

# Activity - Create a Website on S3

## Activity overview

In this activity, you will practice using AWS Command Line Interface (AWS CLI) commands to:

- Create an Amazon Simple Storage Service (Amazon S3) bucket.
- Create a new AWS Identity and Access Management (IAM) user that has full access to the Amazon S3 service.
- Upload files to Amazon S3 to host a simple website for the Café & Bakery.
- Create a batch file that can be used to update the static website when you change any of the website files locally.

Once you have completed the activity, clients will be able to access the website you have deployed to Amazon S3 as shown in this diagram.



To ensure that all students have the same setup, you will practice using the AWS CLI from an Amazon Elastic Compute Cloud (Amazon EC2) instance. In this activity, you will be provided with an Amazon Linux instance that has the AWS CLI pre-installed. You will establish a Secure Shell (SSH) connection to the instance.

## Activity objectives

After completing this activity, you will be able to:

- **Run** AWS CLI commands that use IAM and Amazon S3 services.
- **Deploy** a static website to an S3 bucket.
- **Create** a script that uses the AWS CLI to copy files in a local directory to Amazon S3.

## Business case relevance

## A business request from Café—Launch a website



Sofia mentioned to Nikhil that she would like the café to have a website that will showcase the café visually through images, and provide customers with business details, such as the location of the store, the business hours, and telephone number.

Nikhil is pleased to be tasked with creating the first website for the Café.

During this activity, you will take on the role of Nikhil and work on producing the results that everyone back at the café hopes you can deliver. Perhaps you can even exceed their expectations!

## Activity Steps

---

**Duration:** This activity requires approximately **45 minutes** to complete.

## Accessing the AWS Management Console

---

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

A Start Lab panel opens displaying the lab status.

2. Wait until you see the message "**Lab status: ready**", then click the **X** to close the Start Lab panel.

3. At the top of these instructions, click AWS

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

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

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

# Task 1: Connect to an Amazon Linux EC2 instance by using SSH

---

In this task, you will connect to an existing Amazon Linux EC2 instance that already has the AWS CLI installed on it.

Windows users should follow Task 1.1. Both macOS and Linux users should follow Task 1.2.


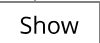

[macOS/Linux users—click here for login instructions](#)

## Task 1.1: Windows SSH


### Windows Users: Using SSH to Connect

💬 These instructions are for Windows users only.

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

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

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

9. Configure your PuTTY session:
  - Click **Session**
  - **Host Name (or IP address):** Copy and paste the **IPv4 Public IP address** for the instance. To find it, return to the EC2 Console and click on **Instances**. Check the box next to the instance you want to connect to and in the *Description* tab copy the **IPv4 Public IP** value.
  - Back in PuTTY, in the **Connection** list, expand  **SSH**
  - Click **Auth** (don't expand it)
  - Click **Browse**
  - Browse to and select the lab#.ppk file that you downloaded
  - Click **Open** to select it

- Click **Open**
10. Click **Yes**, to trust the host and connect to it.
  11. When prompted **login as**, enter: `ec2-user`  
This will connect you to the EC2 instance.
  12. [Windows Users: Click here to skip ahead to the next task.](#)

## Task 1.2: macOS/Linux SSH

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

13. Read through the three bullet points in this step before you start to complete the actions, because you will not be able to see these instructions when the Details panel is open.
  - Click on the `Details` drop down menu above these instructions you are currently reading, and then click `Show`. A Credentials window will open.
  - Click on the **Download PEM** button and save the **labsuser.pem** file.
  - Then exit the Details panel by clicking on the **X**.
14. Open a terminal window, and change directory `cd` to the directory where the labsuser.pem file was downloaded.

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

```
cd ~/Downloads
```

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

```
chmod 400 labsuser.pem
```

16. Return to the AWS Management Console, and in the EC2 service, click on **Instances**. Check the box next to the instance you want to connect to.
17. In the *Description* tab, copy the **IPv4 Public IP** value.
18. Return to the terminal window and run this command (replace **<public-ip>** with the actual public IP address you copied):

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

19. Type `yes` when prompted to allow a first connection to this remote SSH server.  
Because you are using a key pair for authentication, you will not be prompted for a password.

## Task 2: Configure the AWS CLI

**NOTE:** Unlike some other Linux distributions that are available through Amazon Web Services

(AWS), Amazon Linux instances already have the AWS CLI pre-installed on them.

20. Update the AWS CLI software with the credentials.

```
aws configure
```

21. At the prompts, enter the following information:

- **AWS Access Key ID:** Click on the `Details` drop down menu above these instructions, and then click `Show`. Copy the **AccessKey** value and paste it into the terminal window.
- **AWS Secret Access Key:** Copy and paste the **SecretKey** value from the same Credentials screen.
- **Default region name:** `us-west-2`
- **Default output format:** `json`

## Task 3: Create an S3 bucket by using the AWS CLI

---

22. Open the [AWS CLI documentation for S3api](#).

**Tip:** In this activity, you will sometimes use the `s3api` command, and other times you will use the `s3` command. `s3` commands are built on top of the operations that are found in the `s3api` commands.

23. Use the `aws s3api create-bucket` command to create a bucket in Amazon S3. The bucket must have a unique name, such as the combination of your first initial, last name, and three random numbers. For example, `jsmith256` would be a good bucket name if your name is Jane Smith.

- Specify `--region us-west-2` because you want to host the website in the AWS Region that is closest to the people who are most likely to access it (in the London area).
- You must add `--create-bucket-configuration LocationConstraint=us-west-2` to the end of the command.

If the command is successful, you will get a JavaScript Object Notation (JSON)-formatted response with a `Location` name-value pair, where the value reflects the bucket name.

## Task 4: Create a new IAM user that has full access to Amazon S3

---

24. Using the AWS CLI, create a new IAM user with the name: `awss3user`

*Tip:* Use the AWS CLI documentation as needed to figure out the command you must run. Like the previous command that you ran, you will get a JSON-formatted response if the command is successful.

25. Create a login profile for the new user by using the following command:

```
aws iam create-login-profile --user-name awsS3user --password Training123!
```

26. Copy AWS the account number. To do this:

- In the AWS Management Console you opened earlier in this activity, click on the **voclabs/user...** drop down menu in the upper right of the screen.
- Copy the account number that displays. It will be a 12 digit number with dashes in it.
- Still in the drop down menu, choose **Sign Out**.

27. Log in to the AWS Management Console as the new **awsS3user** user. To do this:

- In the browser tab where you just signed out of the AWS Management Console, click the **Sign in to the Console** button. *Note:* Due to console changes, you may also see **Log back in** as an option.
- You will be taken to a sign in screen.
- Select **IAM user**
- In the text field, enter `awsS3user`
- Click **Next**.
- You should see a new login screen with an **Account ID or alias** field. The field should already display the account number. If it does not display the account number, paste in the account number that you copied a moment ago (but remove the dashes after you paste it in).
- Click **Next**
- For **IAM user name**, enter `awsS3user`
- For **Password**, enter `Training123!`
- Click **Sign In**

28. In the console, go to the **Services** menu and choose **S3**.

The Amazon S3 service page will display with an error message that states *Access denied*. This behavior is expected because the new user has not been granted any rights. Leave this page open because you will return to it again soon.

29. Back in the terminal window, find the AWS managed policy that grants full access to Amazon S3. Run this command to find it:

```
aws iam list-policies --query "Policies[?contains(PolicyName,'S3')]"
```

The result set displays policies that have a PolicyName attribute containing the term `S3` somewhere in it. Locate the one that grants full access to Amazon S3. You will use it in the next step.

30. Grant the **awsS3user** full access to the S3 bucket by using the following command:

```
aws iam attach-user-policy --policy-arn  
arn:aws:iam::aws:policy/<policyYouFound> --user-name awsS3user
```

Replace **<policyYouFound>** in the command above with the PolicyName value of the appropriate policy that you found by using the command in the previous step.

31. Return to the AWS Management Console and refresh the browser tab.

The *Access denied* error should go away, and you should now see the bucket that you created by using the AWS CLI earlier in this activity.

## Task 5: Extract the files that you need for this activity

---

32. Back in the SSH terminal, extract the files that you need for this activity by running the following commands:

```
cd ~/sysops-activity-files
tar xvzf static-website-v2.tar.gz
cd static-website
```

Run the `ls` command to confirm that the files extracted correctly. You should see an `index.html` file and directories named **css** and **images**.

33. Delete the tar.gz file. Use the Linux `rm` command, but leave the extracted files.

## Task 6: Upload files to Amazon S3 by using the AWS CLI

---

34. Run the following command to prepare the bucket that you created earlier to function as a website. This process will ensure that `index.html` will be understood to be the index document.

**NOTE:** Be sure to replace `<my-bucket>` in the command below with your actual bucket name.

```
aws s3 website s3://<my-bucket>/ --index-document index.html
```

35. Upload the files to the bucket Replace `<my-bucket>` with the actual bucket name.

```
aws s3 cp /home/ec2-user/sysops-activity-files/static-website/ s3://<my-bucket>/ --recursive --acl public-read
```

Notice that the upload command includes an access control list (ACL) parameter that specifies that the uploaded file should be available to the public to read. It also includes the `recursive` parameter, which indicates that all files in the current directory on your machine should be uploaded.

36. Verify that they were uploaded (again, replace `<my-bucket>` with the actual bucket name):

```
aws s3 ls <my-bucket>
```

37. Back in the **AWS Management Console** browser window, in the S3 service area, click your

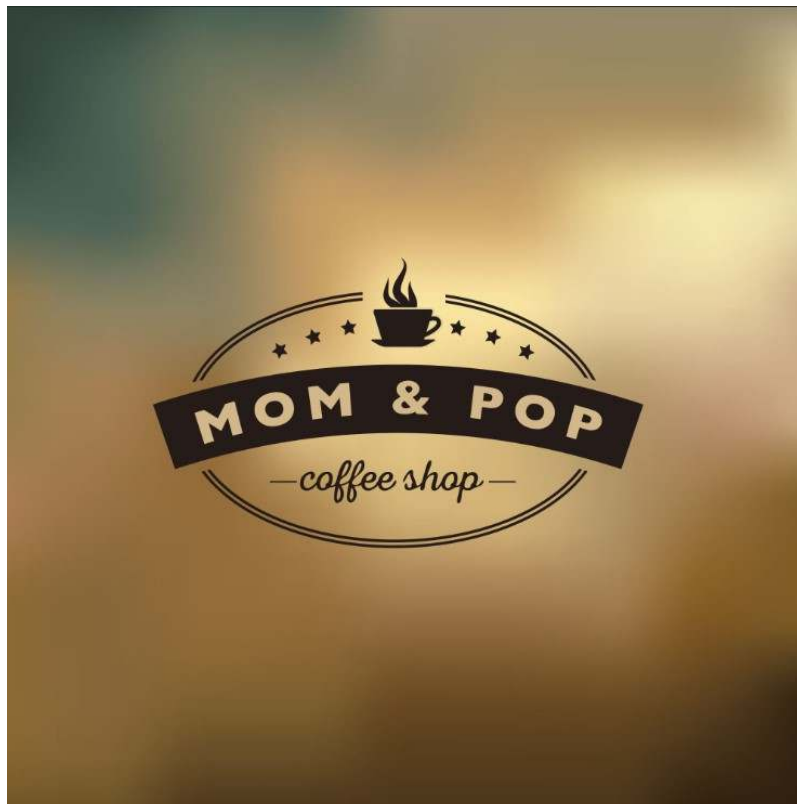
bucket name.

38. Click the **Properties** tab. In the **Static website hosting** area, note that **Bucket hosting** has been enabled, which resulted from running the `aws s3 website` AWS CLI command.

39. Click the **Bucket hosting** link, and then click the **Endpoint** URL that displays.

Congratulations, you have created a static website that is available to the public for viewing!

## A business request from Café—Launch a website



Nikhil showed Sofia the new static website, and she was impressed. Give yourself a pat on the back—nice job!

Martha and Frank were impressed as well. However, it did not take long before they started requesting changes. Martha requested a change to the background color that appears behind the cookies, coffee, and tarts in the second row of photos. Sofia, realizing this might just be the first of many change requests, decides to make updating the website into a batch process by using her programming skills. In this last part of the activity, you take on the role Sofia and work on introducing automation into the process of updating the website.

## Task 7: Create a batch file to make updating the website easily repeatable

---

40. In the terminal window, pull up the history of recent commands by using the following



command:

```
history
```

41. Locate the line where you ran the `aws s3 cp` command. You will put this line in your new batch file soon.
42. To change directories and create an empty file, run the following commands in the SSH terminal session:

```
cd ~  
touch update-website.sh
```

43. Open the empty file in the VI editor.

```
vi update-website.sh
```

44. To enter edit mode in VI, press **A**.
45. Add the standard first line of a bash file, and then add the `s3 cp` line from your history (replace `<my-bucket>` with your actual bucket name):

```
#!/bin/bash  
aws s3 cp /home/ec2-user/sysops-activity-files/static-website/ s3://<my-bucket>/ --recursive --acl public-read
```

46. Write the changes and quit the file by pressing **ESC**, typing `:wq` and then pressing **ENTER**.
47. Make it an executable batch file.

```
chmod +x update-website.sh
```

48. Open the local copy of the `index.html` file in a text editor.

```
vi sysops-activity-files/static-website/index.html
```

49. Modify the file as described:

- Locate the first line that has the HTML code `bgcolor="aquamarine"` and change it to `bgcolor="gainsboro"`
- Locate the line that has the HTML code `bgcolor="orange"` and change it to `bgcolor="cornsilk"`
- Locate the second line that has the HTML code `bgcolor="aquamarine"` and change it to `bgcolor="gainsboro"`
- Save the changes.

50. Run your batch file to update the website.

```
./update-website.sh
```

**NOTE:** The command line output should show that the files were copied to Amazon S3.

51. Return to the browser and refresh the Café page to see the changes to the website.

Congratulations, you just made your first revision to the website!

You now also now have a tool (the script that you created) that makes it easy to push changes from your website source files to Amazon S3.

## Optional challenge

---

Did you notice that your batch file uploads *every file* to Amazon S3 every time you run it, even when most of the files have no changes in them?

- Take a look at the [AWS CLI reference documentation for sync](#), and see if you can replace the `aws s3 cp` line in your file with an `aws s3 sync` line.
- To test it, make a small change to the index.html file. For example, modify one of the colors again) and save the change. Then, run your updated batch file. Did the script become more efficient?

## Lab Complete

---

Congratulations! You have completed the activity.

52. Click  at the top of this page and then click **Yes** to confirm that you want to end the activity.

A panel will appear, indicating that "DELETE has been initiated... You may close this message box now."

53. Click the **X** in the top right corner to close the panel.