Kubernetes (K8s)

* Container orchestration software.
  + Kubernetes means “helmsman” in Greek
* The problem.
  + Congrats all your microservices are containerized!
  + It is a lot easier to deploy containerized applications.
  + It is still difficult to set up large enterprise application which might have a bunch of different containers.
  + You need to keep track of those containers.
    - Create new ones if the demand increases.
    - Remove and restart containers that fail.
    - Need to security firewalls so not everyone can access your containers.
  + You might have 5 or 6 computers which vary in power and have 40 different containers that need to be running and managed across theses computers.
    - You need to make sure no one computer ends up taking on more than it can handle.
    - Each computer is being used optimally.
* K8s will manage all your containers greatly “simplifying” this process.
  + Kubernetes is Hard.
* Kubernetes is declarative.
  + You give Kubernetes files saying what the state of your application should be and Kubernetes makes it happen.
  + You do not program step by step code.
    - You do not make scripts in Kubernetes.
* IP addresses inside a container do not operate like normal IP addresses.
  + Unique identifiers versus where they actually geographically are located.

Kubernetes Key Terms

* Node
  + Just a computer (Kubenetes thinks it cool to call them nodes)
* Master Node
  + The main hub computer.
  + Contains the control plane.
* Kubernetes cluster
  + A master node along with any other nodes that it has power over.
* Kubelet
  + A software program on child nodes (nodes that are controlled by the master node) that connects that peripheral node to the master node.
* Control Plane
  + A RESTful API server you can interact with.
  + Main way of working with Kubernetes.
  + You can also use the command line tool to communicate with the control plane.
* Pod
  + Atomic piece of software that you put on Kubernetes.
  + Every pod is comprised of 1 or more containers.
    - Most pods contain just 1 container.
* Deployment
  + A configuration/formation of pod(s)
    - How instances of that pod you want.
    - When to create those pods.
    - Highly configurable.
    - Kubernetes will maintain this configuration.
* Service
  + Services are how outside http calls can get to pods in the cluster.
  + Services are the main entrance to a cluster.
  + There are a few different types.
    - LoadBalancer service.
* Kubernetes can blur that line between development and operations.
  + To make deployment and maintenance much more “code” and configuration files based.
* Google has SRE (Site Reliability Engineers)
  + Software developers who are also responsible for making sure the applications deliver consistent deployment when deployed.