Play with Docker

Concepts in this exercise:

* Docker engine
* Containers & images
* Image registries and Docker Store (AKA Docker Hub)
* Container isolation

**Comparing container with virtual machine.**

1. The VM is a ***hardware* abstraction**: it takes physical CPUs and RAM from a host, and divides and shares it across several smaller virtual machines. A container is an ***application* abstraction**: the focus is really on the OS and the application, and not so much the hardware abstraction.
2. Many customers actually use both VMs and containers today in their environments and, in fact, may run containers inside of VMs.

**Commonly Used Commands**

docker image pull {image\_name} : pull image from a registry

docker image ls : list available images

docker container run alpine ls -l : start a container, After the ls command finished, the container shut down.

docker container run -it alpine /bin/sh : run the container in an interactive terminal

docker container ls [-a] : list containers

1.2 Container Isolation

Even though each docker container run command used the same alpine image, each execution was a separate, isolated container. Each container has a separate filesystem and runs in a different namespace;

1.3 Terminology

* *Images* - The file system and configuration of an application which are used to create containers. When you executed the command docker container run hello-world, it also did a docker image pull behind the scenes to download the **hello-world** image.
* *Containers* - Running instances of Docker images — containers run the actual applications. A container includes an application and all of its dependencies. It shares the kernel with other containers and runs as an isolated process in user space on the host OS.
* *Docker daemon* - The background service running on the host that manages building, running and distributing Docker containers.