DALHOUSIE UNIVERSITY

CSCI 5410

Serverless Data Processing

Assignment 1

Name : Sagarkumar Pankajbhai Vaghasia

CSID : vaghasia

Banner ID : #B00878629

Gitlab Link :

https://git.cs.dal.ca/vaghasia/csci5410-f23-b00878629/-/tree/A1

PART-A Reading Task

Mitigating Cold Start Problem in Serverless Computing: A Reinforcement Learning Approach

As the technology is evolving the need of infrastructure, storage, resources are increasing for organizations. These needs are fulfilled by serverless computing with the introduction of Function-as-a-Service (FaaS) but alongwith it also has some challenges related to performance. One of them is cold start delay which occurs due to preparation time require for a function to execute. There are some fixed mechanisms used by popular platforms to reduce cold start delay. However, they are not appropriate for dynamic environment. The authors have proposed an optimal approach which determines the best policy for keeping the containers warm according to the invocations of the functions over the time.

The major issue that encountered is preparation time required by the container which is also known as cold start delay. Alongwith this, another point to consider is resource consumption. There are many ways to reduce cold start delay such as reducing container preparation delay by pre-warmed stem cell containers, reducing cold start occurrence by reusing containers. This paper provides an approach which not only reduces the cold start delay but also reduces the resource consumption.

The Openwhisk platform method keeps the containers warm for a fixed period which is called as idle-container window. As the occurrence of the function invocation varies over time, taking a constant occurrence value is not feasible. A two-layer approach is proposed by the authors to make the platform adaptable. In first layer, using the reinforcement learning, the best time for the idle-container window is determined which can reduce the number of the cold start delays. As we know that the longer the window is available, the cold start delays will be shorter, but more resource consumption will take place. So, authors find a way to predict the time intervals between invocations in the past which determines the value of the idle-container window in the future. The first layer steps are done by completing 4 stages: 1) Receiving the function execution log which includes the time interval. 2) Predicting the value of the idle-container window by using an equation of Gaussian probability distribution in terms of mean and standard deviation. 3) Acting by updating the new value of the idle-container window. 4) At the end, Reward function is calculated to balance the number of delays and memory wastes. In second layer, LSTM (long short-term memory) approach is used which predicts the maximum number of concurrent invocations to arrange the pre-warmed containers.

Authors have performed an experiment to compare the efficiency of the proposed two-layer approach with the current approaches (Openwhisk). They took two types of datasets where one includes sequential invocations and the other includes concurrent invocations. They also took computing platform as Openwhisk and setup a training environment. In this experiment, they created 10,000 entries with different average entry rates of invocations per hour such as 5, 10 and 20. Based on results, they came to know that the proposed two-layer approach is more able to predict invocations and execute them in the pre-warmed containers Therefore, it is clear from the outcomes and the figures that the proposed approach is 22.65% more efficient than other approaches.

References :

P. Vahidinia, B. Farahani and F. S. Aliee, "Mitigating Cold Start Problem in Serverless Computing:

A Reinforcement Learning Approach," in IEEE Internet of Things Journal, doi:

10.1109/JIOT.2022.3165127.

URL: https://ieeexplore-ieee-org.ezproxy.library.dal.ca/document/9749611

PART-B AWS S3 Experiment

A flowchart defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. figure 1 is a flowchart which shows the steps for creating the bucket in Amazon S3 console and then uploading index.html file in the bucket.

The steps to create a bucket and host the html file is easier to understand by the following description :

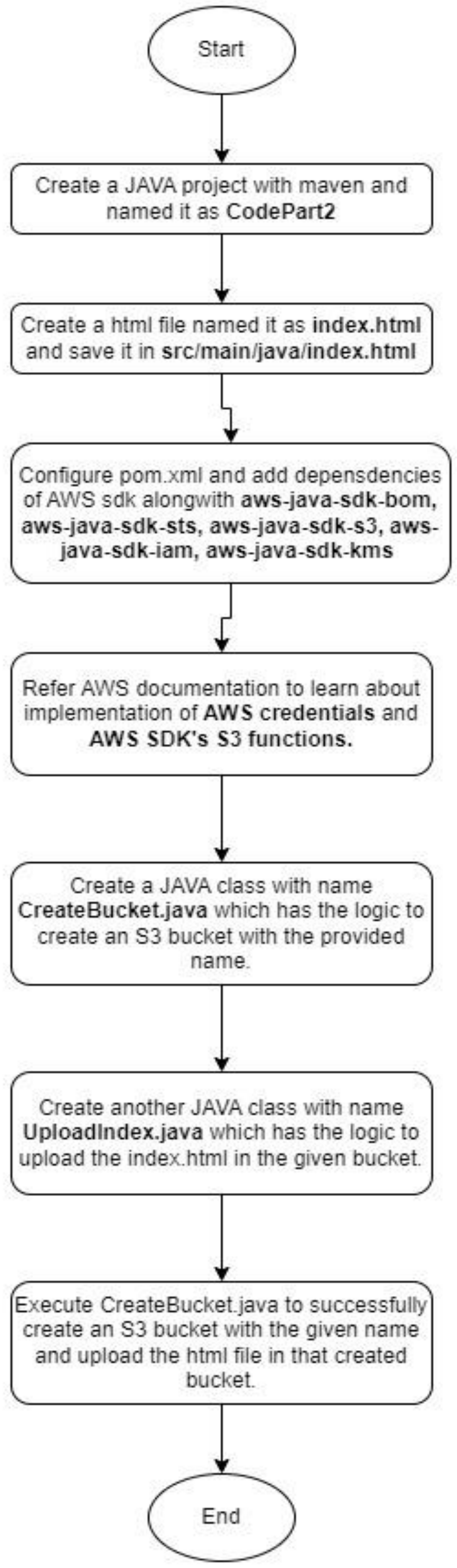
Firstly, I logged in to the Amazon AWS Academy by student login. In the home screen, there is a modules tab which redirects to my modules section. In that, I selected Learner Lab – Foundational Services which took me to the AWS console where I clicked on the Start Lab button the start the lab. Once the lab is started the Red blink near to the AWS will turn into green. After starting the lab, I clicked on AWS details button to get aws_access_key,0020aws_secret_access_key, aws_session_token to setup the connection of JAVA code with AWS. After setup, I clicked on the AWS button with a green blink which took me Console Home. This console home has different options such as IAM, EC2, S3, RDS, Lambda and many more. I simply clicked on S3 as we will be working on S3 for this assignment. This took me to the S3 console home. Here, we can manually create the S3 bucket and upload and host the files. But as we are instructed to do it so by JAVA program, so we are not going to create the bucket manually.

Then, I explored AWS SDKs[1] for java under AWS documentation which give me idea about coding part in java for connecting my java code with AWS and uploading and hosting index.html on AWS.

I write the code which is submitted to GitLab and at the end of this document. My JAVA code created the bucket named as freezingfire and then uploads a file[2] in that bucket which is named as index.html.

After uploading the file to S3 bucket, we have to edit the Bucket Policy[6] which makes the index.html file publicly available. After making changes to the Bucket Policy which I referred from AWS documentation to change bucket policy the index.html file will be available for public access.

At the end, I must edit the static website hosting options under Properties section of my bucket freezingfire. By default, this option is disable. I enabled this option, and selected Host a static website option where I specified index.html under the Index document section and clicked on save changes[3][4][5].


Start
Create a JAVA project with maven and named it as CodePart2
Create a html file named it as index.html and save it in src/main/java/index.html
Configure pom.xml and add depensdencies of AWS sdk alongwith aws-java-sdk-bom, aws-java-sdk-sts, aws-java-sdk-s3, aws-java-sdk-iam, aws-java-sdk-kms
Refer AWS documentation to learn about implementation of AWS credentials and AWS SDK's S3 functions.
Create a JAVA class with name CreateBucket.java which has the logic to create an S3 bucket with the provided name.
Create another JAVA class with name UploadIndex.java which has the logic to upload the index.html in the given bucket.
Execute CreateBucket.java to successfully create an S3 bucket with the given name and upload the html file in that created bucket.
End

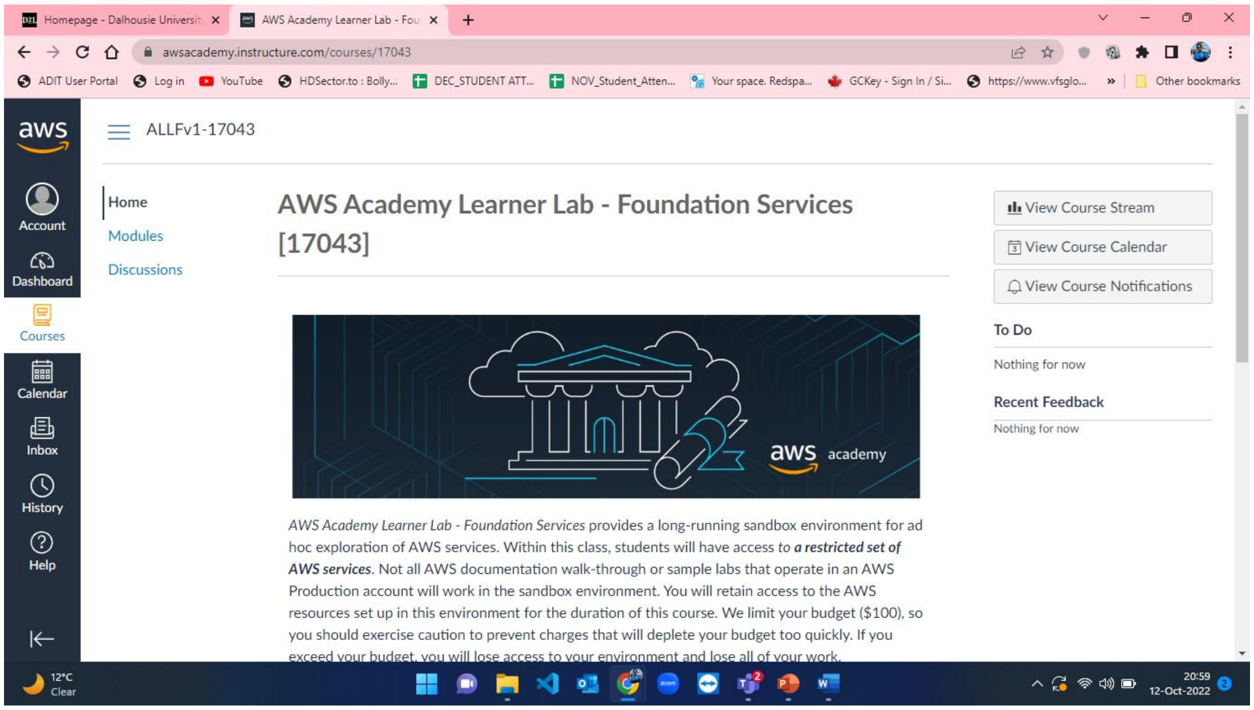
Figure–1 Flowchart of operations

Overall observation of the JAVA SDK :

To access the well-known and standard libraries the JAVA developers need to access the AWS services and utilize it by using AWS SDK for JAVA[1]. This SDK offers assistance for API lifecycles like retries, data sharding, serialization and credential management. AWS SDK for JAVA allows higher level abstractions for development. It also supports many key features Non-Blocking I/O, Automatic Pagination and many more. Many JAVA tools are also supported by AWS such as IntelliJ, Visual Studio, Eclipse and so on. AWS provides the toolkit for popular IDE's.

Steps and Screenshots of the S3 buckets and operations :

Firstly, I have created the AWS academy student account and logged in by using student credentials. The below given figure-2 represents the courses tab in AWS account. From the home screen I have selected the courses tab and in that I have selected the current course named as ALLFv1-17043.



AWS Academy Learner Lab - Foundation Services [17043]

To Do

Nothing for now

Recent Feedback

Nothing for now

AWS Academy Learner Lab - Foundation Services provides a long-running sandbox environment for ad hoc exploration of AWS services. Within this class, students will have access to ***a restricted set of AWS services***. Not all AWS documentation walk-through or sample labs that operate in an AWS Production account will work in the sandbox environment. You will retain access to the AWS resources set up in this environment for the duration of this course. We limit your budget ($100), so you should exercise caution to prevent charges that will deplete your budget too quickly. If you exceed your budget, you will lose access to your environment and lose all of your work.

Figure-2 Courses tab in AWS

After getting into the courses tab, I have selected the Modules section. figure-3 attached figure represents the modules tab in AWS. This tab contains 3 options namely : Learner's Lab – Student Guide.pdf, Learner's Lab – Foundational Services and End of course Feedback Survey. We must select Learner's Lab – Foundational Services for our module work.

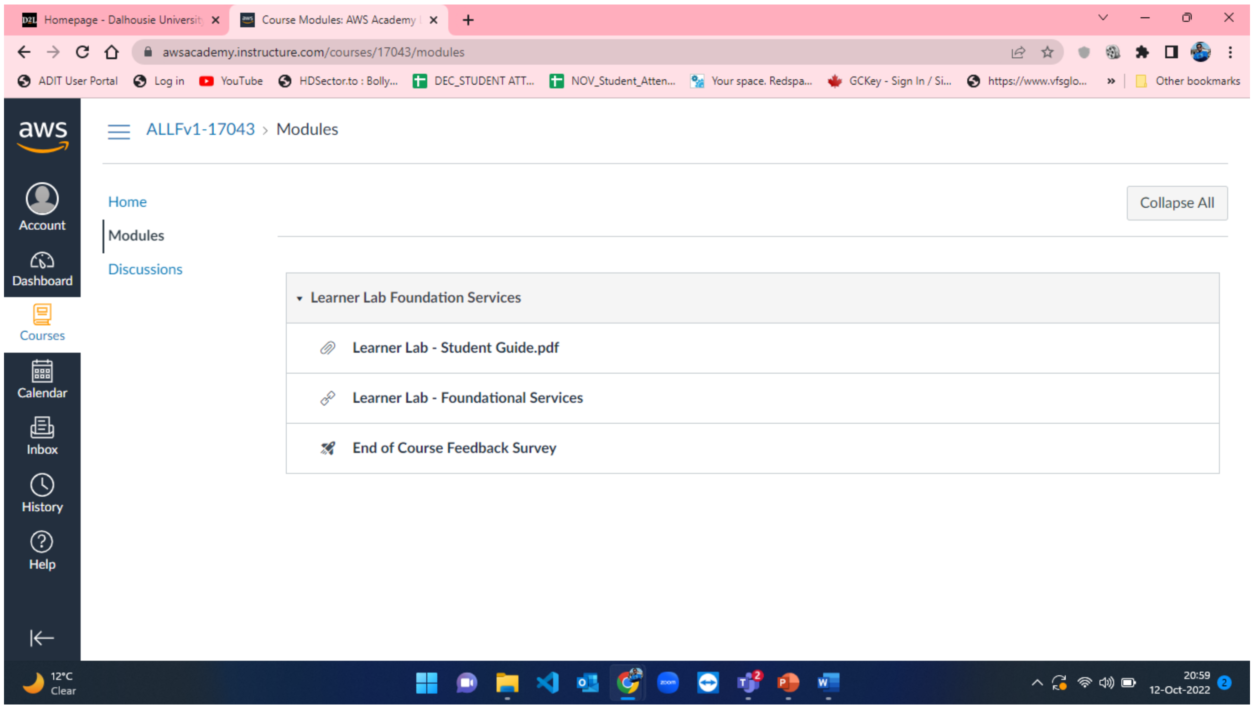
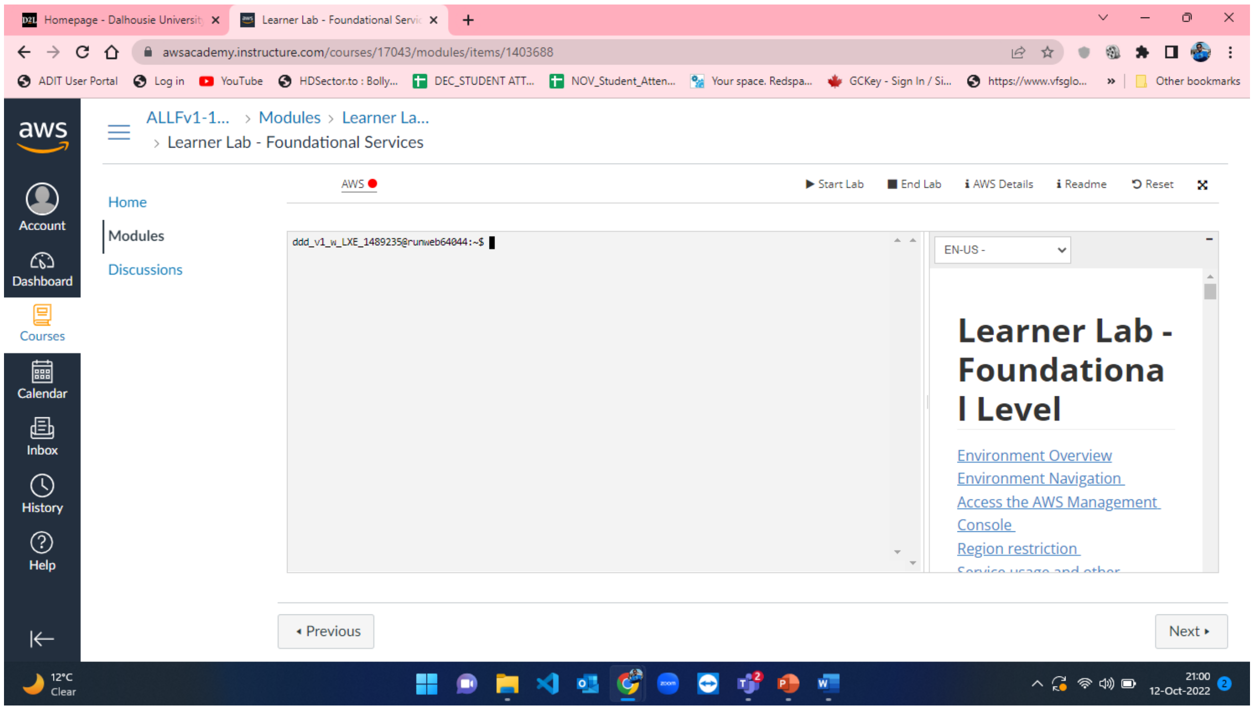

awsacademy.instructure.com/courses/17043/modules
ALLFv1-17043 › Modules
Home
Modules
Discussions
Collapse All
Learner Lab Foundation Services
Learner Lab - Student Guide.pdf
Learner Lab - Foundational Services
End of Course Feedback Survey

Figure-3 Module tab having Learner Lab in AWS

After clicking on Learner's Lab – Foundational Services a basic console is opened with many options. In this I clicked on the Start Lab, which starts the AWS console. Once it is turned on the red blink near AWS will change into green. The below given two snapshots represents AWS stopped and start lab.


awsacademy.instructure.com/courses/17043/modules/items/1403688
ALLFv1-1... › Modules › Learner La...
› Learner Lab - Foundational Services
AWS
Start Lab
End Lab
AWS Details
Readme
Reset
EN-US -
Learner Lab - Foundational Level
Environment Overview
Environment Navigation
Access the AWS Management Console
Region restriction
Previous
Next

```
ddd_v1_w_LXE_1489235@runweb64044:~$
```

Figure-4 AWS basic home console to start lab.

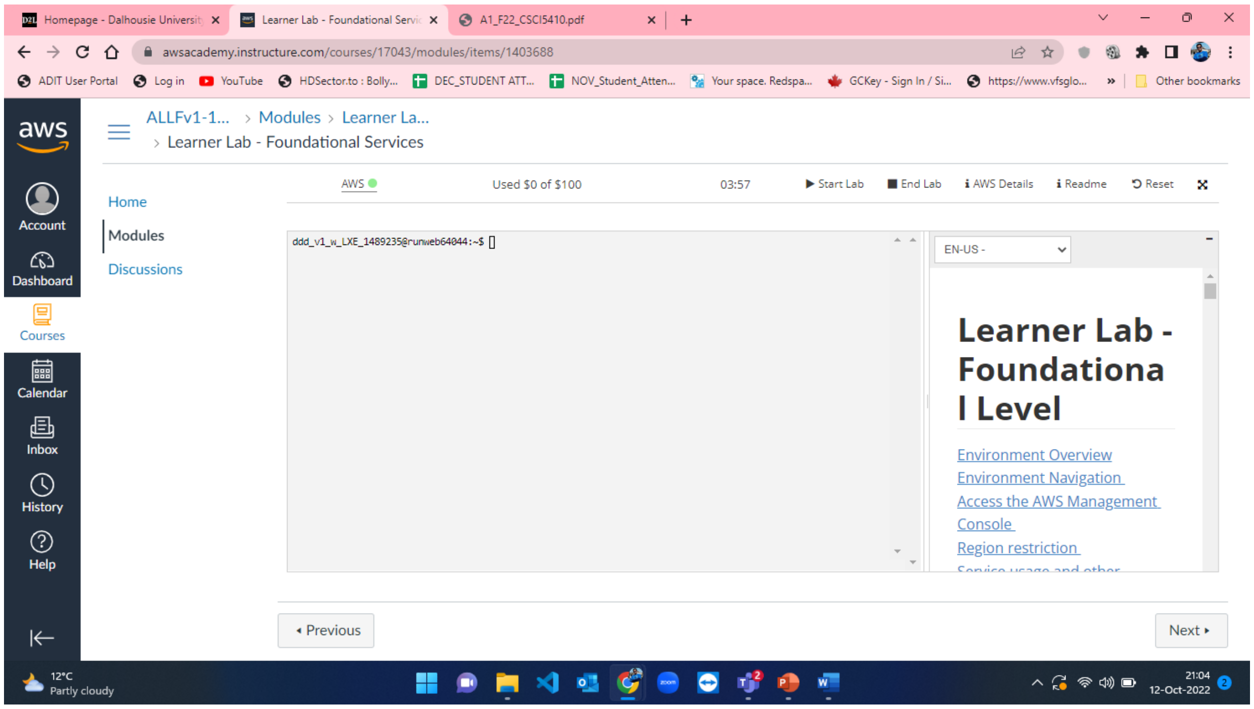

awsacademy.instructure.com/courses/17043/modules/items/1403688
ALLFv1-1... > Modules > Learner La...
> Learner Lab - Foundational Services
Home
Modules
Discussions
AWS
Used $0 of $100
03:57
Start Lab
End Lab
AWS Details
Readme
Reset
ddd_v1_w_LXE_1489235@runweb64044:~$
Learner Lab - Foundational Level
Environment Overview
Environment Navigation
Access the AWS Management Console
Region restriction
Previous
Next

Figure-5 AWS basic home console when lab is started.

After starting lab, clicking on the AWS will open the Console Home in new tab where I selected S3 as we are working on S3 bucket for this assignment. The below attached figure shows the Console home of AWS.

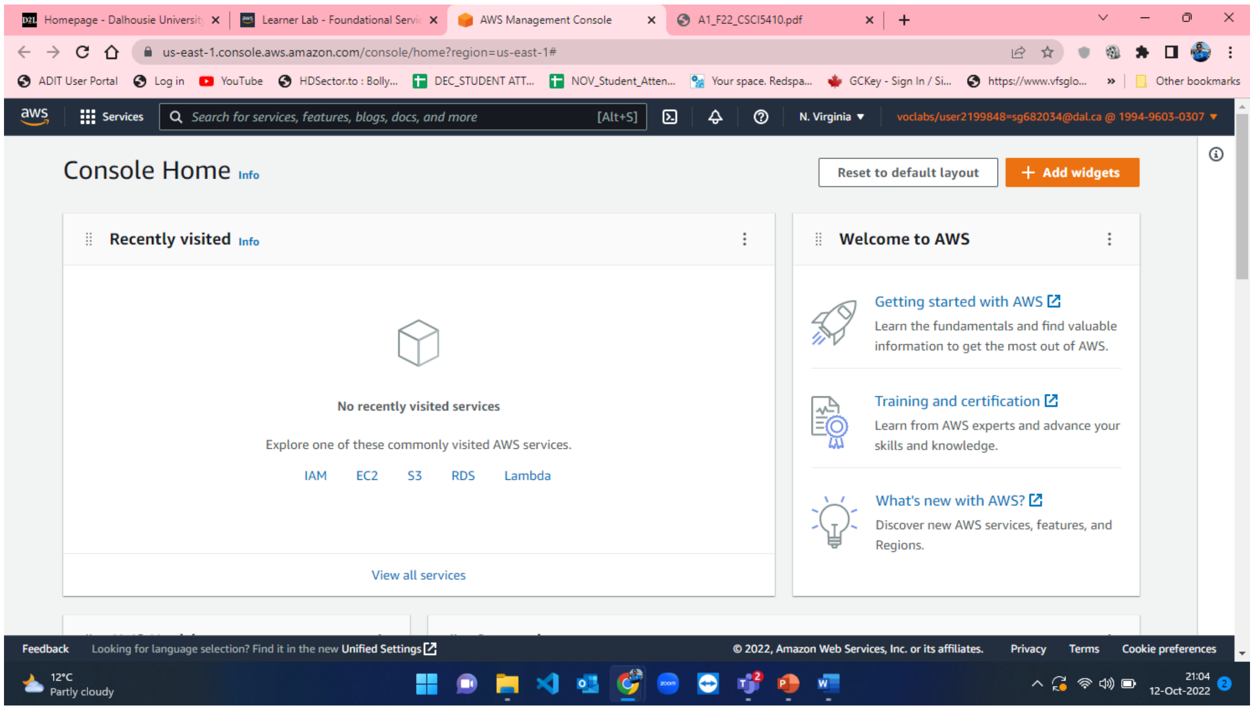
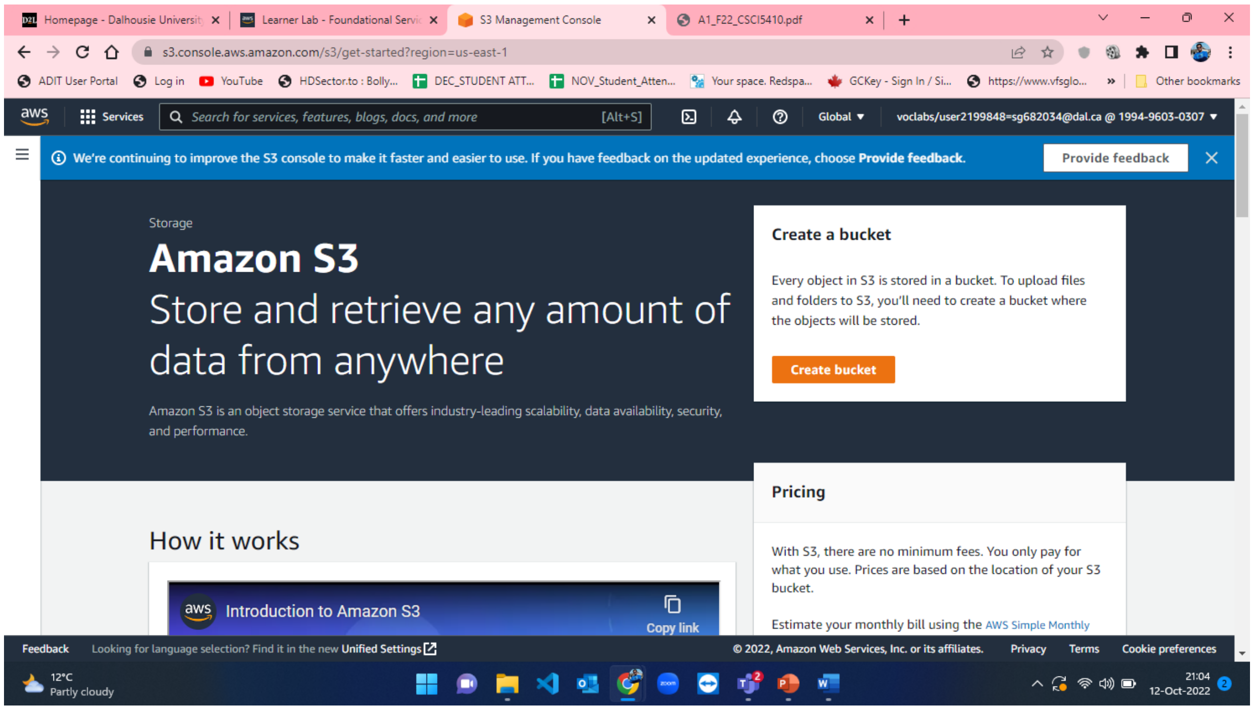

us-east-1.console.aws.amazon.com/console/home?region=us-east-1#
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user2199848=sg682034@dal.ca @ 1994-9603-0307
Console Home Info
Reset to default layout
Add widgets
Recently visited Info
No recently visited services
Explore one of these commonly visited AWS services.
IAM EC2 S3 RDS Lambda
View all services
Welcome to AWS
Getting started with AWS
Learn the fundamentals and find valuable information to get the most out of AWS.
Training and certification
Learn from AWS experts and advance your skills and knowledge.
What's new with AWS?
Discover new AWS services, features, and Regions.

Figure-6 Console home in AWS

On the Console home, clicking on S3 will open S3 tab where we can perform operations related to bucket. Figure-7 and figure-8 shows the S3 home page with common tasks in AWS where we can create a bucket, upload the object, download the object, host a static website and many more.



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Storage

Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Create bucket

Pricing

With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.

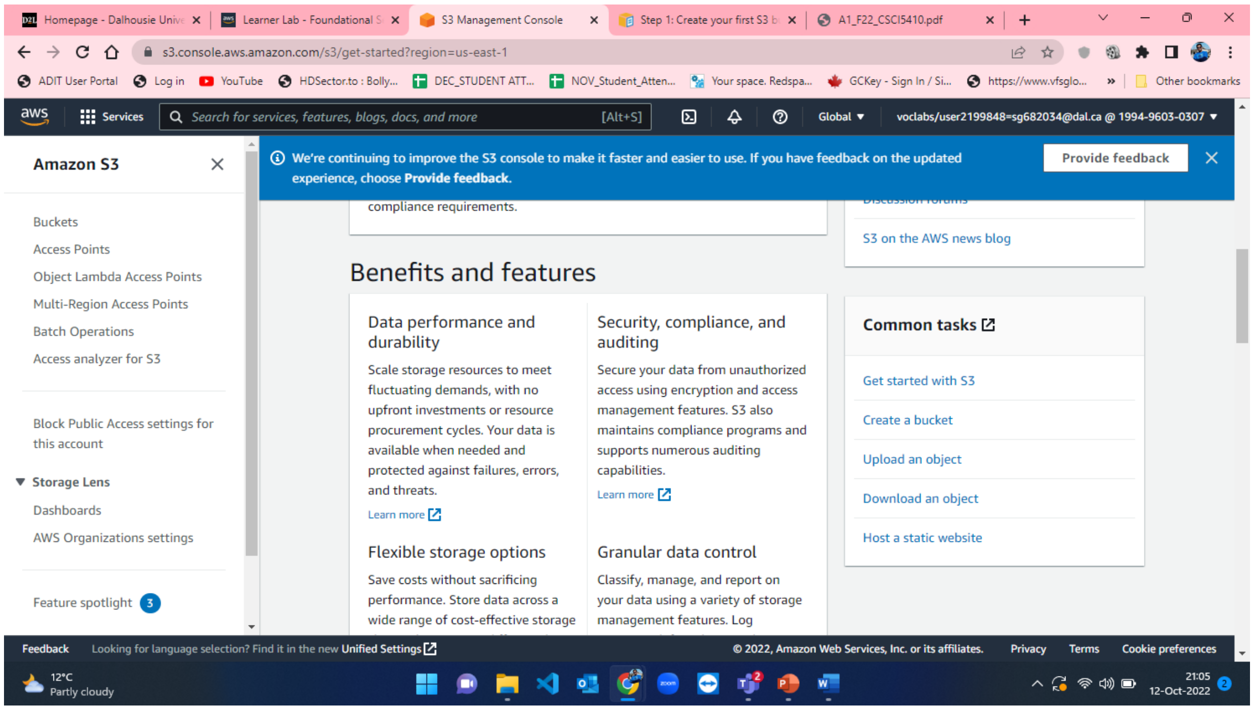
Estimate your monthly bill using the AWS Simple Monthly

How it works

Looking for language selection? Find it in the new **Unified Settings**

© 2022, Amazon Web Services, Inc. or its affiliates.

Figure-7 S3 bucket homepage for operations.


Homepage - Dalhousie Univ
Learner Lab - Foundational S
S3 Management Console
Step 1: Create your first S3 b
A1_F22_CSCI5410.pdf
s3.console.aws.amazon.com/s3/get-started?region=us-east-1
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
voclabs/user2199848=sg682034@dal.ca @ 1994-9603-0307
Provide feedback
Feedback
Privacy
Terms
Cookie preferences
12°C
Partly cloudy
21:05
12-Oct-2022

Amazon S3

Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3

Block Public Access settings for this account

Storage Lens
Dashboards
AWS Organizations settings

Feature spotlight 3

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

compliance requirements.

S3 on the AWS news blog

Benefits and features

Data performance and durability

Scale storage resources to meet fluctuating demands, with no upfront investments or resource procurement cycles. Your data is available when needed and protected against failures, errors, and threats.

Learn more

Security, compliance, and auditing

Secure your data from unauthorized access using encryption and access management features. S3 also maintains compliance programs and supports numerous auditing capabilities.

Learn more

Flexible storage options

Save costs without sacrificing performance. Store data across a wide range of cost-effective storage

Granular data control

Classify, manage, and report on your data using a variety of storage management features. Log

Common tasks

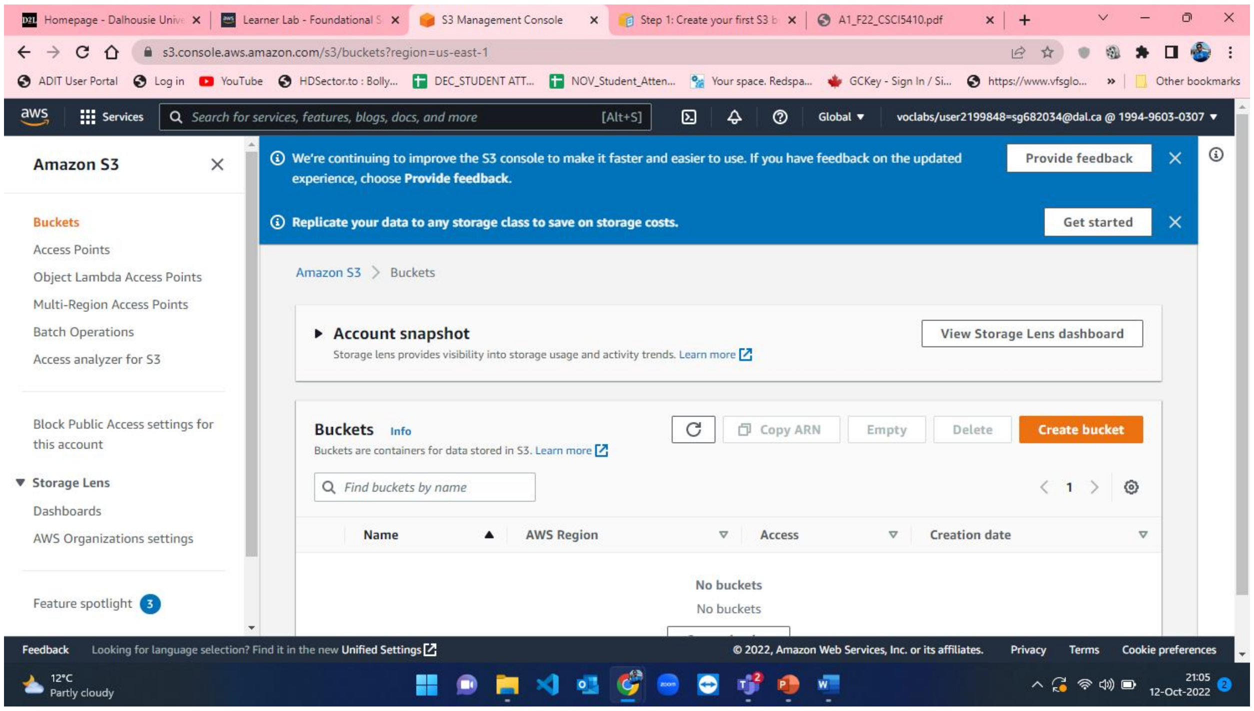
Get started with S3
Create a bucket
Upload an object
Download an object
Host a static website

Looking for language selection? Find it in the new **Unified Settings**

© 2022, Amazon Web Services, Inc. or its affiliates.

Figurer-8 Common task in S3 in AWS

Clicking on Buckets tab from the left panel, will show the number of buckets available in the account. Currently, there are no buckets that is why it shows empty. The below given figure shows buckets tab.



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Replicate your data to any storage class to save on storage costs.

Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date

No buckets

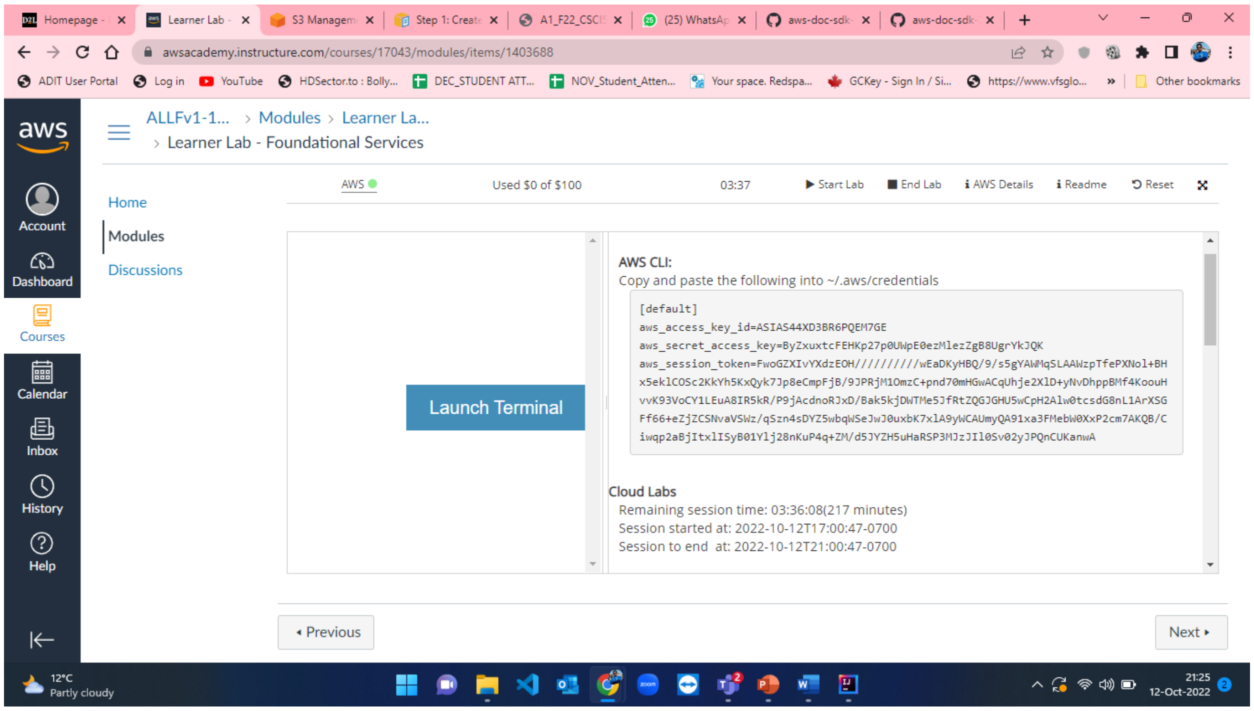
No buckets

Looking for language selection? Find it in the new **Unified Settings**

© 2022, Amazon Web Services, Inc. or its affiliates.

Figure-9 Buckets in AWS

Once exploration completes, first thing I must do is setting up the connection from my java code with AWS this can be done by having aws_access_key, aws_secret_access_key, aws_session_token. These 3 parameters can be obtained from the main console where we started the lab. There is an option named as AWS details in that under AWS CLI it shows these credentials. Figure-10 presents the parameters for the connection.


AWS CLI:
Copy and paste the following into ~/.aws/credentials

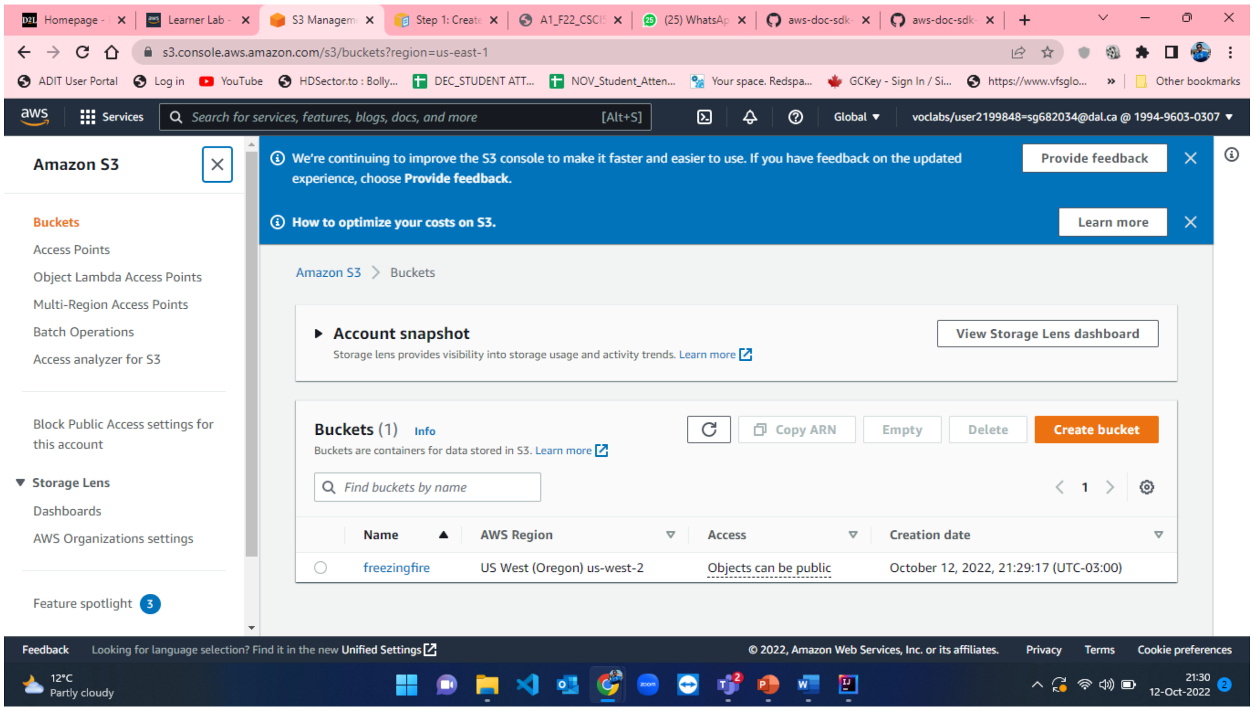
```
[default]
aws_access_key_id=ASIAS44XD3BR6PQEM7GE
aws_secret_access_key=ByZxuxtcFEHKp27p0UWpE0ezMlezZgB8UgrYkJQK
aws_session_token=FwoGZXIvYXdzEOH//////////wEaDKyHBQ/9/s5gYAWMqSLAAWzpTfePXNol+BH
x5eklCOSc2KkYh5KxQyk7Jp8eCmpFjB/9JPRjM1OmzC+pnd70mHGwACqUhje2XlD+yNvDhppBMf4KoouH
vvK93VoCY1LEuA8IR5kR/P9jAcdnoRJxD/Bak5kjDWTMe5JfRtZQGJGHU5wCpH2Alw0tcsdG8nL1ArXSG
Ff66+eZjZCSNvaVSWz/qSzn4sDYZ5wbqWSeJwJ0uxbK7xlA9yWCAUmyQA91xa3FMebW0XxP2cm7AKQB/C
iwqp2aBjItxlISyB01Ylj28nKuP4q+ZM/d5JYZH5uHaRSP3MJzJIl0Sv02yJPQnCUKanwA
```

Cloud Labs
Remaining session time: 03:36:08(217 minutes)
Session started at: 2022-10-12T17:00:47-0700
Session to end at: 2022-10-12T21:00:47-0700

awsacademy.instructure.com/courses/17043/modules/items/1403688
ALLFv1-1... > Modules > Learner La...
> Learner Lab - Foundational Services
Home
Modules
Discussions
AWS
Used $0 of $100
03:37
Start Lab
End Lab
AWS Details
Readme
Reset
Launch Terminal
Previous
Next

Figure-10 Parameters for the connection in AWS.

After writing the java code and setting up the connection with AWS, I wrote JAVA code to create the bucket. The below attached snapshot represents the created bucket named as freezingfire.


We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.

How to optimize your costs on S3.

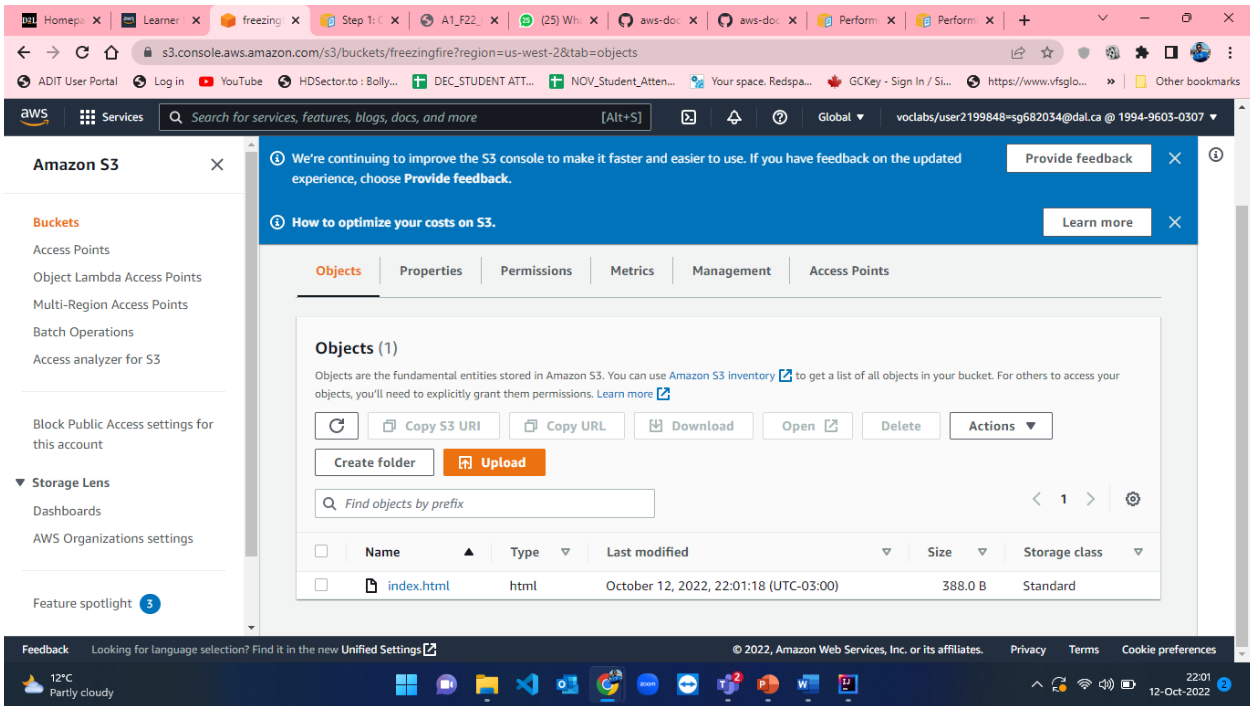
Buckets (1)
Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
freezingfire	US West (Oregon) us-west-2	Objects can be public	October 12, 2022, 21:29:17 (UTC-03:00)

s3.console.aws.amazon.com/s3/buckets?region=us-east-1
Services
Global
voclabs/user2199848=sg682034@dal.ca @ 1994-9603-0307
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
Provide feedback
Learn more
Amazon S3 > Buckets
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
Copy ARN
Empty
Delete
Create bucket
Find buckets by name

Figure-11 Bucket created in AWS

After creating bucket, I have to upload the index.html file to the newly created bucket. This is done by java code. Figure-11 represents the object uploaded under freezingfire bucket.



s3.console.aws.amazon.com/s3/buckets/freezingfire?region=us-west-2&tab=objects

voclabs/user2199848=sg682034@dal.ca @ 1994-9603-0307

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

How to optimize your costs on S3.

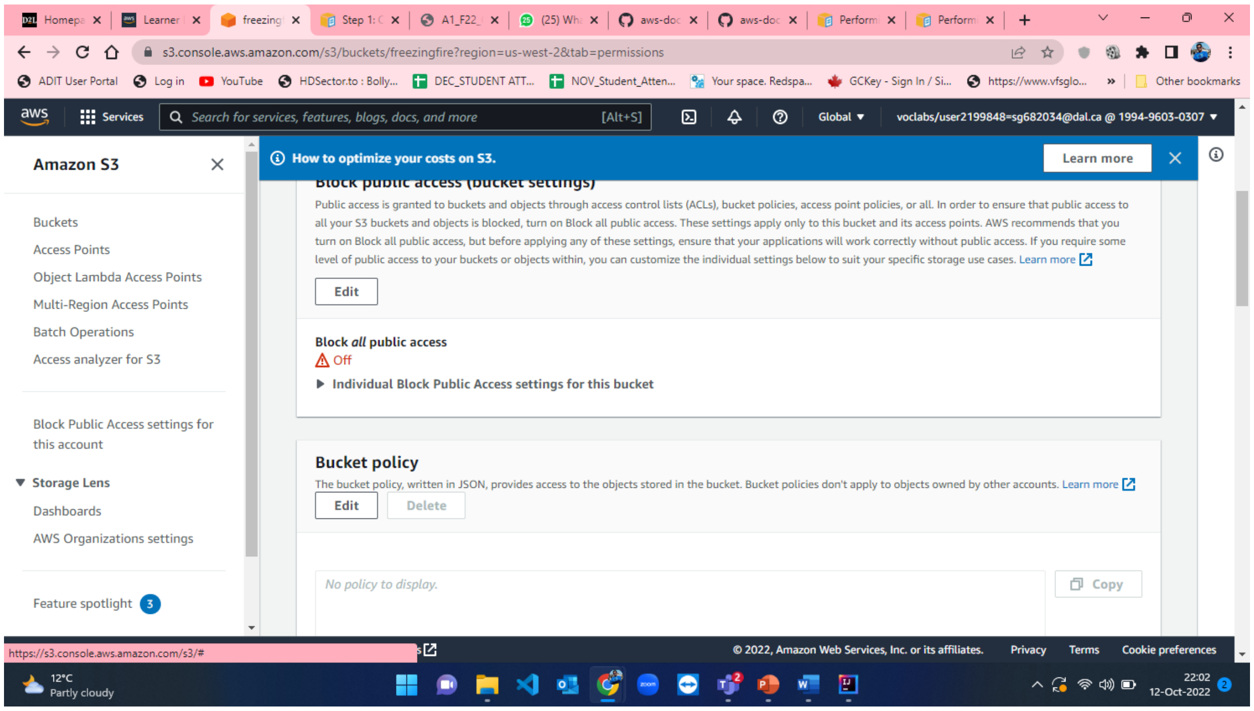
Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
index.html	html	October 12, 2022, 22:01:18 (UTC-03:00)	388.0 B	Standard

Figure-11 Object uploaded under created in AWS

After uploading object to the bucket, I have to change the bucket policy to give access the object by other accounts. I have used AWS documentation to get the JSON script for making the public access of the object. The given below snapshots shows the changes in bucket policy.



Block public access (bucket settings)

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 all your S3 buckets and 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 your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

Block *all* public access
Off

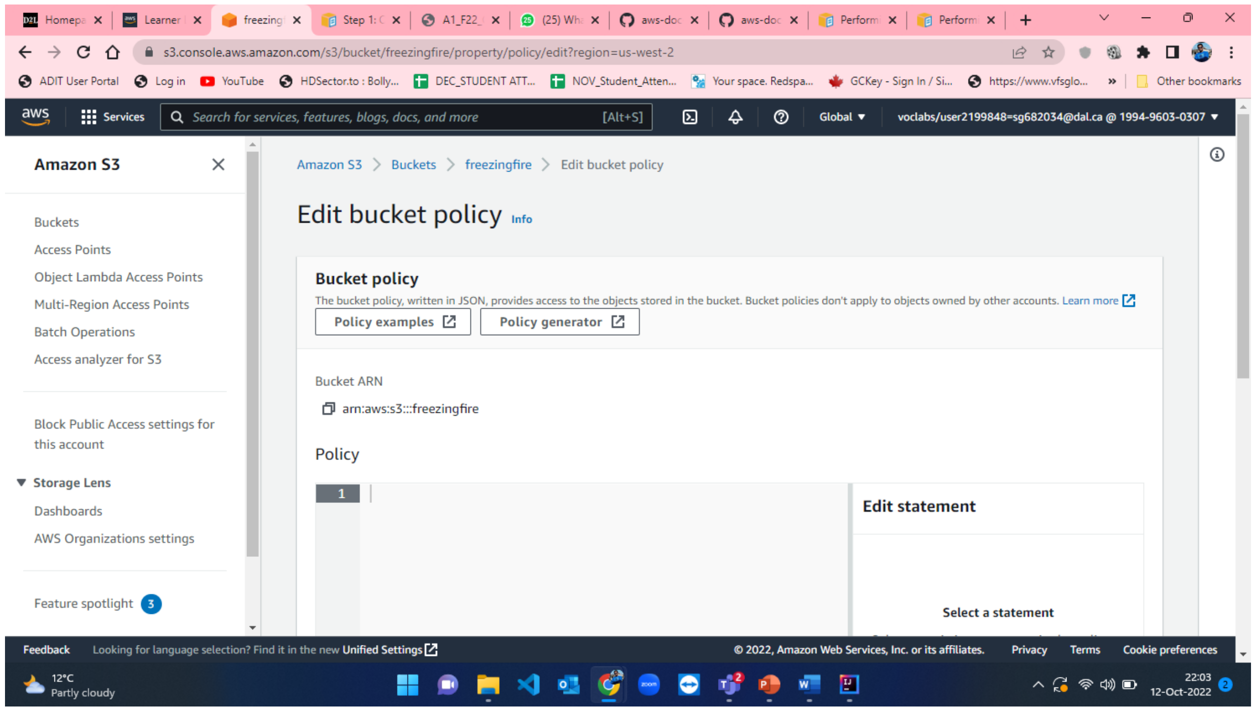
Individual Block Public Access settings for this bucket

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

No policy to display.

Figure-12 Bucket Policy in default mode.



Amazon S3 > Buckets > freezingfire > Edit bucket policy

Edit bucket policy

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

Bucket ARN

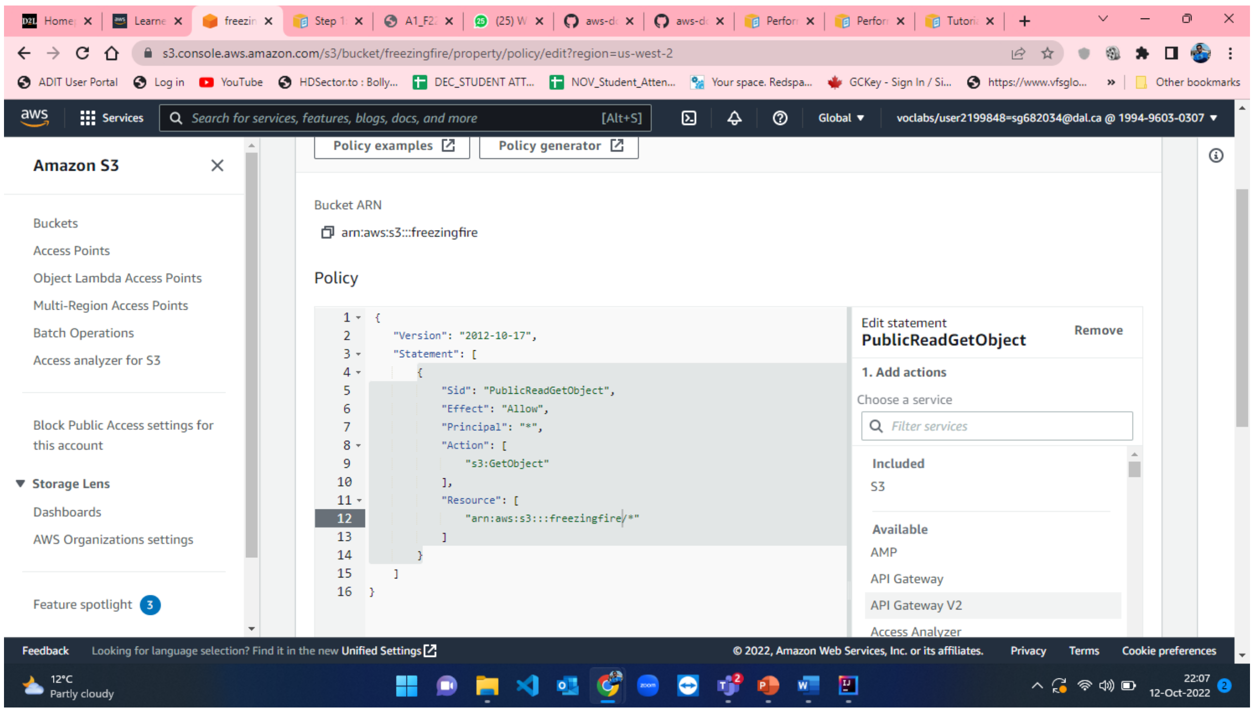
arn:aws:s3:::freezingfire

Policy

Edit statement

Select a statement

Figure-13 Editing blank bucket policy.



Bucket ARN

arn:aws:s3:::freezingfire

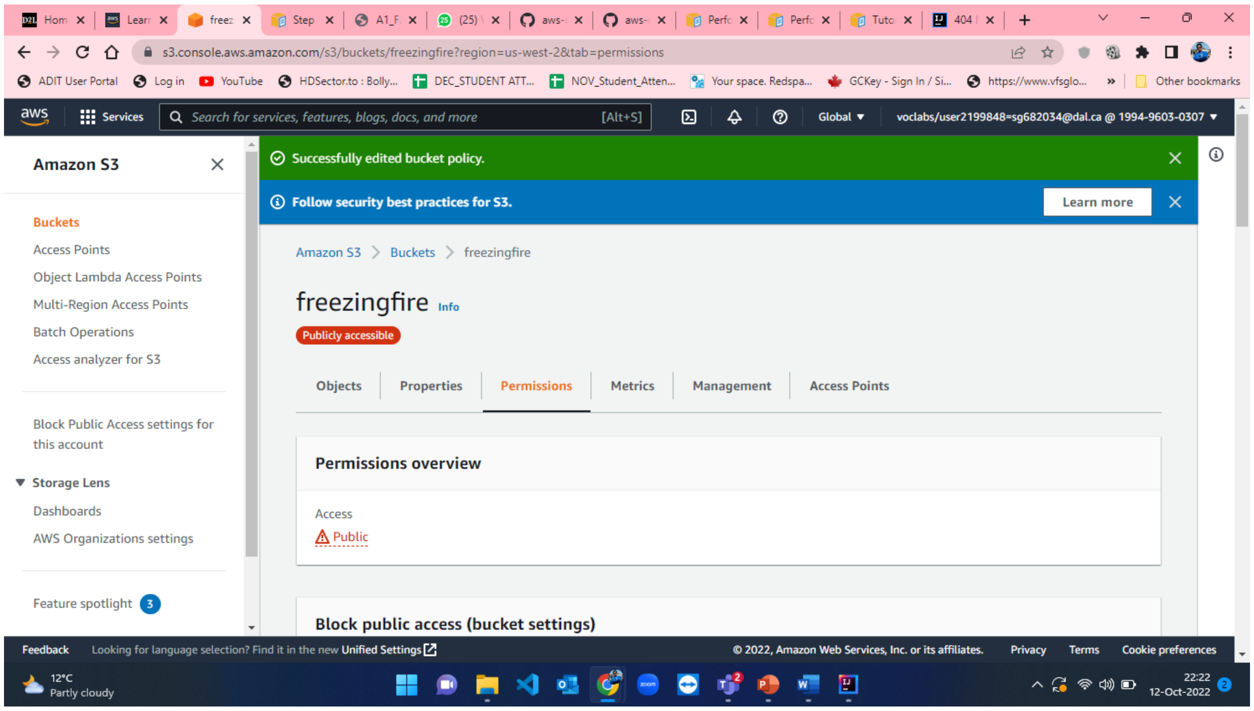
Policy

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

Edit statement: PublicReadGetObject

Included: S3

Figure-14 Adding JSON script for public access in AWS.


s3.console.aws.amazon.com/s3/buckets/freezingfire?region=us-west-2&tab=permissions
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
Learn more
Amazon S3 > Buckets > freezingfire
Info
Objects
Properties
Permissions
Metrics
Management
Access Points

Successfully edited bucket policy.

Follow security best practices for S3.

freezingfire

Publicly accessible

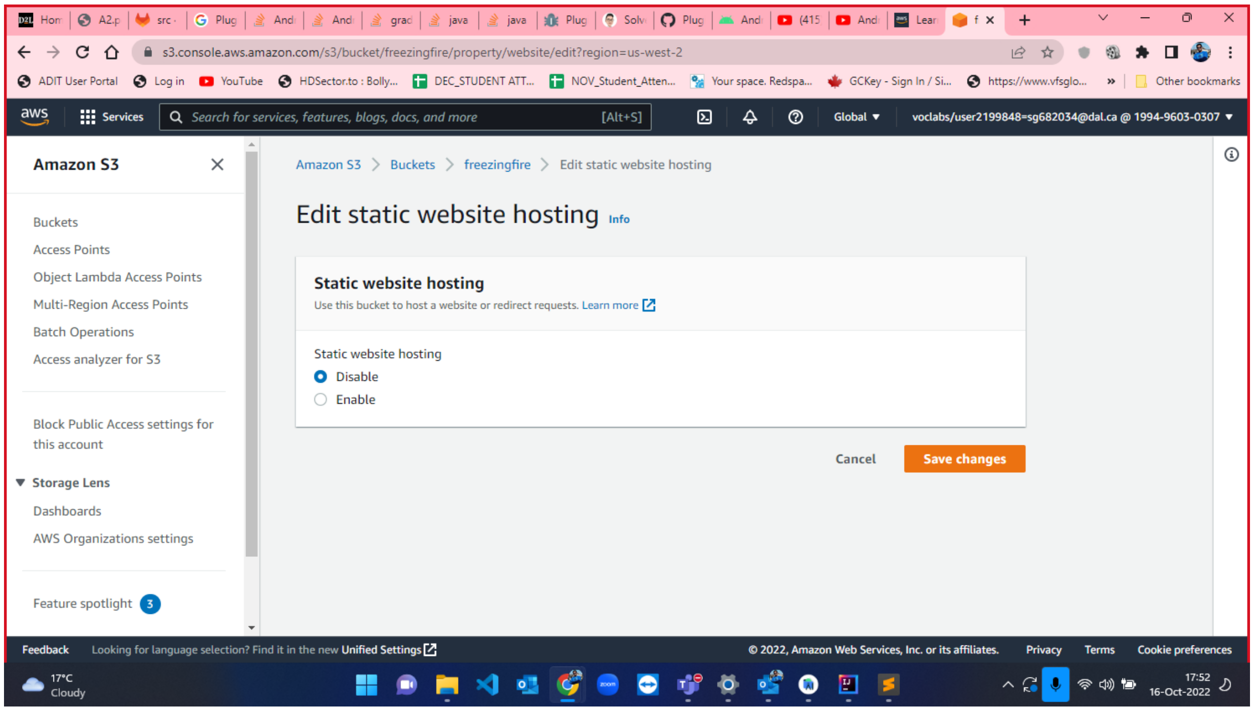
Permissions overview

Access: Public

Block public access (bucket settings)

Figure-15 Successfully saving the bucket policy.

For static web hosting, we have to enable the static web hosting. This operations is shown in the below attached figures.


Amazon S3 > Buckets > freezingfire > Edit static website hosting

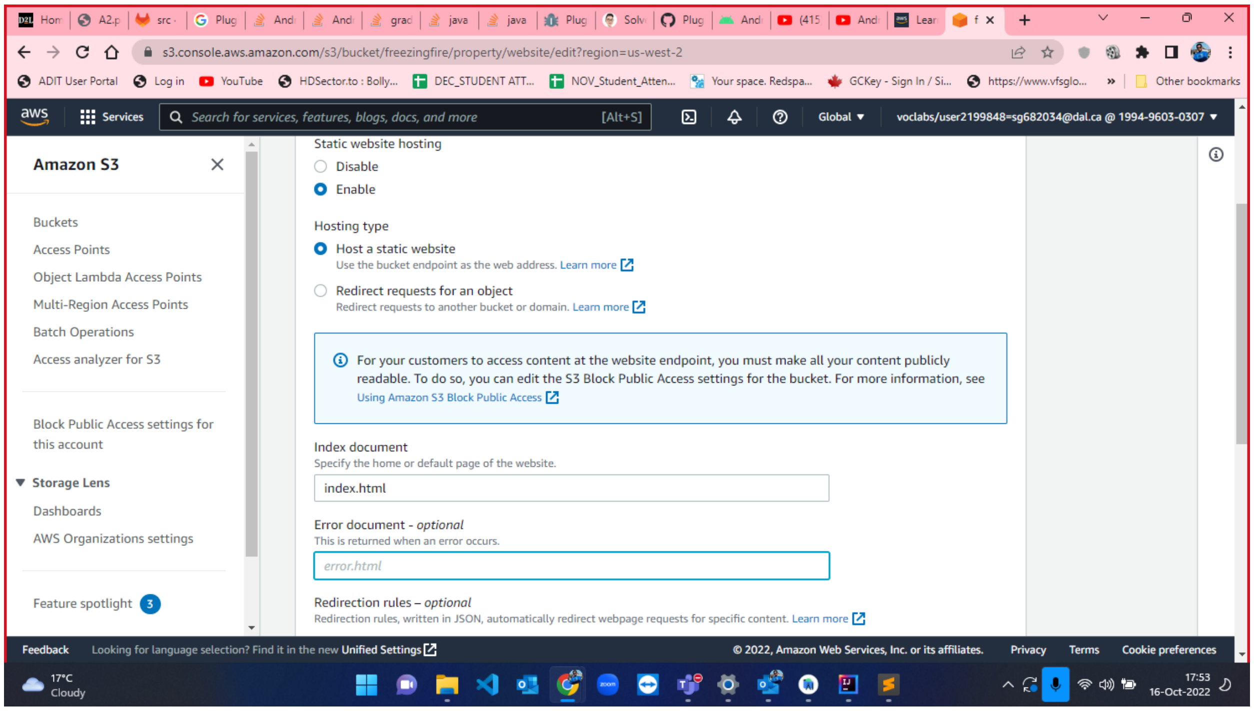
Edit static website hosting

Static website hosting
Use this bucket to host a website or redirect requests. Learn more

Static website hosting
[✓ Disable] Enable

Cancel | Save changes

Figure-16 Default disable option in static web hosting.


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

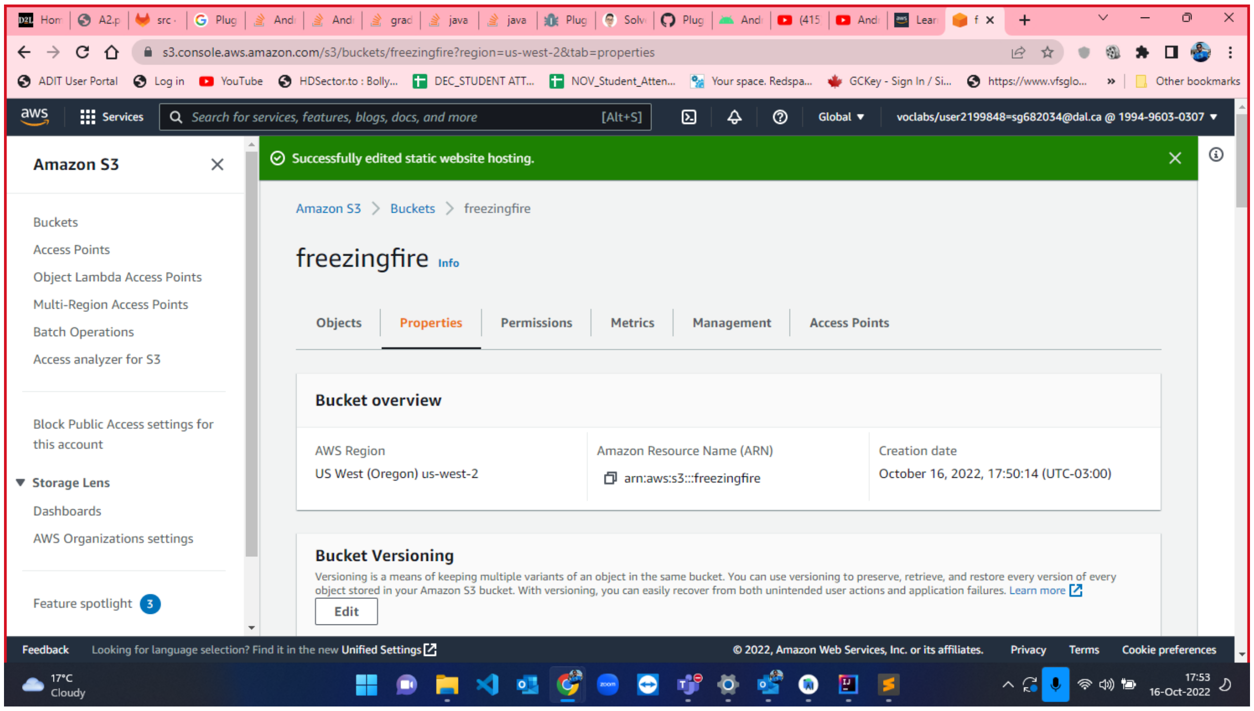
For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access

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

Figure-17 Performing operations to enable the static hosting in AWS



s3.console.aws.amazon.com/s3/buckets/freezingfire?region=us-west-2&tab=properties

Successfully edited static website hosting.

Amazon S3 > Buckets > freezingfire

freezingfire Info

Bucket overview

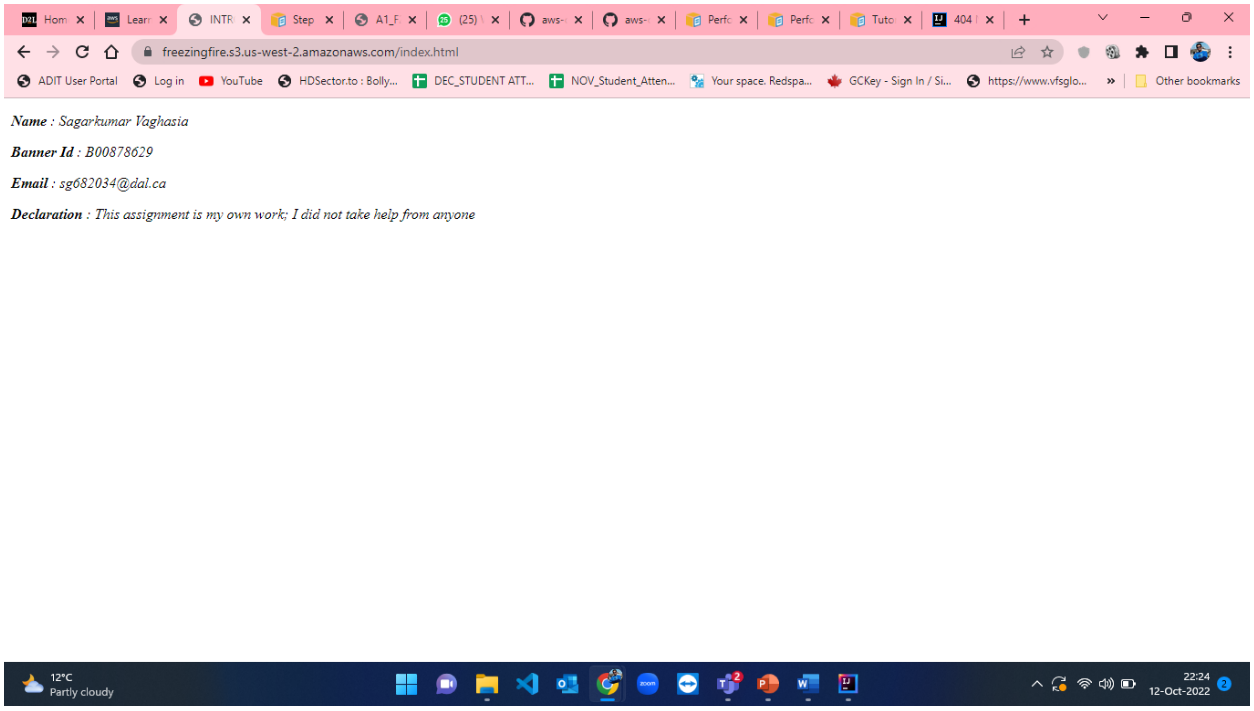
AWS Region	Amazon Resource Name (ARN)	Creation date
US West (Oregon) us-west-2	arn:aws:s3:::freezingfire	October 16, 2022, 17:50:14 (UTC-03:00)

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Figure-18 Successfully edited static web hosting in AWS.

After setting up the static web hosting and the bucket policy we can see the webpage index.html page publicly. This can be seen by the attached figure-19.



freezingfire.s3.us-west-2.amazonaws.com/index.html

Name *: Sagarkumar Vaghasia*

Banner Id *: B00878629*

Email *: sg682034@dal.ca*

Declaration *: This assignment is my own work; I did not take help from anyone*

Figure-19 Hostel index.html in AWS.

JAVA Code:

CreateBucket.java

```
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import com.amazonaws.services.s3.model.Bucket;

import java.util.List;


public class CreateBucket {

    //https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
    public static Bucket getBucket(String bucket_name) {
        final AmazonS3 s3 = AmazonS3ClientBuilder.standard()
                .withRegion(Regions.DEFAULT_REGION)
                .withCredentials(new AWSStaticCredentialsProvider(new
BasicSessionCredentials(
                        "ASIAS44XD3BR76XC7G4Q",
                        "LsNDVqZeLo07YX4oNdzMFdPvniluKStSO5AcPjwF",

"FwoGZXIvYXdzED4aDEZJW3l1fZTHZWy2BSLAARz5s62574lTWroOjeeHpywYRmoGRcRp6PLPWLSI
xG8MKX+df+yaY+SapQKNtzg8a8ZeIqyLL8jyGRC4c34ZaKlx7EopLbZ17s1om5hoPc+kYg1G+n03M
S759AlQDpp3vDIzhZ5HmW6nsCTOOuAKzYhdUj7a4RZ4ea6uqJtYwhkI4MRQQabDoMqwTi4Nl4kKBp
zXlUHEb/5IDBz7wcsLDQ/LUV5pT4iUjF5ZwVJmHZc01ouDBbAZnny6l10d+bT3uijr1bGaBjItUjW
Lg1l1eYP2Fwv7f7byrzTwW3fpoFKYJeS4BHHbQChOCK3yXIzuqQIEBaP5"
                ))).build();
        Bucket named_bucket = null;
        List<Bucket> buckets = s3.listBuckets();
        for (Bucket b : buckets) {
            if (b.getName().equals(bucket_name)) {
                named_bucket = b;
            }
        }
        return named_bucket;
    }

    //https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
    public static Bucket createBucket(String bucket_name) {
        final AmazonS3 s3 = AmazonS3ClientBuilder.standard()
                .withRegion(Regions.DEFAULT_REGION)
                .withCredentials(new AWSStaticCredentialsProvider(new
BasicSessionCredentials(
                        "ASIAS44XD3BR76XC7G4Q",
                        "LsNDVqZeLo07YX4oNdzMFdPvniluKStSO5AcPjwF",

"FwoGZXIvYXdzED4aDEZJW3l1fZTHZWy2BSLAARz5s62574lTWroOjeeHpywYRmoGRcRp6PLPWLSI
xG8MKX+df+yaY+SapQKNtzg8a8ZeIqyLL8jyGRC4c34ZaKlx7EopLbZ17s1om5hoPc+kYg1G+n03M
S759AlQDpp3vDIzhZ5HmW6nsCTOOuAKzYhdUj7a4RZ4ea6uqJtYwhkI4MRQQabDoMqwTi4Nl4kKBp
zXlUHEb/5IDBz7wcsLDQ/LUV5pT4iUjF5ZwVJmHZc01ouDBbAZnny6l10d+bT3uijr1bGaBjItUjW
Lg1l1eYP2Fwv7f7byrzTwW3fpoFKYJeS4BHHbQChOCK3yXIzuqQIEBaP5"
                ))).build();
        Bucket b = null;
        if (s3.doesBucketExistV2(bucket_name)) {
            System.out.format("Bucket %s already exists.\n", bucket_name);
            b = getBucket(bucket_name);
        } else {
            try {
                b = s3.createBucket(bucket_name);
            } catch (AmazonS3Exception e) {
                System.err.println(e.getErrorMessage());
            }
        }
        return b;
    }

    public static void main(String[] args) {

        UploadIndex uploadIndex = new UploadIndex();
        uploadIndex.uploadfile();

        final String USAGE = "\n" +
                "CreateBucket - create an S3 bucket\n\n" +
                "Usage: CreateBucket <bucketname>\n\n" +
                "Where:\n" +
                "  bucketname - the name of the bucket to create.\n\n" +
                "The bucket name must be unique, or an error will result.\n";

        String bucket_name = "freezingfire";

        System.out.format("\nCreating S3 bucket: %s\n", bucket_name);
        Bucket b = createBucket(bucket_name);
        if (b == null) {
            System.out.println("Error creating bucket!\n");
        } else {
            System.out.println("Done!\n");
        }
    }
}
```

UploadIndex.java

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;



import java.io.File;


public class UploadIndex {
//    https://www.codejava.net/aws/upload-file-to-s3-java-console
```

```
    public void uploadfile(){
        CreateBucket createBucket = new CreateBucket();

        final AmazonS3 s3 = AmazonS3ClientBuilder.standard()
                .withRegion(Regions.DEFAULT_REGION)
                .withCredentials(new AWSStaticCredentialsProvider(new
BasicSessionCredentials(
                        "ASIAS44XD3BR76XC7G4Q",
                        "LsNDVqZeLo07YX4oNdzMFdPvniluKStSO5AcPjwF",

"FwoGZXIvYXdzED4aDEZJW3l1fZTHZWy2BSLAARz5s62574lTWroOjeeHpywYRmoGRcRp6PLPWLSI
xG8MKX+df+yaY+SapQKNtzg8a8ZeIqyLL8jyGRC4c34ZaKlx7EopLbZ17s1om5hoPc+kYg1G+n03M
S759AlQDpp3vDIzhZ5HmW6nsCTOOuAKzYhdUj7a4RZ4ea6uqJtYwhkI4MRQQabDoMqwTi4Nl4kKBp
zXlUHEb/5IDBz7wcsLDQ/LUV5pT4iUjF5ZwVJmHZc01ouDBbAZnny6l10d+bT3uijr1bGaBjItUjW
Lg1l1eYP2Fwv7f7byrzTwW3fpoFKYJeS4BHHbQChOCK3yXIzuqQIEBaP5"
                ))).build();

        try{
            System.out.println("File is being uploaded");
            s3.putObject("freezingfire", "index.html", new
File("D:\\Sagar\\SEM-2 (SEP)\\Serverless Data Processing\\Assignment
1\\CodePart2\\src\\main\\java\\index.html"));

        } catch(AmazonServiceException e){
            System.err.println(e.getErrorMessage());
            System.exit(1);
        }

    }


}
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title> INTRODUCTION </title>
    <p><i><b>Name </b> : Sagarkumar Vaghasia </i></p>
    <p><i><b> Banner Id </b>  : B00878629 </i></p>
    <p><i><b>Email</b>  : sg682034@dal.ca </i></p>
    <p><i><b>Declaration</b>  : This assignment is my own work; I did not
take help from anyone </i></p>
</head>
<body>
```

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>CodePart2</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>17</maven.compiler.source>
        <maven.compiler.target>17</maven.compiler.target>
    </properties>

    https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
    <dependencyManagement>
        <dependencies>
            <dependency>
                <groupId>com.amazonaws</groupId>
                <artifactId>aws-java-sdk-bom</artifactId>
                <version>1.11.837</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>com.amazonaws</groupId>
            <artifactId>aws-java-sdk-s3</artifactId>
        </dependency>
        <dependency>
            <groupId>com.amazonaws</groupId>
            <artifactId>aws-java-sdk-sts</artifactId>
        </dependency>
        <dependency>
            <groupId>com.amazonaws</groupId>
            <artifactId>aws-java-sdk-iam</artifactId>
        </dependency>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.bouncycastle</groupId>
            <artifactId>bcprov-jdk16</artifactId>
            <version>1.45</version>
        </dependency>
        <dependency>
            <groupId>com.amazonaws</groupId>
            <artifactId>aws-java-sdk-kms</artifactId>
        </dependency>
    </dependencies>

</project>
```

References :

[1]Amazon.com. [Online]. Available: https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html. [Accessed: 12-Oct-2022].

[2]N. H. Minh, “Upload file to S3 using AWS Java SDK - Java console program,” Codejava.net. [Online]. Available: https://www.codejava.net/aws/upload-file-to-s3-java-console. [Accessed: 12-Oct-2022].

[3]Amazon.com. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteHosting.html. [Accessed: 12-Oct-2022].

[4]Amazon.com. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteEndpoints.html. [Accessed: 12-Oct-2022].

[5]Amazon.com. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/EnableWebsiteHosting.html. [Accessed: 12-Oct-2022].

[6]Amazon.com. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteAccessPermissionsReqd.html. [Accessed: 12-Oct-2022].