**13.** **AWS** **─** **Amazon S3** Amazon Web Services

**Amazon S3** (Simple Storage Service) is a scalable, high-speed, low-cost web-basedservice designed for online backup and archiving of data and application programs. It allows to upload, store, and download any type of files up to 5 GB in size. This service allows the subscribers to access the same systems that Amazon uses to run its own web sites. The subscriber has control over the accessibility of data, i.e. privately/publicly accessible.

**How to Configure S3?**



Following are the steps to configure a S3 account.

**Step 1**: Open the Amazon S3 console using this link:

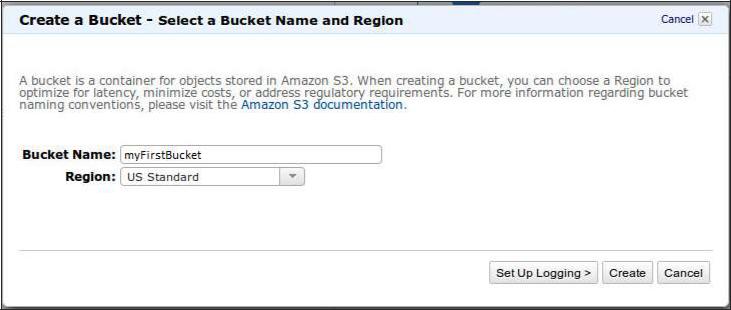
<https://console.aws.amazon.com/s3/>

**Step 2**: Create a Bucket using the following steps.

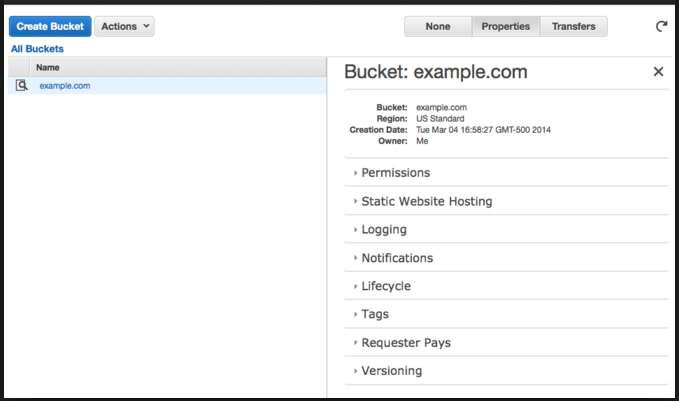
* A prompt window will open. Click the Create Bucket button at the bottom of the page.



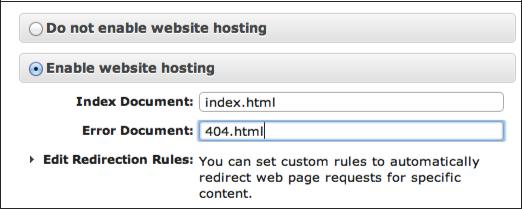
* Create a Bucket dialog box will open. Fill the required details and click the Create button.



The bucket is created successfully in Amazon S3. The console displays the list of buckets and its properties.

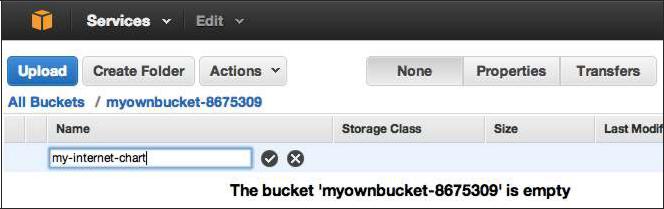


* Select the Static Website Hosting option. Click the radio button Enable website hosting and fill the required details.



**Step 3**: Add an Object to a bucket using the following steps.

* Open the Amazon S3 console using the following link: <https://console.aws.amazon.com/s3/>
* Click the Upload button.

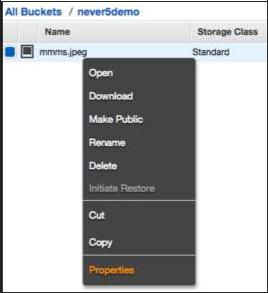


* Click the Add files option. Select those files which are to be uploaded from the system and then click the Open button.



* Click the start upload button. The files will get uploaded into the bucket.

**open/download an object:** In the Amazon S3 console, in the Objects & Folders list,right-click on the object to be opened/downloaded. Then, select the required object.



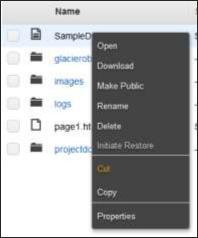
**How to Move S3 Objects?**



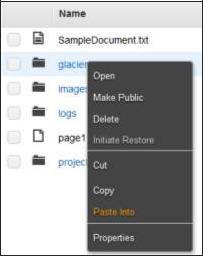
Following are the steps to move S3 objects.

**Step 1**: Open Amazon S3 console.

**Step 2**: Select the files & folders option in the panel. Right-click on the object that is tobe moved and click the Cut option.



**Step 3**: Open the location where we want this object. Right-click on the folder/bucketwhere the object is to be moved and click the Paste into option.



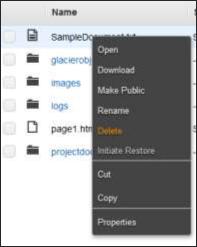
**How to Delete an Object?**



**Step 1**: Open Amazon S3.

**Step 2**: Select the files & folders option in the panel. Right-click on the object that is tobe deleted. Select the delete option.

**Step 3**: A pop-up window will open for confirmation. Click Ok.

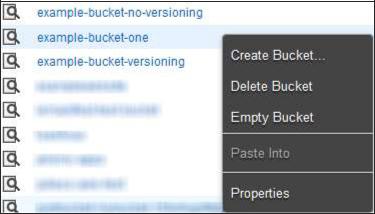


**How to Empty a Bucket?**

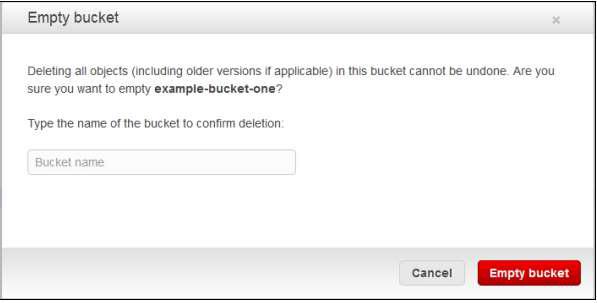


**Step 1**: Open Amazon S3 console.

**Step 2**: Right-click on the bucket that is to be emptied and click the empty bucket option.



**Step 3**: A confirmation message will appear on the pop-up window. Read it carefully andclick the **Empty bucket** button to confirm.



**Amazon S3 Features**



* **Low cost and Easy to Use**: Using Amazon S3, the user can store a large amountof data at very low charges.
* **Secure**: Amazon S3 supports data transfer over SSL and the data gets encryptedautomatically once it is uploaded. The user has complete control over their data by configuring bucket policies using AWS IAM.
* **Scalable**: Using Amazon S3, there need not be any worry about storage concerns.We can store as much data as we have and access it anytime.
* **Higher performance**: Amazon S3 is integrated with Amazon CloudFront, thatdistributes content to the end users with low latency and provides high data transfer speeds without any minimum usage commitments.
* **Integrated with AWS services**: Amazon S3 integrated with AWS servicesinclude Amazon CloudFront, Amazon CLoudWatch, Amazon Kinesis, Amazon RDS, Amazon Route 53, Amazon VPC, AWS Lambda, Amazon EBS, Amazon Dynamo DB, etc.

***AMAZON S3 using java SDK***

***Amazon SDK in NetBeans***

**Step 1:**

Download the jar file from AWS and extract the same and go to Step 3.

***Amazon SDK into Eclipse***

**Step 1:**

Download the Eclipse from <http://vidya.amritanet.edu/Softwares/OpenSource/Eclipse/>

**Step 2: Add Amazon SDK into Eclipse.**

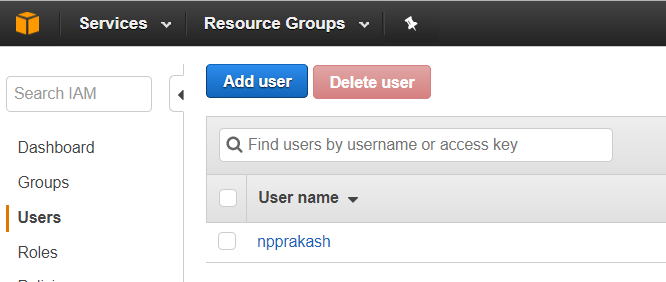
**2.1 To install the AWS Toolkit for Eclipse**

1. Within Eclipse, click **Help** and then click **Install New Software**.
2. In the **Work with** box, type *https://aws.amazon.com/eclipse* and then press Enter.
3. Choose the components of the AWS Toolkit for Eclipse that you want to install. Click **Select All** to install all components at once.
4. Once you have made your selections, click **Next** (or **Finish**) to complete installation.

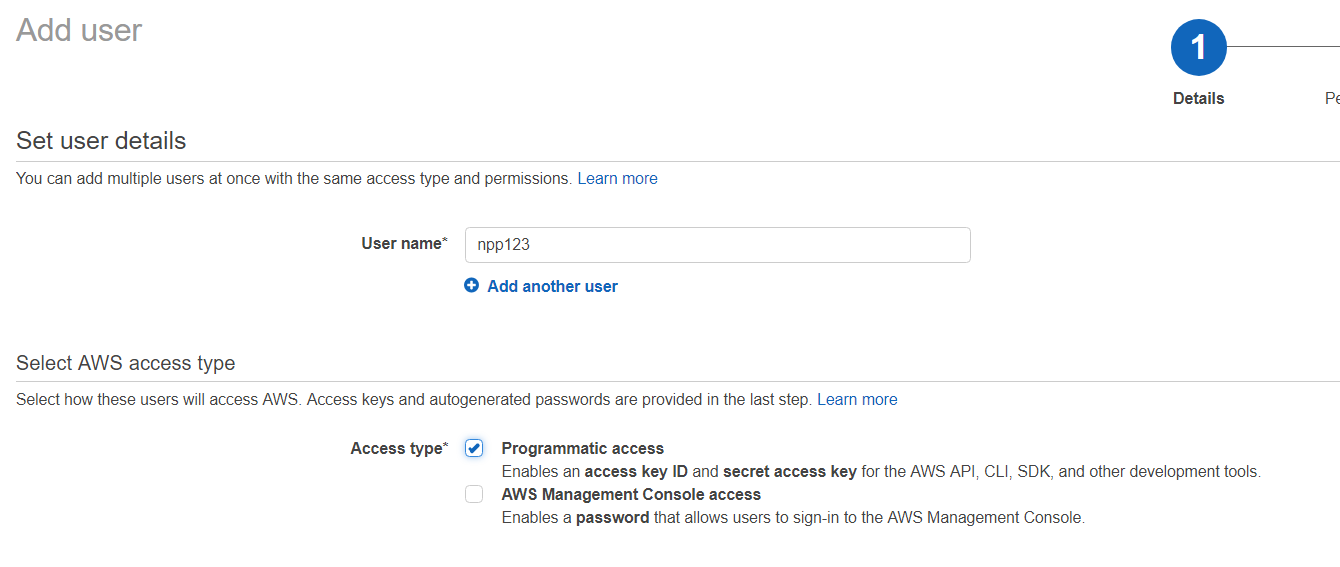
Once you have set up the AWS Toolkit for Eclipse you should configure your AWS Credentials.

**2.2 Set up AWS Credentials**

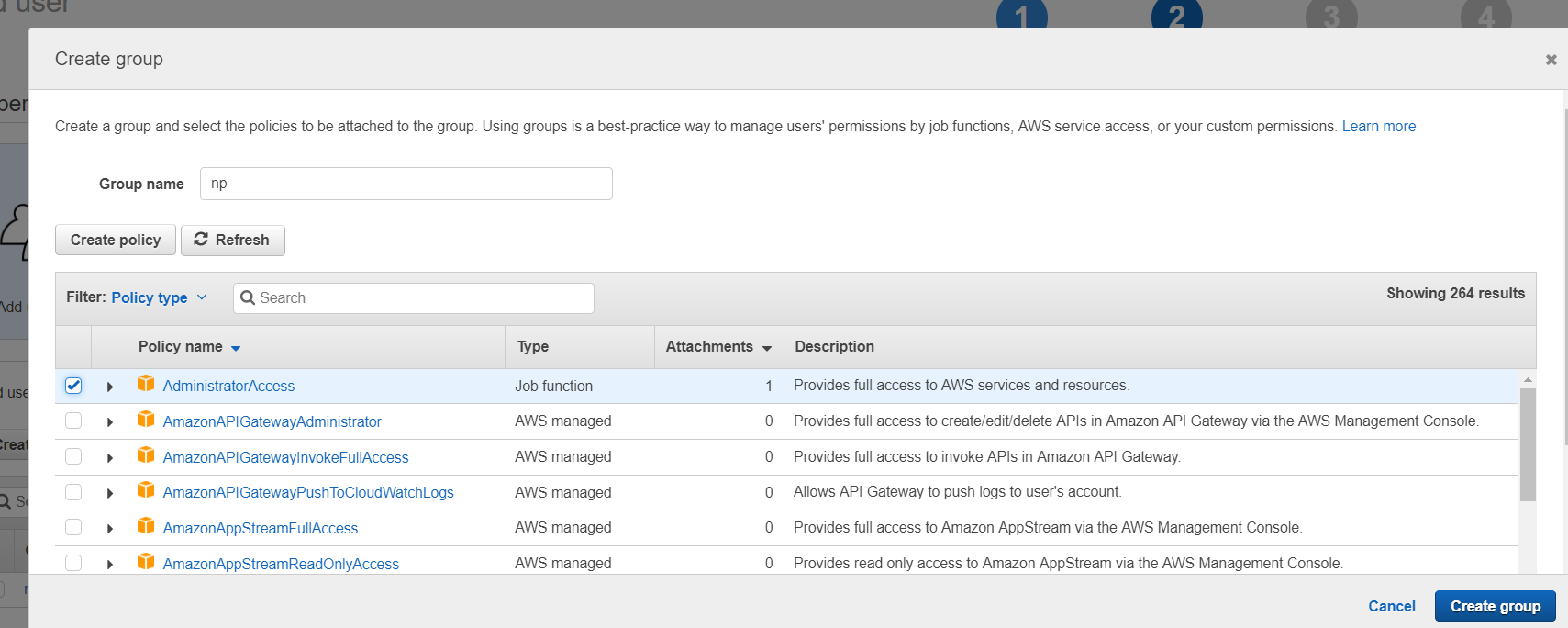
1. Open the IAM console. >> Go to services in AWS and select IAM … //https://console.aws.amazon.com/iam/home?region=us-east-1#/home
2. From the navigation menu, click Users.



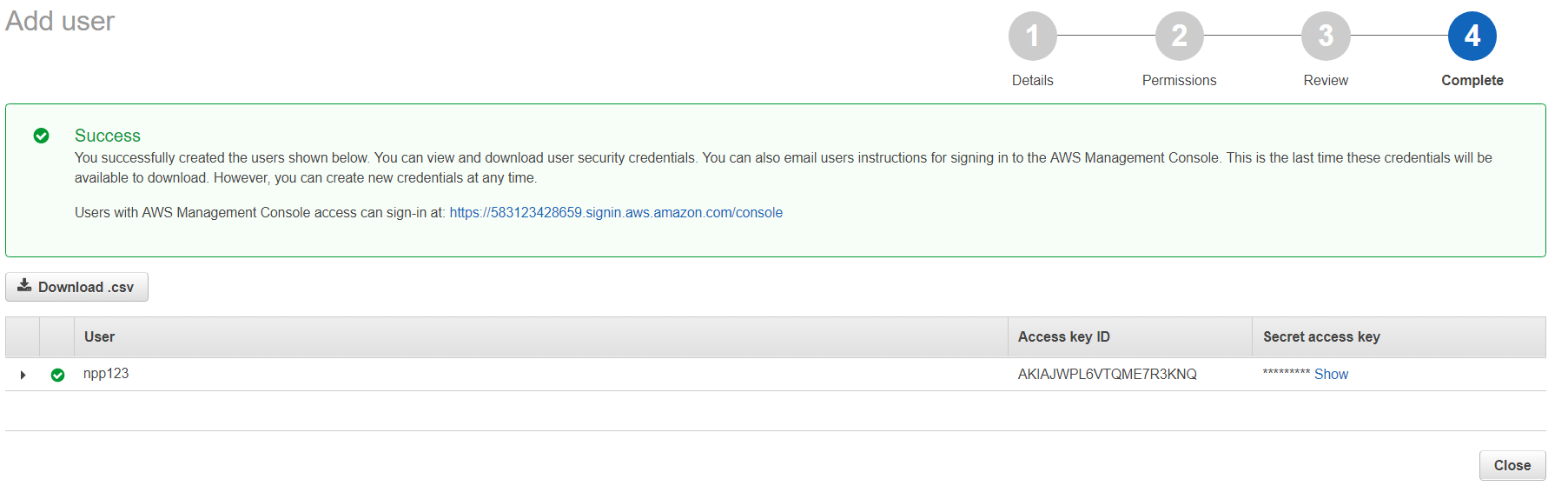
1. Select your IAM user name.



1. Next permissions >> Create a group. Select Policy name as Administrative Access and create a group.



1. After user group creation, then Review 🡪 Create User.
2. The page looks like,



Your keys will look something like this:

* + Access key ID example: AKIAIOSFODNN7EXAMPLE
  + Secret access key example: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

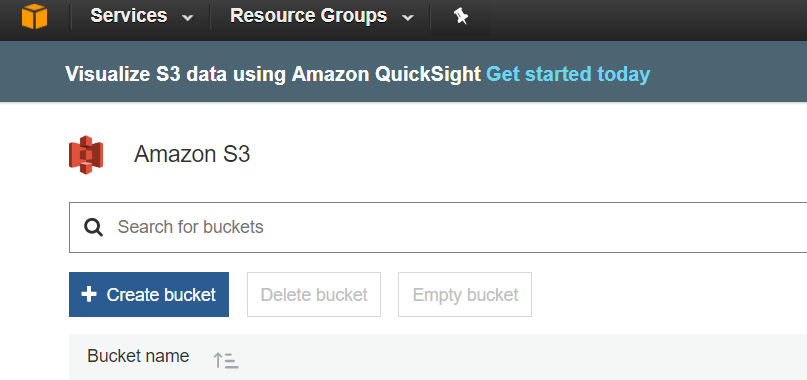
1. Click Download Credentials, and store the keys in a secure location for future reference.

**2.3** **To add your access keys to the AWS Toolkit for Eclipse**

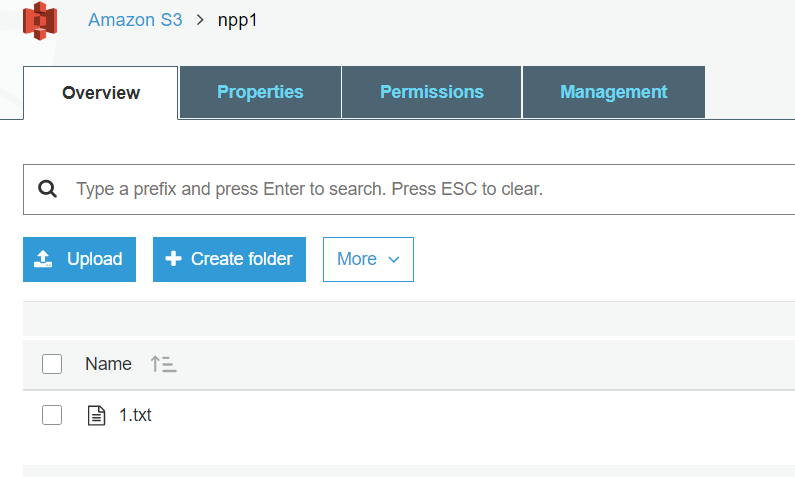
1. Open Eclipse's **Preferences** dialog box and click **AWS Toolkit** in the sidebar.
2. Type or paste your AWS access key ID in the **Access Key ID** box.
3. Type or paste your AWS secret access key in the **Secret Access Key** box.
4. Click **Apply** or **OK** to store your access key information.

**Step 3 : AWS S3 Creation in AWS console.**

Step 3.1 Create bucket.

****

Step 3.2 Only enter the bucket name >> remaining should be default values.

Step 3.3 Once the bucket was created, Select the bucket >> upload >> Choose any file (Go with text file initially Example: 1.txt) and upload the same into the bucket.

Step 3.4. The data was successfully stored in to AWS S3 ☺ ☹ ☺

Step 4 : Create a buck using NetBeans IDE.

Step 4.1 : Create a java application and add a java class named as “CreateBucket”

Step 4.2: Copy & Paste the following code.

public class CreateBucket {

private static String bucketName = "npp2";

public static void main(String[] args) throws IOException {

AmazonS3 s3client = new AmazonS3Client(new ProfileCredentialsProvider());

s3client.setRegion(Region.getRegion(Regions.DEFAULT\_REGION));

try {

if(!(s3client.doesBucketExist(bucketName)))

{

// Note that CreateBucketRequest does not specify region. So bucket is

// created in the region specified in the client.

s3client.createBucket(new CreateBucketRequest(

bucketName));

}

// Get location.

String bucketLocation = s3client.getBucketLocation(new GetBucketLocationRequest(bucketName));

System.out.println("bucket location = " + bucketLocation);

} catch (AmazonServiceException ase) {

System.out.println("Caught an AmazonServiceException, which " +

"means your request made it " +

"to Amazon S3, but was rejected with an error response" +

" for some reason.");

System.out.println("Error Message: " + ase.getMessage());

System.out.println("HTTP Status Code: " + ase.getStatusCode());

System.out.println("AWS Error Code: " + ase.getErrorCode());

System.out.println("Error Type: " + ase.getErrorType());

System.out.println("Request ID: " + ase.getRequestId());

} catch (AmazonClientException ace) {

System.out.println("Caught an AmazonClientException, which " +

"means the client encountered " +

"an internal error while trying to " +

"communicate with S3, " +

"such as not being able to access the network.");

System.out.println("Error Message: " + ace.getMessage());

}

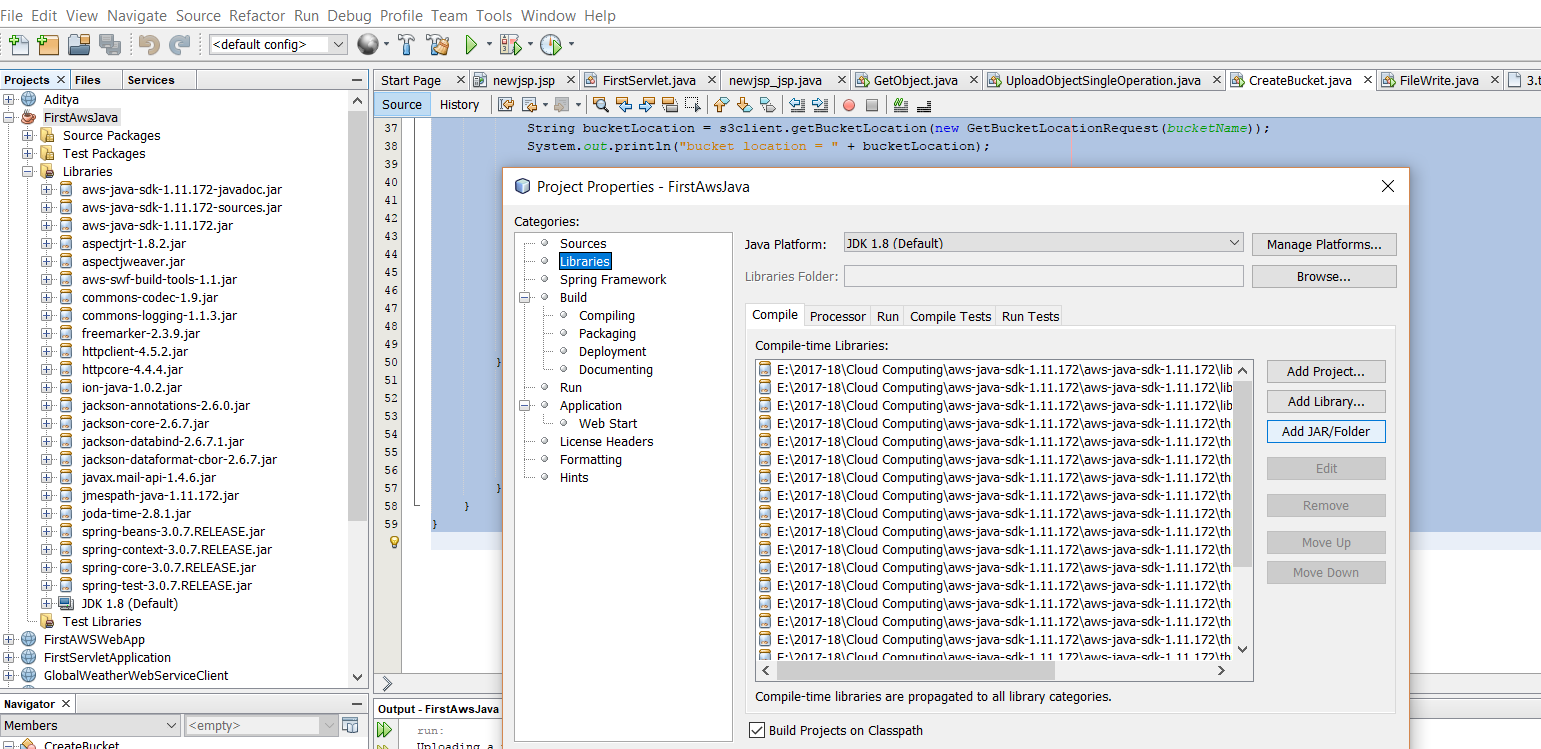
}

}

Note : It may throw some compile time error >> to resolve this add the necessary JARs into the projects.

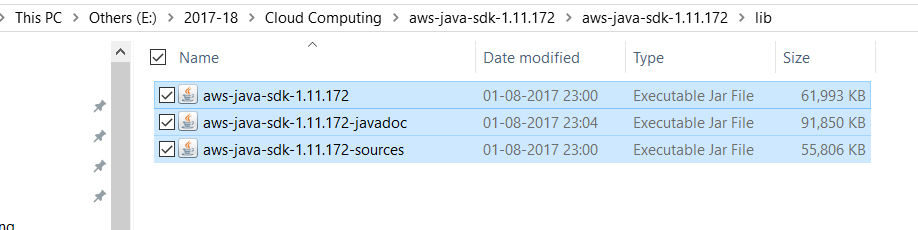
Step 4.4 How to add JARs into the project.

Right the project >> properties >> Libraries >> Add JAR/Folder

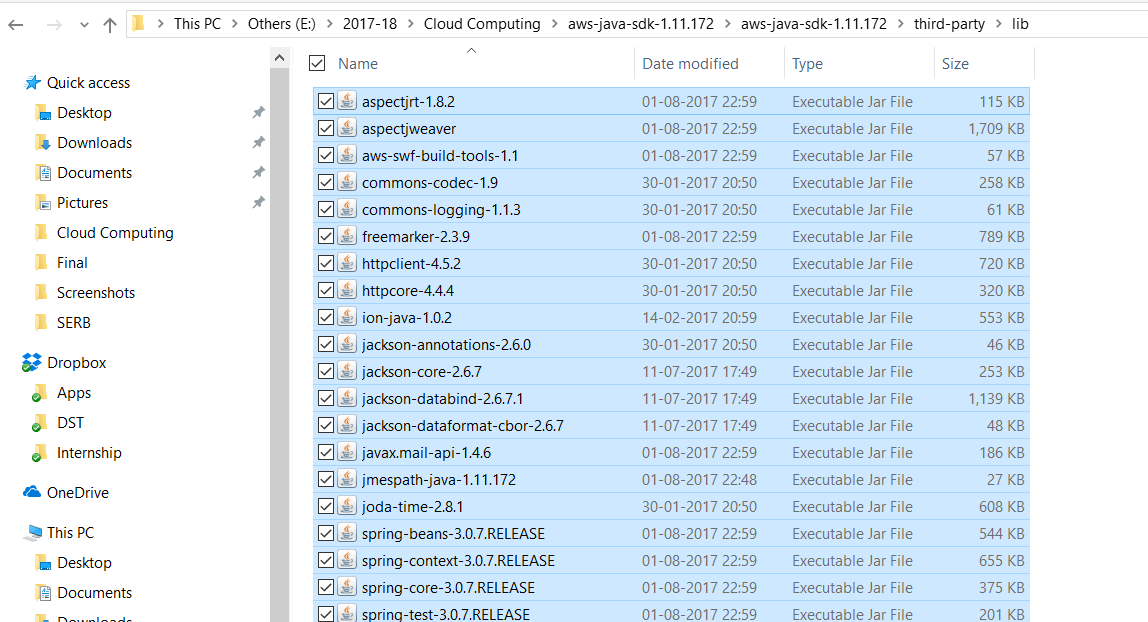


Then select the JAR files from the extracted folder (Step 1).

The following JARs to be selected (all the Jars from lib)



All the Jars from third-party > lib



This will resolve the compilation error.

Step 4.5 Once the bucket was created successfully. It will display bucket location = some data center location

Step 4.6 Upload the object into a bucket. ( Created a class called UploadObjectSingleOperation)

public class UploadObjectSingleOperation {

private static final String bucketName = "npp1"; // Bucket name

private static final String keyName = "3.txt"; // Key name

private static final String uploadFileName = "C:\\Users\\Prakash P\\Documents\\NetBeansProjects\\FirstAwsJava\\3.txt"; //File Path

public static void main(String[] args) throws IOException {

AmazonS3 s3client = new AmazonS3Client(new ProfileCredentialsProvider());

try {

System.out.println("Uploading a new object to S3 from a file\n");

File file = new File(uploadFileName);

s3client.putObject(new PutObjectRequest(

bucketName, keyName, file));

} catch (AmazonServiceException ase) {

System.out.println("Caught an AmazonServiceException, which " +

"means your request made it " +

"to Amazon S3, but was rejected with an error response" +

" for some reason.");

System.out.println("Error Message: " + ase.getMessage());

System.out.println("HTTP Status Code: " + ase.getStatusCode());

System.out.println("AWS Error Code: " + ase.getErrorCode());

System.out.println("Error Type: " + ase.getErrorType());

System.out.println("Request ID: " + ase.getRequestId());

} catch (AmazonClientException ace) {

System.out.println("Caught an AmazonClientException, which " +

"means the client encountered " +

"an internal error while trying to " +

"communicate with S3, " +

"such as not being able to access the network.");

System.out.println("Error Message: " + ace.getMessage());

}

}

}

Step 4.7 : Get the object stored in the AWS S3

public class GetObject {

private static String bucketName = "npp1";

private static String key = "1.txt";

public static void main(String[] args) throws IOException {

@SuppressWarnings("deprecation")

AmazonS3 s3Client = new AmazonS3Client(new ProfileCredentialsProvider());

try {

System.out.println("Downloading an object");

S3Object s3object = s3Client.getObject(new GetObjectRequest(

bucketName, key));

System.out.println("Content-Type: " +

s3object.getObjectMetadata().getContentType());

displayTextInputStream(s3object.getObjectContent());

// Get a range of bytes from an object.

GetObjectRequest rangeObjectRequest = new GetObjectRequest(

bucketName, key);

rangeObjectRequest.setRange(0, 10);

S3Object objectPortion = s3Client.getObject(rangeObjectRequest);

System.out.println("Printing bytes retrieved.");

displayTextInputStream(objectPortion.getObjectContent());

} catch (AmazonServiceException ase) {

System.out.println("Caught an AmazonServiceException, which" +

" means your request made it " +

"to Amazon S3, but was rejected with an error response" +

" for some reason.");

System.out.println("Error Message: " + ase.getMessage());

System.out.println("HTTP Status Code: " + ase.getStatusCode());

System.out.println("AWS Error Code: " + ase.getErrorCode());

System.out.println("Error Type: " + ase.getErrorType());

System.out.println("Request ID: " + ase.getRequestId());

} catch (AmazonClientException ace) {

System.out.println("Caught an AmazonClientException, which means"+

" the client encountered " +

"an internal error while trying to " +

"communicate with S3, " +

"such as not being able to access the network.");

System.out.println("Error Message: " + ace.getMessage());

}

}

private static void displayTextInputStream(InputStream input)

throws IOException {

// Read one text line at a time and display.

BufferedReader reader = new BufferedReader(new

InputStreamReader(input));

while (true) {

String line = reader.readLine();

if (line == null) break;

System.out.println(" " + line);

}

System.out.println(); } }

How do I create a file and write to it in Java? // May be useful for lab evaluation

Creating a text file (note that this will overwrite the file if it already exists):

try{

PrintWriter writer = new PrintWriter("the-file-name.txt", "UTF-8");

writer.println("The first line");

writer.println("The second line");

writer.close();

} catch (IOException e) {

// do something

}