

Anatomy

The Anatomy of a Kubernetes Cluster

A look inside the cluster

<https://medium.com/better-programming/anatomy-of-kubernetes-cluster-24d88f77cf27>



Vivek Sonar

Follow

Mar 10 · 4 min read



A typical Kubernetes cluster will contain:

One or more master nodes.

One or more nodes.

Again, these are just a common variety of computers.

Sometimes they may be large physical servers, and sometimes they can even be Raspberry Pis. But in the case of Google's Kubernetes engine and the Elastic Kubernetes service, they are virtual machines.

Master Node

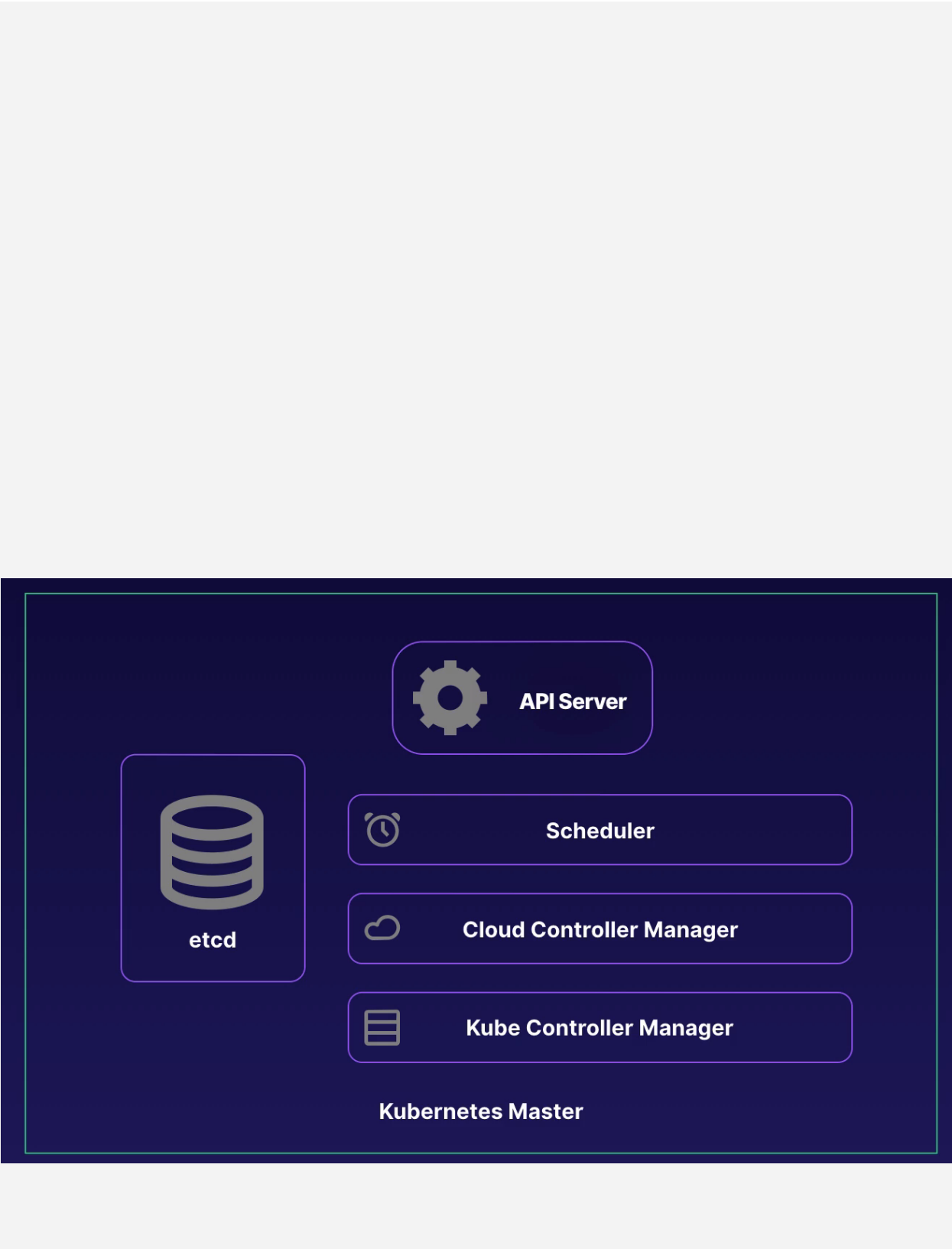
Masters run several components that provide us with something called a *control plane*. They make decisions about the cluster, such as where to schedule specific workloads, etc.

The Master is responsible for the state of the cluster. It continually keeps an eye on everything to make sure that everything is working fine.

Node runs the component that provides the runtime environment (they are basically the workers with the container runtime). These nodes provide the resources for the cluster such as the CPU, RAM, etc.

When you deploy a container on Kubernetes, the Master will pick a node to run it on.

Here's a diagram of a Kubernetes Master node.



API server

The API server is the front end of the control plane; it exposes the API for all the master functions.

Every time you communicate with the Master or something else interacts with the Master (Cloud shell in GCP), it will be through this API server.

Etcd

Etcd is Kubernetes' own database. It stores all Kubernetes' configurations and states. I call it a database, but Etcd is just a key-value store.

Scheduler

The scheduler is responsible for scheduling workloads. What that means is when you want to deploy a container, the Scheduler will pick a node to run that container on.

The node it selects can be affected by a lot of factors such as current load on each available node, requirements of your container, and some other customizable constraints.

Cloud Controller Manager

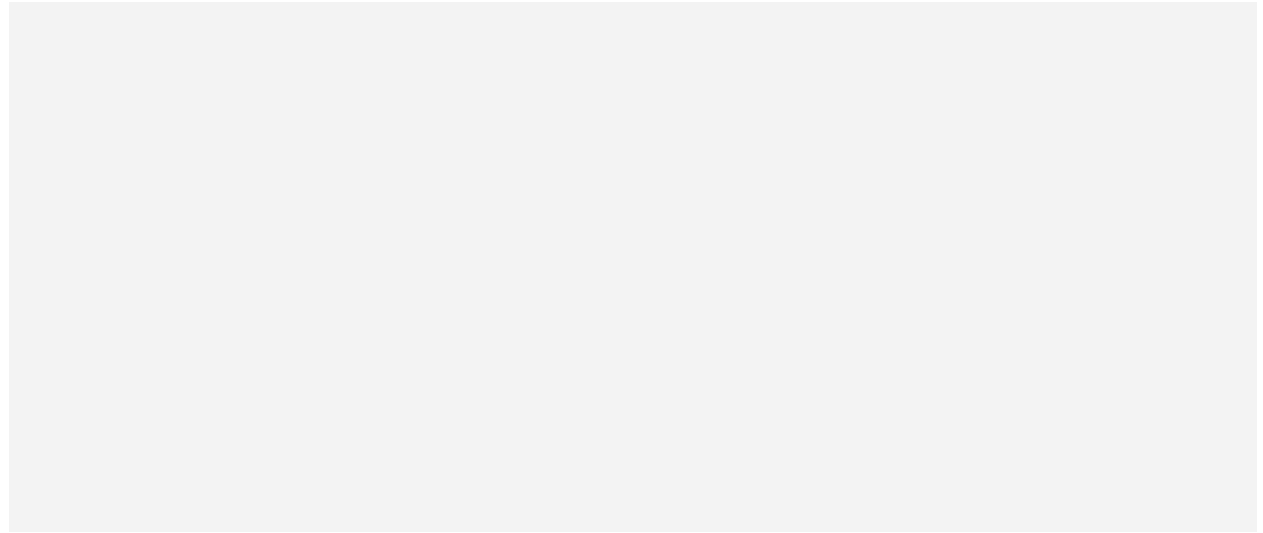
Cloud Controller Manager allows Kubernetes to work with cloud platforms. Remember, Kubernetes is itself an open-source project with contributions from large companies and not just Google, so it doesn't natively contain functionality for GCP only.

This manager is responsible for managing things like networking and load balancing as it translates to the product and services of a particular cloud platform.

Kube Control Manager

The Kube Control Manager's job is to manage a handful of controllers in the cluster. The controllers themselves look after things like nodes and a few other objects.

Now, this might be a lot of information to digest, but when you start the cluster, you will be working mostly with the API server.



Worker Node

Now let's look at the node (Worker) of Kubernetes. They are a lot more straightforward than the Master.

Here's a diagram of the Worker node.



kubelet



kube-proxy



Container Runtime

Kubernetes Node

Kubelet

Kubelet is an agent of Kubernetes. It communicates with the control plane and takes instructions like deploying containers when it's told to.

Kube-proxy

The Kube-proxy is responsible for network connections in and out of the node.

Container runtime

The node will run Docker as a container runtime to allow it to run containers.

Creating a Kubernetes Cluster

There are two ways to create a Kubernetes cluster.

The hard way.

The easy way.

The hard way

In the hard way, you spin-up the virtual machines and configure the Worker nodes and Master node.

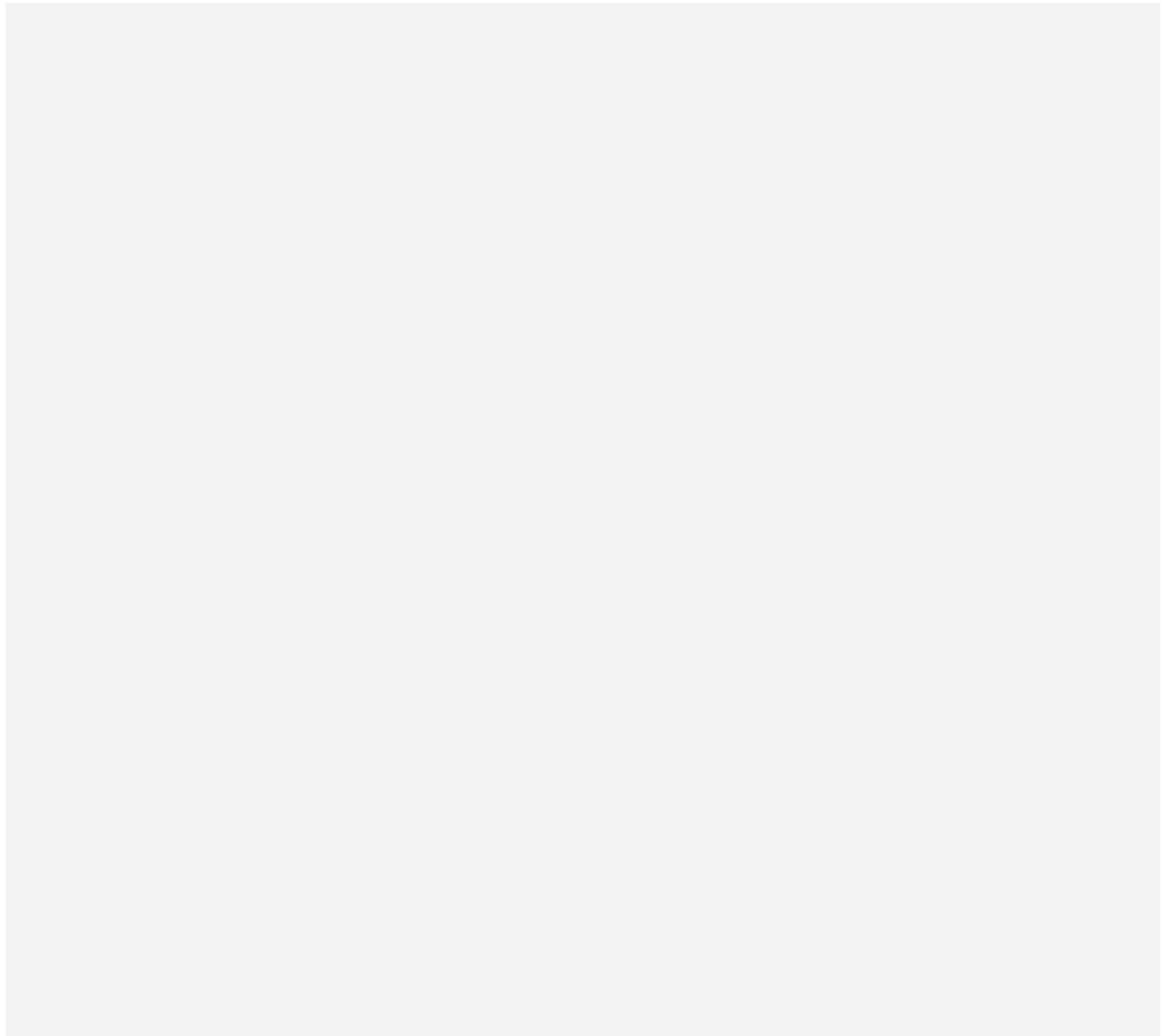
We need to install the Kubernetes software and provision network overlay and set up the certificates so that the components inside can talk with each other.

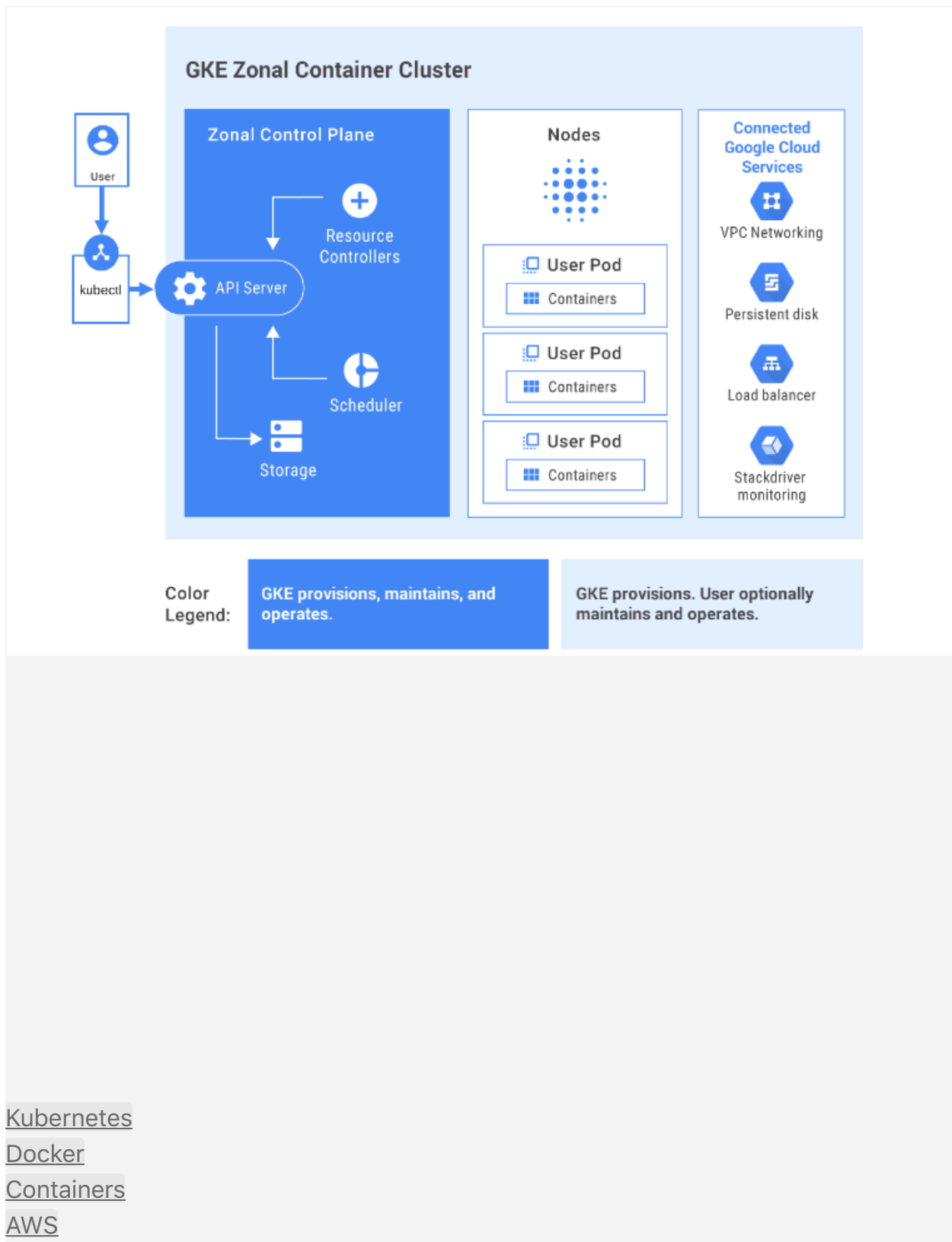
The easy way

For the easy way, we can use the predefined services provided by almost all cloud providers, for example, GKE by Google Cloud Platform.

GKE provisions and manages all the underlying cloud resources automatically. It creates Master nodes and Worker nodes for you, and you don't even touch the Master control plane. It's completely managed for you.

You can compare the architecture of the GKE to our architecture below.





Programming

116 claps



WRITTEN BY

Vivek Sonar

