Kubernetes in One Shot

A Detailed Resource Guide by TrainWithShubham

Welcome to the complete resource guide for the "Kubernetes in One Shot" tutorial by TrainWithShubham.

This guide provides in-depth explanations of the concepts, tools, and projects covered in the video.

Whether you're a beginner or an experienced practitioner, this guide will help solidify your knowledge

and serve as a handy reference.

Contents

1. 1. Introduction to Kubernetes
2. 2. Kubernetes Architecture
3. 3. Cluster Setup
4. 4. Workloads in Kubernetes
5. 5. Networking in Kubernetes
6. 6. Storage in Kubernetes
7. 7. Scaling and Scheduling
8. 8. Cluster Administration
9. 9. Advanced Features
10. 10. Projects Included

1. Introduction to Kubernetes

Kubernetes is an open-source container orchestration platform that automates deployment, scaling, and management of containerized applications.

Layman Description:
Imagine Kubernetes as a manager that ensures your applications (containers) are always running smoothly, scaling when necessary, and available for users.

In-depth:
- History: Developed by Google, Kubernetes has become the standard for container orchestration.
- Benefits: Automates complex tasks like scaling, load balancing, and service discovery.
- Monolithic vs Microservices: Kubernetes supports breaking large applications into small independent services (microservices) for better scalability and resilience.

2. Kubernetes Architecture

Layman Description:

Kubernetes is like a well-coordinated team, with the Control Plane as the leader and the Worker Nodes as the team members.

In-depth:

- Control Plane: Comprises the API Server, etcd (stores cluster data), Scheduler (assigns tasks), and Controller Manager (ensures the desired state).
- Worker Nodes: Run the containers using the kubelet, kube-proxy (networking), and container runtime (e.g., Docker).
- Communication: All components communicate through the API Server, the heart of Kubernetes.

3. Cluster Setup

Layman Description:

You need a playground to run Kubernetes. Clusters can be set up on your local machine, in the cloud, or on bare metal servers.

In-depth:

- KIND (Kubernetes IN Docker): Lightweight and perfect for local testing.
- Minikube: A simple tool to set up a Kubernetes cluster on your local machine.
- AWS EC2 and Kubeadm: For more production-like setups, these are preferred.

4. Workloads in Kubernetes

Layman Description:

Workloads are like tasks assigned to Kubernetes, defining what applications or services need to run.

In-depth:

- Pods: The smallest deployable unit in Kubernetes.
- Deployments: Manages rolling updates and scaling of Pods.
- ReplicaSets: Ensures a specified number of replicas are running.
- StatefulSets: Manages stateful applications like databases.
- DaemonSets: Ensures a Pod runs on every node.
- Jobs and CronJobs: One-time and recurring tasks.

5. Networking in Kubernetes

Layman Description:

Networking ensures that different parts of your application can talk to each other and users can access your services.

In-depth:

- Cluster Networking: Allows Pods to communicate across the cluster.
- Services: Provides stable networking for Pods (ClusterIP, NodePort, LoadBalancer).
- Ingress: Manages HTTP/HTTPS traffic to Services.
- Network Policies: Secures communication between Pods.

6. Storage in Kubernetes

Layman Description:

Storage in Kubernetes is like a shared folder that persists data even if the application crashes.

In-depth:

- Persistent Volumes (PV) and Claims (PVC): Decouples storage from Pods.
- StorageClasses: Automates dynamic provisioning of storage.
- ConfigMaps: Manages non-sensitive application configuration.
- Secrets: Stores sensitive data like passwords securely.

7. Scaling and Scheduling

Layman Description:

Kubernetes can automatically adjust resources based on demand and ensures applications run where they are needed most.

In-depth:

- Horizontal Pod Autoscaler (HPA): Scales Pods based on CPU or memory usage.
- Vertical Pod Autoscaler (VPA): Adjusts resource requests and limits.
- Node Affinity and Taints/Tolerations: Controls which Pods run on which nodes.

8. Cluster Administration

Layman Description:

Cluster administration involves managing security, adding custom features, and monitoring the cluster.

In-depth:

- RBAC (Role-Based Access Control): Manages permissions for users and applications.
- CRDs (Custom Resource Definitions): Extends Kubernetes with new object types.
- Monitoring: Tools like Kubernetes Dashboard and Prometheus provide insights.

9. Advanced Features

Layman Description:

Advanced features make Kubernetes even more powerful and extensible.

In-depth:

- Sidecar and Init Containers: Auxiliary tasks for logging or initialization.
- Operators: Automates complex application management.
- Helm: A package manager for Kubernetes applications.
- Istio: Adds service mesh capabilities for traffic management.

10. Projects Included

Layman Description:

Hands-on projects help you practice what you learn.

In-depth:

- Project 1: Chat application using a 3-tier architecture on Minikube.
- Project 2: Multi-language (.Net, Python) 3-tier app on KIND.
- These projects include deployment, networking, and scaling examples.