

Creating a Search Engine with TF-IDF Algorithm for Wikipedia Data using Apache Spark on AWS EMR Studio to Analyze and Rank Relevant Documents.

Naveen Raju Sreerama Raju Govinda Raju
naveenraju100@gmail.com | +1 224 706 7718 | +91 9886157101 |
<https://www.linkedin.com/in/naveen-raju-s-g-bb1486124>
<https://naveenrajusg.github.io/Portfolio/>

Table of Contents

1) Overview

2) Create and configure Amazon EMR Studio

- A) Studio name and Description**
- B) Networking and Security**
- C) Studio service role**
 - I) Create an IAM role**
 - II) Configure Studio Service role**
 - III) Create a S3 bucket**
 - IV) Continue configuring Studio Service Role**

3) Create Amazon EMR Workspace

4) Cluster creation

5) Attach cluster to workspace and launch PySpark Notebook

6) Code TF-IDF PySpark code

7) Deleting Resources

- A) Delete workspace**
- B) Terminate Cluster**
- C) Delete EMR Studio**
- D) Delete S3 buckets**

1) Overview

Apache PySpark:-

Apache PySpark is an open-source distributed computing framework built on top of Apache Spark and designed for processing and analyzing large-scale data sets. It provides a Python API that allows developers to harness the power of Spark's distributed computing capabilities to perform data transformations, data manipulation, and machine learning tasks efficiently and at scale. In our project, we utilized PySpark on EMR Studio to implement a TF-IDF algorithm for preparing and analyzing Wikipedia data, ultimately building a custom search engine for relevant document retrieval.

TF-IDF (Term Frequency - Inverse Document Frequency) :

Term Frequency just measures how often a word occurs in a document. (A word that occurs frequently is probably important to that document's meaning)

Number of times term t appears in document d

Total number of terms in document d

$$\text{Measure of term frequency of term } t \text{ in a document } d = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

Document Frequency is how often a word occurs in an entire set of documents, I.e., all of Wikipedia or every web page. (This tells us about common words that just appear everywhere o matter what the topic, like "a", "the", "and", etc.)

$$\text{Measure of relevancy of a word to a document} = \frac{\text{Term Frequency}}{\log(\text{Document Frequency})}$$

Or

$$\text{Measure of relevancy of a word to a document} = \text{Term Frequency} * \log(\text{Inverse Document Frequency})$$

That is, it takes how often the word appears in a document, over how often it just appears everywhere. That gives you a measure of how important and unique this word is for this document.

Usually log of the IDF is used, since word frequencies are distributed exponentially. Using log gives us a better weighting of a words overall popularity.

The rationale behind using the log transformation for IDF is to dampen the effect of extremely high and low IDF values. It helps to normalize the IDF scores and makes them more manageable. The IDF value of a term increases logarithmically with the number of documents in the collection that contain the term.

The formula for IDF is typically given as:

$$\text{IDF}(t) = \log(N / (\text{DF}(t)))$$

where:

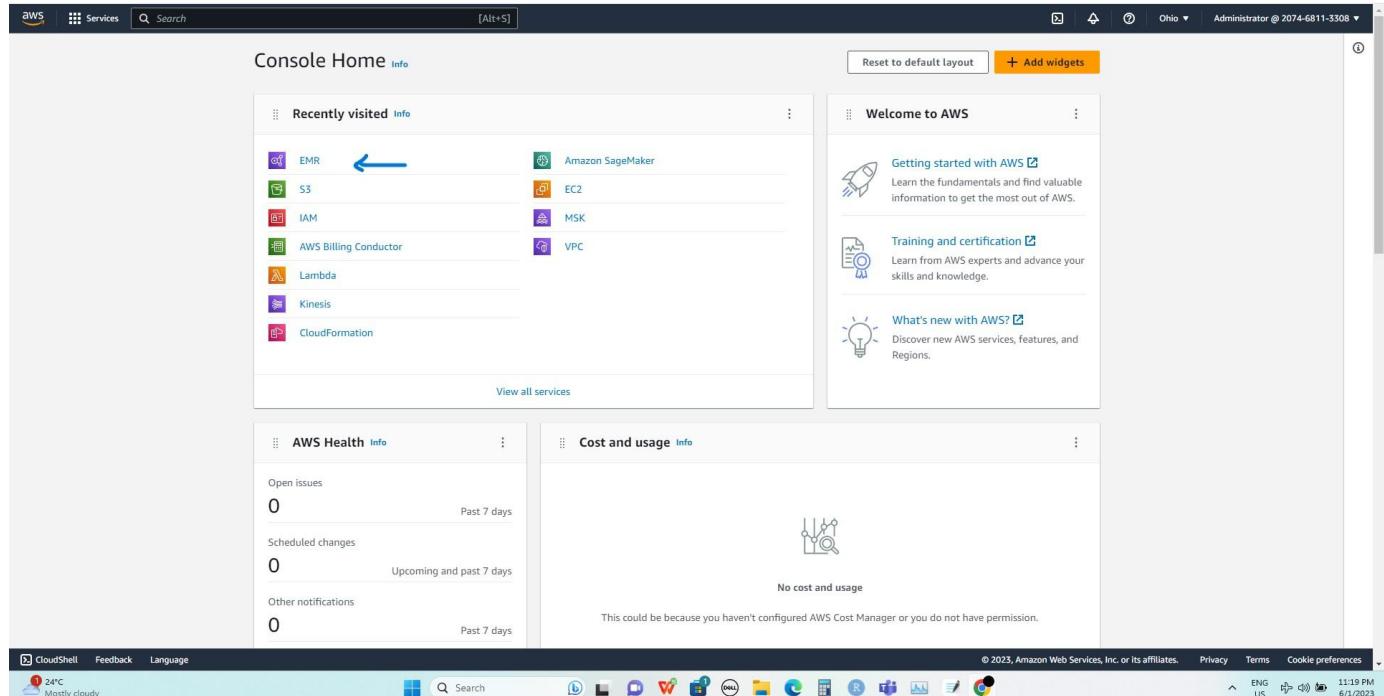
N is the total number of documents in the collection.

DF(t) is the document frequency of term "t," i.e., the number of documents that contain the term.

By taking the logarithm, we can handle a wide range of values and prevent the IDF from being dominated by very large or very small numbers.

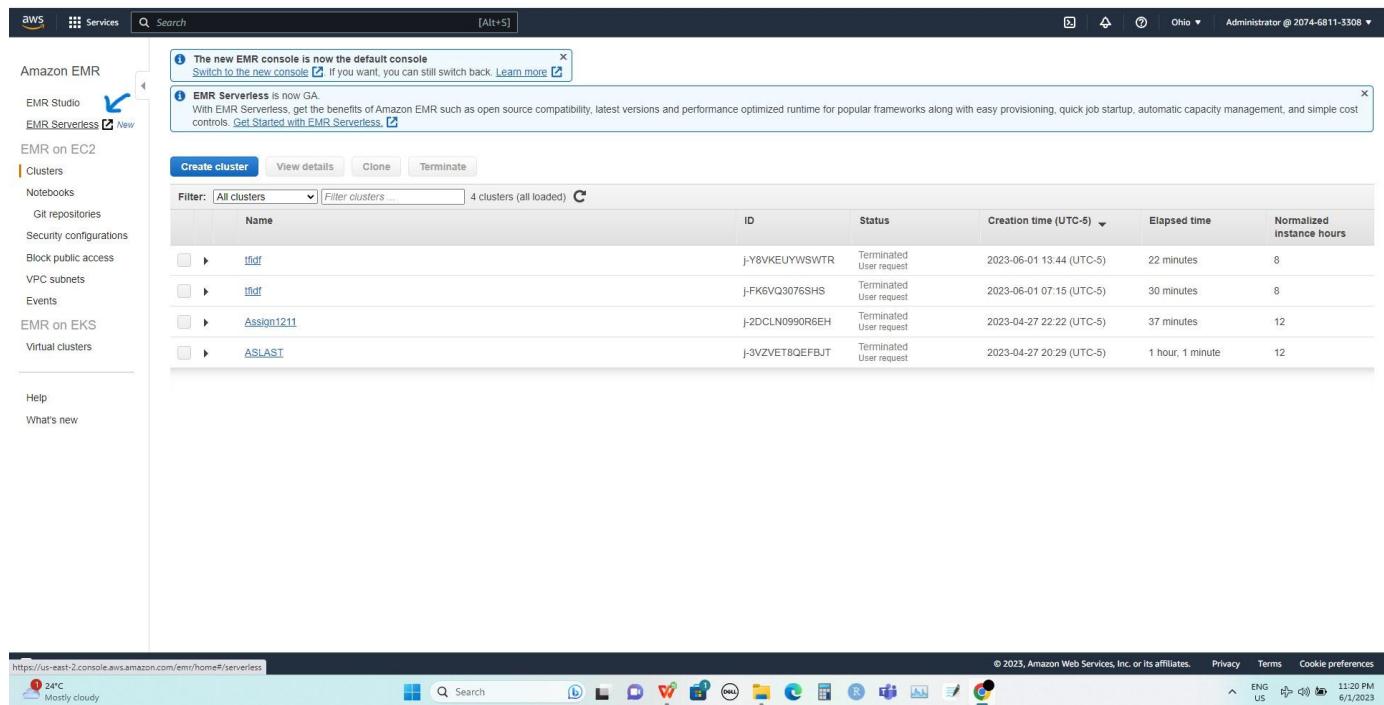
2) Create and configure Amazon EMR Studio

From AWS console navigate to EMR



The screenshot shows the AWS Console Home page. In the top left, under 'Recently visited', the 'EMR' icon is highlighted with a blue arrow pointing to it. Other services listed include S3, IAM, AWS Billing Conductor, Lambda, Kinesis, and CloudFormation. To the right, there's a 'Welcome to AWS' section with links for 'Getting started with AWS', 'Training and certification', and 'What's new with AWS?'. Below these sections are 'AWS Health' and 'Cost and usage' dashboards. At the bottom, the navigation bar includes CloudShell, Feedback, Language, and various system icons.

From AWS EMR console click on EMR serverless



The screenshot shows the AWS EMR console. On the left, a sidebar lists 'Amazon EMR', 'EMR Studio', 'EMR Serverless [New]', 'EMR on EC2', 'Clusters', 'Notebooks', 'Git repositories', 'Security configurations', 'Block public access', 'VPC subnets', 'Events', 'EMR on EKS', and 'Virtual clusters'. The 'EMR Serverless' option is highlighted with a blue arrow. The main content area shows a message about the new EMR console being the default. Below it, a section titled 'EMR Serverless' provides information about its benefits. A table lists four clusters: 'tfdf' (terminated, 22 minutes, 8 hours), 'tfdf' (terminated, 30 minutes, 8 hours), 'Assign1211' (terminated, 37 minutes, 12 hours), and 'ASLAST' (terminated, 1 hour, 1 minute, 12 hours). The table has columns for Name, ID, Status, Creation time (UTC-5), Elapsed time, and Normalized instance hours.

Navigate to EMR Studio - > Studios

The new Amazon EMR console is now the default console. Continue to use the new console, or switch to the old console. Learn more [\[link\]](#)

EMR Notebooks are now available as EMR Studio Workspaces in the new console. You can continue to use EMR Notebooks [\[link\]](#) in the old console until you're ready to use Workspaces. Learn more [\[link\]](#)

Amazon EMR Serverless

Run big data applications without managing clusters and servers

Amazon EMR Serverless is a serverless option in Amazon EMR that makes it easy for data analysts and engineers to run batch jobs and interactive workloads using open-source big data analytics frameworks without configuring, managing, and scaling clusters or servers. You get all the features and benefits of Amazon EMR without the need for experts to plan and manage clusters.

Introduction

Introducing EMR Serverless | Amazon Web Services

Get started

EMR Serverless provides a runtime environment that simplifies running analytics applications using the latest open-source frameworks. Get started with your first application in seconds.

[Get started](#)

What's new

Monitor Amazon EMR Serverless applications in near real-time with CloudWatch metrics

Monitor Amazon EMR Serverless jobs in real-time with native Spark and Hive Tez UI

Documentation

What is Amazon EMR Serverless

Getting started with EMR Serverless

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 11:20 PM 6/1/2023

Click on Create Studio

The new Amazon EMR console is now the default console. Continue to use the new console, or switch to the old console. Learn more [\[link\]](#)

EMR Notebooks are now available as EMR Studio Workspaces. You can access your notebooks as Workspaces and create new Workspaces on the [EMR Studio Workspaces](#). EMR Notebooks users need additional IAM role permissions to access or create Workspaces. You can continue to use EMR Notebooks [\[link\]](#) in the old console until you're ready to use Workspaces. Learn more [\[link\]](#)

Amazon EMR > EMR Studio: Studios

Studio name	Creation time (UTC-05:00)	Authenticated by	Studio Access URL
No Studios			

[Create Studio](#)

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 11:21 PM 6/1/2023

A) Studio name and Description:

Set configuration of a Studio:

Studio name and Description

Studio name : TF-IDF

The new Amazon EMR console is now the default console
Continue to use the new console, or switch to the old console. Learn more

EMR Notebooks are now available as EMR Studio Workspaces in the new console. You can continue to use EMR Notebooks in the old console until you're ready to use Workspaces. Learn more

Amazon EMR > EMR Studio: Studios > Create a Studio

Create a Studio Info

Provide a unique name for your EMR Studio. This will be the name your Studio team sees when they log in. Set the networking and security configuration for the Studio and then assign roles and permissions. After you have created a Studio you will then be able to add users to it and refine permissions.

Studio name and description

Studio name
TF-IDF
Use up to 256 characters (alphanumeric, hyphens, or underscores).

Description
Describe the Studio
256 characters maximum

Tags
No tags associated with the resource.
[Add new tag](#)
You can add 50 more tags.

B) Networking and Security

VPC:

VPC (Choose the VPC that EMR Studio can use to communicate with EMR clusters. From the dropped down select VPC that is usually associated with our AWS account)

Subnets : Choose subnets that EMR Studio can use to communicate with EMR Clusters. From the dropped down select all 3 subnets associated with our account.

Security and access:

Security group act as firewalls with rules that allow network traffic between the EMR cluster and your workspace. You can use default security groups with the minimum required rules

Select Default security group

Default security group:

Select Enable clusters/endpoints and Git repository

Networking and security

VPC Info
Choose the VPC that EMR Studio can use to communicate with EMR clusters. Make sure the VPC is tagged with key `for-use-with-amazon-emr-managed-policies` and value `true`. To manage tags, use [VPC Dashboard](#).

Subnets Info
Choose subnets that EMR Studio can use to communicate with EMR Clusters. Make sure each subnet is tagged with key `for-use-with-amazon-emr-managed-policies` and value `true`. To manage tags, use [VPC Dashboard](#).

Security and access Info
Security groups act as firewalls with rules that allow network traffic between the EMR cluster and your workspace. You can use default security groups with the minimum required rules or specify your own security groups.

Default security group
Select between enabling the user to only use the default EMR cluster or endpoint, or opt to include the ability to edit GIT repositories.

Custom security group
Select from security groups for cluster or endpoints, or select a security group for the studio workspace.

Default security group
 Enable clusters/endpoints and Git repository
 Enable clusters/endpoints

Studio service role Info

CloudShell Feedback Language 24°C Mostly cloudy © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 11:22 PM 6/1/2023

C) Studio service role

Authentication: AWS Identity and Access Management (IAM)

Service role:

The service role defines the allowable actions for EMR Studio when provisioning resources. Examples of such actions included attaching a workspace to a cluster or accessing the S3 backup location for workspace.

I) Create an IAM role

From AWS console navigate to Identity and Access Management (IAM) - > Roles - > Create role

Step A - Select trusted entity:

Trusted entity type : AWS service (Allows AWS services like EC2, Lambda, or others to perform actions in this account)

Use case : EMR (Allows EC2 instance in an Elastic MapReduce cluster to call AWS services such as S3 on your behalf.)

The screenshot shows the 'Create role' wizard in the AWS IAM console. The current step is 'Step 1: Select trusted entity'. In the 'Trusted entity type' section, the 'AWS service' option is selected, with a note: 'Allow AWS services like EC2, Lambda, or others to perform actions in this account.' Below it, there are other options: 'AWS account' (allow entities in other AWS accounts), 'Web identity' (allow users federated by an external provider), 'SAML 2.0 federation' (allow users federated via SAML 2.0), and 'Custom trust policy' (create a custom trust policy). In the 'Use case' section, 'EMR' is selected, with a note: 'Allows EC2 instances in an Elastic MapReduce cluster to call AWS services such as S3 on your behalf.' Other options include 'EC2' and 'Lambda'. At the bottom, there's a dropdown for 'Use cases for other AWS services' which also lists 'EMR', 'EMR Role for EC2', and 'EMR - Cleanup'. The 'Next Step' button is visible at the bottom right.

Step B - Add permissions

Permission Policies:

Policy name - AmazonElasticMapReduceRole (would have been auto selected)

The screenshot shows the 'Create role' wizard in the AWS IAM console, Step 2: 'Add permissions'. In the 'Permissions policies' section, there is one item listed: 'AmazonElasticMapReduceRole' (AWS managed). The 'Type' column shows 'AWS managed' and the 'Attached entities' column shows '1'. Below this, there's a note about setting a 'permissions boundary': 'Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.' The 'Next Step' button is visible at the bottom right.

Step C - Name, review and create

Role name - EMR_Notebooks_DefaultRoles

Click on Create role.

The screenshot shows the 'Name, review, and create' step of the IAM role creation wizard. It includes three tabs: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The 'Name, review, and create' tab is active. The 'Role details' section shows the role name 'EMR_Notebooks_DefaultRoles'. The 'Description' section contains the text: 'Allows Elastic MapReduce to call AWS services such as EC2 on your behalf.' The 'Step 1: Select trusted entities' section displays a JSON policy document:

```
1  {  
2      "Version": "2012-10-17",  
3      "Statement": [  
4          {  
5              "Effect": "Allow",  
6              "Action": [  
7                  "sts:AssumeRole"  
8              ],  
9              "Principal": [  
10                 {  
11                     "Service": [  
12                         "elasticmapreduce.amazonaws.com"  
13                     ]  
14                 }  
15             ]  
16         }  
17     ]  
18 }
```

The 'Step 2: Add permissions' section shows a summary of attached policies, listing 'AmazonElasticMapReduceRole' as an 'AWS managed' policy attached as a 'Permissions policy'. The 'Tags' section indicates no tags are associated with the resource. At the bottom right, there are 'Cancel', 'Previous', and 'Create role' buttons.

Navigate to IAM - > Roles - > EMR_Notebooks_DefaultRoles

The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with navigation links like Dashboard, Access management, Access reports, and Related consoles. The main area displays a table of roles. A blue arrow highlights the transition from the first role entry to the second one.

Role name	Trusted entities	Last activity
AWSGlueServiceRole-MachineLearning	AWS Service: glue	13 days ago
AWSServiceRoleForAmazonSageMakerNotebooks	AWS Service: sagemaker (Service-Linked Role)	39 days ago
AWSServiceRoleForEMRCleanup	AWS Service: elasticmapreduce (Service-Linked Role)	7 hours ago
AWSServiceRoleForResourceExplorer	AWS Service: resource-explorer-2 (Service-Linked Role)	27 minutes ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	12 days ago
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
DemoGlueETLRole	AWS Service: glue	13 days ago
EMR_DefaultRole	AWS Service: elasticmapreduce	9 hours ago
EMR_EC2_DefaultRole	AWS Service: ec2	9 hours ago
EMR_Notebooks_DefaultRole	AWS Service: elasticmapreduce	-
EMR_Notebooks_DefaultRoles	AWS Service: elasticmapreduce	-
kinesis-analytics-ticker-analytics-ap-south-1	AWS Service: kinesisanalytics	13 days ago
KinesisFirehoseServiceRole-ml-raju-ap-south-1-1684398559108	AWS Service: firehose	14 days ago
KinesisFirehoseServiceRole-ml-raju-ap-south-1-1684398905442	AWS Service: firehose	14 days ago
KinesisFirehoseServiceRole-ml-raju-ap-south-1-1684496931324	AWS Service: firehose	13 days ago
KinesisFirehoseServiceRole-PUT-S3-uecXX-ap-south-1-1684389951752	AWS Service: firehose	14 days ago

Click on attach policies

The screenshot shows the AWS IAM Role Summary page for 'EMR_Notebooks_DefaultRoles'. The sidebar includes links for CloudShell, Feedback, Language, and various AWS services. The main area shows the role's summary, including creation date, last activity, ARN, and maximum session duration. The 'Permissions' tab is active, displaying a table of attached policies. A dropdown menu is open over the 'Add permissions' button, with 'Attach policies' highlighted.

Policy name	Type	Description
AmazonElasticMapReduceRole	AWS managed	This policy is on a deprecation path. See documentation for guidance: https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-managed-iam-policies.html .

Visit the following web page : <https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-managed-notebooks-service-role.html> to fetch JSON's related to AmazonElasticMapReduceEditorsRole.

Copy the Json related to AmazonElasticMapReduceEditorsRole:

The screenshot shows the AWS Documentation interface for the Amazon EMR Management Guide. The main content area displays the JSON code for the `AmazonElasticMapReduceEditorsRole`. The JSON is as follows:

```
{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "ec2:AuthorizeSecurityGroupEgress", "ec2:AuthorizeSecurityGroupIngress", "ec2>CreateSecurityGroup", "ec2:DescribeSecurityGroups", "ec2:RevokeSecurityGroupEgress", "ec2>CreateNetworkInterface", "ec2>CreateNetworkInterfacePermission", "ec2>DeleteNetworkInterface", "ec2>DeleteNetworkInterfacePermission", "ec2:DescribeNetworkInterfaces", "ec2:ModifyNetworkInterfaceAttribute", "ec2:DescribeTags", "ec2:DescribeInstances", "ec2:DescribeSubnets", "ec2:DescribeVpcs", "elasticmapreduce>ListInstances", "elasticmapreduce>DescribeCluster", "elasticmapreduce>ListSteps" ], "Resource": "*" } ] }
```

The JSON is highlighted with a blue selection bar. On the right side of the page, there is a sidebar titled "On this page" with links to "EMR Notebooks service role permissions" and "EMR Notebooks updates to AWS managed policies". The bottom of the screen shows a Windows taskbar with various icons and the system clock indicating 11:29 PM on 6/1/2023.

Continue in Attache policies web page:

Paste the AmazonElasticMapReduceEditorsRole Json and click on Review Policy

The screenshot shows the "Create policy" page in the AWS IAM console. The "Visual editor" tab is selected, showing a large text area containing the JSON code for the `AmazonElasticMapReduceEditorsRole`. The JSON code is identical to the one shown in the previous screenshot. At the bottom of the text area, there are status indicators: Security: 0, Errors: 0, Warnings: 0, and Suggestions: 0. To the right of the text area, there is a "Review policy" button highlighted with a blue arrow. The bottom of the screen shows a Windows taskbar with various icons and the system clock indicating 11:29 PM on 6/1/2023.

Name : AmazonElasticMapReduceEditorsRole

The screenshot shows the 'Create policy' page in the AWS IAM console. The 'Name' field is highlighted with a blue arrow. The table below shows permissions for EC2 and EMR services.

Service	Access level	Resource	Request condition
Allow (2 of 377 services) Show remaining 375			
EC2	Limited: List, Write, Permissions management, Tagging	Multiple	Multiple
EMR	Limited: Read	All resources	None

Again click on Attach Policy:

The screenshot shows the 'EMR_Notebooks_DefaultRoles' role page in the AWS IAM console. The 'Permissions' tab is selected. The 'Add permissions' button is highlighted with a blue arrow. The table lists attached policies.

Policy name	Type	Description
AmazonElasticMapReduceRole	AWS managed	This policy is on a deprecation path. Se...
AmazonElasticMapReduceEditorsRole	Customer inline	

Visit the following web page : <https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-managed-notebooks-service-role.html> to fetch JSON's related to S3FullAccessPolicy. Copy the JSON.

The screenshot shows a sidebar on the left with navigation links for Amazon EMR Studio, EMR Notebooks, Plan and configure clusters, Security (selected), Use security configurations to set up cluster security, Data protection, IAM with Amazon EMR (with sub-links for How Amazon EMR works with IAM, Runtime roles for Amazon EMR steps), and Configure service roles for Amazon EMR (with sub-links for Service roles used by Amazon EMR). The main content area displays the JSON code for the S3FullAccessPolicy:

```
{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "s3:*", "Resource": "*" } ] }
```

Below the JSON, a note states: "You can scope down read and write access for your service role to the Amazon S3 location where you want to save your notebook files. Use the following minimum set of Amazon S3 permissions."

Continue in Attache policies web page:
Paste the S3FullAccessPolicy Json and click on Review Policy

The screenshot shows the AWS CloudShell interface. At the top, it says "Create policy". Below that, a note says: "A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more". There are two tabs: "Visual editor" (selected) and "JSON". The JSON tab shows the same policy JSON as the previous screenshot. At the bottom, there are status indicators: "Security: 0", "Errors: 0", "Warnings: 0", and "Suggestions: 0". Below the editor, a message says: "Character count: 938 of 10,240. The current character count includes character for all inline policies in the role: EMR_Notebooks_DefaultRole." On the right, there are "Cancel" and "Review policy" buttons. The status bar at the bottom indicates: "CloudShell Feedback Language 26°C Partly sunny Search © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 11:32 PM 6/1/2023".

The screenshot shows the AWS CloudShell interface. At the top, it says "Create policy". Below that, a note says: "Before you create this policy, provide the required information and review this policy." A "Name*" input field contains "S3FullAccessPolicy". A note below it says: "Maximum 128 characters. Use alphanumeric and '+-, @-' characters." Below the name field is a "Summary" section with a table:

Service	Access level	Resource	Request condition
Allow (1 of 377 services)	Show remaining 376		
S3	Full access	All resources	None

At the bottom, there are "Cancel", "Previous", and "Create policy" buttons. The status bar at the bottom indicates: "CloudShell Feedback Language 26°C Partly sunny Search © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 11:32 PM 6/1/2023".

Identity and Access Management (IAM)

EMR_Notebooks_DefaultRoles

Allows Elastic MapReduce to call AWS services such as EC2 on your behalf.

Summary

Creation date: June 01, 2023, 23:26 (UTC-05:00)

Last activity: None

ARN: arn:aws:iam::207468113308:role/EMR_Notebooks_DefaultRoles

Maximum session duration: 1 hour

Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

Permissions policies (3) Info

You can attach up to 10 managed policies.

Policy name	Type	Description
AmazonElasticMapReduceRole	AWS managed	This policy is on a deprecation path. Se...
AmazonElasticMapReduceEditorsRole	Customer inline	-
S3FullAccessPolicy	Customer inline	-

Permissions boundary - (not set) Info

Select a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 26°C Partly sunny ENG US 11:32 PM 6/1/2023

II) Configure Studio Service role

Service role : EMR_Notebooks_DefaultRoles

III) Create a S3 bucket

Create S3 bucket named “bucket-emr-123”

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more [\[?\]](#)

General configuration

Bucket name: Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming [\[?\]](#)

AWS Region: US East (Ohio) us-east-2

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

Object Ownership Info

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

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

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

Object Ownership: Bucket owner enforced

Block Public Access settings for this bucket

Block public access

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 26°C Partly sunny ENG US 11:38 PM 6/1/2023

The screenshot shows the AWS S3 console. A green banner at the top indicates that a bucket named "bucket-emr-1234" has been successfully created. Below the banner, the "Buckets" section displays two buckets: "aws-emr-resources-asia" and "bucket-emr-1234". The "bucket-emr-1234" row includes a "Create bucket" button.

IV) Continue configuring Studio Service Role

Select S3 bucket create in last step : s3://bucket-emr-1234

Click on Create Studio

The screenshot shows the "Create Studio" configuration page. In the "Service role" section, the "EMR_Notebooks_DefaultRoles" service role is selected. Under "Workspace storage", the S3 bucket "s3://bucket-emr-1234" is specified. At the bottom right, the "Create Studio" button is highlighted.

3) Create Amazon EMR Workspace

On Amazon EMR console click on “Workspaces(Notebooks)”

The new Amazon EMR console is now the default console. Continue to use the new console, or switch to the old console. [Learn more](#)

EMR Notebooks are now available as EMR Studio Workspaces. You can access your notebooks as Workspaces and create new Workspaces on the EMR Studio Workspaces. EMR Notebooks users need additional IAM role permissions to access or create Workspaces. You can continue to use EMR Notebooks in the old console until you're ready to use Workspaces. [Learn more](#)

Studio TF-IDF created successfully

Studios (1) Info

Studio name	Creation time (UTC-05:00)	Authenticated by	Studio Access URL
TF-IDF	June 01, 2023, 23:39	IAM	https://es-BQOFWYHFCBQT30LGNY1...

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 11:40 PM 6/1/2023

Click on “Create Workspace”
 Basically, we create a workspace and then create notebooks inside it.
 Choose a Studio in which to create the Workspace : TF-IDF

Discover Workspaces

Your EMR Notebooks are called EMR Studio Workspaces in the new console.

- Create and organize your Workspaces in secure Studios.
- Collaborate with other members of your team to write and run notebook code.
- Run Jupyter and JupyterLab notebooks in a single Workspace.
- Connect to Git repos from the Git tab in the left sidebar of JupyterLab.
- Browse data with SQL Explorer.
- Provision EMR clusters with Service Catalog.

Find and launch Workspaces

To access the notebook editor, launch a Workspace.

- Refresh the Workspaces table to see any recently created EMR Notebooks from the old console.
- Select your Workspace, then select Launch Workspace to open it in a new tab.
- You can launch your Workspace with customized options, or quickly launch it with default settings.

Attach clusters to Workspaces

Attach a Workspace to an Amazon EMR compute cluster in the console or in JupyterLab.

- Attach in the Amazon EMR console:
 - Select your Workspace, then select Launch Workspace > Launch with options.
 - Choose an EMR cluster to attach to your Workspace.
- Attach in JupyterLab:
 - Select your Workspace, then select Launch Workspace > Quick launch.
 - Inside JupyterLab, open the Cluster tab in the left sidebar.
 - Choose either an EMR on EC2 or EMR on EKS cluster type, find the cluster in the dropdown, then attach it to your Workspace.

Workspaces (Notebooks) (0) Info

EMR Notebooks are now EMR Studio Workspaces. You can organize and run interactive notebooks in Workspaces.

All Studios	Find Workspaces by name, Studio, status, or last modified by					
Workshop name	Studio name	Status	Cluster ID	Creation time (UTC-05:00)	Last modified by	Last modified (UTC-05:00)

No Workspaces

To create and use Workspaces, you'll first need to Create Studio.

Create Workspace

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 11:40 PM 6/1/2023

Create Workspace

Choose a Studio in which to create the Workspace.

TF-IDF

Create Workspace

Workspace details:

Workspace name : TF-IDF

S3 location (place where workspace and notebooks will be saved) : s3://bucket-emr-1234

The screenshot shows the 'Create a Workspace' interface in EMR Studio. The 'Workspace details' section contains fields for 'Workspace name' (TF-IDF), 'Description - optional' (Describe the Workspace), 'S3 location' (s3://bucket-emr-1234), and 'Workspace collaboration' (checkbox for Allow Workspace collaboration). The 'Network configurations' section shows VPC and subnets information. The status bar at the bottom indicates it's 26°C, Partly sunny, and the date is 6/1/2023.

Click on Create Workspace

The screenshot shows the 'Create a Workspace' interface in EMR Studio. The 'Workspace details' section contains fields for 'Workspace name' (TF-IDF), 'Description - optional' (Describe the Workspace), 'S3 location' (s3://bucket-emr-1234), and 'Workspace collaboration' (checkbox for Allow Workspace collaboration). The 'Network configurations' section shows VPC and subnets information. A note at the bottom says 'After creating a Workspace, you can launch it to open up Jupyterlab. You can then attach it to an EMR compute cluster from the Cluster tab in the left sidebar within Jupyterlab.' The 'Create Workspace' button is highlighted in orange. The status bar at the bottom indicates it's 26°C, Partly sunny, and the date is 6/1/2023.

4) Cluster creation

To run code in notebooks, we must first attach workspace to cluster. All notebooks in the workspace share the same cluster. We can choose to attach a workspace to a cluster on EC2 or an EKS (virtual) cluster.

In the first there will be no EC2 cluster. So under Advance Configuration drop down we should create one.

Cluster name : tfidf

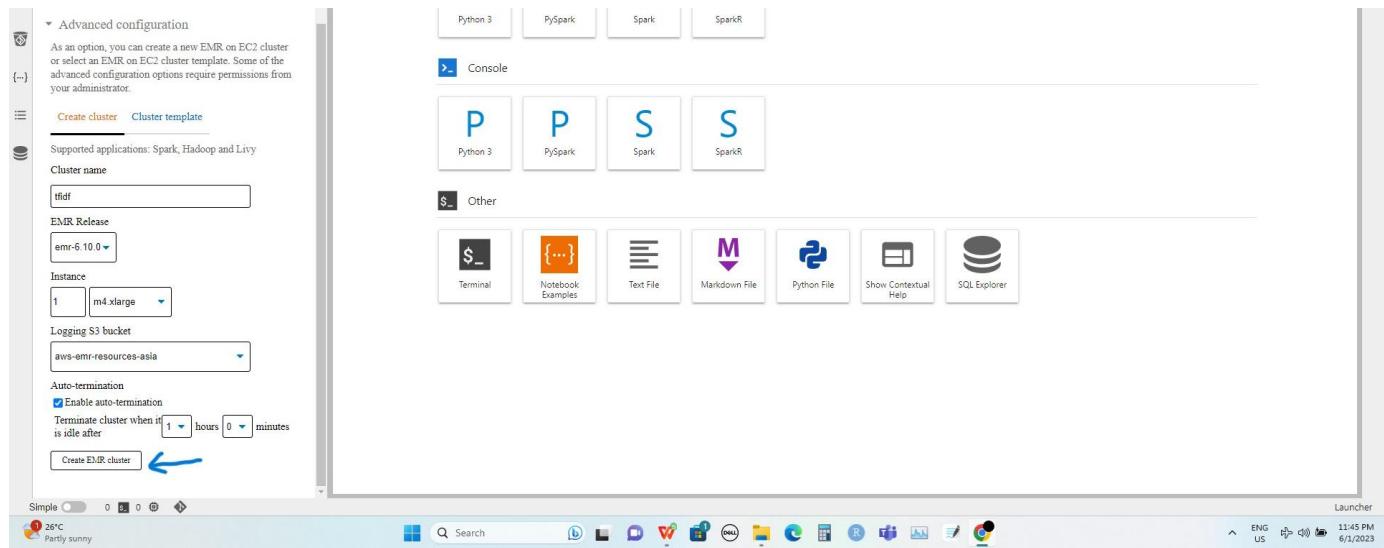
EMR release : emr- 6.10.0

Instance : 1, m5.xlarge

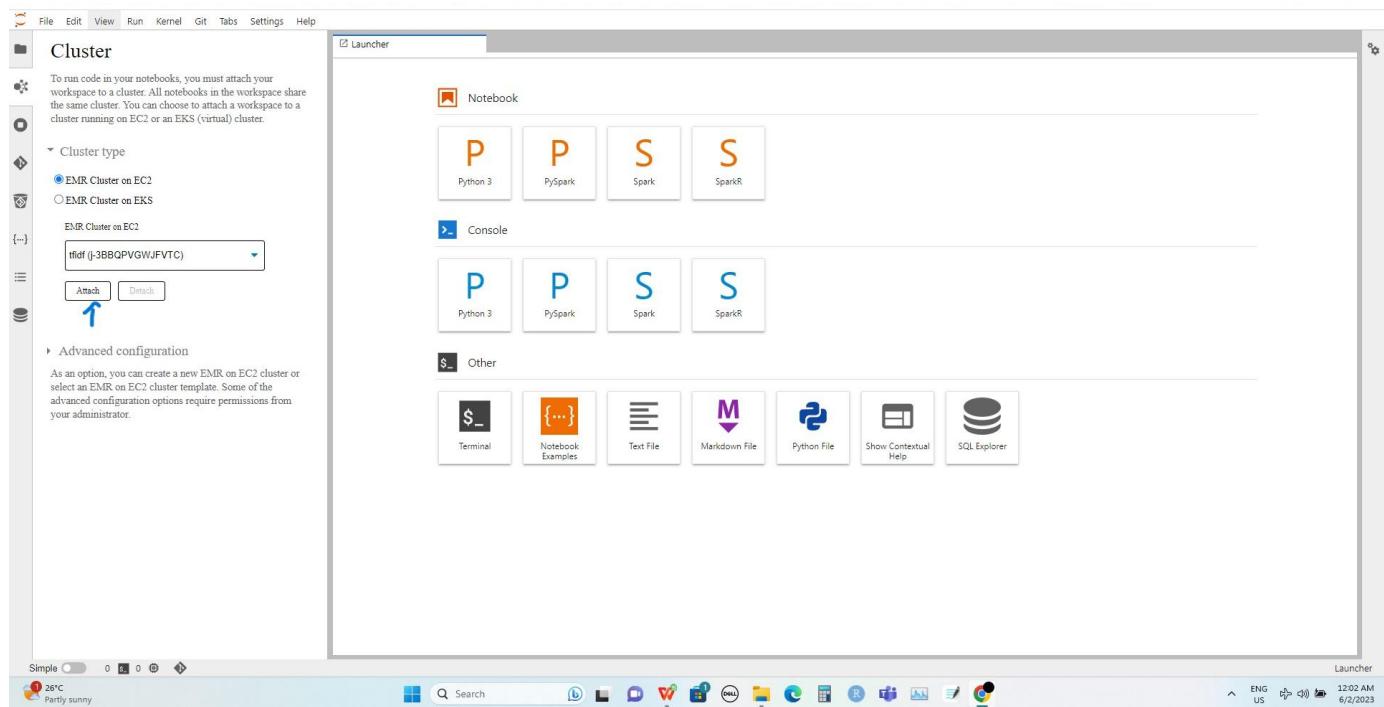
Logging S3 bucket : aws-emr-resources-asia

Auto-termination : Enable auto-termination (Terminate cluster when it idle for 1 hours)

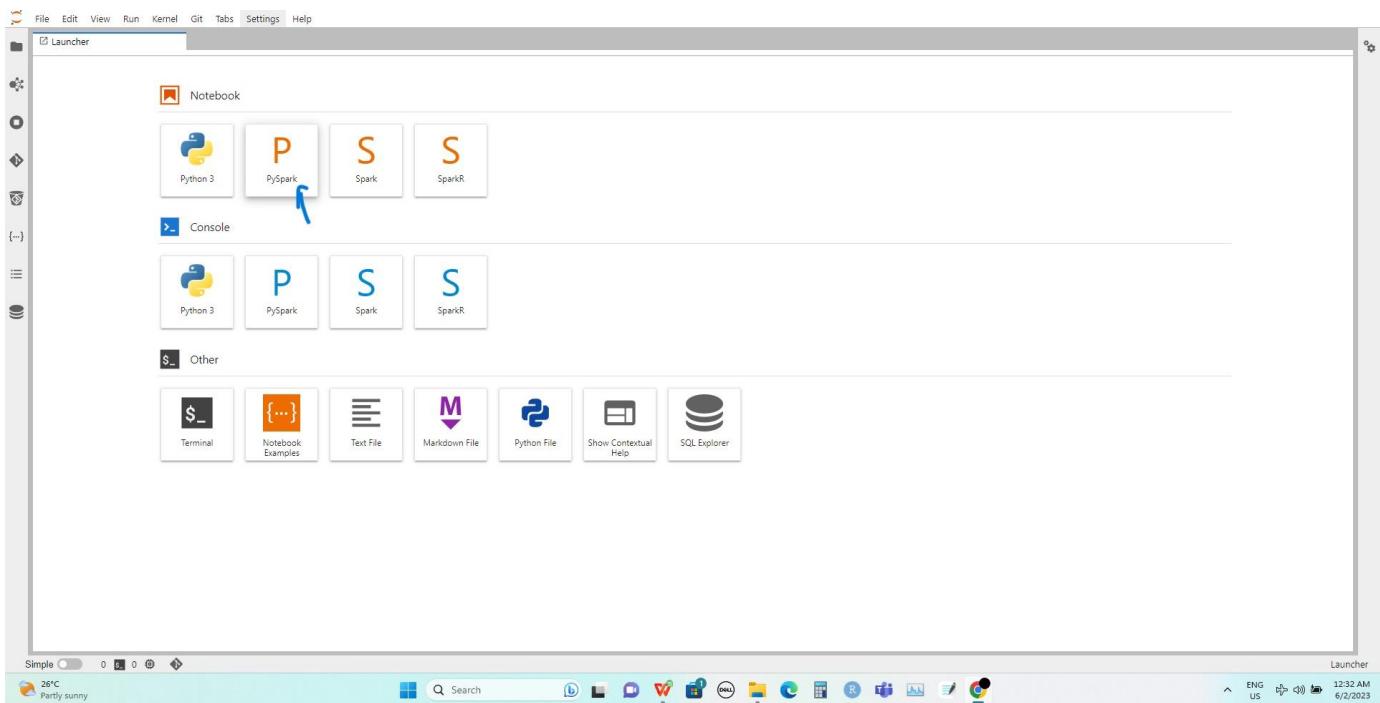
Click on Create EMR cluster



5) Attach cluster to workspace and launch PySpark Notebook



Choose PySpark as launcher:

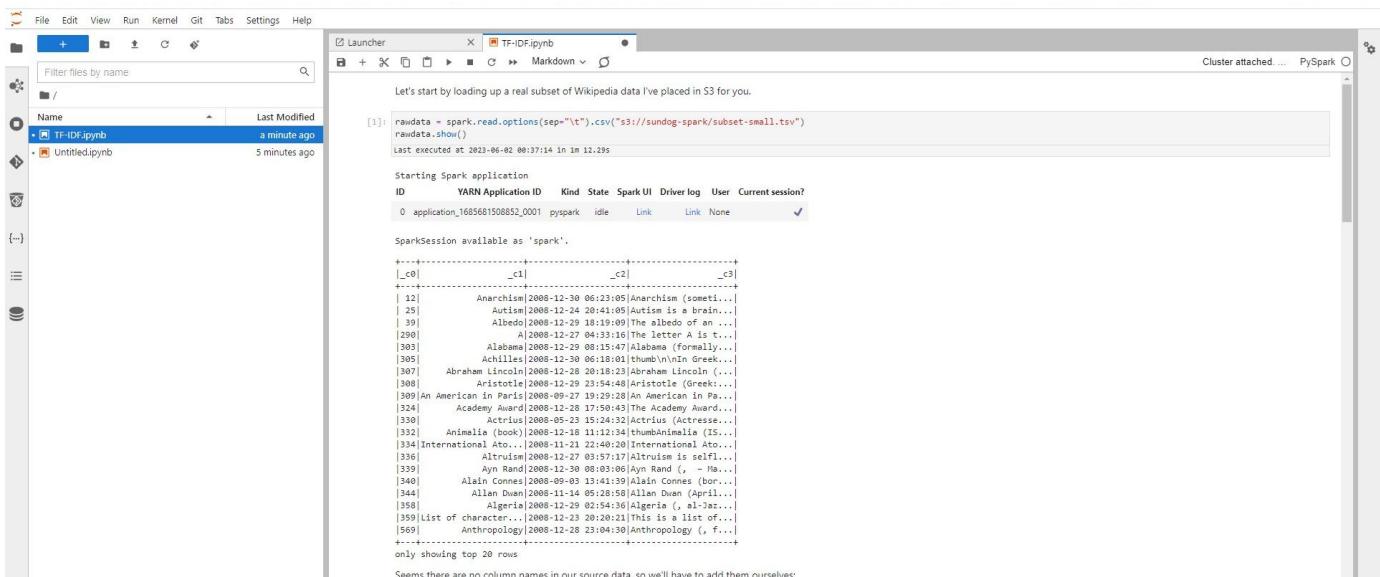


6) Code TF-IDF PySpark code

Code TF-IDF in TF-IDF.ipynb:

```
rawdata = spark.read.options(sep="\t").csv("s3://sundog-spark/subset-small.tsv")
rawdata.show()
```

Explanation: Loading a real subset of Wikipedia data from public S3 bucket and print the dataset.



```
articles = rawdata.toDF("ID", "Title", "Time", "Document")
articles.show()
```

Explanation: Seems there are no column names in our source data, so we will add them.

The screenshot shows a Jupyter Notebook interface. The code cell at the top contains:
[2]:
articles = rawdata.toDF("ID", "Title", "Time", "Document")
articles.show()
Last executed at 2023-06-02 00:37:15 in 1.33s

A "Spark Job Progress" progress bar is shown below the code cell, indicating the status of the data loading and displaying process.

ID	Title	Time	Document
10	Anarchism	2008-12-30 06:23:05	Anarchism (someti...
12	Anarchism	2008-12-24 20:41:05	Autism is a brain...
25	Autism	2008-12-24 20:41:05	Autism is a brain...
30	Albert Einstein	2008-12-27 04:33:16	The letter A is t...
300	A [letter]	2008-12-27 04:33:16	The letter A is t...
303	Alabama	2008-12-29 08:15:47	[Alabama (formally...
305	Achilles	2008-12-30 06:18:01	[thumb]\nIn Greek...
307	Abraham Lincoln	2008-12-28 20:18:23	[Abraham Lincoln (...)
308	Aristotle	2008-12-29 23:54:48	[Aristotle (Greek:...
309	An American In Paris	2008-09-27 19:29:28	An American in Pa...
324	Academy Award	2008-09-27 19:29:28	Academy Award...
330	Actor [book]	2008-05-23 15:24:32	Actor (Atticus F...
332	Animals (book)	2008-12-18 11:12:34	[thumb]Animals (Si...
334	International Atto...	2008-11-22 22:40:20	International Atto...
336	Altruism	2008-12-27 03:57:17	Altruism is self...
339	Ayn Rand	2008-12-30 08:03:06	[Ayn Rand (, - Ma...
340	Alain Connes	2008-09-03 13:41:39	[Alain Connes (bor...
344	Allison Duan	2008-11-14 05:28:58	[Allison Duan (April...
358	Al-Jazeera	2008-12-23 20:20:21	[Al-Jazeera (, al-Jaz...
359	List of character...	2008-12-23 20:20:21	This is a list of c...
569	Anthropology	2008-12-28 23:04:30	[Anthropology (, f...

only showing top 20 rows

```
articles.filter(articles.Document.isNull()).count()
```

Explanation: We should "clean" our data because TF/IDF can't handle null documents, so we should check for Null.

The screenshot shows a Jupyter Notebook interface. The code cell at the top contains:
[3]:
articles.filter(articles.Document.isNull()).count()
Last executed at 2023-06-02 00:38:33 in 17.50s

A "Spark Job Progress" progress bar is shown below the code cell, indicating the status of the filtering process.

1

```
cleanedArticles = articles.filter(articles.Document.isNotNull())
cleanedArticles.filter(articles.Document.isNull()).count()
```

Explanation: There is one null document. So we will remove it. And then count the number of Null records.

The screenshot shows a Jupyter Notebook interface. The code cell at the top contains:
[4]:
cleanedArticles = articles.filter(articles.Document.isNotNull())
cleanedArticles.filter(articles.Document.isNull()).count()
Last executed at 2023-06-02 00:39:02 in 1.34s

A "Spark Job Progress" progress bar is shown below the code cell, indicating the status of the filtering process.

0

Looks like there is one null document. As there is only one and it's clearly corrupt when we look into it, we can just remove it and call it a day.

```
from pyspark.ml.feature import HashingTF, IDF, Tokenizer
```

```
tokenizer= Tokenizer(inputCol="Document", outputCol="words")
```

Explanation:

TF/IDF wants numbers, not words. So now we need to pre-process our data before we can run any fun algorithms on it. We'll first tokenize the articles to split them up into words, and store them in a sparse vector that is now a numeric representation of the words in each article.

"Tokenizer" is the class being instantiated.

"Tokenizer" class from Apache Spark's MLlib library. It is used for tokenizing text documents into individual words.

"inputCol="Document" specifies the name of the input column in the DataFrame that contains the text documents to be tokenized. In this case, the input column is expected to be named "Document"

"outputCol =words" specifies the name of the output column in the DataFrame where the tokenized words will be stored. In this case, the output column will be named "words"

```
wordsData = tokenizer.transform(cleanedArticles)
```

Explanation:

Once we have defined 'Tokenizer' instance you can apply it to a DataFrame using the 'transform()' method.

This will create a new DataFrame called 'tokenizedData' that contains the original columns from 'dataframe', as well as an additional column called 'words' that contain the tokenize words.

```
[5]: from pyspark.ml.feature import HashingTF, IDF, Tokenizer  
tokenizer= Tokenizer(inputCol="Document", outputCol="words")  
wordsData = tokenizer.transform(cleanedArticles)  
Last executed at 2023-06-02 00:39:56 in 796ms
```

```
hashingTF = HashingTF(inputCol="words", outputCol="rawFeatures")  
featurizedData = hashingTF.transform(wordsData)  
featurizedData.show()
```

Explanation:

HashingTF is a class from Apache Spark's MLIB library, it is a feature extraction method that converts a collection of text documents into a numerical feature vector representation.

"HashingTF" is the class being instantiated.

"inputCol="words" specifies the name of the input column in the DataFrame that contains the tokenized words. In our case, it seems that the previous step produced a column named "words" containing the tokenized words.

"outputCol="rawFeatures" specifies the name of the output column in the DataFrame where the feature vectors will be stored. In this case, the output column will be named "rawFeatures".

Once we have defined the "Hashing TF instance, we can apply it to a DataFrame using the transform() method, similar to how it was done with the 'Tokenizer'. The input DataFrame 'wordsData' will be transformed into a new DataFrame 'featurizedData', which will contain the original columns along with an additional column called "rawFeatures" that contains the feature vectors generated by the "HashingTF" transformation. "HashingTF" method uses the hashing trick to map tokens to numerical indices. The number of features in the output vector is determined by the 'numFeatures' parameter, which has a default value of $2^{18} = 262,144$. We can adjust this parameter when instantiating the "HashingTF" object if needed.

Last Modified

Name	Last Modified
TF-IDF.ipynb	seconds ago
Untitled.ipynb	15 minutes ago

[7]: hashingTF = HashingTF(inputCol="words", outputCol="rawFeatures")
featureizedData = hashingTF.transform(wordsData)
featureizedData.show()
Last executed at 2023-06-02 00:47:26 in 15.41s

Spark Job Progress

ID	Title	Time	Document	words	rawFeatures
12	Anarchism	2008-12-30 06:13:05	Anarchism (somet...)	[anarchism, (some...)]	{262144,[116,120,...]
25	Autism	2008-12-24 20:41:05	Autism is a brain...	[autism, is, a, b...]	{262144,[521,1546,...]
39	Albedo	2008-12-29 18:19:09	The albedo of a...	[the, albedo, of,...]	{262144,[1625,179,...]
298	A	2008-12-27 04:33:16	The letter A is t...	[the, letter, a,...]	{262144,[5303,603,...]
303	Albion	2008-12-29 05:18:01	Albion (albion, i...	[albion, (albion, i...)]	{262144,[105,115,...]
305	Achilles	2008-12-29 05:18:01	Thutmohirin Greek...	[thutmohirin, gree...]	{262144,[385,991,...]
307	Abraham	2008-12-28 20:18:23	Abraham Lincoln...	[abraham, lincoln...]	{262144,[115,440,...]
308	Aristotle	2008-12-29 23:54:48	Aristotle (greek...)	[aristotle (greek...)]	{262144,[154,619,...]
309	An American in Paris	2008-09-27 19:29:28	An American in Pa...	[an, american, in,...]	{262144,[2366,313,...]
324	Academy Award	2008-12-28 17:50:43	The Academy Award...[the, academy, aw...]	[the, academy, aw...]	{262144,[161,521,...]
330	Actrius	2008-09-21 15:25:42	Actrius (actresse...)	[actrius, (actres...)]	{262144,[6558,674,...]
332	Animals (book)	2008-12-18 11:12:34	Thumbeanimals (...	[thumbeanimals, (...)]	{262144,[2284,609,...]
334	International Accounting Standard	2008-12-27 03:57:57	International, acco...	[international, acco...]	{262144,[1447,155,...]
336	Albert Einstein	2008-12-27 03:57:57	Albert Einstein is so...[albert, einstein, is, so...]	[albert, einstein, is, so...]	{262144,[501,1272,...]
339	Ayn Rand	2008-12-30 08:03:06	Ayn Rand (- He...)	[-, he...]	{262144,[116,365,...]
340	Alain Connes	2008-09-03 13:41:39	Alain Connes (bor...)	[alain, connes, (...)]	{262144,[154,1595,...]
344	Allan Dwan	2008-11-14 05:28:58	Allan Dwan (ap...)	[allan, dwan, (ap...)]	{262144,[1578,181,...]
358	Algeria	2008-12-29 02:54:36	Algeria (, al-Jaz...)	[algeria, (, al-jaz...)]	{262144,[161,369,...]
359	List of characters...	2008-12-23 20:20:21	This is a list of ...[this, is, a, lis...]	[this, is, a, lis...]	{262144,[19,120,3,...]
569	Anthropology	2008-12-28 23:04:30	Anthropology (, f...)	[anthropology, (, f...)]	{262144,[154,528,...]

only showing top 20 rows

That hashing operation basically computed term frequencies for us by storing how often each hashed word occurred in each article. So we have TF, but we want TF/IDF scores for every term in every document. We'll store these final scores in a new column called "features", which is a sparse vector containing TF/IDF scores for each feature.

```
[ ]: idf = IDF(inputCol="rawFeatures", outputCol="features")
idfModel = idf.fit(featureizedData)
rescaledData = idfModel.transform(featureizedData)

[ ]: rescaledData.show()
```

Saving completed

Simple 0 2 10 PySpark | Idle

28°C Partly sunny

ENGLISH ENG US

12:47 AM 6/2/2023

```
idf = IDF(inputCol="rawFeatures", outputCol="features")
idfModel = idf.fit(featurizedData)
rescaledData = idfModel.transform(featurizedData)
rescaledData.show()
```

Explanation:

IDF is the class being instantiated

"inputCol="rawFeatures" specifies the name of the input column in the DataFrame that contains the raw feature vectors. In your case, in previous step we produced a column named "rawFeatures" containing the feature vectors generated by the "HashingTF transformation.

`outputCol="features"` specifies the name of the output column in the DataFrame where the IDF-weighted feature vectors will be stored. In this case, the output column will be named "features".

Once we have defined the IDF instance, we can apply it to a DataFrame using the fit() method to compute the IDF weights and the transform() method to apply the IDF transformation. The input DataFrame 'featurizedData' will be transformed into a new DataFrame 'rescaledData', which will contain the original columns along with an additional column called "features" that contains the IDF-weighted feature vectors.

The `fit()` method is used to compute the IDF weights based on the input DataFrame. It returns an "IDFModel" object that can be used to transform other DataFrames with the same schema

It's important to note that the IDF weights are typically computed based on a collection of documents to provide a measure of the informativeness of each term. The IDF class in Spark MLlib assumes that each row in the Input DataFrame represents a document, and the IDF weights are computed based on the entire dataset.

```
[8]: ldf = IDF(inputCol="raw_features", outputCol="features")
ldfModel = ldf.fit(featurizedData)
rescaledData = ldfModel.transform(featurizedData)
Last executed at 2023-06-02 00:48:17 in 5.38s
```

▶ Spark Job Progress

```
[0]: rescaledData.show()
Last executed at 2023-06-02 00:59:28 in 15.42s
+-----+-----+-----+-----+
| ID | Title | Time | Document | word | rawFeatures | features |
+-----+-----+-----+-----+
| 12 | Anarchism | 2008-12-30 06:33:05 | [anarchism (cometi...|[anarchism, (comet...| [(262144,[116,129,...| [(262144,[116,129,...|
| 25 | Autism | 2008-12-24 20:41:05 | [autism is a brain...|[autism, is, a, b...| [(262144,[521,1546,...| [(262144,[521,1546,...|
| 39 | Albedo | 2008-12-29 18:19:09 | [the albedo of an ...|[the, albedo, of,...| [(262144,[1625,179,...| [(262144,[1625,179,...|
| 290 | A | 2008-12-27 04:33:16 | [The letter A is t...|[the, letter, a, ...| [(262144,[5303,603,...| [(262144,[5303,603,...|
| 303 | Alabama | 2008-12-29 08:15:47 | [Alabama (formally...|[Alabama, (formal...| [(262144,[93,115,...| [(262144,[93,115,...|
| 305 | Achilles | 2008-12-29 18:18:01 | [the\n\b\n\\n\\nGreek...|[the\\n\\nGreek, ...| [(262144,[305,991,...| [(262144,[305,991,...|
| 307 | Abraham Lincoln | 2008-12-29 20:30:45 | [the president of the United States of America...|[the, president, o...| [(262144,[105,1001,...| [(262144,[105,1001,...|
| 308 | Aristotle | 2008-12-20 23:54:40 | [Aristotle (Greek...|[Aristotle, (Greek...| [(262144,[154,619,...| [(262144,[154,619,...|
| 309 | An American in Paris | 2008-09-27 19:29:28 | [An American in Pa...|[An, American, in,...| [(262144,[1366,213,...| [(262144,[1366,213,...|
| 324 | Academy Award | 2008-12-28 17:50:43 | [The Academy Award...|[the, academy, aw...| [(262144,[161,521,...| [(262144,[161,521,...|
| 330 | Actrius | 2008-05-23 15:24:32 | [Actrius (Actress...|[actrius, (actres...| [(262144,[6558,674,...| [(262144,[6558,674,...|
| 332 | Animalia (book) | 2008-12-18 11:12:34 | [the\\nAnimalia (I...|[the\\nAnimalia (I...| [(262144,[2284,609,...| [(262144,[2284,609,...|
| 334 | International Ato... | 2008-11-21 22:40:20 | [International Ato...|[international, a...| [(262144,[847,925,...| [(262144,[847,925,...|
| 336 | Altruis | 2008-12-17 17:17:17 | [Altruism is self...|[altruism, is, se...| [(262144,[521,1172,...| [(262144,[521,1172,...|
| 339 | Alain Rand | 2008-12-30 08:03:39 | [Alain Rand (Ma...|[Alain, Rand, (Ma...| [(262144,[105,1001,...| [(262144,[105,1001,...|
| 340 | Alain Connes | 2008-09-03 13:41:39 | [Alain Connes (mat...|[alain, connes, (...| [(262144,[154,1599,...| [(262144,[154,1599,...|
| 344 | Allan Duan | 2008-11-14 05:28:58 | [Allan Duan (April...|[allan, duan, (ap...| [(262144,[1578,181,...| [(262144,[1578,181,...|
| 358 | Algeria | 2008-12-29 02:54:36 | [Algeria (, al-Jaz...|[algeria, (, al-...| [(262144,[161,369,...| [(262144,[161,369,...|
| 359 | [List of character... | 2008-12-23 20:20:21 | [This is a lis...|[this, is, a, lis...| [(262144,[19,120,...| [(262144,[19,120,...|
| 369 | Anthropology | 2008-12-28 23:04:30 | [Anthropology (, f...|[anthropology, (...| [(262144,[154,528,...| [(262144,[154,528,...|
+-----+-----+-----+-----+
only showing top 20 rows
```

So let's use this to do a search for the term "Gettysburg". Again, we need numbers, not words, so the first task is to get the hash value for "Gettysburg"

```
[1]: from pyspark.sql.types import *
schema = StructType([StructField("words", ArrayType(StringType()))])
```

Saving completed

Mode: Command ⌂ Ln 1, Col 20 TF-IDF.ipynb

28°C Partly sunny

ENG US ⌂ 12:50 AM 6/2/2023

```
from pyspark.sql.types import *
```

```
schema = StructType([StructField("words", ArrayType(StringType()))])
```

```
df = spark.createDataFrame([[["gettysburg"]]], schema).toDF("words")
df.show()
```

Explanation:The above code defines schema for the DataFarme. Then, it creates a DataFrame called ‘df’ with a single row that contains the word “gettysburg” as the value in the “word” column.

```
*[10]: from pyspark.sql.types import *
schema = StructType([StructField("words", ArrayType(StringType()))])
df = spark.createDataFrame([[["gettysburg"]]], schema).toDF("words")
df.show()
```

Last executed at 2023-06-02 00:56:55 in 13.44s

```
+-----+
| words|
+-----+
|[gettysburg]|
+-----+
```

```
gettysburg = hashingTF.transform(df)
gettysburg.show()
```

Explanation:It applies ‘hashing tf’ transformation on the DataFrame ‘df’ to obtain the feature vector representation of the word “gettysburg”. The resulting DataFrame is assigned to ‘gettysburg’.

```
*[11]: gettysburg = hashingTF.transform(df)
gettysburg.show()
```

Last executed at 2023-06-02 00:57:05 in 3.34s

```
+-----+-----+
| words | rawFeatures |
+-----+-----+
|[gettysburg]| (262144,[116775]),...
```

```
featureVec = gettysburg.select('rawFeatures').collect()
```

Explanation:It selects the “rawFeatures” column from the ‘gettysburg’ DataFrame and collects all the rows into Python list called ‘featureVec’.

```
*[12]: featureVec = gettysburg.select('rawFeatures').collect()
print(featureVec)
```

Last executed at 2023-06-02 00:57:26 in 793ms

» Spark Job Progress

gettysburgID = int(featureVec[0].rawFeatures.indices[0])

Explanation: It retrieves the first element of ‘featureVec’, which corresponds to the feature vector for the word “gettysburg”. It then extracts the index of the first non-zero element in the feature vector using ‘indices[0]’ and converts it to an integer.

```
*[13]: gettysburgID = int(featureVec[0].rawFeatures.indices[0])
print(gettysburgID)
```

Last executed at 2023-06-02 00:57:46 in 97ms

116775

We have the number that represents "Gettysburg". Now we can add another column named "score" that just extracts the TF/IDF value for Gettysburg for each document.

```
from pyspark.sql.types import FloatType
from pyspark.sql.functions import udf
```

```
termExtractor = udf(lambda x: float(x[gettysburgID]), FloatType())
```

```
gettysburgDF = rescaledData.withColumn('score', termExtractor(rescaledData.features))
```

```
gettysburgDF.show()
```

```
[14]: from pyspark.sql.types import FloatType
from pyspark.sql.functions import udf

termExtractor = udf(lambda x: float(x[gettysburgID]), FloatType())
gettysburgDF = rescaledData.withColumn('score', termExtractor(rescaledData.features))

gettysburgDF.show()
```

Last executed at 2023-06-02 00:59:26 in 21.39s

» Spark Job Progress

ID	Title	Time	Document	words	rawFeatures	features	score
121	Anarchism	2008-12-20 06:23:05	Anarchism (some...)	{[archivism, (some...)]}	{[262144, [116, 120, ...]}	{[262144, [116, 120, ...]}	0.0
25	Autism	2008-12-24 20:41:05	Autism is, a b...)	{[autism, is, a, b...]}	{[262144, [521, 1546, ...]}	{[262144, [521, 1546, ...]}	0.0
39	Albedo	2008-12-29 18:19:09	The albedo of an ...	{[the, albedo, of,...]}	{[262144, [1625, 179, ...]}	{[262144, [1625, 179, ...]}	0.0
290	A	2008-12-27 04:33:16	The letter A is t... [the, letter, a, ...]	{[the, letter, a, ...]}	{[262144, [530, 603, ...]}	{[262144, [530, 603, ...]}	0.0
303	Alabama	2008-12-24 08:15:47	Alabama (formal...)	{[alabama, (formal...)]}	{[262144, [93, 115, 3, ...]}	{[262144, [93, 115, 3, ...]}	0.0
305	Abraham Lincoln	2008-12-28 08:18:23	Abraham Lincoln (U.S. president, lincoln, presiden...)	{[abraham, lincoln, presiden...]}	{[262144, [309, 95, 1, ...]}	{[262144, [309, 95, 1, ...]}	0.0
307	Aristotle	2008-12-28 13:54:48	Aristotle (Greek...)	{[aristotle, (gree...)]}	{[262144, [154, 619, ...]}	{[262144, [154, 619, ...]}	30.699241
308	Aristotle	2008-12-28 17:50:43	The Academy Award (...	{[the, academy, aw...]}	{[262144, [161, 521, ...]}	{[262144, [161, 521, ...]}	0.0
324	Academy Award	2008-12-23 15:24:32	Academy Award (Acresse...)	{[actrius, (actress...)]}	{[262144, [6558, 674, ...]}	{[262144, [6558, 674, ...]}	0.0
330	Actrius	2008-05-23 15:24:32	Actrius (Actresse...)	{[actrius, (actress...)]}	{[262144, [228, 609, ...]}	{[262144, [228, 609, ...]}	0.0
332	Animals (book)	2008-12-23 22:40:12	International At... [international, ato...]	{[international, ato...]}	{[262144, [847, 925, ...]}	{[262144, [847, 925, ...]}	0.0
334	International Atto...	2008-11-24 08:15:21	International Atto... [international, ato...]	{[international, ato...]}	{[262144, [521, 927, ...]}	{[262144, [521, 927, ...]}	0.0
336	Alain Connex	2008-12-29 08:03:06	Alain Connex (ber...)	{[alain, connex, (ber...)]}	{[262144, [116, 363, ...]}	{[262144, [116, 363, ...]}	0.0
339	Aym Benyamin	2008-12-29 08:03:06	Alain Connex (ber...)	{[alain, connex, (ber...)]}	{[262144, [154, 159, ...]}	{[262144, [154, 159, ...]}	0.0
340	Alain Connex	2008-09-03 13:41:39	Alain Connex (ber...)	{[alain, connex, (ber...)]}	{[262144, [1578, 181, ...]}	{[262144, [1578, 181, ...]}	0.0
344	Allan Duan	2008-11-14 09:25:58	Allan Duan (Apr...)	{[allan, duan, (ap...)]}	{[262144, [161, 369, ...]}	{[262144, [161, 369, ...]}	0.0
358	Algeria	2008-12-29 02:54:36	Algeria (, al-Jaz...)	{[algeria, (, al-Jaz...)]}	{[262144, [161, 369, ...]}	{[262144, [161, 369, ...]}	0.0
359	List of character...	2008-12-23 20:20:21	This is a list of... [this, is, a, lis...]	{[this, is, a, lis...]}	{[262144, [19, 120, 3, ...]}	{[262144, [19, 120, 3, ...]}	0.0
569	Anthropology	2008-12-28 23:04:30	Anthropology (, f...)	{[anthropology, (, f...)]}	{[262144, [154, 528, ...]}	{[262144, [154, 528, ...]}	0.0

only showing top 20 rows

Now have to sort our articles by score, and we'll have the most relevant articles for ‘Gettysburg’.

```
sortedResults = gettysburgDF.filter("score > 0").orderBy('score', ascending=False).select('ID', 'Title', 'Document', 'score')
```

```
sortedResults.show(truncate=100)
```

```
[15]: sortedResults = gettysburgDF.filter("score > 0").orderBy('score', ascending=False).select('ID', 'Title', 'Document', 'score')
sortedResults.show(truncate=100)
```

Last executed at 2023-06-02 00:59:46 in 5.36s

» Spark Job Progress

ID	Title	Document	score
307	Abraham Lincoln	Abraham Lincoln (February 12, 1809 – April 15, 1865) was the sixteenth President of the United States.	30.699241
1135	Abner Doubleday	Abner Doubleday (June 26, 1819 – January 26, 1899) was a career United States Army officer and Union general.	25.579979
1863	American Civil War	The American Civil War (1861–1865), also known as the War Between the States and several other names,	15.347987
1998	Austin, Texas	Austin is the capital of the U.S. state of Texas and the county seat of Travis County. Situated in the central part of the state, Austin is the second largest city in Texas, after Houston.	5.115996
2085	Assyria	Assyria was a geographic region on the Upper Tigris river, named for its original capital, the ancient city of Nineveh.	5.115996

7) Deleting Resources

A) Delete workspace:

The screenshot shows the EMR Studio interface. On the left, there's a sidebar with 'EMR Studio' at the top, followed by 'Dashboard', 'Workspaces' (which is selected), 'Serverless', 'Applications', and 'Clusters'. Under 'Clusters', it lists 'EMR on EC2' and 'EMR on EKS'. The main content area is titled 'Studio: TF-IDF' and shows a table for 'Workspaces (1) Info'. The table has columns: Workspace name, Status, Cluster ID, Creation time (UTC-05:00), Last modified (UTC-05:00). A single row is present: 'TF-IDF' (Status: Ready, Cluster ID: -, Creation time: June 01, 2023, 23:41, Last modified: June 01, 2023, 23:42). At the top right of the table, there are 'Actions' buttons for Start, Stop, and Delete, with 'Delete' being highlighted. Below the table is a search bar with placeholder text 'Find Workspaces by name, status, or last modified by'. The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating the date and time as 6/2/2023.

B) Terminate Cluster

The screenshot shows the AWS EMR console. At the top, there's a message: 'The new Amazon EMR console is now the default console. Continue to use the new console, or switch to the old console. Learn more.' Below this is a green success message: 'Studio TF-IDF created successfully.' The main content area is titled 'tfidf' and shows a 'Summary' tab. It has four sections: 'Cluster info' (Cluster ID: j-3BQPVGWJFVTC, Configuration: Instance groups, Capacity: 1 Primary, 0 Core, 0 Task), 'Applications' (Amazon EMR version: emr-6.10.0, Installed applications: Spark 3.3.1, Hadoop 3.3.3, Livy 0.7.1, JupyterEnterpriseGateway 2.6.0), 'Cluster management' (Log destination in Amazon S3: aws-emr-resources-asia/elasticmapreduce, Persistent application UIs: Spark History Server, YARN timeline server), and 'Status and time' (Status: Waiting, Creation time: June 01, 2023, 23:45 (UTC-05:00), Elapsed time: 1 hour, 15 minutes). At the top right of the summary section, there's an 'Actions' button with a dropdown menu. The 'Terminate cluster' option is highlighted in this menu. Below the summary, there are tabs for 'Properties', 'Bootstrap actions', 'Instances', 'Steps', 'Applications', 'Configurations', 'Monitoring', 'Events', and 'Tags (1)'. At the bottom, there are three cards: 'Operating system' (Info), 'Cluster logs' (Info), and 'Cluster termination' (Info). The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating the date and time as 6/2/2023.

C) Delete EMR Studio

The screenshot shows the AWS EMR Studio interface. A blue banner at the top indicates that the new Amazon EMR console is now the default console. Below this, another message informs users about the availability of EMR Notebooks as EMR Studio Workspaces. The main content area displays a table titled 'Studios (1) Info' with one entry: 'TF-IDF'. The table includes columns for 'Studio name', 'Creation time (UTC-05:00)', 'Authenticated by', and 'Studio Access URL'. The 'Delete' button is highlighted with a blue arrow.

D) Delete S3 buckets

The screenshot shows the AWS S3 console. A blue banner at the top indicates that the new Amazon S3 console is now the default console. The main content area displays a table titled 'Buckets (2) Info' with two entries: 'aws-emr-resources-asia' and 'bucket-emr-1234'. The table includes columns for 'Name', 'AWS Region', 'Access', and 'Creation date'. The 'Delete' button is highlighted with a blue arrow.