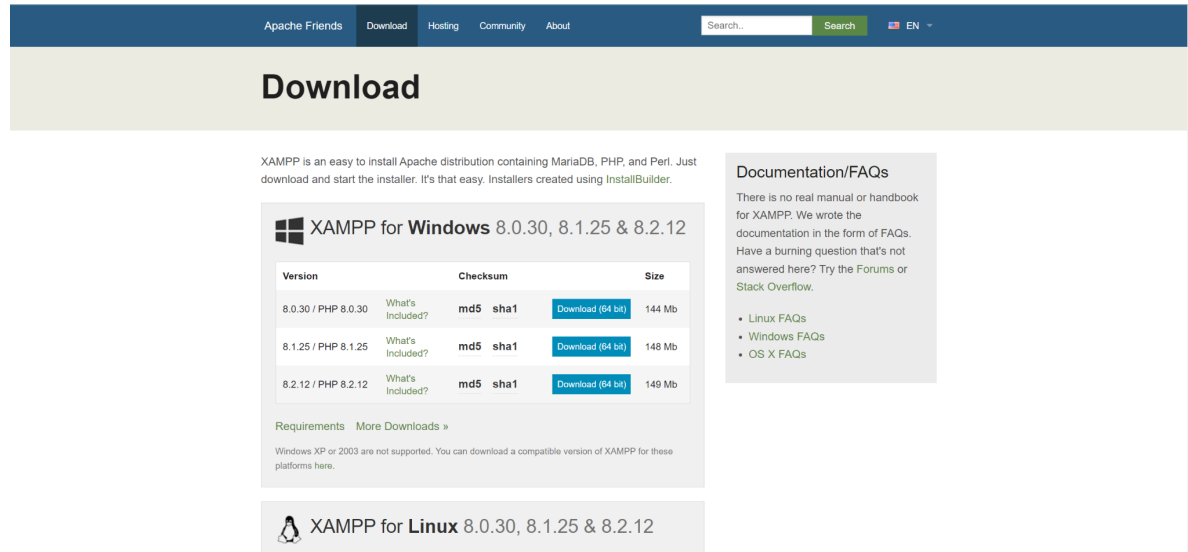


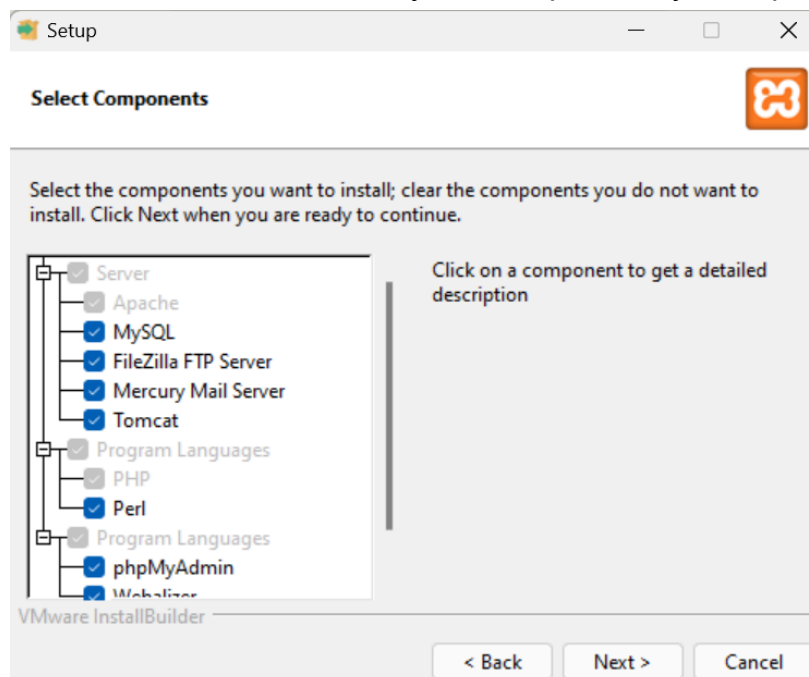
## 1a]Website Hosted Locally on XAMPP

**Step1:** To download the latest version of XAMPP (8.2.12 / PHP 8.2.12) for Windows, visit the official XAMPP website, navigate to the download section, and select the appropriate version for your system.

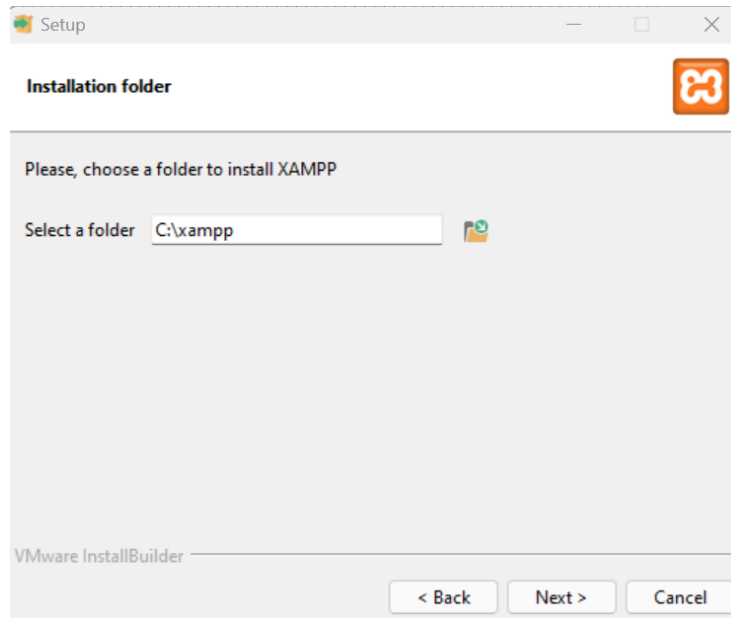
<https://www.apachefriends.org/download.html>



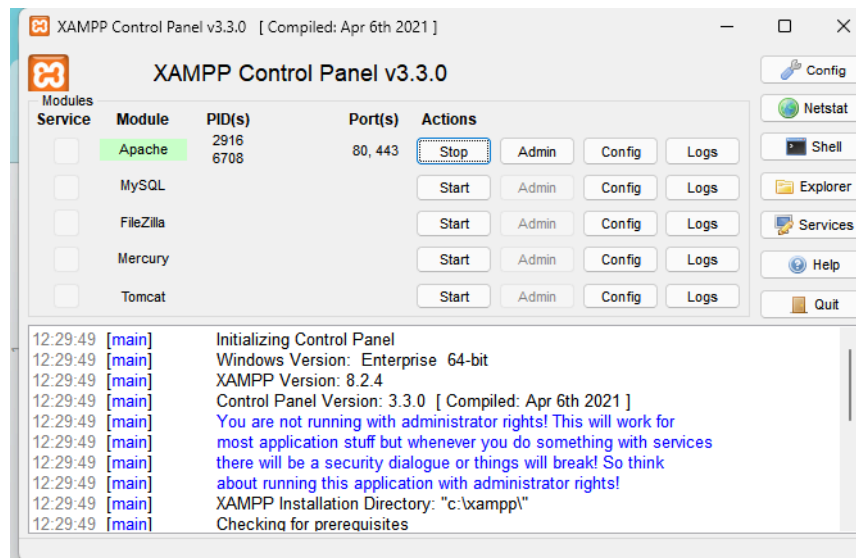
**Step 2:** XAMPP offers various components during installation, including Apache (web server), MySQL (database management), PHP (server-side scripting language), and more. These components are essential for running a local development environment. You can choose to install only the components you require for your specific project.



**Step 3:** Choose the destination folder where you want to install XAMPP and store your local files. This path will be where all your web server files and configurations are saved.

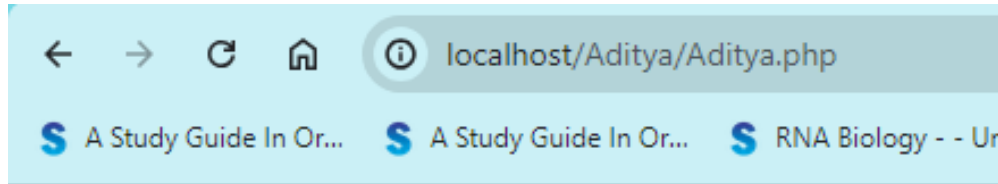


**Step 4:** After successfully downloading XAMPP, open the XAMPP Control Panel, and under the "Module" section, click "Start" next to Apache to activate your local server.



**Step 5:** Go to the following directory and save your files there so that they can be locally hosted

Step 6: Open your web browser and navigate to `localhost`, followed by your folder name (e.g., `localhost/your-folder-name`). You will see a list of available PHP files—select one to run it, and it will be hosted successfully.

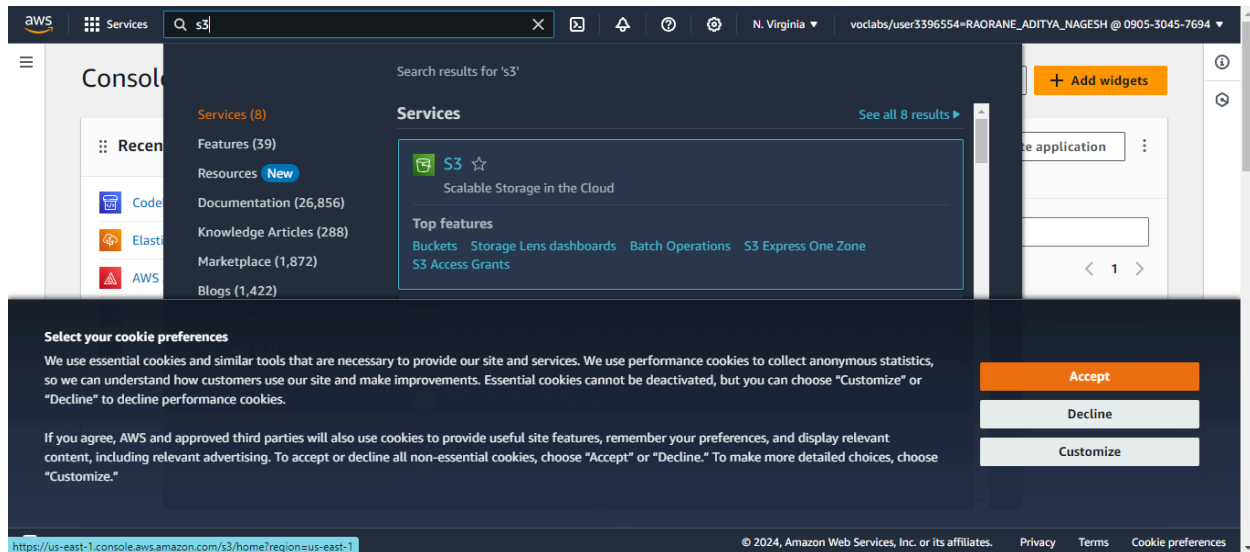


# Welcome

I am Aditya Raorane

## 1a] Website Hosted Remotely on AWS S3 Bucket

Step 1: In your AWS Academy account, navigate to the "Services" search bar, type "S3" and open it to access the Amazon Simple Storage Service.



Step 2: In AWS S3, a bucket is a container for storing objects, with globally unique names and customizable configurations for data management and access control. Click on "Create Bucket" to create one.

The screenshot shows the Amazon S3 console interface. On the left is a navigation menu with options like Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings, Storage Lens, Dashboards, Storage Lens groups, and AWS Organizations settings. The main content area is titled 'Amazon S3' and features an 'Account snapshot' section with a 'View Storage Lens dashboard' button. Below this, there are tabs for 'General purpose buckets' (selected) and 'Directory buckets'. The 'General purpose buckets' section shows a list of buckets. A table lists one bucket:

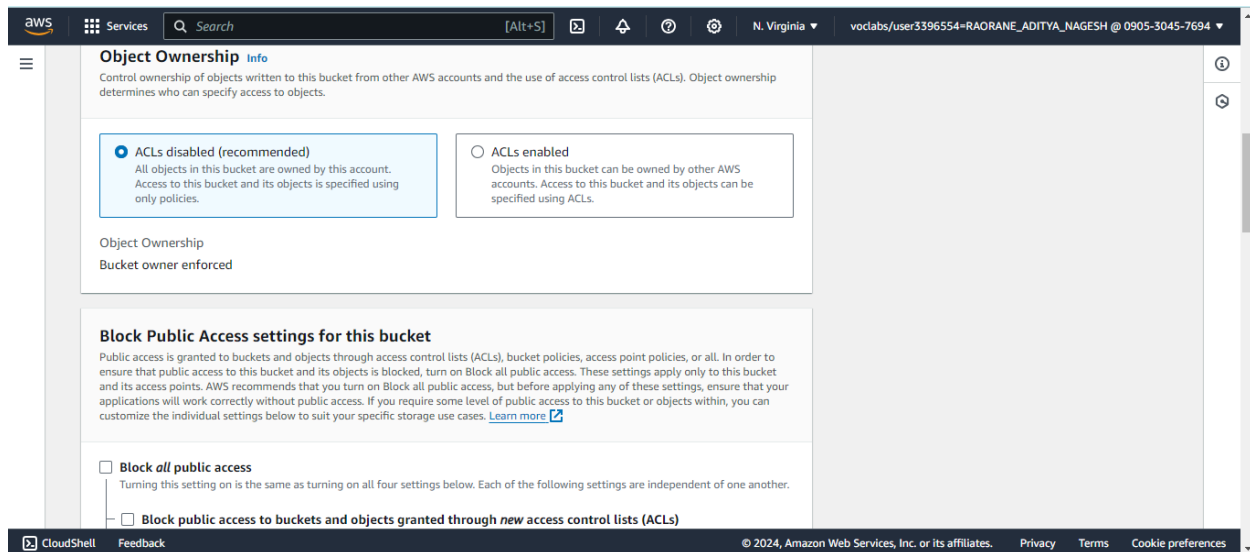
Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">elasticbeanstalk-us-east-1-090530457694</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	August 5, 2024, 14:00:29 (UTC+05:30)

At the bottom of the console, there is a footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

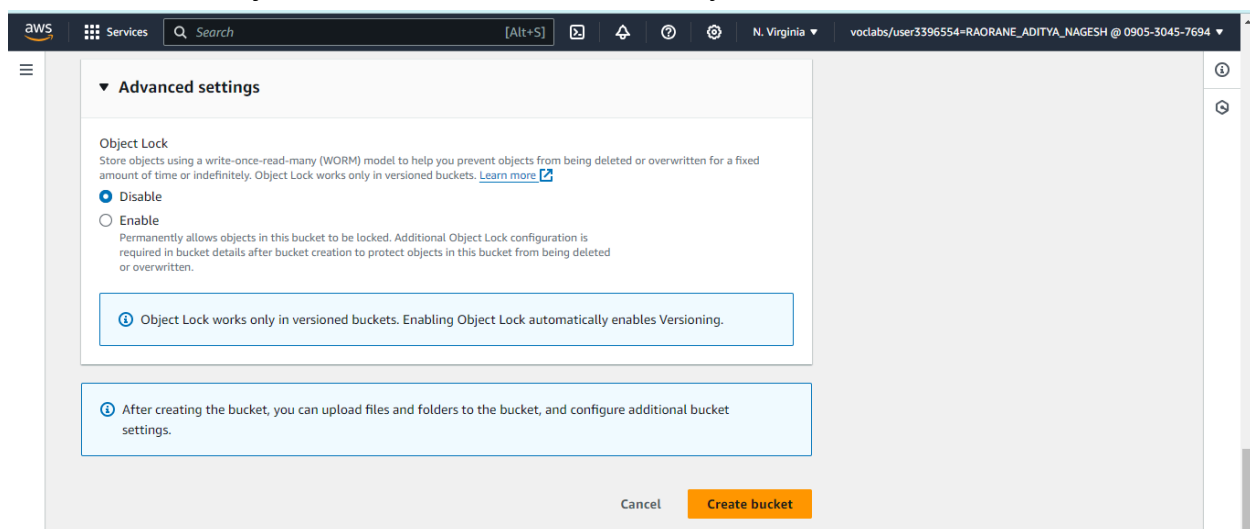
**Step 3:** Under "Bucket Type," select **General Purpose** because creating a bucket sets up a new storage container with a unique name and configurations. If you select **Directory**, it will organize files within an existing bucket, making it easier to manage and retrieve specific files, but it won't create a new storage container.

The screenshot shows the 'Create bucket' page in the Amazon S3 console. The page title is 'Create bucket' with an 'Info' link. Below the title, it says 'Buckets are containers for data stored in S3.' The 'General configuration' section is visible, showing the 'AWS Region' as 'US East (N. Virginia) us-east-1'. Under 'Bucket type', there are two options: 'General purpose' (selected) and 'Directory - New'. The 'General purpose' option is highlighted with a blue border and contains the text: 'Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.' The 'Directory - New' option is also visible with its description. Below the bucket type selection, there is a 'Bucket name' field with the value 'www.raorane.com'. At the bottom, there is a section for 'Copy settings from existing bucket - optional'.

**Step 4:** Under "Object Ownership," select ACLs disabled to ensure that all objects are owned by the bucket owner, providing simplified permissions management and improved security.



**Step 5:** Under "Object Lock," I chose **Disable** to allow unrestricted deletion and modification of objects within the bucket. And finally click on "Create Bucket".



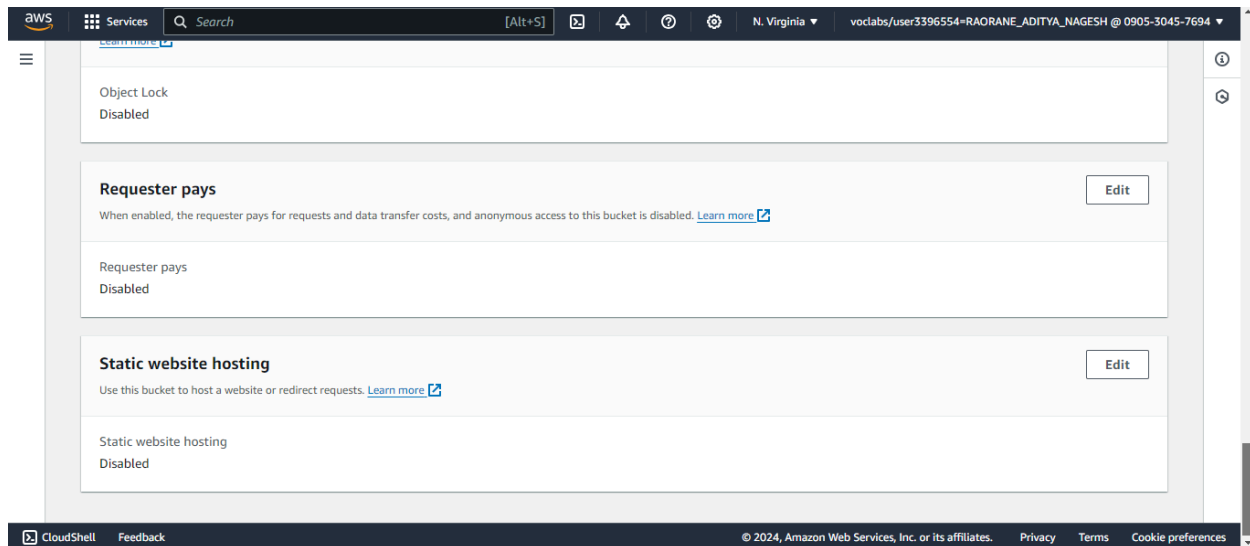
Step 6: Finally, once the bucket is created, click on the bucket link to view its properties and configuration details.

The screenshot displays the AWS Management Console interface. At the top, a green notification banner states: "Successfully created bucket 'www.raorane.com'. To upload files and folders, or to configure additional bucket settings, choose [View details](#)." Below this, the "General purpose buckets" section is active, showing a list of buckets. The bucket "www.raorane.com" is selected. The "Properties" tab for this bucket is open, showing the "Bucket overview" section with the following details:

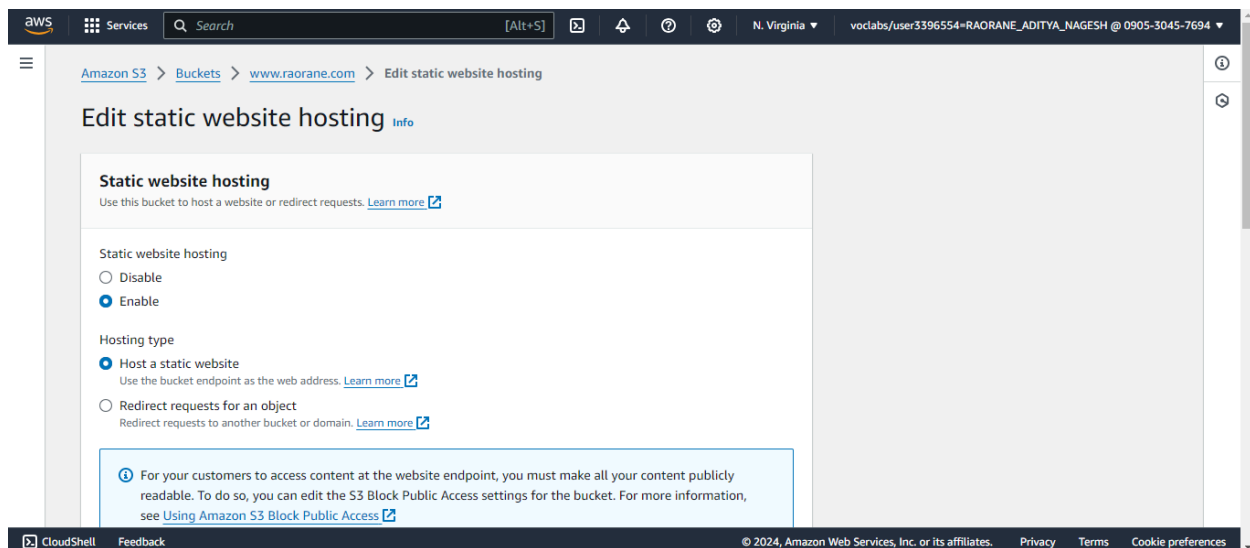
Property	Value
AWS Region	US East (N. Virginia) us-east-1
Amazon Resource Name (ARN)	arn:aws:s3::www.raorane.com
Creation date	August 12, 2024, 15:12:44 (UTC+05:30)

Below the overview, the "Bucket Versioning" section is shown, indicating that "Bucket Versioning" is "Disabled".

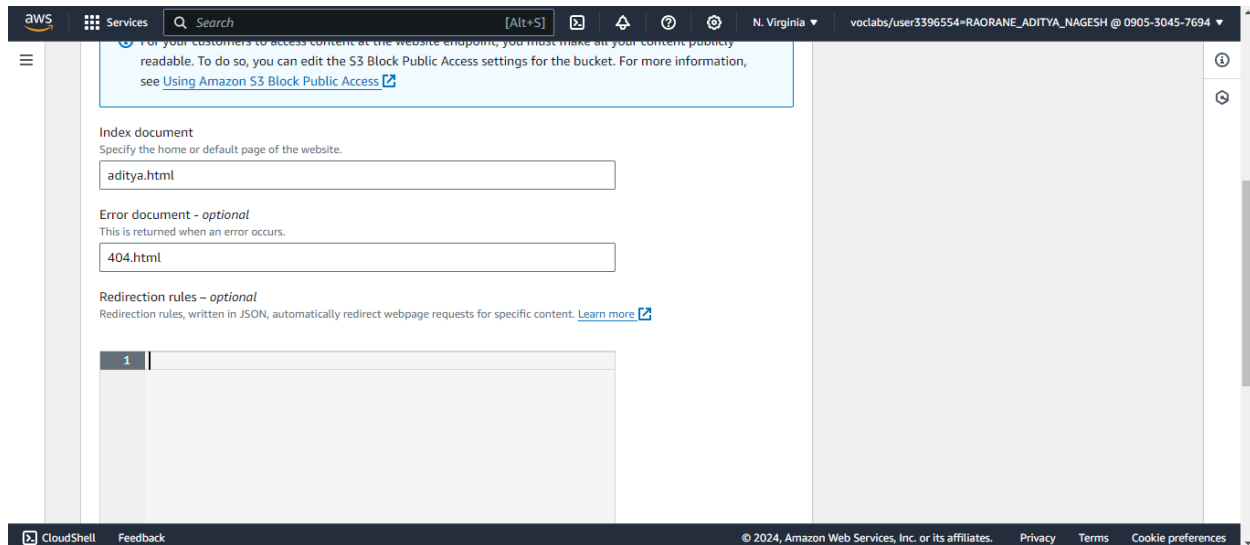
**Step 7:** Search for Static Website Hosting; it is disabled by default but needs to be enabled to serve static web content from your bucket. Enabling Static Website Hosting allows your S3 bucket to serve static content, such as HTML files. You must specify an index document (e.g., index.html) and optionally an error document. This configuration makes your bucket accessible via a web URL, hosting your static website directly



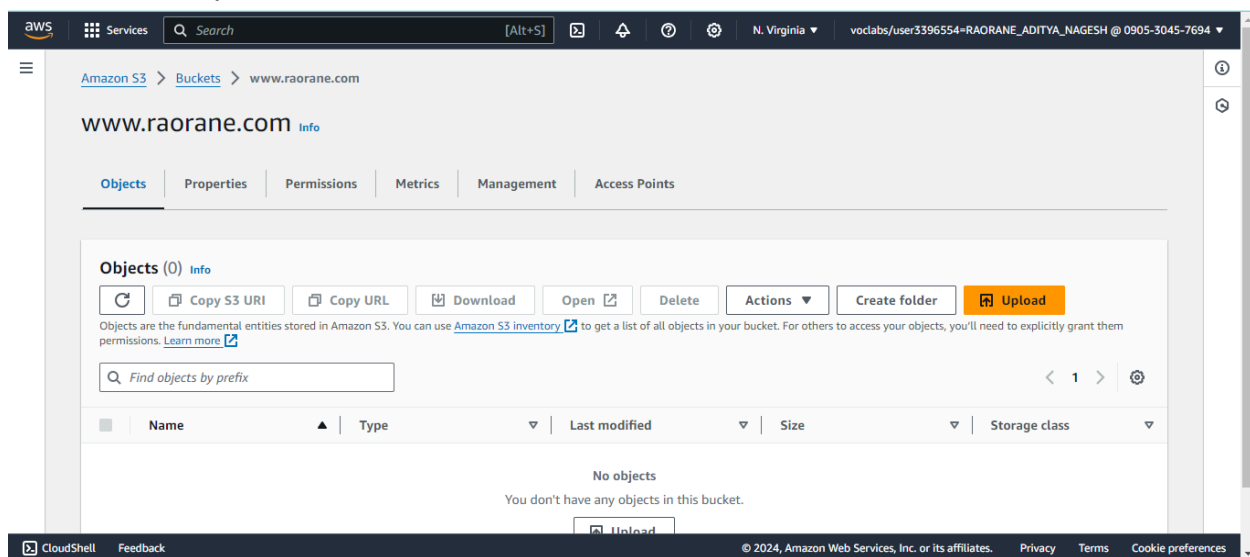
**Step 8:** Enable static website hosting option and select host type as a “static website”



**Step 9:** Specify the index document (e.g., index.html) and the error document (e.g., error.html) along with the desired HTTP error code to display custom error messages.



Step 10: Under the “Objects” section, you will see the page to upload your HTML files. Click on the “Upload” button.





Step 11: Add your file(s) and click the **Upload** button. You will see a confirmation indicating that the file was successfully uploaded.

The screenshot displays the AWS CloudShell 'Upload' interface. The top section shows the 'Upload' title and instructions. Below this is a dashed box for dragging files. The 'Files and folders' section shows a table with one file, 'aditya.html', which is 4.2 KB and of type 'text/html'. The 'Destination' section shows the upload path 's3://www.raorane.com'. The bottom section shows a green banner indicating 'Upload succeeded' with a link to view details. Below the banner, a table shows the upload details for 'aditya.html', including its size (4.2 KB) and status (Succeeded).

**Upload** Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

**Files and folders** (1 Total, 4.2 KB)  
All files and folders in this table will be uploaded.

<input type="checkbox"/>	Name	Folder	Type
<input type="checkbox"/>	aditya.html	-	text/html

**Destination** Info

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Upload succeeded**  
View details below.

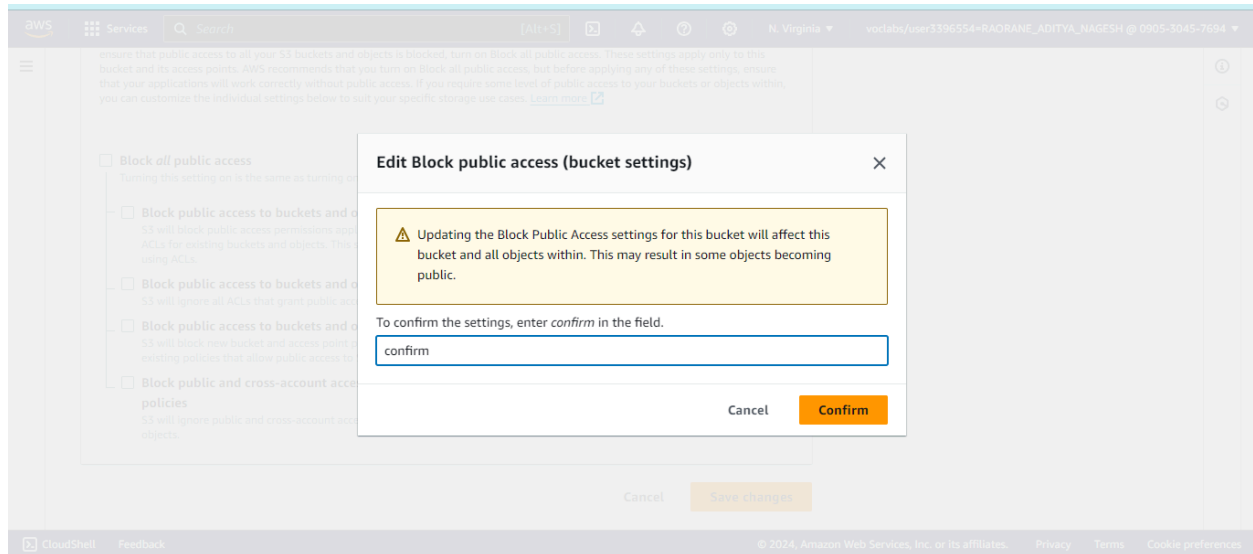
Destination	Succeeded	Failed
s3://www.raorane.com	1 file, 4.2 KB (100.00%)	0 files, 0 B (0%)

**Files and folders** (1 Total, 4.2 KB)

Name	Folder	Type	Size	Status	Error
<a href="#">aditya.html</a>	-	text/html	4.2 KB	Succeeded	-

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Step 12:** Now, if you click on the link, it will show an error 403 Forbidden message due to block policies. To resolve this, you need to configure the bucket's public access settings by unchecking all block public access options.

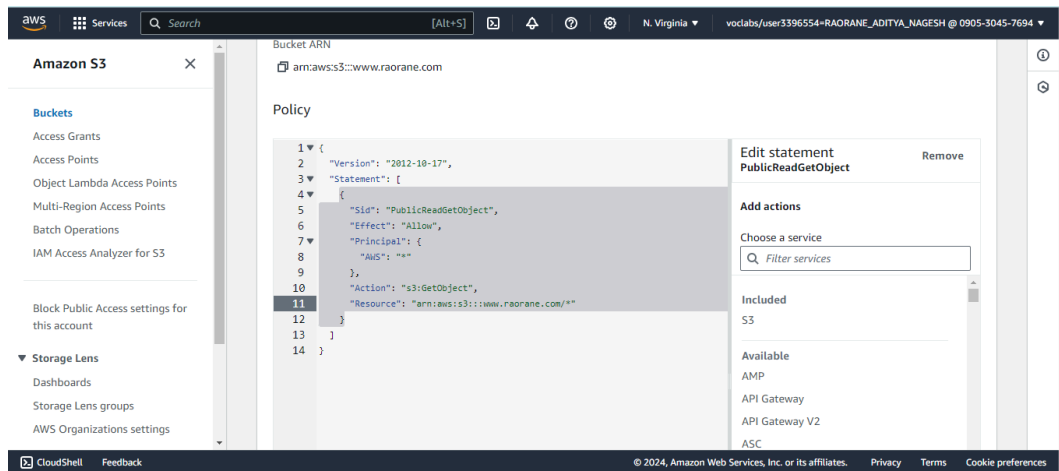


**Step 13:** Scroll down to the **Bucket Policy** section and paste the policy from the following link:

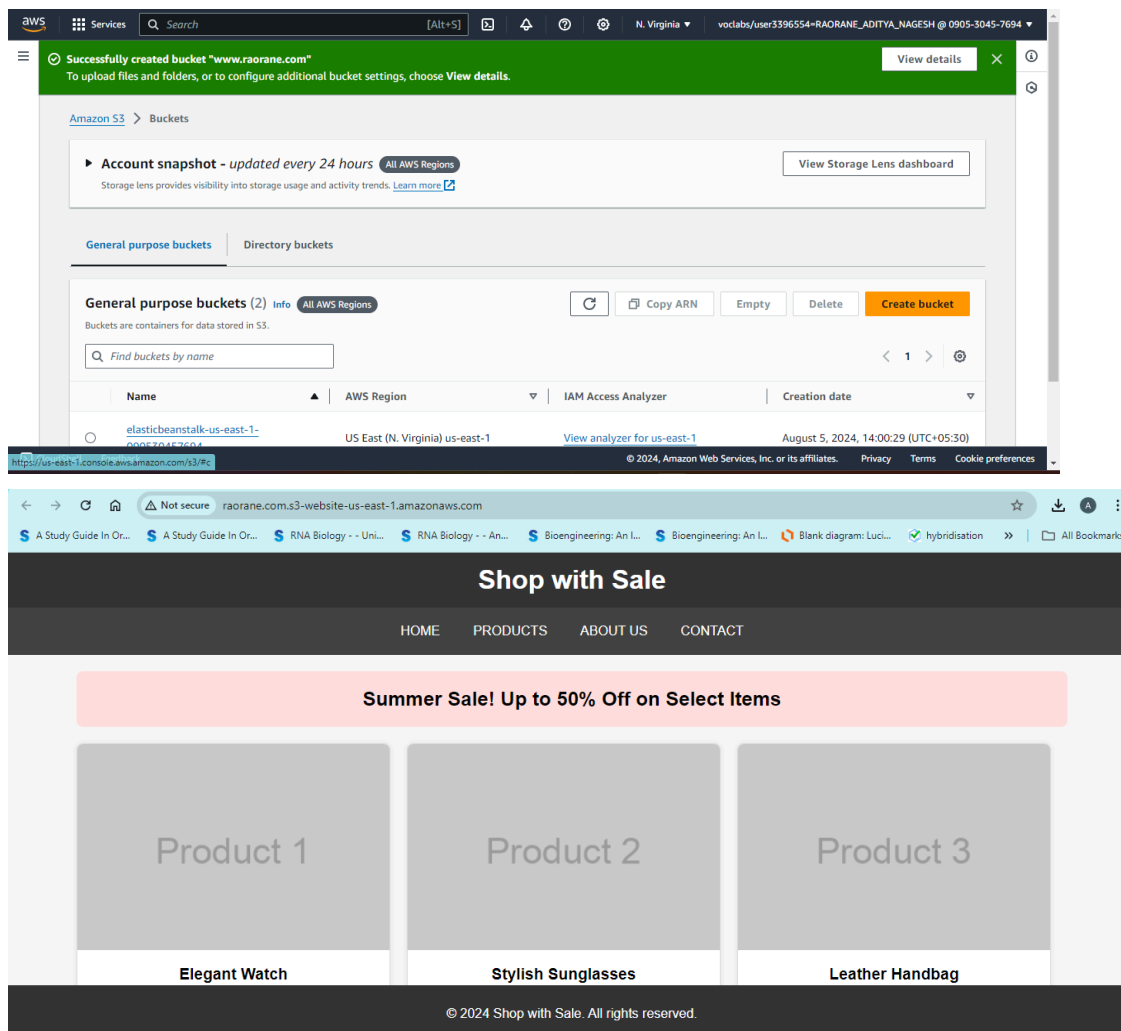
<https://gist.github.com/Savjee/b4b3a21d143a30e7dc07>

To configure access permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::www.raorane.com"
    }
  ]
}
```

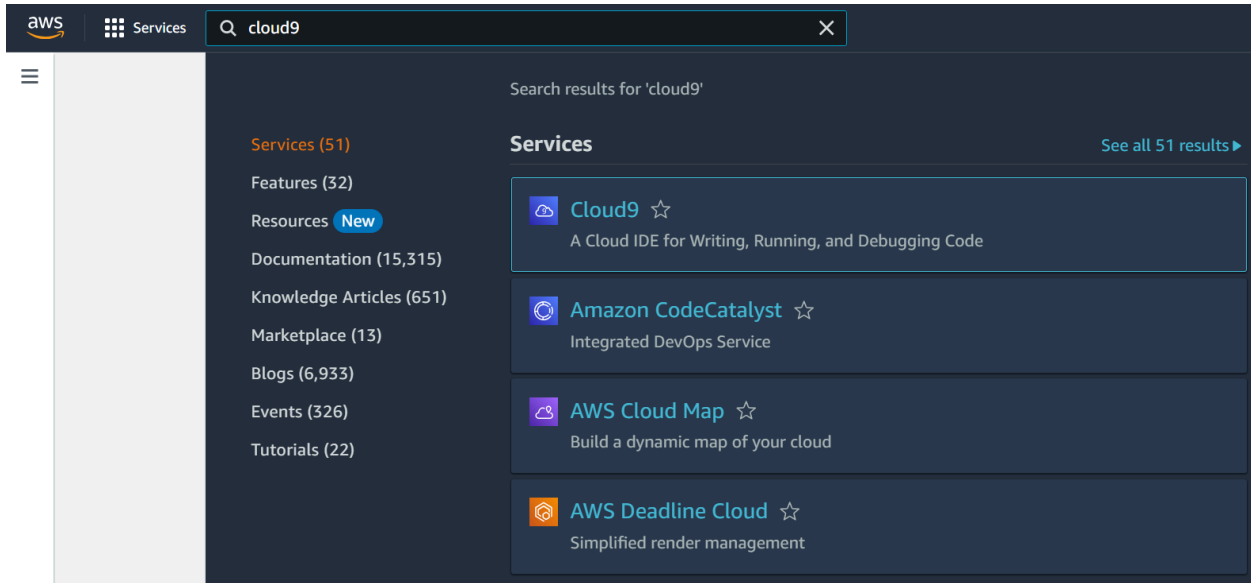


**Step 14:** Finally, you will see a confirmation about the policies. The website is now ready to run—click on the provided link to view the live website.



## 1b)Cloud9 IDE Collaborative Setup

Step 1:In your AWS Academy account, navigate to the "Services" search bar, type "Cloud9" and open it to access the Cloud9 IDE.



Step 2:To set up an environment in AWS Cloud9, create a **new Cloud9 environment** in the AWS Management Console, choosing your preferred instance type and VPC settings. Select "EC2 instance" in the environment type.

A screenshot of the AWS Cloud9 'Create environment' page. The page title is 'Create environment' with an 'Info' link. Below the title, there is a 'Details' section. The 'Name' field is filled with 'Aditya Raorane'. Below it, a note says 'Limit of 60 characters, alphanumeric, and unique per user.' The 'Description - optional' field is filled with 'First experiment'. Below it, a note says 'Limit 200 characters.' The 'Environment type' section has two options: 'New EC2 instance' (selected with a radio button) and 'Existing compute' (unselected with a radio button). Below 'New EC2 instance', a note says 'Cloud9 creates an EC2 instance in your account. The'. Below 'Existing compute', a note says 'You have an existing instance or server that you'd like to'.

## Instance type

- ☒ **t2.micro (1 GiB RAM + 1 vCPU)**  
Free-tier eligible. Ideal for educational users and exploration.
- ☐ **t3.small (2 GiB RAM + 2 vCPU)**  
Recommended for small-sized web projects.
- ☐ **m5.large (8 GiB RAM + 2 vCPU)**  
Recommended for production and general-purpose development.
- ☐ **Other instance type**  
Select an instance type.

t3.nano

## Platform

- ☒ **Amazon Linux 2 (recommended)**
- ☐ Amazon Linux AMI
- ☐ Ubuntu Server 18.04 LTS

## Cost-saving setting

Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After 30 minutes (default)

## IAM role

AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

AWSServiceRoleForAWSCloud9

## ► Network settings (advanced)

No tags associated with the resource.

Add new tag

You can add 50 more tags.

Cancel

Previous step

Next step

## Timeout

How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.

30 minutes

## Network settings Info

## Connection

How your environment is accessed.

- ☐ **AWS Systems Manager (SSM)**  
Accesses environment via SSM without opening inbound ports (no ingress).

- ☒ **Secure Shell (SSH)**  
Accesses environment directly via SSH, opens inbound ports.

## ► VPC settings Info

## ► Tags - optional Info

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.



## The following IAM resources will be created in your account

- **AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

Step 3: Finally the environment with EC2 instances is created.

[AWS Cloud9](#) > Environments

**Environments (1)** [Delete](#) [View details](#) [Open in Cloud9](#) [Create environment](#)

My environments ▼ < 1 > ⚙️

	Name ▲	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
<input type="radio"/>	<a href="#">Aditya Raorane</a>	<a href="#">Open</a>	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::090530457694:assumed-role/voclabs/user3396554=RAORANE_ADITYA_NAGESH

**Aditya Raorane** [Delete](#) [Open in Cloud9](#)

**Details** [Edit](#)

Name Aditya Raorane	Owner ARN arn:aws:sts::090530457694:assumed-role/voclabs/user3396554=RAORANE_ADITYA_NAGESH	Status ⌚ Creating
Description First experiment	Number of members 1	Lifecycle status ⌚ Creating
Environment type EC2 instance		

[EC2 instance](#) | [Network settings](#) | [Tags](#)

**EC2 instance** [Manage EC2 instance](#)

ARN arn:aws:cloud9:us-east-1:090530457694:environment:dcf40a1e1d824b2d9f0d4c0779f5a107	Instance type t2.micro (1 GiB RAM + 1 vCPU)
Platform Amazon Linux 2023	Storage EBS only

Step 4: Configure a username and password for a user and assign the role.

### Specify user details

#### User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , \_ @ - (hyphen)

☒ Provide user access to the AWS Management Console - *optional*  
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Console password

☐ Autogenerated password  
You can view the password after you create the user.

☒ Custom password  
Enter a custom password for the user.

☒ Show password

☒ 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.

**i** 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](#)

Cancel **Next**

**Timeout**  
How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.

30 minutes ▼

#### Network settings [Info](#)

**Connection**  
How your environment is accessed.

☒ **AWS Systems Manager (SSM)**  
Accesses environment via SSM without opening inbound ports (no ingress).

☐ **Secure Shell (SSH)**  
Accesses environment directly via SSH, opens inbound ports.

► **VPC settings** [Info](#)

► **Tags - optional** [Info](#)  
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.

**i** The following IAM resources will be created in your account

- **AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)
- **AWSCloud9SSMAccessRole** and **AWSCloud9SSMInstanceProfile** - A service role and an instance profile are automatically created if Cloud9 accesses its EC2 instance through AWS Systems Manager. If your environments no longer require EC2 instances that block incoming traffic, you can delete these

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

► VPC settings [Info](#)

► Tags - optional [Info](#)

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.

**The following IAM resources will be created in your account**

- **AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)
- **AWSCloud9SSMAccessRole** and **AWSCloud9SSMInstanceProfile** - A service role and an instance profile are automatically created if Cloud9 accesses its EC2 instance through AWS Systems Manager. If your environments no longer require EC2 instances that block incoming traffic, you can delete these roles using the AWS IAM console. [Learn more](#)

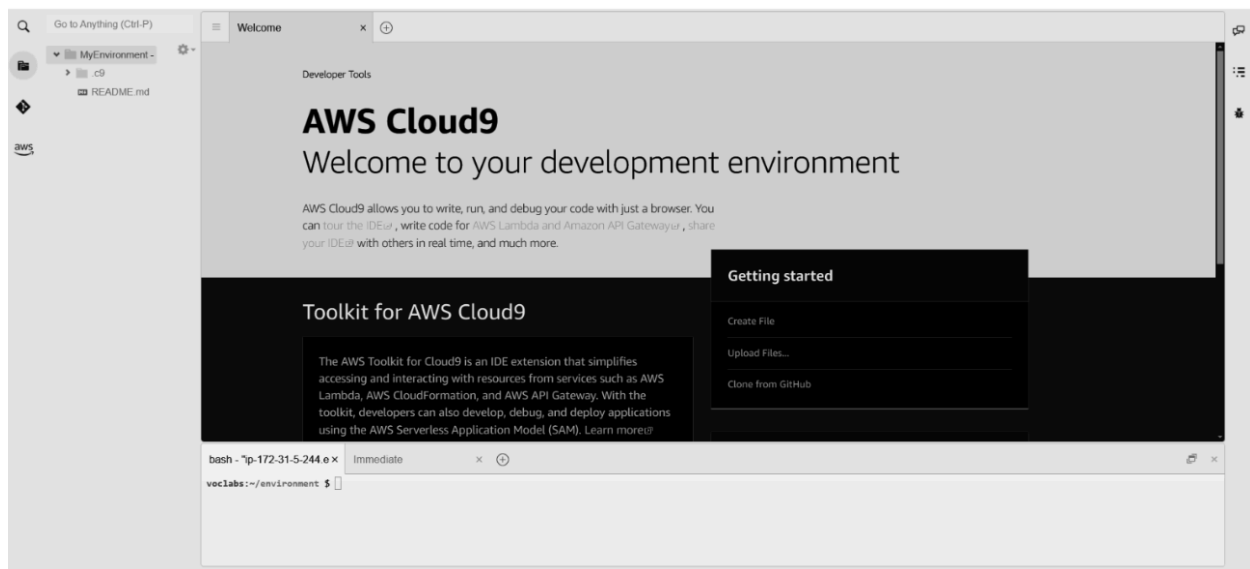
Cancel **Create**

❌ There was an error creating the IAM resources needed for SSM connection.

❌ You don't have the permission required to perform this operation. Ask your administrator to give you permissions.

Note: If you're unable to create a user with your AWS Academy account, it likely stems from limited permissions or role restrictions imposed to prevent unauthorized actions or costs. These accounts often have constraints tailored for educational purposes, so you may need to use a personal account or request assistance from your instructor.

### Step 5: Open Cloud9 IDE.



The screenshot shows the AWS Cloud9 IDE interface. The main window displays "AWS Cloud9 Welcome to your development environment". Below this, there's a "Toolkit for AWS Cloud9" section and a "Getting started" sidebar with options like "Create File", "Upload Files...", and "Clone from GitHub". At the bottom, a terminal window shows a bash prompt.



**Step 6: Open AWS IAM service. Configure a user and a group.****Specify user details**

**User details**

User name

AdityaRaorane

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)

☒ Provide user access to the AWS Management Console - *optional*  
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Console password

☐ Autogenerated password  
You can view the password after you create the user.

☒ Custom password  
Enter a custom password for the user.  
AdityaRaorane@44

☒ Show password

☒ 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.

❗

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](#)

Cancel

Next

IAM > Users > Create user

Step 1  
[Specify user details](#)

Step 2  
**Set permissions**

Step 3  
Review and create

**Set permissions**  
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

**Permissions options**

☒ **Add user to group**  
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ **Copy permissions**  
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☐ **Attach policies directly**  
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

▶ **Set permissions boundary - optional**

Cancel

Previous

Next

## Create user group ✕

Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

**User group name**  
Enter a meaningful name to identify this group.

myweb-app-group

Maximum 128 characters. Use alphanumeric and '+', '@', '-' characters.

### Permissions policies (947)

↺ Create policy

Filter by Type  
All ty... ▼

< 1 2 3 4 5 6 7 ... 48 > ⚙

<input type="checkbox"/>	Policy name	Type	Use...	Description
<input type="checkbox"/>	<a href="#">AdministratorAccess</a>	AWS managed ...	None	Provides full access to AWS services
<input type="checkbox"/>	<a href="#">AdministratorAcce...</a>	AWS managed	None	Grants account administrative perm
<input type="checkbox"/>	<a href="#">AdministratorAcce...</a>	AWS managed	None	Grants account administrative perm
<input type="checkbox"/>	<a href="#">AlexaForBusinessD...</a>	AWS managed	None	Provide device setup access to Alex
<input type="checkbox"/>	<a href="#">AlexaForBusinessF...</a>	AWS managed	None	Grants full access to AlexaForBusin
<input type="checkbox"/>	<a href="#">AlexaForBusinessG...</a>	AWS managed	None	Provide gateway execution access t
<input type="checkbox"/>	<a href="#">AlexaForBusinessLi...</a>	AWS managed	None	Provide access to Lifesize AVS devi
<input type="checkbox"/>	<a href="#">AlexaForBusinessP...</a>	AWS managed	None	Provide access to Poly AVS devices
<input type="checkbox"/>	<a href="#">AlexaForBusinessR...</a>	AWS managed	None	Provide read only access to AlexaFo
<input type="checkbox"/>	<a href="#">AmazonAPIGatewa...</a>	AWS managed	None	Provides full access to create/edit/c
<input type="checkbox"/>	<a href="#">AmazonAPICatap...</a>	AWS managed	None	Provides full access to invoke API

Cancel Create user group

**Step 7:** Once the user group is created click on the server link created next to the user.

myweb-app-group user group created.

IAM > Users > Create user

Step 1  
[Specify user details](#)

Step 2  
**Set permissions**

Step 3  
[Review and create](#)

Step 4  
[Retrieve password](#)

### Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

**Permissions options**

☒ **Add user to group**  
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ **Copy permissions**  
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☐ **Attach policies directly**  
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

**User groups (1)**

Search

<input type="checkbox"/>	Group name	Users	Attached policies	Created
<input type="checkbox"/>	myweb-app-group	0	-	2024-08-08 (Now)

▶ Set permissions boundary - optional

Cancel Previous Next

**Step 8:** Finally search for “AWSCloud9EnvironmentMember” policy and attach it.

Identity and Access Management (IAM)

Search IAM

Dashboard

▼ Access management

- User groups
- Users
- Roles
- Policies

Identity providers

Account settings

▼ Access reports

- Access Analyzer
- External access
- Unused access
- Analyzer settings

IAM > User groups > myweb-app-group

myweb-app-group info

Delete

Summary

Edit

User group name	Creation time	ARN
myweb-app-group	August 08, 2024, 21:35 (UTC+05:30)	arn:aws:iam:014498640047:group/myweb-app-group

Users Permissions Access Advisor

**Users in this group (0)**

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

Search

<input type="checkbox"/>	User name	Groups	Last activity	Creation time
--------------------------	-----------	--------	---------------	---------------

No resources to display

Remove Add users