Containers, containers, containers

**Current abstract**

Everyone loves containers. HP Helion and Cloud Foundry provide more than just your standard container management platform. CF provides full application management, service management and is driven by leveraging the power of containers. In this session we will take a look at containers in CF 2.0, how Helion Stackato handles containers, OCI & RunC, Containers in CF 3 and how we think about Windows today and tomorrow.

# Outline

* Define containers
* Containers in CF 2.0
* Containers in Helion Stackato
* OCI (Open Container Initiative) and RunC
* Containers in CF 3.0
* Windows Today and Tomorrow?

Background Reading

* Container Background
  + <http://diginomica.com/2014/07/02/virtualization-dead-long-live-containerization/#.VkDT_q6rQ0o>
* Have some working knowledge of Docker
  + <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-getting-started>
  + <https://www.datadoghq.com/docker-adoption/>
* Cloud Foundry 2.0 containers
  + read the warden doc - <https://docs.cloudfoundry.org/concepts/architecture/warden.html>
  + Good general discussion on containers <http://anandmanisankar.com/posts/container-docker-PaaS-microservices/>
  + another great article - <http://blog.altoros.com/cloud-foundry-containers-warden-docker-garden.html>
* For HPE Helion Stackato have
  + installed Helion Stackato Virtual Box - <http://www.davidaiken.com/hpe-helion-stackato-tutorials/installing-hpe-helion-stackato-virtualbox/>
  + deployed an application - <http://www.davidaiken.com/hpe-helion-stackato-tutorials/deploying-your-first-application-to-hpe-helion-stackato/>
  + deployed an application on a container - <http://www.davidaiken.com/2015/11/03/introducing-hpe-helion-stackato/>
* Let diego manage your docker app - <https://www.youtube.com/watch?v=tQC8vz1dedI>
* Diego update - <https://www.youtube.com/watch?v=SSxI9eonBVs>
* Garden and RunC - <https://www.cloudfoundry.org/garden-and-runc/>
* be familiar with the work of the OCI - <https://www.opencontainers.org/>
* be familiar with RunC - <https://runc.io/>
  + RunC tutorial - <https://www.cloudgear.net/blog/2015/getting-started-with-runc/>
* Windows Containers - <https://msdn.microsoft.com/en-us/virtualization/windowscontainers/containers_welcome>
* Pivotal CF & Windows <https://blog.pivotal.io/pivotal-cloud-foundry/products/windows-docker-and-buildpack-apps-in-one-platform>

# Docker Demo

## Setup

Grab the docker-demo repo from github <insert github url>

Install Docker for your Windows, Mac or Linux machine.

Create a Docker Hub account.

Tip: The first time you run through the demo, the image uploads will to docker hub will take some time - as will the first stackato deployment. After the first run the stackato deployment will be faster.

### Prepping the base

To speed up the demo and to ensure you don’t need a mad fast internet connection, you can perform the following steps on a fast network.

On Mac or Windows - start the docker terminal

Login to dockerhub with your login credentials

docker login

change to the base folder

review the contents of **Dockerfile**

Build the base image using:

docker build -t username/my-docker-base .

For example:

docker build -t slimypit/my-docker-base .

where slimypit is the docker username

push the container to dockerhub

docker push username/my-docker-base

In the hello/src and hello2/src folders, update the Dockerfile to use the image you just created.

## Demo

Open the docker terminal if you have not already.

view the docker images using:

docker images

Explain this docker base image is something you created from the ubuntu image.

Review the Dockerfile from the base folder - showing the commands that have been executed.

Change to the hello/src folder

review the Dockerfile - check the FROM points to your own base image you created earlier.

Run

docker build -t username/hello .

This will build the image with the node.js application. it should be pretty darn fast.

You can run the app using:

docker run -d -P --name hello1 username/hello

View the running apps using:

docker ps

You can see which port 8080 (remember that from Dockerfile) was mapped too by using:

docker port hello1

If you are on a Mac or Windows machine - use the following and locate the ip address:

docker-machine env default

navigate to the ip from env default and port from the port command.

For example http://192.168.1.63:32771

Change to hello2/src

review the Dockerfile - check the FROM points to your own base images you created earlier.

Run

docker build -t username/hello2 .

docker run -d -P --name hello2 username/hello2

docker ps

docker port hello2

navigate to the new port in your browser.

push the docker image to dockerhub - then flip back to the presentation - we will come back to this in a later demo.

docker push username/hello

## Deploying a Docker Image to Stackato

For the stackato demo - you could run through the whole deploy an app scenario from the post here - <http://www.davidaiken.com/hpe-helion-stackato-tutorials/deploying-your-first-application-to-hpe-helion-stackato/>

The quick version of the docker part of this demo uses a pre-defined container:

stackato push --docker-image slimypit/stackato-node-hello --as hello -n

You can also use the container you used in the previous demo.