Configuring and Managing Kubernetes Networking, Services, and Ingress

KUBERNETES NETWORKING FUNDAMENTALS

Kien Bui
DevOps & Platform Engineer

Course Overview

Kubernetes Networking Fundamentals

Configuring and Managing Application Access with Services

Configuring and Managing Application Access with Ingress

Summary

Kubernetes network model

Network topology

Pod networking Internals

Container Network Interface -(CNI)

Cluster DNS

Kubernetes Networking Model

All Pods can communicate with each other on all Nodes

Agents on a Node can communicate with all Pods on that Node

No Network Address Translation (NAT)

Motivations for the Network Model

Simplicity

Hide Implementation Details

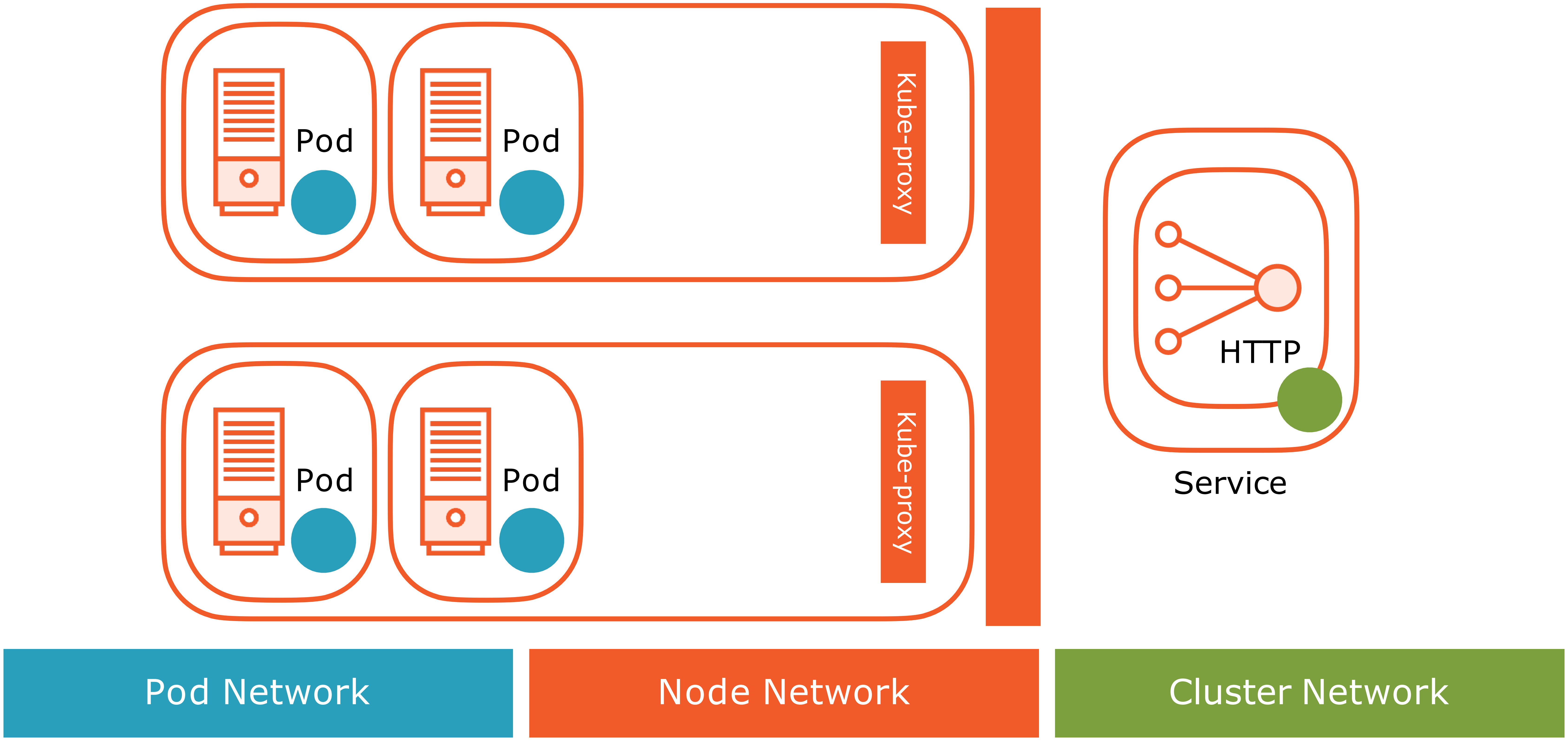
Administrator Controlled

Define in Code

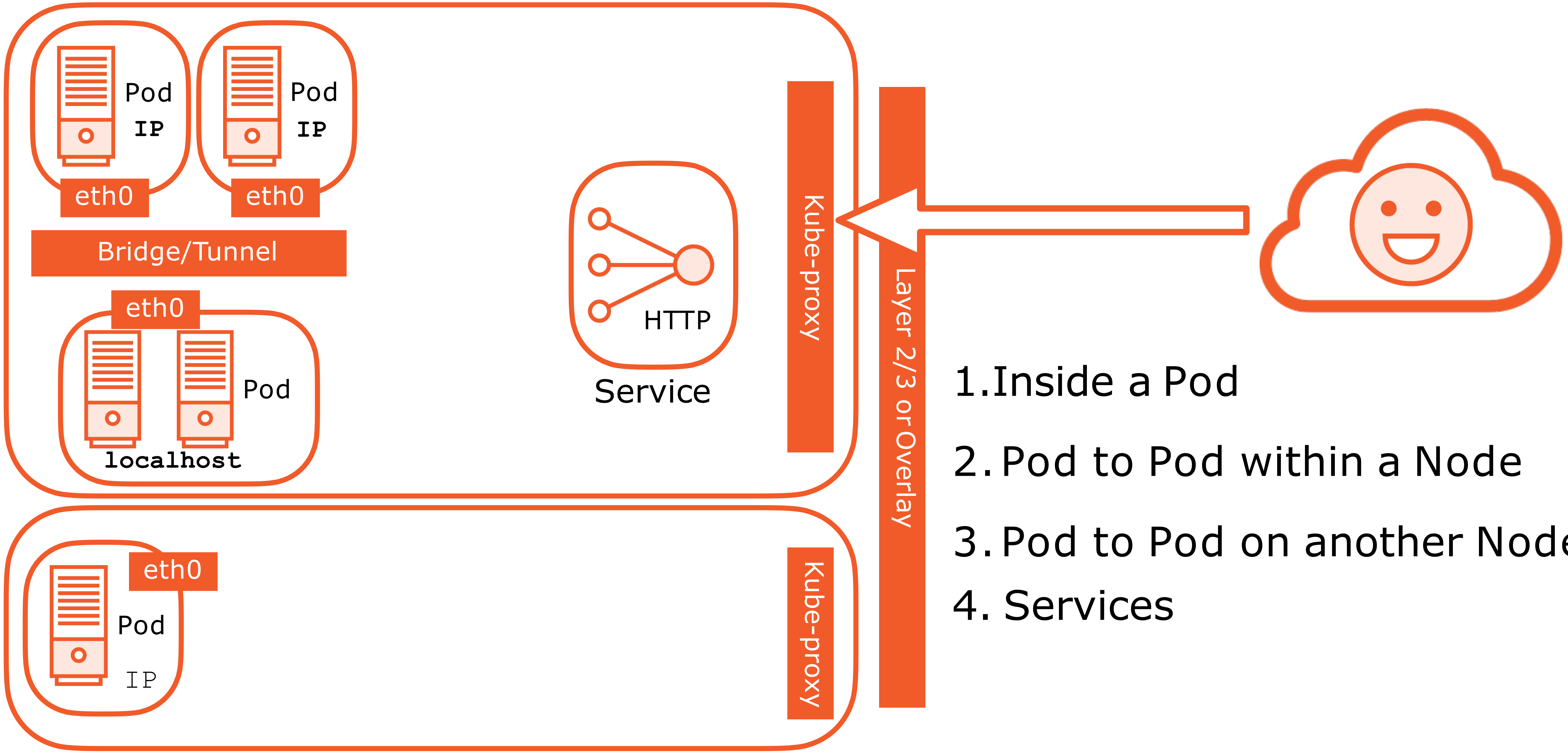
All Pods can communicate to each other

Service Discovery and App Configuration

Kubernetes Network Topology



Pod Networking and Communication



1. Inside a Pod
2. Pod to Pod within a Node
3. Pod to Pod on another Node
4. Services

Pod Networking Internals

Pod share a network namespace

Containers in a Pod communicate over `localhost`

Pause/Infrastructure container

- Starts the networking namespace
- If the application container restarts the network will persist
- Lifecycle of the Pod

Container Network Interface - CNI

Standardized Networking

CNI Plugins

Pod

Implements Kubernetes Network Model

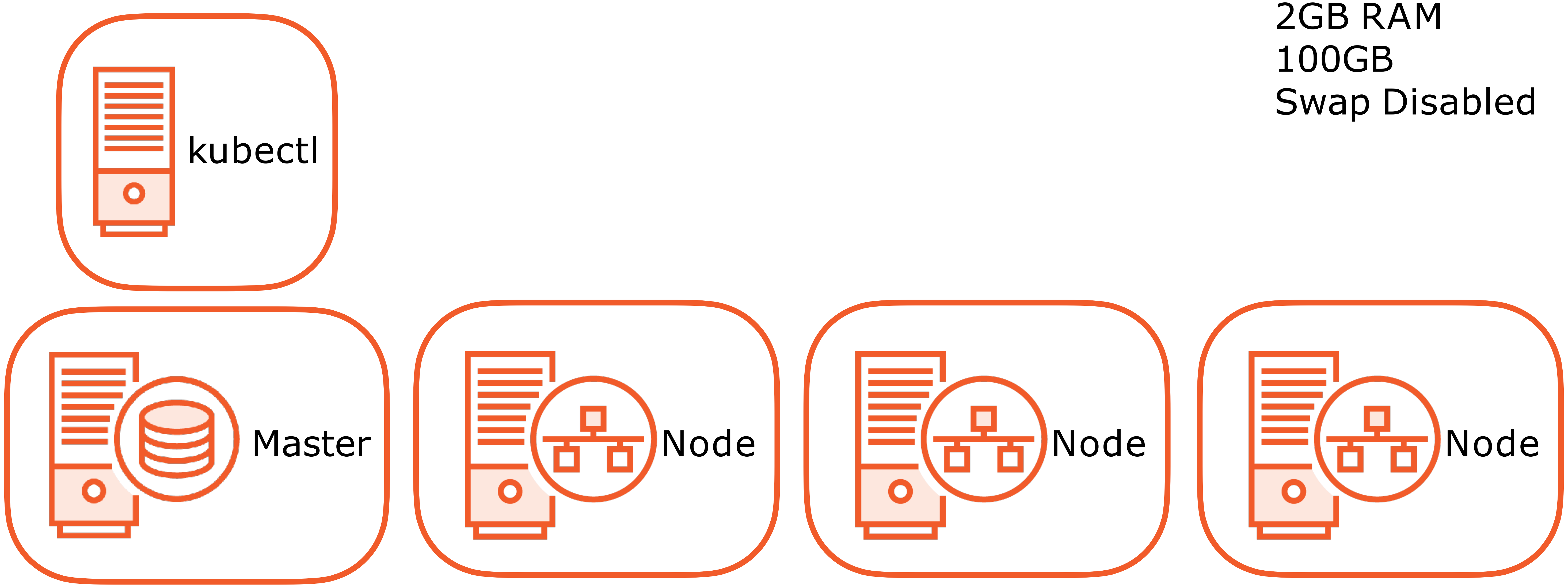
Network Plugin

https://kubernetes.io/docs/concepts/cluster-administration/networking/

Lab Environment

Hostnames set
Host file on each

Ubuntu 16.0.4
VMware Fusion VMs
2vCPU
2GB RAM
100GB
Swap Disabled



c1-master1	**c1-node1**	**c1-node2**	**c1-node3**
172.16.94.10	172.16.94.11	172.16.94.12	172.16.94.13

Kubernetes Installation and Configuration Fundamentals

Demo

Investigating Kubernetes Networking

- Local Cluster - Calico CNI Plugin
- Azure Kubernetes Service -kubenet

Cluster DNS

DNS is available as a Service in a Cluster

Pods are configured to use this DNS

DNS records

- Services - A/AAAA records
- Namespaces - subdomains

Core to Service discovery

Customize both the DNS Service and Pods configuration

Configuring Cluster DNS - Configuring a Forwarder

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: coredns
  namespace: kube-system
data:
  Corefile: |
    .:53 {
        ...
        kubernetes cluster.local in-addr.arpa ip6.arpa {
           pods insecure
           fallthrough in-addr.arpa ip6.arpa
           ttl 30
       }
       forward . /etc/resolv.conf
       ...
    }
```

https://coredns.io/manual/toc/

Configuring Pod DNS - Specifying DNS Servers

```
...
    spec:
      containers:
      - name: hello-world
        image: gcr.io/google-samples/hello-app:1.0
        ports:
        - containerPort: 8080
      dnsPolicy: "None"
      dnsConfig:
        nameservers:
          - 9.9.9.9
        searches:
          - db1.ns1.svc.cluster.local
```

Demo

Investigating the Cluster DNS Service

Configuring CoreDNS to use custom Forwarders

Configuring Pod DNS Configuration

Review

Kubernetes network model

Network topology

Pod networking Internals

Container Network Interface -(CNI)

Cluster DNS

Up Next:
Configuring and Managing Application Access with Services