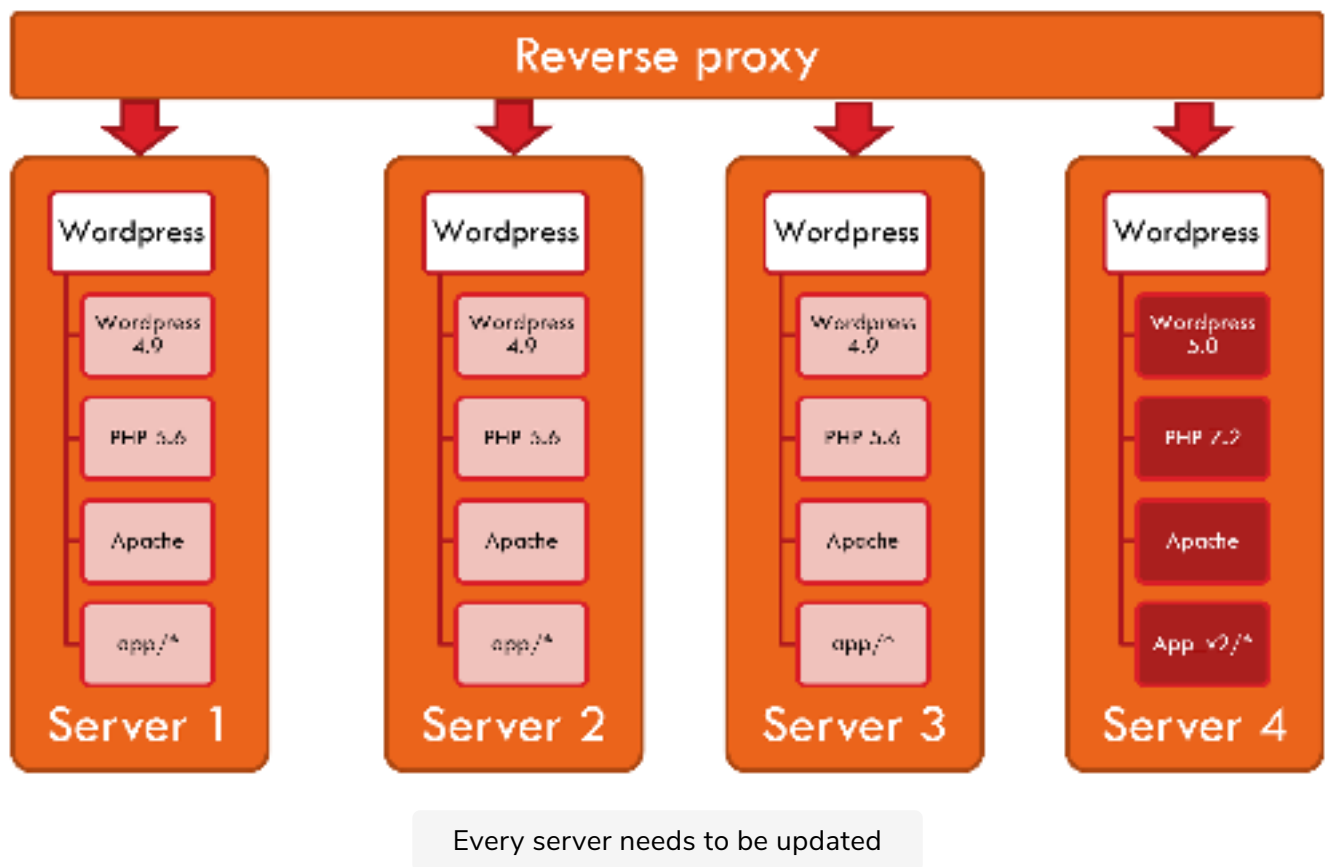


# Allows Seamless Upgrades

In this lesson, you will learn how containers simplify the server upgrading process.

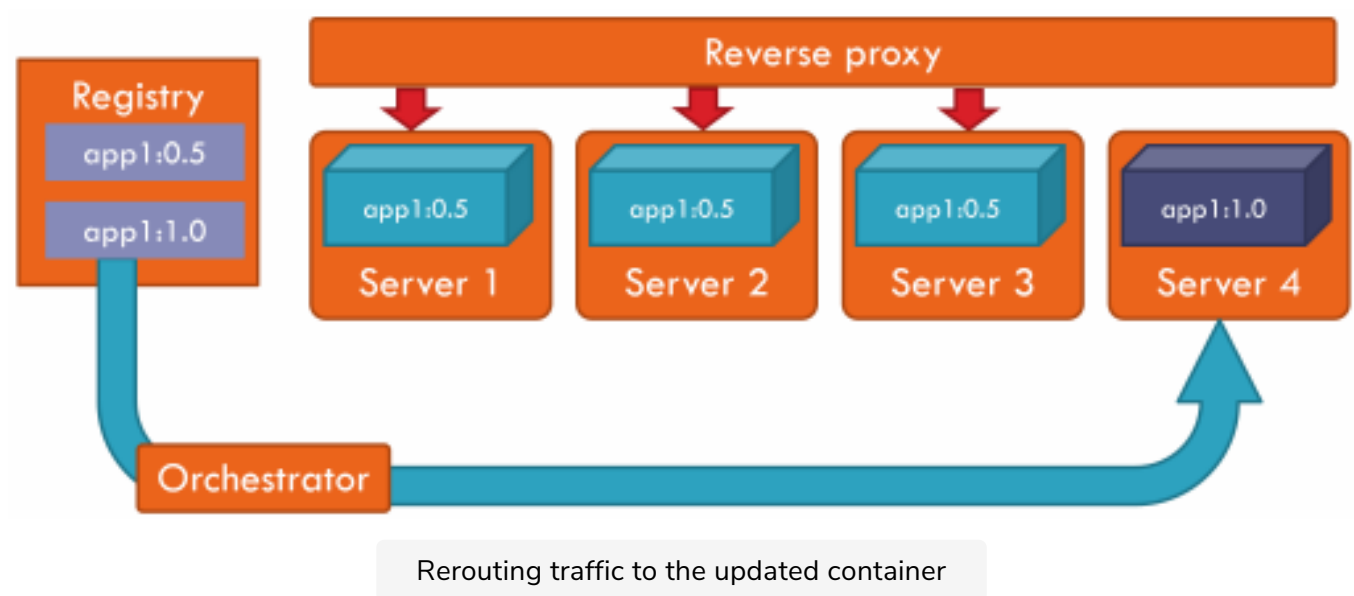
Even in scaled-up scenarios, a container-based approach makes tricky concepts seem trivial. Without containers, your favorite admin will not be happy with you if he has to update every server, including the dependencies.



Of course, in such a case, the update process depends on the application and its dependencies. Don't even try to tell your admins about DevOps if you want to remain alive.

By using containers, it's a simple matter of telling the orchestrator that you want to run a new image version, and it gradually replaces every container with another one running the new version. Whatever technology stack is running inside the containers, telling the orchestrator that you want to run a new image version is a simple command (in fact, by changing just one line, as we'll see).

The illustration below shows the process as it goes on: the orchestrator replaces one container, and then moves on to the other ones. While the new container is not ready, traffic is being routed to the old version containers so that there is no interruption of service.



In the next lesson, we will see how containers are used in the physical world.