National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Alby Sabu
Student ID: 21240906

School of Computing
National College of Ireland

Supervisor: Dr. Shivani Jaswal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Alby Sabu
Student ID:	21240906
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Shivani Jaswal
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	1075
Page Count:	7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Alby Sabu
Date:	13th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Alby Sabu
21240906

1 Introduction

In this document, the steps for setting up the prototype system are discussed. This includes the steps for installing the prerequisites, details of necessary tools and their versions, and also the steps for starting the Hyperledger Fabric network and Web application.

2 System Configuration

The prototype developed for demonstrating the research study was implemented on an Ubuntu 20.04 operating system environment hosted on the hardware configurations as shown in figure 1.

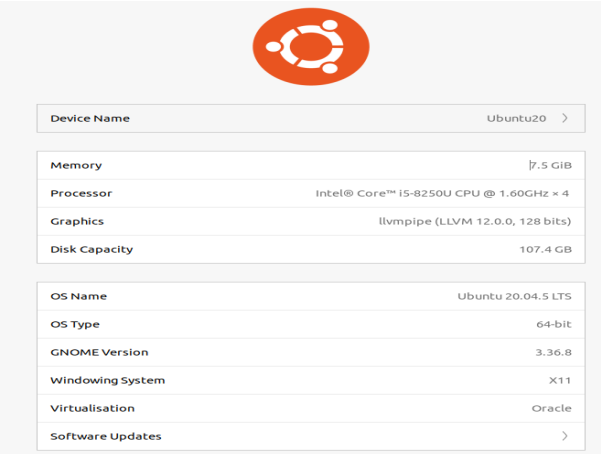

Figure 1: Hardware requirements of the system

3 Installing Prerequisites

To create the Hyperledger Fabric test network as well as to run the developed web application the required dependencies and prerequisites have to be installed Fabric (2023).

3.1 Prerequisites for setting up Hyperledger Fabric test network

- Git: Install git for Ubuntu using $ *sudo apt-get install git* command in the terminal. The installed version is shown in the figure 2.

Figure 2: Installed version of Git

- Curl: Install cURL for ubuntu using $ *sudo apt-get install curl* command in the terminal. The version that was installed is shown in the figure 3.

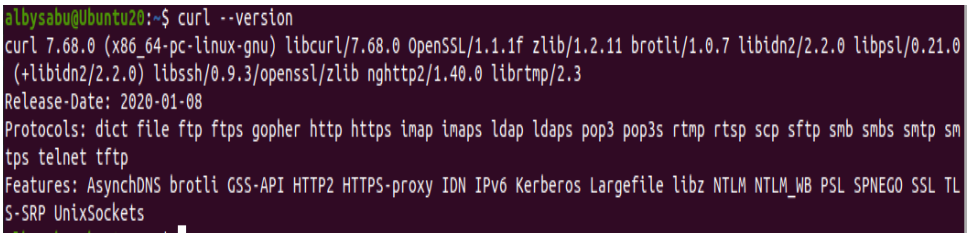

Figure 3: Installed version of cURL

- Docker: Install docker and docker-compose for Ubuntu using the $ *sudo apt-get -y install docker-compose* command and verify the versions installed as shown in figure 4. After successful installation, start the docker daemon using $ *sudo systemctl start docker* in the terminal window.

Figure 4: Installed version of Docker and Docker-compose

- Go: Since the chaincode for the system utilizes the Go programming language for developing the smart contract and writing the required functions it is essential that Go should be installed. To install Go download the installation file for Ubuntu and use the $ *sudo tar -xvf <path of go installation file>* on the terminal.

3.2 Prerequisites for running the web application

For the web application to start, all the required modules and dependencies are to be installed on the system. To install these requirements run $ *pip install -r requirements.txt* after navigating to the directory that contains the source code of the web application. The installed dependencies can be found in the figure 5.

```
asgiref==3.6.0
autopep8==2.0.2
backports.zoneinfo==0.2.1
certifi==2022.12.7
charset-normalizer==3.1.0
Django==4.1.7
django-filter==22.1
djangorestframework==3.14.0
djangorestframework-simplejwt==5.2.2
idna==3.4
importlib-metadata==6.0.0
Markdown==3.4.1
Pillow==9.4.0
pycodestyle==2.10.0
PyJWT==2.6.0
pytz==2022.7.1
requests==2.28.2
sqlparse==0.4.3
tomli==2.0.1
urllib3==1.26.14
zipp==3.15.0

```

Figure 5: Required dependencies and packages

4 Implementation Steps

4.1 Cloning the project source code

The source code required for setting up the project prototype can be cloned from the following GitHub repository: https://github.com/albysabu9/researchproject.git The folder structure is divided into three sub-folders. DigitalAssetManagement contains the code for the web application. Fabric-sample contains the configurations for the Hyperledger fabric test network. Caliper-workspace contains the configuration files for benchmarking the system.

4.2 Starting the Hyperledger Fabric Test network

- To install the Hyperledger Fabric, an installation script is required. Open a terminal in the root directory of the project and run the following code to get the installation script. $ *curl -sSLO https://raw.githubusercontent.com/hyperledger/fabric/main/scripts/install-fabric.sh && chmod +x install-fabric.sh*

- After getting the installation script, run the following code to install.
  $ *./install-fabric.sh d s b*

- To give the directory required permissions run the following code.
  $ *chmod -R 777 ./*

- After the fabric network has completed the installation successfully, start the fabric network using the command $ *sudo ./network.sh up* after moving to the fabric-samples directory. To see the status of the nodes, use the following code

$ *docker ps -a*
The status of the node in the network is shown in figure 6.

Figure 6: Fabric network nodes and running status

- Create a network channel that will enable the two organizations and peers to communicate and create transactions between them. Run the following code after moving to the test-network directory of the project.
  $ *sudo ./network.sh createChannel*
  The created network channel can be listed by running $ *peer channel list* as shown in figure 7.

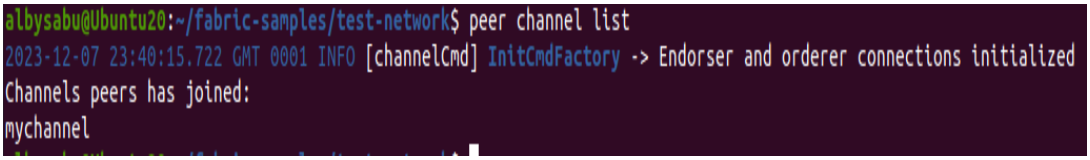

Figure 7: Created channel on the network

- After the channel is created, the developed chaincode with the functions that are to be invoked on the data operations performed by the user needs to be deployed on the created channel.

  $ *sudo ./network.sh deployCC -ccn basic -ccp ../path-to-the-chaincode/chaincode-go -ccl go*
  Once the chaincode is deployed, the web application for the users to interact with the Hyperledger Fabric network can be started.

4.3 Starting the Web Application Interface

- To start the user interface of the web application, First change the directory to the DigitalAssetManagement directory of the project and execute the following command in the terminal.
  $ *python3 manage.py runserver*
  This will open the user interface in the default browser on the localhost URL: http://127.0.0.1:8080
  The login screen will be displayed for the user to login to the client application as shown in figure 8.

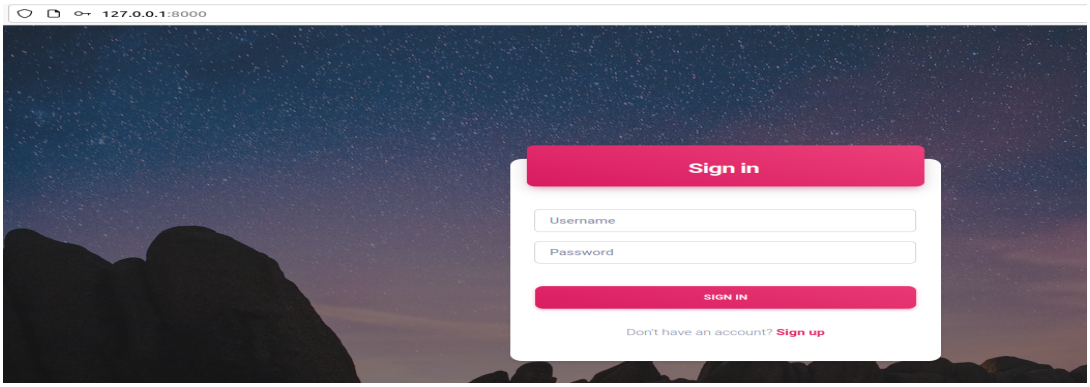

Figure 8: Web application interface

4.4 Querying the Peer CLI to verify the transactions

- The peer nodes in the network can query the Hyperledger fabric blockchain to verify the transactions that are created when the chaincode is invoked on the data operations of the users through the web application. Change the directory to the fabric-sample directory of the project in the terminal and then set the path that will point to the configuration files using the below commands.
  $ *export PATH=$PWD/bin:$PATH*
  $ *export FABRIC_CFG_PATH=$PWD/config/*

- To query the fabric ledger using the Peer CLI, use the following command. The records that are written as transactions to the blockchain can be seen as shown in figure 9.
  $ *peer chaincode query -C mychannel -n basic -c '"Args":["GetAllAssets"]'*

Figure 9: Querying the ledger using peer CLI

5 Evaluation Steps

5.1 Performance benchmarking using Hyperledger Caliper

- To evaluate the performance of the configured Hyperledger fabric system and verify that the solution offers better performance in terms of scalability and efficiency we have used Hyperledger caliper tool in conducting the benchmarking tests Caliper (2023). To perform the benchmarking, move to the caliper-workspace directory of the project in the terminal. The required configuration files for benchmark and network are defined in the caliper workspace directory. The workload modules for read and write operations are also placed in the directory. The directory structure is represented in the figure 10.

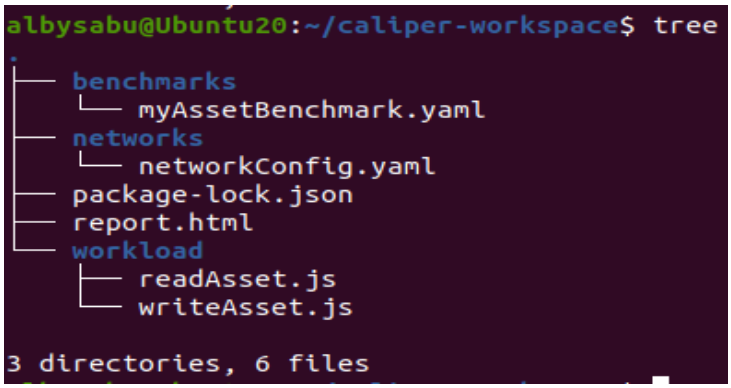

Figure 10: Configuration files required for benchmark

- The benchmark test can be run by executing the caliper CLI command in the terminal as shown in figure 11.

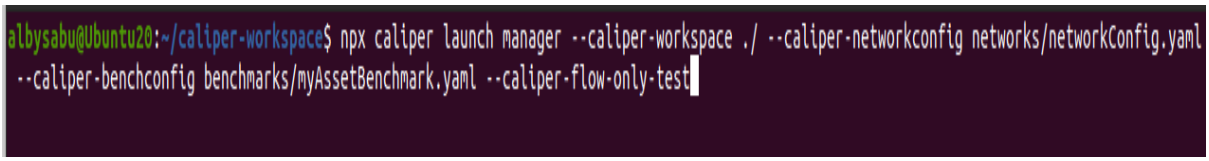

Figure 11: Running the benchmark tests using caliper CLI

- Once the test is successfully executed a report will be generated and saved in the caliper workspace directory that contains all the test information related to the performance metrics and the resource utilization. The benchmark report generated is shown in figure 12. The tests can be repeated multiple times by varying the parameters such as transaction load that is specified in the benchmark configuration file to see the variations that can occur to the metrics like latency, throughput, and resource utilization of the nodes in the network.

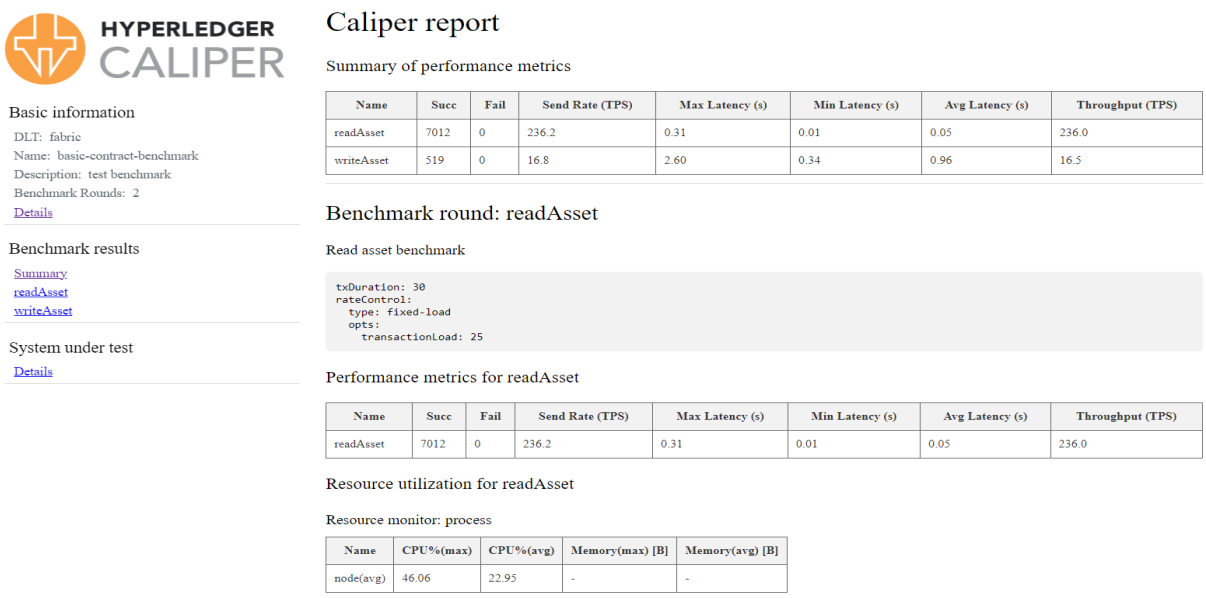

Figure 12: Benchmark report generated by the Caliper tool

References

Caliper, H. (2023). Hyperledger caliper - getting started.
URL: *https://hyperledger.github.io/caliper/v0.5.0/getting-started/*

Fabric, H. (2023). Hyperledger fabric documentation - getting started. Accessed: November 13, 2023.
URL: *https://hyperledger-fabric.readthedocs.io/en/release-2.2/getting*$_s$*tarted.html*