Blockchain self-managed implementation on IBM z14 - Infrastructure Deployment Options and Demo

I016752

Guillaume Hoareau
Certified IT Architect

Guillaume Lasmayous
Certified IT Specialist

IBM Systems
Technical Events
ibm.com/training/events

Technical University

Location 2017

Blockchain Deployed

Hyperledger Fabric deployment options

IBM **Blockchain**

V5.0, 23 August 2017

Blockchain Explained

An Introduction to Blockchain for Business

IBM **Blockchain**

Blockchain education series

Explained Solutions Composed Architected Explored Next Steps

V5.0, 23 August 2017

About this session

Blockchain based on Hyperledger Fabric offers a lot of possibilities in term of implementation.

Beyond the IBM Z infrastructure possible topologies (vertical, horizontal, mixed...) each node has an high level possible configuration thanks to side implementation that can bring added value to a Blockchain self-managed implementation.

Speaker will go through possible topologies and will illustrate his arguments thanks to a real demo implementation running on IBM z14.

Wednesday 15:15-16:15 Atlanta

Friday 12:15-13:15 Barcelona

Agenda

Deployment options

Hyperledger Fabric

Custom Applications

App Layer

API libraries and GUIs
Specialized extensions
Specialized consensus algos
Membership policies
Gateway
Operations dashboard

Value Added Systems

Core APIs

Code execution environment
Ledger data structures
Modular consensus framework
Modular identity services
Network peers

Blockchain Fabric for a Permissioned, Distributed and Shared Ledger

Hyperledger Explorer

Transaction, Block and API explorer

Fabric Composer

Asset Management and Application Development

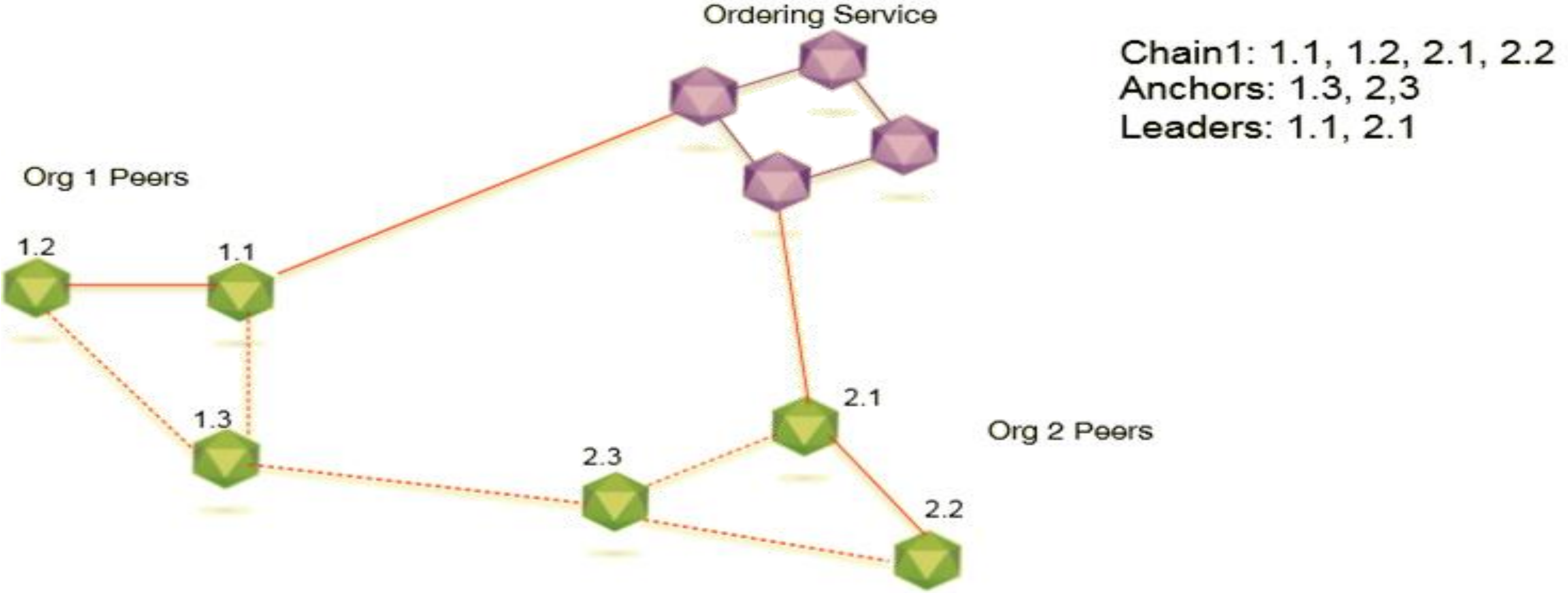
Community + Code

Linux Foundation Hyperledger Project

IBM

Docker Signed Images

Objective: building a business network

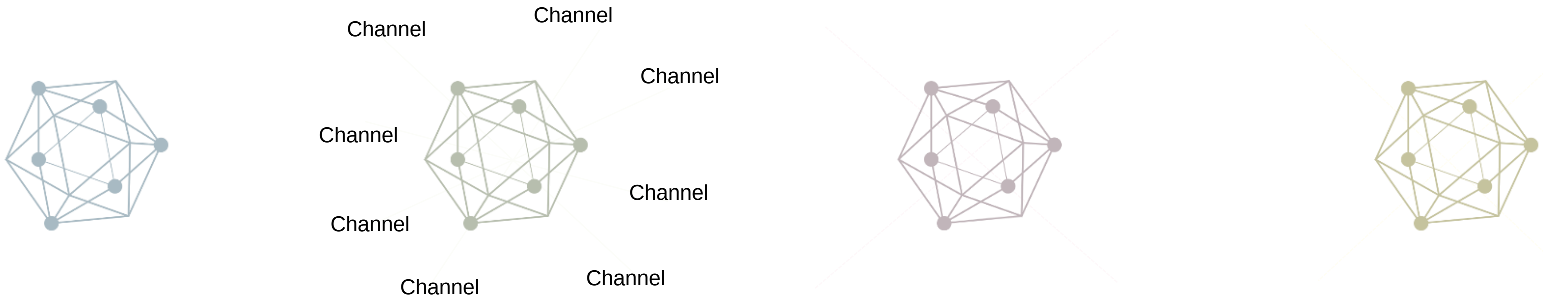


Chain1: 1.1, 1.2, 2.1, 2.2
Anchors: 1.3, 2,3
Leaders: 1.1, 2.1

Infrastructure options

Community + Code
Linux Foundation Hyperledger Project

IBM
Docker Signed Images



Client

Peer

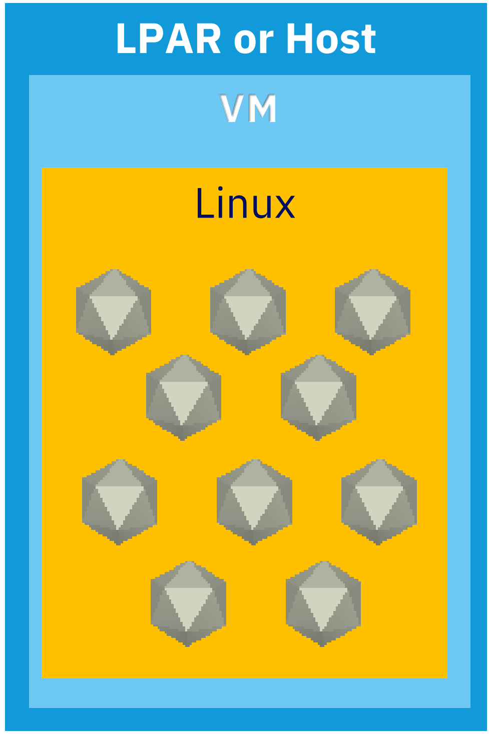
Orderer

CA

Software Requirements	Programming	Go
	Container	docker
Operating System	Linux	SUSE ubuntu redhat
Virtualization Technology	Hypervisor	LPAR KVM SSC z/VM ...

x86_64, ppc64le, IBM Z

Hyperledger Fabric topology options

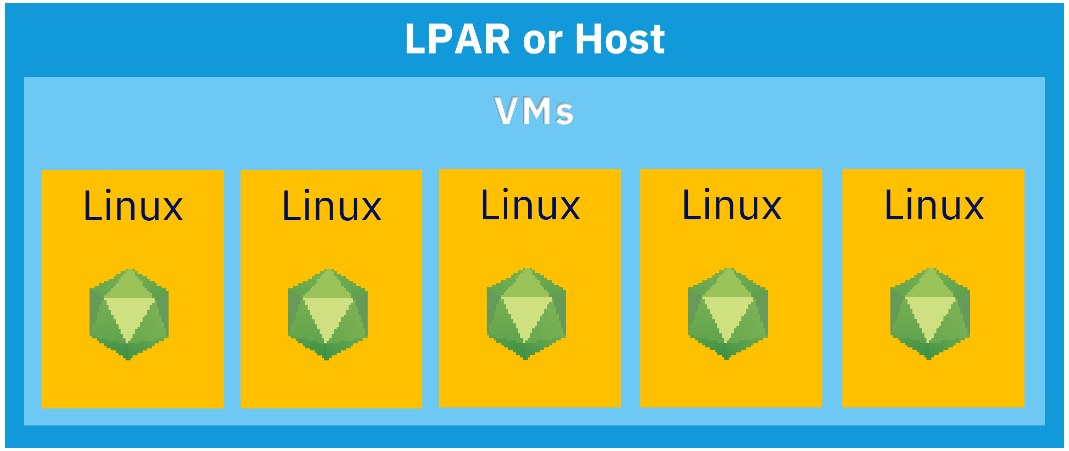


LPAR or Host

VM

Linux

Vertical Peer topology with massive docker containers

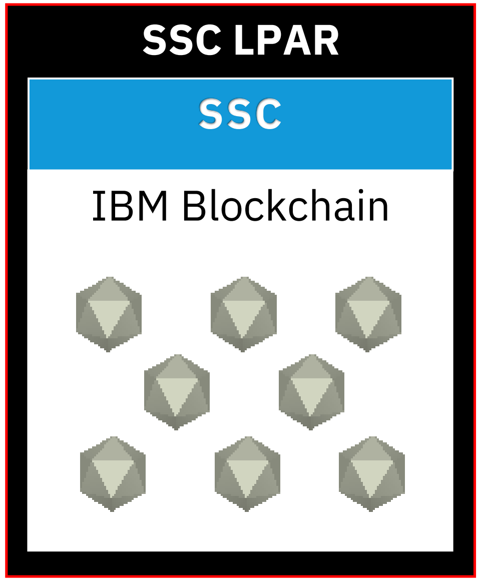


LPAR or Host

VMs

Linux Linux Linux Linux Linux

Horizontal Peer topology with VMs



SSC LPAR

SSC

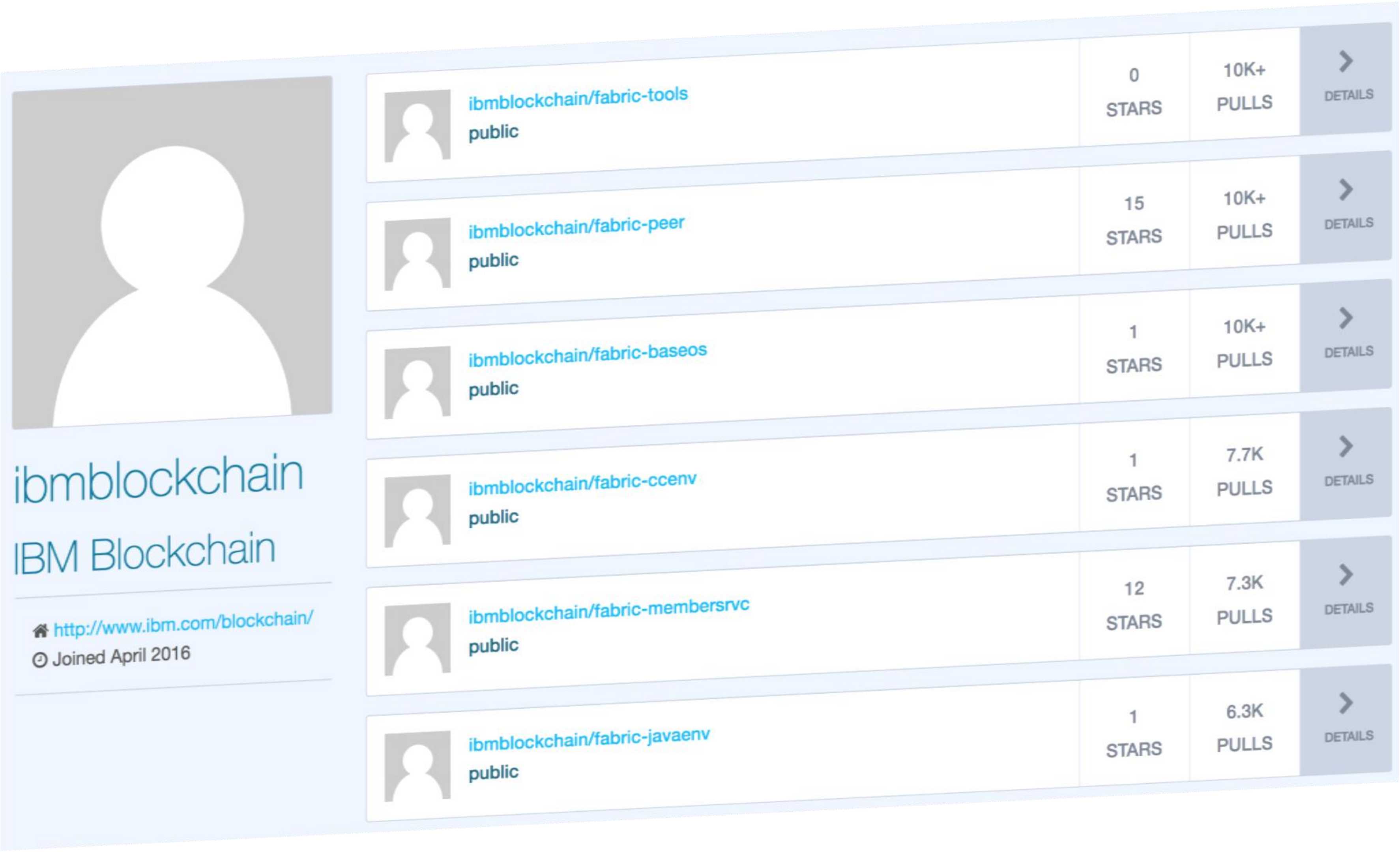
IBM Blockchain

Extreme Isolation and Security with IBM SSC

Docker Peer Linux Peer

A peer is a peer.
100% Compatibility between peers.
Hybrid Topologies are possibles.

Docker images



ibmblockchain

IBM Blockchain

http://www.ibm.com/blockchain/

Joined April 2016

ibmblockchain/fabric-tools public	0 STARS	10K+ PULLS	DETAILS
ibmblockchain/fabric-peer public	15 STARS	10K+ PULLS	DETAILS
ibmblockchain/fabric-baseos public	1 STARS	10K+ PULLS	DETAILS
ibmblockchain/fabric-ccenv public	1 STARS	7.7K PULLS	DETAILS
ibmblockchain/fabric-membersrvc public	12 STARS	7.3K PULLS	DETAILS
ibmblockchain/fabric-javaenv public	1 STARS	6.3K PULLS	DETAILS

Images to be pulled are:

- IBM Signed
- x86_64, ppc64le or s390x Certified
- Tested

Hyperledger Fabric components and options

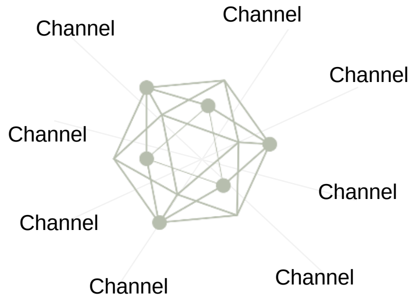
Client

Authenticate users, and propose transactions to peers. Client uses a wallet to store digital identities.

OPTIONS

SDKs:
- NodeJS
- Java
- Go
- Python
- REST
- …

Peer



In charge of running smart contracts and maintaining ledgers.

Peers are connected to Channels.

OPTIONS

State Ledger:
- World state ledger (default golvldb)
- Pluggable worldstate ledger (CouchDB)

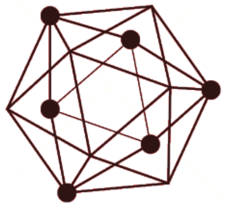
Orderer

In charge of ordering transactions in blocks. It is in charge of running the consensus.

OPTIONS

Consensus:
- SOLO
- KAFKA
- Other alternatives to come (sBFT, PBFT…)

CA



In charge of issuing digital identities across organizations.

OPTIONS

- Key Store (Default SQLite3)
- Key Store (Alternative PostgreSQL
- LDAP Integration
- HA
- …
- Alternative with CryptoGen or existing PKI and CA

Hyperledger Fabric Client application - options

IBM Blockchain

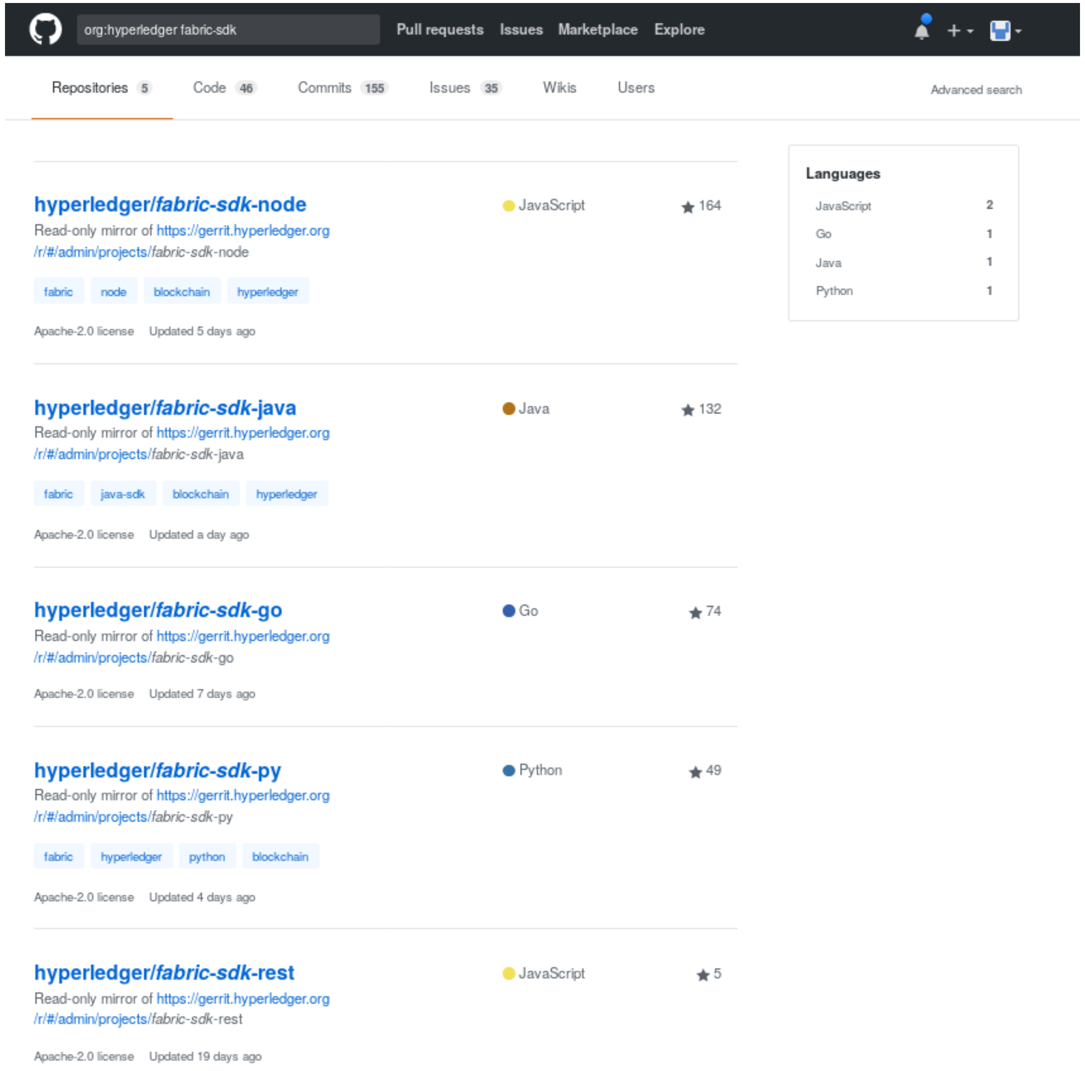
Client

Authenticate users, and propose transactions to peers. Client uses a wallet to store digital identities.

OPTIONS

SDKs:

- NodeJS
- Java
- Go
- Python
- REST
- …



hyperledger/*fabric-sdk*-node ● JavaScript ★ 164

Read-only mirror of https://gerrit.hyperledger.org/r/#/admin/projects/*fabric-sdk*-node

fabric node blockchain hyperledger

Apache-2.0 license Updated 5 days ago

hyperledger/*fabric-sdk*-java ● Java ★ 132

Read-only mirror of https://gerrit.hyperledger.org/r/#/admin/projects/*fabric-sdk*-java

fabric java-sdk blockchain hyperledger

Apache-2.0 license Updated a day ago

hyperledger/*fabric-sdk*-go ● Go ★ 74

Read-only mirror of https://gerrit.hyperledger.org/r/#/admin/projects/*fabric-sdk*-go

Apache-2.0 license Updated 7 days ago

hyperledger/*fabric-sdk*-py ● Python ★ 49

Read-only mirror of https://gerrit.hyperledger.org/r/#/admin/projects/*fabric-sdk*-py

fabric hyperledger python blockchain

Apache-2.0 license Updated 4 days ago

hyperledger/*fabric-sdk*-rest ● JavaScript ★ 5

Read-only mirror of https://gerrit.hyperledger.org/r/#/admin/projects/*fabric-sdk*-rest

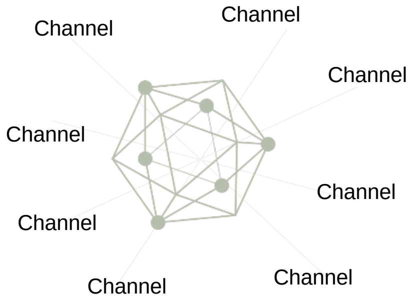
Apache-2.0 license Updated 19 days ago

Languages

JavaScript	2
Go	1
Java	1
Python	1

Hyperledger Fabric Peer – options

Peer



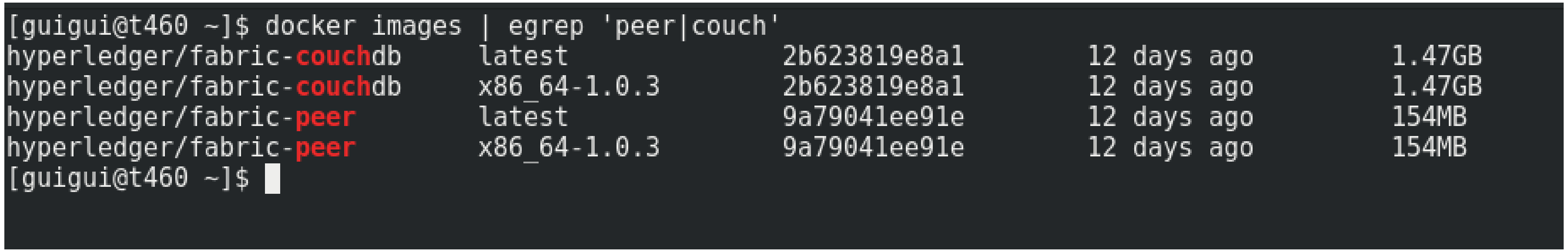
In charge of running smart contracts and maintaining ledgers.

Peers are connected to Channels.

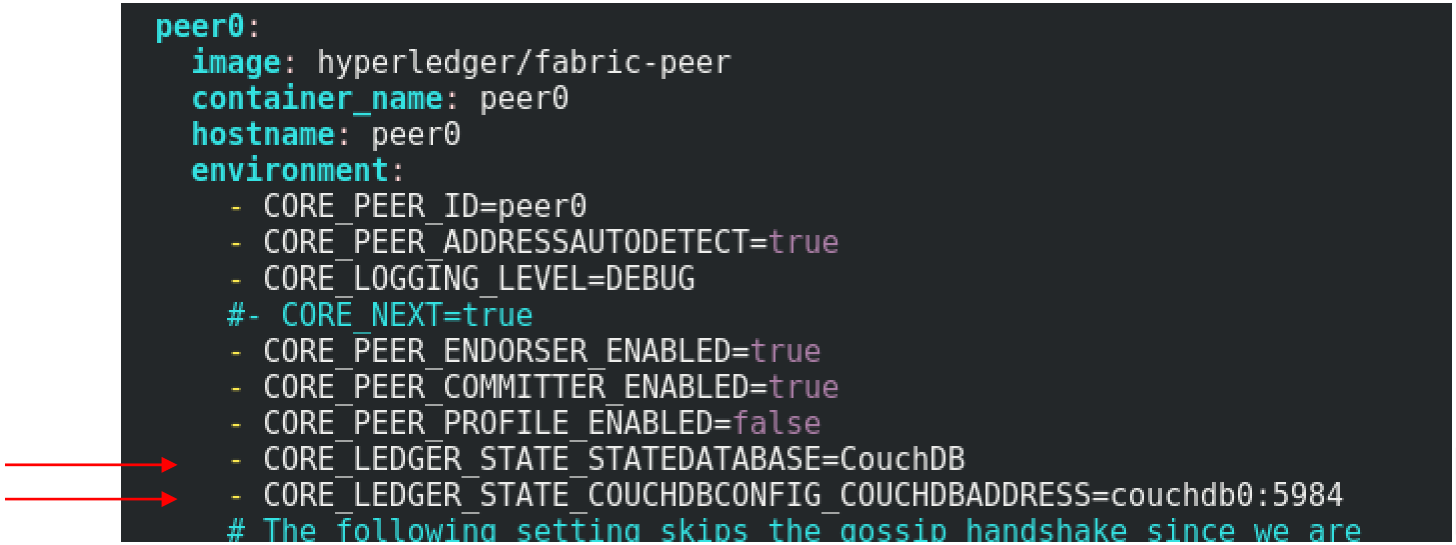
OPTIONS

State Ledger:

- World state ledger (default golvldb)
- Pluggable worldstate ledger (CouchDB)



```
[guigui@t460 ~]$ docker images | egrep 'peer|couch'
hyperledger/fabric-couchdb     latest           2b623819e8a1        12 days ago         1.47GB
hyperledger/fabric-couchdb     x86_64-1.0.3     2b623819e8a1        12 days ago         1.47GB
hyperledger/fabric-peer        latest           9a79041ee91e        12 days ago         154MB
hyperledger/fabric-peer        x86_64-1.0.3     9a79041ee91e        12 days ago         154MB
[guigui@t460 ~]$ 
```



```
peer0:
  image: hyperledger/fabric-peer
  container_name: peer0
  hostname: peer0
  environment:
    - CORE_PEER_ID=peer0
    - CORE_PEER_ADDRESSAUTODETECT=true
    - CORE_LOGGING_LEVEL=DEBUG
    #- CORE_NEXT=true
    - CORE_PEER_ENDORSER_ENABLED=true
    - CORE_PEER_COMMITTER_ENABLED=true
    - CORE_PEER_PROFILE_ENABLED=false
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb0:5984
    # The following setting skips the gossip handshake since we are
```

Hyperledger Fabric Orderer – options

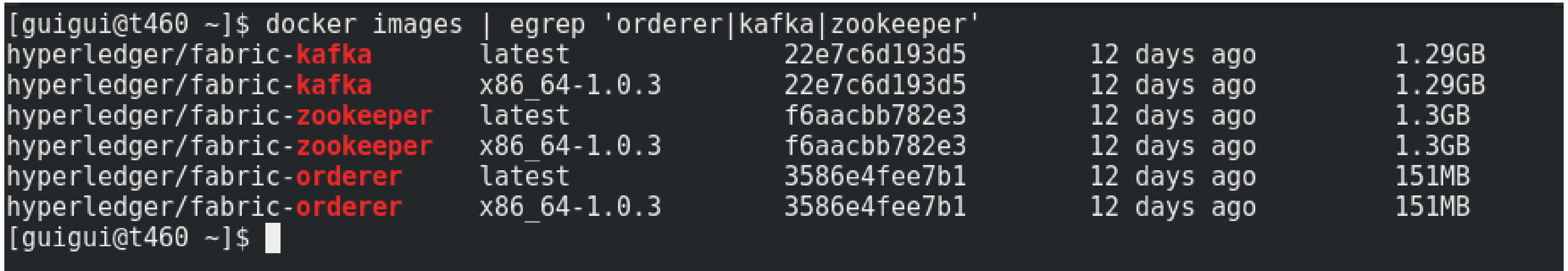
Orderer

In charge of ordering transactions in blocks. It is in charge of running the consensus.

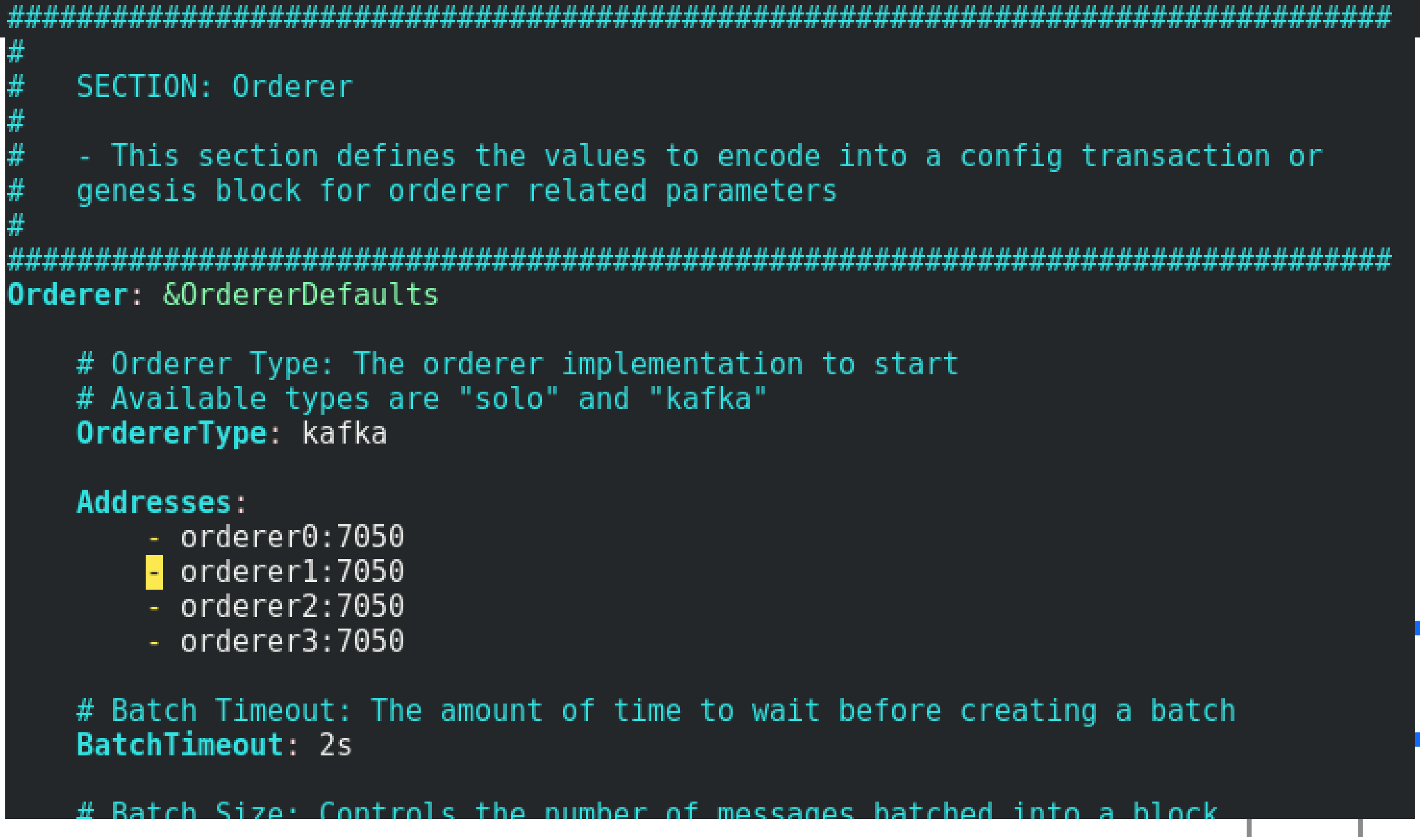
OPTIONS

Consensus:

- SOLO
- KAFKA
- Other alternatives to come (sBFT, PBFT…)



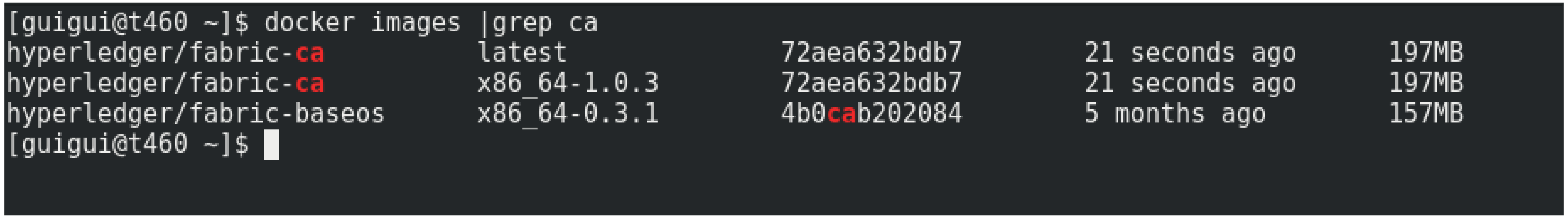
```
[guigui@t460 ~]$ docker images | egrep 'orderer|kafka|zookeeper'
hyperledger/fabric-kafka        latest          22e7c6d193d5        12 days ago         1.29GB
hyperledger/fabric-kafka        x86_64-1.0.3    22e7c6d193d5        12 days ago         1.29GB
hyperledger/fabric-zookeeper    latest          f6aacbb782e3        12 days ago         1.3GB
hyperledger/fabric-zookeeper    x86_64-1.0.3    f6aacbb782e3        12 days ago         1.3GB
hyperledger/fabric-orderer      latest          3586e4fee7b1        12 days ago         151MB
hyperledger/fabric-orderer      x86_64-1.0.3    3586e4fee7b1        12 days ago         151MB
[guigui@t460 ~]$ 
```



```
################################################################################
#
#   SECTION: Orderer
#
#   - This section defines the values to encode into a config transaction or
#   genesis block for orderer related parameters
#
################################################################################
Orderer: &OrdererDefaults

    # Orderer Type: The orderer implementation to start
    # Available types are "solo" and "kafka"
    OrdererType: kafka

    Addresses:
        - orderer0:7050
        - orderer1:7050
        - orderer2:7050
        - orderer3:7050

    # Batch Timeout: The amount of time to wait before creating a batch
    BatchTimeout: 2s

    # Batch Size: Controls the number of messages batched into a block
```

Hyperledger Fabric – Fabric-CA options

CA

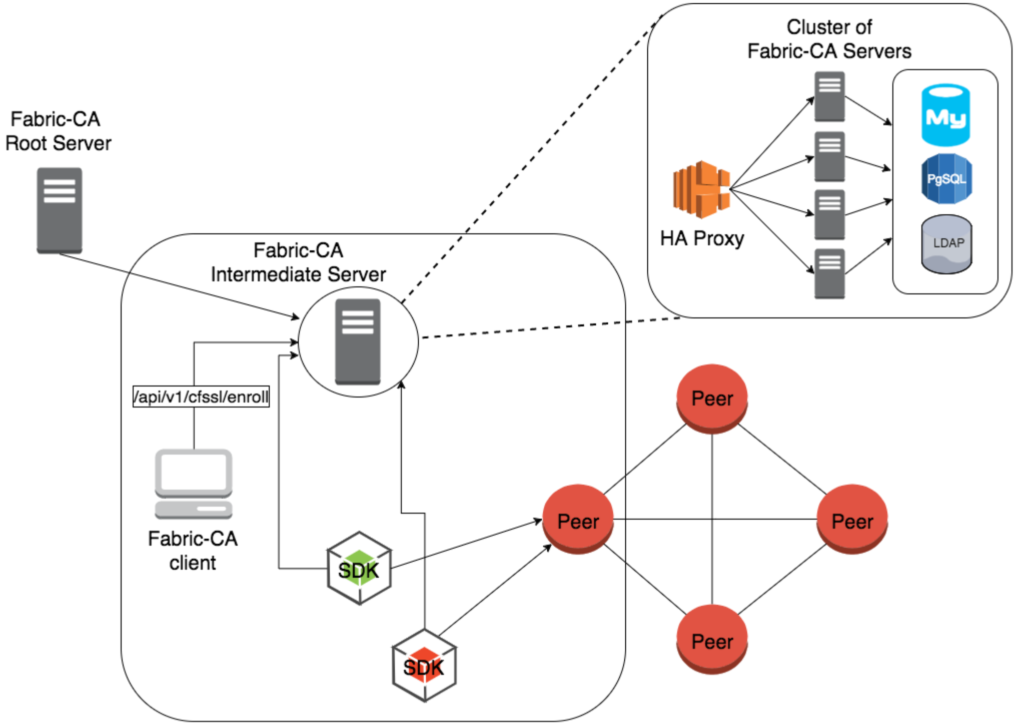


```
[guigui@t460 ~]$ docker images |grep ca
hyperledger/fabric-ca          latest              72aea632bdb7        21 seconds ago      197MB
hyperledger/fabric-ca          x86_64-1.0.3        72aea632bdb7        21 seconds ago      197MB
hyperledger/fabric-baseos      x86_64-0.3.1        4b0cab202084        5 months ago        157MB
[guigui@t460 ~]$ 
```

In charge of issuing digital identities across organizations.

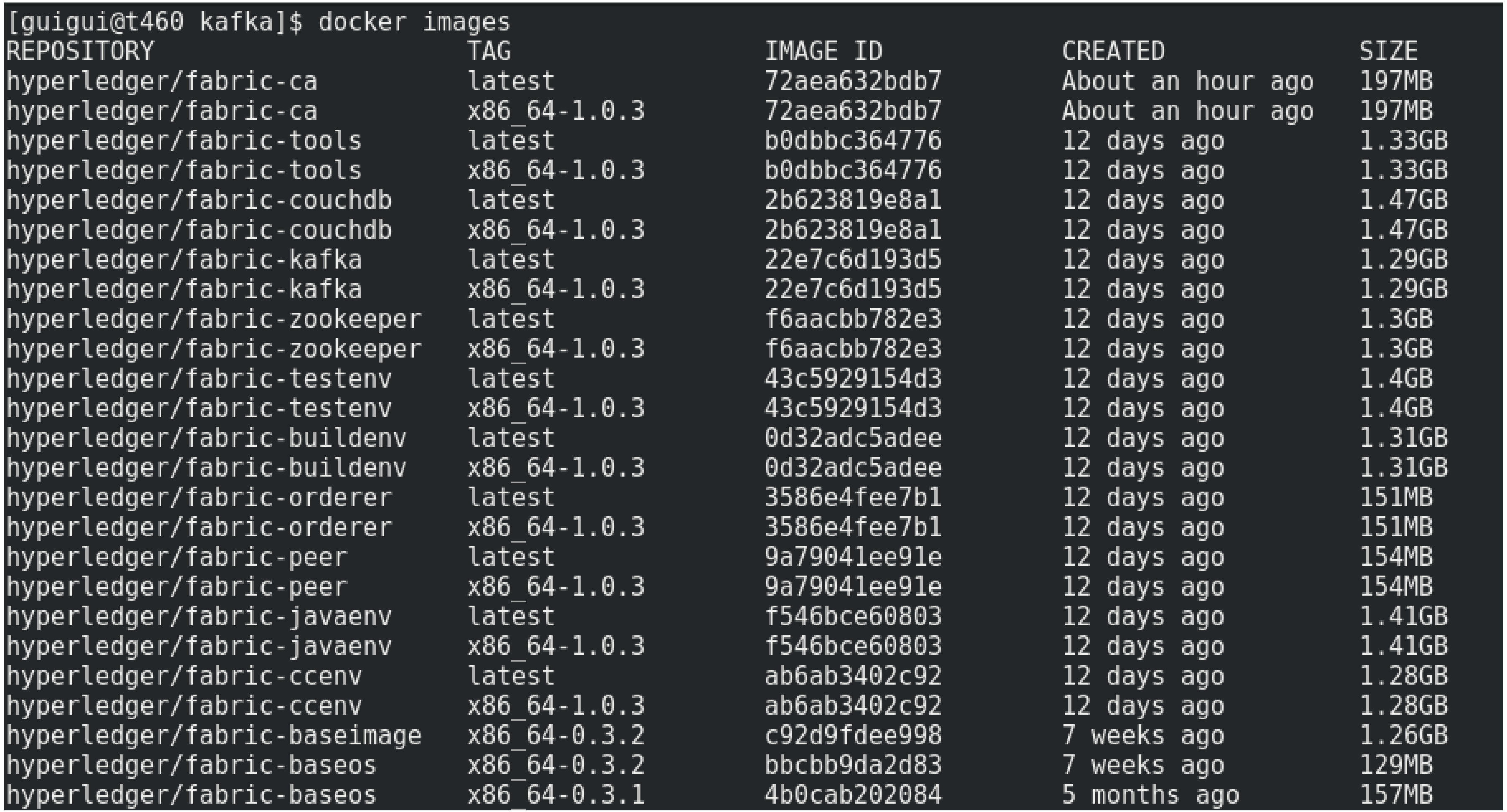
OPTIONS

- Key Store (Default SQLite3)
- Key Store (Alternative PostgreSQL
- LDAP Integration
- HA
- …
- Alternative with CryptoGen or existing PKI and CA



Docker images

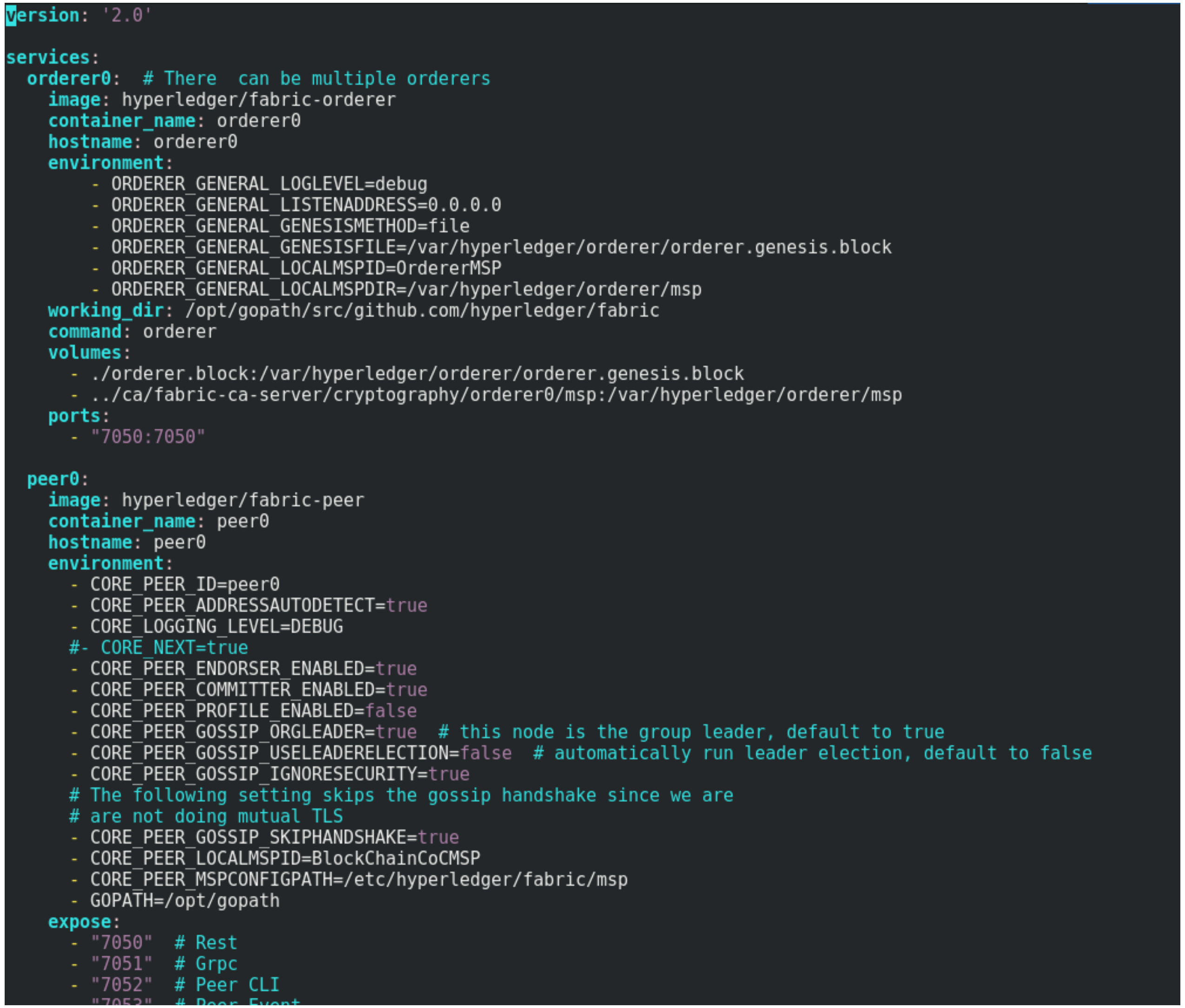
Once built, or pulled from the repository, these are the images which will be added to your Docker host.



```
[guigui@t460 kafka]$ docker images
REPOSITORY                     TAG                 IMAGE ID            CREATED             SIZE
hyperledger/fabric-ca          latest              72aea632bdb7        About an hour ago   197MB
hyperledger/fabric-ca          x86_64-1.0.3        72aea632bdb7        About an hour ago   197MB
hyperledger/fabric-tools       latest              b0dbbc364776        12 days ago         1.33GB
hyperledger/fabric-tools       x86_64-1.0.3        b0dbbc364776        12 days ago         1.33GB
hyperledger/fabric-couchdb     latest              2b623819e8a1        12 days ago         1.47GB
hyperledger/fabric-couchdb     x86_64-1.0.3        2b623819e8a1        12 days ago         1.47GB
hyperledger/fabric-kafka       latest              22e7c6d193d5        12 days ago         1.29GB
hyperledger/fabric-kafka       x86_64-1.0.3        22e7c6d193d5        12 days ago         1.29GB
hyperledger/fabric-zookeeper   latest              f6aacbb782e3        12 days ago         1.3GB
hyperledger/fabric-zookeeper   x86_64-1.0.3        f6aacbb782e3        12 days ago         1.3GB
hyperledger/fabric-testenv     latest              43c5929154d3        12 days ago         1.4GB
hyperledger/fabric-testenv     x86_64-1.0.3        43c5929154d3        12 days ago         1.4GB
hyperledger/fabric-buildenv    latest              0d32adc5adee        12 days ago         1.31GB
hyperledger/fabric-buildenv    x86_64-1.0.3        0d32adc5adee        12 days ago         1.31GB
hyperledger/fabric-orderer     latest              3586e4fee7b1        12 days ago         151MB
hyperledger/fabric-orderer     x86_64-1.0.3        3586e4fee7b1        12 days ago         151MB
hyperledger/fabric-peer        latest              9a79041ee91e        12 days ago         154MB
hyperledger/fabric-peer        x86_64-1.0.3        9a79041ee91e        12 days ago         154MB
hyperledger/fabric-javaenv     latest              f546bce60803        12 days ago         1.41GB
hyperledger/fabric-javaenv     x86_64-1.0.3        f546bce60803        12 days ago         1.41GB
hyperledger/fabric-ccenv       latest              ab6ab3402c92        12 days ago         1.28GB
hyperledger/fabric-ccenv       x86_64-1.0.3        ab6ab3402c92        12 days ago         1.28GB
hyperledger/fabric-baseimage   x86_64-0.3.2        c92d9fdee998        7 weeks ago         1.26GB
hyperledger/fabric-baseos      x86_64-0.3.2        bbcbb9da2d83        7 weeks ago         129MB
hyperledger/fabric-baseos      x86_64-0.3.1        4b0cab202084        5 months ago        157MB
```

Composing images to build the network

Tool like docker-compose are a simple way to organize the deployment of containers to build up the Fabric. Samples are provided in the examples subdirectory of the github repo.



```
version: '2.0'

services:
  orderer0:  # There  can be multiple orderers
    image: hyperledger/fabric-orderer
    container_name: orderer0
    hostname: orderer0
    environment:
        - ORDERER_GENERAL_LOGLEVEL=debug
        - ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
        - ORDERER_GENERAL_GENESISMETHOD=file
        - ORDERER_GENERAL_GENESISFILE=/var/hyperledger/orderer/orderer.genesis.block
        - ORDERER_GENERAL_LOCALMSPID=OrdererMSP
        - ORDERER_GENERAL_LOCALMSPDIR=/var/hyperledger/orderer/msp
    working_dir: /opt/gopath/src/github.com/hyperledger/fabric
    command: orderer
    volumes:
      - ./orderer.block:/var/hyperledger/orderer/orderer.genesis.block
      - ../ca/fabric-ca-server/cryptography/orderer0/msp:/var/hyperledger/orderer/msp
    ports:
      - "7050:7050"

  peer0:
    image: hyperledger/fabric-peer
    container_name: peer0
    hostname: peer0
    environment:
      - CORE_PEER_ID=peer0
      - CORE_PEER_ADDRESSAUTODETECT=true
      - CORE_LOGGING_LEVEL=DEBUG
      #- CORE_NEXT=true
      - CORE_PEER_ENDORSER_ENABLED=true
      - CORE_PEER_COMMITTER_ENABLED=true
      - CORE_PEER_PROFILE_ENABLED=false
      - CORE_PEER_GOSSIP_ORGLEADER=true  # this node is the group leader, default to true
      - CORE_PEER_GOSSIP_USELEADERELECTION=false  # automatically run leader election, default to false
      - CORE_PEER_GOSSIP_IGNORESECURITY=true
      # The following setting skips the gossip handshake since we are
      # are not doing mutual TLS
      - CORE_PEER_GOSSIP_SKIPHANDSHAKE=true
      - CORE_PEER_LOCALMSPID=BlockChainCoCMSP
      - CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/fabric/msp
      - GOPATH=/opt/gopath
    expose:
      - "7050"  # Rest
      - "7051"  # Grpc
      - "7052"  # Peer CLI
      - "7053"  # Peer Event
```