Individual Cloud Computing Journey Week 2 - Applied cloud computing

Aniketh Satyanarayana

For week 2 in the individual cloud commuting journey, I focused on creating a web-application that can be hosted and has a domain name.
I explored quite a few avenues which would help me lead to this destination.
They were:

Using aws ecosystem - and particularly Amplify within it.

Streamlit cloud - so streamlit is a wonderful and easy-to-use front-end service that was particularly developed for presenting data science applications beautifully without much effort on the HTML ,JS, CSS trio or any other equivalent framework but rather primarily focus well on building the data science app itself and getting it working.

I created an application using python and streamlit that helps users know the ticket fare on a hypothetical commute service called A-Train
(inspired from Caltrain, which I enjoy traveling in and the popular Prime Video series Boys, where some of the character references in this application come from!)

The functions performed by this application are:

1)Ask the user for the entry and the exit stations (from a drop-down list of stations that are actually on the Caltrain network itself.)

2)After the user makes choices, the ticket fare for the journey is displayed in $.

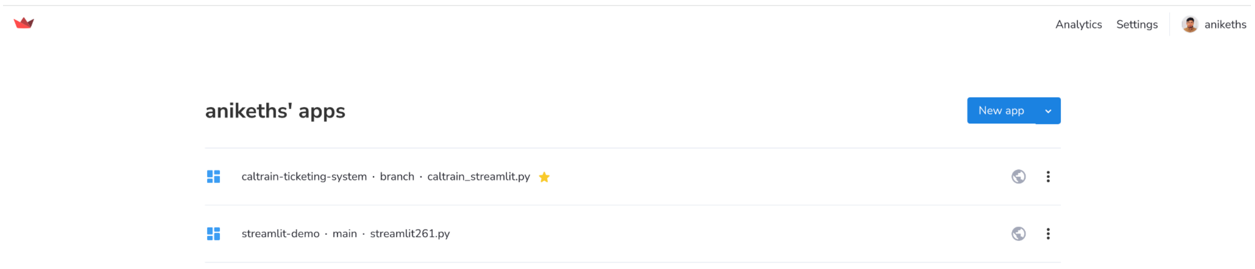
3)The user can review this and decide to book the ticket by making the payment.However this is just a simulation and does not include the UI for the payment yet.

4)There is a download button used here. After hitting this button, the ticket gets downloaded in a .txt file on the local machine of the user.

This file contains the actual fare of the ticket , the entry station and the exit station selected by the user in the step above.

This app is hosted on my Streamlit cloud account and the good thing about it is it comes with a URL that can be shared,

So here's the URL to that.
https://anikeths-caltrain-ticketing-system-caltrain-streamlit-br-p2yink.streamlitapp.com/



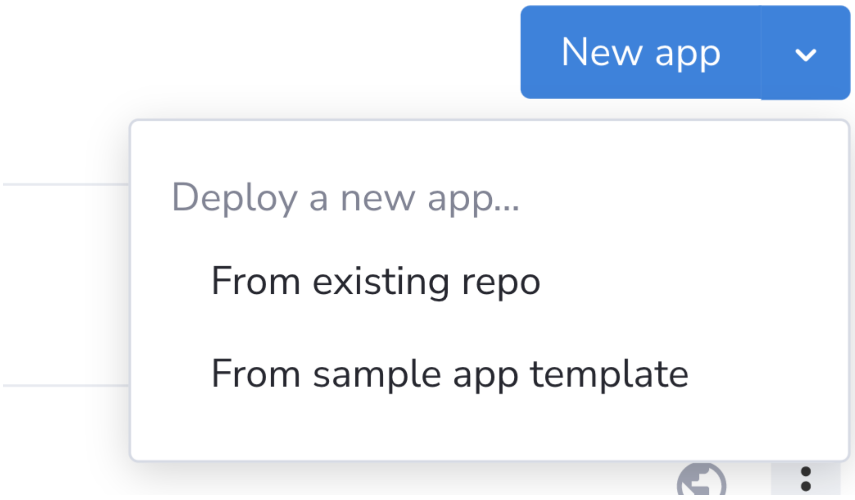
caltrain-ticketing-system · branch · caltrain_streamlit.py

streamlit-demo · main · streamlit261.py

The steps involved were as follows :

1)Write a streamlit compatible code and store it in .py format.

2)Uploaded the code to Github.
https://github.com/aniketh s/Caltrain-Ticketing-System/tree/branch



← Back

Deploy an app

Repository Paste GitHub URL

aniketh/Caltrain-Ticketing-System

Branch

main

Main file path

caltrain_streamlit.py

Advanced settings...

Deploy!

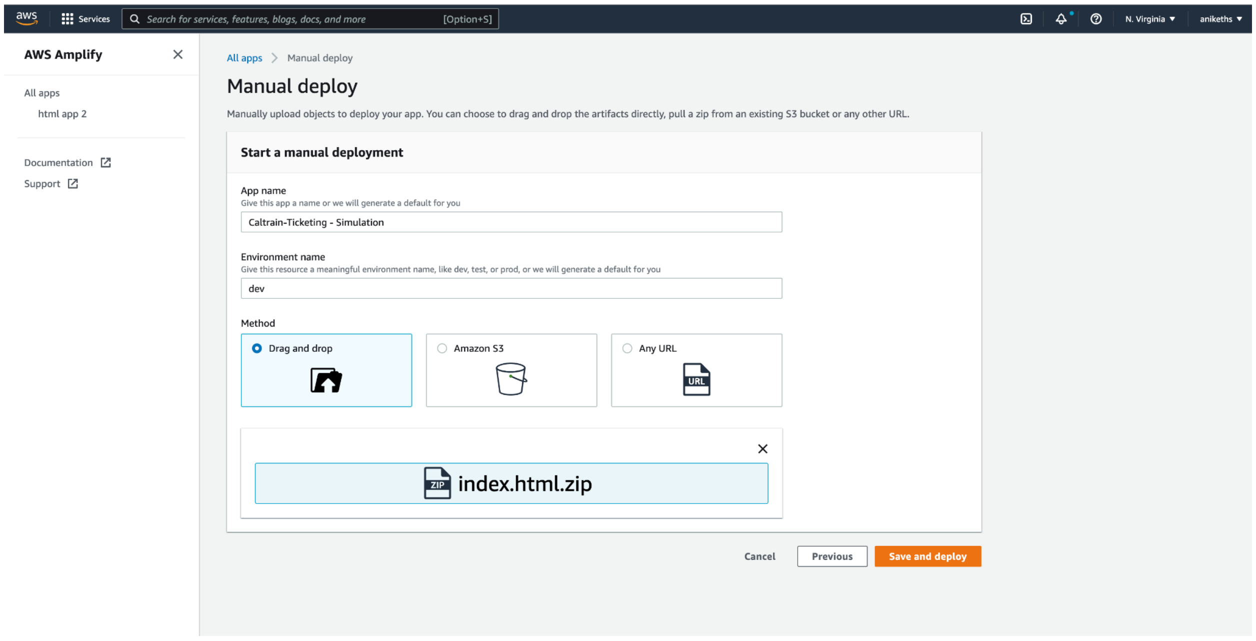
🍞

Your app is in the oven

Alternatively,I can use the AWS Amplify service to host a static website.
Goto AWS Amplify and set it up as follows:

1)Choose manual deployment- Name the app- give an environment name
The environments to this can either be:Dev - development Prod - production Test- test
2)I chose drag-and-drop and then uploaded the .html file in .html.zip format.
(The streamlit cloud app url is embedded within a html iframe src tag)

3)The app gets deployed and a shareable https link is generated.



Manual deploy

Manually upload objects to deploy your app. You can choose to drag and drop the artifacts directly, pull a zip from an existing S3 bucket or any other URL.

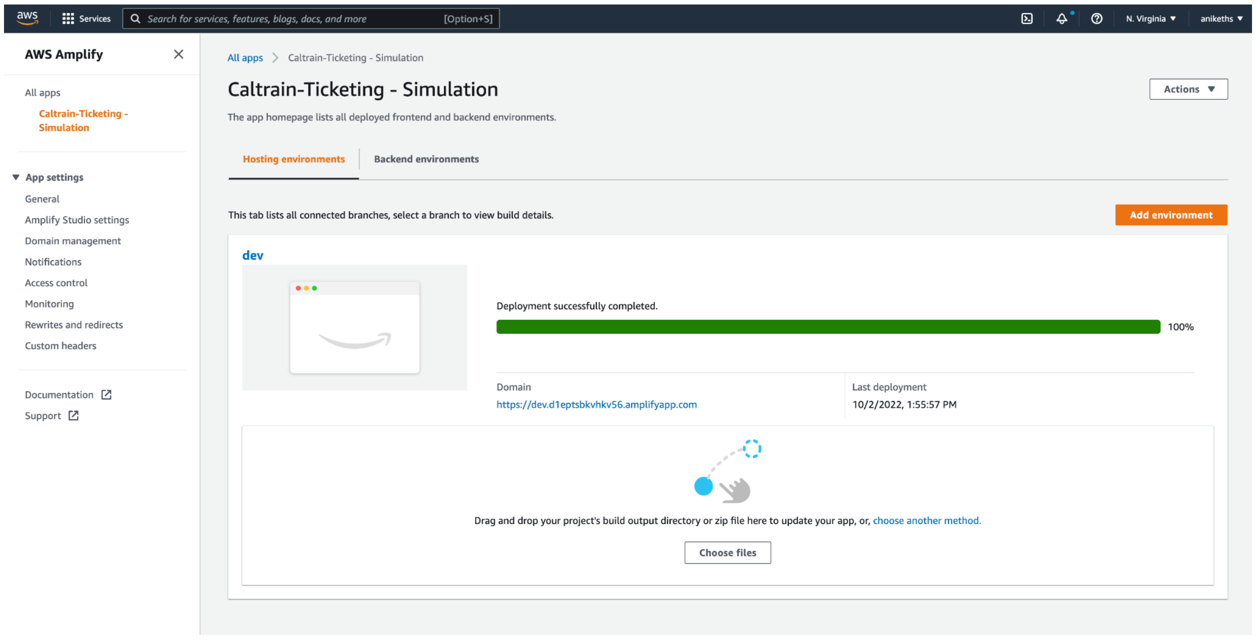
Start a manual deployment

App name
Give this app a name or we will generate a default for you
Caltrain-Ticketing - Simulation

Environment name
Give this resource a meaningful environment name, like dev, test, or prod, or we will generate a default for you
dev

Method
[✓ Drag and drop] Amazon S3 Any URL

index.html.zip


AWS Amplify

All apps

Caltrain-Ticketing - Simulation

App settings
- General
- Amplify Studio settings
- Domain management
- Notifications
- Access control
- Monitoring
- Rewrites and redirects
- Custom headers

Documentation

Support

All apps > Caltrain-Ticketing - Simulation

Caltrain-Ticketing - Simulation

The app homepage lists all deployed frontend and backend environments.

Hosting environments | Backend environments

This tab lists all connected branches, select a branch to view build details.

Add environment

dev

Deployment successfully completed. 100%

Domain: https://dev.d1eptsbkvhkv56.amplifyapp.com

Last deployment: 10/2/2022, 1:55:57 PM

Drag and drop your project's build output directory or zip file here to update your app, or, choose another method.

Choose files

One can always upload the updated .zip file

Here's the link to that
https://dev.d1eptsbkvhkv56.amplifyapp.com

It is rendering poorly due to the styling issues which will be surely taken care of.

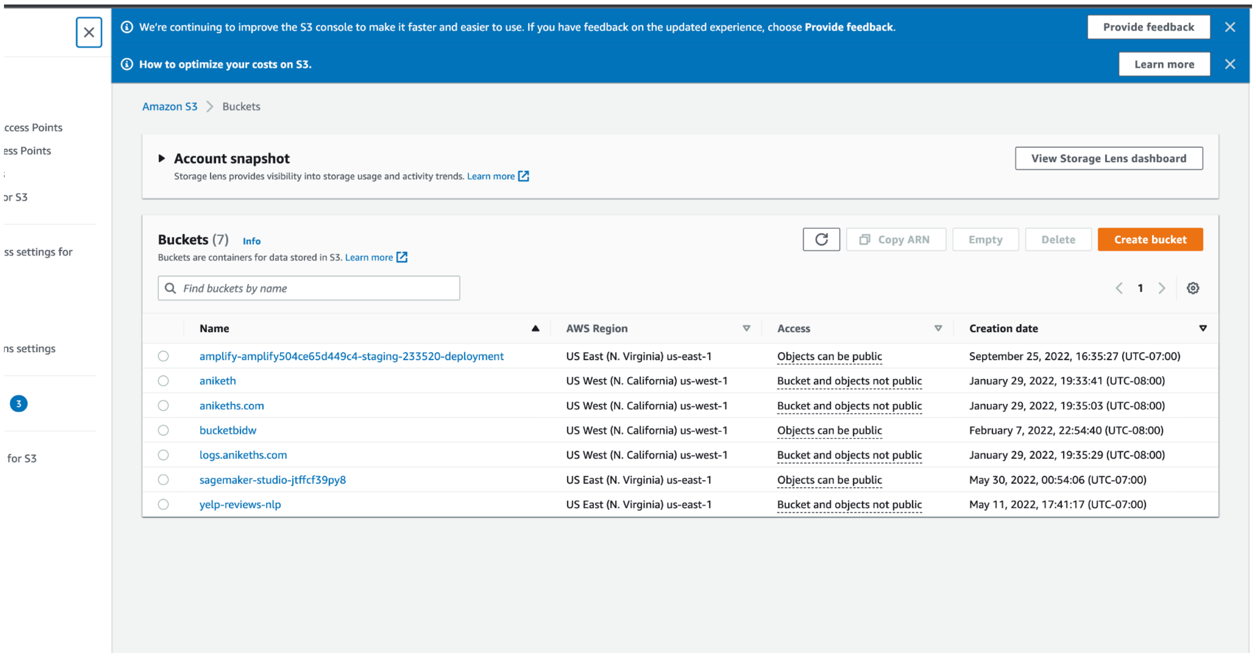
The other work that I did for this week:

Trying to connect streamlit to AWS S3. referring to streamlit docs.
https://docs.streamlit.io/knowledge-base/tutorials/databases/aws-s3

I also explored the possibility of writing files to S3 bucket.

Some configuration had to be done for that.
I referred to the documentation online.

Creating an S3 bucket -



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**.

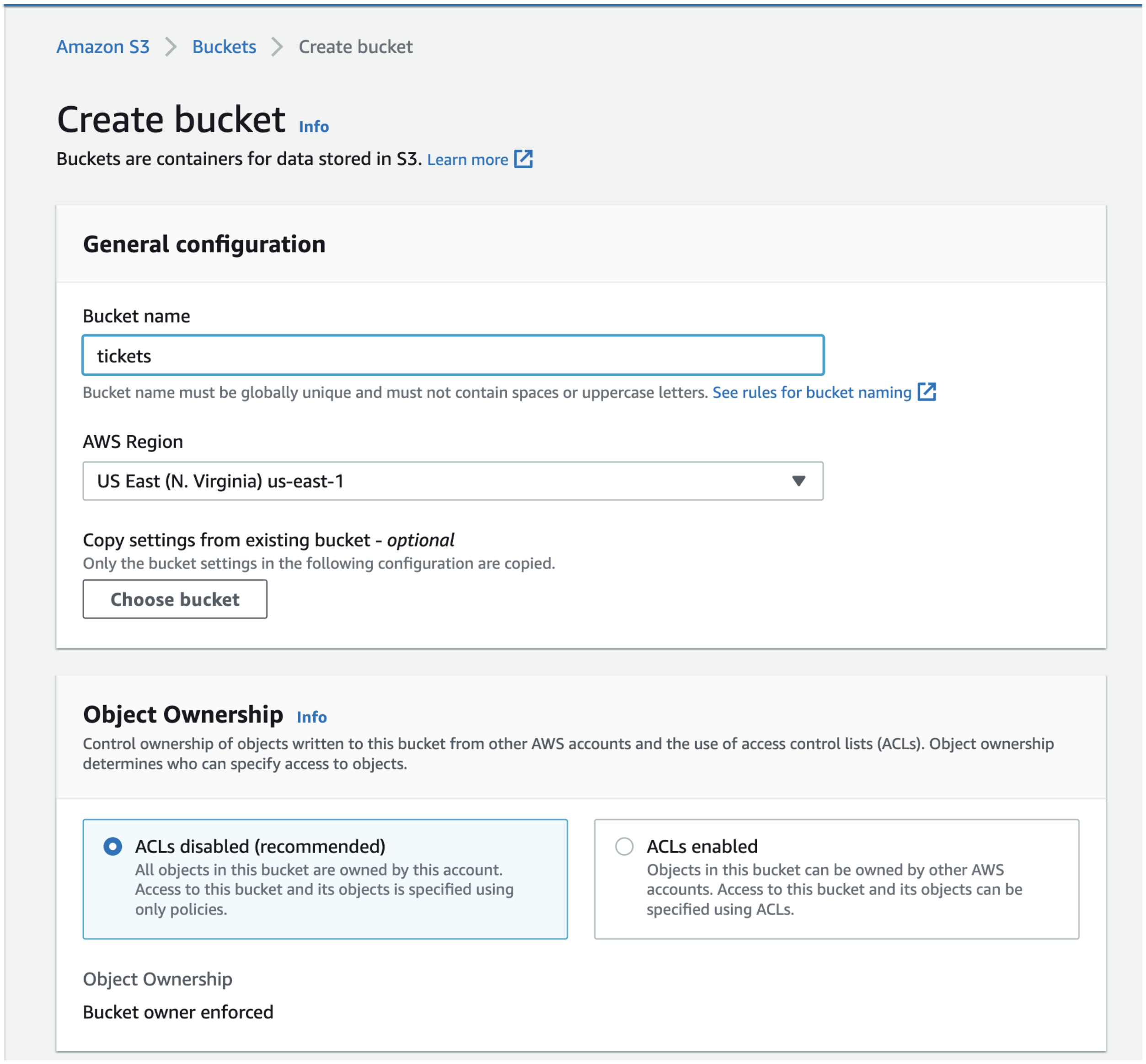
How to optimize your costs on S3.

Amazon S3 > Buckets

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

Buckets (7) Info
Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
amplify-amplify504ce65d449c4-staging-233520-deployment	US East (N. Virginia) us-east-1	Objects can be public	September 25, 2022, 16:35:27 (UTC-07:00)
aniketh	US West (N. California) us-west-1	Bucket and objects not public	January 29, 2022, 19:33:41 (UTC-08:00)
aniketh.com	US West (N. California) us-west-1	Bucket and objects not public	January 29, 2022, 19:35:03 (UTC-08:00)
bucketbidw	US West (N. California) us-west-1	Objects can be public	February 7, 2022, 22:54:40 (UTC-08:00)
logs.aniketh.com	US West (N. California) us-west-1	Bucket and objects not public	January 29, 2022, 19:35:29 (UTC-08:00)
sagemaker-studio-jtffcf39py8	US East (N. Virginia) us-east-1	Objects can be public	May 30, 2022, 00:54:06 (UTC-07:00)
yelp-reviews-nlp	US East (N. Virginia) us-east-1	Bucket and objects not public	May 11, 2022, 17:41:17 (UTC-07:00)


Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name

tickets

Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership Info

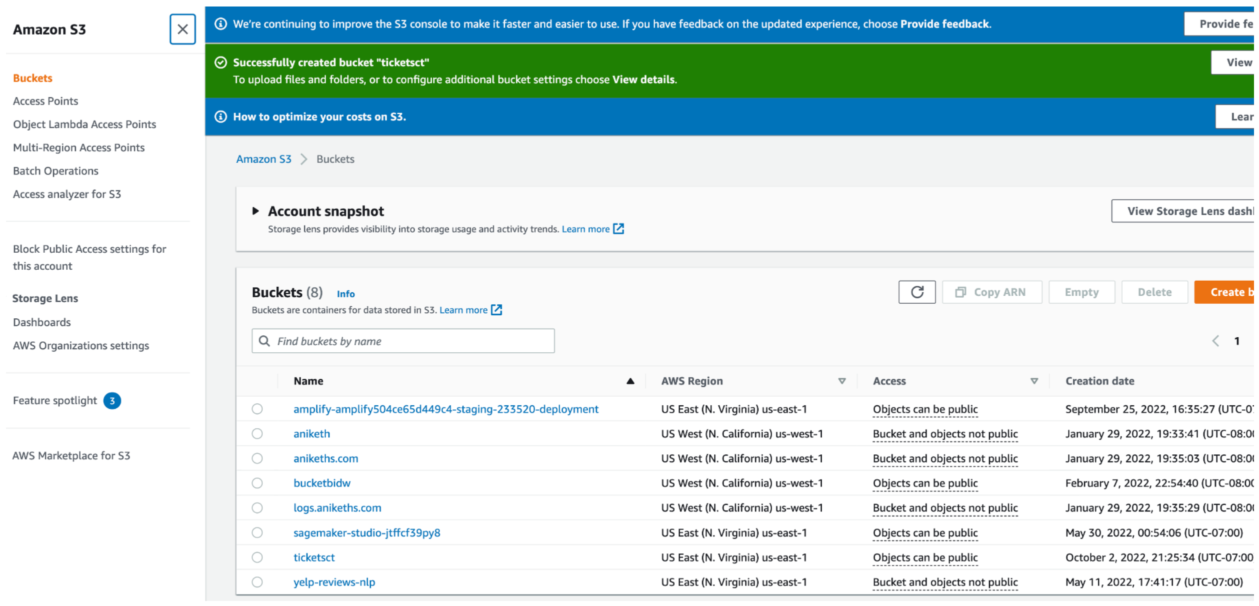
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced



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

AWS Marketplace for S3

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**. Provide fe

Successfully created bucket "ticketsct"
To upload files and folders, or to configure additional bucket settings choose **View details**. View

How to optimize your costs on S3. Lear

Amazon S3 > Buckets

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

View Storage Lens dashl

Buckets (8) Info
Buckets are containers for data stored in S3. Learn more

Copy ARN | Empty | Delete | Create b

Find buckets by name

Name	AWS Region	Access	Creation date
amplify-amplify504ce65d449c4-staging-233520-deployment	US East (N. Virginia) us-east-1	Objects can be public	September 25, 2022, 16:35:27 (UTC-0
aniketh	US West (N. California) us-west-1	Bucket and objects not public	January 29, 2022, 19:33:41 (UTC-08:0
aniketths.com	US West (N. California) us-west-1	Bucket and objects not public	January 29, 2022, 19:35:03 (UTC-08:0
bucketbidw	US West (N. California) us-west-1	Objects can be public	February 7, 2022, 22:54:40 (UTC-08:0
logs.aniketths.com	US West (N. California) us-west-1	Bucket and objects not public	January 29, 2022, 19:35:29 (UTC-08:0
sagemaker-studio-jtffcf39py8	US East (N. Virginia) us-east-1	Objects can be public	May 30, 2022, 00:54:06 (UTC-07:00)
ticketsct	US East (N. Virginia) us-east-1	Objects can be public	October 2, 2022, 21:25:34 (UTC-07:00
yelp-reviews-nlp	US East (N. Virginia) us-east-1	Bucket and objects not public	May 11, 2022, 17:41:17 (UTC-07:00)

```
import boto3

#Creating Session With Boto3.
session = boto3.Session(
aws_access_key_id='<your_access_key_id>',
aws_secret_access_key='<your_secret_access_key>'
)

#Creating S3 Resource From the Session.
s3 = session.resource('s3')

object = s3.Object('<bucket_name>', 'file_name.txt')

txt_data = b'This is the content of the file uploaded from python boto3'

result = object.put(Body=txt_data)

res = result.get('ResponseMetadata')

if res.get('HTTPStatusCode') == 200:
    print('File Uploaded Successfully')
else:
    print('File Not Uploaded')
```

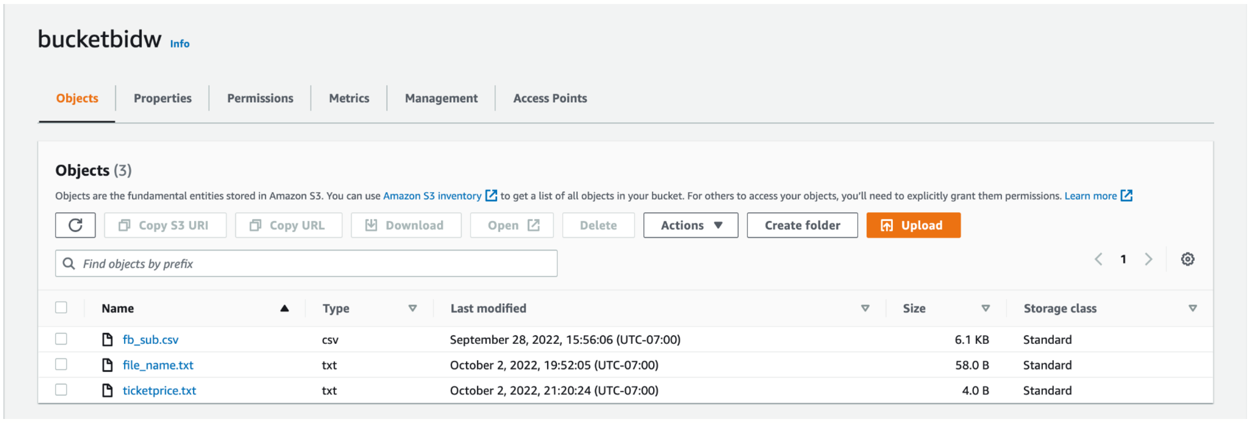
Code part to configure and interact with the cloud:

- Imports s3fs, boto3
- AWS secret credentials

```
import s3fs
import boto3

fs=s3fs.S3FileSystem(anon = False)

#Creating Session With Boto3.
session = boto3.Session(
aws_access_key_id='AKIA2BCQHJ4O3FBPMGYP',
aws_secret_access_key='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'
)

#Creating S3 Resource From the Session.
s3 = session.resource('s3')
object = s3.Object('bucketbidw', 'ticketprice.txt')

result = object.put(Body=bticket_price)
```

After running this code in the colab notebook, a connection was established between the AWS S3 bucket and the colab client.

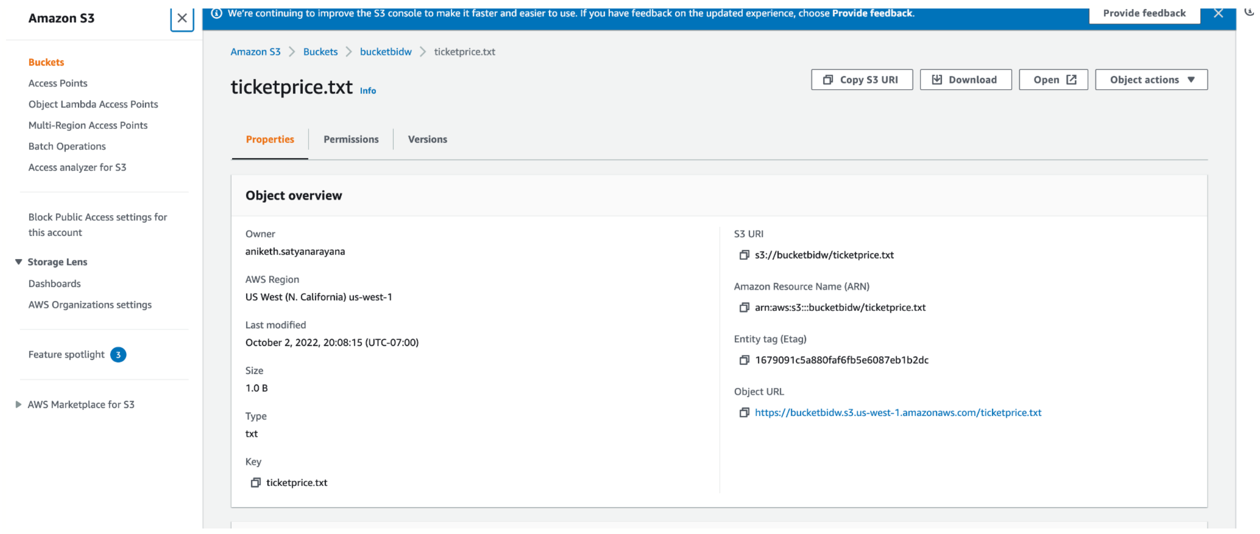
And ultimately and effectively, the ticketprice.txt file was written into the bucket.



Objects (3)

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
fb_sub.csv	csv	September 28, 2022, 15:56:06 (UTC-07:00)	6.1 KB	Standard
file_name.txt	txt	October 2, 2022, 19:52:05 (UTC-07:00)	58.0 B	Standard
ticketprice.txt	txt	October 2, 2022, 21:20:24 (UTC-07:00)	4.0 B	Standard



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**.

ticketprice.txt Info

Object overview

Owner	aniketh.satyanarayana
AWS Region	US West (N. California) us-west-1
Last modified	October 2, 2022, 20:08:15 (UTC-07:00)
Size	1.0 B
Type	txt
Key	ticketprice.txt
S3 URI	s3://bucketbidw/ticketprice.txt
Amazon Resource Name (ARN)	arn:aws:s3:::bucketbidw/ticketprice.txt
Entity tag (Etag)	1679091c5a880faf6fb5e6087eb1b2dc
Object URL	https://bucketbidw.s3.us-west-1.amazonaws.com/ticketprice.txt

Plan for the weeks ahead:

- I intend to use AWS Glue for ETL and AWS Sagemaker and also integrate an AWS chatbot with this application.

- The data entered by the customer poses a huge opportunity for further analysis here and help make useful decisions out of it.

- Some simple tasks like dropping certain columns, renaming the columns etc can be performed on the AWS CLOUD PLATFORM AND the results can be stored on the AWSs3 bucket.

- To check whether the file is there on the s3 bucket, one could easily write a few lines of code.
- Usage of a smtp server to email the ticket to the user is also another exciting feature that can possibly be added to this app.

References:

Blogs online and stackoverflow.