* Course Overview
  + Get up and running with docker
* Introduction
* Module Intro
  + How to install docker
  + Desktop installs
    - Windows, mac
  + Server install
    - Windows, linux
  + Cloud installs
    - Aws, azure
* Docker for Windows
  + A product
  + Get docker environment locally
  + windows 10 64 bit only
  + for test and dev work
  + not for production
  + docker engine will be running on linux inside of hyper v VM
  + can only run linux containers
  + native docker for windows 10 is coming
  + turn on Hyper-v/install Hyper-V
  + program and features -> turn windows features on or off -> check ‘Windows Hypervisor Platform’
  + go to docker website
  + download for windows
  + open command line when docker is installed and running
  + to check version type ‘docker version’
* Docker for Mac
  + linux VM
  + hyper kit
  + data kit
  + moby linux
  + need a moder mac(2010 an on)
  + Os X 10.10.3 or newer
  + go to docker website and install docker for mac
  + open terminal
  + type ‘docker info’ to get information about docker set up
  + type ‘docker images’ to get list of images
* Installing Docker on Windows Server 2016
  + native windows server container
  + open power shell
  + type ‘docker version’ to get the version
  + type ‘docker info’ to get information about currently running docker
  + download a base os image
    - type ‘Install-PackageProvider ContainerImage -Force’
  + find images
    - type ‘Find-ContainerImage’
  + grab image
    - type ‘Install-ContainerImage [name]’
  + to find list of download image
    - type ‘docker images’
  + unless you tell which image version or tag to go with, then the one tagged as latest is run
  + to tag image as latest
    - type ‘docker tag [image id] [image name]:latest
  + to run container
    - type ‘docker run -it [image name] cmd’
    - -it for attach powershell terminal
    - cmd to run cmd shell process inside of it
  + if we didnt tag it as latest, you will have to add the default tag
* Installing Docker on Linux
  + on Ubuntu
  + run command ‘wget -q0- <https://get.docker.com/> | sh’
    - grab install script and pipes it through shell
  + dont want to abuse root account to run docker
* Module Summary
* What is a Container
  + container split up OS resources
    - process ids, NET, File system
    - OS virtualization and assign one to each container
  + hypervisor virtualization, virtualizes physical server resources and builds virtual machines
* The ‘docker run’ Command
  + type ‘docker version’
    - to get version of client and server version
  + type ‘docker info’
    - shows info about docker host
  + type ‘docker run [image name]’
    - to spin up container
    - client talks to deamon
    - deamon finds image on machine or look in docker hub
    - use image as template to create container
  + type ‘docker ps’
    - show container currently running
  + type ‘docker ps -a’
    - shows containers that ran but is now excited
  + type ‘docker images’
    - see images
  + images
    - tag: can version it
    - id: unique hash
    - created
    - size
* Theory of Pulling and Running
  + docker host runs docker client and docker daemon
    - docker engine
  + installing docker gives the client and daemon on same host
  + docker run makes the appropriate api calls to the daemon
  + daemon implements the docker remote api(engine api)
  + docker run starts a new container with the image specified as a template
  + daemon checks local store to see if it already has a copy
    - if it doesn’t it will search for it on docker hub
  + docker hub: docker image registry
    - <https://hubs.docker.com>
    - place to store image to use for containers
    - default but can change registry
  + daemon then pulls image locally
  + daemon then create and spin of container based on configuration in image
* Working with Images
  + Images: stopped containers
  + Containers: running images
  + type ‘docker pull [image name]’ to pull images without starting container
    - looks for image in docker hub
  + type ‘docker pull [image name]:[version number]’ to pull a specific version of an image
  + type ‘docker images’ to show list of images in local image store
  + sign up for docker hub
  + type ‘docker rmi [image]:[tag]’ to delete image from local image store
* Container Lifecycle
  + can start, stopped, restrated, removed
  + ‘docker start [container]’
  + ‘docker stop [container]’
  + ‘docker rm [container]’
  + if you just stop and start containers, the data will persist
  + data will be removed if you remove the container
  + ‘docker run -d --name [name] -p [port:number] [image name]’
    - docker run: ask daemon to create new container
    - -d: start container in detached mode
      * don’t latch it on terminal
    - --name: unique name
    - -p: map network ports
      * ex) 80:8080
      * the second part after: is the port the web server listening to
      * the first part before: port of the docker host
      * map port 80 on docker host to port 8080 inside of the container
  + top level images are stored in the root of the hub
    - ex) nginx, busybox, ubuntu, redis, alpine
  + other images you will need to specify the namespace
    - ex) dockercloud/haproxy, puhsion/baseimage
  + type ‘docker ps’ to see containers
  + type ‘docker stop [container name]’ to stop the container
  + type ‘docker start [container name] to start container
  + type ‘docker run -it --name [name] [image name] /bin/bash
    - to interact with container using a terminal
    - will be inside container in terminal
  + containers are usually single process constructs
  + to get out of container without killing type ‘Ctrl P + Q’
  + typing exit will stop the container
  + type ‘docker stop $(docker ps -aq)’ to stop all containers
    - $(): gives output of inner command
    - docker ps -aq: list all containers
    - q tell it to just return container id’s
* Lesson Recap