KubeCon | CloudNativeCon Recap

Welcome!

- Logistics: building exit is locked, so ask Cradlepoint employee to let you out at any time - we are happy to help!
- We will be recapping KubeCon | CloudNativeCon
    - Matt Messinger
    - Galo Gimenez
    - Chris Campbell
    - Matt Howell
- SWAG giveaway at end

Conference Summary

- November 2019 @ San Diego Convention Center
- Over 12,000 attendees
- Day-Zero Co-Located Events
- Three days w/ keynotes, breakout sessions, vendor area, party
- All keynotes and sessions are on YouTube!

Day Zero: Continuous Delivery Summit

- Hosted by Continuous Delivery Foundation https://cd.foundation
- CD.Foundation founded March 2019 as the open source home for Jenkins, Jenkins X, Spinnaker, and Tekton
- Goals - drive CD adoption and foster tool interoperability
- Went through several CD pipelines using different toolsets
- Key takeaways:
  - Several mentions of “Accelerate” book by Nicole, Jez, Gene (Nicole spoke at DevOpsDays)
  - A lot of focus on building security and compliance into pipeline - it’s more than just deploying code. Example https://kubesec.io (https://kube-score.com/)
  - Canary deployments
  - Rise in GitOps - ArgoCD and Weave Flux

My favorite sessions

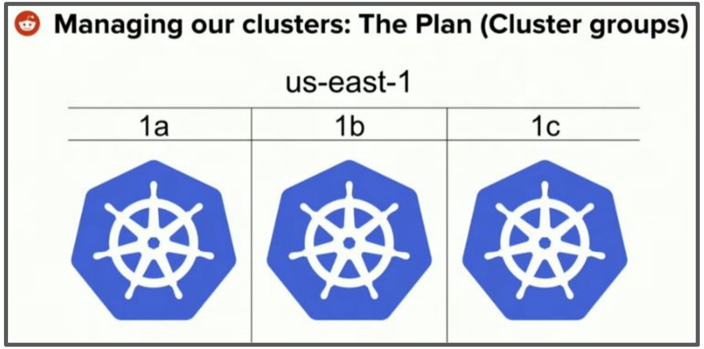
- 10 Weird Ways to Blow Up Your Kubernetes (Airbnb)
- Kubernetes at Reddit: Tales from Production
- The Gotchas of Zero-Downtime Traffic w/ Kubernetes (Weaveworks)

10 Weird Ways to Blow Up Your Kubernetes (Airbnb)

- Learning and pitfalls using Kube at Airbnb
- Don’t maintain your own fork of Kubernetes
- maxSurge: 100% example with Service Mesh (SmartStack) cannot keep up due to slow haproxy startup time
    - Lesson: Kube deploys can cycle pods super fast, whether the rest of your infra can keep up or not
    - Cradlepoint example: kube2iam
- Monster daemonsets - use daemonset to download gigs of translation data to node
    - What happens when daemonset dies?
    - 2000 pods for daemonset with 2000 nodes; cluster cannot hold that many pods, they are OOM killed, fills up etcd with events, brings down cluster
    - Move to sidecar; use admission controller to prevent daemonsets
- Make 2 million docker images in AWS ECR; lifecycle policy doesn't know about images you are using
    - Wrote CronJob to check images being used in cluster -- until you have more than one cluster
    - Fix by running “ECR Cleaner” in its own “management” cluster that has access to other clusters

Kubernetes at Reddit: Tales from Production

- Retrospective on move to Kubernetes (continued from last year)
- 2016 - monolith, 20 devs; 2019 - microservices and over 250 devs
- 2019 - now about 45 services on kube on prod; new service every 2 weeks
  - Most engineering teams have adopted Kube
  - Embedded SREs in service owning teams
  - Support load is manageable for devops team
  - More work to be done on launch automation, docs, tooling, training
- Cluster management
  - Started before EKS -- run a full cluster in each AZ
  - Each self contained, minimizes traffic between AZs
  - Use Spinnaker, Terraform, Helm
  - Currently operate 19 clusters
- Plan to use Open Policy Agent for guard rails
  - Example: policy to prevent open ELB services



Managing our clusters: The Plan (Cluster groups)

Gotchas of Zero-Downtime Traffic (Weaveworks)

- Pod shutdown
  - Pod marked as terminating, service controller removes endpoint
  - PreStop runs PID 1 SIGTERM, wait terminationGracePeriodSeconds, PID SIGKILL
  - Shells, like sh and bash do not pass signals! Do not use CMD, use ENTRYPOINT
- Probes
  - Readiness probe very important when service web traffic (pod startup)
  - Liveness probe is dangerous - app under load killed could could cause cascading failures
  - Liveness should be way longer than readiness
  - Be intentional with timeouts and periods
- PreStop lifecycle hook - start draining connections (sometimes just sleep)
- Demo https://github.com/stealthybox/zero-downtime
  - using kind - a kube cluster in docker https://github.com/kubernetes-sigs/kind
  - Siege - https://github.com/sudermanjr/siege-kube
- Deployment maxUnavailable, strategy.minReadySeconds, strategy.maxSurge