Kubernetes

Cloud Computing and SaaS

Announcements

- Project
    - Checkpoint 4 Skeleton Code Released
        - Recommended to finish by Friday
    - Local Tests Released
    - Make sure to pull from source repo!
- We're getting close to AWS deployment -- Make sure your code passes the tests!

Last Time

- Guest Speaker: Usman Muzaffar
  - Head of Cloudflare Engineering
- SQL Injections and XSS

Today

- Kubernetes
  - Overview
  - History
  - Applications
  - Architecture
  - Benefits
  - Drawbacks

What is Kubernetes?

- Open-Source Container Orchestration Platform
- Containers
    - A high-scale project may have lots of containers
    - How do we manage them?
        - Reliability, Recovery, Repair?
    - We have docker-compose, but it’s only for a single machine -> we need to scale
    - Kubernetes!

Kubernetes - History

- Borg (2003-2004)
  - Small-scale project within Google
  - Created in order to manage internal compute clusters
  - Mostly focused on reliability of long-running services

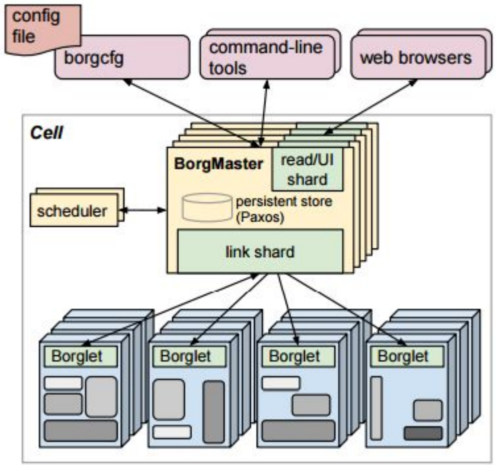


Figure 1: The high-level architecture of Borg. *Only a tiny fraction of the thousands of worker nodes are shown.*

Kubernetes - History

- Omega (2013)
  - Based on Borg
  - More consistent architecture
  - Managed clusters
    - More flexible, scalable scheduler for larger tasks
  - Whitepaper: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41684.pdf

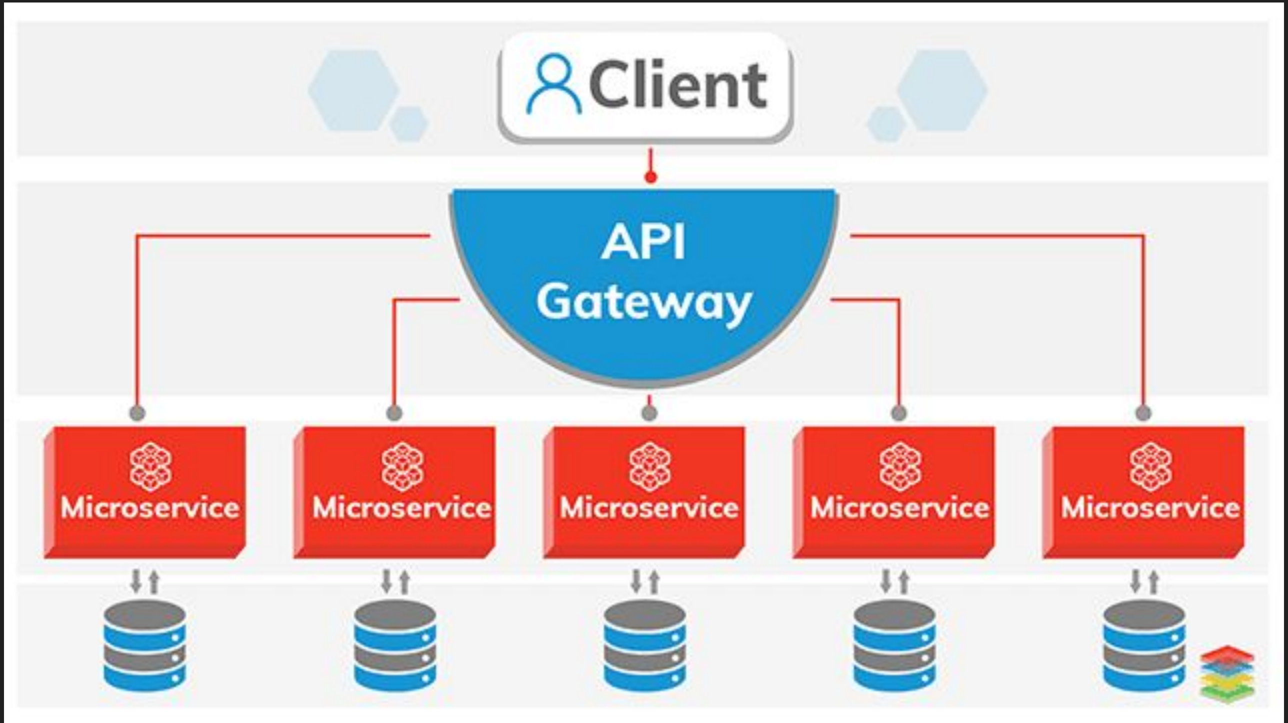
Kubernetes - History

- Kubernetes (2014)
  - Introduced as an open-source version of Borg
- Google partners with Linux Foundation and forms the Cloud Native Computing Foundation (CNCF)
  - Focus on container orchestration and microservice architectures
  - Website: cncf.io

Kubernetes - Applications

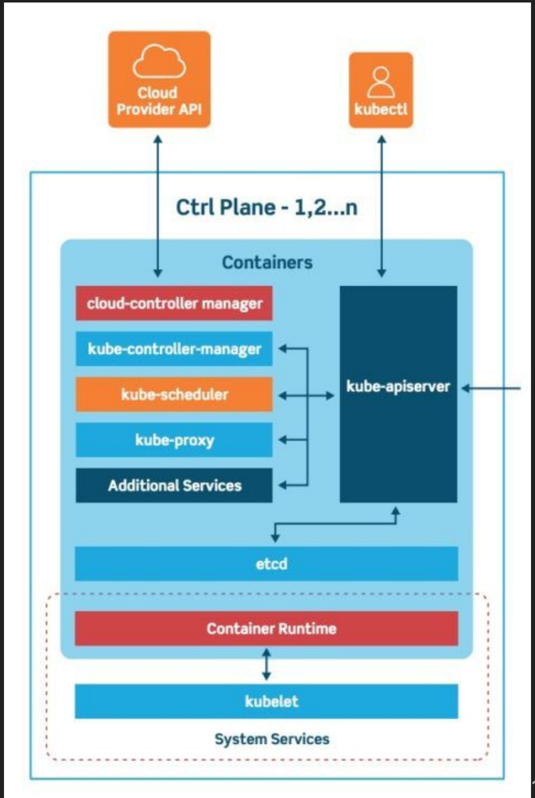
- Cloud Software Development
  - Want to maintain high uptime, Kubernetes provides reliability
- Microservices
  - Kubernetes is optimized to container-based infrastructure
- Scale up/down containers + resources on demand
  - Spawn more containers or shut down current ones
- Replace, Restart, and Replicate resources

Recall: Microservices



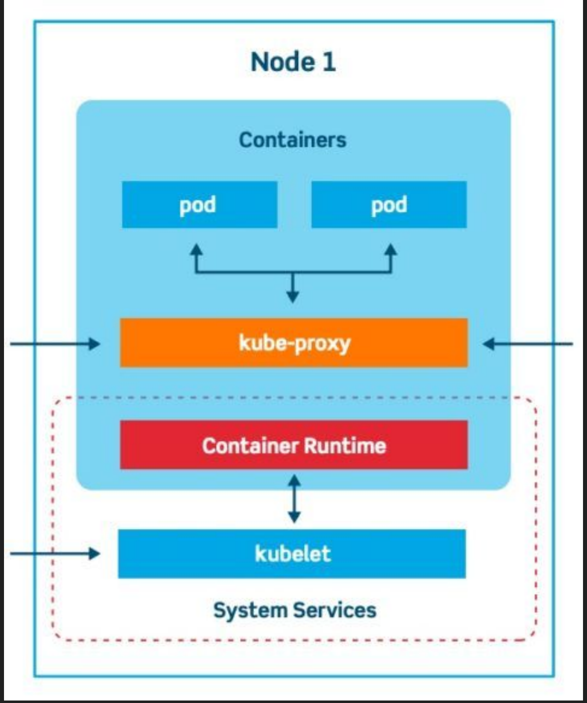
Architecture - Control Plane

- Keeps record of all Kubernetes objects and manages state
- 3 components
    - API Server
        - Provides APIs to scale and update applications
    - Controller Manager
        - Background service - directly manages clusters
    - Scheduler
        - Schedules containers across different clusters
        - Considers configs, resources, etc.



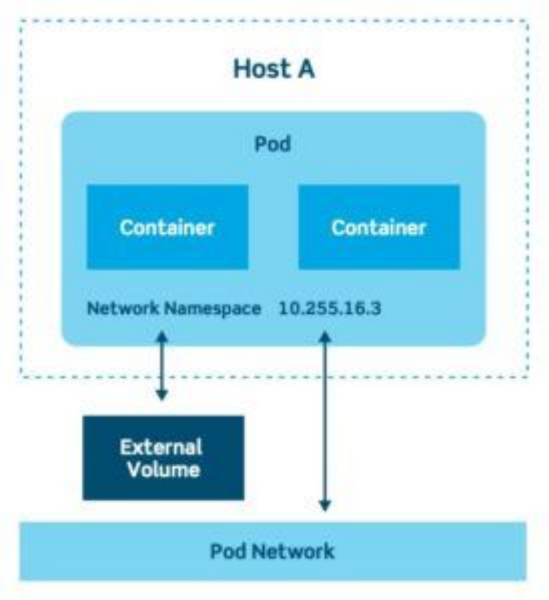
Architecture - Cluster Nodes

- The actual machines that run containers
- Also referred to as a “Kubelet”
- Drives the execution of the containers
    - Usually Docker containers

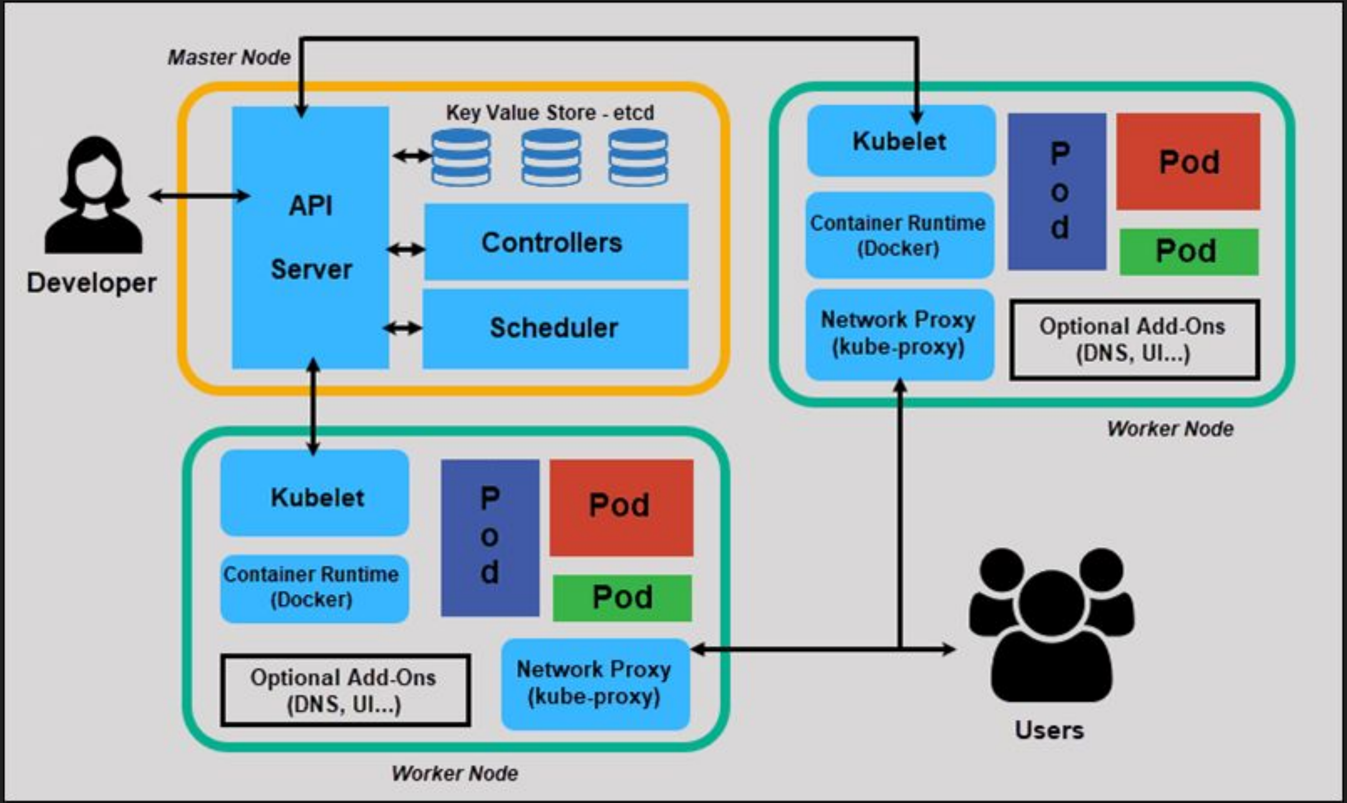


Architecture - Pods

- Key Construct of Kubernetes
- Logical package for a single application
    - Can consist of multiple containers and storage volumes
    - Usually consists of a single container
- Think about a “Pod” as a process in a cluster
    - Each Host/Node can have multiple pods



Architecture - Overview



Architecture - Further Reading

- https://www.redhat.com/en/topics/containers/what-is-kubernetes
- https://www.redhat.com/en/topics/containers/kubernetes-architecture
- https://kubernetes.io/docs/concepts/workloads/pods/
- https://kubernetes.io/docs/concepts/overview/components/

Kubernetes - Benefits

- Load Balancing
  - Can redirect traffic to containers that have the smallest workload
- Storage
  - Can manage/mount/unmount storage as necessary
- State Control
  - Describe what state you want -> Kubernetes enforces it
- Config Management
  - Can store sensitive information, abstracts it from the containers

Kubernetes - Drawbacks

- Not a solution to every application
  - Overkill for small applications
- Very complex application
  - Time invested in learning the ins and outs
- Can be more expensive when migrating old infrastructures
- Is a platform, but doesn't have all the tools of platform
  - Doesn't include monitoring, logging, etc.