

Event Hub Connection & Usage Guide

1■■ Purpose of the Connection

The Event Hub connection allows all project members to send or receive real-time traffic data through Azure Event Hub. It ensures synchronization between the Data Generation, Analytics, and Visualization teams, using a shared communication channel.

2■■ General Structure

The project setup follows this structure:

- 1 **Namespace:** traffic-namespace-uae
- 2 **Event Hub:** traffic-events
- 3 **Teams:** Producers (Data Simulation) and Consumers (Analytics & Visualization)

3■■ Script Execution Instructions

- 1 Each team member should use the shared connection string and Event Hub name in the script.
- 2 Run the script using Python after installing required libraries (azure-eventhub).
- 3 Example environment variables:
- 4

```
CONNECTION_STR = "Endpoint=sb://traffic-namespace-uae.servicebus.windows.net/;SharedAccessKeyName=ManagePolicy;SharedAccessKey=KLbsnli6uP9GtM9ovBiZ7NX1wroU4KKRh+AEhIvUyYI=;EntityPath=traffic-events"
```
- 5

```
EVENTHUB_NAME = "traffic-events"
```

4■■ Accessing the Connection String in Azure Portal

- 1 Log in to Azure Portal.
- 2 Search for 'Event Hubs' in the top search bar.
- 3 Select the Namespace: traffic-namespace-uae.
- 4 From the left panel, open Event Hubs → select 'traffic-events'.
- 5 Navigate to 'Shared access policies'.
- 6 Choose 'ManagePolicy' and click 'Show' next to 'Connection string—primary key'.
- 7 Copy the connection string and paste it into your environment file or script.

5■■ Security Warning

■■ The 'ManagePolicy' grants full control (send, listen, and manage). Do not share it publicly or outside the project team to avoid unauthorized access or data loss.

6■■ Verifying Data Reception

- 1 In the Azure Portal, open the Event Hub (traffic-events).
- 2 Go to 'Metrics' in the left panel.

- 3 Select metrics like 'Incoming Messages' or 'Outgoing Messages' to confirm data flow.
- 4 You should see an increase in message count when the script is running.