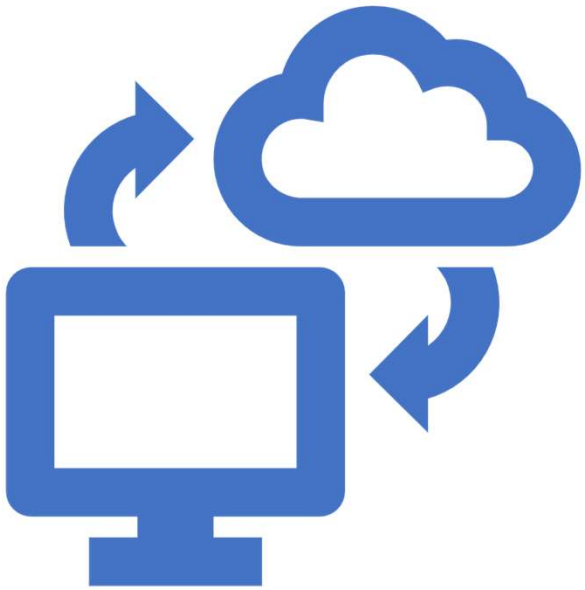
CSCI 5902
Adv. Cloud Architecting
Fall, 2023
Instructor: Dr. Lu Yang

Week 5 – Lec 1
Kubernetes Foundations
Oct 6, 2023

Housekeeping and Feedback

- Start recording
- Start learning GCP CI/CD tools, cloud shell, and Terraform
- PIER tour spreadsheet available Monday afternoon, Oct 9

Oct 10

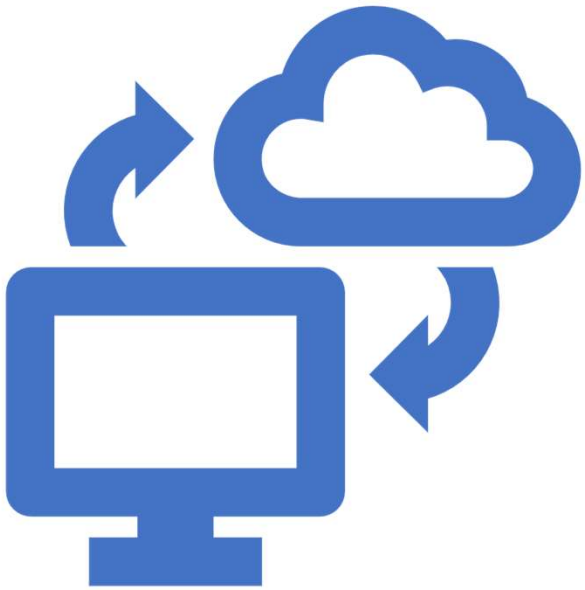


Part 1

Introduction to Kubernetes

Part 2

Kubernetes architecture



Part 1 Introduction to Kubernetes

- Containers on GCP
- Introduction to Kubernetes
- Introduction to Google Kubernetes Engine
- Google compute services
- Summary

How can you get or create containers?

Download containerized software from container registry or artifact registry such as gcr.io.

`docker`

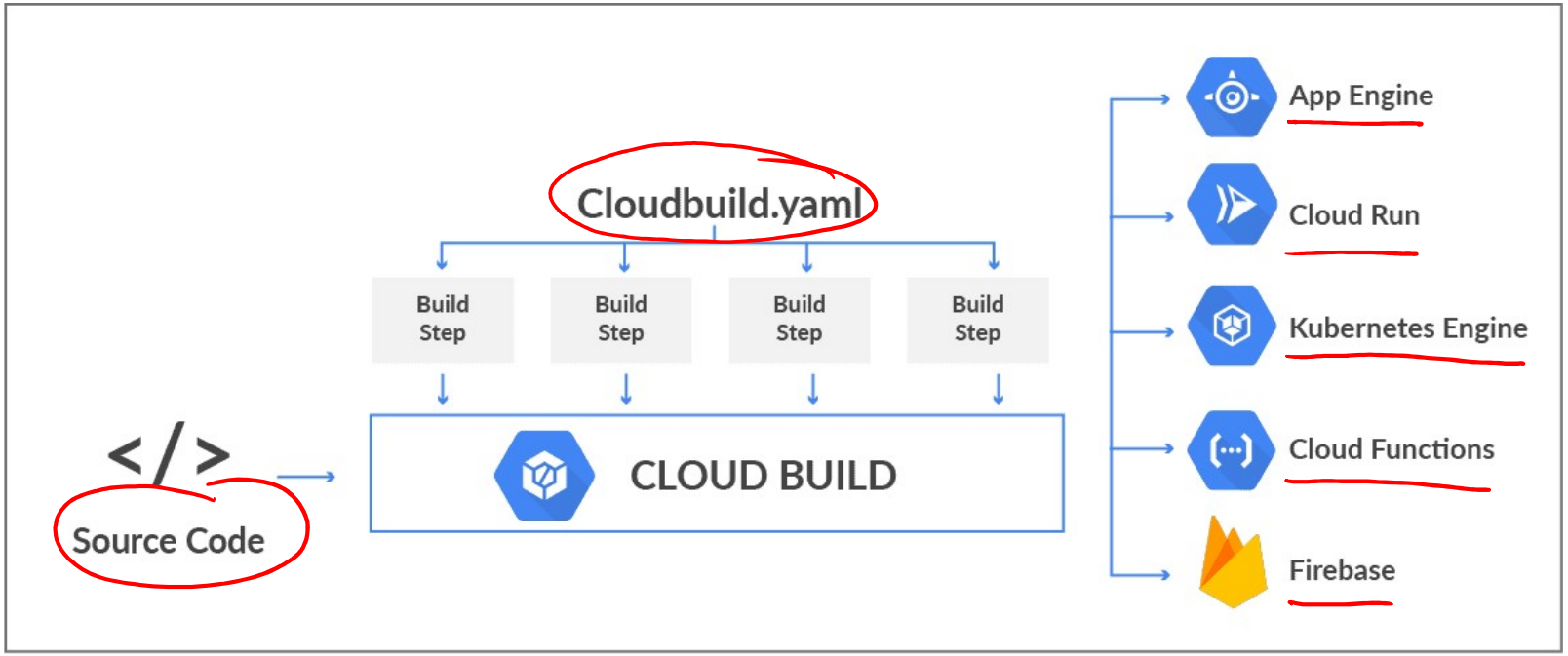
Build your own container using the open-source docker command.

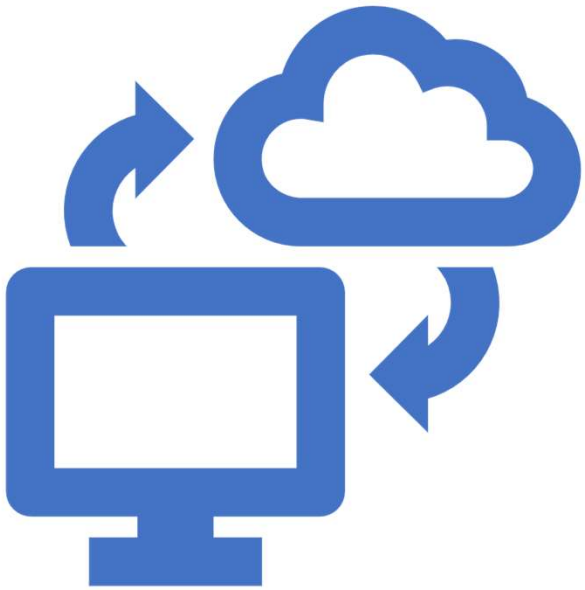
Build your own container using Cloud Build.

Artifact Registry is the recommended service for container image storage and management on Google Cloud. Artifact Registry provides the same container management features as Container Registry and includes additional features and benefits. As a fully-managed service with support for both container images and non-container artifacts, Artifact Registry extends the capabilities of Container Registry.

https://cloud.google.com/artifact-registry/docs/transition/transition-from-gcr

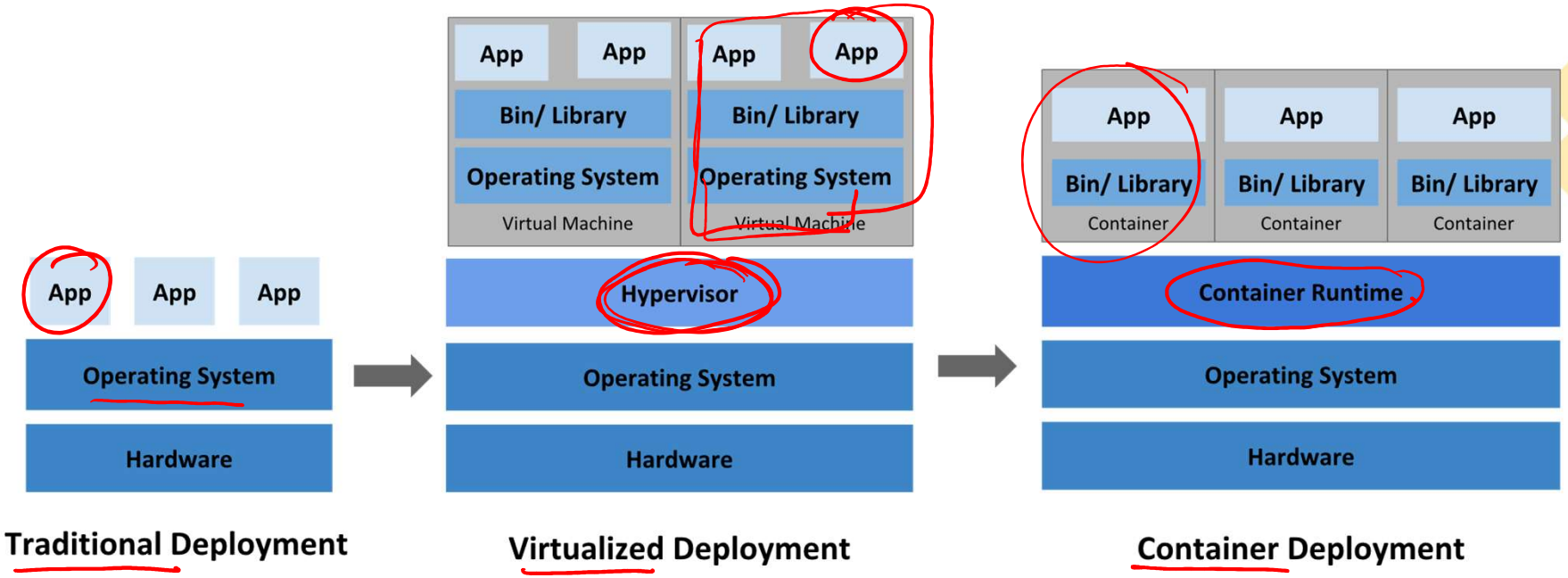
Build and deploy containers on GCP





Part 1 Introduction to Kubernetes

- Introduction to Containers
- Introduction to Kubernetes
- Introduction to Google Kubernetes Engine
- Google compute services
- Summary



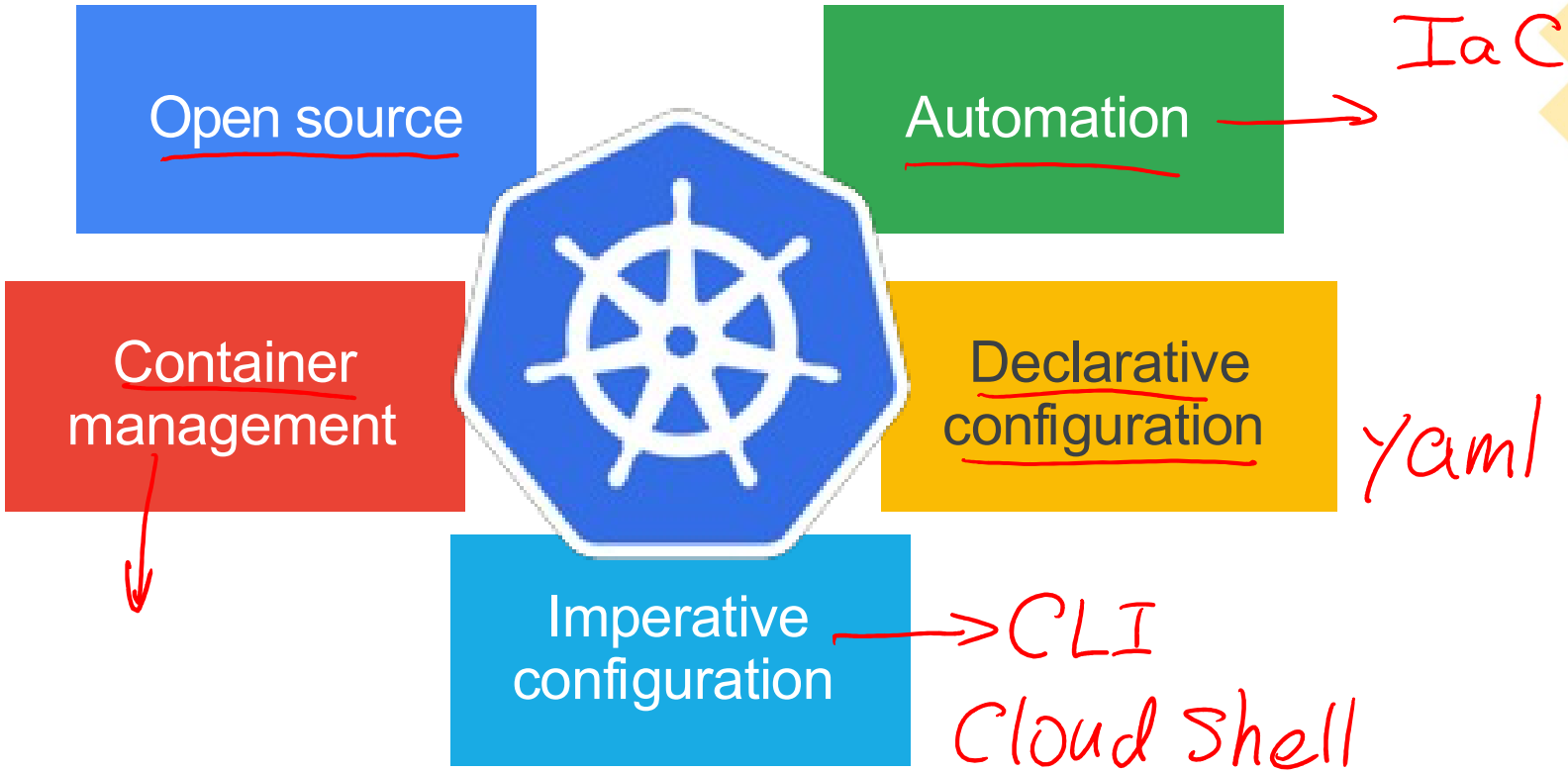
How do you better manager your container infrastructure?

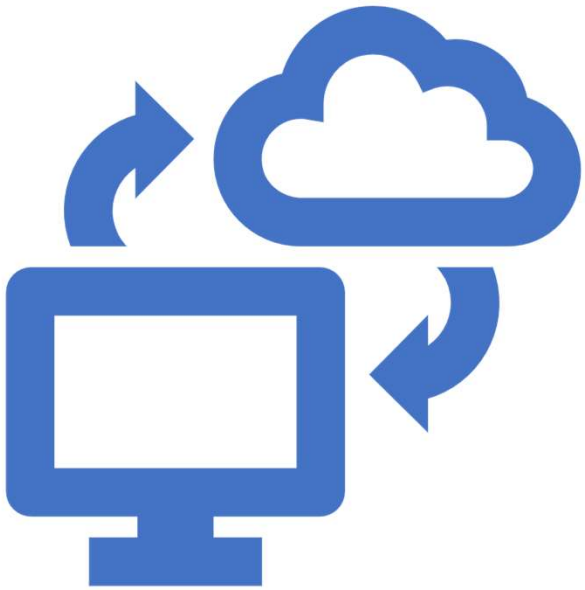
Kubernetes! (https://kubernetes.io/docs/concepts/overview/)

What does Kubernetes do?

- **Service discovery and load balancing** Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable.
- **Storage orchestration** Kubernetes allows you to automatically mount a storage system of your choice, such as local storages, public cloud providers, and more.
- **Automated rollouts and rollbacks** You can describe the desired state for your deployed containers using Kubernetes, and it can change the actual state to the desired state at a controlled rate. For example, you can automate Kubernetes to create new containers for your deployment, remove existing containers and adopt all their resources to the new container.
- **Automatic bin packing** You provide Kubernetes with a cluster of nodes that it can use to run containerized tasks. You tell Kubernetes how much CPU and memory (RAM) each container needs. Kubernetes can fit containers onto your nodes to make the best use of your resources.
- **Self-healing** Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.
- **Secret and configuration management** Kubernetes lets you store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration.

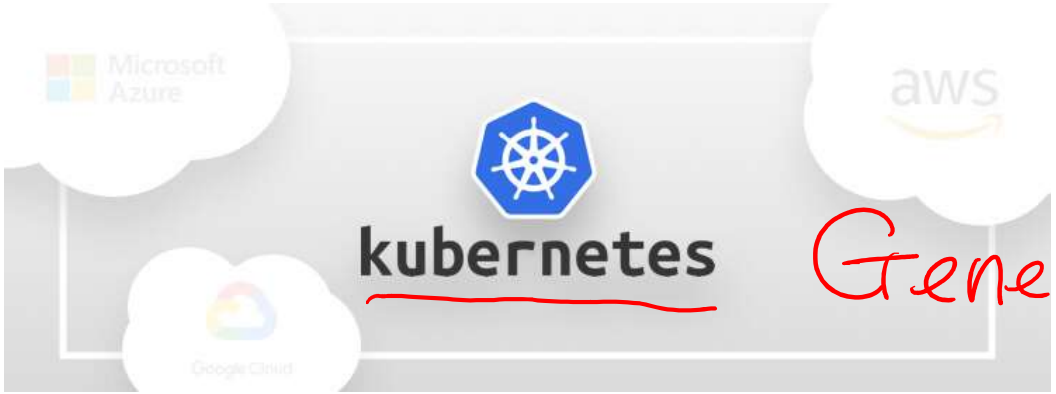
What is Kubernetes?



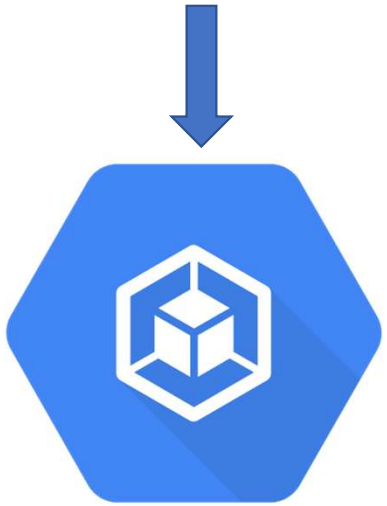


Part 1 Intro to Kubernetes

- Introduction to Containers
- Introduction to Kubernetes
- Introduction to Google Kubernetes Engine
- Google compute services
- Summary



General



Google Kubernetes Engine

- GKE is a managed Kubernetes service on Google infrastructure. GKE helps you to deploy, manage, and scale Kubernetes environments for your containerized applications on Google Cloud.

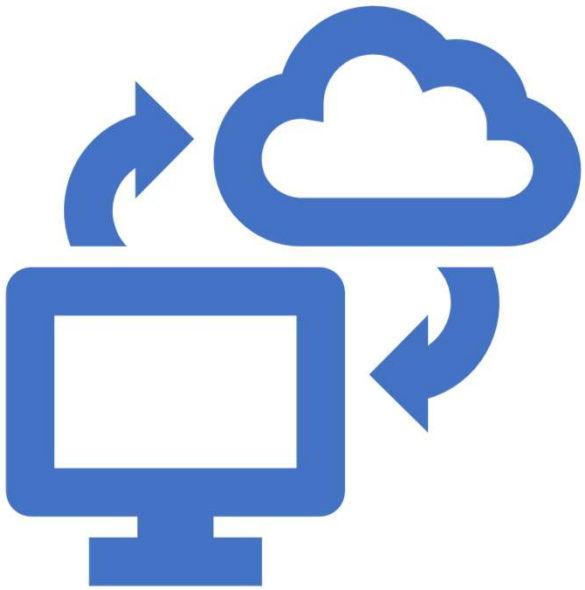
- More specifically, GKE is a component of the Google Cloud compute offerings. It makes it easy to bring your Kubernetes workloads into the cloud.

Features of GKE

Fully managed	Container-optimized OS	Auto upgrade
Auto repair	Cluster scaling	Seamless integration
Identity and access management	Integrated logging and monitoring	Integrated networking
	Cloud Console	

IAM

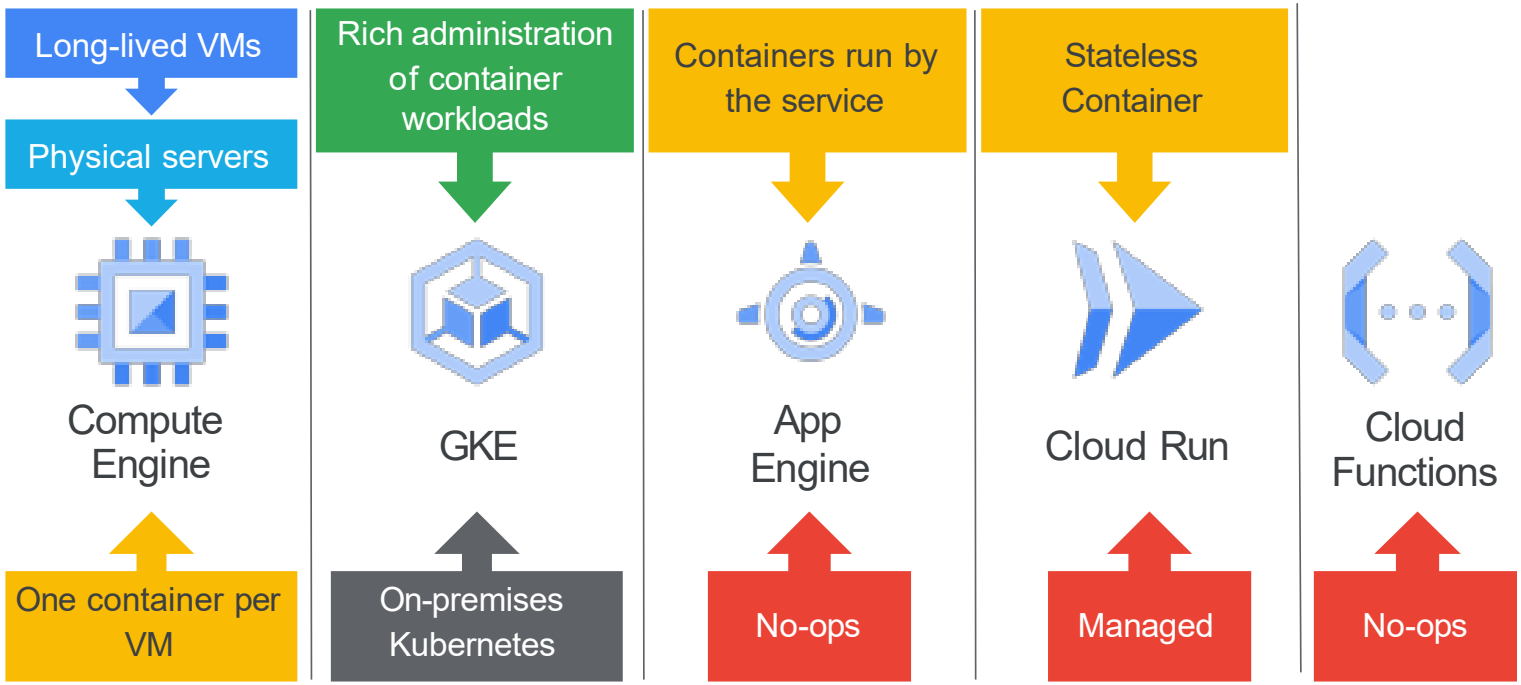
Shell

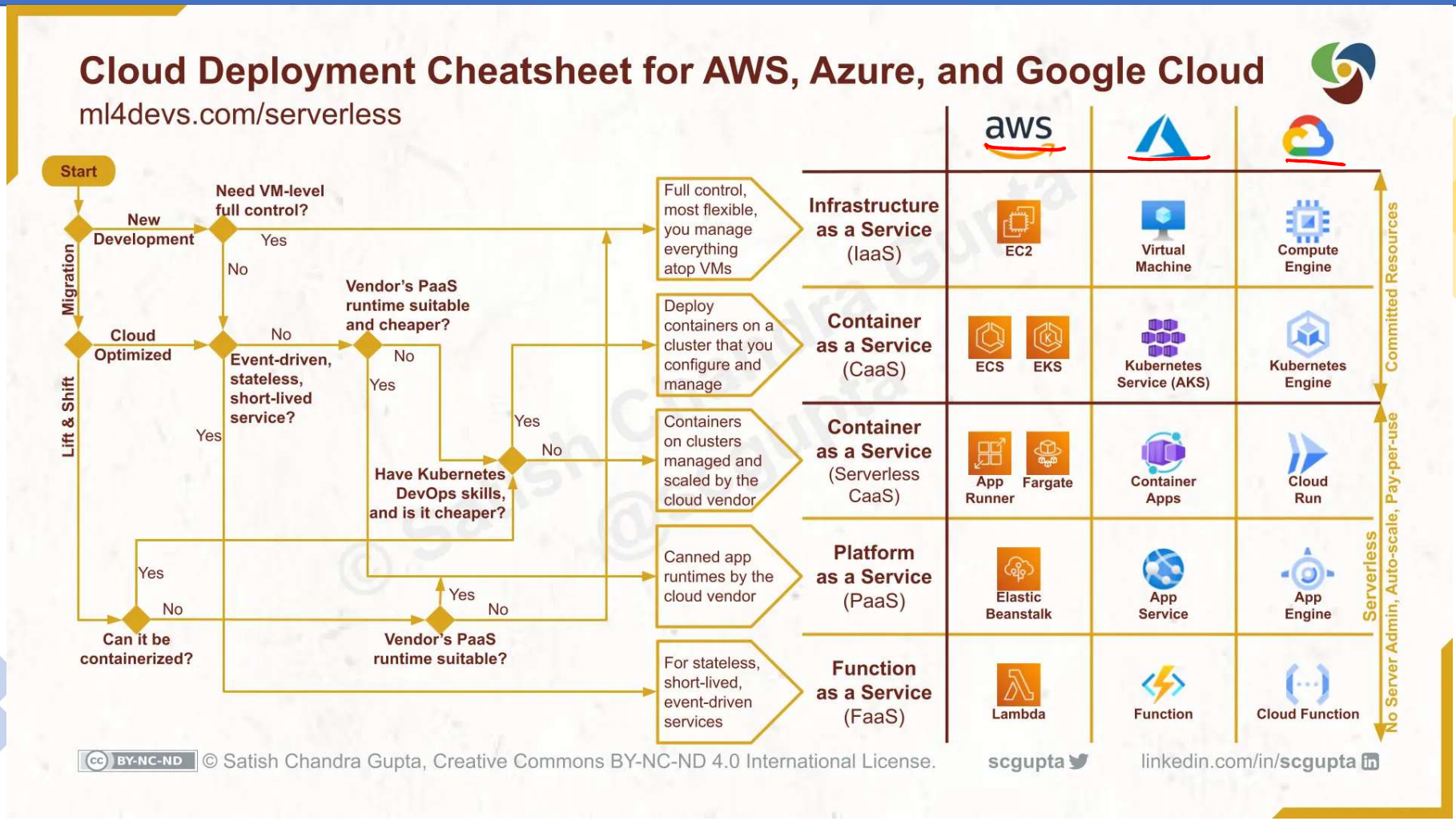


Part 1 Intro to Kubernetes

- Introduction to Containers
- Introduction to Kubernetes
- Introduction to Google Kubernetes Engine
- Google compute services
- Summary

Which compute service should you adopt?





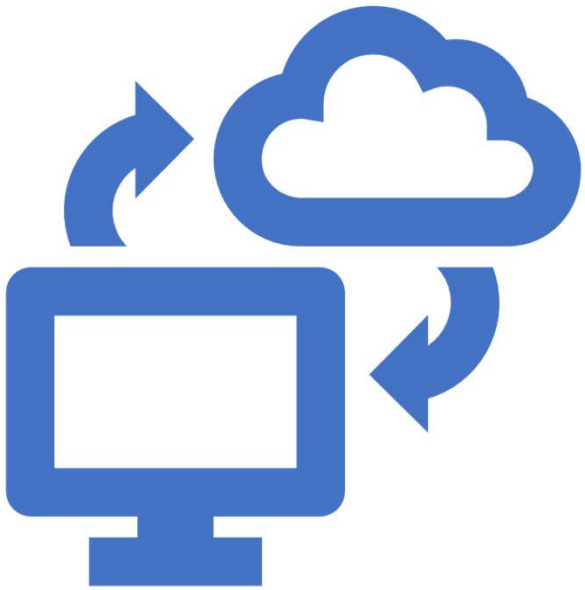
https://www.ml4devs.com/articles/serverless-architecture-for-microservices-on-aws-vs-google-cloud-vs-azure-as-iaas-caas-paas-faas/

Google Kubernetes Engine (GKE)

- Fully managed Kubernetes platform.
- Supports cluster scaling, persistent disks, automated upgrades, and auto node repairs. hard drive/EBS
- Built-in integration with Google Cloud services.
- Hybrid computing
- Multi-cloud computing

GKE use cases

- Containerized applications.
- Cloud-native distributed systems.
- Hybrid applications.



Part 1 Intro to Kubernetes

- Introduction to Containers
- Lab: Working with Cloud Build Introduction to Kubernetes
- Introduction to Kubernetes
- Introduction to Google Kubernetes Engine
- Computing Options
- Summary

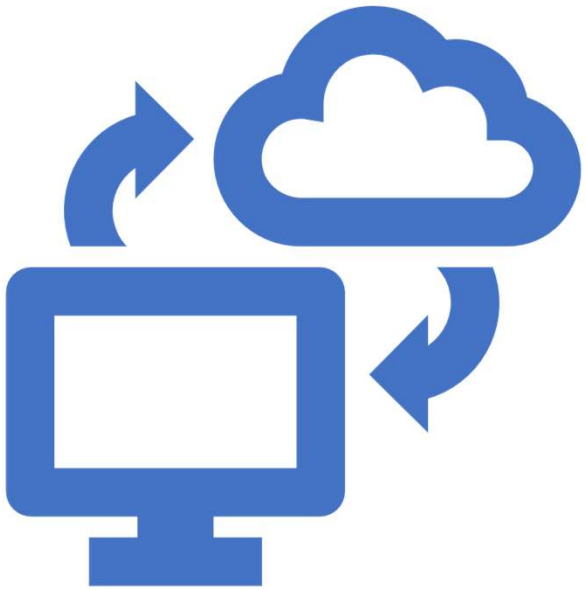
Summary

Create a container using Cloud Build. Google Cloud compute solutions:

- Compute Engine
- App Engine
- Google Kubernetes Engine
- Cloud Run
- Cloud Functions

Store a container in Artifact Registry.

Compare and contrast Kubernetes and Google Kubernetes Engine (GKE) features.



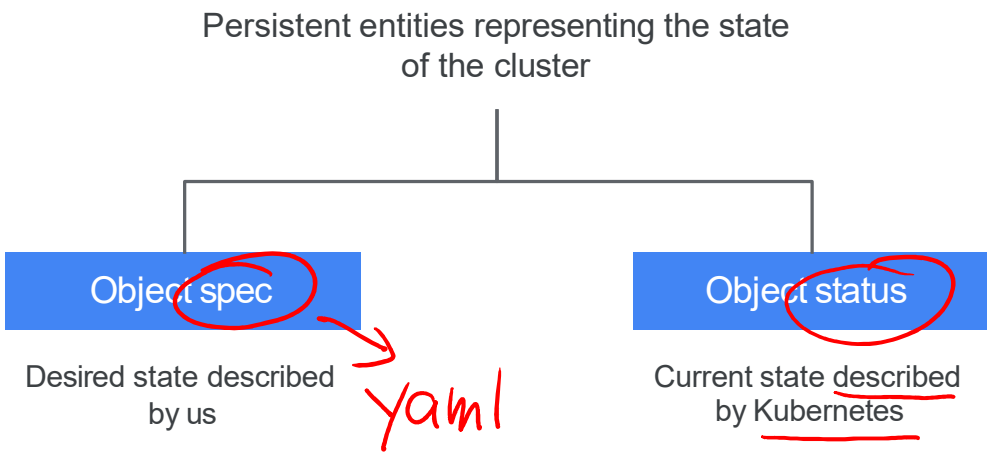
Part 2 Kubernetes Architecture

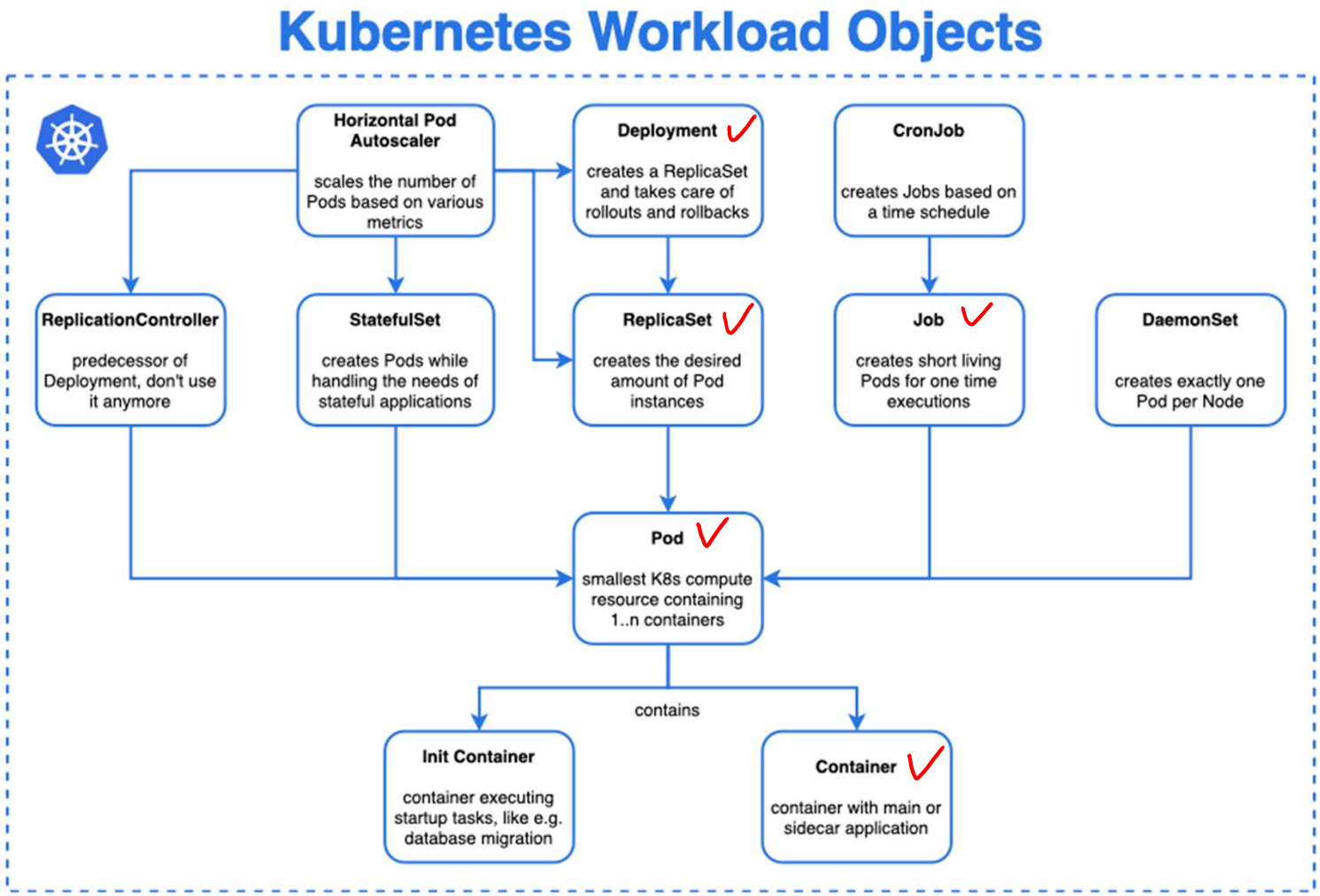
- Kubernetes Concepts
- GKE Architecture
- Object Management
- Lab: Deploying Google Kubernetes Engine
- Summary

Kubernetes objects

A Kubernetes object is defined as a persistent entity that represents the state of something running in a cluster: its desired state and its current state.

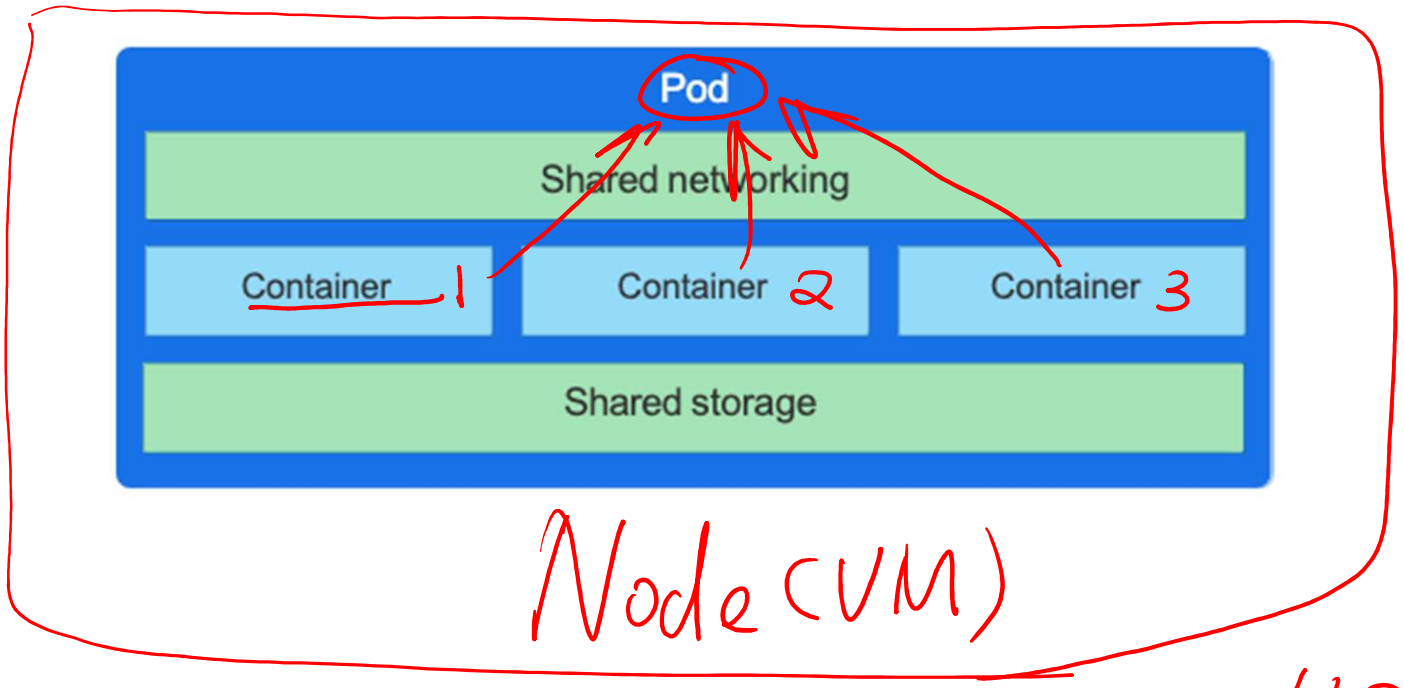
There are two elements to Kubernetes objects





https://shipit.dev/posts/kubernetes-overview-diagrams.html

Containers in a Pod share resources



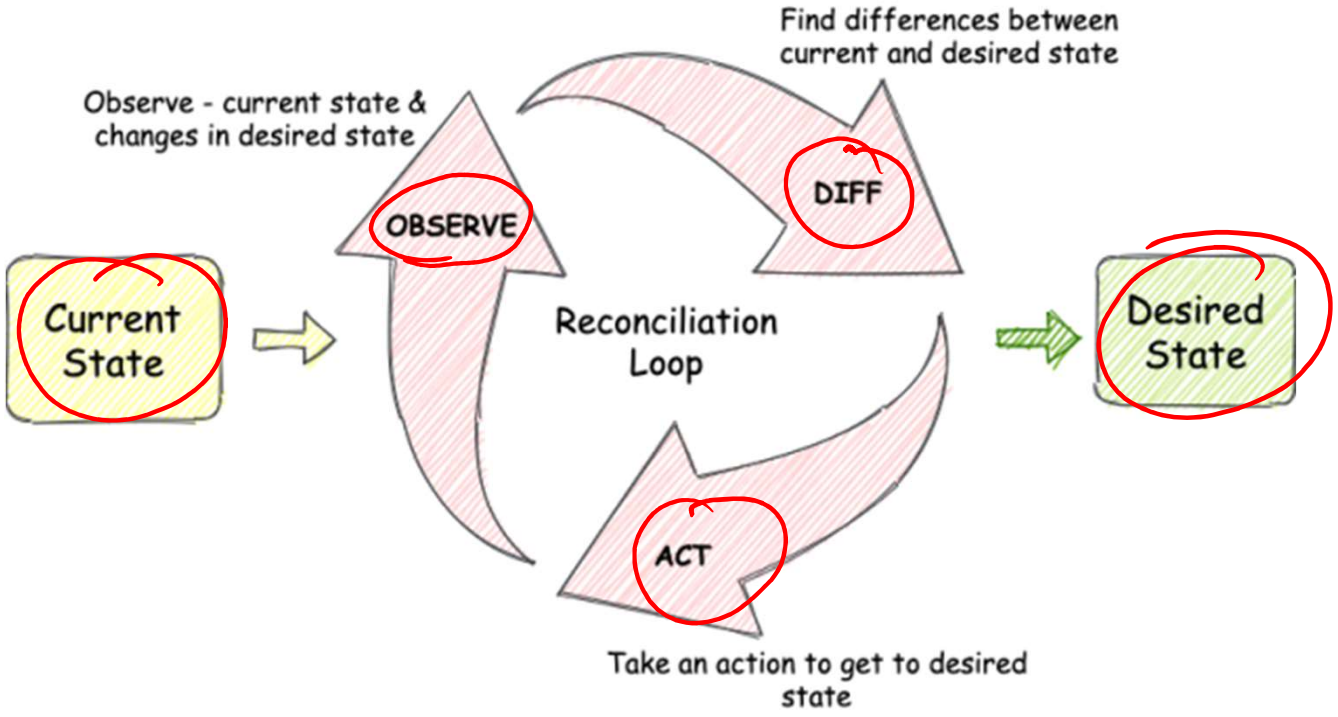
Running three nginx containers

You want three nginx containers running all the time

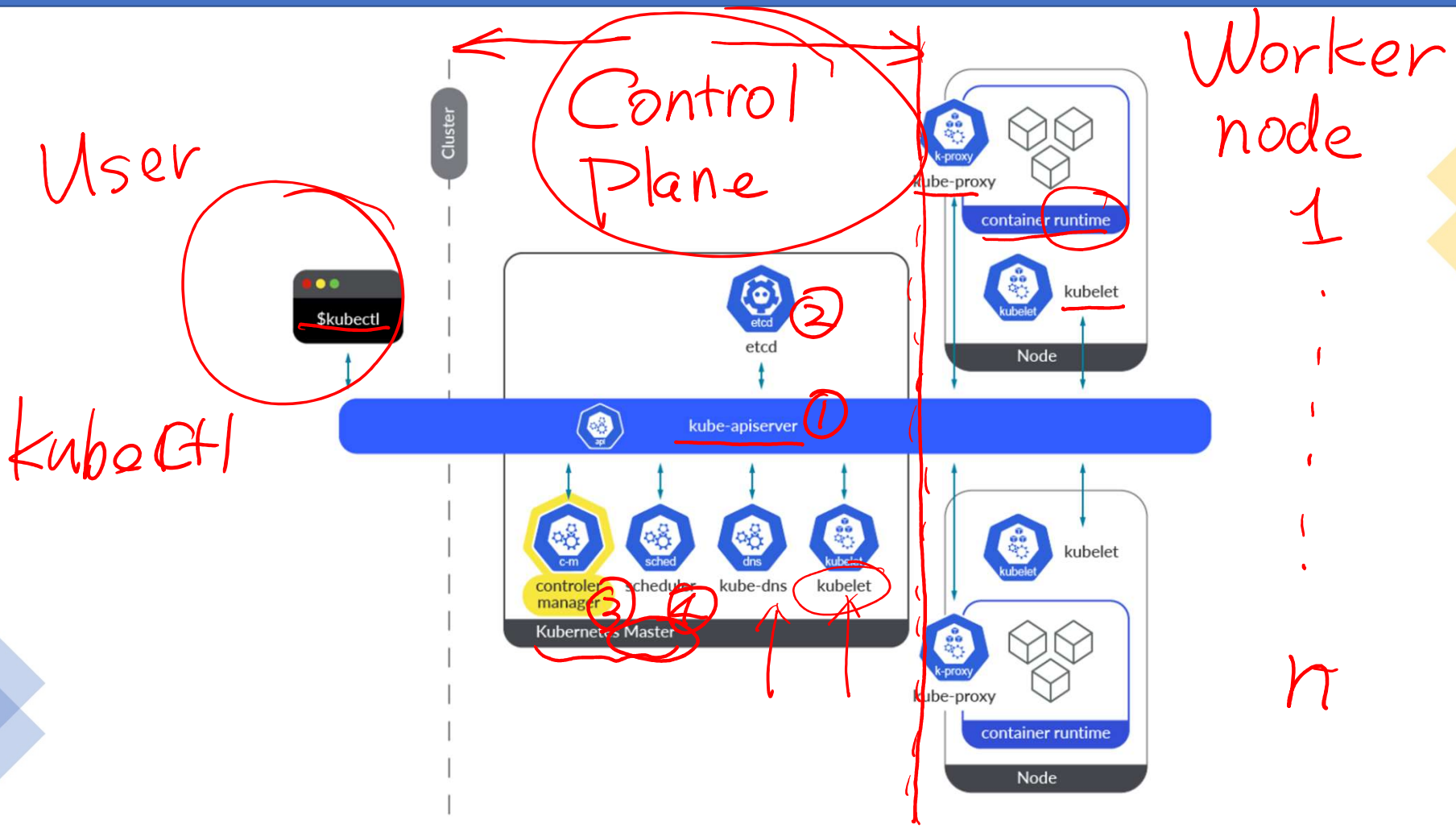
You declare objects that represent those containers

Kubernetes launches those objects and maintains them

Desired state compared to current state

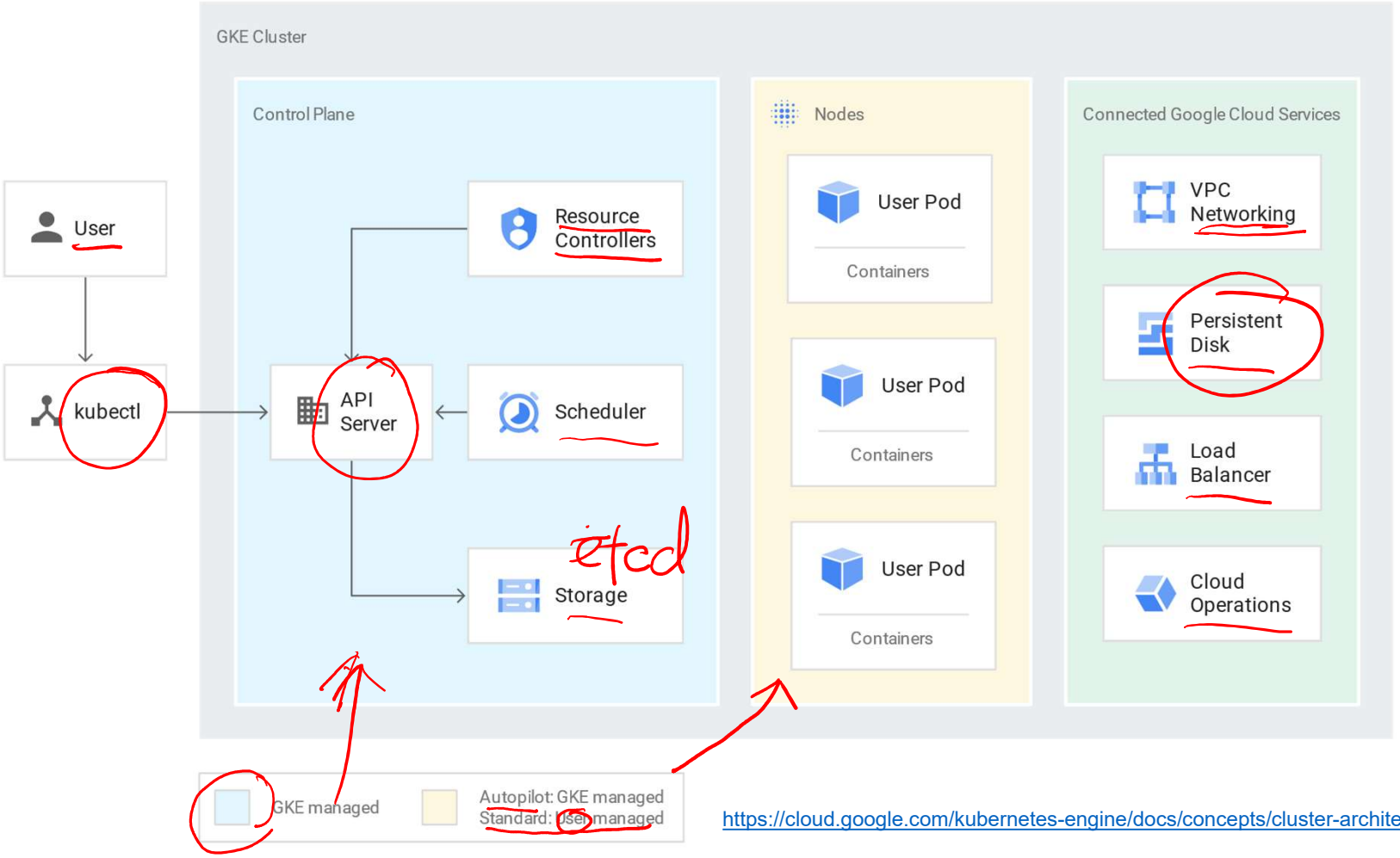


Part 2 Kubernetes Architecture



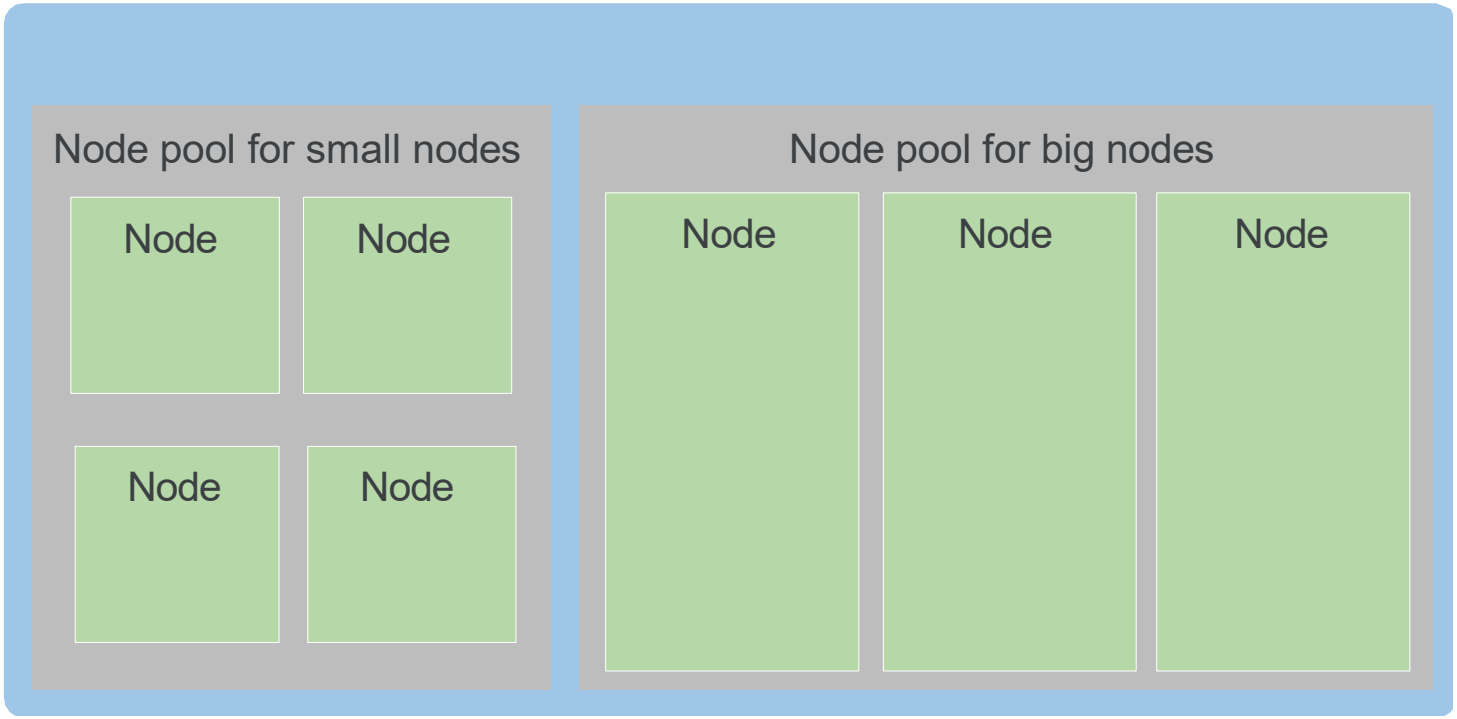
Part 2 Kubernetes Architecture

GKE Architecture



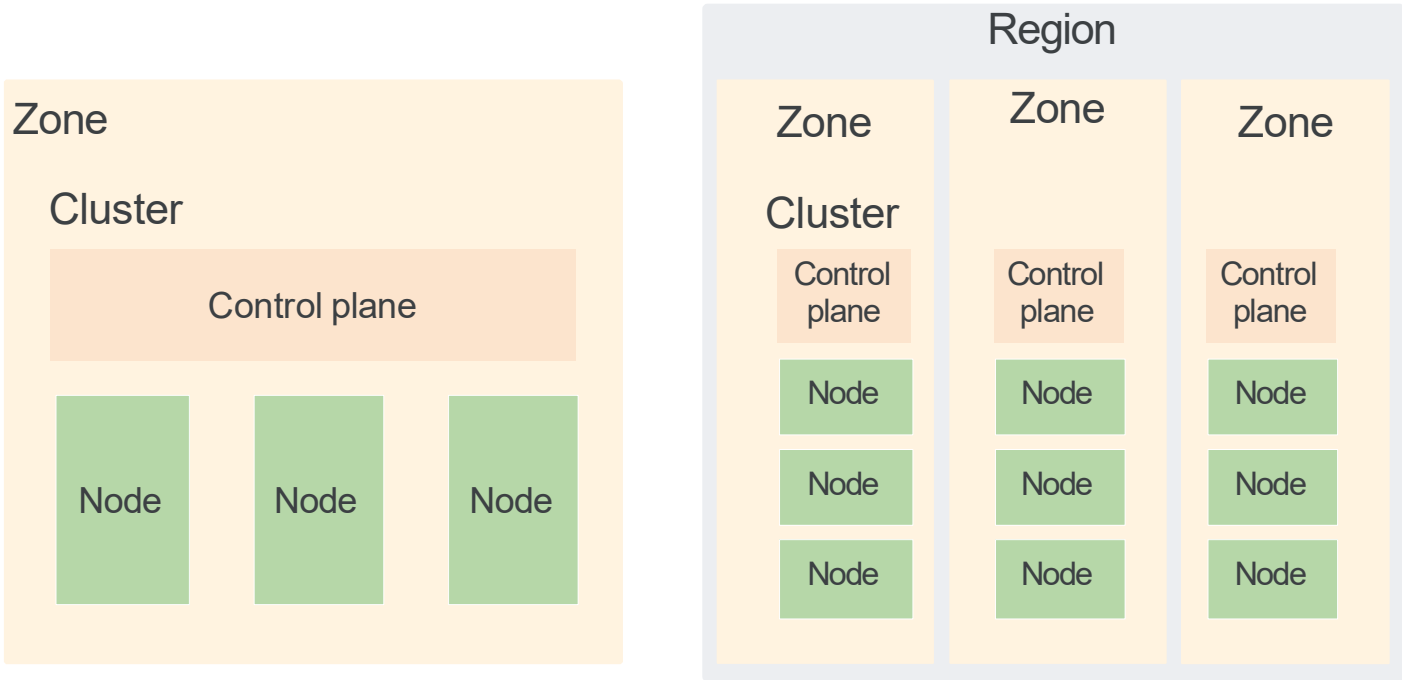
https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-architecture

Node pools (GKE feature)

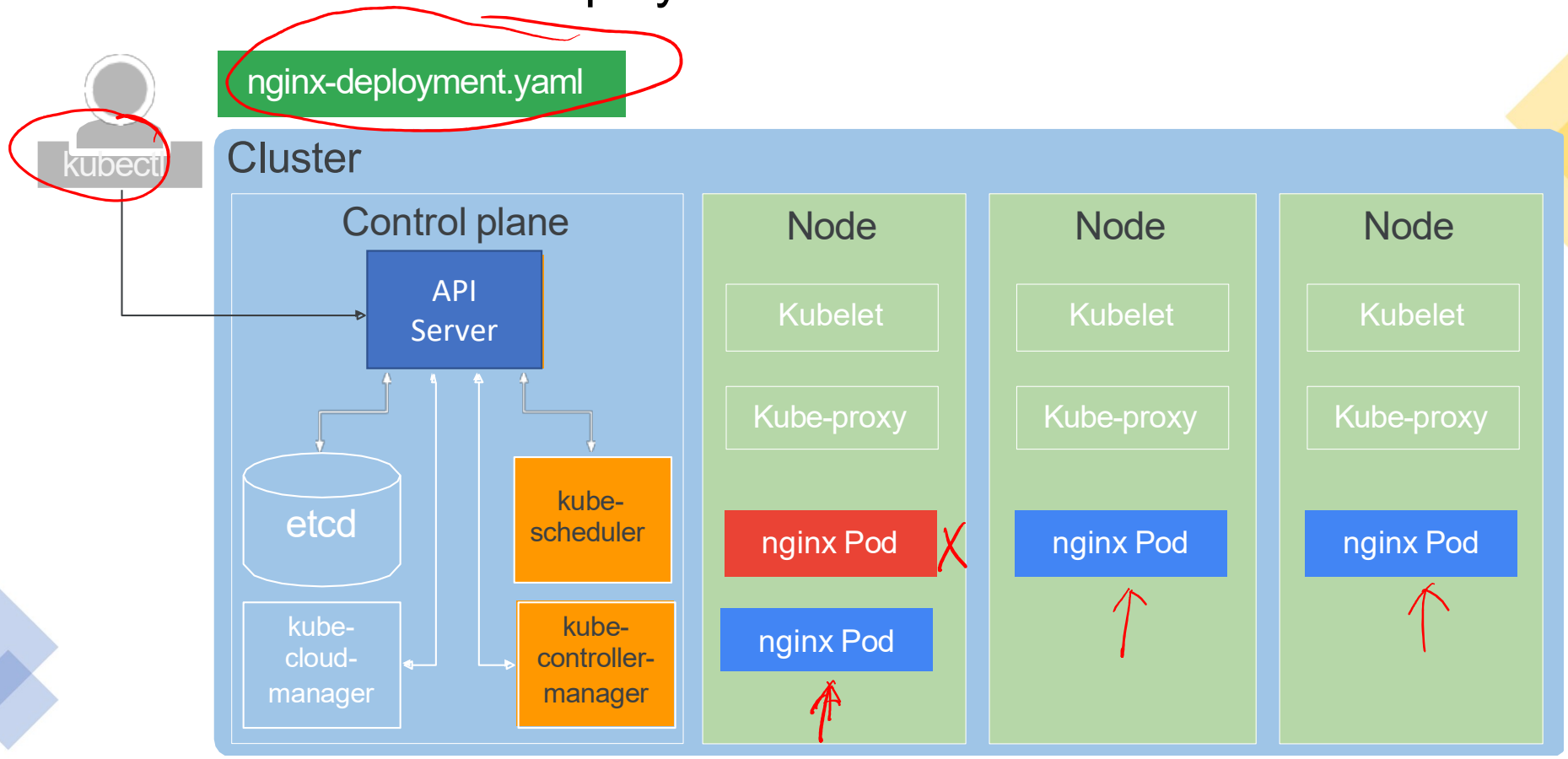


https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-architecture

Zonal and regional clusters

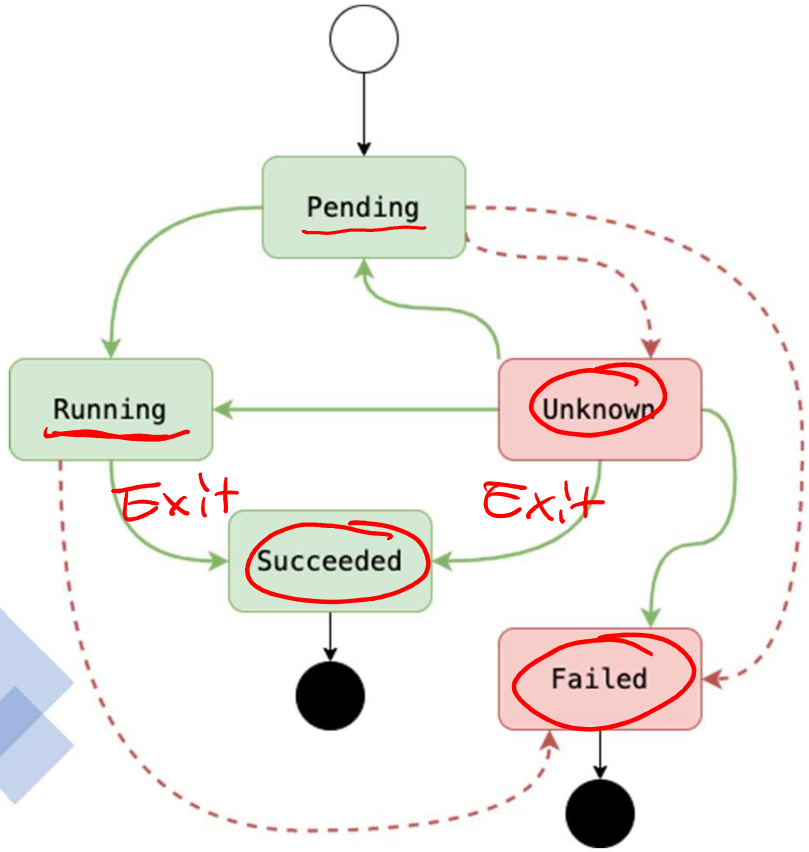


Deployment of a workload



https://kubernetes.io/docs/concepts/architecture/cloud-controller/

The life cycle of a pod



Pod phases

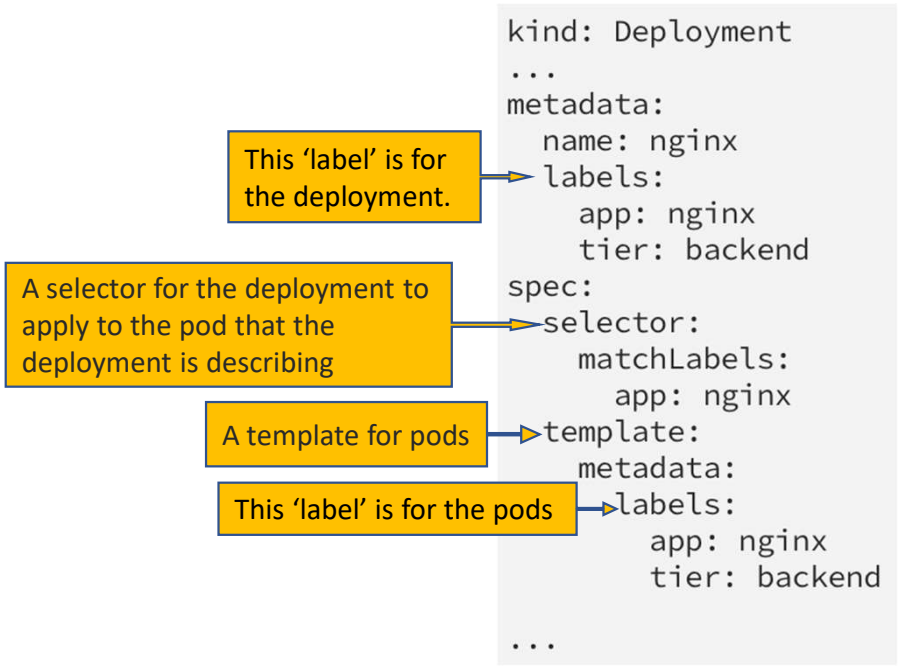
Pending	Pod has been created by the cluster, but one or more of its containers are not yet running. This phase includes time spent being scheduled on a node and downloading images
Running	The Pod has been allotted to a node; all the containers have been created. At least one container is still running, or is in the process of starting or restarting
Succeeded	All containers in the Pod have terminated successfully
Failed	One or more containers terminated with non-zero status
Unknown	The state of the Pod cannot be determined. This occurs due to error while communicating with the node

Container states

Waiting	When the container still pulling image, applying Secret data etc.
Running	When the container executing without any issues
Terminated	When the container exited with non-zero status

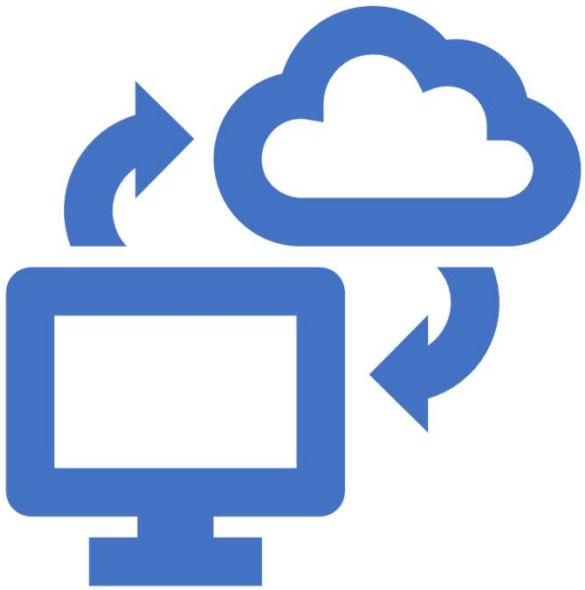
http://millionvisit.blogspot.com/2021/03/kubernetes-for-developers-9-Kubernetes-Pod-Lifecycle.html

Example yaml files



Create a GKE cluster and deploy a workload

https://cloud.google.com/kubernetes-engine/docs/deploy-app-cluster doc
https://www.youtube.com/watch?v=p2LyoePiBo8

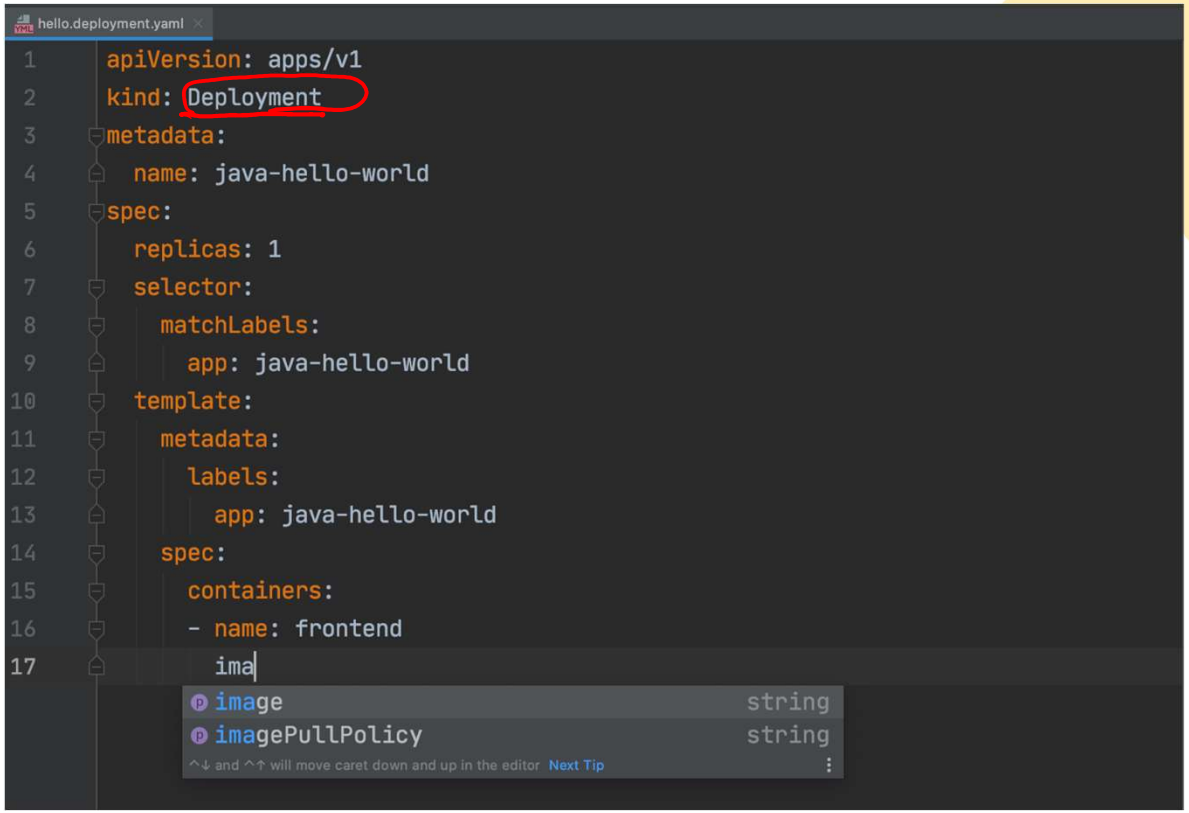


Part 2 Kubernetes Architecture

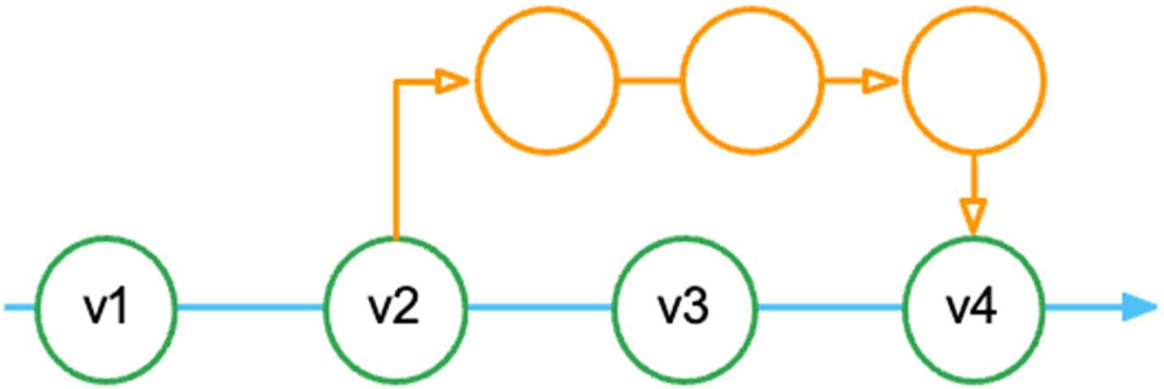
- Kubernetes Concepts
- Kubernetes Components
- Object Management
- Lab: Deploying Google Kubernetes Engine
- Summary

Objects are defined in a YAML file
All objects are identified by a name

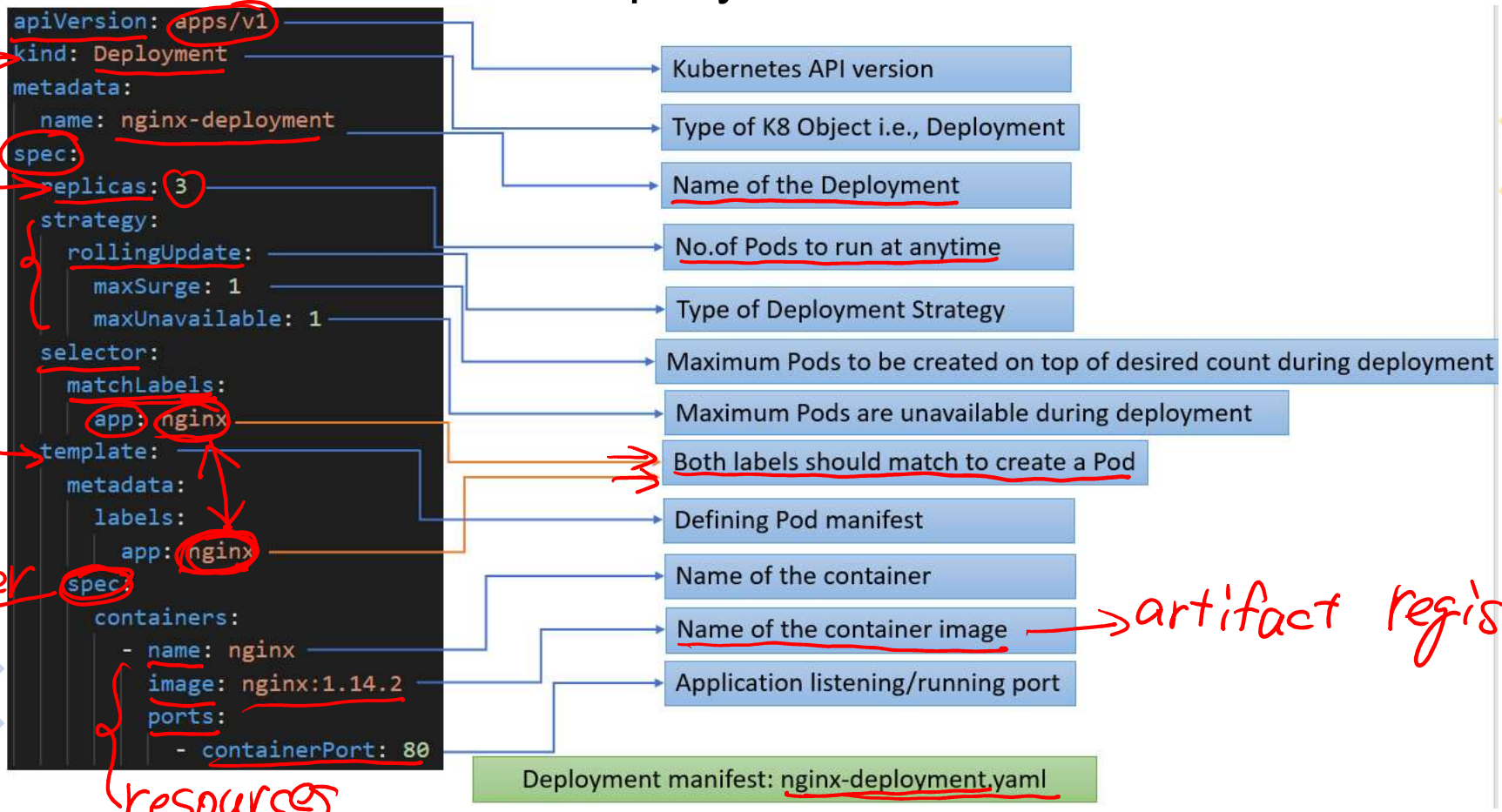
```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx
spec:
  replicas: 3
  selector:
    app: nginx
  template:
    metadata:
      name: nginx
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```



Best practice tip: Use version control on YAML files



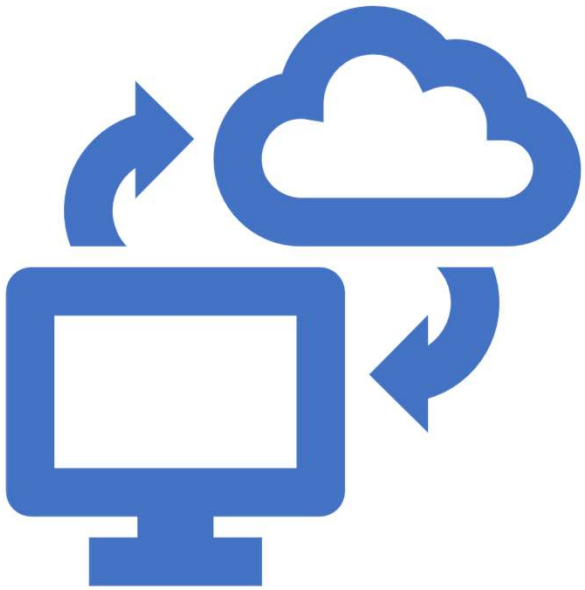
Example yaml file



http://millionvisit.blogspot.com/2021/05/kubernetes-for-developers-14-Kubernetes-Deployment-YAML-manifest.html

Resource management for Pods and Containers

- ✓ Important that containers have enough resources to run.
- ✓ Applications could use more resources than they should.
- ✓ CPU and memory (RAM) resources are the most common resources specified.



Part 2 Kubernetes Architecture

- Kubernetes Concepts
- Kubernetes Components
- Object Management
- Lab: Deploying Google Kubernetes Engine
- Summary

Summary

Kubernetes controllers keep the cluster state matching the desired state.

Kubernetes consists of a family of control plane components, running on the control plane and the nodes.

GKE abstracts away the control plane.

Declare the state you want using manifest files.

Yaml