Quorum Documentation

Before we begin running our Quorum Blockchain, we need to set up the development environment for working with Quorum:

A. Software Requirements

Quorum development generally involves tools like:

- **Oracle JDK 8 JVM:** Minimum support version 8u131.
- **Virtualbox:** Version at time of development is 2.2.4.
- **Vagrant:** Version at time of development is 6.0.6.
- **Node.js:** Minimum support version 12.2.0.
- **Git**.

We use different libraries like **Jnacl, Libsodium and Kalium,** and repositories like **Quorum** and **Tessera**.

B. Setup Instructions

The instructions below will allow you to set up a Quorum development environment and run a basic Quorum example. Here, we define the set-up instructions for Mac.

1. Install Virtual box:
   - Visit ***https://www.virtualbox.org/wiki/Downloads*** for downloading the latest version of virtual box.
   - Click on the links provided, according to the system requirement.
   - Download and run the executable to install virtualbox.

2. Install Vagrant:
   - Visit ***https://www.vagrantup.com/downloads.html*** for downloading the latest version of vagrant.
   - Click on the links provided, according to the system requirement.
   - Download and run the executable to install vagrant.

3. Install Git:
   - Visit https://git-scm.com/ for downloading.
   - Download "64-Bit Git for Windows Setup" and run the executable file.

Note: For Mac or Linux, install the Xcode Command Line Tools that comes with Git among other things.

C. Setting up Quorum Client:

- Open Terminal, and make the working directory, named as "Quorum".
- Clone down the Quorum example, by running: `$ git clone https://github.com/jpmorganchase/quorum-examples`
- Move to the quorum-examples directory by `cd quorum-examples`.
- Run command `vagrant up`, and wait for some time.
- Now, run `vagrant ssh` command, to start the vagrant instance.
- Make Quorum directory and clone down the quorum example in it by running command `$ git clone https://github.com/jpmorganchase/quorum-examples.`

To install ethereum using PPA:

- Now move, run `sudo add-apt-repository -y ppa:ethereum/ethereum` command.
- Run `sudo apt-get update` command for making updates.
- Now install ethereum, by running `sudo apt-get install ethereum`.

To install ethereum from command line:

- Download go by running `$ wget https://d1.google.com/go/go1.12.5.linux-amd.64.tar.gz.`
- Unzip the downloaded tar file by running `$ tar -xvzf go1.12.5.linux-amd.64.tar.gz.`
- Clone down ethereum by running `$ git clone https://github.com/ethereum/go-ethereum.git.`
- Move to go-ethereum folder and run command `make geth`.

D. Installing Tessera:

- Go to main directory and run `sudo apt-get update && apt-get upgrade` command.
- Now run `sudo apt-get install default-jdk`, to download the latest version of java.
- Check the installation, by running `java -version.`
- Now download apache maven, by running `$ wget https://mirrors.estointernet.in/apache/maven-3/3.6.1/binaries/apache-maven-3.6.1-bin.tar.gz.`
- Once downloaded, run `tar -xvzf apache-maven-3.6.1-bin.tar.gz` to unzip the file.
- Now run `sudo apt update` command, to upgrade the packages if any.
- Then run `sudo apt install maven` command to install maven packages.
- Open a new terminal and move to the working directory where maven is installed.
- Run `mvn -version` command to check the installation.

- Then clone down the tessera repository by running:
  `$ git clone https://github.com/jpmorganchase/tessera.git.`
- Move to tessera folder and run `mvn install` command.

E. Running Nodes:

- Move to 7nodes folder.

For running nodes using Raft Consensus:

- Run `./raft-init.sh` command.
- Now run `./raft-start.sh` command. This will start the nodes.
- To stop, run `./stop.sh` command.

For running nodes using Istanbul Consensus:

- Run `./istanbul-init.sh` command.
- Now run `./istanbul-start.sh` command. This will start the nodes.
- To stop, run `./stop.sh` command.

For running nodes using Clique POA Consensus:

- Run `./clique-init.sh` command.
- Now run `./clique-start.sh` command. This will start the nodes.
- To stop, run `./stop.sh` command.

F. Connecting Truffle to Quorum:

To set up Truffle, we're going to start by creating a bare Truffle project, without any contracts or code. For that:

- Now leave the terminal where nodes are running.
- Open a new terminal, and move to the Quorum directory.
- Create a new directory named as `quorumprojects`.
- Run `truffle init` command.
- Open file truffle-config.js file by running `vi truffle-config.js` command.
- Change port number to 22000. Save and exit.
- Now move to contract directory, create a new contract named as `NewSimpleStorage.sol`, and add the commands. Save and exit.
- Run `truffle compile` command to compile the contracts.
- Now create a new migration called `2_deploy_simplestorage.js` within your migrations directory. Make the changes and save.
- Now open a new terminal and run `geth --rpc` command.
- In previous terminal, run `truffle develop` command.
- Now run `migrate` command.

- To get the instance value of NewSimpleStorage contract, run `NewSimpleStorage.deployed().then(function(instance){return instance.get();})`

G. Interacting with node privately:

- Create a file named as **sampletx.js** and save it in the root of your project.
- Set instance value to 65 in sampletx.js file.
- Run `exec sampletx.js` command.
- To get updated instance value, run `NewSimpleStorage.deployed().then(function(instance){return instance.get();})`

H. Reducing Number of Nodes (From 7nodes to 4Nodes):

- Move to 7nodes folder.
- Open `raft-start.sh file` and comment 5,6,7 nodes starting command.
- Save and Exit.
- Open `tessera-start.sh` file and change the instances of `for i in {1..7} to for i in {1..4}`. Save and Exit.
- Run `./raft-init.sh` command.
- Then run `./raft-start.sh` command. This will start the 4 nodes now.