* Docker is a containerization platform
* Kubernetes is a container orchestrator for container platforms like docker
* The Rise of Containerization and Docker
  + Containers solve a critical issue in the life of application development
  + Developers write code on their own local development environment
  + Moving the code to production causes problems
  + Containers solves the issue of portability allowing you to separate code form the underlying infrastructure it is running on
  + Developers package up their application including all b ins and libraries into a small container image
  + In production that container can be run on any computer that has conterization platform
* Advantages of Containers
  + Containers and container platform provide many advantages over traditional virtualization
  + Containers have an extremely small footprint
  + Virtual machines requires a complete copy of the guest operating system
  + Container isolation is done on the kernel level without the need for a guest operating system
  + Libraries can be across containers
  + Can share bins and libraries
  + Infinite scalability
* What is Docker
  + Docker Engine: runtime that allows you to build and run containers
  + Docker File defines everything needed to run the image including the OS network specifications and file locations
  + Use docker file to build a docker image that gets run on the docker engine
* The Need for Orchestration Systems
  + Orchestrating containers provide an abstraction to make a cluster of machines behave like on big maching
  + Provides a way to administer a large number of containers spread across clusters of servers
  + Its filtering and scheduling system enables the selection of optimal nodes in a cluster to deploy containers
* How does Kubernetes work
  + It compares the desired state to the actual state and if they aren’t the same, it takes steps to correct it
* Kubernetes architecture and components
  + Made up of many components that do not know or care about each other
  + They talk to each other through the API server
  + The Control Plane – The Master
  + Nodes – Where pods get scheduled
  + Pods – Holds containers
* The Control Plane – The Master Node
  + The orchestrator, facilitates orchestration
  + There are multiple components in the control plane that help facilitate that orchestration
  + Etcd for storage
  + API server for communication between components
  + Scheduler decides which node pods should run
  + Controller manager responsible for checking the current state against the desired state
* Nodes
  + Make up the collective compute power of the Kubernetes cluster
  + This is where containers actually gets deployed to run
  + Physical infrastructure that your application runs on, the server of VMs in your environment
* Pods
  + Made up of one or more containers
  + When defining your cluster, limits are set for pods which define what resources, CPU, and memory, they need to run
  + Scheduler uses this definition to decide on which nodes to place the pods
* How Does Kubernetes Relate to Docker
  + Docker is a platform and tool for building, distributing, and running Docker containers
  + Kubernetes is a container orchestration system for docker containers
    - Meant to coordinate clusters of nodes at scale in production in an efficient manner
* Can you use Docker without Kubernetes
  + Docker is commonly used without Kubernetes
  + But Kubernetes offers many benefits
* Can you use Kubernetes without docker
  + It can be used with any container runtime