Kubernetes Basics

What is Kubernetes?

Container Orchestration Solution

Automates the deployment and management of containers

Why Kubernetes?

- Fastest growing community
- One of the top GitHub projects
- Aggregates Googles knowledge of the past years
- Better design decisions and implementations than competitors

Move fast and ~~break things~~!

Why container orchestration?

- Managing applications is hard
- Packaging applications is even harder
- Deploying applications is cumbersome

Automation is key!

What does Kubernetes offer?

- Pods based on immutable Docker images
- Deployments managing application updates
- Persistent Storage
- DNS Resolution for Services
- Secret Management
- Config Management

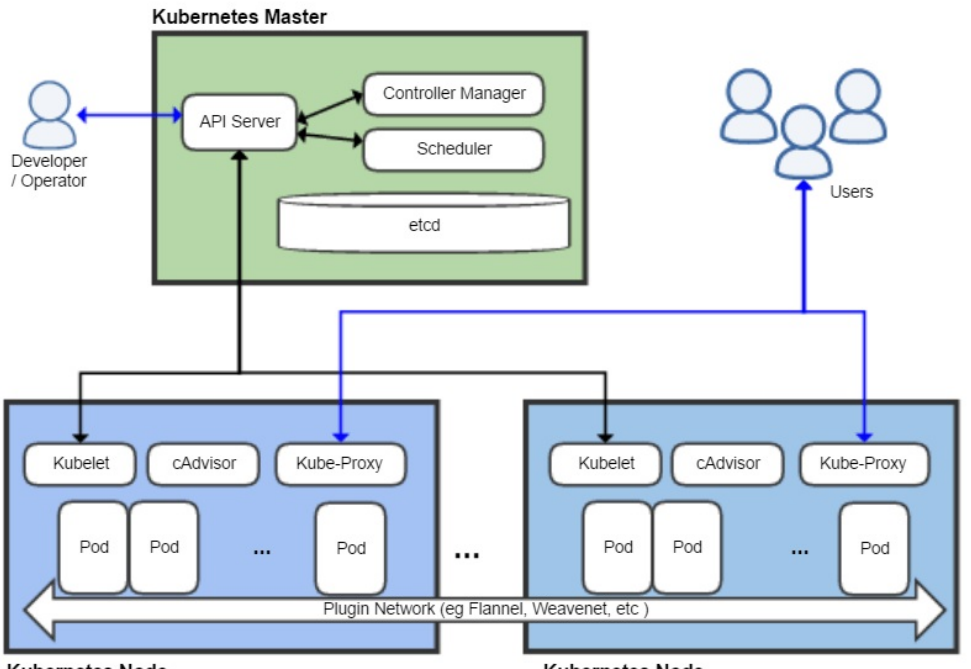
All of this via a "simple" REST API!

What is k8s handling for us?

- Scheduling
- Networking
- Storage
- Rolling Update

- Autoscaling

Kubernetes Architecture


Kubernetes Master
Developer
/ Operator
API Server
Controller Manager
Scheduler
etcd
Users
Kubelet
cAdvisor
Kube-Proxy
Pod
Pod
...
Pod
...
Kubelet
cAdvisor
Kube-Proxy
Pod
Pod
...
Pod
Plugin Network (eg Flannel, Weavenet, etc)
Kubernetes Node
Kubernetes Node

Kubernetes Terminology

Pods and Container

Pod is the smallest unit in Kubernetes

- Pod can consist of multiple containers
- All containers of a pod run on the same node
- Containers in a pod can communicate via localhost
- Containers in a pod share a kernel namespace

Service

Services abstract access to Pods

- Selection of Pods via Labels
- Typically only cluster internal access

Ingress[β]

Ingress allows external access to Services

- Hostname
- URL
- HTTPS possible
- TCP/UDP support in the future

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: "echo-ingress"
spec:

  backend:
    serviceName: "echoheaders"
    servicePort: 80
  tls:
    - hosts:
      - "echo.example.com"
      secretName: "echo-tls"
```

Deployment

Manages a set of Pods

- De facto standard for deploying applications
- Manages multiple iterations of an application
- Creates a ReplicaSets for each release
- Allows rollback to a previous release when desired

Persistent Volume Claim

Requests for persistent Storage

- Automatic provisioning in Azure Cloud
- Different requirements via StorageClass

Namespace

Separation of consumers/environments

- Resource Quotas
- All resources are attached to a namespace
- RBAC makes sharing of namespaces possible

Common CLI commands

kubectl

- CLI for **everything**
- Short for kube control
- Some believe it is pronounced kube-~~cuddle~~cuttle

kubectl create

Create resources in Kubernetes

- Definition in YAML or JSON
- Validation in `kubectl` client

```
kubectl create -f pod.yaml
kubectl create -f http://example.com/pod.yaml
cat pod.yaml | kubectl create -f -
kubectl create -R -f dir/
```

kubectl get

List Kubernetes resources

- Different output formats
- Filtering via labels possible

```
kubectl get all
kubectl get pods
kubectl get dc
kubectl get pods -o wide
kubectl get all -l env=production
kubectl get po/nodejs-ex -o yaml
```

kubectl edit

Edit resource definitions

- Validation in `kubectl` client
- Suitable for debugging

```
kubectl edit dc/nodejs-ex
```

kubectl delete

Deletion of resources

```
kubectl delete rc/nodejs-ex
```