

# AI-02072 Docker vs Traditional

■ Owner	© Rigel Arcayan
■ Tags	Project
■ Created time	@September 5, 2025 10:49 AM

**I'm new to Ubuntu and learning fast, where there is a web application I want to experiment, there is the normal installation and there is also a Docker installation. Explain the traditional setup vs Docker. Perform a pros/cons analysis. Which is better?**

This is an excellent question for anyone starting their journey in web development. The choice between a traditional setup and Docker hinges on your goals, particularly whether you're building a simple personal project or a more complex, reproducible, and portable application.

## Traditional Setup

A traditional setup involves manually installing all the necessary software and dependencies for your web application directly onto your Ubuntu machine. This is like building a house by hand, brick by brick. For example, if your application uses Python and a PostgreSQL database, you would use apt to install Python, pip to install its libraries, and then install and configure the PostgreSQL server.

- Pros:
  - Direct Access: You have complete, native control over the environment. There's no abstraction layer, which can make it easier to debug issues with system-level tools.
  - Performance: There is minimal performance overhead because the application runs directly on the operating system without an extra layer of virtualization.
  - Simplicity for Single Apps: For a single, straightforward application, the setup can seem simpler at first because you don't have to learn a new technology like Docker.
- Cons:

- **Dependency Conflicts:** This is a major drawback. If you want to run multiple applications that require different versions of the same library (e.g., Python 2 vs. Python 3), you'll likely run into version conflicts that are difficult to resolve.
- **Non-portable:** The setup is tied to your specific machine. If you want to run the application on another server or share it with a teammate, they have to repeat the exact same, often manual, installation process. This leads to the infamous "it works on my machine" problem.
- **Difficult Cleanup:** Uninstalling an application can be messy, leaving behind system-wide files and dependencies.

#### Docker Setup

A Docker setup uses containerization. Instead of installing software directly on your OS, you package your application and all its dependencies into a self-contained unit called a container. This container is an isolated environment that can be run on any system with Docker installed. It's like building your application inside a prefabricated, portable box.

- **Pros:**
  - **Consistency and Portability:** This is Docker's biggest advantage. The container ensures that your application runs identically everywhere—on your machine, on a teammate's computer, or on a production server. This eliminates environment-related issues.
  - **Dependency Isolation:** Each container is isolated, so different applications can use conflicting versions of libraries or dependencies without affecting one another.
  - **Ease of Deployment:** With a single command (`docker-compose up`), you can spin up the entire application stack, including the web server, database, and any other services.
  - **Cleanliness:** When you're done, you can simply remove the container, and no lingering files are left on your host system.
- **Cons:**
  - **Learning Curve:** There is an initial learning curve to understand concepts like images, containers, Dockerfiles, and `docker-compose`.

- Performance Overhead: While minimal and often negligible, there is a small performance penalty compared to a native traditional setup because the application runs inside a container.
- Debugging: Debugging can be slightly more complex as you may need to "enter" the container to inspect files or run commands.

Which is Better?

For experimenting with a web application, Docker is the better choice. While the initial learning curve might seem like a barrier, the benefits of consistency, isolation, and portability will save you countless headaches in the long run. Docker is the industry standard for modern application development and deployment. It solves the most common and frustrating problems developers face when setting up environments and ensures that your experiments will work just as well on another machine as they do on yours.

You can learn more about the differences between Docker and a traditional installation in this video. [Docker vs Traditional Installation](#)