9RUM
kakao Cloud Native Platform

공용준(Andrew.kong)
kakao corp.(cloud part)
Who am I
Andrew. Yongjoon kong

• Cloud Technical Advisory for Government Broad Cast Agency
• Adjunct Prof. Ajou Univ
• Korea Data Base Agency Acting Professor for Bigdata
• Member of National Information Agency Bigdata Advisory committee
• Kakaocorp, Cloud Part Lead
• Talks
  • Scalable Loadbalancer with VM orchestrator (2017, netdev, korea)
  • Embrace clouds (2017, openstack days, korea)
  • Full route based network with linux (2016, netdev, Tokyo)
  • SDN without SDN (2015, openstack, Vancouver)
First of All

• What is Cloud?

What the hell is the cloud computing?

- Oracle CEO Larry Harrison
if (kakao) : Cloud.define()

Programmable Resource Management

2016
What is Resource?

Programmable Resource Life Cycle Management

2017
What is Life Cycle?

Programmable Service Management

2018
What is Service?
How Far can you go with your cloud?

CMMI Model
Capability Maturity Model Integration
developed by CMU

ITIL
Service Delivery Strategy
Service Support Strategy
Service Security Strategy

Level 1: Initial
Processes unpredictable, poorly controlled and reactive

Level 2: Managed
Processes characterized for projects and is often reactive.

Level 3: Defined
Processes characterized for the organization and is proactive. (Projects tailor their processes from organization’s standards)

Level 4: Quantitatively Managed
Processes measured and controlled

Level 5: Optimizing
Focus on process improvement

Service Delivery Strategy
Service Support Strategy
Service Security Strategy
### if (kakao) Cloud Producer: CMMI Dev perspective

<table>
<thead>
<tr>
<th>Maturity</th>
<th>Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td></td>
</tr>
<tr>
<td>Managed</td>
<td></td>
</tr>
<tr>
<td>Defined</td>
<td></td>
</tr>
<tr>
<td>Quantitatively Managed</td>
<td>Optimizing</td>
</tr>
</tbody>
</table>

#### Culture & Organization
- Teams organized based on platform/technology
- Defined and documented processes
- Cross functional Cells & Teams
- Orchestrated Deployment
- Fully automatic acceptance tests
- Trend Reports

#### Build & Deploy
- Centralized version control
- Automated build scripts
- No management of artifacts
- Manual deployment
- Environments are manually provisioned
- Cross-team continuous improvement
- Teams responsible all the way to production
- Zero touch Continuous Deployments

#### Release
- Infrequent and unreliable releases
- Manual process
- Infrequent but fully automated and reliable releases in environment
- Frequent releases
- Deployment releases
- Canary releases

#### Data Management
- Data migrations are performed manually, no scripts
- Data migrations using versioned scripts, performed manually
- Changes automated and versioned changes to datastores
- Verify expected business value
- Detects faults and fixed immediately, roll forward

#### Test & Verification
- Automated unit tests
- Separate test environment
- Automatic integration tests
- Static code analysis
- Test coverage analysis
- Fully automated tests
- Automatic performance/security tests
- Manual exploratory testing based on risk-based testing analysis

#### Information & Reporting
- Baseline process metrics
- Manual reporting
- Visible to report runner
- Measure the process
- Automatic reporting
- Visible to team
- Dynamic self-service of information
- Customizable dashboards
- Cross-reference across organizational boundaries
if (kakao): IaaS

What is the purpose of doing IaaS?

<table>
<thead>
<tr>
<th>CMM0</th>
<th>CMM1</th>
</tr>
</thead>
<tbody>
<tr>
<td>legacy</td>
<td>self service</td>
</tr>
<tr>
<td></td>
<td>Dev resource</td>
</tr>
<tr>
<td>output:</td>
<td>output:</td>
</tr>
<tr>
<td>ITF</td>
<td>krane (openstack cloud)</td>
</tr>
</tbody>
</table>
if (kakao): IaaS (service name KRANE)
if (kakao): KRANE(The Tech.)

Data Center Level Scalable Network

**Neutron Floating IP**

**Routing Table**
- Default GW: 192.168.1.1 eth1
- Host Route:
  - dest 10.10.100.2/32 to 10.10.100.1
  - dest 192.168.100.2 to 10.10.100.1

**Compute Node Router**

**Switch Namespace**

- Veth pair
- Gateway: 10.10.100.1

**Compute node**

- Neutron-bridge-agent
- Neutron-dhcp-agent
- Neutron-linuxbridge-agent

**Host Route**
- dest 10.10.100.2/32 to 10.10.100.1
- dest 192.168.100.2 to 10.10.100.1

**Data Center Level Scalable Network**
Cloud CMMI-2 - MaaS

What is the sole purpose of doing MaaS?

<table>
<thead>
<tr>
<th>CMM0</th>
<th>CMM1</th>
<th>CMM2</th>
</tr>
</thead>
<tbody>
<tr>
<td>legacy</td>
<td>self service Dev resource</td>
<td>limited Prod resources</td>
</tr>
<tr>
<td>output: cloudTF</td>
<td>output: krane (openstack cloud)</td>
<td>output: kami (MaaS)</td>
</tr>
</tbody>
</table>
if (kakao): MaaS (service name KEMI)

**KEMI Stats**
- ETL: Extract Transform Loading
- 메트릭데이터 ETL*
- 시계열 DB
- 임계치 기반 알림
- 로깅 정보수집
- 키워드 빈도 기반 알림
- ETL: Extract Transform Loading

**KEMI Alert**
- 일계칙 기반 알림
- 로깅 정보수집
- 전사 하둡
- KRANE
- 임계치 기반 알림
- ETL: Extract Transform Loading

**KEMI Log**
- 로그 데이터 ETL*
- 실시간 DB
- 장기 DB
- Secure DB
- 전사 하둡
- KRANE
- 임계치 기반 알림
- ETL: Extract Transform Loading
if (kakao): KEMI(The Tech.)
if (kakao): KEMI(The Tech.)

Lambda Architecture

RealTime

KEMI Log Aggregator

Batch

App Log

Container Log

KEMI Log Tailer (CLI)

KEMI DIKE

RealTime

Batch

every 5m

every 5~15m

kafka

hadoop

ehive

elasticsearch

kibana

hue

if (kakao)
if (kakao): MaaS (The Key.)

Monitoring Abstraction

curl -XPOST https://sauron/api/v2/query -d '{ "start": 1515917973043, 
"end": 1615917973043, 
"queries": [{
  "metric": "-",
  "aggregator": "max",
  "downsample": "15m-max",
  "filters": [{
    "filter": "data.load.avg_5m",
    "groupBy": true, "tagk": "metric",
    "type": "literal_or" },
  { "filter": "dkos-marathon",
    "groupBy": true, "tagk": "host",
    "type": "literal_or" } ]}'

SAURON API
• Retrieval Standardization
• RBAC
• API limiting

Connection with IMS(CMDB) by Events
What is the sole purpose of doing CaaS (Container As A Service)?

<table>
<thead>
<tr>
<th>CMM0</th>
<th>CMM1</th>
<th>CMM2</th>
<th>CMM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>legacy</td>
<td>self service Dev resource</td>
<td>limited Prod resources</td>
<td>Automated CloudUsage</td>
</tr>
<tr>
<td>output: cloudTF</td>
<td>output: krane (openstack cloud)</td>
<td>output: kemi (MaaS)</td>
<td>output: DKOS (CaaS)</td>
</tr>
</tbody>
</table>
if (kakao): CaaS (service name DKOS)
if (kakao): DKOS(The Tech.)

Network Service Architecture: LB(L4)-LB(L7)-Container

DKOS Worker
DKOS LoadBalancer
SSL Offloading
L3DSR(Direct Server Return)

물리장비 LoadBalancer
if (kakao): DKOS (The Tech.)

**Autoscale**

**KEMI Alert Setup**

**DKOS Autoscale Group Setup**
if (kakao): DKOS(The Tech.)

Autoscale Result
if (kakao): DKOS(The key.)

DKOS is connected to everything
## Product & Service in Kakao Cloud

<table>
<thead>
<tr>
<th>Category</th>
<th>Target</th>
<th>Product</th>
<th>Description</th>
<th>Connected internal service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Project&amp;Role based Authentication</td>
<td>KEYSOEN</td>
<td>프로젝트/룹기반 사용자 인증 및 권한관리</td>
<td>내부 LDAP와 연동</td>
</tr>
<tr>
<td>Compute</td>
<td>Virtual machine</td>
<td>KRANE</td>
<td>가상머신 제공&lt;br&gt;가상 볼륨, 가상머신 기반 DB/LB제공&lt;br&gt;Autoscale 기능 준비중</td>
<td>IMS, *AEZORO, *Meta-kage 연동&lt;br&gt;*Tenth 연동</td>
</tr>
<tr>
<td></td>
<td>Container</td>
<td>DKOS</td>
<td>컨테이너 orchestrator(marathon) 제공.&lt;br&gt;컨테이너용 Loadbalancer 제공&lt;br&gt;컨테이너기반 Machine learning framework제공 (Spark, GPU)</td>
<td>KRANE, KEMI, AEZORO와 연동</td>
</tr>
<tr>
<td>Management</td>
<td>Monitoring</td>
<td>KEMI Stats</td>
<td>전사 메트릭 데이터 (cpu, memory, disk) &amp; custom (application, component) 데이터 수집 및 분석</td>
<td>*IMS와 연동 sauron 연계</td>
</tr>
<tr>
<td></td>
<td>Logging</td>
<td>KEMI Log</td>
<td>전사 로그 데이터 (syslog, app, container) 로그 수집/저장/분석</td>
<td>IMS와 연동&lt;br&gt;공용 하둡, Hbase연동</td>
</tr>
<tr>
<td></td>
<td>Alert</td>
<td>KEMI Alert</td>
<td>KEMI 데이터 기반 알람(특 및 url 전송)</td>
<td></td>
</tr>
<tr>
<td>Developer tools</td>
<td>Package Repository</td>
<td>repo.kakao.com</td>
<td>OS 패키지, Python/Ruby 패키지 제공</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Container Registry / Builder</td>
<td>D2Hub</td>
<td>컨테이너용 이미지 제작 및 등록</td>
<td>Keystone 연동</td>
</tr>
<tr>
<td></td>
<td>DNS as a service</td>
<td>AEZORO</td>
<td>가상머신/컨테이너기반 DNS 자동 등록/삭제</td>
<td>내부 DNS연동</td>
</tr>
</tbody>
</table>

*AEZORO(사내 DNS as a Service API)*<br>IMS(Infra Management System)<br>*Meta-kage(swift(오픈스택 스토리지 API)용 kage)*<br>*Ceph-Tenth: Ceph API 호환 Tenth*
kakao cloud is heading for DevOps (Self service)
• What is cloud native? (by CNCF)
  • Containerized
  • Micro-service oriented
  • Dynamically orchestrated
Little bit more specific

- Cloud Native Applications Characteristics
  - Micro-services
  - Health Reporting
  - Telemetry Data
  - Resilience
  - Declarative

First Ask ‘Why we need all these?’

Dynamically Orchestrated Platform

https://pivotal.io/cloud-native)
if (kakao):
Characteristics to Requirement

- C. N. Applications Characteristics
  - Microservices
  - Health Reporting
  - Telemetry Data
  - Resiliency
  - Declarative

- Platform Requirement
  - Isolation
  - Resource Allocation /Scheduling
  - Service Discovery
  - Monitoring/Logging
  - Metric Aggregation
  - Debugging and Tracing
Cloud CMMI4-Cloud Native Platform

What is the sole purpose of doing Cloud Native Platform?

<table>
<thead>
<tr>
<th>CMM0</th>
<th>CMM1</th>
<th>CMM2</th>
<th>CMM3</th>
<th>CMM4</th>
</tr>
</thead>
<tbody>
<tr>
<td>legacy</td>
<td>self service</td>
<td>limited Prod</td>
<td>Automated CloudUsage</td>
<td>Integrated Service</td>
</tr>
<tr>
<td></td>
<td>Dev resource</td>
<td>resources</td>
<td></td>
<td>Platform</td>
</tr>
<tr>
<td>output:</td>
<td>output: krane</td>
<td>output: kemi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cloudTF</td>
<td>(openstack</td>
<td>(MaaS)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>cloud)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
if(kakao):
    CloudPlatform(ServiceName 9rum)
if(kakao): 9rum (The tech.)

- Cloud Native Applications Characteristics
- Platform Requirement
  - Isolation
  - Resource Allocation and Scheduling
  - Health Reporting
  - Telemetry Data
  - Resiliency
  - Declarative
  - Debugging and Tracing

<table>
<thead>
<tr>
<th>Category</th>
<th>Target</th>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Role based Authentication</td>
<td>KEYSTONE</td>
<td>등급별 사용자 권한</td>
</tr>
<tr>
<td>Compute</td>
<td>Virtual machine</td>
<td>KRANE</td>
<td>가상작업의 제거, 가상화, 가상화된 기기 O/S 논의, Autoscale 기능 준비</td>
</tr>
<tr>
<td>Container</td>
<td>DKOS</td>
<td></td>
<td>계산원 (middleware) 제공, 컨테이너 바운더리, Loadbalancer 제공, 컨테이너 기반 Machine learning framework 제공 (Spark, GPU)</td>
</tr>
<tr>
<td>Monitoring</td>
<td>KEMI Stats</td>
<td></td>
<td>전자 패턴의 데이터 (cpu, memory, disk) &amp; custom application, component의 데이터 수집 및 분석</td>
</tr>
<tr>
<td>Logging</td>
<td>KEMI Log</td>
<td></td>
<td>전자 로그 데이터 (syslog, app, container) 로그 수집/저장/분석</td>
</tr>
<tr>
<td>Alert</td>
<td>KEMI Alert</td>
<td></td>
<td>KEMI 에러 기반 알림/ 대응 (UI 전반)</td>
</tr>
<tr>
<td>Package Repository</td>
<td>rep.kakao.com</td>
<td></td>
<td>OS 패키지, Python/Ruby 패키지 제공</td>
</tr>
<tr>
<td>Developer tools</td>
<td>Container Registry / Builder</td>
<td>D2Hub</td>
<td>컨테이너화, 이미지, 백업 및 동록</td>
</tr>
<tr>
<td>DNS as a service</td>
<td>AEZORO</td>
<td></td>
<td>가상화된 컨테이너 기반 DNS 저장/통합/재배</td>
</tr>
</tbody>
</table>
if (kakao): 9rum (The key)

Integrated SSO.

KRANE

DKOS

KEMI

9rum.cc

keystone DB

Authorization

authentication

Inhouse auth (Hello-MIS)

QUOTA

RBAC

Project

IMS (CMDB)

ERP
if (kakao) : some numbers

- Virtual to Physical ratio (%): 35
- Container slave ratio over Cloud (%): 60
- Indexing Data (TB/Day): 50
- Expense $MM/Yr: 80
- License Expense: 0

*(compared to public cloud, excluding traffic/storage)*
if (kakao): Try to open the source
github.com/kakao/kfield
github.com/kakao/d2hub