Queen's

Master of Management in Artificial Intelligence

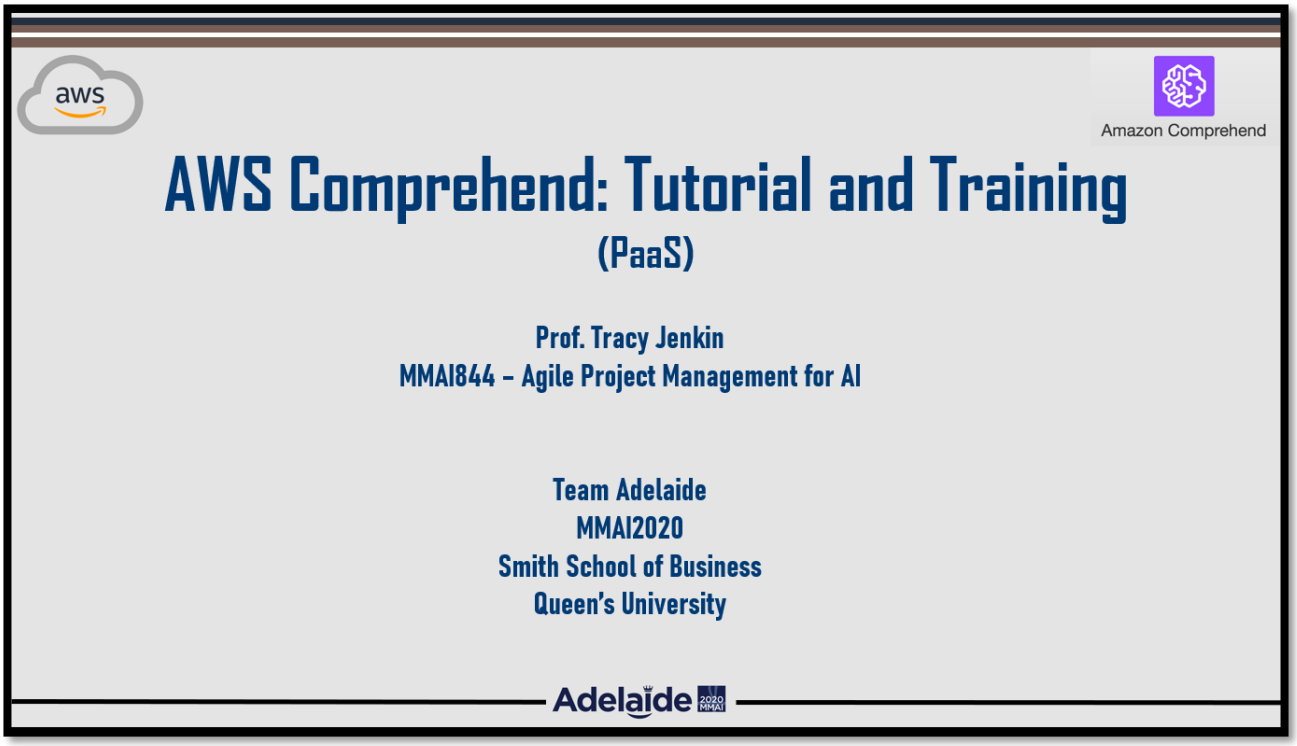
MMAI 844
Agile Project Management for AI

Dr. Tracy Jenkin

AWS Comprehend: Tutorial and Training
July 11, 2020

Team Adelaide:
Francis Bello
Tiancheng Qu
Alexander Bakus
Mamta Gupta
Amit Kumar
Gunpreet Singh
Keerthi Sukhavasi
Section 1

AWS Comprehend: Tutorial and Training (PaaS)

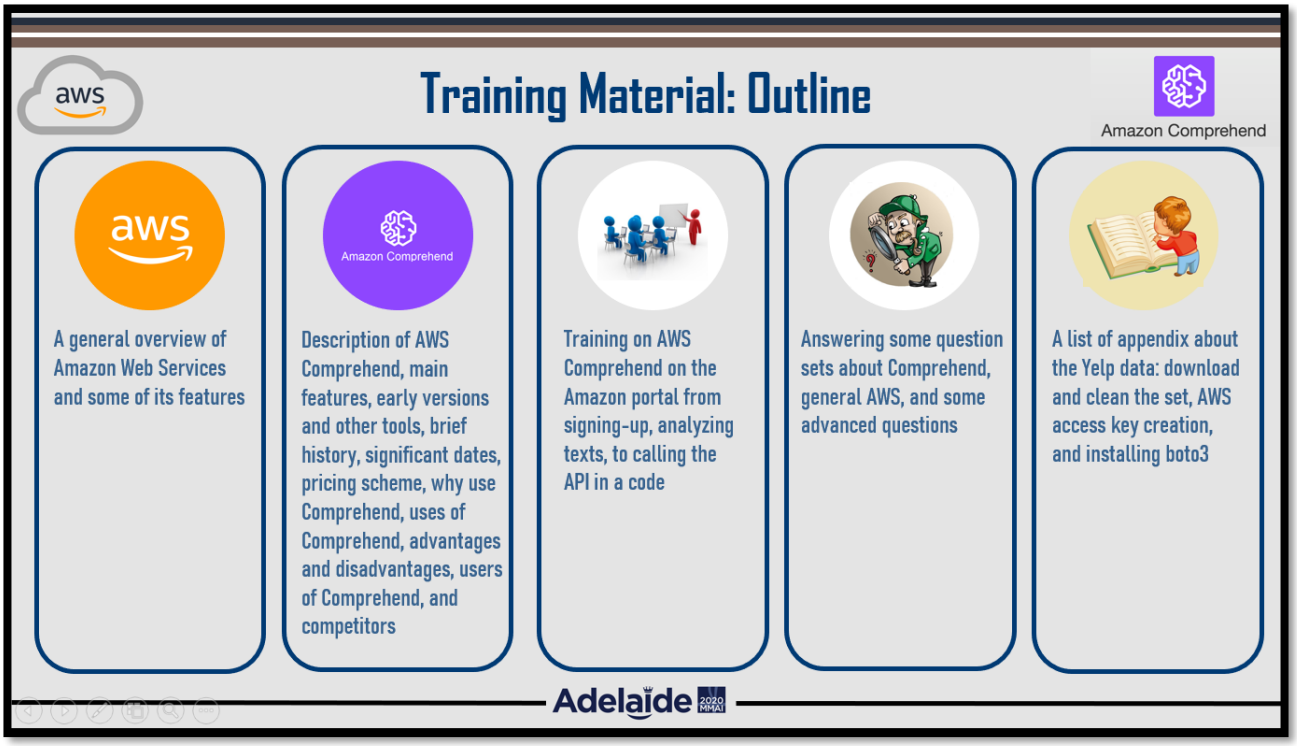


This training material was made for Prof. Tracy Jenkin and students of MMAI taking up Agile Project Management for AI.

For technical support, please send us an email at:
group-teamadelaide@queensu.ca

An online video of this training material is available on YouTube:
https://youtu.be/Zf7sxXnXtHg

A copy of this training material can also be downloaded from:
https://github.com/adelaide-mmai2020/mmai844/blob/master/aws_comprehend_tutorial.pdf

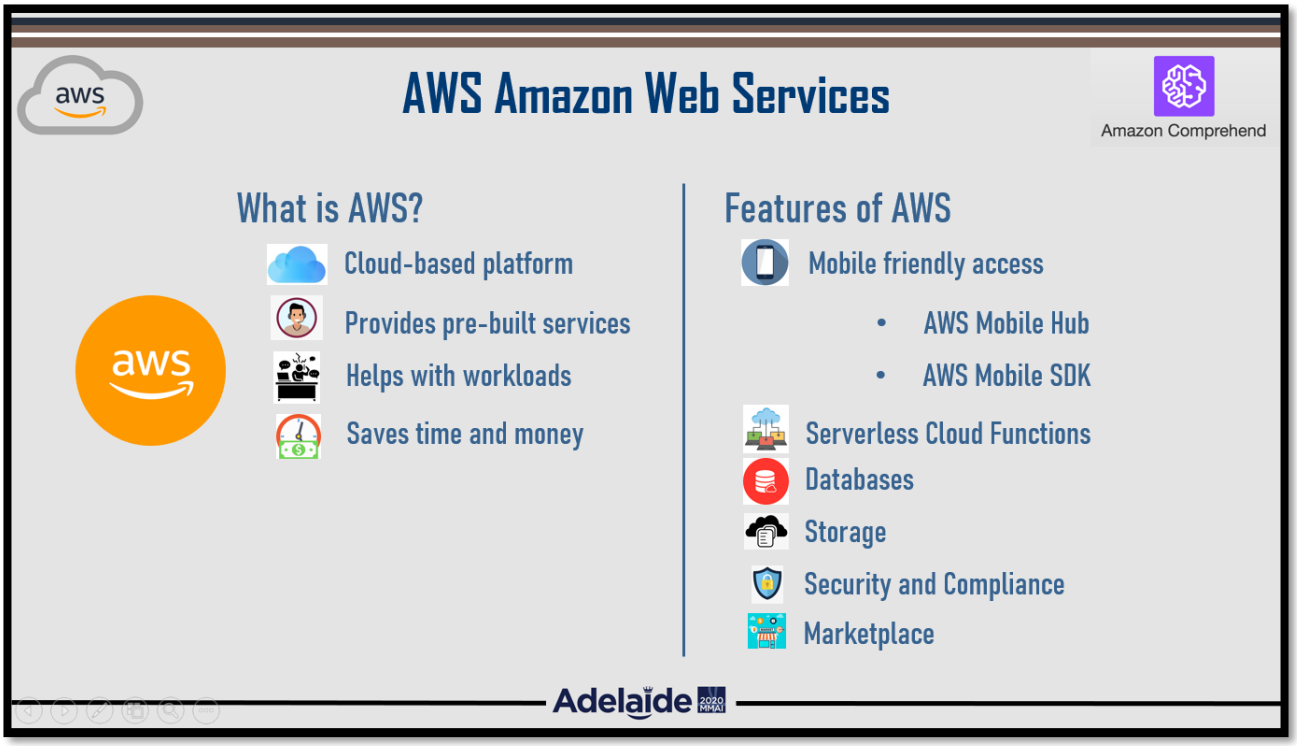


The training material is outlined as follows:

Part –

A. A general overview of AWS and some of its features.
B. A description of AWS Comprehend – description, features, early versions and other tools, etc.
C. Training on AWS Comprehend on Amazon portal from signing-up, analyzing texts, to calling the API in a code.
D. Trivia questions about Comprehend, general AWS, and some advanced questions.
E. Appendix about the Yelp data, AWS access key creation, and installing boto3.

Part A: Amazon Web Services



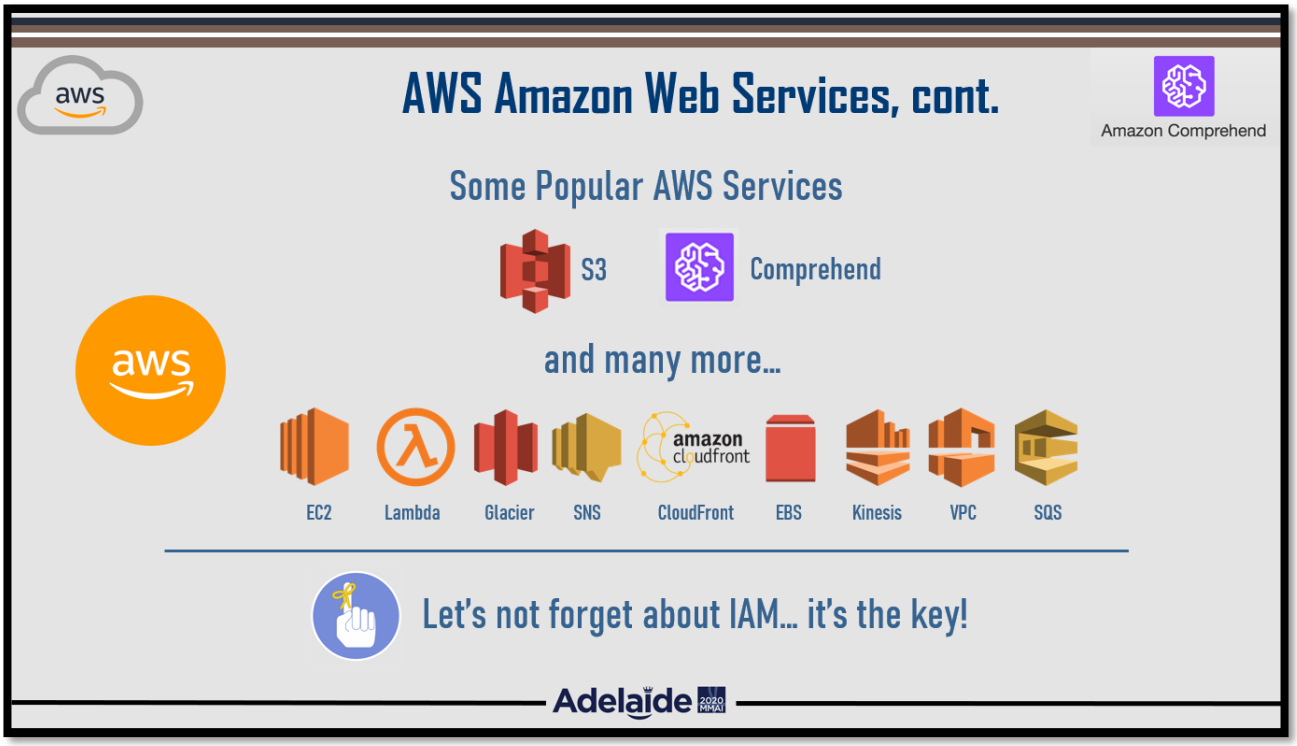
What is AWS?

- Amazon's cloud-based platform used for building solutions for businesses using inter-connected web services.
- Provides a variety of pre-built services which can benefit businesses to build their custom cloud-based solutions.
- Helps companies with a wide variety of workloads such as game development, data processing, warehousing, achieve, development and many more.
- Provides a feature which allows you to choose servers according to your choice which ultimately saves time and money.

Features of AWS:

- Mobile Friendly Access
    - AWS Mobile Hub
        - For Android and IOS
        - Supports and guides towards the suitable and compatible feature for any app
    - AWS Mobile SDK
        - Supports Android, IOS, Web, React Native, Unity and many more
        - This feature allows app to directly access AWS such as DynamoDB, S3 and Lambda
- Serverless Cloud Functions

    - Allows developers to deliver a great experience to their users by providing a back-end code at once to them that responds to their code
    - Amazon API and Amazon Gateway help users run, scale, patch and administer their codes
- Databases
    - All the database service is managed by AWS
        - Relational database – transactional purposes
        - Non-relational database – internet scale applications
        - Data Warehouse – analytics
        - In-memory Data Store – caching & real-time workloads
        - Graph Database – application with highly connected data
- Storage
    - Economical, flexible and easy to use
    - Can be used independently as well as combined to meet any requirement
- Security and Compliance
    - Provides maximum data security
    - Allows customers to scale and innovate
    - Provides security at the protocol and port access level
- Marketplace
    - An online store where a customer can immediately search for the suitable software for his/her business
    - Provides a one-click deployment of the software



Some Popular AWS Services

S3
AWS offers a wonderful service called Amazon Simple Storage Service or Amazon S3 to store and retrieve data from the cloud. S3 allows the user to store, upload, retrieve large files up to 5 TB from the cloud. It is a scalable, low-cost and high-speed web-based service designed for archival and online backup of application programs and data.

Comprehend
Amazon Comprehend is a natural language processing (NLP) service that uses machine learning to find insights and relationships in text. It uses machine learning to help you uncover the insights and relationships in your unstructured data. The service identifies the language of the text; extracts key phrases, places, people, brands, or events; understands how positive or negative the text is; analyzes text using tokenization and parts of speech; and automatically organizes a collection of text files by topic.

In this training literature, we would be learning about:

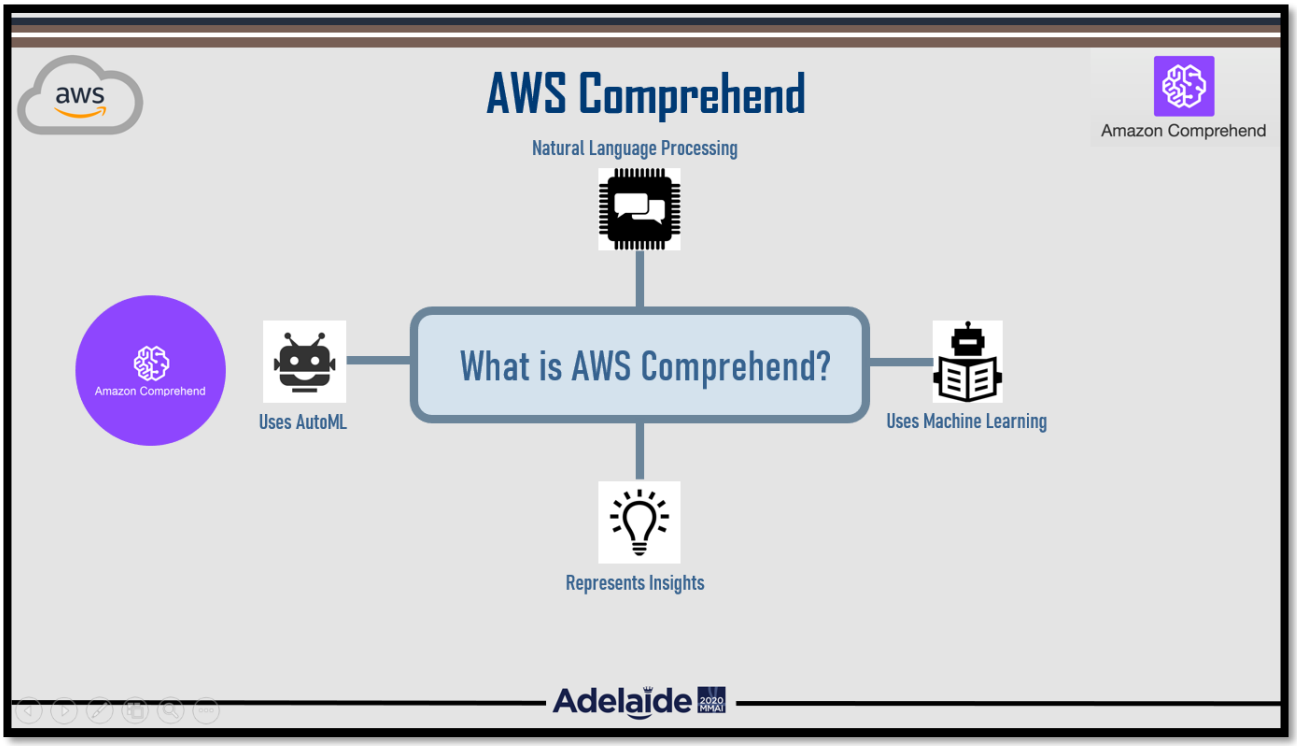
- how to perform sentiment analysis using the AWS Comprehend console
- as well as how to call the AWS Comprehend API in Python/Jupyter codes to perform
    - dominant language detection
    - named entities detection
    - key phrases detection
    - sentiment analysis

- syntax detection

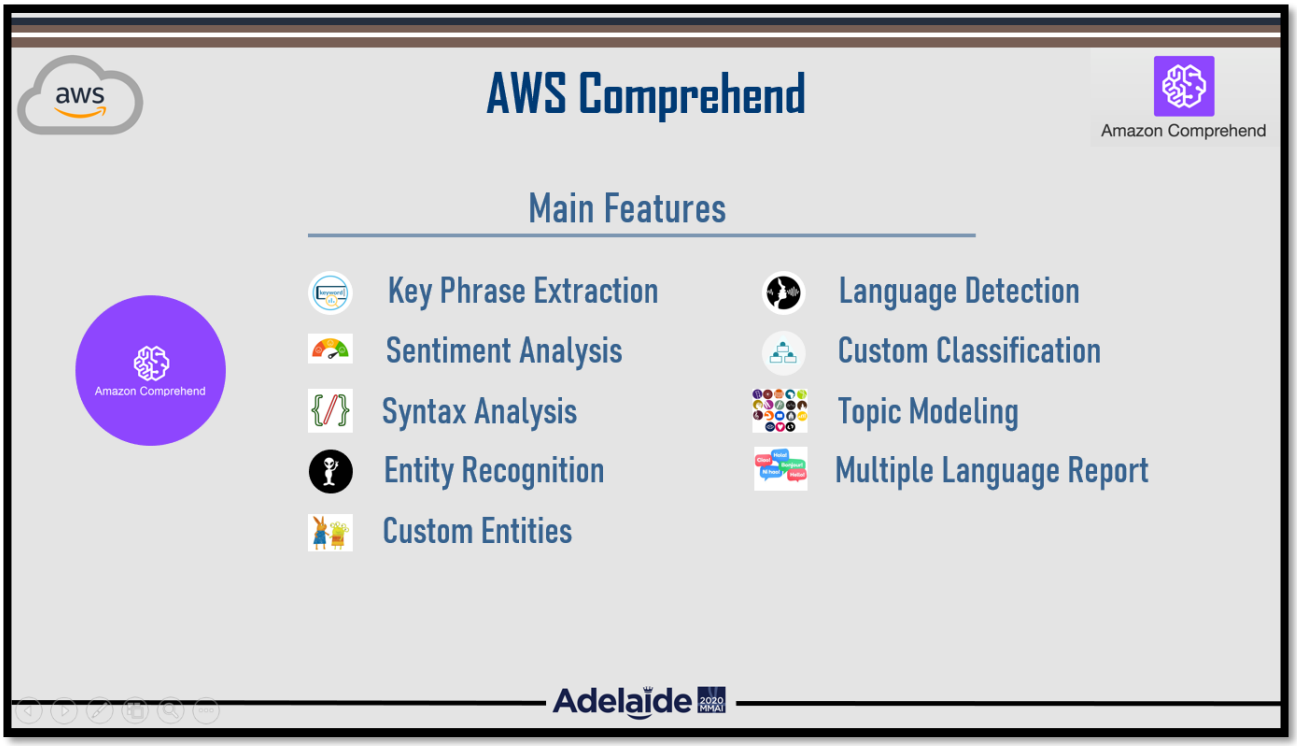
IAM

- Enables to manage access to AWS services and resources securely.
- Using IAM, users can create and manage AWS other users and groups, and use permissions to allow and deny their access to AWS resources.
- IAM is a feature of AWS account offered at no additional charge.
- IAM permissions allows and denies user access to AWS resources.
- IAM is a feature of your AWS account offered at no additional charge.

Part B: AWS Comprehend



Amazon Comprehend is a natural language processing (NLP) service that uses machine learning to find insights and relationships in text. Amazon Comprehend uses machine learning to help you uncover the insights and relationships in your unstructured data like Customer emails, support tickets, product reviews, social media, advertising copy etc. This can be used to represent insights into customer sentiment that can be put to work for your business. The service identifies the language of the text; extracts key phrases, places, people, brands, or events; understands how positive or negative the text is; analyzes text using tokenization and parts of speech; and automatically organizes a collection of text files by topic. You can also use AutoML capabilities in Amazon Comprehend to build a custom set of entities or text classification models that are tailored uniquely to your organization's needs.



Main features of the service are:

Keyphrase Extraction: The Key Phrase extraction API returns the key phrases or talking points and a confidence score to support that this is a key phrase.

Sentiment Analysis: The Sentiment Analysis API returns the overall sentiment of a text (Positive, Negative, Neutral, or Mixed).

Syntax Analysis: The Amazon Comprehend Syntax API enables customers to analyze text using tokenization and Parts of Speech (PoS) and identify word boundaries and labels like nouns and adjectives within the text.

Entity Recognition: The Entity Recognition API returns the named entities ("People," "Places," "Locations," etc.) that are automatically categorized based on the provided text.

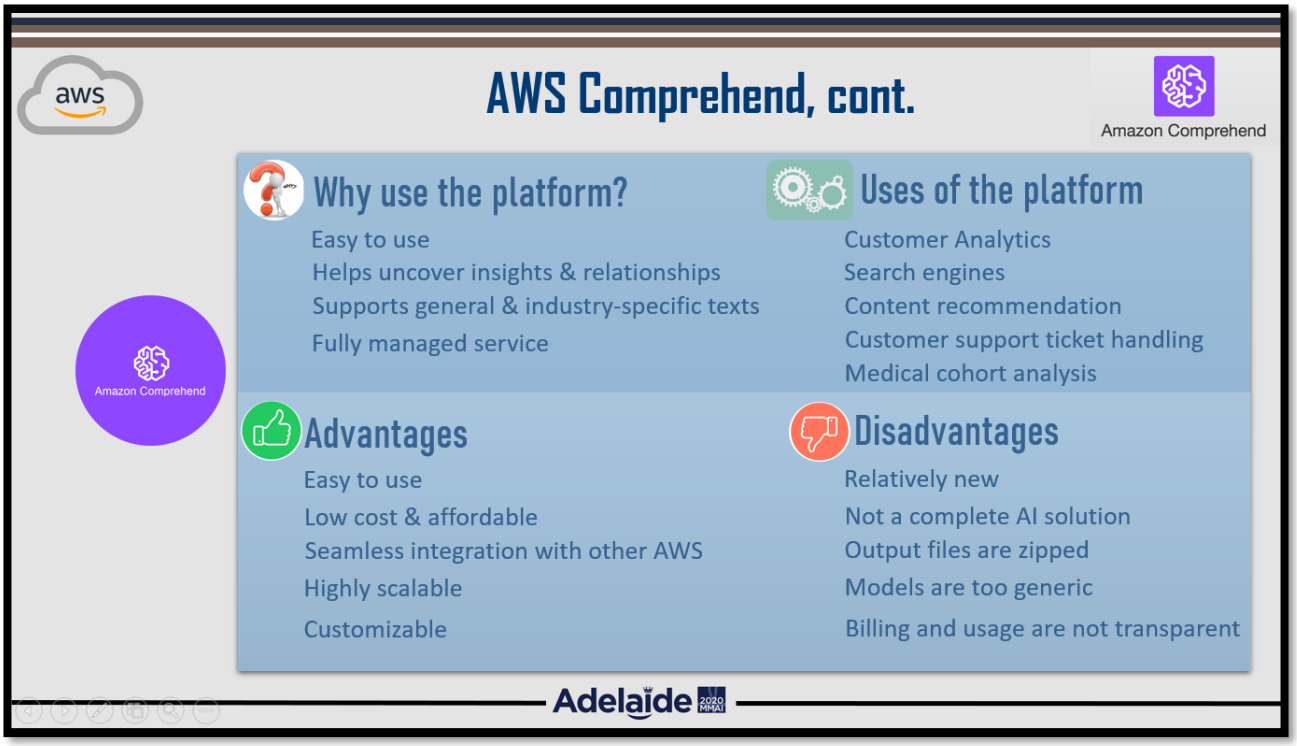
Custom Entities: Custom Entities allows you to customize Amazon Comprehend to identify terms that are specific to your domain. Using AutoML, Comprehend will learn from a small private index of examples (for example, a list of policy numbers and text in which they are used), and then train a private, custom model to recognize these terms in any other block of text. There are no servers to manage, and no algorithms to master.

Language Detection: The Language Detection API automatically identifies text written in over 100 languages and returns the dominant language with a confidence score to support that a language is dominant.

Custom Classification: The Custom Classification API enables you to easily build custom text classification models using your business-specific labels without learning ML.

Topic Modeling: Topic Modeling identifies relevant terms or topics from a collection of documents stored in Amazon S3. It will identify the most common topics in the collection and organize them in groups and then map which documents belong to which topic.

Multiple language support: Amazon Comprehend can perform text analysis on English, French, German, Italian, Portuguese, and Spanish texts. This lets you build applications that can detect text in multiple languages, convert the text to English, French, German, Italian, Portuguese, and Spanish with Amazon Translate, and then use Amazon Comprehend to perform text analysis.



Why use the platform?

- Easy to use NLP platform, minimal to no machine learning experience needed.
- Helps uncover the insights and relationships in unstructured data by identifying things such as:
    - The language of text,
    - Extracts key phrases, places, people, brands or events
    - Positive or negative sentiment, etc.
- Supports general and industry specific text such as email to complex medical information.
- Fully managed service therefore no servers to provision and no machine learning models to build, train, or deploy. Pay only the services used and there is no minimum fees.

Uses of the Platform

- Customer analytics – to analyze customer interactions in the form of support emails, social media posts, online comments, etc. and discover what factors drive positive and negative experiences.
- Search engines – provide better search experience by enabling it to index key phrases, entities, and sentiment. It helps focus the search on the intent and the context of the articles rather than generic keywords.

- Content recommendation – organize and categorize documents by topic for easier discovery, and then personalize content recommendations for readers by recommending other related articles.
- Customer support ticket handling – improve overall customer satisfaction by using custom classification to automatically categorize inbound customers support documents, such as feedback forms, support tickets, and product reviews. Then use custom entities to automatically extract relevant information like loyalty tiers to quickly route documents the team best equipped to solve the customer problem.
- Medical cohort analysis – understand and identify complex medical information found in unstructured text to help make indexing and searching easier and recruit patients to the appropriate clinical trial in a fraction of time.

Advantages

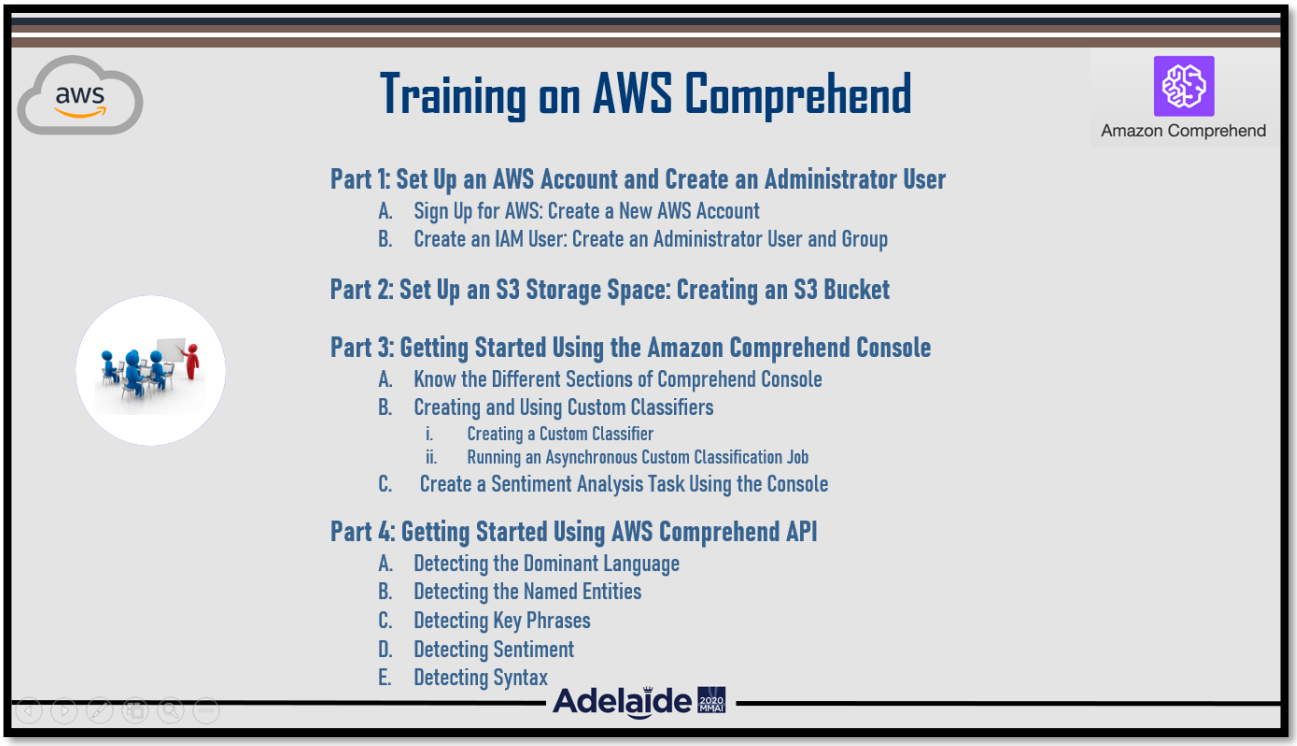
- AWS Comprehend is easy to use as it enables you to build text analysis capabilities such as language detection, sentiment analysis, topic modeling etc. into your applications with a simple API.
- AWS Comprehend is a low cost and affordable service. You only pay for the documents analyzed with no minimum fees or upfront charges.
- AWS Comprehend is designed to seamlessly integrate with other AWS Services such as Amazon S3, AWS Lambda, AWS KMS and so on.
- AWS Comprehend is a highly scalable service as it can analyze millions of documents across a variety of use cases such as customer support tickets, product reviews, social media, medical records and more.
- AWS Comprehend can be customized to suit your business needs with very little Machine Learning experience as it only requires labels and a small set of training examples.

Disadvantages

- Relatively new, as such the interface is missing the rich functionality that practitioners on NLP are accustomed to, and may leave users feeling that more can be done if only the tools were present. Great for starting but missing the depth of already well-known platforms.
- AWS Comprehend is not a complete AI solution, and requires a more DIY set up pace. Works well if you have clearly defined Text Analysis needs, but has limited capabilities to explore options with undefined purpose.
- The outputs from AWS Comprehend are produced in zip files, which need to be extracted and processed further for consumption.

- For Advanced users, the models are too generic, so if working with specific data, the results will leave much to be desired. The model selection is also limited, so if more are needed, the service cannot be relied on, and data teams have to go back to building their own and then deploying them themselves.
- The platform billing and consumption usage is not transparent, so it is difficult to estimate and forecast the monthly cost of using the platform full time.

Part C: Training on AWS Comprehend



We have divided this AWS Comprehend tutorial into 4 parts:

Part 1: Set Up an AWS Account and Create an Administrator User

A. Sign Up for AWS: Create a New AWS Account
B. Create an IAM User: Create an Administrator User and Group

Part 2: Set Up an S3 Storage Space: Creating an S3 Bucket

Part 3: Getting Started Using the Amazon Comprehend Console

A. Know the Different Sections of Comprehend Console
B. Creating and Using Custom Classifiers
    i. Creating a Custom Classifier
    ii. Running an Asynchronous Custom Classification Job
C. Create a Sentiment Analysis Task Using the Console

Part 4: Getting Started Using AWS Comprehend API

A. Detecting the Dominant Language
B. Detecting the Named Entities
C. Detecting Key Phrases
D. Detecting Sentiment
E. Detecting Syntax

Part 1: Set Up an AWS Account and Create an Administrator User

Sign Up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all AWS services, including Amazon Comprehend. You are charged only for the services that you use.

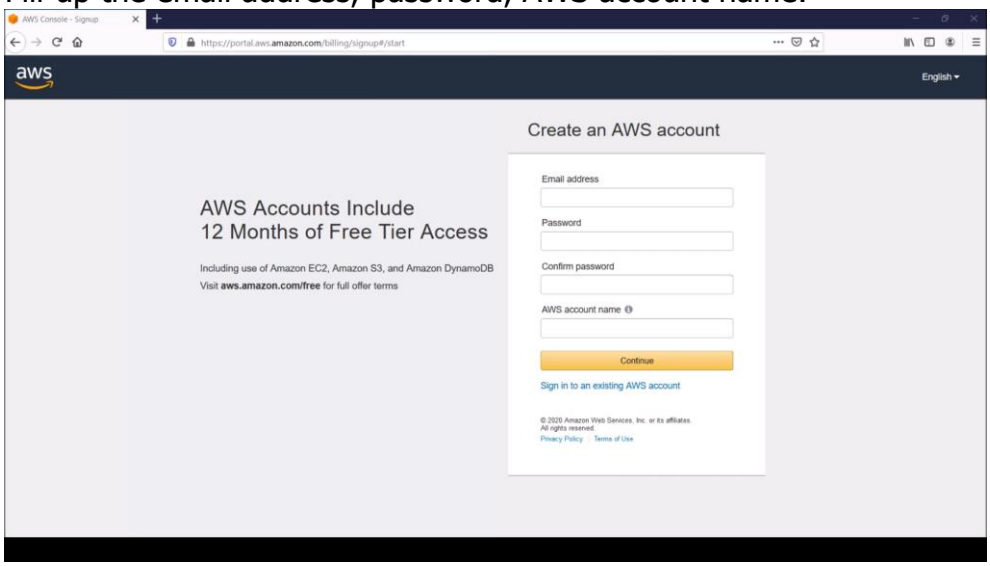
With Amazon Comprehend, you pay only for the resources that you use. If you are a new AWS customer, you can get started with Amazon Comprehend for free.

If you already have an AWS account, skip to the next section.

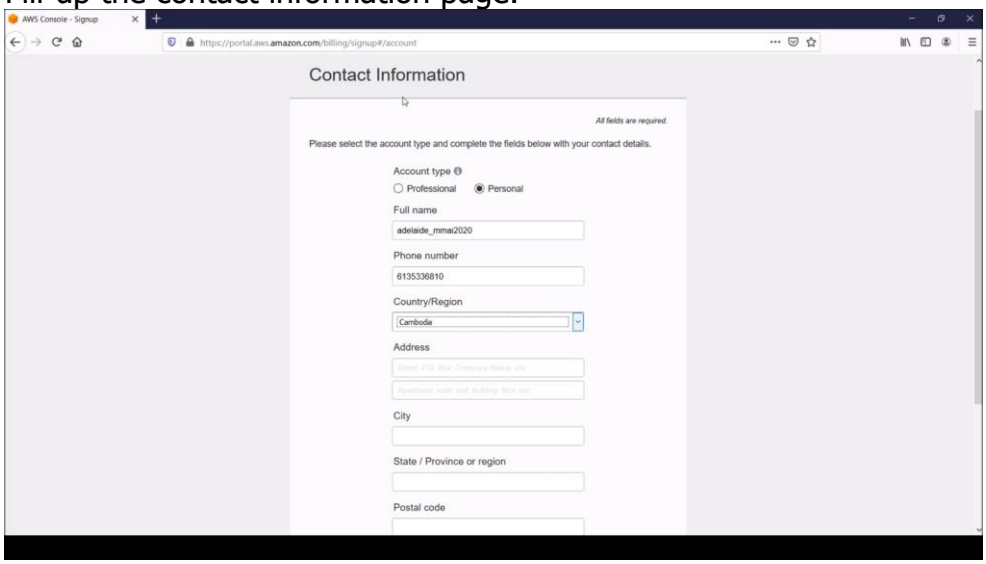
To create an AWS account

1. Open https://portal.aws.amazon.com/billing/signup
2. Follow the online instructions.

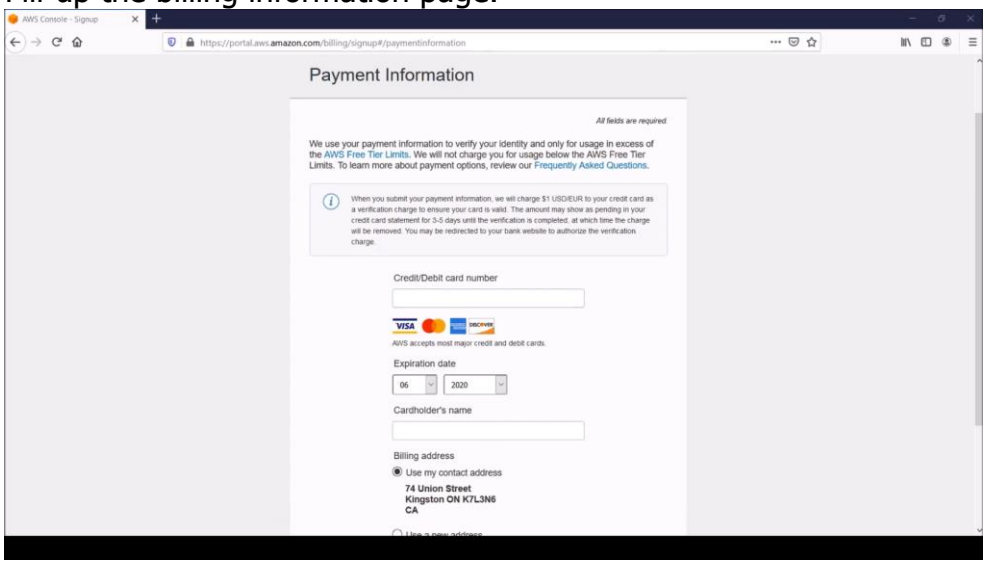
a. Fill-up the email address, password, AWS account name.



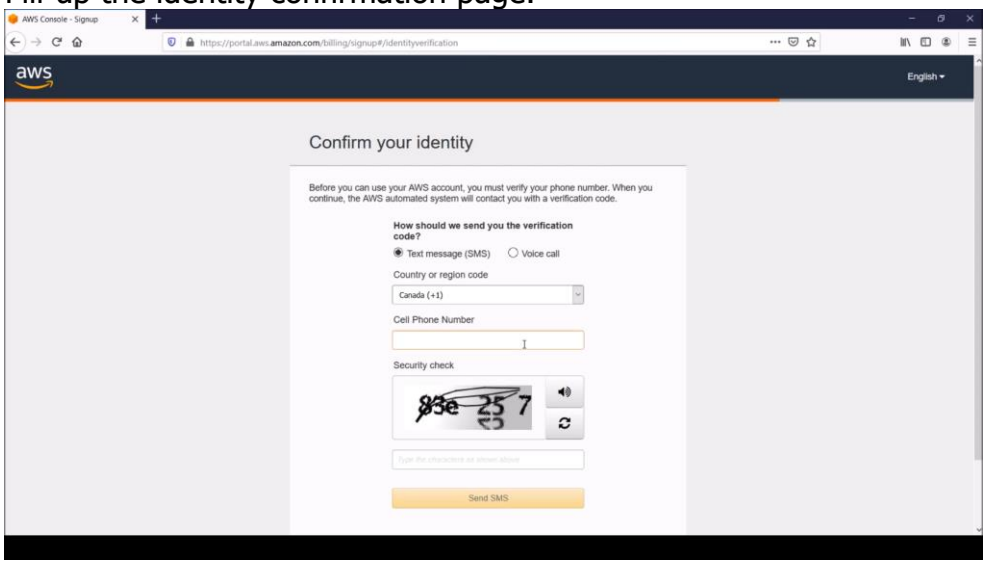
b. Fill-up the contact information page.



c. Fill-up the billing information page.



d. Fill-up the identity confirmation page.



Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Record your AWS account ID.

3. Now it's your turn! Let's apply what we've just learned about AWS Comprehend.

Create an IAM User

Services in AWS, such as Amazon Comprehend, require that you provide credentials when you access them. This allows the service to determine whether you have permissions to access the service's resources.

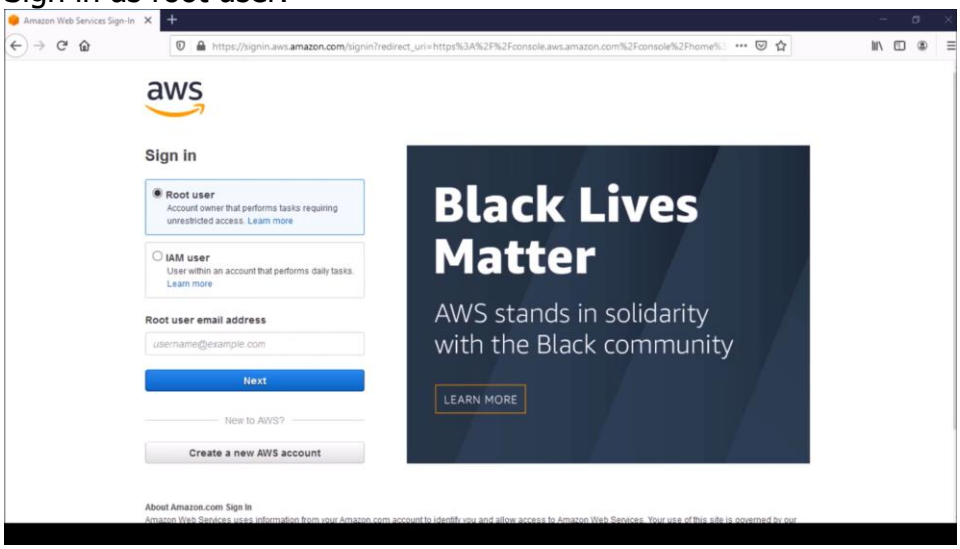
It is strongly recommended that you access AWS using AWS Identity and Access Management (IAM), not the credentials for your AWS account. To use IAM to access AWS, create an IAM user, add the user to an IAM group with administrative permissions, and then grant administrative permissions to the IAM user. You can then access AWS using a special URL and the IAM user's credentials.

The Getting Started we need a user with administrator privileges, “administrator”.

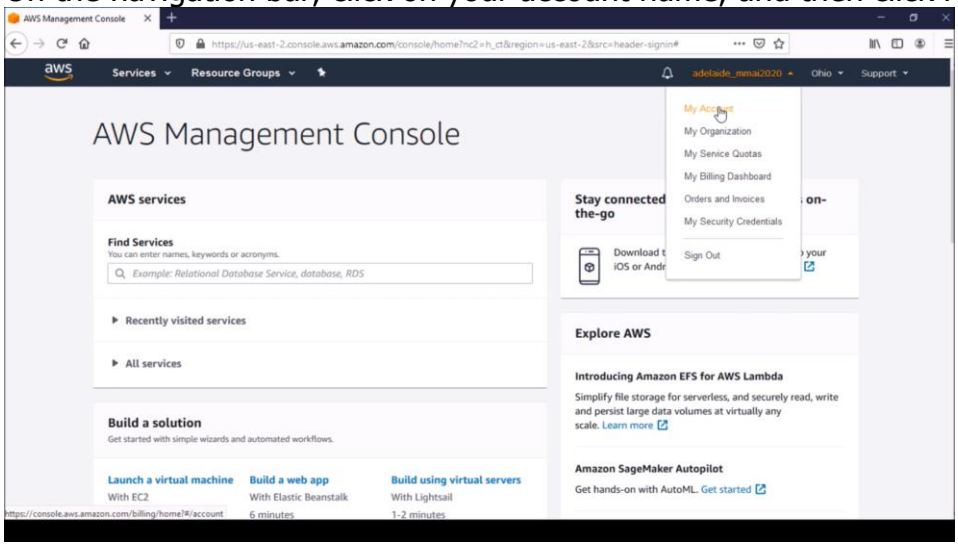
To create an administrator user and sign in to the console

1. Create an administrator user called “administrator” in your AWS account.

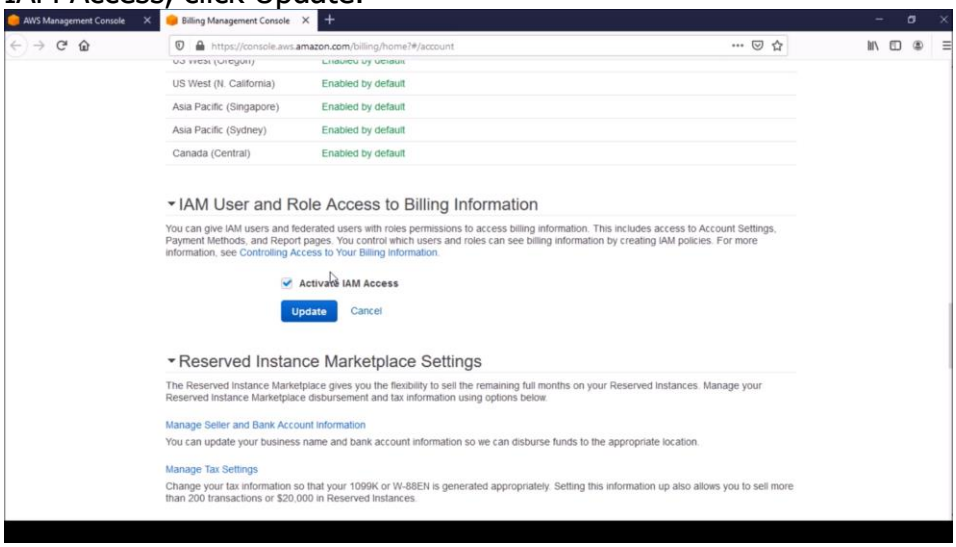
a. Sign in as root user.



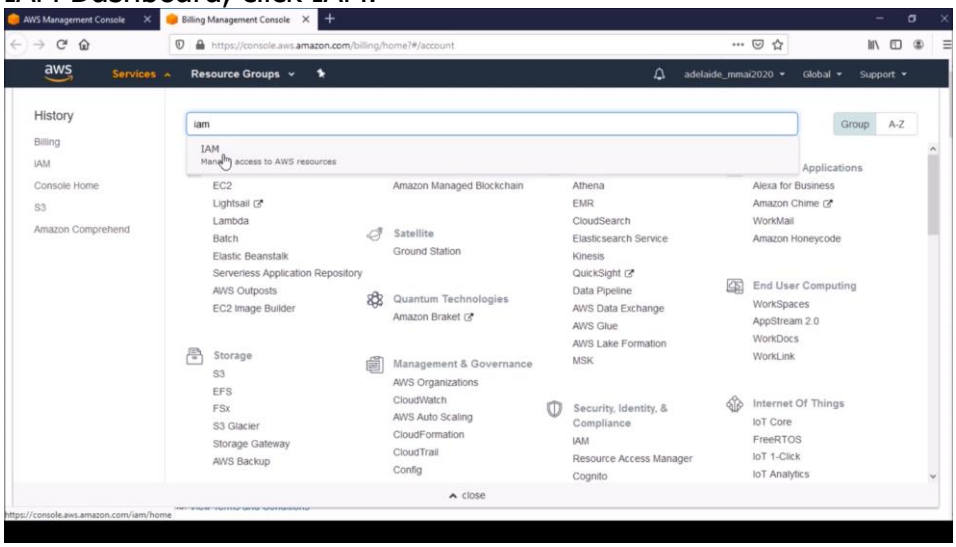
b. On the navigation bar, click on your account name, and then click My Account.



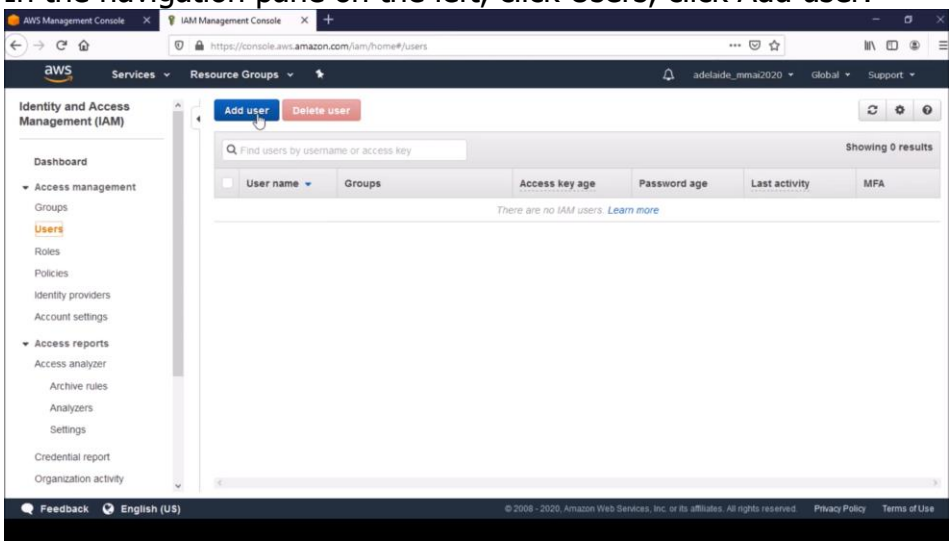
c. Scroll down to IAM User and Role Access to Billing Information, click Activate IAM Access, click Update.



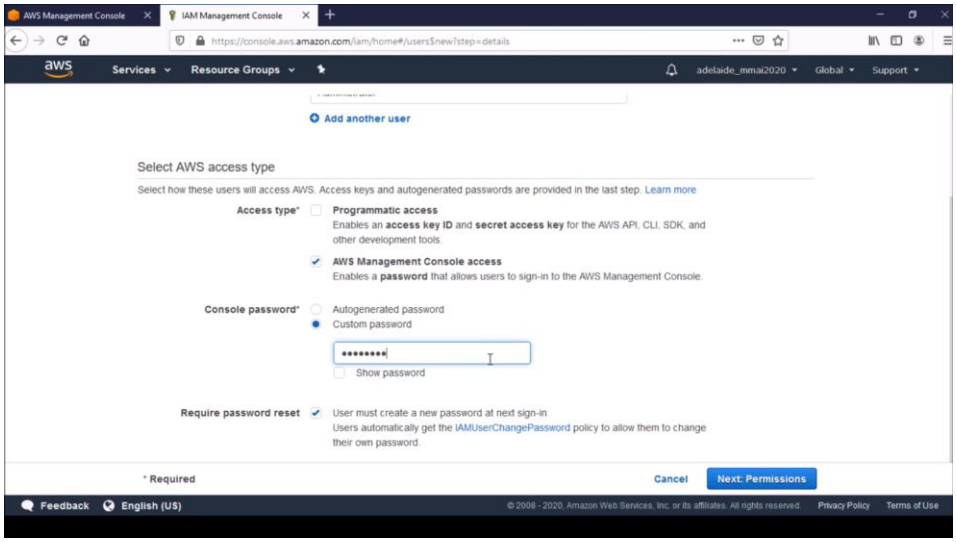
d. Scroll back up to the navigation bar, click Services, type IAM to return to the IAM Dashboard, click IAM.



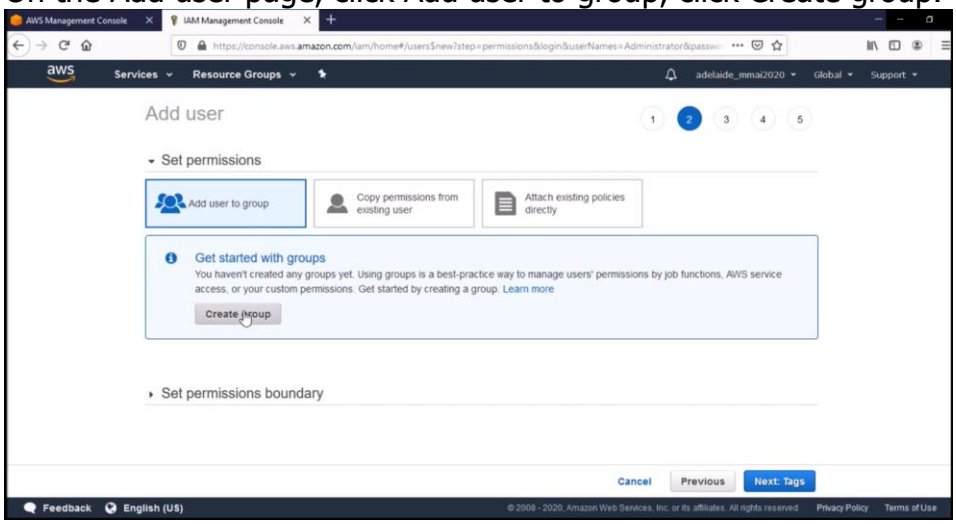
e. In the navigation pane on the left, click Users, click Add user.



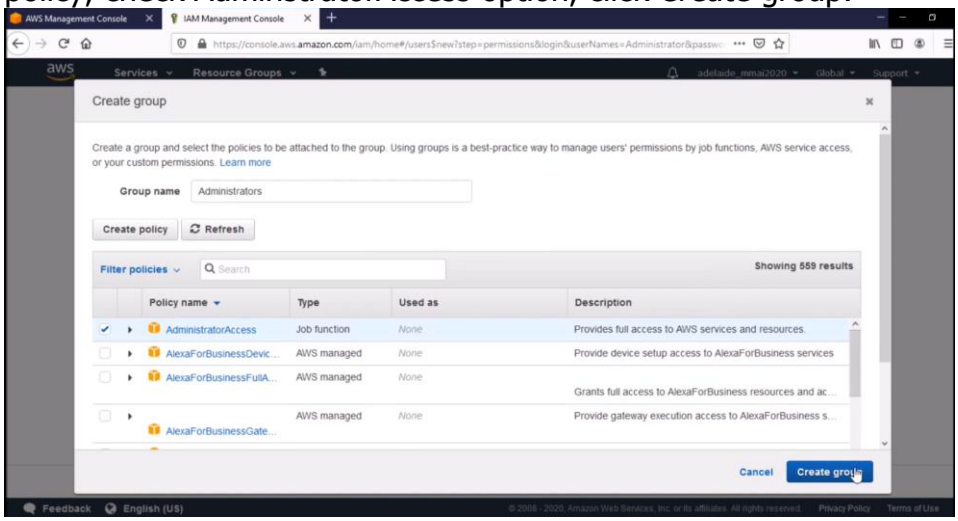
f. On the details page, type "Administrator" for the User name, check AWS Management Console access, select Custom password and supply a password, uncheck the User must create a new password at next sign-in option, click Next: Permissions.



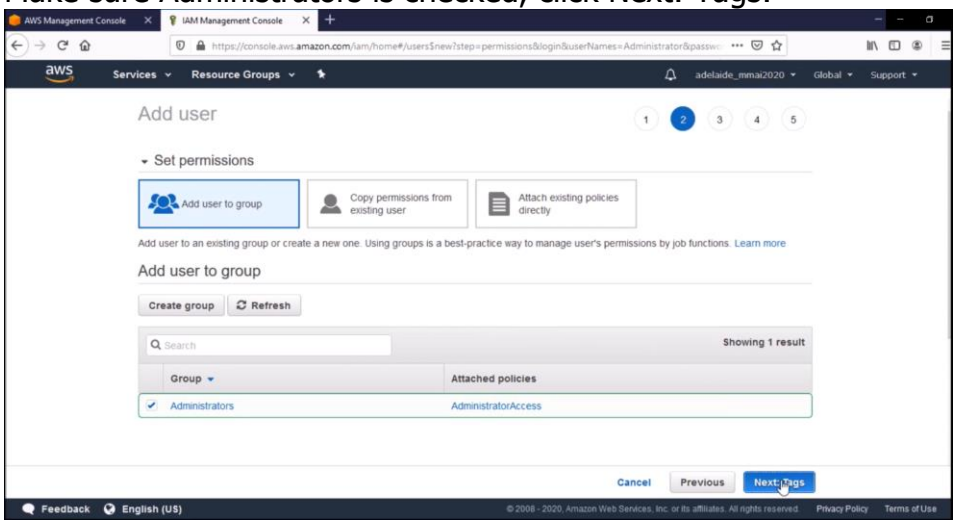
g. On the Add user page, click Add user to group, click Create group.



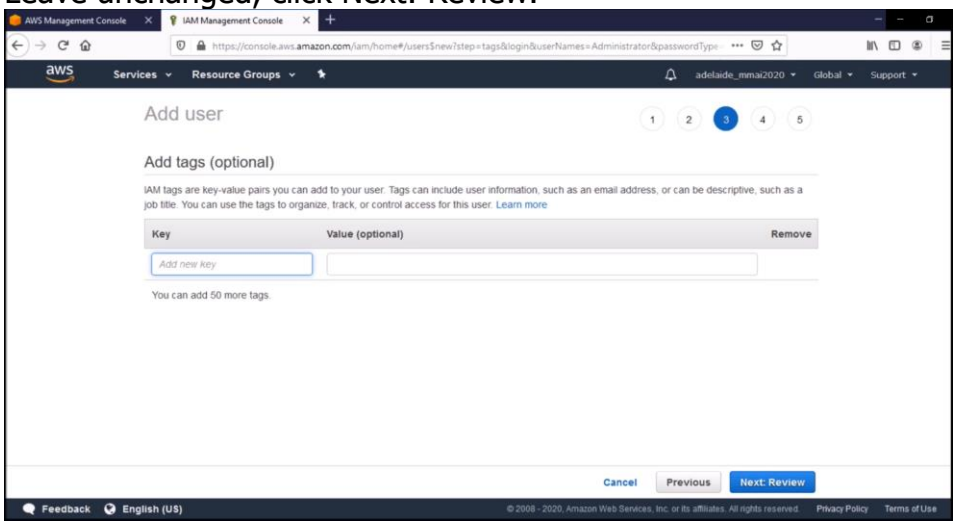
h. On the Create group dialog box, type Administrators for Group name, under policy, check AdminstratorAccess option, click Create group.



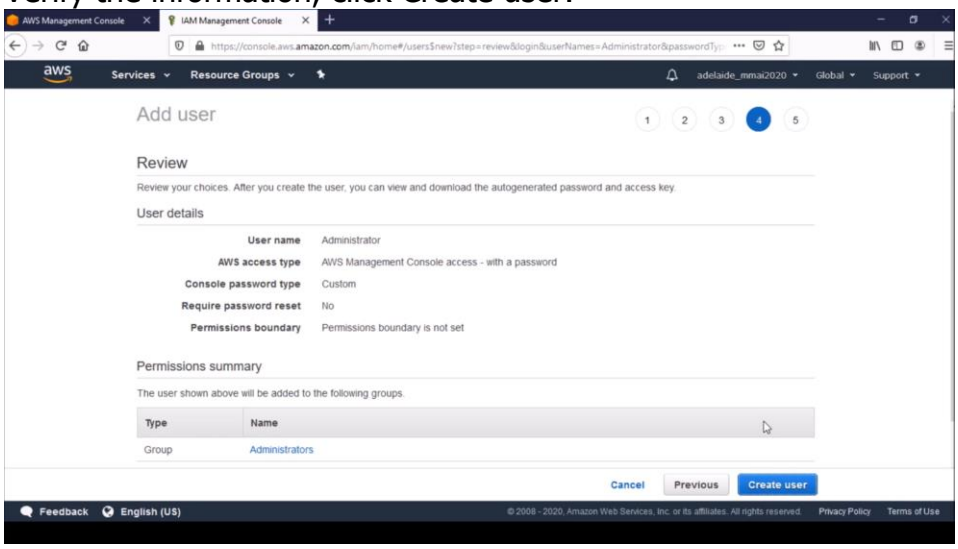
i. Make sure Administrators is checked, click Next: Tags.



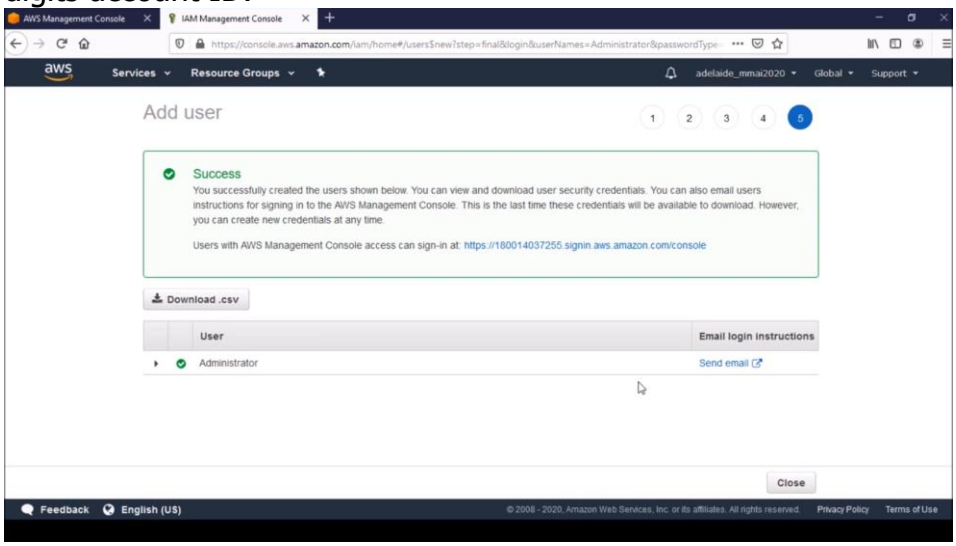
j. Leave unchanged, click Next: Review.



k. Verify the information, click Create user.

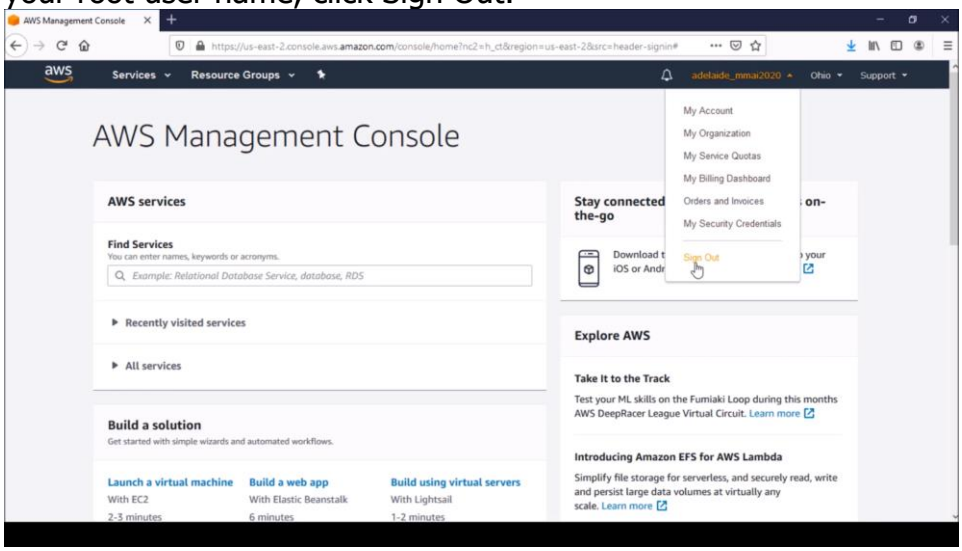


l. User will be created, click Close. Download the CSV file that contains your 12 digits account ID.

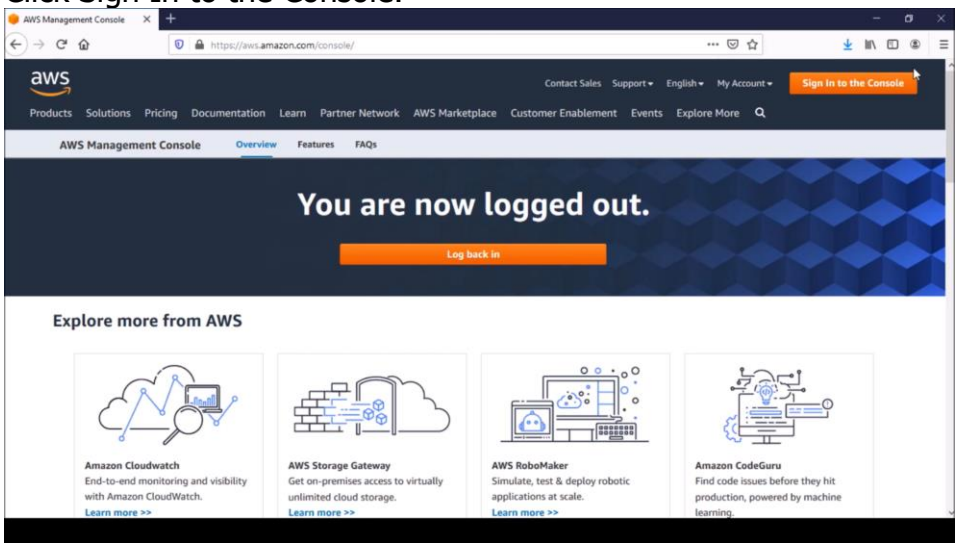


2. Sign in to the AWS Management Console using a special URL.

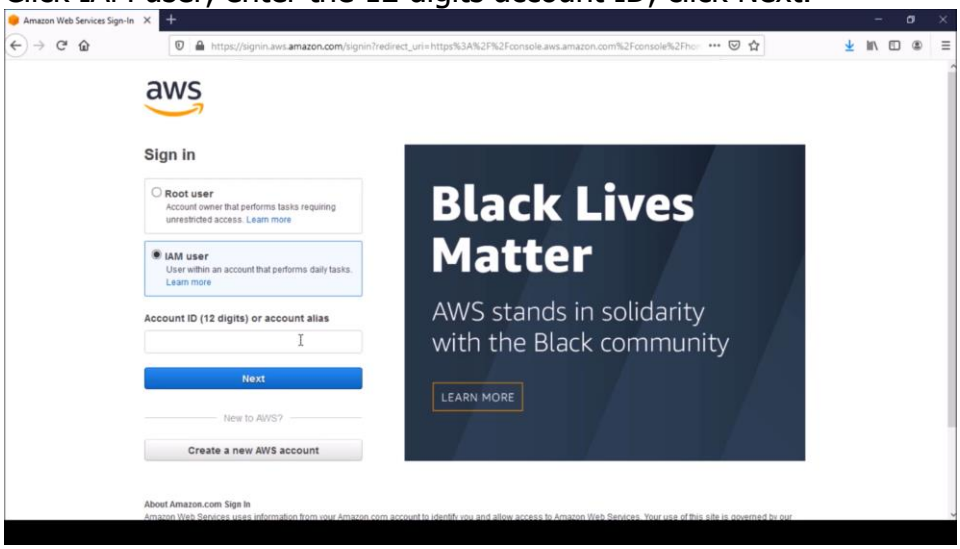
a. As the root user, sign out from the console. On the navigation bar, click on your root user name, click Sign Out.



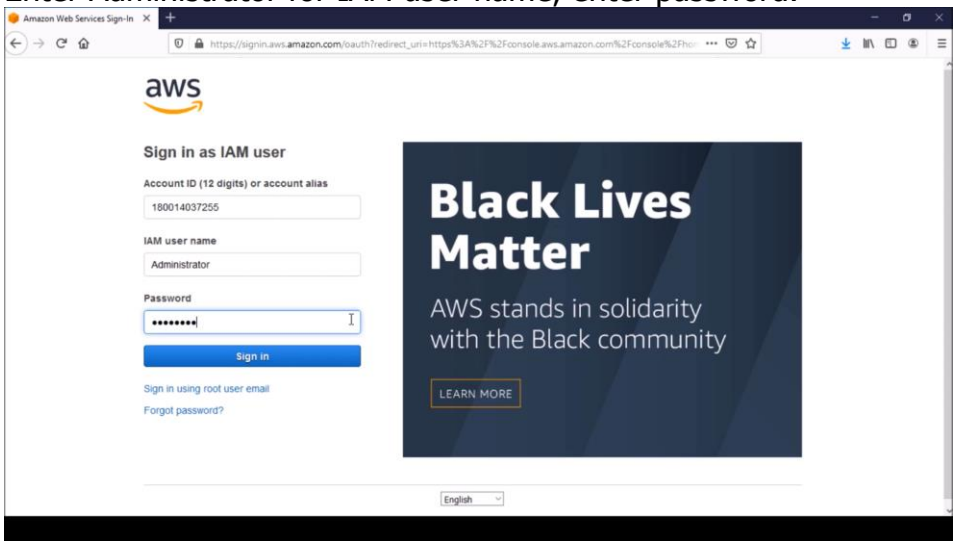
b. Click Sign In to the Console.



c. Click IAM user, enter the 12 digits account ID, click Next.

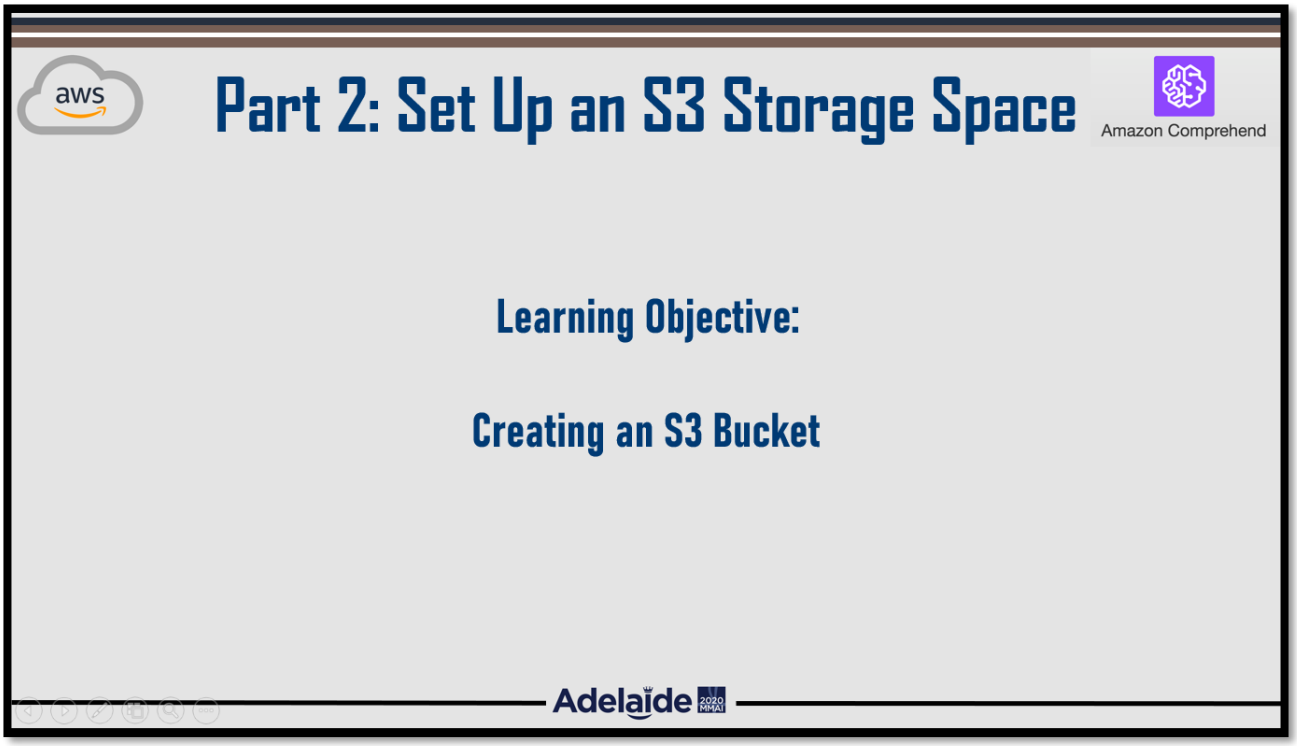


d. Enter Administrator for IAM user name, enter password.



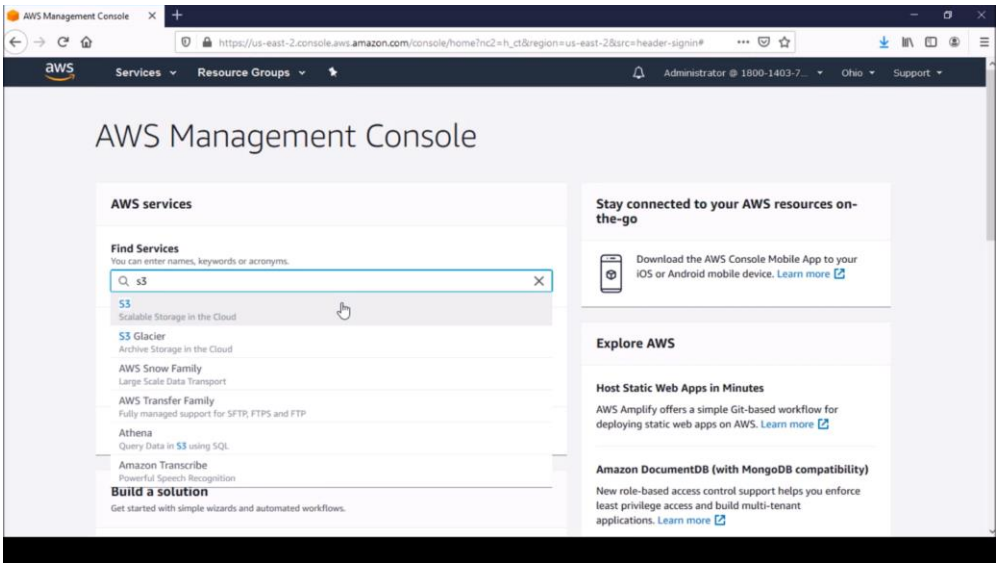
3. Now it's your turn! Let's apply what we've just learned about AWS Comprehend.

Part 2: Set Up an S3 Storage Space
Creating an S3 Bucket

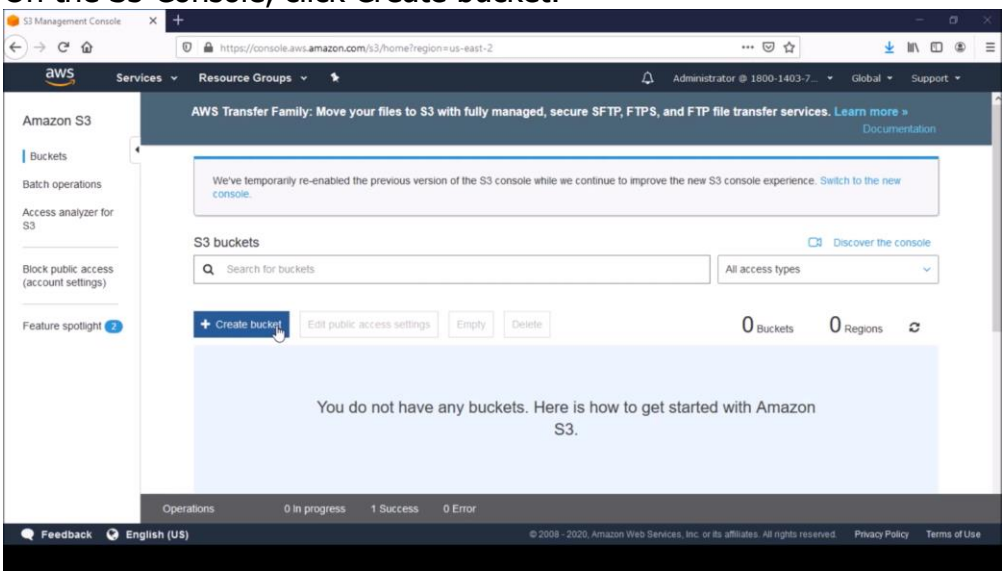


Create an S3 Bucket

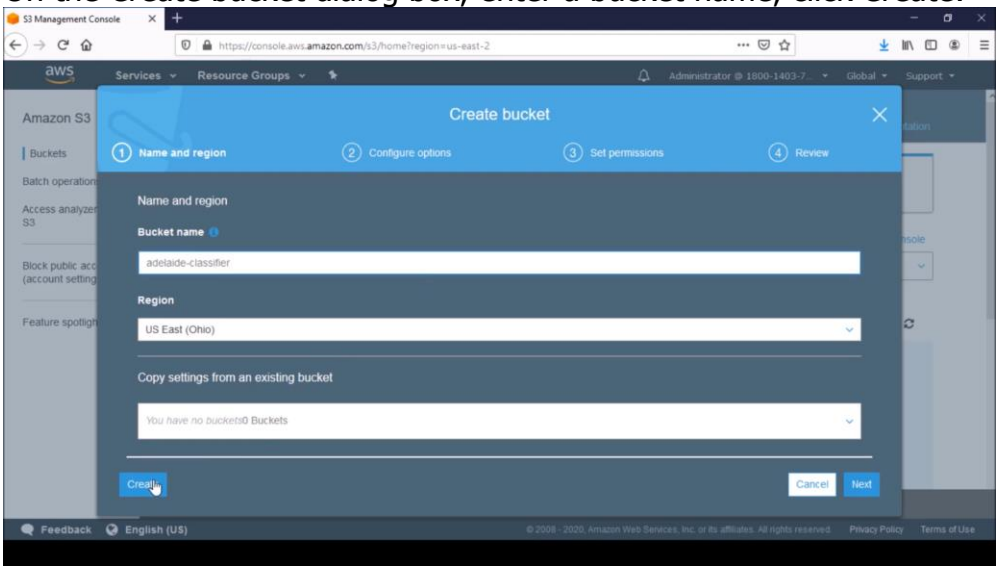
1. On the AWS Management Console, search S3, click S3 Scalable Storage in the Cloud.



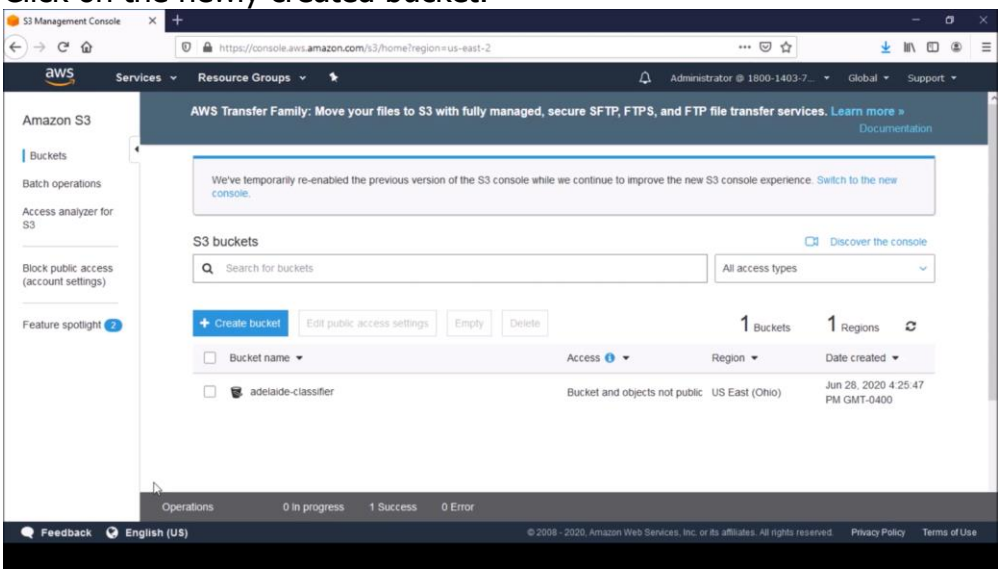
2. On the S3 Console, click Create bucket.



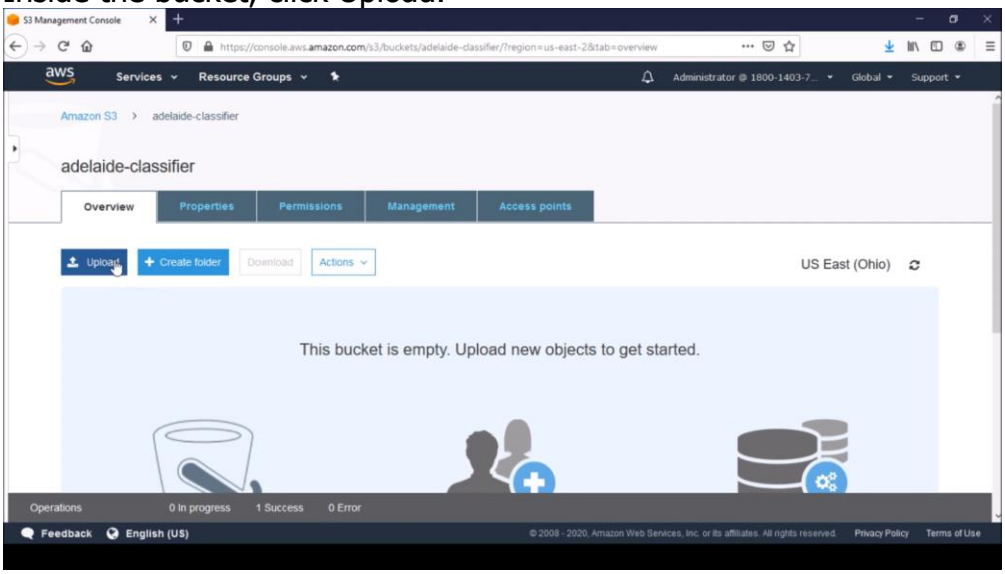
3. On the Create bucket dialog box, enter a bucket name, click Create.



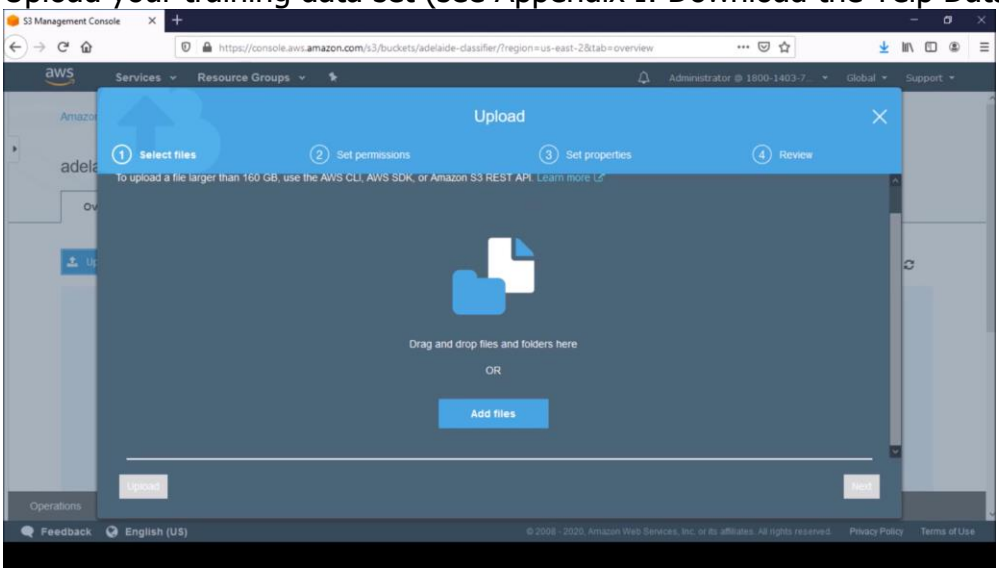
4. Click on the newly created bucket.



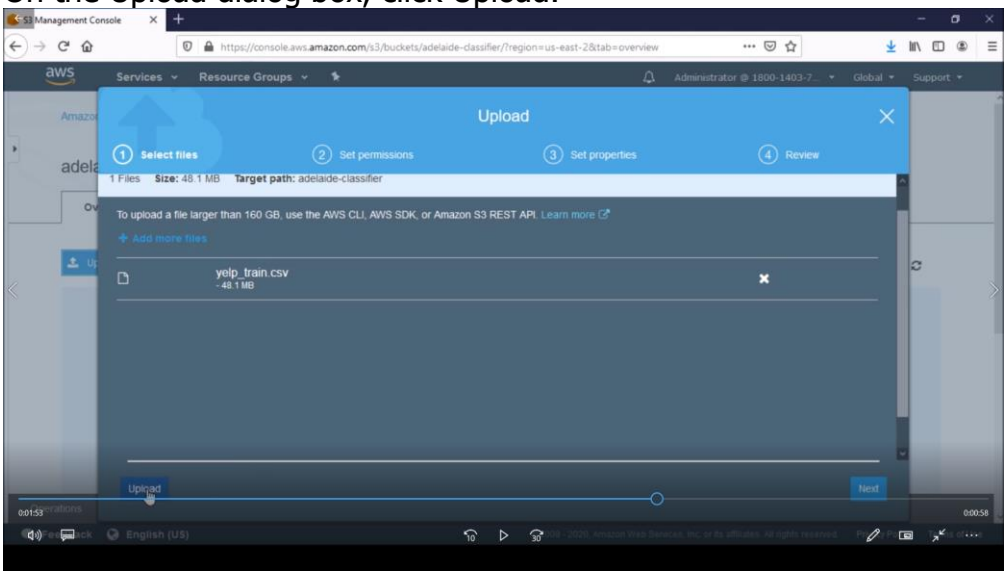
5. Inside the bucket, click Upload.



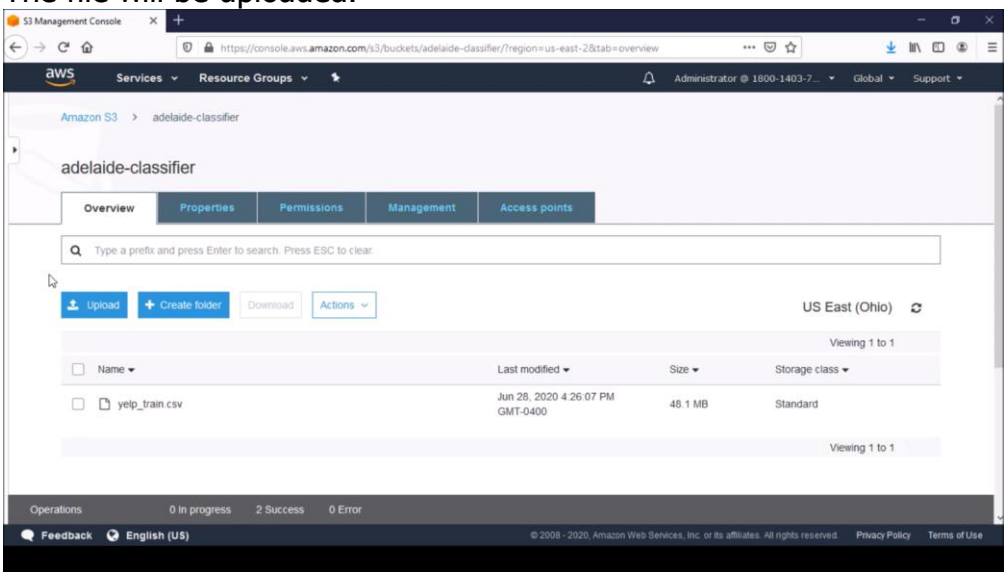
6. Upload your training data set (see Appendix I: Download the Yelp Dataset).



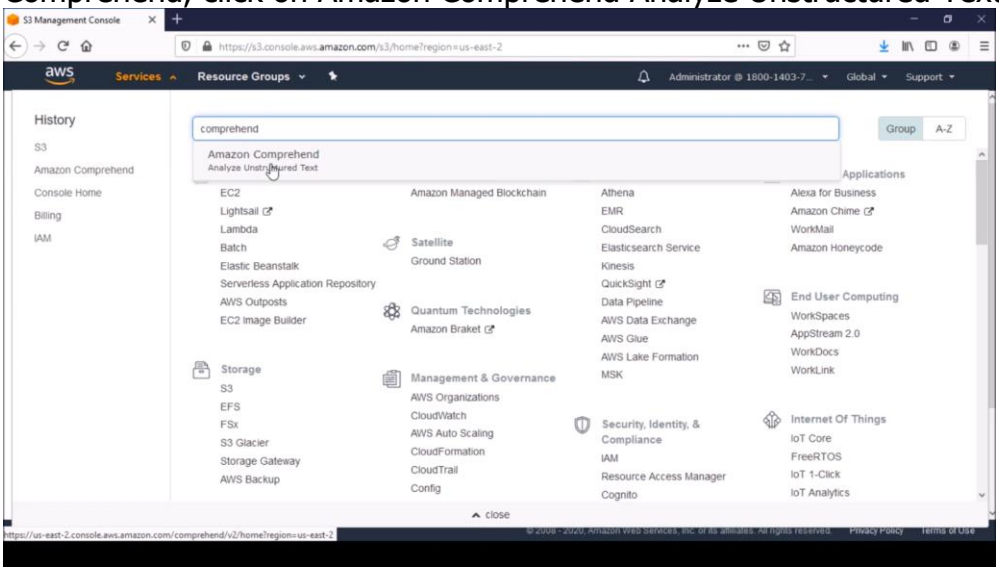
7. On the Upload dialog box, click Upload.



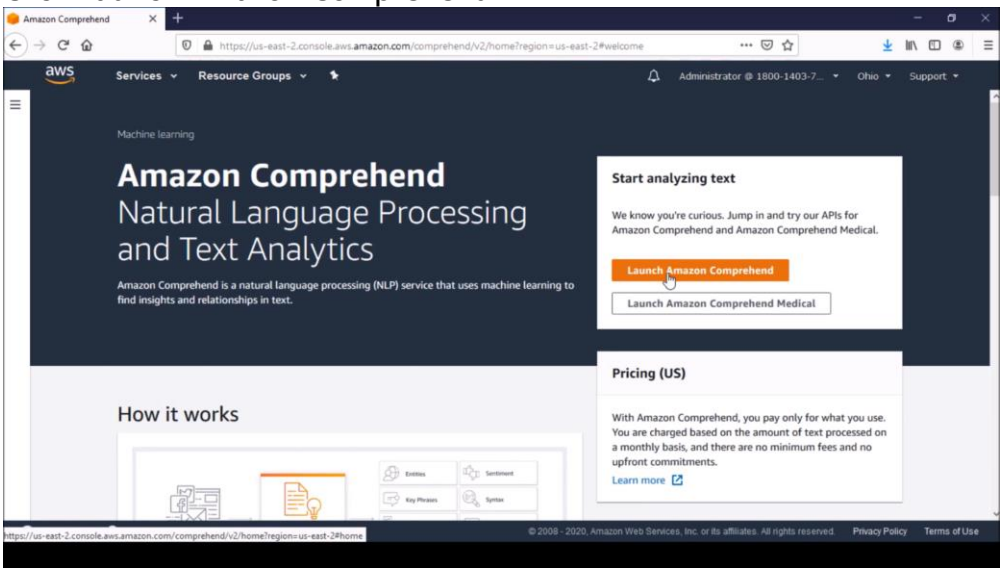
8. The file will be uploaded.



9. Go back to the Comprehend console. On the navigation bar, click Services, type Comprehend, click on Amazon Comprehend Analyze Unstructured Text.

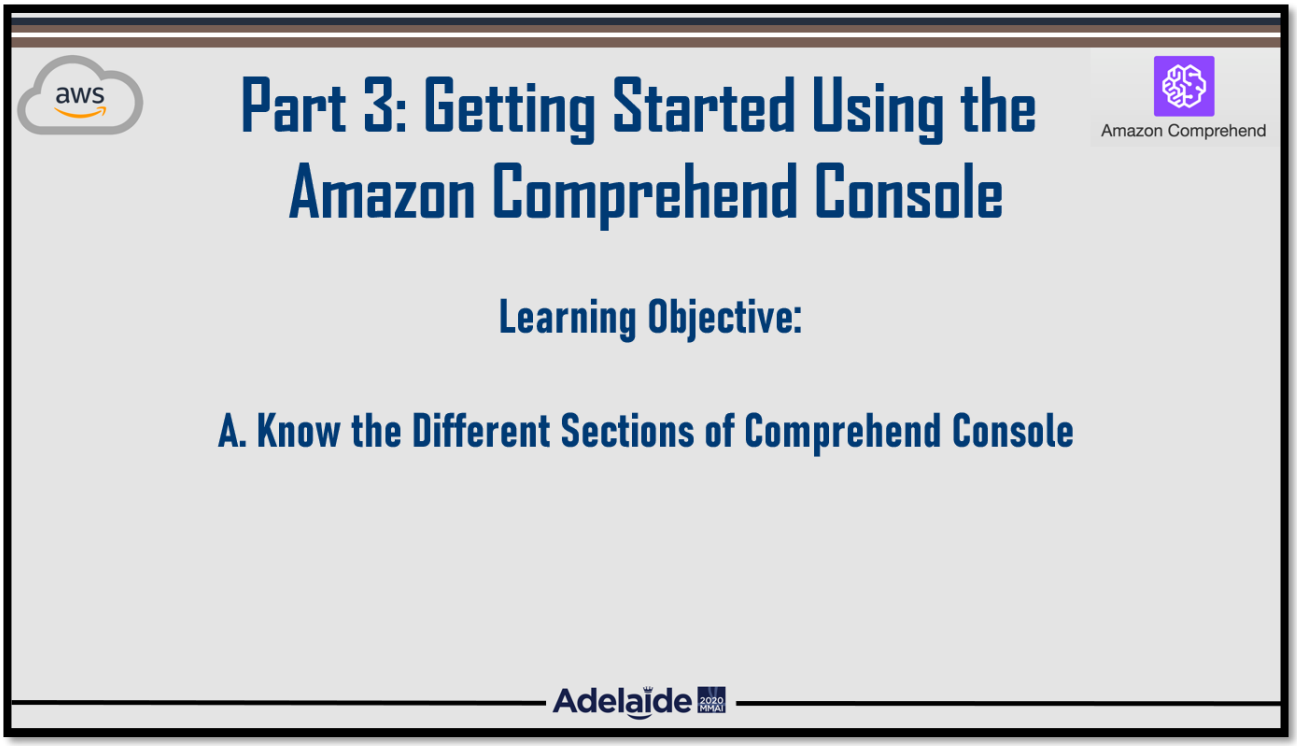


10. Click Launch Amazon Comprehend.



11. Now it's your turn! Let's apply what we've just learned about AWS Comprehend.

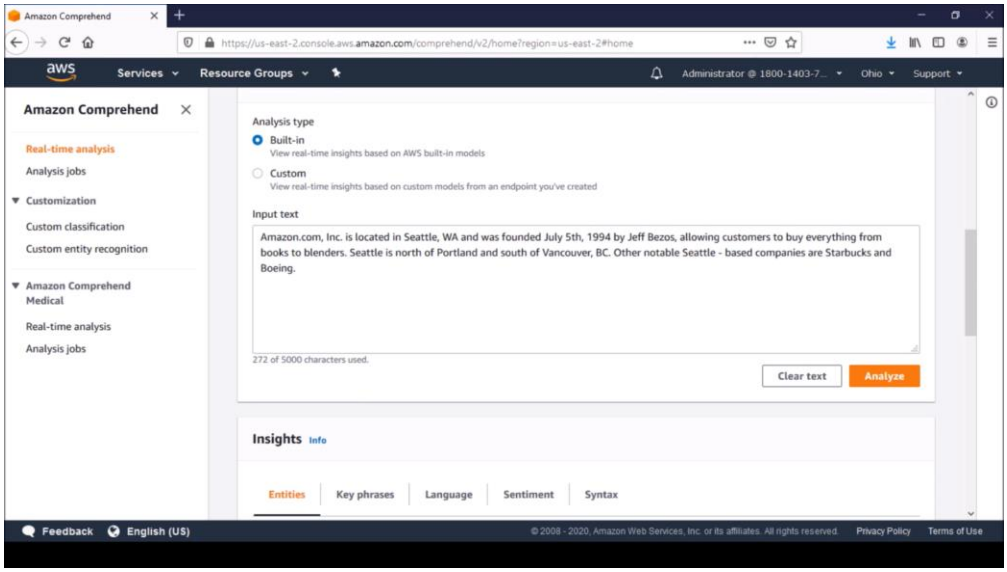
Part 3: Getting Started Using the AWS Comprehend Console



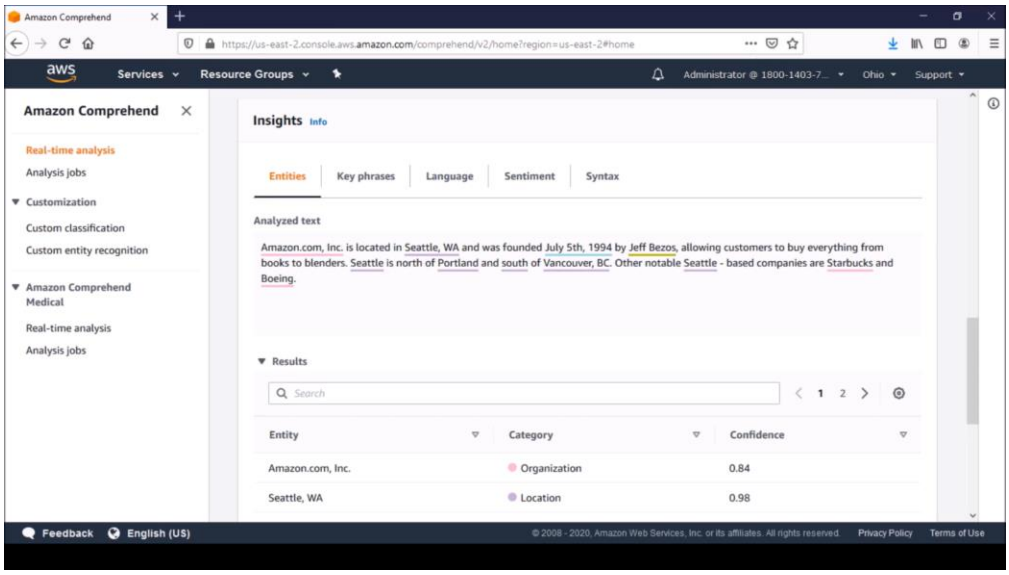
Know the Different Sections of AWS Comprehend Console

The Amazon Comprehend console enables you to analyze the contents of documents up to 1,000 characters long. The results are shown in the console so that you can review the analysis.

You can replace the sample text with your own text either in English or one of the other languages supported by Amazon Comprehend and then choose Analyze to get an analysis of your text.

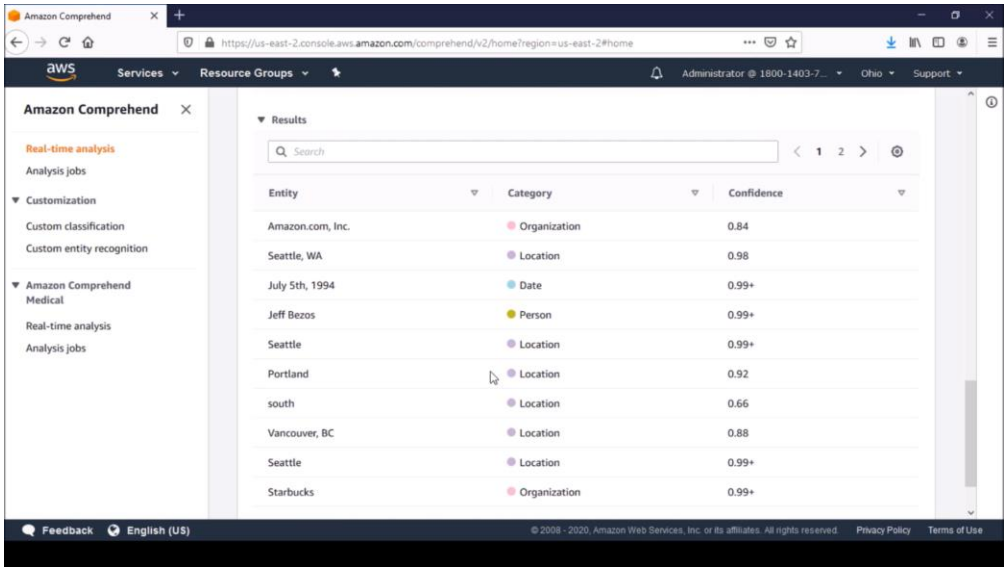


Entity Types Tab

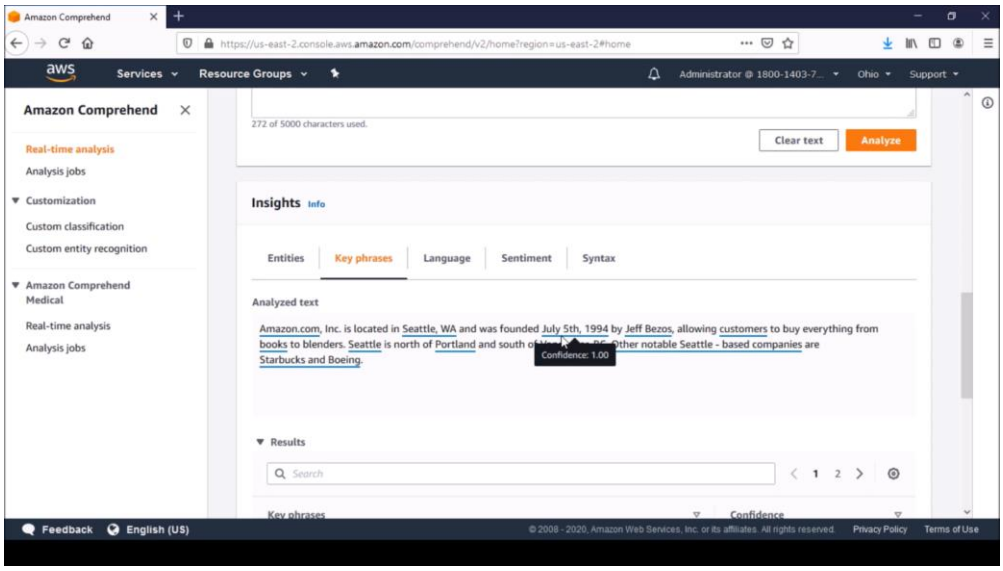


The text is color-coded to indicate different entity types such as organizations, locations, dates, and persons.

Below the text being analyzed, the Results pane shows more information about the text. Each entry shows the entity, its category, and the level of confidence AWS Comprehend has in this analysis.

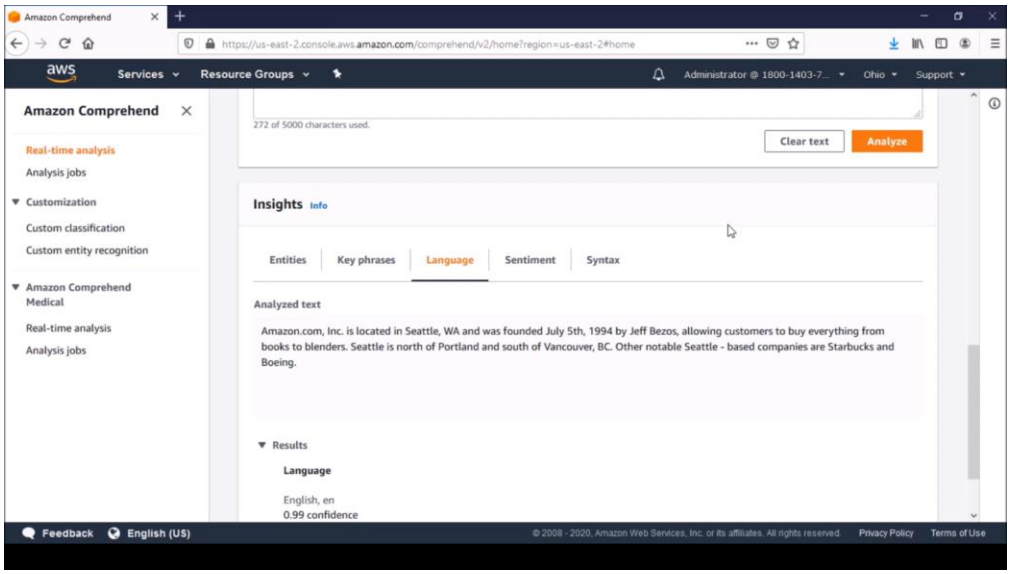


Key Phrases Tab



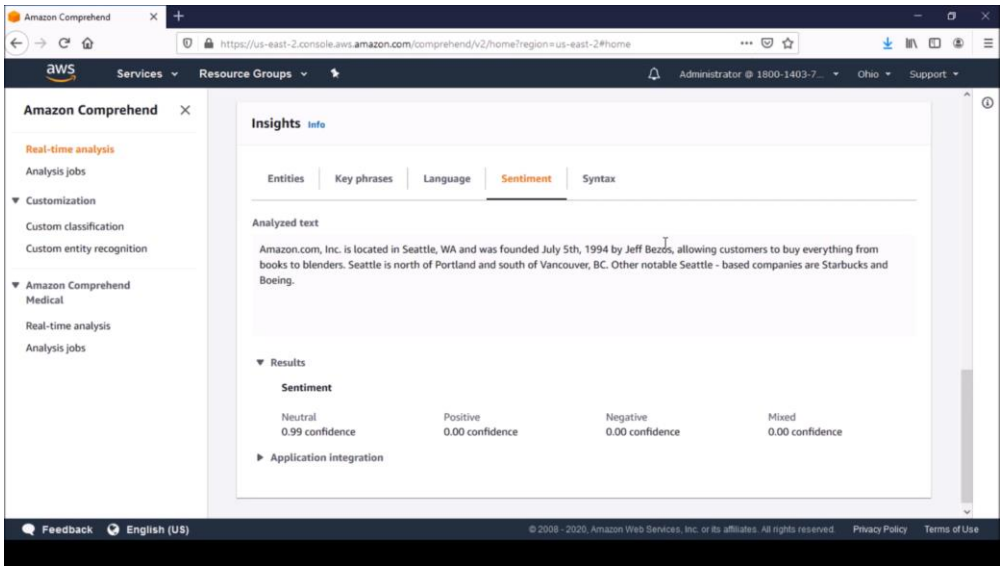
The Key phrases tab lists key noun phrases that Amazon Comprehend detected in the input text and the associated confidence level.

Language Tab



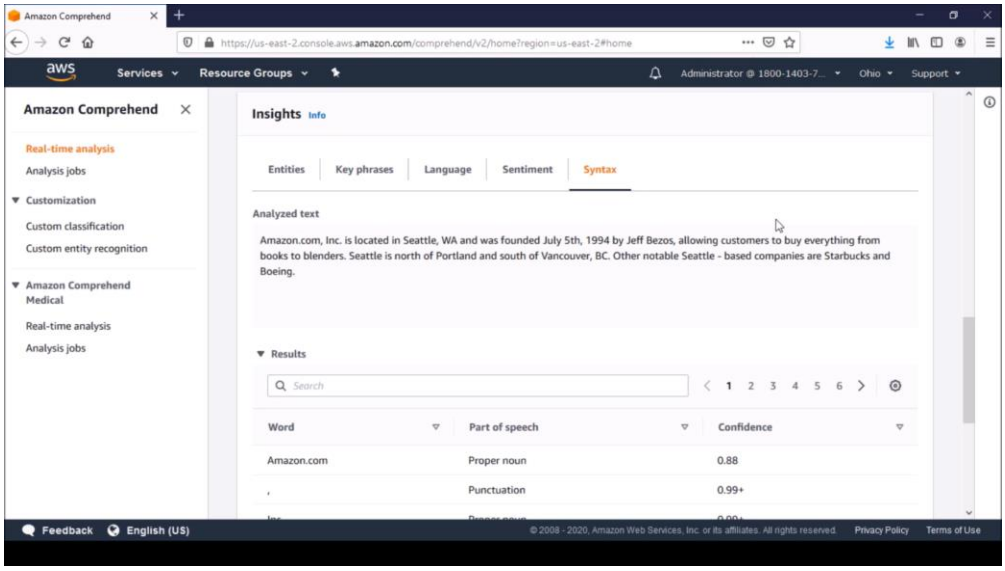
The Language tab shows the dominant language of the text and AWS Comprehend's level of confidence that it's detected the dominant language correctly. Amazon Comprehend can recognize 100 languages.

Sentiment Tab

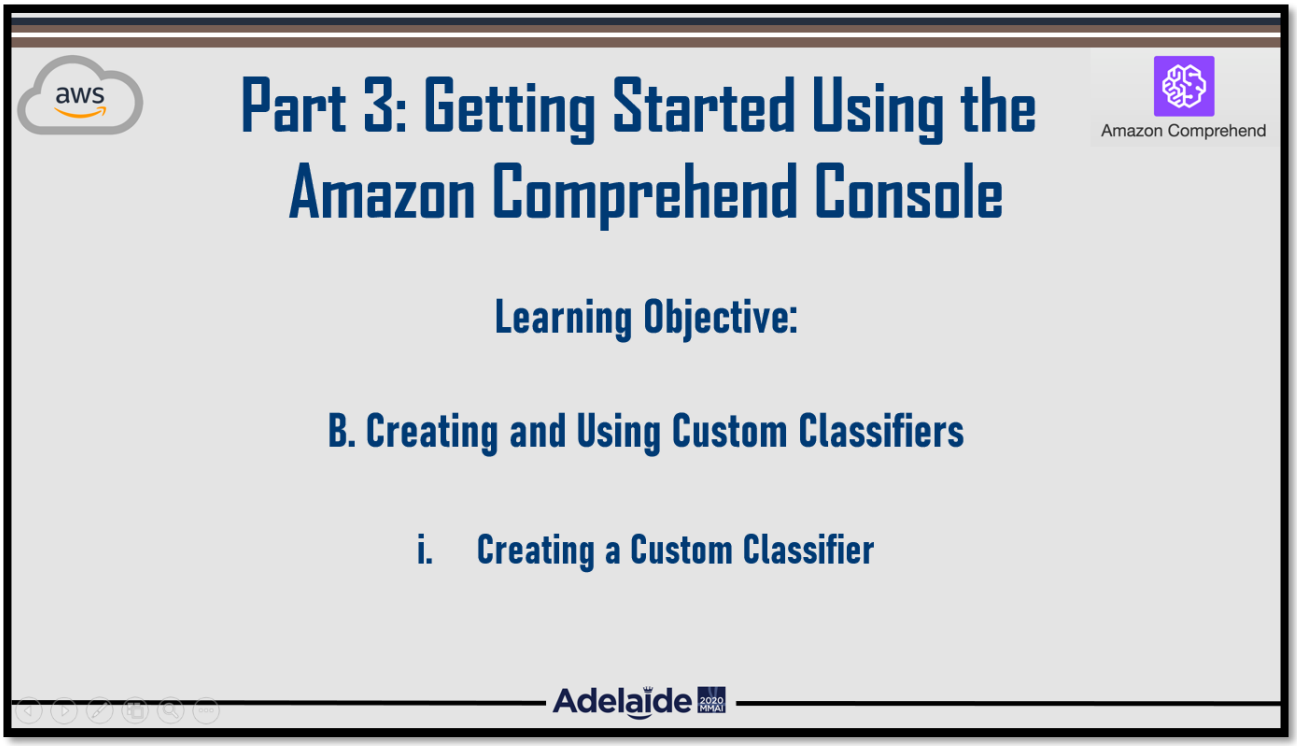


The Sentiment tab shows the overall emotional sentiment of the text. Sentiment can be rated neutral, positive, negative, or mixed. In this case, each emotional sentiment has a confidence rating, providing an estimate by AWS Comprehend for that sentiment being dominant.

Syntax Tab



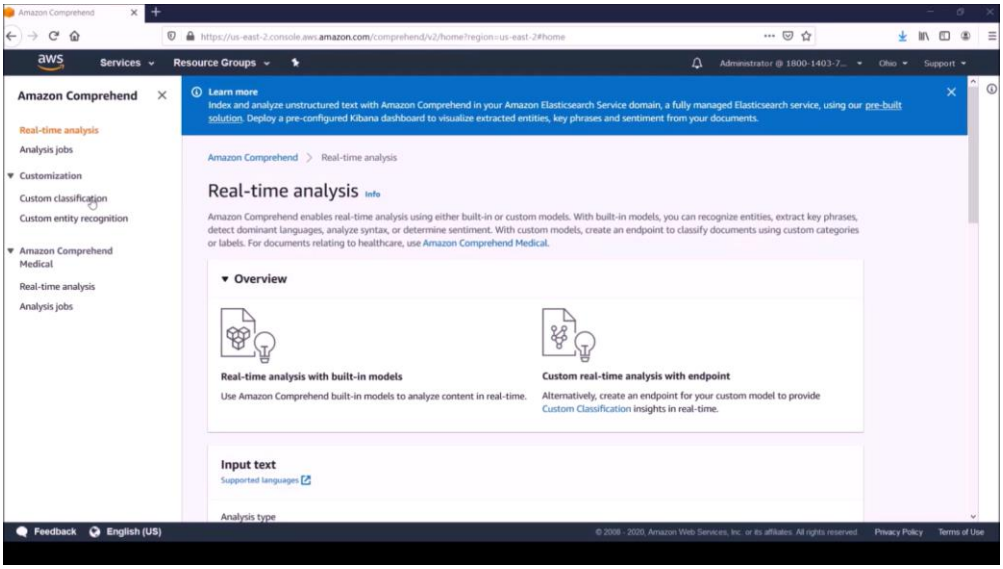
The Syntax tab shows a breakdown of each element in the text, along with its part of speech and the associated confidence score.



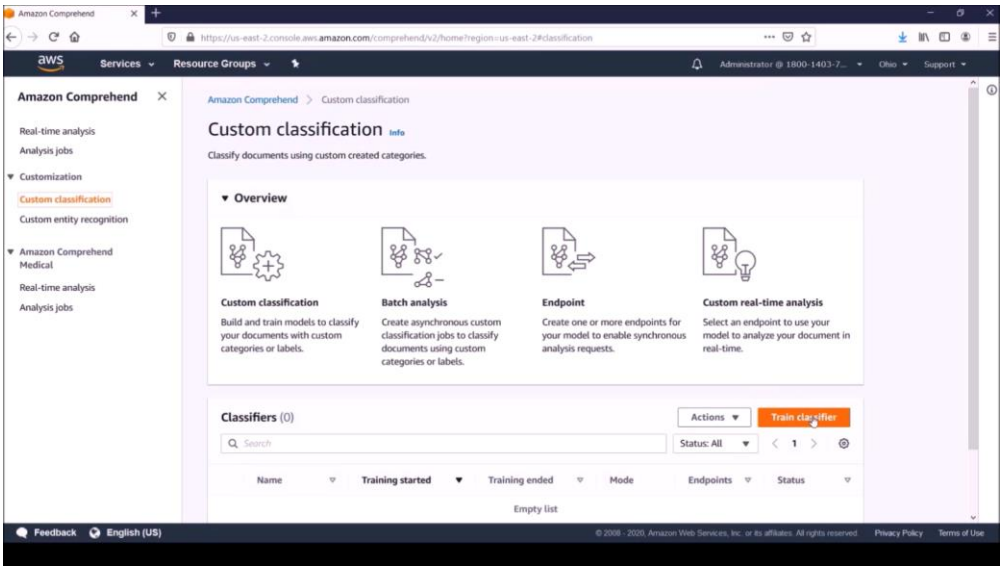
You can create and train custom classifiers using the console, and then run asynchronous classification jobs to analyze your documents. You can also use the same custom model and add an endpoint to it to run custom classification requests to gain real-time (synchronous) insights about your text.

Create a Custom Classifier

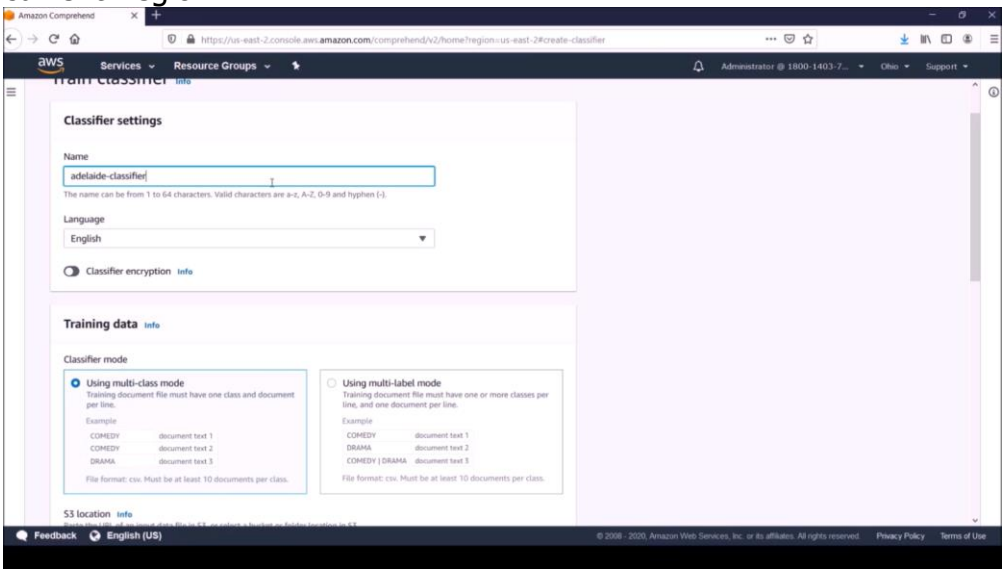
1. On the AWS Comprehend Console, from the left menu, choose Customization and then choose Custom Classification.



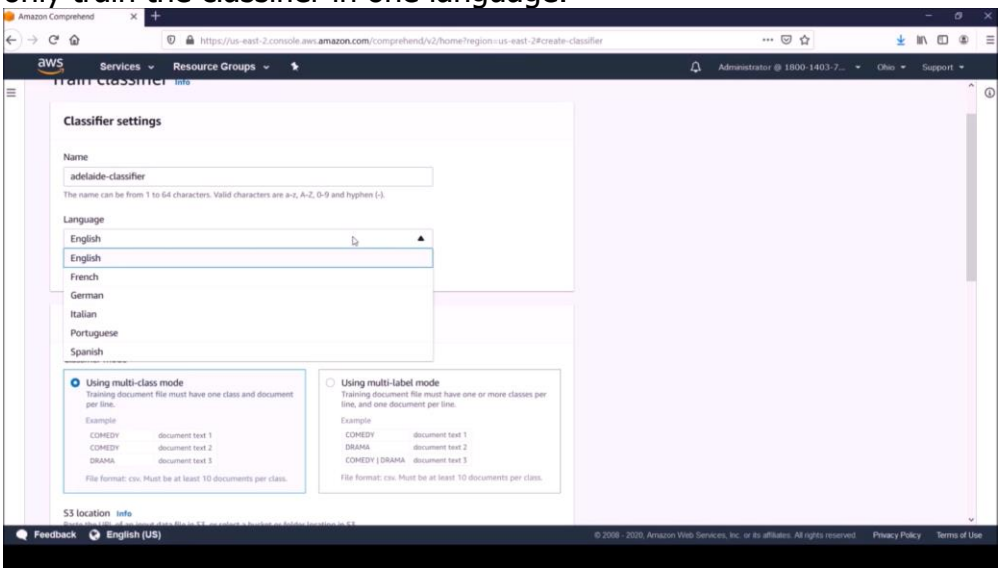
2. Click Train Classifier.



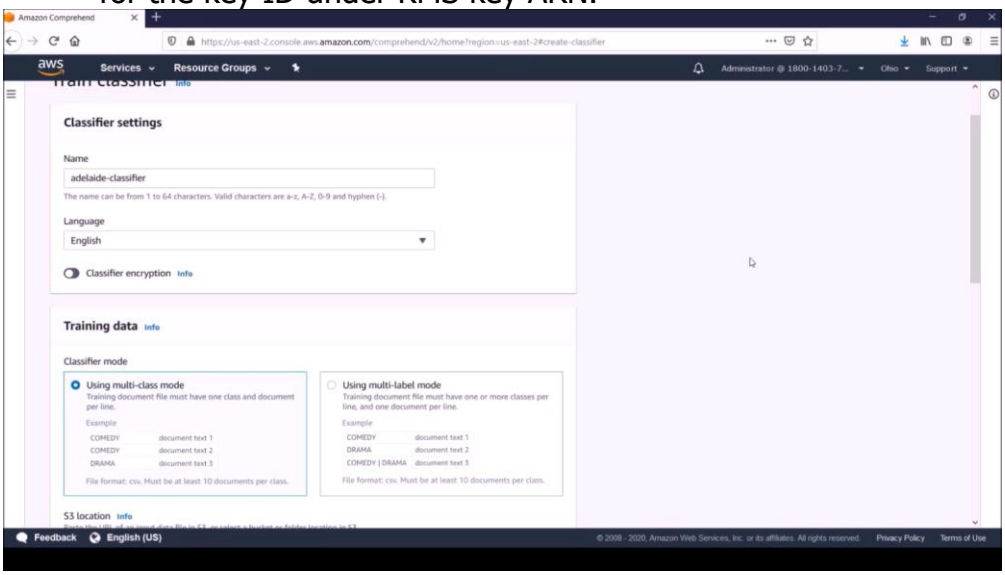
3. Give the classifier a name. The name must be unique within your account and current Region.



4. Select the language of the training documents. You can train a document classifier using any of the languages that work with AWS Comprehend. However, you can only train the classifier in one language.

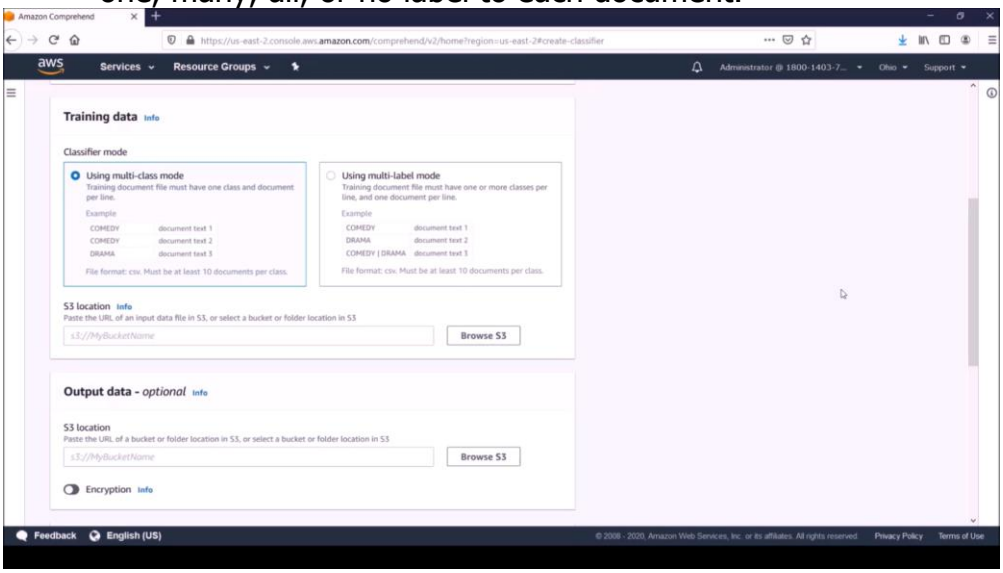


5. If you want to encrypt the data in the storage volume while your training job is being processed, choose Classifier encryption and then choose whether to use a KMS key associated with your current account, or one from another account.
    a. If you are using a key associated with the current account, choose the key ID for KMS key ID.
    b. If you are using a key associated with a different account, enter the ARN for the key ID under KMS key ARN.

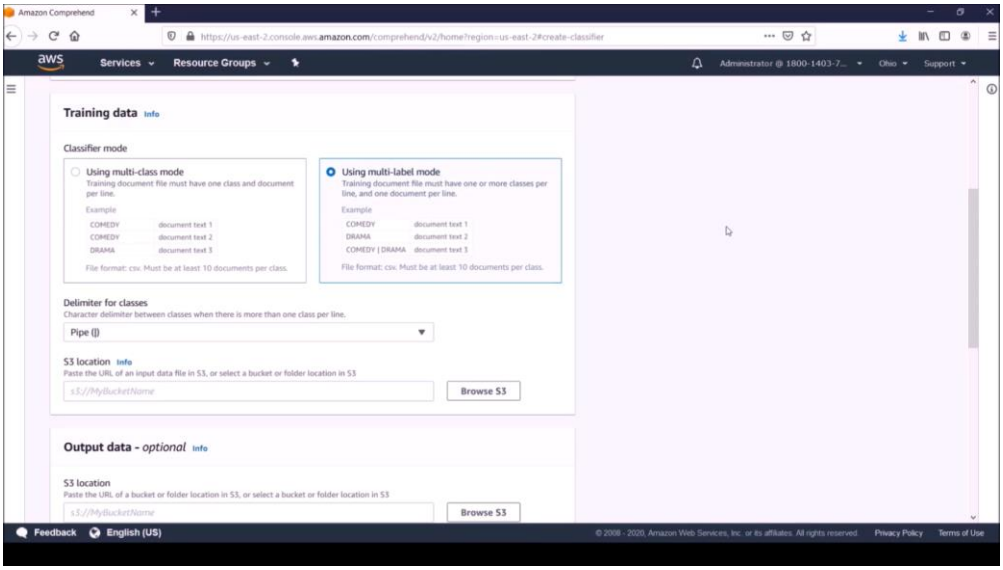


6. Under Training data, choose which classifier mode to use.
    a. Multi-class mode: Choose this option if the categories you are assigning to documents are mutually exclusive and you are training your classifier to assign one and only one label to each document.

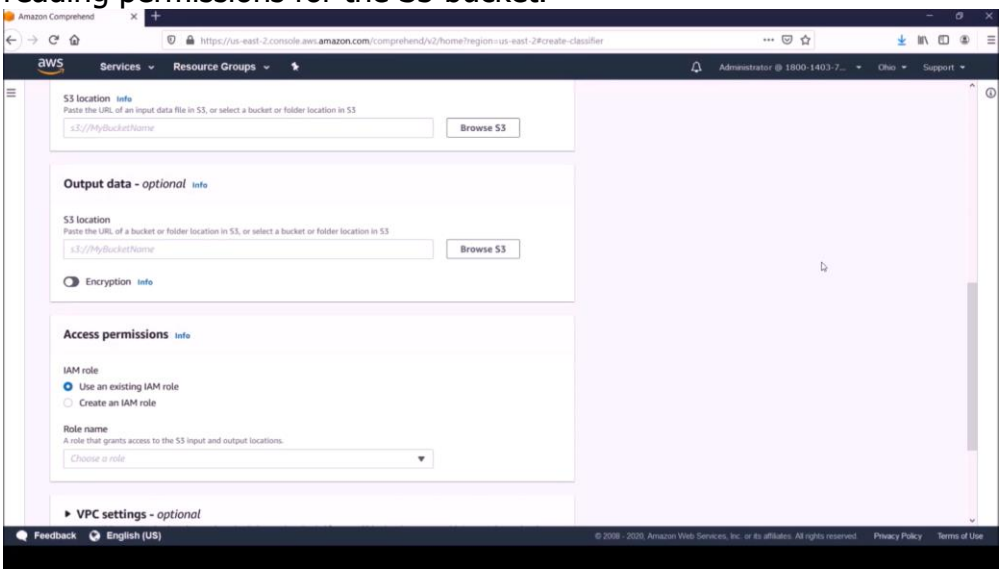
b. Multi-label mode: Choose this option if multiple categories can be applied to a document at the same time and you are training your classifier to assign one, many, all, or no label to each document.



c. If you chose Multi-label mode, choose the character delimiter you want to use to separate labels when there is more than one label per line from Delimiter for labels.



7. Enter the location of the Amazon S3 bucket that contains your training documents or navigate to it by choosing Browse S3 then navigate to your training file. The IAM role you're using for access permissions for the training job must have reading permissions for the S3 bucket.

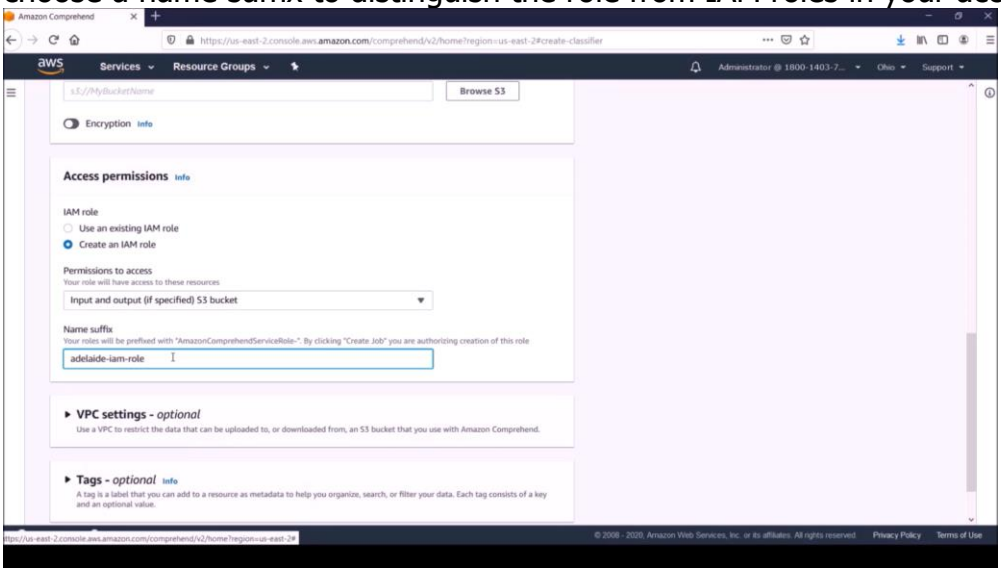


8. If you want Amazon Comprehend to create a confusion matrix that provides metrics on how well the classifier performed during training, enter the location of an Amazon S3 bucket where it will be saved (Output data field).

9. If you choose to encrypt the output result from your training job, choose Encryption and then choose whether to use a KMS key associated with the current account, or one from another account.
    a. If you are using a key associated with the current account, choose the key alias for KMS key ID.
    b. If you are using a key associated with a different account, enter the ARN for the key alias or ID under KMS key ID.

10. Choose Use an existing IAM role, and then choose an existing IAM role that has read permissions for the S3 bucket that contains your training documents. Only roles that have a trust policy that begins with comprehend.amazonaws.com are valid.

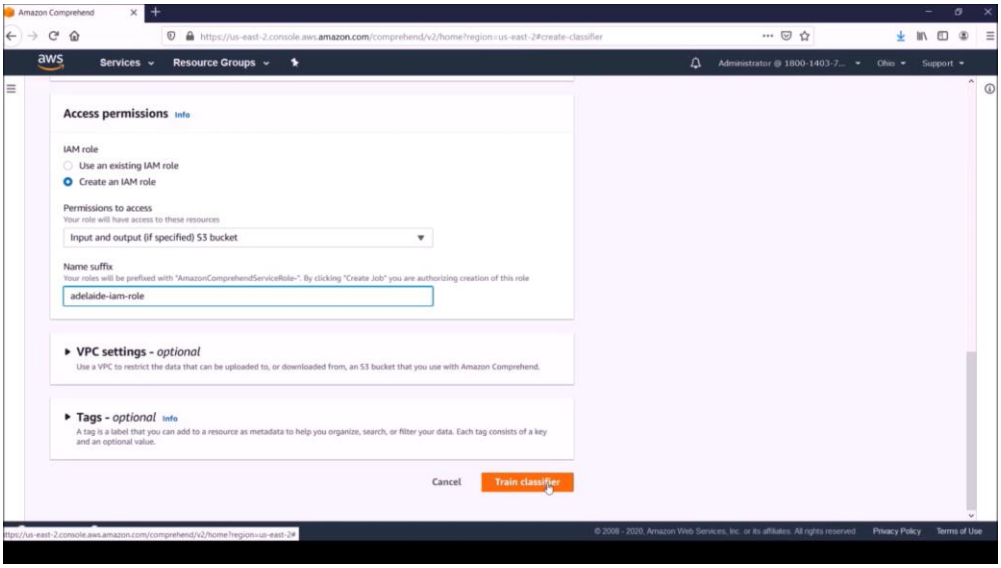
If you don't already have an IAM role with these permissions, choose Create an IAM role to make one. Choose the access permissions to grant this role, and then choose a name suffix to distinguish the role from IAM roles in your account.



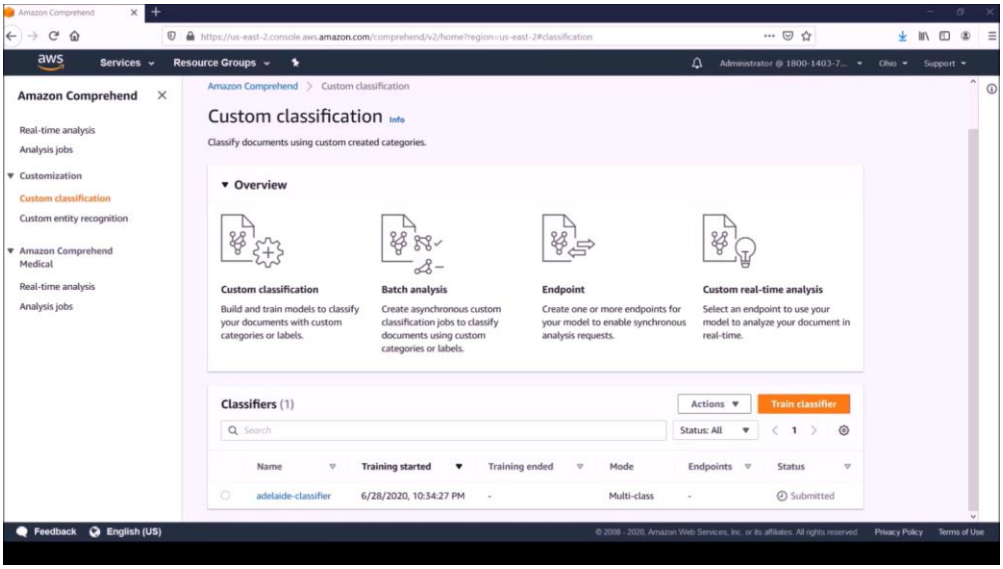
11. To launch your resources into AWS Comprehend from a VPC (Virtual Private Cloud), enter the VPC ID under VPC or choose the ID from the drop-down list.
    a. Choose the subnet under Subnets(s). After you select the first subnet, you can choose additional ones.
    b. Under Security Group(s), choose the security group to use if you specified one. After you select the first security group, you can choose additional ones.

    To add a tag to the custom classifier, enter a key-value pair under Tags. Choose Add tag.

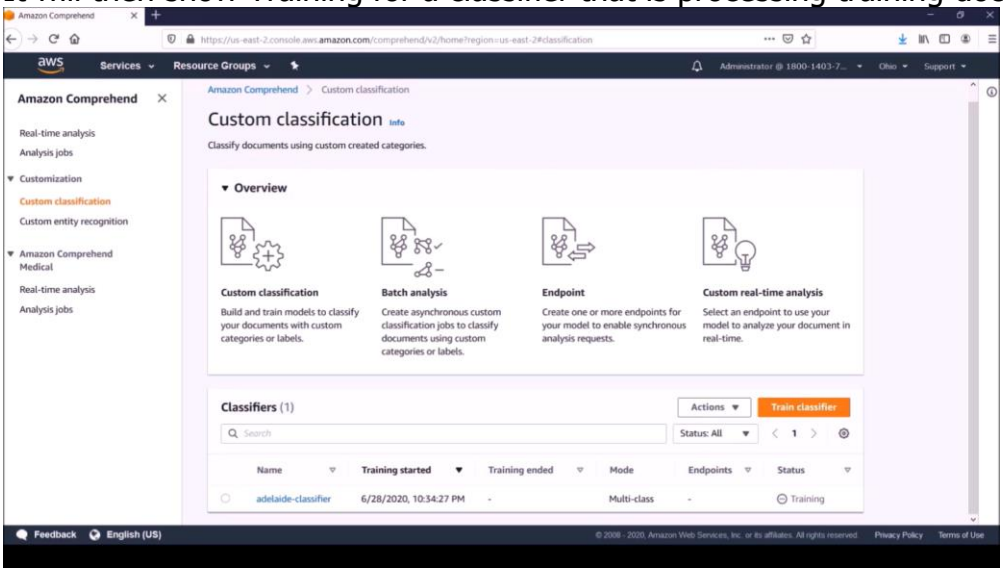
12. Click Train Classifier



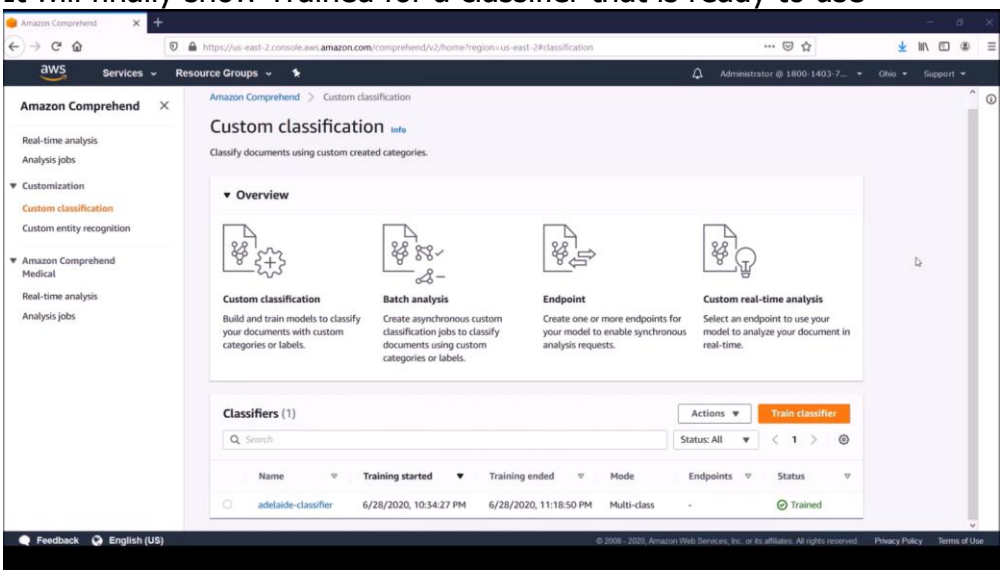
13. The new classifier will then appear in the list, showing its status. It will first show as Submitted.



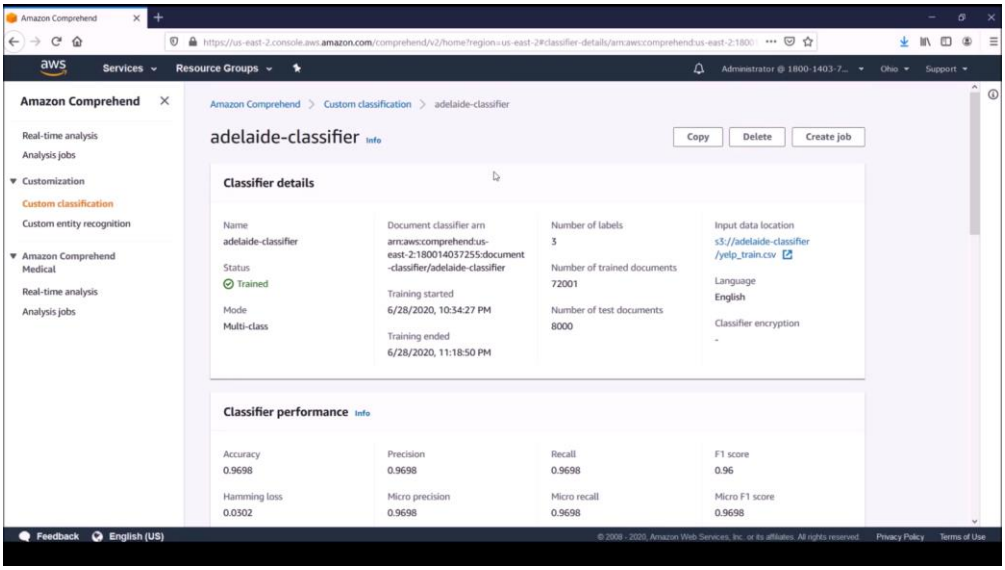
14. It will then show Training for a classifier that is processing training documents.



15. It will finally show Trained for a classifier that is ready to use

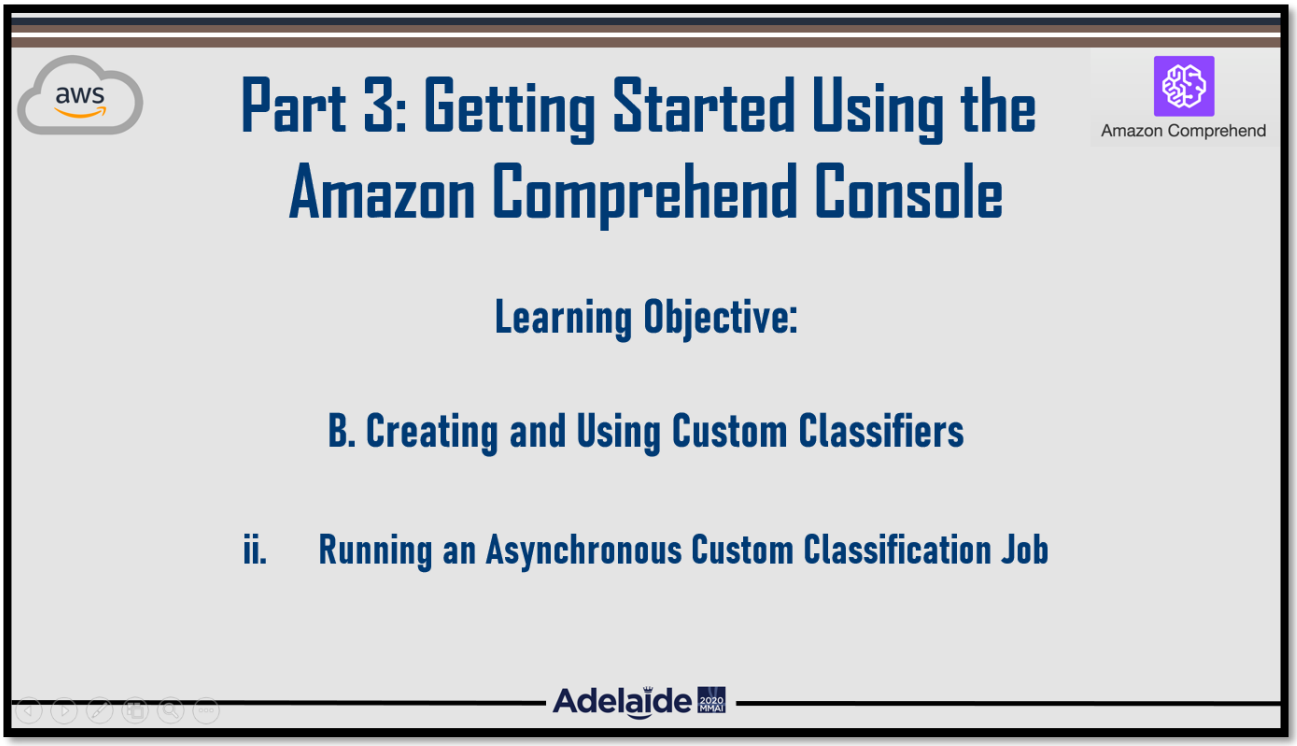


16. Click on the classifier name to see more information about it.



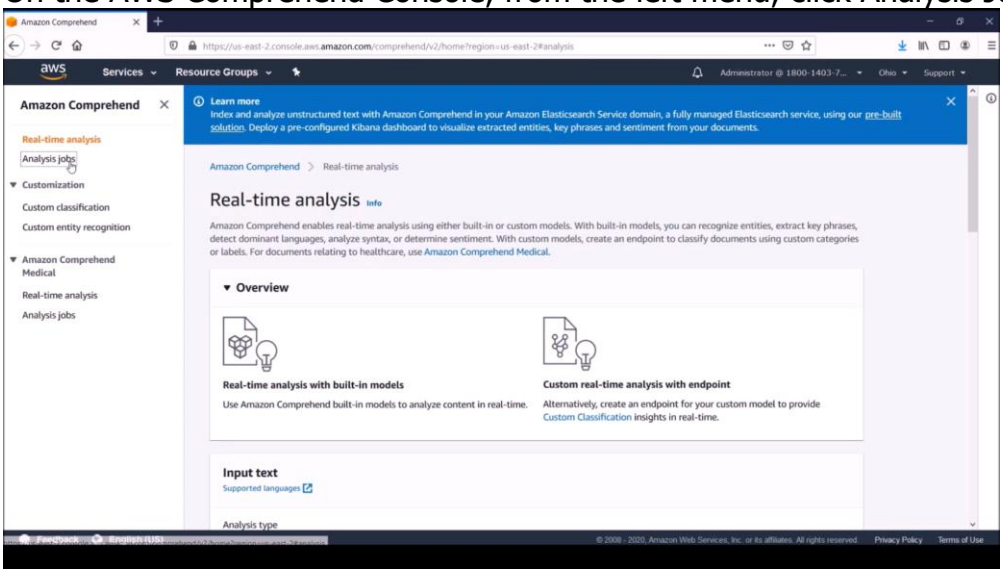
17. Click on Custom Classification to go back to the list.

18. Now it's your turn! Let's apply what we've just learned about AWS Comprehend.

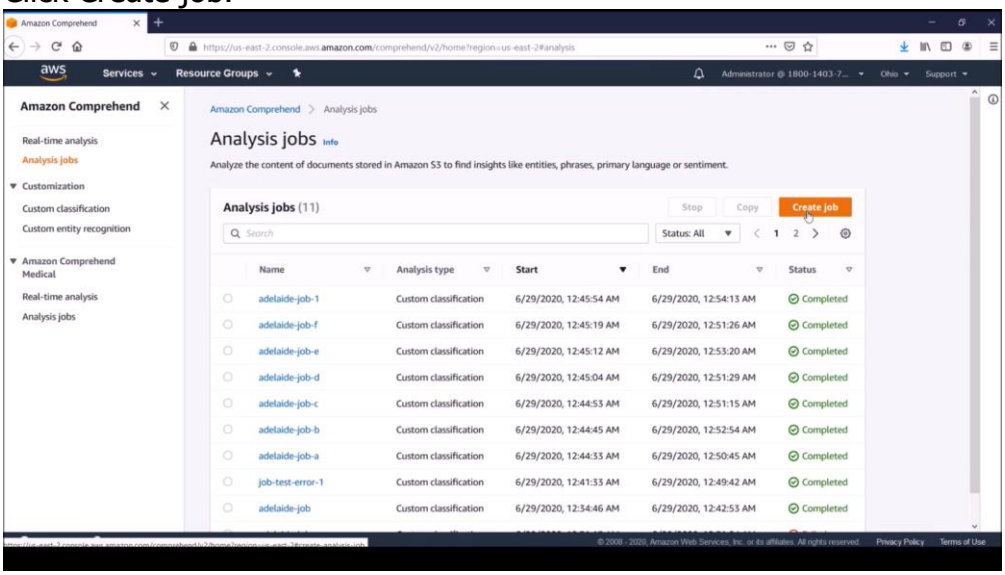


Create an Asynchronous Custom Classification Job

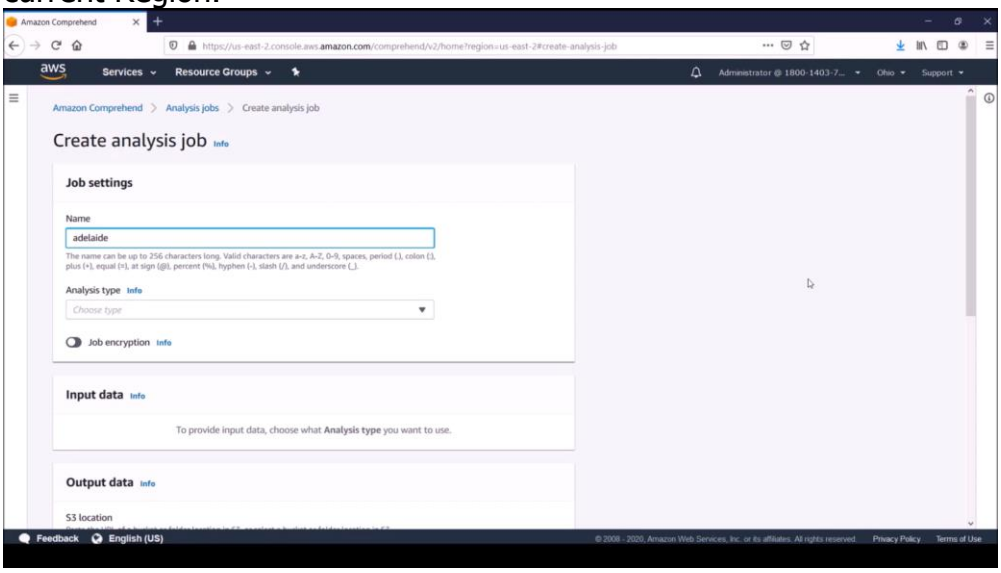
1. On the AWS Comprehend Console, from the left menu, click Analysis Jobs



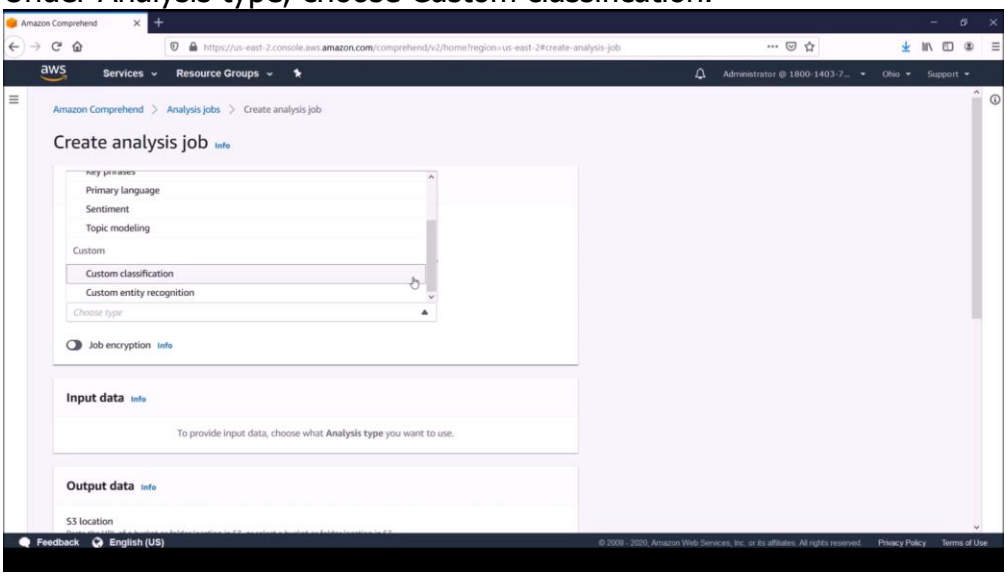
2. Click Create job.



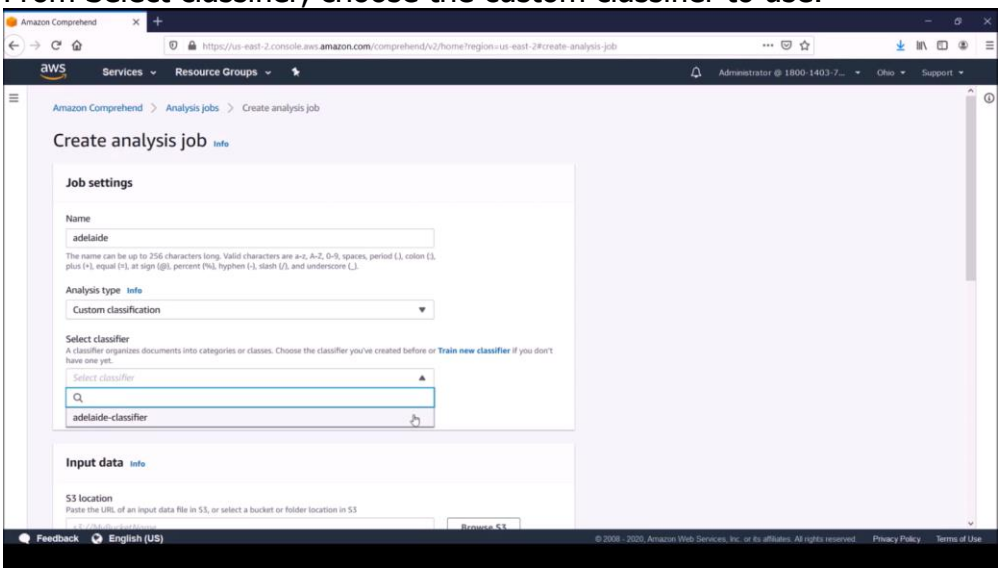
3. Give the classification job a name. The name must be unique your account and current Region.



4. Under Analysis type, choose Custom classification.



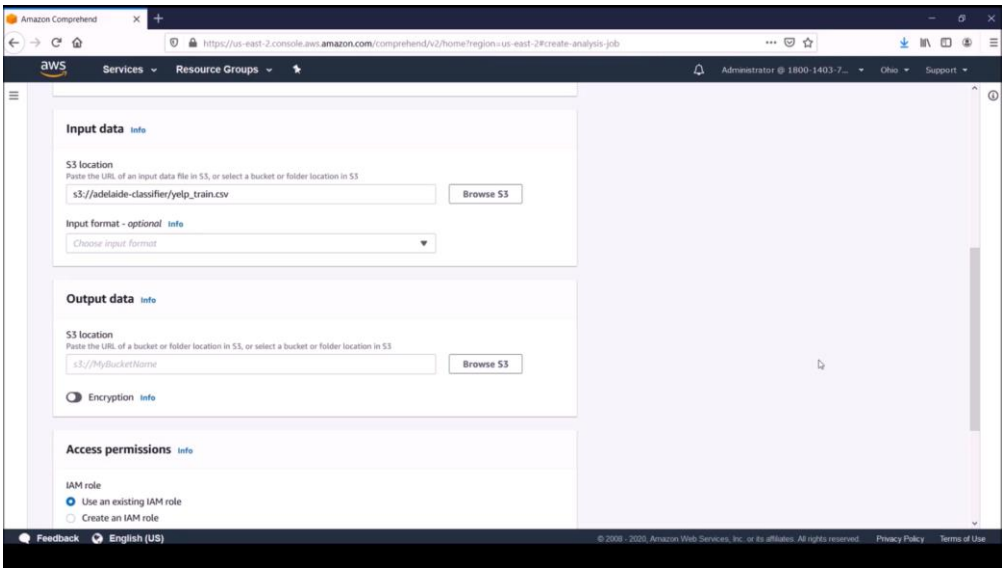
5. From Select classifier, choose the custom classifier to use.



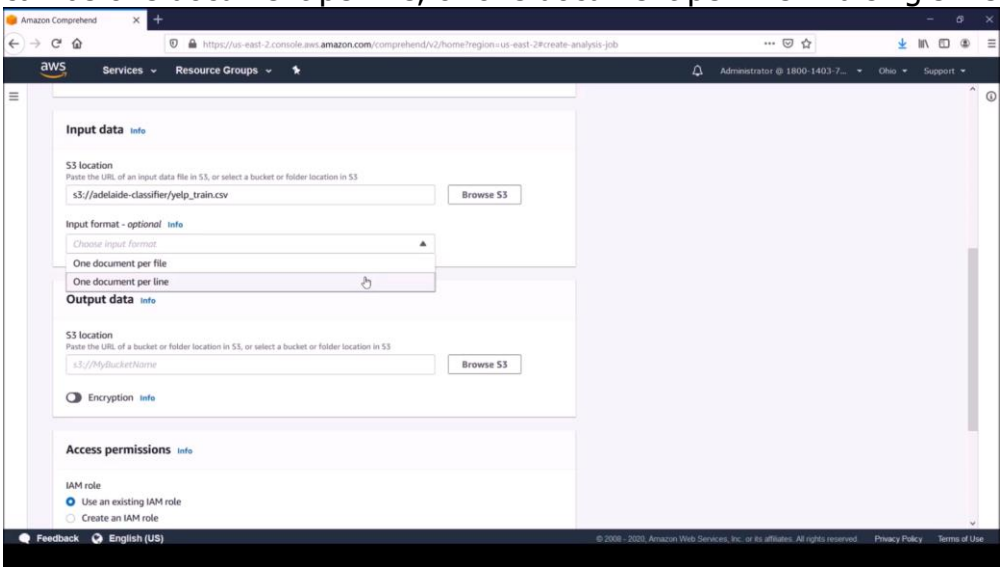
6. If you want to encrypt the data in the storage volume while your training job is being processed, choose Classifier encryption and then choose whether to use a KMS key associated with your current account, or one from another account.
    a. If you are using a key associated with the current account, choose the key ID for KMS key ID.
    b. If you are using a key associated with a different account, enter the ARN for the key ID under KMS key ARN.

7. Under Input data, enter the location of the Amazon S3 bucket that contains your input documents or navigate to it by choosing Browse S3. This bucket must be in the same region as the API that you are calling. The IAM role you're using for

access permissions for the classification job must have reading permissions for the S3 bucket.

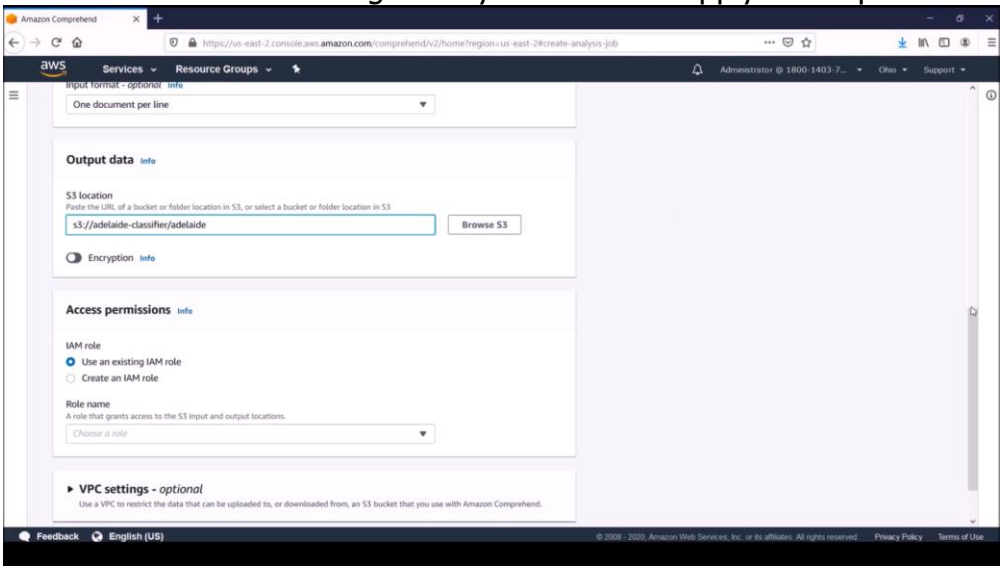


8. Choose the format of the documents to be classified under Input format. These can be one document per file, or one document per line in a single file.



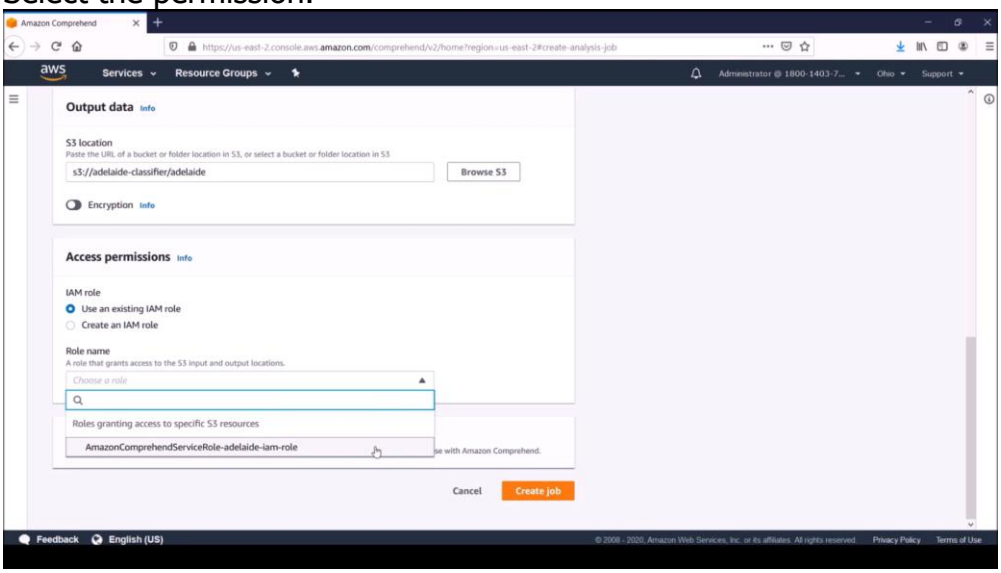
9. Under Output data, enter the location of the Amazon S3 bucket where Amazon Comprehend should write the job's output data or navigate to it by choosing Select folder. This bucket must be in the same region as the API that you are calling. The IAM role you're using for access permissions for the classification job must have write permissions for the S3 bucket.

Click Browse S3 then navigate to your bucket. Supply an output folder.



10. If you choose to encrypt the output result from your job, choose Encryption and then choose whether to use a KMS key associated with the current account, or one from another account.
    a. If you are using a key associated with the current account, choose the key alias or ID for KMS key ID.
    b. If you are using a key associated with a different account, enter the ARN for the key alias or ID under KMS key ID.

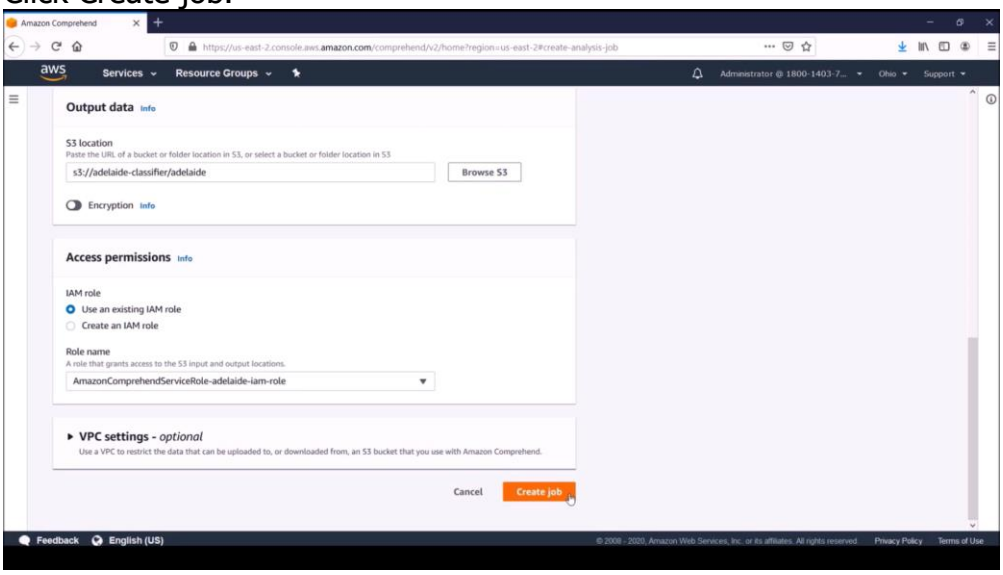
11. Select the permission.



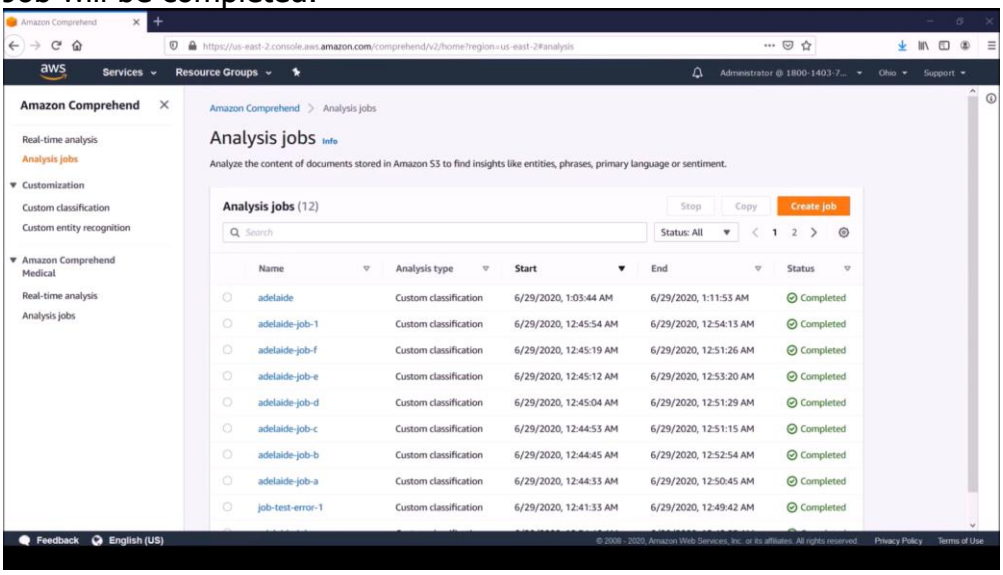
12. To launch your resources into AWS Comprehend from a VPC (Virtual Private Cloud), enter the VPC ID under VPC or choose the ID from the drop-down list.
    a. Choose the subnet under Subnets(s). After you select the first subnet, you can choose additional ones.

b. Under Security Group(s), choose the security group to use if you specified one. After you select the first security group, you can choose additional ones.

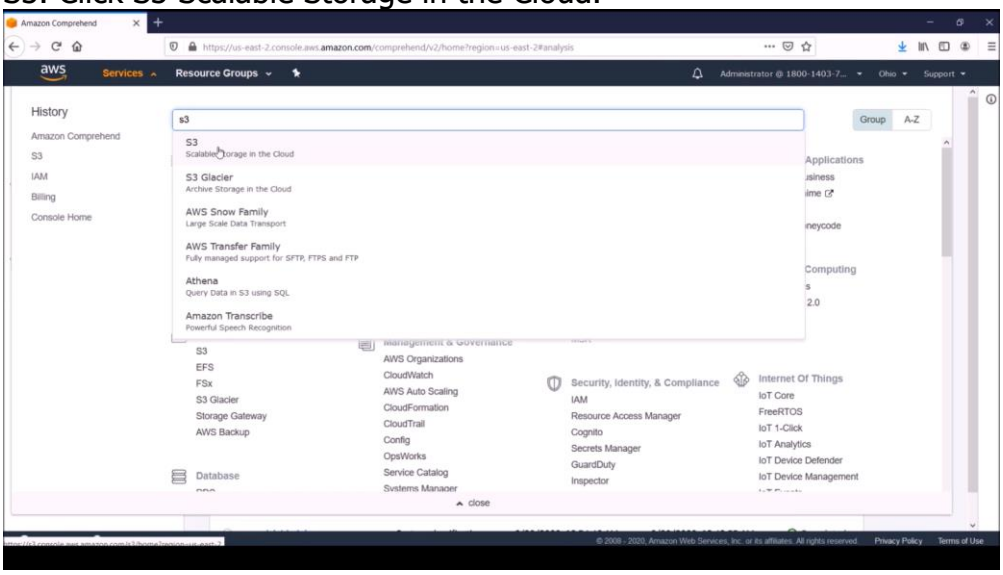
13. Click Create job.



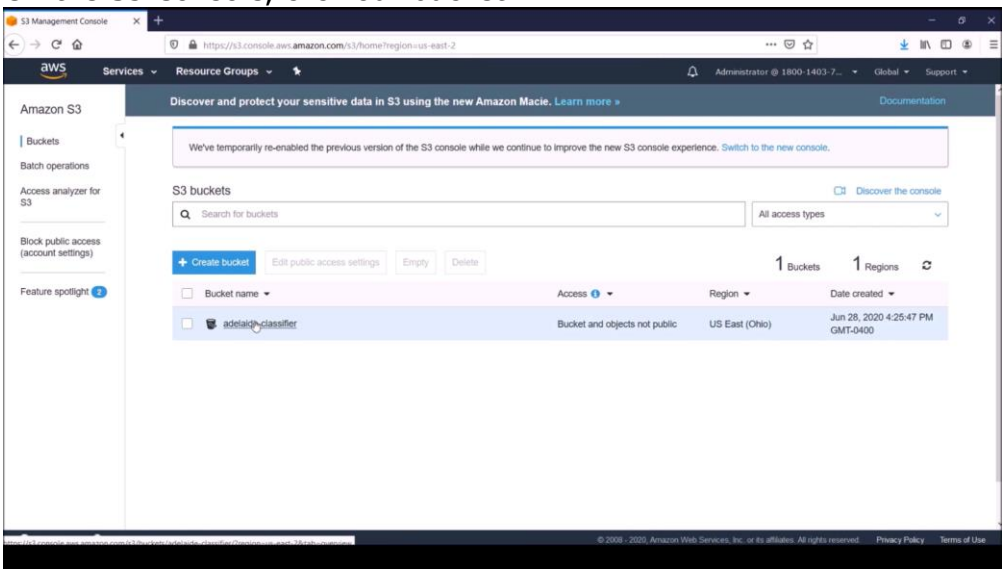
14. Job will be completed.



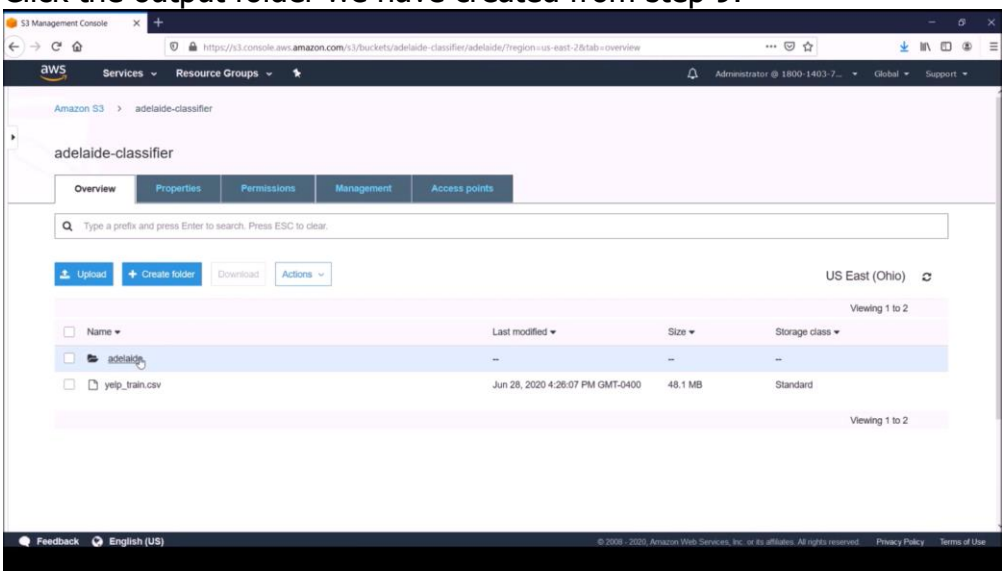
15. Check the result on S3 by clicking Services on the navigation bar and searching for S3. Click S3 Scalable Storage in the Cloud.



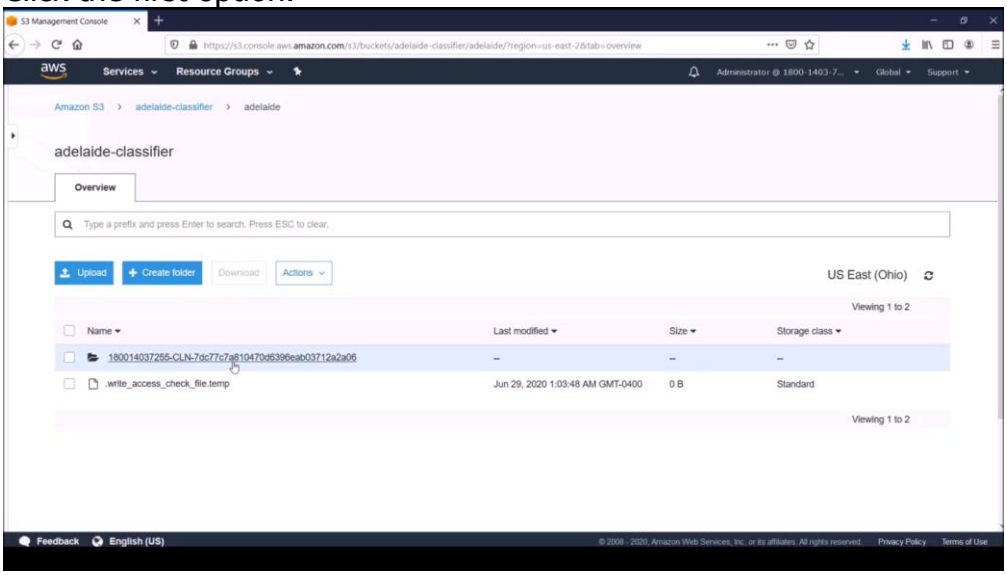
16. On the S3 Console, click our bucket.



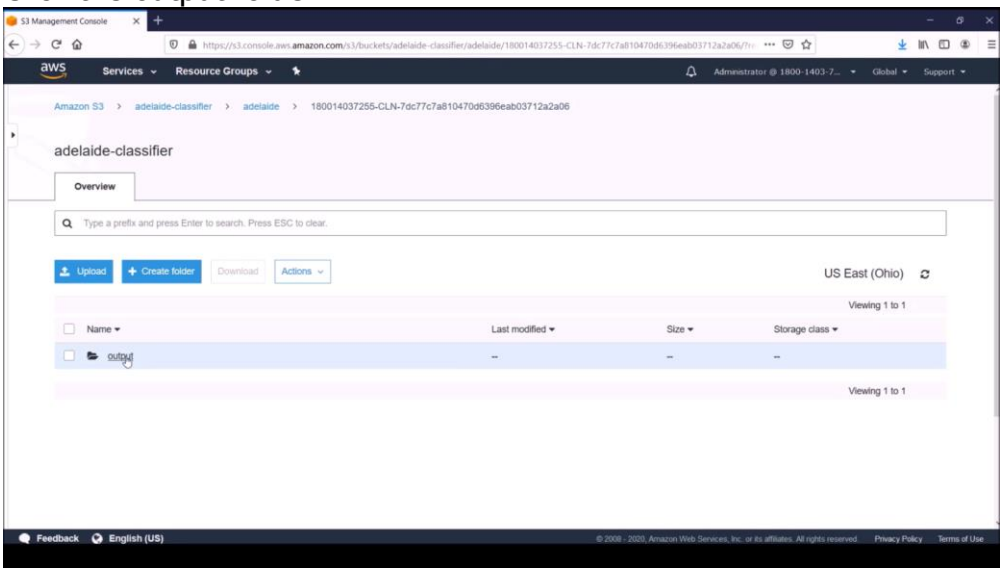
17. Click the output folder we have created from step 9.



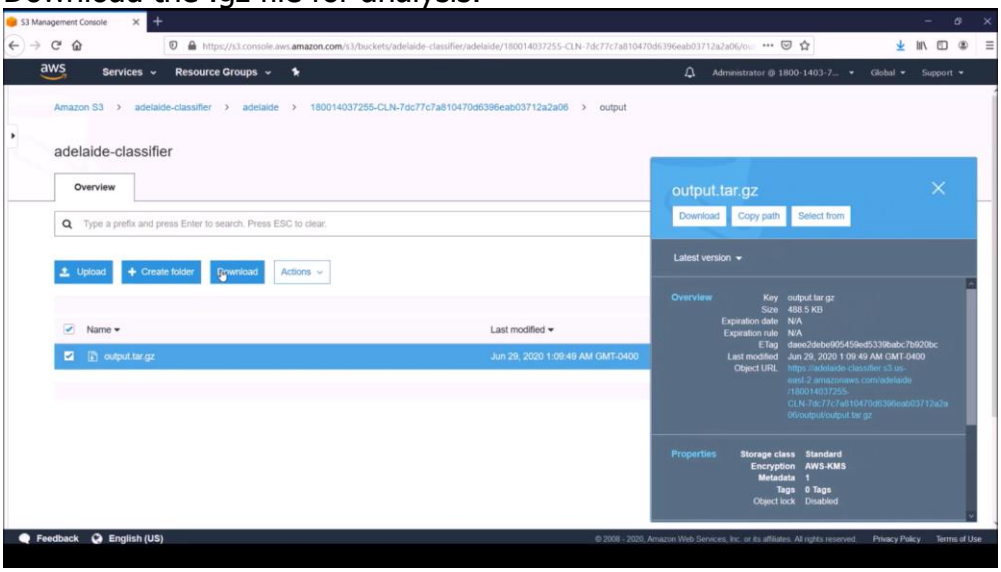
18. Click the first option.



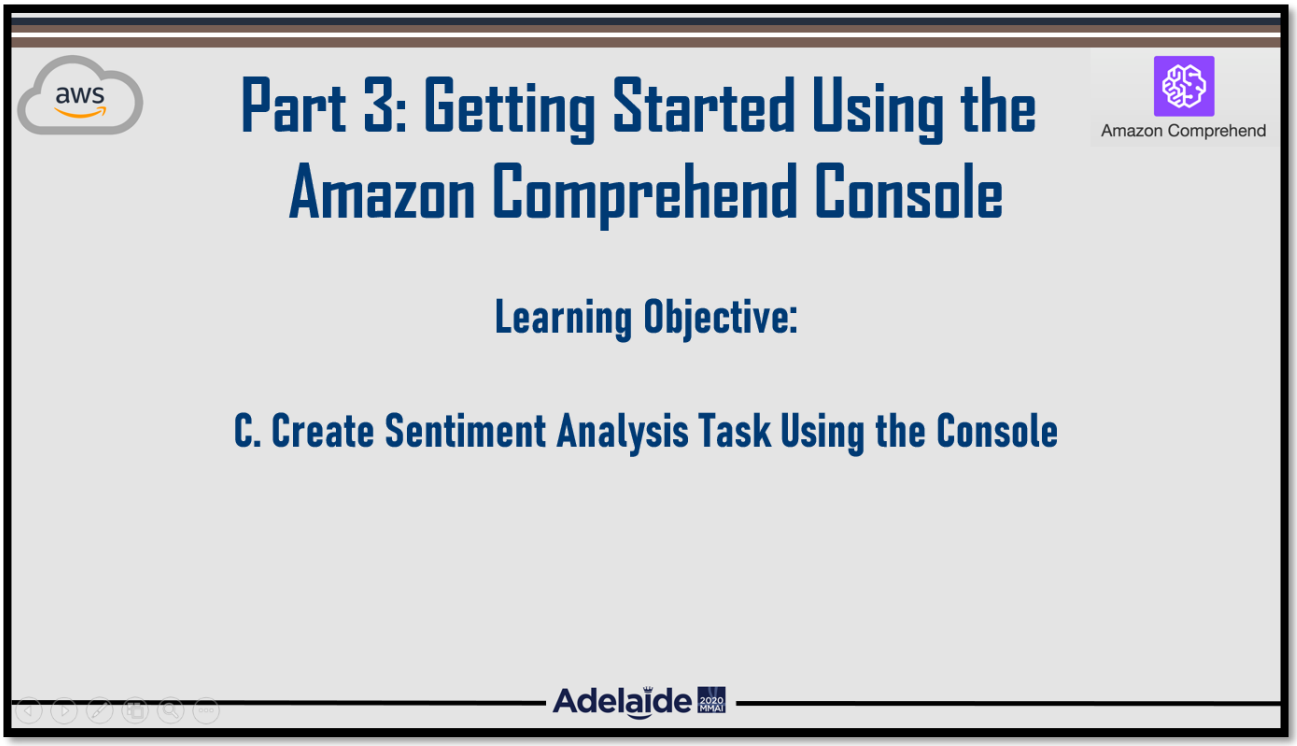
19. Click the output folder.



20. Download the .gz file for analysis.

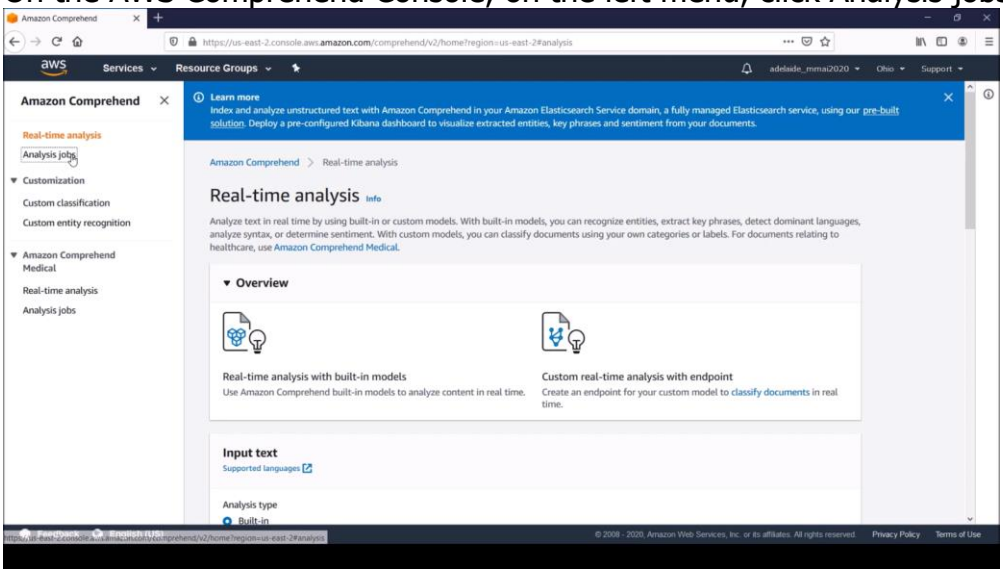


21. Now it's your turn! Let's apply what we've just learned about AWS Comprehend.

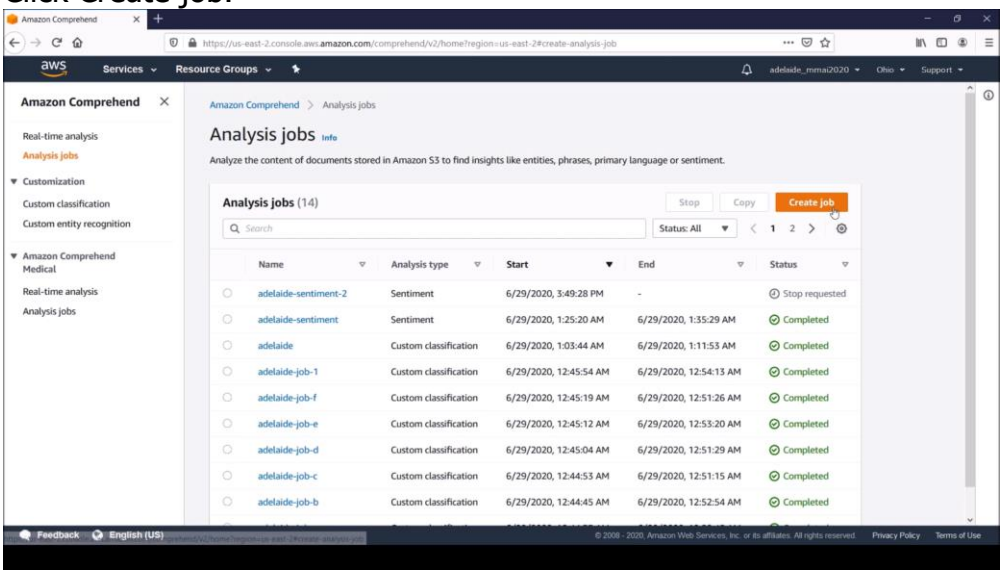


Create a Sentiment Analysis Task Using the Console

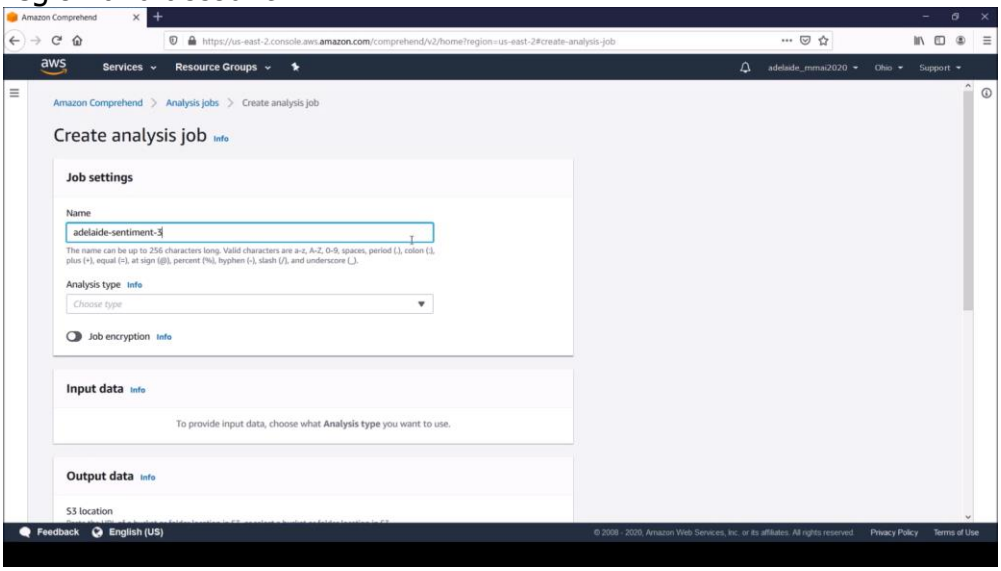
1. On the AWS Comprehend Console, on the left menu, click Analysis jobs.



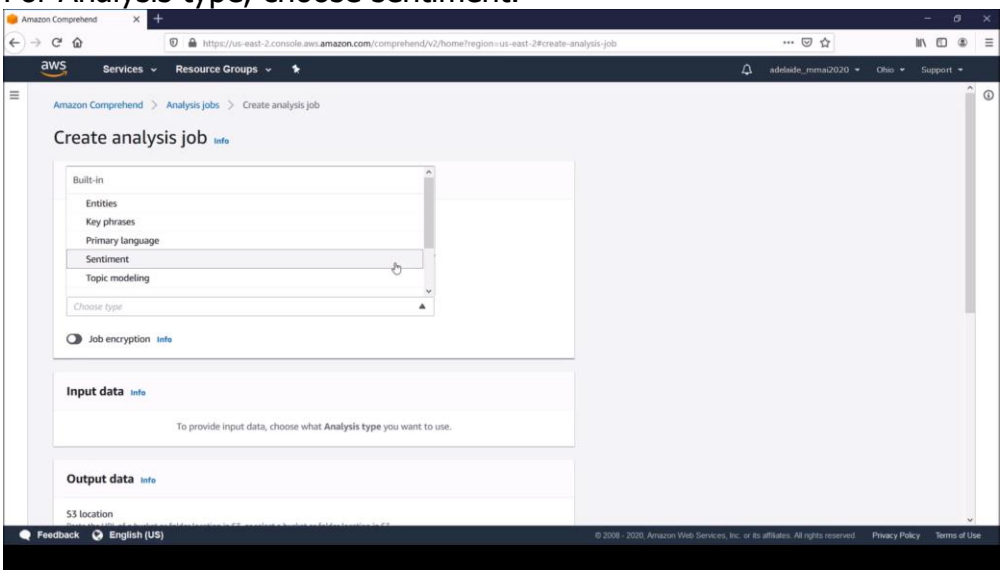
2. Click Create job.



3. Under Job settings, give the job a name. The name must be unique within the region and account.

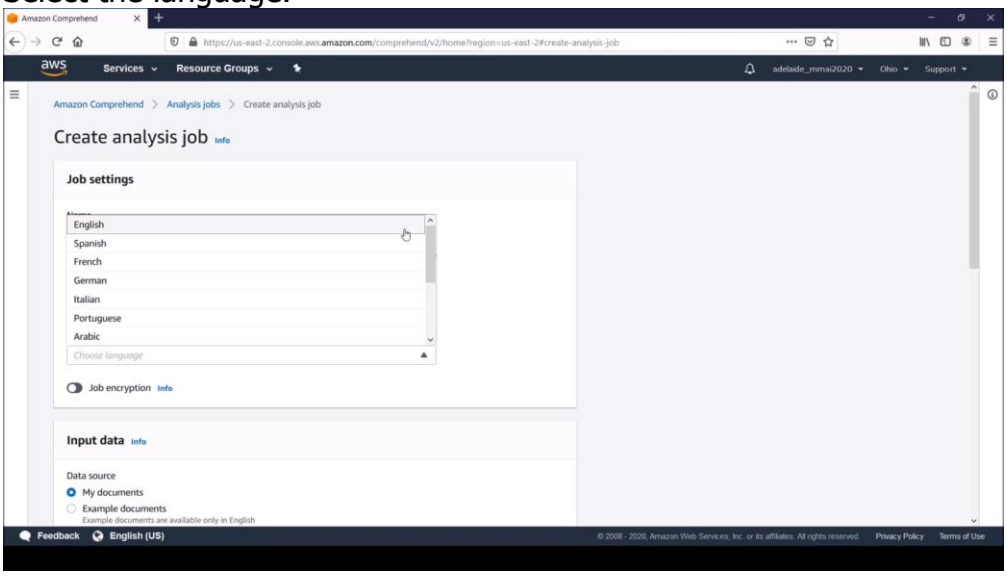


4. For Analysis type, choose sentiment.



You can also select other task such as Entities, Key phrases, Primary language, or Topic modeling.

5. Select the language.



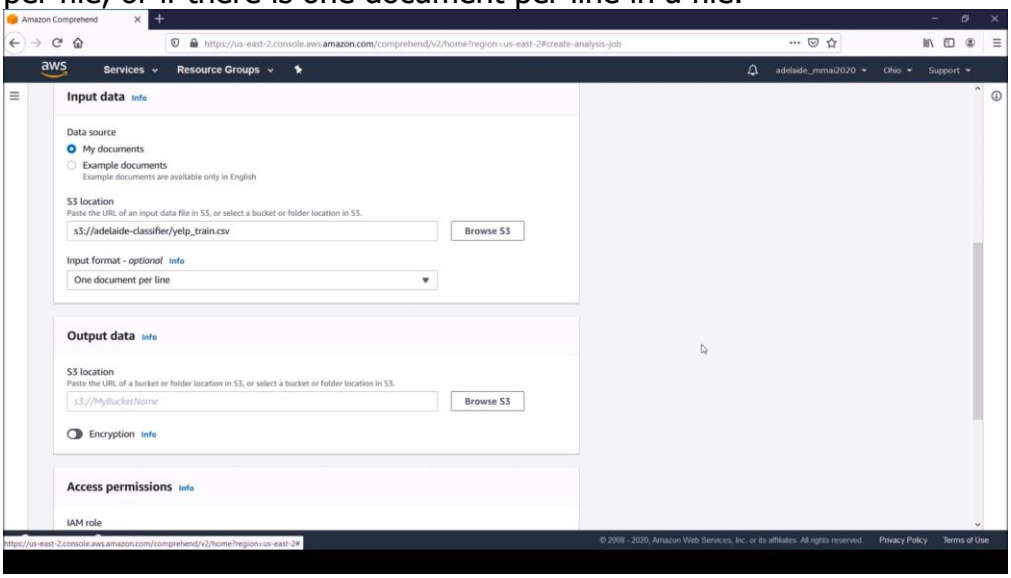
6. If you want to encrypt the data in the storage volume while your training job is being processed, choose Classifier encryption and then choose whether to use a KMS key associated with your current account, or one from another account.
    a. If you are using a key associated with the current account, choose the key ID for KMS key ID.
    b. If you are using a key associated with a different account, enter the ARN for the key ID under KMS key ARN.

7. Choose the data source to use, click Browse S3. You can use either sample data or you can analyze your own data stored in an Amazon S3 bucket.

   If you choose to use your own data, provide the following information:

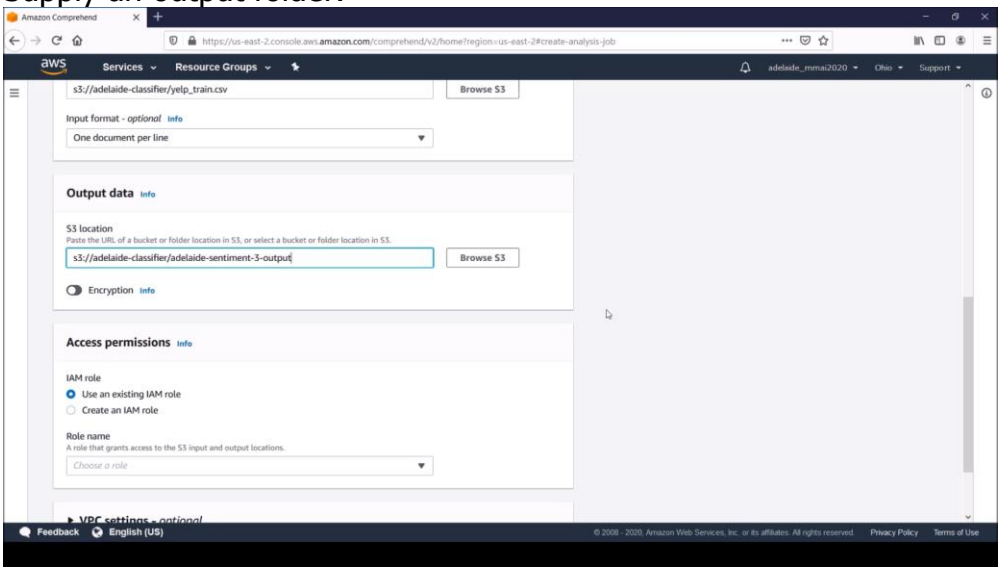
   S3 data location – An Amazon S3 data bucket that contains the documents to analyze. You can choose the folder icon to browse to the location of your data. The bucket must be in the same region as the API that you are calling.

   Input format – Optionally choose whether input data is contained in one document per file, or if there is one document per line in a file.



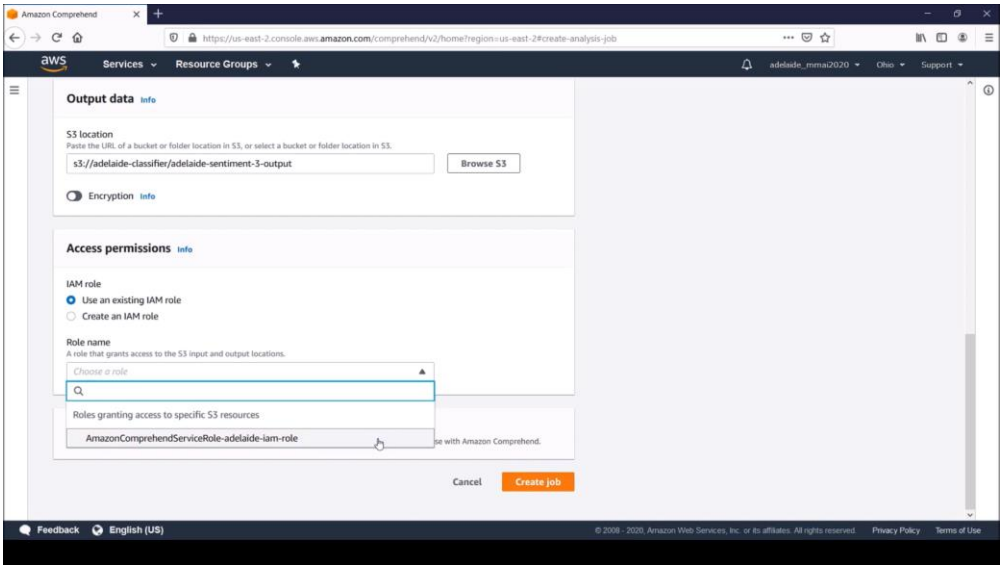
8. Supply an output data location, click Browse S3.

   Supply an output folder.



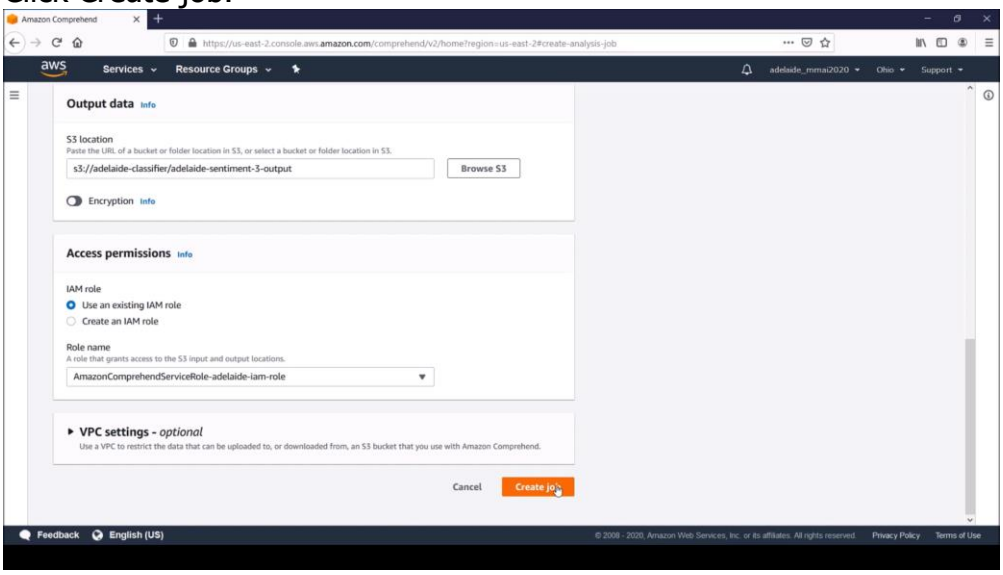
9. If you choose to encrypt the output result from your job, choose Encryption and then choose whether to use a KMS key associated with the current account, or one from another account.
    a. If you are using a key associated with the current account, choose the key alias or ID for KMS key ID.
    b. If you are using a key associated with a different account, enter the ARN for the key alias or ID under KMS key ID.

10. Select the IAM role.

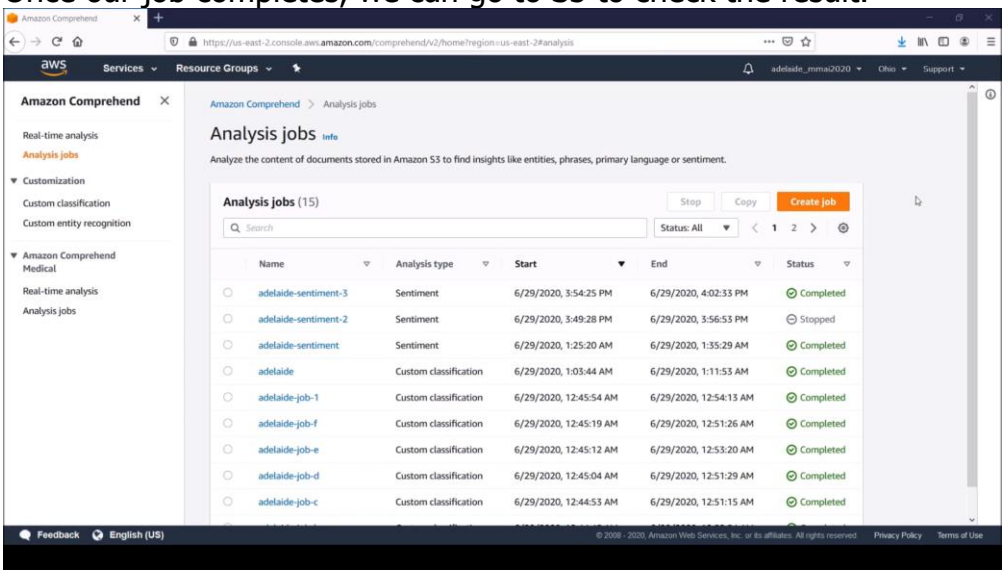


11. To launch your resources into AWS Comprehend from a VPC (Virtual Private Cloud), enter the VPC ID under VPC or choose the ID from the drop-down list.
    a. Choose the subnet under Subnet(s). After you select the first subnet, you can choose additional ones.
    b. Under Security Group(s), choose the security group to use if you specified one. After you select the first security group, you can choose additional ones.

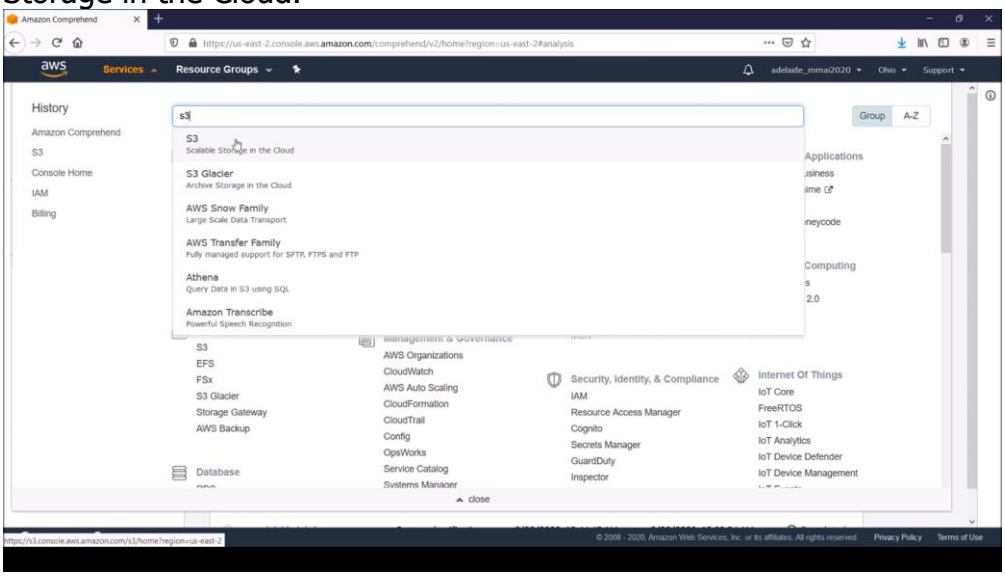
12. Click Create job.



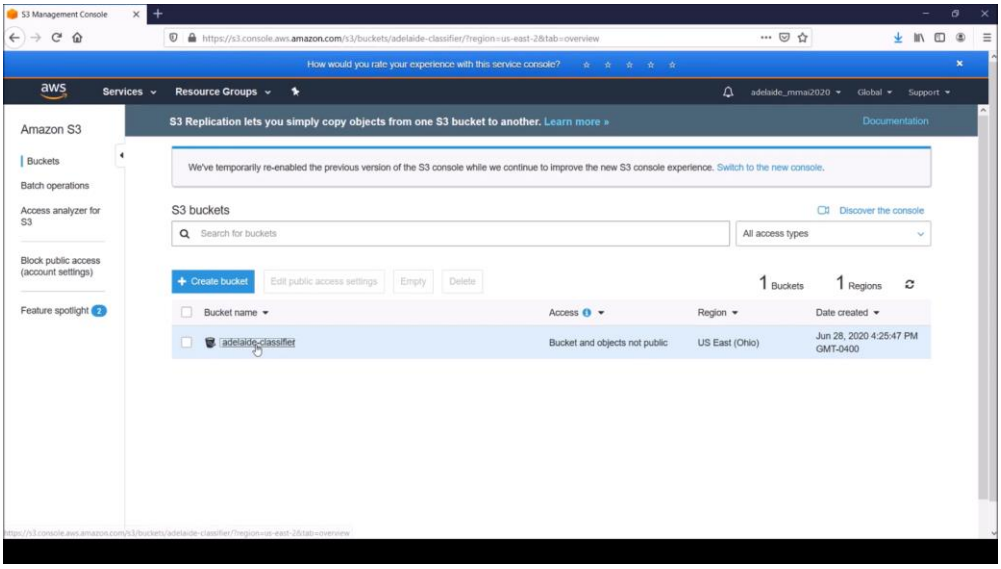
13. Once our job completes, we can go to S3 to check the result.



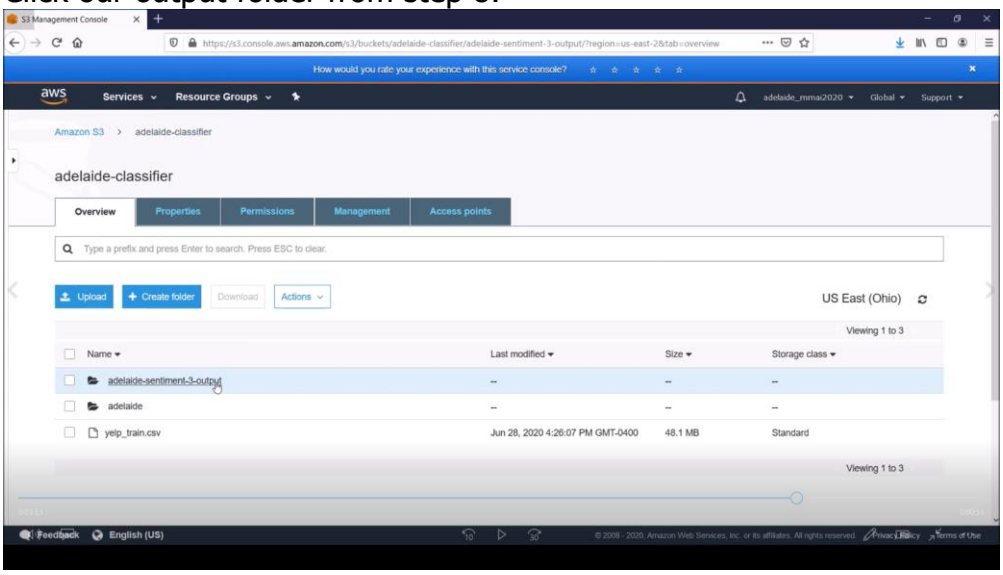
14. On the navigation bar, click on Services, search for s3, click on S3 Scalable Storage in the Cloud.



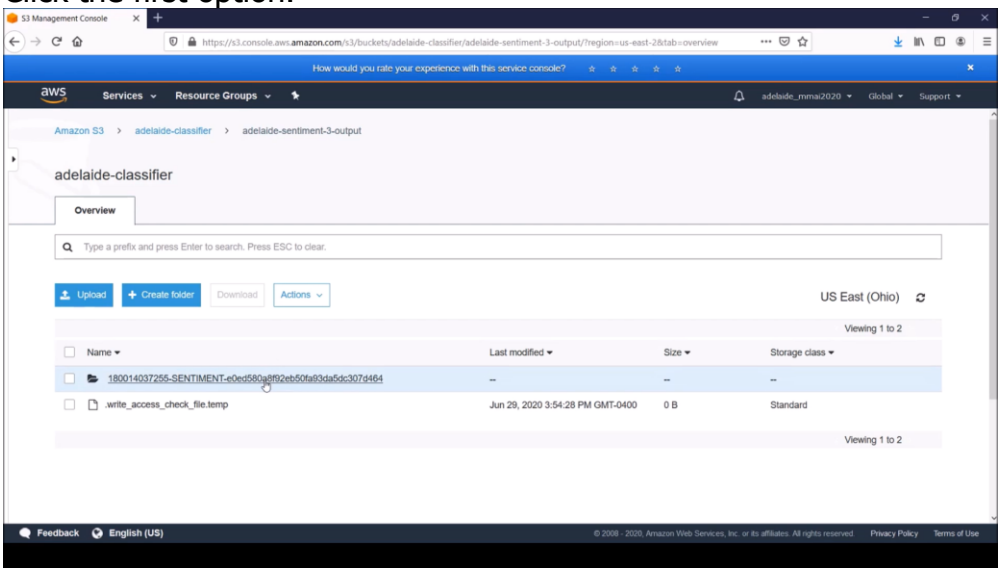
15. Click our bucket.



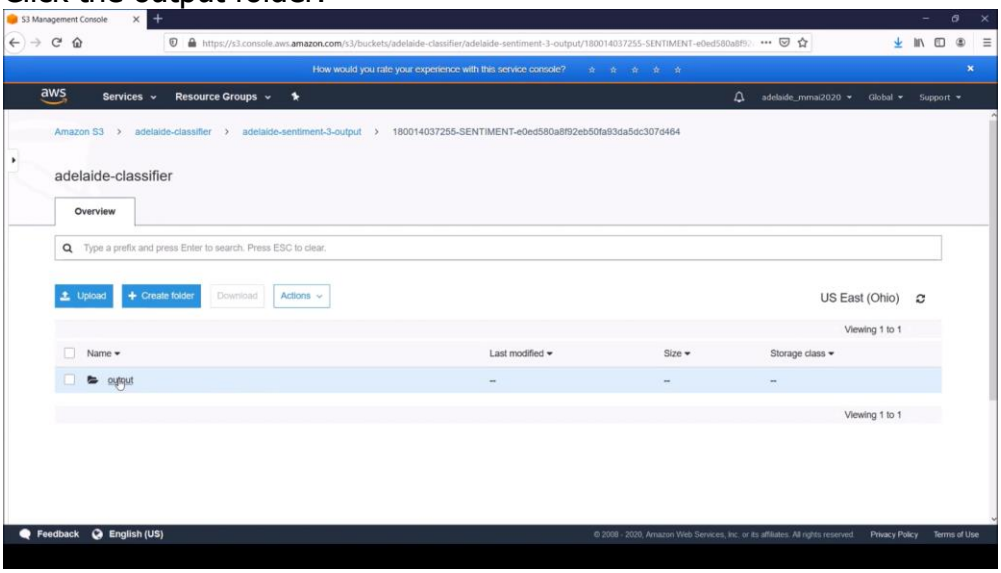
16. Click our output folder from step 8.



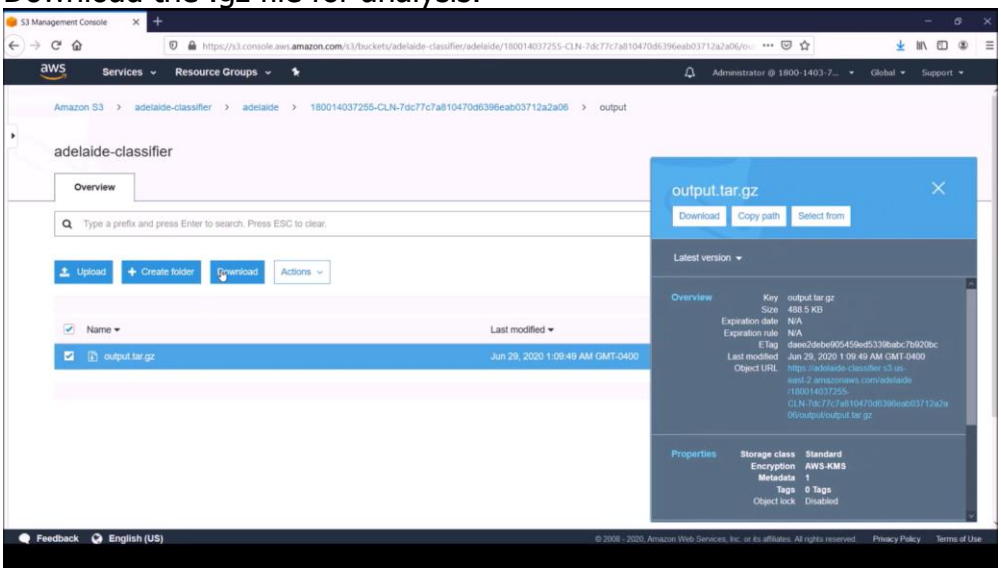
17. Click the first option.



18. Click the output folder.

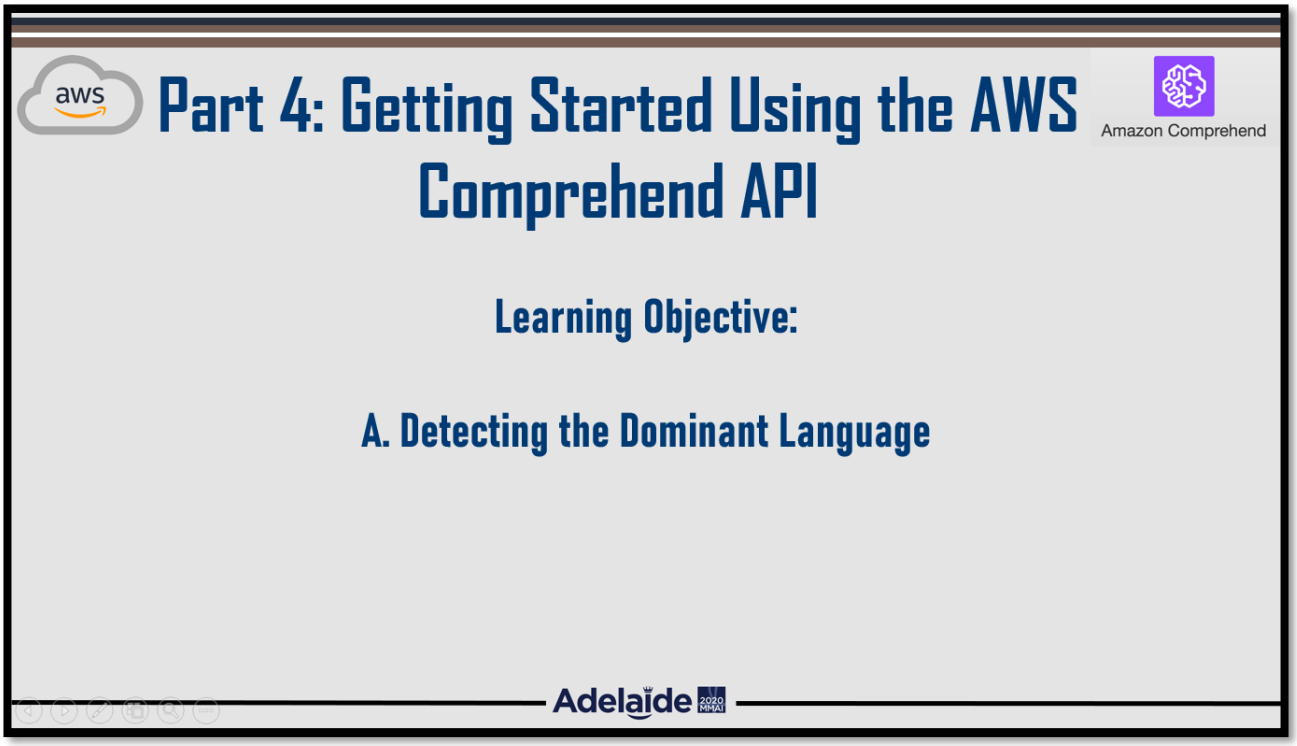


19. Download the .gz file for analysis.



20. Now it's your turn! Let's apply what we've just learned about AWS Comprehend.

Part 4: Getting Started Using the AWS Comprehend API



1. On Jupyter, import boto3 (see Appendix III: Install boto3).

```
In [1]:  1  import boto3
```

2. Create an instance of boto3 (see Appendix II: Creating an AWS Comprehend Access Key).

```
In [2]:  1  comprehend = boto3.client('comprehend',
         2                             region_name='us-east-2',
         3                             aws_access_key_id="AKIAST2NMHED6F25KTJL",
         4                             aws_secret_access_key="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")
```

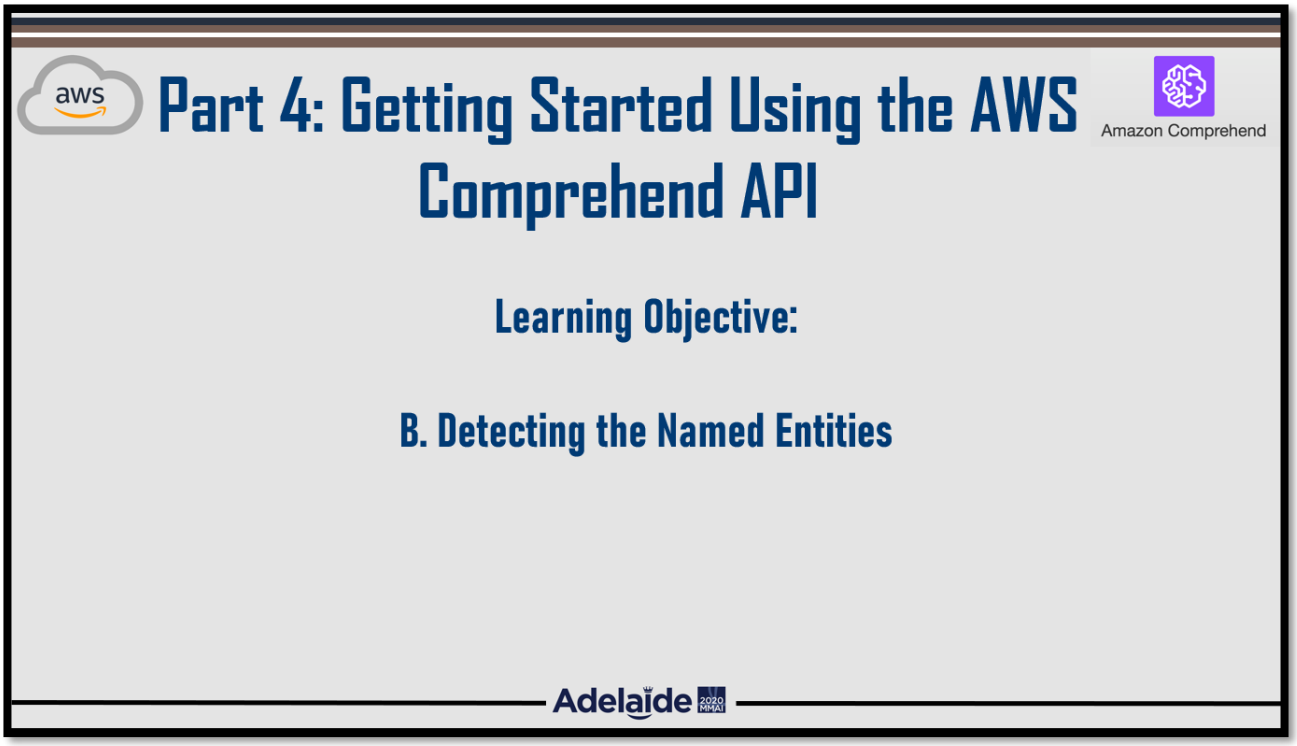
3. Define a text variable.

```
In [3]:  1  sample_tweet="It's always great when I can randomly put my equestrian knowledge to good use at work! Queen's University!"
```

4. Call the detect_dominant_language() method of boto3.

```
In [4]:  1  dom_lang = comprehend.detect_dominant_language(Text=sample_tweet)
         2  print(dom_lang)
```

```
{'Languages': [{'LanguageCode': 'en', 'Score': 0.9916251301765442}], 'ResponseMetadata': {'RequestId': '34603609-623a-49dd-96f4
-030e0dfafddd', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': '34603609-623a-49dd-96f4-030e0dfafddd', 'content-typ
e': 'application/x-amz-json-1.1', 'content-length': '64', 'date': 'Wed, 08 Jul 2020 03:54:00 GMT'}, 'RetryAttempts': 0}}
```



1. On Jupyter, import boto3 (see Appendix III: Install boto3).

```
In [1]:  1  import boto3
```

2. Create an instance of boto3 (see Appendix II: Creating an AWS Comprehend Access Key).

```
In [2]:  1  comprehend = boto3.client('comprehend',
         2                             region_name='us-east-2',
         3                             aws_access_key_id="AKIAST2NMHED6F25KTJL",
         4                             aws_secret_access_key="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")
```

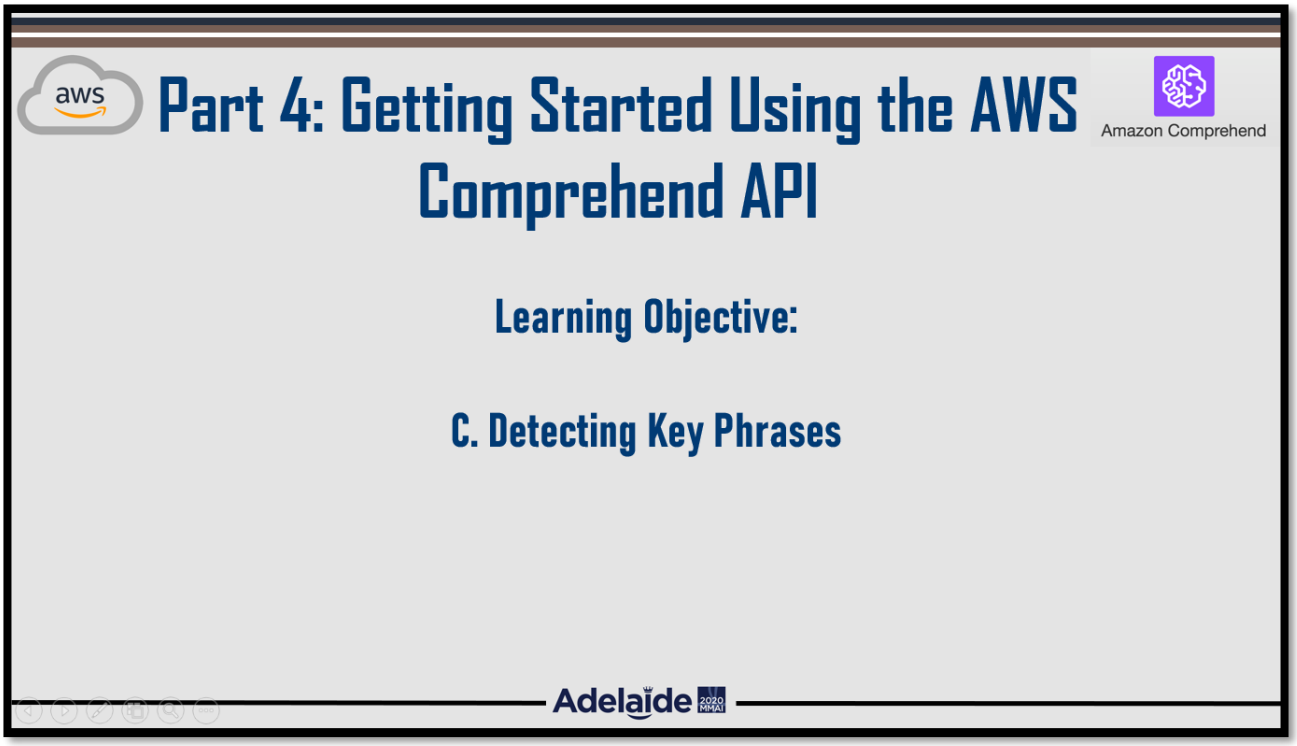
3. Define a text variable.

```
In [3]:  1  sample_tweet="It's always great when I can randomly put my equestrian knowledge to good use at work! Queen's University!"
```

4. Call the detect_entities() method of boto3.

```
In [6]:  1  entities = comprehend.detect_entities(Text=sample_tweet, LanguageCode='en')
         2
         3  print('------- entity : entity type ---------')
         4  for i in range(0, len(entities['Entities'])):
         5      print(entities['Entities'][i]['Text'] + ' : ' + entities['Entities'][i]['Type'] )
```

```
------- entity : entity type ---------
Queen's University : ORGANIZATION
```



1. On Jupyter, import boto3 (see Appendix III: Install boto3).

```
In [1]:  1  import boto3
```

2. Create an instance of boto3 (see Appendix II: Creating an AWS Comprehend Access Key).

```
In [2]:  1  comprehend = boto3.client('comprehend',
         2                             region_name='us-east-2',
         3                             aws_access_key_id="AKIAST2NMHED6F25KTJL",
         4                             aws_secret_access_key="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")
```

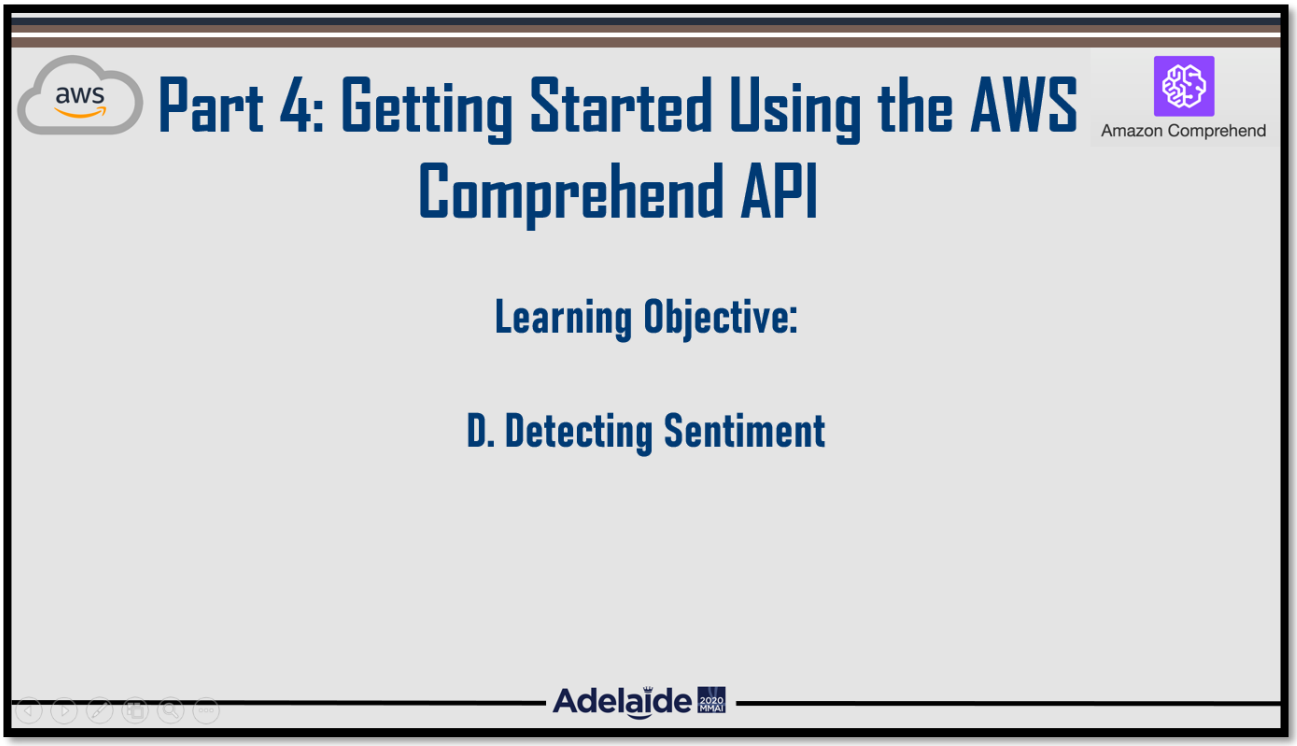
3. Define a text variable.

```
In [3]:  1  sample_tweet="It's always great when I can randomly put my equestrian knowledge to good use at work! Queen's University!"
```

4. Call the detect_key_phrases() method of boto3.

```
In [5]:  1  # Key phrases
         2  phrases = comprehend.detect_key_phrases(Text=sample_tweet, LanguageCode='en')
         3
         4  # Print the phrases:
         5  print('------- phrases ---------')
         6  for i in range(0, len(phrases['KeyPhrases'])):
         7      print((phrases['KeyPhrases'][i]['Text']))
```

```
------- phrases ---------
my equestrian knowledge
good use
work
Queen
University
```



1. On Jupyter, import boto3 (see Appendix III: Install boto3).

```
In [1]:  1  import boto3
```

2. Create an instance of boto3 (see Appendix II: Creating an AWS Comprehend Access Key).

```
In [2]:  1  comprehend = boto3.client('comprehend',
         2                             region_name='us-east-2',
         3                             aws_access_key_id="AKIAST2NMHED6F25KTJL",
         4                             aws_secret_access_key="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")
```

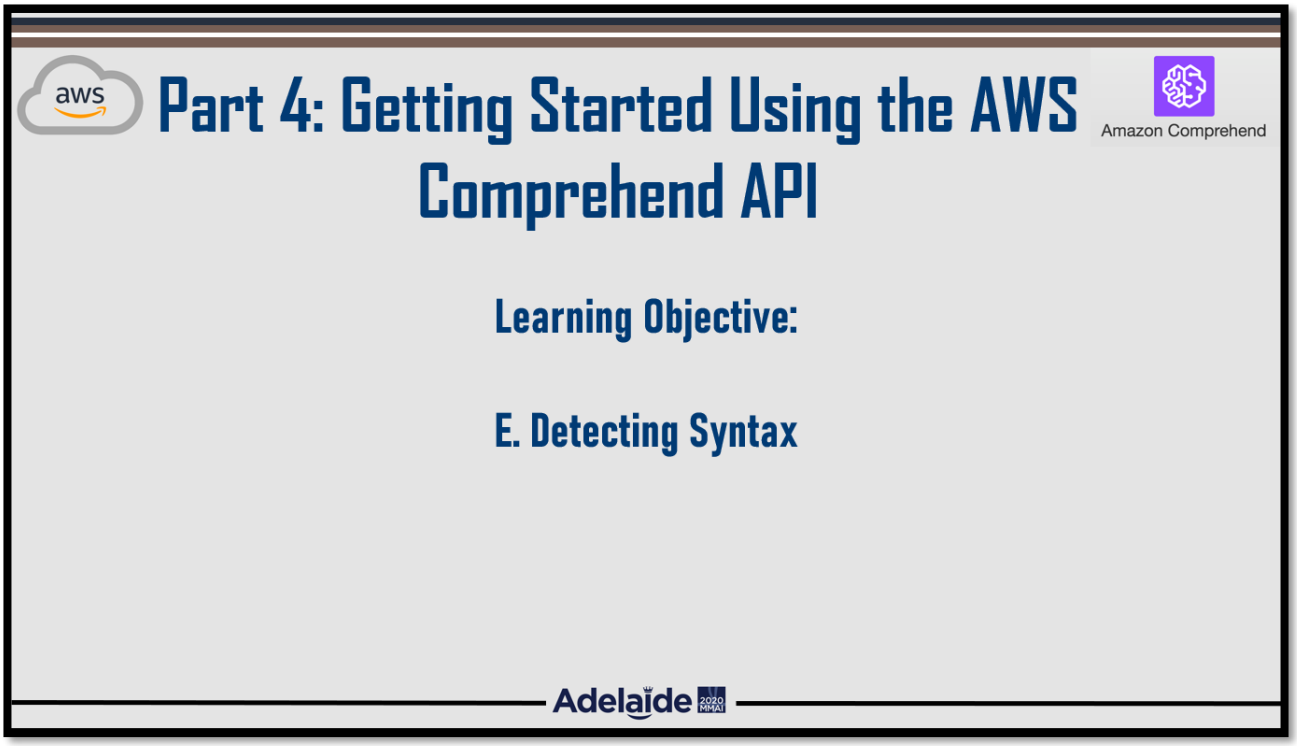
3. Define a text variable.

```
In [3]:  1  sample_tweet="It's always great when I can randomly put my equestrian knowledge to good use at work! Queen's University!"
```

4. Call the detect_sentiment() method of boto3.

```
In [7]:  1  sentiments = comprehend.detect_sentiment(Text=sample_tweet, LanguageCode='en')
         2
         3  print('------- sentiment ---------')
         4  print(sentiments['Sentiment'])
```

```
------- sentiment ---------
POSITIVE
```



1. On Jupyter, import boto3 (see Appendix III: Install boto3).

```
In [1]:  1  import boto3
```

2. Create an instance of boto3 (see Appendix II: Creating an AWS Comprehend Access Key).

```
In [2]:  1  comprehend = boto3.client('comprehend',
         2                            region_name='us-east-2',
         3                            aws_access_key_id="AKIAST2NMHED6F25KTJL",
         4                            aws_secret_access_key="cD8hlkYKEx8OuccJ1pxconMS7ANDDgyMGsJI7Wrp")
```

3. Define a text variable.

```
In [3]:  1  sample_tweet="It's always great when I can randomly put my equestrian knowledge to good use at work! Queen's University!"
```

4. Call the detect_syntax() method of boto3.

```
In [8]:  1  #Syntax
         2  syntax = comprehend.detect_syntax(Text=sample_tweet, LanguageCode='en')
         3
         4  print('-------syntax-------')
         5  print(syntax)
```

```
-------syntax-------
{'SyntaxTokens': [{'TokenId': 1, 'Text': 'It', 'BeginOffset': 0, 'EndOffset': 2, 'PartOfSpeech': {'Tag': 'PRON', 'Score': 0.999
9947547912598}}, {'TokenId': 2, 'Text': '’s', 'BeginOffset': 2, 'EndOffset': 4, 'PartOfSpeech': {'Tag': 'VERB', 'Score': 0.9995
766282081604}}, {'TokenId': 3, 'Text': 'always', 'BeginOffset': 5, 'EndOffset': 11, 'PartOfSpeech': {'Tag': 'ADV', 'Score': 0.9
999983310699463}}, {'TokenId': 4, 'Text': 'great', 'BeginOffset': 12, 'EndOffset': 17, 'PartOfSpeech': {'Tag': 'ADJ', 'Score':
0.9972237348556519}}, {'TokenId': 5, 'Text': 'when', 'BeginOffset': 18, 'EndOffset': 22, 'PartOfSpeech': {'Tag': 'ADV', 'Scor
e': 0.9999743700027466}}, {'TokenId': 6, 'Text': 'I', 'BeginOffset': 23, 'EndOffset': 24, 'PartOfSpeech': {'Tag': 'PRON', 'Scor
e': 0.9998840093612671}}, {'TokenId': 7, 'Text': 'can', 'BeginOffset': 25, 'EndOffset': 28, 'PartOfSpeech': {'Tag': 'AUX', 'Sco
re': 0.9999958276748657}}, {'TokenId': 8, 'Text': 'randomly', 'BeginOffset': 29, 'EndOffset': 37, 'PartOfSpeech': {'Tag': 'AD
V', 'Score': 0.9999892711639404}}, {'TokenId': 9, 'Text': 'put', 'BeginOffset': 38, 'EndOffset': 41, 'PartOfSpeech': {'Tag': 'V
ERB', 'Score': 0.999999165534973l}}, {'TokenId': 10, 'Text': 'my', 'BeginOffset': 42, 'EndOffset': 44, 'PartOfSpeech': {'Tag':
'PRON', 'Score': 0.9999688863754272}}, {'TokenId': 11, 'Text': 'equestrian', 'BeginOffset': 45, 'EndOffset': 55, 'PartOfSpeec
h': {'Tag': 'ADJ', 'Score': 0.8308550715446472}}, {'TokenId': 12, 'Text': 'knowledge', 'BeginOffset': 56, 'EndOffset': 65, 'Par
tOfSpeech': {'Tag': 'NOUN', 'Score': 0.9999425411224365}}, {'TokenId': 13, 'Text': 'to', 'BeginOffset': 66, 'EndOffset': 68, 'P
artOfSpeech': {'Tag': 'ADP', 'Score': 0.9506584405899048}}, {'TokenId': 14, 'Text': 'good', 'BeginOffset': 69, 'EndOffset': 73,
'PartOfSpeech': {'Tag': 'ADJ', 'Score': 0.9889143109321594}}, {'TokenId': 15, 'Text': 'use', 'BeginOffset': 74, 'EndOffset': 7
7, 'PartOfSpeech': {'Tag': 'NOUN', 'Score': 0.9988011121749878}}, {'TokenId': 16, 'Text': 'at', 'BeginOffset': 78, 'EndOffset':
```

Part D: Trivia Questions


aws
Amazon Comprehend
Level 1 Questions: AWS Comprehend
Adelaide
2020
AI
MMXX

What is not a most common use case of Amazon Comprehend?

a. Voice of customer analytics

b. Identifying the main topics from a library of documents

c. Semantic search

d. Knowledge management and discovery

Correct answer:

b. Identifying the main topics from a library of documents

AWS Comprehend allows import or use of a developer's custom NLP model.

a. Yes

b. No

Correct answer:

b. No

What security measures does Amazon Comprehend have?

a. Requests to the API and console are made over a secure (SSL) connection

b. Comprehend utilizes a checksum which ensures that no one tampers with the contents of a message and prevents a malevolent actor from intercepting a legitimate service call and changing it to perform an unacceptable action

c. Focused on stopping threats at the perimeter of the data center

Correct Answer:

a. Requests to the API and console are made over a secure (SSL) connection

Which of the following do you need to manage with Amazon Comprehend?

a. Scaling of resources

b. Code

c. Training data set

d. None of the above

Correct Answer:

d. None of the above

Which metric best tells if Amazon Comprehend is giving accurate results?

a. Accuracy score

b. F1 score

c. Confidence score

d. Logarithmic score

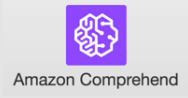
Correct answer:

c. Confidence score


aws
Amazon Comprehend
Congratulations!
You have completed Level 1.
Adelaide
2020
MMXX


aws
Amazon Comprehend
Level 2 Questions: General AWS
Adelaide
2020
MMXX

This AWS service includes EC2, Elastic Beanstalk, Lambda, Auto-Scaling and Lightsat.



a. Computing

b. Storage

c. Networking

Correct answer:

a. Computing

This AWS service includes S3, Glacier, Elastic Block Storage, Elastic File System.

Amazon Comprehend

a. Computing

b. Networking

c. Storage

Correct answer:

c. Storage

In AWS, what is auto-scaling?

Amazon Comprehend

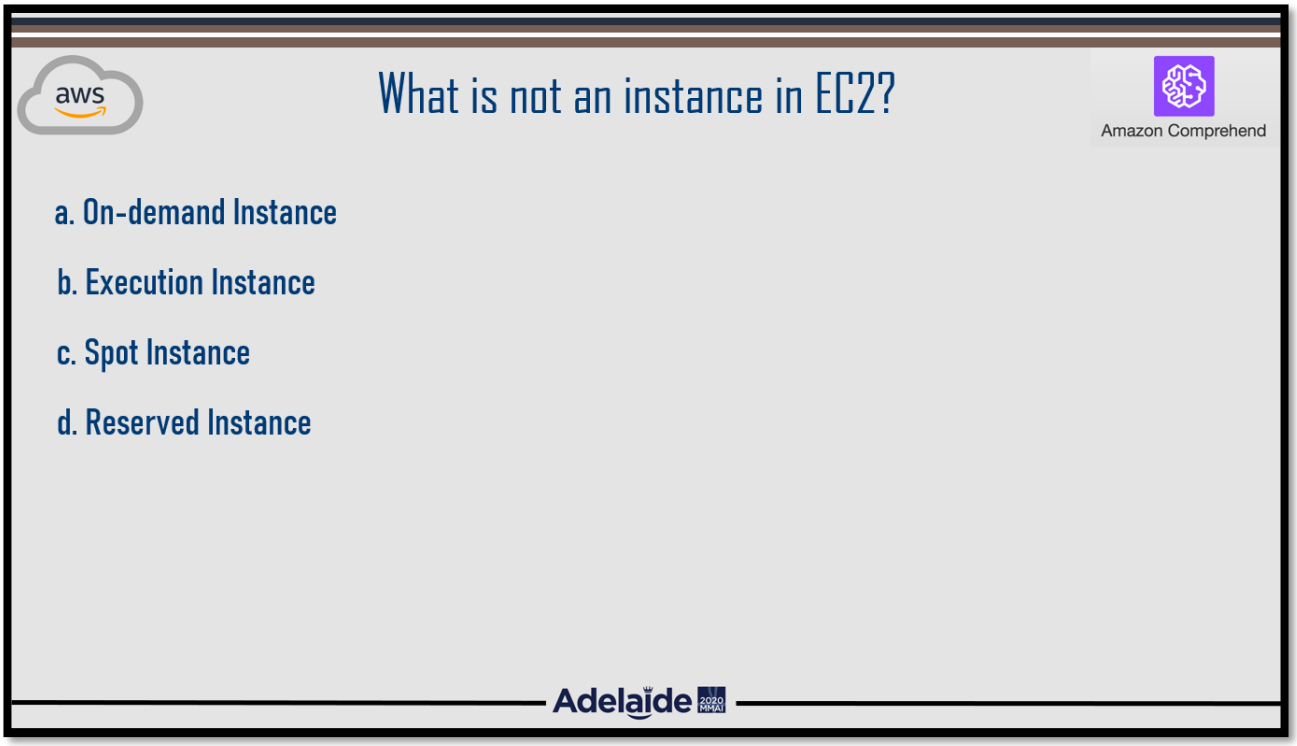
a. A concept where businesses can show personalized content to their audience based on their geographic location without changing the URL

b. Helps to create customized content for the audience of a specific geographical area, keeping their needs in the forefront

c. A function that allows you to provision and launch new instances whenever there is a demand

Correct answer:

c. A function that allows you to provision and launch new instances whenever there is a demand


aws
What is not an instance in EC2?
Amazon Comprehend
a. On-demand Instance
b. Execution Instance
c. Spot Instance
d. Reserved Instance
Adelaide

Correct answer:

b. Execution Instance

What is true about an IAM role?

Amazon Comprehend

a. It defines a set of permissions for making AWS service requests

b. It has full access to all the AWS IAM functionalities

c. It is used to interact with the AWS services directly

Correct answer:

a. It defines a set of permissions for making AWS service requests


aws
Amazon Comprehend
Congratulations!
You have completed Level 2.
Adelaide 2020 MMXI


aws
Amazon Comprehend
Level 3 Questions: Advanced
Adelaide

You are a Machine Learning Engineer who is on the lookout for a solution that will discover sensitive information that your enterprise stores in AWS and then use NLP to classify the data and provide business-related insights. Which among the services would you choose?

a. Firewall Manager

b. IAM

c. Macie

d. CloudHSM

Correct answer:

c. Macie

Suppose you are a game designer and want to develop a game with single-digit millisecond latency, which of the following database services would you use?

a. RDS

b. Neptune

c. Snowball

d. DynamoDB

Correct answer:

d. DynamoDB

If you need to perform real-time monitoring of AWS services and get actionable insights, which service would you use?

a. CloudWatch

b. GuardDuty

c. Firewall Manager

d. EBS

Correct answer:

a. CloudWatch

As a web developer, you are developing an app, targeted especially for the mobile platform. Which of the following lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily?

Amazon Comprehend

a. Shield

b. Macie

c. Inspector

d. Cognito

Correct answer:

d. Cognito

You are a Machine Learning engineer and you are looking for a service that helps you build and train Machine Learning models in AWS. Which among the following is being referred to?

a. SageMaker

b. DeepLens

c. Comprehend

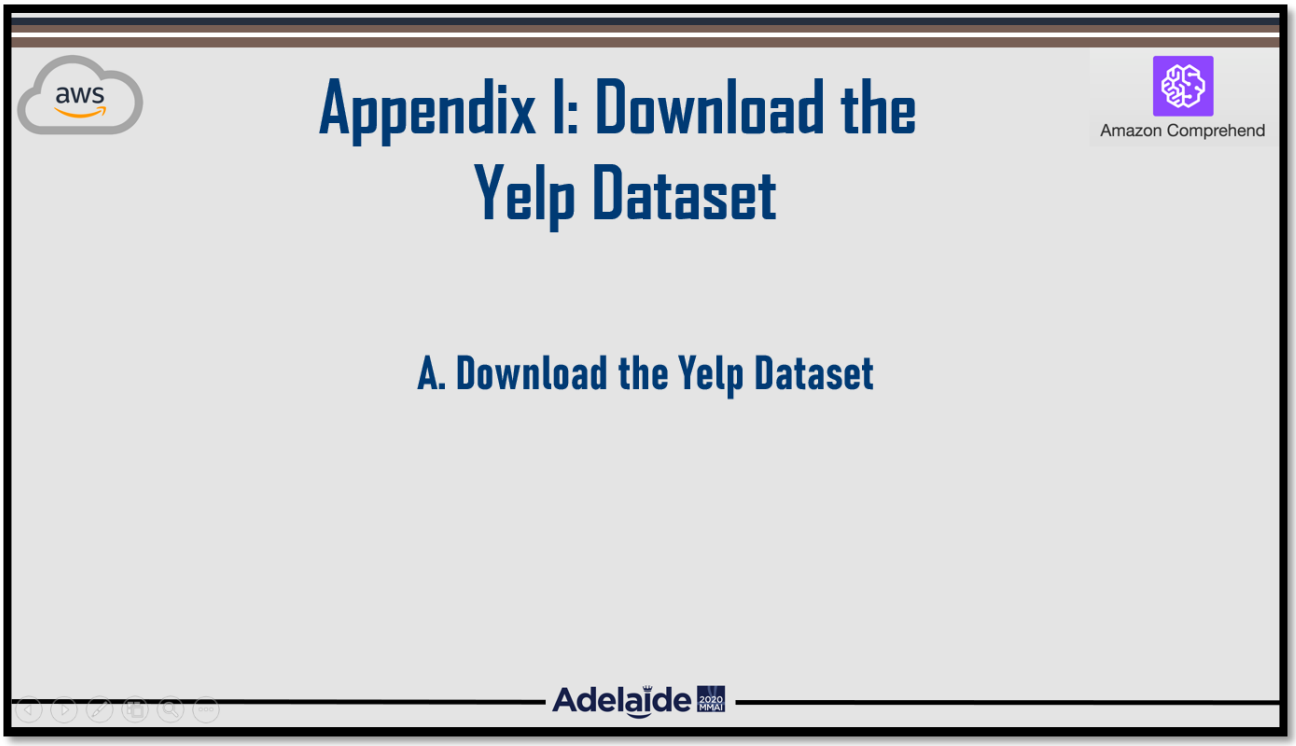
d. Device Farm

Correct answer:

a. SageMaker

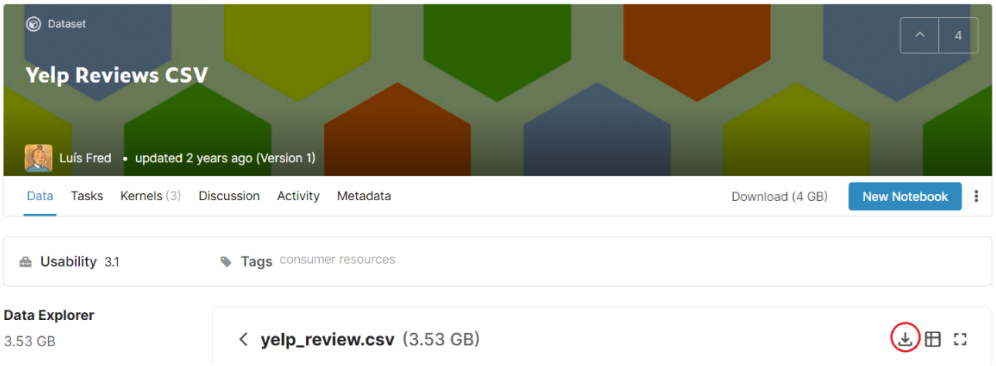

aws
Amazon Comprehend
Congratulations!
You have completed Level 3.
Adelaide
2020
MMXX

Part E: Appendix

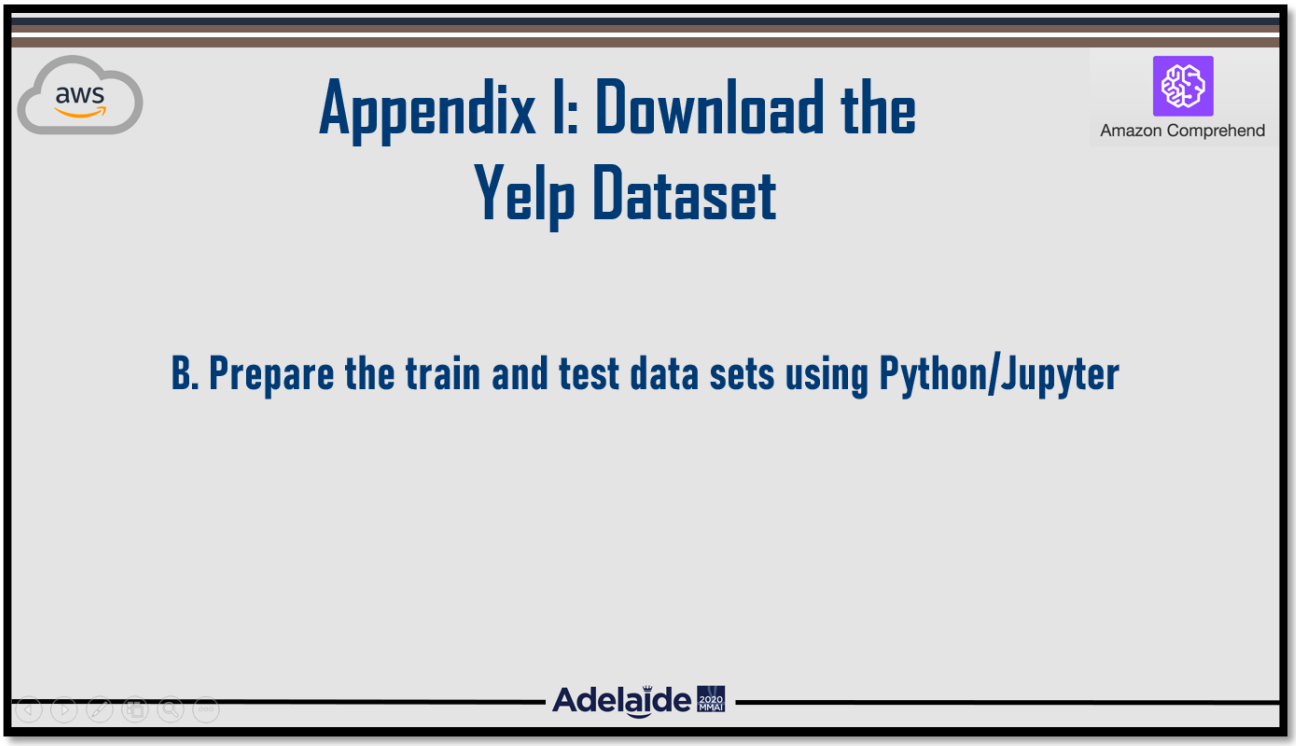


1. Go to https://www.kaggle.com/luisfredgs/yelp-reviews-csv

2. Click on download



3. The file is about 3.53 gigabytes.



1. Import the libraries.

```
In [1]:
1 import pandas as pd
2 import numpy as np
3 from sklearn.model_selection import train_test_split
```

2. Read the file.

```
In [2]:
1 df = pd.read_csv('yelp_reviews_bal.csv')
```

3. Drop the fields that are not needed.

```
In [4]:
1 df = df.drop(['business_id', 'date', 'review_id',
2                'type', 'user_id', 'cool', 'useful', 'funny'], axis=1)
```

4. Remove records having "stars" of 3 and 4.

```
In [5]:
1 df = df.drop(df[(df.stars == 3) | (df.stars == 4)].index)
```

5. Rename the "stars" feature to "sentiment".

```
In [6]:
1 df = df.rename(columns={'stars': 'sentiment'})
```

6. Update records with "sentiment" of 5 to 1, else 0.

```
In [7]:
1 df['sentiment'] = np.where(df['sentiment'] == 5, 1, 0)
```

7. Split the dataset to train and test sets.

```
In [9]:  1  train, test = train_test_split(df, test_size=0.2)
```

8. Save the train and test sets to CSV files.

```
In [10]:  1  train.to_csv('yelp_train.csv', index=False)
```

```
In [11]:  1  test.to_csv('yelp_test.csv', index=False)
```

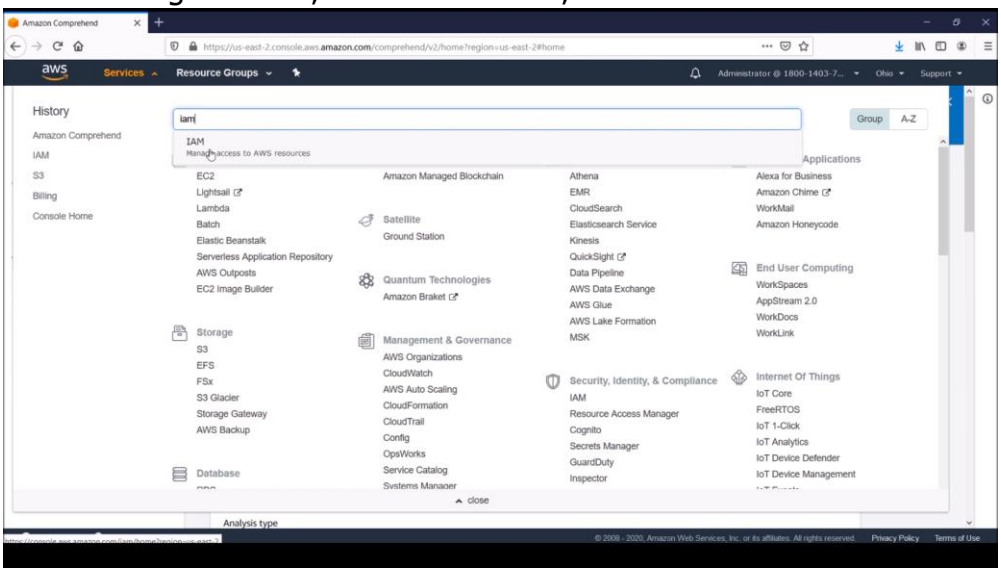
Appendix II: Creating an AWS Comprehend Access Key

Amazon Comprehend

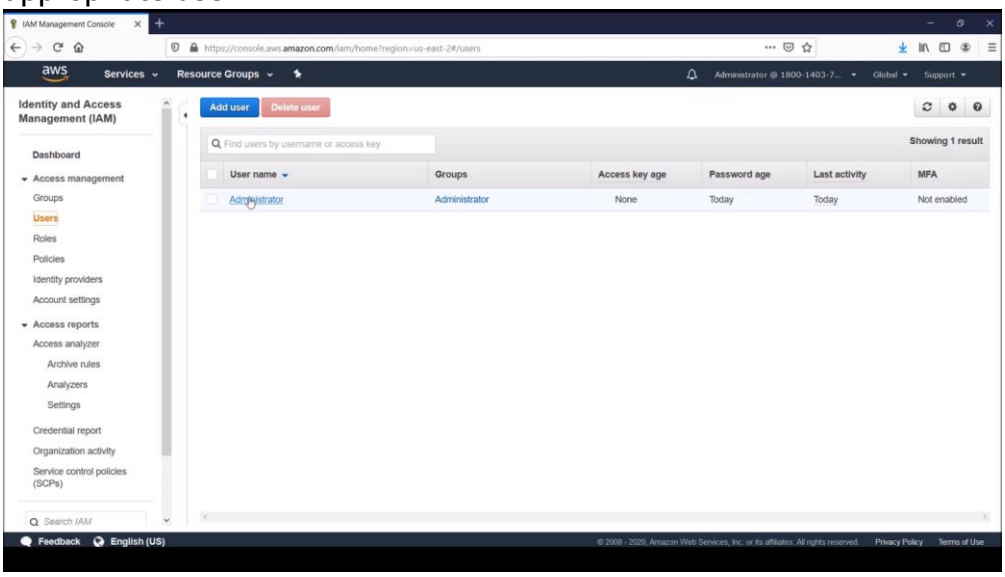
A. Generating the Access Key ID
B. Generating the Secret Access Key

Adelaide 2020 MMAI

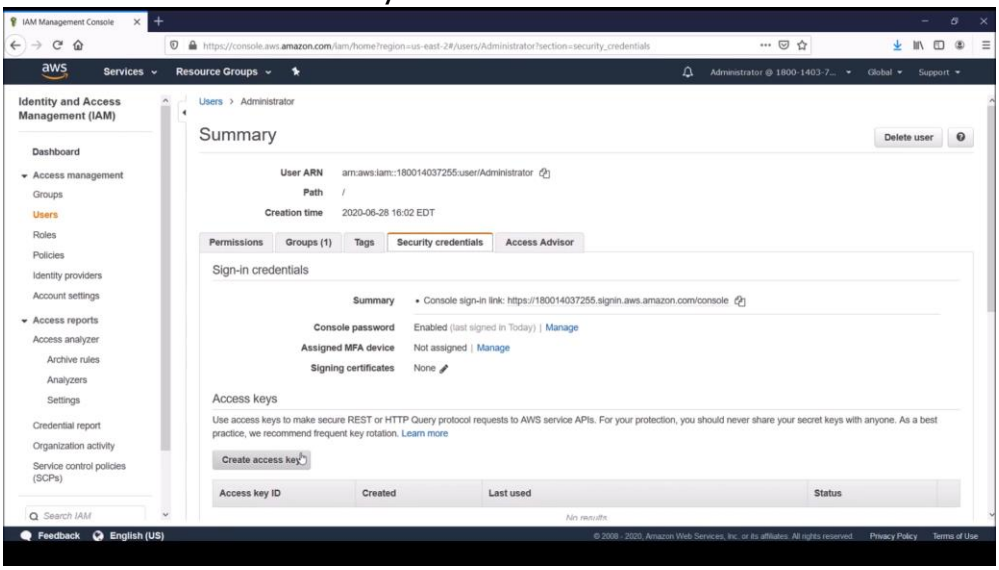
1. On the navigation bar, click on Services, search for IAM.

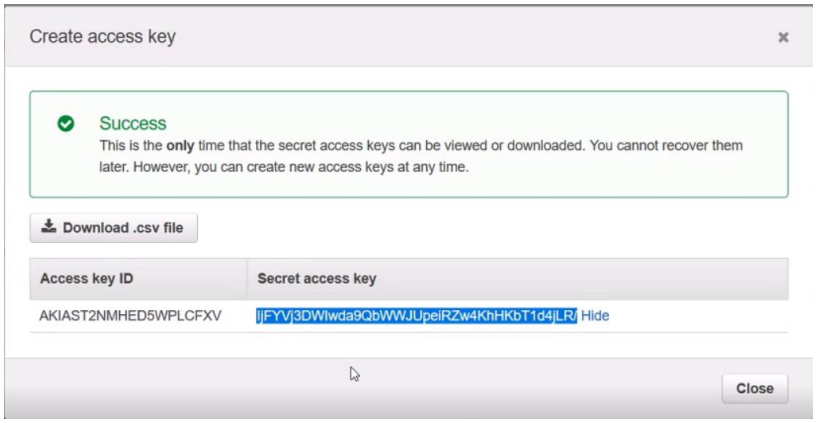


2. On the IAM console, on the left-hand menu, click on Users, and click the appropriate user.



3. On the Summary page, click the Security credentials tab, click Create access key. Take note of the access key.





1. On the Python or Jupyter prompt: pip install boto3

```
1>pip install boto3
```

2. Wait until the service is installed.

```
ollecting boto3
 Downloading https://files.pythonhosted.org/packages/35/d9/f3606e9f4e23132baf44bffb04aac65
    |                                | 133kB 595kB/s
ollecting botocore<1.18.0,>=1.17.12 (from boto3)
 Downloading https://files.pythonhosted.org/packages/99/fa/933d9b4205863bdd24ce151bc4cea72
    |                                | 6.3MB 2.2MB/s
ollecting jmespath<1.0.0,>=0.7.1 (from boto3)
 Downloading https://files.pythonhosted.org/packages/07/cb/5f001272b6faeb23c1c9e0acc04d48e
ollecting s3transfer<0.4.0,>=0.3.0 (from boto3)
 Downloading https://files.pythonhosted.org/packages/69/79/e6afb3d8b0b4e96cefbdc690f741d7d
    |                                | 71kB 1.5MB/s
equirement already satisfied: urllib3<1.26,>=1.20; python_version != "3.4" in c:\users\fbe
equirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\fbello1\appdata\loca
equirement already satisfied: docutils<0.16,>=0.10 in c:\users\fbello1\appdata\local\conti
equirement already satisfied: six>=1.5 in c:\users\fbello1\appdata\local\continuum\anacond
nstalling collected packages: jmespath, botocore, s3transfer, boto3
uccessfully installed boto3-1.14.12 botocore-1.17.12 jmespath-0.10.0 s3transfer-0.3.3
```

References

6 Unique AWS Features - Why Amazon Web Service is Popular
dzone.com

Amazon Comprehend
aws.amazon.com

Amazon Comprehend Customers and Partners
aws.amazon.com

AWS Services List - Top 10 AWS Services
mindmajix.com

Comprehend
boto3.amazonaws.com

Comparison of the Most Useful Text Processing APIs
activewizards.com

Getting Started with Amazon Comprehend
docs.aws.amazon.com

NLP (Natural Language Processing) Tutorial:
What is, History, Example
www.guru99.com

What Is Amazon Comprehend?
docs.aws.amazon.com

What is Amazon S3?
docs.aws.amazon.com

Yelp Reviews CSV
Luis Fred
www.kaggle.com