Docker suckless with Kubernetes

Thuan Duong - Monkey DevOps

February 27, 2017

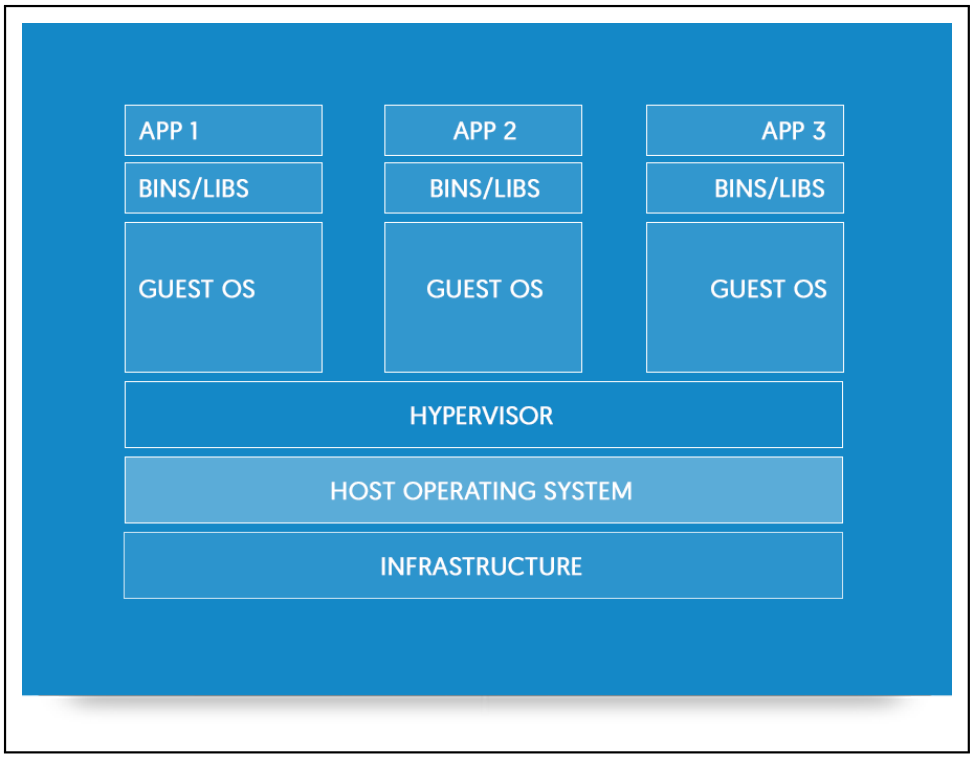
Agenda

1. Docker
2. Kubernetes
   - What is Kubernetes?
   - Concepts
   - Architecture
   - Google Container Engine (GKE)
3. Demo
4. QA

What is Docker?

- An implementation of the container idea
- A package format
- Resource isolation
- An ecosystem

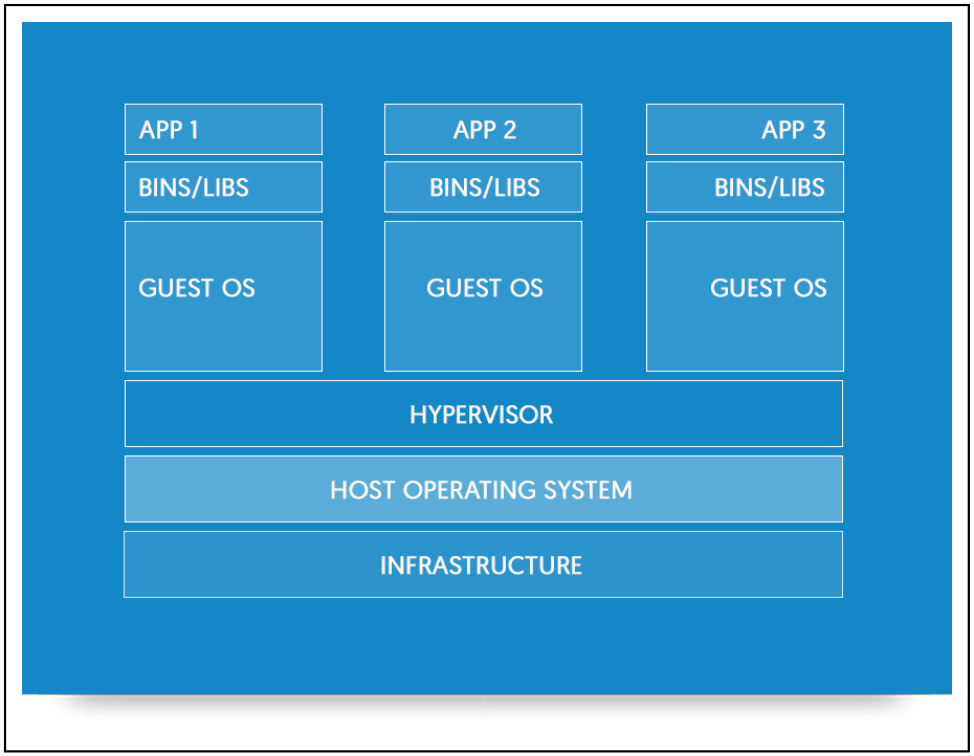
VM and Docker



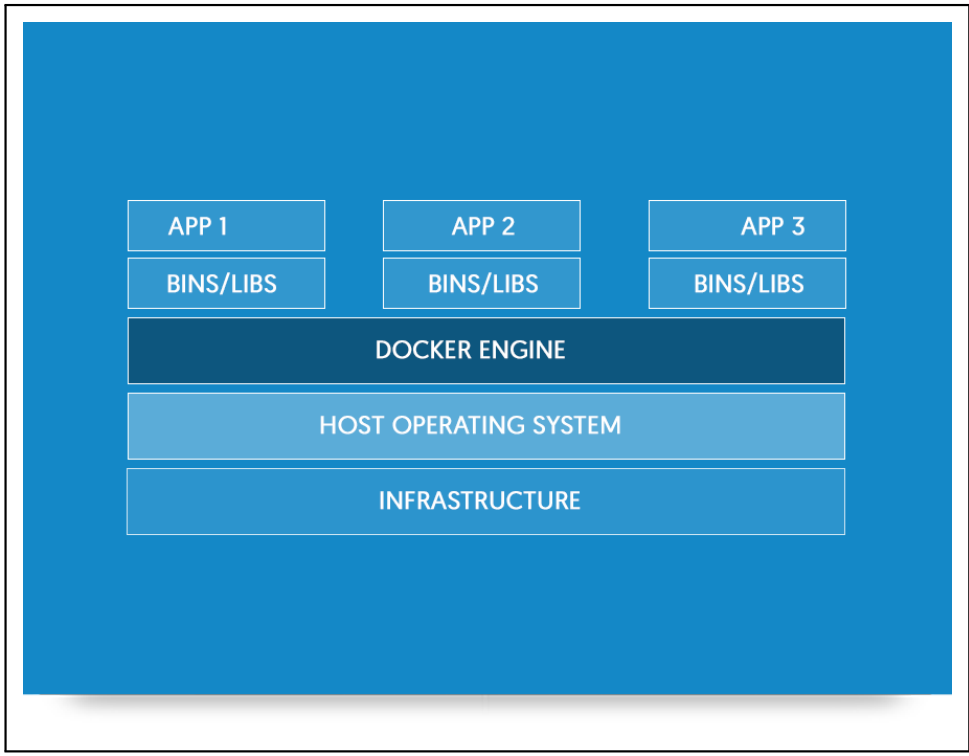
Virtual Machine

[1] https://www.docker.com/what-docker

VM and Docker



Virtual Machine



Docker

[1] https://www.docker.com/what-docker

Resource isolation

Implemented by a number of Linux APIs:

- **cgroups**: Restrict resources a process can consume
  - CPU, memory, disk IO, ...

[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

Resource isolation

Implemented by a number of Linux APIs:

- **cgroups**: Restrict resources a process can consume
  - CPU, memory, disk IO, ...
- **namespaces**: Change a process's view of the system
  - Network interfaces, PIDs, users, mounts

[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

Resource isolation

Implemented by a number of Linux APIs:

- **cgroups**: Restrict resources a process can consume
  - CPU, memory, disk IO, ...
- **namespaces**: Change a process's view of the system
  - Network interfaces, PIDs, users, mounts
- **capabilities**: Limits what a user can do
  - mount, kill, chown

[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

Resource isolation

Implemented by a number of Linux APIs:

- **cgroups**: Restrict resources a process can consume
  - CPU, memory, disk IO, ...
- **namespaces**: Change a process's view of the system
  - Network interfaces, PIDs, users, mounts
- **capabilities**: Limits what a user can do
  - mount, kill, chown
- **chroot**:
  Determines what parts of the filesystem a user can see.

[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

We need more than just packing and isolation

- **Scheduling**: Where should my container run?
- **Lifecycle and health**: Keep my containers running despite failures
- **Discovery**: Where are my containers now?
- **Monitoring**: What's happening with my containers?
- **Auth{n,z}**: Control who can do things to my containers
- **Aggregates**: Compose sets of containers into jobs
- **Scaling**: Make jobs bigger or smaller
- ...

[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

What is Kubernetes?

- Container orchestrator
- Builds on Docker containers also supporting other container technologies
- Multiple cloud and bare-metal environments
- Support existing OSS apps
- Inspired and informed by Google's experiences and internal systems
- **100% Open source**, written in **Go**
- Let users manage **applications**, not machines

[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

Primary concepts

- **Container**: A sealed application package (Docker)

[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

Primary concepts

- **Container**: A sealed application package (Docker)
- **Pod**: A small group of tightly coupled Containers

[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

Primary concepts

- **Container**: A sealed application package (Docker)
- **Pod**: A small group of tightly coupled Containers
- **Labels**: Identifying metadata attached to objects

[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

Primary concepts

- **Container**: A sealed application package (Docker)
- **Pod**: A small group of tightly coupled Containers
- **Labels**: Identifying metadata attached to objects
- **Selector**: A query against labels, producing a set result

[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

Primary concepts

- **Container**: A sealed application package (Docker)
- **Pod**: A small group of tightly coupled Containers
- **Labels**: Identifying metadata attached to objects
- **Selector**: A query against labels, producing a set result
- **Controller**: A reconciliation loop that drives current state towards desired state
  - Deployment
  - StatefulSet
  - Job
  - ...

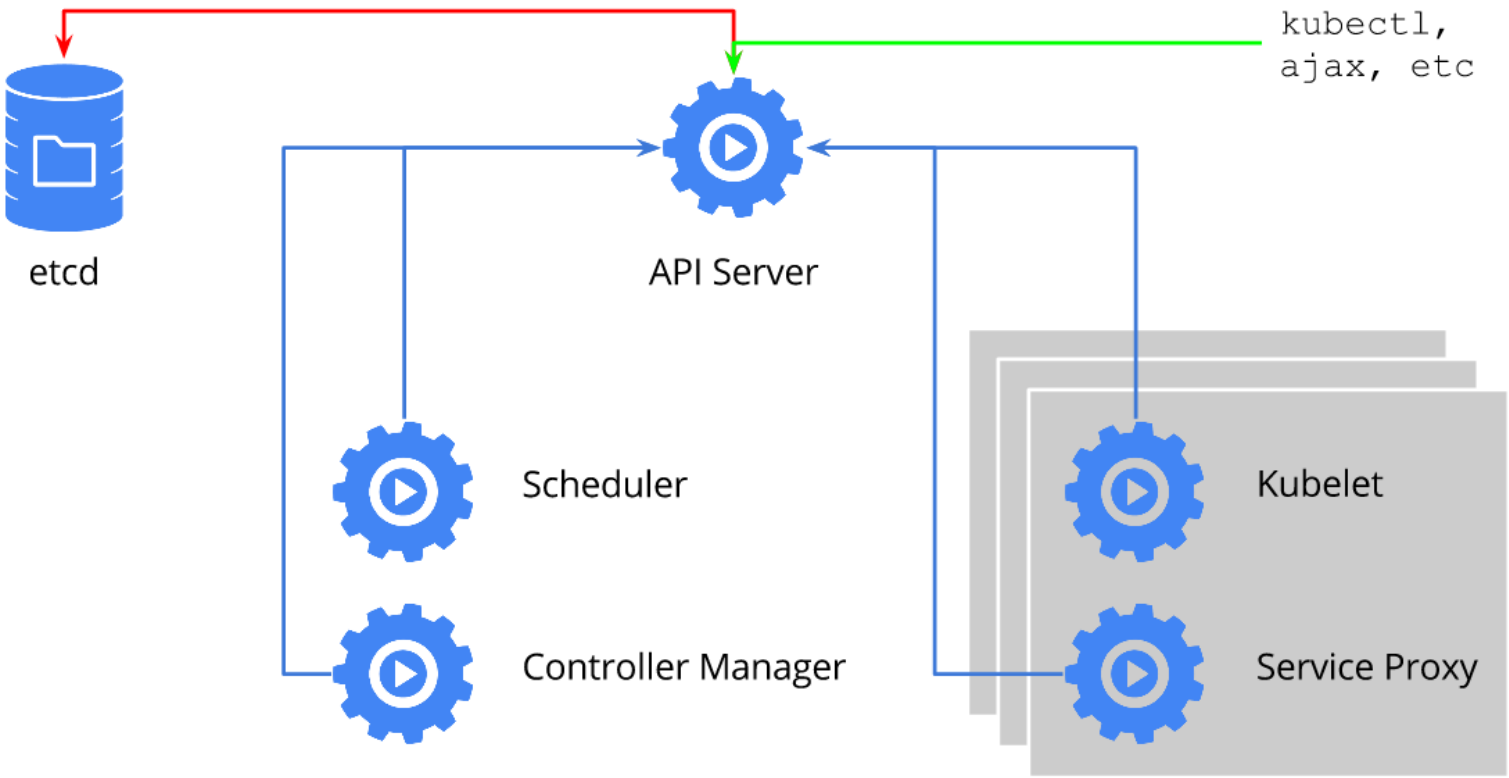
[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

Primary concepts

- **Container**: A sealed application package (Docker)
- **Pod**: A small group of tightly coupled Containers
- **Labels**: Identifying metadata attached to objects
- **Selector**: A query against labels, producing a set result
- **Controller**: A reconciliation loop that drives current state towards desired state
  - Deployment
  - StatefulSet
  - Job
  - ...
- **Service**: A set of pods that work together

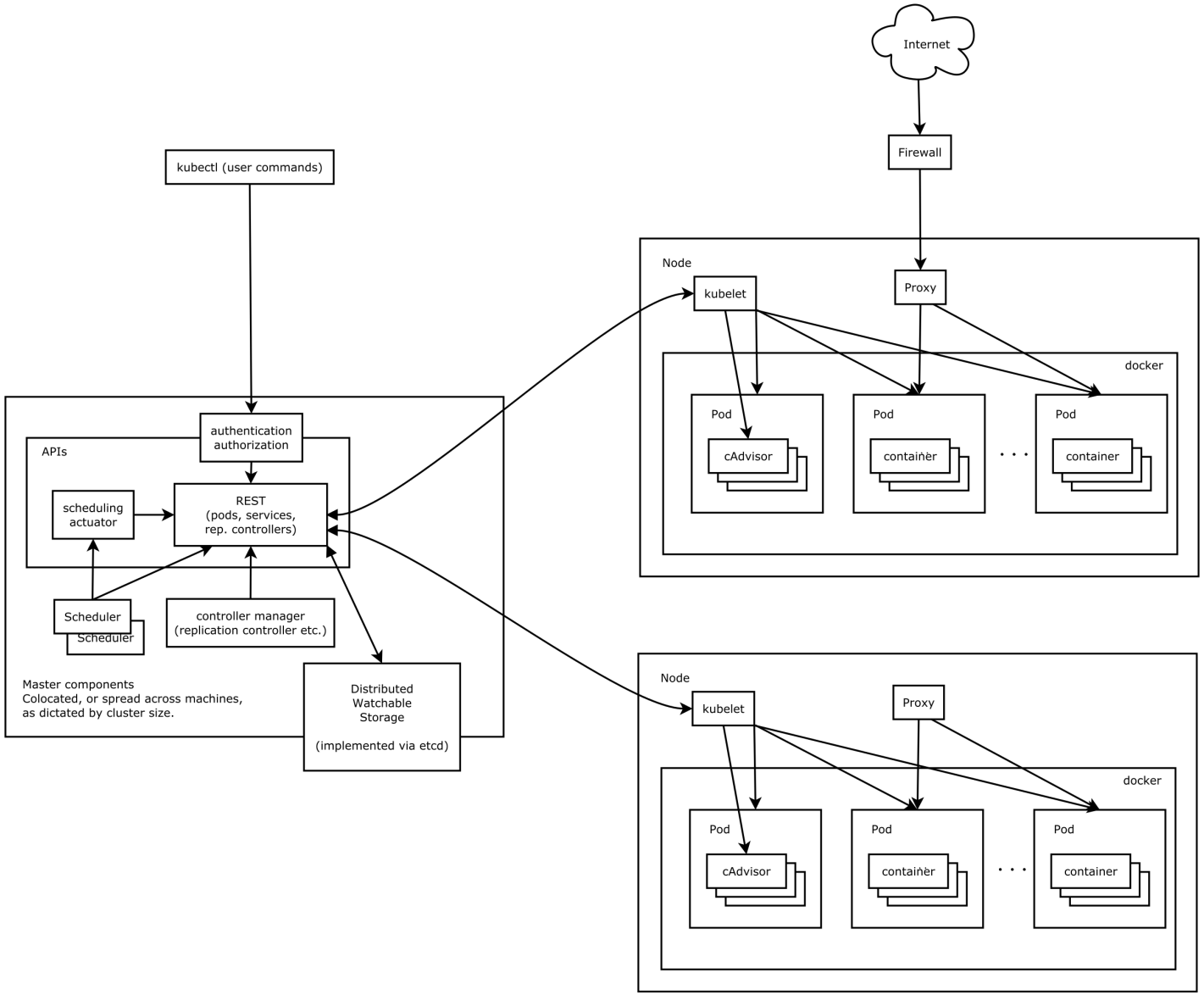
[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

Kubernetes Architecture



[1] Cluster Management with Kubernetes `https://goo.gl/er33Gn`

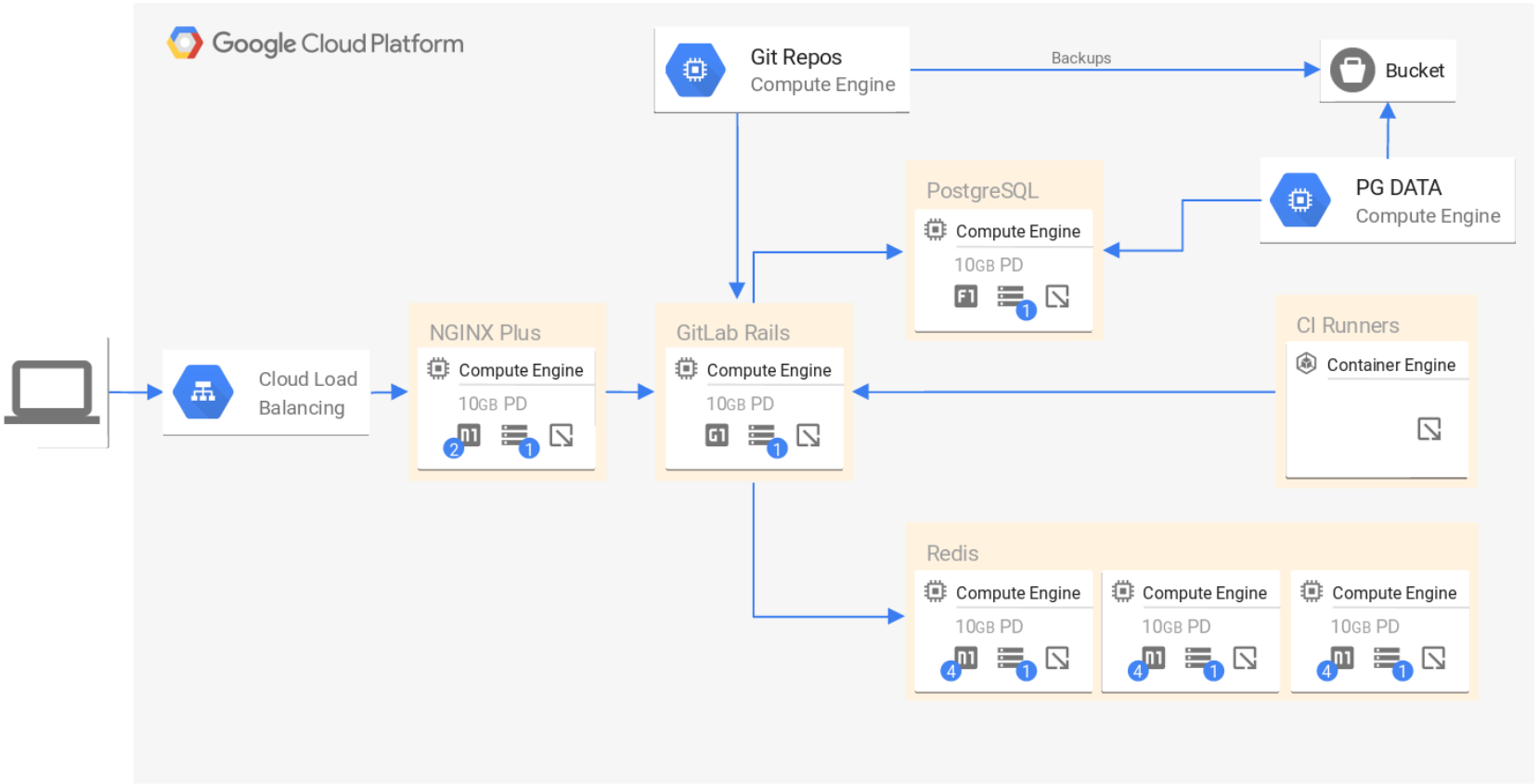
Kubernetes Architecture



Google Container Engine (GKE)

GKE is a powerful cluster manager and orchestration system for running your Docker containers. Container Engine schedules your containers into the cluster and manages them automatically based on requirements you define (such as CPU and memory). It's built on the open source Kubernetes system, giving you the flexibility to take advantage of on-premises, hybrid, or public cloud infrastructure.

Example: Gitlab CI/CD on GCP



Google Container Engine

Questions?