GENERATING CERTIFICATES **USING FABRIC-CA**

- Generating Certificates using docker-compose-ca.yaml file
- Creating registerEnroll.sh file

docker-compose-ca.yaml

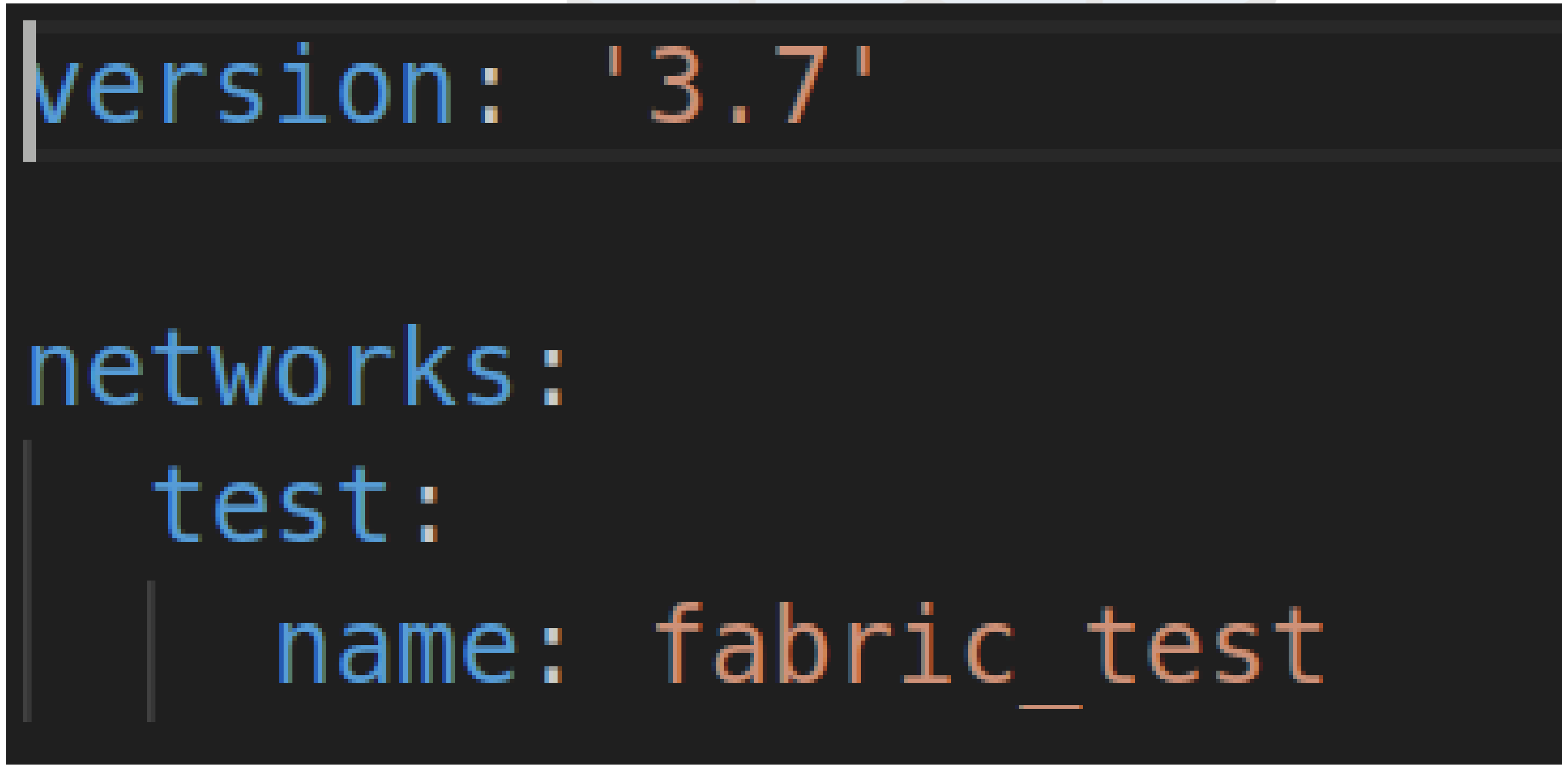
- Let's use a docker compose file to deploy the fabric-ca.
- In this docker file, a CA for every organization is created.
- Create a new file inside the docker folder called docker-compose-ca.yaml

Overview of the docker-compose-ca file

- Specify the services for the ca server
  - ca_org1
  - ca_org2
  - ca_orderer
- Use the **fabric-ca-server start** command to start the fabric-ca-server
- Set the environment variables:
  - FABRIC_CA_HOME
  - FABRIC_CA_SERVER_CA_NAME
  - FABRIC_CA_SERVER_TLS_ENABLED
  - FABRIC_CA_SERVER_PORT
  - FABRIC_CA_SERVER_OPERATIONS_LISTENADDRESS
- Also, specify the container name, image, labels, ports, volumes

docker-compose-ca.yaml

- **version** – Version of docker-compose file
- **networks** – Docker network is defined
    - **name** – Name of the docker network which is currently used.

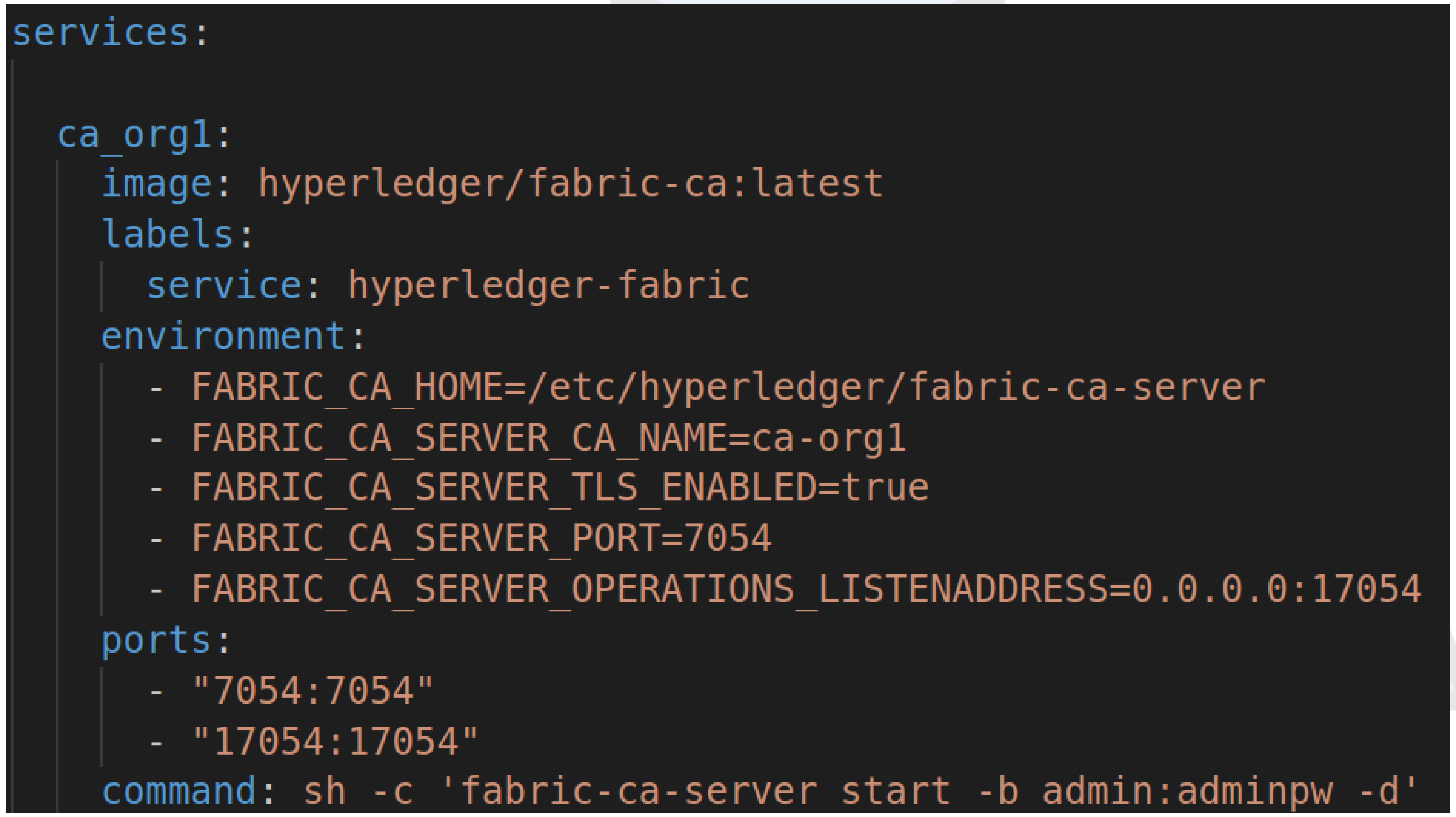


```
version: '3.7'

networks:
  test:
    name: fabric_test
```

docker-compose-ca.yaml

- **services** – Different types of services which help in building the application. Each service represents a different part of your application.(Individual containers)
- **ca_org1**: Specifies the name of the service.(Name of the container created).
- **image: hyperledger/fabric-ca:latest-** Container should be created using this image
- **environment** – Sets up the env variables
- **ports** – used for CA operations (the left port is mounted with the right port of the container)
- **command: sh -c 'fabric-ca-server start -b admin:adminpw -d'** – Starts the fabric ca-server along with admin credentials.

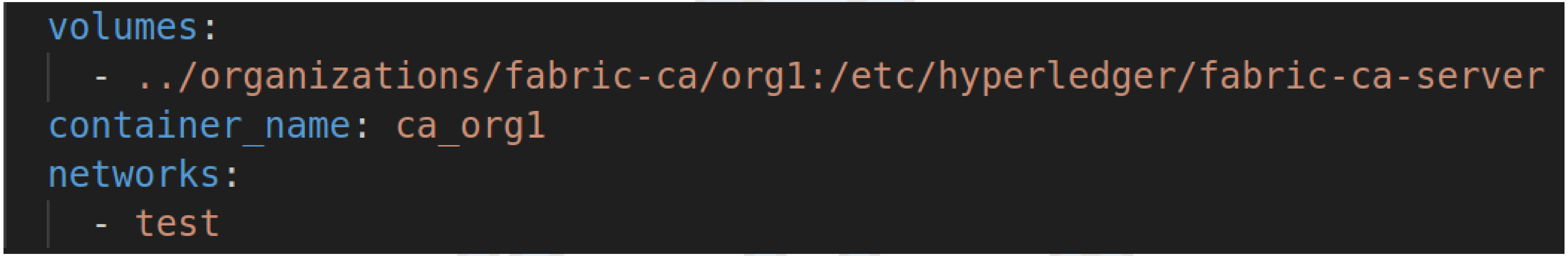


```yaml
services:

  ca_org1:
    image: hyperledger/fabric-ca:latest
    labels:
      service: hyperledger-fabric
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca-org1
      - FABRIC_CA_SERVER_TLS_ENABLED=true
      - FABRIC_CA_SERVER_PORT=7054
      - FABRIC_CA_SERVER_OPERATIONS_LISTENADDRESS=0.0.0.0:17054
    ports:
      - "7054:7054"
      - "17054:17054"
    command: sh -c 'fabric-ca-server start -b admin:adminpw -d'
```

docker-compose-ca.yaml

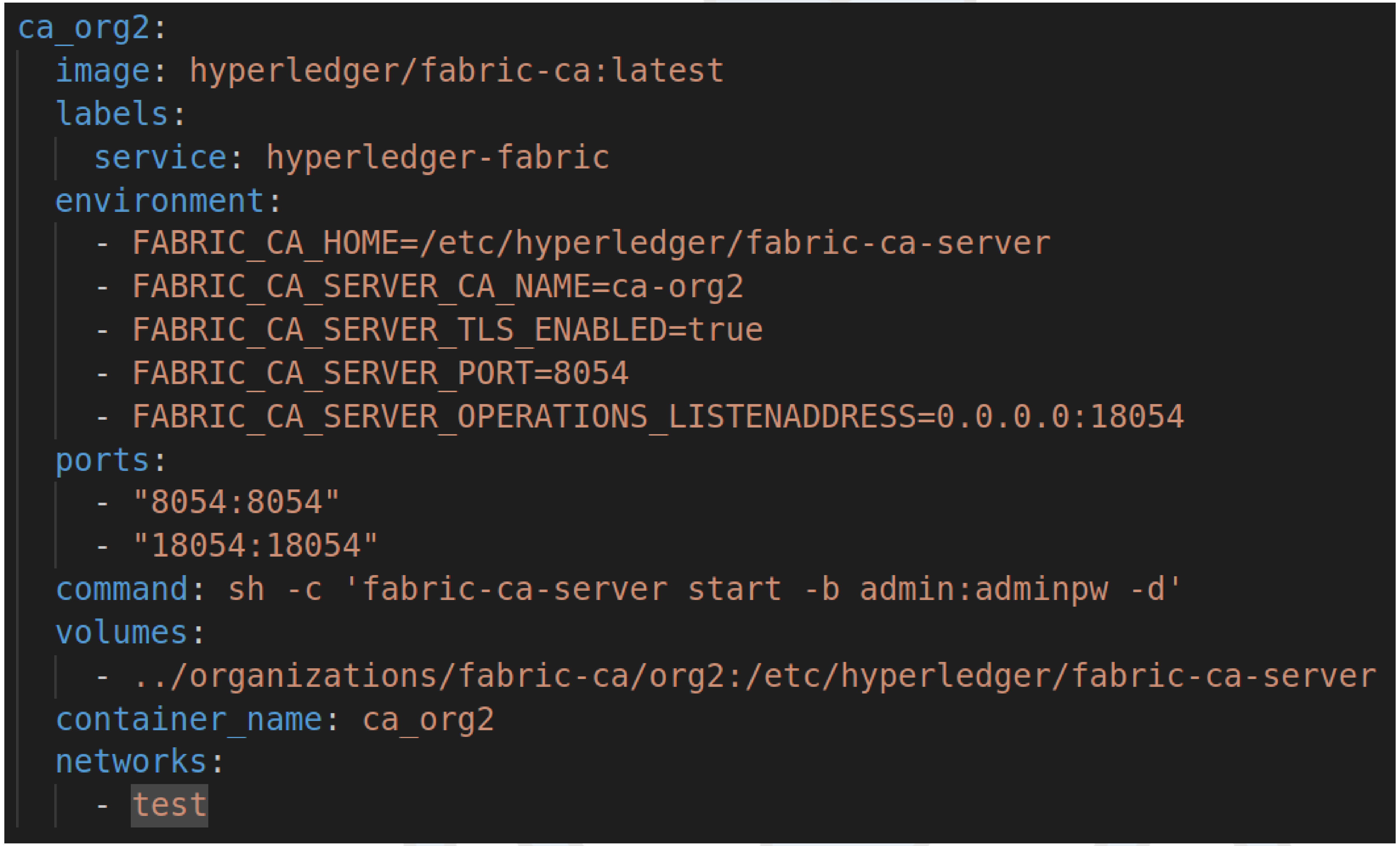
- **volumes** - Specifies volume mounting between the directory in the local system and in the container
- **container_name** - Sets the name of the container
- **networks**
- **test** - Specifies the docker container should belong to the network test.



```
volumes:
  - ../organizations/fabric-ca/org1:/etc/hyperledger/fabric-ca-server
container_name: ca_org1
networks:
  - test
```

docker-compose-ca.yaml

Ports, mspid and folder structures are changed

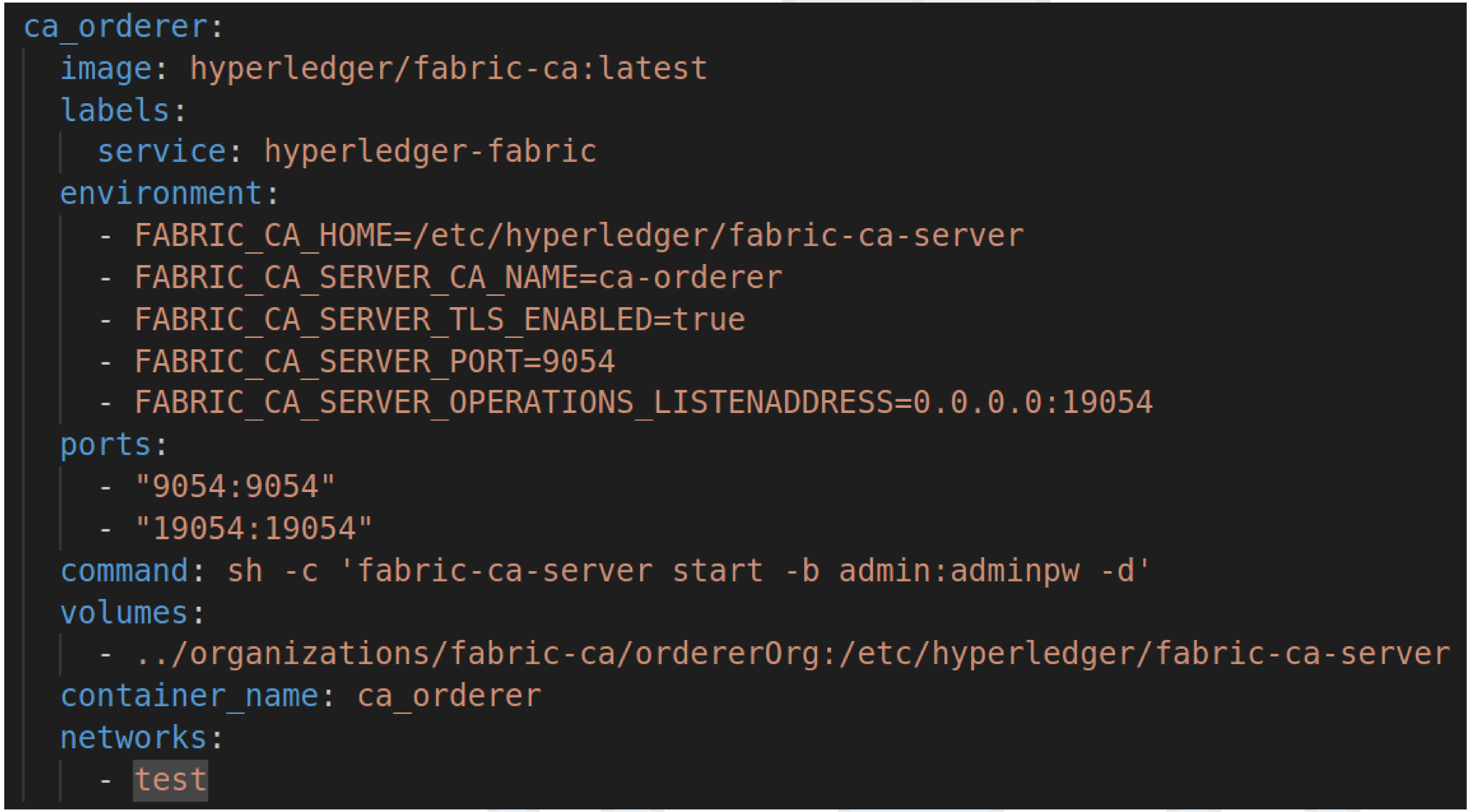


```yaml
ca_org2:
  image: hyperledger/fabric-ca:latest
  labels:
    service: hyperledger-fabric
  environment:
    - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
    - FABRIC_CA_SERVER_CA_NAME=ca-org2
    - FABRIC_CA_SERVER_TLS_ENABLED=true
    - FABRIC_CA_SERVER_PORT=8054
    - FABRIC_CA_SERVER_OPERATIONS_LISTENADDRESS=0.0.0.0:18054
  ports:
    - "8054:8054"
    - "18054:18054"
  command: sh -c 'fabric-ca-server start -b admin:adminpw -d'
  volumes:
    - ../organizations/fabric-ca/org2:/etc/hyperledger/fabric-ca-server
  container_name: ca_org2
  networks:
    - test
```

ca_org2

docker-compose-ca.yaml

Ports, mspid and folder structures are changed



```yaml
ca_orderer:
  image: hyperledger/fabric-ca:latest
  labels:
    service: hyperledger-fabric
  environment:
    - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
    - FABRIC_CA_SERVER_CA_NAME=ca-orderer
    - FABRIC_CA_SERVER_TLS_ENABLED=true
    - FABRIC_CA_SERVER_PORT=9054
    - FABRIC_CA_SERVER_OPERATIONS_LISTENADDRESS=0.0.0.0:19054
  ports:
    - "9054:9054"
    - "19054:19054"
  command: sh -c 'fabric-ca-server start -b admin:adminpw -d'
  volumes:
    - ../organizations/fabric-ca/ordererOrg:/etc/hyperledger/fabric-ca-server
  container_name: ca_orderer
  networks:
    - test
```

ca_orderer