Week 13

What's Next? Containers

Even if you are giant company, most likely you are still not buying 100,000 machines with identical hardware and software. The solution that many companies reached was to *virtualization*. If we want to double the number of our Web Servers, wouldn't it be great if we could just clone an *image* of a working Server as many times as needed? But in order to impose that consistency, we needed to *virtualize* that hardware and software. Virtual machines are a good solution, but they are expensive in terms of resources.

So now the current trend is away from virtual machines and towards *containers*. Containers are similar to virtual machines, except that they don't need to virtualize all hardware. Containers will use the local machine's kernel, but will run from an image that contains all relevant dependencies and configuration. They are essentially lightweight virtual machines, running a single application or sometimes even running a single function in a wider application.

A Virtual Machine is like a House. Each house has its own furnace, water heater, plumbing and structure. Houses are expensive to build and to buy.

A Container is like a unit in an apartment building. The unit is connected to the building's plumbing and heating, but within the unit you can set up the furniture the way you like it. Apartments are cheap to buy and build.

Cloud Services such as Amazon AWS are running from Containers, and they are a Linux-only tool. As more and more computing happens in the Cloud, Linux has never been more relevant. And having at least a surface understanding of Containers is a valuable skill to put on your resume!

Introduction to Docker

Docker is still the most common way of setting up Containers.

Docker also maintains a repository of ready-made images that you can *pull* from the Internet.

Installing Docker

Refer to Docker's Installation Guide to install Docker. The process is a bit simpler in Linux, since we require the Linux kernel. But in Windows, we can simply use a Linux virtual machine to run it.

Getting to Hello World

```
docker run hello-world
```

```
Hello from Docker.
This message shows that your installation appears to be working correctly.
...
```

The first container you run will print 'hello world' and exit.

```
docker pull busybox
```

Since this is a new container, we will *pull* its image from a Docker registry.

```
docker run busybox
```

The first time you run this container, it exits quietly. However, you can see these images with a familiar command:

```
docker ps -a
```

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES
305297d7a235        busybox             "uptime"            11 minutes ago
Exited (0) 11 minutes ago                       distracted_goldstine
14e5bd11d164        hello-world         "/hello"            2 minutes ago
Exited (0) 2 minutes ago                        thirsty_euclid
```

Introduction to Kubernetes

Kubernetes is a keyword you'll see in a lot of job postings. Kubernetes is an open source tool for *orchestration*. Instead of running a few 'hello world' containers, perhaps you are running a *cluster* of containers, with databases, web pages, web apps and so on. If you scale up an application, you will have many containers running together and depending on one another. This is where Kubernetes comes in.

There are some helpful tutorials hosted on the kubernetes website. I encourage you to explore some of these resources, since you may find that having even a basic understanding of these tools will come in handy when you are talking to potential employers!