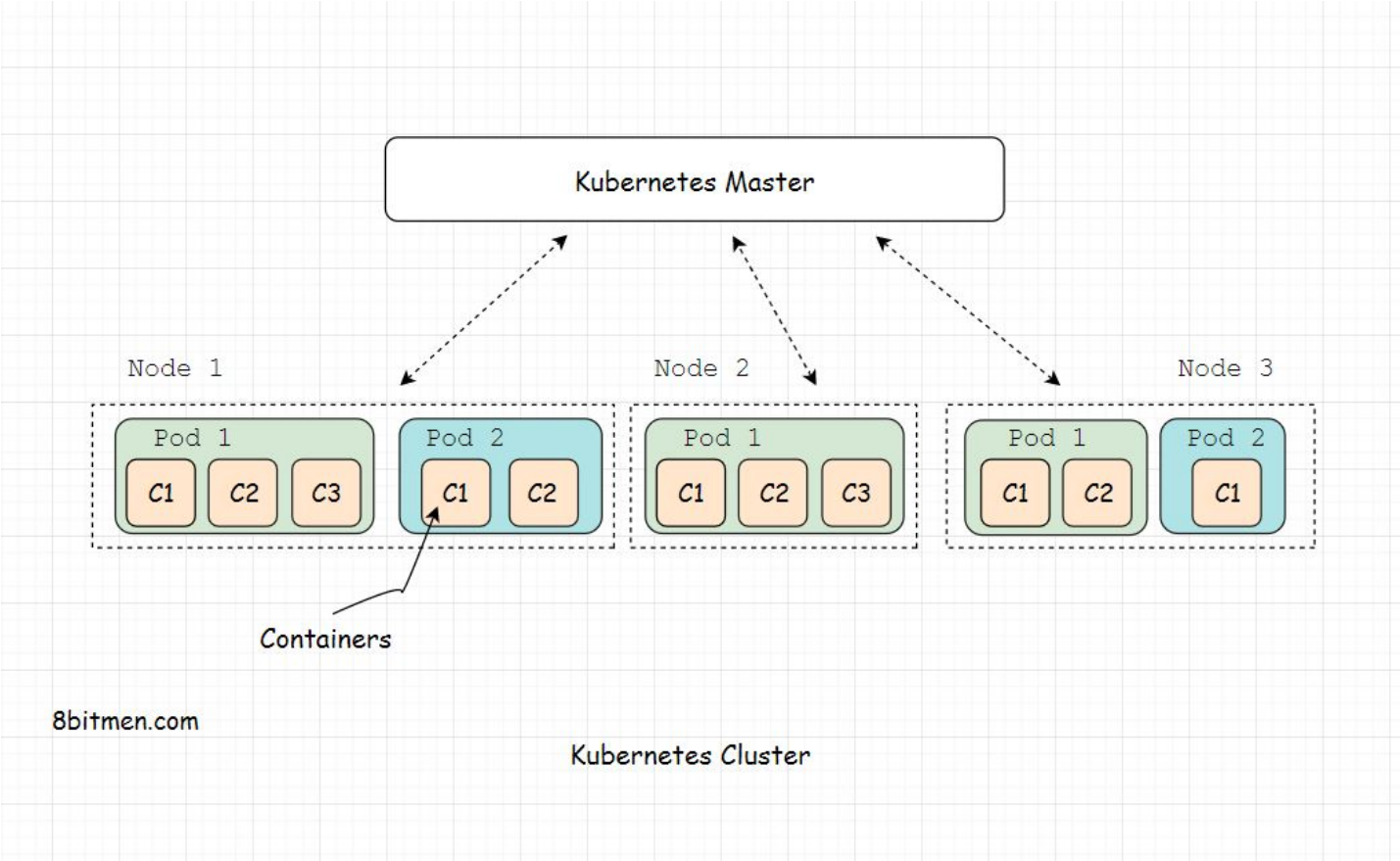
Container Orchestration – Part 2

This lesson continues the discussion about container orchestration.

We'll cover the following

- What are pods?
- Returning to our social network application example



What are pods?

Pods are the smallest deployable units in *Kubernetes* architecture. They are a collection of containers. A pod may contain one or more containers.

A pod running multiple containers is managed as a single entity by Kubernetes. The pod's resources are shared by the containers.

There are primarily two ways containers are run in a pod:

1. Running a single container in a pod

2. Running multiple containers in a pod

When a single container runs in a pod, the pod just acts as a wrapper to the container. Multiple containers can be deployed in a pod to put together related services that can be managed as a single unit. Kubernetes later creates replicas of these pods for *high availability* and *fault tolerance*.

This was a quick overview of Kubernetes and it's architecture. Now, let's get back to our social network application example.

Returning to our social network application example

We had multiple microservices written using different technology stacks in our social media app. Next, we containerized those services and deployed them as containers, pushing all the container images to our private container registry.

Now, we can use a container orchestration tool to manage all the container instances that we have running for all the services. This is the full container deployment workflow, from writing code and deploying it in containers to managing the containers with a container orchestration tool.

> ***Recommended read***: Lessons learned from three container-management systems over a decade at *Google*

This pretty much sums up container orchestration. In the next lesson, we'll discuss *cloud-native* a d *infrastructure as code (IaC)*.