Basic Concepts

In this lesson, you will be introduced to some basic concepts which are essential to understand before moving forward in the course.

WE'LL COVER THE FOLLOWING

- Containers
- Images
- Registries

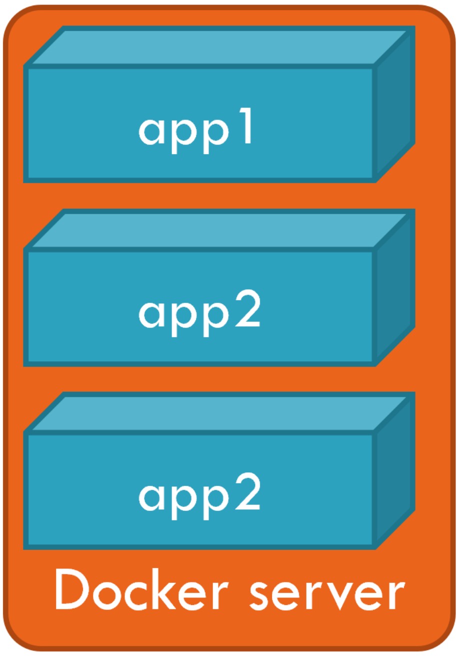
There are three concepts I need you to grasp before we begin: **containers**, **images**, and **registries**.

Containers

A *container* is what we eventually want to run and host in Docker. You can think of it as an isolated machine, or a virtual machine if you prefer.

From a conceptual point of view, a *container* runs inside the Docker host isolated from the other containers and even the host OS. It cannot see the other containers, physical storage, or get incoming connections unless you explicitly state that it can. It contains everything it needs to run: OS, packages, runtimes, files, environment variables, standard input, and output.

Your typical Docker server would look like this — a host for many containers:

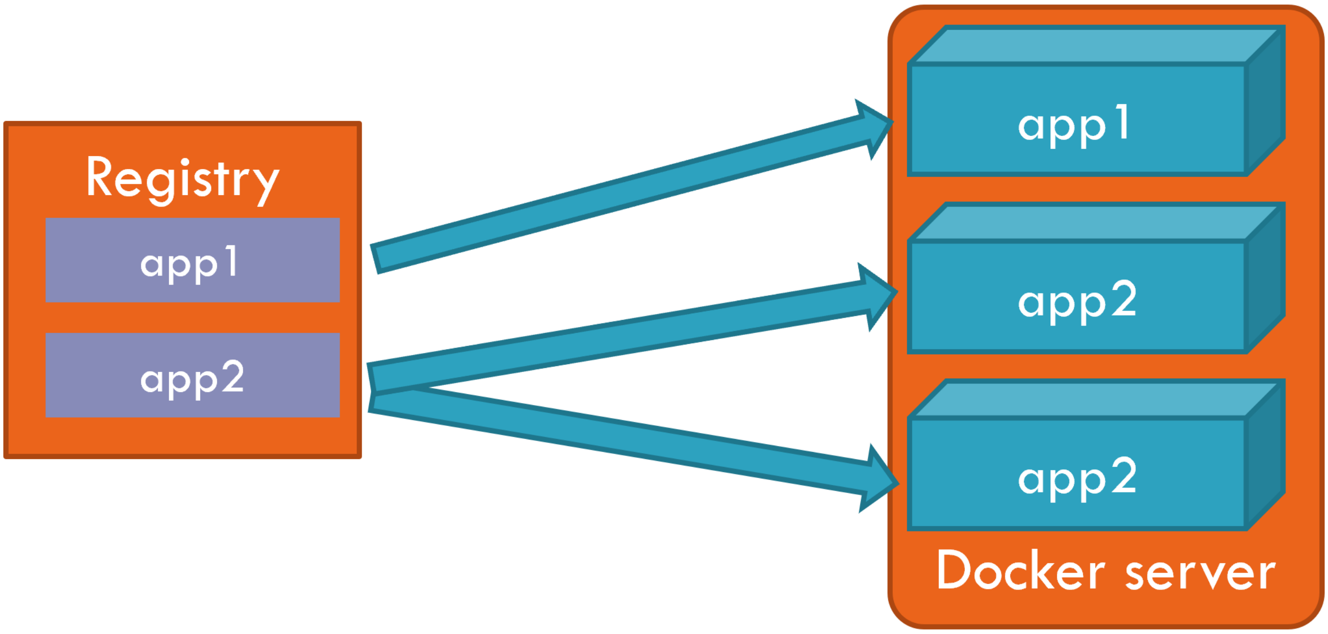


The fact that there are two *app2* containers in the schema above is normal; this is typically the case when a server hosts a release and a test version. Which means you could host both versions on the same server. In fact, each container has its own ID, but let's keep things simple for now.

Images

Any container that runs is created from an *image*. An image describes everything that is needed to create a container; it is a template for containers. You may create as many containers as needed from a single image.

The whole picture looks like:



Creating containers using images

Registries

Images are stored in a *registry*. In the example above, the *app2* image is used to create two containers. Each container lives its own life, and they both share a common root: their image from the registry.

In the coming chapters, you're going to learn how to run containers from existing images, then you'll learn how to create your own images and also upload them to registries.

Let's wrap up this chapter with a quiz to test what you have learned so far.