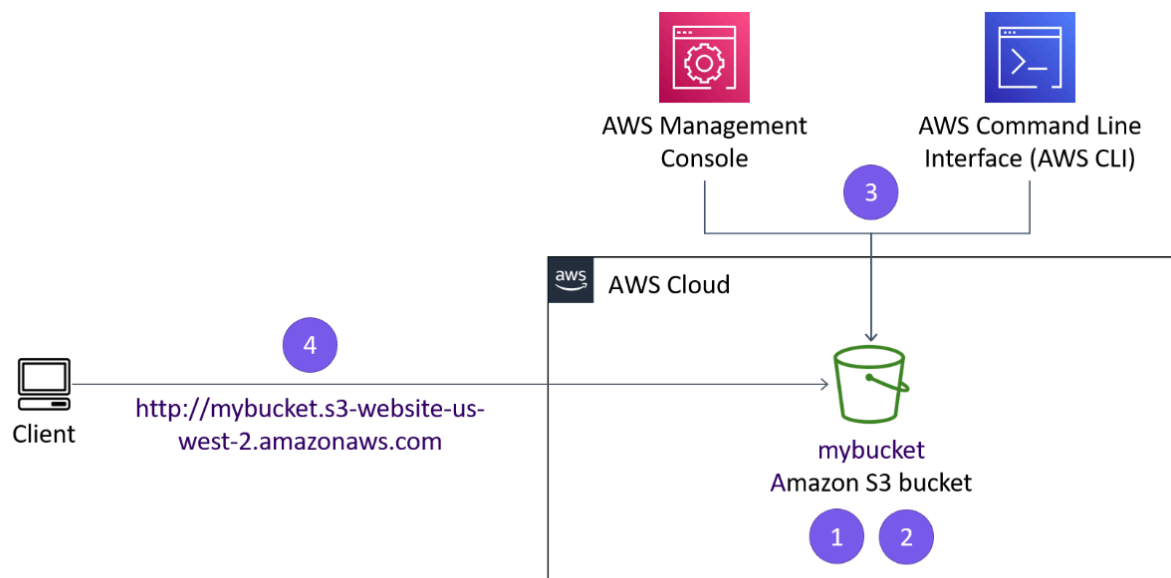


## Creating a Website on S3

### Lab overview

In this lab, you practice using AWS Command Line Interface (AWS CLI) commands from an Amazon Elastic Compute Cloud (Amazon EC2) instance 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.



Clients will be able to access the website you have deployed to Amazon S3. The website URL is similar to this example: `http://s3-website-us-west-2.amazonaws.com`. You can create and access the bucket through the AWS Management Console or the AWS CLI.

## Objectives

---

After completing this lab, you should 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.

## Duration

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

## Accessing the AWS Management Console

---

1. At the top of these instructions, choose **Start Lab** to launch the lab.
2. Wait until the message "Lab status: ready" appears, and then choose **X** to close the **Start Lab** panel.
3. Next to **Start Lab**, choose **AWS**, which opens the AWS Management Console in a new browser tab. The system automatically signs you in.

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

4. Arrange the AWS Management Console so that it appears alongside these instructions.

**Important:** Do not change the lab Region unless specifically instructed to do so.

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

---

In this task, you connect to your Amazon EC2 Instance using AWS Systems Manager Session Manager.

5. Choose the **Details** button at the top, then choose **Show**.
6. Copy the **InstanceSessionUrl** value from the list and then paste it into a new web browser tab.

A console connection is made to the instance inside your web browser window using ssm-user, a prompt is displayed.

Run the following commands to change the user and home directory:

```
sudo su -l ec2-user
pwd
```

**Note:** This is the SSH terminal where you run commands as instructed throughout the lab.

## Task 2: Configure the AWS CLI

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.

7. In the SSH session terminal window, run the configure command to update the AWS CLI software with credentials.

```
aws configure
```

8. At the prompt, configure the following:

- **AWS Access Key ID:** Copy and paste the value for **AccessKey** from pane in the left, into the terminal window.
- **AWS Secret Access Key:** Copy and paste the value for **SecretKey** from pane in the left, into the terminal window.
- **Default region name:** Enter `us-west-2`
- **Default output format:** Enter `json`

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

The `s3api` command creates a new S3 bucket with the AWS credentials in this lab. By default, the S3 bucket is created in the us-east-1 Region.

**Tip:** In this lab, you might use the `s3api` command or the `s3` command. `s3` commands are built on top of the operations that are found in the `s3api` commands.

When you create a new S3 bucket, the bucket must have a unique name, such as the combination of your first initial, last name, and three random numbers. For example, if a user's name is Terry Whitlock, a bucket name could be `twhitlock256`.

9. To create a bucket in Amazon S3, you use the `aws s3api create-bucket` command. When you use this command to create an S3 bucket, you also include the following:

- Specify `--region us-west-2`
- Add `--create-bucket-configuration LocationConstraint=us-west-2` to the end of the command.

The following is an example of the command to create a new S3 bucket. You can use `twhitlock256` as your bucket name, or you can replace `<twhitlock256>` with a bucket name that you prefer to use for this lab.

```
aws s3api create-bucket --bucket <twhitlock256> --region us-west-2 --create-bucket-configuration
LocationConstraint=us-west-2
```

If the command is successful, you will get a JSON-formatted response with a **Location** name-value pair, where the value reflects the bucket name. The following is an example:

```
{
  "Location": "http://twhitlock256.s3.amazonaws.com/"
}
```

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

The AWS CLI command: `aws iam create-user` creates a new IAM user for your AWS account. The option `--user-name` is used to create the name of the user and must be unique within the account.

10. Using the AWS CLI, create a new IAM user with the command `aws iam create-user` and username `awsS3user`:

```
aws iam create-user --user-name awsS3user
```

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

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

12. Copy the AWS account number:

- In the AWS Management Console, choose the account **VocLabsUser...** drop down menu located at the top right of the screen.
- Copy the 12 digit **Account ID** number.
- In the current drop down menu, choose **Sign Out**.

13. Log in to the AWS Management Console as the new **awsS3user** user:

- In the browser tab where you just signed out of the AWS Management Console, choose **Log back in** or **Sign in to the Console**.
- In the sign-in screen, choose the radio button **IAM user**.
- In the text field, paste or enter the account ID with no dashes.
- Choose **Next**.
- A new login screen with **Sign in as IAM user** field will show. The account ID will be filled in from the previous screen.
- For **IAM user name**, enter `awsS3user`
- For **Password**, enter `Training123!`
- Choose **Sign In**

14. On the AWS Management Console, in the **Search** box, enter `S3` and choose **S3**. This option takes you to the Amazon S3 console page.

**Note:** The bucket that you created might not be visible. Refresh the Amazon S3 console page to see if it appears. The **awsS3user** user does not have Amazon S3 access to the bucket that you created, so you might see an error for **Access** to this bucket.

15. In the terminal window, to find the AWS managed policy that grants full access to Amazon S3, run the following command:

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

The result displays policies that have a **PolicyName** attribute containing the term S3. Locate the policy that grants full access to Amazon S3. You use this policy in the next step.

16. To grant the **awsS3user** user full access to the S3 bucket, replace `<policyYouFound>` in following command with the appropriate **PolicyName** from the results, and run the adjusted command:

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

17. Return to the AWS Management Console, and refresh the browser tab.

## Task 5: Adjust S3 bucket permissions

18. On the **AWS Management Console**, on the Amazon S3 console, choose your bucket name.
19. Go to permissions, under **Block public access (bucket settings)**, choose **Edit**
20. DeSelect/UnSelect **Block all public access**
21. Choose **Save changes** (*confirm* on the prompt)
22. On to permissions tab, under **Object Ownership**, choose **Edit**
23. Choose **ACLs enabled**
24. Choose **I acknowledge that ACLs will be restored.**
25. Choose **Save changes**

## Task 6: Extract the files that you need for this lab

A file containing the `static-website` contents for the Amazon S3 bucket will need to be extracted in the following step.

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

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

27. To confirm that the files were extracted correctly, run the `ls` command.

You should see a file named `index.html` and directories named `css` and `images`.

## Task 7: Upload files to Amazon S3 by using the AWS CLI

Once the files are extracted, you upload the contents of the file to Amazon S3. These files include what you explored when you ran the `ls` command.

28. So that the bucket can function as a website, replace `<my-bucket>` in the following command with your bucket name, and run the adjusted command.

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

This process helps ensure that the `index.html` file will be known as the index document.

29. To upload the files to the bucket, replace `<my-bucket>` in the following command with your bucket name, and run the adjusted command:

```
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. This parameter specifies that the uploaded files have public read access. It also includes the recursive parameter, which indicates that all files in the current directory on your machine should be uploaded.

30. To verify that the files were uploaded, replace `<my-bucket>` in the following command with your bucket name, and run the adjusted command:

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

31. On the **AWS Management Console**, on the Amazon S3 console, choose your bucket name.
32. Choose the **Properties** tab. At the bottom of the this tab, note that **Static website hosting** is **Enabled**. Running the `aws s3 website` AWS CLI command turns on the static website hosting for an Amazon S3 bucket. This option is usually turned off by default.
33. To open the URL on a new page, choose the **Bucket website endpoint** URL that displays.
- Congratulations, you have created a static website that is available to the public for viewing!

## Task 8: Create a batch file to make updating the website repeatable

To create a repeatable deployment, you create a batch file by using the VI editor.

34. In the terminal window, to pull up the history of recent commands, run the following command:

```
history
```

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

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

37. To open the empty file in the VI editor, run the following command.

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

38. To enter edit mode in the VI editor, press `i`
39. Next, you add the standard first line of a bash file and then add the `s3 cp` line from your history. To do so, replace `<my-bucket>` in the following command with your bucket name, and copy and paste the adjusted command into your file:

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

40. To write the changes and quit the file, press Esc, enter `:wq` and then press Enter.

41. To make the file an executable batch file, run the following command:

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

42. To open the local copy of the index.html file in a text editor, run the following command:

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

43. To enter edit mode in the VI editor, press `i` and modify the file as follows:

- Locate the first line that has the HTML code `bgcolor="aquamarine"` and change this code to `bgcolor="gainsboro"`
- Locate the line that has the HTML code `bgcolor="orange"` and change this code to `bgcolor="cornsilk"`
- Locate the second line that has the HTML code `bgcolor="aquamarine"` and change this code to `bgcolor="gainsboro"`
- To write the changes and quit the file, press Esc, enter `:wq` and then press Enter.

44. To update the website, run your batch file.

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

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

45. To see the changes to the website, return to the browser and refresh the Café and Bakery page.

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

## 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 to them?

- Take a look at the following document: [AWS CLI reference documentation for sync](#).
- Make a small noticeable change to the index.html file. For example, modify one of the colors, and save the change.
- Run the updated batch file.
- To help make your script more efficient, you replace the `aws s3 cp` command that you've been using with the `aws s3 sync` command from this document. The following is an example of the `aws s3 sync` command that you run in the SSH terminal window. In this command, replace `<my-bucket>` with your bucket name.

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

- Refresh the Café and Bakery site to see your changes.
- How was the `aws s3 sync` command more efficient than the `aws s3 cp` command? Did the `aws s3 sync` command update just the index.html file or upload all the files like the `aws s3 sync` command?

## Conclusion

---

Congratulations! You now have successfully done the following:

- Ran AWS CLI commands that use IAM and Amazon S3 services
- Deployed a static website to an S3 bucket
- Created a script that uses the AWS CLI to copy files in a local directory to Amazon S3

## Lab complete

---

46. At the top of this page, choose **End Lab** and then choose **Yes** to confirm that you want to end the lab.

A panel appears indicating that "You may close this message box now. Lab resources are terminating."

47. To close the **End Lab** panel, choose the **X** in the upper-right corner.