Kubernetes Basics

By Alex Bissessur

What is Kubernetes?

- Industry standard for cluster computing
- Open source technology created by Google
- Kubernetes in Greek translates to “helmsman” or “pilot” of a ship
- “K8S is a system for automating deployment, scaling and management of containerised applications“
  - Simply an abstraction to run a cluster of computers
- Performs a lot of the heavy lifting, including:
  load balancing, automatic scaling, IP/port exposure

What is Containerisation?

- Containers are similar to VMs in practice
- Shares the underlying kernel, so no need for a hypervisor
- Applications bundled with requirements & are isolated
- Containers are extremely lightweight
  (both computing power and storage)
- An Nginx server with an HTML/CSS webpage < 50MB
  Containers may be specialised for certain languages/ frameworks. Eg: node.js

How does Kubernetes Work?

- It works on the basis of containers
  K8S is a container orchestration system
- Containers are packed in pods
  - Pods are the smallest unit for K8S
  - A pod can contain one or more containers
- Most of everything in K8S is defined through manifests (.yaml files)

How does Kubernetes Work?

- Pods can be deployed to your cluster by themselves
  In prod, you would use deployments instead
- Deployments encapsulate pods
  - They use replicasets to manage pods
  - They also keep pods alive, in case the node dies
  - Self healing for pods
  - Rolling updates & easy rollbacks
- Services then expose the pods
  Pods are assigned Ips but they may be moved or deleted (eg during update) and the IP will change
  Services ensure static addressing of your, well, service

Why Kubernetes?

- Horizontal scalability – just add more computers
- Redundancy from having several computers
  - High availability
- Maintenance without causing downtime
  - Blue-green deployment
- Fine control over deployments
  - Can limit resources
- Automating deployments (CI/CD integration)

Why Kubernetes?

- Self-healing – replaces failed pods
- Portability
  - Due to containerised nature, deployments can easily be moved to different clusters
- Easy to manage and manipulate
  - Less workload on sysadmins
- Automatic scaling of pods
  - No panic when your website becomes famous

Why **Not** Kubernetes?

- Simplicity
  - K8S adds overhead to a project
  - Docker compose + rented hosting may achieve same result
- Learning Curve
  - Can be negatively impactful if your devs are not familiar
- Legacy Projects
  - May not be suitable for microservices architecture
- Infrastructure limitations

Demo