Kubernetes

A Guide to Kubernetes

@Jonathan Ndambuki

Table of contents

Kubernetes

```
                    ┌──────────────────────────────┐
                    │ Introduction to Kubernetes   │
  ┌────────────┐    └──────────────────────────────┘
  │ Kubernetes │
  └────────────┘    ┌──────────────────────────────────────────┐
                    │ Kubernetes Architecture and Core Component │
                    └──────────────────────────────────────────┘
```
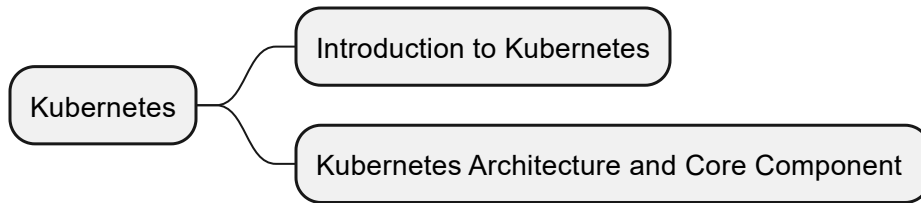
Introduction to Kubernetes

What is the Problem

Manual deployment of containers is hard to maintain , error-prone and annoying. Below are some of the challenges one might face:

- Containers might crash/ go down and need to be replaced

- We might need more container instances upon traffic spikes

- Incoming traffic should be distributed equally

But an ECS seems to Solve the above issue

Yes, and Elastic Container Service(ECS) can solve the above issue but it locks us in, This might not sound as a major problem if you are using AWS but when you want to switch to another cloud provider its an issue because you will have to learn about the specifics, services and config options of the other cloud provider

> ⚠ **Kubernetes to the Rescue!**

What is Kubernetes

Kubernetes, also known as K8s, is an open source system for automating deployment, scaling, and management of containerized applications

It provides a container runtime, container orchestration, container-centric infrastructure orchestration, self-healing mechanisms, service discovery and load balancing. It's used for the deployment, scaling, management, and composition of application containers across clusters of hosts.

Imagine having a bunch of servers, and you want to run your applications on them without worrying about where exactly they run or how they recover if something goes wrong. Kubernetes takes care of that for you, making sure your applications are always running smoothly.
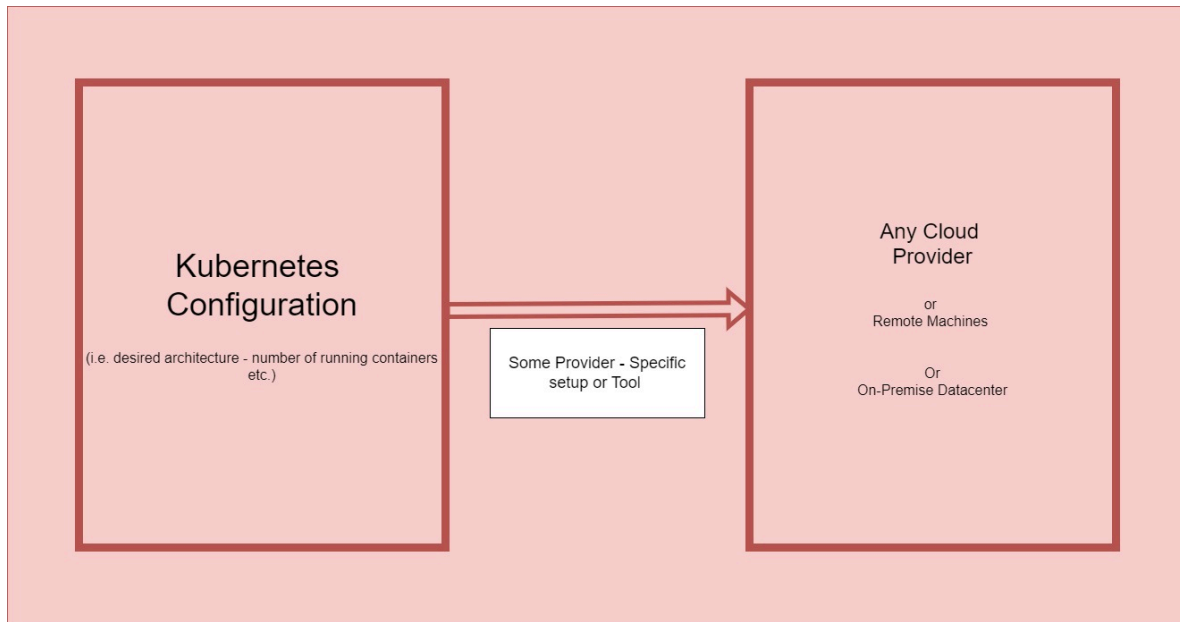
k1.jpg

What Kubernetes is not

⚠️
- It's not a cloud service provider

- Its not a service by a cloud service provider

- It's not restricted to any specific (cloud) Service provider.

- Its not just a software you run on some machine

- its not an alternative to Docker

- its not a paid service
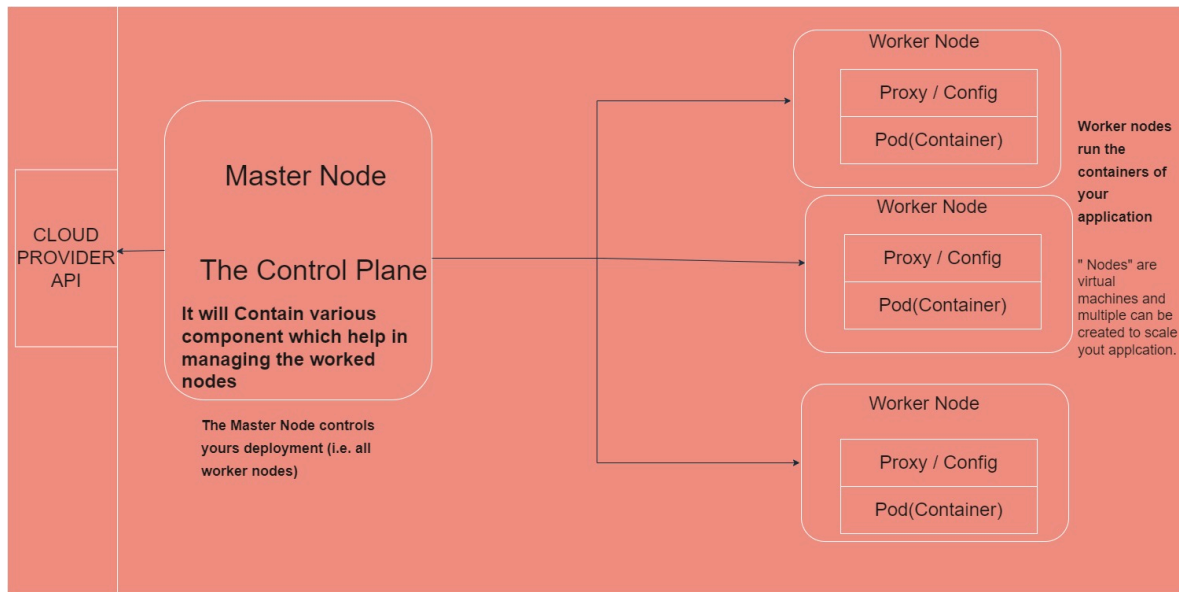
Rather it's:

ℹ️
- It is an open source project

- It can be used with any provider

- Its a collection of concepts and tools

- It works with (docker) containers

- Its is a free open-source project

- its not a paid service

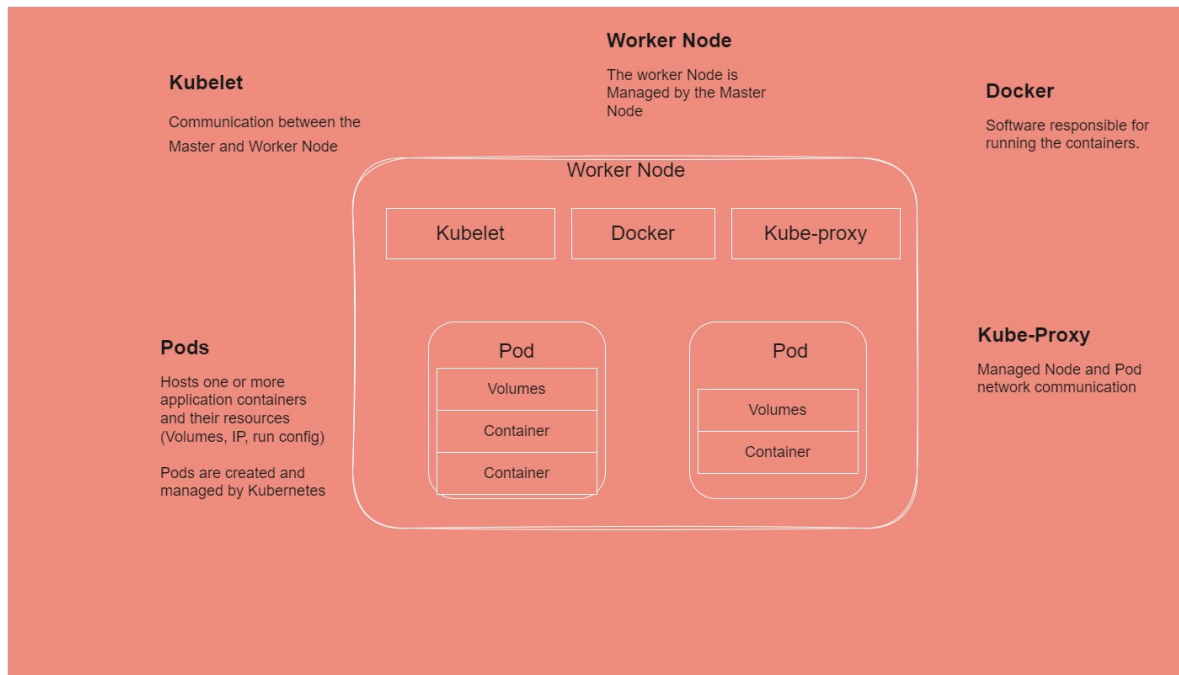⚠️ Kubernetes is like Docker-Compose for Multiple Machines

Core Kubernetes Concepts & Architecture

k2.jpg

The Worker Node

Think of the worker node as one computer/ machine / virtual instance These nodes run the container/containers and are managed by the master Node

k3.jpg

Kubelet:

The Kubelet is an agent that runs on each node. It communicates with the Kubernetes control plane (the master node) to make sure the containers are running as expected. The Kubelet ensures that containers described in a PodSpec are running and healthy.

Container Runtime:

This is the software responsible for running the containers. Common container runtimes include Docker, containerd, and CRI-O. The container runtime pulls the container images from a container registry and runs them as containers on the node. \

Kube-proxy:

Kube-proxy is a network proxy that runs on each node in the Kubernetes cluster. It helps in managing the network rules and communication between services. It ensures that each service can reach any other service in the cluster, handling tasks like load balancing and routing traffic to the correct containers.

Pods:

A Pod is the smallest, most basic deployable object in Kubernetes. Each pod represents a single instance of a running process in your cluster. Pods usually contain one or more containers that are tightly coupled and share resources like storage and networking.

The Master Node

The master node (also known as the control plane) in Kubernetes is responsible for managing the entire cluster. It coordinates all activities within the cluster, like scheduling workloads, maintaining the desired state, and handling scaling and updates. Here's what's inside a Kubernetes master node:
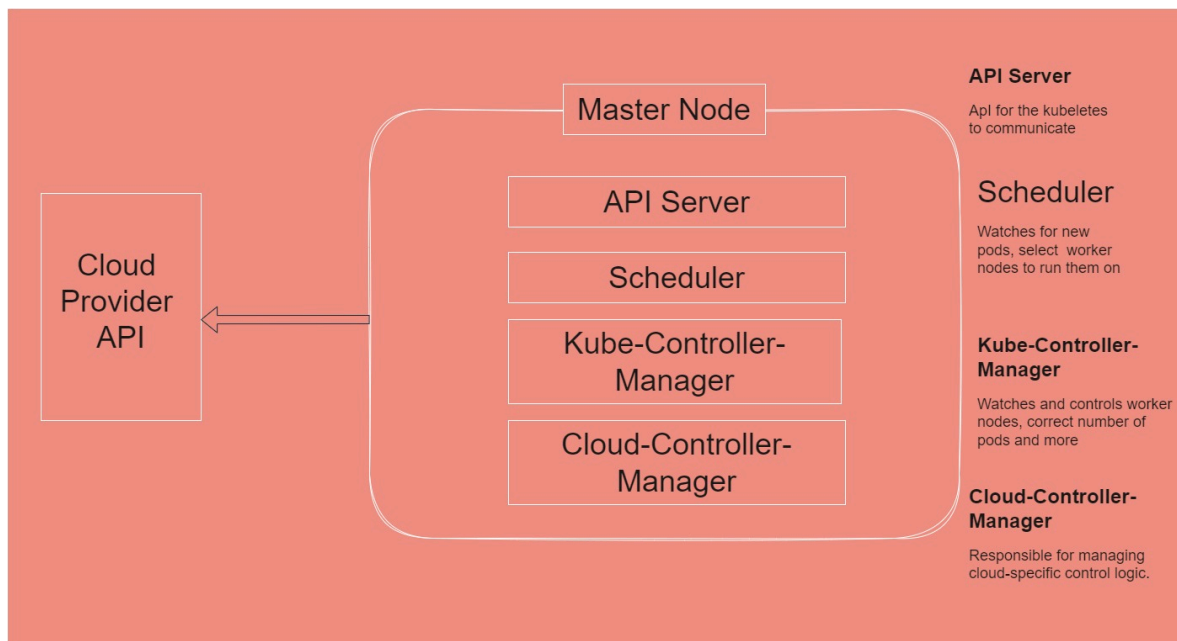
k4.jpg

API Server:

The API Server is the central management entity that exposes the Kubernetes API. It is the front-end for the Kubernetes control plane. All communication with the cluster, including interactions with the nodes, happens through the API Server. It handles RESTful requests and updates the state of the cluster accordingly.

Controller Manager (kube-controller-manager):

The Controller Manager runs controllers, which are background processes responsible for maintaining the desired state of the cluster. Examples include the Node Controller

(which handles node failures), Replication Controller (which ensures that the correct number of pod replicas are running), and the Endpoints Controller (which manages endpoint objects tied to services).

Scheduler (kube-scheduler):

The Scheduler is responsible for placing (or "scheduling") pods onto nodes in the cluster. It looks at the available resources on each node and matches that with the resource requirements of the pods. The scheduler makes decisions to ensure efficient resource utilization and adheres to any constraints defined in the pods' specifications.

Cloud Controller Manager (optional):

The Cloud Controller Manager is responsible for managing cloud-specific control logic. It allows Kubernetes to interact with the underlying cloud provider, handling tasks like node provisioning, load balancers, and storage. This component is more relevant if you're running Kubernetes on a public cloud like AWS, GCP, or Azure.

Kubernetes Installation

We need to install Two Tools:

- Kubectl

- Minikube

KubeCtl

The Kubernetes command-line tool, kubectl, allows you to run commands against Kubernetes clusters. You can use kubectl to deploy applications, inspect and manage cluster resources, and view logs.

> ### Instaling KubeCtl
>
> 1. Download Chocolatey
>
>   Chocolatey (https://chocolatey.org/install)
>
> 2. Follow the steps in the link below to install Kubectl
>
>   Kubectl (https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/#install-nonstandard-package-tools)

Installing Minikube

> ### Installing Minikube
>
> - Installing Minikube
>
>   Minikube (https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download)
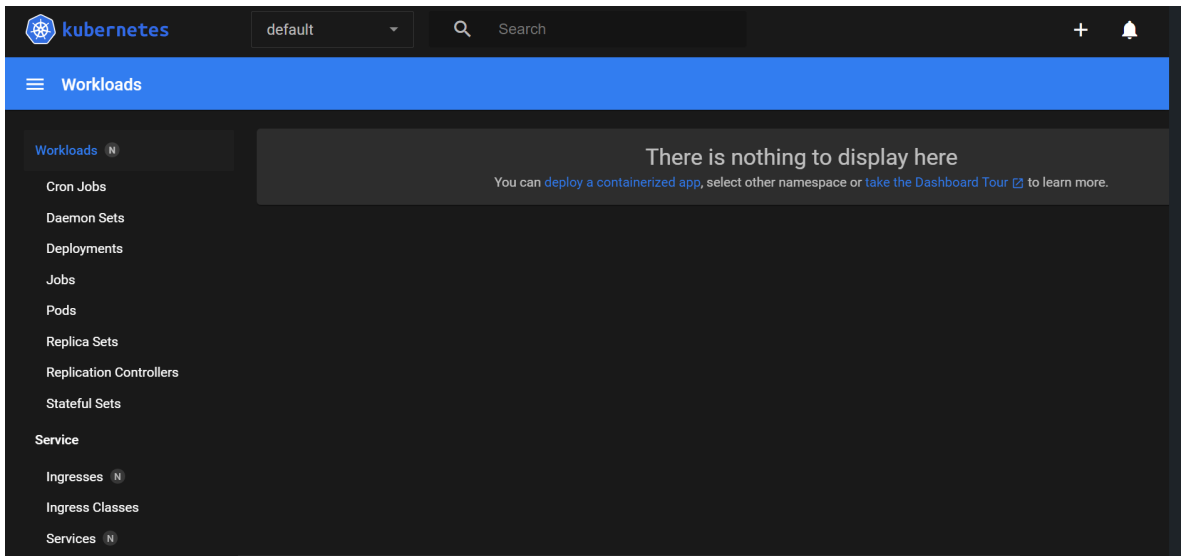
Now Test:

minikube is local Kubernetes, focusing on making it easy to learn and develop for Kubernetes. All you need is Docker (or similarly compatible) container or a Virtual

Machine environment

Run this On the Terminal:

minikube dashboard

The Output:

k3.PNG