

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-Premises

Bare Metal

Virtual Machines

Which one should you choose?

Installation Considerations (con't)



Cluster Networking

Scalability

High Availability

Disaster Recovery

Installation Methods



Desktop



kubeadm

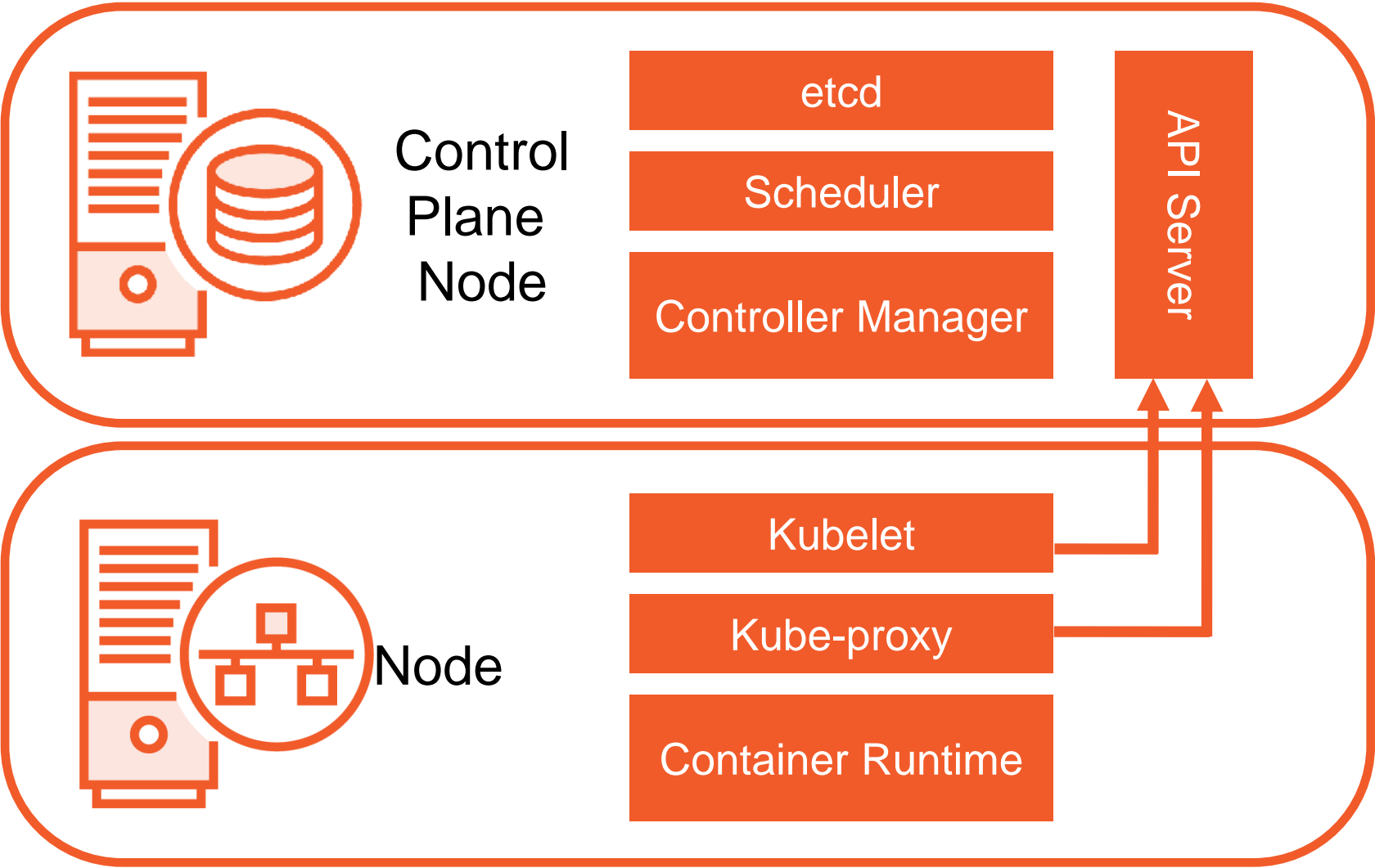


Cloud Scenarios

Installation Requirements

System Requirements	Container Runtime	Networking
Linux - Ubuntu/RHEL	Container Runtime Interface (CRI)	Connectivity between all Nodes
2 CPUs	containerd	Unique hostname
2GB RAM	Docker (Deprecated 1.20)	Unique MAC address
Swap Disabled	CRI-O	

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
Kubelet	10250	Control Plane
NodePort	30000-32767	All

NodePort: exposes our services and ports on each individual node in our cluster

Getting Kubernetes

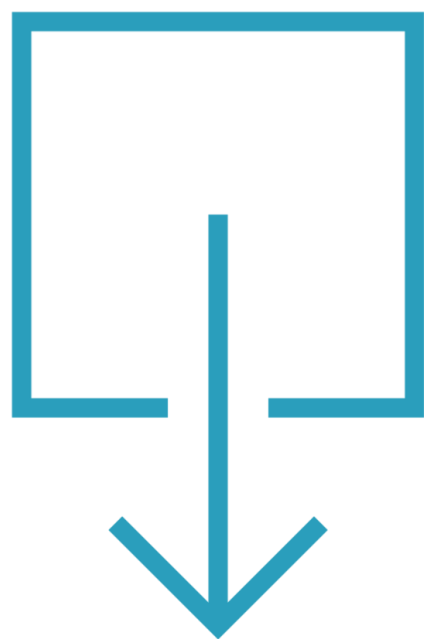
Maintained on GitHub

<https://github.com/kubernetes/kubernetes>

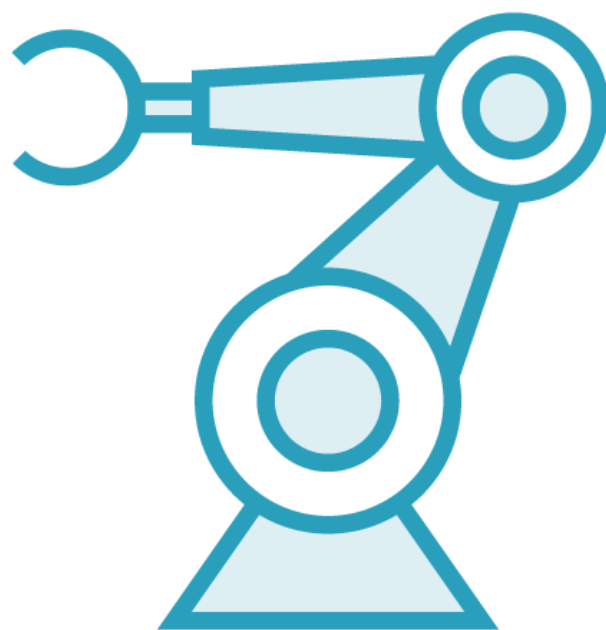
Linux Distribution Repositories

yum and apt

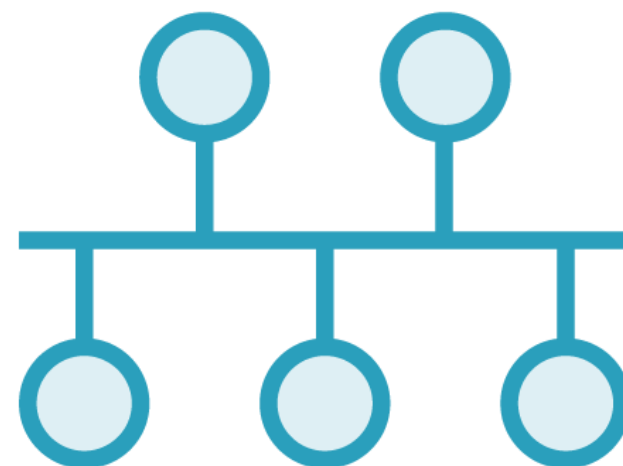
Building Your Cluster



Install and Configure
Packages



Create Your Cluster



Configure Pod
Networking



Join Nodes to Your
Cluster

Required Packages



containerd



kubelet



kubeadm

bootstrapping control plane and nodes



kubectl

Install on all Nodes in your cluster

Getting and Installing Kubernetes on Ubuntu VMs

```
sudo apt-get install -y containerd
```

```
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
```

```
apt-mark hold kubelet kubeadm kubectl containerd
```

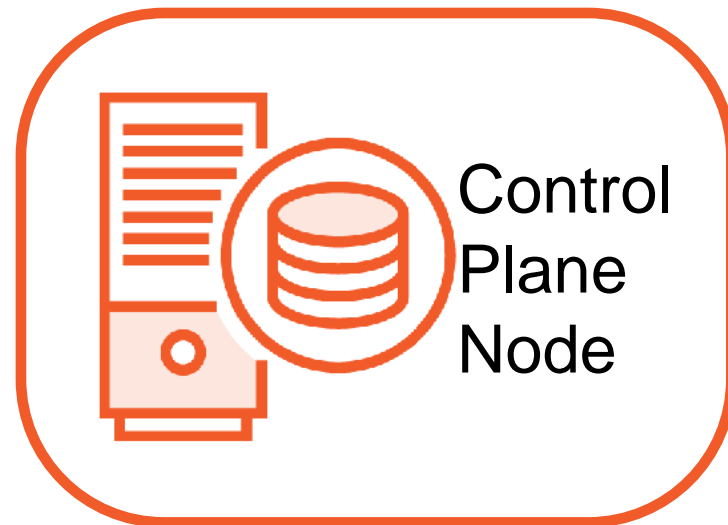
stop updates

Do this on all nodes

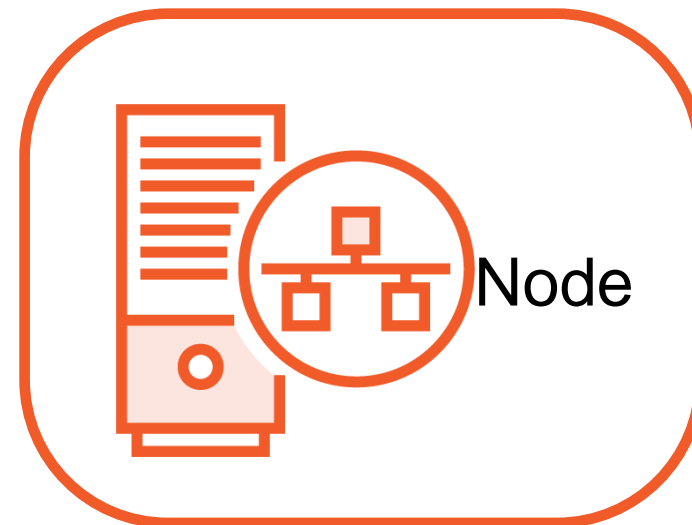
Lab Environment

Hostnames set
Host file on each

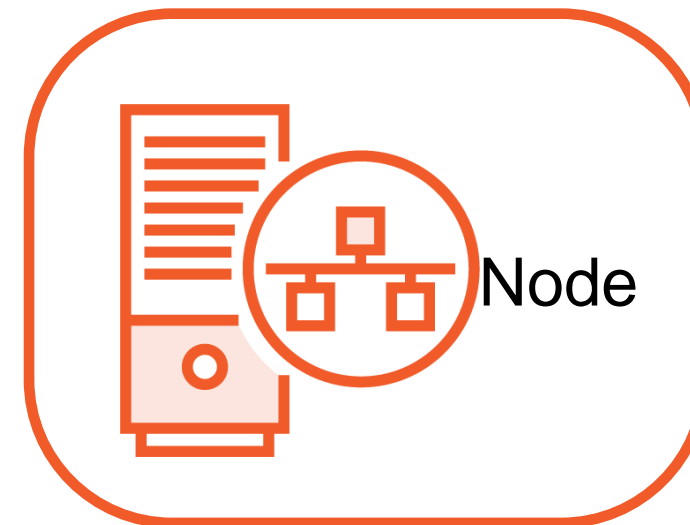
Ubuntu 18.04
VMware Fusion VMs
2vCPU
2GB RAM
100GB
Swap Disabled



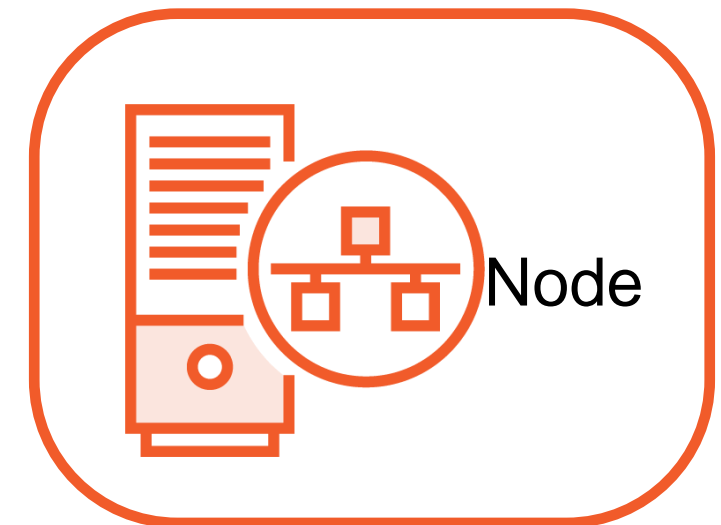
c1-cp1
172.16.94.10
c1-master1



c1-node1
172.16.94.11



c1-node2
172.16.94.12



c1-node3
172.16.94.13

Demo

Install Packages

- containerd
- kubelet
- kubeadm
- kubectl

systemd Units

Process is customizable

Bootstrapping a Cluster with kubeadm

kubeadm init

Pre-flight
checks

Creates a
Certificate
Authority

Generates
kubeconfig files

Generates
Static Pod
Manifests

Wait for the
Control Plane
Pods to Start

Taints the
Control Plane
Node

Generates a
Bootstrap
Token

Starts Add-On components:
DNS and kube-proxy

Certificate Authority



Self signed Certificate Authority (CA)

Can be part of an external PKI

Securing cluster communications

API Server

Authentication of users and cluster components

`/etc/kubernetes/pki`

Distributed to each Node

kubeadm Created kubeconfig Files



Used to define how to connect to your Cluster

Client certificates

Cluster API Server network 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

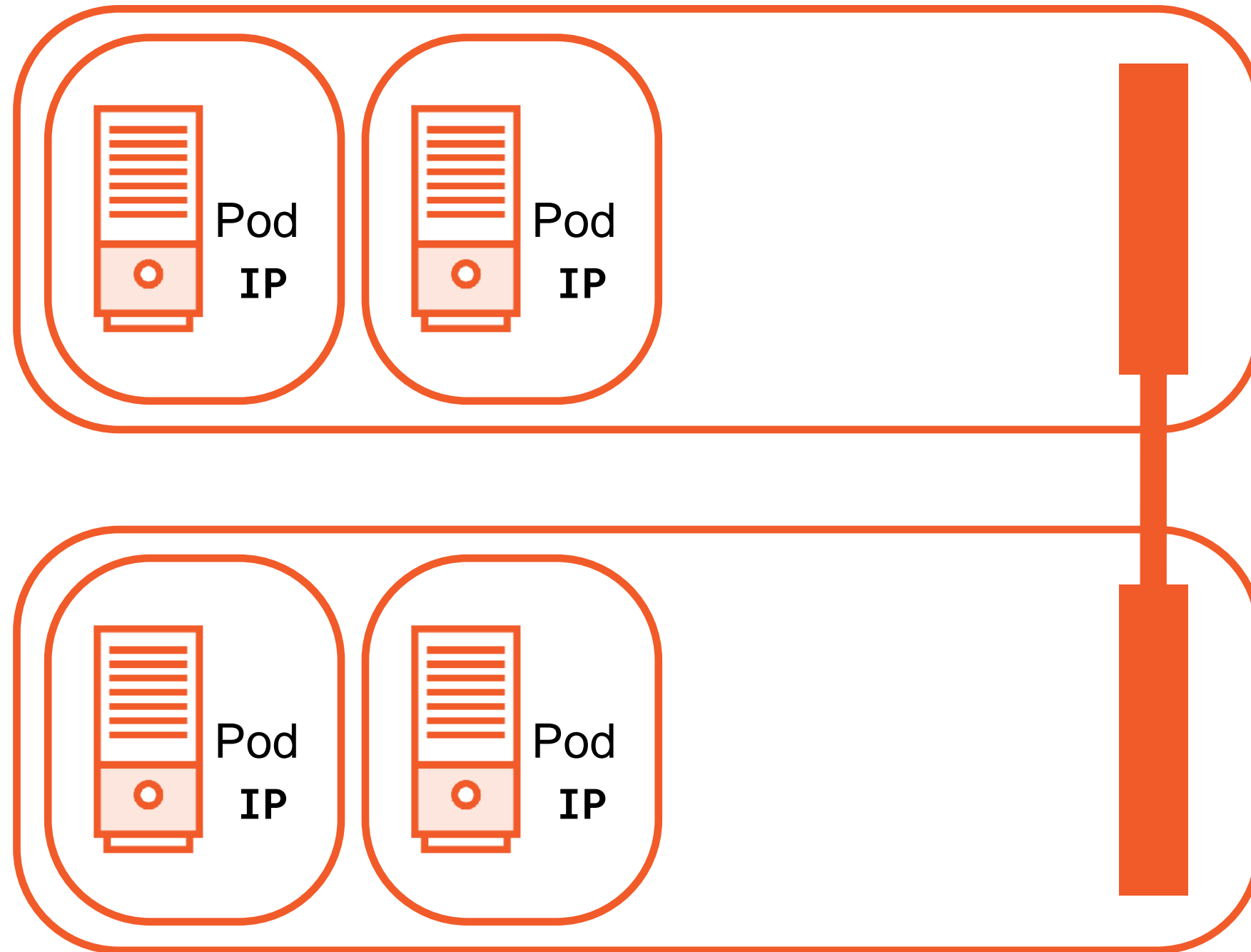
Controller Manager

Scheduler

Watched by the kubelet 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 and Nodes

Overlay networking

▶ Flannel - Layer 3 virtual network

Calico - L3 and policy based traffic management

Weave Net - multi-host network

Creating a Control Plane Node

```
wget https://docs.projectcalico.org/manifests/calico.yaml
```

```
kubeadm config print init-defaults | tee ClusterConfiguration.yaml
```

```
sudo kubeadm init \  
  --config=ClusterConfiguration.yaml \  
  --cri-socket /run/containerd/containerd.sock
```

```
mkdir -p $HOME/.kube
```

```
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
```

```
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

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

Adding a Node to a Cluster

Install Packages

```
kubeadm join
```

Download Cluster
Information

Node submits a CSR

CA Signs the CSR
automatically

Configures
`kubelet.conf`

```
kubeadm join 172.16.94.10:6443 \  
  --token i0pr88.pbid2af0071xhuo1 \  
  --discovery-token-ca-cert-hash \  
  sha256:9a56f13bbae1f77e3a01fecc2bf8c59e6977d9c71c2d3482b988fa47767353d7
```

Adding a Node to a Cluster

Demo

Creating a Cluster

Creating a Pod Network

systemd Units...again!

Static Pod manifests

Joining Nodes to a Cluster

Managed Cloud Deployment Scenarios



Elastic Kubernetes Service (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 Managed Service Cluster

- Azure Kubernetes Services (AKS)

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