OF@TIEN+ IoTCloud

Toward Large-Scale Microservice Orchestration with Kubernetes Technology in IoTcloudServe@TEIN Testbed Infrastructure

Krerk Piromsopa, Ph.D.
Computer Engineering
Chulalongkorn University

IoT
cloudserve
@TEIN

SMART MOBILITY
SMART ENERGY
SMART AGRICULTURE

Facebook Page

https://www.facebook.com/iotcloudserve/

Github

https://github.com/IoTcloudServe

IoTCloud

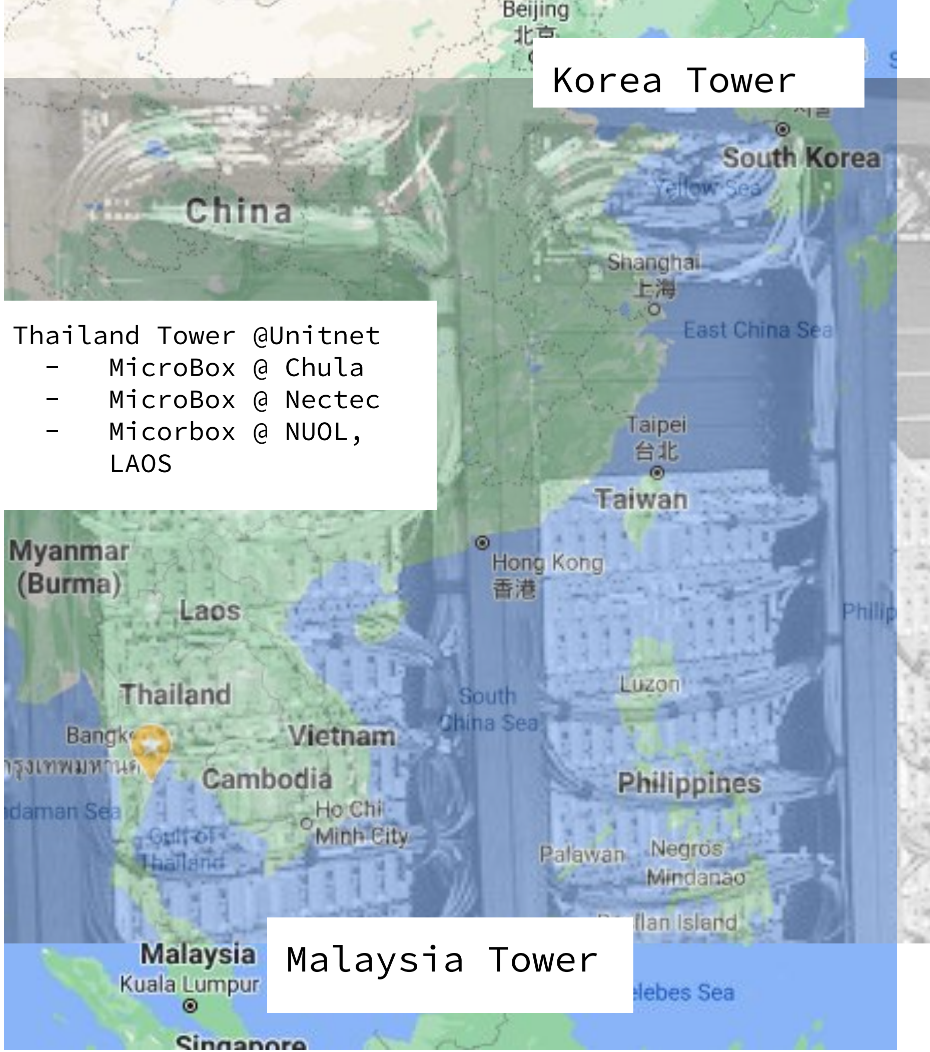
- ★ Infrastructure & Applications
- ★ Large-Scale Microservice Orchestration (Kubernetes Technology)
- ★ Research & Directions
  - ○ Monitoring/Logging/Visibility
  - ○ Control/Policy (Quota)
  - ○ Storage Management
  - ○ Federation Concepts
- ★ Summary

Infrastructure

★ Tower & Storage

★ Microboxes for developments and edge application

★ ...



Korea Tower

Thailand Tower @Unitnet
- MicroBox @ Chula
- MicroBox @ Nectec
- Micorbox @ NUOL, LAOS

Malaysia Tower

Several Opportunities for Research and Developments

- Management
  - Microservices Orchestration
  - Quota/Control Policy
  - Storage Management
  - Federation
  - Edge & Cloud
- Applications
  - Smart Agriculture/Farm
  - Smart Energy
  - Smart Mobility
  - … Smart X

Choices of Software/System Management

★ Application – Docker/Container

★ Orchestration – Kubernetes

★ Interface/Control – Rancher

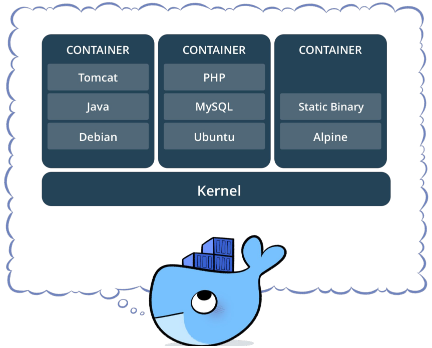
★ Storage – ROOK ceph

★ Monitor – Grafana + Prometheus

★ Logging – EFK

★ Service Mesh – ISTIO

★ Edge/Small Cluster – K3S



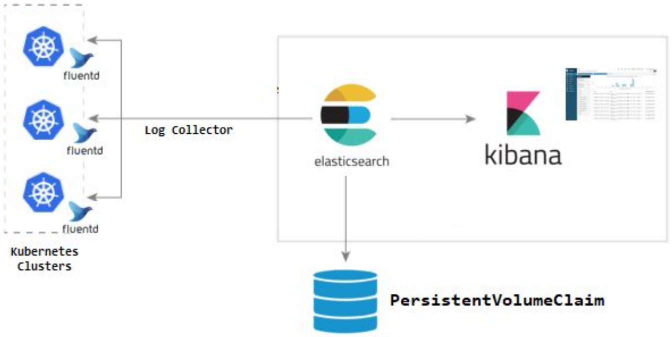
CONTAINER	CONTAINER	CONTAINER
Tomcat	PHP	
Java	MySQL	Static Binary
Debian	Ubuntu	Alpine
Kernel		

More details in the following session.

Monitor/Logging/Visualization

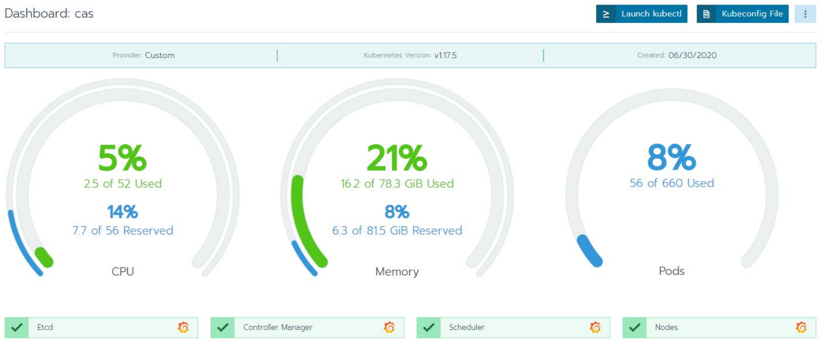
Logs

- ★ Collector - Fluentd
- ★ Storage - Elasticsearch
- ★ Visualization - Kibana



Metrics

- ★ Collector - Prometheus
- ★ Visualization - Graphana
- ★ (built-in with rancher)



Dashboard: cas

Provider: Custom | Kubernetes Version: v1.17.5 | Created: 06/30/2020

5%
2.5 of 52 Used
14%
7.7 of 56 Reserved
CPU

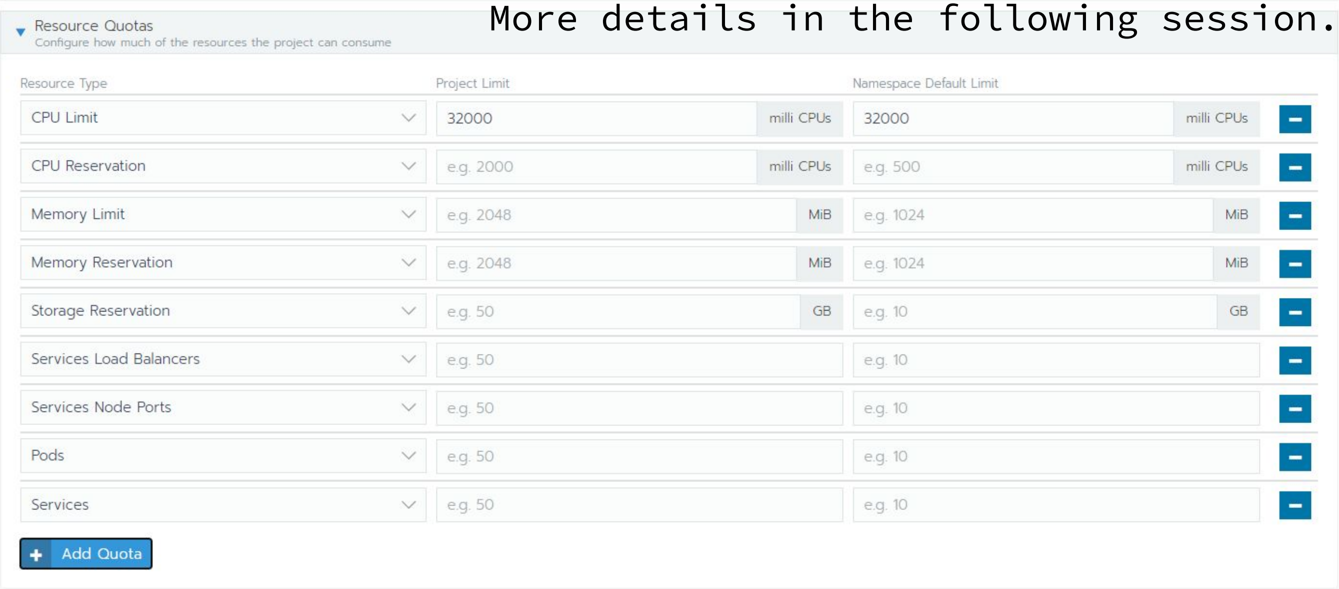
21%
16.2 of 78.3 GiB Used
8%
6.3 of 81.5 GiB Reserved
Memory

8%
56 of 660 Used
Pods

✓ Etcd
✓ Controller Manager
✓ Scheduler
✓ Nodes

Control/Policy

More details in the following session.



▼ Resource Quotas
Configure how much of the resources the project can consume

Resource Type	Project Limit	Namespace Default Limit
CPU Limit	32000 milli CPUs	32000 milli CPUs
CPU Reservation	e.g. 2000 milli CPUs	e.g. 500 milli CPUs
Memory Limit	e.g. 2048 MiB	e.g. 1024 MiB
Memory Reservation	e.g. 2048 MiB	e.g. 1024 MiB
Storage Reservation	e.g. 50 GB	e.g. 10 GB
Services Load Balancers	e.g. 50	e.g. 10
Services Node Ports	e.g. 50	e.g. 10
Pods	e.g. 50	e.g. 10
Services	e.g. 50	e.g. 10

- ★ Standard Kubernetes cluster comes name spaces and quota (via admission plugins)
- ★ Rancher introduces projects and quotas for both project and namespaces.
- ★ A namespace is assigned to a project.

Storage Management

- Ceph supports Shared Files (CephFS), Block (RBD), and Objects (S3).
- Others
  - Longhorn (block only)
  - Minio (Object only)
  - NFS (file only) - not self manage

Microservice Orchestration

Microservice Orchestration in a cluster

- ★ Ingress (nginx), mapping different paths to different services
- ★ Use Horizontal Pod Autoscaler for each service.
- ★ Use Database and/or CephFS for sharing.

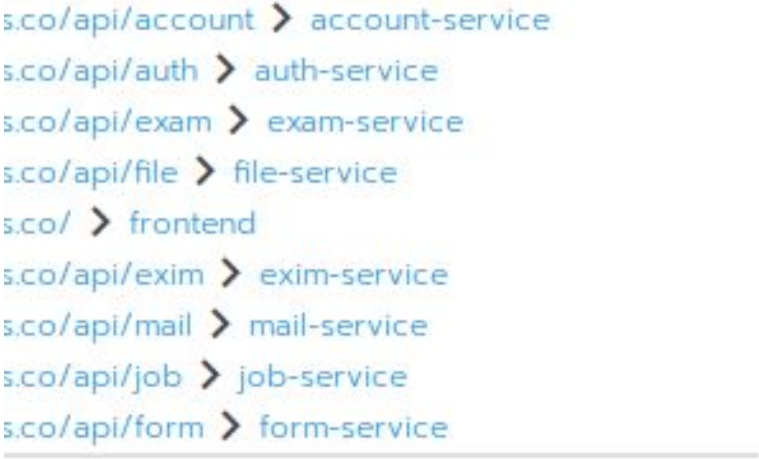
This is practical for small number of services in a cluster.

Microservice Orchestration in a cluster (ctd.)

★ Mapping path - share cookies/sessions

★ Each service can scale independently.

★ Limited to a cluster.

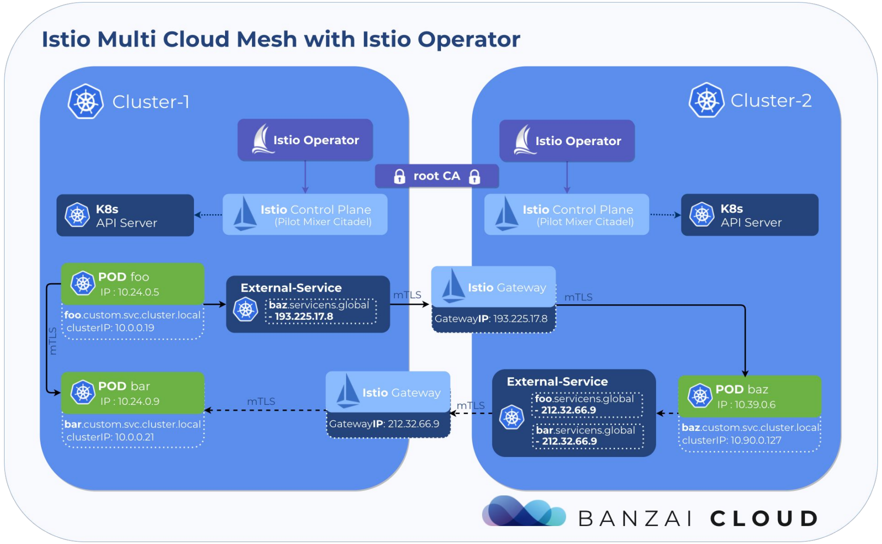


s.co/api/account	>	account-service
s.co/api/auth	>	auth-service
s.co/api/exam	>	exam-service
s.co/api/file	>	file-service
s.co/	>	frontend
s.co/api/exim	>	exim-service
s.co/api/mail	>	mail-service
s.co/api/job	>	job-service
s.co/api/form	>	form-service

More details in the following session.

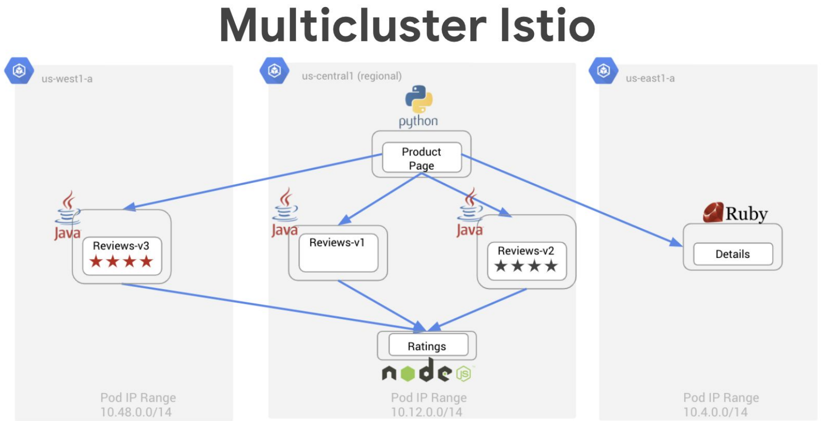
Federation

- ★ Each kubernetes cluster is operated separately.
- ★ Use ISTIO gateway to mesh cross-cluster applications.



Istio Multi Cloud Mesh with Istio Operator

Cluster-1
Istio Operator
root CA
K8s API Server
Istio Control Plane (Pilot Mixer Citadel)
POD foo IP : 10.24.0.5
foo.custom.svc.cluster.local clusterIP: 10.0.0.19
External-Service
baz.servicens.global - 193.225.17.8
mTLS
POD bar IP : 10.24.0.9
bar.custom.svc.cluster.local clusterIP: 10.0.0.21
mTLS
Istio Gateway
GatewayIP: 212.32.66.9
Cluster-2
Istio Operator
Istio Control Plane (Pilot Mixer Citadel)
K8s API Server
Istio Gateway
GatewayIP: 193.225.17.8
mTLS
External-Service
foo.servicens.global - 212.32.66.9
bar.servicens.global - 212.32.66.9
mTLS
POD baz IP : 10.39.0.6
baz.custom.svc.cluster.local clusterIP: 10.90.0.127
BANZAI CLOUD



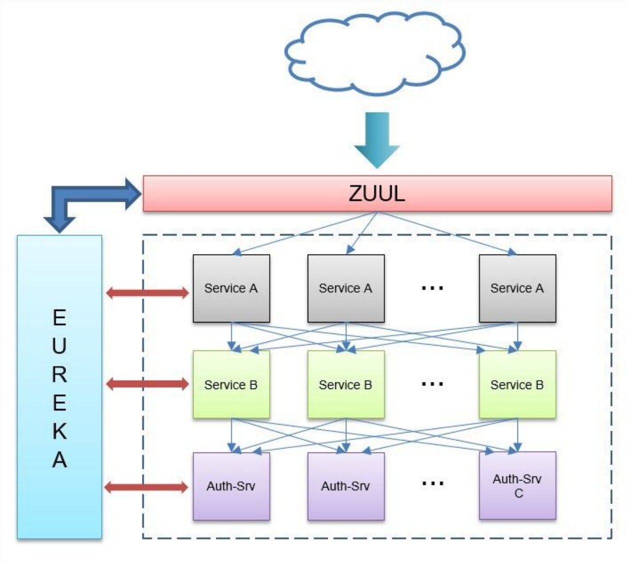
Multicluster Istio

Federation (ctd.)

- ISTIO mesh solves the issue at network layer.
- Application can still be accessed from the same URL without (major) software modification.

Alternative solution(s)

- ★ Netflix ZUUL + Eureka
- ★ Dynamic scale regardless of cluster/cloud providers
- ★ Require software modification



- ★ Apigee (from Google), Kong (opensource), etc...

Apigee API Management
CONNECTED EXPERIENCES
Partner Apps
Consumer Apps
Employee Apps
Cloud Apps
Systems of Record
Internet of Things
BACKEND SYSTEMS
70
Security · Analytics · Operations · Runtime
Monetization · Mediation · Developer Portal · Monitoring

Picture from https://medium.com/@enricobergamo/building-scalable-container-ready-and-secure-microservices-using-spring-boot-netflix-oss-and-78b95e73ed00

We are now capable
of managing
kubernetes clusters.

Best Practices for Cluster Management

★ HA Rancher installation with 2 x K3S nodes + 1 mysql

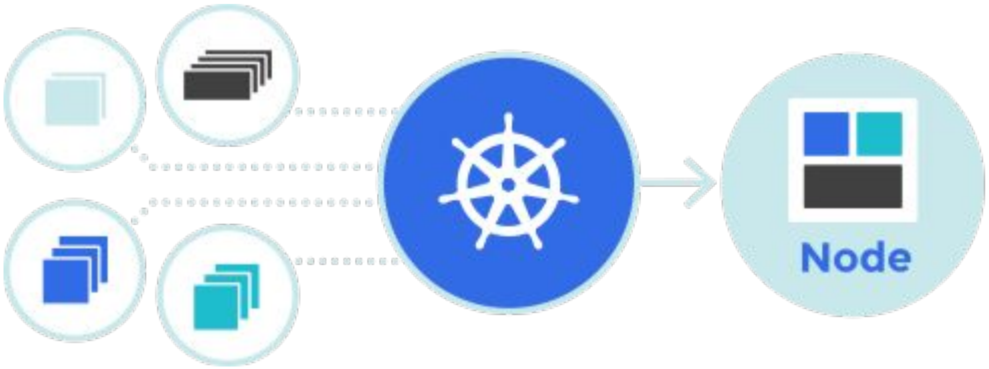
  ○ Use haproxy as a load balancer for K3S (rancher)

★ Provision K8S from rancher

★ Or Add existing cluster to rancher

RANCHER

K3S



Best Practices for Cluster Management (ctd.)

- ★ Databases
  - ○ MariaDB (Galera master-master) + haproxy
  - ○ MongoDB (replica set + sharding)
- ★ Storage
  - ○ Rook CEPH
    (CephFS for sharing, RBD for high performance, Object for S3)
- ★ Scaling
  - ○ Small Applications - ingress + HPA
  - ○ Large Applications - ingress + HPA + ISTIO
    Multi-cluster Applications - ingress + HPA (+ ZUUL + eureka)

This is just the beginning.

Ideas

★ Cloud-native storage with support for edge/off-line services (eg. IoT application, ad-hoc services)

★ Managing edge node with kubernetes (KubeEdge?)

★ more

Thank you

Q&A