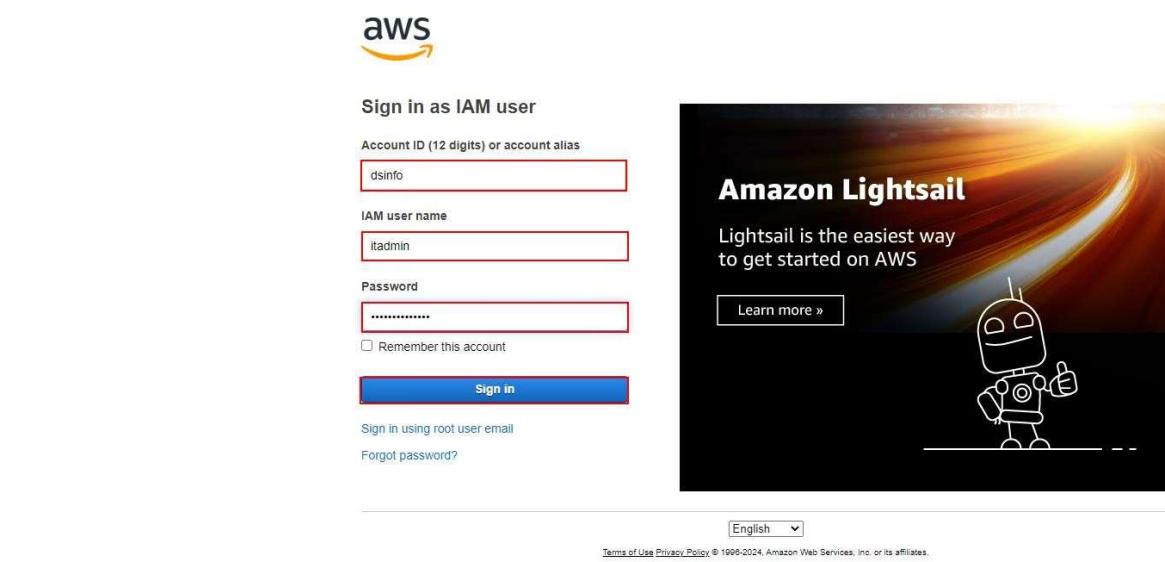
The **EC2 Instance** service refers to a **virtual machine**.

To create an EC2 instance, you need to follow these steps:

Step 1:

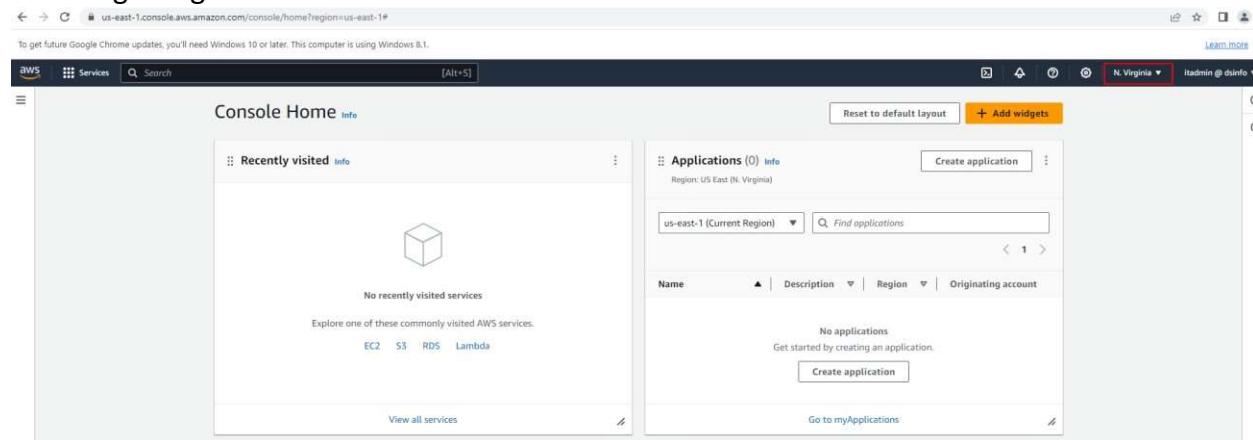
At this stage, you need to log in to the **AWS Management Console** using your **IAM User** and select your desired **Region**.

Logging in to AWS Management Console with IAM User



The image shows the AWS sign-in interface. On the left, there's a form titled "Sign in as IAM user" with fields for "Account ID (12 digits) or account alias" (containing "dsinfo"), "IAM user name" (containing "itadmin"), and "Password" (containing "*****"). There's also a "Remember this account" checkbox and a "Sign in" button. Below the form are links for "Sign in using root user email" and "Forgot password?". On the right, there's a promotional banner for "Amazon Lightsail" featuring a cartoon robot and the text "Lightsail is the easiest way to get started on AWS". At the bottom right of the main area, there's a language selection dropdown set to "English" and a link to "Terms of Use Privacy Policy".

Selecting a Region



The image shows the AWS Console Home page. The top navigation bar indicates the region is set to "N. Virginia" and the user is "itadmin @ dsinfo". The main content area is titled "Console Home" and includes sections for "Recently visited" services (with a placeholder image of a cube) and "Applications" (which currently has 0 applications). There are buttons for "Create application" and "Go to myApplications". The overall layout is clean and organized, typical of the AWS management console.

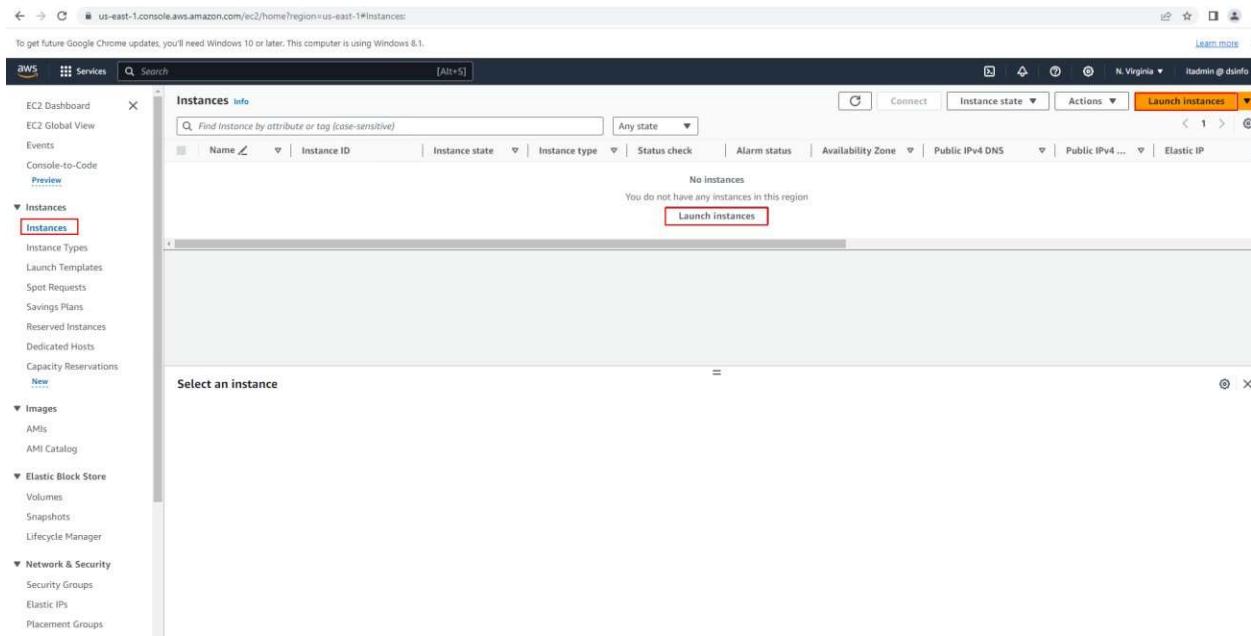
Step 2:

Selecting the EC2 Instance Service

The screenshot shows the AWS Management Console search results for the term 'ec2'. The search bar at the top contains 'ec2'. The left sidebar has a 'Services' section with various categories like Services (13), Features (57), and Resources (New). The main search results are categorized under 'Services' and 'Features'. Under 'Services', the first result is 'EC2' with the description 'Virtual Servers in the Cloud'. This item is highlighted with a red border. Other services listed include EC2 Image Builder, Recycle Bin, and Amazon Inspector. Under 'Features', the results include Dashboard, EC2 Instances, AMIs, and Elastic IPs. To the right of the search results, there is a separate window titled 'Create application' which shows a list of applications with one entry: 'Info (0) Info (Virginia)'. There is a 'Create application' button and a link to 'Go to myApplications'.

Step 3:

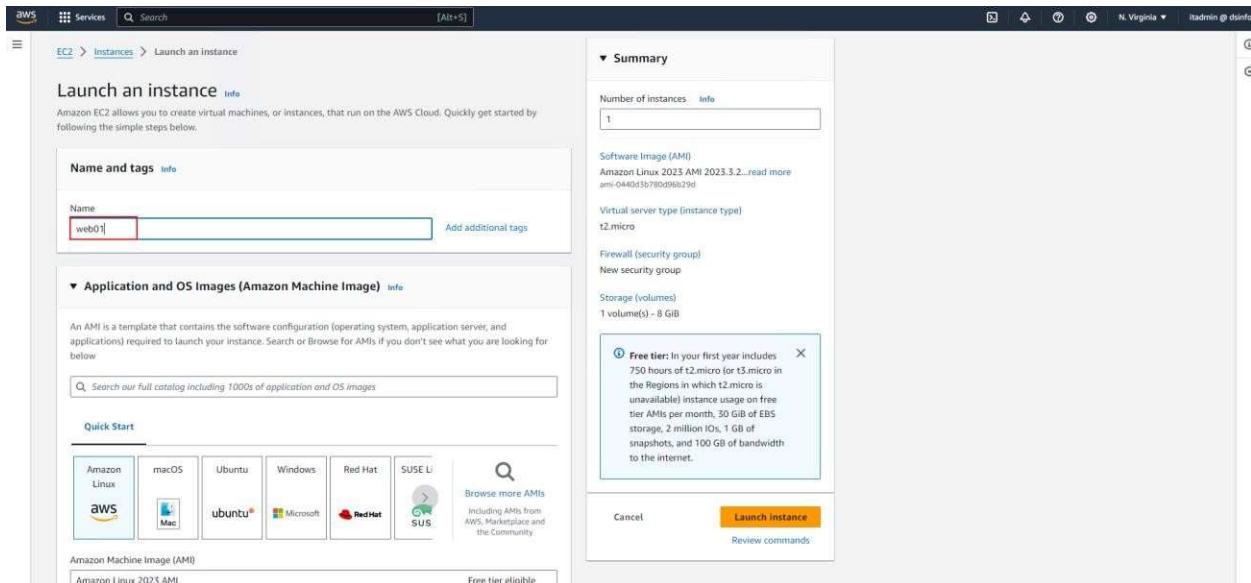
Click on the **Instances** link, then click the **Launch Instance** button.



The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various EC2-related options like Dashboard, Global View, Events, and Capacity Reservations. The main area is titled 'Instances' and shows a message: 'No instances' and 'You do not have any instances in this region'. Below this is a red-bordered 'Launch instances' button. The URL in the browser is us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#instances.

Step 4:

Specify a name for the instance.

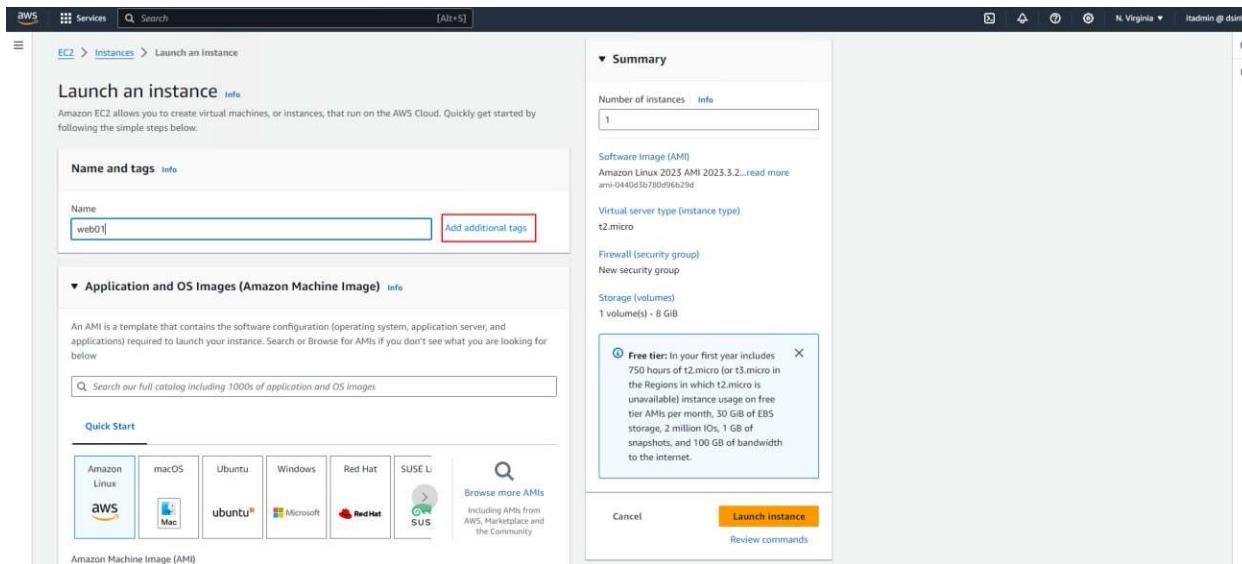


The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' step, the 'Name' field contains 'web01'. In the 'Application and OS Images (Amazon Machine Image)' step, the search bar shows 'Search our full catalog including 1000s of application and OS images'. Below it, under 'Quick Start', there are icons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. A note says 'Free tier eligible'. On the right, the 'Summary' section shows 'Number of instances: 1'. It also lists the 'Software Image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)'. A callout box provides information about the free tier. At the bottom are 'Cancel', 'Launch Instance' (in a yellow button), and 'Review commands'.

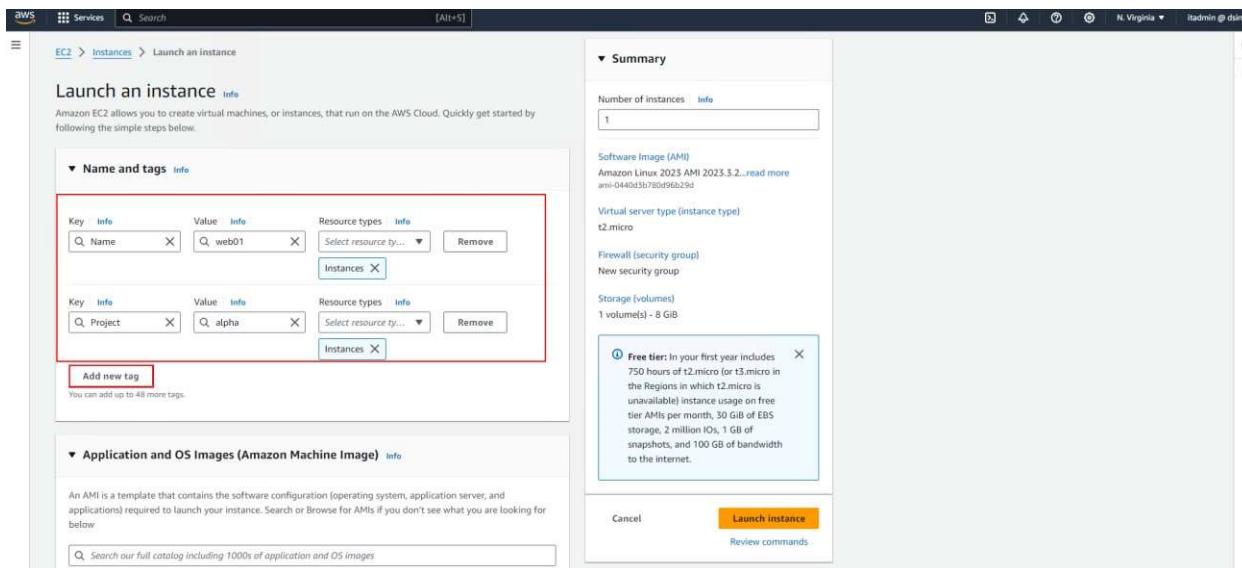
Step 5:

Specify a **TAG** for the instance.

To define a tag, click on the **Add Additional Tag** link.



In this section, you can define a **TAG** for your instance.



Step 6:

Specify the AMI (Amazon Machine Image).

To get future Google Chrome updates, you'll need Windows 10 or later. This computer is using Windows 8.1.

Selected AMI: ami-0440d3b780d96b29d (Quickstart AMIs)

Search for an AMI by entering a search term e.g. "Windows"

Quickstart AMIs (47) My AMIs (0) AWS Marketplace AMIs (9614) Community AMIs (500)

Refine results

All products (15 filtered, 47 unfiltered)

Free tier only info

Free tier eligible Verified provider

Select

64-bit (x86), uefi-preferred

64-bit (Arm), uefi

64-bit (x86)

64-bit (Arm)

Amazon Linux 2023 AMI

ami-0440d3b780d96b29d (64-bit (x86), uefi-preferred) / ami-0f93c02efd1974b8b (64-bit (Arm), uefi)

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

Platform: amazon Root device type: ebs Virtualization: hvm ENA enabled: Yes Select

Amazon Linux 2 AMI (HVM, SSD Volume Type)

ami-0776115aa34c447bd (64-bit (x86)) / ami-0e672303fbcb5cb75 (64-bit (Arm))

Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Platform: amazon Root device type: ebs Virtualization: hvm ENA enabled: Yes Select

Red Hat Enterprise Linux 9 (HVM), SSD Volume Type

ami-0fe630eb857a6ec83 (64-bit (x86)) / ami-0339ee0a14a92575d (64-bit (Arm))

Red Hat Enterprise Linux version 9 (HVM), EBS General Purpose (SSD) Volume Type

Platform: rhel Root device type: ebs Virtualization: hvm ENA enabled: Yes Select

64-bit (x86)

64-bit (Arm)

Step 7:

Select the Instance Type.

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

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

AMI from catalog Quick Start

Amazon Machine Image (AMI)

a2023-ami-2023.3.20240219.0-kernel-1-
x86_64
ami-0440d3b780d96b29d

Catalog Published Architecture Virtualization Root device type ENA Enabled

Quickstart AMIs 2024-02-16T21:29:42.00 0Z x86_64 hvm ebs Yes

Instance type Info | Get advice

t2.micro Family: t2 1 vCPU 1 GiB Memory Current generation: true Free tier eligible

On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.0716 USD per Hour On-Demand Linux base pricing: 0.0116 USD per Hour

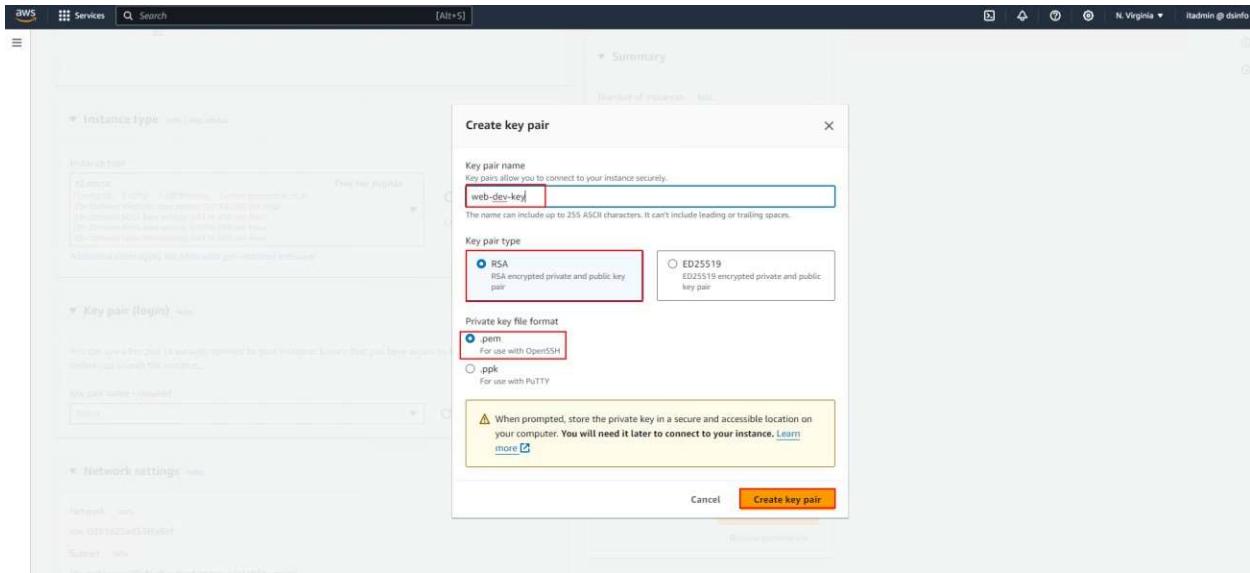
All generations Compare instance types

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel Launch instance Review commands.

Step 8:

At this stage, you need to define a **Key Pair** for logging in to the EC2 instance.



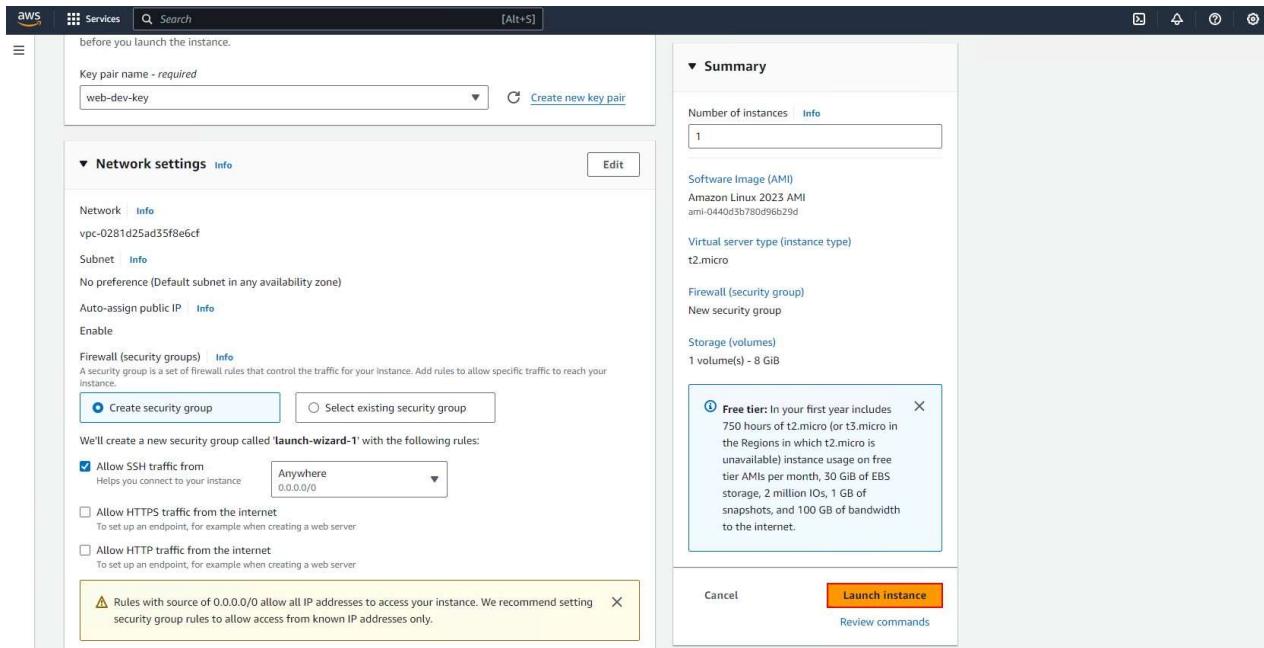
Step 9:

In this section, you can configure the **Security Group** settings.

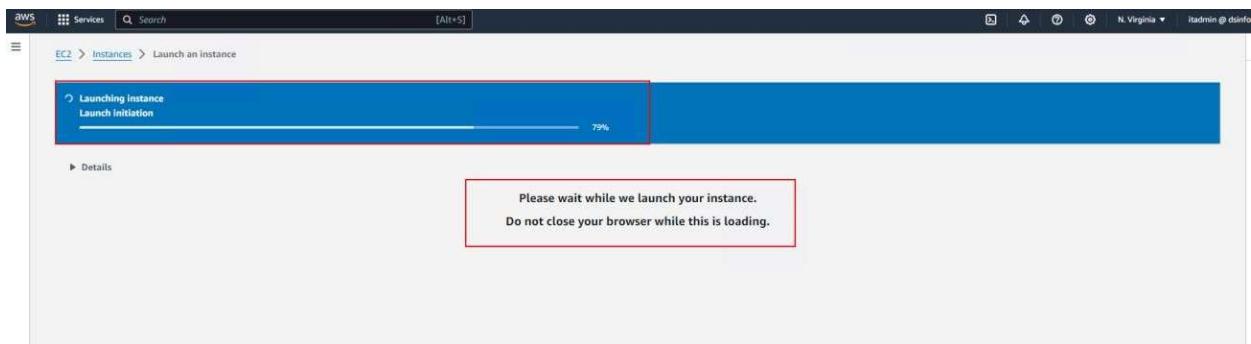
The screenshot shows the AWS Launch Wizard Step 9 interface. At the top, it says "before you launch the instance." A key pair named "web-dev-key" is selected. In the "Network settings" section, a red box highlights the "Firewall (security groups)" configuration. It shows two options: "Create security group" (selected) and "Select existing security group". Below this, under "Allow SSH traffic from", "Anywhere" is chosen. Under "Allow HTTPS traffic from the internet", there is a note: "To set up an endpoint, for example when creating a web server". Under "Allow HTTP traffic from the internet", there is another note: "To set up an endpoint, for example when creating a web server". A warning message at the bottom states: "⚠️ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." To the right, the "Summary" section shows 1 instance, AMI Amazon Linux 2023, t2.micro instance type, and a new security group. A tooltip for the free tier is displayed, and at the bottom are "Cancel", "Launch Instance", and "Review commands" buttons.

Step 10:

At this stage, click the **Launch Instance** button to create the virtual machine.



As shown in the image below, the **EC2 instance** is being created.



To view all instances, click on the **View all Instances** button.

The screenshot shows the AWS EC2 instance launch success page. At the top, a green banner says "Successfully initiated launch of instance i-04be684ae847aa856". Below it, a "Next Steps" section has a search bar and a navigation bar with pages 1 through 6. The main area contains eight cards:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. Includes a "Create billing alerts" button.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Includes a "Connect to instance" button.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Includes a "Connect an RDS database" button.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Includes a "Create EBS snapshot policy" button.
- Manage detailed monitoring**: Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period. Includes a "Manage detailed monitoring" button.
- Create Load Balancer**: Create a application, network gateway or classic Elastic Load Balancer. Includes a "Create Load Balancer" button.
- Create AWS budget**: AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location. Includes a "Create AWS budget" button.
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance. Includes a "Manage CloudWatch alarms" button.

At the bottom right is a red "View all Instances" button.

As you can see, the instance has been successfully created.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, Elastic Block Store, Network & Security, and more. The main area shows a table titled "Instances (1) Info" with one row:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
web01	i-04be684ae847aa856	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-52-204-48-129.co...	52.204.48.129	-

Below the table is a "Select an instance" dropdown menu.

Introduction to EC2 Best Practices

Before launching an EC2 instance, it's a good idea to become familiar with its **best practices**.

When you launch an EC2 instance, you need to have certain **requirements** in place, which include the following:

-Requirement Gathering

Defining the **task** — for example, setting up a web server, specifying the type of operating system, storage size, and compute size.

-Key Pair

Creating a **Key Pair**, which can be done either before or during the instance creation process.

-Security Group

Creating a **Security Group**, which acts as a firewall.

-Instance Launch

Launching the **Instance**.

When you launch an instance, you can select and configure all the **requirements** such as the **Security Group**, **Key Pair**, and more.

How to Create a Key Pair

To create a **Key Pair**, go to the **Key Pairs** section and click on the **Create Key Pair** button.

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, a search bar, and a message: "To get future Google Chrome updates, you'll need Windows 10 or later. This computer is using Windows 8.1." On the far right, it shows "N. Virginia" and "Itadmin @ dsinfo". Below the navigation bar is a toolbar with icons for refresh, actions, and other settings.

The main content area is titled "Key pairs [Info]" and contains a search bar with the placeholder "Find Key Pair by attribute or tag". A table header is visible with columns: Name, Type, Created, Fingerprint, and ID. A message below the table says "No key pairs to display".

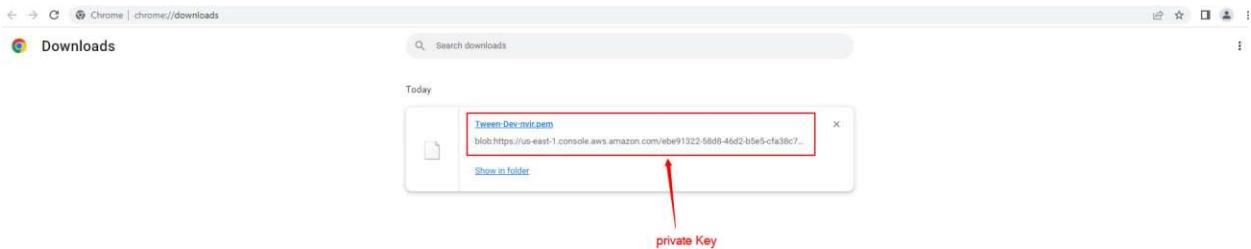
The left sidebar is a navigation menu with the following categories and sub-items:

- Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity Reservations
 - New
- Images
 - AMIs
 - AMI Catalog
- Elastic Block Store
 - Volumes
 - Snapshots
 - Lifecycle Manager
- Network & Security
 - Security Groups
 - Elastic IPs
 - Placement Groups
 - Key Pairs** (highlighted with a red border)
 - Network Interfaces
- Load Balancing
 - Load Balancers
 - Target Groups
 - Trust Stores New

At this stage, specify a **Key Pair Name**, set the key type to **RSA**, and choose the **PEM** format for the private key. Finally, click on the **Create Key Pair** button.

The screenshot shows the 'Create key pair' dialog box on the AWS EC2 service. The 'Name' field is filled with 'Tweet-Dev-nvir'. The 'Key pair type' is set to 'RSA'. The 'Private key file format' is selected as '.pem'. There are optional tags listed below, and a 'Create key pair' button at the bottom right.

Then, at this stage, download the **Private Key**.



The **Public Key** will be injected into your instance.

To get future Google Chrome updates, you'll need Windows 10 or later. This computer is using Windows 8.1.

EC2 Dashboard Services Search [Alt+S] Learn more ×

Key pairs (1/1) info

Find Key Pair by attribute or tag

Name	Type	Created	Fingerprint	ID
Tweener-Dev-envir	rsa	2024/02/26 02:35 GMT-8	b8:54:e8:8a:e2:0b:c1:a1:99:51:3b:e6:2b:79:12:ce:6a:2b:d3:c6	key-0926f25b0b179c978

Actions Create key pair < 1 > ⚙️

EC2 Dashboard Instances Images Elastic Block Store Network & Security Key Pairs

EC2 Global View Events Console-to-Code Preview

Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations New

Images AMIs AMI Catalog

Elastic Block Store Volumes Snapshots Lifecycle Manager

Network & Security Security Groups Elastic IPs Placement Groups Key Pairs

The **Private Key** must match your **Public Key**.

How to Create a Security Group

A **Security Group** is a **firewall**, and we can create it **before** launching an EC2 instance.

You can apply a single **Security Group** to multiple instances.

A **Security Group** works in a **stateful** manner. For example, if you allow access to port **22** for anyone in the **Inbound Rules**, the same rule is automatically applied to the **Outbound Rules**, so you don't need to define outbound rules separately.

To create a **Security Group**, follow these steps:

First, go to the **Security Groups** section and click on the **Create Security Group** button.

Name	Security group ID	Security group name	VPC ID	Description	Owner
sg-0ff8e31663de4497	sg-0ff8e31663de4497	default	vpc-0281d25ad55fbedcf	default VPC security group	093418366137

At this stage, you need to specify a **name** for the Security Group, and then define its **rules**. There are two types of rules in a Security Group:

-Inbound Rule

This rule is for **inbound traffic** to the instance.

-Outbound Rule

This rule is for **outbound traffic** from the instance and usually doesn't need to be modified.

In this section, we intend to **open access to all inbound ports**. To do this, click on the **Add Rule** button under **Inbound Rules**, and then define the rule accordingly.

The screenshot shows the 'Create security group' page in the AWS Management Console. The 'Basic details' section includes fields for 'Security group name' (tween-web-dev-sg), 'Description' (tween-web-dev-sg), and 'VPC' (vpc-0281d25ad35f8e6cf). The 'Inbound rules' section is expanded, showing a message: 'This security group has no inbound rules.' A red box highlights the 'Add rule' button. The 'Outbound rules' section is collapsed.

After clicking the **Add Rule** button, add the following rule to allow **SSH access** to the instance's IP

The screenshot shows the 'Create security group' page after adding an inbound rule. The 'Inbound rules' section now displays a single rule: 'SSH' (Protocol: TCP, Port range: 22, Source: My IP, Destination: 84.32.10.4/32). The 'Outbound rules' section is also visible below.

Finally, to create the **Security Group**, click on the **Create security group** button.

The screenshot shows the AWS Management Console interface for creating a new security group. The top navigation bar includes 'Services' and a search bar. The main form has tabs for 'Type' (set to 'SSH'), 'Protocol' (set to 'TCP'), 'Port range' (set to '22'), 'Source' (set to 'My IP'), and 'Description - optional' (left empty). Below this is an 'Add rule' button. The next section, 'Outbound rules', shows a similar configuration with tabs for 'Type' (set to 'All traffic'), 'Protocol' (set to 'All'), 'Port range' (set to 'All'), 'Destination' (set to 'Custom'), and 'Description - optional' (left empty). An 'Add rule' button is also present here. A yellow warning message at the bottom states: '⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' A 'Tags - optional' section follows, with a note about tags and an 'Add new tag' button. The bottom right features a 'Create security group' button in orange.

As shown in the image below, the **Security Group** has been successfully created.

The screenshot shows the details page for the newly created security group, 'sg-Oea2028daccc1f67c - tween-web-dev-sg'. The left sidebar lists various AWS services like EC2 Dashboard, Instances, Images, and Network & Security. The main content area shows the security group's name, ID, owner, and VPC ID. Under the 'Inbound rules' tab, there is one rule listed: 'sg-Oea40ffcb41934f771' (Name), IPv4 (IP version), SSH (Type), TCP (Protocol), port 22 (Port range), and source 84.32.10.4/32 (Source). The 'Edit inbound rules' button is highlighted with a red box.

As shown in the image below, the **Security Group** has been successfully created.

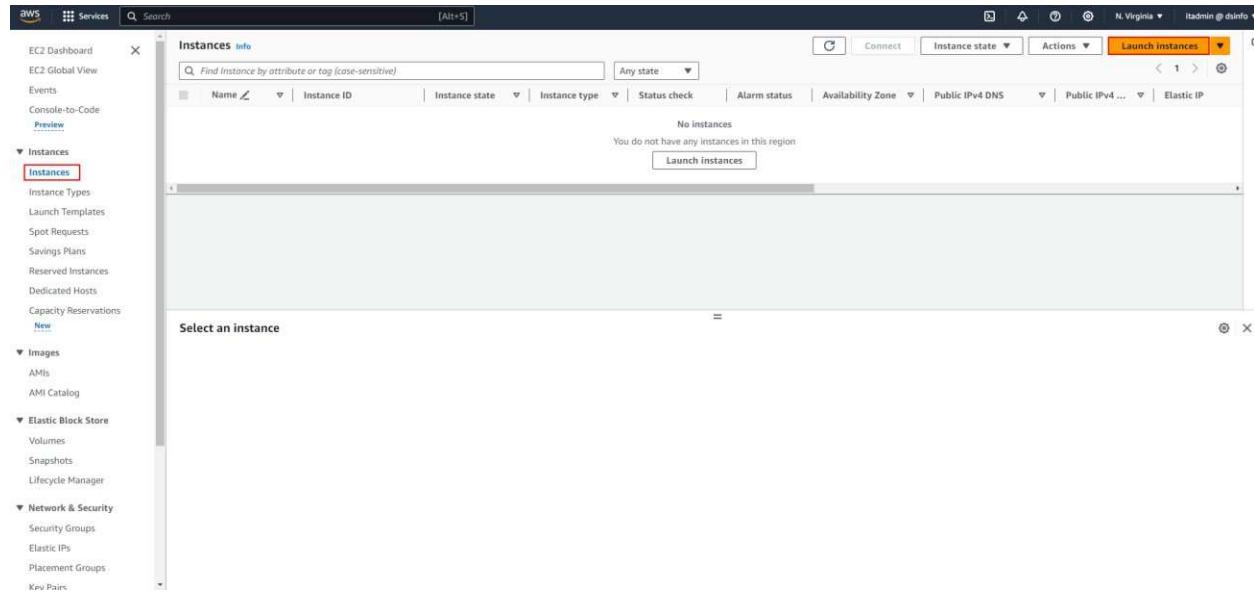
The screenshot shows the AWS EC2 Security Groups page. The left sidebar navigation includes: EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances (with sub-options: Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (with sub-options: AMIs, AMI Catalog), Elastic Block Store (with sub-options: Volumes, Snapshots, Lifecycle Manager), Network & Security (with sub-options: Security Groups, Elastic IPs, Placement Groups, Kv Pair), and a New section. The main content area shows a security group named "sg-0ea2028dacc1f67c - tween-web-dev-sg". The "Details" tab is selected, displaying information such as Security group name (tween-web-dev-sg), Security group ID (sg-0ea2028dacc1f67c), Owner (093418366137), Description (tween-web-dev-sg), and VPC ID (vpc-0281d25ad35f8e6cf). Below this, the "Outbound rules" tab is selected, showing one rule: "sgr-025e0fda1da7ba6cd" (Name) allowing IPv4 traffic on all ports to 0.0.0.0/0. A search bar and a "Manage tags" button are also present.

This **Outbound Rule** allows access to **any IP** and **any port** from the instance to the outside.

Deploying a Web Server on an EC2 Instance

Now, in this section, we intend to create an **EC2 instance** and use the **Key Pair** and **Security Group** that we previously created.

To create an EC2 instance, go to the **Instances** section and click the **Launch Instance** button.



At this stage, click on the **Add Additional Tag** link to create a tag.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' section, there is a text input field with 'e.g. My Web Server' and a red box highlighting the 'Add additional tags' button. Below this, the 'Application and OS Images (Amazon Machine Image)' section is visible, featuring a search bar and a 'Quick Start' grid of AMI icons. On the right, a summary panel shows 'Number of instances' set to 1, and a detailed 'Software Image (AMI)' section. A tooltip for the 'Free tier' is displayed, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth to the internet.' At the bottom are 'Cancel', 'Launch instance' (in orange), and 'Review commands' buttons.

As shown in the image below, we have defined **4 tags** for the instance.

This screenshot shows the same EC2 wizard at the 'Name and tags' step, but with four tags added. The tags are listed in a table:

Key	Value
Name	web01
Environment	Prod
Owner	LeadDevOps
Project	Tween

A red box highlights the 'Add new tag' button and the note 'You can add up to 46 more tags.' The rest of the interface is identical to the previous screenshot, including the summary panel and the 'Launch instance' button.

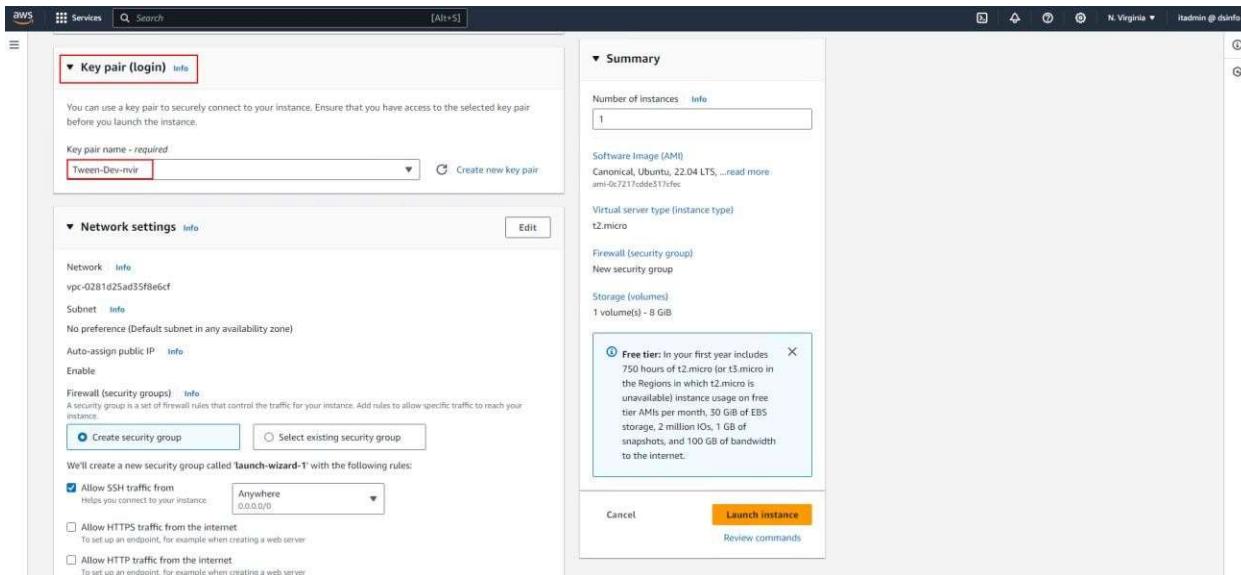
At this stage, you need to select the **AMI type**, and in this scenario, we are choosing **Ubuntu**, which, as you can see, is **Free Tier eligible**.

The screenshot shows the AWS CloudFormation console interface. In the search bar, 'Project' and 'Tween' are listed. Below the search bar, there's a 'Select resource type...' dropdown and a 'Remove' button. A 'Instances' button is also present. Under the 'Application and OS Images (Amazon Machine Image)' section, a tooltip for the 'ubuntu' AMI states: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.' Other AMI options shown include Amazon Linux, macOS, Windows, Red Hat, and SUSE.

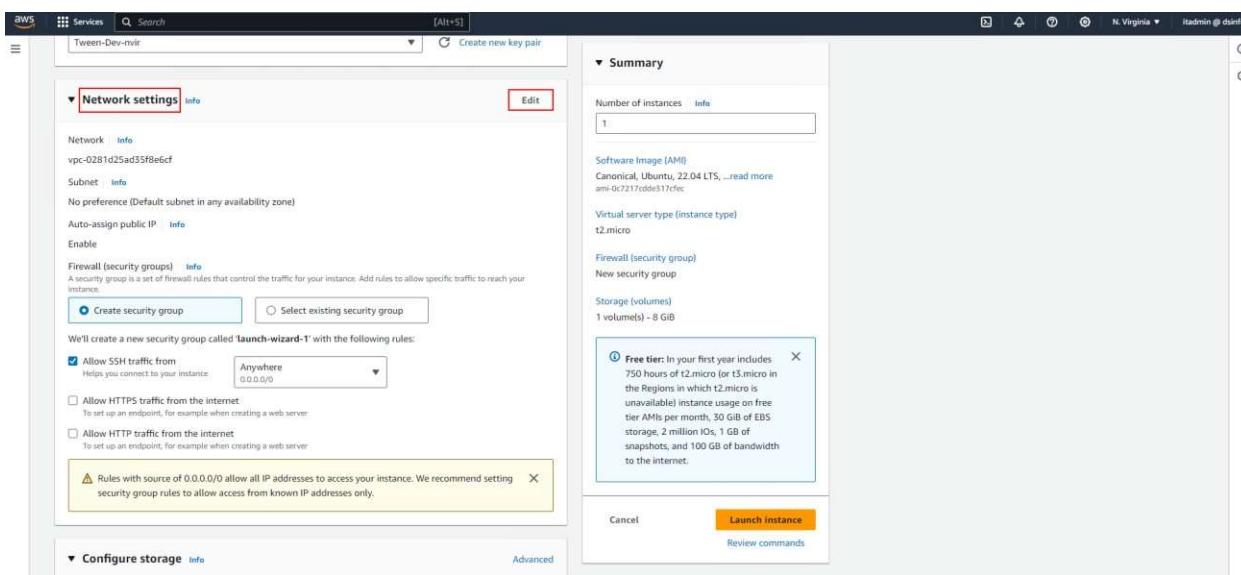
At this stage, you need to specify the **Instance Type**, which defines the hardware specifications of the instance. In this scenario, we are selecting **t2.micro**, which is **Free Tier eligible**.

The screenshot shows the AWS CloudFormation console interface. In the search bar, 'Project' and 'Tween' are listed. Below the search bar, there's a 'Select resource type...' dropdown and a 'Remove' button. A 'Launch instance' button is visible. On the left, under 'Instance type', 't2.micro' is selected and highlighted with a red box. A tooltip for 't2.micro' states: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.' Other instance types listed include t2.small, t2.medium, t2.large, t3.micro, t3.small, t3.medium, t3.large, t3.xlarge, and t3.2xlarge.

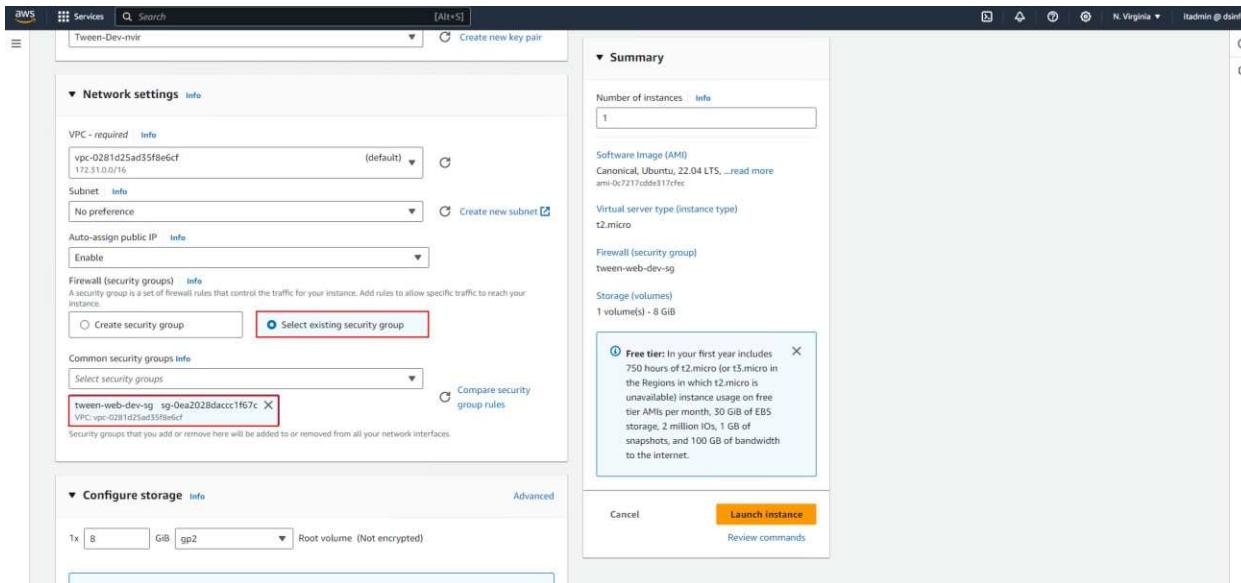
In this section, you need to specify the **Key Pair**, and we will apply the same Key Pair that we created earlier to this instance.



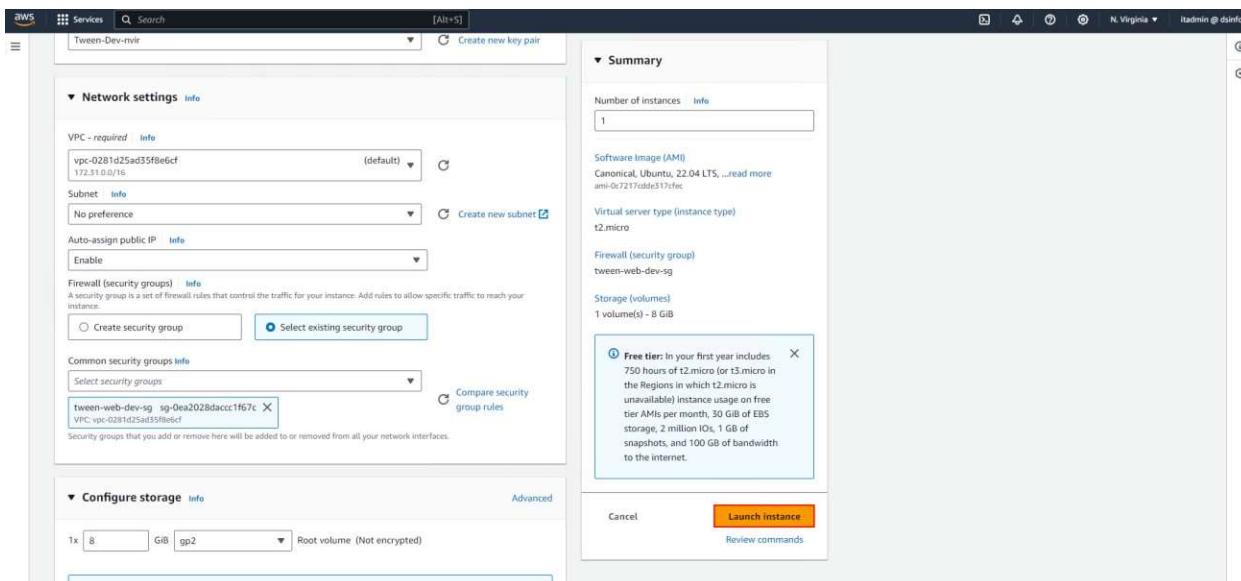
At this stage, to apply the **Security Group**, click on the **Edit** button in the **Network Settings** section.



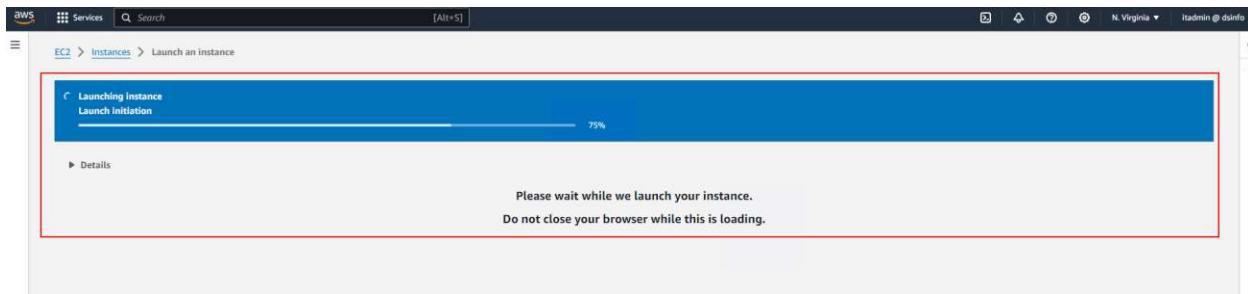
Then, select the **Select Existing Security Group** option, and choose the **Security Group** that we created in the previous steps.



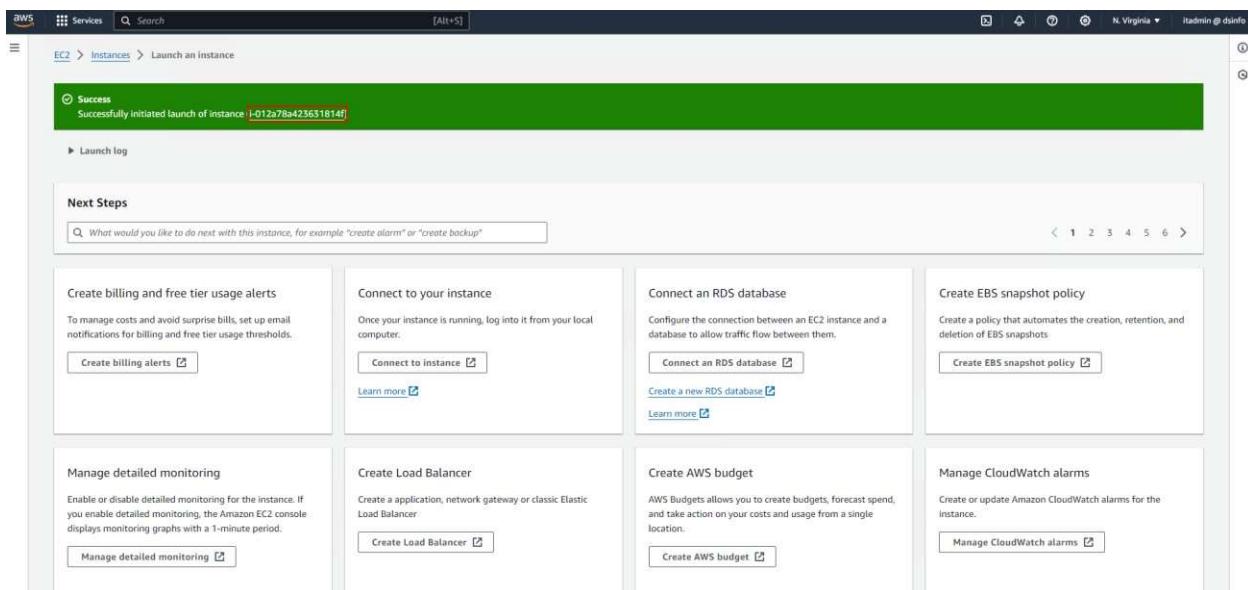
Finally, click on the **Launch Instance** button to create the instance.



As shown in the image below, the instance is being created.



The instance has been created. To view the details and status of this instance, simply click on its **Instance ID**.



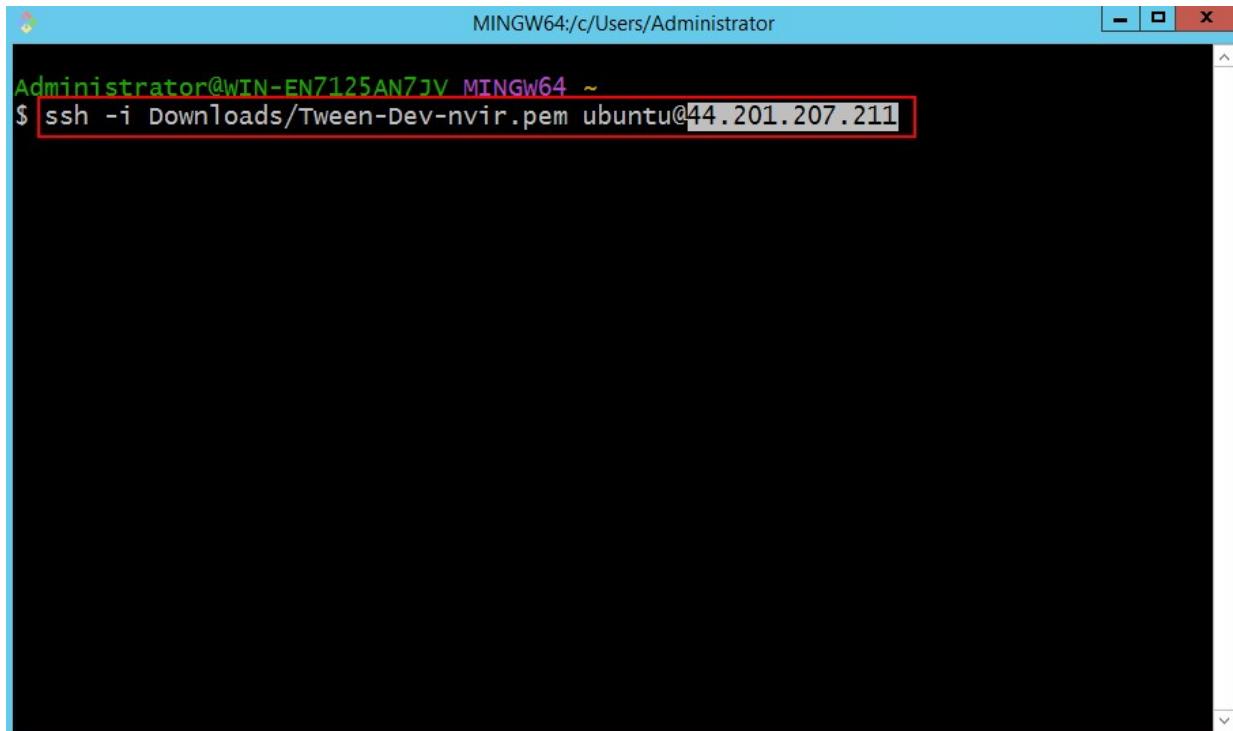
To connect to the instance, click on the **Connect** button.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with links like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, and Key Pairs. The main area displays a table titled "Instances (1/1) info". A single row is listed: "Instance ID: i-012a78a423631814f", "Name: web01", "Instance state: Running", "Status check: Initializing", "Instance type: t2.micro", "Availability Zone: us-east-1a", "Public IPv4 DNS: ec2-44-201-207-211.compute-1.amazonaws.com", "Public IPv4 IP: 44.201.207.211", and "Elastic IP: -". Below the table, a detailed view for "Instance: i-012a78a423631814f (web01)" is shown with tabs for Details, Status and alarms New, Monitoring, Security, Networking, Storage, and Tags. Under the Details tab, it shows Instance ID (i-012a78a423631814f), Public IPv4 address (44.201.207.211), Instance state (Running), Private IP DNS name (ip-172-31-94-165.ec2.internal), Instance type (t2.micro), and VPC ID. To the right, there are sections for Private IP4 addresses (172.31.94.165), Public IP4 DNS (ec2-44-201-207-211.compute-1.amazonaws.com), and AWS Compute Optimizer finding.

In the image below, you can see the required information for SSH to connect to the instance.

The screenshot shows the "Connect to instance" dialog box. It starts with a header "EC2 > Instances > i-012a78a423631814f > Connect to instance". Below that, it says "Connect to your instance i-012a78a423631814f (web01) using any of these options". There are three tabs: "EC2 Instance Connect" (selected), "SSH client" (highlighted with a red box), and "EC2 serial console". Under "EC2 Instance Connect", it shows the instance ID (i-012a78a423631814f). Under "SSH client", it shows the command "ssh -i 'Tween-Dev-nvr.pem' ubuntu@ec2-44-201-207-211.compute-1.amazonaws.com". A note below the command says: "Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username." Two red arrows point to this note: one from the word "Private Key" and another from the word "Username". At the bottom right of the dialog box is a "Cancel" button.

To connect to the instance from your system, you can use **SSH**. Make sure to use the **Private Key** associated with the instance for the connection.

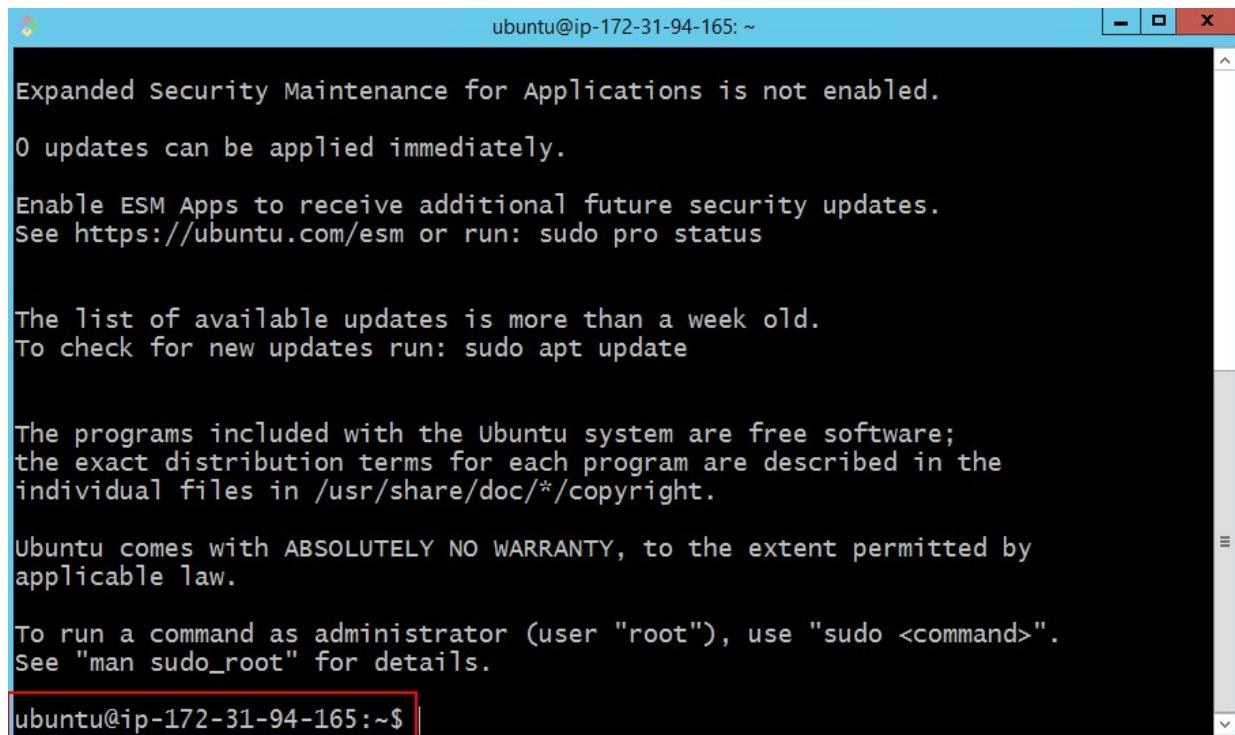


A screenshot of a terminal window titled "MINGW64:/c/Users/Administrator". The window contains the following text:

```
Administrator@WIN-EN7125AN7JV MINGW64 ~
$ ssh -i Downloads/Tween-Dev-nvir.pem ubuntu@44.201.207.211
```

The command `ssh -i Downloads/Tween-Dev-nvir.pem ubuntu@44.201.207.211` is highlighted with a red box.

As shown in the image below, we have successfully connected to the **Ubuntu terminal** on the EC2 instance via **SSH**.



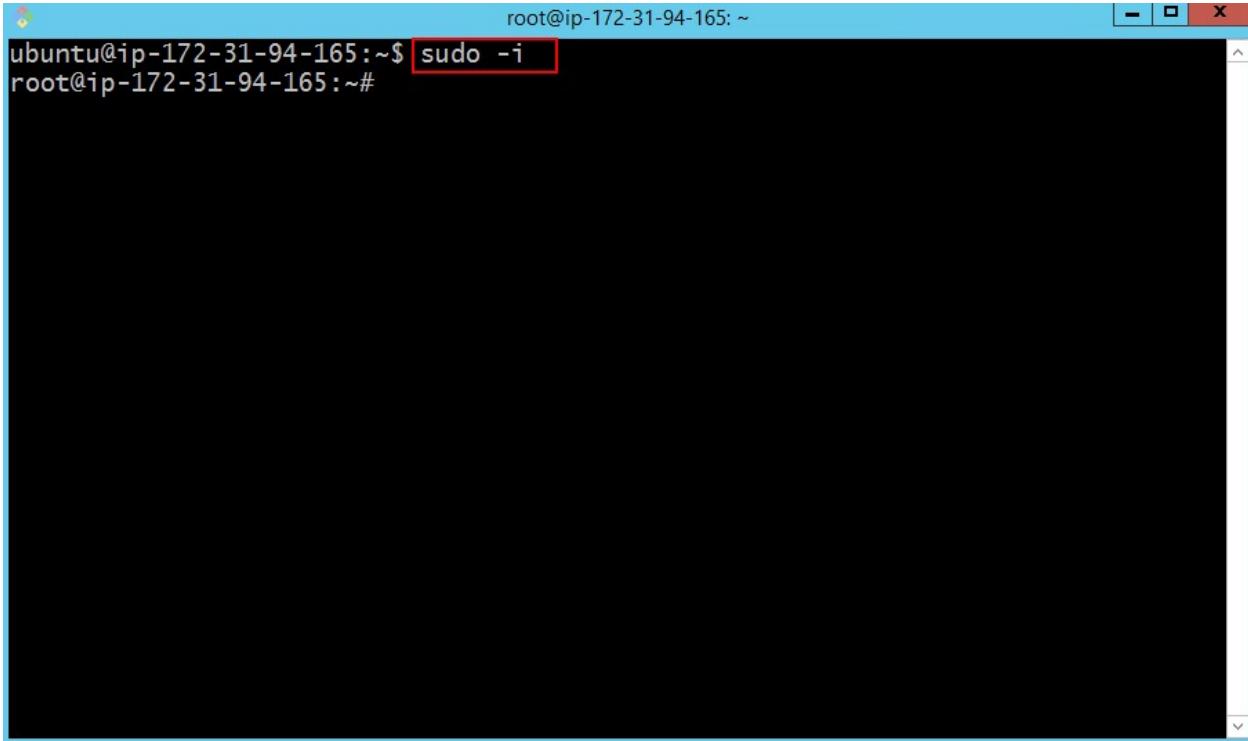
The screenshot shows a terminal window titled "ubuntu@ip-172-31-94-165: ~". The window contains the following text:

```
Expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.
```

The bottom line, "ubuntu@ip-172-31-94-165:~\$", is highlighted with a red border.

Now, in this section, we will install the **Tweet website**, which is one of the templates from **Tooplate**, on this EC2 instance.

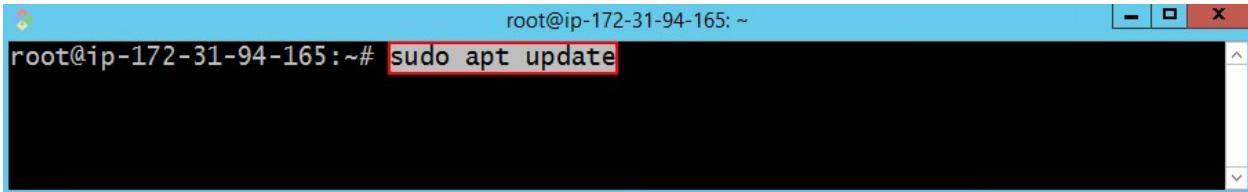
At this stage, use the following command to enter the **Root User** environment



A terminal window titled "root@ip-172-31-94-165: ~". The command "sudo -i" is highlighted with a red box. The response "root@ip-172-31-94-165:~#" is shown below it.

```
root@ip-172-31-94-165:~$ sudo -i
root@ip-172-31-94-165:~#
```

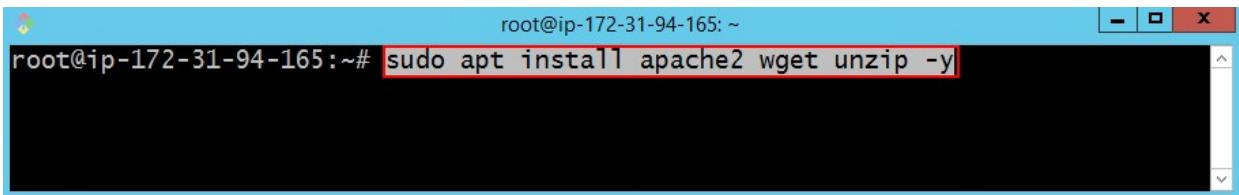
Then, use the following command to **update the repository** for the Linux instance



A terminal window titled "root@ip-172-31-94-165: ~". The command "sudo apt update" is highlighted with a red box.

```
root@ip-172-31-94-165:~# sudo apt update
```

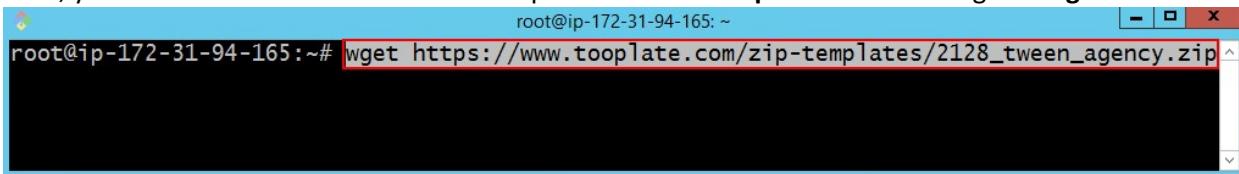
At this stage, use the following command to install the **Apache**, **Wget**, and **Unzip** packages



A terminal window titled "root@ip-172-31-94-165: ~". The command "sudo apt install apache2 wget unzip -y" is highlighted with a red box.

```
root@ip-172-31-94-165:~# sudo apt install apache2 wget unzip -y
```

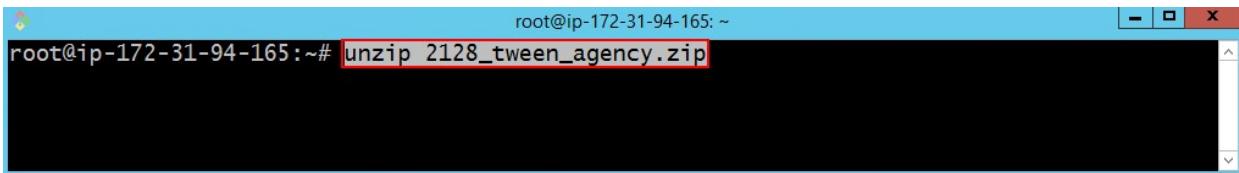
Next, you need to download the **Tween** template from the **Tooplate** website using the **Wget** command.



```
root@ip-172-31-94-165:~# wget https://www.tooplate.com/zip-templates/2128_tween_agency.zip
```

A terminal window titled 'root' with a blue header bar. The command 'wget https://www.tooplate.com/zip-templates/2128_tween_agency.zip' is highlighted in red at the bottom of the window.

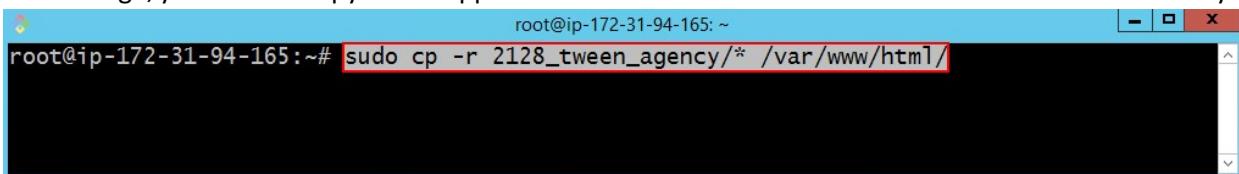
After downloading the **Tween** file, you need to unzip it using the following command



```
root@ip-172-31-94-165:~# unzip 2128_tween_agency.zip
```

A terminal window titled 'root' with a blue header bar. The command 'unzip 2128_tween_agency.zip' is highlighted in red at the bottom of the window.

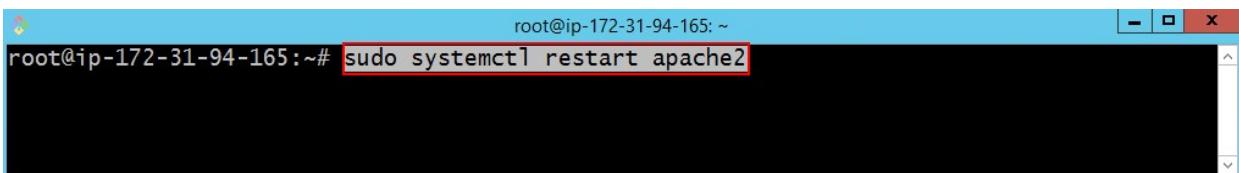
At this stage, you need to copy the unzipped contents of the **Tween** website to the web server's directory.



```
root@ip-172-31-94-165:~# sudo cp -r 2128_tween_agency/* /var/www/html/
```

A terminal window titled 'root' with a blue header bar. The command 'sudo cp -r 2128_tween_agency/* /var/www/html/' is highlighted in red at the bottom of the window.

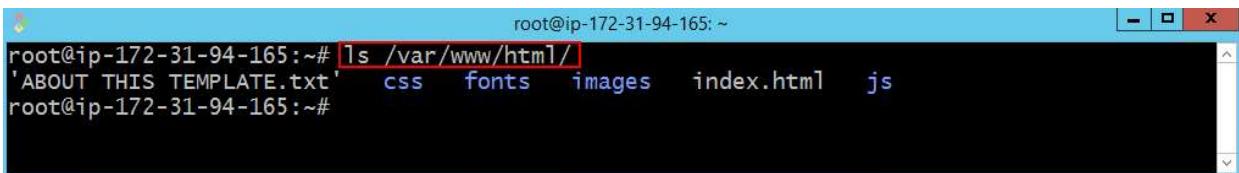
Finally, you need to restart the **Apache** service using the following command



```
root@ip-172-31-94-165:~# sudo systemctl restart apache2
```

A terminal window titled 'root' with a blue header bar. The command 'sudo systemctl restart apache2' is highlighted in red at the bottom of the window.

You can check if the **Tween** website files have been copied to the web server directory using the following command



```
root@ip-172-31-94-165:~# ls /var/www/html/
'ABOUT THIS TEMPLATE.txt'  css  fonts  images  index.html  js
root@ip-172-31-94-165:~#
```

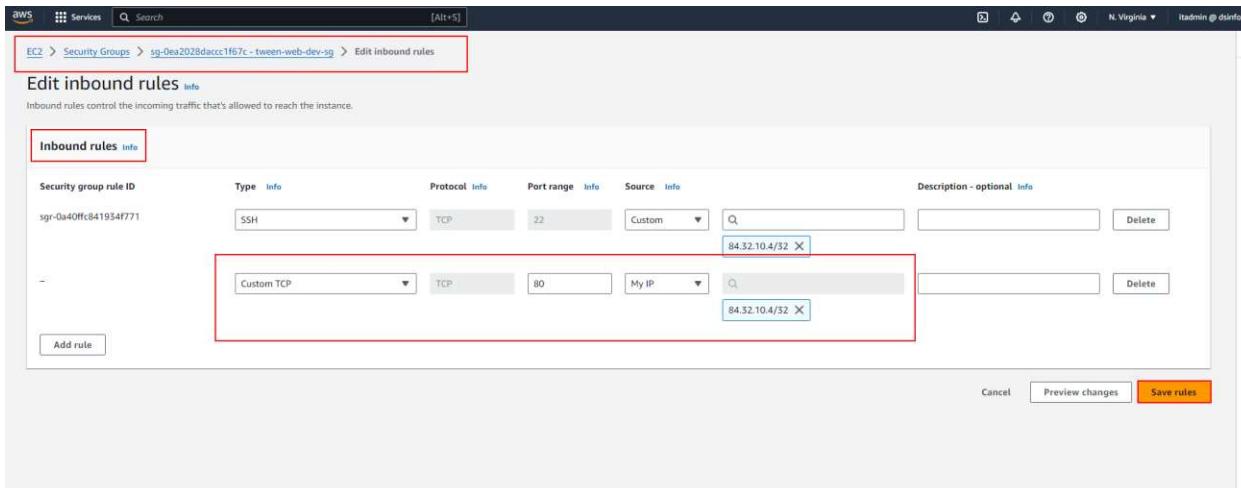
A terminal window titled 'root' with a blue header bar. The command 'ls /var/www/html/' is highlighted in red at the bottom of the window. The output shows several files: 'ABOUT THIS TEMPLATE.txt', 'css', 'fonts', 'images', 'index.html', and 'js'.

You can check if the **Apache** service is running by using the following command

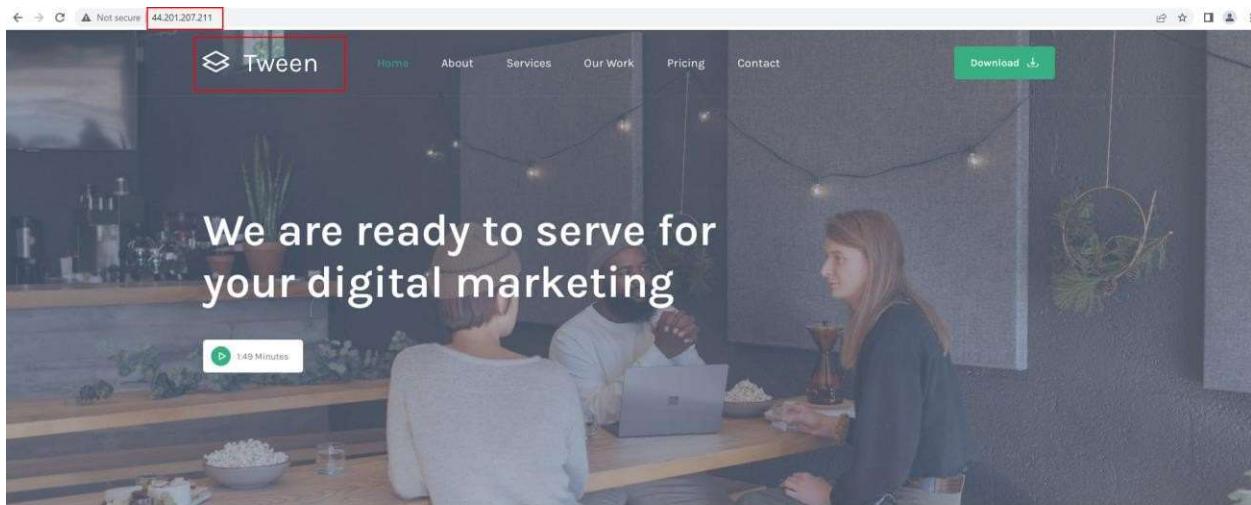
```
root@ip-172-31-94-165:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-02-26 12:44:08 UTC; 1min 42s ago
     Docs: https://httpd.apache.org/docs/2.4/
lines 1-4...skipping...
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-02-26 12:44:08 UTC; 1min 42s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 2223 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 2229 (apache2)
    Tasks: 55 (limit: 1121)
   Memory: 4.7M
      CPU: 30ms
     CGroup: /system.slice/apache2.service
             └─2229 /usr/sbin/apache2 -k start
                 ├─2230 /usr/sbin/apache2 -k start
                 ├─2231 /usr/sbin/apache2 -k start

Feb 26 12:44:08 ip-172-31-94-165 systemd[1]: Starting The Apache HTTP Server...
Feb 26 12:44:08 ip-172-31-94-165 systemd[1]: Started The Apache HTTP Server.
lines 1-16...skipping...
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-02-26 12:44:08 UTC; 1min 42s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 2223 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 2229 (apache2)
```

At this stage, you need to open port 80 in the **Inbound Rule** to allow access to the website from the internet.



Now, we can access the **Tween** website by entering the **Public IP** of the instance in our browser.



Digital Happiness

[Introduction](#) [Profile](#) [FAQs](#)

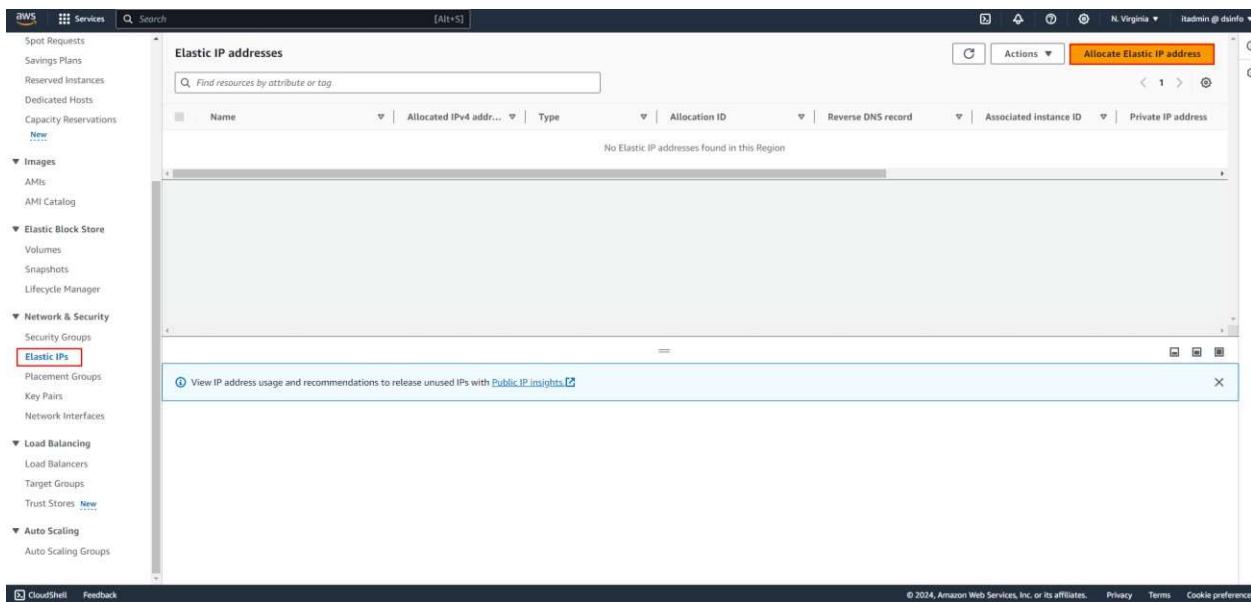
Introduction to Elastic IP

When you launch an EC2 instance, AWS automatically assigns a **Public IP**. However, when you stop the instance, the Public IP is released, and when the instance is started again, it is assigned a new **Public IP**.

To use a **static Public IP**, you need to use AWS's **Elastic IP** feature, which provides you with up to **5 Public IPs**. If you need more IPs, you must submit a support ticket to request additional IPs.

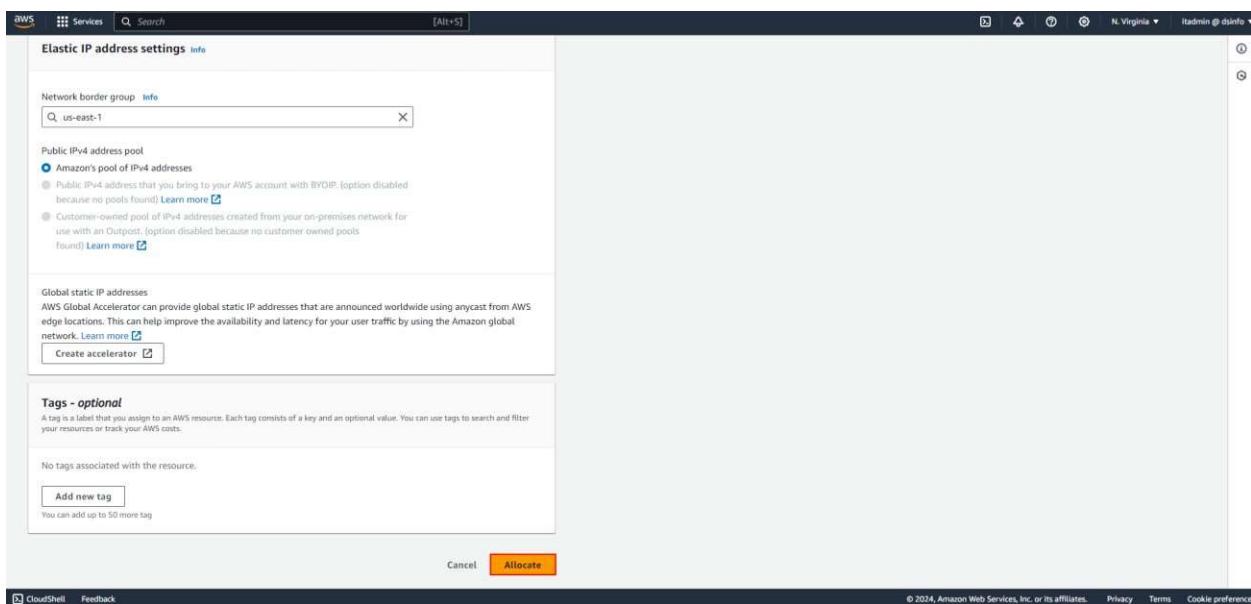
How to Create an Elastic IP

To create an **Elastic IP**, simply click on the **Elastic IPs** link, then click the **Allocate Elastic IP Address** button.



The screenshot shows the AWS Management Console interface for managing Elastic IP addresses. The left sidebar navigation includes services like Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AML Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security (with 'Elastic IPs' selected), Security Groups, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Auto Scaling, and CloudWatch Metrics. The main content area displays a table with columns for Name, Allocated IPv4 address, Type, Allocation ID, Reverse DNS record, Associated instance ID, and Private IP address. A search bar at the top allows filtering by attribute or tag. An orange 'Allocate Elastic IP address' button is located in the top right corner of the main content area.

Finally, click on the **Allocate** button.



The screenshot shows the 'Allocate Elastic IP address settings' configuration page. It includes fields for 'Network border group' (set to 'us-east-1'), 'Public IPv4 address pool' (selected 'Amazon's pool of IPv4 addresses'), 'Global static IP addresses' (disabled), and 'Tags - optional' (no tags associated). At the bottom, there are 'Cancel' and 'Allocate' buttons. The 'Allocate' button is highlighted with a yellow background.

By creating an **Elastic IP**, this **Public IP** will be reserved for us.

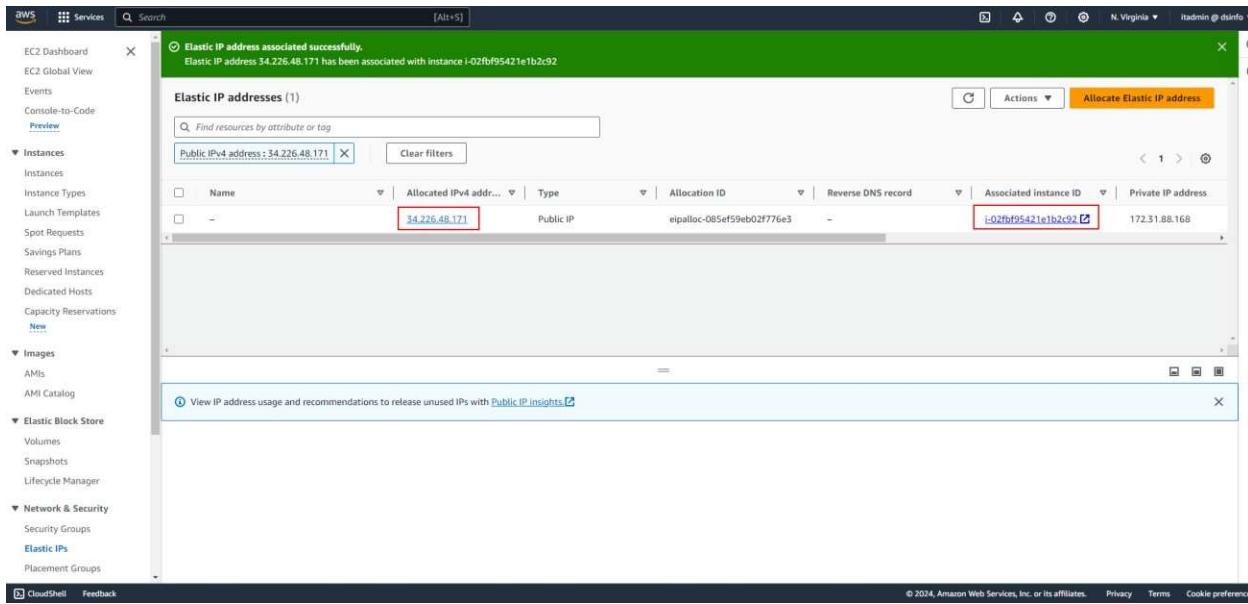
To associate the **Elastic IP** with an instance, go to the **Actions** menu and click on **Associate Elastic IP Address**.

The screenshot shows the AWS EC2 Dashboard. In the center, a green success message box displays: "Elastic IP address allocated successfully. Elastic IP address 34.226.48.171". Below this, the "Elastic IP addresses (1/1)" table lists one entry: "Allocated IPv4 address: 34.226.48.171", "Type: Public IP", and "Allocation ID: eipalloc-085ef59eb02f776e3". To the right of the table, a context menu is open under the "Actions" button, with "Associate Elastic IP address" highlighted. At the bottom of the page, a summary card for the IP address 34.226.48.171 provides basic details like Allocation ID and Type.

Next, in the **Instance** section, select the desired instance, and then click the **Associate** button.

The screenshot shows the "Associate Elastic IP address" dialog box. It starts with a header: "Associate Elastic IP address" and "Choose the instance or network interface to associate to this Elastic IP address (34.226.48.171)". The "Resource type" section has a radio button for "Instance" selected. A warning message states: "If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account." Below this, a note says: "If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address." The "Instance" field contains the ID "i-02fb95421e1b2c92". The "Private IP address" field is empty. The "Reassociation" section has a checkbox for "Allow this Elastic IP address to be reassociated". At the bottom, there are "Cancel" and "Associate" buttons.

As shown in the image below, the **Instance ID** has been successfully associated with our **Elastic IP**.



The screenshot shows the AWS Elastic IP addresses page. A green header bar at the top indicates that an elastic IP address has been associated successfully. The main table displays one row of data:

Name	Allocated IPv4 address	Type	Allocation ID	Reverse DNS record	Associated instance ID	Private IP address
-	34.226.48.171	Public IP	eipalloc-085ef59eb02f776e3	-	i-02fbff95421e1b2c92	172.31.88.168

The Public IPv4 address (34.226.48.171) and the Associated instance ID (i-02fbff95421e1b2c92) are highlighted with red boxes.

Note:

All network settings, such as **Security Group**, **Elastic IP**, etc., are applied to the **Network Interface** on the instance.

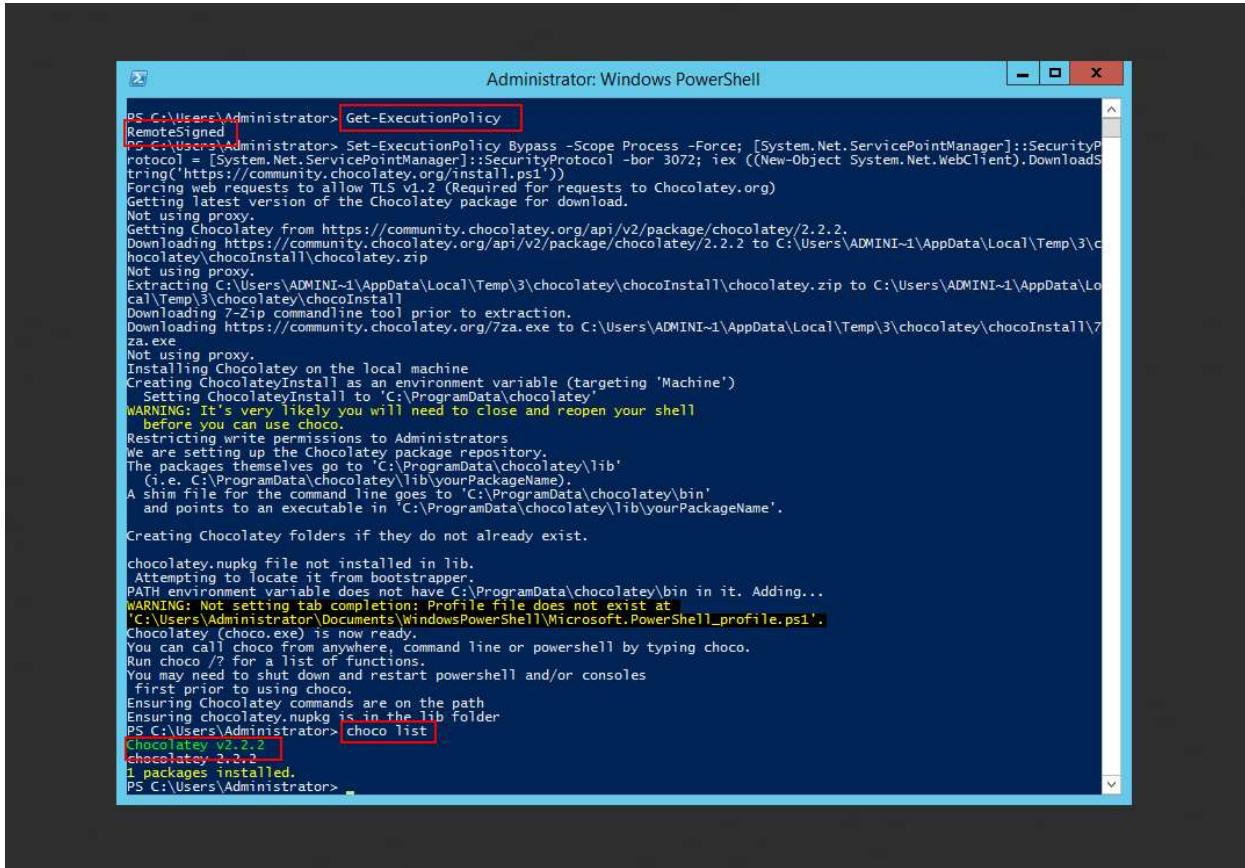
Introduction to AWS CLI

We can perform all the tasks that we do through the **console** or **graphical interface** using the **CLI** (Command Line Interface). As a DevOps professional, you should be familiar with these commands and work with them efficiently.

Installing AWS CLI

To install AWS CLI, you first need to install the **Chocolatey** tool on your **Windows PowerShell** environment, as shown in the image below.

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object
System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```

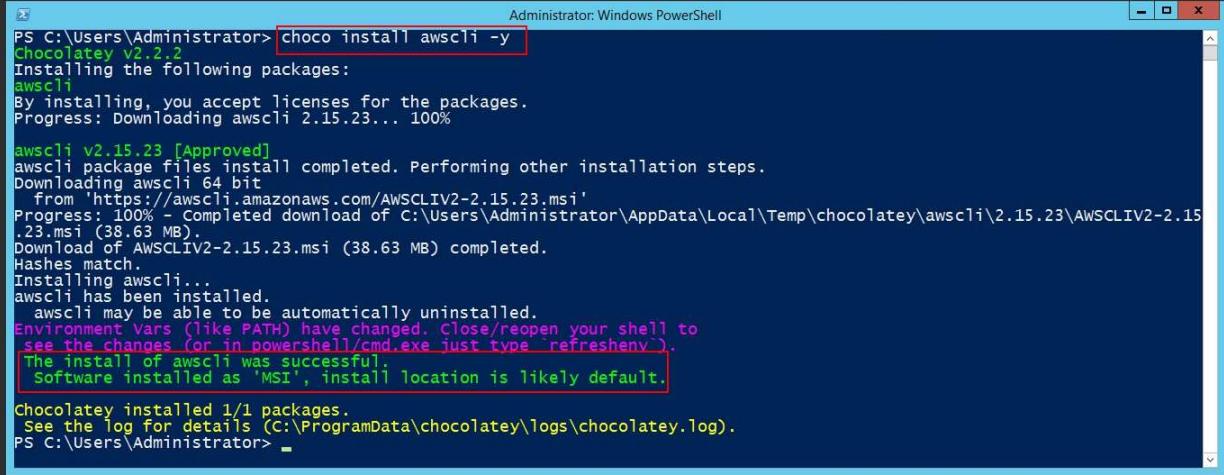


The screenshot shows an Administrator Windows PowerShell window. The command `Get-ExecutionPolicy` is run, showing `RemoteSigned` as the current policy. The command to install Chocolatey is then run, which involves downloading the Chocolatey package and its dependencies. The output shows the progress of the download and extraction of the `chocoInstall\chocolatey.zip` file to `C:\Users\ADMINI~1\AppData\Local\Temp\3\chocoInstall`. It also shows the download of the 7-Zip command-line tool and its execution. The Chocolatey tool is then installed, setting up environment variables and creating necessary folders. Finally, the `choco list` command is run, showing that Chocolatey V2.2.2 has been installed, and 1 package is listed.

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Get-ExecutionPolicy
RemoteSigned
PS C:\Users\Administrator> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
Forcing web requests to allow TLS v1.2 (Required for requests to Chocolatey.org)
Getting latest version of the Chocolatey package for download.
Not using proxy.
Getting Chocolatey from https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2.
Downloading https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2 to C:\Users\ADMINI~1\AppData\Local\Temp\3\chocoInstall\chocolatey.zip
Not using proxy.
Extracting C:\Users\ADMINI~1\AppData\Local\Temp\3\chocoInstall\chocolatey.zip to C:\Users\ADMINI~1\AppData\Local\Temp\3\chocoInstall
Downloading 7-Zip commandline tool prior to extraction.
Downloading https://community.chocolatey.org/7za.exe to C:\Users\ADMINI~1\AppData\Local\Temp\3\chocoInstall\7za.exe
Not using proxy.
Installing Chocolatey on the local machine
Creating ChocolateyInstall as an environment variable (targeting 'Machine')
Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.
Creating Chocolatey folders if they do not already exist.

chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at
'C:\Users\Administrator\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
first prior to using choco.
Ensuring Chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
PS C:\Users\Administrator> choco list
Chocolatey V2.2.2
chocolatey 2.2.2
1 packages installed.
PS C:\Users\Administrator>
```

After installing **Chocolatey** in the **PowerShell** environment, you need to install the **AWS CLI** tool using the following command



```
Administrator: Windows PowerShell
PS C:\Users\Administrator> choco install awscli -y
Chocolatey v2.2.2
Installing the following packages:
awscli
By installing, you accept licenses for the packages.
Progress: Downloading awscli 2.15.23... 100%
awscli v2.15.23 [Approved]
awscli package files install completed. Performing other installation steps.
Downloading awscli 64 bit
  from 'https://awscli.amazonaws.com/AWSCLIV2-2.15.23.msi'
Progress: 100% - Completed download of C:\Users\Administrator\AppData\Local\Temp\chocolatey\awscli\2.15.23\AWSCLIV2-2.15
.23.msi (38.63 MB).
Download of AWSCLIV2-2.15.23.msi (38.63 MB) completed.
Hashes match
Installing awscli...
awscli has been installed.
awscli may be able to be automatically uninstalled.
Environment vars (like PATH) have changed. Close/reopen your shell to
see the changes (or in powershell/cmd.exe just type 'refreshenv').
The install of awscli was successful.
Software installed as 'MSI', install location is likely default.

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Users\Administrator>
```

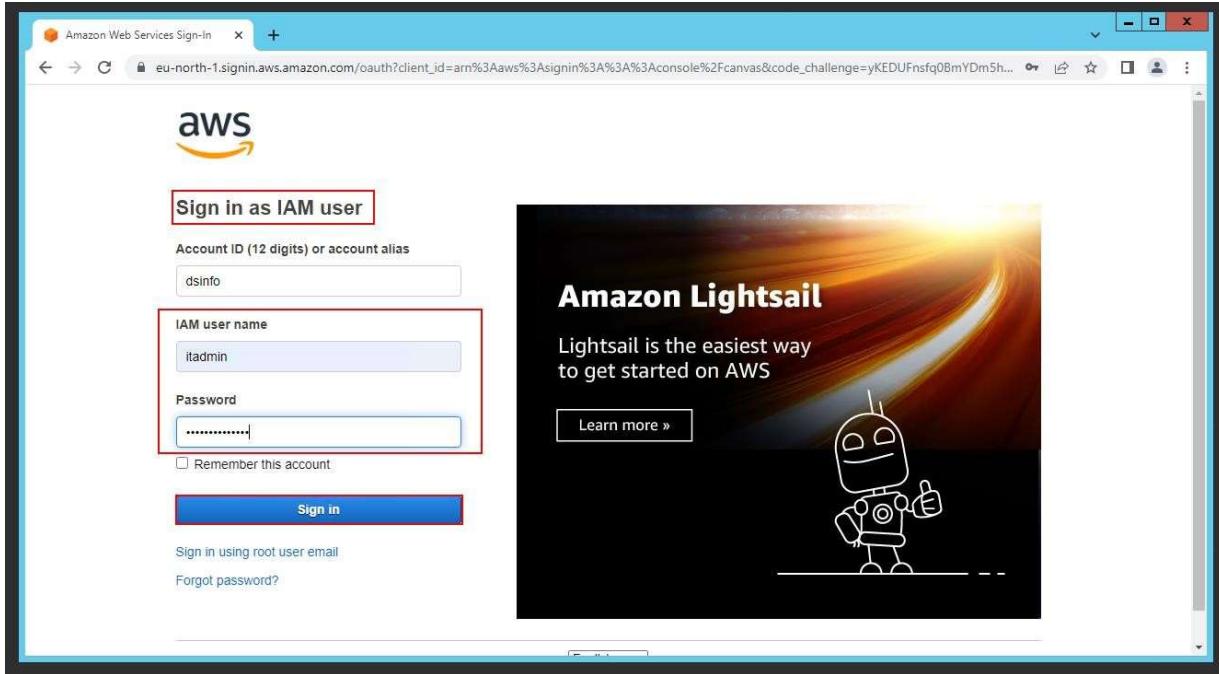
To check if **AWS CLI** has been installed correctly, use the following command in the **Git Bash** environment



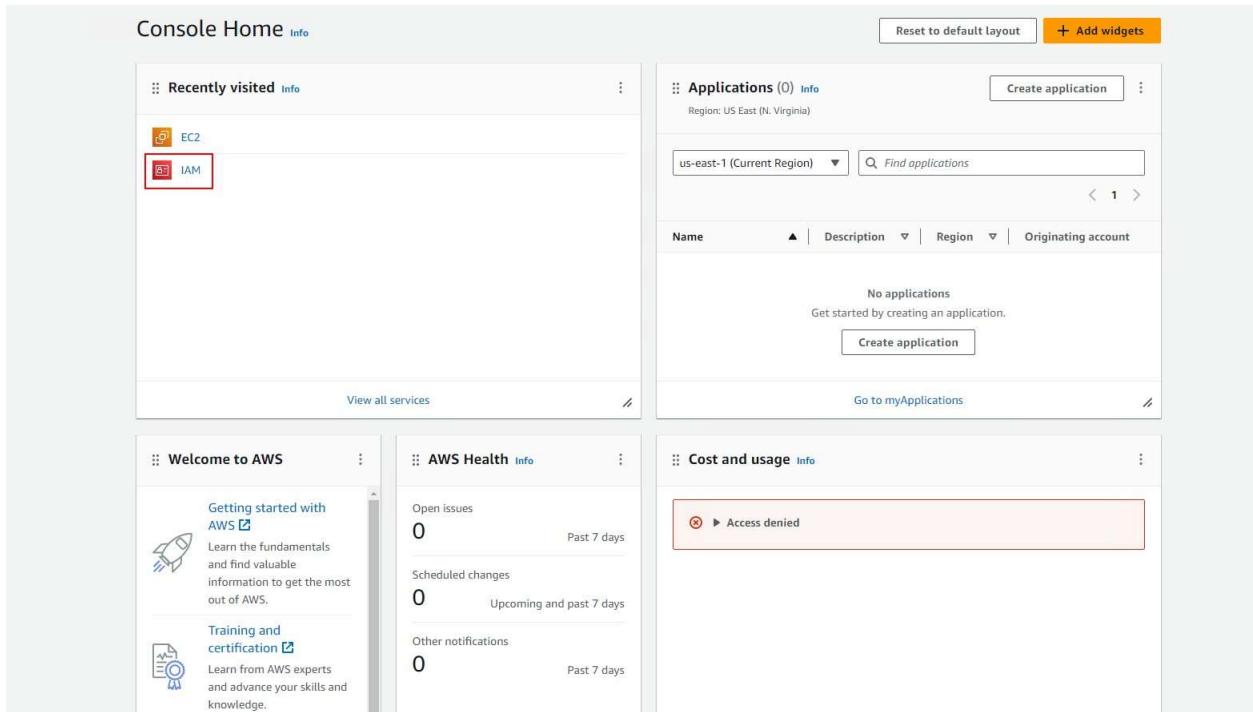
```
Administrator@WIN-EN7125AN7JV MINGW64 ~
$ aws --version
aws-cli/2.15.23 Python/3.11.6 windows/2012ServerR2 exe/AMD64 prompt/off

Administrator@WIN-EN7125AN7JV MINGW64 ~
$
```

At this stage, you need to log in to the **AWS Management Console** using an **IAM User** to create a user for accessing the AWS CLI environment.

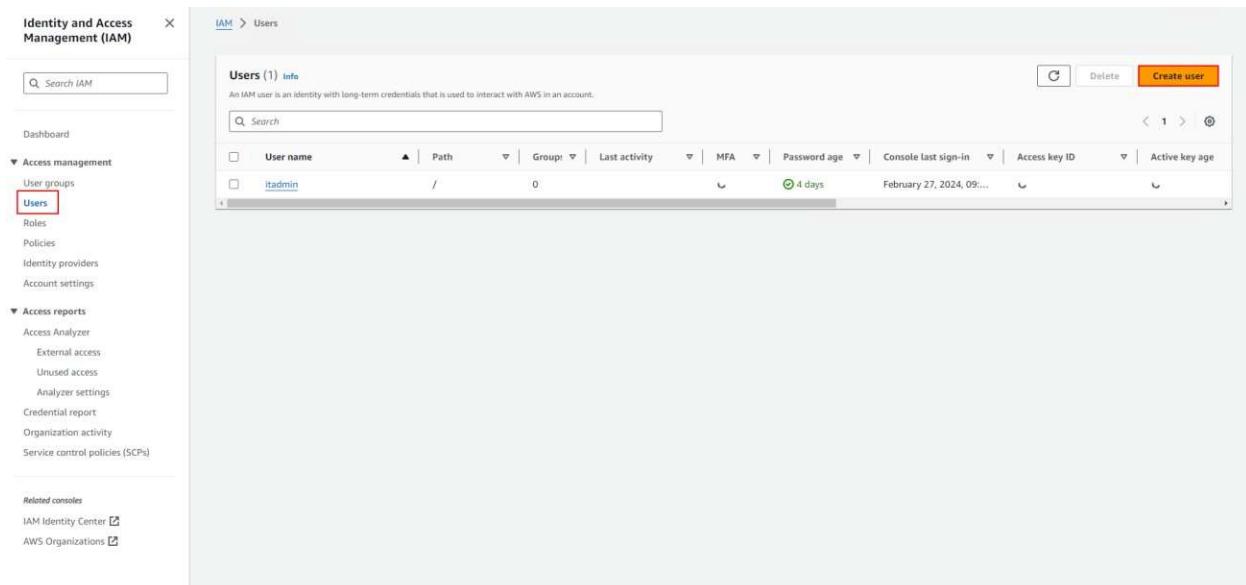


At this stage, click on **IAM** to enter the **IAM Console** page.



The screenshot shows the AWS Console Home page. On the left, there's a sidebar with links like 'Recently visited' (EC2, IAM), 'Welcome to AWS' (Getting started with AWS, Training and certification), and 'AWS Health' (Open issues: 0, Scheduled changes: 0, Other notifications: 0). On the right, there are sections for 'Applications' (0) and 'Cost and usage'. The 'Applications' section has a 'Create application' button. The 'Cost and usage' section has a 'Access denied' message. At the bottom, there's a 'Go to myApplications' link. The 'IAM' link in the sidebar is highlighted with a red box.

To create a user for AWS CLI, click on the **Users** link, and then click on the **Create User** button.



The screenshot shows the AWS IAM 'Users' page. The left sidebar has 'Access management' expanded, with 'Users' selected and highlighted with a red box. The main area shows a table of users with one entry: 'itadmin'. The table includes columns for 'User name', 'Path', 'Group', 'Last activity', 'MFA', 'Password age', 'Console last sign-in', 'Access key ID', and 'Active key age'. At the top right of the table, there are 'Create user', 'Delete', and other buttons. A search bar is also present at the top.

In the **Username** section, define a user but make sure the option **User Access to AWS Management Console** is not selected. Then, click the **Next** button.

The screenshot shows the 'Specify user details' step of the IAM User creation wizard. On the left, a sidebar lists 'Step 1 Specify user details', 'Step 2 Set permissions', and 'Step 3 Review and create'. The main area is titled 'User details' and contains a 'User name' field with 'awscli' typed in. Below the field is a note: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + - (hyphen)'. There is also an optional checkbox for 'Provide user access to the AWS Management Console'. A note below it says: 'If you're providing console access to a person, it's a best practice [link] to manage their access in IAM Identity Center.' At the bottom right are 'Cancel' and 'Next' buttons.

At this stage, select the **Attach policies directly** option, and then choose a **policy** for the user

The screenshot shows the 'Set permissions' step of the IAM User creation wizard. The sidebar shows 'Step 1 Specify user details' is selected. The main area is titled 'Permissions options' and contains three radio button options: 'Add user to group', 'Copy permissions', and 'Attach policies directly'. The 'Attach policies directly' option is selected and highlighted with a red box. Below this, a table titled 'Permissions policies (1/1176)' lists policies. One policy, 'AdministratorAccess', is selected and highlighted with a red box. The table includes columns for 'Policy name', 'Type', and 'Attached entities'. At the bottom right are 'Create policy' and 'Next' buttons.

Policy name	Type	Attached entities
<input type="checkbox"/> AccessAnalyzerServiceRolePolicy	AWS managed	0
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed - job function	1
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	0
<input type="checkbox"/> AdministratorAccess-AWSElasticBeanstalk	AWS managed	0
<input type="checkbox"/> AlexaForBusinessDeviceSetup	AWS managed	0
<input type="checkbox"/> AlexaForBusinessFullAccess	AWS managed	0
<input type="checkbox"/> AlexaForBusinessGatewayExecution	AWS managed	0
<input type="checkbox"/> AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	0

Next, click on the **Create User** button.

The screenshot shows the 'Review and create' step of the IAM 'Create user' wizard. It includes sections for 'User details' (username: awscli, console password type: None, require password reset: No), 'Permissions summary' (AdministratorAccess, AWS managed - job function), and 'Tags - optional' (no tags associated). At the bottom are 'Cancel', 'Previous', and a prominent orange 'Create user' button.

After creating the user, select it, and in the **Security Credentials** section, click on the **Create Access Key** button. The Access Key acts like a **username** and **password**, and AWS CLI can use it to connect to AWS.

The screenshot shows the IAM user details page for 'awscli'. The 'Security credentials' tab is selected, displaying 'Console access' (disabled), 'Last console sign-in' (February 27, 2024, 09:14 UTC-08:00), and 'Access key 1' (Create access key). Below this, the 'Console sign-in' section shows a 'Console sign-in link' (https://signin.aws.amazon.com/console) and 'Console password' (not enabled). The 'Multi-factor authentication (MFA)' section indicates 'No MFA devices' and has a 'Create MFA device' button. The 'Access keys (0)' section shows a note about best practices and a 'Create access key' button.

At this stage, select the **Command Line Interface** option, and then click on the **Next** button.

The screenshot shows the second step of the IAM Access Key creation wizard. The left sidebar lists steps 1, 2 (selected), and 3. Step 2 is titled "Access key best practices & alternatives". It contains a section titled "Use case" with several options: "Command Line Interface (CLI)" (selected), "Local code", "Application running on an AWS compute service", "Third-party service", "Application running outside AWS", and "Other". Below this is a "Alternatives recommended" section with links to "AWS CloudShell" and "AWS CLI V2". A "Confirmation" section at the bottom has a checked checkbox for accepting recommendations. At the bottom right are "Cancel" and "Next" buttons.

Then, click on the **Create Access Key** button.

The screenshot shows the third step of the IAM Access Key creation wizard. The left sidebar lists steps 1, 2, and 3. Step 3 is titled "Set description tag". It has a "Description tag value" input field with placeholder text about purpose and allowed characters. At the bottom are "Cancel", "Previous", and "Create access key" buttons, with "Create access key" being highlighted.

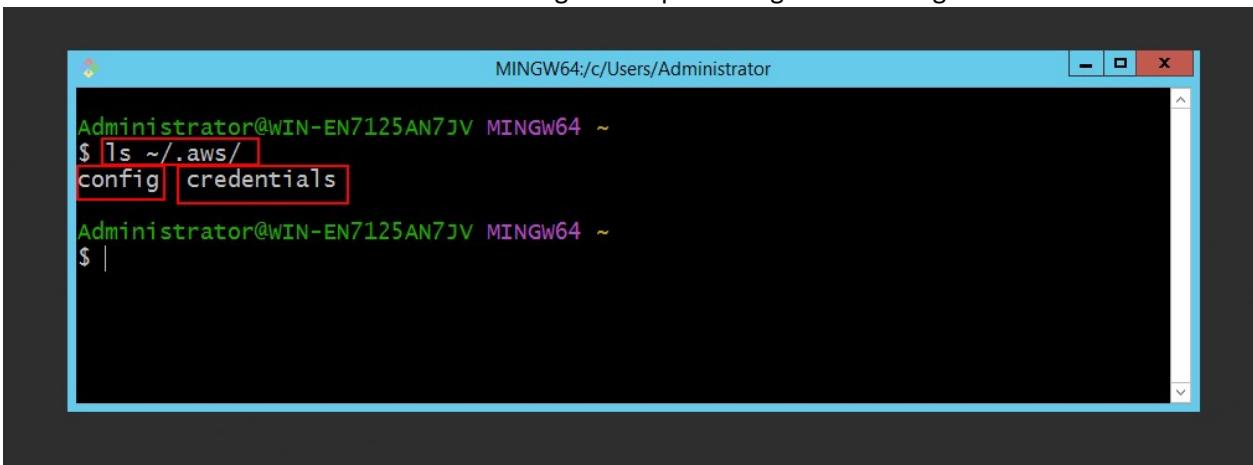
In this section, you will see the **Access Key** and **Secret Access Key**, which are required for logging in through AWS CLI. Click on the **Download .CSV** button and store this information in a secure location.

The screenshot shows the 'Create access key' page in the AWS IAM console. At the top, a green banner says 'Access key created' with a note: 'This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' Below the banner, the navigation path is IAM > Users > awscli > Create access key. The main section is titled 'Retrieve access keys' with an 'Info' link. It shows 'Step 1' (Access key best practices & alternatives) and 'Step 2 - optional' (Set description tag). The 'Access key' field contains 'AKIARLQBLES4TPWDBOXT' and the 'Secret access key' field starts with '4w8PXdPcJY6Dq8+G961BRhTnJT...'. A 'Show' link is next to the secret key. Below this is a 'Access key best practices' section with tips: 'Never store your access key in plain text, in a code repository, or in code.', 'Disable or delete access key when no longer needed.', 'Enable least-privilege permissions.', and 'Rotate access keys regularly.' A note says 'For more details about managing access keys, see the best practices for managing AWS access keys.' At the bottom are 'Download .csv file' and 'Done' buttons.

To allow **AWS CLI** to connect to AWS, it needs to be configured. Use the following command to specify your **Access Key**, **Secret Key**, **Region**, and **Data Format**

The screenshot shows a terminal window titled 'MINGW64:/c/Users/Administrator'. The command \$ aws configure is run. The output shows the configuration parameters: 'AWS Access Key ID [None]: AKIARLQBLES4TPWDBOXT', 'AWS Secret Access Key [None]: 4w8PXdPcJY6Dq8+G961BRhTnJTwsOf7+71nqK0j', 'Default region name [None]: us-east-1', and 'Default output format [None]: json'. The entire output is highlighted with a red box.

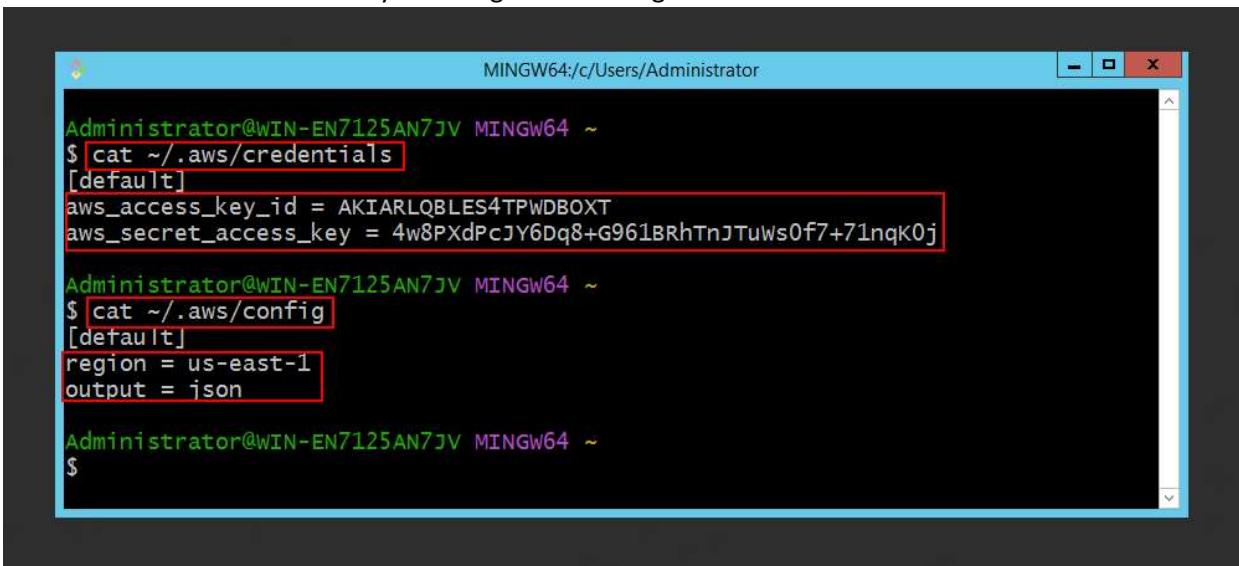
You can view the files created in the AWS configuration path using the following command



```
Administrator@WIN-EN7125AN7JV MINGW64 ~
$ ls ~/.aws/
config  credentials
```

The terminal window title is "MINGW64:/c/Users/Administrator". The command entered is "\$ ls ~/.aws/". The output shows two files: "config" and "credentials". The "config" and "credentials" files are highlighted with red boxes.

You can view the contents of any file using the following command



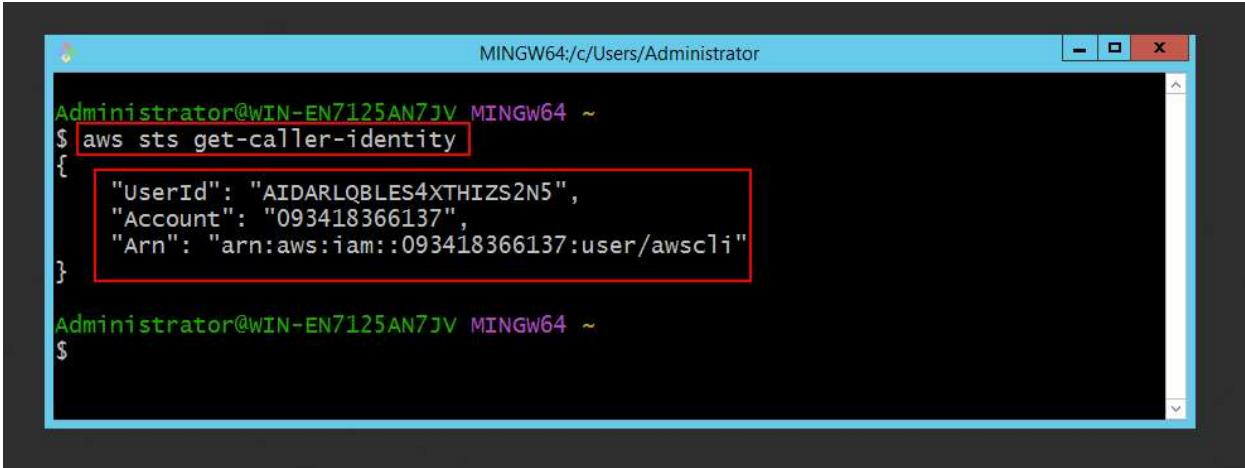
```
Administrator@WIN-EN7125AN7JV MINGW64 ~
$ cat ~/.aws/credentials
[default]
aws_access_key_id = AKIARLQBLE84TPWDBOXt
aws_secret_access_key = 4w8PXdPcJY6Dq8+G961BRhTnJuws0f7+71nqk0j

Administrator@WIN-EN7125AN7JV MINGW64 ~
$ cat ~/.aws/config
[default]
region = us-east-1
output = json

Administrator@WIN-EN7125AN7JV MINGW64 ~
$
```

The terminal window title is "MINGW64:/c/Users/Administrator". The command entered is "\$ cat ~/.aws/credentials". The output shows the [default] section of the credentials file, which contains the AWS access key ID and secret access key. The [default] section and its contents are highlighted with a red box.

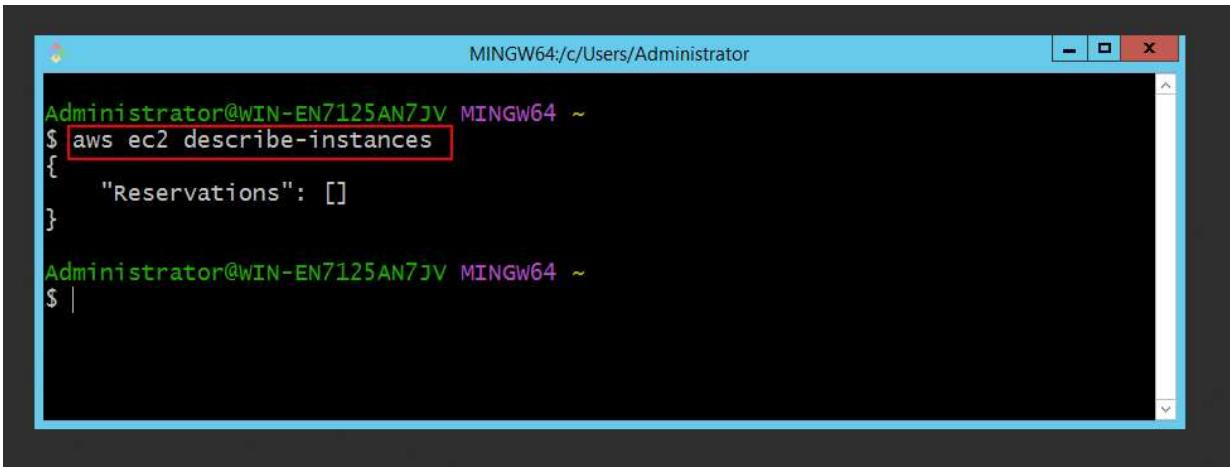
To view the identity using the AWS CLI, you can use the following command



```
Administrator@WIN-EN7125AN7JV MINGW64 ~
$ aws sts get-caller-identity
{
    "UserId": "AIDARLQBLES4XTHIZS2N5",
    "Account": "093418366137",
    "Arn": "arn:aws:iam::093418366137:user/awscli"
}

Administrator@WIN-EN7125AN7JV MINGW64 ~
$
```

To view the details of an EC2 instance, you can use the following command



```
Administrator@WIN-EN7125AN7JV MINGW64 ~
$ aws ec2 describe-instances
{
    "Reservations": []
}

Administrator@WIN-EN7125AN7JV MINGW64 ~
$ |
```

Introduction to Elastic Block Store (EBS)

EBS is generally a **virtual disk** that can be used with an EC2 instance and has two main purposes: one, it can be used as an **EBS Volume** for virtual disk storage; and two, as a **Snapshot**, which is used to back up an EBS Volume.

Features of EBS

- **Block Storage**, like a hard disk
- Used for **EC2 operating systems** or **data storage** for databases and more.

For example, the **EC2 EBS Volume** serves as the **Root Volume**, where the **operating system data** is stored.

You can also store other data such as **databases**, **web server content**, or any other files on it.

- EBS resides within an **Availability Zone**, enabling **replication across zones**, which increases overall **availability**.
- Use of **Snapshots** to back up EBS Volumes.

Types of EBS

You can choose from different types of EBS based on **price** and **performance**, including the following:

- **General Purpose SSD (gp2 / gp3)**

This type of EBS is used for a variety of workloads, such as running a **web server** on it.

- **Provisioned IOPS (io1 / io2)**

It is used for running **large databases** and provides **high performance** for your database workloads.

- **Throughput Optimized HDD (st1)**

This type of EBS is used for **Big Data** and **Data Warehouse** workloads.

- **Cold HDD (sc1)**

It is generally used for setting up a **file server** and has a very **low cost**.

-Magnetic

They have **low performance** and are mainly used for **backups** and **archiving data**, with a very **low cost**.

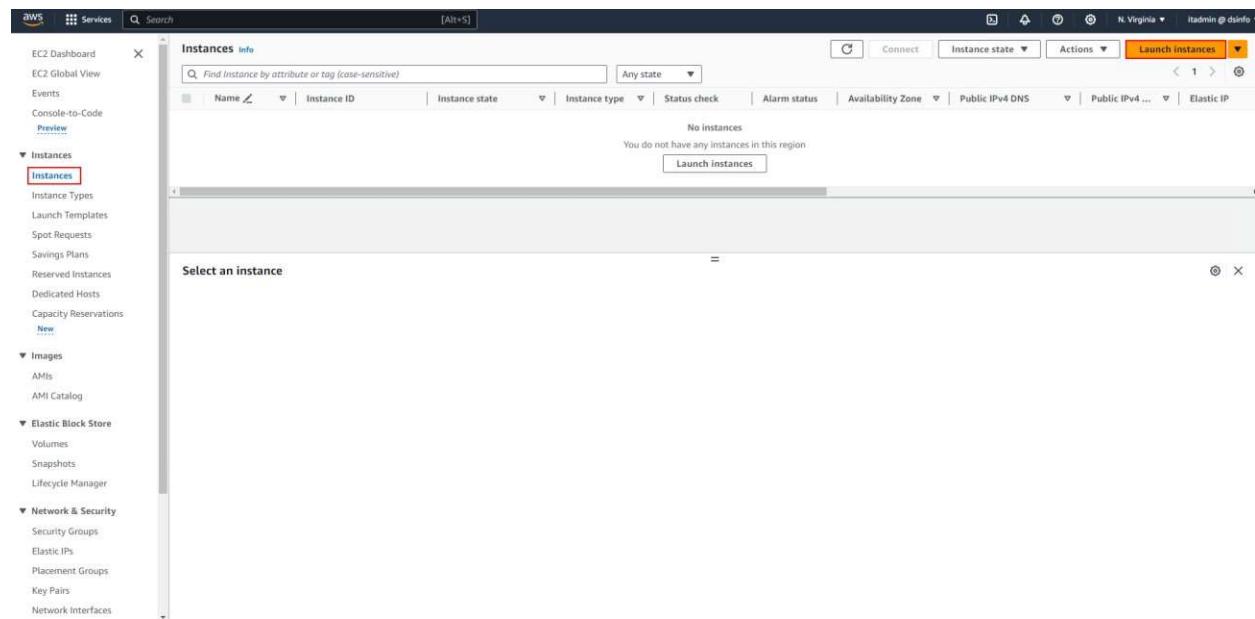
How to Create an EBS Volume

Before creating an EBS Volume, we first want to launch an **EC2 instance** with the **CentOS** operating system. Prior to launching, we will use a **script in the User Data section** so that a web server is automatically set up and a website is deployed on it. In other words, we want the web server setup to happen **automatically during instance creation**, instead of doing it **manually** afterward.

To complete this scenario, you need to follow these steps:

Step 1:

In the **Instances** section, click on the **Launch Instances** button.



Step 2:

In this step, specify a name in the **Name** section, for example: **web01**.

The screenshot shows the AWS EC2 'Launch an instance' wizard. On the left, under 'Name and tags', the 'Name' field is highlighted with the value 'web01'. On the right, the 'Summary' section shows the configuration: 1 instance, Amazon Linux 2023 AMI 2023.3.2, t2.micro instance type, New security group, and 1 volume(s) - 8 GiB storage. A tooltip for the 'Free tier' is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' At the bottom right are 'Cancel', 'Launch instance', and 'Review commands' buttons.

Step 3:

In the **AMI** section, select the **Ubuntu** image.

The screenshot shows the AWS CloudFormation console. In the left sidebar, under 'Create New Stack', 'From scratch' is selected. In the main area, the 'Template' tab is active, showing the CloudFormation template. The 'Parameters' tab is also visible. At the bottom, the 'Next Step' button is highlighted in orange.

At this stage, in the **Instance Type** section, select **t2.micro**.

The screenshot shows the AWS CloudFormation console. In the left sidebar, under 'Create New Stack', 'From scratch' is selected. In the main area, the 'Template' tab is active, showing the CloudFormation template. The 'Parameters' tab is also visible. The 'Instance type' dropdown has 't2.micro' selected. The 'Launch instance' button at the bottom right is highlighted in orange.

Step 4:

In the **Key Pair** section, click on the **Create new key pair** link and create a key pair for login.

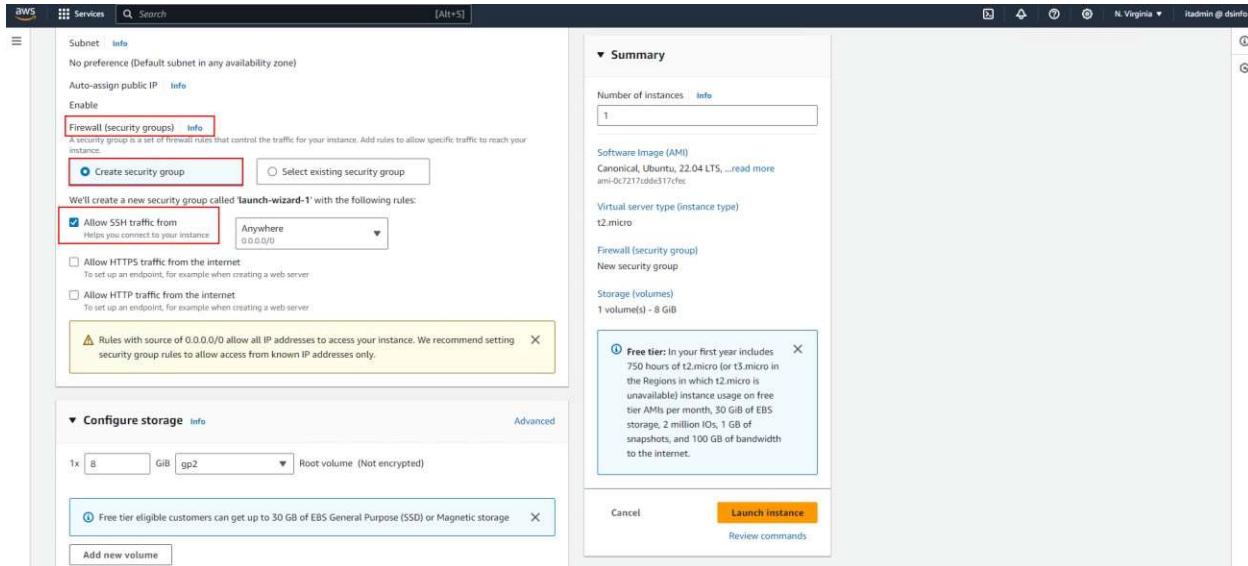
The screenshot shows the AWS CloudFormation console. In the 'Key pair (login)' section, under 'Network settings', there is a note: 'We'll create a new security group called "Launch-wizard-1" with the following rules:'. Below this, there are three checkboxes: 'Allow SSH traffic from Anywhere (0.0.0.0/0)', 'Allow HTTPS traffic from the internet', and 'Allow HTTP traffic from the internet'. The 'Allow SSH traffic from Anywhere (0.0.0.0/0)' checkbox is checked. The 'Summary' section on the right shows the following details: Number of instances: 1; Software Image (AMI): Canonical, Ubuntu, 22.04 LTS; Virtual server type (instance type): t2.micro; Firewall (security group): New security group; Storage (volumes): 1 volume(s) - 8 GiB. A note states: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.'

At this stage, create a Key Pair with the following specifications.

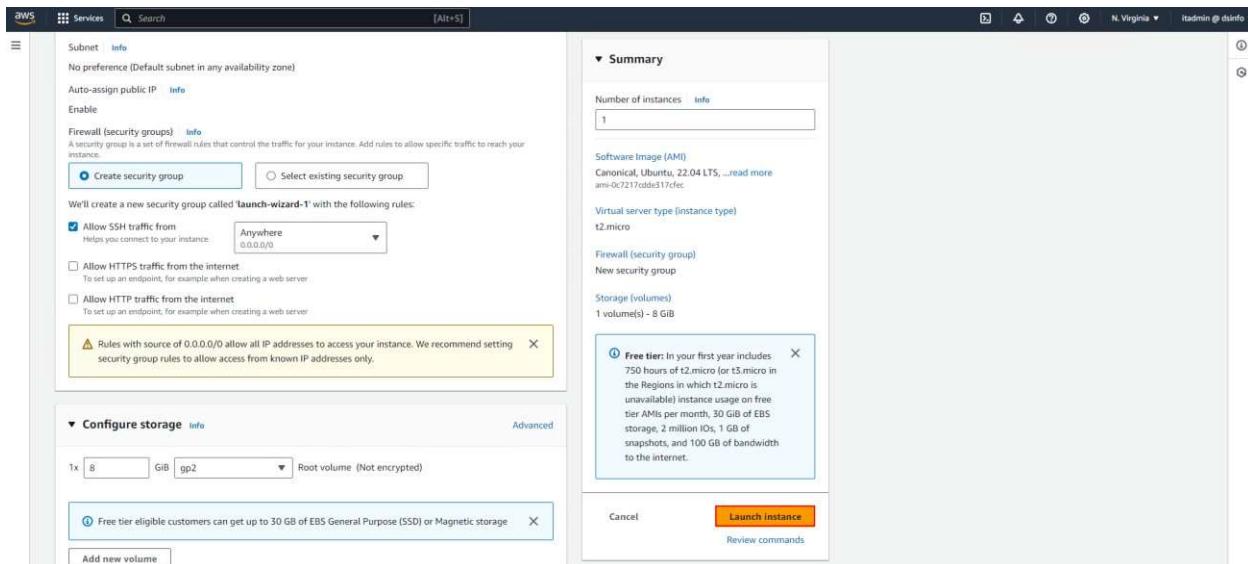
The screenshot shows the 'Create key pair' dialog. The 'Key pair name' field is set to 'key-pair-01'. The 'Key pair type' section has 'RSA' selected. The 'Private key file format' section has 'pem' selected. A note at the bottom of the dialog says: 'When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance.' There are 'Cancel' and 'Create key pair' buttons at the bottom.

Step 5:

In the **Security Group** section, select the **Create Security Group** option.

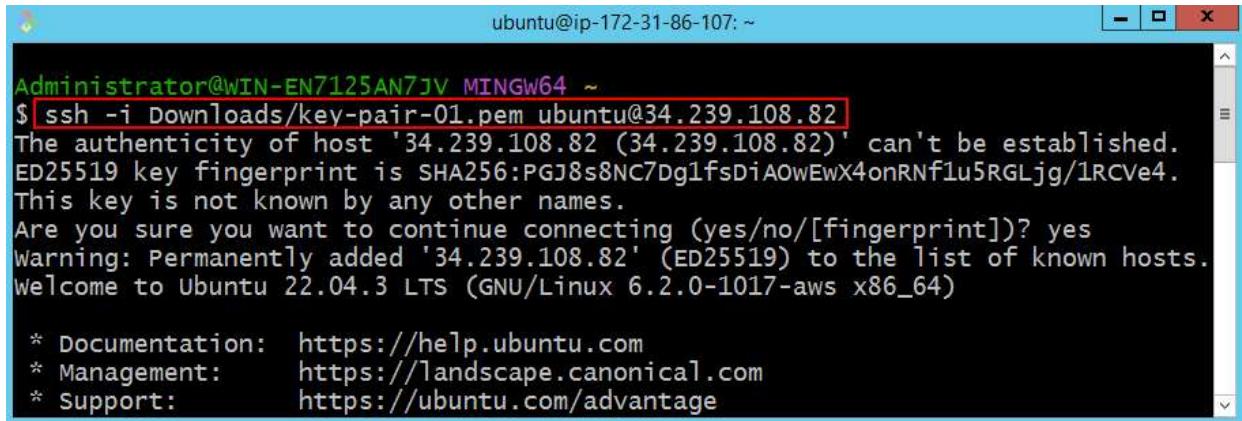


Then, click on the **Launch Instance** button to create the instance.



Step 6:

At this stage, connect to the instance using **SSH** and run the following commands to set up the web server on the instance.



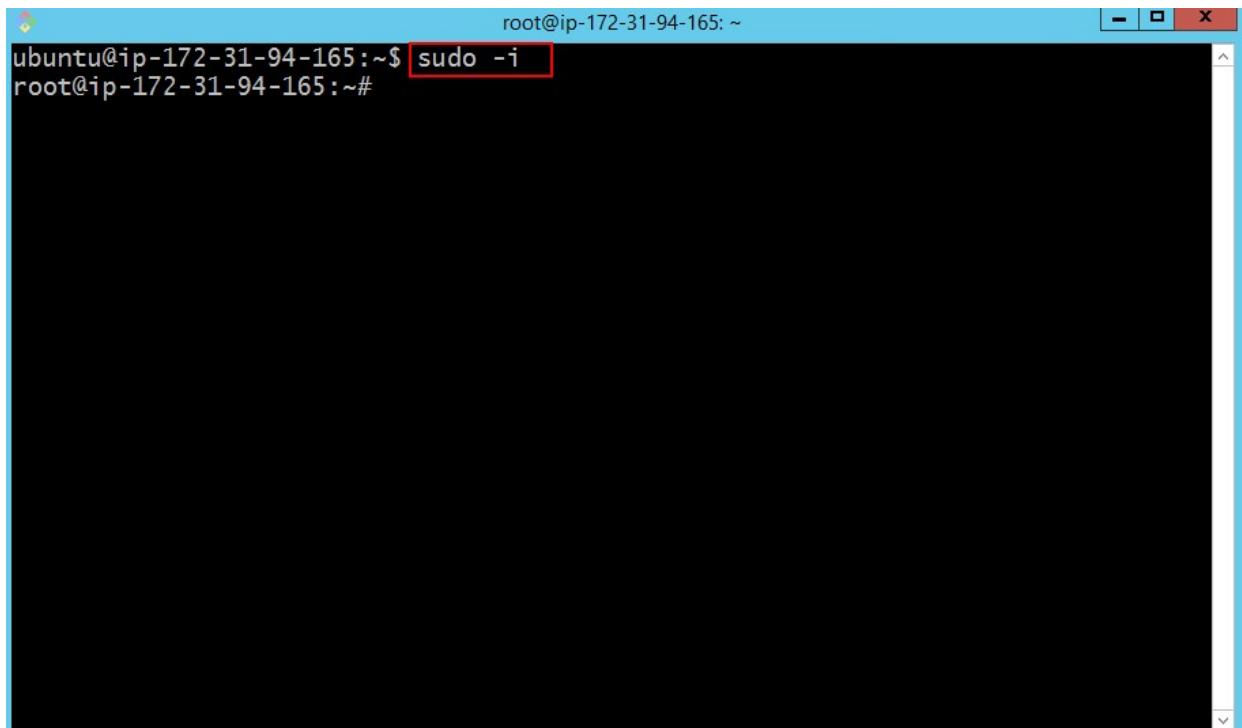
A screenshot of a terminal window titled "ubuntu@ip-172-31-86-107: ~". The window shows the command \$ ssh -i Downloads/key-pair-01.pem ubuntu@34.239.108.82 and its output. The output includes a warning about host key fingerprint, a confirmation prompt, and a welcome message for Ubuntu 22.04.3 LTS. It also lists documentation, management, and support links.

```
Administrator@WIN-EN7125AN7JV MINGW64 ~
$ ssh -i Downloads/key-pair-01.pem ubuntu@34.239.108.82
The authenticity of host '34.239.108.82 (34.239.108.82)' can't be established.
ED25519 key fingerprint is SHA256:PGJ8s8NC7Dg1fsDiAOwEwx4onRNf1u5RGLjg/1RCVe4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
warning: Permanently added '34.239.108.82' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
```

Now, in this section, we want to install the **Tween** website, which is one of the templates from **Tooplate**, on this EC2 instance.

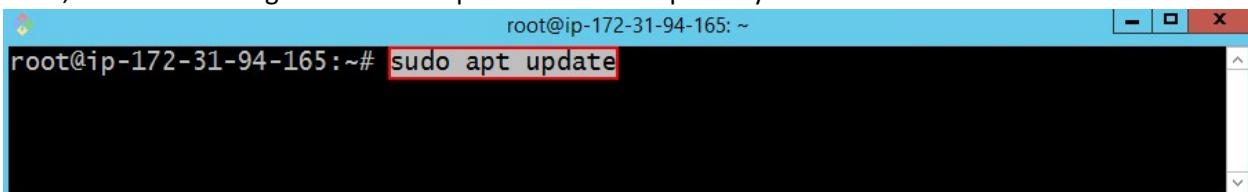
At this stage, use the following command to switch to the root user environment



A screenshot of a terminal window titled "root@ip-172-31-94-165: ~". The window shows the command \$ sudo -i and its output. The output shows the user switching to the root user environment.

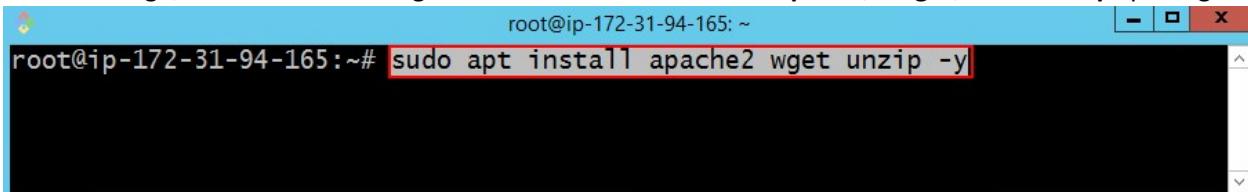
```
root@ip-172-31-94-165:~$ sudo -i
root@ip-172-31-94-165:~#
```

Then, use the following command to update the Linux repository



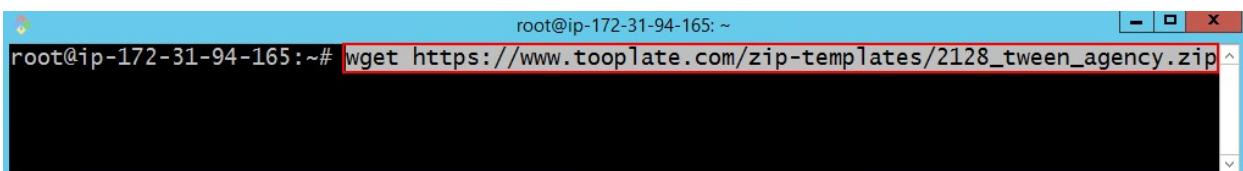
```
root@ip-172-31-94-165:~# sudo apt update
```

At this stage, use the following command to install the **Apache**, **Wget**, and **Unzip** packages



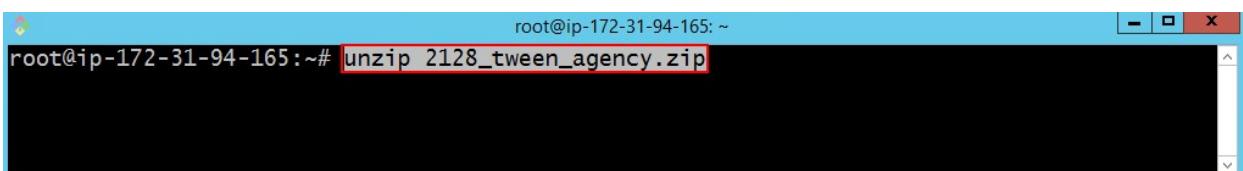
```
root@ip-172-31-94-165:~# sudo apt install apache2 wget unzip -y
```

Next, you need to download the **Tween** file from the **Tooplate** website.



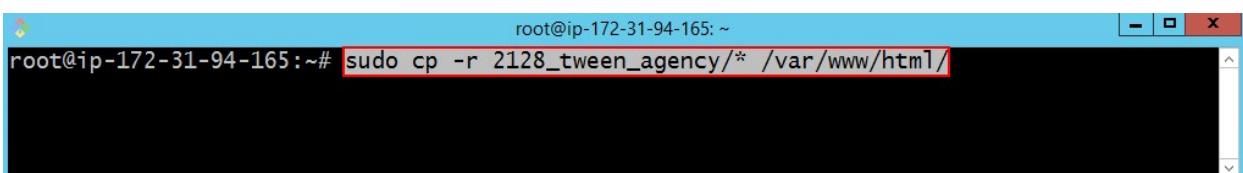
```
root@ip-172-31-94-165:~# wget https://www.tooplate.com/zip-templates/2128_tween_agency.zip
```

After downloading the **Tween** file, you need to unzip it.



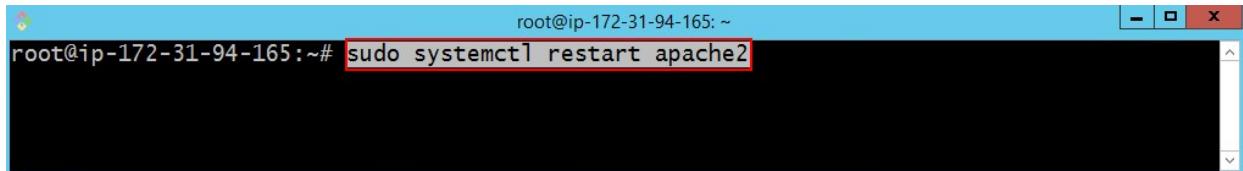
```
root@ip-172-31-94-165:~# unzip 2128_tween_agency.zip
```

At this stage, you need to copy the unzipped contents of the **Tween** website into the **web server directory**.



```
root@ip-172-31-94-165:~# sudo cp -r 2128_tween_agency/* /var/www/html/
```

And finally, you need to restart the **Apache** service.



```
root@ip-172-31-94-165:~# sudo systemctl restart apache2
```

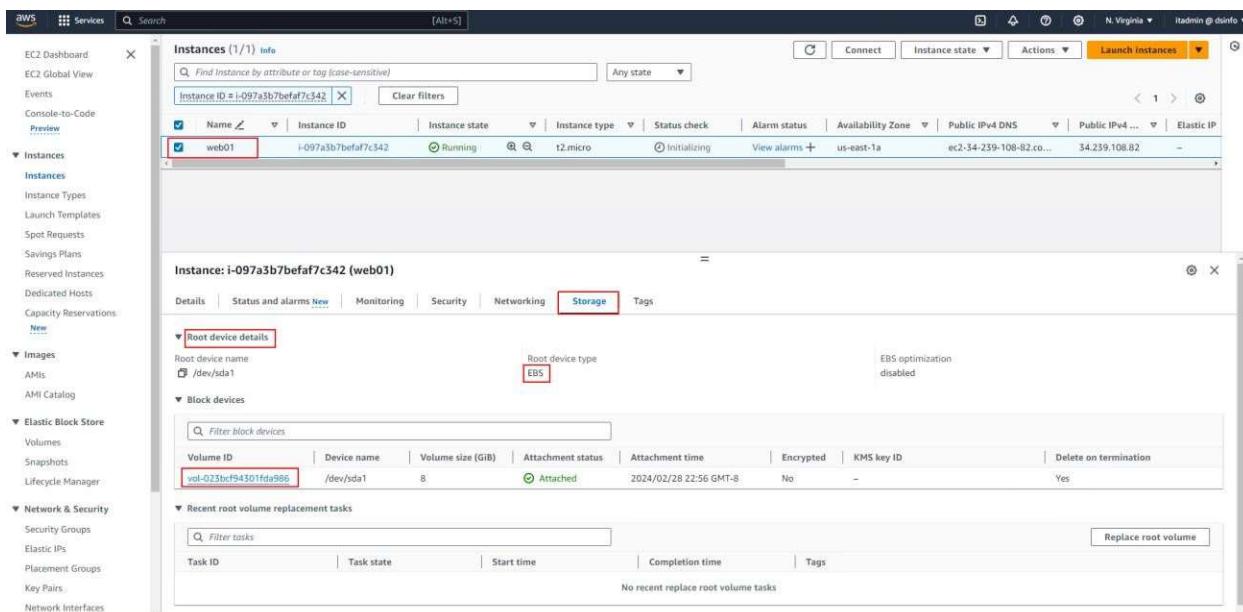
Use the following command to check if the **Tween** website files have been copied to the web server directory



```
root@ip-172-31-94-165:~# ls /var/www/html/
'ABOUT THIS TEMPLATE.txt'    css    fonts    images    index.html    js
root@ip-172-31-94-165:~#
```

Now, we intend to move the **image** folder to a separate **EBS Volume**.

If you click on **Storage** in the **Instance** section, you can view details about the **Root EBS Volume**. By clicking on the **Volume ID**, you will be taken to the **EBS Volume** section.



The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Services, Search, and various AWS services. The main area shows a table of instances. One instance, 'web01', is selected and highlighted with a red box around its name. The 'Storage' tab is active in the details panel for this instance. Under 'Root device details', it shows the root device name as '/dev/sda1' and the root device type as 'EBS'. There's also a note that 'EBS optimization disabled'. Below this, under 'Block devices', there's a table with one row showing a volume ID 'vol-023b5f54301fda986' and a device name '/dev/sda1'. The volume size is 8 GiB, it's attached, and the attachment time is 2024/02/28 22:56 GMT-8. The 'Delete on termination' option is set to 'Yes'. At the bottom of the details panel, there's a section for 'Recent root volume replacement tasks' with a 'Replace root volume' button.

In the **EBS Volume** section, click on the **Name** field and set it to **web01-root-volume**.

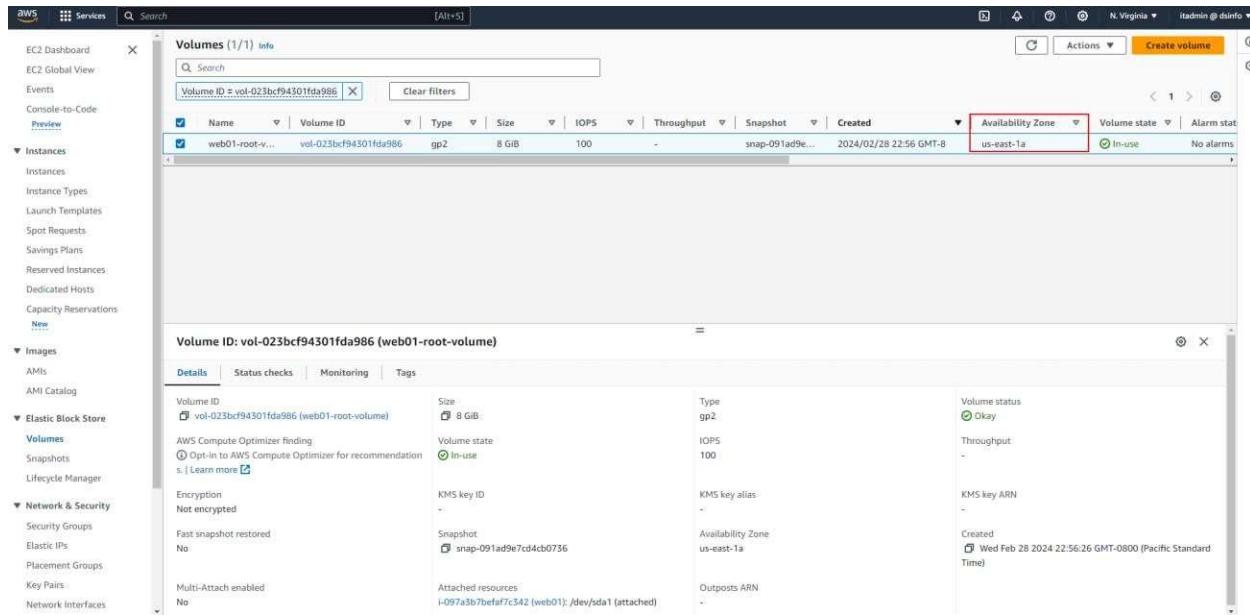
The screenshot shows the AWS EC2 Dashboard with the 'Volumes' section selected. A modal dialog is open over the main table, allowing the user to edit the volume's name. The 'Name' field contains the value 'web01-root-volume', which is highlighted with a red box. Below the field are 'Cancel' and 'Save' buttons. The main table displays one volume entry:

Name	Type	Size	IOPS	Throughput	Snapshot	Created	Availability Zone	Volume state	Alarm stat
vol-023bcf94301fda986	gp2	8 GiB	100	-	snap-091ad9e7cd4cb0756	2024/02/28 22:56 GMT-8	us-east-1a	In-use	No alarms

Below the table, a detailed view of the selected volume (Volume ID: vol-023bcf94301fda986) is shown. The 'Details' tab is selected, displaying the following information:

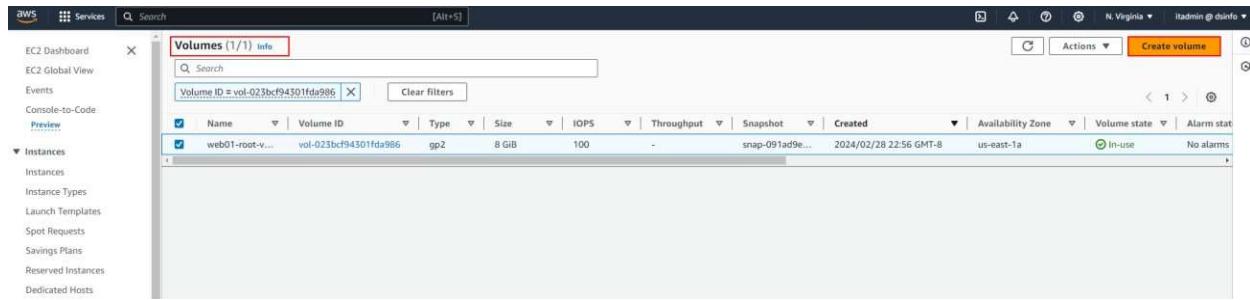
Volume ID	Size	Type	Volume status
vol-023bcf94301fda986	8 GiB	gp2	Okay
AWS Compute Optimizer finding			
Opt-in to AWS Compute Optimizer for recommendation s. Learn more			
Encryption	KMS key ID	KMS key alias	KMS key ARN
Not encrypted	-	-	-
Fast snapshot restored	Snapshot	Availability Zone	Created
No	snap-091ad9e7cd4cb0756	us-east-1a	Wed Feb 28 2024 22:56:26 GMT-0800 (Pacific Standard Time)
Multi-Attach enabled	Attached resources	Outposts ARN	
No	i-097a3b7befaf7c542 (web01): /dev/sda1 (attached)	-	

In the **Availability Zone** section, as you can see, the zone of the **Volume** and the **Instance** is the same — in other words, they must be in the **same zone**.



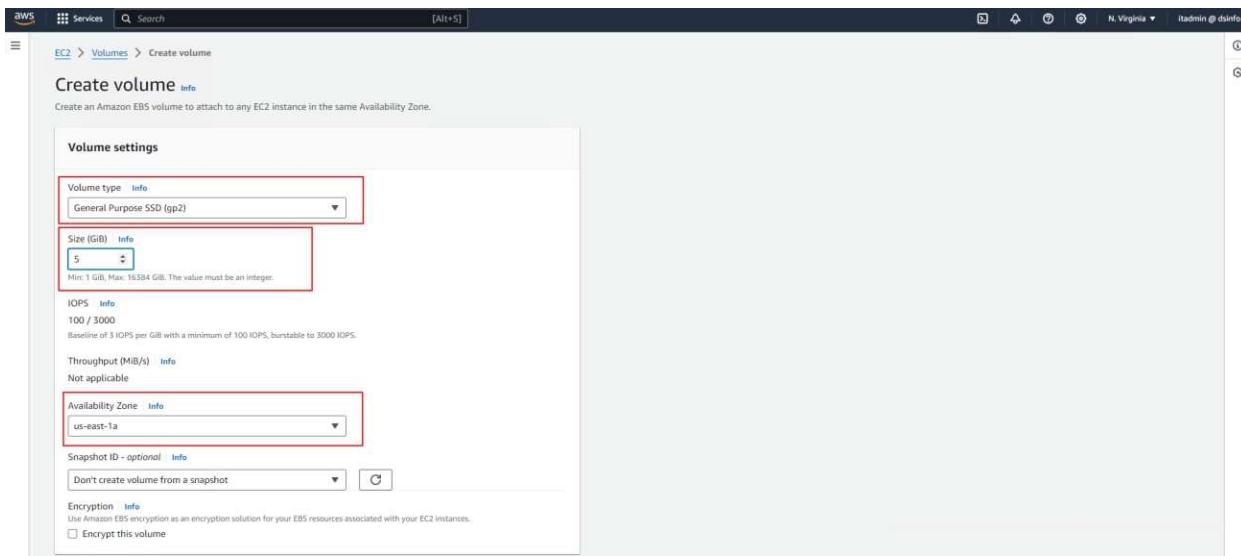
The screenshot shows the AWS EC2 Volumes page. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Images, and Elastic Block Store. Under Elastic Block Store, there are links for Volumes, Snapshots, and Lifecycle Manager. Under Network & Security, there are links for Security Groups, Elastic IPs, Placement Groups, Key Pairs, and Network Interfaces. The main content area displays a table titled "Volumes (1/1) Info". The table has columns: Name, Volume ID, Type, Size, IOPS, Throughput, Snapshot, Created, Availability Zone, Volume state, and Alarm stat. A single row is selected, showing "web01-root-v..." as the Name, "vol-023bcf94301fda986" as the Volume ID, "gp2" as the Type, "8 GiB" as the Size, "100" as the IOPS, "-" as the Throughput, "snap-091ad9e..." as the Snapshot, "2024/02/28 22:56 GMT-8" as the Created date, "us-east-1a" as the Availability Zone, "In-use" as the Volume state, and "No alarms" as the Alarm stat. A red box highlights the "Availability Zone" column header and the "us-east-1a" value.

To create a volume, click on the **Create Volume** button.

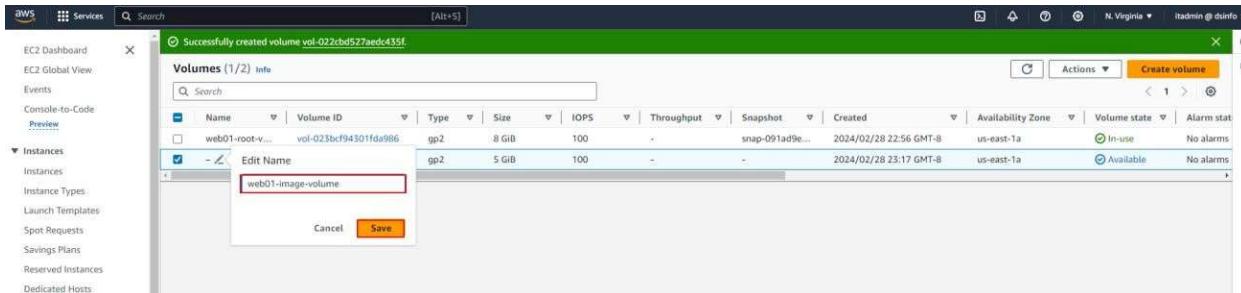


This screenshot is identical to the one above, showing the AWS EC2 Volumes page with a single volume listed in the "us-east-1a" availability zone. The interface and data are the same, with the "Create Volume" button visible at the top right of the main content area.

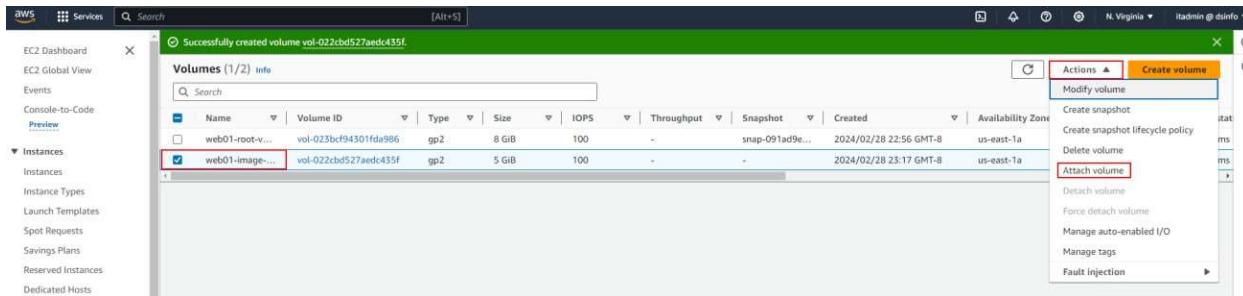
At this stage, select **General Purpose SSD (gp2)** from the **Volume Type** section, set the **Size** to **5GB**, and make sure to choose the same **Availability Zone** as your instance. Finally, click the **Create Volume** button.



At this stage, assign a **name** to the EBS Volume.



To attach the **EBS Volume** to the **EC2 Instance**, click on the **Actions** button and then select **Attach Volume**.

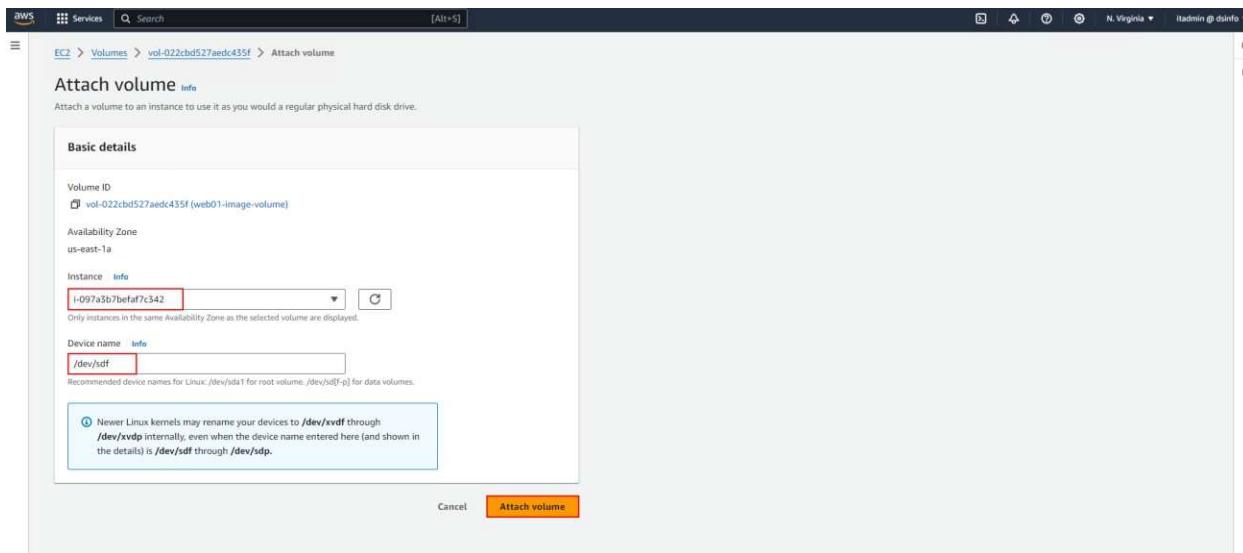


The screenshot shows the AWS EC2 Dashboard with the 'Volumes' section selected. A success message at the top says 'Successfully created volume vol-022cbd527aedc435f'. Below is a table of volumes:

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created	Availability Zone
web01-root-v...	vol-023bcf94301fda986	gp2	8 GiB	100	-	snap-091ad9e...	2024/02/28 22:56 GMT-8	us-east-1a
<input checked="" type="checkbox"/> web01-image-v...	vol-022cbd527aedc435f	gp2	5 GiB	100	-	-	2024/02/28 23:17 GMT-8	us-east-1a

The 'Actions' menu on the right is open, showing options like 'Modify volume', 'Create snapshot', and 'Attach volume'. The 'Attach volume' option is highlighted.

Then, at this stage, select the **Instance name** and click the **Attach Volume** button.



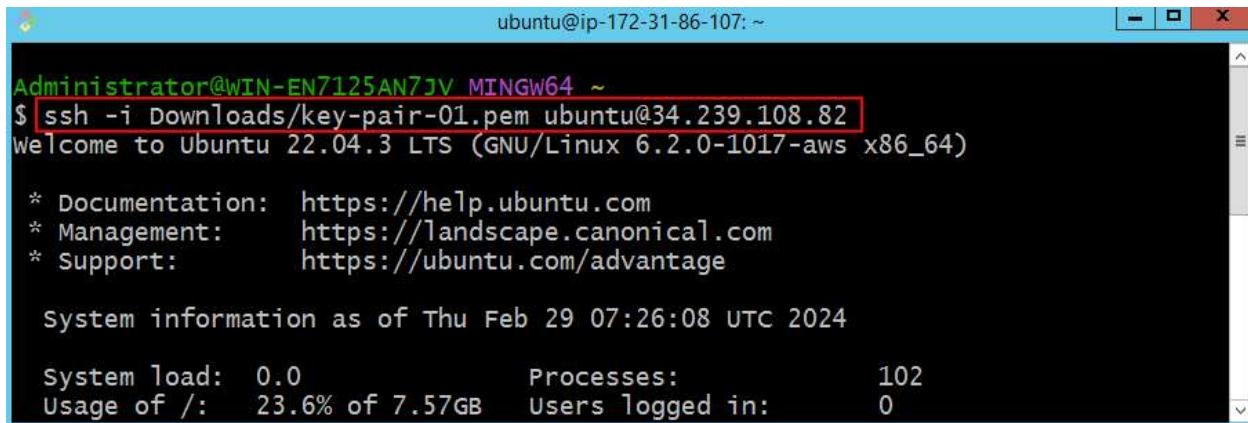
The screenshot shows the 'Attach volume' dialog box. At the top, it says 'Attach a volume to an instance to use it as you would a regular physical hard disk drive.' Below are fields for 'Basic details':

- Volume ID: vol-022cbd527aedc435f (web01-image-volume)
- Availability Zone: us-east-1a
- Instance:
- Device name:

A note below the device name field says: 'Recommended device names for Linux: /dev/sda1 for root volume, /dev/sdf[5-p] for data volumes.'

At the bottom of the dialog are 'Cancel' and 'Attach volume' buttons. A callout box points to the 'Attach volume' button with the text: 'Never Linux kernels may rename your devices to /dev/vdff through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.'

After attaching the **Volume** to your instance, you can log in to your instance.



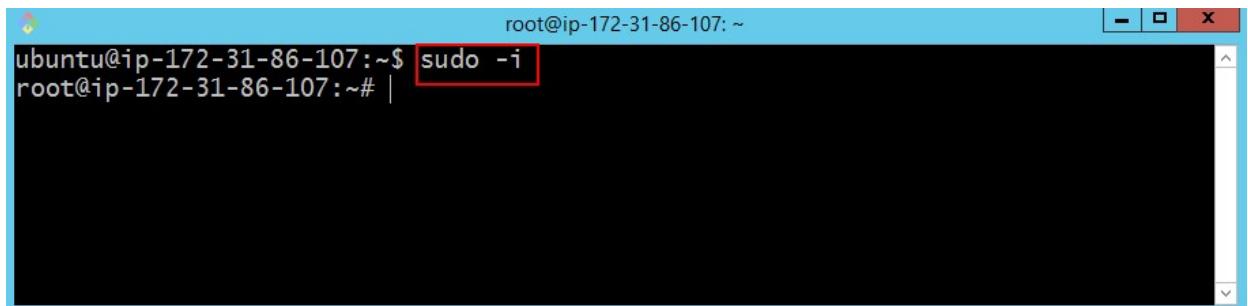
```
ubuntu@ip-172-31-86-107: ~
Administrator@WIN-EN7125AN7JV MINGW64 ~
$ ssh -i Downloads/key-pair-01.pem ubuntu@34.239.108.82
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Thu Feb 29 07:26:08 UTC 2024

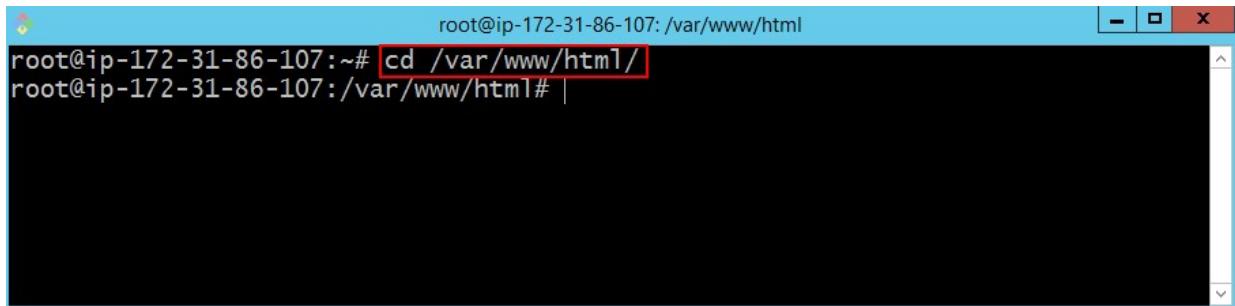
 System load:  0.0          Processes:           102
 Usage of /:   23.6% of 7.57GB  Users logged in:    0
```

After logging in, use the following command to switch to the **Root User**



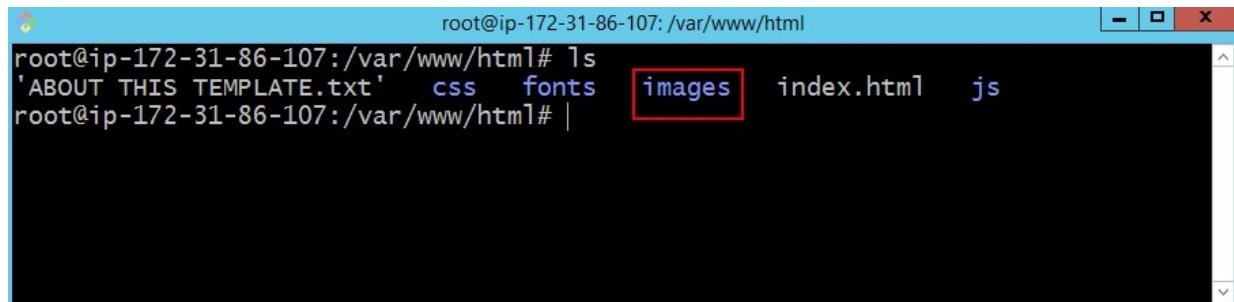
```
root@ip-172-31-86-107: ~
ubuntu@ip-172-31-86-107:~$ sudo -i
root@ip-172-31-86-107:~# |
```

Enter the **web server directory**



```
root@ip-172-31-86-107:~/var/www/html
root@ip-172-31-86-107:~/var/www/html# cd /var/www/html/
root@ip-172-31-86-107:/var/www/html# |
```

We intend to move the **image** folder to a **different storage (EBS volume)**.



```
root@ip-172-31-86-107:/var/www/html# ls  
'ABOUT THIS TEMPLATE.txt'  css  fonts  images  index.html  js  
root@ip-172-31-86-107:/var/www/html# |
```

First, use the following command to view the list of available partitions

```
root@ip-172-31-86-107:/var/www/html# fdisk -l
Disk /dev/loop0: 24.9 MiB, 26112000 bytes, 51000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop1: 55.66 MiB, 58363904 bytes, 113992 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop2: 63.46 MiB, 66547712 bytes, 129976 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop3: 111.95 MiB, 117387264 bytes, 229272 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop4: 40.86 MiB, 42840064 bytes, 83672 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

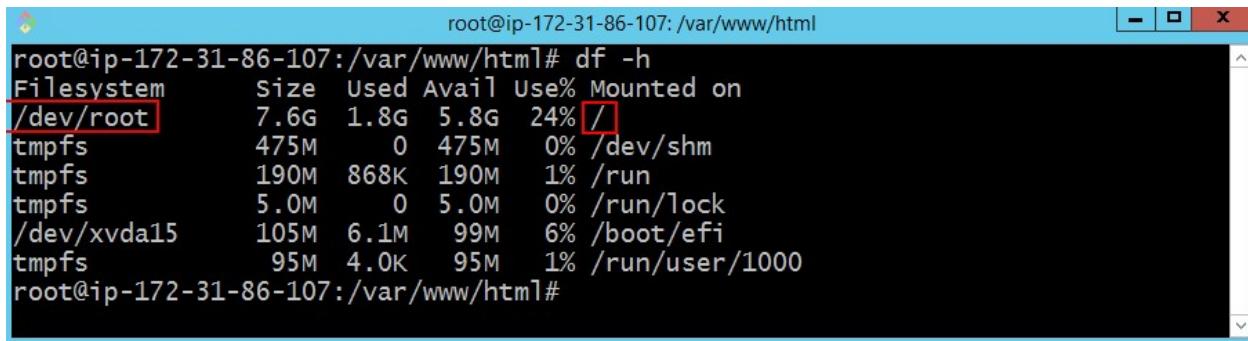
Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 8F814D15-0F5D-40E2-8A1E-E14BBDFCE594

Device      Start    End  Sectors  Size Type
/dev/xvda1   227328 16777182 16549855  7.9G Linux filesystem
/dev/xvda14   2048    10239     8192    4M BIOS boot
/dev/xvda15  10240   227327  217088 106M EFI System

Partition table entries are not in disk order.

Disk /dev/xvdf: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
```

Use the following command to view the mounted partitions



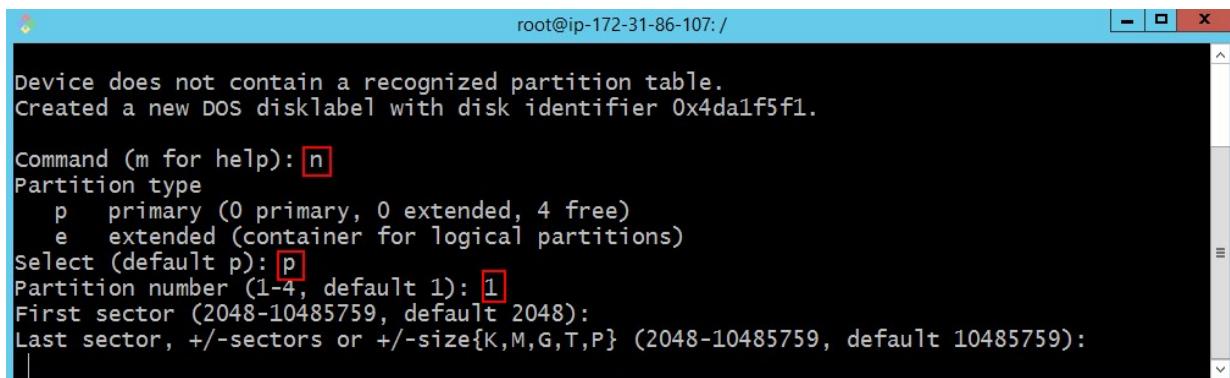
```
root@ip-172-31-86-107:/var/www/html# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       7.6G  1.8G  5.8G  24% /
tmpfs          475M    0  475M   0% /dev/shm
tmpfs          190M  868K  190M   1% /run
tmpfs          5.0M    0  5.0M   0% /run/lock
/dev/xvda15     105M  6.1M   99M   6% /boot/efi
tmpfs          95M  4.0K   95M   1% /run/user/1000
root@ip-172-31-86-107:/var/www/html#
```

At this stage, you need to partition the new volume using the following command



```
root@ip-172-31-86-107:#
root@ip-172-31-86-107:/# fdisk /dev/xvdf
```

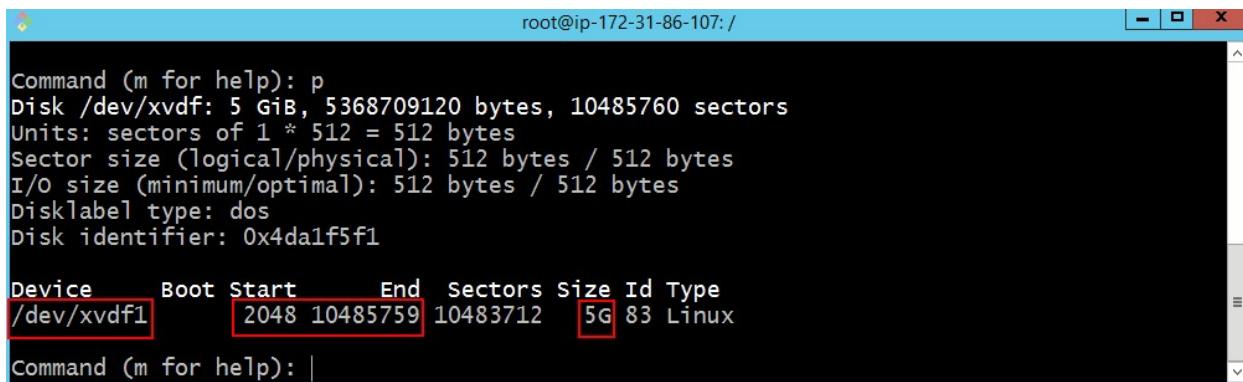
Next, press the **m** key to view the **fdisk help menu**. To create a new partition, press **n**, then **p** for Primary Partition, set the partition number to **1**, and press **Enter**.



```
root@ip-172-31-86-107:/
Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x4dal5f1.

Command (m for help): n
Partition type
  p  primary (0 primary, 0 extended, 4 free)
  e  extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-10485759, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-10485759, default 10485759):
|
```

To view the created partition, simply press the **p** key, which stands for **print**.



```
root@ip-172-31-86-107: /  
Command (m for help): p  
Disk /dev/xvdf: 5 GiB, 5368709120 bytes, 10485760 sectors  
Units: sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disklabel type: dos  
Disk identifier: 0x4da1f5f1  
  
Device      Boot Start      End  Sectors Size Id Type  
/dev/xvdf1          2048 10485759 10483712   5G 83 Linux  
  
Command (m for help): |
```

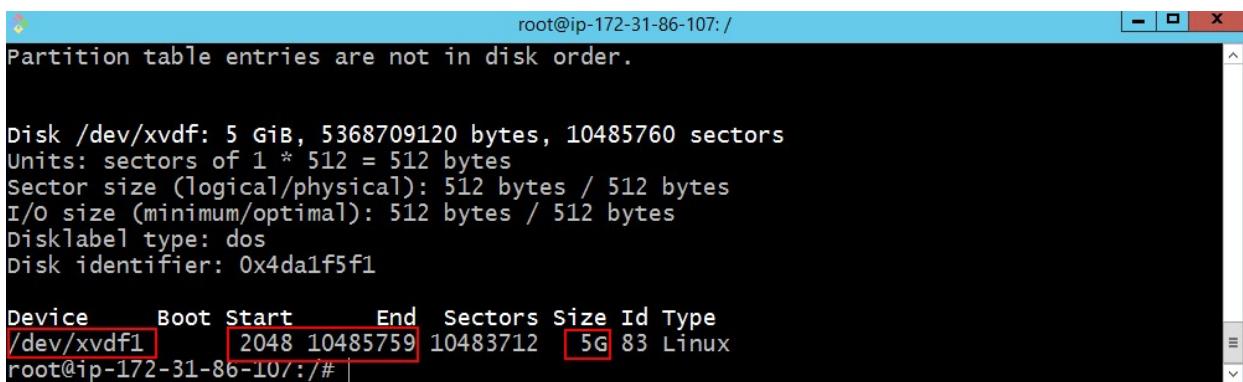
To **write** the changes, press the **w** key.



```
root@ip-172-31-86-107: /  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disklabel type: dos  
Disk identifier: 0x4da1f5f1  
  
Device      Boot Start      End  Sectors Size Id Type  
/dev/xvdf1          2048 10485759 10483712   5G 83 Linux  
  
Command (m for help): w  
The partition table has been altered.  
Calling ioctl() to re-read partition table.  
Syncing disks.  
root@ip-172-31-86-107: #
```

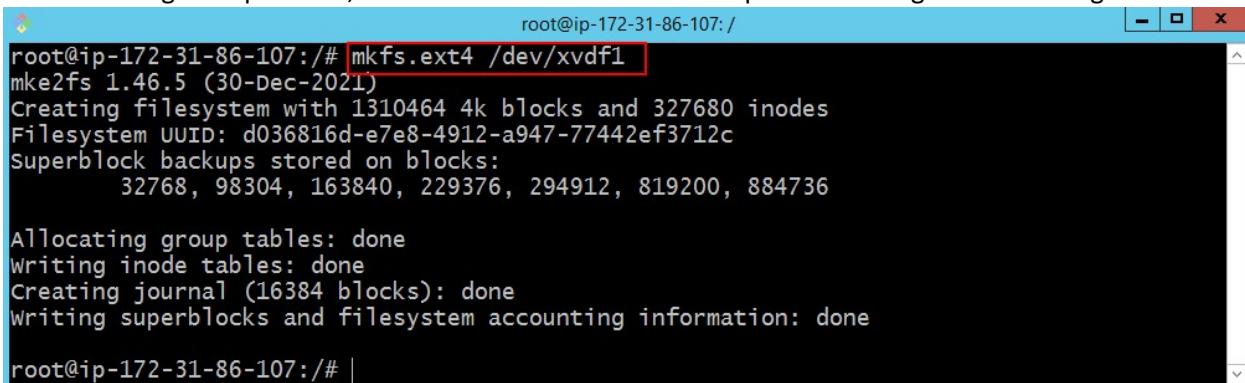
To view the created partition, you can use the following command:

fdisk -l



```
root@ip-172-31-86-107: /  
Partition table entries are not in disk order.  
  
Disk /dev/xvdf: 5 GiB, 5368709120 bytes, 10485760 sectors  
Units: sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disklabel type: dos  
Disk identifier: 0x4da1f5f1  
  
Device      Boot Start      End  Sectors Size Id Type  
/dev/xvdf1          2048 10485759 10483712   5G 83 Linux  
root@ip-172-31-86-107: # |
```

After creating the partition, it's time to format the new partition using the following command

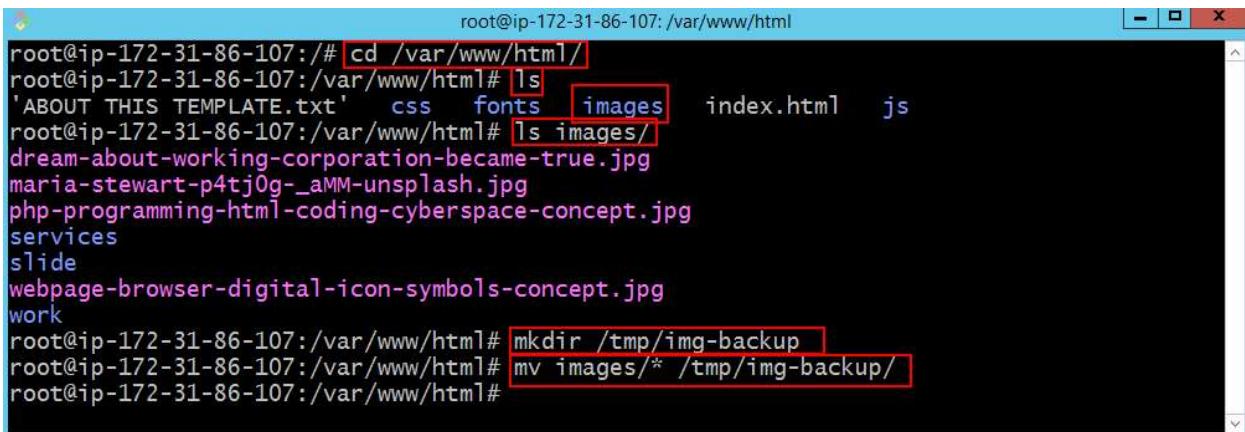


```
root@ip-172-31-86-107:/# mkfs.ext4 /dev/xvdf1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 1310464 4k blocks and 327680 inodes
Filesystem UUID: d036816d-e7e8-4912-a947-77442ef3712c
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

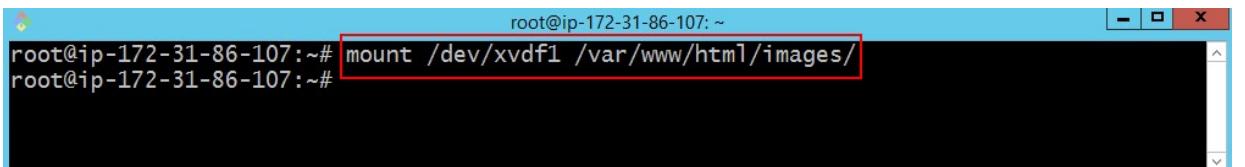
root@ip-172-31-86-107:/# |
```

Now, to mount the **image** folder, go to the **web server directory** and review the contents of the **image** folder. Then, create a folder in the **/tmp** directory and back up the contents of the **image** folder to avoid data loss during the mount process.



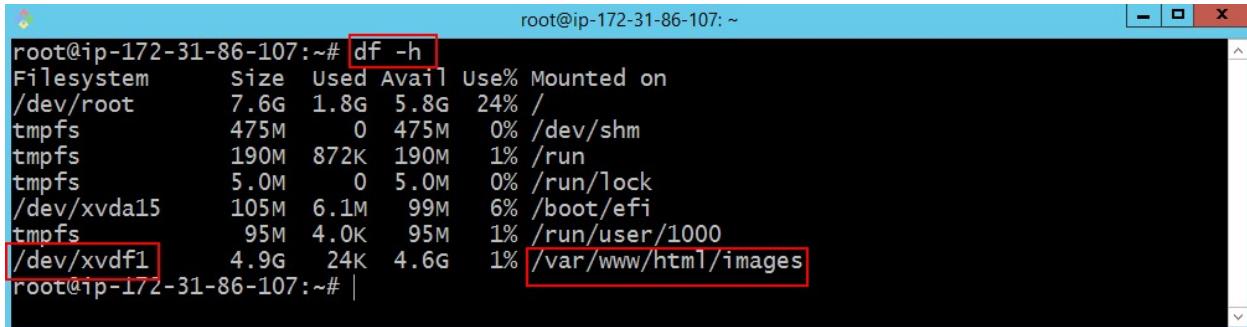
```
root@ip-172-31-86-107:/# cd /var/www/html/
root@ip-172-31-86-107:/var/www/html# ls
'ABOUT THIS TEMPLATE.txt'  css  fonts  images  index.html  js
root@ip-172-31-86-107:/var/www/html# ls images/
dream-about-working-corporation-became-true.jpg
maria-stewart-p4tj0g-_aMM-unsplash.jpg
php-programming-html-coding-cyberspace-concept.jpg
services
slide
webpage-browser-digital-icon-symbols-concept.jpg
work
root@ip-172-31-86-107:/var/www/html# mkdir /tmp/img-backup
root@ip-172-31-86-107:/var/www/html# mv images/* /tmp/img-backup/
root@ip-172-31-86-107:/var/www/html#
```

Now, at this stage, use the following command to mount the new partition to the **images** folder in the web server directory



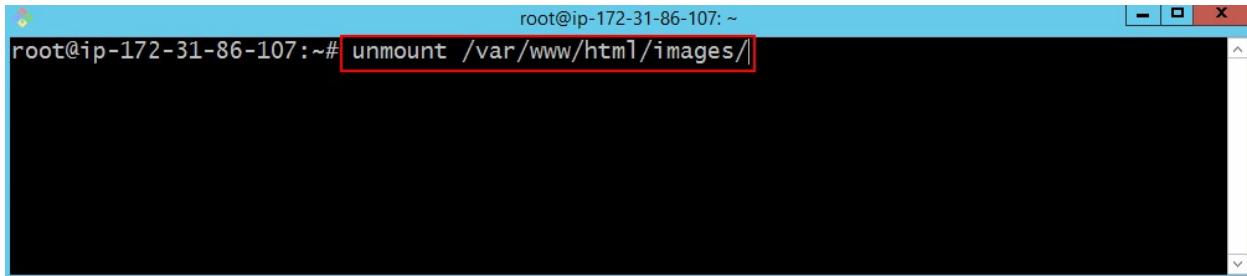
```
root@ip-172-31-86-107:~# mount /dev/xvdf1 /var/www/html/images/
root@ip-172-31-86-107:~#
```

To check whether the mount was successful, use the following command



```
root@ip-172-31-86-107:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       7.6G  1.8G  5.8G  24% /
tmpfs          475M    0  475M   0% /dev/shm
tmpfs          190M  872K  190M   1% /run
tmpfs          5.0M    0  5.0M   0% /run/lock
/dev/xvda15     105M  6.1M  99M   6% /boot/efi
tmpfs          95M  4.0K  95M   1% /run/user/1000
/dev/xvdf1      4.9G  24K  4.6G  1% /var/www/html/images/
root@ip-172-31-86-107:~# |
```

If you want to unmount the volume, you can use the following command



```
root@ip-172-31-86-107:~# umount /var/www/html/images/|
```

One important point to note is that if the instance is stopped and started, or in other words **restarted**, the mounted partition will be **unmounted**.

For **permanent mount**, the mount configuration must be added to the **fstab** file.

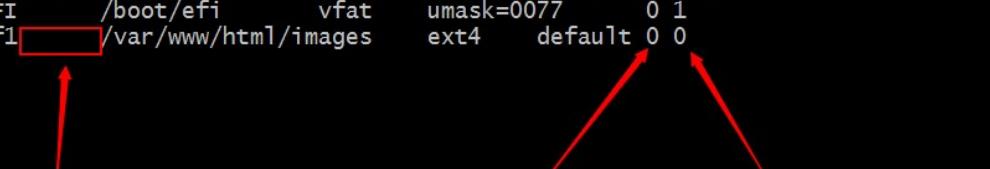
So, open the **fstab** file using the **nano** editor



```
root@ip-172-31-86-107:~# nano /etc/fstab|
```

Now, inside the file, enter the mount information as shown below

```
root@ip-172-31-86-107: ~
GNU nano 6.2
LABEL=cloudimg-rootfs / ext4 discard,errors=remount-ro 0 1
LABEL=UEFI /boot/efi vfat umask=0077 0 1
/dev/xvdf1 /var/www/html/images ext4 default 0 0


```

Then, **save** the file.

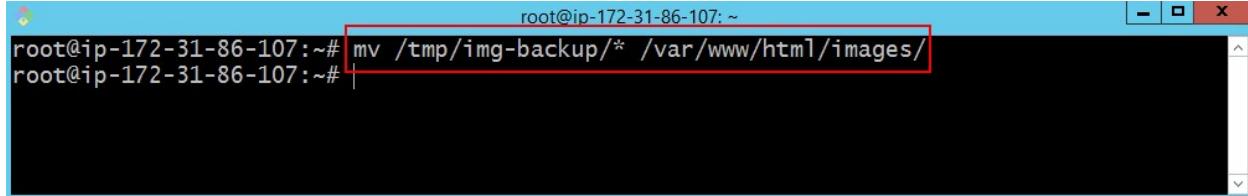
Use the following command to mount all entries defined in the **fstab** file:

```
root@ip-172-31-86-107:~# mount -a  
root@ip-172-31-86-107:~#
```

Again, use the following command to ensure that the mount has been done correctly.

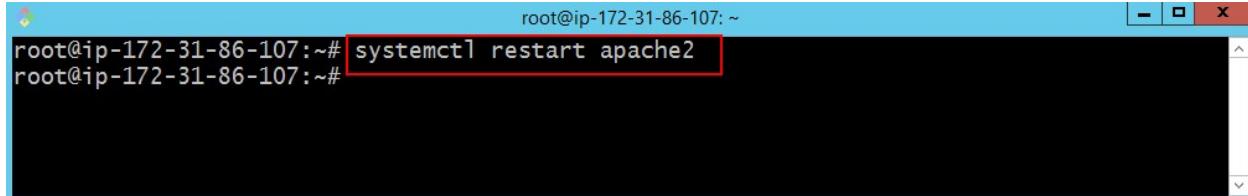
```
root@ip-172-31-86-107:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       7.6G  1.8G  5.8G  24% /
tmpfs           475M    0  475M   0% /dev/shm
tmpfs           190M  872K  190M   1% /run
tmpfs            5.0M    0  5.0M   0% /run/lock
/dev/xvda15     105M  6.1M  99M   6% /boot/efi
tmpfs            95M  4.0K  95M   1% /run/user/1000
/dev/xvdf1      4.9G  24K  4.6G  1% /var/www/html/images
root@ip-172-31-86-107:~# |
```

At this stage, move the files you backed up in the **/tmp** directory back to the web server's **images** folder, which is now located on the new volume. Use the following command



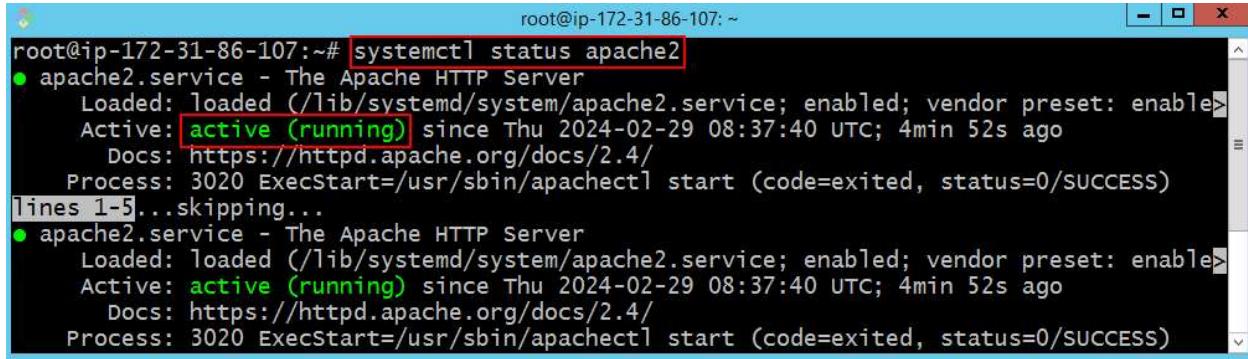
```
root@ip-172-31-86-107:~# mv /tmp/img-backup/* /var/www/html/images/
root@ip-172-31-86-107:~#
```

At this stage, you need to restart the **Apache** service



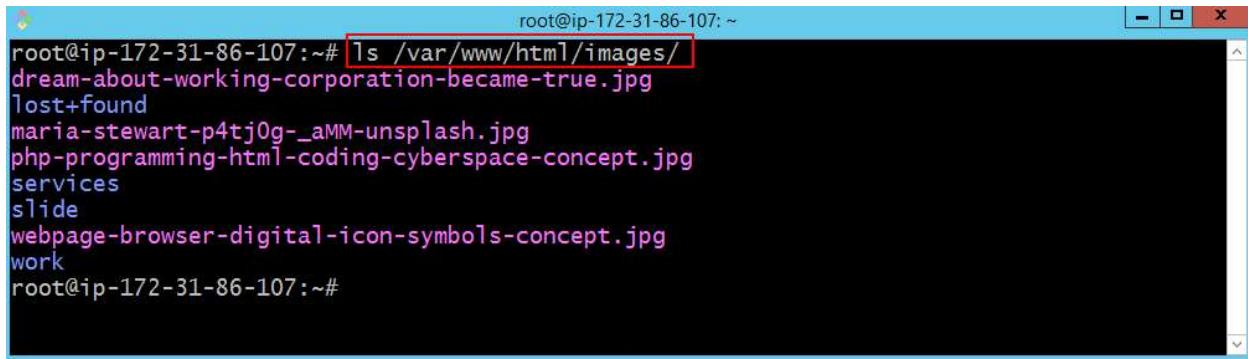
```
root@ip-172-31-86-107:~# systemctl restart apache2
root@ip-172-31-86-107:~#
```

At this stage, before testing, check the status of the **Apache** service



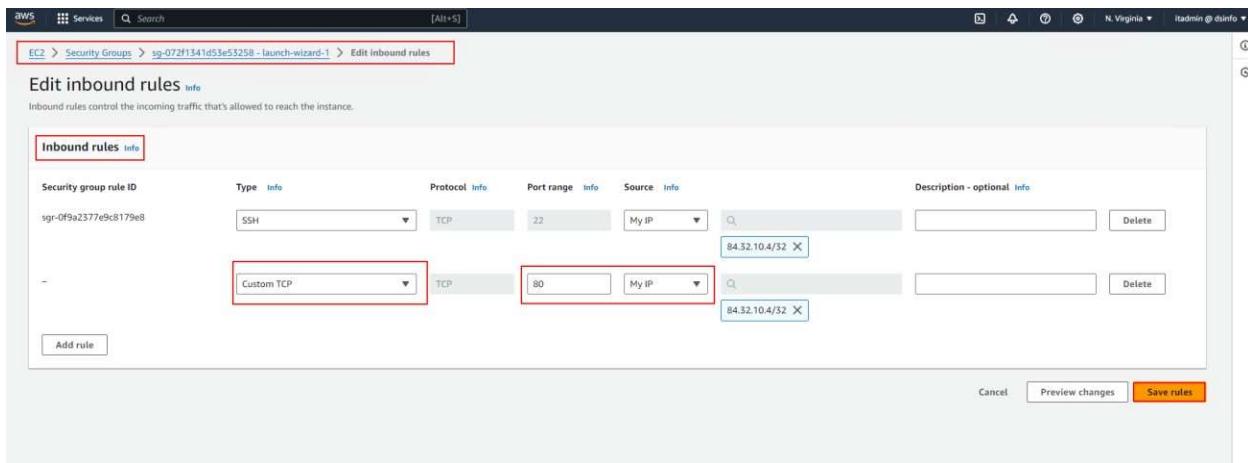
```
root@ip-172-31-86-107:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2024-02-29 08:37:40 UTC; 4min 52s ago
    Docs: https://httpd.apache.org/docs/2.4/
   Process: 3020 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
lines 1-5...skipping...
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2024-02-29 08:37:40 UTC; 4min 52s ago
    Docs: https://httpd.apache.org/docs/2.4/
   Process: 3020 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
```

As you can see, the files are now located in the **images** folder on the web server path, and the **images** folder is currently mounted on the **new partition**.

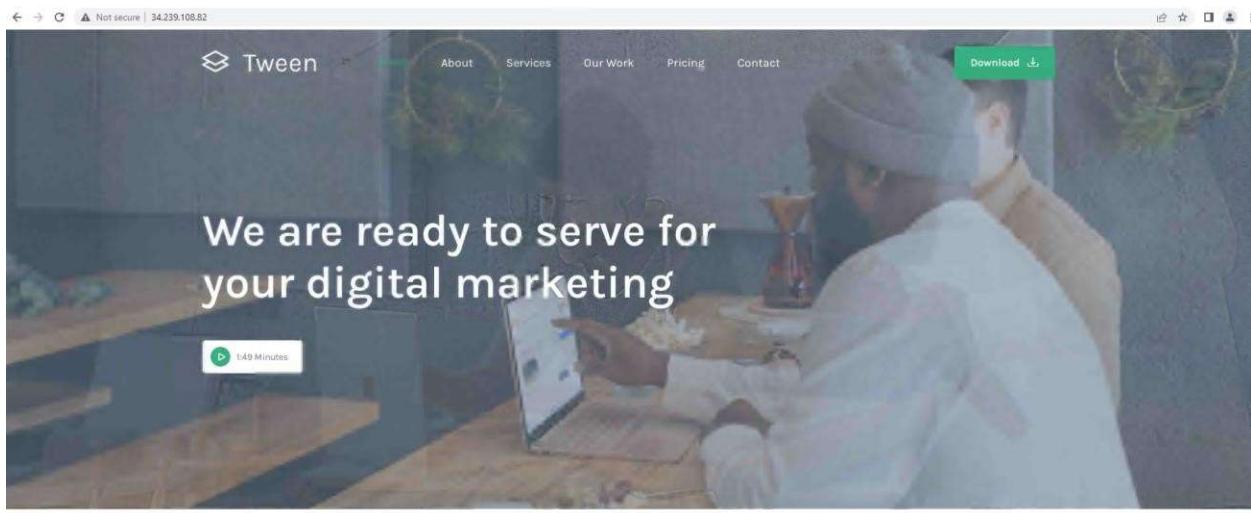


```
root@ip-172-31-86-107:~# ls /var/www/html/images/
dream-about-working-corporation-became-true.jpg
lost+found
maria-stewart-p4tj0g-_aMM-unsplash.jpg
php-programming-html-coding-cyberspace-concept.jpg
services
slide
webpage-browser-digital-icon-symbols-concept.jpg
work
root@ip-172-31-86-107:~#
```

To access the website hosted on the EC2 instance from the internet, you need to add a rule to allow **port 80** in the **Inbound Rules** of your Security Group, as shown in the image below.



For the final test, simply enter the **Public IP Address** of the web server in your browser and check whether all the images in the **images** folder (now on the new partition) are being loaded correctly from the web server path.



Digital Happiness

[Introduction](#) [Profile](#) [FAQs](#)

As shown in the image above, the **mount operation** was completed successfully, and all the images in the **images** folder have been loaded on the website.

Introduction to EBS Snapshots

The **EBS Snapshots** feature is used for **backing up** a volume and **restoring data** in case of a failure.

When a failure occurs and you need to restore data, follow these steps:

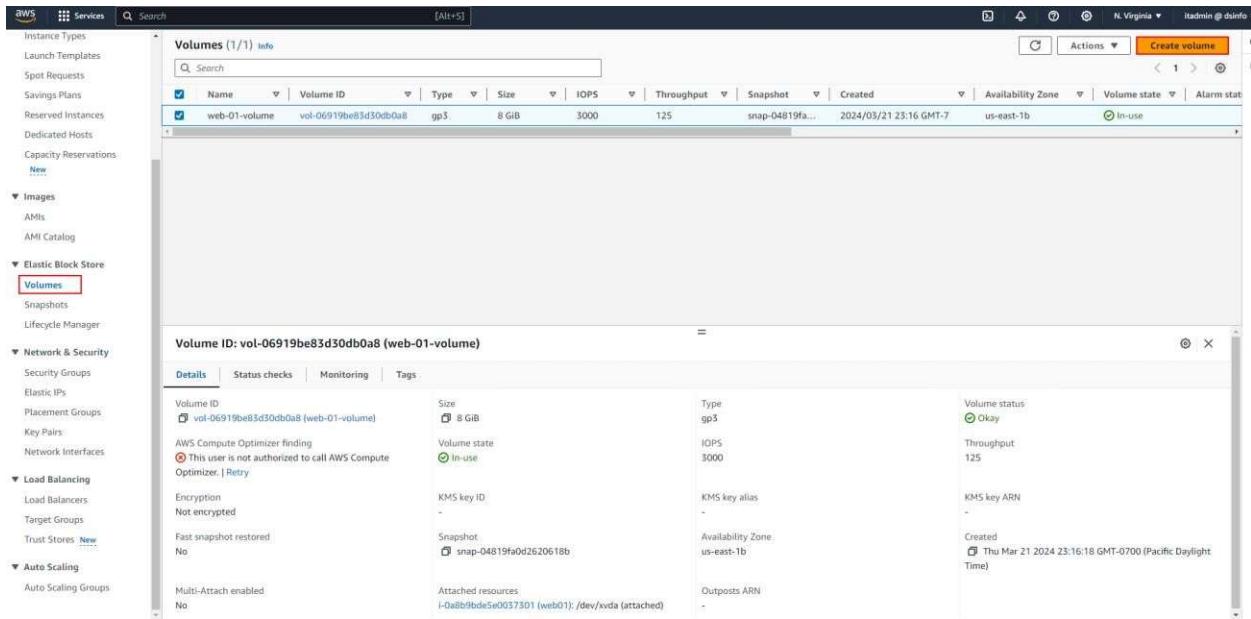
- **Unmount the Volume.**
- **Detach the Volume.**
- **Create a new volume from the snapshot.**
- **Attach the new volume** to the instance.
- **Mount the new volume.**

Therefore, the first thing that should be done is to take a **snapshot** of the volume.

In this section, we will create a new **volume**, attach it to the **EC2 instance**, install a **database**, store the database files on the new volume, and finally, take a **snapshot** of the new volume.

How to Create a New Volume and Attach It to an EC2 Instance

At this stage, click on the **Volumes** section, and then click the **Create Volume** button.



The screenshot shows the AWS Management Console interface for the Elastic Block Store (EBS) service. On the left, there is a navigation sidebar with various options like Instance Types, Launch Templates, and Capacity Reservations. Under the 'Elastic Block Store' section, the 'Volumes' option is selected and highlighted with a red box. The main content area displays a table titled 'Volumes (1/1) Info'. The table has columns for Name, Volume ID, Type, Size, IOPS, Throughput, Snapshot, Created, Availability Zone, Volume state, and Alarm status. A single volume named 'web-01-volume' is listed with its details: Volume ID is 'vol-06919be83d30db0a8', Type is 'gp3', Size is '8 GiB', IOPS is '3000', Throughput is '125', Snapshot is 'snap-04819fa...', Created is '2024/03/21 23:16 GMT-7', Availability Zone is 'us-east-1b', Volume state is 'In-use', and Alarm status is 'OK'. At the top right of the table, there is a 'Create volume' button. Below the table, there is a detailed view for the selected volume, showing its ID as 'vol-06919be83d30db0a8 (web-01-volume)'. The 'Details' tab is selected, displaying various configuration parameters such as Volume ID, Size, Type, Volume status, Encryption, KMS key ID, KMS key alias, KMS key ARN, AWS Compute Optimizer finding, Volume state, Throughput, Availability Zone, Snapshot, Multi-Attach enabled, Attached resources, Outposts ARN, and Created date.

At this stage, specify the **Volume Size** and **Availability Zone**. Make sure to select the **Availability Zone** that matches your EC2 instance.

The screenshot shows the 'Create volume' configuration page. Key fields include:

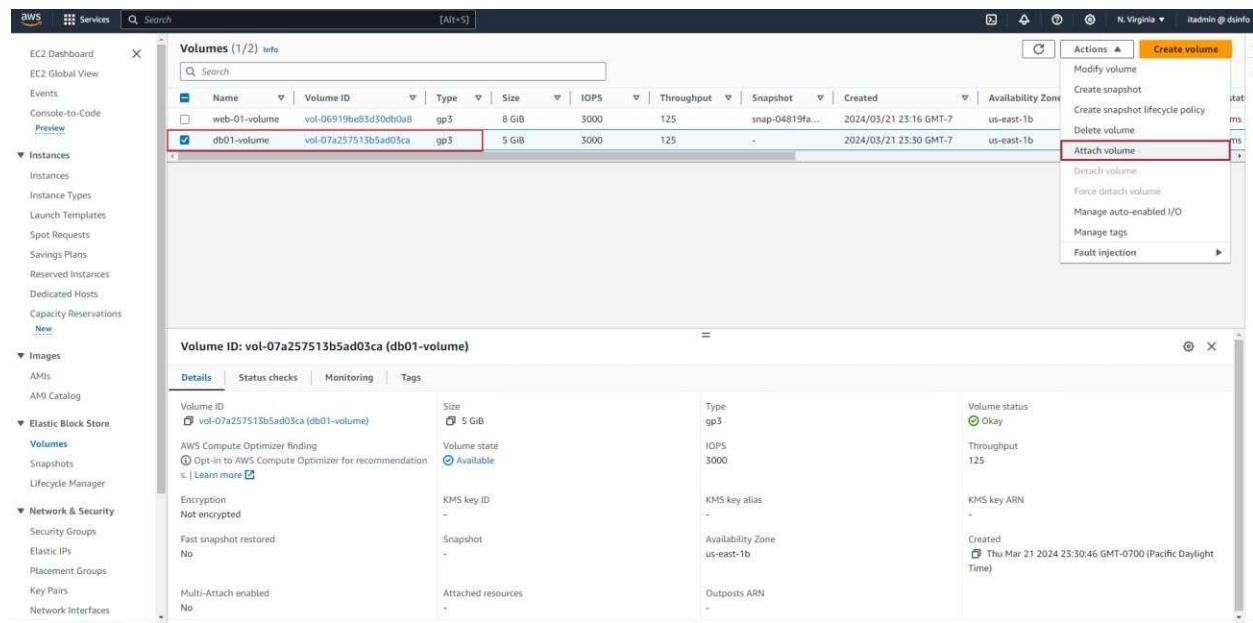
- Volume type:** General Purpose SSD (gp3)
- Size (GiB):** 5
- IOPS:** 3000
- Throughput (MiB/s):** 125
- Availability Zone:** us-east-1b (highlighted with a red box)
- Snapshot ID - optional:** Don't create volume from a snapshot
- Encryption:** Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances. (checkbox unchecked)

At this stage, specify a **Tag** for the volume, and then click on the **Create Volume** button to create the new volume.

The screenshot shows the 'Create volume' configuration page with a tag added:

- Tags - optional:** A tag named "db01-volume" is added under the "Key" column.
- Snapshot summary:** Click refresh to view backup information.
- Create volume:** The final button at the bottom right.

At this stage, select the **new volume**, then from the **Actions** menu, click on **Attach Volume**.

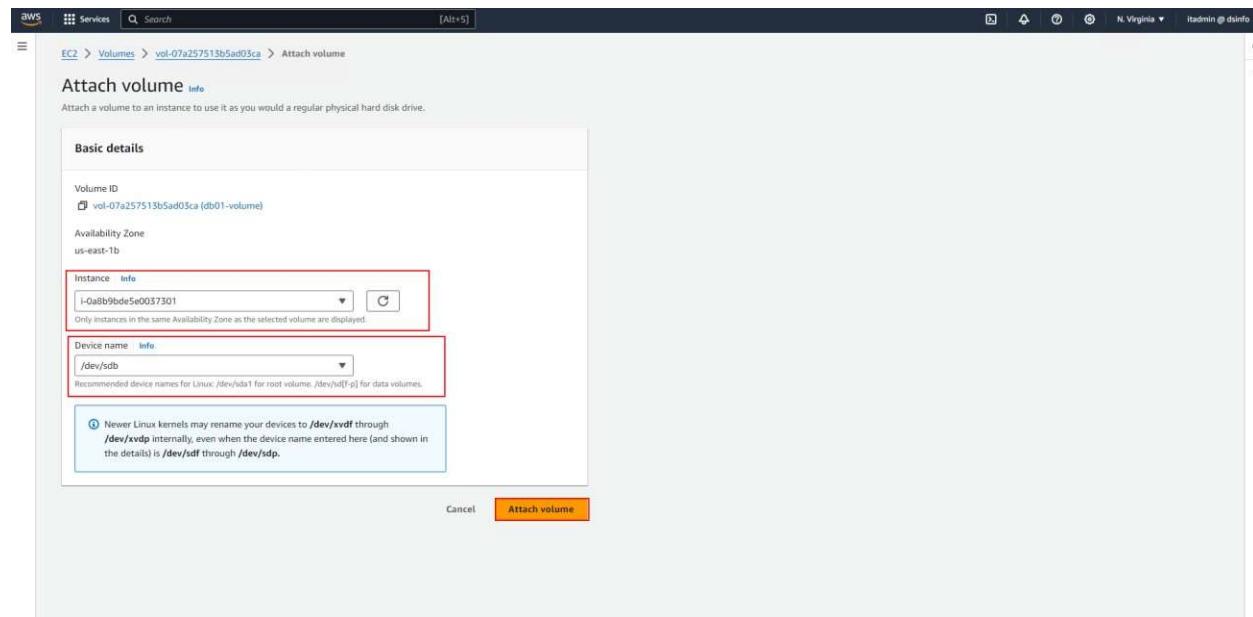


The screenshot shows the AWS EC2 Volumes list. There are two volumes listed:

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created	Availability Zone
web-01-volume	vol-06919be83d30eb0a8	gp3	8 GiB	3000	125	snap-04819fa...	2024/03/21 23:16 GMT-7	us-east-1b
db01-volume	vol-07a257513b5ad03ca	gp3	5 GiB	3000	125	-	2024/03/21 23:30 GMT-7	us-east-1b

The 'db01-volume' row is selected. The Actions menu is open, and the 'Attach volume' option is highlighted with a red box.

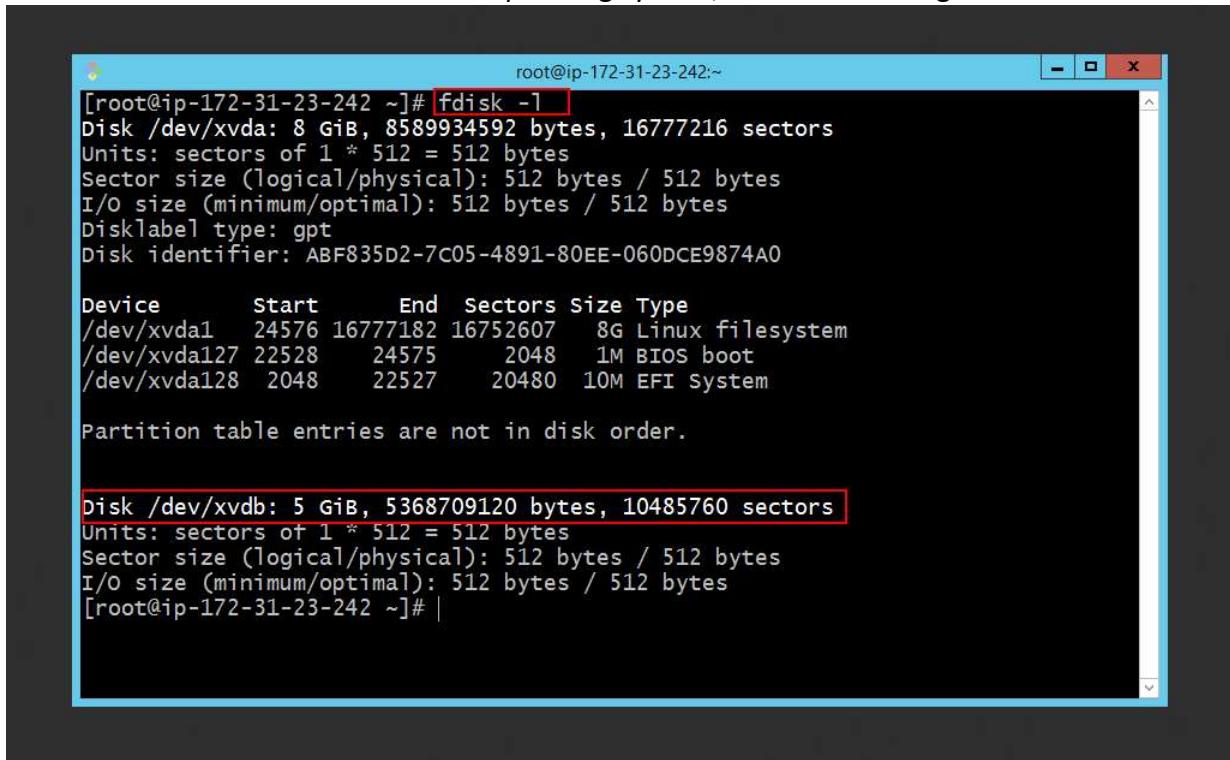
At this stage, from the **Instance** section, select the desired instance, and then click the **Attach Volume** button.



The screenshot shows the 'Attach volume' wizard in the AWS EC2 console. The 'Basic details' step is active. The volume ID is set to 'vol-07a257513b5ad03ca (db01-volume)'. The availability zone is 'us-east-1b'. An instance is selected: 'i-0a8b9bde5e0037301'. The device name is set to '/dev/sdb'. A note at the bottom states: 'Never Linux kernels may rename your devices to /dev/xvdf through /dev/xvd internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.' The 'Attach volume' button is highlighted with a red box.

Then, SSH into your **EC2 instance** and switch to the **Root** user environment by using the following command

To view the new volume in the Linux operating system, use the following command



The screenshot shows a terminal window with the following text:

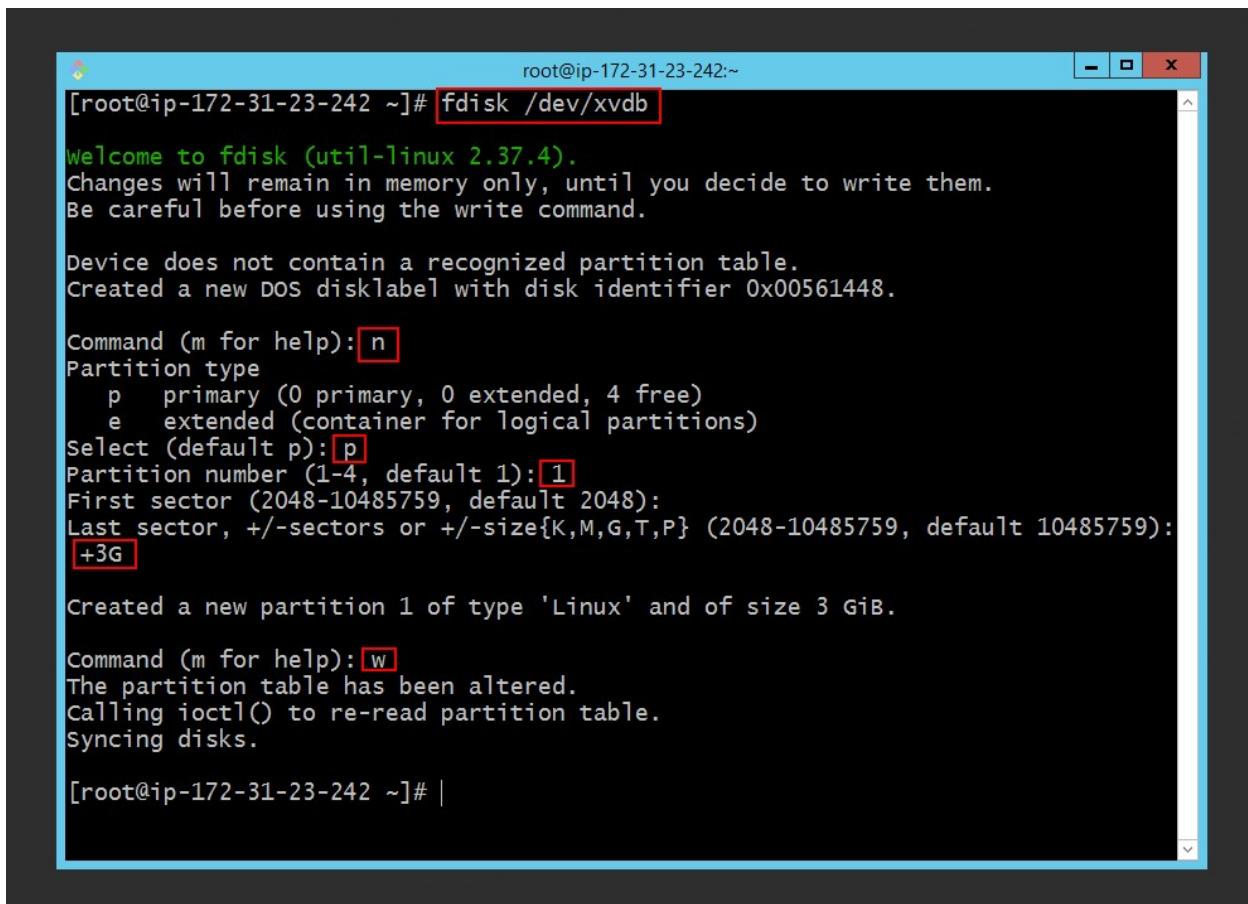
```
root@ip-172-31-23-242 ~]# fdisk -l
Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: ABF835D2-7C05-4891-80EE-060DCE9874A0

Device      Start    End  Sectors Size Type
/dev/xvda1   24576 16777182 16752607  8G Linux filesystem
/dev/xvda127 22528    24575     2048  1M BIOS boot
/dev/xvda128  2048    22527    20480 10M EFI System

Partition table entries are not in disk order.

Disk /dev/xvdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
[root@ip-172-31-23-242 ~]# |
```

Use the following command to partition the new volume. In this command, the switch **n** stands for **New Partition**, **p** stands for **Primary**, and **1** is the partition number. Then, specify the partition size, which we have set to **3G**, and finally press **w** to save the changes.



The screenshot shows a terminal window with a blue header bar containing the text "root@ip-172-31-23-242:~". The main area of the terminal shows the following command being run:

```
[root@ip-172-31-23-242 ~]# fdisk /dev/xvdb
```

Output from the command:

```
Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x00561448.

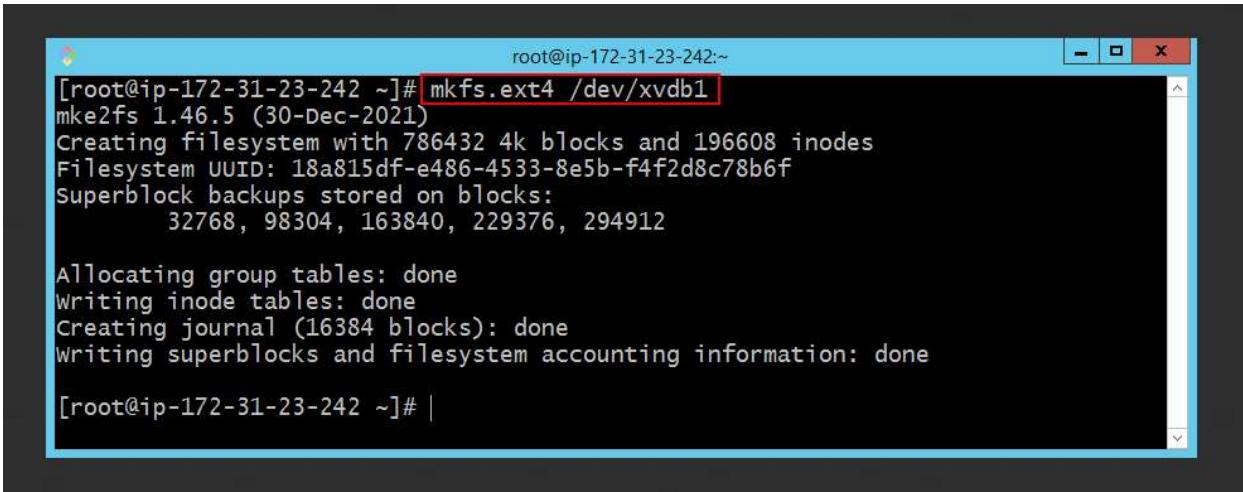
Command (m for help): n
Partition type
  p  primary (0 primary, 0 extended, 4 free)
  e  extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-10485759, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-10485759, default 10485759):
+3G

Created a new partition 1 of type 'Linux' and of size 3 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

[root@ip-172-31-23-242 ~]# |
```

After creating the partition, you need to format it with a file system. Use the following command

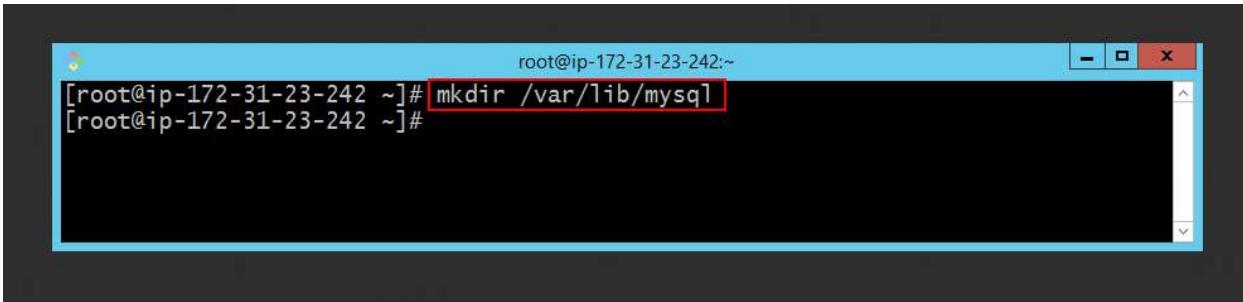


```
root@ip-172-31-23-242:~# mkfs.ext4 /dev/xvdb1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 786432 4k blocks and 196608 inodes
Filesystem UUID: 18a815df-e486-4533-8e5b-f4f2d8c78b6f
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

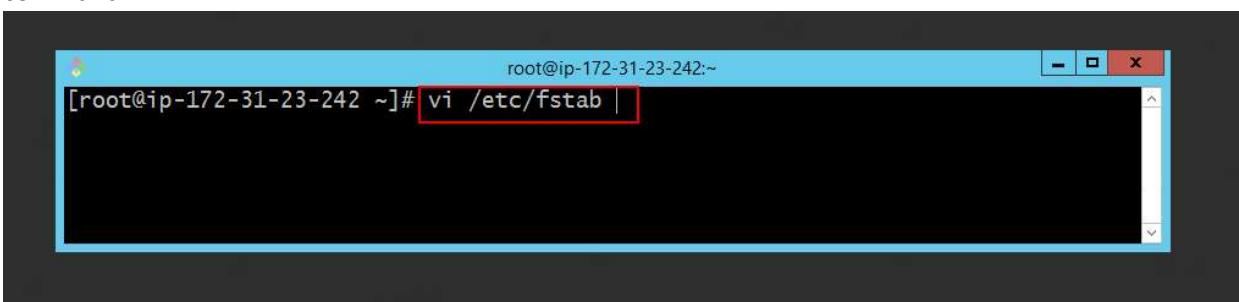
[root@ip-172-31-23-242:~]#
```

At this stage, create the directory for storing the database files using the following command



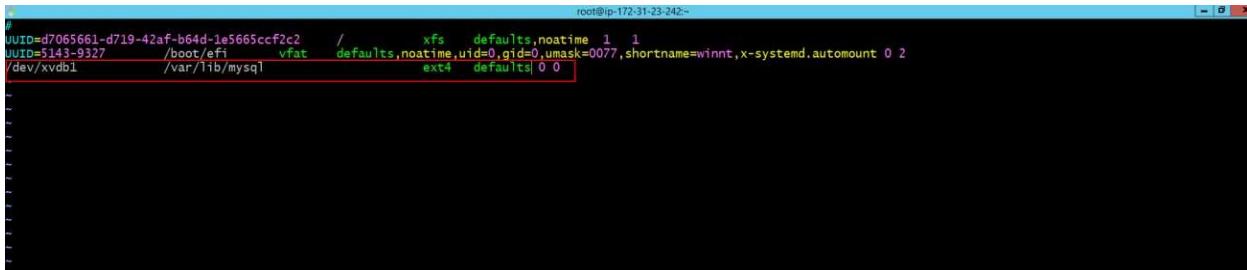
```
root@ip-172-31-23-242:~# mkdir /var/lib/mysql
[root@ip-172-31-23-242:~]#
```

To mount the database path to the new partition, you need to edit the **fstab** file using the following command



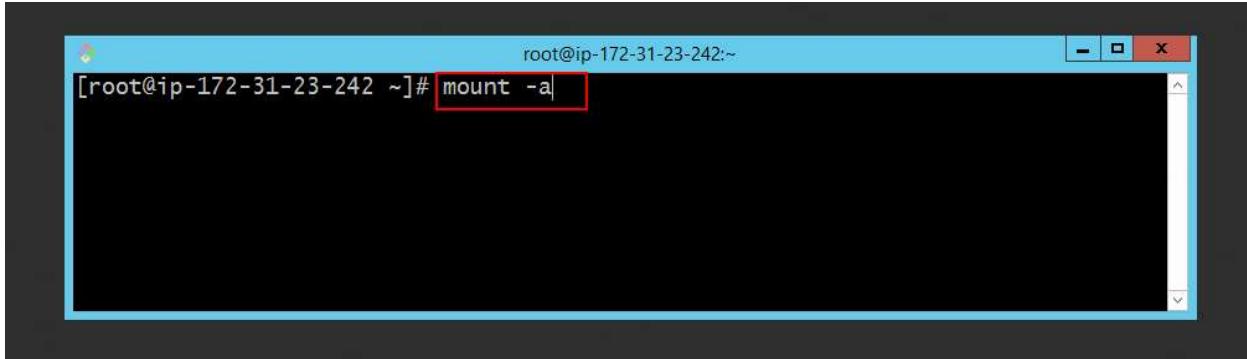
```
root@ip-172-31-23-242:~# vi /etc/fstab |
```

At the end of the **fstab** file, add the following line to mount the partition. After adding the line, save the file



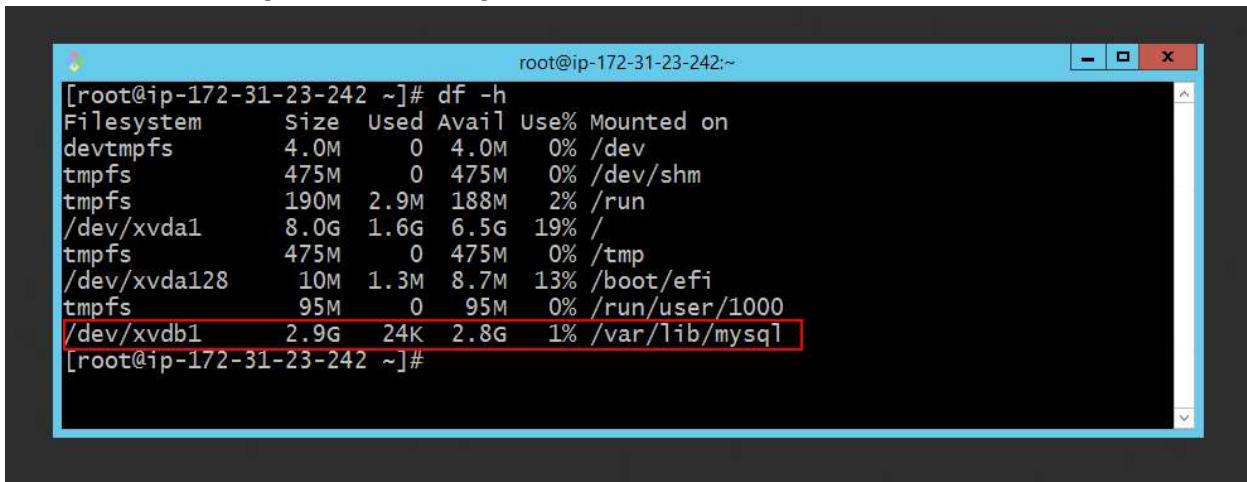
```
#  
UUID=d7065661-d719-42af-b64d-1e5665ccf2c2    /      xfs    defaults,noatime 1 1  
UUID=5143-9327        /boot/efi    vfat   defaults,noatime,uid=0,gid=0,umask=0077,shortname=winnt,x-systemd.automount 0 2  
/dev/xvdb1          /var/lib/mysql    ext4   defaults 0 0
```

To mount the partition, use the following command



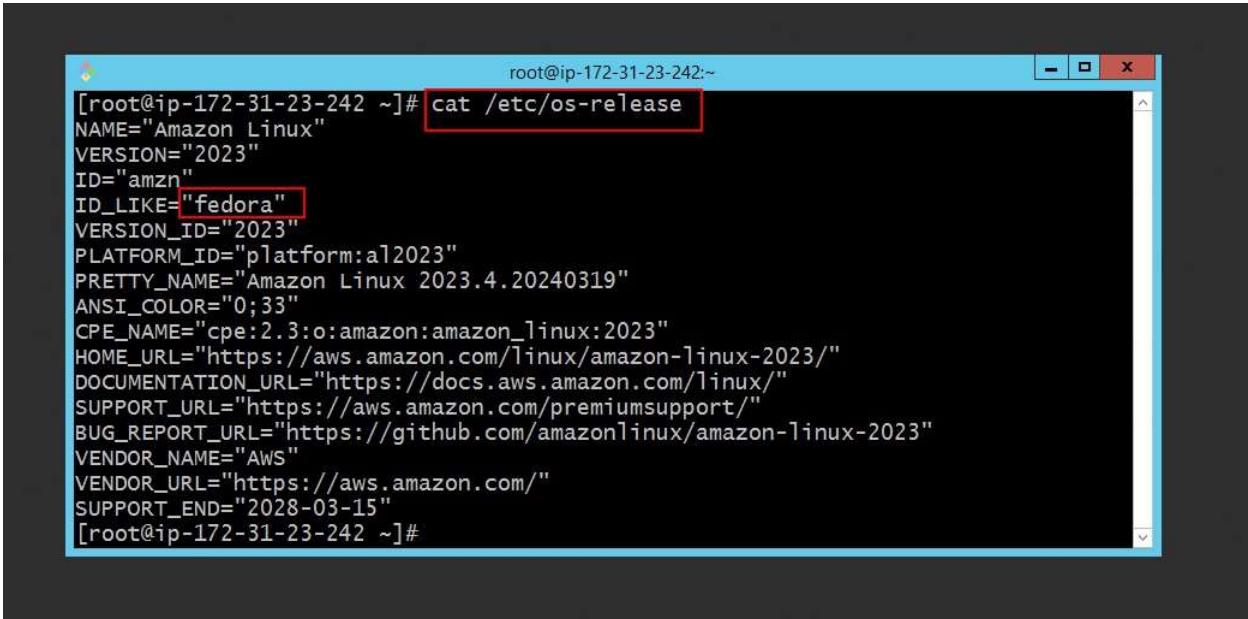
```
[root@ip-172-31-23-242 ~]# mount -a
```

To check the mounting, use the following command



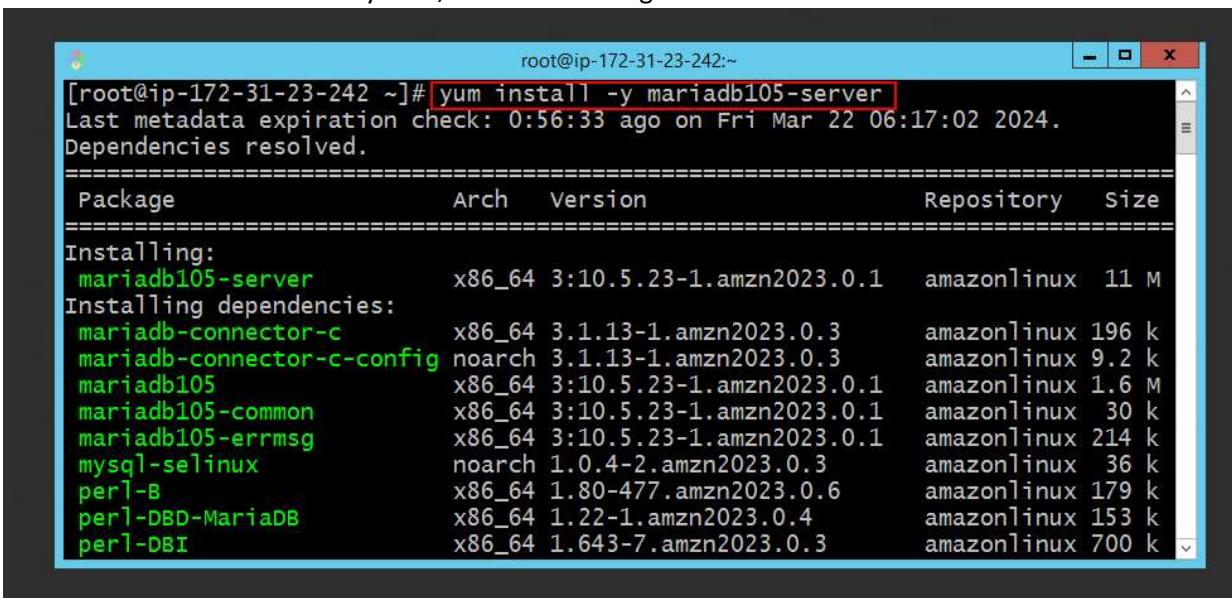
```
[root@ip-172-31-23-242 ~]# df -h  
Filesystem      Size  Used Avail Use% Mounted on  
devtmpfs        4.0M   0    4.0M  0% /dev  
tmpfs          475M   0   475M  0% /dev/shm  
tmpfs          190M  2.9M  188M  2% /run  
/dev/xvda1       8.0G  1.6G  6.5G  19% /  
tmpfs          475M   0   475M  0% /tmp  
/dev/xvda128     10M  1.3M  8.7M  13% /boot/efi  
tmpfs          95M   0   95M  0% /run/user/1000  
/dev/xvdb1       2.9G  24K  2.8G  1% /var/lib/mysql  
[root@ip-172-31-23-242 ~]#
```

Before installing the database, check the version of the operating system using the following command



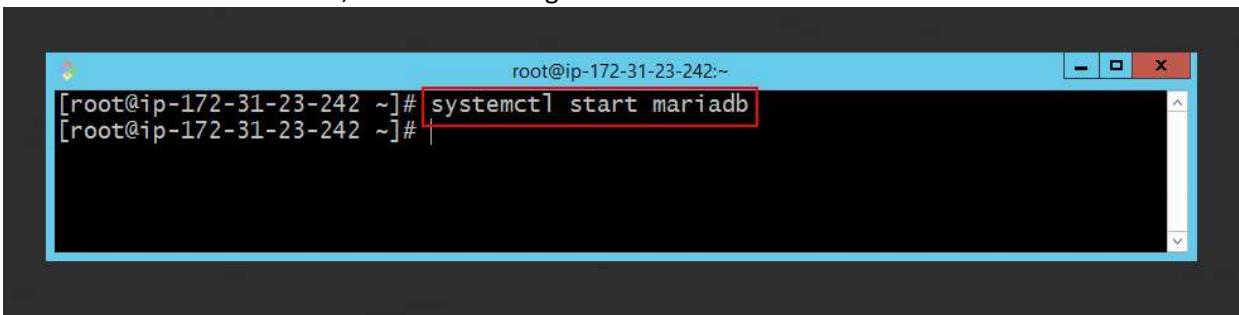
```
root@ip-172-31-23-242 ~]# cat /etc/os-release
NAME="Amazon Linux"
VERSION="2023"
ID="amzn"
ID_LIKE="fedora"
VERSION_ID="2023"
PLATFORM_ID="platform:el2023"
PRETTY_NAME="Amazon Linux 2023.4.20240319"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2023"
HOME_URL="https://aws.amazon.com/linux/amazon-linux-2023/"
DOCUMENTATION_URL="https://docs.aws.amazon.com/linux/"
SUPPORT_URL="https://aws.amazon.com/premiumsupport/"
BUG_REPORT_URL="https://github.com/amazonlinux/amazon-linux-2023"
VENDOR_NAME="AWS"
VENDOR_URL="https://aws.amazon.com/"
SUPPORT_END="2028-03-15"
[root@ip-172-31-23-242 ~]#
```

To install **MariaDB** on a Linux system, use the following command



```
root@ip-172-31-23-242 ~]# yum install -y mariadb105-server
Last metadata expiration check: 0:56:33 ago on Fri Mar 22 06:17:02 2024.
Dependencies resolved.
=====
 Package           Arch   Version        Repository      Size
=====
Installing:
 mariadb105-server    x86_64  3:10.5.23-1.amzn2023.0.1  amazonlinux  11 M
Installing dependencies:
 mariadb-connector-c    x86_64  3.1.13-1.amzn2023.0.3   amazonlinux  196 k
 mariadb-connector-c-config  noarch 3.1.13-1.amzn2023.0.3   amazonlinux  9.2 k
 mariadb105            x86_64  3:10.5.23-1.amzn2023.0.1  amazonlinux  1.6 M
 mariadb105-common     x86_64  3:10.5.23-1.amzn2023.0.1  amazonlinux  30 k
 mariadb105-errmsg     x86_64  3:10.5.23-1.amzn2023.0.1  amazonlinux  214 k
 mysql-selinux         noarch 1.0.4-2.amzn2023.0.3   amazonlinux  36 k
 perl-B                x86_64  1.80-477.amzn2023.0.6   amazonlinux  179 k
 perl-DBD-MariaDB     x86_64  1.22-1.amzn2023.0.4   amazonlinux  153 k
 perl-DBI              x86_64  1.643-7.amzn2023.0.3  amazonlinux  700 k
```

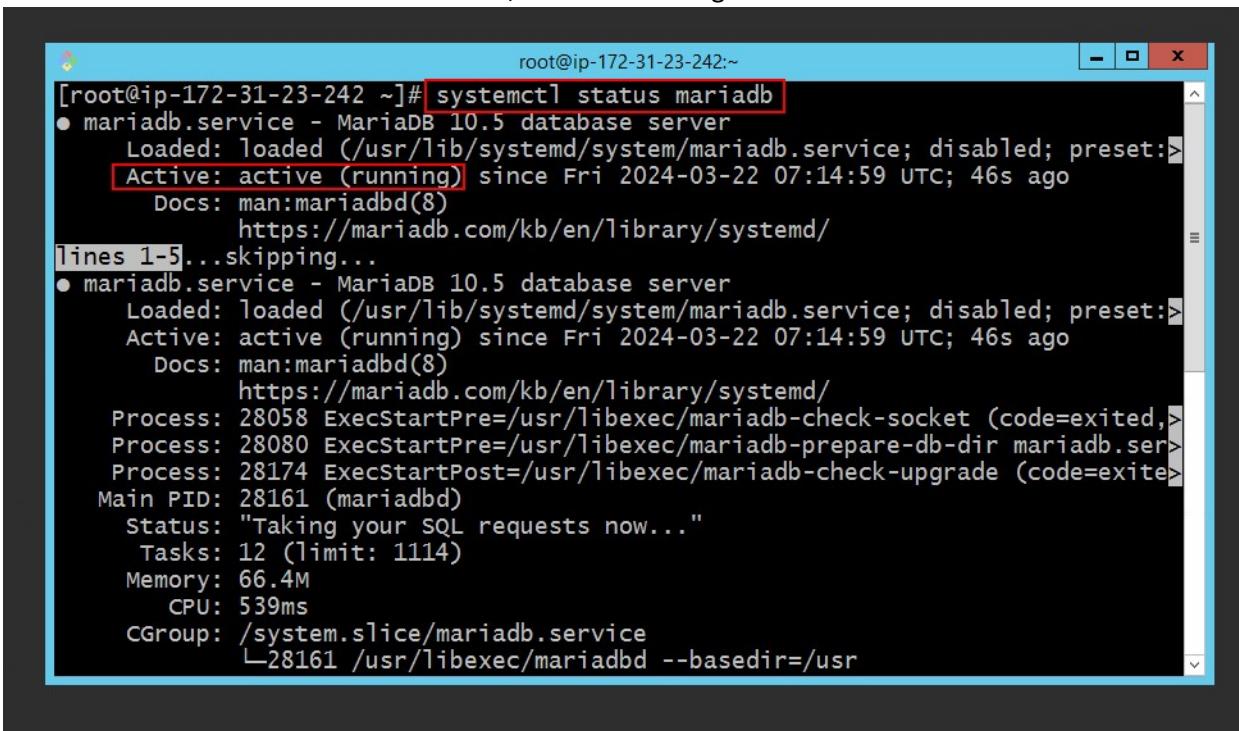
To start the **MariaDB** service, use the following command



```
root@ip-172-31-23-242:~# systemctl start mariadb
[root@ip-172-31-23-242:~#]
```

A screenshot of a terminal window titled "root@ip-172-31-23-242:~". The command "systemctl start mariadb" is entered in the terminal. The entire command line is highlighted with a red box.

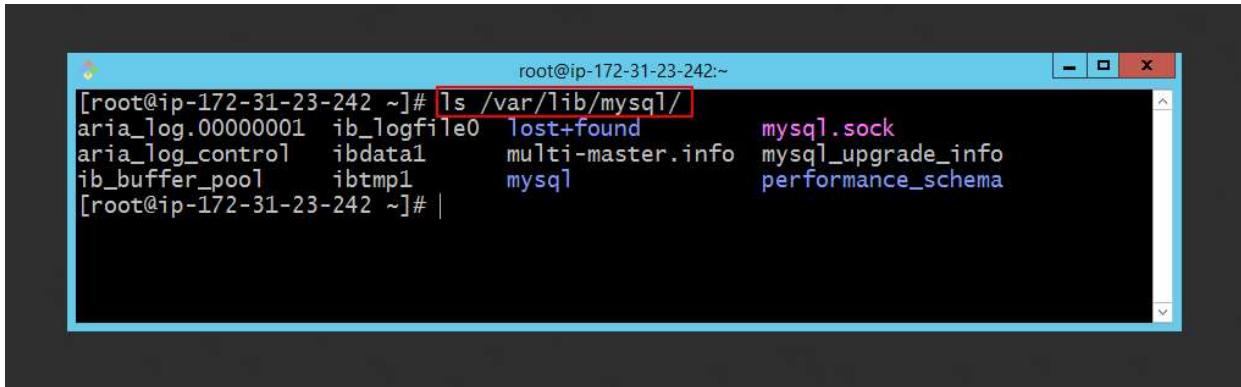
To check the status of the **MariaDB** service, use the following command



```
root@ip-172-31-23-242:~# systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
  Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; preset:>)
  Active: active (running) since Fri 2024-03-22 07:14:59 UTC; 46s ago
    Docs: man:mariadb(8)
          https://mariadb.com/kb/en/library/systemd/
lines 1-5...skipping...
● mariadb.service - MariaDB 10.5 database server
  Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; preset:>)
  Active: active (running) since Fri 2024-03-22 07:14:59 UTC; 46s ago
    Docs: man:mariadb(8)
          https://mariadb.com/kb/en/library/systemd/
  Process: 28058 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited,>
  Process: 28080 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir mariadb.ser>
  Process: 28174 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exite>
  Main PID: 28161 (mariadb)
    Status: "Taking your SQL requests now..."
      Tasks: 12 (limit: 1114)
     Memory: 66.4M
        CPU: 539ms
      CGroup: /system.slice/mariadb.service
              └─28161 /usr/libexec/mariadb --basedir=/usr
```

A screenshot of a terminal window titled "root@ip-172-31-23-242:~". The command "systemctl status mariadb" is entered in the terminal. The entire command line is highlighted with a red box. The output shows two instances of the mariadb.service running, both in an active (running) state. It provides detailed information about each instance, including process IDs, memory usage, and status messages.

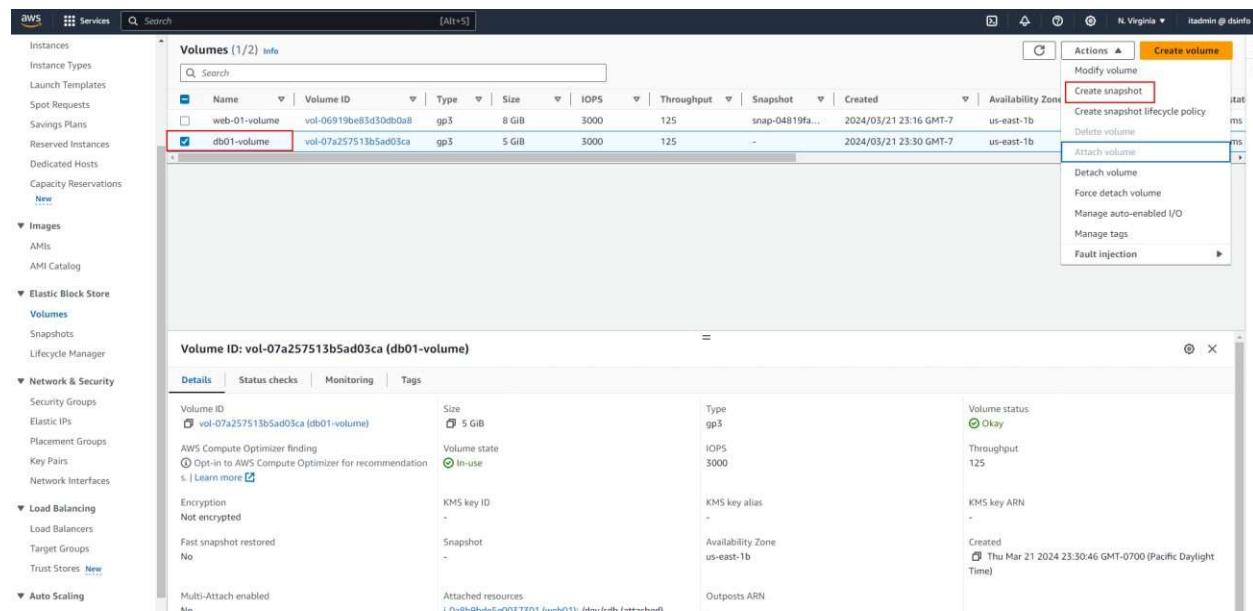
As shown in the image below, after installing the database, the database files are located in the path mounted on the new volume.



```
root@ip-172-31-23-242:~# ls /var/lib/mysql/
aria_log.00000001  ib_logfile0  lost+found      mysql.sock
aria_log_control   ibdata1     multi-master.info mysql_upgrade_info
ib_buffer_pool    ibtmp1     mysql           performance_schema
[root@ip-172-31-23-242 ~]#
```

How to Take a Snapshot of a Volume

At this stage, select the **new volume**, then in the **Actions** menu, click on **Create Snapshot**.



The screenshot shows the AWS Management Console interface for the Elastic Block Store (EBS) service. On the left, there is a navigation sidebar with various AWS services like Instances, Images, and Network & Security. The main area displays a table of volumes. One volume, 'db01-volume' (Volume ID: vol-07a257513b5ad03ca), is selected and highlighted with a red box. In the 'Actions' dropdown menu for this volume, the 'Create snapshot' option is also highlighted with a red box. Below the table, a detailed view of the selected volume is shown. The 'Details' tab is active, displaying information such as Volume ID, Size (5 GiB), Type (gp3), Volume state (In-use), Throughput (125), and Availability Zone (us-east-1b). Other tabs like Status checks, Monitoring, and Tags are also present.

At this stage, provide a **description** in the **Description** field, define a **tag** if needed, and finally, click on **Create Snapshot**.

The screenshot shows the 'Create snapshot' dialog box in the AWS Management Console. The 'Volume ID' is set to 'vol-07a257513b5ad03ca (db01-volume)'. The 'Description' field contains 'db volume snapshot'. A single tag 'Name: db01-volume-snap' is added under the 'Tags' section. The 'Create snapshot' button is highlighted in orange at the bottom right.

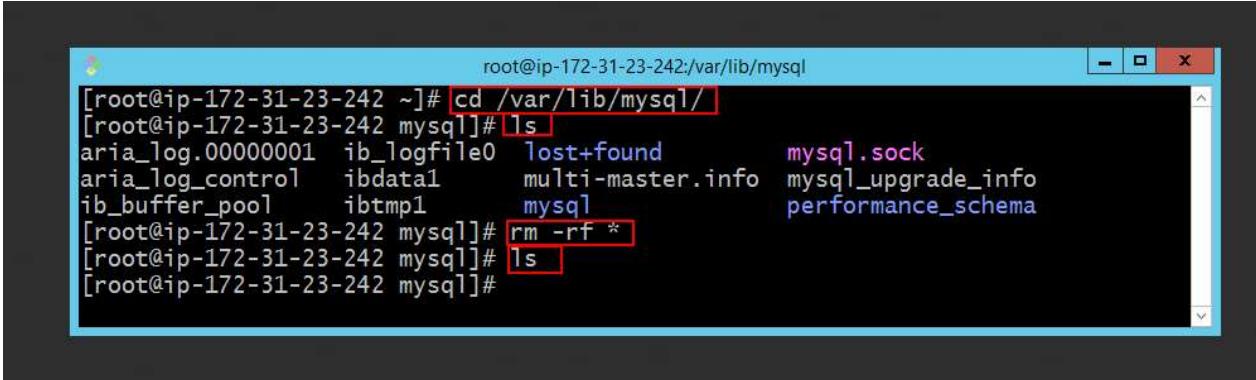
To view the created snapshot, click on the **Snapshots** section.

The screenshot shows the 'Snapshots (1)' list in the AWS Management Console. The table displays one snapshot entry:

Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started	Progress	Encryption	Actions
db01-volume...	snap-daf4099916501f229	5 GB	db volume snapshot	Standard	Completed	2024/03/22 00:53 GMT-7	Available (100%)	Not encrypted	<button>Create snapshot</button>

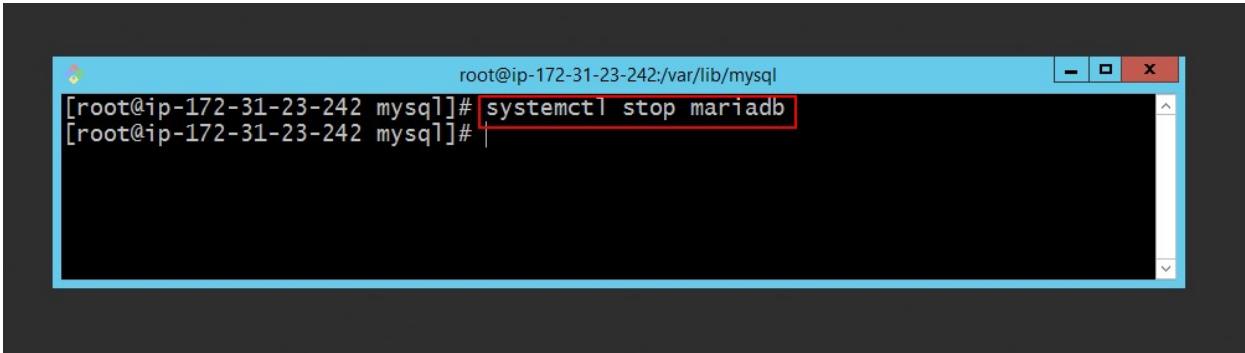
How to Use a Snapshot

At this stage, first, delete all the files in the **database path**. Be aware that you currently have a **snapshot** of this volume, so you can restore the data



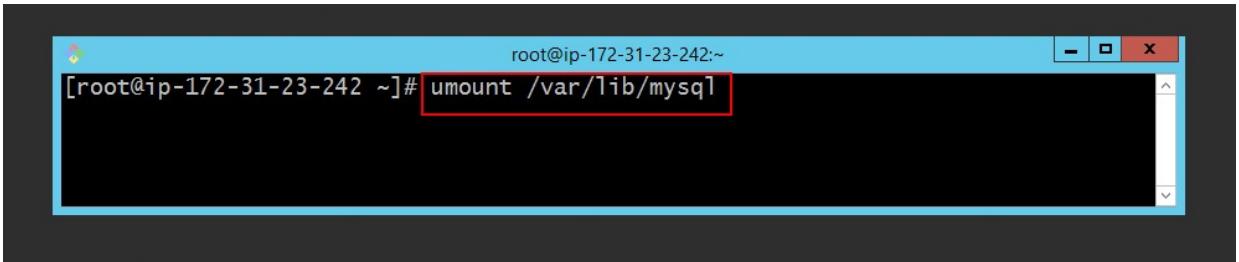
```
root@ip-172-31-23-242:~]# cd /var/lib/mysql
[root@ip-172-31-23-242 mysql]# ls
aria_log.00000001 ib_logfile0  Lost+found      mysql.sock
aria_log_control ibdata1     multi-master.info  mysql_upgrade_info
ib_buffer_pool    ibtmp1     mysql               performance_schema
[root@ip-172-31-23-242 mysql]# rm -rf *
[root@ip-172-31-23-242 mysql]# ls
[root@ip-172-31-23-242 mysql]#
```

Then, stop the **MariaDB** service using the following command



```
root@ip-172-31-23-242:~]# systemctl stop mariadb
[root@ip-172-31-23-242 ~]#
```

At this stage, unmount the **database path** using the following command



```
root@ip-172-31-23-242:~]# umount /var/lib/mysql
[root@ip-172-31-23-242 ~]#
```

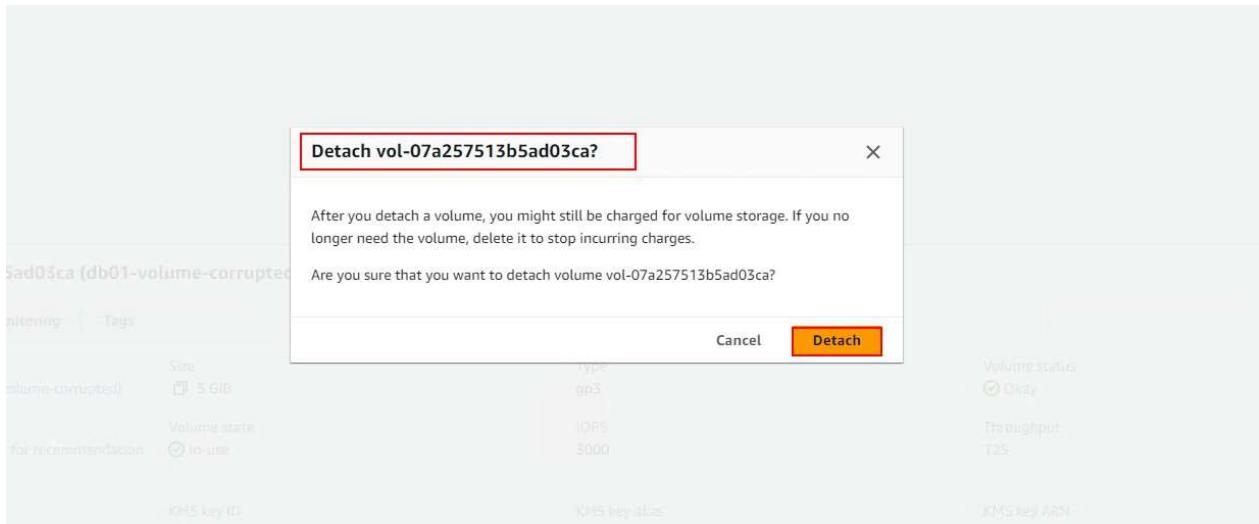
Next, click on the **Volumes** section, and then name the new volume.

The screenshot shows the AWS EC2 Volumes page. In the left sidebar, under 'Elastic Block Store', the 'Volumes' option is selected. In the main content area, there is a table of volumes. One volume, 'db01-volume', has its name changed to 'db01-volume-corrupted'. A modal dialog is open over the table, titled 'Edit Name', with the new name 'db01-volume-corrupted' entered. Below the table, a detailed view of the volume 'db01-volume-corrupted' is shown, including its ID, type (gp3), size (5 GiB), and creation date (2024/03/21 23:30 GMT-7). The volume is listed as 'In-use' in the 'Availability Zone' column.

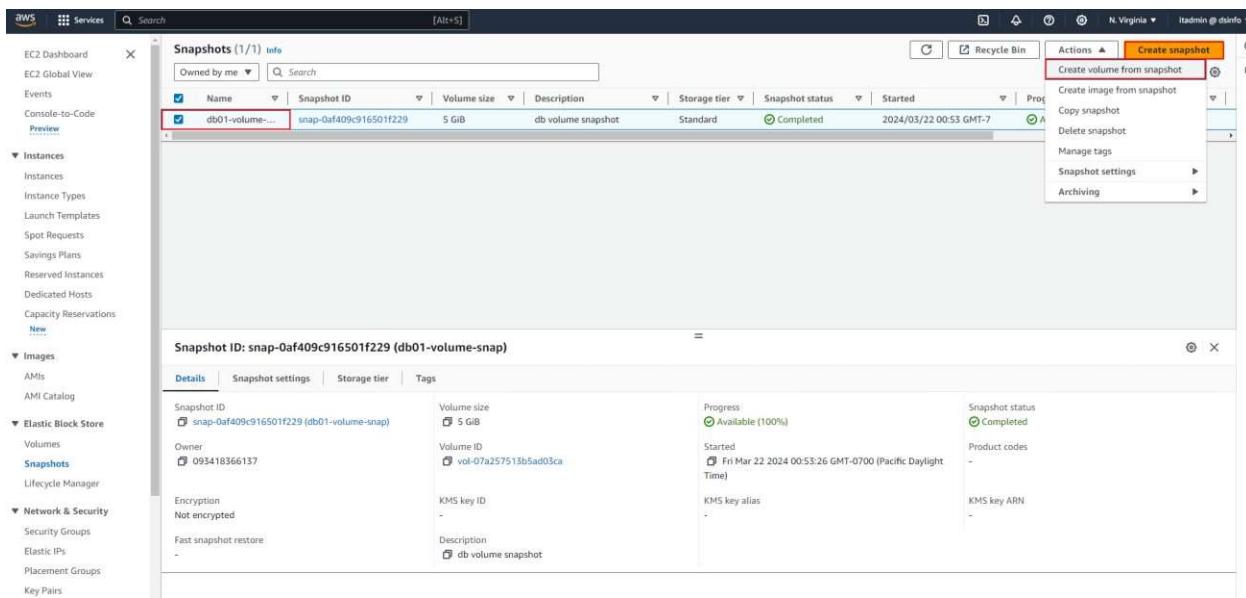
At this stage, select the **new volume**, then from the **Actions** menu, click on **Detach Volume**.

The screenshot shows the same AWS EC2 Volumes page as before. The volume 'db01-volume-corrupted' is selected. The 'Actions' menu is open, and the 'Detach volume' option is highlighted with a red box. Other options in the menu include 'Modify volume', 'Create snapshot', 'Create snapshot lifecycle policy', 'Delete volume', 'Attach volume', 'Force detach volume', 'Manage auto-enabled I/O', and 'Manage tags'.

At this stage, click the **Detach** button to disconnect the volume from the EC2 instance.



Next, in the **Snapshots** section, click on the snapshot that we took in the previous steps, and from the **Actions** menu, click on **Create Volume from Snapshot**.



At this stage, specify the **Volume Type**, **Size**, and **Availability Zone** for the new volume. Make sure the **Availability Zone** matches the zone of your EC2 instance.

The screenshot shows the 'Volume settings' section of the AWS EBS volume creation wizard. Key fields include:

- Volume type:** General Purpose SSD (gp3) (selected)
- Size (GiB):** 5
- IOPS:** 3000
- Throughput (MiB/s):** 125
- Availability Zone:** us-east-1b

Below these fields, there are sections for 'Fast snapshot restore' (disabled) and 'Encryption' (unchecked).

Then, define a **Tag** and click on the **Create Volume** button.

The screenshot shows the 'Snapshot summary' section of the EBS volume creation wizard, summarizing the configuration:

- Throughput (MiB/s): 125
- Availability Zone: us-east-1b
- Encryption: Not enabled for selected snapshot
- Tags - optional:
 - Key: Name, Value: db01-volume-recovered

At the bottom, there are 'Cancel' and 'Create volume' buttons.

At this stage, select the **new volume**, then from the **Actions** menu, click on **Attach Volume** to attach the new volume to your EC2 instance.

The screenshot shows the AWS Volumes list page. A volume named "db01-volume-recovered" is selected. In the Actions menu, the "Attach volume" option is highlighted with a red box. The volume details page is also visible, showing its configuration.

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created	Actions
web-01-volume	vol-06919be83d50db0a8	gp3	8 GiB	3000	125	snap-04819fa...	2024/03/21 23:16 GMT-7	us
db01-volume-corrupted	vol-07a257513b5ad05ca	gp3	5 GiB	3000	125	-	2024/03/21 23:30 GMT-7	us
db01-volume-recovered	vol-071342c8f6763bd39	gp3	5 GiB	3000	125	snap-0af409c...	2024/03/22 01:09 GMT-7	us

Volume ID: vol-071342c8f6763bd39 (db01-volume-recovered)

Details | Status checks | Monitoring | Tags

Volume ID: vol-071342c8f6763bd39 (db01-volume-recovered) | Size: 5 GiB | Type: gp3 | Volume status: Okay | Throughput: 125

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

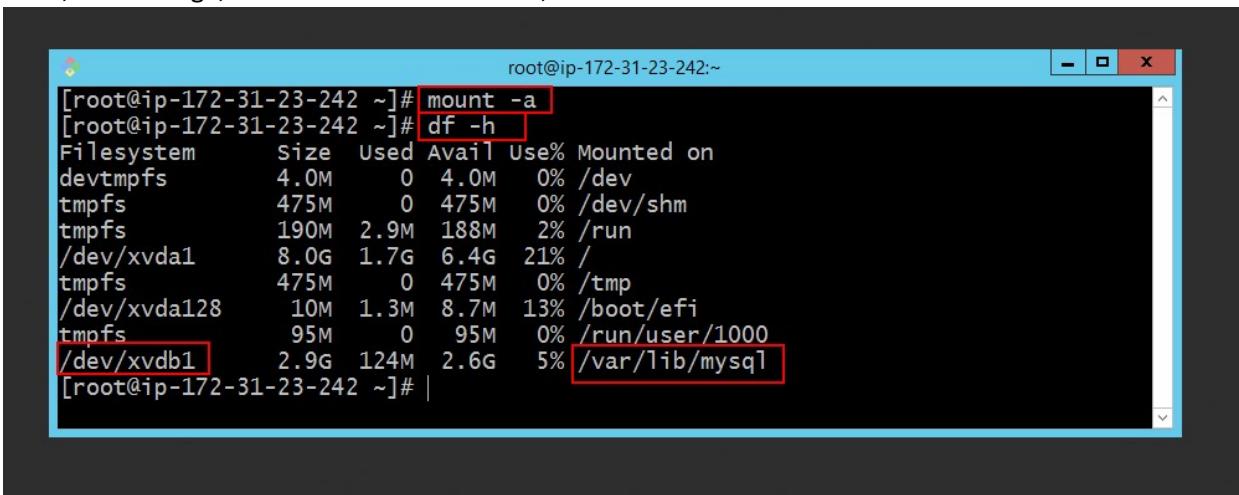
Encryption: Not encrypted | KMS key ID: | KMS key alias: | KMS key ARN: | Created: Fri Mar 22 2024 01:09:56 GMT-0700 (Pacific Daylight Time)

Fast snapshot restored: No | Snapshot: snap-0af409c916501f229 | Availability Zone: us-east-1b | Attached resources: | Outposts ARN: |

In this section, you need to select your **Instance**, and then click on **Attach Volume**.

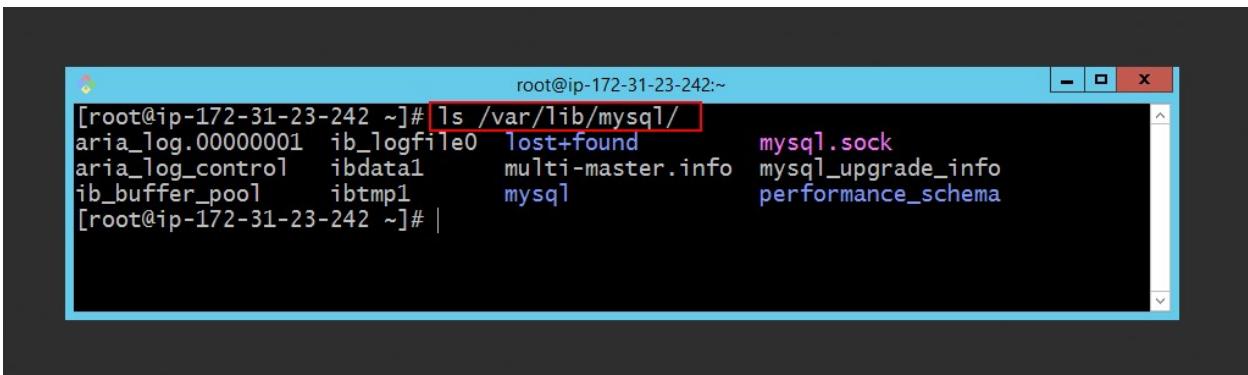
The screenshot shows the "Attach volume" dialog box. The "Basic details" section is displayed, with the "Volume ID" set to "vol-071342c8f6763bd39 (db01-volume-recovered)". The "Availability Zone" is "us-east-1b". The "Instance" dropdown is set to "i-0a8b9bde5e0037301". The "Device name" dropdown is set to "/dev/sdb". A note at the bottom states: "Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdw internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp." The "Attach volumes" button is highlighted with a red box.

Then, at this stage, to mount the new volume, use the command



```
root@ip-172-31-23-242:~# mount -a
root@ip-172-31-23-242:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/devtmpfs        4.0M   0    4.0M  0% /dev
tmpfs           475M   0   475M  0% /dev/shm
tmpfs           190M  2.9M  188M  2% /run
/dev/xvda1       8.0G  1.7G  6.4G 21% /
tmpfs           475M   0   475M  0% /tmp
/dev/xvda128     10M  1.3M  8.7M 13% /boot/efi
tmpfs            95M   0   95M  0% /run/user/1000
/dev/xvdb1       2.9G  124M  2.6G  5% /var/lib/mysql
[root@ip-172-31-23-242:~]
```

As shown in the image below, the database files have been successfully recovered in the **database path**.



```
root@ip-172-31-23-242:~# ls /var/lib/mysql/
aria_log.00000001  ib_logfile0  lost+found          mysql.sock
aria_log_control   ibdata1     multi-master.info   mysql_upgrade_info
ib_buffer_pool     ibtmp1     mysql               performance_schema
[root@ip-172-31-23-242:~]
```

Introduction to AWS ELB

In this section, we will explore one of the exciting features of AWS Cloud, called **Elastic Load Balancer (ELB)**.

When you want to create a cluster of services, you need to set up multiple servers within the cluster. For example, let's assume you create several web servers. You need a **single endpoint** to access these web servers, and this endpoint is referred to as a **Load Balancer**.

If you're using AWS Cloud, you don't need to use your own load balancer. For example, if you're using **NGINX**, **HAproxy**, or any other load balancer, you can easily switch to using **AWS ELB**.

An **ELB** has two types of ports, known as **Frontend Port** and **Backend Port**.

The **Frontend Port** listens for user requests from the internet. For example, when you try to access **Google.com** on port **443**.

The **Backend Port** is the port associated with the service that is listening on the operating system. For example, the **Tomcat** service that runs on port **8080**.

AWS Elastic Load Balancer receives traffic and then distributes it across multiple **targets**.

These targets are typically **EC2 instances**, but they can also be **containers**. You can have multiple **IP addresses** across several zones. In other words, you can have a cluster across multiple zones, and the **Load Balancer** acts as the endpoint for them.

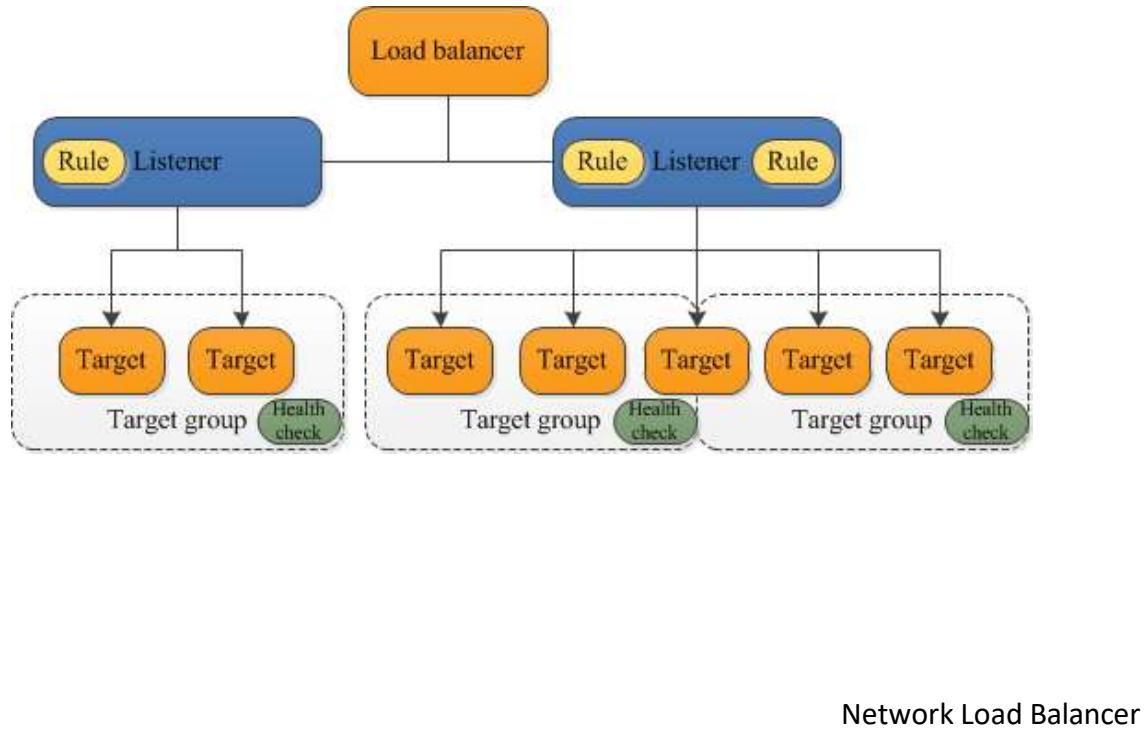
Types of AWS ELB

In AWS, there are several types of Load Balancers, including the following

Application Load Balancer

This **Load Balancer** is only for **web traffic**.

This **Load Balancer** operates at **Layer 7** of the OSI model, meaning there is no need to route traffic based on ports. It can route traffic based on the **content** of the request, such as URL paths and host headers.



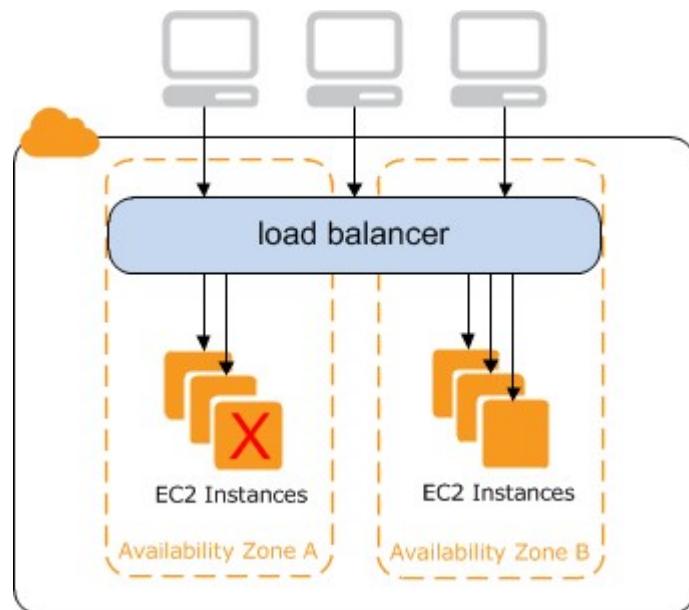
This **Load Balancer** provides **high performance** and is **very costly**.

This **Load Balancer** operates at **Layer 4** of the OSI model and can handle **millions of requests per second**.

-Classic Load Balancer

It is the simplest type of **Load Balancer** and can be easily used by anyone.

This **Load Balancer** receives incoming traffic and forwards it to the **backend servers**, operating at the **network layer** (Layer 4) of the OSI model.



How to Set Up AWS ELB

At this stage, we intend to set up **multiple web servers** and use **HTML templates** from the Tooplate website on each server. Then, we will use a **Load Balancer** to **balance traffic** between these web servers.

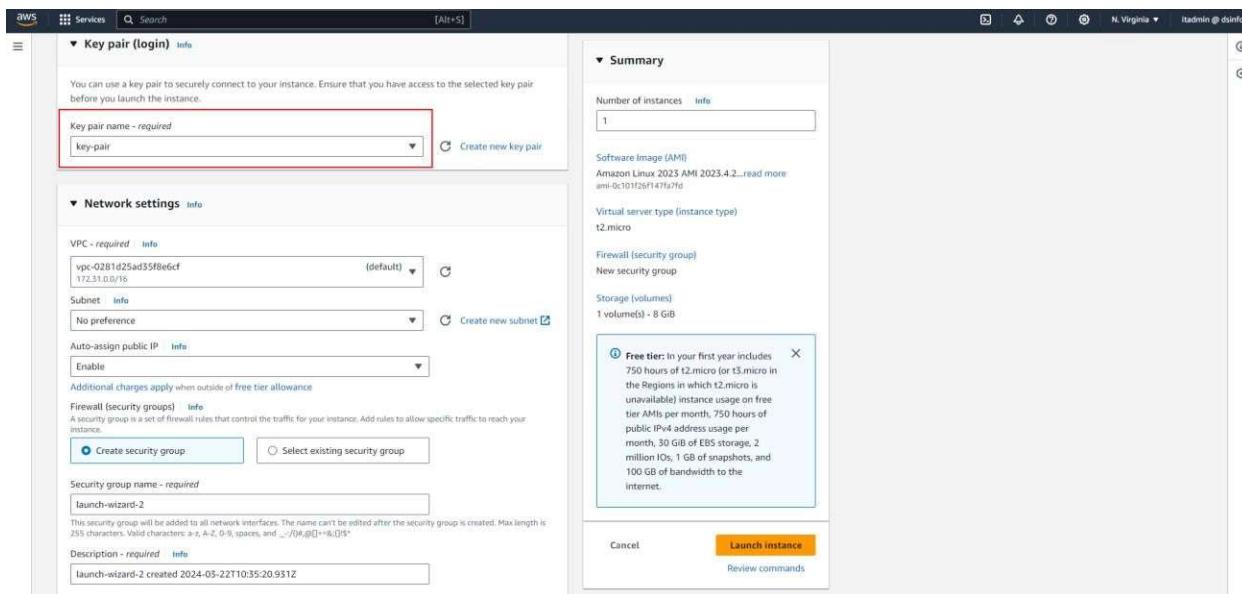
In the first step, we create an **EC2 instance** to set up a **web server**.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, Global View, Events, and Instances (which is selected). Under Instances, there are sub-options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Images. The main content area shows a table titled 'Instances info' with one row. The row details an instance named 'i-0a8b9bde5e0037301 (web01)'. The instance is listed as 'Terminated'. The 'Details' tab is selected, showing information such as Instance ID, Public IPv4 address, Instance type (t2.micro), and VPC ID. A large button at the bottom right of the main content area says 'Launch instances'.

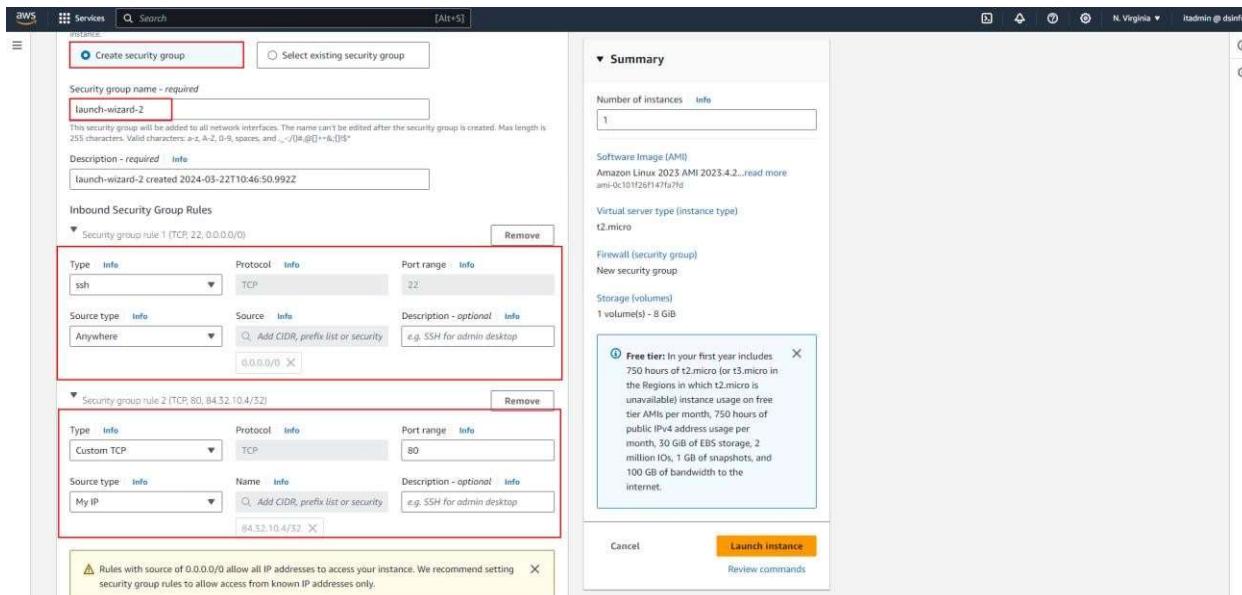
At this stage, specify a **name** for the EC2 instance and select the **AMI type**.

The screenshot shows the 'Launch an instance' wizard. The first step, 'Name and tags', has a 'Name' field containing 'web01'. The second step, 'Application and OS Images (Amazon Machine Image)', shows a search bar and a list of AMI icons including Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. Below this is a 'Quick Start' section with a table showing AMI details. A callout box highlights the 'Free tier' information: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' At the bottom are 'Cancel', 'Launch instance', and 'Review commands' buttons.

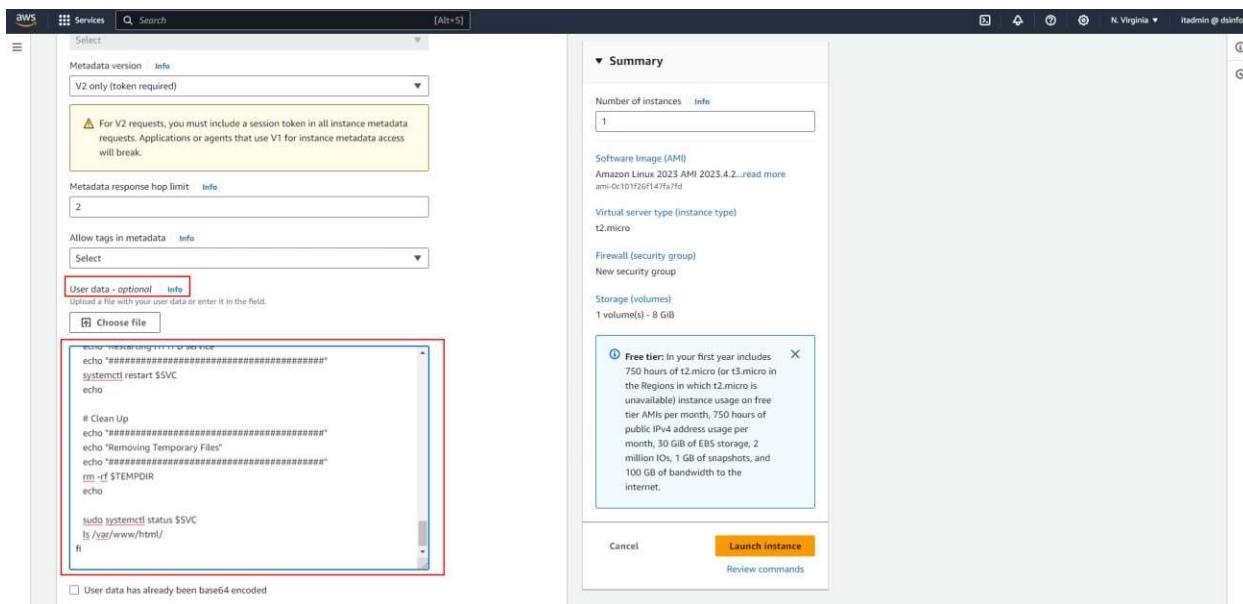
At this stage, you need to define a **Key Pair**.



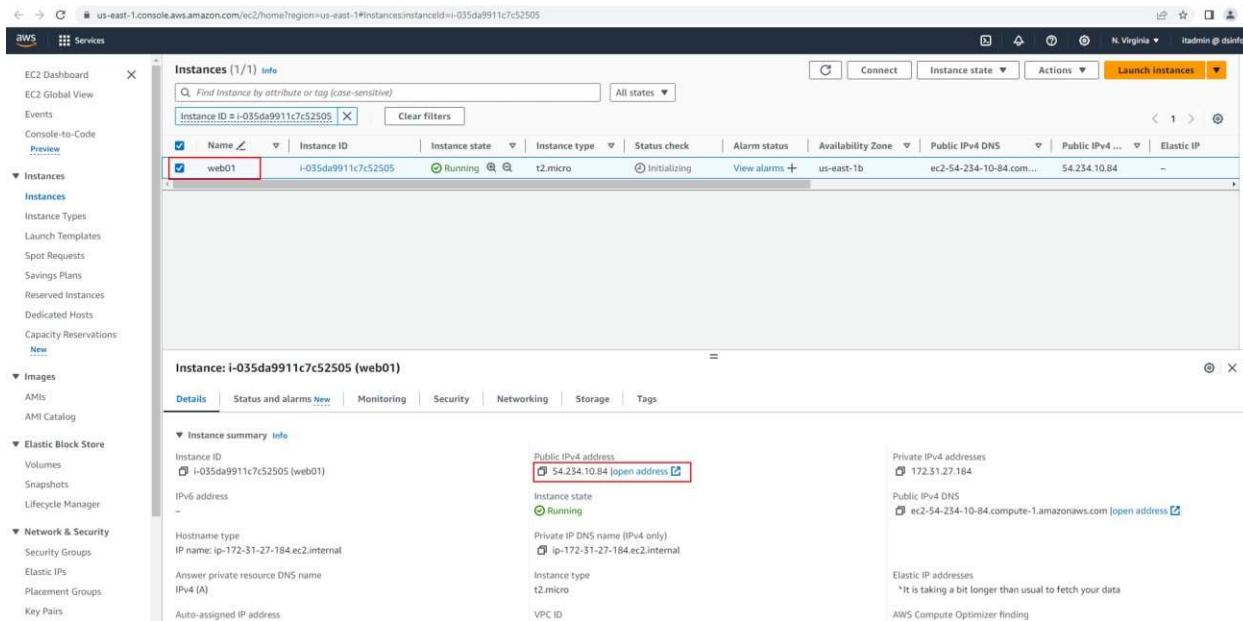
At this stage, you need to configure the **Security Group** settings for this EC2 instance. Open ports **80** and **22** in the **Inbound Rules** section.



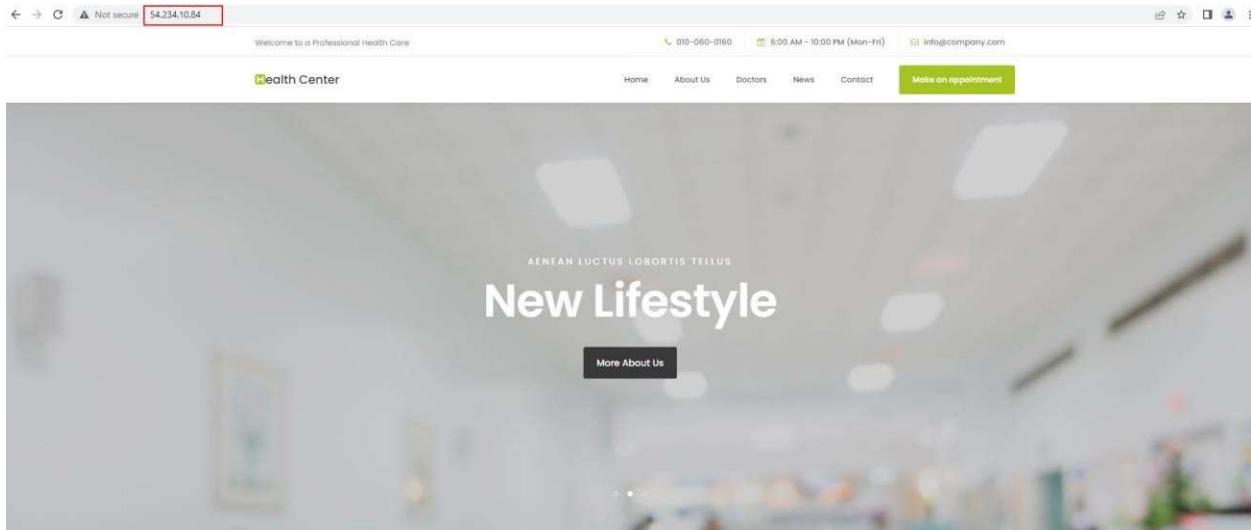
At this stage, paste the **web server setup script** into the **User Data** section.



After the **EC2 instance** is created, select it and then copy its **Public IP**.



To test the web service, enter its **Public IP** in a browser. As you can see, the **web service is functioning correctly**.



At this stage, select the **EC2 instance**, then from the **Actions** menu choose **Image and Templates**, and finally click on **Create Image**.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like EC2 Dashboard, Services, Events, Console-to-Code, Instances (selected), Images, and Elastic Block Store. The main area displays a table of instances. One instance, 'web01' (Instance ID: i-035da9911c7c52505), is selected and highlighted with a red box. In the 'Actions' dropdown menu, the 'Image and templates' option is also highlighted with a red box. At the bottom of the Actions menu, the 'Create image' button is highlighted with a red box.

Choose a **name** for the image, then click on the **Create Image** button.

The screenshot shows the 'Create Image' dialog box. At the top, there is a field for 'Image name' containing 'Health-AM'. Below it is a 'Image description - optional' field with the placeholder 'Image description'. Underneath these fields are two checkboxes: 'No reboot' (unchecked) and 'Enable'. The next section is titled 'Instance volumes' and contains a table with columns: Storage type, Device, Snapshot, Size, Volume type, IOPS, Throughput, Delete on termination, and Encrypted. A row is shown with 'EBS' as the storage type, '/dev/xvda' as the device, 'Create new snapshot from volume' as the snapshot, '8' as the size, 'EBS General Purpose S...' as the volume type, '3000' as the IOPS, 'Delete on termination' checked, and 'Enable' checked. Below the table is a button 'Add volume'. A note below says: 'During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.' The 'Tags - optional' section follows, with a note: 'A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.' There are two radio button options: 'Tag image and snapshots together' (selected) and 'Tag image and snapshots separately'. Both options have a note: 'Tag the image and the snapshots with the same tag.' and 'Tag the image and the snapshots with different tag.' respectively. Below this is a note: 'No tags associated with the resource.' and a button 'Add new tag'. A note at the bottom says: 'You can add up to 50 more tags.' At the bottom right are 'Cancel' and 'Create image' buttons.

Then, click on the **Launch Templates** section and afterward click on the **Create Launch Template** button.

The screenshot shows the 'Launch Templates' page in the AWS EC2 console. The left sidebar has a tree view with 'Launch Templates' selected. The main area shows a table with columns: Launch Template ID, Launch Template Name, Default Version, Latest Version, Create Time, and Created By. A note at the top right says: 'Loading Launch Templates...'. Below the table is a modal window titled 'Select a launch template' with a close button 'X'.

At this stage, specify a **name** and **version** for the template.

The screenshot shows the 'Create launch template' wizard in the AWS Management Console. The current step is 'Launch template name and description'. The 'Launch template name - required' field contains 'Health-Template'. The 'Template version description' field contains 'V1'. A note below says 'Must be unique to this account. Max 128 chars. No spaces or special characters like \\", \", \\", \\".' Below these fields is 'Auto Scaling guidance' with a checkbox for 'Provide guidance to help me set up a template that I can use with EC2 Auto Scaling'. Under 'Template tags' and 'Source template', there are sections for adding tags and selecting a source template. On the right, the 'Summary' section includes 'Software Image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)'. A callout box for the 'Free tier' provides details about instance usage and free services. At the bottom are 'Cancel' and 'Create launch template' buttons.

At this stage, you need to select your **AMI**.

The screenshot shows the 'Create launch template' wizard in the AWS Management Console. The current step is 'Application and OS Images (Amazon Machine Image)'. The 'Search our full catalog including 1000s of application and OS images' bar has 'My AMIs' selected. Below it are filters for 'Don't include in launch template', 'Owned by me' (selected), and 'Shared with me'. The 'Amazon Machine Image (AMI)' section shows a single entry: 'Health-AMI' (ami-056e19828de61afec). Below this is a 'Description' section with a single line. The 'Architecture' section shows 'x86_64' and 'ami-056e19828de61afec'. At the bottom, the 'Instance type' dropdown is set to 'Don't include in launch template', and the 'All generations' checkbox is checked. A callout box for the 'Free tier' provides details about instance usage and free services. At the bottom are 'Cancel' and 'Create launch template' buttons.

At this stage, you need to specify the **Instance Type** and the **Key Pair**.

The screenshot shows the 'Instance type' section with 't2.micro' selected. It includes a note about free tier eligibility and a link to compare instance types. The 'Key pair (login)' section shows 'key-pair' selected. The 'Network settings' section shows 'Select existing security group' chosen. A modal window on the right provides details about the free tier for t2.micro instances.

At this stage, you need to configure the **Security Group**.

The screenshot shows the 'Network settings' section with 'Select existing security group' chosen. The 'Storage (volumes)' section shows 'Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp3))' selected. A modal window on the right provides details about the free tier for t2.micro instances.

At this stage, define a **Tag** for the template.

The screenshot shows the 'Create new template' wizard. In the 'Storage (volumes)' section, there is a note about free tier EBS storage and an 'Add new volume' button. In the 'Resource tags' section, a single tag 'Name: web00' is defined. A detailed note about the free tier is visible in the background.

Then, at this stage, select your **template** and from the **Actions** menu, choose **Launch Instance from Template**.

The screenshot shows the 'Launch templates' page. It displays the 'Health-Template' (lt-0234b4505458f7974). The 'Actions' menu is open, highlighting the 'Launch instance from template' option. The template details include the launch template ID, name, default version, owner, and creation date. The 'Launch template version details' section shows the current version (V1) with its AMI ID, instance type (t2.micro), security group, and network interfaces.

At this stage, specify the **Source Template**.

EC2 > Launch templates > Launch instance from template

Launch instance from template

Launching from a template allows you to launch from an instance configuration that you would have saved in the past. These saved configurations can be reused and shared with other users to standardize launches across an organisation.

Choose a launch template

Source template

Health-Template
ID: It-0234b4505458f7974

1 (Default)
V1

Instance details

Your instance details are listed below. Any fields that are not specified as part of the configuration below will use the template or default values for those fields. Ensure that you have permissions to override these parameters or your instance launch will fail.

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

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

AMIs from catalog | Recents | My AMIs | Quick Start

Summary

Number of instances Info
1

Software Image (AMI)
Health-AMI
ami-056e19828de61afec

Virtual server type (instance type)
t2.micro

Firewall (security group)
launch-wizard-2

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Launch instance | Review commands

At this stage, assign a **Tag** to the instance and then click the **Launch Instance** button.

Security groups info

Select security groups

launch-wizard-2 sg-090128606760a4c13 X

Compare security group rules

Advanced network configuration

Storage (volumes)

EBS Volumes

Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp3))

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

Add new volume

Resource tags

Key Info
Name Value Info
Q. web002 X
Select resource type... Instances X

Add new tag

You can add up to 49 more tags.

Summary

Number of instances Info
1

Software Image (AMI)
Health-AMI
ami-056e19828de61afec

Virtual server type (instance type)
t2.micro

Firewall (security group)
launch-wizard-2

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Launch instance | Review commands

As shown in the image below, a **new instance** has been created from our **template**.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Console-to-Code, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and AMI Catalog. The main area is titled 'Instances (2) Info' and contains a table with two rows. The first row is for 'web001' (Instance ID: i-035da9911c7c52505, Status: Running, Type: t2.micro, 2/2 checks passed, View alarms, Availability Zone: us-east-1b, Public IPv4 DNS: ec2-54-234-10-84.com..., Public IPv4 IP: 54.234.10.84) and the second row is for 'web002' (Instance ID: i-00be1e861e41bc292, Status: Running, Type: t2.micro, 2/2 checks passed, View alarms, Availability Zone: us-east-1b, Public IPv4 DNS: ec2-54-201-104-7.com..., Public IPv4 IP: 34.201.104.7). Both rows have a red box around them. At the top right, there are buttons for 'Launch instances' and other actions.

At this stage, click on the **Load Balancing** section, then click on **Target Groups**, and afterward click the **Create Target Group** button.

The screenshot shows the AWS Load Balancing Target Groups page. On the left, there's a sidebar with navigation links for Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing (which is highlighted with a red box), Target Groups (which is also highlighted with a red box), and Trust Stores. The main area is titled 'Target groups' and contains a table with one row. The row has a red box around it. The table has columns for Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. Below the table, it says 'No target groups' and 'You don't have any target groups in us-east-1'. At the bottom right, there's a 'Create target group' button.

At this stage, select the following options as shown in the image, and then click the **Next** button.

The screenshot shows the 'Specify group details' step of the 'Create target group' wizard. The 'Basic configuration' section is highlighted with a red border. In the 'Choose a target type' dropdown, 'Instances' is selected. Under 'Target group name', the value 'Health-TG' is entered. Below the target group name, there are three protocol options: 'HTTP/2', 'HTTP2', and 'gRPC'. The 'HTTP/2' option is selected. The 'Health checks' section is also highlighted with a red border. The 'Health check protocol' dropdown is set to 'HTTP'. The 'Attributes' and 'Tags - optional' sections are shown below, each with a red border around their respective descriptions.

EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

Instances

- Supports load balancing to instances within a specific VPC
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity

IP addresses

- Facilitates routing to multiple IP addresses and network interfaces on the same instance
- Offers flexibility with microservice based architectures, simplifying inter-application communication
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT

Lambda function

- Facilitates routing to a single Lambda function
- Accessible to Application Load Balancers only

Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer

Target group name

Health-TG

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

HTTP/2

HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP

Health check path

/

Up to 1024 characters allowed.

Advanced health check settings

Attributes

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

Tags - optional

Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel **Next**

At this stage, click on the **Include as Pending Below** button.

The screenshot shows the AWS Lambda Step Functions Create Function wizard. The current step is 'Step 2: Set Function Configuration'. The 'Role' dropdown is highlighted with a red box. Other visible fields include 'Function name' (lambda-1), 'Runtime' (Node.js 16.x), and 'Handler' (index.handler).

Then, at this stage, click on the **Create Target Group** button.

The screenshot shows the AWS Lambda Step Functions Create Function wizard. The current step is 'Step 3: Configure Triggers'. The 'Role' dropdown is highlighted with a red box. Other visible fields include 'Function name' (lambda-1), 'Runtime' (Node.js 16.x), and 'Handler' (index.handler).

As shown in the image below, the instances in the **Target Group** are in a **healthy** state.

The screenshot shows the AWS EC2 Target Groups page. On the left, the navigation menu includes options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Load Balancers, and Target Groups. The Target Groups option is selected and highlighted with a red box. The main content area displays a table titled "Target groups (1/1) Info". A single row is listed: "Health-TG" with ARN "arn:aws:elasticloadbalancing:us-east-1:8080:80", Protocol "HTTP", Target type "Instance", and Load balancer "None associated". The VPC ID is "vpc-0281d25ad35f8e6cf". Below this, a modal window titled "Target group: Health-TG" is open, showing the "Targets" tab selected. It lists two registered targets: "i-00be1e861e41bc292" and "i-035da911c7c52505", both of which are marked as "Normal" in the "Anomaly detection" column.

At this stage, click on the **Load Balancers** section, and then click the **Create Load Balancer** button.

The screenshot shows the AWS EC2 Load Balancers page. The navigation menu is identical to the previous screenshot, with the Target Groups option still highlighted. The main content area shows a table titled "Load balancers". A message at the top states "Introducing resource map for Application Load Balancers. Resource map is a visual representation of the relationships between load balancer resources and provides the ability to view, explore, and troubleshoot the architecture of your load balancer. Resource map can be viewed on the load balancers detail page. Share feedback to help us improve your experience." Below this, the table has columns for Name, DNS name, State, VPC ID, Availability Zones, Type, and Date created. A message "No load balancers" and "You don't have any load balancers in us-east-1" is displayed. At the bottom, a button labeled "Create load balancer" is visible.

At this stage, select **Application Load Balancer**, then click on the **Create** button.

Compare and select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. Learn more [\[?\]](#)

Load balancer types		
Application Load Balancer Info Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers. Create	Network Load Balancer Info Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies. Create	Gateway Load Balancer Info Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls. Create

In this section, choose a **name** for the Load Balancer.

EC2 > Load balancers > Create Application Load Balancer

Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Application Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.
 A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)
Scheme can't be changed after the load balancer is created.
 Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more \[?\]](#)
 Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)
Select the type of IP addresses that your subnets use.
 IPv4
Recommended for internal load balancers.
 Dualstack
Includes IPv4 and IPv6 addresses.

At this stage, select **all Availability Zones** to increase availability.

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, in accordance with your IP address settings.

VPC Info

Select the virtual private cloud (VPC) for your targets or you can create a new VPC [\[?\]](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your target groups [\[?\]](#).

vpc-0281d25ad35f8efc
IPv4 VPC CIDR: 172.31.0.0/16

Mappings Info

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

<input checked="" type="checkbox"/> us-east-1a (use1-az2)	Subnet subnet-073ff5ef9a43683a0
<input checked="" type="checkbox"/> us-east-1b (use1-az4)	Subnet subnet-02e0a980e794cbc1f
<input checked="" type="checkbox"/> us-east-1c (use1-az6)	Subnet subnet-0f30185ac197d551
<input checked="" type="checkbox"/> us-east-1d (use1-az1)	Subnet subnet-0a9892ba955b4150

At this stage, you need to select a **Security Group** for the Load Balancer.

Security groups Info

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#) [\[?\]](#).

Security groups

Select up to 5 security groups [\[?\]](#)

default
sg-08f8ec53665de9a4497 VPC: vpc-0281d25ad35f8efc

Listeners and routing Info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Listener: HTTP:80	Remove	
Protocol: HTTP	Port: 80	Default action: Info
Forward to: Select a target group [?]		
Create target group [?]		

Listener tags - optional

Consider adding tags to your Listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)

You can add up to 50 more tags.

[Add listener](#)

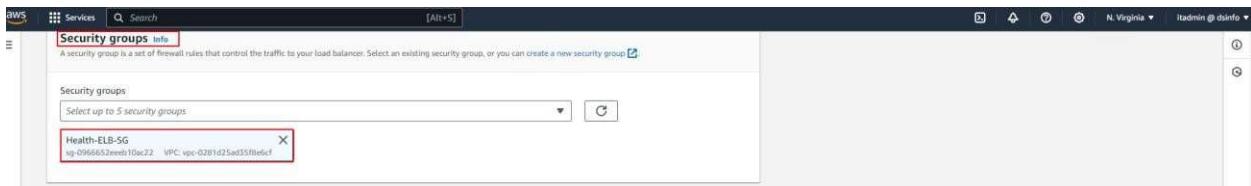
At this stage, specify a **name** for the Security Group.

The screenshot shows the 'Create security group' wizard in the AWS Management Console. The 'Basic details' section is highlighted with a red box. It contains fields for 'Security group name' (Health-ELB-SG) and 'Description' (Health-ELB-SG). Below this is a 'VPC Info' dropdown set to 'vpc-0281d25ad35f8e6cf'. The 'Inbound rules' section is shown below, indicating 'This security group has no inbound rules.' A 'Add rule' button is present.

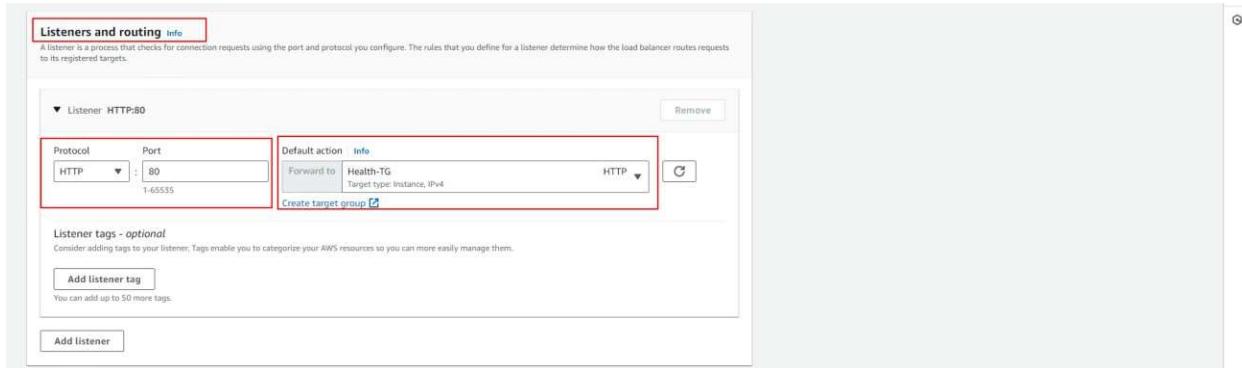
In the **Inbound Rule** section, you need to create the following rules, and then click on the **Create Security Group** button.

The screenshot shows the 'Create security group' wizard with the 'Inbound rules' section highlighted by a red box. Two custom TCP rules are defined: one for port 80 with source 'Anywhere-->' and another for port 80 with source '::/0'. Both rules have an optional description field and a 'Delete' button. A note at the bottom of the 'Inbound rules' section cautions against using '0.0.0.0/0' or '::/0' as sources. The 'Tags - optional' section is also visible at the bottom.

At this stage, select the **Security Group** you have created.



In the **Listener** section, set the **Default Action** to your **Target Group**.



And finally, click on **Create Load Balancer**.



At this stage, you need to select the **Security Group** associated with your instances and click on the **Edit Inbound Rules** button in the **Inbound Rules** section.

The screenshot shows the AWS EC2 console with the 'Security Groups' section selected. A specific security group, 'sg-090128606760a4c13 - launch-wizard-2', is highlighted. The 'Inbound rules' tab is active. A new rule is being configured with the following parameters:

Name	Type	Protocol	Port range	Source
sgr-02767b08477d22a60	SSH	TCP	22	Custom (0.0.0.0/0)
sgr-051a092528319ae14	HTTP	TCP	80	Custom (84.32.10.4/32)
(New Rule)	Custom TCP	TCP	80	sg-0966652eeeb10ac22 (Health-ELB-SG)

Create a new rule of type **TCP** on port **80**, and set its **source** to the **Security Group of the ELB**.

The screenshot shows the 'Edit inbound rules' dialog for the 'sg-090128606760a4c13' security group. A new rule is being added with the following parameters:

Name	Type	Protocol	Port range	Source
(New Rule)	Custom TCP	TCP	80	sg-0966652eeeb10ac22 (Health-ELB-SG)

A red box highlights the 'Source' field, which contains the ID of the Health-ELB-SG security group. A red arrow points from the text 'Health-ELB-SG' to the source field.

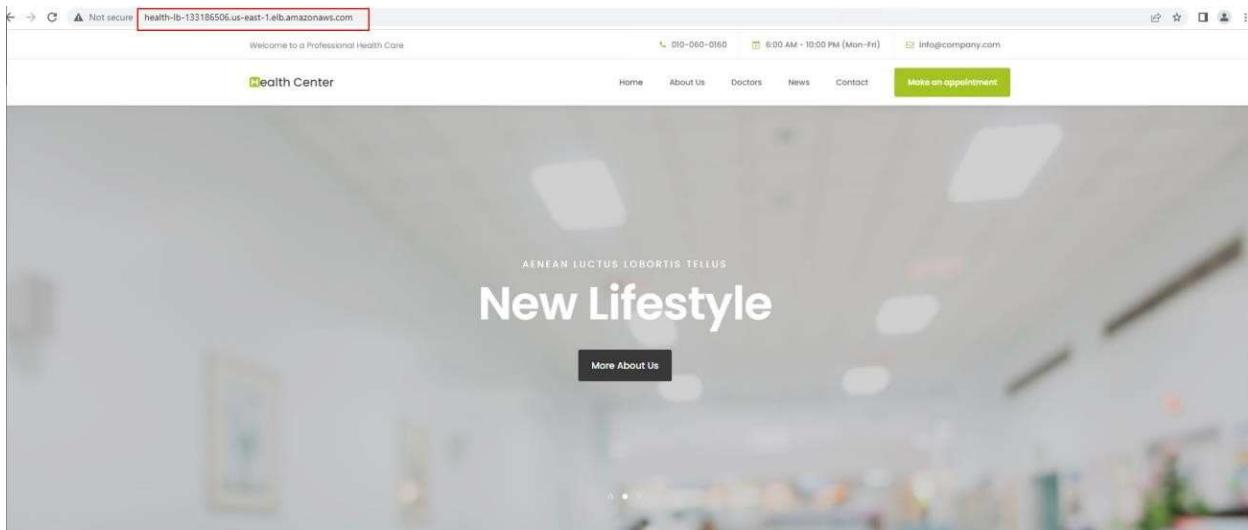
As shown in the image below, the status of the instances in the **Target Group** changes to **Healthy**.

The screenshot shows the AWS EC2 Target groups console. On the left, a navigation sidebar lists various services like Instance Types, Launch Templates, and Network & Security. The main area displays a table for 'Target groups (1/1)'. One row is selected, showing details for 'Health-TG'. The 'Targets' tab is selected in the sub-menu. A table titled 'Registered targets (2)' shows two entries: 'web002' and 'web001', both marked as 'Healthy'. The 'Health status' column contains green circular icons with 'Healthy' text. The 'Health status details' and 'Launch time' columns provide additional information.

At this stage, click on the **Load Balancer** you created and then copy its **DNS Name** link.

The screenshot shows the AWS EC2 Load balancers console. The left sidebar includes 'Load Balancers' under the 'Load Balancing' section. The main area shows a table for 'Load balancers (1/1)'. One row is selected for 'Health-LB', which has a 'DNS name' field containing 'Health-LB-133186506.us-east-1.elb.amazonaws.com'. Below the table, a detailed view for 'Load balancer: Health-LB' is shown. It lists the VPC ID ('vpc-0281d25ad35f8e6cf'), Availability Zones ('us-east-1a (use1-az4), us-east-1b (use1-az2), us-east-1c (use1-az6)'), and the IP address type ('IPv4'). The 'DNS name info' section at the bottom contains the copied URL.

As shown in the image below, the **Load Balancer** is working correctly and is routing traffic to the **EC2 instances**.



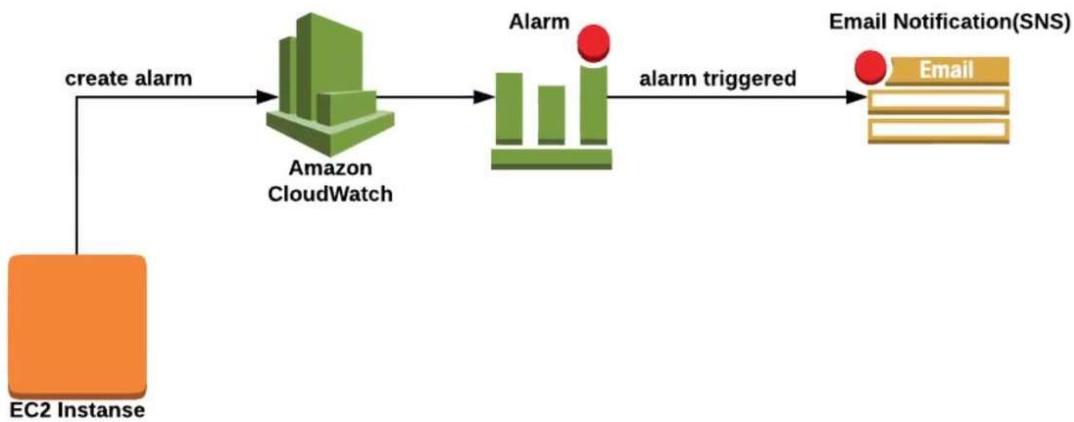
Introduction to AWS CloudWatch

AWS CloudWatch is a **monitoring service**; it is also a **logging solution**.

Use Cases of AWS CloudWatch:

- **Monitoring AWS Resources** such as EC2, RDS, Lambda, EBS, etc.
- **Collecting and tracking metrics** like CPU utilization, memory usage, and disk I/O
- **Setting alarms** to automatically trigger actions when certain thresholds are exceeded
- **Aggregating and storing logs** from applications and AWS services
- **Creating dashboards** for real-time performance visualization
- **Troubleshooting operational issues** by analyzing logs and metrics
- **Automating responses** using CloudWatch Events or Alarms (e.g., stop an instance or send an SNS notification)

In the image below, you can see how **CloudWatch** works.



By default, **CloudWatch** collects data for each metric every **5 minutes** and displays it on graphs. You can reduce this interval to **1 minute**, but there is an additional cost. To enable 1-minute monitoring, go to the **Manage Detailed Monitoring** section and enable the feature.

Introduction to AWS EFS

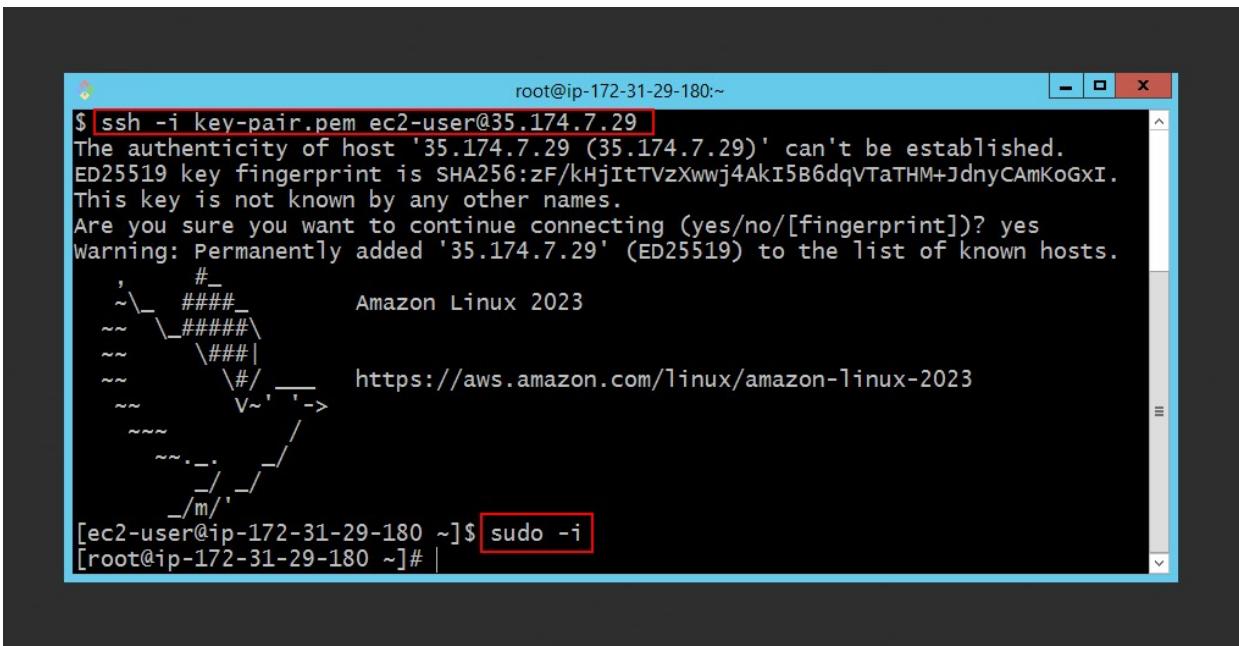
Introduction to AWS EFS (Elastic File System)

AWS **EFS** is a fully managed, scalable, and elastic **network file system** designed for use with AWS Cloud services and on-premises resources. Key features include:

- **Scalable storage** that grows and shrinks automatically as you add or remove files.
- **Shared access** across multiple **EC2 instances** (can be mounted simultaneously).
- Works over **NFS (Network File System)** protocol.
- Provides **high availability and durability** across multiple Availability Zones.
- Ideal for use cases such as **content management, web serving, home directories, development environments, and big data analytics**.

EFS is easy to use, supports standard file system semantics, and is accessible from thousands of EC2 instances concurrently.

At this stage, first, log in to your **EC2 instance**.

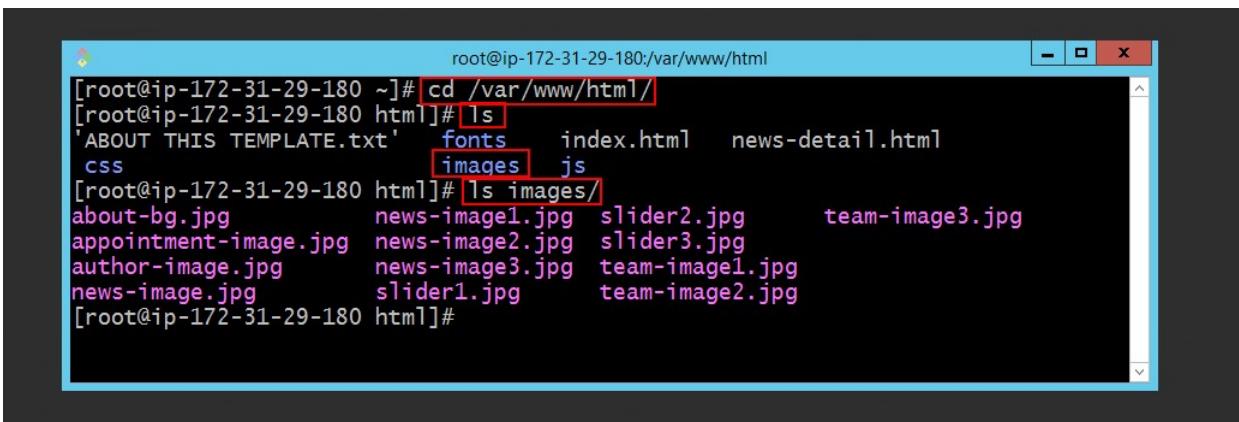


The screenshot shows an SSH session with the following command and output:

```
root@ip-172-31-29-180:~$ ssh -i key-pair.pem ec2-user@35.174.7.29
The authenticity of host '35.174.7.29 (35.174.7.29)' can't be established.
ED25519 key fingerprint is SHA256:zF/kHjItTVzXwwj4AkI5B6dqVTaTHM+JdnyCAmKoGXI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '35.174.7.29' (ED25519) to the list of known hosts.

, _#
~\ _#####
~~ \#####
~~ \|#
~~ \|#/
~~ \|_--> https://aws.amazon.com/linux/amazon-linux-2023
~~ \|_/
~~ \|_/
~/m/|_
[ec2-user@ip-172-31-29-180 ~]$ sudo -i
[root@ip-172-31-29-180 ~]# |
```

As seen in the **web server path**, our web service contains a folder named **Images**, which includes the website's images.



The screenshot shows a terminal session with the following commands and output:

```
root@ip-172-31-29-180:~# cd /var/www/html/
[root@ip-172-31-29-180 html]# ls
'ABOUT THIS TEMPLATE.txt'  fonts  index.html  news-detail.html
css                      images  js
[root@ip-172-31-29-180 html]# ls images/
about-bg.jpg      news-image1.jpg  slider2.jpg    team-image3.jpg
appointment-image.jpg news-image2.jpg  slider3.jpg
author-image.jpg   news-image3.jpg  team-image1.jpg
news-image.jpg     slider1.jpg     team-image2.jpg
[root@ip-172-31-29-180 html]#
```

Go to the **ELB Security Group** section and then click on the **Edit Inbound Rules** button.

The screenshot shows the AWS EC2 console with the 'Security Groups' section selected. A specific security group, 'sg-0ed4d08504a257a3e - SG-ELB', is highlighted with a red box. The 'Inbound rules' tab is also highlighted with a red box. The table below lists three existing inbound rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-0f7fa9205dbd2ca85	IPv4	HTTP	TCP	80	84.32.10.4/32	-
-	sgr-0dd8d2052de349965	-	HTTP	TCP	80	sg-0966652eeeb10ac...	-
-	sgr-0ecdf2d26b1d9d8cf	IPv4	SSH	TCP	22	0.0.0.0/0	-

Create a new rule for the **NFS protocol**, set the **source** to the **ELB Security Group**, and then click on the **Save Rules** button.

The screenshot shows the 'Edit inbound rules' dialog box. A new rule is being added, highlighted with a red box. The rule details are as follows:

Security group rule ID	Type	Protocol	Port range	Source	Description
sgr-0f7fa9205dbd2ca85	HTTP	TCP	80	Custom (84.32.10.4/32)	-
sgr-0dd8d2052de349965	HTTP	TCP	80	Custom (sg-0966652eeeb10ac22)	-
sgr-0ecdf2d26b1d9d8cf	SSH	TCP	22	Custom (0.0.0.0/0)	-
-	NFS	TCP	2049	Custom (sg-0ed4d08504a257a3e, sg-0ed4d08504a257a3e - SG-ELB)	-

A warning message at the bottom states: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." The 'Save rules' button is highlighted with a red box.

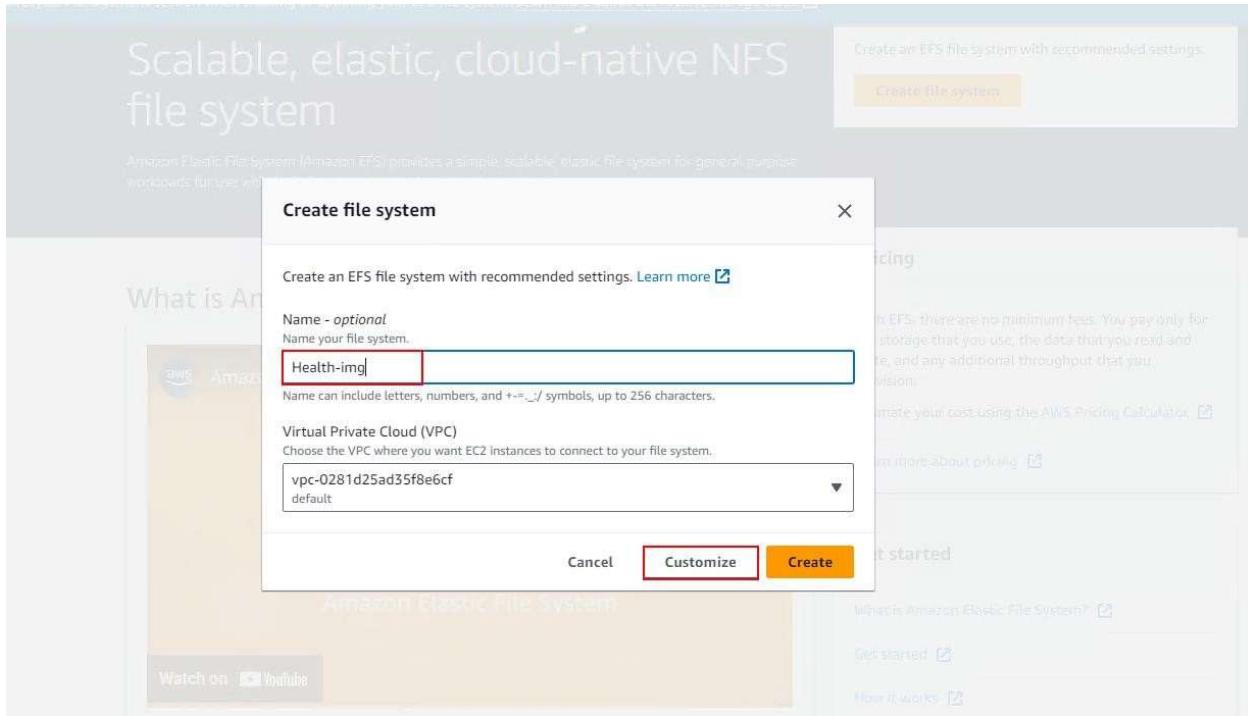
In the search bar, type **EFS**, then click on **EFS** from the results.

The screenshot shows the AWS Services search results for 'EFS'. The 'Services' section is highlighted, and the 'EFS' item is selected, indicated by a red box. The 'Features' section also lists EFS under 'File systems'.

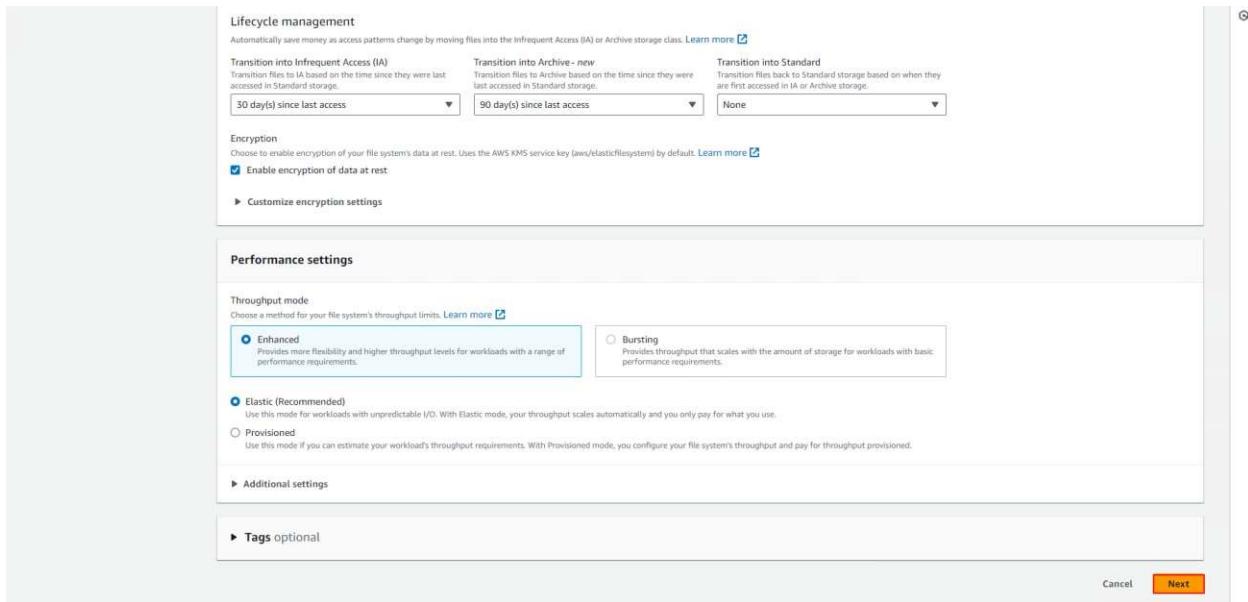
At this stage, click on the **Create File System** button.

The screenshot shows the Amazon Elastic File System (EFS) landing page. The main heading is 'Scalable, elastic, cloud-native NFS file system'. A prominent orange 'Create file system' button is visible. To the right, there are sections for 'Pricing', 'Get started', and 'More resources'.

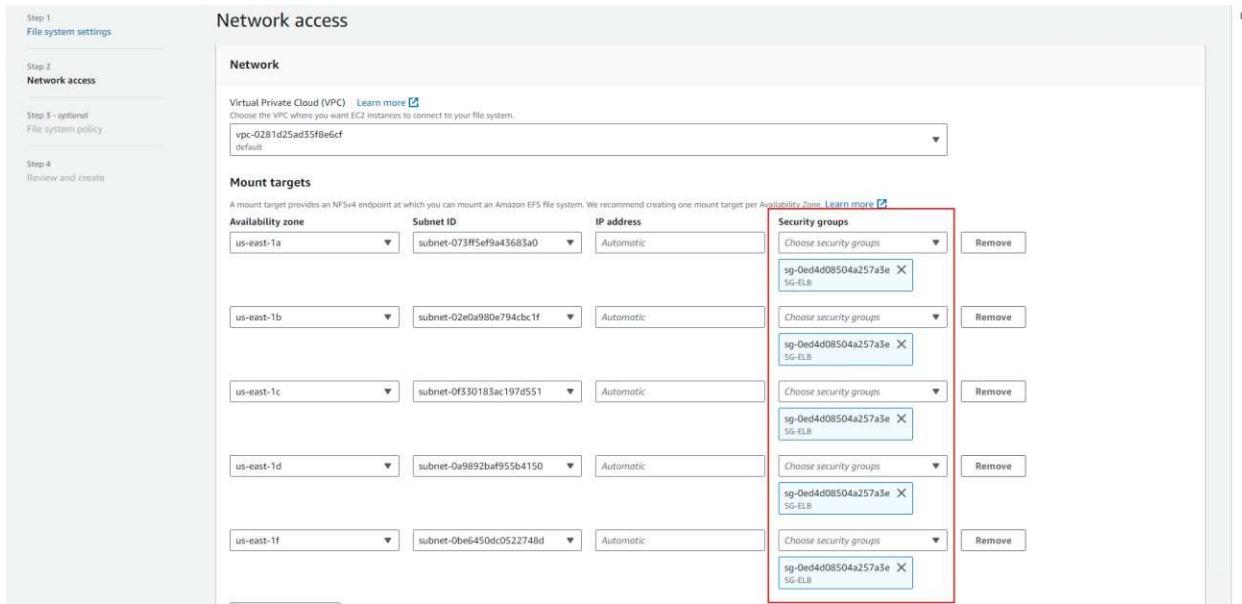
At this stage, choose a **name** for the EFS and select the **VPC**, then click on the **Customize** button.



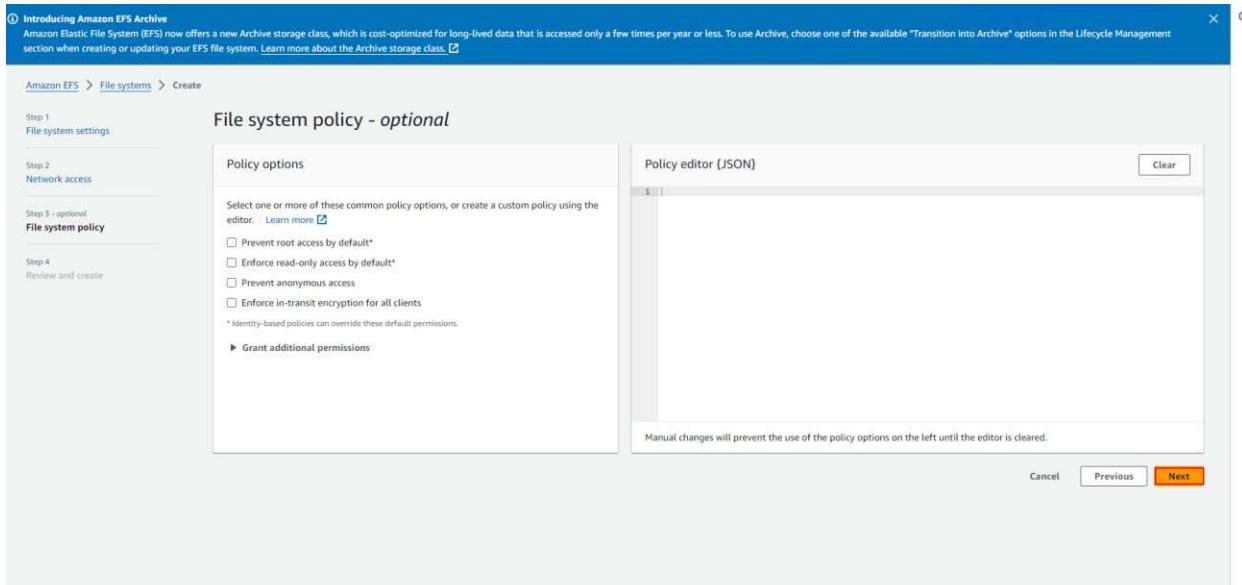
At this stage, click on the **Next** button.



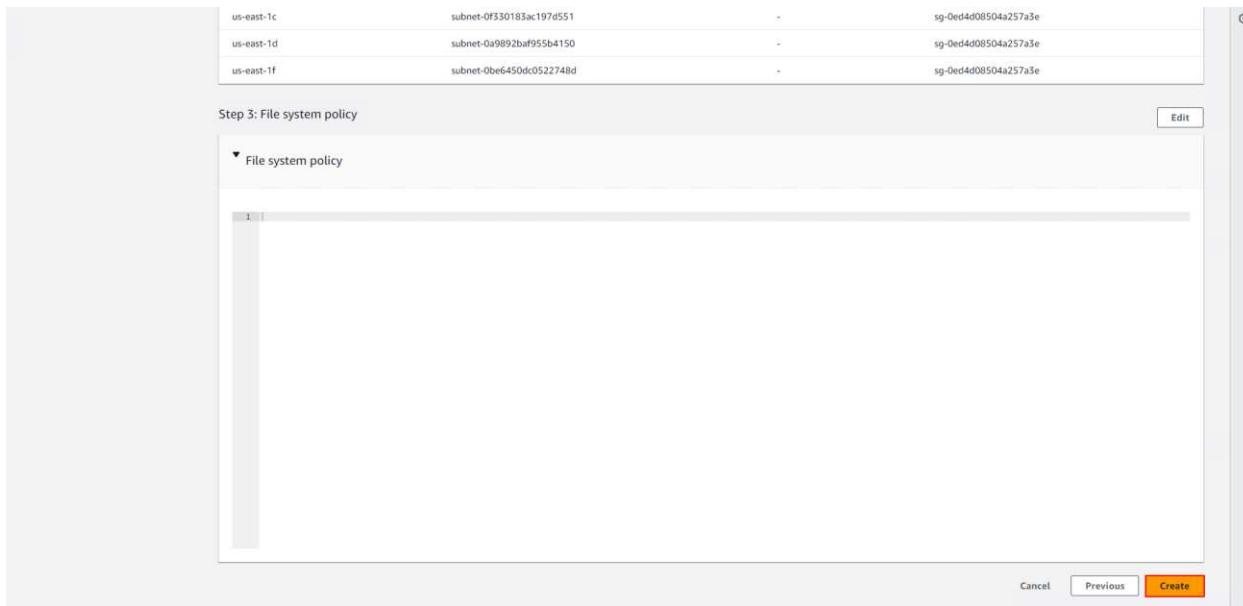
At this stage, set the **Security Group** for all **Availability Zones** to the **ELB Security Group**.



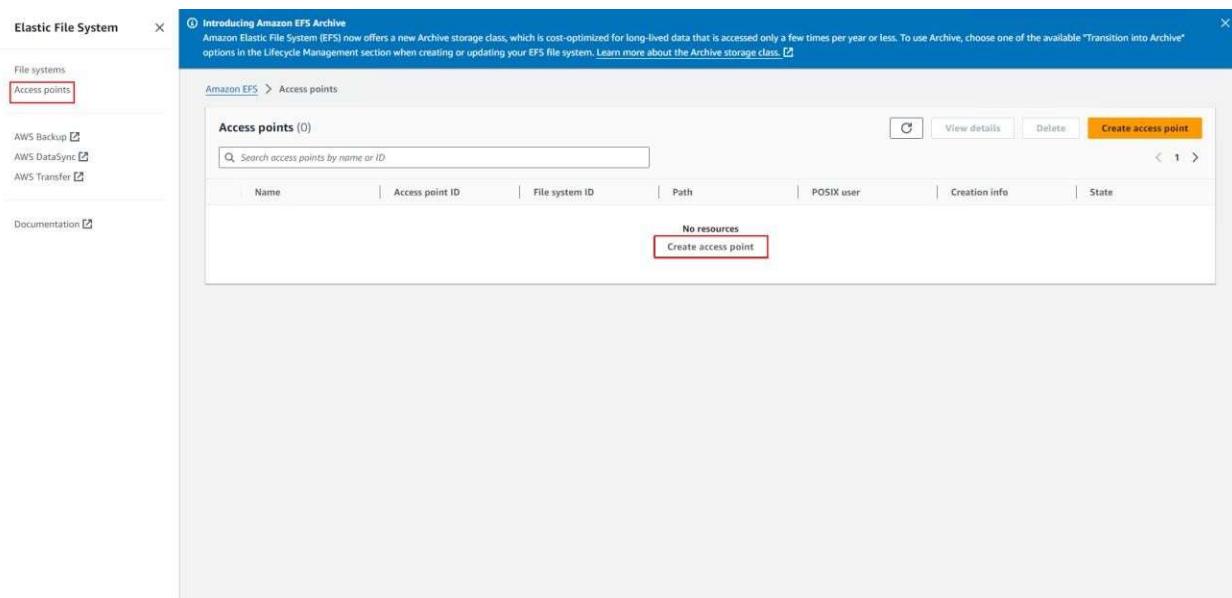
At this stage, click on the **Next** button.



At this stage, click on the **Create** button.



At this stage, click on the **Access Points** section, then click the **Create Access Point** button.



At this stage, you need to select the **EFS file system** that you created in the previous step.

Create access point

An access point is an application-specific entry point into an EFS file system that makes it easier to manage application access to shared datasets. [Learn more](#)

Details

File system

Choose the file system to which your access point is associated.

 fs-dada5@72f4c749ec X

Name - optional

EFS access point name

Name can include letters, numbers, and `-`/`_`/ symbols, up to 256 characters.

Root directory path - optional

Connections use the specified path as the file system's virtual root directory. [Learn more](#)

Defaults to /

Example: "/foo/bar"

POSIX user - optional

The full POSIX identity on the access point that is used for all file operations by NFS clients. [Learn more](#)

User ID

POSIX user ID used for all file system operations using this access point.

 POSIX UID Accepts values from 0 to 4294967295

Then, at this stage, click on the **Create Access Point** button.

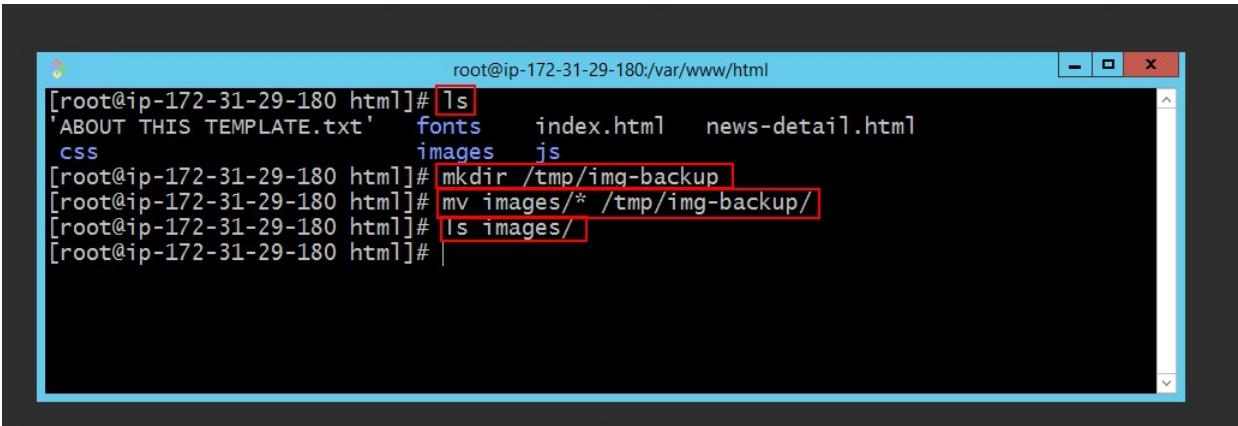
The screenshot shows the 'Root directory creation permissions - optional' section. It includes fields for Owner user ID (a dropdown menu with 'POSIX user ID to apply to path' and a note about accepting values from 0 to 4294967295), Owner group ID (a dropdown menu with 'POSIX group ID to apply to path' and a note about accepting values from 0 to 4294967295), and Access point permissions (a text input field with 'Example: "0755"' and a note about being an octal number). Below this is the 'Tags - optional' section, which allows adding key-value pairs. A table shows a single tag entry: 'Tag key' (empty) and 'Tag value - optional' (empty). Buttons for 'Add tag' and 'Remove tag' are present. At the bottom are 'Cancel' and 'Create access point' buttons.

At this stage, you need to enter the **terminal of your EC2 instance**, and then install the following package to enable access to **EFS**

A terminal window titled 'root@ip-172-31-29-180:~\$'. The command 'sudo yum install -y amazon-efs-utils' is entered and executed. The output shows the package being installed and its dependencies. The transaction summary at the bottom indicates the installation of 'amazon-efs-utils' (version 1.35.2-1.amzn2023, size 55 k) and 'stunnel' (version 5.58-1.amzn2023.0.2, size 156 k).

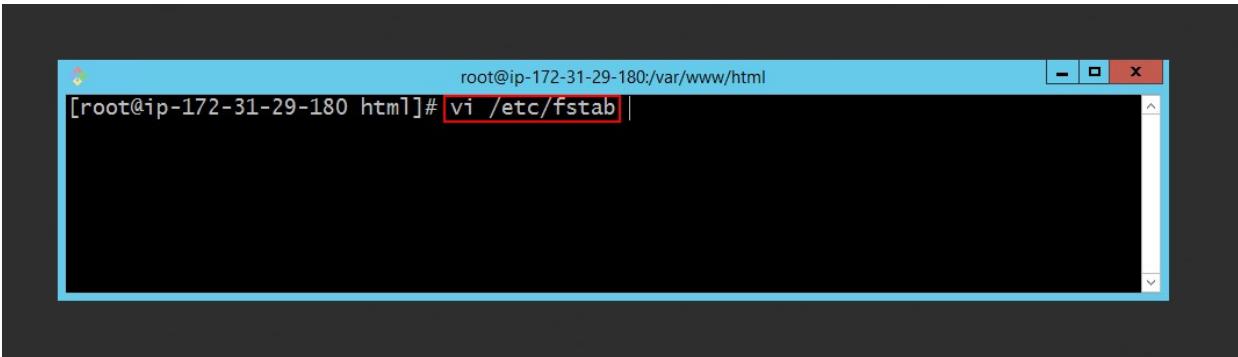
```
root@ip-172-31-29-180:~$ sudo yum install -y amazon-efs-utils
Last metadata expiration check: 1:06:30 ago on Sat Mar 23 08:04:13 2024.
Dependencies resolved.
=====
 Package           Arch      Version       Repository      Size
=====
 Installing:
  amazon-efs-utils    noarch   1.35.2-1.amzn2023  amazonlinux      55 k
 Installing dependencies:
  stunnel            x86_64   5.58-1.amzn2023.0.2  amazonlinux    156 k
Transaction Summary
```

Before mounting **EFS** to the **EC2 instance**, first create a **backup** of the **Images** folder in the web server path



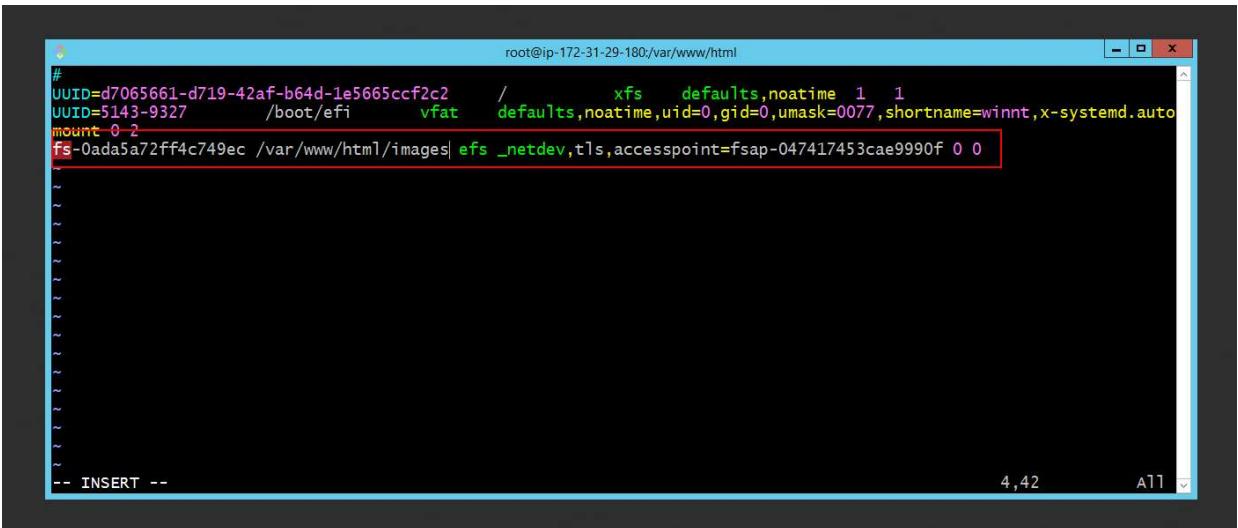
```
root@ip-172-31-29-180:~# ls
'ABOUT THIS TEMPLATE.txt'  fonts  index.html  news-detail.html
css                      images  js
[root@ip-172-31-29-180 ~]# mkdir /tmp/img-backup
[root@ip-172-31-29-180 ~]# mv images/* /tmp/img-backup/
[root@ip-172-31-29-180 ~]# ls images/
[root@ip-172-31-29-180 ~]#
```

To mount the **EFS**, open the **fstab** file using the following command



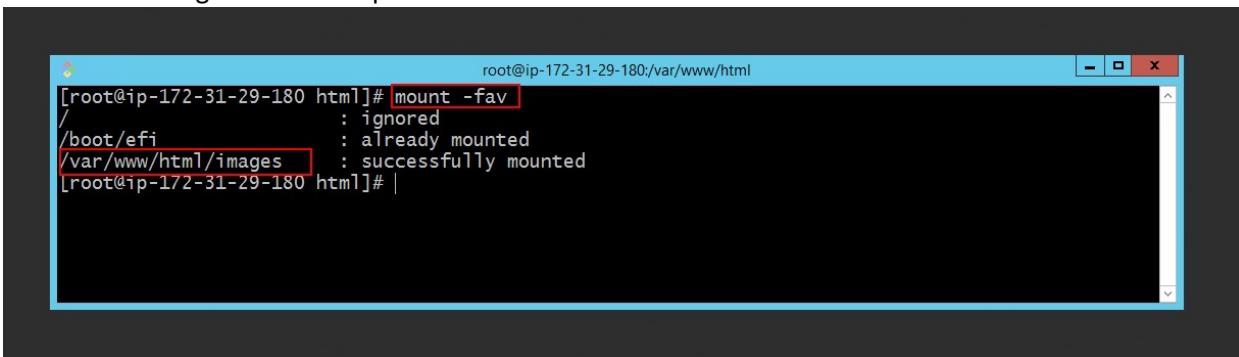
```
root@ip-172-31-29-180:~# vi /etc/fstab |
```

Add the following line to the **fstab** file, then save the file



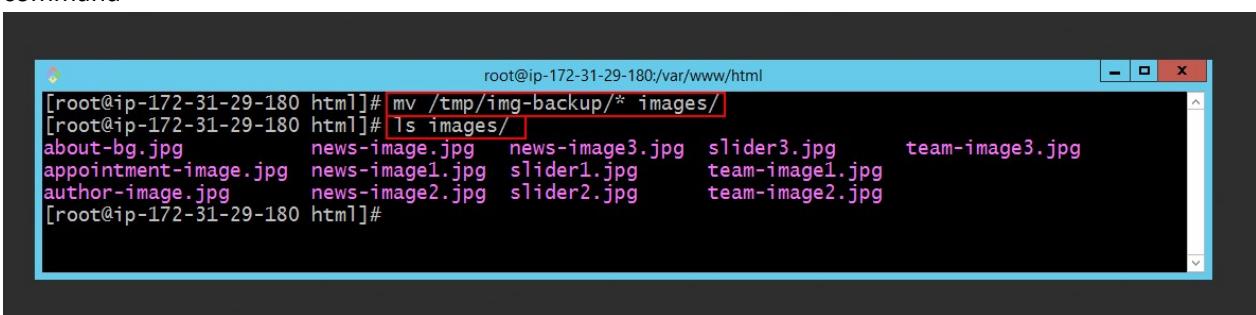
```
#  
UUID=d7065661-d719-42af-b64d-1e5665ccf2c2      /          xfs  defaults,noatime 1  1  
UUID=5143-9327        /boot/efi    vfat   defaults,noatime,uid=0,gid=0,umask=0077,shortname=winnt,x-systemd.mount=0 2  
fs-0ada5a72ff4c749ec /var/www/html/images| efs _netdev,tls,accesspoint=fsap-047417453cae9990f 0 0  
~  
~  
~  
~  
~  
~  
~  
~  
~  
-- INSERT --
```

Use the following command to perform the mount



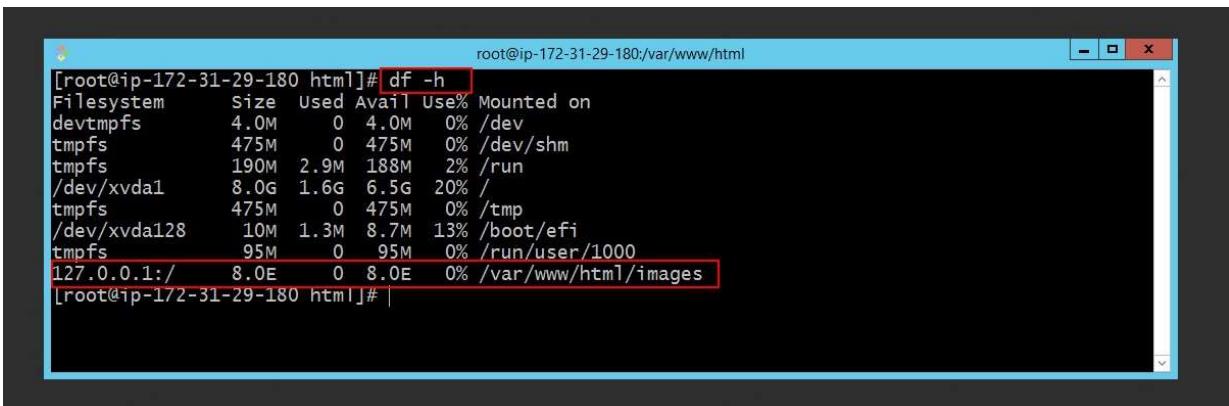
```
root@ip-172-31-29-180 html]# mount -fav
/                         : ignored
/boot/efi                  : already mounted
/var/www/html/images        : successfully mounted
[root@ip-172-31-29-180 html]# |
```

Then, move the files from the **Images** folder backup back to the **web server path** using the following command



```
root@ip-172-31-29-180 html]# mv /tmp/img-backup/* images/
[root@ip-172-31-29-180 html]# ls images/
about-bg.jpg      news-image.jpg   news-image3.jpg   slider3.jpg    team-image3.jpg
appointment-image.jpg news-image1.jpg slider1.jpg    team-image1.jpg
author-image.jpg   news-image2.jpg   slider2.jpg    team-image2.jpg
[root@ip-172-31-29-180 html]#
```

Using the following command, you can verify that the **Images** folder is mounted to your EC2 instance and is now located on the **EFS**



```
root@ip-172-31-29-180 html]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0  4.0M  0% /dev
tmpfs          475M   0  475M  0% /dev/shm
tmpfs          190M  2.9M  188M  2% /run
/dev/xvda1       8.0G  1.6G  6.5G  20% /
tmpfs          475M   0  475M  0% /tmp
/dev/xvda128     10M  1.3M  8.7M  13% /boot/efi
tmpfs          95M   0   95M  0% /run/user/1000
127.0.0.1:/     8.0E   0  8.0E  0% /var/www/html/images
[root@ip-172-31-29-180 html]# |
```

Introduction to AWS Auto Scaling

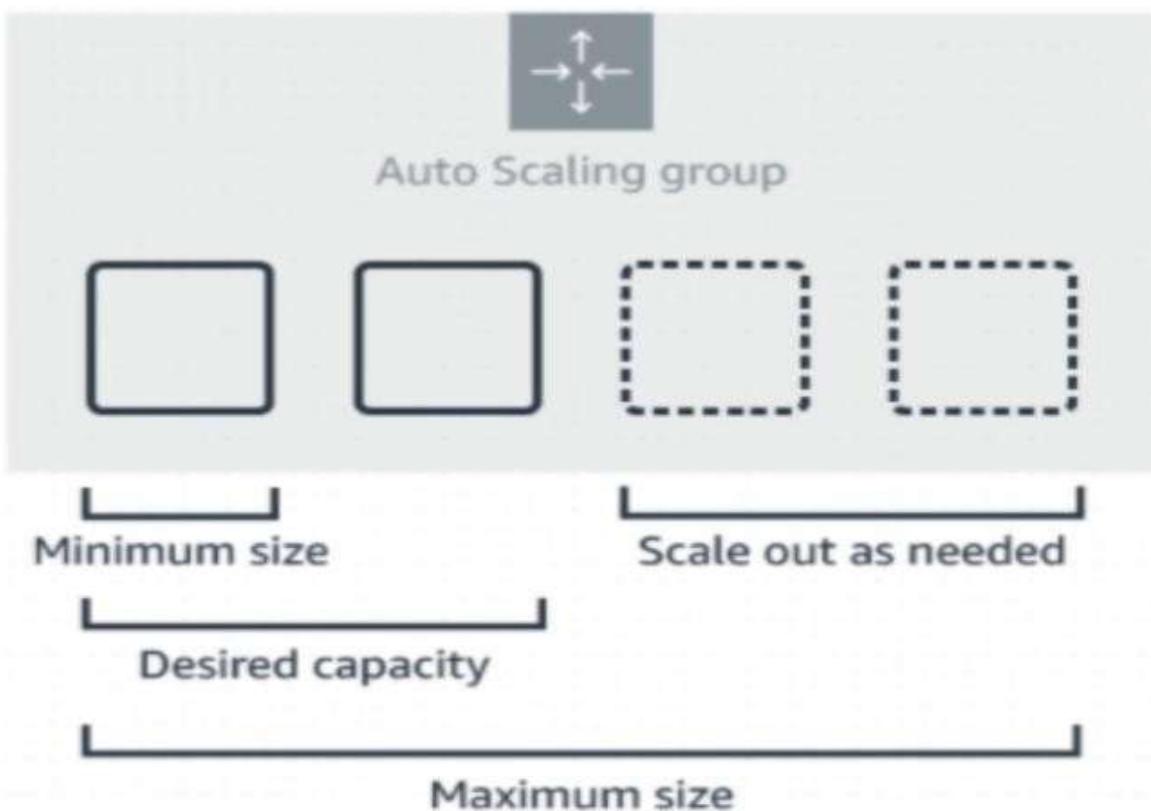
Auto Scaling is a service that can automatically **monitor** and **adjust compute resources** to improve the **performance** of applications hosted on your **AWS** environment.

For example, **Auto Scaling** uses **CloudWatch** to monitor the **CPU usage** of an EC2 instance, and if the CPU usage exceeds the defined **threshold**, it can automatically add one or more instances to the **Auto Scaling Group** to increase performance.

Auto Scaling uses a **Launch Configuration** or **Launch Template** to launch an instance.

A **Scaling Policy** is used to **adjust the capacity**.

An **Auto Scaling Group** contains its own **policies**. For example, a policy can specify that if **CPU usage exceeds 60%**, launch **2 instances**, and if it exceeds **80%**, launch **4 instances**. Alternatively, you can use the **Auto Scaling Group's auto settings** to manage all of this automatically for you.

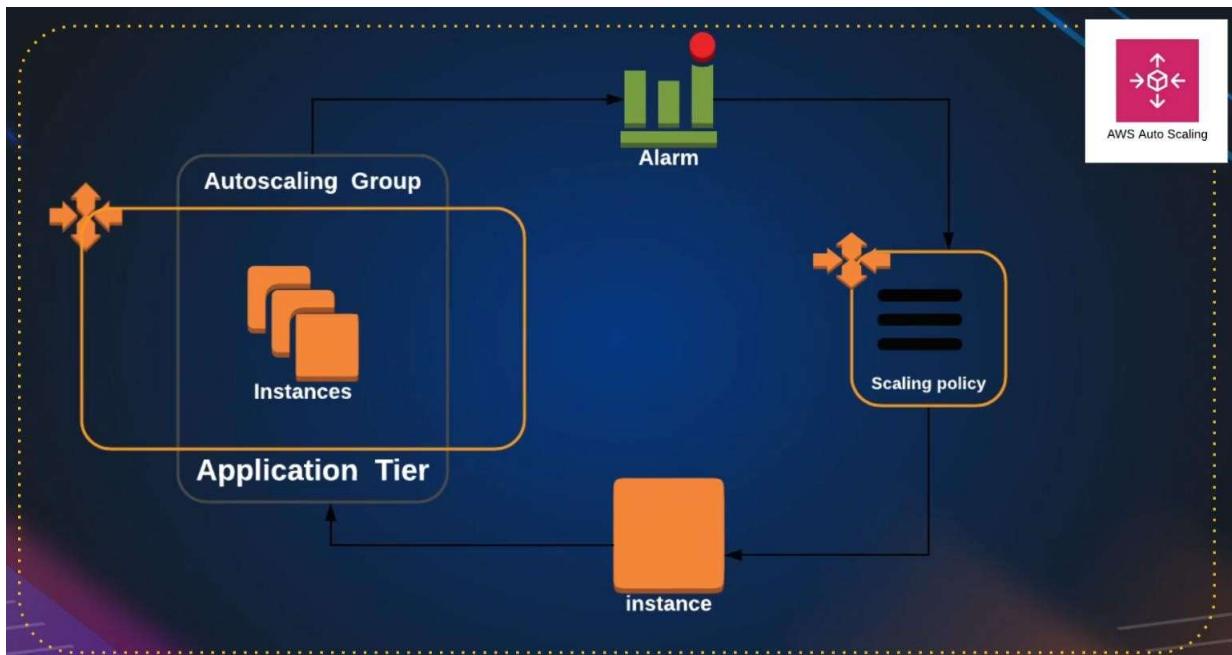


When you create **Auto Scaling**, you can define the **Minimum Size** of the Auto Scaling Group. For example, you can specify **1 instance** as the **minimum instance** in the Auto Scaling Group.

You can also specify the **Desired Capacity**. For example, if you set the **Desired Capacity** to 2 and the **Minimum** to 1, it will launch **2 instances** in the Auto Scaling Group. In other words, there must be **at least one instance** running in the Auto Scaling Group.

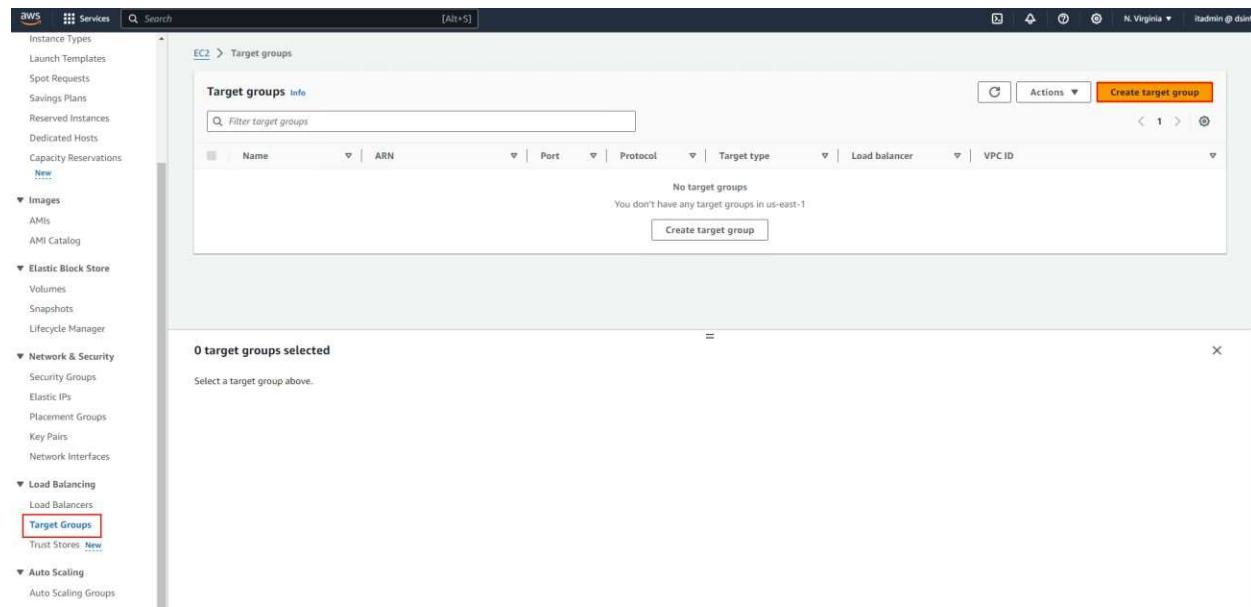
Therefore, when the **Minimum Instance** is set to **1**, during termination, at least **one instance** will remain in the group, and the rest will be terminated. As a result, you'll always have the number of **Desired Instances** running in your **Auto Scaling Group**.

If you set the **Maximum Instance** to 4, the number of instances added will **not exceed 4**.



How to Create an AWS Auto Scaling Group

At this stage, first click on the **Target Groups** section, and then click on the **Create Target Group** button.



In this section, specify a **name** for the Target Group.

A screenshot of the 'Create Target Group' configuration dialog. The 'Application Load Balancer' type is selected. The 'Target group name' field contains 'health-tg' and is highlighted with a red box. The 'Protocol' dropdown is set to 'HTTP' and the 'Port' dropdown is set to '80'. The 'IP address type' section shows 'IPv4' selected. The 'VPC' section lists 'vpc-0281d25ad55f8e6cf' with 'IPv4 VPC CIDR: 172.31.0.0/16'. The 'Protocol version' section has 'HTTP1' selected. The right side of the dialog is mostly empty.

At this stage, click on the **Next** button.

Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol
HTTP

Health check path
Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.
/

Up to 1024 characters allowed.

► Advanced health check settings

Attributes
Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

► Tags - optional
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel **Next**

Then, at this stage, click on the **Create Target Group** button.

Step 2
Register targets

Available instances (1)

Instance ID	Name	State	Security groups	Zone	Private IPv4 address
i-0393518d57102b71a	web	Running	launch-wizard-3	us-east-1c	172.31.37.125

0 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with comma)

Include as pending below

Review targets

Targets (0)

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
No instances added yet Specify instances above, or leave the group empty if you prefer to add targets later.								

0 pending

Cancel Previous **Create target group**

At this stage, click on the **Load Balancers** section, then click on the **Create Load Balancer** button.

The screenshot shows the AWS EC2 console with the 'Load balancers' section selected. On the left, a sidebar lists various services like Instance Types, Images, and Network & Security. The 'Load Balancing' section is expanded, and 'Load Balancers' is selected. At the top right, there is a 'Create load balancer' button. Below it, a message says 'No load balancers' and 'You don't have any load balancers in us-east-1'. A search bar and filter options are also present.

At this stage, select **Application Load Balancer**, and then click the **Create** button.

The screenshot shows the 'Compare and select load balancer type' page. It compares three types: Application Load Balancer, Network Load Balancer, and Gateway Load Balancer. Each section includes a diagram and a 'Create' button. The 'Application Load Balancer' section is highlighted with a red border.

Load balancer type	Description	Action
Application Load Balancer	Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.	Create
Network Load Balancer	Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.	Create
Gateway Load Balancer	Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.	Create

At this stage, specify a **name** for the Load Balancer.

The screenshot shows the 'Create Application Load Balancer' wizard on the 'Basic configuration' step. The 'Load balancer name' field is filled with 'Health-ELB'. The 'Scheme' section is set to 'Internet-facing'. The 'IP address type' is 'IPv4'. The 'Network mapping' section indicates traffic will be routed to targets in selected subnets.

At this stage, select **all Availability Zones** to increase availability.

The screenshot shows the 'Configure routing rules' step of the wizard. Under the 'Add target group' section, six availability zones are selected: us-east-1a (use1-az2), us-east-1b (use1-az4), us-east-1c (use1-az6), us-east-1d (use1-az1), us-east-1e (use1-az3), and us-east-1f (use1-az5). Each selection includes a dropdown for subnet selection.

At this stage, select the **SG-ELB** as the **Security Group**, and in the **Listener** section, set the **Default Action** to the **Target Group** you defined in the previous step.

The screenshot shows the 'Listeners and routing' section of the AWS Load Balancer configuration. Under 'Protocol' (HTTP) and 'Port' (80), the 'Default action' is set to 'health-tg'. A red box highlights this configuration. Below it, there's a 'Create target group' button. The 'Listener tags - optional' section contains an 'Add listener tag' button. At the bottom left is an 'Add listener' button.

And then, at this stage, click on the **Create Load Balancer** button.

The screenshot shows the final review step of the load balancer creation wizard. It includes sections for 'Basic configuration', 'Security groups', 'Network mapping', 'Listeners and routing', 'Service integrations', 'Tags', 'Attributes', and 'Creation workflow and status'. The 'Listeners and routing' section shows the configuration from the previous screenshot. The 'Create load balancer' button is highlighted in orange at the bottom right.

At this stage, click on the **Auto Scaling Groups** section, and then click on the **Create Auto Scaling Group** button.

The screenshot shows the Amazon EC2 Auto Scaling homepage. On the left, there is a navigation sidebar with various AWS services listed under 'EC2'. The 'Auto Scaling Groups' option is highlighted with a red border. The main content area features a large heading 'Amazon EC2 Auto Scaling' and a sub-headline 'helps maintain the availability of your applications'. Below this is a brief description of what Auto Scaling groups are. To the right, there is a 'Create Auto Scaling group' button. Further down, there are sections titled 'How it works' (with a diagram showing an 'Auto Scaling group' containing four instances, with one dashed instance labeled 'Scale out as needed'), 'Pricing' (describing service fees), and 'Getting started' (with links to 'What is Amazon EC2 Auto Scaling?', 'Getting started with Amazon EC2 Auto Scaling', and 'Set up a scaled and load-balanced application'). At the bottom, there is a footer with copyright information and links to 'CloudShell', 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences'.

At this stage, specify a **name** for the Auto Scaling Group, and in the **Launch Template** section, select the **template** you created earlier.

The screenshot shows the 'Choose launch template' step in the 'Create Auto Scaling group' wizard. On the left, there is a sidebar with steps 1 through 7. Step 1 is 'Choose launch template', which is highlighted with a red border. The main content area has two sections: 'Name' and 'Launch template'. In the 'Name' section, there is a field labeled 'Auto Scaling group name' with the value 'Health-ASG'. In the 'Launch template' section, there is a note about accounts created after May 31, 2023. Below this, there is a dropdown menu set to 'Health-Template' and a 'Create a launch template' button. At the bottom, there are fields for 'Description' (V1), 'Launch template' (Health-Template), 'Instance type' (t2.micro), 'AMI ID' (lt-00a9981ad18293377), 'Security groups' (Request Spot Instances), and 'Request Spot Instances' (checkbox). A 'Version' dropdown is also present.

At this stage, click on the **Next** button.

The screenshot shows the AWS Launch Template configuration interface. On the left, a sidebar lists steps: Step 5 - optional (Add notifications), Step 6 - optional (Add tags), and Step 7 - Review. The main area is titled "Launch template" and contains the following details:

- Launch template:** Health-Template (Info)
- Description:** V1
- AMI ID:** ami-06e01697fb4f3ff5b
- Key pair name:** key-pair
- Health-Template:** lt-00a9981ad18293577
- Instance type:** t2.micro
- Security groups:** -
- Request Spot Instances:** No
- Storage (volumes):** -
- Date created:** Sat Mar 23 2024 08:34:49 GMT-0700 (Pacific Daylight Time)

At the bottom right are "Cancel" and "Next" buttons.

At this stage, select **all Availability Zones** to enhance availability.

The screenshot shows the "Choose instance launch options" step. The sidebar lists steps: Step 1: Choose launch template, Step 2: Choose instance launch options (highlighted), Step 3 - optional: Configure advanced options, Step 4 - optional: Configure group size and scaling, Step 5 - optional: Add notifications, and Step 6 - optional: Add tags.

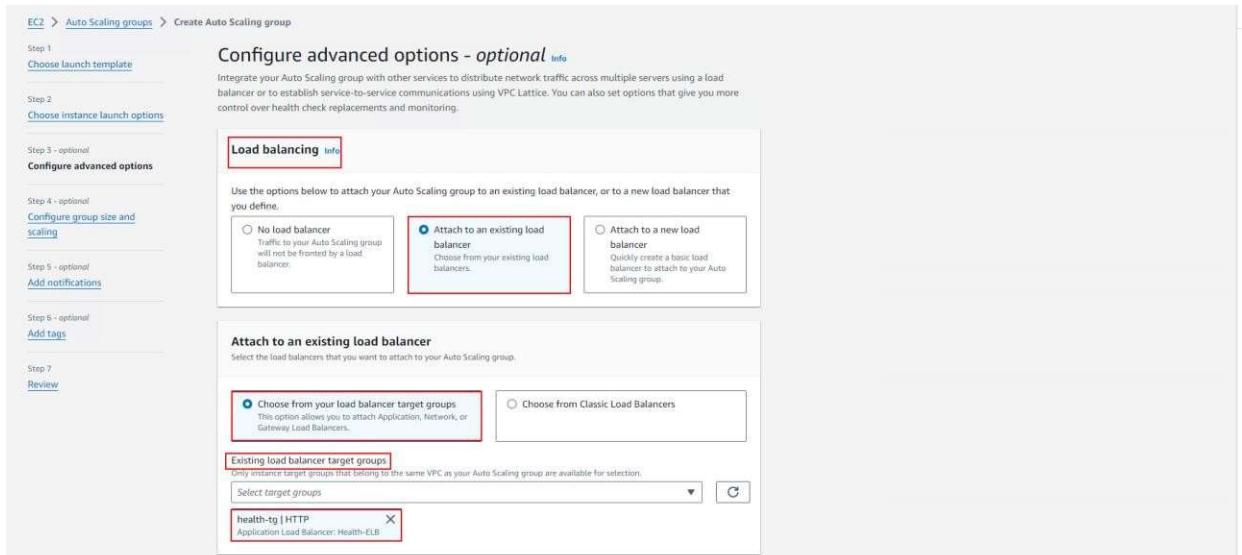
The main area is titled "Choose instance launch options" and includes:

- Instance type requirements:** Info
- Launch template:** Health-Template (Info)
- Version:** Default
- Description:** V1
- Instance type:** t2.micro
- Network:** Info
- A list of subnets:
 - us-east-1a | subnet-073ff5ef9a43683a0 (selected)
 - us-east-1b | subnet-02e0a980e794cbc1f (selected)
 - us-east-1c | subnet-0f330183ac197d551 (selected)
 - us-east-1d | subnet-0a9892bf955b4150 (selected)
 - us-east-1e | subnet-05af0b7eee86bb7b6 (selected)
 - us-east-1f | subnet-0be6450dc0522748d (selected)

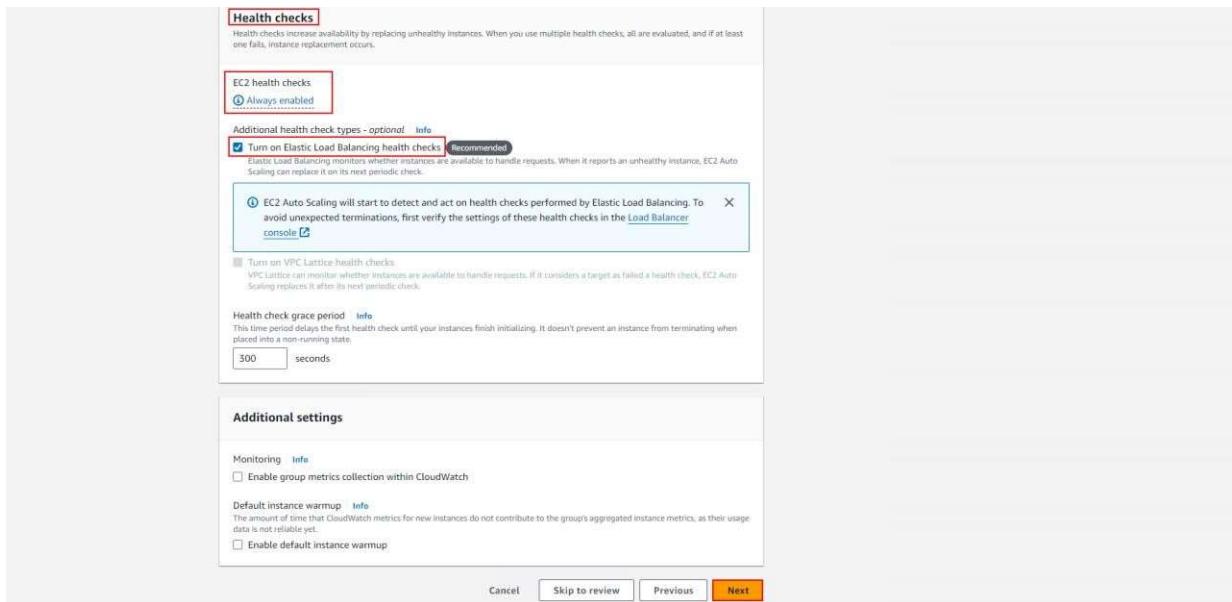
Then, at this stage, click on the **Next** button.



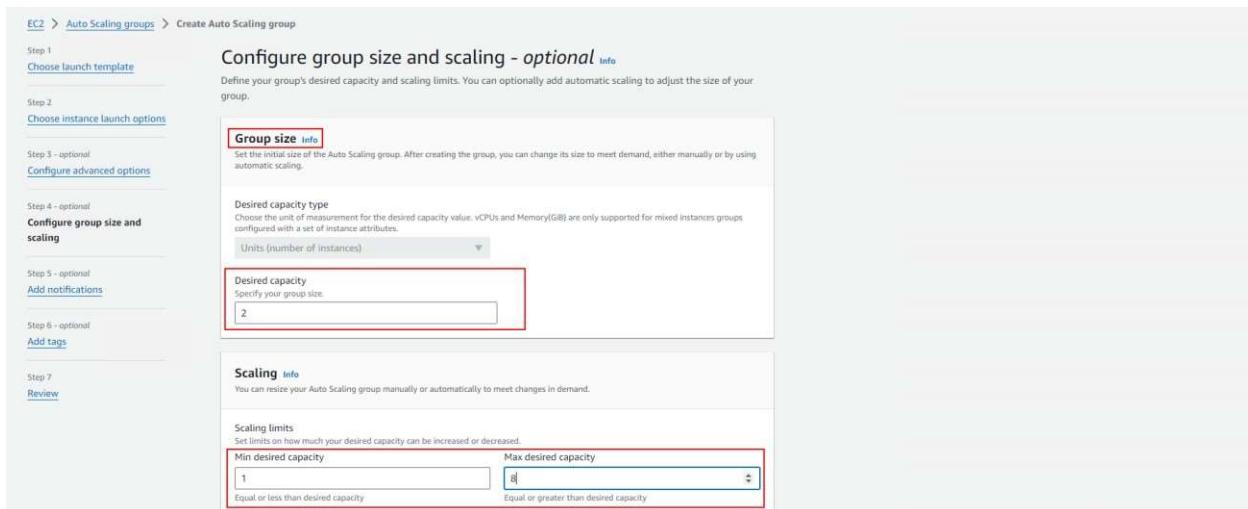
At this stage, in the **Load Balancing** section, select the option **Attach to an existing load balancer**, and then specify the **Target Group**.



At this stage, select the option shown below, and then click on the **Next** button.



At this stage, you need to configure the **Group Size** settings, as shown in the image below.



At this stage, select the **Target Tracking Scaling Policy** option, then set the **Metric Type** to **Average CPU Utilization**, and set the **Target Value** to **50%**. This means that if the average CPU utilization exceeds 50%, Auto Scaling will increase the

number of instances up to the **Maximum Value**, and if it drops to 50% or below, it will reduce the instances back to the **Desired Value**.

Automatic scaling - optional Info
Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

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

Scaling policy name: Target Tracking Policy

Metric type: Info
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.
Average CPU utilization

Target value: 50

Instance warmup: Info
300 seconds

Disable scale in to create only a scale-out policy

At this stage, click on the **Next** button.

Instance maintenance policy - new Info
Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.

Control availability and cost during replacement events
An instance maintenance policy determines how much availability your application has when EC2 Auto Scaling replaces instances. It also establishes guardrails that limit the amount of capacity that can be added or removed when replacing instances.

Choose a replacement behavior depending on your availability requirements

Mixed behavior
For rebalancing events, new instances will launch before terminating others. For all other events, instances terminate and launch at the same time.

Prioritize availability
Launch before terminating
Launch new instances and wait for them to be ready before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.

Control costs
Terminate and launch
Terminate and launch instances at the same time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.

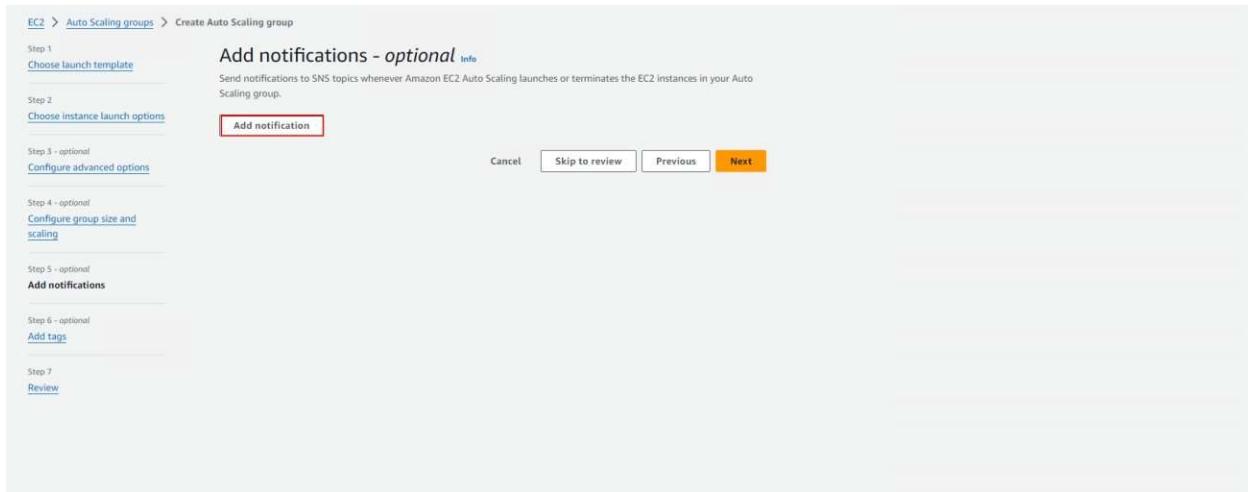
Finalize
Custom behavior
Set custom values for the minimum and maximum amount of available capacity. This gives you greater flexibility in setting how far above and over your desired capacity EC2 Auto Scaling goes when replacing instances.

Instance scale-in protection
Scale-in protection prevents newly launched instances from being terminated by scaling activities. Make sure to remove scale-in protection for the group or individual instances when instances are ready to be terminated.

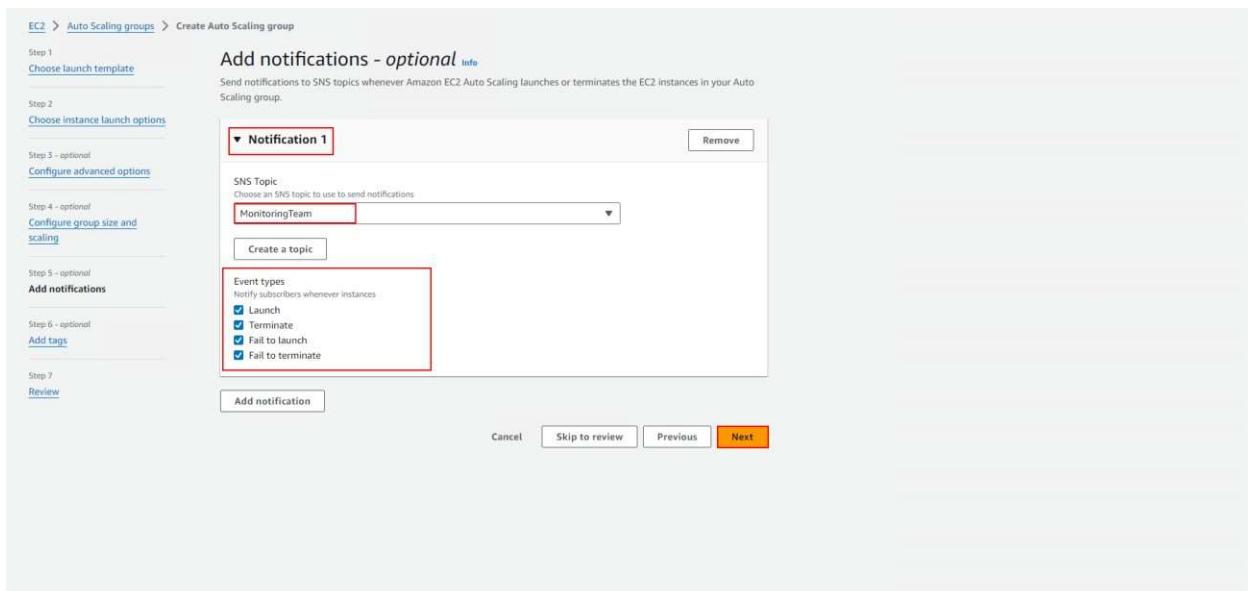
Enable instance scale-in protection

Cancel Skip to review Previous Next

At this stage, click on the **Add Notification** button.



At this stage, select the **SNS Topic** and **Event Type**, then click on the **Next** button.



At this stage, define a **Tag** for the Auto Scaling Group, and then click on the **Next** button.

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1 Choose launch template

Step 2 Choose instance launch options

Step 3 - optional Configure advanced options

Step 4 - optional Configure group size and scaling

Step 5 - optional Add notifications

Step 6 - optional Add tags

Step 7 Review

Add tags - optional Info

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

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

Tags (1)

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

Add tag Remove

Cancel Previous Next

At this stage, click on the **Create Auto Scaling Group** button.

Instance maintenance policy

Replacement behavior: No policy

Min healthy percentage: -

Max healthy percentage: -

Instance scale-in protection

Instance scale-in protection: Enable instance protection from scale in

Step 5: Add notifications Edit

Notifications

Notification 1	Event types
SNS Topic MonitoringTeam	<input checked="" type="checkbox"/> Launch <input checked="" type="checkbox"/> Terminate <input checked="" type="checkbox"/> Fail to launch <input checked="" type="checkbox"/> Fail to terminate

Step 6: Add tags Edit

Tags (1)

Key	Value	Tag new instances
Name	webserver	Yes

Cancel Previous **Create Auto Scaling group**

In the **Activity** section of the **Auto Scaling Group**, you can see that **two instances** have been created.

The screenshot shows the 'Activity' tab of the Auto Scaling group 'Health-ASG'. It displays two entries in the 'Activity notifications' and 'Activity history' sections, both detailing the launch of new EC2 instances. The first entry is for instance ID 'i-09eb3e06c1a6f7e40' and the second for 'i-07ad2b847f7743d37'. Both entries show the cause as 'Launching a new EC2 instance' and the start time as '2024 March 23, 09:07:22 AM -07:00'.

In the **Instance Management** section of the **Auto Scaling Group**, you can view their **Health Status**.

The screenshot shows the 'Instance management' tab of the Auto Scaling group 'Health-ASG'. It lists two instances: 'i-07ad2b847f7743d37' and 'i-09eb3e06c1a6f7e40', both marked as 'InService' with 't2.micro' instance type and 'us-east-1c' availability zone. The 'Health status' column shows green circles indicating they are healthy. The 'Lifecycle hooks' section is empty, and the 'Warm pool' section also indicates no warm pool is currently configured.

And in the **Instances** section, you can view the **instances created by Auto Scaling**.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like EC2 Dashboard, Events, and Instances. The Instances section is expanded, showing a table of running instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 IP, and Elastic IP. Four instances are listed, all labeled 'webserver' and 't2.micro'. The third instance from the top is highlighted with a red border. Below the table, there's a section titled 'Select an instance'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP
web	i-0393318d37102b71a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-54-84-117-56.com...	54.84.117.56	-
webserver	i-0cf889193ba0c092e	Running	t2.micro	Initializing	View alarms +	us-east-1a	ec2-3-84-21-70.compute...	3.84.21.70	-
webserver	i-07ad2b847f7743d37	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-3-89-132-84.comp...	3.89.132.84	-
webserver	i-09eb3e06c1a6f7fe40	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-3-87-248-241.com...	3.87.248.241	-

Introduction to Amazon S3

The **S3 service**, short for **Simple Storage Service**, is a **storage service** that is **very easy to use**.

The **S3 service** is one of the **most popular and oldest services** offered by AWS.

The **S3 service** is an **internet-based storage service**. You can use **S3 to store** and retrieve data from **anywhere**, at **any time**.

You can think of the **S3 service** as being similar to **Google Drive** or **Dropbox**, or even **more powerful** than them.

S3 is an **Object Storage** service where you can upload any type of file, such as **documents**, **pictures**, or **videos**, and access them from **anywhere**.

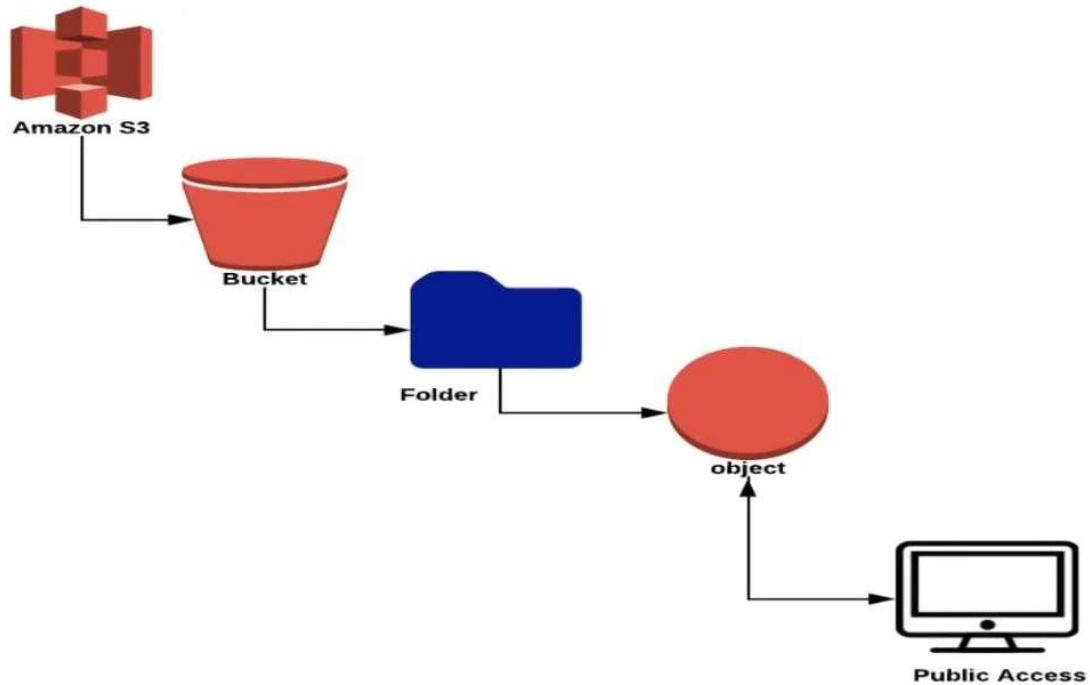
When you upload any data to your **S3 bucket**, the **bucket** acts as a **top-level storage container**—similar to a **folder** in S3.

Your data is **replicated across multiple S3 systems**, and there are **no limits** on storage.

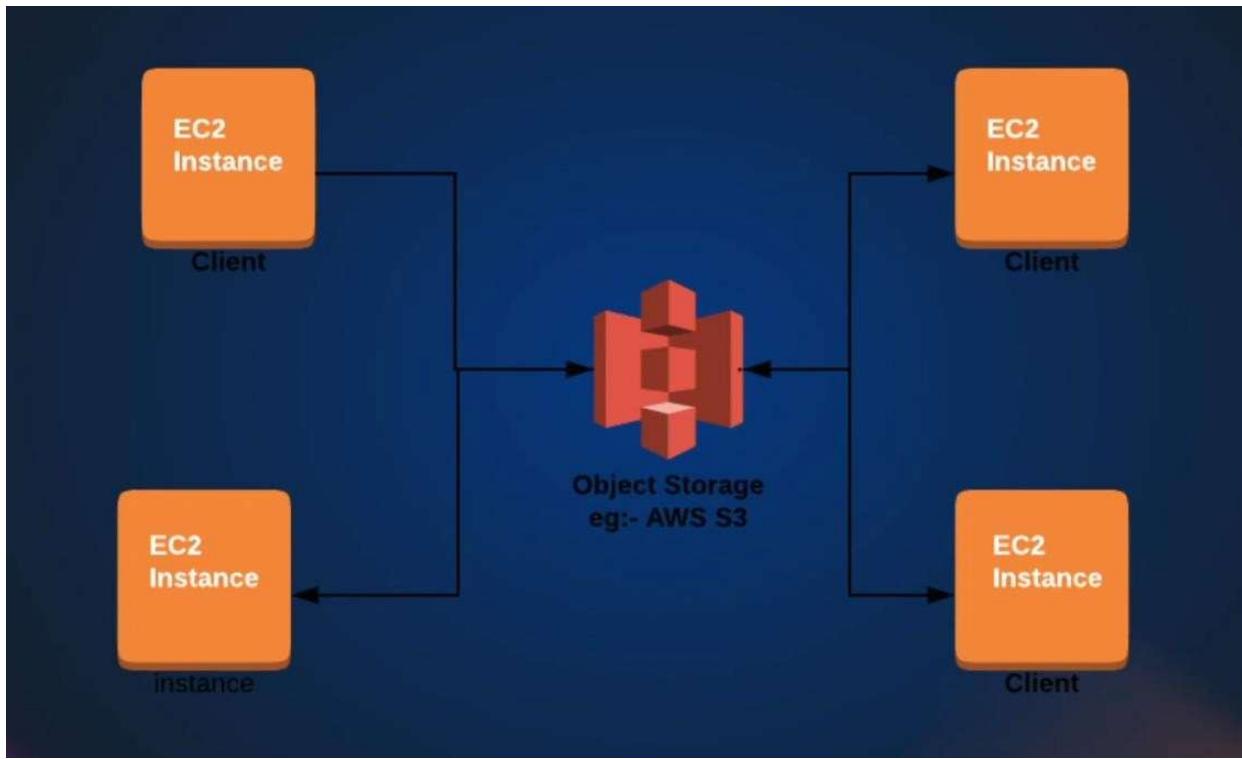
The **S3 service** offers **unlimited storage**, allowing you to store **as much data as you want**.

The data stored in S3 is referred to as an **Object**, and the storage container itself is called a **Bucket**.

The **bucket name** must be **unique**.



One common use of an **S3 bucket** is when you want to **store data** from your **EC2 instances** on **S3**.



Types of S3 Storage Classes:

1. **S3 Standard**
 - Designed for **frequent access**
 - High durability and availability
 - Suitable for general-purpose storage
2. **S3 Intelligent-Tiering**
 - Automatically moves data between two tiers (frequent and infrequent access)
 - Optimizes costs without performance impact
3. **S3 Standard-IA (Infrequent Access)**
 - For data that is accessed **less frequently**, but still requires rapid access
 - Lower storage cost, higher retrieval cost
4. **S3 One Zone-IA**
 - Similar to Standard-IA but stored in a **single Availability Zone**
 - Lower cost, but less resilient to AZ failure
5. **S3 Glacier**
 - For **archival** and long-term backups

- Retrieval time from **minutes to hours**
- 6. **S3 Glacier Deep Archive**
 - **Lowest-cost** storage class
 - Retrieval time can take **up to 12 hours**
 - Best for long-term **compliance** and **archiving**

Each class is optimized for a different **use case** and **access pattern**, allowing you to choose based on cost, retrieval speed, and availability needs.

Introduction to Storage Lifecycle Policy in S3

An **S3 Lifecycle Policy** is a set of rules that **automates the transition** of objects between **storage classes** or **deletes them** after a certain period of time. This helps you **optimize costs** and manage data efficiently.

Key Capabilities:

- **Transition Rules**
Automatically move objects from one storage class to another (e.g., from Standard to Glacier) after a specific number of days.
- **Expiration Rules**
Automatically delete objects after a defined period of time (e.g., delete logs older than 90 days).

Common Use Cases:

- Move **infrequently accessed data** to **S3 Standard-IA** after 30 days.
- Archive data to **Glacier** or **Deep Archive** after 90 or 180 days.
- **Delete old backups or log files** after 1 year to save on storage costs.

Lifecycle policies are defined at the **bucket level**, and you can apply different rules based on **prefixes** (folders) or **tags**.



Introduction to S3 Charges (S3 Pricing)

Amazon S3 charges are based on multiple factors, and understanding them helps you manage costs effectively.

Key Factors That Affect S3 Pricing:

- Storage Used**
 - You are charged based on the total amount of data stored per month, depending on the **storage class** (e.g., Standard, Glacier).
- Number of Requests**
 - Charges apply for **GET, PUT, COPY, POST, DELETE** requests.
 - More requests = higher cost, especially in high-traffic applications.
- Data Transfer**
 - Data transferred out** of S3 to the internet is billed (first 1 GB per month is free).
 - Data transfer within the same region** (e.g., S3 to EC2) is often free.
- Management Features**
 - Features like **replication, object tagging, lifecycle policies, and inventory** may incur additional costs.
- Early Deletion Fees**
 - Some classes like **Glacier** and **Standard-IA** have minimum storage durations. Deleting files early may lead to extra charges.

Tips to Reduce S3 Costs:

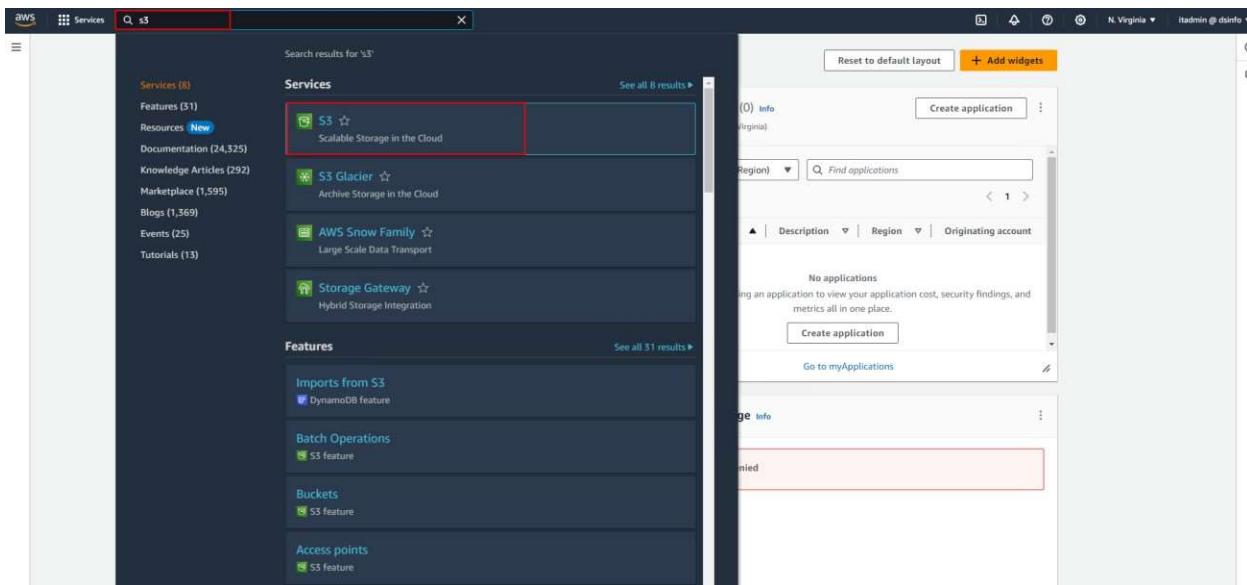
- Use **lifecycle policies** to move infrequently accessed data to lower-cost storage classes.

- Delete unused or outdated objects regularly.
- Minimize unnecessary read/write operations.

Always check the **AWS Pricing Calculator** for accurate and updated cost estimates.

How to Use AWS S3

To use S3, type "S3" in the search bar and then click on S3 from the results.



At this stage, click on the **Create Bucket** button.



At this stage, you need to specify an **AWS Region** and a **Bucket Name**.

A screenshot of the 'Create bucket' configuration page in the AWS Management Console. The URL in the address bar is 'Amazon S3 > Buckets > Create bucket'. The main form is titled 'General configuration'. It includes fields for 'AWS Region' (set to 'US East (N. Virginia) us-east-1'), 'Bucket type' (set to 'Info'), and 'Bucket name' (set to 'my-s3-89'). There are also sections for 'Copy settings from existing bucket - optional' and a 'Choose bucket' dropdown. A note at the bottom states 'Format: s3://bucket/prefix:'.

At this stage, if you select the "**Block all public access**" option, access to the S3 bucket from the public network will not be allowed. In this section, you can also enable **Bucket Versioning**.

The screenshot shows the "Block Public Access settings for this bucket" section with the "Block all public access" checkbox selected. It also shows the "Bucket Versioning" section with the "Disable" button selected.

At this stage, click on the **Create Bucket** button.

The screenshot shows the "Tags - optional (0)" section with a note about using tags to track storage costs and organize buckets. The "Default encryption" section shows "Server-side encryption is automatically applied to new objects stored in this bucket." Under "Encryption type", "Server-side encryption with Amazon S3 managed keys (SSE-S3)" is selected. The "Bucket Key" section notes that using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. The "Advanced settings" section contains a note about uploading files after creation. At the bottom are "Cancel" and "Create bucket" buttons.

At this stage, select the **bucket** you just created.

The screenshot shows the 'Amazon S3 > Buckets' page. A green header bar at the top indicates 'Successfully created bucket "my-s3-89"'. Below it, a message says 'To upload files and folders, or to configure additional bucket settings, choose View details.' On the left, there's an 'Account snapshot' section with a link to 'View Storage Lens dashboard'. The main area has tabs for 'General purpose buckets' (selected) and 'Directory buckets'. Under 'General purpose buckets', there's a table with one row:

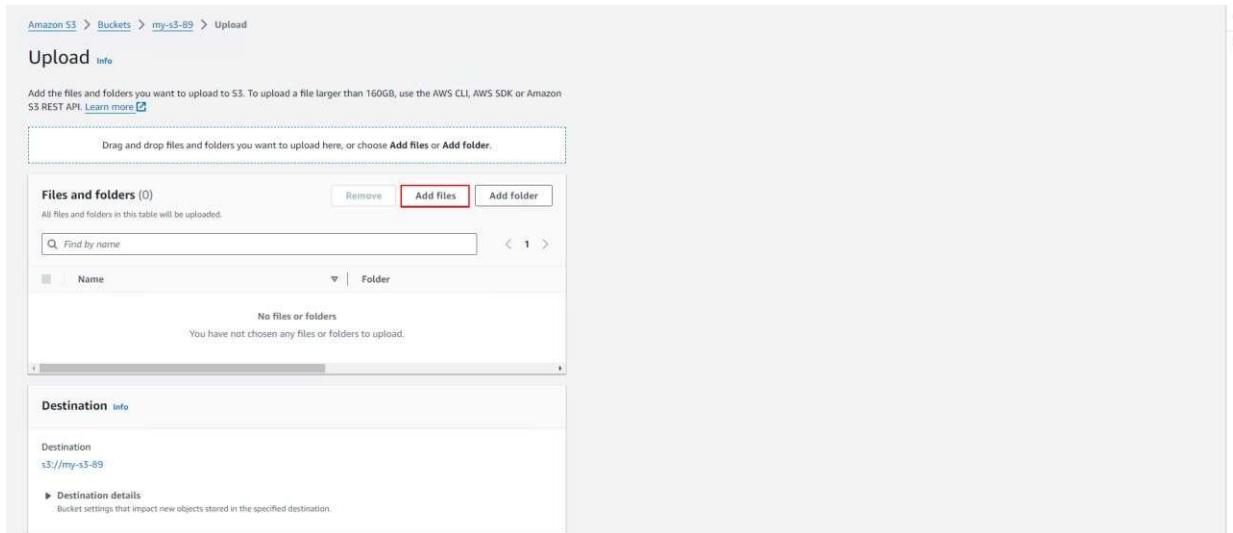
Name	AWS Region	Access	Creation date
my-s3-89	US East (N. Virginia) us-east-1	Bucket and objects not public	March 24, 2024, 23:34:36 (UTC-07:00)

Actions for this row include: Copy ARN, Empty, Delete, and Create bucket (which is highlighted in orange).

In the **Objects** section, you can upload your files to the **S3 bucket**. To upload a file, click on the **Upload** button.

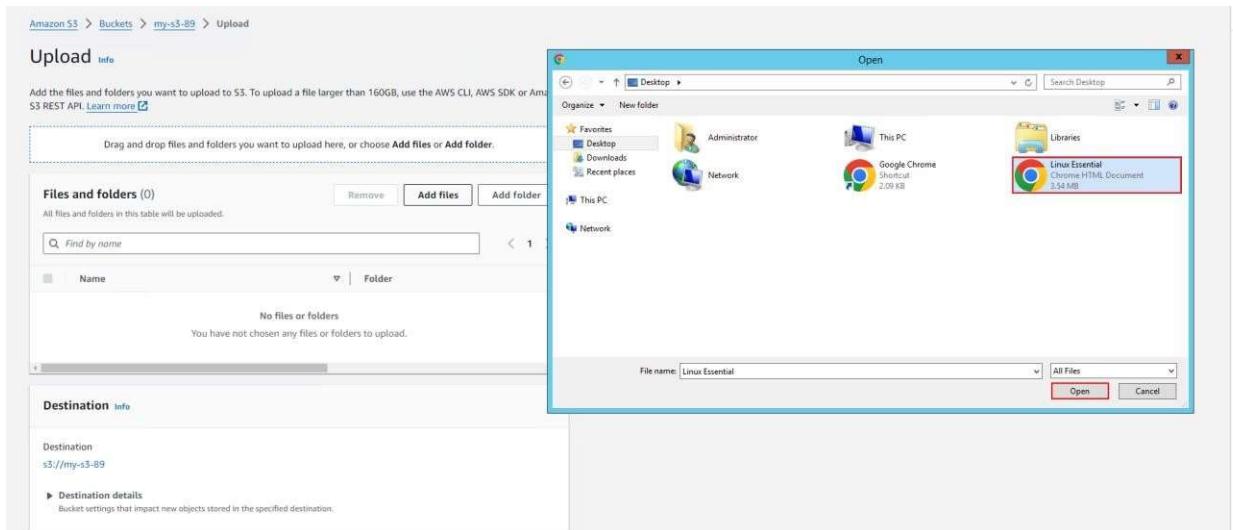
The screenshot shows the 'Amazon S3 > Buckets > my-s3-89' page. The 'Objects' tab is selected. At the top, there are several actions: Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload (which is highlighted in orange). Below these are search and filter fields: 'Find objects by prefix' and 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. A message states 'No objects' and 'You don't have any objects in this bucket.' At the bottom is a large 'Upload' button.

At this stage, click on the **Add Files** button.



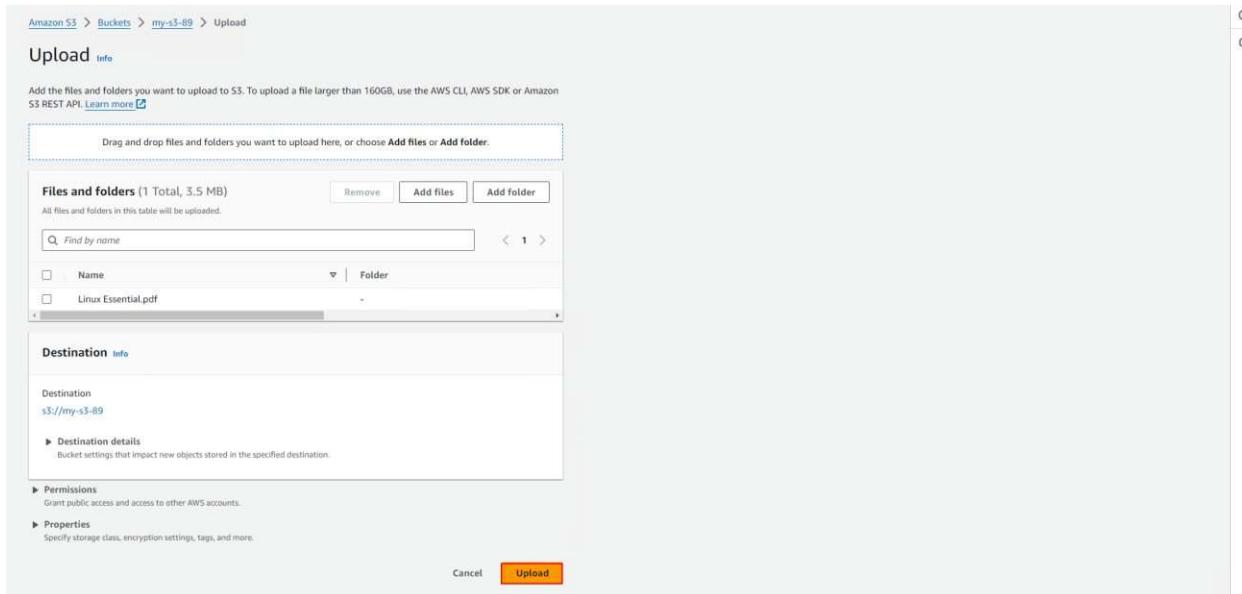
The screenshot shows the AWS S3 'Upload' interface. At the top, it says 'Amazon S3 > Buckets > my-s3-89 > Upload'. Below that is a 'Upload' section with a 'Drag and drop files and folders you want to upload here...' area. A red box highlights the 'Add files' button. To the right of the upload area is a 'Files and folders (0)' table with columns for 'Name' and 'Folder'. Below the table, it says 'No files or folders' and 'You have not chosen any files or folders to upload.' Further down, there's a 'Destination' section with 'Destination' set to 's3://my-s3-89' and a 'Destination details' link.

Select your desired file from your computer, then click the **Open** button.

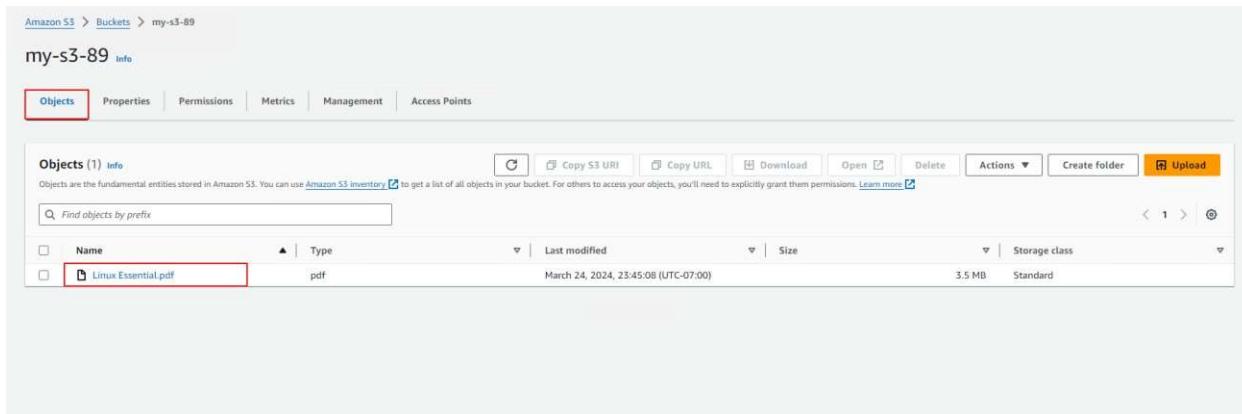


The screenshot shows the same AWS S3 'Upload' interface as above, but with a Windows 'Open' file dialog box overlaid. The dialog shows a list of files on the desktop, including 'Administrator', 'This PC', 'Network', 'Google Chrome Shortcut', and 'Linux Essential'. A red box highlights the 'Linux Essential' file. At the bottom of the dialog, the 'File name' field contains 'Linux Essential' and the 'Open' button is highlighted.

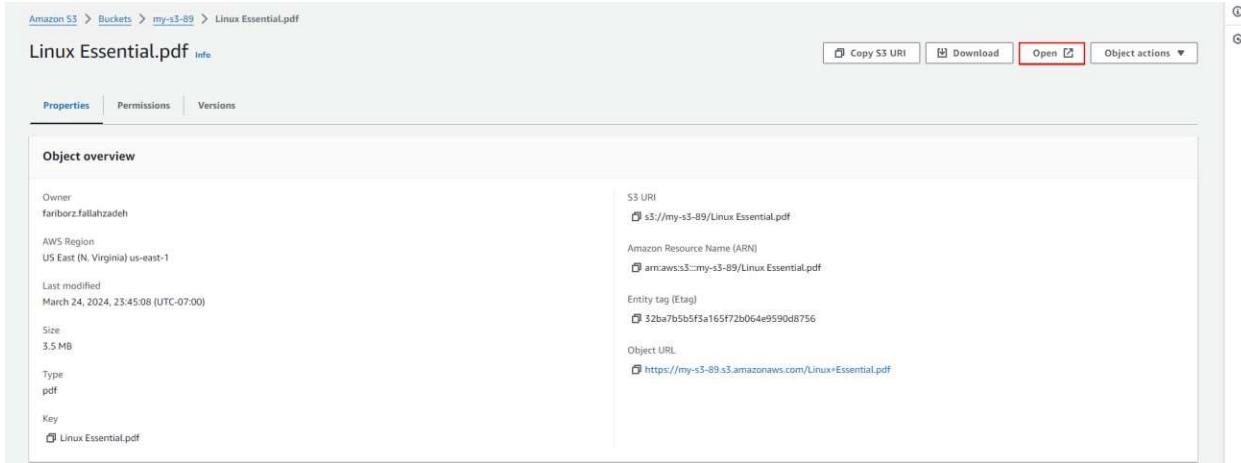
Then, click on the **Upload** button.



As shown in the image below, our file has been successfully uploaded to the **S3 Bucket**.

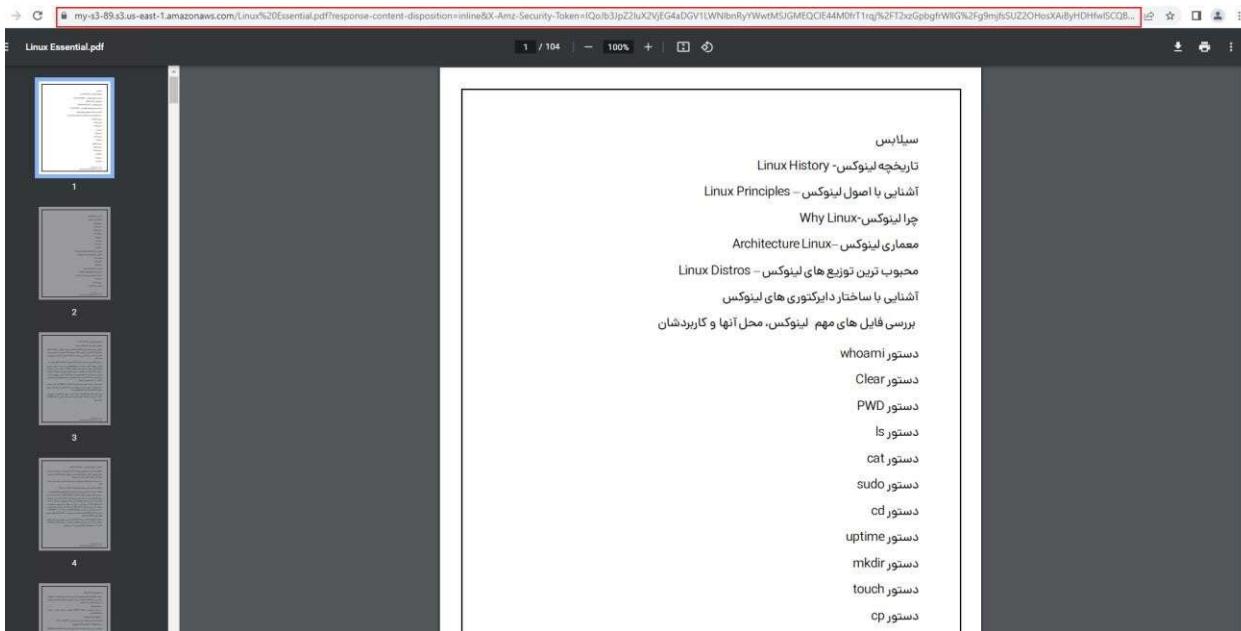


If you click on the **uploaded file**, you will see its details. To open the file, click on the **Open** button.



The screenshot shows the 'Object overview' section of the Amazon S3 console. At the top, there are tabs for 'Properties', 'Permissions', and 'Versions'. Below the tabs, the 'Object overview' section displays various metadata fields. In the top right corner of the overview section, there is a 'Copy S3 URI' button, a 'Download' button, a red-bordered 'Open' button, and a 'Object actions' dropdown menu. The 'Open' button is the primary focus, indicating it is the action taken to view the file's contents.

As you can see, the selected file has been opened on the web.



The screenshot shows a web browser displaying the PDF document 'Linux Essential.pdf'. The browser interface includes a title bar with the URL 'my-s3-89.s3.us-east-1.amazonaws.com/linux%20Essential.pdf?response-content-disposition=inline&X-Amz-Security-Token=IQoIB3IjZ2luXlVjLG4aDGV1WNlbnRyYWwvMSJGMEQiE44M0f71rq%2FT2xGpbgfWlG%2Fg9mjtsSUZ2OHosXAiByHDHfwSCQ8...', a navigation bar with zoom controls, and a sidebar on the left showing document thumbnails numbered 1 through 4. The main content area displays the first page of the PDF. To the right of the main content, there is a sidebar containing Persian text related to Linux topics such as 'سیلابس', 'تاریخچه لینوکس', 'آشنایی با اصول لینوکس', 'Why Linux', 'Architecture Linux', 'مهمازی لینوکس', 'محبوب ترین توزیع های لینوکس', 'آشنایی با ساختار دیرکتوری های لینوکس', 'بررسی فایل های مهم لینوکس', 'محل آنها و کاربردشان', and a list of common Linux commands: 'whoami', 'Clear', 'PWD', 'ls', 'cat', 'sudo', 'cd', 'uptime', 'mkdir', 'touch', and 'cp'.

In the file's **Details** section, there is a field called **Object URL**, which is the access link to the file. Copy this URL and paste it into the **address bar** of a browser on your system.

The screenshot shows the 'Object overview' section of the Amazon S3 console. It displays various metadata for the file, including its owner, AWS Region, last modified date, size, type, and key. The 'Object URL' field is highlighted with a red box and contains the value: <https://my-s3-89.s3.amazonaws.com/Linux+Essential.pdf>.

As you can see, access to the file through the **public network** is not allowed.

The screenshot shows a web browser displaying an XML error document. The URL in the address bar is <https://my-s3-89.s3.amazonaws.com/Linux+Essential.pdf>. The error message indicates that access is denied due to a missing grant.

```
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>03PT8R86BTCTP#P#</RequestId>
<HostId>khRHTXjI+YCD5H#j13hw/hCtQg0y5d014UE3N8EpihLGLDzp+Vn7r#B514L00Y+98jfe/dE5s+</HostId>
</Error>
```

To enable **public access** to the file, go to the **Permissions** tab of the selected file, then click on the **Edit** button under the **Block Public Access** section.

Amazon S3 > Buckets > my-s3-89

my-s3-89 [Info](#)

Objects Properties **Permissions** Metrics Management Access Points

Permissions overview

Access

Bucket and objects not public

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access

On

► Individual Block Public Access settings for this bucket

Edit

At this stage, **uncheck** the option **Block all public access**, and then click on the **Save Changes** button.

Amazon S3 > Buckets > my-s3-89 > Edit Block public access (bucket settings)

Edit Block public access (bucket settings) [Info](#)

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

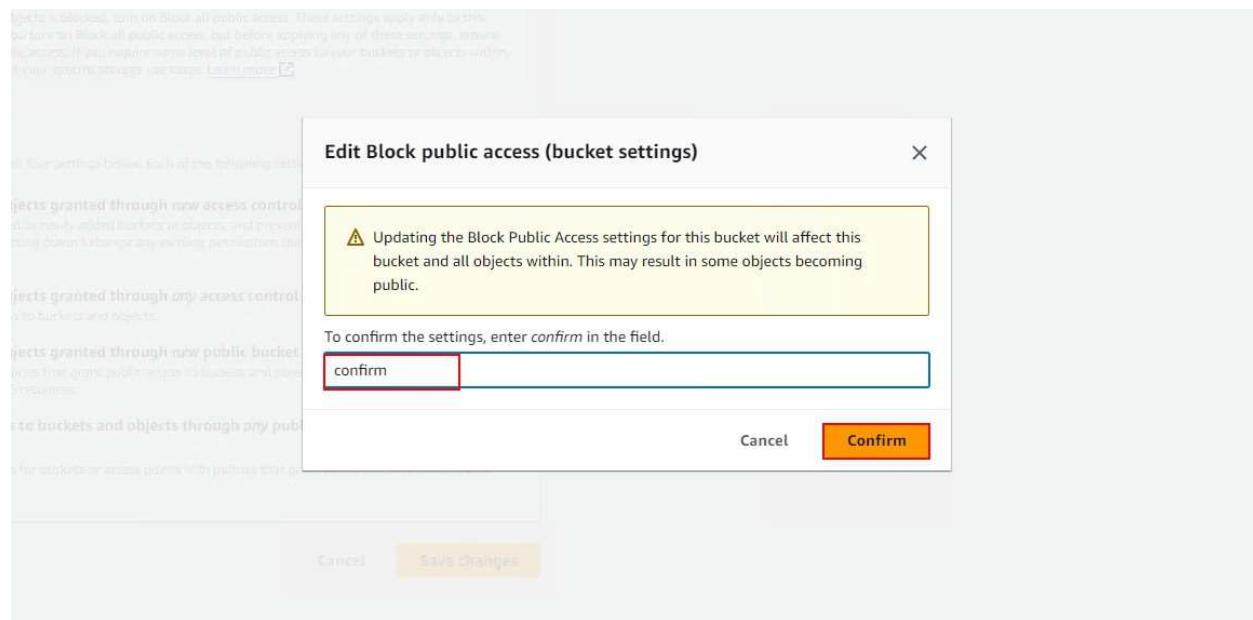
Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

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

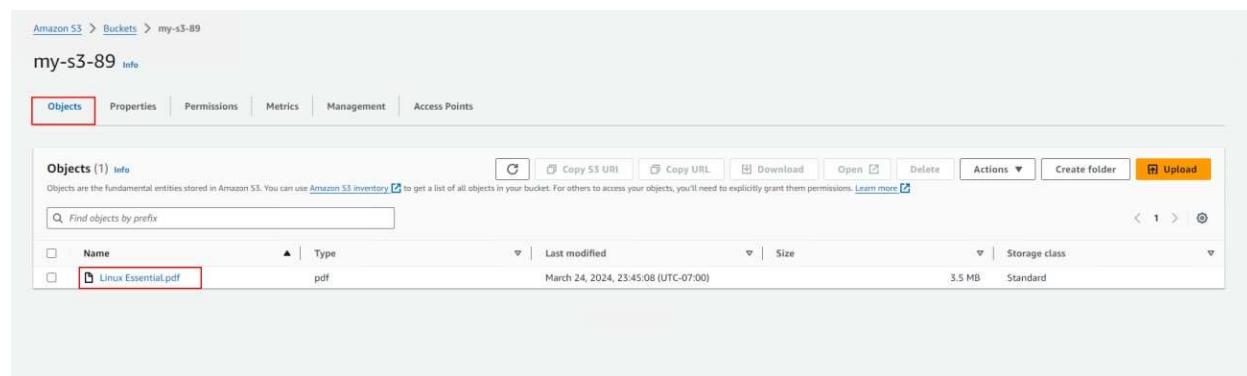
Next, type the word **Confirm** in the provided text box, then click the **Confirm** button.



As you can see, when testing web access to the file again, **access is still not allowed**.



To resolve the file access issue, click on the **desired file**.



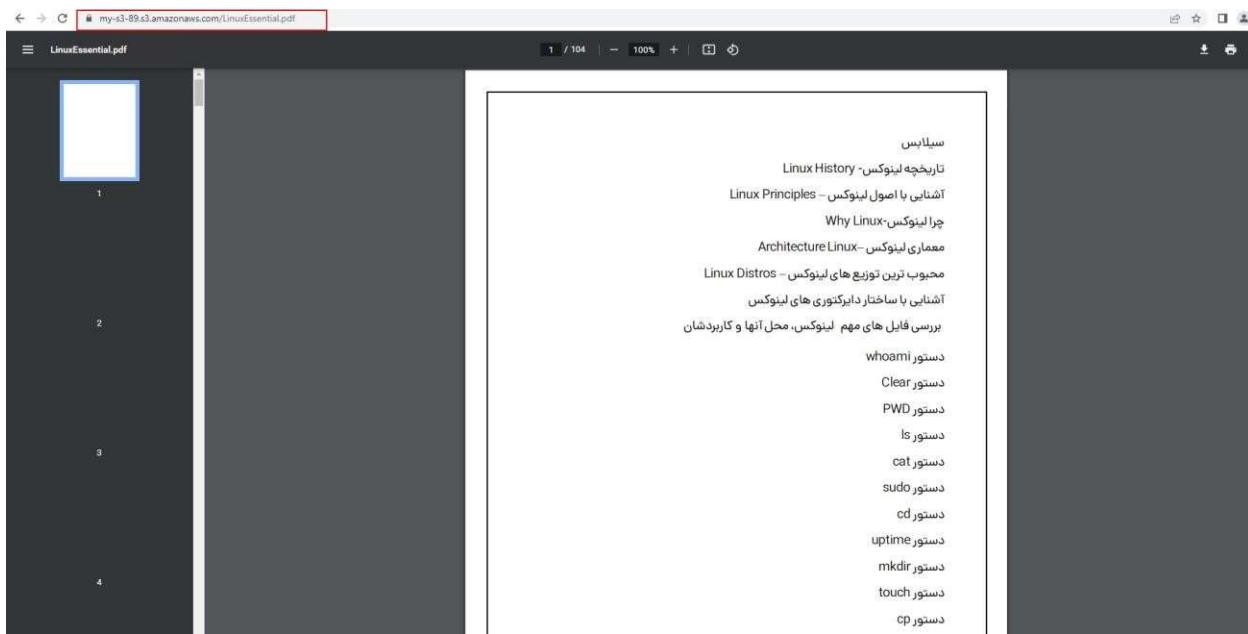
Then, in the **Object actions** menu, click on **Make public using ACL**.

The screenshot shows the 'Object overview' section of the Amazon S3 console. On the right, the 'Object actions' menu is open, displaying various options like 'Copy S3 URI', 'Download', 'Open', and 'Object actions'. The 'Object actions' dropdown menu is expanded, showing options such as 'Share with a presigned URL', 'Calculate total size', 'Copy', 'Move', 'Initiate restore', 'Query with S3 Select', 'Edit actions', 'Rename object', 'Edit storage class', 'Edit server-side encryption', 'Edit metadata', 'Edit tags', and 'Make public using ACL'. The 'Make public using ACL' option is highlighted with a red box.

Then, click on the **Make Public** button.

The screenshot shows the 'Specified objects' section of the 'Make public' dialog. It lists a single object: 'LinuxEssential.pdf'. Below the table is a 'Cancel' button and a prominent 'Make public' button, which is highlighted with a red box.

As you can see, the file is now **accessible via the web** and through the **public network**.

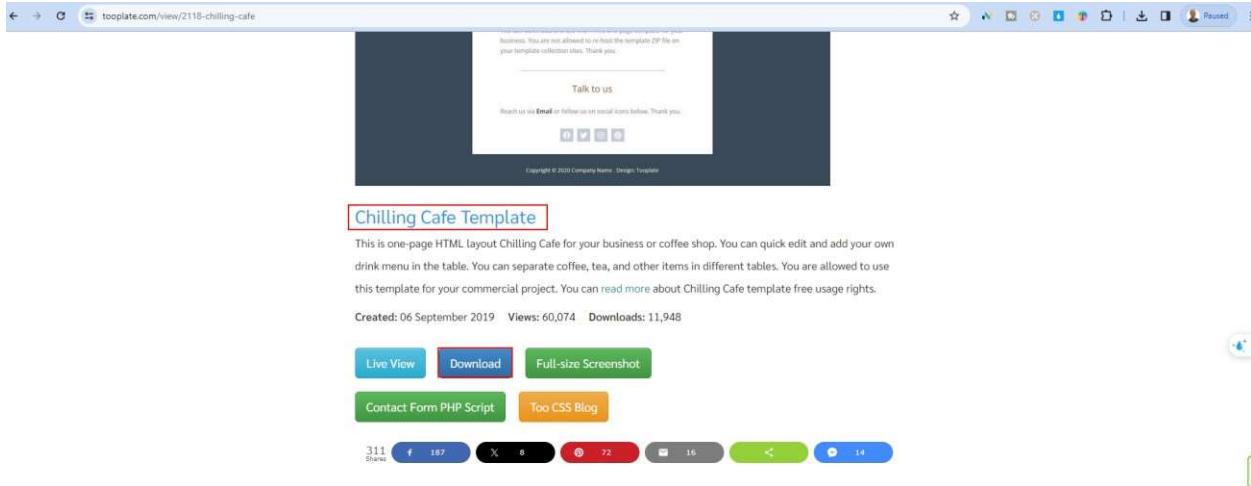


How to Host a Static Website Using an S3 Bucket

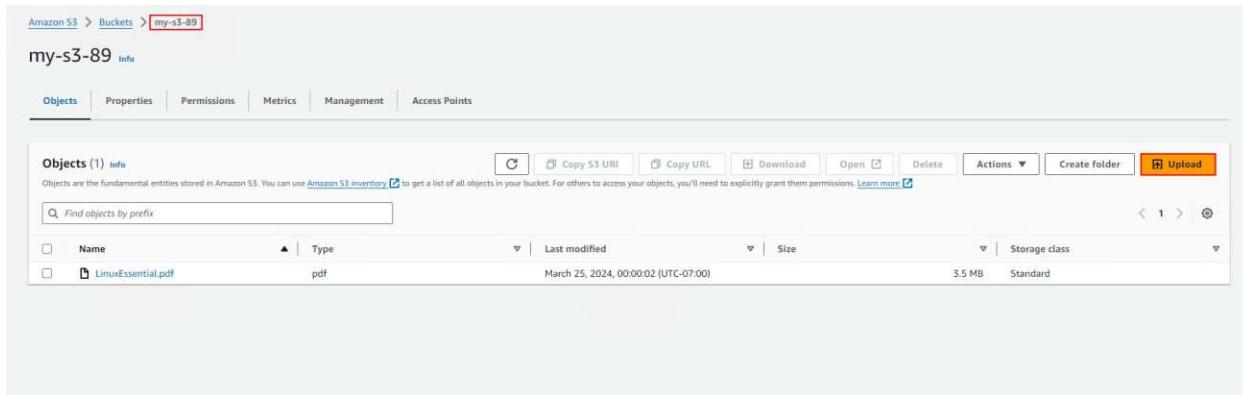
We can use an **S3 Bucket as a host** for web files as well.

To better understand this, we'll use a **template from the Tooplate website**, download one of the templates, upload it to an **S3 bucket**, and use that **S3 bucket as a web server**.

First, go to the **Tooplate website** and download one of the **template files**.

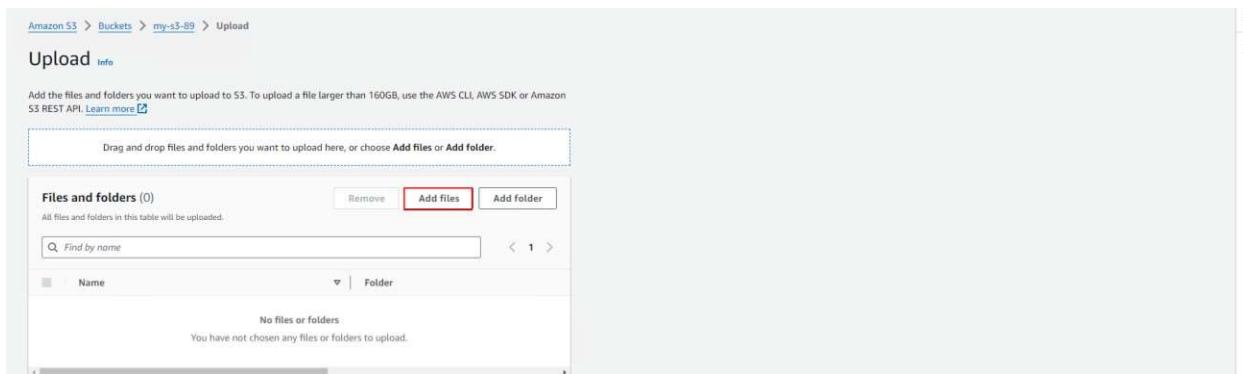


Then, go to the **S3 bucket** you created in the previous steps and click on the **Upload** button.



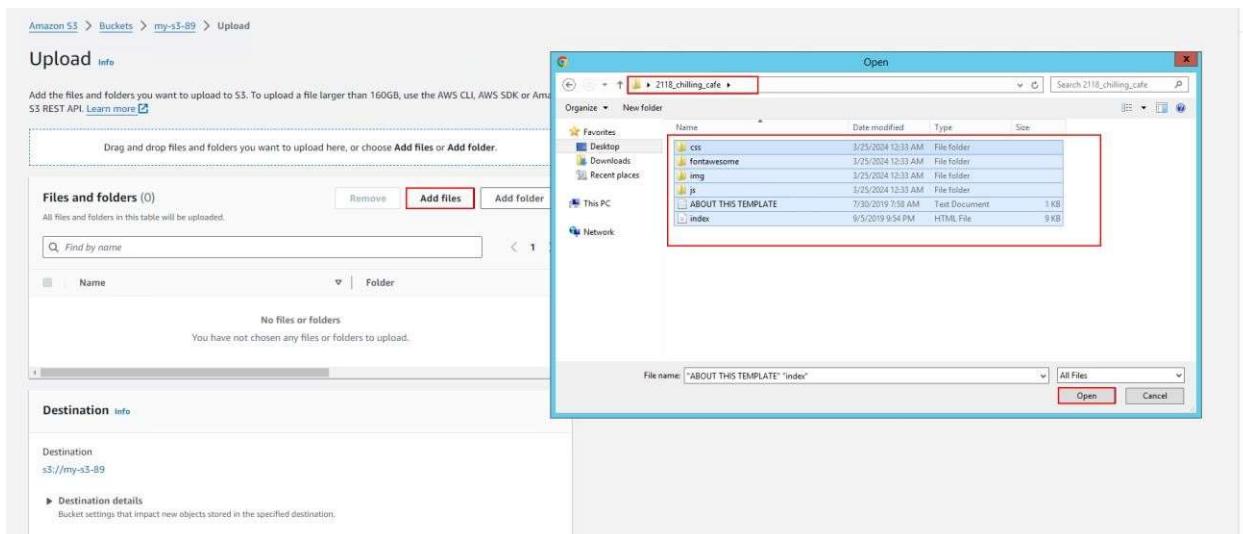
The screenshot shows the Amazon S3 console interface. The top navigation bar includes 'Amazon S3', 'Buckets', and 'my-s3-89'. Below the navigation is a toolbar with 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The main area is titled 'Objects (1) info' and contains a table with one item: 'LinuxEssential.pdf' (pdf, 3.5 MB, Standard storage class). At the bottom right of the table is an 'Upload' button, which is highlighted with a red box.

At this stage, click on the **Add Files** button.



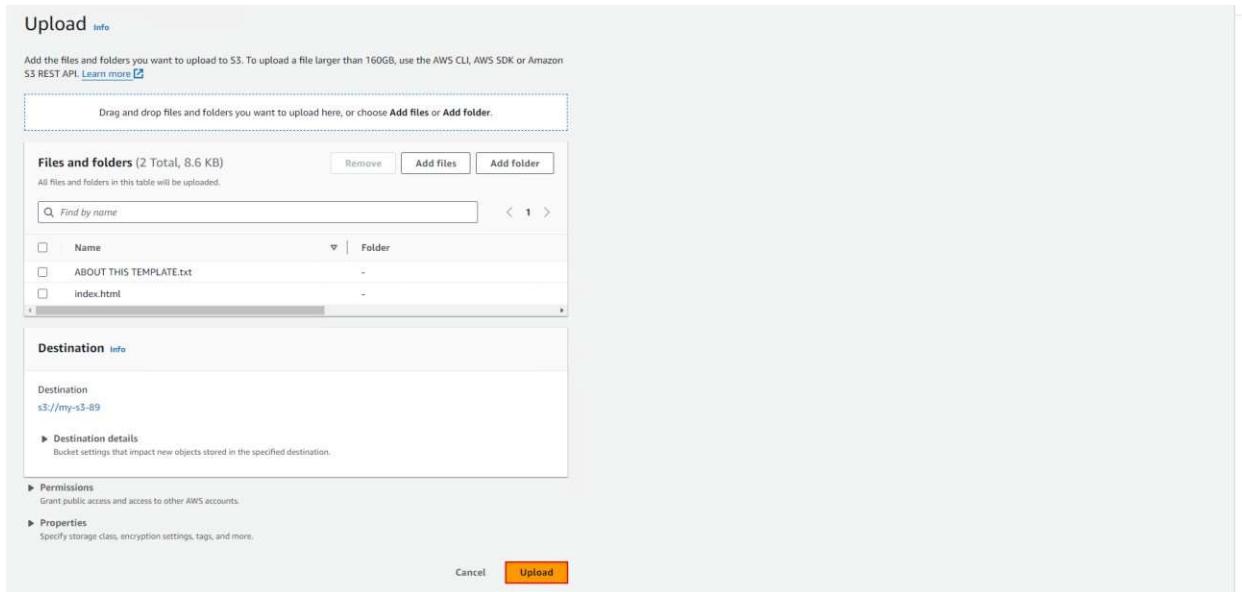
The screenshot shows the 'Upload' step in the AWS S3 console. The top navigation bar includes 'Amazon S3', 'Buckets', 'my-s3-89', and 'Upload'. The main area has a large 'Drag and drop files and folders you want to upload here...' area. Below it is a 'Files and folders (0)' section with a 'Remove' button, an 'Add files' button (which is highlighted with a red box), and an 'Add folder' button. A message says 'All files and folders in this table will be uploaded.' and 'No files or folders'.

Then, select the files and click the **Open** button.

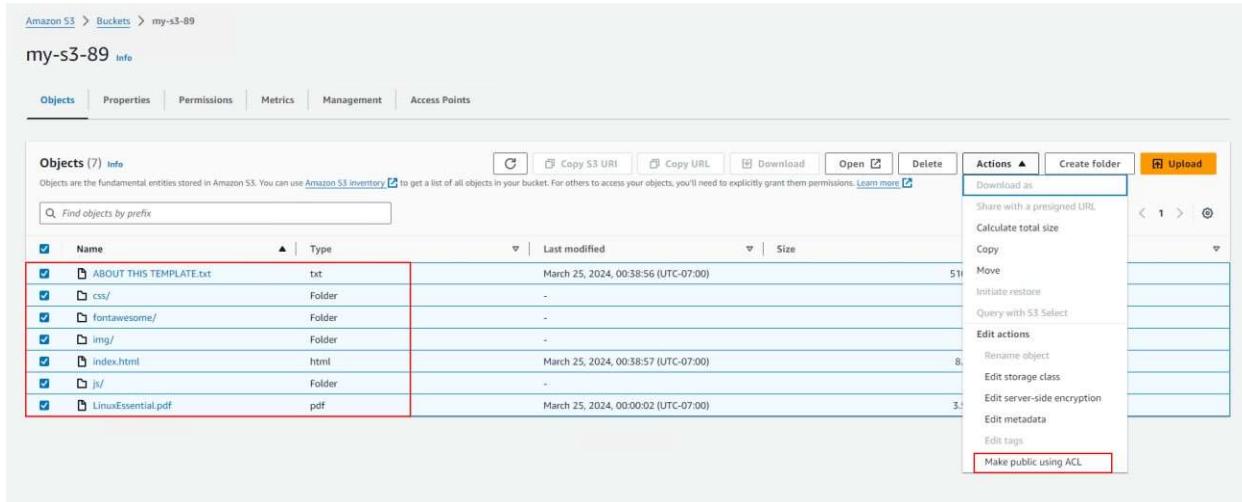


The screenshot shows the 'Upload' step in the AWS S3 console with a file selection dialog open. The dialog title is 'Open' and shows a list of files from a folder named '2118_chilling_cafe'. The 'index' file is selected, indicated by a red box around the file entry in the list. The file list includes: css, fontawesome, img, js, ABOUT THIS TEMPLATE, and index. The 'File name' dropdown at the bottom shows 'ABOUT THIS TEMPLATE "index"'. At the bottom right of the dialog are 'Open' and 'Cancel' buttons.

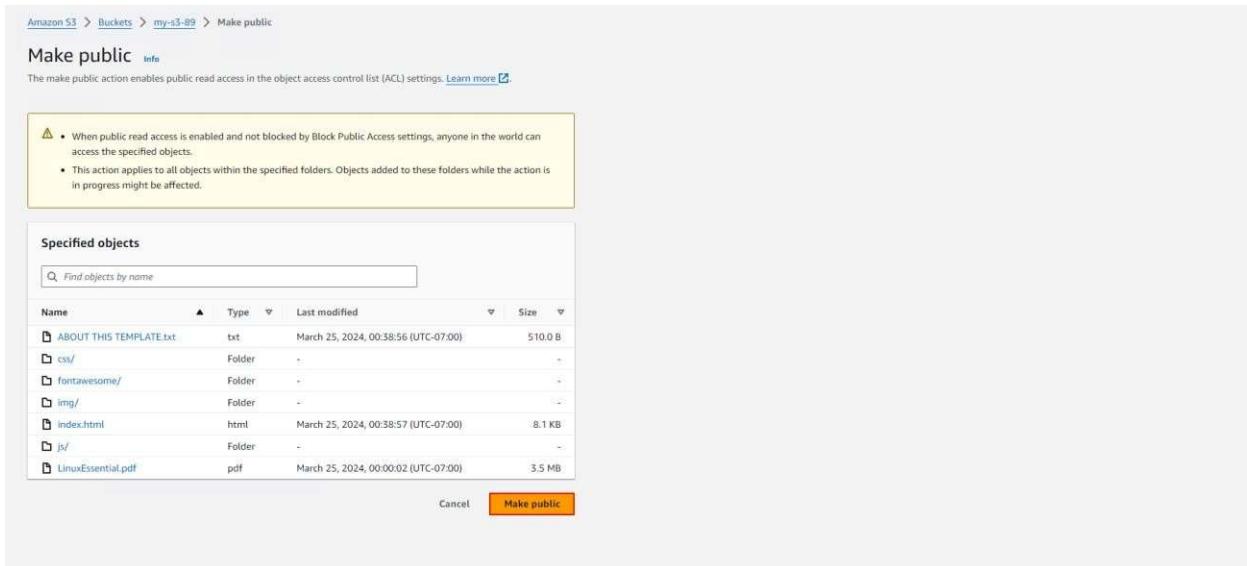
Then, click on the **Upload** button.



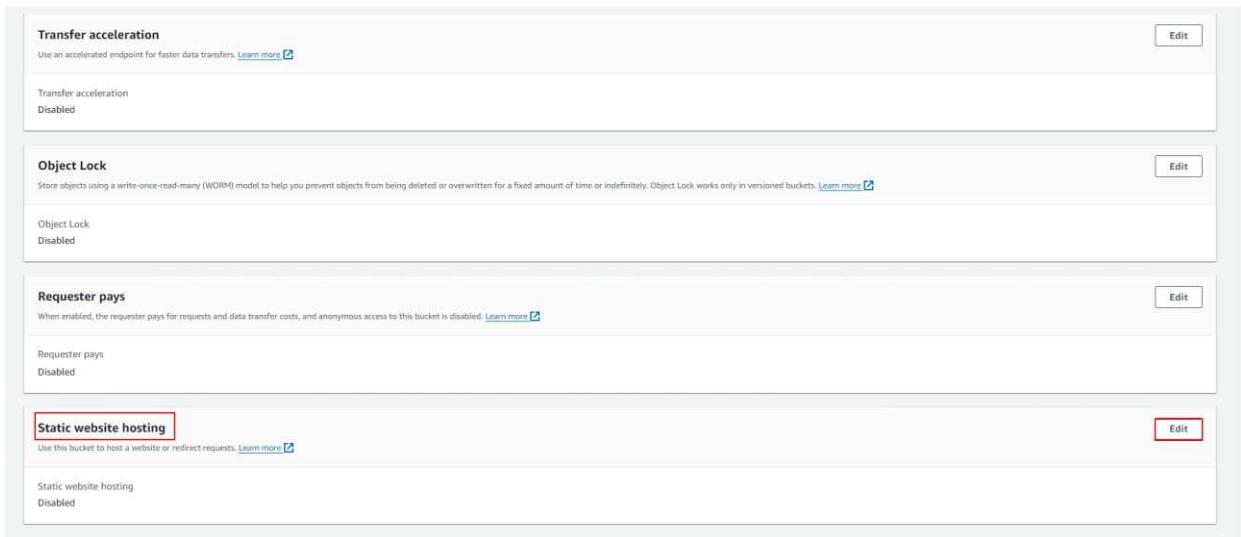
At this stage, select all the **template files**, then from the **Actions** menu, choose **Make Public Using ACL**.



Next, click on the **Make Public** button.



Then, in the **S3 Bucket settings**, go to the **Static Website Hosting** section and click the **Edit** button.



Then, enter the following settings in this section

The screenshot shows the 'Edit static website hosting' configuration page for a bucket named 'my-s3-89'. The 'Static website hosting' section is active, with 'Enable' selected. Under 'Hosting type', 'Host a static website' is selected, with a note about using the bucket endpoint as the web address. A callout box highlights this note. The 'Index document' field contains 'index.html'. The 'Error document - optional' field contains 'error.html'. A note about redirection rules is present at the bottom.

And finally, click on the **Save Changes** button.

The screenshot shows the 'Redirection rules' configuration dialog box. It displays a JSON editor with a single rule entry. The rule is defined as follows:

```
1
{
  "Redirect": {
    "Status": 301,
    "Protocol": "HTTP",
    "HostName": "www.example.com",
    "ReplaceKeyPrefix": "http://www."
  }
}
```

Below the editor, the status bar shows 'JSON' and 'Ln 1, Col 1'. At the bottom right are 'Cancel' and 'Save changes' buttons, with 'Save changes' being highlighted.

As you can see, after completing the settings, a **URL** is provided which allows access to your **website files**.

Successfully edited static website hosting.

Use this endpoint or enter a custom domain. [Learn more](#) [Edit](#)

Transfer acceleration
Disabled

Object Lock
Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. Object Lock works only in versioned buckets. [Learn more](#) [Edit](#)

Object Lock
Disabled

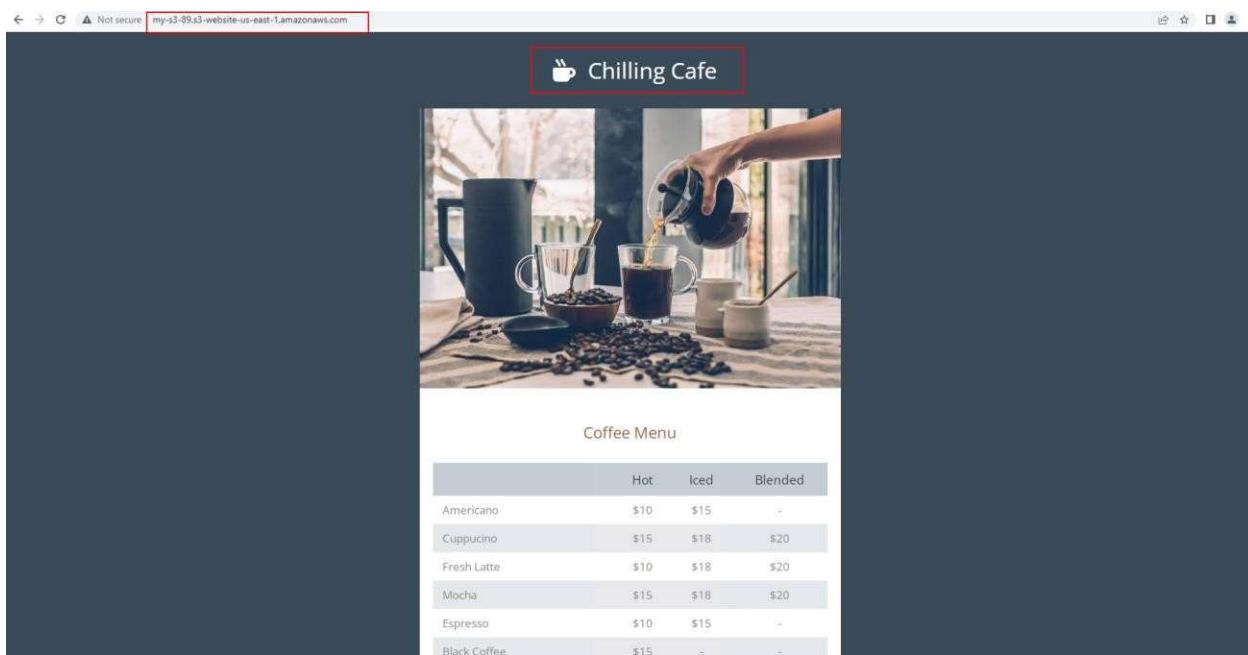
Requester pays
When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. [Learn more](#) [Edit](#)

Requester pays
Disabled

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#) [Edit](#)

Static website hosting
Enabled
Hosting type
Bucket hosting
Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#) <http://my-s3-89.s3-website-us-east-1.amazonaws.com>

When you enter the above link in your **browser**, you will see the **desired website** that is now **hosted on your S3 bucket**.

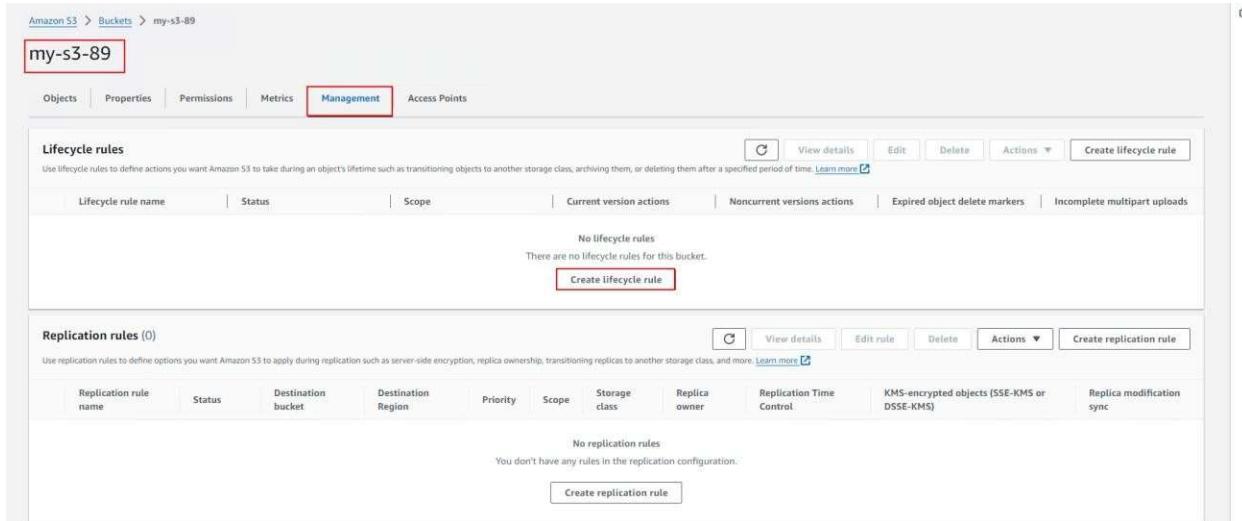


How to Use S3 Bucket Lifecycle

You can define a **Lifecycle** for an **S3 Bucket**, which means your files will transition from one **storage class** to another over time and eventually be **deleted**.

Using a **lifecycle** helps **reduce costs** for maintaining files within the **S3 bucket**.

To define an **S3 Bucket Lifecycle**, click on your **S3 bucket**, then go to the **Management** tab and click on the **Create lifecycle rule** button.



The screenshot shows the Amazon S3 Management console for a bucket named "my-s3-89". The "Management" tab is selected. In the "Lifecycle rules" section, there is a message: "No lifecycle rules There are no lifecycle rules for this bucket." Below this is a "Create lifecycle rule" button. In the "Replication rules (0)" section, there is a message: "No replication rules You don't have any rules in the replication configuration." Below this is a "Create replication rule" button.

At this stage, specify a **name** for the lifecycle rule and define a **prefix**.

Amazon S3 > Buckets > my-s3-89 > Lifecycle configuration > Create lifecycle rule

Create lifecycle rule [Info](#)

Lifecycle rule configuration

Lifecycle rule name
archive-policies

Up to 255 characters

Choose a rule scope
 Limit the scope of this rule using one or more filters
 Apply to all objects in the bucket

Filter type
You can filter objects by prefix, object tags, object size, or whatever combination suits your usecase.

Prefix
Add filter to limit the scope of this rule to a single prefix.
log

Don't include the bucket name in the prefix. Using certain characters in key names can cause problems with some applications and protocols. [Learn more](#)

Object tags
You can limit the scope of this rule to the key/value pairs added below.

[Add tag](#)

At this stage, select the option "**Move current version of objects between storage classes**" to enable automatic transitions between storage classes.

Then, select "**Expire current version of objects**" to define after how many days the data should be **deleted**.

Then, you need to specify the **number of days** the data should remain in each **storage class** before being transitioned or deleted.

In this step, you must define how many **days** the data should remain in each **storage class** before it should **transition** to the next storage class.

The screenshot shows the 'Transition current versions of objects between storage classes' section. It includes a table for defining storage class transitions and a table for defining expiration rules. The 'Days after object creation' field for the first transition has a value of '30'. The 'Days after object creation' field for the second transition has a value of '60'. The 'Days after object creation' field for the expiration rule has a value of '400'. The 'Review transition and expiration actions' section shows that for current version actions, the value is 'Day 0', and for noncurrent versions actions, the value is also 'Day 0'.

Choose storage class transitions	Days after object creation
Standard-IA	30
One Zone-IA	60

Days after object creation
400

Review transition and expiration actions	
Current version actions	Noncurrent versions actions
Day 0	Day 0

And finally, click on the **Create Rule** button.

Days after object creation
400

Review transition and expiration actions

Current version actions	Noncurrent versions actions
Day 0 <ul style="list-style-type: none">Objects uploaded	Day 0 <ul style="list-style-type: none">No actions defined.
↓	
Day 30 <ul style="list-style-type: none">Objects move to Standard-IA	
↓	
Day 60 <ul style="list-style-type: none">Objects move to One Zone-IA	
↓	
Day 400 <ul style="list-style-type: none">Objects expire	

Cancel Create rule

As you can see, the **lifecycle rule** has been successfully created for our **S3 bucket**.

Amazon S3 > Buckets > my-s3-b9 > Lifecycle configuration

Lifecycle configuration [Info](#)

To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their lifecycle. A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. Lifecycle rules run once per day.

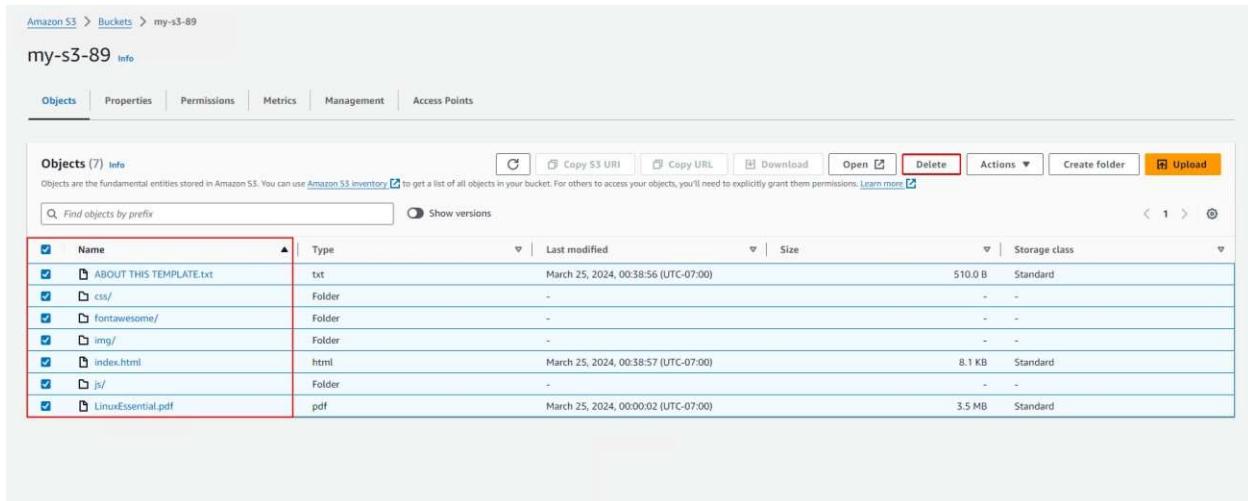
Lifecycle rules (1)

Use lifecycle rules to define actions you want Amazon S3 to take during an object's lifetime such as transitioning objects to another storage class, archiving them, or deleting them after a specified period of time. [Learn more](#)

Lifecycle rule name	Status	Scope	Current version actions	Noncurrent versions actions	Expired object delete mar...	Incomplete multipart upl...
archive-polices	Enabled	Filtered	Transition to Standard-IA, then One	-	-	-

< 1 > @

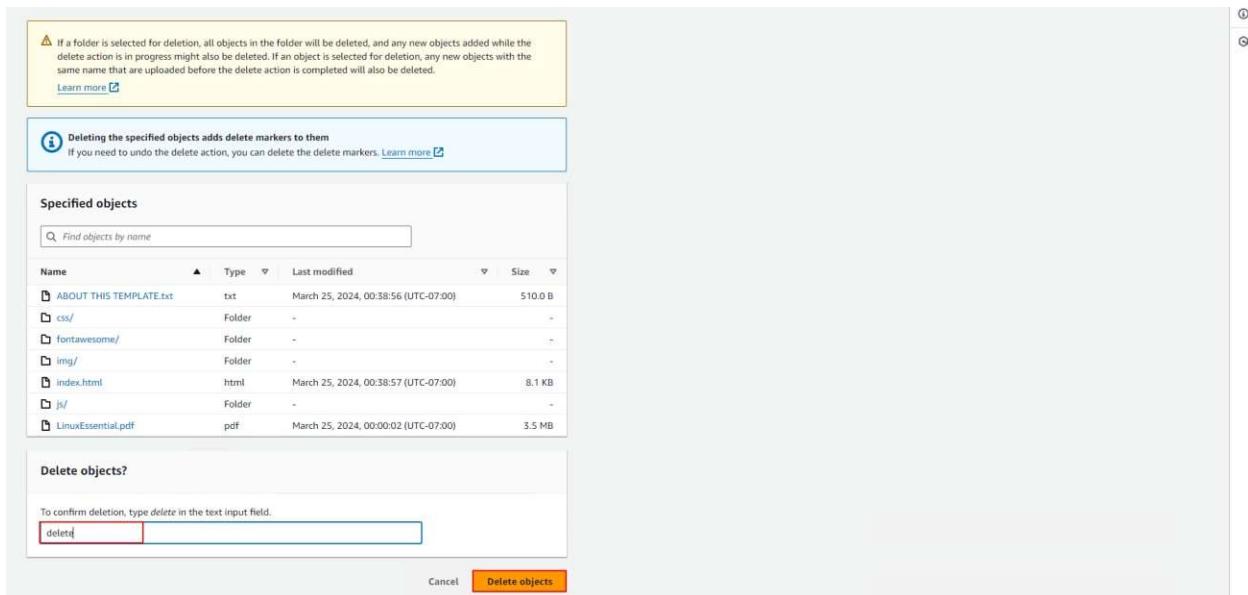
To delete files from an **S3 bucket**, select the files and then click the **Delete** button.



The screenshot shows the Amazon S3 console interface. At the top, there's a navigation bar with 'Amazon S3 > Buckets > my-s3-89'. Below it is a toolbar with tabs: Objects (highlighted), Properties, Permissions, Metrics, Management, and Access Points. The main area is titled 'Objects (7) Info' and contains a table of objects. The table has columns for Name, Type, Last modified, Size, and Storage class. Several objects are selected, indicated by a checked checkbox in the first column. A red box highlights the 'Delete' button in the toolbar above the table. The table data is as follows:

Name	Type	Last modified	Size	Storage class
ABOUT THIS TEMPLATE.txt	txt	March 25, 2024, 00:38:56 (UTC-07:00)	510.0 B	Standard
css/	Folder	-	-	-
fontawesome/	Folder	-	-	-
img/	Folder	-	-	-
index.html	html	March 25, 2024, 00:38:57 (UTC-07:00)	8.1 KB	Standard
js/	Folder	-	-	-
LinuxEssential.pdf	pdf	March 25, 2024, 00:00:02 (UTC-07:00)	3.5 MB	Standard

In the text box, type the word **Delete**, then click on **Delete objects**.



The screenshot shows a confirmation dialog box. At the top, there's a warning message: '⚠ If a folder is selected for deletion, all objects in the folder will be deleted, and any new objects added while the delete action is in progress might also be deleted. If an object is selected for deletion, any new objects with the same name that are uploaded before the delete action is completed will also be deleted.' Below this is an information message: ' ⓘ Deleting the specified objects adds delete markers to them. If you need to undo the delete action, you can delete the delete markers. Learn more'.

The main area is titled 'Specified objects' and shows a table of the same objects as the previous screenshot. A red box highlights the 'Delete' button in the bottom right corner of the dialog. The table data is identical to the one above.

At the bottom, there's a text input field with the placeholder 'To confirm deletion, type delete in the text input field.' and a red box highlighting the word 'delete'. To the left of the input field is a 'Cancel' button, and to the right is a large orange 'Delete objects' button.

To delete an **S3 bucket**, select the bucket and then click on the **Delete** button.

The screenshot shows the 'Buckets' page in the Amazon S3 console. At the top, there's an 'Account snapshot' section with a link to 'View Storage Lens dashboard'. Below it, there are tabs for 'General purpose buckets' (which is selected) and 'Directory buckets'. A search bar labeled 'Find buckets by name' is present. The main table lists one bucket:

Name	AWS Region	Access	Creation date
my-s3-89	US East (N. Virginia) us-east-1	Objects can be public	March 24, 2024, 23:34:36 (UTC-07:00)

Actions for this bucket include 'Copy ARN', 'Empty', and 'Delete' (which is highlighted with a red box). There are also 'Create bucket' and other navigation buttons at the bottom right.

In the text box, type **Permanently Delete**, then click the **Empty** button.

The screenshot shows the 'Empty bucket' confirmation dialog. The URL in the address bar is 'Amazon S3 > Buckets > my-s3-89 > Empty bucket'. The title is 'Empty bucket' with a 'Info' link. A warning box contains:

- Emptying the bucket deletes all objects in the bucket and cannot be undone.
- Objects added to the bucket while the empty bucket action is in progress might be deleted.
- To prevent new objects from being added to this bucket while the empty bucket action is in progress, you might need to update your bucket policy to stop objects from being added to the bucket.

Below the warning is a note: 'If your bucket contains a large number of objects, creating a lifecycle rule to delete all objects in the bucket might be a more efficient way of emptying your bucket.' with a 'Learn more' link. The main confirmation message is 'Permanently delete all objects in bucket "my-s3-89"?'. A text input field contains 'permanently delete'. At the bottom are 'Cancel' and 'Empty' buttons, with 'Empty' highlighted with a red box.

Introduction to AWS RDS

AWS RDS is a **managed database service** provided by Amazon that makes it easy to set up, operate, and scale a **relational database** in the cloud.

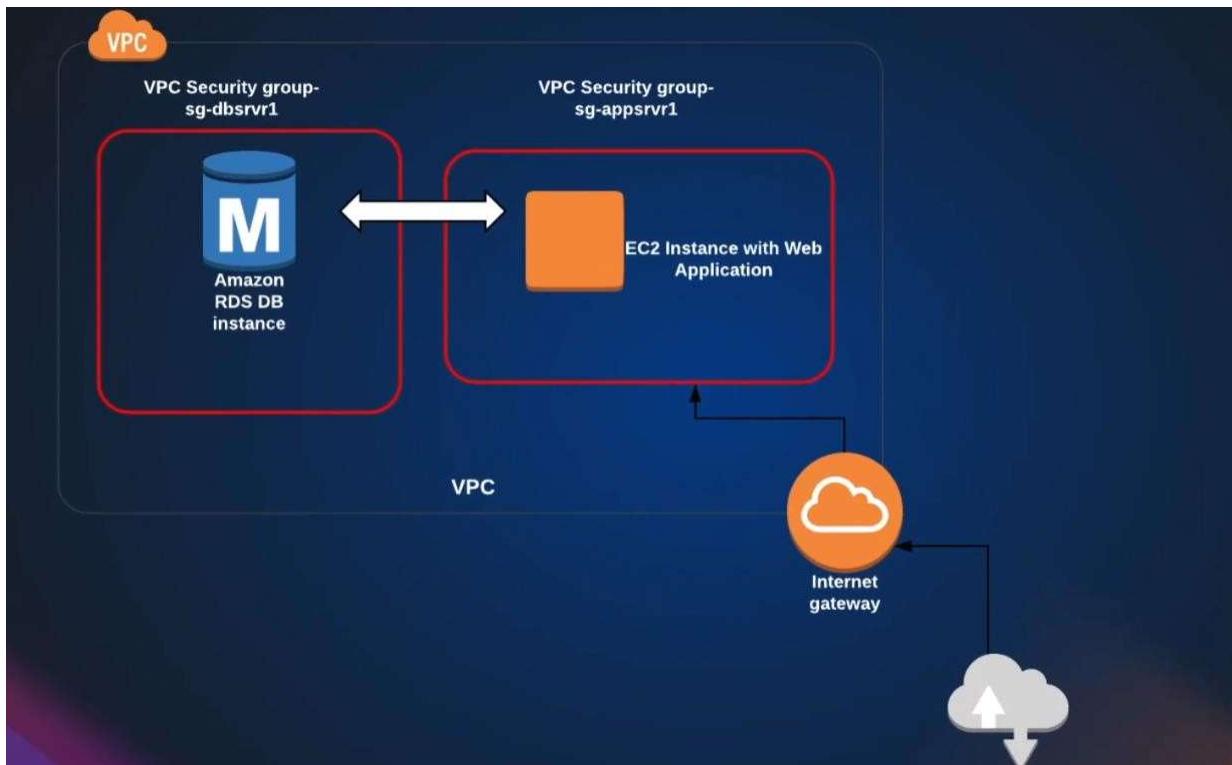
Key Features of AWS RDS:

- **Fully managed:** AWS handles backups, patching, monitoring, and scaling.
- **Supports multiple database engines**, including:
 - Amazon Aurora
 - MySQL
 - PostgreSQL
 - MariaDB
 - Oracle
 - SQL Server
- **High availability** with Multi-AZ deployment.
- **Automated backups and snapshots.**
- **Scalable performance:** You can easily change compute and storage resources.
- **Security:** Integration with IAM, encryption at rest and in transit, VPC support.

Common Use Cases:

- Hosting **production databases** for web or mobile apps.
- Running **analytical workloads** on structured data.
- Creating **high-availability** database environments with failover.

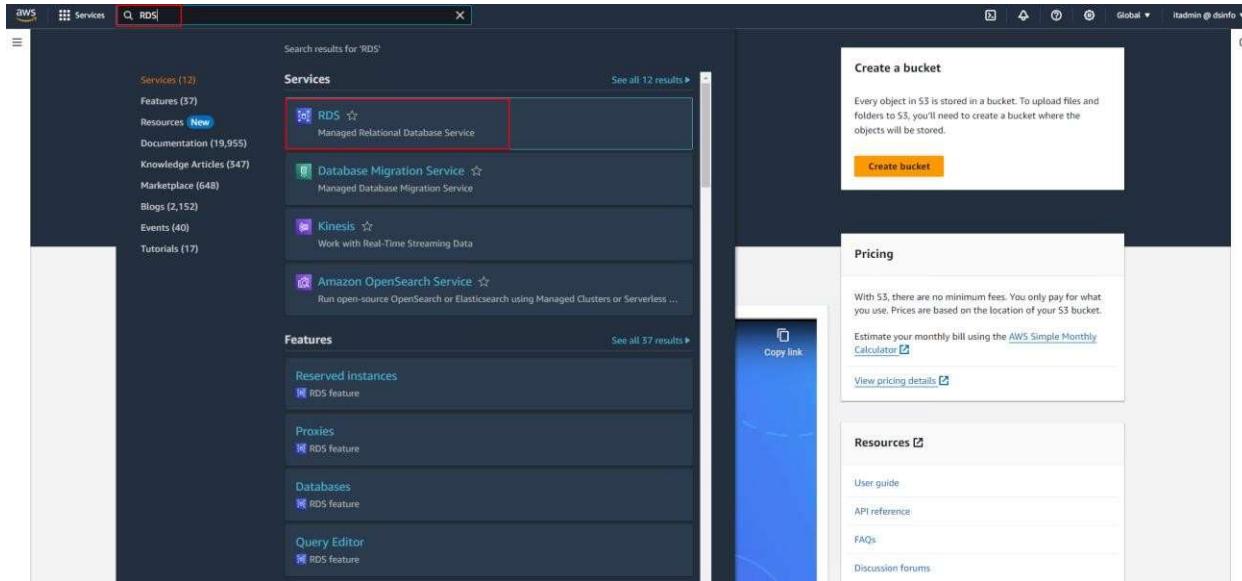
With AWS RDS, you don't need to worry about the complexity of managing database software or infrastructure — AWS takes care of it for you.



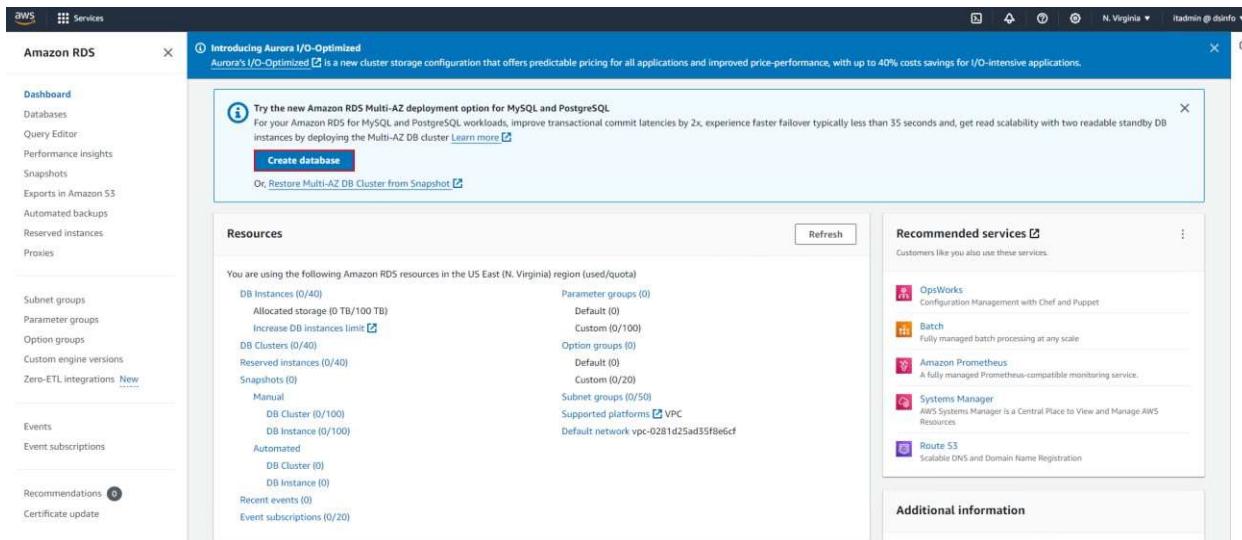
How to Set Up AWS RDS

The **RDS** service supports various types of databases.

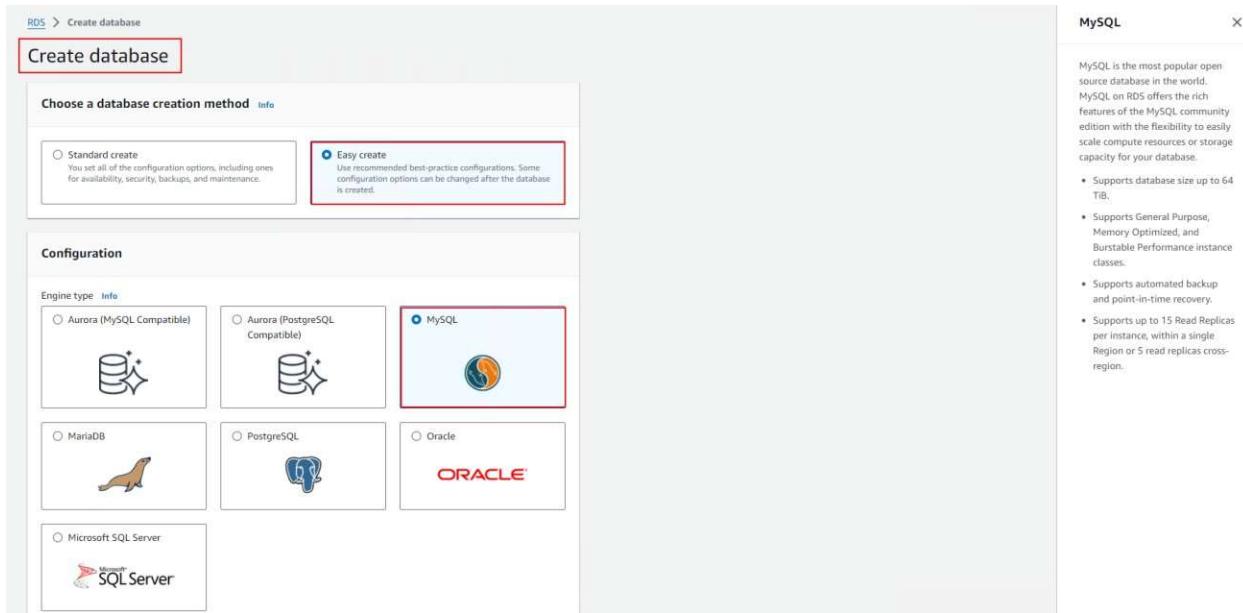
To use **RDS**, type **RDS** in the search bar and then click on **RDS** from the results.



To create a database using **RDS**, click on the **Create Database** button.



At this stage, you can select your desired **database engine**.



RDS > Create database

Create database

Choose a database creation method Info

Standard create You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Configuration

Engine type Info

Aurora (MySQL Compatible) 

Aurora (PostgreSQL Compatible) 

MySQL 

MariaDB 

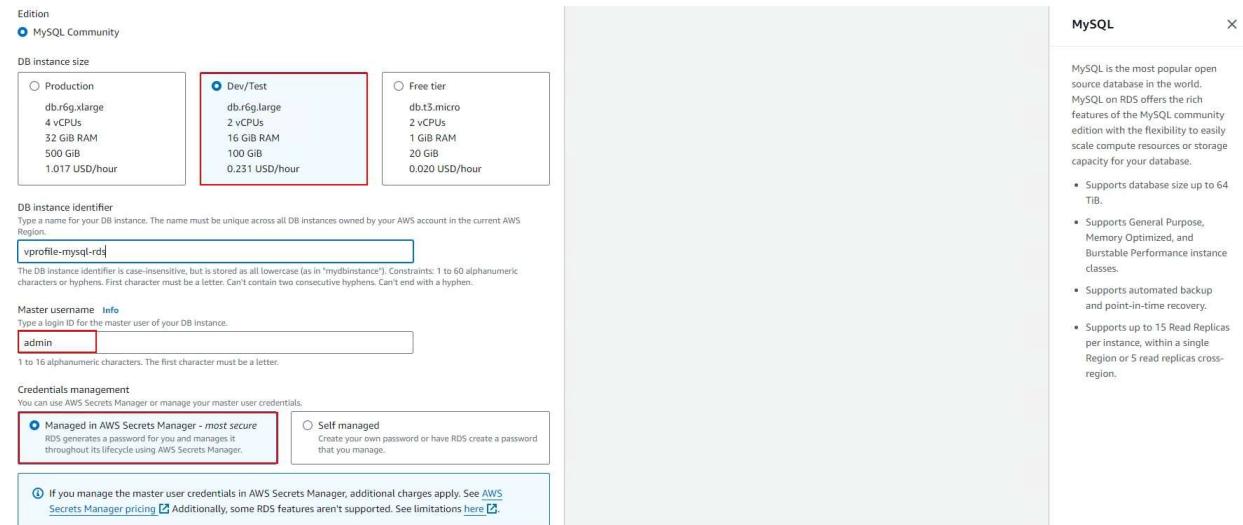
PostgreSQL 

Oracle 

Microsoft SQL Server 

MySQL

At this stage, you need to specify the required **resources** for the database, along with the **username** and **password** for database access.



Edition Info

MySQL Community

DB instance size

Production db.r6g.xlarge 4 vCPUs 32 GiB RAM 500 GiB 0.017 USD/hour

Dev/Test db.r6g.large 2 vCPUs 16 GiB RAM 100 GiB 0.231 USD/hour

Free tier db.t3.micro 2 vCPUs 1 GiB RAM 20 GiB 0.020 USD/hour

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

vprofile-mysql-rds

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username Info

Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management

You can use AWS Secrets Manager or manage your master user credentials.

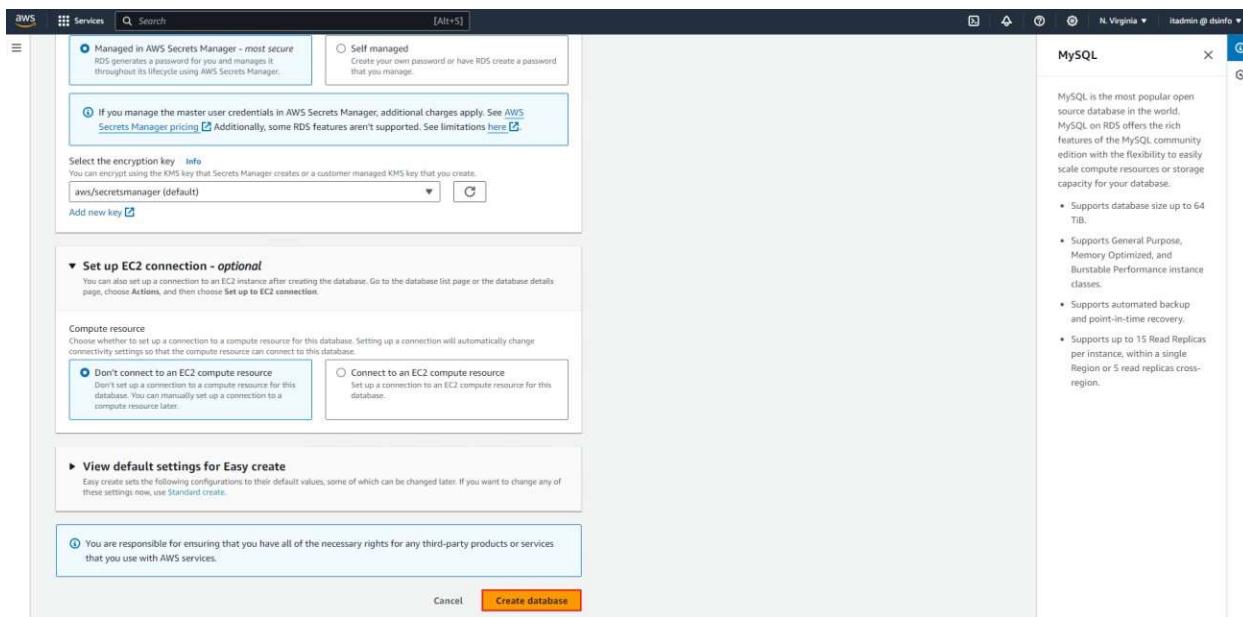
Managed in AWS Secrets Manager - most secure RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

Self managed Create your own password or have RDS create a password that you manage.

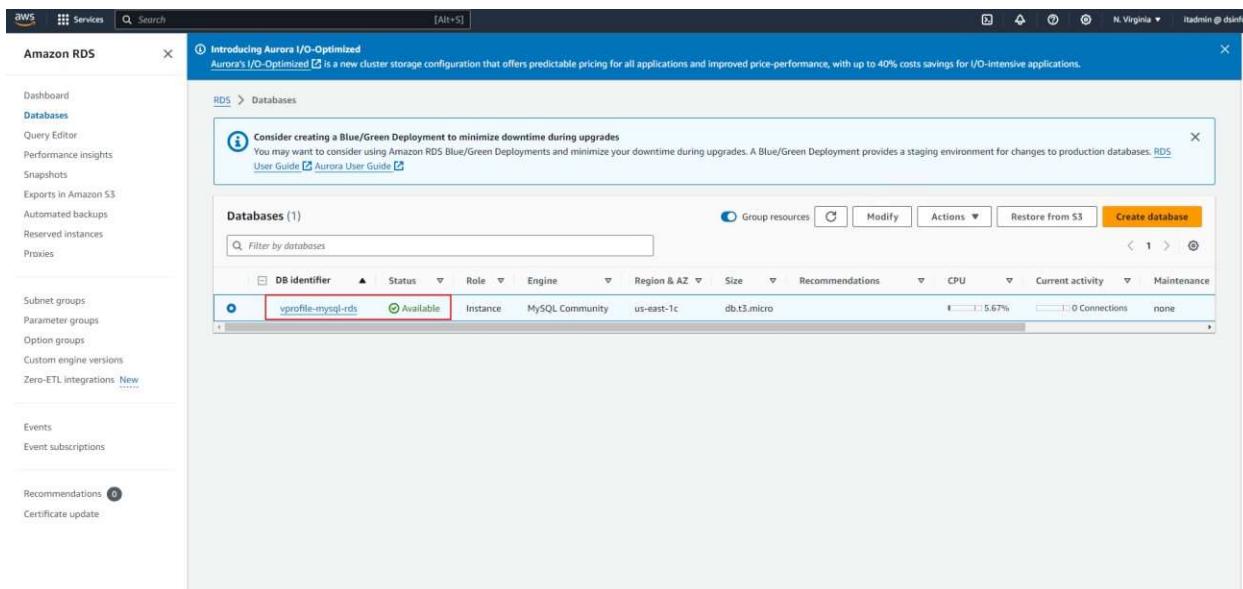
If you manage the master user credentials in AWS Secrets Manager, additional charges apply. See [AWS Secrets Manager pricing](#). Additionally, some RDS features aren't supported. See limitations [here](#).

MySQL

And finally, click on the **Create Database** button.



As shown in the image below, our desired **database has been created**. Now, click on your database to configure its **advanced settings**.



In this section, you can specify the **amount of resources** required for the database, the **type of storage**, and its **size**.

The screenshot shows two configuration sections for an AWS RDS instance.

Instance configuration: This section allows selecting the DB instance class. The 'Burstable classes (includes t classes)' option is selected and highlighted with a red box. Below it, the chosen class is shown: db.t3.micro, with details: 2 vCPUs, 1 GB RAM, Network: 2,085 Mbps.

Storage: This section allows defining storage type and allocated storage size. The 'General Purpose SSD (gp2)' option is selected and highlighted with a red box. The allocated storage is set to 20 GiB. A note at the bottom states: "Provisioning less than 100 GiB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance." A link to "Learn more" is provided.

In this section, you can define the **maximum size** of the database and specify whether you want your database to be **replicated across multiple Availability Zones** or not.

The screenshot shows two configuration sections for an AWS RDS instance.

Storage autoscaling: This section enables storage scaling. The "Enable storage autoscaling" checkbox is checked and highlighted with a red box. The maximum storage threshold is set to 1000 GiB. A note states: "Enabling this feature will allow the storage to increase after the specified threshold is exceeded." Another note specifies: "Charges will apply when your database autoscales to the specified threshold." A link to "Learn more" is provided.

Availability & durability: This section defines the deployment setup. The "Do not create a standby instance" option is selected and highlighted with a red box. Other options include "Create a standby instance (recommended for production usage)" which creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

In this section, you can configure **monitoring** for the database.

Database authentication options - [Info](#)

Password authentication
Authenticates using database passwords.

Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

Monitoring

Enable Enhanced Monitoring
Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Granularity
60 seconds

Monitoring Role
rds-monitoring-role
Clicking "Create database" will authorize RDS to create the IAM role rds-monitoring-role

Additional configuration
Database options, backup turned on, Enhanced Monitoring turned on, maintenance, CloudWatch Logs, delete protection turned off

Database options

DB parameter group [Info](#)
default.mysql8.0

Option group [Info](#)
default:mysql-8-0

In this section, you can specify the **version** of your database.

Additional configuration
Database options, backup turned on, Enhanced Monitoring turned on, maintenance, CloudWatch Logs, delete protection turned off

Database options

DB parameter group [Info](#)
default.mysql8.0

Option group [Info](#)
default:mysql-8-0

Backup

Enable automated backups
Creates a point-in-time snapshot of your database.

Backup retention period [Info](#)
The number of days (1-35) for which automatic backups are kept.
7 days

Backup window [Info](#)
The daily time range (in UTC) during which RDS takes automated backups.

Choose a window
 No preference

Start time
06 : 43 UTC

Duration
0.5 hours

Copy tags to snapshots

In this section, you need to configure the **logging settings**, and then click the **Continue** button.

The screenshot shows the 'Log exports' and 'Maintenance' sections of the AWS RDS Modify DB Instance configuration page. The 'Log exports' section has checkboxes for Audit log, Error log, General log, and Slow query log, all of which are checked. The 'Maintenance' section includes an 'Auto minor version upgrade' checkbox (checked) and a 'DB instance maintenance window' configuration (Start day: Friday, Start time: 08:33 UTC, Duration: 0.5 hours). A 'Delete protection' checkbox is also present but unchecked. At the bottom right are 'Cancel' and 'Continue' buttons, with 'Continue' being highlighted.

And at this stage, you need to click on the **Modify DB Instance** button.

The screenshot shows the 'Summary of modifications' section of the AWS RDS Modify DB instance page. It lists a modification: 'Enable publish to cloudWatch logs' with a current value of 'Audit log,Error log,General log,Slow query log'. Below this is the 'Schedule modifications' section, which shows the 'When to apply modifications' dropdown set to 'Apply during the next scheduled maintenance window' (March 29, 2024 01:13 - 02:01 (UTC-07:00)). There is also an 'Apply immediately' option. At the bottom are 'Cancel', 'Back', and 'Modify DB instance' buttons, with 'Modify DB instance' being highlighted.

By selecting the database and using the **Actions** menu, you can perform a variety of tasks on your database.

The screenshot shows the AWS RDS console interface. On the left, there's a sidebar with various navigation options like Dashboard, Databases, Query Editor, etc. The main area shows a table of databases, with one row selected: 'vprofile-mysql-rds' (Status: Available). A context menu is open over this row, with 'Actions' highlighted. The menu contains several options: Quick Actions - New (which is currently selected), Convert to Multi-AZ deployment, Stop temporarily, Reboot, Delete, Set up EC2 connection, Set up Lambda connection, Create read replica, Create Aurora read replica, Create Blue/Green Deployment - new, Promote, Take snapshot, Restore to point in time, Migrate snapshot, Create zero-ETL integration, Create RDS Proxy, and Create ElastiCache cluster - new. A success message at the top says 'Successfully modified vprofile-mysql-rds.'