

Introduction to Containers

This lesson provides insight into container technology.

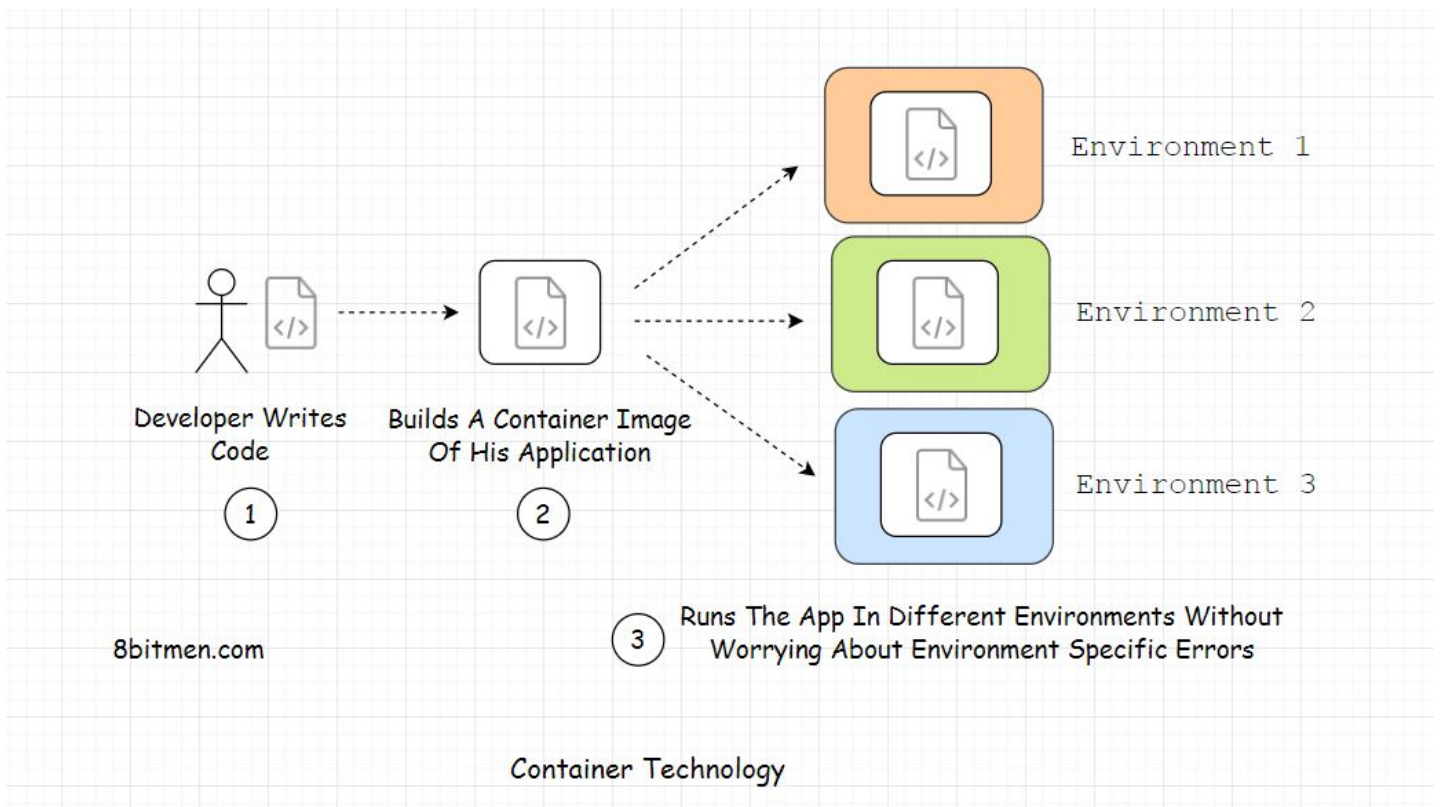
We'll cover the following

- What is a container?
- How does container technology improve developer productivity?

What is a container?

In cloud computing, *containers* are a technology that enables developers to package their software, including application code, dependencies, and configurations with specific versions of frameworks, programming languages, libraries, and so on, in a module as one standard unit.

This standard unit of software can be deployed across different environments, such as a public cloud, on-prem, or a developer's laptop, smoothly without facing any environment-specific errors.



Containers encapsulate the application from the environment it runs in and

ensures a consistent runtime for the software across different platforms. This increases the developer's productivity by notches.

How exactly?

Let's find out.

How does container technology improve developer productivity?

Before containers became mainstream common issue developers encountered when deploying their code on different environments, such as pre-production, production, or even on the fellow developer's laptop, was that the application worked totally fine on the machine of the developer who wrote the code but threw errors when being run in a new environment.

If you've deployed code in *Linux* boxes in the past, you surely remember library dependency errors popping up all the time. We had to manually update the box with a certain dependency to get rid of the errors and run our code.

I remember having a *txt* file that was shared with everyone on my team. The file had a list of all the errors that we might encounter when deploying the code in the *Linux* box and the ways to get rid of those errors. It was the responsibility of the developer to update the file with the proceeding steps, in case they came across a new error.

All this happened due to the difference in the environments of the developer's laptop and the new environment where the code was being deployed.

The new environment would have a different configuration, a different version of the technology installed, or a missing dependency. This would throw an error.

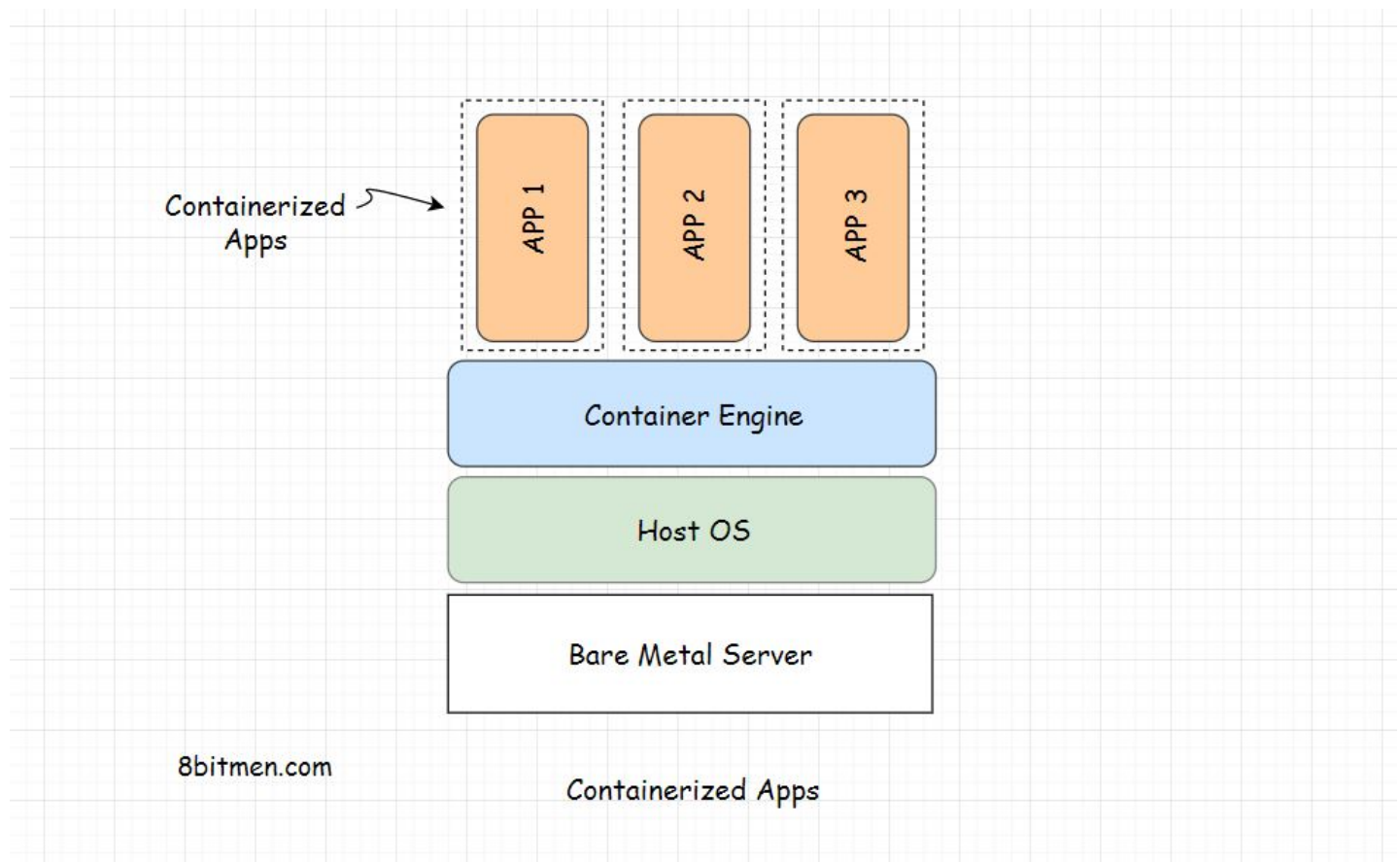
The developer would have to spend hours first trying to figure out the issue and then fix it.

Containers let you off the hook for these things since everything required to run the application, like the code, libraries, configuration files, and so on, are all packaged in a single module.

This module can be easily deployed in any environment without having to worry about environment-specific errors.

With containers, developers can focus on writing code. If the application works

With containers, developers can focus on writing code. If the application works fine on the developer's machine, they can be certain it will work fine in the new environment they run their code in.



Containerization provides an abstraction from the environment the application runs in.

Containers are the go-to tech in today's modern application development landscape because of the deployment benefits they offer. All the large-scale services like *YouTube*, *Google Search*, *Pokémon Go*, and modern banks run on containers.

Recommended read: [Pokémon Go powered by containers](#)

[Monzo](#), a modern digital bank, uses Docker containers with Kubernetes to run code.

Speaking of container technology, [Docker](#) is the most popular open-source container technology used in the industry (see details [here](#)) along with [Kubernetes](#), which is a container orchestration system. There will be more on container orchestration later.

These technologies facilitate automated continuous deployment, consistent

These technologies enable automated continuous deployment, consistent behavior across different platforms, container monitoring, and so on, which is key in modern application deployment. In the upcoming lessons, you will learn more about this.

In the next lesson, let's look at how containers are different from virtual machines.