**References:** [**https://towardsdatascience.com/how-i-connect-an-s3-bucket-to-a-databricks-notebook-to-do-analytics-a7b5258619b8**](https://towardsdatascience.com/how-i-connect-an-s3-bucket-to-a-databricks-notebook-to-do-analytics-a7b5258619b8)

[**https://techcommunity.microsoft.com/t5/analytics-on-azure/data-bricks-in-azure-vs-aws/m-p/1776174**](https://techcommunity.microsoft.com/t5/analytics-on-azure/data-bricks-in-azure-vs-aws/m-p/1776174)

**How Databricks on Azure different from AWS:**

There are a lot of reasons I would choose Azure Databricks compared to Databricks on AWS. At a high level, Azure Databricks is a first party service on Azure. What that means is that it's more than a partnership- there are deep integrations between Azure services and Azure Databricks. Examples of what this entails:

* Azure Key Vault Secret Stores in Databricks- Users can work with secrets they have access to, without the secrets ever being exposed to the end user
* Azure Active Directory Support- User credentials on Azure used to authenticate to Azure Databricks, and data in Azure Data Lake Storage accessed through those credentials 'passed through' from Azure Databricks. This allows teams to configure permissions on files to users and groups, and Azure Databricks authenticates the rest
* Top level integration with Azure Data Factory, allowing scheduling of notebooks as jobs in a data estate
* Integrating easily with other Azure Data Services (Cosmos DB, Synapse) through service endpoints on private networks

As a general rule, the integrations to the rest of the Azure platform are deeper on Azure Databricks, compared to how even Databricks on AWS integrates with other AWS services. Overall, this builds a more seamless and streamlined experience for building out your data estate with Databricks.

**How I connect an S3 bucket to a Databricks notebook to do analytics:**

**Adding a new AWS user**

To be able to read the data from our S3 bucket, we will have to give access from AWS for this we need to add a new AWS user:

We start by going to the AWS IAM service ->Users ->Add a user

We enter the name of the user as well as the type of access.

We then give this user access to S3.

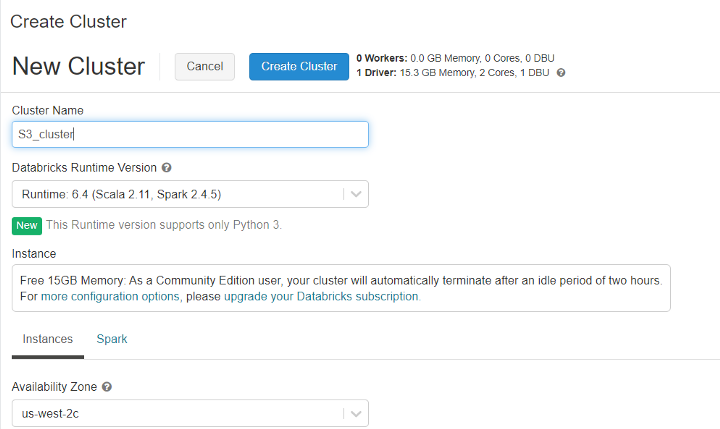
We click on “view” and save these keys for later use.

Now that the user has been created, we can go to the connection from Databricks.

**Configure your Databricks notebook**

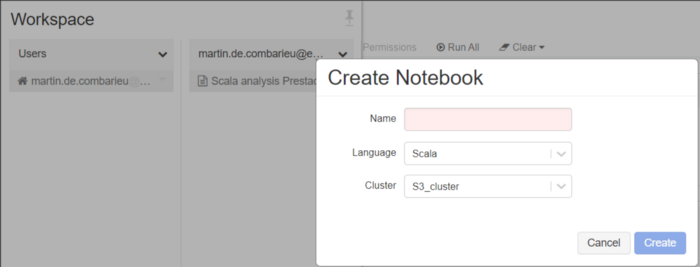
Now that our user has access to the S3, we can initiate this connection in databricks.

If your account was just created, you would have to create a new cluster to run your notebook. Go to the cluster tab -> create cluster



Give it the name you want and the last stable runtime version. For this tutorial, I’m using Scala to process the data. Then click on “Create Cluster”

Now you have to create a notebook to run your code. To do that, go to workspace -> “Username” -> Create



**Connect and retrieve S3 data from Databricks**

**Connection:**



To connect your just created notebook to your AWS S3 bucket you just have to replace you access and secret key by the one you saved when you created a user earlier, remember?

You also have to replace the “AwsBucketName” attribute by your S3 bucket name.

The mount name just corresponds to the local name of the variable that will contain your S3 data. You can choose the name that you want.

**Retrieve data**

To retrieve data from the S3 bucket use this code.



Here just have to pay attention to the path. Pay attention to the storage structure of your S3 to retrieve only the data you are interested in.

Now just transform your data into a beautiful dataset just by using this line of code.

