Introduction to Docker

In this lesson we will learn about Docker and how it can be used run the infrastructure components needed by our microservices.

What is a docker?

Docker is a platform that provides the ability to package and run an application in a loosely isolated environment called a container. Here's how it works:

We start with our Catalog service, which as we know needs a database to store and query for items. The piece of software that knows how to standup and operate this database, in our case, is MongoDB. The MongoDB server, along with all its software prerequisites, has already been packaged in what we call a docker image and this image is publicly available in a place called Docker Hub.

We want to download and run this image in our box and to do that we use the docker engine, which is one of the pieces you installed at the start of this module. All docker images are guaranteed to run anywhere the docker engine is available.

To pull and run this image we will use the **docker run** instruction in our box. Once the image is run or executed it becomes what is known as a docker container, a MongoDB container in this case. This is a fully working MongoDB server ready to respond to the requests of our Catalog microservice.

The Catalog service will create a database via the MongoDB docker container and will start interacting with it via the MongoDB driver for .NET. Notice that even when this container manages the catalog database, the database itself does not live inside the container, but outside of it, to ensure it is not deleted once the container is destroyed.

Let's try it out.