* Course Overview
* Course Intro
  + Containers vs virtual machine
  + Docker
  + Kubernetes
  + Workloads
  + Orchestration
* Containers: Primer
* The Bad Old Days
  + Application for the most part on servers
  + One server per app
* Hello VMware
  + Can runs tons of apps on a single server
  + Not a perfect solution
* Vm Warts
  + Server will create a VM for each app it needs to run
  + Each one is a slice of the physical server hardware
  + Each VM will need its own dedicated operating system
  + Each OS will take a sizable portion of the resources to run
  + License cost, admin(patching, etc)
* Containers
  + Instead of multiple VM and multiple OS
  + Install one OS then on top of that create containers for each application
  + Each container is a slice of the OS
  + Inside the containers we run apps
  + Containers are smaller and more efficient than virtual machines
  + Virtual machine: software construct that acts as physical machine
    - OS steals resources
  + Containers are ideal for spinning things up and tearing things down on demand
* Container Demo
  + Docker can run on VMs
  + Docker does linux and windows
  + Docker on linux only runs linux apps
  + Docker on windows only runs window apps
  + May be possible to run your linux apps on docker on windows
  + Dock image: prepacked application or a vm template
    - Got everything wrapped up into a bundle that you need to run an application
  + Command line to fire up container from the image
    - docker container run -d - -name web -p 8080:8080 nigelpoulton/ctr-demo:1
    - docker container run -d is command fire up contain
    - - - name web sets the name of the contain to web
    - -p 8080:080 sets the network port to expose
    - nigelpulton/ctr-demo is the name of the image
  + Command line to stop container
    - docker stop [name of container]
    - ex) docker stop web
  + command line to start container again
    - docker container start [name of container]
    - ex) docker container start web
* Recap
  + Monolith/Legacy App has everything baked into single binary(program)
    - All functions in one program
    - Nightmare to update and fix
    - Have to update the entire codebase
  + Cloud native, microservice
    - Break out components and make each on its own app or service
    - The components still take to each other
  + Containers pave way for cloud service and microservice
  + Containers and virtualization usually are used together
* Docker
  + Docker inc, docker project
* Docker, Inc
  + Start as company(dotcloud) that provided developer platform on top of AWS
  + Had been using containers to build their platform on top of AWS
  + Had tech as internal tool to help spin up and manager container
  + Turn that tool to Docker
* Docker the Technology
  + Containers are like fast light weight VMs
  + Docker makes running apps inside of containers really easy
  + Community Edition(CE)
    - Open source
    - Free
    - Can contribute back
  + Enterprise Edition(EE)
    - Additional features
    - Official support
  + Containerizing Apps: apps running in containers
* Docker Demo