Container Orchestration – Part 1

This lesson introduces container orchestration.

We'll cover the following

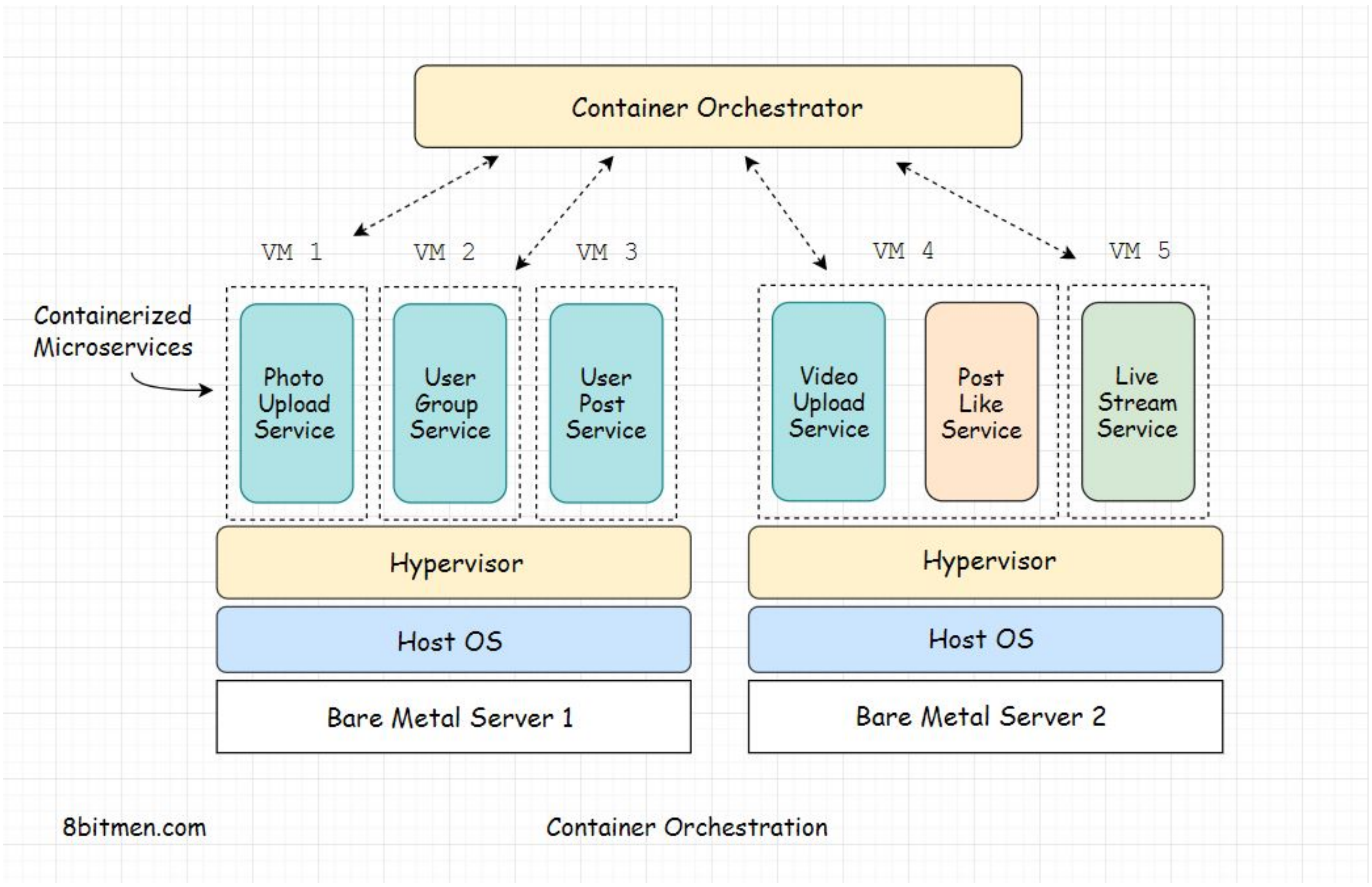
Orchestration overview

Orchestration in computing means automated management, coordination, and configuration of software. When running our software, we generally need to leverage orchestration when things get too complex or too big in number to be handled manually.

When we apply orchestration to containers, it's known as *Container orchestration*.

Internet services at the scale of *Uber, Facebook*, and. so on, run thousands of microservices and hundreds of thousands of container instances of those microservices in different regions across the world. Managing such a large number of processes is not trivial and requires serious effort.

This makes container orchestration key in the deployment of microservices in today's application development landscape. There are several tools available that facilitate this process, including *Kubernetes, Docker Swarm, Mesos*, and so on. *Kubernetes* is the most popular container orchestration tool used in the industry.



Let's take a look at some of the features of container orchestration tools.

Features of container orchestration tools

Container orchestration tools help with:

- Automatically scaling the containers in different *Regions*, *AZs*, and *data centers* across the world
- Load balancing and logically routing traffic to the containers running in different zones
- Allocating resources like compute and memory to the containers and monitoring the health of the clusters running the containers
- Ensuring container availability based on resource consumption to provision additional container instances in the cluster
- Portability of services running in containers, across data centers, and int multi-cloud environments

These are some of the important tasks that container orchestration tools perform. Now, let's take a quick look into *Kubernetes* – the most widely used container

Now, let's take a quick look into *Kubernetes* – the most widely used container orchestration tool in the industry.

Kubernetes

Kubernetes also known as *K8s* was originally developed and internally used by *Google* to successfully run its infrastructure. Later, *Google* released the project as open-source in the year *2014*.

Kubernetes is also offered as a managed service by *Google Cloud* as *Google Kubernetes Engine*.

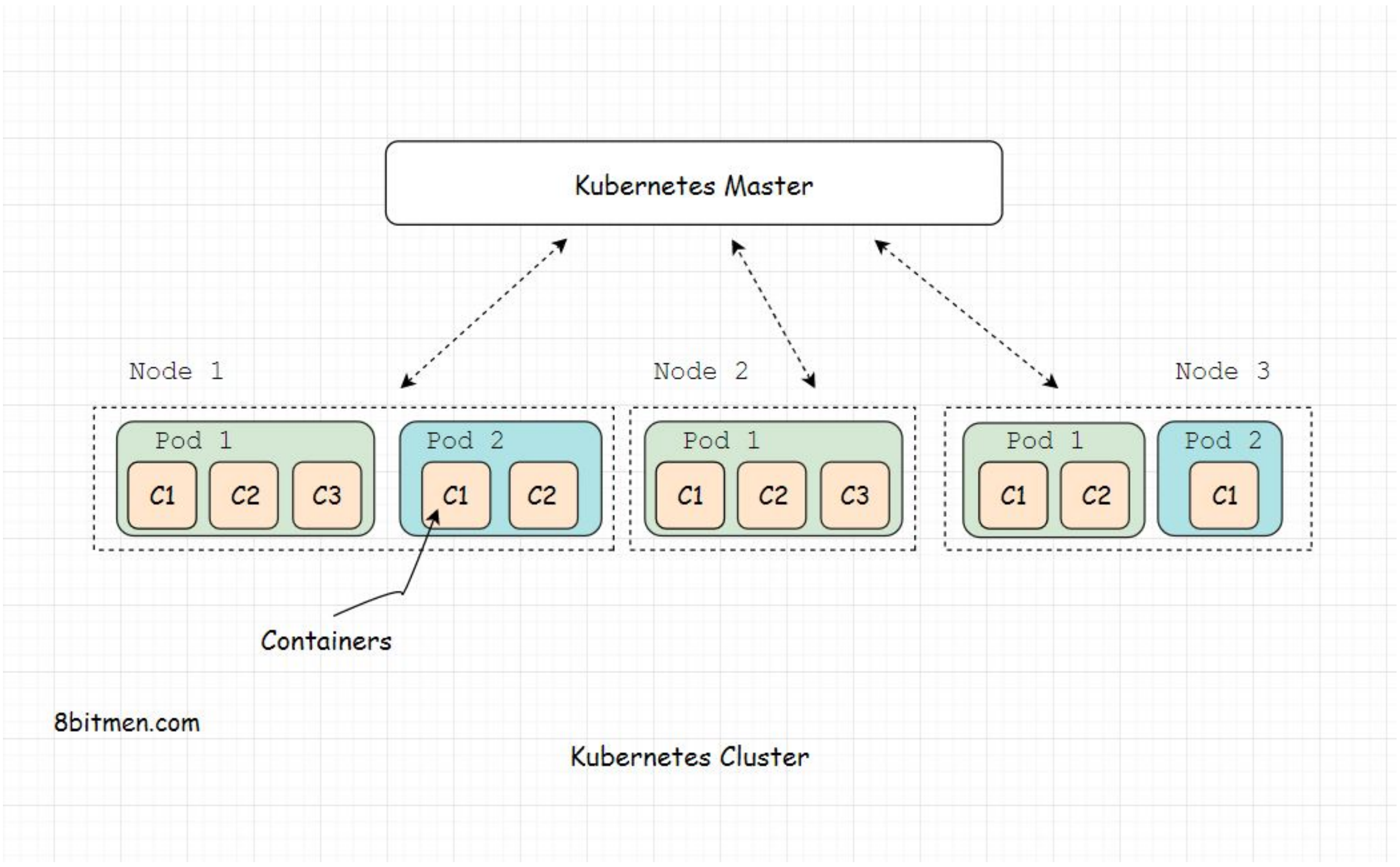
This orchestration tool provides automated container orchestration; it has all the features that container orchestration tools typically offer. It keeps a check on hundreds of thousands of containers typically run by a large-scale service across different geographic regions.

Here is a list of businesses using Kubernetes.

The tool monitors the health and running state of the containers to ensure service uptime. It takes care to balance the traffic load across the containers and schedules the start of new containers and the termination of the existing ones to facilitate auto-scaling in the cluster.

All the processes in the clusters are continually logged, enabling the reliability team to debug the system in case of any errors.

Kubernetes also provides unique ids to containers so they can be easily located when their number gets too high-- something along the lines of how ZooKeeper manages a cluster.



Above is a very basic, high-level diagram of the *Kubernetes* cluster deployment. The key elements of Kubernetes architecture are *containers, nodes* & *pods*.

A typical Kubernetes cluster contains *VMs* or *bare-metal* instances called *nodes* that run the containerized applications in *pods*. All the nodes in the cluster are managed by a master node. The master node acts as a leader in the cluster.

Let’s continue this discussion in the next lesson.