Introduction to Kubernetes

Neil Peterson
@nepeters

Delivered by KNect365 TMT
an Informa business

Session Topics

- Quick primer on containers
- Container mgmt solutions
- Kubernetes basics
- Kubernetes deeper dive
- Kubernetes beyond containers

Quick Container Primer

What is a container

- Application and process delivery mechanism
- Stateless with all included dependencies
- Fast start, hyper dense, and portable

Demo – Container

Container Management Solutions

Why do we need additional solutions to manage containers

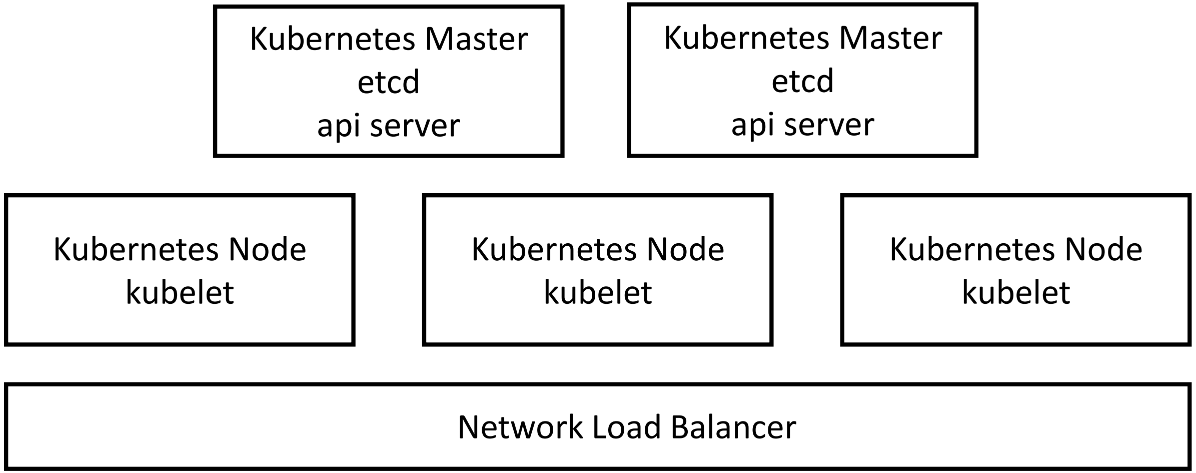
- Abstract hardware
- Manage state / health / lifecycle
- Manage networking, discoverability, etc.
- Manage sensitive data
- Manage scale

Kubernetes

A very popular option for managing containers (and potentially other things).

- Mature open source project
- Based on learning from internal Google projects
- Extensible / flexible API architecture
- API is expressed in object types
- Not necessarily just about containers

Kubernetes Infrastructure



Kubernetes Master
etcd
api server

Kubernetes Master
etcd
api server

Kubernetes Node
kubelet

Kubernetes Node
kubelet

Kubernetes Node
kubelet

Network Load Balancer

Pods

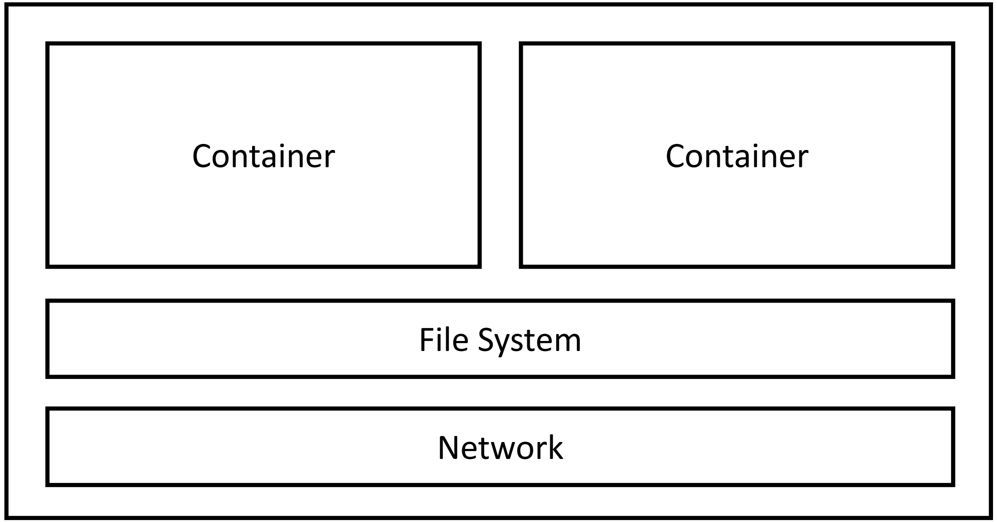
The primary primitive for running containerized workload.

- Unit of scheduling / resource envelope
- May contain one or more containers
- Containers share network, volumes, isolation components

Delivered by KNect365 TMT an Informa business

Pods Cont.

What does a pod look like?



Kubernetes Deployments

Pod scheduling abstract.

- Contains, POD, Container, and replica spec
- Replica set – how many instances of a pod

Kubernetes Services

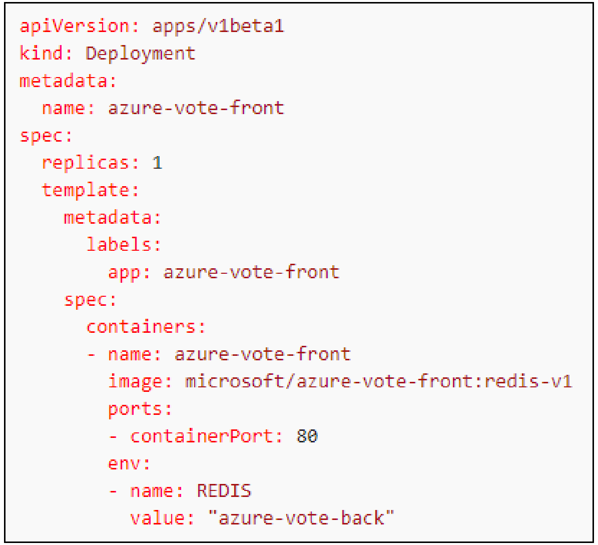
Kubernetes object expressing pod networking endpoint (internal / external IP address).

- Service is associate with pod through label selector
- ClusterIP – exposed on cluster-internal IP
- NodePort – exposed on nodes IP .via static port
- LoadBalancer – exposed externally on providers NLB
- ExternalName – map service to DNS name

Kubernetes Manifest

YAML file to declare desired state of Kubernetes object types.

- Define Kubernetes type
- Define type specification
- Labels / Annotations
- Metadata



```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: azure-vote-front
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: azure-vote-front
    spec:
      containers:
      - name: azure-vote-front
        image: microsoft/azure-vote-front:redis-v1
        ports:
        - containerPort: 80
        env:
        - name: REDIS
          value: "azure-vote-back"
```

Kubernetes Dashboard

Graphical user interface for interacting with a Kubernetes cluster.

- Create, update, delete objects
- Visual representation of state
- Take care to properly secure

Demo – Kubernetes

Delivered by KNect365 TMT an Informa business

Node Scaling

Provide more or less computer infrastructure for the cluster.

- Manually scale
- Auto scale (depending on provider)
- Take care to provide stateless workload

Pod Scaling

Provide more or less processing power for an application.

- Manually scale by increasing pod replica set count
- Auto scale using Horizontal Pod Autoscaling (HPA)
- Provide custom metrics for scale operations

Rolling Deployments

Zero downtime application updates.

- Deployment must contain more than one replica
- Pod are incrementally updated
- Configurable update schema (max unavailable / available)
- Rollback to previous version

Demo – Kubernetes Scale and App Updates

Delivered by KNect365 TMT an Informa business

Kubernetes Governance

Like all systems, security and governance should be a concern.

- Limit access to resources
- Provide isolated compute environments
- Ensure cluster safety

Namespaces

Unit of multi-tenancy cluster isolation.

- Provide isolated environments for teams / projects
- Can be accessed controlled
- Not necessary for running multiple applications
- Not every Kubernetes object can be added to a namespace

Role Based Access Control

Provide granular access to resources and operations.

- RBAC must be enabled on cluster
- Role / Cluster Role: rules that represent a set of permissions
- Role Binding / Cluster Role Binding: applies role to identity

Demo – Governance

Extending Kubernetes

Kubernetes is extensible to provide additional capability.

- Custom Resource Definitions (CRD)
- Operators
- Aggregated API extensions

Extending Kubernetes (CRD)

Custom Resource Definitions define a new Kuberntes object type (API Object).

- No code necessary *
- Instances of a CRD contain structured data
- Provides identical kubectl operational capibilities

Delivered by KNect365 TMT an Informa business

Extending Kubernetes (Operator)

Operators (controller) provide custom resource logic.

- Watch for new instances of a custom resource
- Grabs the structure data
- Performs some action

Demo – Extending Kubernetes