ASSIGNMENT

1. Steps for installing and instantiating the Chaincode on HLF2.2

Ans. To install and instantiate chaincode on Hyperledger Fabric 2.2 the following steps are to be followed.

1. Prepare Chaincode
2. Package Chaincode
3. Install Chaincode
4. Approve Chaincode definition
5. Commit Chaincode Definition
6. Instantiate Chaincode
7. Verify Instantiation

a. Prepare Chaincode: Write your chaincode logic in a programming language supported by Hyperledger Fabric, such as Go, Node.js, or Java. Ensure that your chaincode implements the required interface methods defined by the Fabric shim.

b. Package Chaincode: Package your chaincode into a tarball (.tar.gz) archive. You can use the peer lifecycle chaincode package command to do this.

c. Install Chaincode: Install the chaincode package on each peer that will host the chaincode. Use the peer lifecycle chaincode install command, providing the path to the chaincode package.

d. Approve Chaincode Definition: Obtain an endorsement from the necessary organizations endorsing the chaincode definition. Use the peer lifecycle chaincode approveformyorg command to do this.

e. Commit Chaincode Definition: Once the chaincode definition is approved by the necessary organizations, commit the chaincode definition to the channel. Use the peer lifecycle chaincode commit command.

f. Instantiate Chaincode: Instantiate the chaincode on the channel. Use the peer chaincode instantiate command, specifying the channel, chaincode name, version, and other necessary parameters.

g. Verify Instantiation: Verify that the chaincode is successfully instantiated by querying its status on the channel using the peer lifecycle chaincode querycommitted command.

1. Explain Cryptogen and Configtxgen

Ans. Cryptogen and Configtxgen are two essential tools provided by Hyperledger Fabric for setting up and configuring a Fabric network.

1. Cryptogen: Generates the required cryptographic materials,

- Cryptogen is a tool used for generating cryptographic material (certificates and keys) for various network entities like peers, orderers, organizations, and users.

- It simplifies the process of creating the required cryptographic materials for a Fabric network by automating the generation of certificates and keys.

- With Cryptogen, you can define the network's topology in a configuration file (crypto-config.yaml) where you specify the organizations, their peers, and other network entities. The tool then reads this configuration file and generates the necessary cryptographic materials based on the defined topology.

- The generated cryptographic materials are essential for securing communications within the Fabric network, enforcing access control, and ensuring the integrity of transactions.

2. Configtxgen:

- Configtxgen is a tool used for generating the Fabric network configuration transaction artifacts.

- It is primarily used for creating the genesis block for the ordering service and channel configuration transactions.

- The genesis block contains the initial configuration of the ordering service, including information about the consortium, the organizations participating in the network, and their cryptographic material.

- Configtxgen also generates channel configuration transaction artifacts that define the policies, capabilities, and initial settings for a Fabric channel.

- Using a configuration file (configtx.yaml), you define the network's configuration, including the consortium's organizations, orderer profiles, channel configuration, and policies. configtxgen then reads this configuration file and generates the necessary transaction artifacts.

- These configuration transaction artifacts are essential for bootstrapping a new Fabric network, initializing channels, and updating the network configuration over time.

Summary- Cryptogen is used for generating cryptographic materials, while configtxgen is used for generating configuration transaction artifacts for initializing and configuring a Hyperledger Fabric network. Both tools are essential for setting up and managing Fabric networks efficiently.

1. Develop a Chaincode for storing the data into Blockchain
2. Store
3. Retrieve
4. Update
5. Gethistory
6. GetbyNonPrimary (Using CouchDB Rich Queries)

a. Store data:

In this operation, we’ll store data on the blockchain ledger.

The store function will accept a key and a value, and it will write this data to the ledger.

b. Retrieve Data:

The retrieve function will allow us to read data from the ledger based on a given key.

c. Update Data:

To update existing data, we’ll create an update function.

This function will accept a key and a new value, and it will overwrite the existing value associated with that key.

d. Get History:

The getHistory function retrieves the historical changes for a given key.

It allows you to track how the value associated with a key has changed over time

e. Get by Non-Primary Key:

To query data using non-primary keys (such as attributes other than the key itself), we can use CouchDB rich queries.

Define an index in the configtx.yaml file for the desired attribute.

Use the getQueryResult function to retrieve data based on the indexed attribute.

https://github.com/shiva2231/SIMPLIFY-SOFTECH/commit/5e8025ec464e8a2a9f0d80af6a28d8f608183da0