

1. Fresh install of **Ubuntu 16.04** and download all the updates before proceeding to install pre-requisites for Hyperledger Fabric which are:
 1. Install cURL.
 2. Install Node.js and npm package manager.
 3. Install Go Language.
 4. Install Docker and Docker Compose.
-

2. Installing cURL

On most Ubuntu/Linux system curl is already installed but if it is not then follow these steps:

Open a terminal window: **CTRL+ALT+T**.

Type the following command and enter your password:

\$ sudo apt install curl

To check, run the following command in your terminal/command line:

\$ curl -V

Note: The "V" is capitalized.

3. **Install docker from Ubuntu software (community edition- CE)** or from <https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/#install-docker-ce-1>

1. Manage Docker as a Non-Root User

If you don't want to use **sudo** when you use the **docker** command, create a Unix group called **docker** and add users to it. When the **docker** daemon starts, it makes the ownership of the Unix socket read/writable by the **docker** group.

Warning: The **docker** group grants privileges equivalent to the **root** user. For details on how this impacts security in your system, see [Docker Daemon Attack Surface](#).

To create the **docker** group and add your user:

- 1. Create the docker group:

\$ sudo groupadd docker

2. Add your user to the **docker** group:

\$ sudo usermod -aG docker \$USER

3. Restart computer and log back in, so that your group membership is re-evaluated.

4. On a desktop Linux environment such as X Windows, log out of your session completely and then log back in.

5. Verify that you can run Docker commands without **sudo**.

\$ docker run hello-world

6. This command downloads a test image and runs it in a container. When the container runs, it prints an informational message and exits.

2. Docker Compose

To install Docker Compose, run the following commands in your terminal/command line:

\$ sudo apt update

\$ sudo apt install docker-compose

Check to make sure that you have Docker version 17.03.1- or greater, and Docker Compose version 1.9.0 or greater:

\$ docker --version && docker-compose --version

4. Installing Node.js and npm

Install npm first and nodejs later-- else there are problems.

\$ sudo apt install npm

To install **Node.js**, run the following commands in your terminal/command line:

\$ sudo bash -c "cat >/etc/apt/sources.list.d/nodesource.list" <<EOL

deb https://deb.nodesource.com/node_6.x xenial main

deb-src https://deb.nodesource.com/node_6.x xenial main

EOL

\$ curl -s https://deb.nodesource.com/gpgkey/nodesource.gpg.key | sudo apt-key add -

\$ sudo apt update

\$ sudo apt install nodejs

Verify the installation, as well as the versions of both **Node.js** and **npm**, and make sure the **Node.js** version you are installing is greater than v6.9 (do not use v7), and the **npm** version is greater than 3.x:

```
$ node --version && npm --version
```

5. Installing Go Language

Visit <https://golang.org/dl/> and make note of the latest stable release (**v1.8 or later**).

To install Go language, run the following commands in your terminal/command line:

```
$ sudo apt update
```

```
$ sudo curl -O https://storage.googleapis.com/golang/go1.9.2.linux-amd64.tar.gz
```

Note: Switch out the black portion of the URL with the correct filename.

```
$ sudo tar -xvf go1.9.2.linux-amd64.tar.gz
```

```
$ sudo mv go /usr/local
```

```
$ echo 'export PATH=$PATH:/usr/local/go/bin' >> ~/.profile
```

```
$ source ~/.profile
```

Check that the Go version is v1.8 or later:

```
$ go version
```

6. Installing Hyperledger Fabric Docker Images and Binaries

Next, we will download the latest released Docker images for Hyperledger Fabric, and tag them with the **latest** tag. Execute the command from within the directory into which you will extract the platform-specific binaries:

```
$ curl -sSL https://goo.gl/Q3YRTi | bash
```

Note: Check <https://hyperledger-fabric.readthedocs.io/en/latest/samples.html#binaries> for the latest URL (the blue portion in the above **curl** command) to pull in binaries.

This command downloads binaries for **cryptogen**, **configtxgen**, **configtxlator**, **peer** AND downloads the Hyperledger Fabric Docker images. These assets are placed in a **bin** subdirectory of the current working directory.

To confirm and see the list of Docker images you've just downloaded, run:

\$ docker images

The expected response is:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hyperledger/fabric-orderer	x86_64-1.0.3	fd1055ee597a	37 hours ago	151MB
hyperledger/fabric-peer	x86_64-1.0.3	b7f253e87c0c	37 hours ago	154MB
hyperledger/fabric-ccenv	x86_64-1.0.3	2e5898d8b21b	38 hours ago	1.28GB
hyperledger/fabric-tools	latest	ba9750b2565d	4 weeks ago	1.33GB
hyperledger/fabric-tools	x86_64-1.0.2	ba9750b2565d	4 weeks ago	1.33GB
hyperledger/fabric-couchdb	latest	3f922f54bd68	4 weeks ago	1.47GB
hyperledger/fabric-couchdb	x86_64-1.0.2	3f922f54bd68	4 weeks ago	1.47GB
hyperledger/fabric-kafka	latest	0b4b1d249e65	4 weeks ago	1.29GB
hyperledger/fabric-kafka	x86_64-1.0.2	0b4b1d249e65	4 weeks ago	1.29GB
hyperledger/fabric-zookeeper	latest	1efb063147d3	4 weeks ago	1.3GB
hyperledger/fabric-zookeeper	x86_64-1.0.2	1efb063147d3	4 weeks ago	1.3GB
hyperledger/fabric-orderer	latest	6efd17e86e65	4 weeks ago	151MB
hyperledger/fabric-orderer	x86_64-1.0.2	6efd17e86e65	4 weeks ago	151MB
hyperledger/fabric-peer	latest	0e2ed51971c9	4 weeks ago	154MB
hyperledger/fabric-peer	x86_64-1.0.2	0e2ed51971c9	4 weeks ago	154MB
hyperledger/fabric-javaenv	latest	0be45dbd7ff4	4 weeks ago	1.41GB
hyperledger/fabric-javaenv	x86_64-1.0.2	0be45dbd7ff4	4 weeks ago	1.41GB
hyperledger/fabric-ccenv	latest	d0f166e1a89e	4 weeks ago	1.28GB
hyperledger/fabric-ccenv	x86_64-1.0.2	d0f166e1a89e	4 weeks ago	1.28GB
hyperledger/fabric-baseos	x86_64-0.3.2	bbcbb9da2d83	5 weeks ago	129MB

Note the tags for each of the repositories above boxed in red. If the Docker images are not already tagged with the **latest** tag, perform the following command for each of the Docker images:

\$ docker tag hyperledger/fabric-tools:x86_64-1.0.2 hyperledger/fabric-tools:latest

Swap out the **blue** portion with the tags you see in your list of repositories. Also, swap out the **red** portion with the name of the Docker image you are switching the tag for (e.g.: **fabric-tools**, **fabric-ccenv**, **fabric-orderer**, etc.). Repeat this step for all Docker images you see in the list.

In the screenshot above, the Docker images are already tagged. If this is the case for you, you do not need to do this extra step.

As an additional measure, you may want to add the **bin** subdirectory to your **PATH** environment variable, so these can be picked up without needing to qualify the **PATH** to each binary. You can do this by running the following:

```
$ export PATH=$PWD/bin:$PATH
```

To install the Hyperledger Fabric sample code which will be used in the tutorials, do:

```
$ git clone https://github.com/hyperledger/fabric-samples.git
```

```
$ cd fabric-samples/first-network
```

7. Getting Started with Your First Network

Are you ready to get started? Run this command (within the **first-network** folder):

```
$ ./byfn.sh -m generate
```

A brief description will appear, along with a **Y/N** command line prompt. Respond with a **Y** **<Enter>** to continue.

This step generates all of the certificates and keys for all our various network entities, including the genesis block used to bootstrap the ordering service and a collection of configuration transactions required to create a channel.

Next, you can start the network with the following command:

```
$ ./byfn.sh -m up
```

Another command line will appear, reply with **Y** **<Enter>** to continue.

Logs will appear in the command line, showing containers being launched, channels being created and joined, chaincode being installed, instantiated, and invoked on all the peers, as well as various transaction logs.

Troubleshooting Note:

If you have difficulties with the two previous commands and you suspect that your Docker images may be at fault, you can start back from scratch, which will delete and untag the Docker images.

```
$ docker rmi -f $(docker images -q)
```

Once you run this command, return to the *Installing Hyperledger Fabric Docker Images and Binaries* page, at the beginning of this section.

1. Finally, let's test bringing down this network.

Within the same terminal, do **Control+C** to exit the current execution.

Then, run the following command:

```
$ ./byfn.sh -m down
```

Another command line will appear, reply with **Y <Enter>** to continue.

This command will kill your containers, remove the crypto material and four artifacts, and delete the chaincode images from your Docker Registry.

And that's it for a simple demonstration!

These simple steps show how we can easily spin up and bring down a Hyperledger Fabric network, given the code we have. In the next section, we will learn more about chaincode.

Optional Installs

8. **Install Kitematic** – to view docker images and logs from <https://github.com/docker/kitematic/releases/>

9. **Install atom editor**

```
sudo add-apt-repository ppa:webupd8team/atom
```

```
sudo apt update; sudo apt install atom
```