

部署应用程序服务端与前端

服务端部署

1. 镜像恢复

```
docker load -i api-server.tar.gz
docker load -i demo-chaincode-portal2.tar.gz
```

执行 `docker images` 查看镜像列表

```
root@ubuntu:~/Desktop/server# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
api-server           latest              59a90625b192       43 minutes ago     777MB
<none>               <none>              90e22120f27a       5 hours ago        787MB
demo-chaincode-portal2 latest              f7b4d8a28778       2 months ago       1.23GB
```

2. 先执行如下命令启动容器

```
docker run -p 8080:8080 -it api-server:latest sh
```

3. 容器启动后，需要先配置hosts域名解析规则，添加对应的order和peer节点的域名及EIP。

```
echo "49.4.3.233 orderer-773e7ba33ef2cab07d34450e7ca1a1cd826992e0-0.orderer-773e7ba33ef2cab07d34450e7ca1a1cd826992e0.default.svc.cluster.local" >>/etc/hosts
echo "49.4.3.233 orderer-773e7ba33ef2cab07d34450e7ca1a1cd826992e0-1.orderer-773e7ba33ef2cab07d34450e7ca1a1cd826992e0.default.svc.cluster.local" >>/etc/hosts
echo "49.4.3.233 orderer-773e7ba33ef2cab07d34450e7ca1a1cd826992e0-2.orderer-773e7ba33ef2cab07d34450e7ca1a1cd826992e0.default.svc.cluster.local" >>/etc/hosts
echo "49.4.3.233 orderer-773e7ba33ef2cab07d34450e7ca1a1cd826992e0-3.orderer-773e7ba33ef2cab07d34450e7ca1a1cd826992e0.default.svc.cluster.local" >>/etc/hosts
echo "49.4.3.233 peer-efa57c13cba33cb1c4b16dc296020aa72131fdc4-0.peer-efa57c13cba33cb1c4b16dc296020aa72131fdc4.default.svc.cluster.local" >>/etc/hosts
echo "49.4.3.233 peer-6c6ab88c970aedc70dd86588a3f5dc9dc1022237-0.peer-6c6ab88c970aedc70dd86588a3f5dc9dc1022237.default.svc.cluster.local" >>/etc/hosts
echo "49.4.3.233 peer-bde653e28dcb15bbf956e86c07107a2955b20be1-0.peer-bde653e28dcb15bbf956e86c07107a2955b20be1.default.svc.cluster.local" >>/etc/hosts
```

4. 执行./api-server启动服务。

```

root@ubuntu: ~/Desktop/server
172.17.0.2 807293e72444
root@807293e72444:/opt/gopath/src/github.com/hyperledger/api-server# echo "49.4.54.236 orderer-0cbe7ae07e47df2e621c29a4318d106292148b69-0.orderer-0cbe7ae07e47df2e621c29a4318d106292148b69.default.svc.cluster.local" >>etc/hosts
root@807293e72444:/opt/gopath/src/github.com/hyperledger/api-server# echo "49.4.54.236 orderer-0cbe7ae07e47df2e621c29a4318d106292148b69-1.orderer-0cbe7ae07e47df2e621c29a4318d106292148b69.default.svc.cluster.local" >>etc/hosts
root@807293e72444:/opt/gopath/src/github.com/hyperledger/api-server# echo "49.4.54.236 orderer-0cbe7ae07e47df2e621c29a4318d106292148b69-2.orderer-0cbe7ae07e47df2e621c29a4318d106292148b69.default.svc.cluster.local" >>etc/hosts
root@807293e72444:/opt/gopath/src/github.com/hyperledger/api-server# echo "49.4.54.236 orderer-0cbe7ae07e47df2e621c29a4318d106292148b69-3.orderer-0cbe7ae07e47df2e621c29a4318d106292148b69.default.svc.cluster.local" >>etc/hosts
root@807293e72444:/opt/gopath/src/github.com/hyperledger/api-server# echo "49.4.54.236 peer-865fd2221e12590cc8ec9fb8eabda1dd7cfd14fd-0.peer-865fd2221e12590cc8ec9fb8eabda1dd7cfd14fd.default.svc.cluster.local" >>etc/hosts
root@807293e72444:/opt/gopath/src/github.com/hyperledger/api-server# echo "49.4.54.236 peer-44c1d1c1dc8e15d6543aaca24bff45a5c85a15c7-0.peer-44c1d1c1dc8e15d6543aaca24bff45a5c85a15c7.default.svc.cluster.local" >>etc/hosts
root@807293e72444:/opt/gopath/src/github.com/hyperledger/api-server# echo "49.4.54.236 peer-aca43d18f0774a2ee8939b38e92677cd25c61c50-0.peer-aca43d18f0774a2ee8939b38e92677cd25c61c50.default.svc.cluster.local" >>etc/hosts
root@807293e72444:/opt/gopath/src/github.com/hyperledger/api-server# cat /etc/hosts
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff00::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2 807293e72444
49.4.54.236 orderer-0cbe7ae07e47df2e621c29a4318d106292148b69-0.orderer-0cbe7ae07e47df2e621c29a4318d106292148b69.default.svc.cluster.local
49.4.54.236 orderer-0cbe7ae07e47df2e621c29a4318d106292148b69-1.orderer-0cbe7ae07e47df2e621c29a4318d106292148b69.default.svc.cluster.local
49.4.54.236 orderer-0cbe7ae07e47df2e621c29a4318d106292148b69-2.orderer-0cbe7ae07e47df2e621c29a4318d106292148b69.default.svc.cluster.local
49.4.54.236 orderer-0cbe7ae07e47df2e621c29a4318d106292148b69-3.orderer-0cbe7ae07e47df2e621c29a4318d106292148b69.default.svc.cluster.local
49.4.54.236 peer-865fd2221e12590cc8ec9fb8eabda1dd7cfd14fd-0.peer-865fd2221e12590cc8ec9fb8eabda1dd7cfd14fd.default.svc.cluster.local
49.4.54.236 peer-44c1d1c1dc8e15d6543aaca24bff45a5c85a15c7-0.peer-44c1d1c1dc8e15d6543aaca24bff45a5c85a15c7.default.svc.cluster.local
49.4.54.236 peer-aca43d18f0774a2ee8939b38e92677cd25c61c50-0.peer-aca43d18f0774a2ee8939b38e92677cd25c61c50.default.svc.cluster.local
root@807293e72444:/opt/gopath/src/github.com/hyperledger/api-server# ./api-server
2018/05/22 07:39:52 [I] [asm_and64.s:2361] http server running on http://0.0.0.0:8080

```

注意：

为了便于开发，我们可以在虚拟机的环境中同样添加上面的配置信息，同时将证书文件放到指定的目录中，让部署好的前端直接访问我们的开发环境。

前端部署

1. 执行如下命令启动容器

```
docker run -p 4200:4200 -e API_SERVER_IP=127.0.0.1 -e API_SERVER_PORT=8080 -it demo-chaincode-portal2:latest sh
```

2. 容器启动后，进入容器执行./start.sh启动服务

```

root@ubuntu: ~/Desktop/server
# ./start.sh
> baasbankdemo@1.0.0 start /opt/bankdemo
> ng serve --proxy-config proxy.conf.js

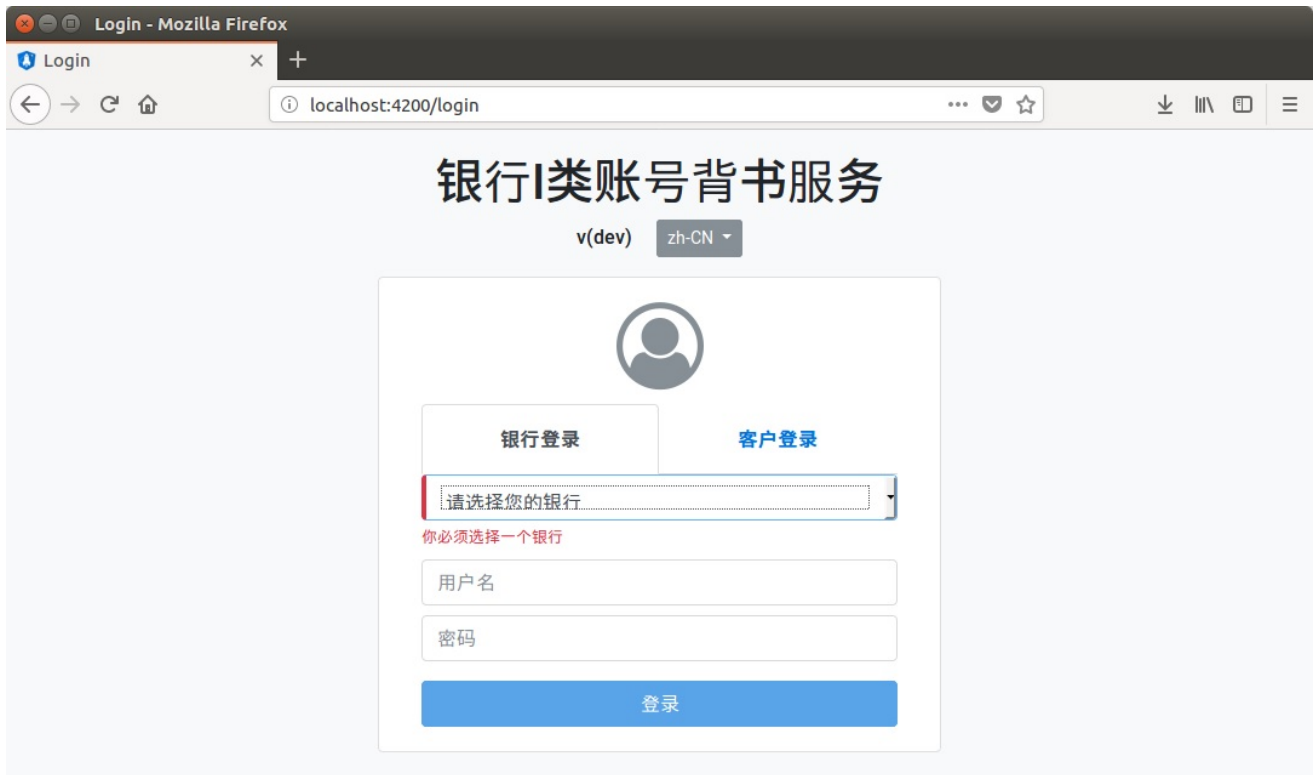
** NG Live Development Server is listening on 0.0.0.0:4200, open your browser on http://localhost:4200/ **
10% building modules 3/3 modules 0 active[HPM] Proxy created: /api -> http://127.0.0.1:8080/blockchain
[HPM] Proxy rewrite rule created: "/api" -> ""
Date: 2018-05-22T07:48:22.551Z
Hash: 9a50f7c3a071f503841d
Time: 11727ms
chunk {inline} inline.bundle.js (inline) 5.79 kB [entry] [rendered]
chunk {main} main.bundle.js (main) 456 kB [initial] [rendered]
chunk {polyfills} polyfills.bundle.js (polyfills) 950 kB [initial] [rendered]
chunk {styles} styles.bundle.js (styles) 651 kB [initial] [rendered]
chunk {vendor} vendor.bundle.js (vendor) 10.4 MB [initial] [rendered]

webpack: Compiled successfully.
webpack: Compiling...
Date: 2018-05-22T07:48:23.593Z
Hash: 019c2c66aba7b13a1047
Time: 633ms
chunk {inline} inline.bundle.js (inline) 5.79 kB [entry]
chunk {main} main.bundle.js (main) 456 kB [initial]
chunk {polyfills} polyfills.bundle.js (polyfills) 950 kB [initial]
chunk {styles} styles.bundle.js (styles) 651 kB [initial]
chunk {vendor} vendor.bundle.js (vendor) 10.4 MB [initial]

webpack: Compiled successfully.

```

使用浏览器，打开<http://localhost:4200/>链接



注意：

- 1、银行登陆的用户名分别是：xxx1、xxx2、xxx3，密码是：password
- 2、客户登陆的用户名分别是：customer1、customer2、customer3，密码同上
- 3、导入CSV账户信息时，并不是每次都能成功，大家可以更换着银行多试几次。