Running containers at scale. From Nomad to Kubernetes and lesson learned

Viet Huynh 16 Jan 2018

Agenda

1. Chotot k8s system overview
2. Kube-system
3. Platform observability
4. Deployment
5. Application
6. Tips & tricks
7. Team alignment

0. Nomad

- Back to history
- Why Nomad?
- Nomad issues
- Why Kubernetes?

1. Chotot k8s system overview

- 3 etcd, 2 masters + 1 HA, 13 workers (physical)
- 100+ services, jobs
- 100% backend microservices
- 80% web
- cronJob, internal dashboard, 3rd party tooling, ephemeral caching

2. Kube-system

- Kube-api
- Kube-dns
- Networking
- Etcd
- Resource quotas

3. Platform observability

- Node problem detector
- Efficient logging
- Efficient metrics collecting
- Performance test

3.1 Node problem detector

Why need it?

- Had a couple of kernel panic, no reason, mostly assumed server’s under stress.
- Need to know before it happens.

3.2 Efficient logging

- Tailable
- Local log
- Obstructive middleware

3.3 Efficient metrics collecting

- K8s metrics
- Application metrics
- Customization
- Long-term storage

3.4 Performance test

- Early and frequently
- Cloud networking perf test is a beast
- PerfKitBenchmarker
- In-house “benchmarks” tool kit

4. Deployment

- Blue/green deployment strategy
- Helm common template
- Containers security practice (Todo)

5. Application

- Java workload on containers.
- Go
  + Ram is abundant due to optimized Go service.
  + Memory fragmentation could be a problem.
- NodeJS
  + Web server needs whole lots CPU on initiative and minimal on run.
  + Skewed cpu resources limit to avoid crash loop.

6. Kubectl tips and tricks

 # get all pods sort by name

kubectl get po -o jsonpath='{range.items[*]}{.metadata.name}{"\n"}{end}'

 # Get all pods, which as restart_count > 0

kubectl get po -o jsonpath='{range.items[?(@.status.containerStatuses[0].restartCount>0)]}{.status.containerStatuses[0].name}{"\n"}{end}'

 # Get all non-running pods

kubectl get po -o jsonpath='{range.items[?(@.status.phase != "Running")]}{.metadata.name}{"\n"}{end}'

 # Get most used cpu pods

kubectl top pods | tail -n +2 | sort -nr -k 2 | awk '{print $1}' | head -n 1

 # Get all nodes, and their IP

kubectl get no -o jsonpath='{range.items[*]}{.metadata.name}{"\t"}{.status.addresses[?(@.type=="InternalIP")].address}{"\n"}{end}'

7. Technical alignment

- Devops culture, train your software engineer k8s mindset and tooling so you could focus on improving the platform.

THANKS!

Any questions?

Looking for teammates to build next generation cloud native platform, like Hyperconverged infrastructure, Chaos Engineering.

Chat to us at skype: tuan-viet.huynh or email to viethuynh@chotot.vn