Hyperledyer Fabric: Manual de instalación

René Dávila - Jorge Solano

Manual de instalación de Hyperledger Fabric.

Prerequisitos

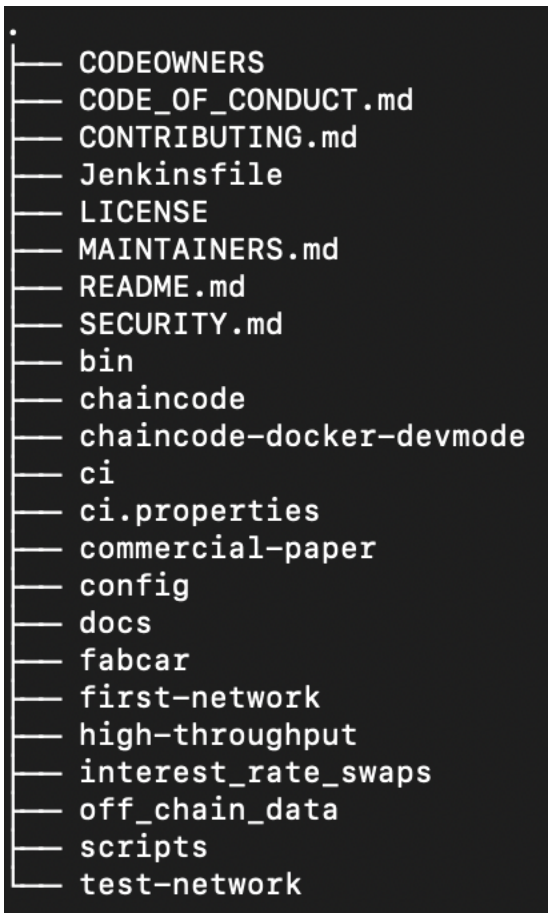
1. Instalar git:
   https://git-scm.com/downloads.
2. Instalar cURL:
   https://curl.haxx.se/download.html.
3. Instalar Docker y Docker Compose (Se requiere la versión 17.06.2-ce o superior):
   https://www.docker.com/get-docker.
4. Instalar Go:
   https://golang.org/dl/.
5. Instalar Node.js y NPM: https://nodejs.org/en/download/.
6. Instalar Python: https://www.python.org/downloads/.

Para construir el escenario de la primera red en Hyperleadger Fabric es necesario, después de instalar los prerequisitos, instalar los ejemplos, binarios e imágenes Docker que se encuentran en https://hyperledger-fabric.readthedocs.io/en/latest/install.html. Para ello hay que inicar Docker Desktop y ejecutar la línea instrucción:

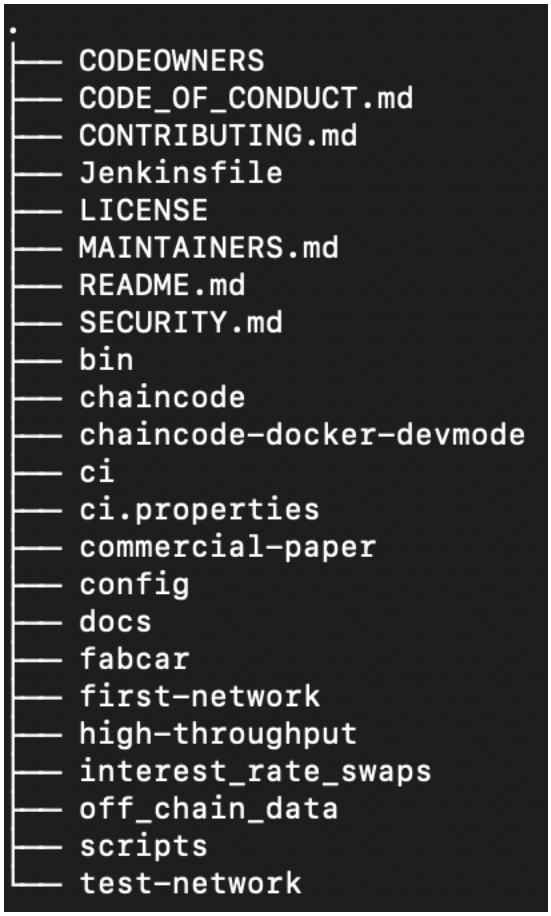
curl -sSL https://bit.ly/2ysbOFE | bash -s

La instrucción anterior descarga la carpeta *fabric-samples* en la ubicación actual

La fábrica de ejemplos descarga varios proyectos con diferentes escenarios.



Para iniciar se recomienda ejecutar el ejemplo test-network, el cual permite ejecutar nodos en la máquina local. Este proyecto también permite probar aplicaciones y contratos inteligentes.



Si se ejecuta el script byfn.sh (build your first network) sin argumentos, como salida se obtiene una descripción de diversas opciones para ejecutar el ejemplo.

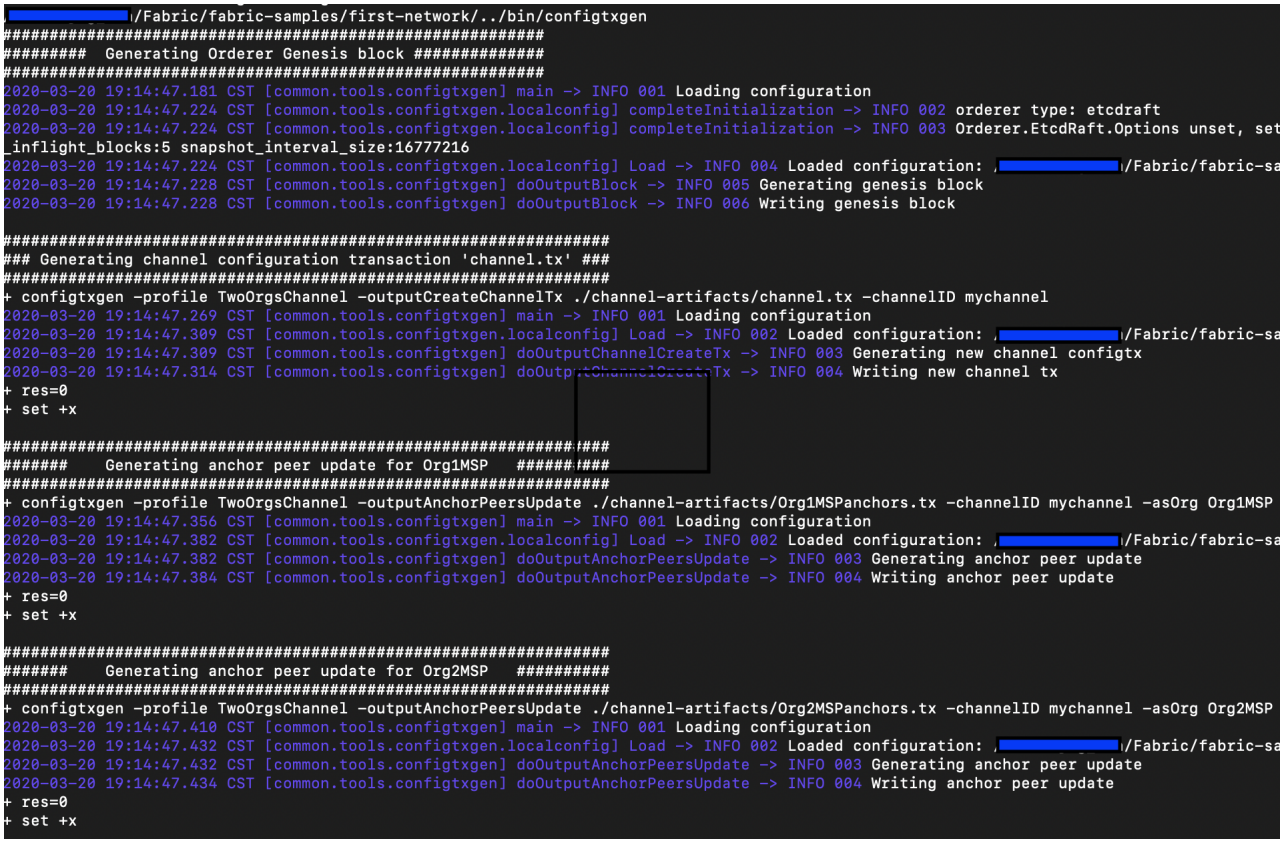
```
Usage:
  byfn.sh <mode> [-c <channel name>] [-t <timeout>] [-d <delay>] [-f <docker-compose-file>] [-s <dbtype>] [
-l <language>] [-o <consensus-type>] [-i <imagetag>] [-a] [-n] [-v]
    <mode> - one of 'up', 'down', 'restart', 'generate' or 'upgrade'
      - 'up' - bring up the network with docker-compose up
      - 'down' - clear the network with docker-compose down
      - 'restart' - restart the network
      - 'generate' - generate required certificates and genesis block
      - 'upgrade'  - upgrade the network from version 1.3.x to 1.4.0
    -c <channel name> - channel name to use (defaults to "mychannel")
    -t <timeout> - CLI timeout duration in seconds (defaults to 10)
    -d <delay> - delay duration in seconds (defaults to 3)
    -s <dbtype> - the database backend to use: goleveldb (default) or couchdb
    -l <language> - the programming language of the chaincode to deploy: go (default), javascript, or java
    -i <imagetag> - the tag to be used to launch the network (defaults to "latest")
    -a - launch certificate authorities (no certificate authorities are launched by default)
    -n - do not deploy chaincode (abstore chaincode is deployed by default)
    -v - verbose mode
  byfn.sh -h (print this message)

Typically, one would first generate the required certificates and
genesis block, then bring up the network. e.g.:

        byfn.sh generate -c mychannel
        byfn.sh up -c mychannel -s couchdb
        byfn.sh up -c mychannel -s couchdb -i 1.4.0
        byfn.sh up -l javascript
        byfn.sh down -c mychannel
        byfn.sh upgrade -c mychannel

Taking all defaults:
        byfn.sh generate
        byfn.sh up
        byfn.sh down
```

Para generar la red **Artifacts** del ejemplo byfn hay que ejecutar el script con la opción generate (./byfn.sh generate), lo cual genera los certificados de las organizaciones (org1 y org2) y el canal de comunicación (mychannel).



Para levantar la red se ejecuta la instrucción ./byfn.sh up