DEVIEW 2018

# 쿠팡 서비스 클라우드 마이그레이션
## (Coupang Cloud Journey)

Oct. 11. 2018 / 양원석

coupang

# About Me

양원석

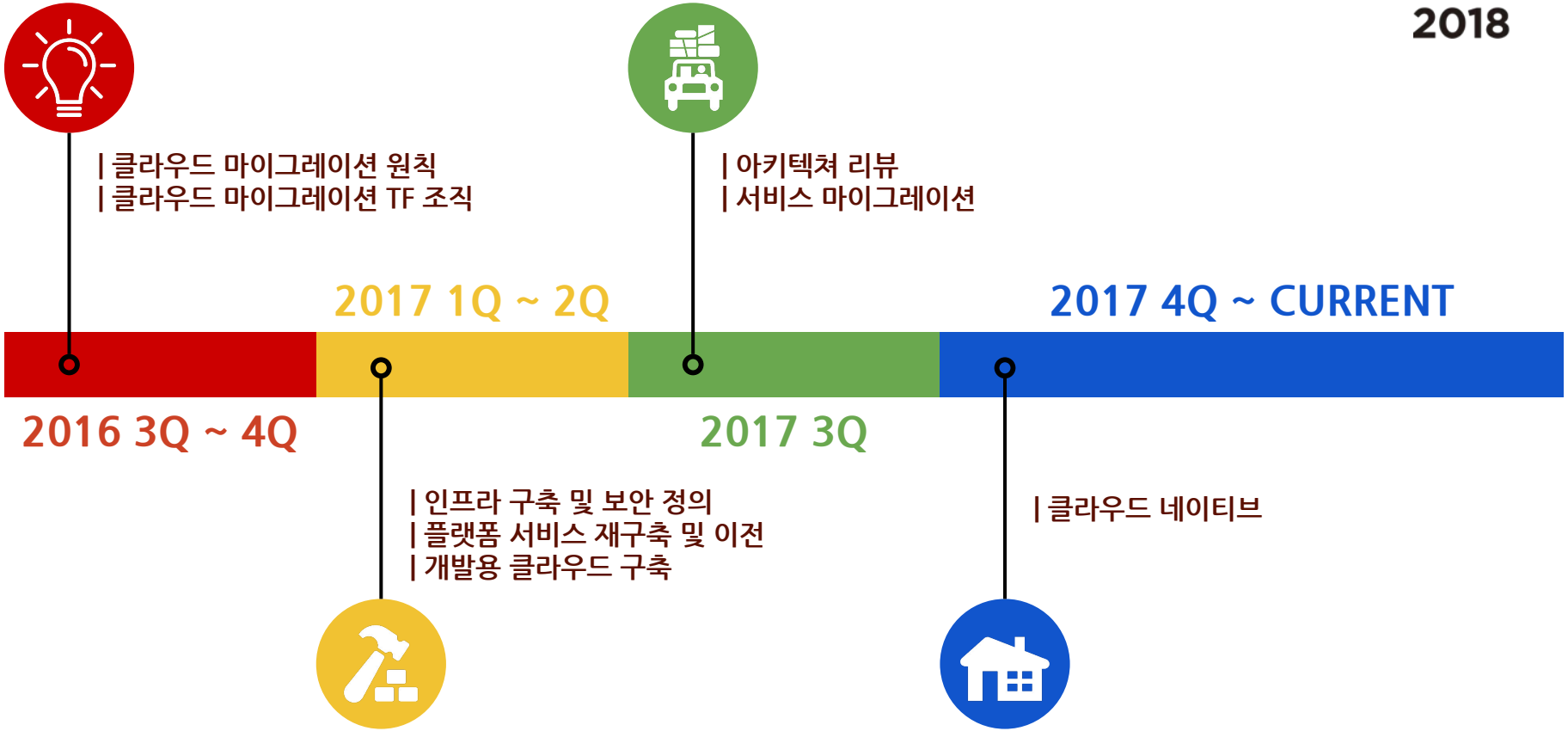
Principal S/W Engineer

Coupang, Core Platform Systems

2015. 9 ~

API Gateway, Common Framework

지난 2년 동안
쿠팡 서비스 클라우드 이전하면서
마주쳤던 문제들과 해결책
그리고
클라우드와 마이크로서비스가 만나면서
마주친 새로운 문제들과
정리했던 생각들을 공유합니다.


| 클라우드 마이그레이션 원칙
| 클라우드 마이그레이션 TF 조직
2016 3Q ~ 4Q
2017 1Q ~ 2Q
| 인프라 구축 및 보안 정의
| 플랫폼 서비스 재구축 및 이전
| 개발용 클라우드 구축
| 아키텍쳐 리뷰
| 서비스 마이그레이션
2017 3Q
2017 4Q ~ CURRENT
| 클라우드 네이티브

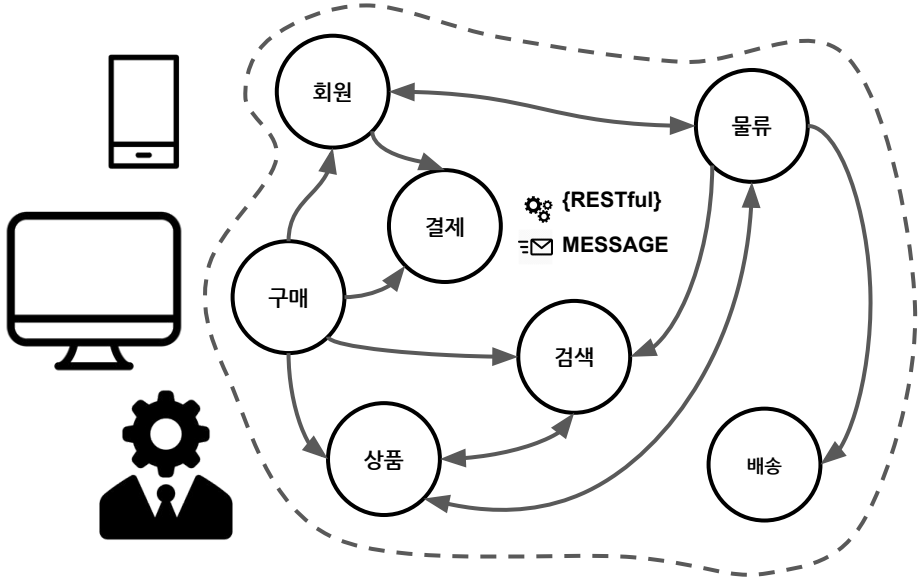
# 2016년, 여름

## 쿠팡 서비스 상태

- 100개의 microservice
- 목동 IDC, 분당 IDC

## 문제점

- 추가 및 확장 진행 시간
- 확장하지 못해서 장애 발생



# 클라우드 이전 원칙

확장성을 확보하기 위해 클라우드로 이전한다. (Scalability)

서비스는 무중단으로 이전한다. (Availability)

고객에게 만족도에 영향을 주지 않는다. (Performance)

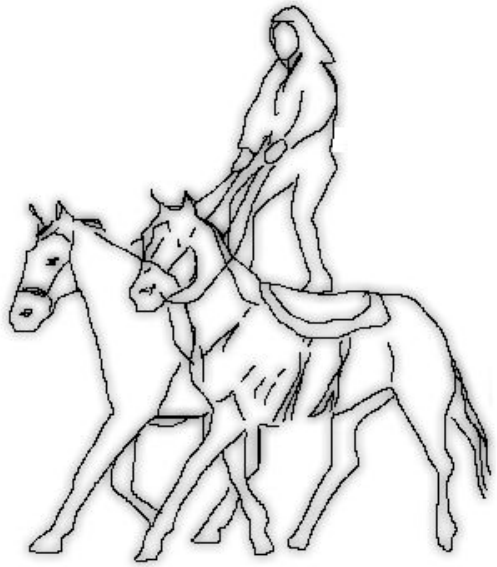
# 클라우드 이전 전략

## Roman Ride

- 데이터센터와 클라우드 동시운영
- 리스크 최소화

  작은 변화

  빠른 rollback



# 클라우드 이전 준비

Dynamic Routing

Canary Testing

Log 수집, 저장

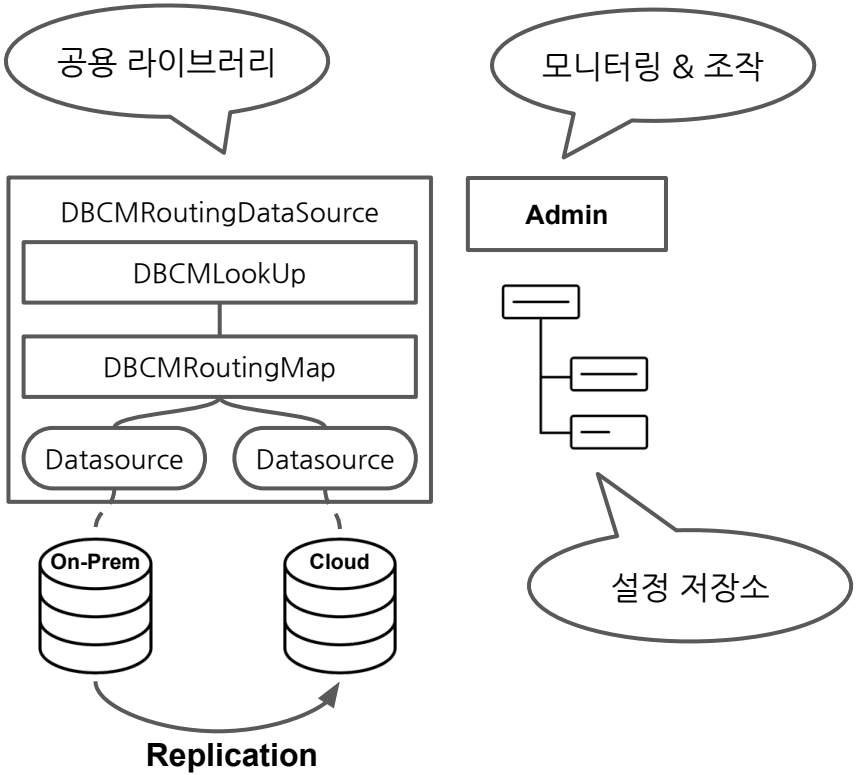
# 클라우드 이전 준비

- Dynamic Routing
- Canary Testing
- Log 수집, 저장

# Dynamic Routing I

## DB Connection Manager

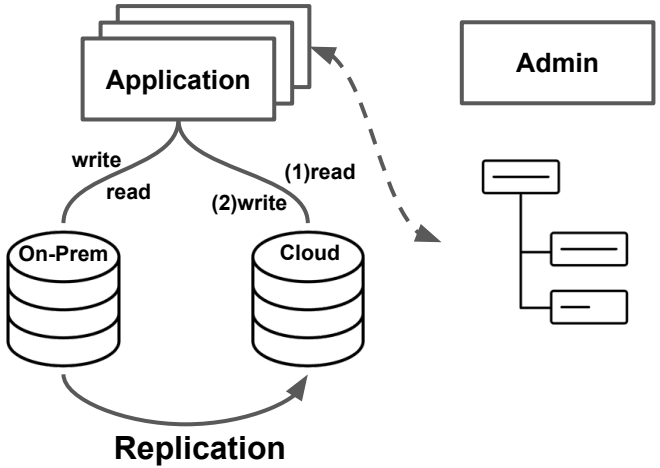
- 공통 라이브러리 형태
- Dynamic Config
- 상태 모니터링과 조작을 위한 Admin
- 빠른 rollback 지원



# Dynamic Routing I

## DB 이관 순서

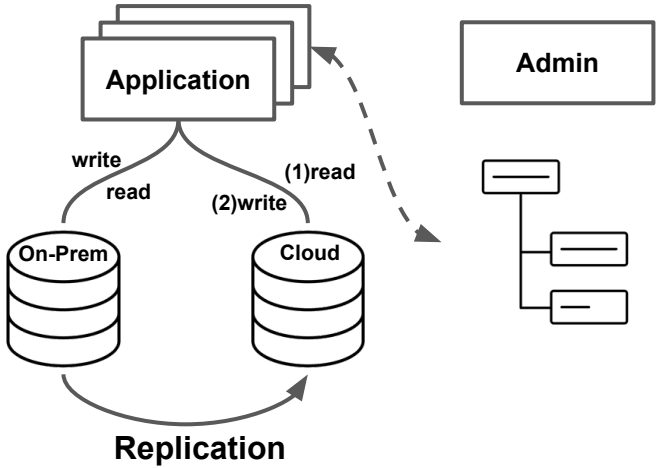
- DB Replication
- Read DB 연결 이동
- Write DB 연결 이동



# Dynamic Routing I

## Write 기능 일시 실패

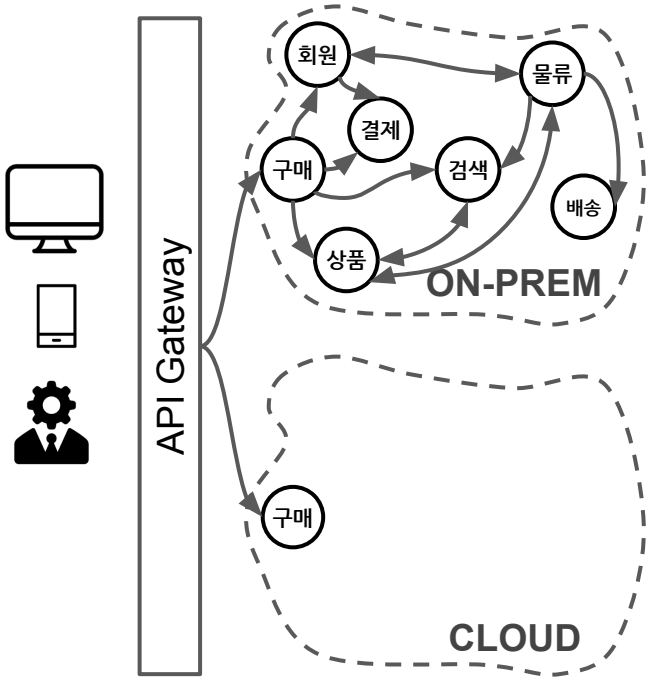
- Conflict를 막기 위한 전략
- microservice들의 retry를 활용
- Long Transaction 강제 실패



# Dynamic Routing II

## API G/W를 통한 트래픽 조절

- 기존 API G/W 활용
- 2개 의 Domain Name을 사용 트래픽 조절
- 빠른 Rollback 지원



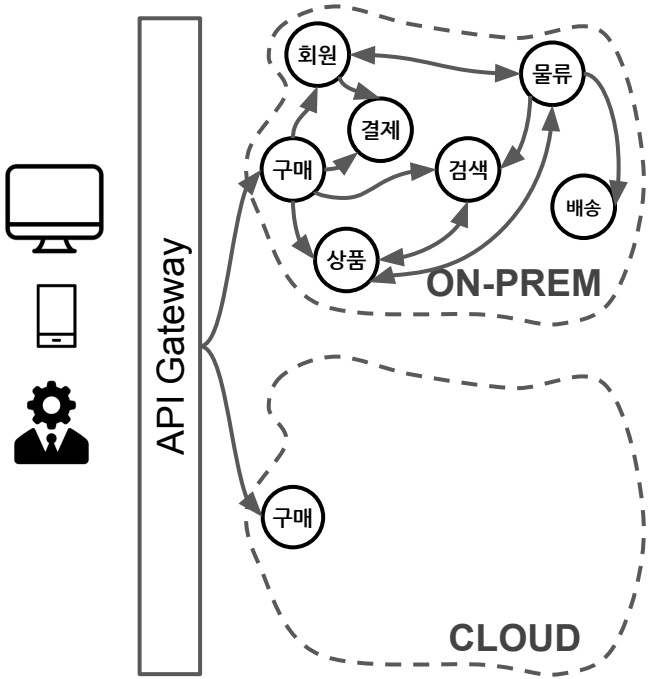
# Dynamic Routing II

## 마이크로서비스 클라우드 이관

- 트래픽이 작고, 영향도가 낮은 것부터 진행
- 0 - 100% 까지 트래픽 Ramp-up
- 빠른 Rollback

instance size 이슈

Load Balancer Warm-up 이슈



# 클라우드 이전 준비

Dynamic Routing

Canary Testing

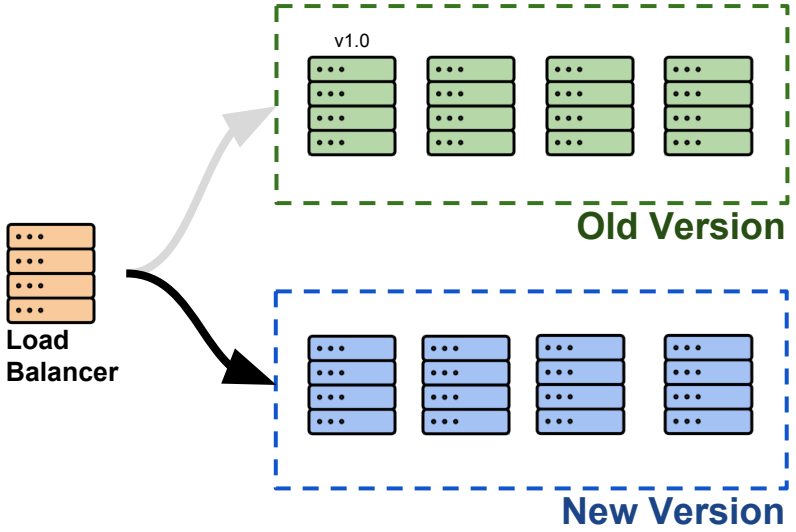
Log 수집, 저장

# Canary Testing

## Blue Green Deployment

| 무중단 배포

| 빠른 Rollback 지원

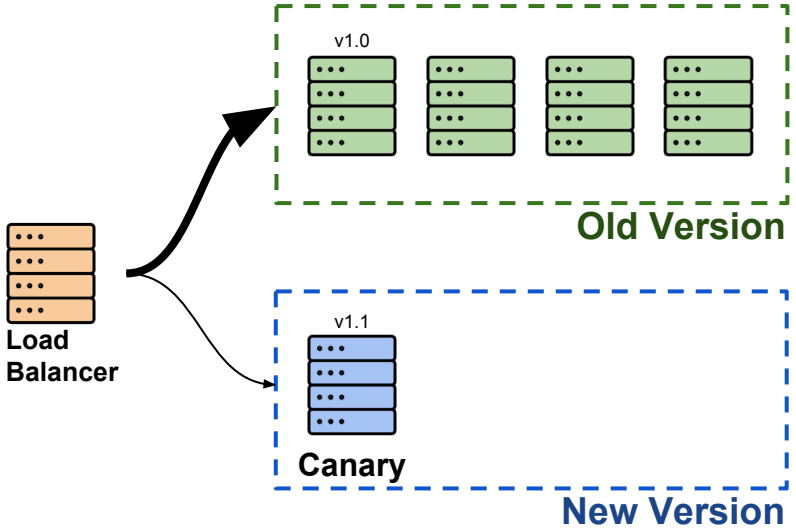


# Canary Testing

- 기존 환경과 비교 테스트
- 10분간 테스트 진행

metric 정보 비교

cpu, memory, load, etc

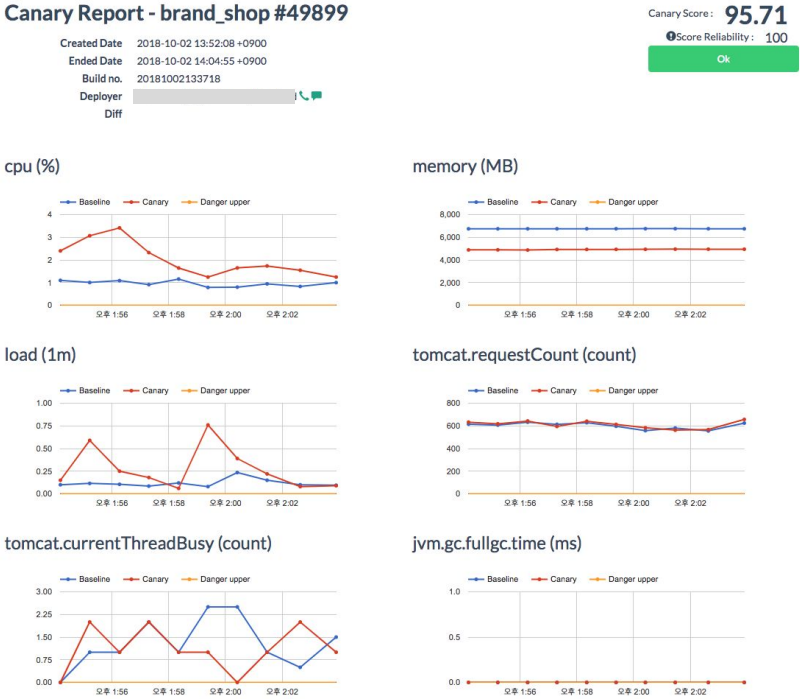


# Canary Testing

- 기존 환경과 비교 테스트
- 10분간 테스트 진행

metric 정보 비교

cpu, memory, load, etc



# 클라우드 이전 준비

Dynamic Routing

Canary Testing

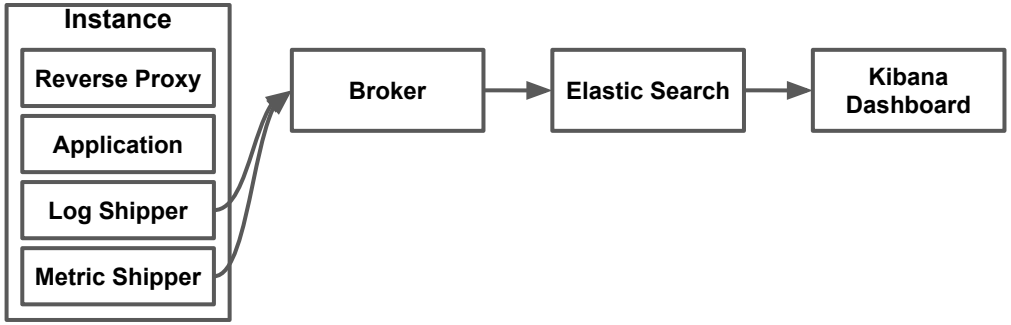
Log 수집, 저장

# Log 수집, 검색

## ELK Stack

- docker image 형태
- app, metric, syslog 수집
- custom log는 공통 디렉토리 수집

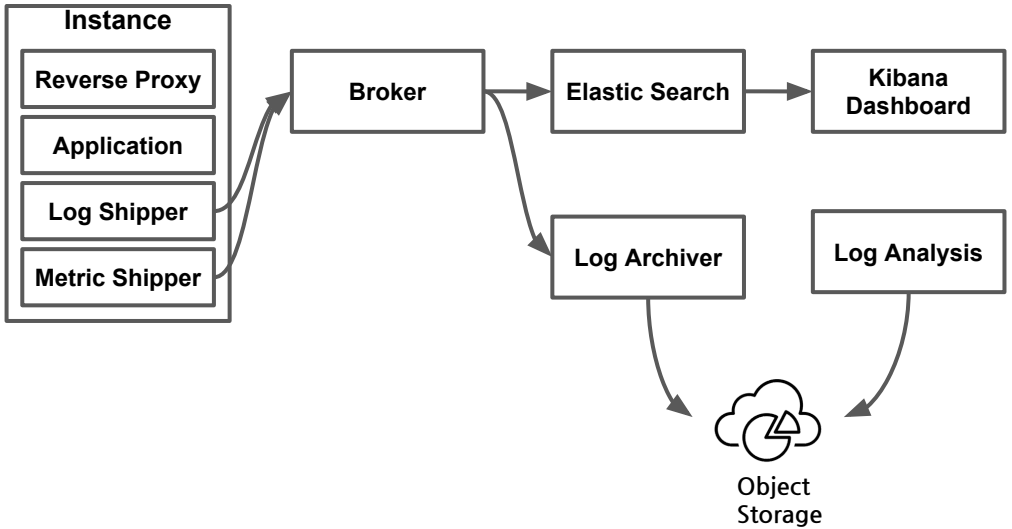
/pang/logs/app/



# Log 저장, 분석

## 로그 저장 & 분석

- Object Storage
- 압축, 라이프 사이클 적용



# 2017년, 8월

## 쿠팡 서비스 상태

- 클라우드로 이동 완료
- 200개의 microservice
- 5000대의 인스턴스

## 기존 문제점은?

- 추가 및 확장 리드 타임 감소
- 확장관련 장애 감소

**쿠팡, 자체기술 활용 3개월만에 클라우드 이전 완료**

IT 기술력 입증
마이크로 서비스 아키텍처가 큰 역할

김언한 기자 (unhankim@ebn.co.kr) **기사더보기 +**

등록 : 2017-08-10 09:01

댓글보기 | 페이스북 | 트위터 | 카카오스토리 | 밴드 | 구글+ | 가 + -

coupang

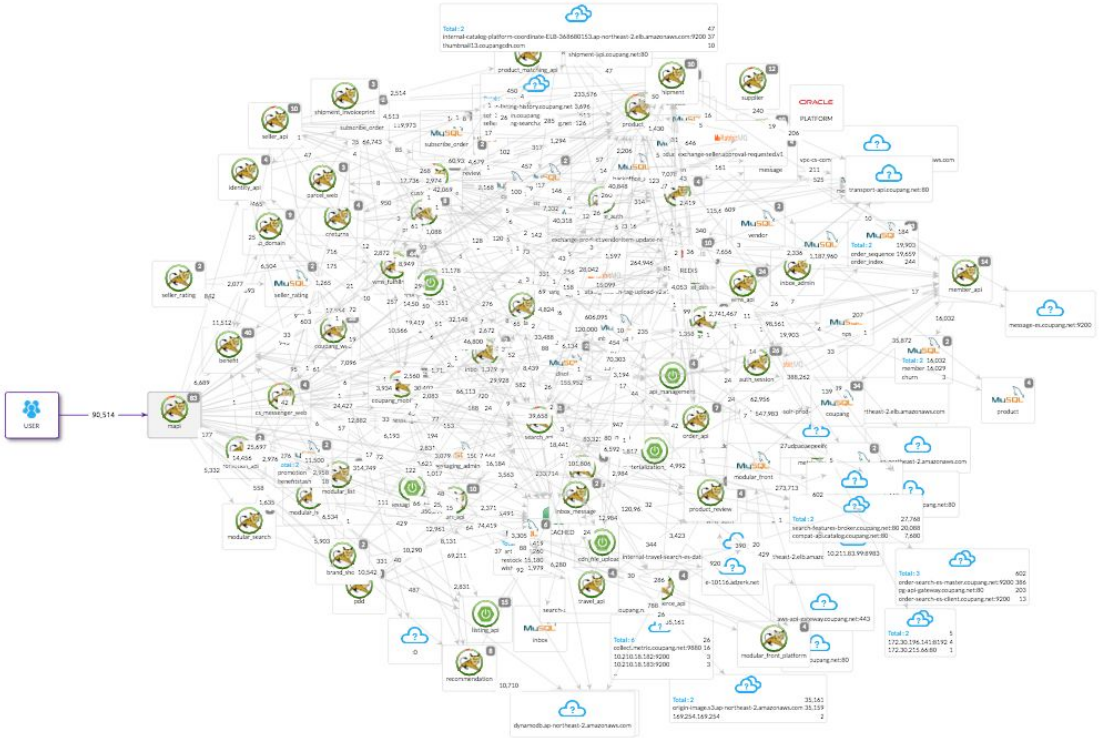
쿠팡은 자체 기술력으로 쿠팡 이커머스 서비스 전체의 클라우드 이전(마이그레이션)을 완료했다고 10일 밝혔다.

# 그리고 1년 …

# 2018년, 현재

## 쿠팡 서비스 상태

- 313 release/day
- 300개의 microservice
- 10,000대의 인스턴스
- 18,000,000 metric/day
- 7,000,000,000 req/day



# 새로운 문제들

## 전파되는 장애

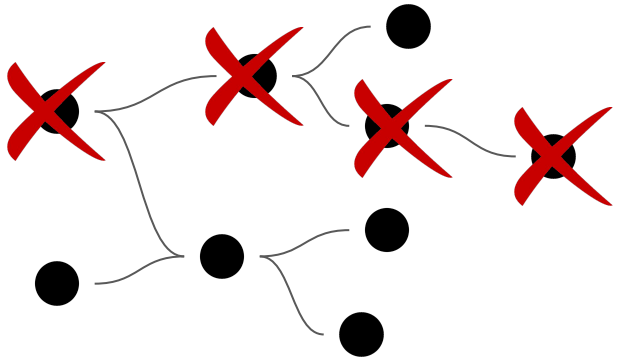
| 낮아지는 SLA ($99.99^{10} = 99.9$)

## 예상치 못한 곳에서 발생하는 장애

| Noisy Neighbor Problem

공용 자원, 클라우드 서비스 제공 자원

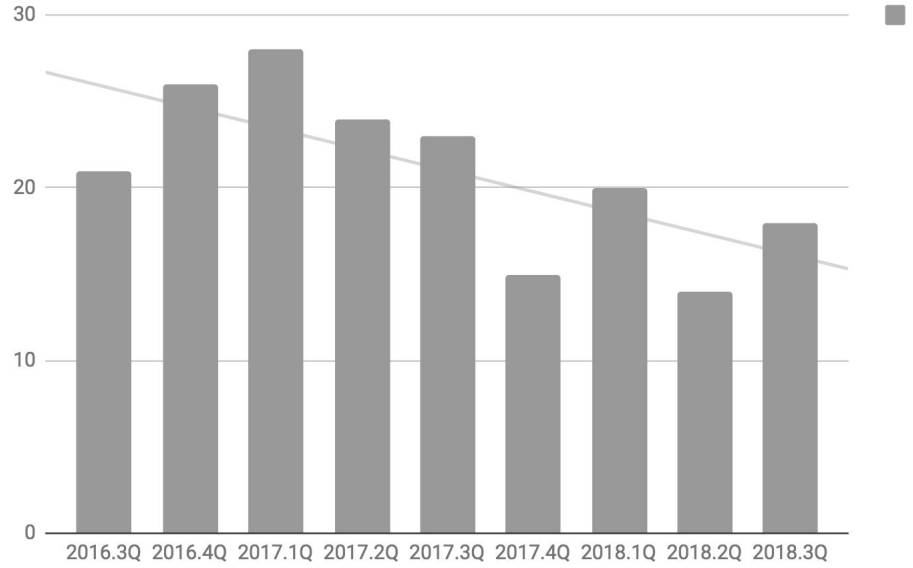
| 자동화와 장애



# 클라우드 이후 1년간

Blocker, Critical 장애 : 67건

관련 작업 : 167건



# 마이크로서비스와 클라우드를 통해 배운 것

- 모든 것에서 실패 가능
- 혼돈 속에 살기
- Auto Scaling
- 다른 장애로 부터 배우기

# 마이크로서비스와 클라우드를 통해 배운 것

**모든 것에서 실패 가능**

혼돈 속에 살기

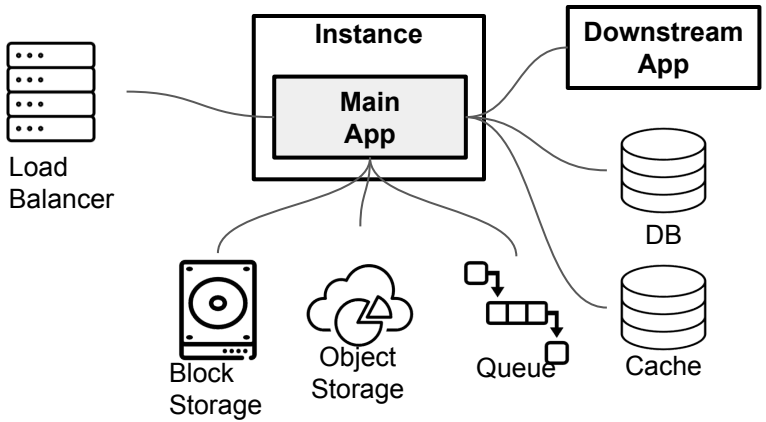
Auto Scaling

다른 장애로 부터 배우기

# 사용하는 모든 것에서 실패가 가능하다.

## 모든 것을 리소스로 생각하고 대비 필요

- Retry
- Fallback
- Circuit Breaker

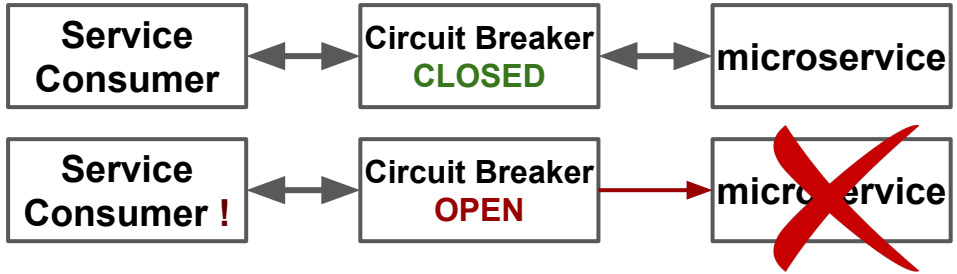


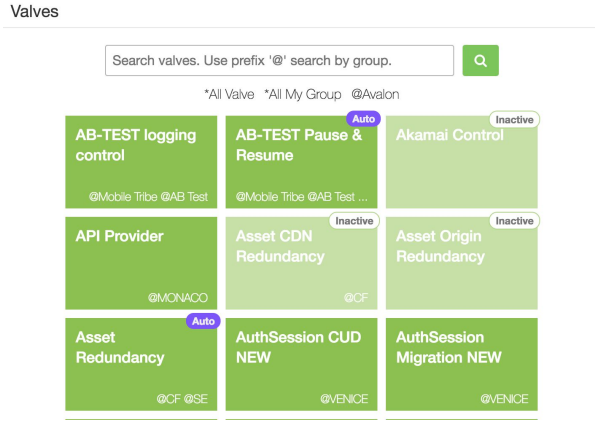
# Circuit Breaker

## 오작동 하는 서비스 연동 중지

- 장애 전파 방지
- 빠르게 실패하고 자동 회복
- 자체 솔루션, Hystrix

중앙관리, 분산




# 예측 못하는 것을 예측하라

## Fault Injection Testing

- 복구 기능 테스트
- 약점 찾아내기
- Chaos Engineering

Chaos Monkey

# 마이크로서비스와 클라우드를 통해 배운 것

모든 것에서 실패 가능

**혼돈속에서 살기**

Auto Scaling

다른 장애로 부터 배우기

# 장애 채널 스케치

"장애 복구 되었나요?"

"1시간 내에 배포나 변경된 내역 확인 부탁드려요!"

- 복잡한 시스템 상황에서 상태 확인 어려움
- 모든 서비스 관계를 알기 어려움

# 안정 상태 찾기

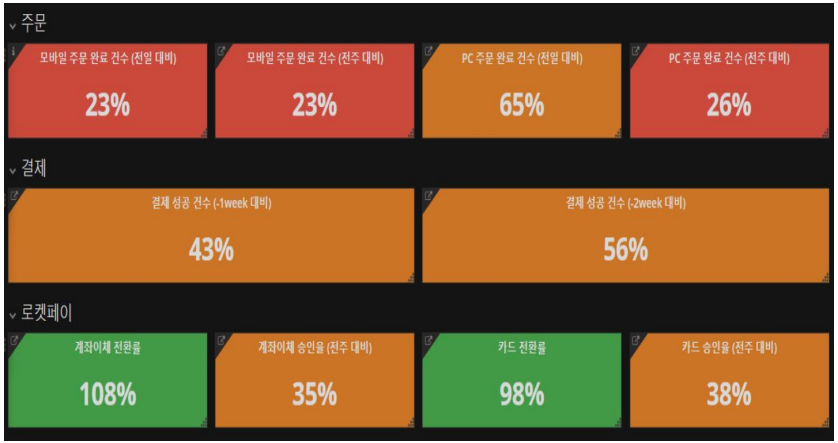
## 주문, 결제 카운트

- 서비스의 건강도 측정
- 주기 적극 활용

매달 1일 00시

일요일 밤 23:59분

매일 23:59

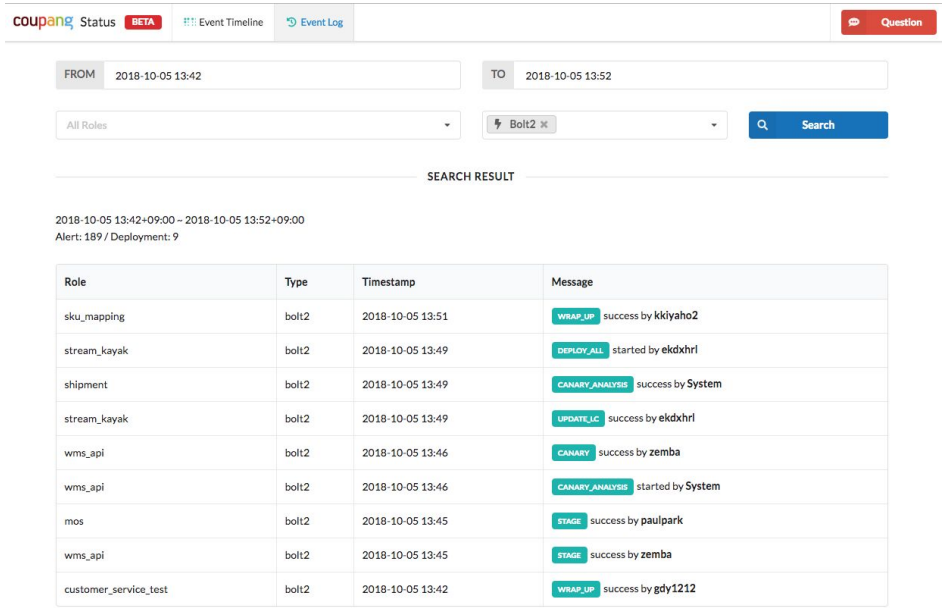


# 변경 내역 확인 하기

## 모두에게 필요한 기본 정보 수집

- 각 서비스의 상태
- 배포 이력
- 인프라 변경 이력

dns, security, auth



# 마이크로서비스와 클라우드를 통해 배운 것

모든 것에서 실패 가능

혼돈 속에서 살기

**Auto Scaling**

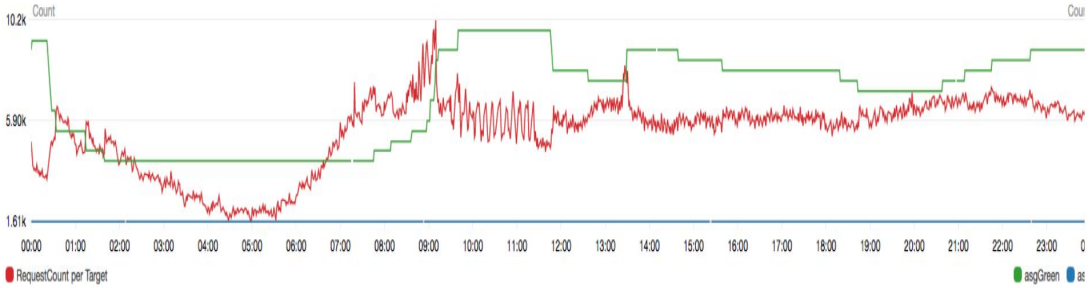
다른 장애로 부터 배우기

# Auto Scaling

## Auto Scaling

- 요청에 따라 자동 조절
- 이벤트 준비 시간 단축
- Target Tracking Policy

메트릭 정보(CPU, Request 등)



# Auto Scaling의 조건

## 폐기 가능 (Disposability)

| 빠른 시작과 빠른 정상 종료 보장

시작이 오래걸리면 Auto Scaling이 트래픽을 따라가지 못함

정상 종료가 오래 걸리면 새로운 배포시 리소스 문제 발생

| 빠르게 늘리고 천천히 줄인다

# 마이크로서비스와 클라우드를 통해 배운 것

모든 것에서 실패 가능

안정 상태

Auto Scaling

다른 장애로 부터 배우기

# 다른 장애로부터 배우기

## 사건 사고는 필연적인 것

- 대용량의 복잡한 분산 시스템
- 끊임없는 변화
- 지속적인 안정화

# 장애 리포트

## 타임라인

| Detection 에 걸린 시간

| 원인 찾는데 걸린 시간

| 복구에 걸린 시간

## 원인 찾기

| 고객 관점에서 5 why 작성

## 재발 방지

| Poka-Yoke

14:29 xx 서비스 배포 완료
14:32 yy 서비스 및 3개 서비스 에러카운트 증가로 alert 발생
14:33 oncall 담당자 noti 완료
14:34 주문 숫자 하락 확인/장애채널 생성
14:36 장애 등급 메이저 상향
14:42 xx 서비스 롤백 결정
14:45 xx 서비스 롤백 완료
14:47 주문 숫자 정상화 확인
14:50 서비스 정상화 판정

Q1. 왜 고객 주문을 하지 못했는가?
A1. 고객이 주문 페이지에 접근 하지 못했다.

Q2. 왜 고객이 주문 페이지에 접근 하지 못했는가?
A2. 마이쿠팡 페이지에서 주문 페이지로 넘어가는 동안 문제가 발생했다.

Q3. 왜 주문 페이지로 넘어가는 동안 문제가 발생했는가?
A3. xx 서비스가 사용하고 있는 YY 서비스가 응답이 느려지면서 문제가 발생했다.

...

# 다른 장애로 부터 배우기

## Site Reliability Engineering (SRE)

- Service Reliability를 책임
- 복잡한 장애 상황에서 컨트롤 타워
- 장애에 대한 지식 공유
- 장애 재발 방지 및 복구 자동화를 위한 노력

# 정리

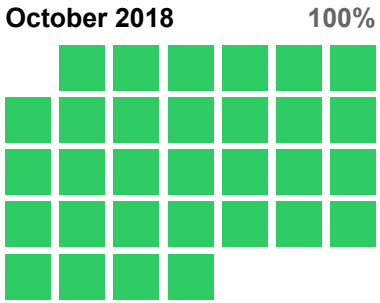
## 잘한것

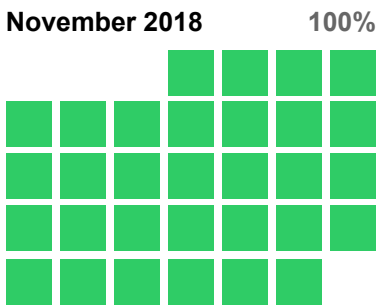
- 작은 변화와 빠른 rollback
- 공통 배포 파이프라인 유지
- 만든 사람이 운영하는 문화
- 장애 관리 문화

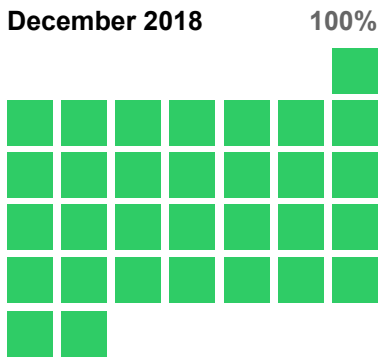
## 다르게해보고싶은것

- 복잡도 관리
- 도커 오케스트레이션 적용
- 클라우드 네이티브

# 감사합니다.



# 질문은 Slido에 남겨 주세요.

sli.do

#deview

TRACK1