

Resume-chain install instruction

This project implements a data-sharing application between Ethereum and Hyperledger Fabric. It can also be regarded as the interoperability between permissionless and permissioned blockchains.

The instruction will let you get a copy of the project Resume-chain and run the program on your local machine for development and testing purposes. The instruction is based on Ubuntu 22.04.1 Desktop (64-bit) operating system. We recommend that you run the program on that operating system.

1. Prerequisites Hyperledger Fabric 2.0

The following prerequisites have to be installed to run Hyperledger Fabric 2.0 locally on your machine:

(1) Install docker

Input the following commands in terminal.

```
$sudo apt-get install docker.io
```

```
ubuntu@ubuntu-machine: $ sudo apt-get install docker.io
[sudo] password for ubuntu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Start the docker service and make it autorun when opening the operating system.

```
$sudo systemctl start docker
$sudo systemctl enable docker
```

Add user to the affiliate group.

```
$sudo usermod -a -G docker $USER
```

Install docker-compose.

```
$sudo apt-get install docker-compose
```

```
ubuntu@ubuntu-Machine: $ sudo systemctl start docker
ubuntu@ubuntu-machine: $ sudo systemctl enable docker
ubuntu@ubuntu-machine: $ sudo usermod -a -G docker $USER
ubuntu@ubuntu-machine: $ sudo apt-get install docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python3-attr python3-docker python3-dockerpty python3-docopt python3-dotenv
  python3-jsonschema python3-persistent python3-texttable python3-websocket
Suggested packages:
  python3-attr-doc python3-jsonschema-doc
The following NEW packages will be installed:
  docker-compose python3-attr python3-docker python3-dockerpty python3-docopt
  python3-dotenv python3-jsonschema python3-persistent python3-texttable
  python3-websocket
0 upgraded, 10 newly installed, 0 to remove and 23 not upgraded.
```

(2) Install Hyperledger Fabric 2.0 binaries

```
$curl -sSL https://bit.ly/2ysbOFE | bash -s -- 2.0.0
```

```
ubuntu@ubuntu-machine: $ curl -sSL https://bit.ly/2ysbOFE| bash -s -- 2.0.0
Clone hyperledger/fabric-samples repo
====> Cloning hyperledger/fabric-samples repo
Cloning into 'fabric-samples'...
remote: Enumerating objects: 11487, done.
remote: Counting objects: 100% (83/83), done.
remote: Compressing objects: 100% (54/54), done.
remote: Total 11487 (delta 35), reused 56 (delta 23), pack-reused 11404
Receiving objects: 100% (11487/11487), 21.80 MiB | 882.00 KiB/s, done.
Resolving deltas: 100% (6151/6151), done.
====> Checking out v2.0.0 of hyperledger/fabric-samples

Pull Hyperledger Fabric binaries
====> Downloading version 2.0.0 platform specific fabric binaries
====> Downloading: https://github.com/hyperledger/fabric/releases/download/v2.0.0/hyperledger-fabric-linux-amd64-2.0.0.tar.gz
  % Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total Spent   Left  Speed
  0     0     0     0     0      0      0 ---:--- ---:--- ---:--- 0
100  72.7M 100  72.7M  0     0  5620k      0  0:00:13  0:00:13 ---:--- 6029k
==> Done.
====> Downloading version 1.5.5 platform specific fabric-ca-client binary
====> Downloading: https://github.com/hyperledger/fabric-ca/releases/download/v1.5.5/hyperledger-fabric-ca-linux-amd64-1.5.5.tar.gz
```

If curl is uninstalled, you can input one of these commands to install.

```
$sudo apt install curl
```

or

```
$sudo snap install curl
```

```
ubuntu@ubuntu-machine: $ curl -sSL https://bit.ly/2ysbOFE| bash -s -- 2.0.0
Command 'curl' not found, but can be installed with:
sudo snap install curl # version 7.86.0, or
sudo apt install curl # version 7.81.0-1ubuntu1.6
See 'snap info curl' for additional versions.
```

(3) Install npm and node.js

```
$curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
$sudo apt-get install -y nodejs
$sudo apt-get install build-essential
```

```
ubuntu@ubuntu-machine: $ curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
## Installing the NodeSource Node.js 14.x repo...

## Populating apt-get cache...
+ apt-get update
Hit:1 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:2 https://packages.microsoft.com/repos/edge stable InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://hk.archive.ubuntu.com/ubuntu jammy InRelease
Hit:5 http://hk.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:6 http://hk.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
W: Target Packages (main/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list.d/google-chrome.list:3 and /etc/apt/sources.list.d/google.list:1
W: Target Packages (main/binary-all/Packages) is configured multiple times in /etc/apt/sources.list.d/google-chrome.list:3 and /etc/apt/sources.list.d/google.list:1
W: Target Translations (main/i18n/Translation-en_US) is configured multiple times in /etc/apt/sources.list.d/google-chrome.list:3 and /etc/apt/sources.list.d/google.list:1
W: Target Translations (main/i18n/Translation-en) is configured multiple times in /etc/apt/sources.list.d/google-chrome.list:3 and /etc/apt/sources.list.d/google.list:1
W: Target DEP-11 (main/dep11/Components-amd64.yml) is configured multiple times in /etc/apt/sources.list.d/google-chrome.list:3 and /etc/apt/sources.list.d/google.list:1
W: Target DEP-11 (main/dep11/Components-all.yml) is configured multiple times in /etc/apt/sources.list.d/google-chrome.list:3 and /etc/apt/sources.list.d/google.list:1
```

```
ubuntu@ubuntu-machine:~$ sudo apt-get install -y nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdii-2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 23 not upgraded.
Need to get 25.7 MB of archives.
After this operation, 125 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_14.x jammy/main amd64 nodejs amd64 14.21.1-deb-1nodesource1 [25.7 MB]
Fetched 25.7 MB in 4s (6,121 kB/s)
Selecting previously unselected package nodejs.
(Reading database ... 213247 files and directories currently installed.)
Preparing to unpack .../nodejs_14.21.1-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (14.21.1-deb-1nodesource1) ...
Setting up nodejs (14.21.1-deb-1nodesource1) ...
Processing triggers for man-db (2.10.2-1) ...
ubuntu@ubuntu-machine:~$ sudo apt-get install build-essential
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.9ubuntu3).
```

(4) Clone this repository

```
$git clone https://github.com/jaysonli765/Resume-chain.git
```

```
ubuntu@ubuntu-machine:~$ git clone https://github.com/jaysonli765/Resume-chain
Cloning into 'Resume-chain'...
remote: Enumerating objects: 684, done.
remote: Total 684 (delta 0), reused 0 (delta 0), pack-reused 684
Receiving objects: 100% (684/684), 97.84 MiB | 3.38 MiB/s, done.
Resolving deltas: 100% (156/156), done.
```

Install required node dependencies from the package.json file in the folder "website".

```
$cd Resume-chain/website
$npm install
```

```
ubuntu@ubuntu-machine:~/Resume-chain/website$ npm install
> node@10.21.0 preinstall /home/ubuntu/Resume-chain/website/node_modules/node
> node installArchSpecificPackage

+ node-linux-x64@10.21.0
added 1 package in 1.74s
found 0 vulnerabilities

> pkcs11js@1.3.1 install /home/ubuntu/Resume-chain/website/node_modules/pkcs11js
> node-gyp rebuild
make: Entering directory '/home/ubuntu/Resume-chain/website/node_modules/pkcs11js/build'
  CXX(target) Release/obj.target/pkcs11/src/main.o
```

Upgrade the required packages.

```
$npm audit fix
```

```
ubuntu@ubuntu-machine:~/Resume-chain/website$ npm audit fix
npm WARN website@1.0.0 No description
npm WARN website@1.0.0 No repository field.

removed 1 package and updated 1 package in 1.8s

33 packages are looking for funding
  run `npm fund` for details

fixed 10 of 23 vulnerabilities in 511 scanned packages
  13 vulnerabilities required manual review and could not be updated
```

2. Prerequisites Ethereum

Ganache is software which can generate local Ethereum on your machine, and you can connect the local Ethereum with the program and test the program.

The following prerequisites have to be installed to run Ganache locally on your machine:

(1) Install truffle

```
$sudo npm install -g truffle
```

```
ubuntu@ubuntu-machine:~$ sudo npm install -g truffle
[sudo] password for ubuntu:
npm WARN deprecated apollo-server@3.11.1: The `apollo-server` package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the `@apollo/server` package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated uid@2.0.1: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
```

(2) Install Ganache

Input the following command in terminal to download Ganache v2.5.4.

```
$wget https://github.com/trufflesuite/ganache-ui/releases/download/v2.5.4/ganache-2.5.4-linux-x86_64.AppImage
```

```
ubuntu@ubuntu-machine:~$ wget https://github.com/trufflesuite/ganache-ui/releases/download/v2.5.4/ganache-2.5.4-linux-x86_64.AppImage
--2022-12-06 18:01:09-- https://github.com/trufflesuite/ganache-ui/releases/download/v2.5.4/ganache-2.5.4-linux-x86_64.AppImage
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/79269625/8e2f9000-1889-11eb-84dd-6abfbbae2deb0?X-Amz-Algoritm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWJYAX4CSVEHS3A%2F20221206%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20221206T001009Z&X-Amz-Expires=3600&X-Amz-Signature=2eb507ebc302913c3dff4d10ab7066ed2784a040dea9dccef1729be0d991b5c&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=79269625&response-content-disposition=attachment%3B%20filename%3Dganache-2.5.4-linux-x86_64.AppImage&response-content-type=application%2Foctet-stream [following]
```

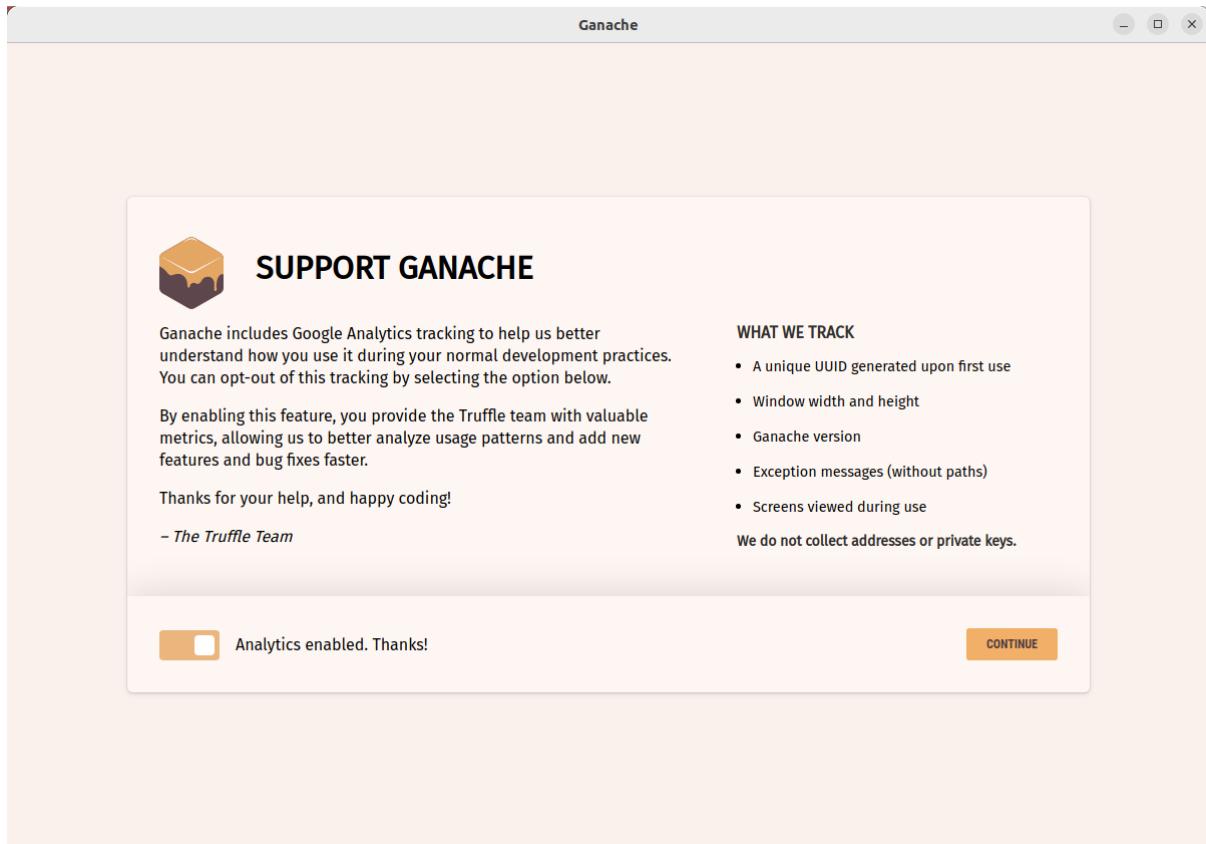
Modify permission of the file.

```
$chmod +x ganache-2.5.4-linux-x86_64.AppImage
```

Input the following command to run Ganache.

```
$sudo ./ganache-2.5.4-linux-x86_64.AppImage
```

Or right-click “ganache-2.5.4-linux-x86_64.AppImage” on the folder and click “Run”.



If libfuse2 is uninstalled, you can input the following command to install.

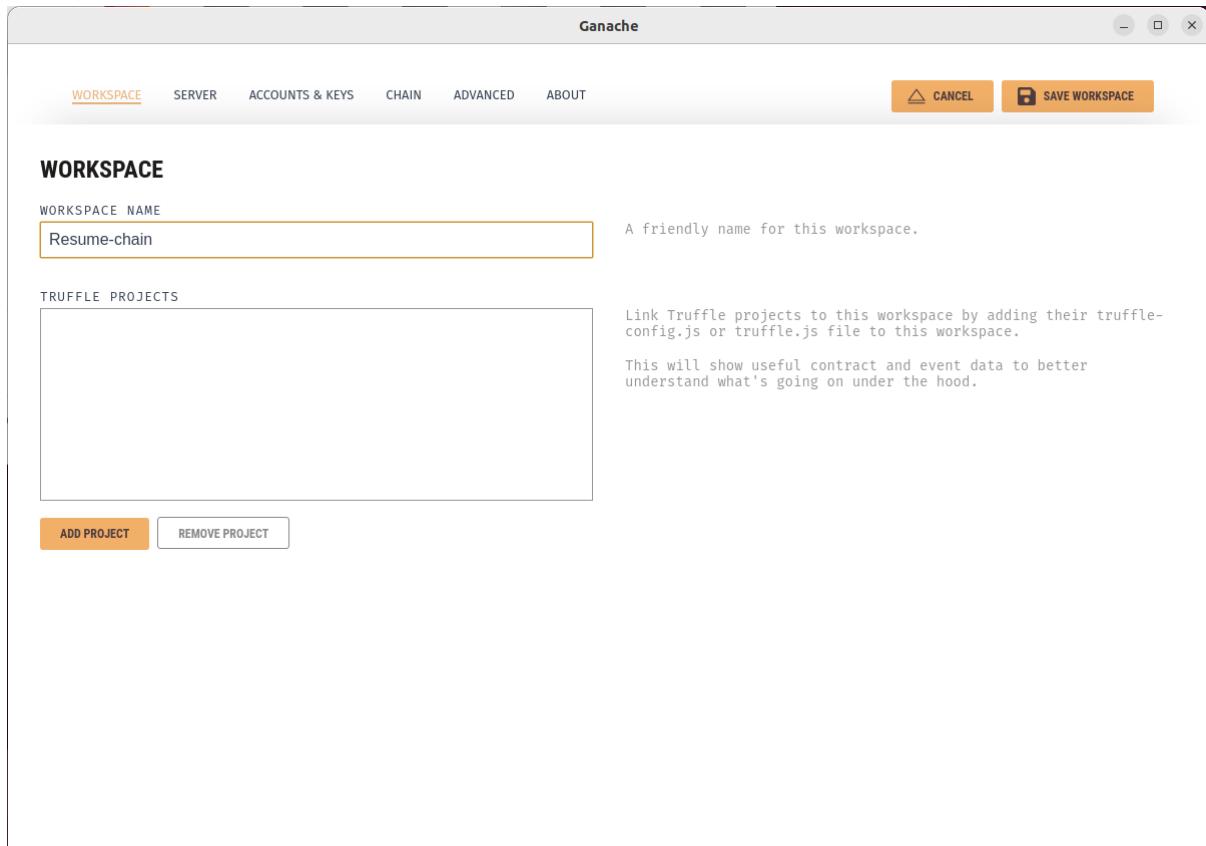
```
$sudo apt install libfuse2
```

```
ubuntu@ubuntu-machine: $ sudo ./ganache-2.5.4-linux-x86_64.AppImage
dlopen(): error loading libfuse.so.2

AppImages require FUSE to run.
You might still be able to extract the contents of this AppImage
if you run it with the --appimage-extract option.
See https://github.com/AppImage/AppImageKit/wiki/FUSE
for more information
ubuntu@ubuntu-machine: $ sudo apt install libfuse2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  libfuse2
0 upgraded, 1 newly installed, 0 to remove and 23 not upgraded.
```

(3) Config Ganache

Click “NEW WORKSPACE”, and enter a name for “WORKSPACE NAME”. For example, we input “Resume-chain” as the workspace name.



Click “ADD PROJECT” and add the path of the truffle-config.js file. For Resume-chain project, the path is “./Resume-chain/eth/truffle/truffle-config.js”. Then click “SAVE WORKSPACE”. You will see the following screen.

Ganache						
ACCOUNTS	BLOCKS	TRANSACTIONS	CONTRACTS	EVENTS	LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 0	GAS PRICE 2000000000	GAS LIMIT 6721975	HARDFORK MUIRGLAICER	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING
MNEMONIC <small>?</small> knock security yellow shuffle indoor ugly submit size client limit motion tongue						HD PATH m/44'/60'/0'/0/account_index
ADDRESS 0x24174Df5Ca74b5CA6D339856c2517ec71425F6a7	BALANCE 100.00 ETH			TX COUNT 0	INDEX 0	
ADDRESS 0x01e4F5B3147Ec873a1D60CadaccDDe398c786421	BALANCE 100.00 ETH			TX COUNT 0	INDEX 1	
ADDRESS 0x763a8D5A9d968564696773306FBc2a2343d244F1	BALANCE 100.00 ETH			TX COUNT 0	INDEX 2	
ADDRESS 0xdef5eE5e8caAd0E55a9fAE1E791e7c9F257633bd	BALANCE 100.00 ETH			TX COUNT 0	INDEX 3	
ADDRESS 0xF2F4729D150fe848125EC7df8be9a945ee1C2532	BALANCE 100.00 ETH			TX COUNT 0	INDEX 4	
ADDRESS 0xF5c6A65Aa99d2DA7db5D5Dfb9345d30c3f606Da2	BALANCE 100.00 ETH			TX COUNT 0	INDEX 5	
ADDRESS 0xC0fFcac6C2d794b1e88adF9347E1919B9e7C31E4	BALANCE 100.00 ETH			TX COUNT 0	INDEX 6	
ADDRESS	BALANCE			TX COUNT	INDEX	

Enter the truffle folder, and migrate the contract to the blockchain created by Ganache.

```
$cd Resume-chain/eth/truffle
$truffle migrate
```

```

ubuntu@ubuntu-machine:~/Resume-chain/eth/truffle
ubuntu@ubuntu-machine:~/Resume-chain/eth/truffle$ truffle migrate

Compiling your contracts...
=====
> Compiling ./contracts/Resume.sol
> Artifacts written to /home/ubuntu/Resume-chain/eth/truffle/build/contracts
> Compiled successfully using:
  - solc: 0.8.17+commit.8df45f5f.Emscripten.clang
  Fetching solc version list from solc-bin. Attempt #1
✓ Downloading compiler. Attempt #1.

Starting migrations...
=====
> Network name: 'development'
> Network id: 5777
> Block gas limit: 6721975 (0x6691b7)

1_Migration_CV.js
=====
.. Fetching solc version list from solc-bin. Attempt #1
Replacing 'Resume'.r. Attempt #1.

-----
> transaction hash: 0xbF2e5869a5bf314b36d153b50663b9435b90b793d8e8476f10469ced86361875
> Blocks: 0 Seconds: 0
> contract address: 0x4f521aa5Ddfc76aA1c41590Dd0133Db7BBa62a67
> block number: 1
> block timestamp: 1670328187
> account: 0x24174DF5Ca74b5CA6D339856c2517ec71425F6a7
> balance: 99.98562002
> gas used: 718999 (0xaf897)
> gas price: 20 gwei
> value sent: 0 ETH
> total cost: 0.01437998 ETH

> Saving artifacts
-----
> Total cost: 0.01437998 ETH

```

(4) Add contract address into program

Click the “CONTRACTS” tab in Ganache, and get the contract address. In this example, the address is “0x4f521aa5Ddfc76aA1c41590Dd0133Db7BBa62a67”.

The screenshot shows the Ganache interface with the following details:

- Top Bar:** Shows the Ganache logo and navigation icons.
- Header:** Includes tabs for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS (which is highlighted in orange), EVENTS, and LOGS. A search bar is also present.
- Current Block:** CURRENT BLOCK 1, GAS PRICE 20000000000, GAS LIMIT 6721975, HARDFORK MUIRGLAGIER, NETWORK ID 5777, RPC SERVER HTTP://127.0.0.1:7545, MINING STATUS AUTOMINING.
- Workspace:** WORKSPACE RESUME-CHAIN, SWITCH, and a settings gear icon.
- Contracts Tab Content:**
 - Resume Contract:** ADDRESS 0x4f521aa5Ddfc76aA1c41590Dd0133Db7BBa62a67, BALANCE 0.00 ETH.
 - Creation TX:** 0xbF2e5869a5BF314b36d153b50663b9435b90b793d8e8476f10469CED86361875.
 - Storage:** Shows a JSON object: { resumeMap : {} }.
 - Transactions:** NO TRANSACTIONS.
 - Events:** NO EVENTS.

Enter “./Resume-chain/website”, and open the file “employee.html”. Replace the contract address with your address in line 178. Then open the file “employer.html” and repeat this operation in line 179.

3. Start the Hyperledger Fabric network and website

Input the following commands to start the program.

```
$cd Resume-chain  
$sudo ./start_Network.sh
```

```
ubuntu@ubuntu-machine:~/Resume-chain$ sudo ./start_Network.sh
Stopping for channel 'mychannel' with CLI timeout of '10' seconds and CLI delay of '3' seconds
proceeding ...
WARNING: The BYFN_CA1_PRIVATE_KEY variable is not set. Defaulting to a blank string.
WARNING: The BYFN_CA2_PRIVATE_KEY variable is not set. Defaulting to a blank string.
```

The program has been started successfully.

Open (<http://localhost:8000/employee>) and (<http://localhost:8000/employer>) in the browser. We can see the following two website pages.

The employee page:

Find Your Candidate Upload Your Data localhost:8000/employee

Upload Your Data

Name	<input type="text"/>
Phone number	<input type="text"/>
Education	<input type="text"/>
Working experience	<input type="text"/>

Query data from Ethereum

Type in name of CV

The employer page:

Find Your Candidate Upload Your Data localhost:8000/employer

Find Your Candidate

Fetch Candidate Information from Ethereum to Hyperledger

Type in name of CV

Name	<input type="text"/>
Phone number	<input type="text"/>
Education	<input type="text"/>
Working experience	<input type="text"/>

Type in name of CV