Violation Detection Hyperledger Network Specification

# Ubuntu Installation

Latest docs available at <https://hyperledger.github.io/composer/latest/installing/installing-prereqs.html> and https://hyperledger.github.io/composer/latest/installing/development-tools.html

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:/ -servers

export FABRIC\_VERSION=hlfv12

./downloadFabric.sh

## 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.

## 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!

# Generating the Business Network and deploying the ReST server

## Step One: Creating a business network structure

The key concept for Hyperledger Composer is the **business network definition (BND)**. It defines the data model, transaction logic and access control rules for your blockchain solution. To create a BND, we need to create a suitable project structure on disk.

The easiest way to get started is to use the Yeoman generator to create a skeleton business network. This will create a directory containing all of the components of a business network.

1. Create a skeleton business network using Yeoman. This command will require a business network name, description, author name, author email address, license selection and namespace.

yo hyperledger-composer:businessnetwork

1. Enter violation-detection for the network name, and desired information for description, author name, and author email.
2. Select Apache-2.0 as the license.
3. Select composer.tokenshare as the namespace.
4. Select Yes when asked whether to generate an empty network or not

## Step Two: Defining a business network

A business network is made up of assets, participants, transactions, access control rules, and optionally events and queries. Once the business network is generated, relocate the model file (.cto) and the script file (.js) from the Trinetra\_Genesis Github directory into the generated business network under the directories /models and /lib respectively. Access control and query files are to be moved into the parent project folder.

## Step Three: Generate a business network archive

Now that the business network has been defined, it must be packaged into a deployable business network archive (.bna) file.

1. Using the command line, navigate to the violation-detection directory.
2. From the violation-detection directory, run the following command:

composer archive create -t dir -n .

After the command has run, a business network archive file called tutorial-network@0.0.1.bna has been created in the tutorial-network directory.

## Step Four: Deploying the business network

After creating the .bna file, the business network can be deployed to the instance of Hyperledger Fabric. Normally, information from the Fabric administrator is required to create a PeerAdmin identity, with privileges to install chaincode to the peer as well as start chaincode on the composerchannel channel. However, as part of the development environment installation, a PeerAdmin identity has been created already.

After the business network has been installed, the network can be started. For best practice, a new identity should be created to administer the business network after deployment. This identity is referred to as a network admin.

#### Retrieving the correct credentials

A PeerAdmin business network card with the correct credentials is already created as part of development environment installation.

#### Deploying the business network

Deploying a business network to the Hyperledger Fabric requires the Hyperledger Composer business network to be installed on the peer, then the business network can be started, and a new participant, identity, and associated card must be created to be the network administrator. Finally, the network administrator business network card must be imported for use, and the network can then be pinged to check it is responding.

1. To install the business network, from the violation-detection directory, run the following command:

composer network install --card PeerAdmin@hlfv1 --archiveFile violation-detection.bna

The composer network install command requires a PeerAdmin business network card (in this case one has been created and imported in advance), and the the file path of the .bna which defines the business network.

1. To start the business network, run the following command:

composer network start --networkName violation-detection --networkVersion 0.2.6-deploy.29 --networkAdmin admin --networkAdminEnrollSecret adminpw --card PeerAdmin@hlfv1 --file networkadmin.card

The composer network start command requires a business network card, as well as the name of the admin identity for the business network, the name and version of the business network and the name of the file to be created ready to import as a business network card.

1. To import the network administrator identity as a usable business network card, run the following command:

composer card import --file networkadmin.card

The composer card import command requires the filename specified in composer network start to create a card.

1. To check that the business network has been deployed successfully, run the following command to ping the network:

composer network ping --card admin@violation-detection

The composer network ping command requires a business network card to identify the network to ping.

## Step Five: Generating a REST server

Hyperledger Composer can generate a bespoke REST API based on a business network. For developing a web application, the REST API provides a useful layer of language-neutral abstraction.

1. To create the REST API, navigate to the tutorial-network directory and run the following command:

composer-rest-server

1. Enter admin@violation-detection as the card name.
2. Select **never use namespaces** when asked whether to use namespaces in the generated API.
3. Select **No** when asked whether to secure the generated API.
4. Select **Yes** when asked whether to enable event publication.
5. Select **No** when asked whether to enable TLS security.

The generated API is connected to the deployed blockchain and business network.

# Peer specification

Running a “docker ps” command after starting the Fabric network in the previous section, you will see 5 services running as Docker containers. These are:

* **Fabric CA (hyperledger/fabric-ca:1.2.1):** used to create credentials (certificates) for peers belonging to members.
* **Fabric Orderer (hyperledger/fabric-orderer:1.2.1):** A single ordering service node that converts valid transactions into blocks.
* **CouchDB (hyperledger/fabric-couchdb:0.4.10):** Holds current abstracted version of world state
* **Hyperledger Fabric Peer Admin (hyperledger/fabric-peer:1.2.1):** Initializes and oversees the fabric network structure (network definition, creation of channels), and is responsible for registering new member peers into the business network.
* **Chaincode installed member peer (dev-peer0.org1.example.com-violation-detection -0.0.1):** Serves as the node that is used to issue transactions and trigger chaincode in the network.

# Glossary

* **Peer Node:** A network entity that maintains a ledger and runs chaincode containers in order to perform read/write operations to the ledger. Peers are owned and maintained by members.
* **Ordering Service:** A defined collective of nodes that orders transactions into a block. The ordering service exists independent of the peer processes and orders transactions on a first-come-first-serve basis for all channel’s on the network. The ordering service is designed to support pluggable implementations beyond the out-of-the-box SOLO and Kafka. The ordering service is a common binding for the overall network; it contains the cryptographic identity material tied to each member.
* **Endorsement:** Refers to the process where specific peer nodes execute a chaincode transaction and return a proposal response to the client application. The proposal response includes the chaincode execution response message, results (read set and write set), and events, as well as a signature to serve as proof of the peer’s chaincode execution. Chaincode applications have corresponding endorsement policies, in which the endorsing peers are specified.
* **Asset**: An asset can be anything of value. A house is an example of a physical asset, and a mortgage is an example of non-physical asset. Assets within Hyperledger Composer can be defined to encompass any physical or non-physical asset.
* **Business Network Archive**: A business network archive (.bna) is a compressed business network definition which contains at least a business network model and transaction processor functions and may optionally contain an access control file. Business network archives can be deployed to a Hyperledger fabric.
* **Business Network Definition**: A business network definition is made up of the business network model, transaction processor functions and an access control file. The business network definition describes all assets, participants, transactions, and operations for a given solution, and can be interacted with by using a command line interface or an API.
* **Business Network Model**: The business network model describes the assets, participants, and transactions in the business network. The model is in effect the static object structure of the overall business network.
* **Connection Profile**: Connection profiles are .json files used by Hyperledger Composer to connect to an instance of Hyperledger Fabric.
* **Identity**: An identity is a distinct unique identifier associated with a participant. When joining a business network, an identity is issued to a participant which is used by the participant to interact with the business network. Identity documents normally expire after a given length of time, but can be issued or revoked manually. Hyperledger Composer uses Hyperledger Fabric enrollment certificates as identity documents.
* **Participant**: Participants represent the organizations or people who take part in the digital business network. Participants are defined in the business network model.
* **Registry**: Registries are stores of assets held on the blockchain. The contents of the registry are validated using the blockchain consensus mechanism.
* **Transaction**: Transactions are submitted by a participant to affect the assets held in the asset registries on the Hyperledger blockchain. Transactions with a business network are defined in the business network model, and their operations are defined in the transaction processor function file.