



FALMOUTH
UNIVERSITY

Kubernetes



FALMOUTH
UNIVERSITY

COMP260: Distributed Systems

9: Kubernetes

Register Attendance



Figure 1: Attendance monitoring is in place. It is your responsibility to ensure that you have signed yourself in.

Kubernetes

"Kubernetes is the operating system of the cloud native world, providing a reliable and scalable platform for running containerised workloads." (Arun-del, Domingus, 2019)



Kubernetes

- ▶ Extremely popular
- ▶ Connects many servers together (orchestration)
- ▶ There are managed versions of Kubernetes (good for most options)
- ▶ Can be quite difficult to get right
- ▶ Avoid vendor lock-in

Architecture - Cluster

"A Kubernetes cluster is a set of node machines for running containerized applications." (RedHat, 2019)

- ▶ Contains a worker node and a master node."
- ▶ Comprise of multiple nodes
- ▶ Nodes are basically just individual machines

Architecture - Master Node

- ▶ Control Plane (the brain of the cluster)
- ▶ Stores info about all **worker nodes**
- ▶ Multiple instances - highly available
- ▶ Manages **deployments**
- ▶ Doesn't **usually** run user workloads
- ▶ failure might cause erratic behavior

Architecture - Worker Nodes

- ▶ manages the container runtime (Docker)
- ▶ User workloads
- ▶ failure is acceptable

Fundamentals - Deployments

“For every program that Kubernetes has to supervise, it creates a corresponding Deployment object, which records some information about the program: the name of the container image, the number of replicas you want to run, and whatever else it needs to know to start the container.”

(Arundel, Domingus, 2019)

- ▶ Supervising and scheduling
- ▶ The **Deployment Controller** maintains the desired spec
- ▶ Restart always by default
- ▶ Restart only on failure

```
kubectl get deployments
```

Fundamentals - Pods

“A **Pod** is the Kubernetes object that represents a group of one or more containers”

(Arundel, Domingus, 2019)

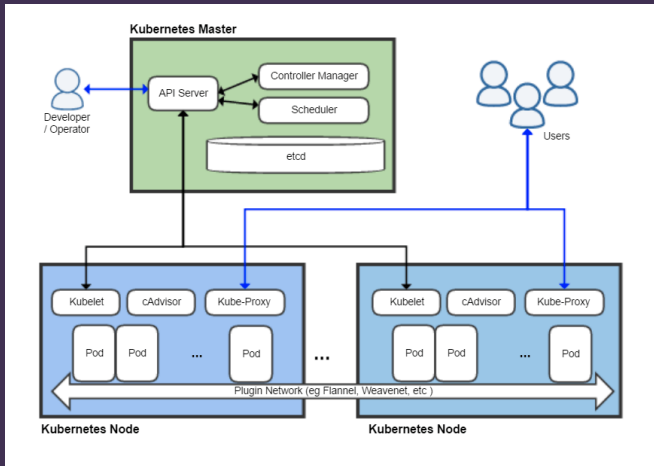
- ▶ Most Pods have one container
- ▶ dependant container should scheduled together and thus live in the same Pod
- ▶ Defined via Pod Specification
- ▶ When a deployment decides a new replica is needed, it creates a Pod resource in the Kubernetes database and the Pod is added to a queue for the Scheduler to handle

Fundamentals - Services

Services solve a fundamental issue by providing a single, unchanging IP address or DNS name to route traffic to the appropriate Pods.

- ▶ Think of a service as a web proxy or load balancer
- ▶ forwards requests to backend Pods
- ▶ Any ports not just web (80, 443)

Cluster Overview



Kubectl

Multipurpose tool for:

- ▶ Applying configurations
- ▶ Creating, deleting and modifying resources
- ▶ Querying the state of the Cluster

examples:

- ▶ `kubectl get nodes`
- ▶ `kubectl describe`

Helm Package Manager



Helm Package Manager

- ▶ Part of the Cloud Native Computing Foundation (CNCF) family
- ▶ Widespread
- ▶ Command line interface (CLI)
- ▶ **Helm Charts** contains resource definitions required to run app
- ▶ **Release** a particular instance of a chart running on a cluster
- ▶ **Helm repository** to host and share charts
- ▶ Charts are often stored in the application repo

Helm Charts

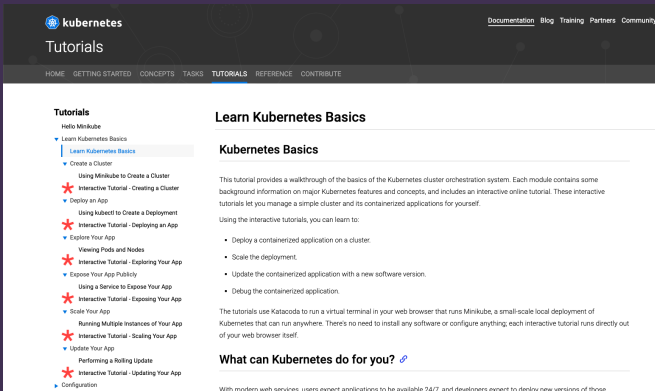
- ▶ Can be installed multiple times in the same cluster
- ▶ Each Helm Chart release has a unique name (-name flag)
- ▶ `helm list` output current releases
- ▶ `helm status [release name]` provides details about status of specific release
- ▶ Streamline the process of installing applications

Helm Chart Structure

```
helm create helm-chart
```

```
wordpress/  
  Chart.yaml          # A YAML file containing information about the chart  
  LICENSE             # OPTIONAL: A plain text file containing the license for the chart  
  README.md          # OPTIONAL: A human-readable README file  
  values.yaml         # The default configuration values for this chart  
  values.schema.json  # OPTIONAL: A JSON Schema for imposing a structure on the values.yaml file  
  charts/            # A directory containing any charts upon which this chart depends.  
  crds/              # Custom Resource Definitions  
  templates/         # A directory of templates that, when combined with values,  
                    # will generate valid Kubernetes manifest files.  
  templates/NOTES.txt # OPTIONAL: A plain text file containing short usage notes
```

Activity: Kubernetes Bootcamp

A screenshot of the Kubernetes Tutorials page. The header shows the Kubernetes logo and navigation links: Documentation, Blog, Training, Partners, Community. Below the header is a navigation bar with links: HOME, GETTING STARTED, CONCEPTS, TASKS, TUTORIALS (highlighted), REFERENCE, and CONTRIBUTE. The main content area is titled 'Tutorials' and lists various interactive tutorials. The 'Learn Kubernetes Basics' tutorial is highlighted. The right sidebar contains the title 'Learn Kubernetes Basics' and a description of the tutorial, followed by a list of topics covered and a link to the tutorial. The bottom of the sidebar mentions that the tutorials use Katacoda to run a virtual terminal in a web browser.

kubernetes

Documentation Blog Training Partners Community

Tutorials

HOME GETTING STARTED CONCEPTS TASKS **TUTORIALS** REFERENCE CONTRIBUTE

Tutorials

Hello Minikube

- Learn Kubernetes Basics
 - Learn Kubernetes Basics**
 - Create a Cluster
 - Using Minikube to Create a Cluster
 - Interactive Tutorial - Creating a Cluster
 - Deploy an App
 - Using Kubectl to Create a Deployment
 - Interactive Tutorial - Deploying an App
 - Explore Your App
 - Viewing Pods and Nodes
 - Interactive Tutorial - Exploring Your App
 - Expose Your App Publicly
 - Using a Service to Expose Your App
 - Interactive Tutorial - Exposing Your App
 - Scale Your App
 - Running Multiple Instances of Your App
 - Interactive Tutorial - Scaling Your App
 - Update Your App
 - Performing a Rolling Update
 - Interactive Tutorial - Updating Your App
- Configuration

Learn Kubernetes Basics

Kubernetes Basics

This tutorial provides a walkthrough of the basics of the Kubernetes cluster orchestration system. Each module contains some background information on major Kubernetes features and concepts, and includes an interactive online tutorial. These interactive tutorials let you manage a simple cluster and its containerized applications for yourself.

Using the interactive tutorials, you can learn to:

- Deploy a containerized application on a cluster.
- Scale the deployment.
- Update the containerized application with a new software version.
- Debug the containerized application.

The tutorials use Katacoda to run a virtual terminal in your web browser that runs Minikube, a small-scale local deployment of Kubernetes that can run anywhere. There's no need to install any software or configure anything; each interactive tutorial runs directly out of your web browser itself.

What can Kubernetes do for you? [🔗](#)

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

* Official Kubernetes Interactive Tutorials