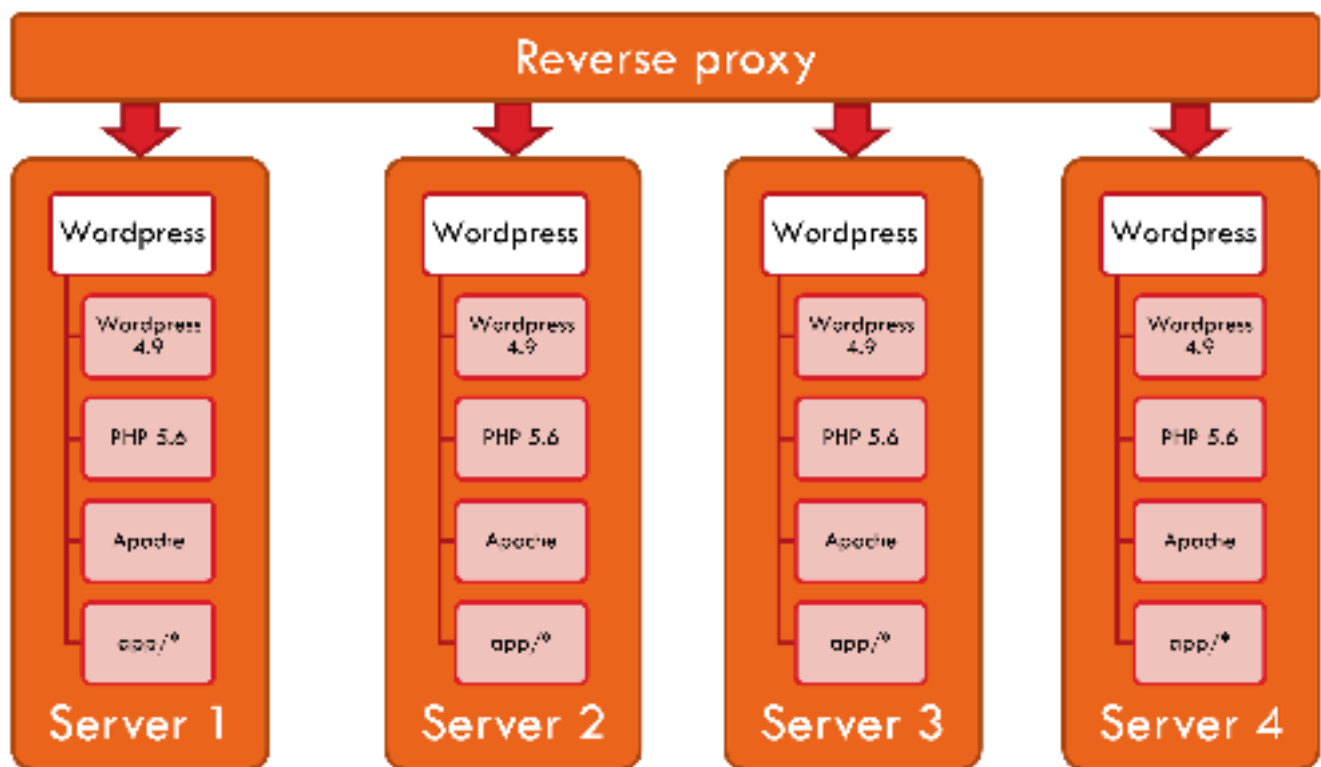


Allows Easy Scaling Up

In this lesson, you will learn how Docker containers help in scaling when a single server isn't enough to handle a single application.

When a server application needs to handle a higher usage than what a single server can handle, the solution is well-known, place a reverse proxy in front of it, and duplicate the server as many times as needed. In our previous Wordpress application example, this meant duplicating the server together with all of its dependencies:



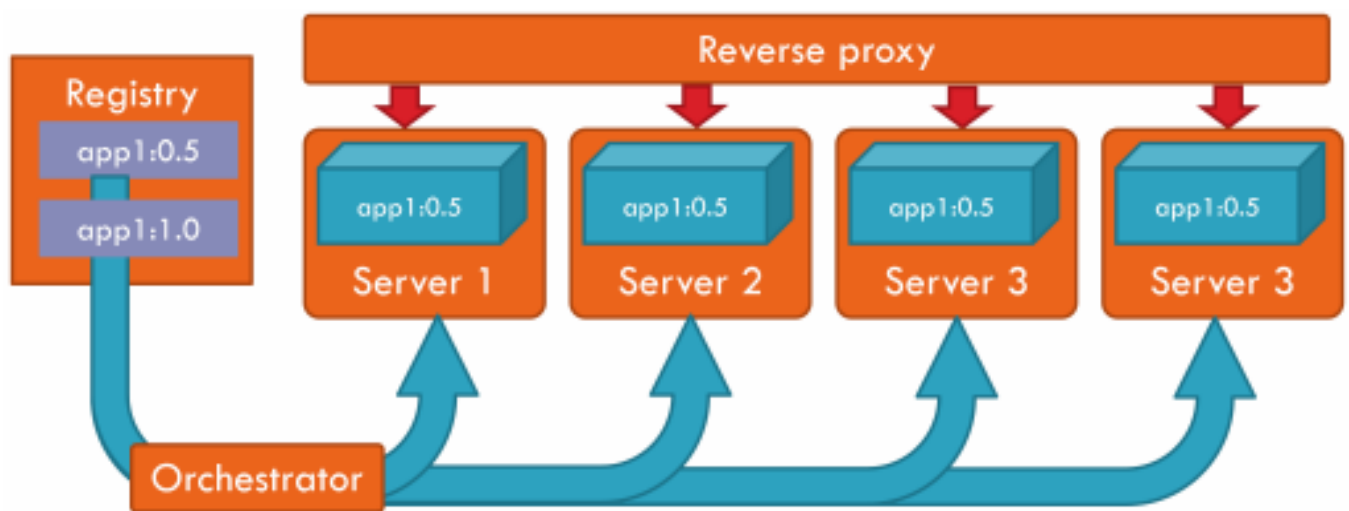
A model of using a reverse proxy with duplicate servers

That is only going to make things worse when upgrading: we'll need to upgrade each server's dependencies together with all of the conflicts that may induce.

Again, containers have a solution for this. As we'll see in the [Basic Concepts](#) chapter, containers are based on images. You can run as many containers as you wish from a single image — all the containers will support the exact same

dependencies.

Better yet: when using an orchestrator, you merely need to state how many containers you want and the image name and the orchestrator creates that many containers on all of your Docker servers. We'll see this in the orchestrators part of this course. This is how it looks:



A model of creating duplicate containers using an orchestrator

In the next lesson, we will look at yet another advantage of containers.