

# Compose 部署 Fabric-explorer

作者: [jockming112](#)

原文链接: <https://ld246.com/article/1608046653955>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 版本选择

当前使用的版本是v.1.1.3

## 注意

1. 你得实现准备好一个运行正常的fabric网络。网络部署可以参考我的其他博客。
2. 将证书目录crypto-config复制到explorer项目下。
3. 提醒下，防火墙注意要开放端口，云服务器注意安全组添加端口。

**NOTE:** 本文使用docker部署，可以经可能避免node安装的一些坑。

### Hyperledger Explorer Version odeJS Version Supported

v1.1.3 (Sep 28, 2020)  
2.16.x

v1.1.2 (Aug 12, 2020)  
2.16.x

### Fabric Version Supported

v1.4.0 to v2.2.0

v1.4.0 to v2.2.0

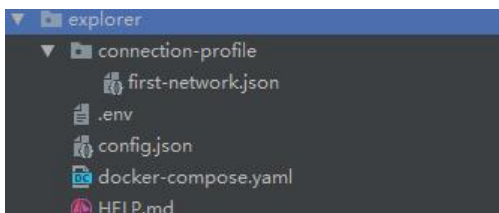
## 拉取镜像(可选操作)

```
docker pull hyperledger/explorer-db:1.1.3  
docker pull hyperledger/explorer:1.1.3
```

## 拉取配置(可选操作)

```
# 建议使用本项目内的配置，官方配置拉取下来需要修改  
wget https://github.com/hyperledger/blockchain-explorer/tree/v1.1.3/examples/net1/config.j  
on  
wget https://github.com/hyperledger/blockchain-explorer/tree/v1.1.3/examples/net1/connect  
on-profile/first-network.json -P connection-profile  
wget https://github.com/hyperledger/blockchain-explorer/tree/v1.1.3/docker-compose.yaml
```

## 项目结构



## first-network.json

```
{  
  "name": "first-network",  
  "version": "1.4.1",  
  "client": {
```

```

"tlsEnable": true,
"adminCredential": {
  "id": "admin",
  "password": "admin"
},
"enableAuthentication": true,
"organization": "Org1",
"connection": {
  "timeout": {
    "peer": {
      "endorser": "300"
    }
  },
  "orderer": "300"
}
},
"channels": {
  "score": {
    "peers": {
      "peer0.org1.example.com": {},
      "peer1.org1.example.com": {}
    }
  },
  "connection": {
    "timeout": {
      "peer": {
        "endorser": "6000",
        "eventHub": "6000",
        "eventReg": "6000"
      }
    }
  }
}
},
"organizations": {
  "Org1": {
    "mspId": "Org1MSP",
    "adminPrivateKey": {
      "path": "/tmp/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp/keystore/6b79f64e047bb7f31ab0d9d295e3d25f2a168f2f24c8ad3ef15e46b93130f4c_sk"
    },
    "peers": ["peer0.org1.example.com", "peer1.org1.example.com"],
    "signedCert": {
      "path": "/tmp/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp/signcerts/Admin@org1.example.com-cert.pem"
    }
  }
},
"peers": {
  "peer0.org1.example.com": {
    "tlsCACerts": {
      "path": "/tmp/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt"
    }
  },

```

```

    "url": "grpcs://peer0.org1.example.com:7051"
  },
  "peer1.org1.example.com": {
    "tlsCACerts": {
      "path": "/tmp/crypto/peerOrganizations/org1.example.com/peers/peer1.org1.example.c
m/tls/ca.crt"
    },
    "url": "grpcs://peer1.org1.example.com:8051"
  }
}
}
}

```

## **.env**

```

## Docker-compose config for setting project name
COMPOSE_PROJECT_NAME=fabric-explorer

## Image tags - Fabric versions.
IMAGE_TAG_EXPLORER=1.1.3

## At deployment time, modify the IP based on your host
ORDERER="orderer.example.com:47.108.143.240"
PEER0_ORG1="peer0.org1.example.com:47.108.143.240"
PEER1_ORG1="peer1.org1.example.com:47.108.143.240"

```

## **config.json**

```

{
  "network-configs": {
    "first-network": {
      "name": "first-network",
      "profile": "./connection-profile/first-network.json"
    }
  },
  "license": "Apache-2.0"
}

```

## **docker-compose.yaml**

```

version: '2.1'

networks:
  mynetwork.com:
    external:
      name: net_test

services:
  explorerdb.mynetwork.com:
    image: hyperledger/explorer-db:${IMAGE_TAG_EXPLORER}
    container_name: explorerdb.mynetwork.com
    hostname: explorerdb.mynetwork.com

```

```
environment:
- DATABASE_DATABASE=fabricexplorer
- DATABASE_USERNAME=fabric
- DATABASE_PASSWORD=123456
healthcheck:
test: "pg_isready -h localhost -p 5432 -q -U postgres"
interval: 30s
timeout: 10s
retries: 5
volumes:
# Pay attention to synchronization time
# echo 'Asia/Shanghai' > /etc/timezone/timezone
- /etc/timezone/timezone:/etc/timezone
- /etc/localtime:/etc/localtime
- /var/run:/host/var/run/
- ./pgdata:/var/lib/postgresql/data
ports:
- 5432:5432
networks:
- mynetwork.com
```

```
explorer.mynetwork.com:
image: hyperledger/explorer:${IMAGE_TAG_EXPLORER}
container_name: explorer.mynetwork.com
hostname: explorer.mynetwork.com
environment:
- DATABASE_HOST=47.108.143.240
- DATABASE_DATABASE=fabricexplorer
- DATABASE_USERNAME=fabric
- DATABASE_PASSWD=123456
- LOG_LEVEL_APP=debug
- LOG_LEVEL_DB=debug
- LOG_LEVEL_CONSOLE=info
- LOG_CONSOLE_STDOUT=true
- DISCOVERY_AS_LOCALHOST=false
#command: sh -c "node /opt/explorer/main.js && tail -f /dev/null"
volumes:
# Pay attention to synchronization time
# echo 'Asia/Shanghai' > /etc/timezone/timezone
- /etc/timezone/timezone:/etc/timezone
- /etc/localtime:/etc/localtime
- /var/run:/host/var/run/
- ./config.json:/opt/explorer/app/platform/fabric/config.json
- ./connection-profile:/opt/explorer/app/platform/fabric/connection-profile
- ./crypto-config:/tmp/crypto
- ./walletstore:/opt/wallet
ports:
- 8080:8080
extra_hosts:
- ${ORDERER}
- ${PEER0_ORG1}
- ${PEER1_ORG1}
depends_on:
explorerdb.mynetwork.com:
```

```
condition: service_healthy
networks:
- mynetwork.com
```

## 创建网络

```
docker network create net_test
```

## 修改配置

# 如果重新生成了组织证书，请替换组织管理员证书，注意替换通道名，组织名称，管理员账号密码及管理员证书。

```
vi ./connection-profile/first-network.json
```

## 启动网络

# 启动过程会花费点时间，注意检查日志，看是否正常启动，启动完成即可访问页面

```
docker-compose up -d
```

## 访问页面

```
http://your_ip:8080
```