Kubernetes from 0 to production in 45 minutes

Bastian Hofmann

Agenda

01
- Simple web app

02
- Deploy it to Kubernetes

03
- You will learn about how Kubernetes works along the way

What I do

What is Qdrant?

Qdrant is a vector similarity search engine (or vector database)

- Semantic search
- Recommendations
- Fraud detection
- Anomaly detection
- Generative AI
- ...

kubernetes

Common compute platform across infrastructure

- ✓ Common API & Packaging
- ✓ Health Checks/HA
- ✓ Load Balancing
- ✓ Overlay Networking
- ✓ Network Security Policies
- ✓ Backup and Recovery
- ✓ Autoscaling
- ✓ Service Discovery
- ✓ Networking
- ✓ RBAC & Access Control

About containers

```go
package main

import (
    "fmt"
    "net/http"
)

func hello(w http.ResponseWriter, req *http.Request) {
    fmt.Fprintf(w, "Hello, world!\n")
}

func headers(w http.ResponseWriter, req *http.Request) {
    for name, headers := range req.Header {
        for _, h := range headers {
            fmt.Fprintf(w, "%v: %v\n", name, h)
        }
    }
}

func main() {
    http.HandleFunc("/hello", hello)
    http.HandleFunc("/headers", headers)
    http.ListenAndServe(":80", nil)
}
```

Application Code

Container Image

Application Dependencies

Demo

Containers are great

Managing a couple – no problem

Containers are great …… but

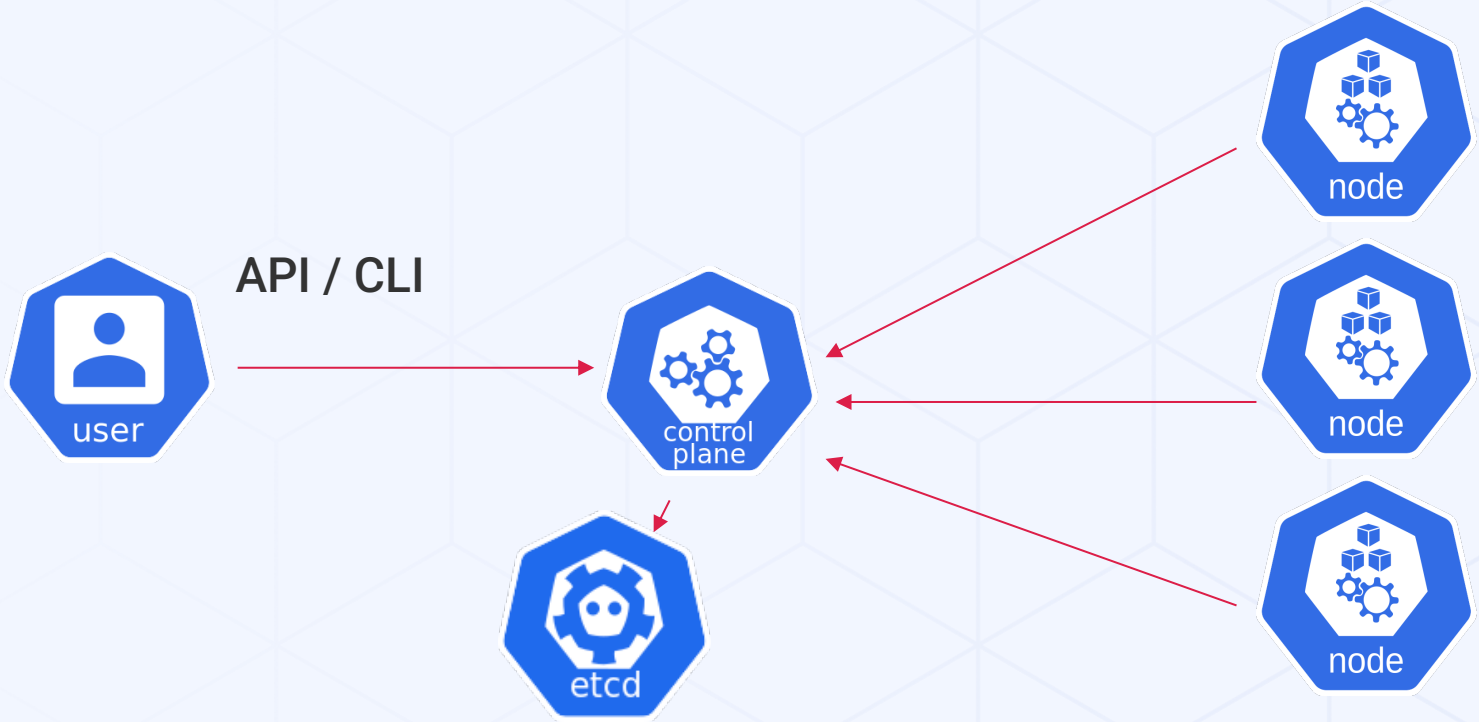
How about managing many?

How do we address:

Networking, Security, Scheduling, Automation, etc?

Container Orchestration

Kubernetes Architecture



Declarative API

Everything is a RESTful resource

CRUD on those resources

Pod

- One instance of your application
- Contains one or more containers
- Has an individual IP address within the Kubernetes network

Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: my-application
  namespace: application-namespace
  labels:
    app: my-application
spec:
  containers:
    image: company/my-image:1.0.0
    ports:
    - containerPort: 8080
status:
  ...
```

Deployment

- Manages lifecycle of Pods
- Multiple replicas
- Different update strategies

Deployment


```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-application
  namespace: application-namespace
spec:
  replicas: 2
  strategy:
    type: RollingUpdate
  template:
    spec:
      containers:
      - image: company/my-image:1.0.0
        ports:
        - containerPort: 8080
```

Service

- Kubernetes internal “load balancer”
- Static internal IP address mapped to healthy Pods
- Has a DNS entry in the Kubernetes internal DNS
- Different types to also expose workloads outside of the cluster

Service

```
apiVersion: v1
kind: Service
metadata:
  name: my-application
  namespace: application-namespace
spec:
  ports:
  - port: 80
    targetPort: 8080
  selector:
    app: my-application
  type: ClusterIP
```

Ingress

- Configuration of an Ingress Controller to route incoming traffic to the cluster to internal Services based on HTTP host and path
- Configure TLS

Ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-application
  namespace: application-namespace
spec:
rules:
  - host: my-application.example.com
    http:
      paths:
      - backend:
          service:
            name: my-application
            port:
              number: 8080
        path: /
        pathType: Prefix
```

ConfigMap & Secret

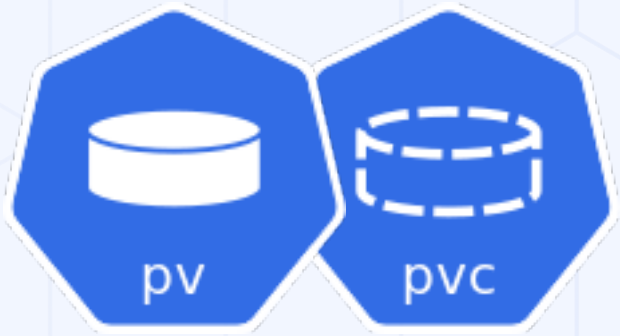
- Internal key/value store for configuration settings
- Can be used in Pods through environment variables or volume mounts

ConfigMap & Secret

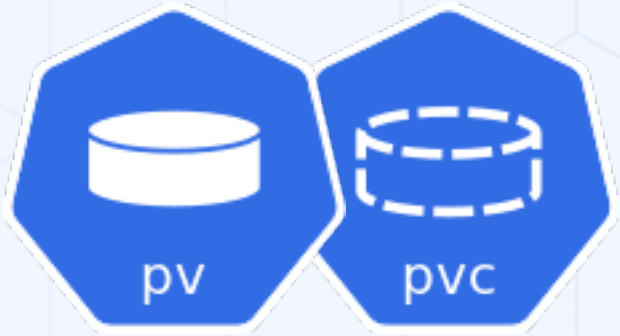
```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-application
  namespace: application-namespace
data:
  CONFIG_KEY: some-value
  ANOTHER_KEY: another value
```

PersistentVolumeClaim & PersistentVolume

- Mount persistent storage into Pods



PersistentVolumeClaim & PersistentVolume




```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-application-volume
  namespace: application-namespace
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 8Gi
  storageClassName: some-storage-provider
```

There are many more

CustomResourceDefinition

- Extend the Kubernetes API with additional resource types
- Kubernetes operators can then implement additional abstractions and logic

Demo

Resources

- https://k3s.io/
- https://k3d.io/
- https://longhorn.io/
- https://www.rancher.com/
- https://cert-manager.io/
- https://artifacthub.io/packages/helm/bitnami/redis
- https://helm.sh/
- https://kubernetes.io/

Thank you

Bastian.Hofmann@qdrant.com