# Installing and Configuring Kubernetes

**Anthony E. Nocentino**
ENTERPRISE ARCHITECT @ CENTINO SYSTEMS
@nocentino www.centinosystems.com

# Course Overview

Introduction

Exploring Kubernetes Architecture

Installing and Configuring Kubernetes

Working with Your Kubernetes Cluster

# Overview

Installation Considerations

Installation Overview

Getting Kubernetes

Installing a Cluster With `kubeadm`

Creating a Cluster in the Cloud

# Installation Considerations

**Where to install?**

- **Cloud**
  - **IaaS - Virtual Machines**
  - **PaaS - Managed Service**
- **On-Prem**
  - **Bare Metal**
  - **Virtual Machines**
- **Which one should you choose?**

**https://kubernetes.io/docs/setup/pick-right-solution/**

# Installation Considerations (con’t)

Cluster Networking

Scalability

High Availability

Disaster Recovery

# Installation Methods

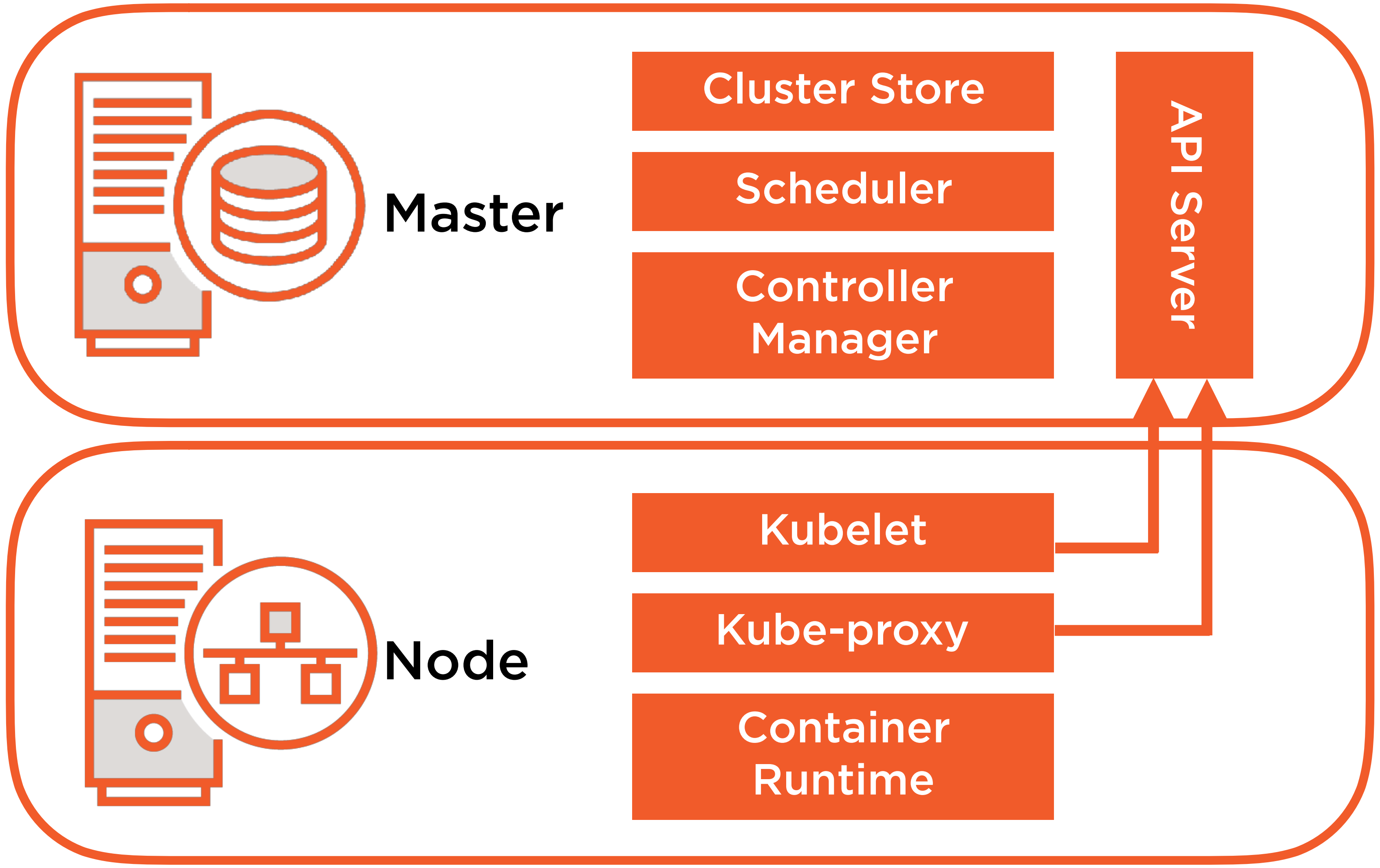
|  |  |
|---|---|
| Desktop | kubeadm |
| From Scratch | Cloud Scenarios |

https://kubernetes.io/docs/setup/scratch/
https://github.com/kelseyhightower/kubernetes-the-hard-way/

# Installation Requirements

| System Requirements | Container Runtime | Networking |
|---|---|---|
| Linux - Ubuntu/CentOS | Container Runtime Interface (CRI) | Connectivity between all Nodes |
| 2 CPUs | Docker | |
| 2GB RAM | | |
| Swap Disabled | | |

# Cluster Network Ports



| Component | Ports (tcp) | Used By |
|---|---|---|
| API | 6443 | All |
| etcd | 2379-2380 | API/etcd |
| Scheduler | 10251 | Self |
| Controller Manager | 10252 | Self |
| Kubelet | 10250 | Control Plane |

| Component | Ports (tcp) | Used By |
|---|---|---|
| Kubelet | 10250 | Control Plane |
| NodePort | 30000-32767 | All |

# Getting Kubernetes

Maintained on GitHub

https://github.com/kubernetes/kubernetes

Linux Distribution Repositories

yum and apt

# Building Your Cluster

**Install Kubernetes**

**Create Your Cluster**

**Configure Pod Networking**

**Join Nodes to your Cluster**

# Required Packages

kubelet

kubeadm

kubectl

Container Runtime - Docker

**Install on all Nodes in your cluster**

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF

apt-get update
apt-get install -y kubelet kubeadm kubectl docker.io
apt-mark hold kubelet kubeadm kubectl docker.io
```

# Getting and Installing Kubernetes on Ubuntu VMs

**Do this on all nodes**

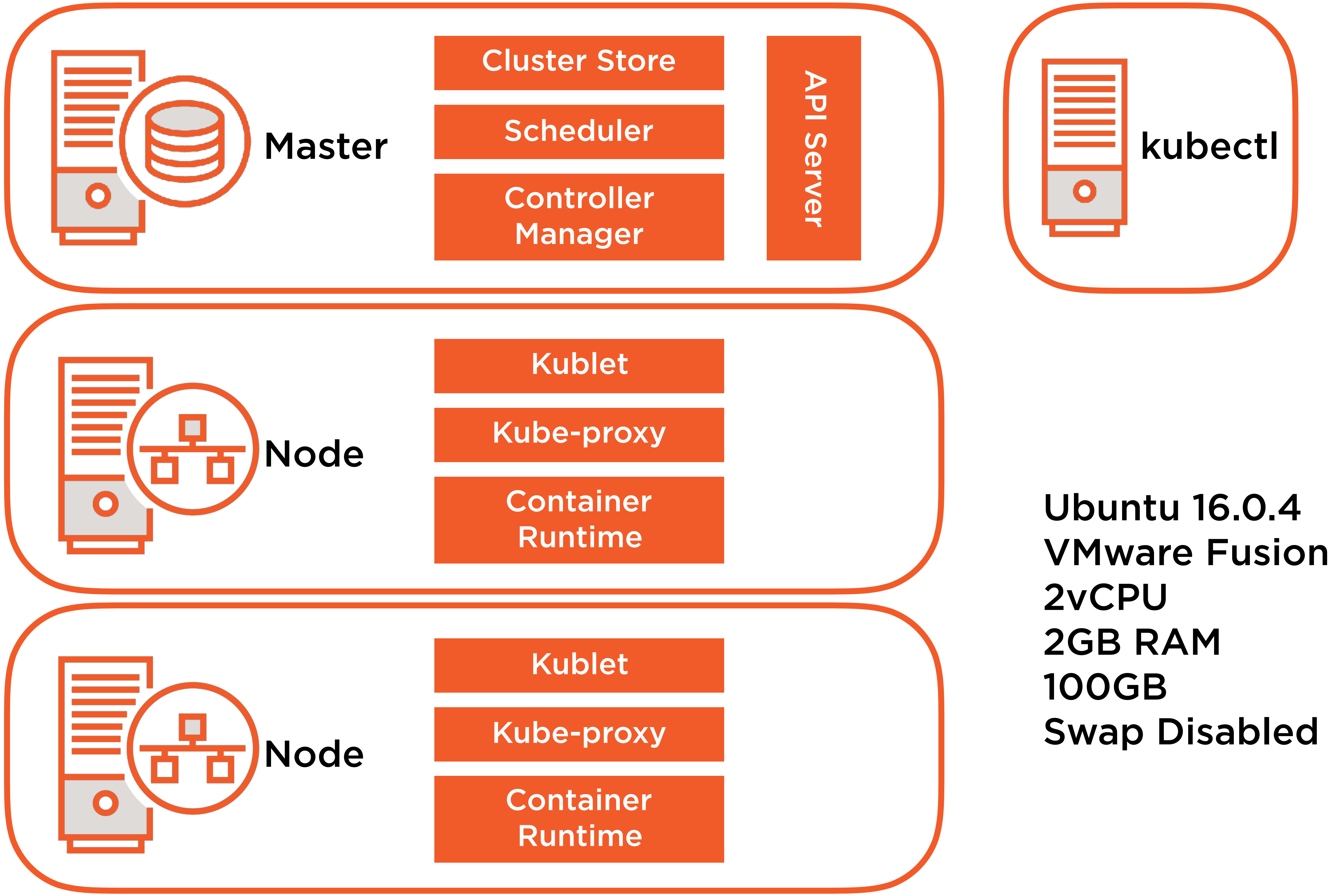
# Lab Environment

Hostnames set
Host file on each

c1-master1
172.16.94.10

c1-node1
172.16.94.11

c1-node2
172.16.94.12



Master

Cluster Store

Scheduler

Controller Manager

API Server

kubectl

Node

Kublet

Kube-proxy

Container Runtime

Node

Kublet

Kube-proxy

Container Runtime

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

# Demo

Installing

- kubelet
- kubeadm
- kubectl
- docker

systemd Units

Process is customizable

# tstrapping a Cluster with kubeadm

`kubeadm init`

Pre-flight checks

Creates a Certificate Authority

Generates kubeconfig files

Generates Static Pod Manifests

Starts up the Control Plane

Taints the Master

Generates a Bootstrap Token

Starts Add-On Pods: DNS and kube-proxy

# Certificate Authority

**Self signed CA**

**Can be part of an external PKI**

**Securing cluster communications**

- **API Server**

**Authentication of users and kubelets**

- `/etc/kubernetes/pki`

**Distributed to each Node**

https://kubernetes.io/docs/reference/setup-tools/kubeadm/kubeadm-init/#external-ca-mode

# kubeadm Created kubeconfig Files

**Used to define how to connect to your Cluster**

- **Certificate information**
- **Cluster location**

`/etc/kubernetes`

- `admin.conf` **(kubernetes-admin)**
- `kubelet.conf`
- `controller-manager.conf`
- `scheduler.conf`

# Static Pod Manifests

**Manifest describes a configuration**

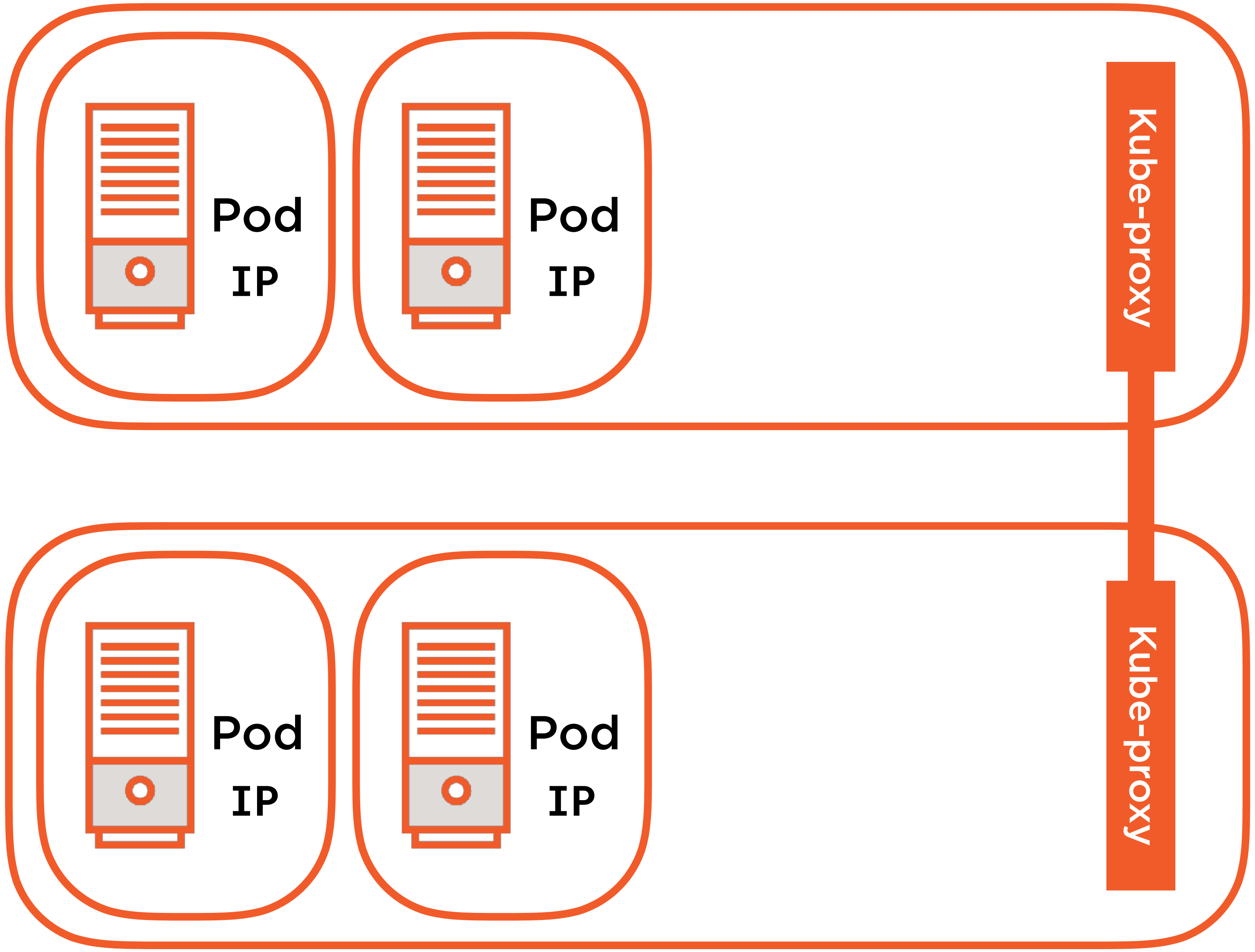
`/etc/kubernetes/manifests`

- **etcd**
- **API Server**
- **Controller Manager**
- **Scheduler**

**Monitored by the kubelet and started automatically when the system starts and over time**

**Enable the startup of the cluster...without the cluster**

# Pod Networking



Single, un NATed IP address per Pod

Direct routing - configure infrastructure to support IP reachability between Pods on Nodes

Overlay networking

- Flannel - Layer 3 virtual network
- Calico - L3 and policy based traffic management
- Weave Net - multi-host Docker network

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

```
wget https://docs.projectcalico.org/v3.3/getting-started/kubernetes/
installation/hosted/rbac-kdd.yaml
wget https://docs.projectcalico.org/v3.3/getting-started/kubernetes/
installation/hosted/kubernetes-datastore/calico-networking/1.7/calico.yaml

kubeadm init --pod-network-cidr=192.168.0.0/16

kubectl apply -f rbac-kdd.yaml
kubectl apply -f calico.yaml
```

# Creating a master

```
Your Kubernetes master has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

You can now join any number of machines by running the following on each node
as root:
  kubeadm join 172.16.94.10:6443
   --token i0pr88.pbid2af0071xhuo1
   --discovery-token-ca-cert-hash sha256:9a56f13bbae1f77e3a01fecc2bf8c59e6977d9c71c2d3482b988fa47767353d7
```

# Adding a Node to a Cluster

`kubeadm join`

Download Cluster Information

Node submits a CSR

CA Signs the CSR automatically

Configures `kubelet.conf`

# Demo

Creating a Cluster

systemd Units...again!

Static manifests

Joining Nodes to a Cluster

# Managed Cloud Deployment Scenarios

**Elastic Container Service for Kubernetes (EKS)**

**https://aws.amazon.com/getting-started/projects/deploy-kubernetes-app-amazon-eks/**

**Google Kubernetes Engine (GKE)**

**https://cloud.google.com/kubernetes-engine/docs/how-to/**

**Azure Kubernetes Services (AKS)**

**https://docs.microsoft.com/en-us/azure/aks/kubernetes-walkthrough**

# Demo

Creating a Cluster in the Cloud

- Azure Kubernetes Services (AKS)
- Google Kubernetes Engine (GKE)

# Summary

Installation Considerations

Installation Overview

Getting Kubernetes

Installing a Cluster With `kubeadm`

Creating a Cluster in the Cloud

# What’s Next!

## Working With Your Cluster