A.i. Blockchain
Artificial Intelligence

Using the AiB KafkaBlockchain Demo on AWS

- Login to the AWS console.
- Select the N. Virginia AWS region.
- Create VPC.
    - Navigate to the VPC service.
    - Launch the VPC wizard.
    - Select *VPC with a Single Public Subnet*.
    - IPv4 CIDR Block: *10.0.0.0/16*
    - VPC name: *KafkaBlockchain*
    - Otherwise accept the defaults for VPC with a Single Public Subnet
    - Click Create VPC
- Navigate to the EC2 service to Launch an AMI Instance.
- Navigate to the AMIs page (AMIs or My AMIs)
- Select the KafkaBlockchain AMI and click Select.
- Select the *t2.xlarge General Purpose instance type* - having 4 vCPUs and 16 GB RAM.
- Configure Instance details.
    - Network: *KafkaBlockchain*
    - Auto-assign Public IP: *Enable*
    - Otherwise accept the defaults for Instance Details.
- Add Storage.
    - Accept the defaults for Add Storage. (30 GiB)
- Tag Spot Request.
    - Skip this.
- Configure Security Group.
    - Add new Security group, name: *KafkaBlockchain*
    - Description: *KafkaBlockchain security group*
    - Otherwise accept the defaults and ignore Warning for Configure Security Group.
- Review Instance Launch.
    - accept the summary and click Launch.
- Launch.
    - Choose to Create a new key pair.
    - Key pair name: *KafkaBlockchain*
    - Download the Key Pair to the X509 directory on the development workstation.
    - Launch the instance, and navigate to the link indicated by the text "The following instance launches have been initiated:".
    - Optionally edit the Name to: *KafkaBlockchain*
- Wait until the Instance State is "running", and Status Checks are 2/2 checks.

- Perform the following steps from a terminal session in the developer's workstation.
- `$ chmod 400 X509/KafkaBlockchain.pem`
- `$ ssh -i "X509/KafkaBlockchain.pem" ubuntu@ec2-34-239-124-118.compute-1.amazonaws.com` (Public DNS (IPv4))
- `$ sudo apt update`
- `$ sudo apt upgrade -y`
    - `Keep the local version currently installed (if prompted)`
- `$ sudo reboot` (afterwords wait a minute or so for the instance to restart with the apt software updates)
- `$ ssh -i "X509/KafkaBlockchain.pem" ubuntu@ec2-34-239-124-118.compute-1.amazonaws.com`
- `Confirm that all the KafkaBlockchain unit tests pass.`
    - `$ cd ~/git/KafkaBlockchain; mvn test`
- `Launch ZooKeeper in a terminal session`
    - `$ cd ~/kafka_2.12-2.5.0; bin/zookeeper-server-start.sh config/zookeeper.properties`
- `Launch Kafka in a second terminal session after ZooKeeper initializes.`
    - `$ ssh -i "X509/KafkaBlockchain.pem" ubuntu@ec2-34-239-124-118.compute-1.amazonaws.com`
    - `$ cd ~/kafka_2.12-2.5.0; bin/kafka-server-start.sh config/server.properties`
- `Launch the following demonstrations in a third terminal session after Kafka initializes.`
    - `ssh -i "X509/KafkaBlockchain.pem" ubuntu@ec2-34-239-124-118.compute-1.amazonaws.com`
- **Kafka blockchain demonstration**
    - Navigate to this project's directory, and launch this script in a third terminal session which runs the KafkaBlockchain demo. The first program produces four payloads on the blockchain and then consumes them. A second program verifies the whole blockchain.
    - `$ cd ~/git/KafkaBlockchain`
    - `$ scripts/run-kafka-blockchain-demo.sh`
    - Remove the previous blockchain messages by running this script.
    - `$ scripts/run-kafka-blockchain-demo-reset.sh`
- **Kafka blockchain multiple partition demonstration**
    - The first program produces four payloads on the blockchain and then consumes them. A second program verifies the whole blockchain.
    - `$ scripts/run-kafka-blockchain-multiple-partition-demo.sh`
    - Remove the previous blockchain messages by running this script.
    - `$ scripts/run-kafka-blockchain-multiple-partition-demo-reset.sh`
- **Kafka encrypted blockchain demonstration**
    - The program produces four encrypted payloads on the blockchain and then consumes them with decryption. A second program verifies the whole blockchain without needing to decrypt it.
    - `$ scripts/run-kafka-blockchain-encryption-demo.sh`

    - Remove the previous blockchain messages by running this script.
    - `$ scripts/run-kafka-blockchain-demo-encryption-reset.sh`
- **Kafka blockchain benchmark with 10 million messages**
    - This script runs the KafkaBlockchain benchmark. The producer creates 10 million messages for the Kafka blockchain with three partitions. A second program reads the 10 million messages and verifies the correctness of the blockchain. The demonstration payload is efficiently serialized using tailored methods that implement Externalizable. The benchmark takes about 80 seconds to create the blockchain messages. And about 130 seconds more to verify the integrity of the 10 million blockchain messages with a consumer thread for each partition, by recomputing the SHA-256 hash for each message.
    - `$ scripts/run-kafka-blockchain-benchmark.sh`
    - Remove the previous 10 million blockchain messages by running this script.
    - `$ scripts/run-kafka-blockchain-benchmark-reset.sh`