

Multimodal Reasoning AI Agent

Upload an image and provide a reasoning-based task for the AI Agent. The AI Agent will analyze the image and respond based on your input.

Upload Image



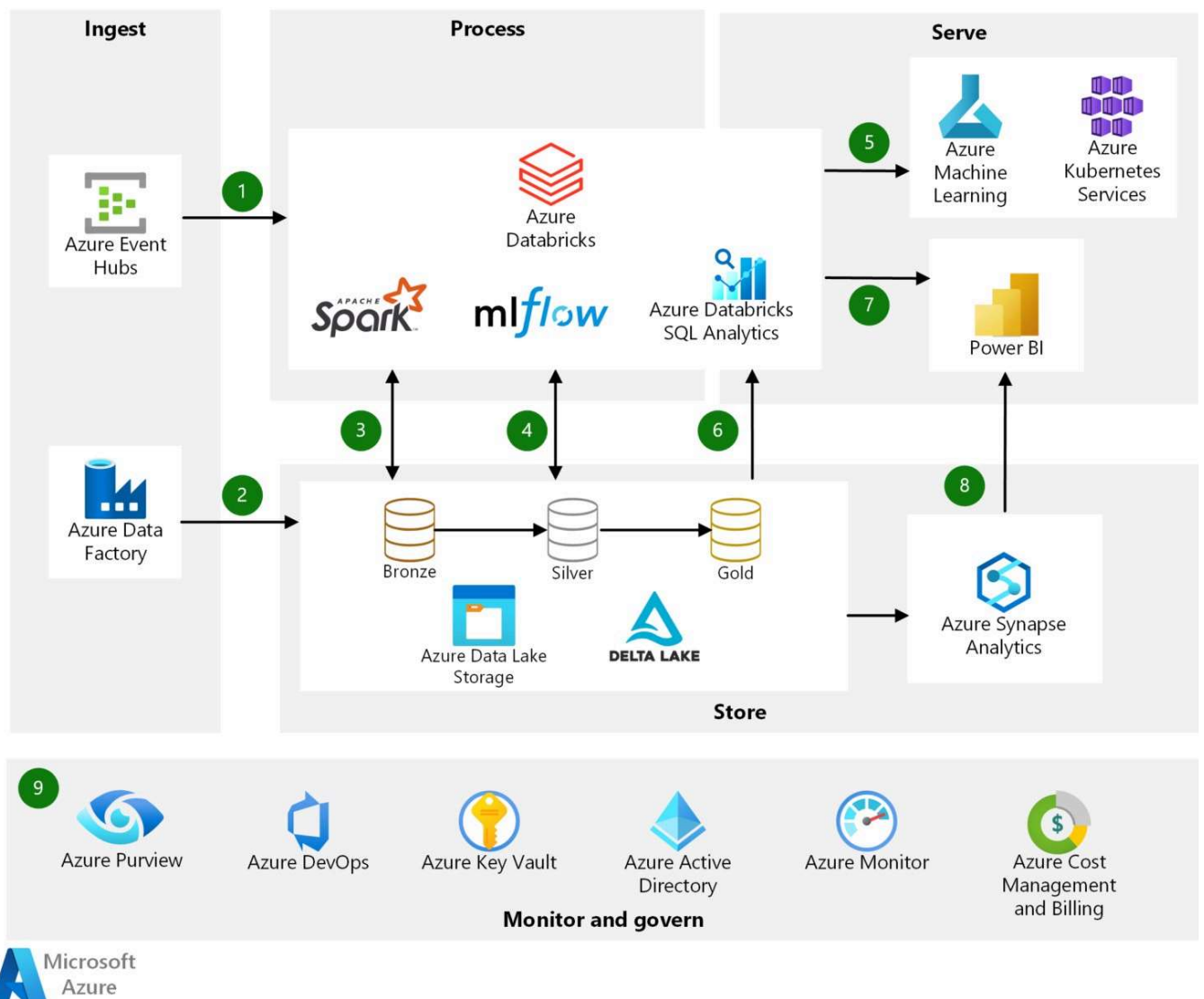
Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files



azure-databricks-modern-analytics-architecture-diagram.png 215.1KB



Uploaded Image

Enter your task/question for the AI Agent:

write code for this data flow

Analyze Image

AI Response:

```
# This code illustrates the data flow depicted in the image.
# It provides conceptual code snippets for each step, as actual
# implementation would require specific configurations and data.

# 1. Ingest: Azure Event Hubs to Azure Databricks (using Spark Streaming)
from pyspark.sql.functions import *
from pyspark.sql.types import *

# Configure Event Hubs connection details
ehConf = {
    "eventhubs.connectionString" : "Endpoint=sb://<your_namespace>.servicebus.window
    "eventhubs.consumerGroup" : "your_consumer_group"
}

# Read data from Event Hubs as a streaming DataFrame
df_eh = spark.readStream.format("eventhubs").options(**ehConf).load()

# Process the incoming events (example: print the body)
df_eh.select(col("body").cast("string")).writeStream.outputMode("append").format("

# 2. Ingest: Azure Data Factory to Azure Data Lake Storage (Bronze)
# This is typically configured within the Azure Data Factory UI.
# Example ADF pipeline activity (Conceptual):
# {
#   "name": "CopyDataToBronze",
#   "type": "Copy",
#   "inputs": [
#     {
#       "referenceName": "SourceDataset", # Defines connection to source (e.g., BL
#       "type": "DatasetReference"
#     }
#   ],
#   "outputs": [
#     {
#       "referenceName": "BronzeLakeDataset", # Defines connection to ADLS Gen2 (B
#       "type": "DatasetReference"
#     }
#   ],
# }
```

```

# "typeProperties": {
#   "source": {
#     "type": "DelimitedTextSource", # Or other source type
#     "formatSettings": {
#       "type": "DelimitedTextReadSettings"
#     }
#   },
#   "sink": {
#     "type": "DelimitedTextSink", # Or other sink type (e.g., Parquet)
#     "storeSettings": {
#       "type": "AzureBlobFSWriteSettings"
#     }
#   }
# }

# 3. Process: Azure Data Lake Storage (Bronze) to Azure Databricks (using Spark)
# Read data from the Bronze layer
bronze_path = "abfss://<your_container>@<your_storage_account>.dfs.core.windows.net"
df_bronze = spark.read.format("parquet").load(bronze_path)

# Perform transformations
df_silver = df_bronze.select("column1", "column2", expr("transform_function(column1, column2)"))

# 4. Process: Azure Databricks to Azure Data Lake Storage (Silver) using Delta Lake
silver_path = "abfss://<your_container>@<your_storage_account>.dfs.core.windows.net"
df_silver.write.format("delta").mode("overwrite").save(silver_path)

# Using Delta Table API (Recommended)
from delta.tables import *

deltaTableSilver = DeltaTable.forPath(spark, silver_path)
deltaTableSilver.alias("oldData") \
    .merge(
        df_silver.alias("newData"),
        "oldData.column1 = newData.column1"
    ) \
    .whenMatchedUpdate(set = { "column2": "newData.column2" }) \
    .whenNotMatchedInsertAll() \
    .execute()

# 4. Process: Azure Databricks to Azure Data Lake Storage (Gold) using Delta Lake
gold_path = "abfss://<your_container>@<your_storage_account>.dfs.core.windows.net/"
df_gold = df_silver.groupBy("transformed_column").agg(count("*").alias("record_count"))
df_gold.write.format("delta").mode("overwrite").save(gold_path)

# 5. Serve: Azure Databricks to Azure Machine Learning
# This typically involves training a model in Databricks and then registering it in MLflow
# Example Python code in Databricks:

```

```

from sklearn.linear_model import LogisticRegression
from mlflow import MlflowClient
import mlflow

# Load training data
train_df = spark.read.format("delta").load(silver_path)
X_train = train_df.select("feature1", "feature2").toPandas()
y_train = train_df.select("target").toPandas()

# Train a model
model = LogisticRegression()
model.fit(X_train, y_train.values.ravel())

# Log the model with MLflow
with mlflow.start_run():
    mlflow.sklearn.log_model(model, "logistic_regression_model")
    run_id = mlflow.active_run().info.run_uuid

# Register the model in Azure Machine Learning
# Requires Azure ML SDK and connection to your AML workspace.
# Example Azure CLI command:
# az ml model register --name my_logistic_model --path runs://{run_id}/logistic_reg

# 6. Process: Azure Data Lake Storage (Gold) to Azure Databricks SQL Analytics
# Connect to your Databricks SQL Warehouse
# Create an external table pointing to the Gold layer
# Example SQL in Databricks SQL Analytics:
"""
CREATE TABLE IF NOT EXISTS gold_table
USING DELTA
LOCATION 'abfss://<your_container>@<your_storage_account>.dfs.core.windows.net/gol
"""

# 7. Serve: Azure Databricks SQL Analytics to Power BI
# This is configured within Power BI Desktop by connecting to the
# Databricks SQL Analytics endpoint.
# Steps in Power BI:
# 1. Get Data -> Azure -> Azure Databricks
# 2. Enter Server Hostname and HTTP Path from your Databricks SQL Warehouse connec
# 3. Choose data connectivity mode (Import or DirectQuery).
# 4. Select the 'gold_table' or write a custom SQL query.

# 8. Serve: Azure Data Lake Storage (Gold) to Azure Synapse Analytics
# Connect to your Synapse Analytics workspace
# Create an external table pointing to the Gold layer
# Example SQL in Synapse Analytics (SQL Pool):
"""
CREATE EXTERNAL TABLE gold_synapse_table
(

```

```

        transformed_column VARCHAR(100),
        record_count INT
    )
WITH (
    LOCATION = 'abfss://<your_container>@<your_storage_account>.dfs.core.windows.n
    DATA_SOURCE = [Your Data Source Name], -- Configure this in Synapse
    FILE_FORMAT = [Parquet Format Name]      -- Configure this in Synapse
);
"""

```

9. Monitor and Govern: Azure Purview, Azure DevOps, Azure Key Vault, Azure Activ
These services are used for governance, security, deployment, monitoring, and co
Code examples are less relevant here as these services are largely configured
through their respective interfaces or infrastructure-as-code deployments.

Example Azure Monitor query to check Databricks cluster status (Conceptual):
Azure CLI command:
az monitor metrics list --resource <databricks_cluster_resource_id> --metric "cp

Example of accessing secrets from Azure Key Vault in Databricks:
Requires setting up a secret scope backed by Azure Key Vault.
dbutils.secrets.get(scope = "<your_keyvault_scope>", key = "<your_secret_name>")

```

print("Code snippets illustrating the data flow are provided above.")

```