**Content**

[1. Introduction into the world of Kubernetes 2](#_Toc80527125)

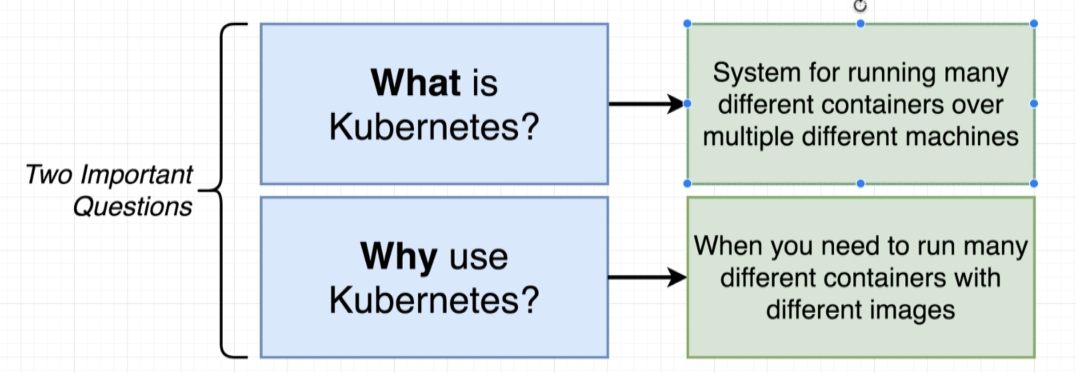
[1. Development / Production Kubernetes 4](#_Toc80527126)

[1.1.1. Docker compose vs Kubernetes 5](#_Toc80527127)

[1.1.2. Configuration files 6](#_Toc80527128)

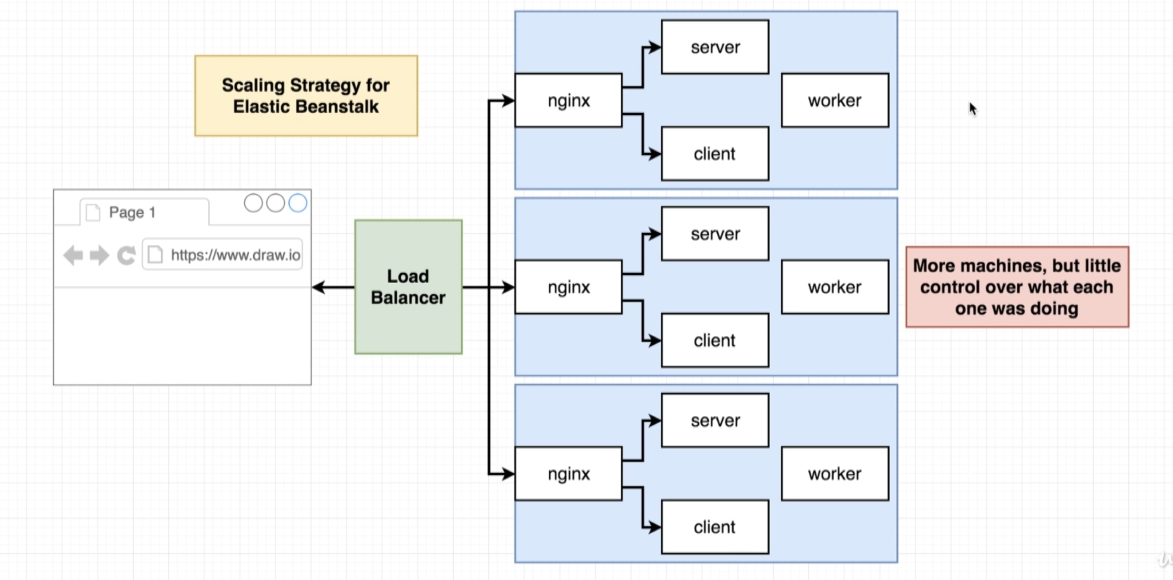
[2. What an object is in Kubernetes 7](#_Toc80527129)

1. Introduction into the world of Kubernetes

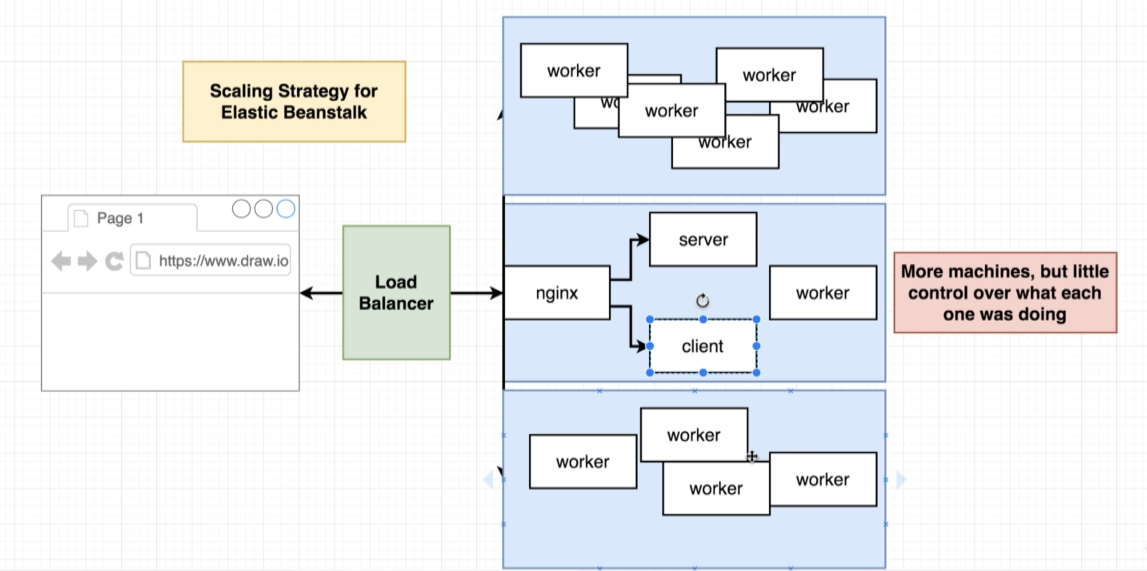


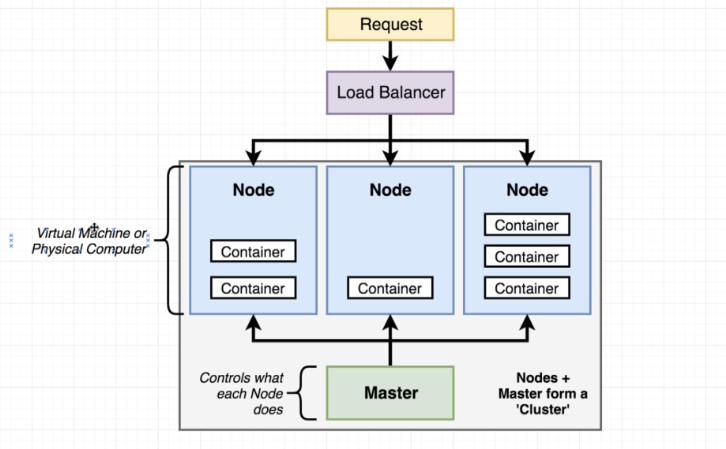
How do we want to scale our multi-container app?

Like this? Do we need so much nginx instances? The worker app does all of the hard work.



I’d like to have only more workers. In the world of Kubernetes we have much more easily gotten to a scaling flow like this:





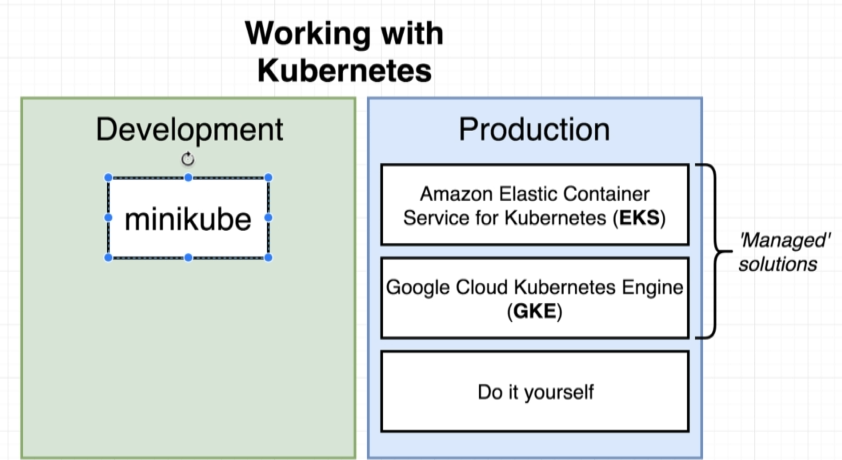
**Cluster.** Kubernetes cluster. A **cluster** in the world on Kubernetes is an assembly of a **Master and 1 or more nodes.**

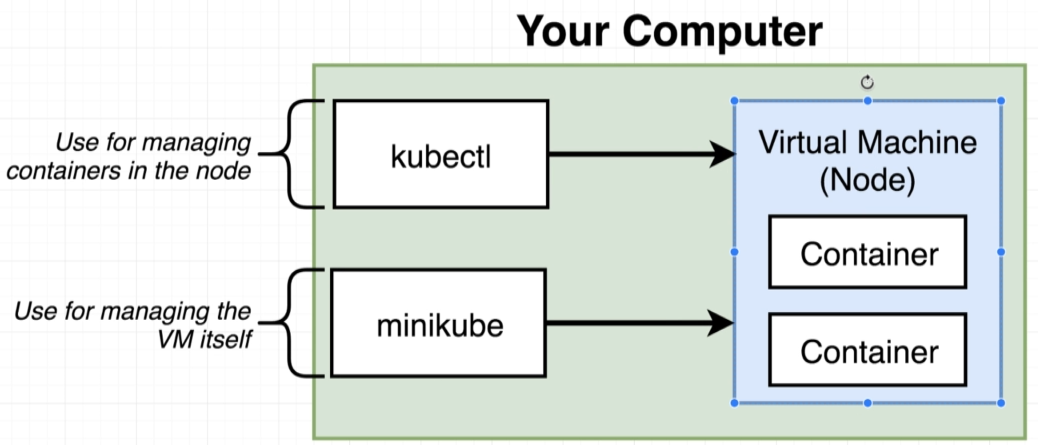
**Node. A node** is a virtual machine or a physical computer, that is going to run some number of different containers. Each of the machine (blue) can be used to run different sets of containers. Each of them can run different containers, (images) in different numbers.

**Master**. All of the Nodes are managed by something called **Master**. This master has a set of different programs running on it and control what each of these different nodes is running at any given time. Developers interact with the cluster though the master. We give some set of instructions to the master e.g. please run 5 containers using the client worker image. The master receives that command and relays it to all of the different nodes.

Outside of our cluster there is a Load Balancer, which will take some amount of outside traffic in the form of network requests and relay all those requests into one of our different nodes.

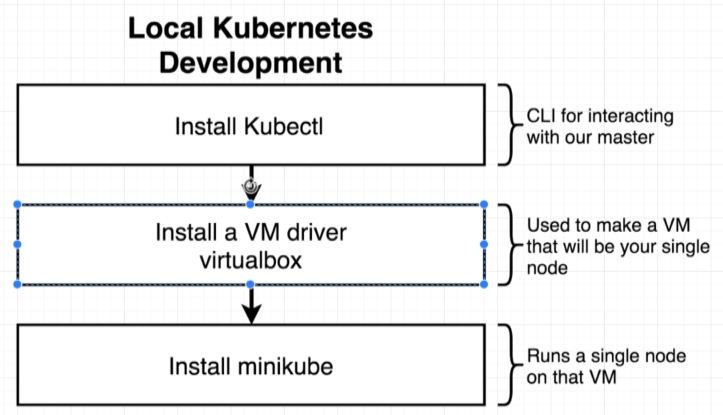
* 1. Development / Production Kubernetes



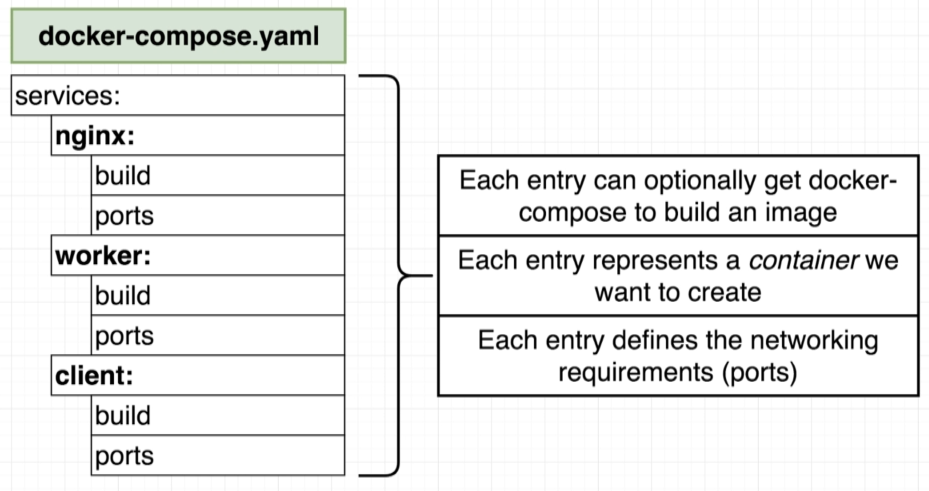


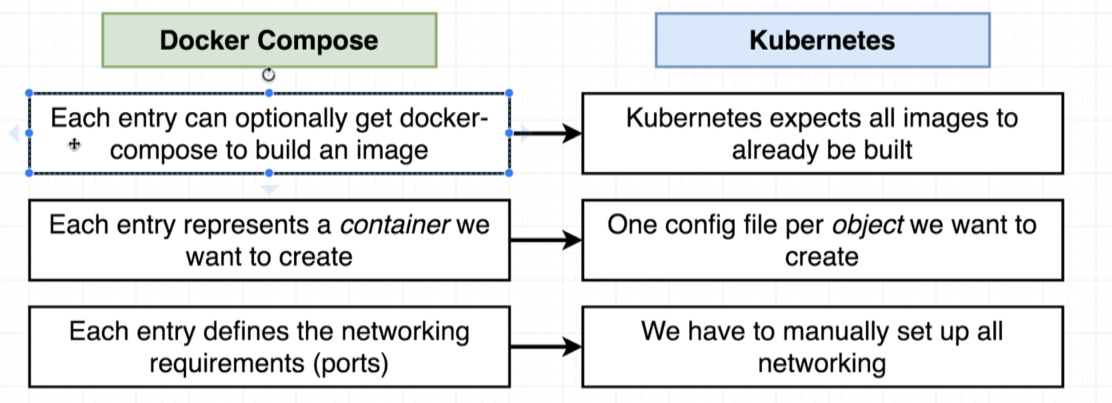
**Minukube** is a command line tool, whose purpose is going to be to set up a tiny little Kubernetes custer on your local computer. Local only. Kubectl we use both locally bot in the production.

When we are talking about production we often use “Managed solutions” – take care a lot of low level task and security.



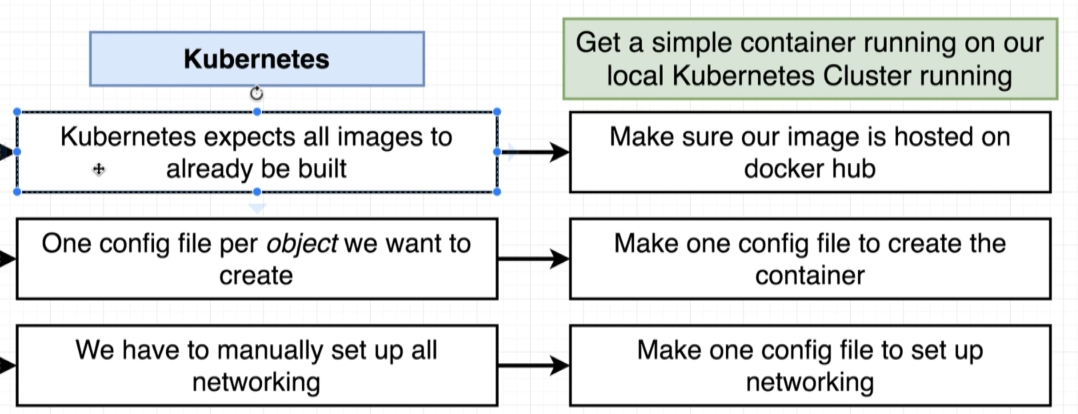
* + 1. Docker compose vs Kubernetes

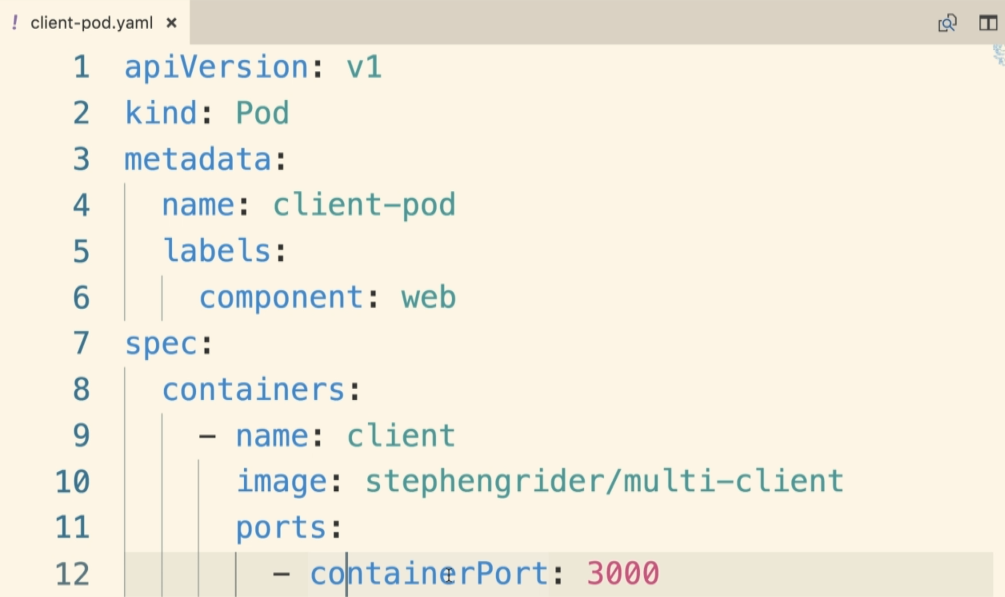


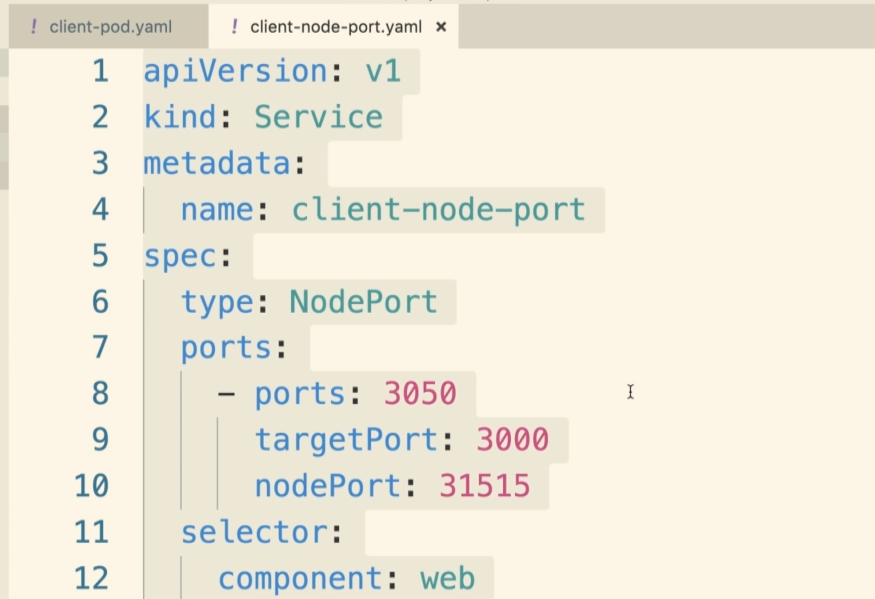


Object:

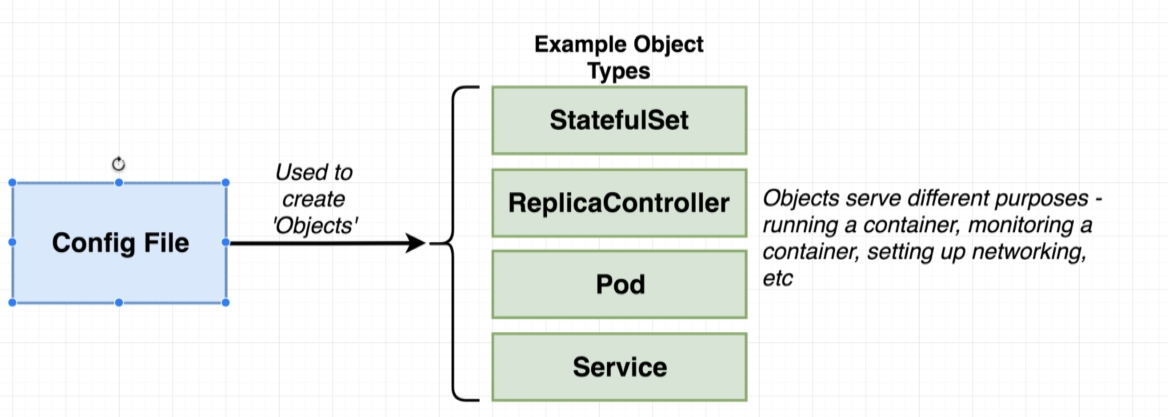
* + 1. Configuration files

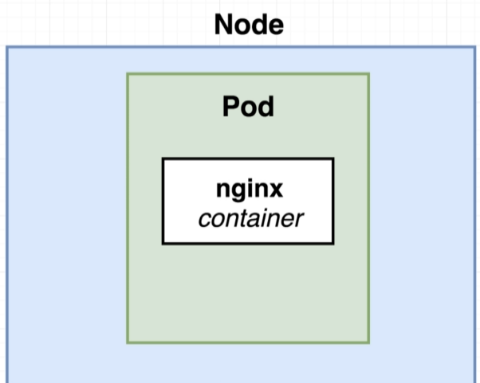




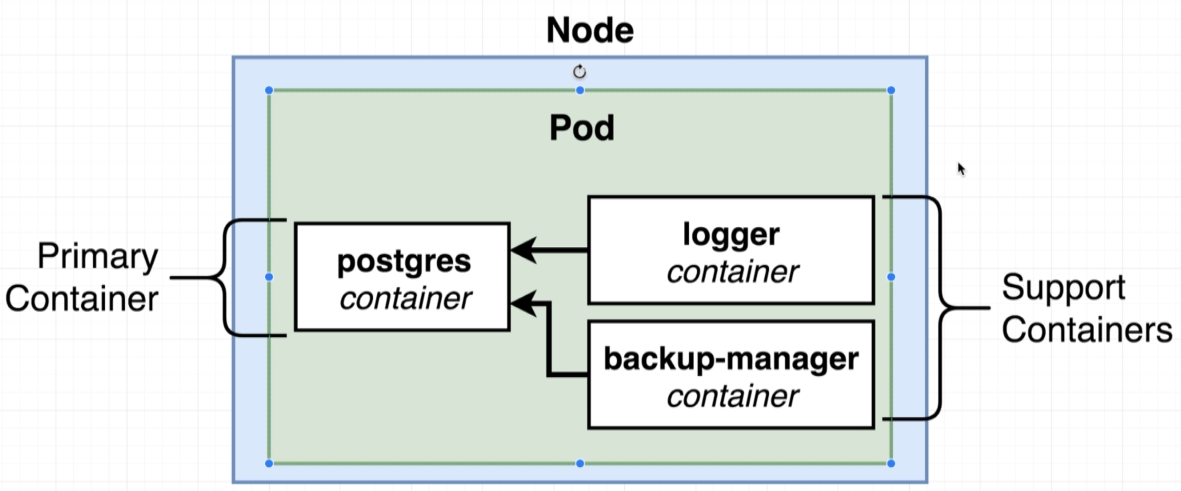


* 1. What an object is in Kubernetes

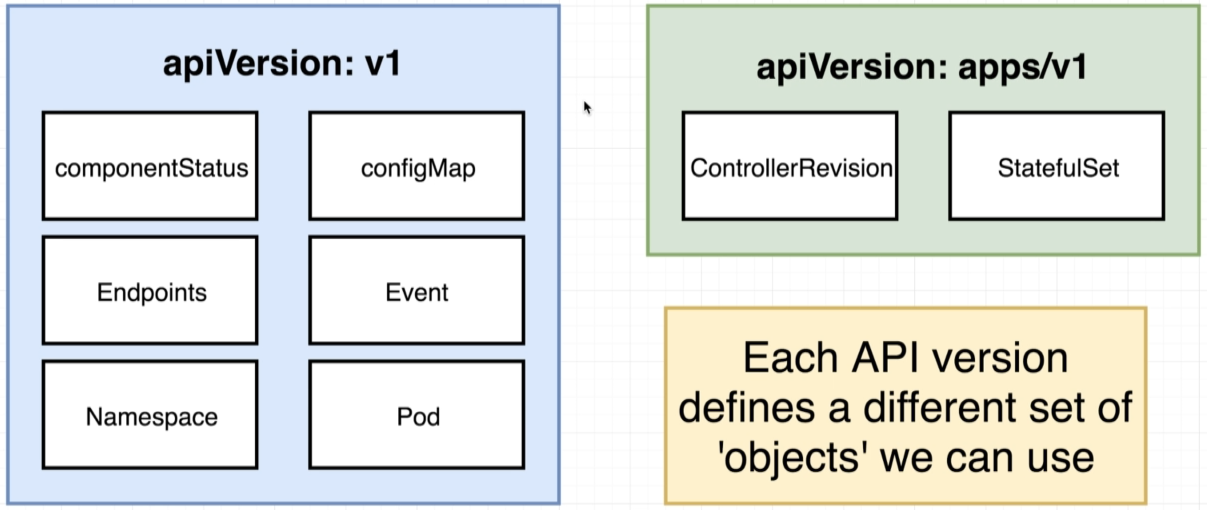


Pod – run a container. Node is the virtual machine created by the minikube. The pod itself is essentially a grouping of containers with a very common purpose. In the world of k8 we do not have the ability to just run one naked single container by itself with no associated overhead. The smallest thing that can be deployed is a pod. Pod needs to have at least 1 container. Pod is some kind of grouping the containers. In a pod we group together containers that have a very tightly coupled relationship – must be executed with each other.

e.g. the logger and the backup manager have nothing to do without the postgres container.



Service – set up networking



1. K3s

K3s is a simple, lightweight Kubernetes distribution, desined for situations where comute resources may be limited.

* Edge
* IoT
* Continuous Integration
* ARM or Embedded
* Development
* Anytime you need a simple k8s

