

Microservice & Immutable Infrastructure

엔터프라이즈 시스템/네트워크 운영자 대상
(for IT Pros and System Administrators)

2018. 09.

안종석
james@jslab.kr

- 사용 유효기간을 2019년 3월 까지로 권합니다.
- 실습 교재는 별도 입니다.

JS Lab

목차

- A. 마이크로서비스 (별도 교재)
- B. 컨테이너 (별도 교재)
- C. Immutable Infrastructure
- D. 실습 (별도 교재)

JS Lab

C. Immutable Infrastructure

❖ 가용성 등급 (Availability Level)

- 가정의 전기 공급 연간 가용성(예): 독일 21분(99.9960%), 영국 75분(99.9857%), 프랑스 71분(99.9865%), 폴란드 260분(99.9565%), 미국 127분(99.9759%)

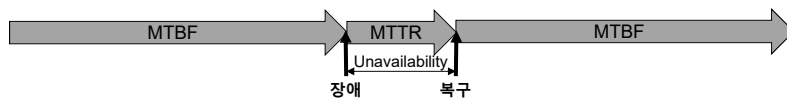
가용성 % (Availability)	Downtime per year	Downtime per month	Downtime per week
99.8%	17.5 시간	86.2 분	20.2 분
99.9% (three nine)	8.8 시간	43.2 분	10.1 분
99.99% (four nine)	52.6 분	4.3 분	1.0 분
99.999% (five nine)	5.3 분	25.9 초	6.1 초

JS Lab

C. Immutable Infrastructure

❖ MTBF 와 MTTR

- Mean Time Between Failure (MTBF)
 - 하드디스크(예)는 1000개를 3개월 시험하여 5개가 고장 시 1년에 20개로 산정하고, 총 8,760,000(=1,000 디스크 X 365 X 24) 시간 중 175,200(=20 디스크 X 365 X 24)을 장애 시간으로 계산
- Mean Time To Repair (MTTR)



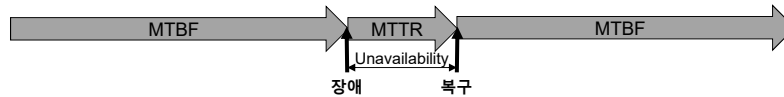
구성 (예)	MTBF (시간)	MTBF (일)	MTBF (년)
일반 하드 디스크	750,000	31,250	85.6
일반 전원 공급 장치	100,000	4,167	11.4
일반 팬(Fan)	100,000	4,167	11.4
이더넷 네트워크 스위치	350,000	14,583	40.0
RAM	1,000,000	41,667	114.1

JS Lab

C. Immutable Infrastructure

❖ MTTR (Mean Time Between Failure)

- 장애 인지 (알람 메시지 이전 시간)
- 알람 프로세싱
- 장애 원인 확인
- 복구 정보 확인
- 대체 장비 공급
- 기술자가 대체 장비와 도달 시간
- 물리적 복구 시간
- 대체 장비 테스트 후 재시작 시간



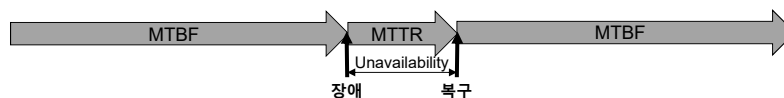
JS Lab

Community for KOREN AI Network Lab
james@jab.kr

C. Immutable Infrastructure

❖ 가용성 패턴

- Redundancy: SPOF를 피하는 이중화 (예: 전원장치 이중화등)
- Failover: 반자동으로 대기중(Standby) 시스템으로 전환하며 VMware의 'High Availability'나 Windows Server의 'Failover Clustering'이 있음
- Fallback: 수동전환으로 Hot site, Warm site, Cold site 등이 있음



JS Lab

Community for KOREN AI Network Lab
james@jab.kr

C. Immutable Infrastructure

❖ im·mu·ta·ble [imjú:təbəl] a.

변경할 수 없는, 불변의, 변치[바뀌지] 않는.

㉟-bly ad.

~~ness n.

im·mù·ta·bil·i·ty n.

How To Do Immutable?

- Provision a new server.
- Test the new server.
- Change the reference to the new server.
- Keep the old server around for rollback.

JS Lab

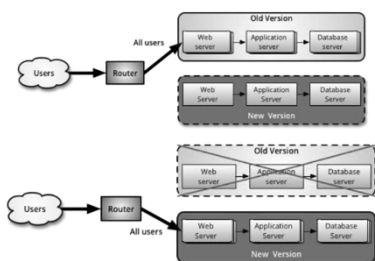
Community for KOREN AI Network Lab
james@jab.kr

C. Immutable Infrastructure

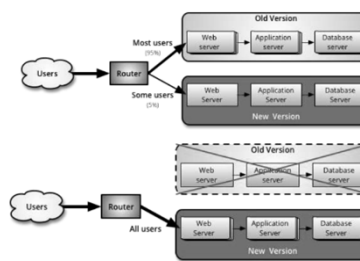
❖ Deployments

- Blue/Green Deployments
- Canary Deployments

Blue/Green Deployments



Canary Deployments

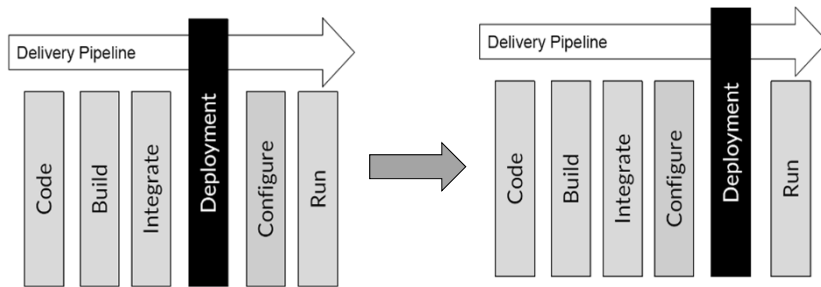


JS Lab

Community for KOREN AI Network Lab
james@jab.kr

C. Immutable Infrastructure

❖ Shifting Configuration BEFORE Deployment



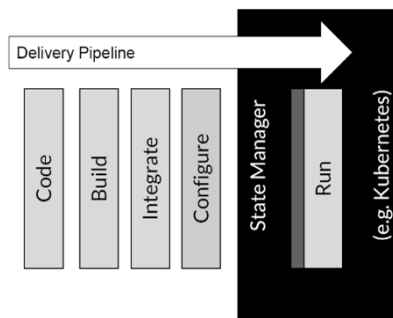
Community for KOREN AI Network Lab
james@slab.kr

JS Lab

C. Immutable Infrastructure

❖ Which Enables... Delegating Operations

- If you can make your artifacts immutable then you can delegate management of them to a platform like Kubernetes.
- Kubernetes does not configure infrastructure. It maintains state based on a manifest.

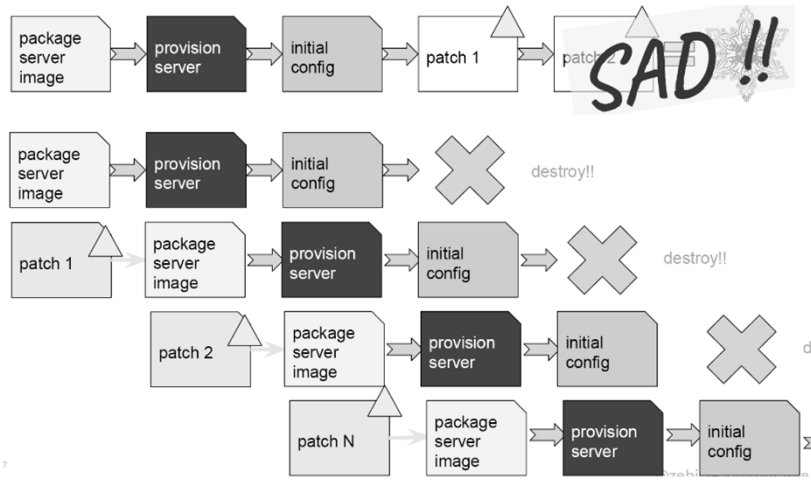


Community for KOREN AI Network Lab
james@slab.kr

JS Lab

C. Immutable Infrastructure

❖ Immutable is a DevOps Pattern

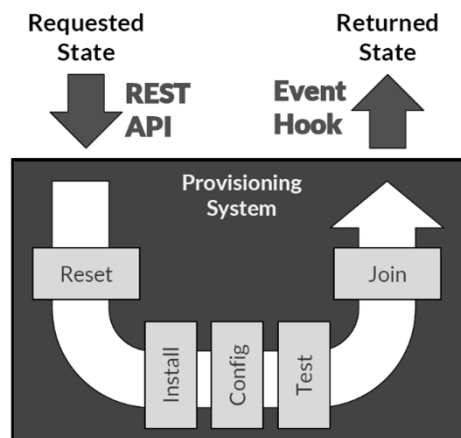


JS Lab

C. Immutable Infrastructure

❖ Cloud-like Integration and Staged Workflow

❖ Immutable Provisioning systems treat infrastructure as a black box

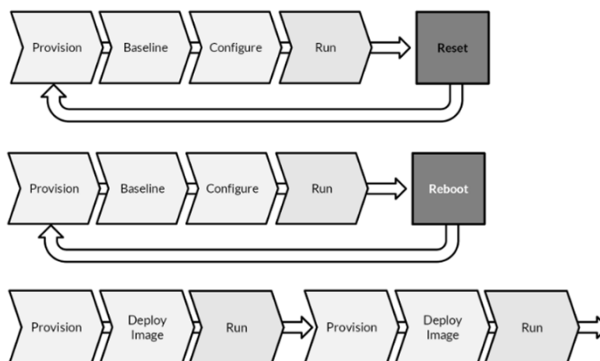


JS Lab

C. Immutable Infrastructure

❖ Immutable Patterns

- **Baseline + Configuration**
- **Live Boot + Configuration**
- **Image Deploy**



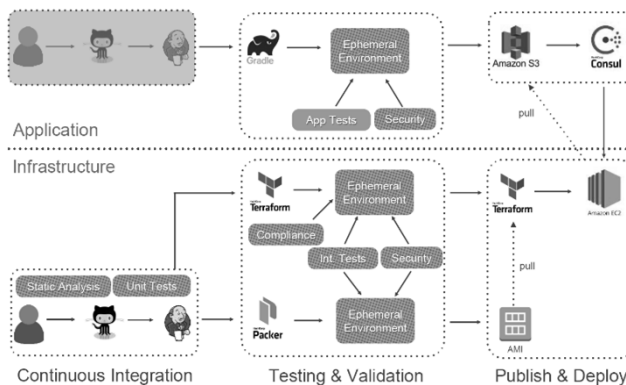
JS Lab

Community for KOREN AI Network Lab
james@jlab.kr

C. Immutable Infrastructure

❖ Containerized Services

- **Infra Module - Container Management System**
- **Fully Decoupled from Apps**
- **Apps are deployed with Container Management System specific tools**



JS Lab

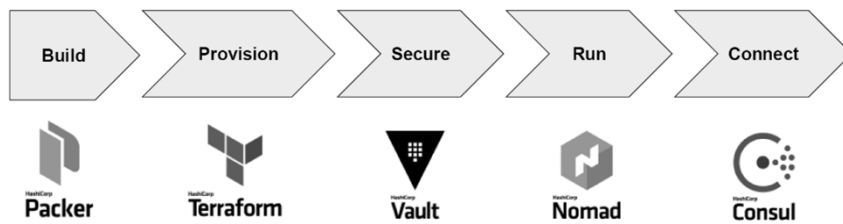
Community for KOREN AI Network Lab
james@jlab.kr

C. Immutable Infrastructure

❖ How do you do this with immutability?

- 머신 프로비저닝 (How do we provision a machine?)
- 요소간 연결 (How do we connect various parts?)
- 릴리즈 (How do we release?)
- 비용 제어 (How can we keep costs in control?)
- 서비스 방향 (Where have all the services gone?)
- 인프라 접속 (Who has access?)

❖ Tooling @ Cloud (예)



JS Lab

C. Immutable Infrastructure

❖ K8S

- Orchestration
- Deployment
- Scaling
- Data Plane

❖ Istio

- Policy Enforcement
- Traffic Management
- Telemetry
- Control Plane



JS Lab

Q&A



목차

- A. 마이크로서비스
- B. 컨테이너
- C. Immutable Infrastructure
- D. 실습 (별도 교재)

5. Q&A

