**Step-by-Step: NiFi Setup in GUI Mode on Azure**

**Step 1: Create Azure Linux VM (Ubuntu 20.04)**

**Via Azure Portal:**

1. Go to portal.azure.com
2. Navigate to **Virtual Machines** → **Create**
3. Fill the details:
   * **Name**: nifi-vm
   * **Region**: East US (or your region)
   * **Image**: Ubuntu 20.04 LTS
   * **Size**: Standard\_DS2\_v2
   * **Authentication**: SSH or password
4. **Allow port 8080** under **Inbound Ports** for NiFi UI
5. Click **Review + Create** → **Create**

**Step 2: Connect to VM via Remote Desktop**

ssh azureuser@<your-vm-public-ip>

**Step 3: Install Java & NiFi**

Use [Windows Terminal]:

ssh azureuser@<your-vm-public-ip>

# Install Java

sudo apt update

sudo apt install openjdk-11-jdk -y

# Download NiFi

wget https://downloads.apache.org/nifi/1.26.0/nifi-1.26.0-bin.tar.gz

tar -xvzf nifi-1.26.0-bin.tar.gz

mv nifi-1.26.0 nifi

# Start NiFi

cd nifi/bin

./nifi.sh start

**Step 4: Open NiFi GUI in Your Browser**

1. Visit:  
   http://<your-vm-public-ip>:8080/nifi  
   *(Open port 8080 in Azure NSG if not done)*
2. You’ll see the NiFi visual canvas.

**Step 5: Build Your Flow in NiFi GUI**

**Drag Processors**

From left menu → Drag the following:

| **Processor Name** | **Purpose** |
| --- | --- |
| **GenerateFlowFile** | Simulate streaming transactions |
| **UpdateAttribute** | Set static fields like userId, location |
| **ReplaceText** | Replace with dynamic JSON transaction |
| **PutKafkaRecord\_2\_0** | Send message to Azure Event Hub (Kafka) |

**Configure Each Processor**

**1. GenerateFlowFile**

* Set Scheduling → Run every 3 seconds
* Set Custom Text to any dummy string ("trigger")

**2. ReplaceText**

* Strategy: Replace Entire Text
* Replacement Value:

json

{

"transactionId": "${UUID()}",

"cardNumber": "9876-XXXX-XXXX-4321",

"amount": ${math:random():multiply(100000):floor()},

"location": "${location:random('Delhi','Mumbai','London','NYC')}",

"timestamp": "${now():format('yyyy-MM-dd''T''HH:mm:ss''Z')}",

"userId": "${userId:random('U101','U102','U103')}"

}

**3. PutKafkaRecord\_2\_0**

* Kafka Brokers: fraud-ns.servicebus.windows.net:9093
* Topic: transactions
* Security Protocol: SASL\_SSL
* SASL Mechanism: PLAIN
* Username: $ConnectionString
* Password: <Your Event Hub connection string>
* Record Reader: JsonTreeReader
* Record Writer: JsonRecordSetWriter