Cloud 환경기반 서비스의 효율적이고 안정적인 운영을 위한 DevOps 사례 소개

무선사업부 | Cloud운영그룹 | 김영욱
18/10/2018
오픈소스와 DevOps

발표자 소개 01
DevOps 소개 02
Infrastructure as Code 03
Immutable Server 04
Continuous Delivery 05
DevOps, Bixby
TechOps, SmartThings
Architect, MILK Music
WORKED FINE IN DEV

OPS PROBLEM NOW
### Alerts/Alarm
- Get notification from monitor system

### Check
- Check status of application/system

### Restart service
- If service is unhealthy, try to restart service
- Create a incident ticket to JIRA

### Reboot instance
- If service is still going unhealthy after restart, try to make snapshot for inspection
- And reboot instance

### Terminate instance
- Finally, terminate instance to bring new one automatically

### Communication
- Channel
- Notification
- Alert
- Ticket
Alerts/Alarm
- Get notification from monitor system

Check
- Check status of application/system

Restart service
- If service is unhealthy, try to restart service
- Create a incident ticket to JIRA

Reboot instance
- If service is still going unhealthy after restart, try to make snapshot for inspection
- And reboot instance

Terminate instance
- Finally, terminate instance to bring new one automatically

Communication
- Channel
- Notification
- Alert
- Ticket
<table>
<thead>
<tr>
<th>Immutable Server</th>
<th>Packer, Ansible, Chef</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure as Code</td>
<td>Terraform</td>
</tr>
<tr>
<td>Release Pipeline</td>
<td></td>
</tr>
</tbody>
</table>
| Deployment Strategies | • Blue/green deployment  
• Canary release  
• Rolling update |
<p>| Cluster Management | Spinnaker |</p>
<table>
<thead>
<tr>
<th>Automated Test</th>
<th><img src="image" alt="nGrinder" /></th>
<th><img src="image" alt="LOCUST" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor</td>
<td><img src="image" alt="DATADOG" /></td>
<td><img src="image" alt="ZIPKIN" /></td>
</tr>
<tr>
<td>Log Stream</td>
<td><img src="image" alt="sumologic" /></td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td><img src="image" alt="slack" /></td>
<td><img src="image" alt="Jira" /></td>
</tr>
</tbody>
</table>
Infrastructure (as) Code
Infrastructure as Code

**Blueprint Your Infrastructure**
- Compute
- Application
- Storage
- Security
- Network
- Configuration Standards

**Build Anything**
- Provision compute resources in the cloud
  - Infrastructure
  - Application Stacks
  - HPC
  - Linux, Windows, Unix

**Manage It Simply**
- Automatically reconfigure everything
- Re-provision for disaster recovery

---

**Application**
- Source Code
- Build
- Test
- Release

**Architecture**
- Source Code
- Configuration
- Test
- Deploy
Infrastructure as Code, Why?

- Because humans make mistakes
- Because human brain is terribly bad at repetitive tasks
- Because human is slow compared to a bash script
- ... and because we are humans

Source: https://www.slideshare.net/OCTOTechnology/introduction-to-devops-28779951
코딩하자
Infrastructure as Code

- Version control & Code review
- Simulation
- Security as Code
- One click Disaster Recovery

Desired Architecture → Terraform Code → Terraform Plan/Apply → Provisioned Architecture

```
$ terraform apply
aws_instance.example: Creating...
  ami: "ami-0d729a60"
  instance_type: "t2.micro"

aws_instance.example: Still creating... (15s elapsed)
aws_instance.example: Creation complete

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

```
$ cat example.tf
provider "aws" {
  access_key = "ACCESS_KEY_HERE"
  secret_key = "SECRET_KEY_HERE"
  region = "us-east-1"
}
resource "aws_instance" "example" {
  ami = "ami-0d729a60"
  instance_type = "t2.micro"
}
```
Infrastructure Building Blocks

Terraform Module Registry

Spinnaker
Jenkins
VPC

Your Infrastructure

Terraform codes

AWS seoul
Web Server
Application Server
Database Server

AWS ohio
Web Server
Application Server
Database Server

AWS singapore
Web Server
Application Server
Database Server

AWS ireland
Web Server
Application Server
Database Server

SOSCON 2018
```hcl
# Server group for microservice

### Security group for autoscaling group
resource "aws_security_group" "this" {
  name = "$[local.name]"
  description = "$[local.name] ASG 50"
  vpc_id = "$[var.vpc]"
  tags {
    Name = "$[local.name]"
  }
}

resource "aws_security_group_rule" "ingress_rules" {
  count = "$[length(var.ingress_rules)]"
  type = "ingress"
  cidr_blocks = ["$[element(var.ingress_rules[count.index], 0)]"]
  from_port = "$[element(var.ingress_rules[count.index], 2)]"
  to_port = "$[element(var.ingress_rules[count.index], 3)]"
  protocol = "$[element(var.ingress_rules[count.index], 4)]"
  description = "$[element(var.ingress_rules[count.index], 5)]"
  security_group_id = "$[aws_security_group.this.id]"
}

resource "aws_security_group_rule" "egress_rules" {
  count = "$[length(var.egress_rules)]"
  type = "egress"
  cidr_blocks = ["$[element(var.egress_rules[count.index], 0)]"]
  from_port = "$[element(var.egress_rules[count.index], 1)]"
  to_port = "$[element(var.egress_rules[count.index], 2)]"
  protocol = "$[element(var.egress_rules[count.index], 3)]"
  description = "$[element(var.egress_rules[count.index], 4)]"
  security_group_id = "$[aws_security_group.this.id]"
}

### Launch configuration
resource "aws_launch_configuration" "this" {
  count = "$[var.s3bucket_name]"
  image_id = "$[var.image_id]"
  instance_type = "$[var.instance_type]"
  key_name = "$[var.s3_bucket_key_name]"
}
```
Immutable Server
Artifact
Machine
Infrastructure
Immutable Infrastructure

We can replace them with new image in an incident. We don’t need to touch the running server to fix problem.
Packer is a tool for creating machine and container images for multiple platforms from a single source configuration.
How to Build/Bake Immutable Images

Ansible Roles
- JRE
- Linux

Base AMI

Ansible Roles
- Your Code
- Spring
- Log4j

Application AMIs
Ready to Deploy

Packer Template

Base

Packer Template
# add repository of spinmaker releases
- name: Add a repository of stable releases
  apt_key:
    url: https://keyserver.ubuntu.com/pki/KEYID/2018/12/12/720.0x405.0
    state: present
  apt_repos:
      state: present
      update_cache: yes
      when:
        - spinmaker_release is defined
        - spinmaker_release == "stable"

- name: Add a repository of development builds
  apt_key:
    url: https://keyserver.ubuntu.com/pki/KEYID/2018/12/12/720.0x405.0
    state: present
  apt_repos:
    - repo: deb [arch=amd64] https://dl.bintray.com/spinmaker-releases/debian trusty spinmaker
      state: present
      update_cache: yes
      when:
        (spinmaker_release is undefined or spinmaker_release != "stable")

# install package
- name: spinmaker-cloud-driver-1[ clouddriver_version ] state=present force=yes
  when:
    - clouddriver_version is defined
    - clouddriver_version == "latest"

- name: spinmaker-cloud-driver state=present force=yes
  when:
    (clouddriver_version is undefined or clouddriver_version == "latest")
인류 역사를 바꾼 100대 사건 중 1위!

세계 최초 금속활자 탄생의 미스터리

6월 역사의 첫으로 쓰여야 한다
Continuous Delivery
Production-ready software

Fast, automated feedback on the production readiness of your applications every time there is a change – to code, infrastructure, or configuration
### Continuous Delivery Pipeline

<table>
<thead>
<tr>
<th>Pipeline instance</th>
<th>Commit stage</th>
<th>Acceptance test stage</th>
<th>Release stage</th>
<th>UAT stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.133.12</td>
<td>Completed</td>
<td>Failed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0.133.13</td>
<td>Completed</td>
<td>Completed</td>
<td>Completed</td>
<td>Completed</td>
</tr>
<tr>
<td>0.0.133.14</td>
<td>Completed</td>
<td>Completed</td>
<td>Pending Trigger now</td>
<td>Pending Trigger now</td>
</tr>
<tr>
<td>0.0.133.15</td>
<td>Failed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0.133.16</td>
<td>Running</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SOSCON 2018
Let’s don’t manage the whole mess.

Just use the push button.
All connections between **blue** and **red** or **green** and **red** utilize mTLS and a certificate signed by our **internal certificate authority**.

All connections to **GATE** utilize mTLS and a certificate signed by our **external certificate authority**.
### Application Status

<table>
<thead>
<tr>
<th>Last Push</th>
<th>ap-northeast-2</th>
<th>eu-west-1</th>
<th>us-east-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>T J</td>
<td>2 months ago</td>
<td>2 % : 100%</td>
<td>2 % : 100%</td>
</tr>
<tr>
<td>M T E</td>
<td>a month ago</td>
<td>4 % : 100%</td>
<td>4 % : 100%</td>
</tr>
<tr>
<td>O I</td>
<td>20 hours ago</td>
<td>4 % : 100%</td>
<td>4 % : 100%</td>
</tr>
<tr>
<td>D U N I</td>
<td>19 hours ago</td>
<td>17 % : 100%</td>
<td>45 % : 100%</td>
</tr>
<tr>
<td>P I N I API</td>
<td>a year ago</td>
<td>2 % : 100%</td>
<td>-</td>
</tr>
<tr>
<td>N R C I</td>
<td>19 hours ago</td>
<td>49 % : 100%</td>
<td>240 % : 100%</td>
</tr>
<tr>
<td>W A Y</td>
<td>a month ago</td>
<td>4 % : 100%</td>
<td>4 % : 100%</td>
</tr>
<tr>
<td>G I R</td>
<td>a year ago</td>
<td>3 % : 100%</td>
<td>-</td>
</tr>
<tr>
<td>K</td>
<td>2 months ago</td>
<td>1 % : 100%</td>
<td>1 % : 100%</td>
</tr>
<tr>
<td>G O I</td>
<td>a year ago</td>
<td>1 % : 100%</td>
<td>1 % : 100%</td>
</tr>
<tr>
<td>U T</td>
<td>a day ago</td>
<td>8 % : 100%</td>
<td>23 % : 100%</td>
</tr>
<tr>
<td>A L G V</td>
<td>18 hours ago</td>
<td>24 % : 100%</td>
<td>90 % : 100%</td>
</tr>
<tr>
<td>P I E V O R</td>
<td>a day ago</td>
<td>5 % : 100%</td>
<td>6 % : 100%</td>
</tr>
<tr>
<td>B L E E T</td>
<td>a month ago</td>
<td>5 % : 100%</td>
<td>2 % : 100%</td>
</tr>
<tr>
<td>V I R Y</td>
<td>14 days ago</td>
<td>2 % : 100%</td>
<td>2 % : 100%</td>
</tr>
<tr>
<td>N B E</td>
<td>25 days ago</td>
<td>4 % : 100%</td>
<td>4 % : 100%</td>
</tr>
<tr>
<td>V N</td>
<td>25 days ago</td>
<td>4 % : 100%</td>
<td>4 % : 100%</td>
</tr>
<tr>
<td>K E C</td>
<td>a month ago</td>
<td>2 % : 100%</td>
<td>2 % : 100%</td>
</tr>
<tr>
<td>C H E R E R</td>
<td>2 months ago</td>
<td>2 % : 100%</td>
<td>2 % : 100%</td>
</tr>
<tr>
<td>W D I N</td>
<td>a month ago</td>
<td>2 % : 100%</td>
<td>4 % : 100%</td>
</tr>
<tr>
<td>W E J</td>
<td>5 months ago</td>
<td>5 % : 100%</td>
<td>4 % : 100%</td>
</tr>
</tbody>
</table>

### Pipeline Status

**DATA:** Deploy on Multi regions (get-readyBuild #10, started 2018-03-15 16:15:52 KST)

**COUNT:** PROD deploy on multiple regions (get-readyBuild #368, started 2018-03-15)

**RQD:** Deploy on AP03 (started 2018-03-15 11:09:03 KST)

**UTE:** PROD deploy on multiple regions (get-readyBuild #130, started 2018-03-15)

**DAPTOR:** Deploy on NA03 (PROD deploy on multi regions:Build #45, started 2018-03-15 10:45:20 KST)

**DAPTOR:** Deploy on AP03 (PROD deploy on multi regions:Build #45, started 2018-03-15 10:45:20 KST)

**D:** Deploy on NA03 (PROD deploy on multi regions:Build #39, started 2018-03-15)

**DAPTOR:** Deploy on EU02 (PROD deploy on multi regions:Build #45, started 2018-03-15 09:35:21 KST)

**D:** Deploy on EU02 (PROD deploy on multi regions:Build #39, started 2018-03-15)
Pipelines are the key deployment management construct in Spinnaker. They consist of a sequence of actions, known as stages.
**Spinnaker**

**Multi-Cloud** Continuous Delivery platform for releasing software changes with high velocity and confidence.

Key features
- Delivery Pipelines
- Deployment Strategies
- Cluster Management
- Jenkins Integration
- Supports Multi Cloud ([AWS, GCP, Azure, Kubernetes, ECS, DC/OS, Cloud Foundry])

We have experience in utilizing and hosting Spinnaker about 2 years.
Deployment Strategies
(Immutable Infrastructure)

Red/black (Blue/green)

Rolling red/black

Canary
Multi Cloud

VMs
Containers
Serverless
THANK YOU