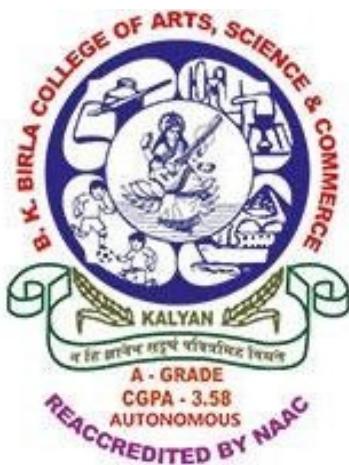


B. K. BIRLA COLLEGE (AUTONOMOUS), KALYAN  
(DEPARTMENT OF COMPUTER SCIENCE)



**CERTIFICATE**

*This is to certify that Mr. Subodh Kamble Roll No 20 Exam Seat No \_\_\_\_\_ has satisfactorily completed the Practical in Advanced Computing (Web3 Technologies) as laid down in the regulation of University of Mumbai for the purpose of M.Sc. Computer Science Semester-III(Practical) Examination 2023-2024.*

*Date: 21/12/2023*

*Place: Kalyan*

---

*Signature of Examiners*

---

*Professor In-charge*

---

*Head  
Dept. Of Computer Science*

**B. K. BIRLA COLLEGE OF ARTS, SCIENCE & COMMERCE  
(AUTONOMOUS), KALYAN**

**DEPARTMENT OF COMPUTER SCIENCE  
Academic Year 2023-24**

**ROLL NO - 20**

**STUDENT NAME – SUBODH KAMBLE**

**CLASS: MSC CS-II**

**SEMESTER: III**

**COURSE CODE: PSCS3011**

**COURSE NAME – PRACTICAL OF  
ADVANCED COMPUTING**

**FACULTY IN-CHARGE – MRS. POOJA  
CHETTIAR**

<b>INDEX</b>			
<b>SR. NO .</b>	<b>DATE</b>	<b>PARTICULAR</b>	<b>SIGNATURE</b>
1	10/10/2023	Install and understand Docker container, Node.js, Java and Hyperledger Fabric, Ethereum and perform necessary software installation on local machine/create instance on Cloud to run.	
2	16/10/2023	Create and deploy a block chain network using Hyperledger Fabric SDK for Java	
3	30/10/2023	Interact with a block chain network. Execute transactions and requests against a block chain network by creating an app to test the network and its rules	
4	01/11/2023	Deploy an asset-transfer app using block chain. Learn app development within a Hyperledger Fabric network	
5	23/11/2023	Practical on MetaMask Ganache	
6	30/11/2023	Practical on voting dApps	
7	06/12/2023	Practical on Lottery dApps	
8	13/12/2023	Practical on Supply Chain dApps	

## Advanced Computing (Web3 Technologies)

**Practical 1 :** Install and understand Docker container, Node.js, Java and Hyperledger Fabric, Ethereum and perform necessary software installation on local machine/create instance on Cloud to run.

### Docker Installation:

#### System Requirements:

- Linux: 64-bit kernel and CPU support for virtualization. KVM virtualization support is recommended.
- macOS: macOS 10.14 or higher.
- Windows: Windows 10 or 11 64-bit. Hyper-V and Windows Subsystem for Linux 2 (WSL 2) must be enabled.

#### Installation steps

1. **Download Docker:** Go to the Docker website and download the appropriate Docker installer for your operating system.
2. **Install Docker:** Follow the on-screen instructions to install Docker.
3. **Verify installation:** Open a terminal window and run the following command:

**Command 1:** *docker version*

```

Client: Docker Engine - Community
Version:          24.0.7
API version:      1.43
Go version:       go1.20.10
Git commit:       afdd53b
Built:            Thu Oct 26 09:07:41 2023
OS/Arch:          linux/amd64
Context:          default

Server: Docker Engine - Community
Engine:
  Version:          24.0.7
  API version:      1.43 (minimum version 1.12)
  Go version:       go1.20.10
  Git commit:       311b9ff
  Built:            Thu Oct 26 09:07:41 2023
  OS/Arch:          linux/amd64
  Experimental:    false
  containerd:
    Version:         1.6.24
    GitCommit:       61f9fd88f79f081d64d6fa3bb1a0dc71ec870523
  runc:
    Version:         1.1.9
    GitCommit:       v1.1.9-0-gccaecfc
  docker-init:
    Version:         0.19.0
    GitCommit:       de40ad0
[1] ~ /D/s/m/sem3
[0] 0:fish*                                     "fish /home/subodh/Doc" 21:36 02-Nov-23

```

**Start the Docker daemon:** The Docker daemon is the service that runs Docker containers.

**Check the status of the Docker daemon:** To check the status of the Docker daemon, run the following command:

**Command 2:** *sudo systemctl status docker*

```
[I] ~/D/s/m/sem3 ___ systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2023-11-02 09:42:51 IST; 12h ago
TriggeredBy: _ docker.socket
    Docs: https://docs.docker.com
  Main PID: 1312 (dockerd)
    Tasks: 15
   Memory: 96.3M
      CPU: 11.224s
     CGroup: /system.slice/docker.service
             └─1312 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 02 09:42:34 subodh-HP-Laptop-15-bs0xx dockerd[1312]: time="2023-11-02T09:42:34.166587458+05:30" level=info msg="Startin>
Nov 02 09:42:34 subodh-HP-Laptop-15-bs0xx dockerd[1312]: time="2023-11-02T09:42:34.181458470+05:30" level=info msg="detect>
Nov 02 09:42:39 subodh-HP-Laptop-15-bs0xx dockerd[1312]: time="2023-11-02T09:42:39.092874607+05:30" level=info msg="[graph]>
Nov 02 09:42:42 subodh-HP-Laptop-15-bs0xx dockerd[1312]: time="2023-11-02T09:42:42.415522979+05:30" level=info msg="Loadin>
Nov 02 09:42:47 subodh-HP-Laptop-15-bs0xx dockerd[1312]: time="2023-11-02T09:42:47.792056826+05:30" level=info msg="Defaul>
Nov 02 09:42:48 subodh-HP-Laptop-15-bs0xx dockerd[1312]: time="2023-11-02T09:42:48.213814850+05:30" level=info msg="Loadin>
Nov 02 09:42:49 subodh-HP-Laptop-15-bs0xx dockerd[1312]: time="2023-11-02T09:42:49.811922894+05:30" level=info msg="Docker>
Nov 02 09:42:49 subodh-HP-Laptop-15-bs0xx dockerd[1312]: time="2023-11-02T09:42:49.861348303+05:30" level=info msg="Daemon>
Nov 02 09:42:51 subodh-HP-Laptop-15-bs0xx systemd[1]: Started Docker Application Container Engine.
Nov 02 09:42:51 subodh-HP-Laptop-15-bs0xx dockerd[1312]: time="2023-11-02T09:42:51.167272041+05:30" level=info msg="API li>
lines 1-22/22 (END)
```

[0] 0:systemctl\* "systemctl status dock" 22:19 02-Nov-23

**Test Docker:** To test Docker, run the following command:

**Command 3:** *docker run hello-world*

```
[I] ~/D/s/m/sem3 ___ docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

**Docker** is a containerization platform that allows developers to build, test, and deploy applications quickly and efficiently. It does this by packaging applications into standardized units called containers, which have everything the application needs to run, including code, libraries, system tools, and runtime.

**Docker is used for a variety of reasons, including:**

1. **Portability:** Docker containers can be run on any machine that has the Docker Engine installed, regardless of the underlying operating system. This makes it easy to deploy applications to different environments, such as development, staging, and production.
2. **Reproducibility:** Docker containers are always in a known state, which makes it easy to reproduce them across different environments. This is important for ensuring that applications behave consistently in different environments.
3. **Scalability:** Docker containers can be easily scaled up or down, depending on the needs of the application. This makes it easy to scale applications to meet demand.
4. **Efficiency:** Docker containers are very efficient in terms of resource usage. They share the underlying operating system kernel, which means that they use less memory and CPU than traditional virtual machines.

**To run a Dockerfile locally, you can follow these steps:**

1. **Build the Docker image:** To build the Docker image, run the following command:

***docker build -t <image-name>***

Where <image-name> is the name of the Docker image you want to create.

2. **Run the Docker image:** To run the Docker image, run the following command:

***docker run -it <image-name>***

Where <image-name> is the name of the Docker image you want to run.

3. **Attach to the Docker container:** To attach to the Docker container, run the following command:

***docker attach <container-id>***

Where <container-id> is the ID of the Docker container you want to attach to.

4. **Interact with the Docker container:** Once you are attached to the Docker container, you can interact with it as if you were inside a regular terminal window.
5. **Exit the Docker container:** To exit the Docker container, press Ctrl+C.

### Example

The following Dockerfile creates a simple web application:

```
FROM nginx:latest
COPY . /usr/share/nginx/html
CMD ["nginx", "-g", "daemon off;"]
```

**Command 4: docker run node**

```
[I] ~ ))> docker run node
Unable to find image 'node:latest' locally
latest: Pulling from library/node
8457fd5474e7: Pull complete
13baa2029dde: Pull complete
325c5bf4c2f2: Pull complete
7e18a660069f: Pull complete
c30e0acec6d5: Pull complete
6cc2ae4489b2: Pull complete
2c7b705ed4b2: Pull complete
36b273626836: Pull complete
Digest: sha256:0052410af98158173b17a26e0e2a46a3932095ac9a0ded660439a8ffae65b1e3
Status: Downloaded newer image for node:latest
[I] ~ ))>
```

**Command 5 (To check available docker images):** *docker image*

[I] ~ )) docker images				
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
node	latest	b612cbc8128d	34 hours ago	1.1GB
hello-world	latest	9c7a54a9a43c	6 months ago	13.3kB
hyperledger/fabric-ca	1.4.1	3a1799cd45d7	4 years ago	252MB
hyperledger/fabric-orderer	1.4.1	ec4ca236d3d4	4 years ago	173MB
hyperledger/fabric-peer	1.4.1	a1e3874f338b	4 years ago	178MB

To build and run this Dockerfile, you would run the following commands:

### **Command 6 : docker build**

**Command 7:** docker run -it -p 8080:80 <hash> /bin/bash

```
[I] ~/D/s/m/sem3 _ time docker build .
[+] Building 46.1s (5/20)                                            docker:default
=> [internal] load build definition from Dockerfile                  1.0s
=> => transferring dockerfile: 1.69kB                                0.0s
=> [internal] load .dockerignore                                    1.4s
=> => transferring context: 2B                                     0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest    9.8s
=> [auth] library/ubuntu:pull token for registry-1.docker.io       0.0s
=> [ 1/16] FROM docker.io/library/ubuntu@sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f 19.7s
=> => resolve docker.io/library/ubuntu@sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f 0.2s
=> => sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f / 1.13kB   1.13kB / 1.13kB  0.0s
=> => sha256:c9cf959fd83770dfdefd8fb42cef0761432af36a764c077aed54bbc5bb25368 4248 / 4248  0.0s
=> => sha256:e4c58958181a5925816faa528ce959e487632f4cf8192f8132f71b32df2744b4 2.30kB / 2.30kB  0.0s
=> => sha256:aec8e493d3972efa43bf4de4e3cdba59cf878f78f59c82fb3e48c87cb22a12e 29.54MB / 29.54MB 16.7s
=> => extracting sha256:aec8e493d3972efa43bf4de4e3cdba59cf878f78f59c82fb3e48c87cb22a12e 1.55s
=> [ 2/16] RUN apt-get update && apt-get install -y software-properties-common curl wget unzip 15.1s
=> => # Get:5 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
=> => # Get:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
=> => # Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1185 kB]
=> => # Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
=> => # Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1389 kB]
=> => # Get:10 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
```

```

--> => sha256:e4c58958181a5925816faa528ce959e487632f4cf192f8132f71b32df2744b4 2.30kB / 2.30kB          0.0s
--> => sha256:aece8493d3972efa43bfd4ee3cdba659c0f787f8f59c82fb3e48c87ccb22a12e 29.54MB / 29.54MB        16.7s
--> => extracting sha256:aece8493d3972efa43bfd4ee3cdba659c0f787f8f59c82fb3e48c87ccb22a12e                   1.5s
--> [ 2/16] RUN apt-get update && apt-get install -y      software-properties-common      curl      wget      unzip    1218.3s
--> [ 3/16] RUN curl -sL https://deb.nodesource.com/setup_20.x | bash -
--> [ 4/16] RUN apt-get install -y nodejs                  36.7s
--> [ 5/16] RUN apt-get install -y openjdk-11-jdk           470.9s
--> [ 6/16] RUN add-apt-repository ppa:ethereum/ethereum      24.1s
--> [ 7/16] RUN apt-get update -y                          10.4s
--> [ 8/16] RUN apt-get install -y solc                     45.0s
--> [ 9/16] RUN wget https://go.dev/dl/go1.21.3.linux-amd64.tar.gz   115.9s
--> [10/16] RUN tar -C /usr/local -xzf go1.21.3.linux-amd64.tar.gz  14.6s
--> [11/16] RUN rm go1.21.3.linux-amd64.tar.gz            2.7s
--> [12/16] RUN apt-get install -y docker.io      libtool      libltdl-dev      libssl-dev      autoconf      autom  787.7s
--> [13/16] RUN service --status-all                   3.0s
--> [14/16] RUN apt-get clean                         # can skip          2.8s
--> [15/16] RUN rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/* # can skip          5.6s
--> [16/16] WORKDIR /app                            6.4s
--> exporting to image                                115.8s
--> => exporting layers                               115.7s
--> => writing image sha256:15d5030c67b183a74ed5a6714a311be9ab995849be20eb2a8f45a93404c9b6df 0.1s

Executed in 48.42 mins   fish      external
  usr time 25.29 secs  0.00 millis  25.29 secs
  sys time 18.24 secs  2.06 millis  18.24 secs

[I] ~/D/s/m/sem3  _ _ _ [I]
[O] 0:fish*                                     "fish /home/subodh/Doc" 23:25 02-Nov-23

```

**Command 8:** `docker run -it 15d5030c67b183 /bin/bash`

```

[I] ~/D/s/m/sem3  _ _ _ docker run -it 15d5030c67b183a74ed5a6714a311be9ab995849be20eb2a8f45a93404c9b6df /bin/bash
root@3a42339b5eb9:/app# ls -lah
total 8.0K
drwxr-xr-x 2 root root 4.0K Nov  2 17:53 .
drwxr-xr-x 1 root root 4.0K Nov  2 18:02 ..
root@3a42339b5eb9:/app# 

```

**Commands to check installed software version in container :**

```

> go -version
> solc -version
> node -v

```

**Practical 2:** Create and deploy a block chain network using Hyperledger Fabric SDK for Java.

Prerequisites:

1. **Java Development Kit (JDK) 11 or later:** Install JDK 11 or a later version on your system. You can download it from <https://www.oracle.com/java/technologies/downloads/>.

```
[I] ~/h/prac2 ___ java --version
openjdk 11.0.20.1 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
[I] ~/h/prac2 ___
[O] 0:fish*
```

2. **Maven 3.8.1 or later:** Install Maven 3.8.1 or a later version on your system. You can download it from <https://maven.apache.org/download.cgi>.

```
[I] ~/h/prac2 ___ mvn -v
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 11.0.20.1, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en_IN, platform encoding: UTF-8
OS name: "linux", version: "6.2.0-36-generic", arch: "amd64", family: "unix"
[I] ~/h/prac2 ___
[O] 0:fish*
```

3. **Docker Desktop 20.10.14 or later (if using Dockized environment):** Install Docker Desktop 20.10.14 or a later version if you want to use the Dockerized environment

```
[I] ~/h/prac2 ___ docker --version
Docker version 24.0.7, build afdd53b
[I] ~/h/prac2 ___ docker-compose --version
docker-compose version 1.29.2, build unknown
[I] ~/h/prac2 ___
[O] 0:fish*
```

### Installing Hyperledger Fabric SDK for Java:

- **Download the Hyperledger Fabric SDK for Java:** Go to the Hyperledger Fabric releases page (<https://github.com/hyperledger/fabric/releases>) and download the latest binary distribution for your operating system.
- **Extract the SDK:** Extract the downloaded binary archive to a location of your choice. For example, you can extract it to a directory named fabric-sdk-java in your home directory.
- **Set environment variables:** Add the following environment variables to your system's environment variables:

```
export FABRIC_HOME=/path/to/fabric-sdk-java
```

```
export JAVA_HOME=/path/to/jdk11.0.17
```

Replace the paths with the actual locations of your fabric-sdk-java and jdk11.0.17 directories.

- Setting up the Fabric Test Network:
- **Clone the Fabric Samples repository:** Open a terminal window and navigate to a directory where you want to store the Fabric Samples repository. Then, clone the repository using the following command:

```
git clone https://github.com/hyperledger/fabric-samples.git
```

- Go to the test-network directory: Navigate into the test-network directory within the cloned repository:

```
cd fabric-samples/test-network
```

```
[I] ~/h/prac2 ... ls
fabric-samples
[I] ~/h/prac2 ... cd fabric-samples/
[I] ~/h/p/fabric-samples ... ls
asset-transfer-abac      bin                  hardware-security-module  test-network-k8s
asset-transfer-basic     builders             high-throughput           test-network-nano-bash
asset-transfer-events    CHANGELOG.md        LICENSE                 token-erc-1155
asset-transfer-ledger-queries ci                MAINTAINERS.md          token-erc-20
asset-transfer-private-data CODE_OF_CONDUCT.md off_chain_data        README.md            token-erc-721
asset-transfer-sbe       CODEOWNERS          SECURITY.md            token-sdk
asset-transfer-secured-agreement config            CONTRIBUTING.md       SECURITY.md            token-utxo
auction-dutch           CONTRIBUTING.md       test-application
auction-simple          full-stack-asset-transfer-guide test-network
[I] ~/h/p/fabric-samples ... [main]

[I] ~/h/p/fabric-samples ... cd test-network/
[I] ~/h/p/f/test-network ... ls
addOrg3                  compose             network.config  prometheus-grafana  setOrgEnv.sh
bft-config               configtx            network.sh      README.md           system-genesis-block
CHAINCODE_AS_A_SERVICE_TUTORIAL.md monitorDocker.sh organizations  scripts
[I] ~/h/p/f/test-network ... [main]
```

Launch the Fabric test network: Use the following command to launch the Fabric test network:

```
./network.sh up
```

```
Using docker and docker-compose
Starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb' with crypto from 'cryptogen'
LOCAL_VERSION=v2.5.4
DOCKER_IMAGE_VERSION=v2.5.4
/home/subodh/hyperledger/prac2/fabric-samples/test-network/../bin/cryptogen
Generating certificates using cryptogen tool
Creating Org1 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml --output=organizations
org1.example.com
+ res=0
Creating Org2 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml --output=organizations
org2.example.com
+ res=0
Creating Orderer Org Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organizations
+ res=0
Generating CCP files for Org1 and Org2
Creating network "fabric_test" with the default driver
Creating orderer.example.com ... done
Creating peer0.org2.example.com ... done
Creating peer0.org1.example.com ... done
Creating cli ...
[0] 0:bash*          "./network.sh up /home" 19:55 24-Nov-23
```

```
Creating Org2 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml --output=organizations
org2.example.com
+ res=0
Creating Orderer Org Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organizations
+ res=0
Generating CCP files for Org1 and Org2
Creating network "fabric_test" with the default driver
Creating orderer.example.com ... done
Creating peer0.org2.example.com ... done
Creating peer0.org1.example.com ... done
Creating cli ...
CONTAINER ID   IMAGE             COMMAND           CREATED          STATUS          PORTS          NAMES
52b1ac2d9406   hyperledger/fabric-tools:latest   "/bin/bash"        10 seconds ago   Up Less than a second   cli
d1266b1804e9   hyperledger/fabric-peer:latest    "peer node start"  20 seconds ago   Up 10 seconds   0.0.0.0:905
1->9051/tcp, ::9051->9051/tcp, 7051/tcp, 0.0.0.0:9445->9445/tcp, ::9445->9445/tcp
0.org2.example.com
b422de800235   hyperledger/fabric-orderer:latest  "orderer"         20 seconds ago   Up 10 seconds   0.0.0.0:705
0->7050/tcp, ::7050->7050/tcp, 0.0.0.0:7053->7053/tcp, ::7053->7053/tcp, 0.0.0.0:9443->9443/tcp, ::9443->9443/tcp
orderer.example.com
73c1a60446f6   hyperledger/fabric-peer:latest    "peer node start"  20 seconds ago   Up 10 seconds   0.0.0.0:705
1->7051/tcp, ::7051->7051/tcp, 0.0.0.0:9444->9444/tcp, ::9444->9444/tcp
0.org1.example.com
[1] -/h/p/f/test-network ...
[0] 0:fish*          "fish /home/subodh/hyp" 19:56 24-Nov-23
```

This command will start a Hyperledger Fabric test network with two peers, an ordering service, and three certificate authorities.

*./network.sh down*

This command will stop Hyperledger Fabric test network with two peers.

### Interacting with the Fabric Test Network:

Install the Fabric CLI tools: Download the appropriate Fabric CLI binaries for your operating system from the Hyperledger Fabric releases page (<https://github.com/hyperledger/fabric/releases>). Extract the downloaded archive and add the following path to your environment variables:

**!Optional:**

*export FABRIC\_BIN=/path/to/fabric-binaries*

Replace the path with the actual location of your extracted Fabric CLI binaries.

**Install the Fabric Java SDK tools:** Install the Fabric Java SDK tools using Maven:

*mvn install*

This command will install the necessary tools for interacting with the Fabric test network from Java applications.

*./network.sh up createChannel*

```
7448a76fb8a5 hyperledger/fabric-peer:latest "peer node start" 43 seconds ago Up 24 seconds 0.0.0.0:9051->9051/tcp
tcp ::::9051->9051/tcp, 7051/tcp, 0.0.0.0:9445->9445/tcp, :::9445->9445/tcp
peer0.org2.e
xample.com
Using docker and docker-compose
Generating channel genesis block 'mychannel.block'
Using organization 1
/home/subodh/hyperledger/prac2/fabric-samples/test-network/../bin/configtxgen
+ '[' 0 -eq 1 ']'
+ configtxgen -profile ChannelUsingRaft -outputBlock ./channel-artifacts/mychannel.block -channelID mychannel
2023-11-24 20:14:13.536 IST 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-11-24 20:14:13.544 IST 0002 INFO [common.tools.configtxgen.localconfig] completeInitialization -> orderer type: etcdr
af
2023-11-24 20:14:13.544 IST 0003 INFO [common.tools.configtxgen.localconfig] completeInitialization -> Orderer.EtcdRaft.Opt
ions unset, setting to tick_interval:"500ms" election_tick:10 heartbeat_tick:1 max_inflight_blocks:5 snapshot_interval_size
:16777216
2023-11-24 20:14:13.544 IST 0004 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/subodh/hyp
erledger/prac2/fabric-samples/test-network/configtx/configtx.yaml
2023-11-24 20:14:13.548 IST 0005 INFO [common.tools.configtxgen] doOutputBlock -> Generating genesis block
2023-11-24 20:14:13.548 IST 0006 INFO [common.tools.configtxgen] doOutputBlock -> Creating application channel genesis bloc
k
2023-11-24 20:14:13.548 IST 0007 INFO [common.tools.configtxgen] doOutputBlock -> Writing genesis block
+ res=0
Creating channel mychannel
Adding orderers
+ . scripts/orderer.sh mychannel
+ '[' 0 -eq 1 ']'
+ res=0
[8] 0:bash*                                         "./network.sh up creat" 20:14 24-Nov-23
```

```
Creating channel mychannel
Adding orderers
+ . scripts/orderer.sh mychannel
+ '[' 0 -eq 1 ']'
+ res=0
Status: 201
{
    "name": "mychannel",
    "url": "/participation/v1/channels/mychannel",
    "consensusRelation": "consenter",
    "status": "active",
    "height": 1
}

Channel 'mychannel' created
Joining org1 peer to the channel...
Using organization 1
+ peer channel join -b ./channel-artifacts/mychannel.block
+ res=1
+ peer channel join -b ./channel-artifacts/mychannel.block
+ res=1
+ peer channel join -b ./channel-artifacts/mychannel.block
+ res=0
2023-11-24 20:14:33.689 IST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-11-24 20:14:36.789 IST 0002 INFO [channelCmd] executeJoin -> Successfully submitted proposal to join channel
Joining org2 peer to the channel...
Using organization 2
[8] 0:bash*                                         "./network.sh up creat" 20:14 24-Nov-23
```

## Package the smart contract Java:

```
Cd /fabric-samples/asset-transfer-basic/chaincode-java
```

```
[I] ~/h/p/f/test-network ___ cd ..
[I] ~/h/p/fabric-samples ___ ls
asset-transfer-abac           bin                   hardware-security-module test-network-k8s
asset-transfer-basic          builders               high-throughput        test-network-nano-bash
asset-transfer-events         CHANGELOG.md        LICENSE                token-erc-1155
asset-transfer-ledger-queries ci                    MAINTAINERS.md      token-erc-28
asset-transfer-private-data   CODE_OF_CONDUCT.md off_chain_data    token-erc-721
asset-transfer-sbe            CODEOWNERS          README.md             token-sdk
asset-transfer-secured-agreement config               SECURITY.md        token-utxo
auction-dutch                 CONTRIBUTING.md     test-application
auction-simple                full-stack-asset-transfer-guide test-network
[I] ~/h/p/fabric-samples ___ cd asset-transfer-basic/
[I] ~/h/p/f/asset-transfer-basic ___ ls
application-gateway-go       application-java     chaincode-external   chaincode-typescript
application-gateway-java     application-javascript chaincode-go        README.md
application-gateway-typescript application-typescript chaincode-java      rest-api-go
application-go                application-typescript-hsm chaincode-javascript rest-api-typescript
[I] ~/h/p/f/asset-transfer-basic ___ cd chaincode-java/
[I] ~/h/p/f/a/chaincode-java ___ ls
build.gradle config docker Dockerfile gradle gradlew gradlew.bat README.md settings.gradle src
[I] ~/h/p/f/a/chaincode-java ___ mvn install
```

*docker build .*

```
[I] ./h/p/f/a/chaincode-java -- less Dockerfile
[I] ./h/p/f/a/chaincode-java -- docker build .
[+] Building 8.1s (4/6)
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.02kB
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/gradle:7-jdk11-alpine
=> [internal] load metadata for docker.io/library/openjdk:11-jre
=> [auth] library/gradle:pull token for registry-1.docker.io
=> [auth] library/openjdk:pull token for registry-1.docker.io

[0] 0:docker*                               "docker build . /home/" 20:27 24-Nov-23

[+] Building 663.5s (16/19)
=> => sha256:001cb2626ad5763b25b439ee0052f6692e5c0f2d5d982a00a8819ace5e521452 55.00MB / 55.00MB
=> => sha256:d940b986e964657da49910b495173d6c4f0d98b473c5f82fd32723d165 5.16MB / 5.16MB
=> => sha256:20a8746827ec1b0043571e4788693eab7e9b0a95301176512791f8c317e281a 10.88MB / 10.88MB
=> => sha256:8510da692cdaa60e4746c14dd90905695eade5888e2ad640786a2be9dc42a0224 5.66MB / 5.66MB
=> => extracting sha256:001cb2626ad5763b25b439ee0052f6692e5c0f2d5d982a00a8819ace5e521452
=> => sha256:b6d8a4395b3d48ae99a6c83b1da7297dfc0678767272973ada75ed772252e392 211B / 211B
=> => sha256:b03feaa6c3ad7464587adff11c70121a7bb47883183ea13fb6b9f73727cfa2a6f7 45.77MB / 45.77MB
=> => extracting sha256:001cb2626ad5763b25b439ee0052f6692e5c0f2d5d982a00a8819ace5e521452
=> => sha256:20a8746827ec1b0043571e4788693eab7e9b0a95301176512791f8c317e281a 0.55MB / 0.55MB
=> => extracting sha256:20a8746827ec1b0043571e4788693eab7e9b0a95301176512791f8c317e281a 0.55MB / 0.55MB
=> => extracting sha256:8510da692cdaa60e4746c14dd90905695eade5888e2ad640786a2be9dc42a0224
=> => sha256:b6d8a4395b3d48ae99a6c83b1da7297dfc0678767272973ada75ed7772572e392
=> => extracting sha256:b03feaa6c3ad7464587adff11c70121a7bb47883183ea13fb6b9f73727cfa2a6f7
=> https://github.com/krallin/tini/releases/download/v0.19.0/tini
=> [auth] library/openjdk:pull token for registry-1.docker.io
=> [stage-1 2/6] ADD https://github.com/krallin/tini/releases/download/v0.19.0/tini /tini
=> [stage-1 3/6] RUN chmod +x /tini
=> [gradle_build 2/4] COPY src/ src/
=> [gradle_build 3/4] COPY build.gradle .
=> [stage-1 4/6] RUN addgroup --system javauser && useradd -g javauser javauser
=> [gradle_build 4/4] RUN gradle --no-daemon build shadowJar -x checkstyleMain -x checkstyleTest
=> => # > Task :shadowJar
=> => # > Task :startShadowScripts
=> => # > Task :shadowWDisttar
=> => # > Task :shadowWDistZip
=> => # > Task :assemble
=> => # > Task :compileTestJava

[0] 0:docker*                               "docker build . /home/" 20:38 24-Nov-23
```

```
./network.sh up createChannel -c mychannel -ca
```

## Deploy the smart contract Java

```
./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-java/ -ccl java
```

```
bft-config           configtx   network.config  prometheus-grafana  setOrgEnv.sh
CHAINCODE_AS_A_SERVICE_TUTORIAL.md  log.txt    network.sh    README.md      system-genesis-block
[1] -h/p/f/test network __ sudo ./network.sh deployCC -ccp ../asset-transfer-events/chaincode-java/ -ccl java
Using docker and docker-compose
deploying chaincode on channel 'mychannel'
executing with the following
- CHANNEL_NAME: mychannel
- CC_NAME: basic
- CC_SRC_PATH: ../asset-transfer-events/chaincode-java/
- CC_SRC_LANGUAGE: java
- CC_VERSION: 1.0.1
- CC_SEQUENCE: auto
- CC_END_POLICY: NA
- CC_COLL_CONFIG: NA
- CC_INIT_FCN: NA
- DELAY: 3
- MAX_RETRY: 5
- VERBOSE: false
executing with the following
- CC_NAME: basic
- CC_SRC_PATH: ../asset-transfer-events/chaincode-java/
- CC_SRC_LANGUAGE: java
- CC_VERSION: 1.0.1
Compiling Java code...
/home/subodh/hyperledger/prac2/fabric-samples/asset-transfer-events/chaincode-java /home/subodh/hyperledger/prac2/fabric-samples/test-network
Downloading https://services.gradle.org/distributions/gradle-6.1.1-bin.zip
.....10%.....20%.....30%.....40%.....50%.....60%.....70%.....80%.....90%...
[1] 0:sudo*                                         "sudo ./network.sh dep" 00:04 25-Nov-23
```

**Practical 3 :** Interact with a block chain network. Execute transactions and requests against a blockchain network by creating an app to test the network and its rules.

## Setup, Running and Testing a Fabric Application:

### 1. Setup the Blockchain Network (generate network):

```
$ git clone https://github.com/IBM/blockchain-application-using-fabric-java-sdk
```

```
[I] ~/h/p/fabric-samples ___ cd blockchain-application-using-fabric-java-sdk/
[I] ~/h/p/f/blockchain-application-using-fabric-java-sdk ___ ll
total 56K
-rw-rw-r-- 1 subodh subodh 569 Nov 26 15:53 CONTRIBUTING.md
-rw-rw-r-- 1 subodh subodh 1.6K Nov 26 15:53 DEBUGGING.md
drwxrwxr-x 2 subodh subodh 4.0K Nov 26 15:53 images
drwxrwxr-x 3 subodh subodh 4.0K Nov 26 15:53 java
-rw-rw-r-- 1 subodh subodh 12K Nov 26 15:53 LICENSE
-rw-rw-r-- 1 subodh subodh 3.1K Nov 26 15:53 MAINTAINERS.md
drwxrwxr-x 2 subodh subodh 4.0K Nov 26 15:53 network
drwxrwxr-x 5 subodh subodh 4.0K Nov 26 15:53 network_resources
-rw-rw-r-- 1 subodh subodh 16K Nov 26 15:53 README.md
[I] ~/h/p/f/blockchain-application-using-fabric-java-sdk ___
```

```
$ cd /network/
```

```
[I] ~/h/p/f/blockchain-application-using-fabric-java-sdk ___ cd network/
[I] ~/h/p/f/b/network ___ ll
total 36K
-rw-rw-r-- 1 subodh subodh 975 Nov 26 15:53 build.sh
-rw-rw-r-- 1 subodh subodh 5.0K Nov 26 15:53 configtx.yaml
-rw-rw-r-- 1 subodh subodh 4.7K Nov 26 15:53 crypto-config.yaml
-rw-rw-r-- 1 subodh subodh 7.0K Nov 26 15:53 docker-compose.yaml
-rw-rw-r-- 1 subodh subodh 175 Nov 26 15:53 stop.sh
-rw-rw-r-- 1 subodh subodh 419 Nov 26 15:53 teardown.sh
[I] ~/h/p/f/b/network ___
```

```
$ chmod +x build.sh stop.sh teardown.sh
```

```
-rwxrwxr-x 1 subodh subodh 975 Nov 26 15:53 build.sh
-rw-rw-r-- 1 subodh subodh 5.0K Nov 26 15:53 configtx.yaml
-rw-rw-r-- 1 subodh subodh 4.7K Nov 26 15:53 crypto-config.yaml
-rw-rw-r-- 1 subodh subodh 7.0K Nov 26 15:53 docker-compose.yaml
-rwxrwxr-x 1 subodh subodh 175 Nov 26 15:53 stop.sh
-rwxrwxr-x 1 subodh subodh 419 Nov 26 15:53 teardown.sh
[I] ~/h/p/f/b/network ___ ./build.sh generate
master - Stopping the previous network (if any)
Removing network network_custom
WARNING: Network network_custom not found.

Setting up the Hyperledger Fabric 1.1 network
Creating network "network_custom" with the default driver
Pulling ca.org1.example.com (hyperledger/fabric-ca:1.4.1)...
1.4.1: Pulling from hyperledger/fabric-ca
34667c7e4631: Downloading [=====] 10.2MB/43.56MB
d18d76a881a4: Download complete
119c7358fbfc: Download complete
2aaaf15f7eff0: Download complete
3f89de4cf84b: Download complete
24194f819972: Downloading [=====] 3.384MB/11.88MB
a3d7dc54e295: Download complete
9a3c9a0ecc41: Waiting
dc5d0553c3a1: Waiting
e1f5909edf17: Waiting
8d735007731d: Waiting
[0] 0:bash* ./build.sh generate /* 16:02 26-Nov-23
```

`$ ./build.sh generate`

```
c06474c386d6: Pull complete
f215c85a7c15: Pull complete
Digest: sha256:09f3ica4dabe1eb2af870ea062561ca686fc59a296ecc3b4bd7e32102c48934
Status: Downloaded newer image for hyperledger/fabric-orderer:1.4.1
Pulling peer0.org1.example.com (hyperledger/fabric-peer:1.4.1)...
1.4.1: Pulling from hyperledger/fabric-peer
34667c7e4631: Already exists
d18d7ea881a4: Already exists
119c7358fbfc: Already exists
2aaef13f3eff0: Already exists
3f89de4cf84b: Already exists
24194f819972: Already exists
979cc59b6d95: Already exists
8eb49214b84e: Pull complete
0e108783b4cb: Pull complete
Digest: sha256:05315d05b2892d34b4ed48f6502d28fe15a71090c36a39c97022a44475a984ad
Status: Downloaded newer image for hyperledger/fabric-peer:1.4.1
Creating orderer.example.com ... done
Creating ca_peerOrg1 ... done
Creating ca_peerOrg2 ... done
Creating peer0.org2.example.com ... done
Creating peer1.org1.example.com ... done
Creating peer1.org2.example.com ... done
Creating peer0.org1.example.com ... done

Network setup completed!!

[1] ~h/p/f/b/network [master]
[0] 0:fish*
```

`$ ./network.sh up`

```
Stopping peer0.org1.example.com ... done
Stopping peer1.org2.example.com ... done
Stopping peer1.org1.example.com ... done
Stopping peer0.org2.example.com ... done
Stopping ca_peerOrg2 ... done
Stopping orderer.example.com ... done
Stopping ca_peerOrg1 ... done
Removing peer0.org1.example.com ... done
Removing peer1.org2.example.com ... done
Removing peer1.org1.example.com ... done
Removing peer0.org2.example.com ... done
Removing ca_peerOrg2 ... done
Removing orderer.example.com ... done
Removing ca_peerOrg1 ... done
Removing network network_custom

Setting up the Hyperledger Fabric 1.1 network
Creating network "network_custom" with the default driver
Creating ca_peerOrg1 ...
Creating ca_peerOrg2 ...
Creating orderer.example.com ...

[0] 0:bash*
```

```
Stopping peer1.org2.example.com ... done
Stopping peer1.org1.example.com ... done
Stopping peer0.org2.example.com ... done
Stopping ca_peerOrg2 ... done
Stopping orderer.example.com ... done
Stopping ca_peerOrg1 ... done
Removing peer0.org1.example.com ... done
Removing peer1.org2.example.com ... done
Removing peer1.org1.example.com ... done
Removing peer0.org2.example.com ... done
Removing ca_peerOrg2 ... done
Removing orderer.example.com ... done
Removing ca_peerOrg1 ... done
Removing network network_custom

Setting up the Hyperledger Fabric 1.1 network
Creating network "network_custom" with the default driver
Creating ca_peerOrg1 ... done
Creating ca_peerOrg2 ... done
Creating orderer.example.com ... done
Creating peer1.org2.example.com ... done
Creating peer1.org1.example.com ... done
Creating peer0.org2.example.com ... done
Creating peer0.org1.example.com ... done

Network setup completed!!

[1] ~h/p/f/b/network [master]
[0] 0:fish*
```

```
$ ./network.sh up -l java

Stopping the previous network (if any)
Stopping peer0.org2.example.com ... done
Stopping peer1.org2.example.com ... done
Stopping peer0.org1.example.com ... done
Stopping peer1.org1.example.com ... done
Stopping ca_peerOrg2 ... done
Stopping ca_peerOrg1 ... done
Stopping orderer.example.com ... done
Removing peer0.org2.example.com ... done
Removing peer1.org2.example.com ... done
Removing peer0.org1.example.com ... done
Removing peer1.org1.example.com ... done
Removing ca_peerOrg2 ... done
Removing ca_peerOrg1 ... done
Removing orderer.example.com ... done
Removing network network_custom

Setting up the Hyperledger Fabric 1.1 network
Creating network "network_custom" with the default driver
Creating ca_peerOrg2 ...
Creating ca_peerOrg1 ...
Creating orderer.example.com ...

[0] 0:bash*
```

```
Setting up the Hyperledger Fabric 1.1 network
Creating network "network_custom" with the default driver
Creating ca_peerOrg2 ... done
Creating ca_peerOrg1 ... done
Creating orderer.example.com ... done
Creating peer0.org1.example.com ... done
Creating peer0.org2.example.com ... done
Creating peer1.org1.example.com ... done
Creating peer1.org2.example.com ... done

Network setup completed!!

[I] ~/h/p/f/b/network _ _ _ [ ]
[0] 0:fish*
```

\$ docker-compose -f docker-compose.yaml up

OR

\$ docker-compose up # this command will build everything needed

Keep this running open another terminal

## **2. compiling and setting java application using maven:**

\$ cd ..//java/

\$ mvn install

```

authCacheMaxSize: 1000, authCachePurgeRatio: 0.750000
peer1.org2.example.com | 2023-11-26 10:45:32.415 UTC [nodeCmd] registerDiscoveryService -> INFO 01d Discovery service activated
peer1.org2.example.com | 2023-11-26 10:45:32.415 UTC [nodeCmd] serve -> INFO 01e Starting peer with ID=[name:"peer1.org2.example.com"], network ID=[dev], address=[peer1.org2.example.com:7051]
peer1.org2.example.com | 2023-11-26 10:45:32.416 UTC [nodeCmd] serve -> INFO 01f Started peer with ID=[name:"peer1.org2.example.com"], network ID=[dev], address=[peer1.org2.example.com:7051]

Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.jar (12 kB at 252 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.jar (230 kB at 800 kB/s)
[INFO] Installing /home/subodh/hyperledger/prac2/fabric-samples/blockchain-application-using-fabric-java-sdk/java/target/blockchain-java-sdk-0.0.1-SNAPSHOT.jar to /home/subodh/.m2/repository/blockchain-java-sdk/blockchain-java-sdk/0.0.1-SNAPSHOT/blockchain-java-sdk-0.0.1-SNAPSHOT.jar
[INFO] Installing /home/subodh/hyperledger/prac2/fabric-samples/blockchain-application-using-fabric-java-sdk/java/pom.xml to /home/subodh/.m2/repository/blockchain-java-sdk/blockchain-java-sdk/0.0.1-SNAPSHOT/blockchain-java-sdk-0.0.1-SNAPSHOT.pom
[INFO] Installing /home/subodh/hyperledger/prac2/fabric-samples/blockchain-application-using-fabric-java-sdk/java/target/blockchain-java-sdk-0.0.1-SNAPSHOT-jar-with-dependencies.jar to /home/subodh/.m2/repository/blockchain-java-sdk/blockchain-java-sdk/0.0.1-SNAPSHOT/blockchain-java-sdk-0.0.1-SNAPSHOT-jar-with-dependencies.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 54.707 s
[INFO] Finished at: 2023-11-26T16:25:57+05:30
[INFO] -----
[1] ~h/p/f/b/java -- master
[0] 0:fish* "fish /home/subodh/hyp" 16:27 26-Nov-23

```

\$ cd target/

```

[1] ~h/p/f/b/java -- ll
total 12K
-rw-rw-r-- 1 subodh subodh 1.6K Nov 26 15:53 pom.xml
drwxrwxr-x 3 subodh subodh 4.0K Nov 26 15:53 src
drwxrwxr-x 7 subodh subodh 4.0K Nov 26 16:25 target
[1] ~h/p/f/b/java -- cd target/
[1] ~h/p/f/b/j/target -- ls
archive-tmp classes maven-status
blockchain-java-sdk-0.0.1-SNAPSHOT.jar generated-sources maven-archiver
blockchain-java-sdk-0.0.1-SNAPSHOT-jar-with-dependencies.jar maven-archiver
[1] ~h/p/f/b/j/target -- master
[0] 0:fish* "fish /home/subodh/hyp" 16:29 26-Nov-23

```

\$ mv blockchain-java-sdk-0.0.1-SNAPSHOT-jar-with-dependencies.jar blockchain-client.jar

```

[1] ~h/p/f/b/j/target -- mv blockchain-java-sdk-0.0.1-SNAPSHOT-jar-with-dependencies.jar blockchain-client.jar master

```

\$ cp blockchain-client.jar ../../network\_resources

### 3. Create and Initialize the channel:

\$ cd ../../network\_resources

\$ java -cp blockchain-client.jar org.example.network.CreateChannel

```

peer1.org2.example.com | 2023-11-26 11:12:32.324 UTC [gossip.service] func1 -> INFO 03a Elected as a leader, starting delivery service for channel mychannel
ca_peerOrg2 | 2023/11/26 11:14:01 [DEBUG] Cleaning up expired nonces for CA 'ca-org2'
ca_peerOrg1 | 2023/11/26 11:14:07 [DEBUG] Cleaning up expired nonces for CA 'ca-org1'

log4j:WARN No appenders could be found for logger (org.hyperledger.fabric.sdk.helper.Config).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Nov 26, 2023 4:42:16 PM org.example.util.Util deleteDirectory
INFO: Deleting - users
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.protobuf.UnsafeUtil (file:/home/subodh/hyperledger/prac2/fabric-samples/blockchain-application-using-fabric-java-sdk/network_resources/blockchain-client.jar) to field java.nio.Buffer.address
WARNING: Please consider reporting this to the maintainers of com.google.protobuf.UnsafeUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Nov 26, 2023 4:42:26 PM org.example.network.CreateChannel main
INFO: Channel created mychannel
Nov 26, 2023 4:42:26 PM org.example.network.CreateChannel main
INFO: peer0.org2.example.com at grpc://localhost:8051
Nov 26, 2023 4:42:26 PM org.example.network.CreateChannel main
INFO: peer1.org1.example.com at grpc://localhost:7056
Nov 26, 2023 4:42:26 PM org.example.network.CreateChannel main
INFO: peer1.org2.example.com at grpc://localhost:8056
Nov 26, 2023 4:42:26 PM org.example.network.CreateChannel main
INFO: peer0.org1.example.com at grpc://localhost:7051
[1] ~h/p/f/b/network_resources -- master
[0] 0:fish* "fish /home/subodh/hyp" 16:45 26-Nov-23

```

#### 4. Register and enroll users:

```
$ java -cp blockchain-client.jar org.example.user.RegisterEnrollUser
```

```
[rl ~/h/p/f/b/network_resources ... java -cp blockchain-client.jar org.example.user.RegisterEnrollUser           master ...
Nov 26, 2023 4:57:41 PM org.example.util.Util deleteDirectory
INFO: Deleting - users
log4j:WARN No appenders could be found for logger (org.hyperledger.fabric.sdk.helper.Config).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Nov 26, 2023 4:57:43 PM org.example.client.CAClient enrollAdminUser
INFO: CA -http://localhost:7054 Enrolled Admin.
Nov 26, 2023 4:57:43 PM org.example.client.CAClient registerUser
INFO: CA -http://localhost:7054 Registered User - user1700998063072
Nov 26, 2023 4:57:43 PM org.example.client.CAClient enrollUser
INFO: CA -http://localhost:7054 Enrolled User - user1700998063072
[I] ~/h/p/f/b/network_resources ...
[O] 0:fish*
```

#### 5. Stop The network:

```
[I] ~/h/p/f/b/network ... ./stop.sh

# Shut down the Docker containers that might be currently running.
docker-compose -f docker-compose.yml stop
Stopping peer1.org2.example.com ... done
Stopping peer0.org2.example.com ... done
Stopping ca_peerOrg1 ... done
Stopping orderer.example.com ... done
Stopping ca_peerOrg2 ... done
[I] ~/h/p/f/b/network ...
```

**Practical 4:** Deploy an asset-transfer app using block chain. Learn app development within a Hyperledger Fabric network.

## Prerequisites:

1. Docker installed on your system
  2. Hyperledger Fabric downloaded and installed
  3. Familiarity with go lang <1.6/nodejs

```
$ cd fabric-samples/test-network
```

```
[I] ~/h/prac2 ___ cd fabric-samples/test-network/
[I] ~/h/p/f/test-network ___ ll
total 88K
drwxrwxr-x 4 subodh subodh 4.0K Nov 24 18:34 addOrg3
drwxrwxr-x 2 subodh subodh 4.0K Nov 24 18:34 bft-config
-rw-rw-r-- 1 subodh subodh 14K Nov 24 18:34 CHAINCODE_AS_A_SERVICE_TUTORIAL.md
drwxrwxr-x 4 subodh subodh 4.0K Nov 24 18:34 compose
drwxrwxr-x 2 subodh subodh 4.0K Nov 24 18:34 configtx
-rwxrwxr-x 1 subodh subodh 774 Nov 24 18:34 monitorDocker.sh
-rw-rw-r-- 1 subodh subodh 1.5K Nov 24 18:34 network.config
-rwxrwxr-x 1 subodh subodh 23K Nov 24 18:34 network.sh
drwxrwxr-x 5 subodh subodh 4.0K Nov 25 23:06 organizations
drwxrwxr-x 5 subodh subodh 4.0K Nov 24 18:34 prometheus-grafana
-rw-rw-r-- 1 subodh subodh 3.5K Nov 24 18:34 README.md
drwxrwxr-x 3 subodh subodh 4.0K Nov 24 18:34 scripts
-rwxrwxr-x 1 subodh subodh 2.3K Nov 24 18:34 setOrgEnv.sh
drwxrwxr-x 2 subodh subodh 4.0K Nov 24 18:34 system-genesis-block
[I] ~/h/p/f/test-network ___
```

```
$ ./network.sh down
```

```
$ ./network.sh up createChannel -c mychannel
```

```
$ ./network.sh deployCC -ccn secured -ccp ../asset-transfer-secured-agreement/chaincode-go/ -ccl go -cce "OR('Org1MSP.peer','Org2MSP.peer')"
```

```
- CC_SRC_LANGUAGE: go
- CC_VERSION: 1.0.1
Vendorizing Go dependencies at ../asset-transfer-secured-agreement/chaincode-go/
~/hyperledger/prac2/fabric-samples/asset-transfer-secured-agreement/chaincode-go ~/hyperledger/prac2/fabric-samples/test-network
go: downloading github.com/hyperledger/fabric-contract-api-go v1.2.1
go: downloading github.com/hyperledger/fabric-chaincode-go v0.0.0-20230228194215-b84622ba6a7a
go: downloading google.golang.org/protobuf v1.28.1
go: downloading github.com/hyperledger/fabric-protos-go v0.3.0
go: downloading google.golang.org/grpc v1.53.0
go: downloading github.com/xeipuuv/gojsonschema v1.2.0
go: downloading github.com/go-openapi/spec v0.20.8
go: downloading github.com/gobuffalo/packr v1.30.1
go: downloading google.golang.org/genproto v0.0.0-20230110181048-76db0878b65f
go: downloading golang.org/x/net v0.7.0
go: downloading github.com/xeipuuv/gojsonreference v0.0.0-20180127040603-bd5ef7bd5415
go: downloading github.com/go-openapi/jsonpointer v0.19.5
go: downloading github.com/go-openapi/jsonreference v0.20.0
go: downloading github.com/go-openapi/swag v0.21.1
go: downloading golang.org/x/sys v0.5.8
go: downloading github.com/gobuffalo/envy v1.10.1
go: downloading github.com/gobuffalo/packd v1.0.1
go: downloading github.com/xeipuuv/gojsonpointer v0.0.0-20190905194746-02993c407fb
go: downloading github.com/mailru/easyjson v0.7.7
go: downloading github.com/joho/godotenv v1.4.0
go: downloading github.com/rogpeppe/go-internal v1.8.1
go: downloading github.com/josharian/intern v1.0.0
[0] 0:bash*          "../network.sh deployCC" 09:57 27-Nov-23
```

```
Chaincode definition committed on channel 'mychannel'
Using organization 1
Querying chaincode definition on peer0.org1 on channel 'mychannel'...
Attempting to Query committed status on peer0.org1, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID mychannel --name secured
+ res=0
Committed chaincode definition for chaincode 'secured' on channel 'mychannel':
Version: 1.0.1, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true]
Query chaincode definition successful on peer0.org1 on channel 'mychannel'
Using organization 2
Querying chaincode definition on peer0.org2 on channel 'mychannel'...
Attempting to Query committed status on peer0.org2, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID mychannel --name secured
+ res=0
Committed chaincode definition for chaincode 'secured' on channel 'mychannel':
Version: 1.0.1, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true]
Query chaincode definition successful on peer0.org2 on channel 'mychannel'
Chaincode initialization is not required
[1] ~h/p/f/test-network ... main -
```

- Set the environment variables to operate as Org1 (posix compl shell)

```
$ export PATH=${PWD}/../bin:${PWD}:$PATH
$ export FABRIC_CFG_PATH=$PWD/..config/
$ export CORE_PEER_TLS_ENABLED=true
$ export CORE_PEER_LOCALMSPID="Org1MSP"
$ export
CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
$ export
CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
```

```
$ export CORE_PEER_ADDRESS=localhost:7051
```

- Set the environment variables to operate as Org2

```
$ export PATH=${PWD}/..bin:${PWD}:$PATH
```

```
$ export FABRIC_CFG_PATH=${PWD}/config/
```

```
$ export CORE_PEER_TLS_ENABLED=true
```

```
$ export CORE_PEER_LOCALMSPID="Org2MSP"
```

```
$ export
```

CORE\_PEER\_MSPCONFIGPATH=\${PWD}/organizations/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp

```
$ export
```

CORE\_PEER\_TLS\_ROOTCERT\_FILE=\${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt

```
$ export CORE_PEER_ADDRESS=localhost:9051
```

## Create an asset

Operate from the Org1 terminal (top Terminal):

```
$ export ASSET_PROPERTIES=$(echo -n
"\"object_type\":\"asset_properties\",\"color\":\"blue\",\"size\":35,\"salt\":"
a94a8fe5ccb19ba61c4c0873d391e987982fbbd3\""} | base64 | tr -d '\n')
```

```
subodh@subodh-HP-Laptop-15-bs0xx:/hyperledger/prac2/fabric-samples/test-network$ peer chaincode invoke -o localhost:7050 -ordererTLSHostnameOverride orderer.example.com --tls --cafile ${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlscacert.pem -C mychannel -n secured -c '{"function":"CreateAsset", "Args":["A new asset for Org1MSP"]}' --transient "{\"asset_properties\":\"$ASSET_PROPERTIES\"}"
2023-11-27 10:28:12.501 IST 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status: 200 payload:"d9923f21b770adbc79cbcc47a3aeecc81dc7f030bd129155301ce3932be7fbcc"
```

```
$ export
```

ASSET\_ID=d9923f21b770adbc79cbcc47a3aeecc81dc7f030bd129155301ce3932be7fbcc

## Success message:

{"object\_type":"asset\_properties","color":"blue","size":35,"salt":"a94a8fe5ccb19ba61c4c0873d391e987982fbbd3"}

```
subodh@subodh-HP-Laptop-15-bs0xx:~/hyperledger/prac2/fabric-samples/test-network$ peer chaincode query -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n secured -c "{\"function\":\"GetAssetPrivateProperties\",\"Args\":[\"$ASSET_ID\"]}" {"object_type":"asset_properties","color":"blue","size":35,"salt":"a94a8fe5ccb19ba61c4c0873d391e987982fbdbd3"}
```

*\$ peer chaincode query -o localhost:7050 --ordererTLSHostnameOverride  
orderer.example.com --tls --cafile  
"\${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.e  
xample.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n  
secured -c "{\"function\":\"ReadAsset\",\"Args\":[\"\$ASSET\_ID\"]}"*

#### Success message:

```
{"objectType":"asset","assetID":"d9923f21b770adbc79cbcc47a3aecc81dc7f0  
30bd129155301ce3932be7fbcc","ownerOrg":"Org1MSP","publicDescription":"  
A new asset for Org1MSP"}
```

```
subodh@subodh-HP-Laptop-15-bs0xx:~/hyperledger/prac2/fabric-samples/test-network$ peer chaincode query -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n secured -c "{\"function\":\"ReadAsset\",\"Args\":[\"$ASSET_ID\"]}" {"objectType":"asset","assetID":"d9923f21b770adbc79cbcc47a3aecc81dc7f030bd129155301ce3932be7fbcc","ownerOrg":"Org1MSP","pu  
blicDescription":"A new asset for Org1MSP"}
```

*\$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride  
orderer.example.com --tls --cafile  
"\${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.e  
xample.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n  
secured -c  
"{\"function\":\"ChangePublicDescription\",\"Args\":[\"\$ASSET\_ID\",\"This  
asset is for sale\"]}"*

#### Success message:

```
2023-11-27 10:38:09.198 IST 0001 INFO [chaincodeCmd]  
chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
```

```
subodh@subodh-HP-Laptop-15-bs0xx:~/hyperledger/prac2/fabric-samples/test-network$ peer chaincode invoke -o localhost:7050 -  
-ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n secured -c "{\"function\":\"ChangePublic  
Description\",\"Args\":[\"$ASSET_ID\",\"This asset is for sale\"]}"  
2023-11-27 10:38:09.198 IST 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:  
200
```

*\$ peer chaincode query -o localhost:7050 --ordererTLSHostnameOverride  
orderer.example.com --tls --cafile  
"\${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.e  
xample.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n  
secured -c "{\"function\":\"ReadAsset\",\"Args\":[\"\$ASSET\_ID\"]}"*

**Success message:**

```
{"objectType":"asset","assetID":"d9923f21b770adbc79cbcc47a3aeecc81dc7f030bd129155301ce3932be7fbcc","ownerOrg":"Org1MSP","publicDescription":"This asset is for sale"}
```

**Operate from the Org2 terminal (bottom Terminal):**

*\$ export*

```
ASSET_ID=d9923f21b770adbc79cbcc47a3aeecc81dc7f030bd129155301ce3932be7fbcc
```

```
$ peer chaincode query -o localhost:7050 --ordererTLSHostnameOverride
orderer.example.com --tls --cafile
"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.e
xample.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n
secured -c "{\"function\":\"ReadAsset\",\"Args\":[\"$ASSET_ID\"]}"
```

**Success message:**

```
{"objectType":"asset","assetID":"d9923f21b770adbc79cbcc47a3aeecc81dc7f030bd129155301ce3932be7fbcc","ownerOrg":"Org1MSP","publicDescription":"This asset is for sale"}
```

```
$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride
orderer.example.com --tls --cafile
"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.e
xample.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n
secured -c
"{"function":"ChangePublicDescription","Args":[$ASSET_ID,"the
worst asset"]}"
```

**Success message:**

```
2023-11-27 10:46:36.176 IST 0001 INFO [chaincodeCmd]
chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
```

```
subodh@subodh-HP-Laptop-15-bs0xx:~/hyperledger/prac2/fabric-samples/test-network$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n secured -c "{\"function\":\"ChangePublicDescription\"},\"Args\":[\"$ASSET_ID\", \"\\\"the worst asset\\\"\"]}"
2023-11-27 10:46:36.176 IST 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status: 200
```

## Agree to sell the asset

```
$ export ASSET_PRICE=$(echo -n
"{"asset_id": "$ASSET_ID", "trade_id": "109f4b3c50d7b0df729d299bc6f8e9ef9066971f", "price": 110}" | base64 | tr -d '\n')
```

```
$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile
"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n secured -c "{\"function\":\"AgreeToSell\"},\"Args\":[\"$ASSET_ID\"]}" --transient
{"asset_price": "$ASSET_PRICE"}
```

## Success message:

2023-11-27 10:51:32.734 IST 0001 INFO [chaincodeCmd]  
chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200

```
subodh@subodh-HP-Laptop-15-bs0xx:~/hyperledger/prac2/fabric-samples/test-network$ export ASSET_PRICE=$(echo -n "{\"asset_id\": \"$ASSET_ID\", \"trade_id\": \"109f4b3c50d7b0df729d299bc6f8e9ef9066971f\", \"price\": 110}" | base64 | tr -d '\n')
peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n secured -c "{\"function\":\"AgreeToSell\"},\"Args\":[\"$ASSET_ID\"]}" --transient {"asset_price": "$ASSET_PRICE"}
2023-11-27 10:51:32.734 IST 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status: 200
```

```
$ peer chaincode query -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile
"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n secured -c "{\"function\":\"GetAssetSalesPrice\"},\"Args\":[\"$ASSET_ID\"]}"
```

## Success message:

```
{"asset_id": "d9923f21b770adbc79cbcc47a3aeccc81dc7f030bd129155301ce3932be7fbcc", "trade_id": "109f4b3c50d7b0df729d299bc6f8e9ef9066971f", "price": 110}
```

```
subodh@subodh-HP-Laptop-15-bs0xx:~/hyperledger/prac2/fabric-samples/test-network$ peer chaincode query -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n secured -c "{\"function\":\"GetAssetSalesPrice\"},\"Args\":[\"$ASSET_ID\"]}"
{"asset_id": "d9923f21b770adbc79cbcc47a3aeccc81dc7f030bd129155301ce3932be7fbcc", "trade_id": "109f4b3c50d7b0df729d299bc6f8e9ef9066971f", "price": 110}
```

Update the asset description as Org2

```
$ peer chaincode query -o localhost:7050 --ordererTLSHostnameOverride
orderer.example.com --tls --cafile
"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.e
xample.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n
secured -c
"{"function":"GetAssetPrivateProperties","Args":[$ASSET_ID]}"
```

*Success message:*

```
{"object_type":"asset_properties","color":"blue","size":35,"salt":"a94a8fe5ccb
19ba61c4c0873d391e987982fbcd3"}
```

```
subodh@subodh-HP-Laptop-15-bs0xx:~/hyperledger/prac2/fabric-samples/test-network$ peer chaincode query -o localhost:7050 --
-ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.e
xample.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n secured -c {"function":"GetAssetPrivateProperties","Args":[$ASSET_ID]}
```

```
$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride
orderer.example.com --tls --cafile
"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.e
xample.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n
secured -c
"{"function":"ChangePublicDescription","Args":[$ASSET_ID,"This
asset is not for sale"]}"
```

*Success message:*

```
2023-11-27 11:18:30.055 IST 0001 INFO [chaincodeCmd]
chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
```

```
subodh@subodh-HP-Laptop-15-bs0xx:~/hyperledger/prac2/fabric-samples/test-network$ peer chaincode invoke -o localhost:7050 -
-ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.e
xample.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n secured -c {"function":"ChangePublicDescription","Args":[$ASSET_ID,"This asset is not for sale"]}"
```

```
$ peer chaincode query -o localhost:7050 --ordererTLSHostnameOverride
orderer.example.com --tls --cafile
"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.e
xample.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n
secured -c {"function":"ReadAsset","Args":[$ASSET_ID]}
```

*Success message:*

```
{"objectType":"asset","assetID":"d9923f21b770adbc79cbcc47a3aeecc81dc7f030bd129155301ce3932be7fbcc","ownerOrg":"Org1MSP","publicDescription":"This asset is not for sale"}
```

```
subodh@subodh-HP-Laptop-15-bs0xx:~/hyperledger/prac2/fabric-samples/test-network$ peer chaincode query -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n secured -c "{\"function\":\"ReadAsset\",\"Args\":[\"$ASSET_ID\"]}"
{"objectType":"asset","assetID":"d9923f21b770adbc79cbcc47a3aeecc81dc7f030bd129155301ce3932be7fbcc","ownerOrg":"Org1MSP","publicDescription":"This asset is not for sale"}
```

## Clean up (stopping chaincode):

```
$ ./network.sh down
```

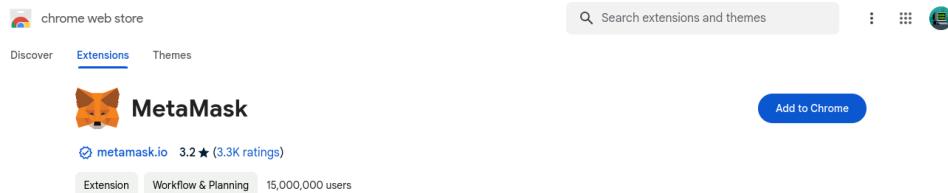
```
Removing remaining containers
4bd1008ea531
797feba02da6
Renoving generated chaincode docker images
Untagged: dev-peer0.org2.example.com-secured 1.0.1-843dc47073bc9ebd5af86d7f422fb794fc450d737c1fa4a1cb2b9cb2c4983d6b-cdfa6a2036afcd05be33a3f61997349ace01a5365ec66f47b99ff9559202eb64b:latest
Deleted: sha256:a330d1940df339af36a1b0eb2a95abf2ad5bade214618e3df91753ebf57257cc
Deleted: sha256:252d23302de4b0a25e050573b6bb2ae6980fa98a1a1332bed760db12b8e916
Deleted: sha256:d0510e0bdd4d0d963a2c42d43941f86a0fd613bada9660aaaf93ebef9cabbd0aeac
Deleted: sha256:66bb9e10b5e19fad38863722e03764c30c53c7387e161351bbc85b32182d1be4
Untagged: dev-peer0.org1.example.com-secured_1.0.1-843dc47073bc9ebd5af86d7f422fb794fc450d737c1fa4a1cb2b9cb2c4983d6b-76cd426ef69ed63b4d63ba3d23c81abef422c4d8acb6ef151b0411017491690:latest
Deleted: sha256:31149b1ee56bce8dd40f638f989e48970b42a53d6874b29281d17c0932b62259
Deleted: sha256:2484f45def4bff0f6e950dbff49564213fcdd2ddd7af3085c1356d0365f261b3
Deleted: sha256:5ed5647a58b576106fbe93c72062d3181254bd78cef448fd1480a98b6b2ab35
Deleted: sha256:73bb8b199e6bfff93b4f7a0b5fc799108ce6046fe5412391897f6f3e9adb24b
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
5f4d90098f5b: Pull complete
Digest: sha256:3fbcc632167424a6d997e74f52b878d7cc478225cffac6bc977eedfe51c7f4e79
Status: Downloaded newer image for busybox:latest
```

Resource/credit: [official Docs](#)

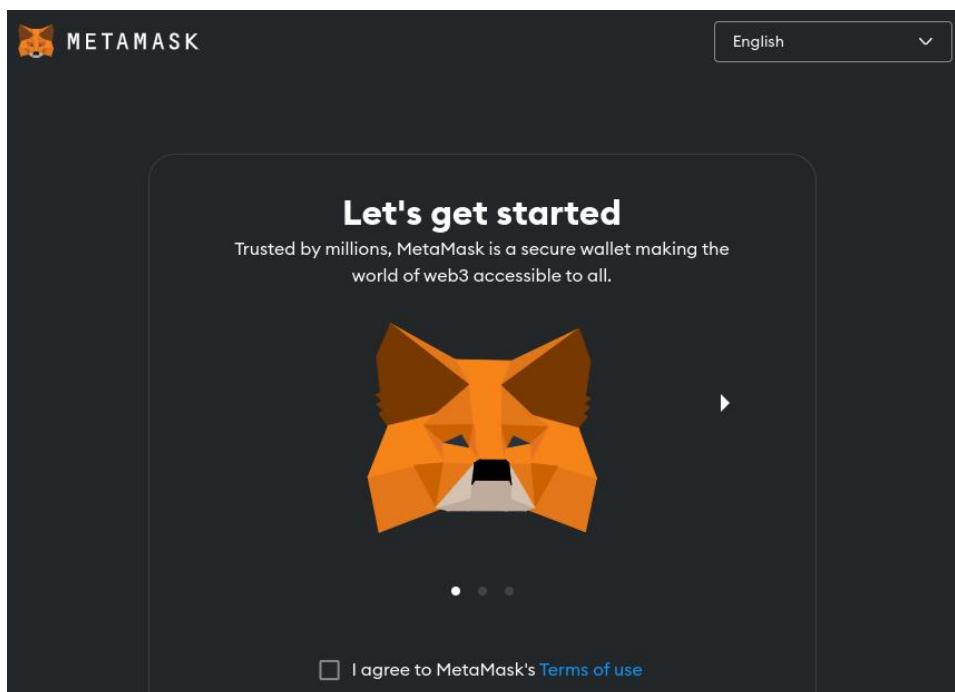
**Practical:** Set up MetaMask, set up Ganache , create two accounts with 100 ETH each, and transfer ETH between the accounts.

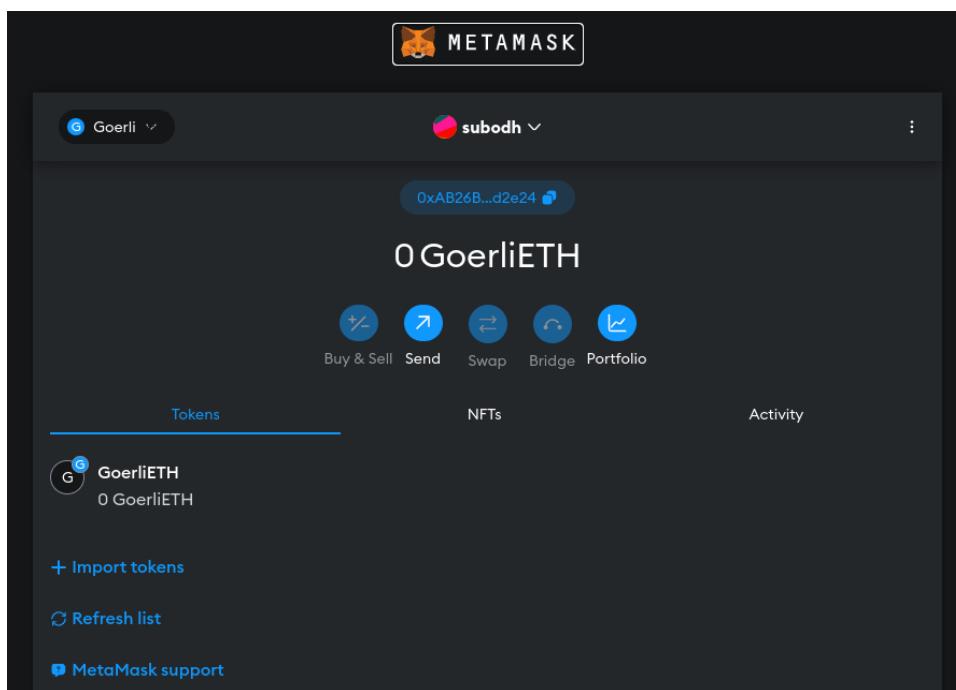
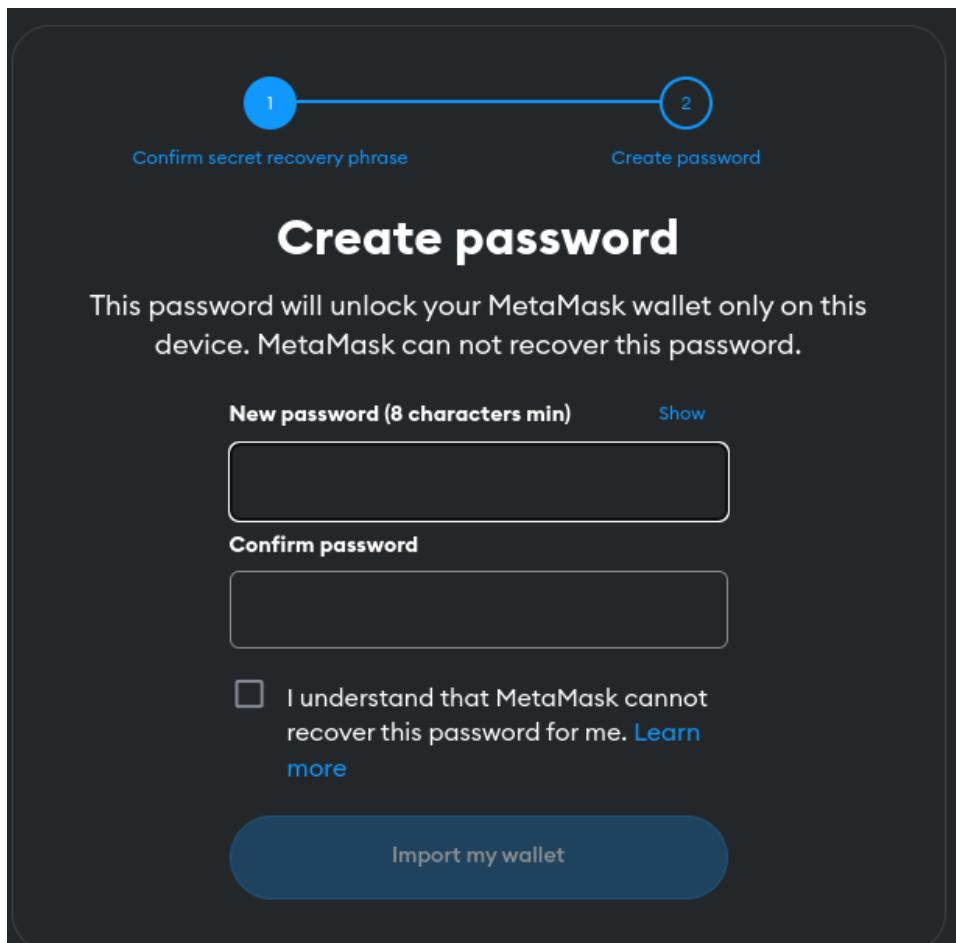
### Install MetaMask Extension:

1. Go to the MetaMask website or the appropriate extension store for your browser (e.g., Chrome Web Store, Firefox Add-ons).



2. Add MetaMask to your browser and follow the on-screen instructions to create an account and set up a password. Save the seed phrase generated during the setup process in a secure location.





## Setting up and Ganache :

MNEMONIC	HD PATH			
harsh climb blind mobile coil wild million print head spread join pulp	m/44'/60'/0'/0/account_index			
ADDRESS	BALANCE	TX COUNT	INDEX	
0x92468A946C1359caF63dB4AD2Be1f3fA9A7e9FD	100.00 ETH	0	0	🔗
0x5AD78460f53d73872950638ACa1928cd3A3f6c6f	100.00 ETH	0	1	🔗
0xe927173E6d85BEC3DEAb4Ed91F32236f92017BeD	100.00 ETH	0	2	🔗
0xE21B8ffa2443D9D75D07A25652B1B9C89bf546E7	100.00 ETH	0	3	🔗
0xd809739033b42a5aa347B419eF06BE1233fcfc91	100.00 ETH	0	4	🔗
0x565Abee896B41008e335552c9D2F378a52E277B7	100.00 ETH	0	5	🔗

### Accessing Ganache Accounts:

After Ganache starts, it will display a list of accounts along with their private keys, addresses, and initial balances.

### Import Accounts into MetaMask:

1. Open MetaMask in your browser and click on "Import Account."
2. Copy the private key of the first account from Ganache and import it into MetaMask by pasting the key.
3. Repeat the process for the second account.

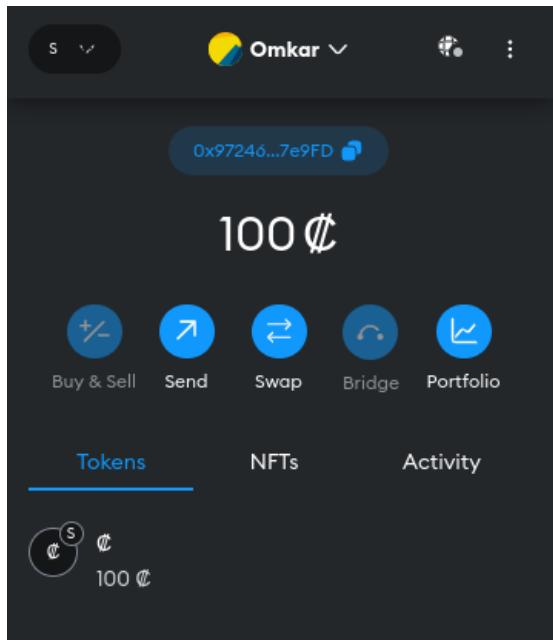
### Transferring ETH Between Accounts:

#### Access Ganache Accounts:

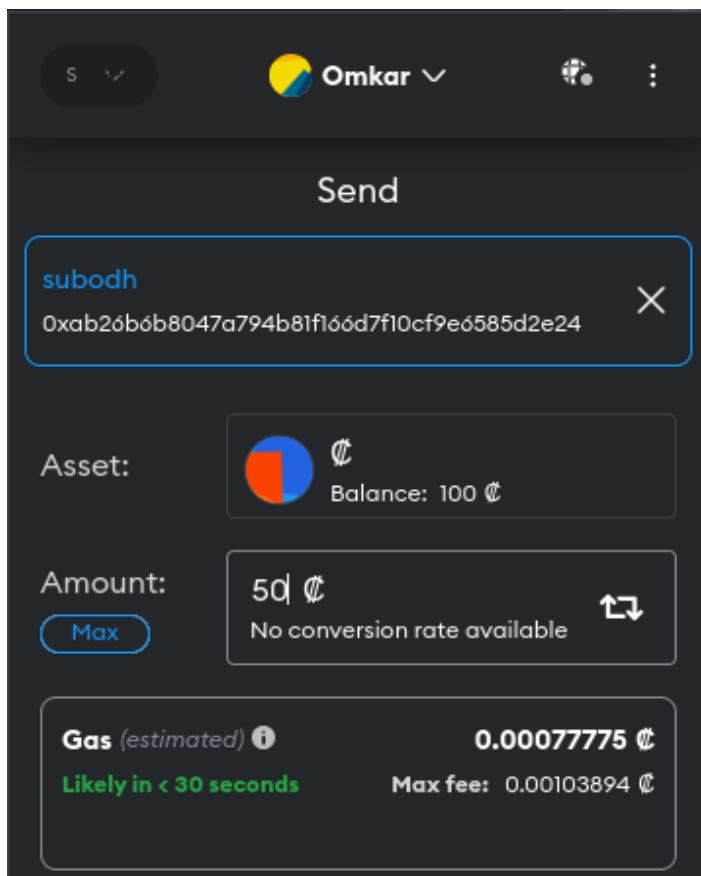
Note the addresses of the two accounts you imported into MetaMask from Ganache.

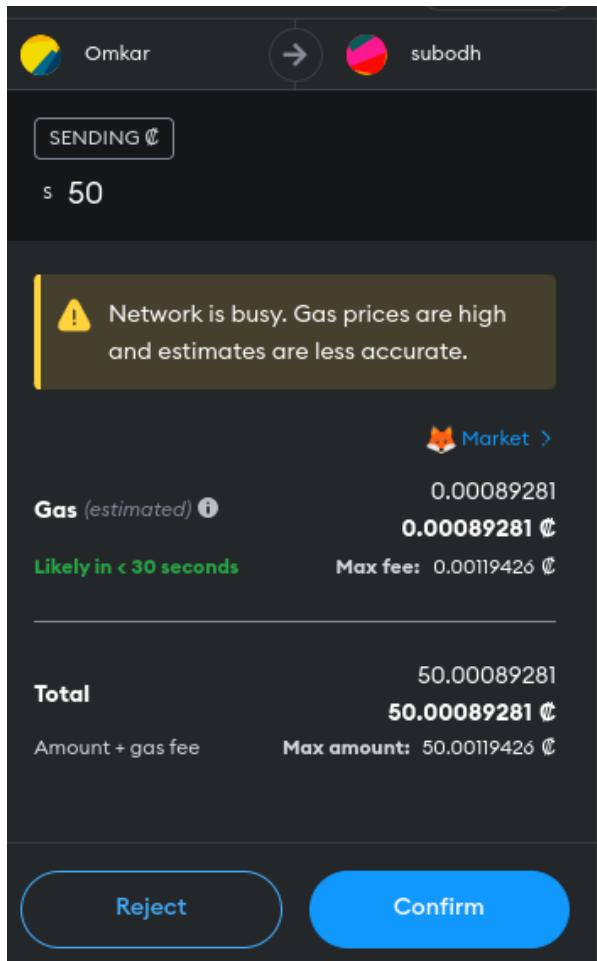
#### Transfer ETH from Account 1 to Account 2:

1. In MetaMask, select Account 1.

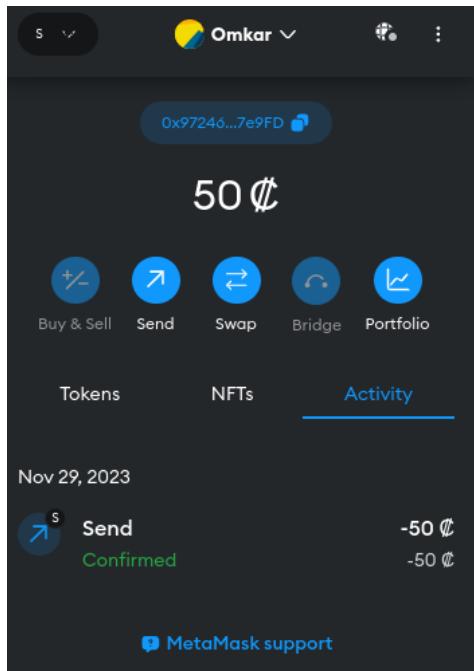


2. Click on "Send" or "Transfer" and enter the address of Account 2 in the recipient field.





3. Enter the amount of ETH you want to send and confirm the transaction.
4. Confirm the transaction in MetaMask by paying the gas fee.
5. Wait for the transaction to be confirmed on the local Ganache blockchain.



### Verify Balance Transfer:

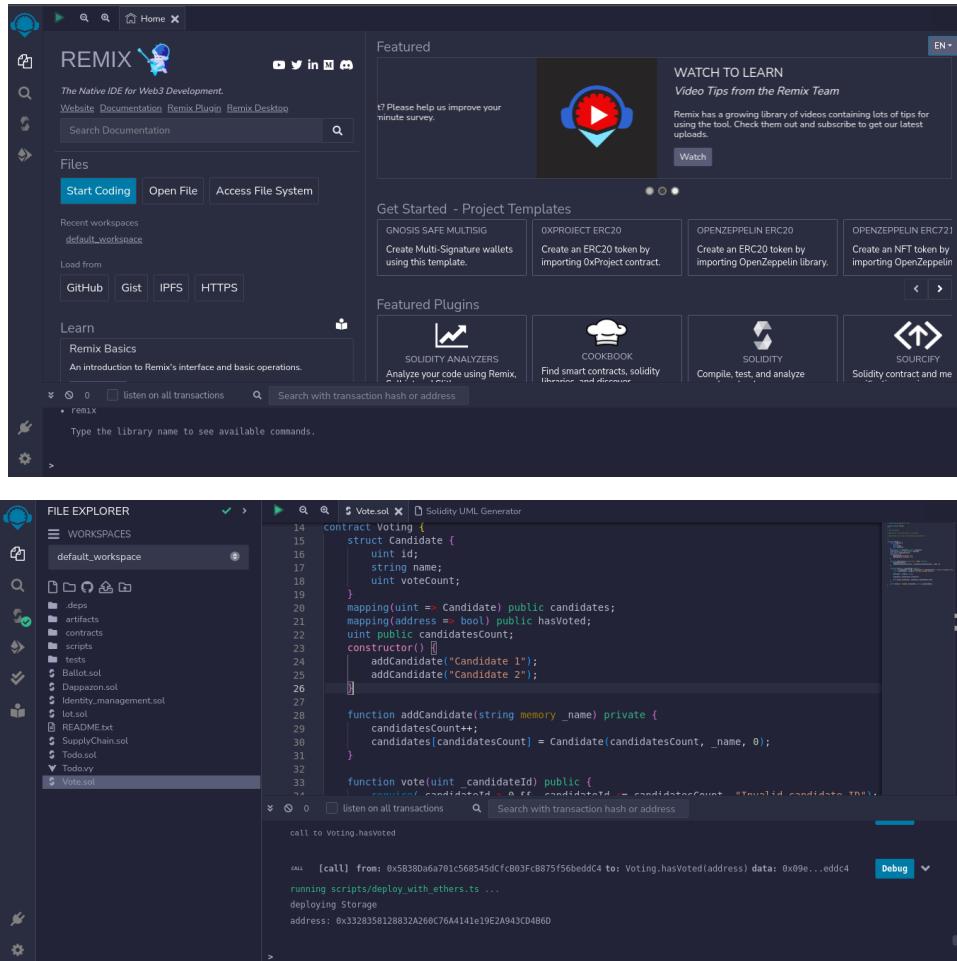
Switch to Account 2 in MetaMask and check the updated balance to confirm the receipt of ETH.

The screenshot shows the MetaMask wallet interface for account 'subodh'. The top bar shows the account name 'subodh' with a balance of '50¢'. Below the account info are five circular buttons: 'Buy & Sell', 'Send', 'Swap', 'Bridge', and 'Portfolio'. A navigation bar at the bottom includes tabs for 'ACCOUNTS', 'BLOCKS', 'TRANSACTIONS', 'CONTRACTS', 'EVENTS', and 'LOGS'. The 'BLOCKS' tab is currently selected. Below the navigation bar, a table provides details of the first block ('BLOCK 1') mined by this account. The table includes columns for 'GAS USED' (21600), 'GAS PRICE' (20000000000), 'GAS LIMIT' (6721975), 'MINER' (MERGE), 'NETWORK ID' (5777), 'RPC SERVER' (HTTP://127.0.0.1:7545), 'MINING STATUS' (AUTOMINING), and 'BLOCK HASH' (0xa4125b101bb6c006a52c82035f621f37e3c4d0684b4b431d9e4d3611f9561650). The 'TRANSACTIONS' tab is also visible at the bottom.

**Practical:** setup and deploy Voting Smart contract in solidity programming language using Remix Ide.

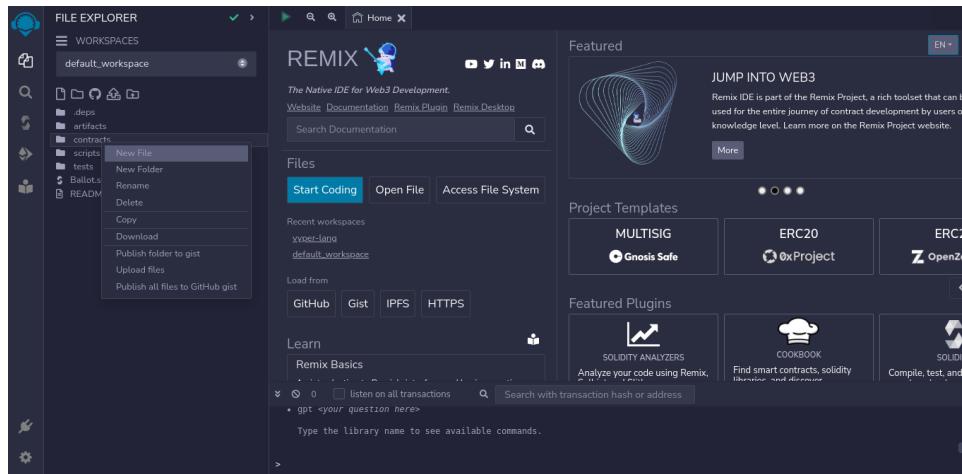
**Access Remix IDE (<https://remix.ethereum.org/>) :**

Open your web browser and go to Remix IDE. This is an online Solidity IDE provided by Ethereum.



### Create a New File:

Click on the "file" icon on the left sidebar to create a new file in contract folder. Name the file Voting.sol or any other relevant name for your voting contract.



## Write Your Smart Contract:

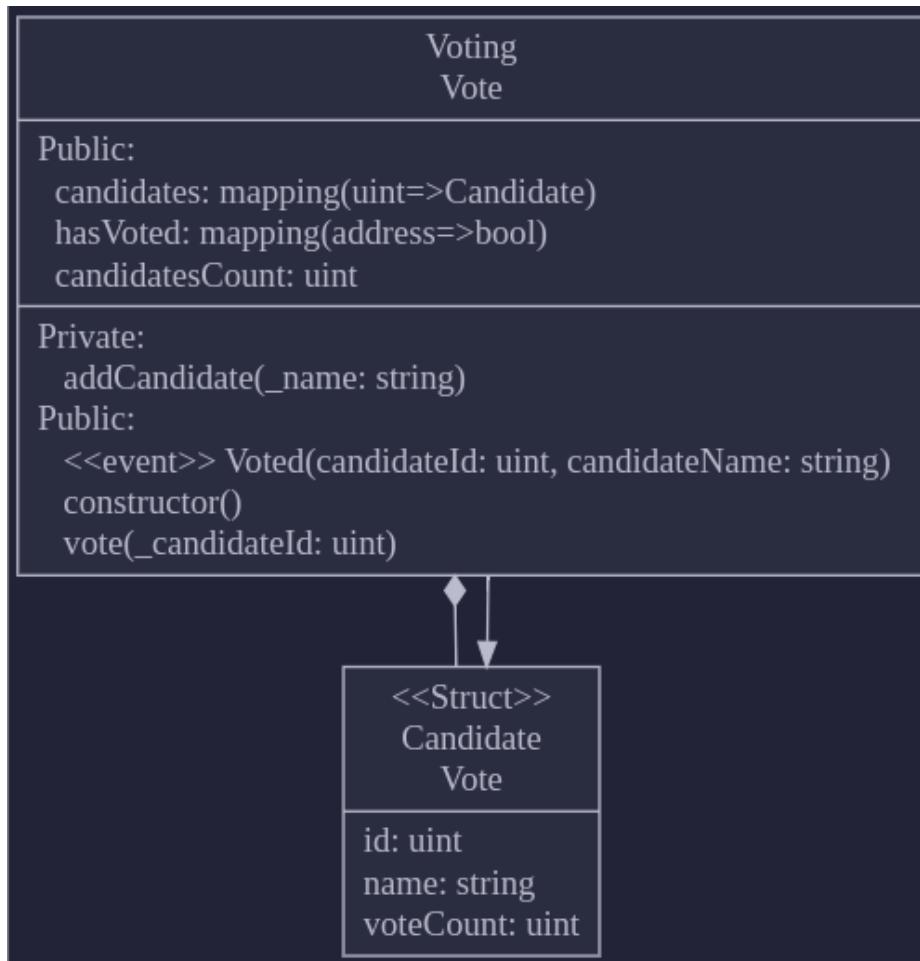
In the newly created file, write your Solidity code for the voting smart contract. example:

```
// SPDX-License-Identifier: MIT

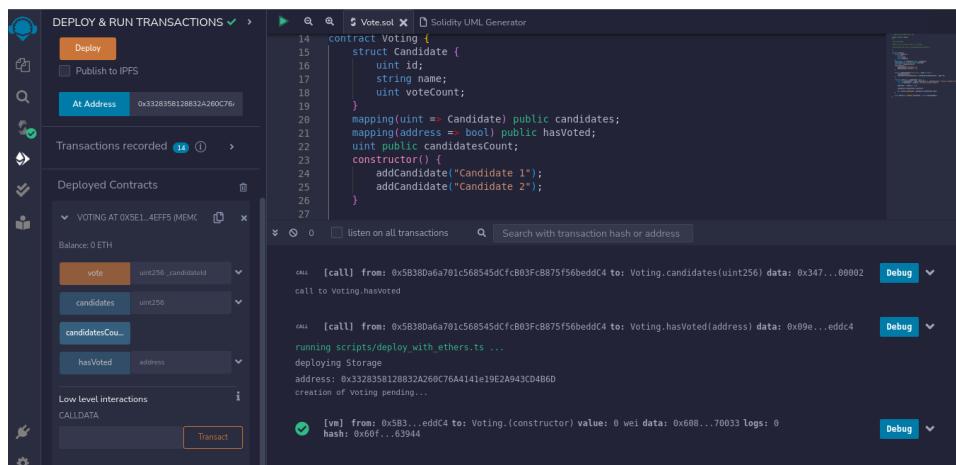
pragma solidity ^0.8.0;

/**
 * @title Storage
 * @dev Store & retrieve value in a variable
 * @custom:dev-run-script scripts/deploy_with_ETHERS.ts
 */
contract Voting {
    struct Candidate {
        uint id;
        string name;
        uint voteCount;
    }
    mapping(uint => Candidate) public candidates;
    mapping(address => bool) public hasVoted;
```

```
uint public candidatesCount;
constructor() {
    addCandidate("Candidate 1");
    addCandidate("Candidate 2");
}
function addCandidate(string memory _name) private {
    candidatesCount++;
    candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
}
function vote(uint _candidateId) public {
    require(_candidateId > 0 && _candidateId <= candidatesCount, "Invalid
candidate ID");
    require(!hasVoted[msg.sender], "You have already voted");
    hasVoted[msg.sender] = true;
    candidates[_candidateId].voteCount++;
    emit Voted(_candidateId, candidates[_candidateId].name);
}
event Voted(uint indexed candidateId, string candidateName);
```

**UML diagram for smart contract****Compiler Settings:**

On the Remix interface, go to the "Solidity Compiler" tab located on the left sidebar. Choose the appropriate compiler version based on your smart contract code. Click "Compile" to compile your contract.

**Deploying and Interacting with the Smart Contract:**

## Deploy the Contract:

Switch to the "Deploy & Run Transactions" tab in Remix.

Ensure the correct environment is selected (e.g., JavaScript VM, Injected Web3, etc.).

Click on "Deploy" to deploy your contract. You'll get a deployed contract address.

```

contract Voting {
    struct Candidate {
        uint id;
        string name;
        uint voteCount;
    }
    mapping(uint => Candidate) public candidates;
    mapping(address => bool) public hasVoted;
    uint public candidatesCount;
    constructor() {
        addCandidate("Candidate 1");
    }
    function addCandidate(string memory _name) {
        candidatesCount++;
        candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
    }
    function vote(uint _candidateId) {
        candidates[_candidateId].voteCount++;
        hasVoted[msg.sender] = true;
    }
    function getCandidate(uint id) view returns (string memory) {
        return candidates[id].name;
    }
    function totalVotesForCandidate(uint id) view returns (uint) {
        return candidates[id].voteCount;
    }
    function totalVotes() view returns (uint) {
        return candidatesCount;
    }
    function hasVotedCheck(address account) view returns (bool) {
        return hasVoted[account];
    }
}

```

The screenshot shows two instances of the Remix interface. The top instance shows the deployment of the contract with the address 0x4064B5033423Dcb6391Ac. The bottom instance shows the interaction with the deployed contract, specifically sending a vote for candidate 1. Both instances show the transaction logs and the current state of the contract's storage.

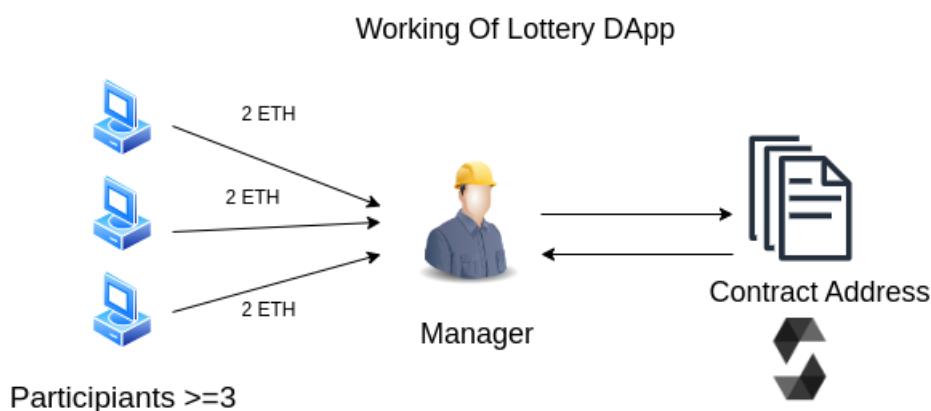
You can see all the successful interactions for the candidate values

**Practical 7 : Build a decentralized Lottery application (DApp) ,Using ERC-20 Tokens that combines Ethereum's Web3 and Solidity smart contracts deploy on Test network.**

Creating a lottery DApp involves several steps, including smart contract development using Solidity, setting up a local blockchain with Ganache, and interacting with the DApp using MetaMask.

Prerequisites:

1. Remix IDE
2. Metamask
3. Ganache



### Overview:

1. Install Ganache and start a local blockchain network.
2. Create a new workspace or use an existing one within Ganache.
3. Setup metamask ,and link metamask with ganache.
4. Create 4 accounts with 100 ETH each (1 manager ,3 participants)

Note: Manager is not a participant so manager don't need to have ETH in his wallet.

5. Open remix IDE write solidity code, connect metamask wallet to IDE
6. Deploy the smart contract and interact with it using metamask.

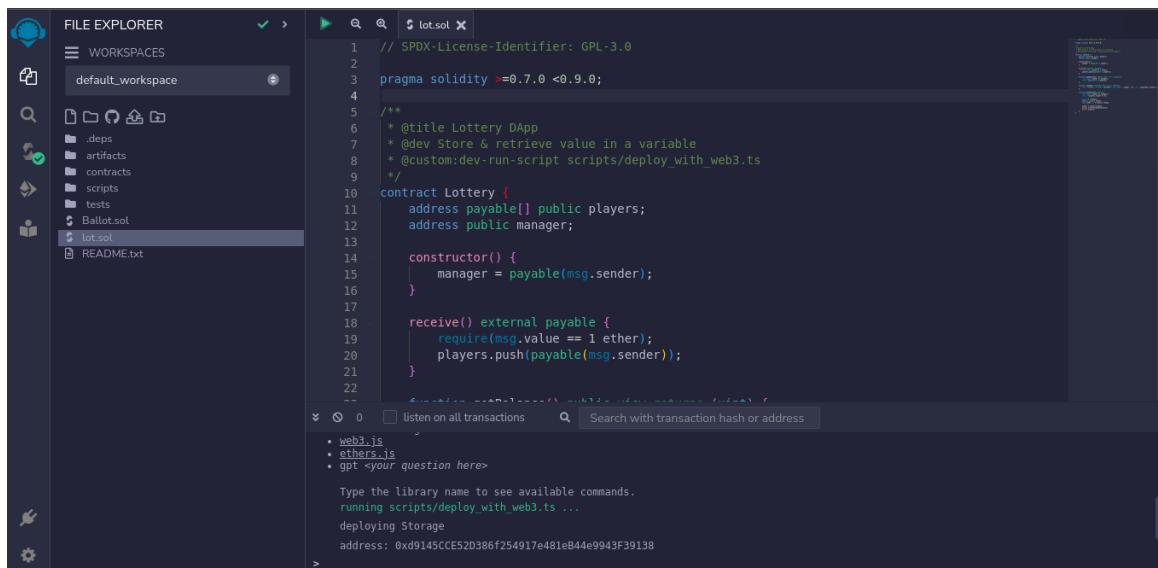
### Steps to follow:

## 1. Go to Remix IDE (<https://remix.ethereum.org/> ).

- It's an online Solidity IDE provided by Ethereum, where you can write, compile, and deploy Solidity smart contracts.

## 2. Step 2: Creating a New File

- Click on the "+" icon on the left-hand side or the "File" menu at the top to create a new file.
- Name the file with a .sol extension, for example, Lottery.sol.
- Write your Solidity code into this file.



```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;

/*
 * @title Lottery DApp
 * @dev Store & retrieve value in a variable
 * @custom:dev-run-script scripts/deploy_with_web3.ts
 */
contract Lottery {
    address payable[] public players;
    address public manager;

    constructor() {
        manager = payable(msg.sender);
    }

    receive() external payable {
        require(msg.value == 1 ether);
        players.push(payable(msg.sender));
    }
}
```

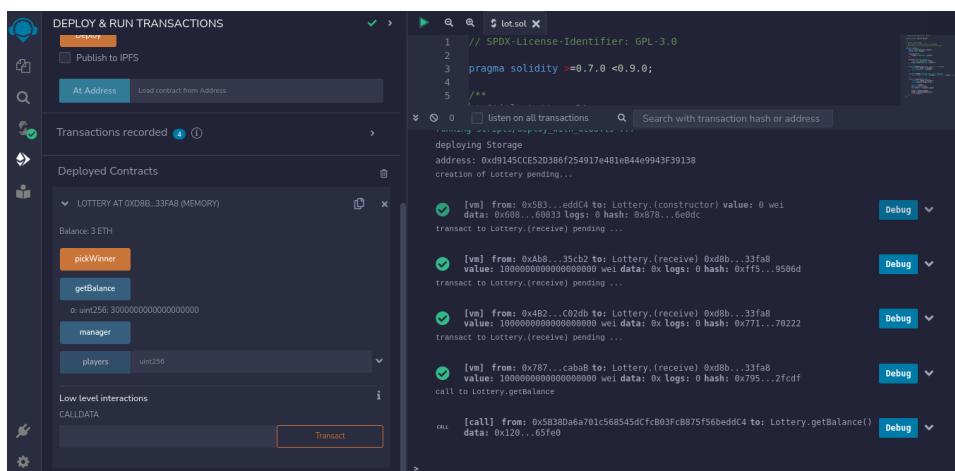
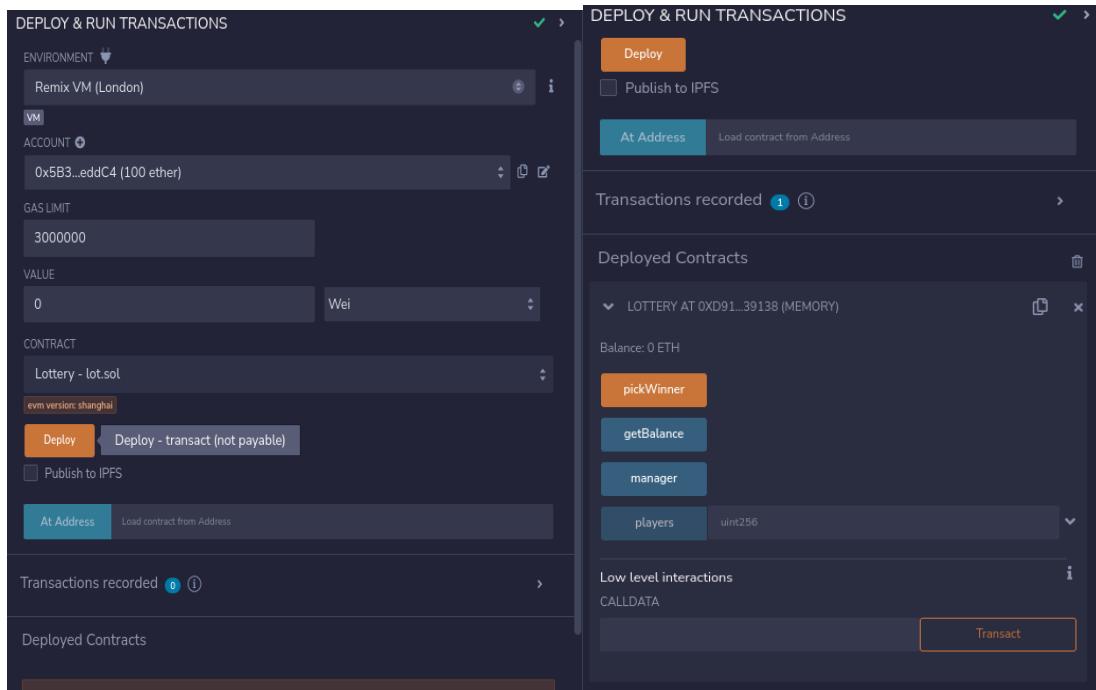
The Remix IDE interface includes a FILE EXPLORER on the left showing files like .deps, artifacts, contracts, scripts, tests, Ballot.sol, and lot.sol. The lot.sol file is selected. The main area shows the Solidity code for the Lottery contract. At the bottom, there are tabs for Solidity Compiler, Web3.js, Ethers.js, and GPT, with Solidity Compiler selected. A status bar at the bottom shows the address 0xd9145CCE52D386f254917e481e844e9943F39138.

## Step 3: Compiling and Run the Contract

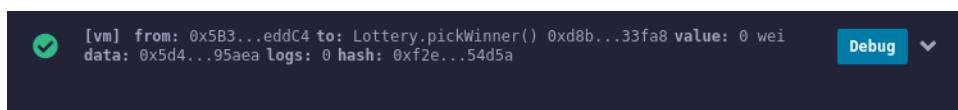
- On the left-hand side panel, go to the "Solidity Compiler" tab (it looks like a stack of papers).
- Ensure that the appropriate Solidity compiler version is selected.
- Click on "Compile Lottery.sol" (or the name of your file) to compile the contract.
- Verify that there are no errors in the compilation output panel.
- Click on deploy for first account to set it as manager.
- Then select another account set value to 2 Ether.
- Click on Transact button.
- Repeat more than equal to 3 times to set the number of participants to create Transaction pool for lottery.

Remix will display transaction details and logs when you perform actions like deploying a contract or calling its functions.

You can also check the transaction hash, gas usage, and status of transactions executed on the Ethereum blockchain.



- You can see the balance of the Transaction pool as manager
- Also, can see the transactions executed successfully.
- To Execute lottery, click on pickwinner button as manager the winner will be picker from all the participants.



- Above shows the Lottery Execution log into the console.

- After the correct lottery execution pool price will be send to the winner account, then reset the lottery DApp to prepare another Lottery Round

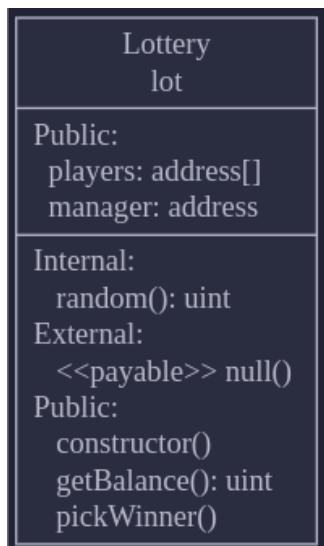
The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar lists accounts with their addresses and ether balances. One account, '0x4B2...C02db (101.99999999999951651 ether)', is highlighted. Below this is a list of recorded transactions. On the right, the code editor displays the Solidity source code for the 'lot.sol' contract. The code defines a lottery contract with a constructor, a receive function for deposits, and a 'pickWinner' function for selecting a random winner. Several transaction logs are shown on the right, indicating deposits and the execution of the 'pickWinner' function.

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;

/**
 * @title Lottery DApp
 * @dev Store & retrieve value in a variable
 * @custom:dev-run-script scripts/deploy_with_web3.ts
 */
contract Lottery {
    // ...
}
```

You can clearly see the pool price get transferred to address:

0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db



UML diagram to show the architecture, design, and implementation of our Lottery DApp.

**Aim:** Practical on Supply Chain Dapp.

## Background Information:

A Supply Chain DApp leverages blockchain technology and smart contracts to enhance the efficiency, transparency, and security of supply chain processes. Key components include product traceability, decentralized supplier management, real-time inventory tracking, quality assurance with immutable records, and transparent logistics operations. Smart contracts automate various aspects, such as agreements, reordering, and quality checks, ensuring a reliable and tamper-resistant supply chain.

## Steps to follow:

**Step 1:** Go to Remix IDE (<https://remix.ethereum.org/> ).

- It's an online Solidity IDE provided by Ethereum, where you can write, compile, and deploy Solidity smart contracts.

**Step 2:** Creating a New File

- Click on the "+" icon on the left-hand side or the "File" menu at the top to create a new file.
- Name the file with a .sol extension, for example, Supplychain.sol.
- Write your Solidity code into this file.

## Code:

```
// SPDX-License-Identifier: GPL-3
pragma solidity ^0.8.0;

/**
 * @title Storage
 * @dev Store & retrieve value in a variable
 * @custom:dev-run-script ./scripts/deploy_with_ETHERS.ts
 */

contract SupplyChain {
```

```
    uint public productCount = 0;
```

```
    struct Product {
```

```

        uint id;
        string name;
        uint quantity;
        address owner;
        address payable[] history;
    }

mapping(uint => Product) public products;

event ProductCreated(uint id, string name, uint quantity, address owner);
event ProductTransferred(uint id, address from, address to);

function createProduct(string memory _name, uint _quantity) public {
    productCount++;
    address payable[] memory initialHistory;
    products[productCount] = Product(productCount, _name, _quantity, msg.sender,
initialHistory);
    emit ProductCreated(productCount, _name, _quantity, msg.sender);
}

function transferProduct(uint _productId, address _newOwner) public {
    require(_productId > 0 && _productId <= productCount, "Invalid ID");
    Product storage product = products[_productId];
    require(msg.sender == product.owner, "Only the owner can transfer the product");

    product.owner = _newOwner;
    product.history.push(payable(_newOwner));

    emit ProductTransferred(_productId, msg.sender, _newOwner);
}

```

```

function getProductHistory(uint _productId) public view returns (address payable[] memory) {
    require(_productId > 0 && _productId <= productCount, "Invalid product ID");
    return products[_productId].history;
}
}

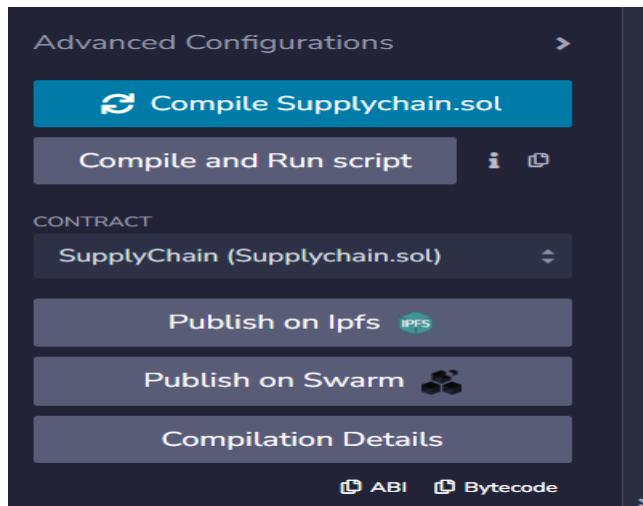
```

### Step 3: Compiler Settings

On the Remix interface, go to the "Solidity Compiler" tab located on the left sidebar.

Choose the appropriate compiler version based on your smart contract code.

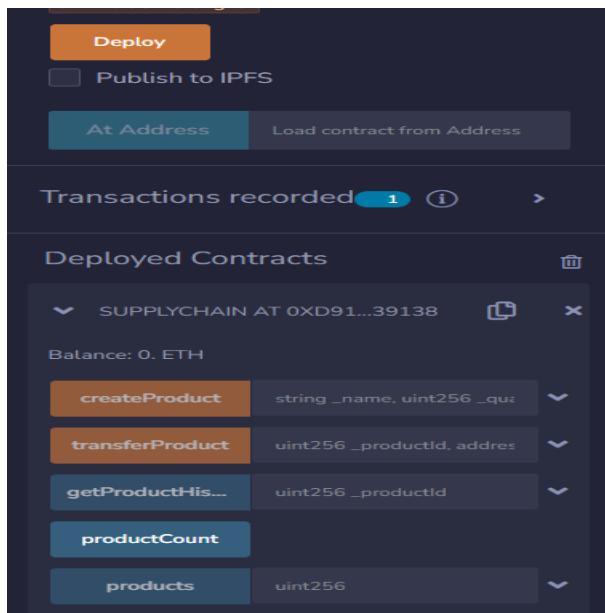
Click "Compile" to compile your contract.



### Step 4: Deploy the Contract

Switch to the "Deploy & Run Transactions" tab in Remix.

Click on "Deploy" to deploy your contract. You'll get a deployed contract address.



### Step 5:

Click on createProduct and add your product name in double quotes for example “cake” and then add quantity of the product.

Click on transact button you will get succeed transaction address.

