Google Container Engine

Google Cloud Platform

Agenda

1 Kubernetes and Container Engine

2 Container Cluster Resources

3 Interacting with Container Engine

4 Lab

Kubernetes

- Sometimes shortened to ‘k8s’
- Open source orchestration system
- Based on over 10 years experience at Google
- Built for a multi-cloud world:
    - Public
    - Private
    - Hybrid
- Modular design
- Can run almost anywhere

Google Container Engine

- Based on **open source** Kubernetes orchestration system
- Orchestrate and **schedule** Docker containers
- Consumes **Compute Engine** instances and resources
- Uses a **declarative syntax** to manage applications
- **Decouple** operational and development concerns
- Manages and maintains
  - Logging
  - Health management
  - Monitoring
  - Scaling

Agenda

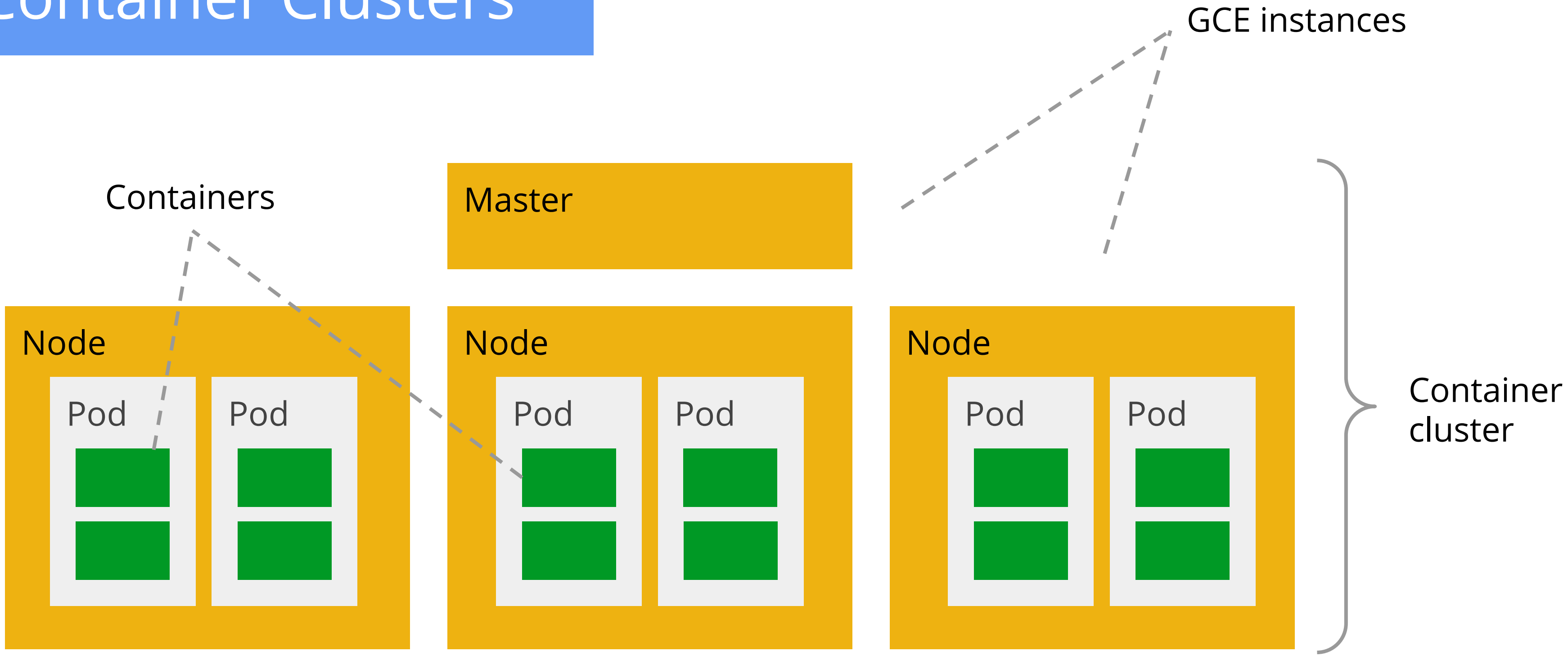
1 Kubernetes and Container Engine

2 Container Cluster Resources

3 Interacting with Container Engine

4 Lab

Container Clusters



Google Container Engine clusters are used to reliably scale container based workloads

Cluster Resources

Pods

Pods are **ephemeral** units that are used to manage one or more tightly coupled containers.

They enable **data sharing** and **communication** among their constituent components.

Replication Controllers

Replication controllers create new pod "replicas" from a **template** and ensure that a configurable number of those pods are running.

Services

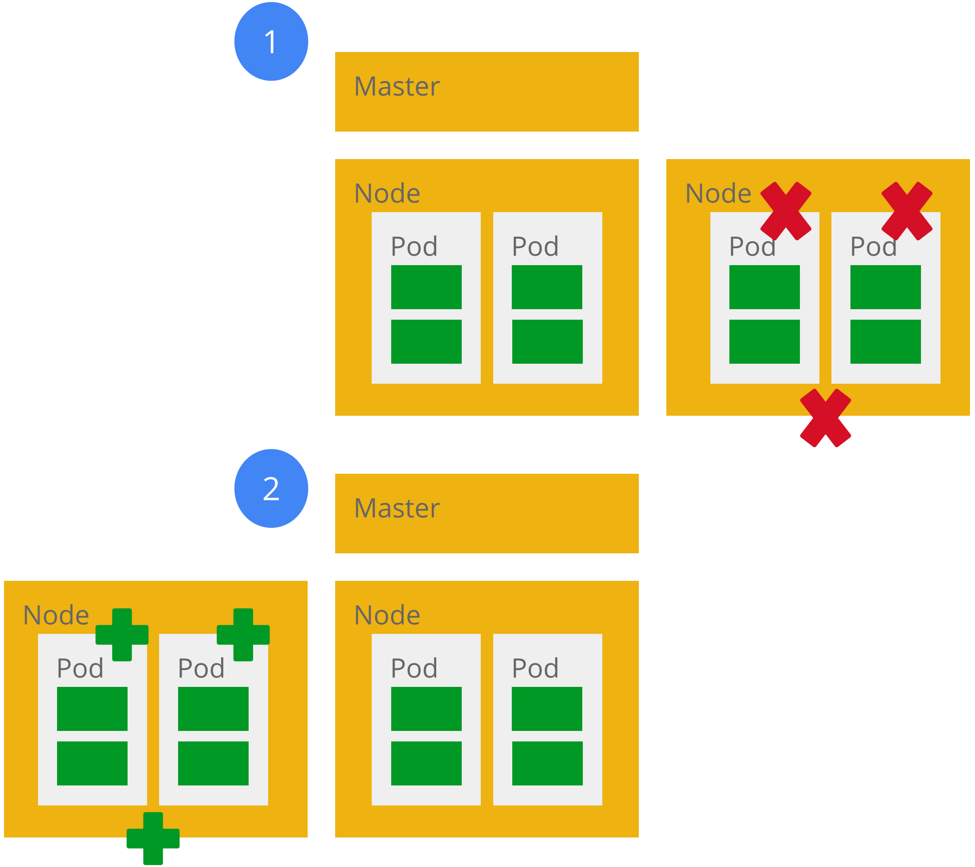
Services provide a bridge based on an **IP and port pair** for non-Kubernetes-native client applications to access backends without needing to write code that is Kubernetes-specific.

Pods

- When a host node instance dies, the pods scheduled to that node are deleted permanently

- Specific pods are never rescheduled to the new node instances

- Instead they must be replaced with new pods

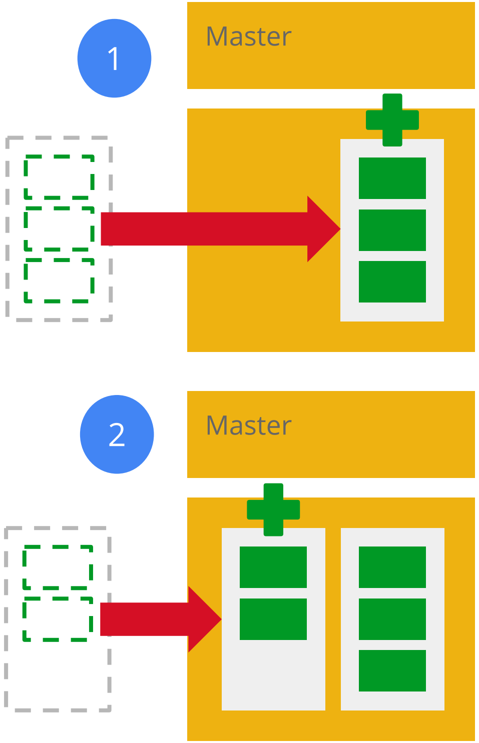


Replication Controllers

- Template settings are only used for new pod replica creation

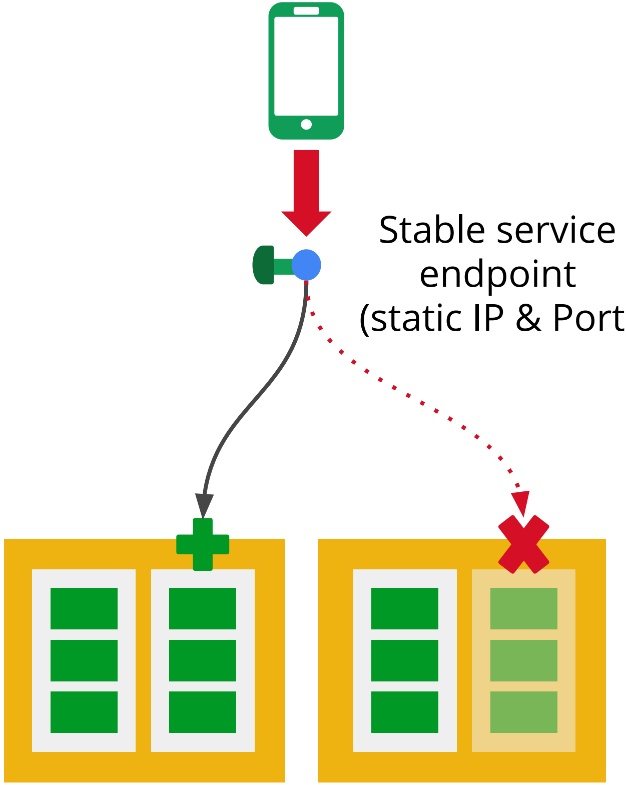
- Changes to a template do not affect already-running pods or the switch to a new template

- Best practice to use a replication controller even for a single pod



Services

- Pods are **ephemeral** and can be started or stopped at any time by the replication controller
- Pods will dynamically grow and shrink
- A service provides a stable **endpoint**



Stable service endpoint (static IP & Port)

Agenda

1 Kubernetes and Container Engine

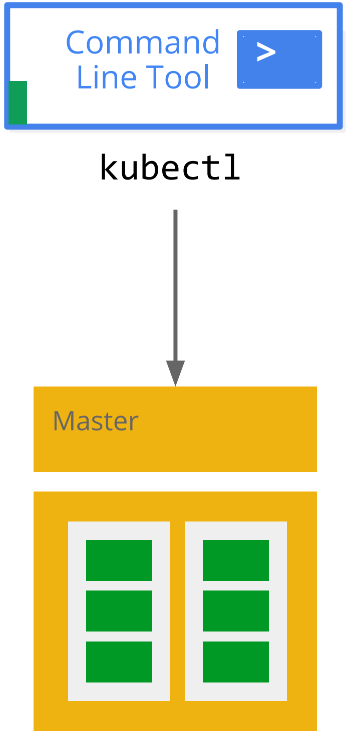
2 Container Cluster Resources

3 Interacting with Container Engine

4 Lab

kubectl

- kubectl is a command line utility that sends requests to the Kubernetes cluster manager
- Run **remotely** (not included in Cloud SDK)
- Frequently used kubectl commands include: get, describe, create, update, delete
  - Example - To get a list of pods:
    `kubectl get pods`



Agenda

1 Kubernetes and Container Engine

2 Container Cluster Resources

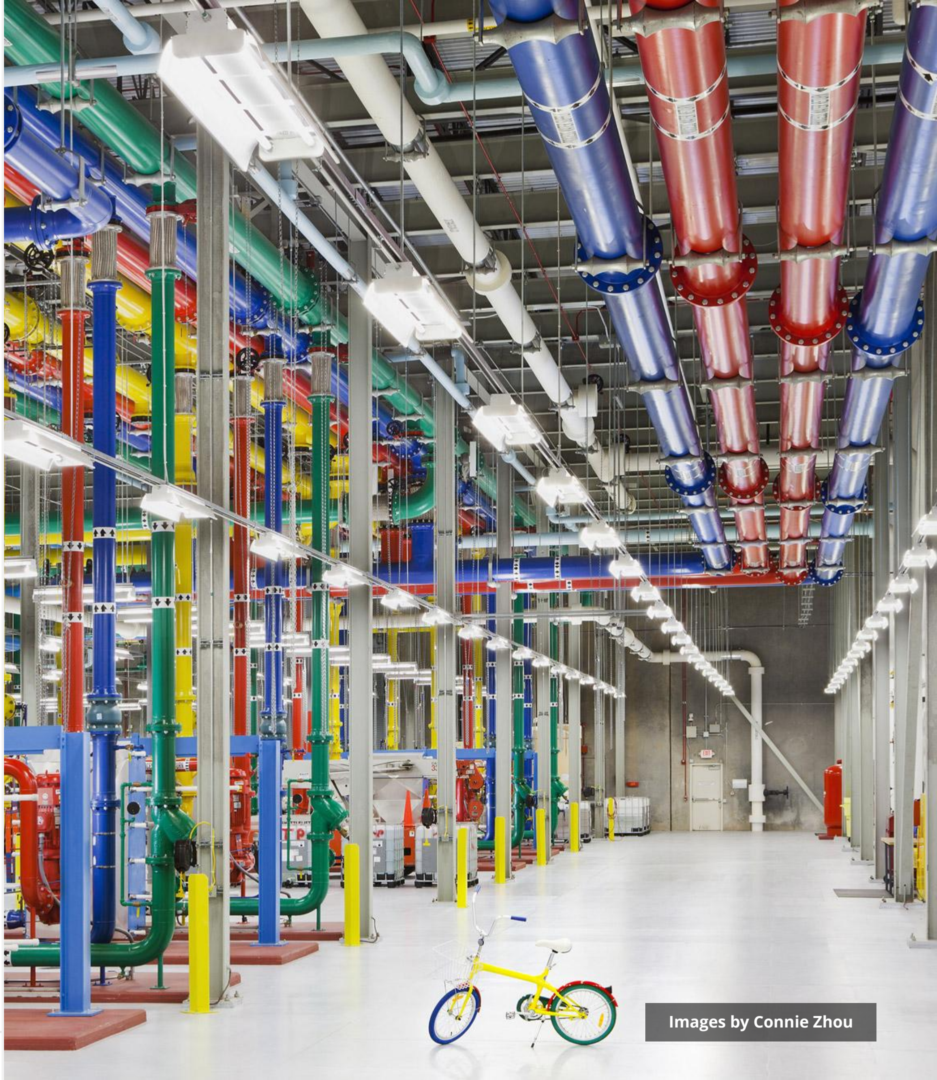
3 Interacting with Container Engine

4 Lab

Lab (1 of 2)

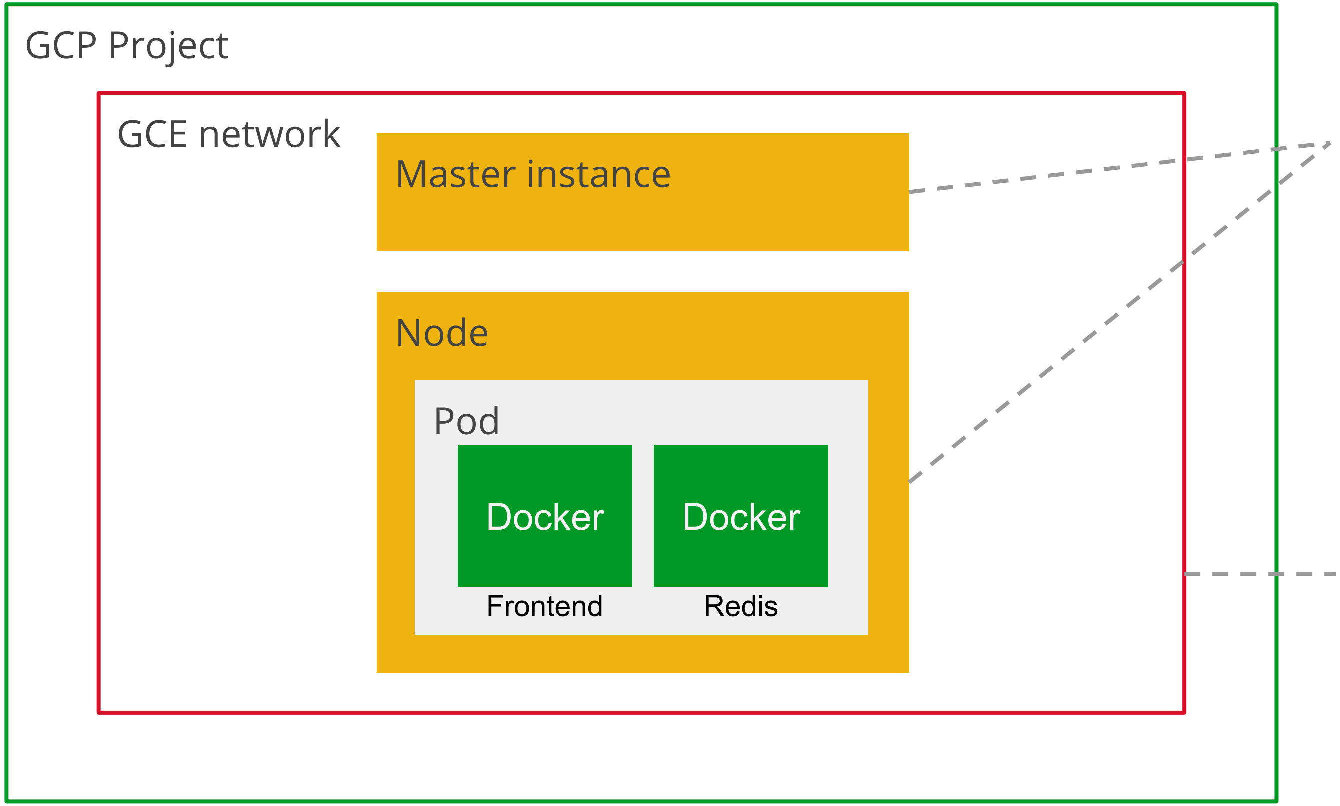
Deploy the Guestbook application on Container Engine.

1. Create a Container Engine cluster
2. Deploy a pod with the application containers
3. Test the application in your browser
4. Finish by deleting the container cluster



Images by Connie Zhou

Lab (2 of 2)



Container Engine cluster comprised of 2 Compute Engine instances; a master & the node that will run the application pod

Will need to create firewall rule to allow traffic on port 80 (HTTP) again

Resources

- Container Engine: Features Pricing Documentation
  https://cloud.google.com/container-engine/
- Kubernetes
  http://kubernetes.io/
- An introduction to containers, Kubernetes, and the trajectory of modern cloud computing
  http://googlecloudplatform.blogspot.co.uk/2015/01/in-coming-weeks-we-will-be-publishing.html
- Hello Wordpress
  https://cloud.google.com/container-engine/docs/hello-wordpress

cloud.google.com

Images by Connie Zhou