

Comm Network Mgt (CIS 5659)

Launch a Static Website on Amazon S3

Sai Madhuhas Kotini

University of Central Missouri

Dr. Shah M A Mumin

Web-Page Project:

Book a ride to the university is a user-friendly platform to book a comfortable ride at your convenience to the university. Its goal is to provide safe, comfortable, and affordable transportation for students.

Requirements:

The webpage consists of HTML Forms and CSS.

The HTML Forms has the following

- User Information: It has name of the students.
- Pick-up and Drop-off Location details.
- Date and Time for the ride requested.
- A button for “Book a Ride” for confirmation of the ride requested.
- A webpage with ride details should display along with “Download confirmation”.
- Upon clicking on the button “Download Confirmation” a text file with user ride details and Confirmation Number should download.

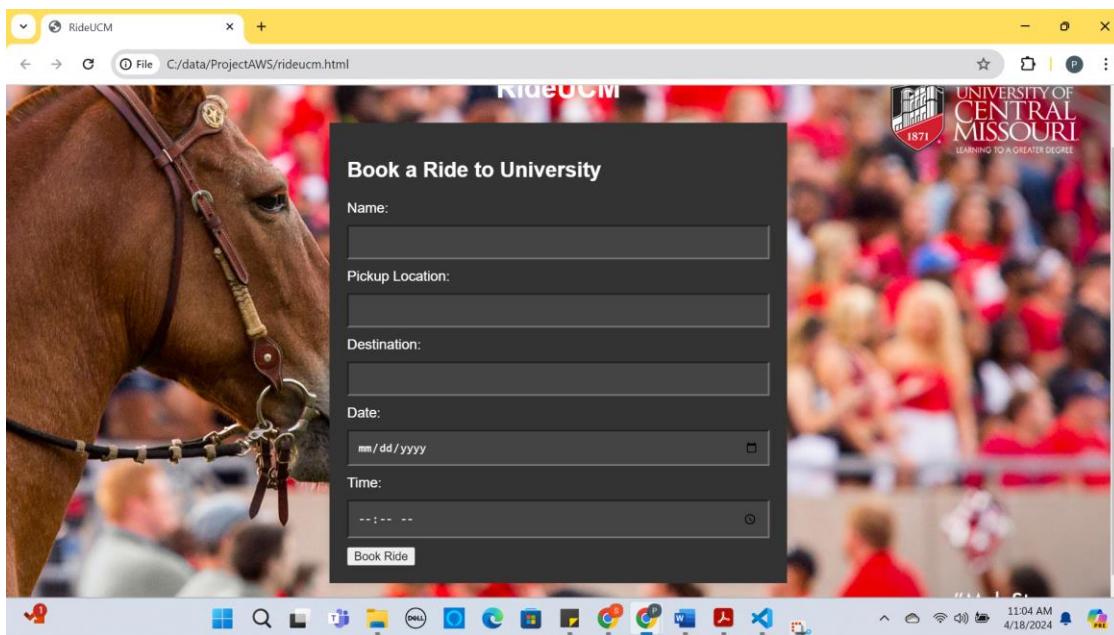
Functionality:

When the user enters all the details on the webpage and clicks on the submit button(Book a ride). The ride details will be displayed on a different webpage with a download confirmation button.

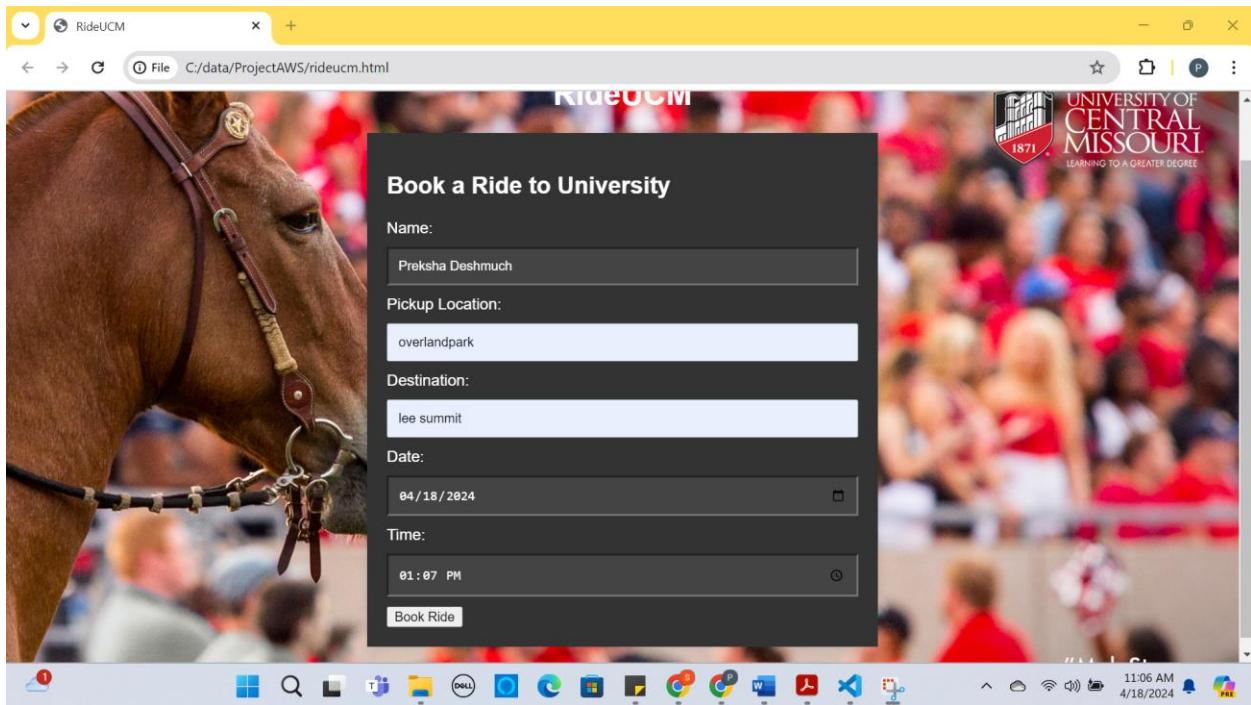
Once the user clicks on the download confirmation button a text file with the ride details will download to the user's device.

Operationalization with screenshots:

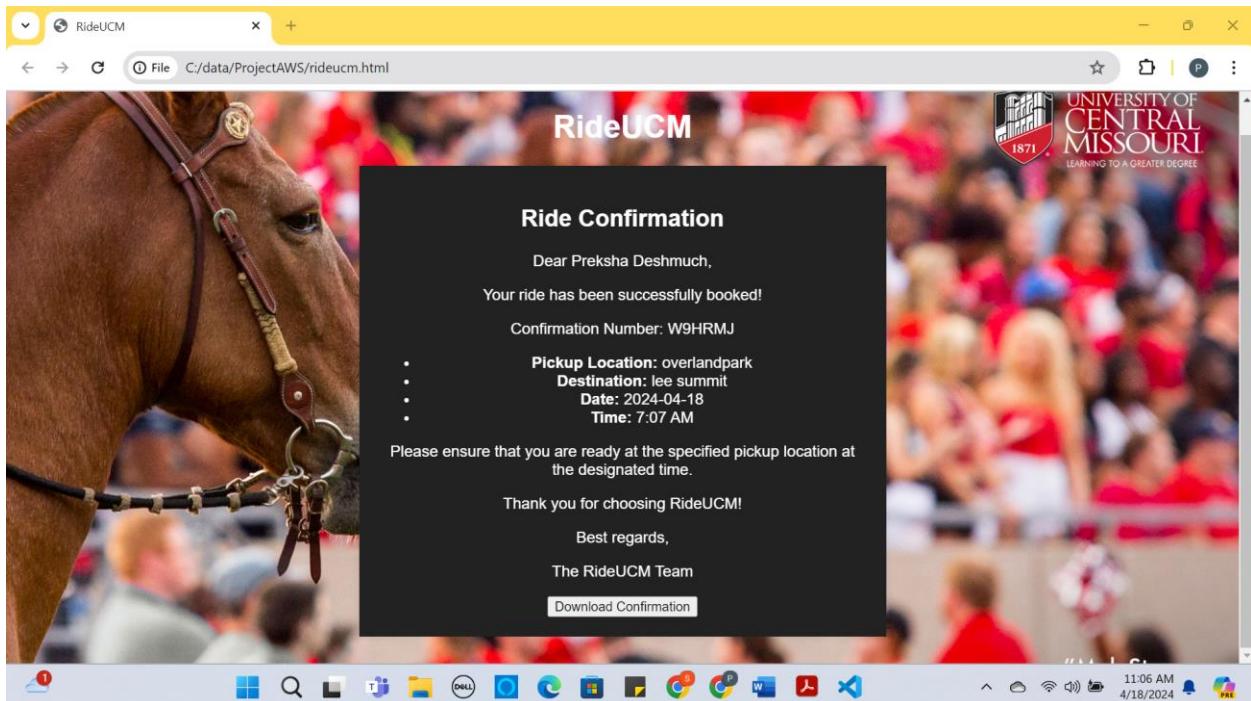
Webpage:



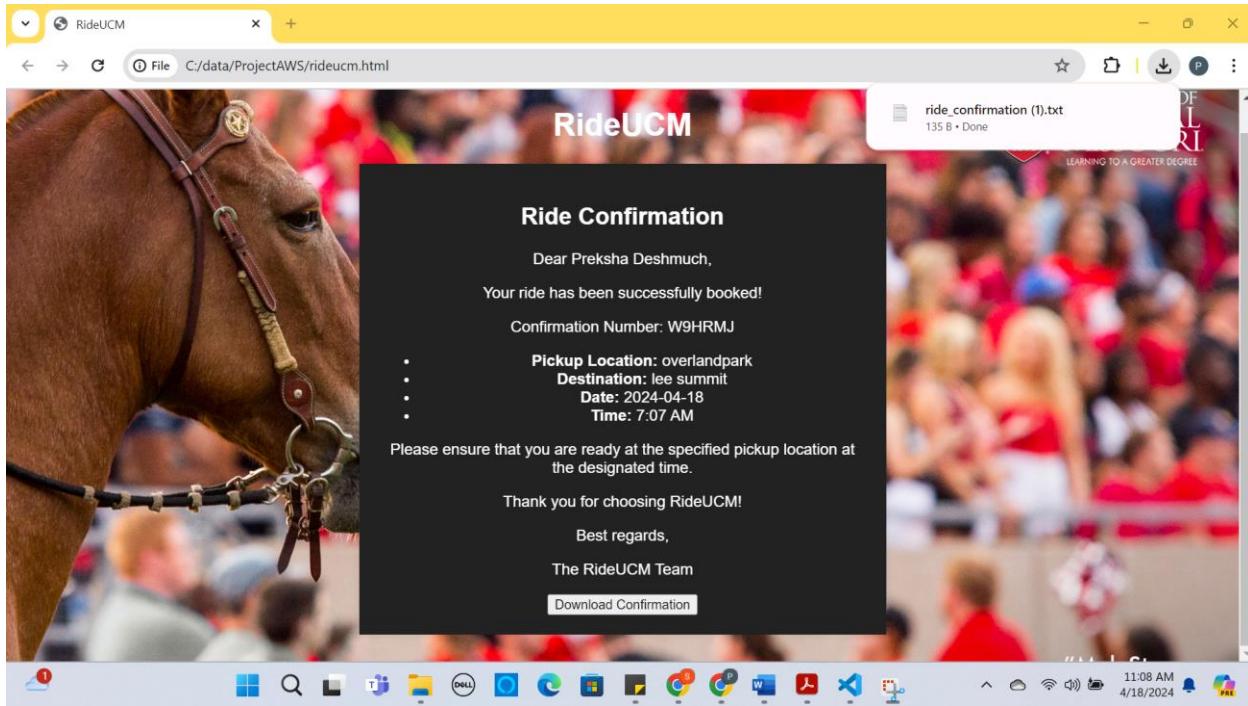
Enter the student details:



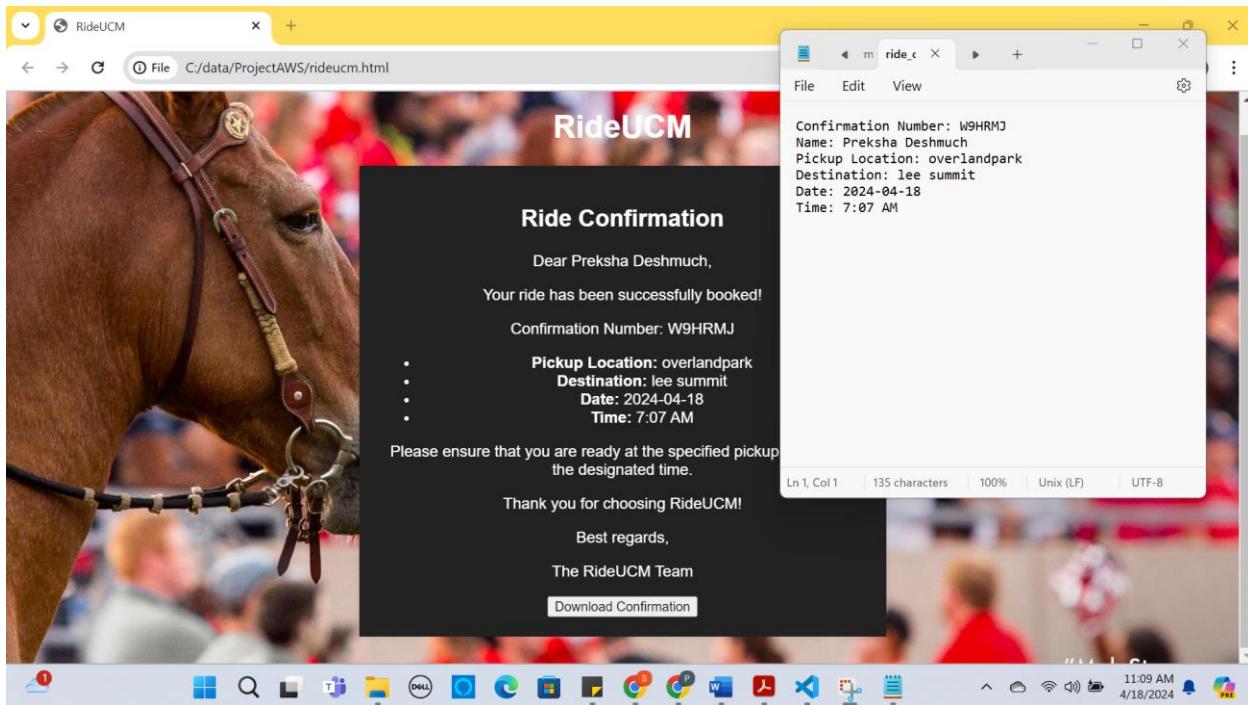
Click on the Book Ride Button:



Click on the Download Confirmation:



Text File after clicking Download Confirmation



AWS Services used:

S3

CloudFront

Route 53

CloudWatch

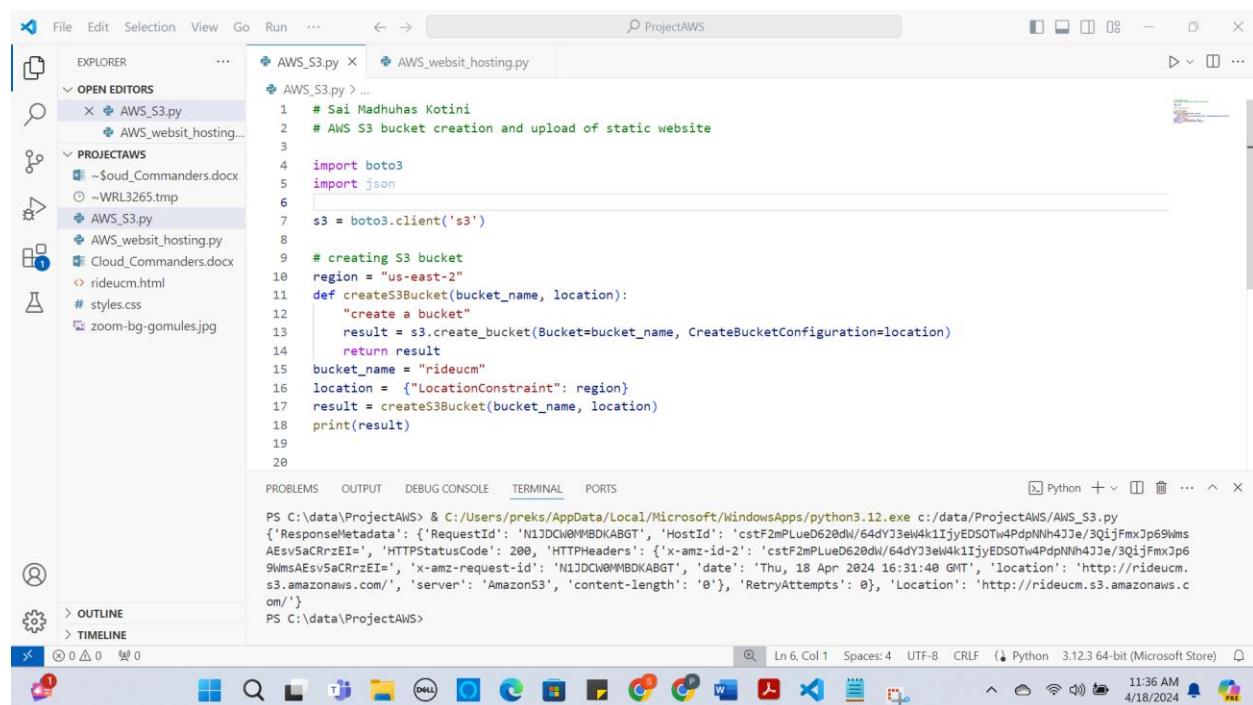
Certificate Manager

Launching a Static website in Amazon S3 Bucket:

IAM User: preksha

Account Number: 381491822006

Step 1: Create a S3 Bucket name “rideucm” using Python Boto3.



The screenshot shows the Microsoft Visual Studio Code interface. The Explorer sidebar on the left lists files in the 'PROJECTAWS' folder, including 'AWS_S3.py', 'AWS_websit_hosting.py', and 'Cloud_Commanders.docx'. The 'OPEN EDITORS' section shows two files: 'AWS_S3.py' and 'AWS_websit_hosting.py'. The 'AWS_S3.py' editor contains Python code for creating an S3 bucket. The 'TERMINAL' tab at the bottom shows the output of a Python command to run the script, resulting in a successful creation of the 'rideucm' bucket in the 'us-east-2' region.

```
# Sai Madhuhas Kotini
# AWS S3 bucket creation and upload of static website

import boto3
import json

s3 = boto3.client('s3')

# creating S3 bucket
region = "us-east-2"
def createS3Bucket(bucket_name, location):
    "create a bucket"
    result = s3.create_bucket(Bucket=bucket_name, CreateBucketConfiguration=location)
    return result

bucket_name = "rideucm"
location = {"LocationConstraint": region}
result = createS3Bucket(bucket_name, location)
print(result)
```

```
PS C:\data\ProjectAWS> & C:/Users/preks/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/data/ProjectAWS/AWS_S3.py
{'ResponseMetadata': {'RequestId': 'N1JDCW0MBDKABGT', 'HostId': 'cstF2mPlueD620dW/64dyJ3eW4k1IjyEDSOTw4PdpNNh4JJe/3QijFmxJp69WmsA
Esv5aCrzEI=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': 'cstF2mPlueD620dW/64dyJ3eW4k1IjyEDSOTw4PdpNNh4JJe/3QijFmxJp69WmsA
Esv5aCrzEI', 'x-amz-request-id': 'N1JDCW0MBDKABGT', 'date': 'Thu, 18 Apr 2024 16:31:40 GMT', 'location': 'http://rideucm.s3.amazonaws.com/'}, 'server': 'AmazonS3', 'content-length': '0'}, 'RetryAttempts': 0}, 'Location': 'http://rideucm.s3.amazonaws.com/'}
PS C:\data\ProjectAWS>
```

The screenshot shows the AWS S3 Bucket dashboard. On the left, a sidebar lists options like Buckets, Access Grants, and Storage Lens. The main area displays the 'rideucm' bucket under 'General purpose buckets'. Key details shown include:

- Average object size: 49.0 B
- Bucket Type: General purpose buckets (1)
- Name: rideucm
- AWS Region: US East (Ohio) us-east-2
- IAM Access Analyzer: View analyzer for us-east-2
- Creation date: April 18, 2024, 11:31:40 (UTC-05:00)

The dashboard also features tabs for 'General purpose buckets' and 'Directory buckets', along with buttons for 'Create bucket' and 'Empty'.

Step 2: Enable Static Website Hosting through AWS console

- Login to the AWS console with IAM user and password.
- Goto the S3 Bucket dashboard.
- Click on the S3 bucket "rideucm" which is created using python boto3.
- Choose Properties.

The screenshot shows the 'Properties' tab of the 'rideucm' bucket's properties page. The sidebar and top navigation bar are identical to the previous screenshot. The main content area includes:

- Bucket overview:**
 - AWS Region: US East (Ohio) us-east-2
 - Amazon Resource Name (ARN): arn:aws:s3:::rideucm
 - Creation date: April 18, 2024, 11:31:40 (UTC-05:00)
- Bucket Versioning:** A section explaining the concept of bucket versioning and a link to learn more.

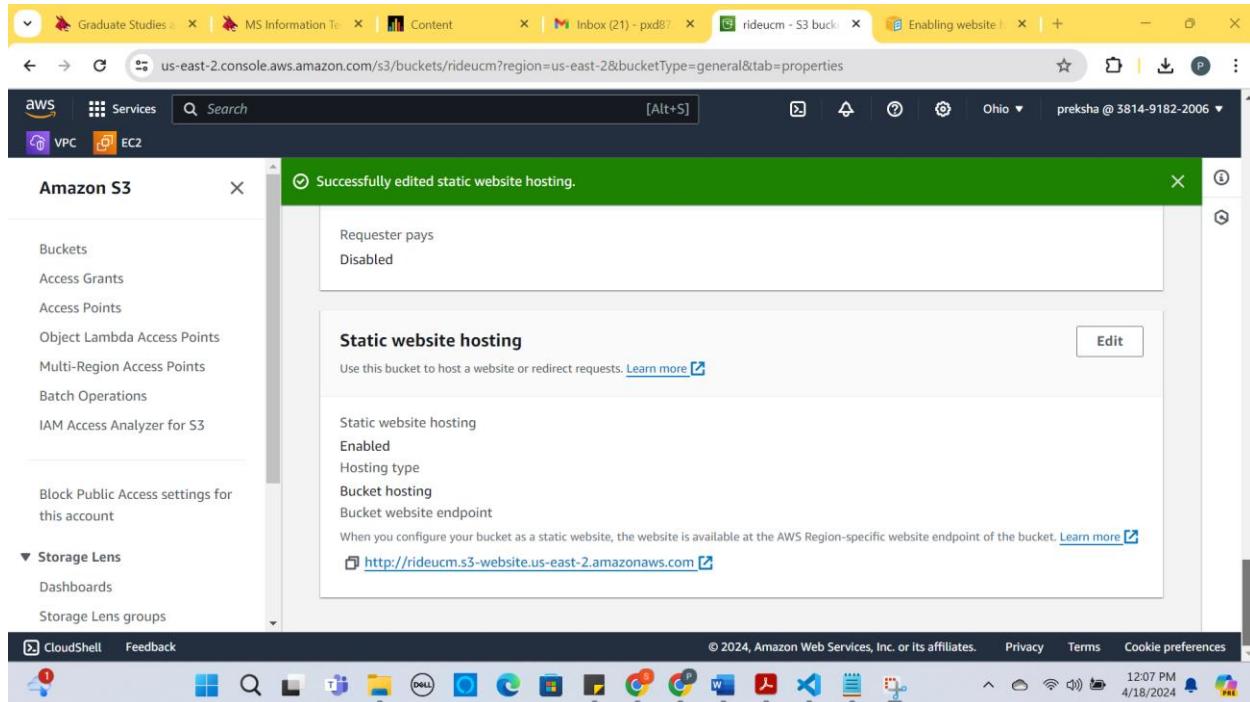
- In the Bucket overview under static website hosting click enable.

The screenshot shows the AWS S3 Bucket Properties page for the bucket 'rideucm'. In the 'Static website hosting' section, there is a link to 'Edit' which is highlighted in yellow. Below the link, it says 'Static website hosting' and 'Disabled'.

- In the Index document, enter the file name as the webpage name “index.html”.

The screenshot shows the AWS S3 Bucket Properties page for the bucket 'rideucm'. A green success message at the top states 'Successfully edited static website hosting.' Below the message, the 'Static website hosting' section now shows 'Enabled' instead of 'Disabled'. The status message 'Static website hosting' is also listed below the link.

- Click on save changes.



- EndPoint URL:

<http://rideucm.s3-website.us-east-2.amazonaws.com>

- Enabling public access for the static website.
- Click on the permission tab for the Bucket “rideucm”
- Click on the Edit button for Block Public access settings by default it will be on.

Edit Block public access (bucket settings)

Block public access (bucket settings)

Block all public access

Block public access to buckets and objects granted through new access control lists (ACLs)

Block public access to buckets and objects granted through any access control lists (ACLs)

Successfully edited Block Public Access settings for this bucket.

- Uncheck the Block of all public access.

Permissions

Permissions overview

Block public access (bucket settings)

Block all public access

Block public access to buckets and objects granted through new access control lists (ACLs)

Block public access to buckets and objects granted through any access control lists (ACLs)

Successfully edited Block Public Access settings for this bucket.

- Set Bucket Policy.

```

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

```

Step 3: Upload the website(webpage) contents in the Se Bucket “rideucm” using Python Boto3.

```

# Sal Madhuhas Kotini
# AWS S3 bucket creation and upload of static website
import boto3
import json

# function that will upload my static website to my newly created bucket
def upload_file(file_name, bucket_name):
    try:
        s3.upload_file(file_name, bucket_name, file_name)
        print(f"Successfully uploaded {file_name} to {bucket_name}.")
        return True
    except Exception as e:
        print(f"Error uploading {file_name} to {bucket_name}: {e}")
        return False

upload_file('index.html', 'rideucm')
upload_file('style.css', 'rideucm')
upload_file('zoom-bg-gomules.jpg', 'rideucm')

# List the content of my bucket
def list_bucket_contents(bucket_name):
    response = s3.list_objects_v2(Bucket=bucket_name)
    for obj in response['Contents']:
        print(obj['Key'])

list_bucket_contents('rideucm')

```

The screenshot shows the AWS S3 console interface. At the top, there are three tabs: "UCM Single Sign-On", "rideucm - S3 bucket | S3 | us-east-2", and "(Optional) Configuring a website". The middle tab is active. Below the tabs, the AWS navigation bar includes "Services", "Search", and links for "VPC" and "EC2". The main content area is titled "Objects (3) Info". It features a toolbar with buttons for "Copy S3 URI", "Copy URL", "Download", "Open", "Delete", "Actions", "Create folder", and "Upload". A sub-toolbar below it includes "Find objects by prefix". A table lists three objects:

| | Name | Type | Last modified | Size | Storage class |
|--------------------------|---------------------|------|--------------------------------------|--------|---------------|
| <input type="checkbox"/> | index.html | html | April 18, 2024, 14:58:15 (UTC-05:00) | 5.3 KB | Standard |
| <input type="checkbox"/> | style.css | css | April 18, 2024, 14:58:15 (UTC-05:00) | 1.6 KB | Standard |
| <input type="checkbox"/> | zoom-bg-gomules.jpg | jpg | April 18, 2024, 14:58:15 (UTC-05:00) | 1.6 MB | Standard |

At the bottom of the page, there are links for "CloudShell", "Feedback", and "Cookie preferences". The status bar at the bottom right shows "2:59 PM 4/18/2024".

Step 4 : Access the static website using the endpoint url from S3

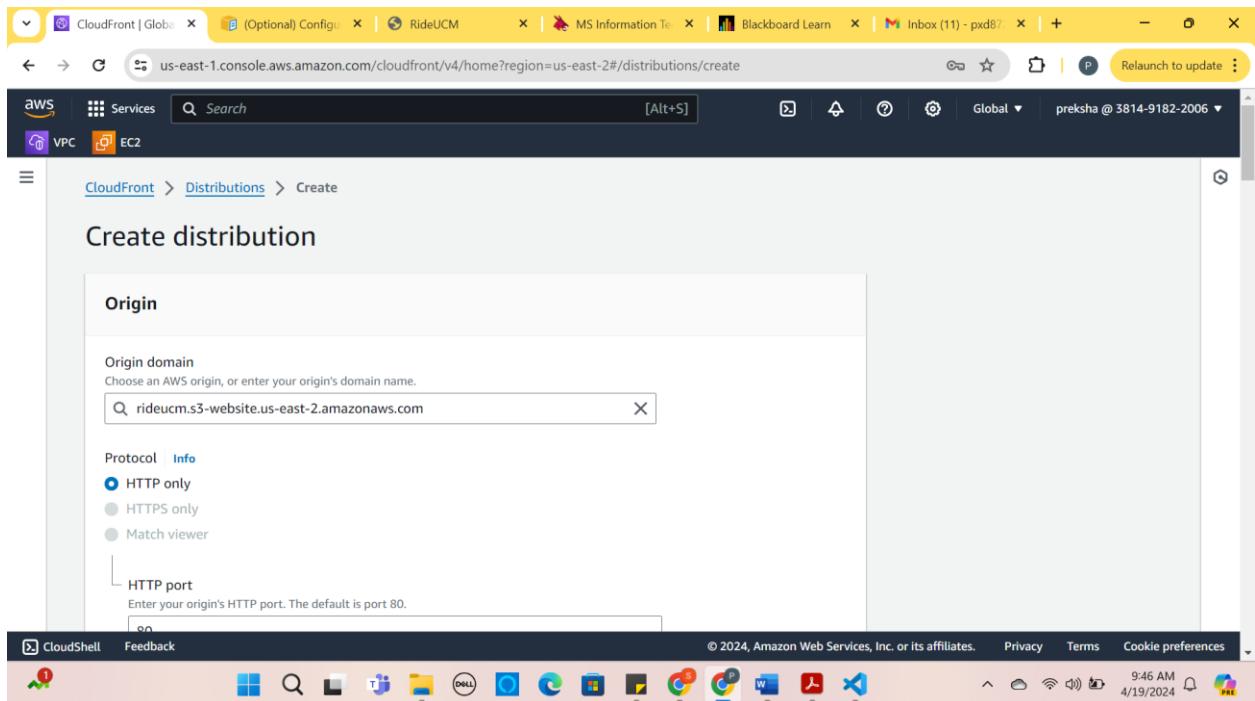
<http://rideucm.s3-website.us-east-2.amazonaws.com/>

The screenshot shows a web browser window with the address bar containing "Not secure rideucm.s3-website.us-east-2.amazonaws.com". The main content area displays a landing page for "RideUCM". The page has a dark overlay with white text. At the top, it says "Book a Ride to University". Below that are four input fields: "Name:", "Pickup Location:", "Destination:", and "Date:". The "Date:" field contains the placeholder "mm/dd/yyyy". Below the date field is a "Time:" field with a dropdown menu showing "---- --". At the bottom of the form is a "Book Ride" button. In the background, there is a blurred image of a horse's head and a crowd of people. The top right corner of the page shows the University of Central Missouri logo and the text "UNIVERSITY OF CENTRAL MISSOURI LEARNING TO A GREATER DEGREE". The status bar at the bottom right shows "3:15 PM 4/18/2024".

Using Amazon CloudFront distribution is used to launch our website globally:

How to create the distribution for the website rideucm:

- Choose the origin domain as the endpoint of the website from S3 Bucket.
- Select HTTP port 80
- The name of the origin would be the URL of the website.
- Default cache behaviour is set as it is.
- Cache key and origin requests is set to the S3.
- Click on the create distribution.



The screenshot shows the AWS CloudFront Security Dashboard. At the top, there's a blue header bar with the title "Introducing the CloudFront Security Dashboard" and a message: "The new security tab is a unified place to configure, manage, and monitor security for your CloudFront distribution. The built-in dashboard gives you visibility into top security trends, allowed and blocked traffic, as well as visibility and controls for bots. CloudFront geographic restrictions are now part of the security dashboard." Below this, a green success message says "Successfully created new distribution." The main area shows a distribution named "E2B25EXP81ZJXS". The "General" tab is selected. The "Details" section shows the distribution domain name as "d1msjmf3cws0yo.cloudfront.net" and the ARN as "arn:aws:cloudfront::381491822006:distribution/E2B25EXP81ZJXS". The "Last modified" status is "Deploying". The "Settings" section includes fields for "Description" (empty), "Alternate domain names" (empty), and "Standard logging" (set to "Off"). The "Continuous deployment" section is empty. The bottom of the screen shows the Windows taskbar with various pinned icons.

- Distributed Domain Name: <https://d1msjmf3cws0yo.cloudfront.net>

Continuous deployment sections in the CloudFront:

We can deploy or release the new content of the website at different edge locations and see it works.

The screenshot shows the AWS CloudFront distribution settings page for a distribution named 'd23hmqco28jzby.cloudfront.net'. The distribution ARN is arn:aws:cloudfront::381491822006:distribution/E16O3CQIL9LQJA and it is set to UTC. A sidebar on the left lists various CloudFront management options like Policies, Functions, and Telemetry. On the right, a 'Continuous deployment' section is highlighted, which allows users to safely test, measure, and release configuration changes to viewers. It includes a detailed description of what continuous deployment is, a 'Learn more' link, and a note about using it to safely test configuration changes.

View Metrics:

This is a distribution metrics dashboard that shows the data of the website.

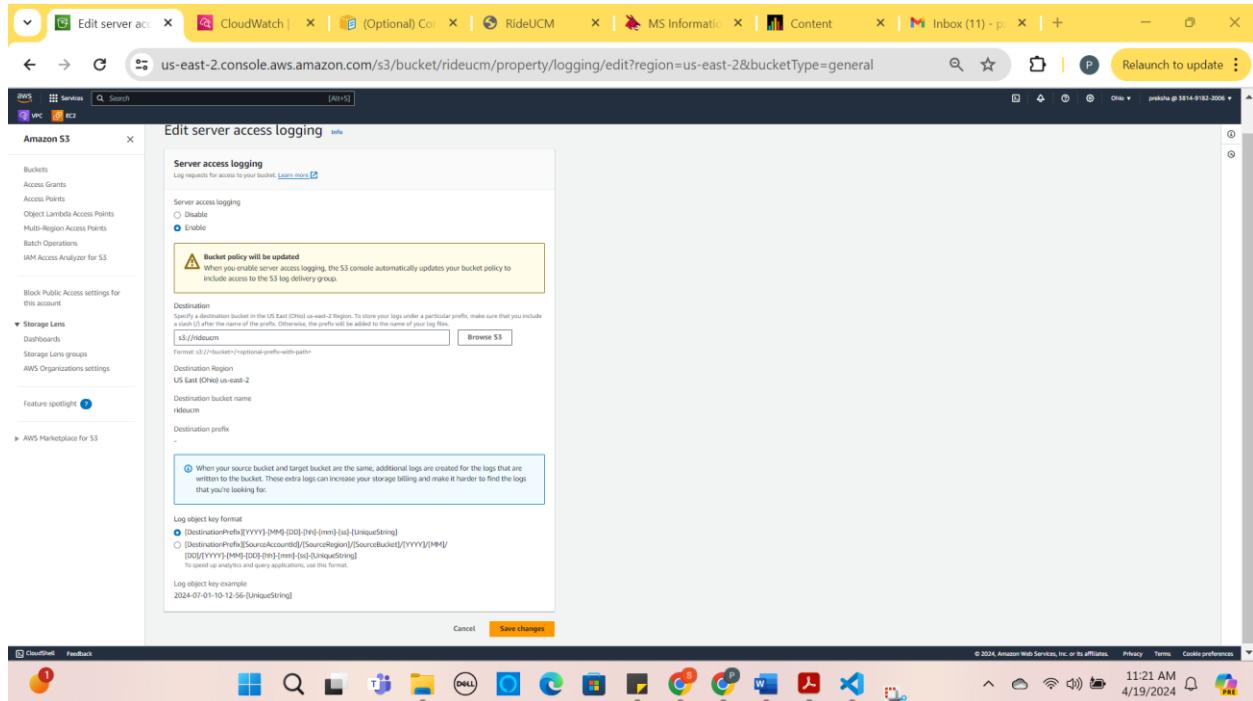
The screenshot shows the AWS CloudFront distribution metrics dashboard for the same distribution. It features two main charts: 'Requests (sum)' and 'Data transfer'. Both charts show data for the current day (13:00 to 15:00) in UTC. The 'Requests' chart shows 1 request, while the 'Data transfer' chart shows 1 byte uploaded and 0 bytes downloaded. Below these charts is a section for 'Error rate (as a percentage of total requests)', which also shows 1 error. The dashboard includes standard AWS navigation and monitoring tools at the top and bottom.

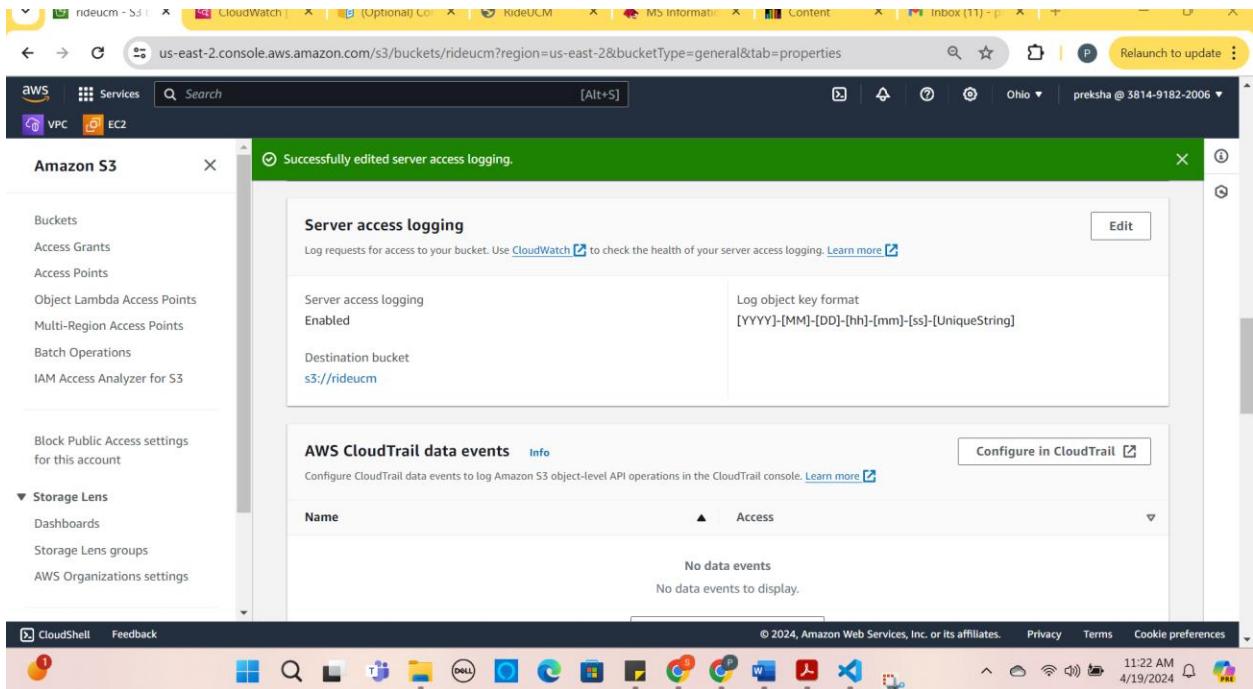
Creating Amazon CloudWatch for the website in the S3:

- Enable the server logs for the S3 Bucket.

Steps to do that:

1. Select the Bucket you want to set up Cloud watch for like rideucm.
2. Go to the properties tab navigate to the “server access logs” and click enable.
3. Select a Destination in our case that would be our Bucket
4. Click Save changes.





- Create a Cloud Watch Log Group:
 1. Navigate to the cloud watch dashboard
 2. Click on the log from the left panel.
 3. Click on Create Log Group and enter the name of the log group and finish creating it.

The screenshot shows the AWS CloudWatch Log Groups page. The URL is us-east-2.console.aws.amazon.com/cloudwatch/home?region=us-east-2#logsV2:log-groups/log-group/cloudcommanders. The left sidebar shows navigation options like Dashboards, Alarms, Logs (Log groups, Log Anomalies, Live Tail, Logs Insights), Metrics (All metrics, Explorer, Streams), X-Ray traces, and Events. The main content area displays the 'cloudcommanders' log group details. It includes sections for Log class (Info, Standard), ARN (arn:aws:logs:us-east-2:381491822006:log-group:cloudcommanders:"), Metric filters (0), Subscription filters (0), Contributor Insights rules, Anomaly detection (Configure), Data protection, Sensitive data count, Creation time (Now), Retention (Never expire), and KMS key ID. At the bottom, there are tabs for Log streams, Tags, Anomaly detection, Metric filters, Subscription filters, Contributor Insights, and Data protection. A 'Search log group' button is also present.

- **Create CloudWatch Log Streams:**

After creating the log groups now create log streams.

This screenshot is identical to the one above, showing the 'cloudcommanders' log group details. However, the 'Log streams' tab is now selected at the bottom of the page. The 'Log streams' section shows 1/1 entry, a search bar, and a checkbox for 'Log stream'. The checked item is 'cloudcommanderslogstream'. The rest of the interface is the same, including the sidebar and the top navigation bar.

- **Now Create CloudWatch Metric Filter:**

1.In the log groups “cloudcommanders” click on the actions.

2. In actions click on the create metric filters.

The screenshot shows the AWS CloudWatch Log Groups page. On the left, there's a sidebar with navigation links like Dashboards, Alarms, Logs (Log groups, Log Anomalies, Live Tail, Logs Insights), Metrics, X-Ray traces, and Events. The main area shows a log group named 'cloudcommanders'. The 'Actions' menu is open, displaying options such as Delete log group, Edit retention setting, Create metric filter, Create data protection policy, Subscription filters, Create Contributor Insights rule, Export data to Amazon S3, and View all exports to Amazon S3. The 'Subscription filters' option is highlighted with a yellow box. The top navigation bar includes the AWS logo, services like VPC and EC2, a search bar, and user information (Ohio, preksha @ 3814-9182-2006). The bottom navigation bar shows various AWS services and system status.

4. Define filter pattern.

The screenshot shows the 'Create metric filter' wizard. Step 1 is 'Define pattern'. It has three tabs: Step 1 (Define pattern), Step 2 (Assign metric), and Step 3 (Review and create). The 'Define pattern' tab is active. It contains a 'Create filter pattern' section with a note about pattern syntax, a 'Filter pattern' input field containing '[host, logName, user, timestamp, request, statusCode=4*, size]', and a 'Test pattern' section with a dropdown for 'Select log data to test' set to 'cloudcommanderslogstream' and a text area for 'Log event messages' which is currently empty. The bottom navigation bar is identical to the previous screenshot.

5. Assign Metric.

The screenshot shows the AWS CloudWatch Log Groups interface. On the left, there's a navigation sidebar with options like Dashboards, Alarms, Logs (Log groups, Log Anomalies, Live Tail, Logs Insights), Metrics (All metrics, Explorer, Streams), X-Ray traces, Events, Application Signals, Network monitoring, and Insights (Settings, Getting Started, What's new). The main area is titled "Assign metric". It has three steps: Step 1 (Define pattern), Step 2 (Assign metric), and Step 3 (Review and create). Step 2 is active. It contains a "Create filter name" section with a note about log events matching the pattern and a "Filter name" field set to "Cloudfilter". Below it is a "Metric details" section with fields for Metric namespace (set to "commanders"), Metric name (set to "metricrideucm"), and Metric value (set to "99.0"). The status bar at the bottom shows the URL as "us-east-2.console.aws.amazon.com/cloudwatch/home?region=us-east-2#logsV2:log-groups/log-group/cloudcommanders/crea...", the date as "4/19/2024", and the time as "11:52 AM".

6. Review and Create.

This screenshot continues from the previous one, showing the "Review and create" step of the metric filter creation process. The interface is identical to the previous step, with the "Assign metric" section now containing the assigned values: Filter name "Cloudfilter", Metric namespace "commanders", Metric name "metricrideucm", Metric value "99.0", and Unit "Milliseconds". The "Create metric filter" button is highlighted in orange at the bottom right. The status bar at the bottom shows the URL as "us-east-2.console.aws.amazon.com/cloudwatch/home?region=us-east-2#logsV2:log-groups/log-group/cloudcommanders/crea...", the date as "4/19/2024", and the time as "11:53 AM".

The screenshots illustrate the creation and configuration of a metric filter in AWS CloudWatch Metrics. In the first screenshot, a new metric filter named "Cloudfilter" has been created for the log group "cloudcommanders". The second screenshot shows the detailed configuration of this filter, including its filter pattern, metric name, value, unit, and dimensions.

Screenshot 1: Log group details

- Metric filter "Cloudfilter" has been created.**
- Log group details:**
 - Log class: Info
 - Standard
 - ARN: arn:aws:logs:us-east-2:381491822006:log-group:cloudcommanders:*
 - Creation time: 24 minutes ago
 - Retention: Never expire
 - Stored bytes: -
 - Metric filters: 1
 - Subscription filters: 0
 - Contributor Insights rules: -
 - KMS key ID: -
 - Anomaly detection: Configure
 - Data protection: -
 - Sensitive data count: -
- Metric filters (1)**

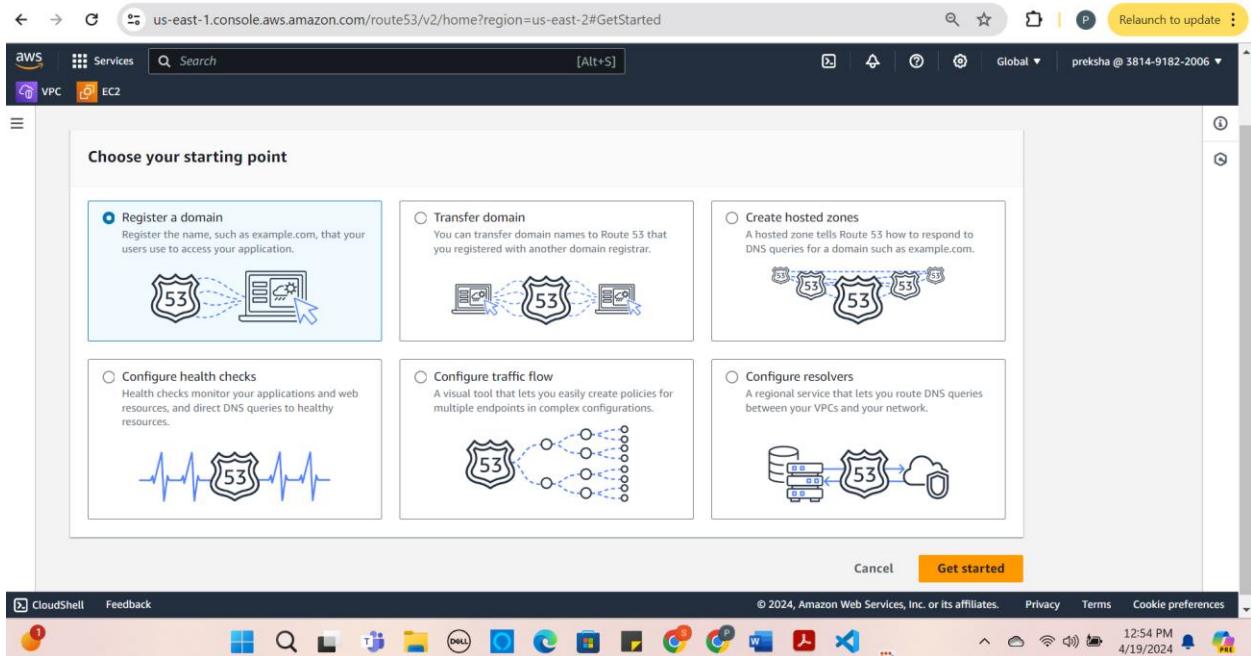
Screenshot 2: Metric filters list

- Metric filters (1)**
- Cloudfilter**
- Filter pattern: [host, logName, user, timestamp, request, statusCode=4*, size]
- Metric: commanderr / metricrideucm
- Metric value: 99.0
- Default value: -
- Unit: Milliseconds
- Dimensions: -
- Alarms: None.

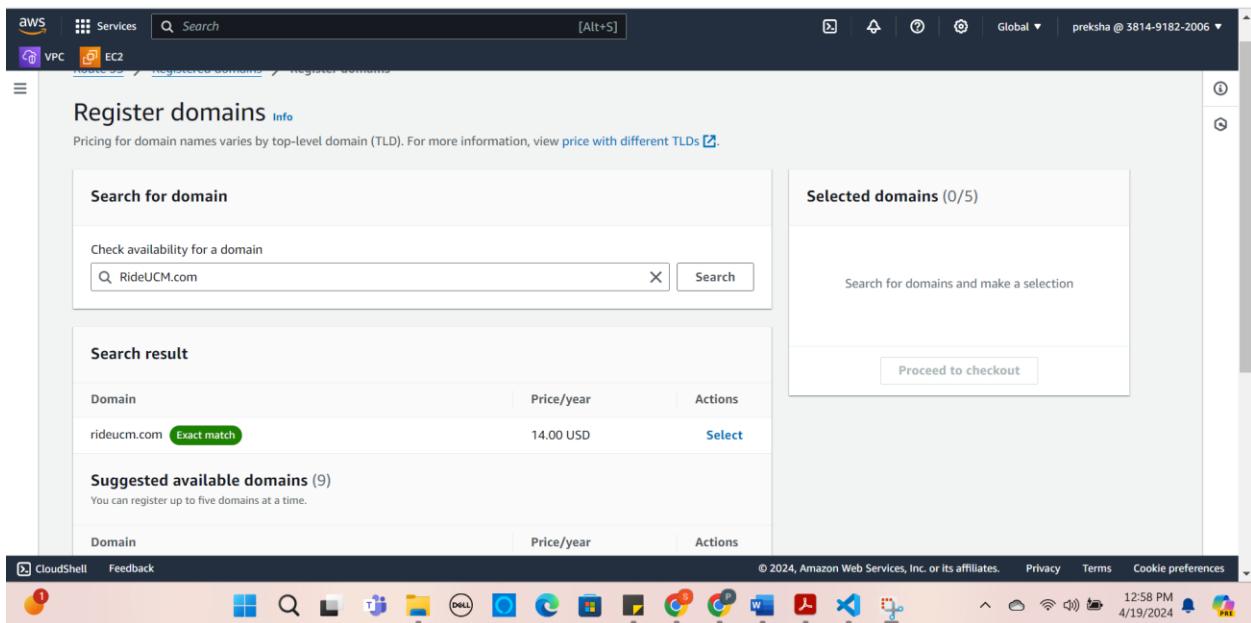
By setting this filter pattern and metrics, now cloudwatch will access logs from S3 Bucket. Now CloudWatch allows you to gain traffic flow and data flow of the website, performance, and issues when unwanted users access the website.

Creating a Domain name for the website using Amazon Route 53:

- Select Amazon Route 53 and click on Get Started.
- Choose Register a Domain.



- Register Domains and select your FQDN for your website and click on select.



- Pricing for the domain name.

The screenshot shows the AWS Route 53 Pricing step of a domain registration process. The domain name is rideucm.com, registered for 1 year at \$14.00 USD with auto-renew enabled. A note indicates that an email will be sent to the registrant contact before expiration to remind them about auto-renew. The subtotal is listed as 14.00 USD.

Domain name: rideucm.com

Duration (price): 1 year (14.00 USD)

Auto-renew: On

Subtotal: 14.00 USD

- The last step is Review and Submit and click submit.

The screenshot shows the AWS Route 53 Review and Submit step. It displays the domain registration details from the previous step and allows the user to review and edit contact information. The registrant contact is listed as preksha deshmukh.

Step 1: Pricing

| Domain name | Price | Year | Auto-renew |
|-------------|-----------|--------|------------|
| rideucm.com | 14.00 USD | 1 year | No |

Subtotal: 14.00 USD

Step 2: Contact information

| Registrant contact | Admin contact |
|--------------------|------------------|
| preksha deshmukh | preksha deshmukh |

Route 53 domain x Certificate Manager x (Optional) CloudFront x RideUCM x MS Information x Courses x Inbox (11) - p x +

us-east-1.console.aws.amazon.com/route53/domains/home?region=us-east-2#/ListRequests

aws Services Search [Alt+S]

VPC EC2

Route 53 Requests

Starting April 24, any communications you will receive about your domains will come from [noreply@registrar.amazon](#) email address instead of [route53-dev-admin@amazon.com](#).

Requests Info

| Operation ID | Domain name | Message | Status | Type | Submitted |
|--------------------------------------|-------------|---------|-------------|-----------------|-----------------------------------|
| 5869a73e-22b0-4696-a178-420663d65780 | rideucm.com | - | In progress | Register domain | April 19, 2024, 13:25 (UTC-05:00) |

Actions ▾

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 1:26 PM 4/19/2024

Route 53 | Global x Certificate Manager x (Optional) CloudFront x RideUCM x MS Information x Courses x Registration x +

us-east-1.console.aws.amazon.com/route53/domains/home?region=us-east-2#/ListRequests

aws Services Search [Alt+S]

VPC EC2

Route 53 Requests

Starting April 24, any communications you will receive about your domains will come from [noreply@registrar.amazon](#) email address instead of [route53-dev-admin@amazon.com](#).

Requests Info

| Operation ID | Domain name | Message | Status | Type | Submitted |
|--------------------------------------|-------------|---------|------------|-----------------|-----------------------------------|
| 5869a73e-22b0-4696-a178-420663d65780 | rideucm.com | - | Successful | Register domain | April 19, 2024, 13:25 (UTC-05:00) |

Actions ▾

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 1:38 PM 4/19/2024

Setting Up the Amazon Certificate Manager for the static website Launched in S3 Bucket:

- Request a Certificate:
- Request a public certificate.

The screenshot shows the 'Request certificate' page in the AWS Certificate Manager (ACM). The left sidebar lists 'AWS Certificate Manager (ACM)' with options like 'List certificates', 'Request certificate' (which is selected), 'Import certificate', and 'AWS Private CA'. The main content area is titled 'Request certificate' and has a 'Certificate type' section. It shows two options: 'Request a public certificate' (selected) and 'Request a private certificate'. A note below states: 'Requesting a private certificate requires the creation of a private certificate authority (CA). To create a private CA, visit AWS Private Certificate Authority'. At the bottom are 'Cancel' and 'Next' buttons. The browser's address bar shows the URL: us-east-2.console.aws.amazon.com/acm/home?region=us-east-2#/certificates/request.

- Select Domain names example rideucm.com and which method for validating the certificate. The standard recommended to do DNS Validation. Click request

The screenshot shows the 'Request public certificate' page in the AWS Certificate Manager (ACM). The left sidebar shows 'Request certificate' is selected. The main content area has a 'Domain names' section where 'rideucm.com' is entered into a 'Fully qualified domain name' input field. Below it is a button 'Add another name to this certificate' and a note about adding additional names. At the top, a success message says: 'Successfully requested certificate with ID 4d8fee1e-9d07-48b4-a8bd-6b817fda1282'. The browser's address bar shows the URL: us-east-2.console.aws.amazon.com/acm/home?region=us-east-2#/certificates/request/public.

- Click on the view certificate.

The screenshot shows the AWS Certificate Manager (ACM) interface. On the left, there's a sidebar with options like 'List certificates', 'Request certificate', 'Import certificate', and 'AWS Private CA'. The main area displays a certificate named 'dbf9d95a-2d73-4a34-9514-d76e5d3fafee'. Under 'Certificate status', it shows the identifier 'dbf9d95a-2d73-4a34-9514-d76e5d3fafee', ARN 'arn:aws:acm:us-east-2:381491822006:certificate/dbf9d95a-2d73-4a34-9514-d76e5d3fafee', and Type 'Amazon Issued'. The status is listed as 'Pending validation' with a 'Info' link. Below this, the 'Domains (1)' section lists 'rideucm.com' with a status of 'Pending validation'. A 'Resend validation email' button is available. At the bottom, there's a 'Details' section and a standard Windows taskbar with various application icons.

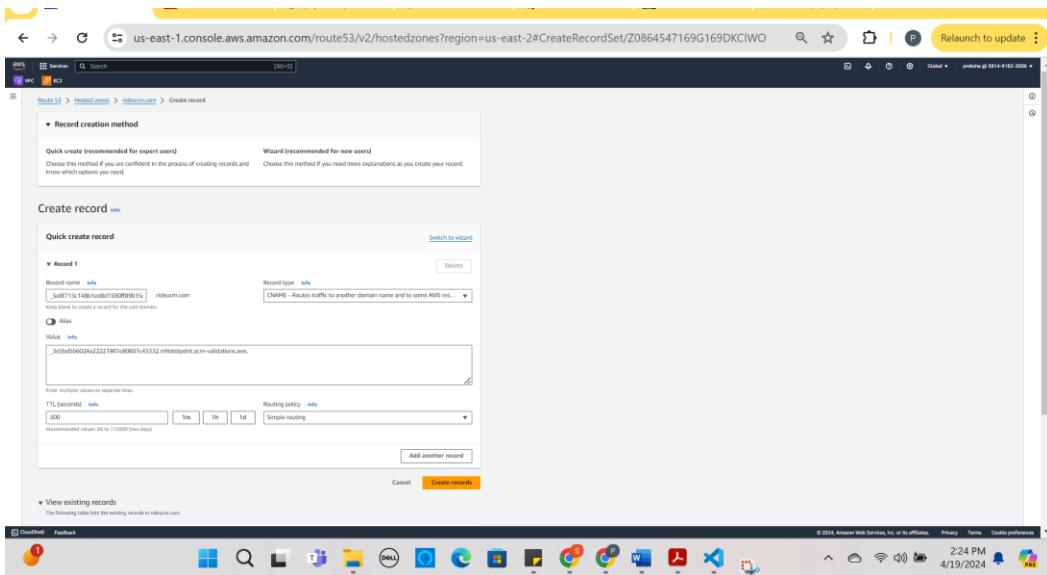
- For the Certificate to be issued we need to do one more step by making DNS validation.

1. Navigate to Route 53.
2. Select Hosted zones and click on Create Records.
3. Set the record type to CNAME.
4. The record name would be the subdomain of the certificate.

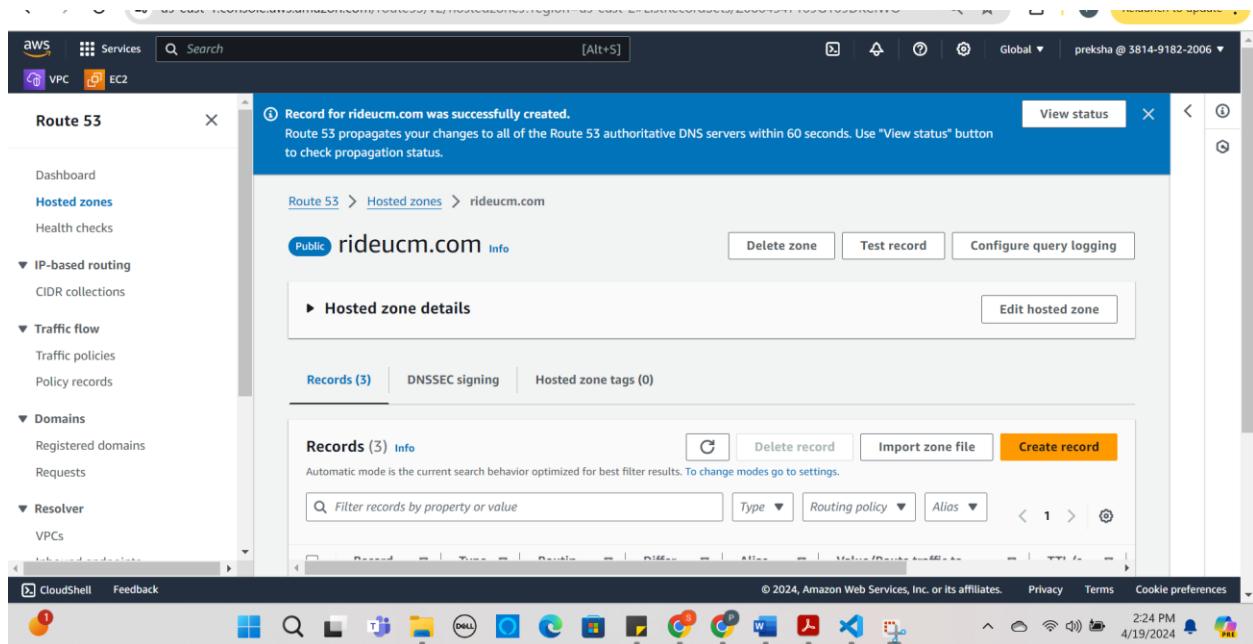
Example: CNAME: _5e8f713c148b1ed8d1590ff89015c046.rideucm.com.

Cvalue : _3e5bd5b6024a22221981e80601c43332.mhbtsbpdn.acm-validations.aws.

5. Click on create records.



7. Record is successfully added.



The above changes to the DNS configuration would take time. Once the CNAME record is updated Amazon Certificate Manager would be able to validate domain ownership and issue the SSL/TLS certificate.