

# **Deploy to Kubernetes from GitHub Container Registry**

# Hey! I'm Chamod

Find me at [@chamodshehanka](https://twitter.com/chamodshehanka)  
Visit [chamodshehanka.com](https://chamodshehanka.com)



# Table of Content

- What are Containers
- What is GitHub Container Registry
- Why GitHub Container Registry
- What's Kubernetes
- Advantages of Kubernetes
- CD Pipeline to Deploy
- Demo
- QnA



# What are Containers?

@chamodshehanka

**foss  
asia**



Learn from memes

# GitHub Container Registry

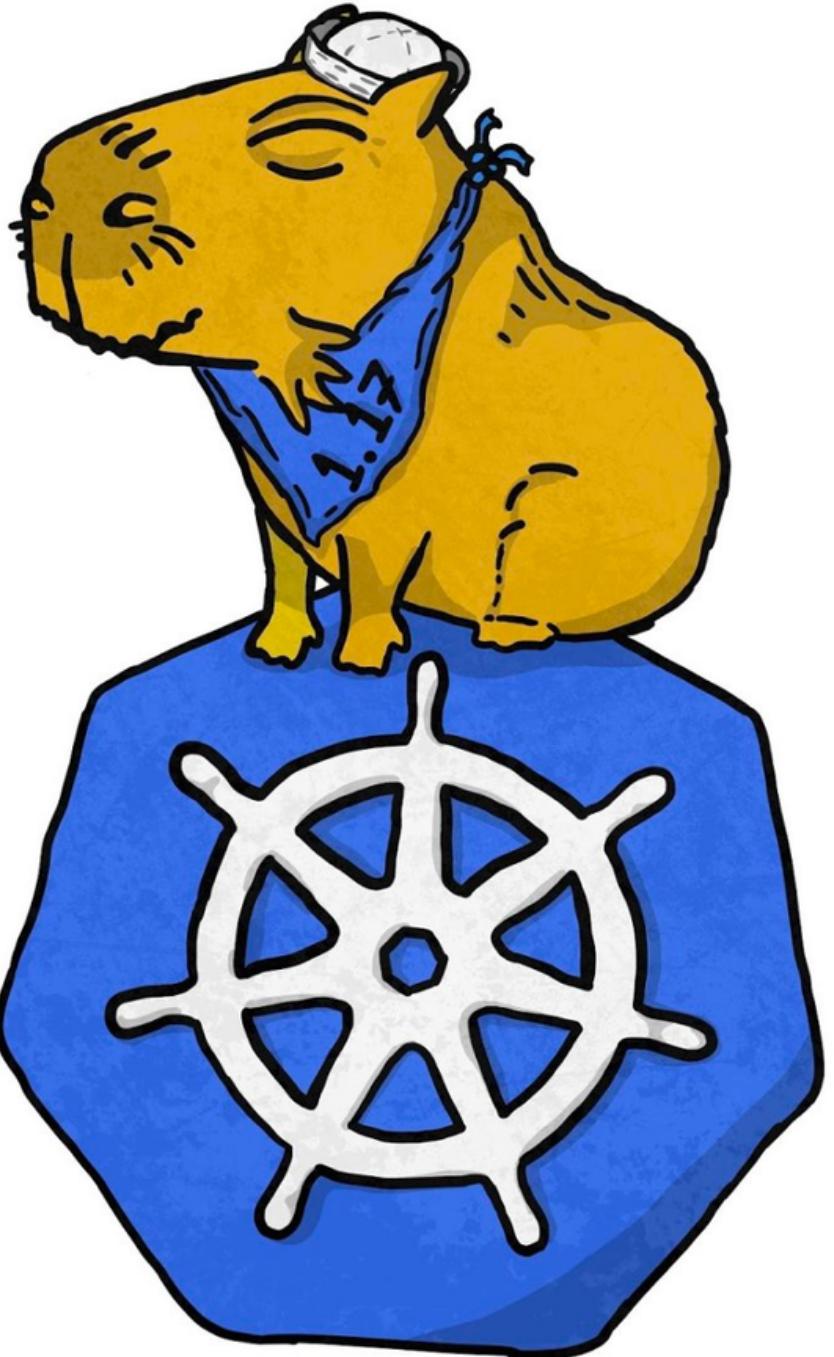
You can store and manage Docker and OCI images in the Container registry, which uses the package namespace **<https://ghcr.io>**



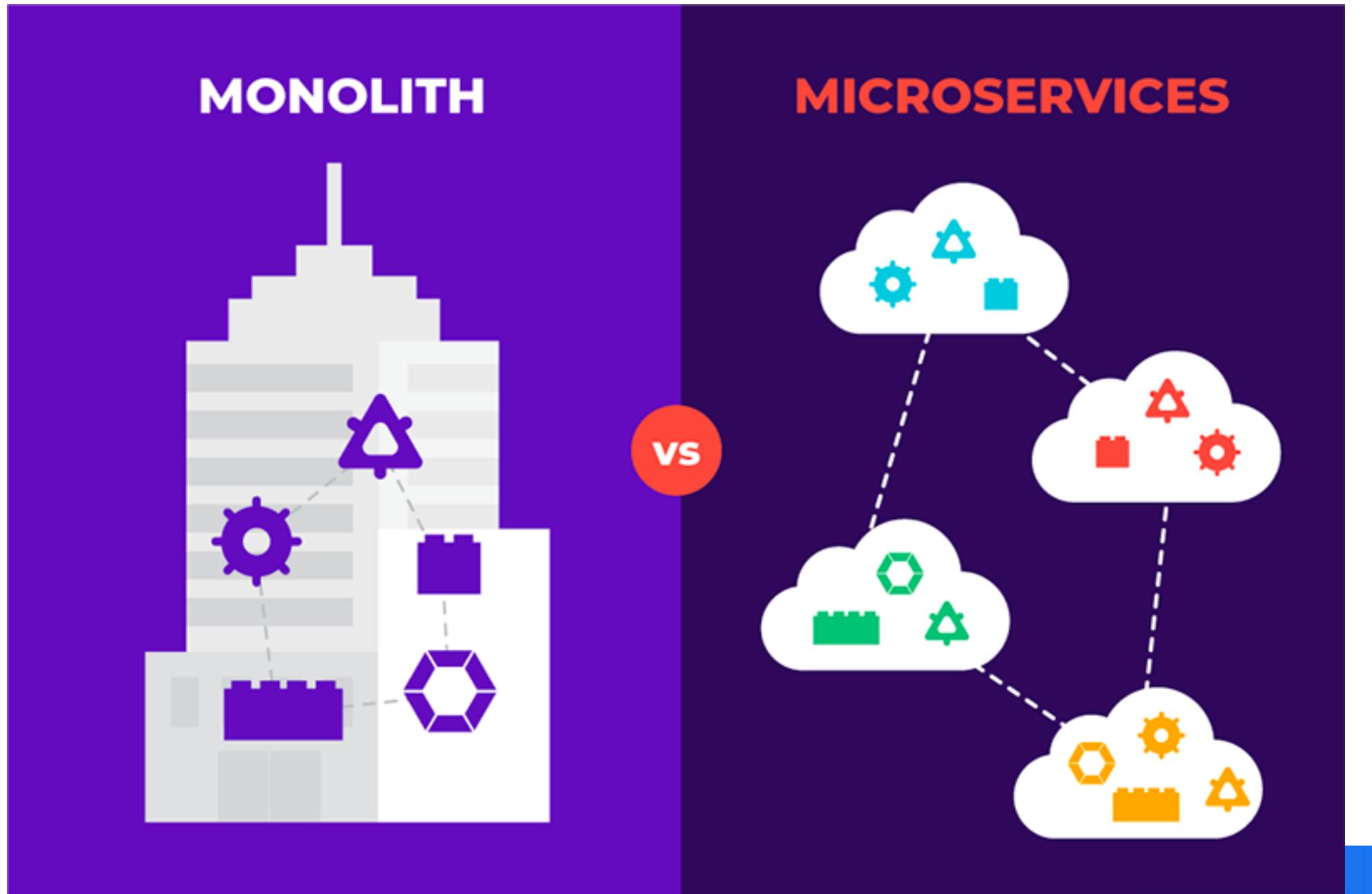
# Definition of Kubernetes

Open Source orchestration engine for automating deployments, scaling and management of containerized apps.

Initially developed by **Google**  
Now maintaining by **CNCF - Cloud Native Computing Foundation**

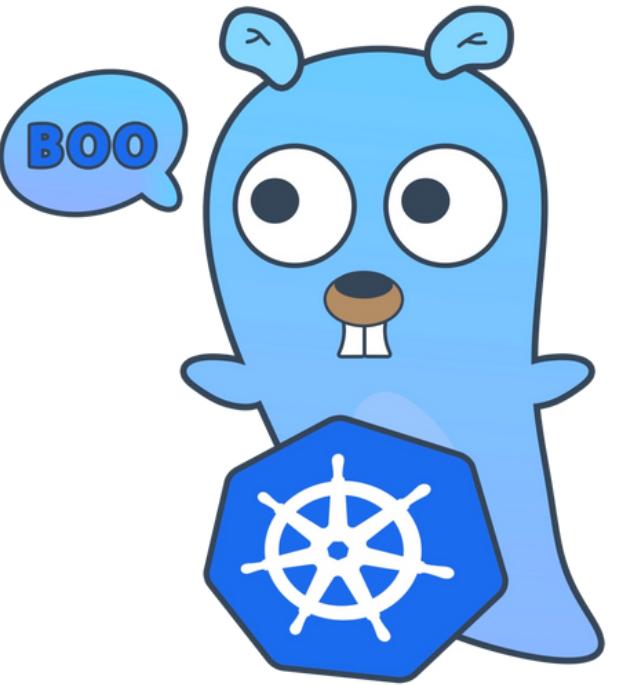


# Monolith vs Microservices



# Why Kubernetes (K8s)

- Highly Available
- Scalability
- Disaster Recovery
- and more....



# Basic Concepts in K8s

- Nodes - worker machines that run containers
- Pods - Pods are the smallest deployable units in Kubernetes
- Deployments - Deployments provide declarative updates for pods
- Services - Services provide a stable IP address and DNS name for a set of pods
- ConfigMaps and Secrets - ConfigMaps and Secrets are Kubernetes objects that store configuration data and sensitive information
- Namespaces - Namespaces provide a way to divide a cluster into multiple virtual clusters

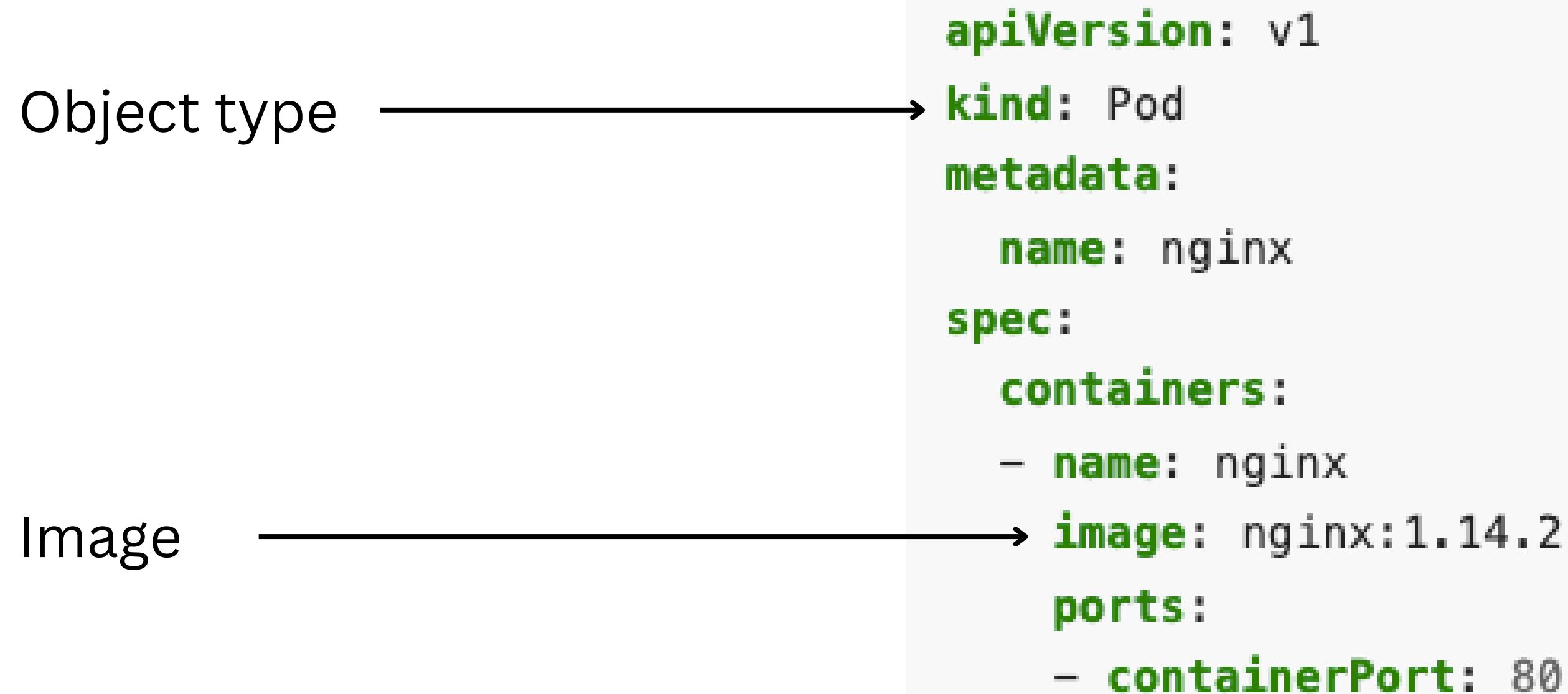
# Kubernetes yaml definition

Object type

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```

Image

# Kubernetes yaml definition



# 2 Ways to Create K8s Objects

## 1. Imperative

```
kubectl [command] [TYPE] [NAME] [flags]
```

```
kubectl get pods --sort-by=.metadata.name
```

## 2. Declarative

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
  ports:
  - containerPort: 80
```



# Continuous Delivery (C/D) Pipeline



# Demo

@chamodshehanka



**foss  
asia**

# Where to next?

- Try Docker, GHCR and Kubernetes.
- Play with Docker - <https://www.docker.com/play-with-docker>
- Level up skills with GitHub Skills (<https://skills.github.com/>)
- Use minikube for practice Kubernetes

# Any Questions?



Follow me  
@chamodshehanka



# Thank You