

# Kubernetes Charm

## Introduction



# Index

1. Introduction
  - a. What is Kubernetes?
  - b. What is a charm?
  - c. What is Juju
  - d. What is a Kubernetes charm?
2. Setting up the environment
3. Building Layers
4. Interfaces
5. Useful commands

# What is Kubernetes?

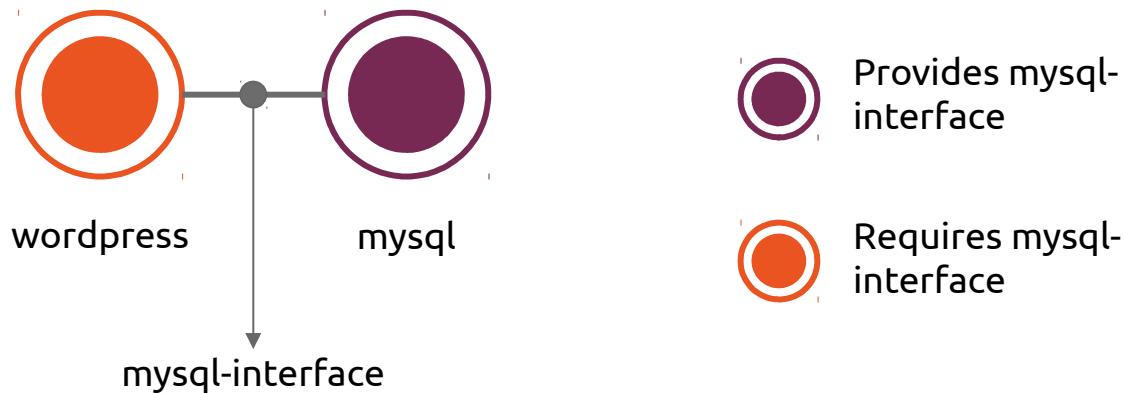
- Kubernetes is an open-source platform for managing containerized workloads and services
  - Container-centric management environment
  - Computing, networking and storage orchestration
- This tutorial aims to teach how to use Kubernetes as the backend cloud in which the charms are deployed
  - Each charm is deployed as a [Pod](#) in Kubernetes



# What is a charm?

- A charm is a set of scripts for deploying and operating software
  - Event handling built in
  - It's organized by layers → Helps reusing code
  - It can provide/require interfaces to exchange data with other charms
  - Utilizes Juju to deploy across multiple substrates

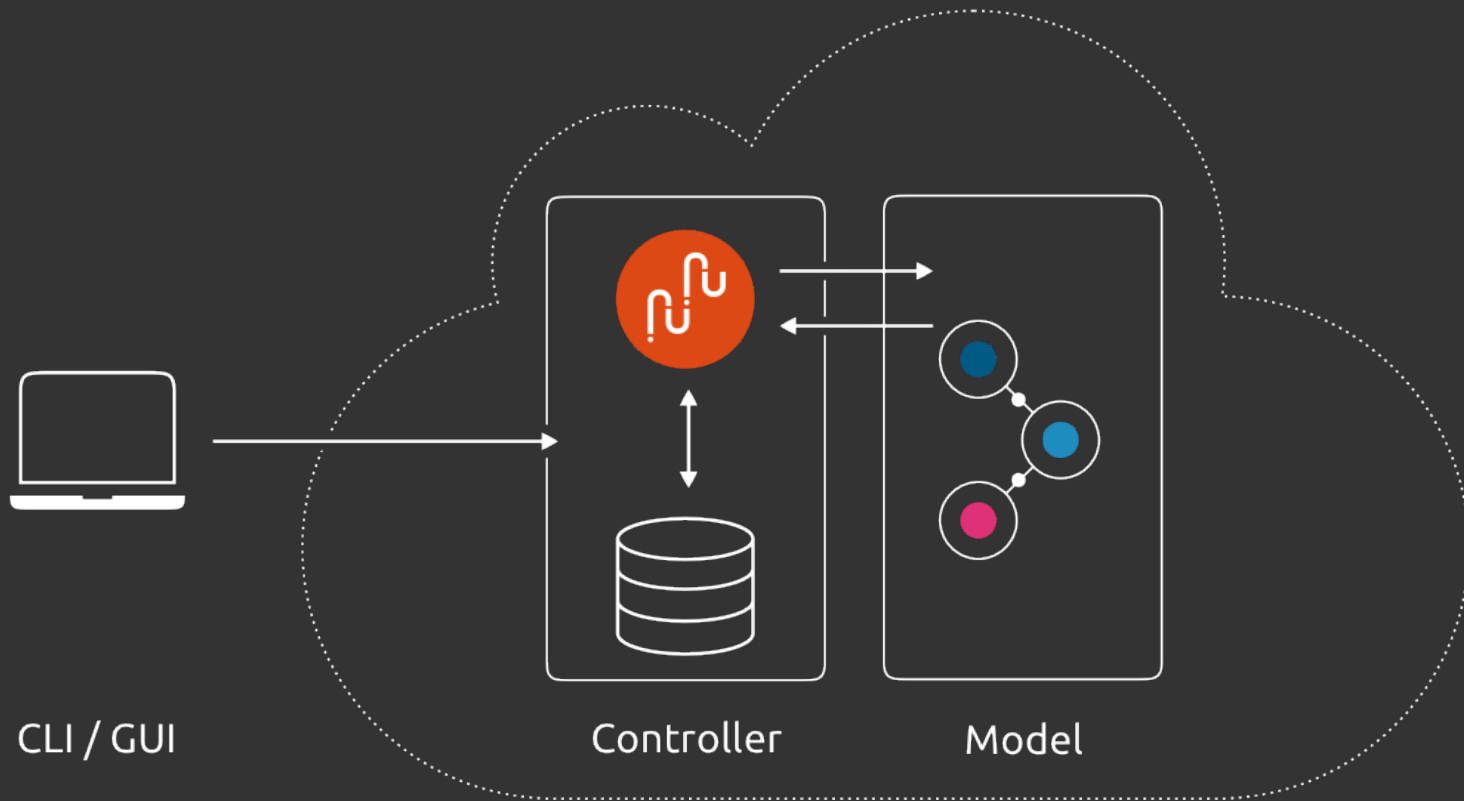
- Example:



# What is Juju?

- Juju is an open source modeling tool, composed of a controller, models, and charms, for **operating software in the cloud**.
- Juju can handle configuration, **relationships** between services, **lifecycle** and **scaling**.
- This ensures that common elements such as databases, messaging systems, key value stores, logging infrastructure and other 'glue' functions are available as charms for automatic integration, reducing the burden on vendors and integrators.

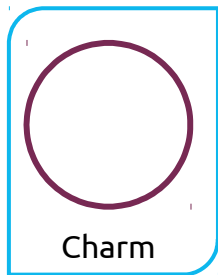
# What is Juju?



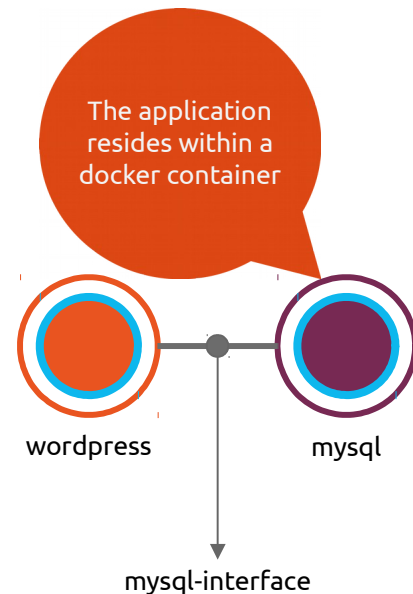
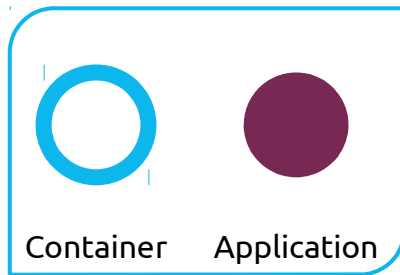
# What is a Kubernetes charm?

- A Kubernetes charm runs as an **operator** in Kubernetes
  - Pull docker image
  - Make relationships with other charm operators
  - Instantiate pod using the pulled image
- Kubernetes charms are extremely simple
  - The complexity resides within the docker images

Operator Pod



Workload Pod



CANONICAL