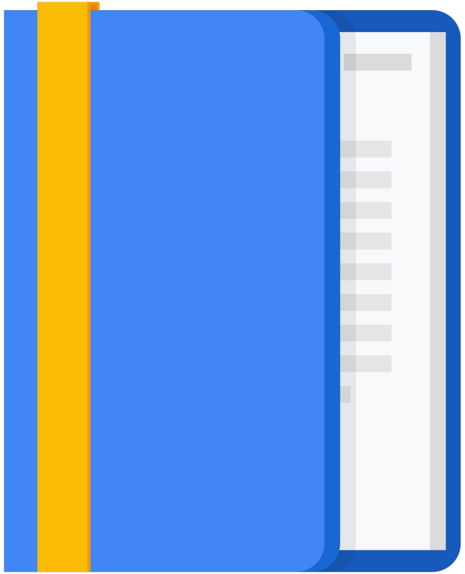
Introduction to Kubernetes

Agenda

What is Kubernetes

Why should I care

How does it work



What is Kubernetes?

Kubernetes is a platform for working with containers

- portable, open-source, **container-centric** management platform
- Built-in primitives for **deployments, scaling, monitoring, and more**
- Inspired by **Google’s internal systems**

What is Kubernetes?

At its core Kubernetes gives three things:

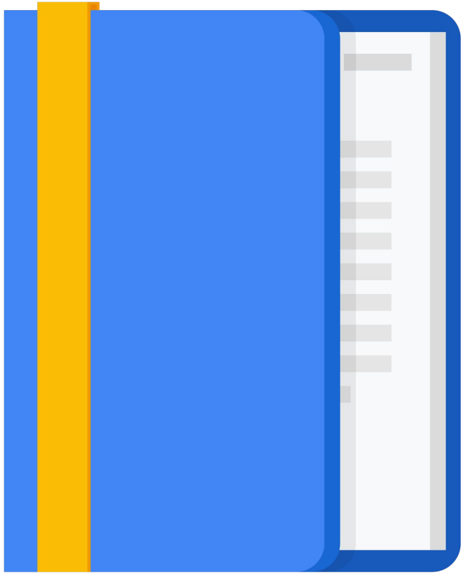
- Deployment
- Scaling
- Monitoring

Agenda

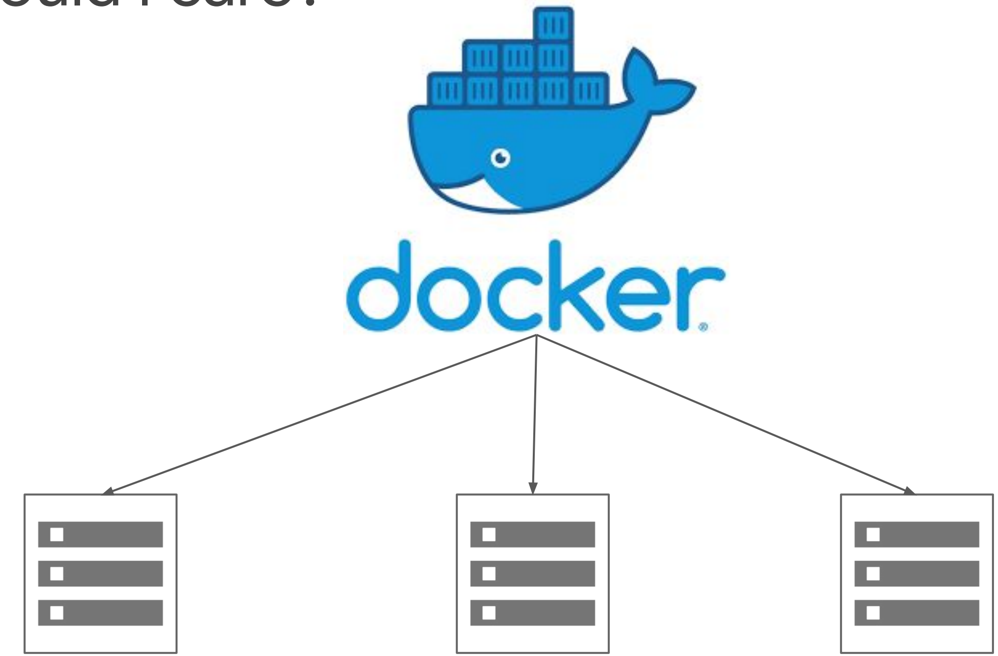
What is Kubernetes

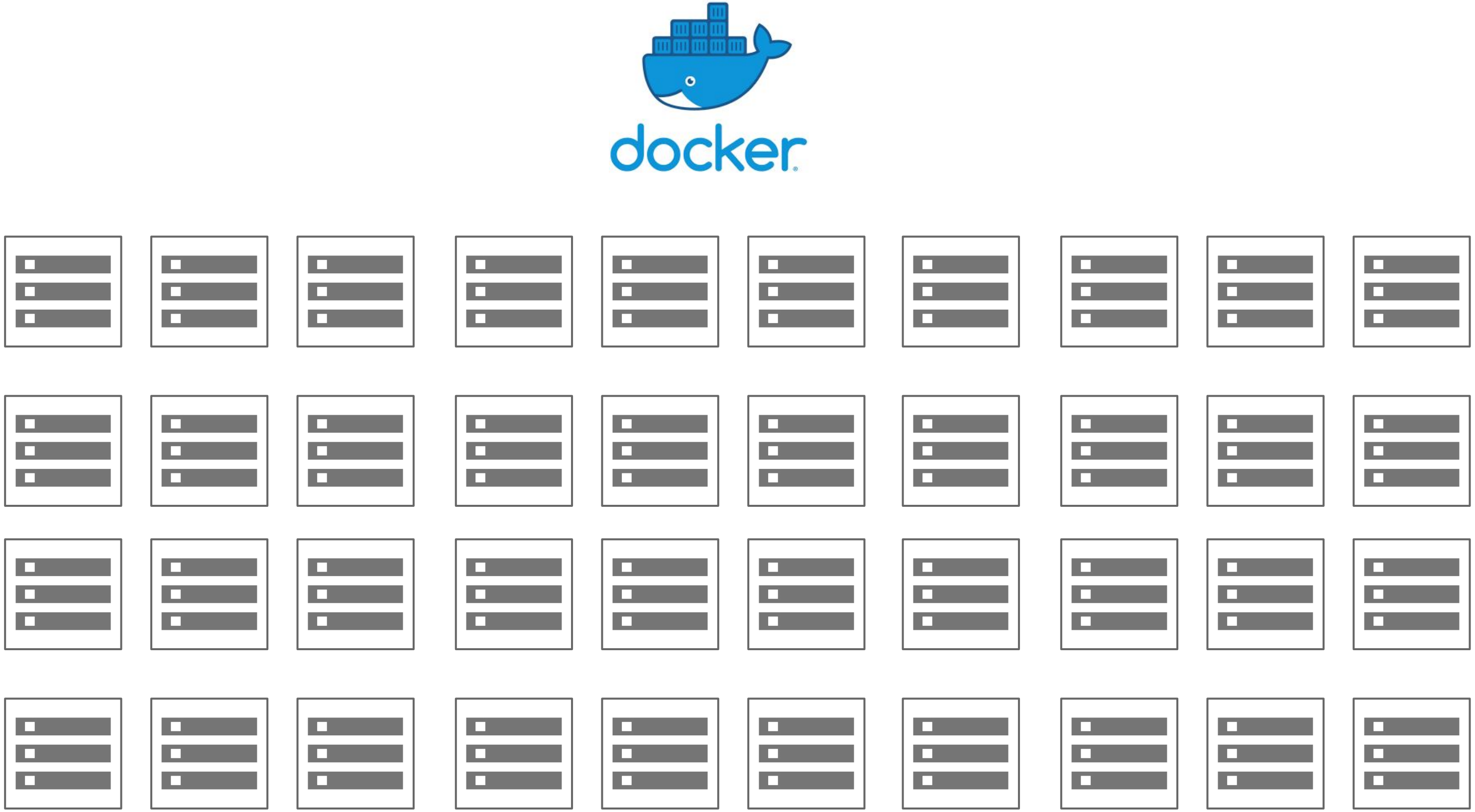
Why should I care

How does it work



Why should I care?



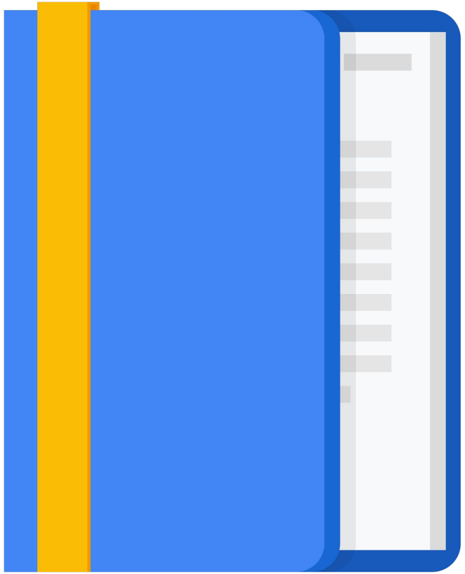

docker

Agenda

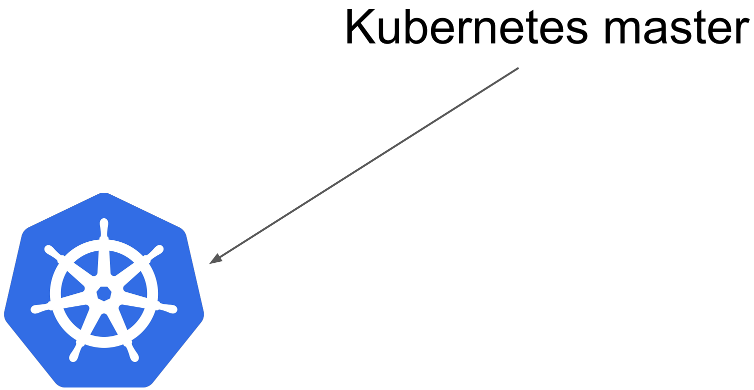
What is Kubernetes

Why should I care

How does it work

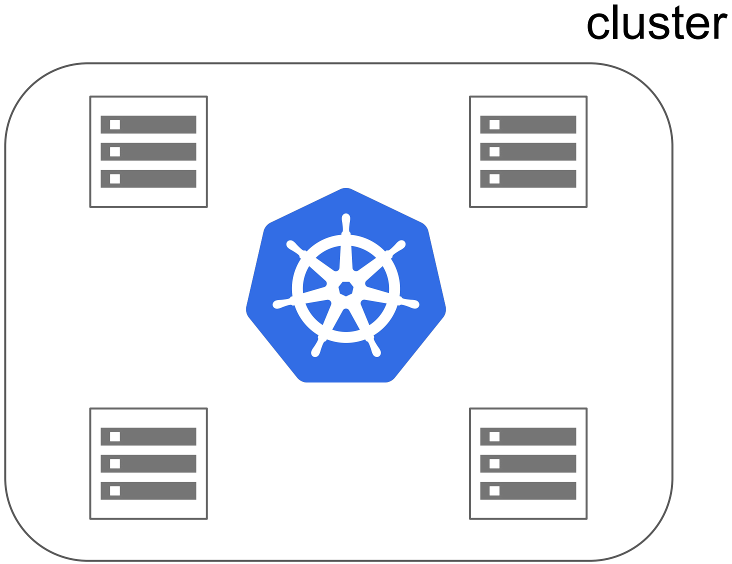


How does it work?



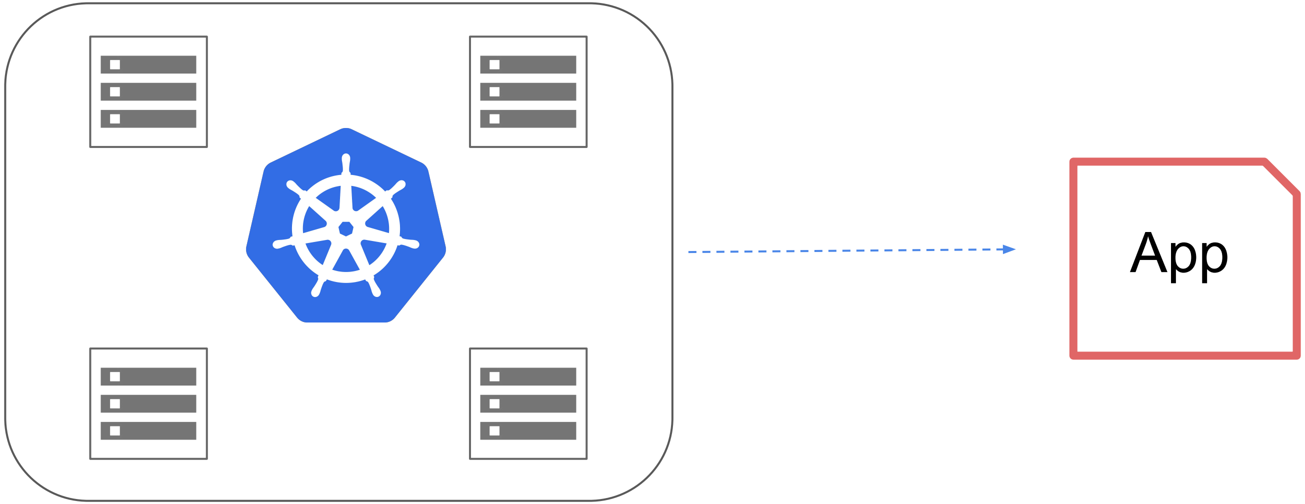
Kubernetes master

How does it work?



How does it work?

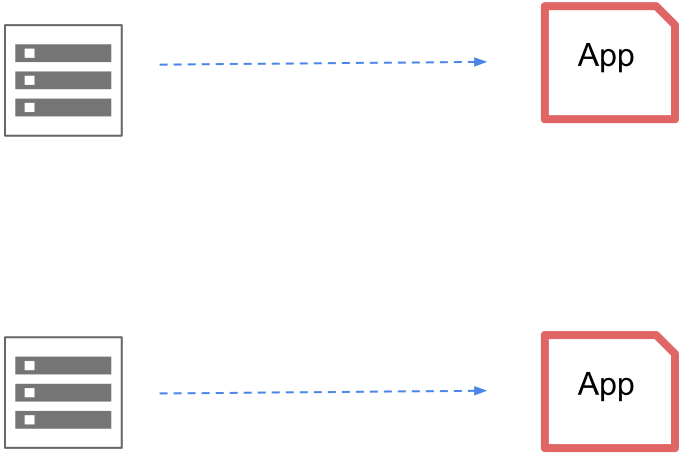
Deployment



App

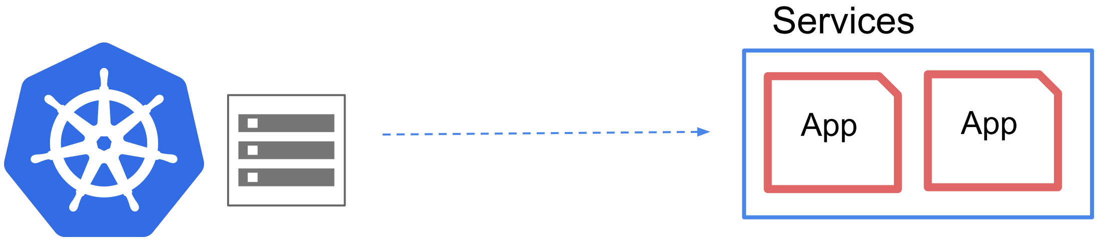
How does scaling work?

Naive Scaling

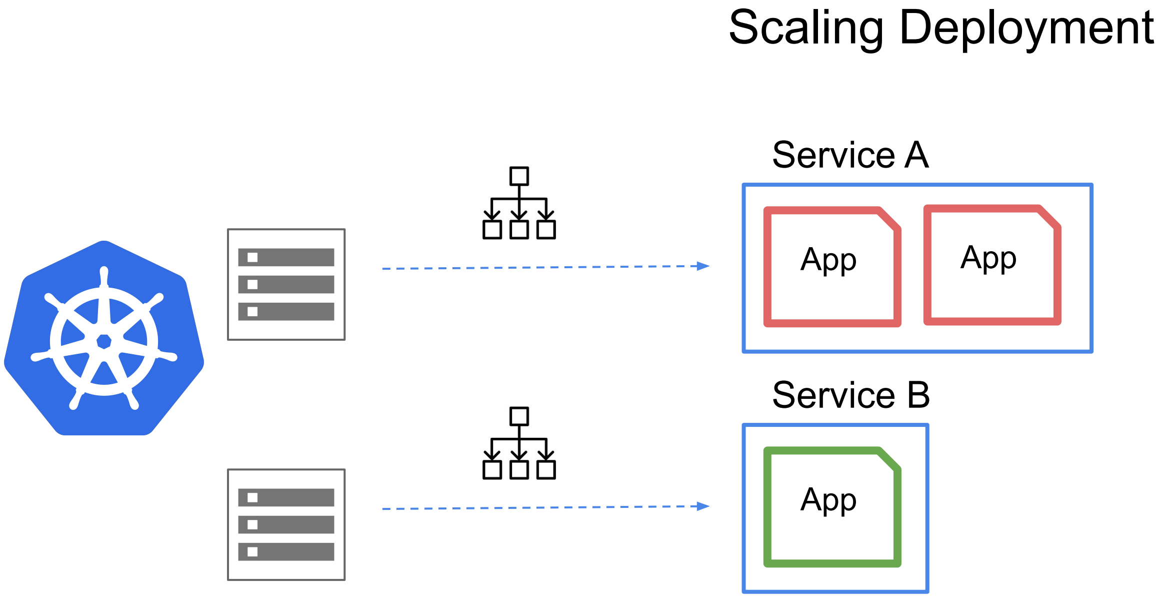


How does scaling work?

Scaling Deployment

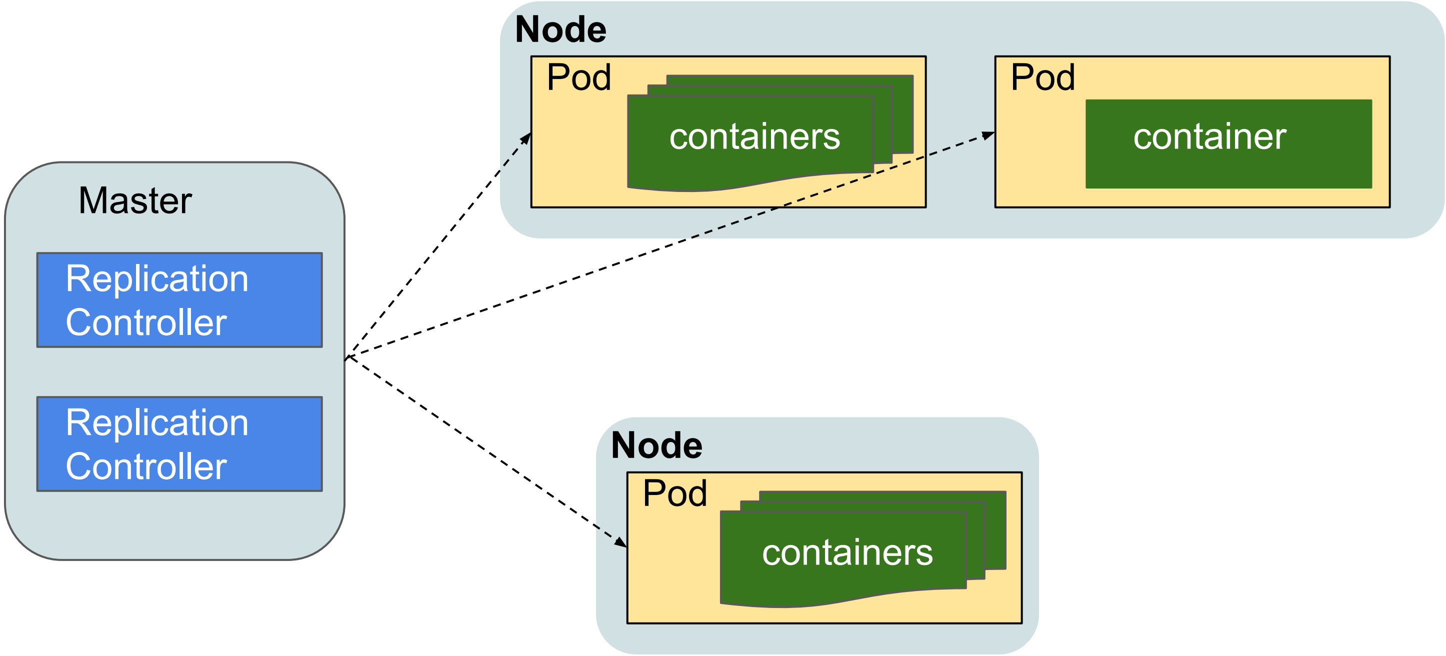


How does scaling work?



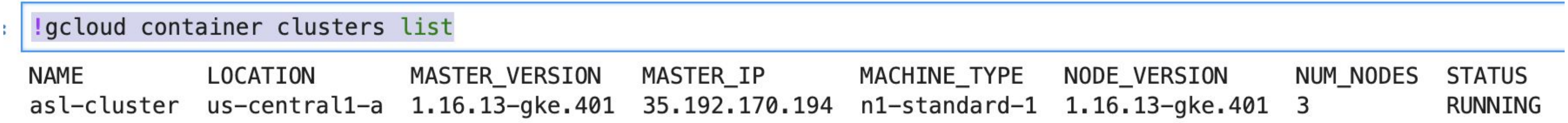
Terminology

- **Node** - an instance of a computer, running Kubernetes
  - Kubelet - application that is running, communicating with master node
  - communicates with master
- **Pod**
  - runs one or more containers
- **Service**
  - Handles requests either coming from inside Kubernetes cluster (node to node) or outside
  - Usually contains a load balancer - how requests should be routed and handled
- **Deployment** - defines desired state for Kubernetes
  - deployments keep the pods up and running even when the nodes they run on fail


Master
Replication Controller
Replication Controller
Node
Pod
containers
Pod
container
Node
Pod
containers

Creating a Kubernetes cluster on Google Cloud

```
gcloud container clusters create $CLUSTER_NAME
gcloud container clusters get-credentials $CLUSTER_NAME
```



```
!gcloud container clusters list
```

```
NAME         LOCATION       MASTER_VERSION   MASTER_IP       MACHINE_TYPE   NODE_VERSION     NUM_NODES  STATUS
asl-cluster  us-central1-a  1.16.13-gke.401  35.192.170.194  n1-standard-1  1.16.13-gke.401  3          RUNNING
```

Using kubectl to deploy a container

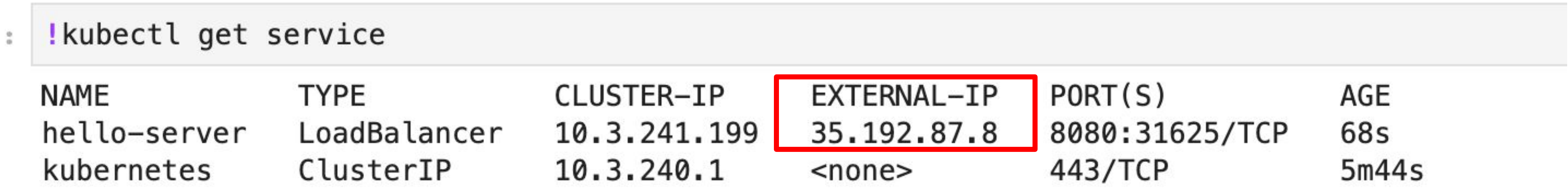
Container Image URI in the Container Registry

```
IMAGE_URI=gcr.io/google-samples/hello-app:1.0

kubectl create deployment hello-server --image=$IMAGE_URI
```

Creating a service to reach the deployed container

```
kubectl expose deployment hello-server --type=LoadBalancer --port 8080
```



```
!kubectl get service
```

```
NAME           TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
hello-server   LoadBalancer   10.3.241.199   35.192.87.8   8080:31625/TCP   68s
kubernetes     ClusterIP      10.3.240.1     <none>        443/TCP          5m44s
```

Lab 1

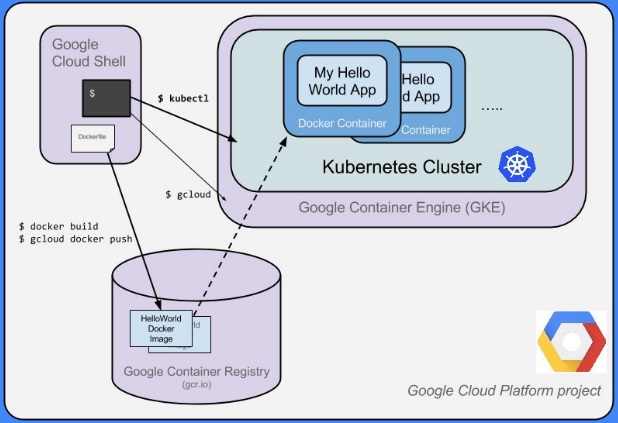
In this lab, you get hands on practice with container creation and application deployment with GKE.

docker_and_kubernetes/labs/2_intro_k8s.ipynb

Lab 2 (Optional)

In this lab, you'll

- Create a Node.js server.
- Create a Docker container image.
- Create a container cluster.
- Create a Kubernetes pod.
- Scale up your services.



docker_and_kubernetes/labs/3_k8s_hello_node.ipynb cloud.google.com