

MICROSOFT AZURE

1. Mainly get the Azure Subscription using free student ID .
2. Login to the azure account using the Subscribed email id .
3. Go to the " More services " from Home page of the Azure, then select the "Resource Group" .

The screenshot shows the Microsoft Azure Home page. At the top right, there is a red circle around a button labeled "More services" with an arrow icon. A red arrow points from the text "Select this option" to this button. The page also features sections for "Azure services" and "Resources".

The screenshot shows the Microsoft Azure All services page. On the left, there is a sidebar with a section titled "Categories" where "Resource groups" is listed under "AI + machine learning". A red circle highlights the "Resource groups" icon, and a red arrow points from the text "Select this option" to it. The main content area displays various service categories like "Azure Active Directory", "Virtual machines", "App Services", etc., with a list of AI services below.

4. Then select the “Create” option to create Resource Group .

The screenshot shows the Microsoft Azure Resource groups page. At the top, there is a navigation bar with various links like 'Learner dashboard...', 'Welcome | AWS Trai...', 'Swayam', 'NextStep- Tata Con...', 'Chat GPT', 'VTU Examination P...', 'Electronic Docume...', 'vandit-bera/React...', and 'Voters' Services Por...'. Below the navigation bar, the main title is 'Resource groups' with a subtitle 'presidencyuniversity.in (presidencyuniversity.in)'. A red circle highlights the '+ Create' button. A red arrow points from the text 'Select this option' to the '+ Create' button. There are also filter options like 'Filter for any field...', 'Subscription equals all', 'Location equals all', and 'Add filter'. The results section shows 'Showing 0 to 0 of 0 records.' Below this, there is a large hexagonal icon and the text 'No resource groups to display'. A message says 'Try changing or clearing your filters.' followed by a 'Create resource group' button. The bottom of the screen shows the Windows taskbar with icons for File Explorer, Edge, Excel, Word, etc., and the system tray showing the date and time as 2023-08-16.

5. Now the Create Resource Group will be opened , Mention the **Group name** and **Region** and Select the “Review+Create “ option .

The screenshot shows the 'Create a resource group' page. At the top, there is a navigation bar with various links like 'Learner dashboard...', 'Welcome | AWS Trai...', 'Swayam', 'NextStep- Tata Con...', 'Chat GPT', 'VTU Examination P...', 'Electronic Docume...', 'vandit-bera/React...', and 'Voters' Services Por...'. Below the navigation bar, the main title is 'Create a resource group' with a subtitle 'All services > Resource groups >'. The page has tabs for 'Basics', 'Tags', and 'Review + create'. Under 'Project details', there is a 'Subscription' dropdown set to 'Azure for Students' and a 'Resource group' input field containing 'Kanini_Santhosh', which is circled in red. A red arrow points from the text 'Fill this details' to the 'Resource group' field. Under 'Resource details', there is a 'Region' dropdown set to '(Asia Pacific) South India', which is also circled in red. A red arrow points from the text 'Fill this details' to the 'Region' field. At the bottom, there are buttons for 'Review + create', 'Previous', and 'Next : Tags >'. A red circle highlights the 'Review + create' button. A red arrow points from the text 'Next select this option' to the 'Review + create' button. The bottom of the screen shows the Windows taskbar with icons for File Explorer, Edge, Excel, Word, etc., and the system tray showing the date and time as 2023-08-16.

6. Now the Resource Group is created.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with various links and a user profile. Below it, the main content area is titled "Resource groups". A success message box is displayed, stating "Resource group created" and "Creating resource group 'Kanini_Santhosh' in subscription 'Azure for Students' succeeded." There are buttons for "Go to resource gr..." and "Pin to dashboard...". The main table lists one record: "Kanini_Santhosh" with "Subscription equals all" and "Location equals all". The table has columns for Name, Subscription, and Location. At the bottom, there are pagination controls ("Page 0 of 0") and a "Give feedback" link.

Creating the Data Storage Account for the Created Resource Group :

1. Go inside the Create resource group and select the “Create” option.

The screenshot shows the Microsoft Azure portal interface, specifically the "Kanini_Santhosh" resource group overview page. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, Resource visualizer, Events, Settings, Deployments, Security, Deployment stacks, Policies, Properties, Locks, and Cost Management. The "Overview" tab is selected. In the center, there's a red circle and a red arrow pointing to the "+ Create" button in the top navigation bar. Below the navigation bar, there are tabs for "Resources" and "Recommendations", with "Resources" being the active tab. A red text overlay says "Select this option". The main content area shows a message "No resources match your filters" with a note "Try changing or clearing your filters." At the bottom, there's a search bar and a taskbar with various icons.

Get Started

Service Providers

Management

Private Marketplace

Private Offer Management

My Marketplace

Favorites

Recently created

Private plans

Categories

Storage (77)

Compute (47)

IT & Management Tools (41)

Security (33)

Storage account

Storage Account Using ARM Template

Azure Storage Mover

APEX Protection Storage for Microsoft Azure (DDVE)

Storage Account Using ARM

S3 API for Azure Blob Storage (Flexify.IO)

Storage account

Storage Account Using ARM Template

Azure Storage Mover

APEX Protection Storage for Microsoft Azure (DDVE)

Storage Account Using ARM

S3 API for Azure Blob Storage (Flexify.IO)

Type here to search

31°C 3:12 PM 2023-08-16

2. Now fill up the required fields to create storage account and select "Standard " for performance .

Basics Advanced Networking Data protection Encryption Tags Review

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more about Azure storage accounts](#)

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription: Azure for Students

Resource group: Kanini_Santhosh [Create new](#)

Instance details

Storage account name: kaninipracticestorage

Region: (Asia Pacific) South India

Performance: Standard: Recommended for most scenarios (general-purpose v2 account)

Review

Give feedback

Type here to search

31°C 3:16 PM 2023-08-16

3. Next Review the details and create the storage account .

The screenshot shows the 'Create a storage account' wizard in the Microsoft Azure portal. The 'Review' tab is selected. The configuration includes:

- Subscription:** Azure for Students
- Resource Group:** Kanini_Santhosh
- Location:** southindia
- Storage account name:** kaninipracticestorage
- Deployment model:** Resource manager
- Performance:** Standard
- Replication:** Read-access geo-redundant storage (RA-GRS)

Advanced settings include:

- Enable hierarchical namespace: Disabled
- Enable network file system v3: Disabled
- Allow cross-tenant replication: Enabled
- Access tier: Hot
- Enable SFTP: Disabled
- Large file shares: Disabled

Networking settings include:

- Network connectivity: Public endpoint (all networks)
- Default routing tier: Microsoft network routing

At the bottom, there is a 'Create' button highlighted with a red circle, and a note: "select create to create the storage account".

Creating the Data bricks for the Created Resource Group :

1. Go inside the Create resource group and select the “Create” option.

The screenshot shows the 'Kanini_Santhosh' Resource Group page in the Microsoft Azure portal. The 'Resources' tab is selected. The '+ Create' button is highlighted with a red circle, and the text 'Select this option' is overlaid next to it.

The left sidebar shows the following navigation:

- All services > Resource groups
- Kanini_Santhosh (Resource group)
- Overview
- Activity log
- Access control (IAM)
- Tags
- Resource visualizer
- Events
- Settings
 - Deployments
 - Security
 - Deployment stacks
 - Policies
 - Properties
 - Locks
- Cost Management

The main area displays a message: "No resources match your filters" with a note: "Try changing or clearing your filters."

Home > Marketplace

Get Started

Service Providers

Management

Private Marketplace

Private Offer Management

My Marketplace

Favorites

Recently created

Private plans

Categories

Analytics (48)

AI + Machine Learning (15)

Search resources, services, and docs (G+)

Pricing : All X Operating System : All X Publisher Type : All X Product Type : All X

Azure services only Publisher name : All X

Showing 1 to 20 of 56 results for 'databricks': [Clear search](#)

Search the databricks key word

Tile view

databricks

Search the databricks key word

select the create azure databricks

Is Marketplace helpful?

3:24 PM 2023-08-16

Home > Marketplace >

Create an Azure Databricks workspace

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * [Create new](#)

Instance Details

Workspace name *

Region *

Pricing Tier *

Managed Resource Group name

fill up the required fields

select the option to create

Review + create < Previous Next : Networking >

31°C 3:29 PM 2023-08-16

2. Now the data bricks account got created .

The screenshot shows the Microsoft Azure portal interface. The main page displays a deployment status for a resource group named 'Kanini_Santhosh'. A green checkmark indicates 'Deployment succeeded' with the message: 'Deployment 'Kanini_Santhosh_kaninipracticedatabricks' to resource group 'Kanini_Santhosh' was successful.' Below this, there's a summary of the deployment details: Deployment name: Kanini_Santhosh_kaninipracticedatabricks, Subscription: Azure for Students, Resource group: Kanini_Santhosh. The deployment started at 8/16/2023, 3:31:31 PM with a Correlation ID: e02f5a7b-bedd-48ce-8469-a9970ded76c8. On the left sidebar, there are tabs for Overview, Inputs, Outputs, and Template. A 'Deployment details' section is expanded, showing the deployment status. A 'Next steps' section is collapsed. At the bottom, there are buttons for 'Go to resource' and 'Give feedback'. A tooltip 'once databricks created, launch workspace' points to the 'Launch Workspace' button, which is highlighted with a red circle.

3. Now launch the workspace of databricks .

The screenshot shows the Microsoft Azure portal interface. The main page displays the overview of a Databricks workspace named 'kaninipracticedatabricks'. The workspace is listed under the 'Azure Databricks Service' category. The 'Overview' tab is selected. The workspace status is 'Active'. It is located in the 'South India' region and is part of the 'Kanini_Santhosh' resource group. The workspace is associated with the 'Azure for Students' subscription and has a subscription ID of 0d7bf469-f8f5-47be-843f-603b1ee06366. There are tags listed as '(edit)' and 'Add tags'. To the right of the workspace details, there is a JSON View link. Below the workspace details, there is a large Databricks logo icon. Underneath the icon, there are two buttons: 'Launch Workspace' (highlighted with a red circle) and 'Upgrade to Premium'. A red callout arrow points from the text 'once databricks created, launch workspace' to the 'Launch Workspace' button. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time as 2023-08-16 3:35 PM.

4. Create a Note book for working on the data :

The screenshot shows the Microsoft Azure Databricks workspace interface. On the left sidebar, under the 'New' tab, the 'Workspace' option is highlighted with a red circle. A red arrow points from this circle to the text 'Select the workspace'. In the main workspace area, there is a 'Users' folder under 'Workspace'. A red circle highlights the 'Create' option in the context menu for this folder. A red arrow points from this circle to the text '2. Select the users'. Another red circle highlights the 'Notebook' option in the same context menu. A red arrow points from this circle to the text '3. right click on it and create a notebook'.

5. Once you create a notebook , create sample program and run the Program , You will get a error :

The screenshot shows a Databricks notebook titled 'PracticeDataBrics' in Python. The code cell contains the following Python code: `print("Hello , Santhosh !!")`. A red circle highlights this code. A red arrow points from this circle to the text 'Rename the Document and Run a Sample Program'. The top navigation bar includes 'File', 'Edit', 'View', 'Run', 'Help', and 'Last edit was now'.

6. Now when you run the program, you will get a error , so click on the “Go To Compute”.

The screenshot shows a Microsoft Azure Databricks workspace. On the left, there's a sidebar with various options like Workspace, Recents, Data, Workflows, Compute, SQL, Data Engineering, Job Runs, Data Ingestion, Delta Live Tables, Machine Learning, Experiments, Features, Models, and Serving. The main area shows a notebook titled 'PracticeDataBrics' with a Python command cell. A modal dialog box is centered over the workspace, displaying the message: "No compute resource available. You do not have an available compute resource to which this notebook can be attached. Please go to the 'Compute' page to create a resource or contact your administrator." At the bottom right of the modal is a blue button labeled "Go to Compute". A red arrow points from the text "click on go to compute" to this button, which is also circled in red.

7. Now fill up the mandatory required field as mentioned in the picture .

The screenshot shows the 'Create Cluster' wizard in Databricks. The left sidebar includes options like New, Workspace, Recents, Data, Workflows, Compute, SQL, Data Engineering, Job Runs, Data Ingestion, Delta Live Tables, Machine Learning, Experiments, Features, Models, Serving, Marketplace, Partner Connect, Disable new UI, Provide feedback, and Collapse menu. The main form is titled 'SANTHOSH N's Cluster'. It has sections for 'Compute', 'UI preview', and 'Provide feedback'. Under 'Compute', there are fields for 'Multi node' (radio button), 'Single node' (radio button, selected and highlighted with a red circle), 'Access mode' (dropdown set to 'Single user access'), 'Runtime' (dropdown set to '12.2 LTS (Scala 2.12, Spark 3.3.2)', highlighted with a red circle), 'Use Photon Acceleration' (checkbox checked), 'Node type' (dropdown set to 'Standard_DS3_v2', highlighted with a red circle), and 'Terminate after' (input field set to '20 minutes of inactivity', highlighted with a red circle). To the right, a 'Summary' section shows '1 Driver', '14 GB Memory, 4 Cores', 'Runtime: 12.2.x-scala2.12', and 'Photon Standard_DS3_v2 1.5 DBU/h'. Red text on the right side of the cluster form says: "Now mandatorily select only this options for computation process and select create compute process". Below this, another red text says: "*** - Mandatory steps to review".

SANTHOSH N's Cluster ✓

Once completed u will get verification

Summary

1 Driver 14 GB Memory, 4 Cores
Runtime 12.2.x-scala2.12
Photon Standard_DS3_v2 1.5 DBU/h

Performance

Databricks Runtime Version 12.2 LTS (includes Apache Spark 3.3.2, Scala 2.12)

Use Photon Acceleration

Node type Standard_DS3_v2 14 GB Memory, 4 Cores

Terminate after 20 minutes of inactivity

Tags

No custom tags

Advanced options

Type here to search

31°C 4:05 PM 2023-08-16

8. Now Enable the “DBFS File Browser” :

User Settings

Admin Settings

Azure Portal Admin Settings

Manage Account

Log out

Go to Profile and click on it

Select the admin Settings

https://adb-1609733603075642.2.azure.databricks.net/?o=1609733603075642#setting/accounts

1 20 / page

9. Inside the admin setting click on the workspace settings :

The screenshot shows the Databricks Admin Settings interface. On the left, there's a sidebar with various navigation options like Workspace, Recents, Data, Workflows, Compute, etc. The main area is titled 'Admin Settings' and has tabs for Users, Service principals, Groups, Global init scripts, **Workspace settings**, and Notification destinations. A red circle highlights the 'Workspace settings' tab. Below it, there's a table for managing users, showing columns for Username, Name, Status, Admin, and Allow cluster creation. A specific user entry is selected, showing the email 201910100483@presidencyuniversity.in. A red arrow points from the 'Workspace settings' tab to this user entry with the text: "now inside the admin setting, navigate to the workspace settings or click on it !!".

10. Now scroll down to the option “DBFS FILE BROWSER” and “ENABLE THE OPTION” :

The screenshot shows the 'Admin Settings' page with the 'Advanced' section expanded. Under 'Advanced', there's a list of options with toggle switches. One option, 'DBFS File Browser: Enabled', has a red circle around its toggle switch. A red arrow points from this circle to another red circle on the same row, with the text 'Enable this option'. To the right of the list, a blue box displays a success message: 'DBFS File Browser Enabled' and 'You must refresh the page for this change to take effect.' The bottom of the screen shows the Windows taskbar with various icons.

11. Now go to "Data" inside Databricks and Select the "Browse DBFS" :

The screenshot shows the Databricks Data Explorer interface. On the left sidebar, the 'Data' option is highlighted with a red circle and a red arrow pointing to it from the text '1. Select the "Data" option in Data bricks'. At the top right, there is a 'Browse DBFS' button, which is also circled in red with a red arrow pointing to it from the text 'Click on Browse DBFS'. The main area shows a 'default' database with a 'Tables' tab and a message stating 'You either don't have access to any tables or there are no tables in this schema. Please contact your administrator.'

12. Now once DBFS is opened , click on the upload button to upload the datasets :

The screenshot shows the Databricks DBFS interface. On the left sidebar, the 'Data' option is highlighted with a red circle and a red arrow pointing to it from the text '1. Select the "Data" option in Data bricks'. In the center, there is an 'Upload' button, which is circled in red with a red arrow pointing to it from the text 'Now upload the required datasets for process as per requirement'. The main area shows a 'FileStore/tables' list with a 'Prefix search' bar and a message stating 'You either don't have access to any tables or there are no tables in this schema. Please contact your administrator.'

The screenshot shows the Azure Databricks interface. On the left, there's a sidebar with various options like Workspace, Data, Workflows, Compute, and SQL. The main area is titled 'DBFS' and shows a list of files in '/FileStore/tables'. A modal window titled 'Upload Data to DBFS' is open, showing a file named 'countries_mu' selected. The file size is 83.3 KB. Below the file list, it says 'File uploaded to /FileStore/tables/countries_mu.json'. There's a 'Done' button at the bottom right of the modal.

13. Once uploaded we can check the dataset using file path as below :

The screenshot shows the Azure Databricks workspace. On the left, there's a sidebar with options like Workspace, Data, Workflows, Compute, and SQL. The main area is titled 'PracticeDataBrics' and shows a Python notebook. The code in the notebook reads a CSV file from DBFS and displays its contents. The output shows the first few rows of the 'countries' dataset.

COUNTRY_ID	NAME	NATIONALITY	COUNTRY_CODE	ISO_ALPHA2	CAPITAL
1	Afghanistan	Afghan	AFG	AF	Kabul
2	Albania	Albanian	ALB	AL	Tirana
3	Algeria	Algerian	DZA	DZ	Algiers
4	American Samoa	American Samoan	ASM	AS	Pago Pago
5	Andorra	Andorran	AND	AD	Andorra la Vella
6	Angola	Angolan	AGO	AO	Luanda
7	Anguilla	Anguillan	AIA	AI	The Valley

MEDALLION ARCHITECTURE

Medallion Architecture is a system for logically organising data within a Data Lakehouse.

A standard medallion architecture consists of 3 main layers, in order: Bronze, Silver and Gold.

The increasing quality of precious metal in the names is no accident and represents an increasing level of structure and validation when moving through the layers. This architecture is sometimes also known as multi-hop architecture.



Creating the Datalake In AZURE:

1. Go inside the Create resource group and select the “Create” option.

The screenshot shows the Microsoft Azure Resource Groups page for a resource group named "Kanini_Santhosh". A red circle highlights the "+ Create" button in the top navigation bar. Below it, a red arrow points to the "Essentials" section, which is labeled "Select this option". The main content area displays a message: "No resources match your filters" with a "Try changing or clearing your filters." link. The left sidebar contains links for Overview, Activity log, Access control (IAM), Tags, Resource visualizer, Events, Settings, Deployments, Security, Deployment stacks, Policies, Properties, Locks, and Cost Management. The bottom of the screen shows a Windows taskbar with various pinned icons and system status information.

The screenshot shows the Microsoft Azure Marketplace search results for "storage account". A red circle highlights the search input field containing "storage account". Below it, a red arrow points to the "Create Storage Account" button in the "Storage account" listing. The search filters at the top include "Pricing : All", "Operating System : All", "Publisher Type : All", "Product Type : All", and "Publisher name : All". The results show 1 to 20 of 173 items. The "Storage account" listing includes details for Microsoft Azure Service, Storage Account Using ARM Template, Azure Storage Mover, APEX Protection Storage for Microsoft Azure (DDVE), Storage Account Using ARM, and S3 API for Azure Blob Storage (Flexify.IO). The bottom of the screen shows a Windows taskbar with various pinned icons and system status information.

2. Now fill up the required fields to create storage account and select "Standard " for performance .

Microsoft Azure

Create a storage account

Basics Advanced Networking Data protection Encryption Tags Review

Default to Azure Active Directory authorization in the Azure portal

Minimum TLS version

Permitted scope for copy operations (preview)

Hierarchical Namespace

Hierarchical namespace, complemented by Data Lake Storage Gen2 endpoint, enables file and directory semantics, accelerates big data analytics workloads, and enables access control lists (ACLs) [Learn more](#)

Enable hierarchical namespace **Enable Hierarchical namespace**

Now click on "Review" and Select "Create" to create datalake

Review < Previous Next : Networking >

kaninilakedata_1692186371302 | Overview

Deployment

Search Delete Cancel Redeploy Download Refresh

Overview Inputs Outputs Template

Your deployment is complete

Deployment name: kaninilakedata_1692186371302
Subscription: Azure for Students
Resource group: Kanini_Santhosh

Start time: 8/16/2023, 5:16:14 PM
Correlation ID: 0042ba5f-6829-4f82-a097-38a31d6d2272

Deployment details Next steps Go to resource

Give feedback Tell us about your experience with deployment

Cost Management
Get notified to stay within your budget and prevent unexpected charges on your bill.
Set up cost alerts >

Microsoft Defender for Cloud
Secure your apps and infrastructure
Go to Microsoft Defender for Cloud >

Free Microsoft tutorials
Start learning today >

Work with an expert
Azure experts are service provider partners

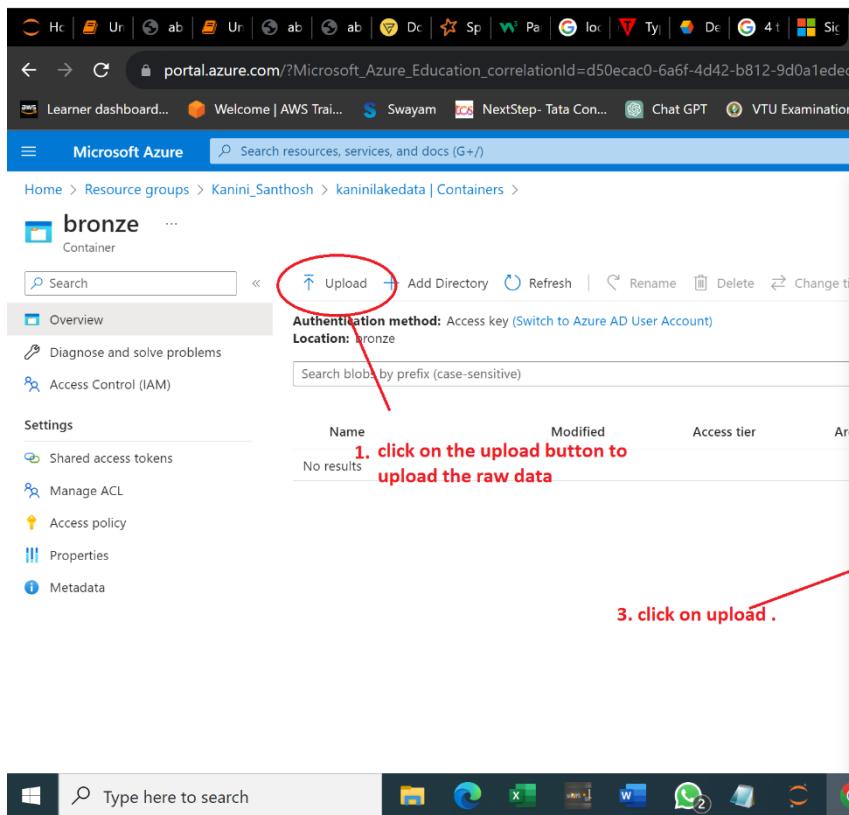
31°C 5:19 PM 2023-08-16

3. Now we need to create 3 containers : BRONZE,SILVER,GOLD

The screenshot shows the Microsoft Azure portal interface for a storage account named 'kaninilakedata'. On the left sidebar, under 'Data storage', the 'Containers' option is selected. A red circle highlights this selection. In the main content area, there is a table listing existing containers: '\$logs' (Last modified: 8/16/2023, 5:16:44 PM, Public access level: Private). A red circle highlights the '+ Container' button at the top left of the table. A red arrow points from this button to the text '2. Click on "Container" to create a new container'. To the right of the table, a 'New container' dialog box is open. It has a 'Name' field containing 'bronze', which is also highlighted with a red circle. A red arrow points from this field to the text '3. name your containers'. Below the name field is a dropdown for 'Public access level' set to 'Private (no anonymous access)'. At the bottom right of the dialog is a blue 'Create' button, which is also highlighted with a red circle. A red arrow points from this button to the text '4. click on "create"'. The status bar at the bottom of the browser window shows the date and time as '2023-08-16 5:22 PM'.

The screenshot shows the same Microsoft Azure Storage account interface after the containers have been created. The 'Containers' table now includes three additional rows: 'bronze' (Last modified: 8/16/2023, 5:26:00 PM), 'gold' (Last modified: 8/16/2023, 5:26:12 PM), and 'silver' (Last modified: 8/16/2023, 5:26:07 PM). A red box highlights the first three rows ('bronze', 'gold', 'silver'). A red arrow points from this box to the text 'three containers created'. In the top right corner of the interface, a success message box is displayed with the text 'Successfully created storage container' and 'Successfully created storage container 'gold''. The status bar at the bottom of the browser window shows the date and time as '2023-08-16 5:22 PM'.

4. Now upload the data inside the bronze container :



1. click on the upload button to upload the raw data

Upload blob

5 file(s) selected: customers.csv, order_items.csv, orders.csv...

Drag and drop files here or [Browse for files](#)

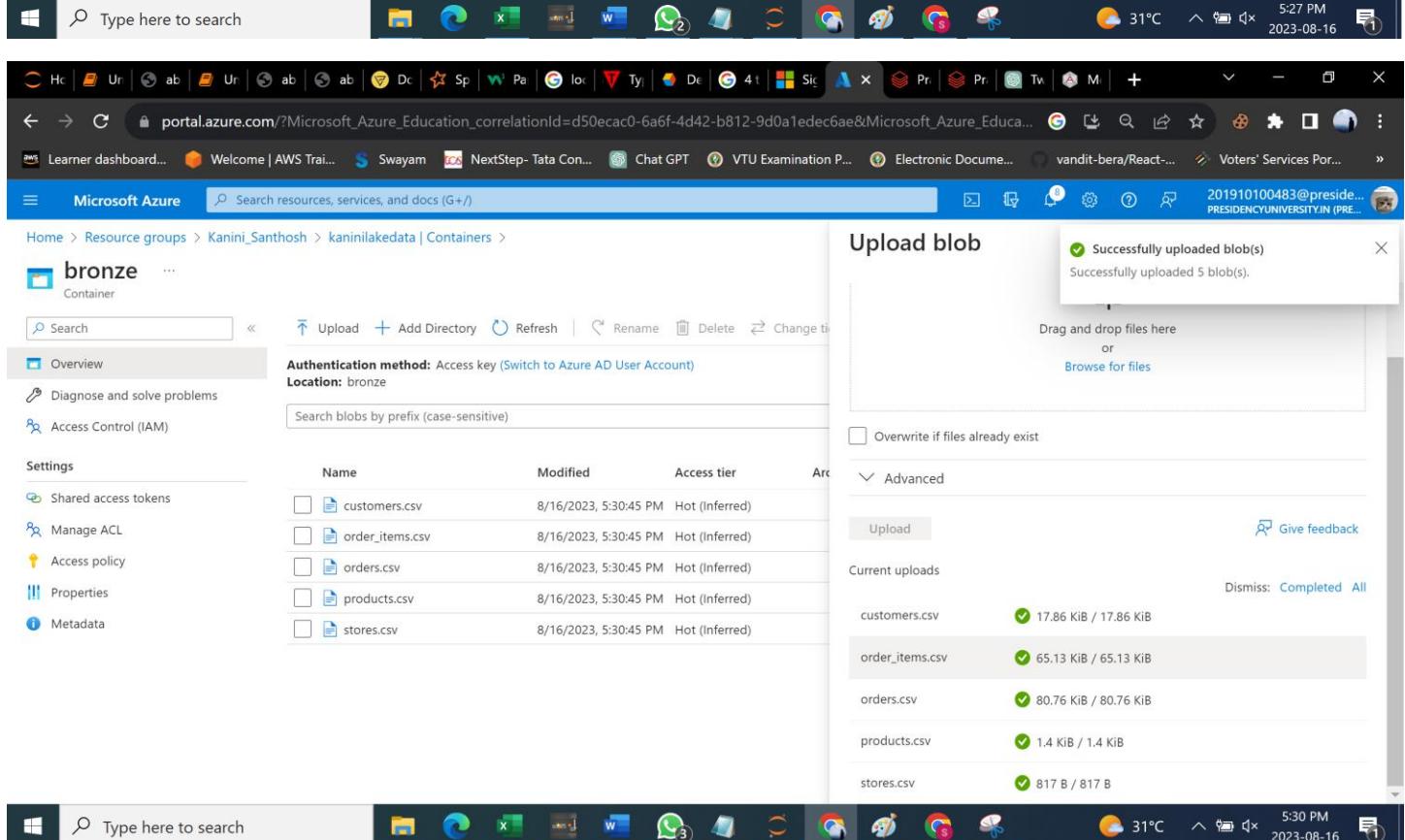
Overwrite if files already exist

Advanced

Upload

2. Browse files and upload the files

3. click on upload .



Successfully uploaded blob(s)
Successfully uploaded 5 blob(s).

Drag and drop files here or [Browse for files](#)

Overwrite if files already exist

Advanced

Upload

Current uploads

customers.csv	✓ 17.86 KiB / 17.86 KiB
order_items.csv	✓ 65.13 KiB / 65.13 KiB
orders.csv	✓ 80.76 KiB / 80.76 KiB
products.csv	✓ 1.4 KiB / 1.4 KiB
stores.csv	✓ 817 B / 817 B

Dismiss: [Completed](#) [All](#)

5. Now Browse back to the DataBricks and create two folders inside the DBFS and upload the given dataset :

DBFS

/FileStore/tables

Prefix search

bronze

silver

bronze

silver

countries_multiline.json

countries_single_line.json

countries.csv

countries.txt

country_regions.csv

customers.csv

order_items.csv

orders.csv

products.csv

stores.csv

2 folders Created

Create folder

Upload here

No data to display, active cluster or warehouse is required

hive metastore

DBFS

/FileStore/tables/bronze

bronze

silver

customers.csv

order_items.csv

orders.csv

products.csv

stores.csv

bronze

silver

bronze

silver

countries_multiline.json

countries_single_line.json

countries.csv

countries.txt

country_regions.csv

customers.csv

order_items.csv

orders.csv

products.csv

stores.csv

Dataset Uploaded

No data to display, active cluster or warehouse is required

hive metastore

6. Now use the workspace to clean and convert the data into parquet and to store it into silver folder for further process :

```
print("Hello , Santhosh !!")
Hello , Santhosh !!
countries_df =
spark.read.csv("dbfs:/FileStore/tables/countries.csv",header=True)
countries_df.display()
import pandas as pd

# Load the CSV file into a DataFrame
data = pd.read_csv("/dbfs/FileStore/tables/countries.csv",
encoding='ISO-8859-1')

print(data.columns)

# Display the first few rows of the DataFrame
print(data.head())

# Filtering
filtered_data = data[data['POPULATION'] > 100000000]

# Grouping and Aggregation
grouped_data = data.groupby('Region').agg({
    'POPULATION': 'sum',
    'Area': 'mean',
    'GDP': 'sum'
}).reset_index()

# Sorting
sorted_data = data.sort_values(by='Population', ascending=False)

# Adding Calculated Columns
data['Population_Density'] = data['Population'] / data['Area']

# Display the first few rows after adding calculated columns
print(data.head())

# Data Enrichment (hypothetical data, not present in the original
# file)
enrichment_data = pd.DataFrame({
    'Country': ['Country1', 'Country2', 'Country3'],
    'GDP_Per_Capita': [1500, 2500, 1800]
})
merged_data = data.merge(enrichment_data, on='Country', how='left')

# Time Series Analysis (hypothetical time-related data)
time_series_data = pd.DataFrame({})
```

```

'Date': pd.date_range(start='2023-01-01', periods=len(data)),
'Country': data['Country'],
'New_Cases': [500, 200, 50, ...] # Example COVID-19 new cases
data
})
merged_data = data.merge(time_series_data, on=['Country', 'Date'],
how='left')

# Save the transformed data to new CSV files
filtered_data.to_csv('/dbfs/FileStore/tables/filtered_data.csv',
index=False)
grouped_data.to_csv('/dbfs/FileStore/tables/grouped_data.csv',
index=False)
sorted_data.to_csv('/dbfs/FileStore/tables/sorted_data.csv',
index=False)
data.to_csv('/dbfs/FileStore/tables/with_calculated_columns.csv',
index=False)
merged_data.to_csv('/dbfs/FileStore/tables/enriched_data.csv',
index=False)

Index(['COUNTRY_ID', 'NAME', 'NATIONALITY', 'COUNTRY_CODE',
'ISO_ALPHA2',
'CAPITAL', 'POPULATION', 'AREA_KM2', 'REGION_ID',
'SUB_REGION_ID',
'INTERMEDIATE_REGION_ID', 'ORGANIZATION_REGION_ID'],
dtype='object')
COUNTRY_ID      NAME    NATIONALITY COUNTRY_CODE ISO_ALPHA2
0             1 Afghanistan      Afghan       AFG        AF
1             2     Albania      Albanian       ALB        AL
2             3     Algeria      Algerian       DZA        DZ
3             4 American Samoa American Samoan      ASM        AS
4             5     Andorra     Andorran       AND        AD

SUB_REGION_ID   CAPITAL  POPULATION  AREA_KM2  REGION_ID
0            Kabul  38041754  652230.0      30    30.0
1           Tirana  2880917   28748.0      20    70.0
2          Algiers  43053054 2381741.0      50    40.0

```

```
3      Pago Pago      55312    199.0      40      20.0
4  Andorra la Vella      77142    468.0      20      70.0
```

```
INTERMEDIATE_REGION_ID  ORGANIZATION_REGION_ID
0                      NaN                  30
1                      NaN                  20
2                      NaN                  20
3                      NaN                  30
4                      NaN                  20
```

```
-----
-----
KeyError                                Traceback (most recent call
last)
File <command-171310774024009>:16
  13 filtered_data = data[data['POPULATION'] > 100000000]
  15 # Grouping and Aggregation
---> 16 grouped_data = data.groupby('Region').agg({
  17     'POPULATION': 'sum',
  18     'Area': 'mean',
  19     'GDP': 'sum'
  20 }).reset_index()
  22 # Sorting
  23 sorted_data = data.sort_values(by='Population',
ascending=False)

File
/databricks/python/lib/python3.9/site-packages/pandas/core/frame.py:77
12, in DataFrame.groupby(self, by, axis, level, as_index, sort,
group_keys, squeeze, observed, dropna)
  7707 axis = self._get_axis_number(axis)
  7709 # https://github.com/python/mypy/issues/7642
  7710 # error: Argument "squeeze" to "DataFrameGroupBy" has
incompatible type
  7711 # "Union[bool, NoDefault]"; expected "bool"
-> 7712 return DataFrameGroupBy(
  7713     obj=self,
  7714     keys=by,
  7715     axis=axis,
  7716     level=level,
  7717     as_index=as_index,
  7718     sort=sort,
  7719     group_keys=group_keys,
  7720     squeeze=squeeze, # type: ignore[arg-type]
  7721     observed=observed,
  7722     dropna=dropna,
  7723 )
```

```

File
/databricks/python/lib/python3.9/site-packages/pandas/core/groupby/
groupby.py:882, in GroupBy.__init__(self, obj, keys, axis, level,
grouper, exclusions, selection, as_index, sort, group_keys, squeeze,
observed, mutated, dropna)
    879 if grouper is None:
    880     from pandas.core.groupby.grouper import get_grouper
--> 882     grouper, exclusions, obj = get_grouper(
    883         obj,
    884         keys,
    885         axis=axis,
    886         level=level,
    887         sort=sort,
    888         observed=observed,
    889         mutated=self.mutated,
    890         dropna=self.dropna,
    891     )
    893 self.obj = obj
    894 self.axis = obj._get_axis_number(axis)

File
/databricks/python/lib/python3.9/site-packages/pandas/core/groupby/
grouper.py:882, in get_grouper(obj, key, axis, level, sort, observed,
mutated, validate, dropna)
    880     in_axis, level, gpr = False, gpr, None
    881     else:
--> 882         raise KeyError(gpr)
    883 elif isinstance(gpr, Grouper) and gpr.key is not None:
    884     # Add key to exclusions
    885     exclusions.add(gpr.key)

KeyError: 'Region'

from pyspark.sql.types import IntegerType, StringType, DoubleType,
StructField, StructType

orders_path = "/FileStore/tables/bronze/orders.csv"

orders_schema = StructType([
    StructField("ORDER_ID", IntegerType(), False),
    StructField("ORDER_DATETIME", StringType(), False),
    StructField("CUSTOMER_ID", IntegerType(), False),
    StructField("ORDER_STATUS", StringType(), False),
    StructField("STORE_ID", IntegerType(), False)
])

orders=spark.read.csv(path=orders_path,header=True,schema=orders_schema)

```

```

from pyspark.sql.functions import to_timestamp

orders = orders.select('ORDER_ID', \
    to_timestamp(orders['order_datetime'], "dd-MMM-yy \
kk.mm.ss.SS").alias('ORDER_TIMESTAMP'), \
    'CUSTOMER_ID', \
    'ORDER_STATUS', \
    'STORE_ID'
)

# filtering the records to display only 'COMPLETE' orders
# assigning the result back to the orders dataframe
orders = orders.filter(orders['order_status']=="COMPLETE")

stores_path = "/FileStore/tables/bronze/stores.csv"

stores_schema = StructType([
    StructField("STORE_ID", IntegerType(), False),
    StructField("STORE_NAME", StringType(), False),
    StructField("WEB_ADDRESS", StringType(), False),
    StructField("LATITUDE", DoubleType(), False),
    StructField("LONGITUDE", DoubleType(), False)
]
)
stores=spark.read.csv(path=stores_path, header=True,
schema=stores_schema)

# joining the orders and stores via a 'left' join, the orders table is
# the left table.
# this operation adds the store_name to the orders dataframe
# the final operation is a select method to select only the required
# columns and assign it back to the orders dataframe

orders = orders.join(stores, orders['store_id']==stores['store_id'],
'left').select('ORDER_ID', 'ORDER_TIMESTAMP', 'CUSTOMER_ID',
'STORE_NAME')

# writing the orders dataframe as a parquet file in the silver layer,
# should use mode = 'overwrite' in this instance
orders.write.parquet("/FileStore/tables/silver/orders",
mode='overwrite')

order_items_path = "/FileStore/tables/bronze/order_items.csv"

order_items_schema = StructType([
    StructField("ORDER_ID", IntegerType(), False),
    StructField("LINE_ITEM_ID", IntegerType(), False),
    StructField("PRODUCT_ID", IntegerType(), False),
    StructField("UNIT_PRICE", DoubleType(), False),
    StructField("QUANTITY", IntegerType(), False)
]
)

```

```
)  
  
order_items=spark.read.csv(path=order_items_path, header=True,  
schema=order_items_schema)  
  
order_items = order_items.drop('LINE_ITEM_ID')  
  
order_items.write.parquet("/FileStore/tables/silver/order_items",  
mode='overwrite')  
  
products_path = "/FileStore/tables/bronze/products.csv"  
  
products_schema = StructType([  
    StructField("PRODUCT_ID", IntegerType(), False),  
    StructField("PRODUCT_NAME", StringType(), False),  
    StructField("UNIT_PRICE", DoubleType(), False)  
])  
  
products=spark.read.csv(path=products_path, header=True,  
schema=products_schema)  
  
# writing the parquet file  
products.write.parquet('/FileStore/tables/silver/products',  
mode='overwrite')  
  
customers_path = "/FileStore/tables/bronze/customers.csv"  
  
customers_schema = StructType([  
    StructField("CUSTOMER_ID", IntegerType(), False),  
    StructField("FULL_NAME", StringType(), False),  
    StructField("EMAIL_ADDRESS", StringType(), False)  
])  
  
customers=spark.read.csv(path=customers_path, header=True,  
schema=customers_schema)  
  
# writing the parquet file  
customers.write.parquet('/FileStore/tables/silver/customers',  
mode='overwrite')  
  
from pyspark.sql.functions import *  
  
orders = spark.read.parquet('/FileStore/tables/silver/orders')  
order_items =  
spark.read.parquet('/FileStore/tables/silver/order_items')
```

```

products = spark.read.parquet('/FileStore/tables/silver/products')
customers = spark.read.parquet('/FileStore/tables/silver/customers')

order_details = orders.select(
    'ORDER_ID',
    to_date('order_timestamp').alias('DATE'),
    'CUSTOMER_ID',
    'STORE_NAME'
)

# joining the order_details and order_items dataframe on the
'order_id' column of both tabs
# selecting the relevant columns from the resulting dataframes and
storing it back to the order_details variable
order_details = order_details.join(order_items,
    order_items['order_id']==order_details['order_id'], 'left'). \
    select(order_details['ORDER_ID'], order_details['DATE'],
    order_details['CUSTOMER_ID'], order_details['STORE_NAME'],
    order_items['UNIT_PRICE'], order_items['QUANTITY'])

# creating a total amount column at the record level
order_details = order_details.withColumn('TOTAL_SALES_AMOUNT',
    order_items['UNIT_PRICE']*order_details['QUANTITY'])

# grouping the order_details dataframe and taking the sum of the total
amount, renaming this to 'TOTAL_ORDER_AMOUNT'
# assigning the result back to the order_details dataframe
order_details = order_details. \
    groupBy('ORDER_ID', 'DATE', 'CUSTOMER_ID', 'STORE_NAME'). \
    sum('TOTAL_SALES_AMOUNT'). \
    withColumnRenamed('sum(TOTAL_SALES_AMOUNT)', 'TOTAL_ORDER_AMOUNT')

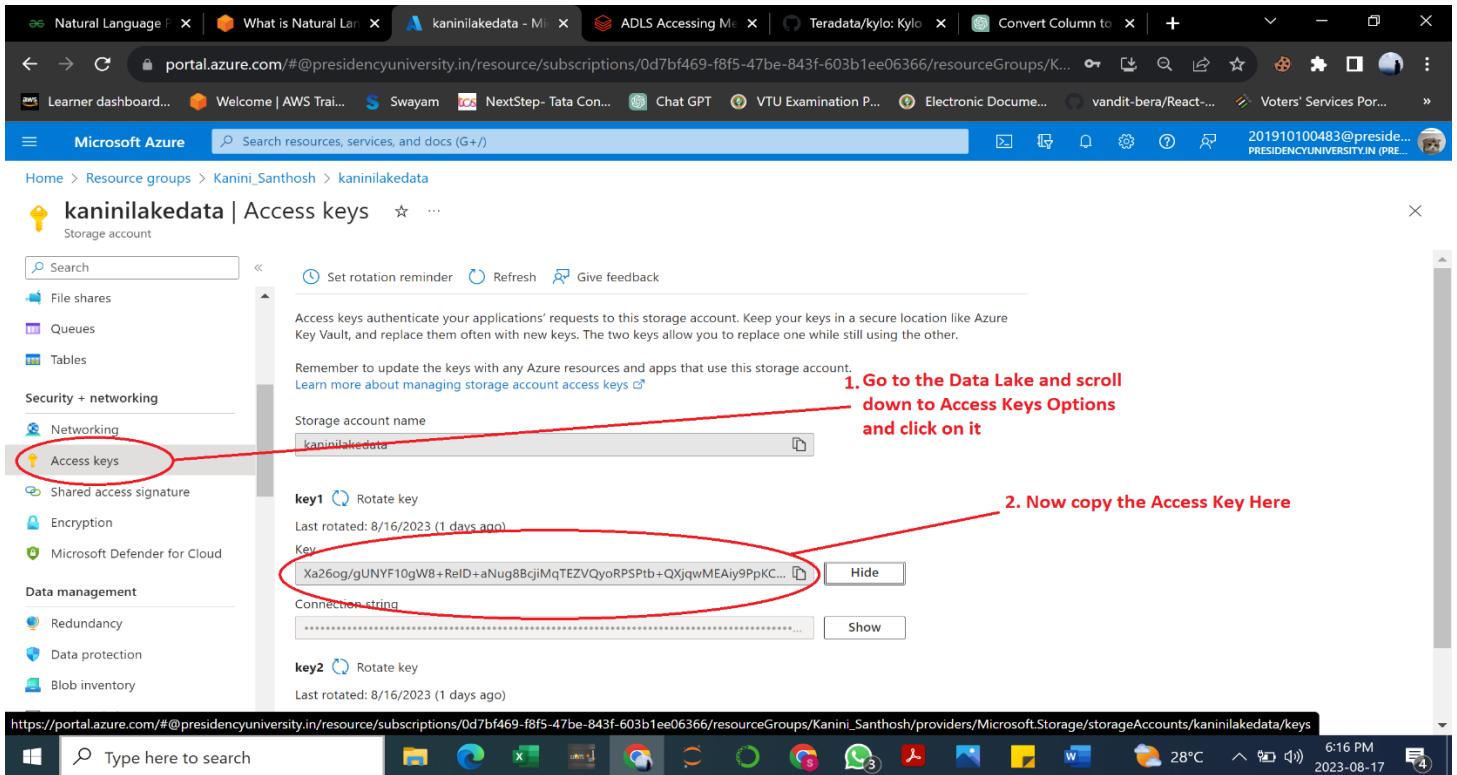
# rounding the TOTAL_ORDER_AMOUNT to 2 dp
order_details = order_details.withColumn('TOTAL_ORDER_AMOUNT',
    round('TOTAL_ORDER_AMOUNT',2))

# writing the order_details dataframe as a parquet file in the gold
layer
order_details.write.parquet('/FileStore/tables/gold/order_details',
    mode='overwrite')

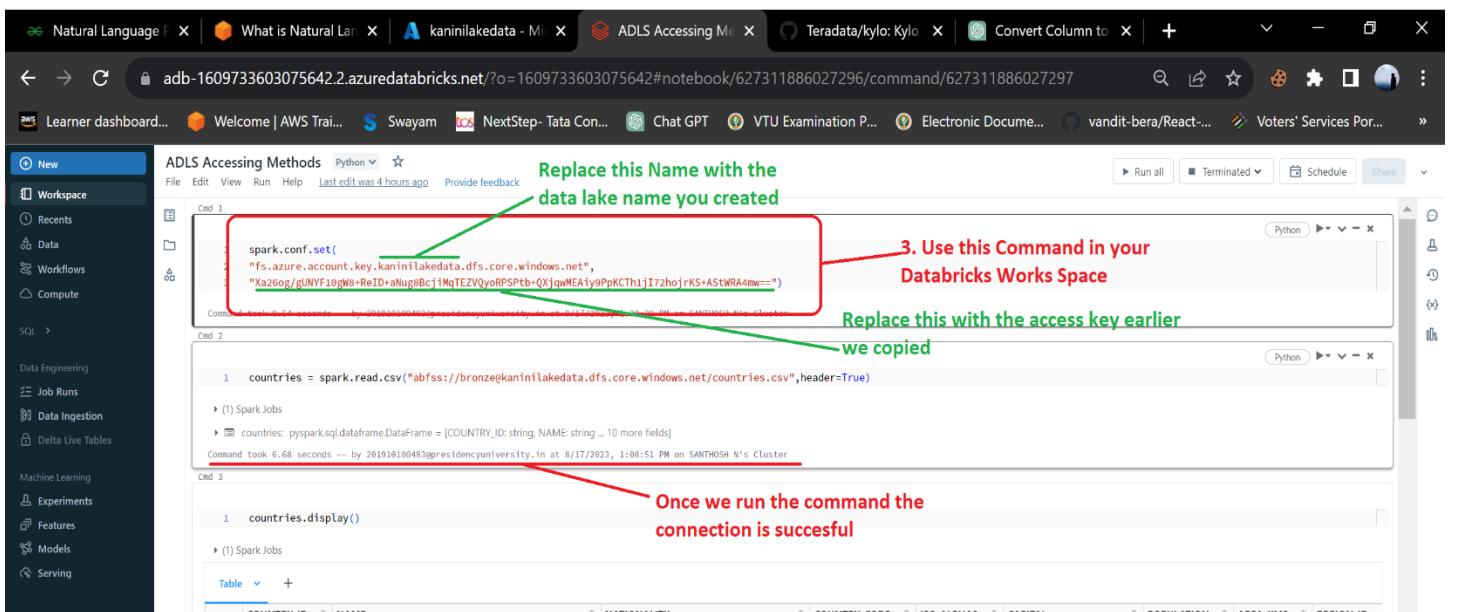
```

CONNECTING DATA STORAGE TO DATA LAKE :

Method 1 : Using Access Keys



The screenshot shows the 'Access keys' section of the Azure Storage account settings. A red circle highlights the 'Access keys' link in the left sidebar. A red arrow points from this link to the text '1. Go to the Data Lake and scroll down to Access Keys Options and click on it'. Another red arrow points from the 'key1' key value to the text '2. Now copy the Access Key Here'. The URL in the browser is https://portal.azure.com/#@presidencyuniversity.in/resource/subscriptions/0d7bf469-f8f5-47be-843f-603b1ee06366/resourceGroups/Kanini_Santhosh/providers/Microsoft.Storage/storageAccounts/kaninilakedata/keys.



The screenshot shows a Databricks workspace with three code cells. The first cell contains the command: spark.conf.set("fs.azure.account.key.kaninilakedata.dfs.core.windows.net", "Xa26og/gUNYF10gW8+ReID+aNug8BcjIMqTEZVQyoRSPt+QXjqwMEAiy9PpKTh1jI72hojrKS+ASTWRA4mw=="). A green bracket highlights this command with the text 'Replace this Name with the data lake name you created'. A red arrow points from this bracket to the text '3. Use this Command in your Databricks Works Space'. The second cell shows the output of the command: countries = spark.read.csv("abfss://bronze@kaninilakedata.dfs.core.windows.net/countries.csv", header=True). A green bracket highlights the file path with the text 'Replace this with the access key earlier we copied'. A red arrow points from this bracket to the text 'Once we run the command the connection is successful'. The third cell shows the output: countries.display(). A red arrow points from this cell to the text 'Once we run the command the connection is successful'.

Code for Using Access Keys :

```
1. spark.conf.set(  
    "fs.azure.account.key.kaninilakedata.dfs.core.windows.net",  
    "Xa26og/gUNYF10gW8+ReID+aNug8BcjIMqTEZVQyoRSPt+QXjqwMEAiy9PpKTh1jI72hojrKS+ASTWRA4mw  
    ==")  
2. countries =  
    spark.read.csv("abfss://bronze@kaninilakedata.dfs.core.windows.net/countries.csv", header=T  
rue)
```

Method 2 : By Generating SAS Tokens

The screenshot shows the 'kaninilakedata' storage account's 'Shared access signature' configuration. A red circle highlights the 'Shared access signature' option in the left sidebar. A red arrow labeled '1. Got to the data lake and select Shared access signature option' points to this circle. Another red circle highlights the 'Service', 'Container', and 'Object' checkboxes under 'Allowed resource types'. A red arrow labeled '2. Enable this 3 options' points to this circle. The 'Start' and 'End' times for the SAS token are set to 08/17/2023 and 08/18/2023 respectively. The 'Allowed IP addresses' dropdown is expanded, showing several IP ranges.

The screenshot shows the same configuration page after generating a SAS token. A red circle highlights the 'Generate SAS and connection string' button. A red arrow labeled '3. Scroll Down and Click on the" Generate SAS " Button' points to this button. A red circle highlights the 'SAS token' field, which contains a long URL starting with '?sv=2022-11-02&ss=bfq...'. A red arrow labeled '4. Copy the SAS TOKEN and Remove the "?" at the beginning of the file.' points to this field.

Replace the Data Lake name

```

1 spark.conf.set("fs.azure.account.auth.type.adfgen21608.dfs.core.windows.net", "SAS")
2 spark.conf.set("fs.azure.sas.token.provider.type.adfgen21608.dfs.core.windows.net", "org.apache.hadoop.fs.azurebfs.sas.FixedSASTokenProvider")
3 spark.conf.set("fs.azure.sas.fixed.token.kaninilakedata.dfs.core.windows.net", "sv=2022-11-02&ss=bfqt&srt=sco&sp=rwldacupyx&se=2023-08-17T15:49:42Z&st=2023-08-17T07:49:42Z&spr=https&sig=fBINjB5Y%2FmpSFDeCFT%2BySk4bqYpRocu1gUb1wVU%2BM%3D")

```

Use the Copied SAS Token Here

5.Now Use this code in Databricks work space to access Data storage

Code for Using SAS Tokens :

- ✓

```
spark.conf.set("fs.azure.account.auth.type.adfgen21608.dfs.core.windows.net", "SAS")
```
- ```
spark.conf.set("fs.azure.sas.token.provider.type.adfgen21608.dfs.core.windows.net",
```
- ```
"org.apache.hadoop.fs.azurebfs.sas.FixedSASTokenProvider")
```
- ```
spark.conf.set("fs.azure.sas.fixed.token.kaninilakedata.dfs.core.windows.net", "sv=2022-
```
- ```
11-02&ss=bfqt&srt=sco&sp=rwldacupyx&se=2023-08-17T15:49:42Z&st=2023-08-
```
- ```
17T07:49:42Z&spr=https&sig=fBINjB5Y%2FmpSFDeCFT%2BySk4bqYpRocu1gUb1wVU%2BM%3D")
```
  
- ✓ 

```
countries_region=
```
- ```
spark.read.csv("abfss://bronze@kaninilakedata.dfs.core.windows.net/country_regions.c
```
- ```
sv", header=True)
```

## Method 3 : Using App Registration and Mounting the Containers

1. In your Azure Home Page , Search "APP Registration" and select the option displayed

Microsoft Azure

portal.azure.com/#home

Search bar: app registr...

Services (38) Documentation (99+)

All Azure Active Directory (1) Resources (0) Resource Groups (0)

Marketplace (0)

App registrations

App proxy Function App Application gateways Application groups

Event Grid Partner Registrations

App Configuration

Tutorial: Register a web application in Azure Active Directory B2C ... Create a self-signed public certificate to authenticate your applica...

Configure required Azure AD Graph permissions for an app regist... Connect privately to an App Service apps using private endpoint

Customize and build your mobile app using the wrap wizard - Po... Create a secure ASP.NET MVC 5 web app with log in, email confir...

Azure Key Vault configuration provider in ASP.NET Core

Quickstart: Build your first static web app

Give feedback

Subscription 6 hours ago

Resource group 22 hours ago

26°C 6:49 PM 2023-08-17

2. Now Click on "New Registration"

Microsoft Azure

portal.azure.com/#view/Microsoft\_AAD\_RegisteredApps/ApplicationsListBlade

Home > App registrations

New registration Endpoints Troubleshooting Refresh Download Preview features Got feedback?

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

All applications Owned applications Deleted applications

Start typing a display name or application (client) ID to filter these results Add filters

1 applications found

| Display name ↑↓ | Application (client) ID              | Created on ↑↓ | Certificates & secrets |
|-----------------|--------------------------------------|---------------|------------------------|
| kaniniapp       | 45e145cd-cc19-4979-9e29-2a1d6027191d | 8/17/2023     | Current                |

26°C 6:52 PM 2023-08-17

Natural Language F | What is Natural Lan | Register an applicat | ADLS Accessing Me | Teradata/kylo: Kylo | Convert Column to | +

learner dashboard... Welcome | AWS Trai... Swayam NextStep - Tata Con... Chat GPT VTU Examination P... Electronic Docume... vandit-bera/React-... Voters' Services Por...

Microsoft Azure Search resources, services, and docs (G+)

Home > App registrations >

## Register an application

\* Name  
The user-facing display name for this application (this can be changed later).

Supported account types  
Who can use this application or access this API?  
 Accounts in any organizational directory (Any Azure AD directory - Multitenant)  
 Accounts in this organizational directory only (presidencyuniversity.in only - Single tenant)  
 Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)  
 Personal Microsoft accounts only

Help me choose...

By proceeding, you agree to the Microsoft Platform Policies [View policies](#)

**3. Now give a valid App name**

**4. If you are using a student account use Multitenant**

**5. later, click on the Register button and complete the process**

**Register**

Natural Language F | What is Natural Lan | kaniniapp - Micros | ADLS Accessing Me | Teradata/kylo: Kylo | Convert Column to | +

learner dashboard... Welcome | AWS Trai... Swayam NextStep - Tata Con... Chat GPT VTU Examination P... Electronic Docume... vandit-bera/React-... Voters' Services Por...

Microsoft Azure Search resources, services, and docs (G+)

## kaniniapp

Overview Endpoints Preview features

**Essentials**

|                         |                                        |
|-------------------------|----------------------------------------|
| Display name            | : kaniniapp                            |
| Application (client) ID | : 45e145cd-cc19-4979-9e29-a1d6027191d  |
| Object ID               | : b7cae4d7-aa02-48dd-b68b-6b661a79a8b6 |
| Directory (tenant) ID   | : bf93bb5e-ecf0-4e3d-be0e-79b5cc527a48 |
| Supported account types | : Multiple organizations               |

Client credentials : Add a certificate or secret  
Redirect URIs : Add a Redirect URI  
Application ID URI : Add an Application ID URI  
Managed application in I... : kaniniapp

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

Starting November 9th, 2020 end users will no longer be able to grant consent to newly registered multitenant apps without verified publishers. [Add MPN ID to verify publisher](#)

**7. Click on the "Add a Certificate"**

**6. Now copy this Both Client and directory ID and save it for later process**

**Build your application with the Microsoft identity platform**

Home > App registrations > kaniniapp

kaniniapp | Certificates & secrets

Search

Overview Quickstart Integration assistant

Manage

- Branding & properties
- Authentication
- Certificates & secrets**
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators
- Manifest

Got feedback?

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret).

Certificates (0) Client secrets (0) Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

| Description     | Expires   | Value                            | Secret ID                             |
|-----------------|-----------|----------------------------------|---------------------------------------|
| kaniniappsecret | 2/13/2024 | sR6Q~_euVAZLM6Wc8QKlrZTFPhHR4... | 6bf29dc5-270c-408e-998d-82bb96be5e... |

Add Cancel

**9. Create a Client Secret**  
**Description as per your choice**

Home > App registrations > kaniniapp

kaniniapp | Certificates & secrets

Search

Overview Quickstart Integration assistant

Manage

- Branding & properties
- Authentication
- Certificates & secrets**
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators
- Manifest

Got feedback?

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret).

Client secrets (1) Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

| Description     | Expires   | Value                            | Secret ID                             |
|-----------------|-----------|----------------------------------|---------------------------------------|
| kaniniappsecret | 2/13/2024 | sR6Q~_euVAZLM6Wc8QKlrZTFPhHR4... | 6bf29dc5-270c-408e-998d-82bb96be5e... |

**11. Once Created , copy the value and save it for later process**

Update application credentials  
Successfully updated application kaniniapp credentials

```
1 configs = {"fs.azure.account.auth.type": "OAuth",
2 "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
3 "fs.azure.account.oauth2.client.id": "45e145cd-cc19-4979-9e29-2a1d6827191d",
4 "fs.azure.account.oauth2.client.secret": "07e8q-6a8922q7k3yWrrman3Zoneiyfssdbb7U",
5 "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/bf93bb5e-ecf8-4e3d-bebe-79b5cc527a48/oauth2/token"}
6
7 # Optionally, you can add <directory-name> to the source URI of your mount point.
8 dbutils.fs.mount(
9 source = "abfss://bronze@kaninapp.azuredatalakestorage.dfs.core.windows.net/",
10 mount_point = "/mnt/bronze",
11 extra_configs=configs)
```

Replace the Client ID  
Replace this Secret value  
Replace this Directory ID

12. Now use this code and run the command to mount the dataset

Now the Data is mounted and Bronze folder is created

### **Code for Using App Registration and mounting :**

```
configs = {"fs.azure.account.auth.type": "OAuth",
 "fs.azure.account.oauth.provider.type":
"org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
 "fs.azure.account.oauth2.client.id": "45e145cd-cc19-4979-9e29-2a1d6027191d",
 "fs.azure.account.oauth2.client.secret":
"D7x8Q~6s8922aQ7ZkJyrWrnmam3Z0me1yfsdDbZU",
 "fs.azure.account.oauth2.client.endpoint":
"https://login.microsoftonline.com/bf93bb5e-ecf0-4e3d-be0e-79b5cc527a48/oauth2/token"}

Optionally, you can add <directory-name> to the source URI of your mount point.
dbutils.fs.mount(
 source = "abfss://bronze@kaninilakedata.dfs.core.windows.net/",
 mount_point = "/mnt/bronze",
 extra_configs=configs)
```