Requirements:

- Hyperledger Fabric (v1.4.4) – Binaries and samples (setup instructions in the next section)
- Go (v1.12 and above)
- Docker with Docker-Compose (Docker Desktop preferred)
- Git and Curl
- Java 1.8 (For Fabric's Java SDK)
- Maven v3.6 +
- Eclipse IDE configured with Apache Tomcat server v8.5 (To run the Webapp)

Setup and Run instructions with explanation:

1) Make sure that you have installed all the prerequisites in the previous section before beginning this setup

2) Firstly, we need to install the binaries of Hyperledger Fabric, so we can access all its components, that is, the blockchain framework itself.
   To do this, open Git Bash/Terminal, navigate to a directory of your choice and run the command:

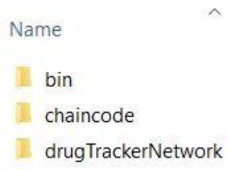
   ```
   curl -sSL http://bit.ly/2ysbOFE | bash -s -- 1.4.4 1.4.4 0.4.18
   ```

   This will install all the Fabric binaries, set up environment variables (Component binaries of Fabric can be seen in the **bin** folder), along with samples

3) For assurance, the platform-specific binaries can be added to your PATH environment variable manually as well using command:

   ```
   export PATH=<location where you ran curl command>/bin:$PATH
   ```

4) Now that Hyperledger Fabric is ready, we can extract our DrugTracking Network folder with chaincode in a directory of your choice. Once you extract the **drugTrackerNetwork** and **chaincode** folders, copy and paste the '**bin**' folder from the 2nd step along with these two directories.
   Your directory structure should be like:



Name

bin

chaincode

drugTrackerNetwork

5) The above folder setup constitutes the blockchain network for this use case. We have used one of the configurations provided by Fabric itself and extend it to our network configuration and use-case. To start up the network, first we need to generate cryptographic files and MSP certificates for all the components in the network. For this we use Fabric's **cryptogen** tool for development purposes (remember in a real scenario, we need to use authorized certs).

   A **drugTrackerNetwork** folder contains a file **drugledgerscript.sh.** This file basically will start up the entire network and deploy it as containers in Docker.

   Navigate inside **drugTrackerNetwork** folder and run the command,

```
$ sh drugledgerscript.sh -m generate
```

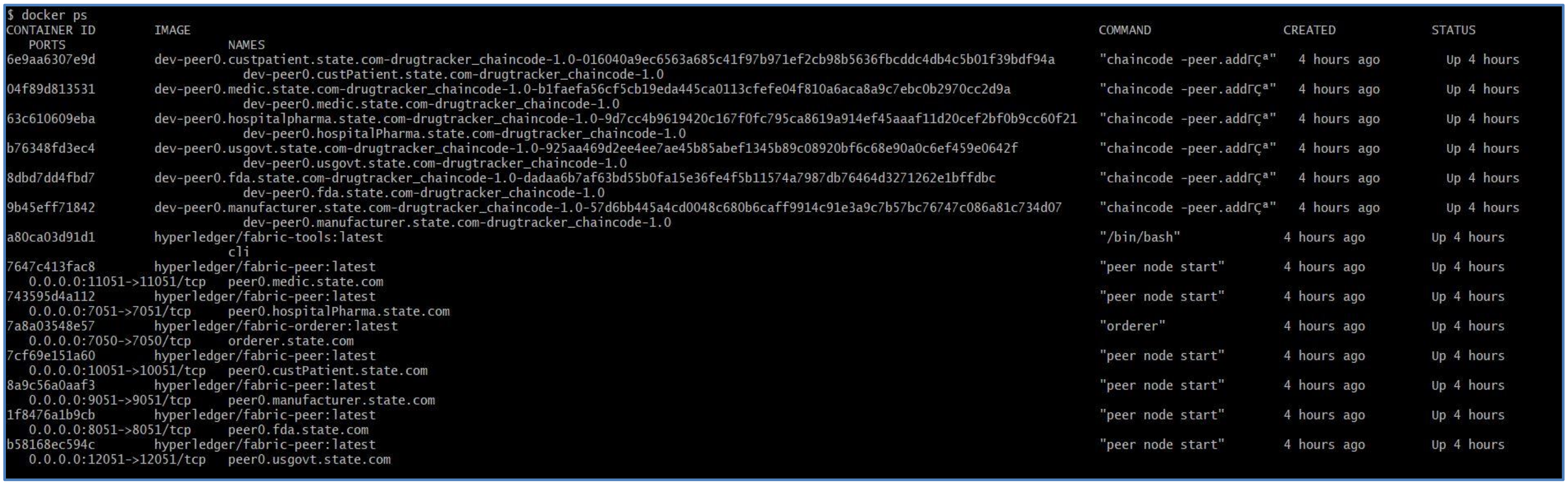
   The above command will generate necessary crypto material and MSPs and put them in **crypto-config** folder

   Then run the command,

```
$ sh drugledgerscript.sh -m up
```

   The above command will start up the entire network (with peers, orderer, chaincode installed and instantiated in the peers, a client container to access them). Once the above command is successful, and END echo comes up. To view the started blockchain network container enter the command **docker ps**

   The screen will be like this:



```
$ docker ps
CONTAINER ID        IMAGE                                                                                                                      COMMAND                  CREATED             STATUS
   PORTS                      NAMES
6e9aa6307e9d        dev-peer0.custpatient.state.com-drugtracker_chaincode-1.0-016040a9ec6563a685c41f97b971ef2cb98b5636fbcddc4db4c5b01f39bdf94a     "chaincode -peer.addΓÇª"   4 hours ago         Up 4 hours
                            dev-peer0.custPatient.state.com-drugtracker_chaincode-1.0
04f89d813531        dev-peer0.medic.state.com-drugtracker_chaincode-1.0-b1faefa56cf5cb19eda445ca0113cfefe04f810a6aca8a9c7ebc0b2970cc2d9a           "chaincode -peer.addΓÇª"   4 hours ago         Up 4 hours
                            dev-peer0.medic.state.com-drugtracker_chaincode-1.0
63c610609eba        dev-peer0.hospitalpharma.state.com-drugtracker_chaincode-1.0-9d7cc4b9619420c167f0fc795ca8619a914ef45aaaf11d20cef2bf0b9cc60f21  "chaincode -peer.addΓÇª"   4 hours ago         Up 4 hours
                            dev-peer0.hospitalPharma.state.com-drugtracker_chaincode-1.0
b76348fd3ec4        dev-peer0.usgovt.state.com-drugtracker_chaincode-1.0-925aa469d2ee4ee7ae45b85abef1345b89c08920bf6c68e90a0c6ef459e0642f          "chaincode -peer.addΓÇª"   4 hours ago         Up 4 hours
                            dev-peer0.usgovt.state.com-drugtracker_chaincode-1.0
8dbd7dd4fbd7        dev-peer0.fda.state.com-drugtracker_chaincode-1.0-dadaa6b7af63bd55b0fa15e36fe4f5b11574a7987db76464d3271262e1bffdbc             "chaincode -peer.addΓÇª"   4 hours ago         Up 4 hours
                            dev-peer0.fda.state.com-drugtracker_chaincode-1.0
9b45eff71842        dev-peer0.manufacturer.state.com-drugtracker_chaincode-1.0-57d6bb445a4cd0048c680b6caff9914c91e3a9c7b57bc76747c086a81c734d07    "chaincode -peer.addΓÇª"   4 hours ago         Up 4 hours
                            dev-peer0.manufacturer.state.com-drugtracker_chaincode-1.0
a80ca03d91d1        hyperledger/fabric-tools:latest                                                                                            "/bin/bash"              4 hours ago         Up 4 hours
                      cli
7647c413fac8        hyperledger/fabric-peer:latest                                                                                             "peer node start"        4 hours ago         Up 4 hours
   0.0.0.0:11051->11051/tcp   peer0.medic.state.com
743595d4a112        hyperledger/fabric-peer:latest                                                                                             "peer node start"        4 hours ago         Up 4 hours
   0.0.0.0:7051->7051/tcp     peer0.hospitalPharma.state.com
7a8a03548e57        hyperledger/fabric-orderer:latest                                                                                          "orderer"                4 hours ago         Up 4 hours
   0.0.0.0:7050->7050/tcp     orderer.state.com
7cf69e151a60        hyperledger/fabric-peer:latest                                                                                             "peer node start"        4 hours ago         Up 4 hours
   0.0.0.0:10051->10051/tcp   peer0.custPatient.state.com
8a9c56a0aaf3        hyperledger/fabric-peer:latest                                                                                             "peer node start"        4 hours ago         Up 4 hours
   0.0.0.0:9051->9051/tcp     peer0.manufacturer.state.com
1f8476a1b9cb        hyperledger/fabric-peer:latest                                                                                             "peer node start"        4 hours ago         Up 4 hours
   0.0.0.0:8051->8051/tcp     peer0.fda.state.com
b58168ec594c        hyperledger/fabric-peer:latest                                                                                             "peer node start"        4 hours ago         Up 4 hours
   0.0.0.0:12051->12051/tcp   peer0.usgovt.state.com
```

6) The blockchain network is setup and ready. You can see the peers, the orderer, cli, and chaincode installed in each peer in the above screenshot. NETWORK IS READY !

7) Before starting the web application (or whenever the network is restarted), copy the **crypto-config** folder generated in the drugTrackerNetwork directory and copy it to USER_HOME directory for your system (This is where the webapp reads the certificates for the peer and orderer).

8) We are now ready to access the network by means of Fabric Java SDK and do operations on the blockchain through the web application. The **Webapp** folder, contains the necessary files for the webapp with **.pom,** that contains all the SpringMVC and Fabric SDK dependencies for the Webapp.

9) Import the application as an existing Maven project in Eclipse IDE. Once the project is imported, build it with the Maven command, **mvn clean install** (either as run configuration) or in command prompt. This will generate a 'war' file in the Target folder, which is deployable in a server (in our case Apache Tomcat running in port 8080). Once you start the server, the application is deployed and ready to be accessed at http://localhost:8080/drugtrack-ledger/

10) The above URL would redirect you to a login page. This forms the interface for login of each peer and to perform operations. **The credentials (username and password are the same) are manufacturer, fda, cust, govt, hosp, medic.** To login, each peer interface MUST provide a certificate file to login (forms the crypto material for authorizing and signing transactions). These certificates can be found at "**USER_HOME\crypto-config\peerOrganizations\manufacturer.state.com\users\Admin@manufacturer.stat**

e.com\msp\signcerts"(this is the cert for the manufacturer, similarly for each peer the cert will be found in their respective folders) The certificate file acts like an ID card for entry.

Now you can login into each peer interface and perform operations on the blockchain as per the use-case flow.

More about the project in the Report/Guide. Some important points to note are listed in the 'Points to Note' section in the Report/Guide. Refer to that for more clarity with respect to setup and implementation.

Demo video link: https://youtu.be/sq-0NH2bb5U