KubeCon | CloudNativeCon

China 2018

Managing and Securing Blockchain Applications on Kubernetes

Henry Zhang, Chief Architect, VMware China
Yang Yu, Staff Engineer, VMware China

About Us

Henry Zhang

- Chief Architect, VMware China R&D
  - Current focus: Blockchain, Cloud Native Apps, IoT
- Founder of Project Harbor, an open source container registry hosted by CNCF
- Hyperledger Cello Contributor
- Coauthor of two blockchain books (in Chinese)
  - Blockchain Technical Guides
  - Blockchain Core Technologies and Applications

Yang Yu

- Staff Engineer, VMware China R&D
- Working on Kubernetes CNI plugin for VMware NSX transformers
- Familiar with OpenStack's networking component Neutron
- Speaker of KubeCon Europe 2018

Agenda

1. Business Blockchain Overview
2. Why Kubernetes for Blockchain
3. Deploy Fabric on Kubernetes
4. Summary

Business Blockchain Overview

Business Blockchain Requirements

Public Blockchain not for Business

- Bitcoin, Ethereum cannot meet business requirements
  - Confidentiality
  - Slow confirmation
  - Finality
  - Throughput
  - Licensing

- Need a business friendly solution.

Hyperledger

- A Linux foundation project launched in Dec 2015
- 30 founding members
  - Technology: IBM, Intel, Cisco, etc
  - Finance: JP Morgan, Well Fargos, DTCC, etc
  - Blockchain: R3, ConsenSys, DAH, etc.
- 200+ members
- 170+ contributors
- 10K+ commits

Hyperledger Members

Premier Member

General Member

Goals of Hyperledger

- Enterprise class distributed ledger technology based on blockchain
- A cross industry blockchain platform
- Modular, performance and reliability
- Business friendly licensing (Apache V2.0)

Comparison of blockchain projects

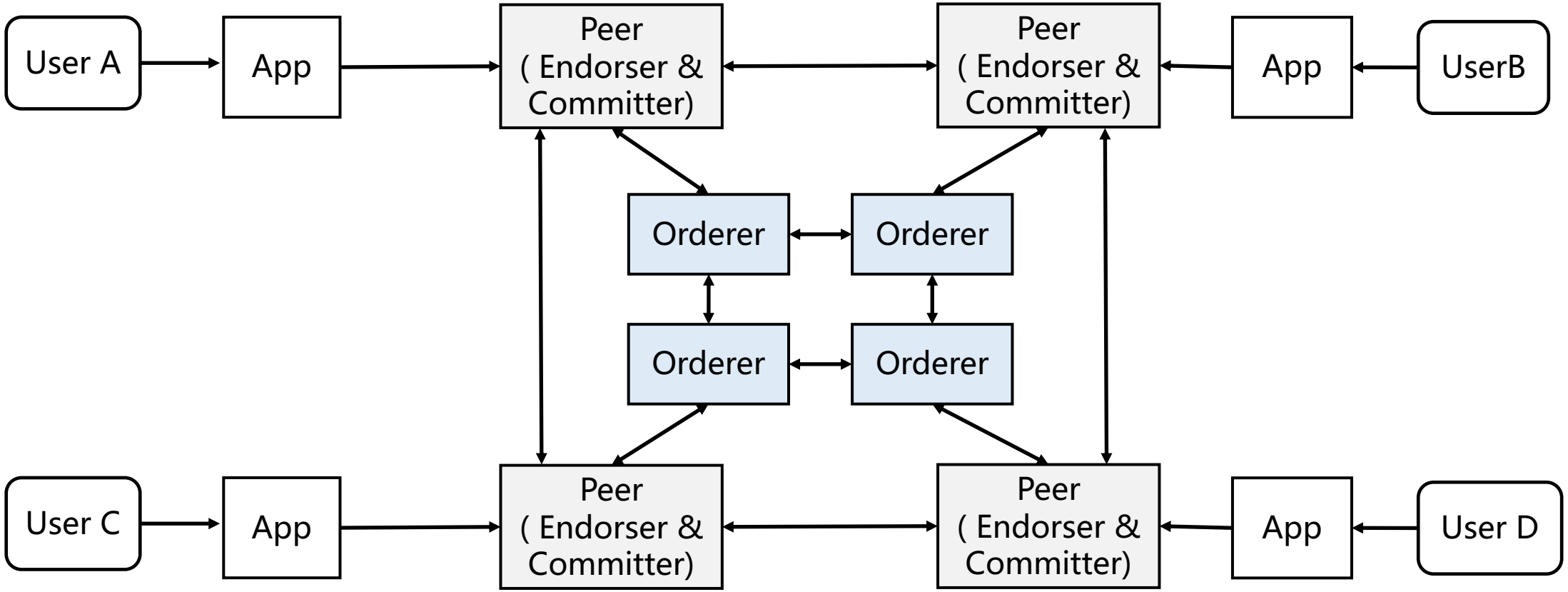
	Hyperledger (Fabric)	Bitcoin	Ethereum
Positioning	Generic permissioned chain	Cryptocurrency	Generic public chain
Management	Linux Foundation	Community	Community
Currency	No	BTC	Ether
Mining	No	Yes	Yes
State data	K/V, document	transaction	account data
Consensus	Solo, Kafka	PoW	PoW, PoS
network	Public or private	public	public
Privacy	Yes	No	No
Smart contracts	Go, Node.js	No	Solidity

Projects of Hyperledger

Hyperledger Fabric

- Open sourced in Dec 2015
- Developed in Golang and used container (Docker)
- Key features:
  - Transaction Confidentiality
  - Access control
  - Dynamic scaling of nodes
  - Better throughput
  - Upgradable smart contracts (chaincode)

Fabric Deployment Model



Some arrows are omitted for simplicity

Why Kubernetes for Blockchain?

What is Kubernetes

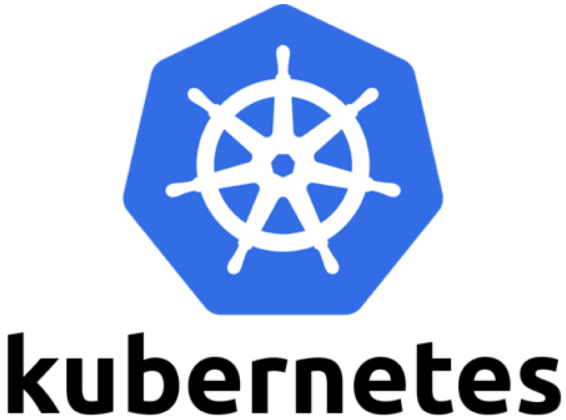
- **Kubernetes,** is an open-source platform for managing, automating deployment, scaling, and operating containerized applications across a cluster of worker nodes.

Capabilities:

- Deploy your applications quickly and predictably
- Scale your applications on the fly
- Seamlessly roll out new features
- Optimize use of your hardware by using only the resources you need

Role:

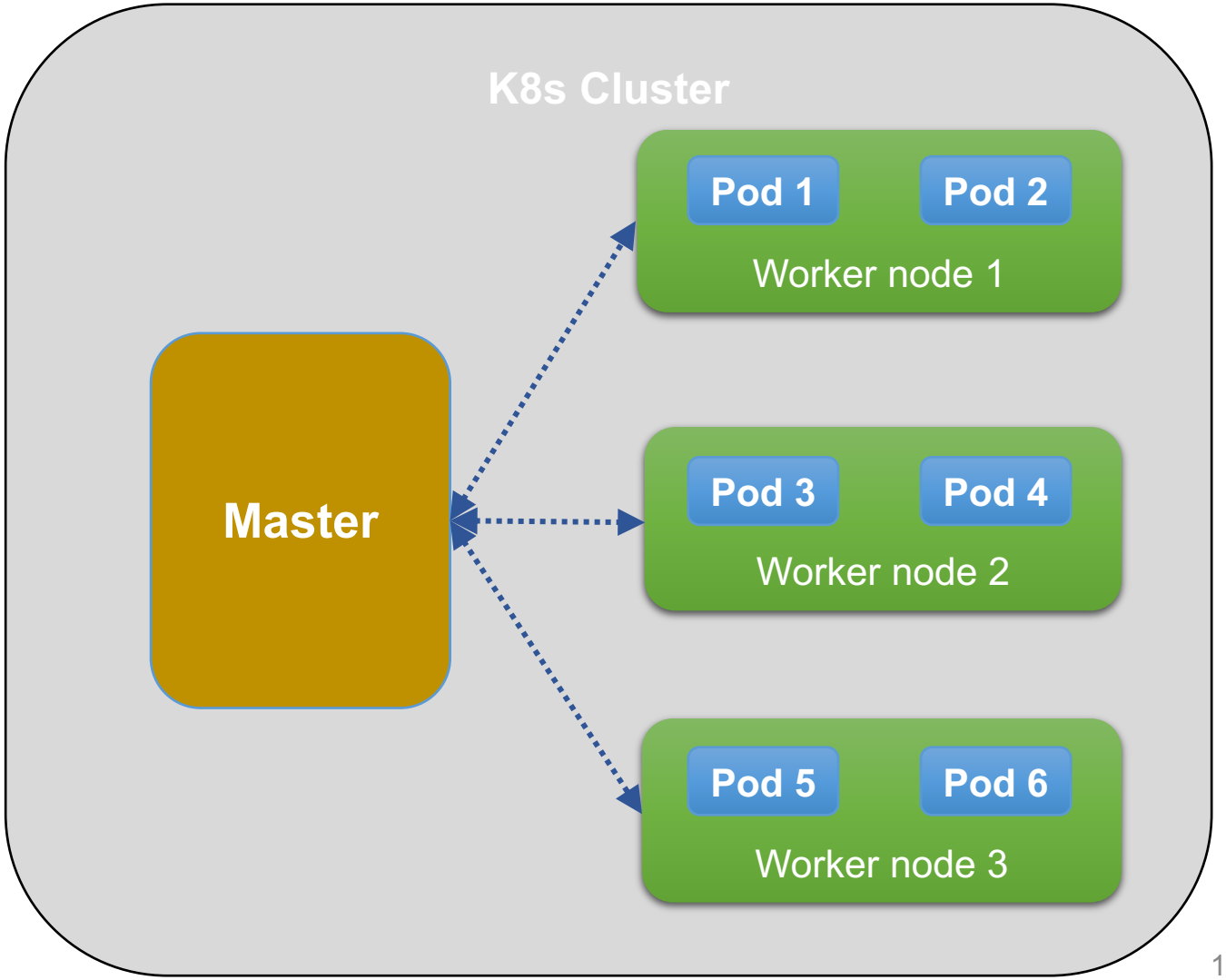
- K8s sits in the Container as a Service (CaaS) or Container orchestration layer



K8s introduces a lot new concepts

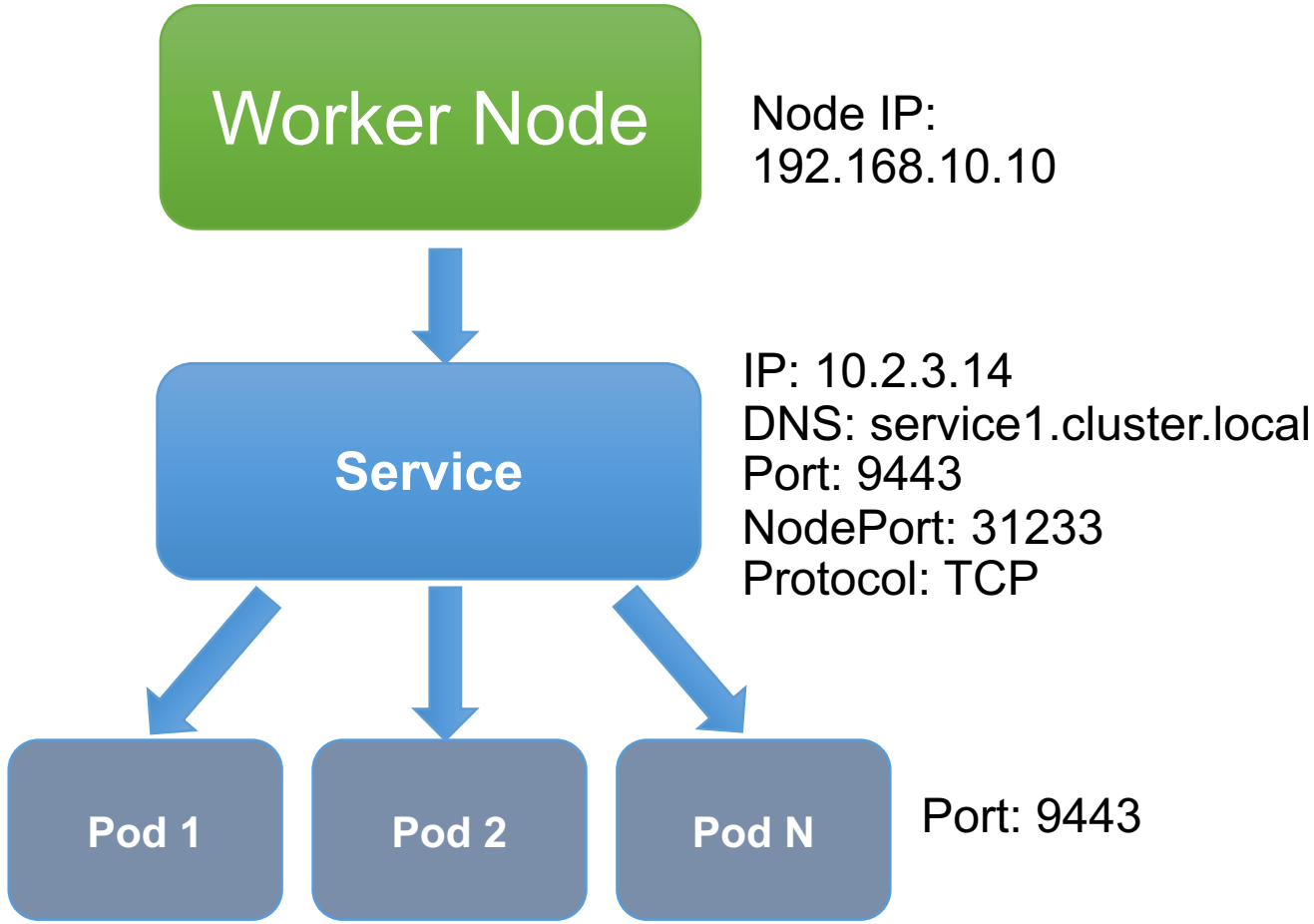
- 10,000 ft. View

- Cluster
- Master
- Workers (nodes)
- Pods



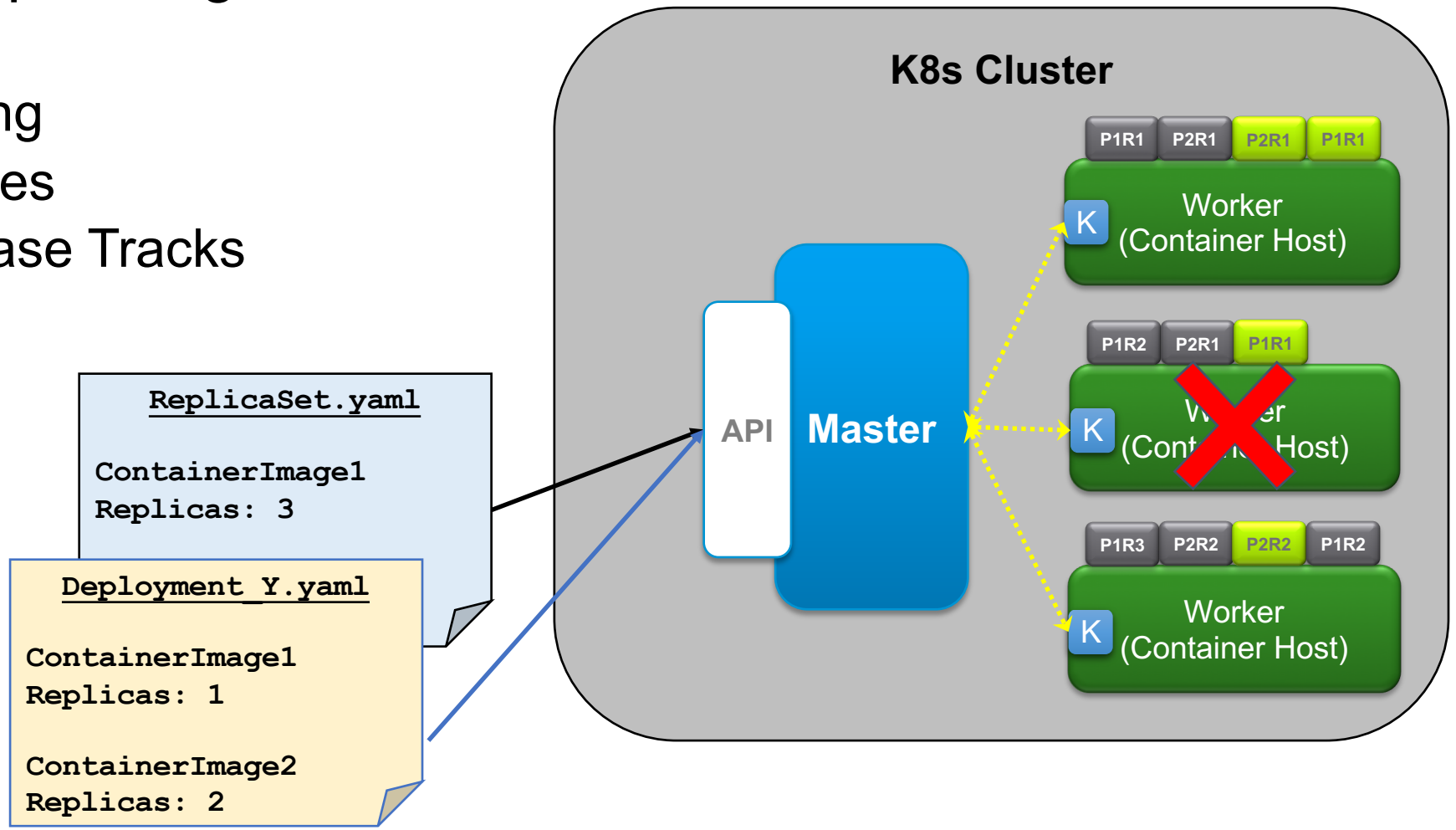
Services

- Services Types
  - ClusterIP
  - NodePort
  - Loadbalancer
- Service Discovery
  - DNS
  - Environmental variables



Replication Controller

- Features for replicating Pods
  - Auto-healing
  - Manual Scaling
  - Rolling Updates
  - Multiple Release Tracks



Challenges of Fabric

- Barriers to use Fabric：
  - Complex configuration
  - Hard to scale out
  - Need to monitor status of nodes and bring up crashed nodes.

- Need an efficient way to manage blockchain

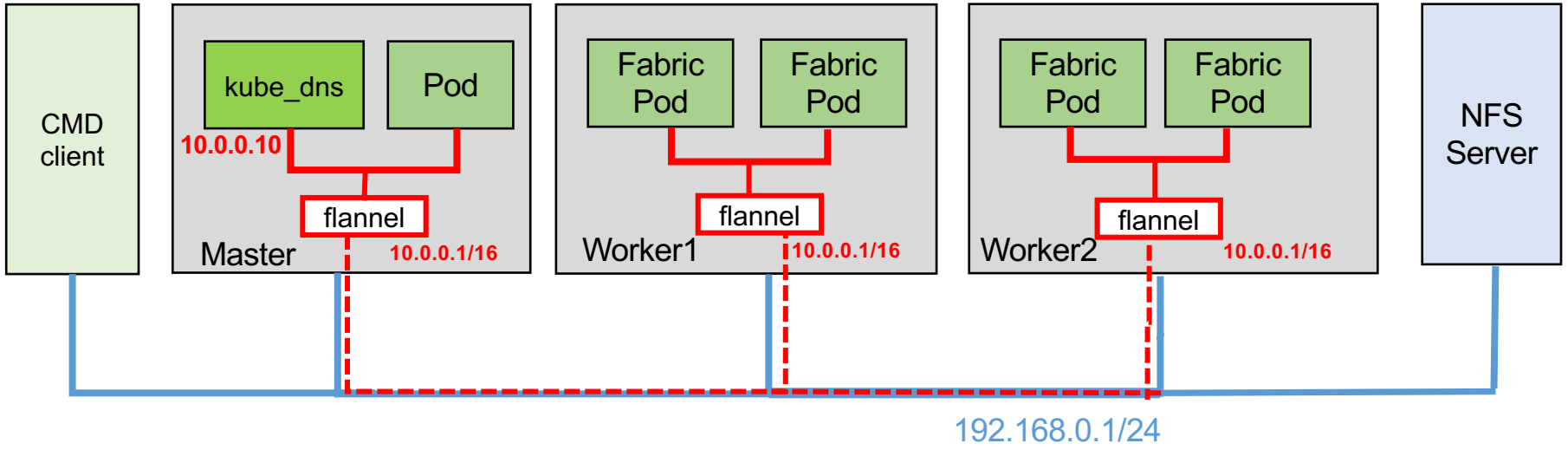
Why Kubernetes?

- Fabric
  - Components are encapsulated in containers
  - Need the flexibility to configure and scale out

- Kubernetes
  - Microservice oriented
  - Easy scaling
  - Tooling for operational management
  - Multi-tenancy support to segregation workload

Deploying Fabric on Kubernetes

Network topology of Fabric Deployment



Network

- All pods are deployed on Kubernetes' overlay network (such as flannel)
- Mapping Kubernetes' namespace to Fabric organization
- Using namespaces to separate components of different organizations
- Applying network policy to enforce isolation between organizations

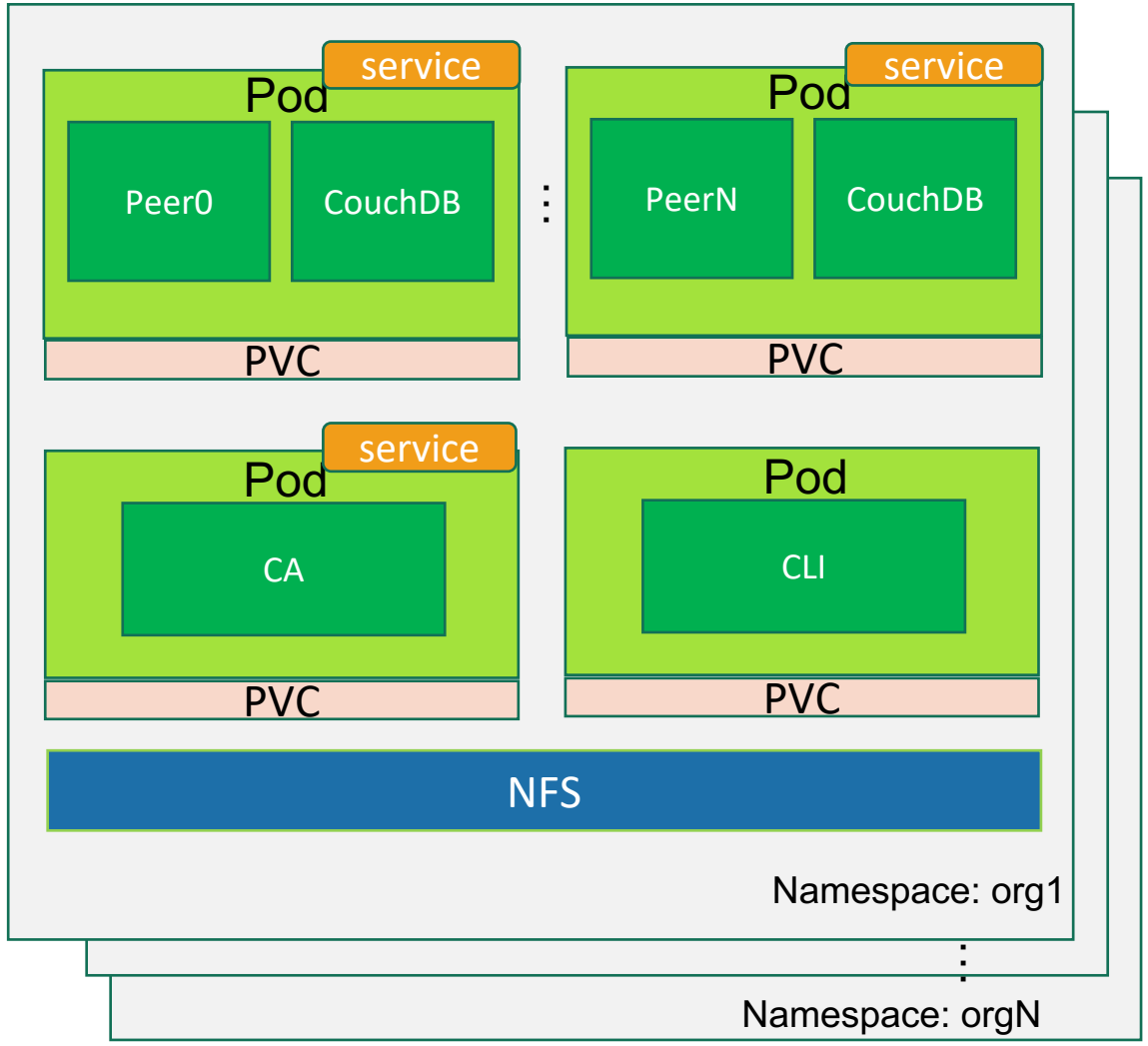
Storage

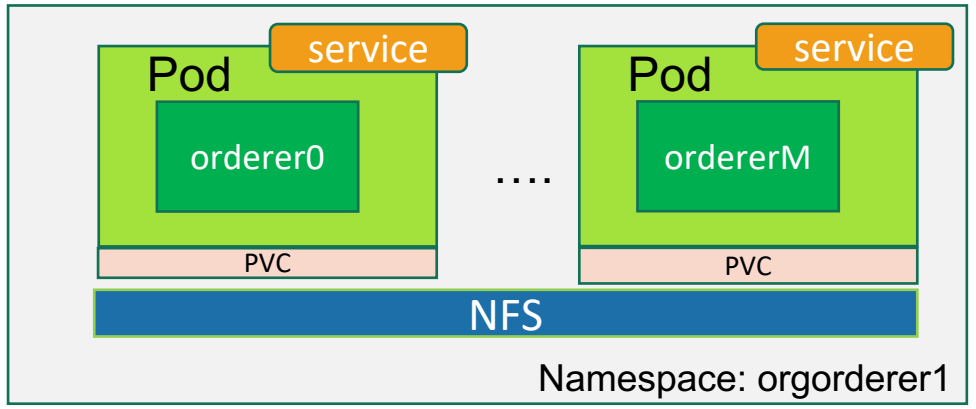
- Configuration and data files are placed in shared storage (NFS)
- Support pod portability between worker nodes
- Use PV and PVC to limit visibility of data for peer nodes of Fabric

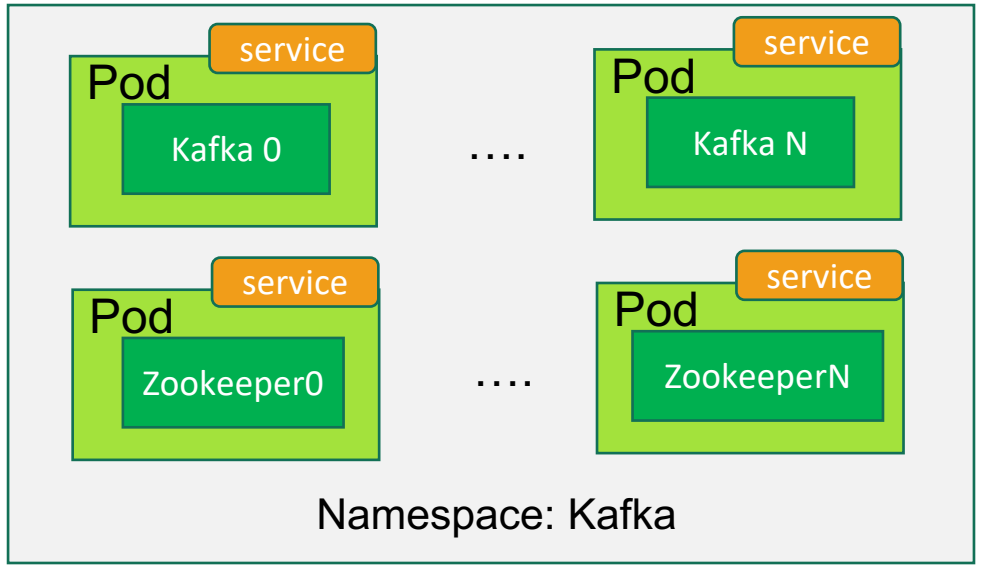
Mapping Fabric Components into Pods

- Peer Pod：Fabric peer，couchDB
- CA Server Pod： Fabric CA Server
- CLI Pod: (Optional) CLI environment for the org
- Orderer Pod: Orderer nodes
- Kafka Pod：Nodes of Kafka
- Zookeeper Pod: Nodes of Zookeeper

Separate components by namespaces







Exposing Services

- Using NodePort for CA、peer and Orderer services
- Port mapping rules :(N>=1, M>=0)
  - Organization orgN port ranges：`30000+(N-1)*100 ~ 30000+(N)*100-1`
  - orgN's CA service port：`ca.orgN:7054 ->  worker:30000+(N-1)*100`
  - Peer M of orgN exposes two ports: 7051, 7052

```
peerM.orgN:7051 -> worker:30000+(N-1)*100 + 2 * M + 1
peerM.orgN:7052 -> worker:30000+(N-1)*100 + 2 * M + 2
```

  - ordererN port：`ordererN:7050 -> worker:33700+N`

Launching Fabric Cluster

- Automate the process by scripts
  - Generate configuration files of Fabric components
  - Generate Pod definitions
  - Create definitions of PV and PVC
  - Start the cluster using kubectl commands

Using Fabric Cluster

- List all pods under namespace of org1 :

```
$ kubectl get pods -namespaces org1
```

```
NAME                                READY     STATUS    RESTARTS   AGE
ca-2708682628-qpz64                 1/1       Running   0          2h
cli-2586364563-vclmr                1/1       Running   0          2h
peer0-org2-3143546256-9prph         2/2       Running   0          2h
peer1-org2-110343575-06pvc          2/2       Running   0          2h
```

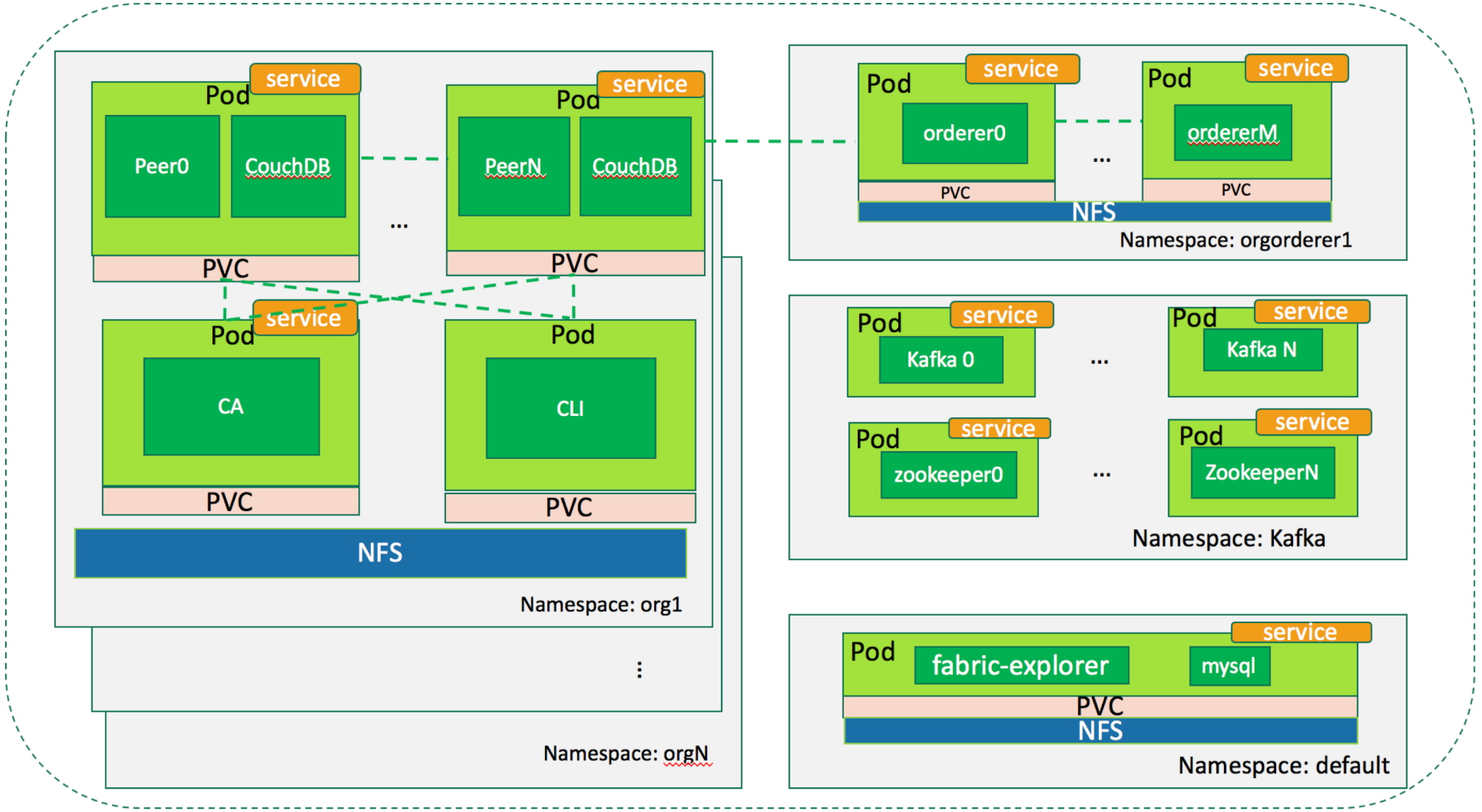
- Enter cli-2586364563-vclmr Pod:

```
$ kubectl exec -it cli-2586364563-vclmr bash --namespace=org1
```

- Run Fabric command, e.g. create a new channel:

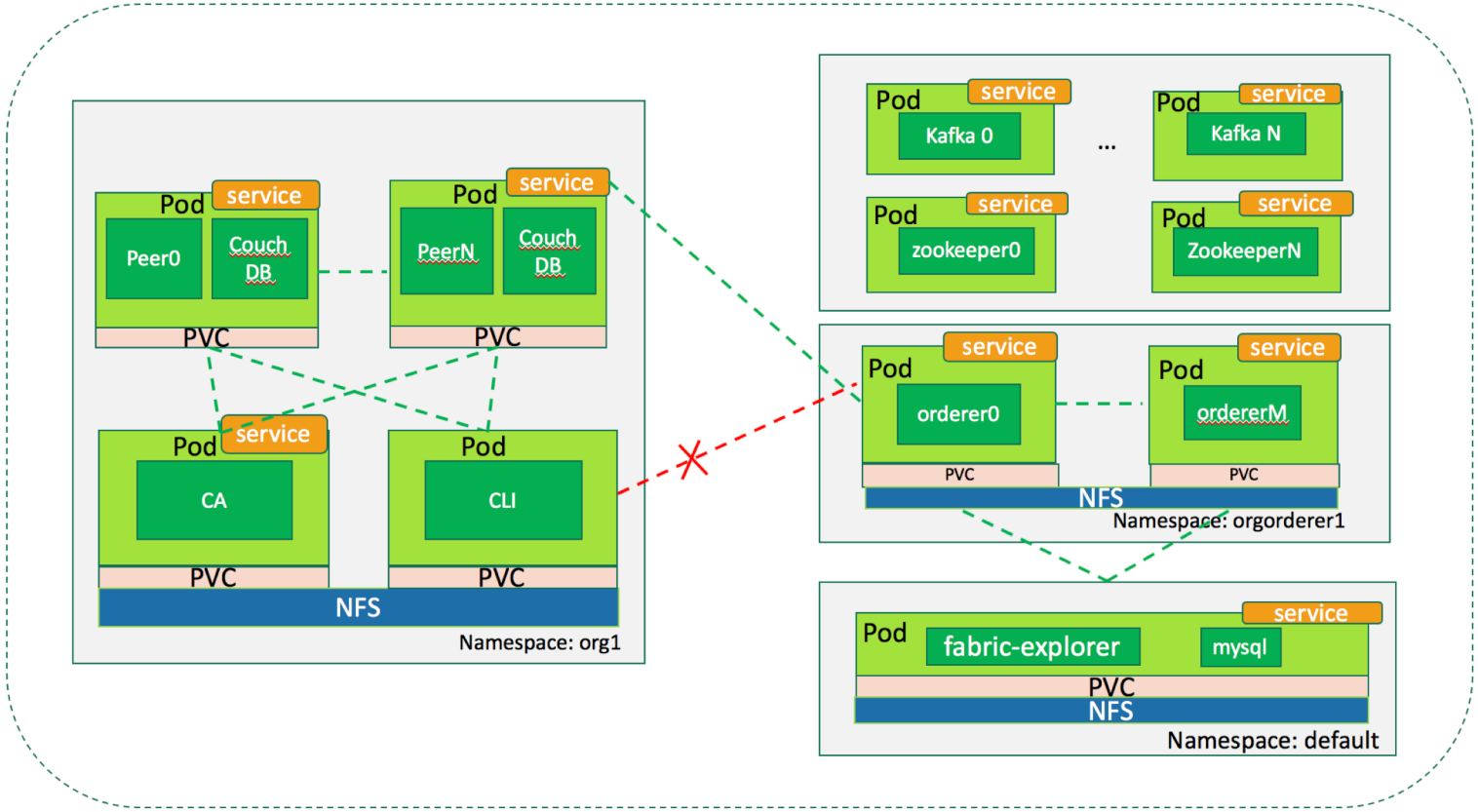
```
$ peer channel create -o orderer0.orgorderer1:7050 \
-c mychannel -f ./channel-artifacts/channel.tx
```

Fabric Network Connection



Network Policy Creation per Namespace

1. Allow service 7051-7053 for peers to access in very organization
   a. CA and CLI pods will not be accessed from outside
   b. Orderer pod could access to peer service
2. Allow service 7050 for Orderer to access in Orderer organization
   a. Only order pod could access to Kafka
   b. Explorer pod could access to order service
3. Allow service explorer for external access



Network Policy Definition Snippet

1. Deny all ingress traffic for peer/orderer/explorer namespace
2. Allow the ingress traffic for specified services in peer/orderer/explorer namespace
3. Allow peer services from the Kubernetes nodes which are hosting peer pods

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
 namespace: ordererorg1-myfabric
 name: deny-order-from-other-
namespaces spec:
 podSelector:
   matchLabels:
 ingress:
 - from:
   - podSelector: {}
```

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
 namespace: ordererorg1-myfabric
 name: orderer-allow-all-
namespaces
spec:
 podSelector:
   matchLabels:
      app: hyperledger
      role: orderer
 ingress:
 - from:
   - namespaceSelector: {}
```

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
 namespace: peerorg1-myfabric
 name: peer-allow-nodes
spec:
 podSelector:
   matchLabels:
      app: hyperledger
      role: peer
 ingress:
 - from:
   - ipBlock:
       cidr:  <k8s-node-ip>
```

Traceflow from Orderer to Peer (1/3)

Traceflow helps inspect the path of a packet as it travels from one logical port to a single or multiple logical ports.

Traffic Type:* Unicast

Source		Destination	
Type	Logical Port	Type	Logical Port
Port*	ff-orderer0-ordererorg1-myfabric-6c5c886c8c	Port*	pks-7ce47543-d24d-4dc9-a3c8-f76303efcff-p
IP Address*	40.0.5.2	IP Address*	40.0.8.4
MAC Address*	02:50:56:00:68:07	MAC Address*	02:50:56:00:68:10

Traceflow from Orderer to Peer (2/3)

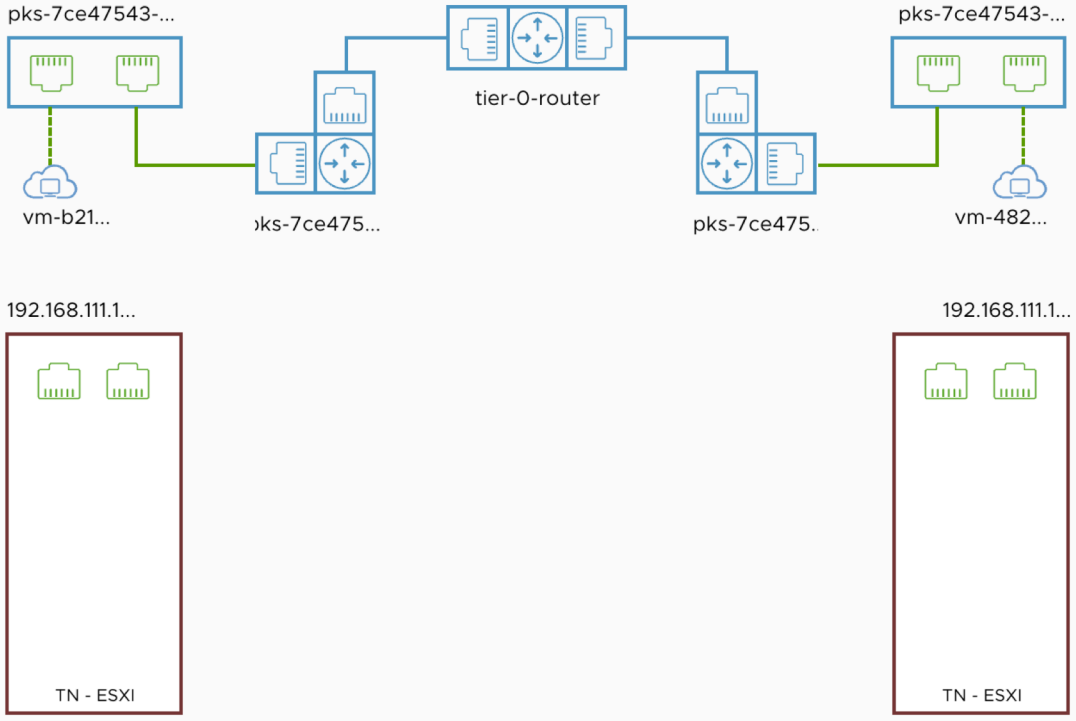
Source		Destination	
Port	pks-7ce47543-d24d-4dc9-a3c8-f76303efcff-orderer0-ordererorg1-myfabric-6c5c886c8c	Port	pks-7ce47543-d24d-4dc9-a3c8-f76303efcff-peer0-peerorg1-myfabric-68bbdf65f8-g5tn7
IP/MAC	40.0.5.2/02:50:56:00:68:07	IP/MAC	40.0.8.4/02:50:56:00:68:10

RE-TRACE

EDIT

NEW TRACE

Trace Results



Show: ALL 0 DELIVERED 1 DROPPED

Physical H	Observation Type	Transport Node	Component
0	Dropped by Firewall: 1484	192.168.111.130	pks-7ce47543-d24d-4dc9-a...

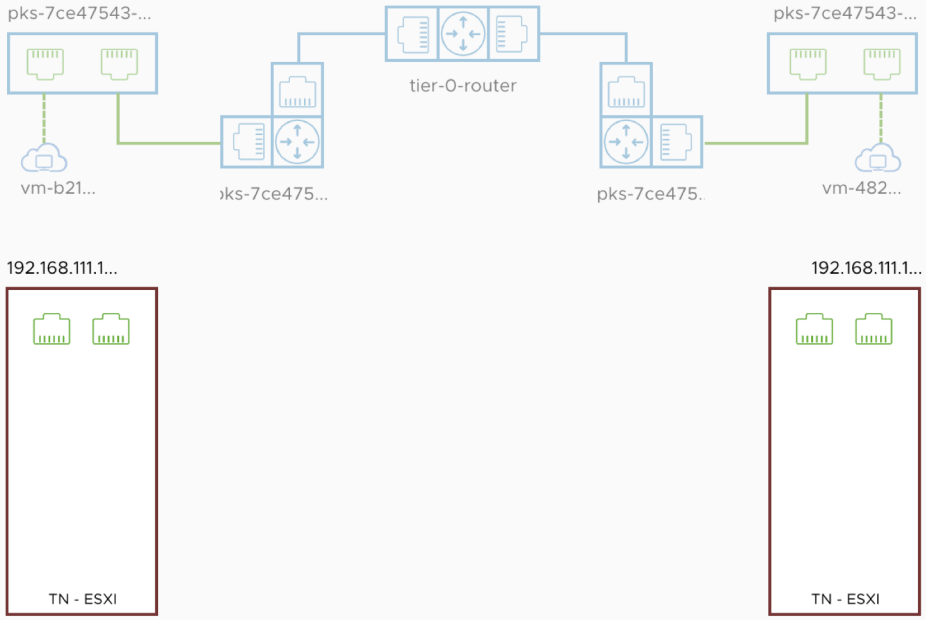
Traceflow from Orderer to Peer (3/3)

Traceflow

Source		Destination	
Port	pks-7ce47543-d24d-4dc9-a3c8-f76303efcff-orderer0-ordererorg1-myfabric-6c5c886c8c	Port	pks-7ce47543-d24d-4dc9-a3c8-f76303efcff-peer0-peerorg1-myfabric-68bbdf65f8-g5tn7
IP/MAC	40.0.5.2/02:50:56:00:68:07	IP/MAC	40.0.8.4/02:50:56:00:68:10

RE-TRACE | EDIT | NEW TRACE

Trace Results



Show: ALL 1 DELIVERED 0 DROPPED

Physical Ho	Observation Type	Transport Node	Component
0	Forwarded	192.168.111.130	pks-7ce47543-d24d-4dc9-a3c8-f763...
0	Forwarded	192.168.111.130	transit-rl-45a4b9af-db75-436f-94fa-...
0	Received	192.168.111.130	tier-0-router
0	Forwarded	192.168.111.130	tier-0-router
0	Forwarded	192.168.111.130	transit-rl-37cc9c05-661e-44b1-a088-...
0	Received	192.168.111.130	pks-7ce47543-d24d-4dc9-a3c8-f763...
0	Forwarded	192.168.111.130	pks-7ce47543-d24d-4dc9-a3c8-f763...
0	Received	192.168.111.130	pks-7ce47543-d24d-4dc9-a3c8-f763...
0	Received	192.168.111.130	Distributed Firewall
0	Forwarded	192.168.111.130	Distributed Firewall (Rule ID: 1505)
0	Delivered	192.168.111.130	pks-7ce47543-d24d-4dc9-a3c8-f763...

Summary

- Kubernetes is an important platform to run Blockchain as a Service (BaaS)
- Deploying and operating Hyperledger Fabric on Kubernetes simplifies the management.
- Helm chart could be used to deploy Fabric on Kubernetes (see below blog)

- Download: https://labs.vmware.com/flings/blockchain-on-kubernetes

- Source code and blogs：
  - https://github.com/hainingzhang/articles
  - http://www.think-foundry.com/deploy-hyperledger-fabric-on-kubernetes-part-1/
  - http://www.think-foundry.com/hyperledger-fabric-deployment-using-helm-chart/

KubeCon | CloudNativeCon

China 2018