

Network Concepts

We will see how we can deploy a basic hyperledger fabric network to enable local development.

WE'LL COVER THE FOLLOWING ^

- Understanding Docker
- Components of Network:
 - PEER:
 - ORDERER:
 - CERTIFICATE AUTHORITY:
 - TOOLING:
- Pre-reqs to network up:

Hyperledger Fabric forms a distributed network and in order to run on a single machine, we use docker containers to run individual distributed components. Each component runs in a separate container instance and connects to other containers to form a network.

Understanding Docker

A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Understanding docker and docker-compose is crucial to understanding the dev environment we will use in this course.

Please visit the following links to get a basic knowledge about docker if you do not already have it:

- [What Is Docker And How It Works?](#)
- [Make a Local Docker Container for php](#)

For development purposes we will run a small network on our machine in

For development purposes we will run a small network on our machine in docker containers. This network configuration is called basic-network and is provided in official hyperledger fabric-samples repo (link: <https://github.com/hyperledger/fabric-samples>).

Components of Network:

Lets get to know the components of a basic network.

PEER:

- **peer container:** Runs the peer node
- **couch db container:** Stores the state database of peer node

ORDERER:

- **orderer container:** Solo orderer node to keep dev environment simple. In a real-world network the ordering service is distributed with multiple nodes communicating with each other.

CERTIFICATE AUTHORITY:

- **fabric-ca container:** root certificate authority for issuing membership certificates to all nodes and users. Since we will use single CA, our network is comprised of a single org called “example dot com”.

TOOLING:

- **fabric-cli container:** This container has some cli tools that help us interact with network nodes to deploy chaincode etc.

Pre-reqs to network up:

Before we can bring up a network we need to do a couple of things:

1. Generate keys and certificates for all infrastructure entities. The key will be given to each entity and the certificate with the public key will be provided to all other entities. Each certificate generated will be signed by common **certificate-authority** or **CA**. There are multiple ways of generating this essential crypto material. To keep things simple for now, we will use pre-generated crypto material **/infra-basic-network/crypto-config**. This material was generated using the **cryptogen** utility provided by the x`xhyperledger fabric. We could also use OpenSSL to generate

this.

2. Generate `genesis block` and `config transaction`. You will do this using `confitxgen` tool.

Now lets deploy our first network.

In the next lesson, we will deploy our network.