

# Getting started with Kubernetes (Workshop)

NATALIE SEREBRYAKOVA, SOFTWARE ENGINEER

# What is Kubernetes?

Kubernetes is an open source system for managing containerized applications across multiple hosts, providing basic mechanisms for deployment, maintenance, and scaling of applications. The open source project is hosted by the Cloud Native Computing Foundation

Kubernetes defines a set of building blocks, which collectively provide mechanisms that deploy, maintain, and scale applications. (Wiki)

# Why to use Kubernetes?

With modern web services, users expect applications to be available 24/7, and developers expect to deploy new versions of those applications several times a day.

Containerization helps package software to serve these goals, enabling applications to be released and updated in an easy and fast way without downtime.

# Workshop

We'll learn about Kubernetes by deploying a simple web application across a Kubernetes cluster.

The goal is to deploy a local development Kubernetes cluster using minikube.

# Setup

- ***Install a virtualization software.***

VirtualBox (KVM2 will work too) Here's the downloads page where you can get started easily.

- ***Install kubectl***

Kubectl is the CLI tool for interacting with the Kubernetes cluster. Jump over to the official page, pick your OS and run the provided commands.

- ***Install Docker***

*Docker will be tasked with creating and managing containers.*

- ***Finally, install Minikube.***

<https://github.com/kubernetes/minikube/releases>

# Kubernetes Core Concepts

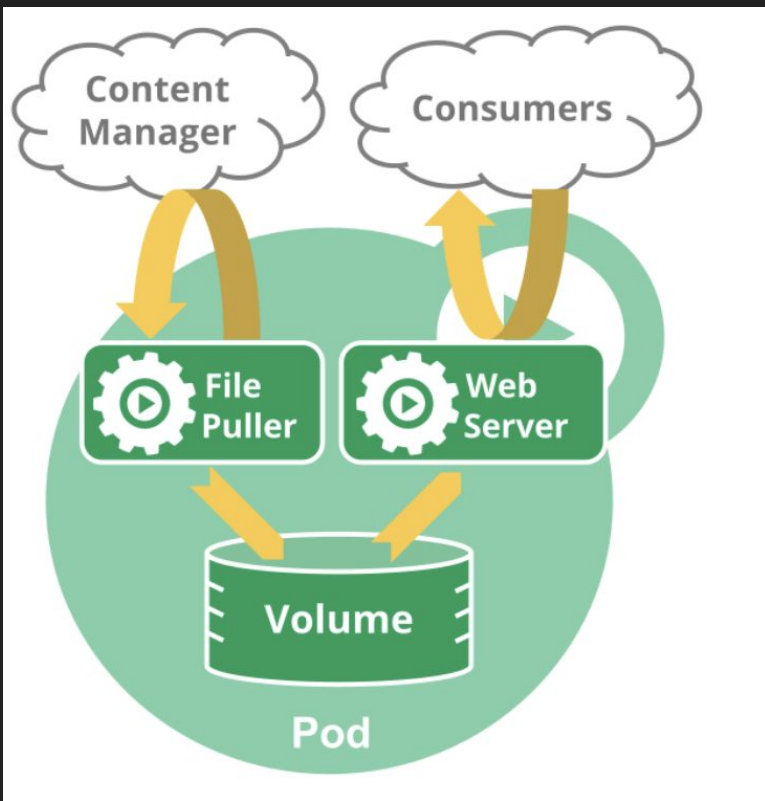
# What is the Scheduler?

- Allows to shift model of infrastructure from instances to a pool of resources.
- “I have an application. Here is a rough estimation of its resource needs. Run it for me somewhere.”
- As Nodes come and go, the Scheduler adjusts where things are running to prevent disruption.
- The Scheduler can be instructed to have an application prefer or avoid to run on the same Nodes as other applications.
- We can also ask it to schedule our application on a specific *type* of Node (ie: “I need fast network or disk”).

Pods



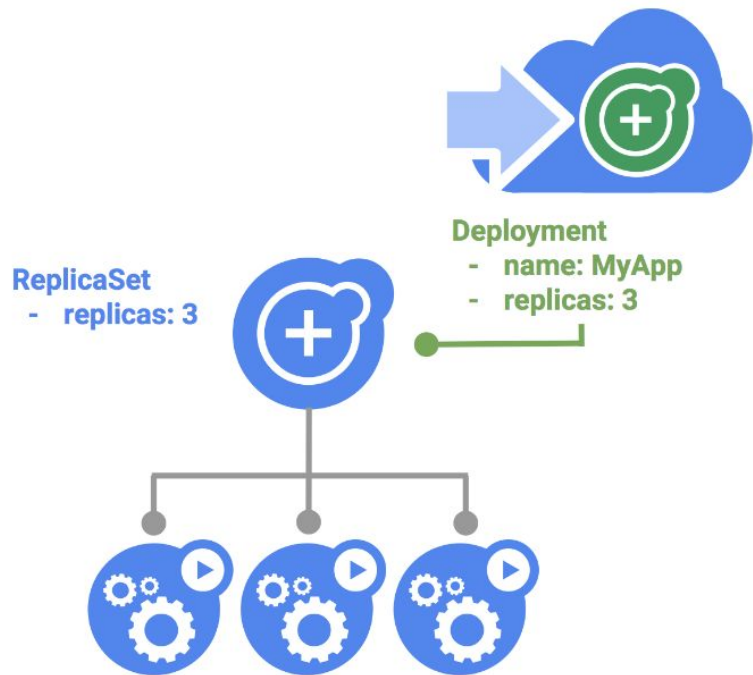
# Pod



- A unit of schedulable work
- Could be a daemon or a job
- Contains at least one container
- Has its own IP address, port range
- Containers in the same Pod:
  - Can share IP, port range, kernel namespace
  - Can reach each other via localhost
  - Can share volumes

# Deployments

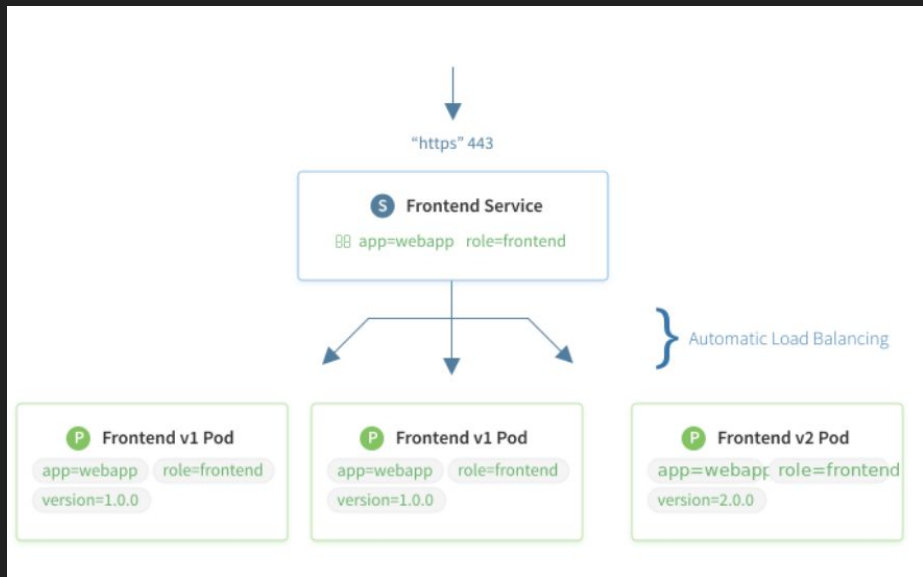
# Deployment



- Used to stamp out an arbitrary quantity of Pods, called replicas
- Can scale an application by editing the Deployment's replicaCount
- If Pods become unhealthy, the Deployment makes sure replacements are stood up
- When releasing new versions of an app, the Deployment supports various rollout methods (rolling, blue/green, etc).

Services

# Service



- Service discovery for Kubernetes
- Instead of addressing Pod IPs (which change frequently) individually, a Service can provide a steady IP + hostname to address a set of Pods
- Routes traffic to Pods who have certain labels (via a selector)

# Recap

- The Scheduler determines where to run Pods
  - We (users) think about resources rather than instances
- Pods are schedulable work (daemons, jobs, etc)
  - Pods have at least one container
  - Each Pod has its own IP and port space
  - Containers in the same Pod can reach one another
  - ...and also share volumes
- Deployments
  - Stamp out multiple Pods and maintain desired replica count
- Services
  - Provides a stable IP+hostname to address a set of Pods

# More resources

- [Kubernetes Object types](#)
- [How-to Guides](#)
- [Tutorial](#)

Let's clone:

```
git clone https://github.com/99/meetup-workshop
```