**Working with Amazon S3 using the AWS SDK for Python (Boto3)**

* To Setup Cloud9 IDE – Ref doc : Setting up Cloud9 IDE

### Connect to the AWS Cloud9 IDE.

* + From the **Services** menu, search for and select **Cloud9**.

You should see an existing IDE that's named **Cloud9 Instance**.

* + Choose **Open IDE**.

The AWS Cloud9 IDE loads in a new browser tab.

Upgrade the version of python and Install the AWS SDK for Python.

In the AWS Cloud9 bash terminal (located at the bottom of the IDE), run the following commands:

sudo yum -y remove python36

sudo yum -y install python38

sudo update-alternatives --set python /usr/bin/python3.8

sudo pip install boto3

* Download and extract the files.
  + In the same terminal, run the following command:

**wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCDEV-2/lab-02-s3/code.zip -P /home/ec2-user/environment**

* + The code.zip file is downloaded to the AWS Cloud9 instance. The file is listed in the left navigation pane.
  + Run the following commands to extract the file:

**unzip code.zip**

You will use the files that you downloaded and extracted later.

* Upgrade the AWS Cloud9 environment to AWS CLI version 2.
  + To upgrade to AWS CLI version 2, run the following two commands:

**curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"**

**unzip awscliv2.zip && sudo ./aws/install**

* + To confirm that AWS CLI version 2 is now the installed version, run the aws --version command again.

### Creating an S3 bucket by using the AWS CLI

* create an S3 bucket to host your website. You will complete this task by using the AWS CLI.
* In the AWS Cloud9 Bash terminal, run the following command, replacing <*bucket-name*> with your bucket name.

For the *bucket name*, use the following items, separated by dashes (-):

* Your initials in lowercase
* Today's date in the format *YYYY-MM-DD*
* The word *s3site*

For example, *sm-2022-08-26-s3site*.

**aws s3api create-bucket --bucket <bucket-name> --region us-east-1**

The terminal should confirm that the bucket was created by returning output similar to this example:

**{**

**"Location": "/sm-2022-08-26-s3site"**

**}**

Copy the bucket name a text file in the AWS Cloud9 IDE or on your computer so that you can easily copy it during later steps. The bucket name does not include the leading slash (/) that was returned in the location name-value pair.

### Setting a bucket policy on the bucket by using the SDK for Python

* Create the policy document.
  + In the **AWS Cloud9** IDE, in the navigation pane, choose the **Cloud9 Instance** directory.
  + Choose **File** > **New File** and then choose **File** > **Save**.
  + Name the empty file website\_security\_policy.json and choose **Save**.
  + In the website\_security\_policy.json file, paste the following code:

{

      "Version": "2008-10-17",

      "Statement": [

          {

              "Effect": "Allow",

              "Principal": "\*",

              "Action": "s3:GetObject",

              "Resource": [

                  "arn:aws:s3:::<bucket-name>/\*",

                  "arn:aws:s3:::<bucket-name>"

              ],

              "Condition": {

                  "IpAddress": {

                      "aws:SourceIp": [

                          "<ip-address>/32"

                      ]

                  }

              }

          },

          {

              "Sid": "DenyOneObjectIfRequestNotSigned",

              "Effect": "Deny",

              "Principal": "\*",

              "Action": "s3:GetObject",

              "Resource": "arn:aws:s3:::<bucket-name>/report.html",

              "Condition": {

                  "StringNotEquals": {

                      "s3:authtype": "REST-QUERY-STRING"

                  }

              }

          }

      ]

}

* In the policy document, replace all three <bucket-name> entries with your actual bucket name.
* Finally, replace <ip-address> with the IP address that is being used to connect your computer to the internet. You can find your IP address by visiting [whatismyip.com](https://www.whatismyip.com/)

**Save** the file.

* **Analysis**: This policy document consists of two statements. The first statement allows GetObject requests from your IP address. The second statement denies access to an object in the bucket that's named report.html, unless a specific condition is met. The report.html file doesn't exist in the bucket yet.
* Apply the bucket policy to your bucket by using the SDK for Python.
* In the navigation pane, expand the **python\_3 directory** and open the **permissions.py** file (which contains starter code).
* Edit the file by replacing <bucket-name> with the name of your bucket.
* Save the changes.
* Finally, in the terminal, navigate to the **python\_3** directory and run the following code:

cd python\_3

python3 permissions.py

If the command completed successfully, you should see the message DONE in the terminal output.

Return to the browser tab where the Amazon S3 console is open, and observe that the bucket policy has been applied to the bucket.

### Uploading objects to the bucket to create the website

* Run the code in the terminal. You should still be in the python\_3 directory.
* Be sure to replace <bucket-name> below with your actual bucket name

**aws s3 cp ../resources/website s3://<bucket-name>/ --recursive --cache-control "max-age=0"**

### Testing access to the website

* Load the website in a browser tab.
  + Return to the browser tab with the Amazon S3 console.
  + Choose your bucket name, and then choose **Objects**.

If the files you just uploaded do not display, choose the refresh icon to view them.

* + Choose the **index.html** file.
  + Copy the **Object URL**. It will be in the following format. https://<bucket-name>.s3.amazonaws.com/index.html

**Note**: In this scenario, the S3 bucket you created is intentionally not configured for static website hosting (a feature available in the bucket properties). Instead, you will access the website using the Object URL of the index.html file.

* + Verify that your website displays by pasting that full URI into your browser. Ensuring you are on the same network (as it will block anything other than the IPv4 you specified earlier)

Try to access the same URL from a location outside of your network.

* Ways that you can test outside access:
  + If you have mobile phone with a cellular network connection, try loading the same URL in a browser on the mobile phone.
  + Another way to test is by running the following command in the AWS Cloud9 Bash terminal (where <*bucket-name*> is the actual bucket name):

**curl https://<bucket-name>.s3.amazonaws.com/index.html**

**Analysis**: Regardless whether you load the page from another location or use the curl command from the Cloud9 instance, an attempt is made to retrieve the page that you specified. It should return an AccessDenied error because your mobile device or the AWS Cloud9 instance (whichever one you chose to use) connects to the internet by using a different IP address than your computer.

The essential point is that you should only be able to access the website if the device uses the IPv4 address that's specified in the S3 bucket policy. In the café story, this IPv4 address is the café's network IP address.

* Test that the website is loading.
  + Back in the browser tab where the website loads correctly, choose **Login** on the top right of the header.

An alert will display saying No API to call This behavior is also expected at this point in the course.