**Apache Kafka Deployment Script:**

Deploying an **Apache Kafka Cluster using EC2 instances** is a common real-world scenario. Here's a **step-by-step guide** tailored for AWS EC2 deployment.

**⚙️ Overview**

We'll deploy:

* **3 EC2 instances** for Zookeeper (ZK1, ZK2, ZK3)
* **3 EC2 instances** for Kafka brokers (K1, K2, K3)

You can consolidate Zookeeper and Kafka on the same instances for simplicity (not recommended for production), or separate them for better performance and reliability.

**🧱 Prerequisites**

**✅ AWS Setup**

* AWS account
* SSH key pair
* VPC with:
  + Public or private subnets
  + Security group allowing traffic on:
    - TCP 22 (SSH)
    - TCP 2181 (Zookeeper)
    - TCP 9092 (Kafka)
    - TCP 2888, 3888 (Zookeeper peer communication)
    - Optional: TCP 8080 (Kafka UI)
* Amazon Linux 2 or Ubuntu 22.04 LTS AMIs

**🪜 Step-by-Step Deployment**

**Step 1: Launch EC2 Instances**

* Launch **6 EC2 instances** (t2.medium or larger).
* Use the same key pair for SSH.
* Name them clearly:
  + ZK1, ZK2, ZK3
  + K1, K2, K3

**Step 2: Install Java on All Instances**

SSH into each instance:

sudo yum update -y # Amazon Linux

sudo yum install java-11-openjdk -y

# or on Ubuntu

sudo apt update && sudo apt install openjdk-11-jdk -y

java -version

**Step 3: Create Kafka User (Optional)**

sudo useradd kafka -m

sudo passwd kafka

sudo su - kafka

**Step 4: Download Kafka on All Nodes**

wget https://downloads.apache.org/kafka/3.7.0/kafka\_2.13-3.7.0.tgz

tar -xvzf kafka\_2.13-3.7.0.tgz

cd kafka\_2.13-3.7.0

**Step 5: Configure Zookeeper on ZK1, ZK2, ZK3**

**Create a data directory**

mkdir -p ~/zookeeper/data

**Edit Zookeeper config**

nano config/zookeeper.properties

Change content:

properties

dataDir=/home/ec2-user/zookeeper/data

clientPort=2181

tickTime=2000

initLimit=5

syncLimit=2

server.1=ZK1\_PRIVATE\_IP:2888:3888

server.2=ZK2\_PRIVATE\_IP:2888:3888

server.3=ZK3\_PRIVATE\_IP:2888:3888

Set the ID on each:

echo "1" > ~/zookeeper/data/myid # change to 2 or 3 for ZK2 and ZK3

**Start Zookeeper**

bin/zookeeper-server-start.sh -daemon config/zookeeper.properties

**Step 6: Configure Kafka Brokers (on K1, K2, K3)**

**Create logs directory:**

mkdir -p ~/kafka-logs

**Edit Kafka config:**

nano config/server.properties

Change:

properties

broker.id=1 # use 2 for K2, 3 for K3

listeners=PLAINTEXT://PRIVATE\_IP:9092

log.dirs=/home/ec2-user/kafka-logs

zookeeper.connect=ZK1\_PRIVATE\_IP:2181,ZK2\_PRIVATE\_IP:2181,ZK3\_PRIVATE\_IP:2181

**Start Kafka:**

bin/kafka-server-start.sh -daemon config/server.properties

**Step 7: Test the Kafka Cluster**

On any Kafka node:

**Create a topic:**

bin/kafka-topics.sh --create \

--bootstrap-server PRIVATE\_IP:9092 \

--replication-factor 3 \

--partitions 3 \

--topic test-topic

**Produce:**

bin/kafka-console-producer.sh --broker-list PRIVATE\_IP:9092 --topic test-topic

**Consume:**

bin/kafka-console-consumer.sh --bootstrap-server PRIVATE\_IP:9092 --topic test-topic --from-beginning

**🛡️ Security Group Rules (example)**

| **Port** | **Protocol** | **Purpose** |
| --- | --- | --- |
| 22 | TCP | SSH |
| 2181 | TCP | Zookeeper client port |
| 2888 | TCP | Zookeeper peer comm |
| 3888 | TCP | Zookeeper leader election |
| 9092 | TCP | Kafka broker port |

Restrict access to internal IPs in production.

**📈 (Optional) Monitoring & UI**

You can run:

* **Kafka UI**: <https://github.com/provectus/kafka-ui>
* **Prometheus & Grafana** for Kafka metrics

**🧰 Helpful Automation**

Want a **Bash script** or **Terraform module** to automate EC2 + Kafka cluster provisioning? Let me know! I can generate them for you.