K8S On-premise: Incident & Lesson Learned ZaloPay Merchant Platform (MEP)

Châu Nguyễn Nhật Thanh
Head of MEP - VNG Corp.

VNG Keep Dreaming 15

Contents

- About me
- Why k8s on premise?
- MEP K8S stack
- Issues
- Lesson learned
- Next step

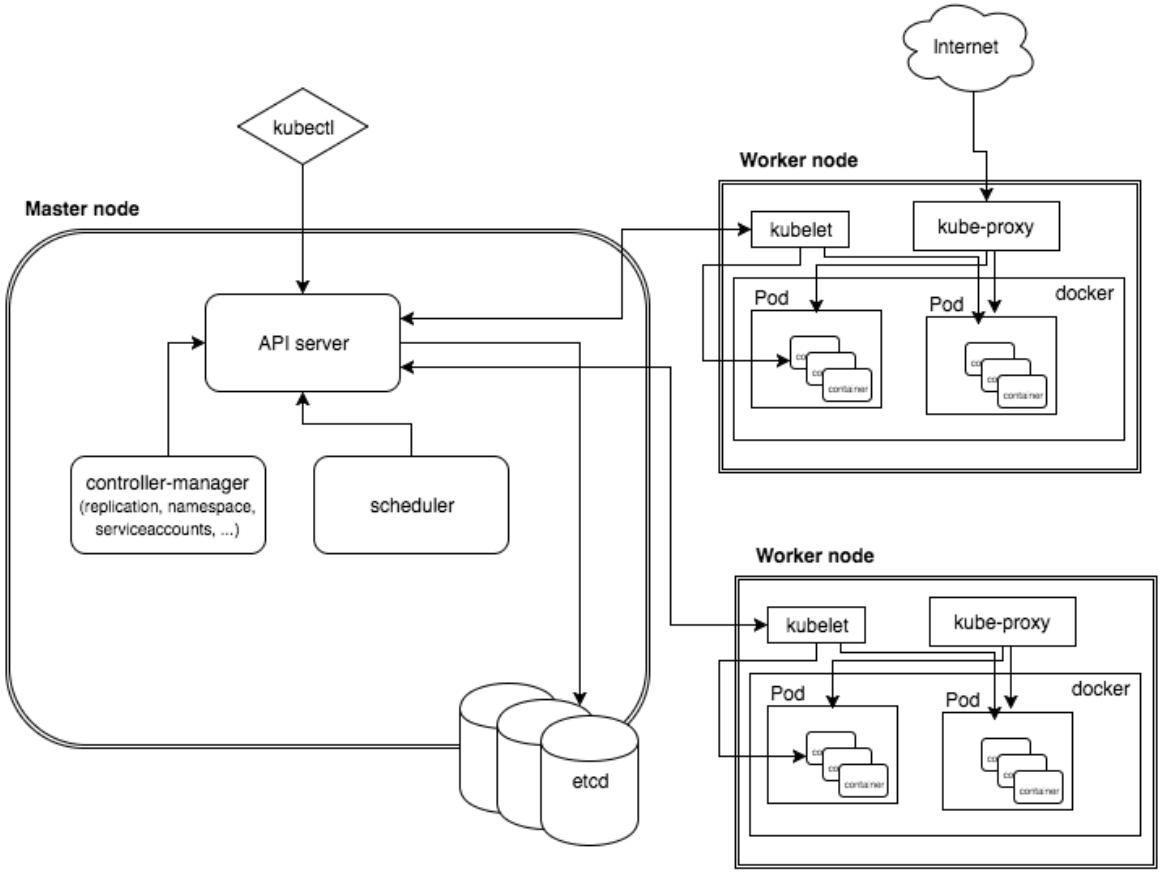
Me

- M.Sc Uni Duisburg, Germany
- Tech Lead:
    - ZingMe
    - CSMBoot, CSMPlay, CSM
    - GBC
    - IoT Lab
    - ZaloPay Merchant Platform (MEP)

- K8S newbie

Why K8S

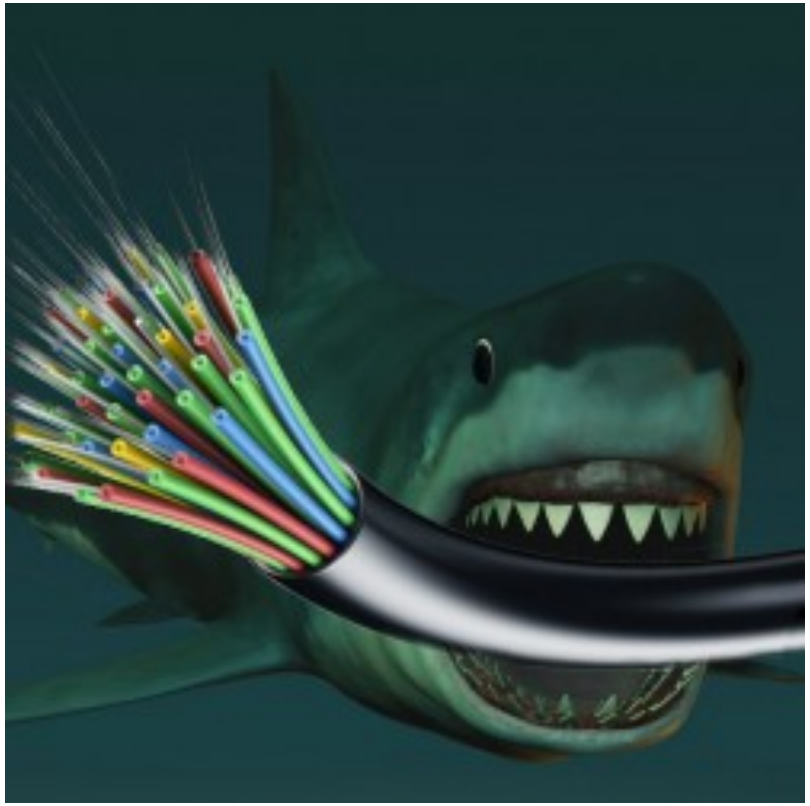
- Trend: micro service, container
- Orchestrating across hosts
- Easy to manage and scale app
- Save cost !!!



Source: https://x-team.com/blog/introduction-kubernetes-architecture

Why on premise

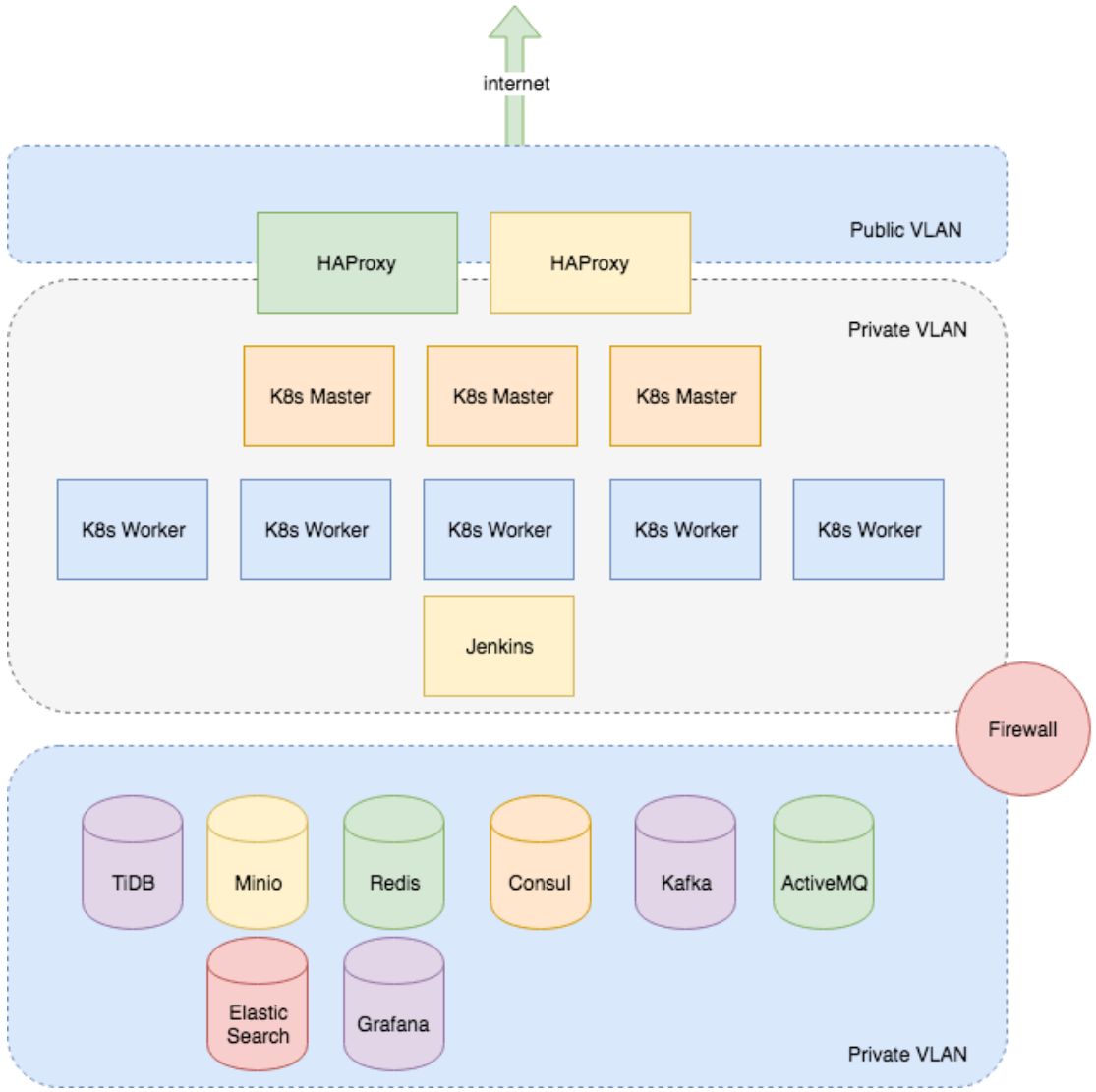
- Fintech -> secure data
- Save cost !!!

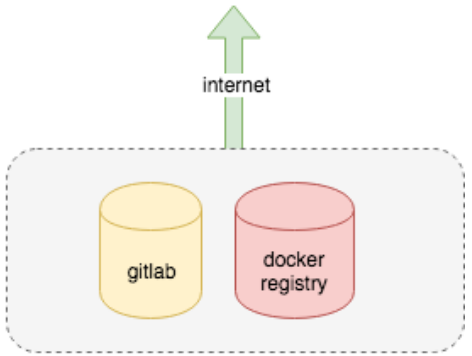


MEP K8S Stack

- Deploy Architect
  - Load balancing
  - Access internet
  - Storage
    - DB
    - File
- CI/CD
- Log Collect
- Tracing
- Monitor & Alert

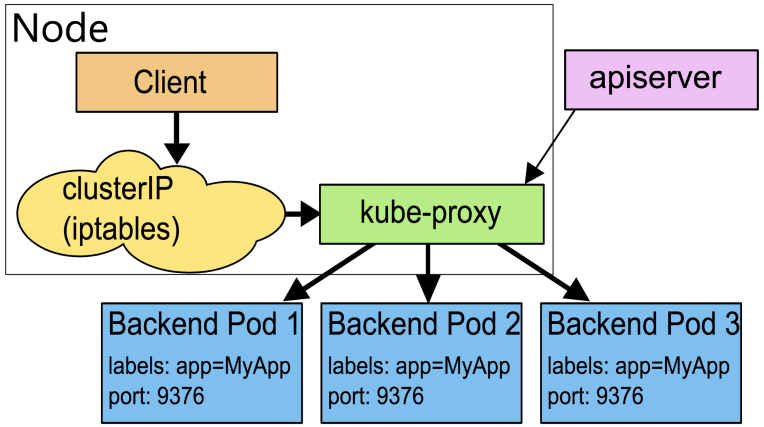
MEP K8S Stack

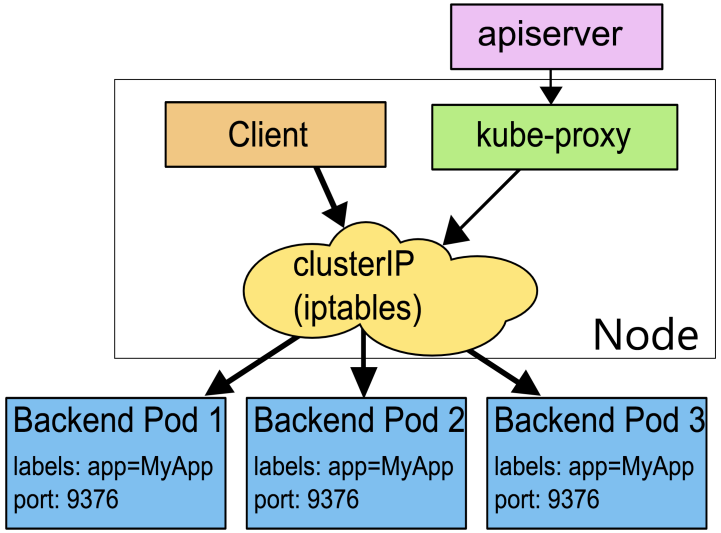


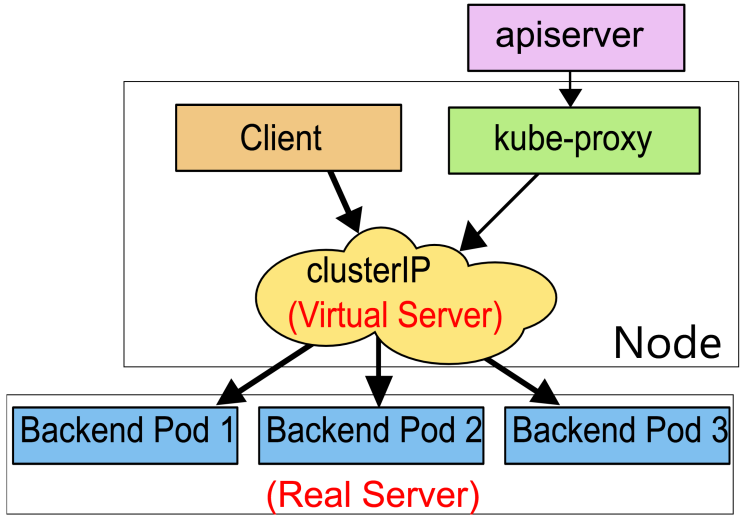


Load balancing

- Why do we need proxy ?
- Proxy mode
  - User space proxy mode (from v1.0)
  - IPTables proxy mode (from v1.1)
  - IPVS proxy mode (from v1.2)







Load balancing

- Service type:
  - ClusterIP
  - NodePort
  - LoadBalancer => our choice
    - Using MetalLB layer 2 (ARP)

Access Internet

- Using HTTP(S) Proxy installed in HAProxy node
- In K8s

To work around that problem I would suggest to create ConfigMap with that proxy values e.g.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: your-config-map-name
  labels:
    app: your-best-app
data:
  HTTPS_PROXY: http://ssnproxy.ssn.xxx.com:80/
  HTTP_PROXY: http://ssnproxy.ssn.xxx.com:80/
```

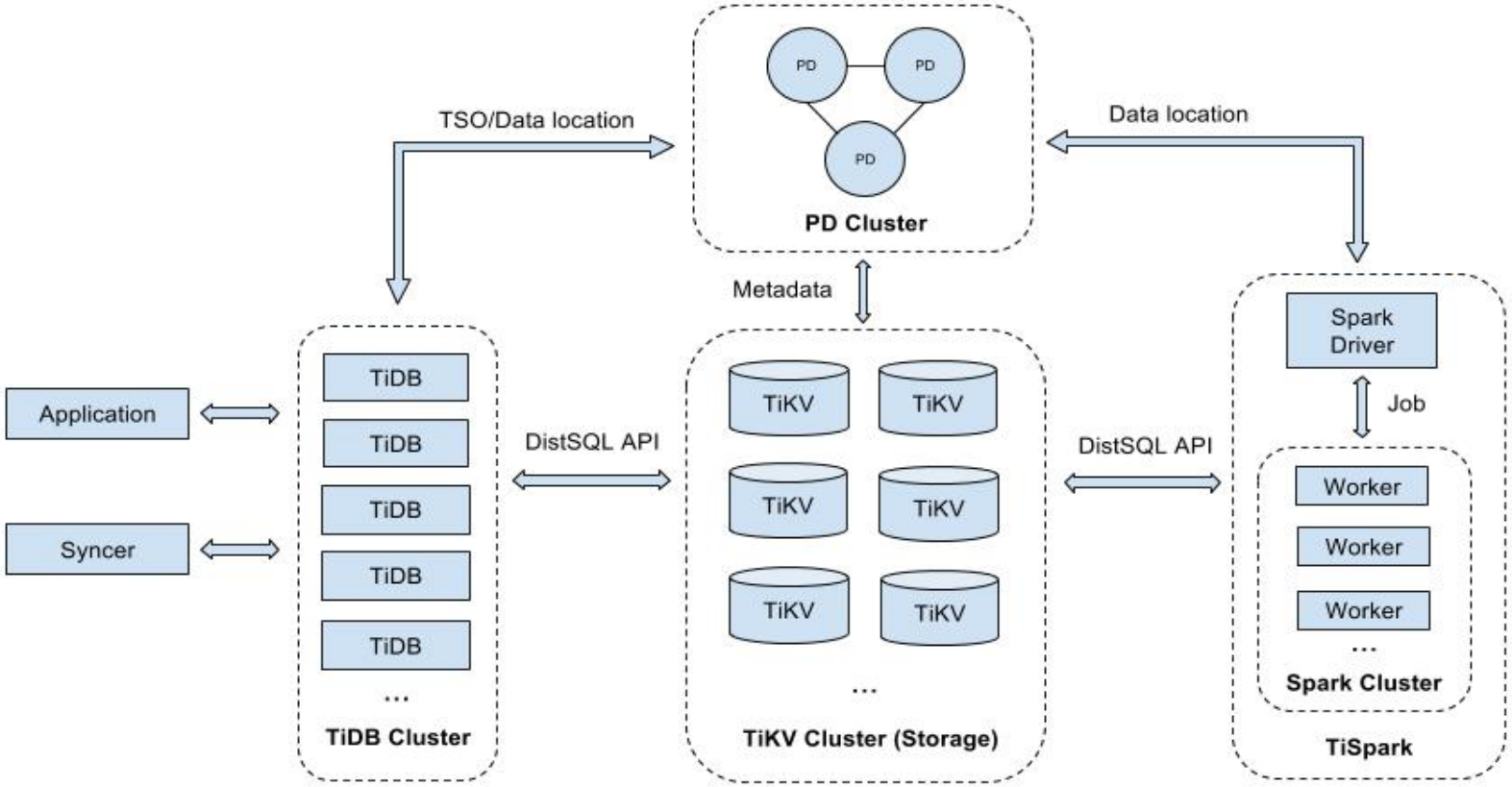
and mount them to deployment as environment variables using

```
envFrom:
  - configMapRef:
      name: your-config-map-name
```

Storage

- DB
  - Separate VLAN
  - SQL: TiDB
  - NoSQL: Redis, Cassandra
  - Queueing: Kaffka, ActiveMQ
  - Search: Elastic Search
- File
  - Minio

Storage

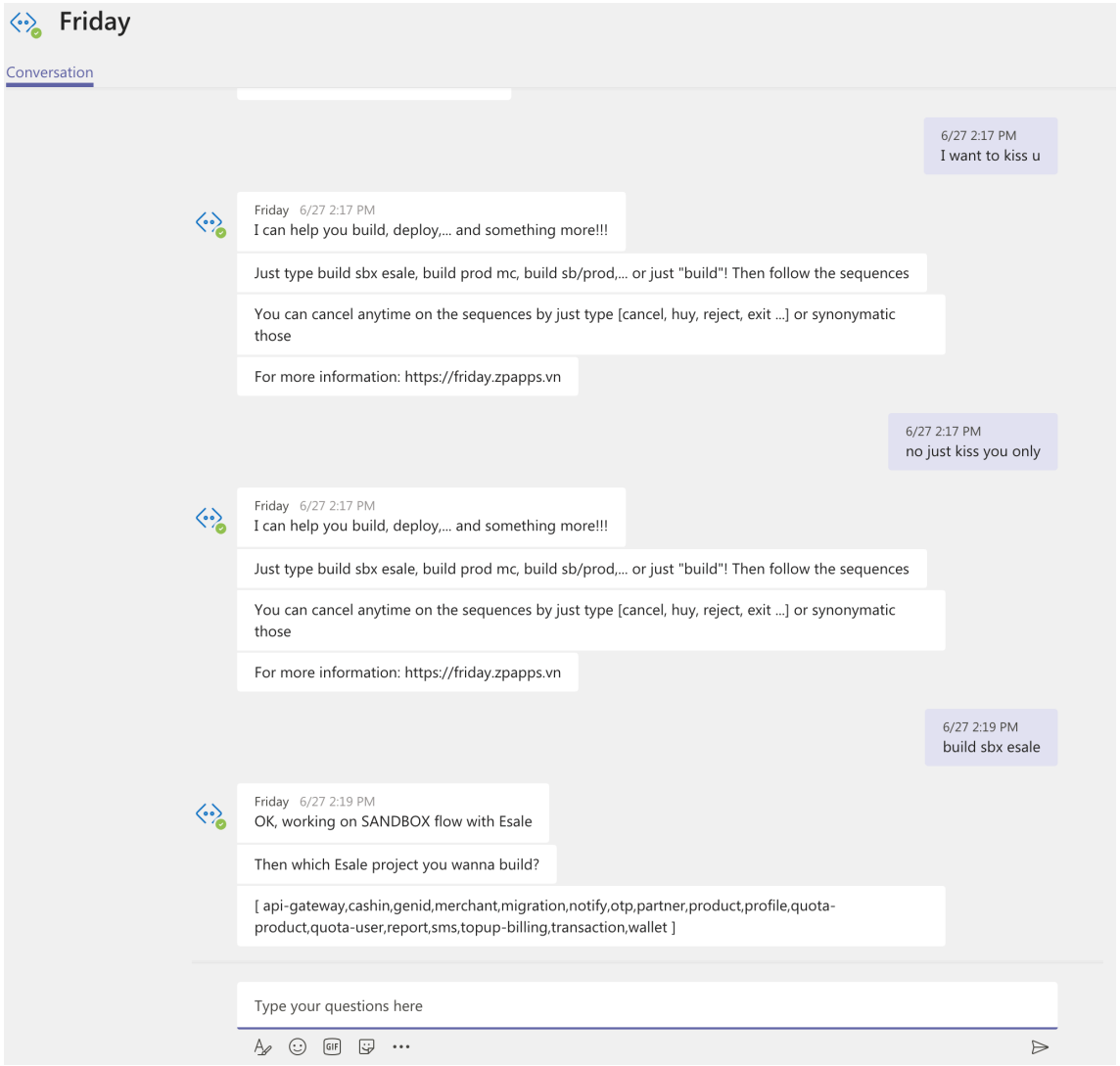


https://upload.wikimedia.org/wikipedia/commons/1/1f/TiDB_Architecture.jpg

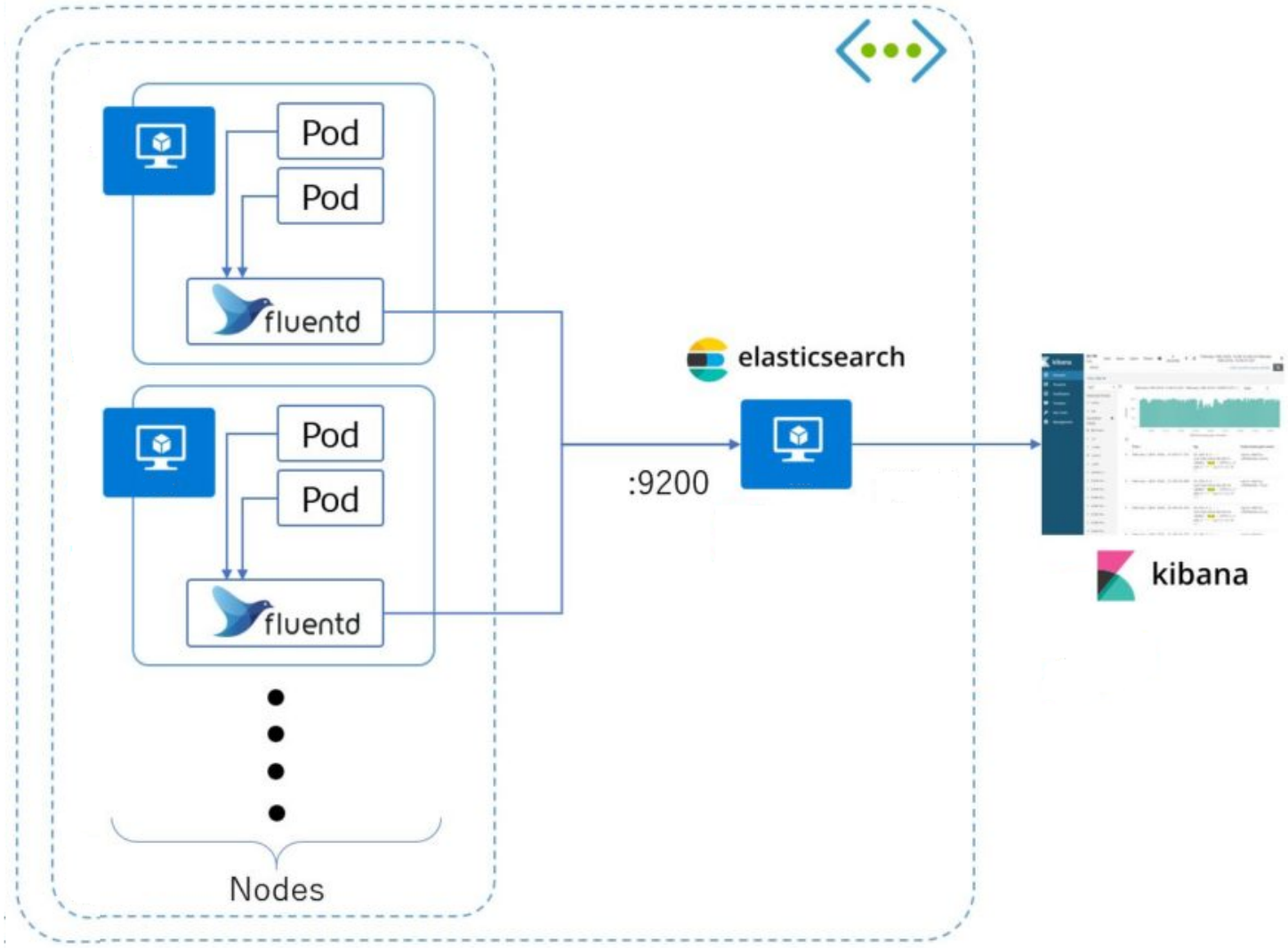
CI/CD v1

- Gitlab hook when commit with comment “BUILD <ENV>”
  - Why don’t we use GitLab Webhook?
- Call Jenkins Pipeline
  - Build the code
  - Test the code
  - Deploy to K8s
- Manual config HAProxy

CI/CD v2


Friday
Conversation
6/27 2:17 PM
I want to kiss u
Friday 6/27 2:17 PM
I can help you build, deploy,... and something more!!!
Just type build sbx esale, build prod mc, build sb/prod,... or just "build"! Then follow the sequences
You can cancel anytime on the sequences by just type [cancel, huy, reject, exit ...] or synonymatic those
For more information: https://friday.zpapps.vn
6/27 2:17 PM
no just kiss you only
Friday 6/27 2:17 PM
I can help you build, deploy,... and something more!!!
Just type build sbx esale, build prod mc, build sb/prod,... or just "build"! Then follow the sequences
You can cancel anytime on the sequences by just type [cancel, huy, reject, exit ...] or synonymatic those
For more information: https://friday.zpapps.vn
6/27 2:19 PM
build sbx esale
Friday 6/27 2:19 PM
OK, working on SANDBOX flow with Esale
Then which Esale project you wanna build?
[api-gateway,cashin,genid,merchant,migration,notify,otp,partner,product,profile,quota-product,quota-user,report,sms,topup-billing,transaction,wallet]
Type your questions here

Log collector

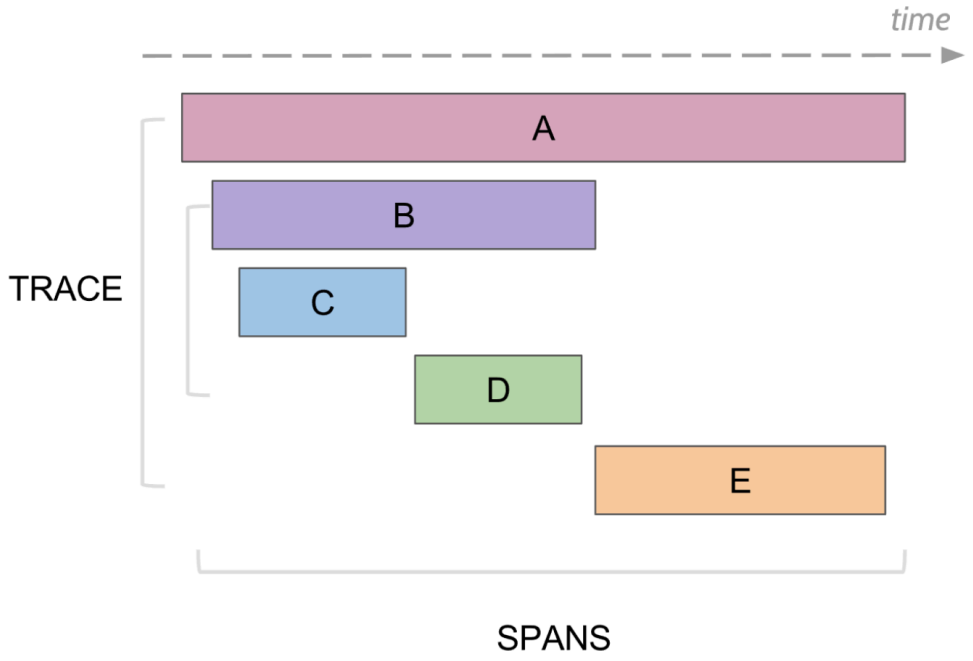


https://medium.com/@carlosedp/log-aggregation-with-elasticsearch-fluentd-and-kibana-stack-on-arm64-kubernetes-cluster-516fb64025f9

Tracing

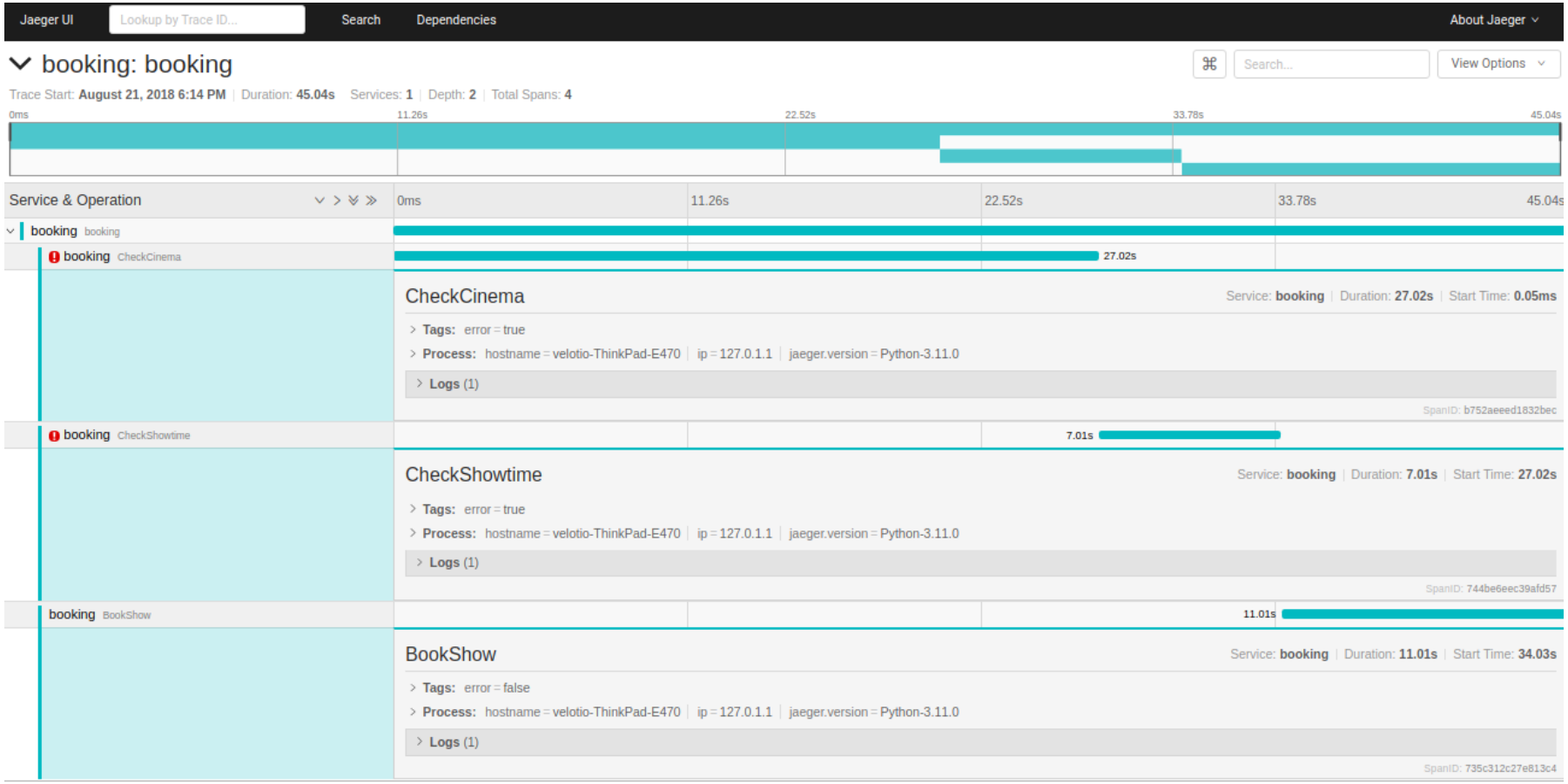
Span — It represents a logical unit of work that has an operation name, the start time of the operation, and the duration.

Trace — A Trace tells the story of a transaction or workflow as it propagates through a distributed system. It is simply a set of spans sharing a *TraceID*. Each component in a distributed system contributes its own span.

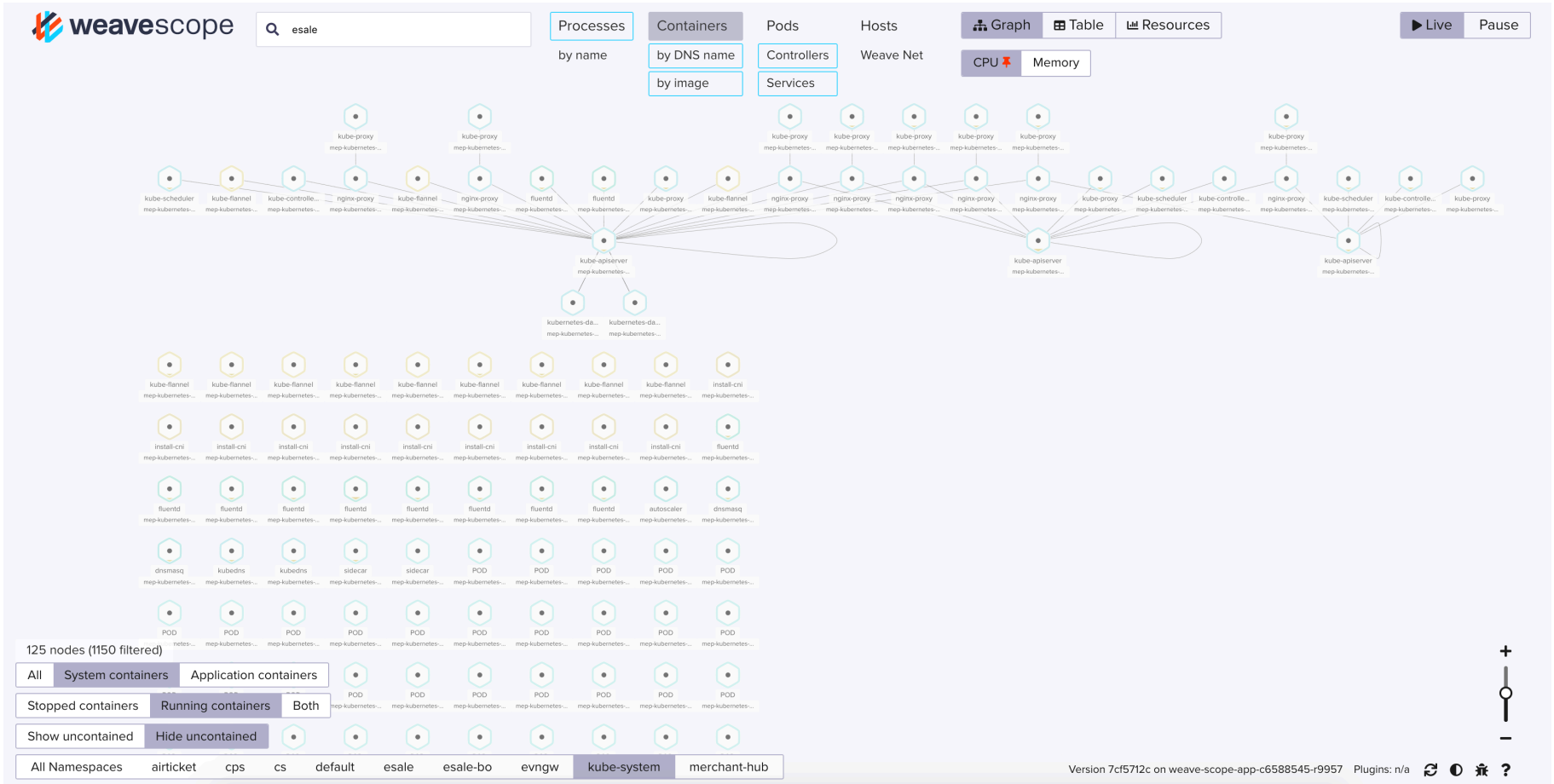


https://medium.com/velotio-perspectives/a-comprehensive-tutorial-to-implementing-opentracing-with-jaeger-a01752e1a8ce

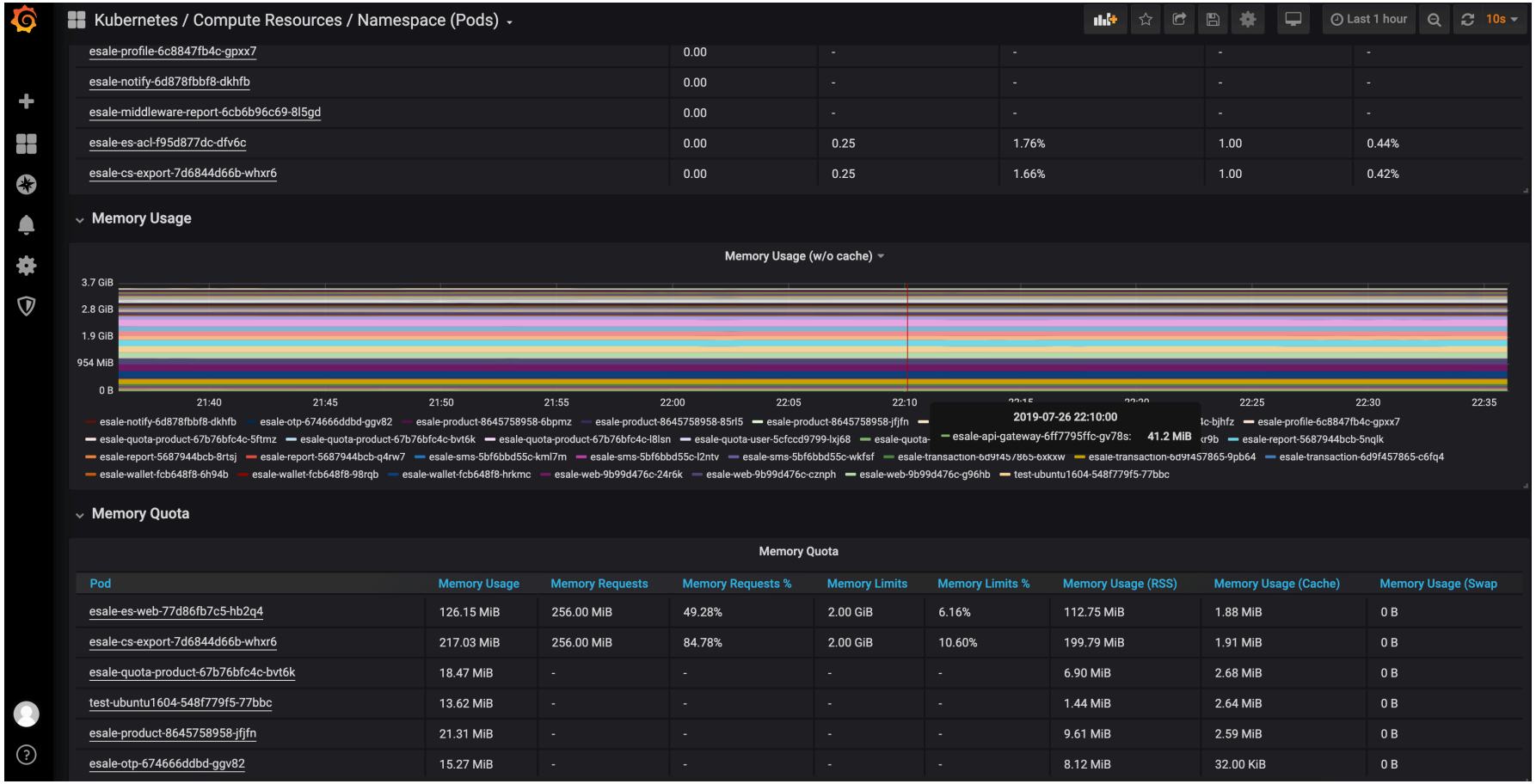
Tracing


Jaeger UI
Lookup by Trace ID...
Search
Dependencies
About Jaeger
booking: booking
Search...
View Options
Trace Start: August 21, 2018 6:14 PM | Duration: 45.04s | Services: 1 | Depth: 2 | Total Spans: 4
0ms
11.26s
22.52s
33.78s
45.04s
Service & Operation
booking booking
booking CheckCinema
27.02s
CheckCinema
Service: booking | Duration: 27.02s | Start Time: 0.05ms
Tags: error = true
Process: hostname = velotio-ThinkPad-E470 | ip = 127.0.1.1 | jaeger.version = Python-3.11.0
Logs (1)
SpanID: b752aeeed1832bec
booking CheckShowtime
7.01s
CheckShowtime
Service: booking | Duration: 7.01s | Start Time: 27.02s
Tags: error = true
Process: hostname = velotio-ThinkPad-E470 | ip = 127.0.1.1 | jaeger.version = Python-3.11.0
Logs (1)
SpanID: 744be6eec39afd57
booking BookShow
11.01s
BookShow
Service: booking | Duration: 11.01s | Start Time: 34.03s
Tags: error = false
Process: hostname = velotio-ThinkPad-E470 | ip = 127.0.1.1 | jaeger.version = Python-3.11.0
Logs (1)
SpanID: 735c312c27e813c4

Monitor & Alert



Monitor & Alert



Monitor & Alert

Monitor & Alert

- Monitor dashboard from K8S
- Monitor HAProxy log
  - Parse ERR 5XX
  - Alert by SMS to tech leader

Issues 1

- Scale node break the production farm
  - Multi interface because of sticky node
  - Kubespray choose default route when no IP in inventory file

```
mep@template-ubuntu1604:~$ cat kubespray/inventory/dev-cluster/inventory.ini
# ## Configure 'ip' variable to bind kubernetes services on a
# ## different ip than the default iface
# ## We should set etcd_member_name for etcd cluster. The node that is not a etcd member do not need to set the value, or can set t
[all]
 dev-master-95 ansible_host=10.205.21.95   ip=10.205.21.95 etcd_member_name=etcd1
 dev-master-96 ansible_host=10.205.21.96   ip=10.205.21.96 etcd_member_name=etcd2
 dev-master-97 ansible_host=10.205.21.97   ip=10.205.21.97 etcd_member_name=etcd3
 dev-node-98 ansible_host=10.205.21.98   ip=10.205.21.98
 dev-node-99 ansible_host=10.205.21.99   ip=10.205.21.99
 dev-node-100 ansible_host=10.205.21.100   ip=10.205.21.100
# ## configure a bastion host if your nodes are not directly reachable
# bastion ansible_host=x.x.x.x ansible_user=some_user

[kube-master]
dev-master-95
dev-master-96
dev-master-97

[etcd]
dev-master-95
dev-master-96
dev-master-97

[kube-node]
dev-node-98
dev-node-99
dev-node-100
# node2
# node3
# node4
# node5
# node6

[k8s-cluster:children]
kube-master
kube-node
```

Issues 2

- Cannot join node which joined before
  - Kubeadm installed
  - Kubelet cannot start
- How to fix ?

Issues 3

- 2 node die
  - Product has problem
  - Biz pressure: request to shutdown product because of tech incidents
  - Try to fix => more problem
- How to fix ?

Lesson learned

- Try to make DEV ~ PROD
- Try to understand the root causes
- Practice & Practice & Practice
- Chaos engineering is VERY IMPORTANT for production
- Need supporting from Biz Owner to apply new technology

Next steps

- Upgrade to latest version k8s, tidb
- Consolidate monitor tools
- Make Alert system smarter
- Apply Ingress controller : Nginx Ingress Controller
- Try Persistence Volume: OpenEBS
- Redis cluster solution
- Fully automation CI/CD

https://twitter.com/danielepolencic/status/1172961505144377350