Test drive Hyperledger™ Composer Playground and blockchain demo app with Hyperledger™ Fabric on IBM LinuxONE™ server

Yongkook(Alex) Kim @likepunk

System z Solutions Architect - Vicom Infinity

http://blockchain.infinite-blue.com

2017 June 23 VMWorkshop at Ohio State University
The goal of this lab is:

1) to let you experience bringing up the Hyperledger™ network/fabrics on LinuxONE™ server
2) and then deploy sample applications using either Fabric-composer playground and/or Marbles App on LinuxONE™ server

Estimated run time: 2 hours

This lab utilizes Hyperledger™ Fabric v0.6 and composer-playground v0.7.5 at the time of writing this material (June 22, 2017). It may be updated later when Fabric version 1.0 and composer-playground gets stabilized in the future.

Special thanks to following people who helped make this lab possible:

Barry Silliman @ IBM Washington Systems Center
John Harrison @ IBM Washington Systems Center
Steve LaFalce @ IBM z Systems Technical Enablement
Chee Ye @ IBM z Systems Technical Enablement
Matt Lucas @ IBM Global Blockchain Enablement
[0] Prepare LinuxONE™ Community Cloud Server

Sign Up for the LinuxONE™ community server

Visit LinuxONE™ cloud service web site (https://developer.ibm.com/linuxone/request-a-trial) and request for the free service and create SLES12 SP2 - Medium sized server. It may take up to a day or two to get a login credentials. Once you get id/passwd, sign in as below.

Click on left side ‘Virtual Servers/ Manage Instances’ where you can create a Linux server.
Enter instance details(step1), select SLES12 SP2(step2) and LinuxONE-Medium as a flavor(step3).
Step 1: Enter instance details

- Instance name: AlexSLES12
- Instance Description: VMwareshop Blockchain

Step 2: Select an image

- RHEL 6.7
- RHEL 7.2
- SLES 11 SP4
- SLES12 SP2

Step 3: Select a flavor

- LinuxONE-Small
  - CPU: 1 core(s)
  - Memory: 2048MB
- LinuxONE-Medium
  - CPU: 2 core(s)
  - Memory: 4096MB

Current selection:

- Name: AlexSLES12
Using SSH key that was provided, login to the terminal. If you don’t have SSH terminal nor don’t know how to connect please refer to this document → https://developer.ibm.com/linuxone/wp-content/uploads/sites/57/virtual-servers-quick-start.pdf

Install docker-compose tool

Since most of settings are already pre-built in LinuxONE community server, only thing you need to do is installing ‘docker-compose’ tool. Follow directions from below or website for other distros and make sure you have docker-compose running.

https://github.com/linux-on-ibm-z/docs/wiki/Building-Docker-Compose?cm_mc_uid=8128849110413994410116&cm_mc_sid_50200000=1497929327

1) Install dependencies

```bash
sudo zypper install -y python-pyOpenSSL python-setuptools
```

2) Install pip with easy_install

```bash
sudo easy_install pip
```
3) Install docker-compose itself
   
   `sudo pip install docker-compose==1.13.0`

4) verify the Docker Compose installation
   
   `docker-compose version`
[1] Install/setup Hyperledger Fabric v0.6

Following steps will guide you to install/run Hyperledger Fabric on your server. You can either run it with {a single peer + membership services} OR {four peers + membership services}

Reference: https://hub.docker.com/r/ibmblockchain/fabric-peer/

1.1 Switch user to root:

During the lab, all the commands will be entered as ‘root’ user id unless specified otherwise. If you are not currently switched to root yet, please enter: ‘sudo su –’ at the shell command line.

1.2 Pre-requisites

- git version 2.12 or higher (source management tool, already installed for the lab)
- Docker version 1.12 or higher (already installed for the lab)
- Docker-compose version 1.11.2 or higher (already installed for the lab, so you don’t need to do anything here. If not, or if you want to try in your own LinuxONE image, follow steps in below site: https://github.com/linux-on-ibm-z/docs/wiki/Building-Docker-Compose)

About Docker:

Docker is a container/virtualization technology and a key component of Hyperledger. It is not required to know during the lab but it is important to understand basic operations if you want to continue working with Hyperledger.

If you want to learn more about Docker, please refer to https://docs.docker.com/

There is a very nice all-inclusive playground website you can learn and try on Docker technology in here: http://training.play-with-docker.com/

1.3 Clone docker-compose files and bring up the Hyperledger fabric

Change directory to where you want to copy Hyperledger fabric configuration files. For this lab go to /data:
Please enter ‘cd /data’,
and then enter ‘git clone https://github.com/IBM-Blockchain/fabric-images.git’
Now you will see ‘git’ is cloning files from github site down to your local directory. Once you get prompt back, change directory to /data/fabric-images/docker-compose/

Please enter ‘cd fabric-images/docker-compose’

Once changed the directory,

Please enter ‘. setenv.sh’

→ make sure you enter dot(.) and space. This will check what HW platform the Hyperledger fabric will be running on. We are running on z Systems(s390x). Now bring up the Hyperledger fabric by entering following.

Please enter ‘docker-compose -f single-peer-ca.yaml up’

Docker-compose is a command to invoke multiple Docker images to bring up, and connect them together to make a ‘service’ or ‘services’. Note that you are bringing up a Hyperledger network with single peers and one membership services. You can also have four peers + membership services if you specify ‘four-peer-ca.yaml’ instead.
Now you will see a lot of messages scrolling and most of time it is hard to see – not to worry about them for now. Once you see dockercompose_baseimage_1 exited with code 0 (1st screenshot), and about a minute or so later you will see a message ‘Discovery knows about:’ for each peer.

Now Hyperledger fabric is ready to take an action!

All of the messages are coming from two(2) different Docker images (Membership Services + 1 peer). If this was four-peer model (if you used four-peer-ca.yaml for docker-compose) you can see there are messages keep checking the connections to each peer to see if there is any new transaction is happening. Let this screen runs itself and move to the next step.

** To move to next section, open another terminal and login to your LinuxONE server

There are many ways to verify if the Hyperledger fabric is running okay. For this lab, you can either use a web browser or command shell to verify it.

1.4 Testing and verifying your local network using Web Browser

Let’s verify if the Hyperledger fabric is working with a web browser. Hyperledger Fabric’s peer has a RestAPI port, which is normally mapped to 7050. Open a web browser and type following in the address section.

<your LinuxONE server IP>:7050/chain

You will see something like following screen:

What this means is currently Hyperledger fabric created the genesis block (initial block of the blockchain) with initial hash value.
Now below you will see how peers are registered in the Hyperledger Fabric, and their address/port, peer type and cryptographic ID of the peer by typing your LinuxONE server IP>:7050/network/peers

To get more information about how to use pre-defined REST APIs for Hyperledger peers, please visit http://hyperledger-fabric.readthedocs.io/en/stable/API/CoreAPI/#rest-api

** To go to section #2, please open another terminal window and connect to LinuxONE shell and go to next page.
2. Install/run Hyperledger Fabric Composer Playground

Following steps will guide you to install/run Hyperledger Fabric Composer Playground (composer-playground) app for on your server that fabric is running. Composer-playground app is node.js based web tool that enables a developer to create Hyperledger applications (chaincode), related access controls, and combine them with business logics relatively simple way. For more information, visit https://hyperledger.github.io/composer/.

2.1 Map port 8080 to open and listen

Since we are using Docker container for the app and it connects to a local Hyperledger fabric also on docker containers, we will make composer-playground app to use host network and interface external ip address/port. To enable it we will need to enter following iptables command. If you like to know more about why please refer to this web site (http://www.dasblinkenlichten.com/docker-networking-101-host-mode/)

Make sure you are still in ‘root’ id (sudo su), and execute following command:

```
sudo iptables -I INPUT -p tcp -m tcp --dport 8080 -j ACCEPT
```

2.2 Run composer-playground app using docker run

2.2.1) Now enter following command to download already installed composer-playground app from the docker hub, and invoke webservices.

From any directory enter:

```
sudo docker run --net=host -it likepunk/composer_playground:v075_s390 node
cli
```

(Note for two dashes for --net)
2.2.2) Bring up a web-browser and enter `<your LinuxONE IP address:8080>` in the address. You should be able to see following screen. Click a globe shaped icon at upper right corner. It will pop up a window for connection profiles settings. Choose Hyperledger Fabric v0.6 at this time (v1.0 is not working with this version of composer-playground).
Import/Create a Connection Profile

Upload a Connection Profile from your computer:

Drop here to upload or browse

Or create a connection profile, select your Blockchain platform:

- Hyperledger Fabric v0.6
  v0.6 Connection Profile.

- Hyperledger Fabric v1.0
  v1.0 Connection Profile.

[Cancel] [Add]
2.2.3) Enter each section as shown in the screenshot below. Pick a connection profile name. Leave Peer URL, Membership Services URL, Event Hub URL as they are. Change ‘Key Value Store Directory Path’ into ‘./’ (dot slash).

Change Deploy Wait Time and Invoke Wait Time into 500, 100 each.
2.2.4) Once Saved, you can click ‘Use this Profile’ button. It will ask you to connect with an identity. Click ‘Add identity’ for now and enter following value ID/SECRET, and then click ‘Connect’.

ID: test_user1
SECRET: jGI6ImkuDo
You should see a pop-up messages saying it’s trying to deploy business network files and once it’s done it will say ‘Connection Successful’ in upper-right corner.

Now you’re ready to move with the Hyperledger Composer Lab. Bring up Hyperledger composer playground lab PDF( http://bit.ly/2t1h5je ) and jump to section 1.2. NOTE: While you’re doing this lab, try to examine blockchain contents using browser that was explained in section 1.4. and another example shown below to check actual contents of blockchain data.
You can check out the initial block's details by entering following in the address section:

<your LinuxONE server IP>:7050/chain/blocks/1

You can change the number at the end to explore different blocks – use it as an index based on history of the blocks.

Above JSON formatted data shows the contents of the block#1. You can copy the contents of the payload(highlighted) and pasted into following website and DECODE to get actual payload text – it’s encoded with base64 encoding algorithm. Sometimes you need to decode again what’s decoded to get more details.

https://www.base64decode.org/
This is a bonus lab – in case you have extra time left, feel free to try it

To go to section #3, please open another terminal window and connect to LinuxONE shell and begin the lab.

[3] Running Marbles App

Following steps will guide you to install/run Marbles app demo for Hyperledger on your server. You can install Marbles app either locally or at the IBM Bluemix easily with instruction described in github below, but to minimize errors we will provide you an docker image already integrated with all components needed. [https://github.com/ibm-blockchain/marbles/tree/v2.0](https://github.com/ibm-blockchain/marbles/tree/v2.0)

(if you have completed section [2] and removed all the docker containers in cache, go to section [1] and run fabric network then come back to this step)

Assuming you completed the first lab successfully, we will deploy a sample app called ‘Marbles’. It’s a very intuitive GUI based blockchain app to create/trade marbles. Marbles represent assets, and whenever an asset gets created and traded, new blocks will be created and shown in the console. You can ask your peers to connect to your website and simulate multi-user based blockchain network communication.

FYI, our implementation of Marbles app is a bit different from the original IBM Marbles. We(Vicom Infinity) modified GUI so that we can simulate four-peer blockchain network.

Again, this is the lab based on Linux for IBM z Systems™ or IBM LinuxONE ™, so the binary file for Docker image is for s390x architecture. It was tested with SLES12 SP2 as well as RHEL7.2. If you like to get a Docker image for x86, please contact us.

3.1 Map port 3000 to open and listen

Since we are using Docker container for the app and it connects to a local Hyperledger fabric also on Docker containers, we will make marbles app to use host network and interface external ip address/port. To enable it we will need to enter following iptables command. If you like to know more about why please refer to this web site [http://www.dasblinkenlichten.com/docker-networking-101-host-mode/](http://www.dasblinkenlichten.com/docker-networking-101-host-mode/)
Make sure you are still in ‘root’ id (sudo su), and execute following command:

```
iptables -I INPUT -p tcp -m tcp --dport 3000 -j ACCEPT
```

```
[root@openmf ~]# iptables -I INPUT -p tcp -m tcp --dport 3000 -j ACCEPT
```

3.2. Run marbles app using Docker run

Now enter following command to download already installed marbles app from the docker hub, and invoke web services.

From any directory enter: `docker run --net=host likepunk/marbles` (note for two dashes for -net)

```
docker run --net=host likepunk/marbles
```

You will see following start-up logs. Wait until you see a message like `---Websocket Up---`
3.3. Play Marbles!

Now bring any web browser and type IP address of your server with port number 3000. If you don’t know your host server ip, then type following: `ip addr show|grep eth0`
Now connect to your marbles app using the ip shown above with port 3000 at the browser:

http://<your LinuxONE's IP address>:3000 and you will see Marbles web page shows as following:

![Marbles Web Page](image)

Click on ‘CREATE’ on the center then create a marble asset of your choice
You can see a large green marble gets created, at the same time a new block is added to the blockchain. You can see it in action by clicking ‘VIEW TX BLOCKS’. It will show you details of the block(chaincode ID, payload etc).
VI Marbles

Vicom

Infinity

Bluecorp

Redcorp

TIME 04/28/2017 05:15AM UTC

Block Height: 10
- Created: 04-28-2017 05:15am UTC
- UUID: undefined
- Type: 2
- CC ID: 1be665d000a90b28383d32d1117dcd4f5027d47640044d0f1de5349a1a657e855ebb7223272da9763f426e8d8c6991b81222d0147fe0b060a178dc
- Payload: init_marble green1 green35 vicom

Block Height: 10
- Created: 04-28-2017 05:15am UTC
- UUID: undefined
- Type: 2
- CC ID: 1be665d000a90b28383d32d1117dcd4f5027d47640044d0f1de5349a1a657e855ebb7223272da9763f426e8d8c6991b81222d0147fe0b060a178dc
- Payload: init_marble green1 green35 vicom
Now try to switch ID to INFINITY (or ask other people to connect using another browser/PC or mobile device), and create marbles. You can move marbles by drag&drop.

Now you’re done with the Lab! You can bring down all the docker images before next Lab by typing following:

```
# docker stop $(docker ps -a -q)
# docker rm -f $(docker ps -a -q)
```
Explorer further more with trading menu. To get more information about it please see below github site. There is also a newer version of Marbles app works with Hyperledger Fabric v1.0.

https://github.com/ibm-blockchain/marbles/tree/v2.0

[x] Extra Information/Debugging

If you experience an error that complains about Chaincode not being deployed in the Hyperledger fabric logs or Marbles app showing it can’t find the Hyperledger fabric, try insert following into ‘four-peer-ca.yaml’ file under fabric-images/docker-compose:

```yaml
- CORE_CHAINCODE_DEPLOYTIMEOUT=500000
```
Once edited and saved, stop prior docker processes and remove all of the containers from cache by entering:

```
# docker stop $(docker ps -a -q)
# docker rm -f $(docker ps -a -q)
```

This is very useful command whenever you want to start from scratch. You want to make sure there’s no residues from prior runs.

After you get the prompt back, type following commands to bring up the Hyperledger fabric as well as run marbles demo (you will need two shell screens).

```
# docker-compose -f four-peer-ca.yaml up
# docker run --net=host likepunk/marbles
```