

Kubernetes Architecture

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

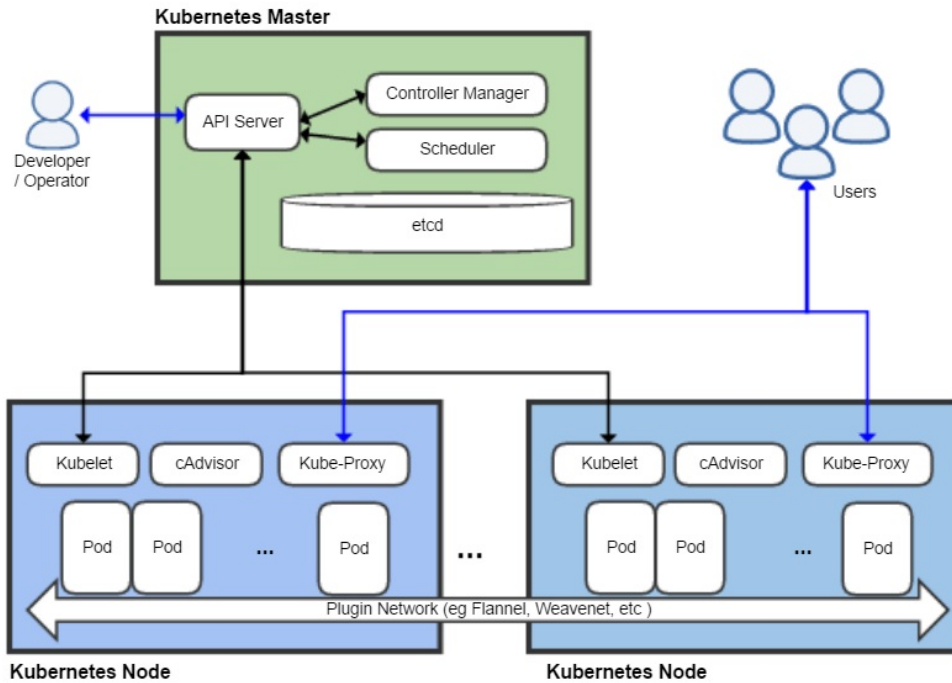
All of this via a "simple" REST API!

What is k8s handling for us?

- Scheduling
- Networking
- Storage
- Rolling Update

- Autoscaling

Kubernetes Architecture



Common CLI commands

kubectl

- CLI for **everything**
- Short for kube control
- **Some** believe it is pronounced kube-cuddlecuddle

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
```

