

test-network-k8s

Fabric, Ahoy!

Hyperledger Fabric Contributor Meeting – October 13, 2021

Josh Kneubuhl - jkneubuh@us.ibm.com

What:

[Test Network](#) for Kubernetes

Why?

- Simple setup : *single action*
- Simple recipe for HLF development : *local = cloud*
- Fabric Kubernetes guide : *lost at sea...*

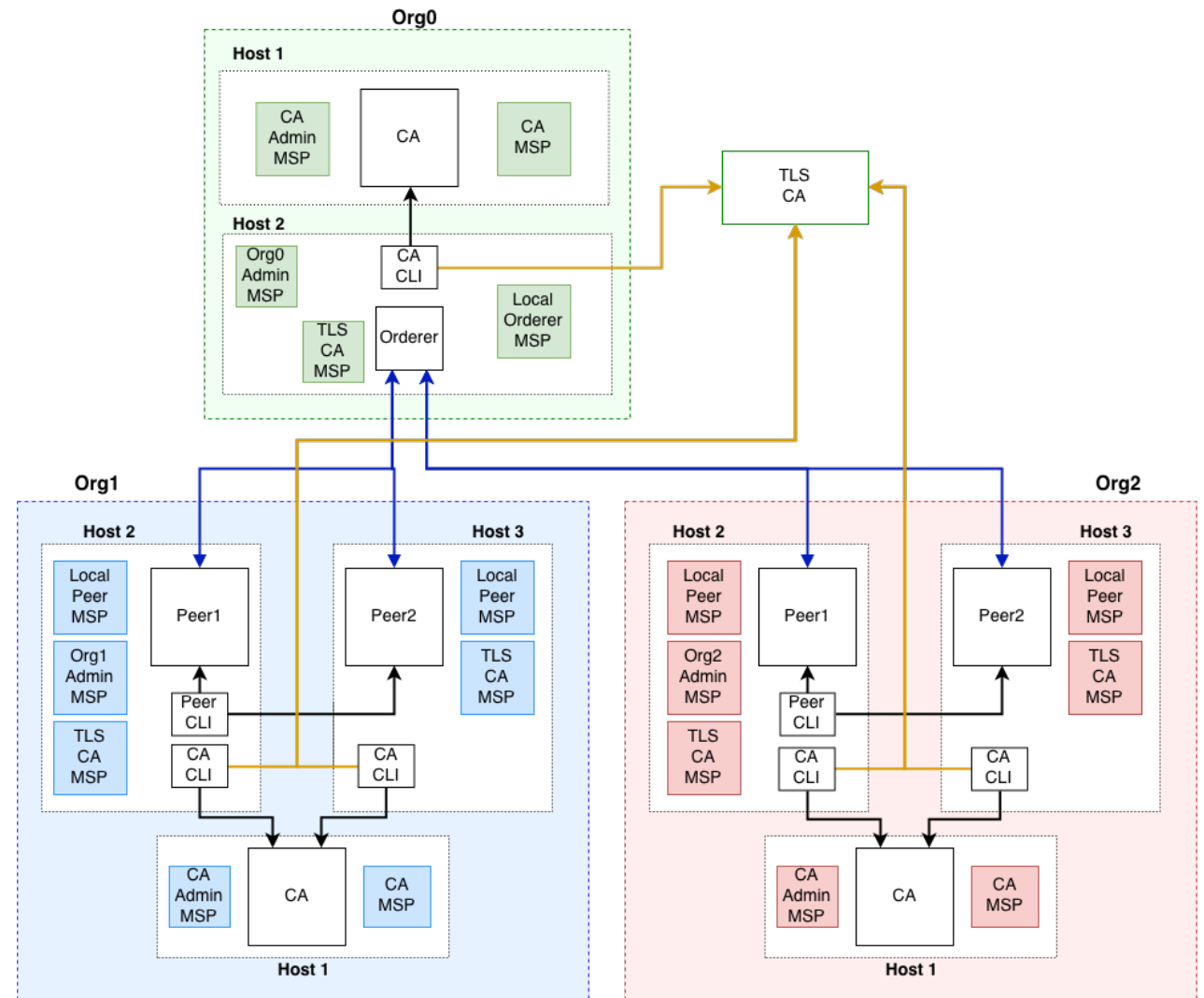
How?

- Everything runs in containers.
- Everything runs on Kubernetes.
- No Kube? No prob - everything runs locally. *KIND*
- Transition to *Chaincode-as-a-Service*. *! docker.d*

Turbulence?

Realistic CA Deployment:

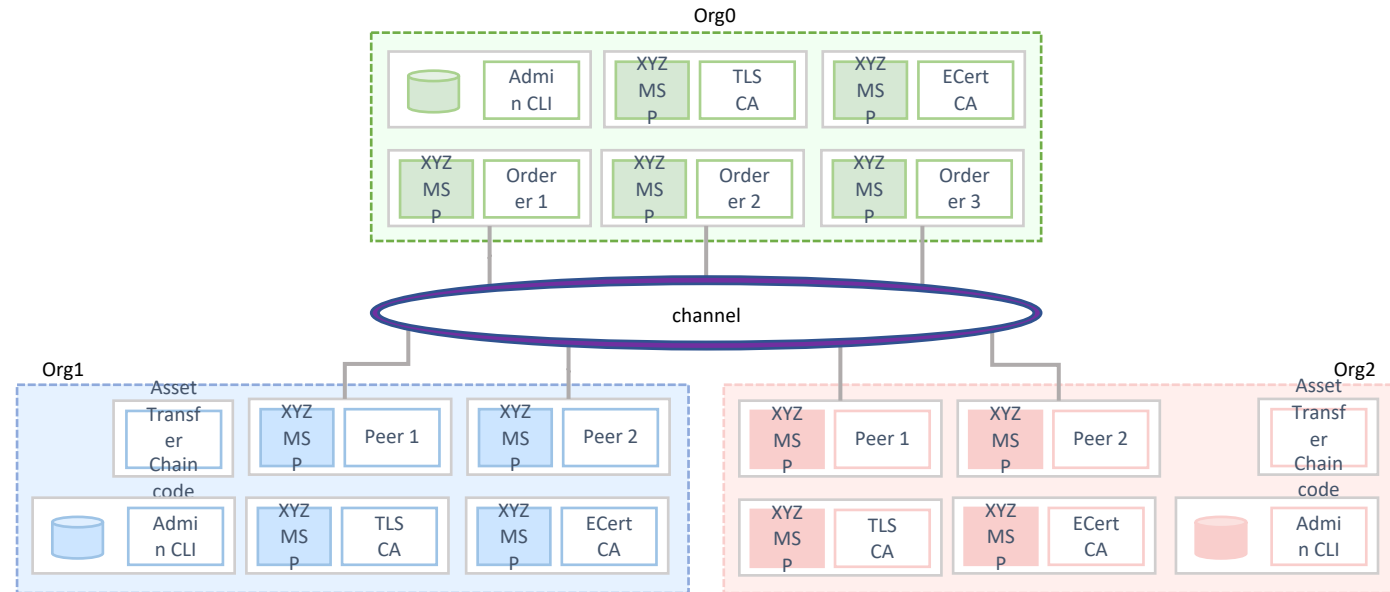
- CA bootstrap is finicky
- fabric-ca-client -> MSP folders
- Consortium Organizers *distribute MSP*
- Includes both TLS and Org CA

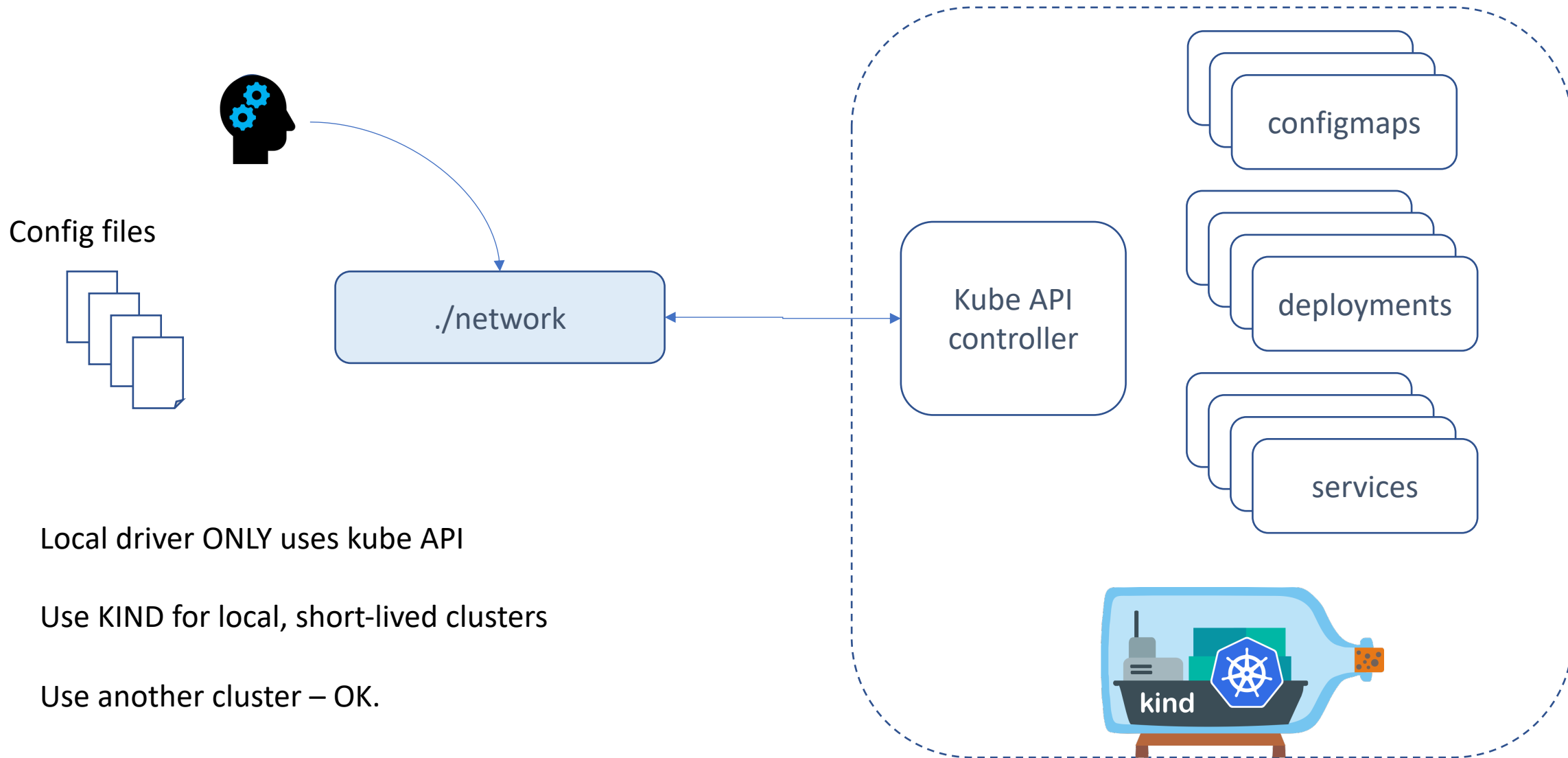


“Noone understands HLF certificates...”

Kube Test Network

- Twelve Enrollments
- Six CAs
- Four Peers
- Three Orgs
- Three Org Volumes
- Three Org Admin CLIs
- Three Orderers
- Two Smart Contracts
- One Channel
- One Sample Client App
- One Gateway Client App





Local driver ONLY uses kube API

Use KIND for local, short-lived clusters

Use another cluster – OK.

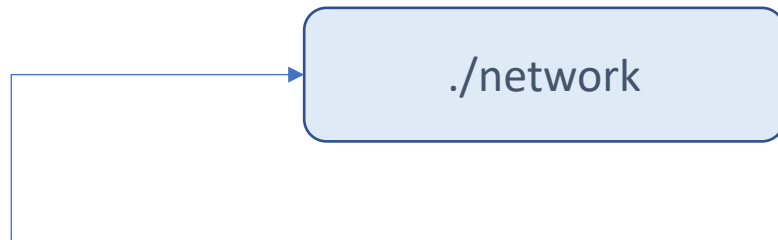


Live Demo

Scenarios:

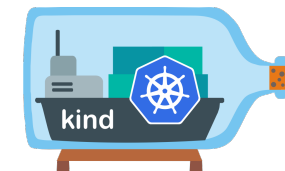
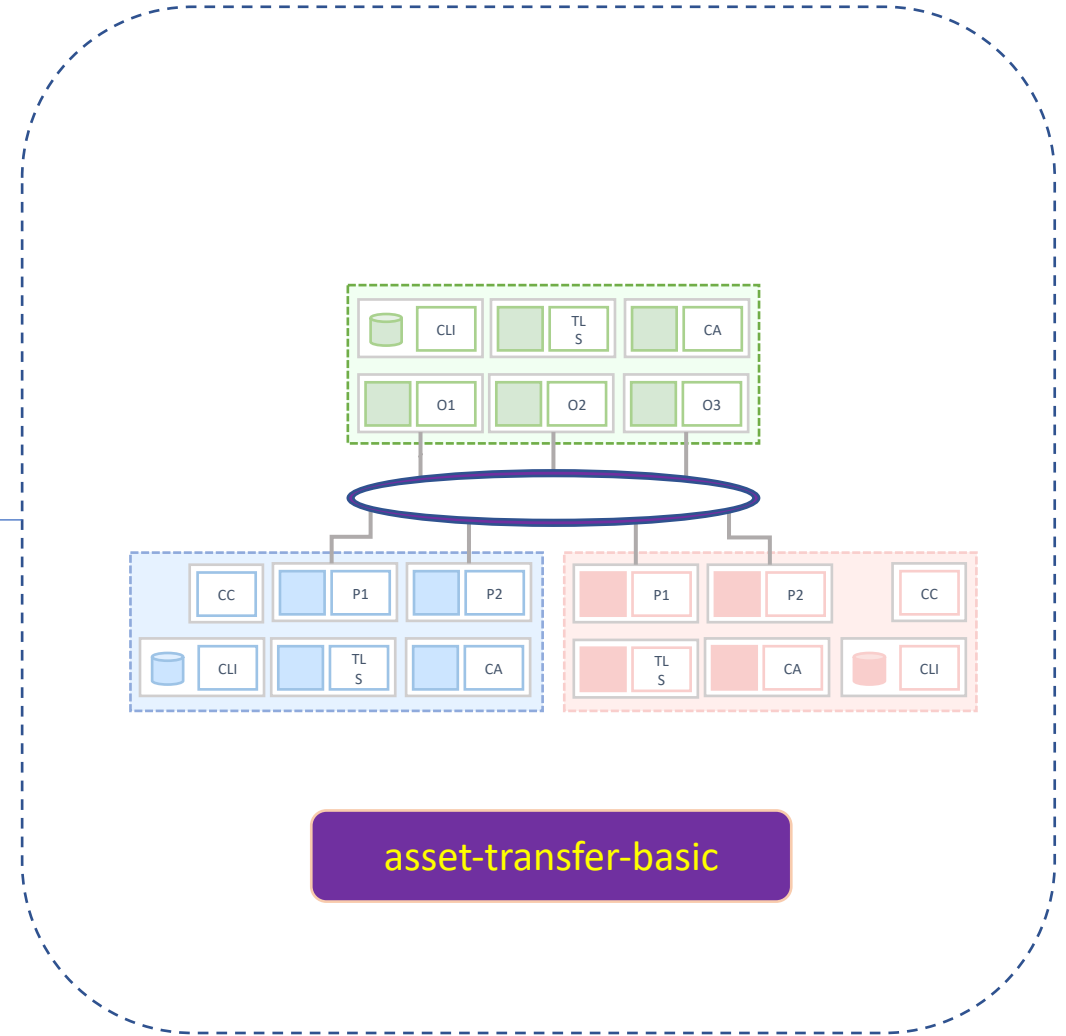
- Test network on KIND
- Fabric Client on KIND
- Custom chaincode
- Debug chaincode!
- Test network on Cloud
- Fabric Operations Console

Test Network on KIND



```
export TEST_NETWORK_FABRIC_VERSION=2.4.0-beta
```

```
./network kind  
./network up  
./network channel create  
./network chaincode deploy  
./network chaincode invoke  
./network chaincode query
```

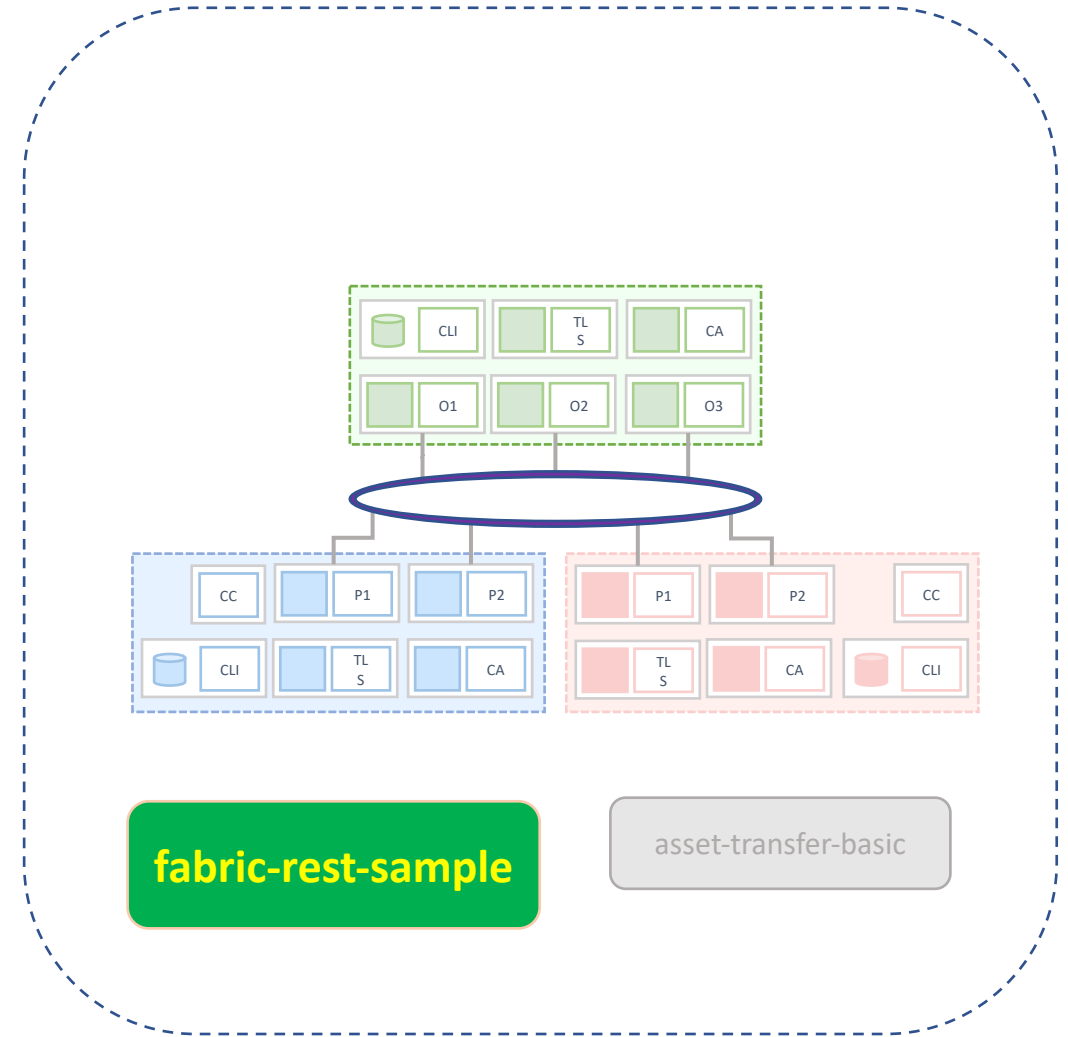


rest-sample on KIND

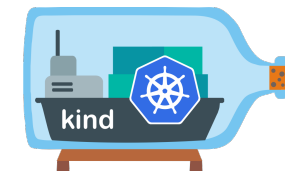
./network

```
./network rest-easy
```

```
export ACCESS_KEY=<guid>  
curl localhost/api/assets
```



NGINX



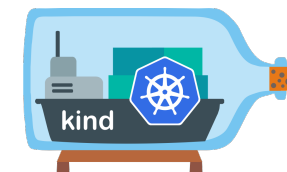
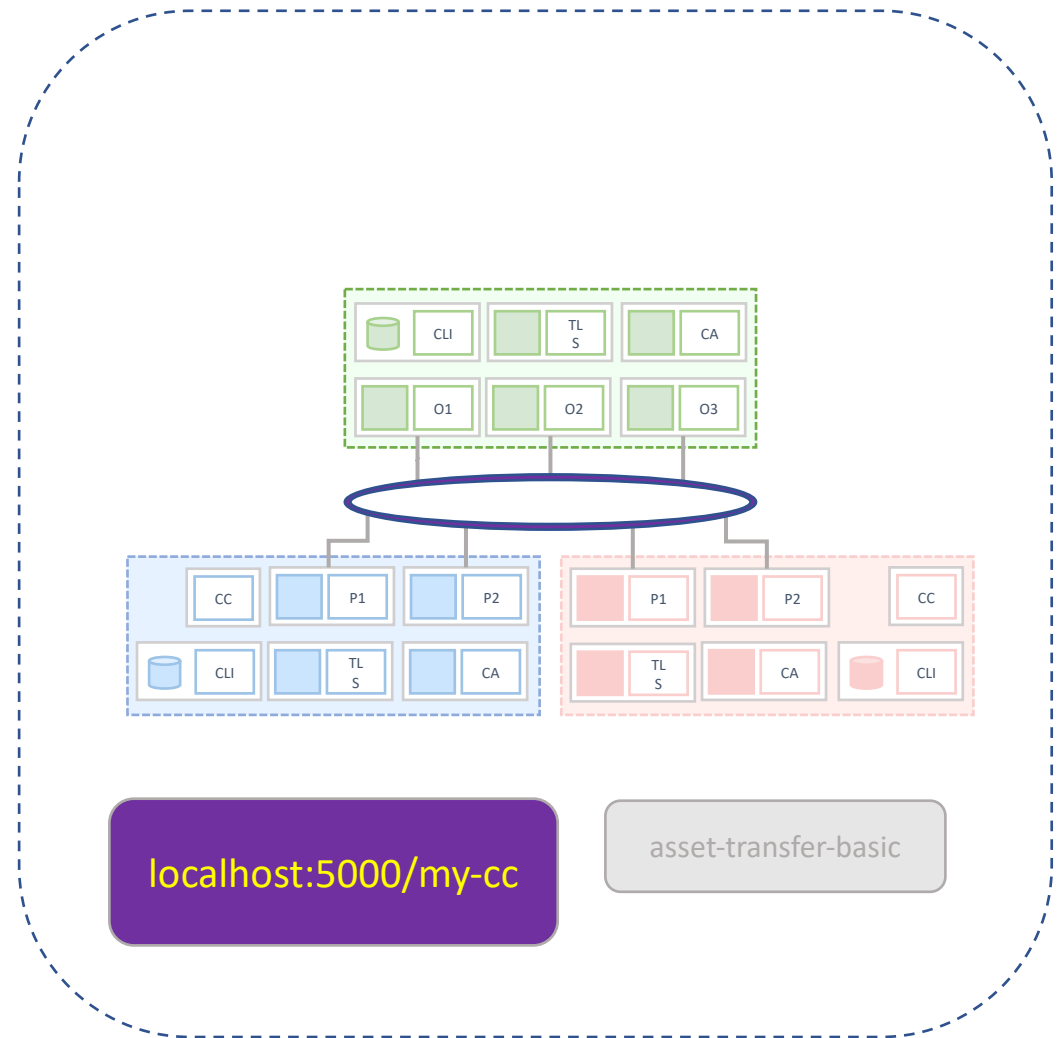
Custom Chaincode

./network

```
export TEST_NETWORK_CHAINCODE_IMAGE=localhost:5000/my-cc
```

```
vi assetTransfer.go  
docker build; docker push
```

localhost:5000 container registry



Debug Chaincode



localhost:5000/asset-transfer-basic

```
export TEST_NETWORK_CHAINCODE_NAME=debug
```

```
vi assetTransfer.go
```

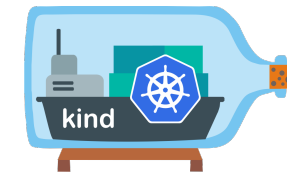
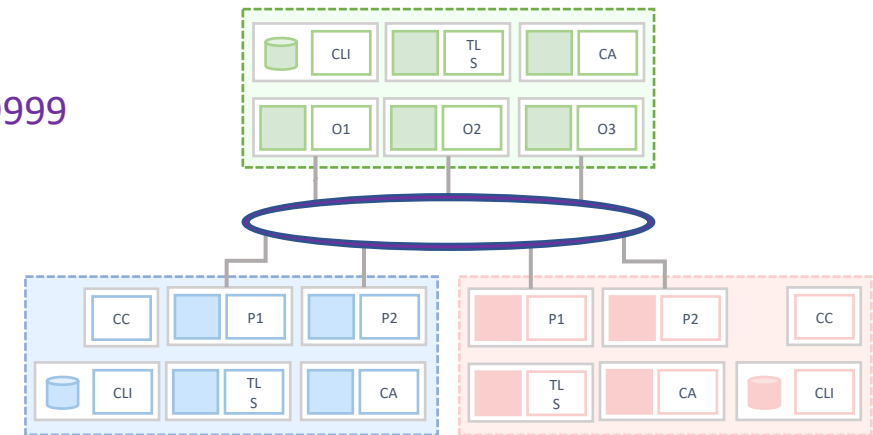
```
docker build
```

```
network chaincode install
```

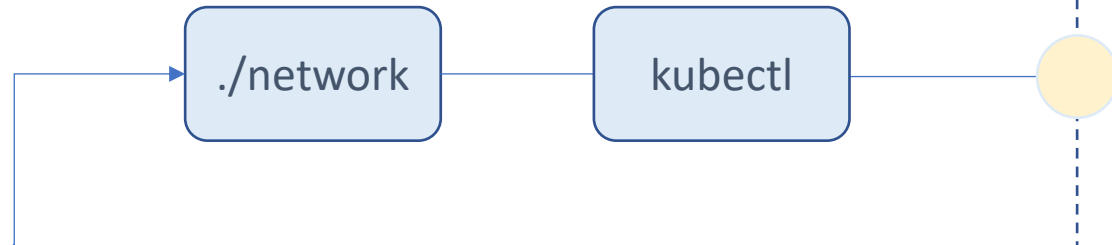
```
docker run -e CHAINCODE_ID -p 9999:9999 my-cc
```

```
network chaincode activate
```

host.docker.internal:9999

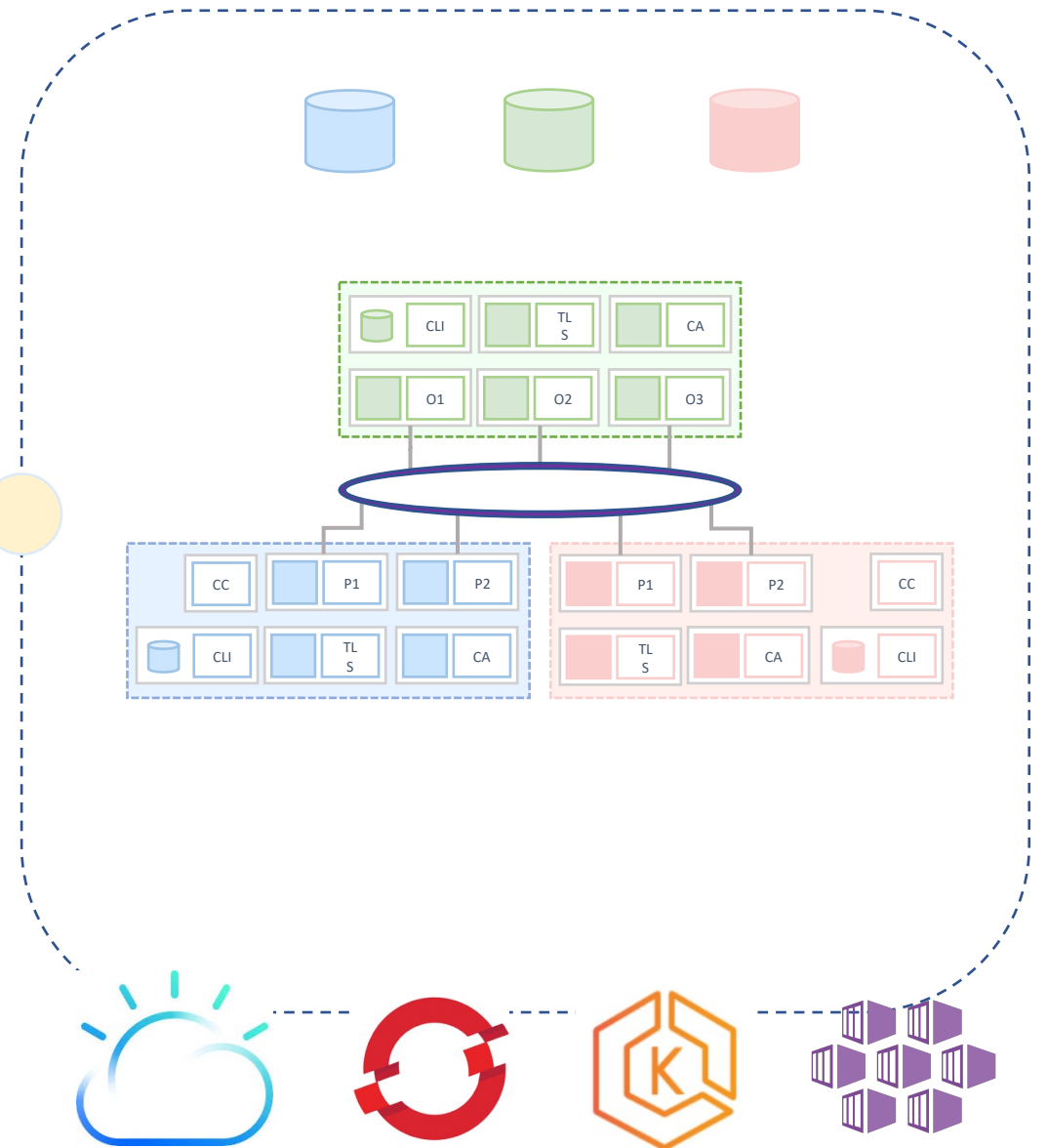


Test Network on Cloud



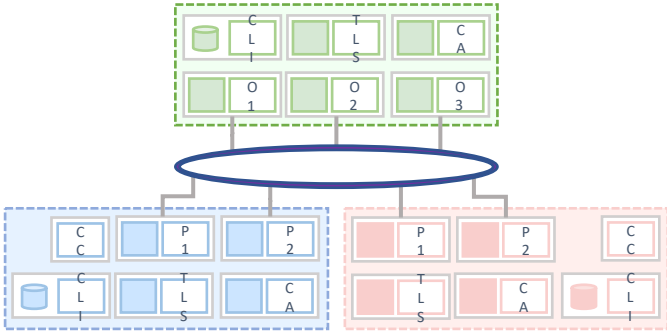
~~./network kind~~

- ./network up
- ./network channel create
- ./network chaincode deploy
- ./network chaincode invoke
- ./network chaincode query



<https://admin:password@console.vcap.me>

Operations Console



```
./network kind
./network up
./network console
```

Fabric Operations Console



Rough Edges

- Needs time to “settle”
- Mutual TLS
- Chaincode TLS
- Intermediate CAs
- Time-bomb Certificates
- Docs, samples, ???

Key Outcomes

- Debug locally ... Deploy to cloud
- Local Dev = CI/CD = Test = Q/A = Prod
- Zero investment clusters
- Predictable, Portable, 100% reproducible:
 - External CC Builders on Kubernetes
 - Cross-Namespace Kube DNS and CSRs
 - *5x orderers*
- Call for community alignment:

+/- secrets, ledger

New patterns (CI/CD, test, ??)

“What’s wrong with XYZ?”

Kubernetes SIG

Discussion + Q/A

test-network-k8s

<https://github.com/hyperledger/fabric-samples/tree/main/test-network-k8s>

feature/fabric-operations-console

<https://github.com/jkneubuh/fabric-samples/tree/feature/fabric-operations-console/test-network-k8s>

fabric-operations-console

<https://github.com/hyperledger-labs/fabric-operations-console>