

NCSA Kubernetes Developer Bootstrap

- Background
- Model
- Tour of the objects
- Getting Started
- Resources

Kubernetes Strengths

- Maintaining services - automated
- Scaling container based services
- Deploying and managing microservice-oriented systems
- Recovery and resilience to hw and sw faults
 - Pull the plug, plug it back in, comes back as it was
- Portable - not tied to container impl. Storage impl. etc.
- Scalability - laptop to thousands of nodes
- Outstanding community

Difficulties

- Minimal IT core services and multi-tenancy
- K8s is a container OS - model from google DC model
- No PaaS abstractions - need to build or use others
- Many low level ropes to get hung on
 - Hacking before understanding will hurt
- Fly by-wire - no console logins*
 - Need tooling for monitoring/management
- A dizzying pace - K8s, tools, ecosystem move so fast
- Difficult to deploy in production - no common process

You should understand

- Google's data-center model and GIFFE, SRE
 - Echos of the borg - the mother of Kubernetes
- Advanced container facilities and linux process interfaces
 - Namespaces, linux capabilities, control groups, container models and implementations
- 12 factor model
- Microservices concepts and philosophy
 - More than tiny things services and REST

Core Concepts – API, Object Storage

- API – Create, remove, alter objects, query objects, watch events.
- Declarative – say what you want.
 - Reconciliation strategies makes it happen
 - Scale up, down, drain node, add node....
- Etcd under the hood stores everything
 - sequentially consistent consensus key-value store
- Pull the plug, bring it up. It resolves as it was.

Core Concepts - Execution

- Pod - 1 or more tightly integrated containers with common IPC and network namespace.
- Strictly ephemeral – a container is never restarted ever, abandoned and replaced
- Scheduler can kill containers any-time to manage cluster
- Hooks for start/stop/health/readiness
- Pod has cluster IP and cluster DNS for discovery
- pod cluster addresses are ephemeral - use discovery

Core Concepts - Network

- Everything addressable via IP has a *cluster* address
 - nodes, pods, endpoints, services, replicas ...
- Everything can talk in-cluster
 - But pods can move, scale, etc. Addresses are not stable, but discovery (DNS) is authoritative
- Services map from external into replicas
 - automatically or via an external load balancer
- No default multi-tenant network isolation (yet)
 - Use SDN: opencontrail, calico, weave

Core Concepts - ConfigMap & Secrets

- 12-factor config in environment - ConfigMap
 - Key-values from yaml, or filesystem source
- Consumable as env vars in pod spec or via mounted path - i.e. /etc/nginx.conf
- Pod with mount can respond to updates - inotifywatch
- Secrets - sensitive storage (passwords, keys, creds, ...)
 - Stored encrypted until use

Core Concepts – labels & selectors

- All objects can have arbitrary labels - key/value
 - Node, pod, service, configmaps, deployments, etc..
- API queries can select based on labels:
 - environment=production,tier=frontend
- Scheduler placement
 - Simple affinity: node-has-gpu: true
 - Topological affinity - advanced general run-time, attraction/aversion with set calculus

Core Concepts – Namespaces & Auth

- Namespace is an administrative domain
 - Think personal sub-cluster
 - API
- 1.3 now has authz, authn, RBAC and ABAC
 - Spec roles and attributes against API objects
- Services account hierarchy
 - DCO > SRE > service > user account.

Other Options and Related Systems

- RedHat OpenShift - Redhat is a huge K8s contributor
 - Kubernetes behind more-traditional IT PaaS admin, management, developer, deployment web tools
- Docker Swarm
 - Low-level container OS alternative to Kubernetes
 - “Consumer-level” ease, but docker is playing catchup
- MESOS
 - More mature DCOS
 - Many run K8s with MESOS scheduler
- Tectonic - CoreOS rkt/netes - kubernetes on rkt container runtime

Kubernaut Learnings

- Talks: Look for anything in the last year, +1yr is questionable
 - K8s Architects: Brendan Burns, Tim Hockin
 - K8s monsters: Kelsey Hightower,
 - Systems Architectures: Adrian Cockcroft, Martin Fowler, Joe Beda
 - Workshops/Conferences: KubeCon, CoreOSfest, Scale
- Groups:
 - K8s community meeting - weekly
 - K8s sigs: <https://github.com/kubernetes/community/blob/master/README.md#special-interest-groups-sig>
 - Most sigs have regular meetings
- PaaS:
 - Openshift, deis, fabric8, Kel, <https://trello.com/b/aSuCIZQo/docker-and-paas-resources>
- **Running:** Run local in a VM - **minikube**
- Getting help: kubernetes office hours, k8s slack is manned by the engineering team
 -