Configuring and Managing Kubernetes Networking, Services and Ingress

KUBERNETES NETWORKING FUNDAMENTALS



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

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

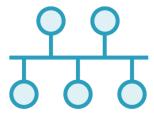
No Network Address Translation (NAT)

Motivations for the Network Model



Simplicity





Hide Implementation Details



All Pods can communicate to each other

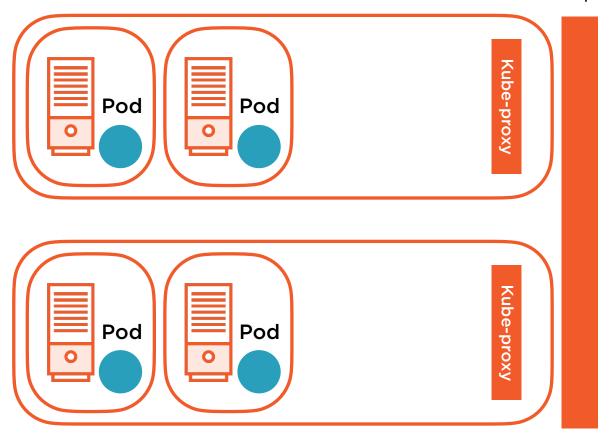


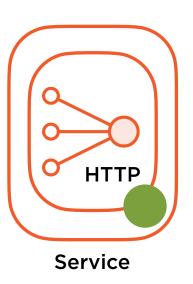
Administrator Controlled



Service Discovery and App Configuration

Kubernetes Network Topology



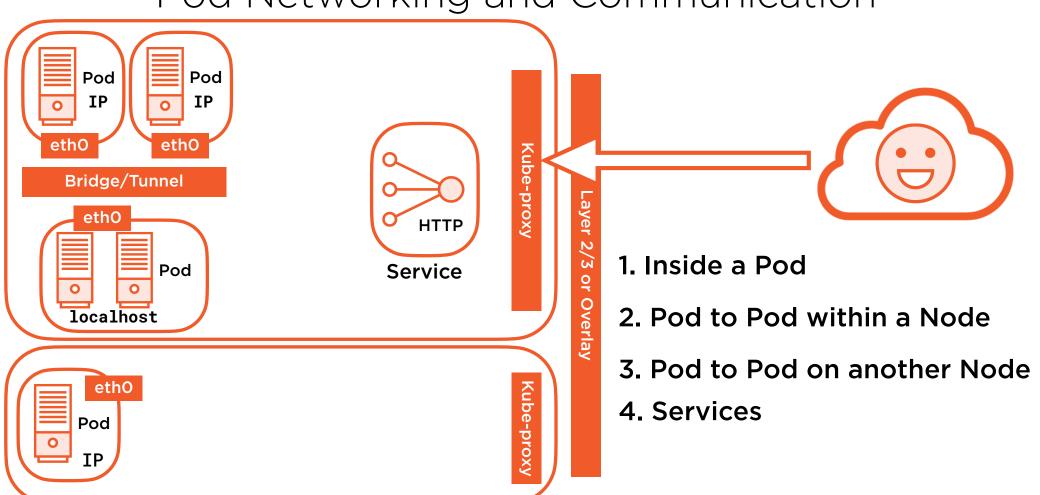


Pod Network

Node Network

Cluster Network

Pod Networking and Communication



Pod Networking Internals



Pod share a network namespace

Containers in a Pod communicate over localhost

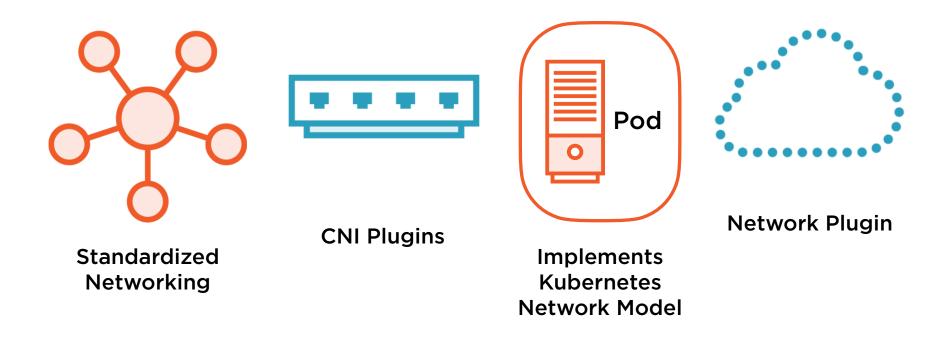
Pause/Infrastructure container

Starts the networking namespace

If the application container restarts the network will persist

Lifecycle of the Pod

Container Network Interface - CNI



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

Ubuntu 18.0.4 Hostnames set Lab Environment **VMware Fusion VMs** Host file on each 2vCPU **2GB RAM** 100GB **Swap Disabled** kubectl Control Node Plane Node Node Node c1-node1 c1-node2 c1-cp1 c1-node3

Kubernetes Installation and Configuration Fundamentals

172,16,94,12

172.16.94.13

172.16.94.11

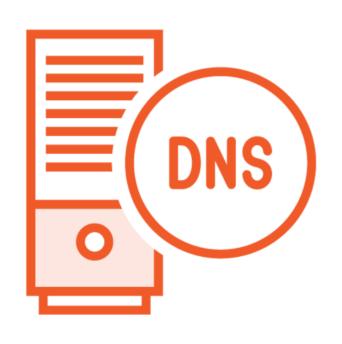
172, 16, 94, 10

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 . 1etc1r@solv.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

Investigated Pod and Service DNS Records

Review

Kubernetes network model

Network topology

Pod networking Internals

Container Network Interface - (CNI)

Cluster DNS

Up Next:

Configuring and Managing Application Access with Services