* Course Overview
  + Kubernetes building blocks
  + Role of pods
  + Services
  + Secrets
* Overview
  + Kubernetes from a developer perspective
  + Creating pods
  + Creating deployments
  + Creating services
  + Understanding storage options
  + Creating ConfigMaps and Secrets
* Introduction
  + Can use Kubernetes in non production environment
* Kubernetes Overview
  + Open source system for automating deployment, scaling and management of containerized applications
  + How are you managing containers
    - Load balance to server with containers
  + Can use docker compose to manage containers
    - Not meant for production but can be used
  + Kubernetes is the conductor of a container orchestra
  + Key Kubernetes Features
    - Service discovery, load balancing
    - Storage orchestration/volumes
    - Automate rollouts/rollbacks
    - Self healing
    - Secret and configuration management
    - Horizontal scaling
    - More features
* The Big Picture
  + Container and cluster management
  + Kubernetes open source project
  + Supported by all major cloud platforms
  + Provides a “declarative” way to define cluster’s state
  + One or more master nodes
    - Manage worker nodes
  + Together they create a cluster
  + Master will start a pod on each node
  + Pod is a way to host a container
  + Pod is suit and container is a person in the suit
  + Deployment and replicaset to deploy pods
  + Need service for pods to talk to each other and the outside world
  + Node is like a VM, can run one or more pods
  + Store and controller manager
    - Store is database for master node to track things
    - Controller manager deals with request scheduling
  + kubectl command line tool
  + kubelet registers node with cluster and talks to manager
  + container runtime
  + Kube-Proxy: unique ip address for pods
* Benefits and Use Cases
  + benefits
    - accelerate developer onboarding
    - eliminate app conflicts
    - environment consistency
    - ships software faster
  + Key Kubernetes benefits
    - orchestrate containers
    - zero downtime deployments
    - self healing
    - scale containers
  + develop use cases
    - emulate production locally
    - move from docker compose to kubernetes
    - create an end to end testing environment
    - ensure application scales properly
    - ensure secrets/config are working properly
    - performance testing scenarios
    - workload scenarios(CI/CD and more)
    - learn how to leverage deployment options
    - help devops create resources and solve problems
* Running Kubernetes Locally
  + Minikube
  + docker desktop
  + kind: kubernetes in docker
  + kubeadam: full kubernetes, administrators
  + click on docker desktop icon
    - click on on settings
    - click on kubernetes
    - check enable kubernetes
    - click apply & restart
  + if it doesnt work, click restart docker
* Getting Started with kubecti
  + kubectl command line tool
  + type ‘kubectl version’ to check kubernetes version
  + type ‘kubectl cluster-info’ to get cluster information
  + ‘kubectl get all’ retrieve information about kubernets pods, services, etc
  + ‘kubectl run [container-name] --image=[image-name]’ simple way to create a deployment for a pod
  + ‘kubectl port-forward [pod] [ports]’ to forward a port to allow external access
    - normally a pod has a cluster ip, so only items in cluster can talk to pod
    - port-forward to expose it to external items
  + ‘kubectl expose ..’ expose a port for a deployment/prod
  + ‘kubectl create [resource]’ create a resource
  + ‘kubectl apply [resource’ create or modify a resource
  + ‘kubectl get pods’ to get all the pods
  + ‘kubectl get services’ to get all the services
* Web UI Dashboard