

JAY-RALD CALIXTRO DELA CRUZ

Blockchain Cadet

INSTALLATION OF HYPERLEDGER

Laptop Specification

Manufacturer	Acer
Unit	Aspire 3 – A315-41G-R4BW
Processor	AMD Ryzen 5 2500U
RAM	4 GB DDR4
OS	Ubuntu 18.04.1 LTS
OS Type	64 Bit
Kernel	Linux 4.15.0-43-generic (x86_64)

Installation Method

The first step is to install all the hyperledger prerequisites provided.

- Operating Systems: Ubuntu Linux 14.04 / 16.04 LTS (both 64-bit), or Mac OS 10.12
- Docker Engine: Version 17.03 or higher
- Docker-Compose: Version 1.8 or higher
- Node: 8.9 or higher (note version 9 is not supported)
- npm: v5.x
- git: 2.9.x or higher
- Python: 2.7.x
- A code editor of your choice, we recommend VSCode.

Since my laptop is running on Ubuntu Operating System, I installed all prerequisites using this commands on Terminal.

- `curl -O https://hyperledger.github.io/composer/v0.19/prereqs-ubuntu.sh`
- `chmod u+x prereqs-ubuntu.sh`
- `./prereqs-ubuntu.sh`

After Running all commands without any error I proceeded to the installation of Development Environment. I installed Environment using this commands on Terminal.

To Install CLI Tools :

- `npm install -g composer-cli@0.19`
- `npm install -g composer-rest-server@0.19`
- `npm install -g generator-hyperledger-composer@0.19`
- `npm install -g yo`

To Install Playground :

- `npm install -g composer-playground@0.19`

To Install Hyperledger Fabric :

- `mkdir ~/fabric-dev-servers && cd ~/fabric-dev-servers`
- `curl -O https://raw.githubusercontent.com/hyperledger/composer-tools/master/packages/fabric-dev-servers/fabric-dev-servers.tar.gz`
- `tar -xvf fabric-dev-servers.tar.gz`
- `cd ~/fabric-dev-servers`
- `export FABRIC_VERSION=hlfv11`
- `./downloadFabric.sh`

Install Go language

- download Go for Linux here <https://golang.org/dl/>
- Go to the directory where the file located.
- Right click inside the folder and Open in terminal
- Type "**tar -C /usr/local -xzf go**" then press tab to complete then enter it should be look like this Eg: "**tar -C /usr/local -xzf go1.11.5.linux-amd64.tar.gz**"
 - Add Go environment variable
 - Type this

'export PATH=\$PATH:/usr/local/go/bin'

```
'export GOPATH=$HOME/go'  
'export PATH=$PATH:$GOPATH/bin'
```

I encountered no errors while installing or using any commands of hyperledger.

INVOICE TRACKING PROGRAM

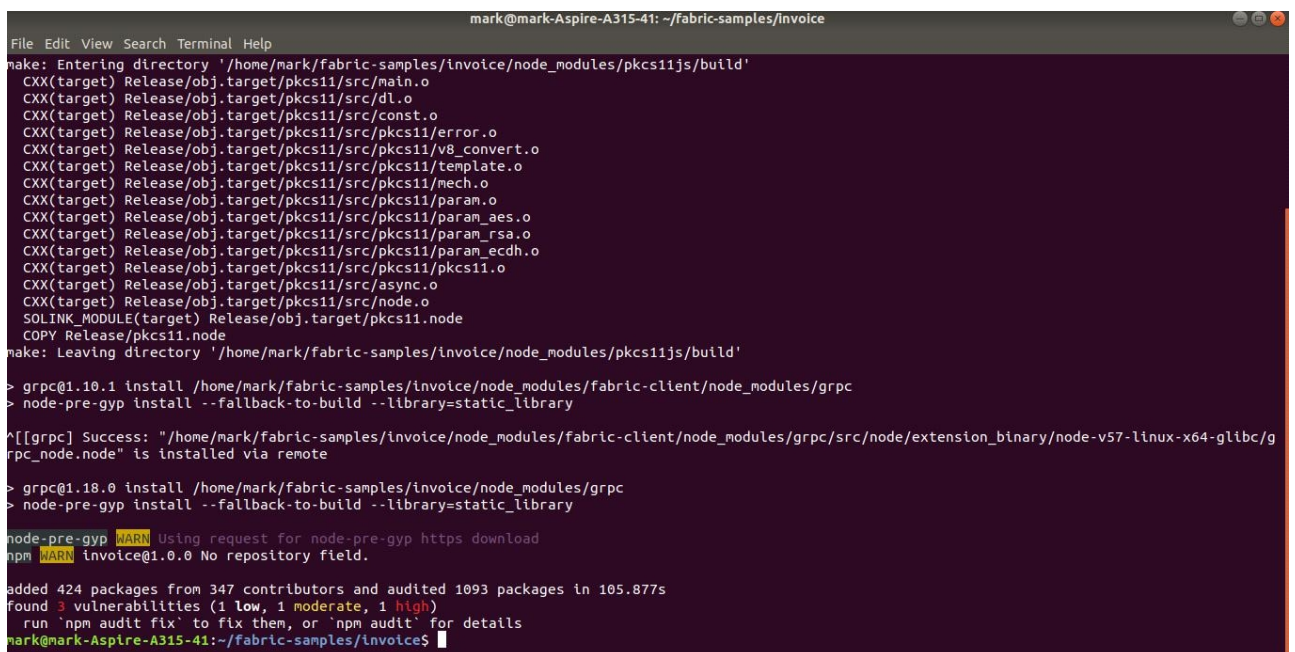
- ✦ First, Clone this repository open terminal and type :

“*mkdir hyperledger && cd hyperledger*”

“*curl -sSL http://bit.ly/2ysbOFE | bash -s -- 1.4.0*”

“*git clone https://github.com/khrandm/blockchain-training-labs*”
- ✦ Open the **blockchain-training-labs** folder then copy folders named **chaincode** and **supply**.
- ✦ Go to **fabric-sample** folder and paste the folders you copied .
- ✦ Go back to the **fabric sample** directory, right-click then **Open in Terminal** and type

“*npm install*”



```
mark@mark-Aspire-A315-41: ~/fabric-samples/invoice
File Edit View Search Terminal Help
make: Entering directory '/home/mark/fabric-samples/invoice/node_modules/pkcs11js/build'
CXX(target) Release/obj.target/pkcs11/src/main.o
CXX(target) Release/obj.target/pkcs11/src/dl.o
CXX(target) Release/obj.target/pkcs11/src/const.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/error.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/v8_convert.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/template.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/mech.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/param.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/param_aes.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/param_rsa.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/param_ecdh.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/pkcs11.o
CXX(target) Release/obj.target/pkcs11/src/async.o
CXX(target) Release/obj.target/pkcs11/src/node.o
SOLINK_MODULE(target) Release/obj.target/pkcs11.node
COPY Release/pkcs11.node
make: Leaving directory '/home/mark/fabric-samples/invoice/node_modules/pkcs11js/build'
> grpc@1.10.1 install /home/mark/fabric-samples/invoice/node_modules/fabric-client/node_modules/grpc
> node-pre-gyp install --fallback-to-build --library=static_library

[[grpc] Success: "/home/mark/fabric-samples/invoice/node_modules/fabric-client/node_modules/grpc/src/node/extension_binary/node-v57-linux-x64-glibc/g
rpc_node.node" is installed via remote

> grpc@1.18.0 install /home/mark/fabric-samples/invoice/node_modules/grpc
> node-pre-gyp install --fallback-to-build --library=static_library

node-pre-gyp WARN Using request for node-pre-gyp https download
npm WARN invoice@1.0.0 No repository field.

added 424 packages from 347 contributors and audited 1093 packages in 105.877s
found 3 vulnerabilities (1 low, 1 moderate, 1 high)
  run 'npm audit fix' to fix them, or 'npm audit' for details
mark@mark-Aspire-A315-41:~/fabric-samples/invoice$
```

NOTE: If get stuck in this **node-pre-gyp WARN Using request for node-pre-gyp https download** just “**Ctrl + C**”.

- ✦ Type this to terminal
 - # go get github.com/golang/protobuf/proto
 - # go get github.com/hyperledger/fabric/common/attrmgr
 - # go get github.com/pkg/errors
 - # go get github.com/hyperledger/fabric/core/chaincode/lib/cid
- ✦ Go to this Folder **Home/go/src/github.com** and copy folders named **hyperledger pkg** and **golang**
- ✦ Go back to **Fabric-samples** folder and open **Chaincode** folder
- ✦ Right click the directory and select **open in terminal**
- ✦ You need to delete existing folder in order to paste the copied folder . To delete secured folder Type “ **sudo rm -R hyperledger pkg golang** “ and press enter.
- ✦ Paste the file you copied
- ✦ Go to **fabric-sample > supply** folder
- ✦ And Type “**./startFabric.sh**”

you should see this :

```
mark@mark-Aspire-A315-41: ~/fabric-samples/invoice
File Edit View Search Terminal Help
# wait for Hyperledger Fabric to start
# incase of errors when running later commands, issue export FABRIC_START_TIMEOUT=<larger number>
export FABRIC_START_TIMEOUT=10
#echo ${FABRIC_START_TIMEOUT}
sleep ${FABRIC_START_TIMEOUT}

# Create the channel
docker exec -e "CORE_PEER_LOCALMSPID=Org1MSP" -e "CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/users/Admin@org1.example.com/msp" peer0.org1.example.co
m peer channel create -o orderer.example.com:7050 -c mychannel -f /etc/hyperledger/configtx/channel.tx
2019-02-08 06:09:47.362 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2019-02-08 06:09:47.811 UTC [cli.common] readBlock -> INFO 002 Received block: 0
# Join peer0.org1.example.com to the channel.
docker exec -e "CORE_PEER_LOCALMSPID=Org1MSP" -e "CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/users/Admin@org1.example.com/msp" peer0.org1.example.co
m peer channel join -b mychannel.block
2019-02-08 06:09:48.190 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2019-02-08 06:09:51.263 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
Creating cli ...
Creating cli ... done
2019-02-08 06:09:58.015 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default esc
2019-02-08 06:09:58.015 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vsc
2019-02-08 06:10:02.000 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
2019-02-08 06:10:02.391 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default esc
2019-02-08 06:10:02.391 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vsc
2019-02-08 06:10:16.960 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200

Total setup execution time : 66 secs ...

Start by installing required packages run 'npm install'
Then run 'node enrollAdmin.js', then 'node registerUser'

The 'node invoke.js' will fail until it has been updated with valid arguments
The 'node query.js' may be run at anytime once the user has been registered

mark@mark-Aspire-A315-41:~/fabric-samples/invoice$
```

✦ Sometimes some of function are not running to fix that, you need to update your chaincode by typing this:

- ***docker exec -it cli bash***
- ***peer chaincode install -n supply -v 1.1 -l "golang" -p "github.com/supply/go"***
- ***peer chaincode upgrade -n supply -v 1.1 -o orderer.example.com:7050 -C mychannel -l "golang" -p "github.com/supply/go" -c '{"Args":[""]}' -P "OR ('Org1MSP.member','Org2MSP.member')"***
- ***exit***

✦ Then type

```
' node enrollAdmin.sh'  
' node registerSupplier.sh'  
' node registerOem.sh '  
' node registerBank.sh'
```

✦ To run the program type **' node app.js**

To push or add data

✦ Open **POSTMAN**.

✦ **NOTE:** If you don't have postman, to download type in the terminal **"snap install postman"**

✦ Select **"POST"** as a method then add url **"localhost:3000/invoice"**

✦ Switch to **header** Tab and Add **'user'** key with a value of **'supplier'**
(Note : Supplier only has a privilege to raise or add Invoice)

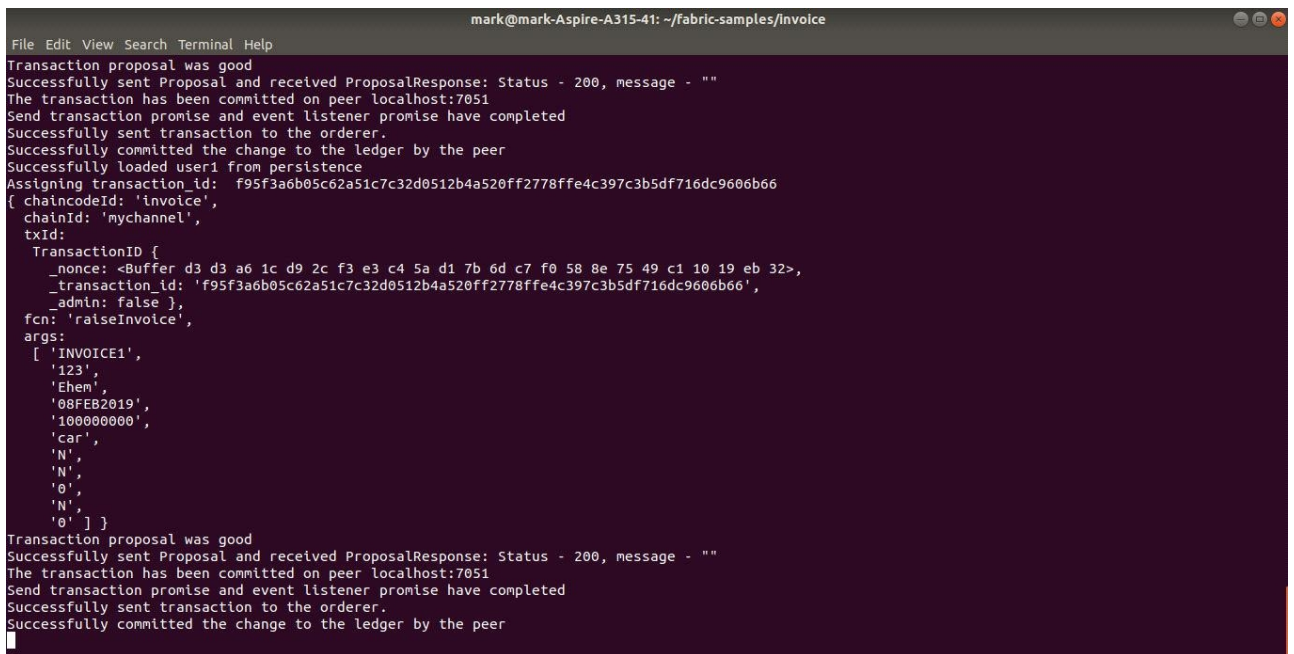
✦ **Switch to the Body Tab and select x-www-form-urlencoded and add this parameters as a key:**

```
invoicenumber:INVOICE6  
billedto:OEM  
invoicedate:02/08/19  
invoiceamount:10000
```

itemdescription:KEYBOARD
goodreceived:False
ispaid:False
paidamount:0
repaid:False
repaymentamount:0

Click the **Send** button.

You should see this:

A screenshot of a terminal window with a dark background and light-colored text. The window title is 'mark@mark-Aspire-A315-41: ~/fabric-samples/invoice'. The terminal output shows a series of messages indicating a successful transaction: 'Transaction proposal was good', 'Successfully sent Proposal and received ProposalResponse: Status - 200, message - ""', 'The transaction has been committed on peer localhost:7051', 'Send transaction promise and event listener promise have completed', 'Successfully sent transaction to the orderer.', 'Successfully committed the change to the ledger by the peer', 'Successfully loaded user1 from persistence', 'Assigning transaction id: f95f3a6b05c62a51c7c32d0512b4a520ff2778ffe4c397c3b5df716dc9606b66', and a JSON object representing the transaction details. The JSON object includes 'chainId: 'mychannel'', 'txId:', 'TransactionID' with a nonce, transaction_id, and admin status, 'fcn: 'raiseInvoice'', and 'args' containing an array of values for an invoice update.

```
mark@mark-Aspire-A315-41: ~/fabric-samples/invoice
File Edit View Search Terminal Help
Transaction proposal was good
Successfully sent Proposal and received ProposalResponse: Status - 200, message - ""
The transaction has been committed on peer localhost:7051
Send transaction promise and event listener promise have completed
Successfully sent transaction to the orderer.
Successfully committed the change to the ledger by the peer
Successfully loaded user1 from persistence
Assigning transaction id: f95f3a6b05c62a51c7c32d0512b4a520ff2778ffe4c397c3b5df716dc9606b66
{ chaincodeId: 'invoice',
  chainId: 'mychannel',
  txId:
    TransactionID {
      _nonce: <Buffer d3 d3 a6 1c d9 2c f3 e3 c4 5a d1 7b 6d c7 f0 58 8e 75 49 c1 10 19 eb 32>,
      _transaction_id: 'f95f3a6b05c62a51c7c32d0512b4a520ff2778ffe4c397c3b5df716dc9606b66',
      _admin: false },
  fcn: 'raiseInvoice',
  args:
    [ 'INVOICE1',
      '123',
      'Ehem',
      '08FEB2019',
      '100000000',
      'car',
      'N',
      'N',
      '0',
      'N',
      '0' ] }
Transaction proposal was good
Successfully sent Proposal and received ProposalResponse: Status - 200, message - ""
The transaction has been committed on peer localhost:7051
Send transaction promise and event listener promise have completed
Successfully sent transaction to the orderer.
Successfully committed the change to the ledger by the peer
```

Updating a GoodReceived

(Note : OEM only has a privilege to update or change GoodReceived)

- ✦ Select **“PUT”** as a method then add url **“localhost:3000/invoice”**
- ✦ Switch to **header** Tab and Add **‘user’** key with a value of **‘oem’**
- ✦ Switch to the **Body** Tab and select **x-www-form-urlencoded** and uncheck all keys except the **invoicenum** and **GoodReceived** .
- ✦ Set value of **GoodReceived** to **True** and **invoicenum** to **INVOICE6** (you can select the voice number you want)
- ✦ Then click **“Send”** button.

Bank transfer payment to Supplier

(Note : Bank only has a privilege to transfer payment to supplier)

- ✦ Select **“PUT”** as a method then add url **“localhost:3000/invoice”**
- ✦ Switch to **header** Tab and Add **‘user’** key with a value of **‘bank’**
- ✦ Switch to the **Body** Tab and select **x-www-form-urlencoded** and uncheck all keys except the **invoicenum** and **paidamount** .
- ✦ Set value of **paidamount** to **10000** and **invoicenum** to **INVOICE6** (you can select the voice number you want)
- ✦ Then click **“Send”** button.
- ✦ It will automatically change the **isPaid** value to **True**

OEM transfer payment to Bank

(Note : OEM only has a privilege to transfer payment to Bank)

- ✦ Select **“PUT”** as a method then add url **“localhost:3000/invoice”**
- ✦ Switch to **header** Tab and Add **‘user’** key with a value of **‘oem’**
- ✦ Switch to the **Body** Tab and select **x-www-form-urlencoded** and uncheck all keys except the **invoicenum** and **repaid** .
- ✦ Set value of **repaymentamount** to greater than **paidamount** (i.e. **11000**) and **invoicenum** to **INVOICE6** (you can select the voice number you want)
- ✦ Then click **“Send”** button.
- ✦ It will automatically change the **repaid** value to **True**

To view the data inside

- ✦ Select **“GET”** then type **“localhost:3000/”** and click **“Send”** it will return all data.
- ✦ Switch to the **Body** Tab and select **x-www-form-urlencoded** and uncheck all keys.
- ✦ Click **Send**.

REFERENCE:

<https://hyperledger.github.io/composer/v0.19/installing/development-tools.html>

<https://hyperledger.github.io/composer/v0.19/installing/installing-prereqs.html#ubuntu>

