## ****1. Data Producer: Creating a Python Script to Generate JSON Data****

### ****Step 1: Install Python on Your Laptop****

* **What is Python?** It's a programming language that helps you write scripts (like instructions) for your computer.
* **How to Install:**
  1. Open your web browser (like Chrome or Firefox).
  2. Go to [python.org](https://www.python.org/downloads/" \t "_new).
  3. Click the big yellow button that says "Download Python".
  4. Once downloaded, open the installer and follow the instructions (just click "Next" until it's done).

### ****Step 2: Open a Text Editor****

* **What is a Text Editor?** It's a program where you can write your Python script.
* **Options:**
  + **Windows:** Use "Notepad" or download "Visual Studio Code" from [code.visualstudio.com](https://code.visualstudio.com/" \t "_new).
  + **Mac/Linux:** Use "TextEdit" or "Visual Studio Code".

### ****Step 3: Write the Python Script****

**Open your text editor.**

**Type the following code:**

python

Copy code

import jsonimport randomfrom datetime import datetimeimport time

def generate\_data():

data = {

"id": random.randint(1, 1000),

"value": random.uniform(10.0, 100.0),

"timestamp": datetime.now().isoformat()

}

return data

while True:

sample\_data = generate\_data()

json\_data = json.dumps(sample\_data)

print(json\_data)

time.sleep(5) # Wait for 5 seconds before generating next data

**Save the File:**

* + Click on "File" > "Save As".
  + Name it data\_producer.py.
  + Choose a folder you can easily find, like "Documents".

### ****Step 4: Run the Python Script****

1. **Open Command Prompt (Windows) or Terminal (Mac/Linux).**
2. **Navigate to the Folder:**
   * Type cd Documents (or the folder where you saved the script) and press Enter.
3. **Run the Script:**
   * Type python data\_producer.py and press Enter.
   * You should see JSON data printed every 5 seconds!

## ****2. Message Queue: Setting Up Azure Event Hubs****

### ****Step 1: Log into Azure Portal****

* Open your web browser and go to [Azure Portal](https://portal.azure.com/" \t "_new).
* Enter your Azure account email and password, then click "Sign In".

### ****Step 2: Create an Event Hubs Namespace****

1. **Click on "Create a resource":**
   * It's a big "+ Create a resource" button on the top left.
2. **Search for "Event Hubs" in the search bar and select it.**
3. **Click "Create".**
4. **Fill in the Details:**
   * **Subscription:** Choose your subscription.
   * **Resource group:** Click "Create new", type a name like DataPipelineRG, and click "OK".
   * **Namespace Name:** Enter a unique name, like datapipelinenamespace.
   * **Location:** Choose a location close to you, like East US.
   * **Pricing Tier:** Choose the basic option for now.
5. **Click "Review + create", then "Create".**
6. **Wait a few minutes** until Azure sets up your Event Hubs namespace.

### ****Step 3: Create an Event Hub****

1. **Go to your new Event Hubs namespace** by clicking "Go to resource".
2. **Click on "Event Hubs" in the left menu.**
3. **Click "+ Event Hub"** to create a new Event Hub.
4. **Name Your Event Hub:** Enter a name like samplehub.
5. **Leave other settings as default** and click "Create".

### ****Step 4: Get Connection String****

1. **In the Event Hubs Namespace**, click on "Shared access policies" under "Settings" in the left menu.
2. **Click on "RootManageSharedAccessKey".**
3. **Copy the "Connection string-primary key":**
   * Click the copy icon next to it.

### ****Step 5: Update Your Python Script to Send Data to Event Hub****

**Install Azure Event Hub Library:**

* + Open Command Prompt or Terminal.
  + Type pip install azure-eventhub and press Enter.

**Modify Your** data\_producer.py **Script:**

* + Open data\_producer.py in your text editor.
  + Update the script to send data to Event Hub:

python

Copy code

import jsonimport randomfrom datetime import datetimeimport timefrom azure.eventhub import EventHubProducerClient, EventData

# Replace with your connection string and event hub name

CONNECTION\_STR = "Your\_Event\_Hub\_Connection\_String"

EVENTHUB\_NAME = "samplehub"

producer = EventHubProducerClient.from\_connection\_string(conn\_str=CONNECTION\_STR, eventhub\_name=EVENTHUB\_NAME)

def generate\_data():

data = {

"id": random.randint(1, 1000),

"value": random.uniform(10.0, 100.0),

"timestamp": datetime.now().isoformat()

}

return data

while True:

sample\_data = generate\_data()

json\_data = json.dumps(sample\_data)

event\_data = EventData(json\_data)

with producer:

producer.send\_batch([event\_data])

print(f"Sent: {json\_data}")

time.sleep(5) # Wait for 5 seconds before sending next data

**Paste Your Connection String:**

* + Replace "Your\_Event\_Hub\_Connection\_String" with the connection string you copied earlier.

**Save the File**.

### ****Step 6: Run the Updated Python Script****

* In Command Prompt or Terminal, navigate to your script's folder.
* Type python data\_producer.py and press Enter.
* Now, your script is sending data to Azure Event Hub!

## ****3. Serverless SQL Solution: Setting Up Azure SQL Database****

### ****Step 1: Create an Azure SQL Database****

1. **In Azure Portal**, click on "Create a resource".
2. **Search for "Azure SQL"** and select "Azure SQL".
3. **Choose "Azure SQL Database"** and click "Create".
4. **Fill in the Details:**
   * **Subscription:** Your subscription.
   * **Resource group:** Use DataPipelineRG (or the one you created earlier).
   * **Database name:** Enter DataPipelineDB.
   * **Server:** Click "Create new".
     + **Server name:** Enter something like datapipeline-server.
     + **Server admin login:** Choose a username, like sqladmin.
     + **Password:** Choose a strong password and remember it!
     + **Location:** Same as your Event Hub, like East US.
     + Click "OK".
   * **Compute + storage:** Click "Configure database".
     + Choose "Serverless" tier.
     + Set "Auto-pause" to 1 hour to save costs.
     + Click "Apply".
5. **Click "Review + create", then "Create"**.
6. **Wait a few minutes** for Azure to set up the SQL Database.

### ****Step 2: Configure Firewall to Allow Access****

1. **Go to Your SQL Server** (click on "Go to resource" after creation).
2. **Click on "Networking"** in the left menu.
3. **Add Your Client IP:**
   * Under "Firewall rules", click "+ Add your client IP".
   * This lets your computer connect to the database.
4. **Click "Save"**.

### ****Step 3: Connect to Your SQL Database****

1. **Download SQL Server Management Studio (SSMS)** if you don't have it:
   * Go to SSMS Download.
   * Download and install it by following the on-screen instructions.
2. **Open SSMS**.
3. **Connect to Server:**
   * **Server name:** Enter your server name, like datapipeline-server.database.windows.net.
   * **Authentication:** Choose "SQL Server Authentication".
   * **Login:** Enter your admin username (e.g., sqladmin).
   * **Password:** Enter your password.
   * Click "Connect".

### ****Step 4: Create a Table to Store Data****

1. **In SSMS**, find your database (DataPipelineDB) in the left panel.
2. **Right-click on "Tables"** > "New" > "Table".
3. **Define Columns:**
   * **id:** INT, Not Null
   * **value:** FLOAT, Not Null
   * **timestamp:** DATETIME, Not Null
4. **Save the Table:**
   * Click "Save" (the floppy disk icon).
   * Name it SampleData.
5. **Now, your table is ready to receive data!**

## ****4. Data Consumer: Setting Up an Azure VM and Python Script****

### ****Step 1: Create an Azure Virtual Machine (VM)****

1. **In Azure Portal**, click on "Create a resource".
2. **Search for "Ubuntu Server 22.04"** and select it.
3. **Click "Create".**
4. **Fill in the Details:**
   * **Subscription:** Your subscription.
   * **Resource group:** Use DataPipelineRG.
   * **VM name:** Enter DataConsumerVM.
   * **Region:** Same as others, like East US.
   * **Image:** Should be "Ubuntu Server 22.04 LTS".
   * **Size:** Choose B1s (1 vCPU, 1 GB RAM).
   * **Authentication type:** Choose "Password".
     + **Username:** Enter a name, like azureuser.
     + **Password:** Create a strong password.
   * **Inbound ports:** Check "SSH (22)" to connect later.
5. **Click "Review + create", then "Create"**.
6. **Wait a few minutes** for Azure to set up your VM.

### ****Step 2: Connect to Your VM via SSH****

1. **Find Your VM's Public IP Address:**
   * Go to your VM's page in Azure Portal.
   * Look for the "Public IP address" at the top.
2. **Open Terminal (Mac/Linux) or PuTTY (Windows):**
   * **Mac/Linux:** Use Terminal.
   * **Windows:** Download PuTTY from [putty.org](https://www.putty.org/" \t "_new) if you don't have it.
3. **Connect Using SSH:**
   * **Mac/Linux:** Type ssh azureuser@Your\_VM\_IP and press Enter.
   * **Windows (PuTTY):**
     + Open PuTTY.
     + Enter your VM's IP in the "Host Name" field.
     + Click "Open".
4. **Enter Your Password** when prompted.

### ****Step 3: Install Python and Required Libraries on VM****

1. **Update Packages:**
   * Type sudo apt update and press Enter.
2. **Install Python and pip:**
   * Type sudo apt install python3 python3-pip -y and press Enter.
3. **Install Azure Event Hub and pyodbc Libraries:**
   * Type pip3 install azure-eventhub pyodbc and press Enter.
4. **Install ODBC Driver for SQL Server:**
   * Type the following commands one by one and press Enter after each:

bash

Copy code

sudo su

curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -

curl https://packages.microsoft.com/config/ubuntu/22.04/prod.list > /etc/apt/sources.list.d/mssql-release.listexit

sudo apt update

sudo ACCEPT\_EULA=Y apt install msodbcsql17 -y

sudo apt install unixodbc-dev -y

### ****Step 4: Create the Python Consumer Script****

**Create a New File:**

* + Type nano data\_consumer.py and press Enter.

**Type the Following Code:**

python

Copy code

import jsonimport pyodbcfrom azure.eventhub import EventHubConsumerClient

# Replace with your connection string and event hub name

CONNECTION\_STR = "Your\_Event\_Hub\_Connection\_String"

EVENTHUB\_NAME = "samplehub"

CONSUMER\_GROUP = "$Default"

# SQL Database connection details

SERVER = "datapipeline-server.database.windows.net"

DATABASE = "DataPipelineDB"

USER = "sqladmin"

PASSWORD = "Your\_SQL\_Password"

def on\_event(partition\_context, event):

data = json.loads(event.body\_as\_str())

print(f"Received: {data}")

# Connect to SQL Database

conn = pyodbc.connect(

f'DRIVER={{ODBC Driver 17 for SQL Server}};SERVER={SERVER};DATABASE={DATABASE};UID={USER};PWD={PASSWORD}'

)

cursor = conn.cursor()

# Insert data into SampleData table

cursor.execute(

"INSERT INTO SampleData (id, value, timestamp) VALUES (?, ?, ?)",

data["id"],

data["value"],

data["timestamp"]

)

conn.commit()

cursor.close()

conn.close()

partition\_context.update\_checkpoint(event)

client = EventHubConsumerClient.from\_connection\_string(

CONNECTION\_STR,

CONSUMER\_GROUP,

EVENTHUB\_NAME

)

try:

with client:

client.receive(

on\_event=on\_event,

starting\_position="-1" # "-1" is from the beginning of the stream

)except KeyboardInterrupt:

print("Stopped receiving.")

**Replace Placeholders:**

* + "Your\_Event\_Hub\_Connection\_String"**:** Use the same connection string you used in the producer script.
  + "Your\_SQL\_Password"**:** Enter the password you set for your SQL admin user.

**Save and Exit:**

* + Press CTRL + X, then Y, and press Enter to save the file.

### ****Step 5: Run the Python Consumer Script****

1. **In the SSH Terminal**, navigate to where you saved data\_consumer.py (usually home directory).
2. **Run the Script:**
   * Type python3 data\_consumer.py and press Enter.
3. **Watch It Work:**
   * The script will receive data from Event Hub and insert it into your SQL Database.
   * You should see messages like Received: {'id': 123, 'value': 45.67, 'timestamp': '2024-10-19T12:34:56.789123'}.

## ****5. Verify Data in Azure SQL Database****

### ****Step 1: Open SSMS on Your Computer****

* **Launch SSMS** and connect to your Azure SQL Database as you did earlier.

### ****Step 2: Query the Data****

**Open a New Query Window:**

* + Click on "New Query" at the top.

**Type the Following SQL Command:**

sql

Copy code

SELECT \* FROM SampleData;

**Execute the Query:**

* + Click "Execute" or press F5.

**See Your Data:**

* + You should see all the JSON data that your Python scripts have sent and inserted!

## ****6. Visualize Your Data (Optional Bonus Step)****

### ****Step 1: Use Azure Power BI or Another Visualization Tool****

* **Power BI** is a great tool to create charts and graphs from your data.
* You can download it from [powerbi.microsoft.com](https://powerbi.microsoft.com/" \t "_new).

### ****Step 2: Connect Power BI to Your Azure SQL Database****

1. **Open Power BI** and click on "Get Data".
2. **Choose "SQL Server"** and click "Connect".
3. **Enter Server and Database Details:**
   * **Server:** datapipeline-server.database.windows.net
   * **Database:** DataPipelineDB
4. **Authentication:**
   * Choose "Database" and enter your SQL admin username and password.
5. **Select Your Table:**
   * Choose SampleData and load the data.
6. **Create Visuals:**
   * Use the tools in Power BI to create charts, graphs, and dashboards from your data!

## ****Congratulations! 🎉****

You've successfully built an Azure Data Pipeline:

1. **Produced Data** with a Python script.
2. **Queued Data** using Azure Event Hubs.
3. **Stored Data** in a serverless Azure SQL Database.
4. **Consumed Data** with a Python script on an Azure VM.

Now you can see your data flowing from one place to another, just like a real data pipeline! Great job!