# Cloud Infrastructure Stack

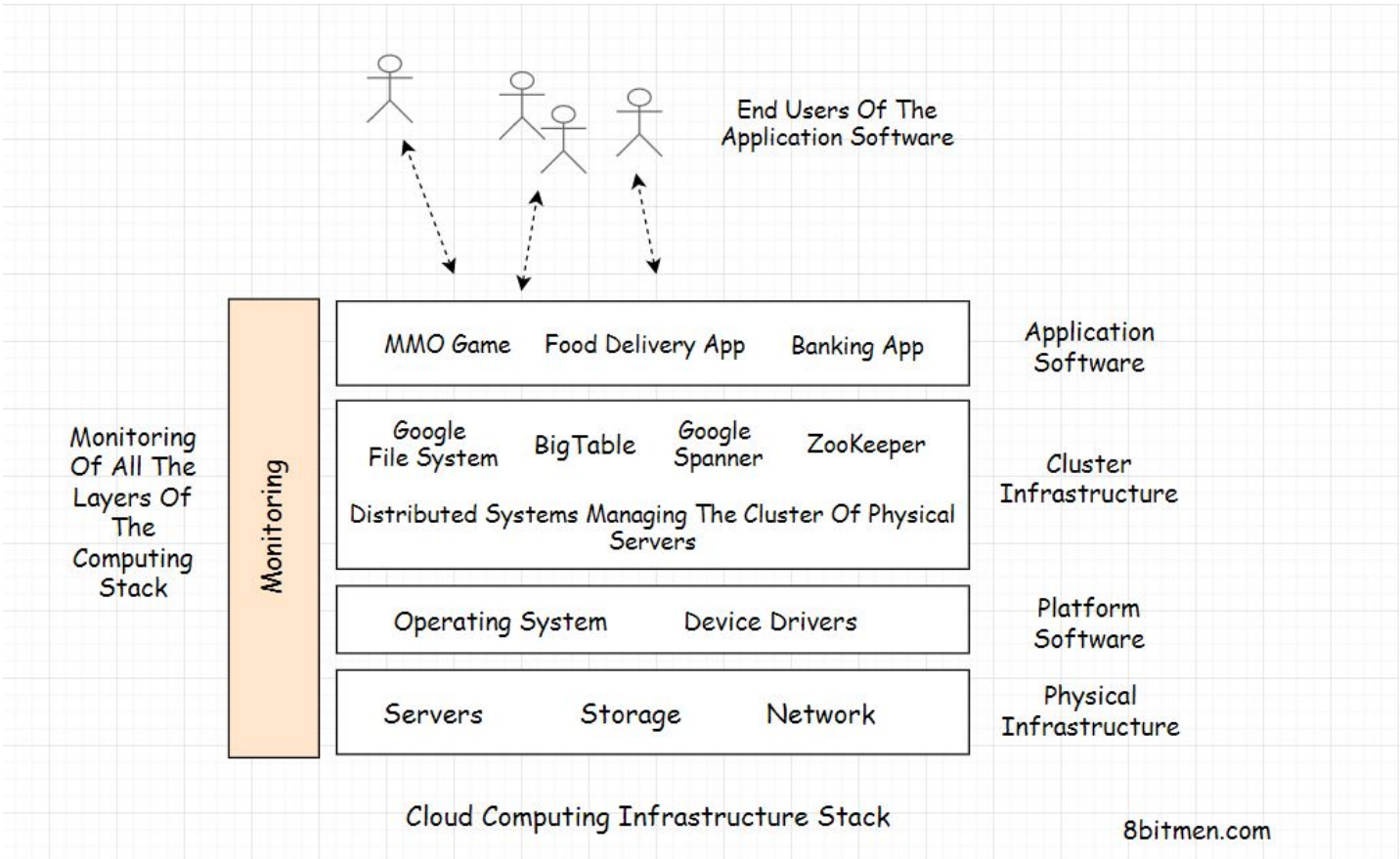
This lesson provides insight into the cloud computing infrastructure stack.

**We'll cover the following**

- Physical infrastructure
- Platform software
- Cluster infrastructure
- Application software
- System monitoring

Before I get down to discussing *bare metal servers, virtualization, virtual machines, containers*, and the related technologies, it's important to have a fundamental understanding of the different layers involved in the cloud computing infrastructure stack.

We'll begin this lesson by having a look at the cloud infrastructure diagram. Then, we'll discuss each layer of the stack step by step.



## Physical infrastructure #

Starting at the bottom, the base layer of the cloud computing stack consists of the physical infrastructure. That includes the computer servers mounted in a server rack, physical storage like the *hard drives, Flash SSDs, and the RAM* attached to the compute servers, and the data center network that connects all the physical infrastructure.

## Platform software #

Above the physical infrastructure layer lies the platform software layer composed of device drivers, firmware, an operating system, and so on. Platform software manages the physical hardware and provides an abstraction layer for the distributed system's software to interact with the hardware.

## Cluster infrastructure #

The cluster infrastructure layer runs on the platform software layer. This consists of distributed systems that manage the hardware resources and are responsible for cluster management.

They facilitate operations, such as message passing between the nodes running in a cluster, distributed node coordination, cluster synchronization, and reaching a consensus, at the cluster level. Some of the examples of distributed-system software are *Apache Zookeeper, and Google File System*. Distributed-storage systems include *Google Bigtable, Hadoop, and Google Spanner*.

Just like an *OS* manages a physical server, cluster level software manages groups of server nodes. We've already discussed this in detail in the clustering chapter.

The cluster infrastructure layer provides an abstraction to the complexity of distributed systems via *APIs*. The application programmer does not really have to worry about how data is dynamically partitioned across several nodes in the cluster, how complex memory operations are executed, how the system ensures fault tolerance, and so on.

Cluster-level software also takes care of cluster resource management, like distributing tasks to the less occupied server nodes to keep a balance among the different nodes in the cluster.

different nodes in the cluster.

*Kubernetes* is one good example of a cluster resource management software that tracks the resource consumption across the container cluster. More on *Kubernetes* and *containers* in the upcoming lessons.

## Application software #

The application software layer contains the applications that we build using the APIs of the cluster level software. These applications are our business applications like social networks, a Fintech app, a food delivery app, an online multiplayer game, and so on.

Distributed systems enable us to run and scale our applications across thousands of nodes running in several different clusters. Running a service in a distributed environment ensures *scalability* and *high availability*.

## System monitoring #

To ensure good system health and continual uptime, we need to continually monitor all the layers of the computing stack. As you see in the diagram, the system monitoring layer is a vertical layer that applies to all the layers of the cloud computing stack.

Monitoring the system, including keeping track of the server uptime, performance, latency, and throughput and resource consumption helps in weeding out bugs, getting rid of system bottlenecks, optimizing resource consumption, setting up alerts, writing efficient code, and more. All these things are vital in building highly-performant software.

All the stats are ideally monitored with the help of a web-based dashboard; *Grafana*, an open-source monitoring tool, is one good example.

*Google Stackdriver monitoring* is another example of a cloud-based monitoring tool. There will be more on Kubernetes and containers in the upcoming lessons.

In the next lesson, we'll unpack multitenancy with regards to application hosting.