# **Blockcain Setup guide**

To run Hyperledger Composer and Hyperledger Fabric, we recommend you have at least 4Gb of memory.

The following are prerequisites for installing the required development tools:

* 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 and higher 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.

\*\*If installing Hyperledger Composer using Linux, be aware of the following advice:

* Login as a normal user, rather than root.
* Do not su to root.
* When installing prerequisites, use curl, then unzip using sudo.
* Run prereqs-ubuntu.sh as a normal user. It may prompt for root password as some of it's actions are required to be run as root.
* Do not use npm with sudo or su to root to use it.
* Avoid installing node globally as root.\*\*

If you're running on Ubuntu, you can download the prerequisites using the following commands:

curl -O https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh

chmod u+x prereqs-ubuntu.sh

Next run the script - as this briefly uses sudo during its execution, you will be prompted for your password.

./prereqs-ubuntu.sh

# **Installing components**

### **Step 1: Install the CLI tools**

There are a few useful CLI tools for Composer developers. The most important one is composer-cli, which contains all the essential operations, so we'll install that first. Next, we'll also pick up generator-hyperledger-composer, composer-rest-server and Yeoman. Those last 3 are not core parts of the development environment, but they'll be useful if you're following the tutorials or developing applications that interact with your Business Network, so we'll get them installed now.

Note that you **should not** use su or sudo for the following npm commands.

1. Essential CLI tools:

npm install -g composer-cli@0.20

1. Utility for running a REST Server on your machine to expose your business networks as RESTful APIs:

npm install -g composer-rest-server@0.20

1. Useful utility for generating application assets:

npm install -g generator-hyperledger-composer@0.20

1. Yeoman is a tool for generating applications, which utilises generator-hyperledger-composer:

npm install -g yo

### **Step 2: Install Playground**

If you've already tried Composer online, you'll have seen the browser app "Playground". You can run this locally on your development machine too, giving you a UI for viewing and demonstrating your business networks.

1. Browser app for simple editing and testing Business Networks:

npm install -g composer-playground@0.20

### **Step 3: Set up your IDE**

Whilst the browser app can be used to work on your Business Network code, most users will prefer to work in an IDE. Our favourite is VSCode, because a Composer extension is available.

1. Install VSCode from this URL: <https://code.visualstudio.com/download>
2. Open VSCode, go to Extensions, then search for and install the Hyperledger Composer extension from the Marketplace.

### **Step 4: Install Hyperledger Fabric**

This step gives you a local Hyperledger Fabric runtime to deploy your business networks to.

1. In a directory of your choice (we will assume ~/fabric-dev-servers), get the .tar.gz file that contains the tools 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

A zip is also available if you prefer: just replace the .tar.gz file with fabric-dev-servers.zip and the tar -xvf command with a unzip command in the preceding snippet.

1. Use the scripts you just downloaded and extracted to download a local Hyperledger Fabric v1.2 runtime:

cd ~/fabric-dev-servers

export FABRIC\_VERSION=hlfv12

./downloadFabric.sh

Congratulations, you've now installed everything required for the typical Developer Environment. Read on to learn some of the most common things you'll do with this environment to develop and test your Blockchain Business Networks.

# **Controlling your dev environment**

## **Starting and stopping Hyperledger Fabric**

You control your runtime using a set of scripts which you'll find in ~/fabric-dev-servers if you followed the suggested defaults.

The first time you start up a new runtime, you'll need to run the start script, then generate a PeerAdmin card:

cd ~/fabric-dev-servers

export FABRIC\_VERSION=hlfv12

./startFabric.sh

./createPeerAdminCard.sh

You can start and stop your runtime using ~/fabric-dev-servers/stopFabric.sh, and start it again with ~/fabric-dev-servers/startFabric.sh.

At the end of your development session, you run ~/fabric-dev-servers/stopFabric.sh and then ~/fabric-dev-servers/teardownFabric.sh. Note that if you've run the teardown script, the next time you start the runtime, you'll need to create a new PeerAdmin card just like you did on first time startup.

The local runtime is intended to be frequently started, stopped and torn down, for development use. If you're looking for a runtime with more persistent state, you'll want to run one outside of the dev environment, and deploy Business Networks to it. Examples of this include running it via Kubernetes, or on a managed platform such as IBM Cloud.

Also start the OpenLdap Docker images and make sure to enter some persons in it.

## **Start the web app ("Playground")**

To start the web app, run:

composer-playground

It will typically open your browser automatically, at the following address: <http://localhost:8080/login>

You should see the PeerAdmin@hlfv1 Card you created with the createPeerAdminCard script on your "My Business Networks" screen in the web app: if you don't see this, you may not have correctly started up your runtime!

Congratulations, you've got all the components running, and you also know how to stop and tear them down when you're done with your dev session.