Roll No: 45/A **Exam Seat No:**

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

Hashu Advani Memorial Complex, Collector's Colony, R. C. Marg, Chembur, Mumbai – 400074. Contact No. 02261532532

Since 1962

CERTIFICATE

Certified that Mr. Pushkar Prasad Sane of FYMCA-A has satisfactorily completed a course of the necessary experiments in AI Development Tool under my supervision in the Institute of Technology in the academic year 2023 -2024.

Principal

Head of Department
(Dr. Shivkumar Goel)

External Examiner

Subject Teacher
(Dr. Dhanamma Jagli)

V.E.S. Institute of Technology, Collector Colony, Chembur, Mumbai
Department of M.C.A

INDEX

SR. NO	CONTENTS	DATE OF PREPARATION	DATE OF SUBMISSION	MARKS	FACULTY SIGN
1.	**Assignment 1** 1. What is AWS ? 2. What is Cloud Computing ? Who uses Cloud Computing ? 3. What are benefits of Cloud Computing ?	25-01-2024	31-01-2024		
2.	**Practical 1.1** Aim- To set up various components of AI ML on AWS from IAM user.	25-01-2024	31-01-2024		
3.	**Practical 1.2** Aim :- To create a free S3 and EC2 instance on AWS	09-02-2024	14-02-2024		
4.	**Practical 2.1** Aim :- To study various components Amazon Sagemaker	15-02-2024	22-02-2024		
5.	**Practical 2.2** Aim:- To study various components of AWS studio and implement model in Canvas	21-03-2024	21-03-2024		
6.	**Practical 3** Aim:- To implement Data Labeling in AWS Sagemaker GroundTruth	21-03-2024	25-03-2024		

7.	**Practical 4** Aim:- To implement EDA using Data Wrangler in AWS	**21-03-2024**	**28-03-2024**		
8.	**Practical 5** Aim:- To implement sagemaker XGBoost algorithm	**21-03-2024**	**30-03-2024**		

Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 0**	
Title of Lab Assignment: Study of Azure Cloud, AWS and Google Cloud.		
DOP: 31-01-2024	**DOS: 01-02-2024**	
CO Mapped: **CO1**	**PO Mapped:** **PO1, PO2, PO3, PSO1, PSO2**	**Signature:**

Practical No. 0

Aim: Study of Azure Cloud, AWS and Google Cloud.

Description:

1. **Azure Cloud**

   Azure Cloud, commonly referred to as Microsoft Azure, is a cloud computing platform and service provided by Microsoft. Launched in 2010, Azure offers a wide range of cloud services, including computing power, storage solutions, networking capabilities, databases, analytics, artificial intelligence, and more. It enables individuals and organizations to build, deploy, and manage applications and services through Microsoft's global network of data centers.

   Key Features and Components of Azure:

   a. Compute Services:
      i. Azure Virtual Machines (VMs): Allows users to deploy and run Windows or Linux virtual machines in the cloud.
      ii. Azure App Service: A fully managed platform for building, deploying, and scaling web apps.

   b. Storage Services:
      i. Azure Blob Storage: Object storage service for unstructured data.
      ii. Azure Table Storage: NoSQL key-value store for semi-structured data.
      iii. Azure File Storage: Fully managed file shares in the cloud.

   c. Database Services:
      i. Azure SQL Database: A fully managed relational database service.
      ii. Azure Cosmos DB: A globally distributed, multi-model database for various types of data.

   d. Networking Services:
      i. Azure Virtual Network: Allows users to create private, isolated networks in the cloud.
      ii. Azure Load Balancer: Distributes incoming network traffic across multiple servers.

e. Identity and Access Management:
    i. Azure Active Directory (AD): Provides identity and access management services for applications and services.
f. AI and Machine Learning:
    i. Azure Machine Learning: Enables building, training, and deploying machine learning models.
g. Analytics and Big Data:
    i. Azure Synapse Analytics (formerly SQL Data Warehouse): An analytics service for large volumes of data.
    ii. Azure Data Lake Storage: Scalable and secure data lake for big data analytics.
h. Internet of Things (IoT):
    i. Azure IoT Hub: Connects, monitors, and manages IoT assets.
i. Security and Compliance:
    i. Azure Security Center: Provides advanced threat protection across hybrid cloud workloads.
    ii. Azure Policy: Helps enforce organizational standards and compliance.
j. DevOps and Developer Tools:
    i. Azure DevOps Services: A set of development tools for planning, tracking, and delivering software.

2. **Amazon Web Services:**

Amazon Web Services (AWS) is a comprehensive and widely adopted cloud computing platform provided by Amazon. It offers a vast array of cloud services, allowing individuals, businesses, and organizations to access computing power, storage, databases, machine learning, analytics, and other resources over the internet. AWS is known for its flexibility, scalability, and reliability, making it a popular choice for various applications and workloads.

Here are some key aspects of AWS:

a. Service Offering:
    i. AWS provides a broad range of services, including computing power (Amazon EC2), storage (Amazon S3), databases (Amazon RDS), machine learning (Amazon SageMaker), serverless computing (AWS Lambda), and more.
b. Global Infrastructure:
    i. AWS operates a global network of data centers, referred to as Availability Zones, located in different regions around the world. This infrastructure allows users to deploy applications and services close to their end-users, reducing latency and improving performance.
c. Pricing Model:
    i. AWS follows a pay-as-you-go pricing model, where users are billed based on their actual usage of resources. This flexibility is beneficial for businesses as they only pay for the resources they consume.
d. Security and Compliance:
    i. AWS places a strong emphasis on security, offering various security services and features. It has a robust compliance program, meeting various industry-specific standards and certifications.
e. Ecosystem and Marketplace:
    i. AWS has a vast ecosystem of partners, third-party tools, and a marketplace where users can find and deploy additional services, solutions, and applications to enhance their cloud environment.

f. Community and Support:
    i. AWS has a large and active community of developers, architects, and businesses. It provides extensive documentation, training resources, and customer support to help users make the most of the platform.
g. Innovation and Evolution:
    i. AWS continually introduces new services and features, staying at the forefront of cloud innovation. This allows users to leverage cutting-edge technologies and stay competitive in their respective industries

3. Google Cloud:

Google Cloud Platform (GCP) is a suite of cloud computing services offered by Google. It provides a wide range of infrastructure and platform services for building, deploying, and scaling applications in the cloud. GCP is designed to offer powerful and flexible solutions for businesses, developers, and data scientists. Here are some key aspects of Google Cloud:

Key Services and Features:

a. Compute Services:
    i. Google Compute Engine (GCE): Infrastructure as a Service (IaaS) that allows users to run virtual machines in the Google Cloud.
b. Storage Services:
    i. Google Cloud Storage: Object storage service designed for secure and scalable storage of data.
c. Databases:
    i. Google Cloud SQL: Fully managed relational database service.
    ii. Google Cloud Firestore: NoSQL document database for mobile and web application development.
d. Big Data and Analytics:
    i. BigQuery: Serverless, highly scalable, and cost-effective multi-cloud data warehouse for analytics.
    ii. Dataproc: Fully managed Apache Spark and Hadoop service for big data processing.

e. Machine Learning and AI:
    i. AI Platform: End-to-end platform for building, testing, and deploying machine learning models.
    ii. TensorFlow: An open-source machine learning framework developed by the Google Brain team.
f. Networking:
    i. Virtual Private Cloud (VPC): Provides networking functionality for GCP resources.
    ii. Cloud Load Balancing: Distributes incoming network traffic across multiple instances.
g. Security and Identity:
    i. Identity and Access Management (IAM): Manages access control for GCP resources.
    ii. Cloud Identity-Aware Proxy (IAP): Controls access to cloud applications.
h. Serverless Computing:
    i. Cloud Functions: Executes event-driven functions without the need to provision or manage servers.
i. Containers and Kubernetes:
    i. Google Kubernetes Engine (GKE): Managed Kubernetes service for deploying, managing, and scaling containerized applications.
j. Internet of Things (IoT):
    i. Cloud IoT Core: Fully managed service for connecting, managing, and ingesting data from IoT devices.

<u>Unique Aspects:</u>

a. Data Centers and Global Network:
    i. Google Cloud operates a global network of data centers to provide low-latency access to services.
b. Emphasis on Data Analytics and Machine Learning:
    i. GCP is known for its strengths in data analytics, machine learning, and artificial intelligence services.

c. Containerization and Kubernetes:
    i. Google has been a pioneer in containerization, and GCP offers strong support for Kubernetes.
d. Integration with Google Workspace:
    i. Integration with Google Workspace (formerly G Suite) for collaboration and productivity tools.
e. Open Source Contributions:
    i. Google has a history of contributing to open-source projects, and many of its cloud services are built on open-source technologies.

Azure, AWS, and Google Cloud Platform (GCP) are all cloud service providers, but they differ in various aspects, including their services, architecture, pricing models, and target audiences. Choosing between Azure, AWS, and GCP often depends on specific organizational requirements, existing technology stacks, and individual preferences. Many organizations adopt a multi-cloud strategy to leverage the strengths of each provider for different aspects of their infrastructure and applications.

Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 1**	
Title of Lab Assignment: Introduction to AWS signup and Service Legal Agreement, Introduction to AWS sagemaker (Signup, freetier, billing & Iam), AIML components in AWS, Simple Storage Service (S3) and Elastic Computing Cloud (EC2) Introduction.		
DOP: 25-01-2024	**DOS: 06-02-2024**	
CO Mapped: **CO1**	**PO Mapped:** **PO1, PO2, PO3, PSO1, PSO2**	**Signature:**

Practical No. 1

Aim: Introduction to AWS signup and Service Legal Agreement, Introduction to AWS sagemaker (Signup, freetier, billing & Iam), AIML components in AWS, Simple Storage Service (S3) and Elastic Computing Cloud (EC2) Introduction.

Description:

Introduction to AWS Sign Up and Service Legal Agreement: When signing up for Amazon Web Services (AWS), users embark on a streamlined process that involves creating an AWS account. This account provides access to a comprehensive suite of cloud services. During the signup, users must agree to AWS's Service Legal Agreement, which outlines terms and conditions governing the use of AWS services, ensuring a clear understanding of responsibilities and obligations.

Here's a step-by-step guide to help you with the AWS sign-up process:

1. **Visit the AWS Website:**
   Open your web browser and go to the official AWS website at https://aws.amazon.com/

2. **Choose "Sign in to the Console":**
   On the AWS homepage, click on the "Sign in to the Console" button located in the top right corner.

3. **Create an AWS Account:**
   If you don't already have an AWS account, click on the "Create a new AWS account" button. If you have an existing AWS account, you can sign in with your credentials.

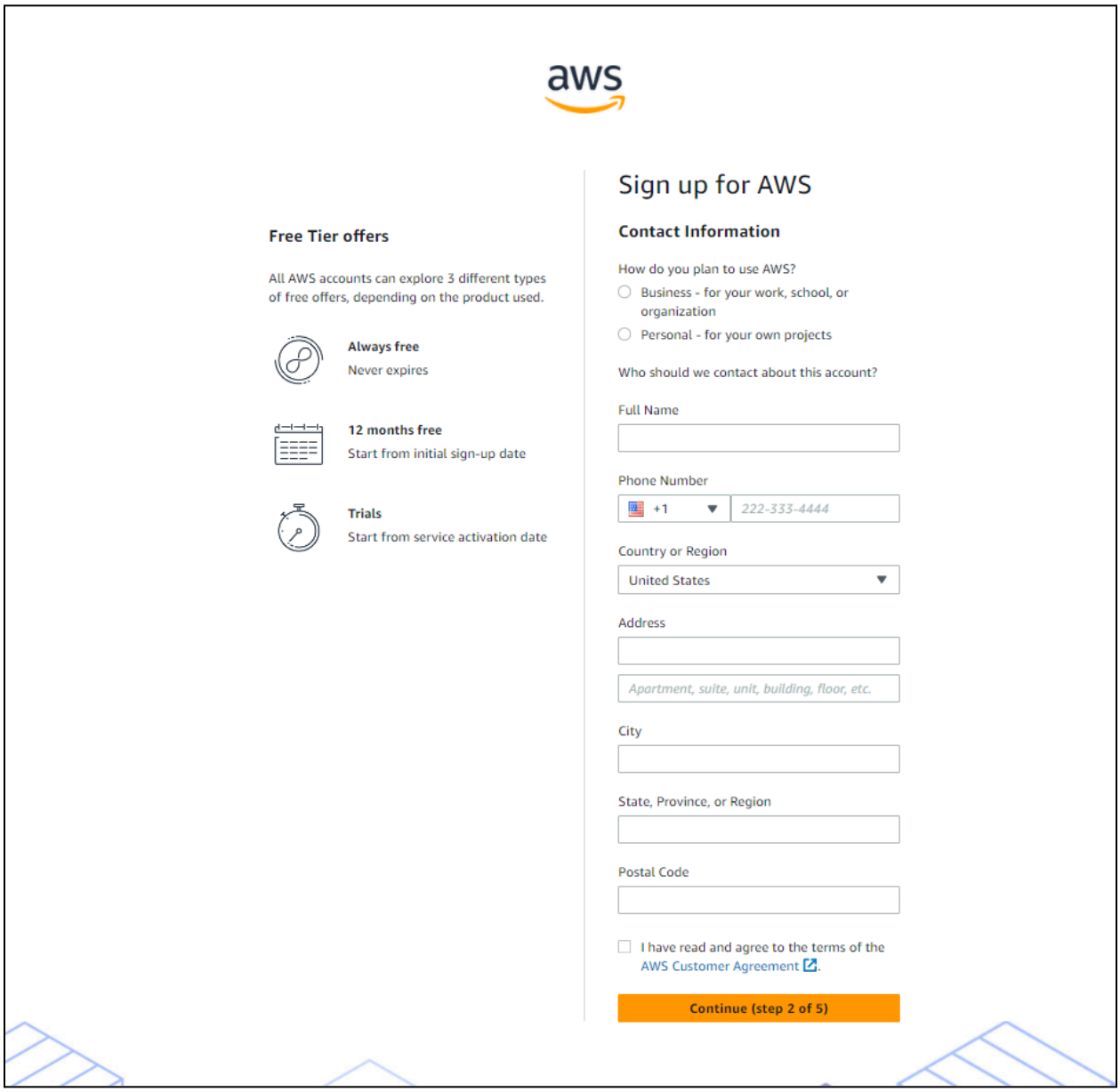
4. **Provide Your Email Address:**
   Enter the email address you want to associate with your AWS account. Make sure to use a valid email address as AWS will send important notifications and account-related information to this address.

5. **Set up a Password:**

   Choose a secure password for your AWS account. AWS has specific requirements for passwords, so ensure that your chosen password meets the criteria.

6. **Enter Your Account Information:**

   Fill in your account information, including your account name (which will be used as your AWS account alias), full name, and contact information.

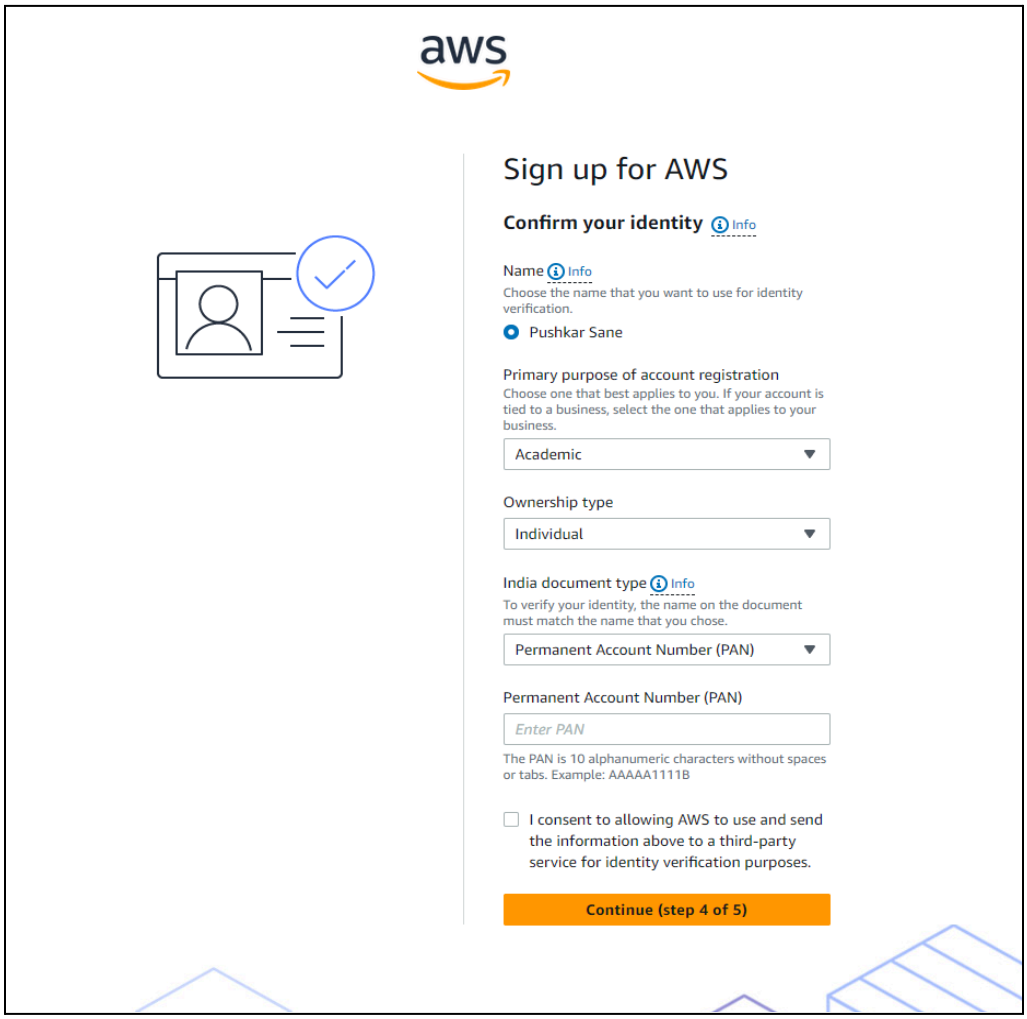


7. **Payment Information:**

   Provide your payment information. AWS requires valid payment details even if you plan to use free-tier services. This is to verify your identity and prevent misuse of AWS resources.

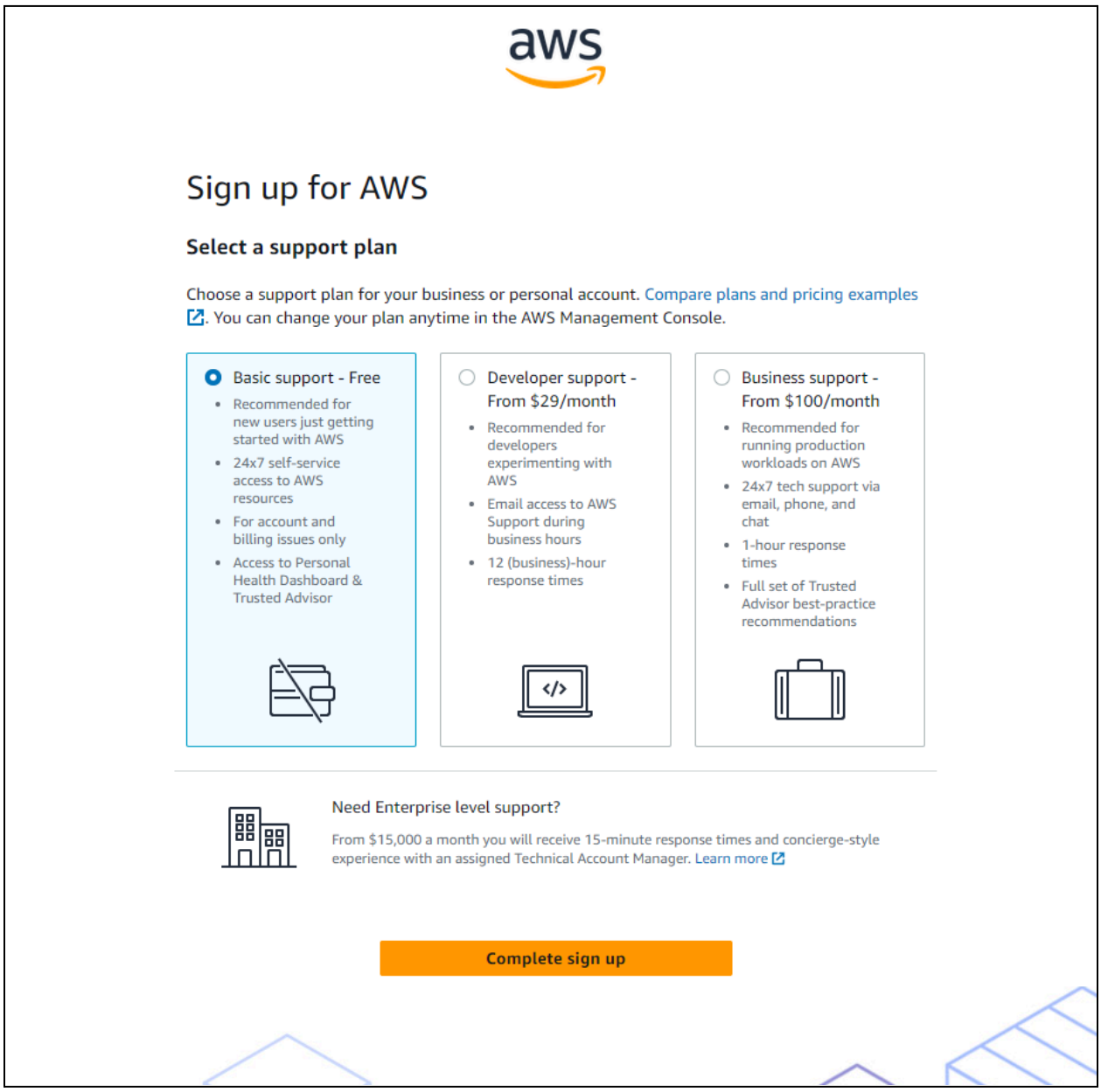
8. **Identity Verification:**

   AWS may perform an identity verification process to ensure the security of your account. This may involve a phone call or text message verification.



9. **Choose a Support Plan:**

   AWS offers different support plans, including a free basic plan and various premium plans with additional features. Choose the plan that best fits your needs.

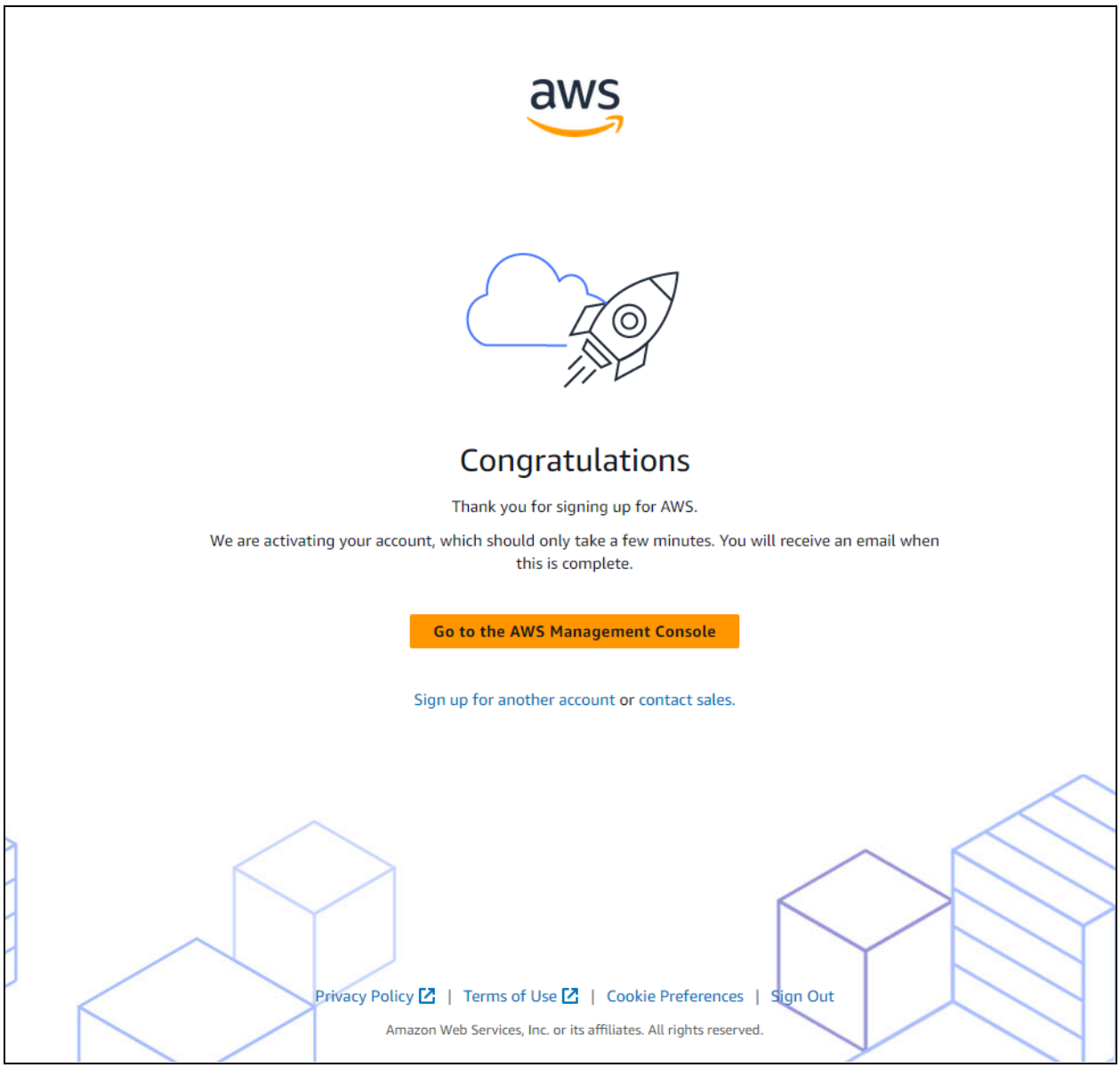


10. **Review and Confirm:**

    Review the information you provided, including your email address, account details, and payment information. Once you are satisfied, click on the "Create Account and Continue" button.

11. Confirmation Email:

AWS will send a confirmation email to the address you provided. Open the email and click on the confirmation link to verify your email address.

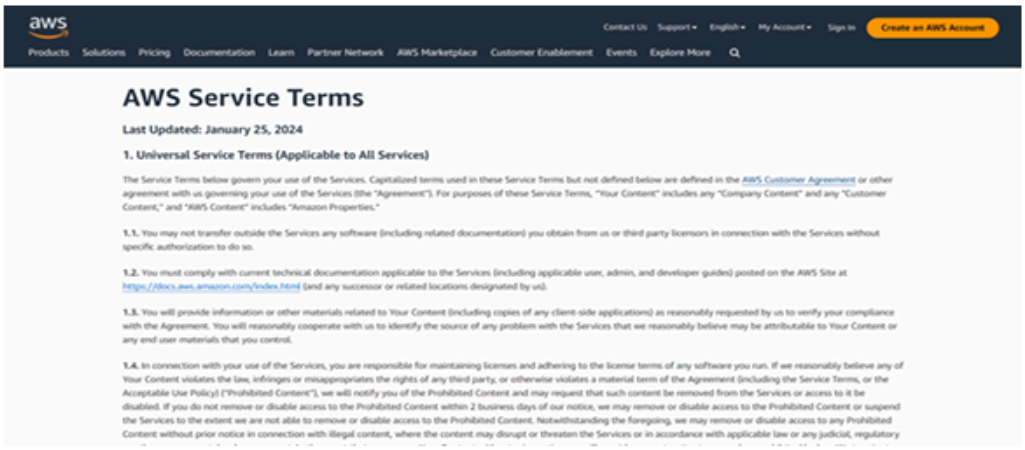


12. Set up Multi-Factor Authentication (Optional):

For added security, consider setting up Multi-Factor Authentication (MFA) for your AWS account. This can be done during the sign-up process or later within the AWS Management Console.

Once you've completed these steps, your AWS account will be created, and you can start using the AWS services. Keep your account credentials secure, and consider enabling additional security measures, such as IAM (Identity and Access Management) policies, for better control over your AWS resources.

Service Legal Agreement:



aws

Products Solutions Pricing Documentation Learn Partner Network AWS Marketplace Customer Enablement Events Explore More

Contact Us Support English My Account Sign In Create an AWS Account

AWS Service Terms

Last Updated: January 25, 2024

1. Universal Service Terms (Applicable to All Services)

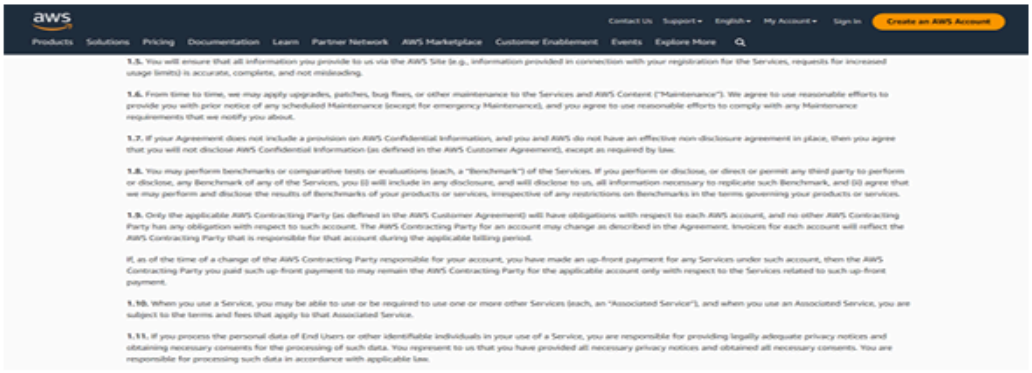
The Service Terms below govern your use of the Services. Capitalized terms used in these Service Terms but not defined below are defined in the AWS Customer Agreement or other agreement with us governing your use of the Services (the "Agreement"). For purposes of these Service Terms, "Your Content" includes any "Company Content" and any "Customer Content," and "AWS Content" includes "Amazon Properties."

1.1. You may not transfer outside the Services any software (including related documentation) you obtain from us or third party licensors in connection with the Services without specific authorization to do so.

1.2. You must comply with current technical documentation applicable to the Services (including applicable user, admin, and developer guides) posted on the AWS Site at https://docs.aws.amazon.com/index.html (and any successor or related locations designated by us).

1.3. You will provide information or other materials related to Your Content (including copies of any client-side applications) as reasonably requested by us to verify your compliance with the Agreement. You will reasonably cooperate with us to identify the source of any problem with the Services that we reasonably believe may be attributable to Your Content or any end user materials that you control.

1.4. In connection with your use of the Services, you are responsible for maintaining licenses and adhering to the license terms of any software you run. If we reasonably believe any of Your Content violates the law, infringes or misappropriates the rights of any third party, or otherwise violates a material term of the Agreement (including the Service Terms, or the Acceptable Use Policy) ("Prohibited Content"), we will notify you of the Prohibited Content and may request that such content be removed from the Services or access to it be disabled. If you do not remove or disable access to the Prohibited Content within 2 business days of our notice, we may remove or disable access to the Prohibited Content or suspend the Services to the extent we are not able to remove or disable access to the Prohibited Content. Notwithstanding the foregoing, we may remove or disable access to any Prohibited Content without prior notice in connection with illegal content, where the content may disrupt or threaten the Services or in accordance with applicable law or any judicial, regulatory



aws

Products Solutions Pricing Documentation Learn Partner Network AWS Marketplace Customer Enablement Events Explore More

Contact Us Support English My Account Sign In Create an AWS Account

1.5. You will ensure that all information you provide to us via the AWS Site (e.g., information provided in connection with your registration for the Services, requests for increased usage limits) is accurate, complete, and not misleading.

1.6. From time to time, we may apply upgrades, patches, bug fixes, or other maintenance to the Services and AWS Content ("Maintenance"). We agree to use reasonable efforts to provide you with prior notice of any scheduled Maintenance (except for emergency Maintenance), and you agree to use reasonable efforts to comply with any Maintenance requirements that we notify you about.

1.7. If your Agreement does not include a provision on AWS Confidential Information, and you and AWS do not have an effective non-disclosure agreement in place, then you agree that you will not disclose AWS Confidential Information (as defined in the AWS Customer Agreement), except as required by law.

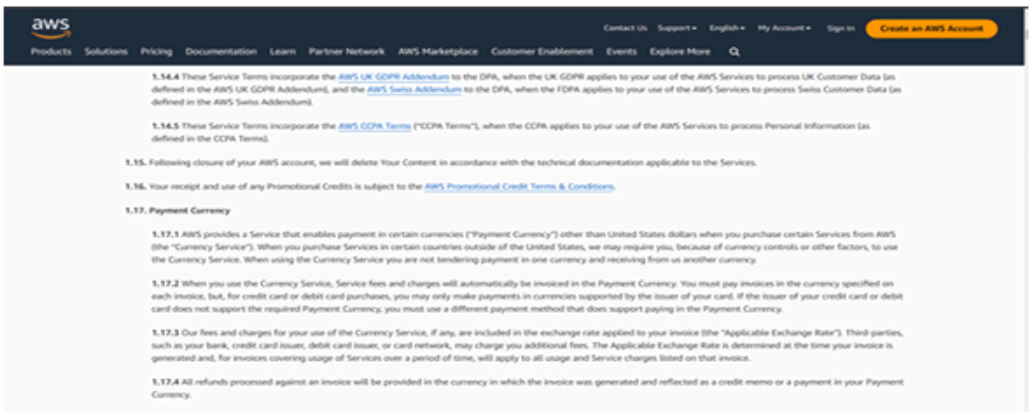
1.8. You may perform benchmarks or comparative tests or evaluations (each, a "Benchmark") of the Services. If you perform or disclose, or direct or permit any third party to perform or disclose, any Benchmark of any of the Services, you (i) will include in any disclosure, and will disclose to us, all information necessary to replicate such Benchmark, and (ii) agree that we may perform and disclose the results of Benchmarks of your products or services, irrespective of any restrictions on Benchmarks in the terms governing your products or services.

1.9. Only the applicable AWS Contracting Party (as defined in the AWS Customer Agreement) will have obligations with respect to each AWS account, and no other AWS Contracting Party has any obligation with respect to such account. The AWS Contracting Party for an account may change as described in the Agreement. Invoices for each account will reflect the AWS Contracting Party that is responsible for that account during the applicable billing period.

If, as of the time of a change of the AWS Contracting Party responsible for your account, you have made an up-front payment for any Services under such account, then the AWS Contracting Party you paid such up-front payment to may remain the AWS Contracting Party for the applicable account only with respect to the Services related to such up-front payment.

1.10. When you use a Service, you may be able to use or be required to use one or more other Services (each, an "Associated Service"), and when you use an Associated Service, you are subject to the terms and fees that apply to that Associated Service.

1.11. If you process the personal data of End Users or other identifiable individuals in your use of a Service, you are responsible for providing legally adequate privacy notices and obtaining necessary consents for the processing of such data. You represent to us that you have provided all necessary privacy notices and obtained all necessary consents. You are responsible for processing such data in accordance with applicable law.



aws

Products Solutions Pricing Documentation Learn Partner Network AWS Marketplace Customer Enablement Events Explore More

Contact Us Support English My Account Sign In Create an AWS Account

1.14.4 These Service Terms incorporate the AWS UK GDPR Addendum to the DPA, when the UK GDPR applies to your use of the AWS Services to process UK Customer Data (as defined in the AWS UK GDPR Addendum), and the AWS Swiss Addendum to the DPA, when the FDPA applies to your use of the AWS Services to process Swiss Customer Data (as defined in the AWS Swiss Addendum).

1.14.5 These Service Terms incorporate the AWS CCPA Terms ("CCPA Terms"), when the CCPA applies to your use of the AWS Services to process Personal Information (as defined in the CCPA Terms).

1.15. Following closure of your AWS account, we will delete Your Content in accordance with the technical documentation applicable to the Services.

1.16. Your receipt and use of any Promotional Credits is subject to the AWS Promotional Credit Terms & Conditions.

1.17. Payment Currency

1.17.1 AWS provides a Service that enables payment in certain currencies ("Payment Currency") other than United States dollars when you purchase certain Services from AWS (the "Currency Service"). When you purchase Services in certain countries outside of the United States, we may require you, because of currency controls or other factors, to use the Currency Service. When using the Currency Service you are not tendering payment in one currency and receiving from us another currency.

1.17.2 When you use the Currency Service, Service fees and charges will automatically be invoiced in the Payment Currency. You must pay invoices in the currency specified on each invoice, but, for credit card or debit card purchases, you may only make payments in currencies supported by the issuer of your card. If the issuer of your credit card or debit card does not support the required Payment Currency, you must use a different payment method that does support paying in the Payment Currency.

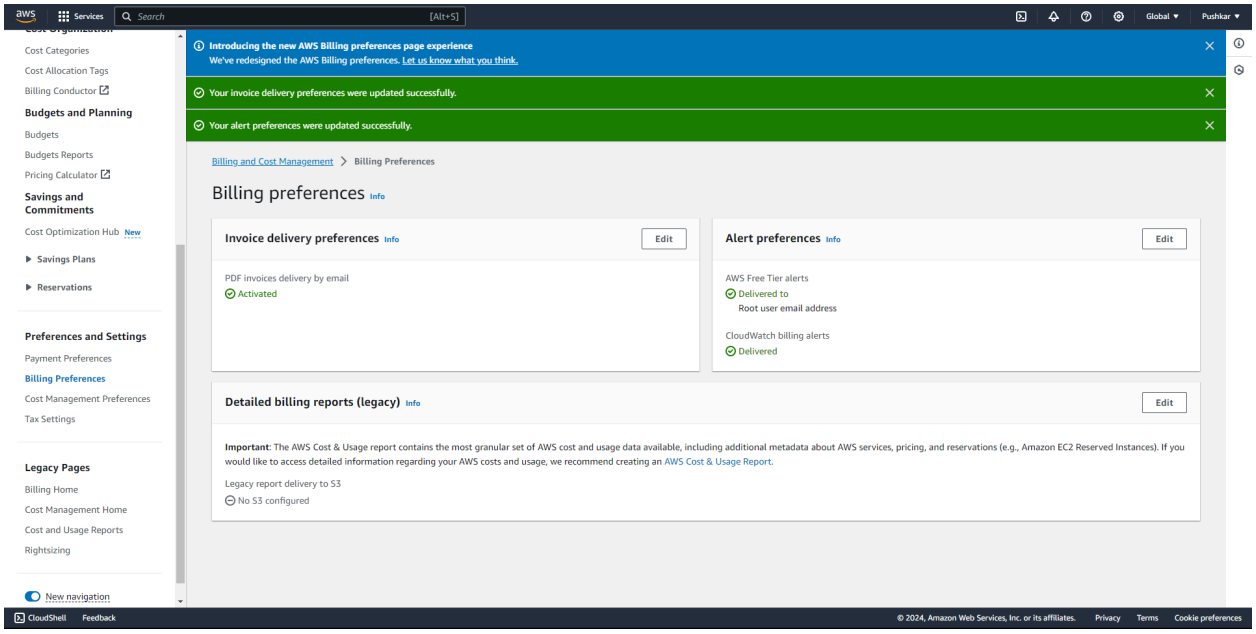
1.17.3 Our fees and charges for your use of the Currency Service, if any, are included in the exchange rate applied to your invoice (the "Applicable Exchange Rate"). Third-parties, such as your bank, credit card issuer, debit card issuer, or card network, may charge you additional fees. The Applicable Exchange Rate is determined at the time your invoice is generated and, for invoices covering usage of Services over a period of time, will apply to all usage and Service charges listed on that invoice.

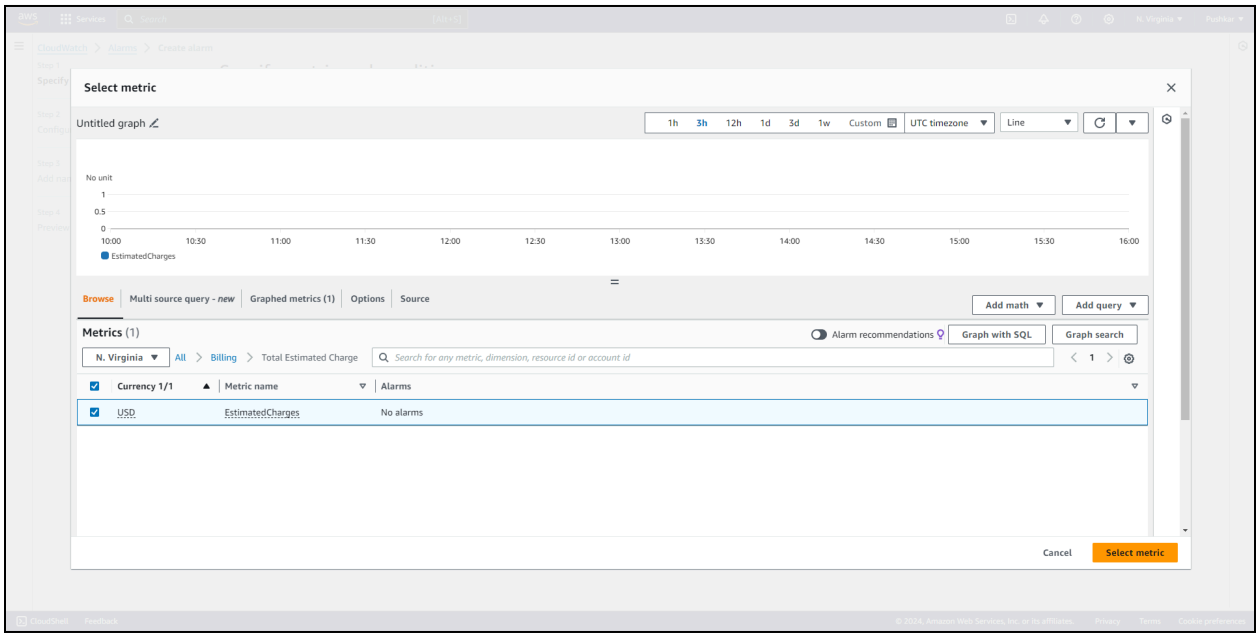
1.17.4 All refunds processed against an invoice will be provided in the currency in which the invoice was generated and reflected as a credit memo or a payment in your Payment Currency.

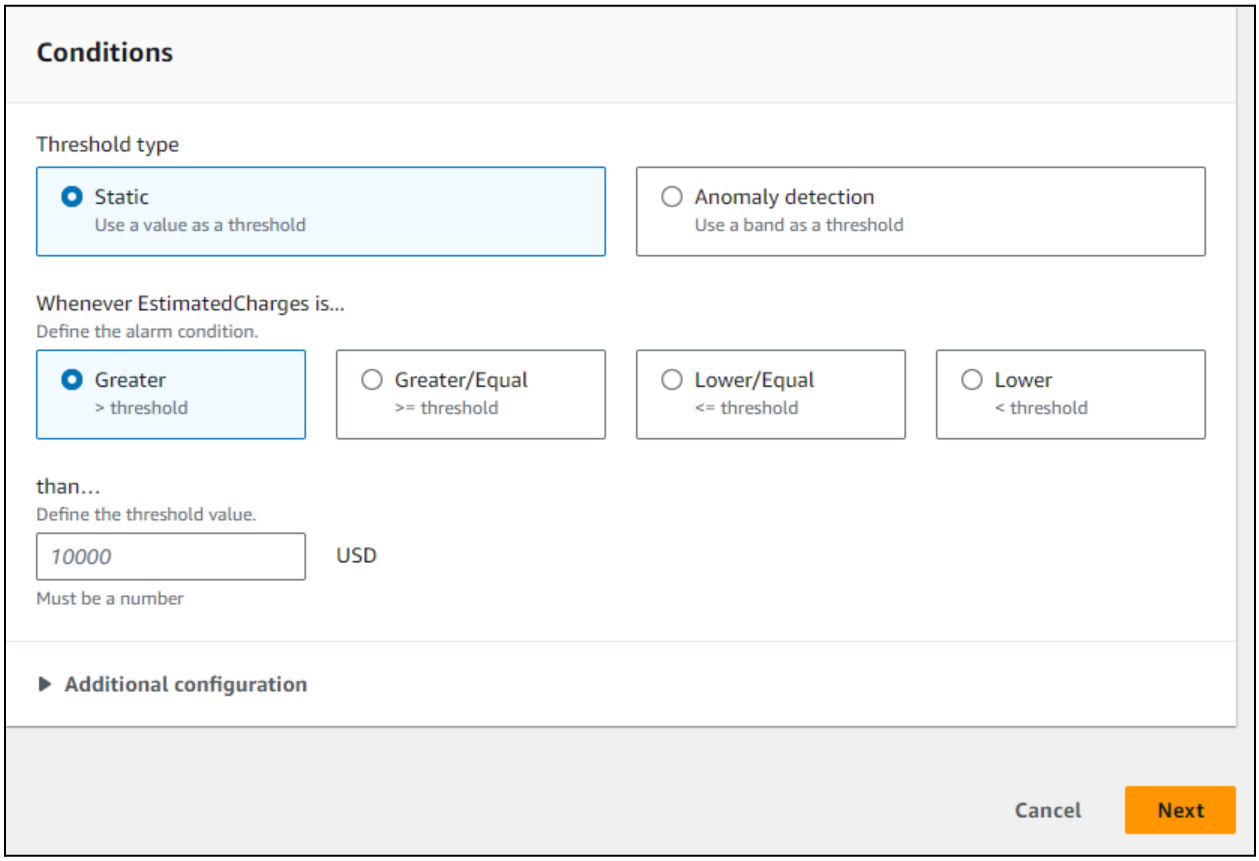
Introduction to AWS SageMaker:

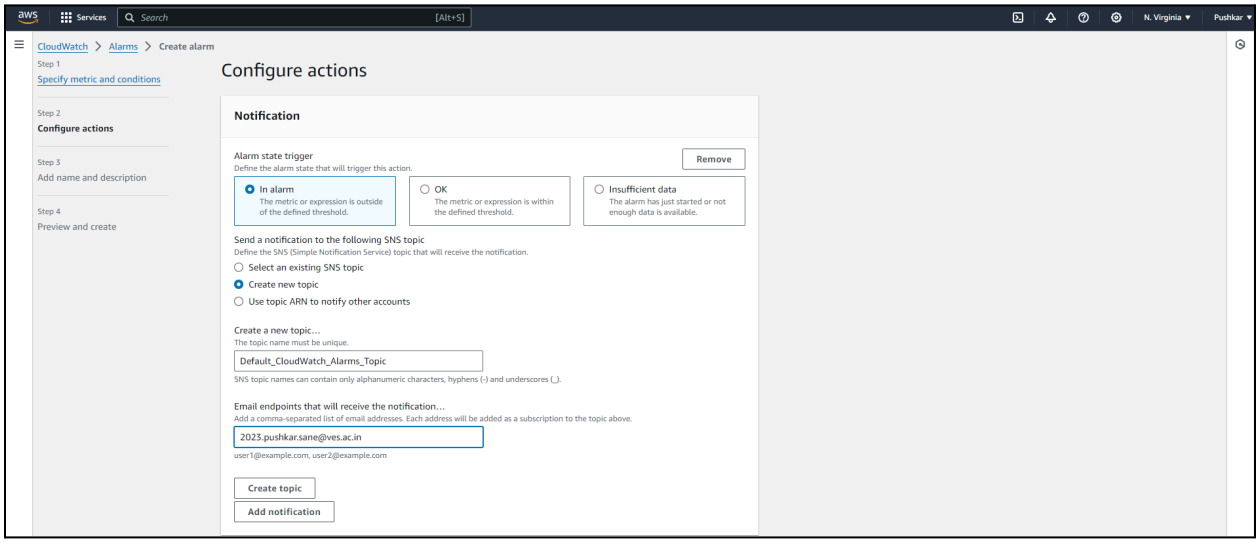
Amazon SageMaker is a managed machine learning service by AWS. It streamlines the entire machine learning workflow, from data labeling and model training to deployment and monitoring. With built-in algorithms, hyperparameter optimization, and easy integration with AWS ecosystem, SageMaker makes machine learning accessible to developers and data scientists.

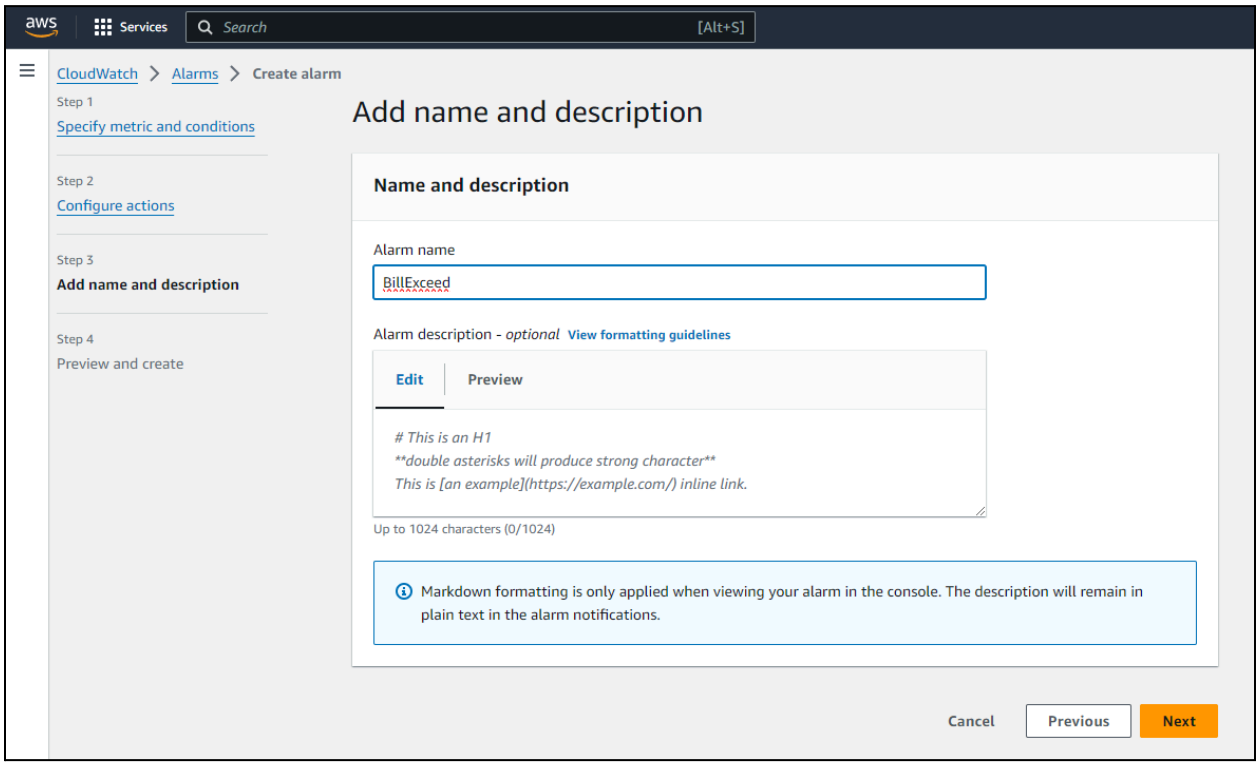
Billing Process:

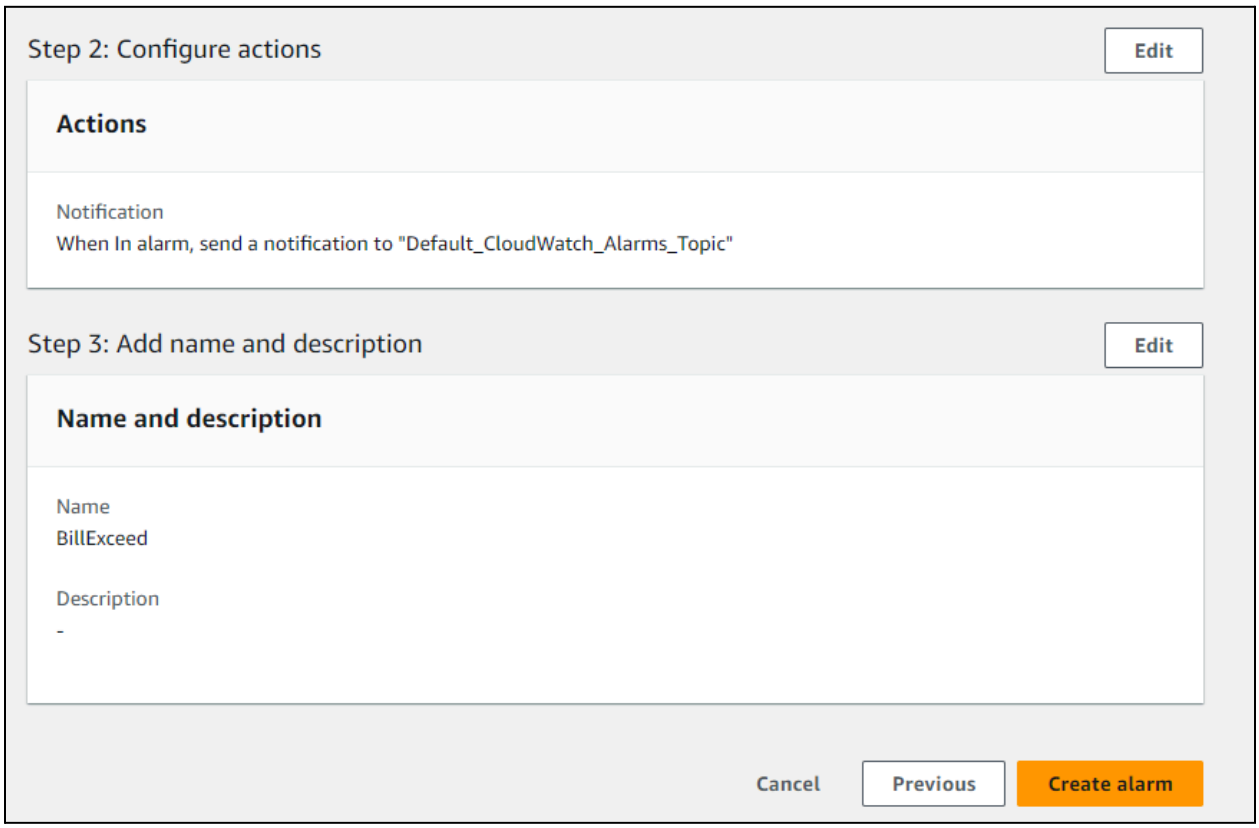


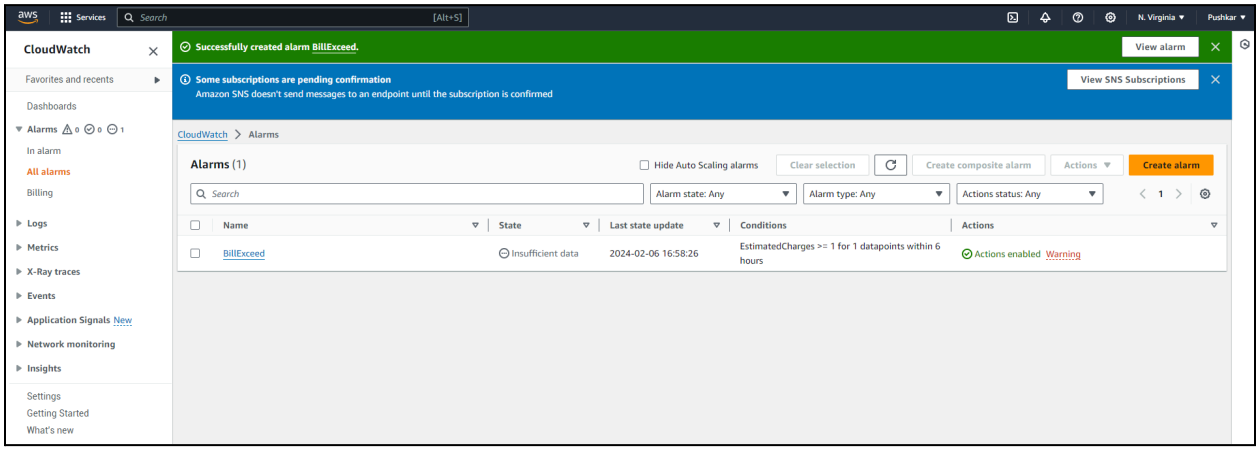



Conditions
Threshold type
Static
Use a value as a threshold
Anomaly detection
Use a band as a threshold
Whenever EstimatedCharges is...
Define the alarm condition.
Greater
> threshold
Greater/Equal
>= threshold
Lower/Equal
<= threshold
Lower
< threshold
than...
Define the threshold value.
10000
USD
Must be a number
Additional configuration
Cancel
Next

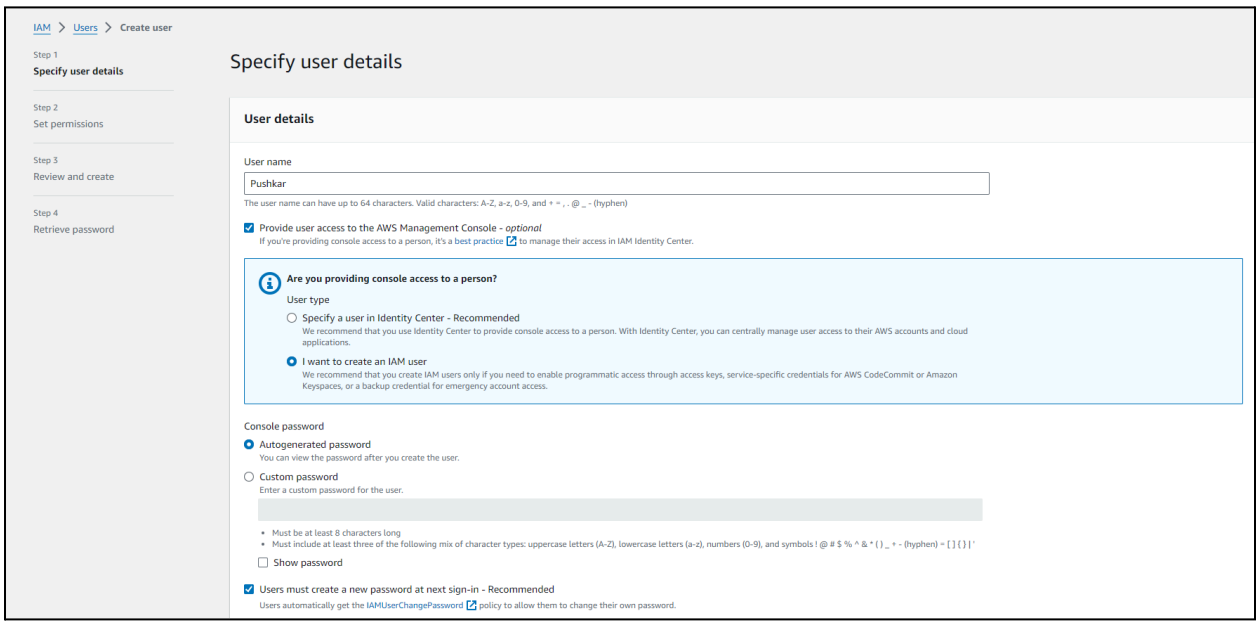

aws
Services
Search
[Alt+S]
N. Virginia
Pushkar
CloudWatch
Alarms
Create alarm
Step 1
Specify metric and conditions
Step 2
Configure actions
Step 3
Add name and description
Step 4
Preview and create
Configure actions
Notification
Alarm state trigger
Define the alarm state that will trigger this action.
Remove
In alarm
The metric or expression is outside of the defined threshold.
OK
The metric or expression is within the defined threshold.
Insufficient data
The alarm has just started or not enough data is available.
Send a notification to the following SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.
Select an existing SNS topic
Create new topic
Use topic ARN to notify other accounts
Create a new topic...
The topic name must be unique.
Default_CloudWatch_Alarms_Topic
SNS topic names can contain only alphanumeric characters, hyphens (-) and underscores (_).
Email endpoints that will receive the notification...
Add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.
2023.pushkar.sane@ves.ac.in
user1@example.com, user2@example.com
Create topic
Add notification

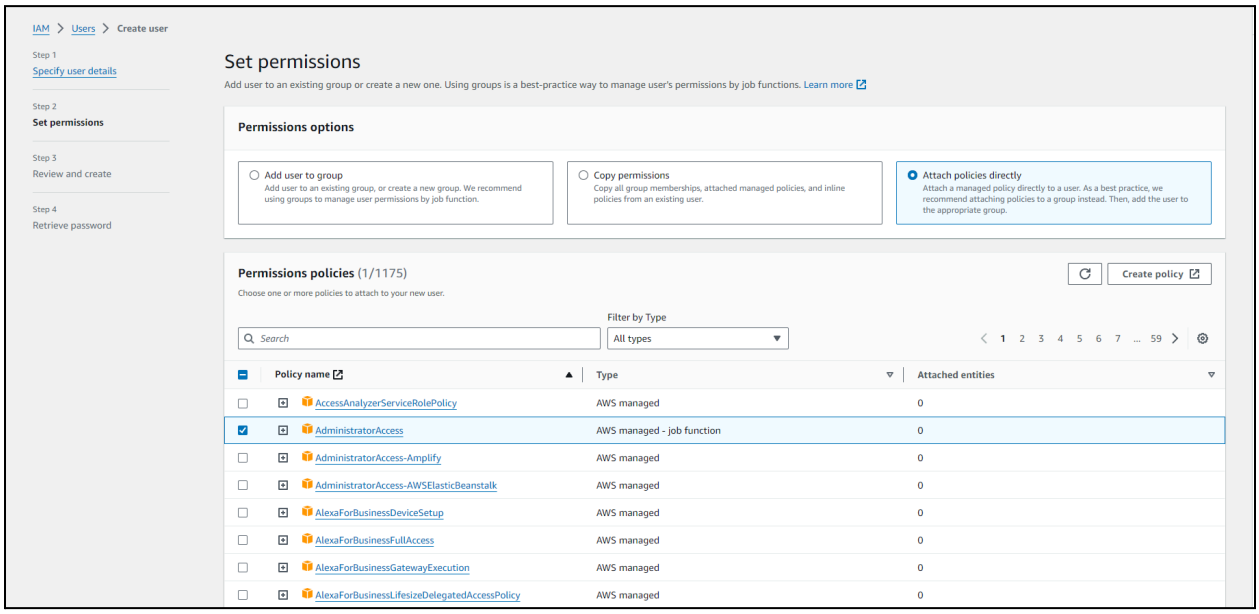

aws
Services
Search
[Alt+S]
CloudWatch
Alarms
Create alarm
Step 1
Specify metric and conditions
Step 2
Configure actions
Step 3
Add name and description
Step 4
Preview and create
Add name and description
Name and description
Alarm name
BillExceed
Alarm description - optional
View formatting guidelines
Edit
Preview
This is an H1
double asterisks will produce strong character
This is [an example](https://example.com/) inline link.
Up to 1024 characters (0/1024)
Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.
Cancel
Previous
Next

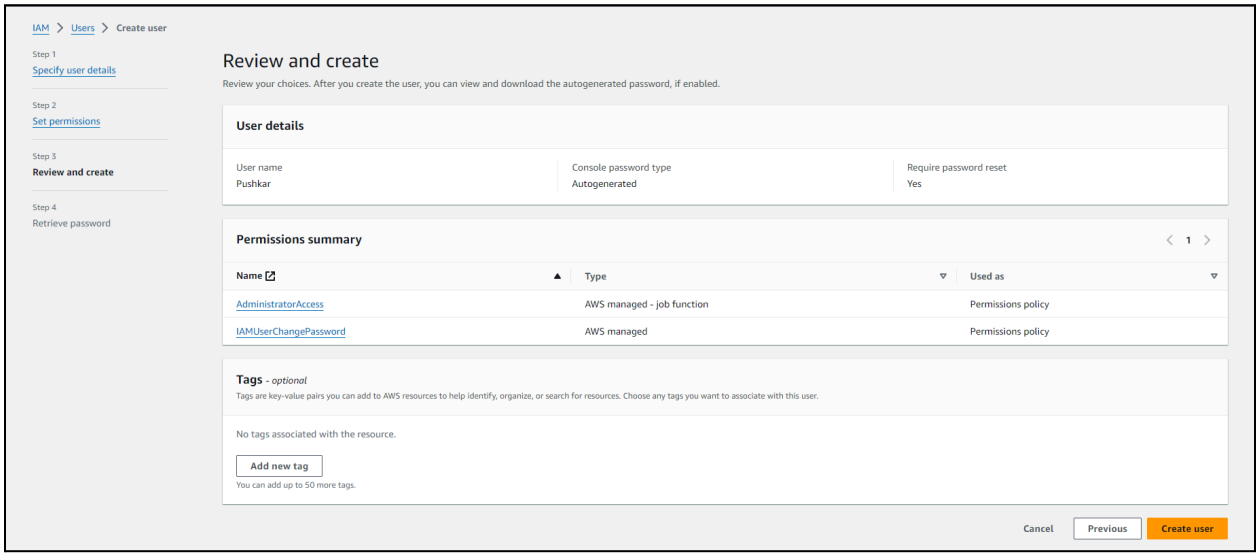

Step 2: Configure actions
Edit
Actions
Notification
When In alarm, send a notification to "Default_CloudWatch_Alarms_Topic"
Step 3: Add name and description
Edit
Name and description
Name
BillExceed
Description
-
Cancel
Previous
Create alarm

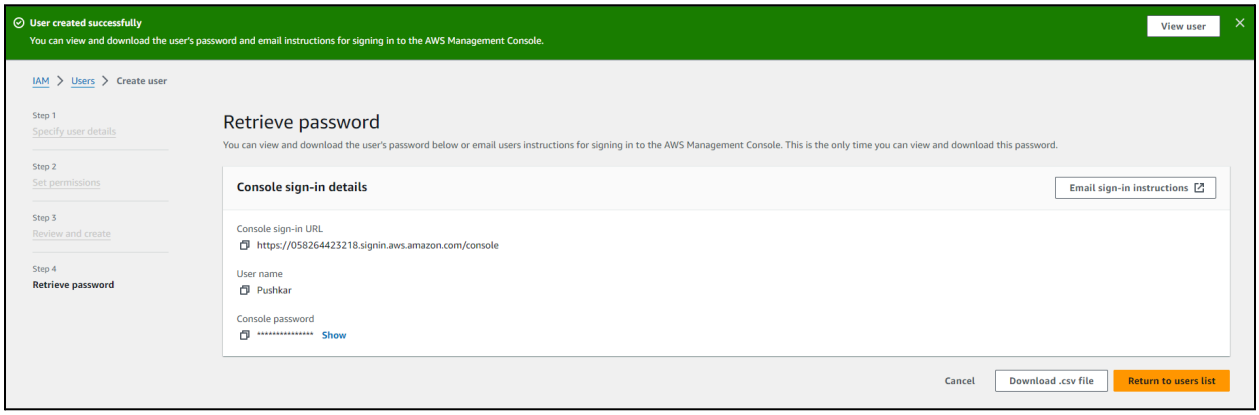

aws
Services
Search
[Alt+S]
N. Virginia
Pushkar
CloudWatch
Favorites and recents
Dashboards
Alarms 0 0 1
In alarm
All alarms
Billing
Logs
Metrics
X-Ray traces
Events
Application Signals New
Network monitoring
Insights
Settings
Getting Started
What's new
Successfully created alarm BillExceed.
View alarm
Some subscriptions are pending confirmation
Amazon SNS doesn't send messages to an endpoint until the subscription is confirmed
View SNS Subscriptions
CloudWatch > Alarms
Alarms (1)
Hide Auto Scaling alarms
Clear selection
Create composite alarm
Actions
Create alarm
Search
Alarm state: Any
Alarm type: Any
Actions status: Any
Name
State
Last state update
Conditions
Actions
BillExceed
Insufficient data
2024-02-06 16:58:26
EstimatedCharges >= 1 for 1 datapoints within 6 hours
Actions enabled
Warning

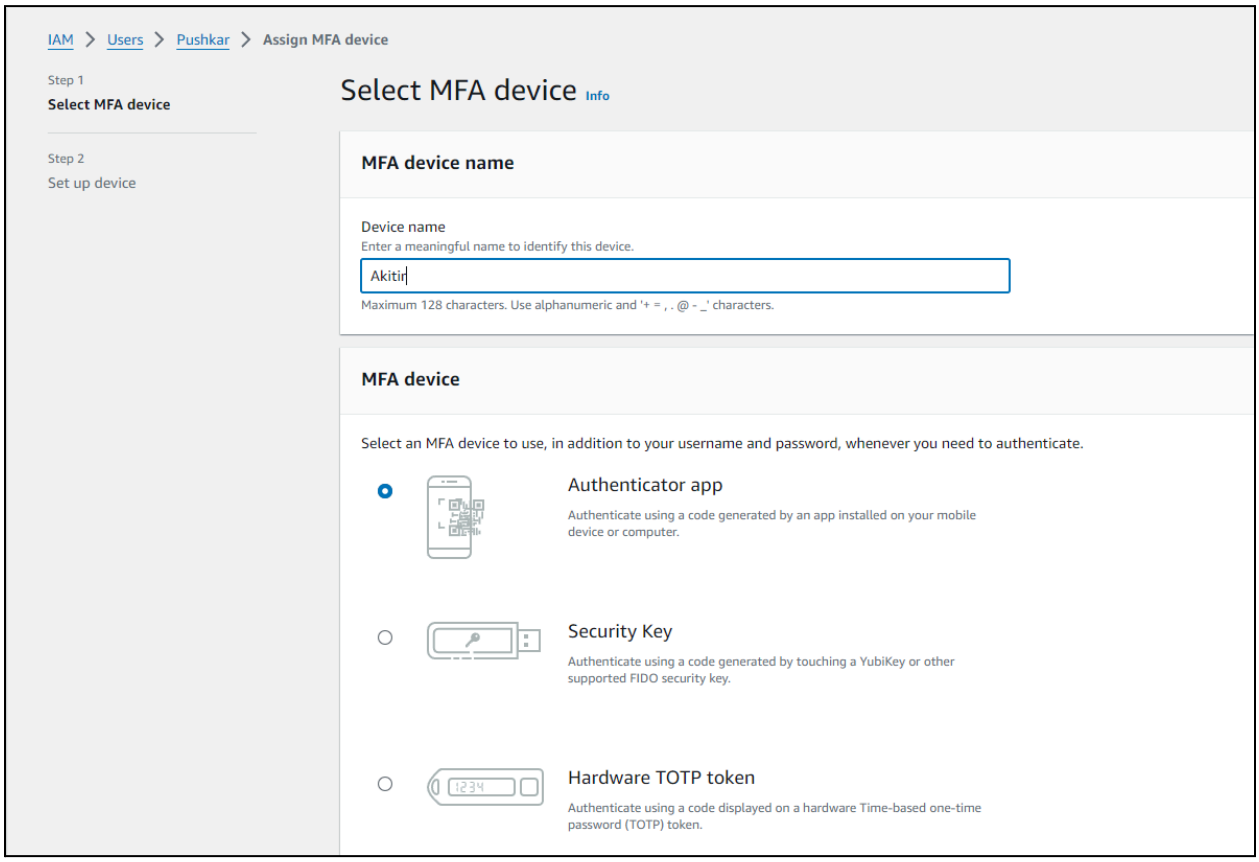
IAMUser:

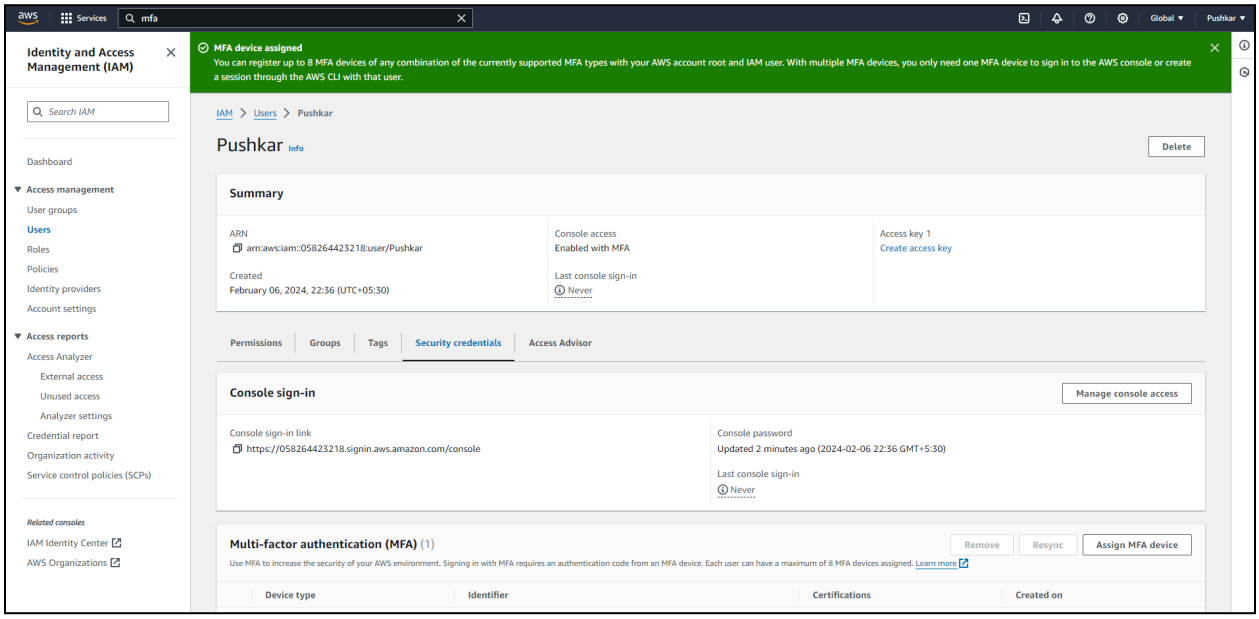

IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Specify user details
User details
User name
Pushkar
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)
Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.
Are you providing console access to a person?
User type
Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.
I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.
Console password
Autogenerated password
You can view the password after you create the user.
Custom password
Enter a custom password for the user.
Must be at least 8 characters long
Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # $ % ^ & * ( ) _ + - (hyphen) = [ ] { } | '
Show password
Users must create a new password at next sign-in - Recommended
Users automatically get the IAMUserChangePassword policy to allow them to change their own password.


IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Set permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.
Permissions policies (1/1175)
Choose one or more policies to attach to your new user.
Create policy
Filter by Type
Search
All types
Policy name
Type
Attached entities
AccessAnalyzerServiceRolePolicy AWS managed 0
AdministratorAccess AWS managed - job function 0
AdministratorAccess-Amplify AWS managed 0
AdministratorAccess-AWSElasticBeanstalk AWS managed 0
AlexaForBusinessDeviceSetup AWS managed 0
AlexaForBusinessFullAccess AWS managed 0
AlexaForBusinessGatewayExecution AWS managed 0
AlexaForBusinessLifesizeDelegatedAccessPolicy AWS managed 0


IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Review and create
Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.
User details
User name
Pushkar
Console password type
Autogenerated
Require password reset
Yes
Permissions summary
Name
Type
Used as
AdministratorAccess AWS managed - job function Permissions policy
IAMUserChangePassword AWS managed Permissions policy
Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Cancel
Previous
Create user


User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.
View user
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Retrieve password
You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.
Console sign-in details
Email sign-in instructions
Console sign-in URL
https://058264423218.signin.aws.amazon.com/console
User name
Pushkar
Console password
Show
Cancel
Download .csv file
Return to users list


IAM > Users > Pushkar > Assign MFA device
Step 1
Select MFA device
Step 2
Set up device
Select MFA device Info
MFA device name
Device name
Enter a meaningful name to identify this device.
Akitir
Maximum 128 characters. Use alphanumeric and '+ = , . @ - _' characters.
MFA device
Select an MFA device to use, in addition to your username and password, whenever you need to authenticate.
Authenticator app
Authenticate using a code generated by an app installed on your mobile device or computer.
Security Key
Authenticate using a code generated by touching a YubiKey or other supported FIDO security key.
Hardware TOTP token
Authenticate using a code displayed on a hardware Time-based one-time password (TOTP) token.


aws
Services
mfa
Global
Pushkar
Identity and Access Management (IAM)
MFA device assigned
You can register up to 8 MFA devices of any combination of the currently supported MFA types with your AWS account root and IAM user. With multiple MFA devices, you only need one MFA device to sign in to the AWS console or create a session through the AWS CLI with that user.
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies (SCPs)
Related consoles
IAM Identity Center
AWS Organizations
IAM > Users > Pushkar
Pushkar Info
Delete
Summary
ARN
arn:aws:iam::058264423218:user/Pushkar
Created
February 06, 2024, 22:36 (UTC+05:30)
Console access
Enabled with MFA
Last console sign-in
Never
Access key 1
Create access key
Permissions
Groups
Tags
Security credentials
Access Advisor
Console sign-in
Manage console access
Console sign-in link
https://058264423218.signin.aws.amazon.com/console
Console password
Updated 2 minutes ago (2024-02-06 22:36 GMT+5:30)
Last console sign-in
Never
Multi-factor authentication (MFA) (1)
Remove
Resync
Assign MFA device
Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more
Device type
Identifier
Certifications
Created on

AIML Components in AWS:

AWS provides a robust set of Artificial Intelligence and Machine Learning (AIML) services. These components include Amazon SageMaker for end-to-end machine learning workflows, Amazon Comprehend for natural language processing, Amazon Rekognition for image and video analysis, and more. These AIML services empower users to integrate powerful machine learning capabilities into their applications without the need for extensive expertise in the field.

Simple Storage Service (S3) Introduction:

Amazon S3 (Simple Storage Service) is a scalable object storage service in AWS, designed to store and retrieve any amount of data from anywhere on the web. S3 offers industry-leading durability, availability, and scalability, making it ideal for diverse storage needs. Users can organize data in buckets, control access through flexible permissions, and leverage features like versioning and lifecycle policies.

Elastic Computing Cloud (EC2) Introduction:

Amazon EC2 (Elastic Compute Cloud) forms the backbone of AWS's compute services, offering resizable virtual servers in the cloud. Users can choose from a variety of instance types based on their specific computational requirements. EC2 instances provide flexibility, scalability, and control over computing resources, enabling users to run applications efficiently and cost-effectively in the AWS cloud.

Conclusion: In conclusion, this practical provided an introduction to essential aspects of Amazon Web Services (AWS), beginning with the sign-up process and familiarization with service legal agreements. It then delved into Amazon SageMaker, emphasizing its ease of access, free tier availability, billing structure, and integration with AWS Identity and Access Management (IAM). Additionally, the practical covered key components of artificial intelligence and machine learning (AIML) within AWS, followed by an overview of fundamental services like Simple Storage Service (S3) and Elastic Compute Cloud (EC2). These foundational concepts equip users with the knowledge necessary to leverage AWS effectively for their computing and machine learning needs.

Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 3, 4**	
Title of Lab Assignment: Introduction to AWS SageMaker, Data Labeling in AWS Sage maker Ground Truth, Labeling Text, Bounding Boxes and Semantic Segmentation in Ground Truth.		
DOP: 14-02-2024	**DOS: 22-02-2024**	
CO Mapped:	**PO Mapped:**	**Signature:**

Practical No. 3 & 4

Aim: Introduction to AWS SageMaker, Data Labeling in AWS Sage maker Ground Truth, Labeling Text, Bounding Boxes and Semantic Segmentation in Ground Truth.

Theory:

Amazon SageMaker is a fully managed service by AWS that simplifies the machine learning workflow, from data preparation to model deployment. It provides a range of tools and services for building, training, and deploying machine learning models at scale. With SageMaker, developers and data scientists can efficiently experiment with different algorithms, automate model training with Autopilot, and deploy models with ease using managed endpoints. SageMaker also offers capabilities for data labeling, model monitoring, and governance, making it a comprehensive platform for AI development.

Components of AWS Sagemaker:

1. **Jump Start:**
    - Amazon SageMaker JumpStart provides access to popular machine learning algorithms and model templates.
    - Amazon SageMaker JumpStart provides access to pre-built machine learning models and solutions, accelerating the model development process.
    - It offers a curated selection of algorithms and model templates, allowing users to quickly get started with common machine learning tasks without the need for extensive setup or configuration.

      To use JumpStart:
    - Log in to the AWS Management Console.
    - Navigate to Amazon SageMaker.
    - Choose "JumpStart" from the left-hand menu.
    - Select the algorithm or model template that fits your needs.
    - Follow the provided instructions to configure and deploy the chosen model.

2. **Ground Truth:**
    - Amazon SageMaker Ground Truth helps label data for machine learning models efficiently.

- Amazon SageMaker Ground Truth is a data labeling service that makes it easy to annotate datasets for training machine learning models.
- Ground Truth offers both human annotation and automated data labeling capabilities, helping users generate high-quality labeled datasets efficiently.

Steps to use Ground Truth:

- Log in to the AWS Management Console.
- Navigate to Amazon SageMaker.
- Choose "Ground Truth" from the left-hand menu.
- Create a labeling job by specifying the input data, labeling categories, and workforce.
- Launch the labeling job and monitor its progress.
- Once labeling is complete, use the labeled data for training your model.

3. **Governance:**
   - SageMaker provides governance features to manage access, security, and compliance of machine learning workflows.
   - SageMaker Governance features enable users to manage access, security, and compliance for machine learning workflows.
   - It provides tools for defining IAM policies, auditing model activities, implementing encryption, and enforcing data privacy and compliance requirements.

   Steps for governance:

   - Navigate to SageMaker in the AWS Management Console.
   - Set up IAM roles and policies to control access to SageMaker resources.
   - Use AWS CloudTrail to monitor API activity.
   - Implement encryption for data at rest and in transit.
   - Configure VPC settings for SageMaker resources to control network access.

4. **Notebook:**
   - Amazon SageMaker Notebooks provide a fully managed Jupyter notebook instance for building and testing models.
   - SageMaker Notebook Instances provide a fully managed Jupyter notebook environment for building, experimenting, and collaborating on machine learning projects.

- Users can leverage built-in libraries and frameworks, scale compute resources as needed, and integrate with other SageMaker components seamlessly.

To use SageMaker Notebooks:

- Navigate to SageMaker in the AWS Management Console.
- Choose "Notebook instances" from the left-hand menu.
- Create a new notebook instance, selecting the instance type and IAM role.
- Open the Jupyter notebook interface and start writing code.
- Save your work to Amazon S3 or GitHub for version control.

5. **Processing:**
   - SageMaker Processing allows you to preprocess data and run custom data transformations at scale.
   - SageMaker Processing allows users to preprocess and analyze data at scale before training machine learning models.
   - It supports custom data transformations, parallel execution of processing tasks, and integration with other AWS services for data storage and retrieval.

   Steps for processing:

   - Navigate to SageMaker in the AWS Management Console.
   - Choose "Processing jobs" from the left-hand menu.
   - Create a processing job, specifying the input data, processing script, and output location.
   - Monitor the status of the processing job and view logs for debugging if needed.

6. **Training:**
   - SageMaker Training enables you to train machine learning models using built-in algorithms or custom scripts.
   - SageMaker Training enables users to train machine learning models using built-in algorithms, custom code, or frameworks like TensorFlow and PyTorch.
   - It offers distributed training capabilities, automatic model tuning, and support for training on large datasets stored in Amazon S3.

   Steps for training:

   - Navigate to SageMaker in the AWS Management Console.
   - Choose "Training jobs" from the left-hand menu.

- Create a training job, specifying the algorithm or script, input data location, and output location.
- Configure hyperparameters and instance types for training.
- Monitor the training job's progress and view logs for insights.

7. **Inference:**
   - Amazon SageMaker Inference allows you to deploy trained models for real-time or batch inference.
   - SageMaker Inference enables users to deploy trained models for real-time or batch inference, serving predictions at scale with low latency.
   - It provides managed endpoints, automatic scaling, and integration with AWS Lambda and API Gateway for building scalable inference pipelines.

   To deploy a model for inference:
   - Navigate to SageMaker in the AWS Management Console.
   - Choose "Endpoints" from the left-hand menu.
   - Create a new endpoint, specifying the trained model, instance type, and number of instances.
   - Once the endpoint is created, you can use it to make predictions by sending HTTP requests.

8. **Augmented AI:**
   - SageMaker Augmented AI (A2I) helps you build human review workflows for ML predictions to improve model accuracy.
   - SageMaker Augmented AI (A2I) helps improve the accuracy of machine learning models by integrating human review into the prediction process.
   - It allows users to create custom review workflows, route predictions to human reviewers, and incorporate feedback to continuously improve model performance

   Steps for AI:
   - Navigate to SageMaker in the AWS Management Console.
   - Choose "Ground Truth" from the left-hand menu.
   - Create a human review workflow by specifying the conditions for sending predictions to human reviewers.
   - Configure the workforce and define the instructions for reviewers.

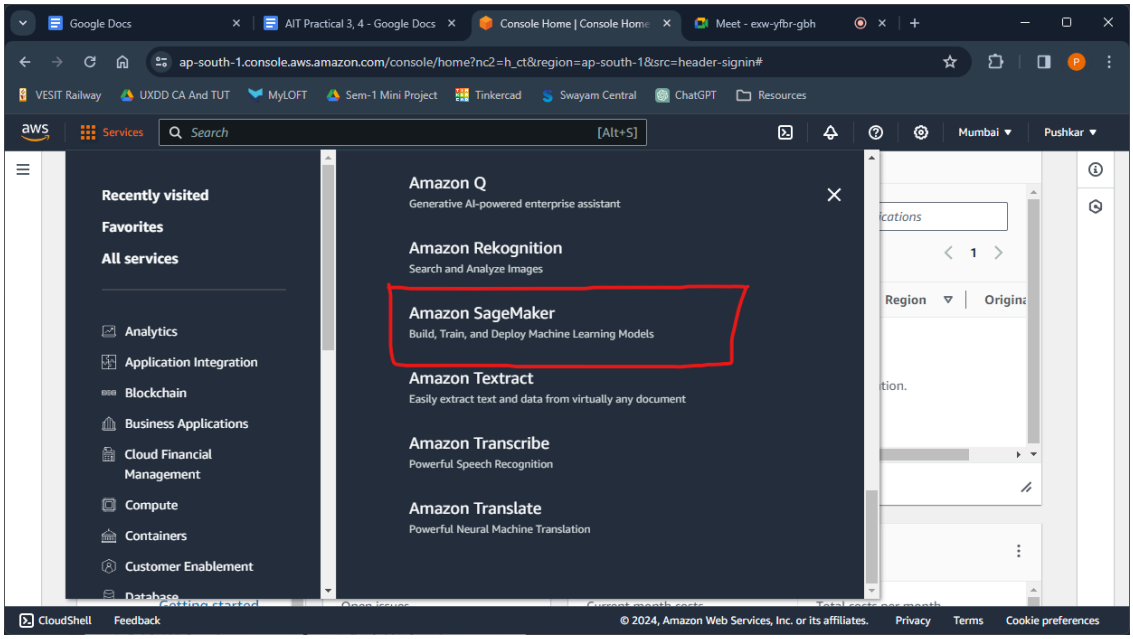
- Monitor the human review process and incorporate feedback to improve model performance.

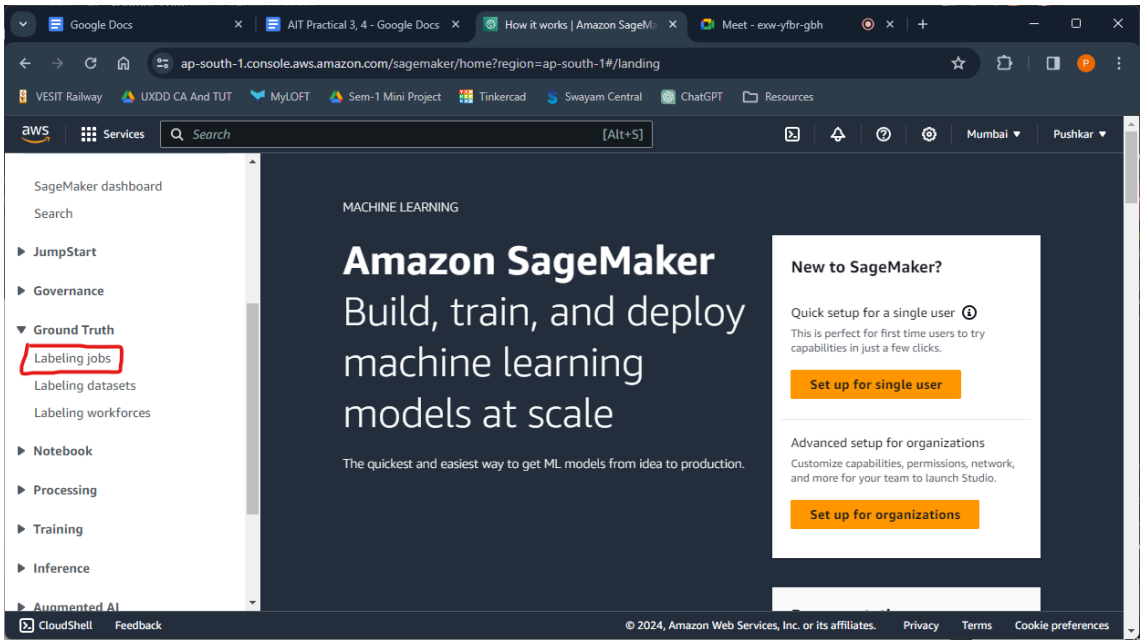
9. **AWS Marketplace (GluonCV YOLO v3 object detector):**
    - The AWS Marketplace offers pre-built machine learning models and algorithms for easy integration into SageMaker.
    - The AWS Marketplace offers a wide range of pre-built machine learning models and algorithms, including the GluonCV YOLO v3 object detector.
    - Users can easily deploy these models in SageMaker for tasks like object detection, image classification, and natural language processing, accelerating the development of AI applications.

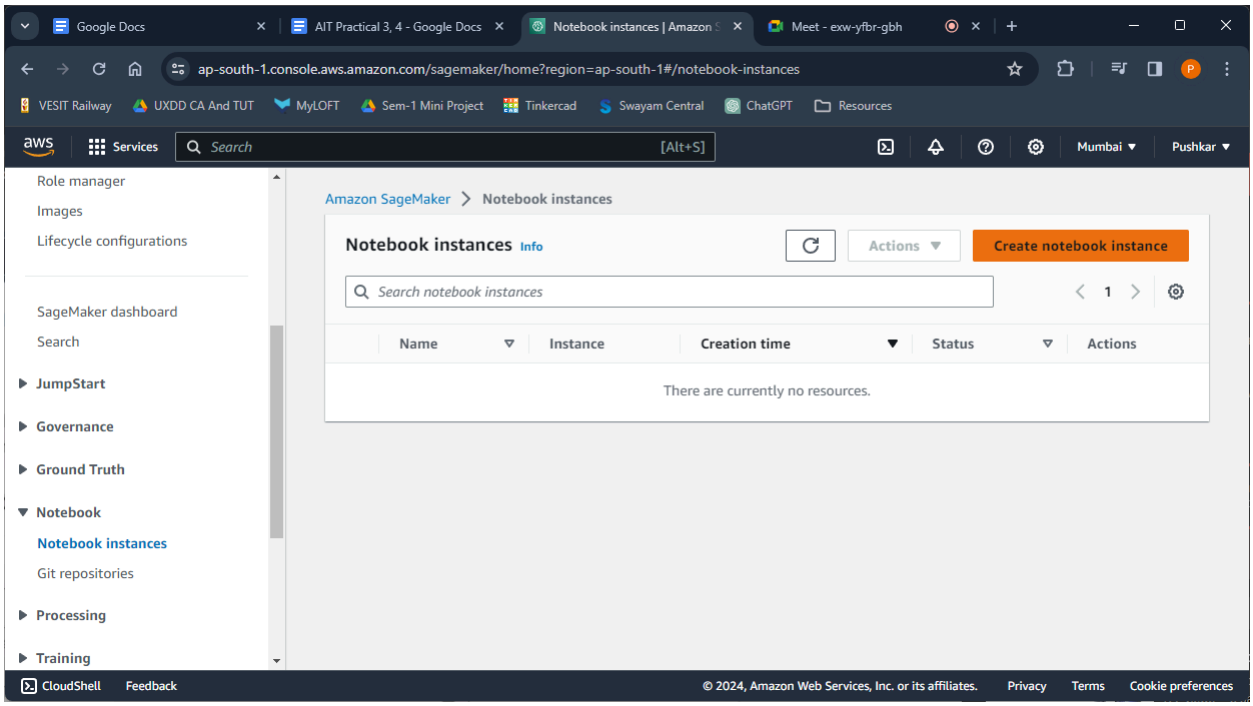
      To use GluonCV YOLO v3 object detector:
    - Navigate to the AWS Marketplace in the AWS Management Console.
    - Search for "GluonCV YOLO v3 object detector" and subscribe to the listing.
    - Follow the provided instructions to deploy the model in SageMaker.
    - Once deployed, you can use the model for object detection tasks by sending input images to the endpoint.

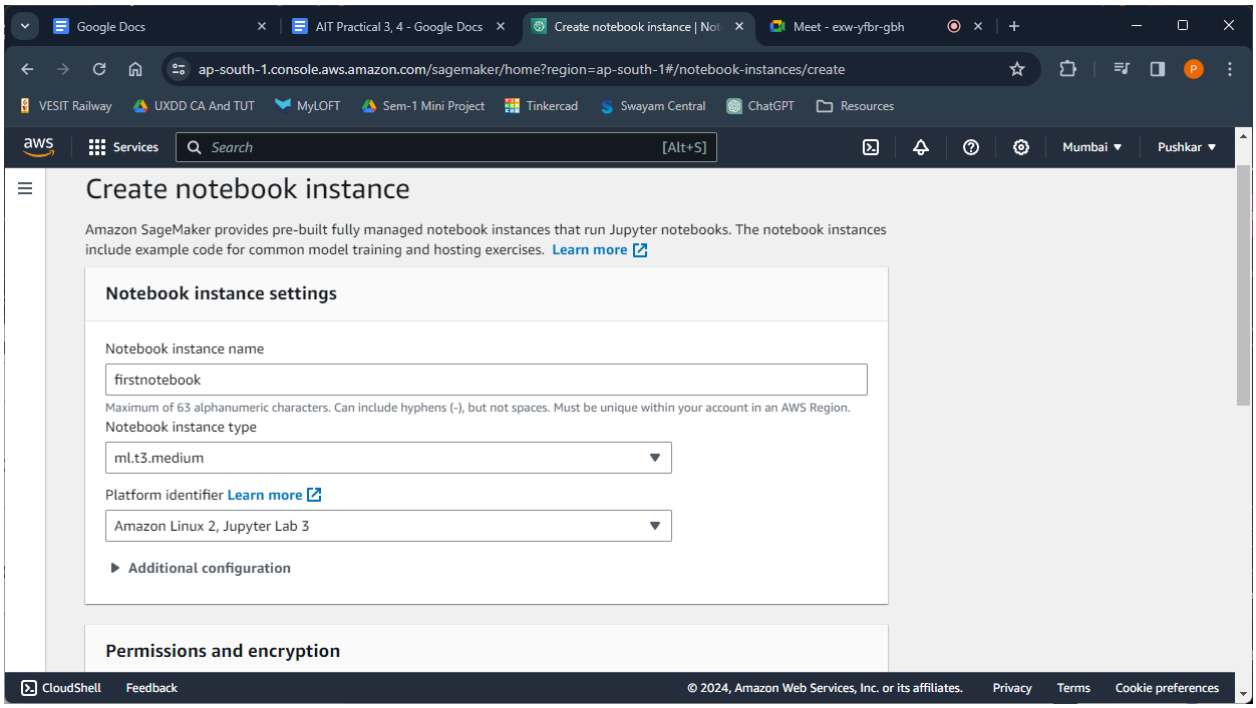
Remember to manage costs effectively by monitoring resource usage and shutting down instances when not in use. Additionally, ensure compliance with data privacy regulations when handling sensitive data.

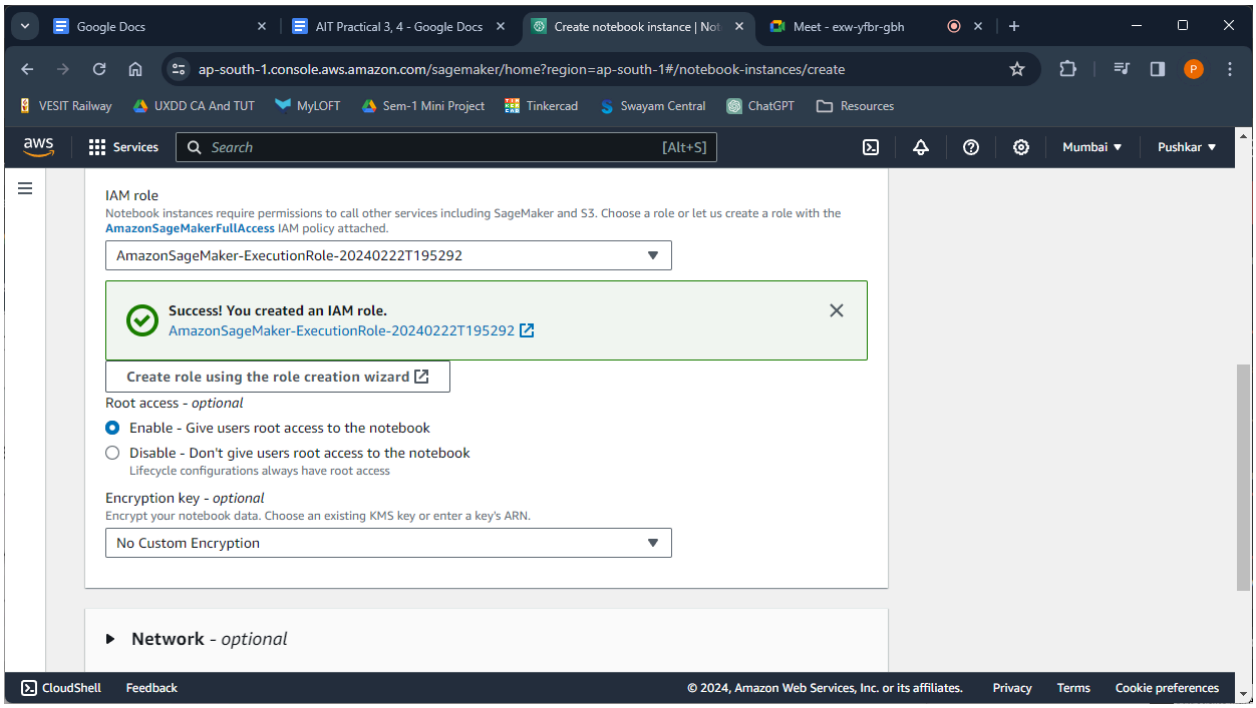
Steps to explore the AWS sagemaker:

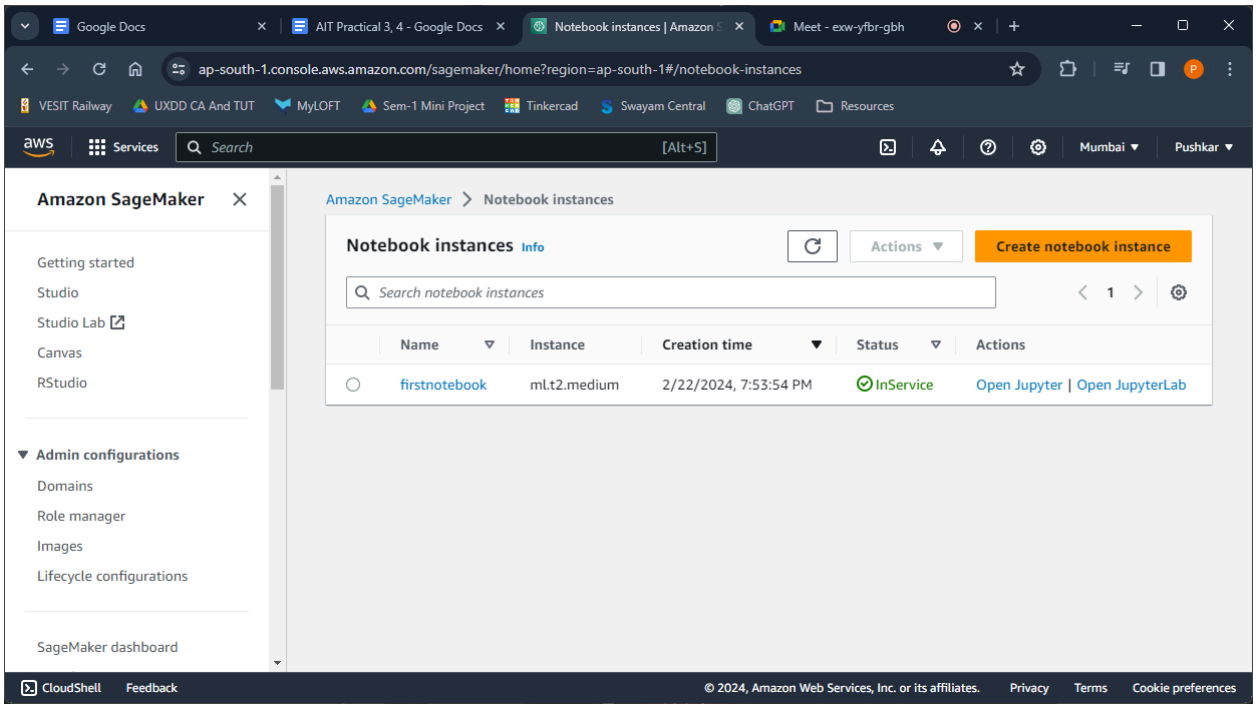


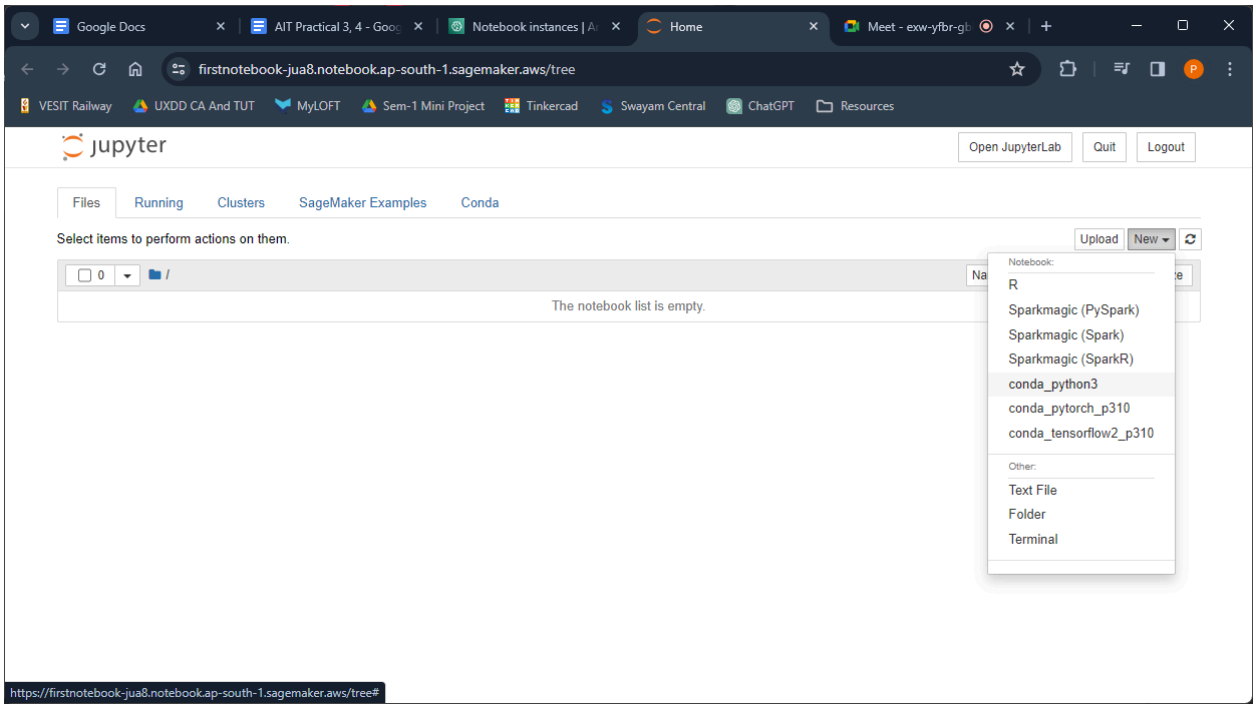


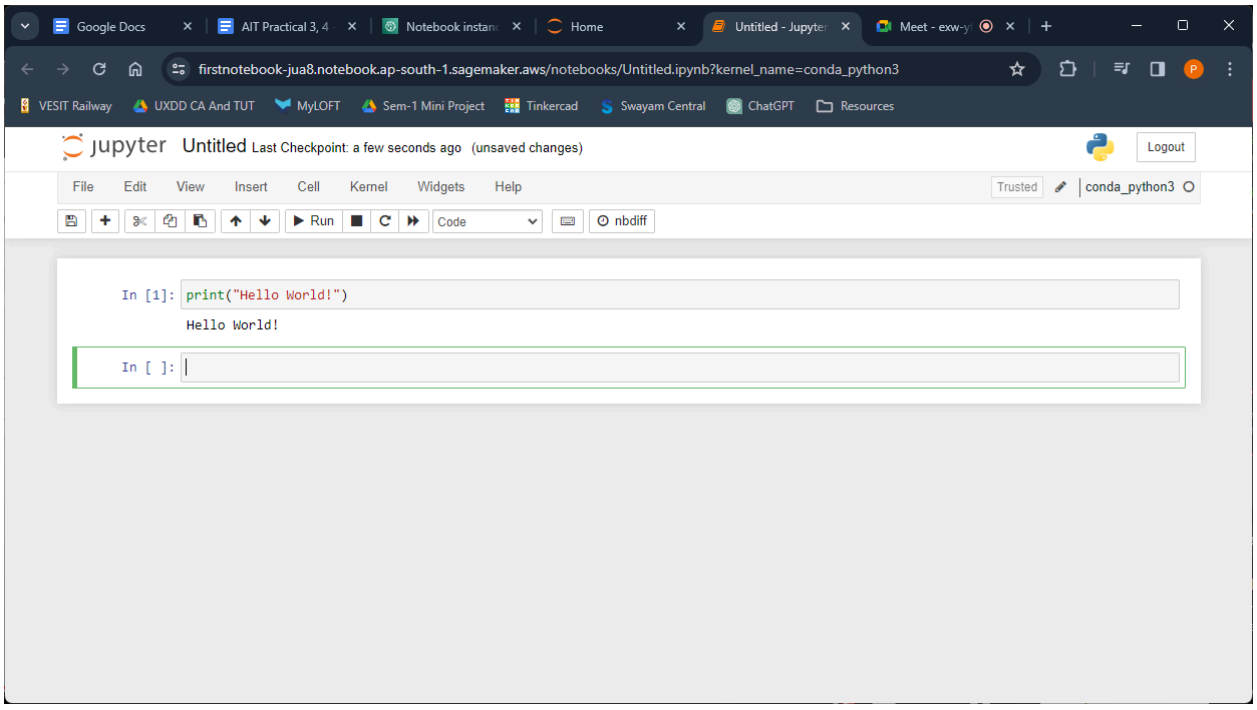

Google Docs
AIT Practical 3, 4 - Google Docs
Notebook instances | Amazon
Meet - exw-yfbr-gbh
ap-south-1.console.aws.amazon.com/sagemaker/home?region=ap-south-1#/notebook-instances
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
aws
Services
Search
[Alt+S]
Mumbai
Pushkar
Role manager
Images
Lifecycle configurations
SageMaker dashboard
Search
JumpStart
Governance
Ground Truth
Notebook
Notebook instances
Git repositories
Processing
Training
Amazon SageMaker > Notebook instances
Notebook instances Info
Actions
Create notebook instance
Search notebook instances
Name
Instance
Creation time
Status
Actions
There are currently no resources.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

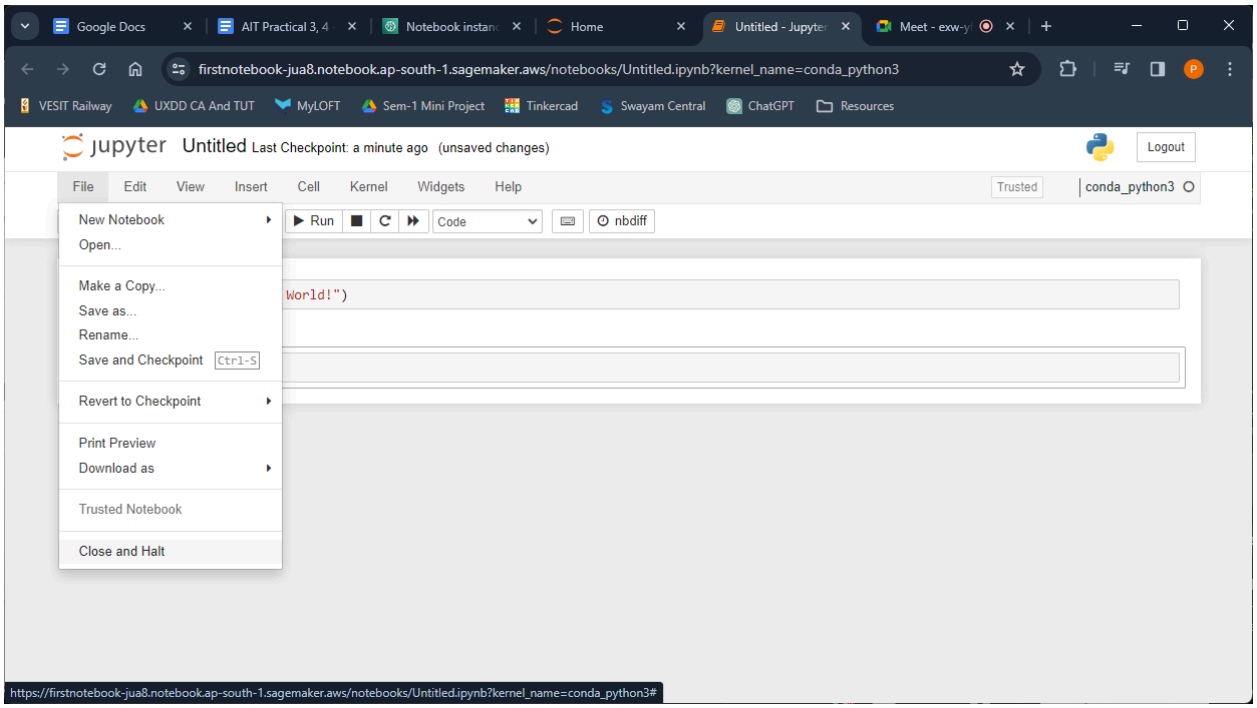

Google Docs
AIT Practical 3, 4 - Google Docs
Create notebook instance | Not
Meet - exw-yfbr-gbh
ap-south-1.console.aws.amazon.com/sagemaker/home?region=ap-south-1#/notebook-instances/create
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
aws
Services
Search
[Alt+S]
Mumbai
Pushkar
Create notebook instance
Amazon SageMaker provides pre-built fully managed notebook instances that run Jupyter notebooks. The notebook instances include example code for common model training and hosting exercises. Learn more
Notebook instance settings
Notebook instance name
firstnotebook
Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.
Notebook instance type
ml.t3.medium
Platform identifier Learn more
Amazon Linux 2, Jupyter Lab 3
Additional configuration
Permissions and encryption
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

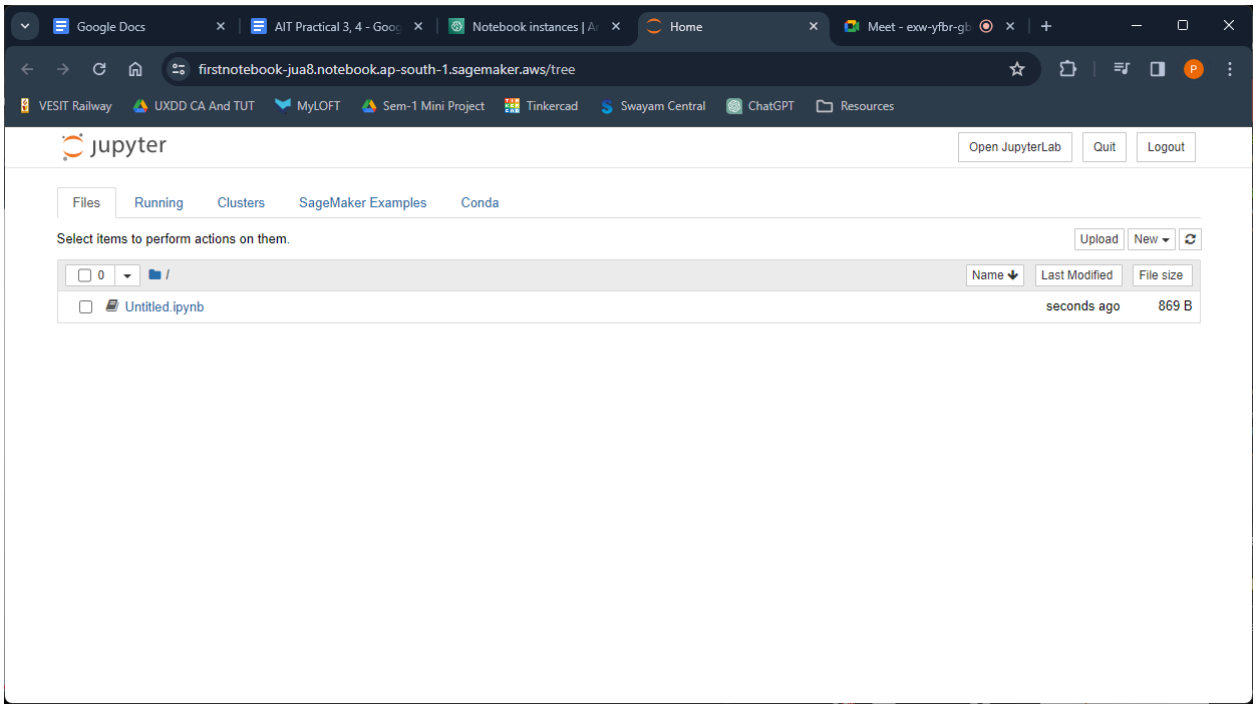

Google Docs
AIT Practical 3, 4 - Google Docs
Create notebook instance | No
Meet - exw-yfbr-gbh
ap-south-1.console.aws.amazon.com/sagemaker/home?region=ap-south-1#/notebook-instances/create
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
aws
Services
Search
[Alt+S]
Mumbai
Pushkar
IAM role
Notebook instances require permissions to call other services including SageMaker and S3. Choose a role or let us create a role with the AmazonSageMakerFullAccess IAM policy attached.
AmazonSageMaker-ExecutionRole-20240222T195292
Success! You created an IAM role.
AmazonSageMaker-ExecutionRole-20240222T195292
Create role using the role creation wizard
Root access - optional
Enable - Give users root access to the notebook
Disable - Don't give users root access to the notebook
Lifecycle configurations always have root access
Encryption key - optional
Encrypt your notebook data. Choose an existing KMS key or enter a key's ARN.
No Custom Encryption
Network - optional
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

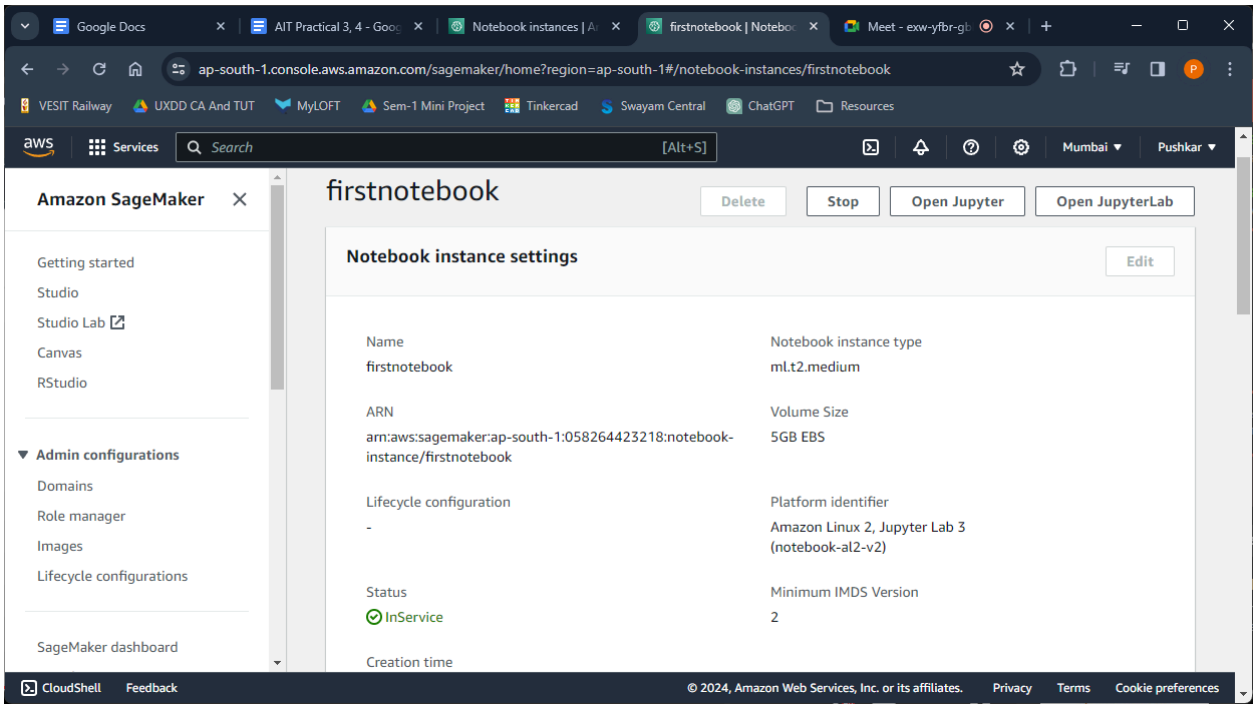

Google Docs
AIT Practical 3, 4 - Google Docs
Notebook instances | Amazon
Meet - exw-yfbr-gbh
ap-south-1.console.aws.amazon.com/sagemaker/home?region=ap-south-1#/notebook-instances
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
aws
Services
Search
[Alt+S]
Mumbai
Pushkar
Amazon SageMaker
Getting started
Studio
Studio Lab
Canvas
RStudio
Admin configurations
Domains
Role manager
Images
Lifecycle configurations
SageMaker dashboard
Amazon SageMaker > Notebook instances
Notebook instances Info
Actions
Create notebook instance
Search notebook instances
1
Name
Instance
Creation time
Status
Actions
firstnotebook
ml.t2.medium
2/22/2024, 7:53:54 PM
InService
Open Jupyter | Open JupyterLab
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

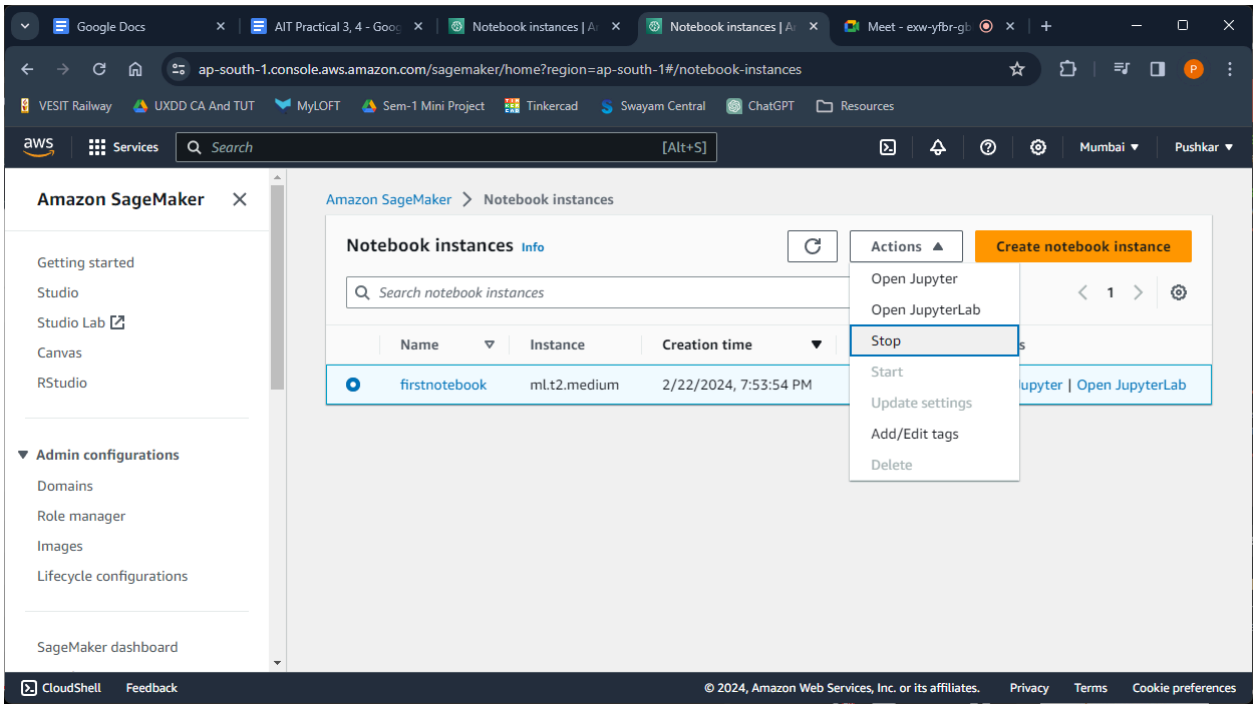

Google Docs
AIT Practical 3, 4 - Goo
Notebook instances | A
Home
Meet - exw-yfbr-gb
firstnotebook-jua8.notebook.ap-south-1.sagemaker.aws/tree
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
jupyter
Open JupyterLab
Quit
Logout
Files
Running
Clusters
SageMaker Examples
Conda
Select items to perform actions on them.
Upload
New
The notebook list is empty.
Notebook:
R
Sparkmagic (PySpark)
Sparkmagic (Spark)
Sparkmagic (SparkR)
conda_python3
conda_pytorch_p310
conda_tensorflow2_p310
Other:
Text File
Folder
Terminal
https://firstnotebook-jua8.notebook.ap-south-1.sagemaker.aws/tree#

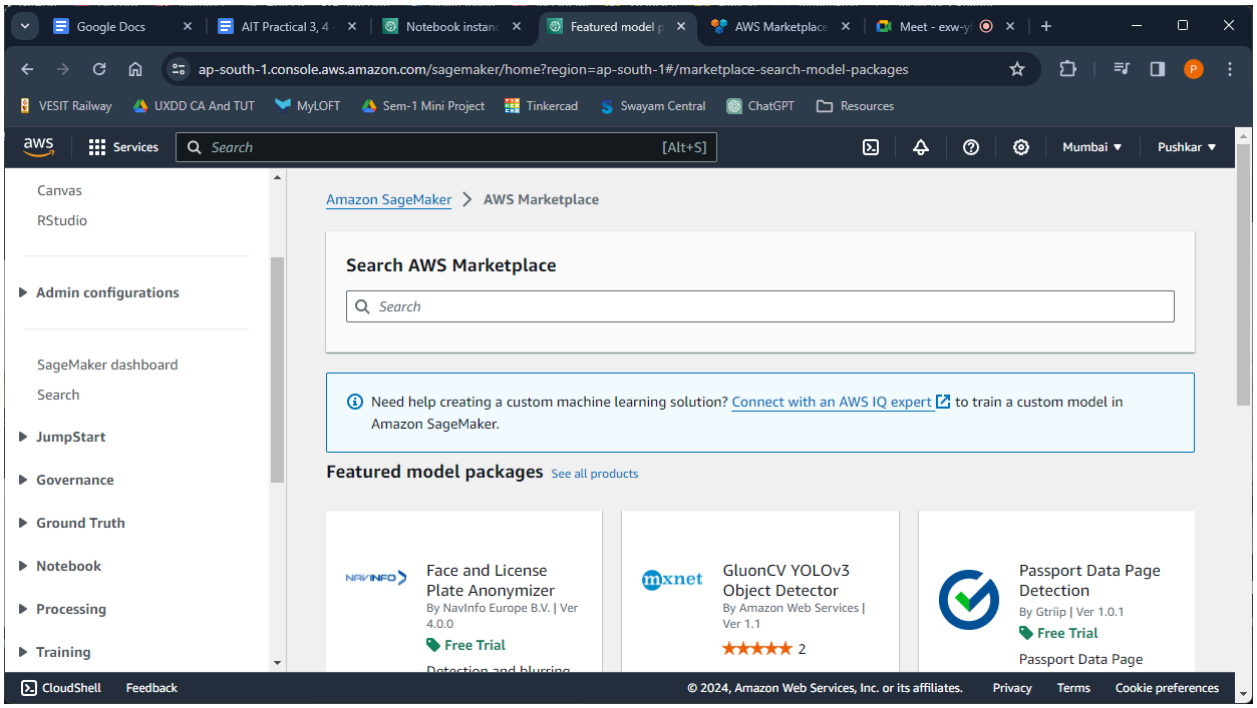

Google Docs
AIT Practical 3, 4
Notebook instanc
Home
Untitled - Jupyter
Meet - exw-y
firstnotebook-jua8.notebook.ap-south-1.sagemaker.aws/notebooks/Untitled.ipynb?kernel_name=conda_python3
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
jupyter
Untitled
Last Checkpoint: a few seconds ago
(unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
conda_python3
Run
Code
nbdiff
In [1]:
print("Hello World!")
Hello World!
In [ ]:

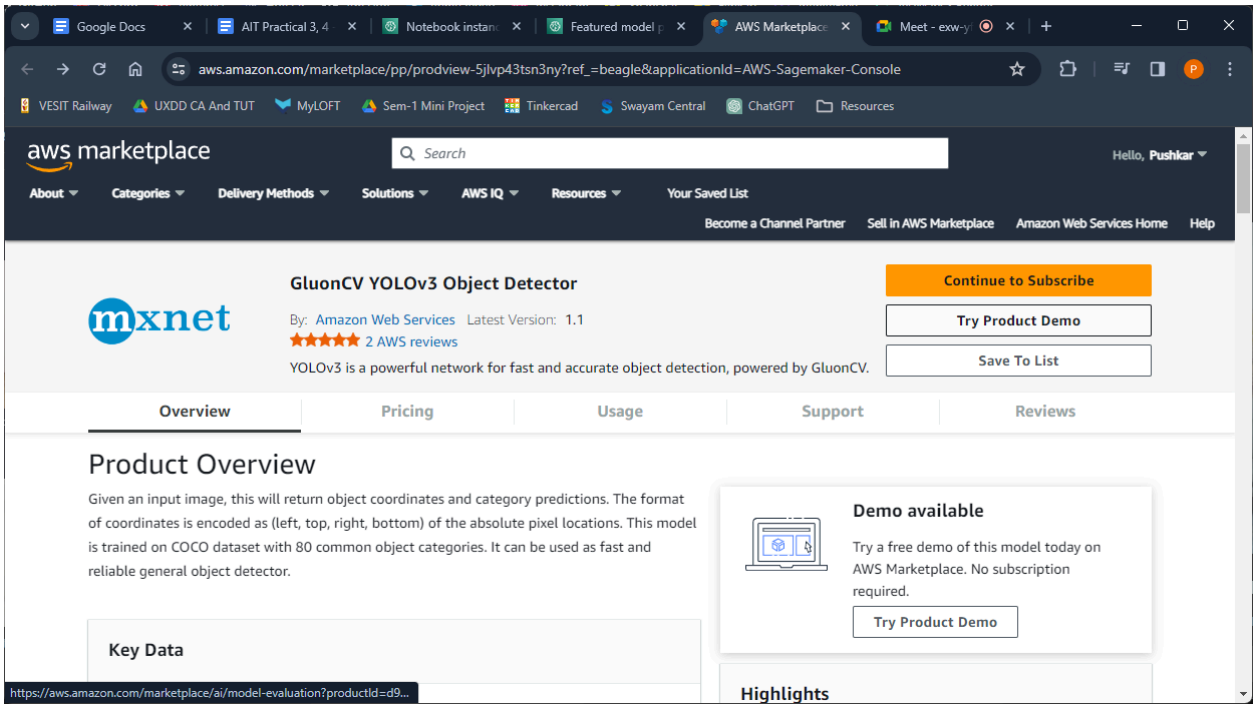

Google Docs
AIT Practical 3, 4
Notebook instanc
Home
Untitled - Jupyter
Meet - exw-y
firstnotebook-jua8.notebook.ap-south-1.sagemaker.aws/notebooks/Untitled.ipynb?kernel_name=conda_python3
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
jupyter Untitled Last Checkpoint: a minute ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
conda_python3
New Notebook
Open...
Make a Copy...
Save as...
Rename...
Save and Checkpoint Ctrl-S
Revert to Checkpoint
Print Preview
Download as
Trusted Notebook
Close and Halt
Run
Code
nbdiff
World!")
https://firstnotebook-jua8.notebook.ap-south-1.sagemaker.aws/notebooks/Untitled.ipynb?kernel_name=conda_python3#

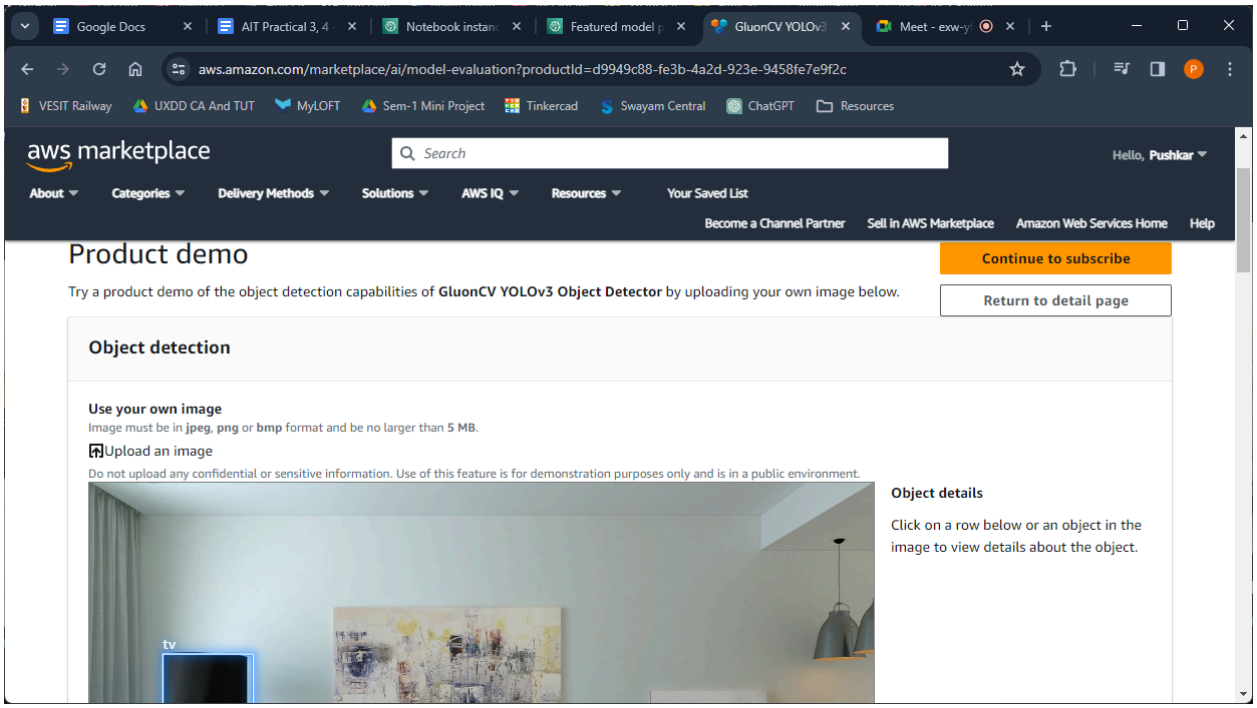

Google Docs
AIT Practical 3, 4 - Goog
Notebook instances | A
Home
Meet - exw-yfbr-gb
firstnotebook-jua8.notebook.ap-south-1.sagemaker.aws/tree
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
jupyter
Open JupyterLab
Quit
Logout
Files
Running
Clusters
SageMaker Examples
Conda
Select items to perform actions on them.
Upload
New
Name
Last Modified
File size
Untitled.ipynb
seconds ago
869 B

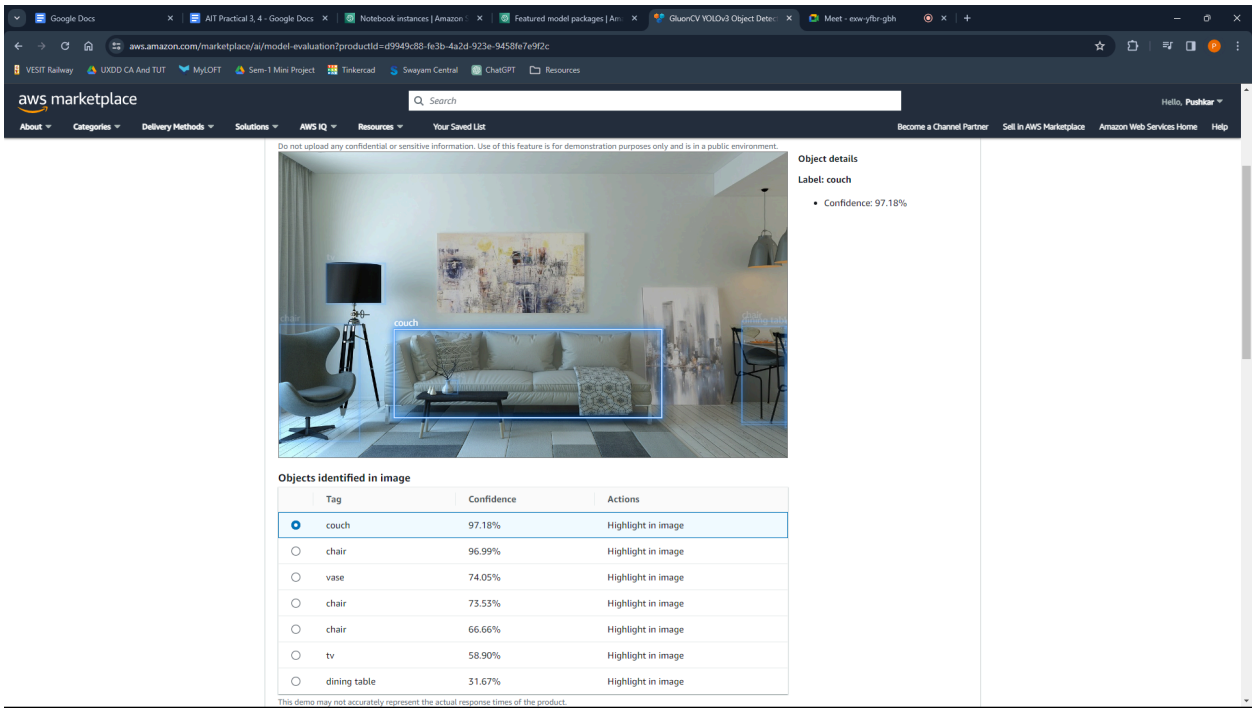

Google Docs
AIT Practical 3, 4 - Goog
Notebook instances | Ar
firstnotebook | Notebo
Meet - exw-yfbr-gb
ap-south-1.console.aws.amazon.com/sagemaker/home?region=ap-south-1#/notebook-instances/firstnotebook
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
aws
Services
Search
[Alt+S]
Mumbai
Pushkar
Amazon SageMaker
Getting started
Studio
Studio Lab
Canvas
RStudio
Admin configurations
Domains
Role manager
Images
Lifecycle configurations
SageMaker dashboard
firstnotebook
Delete
Stop
Open Jupyter
Open JupyterLab
Notebook instance settings
Edit
Name
firstnotebook
Notebook instance type
ml.t2.medium
ARN
arn:aws:sagemaker:ap-south-1:058264423218:notebook-instance/firstnotebook
Volume Size
5GB EBS
Lifecycle configuration
-
Platform identifier
Amazon Linux 2, Jupyter Lab 3 (notebook-al2-v2)
Status
InService
Minimum IMDS Version
2
Creation time
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Google Docs
AIT Practical 3, 4 - Goog
Notebook instances | Ar
Notebook instances | Ar
Meet - exw-yfbr-gb
ap-south-1.console.aws.amazon.com/sagemaker/home?region=ap-south-1#/notebook-instances
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
aws
Services
Search
[Alt+S]
Mumbai
Pushkar
Amazon SageMaker
Getting started
Studio
Studio Lab
Canvas
RStudio
Admin configurations
Domains
Role manager
Images
Lifecycle configurations
SageMaker dashboard
Amazon SageMaker > Notebook instances
Notebook instances Info
Actions
Create notebook instance
Search notebook instances
Open Jupyter
Open JupyterLab
Stop
Start
Update settings
Add/Edit tags
Delete
Name
Instance
Creation time
firstnotebook
ml.t2.medium
2/22/2024, 7:53:54 PM
upyter | Open JupyterLab
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Google Docs
AIT Practical 3, 4
Notebook instanc
Featured model
AWS Marketplace
Meet - exw-y
ap-south-1.console.aws.amazon.com/sagemaker/home?region=ap-south-1#/marketplace-search-model-packages
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
aws
Services
Search
[Alt+S]
Mumbai
Pushkar
Canvas
RStudio
Admin configurations
SageMaker dashboard
Search
JumpStart
Governance
Ground Truth
Notebook
Processing
Training
Amazon SageMaker > AWS Marketplace
Search AWS Marketplace
Search
Need help creating a custom machine learning solution? Connect with an AWS IQ expert to train a custom model in Amazon SageMaker.
Featured model packages See all products
Face and License Plate Anonymizer
By NavInfo Europe B.V. | Ver 4.0.0
Free Trial
GluonCV YOLOv3 Object Detector
By Amazon Web Services | Ver 1.1
2
Passport Data Page Detection
By Gtriip | Ver 1.0.1
Free Trial
Passport Data Page
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Google Docs
AIT Practical 3, 4
Notebook instanc
Featured model
AWS Marketplace
Meet - exw-y
aws.amazon.com/marketplace/pp/prodview-5jlvp43tsn3ny?ref_=beagle&applicationId=AWS-Sagemaker-Console
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
aws marketplace
Search
Hello, Pushkar
About
Categories
Delivery Methods
Solutions
AWS IQ
Resources
Your Saved List
Become a Channel Partner
Sell in AWS Marketplace
Amazon Web Services Home
Help
GluonCV YOLOv3 Object Detector
By: Amazon Web Services Latest Version: 1.1
2 AWS reviews
YOLOv3 is a powerful network for fast and accurate object detection, powered by GluonCV.
Continue to Subscribe
Try Product Demo
Save To List
Overview
Pricing
Usage
Support
Reviews
Product Overview
Given an input image, this will return object coordinates and category predictions. The format of coordinates is encoded as (left, top, right, bottom) of the absolute pixel locations. This model is trained on COCO dataset with 80 common object categories. It can be used as fast and reliable general object detector.
Demo available
Try a free demo of this model today on AWS Marketplace. No subscription required.
Try Product Demo
Key Data
Highlights
https://aws.amazon.com/marketplace/ai/model-evaluation?productId=d9...





Conclusion: Successfully explored all the components of the AWS Sagemaker.

Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 5 & 6**	
Title of Lab Assignment: Feature Engineering, one hot Encoding, Normalization, Standardization, EDA using SageMaker DataWrangler Linear and Multiple Linear Regression using SageMaker.		
DOP: 16-03-2024	**DOS: 16-03-2024**	
CO Mapped: **CO3**	**PO Mapped:** **PO2, PO3, PO4, PO5, PO6, PO7, PSO1, PSO2**	**Signature:**

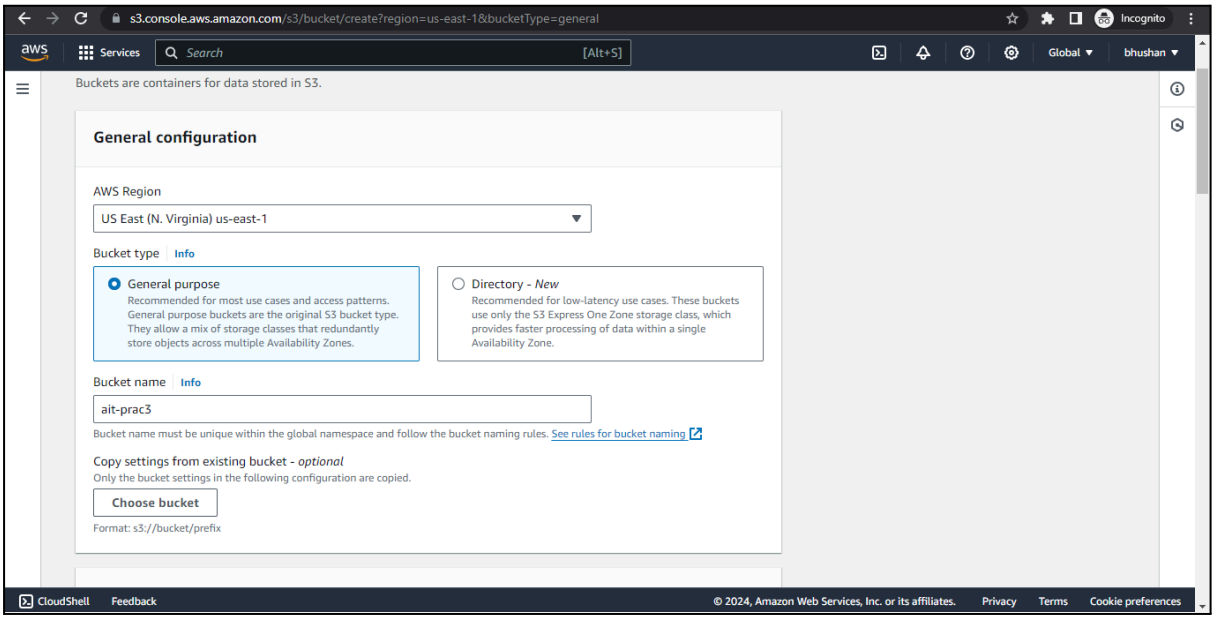
Practical 5 & 6

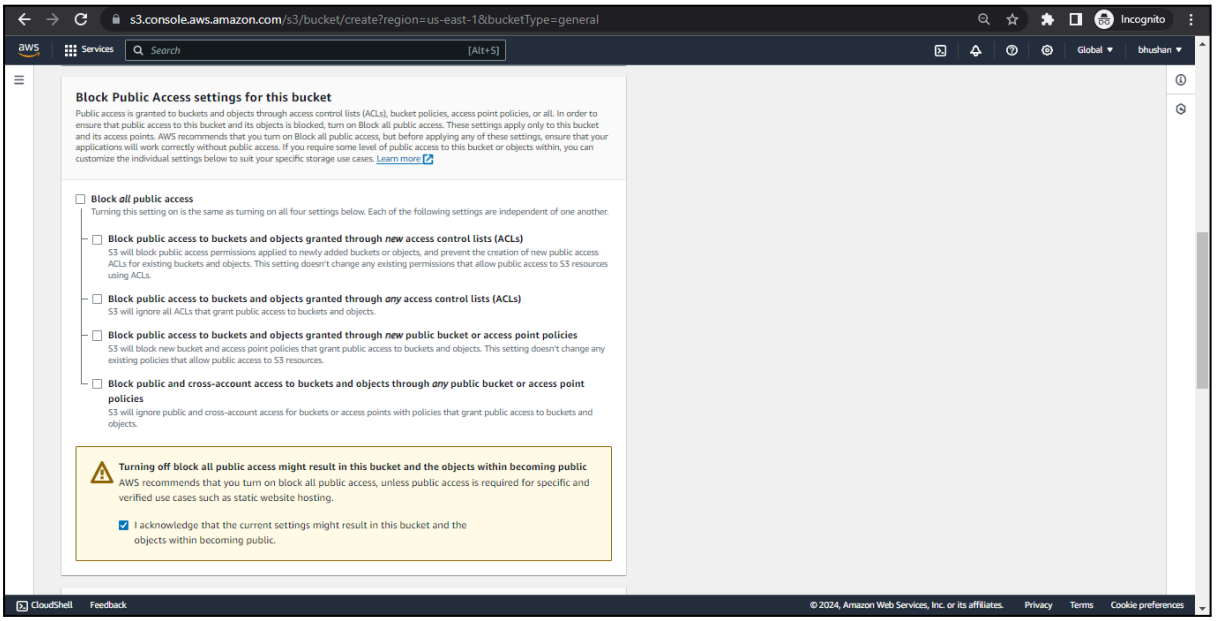
Aim: Feature Engineering, one hot Encoding, Normalization, Standardization, EDA using SageMaker DataWrangler Linear and Multiple Linear Regression using SageMaker.

Description:

Prerequisite

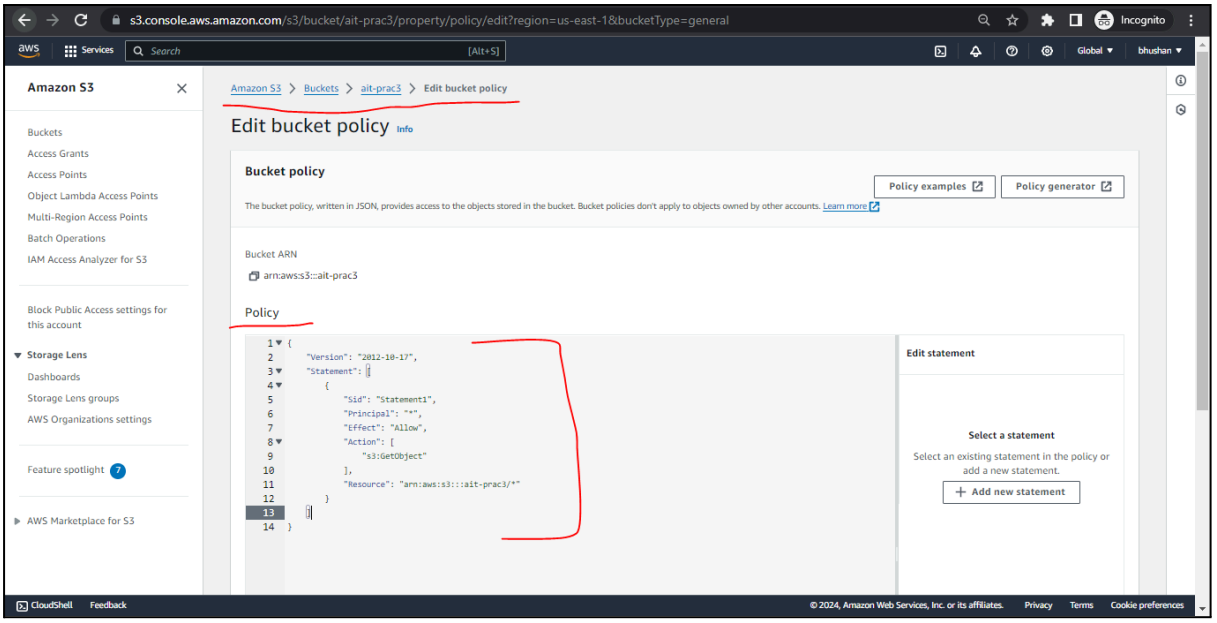
Create an S3 bucket and keep aside (it will be used later)



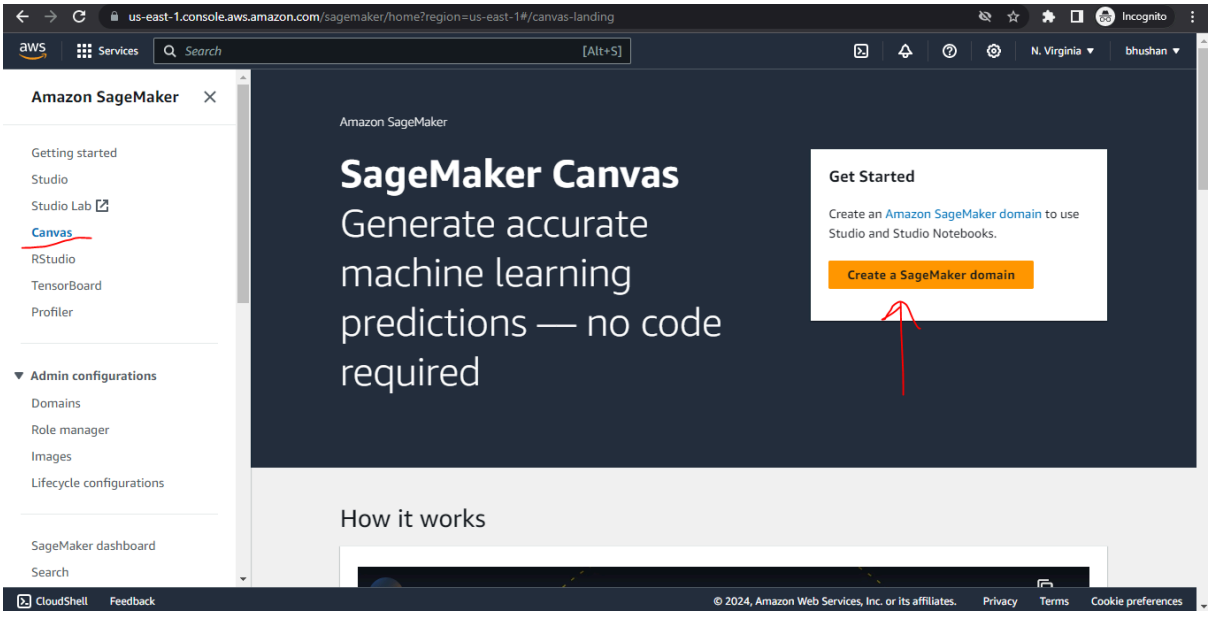


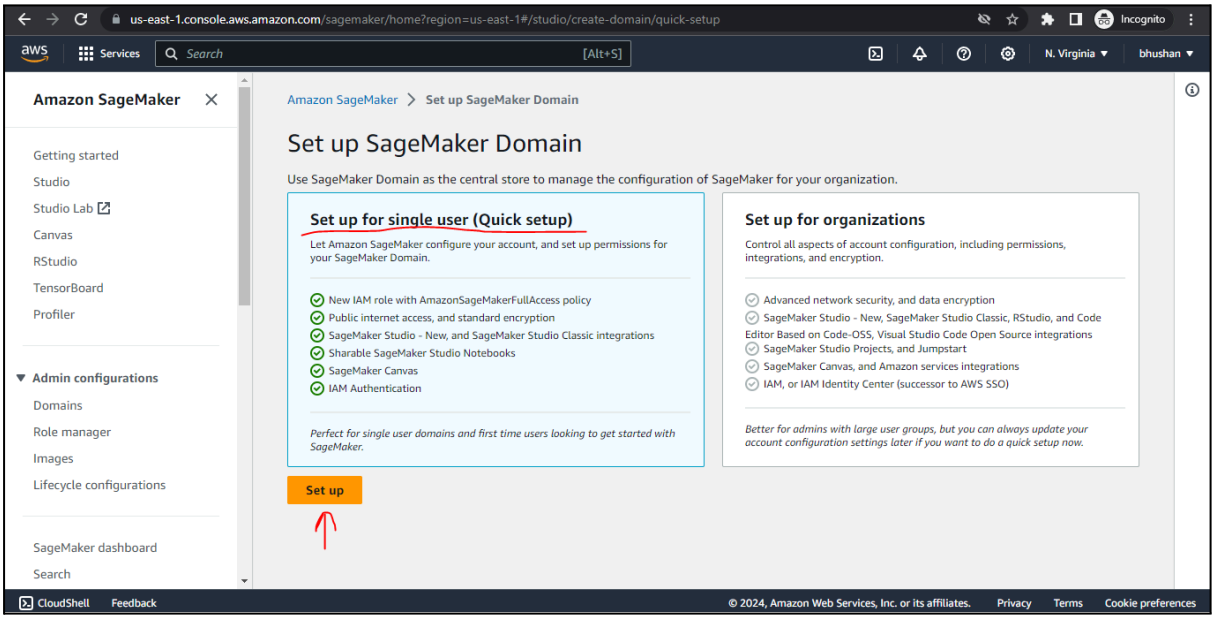
Click on “create” to create the bucket

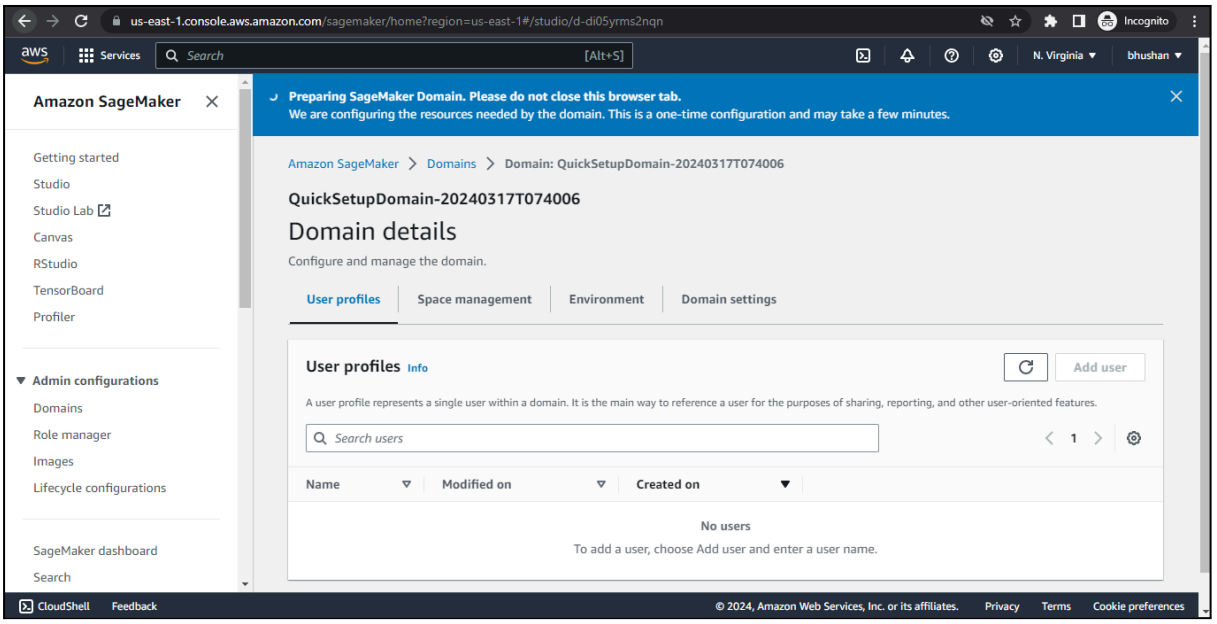
Change the permissions

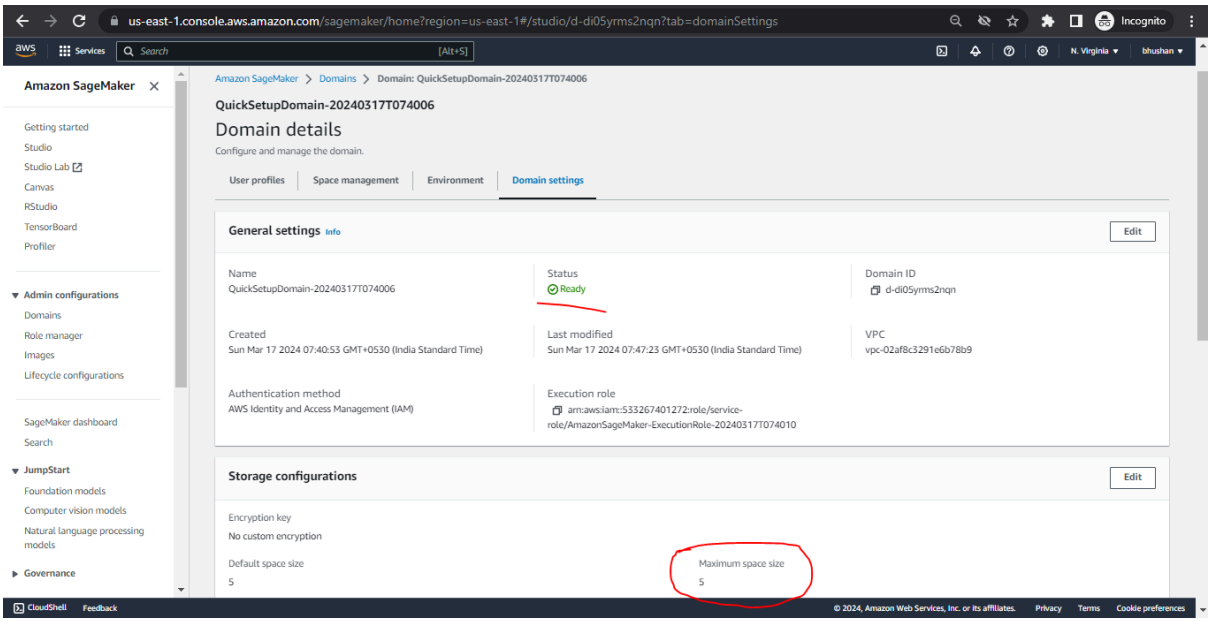


SageMaker Canvas

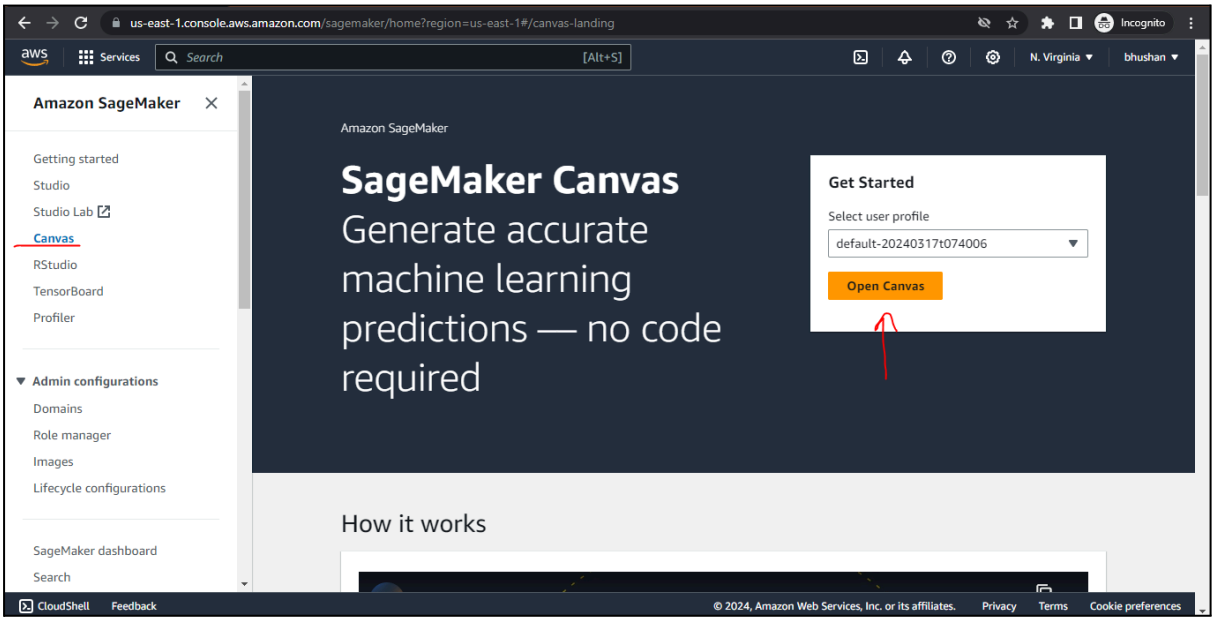



us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/studio/create-domain/quick-setup
Incognito
aws
Services
Search
[Alt+S]
N. Virginia
bhushan
Amazon SageMaker
Getting started
Studio
Studio Lab
Canvas
RStudio
TensorBoard
Profiler
Admin configurations
Domains
Role manager
Images
Lifecycle configurations
SageMaker dashboard
Search
Amazon SageMaker > Set up SageMaker Domain
Set up SageMaker Domain
Use SageMaker Domain as the central store to manage the configuration of SageMaker for your organization.
Set up for single user (Quick setup)
Let Amazon SageMaker configure your account, and set up permissions for your SageMaker Domain.
New IAM role with AmazonSageMakerFullAccess policy
Public internet access, and standard encryption
SageMaker Studio - New, and SageMaker Studio Classic integrations
Sharable SageMaker Studio Notebooks
SageMaker Canvas
IAM Authentication
Perfect for single user domains and first time users looking to get started with SageMaker.
Set up for organizations
Control all aspects of account configuration, including permissions, integrations, and encryption.
Advanced network security, and data encryption
SageMaker Studio - New, SageMaker Studio Classic, RStudio, and Code Editor Based on Code-OSS, Visual Studio Code Open Source integrations
SageMaker Studio Projects, and Jumpstart
SageMaker Canvas, and Amazon services integrations
IAM, or IAM Identity Center (successor to AWS SSO)
Better for admins with large user groups, but you can always update your account configuration settings later if you want to do a quick setup now.
Set up
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

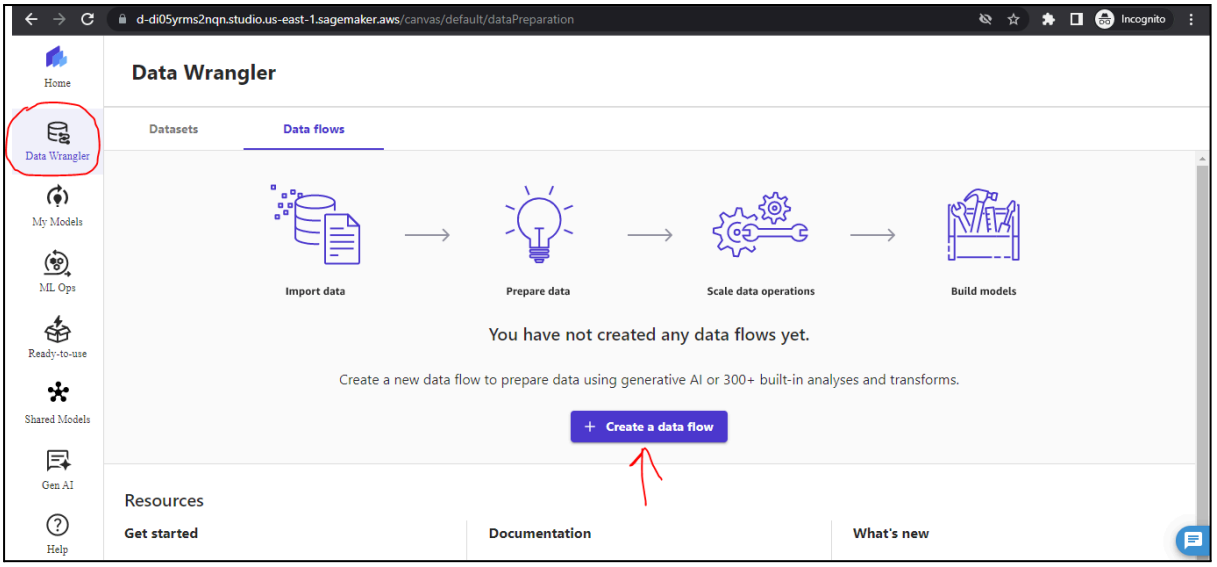

us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/studio/d-di05yrms2nqn
Incognito
aws
Services
Search
[Alt+S]
N. Virginia
bhushan
Amazon SageMaker
Getting started
Studio
Studio Lab
Canvas
RStudio
TensorBoard
Profiler
Admin configurations
Domains
Role manager
Images
Lifecycle configurations
SageMaker dashboard
Search
Preparing SageMaker Domain. Please do not close this browser tab.
We are configuring the resources needed by the domain. This is a one-time configuration and may take a few minutes.
Amazon SageMaker > Domains > Domain: QuickSetupDomain-20240317T074006
QuickSetupDomain-20240317T074006
Domain details
Configure and manage the domain.
User profiles
Space management
Environment
Domain settings
User profiles Info
Add user
A user profile represents a single user within a domain. It is the main way to reference a user for the purposes of sharing, reporting, and other user-oriented features.
Search users
1
Name
Modified on
Created on
No users
To add a user, choose Add user and enter a user name.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

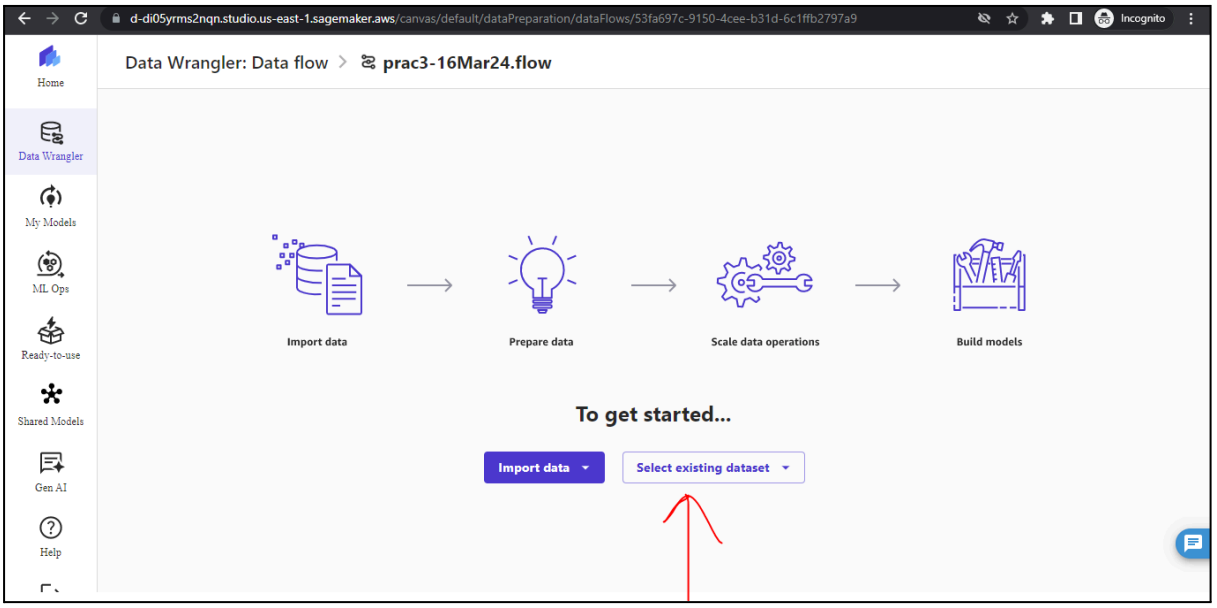


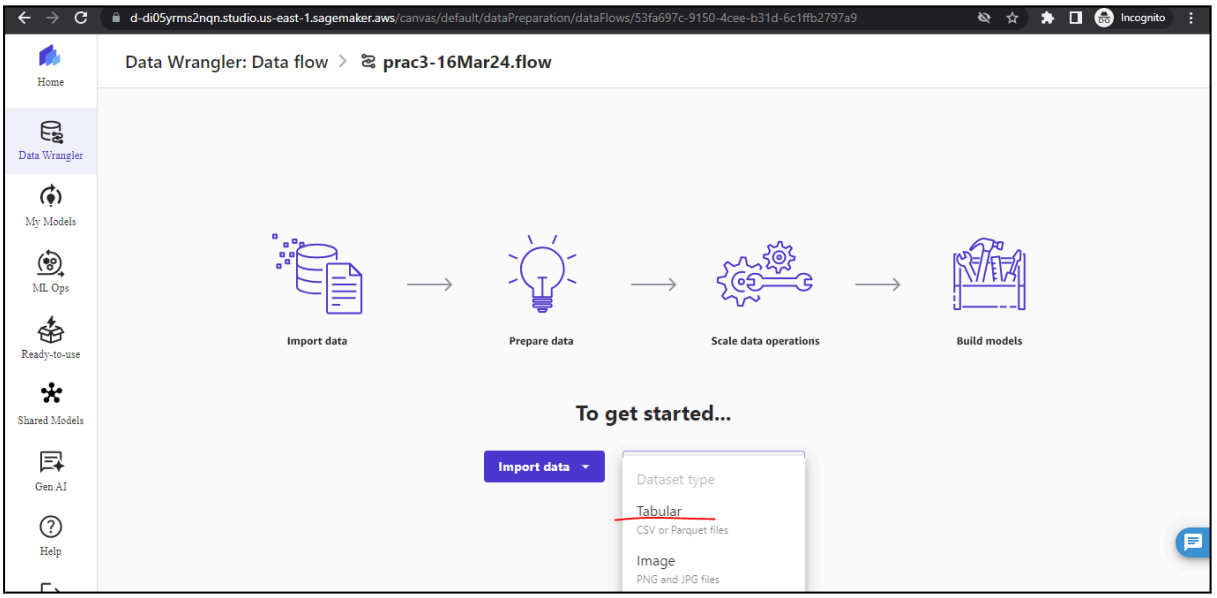
Data Preparation using SageMaker DataWrangler



Import Dataset

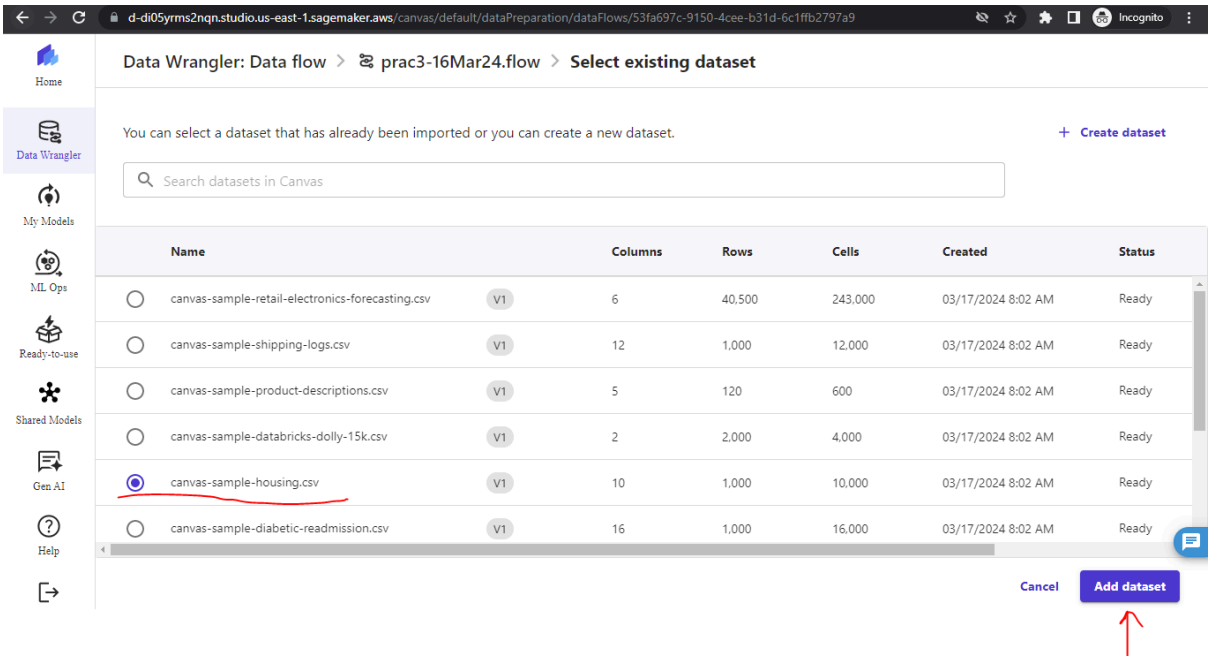




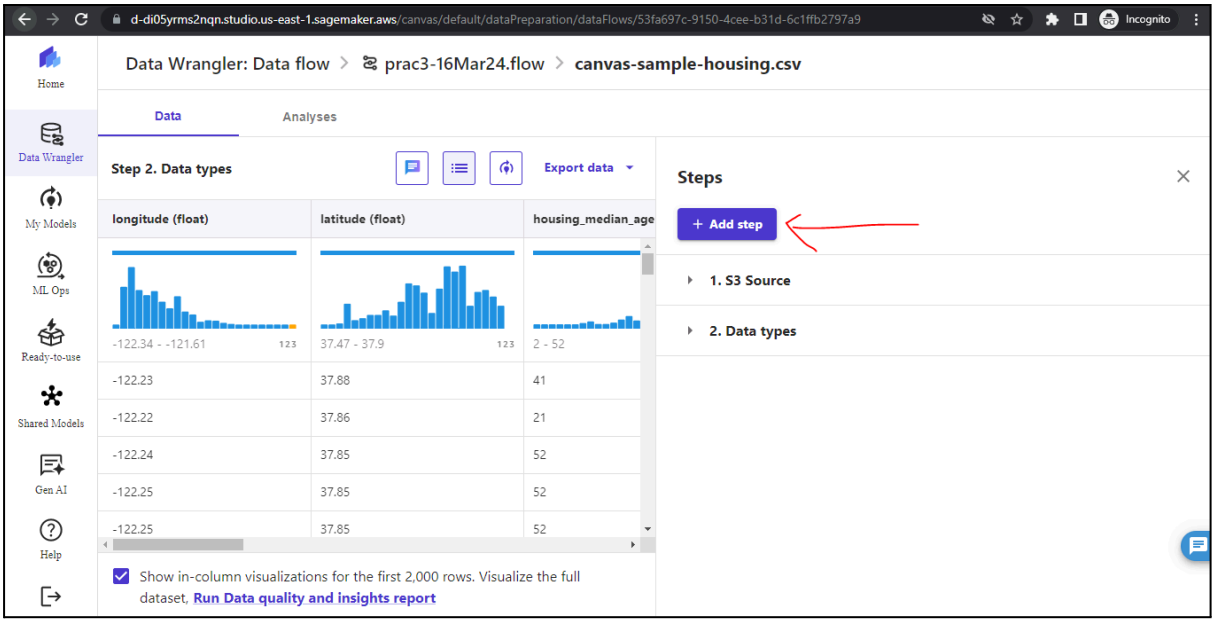


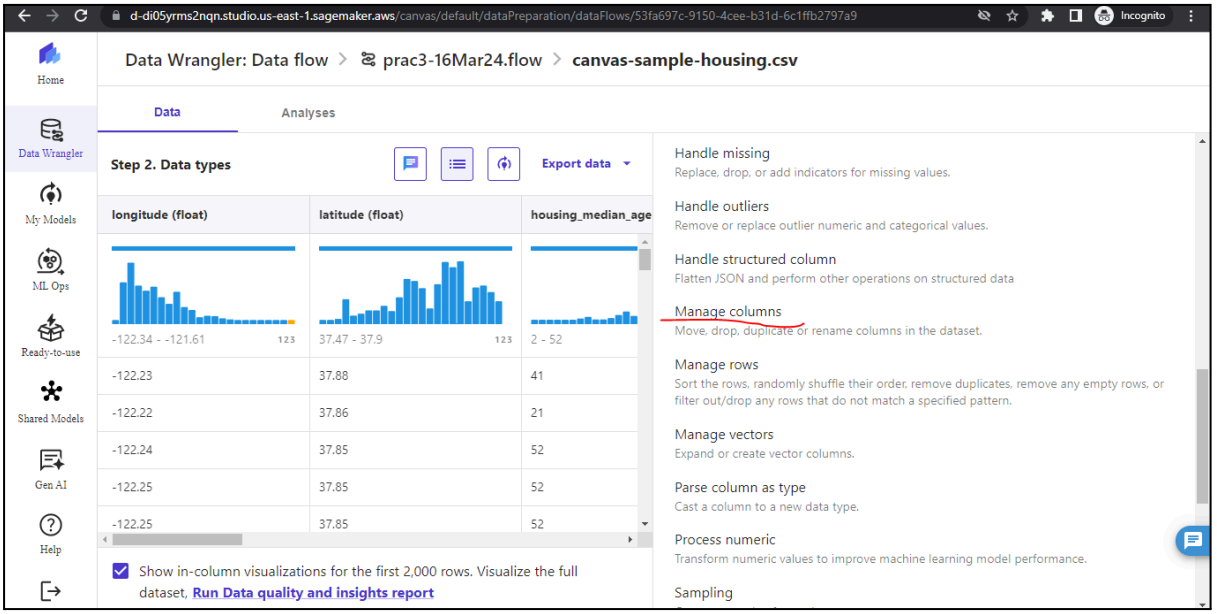
Reference for sample datasets explanation:
https://docs.aws.amazon.com/sagemaker/latest/dg/canvas-sample-datasets.html

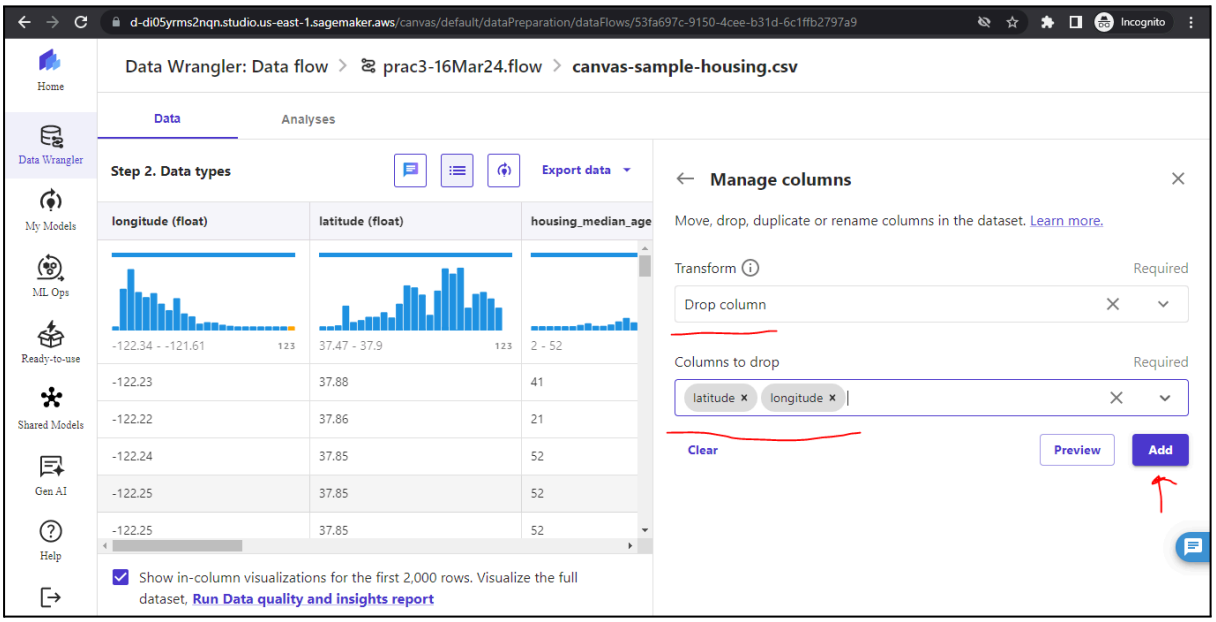
canvas-sample-housing.csv: This dataset contains data on the characteristics tied to a given housing price. You can use this dataset to predict housing prices. Use the **median_house_value** column as the target column, and use the numeric prediction model type with this dataset. To learn more about building a model with this dataset, see the SageMaker Canvas workshop page. This is the California housing dataset obtained from the StatLib repository.



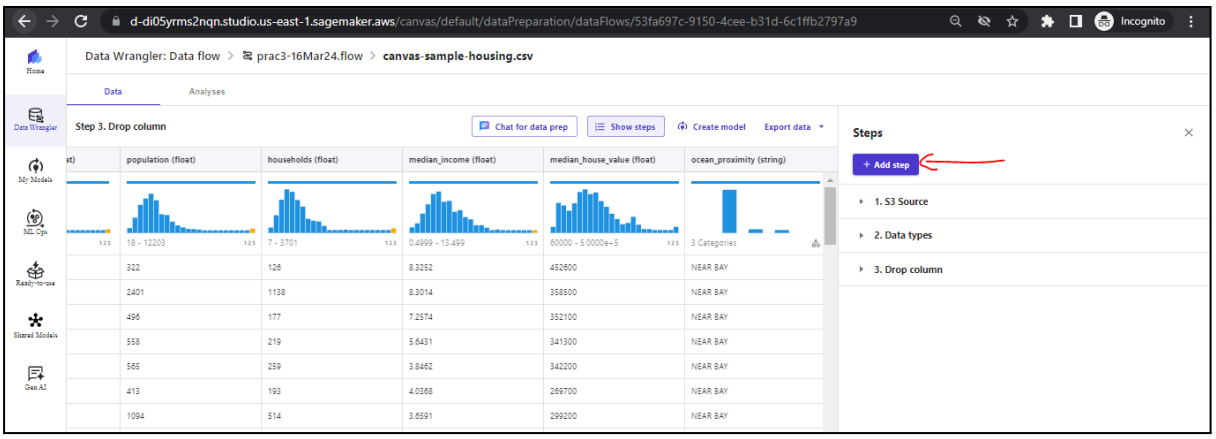
Delete irrelevant columns

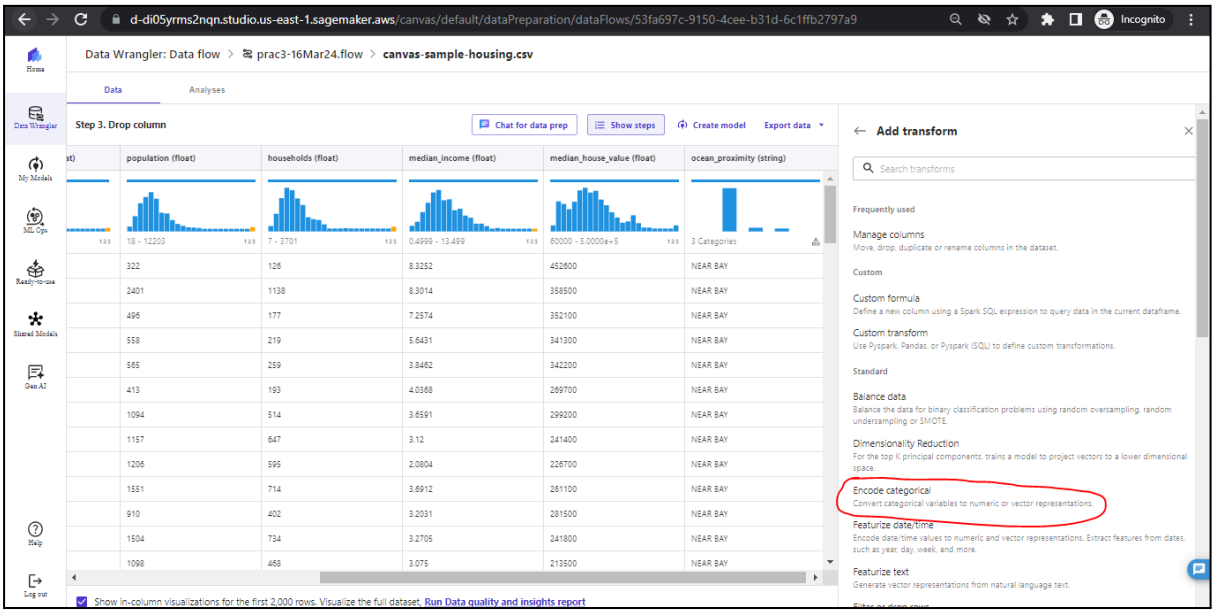


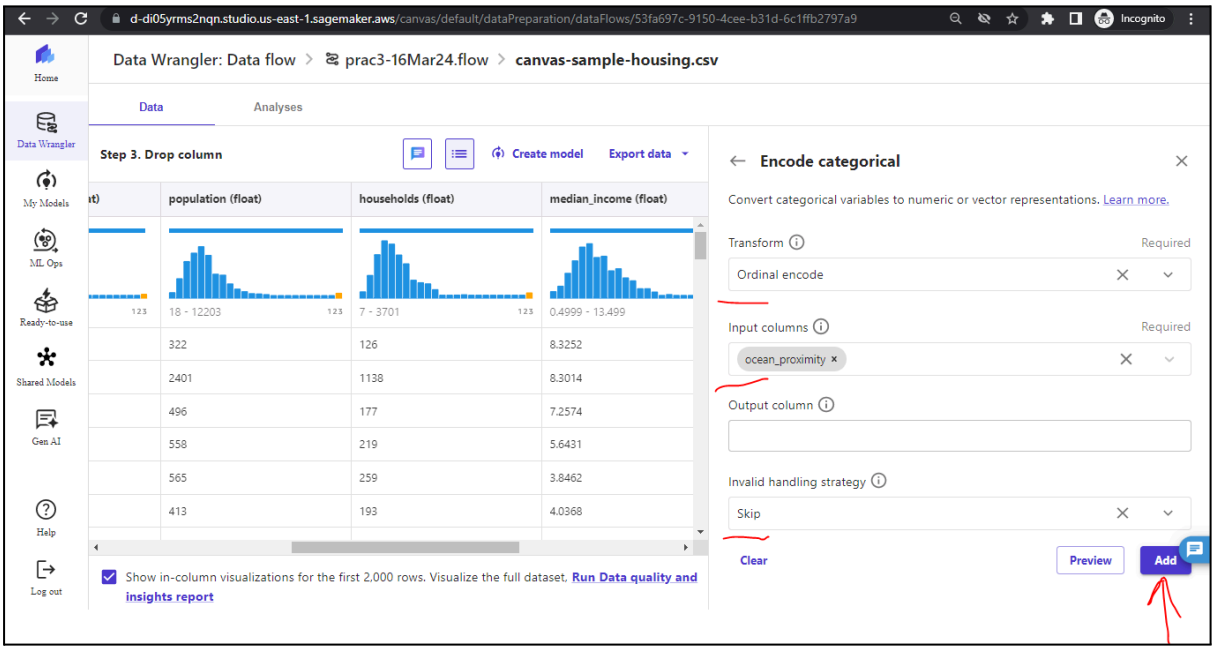




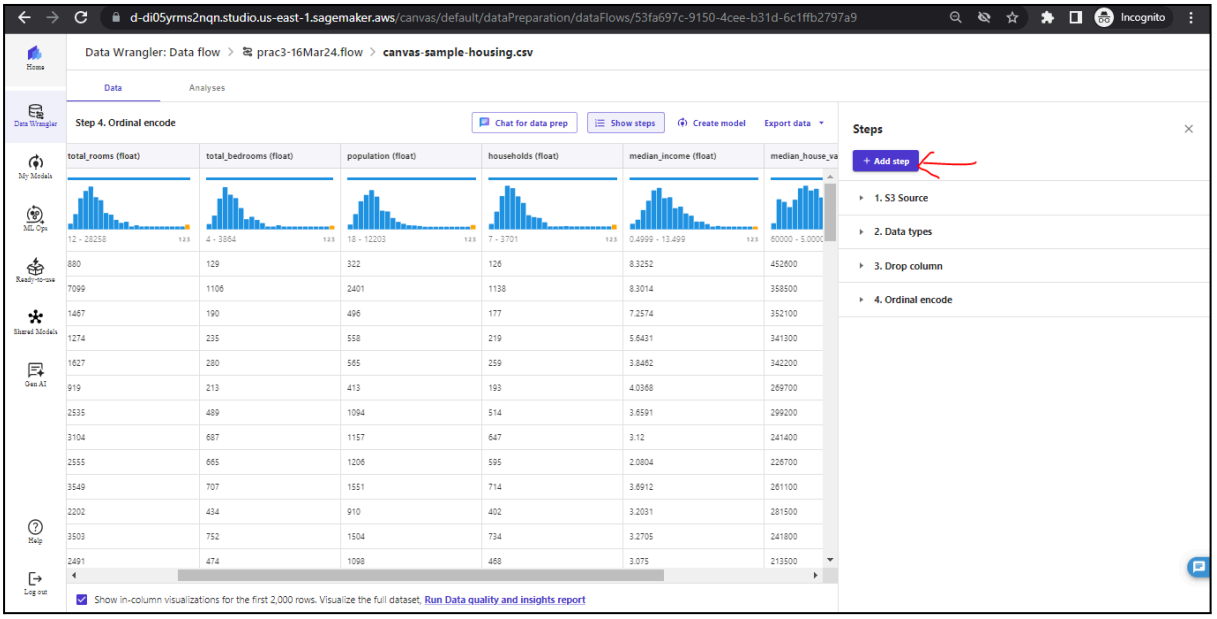
Ordinal Encoding for categorical column

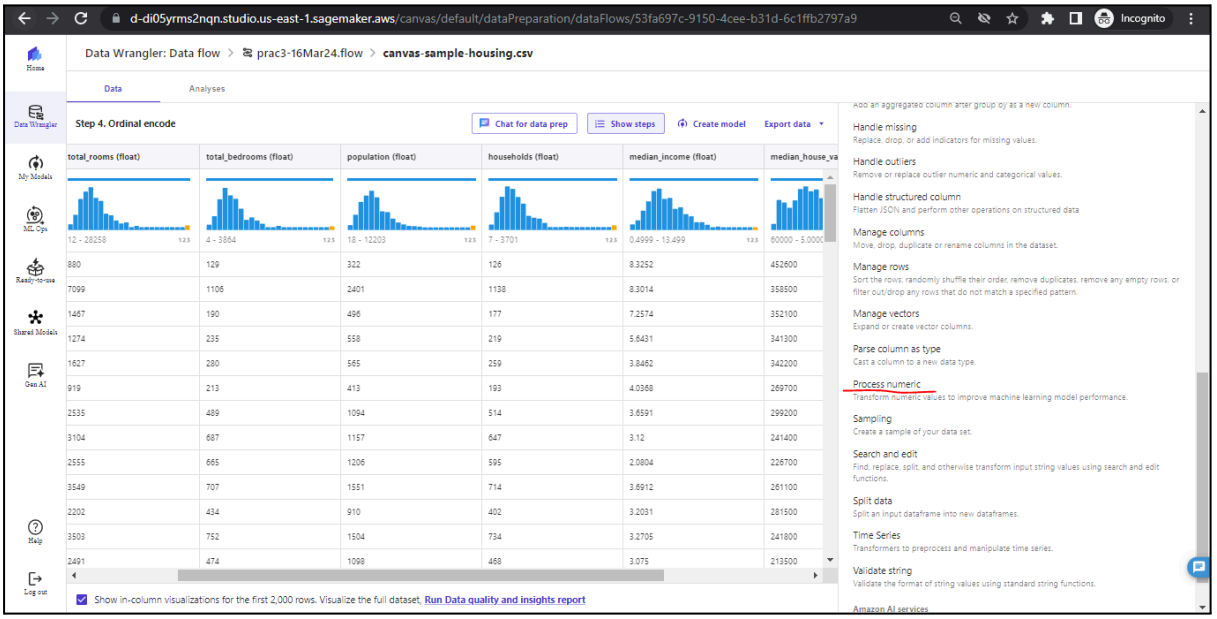


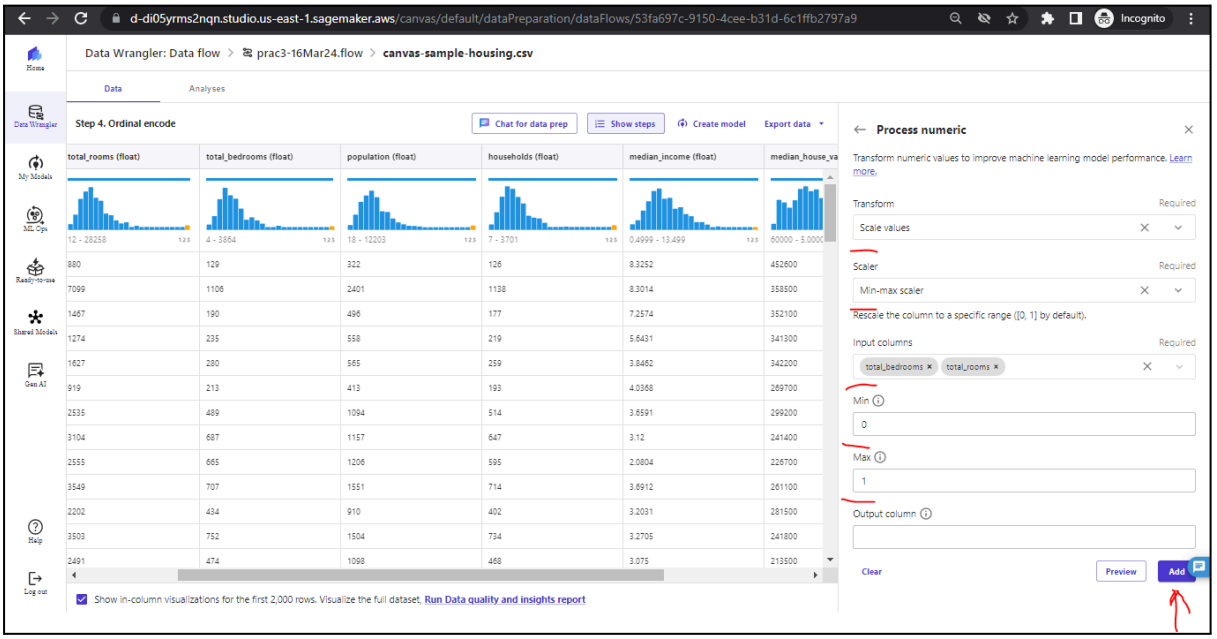




Min-Max Scaling

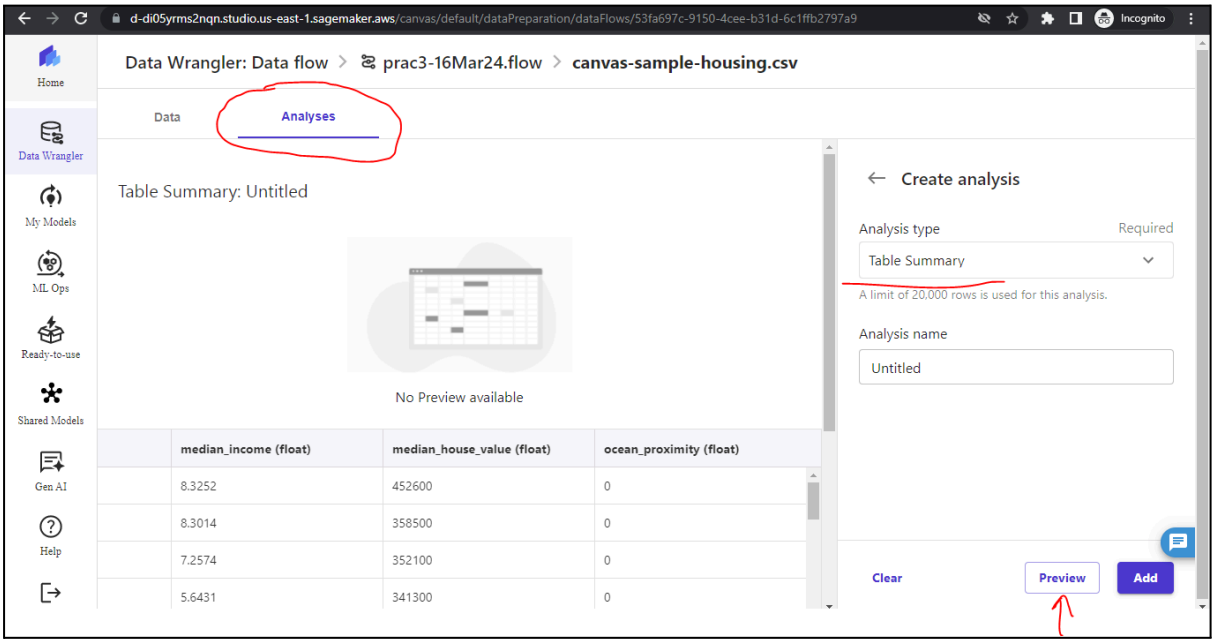


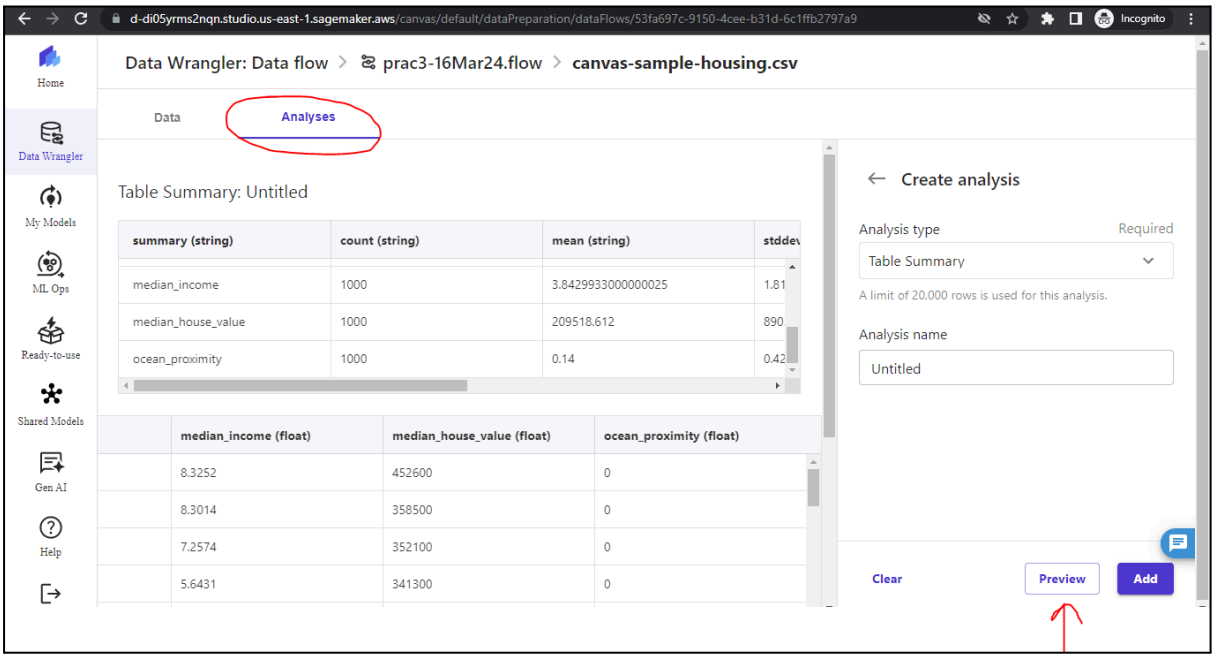

d-di05yrms2nqn.studio.us-east-1.sagemaker.aws/canvas/default/dataPreparation/dataFlows/53fa697c-9150-4cee-b31d-6c1ffb2797a9
Incognito
Data Wrangler: Data flow > prac3-16Mar24.flow > canvas-sample-housing.csv
Data
Analyses
Step 4. Ordinal encode
Chat for data prep
Show steps
Create model
Export data
total_rooms (float)
total_bedrooms (float)
population (float)
households (float)
median_income (float)
Handle missing
Handle outliers
Handle structured column
Manage columns
Manage rows
Manage vectors
Parse column as type
Process numeric
Sampling
Search and edit
Split data
Time Series
Validate string
Show in-column visualizations for the first 2,000 rows. Visualize the full dataset, Run Data quality and insights report


d-di05yrms2nqn.studio.us-east-1.sagemaker.aws/canvas/default/dataPreparation/dataFlows/53fa697c-9150-4cee-b31d-6c1ffb2797a9
Incognito
Data Wrangler: Data flow > prac3-16Mar24.flow > canvas-sample-housing.csv
Step 4. Ordinal encode
Process numeric
Transform numeric values to improve machine learning model performance. Learn more.
Transform
Required
Scale values
Scaler
Min-max scaler
Rescale the column to a specific range ([0, 1] by default).
Input columns
total_bedrooms
total_rooms
Min
0
Max
1
Output column
Clear
Preview
Add

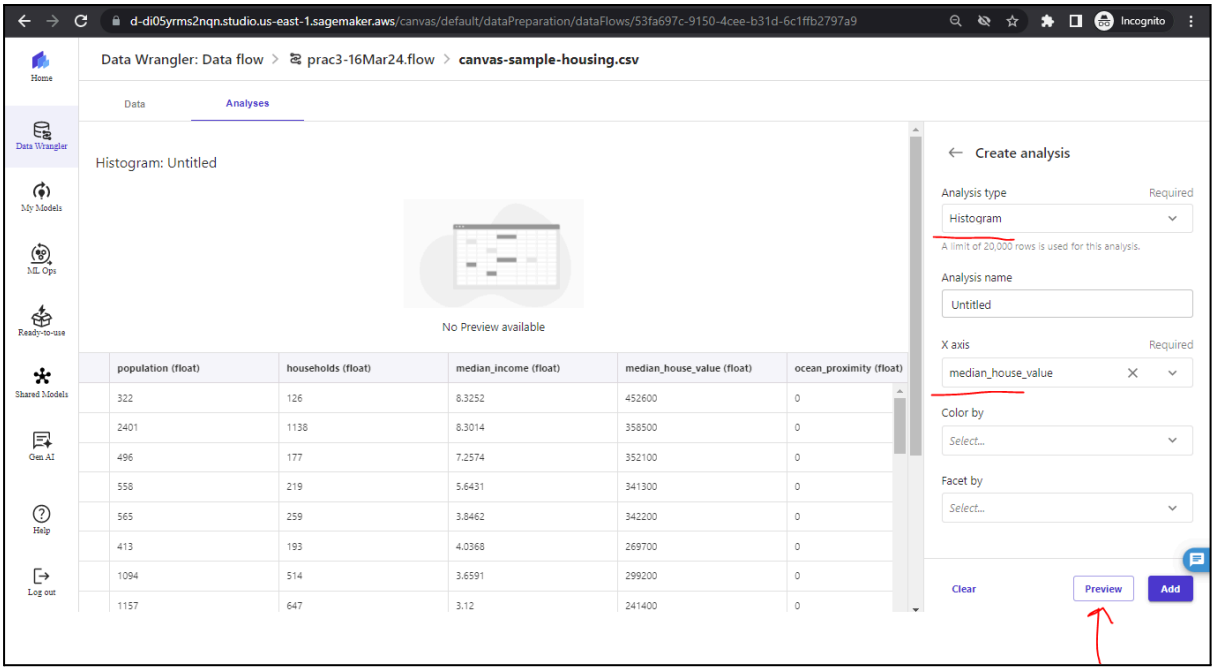
EDA using SageMaker DataWrangler

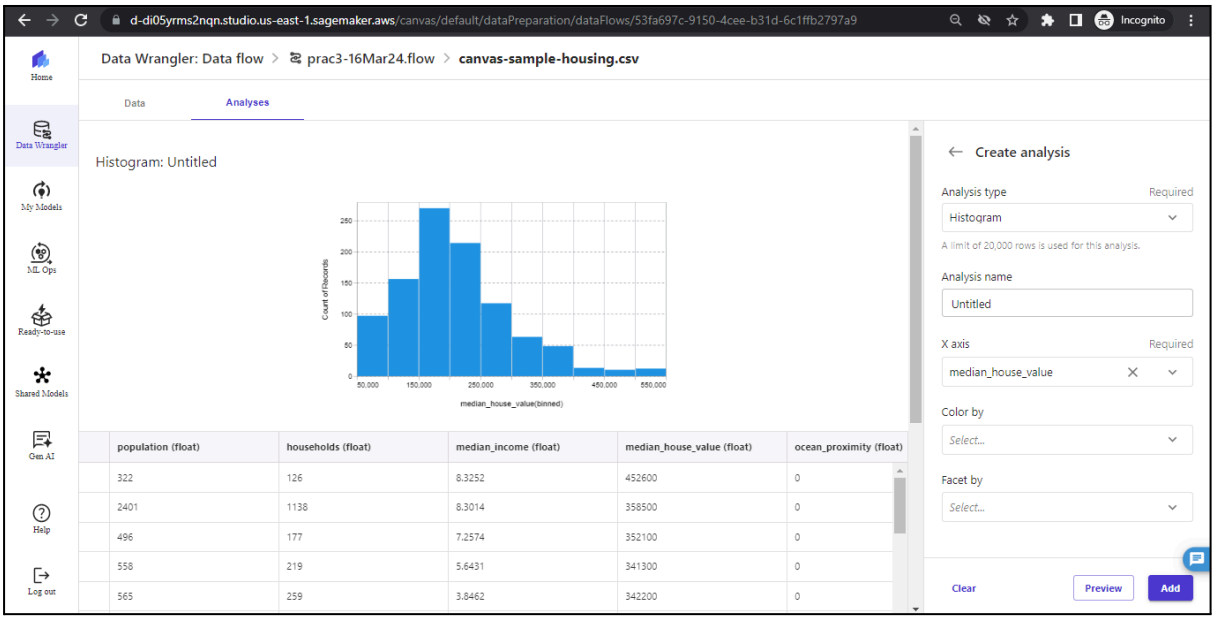
Table Summary





Checking if the target column is normally distributed (bell shaped graph)





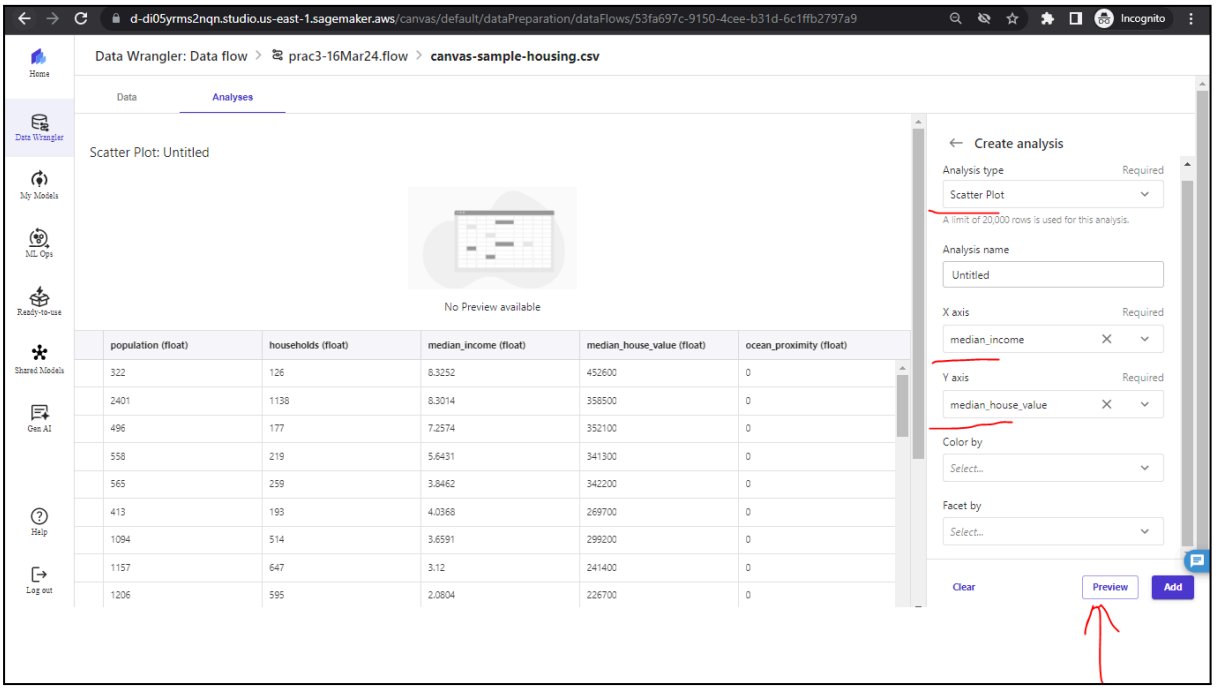
Data is normally distributed but slightly skewed towards the left

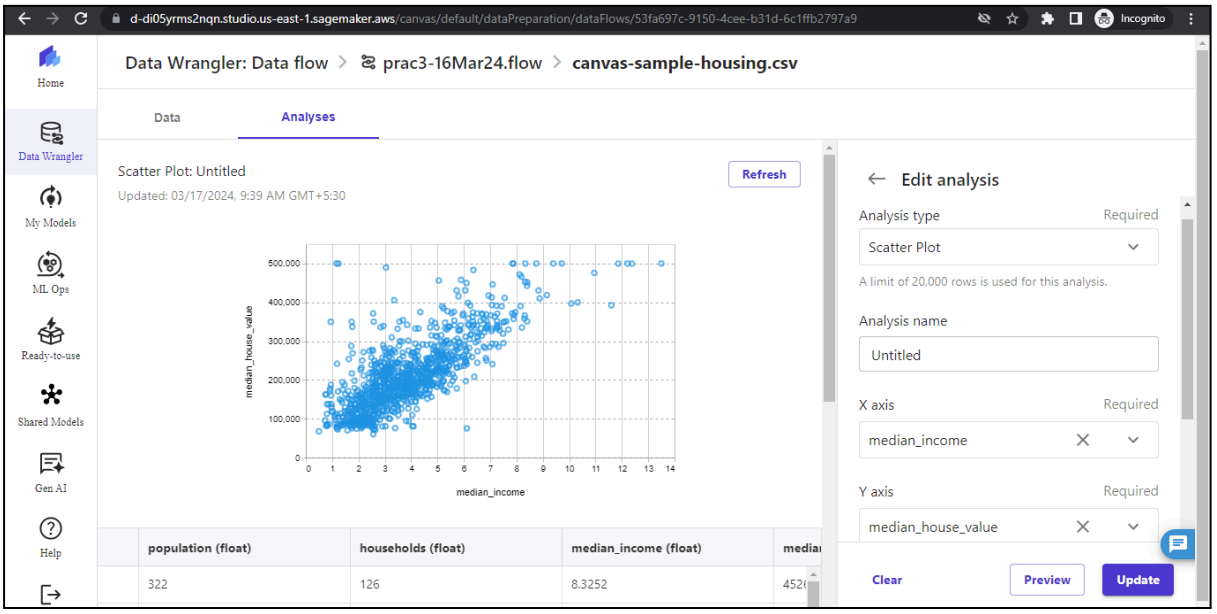
Since this is a sample dataset, we can proceed with a slightly skewed data also

Checking if the relationship between the predictor and target variable is linear

The **dependent variable** is called the **target** and the **independent variable** is called the **predictor**

The **dependent variable (target) depends** on the **independent variable (predictor)**

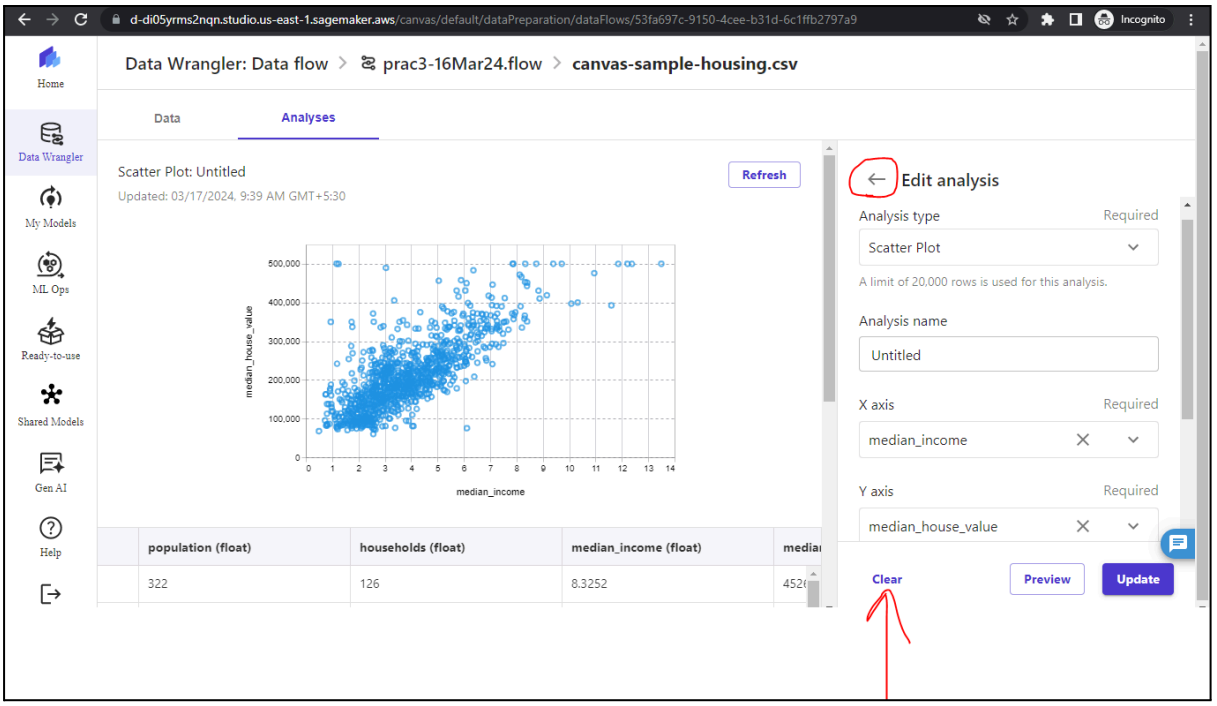


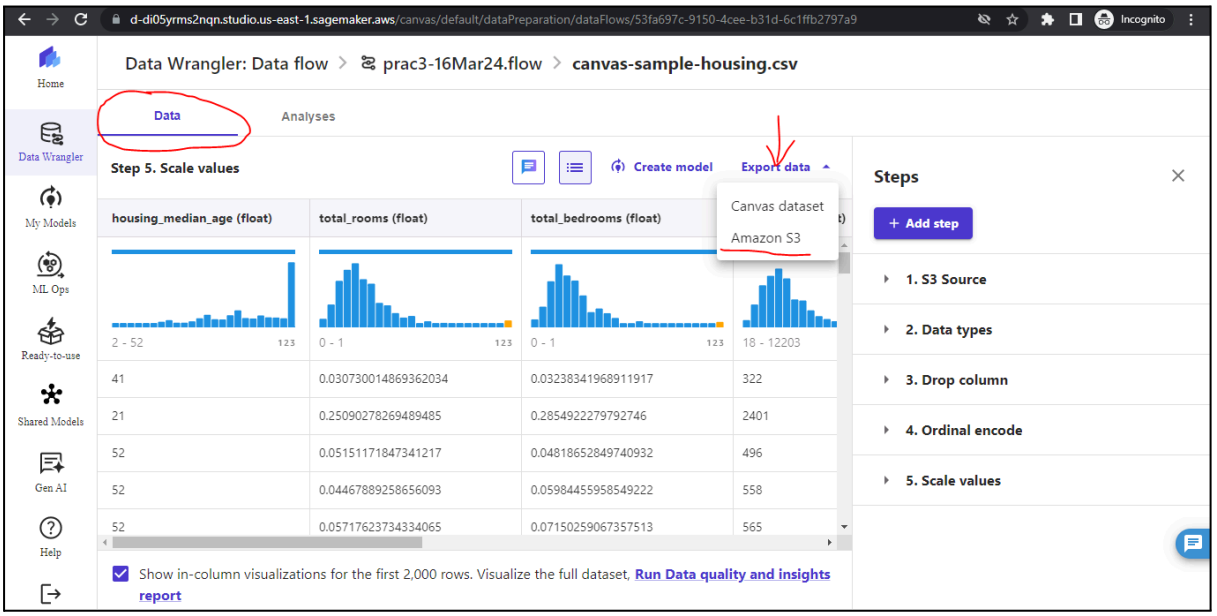


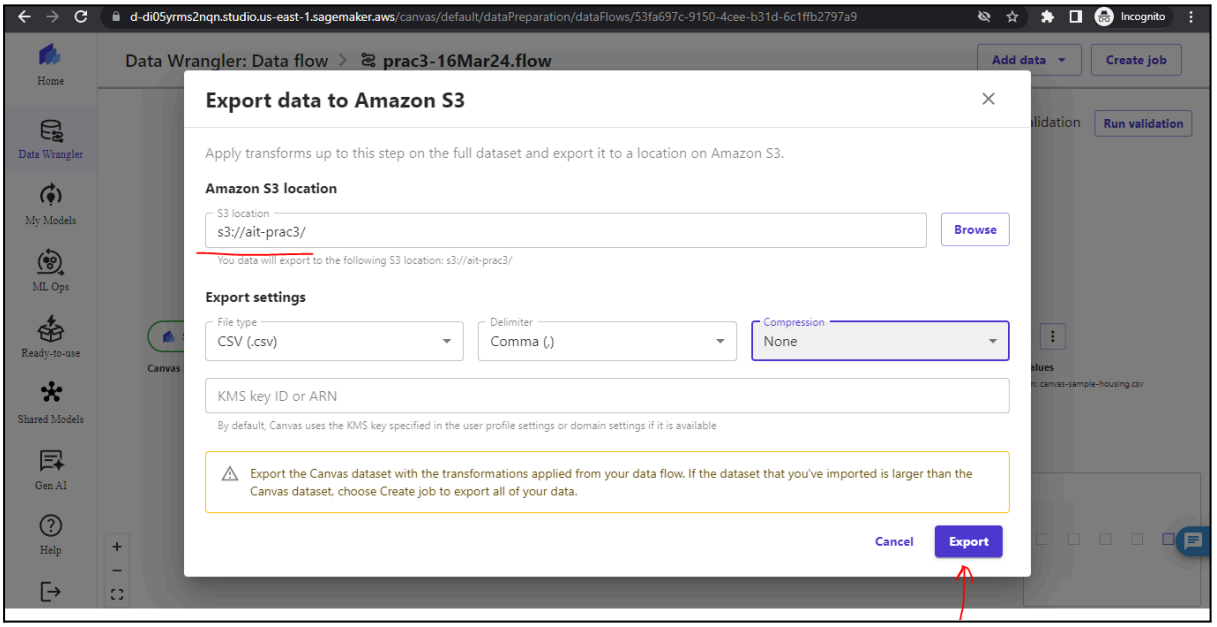
The columns' relation is roughly linear

The relationship looks roughly linear, so we can proceed with the linear regression.

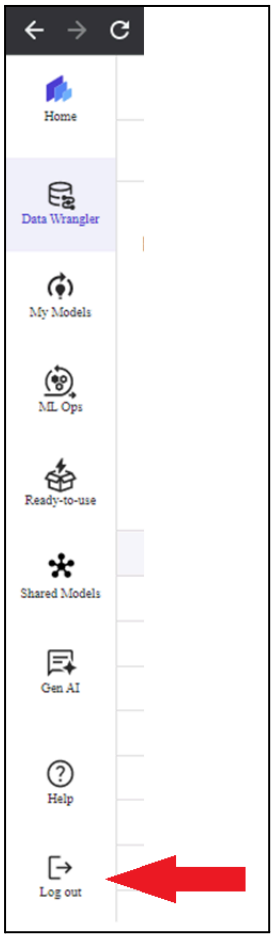
Export prepared data





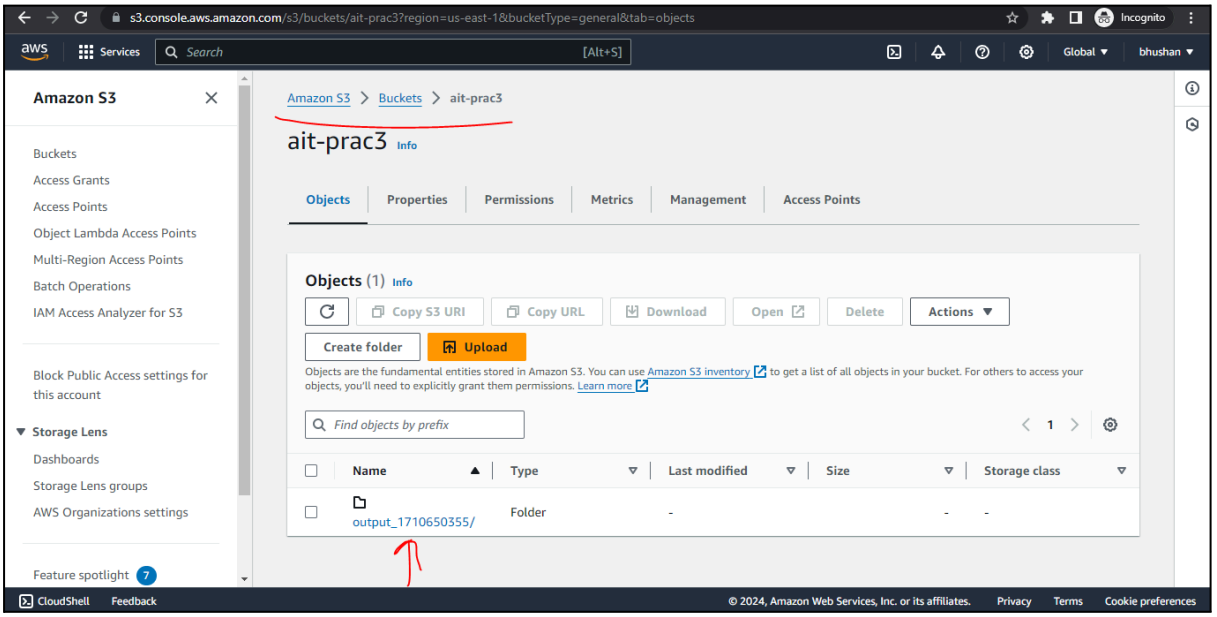


Logout of the Canvas after the data is exported

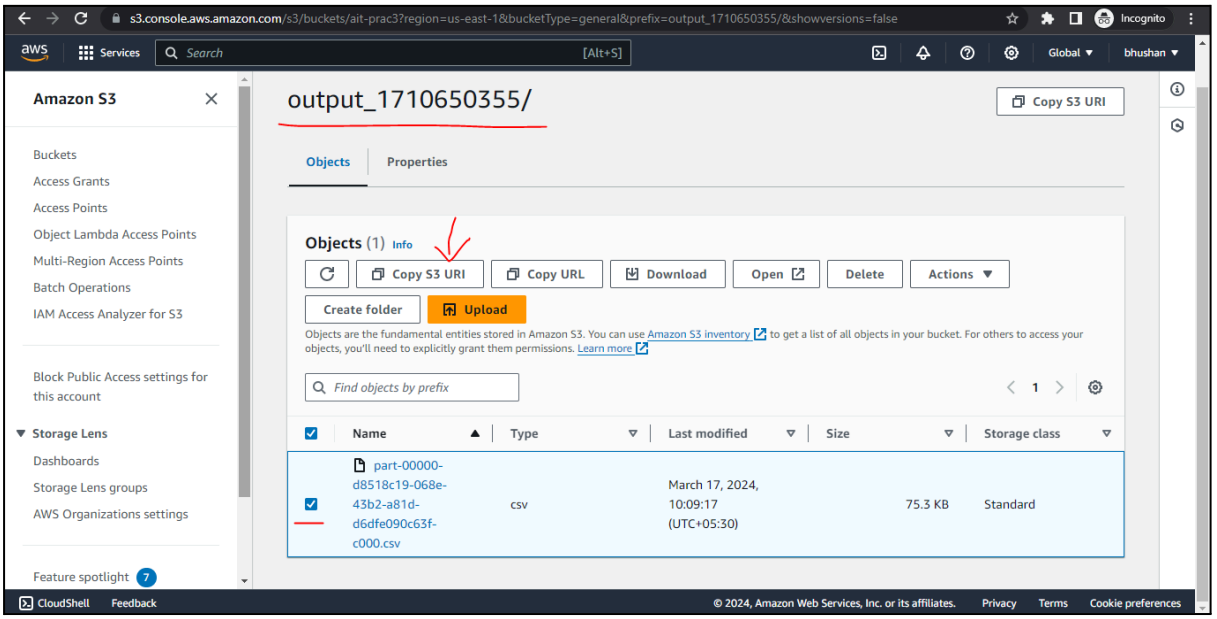


Important: Delete the SageMaker-domain-user and the SageMaker-domain

Locate the exported CSV file in the bucket

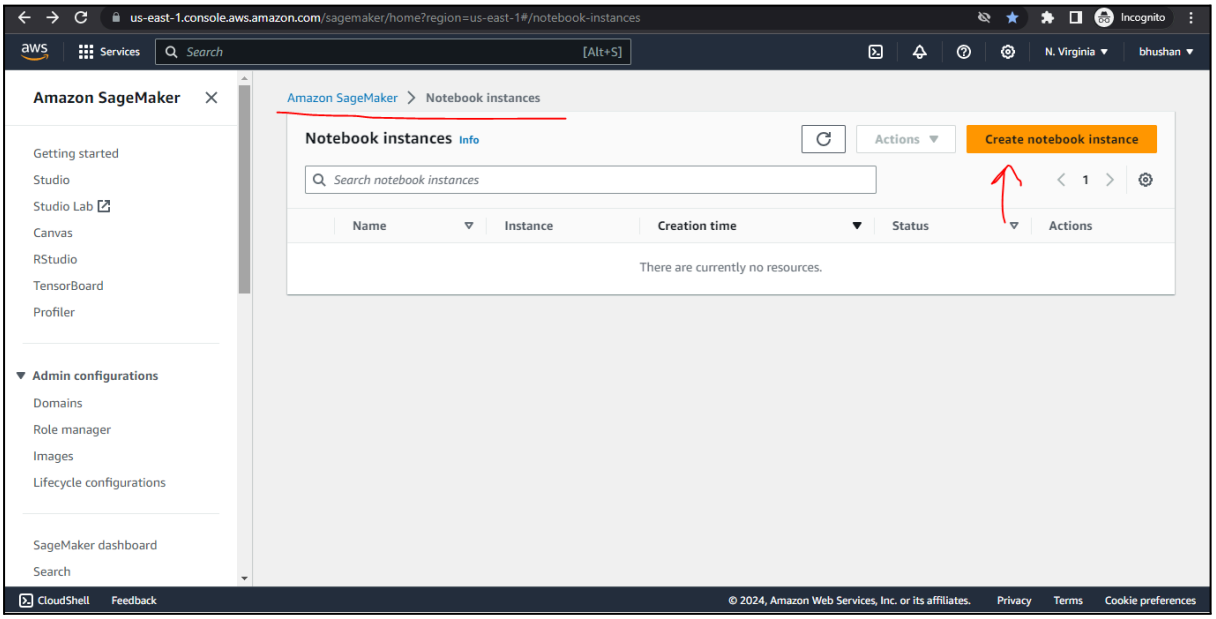


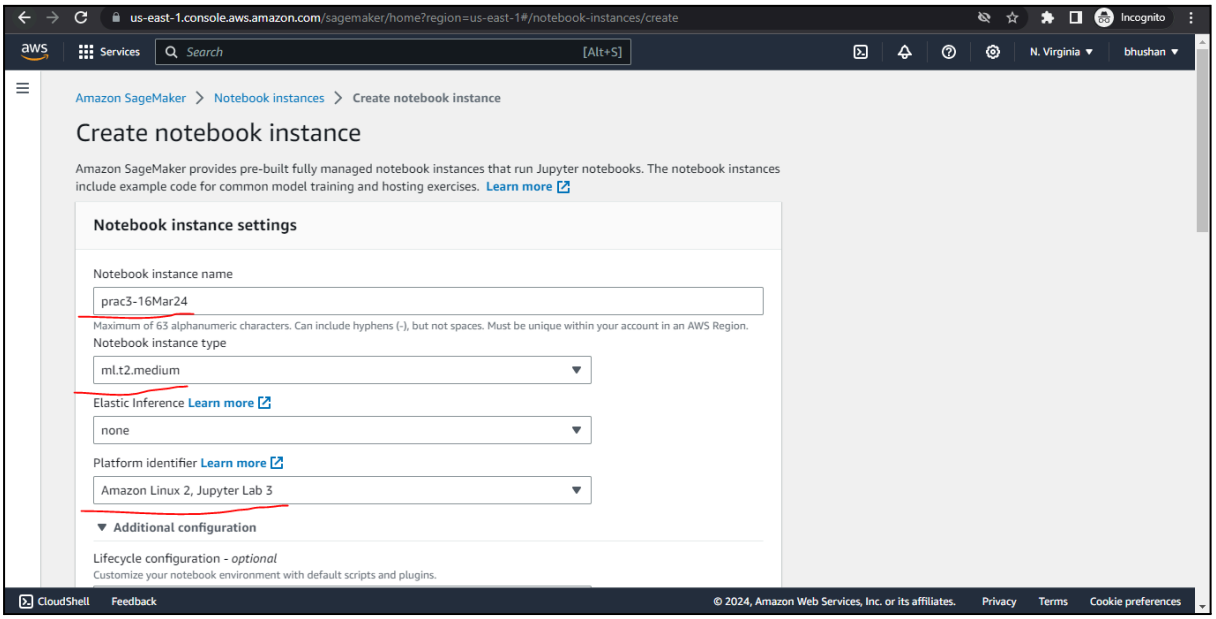
Copy the S3 URI of the csv file (to be used later in the python program)



Simple Linear Regression using SageMaker

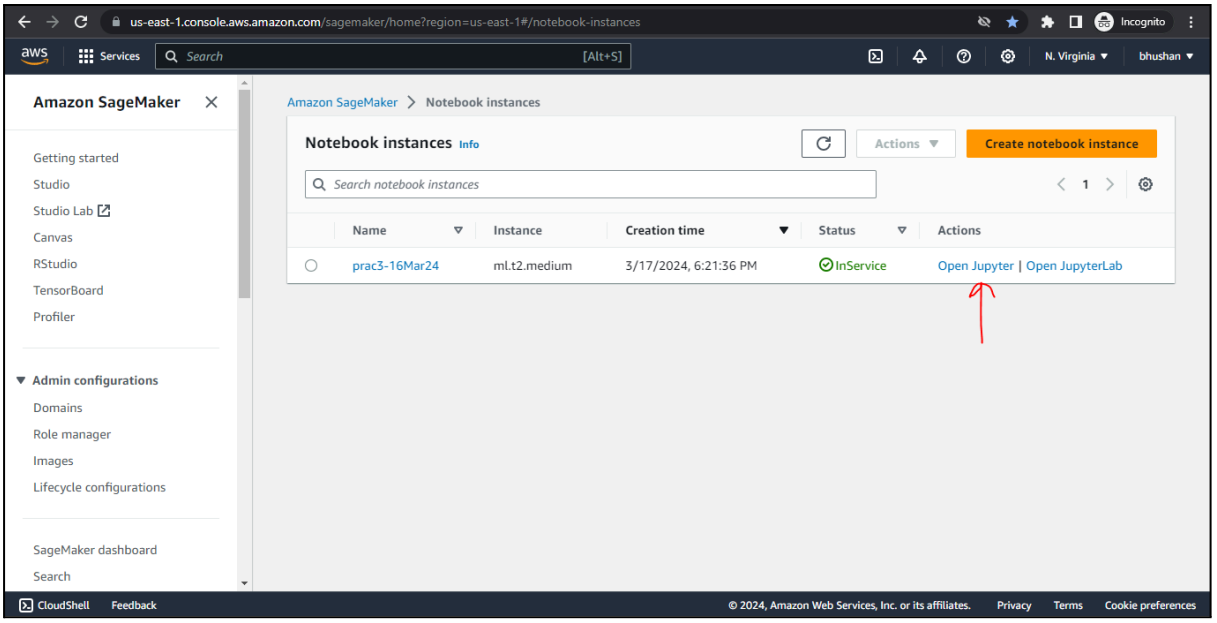
Create a Notebook

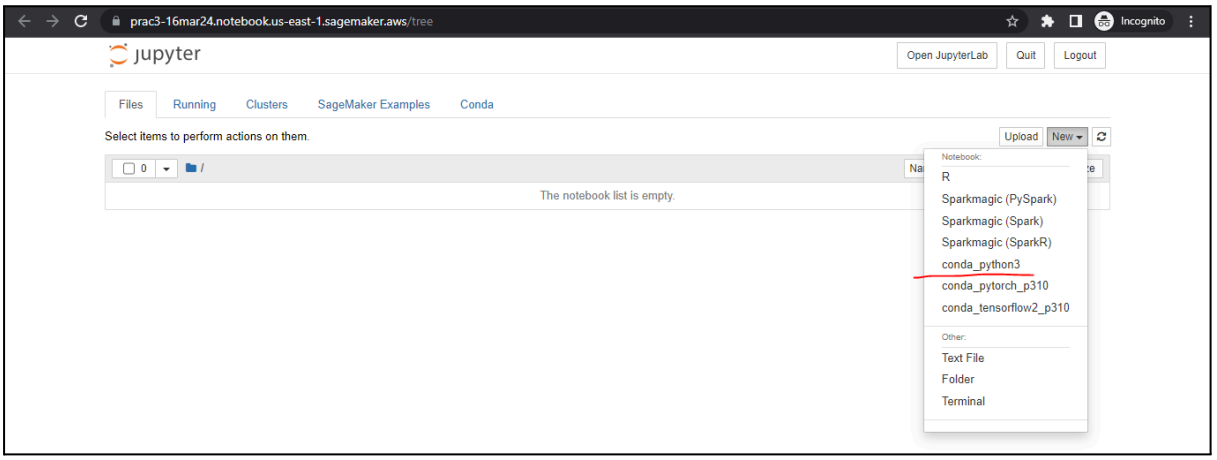




Click on “create” to create the notebook

Open in Jupyter to add the code





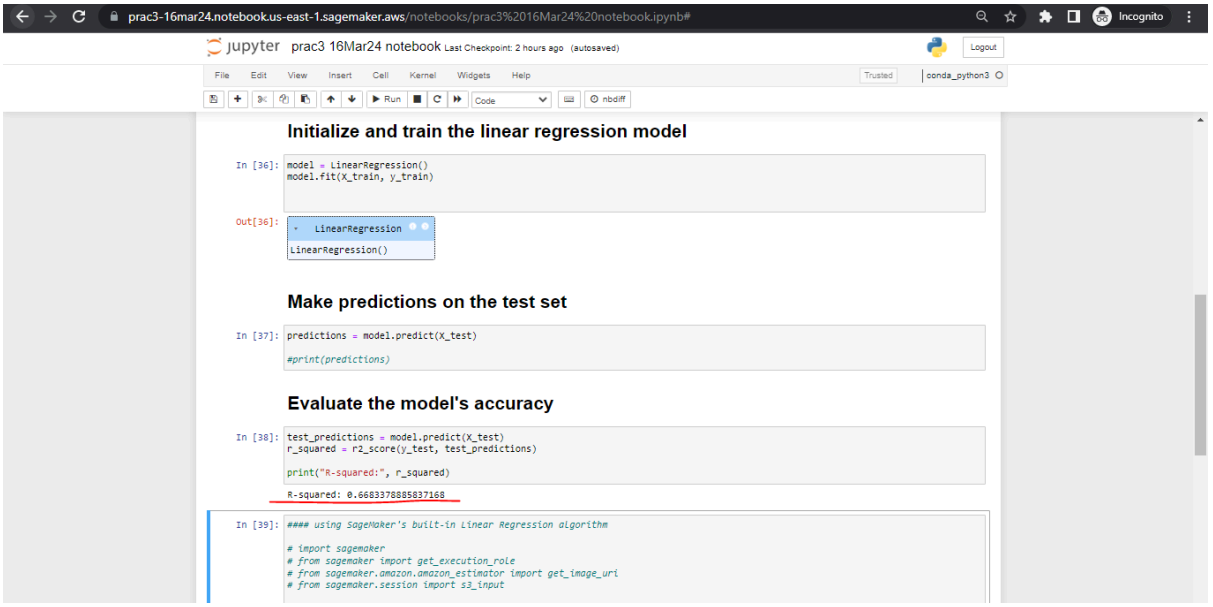
Add the following code

```
# Using linear regression algorithm from sklearn library
import pandas as pd
import boto3
from io import StringIO
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
# Load the dataset
s3_bucket = 'ait-prac3'
```

```
s3_key =
'output_1710650355/part-00000-d8518c19-068e-43b2-a81d-d6dfe090c63f-c000.csv'
s3_client = boto3.client('s3')
response = s3_client.get_object(Bucket=s3_bucket, Key=s3_key)
data = pd.read_csv(response['Body'])
# data = pd.read_csv(response['Body'], na_values=['NA', 'N/A', 'NaN', ''])
# Split the data into features (X) and target variable (y)
X = data[['median_income']]
y = data[['median_house_value']] # Notice the double square brackets to keep it as a
DataFrame
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Initialize and train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_test)
#print(predictions)
# Extract the coefficients
# slope = model.coef_[0][0]
# intercept = model.intercept_[0]
# Print the equation of the linear regression line
# print(f"Linear Regression Equation: y = {slope:.2f}x + {intercept:.2f}")
# # Evaluate the model's accuracy
test_predictions = model.predict(X_test)
r_squared = r2_score(y_test, test_predictions)
print("R-squared:", r_squared)
# using SageMaker's built-in Linear Regression algorithm
# import sagemaker
# from sagemaker import get_execution_role
# from sagemaker.amazon.amazon_estimator import get_image_uri
# from sagemaker.session import s3_input
# Define your S3 bucket and prefix
# bucket = 'your-s3-bucket'
# prefix = 'your-prefix'
```

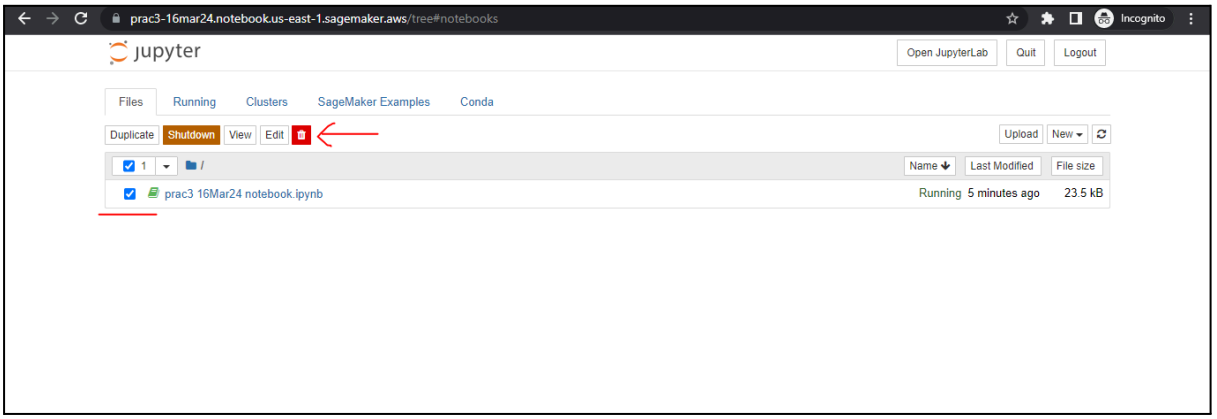
```
# Set up SageMaker session and role
# sagemaker_session = sagemaker.Session()
# role = get_execution_role()
# Specify the location of your dataset in S3
# train_data = 's3://{}/{}/{}'.format(bucket, prefix, 'canvas-sample-housing.csv')
# Create a LinearLearner estimator
# linear_container = get_image_uri(sagemaker_session.boto_region_name, 'linear-learner')
# linear = sagemaker.estimator.Estimator(linear_container,
#                                        role,
#                                        train_instance_count=1,
#                                        # train_instance_type='ml.m4.xlarge',
#                                        train_instance_type='',
#                                        output_path='s3://{}/{}/output'.format(bucket, prefix),
#                                        sagemaker_session=sagemaker_session)
# Set hyperparameters
# linear.set_hyperparameters(predictor_type='regressor',
#                            mini_batch_size=50,
#                            epochs=3)
# Train the model
# linear.fit({'train': s3_input(train_data, content_type='text/csv')})
```

Output

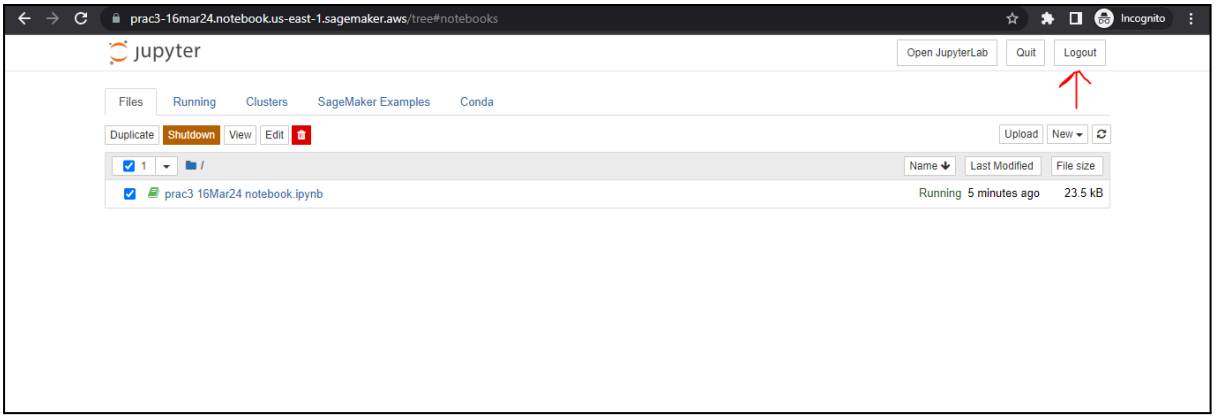


Since the R^2 value is 0.66, we can say that the linear regression model is 66% accurate

Delete the Jupyter Notebook



Logout from Jupyter



CLEAN UP

1. Stop all the services that are running.
2. Delete all the resources that were created for this practical (including the buckets and the notebooks, etc.)

Conclusion: We saw data Preparation and EDA using AWS SageMaker DataWrangler, linear regression using SageMaker.

Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 7, 8, 9**	
Title of Lab Assignment: XGBoost Algorithm, Data Split for Sagemaker, Hyperparameter Search Optimization, Hyperparameters Optimization Strategies, Bias Variance Tradeoff, L2 Regularization (Ridge Regression), L1 Regularization (Lasso Regression), Hyperparameters Optimization Using GridSearchCV.		
DOP: 21-02-2024	**DOS: 28-03-2024**	
CO Mapped: **CO4**	**PO Mapped:** **PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PO9, PO11, PO12, PSO1, PSO2**	**Signature:**

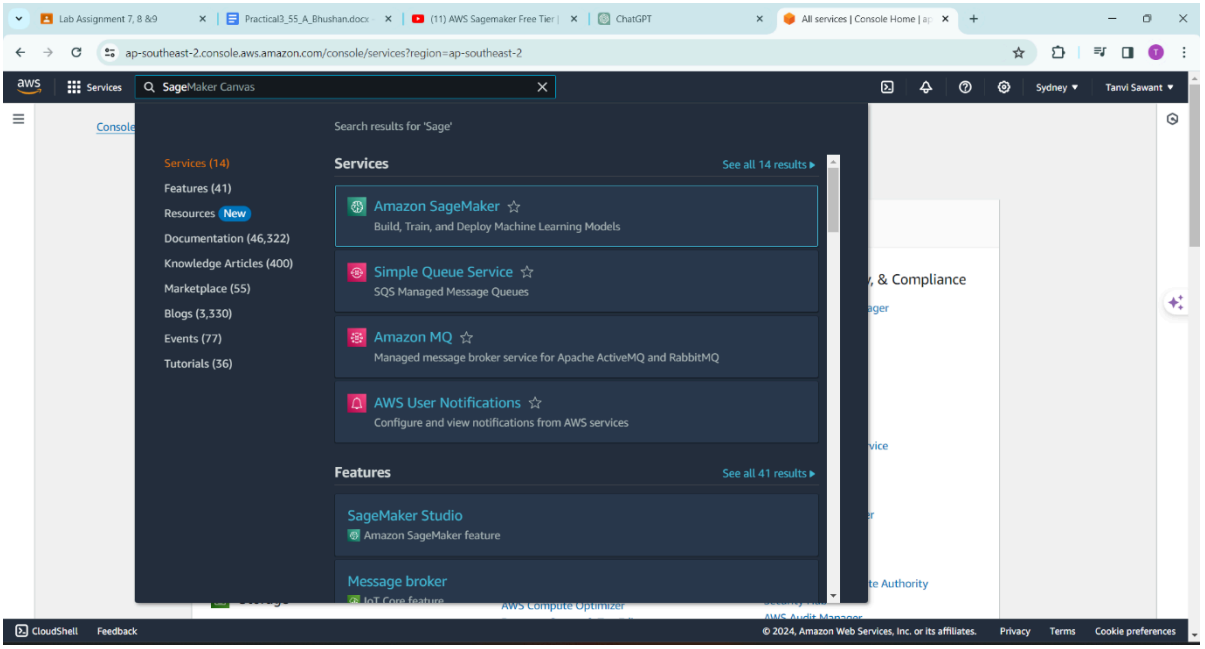
Practical No. 7, 8, 9

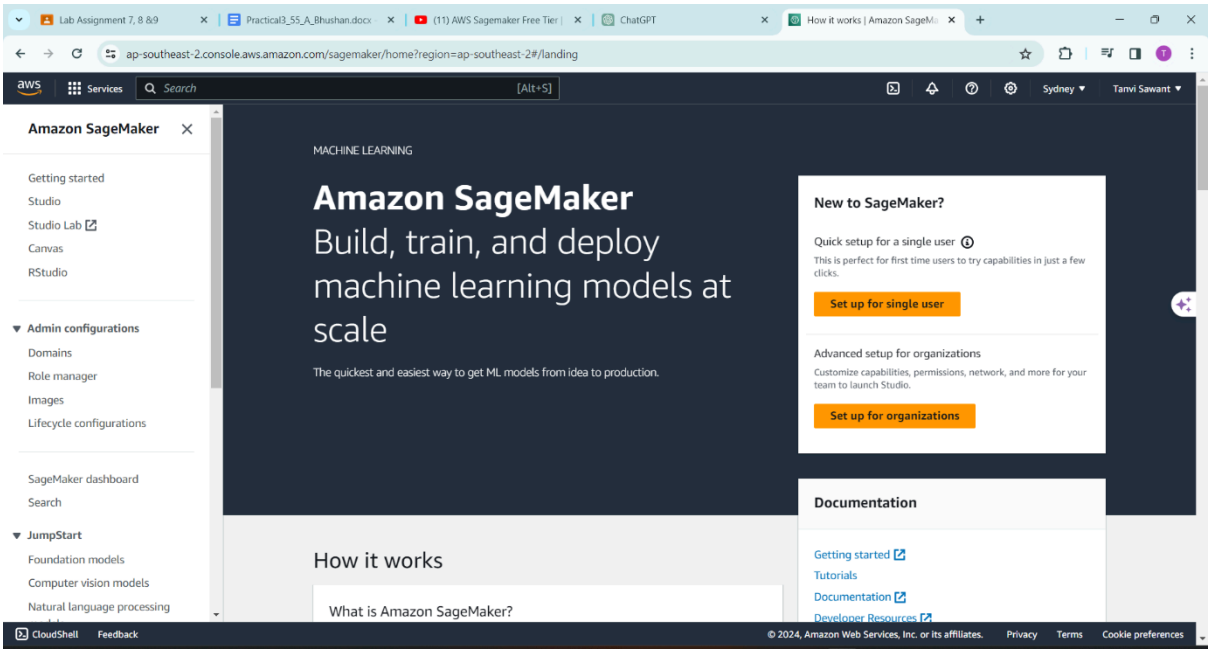
Sign Up for AWS Account:

- Go to the AWS website and sign up for a free tier account if you haven't already.
- Follow the steps to create your account, providing necessary information and setting up your billing preferences.

Navigate to Amazon SageMaker:

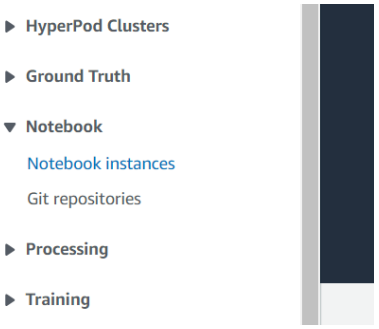
- Once logged in to your AWS Management Console, navigate to the Amazon SageMaker service.
- Click on "Services" in the top-left corner, then type "SageMaker" in the search bar, and click on the SageMaker service.





Create a Notebook Instance:

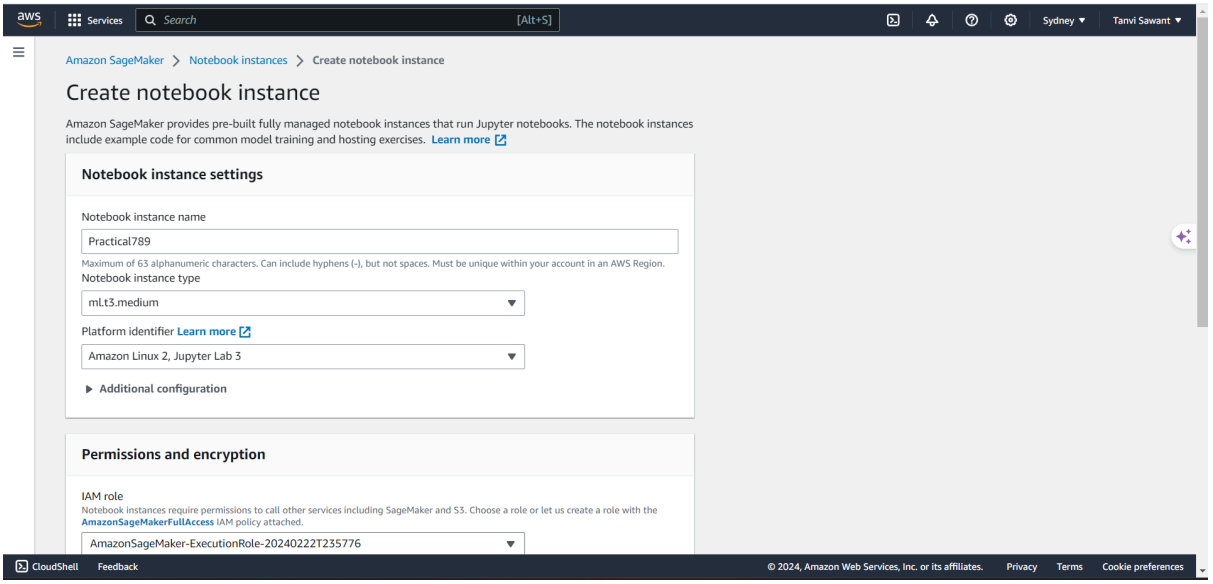
- In the SageMaker dashboard, click on "Notebook instances" in the left sidebar.
- Click on the "Create notebook instance" button.
- Enter a name for your notebook instance, choose an instance type eligible for the free tier (like **ml.t2.medium**), and leave other settings as default.



Configure Notebook Instance:

- In the "Notebook instance settings" section, give your notebook instance a name. Something descriptive like "MyMLNotebook" would work.
- For "Notebook instance type," choose an instance type that is eligible for the free tier, such as **ml.t2.medium**. This ensures you don't incur any charges beyond the free tier limits.

- Leave other settings as default for now. You can explore advanced settings later as needed.



Create IAM Role (Optional):

If you haven't created an IAM role before, you might be prompted to do so. This role grants necessary permissions to SageMaker to access other AWS services on your behalf.
Click on "Create a new role" if prompted, and follow the instructions to create a new IAM role. You can give it a name like "SageMakerRole".

Encryption (Optional):

If you want to encrypt the data stored on the notebook instance's EBS volume, you can choose an AWS Key Management Service (KMS) key. This step is optional for the purposes of this tutorial.

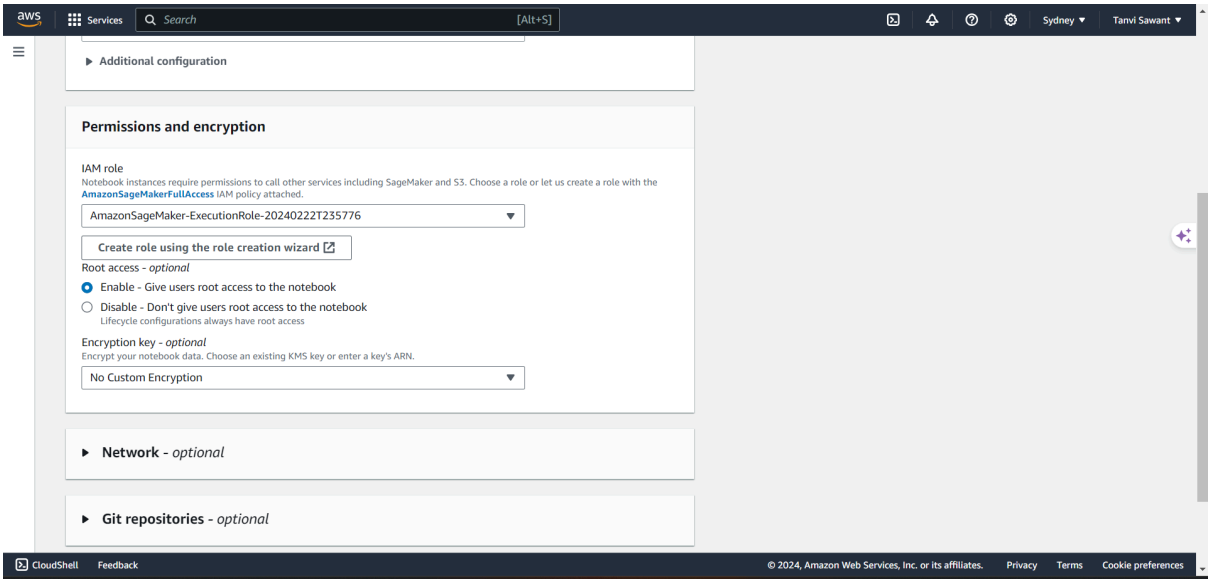
Networking Configuration (Optional):

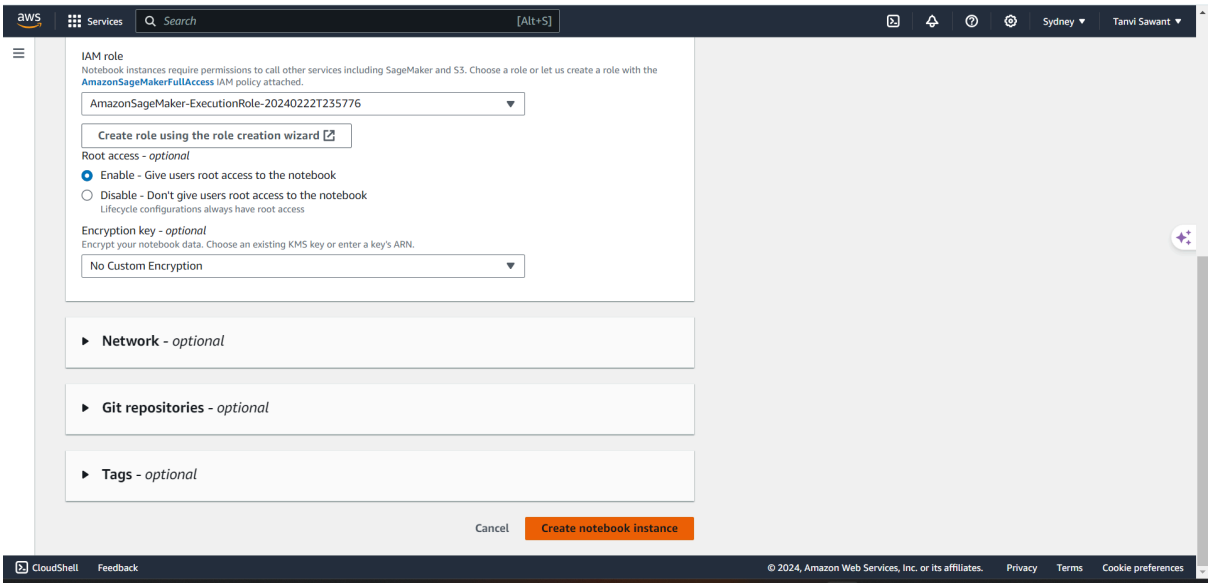
You can choose a VPC (Virtual Private Cloud) and Subnet for your notebook instance if you have specific networking requirements. For simplicity, you can leave this as default.

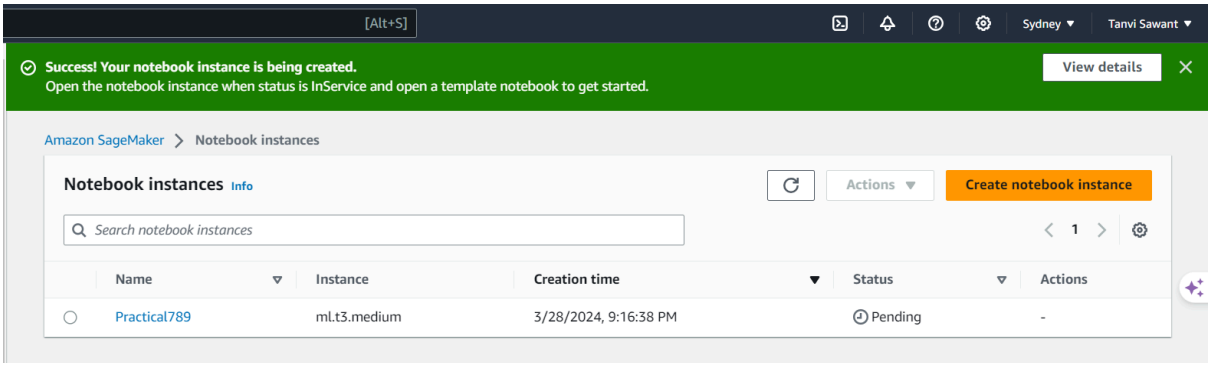
Create Notebook Instance:

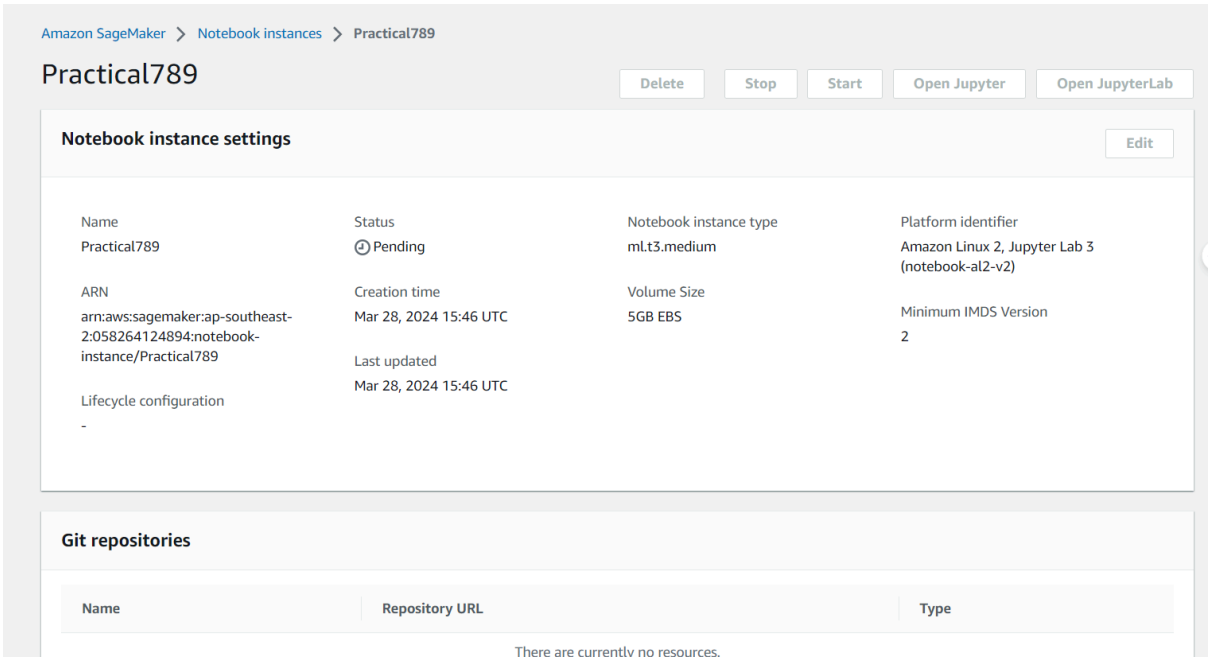
Once you've configured the settings according to your preferences, scroll down and click on the "Create notebook instance" button at the bottom.

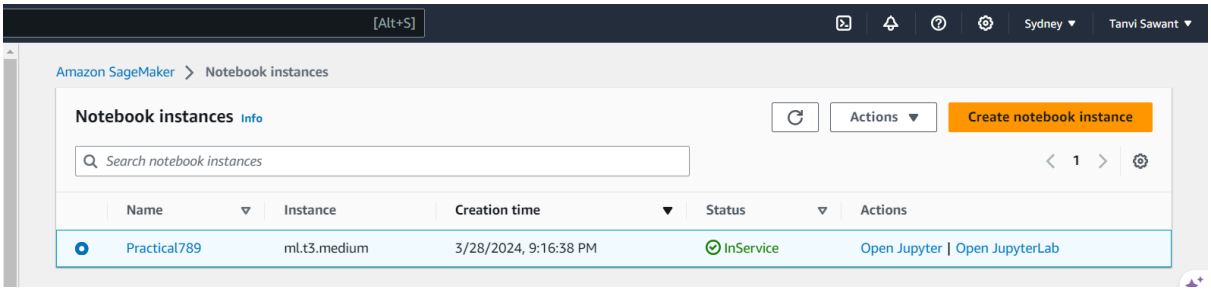
Amazon SageMaker will now provision your notebook instance, which may take a few minutes.





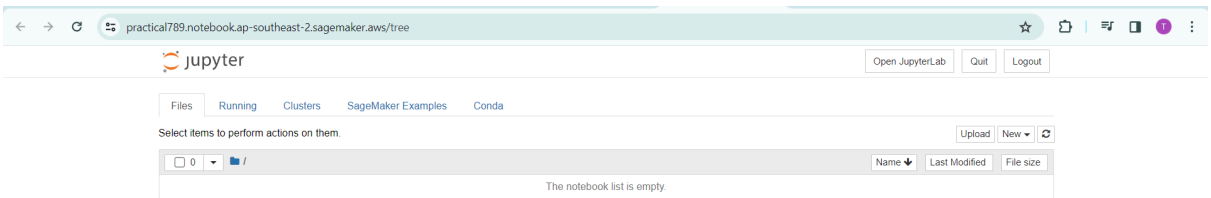






Open Jupyter Notebook:

- Once the notebook instance is created (it may take a few minutes), click on "Open Jupyter" next to your notebook instance name.
- This will open a Jupyter Notebook interface in a new tab.



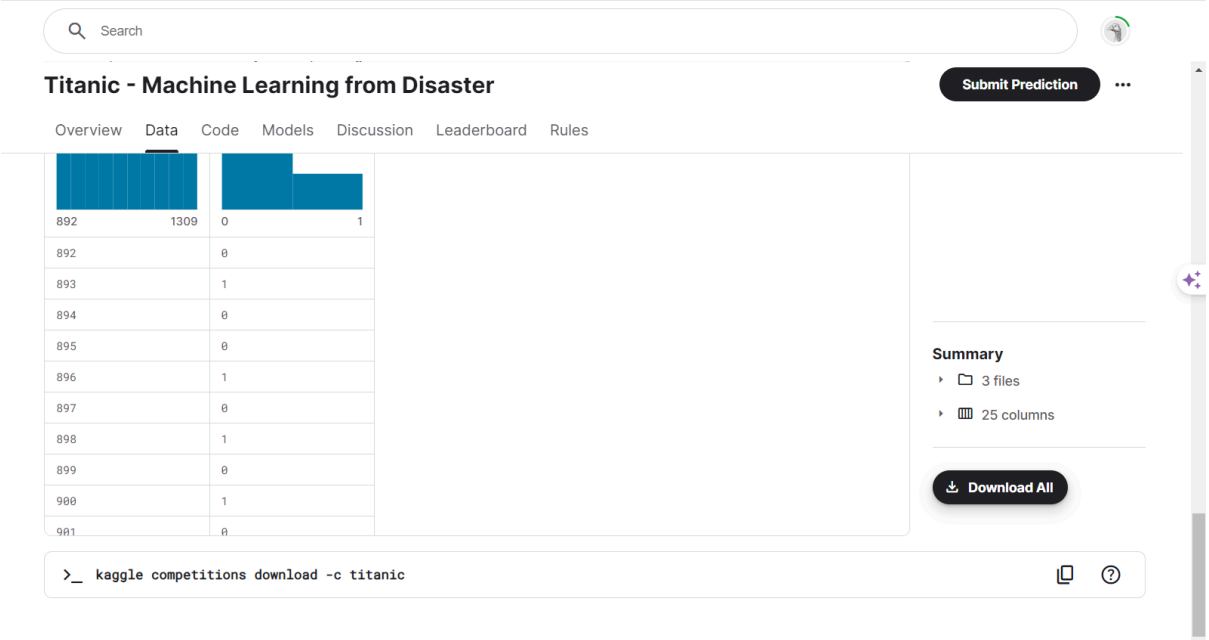
Dataset: Titanic: Machine Learning from Disaster

This dataset contains information about passengers aboard the Titanic, including details like age, gender, passenger class, ticket fare, and whether they survived the disaster. It's commonly used for classification tasks and is suitable for practicing data preprocessing, model building, and evaluation.

Here are the steps to download the dataset from Kaggle:

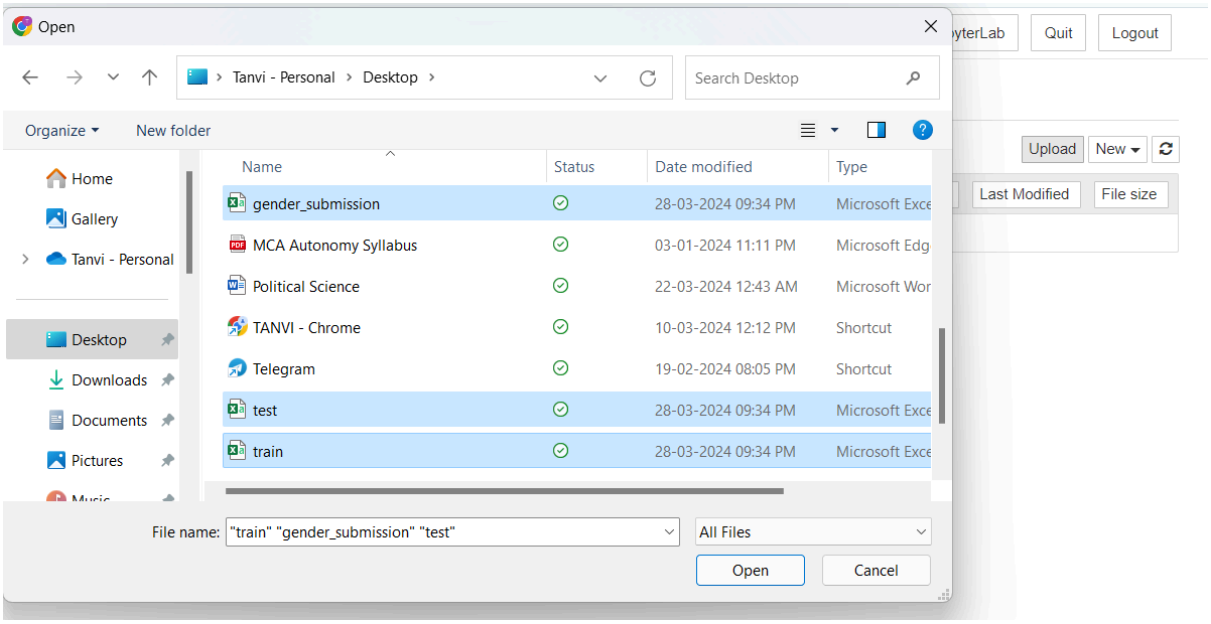
1. Sign in to your Kaggle account or create a new one.
2. Navigate to the Titanic competition page.
3. Scroll down to the "Data" section.
4. Click on the "Download All" button to download the entire dataset as a zip file.
5. Extract the zip file to access the dataset files (e.g., train.csv, test.csv).

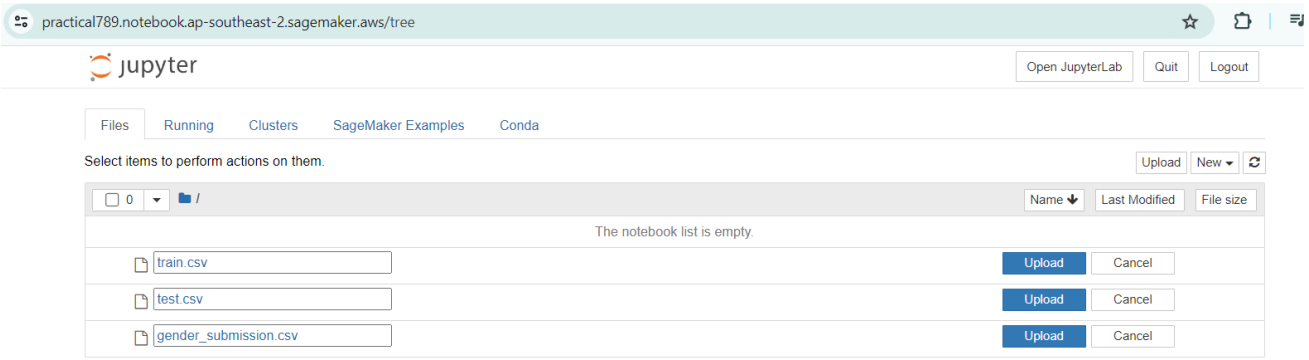
Once you have the dataset files, you can upload them to your Jupyter Notebook environment on AWS SageMaker as explained earlier. This dataset is well-suited for practicing data preprocessing, model training, and evaluation tasks.



Prepare Your Data:

- Upload the dataset files (**train.csv**, **test.csv**, **gender_submission.csv**) to your Jupyter Notebook environment.





- Use pandas to load the datasets into separate DataFrames.
- Perform any necessary data cleaning and preprocessing steps.

Create a New Notebook:

- In the Jupyter Notebook interface, click on the "New" button on the top right corner.
- Select "Python 3" to create a new Python notebook.

Prepare Your Data:

- In this step, you'll upload your datasets (test.csv, train.csv, gender_submission.csv) to your Jupyter Notebook environment and start working with them.

In [16]:
```
import pandas as pd

# Upload the datasets
train_data = pd.read_csv('train.csv')
test_data = pd.read_csv('test.csv')
submission_data = pd.read_csv('gender_submission.csv')

# Now you can start working with your datasets
```

What is XGBoost Algorithm?

XGBoost is a robust machine-learning algorithm that can help you understand your data and make better decisions.

XGBoost is an implementation of gradient-boosting decision trees. It has been used by data scientists and researchers worldwide to optimize their machine-learning models.

What is XGBoost in Machine Learning?

XGBoost is designed for speed, ease of use, and performance on large datasets. It does not require optimization of the parameters or tuning, which means that it can be used immediately after installation without any further configuration.

XGBoost Features

XGBoost is a widespread implementation of gradient boosting.

❖ XGBoost offers regularization, which allows you to control overfitting by introducing L1/L2 penalties on the weights and biases of each tree.

❖ Another feature of XGBoost is its ability to handle sparse data sets using the weighted quantile sketch algorithm. This algorithm allows us to deal with non-zero entries in the feature matrix while retaining the same computational complexity as other algorithms.

❖ XGBoost also has a block structure for parallel learning. It makes it easy to scale up on multicore machines or clusters. It also uses cache awareness, which helps reduce memory usage when training models with large datasets.

❖ XGBoost offers out-of-core computing capabilities using disk-based data structures instead of in-memory ones during the computation phase.

XgBoost Formula

The XGBoost algorithm builds an ensemble of decision trees in a sequential manner, where each subsequent tree corrects the errors made by the previous ones. The final prediction is a weighted sum of the predictions from all the trees in the ensemble. The general formula for predicting the output of a new data point using XGBoost can be expressed as follows

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i)$$

where:

- $\hat{y}_i$ is the predicted output for the i-th data point.
- $\mathbf{x}_i$ represents the feature vector for the i-th data point.
- K is the total number of trees in the ensemble.
- $f_k(\mathbf{x}_i)$ is the prediction made by the k-th tree for the i-th data point.

Conclusion: Overall, XGBoost is widely used in both academia and industry due to its excellent performance, versatility, and efficiency in handling various machine learning tasks. Performing data splitting, hyperparameter search optimization, and hyperparameter optimization strategies in AWS SageMaker involves several steps. Below is a detailed guide on how to perform these tasks:

1. Data Split for SageMaker.

Before training a machine learning model, it is essential to split the dataset into training, validation, and test sets. SageMaker provides utilities to perform this data split.

```
import sagemaker
from sagemaker import get_execution_role
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
```

```
# Assume you have a dataframe df with features and target
# Split data into train, validation, and test sets
train_data, temp_data = train_test_split(df, test_size=0.3, random_state=42)
val_data, test_data = train_test_split(temp_data, test_size=0.5, random_state=42)

# Save the data to CSV files
train_data.to_csv('train.csv', index=False, header=False)
val_data.to_csv('validation.csv', index=False, header=False)
test_data.to_csv('test.csv', index=False, header=False)

# Upload the data to S3
role = get_execution_role()
sess = sagemaker.Session()
bucket = sess.default_bucket()

train_data_s3_path = sess.upload_data(path='train.csv', bucket=bucket, key_prefix='data')
val_data_s3_path = sess.upload_data(path='validation.csv', bucket=bucket, key_prefix='data')
test_data_s3_path = sess.upload_data(path='test.csv', bucket=bucket, key_prefix='data')
```

2. Hyperparameter Search Optimization

SageMaker provides built-in hyperparameter optimization (HPO) to automate the process of finding the best hyperparameters for your model.

```
from sagemaker.tuner import HyperparameterTuner, IntegerParameter, ContinuousParameter

# Define hyperparameter ranges
hyperparameter_ranges = {
    'learning_rate': ContinuousParameter(0.001, 0.1),
    'num_trees': IntegerParameter(50, 200),
    'max_depth': IntegerParameter(5, 15)
}
```

```
# Create an estimator
estimator = sagemaker.estimator.Estimator(image_uri='image-uri', # specify your image uri
                                          role=role,
                                          instance_count=1,
                                          instance_type='ml.m5.large',
                                          hyperparameters={'objective': 'regression'},
                                          output_path='s3://{}/output'.format(bucket))

# Create a HyperparameterTuner object
tuner = HyperparameterTuner(estimator=estimator,
                            objective_metric_name='validation:rmse', # specify the metric to optimize
                            objective_type='Minimize',
                            hyperparameter_ranges=hyperparameter_ranges,
                            max_jobs=20,
                            max_parallel_jobs=4)

# Fit the tuner
tuner.fit({'train': train_data_s3_path, 'validation': val_data_s3_path})
```

3. Hyperparameter Optimization Strategies

AWS SageMaker's HPO supports multiple optimization strategies, such as random search, Bayesian optimization, and more. The strategy parameter in the HyperparameterTuner can be used to specify the optimization strategy.

1. Random Search

Random search is a simple and effective method for hyperparameter optimization. SageMaker's HyperparameterTuner supports random search as one of its optimization strategies.

```
from sagemaker.tuner import HyperparameterTuner, ContinuousParameter, IntegerParameter

# Define hyperparameter ranges
hyperparameter_ranges = {
    'learning_rate': ContinuousParameter(0.001, 0.1),
    'num_trees': IntegerParameter(50, 200),
    'max_depth': IntegerParameter(5, 15)
}
```

```
# Create an estimator
estimator = sagemaker.estimator.Estimator(image_uri='image-uri', # specify your image uri
                                          role=role,
                                          instance_count=1,
                                          instance_type='ml.m5.large',
                                          hyperparameters={'objective': 'regression'},
                                          output_path='s3://{}/output'.format(bucket))

# Create a HyperparameterTuner object with random search strategy
tuner_random = HyperparameterTuner(estimator=estimator,
                                   objective_metric_name='validation:rmse',
                                   objective_type='Minimize',
                                   hyperparameter_ranges=hyperparameter_ranges,
                                   max_jobs=20,
                                   max_parallel_jobs=4,
                                   strategy='Random')

# Fit the tuner
tuner_random.fit({'train': train_data_s3_path, 'validation': val_data_s3_path})
```

2. Bayesian Optimization

Bayesian optimization is an advanced method for hyperparameter optimization, which builds a probabilistic model of the objective function and uses it to select the most promising hyperparameters

```
# Create a HyperparameterTuner object with Bayesian optimization strategy
tuner_bayesian = HyperparameterTuner(estimator=estimator,
                                     objective_metric_name='validation:rmse',
                                     objective_type='Minimize',
                                     hyperparameter_ranges=hyperparameter_ranges,
                                     max_jobs=20,
                                     max_parallel_jobs=4,
                                     strategy='Bayesian')

# Fit the tuner
tuner_bayesian.fit({'train': train_data_s3_path, 'validation': val_data_s3_path})
```

3. Customized Optimization Strategy

You can also implement a customized optimization strategy by subclassing the HyperparameterTunerStrategy class and implementing the required methods.

```
from sagemaker.tuner import HyperparameterTunerStrategy

class CustomStrategy(HyperparameterTunerStrategy):
    def hyperparameter_tuning_job_config(self):
        config = super(CustomStrategy, self).hyperparameter_tuning_job_config()
        config['HyperParameterTuningJobConfig']['Strategy'] = 'Custom'
        return config

# Create a HyperparameterTuner object with custom strategy
tuner_custom = HyperparameterTuner(estimator=estimator,
                                   objective_metric_name='validation:rmse',
                                   objective_type='Minimize',
                                   hyperparameter_ranges=hyperparameter_ranges,
                                   max_jobs=20,
                                   max_parallel_jobs=4,
                                   strategy=CustomStrategy)

# Fit the tuner
tuner_custom.fit({'train': train_data_s3_path, 'validation': val_data_s3_path})
```

4. Bias-Variance Tradeoff

The bias-variance tradeoff is a fundamental concept in machine learning that relates to the model's ability to generalize to unseen data. A high-bias model is too simplistic and may underfit the training data (high training error), whereas a high-variance model is too complex and may overfit the training data (low training error but high validation error).

To manage the bias-variance tradeoff, you can adjust hyperparameters and model complexity. SageMaker's Hyperparameter Optimization (HPO) can help find the optimal hyperparameters that balance bias and variance by searching for the hyperparameter values that result in the best validation performance.

1. Define Hyperparameter Ranges

```
from sagemaker.tuner import HyperparameterTuner, ContinuousParameter, IntegerParameter

# Define hyperparameter ranges
hyperparameter_ranges = {
    'learning_rate': ContinuousParameter(0.001, 0.1),
    'num_trees': IntegerParameter(50, 200),
    'max_depth': IntegerParameter(5, 15)
}
```

2. Create an Estimator

```
import sagemaker

# Create an estimator
estimator = sagemaker.estimator.Estimator(image_uri='image-uri', # specify your image uri
                                          role=role,
                                          instance_count=1,
                                          instance_type='ml.m5.large',
                                          hyperparameters={'objective': 'regression'},
                                          output_path='s3://{}/output'.format(bucket))
```

3. Create a HyperparameterTuner Object

```
# Create a HyperparameterTuner object
tuner = HyperparameterTuner(estimator=estimator,
                            objective_metric_name='validation:rmse',
                            objective_type='Minimize',
                            hyperparameter_ranges=hyperparameter_ranges,
                            max_jobs=20,
                            max_parallel_jobs=4,
                                strategy='Bayesian')  # Use Bayesian optimization as the optimization strategy
```

4. Fit the HyperparameterTuner

```
# Fit the tuner
tuner.fit({'train': train_data_s3_path, 'validation': val_data_s3_path})
```

5. Analyze the Results

After the hyperparameter tuning job is completed, you can analyze the results to understand how different hyperparameters affect the bias-variance tradeoff. You can use the Amazon SageMaker console or SDK to analyze the results and identify the optimal hyperparameters that balance bias and variance.

```
# Get the best hyperparameters
best_hyperparameters = tuner.best_estimator().hyperparameters()

print("Best hyperparameters:")
print(best_hyperparameters)
```

5. L2 Regularization (Ridge Regression) and L1 Regularization (Lasso Regression)

```
hyperparameters={
    'alpha': 1.0,  # L2 regularization parameter
    'objective': 'reg:linear'
}

hyperparameters={
    'alpha': 1.0,  # L1 regularization parameter
    'objective': 'reg:linear',
    'lambda': 1.0  # L2 regularization parameter (for Lasso in XGBoost)
}
```

6. Hyperparameters Optimization Using GridSearchCV

```
from sklearn.model_selection import GridSearchCV
from sagemaker.sklearn.estimator import SKLearn

# Define hyperparameter grid
param_grid = {
    'alpha': [0.1, 1.0, 10.0],
    'learning_rate': [0.001, 0.01, 0.1],
    'max_depth': [3, 5, 7],
```

```
    'n_estimators': [50, 100, 200]
}

# Create a SKLearn estimator
sklearn_estimator = SKLearn(entry_point='script.py', # your script
                            role=role,
                            instance_count=1,
                            instance_type='ml.m5.large',
                            framework_version='0.23-1',
                            sagemaker_session=sess)

# Create GridSearchCV object
grid_search = GridSearchCV(estimator=sklearn_estimator,
                           param_grid=param_grid,
                           cv=3,
                           scoring='neg_mean_squared_error',
                           n_jobs=-1)

# Fit GridSearchCV
grid_search.fit({'train': train_data_s3_path})
```

Conclusion

In this guide, we explored the essential aspects of hyperparameter optimization (HPO) in AWS SageMaker, focusing on managing the bias-variance tradeoff, which is a fundamental concept in machine learning.

We began by defining the hyperparameter ranges that we wanted to optimize, such as the learning rate, number of trees, and maximum depth. These hyperparameters play a crucial role in determining the complexity of the model and, consequently, its ability to generalize to unseen data.

Next, we created an estimator using SageMaker's Estimator class, specifying the necessary configurations such as the Docker image URI, IAM role, instance type, and hyperparameters.

We then utilized SageMaker's HyperparameterTuner to perform the hyperparameter optimization. We selected Bayesian optimization as the optimization strategy, which is an advanced method that builds a probabilistic model of the objective function to select the most promising hyperparameters. The HyperparameterTuner automatically searched for the

optimal hyperparameters within the defined ranges, aiming to minimize the validation root mean squared error (RMSE).

After fitting the HyperparameterTuner, we analyzed the results to retrieve the best hyperparameters that balance the bias-variance tradeoff effectively. By understanding how different hyperparameters affect the model's generalization performance, we can make informed decisions to adjust the model's complexity and improve its ability to generalize to unseen data.

In conclusion, AWS SageMaker's Hyperparameter Optimization provides a powerful and efficient way to find the optimal hyperparameters that balance bias and variance, thereby enhancing the model's performance and generalization capabilities. By carefully tuning the hyperparameters and analyzing the results, machine learning practitioners can develop models that deliver superior performance on real-world tasks.