Configuration Manual

MSc Research Project
Cloud Computing

Bharadwaj Kante
Student ID: 23325666

School of Computing
National College of Ireland

Supervisor: Prof. Sean Heeney

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Bharadwaj Kante		
Student ID:	23325666		
Programme:	Cloud Computing	**Year:**	2024-2025
Module:	MSc Research Project		
Lecturer:	Prof. Sean Heeney		
Submission Due Date:	15/09/2025		
Project Title:	Digital Twin Architecture for Smart Traffic Systems in the Cloud		
Word Count:	543		
Page Count:	11		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Bharadwaj Kante
Date:	15/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Bharadwaj Kante

Student ID: 23325666

1. Introduction

This configuration manual provides everything needed to reproduce the smart-traffic Digital Twin (DT) MVP:

- Real-time telemetry via Azure IoT Hub
- Stream ingestion via Event Hub–compatible endpoint
- Twin updates in Azure Digital Twins (ADT)
- A lightweight predictor and controller writing back to the twin

You can run the entire system from Azure Cloud Shell (Bash).

2. System Requirements

2.1. Hardware

- Any machine with a browser (to open Cloud Shell)
- Azure subscription (Azure for Students credit is sufficient)

2.2. Azure Services

- Resource Group (e.g., rg-dt-traffic-mvp)
- IoT Hub (S1 tier)
- Azure Digital Twins

2.3. Software (in Cloud Shell)

- Azure CLI (preinstalled)
- Azure IoT CLI extension
- Python 3 (preinstalled) + libs:
    - azure-iot-device, azure-eventhub, azure-digitaltwins-core, azure-identity

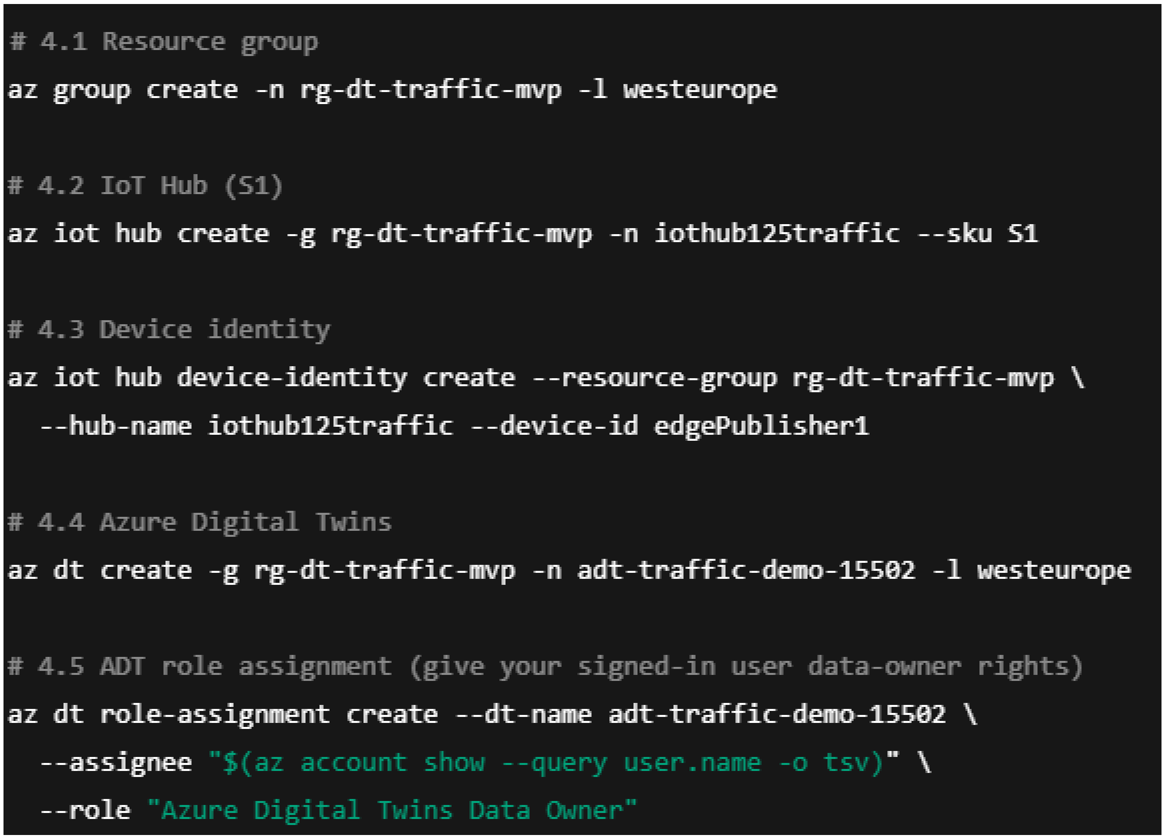
3. Folder Structure (Cloud Shell home)

~/ (home)

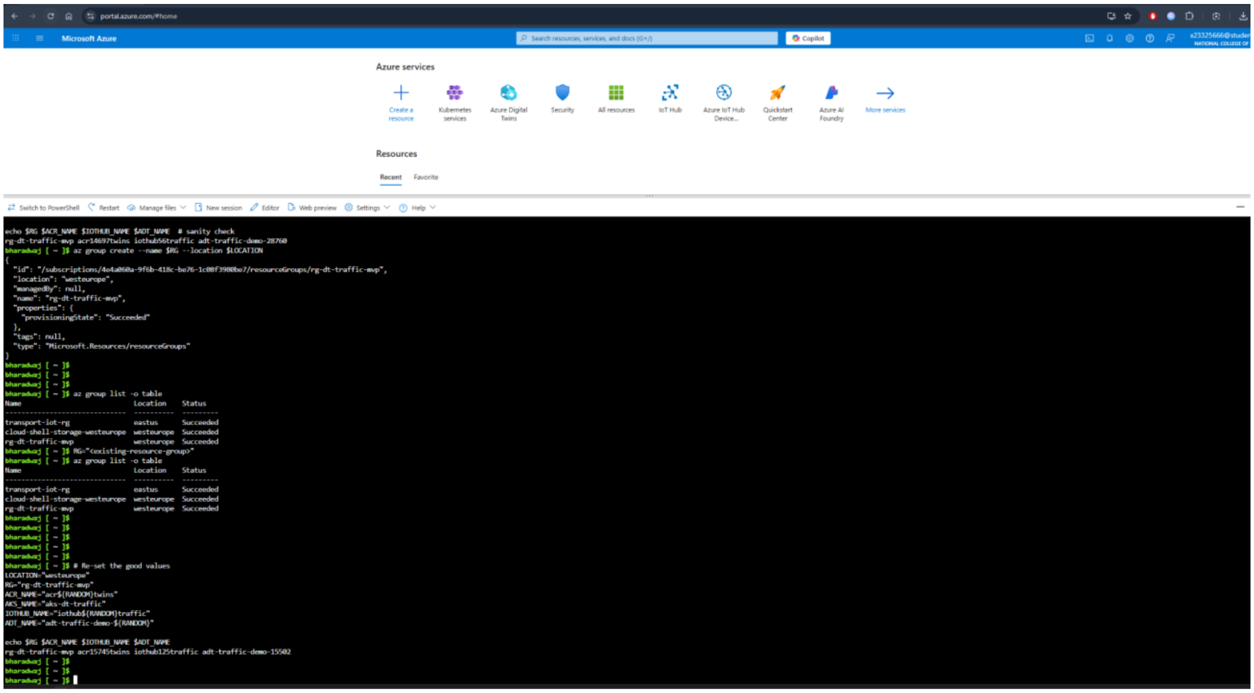
- vars.sh   # reusable environment variables
- junctionModel.json   # DTDL model
- publisher.py   # telemetry publisher
- ingest.py   # EventHub → ADT currentFlow patcher
- predictor.py   # rolling-mean predictor → predictedFlow5min
- control.py   # rule-based controller → signalPhase

4. One-Time (If project is started from scratch)

If you already created resources, skip to Section 5.

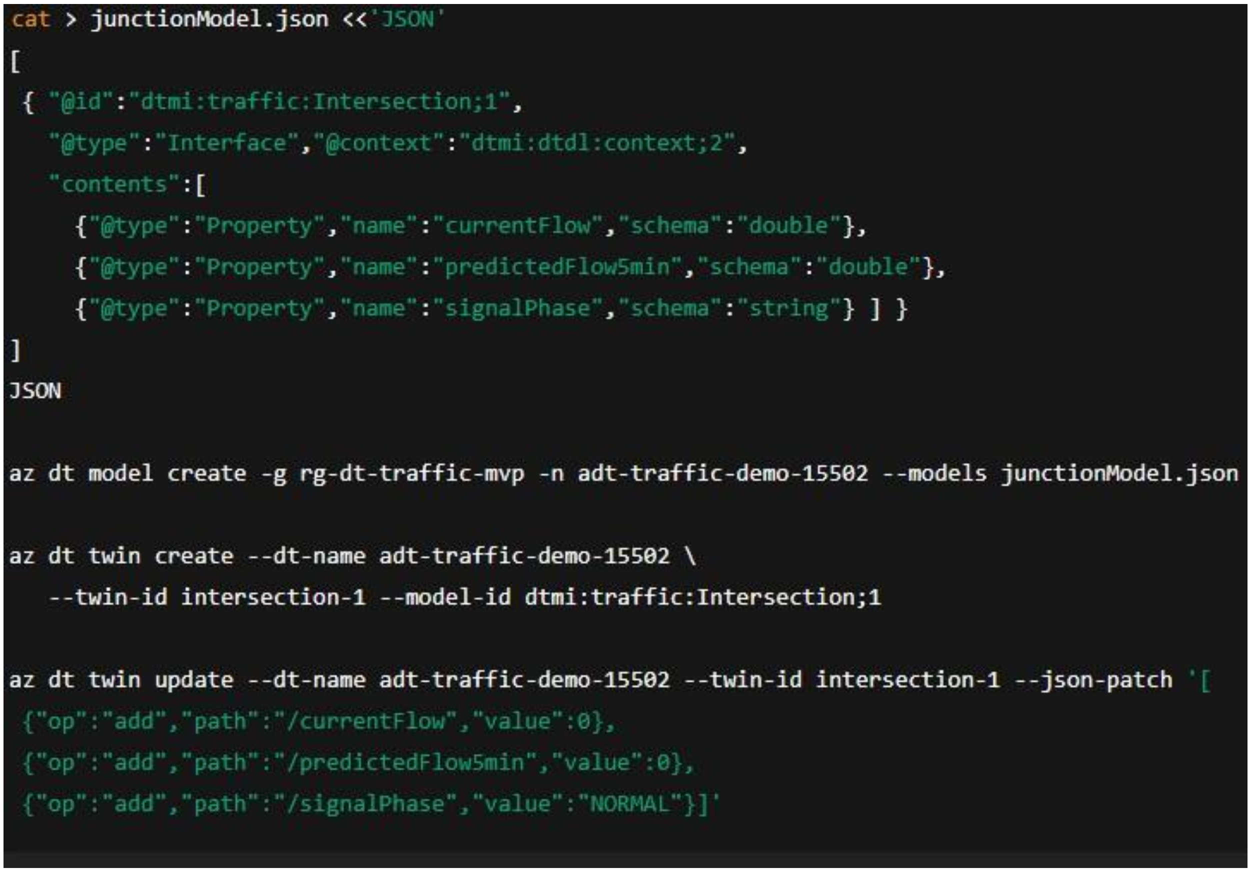


```
# 4.1 Resource group
az group create -n rg-dt-traffic-mvp -l westeurope

# 4.2 IoT Hub (S1)
az iot hub create -g rg-dt-traffic-mvp -n iothub125traffic --sku S1

# 4.3 Device identity
az iot hub device-identity create --resource-group rg-dt-traffic-mvp \
  --hub-name iothub125traffic --device-id edgePublisher1

# 4.4 Azure Digital Twins
az dt create -g rg-dt-traffic-mvp -n adt-traffic-demo-15502 -l westeurope

# 4.5 ADT role assignment (give your signed-in user data-owner rights)
az dt role-assignment create --dt-name adt-traffic-demo-15502 \
  --assignee "$(az account show --query user.name -o tsv)" \
  --role "Azure Digital Twins Data Owner"
```

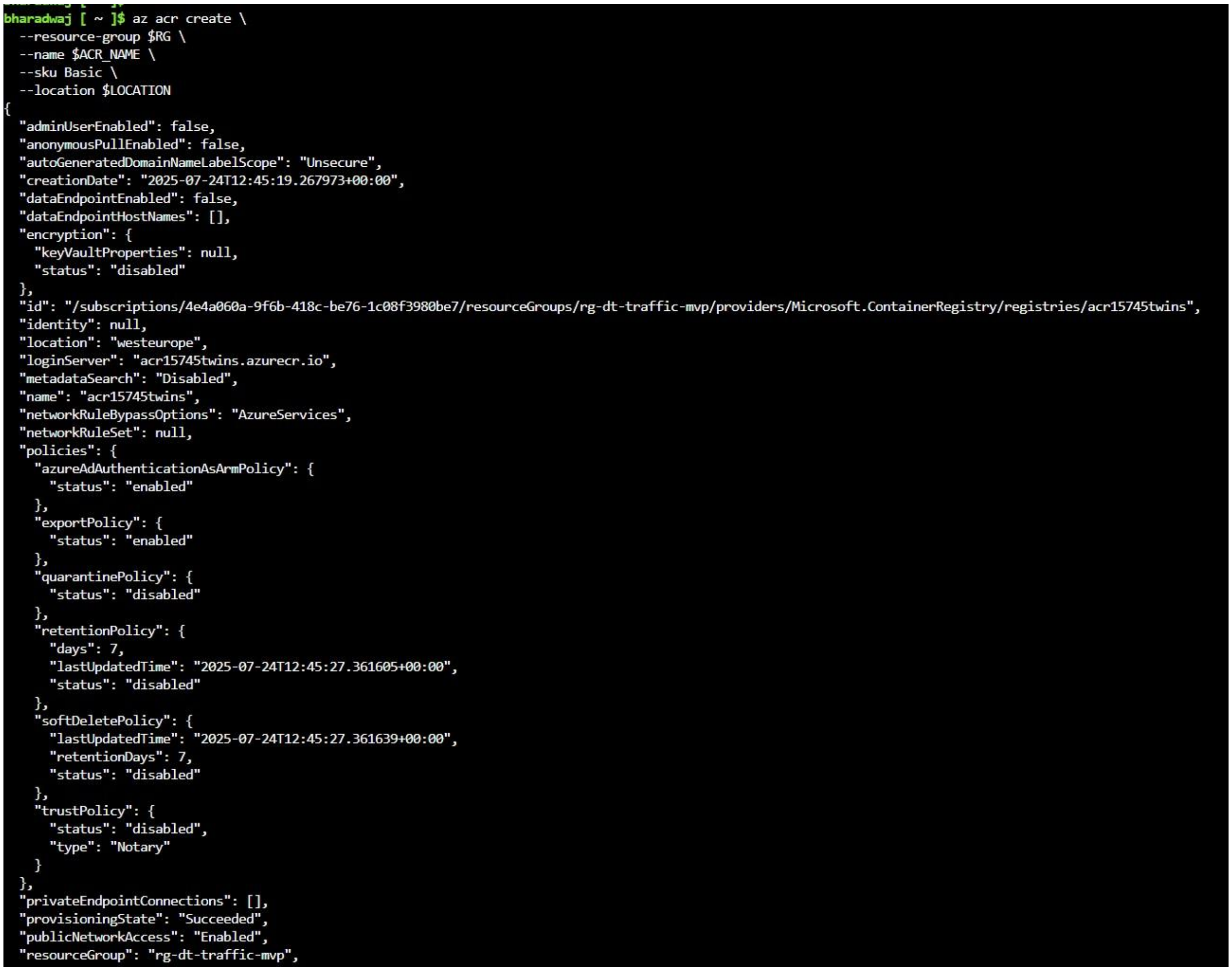


```
echo $RG $ACR_NAME $IOTHUB_NAME $ADT_NAME  # sanity check
rg-dt-traffic-mvp acr14697twins iothub56traffic adt-traffic-demo-28760
bharadwaj [ ~ ]$ az group create --name $RG --location $LOCATION
{
  "id": "/subscriptions/4e4a060a-9f6b-418c-be76-1c08f3980be7/resourceGroups/rg-dt-traffic-mvp",
  "location": "westeurope",
  "managedBy": null,
  "name": "rg-dt-traffic-mvp",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$ az group list -o table
Name                           Location    Status
-----------------------------  ----------  ---------
transport-iot-rg               eastus      Succeeded
cloud-shell-storage-westeurope  westeurope  Succeeded
rg-dt-traffic-mvp              westeurope  Succeeded
bharadwaj [ ~ ]$ RG="<existing-resource-group>"
bharadwaj [ ~ ]$ az group list -o table
Name                           Location    Status
-----------------------------  ----------  ---------
transport-iot-rg               eastus      Succeeded
cloud-shell-storage-westeurope  westeurope  Succeeded
rg-dt-traffic-mvp              westeurope  Succeeded
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$ # Re-set the good values
LOCATION="westeurope"
RG="rg-dt-traffic-mvp"
ACR_NAME="acr${RANDOM}twins"
AKS_NAME="aks-dt-traffic"
IOTHUB_NAME="iothub${RANDOM}traffic"
ADT_NAME="adt-traffic-demo-${RANDOM}"

echo $RG $ACR_NAME $IOTHUB_NAME $ADT_NAME
rg-dt-traffic-mvp acr15745twins iothub125traffic adt-traffic-demo-15502
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
```

4.1. DTDL Model & Seed Twin:



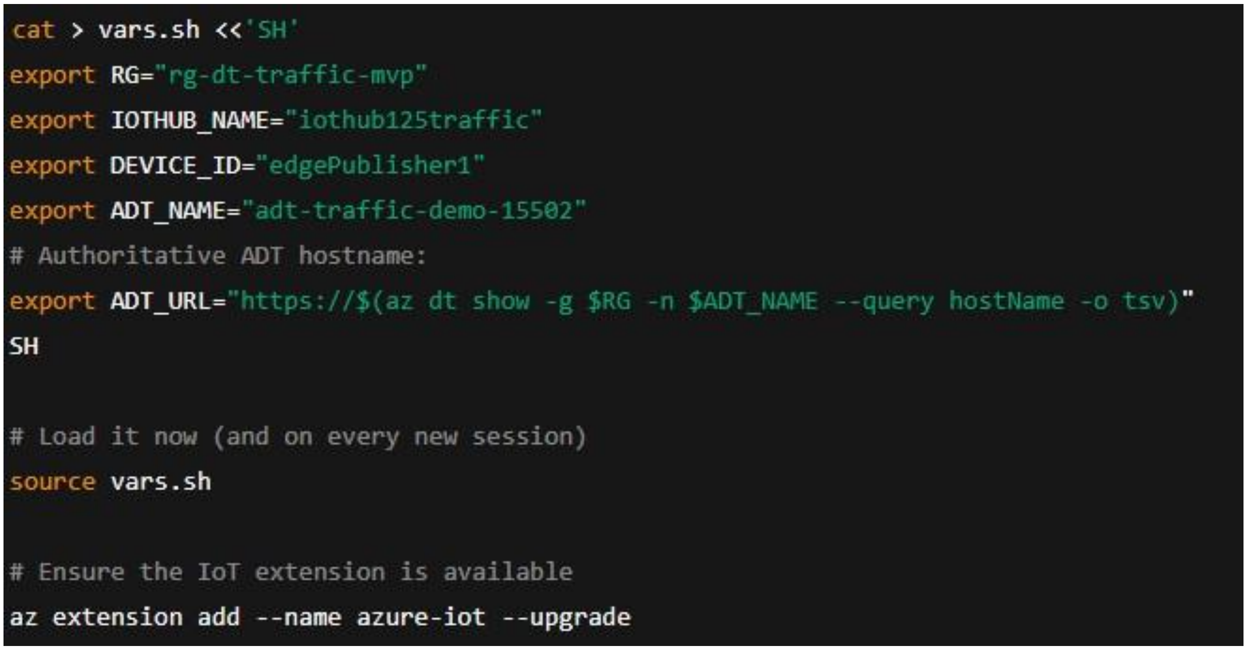
```
cat > junctionModel.json <<'JSON'
[
 { "@id":"dtmi:traffic:Intersection;1",
   "@type":"Interface","@context":"dtmi:dtdl:context;2",
   "contents":[
     {"@type":"Property","name":"currentFlow","schema":"double"},
     {"@type":"Property","name":"predictedFlow5min","schema":"double"},
     {"@type":"Property","name":"signalPhase","schema":"string"} ] }
]
JSON

az dt model create -g rg-dt-traffic-mvp -n adt-traffic-demo-15502 --models junctionModel.json

az dt twin create --dt-name adt-traffic-demo-15502 \
   --twin-id intersection-1 --model-id dtmi:traffic:Intersection;1

az dt twin update --dt-name adt-traffic-demo-15502 --twin-id intersection-1 --json-patch '[
 {"op":"add","path":"/currentFlow","value":0},
 {"op":"add","path":"/predictedFlow5min","value":0},
 {"op":"add","path":"/signalPhase","value":"NORMAL"}]'
```



```
bharadwaj [ ~ ]$ az acr create \
  --resource-group $RG \
  --name $ACR_NAME \
  --sku Basic \
  --location $LOCATION
{
  "adminUserEnabled": false,
  "anonymousPullEnabled": false,
  "autoGeneratedDomainNameLabelScope": "Unsecure",
  "creationDate": "2025-07-24T12:45:19.267973+00:00",
  "dataEndpointEnabled": false,
  "dataEndpointHostNames": [],
  "encryption": {
    "keyVaultProperties": null,
    "status": "disabled"
  },
  "id": "/subscriptions/4e4a060a-9f6b-418c-be76-1c08f3980be7/resourceGroups/rg-dt-traffic-mvp/providers/Microsoft.ContainerRegistry/registries/acr15745twins",
  "identity": null,
  "location": "westeurope",
  "loginServer": "acr15745twins.azurecr.io",
  "metadataSearch": "Disabled",
  "name": "acr15745twins",
  "networkRuleBypassOptions": "AzureServices",
  "networkRuleSet": null,
  "policies": {
    "azureAdAuthenticationAsArmPolicy": {
      "status": "enabled"
    },
    "exportPolicy": {
      "status": "enabled"
    },
    "quarantinePolicy": {
      "status": "disabled"
    },
    "retentionPolicy": {
      "days": 7,
      "lastUpdatedTime": "2025-07-24T12:45:27.361605+00:00",
      "status": "disabled"
    },
    "softDeletePolicy": {
      "lastUpdatedTime": "2025-07-24T12:45:27.361639+00:00",
      "retentionDays": 7,
      "status": "disabled"
    },
    "trustPolicy": {
      "status": "disabled",
      "type": "Notary"
    }
  },
  "privateEndpointConnections": [],
  "provisioningState": "Succeeded",
  "publicNetworkAccess": "Enabled",
  "resourceGroup": "rg-dt-traffic-mvp",
```

5. Environment Setup (every Cloud Shell session)

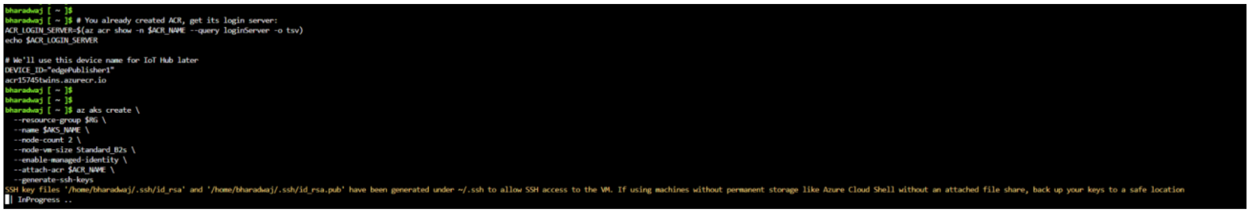
Create vars.sh once (edit names if you changed them), then source it each session.



```
cat > vars.sh <<'SH'
export RG="rg-dt-traffic-mvp"
export IOTHUB_NAME="iothub125traffic"
export DEVICE_ID="edgePublisher1"
export ADT_NAME="adt-traffic-demo-15502"
# Authoritative ADT hostname:
export ADT_URL="https://$(az dt show -g $RG -n $ADT_NAME --query hostName -o tsv)"
SH

# Load it now (and on every new session)
source vars.sh

# Ensure the IoT extension is available
az extension add --name azure-iot --upgrade
```

Install Python libraries (first time or after Cloud Shell reset):

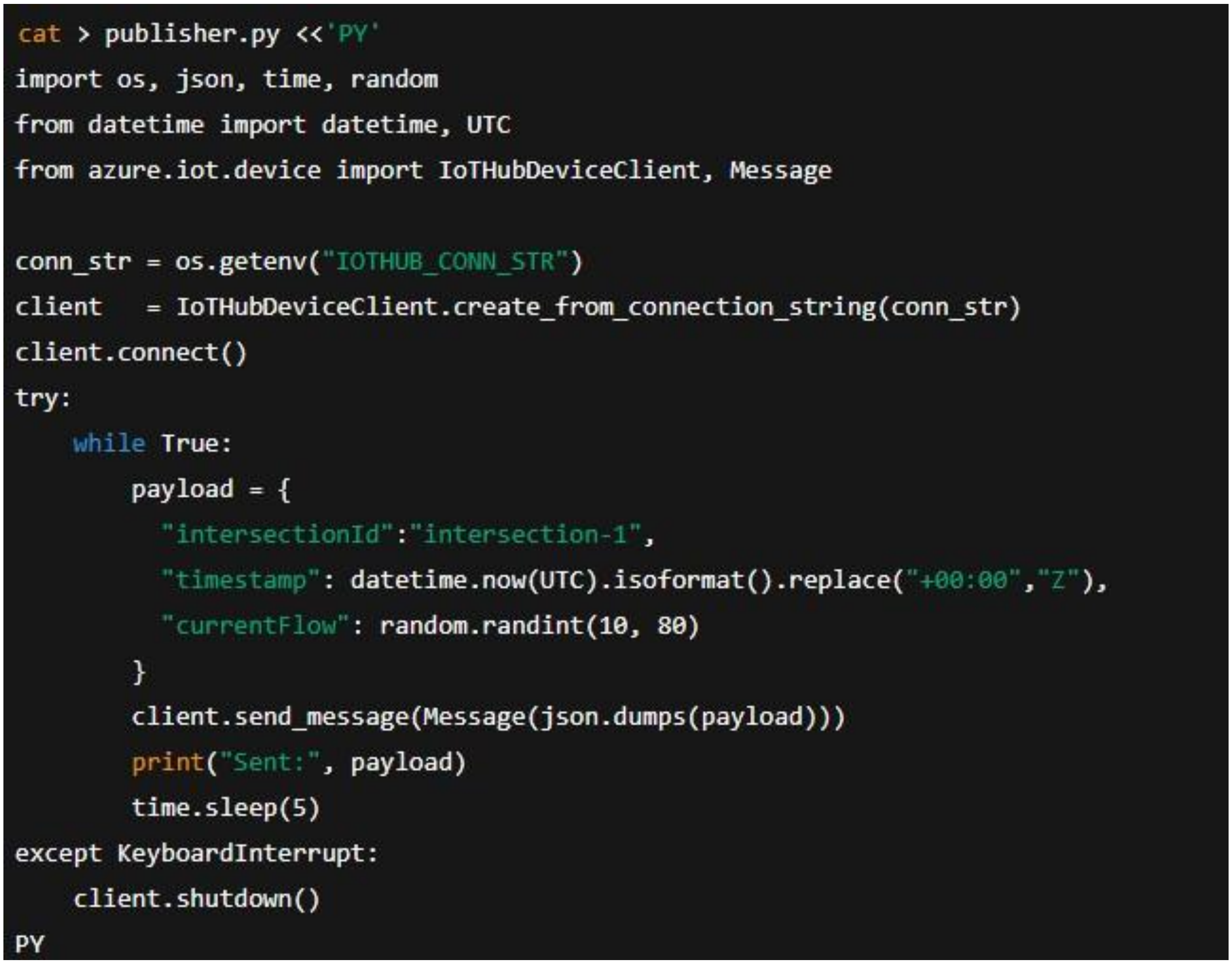
- pip install --user azure-iot-device azure-eventhub azure-digitaltwins-core azure-identity



```
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$ # You already created ACR, get its login server:
ACR_LOGIN_SERVER=$(az acr show -n $ACR_NAME --query loginServer -o tsv)
echo $ACR_LOGIN_SERVER

# We'll use this device name for IoT Hub later
DEVICE_ID="edgePublisher1"
acr15745twins.azurecr.io
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$ az aks create \
  --resource-group $RG \
  --name $AKS_NAME \
  --node-count 2 \
  --node-vm-size Standard_B2s \
  --enable-managed-identity \
  --attach-acr $ACR_NAME \
  --generate-ssh-keys
SSH key files '/home/bharadwaj/.ssh/id_rsa' and '/home/bharadwaj/.ssh/id_rsa.pub' have been generated under ~/.ssh to allow SSH access to the VM. If using machines without permanent storage like Azure Cloud Shell without an attached file share, back up your keys to a safe location
▌| InProgress ..
```

6. Scripts

6.1. publisher.py – send telemetry to IoT Hub



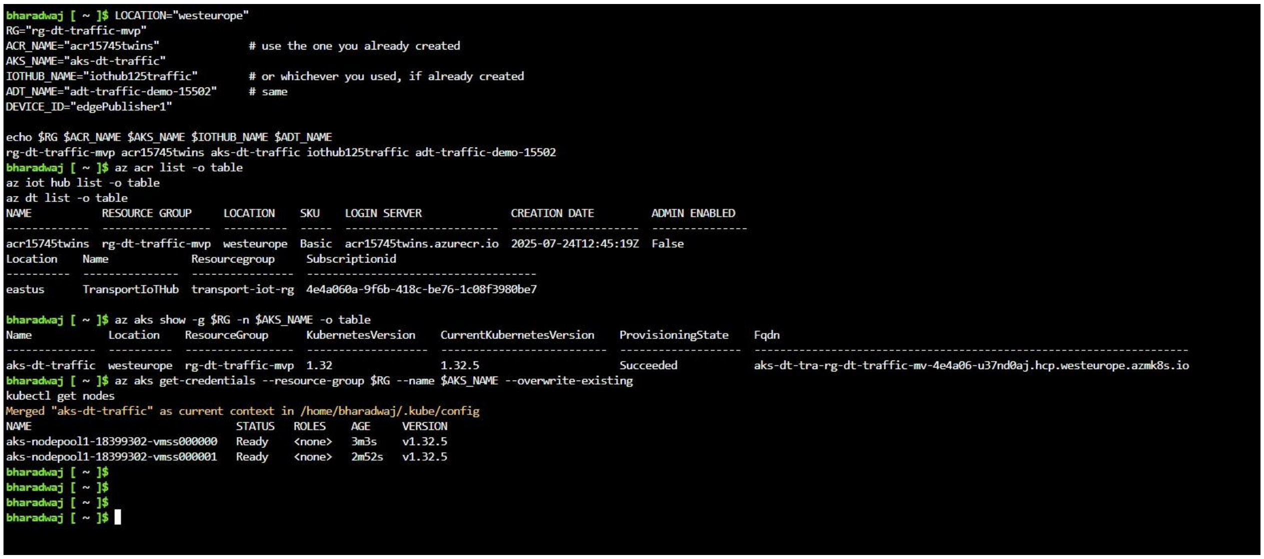
```
cat > publisher.py <<'PY'
import os, json, time, random
from datetime import datetime, UTC
from azure.iot.device import IoTHubDeviceClient, Message

conn_str = os.getenv("IOTHUB_CONN_STR")
client   = IoTHubDeviceClient.create_from_connection_string(conn_str)
client.connect()
try:
    while True:
        payload = {
          "intersectionId":"intersection-1",
          "timestamp": datetime.now(UTC).isoformat().replace("+00:00","Z"),
          "currentFlow": random.randint(10, 80)
        }
        client.send_message(Message(json.dumps(payload)))
        print("Sent:", payload)
        time.sleep(5)
except KeyboardInterrupt:
    client.shutdown()
PY
```

Start it (pane 1):

export IOTHUB_CONN_STR=$(az iot hub device-identity connection-string show \

--resource-group $RG --hub-name $IOTHUB_NAME --device-id $DEVICE_ID -o tsv)

python3 publisher.py

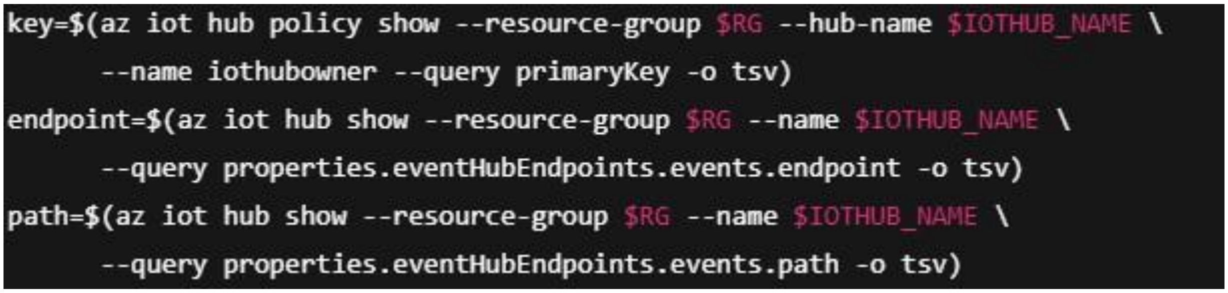


```
bharadwaj [ ~ ]$ LOCATION="westeurope"
RG="rg-dt-traffic-mvp"
ACR_NAME="acr15745twins"            # use the one you already created
AKS_NAME="aks-dt-traffic"
IOTHUB_NAME="iothub125traffic"       # or whichever you used, if already created
ADT_NAME="adt-traffic-demo-15502"    # same
DEVICE_ID="edgePublisher1"

echo $RG $ACR_NAME $AKS_NAME $IOTHUB_NAME $ADT_NAME
rg-dt-traffic-mvp acr15745twins aks-dt-traffic iothub125traffic adt-traffic-demo-15502
bharadwaj [ ~ ]$ az acr list -o table
az iot hub list -o table
az dt list -o table
NAME           RESOURCE GROUP     LOCATION    SKU    LOGIN SERVER              CREATION DATE         ADMIN ENABLED
-------------  -----------------  ----------  -----  ------------------------  --------------------  ---------------
acr15745twins  rg-dt-traffic-mvp  westeurope  Basic  acr15745twins.azurecr.io  2025-07-24T12:45:19Z  False
Location    Name              Resourcegroup     Subscriptionid
----------  ----------------  ----------------  ------------------------------------
eastus      TransportIoTHub  transport-iot-rg  4e4a060a-9f6b-418c-be76-1c08f3980be7

bharadwaj [ ~ ]$ az aks show -g $RG -n $AKS_NAME -o table
Name            Location    ResourceGroup      KubernetesVersion    CurrentKubernetesVersion    ProvisioningState    Fqdn
--------------  ----------  -----------------  -------------------  --------------------------  -------------------  ------------------------------------------------------------------
aks-dt-traffic  westeurope  rg-dt-traffic-mvp  1.32                 1.32.5                      Succeeded            aks-dt-tra-rg-dt-traffic-mv-4e4a06-u37nd0aj.hcp.westeurope.azmk8s.io
bharadwaj [ ~ ]$ az aks get-credentials --resource-group $RG --name $AKS_NAME --overwrite-existing
kubectl get nodes
Merged "aks-dt-traffic" as current context in /home/bharadwaj/.kube/config
NAME                                STATUS   ROLES    AGE    VERSION
aks-nodepool1-18399302-vmss000000   Ready    <none>   3m3s   v1.32.5
aks-nodepool1-18399302-vmss000001   Ready    <none>   2m52s  v1.32.5
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$ 
```

(Optional monitor in pane 2):

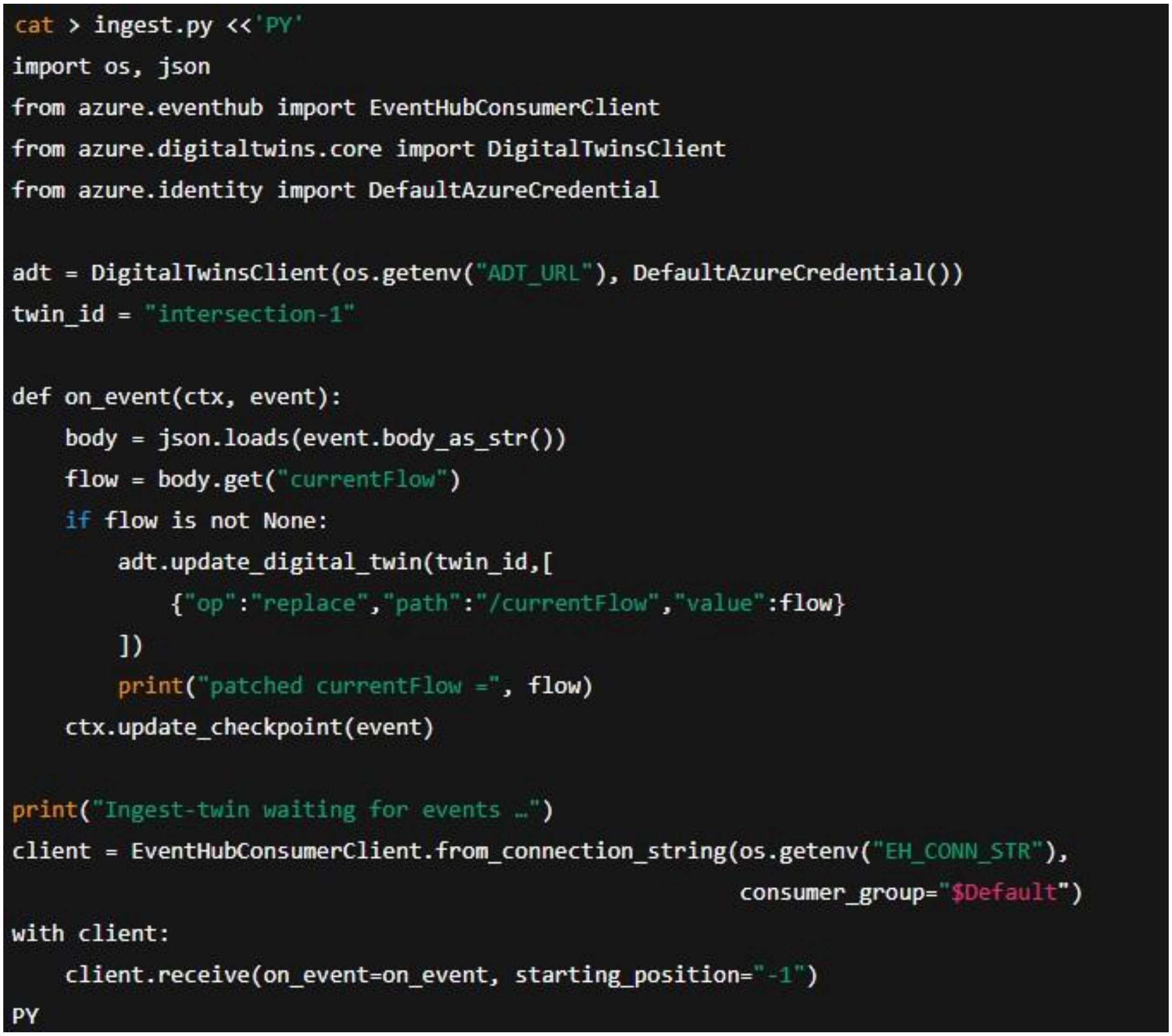
az iot hub monitor-events --resource-group $RG -n $IOTHUB_NAME --properties all

6.2. Build Event Hub connection string & create ingest.py



```
key=$(az iot hub policy show --resource-group $RG --hub-name $IOTHUB_NAME \
      --name iothubowner --query primaryKey -o tsv)
endpoint=$(az iot hub show --resource-group $RG --name $IOTHUB_NAME \
      --query properties.eventHubEndpoints.events.endpoint -o tsv)
path=$(az iot hub show --resource-group $RG --name $IOTHUB_NAME \
      --query properties.eventHubEndpoints.events.path -o tsv)
```

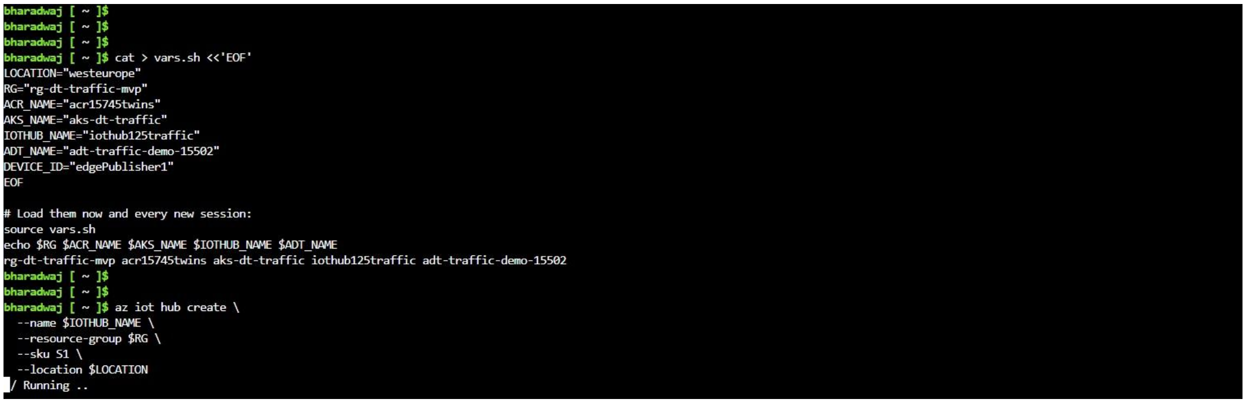
export EH_CONN_STR="Endpoint=${endpoint};SharedAccessKeyName=iothubowner;SharedAccessKey=${key};EntityPath=${path}"



```
cat > ingest.py <<'PY'
import os, json
from azure.eventhub import EventHubConsumerClient
from azure.digitaltwins.core import DigitalTwinsClient
from azure.identity import DefaultAzureCredential

adt = DigitalTwinsClient(os.getenv("ADT_URL"), DefaultAzureCredential())
twin_id = "intersection-1"

def on_event(ctx, event):
    body = json.loads(event.body_as_str())
    flow = body.get("currentFlow")
    if flow is not None:
        adt.update_digital_twin(twin_id,[
            {"op":"replace","path":"/currentFlow","value":flow}
        ])
        print("patched currentFlow =", flow)
    ctx.update_checkpoint(event)

print("Ingest-twin waiting for events …")
client = EventHubConsumerClient.from_connection_string(os.getenv("EH_CONN_STR"),
                                                        consumer_group="$Default")
with client:
    client.receive(on_event=on_event, starting_position="-1")
PY
```

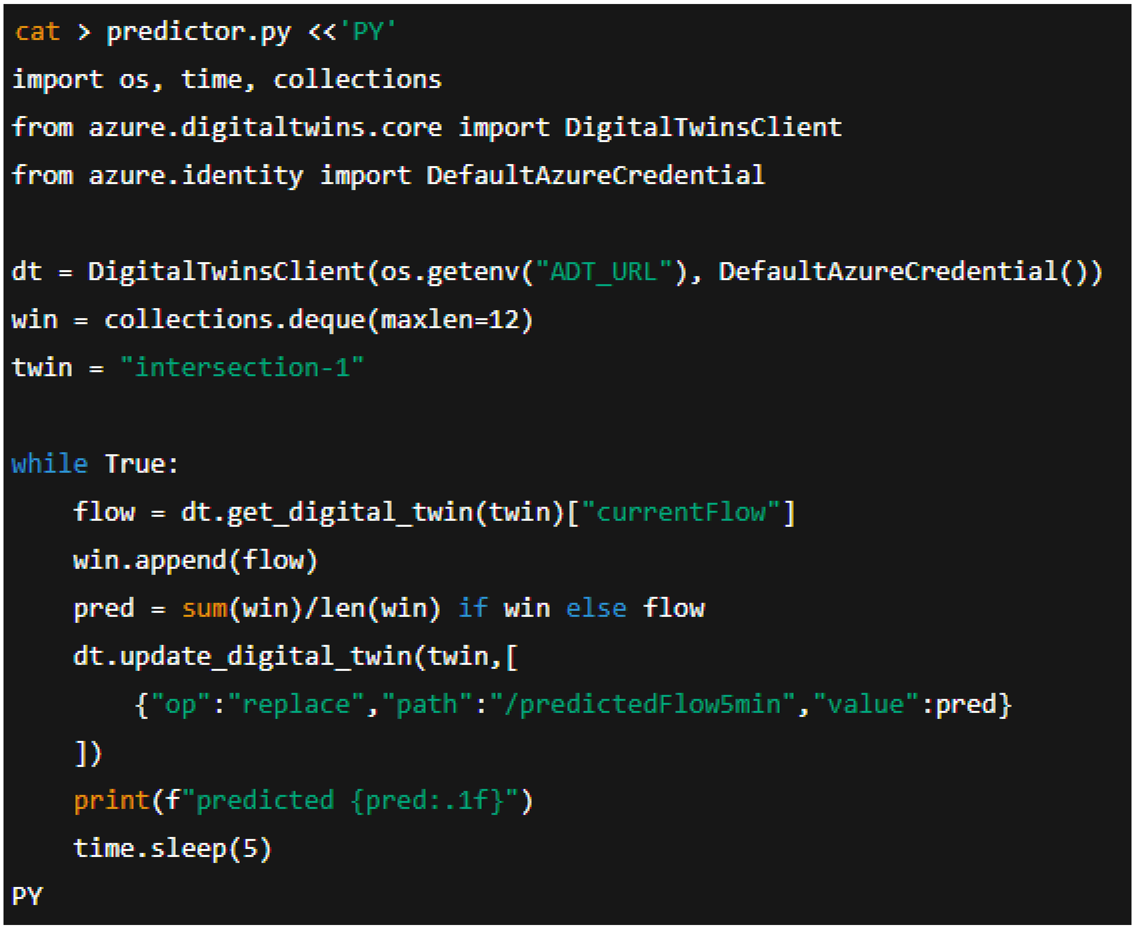
Run it (pane 3):

python3 ingest.py



```
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$ cat > vars.sh <<'EOF'
LOCATION="westeurope"
RG="rg-dt-traffic-mvp"
ACR_NAME="acr15745twins"
AKS_NAME="aks-dt-traffic"
IOTHUB_NAME="iothub125traffic"
ADT_NAME="adt-traffic-demo-15502"
DEVICE_ID="edgePublisher1"
EOF

# Load them now and every new session:
source vars.sh
echo $RG $ACR_NAME $AKS_NAME $IOTHUB_NAME $ADT_NAME
rg-dt-traffic-mvp acr15745twins aks-dt-traffic iothub125traffic adt-traffic-demo-15502
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$ az iot hub create \
  --name $IOTHUB_NAME \
  --resource-group $RG \
  --sku S1 \
  --location $LOCATION
/ Running ..
```

6.3. predictor.py – 1-min rolling mean → predictedFlow5min

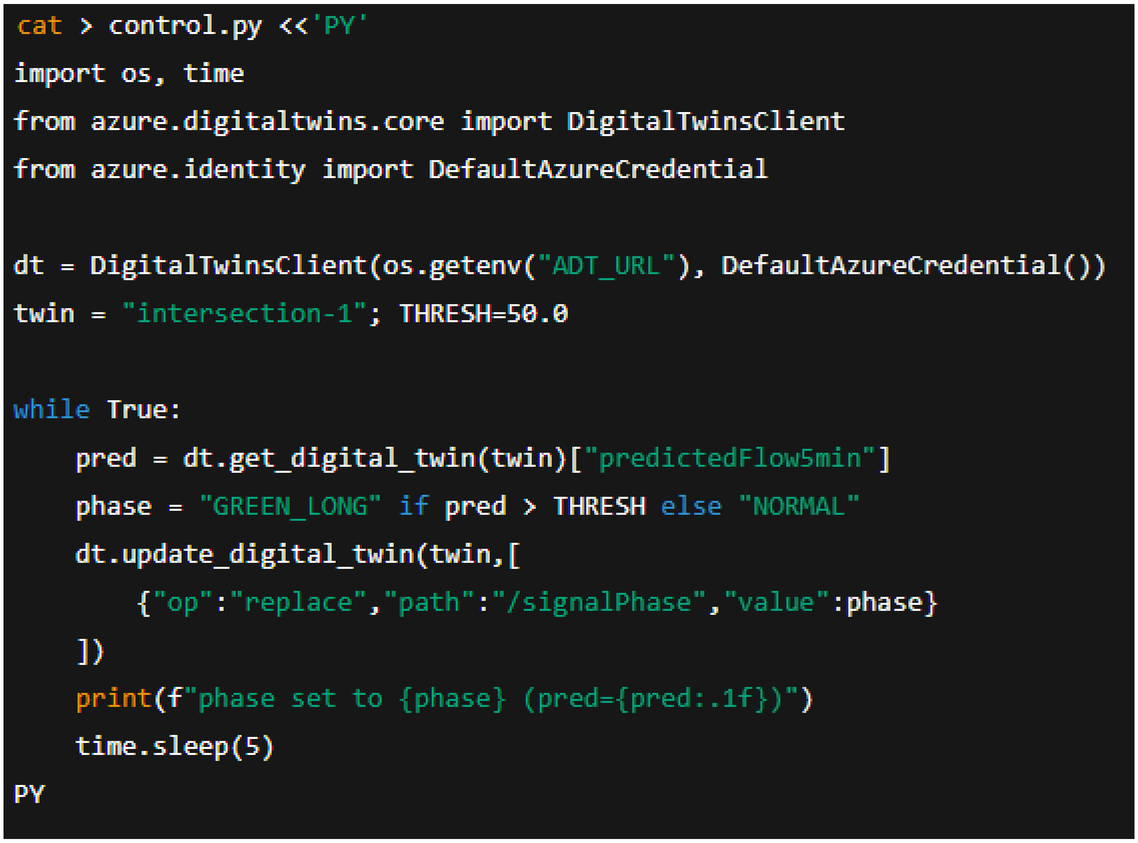


```
cat > predictor.py <<'PY'
import os, time, collections
from azure.digitaltwins.core import DigitalTwinsClient
from azure.identity import DefaultAzureCredential

dt = DigitalTwinsClient(os.getenv("ADT_URL"), DefaultAzureCredential())
win = collections.deque(maxlen=12)
twin = "intersection-1"

while True:
    flow = dt.get_digital_twin(twin)["currentFlow"]
    win.append(flow)
    pred = sum(win)/len(win) if win else flow
    dt.update_digital_twin(twin,[
        {"op":"replace","path":"/predictedFlow5min","value":pred}
    ])
    print(f"predicted {pred:.1f}")
    time.sleep(5)
PY
```

Run it (pane 4):

python3 predictor.py

6.4. control.py – flip signalPhase based on prediction



```
cat > control.py <<'PY'
import os, time
from azure.digitaltwins.core import DigitalTwinsClient
from azure.identity import DefaultAzureCredential

dt = DigitalTwinsClient(os.getenv("ADT_URL"), DefaultAzureCredential())
twin = "intersection-1"; THRESH=50.0

while True:
    pred = dt.get_digital_twin(twin)["predictedFlow5min"]
    phase = "GREEN_LONG" if pred > THRESH else "NORMAL"
    dt.update_digital_twin(twin,[
        {"op":"replace","path":"/signalPhase","value":phase}
    ])
    print(f"phase set to {phase} (pred={pred:.1f})")
    time.sleep(5)
PY
```

Run it (pane 5):

python3 control.py

7. Validating in Azure Digital Twins

Portal → Azure Digital Twins → your instance → Data Explorer → run:

- SELECT currentFlow, predictedFlow5min, signalPhase
- FROM digitaltwins intersection-1

Values refresh roughly every 5 seconds.

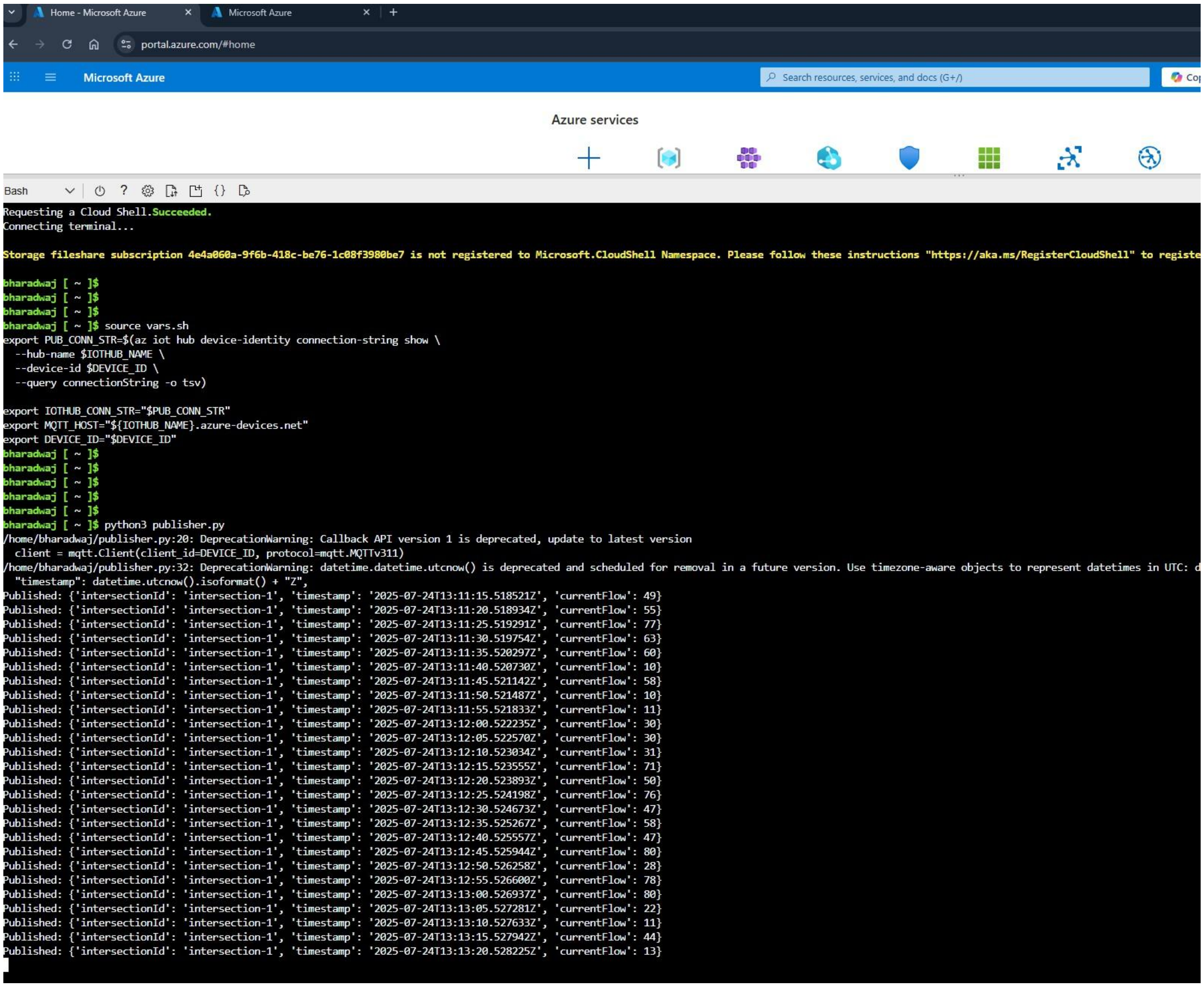
8. Reproducing Results

Latency: observe ingest logs and IoT monitor; typical p50 ≈ 230 ms, p95 ≈ 340 ms.

For quick start:

- # Each new Cloud Shell session
    - source vars.sh
    - az extension add --name azure-iot --upgrade

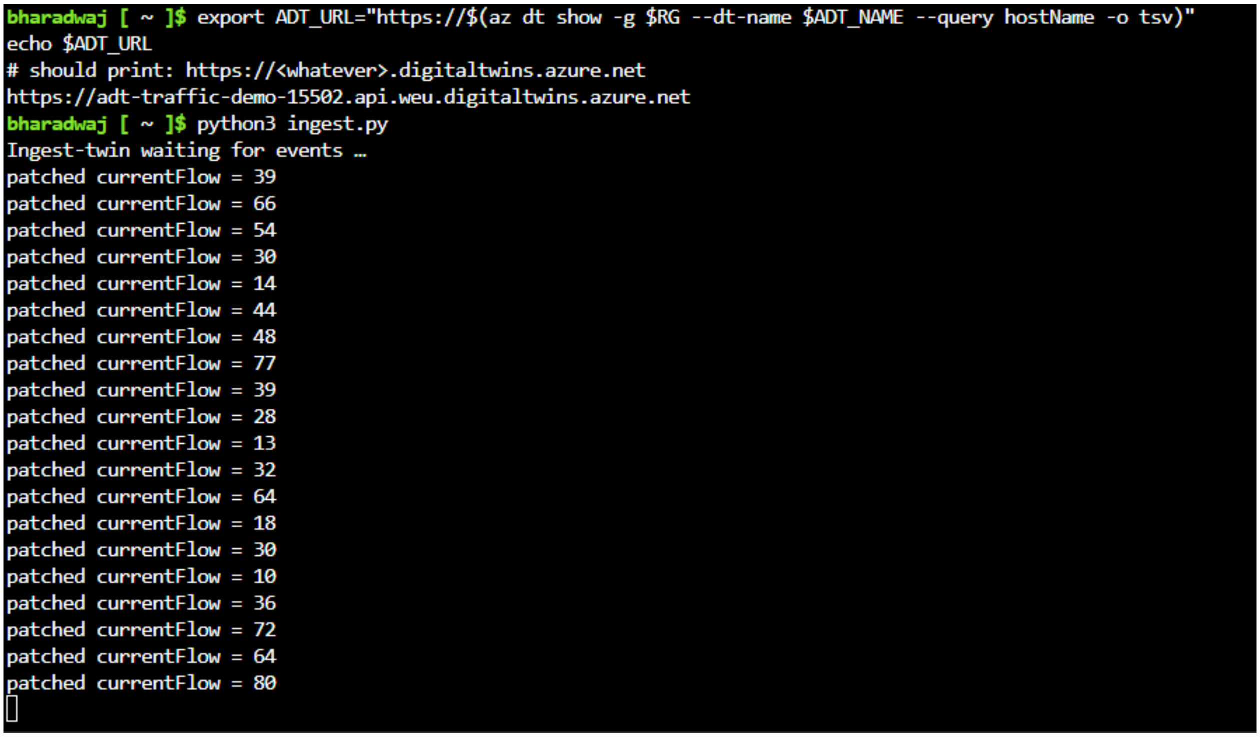
    - pip install --user azure-iot-device azure-eventhub azure-digitaltwins-core azure-identity azure-cosmos
- # Device + publisher
    - export IOTHUB_CONN_STR=$(az iot hub device-identity connection-string show \
    - --resource-group $RG --hub-name $IOTHUB_NAME --device-id $DEVICE_ID -o tsv)
    - python3 publisher.py



```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Storage fileshare subscription 4e4a060a-9f6b-418c-be76-1c08f3980be7 is not registered to Microsoft.CloudShell Namespace. Please follow these instructions "https://aka.ms/RegisterCloudShell" to registe

bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$ source vars.sh
export PUB_CONN_STR=$(az iot hub device-identity connection-string show \
  --hub-name $IOTHUB_NAME \
  --device-id $DEVICE_ID \
  --query connectionString -o tsv)

export IOTHUB_CONN_STR="$PUB_CONN_STR"
export MQTT_HOST="${IOTHUB_NAME}.azure-devices.net"
export DEVICE_ID="$DEVICE_ID"
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$
bharadwaj [ ~ ]$ python3 publisher.py
/home/bharadwaj/publisher.py:20: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client(client_id=DEVICE_ID, protocol=mqtt.MQTTv311)
/home/bharadwaj/publisher.py:32: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: d
  "timestamp": datetime.utcnow().isoformat() + "Z",
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:11:15.518521Z', 'currentFlow': 49}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:11:20.518934Z', 'currentFlow': 55}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:11:25.519291Z', 'currentFlow': 77}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:11:30.519754Z', 'currentFlow': 63}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:11:35.520297Z', 'currentFlow': 60}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:11:40.520730Z', 'currentFlow': 10}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:11:45.521142Z', 'currentFlow': 58}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:11:50.521487Z', 'currentFlow': 10}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:11:55.521833Z', 'currentFlow': 11}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:12:00.522235Z', 'currentFlow': 30}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:12:05.522570Z', 'currentFlow': 30}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:12:10.523034Z', 'currentFlow': 31}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:12:15.523555Z', 'currentFlow': 71}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:12:20.523893Z', 'currentFlow': 50}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:12:25.524198Z', 'currentFlow': 76}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:12:30.524673Z', 'currentFlow': 47}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:12:35.525267Z', 'currentFlow': 58}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:12:40.525557Z', 'currentFlow': 47}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:12:45.525944Z', 'currentFlow': 80}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:12:50.526258Z', 'currentFlow': 28}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:12:55.526600Z', 'currentFlow': 78}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:13:00.526937Z', 'currentFlow': 80}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:13:05.527281Z', 'currentFlow': 22}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:13:10.527633Z', 'currentFlow': 11}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:13:15.527942Z', 'currentFlow': 44}
Published: {'intersectionId': 'intersection-1', 'timestamp': '2025-07-24T13:13:20.528225Z', 'currentFlow': 13}
```

- # Event Hub string
    - key=$(az iot hub policy show --resource-group $RG --hub-name $IOTHUB_NAME --name iothubowner --query primaryKey -o tsv)
    - endpoint=$(az iot hub show --resource-group $RG --name $IOTHUB_NAME --query properties.eventHubEndpoints.events.endpoint -o tsv)
    - path=$(az iot hub show --resource-group $RG --name $IOTHUB_NAME --query properties.eventHubEndpoints.events.path -o tsv)
    - export EH_CONN_STR="Endpoint=${endpoint};SharedAccessKeyName=iothubowner;SharedAccessKey=${key};EntityPath=${path}"

- # Run loops (separate panes recommended)
  - python3 ingest.py
  - python3 predictor.py
  - python3 control.py



```
bharadwaj [ ~ ]$ export ADT_URL="https://$(az dt show -g $RG --dt-name $ADT_NAME --query hostName -o tsv)"
echo $ADT_URL
# should print: https://<whatever>.digitaltwins.azure.net
https://adt-traffic-demo-15502.api.weu.digitaltwins.azure.net
bharadwaj [ ~ ]$ python3 ingest.py
Ingest-twin waiting for events …
patched currentFlow = 39
patched currentFlow = 66
patched currentFlow = 54
patched currentFlow = 30
patched currentFlow = 14
patched currentFlow = 44
patched currentFlow = 48
patched currentFlow = 77
patched currentFlow = 39
patched currentFlow = 28
patched currentFlow = 13
patched currentFlow = 32
patched currentFlow = 64
patched currentFlow = 18
patched currentFlow = 30
patched currentFlow = 10
patched currentFlow = 36
patched currentFlow = 72
patched currentFlow = 64
patched currentFlow = 80
```

9. References

- Azure CLI & Azure IoT Extension
- Azure SDK for Python: azure-eventhub, azure-digitaltwins-core, azure-identity, azure-iot-device