Analytics dashboard using Azure Databricks

Introduction:

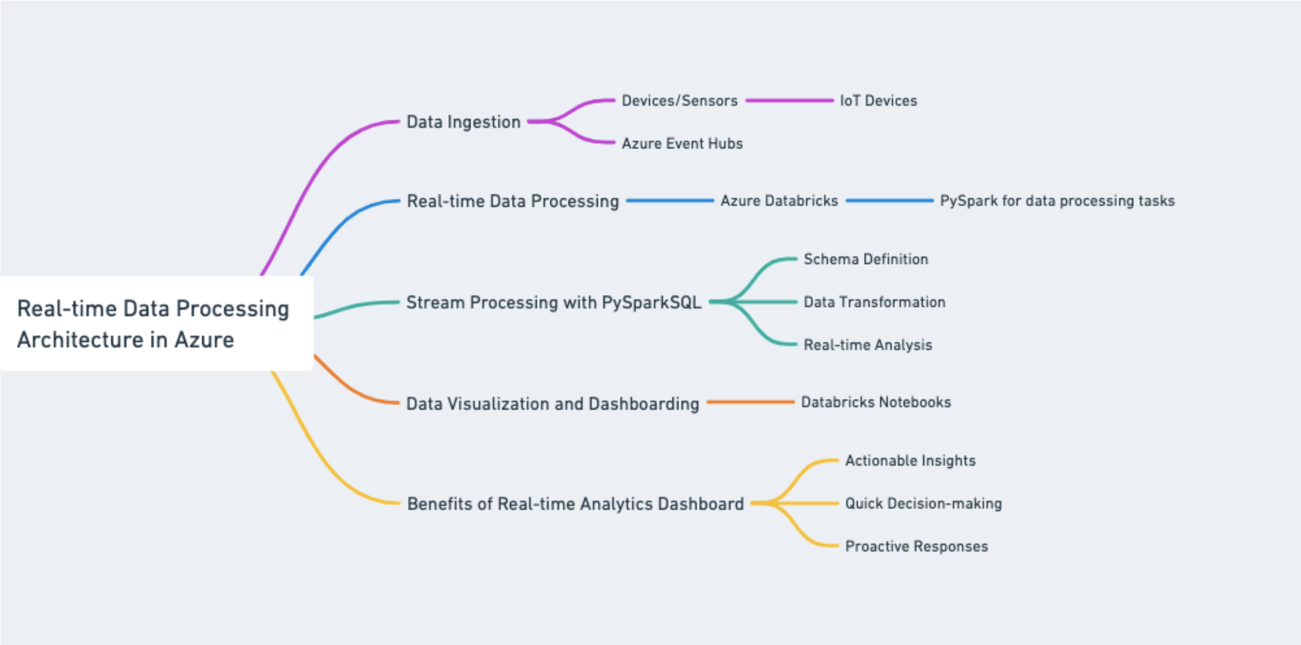
In today's data-driven world, the ability to extract insights from streaming data in real-time is paramount for businesses to stay competitive and responsive. This project focuses on harnessing the power of Azure Databricks and PySpark to create a real-time analytics dashboard. By ingesting streaming data, processing it with PySparkSQL, and visualizing the results dynamically, organizations can gain actionable insights and make informed decisions instantaneously.

The real-time analytics dashboard created through this project not only enables organizations to monitor key metrics and KPIs as they evolve but also empowers stakeholders to respond proactively to changing conditions.

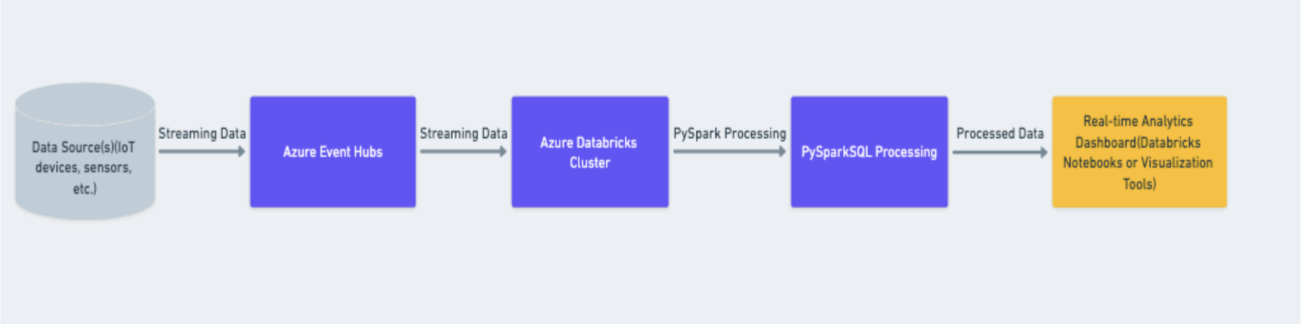
The aim of this project is to develop a real-time analytics dashboard leveraging Azure Databricks and PySpark. The objective is to ingest streaming data, process it using PySparkSQL, and visualize the results in real-time using Databricks notebooks or other visualization tools. This project aligns with the growing demand for real-time analytics solutions in various industries, providing insights into streaming data as it arrives.

Project Overview

The primary goal of this project is to develop a real-time analytics dashboard using Azure Databricks and PySpark. The project aims to ingest streaming data, process it with PySparkSQL, and visualize the results in real-time. By achieving this objective, the project enables organizations to gain immediate insights from streaming data sources, facilitating quick decision-making and proactive responses to evolving situations.


Real-time Data Processing Architecture in Azure
Data Ingestion
Devices/Sensors
IoT Devices
Azure Event Hubs
Real-time Data Processing
Azure Databricks
PySpark for data processing tasks
Stream Processing with PySparkSQL
Schema Definition
Data Transformation
Real-time Analysis
Data Visualization and Dashboarding
Databricks Notebooks
Benefits of Real-time Analytics Dashboard
Actionable Insights
Quick Decision-making
Proactive Responses

Architecture Diagram-



Execution overview-

1. **Project Setup:**
    - Set up Azure Databricks environment, including creating a Databricks workspace and configuring a cluster with appropriate specifications.
    - Provision necessary Azure resources such as Azure Event Hubs for streaming data ingestion.

- Install required libraries and dependencies within the Databricks environment for PySpark development.

2. **Data Ingestion:**

   - Configure Azure Event Hubs as the streaming data ingestion service.
   - Set up event listeners to continuously ingest streaming data from the data source(s) into Azure Databricks.

3. **Data Processing with PySpark:**

   - Define schemas for the streaming data to ensure proper parsing and transformation.
   - Implement PySparkSQL queries and transformations to process the streaming data in real-time.
   - Apply PySpark functions for tasks such as filtering, aggregation, and enrichment based on the analytical requirements.

4. **Real-time Analytics Dashboard Development:**

   - Create Databricks notebooks or leverage compatible visualization tools for building the real-time analytics dashboard.
   - Develop interactive charts, graphs, and widgets to visualize the insights derived from the streaming data.
   - Implement live updates and streaming capabilities within the dashboard to reflect changes in the underlying data in real-time.
   - Enhance user experience by providing interactive controls and filters for dynamic exploration of the data.

5. **Performance Optimization:**

   - Optimize PySpark code and queries to improve processing performance and reduce latency in real-time analytics.
   - Utilize caching, partitioning, and other optimization techniques to enhance the efficiency of data processing operations.
   - Monitor cluster performance and resource utilization to ensure scalability and reliability under varying workload conditions.

6. **Testing and Validation:**

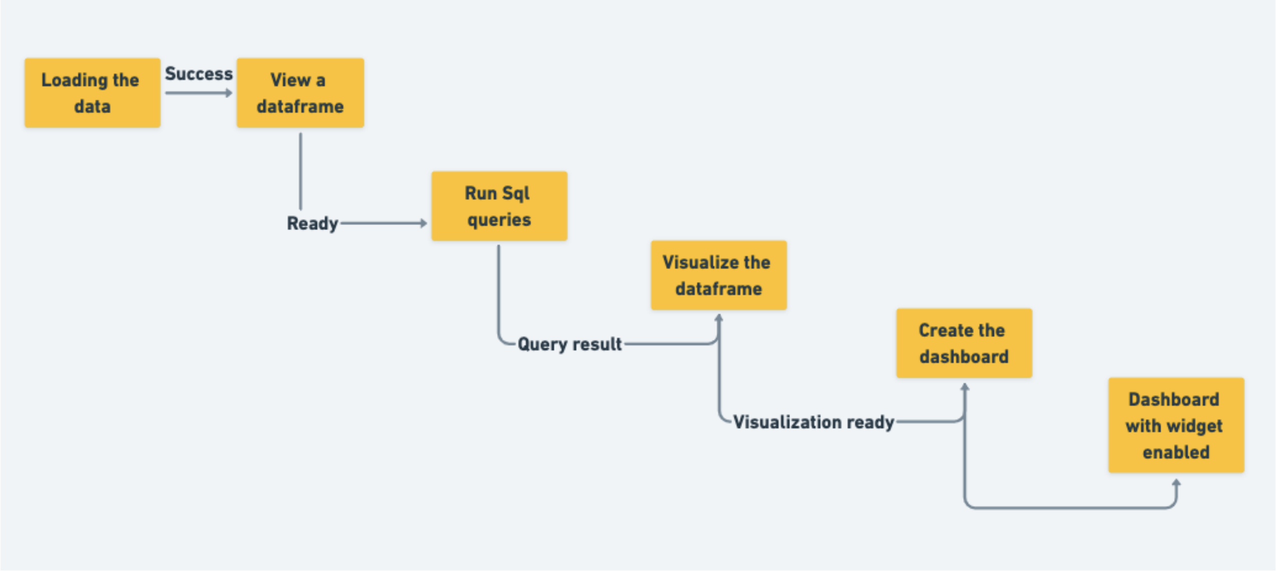
- Conduct thorough testing of the real-time analytics dashboard to ensure its functionality, performance, and reliability.
- Validate the accuracy of the insights generated by comparing them with expected results and business requirements.
- Address any issues or bugs identified during testing and make necessary refinements to the implementation.

7. Deployment and Integration:

- Deploy the real-time analytics dashboard to the production environment, making it accessible to stakeholders and end-users.
- Integrate the dashboard with other systems or data sources as needed to enrich the analysis and provide comprehensive insights.
- Ensure seamless integration with existing workflows and processes within the organization.

8. Documentation and Knowledge Sharing:

- Document the project architecture, implementation details, and deployment instructions for future reference.
- Conduct knowledge sharing sessions to disseminate insights gained from the project and promote best practices in real-time analytics and data visualization.



Flow diagram of the project

Key components/Technologies Used

1. **Azure Databricks:**
    - Provides a cloud-based platform for big data processing and analytics.
    - Offers scalable clusters with Apache Spark capabilities for distributed data processing.
    - Enables seamless integration with other Azure services for data ingestion, storage, and visualization.
2. **PySpark:**
    - Python API for Apache Spark, facilitating data processing and analytics tasks.
    - Allows for the development of distributed data processing applications using the Spark framework.
    - Provides rich libraries and functions for data manipulation, querying, and analysis.
3. **Azure Event Hubs:**
    - Scalable event ingestion service for streaming data.
    - Facilitates the ingestion of high-volume, real-time data streams from various sources.
    - Integrates with Azure Databricks for seamless data ingestion into the analytics pipeline.
4. **Data Sources:**
    - IoT devices, sensors, application logs, or other streaming data sources.
    - Provide real-time data streams containing valuable insights for analysis.
5. **Databricks Notebooks:**
    - Interactive development environment for data exploration, analysis, and visualization.
    - Supports various programming languages including Python, SQL, and Scala.
    - Enables the creation of interactive dashboards and visualizations for real-time analytics.
6. **Visualization Tools:**

- Matplotlib, Plotly, Bokeh, or other Python libraries for creating interactive visualizations.
- Used to build real-time dashboards with charts, graphs, and widgets for data visualization.

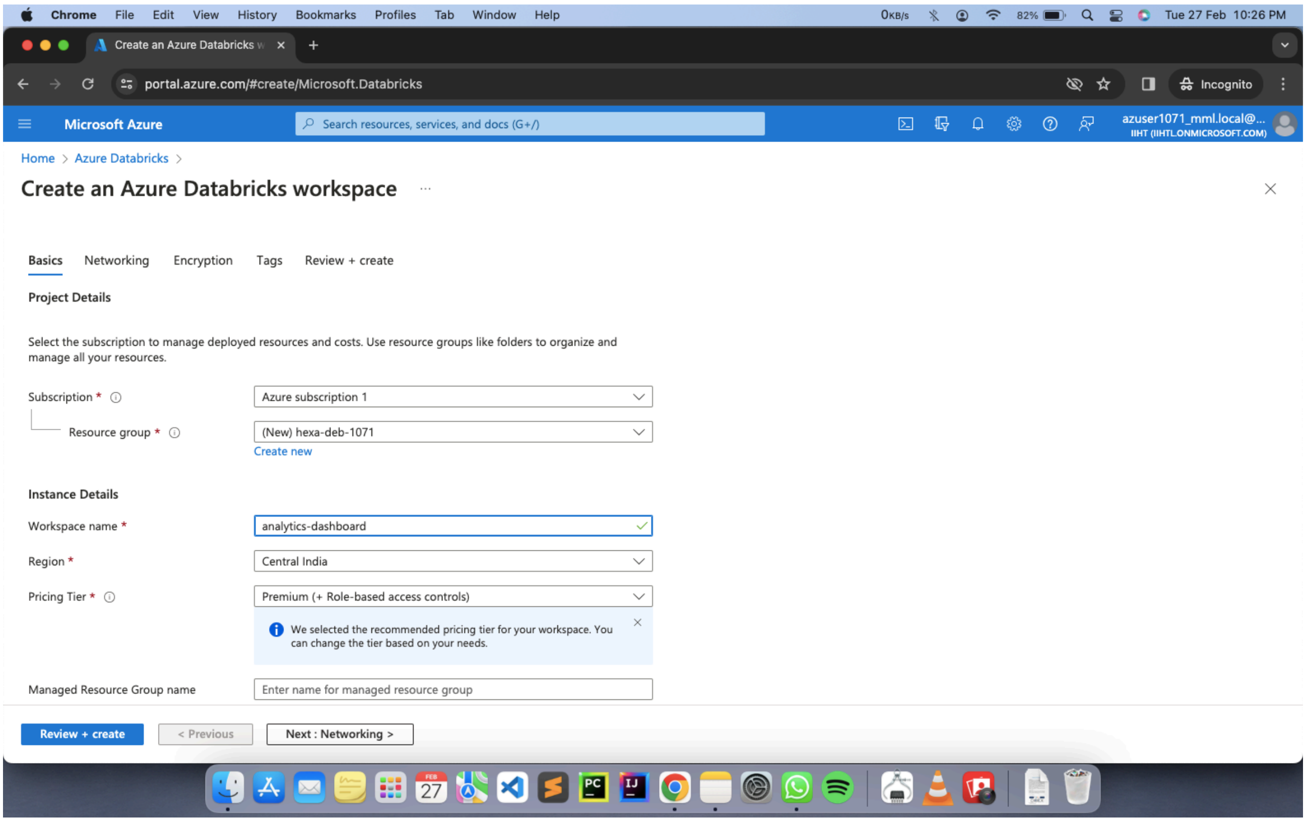
7. **PySparkSQL:**

- Component of PySpark for executing SQL queries against Spark dataframes.
- Allows for querying and processing structured data in real-time.
- Enables the implementation of complex data transformations and aggregations for analytics.

How it works-

1. <u>Setting Up Azure Databricks Environment:</u>

   - Sign in to the Azure portal and create an Azure Databricks workspace. Configure workspace settings, including pricing tier, region, and workspace name.



Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tue 27 Feb 10:26 PM
Create an Azure Databricks w
portal.azure.com/#create/Microsoft.Databricks
Incognito
Microsoft Azure
Search resources, services, and docs (G+/)
azuser1071_mml.local@...
IIHT (IIHTL.ONMICROSOFT.COM)
Home > Azure Databricks >

Create an Azure Databricks workspace

Basics Networking Encryption Tags Review + create

Project Details

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

Subscription * : Azure subscription 1

Resource group * : (New) hexa-deb-1071

Create new

Instance Details

Workspace name * : analytics-dashboard

Region * : Central India

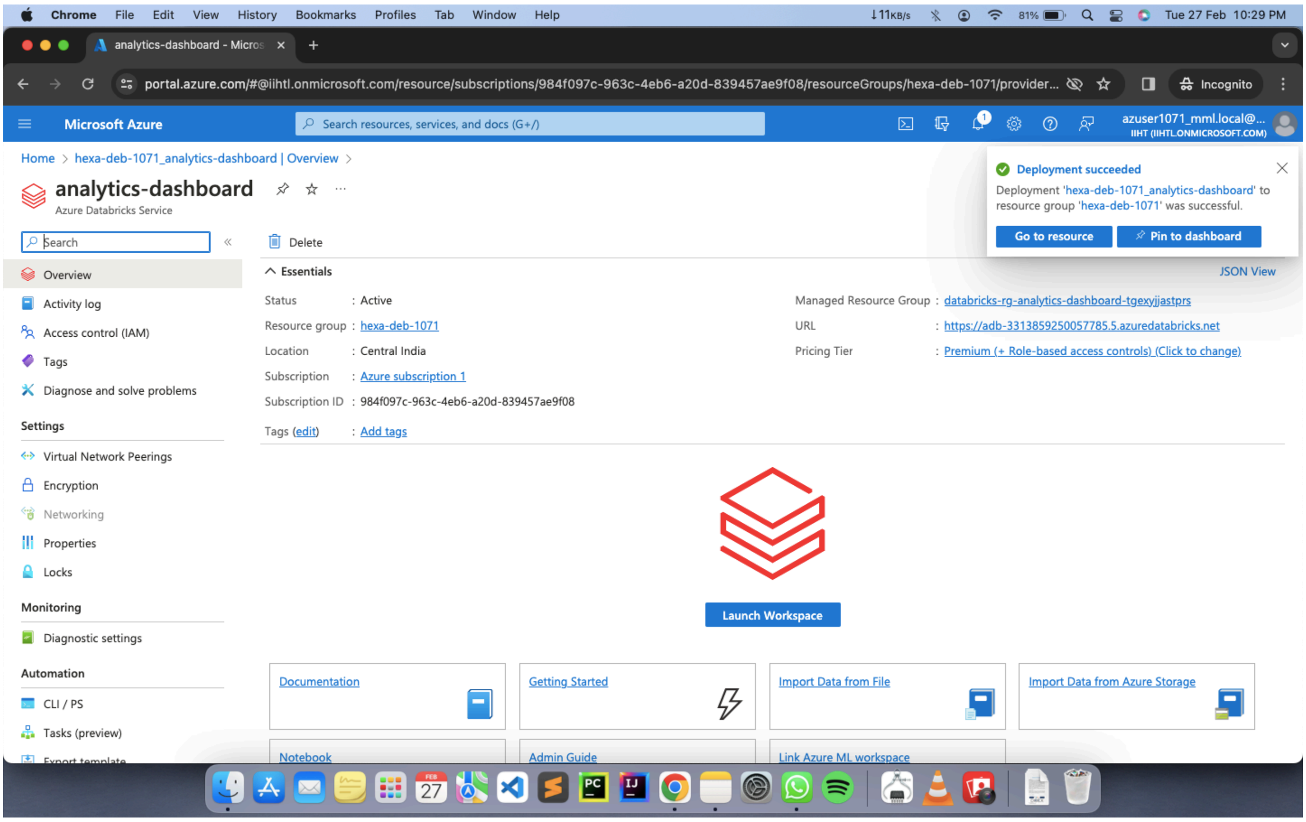
Pricing Tier * : Premium (+ Role-based access controls)

We selected the recommended pricing tier for your workspace. You can change the tier based on your needs.

Managed Resource Group name : Enter name for managed resource group

2. Developing PySpark Notebooks:

- Create a new PySpark notebook within the Databricks workspace. Begin writing PySpark code to perform ETL operations, data transformations, and other data processing tasks.



portal.azure.com/#@iihtl.onmicrosoft.com/resource/subscriptions/984f097c-963c-4eb6-a20d-839457ae9f08/resourceGroups/hexa-deb-1071/provider...

Home > hexa-deb-1071_analytics-dashboard | Overview >

analytics-dashboard

Azure Databricks Service

Deployment succeeded

Deployment 'hexa-deb-1071_analytics-dashboard' to resource group 'hexa-deb-1071' was successful.

Essentials

Status : Active

Resource group : hexa-deb-1071

Location : Central India

Subscription : Azure subscription 1

Subscription ID : 984f097c-963c-4eb6-a20d-839457ae9f08

Tags (edit) : Add tags

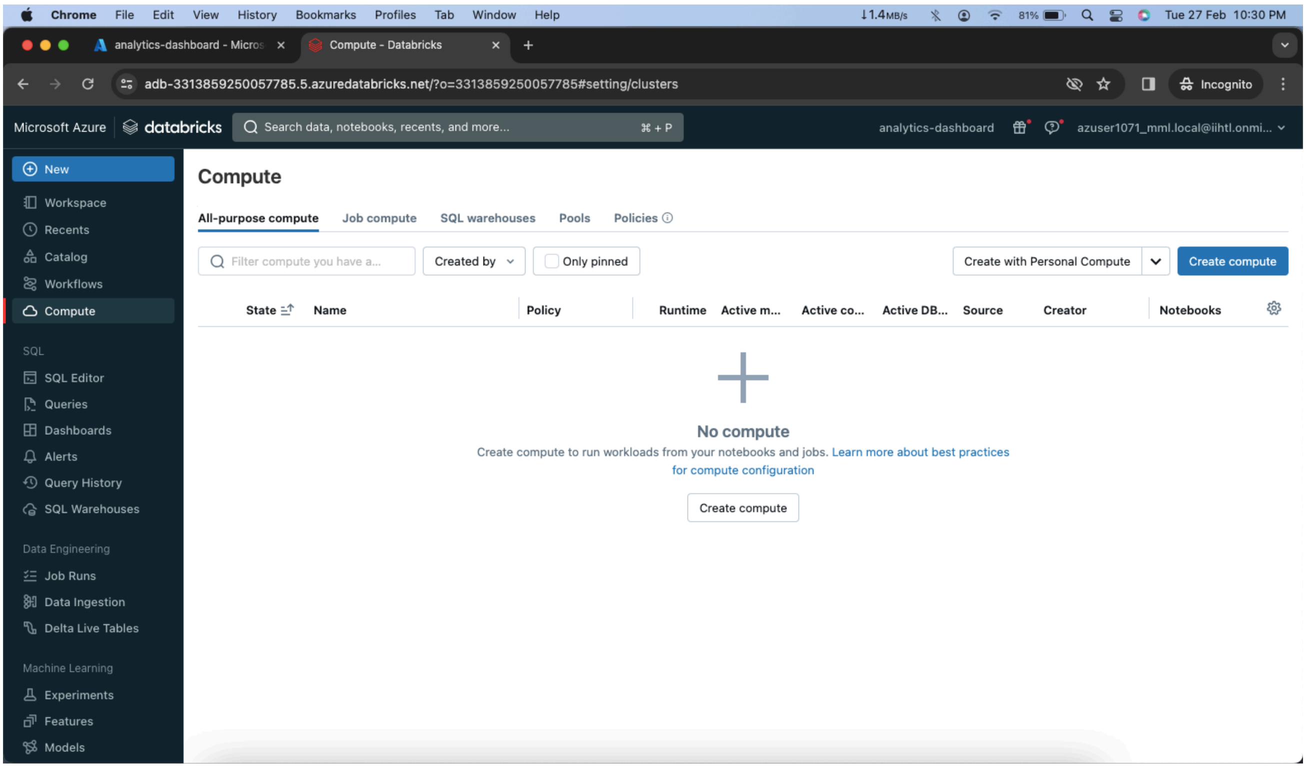
Managed Resource Group : databricks-rg-analytics-dashboard-tgexyjjastprs

URL : https://adb-3313859250057785.5.azuredatabricks.net

Pricing Tier : Premium (+ Role-based access controls) (Click to change)

3. Create Cluster and Connecting to notebook

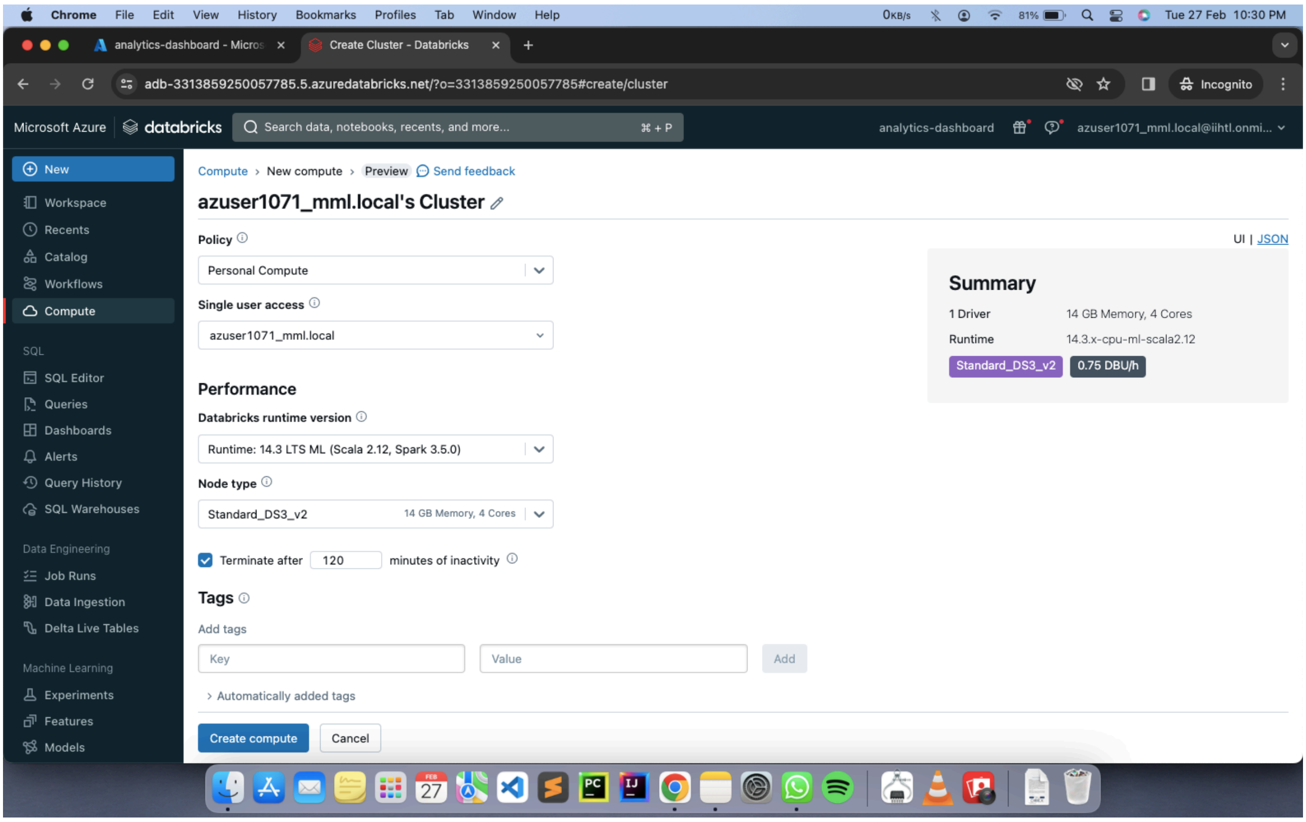
- The cluster is created with 4 working nodes and autoscaling is enabled which automatically adjust cluster size to accommodate changes in workload demand, allowing for seamless scalability without manual intervention.



Compute

No compute

Create compute to run workloads from your notebooks and jobs. Learn more about best practices for compute configuration



azuser1071_mml.local's Cluster

Policy: Personal Compute

Single user access: azuser1071_mml.local

Performance

Databricks runtime version: Runtime: 14.3 LTS ML (Scala 2.12, Spark 3.5.0)

Node type: Standard_DS3_v2 14 GB Memory, 4 Cores

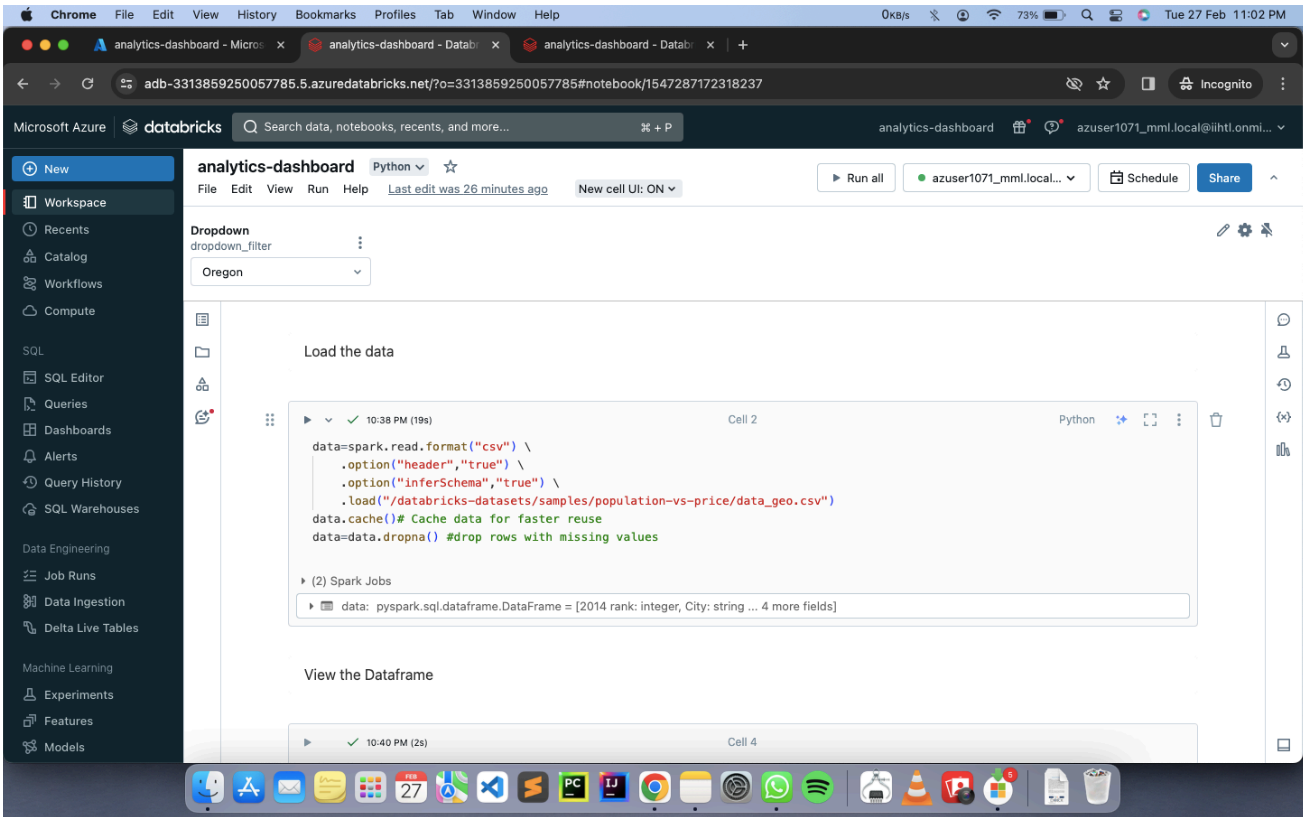
☑ Terminate after 120 minutes of inactivity

Tags

Summary

1 Driver	14 GB Memory, 4 Cores
Runtime	14.3.x-cpu-ml-scala2.12
Standard_DS3_v2	0.75 DBU/h

4. Importing Necessary libraries and Loading the data



Load the data

10:38 PM (19s) Cell 2 Python

```
data=spark.read.format("csv") \
    .option("header","true") \
    .option("inferSchema","true") \
    .load("/databricks-datasets/samples/population-vs-price/data_geo.csv")
data.cache()# Cache data for faster reuse
data=data.dropna() #drop rows with missing values
```

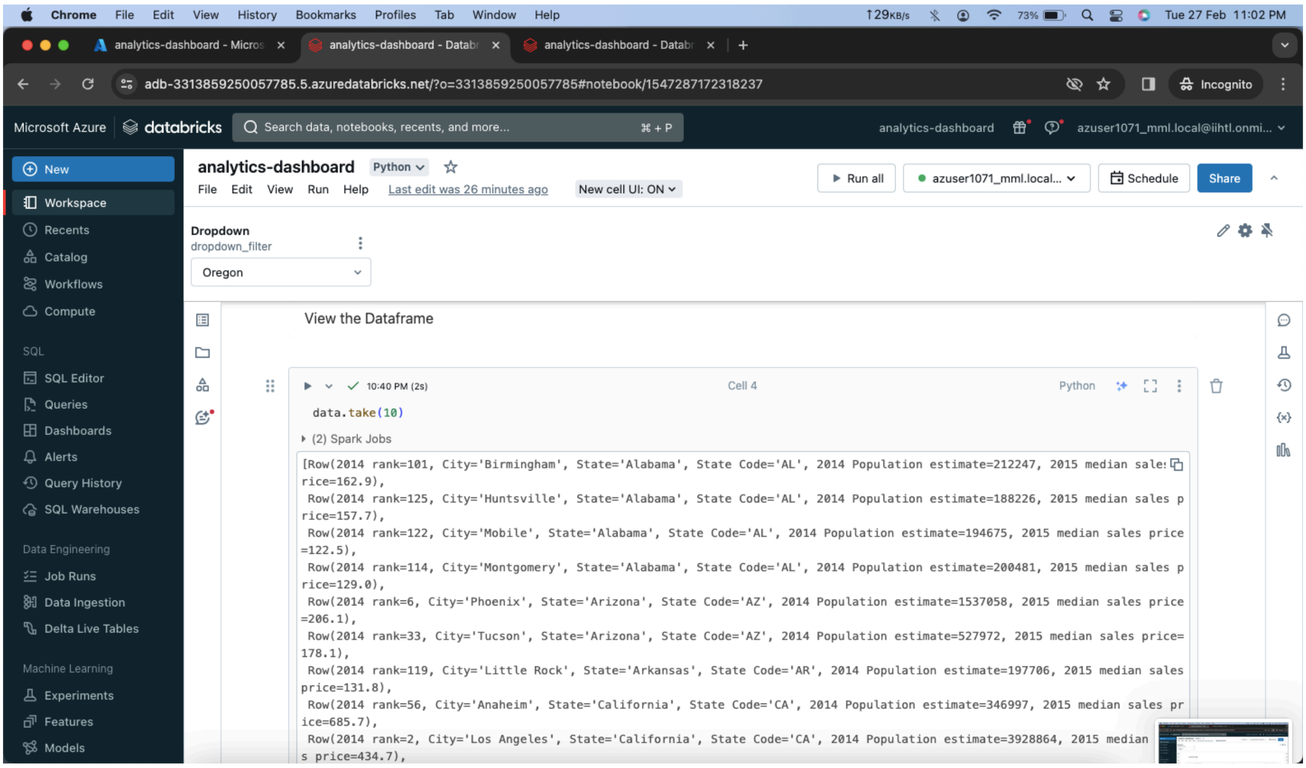
(2) Spark Jobs

data: pyspark.sql.dataframe.DataFrame = [2014 rank: integer, City: string ... 4 more fields]

View the Dataframe

10:40 PM (2s) Cell 4

5. View a dataframe-



View the Dataframe

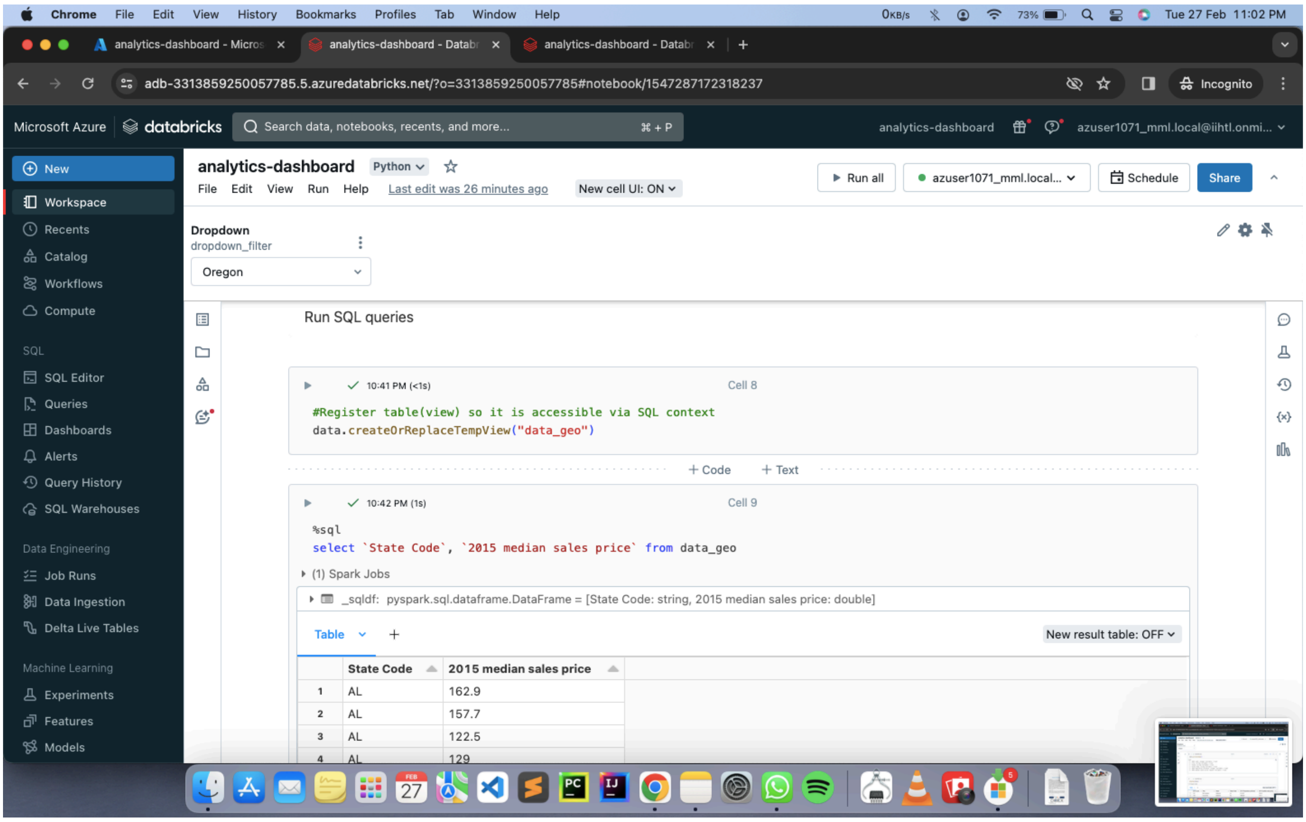
10:40 PM (2s) Cell 4 Python

```
data.take(10)
```

(2) Spark Jobs

```
[Row(2014 rank=101, City='Birmingham', State='Alabama', State Code='AL', 2014 Population estimate=212247, 2015 median sales p
rice=162.9),
 Row(2014 rank=125, City='Huntsville', State='Alabama', State Code='AL', 2014 Population estimate=188226, 2015 median sales p
rice=157.7),
 Row(2014 rank=122, City='Mobile', State='Alabama', State Code='AL', 2014 Population estimate=194675, 2015 median sales price
=122.5),
 Row(2014 rank=114, City='Montgomery', State='Alabama', State Code='AL', 2014 Population estimate=200481, 2015 median sales p
rice=129.0),
 Row(2014 rank=6, City='Phoenix', State='Arizona', State Code='AZ', 2014 Population estimate=1537058, 2015 median sales price
=206.1),
 Row(2014 rank=33, City='Tucson', State='Arizona', State Code='AZ', 2014 Population estimate=527972, 2015 median sales price=
178.1),
 Row(2014 rank=119, City='Little Rock', State='Arkansas', State Code='AR', 2014 Population estimate=197706, 2015 median sales
price=131.8),
 Row(2014 rank=56, City='Anaheim', State='California', State Code='CA', 2014 Population estimate=346997, 2015 median sales pr
ice=685.7),
 Row(2014 rank=2, City='Los Angeles', State='California', State Code='CA', 2014 Population estimate=3928864, 2015 median sale
s price=434.7),
```

6. Run SQL queries



Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tue 27 Feb 11:02 PM
analytics-dashboard - Micros
analytics-dashboard - Databr
adb-3313859250057785.5.azuredatabricks.net/?o=3313859250057785#notebook/1547287172318237
Incognito
Microsoft Azure databricks
Search data, notebooks, recents, and more...
azuser1071_mml.local@iihtl.onmi...
New
Workspace
Recents
Catalog
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Experiments
Features
Models
analytics-dashboard Python
File Edit View Run Help Last edit was 26 minutes ago
New cell UI: ON
Run all
azuser1071_mml.local...
Schedule
Share

Dropdown
dropdown_filter
Oregon

Run SQL queries

10:41 PM (<1s) Cell 8

```
#Register table(view) so it is accessible via SQL context
data.createOrReplaceTempView("data_geo")
```

10:42 PM (1s) Cell 9

```
%sql
select `State Code`, `2015 median sales price` from data_geo
```

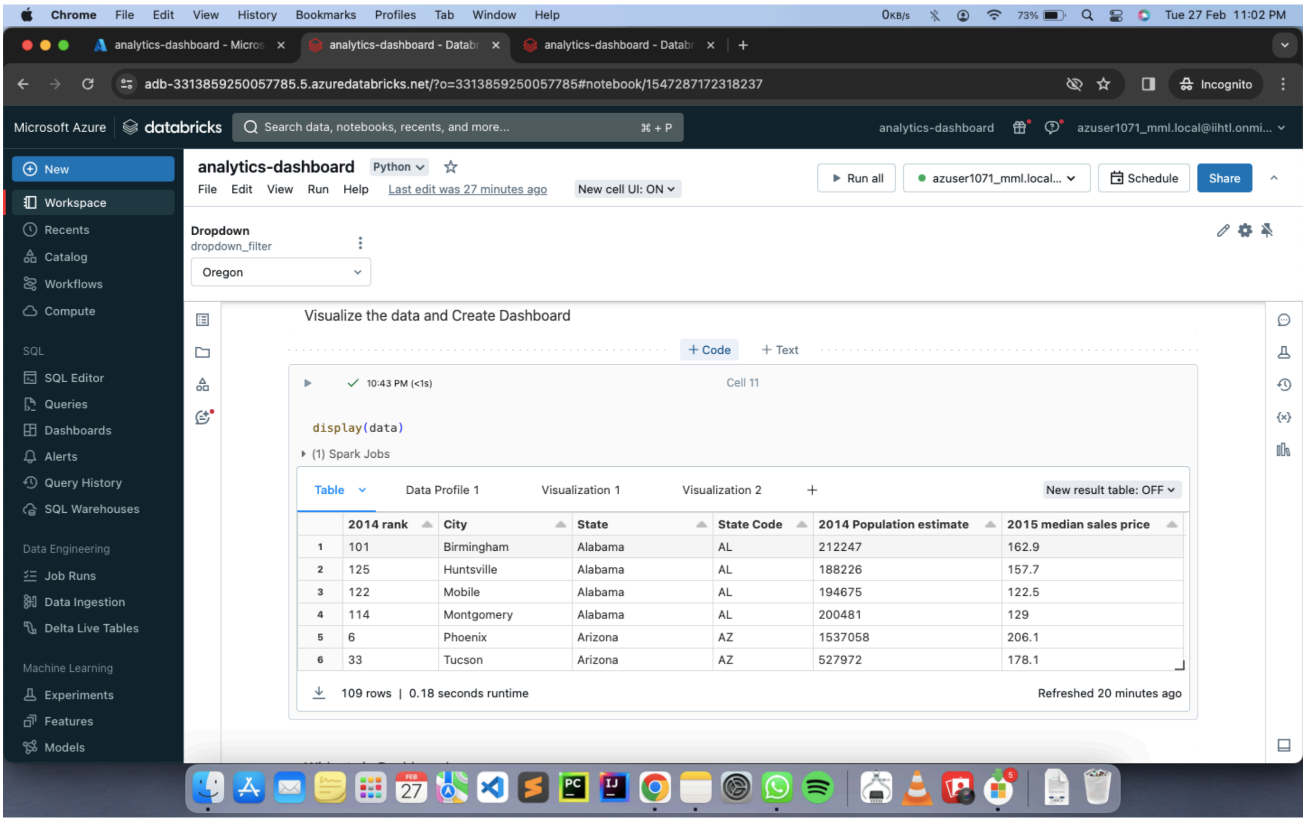
(1) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [State Code: string, 2015 median sales price: double]

Table New result table: OFF

	State Code	2015 median sales price
1	AL	162.9
2	AL	157.7
3	AL	122.5
4	AL	129

7. Visualize the dataframe



Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tue 27 Feb 11:02 PM
adb-3313859250057785.5.azuredatabricks.net/?o=3313859250057785#notebook/1547287172318237
Microsoft Azure databricks
analytics-dashboard Python
File Edit View Run Help Last edit was 27 minutes ago
New cell UI: ON
Run all
Schedule
Share

Dropdown
dropdown_filter
Oregon

Visualize the data and Create Dashboard

10:43 PM (<1s) Cell 11

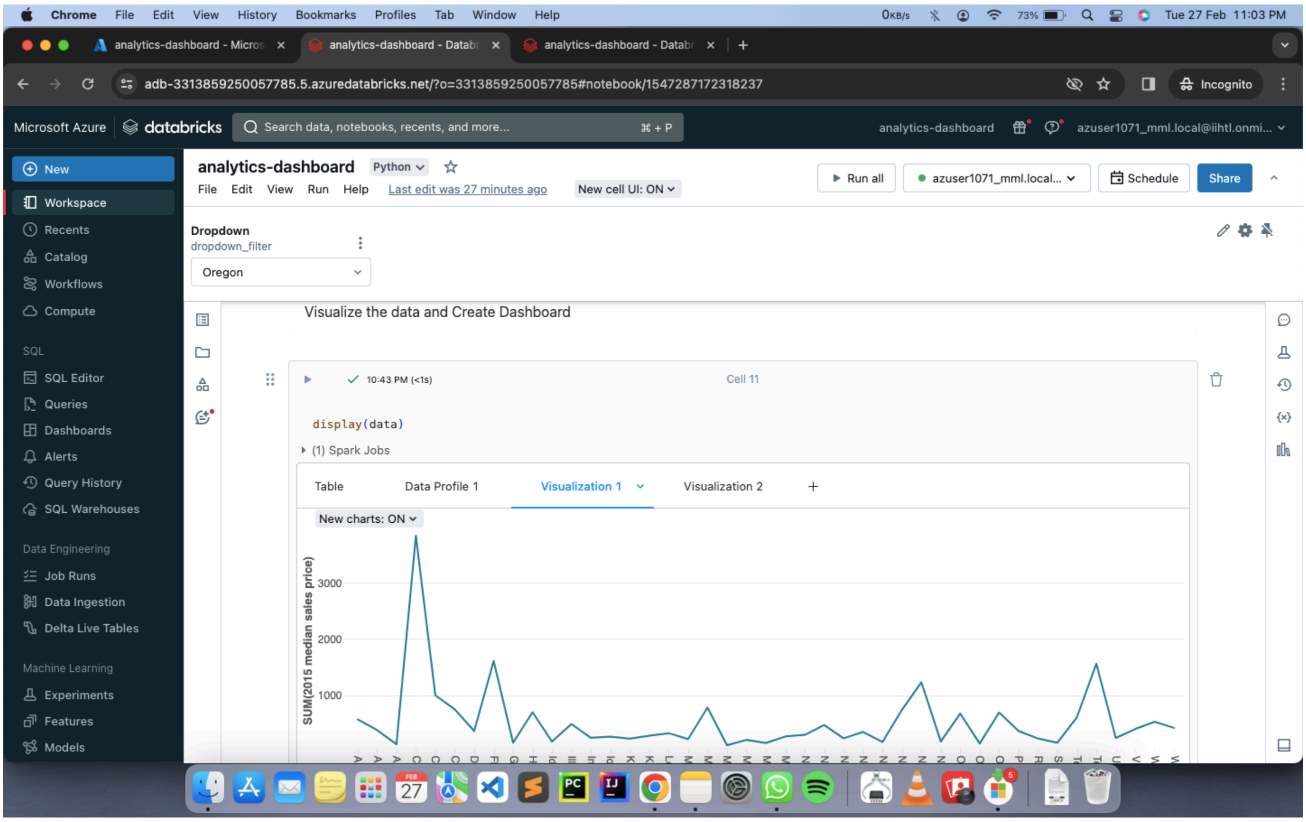
```
display(data)
```

(1) Spark Jobs

Table Data Profile 1 Visualization 1 Visualization 2 New result table: OFF

	2014 rank	City	State	State Code	2014 Population estimate	2015 median sales price
1	101	Birmingham	Alabama	AL	212247	162.9
2	125	Huntsville	Alabama	AL	188226	157.7
3	122	Mobile	Alabama	AL	194675	122.5
4	114	Montgomery	Alabama	AL	200481	129
5	6	Phoenix	Arizona	AZ	1537058	206.1
6	33	Tucson	Arizona	AZ	527972	178.1

109 rows | 0.18 seconds runtime Refreshed 20 minutes ago



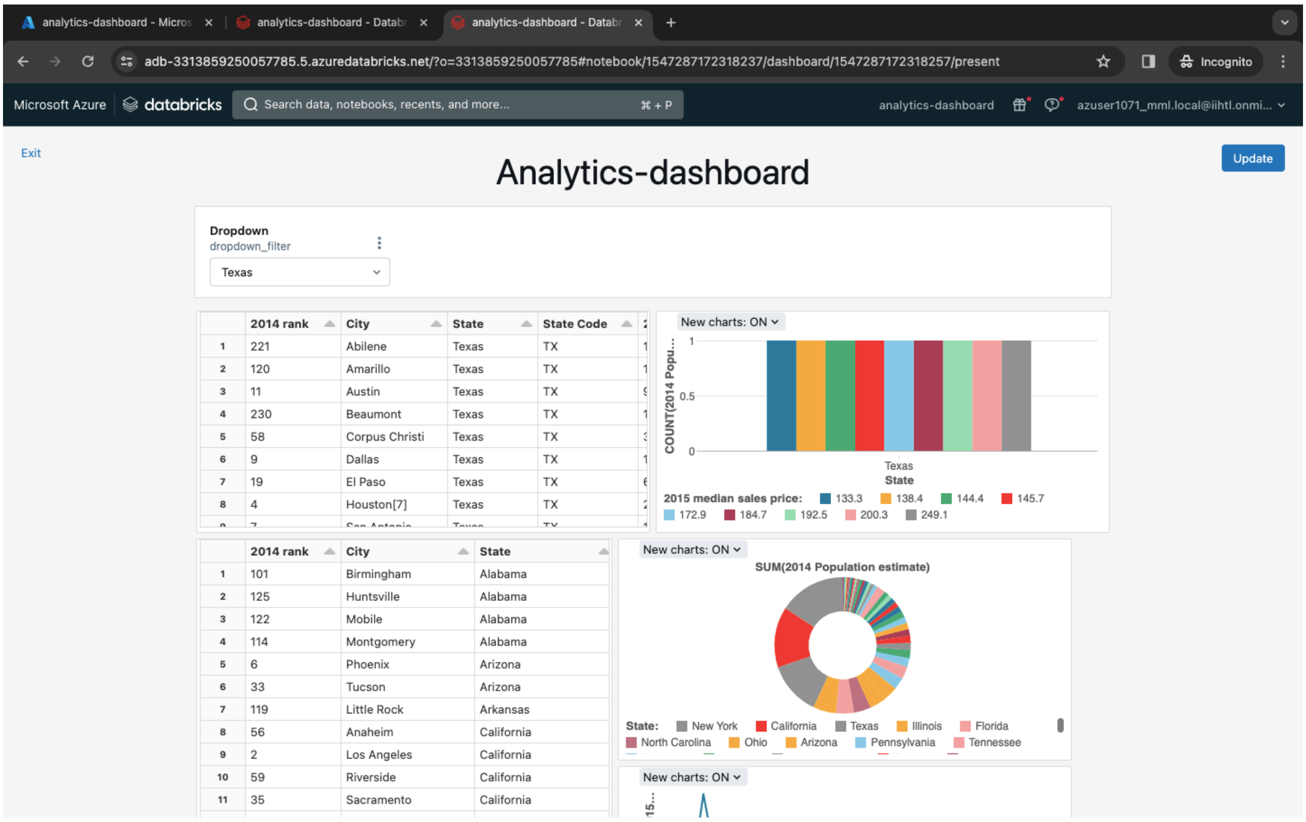
Dropdown
dropdown_filter
Oregon

Visualize the data and Create Dashboard

```
display(data)
```

(1) Spark Jobs

8.Create the dashboard—


analytics-dashboard - Micros
analytics-dashboard - Datab
analytics-dashboard - Datab
adb-3313859250057785.5.azuredatabricks.net/?o=3313859250057785#notebook/1547287172318237/dashboard/1547287172318257/present
Incognito
Microsoft Azure
databricks
Search data, notebooks, recents, and more...
analytics-dashboard
azuser1071_mml.local@iihtl.onmi...
Exit
Update
New charts: ON
COUNT(2014 Popu...
1
0.5
0
Texas
State
2015 median sales price: 133.3 138.4 144.4 145.7 172.9 184.7 192.5 200.3 249.1
New charts: ON
SUM(2014 Population estimate)
State: New York California Texas Illinois Florida North Carolina Ohio Arizona Pennsylvania Tennessee
New charts: ON
15...

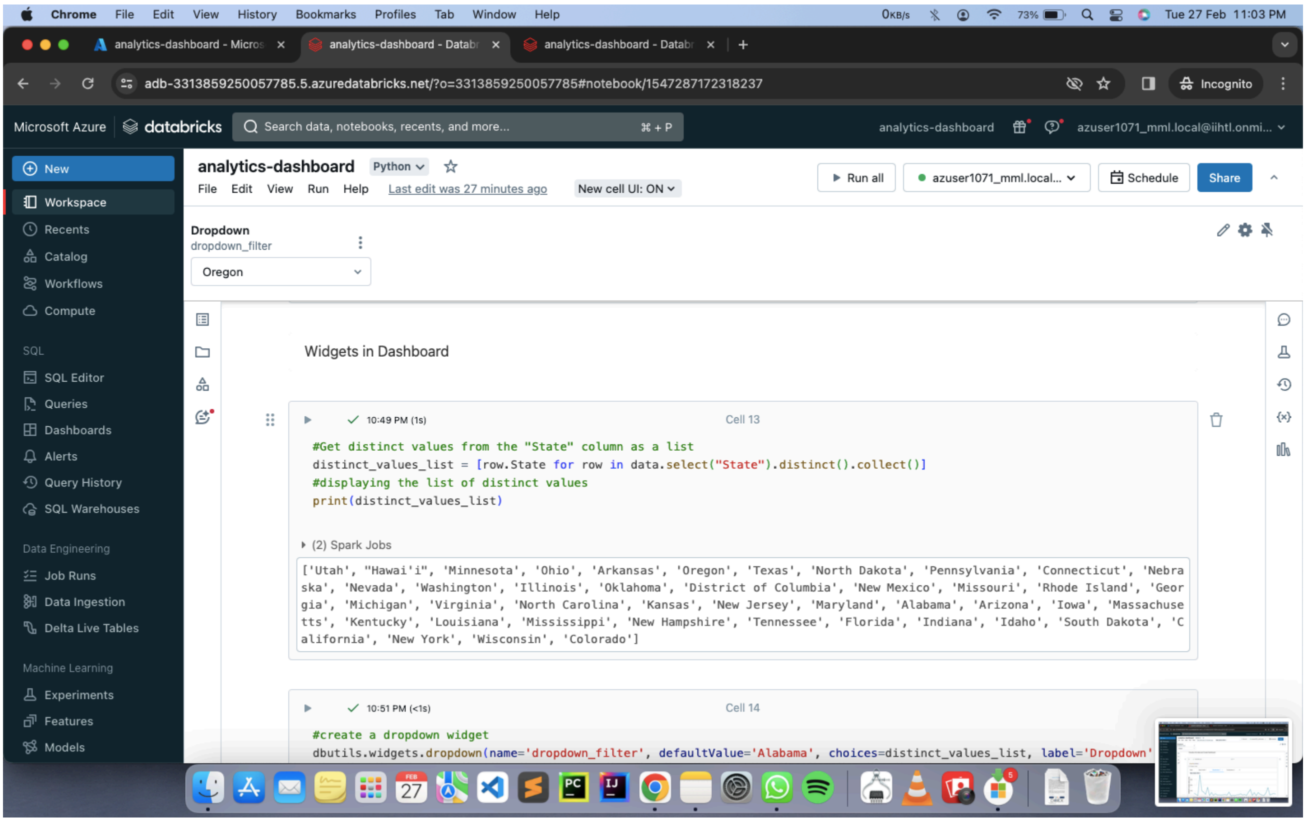
Analytics-dashboard

Dropdown
dropdown_filter
Texas

	2014 rank	City	State	State Code
1	221	Abilene	Texas	TX
2	120	Amarillo	Texas	TX
3	11	Austin	Texas	TX
4	230	Beaumont	Texas	TX
5	58	Corpus Christi	Texas	TX
6	9	Dallas	Texas	TX
7	19	El Paso	Texas	TX
8	4	Houston[7]	Texas	TX
9	7	San Antonio	Texas	TX

	2014 rank	City	State
1	101	Birmingham	Alabama
2	125	Huntsville	Alabama
3	122	Mobile	Alabama
4	114	Montgomery	Alabama
5	6	Phoenix	Arizona
6	33	Tucson	Arizona
7	119	Little Rock	Arkansas
8	56	Anaheim	California
9	2	Los Angeles	California
10	59	Riverside	California
11	35	Sacramento	California

9.Dashboard with widget enabled

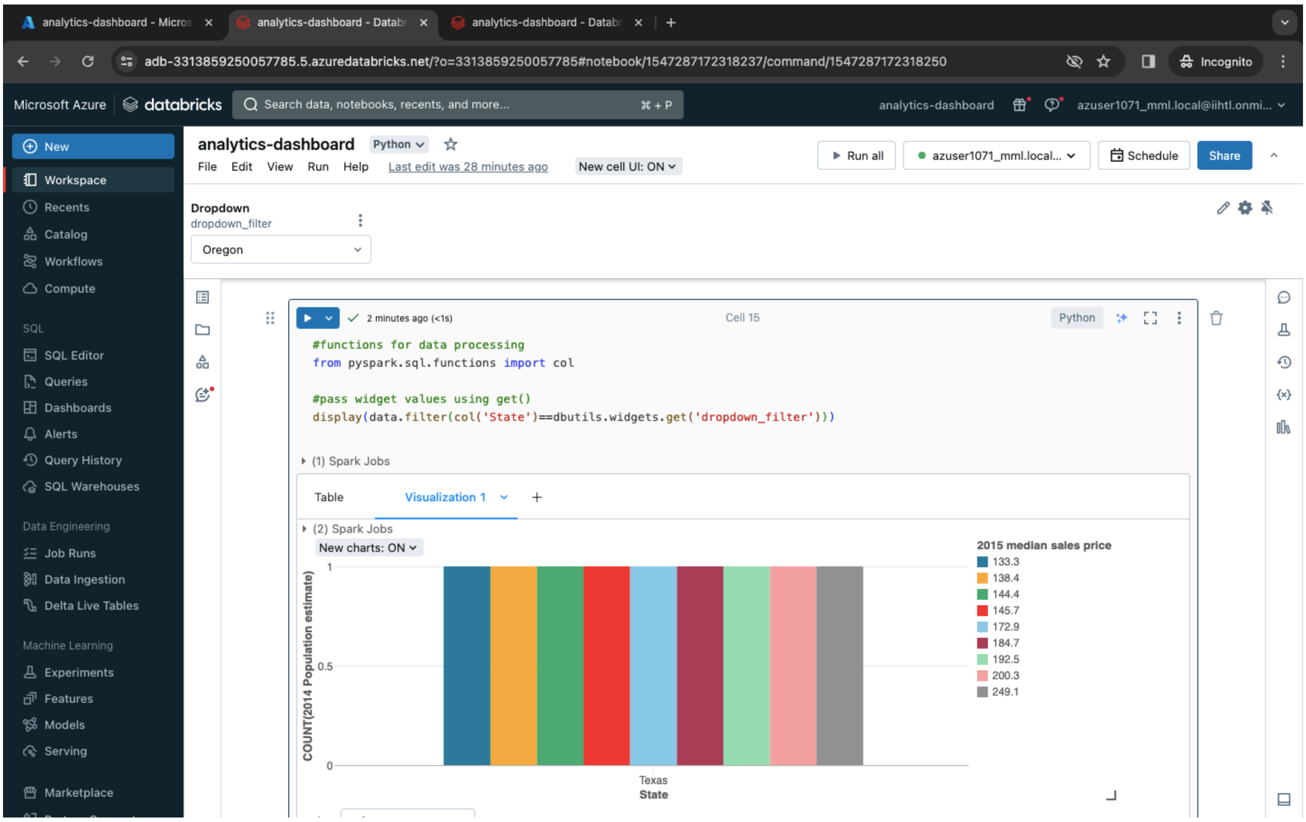


```
#Get distinct values from the "State" column as a list
distinct_values_list = [row.State for row in data.select("State").distinct().collect()]
#displaying the list of distinct values
print(distinct_values_list)
```

(2) Spark Jobs

```
['Utah', "Hawai'i", 'Minnesota', 'Ohio', 'Arkansas', 'Oregon', 'Texas', 'North Dakota', 'Pennsylvania', 'Connecticut', 'Nebra
ska', 'Nevada', 'Washington', 'Illinois', 'Oklahoma', 'District of Columbia', 'New Mexico', 'Missouri', 'Rhode Island', 'Geor
gia', 'Michigan', 'Virginia', 'North Carolina', 'Kansas', 'New Jersey', 'Maryland', 'Alabama', 'Arizona', 'Iowa', 'Massachuse
tts', 'Kentucky', 'Louisiana', 'Mississippi', 'New Hampshire', 'Tennessee', 'Florida', 'Indiana', 'Idaho', 'South Dakota', 'C
alifornia', 'New York', 'Wisconsin', 'Colorado']
```

```
#create a dropdown widget
dbutils.widgets.dropdown(name='dropdown_filter', defaultValue='Alabama', choices=distinct_values_list, label='Dropdown'
```



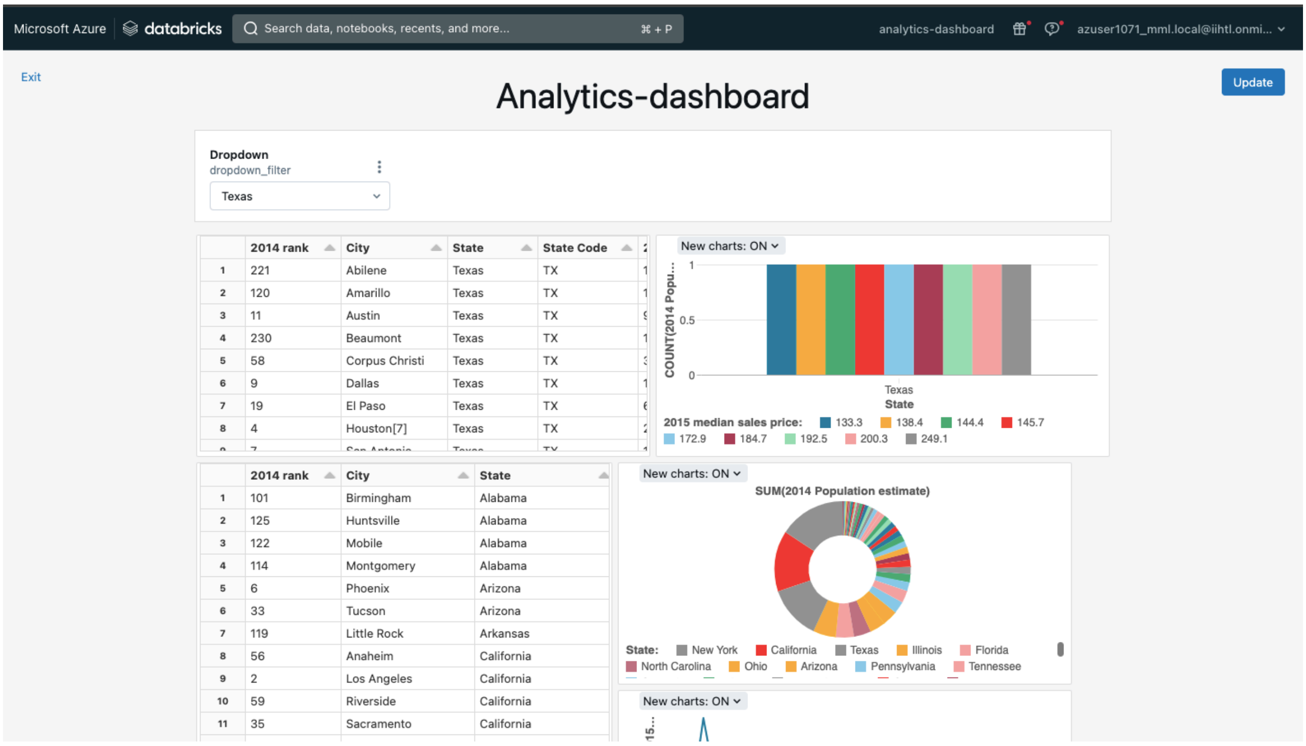
```
#functions for data processing
from pyspark.sql.functions import col

#pass widget values using get()
display(data.filter(col('State')==dbutils.widgets.get('dropdown_filter')))
```

(1) Spark Jobs

(2) Spark Jobs

Final Output-



Microsoft Azure
databricks
Search data, notebooks, recents, and more...
⌘ + P
analytics-dashboard
azuser1071_mml.local@iihtl.onmi...
Exit
Update

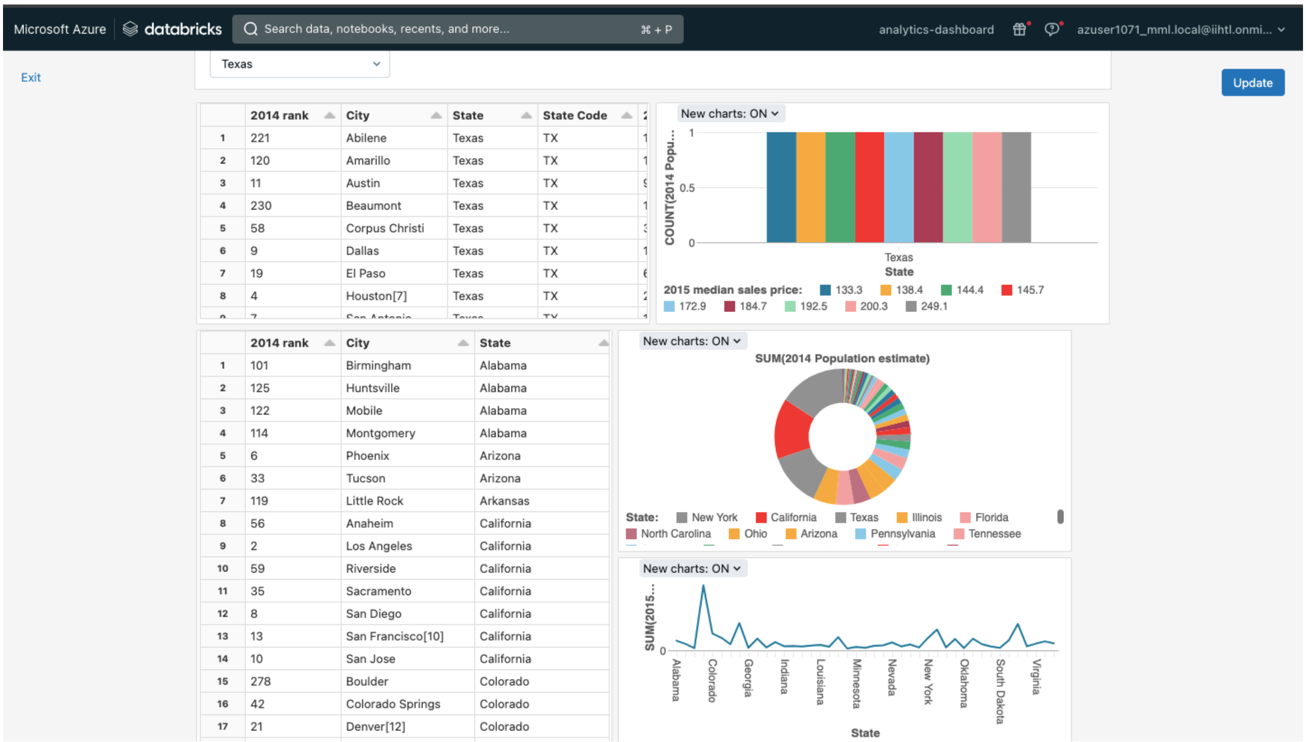
Analytics-dashboard

Dropdown
dropdown_filter

Texas

	2014 rank	City	State	State Code	2
1	221	Abilene	Texas	TX	1
2	120	Amarillo	Texas	TX	1
3	11	Austin	Texas	TX	9
4	230	Beaumont	Texas	TX	1
5	58	Corpus Christi	Texas	TX	3
6	9	Dallas	Texas	TX	1
7	19	El Paso	Texas	TX	6
8	4	Houston[7]	Texas	TX	2
9	7	San Antonio	Texas	TX	1

	2014 rank	City	State
1	101	Birmingham	Alabama
2	125	Huntsville	Alabama
3	122	Mobile	Alabama
4	114	Montgomery	Alabama
5	6	Phoenix	Arizona
6	33	Tucson	Arizona
7	119	Little Rock	Arkansas
8	56	Anaheim	California
9	2	Los Angeles	California
10	59	Riverside	California
11	35	Sacramento	California



Microsoft Azure
databricks
Search data, notebooks, recents, and more...
⌘ + P
analytics-dashboard
azuser1071_mml.local@iihtl.onmi...
Exit
Update

Texas

	2014 rank	City	State	State Code	2
1	221	Abilene	Texas	TX	1
2	120	Amarillo	Texas	TX	1
3	11	Austin	Texas	TX	9
4	230	Beaumont	Texas	TX	1
5	58	Corpus Christi	Texas	TX	3
6	9	Dallas	Texas	TX	1
7	19	El Paso	Texas	TX	6
8	4	Houston[7]	Texas	TX	2
9	7	San Antonio	Texas	TX	1

	2014 rank	City	State
1	101	Birmingham	Alabama
2	125	Huntsville	Alabama
3	122	Mobile	Alabama
4	114	Montgomery	Alabama
5	6	Phoenix	Arizona
6	33	Tucson	Arizona
7	119	Little Rock	Arkansas
8	56	Anaheim	California
9	2	Los Angeles	California
10	59	Riverside	California
11	35	Sacramento	California
12	8	San Diego	California
13	13	San Francisco[10]	California
14	10	San Jose	California
15	278	Boulder	Colorado
16	42	Colorado Springs	Colorado
17	21	Denver[12]	Colorado

Conclusion

In conclusion, the real-time analytics dashboard project utilizing Azure Databricks and Py-Spark offers a transformative solution for organizations seeking to harness the power of streaming data for actionable insights and informed decision-making. By ingesting, processing, and visualizing streaming data in real-time, the project enables stakeholders to monitor key metrics, detect trends, and respond proactively to changing conditions.

The project demonstrated the following key outcomes and benefits:

1. **Real-time Insights:** By processing streaming data in real-time, the dashboard provides immediate insights into operational performance, customer behaviour, and emerging trends.

2. **Agility and Scalability:** Leveraging Azure Databricks' cloud-based infrastructure, the solution offers agility and scalability to adapt to evolving data requirements and handle growing data volumes.

3**. Decision Support:** The interactive dashboards and visualizations empower stakeholders with the information needed to make data-driven decisions quickly and confidently.

4**. Efficiency and Performance:** Through optimization techniques and distributed computing capabilities of PySpark, the project ensures efficient data processing and low-latency analytics.

6. **Collaboration and Knowledge Sharing:** The project fosters collaboration among data engineers, data scientists, and analysts through shared notebooks, documentation, and knowledge-sharing sessions.