Azure Databricks and AWS S3 Storage

Databricks is an integrated analytics environment powered by Apache Spark which let you connect and read from many data sources such as AWS S3, HDFS, MySQL, SQL Server, Cassandra etc. We are focusing on S3 , Since it is very easy to work with.

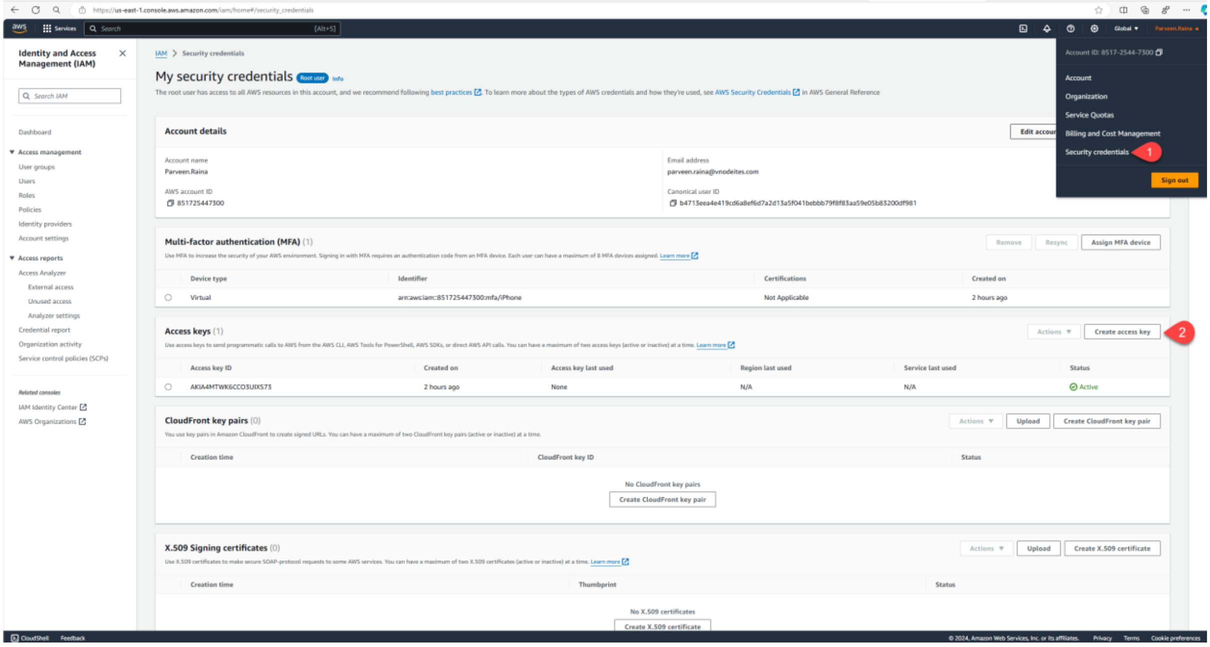
Prerequisites:

- *you must have AWS & Azure account with subscription*
- *S3 bucket created and contains some objects to work with*

Creating S3 storage bucket is beyond this section.

Step 1.

In AWS Management Console, go to 'Security Credentials' under your user account name. Then , you click 'Access keys (access key ID and secret access key)' section under My Security Credentials which will be provided with option called 'Create New Access Key'. once you click that option, you will be given **access key ID and secret access key.** please save it in notepad by pressing show access key or download key file for later use.



My security credentials

The root user has access to all AWS resources in this account, and we recommend following best practices. To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference

Account details

Account name: Parveen.Raina

Email address: parveen.raina@vnodeites.com

AWS account ID: 851725447300

Canonical user ID: b4713eea4e419cd6a8ef6d7a2d13a5f041bebbb79f8f83aa59e05b83200df981

Multi-factor authentication (MFA) (1)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more

	Device type	Identifier	Certifications	Created on
○	Virtual	arn:aws:iam::851725447300:mfa/iPhone	Not Applicable	2 hours ago

Access keys (1)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

	Access key ID	Created on	Access key last used	Region last used	Service last used	Status
○	AKIA4MTWK6CCO3UIX573	2 hours ago	None	N/A	N/A	Active

CloudFront key pairs (0)

You use key pairs in Amazon CloudFront to create signed URLs. You can have a maximum of two CloudFront key pairs (active or inactive) at a time.

	Creation time	CloudFront key ID	Status

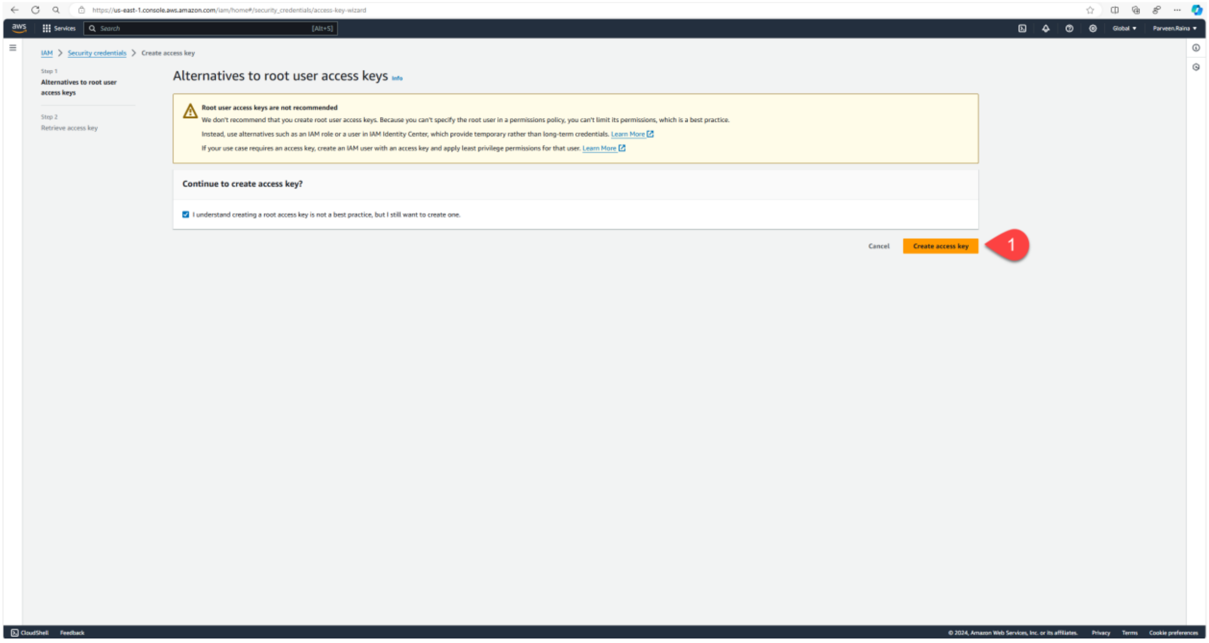
No CloudFront key pairs

X.509 Signing certificates (0)

Use X.509 certificates to make secure SOAP-protocol requests to some AWS services. You can have a maximum of two X.509 certificates (active or inactive) at a time. Learn more

	Creation time	Thumbprint	Status

No X.509 certificates



https://us-east-1.console.aws.amazon.com/iam/home#/security_credentials/access-key-wizard
Services
Search [Alt+S]
Global
Parveen.Raina
IAM > Security credentials > Create access key
Step 1
Alternatives to root user access keys
Step 2
Retrieve access key
Cancel
Create access key
1
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Alternatives to root user access keys Info

Root user access keys are not recommended

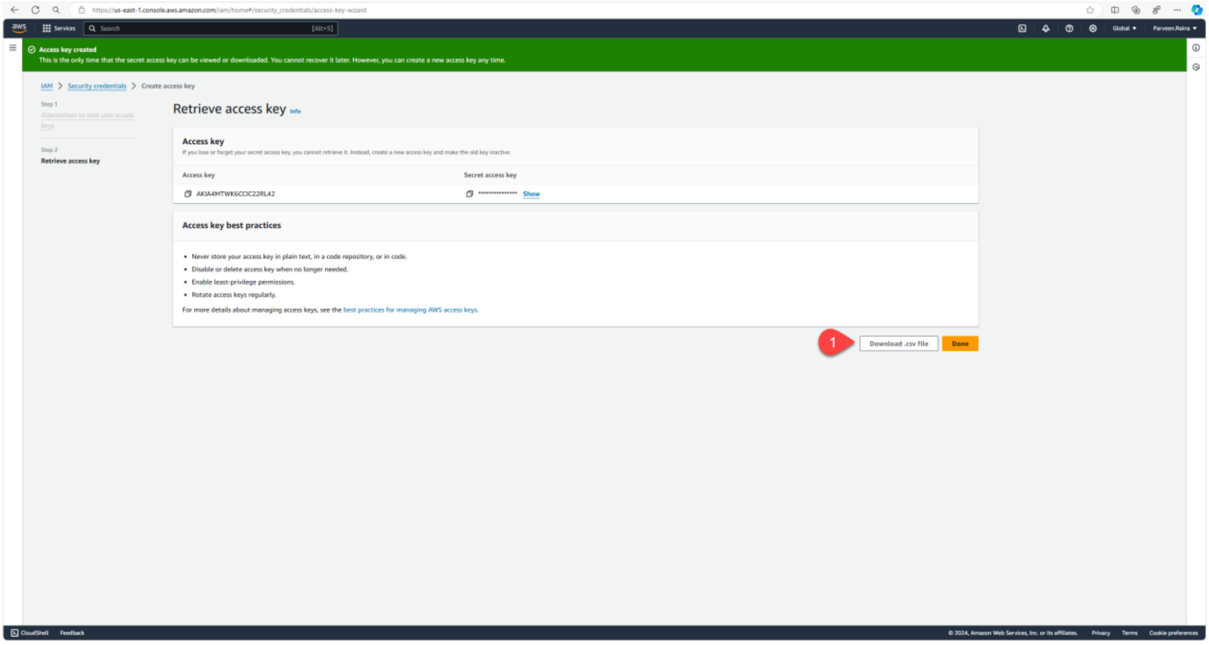
We don't recommend that you create root user access keys. Because you can't specify the root user in a permissions policy, you can't limit its permissions, which is a best practice.

Instead, use alternatives such as an IAM role or a user in IAM Identity Center, which provide temporary rather than long-term credentials. Learn More

If your use case requires an access key, create an IAM user with an access key and apply least privilege permissions for that user. Learn More

Continue to create access key?

☑ I understand creating a root access key is not a best practice, but I still want to create one.



https://us-east-1.console.aws.amazon.com/iam/home#/security_credentials/access-key-wizard
Services
Search [Alt+S]
Global
Parveen.Raina
IAM > Security credentials > Create access key
Step 1
Alternatives to root user access keys
Step 2
Retrieve access key
Show
1
Download .csv file
Done
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Retrieve access key Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIA4MTWK6CCIC22RL42	*************** Show

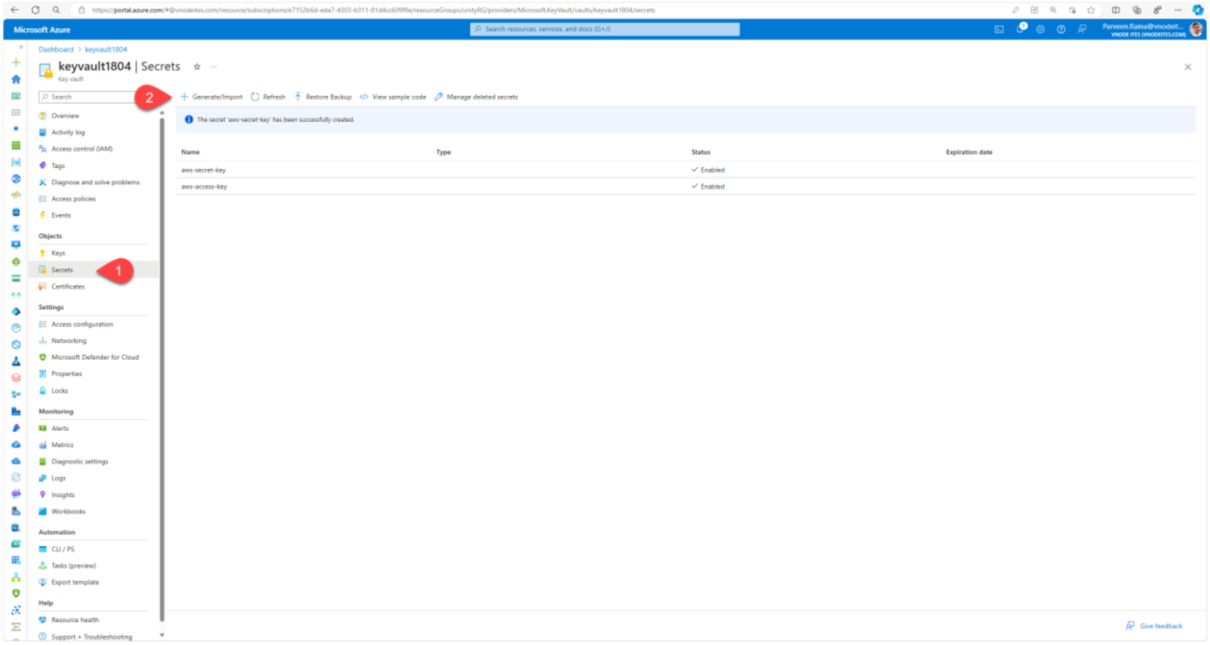
Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

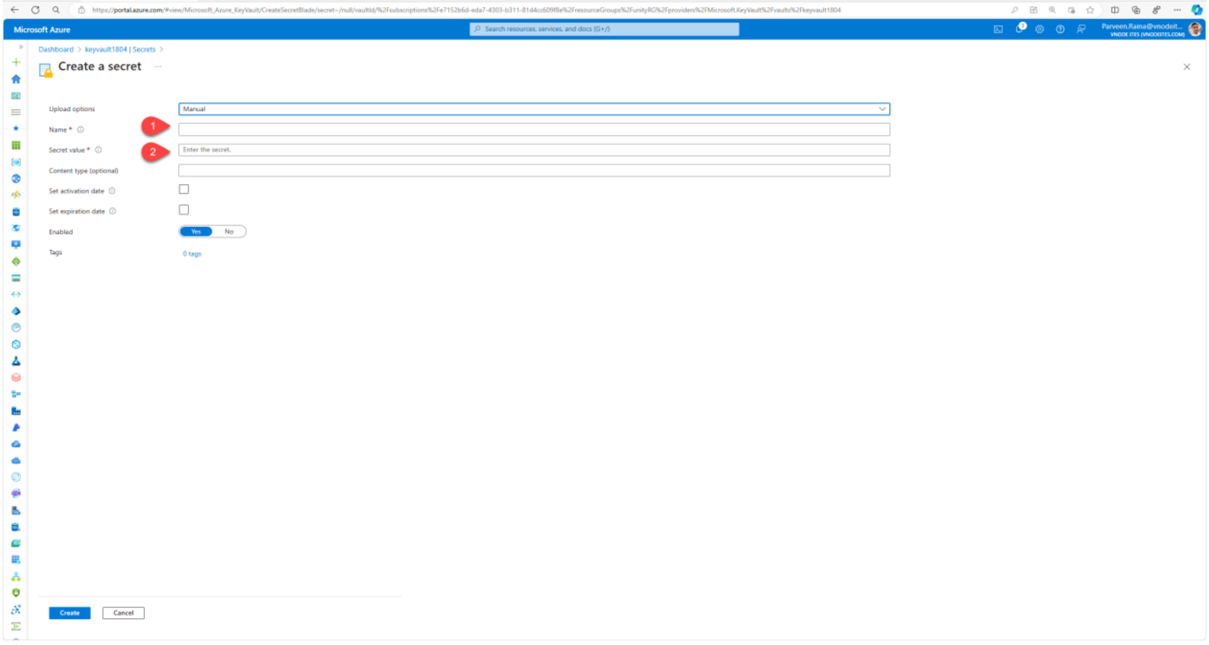
Step 2.

Since we access S3 bucket using databricks-backed scope, Secrets should be created by putting access key & secret key values in Azure key vault. Go to Azure Key Vault, in the resource menu, click secrets under Settings category. Then, click + sign (Generate/Import) in the command bar. you will be prompted with below window



The secret 'aws-secret-key' has been successfully created.

Name	Type	Status	Expiration date
aws-secret-key		✓ Enabled	
aws-access-key		✓ Enabled	

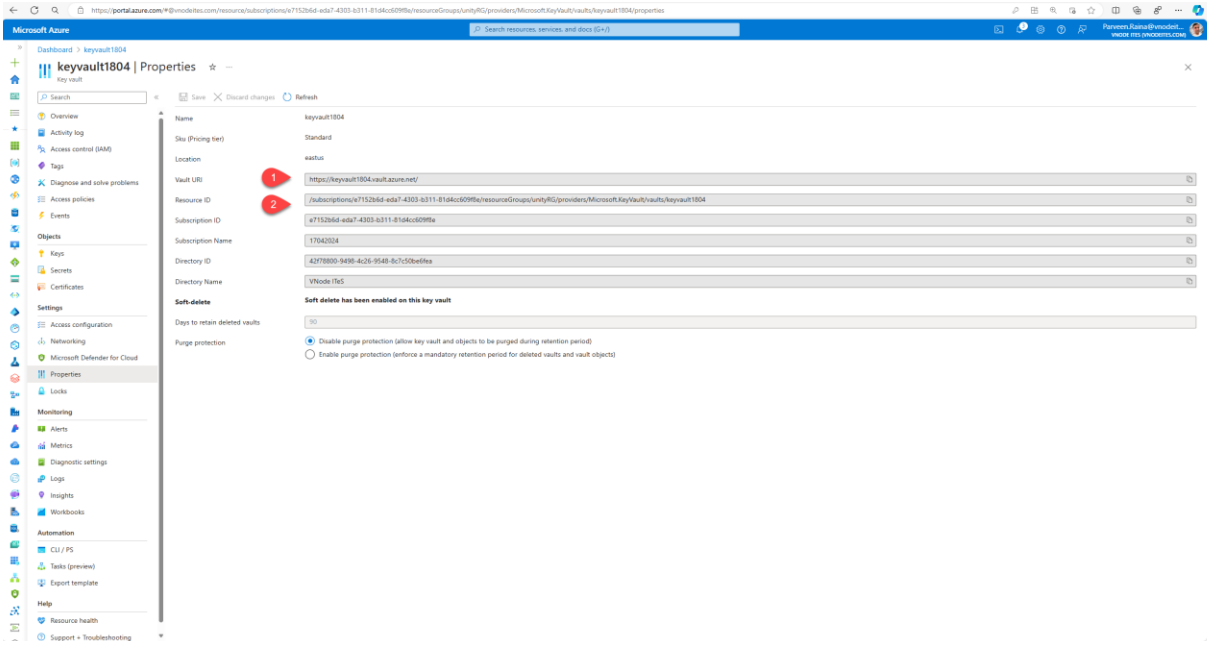


Upload options	Manual
Name *	
Secret value *	Enter the secret.
Content type (optional)	
Set activation date	□
Set expiration date	□
Enabled	[✓ Yes] No
Tags	0 tags

Give ***aws-access-key*** and and ***aws-secret-key*** for name and paste copied ***access key & secret access key*** values in step 1 in the place of Value. By doing so two times, two secrets will be created. leave others default and click create. you are done with creating secret for accessing storage account.

Step 3.

Navigate to properties under resource menu of Key Vault & copy DNS name and Resource ID and save it in notepad. This will be used whilst creating secret scope.



https://portal.azure.com/#@vnodeites.com/resource/subscriptions/e7152b6d-eda7-4303-b311-81d4cc609f8e/resourceGroups/unityRG/providers/Microsoft.KeyVault/vaults/keyvault1804/properties

Dashboard > keyvault1804

keyvault1804 | Properties

Key vault

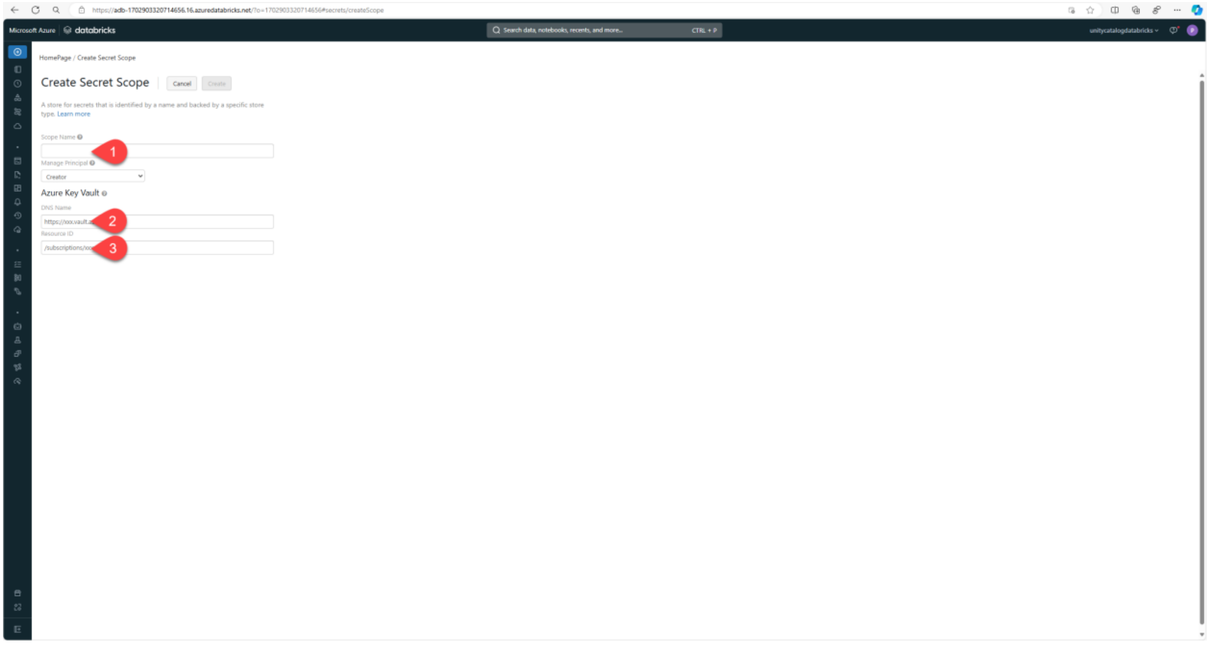
Name	keyvault1804
Sku (Pricing tier)	Standard
Location	eastus
Vault URI	https://keyvault1804.vault.azure.net/
Resource ID	/subscriptions/e7152b6d-eda7-4303-b311-81d4cc609f8e/resourceGroups/unityRG/providers/Microsoft.KeyVault/vaults/keyvault1804
Subscription ID	e7152b6d-eda7-4303-b311-81d4cc609f8e
Subscription Name	17042024
Directory ID	42f78800-9498-4c26-9548-8c7c50be6fea
Directory Name	VNode ITeS
Soft-delete	**Soft delete has been enabled on this key vault**
Days to retain deleted vaults	90
Purge protection	● Disable purge protection (allow key vault and objects to be purged during retention period) ○ Enable purge protection (enforce a mandatory retention period for deleted vaults and vault objects)

Step 4.

Go to https://<your_azure_databricks_url>#secrets/createScope

Ex- https://adb-1702903320714656.16.azuredatabricks.net/#secrets/createScope

you will be directed to below window



https://adb-1702903320714656.16.azuredatabricks.net/?o=1702903320714656#secrets/createScope

HomePage / Create Secret Scope

Create Secret Scope

A store for secrets that is identified by a name and backed by a specific store type. Learn more

Scope Name

Manage Principal: Creator

Azure Key Vault

DNS Name: https://xxx.vault.a

Resource ID: /subscriptions/xx

Give scope name ***'aws'*** which is uniquely identified in the database maintained by Databricks, leave Manage Principal as Creator and paste copied values of DNS name & resource ID in place of DNS Name and Resource ID fields. Click Create & you will be given with success message. Remember scope name or save it in file.

Step 5.

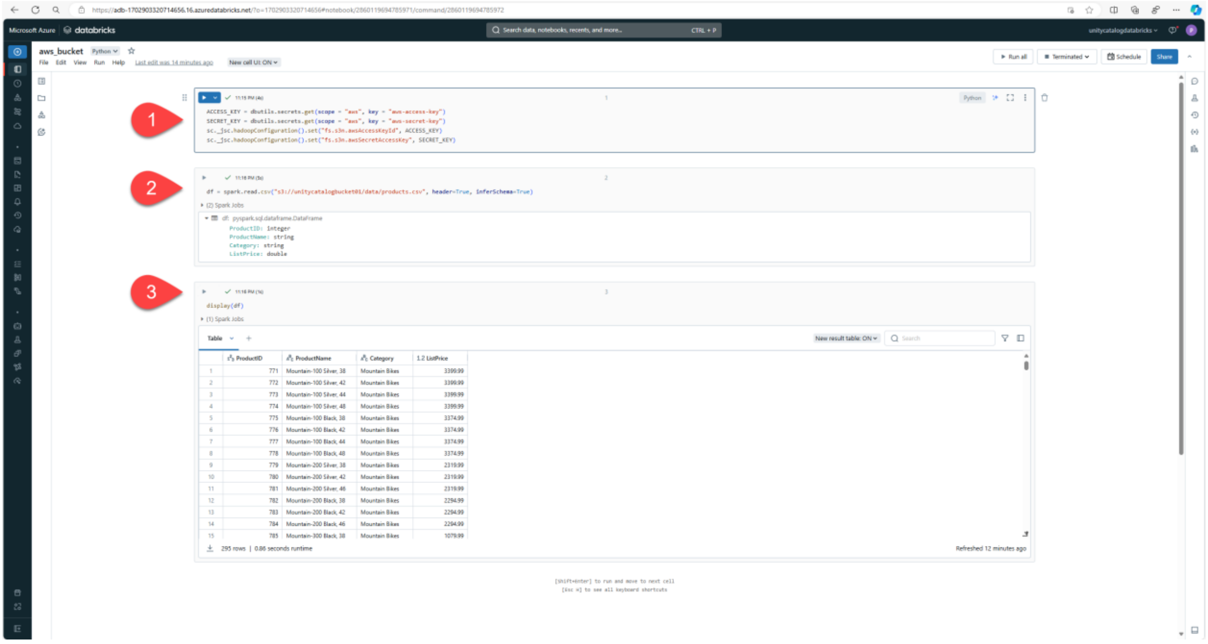
Accessing S3 bucketing directly by setting AWS keys in the spark context. Secrets utilities of Databricks utilities (DBUtils) will help us fetching sensitive credentials information without making them visible.

ACCESS_KEY = dbutils.secrets.get(scope = "aws", key = "aws-access-key")

SECRET_KEY = dbutils.secrets.get(scope = "aws", key = "aws-secret-key")

sc._jsc.hadoopConfiguration().set("fs.s3n.awsAccessKeyId", ACCESS_KEY)

sc._jsc.hadoopConfiguration().set("fs.s3n.awsSecretAccessKey", SECRET_KEY)



1

```
ACCESS_KEY = dbutils.secrets.get(scope = "aws", key = "aws-access-key")
SECRET_KEY = dbutils.secrets.get(scope = "aws", key = "aws-secret-key")
sc._jsc.hadoopConfiguration().set("fs.s3n.awsAccessKeyId", ACCESS_KEY)
sc._jsc.hadoopConfiguration().set("fs.s3n.awsSecretAccessKey", SECRET_KEY)
```

2

```
df = spark.read.csv("s3://unitycatalogbucket01/data/products.csv", header=True, inferSchema=True)
```

```
(2) Spark Jobs
df: pyspark.sql.dataframe.DataFrame
    ProductID: integer
    ProductName: string
    Category: string
    ListPrice: double
```

3

```
display(df)
```

(1) Spark Jobs

	ProductID	ProductName	Category	1.2 ListPrice
1	771	Mountain-100 Silver, 38	Mountain Bikes	3399.99
2	772	Mountain-100 Silver, 42	Mountain Bikes	3399.99
3	773	Mountain-100 Silver, 44	Mountain Bikes	3399.99
4	774	Mountain-100 Silver, 48	Mountain Bikes	3399.99
5	775	Mountain-100 Black, 38	Mountain Bikes	3374.99
6	776	Mountain-100 Black, 42	Mountain Bikes	3374.99
7	777	Mountain-100 Black, 44	Mountain Bikes	3374.99
8	778	Mountain-100 Black, 48	Mountain Bikes	3374.99
9	779	Mountain-200 Silver, 38	Mountain Bikes	2319.99
10	780	Mountain-200 Silver, 42	Mountain Bikes	2319.99
11	781	Mountain-200 Silver, 46	Mountain Bikes	2319.99
12	782	Mountain-200 Black, 38	Mountain Bikes	2294.99
13	783	Mountain-200 Black, 42	Mountain Bikes	2294.99
14	784	Mountain-200 Black, 46	Mountain Bikes	2294.99
15	785	Mountain-300 Black, 38	Mountain Bikes	1079.99

295 rows | 0.86 seconds runtime

Refreshed 12 minutes ago

Spark context is now configured to connect to AWS S3 bucket by above diagrammatically.

Lets check whether we can be able to read below .csv file stored in my_bucket

df = spark.read.csv("s3://unitycatalogbucket01/data/products.csv", header=True, inferSchema=True)