

5CS022 Distributed and Cloud Systems Programming

Exploring AWS S3, AWS Amplify and Beanstalk Cloud Services

Overview

In this workshop you will explore 3 related Cloud Computing services with Amazon AWS: S3, AWS Amplify and Beanstalk. These 3 services are often used together to create cloud-based web application services using AWS.

Part 1 and Part 2 will show you how to deploy and publish a static website to the internet via AWS S3 and AWS Amplify, and Part 3 will show you how to deploy and publish a dynamic web application via Elastic Beanstalk.

You will need to use the Google Chrome web browser for this workshop.

Part 1. Creating an S3 Bucket to host a static web page

Follow these steps to create an Amazon Simple Storage Service (Amazon S3) bucket to host a static website.

A static website is fixed and displays the same content for each user. In contrast, a dynamic website uses advanced programming to provide user interaction and display different content depending on the user's selections.

Access the AWS Management Console

1. Go to <https://awsacademy.instructure.com/> and login.
2. Then go to "Modules" and then Learner Lab - Foundational Services.
3. To start the lab session, choose Start **Lab** in the upper-right corner of the page.
 - The lab session starts.
 - A timer displays in the upper-right corner of the page and shows the time remaining in the session.

Tip: To refresh the session length at any time, choose **Start Lab** again before the timer reaches 0:00.

4. Before continuing, wait until the lab environment is ready. The environment is ready when the lab details appear on the right side of the page and the circle icon next to the **AWS** link in the upper-left corner turns green.

AWS ●

```
eee_w_2753512@runweb109770:~$
```

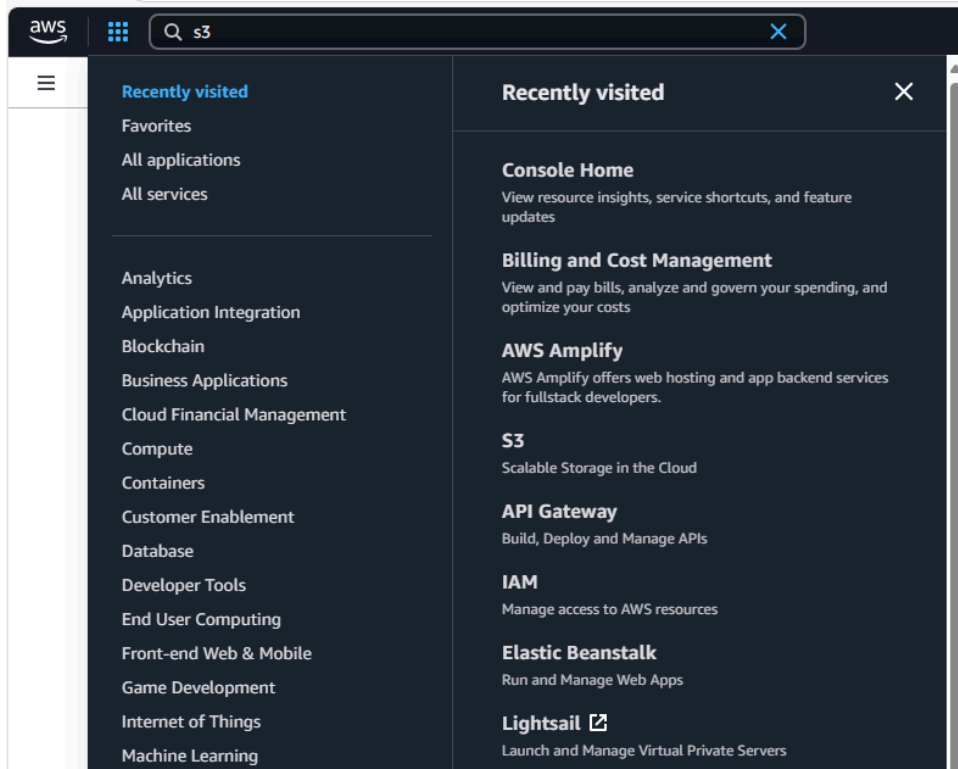
5. To connect to the AWS Management Console, choose the **AWS** link in the upper-left corner, above the terminal window.

A new browser tab opens and connects you to the AWS Management Console.

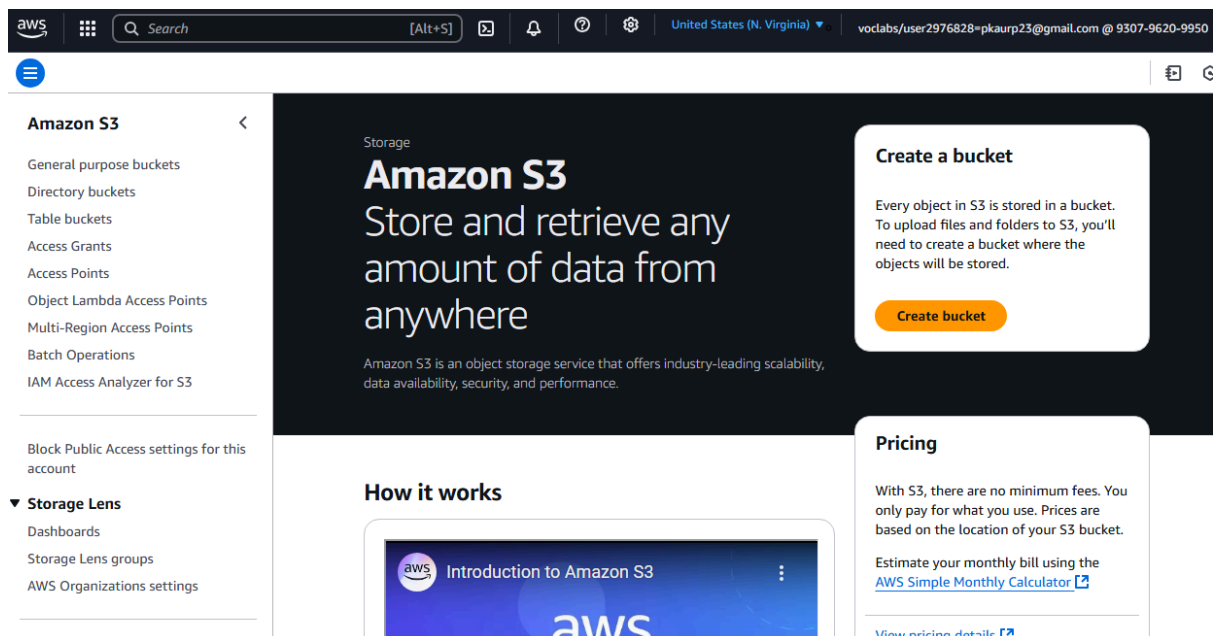
Tip: If a new browser tab does not open, a banner or icon is usually at the top of your browser with the message that your browser is preventing the site from opening pop-up windows. Choose the banner or icon, and then choose **Allow pop-ups**.

Task 1. Create an S3 bucket

4. Choose the **Services** menu, locate the **Storage** services, and select **S3** or just simply search S3 and select **S3 “Scalable Storage in the Cloud”**.



5. Select **Create bucket** on the right side of the page.



6. For **Bucket name**, enter a unique Domain Name System (DNS)-compliant name for your new bucket that is specific to you, for example, " pk5cs022"

These are the DNS naming guidelines:

- The name must be unique across all existing bucket names in Amazon S3.

- The name must only contain lowercase characters.
- The name must start with a letter or number.
- The name must be between 3 and 63 characters long.
- After you create the bucket, you cannot change the name, so choose wisely.


7. For Object Ownership, select **ACLs enabled**.

Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☐ **ACLs disabled (recommended)**
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.


☒ **ACLs enabled**
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

 We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

Object Ownership

☒ **Bucket owner preferred**
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

☐ **Object writer**
The object writer remains the object owner.

 If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. [Learn more](#)

8. Uncheck(clear) the **Block all public access** box because you want to be able to test if the website is working.

A warning message similar to **Turning off block all public access might result in this bucket and the objects within becoming public** appears below the security setting you deselected.

9. Below the warning, check the box next to **I acknowledge that...**

Block Public Access settings for this bucket

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 this bucket and its 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 this bucket 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.

 **Turning off block all public access might result in this bucket and the objects within becoming public**
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

10. Scroll to the bottom of the page, and select **Create bucket**.

Your new bucket appears in the **Buckets** list.

General purpose buckets (4) Info All AWS Regions

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
elasticbeanstalk-us-east-1-930796209950	US East (N. Virginia) us-east-1	View analyzer for us-east-1	January 23, 2025, 13:41:43 (UTC+00:00)
pk210125	US East (N. Virginia) us-east-1	View analyzer for us-east-1	January 21, 2025, 16:57:48 (UTC+00:00)
pk5cs022	US East (N. Virginia) us-east-1	View analyzer for us-east-1	January 23, 2025, 15:32:39 (UTC+00:00)
pooja123456	US East (N. Virginia) us-east-1	View analyzer for us-east-1	January 21, 2025, 12:32:36 (UTC+00:00)

Task 2. Add a bucket policy to make the content publicly available

11. Click on the link for your bucket's name, and then select the **Permissions** tab.

12. In the **Bucket policy** section, choose **Edit**.

13. To grant public read access for your website, copy the following bucket policy, and paste it in the policy editor, replacing the default one that is already there:

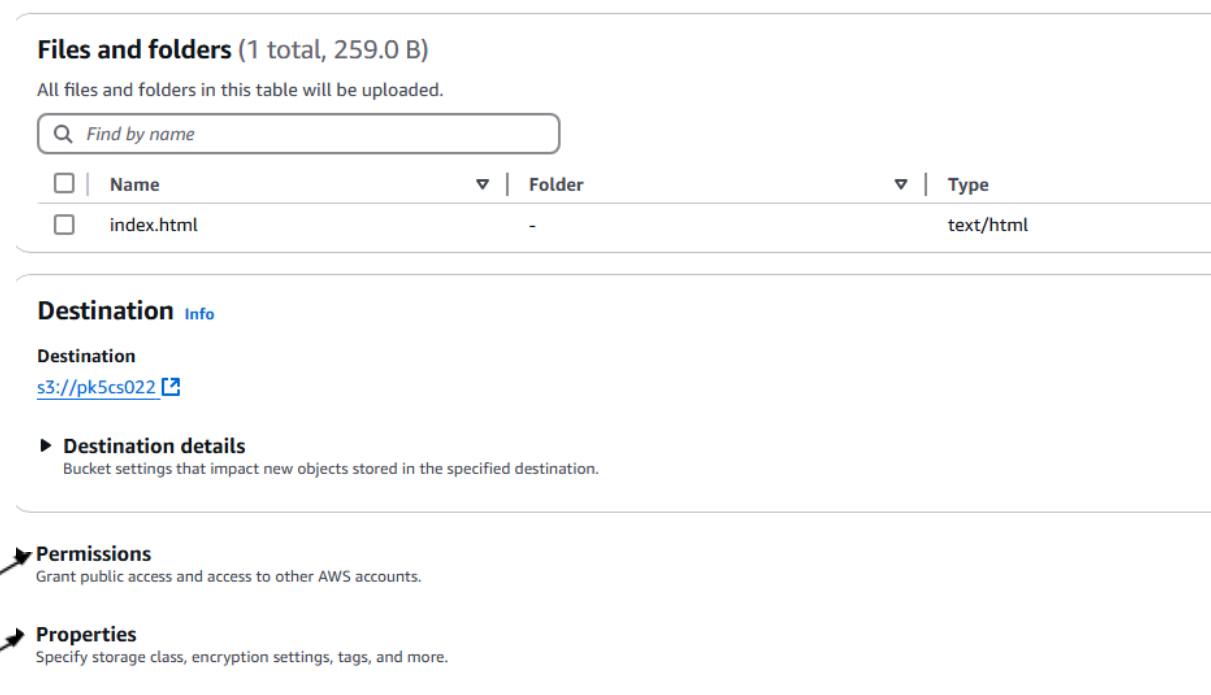
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3::: pk5cs022/*"
      ]
    }
  ]
}
```

14. Obviously, replace "pk5cs022" with the name of your own bucket.
15. Select **Save changes** at the bottom right of the page.

Task 3. Upload an HTML document

In this task, you upload an HTML document to your new bucket.

16. Download the sample HTML web page from:
https://drive.google.com/file/d/1wwk9FqoVpa_mmjNQqYLMCIGKoG7wV5mX/view?usp=sharing and save it as: **index.html**
17. In the S3 buckets console, choose the **Objects** tab.
18. Upload the index.html file to your bucket.
 - Choose **Upload**.
 - Drag and drop the index.html file onto the upload page.
 - As an alternative, choose **Add files**, navigate to the file, and choose **Open**.
19. Expand the **Permissions** section.



Files and folders (1 total, 259.0 B)

All files and folders in this table will be uploaded.

Find by name

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

Destination [Info](#)

Destination

[s3://pk5cs022](#)

► **Destination details**

Bucket settings that impact new objects stored in the specified destination.

► **Permissions**

Grant public access and access to other AWS accounts.

► **Properties**

Specify storage class, encryption settings, tags, and more.

20. Under **Predefined ACLs**, select **Grant public-read access**.

A warning message similar to **Granting public-read access is not recommend** appears below the setting you selected.

21. Below the warning, check the box next to **I understand....**

▼ **permissions**
Grant public access and access to other AWS accounts.

Access control list (ACL)
Grant basic read/write permissions to other AWS accounts. [Learn more](#)

① AWS recommends using S3 bucket policies or IAM policies for access control. [Learn more](#)

Access control list (ACL)

☒ Choose from predefined ACLs

☐ Specify individual ACL permissions

Predefined ACLs

☐ Private (recommended)
Only the object owner will have read and write access.

☒ Grant public-read access
Anyone in the world will be able to access the specified objects. The object owner will have read and write access. [Learn more](#)

⚠ **Granting public-read access is not recommended**
Anyone in the world will be able to access the specified objects. [Learn more](#)

☒ I understand the risk of granting public-read access to the specified objects.

22. Expand the **Properties** section.

This section lists the storage classes that are available in Amazon S3..

Ensure that the **Standard** storage class for general purpose is selected.

Specify storage class, retention settings, tags, and more.

Storage class Info								
Amazon S3 offers a range of storage classes designed for different use cases. Learn more or see Amazon S3 pricing								
	Storage class	Designed for	Bucket type	Availability Zones	Min storage duration	Min billable object size	Monitoring and auto-tiering fees	Retrieval fees
<input checked="" type="radio"/>	Standard	Frequently accessed data (more than once a month) with milliseconds access	General purpose	≥ 3	-	-	-	-
<input type="radio"/>	Intelligent-Tiering	Data with changing or unknown access patterns	General purpose	≥ 3	-	-	Per-object fees apply for objects >= 128 KB	-
<input type="radio"/>	Standard-IA	Infrequently accessed data (once a month) with milliseconds access	General purpose	≥ 3	30 days	128 KB	-	Per-GB fees apply
<input type="radio"/>	One Zone-IA	Recreatable, infrequently accessed data (once a month) with milliseconds access	General purpose or directory	1	30 days	128 KB	-	Per-GB fees apply
<input type="radio"/>	Glacier Instant Retrieval	Long-lived archive data accessed once a quarter with instant retrieval in milliseconds	General purpose	≥ 3	90 days	128 KB	-	Per-GB fees apply
<input type="radio"/>	Glacier Flexible Retrieval (formerly Glacier)	Long-lived archive data accessed once a year with retrieval of minutes to hours	General purpose	≥ 3	90 days	-	-	Per-GB fees apply
<input type="radio"/>	Glacier Deep Archive	Long-lived archive data accessed less than once a year with retrieval of hours	General purpose	≥ 3	180 days	-	-	Per-GB fees apply
<input type="radio"/>	Reduced redundancy	Noncritical, frequently accessed data with milliseconds access (not recommended as S3 Standard is more cost effective)	General purpose	≥ 3	-	-	-	Per-GB fees apply

23. At the bottom of the page, choose **Upload**.

24. Choose **Close**.

The index.html file appears in the **Objects** list.

pk5cs022 [Info](#)

[Objects](#) | [Metadata - Preview](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Mana](#)

Objects (1) [Info](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a l
permissions. [Learn more](#)

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	index.html	html	January 23, 2020 (UTC+00:00)

Task 4. Test your website

26. Go back to your S3 Bucket console for your bucket.

pk5cs022 [Info](#)

[Objects](#) | [Metadata - Preview](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Acces](#)

Bucket overview

AWS Region
US East (N. Virginia) us-east-1

Amazon Resource Name (ARN)
[arn:aws:s3:::pk5cs022](#)

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve
versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning
Disabled

Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object v
[Learn more](#)
Disabled

27. Select the **Properties** tab, and scroll down to the **Static website hosting** section (the last one).

Static website hosting [Edit](#)

Use this bucket to host a website or redirect requests. [Learn more](#)

[We recommend using AWS Amplify Hosting for static website hosting](#)
Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about [Amplify Hosting](#) or [View your existing Amplify apps](#)

[Create Amplify app](#)

S3 static website hosting
Disabled

28. Choose **Edit**.

29. Select **Enable**.

30. In the **Index document** text box, enter index.html

Edit static website hosting [Info](#)

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

☐ Disable
☒ Enable

Hosting type

☒ Host a static website
Use the bucket endpoint as the web address. [Learn more](#)

☐ Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

ⓘ For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the [S3 Block Public Access](#)

Index document
Specify the home or default page of the website.

index.html

Error document - optional
This is returned when an error occurs.

error.html

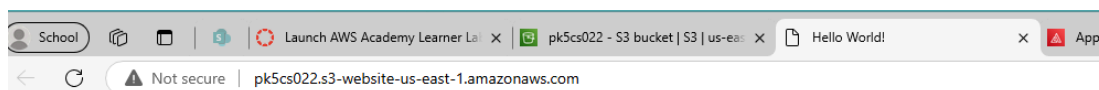
Redirection rules - optional
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)

1		

31. Select **Save changes**.

32. Scroll down to the **Static website hosting** section again, and right click on the **Bucket website endpoint** URL and open it in a new tab.

The **Hello World** web page should display. You have successfully hosted a static website using an S3 bucket.



Welcome to 5CS022 Distributed and Cloud Systems Programming.

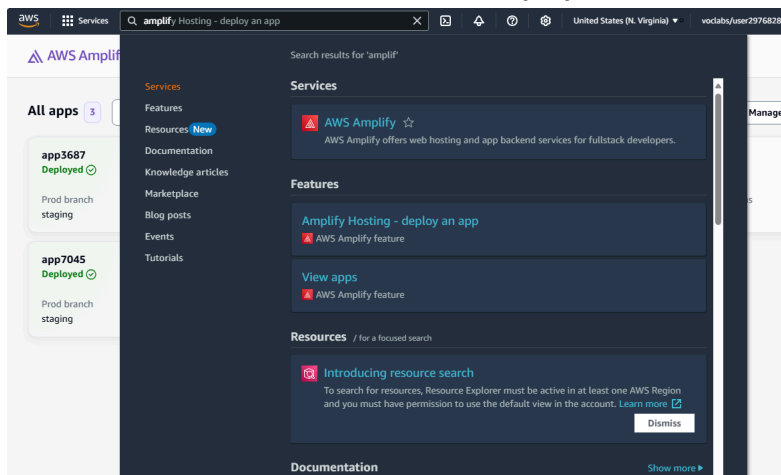
This is a sample web page for AWS S3 Static Web Page Hosting.

Part 2 Using AWS Amplify to Host a Website Stored on S3

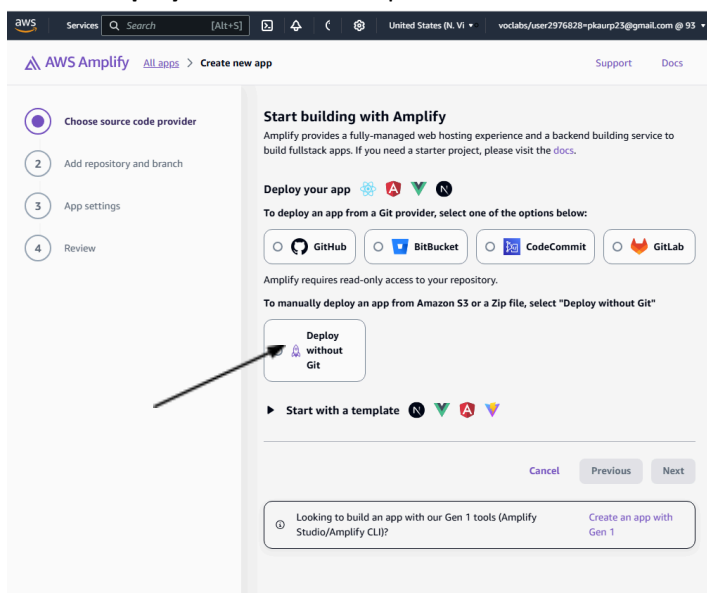
In this part, you will use AWS Amplify to host and manage your website stored in the Amazon S3 bucket. AWS Amplify simplifies deployment, provides secure global hosting, and automatically manages scaling and updates for your website, making it easier to maintain and deliver content efficiently.

Task 1. Create a AWS Amplify app to serve your website

1. Go to the main AWS Console
2. Choose the **Services** menu, search **AWS Amplify**.



3. On the right-hand side, click **Create New App**.
4. Choose **Deploy without Git** and press **next**.



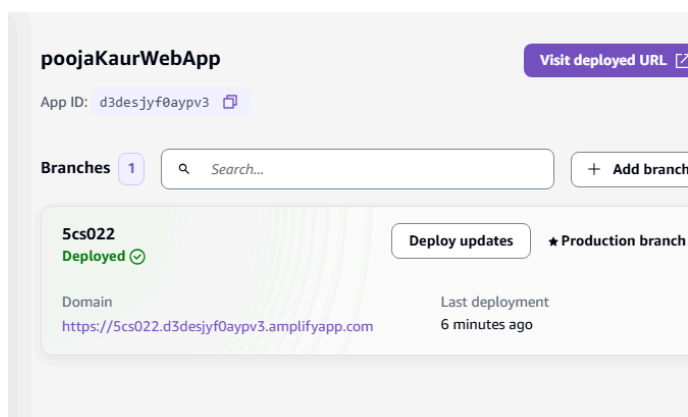
5. In the App Name field, type **NameSurnameWebApp** (e.g., poojaKaurWebApp).
6. In the Branch Name field, type **5cs022**.
7. Select **Amazon S3** as the method.

8. Click **Browse S3**, choose the S3 bucket you created in Part 1
9. In the branch name type **5cs022** and select **Amazon S3** as Method and press **Choose Prefix**.
10. Once your form looks like below click on **Save and Deploy**.

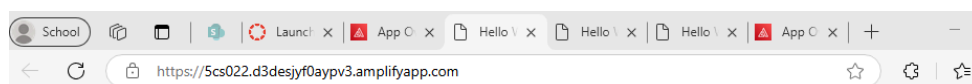
The screenshot shows the 'Create new app' page in the AWS Amplify console. On the left, there are two steps: 'Choose create method' (marked with a green check) and 'Start a manual deployment' (marked with a purple circle). The main area is titled 'Start a manual deployment' and contains the following fields and options:

- App name:** poojaKaurWebApp
- Branch name:** 5cs022
- Method:** Three radio buttons are present: 'Drag and drop' (unselected), 'Amazon S3' (selected), and 'Any URL' (unselected).
- S3 location of objects to host:** A text field containing 's3://pk5cs022/' and a 'Browse S3' button.
- Buttons at the bottom:** 'Cancel', 'Previous', and 'Save and deploy'.

11. Wait until the **Status** says *Deployed*, you can test your app.



12. Once deployed, click on the **Domain** link provided by AWS Amplify, and you should see the index.html page you uploaded to the S3 bucket.

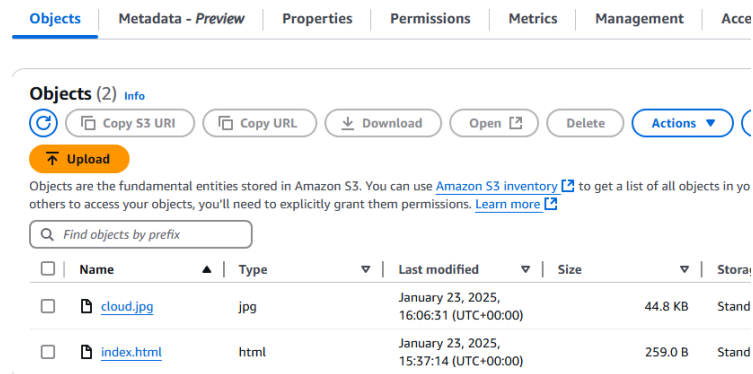


13. Copy the **Domain Name** value for your distribution and save it to a text editor to use in a later step.

Task 2. Create a new HTML page

14. Create a new HTML file to test the distribution.

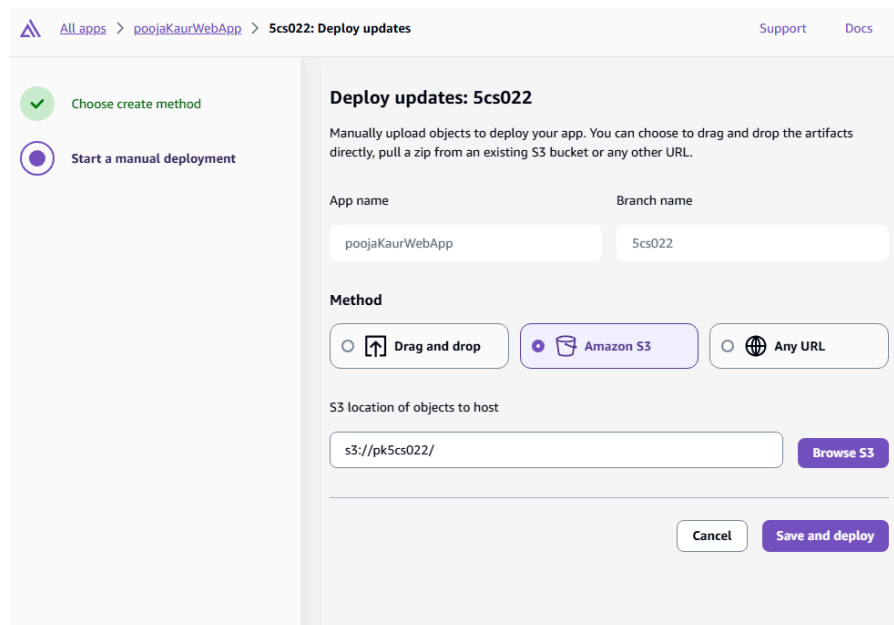
- Find and download an image from the internet.
- Navigate to your S3 bucket and upload the image file to it, making sure to grant public access as you did when uploading the HTML file earlier in this lab (follow Task 3 in part 1).



15. Go back to **AWS Amplify** and click on the name of your app.

16. Press **Deploy Updates** to redeploy changes whenever you make changes to your s3 bucket.

17. This will take you to the same process as before, however this time is to update the domain so just select **Amazon S3** and **Save and Deploy** (leave the S3 location as it was).



18. Wait a few minutes for it to be deployed.

19. Create a new text file using Notepad and copy the following text into it:

<html>

```

<head>My Amplify App</head>

<body>

  <p>My test content goes here.</p>

  <p>

</body>

</html>

```

- Replace **domain-name** with the domain name that you copied earlier for your AWS Amplify.
- Replace **object-name** with the file name of the picture file ([cloud.jpg](#)) that you uploaded to your S3. The edited line of code should look similar to the following:

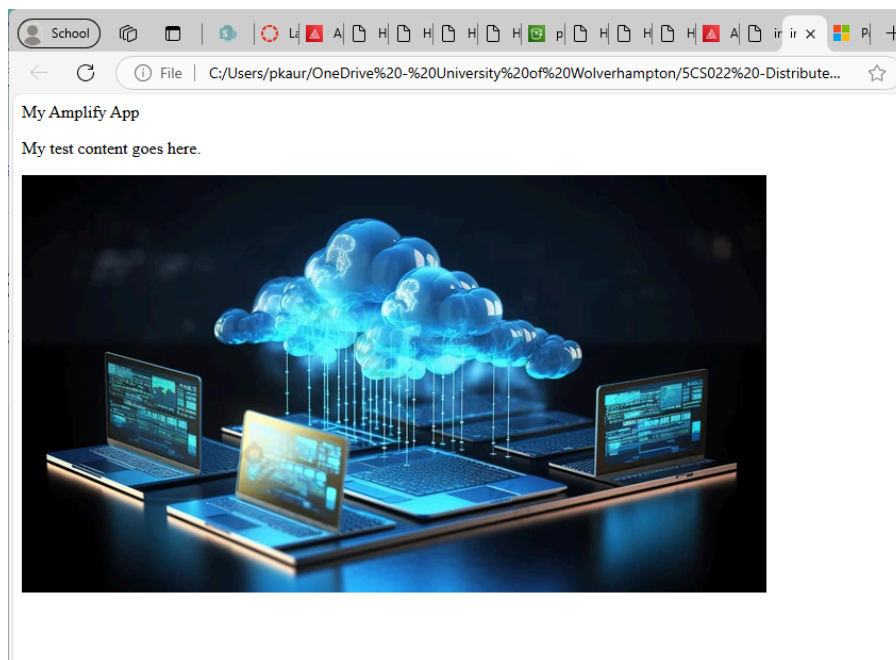
```

<p>

```

- Obviously, replace my example URL with yours.
- Save the text file with an HTML extension.

20. Use a web browser to open the HTML file, mine looks like below.



If the image that you uploaded shows, your deployed Amplify app was successful. If not, try again later or with a different image but make sure the link of the image is the domain you have just created.

Explore these two services further by creating and uploading other HTML pages and images and deploying them to the S3 and AWS Amplify services.

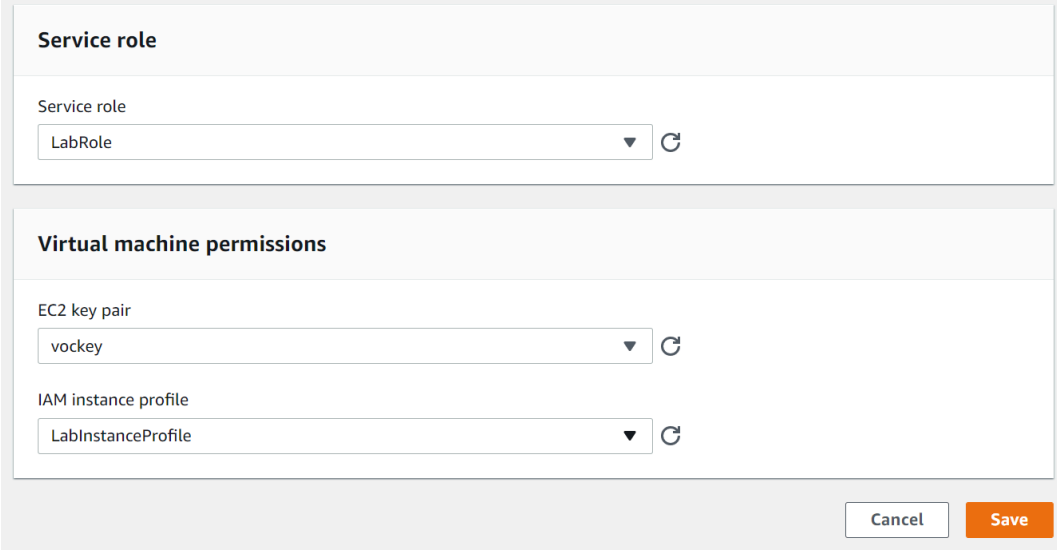
Remember that every time you upload a file on S3 you must re-deploy (deploy updates) your app on Amplify to see the changes on your domain.

Part 3. Creating a web application using AWS Elastic Beanstalk

In this lab, you will deploy and publish a PHP web application using AWS Elastic Beanstalk.

Task 1. Deploy an application using Elastic Beanstalk

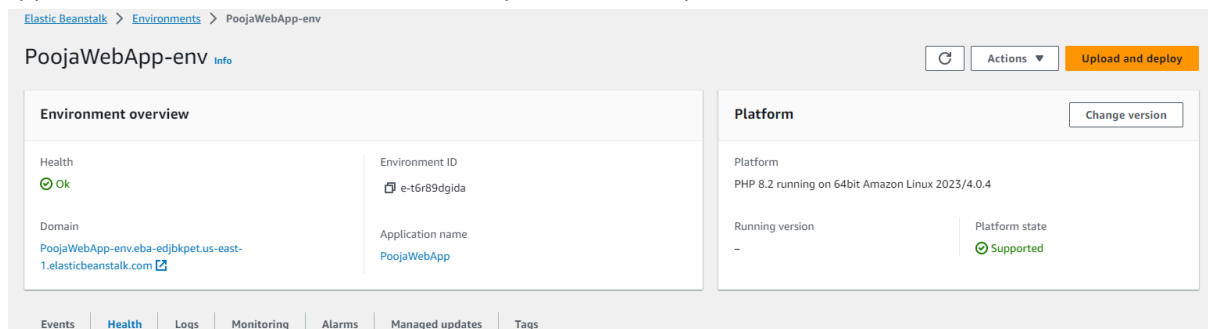
1. Go to the main AWS Console
2. Choose the **Services** menu, locate the **Compute** services, and choose **Elastic Beanstalk**.
3. Choose **Create Application**.
4. For **Application name**, enter a name for your application, younameWebApp (e.g. PoojaWebApp)
5. For **Platform**, select **PHP**.
6. For **Application code**, select **Sample application**.
7. For **Presets** select “**Custom configuration**”
8. Scroll down to “**Security**” and click Edit.
9. Select the following options:



The screenshot shows the 'Security' configuration page in the AWS Elastic Beanstalk console. It contains two main sections: 'Service role' and 'Virtual machine permissions'. The 'Service role' section has a dropdown menu currently showing 'LabRole'. The 'Virtual machine permissions' section contains two sub-sections: 'EC2 key pair' with a dropdown showing 'vockey', and 'IAM instance profile' with a dropdown showing 'LabInstanceProfile'. At the bottom right of the form are 'Cancel' and 'Save' buttons.

10. Click Skip to review and Submit.
11. Click **Create app**.

Wait for the console as Elastic Beanstalk creates and runs the necessary resources to run the application. It takes a few minutes for the process to complete.



Elastic Beanstalk creates an Amazon Simple Storage Service (Amazon S3) storage bucket and a security group, launches an Amazon Elastic Compute Cloud (Amazon EC2) instance and runs the code.

When complete, the screen changes to show the newly created environment. It is ready for you to upload a PHP application.

10. Download the sample web application from the following link:

<https://drive.google.com/file/d/1pZuuWaDrm53OJaktxVUZ-bLmEfXA1vk4/view?usp=sharing>

You should now have a file called *php.zip*.

11. Return to the Elastic Beanstalk console tab.
12. Choose **Upload and deploy**.
13. Choose **Choose file**, navigate to and select the *php.zip* file that you downloaded, and choose **Open**.
14. Choose **Deploy**.

Upload and deploy

To deploy a previous version, go to the [Application Versions](#) page.

Upload application

Choose file

File name : **php.zip**

Version label

Sample Application-1

Deployment Preferences

The application version will be deployed using the **All at once** policy.

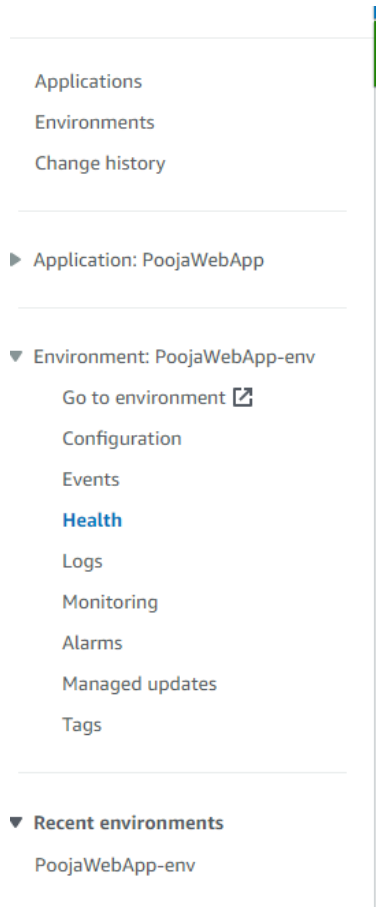
Current number of instances: **1**

Cancel

Deploy

The application deploys to the environment using all of the cloud resources Elastic Beanstalk provisioned.

15. To see your PHP website, in the left navigation pane, choose **Go to environment**.



The web application opens in a new tab.

Congratulations!

Your AWS Elastic Beanstalk *PHP* application is now running on your own dedicated environment in the AWS Cloud

You are running PHP version 8.0.13

This environment is launched with Elastic Beanstalk PHP Platform

What's Next?

- [AWS Elastic Beanstalk overview](#)
- [Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git](#)
- [Using Amazon RDS with PHP](#)
- [Customizing the Software on EC2 Instances](#)
- [Customizing Environment Resources](#)

AWS SDK for PHP

- [AWS SDK for PHP home](#)
- [PHP developer center](#)
- [AWS SDK for PHP on GitHub](#)

This is the end of this workshop. Now go back to the AWS Academy website and click on “End lab” to stop AWS from consuming any further credits.