

Life Before Docker

Docker solves the complex and time-consuming environment migration while deploying apps to production. It also adds extra layers to extend the container's functionalities.

We'll cover the following

- Development issues without Docker

Before you start learning Docker, you should know why you want to learn it and what you will be able to do with it. Before the birth of Docker, the software industry had some major issues. One of them was the unpredicted behavior of an application in case of server migration. Let's take a look at some of the issues the industry faced before Docker became widely used.

Development issues without Docker

1. **Dependency matrix:** Whenever software is built on top of any runtime or platform with a specific version of a language, e.g, Java 8 or Python 3.5, having the same version installed on the system was a challenge as it might need a different version for system applications.
2. **Time-consuming migration:** As soon as the software is migrated to the new environment, managers, developers, and the system administrators used to start hunting the bugs produced because of a new environment. One question that was frequently asked was "What is the difference between this environment and the last environment where everything worked fine?"
3. **"It works on my machine!":** This was the biggest problem whenever a new developer joined the team and needed the project to be set up on his system. For every troubleshoot, developers used to say "It works on my system. I don't know why it is not working on yours".



Sarcasm about the Docker machine

Docker addressed these issues and tried to solve these problems for their organization. Docker was an open-source project started by a company called dotCloud. After its adoption, Docker Inc. was founded. So, let's see Docker's functionalities and how it addresses all these issues.