Docker is perfect for helping you with the development lifecycle. Docker allows your developers to develop on local containers that contain your applications and services. It can then integrate into a continuous integration and deployment workflow.

For example, your developers write code locally and share their development stack via Docker with their colleagues. When they are ready, they push their code and the stack they are developing onto a test environment and execute any required tests. From the testing environment, you can then push the Docker images into production and deploy your code.

**A Docker image is** a read-only template. For example, an image could contain an Ubuntu operating system with Apache and your web application installed. Images are used to create Docker containers. Docker provides a simple way to build new images or update existing images, or you can download Docker images that other people have already created. Docker images are the **build** component of Docker.

**Docker Registries**

Docker registries hold images. These are public or private stores from which you upload or download images. The public Docker registry is called [Docker Hub](http://hub.docker.com/). It provides a huge collection of existing images for your use. These can be images you create yourself or you can use images that others have previously created. Docker registries are the **distribution** component of Docker.

#### Docker containers

Docker containers are similar to a directory. A Docker container holds everything that is needed for an application to run. Each container is created from a Docker image. Docker containers can be run, started, stopped, moved, and deleted. Each container is an isolated and secure application platform. Docker containers are the **run** component of Docker.