Bare Metal, Hypervisor, and VMs

This lesson covers bare-metal servers, hypervisors, and virtual machines in detail.

We'll cover the following

What is a bare-metal server?

A *bare-metal server* is a single-tenant physical server without any software installed. It's just raw hardware. When using a bare metal server, a business has sole access to the server, there is no sharing of resources with any other entity.

Originally, all servers used for the enterprise deployment were bare-metal servers set up on-prem. It was with the advent of virtualization, that running multiple applications in virtual environments on a single server became possible.

The use of bare metal servers ensures *performance, reliability*, and *security*. In the present computing landscape, bare metal servers are also available as a service termed as *Bare Metal as a Service*.

There are dedicated cloud providers that provide bare-metal hardware via the web to fulfill the infrastructure requirement of businesses.

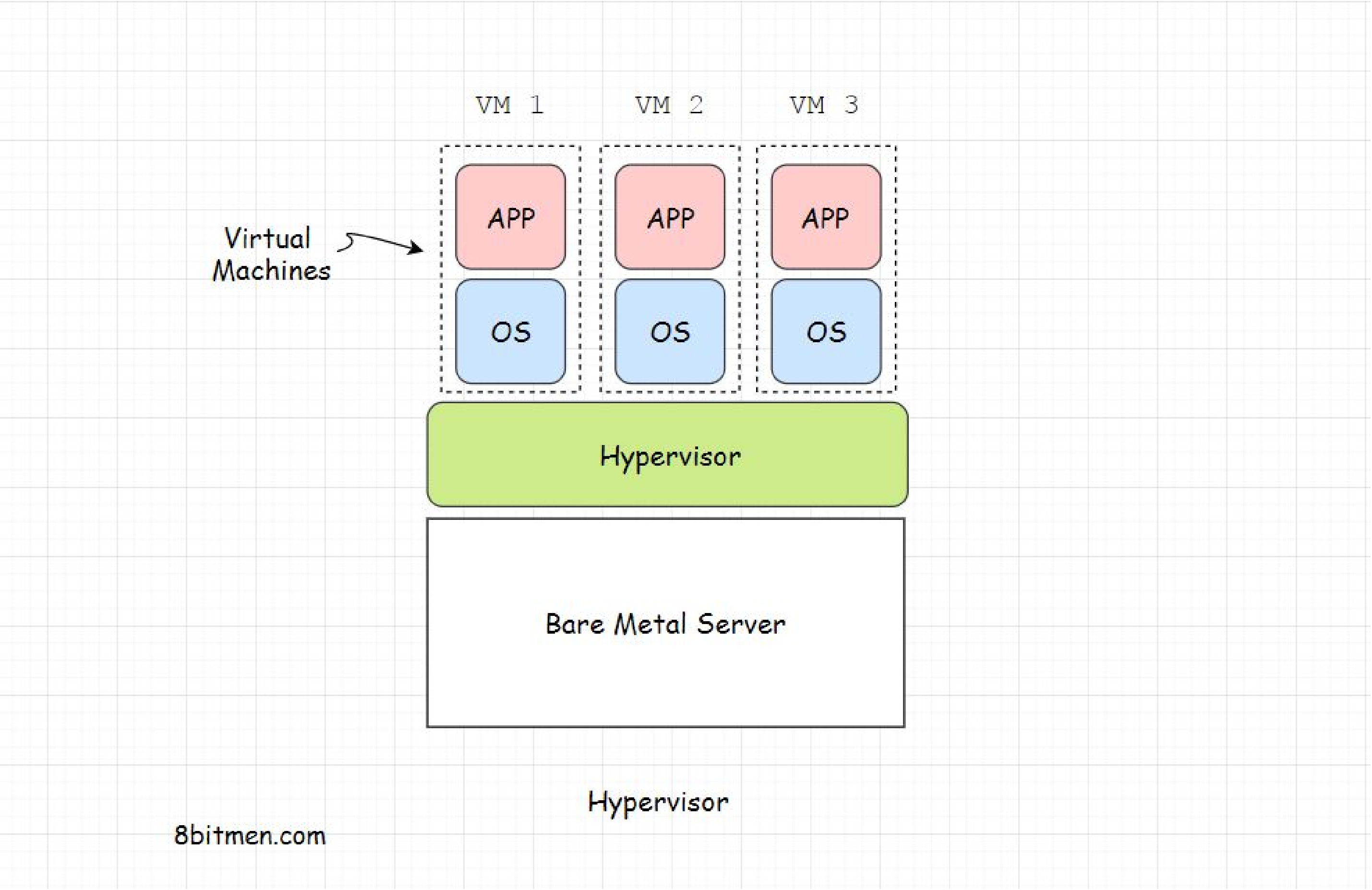
With software called *hypervisors*, a business can install multiple operating systems on a bare metal server to run multiple applications.

Let's learn more about hypervisors.

What is a hypervisor?

What is a hypervisor?

A *hypervisor, also known as the virtual machine monitor*, is a software that is installed on a bare-metal server to enable the deployment of multiple applications on it. This is done by creating virtual environments on the bare-metal server and splitting up its resources to be allocated to different applications.



After the installation of the hypervisor, After the installation of the hypervisor, a bare-metal server has several virtual machines, also known as virtual machines (VMs), running on it.

Virtual machine (VM)

A virtual machine provides virtual isolated environments having their own *CPU, memory*, etc. to run different applications. All the virtual environments, created on the same physical hardware, give the impression of multiple physical machines running multiple applications.

An operating system running in a virtual environment will run in the same way it would run directly on the physical hardware.

The need for creating several different virtual environments

Different virtual environments are created to make the most out of the resources of a physical server. A single application deployed on the bare metal server might only utilize 30% of the server resources. For the better utilization of the hardware, multiple applications are deployed on the bare metal with the help of virtualization.

This approach is more economical because it averts the need to purchase additional servers. It also facilitates less physical space consumption in the data center and cuts down energy usage.

The physical hardware that is the bare metal is called the *host*, and the multiple virtual machines created on the host are called the *guests*. A guest can also be moved from one host to the other.

Say I have a server running *Linux* with *32GB* of *RAM* and I want to run a couple of applications that would run on *Windows OS*. To pull this off, I can create three *VMs* on my server with *8 GB RAM* each, running Windows OS to run my applications.

In this scenario, running the applications on VMs averted the need for separate physical servers having Windows OS installed.

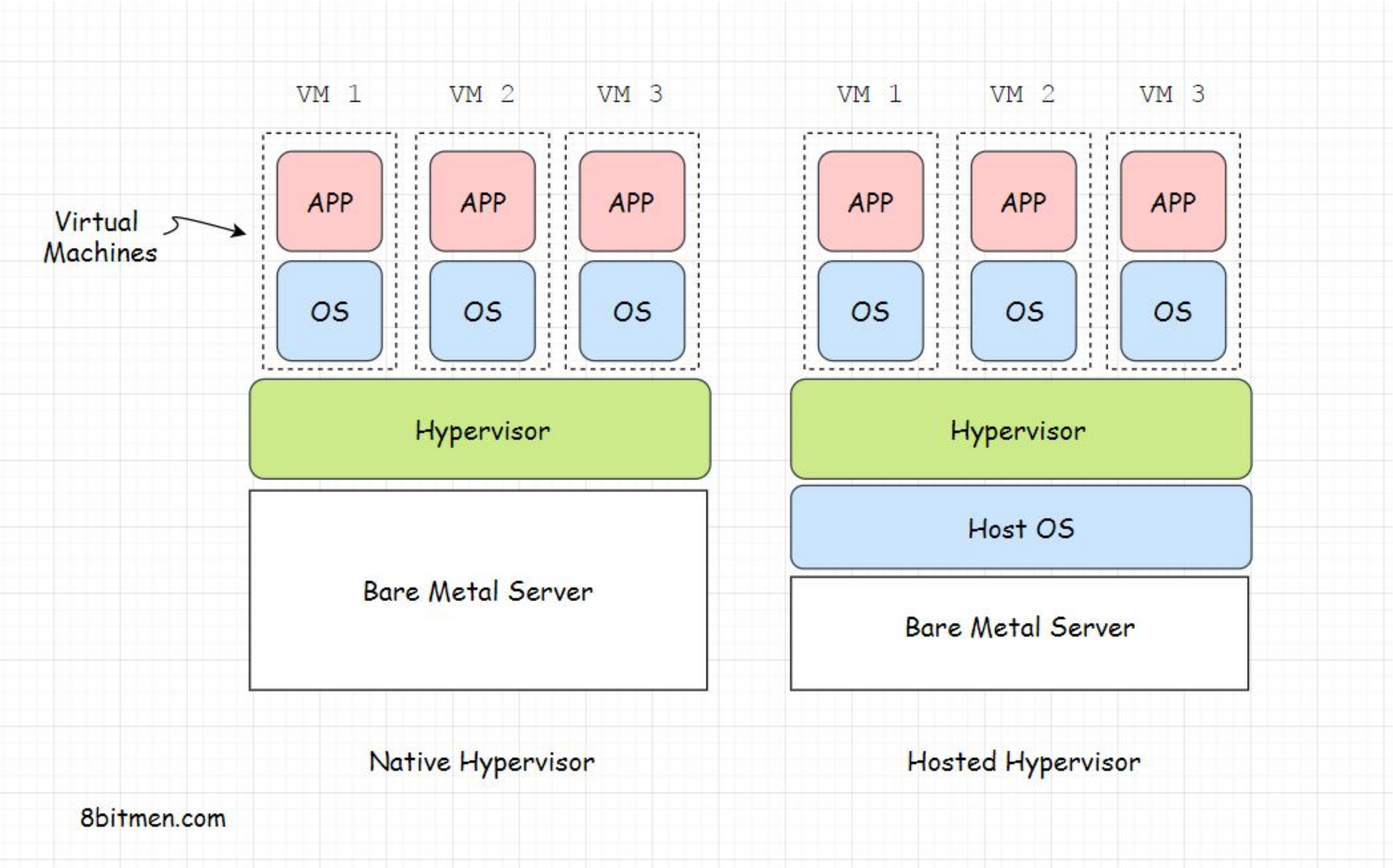
VMware is a company that has industry authority in providing virtualization software and services. It is one of the first companies to successfully virtualize the *x86* architecture.

If you need an understanding of how virtual memory is mapped to the physical memory, you can read about virtual memory on Wikipedia.

Types of hypervisors

There are two types of hypervisors that enable virtualization:

1. Those that run directly on the bare metal server, also known as the native hypervisors.
2. Those that run on the operating system installed on the bare metal server, also known as the hosted hypervisors.



Native hypervisor

These hypervisors run directly on the host. There is no *OS* involved; the hypervisor has direct access to the hardware. Since there is no *OS* involved, native hypervisors are the most efficient in terms of performance.

Hosted hypervisor

These hypervisors run on the operating system that is installed on the bare metal. The involvement of the *OS* in between adds a bit of latency in the virtualization process.

The upside of a hosted hypervisor is that, since it doesn't directly interact with the hardware, it can be easily installed on a wide range of hardware. The compatibility issues are minimal. However, setting up a hypervisor directly on the hardware is a bit complex.

So, choosing a between *type 1* and *type 2* hypervisor is a trade-off between complexity and performance.

In the next lesson, let's talk about containers.