
opennaru
JBoss Troubleshooting Essential
장애 처리와 튜닝 프로세스(Thread Dump 분석)
KNOWLEDGE

opennaru

Application Performance Management

트러블 슈팅 절차 및 방법

KHAN
[apm]

KNOWLEDGE

시스템 구축 후 - 고민들

opennaru

KHAN [apm]

✓ 오늘 **온라인방문자 수가 급격히 줄었는데**요?
✓ 다음달에 **온라인 광고**를 해야 하는데 문제는 없겠조?
✓ **주기적으로 장애**가 발생하는데 대책은 없나요?
✓ 사용자가 조금 만 늘어도 **시스템이 다운이니 불안해서 뭘 할 수가 없네요.**

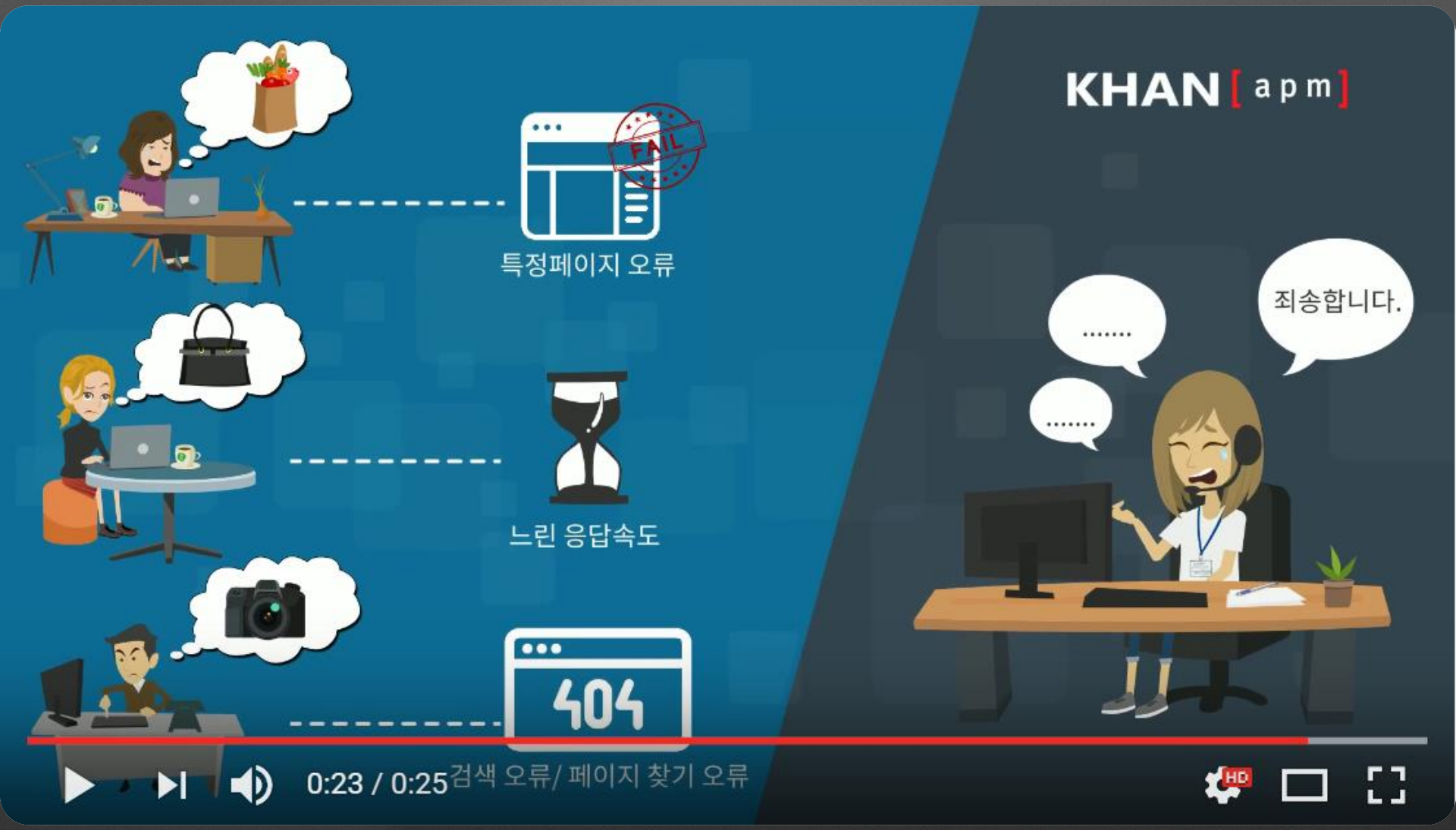
✓ 확인해 봤는데 **소스는 문제가 없어요.** WAS 문제 아닌가요?
✓ 느린 페이지에서 사용하는 **DB 쿼리를 보고 싶은데요.**
✓ 개발할 땐 빠른데 **운영시스템에서만 느린 이유**가 무엇인가요?

✓ 서비스가 갑자기 느려졌는데 **WAS 문제일까요?**
✓ 실시간 동시 사용자수나 TPS 등의 **기본 운영 정보를 알고 싶은데요.**
✓ 서비스가 느려질 때 **WAS를 Restart** 하면 정상적으로 동작해요.
✓ **특정페이지 나 서비스가 느린 이유는** 무엇일까요?

0:28 / 0:28

https://www.youtube.com/watch?v=jYjGoNEV-_E&index=12&list=PLaFPOkYzLL-_QWKsbT40cxwr_-cijU6L

고객이 체험하는 것은?



https://www.youtube.com/watch?v=jMUpM-1hx7Q&index=6&list=PLaFPOkYzLL-_QWKsbT40cxwr_-cijeU6L

엔터프라이즈 모니터링에서는 문제 없음

opennaru

https://www.youtube.com/watch?v=eE7rPuksd1c&index=7&list=PLaFPOkYzLL-_QWKsbT40cxwr_-cijeU6L

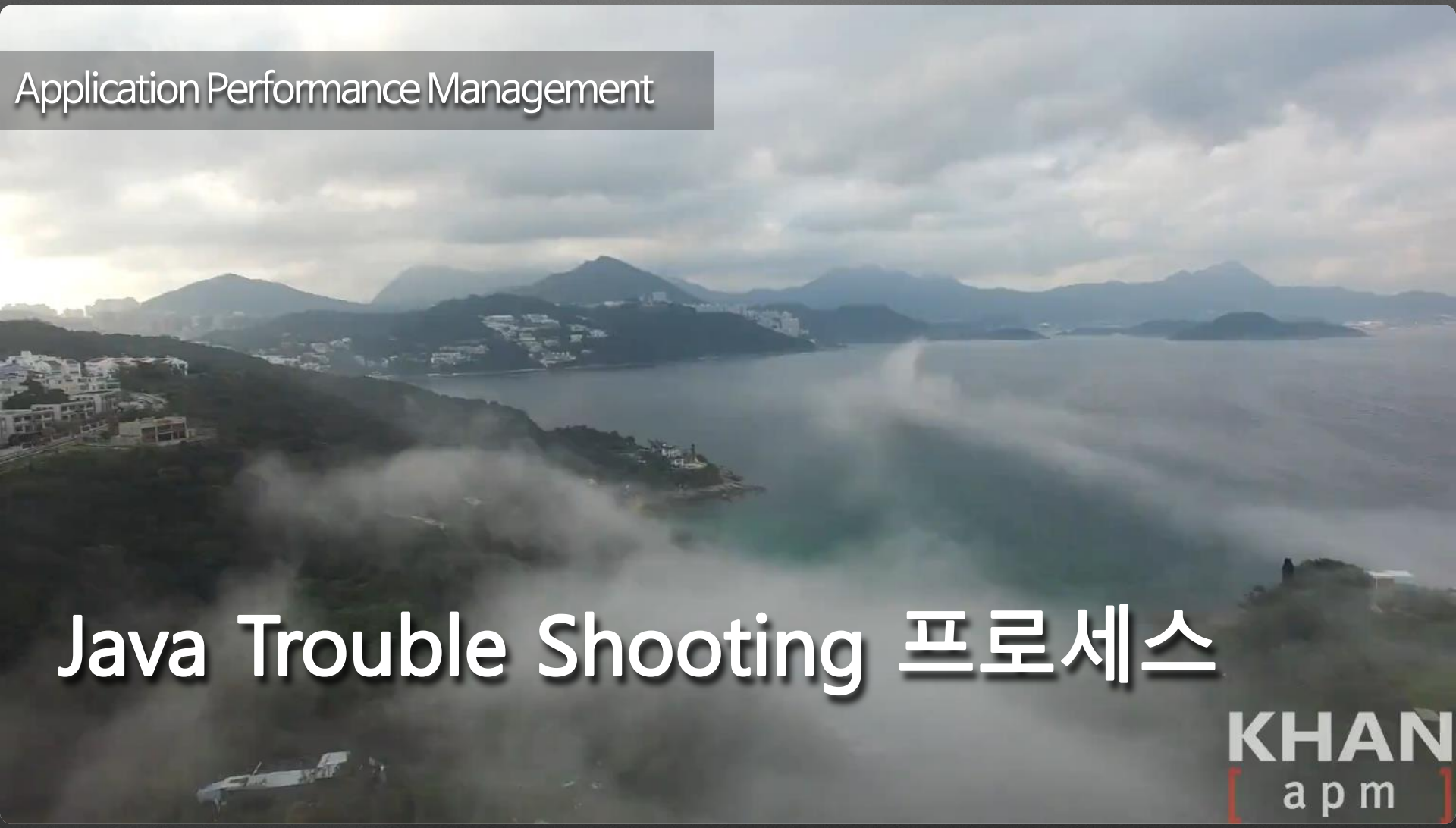
각 분야에서는 전문가 – 코끼리 만지기

opennaru

https://www.youtube.com/watch?v=P5S0gI-4hZI&list=PLaFPOkYzLL-_QWKsbT40cxwr_-cijeU6L&index=10

opennaru

Application Performance Management



Java Trouble Shooting 프로세스

KHAN
[apm]

Troubleshooting 공통 프로세스

장애발생

상황수집
(긴급조치,로그수집)

진단&모니터링

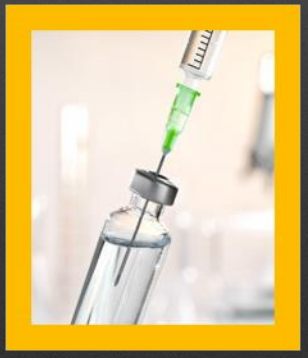
원인분석&재현

조치

복구

증상과 원인

진단 및 모니터링

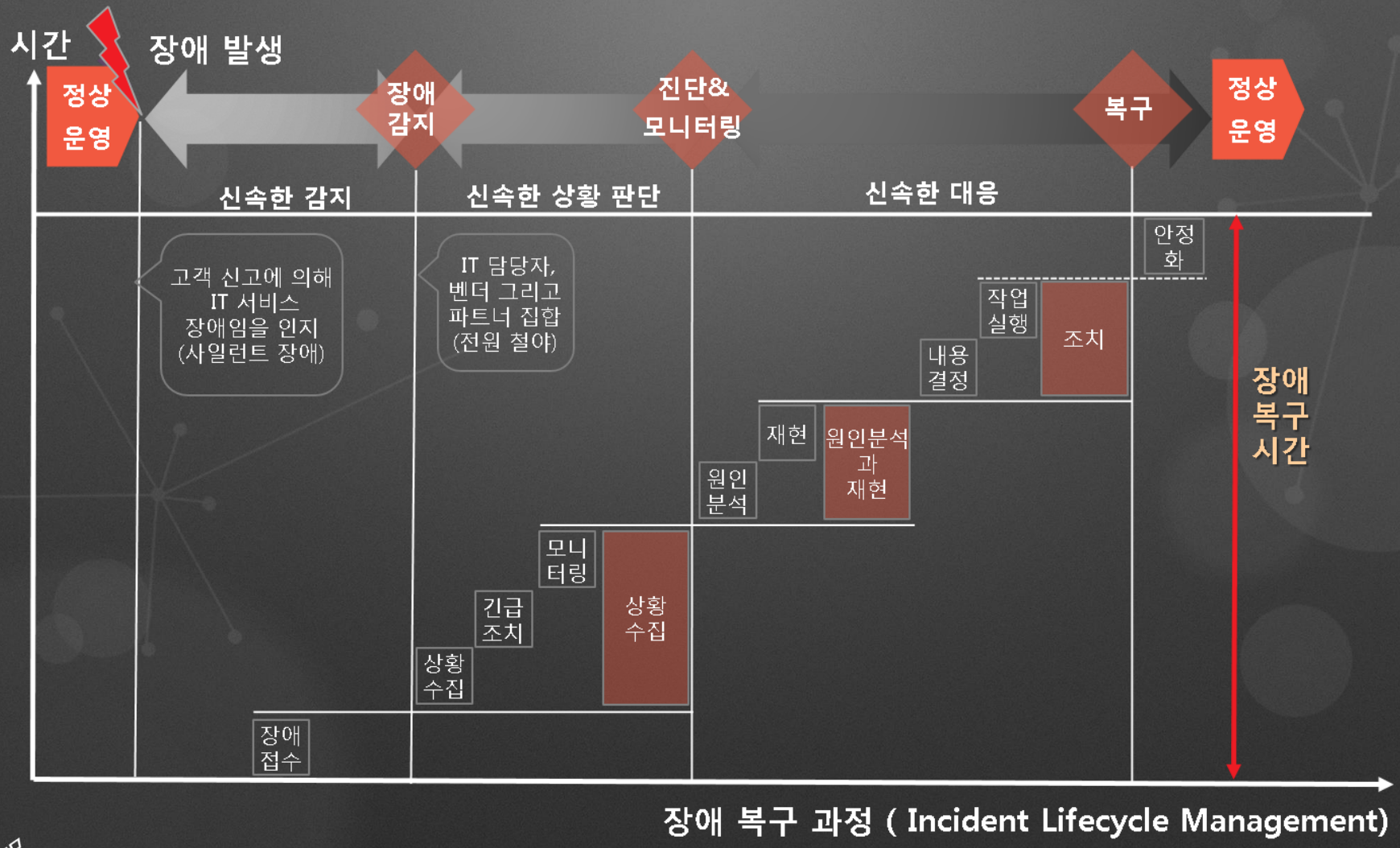


조치

방지대책

Incident & Problem Management

- 장애복구시간을 단축할 수 있는 도구는 ?



증상과 원인 - 장애 내용 파악

opennaru

장애파악 → 정보파악 → 자료수집 → 문제구간식별 → 조치 → 방지대책

- 장애 내용 및 상황 확인
  - 장애 발생 과정
  - 시스템 정지 여부
  - 장애 발생 빈도
  - 장애 발생 조건
  - 장애 재현 가능 여부
  - 장애 재현 절차
  - 회피 방법 유무

증상과 원인

증상과 원인 - 현황 파악

opennaru

장애파악 → 정보파악 → 자료수집 → 문제구간식별 → 조치 → 방지대책

- 기본 정보 파악
  - JBoss EAP 버전
  - OS 종류/버전
  - JVM 종류/ 버전
  - 데이터베이스 종류/ 버전
  - etc...
- 설정 파일 파악
  - JBoss 설정 파일(domain.xml, host.xml, standalone.xml 등)
  - JBoss 시작 스크립트
  - etc...

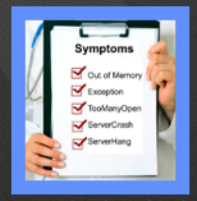
증상과 원인

증상과 원인 - 수집 자료는?

opennaru

장애파악 | 정보파악 | **자료수집** | 문제구간식별 | 조치 | 방지대책

- **기본 수집 자료**
  - 로그!
  - Thread dump!
  - 기본 환경 설정 정보(버전, 설정, 스크립트)
- **관련된 상황은?**
  - 언제
  - 어떤 서버, 어느 인스턴스에서 정확한 OS, JVM, WAS 버전은?
  - 발생 주기는?
  - 재현은 되는지?
- **어떤 영향을 미치는지?**
  - Business 부서의 영향은?
- **관련된 네트웍, H/W 및 S/W는 어떤 것인지?**
  - 대상 시스템에 로그나 상황은 어떤지 체크



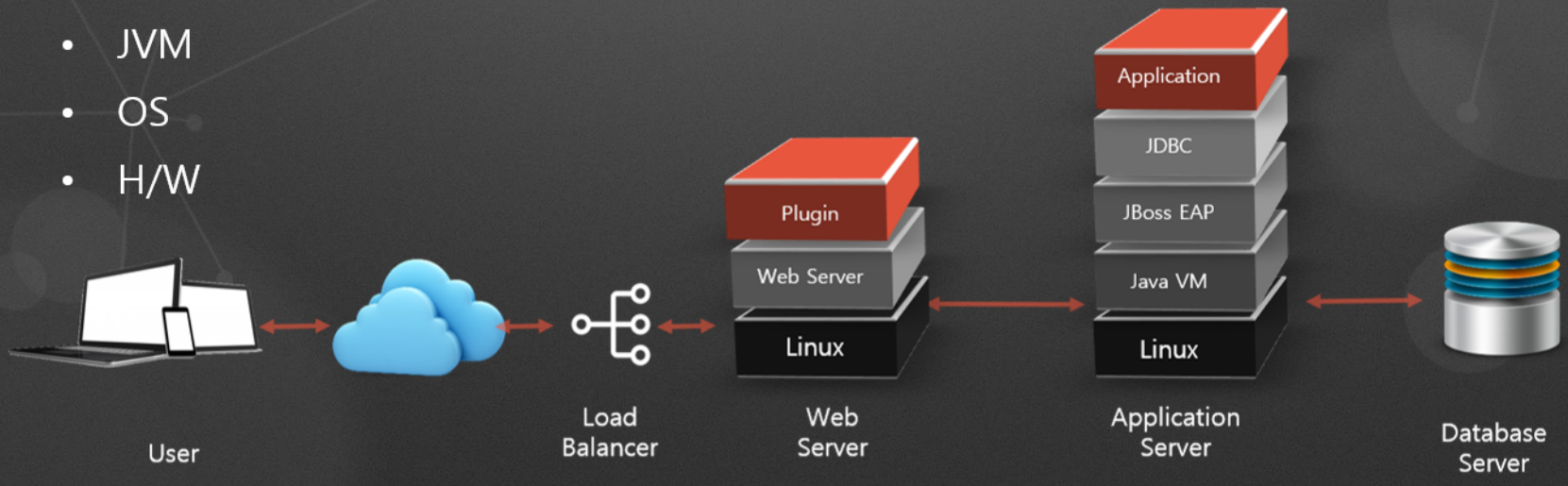
증상과 원인

진단 및 모니터링 - 정보수집

opennaru

장애파악 → 정보파악 → 자료수집 → **문제구간식별** → 조치 → 방지대책

- **문제가 발생 구간 파악 및 간단한 샘플 재현**
  - Application
  - JBoss
  - Database
  - Web Server
  - JVM
  - OS
  - H/W



조치

opennaru

장애파악 → 정보파악 → 자료수집 → 문제구간식별 → **조치** → 방지대책

- **조치 내용**
  - 설정 변경
  - 어플리케이션 수정
  - 버전업
  - Patch
  - etc...
- **조사 방법**
  - 재현 스텝
  - 디버그 정보

조치

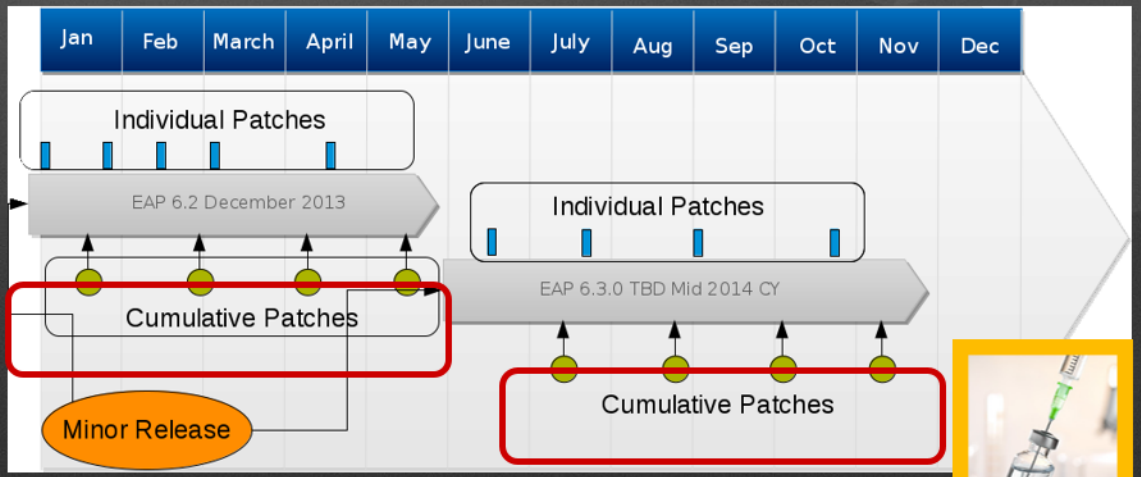
조치 - Patch

opennaru

장애파악 → 정보파악 → 자료수집 → 문제구간식별 → **조치** → 방지대책

- 패치 업데이트는 ZIP 및 EAP의 설치 기반의 RPM 형식으로 전달 되며, zip 형태의 패치 파일은 JBoss CLI 의 Patch 명령어를 통해 적용
- 적은 개별 패치들은 관리자로 하여금 관리 부담 감소
- 심각도가 높고 시간이 급한 케이스에 대한 버그 픽스는 기존 방법 대로 개별 패치들로 배포

- 여러 수정사항을 포함한 누적패치는 함께 테스트 되어 제품의 질을 향상시키고, 위험성을 감소



Source : EAP 6.2 이상부터 유지관리 배포방식 변경
https://access.redhat.com/site/ko/articles/645133

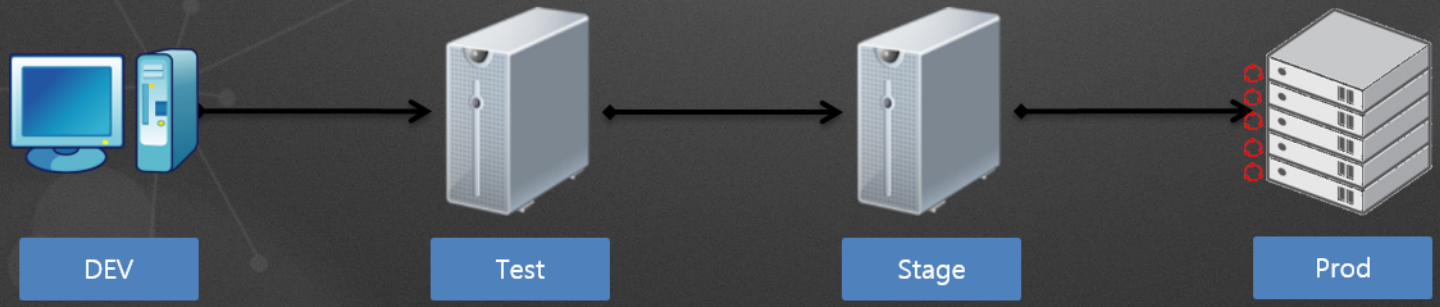
조치

방지 대책

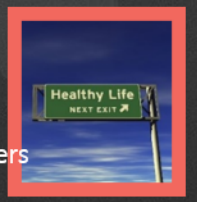
opennaru

장애파악 → 정보파악 → 자료수집 → 문제구간식별 → 조치 → **방지대책**

- 안정된 코드 테스트를 애플리케이션과 검증을 위한 환경 구축
- 서비스 품질을 높이기 위한 기본 프레임워크
- Subversion (SVN) 또는 GitHub
- 빈번한 커밋과 자동화된 빌드와 자동 피드백
- 테스트 프레임워크 적용



- Your box
  → Light integration
- Test
  → Continuous integration
- Stage
  → Build it, isolate it, test it.
- Production
  → Your customers are your integration testers



방지대책

트러블 슈팅의 기본

opennaru

- **로그 등 관련 자료는 충분히 확보해야**
  - 로그는 첫 번째 문제부터 살펴봐야
- **섣부른 추측은 금물**
  - 전해들은 말을 100% 믿지 말라
  - 증거에 의존하여 분석
  - 문서를 100% 믿지 말라
- **문제분석**
  - 문제를 분리하는 방법(Divide & Conquer)
  - 코드 샘플링을 통한 분리 / 재현
  - 재현 & 테스트 & 재현 & 테스트

opennaru

Application Performance Management

서버가 느려질 때 분석방법
스레드 덤프 분석

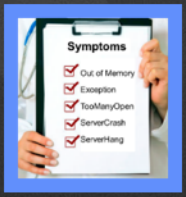
KHAN apm

Server Hang

opennaru

- 현상
  - 운영 중인 서버의 애플리케이션에서 새로운 요청을 받지 못함
  - Request가 타임 아웃되며, 응답시간이 점점 느려짐.

- 주된 원인
  - 애플리케이션의 설계나 구현의 버그
  - 잘못된 용량 산정(Capacity Planning)
  - 모든 스레드가 처리 중이어서, 새로운 요청을 받지 못하는 상태
    - 처리시간이 오래 걸리는 작업의 수행
    - 데이터베이스와 같은 외부 자원의 응답 대기
    - 자원의 경합
  - Full GC 시간이 오래 걸림
    - 튜닝 부족이나 적합치 않은 GC 알고리즘 선택
  - JVM이나 미들웨어의 버그



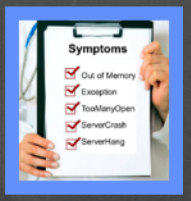
증상과 원인

Server Hang이란?

opennaru

- 서버가 전혀 Response를 주지 못함 ➔ 서버 Hang
- 느리더라도 Response를 준다 ➔ 서버 Slowdown

- 서버 Hang의 증상
  - Request가 처리되지 않는다
  - 새로운 Request를 받지 못하고 기존 Request가 Timeout되거나 응답이 없는 상황
  - 서버가 아무 작업도 하지 않는 것 같다
  - 서버의 증상:
    - Hang이 오래 지속되면 JVM Crash될 수도 있다(항상은 아님)
    - 그냥 두면 Hang이 풀리는 경우도 있다(리소스 경합이 발생하여 Hang이 발생한 경우에는 Resource가 풀리면 Hang이 풀린다)
    - 대부분 Hang이 발생된 후에는 관리자의 조치가 없으면 Hang 상태로 무한이 남아있게 된다

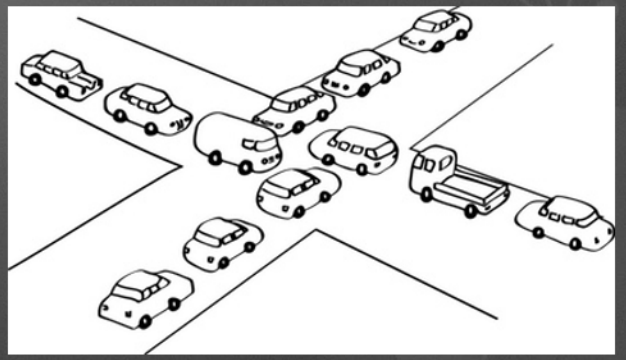


증상과 원인

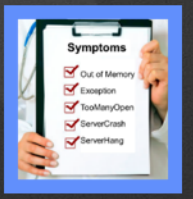
Server Hang의 원인

opennaru

- 리소스 부족
  - 스레드나 메모리 부족 → CPU 과점유나 OutOfMemory 에러발생
  - 파일 핸들이 부족한 경우 → Too many Open Files 에러가 발생
  - 리소스 경합이나 DeadLock이 발생하는 경우
    - JDBC Dead Lock이 발생하는 경우
    - JNDI Lookup에서 경합이 발생하는 경우
    - Application Dead Lock
  - JSP 컴파일 주기가 짧아서 발생하는 Hang



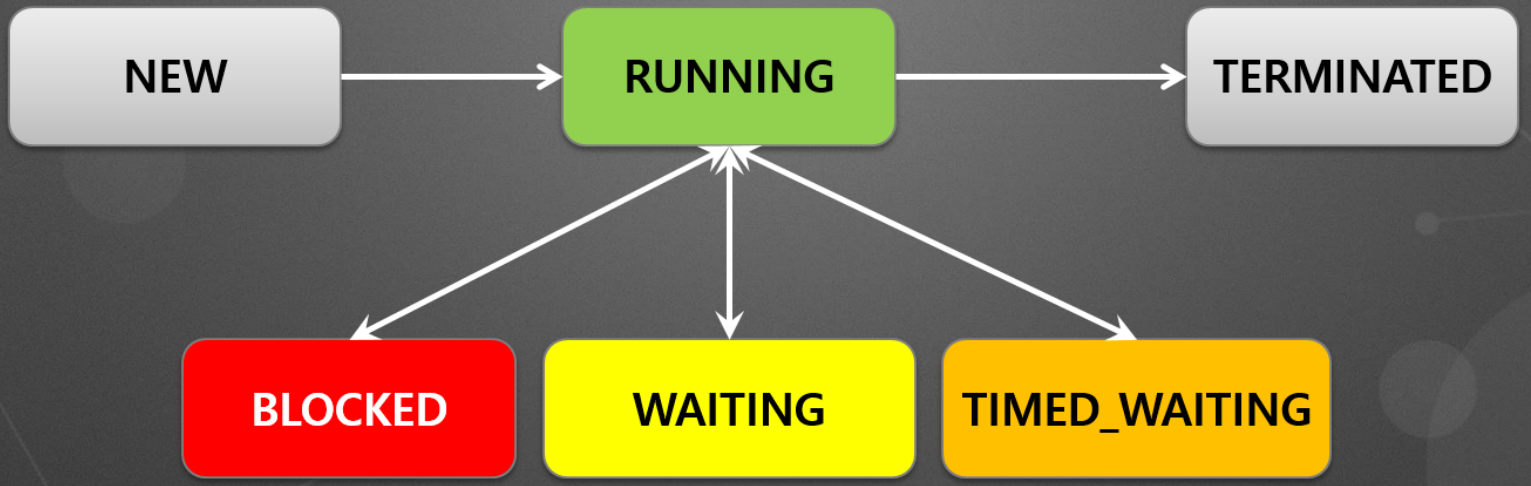
- Hang 처럼 보이는 경우
  - 리소스 경합의 경우 리소스가 가용해지면 조금씩 풀리는 경우
  - Full GC 시간이 오래 걸리는 경우
  - JVM에서 Code Optimization시 Hang이 걸리는 경우



증상과 원인

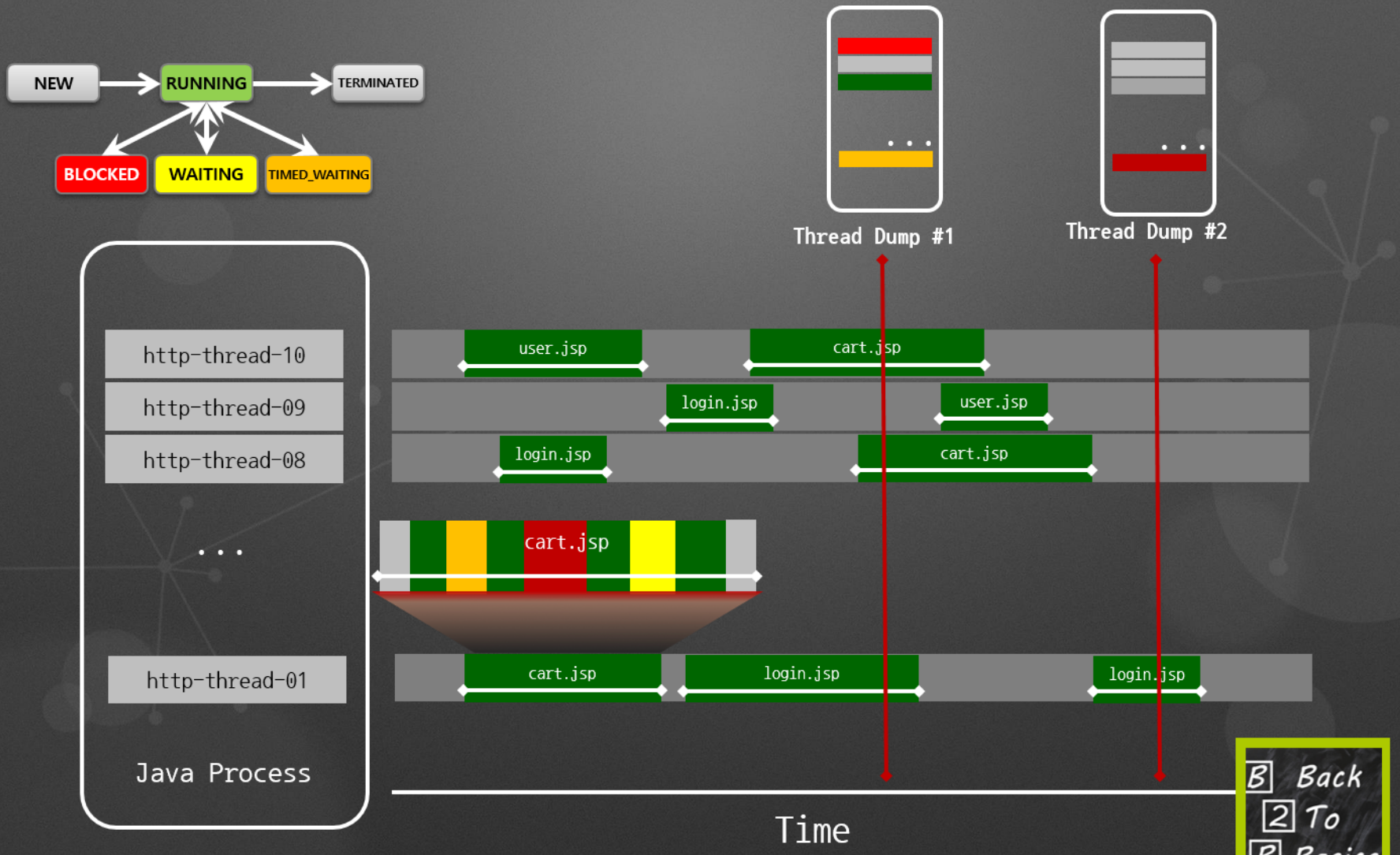
Thread 상태

- java.lang.Thread.State



- NEW – 생성되어있지만 실행되지 않은 상태
- RUNNING – 작업 수행 중
- BLOCKED – 다른 스레드의 Lock 해제를 기다리는 중
- WAITING – wait(), join(), park() 메소드로 대기 중
- TIMED_WAITING – WAITING과 같지만, 정해진 시간만 대기
- TERMINATED – 종료됨

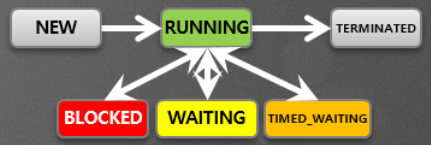
Java Thread Dump의 이해



Back 2 To Basics
개념과 배경지식

스레드 덤프의 정보 구성

- 스레드 덤프의 각 스레드의 정보 구성



스레드 이름 우선순위 스레드ID OS 스레드ID 스레드 상태

```
"http-thread-pool-threads - 250" prio=10 tid=0x00007fab9c192800 nid=0xdbc9 waiting on condition [0x00007faa6ce2c000]
   java.lang.Thread.State: WAITING (parking)
            at sun.misc.Unsafe.park(Native Method)
            - parking to wait for  <0x000000078b428170> (a java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
            at java.util.concurrent.locks.LockSupport.park(LockSupport.java:186)
            at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(AbstractQueuedSynchronizer.java:2043)
            at java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
            at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1068)
            at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1130)
            at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
            at java.lang.Thread.run(Thread.java:744)
            at org.jboss.threads.JBossThread.run(JBossThread.java:122)
```

스레드 호출 Stack

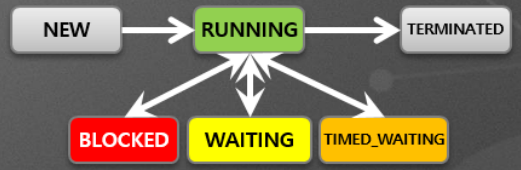
Back 2 To Basics
개념과 배경지식

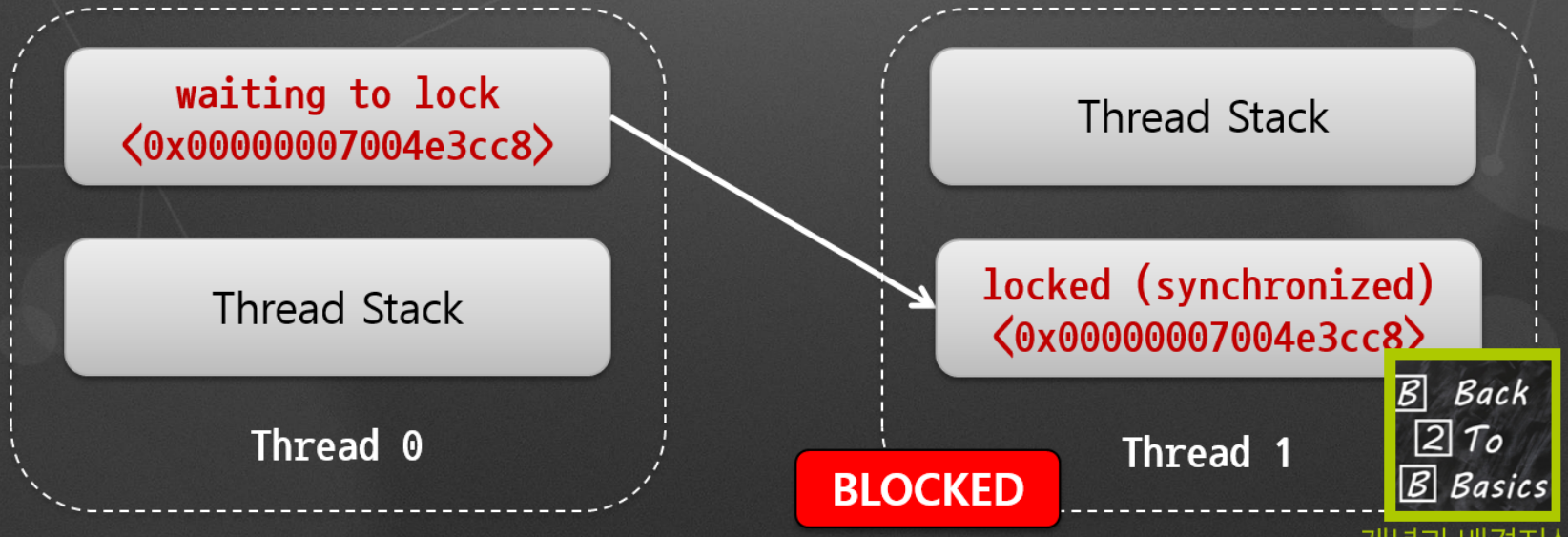
Thread Dump의 Blocked의 의미

- 여러 스레드에서 같은 변수나 메소드를 변경할 경우
- synchronized로 다른 스레드의 처리가 끝날 때 까지 제어가 넘어가지 않도록 보호

```
// 특정한 객체에 lock을 걸고자 할 때
synchronized(객체의 참조변수){
    // ...
}

// 메서드에 lock을 걸고자할 때
public synchronized void methodName(){
    // ...
}
```

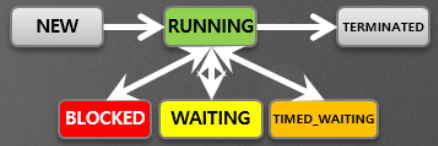


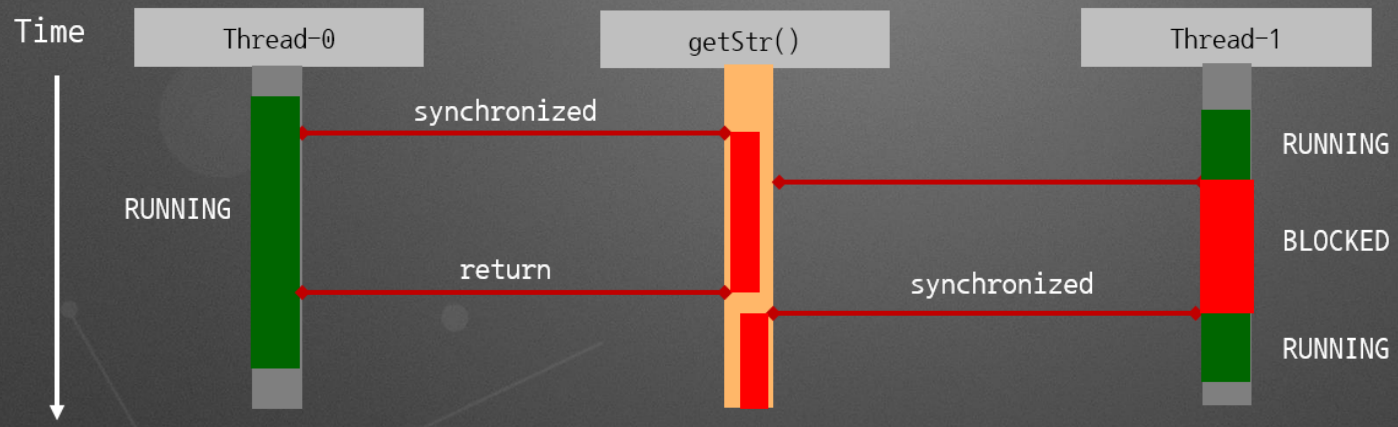


개념과 배경지식

Java Thread Dump의 BLOCKED의 의미

- waiting for monitor entry
  - 오브젝트에 대한 락의 해제를 기다리고 있는 상태

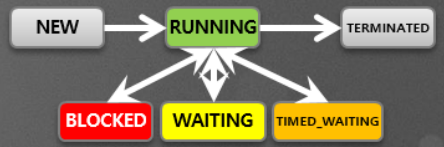


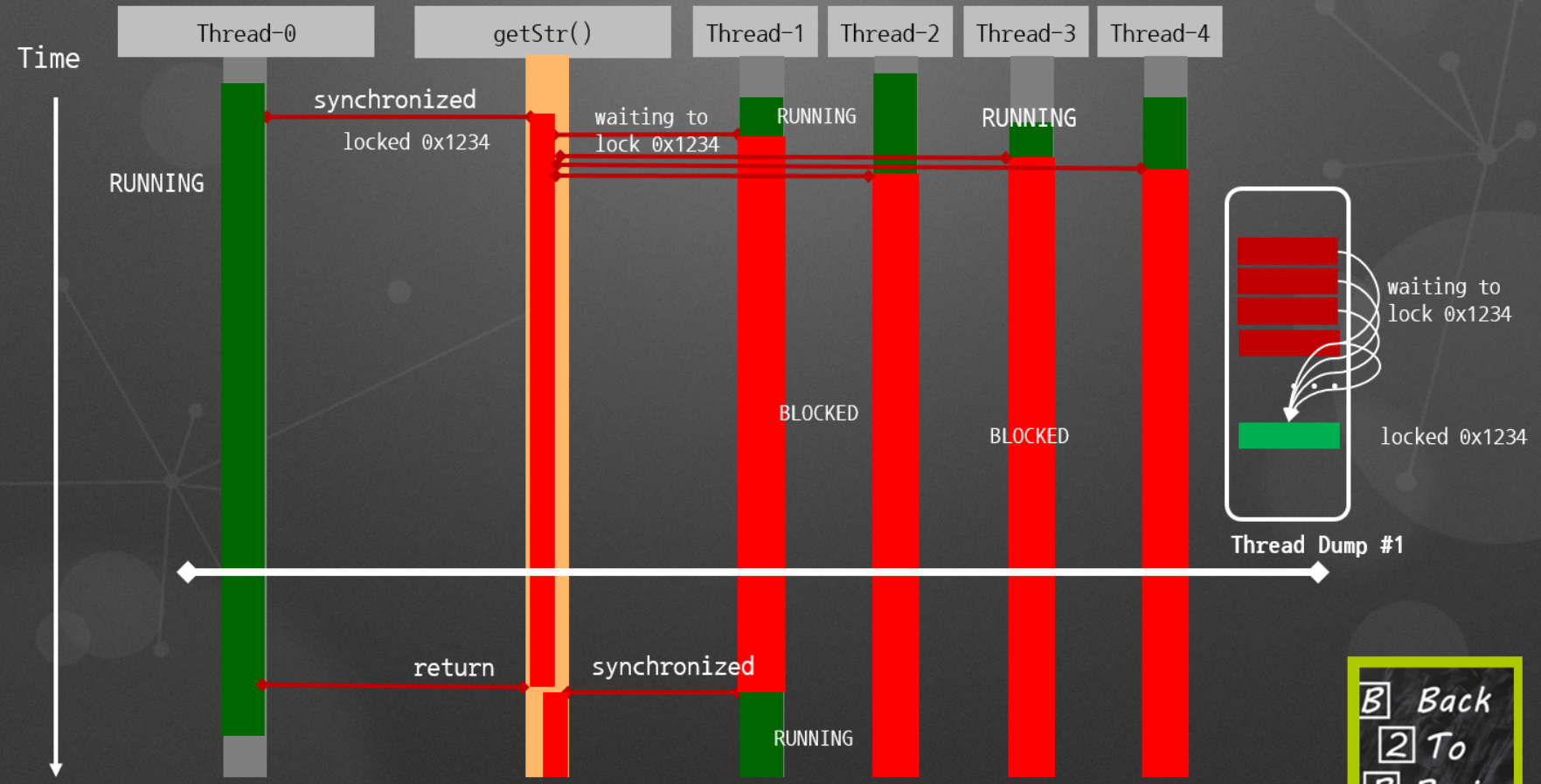



```
"Thread-1" prio=10 tid=0x00007f69fc0b8800 nid=0x7358 waiting for monitor entry [0x00007f6a0050a000]
   java.lang.Thread.State: BLOCKED (on object monitor)
        at sample.SampleMain.getStr(SampleMain.java:23)
        - waiting to lock <0x000000077acacf60> (a java.lang.Class for sample.SampleMain)
        at sample.SampleMain.run(SampleMain.java:17)
        at java.lang.Thread.run(Thread.java:679)
"Thread-0" prio=10 tid=0x00007f69fc0b6800 nid=0x7357 waiting on condition [0x00007f6a0060b000]
   java.lang.Thread.State: TIMED_WAITING (sleeping)
        at java.lang.Thread.sleep(Native Method)
        at sample.SampleMain.getStr(SampleMain.java:23)
        - locked <0x000000077acacf60> (a java.lang.Class for sample.SampleMain)
        at sample.SampleMain.run(SampleMain.java:17)
        at java.lang.Thread.run(Thread.java:679)
```

Thread BLOCKED 상태가 길어지면 발생하는 상황

opennaru





Back 2 To Basics
개념과 배경지식

Application Performance Management

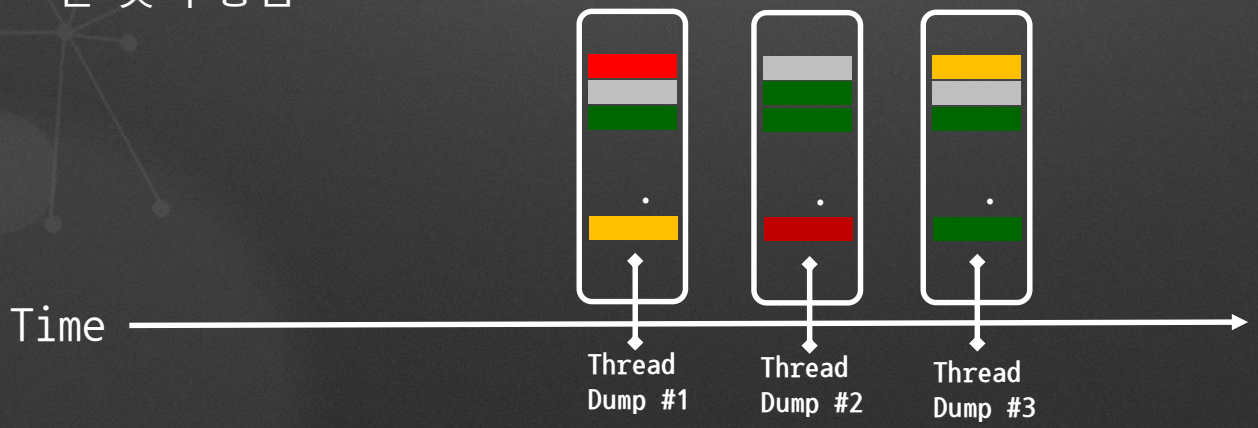
Thread Dump 분석 방법

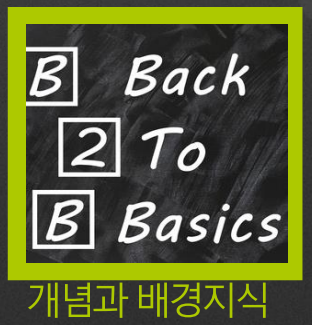
KHAN
[apm]

스레드 덤프

- **스레드 덤프란?**
  - JVM 내부에서 현재 수행중인 스레드의 SnapShot을 텍스트로 출력
  - 서버 Hang, 서버 Slowdown등 서버에서 발생하는 문제를 해결하는 가장 유용한 도구
  - 약 3~5초간격으로 3~5회 연속해서 스레드 덤프를 받아야 분석이 가능하다

- **언제 받아야 하나?**
  - 운영 중인 서버에서도 수행가능(많은 시간이 걸리지 않는 작업임)
  - 서버 Hang, Slowdown시, 서버가 Crash되거나 OutOfMemory가 발생한 직후에 받는 것이 좋음





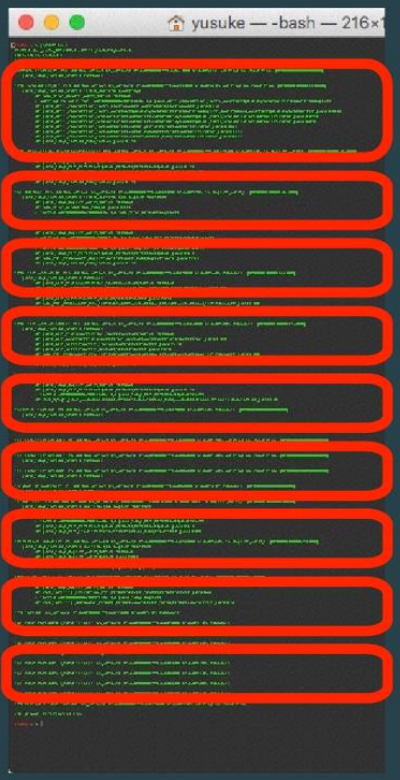
스레드 덤프를 분석

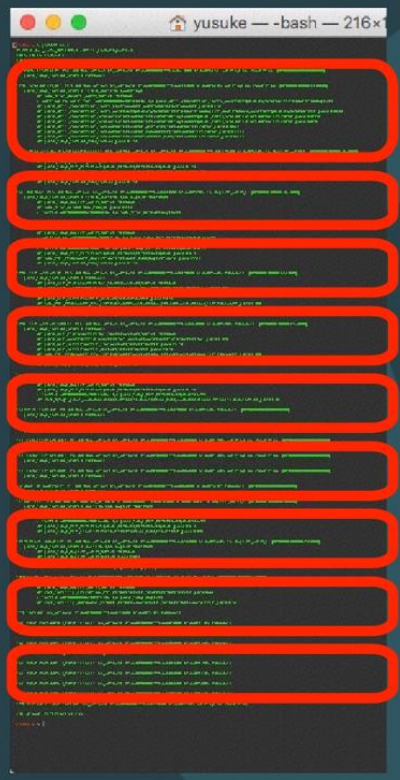
opennaru

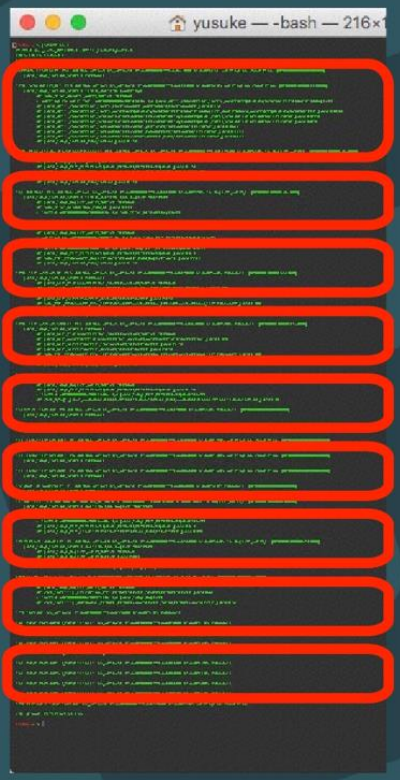
Take at least 3 times, with 3 to 5 seconds interval

진단 및 모니터링

Thread 1
Thread 2
Thread 3
Thread 4
Thread 5
Thread 6
Thread 7
Thread 8
Thread 9

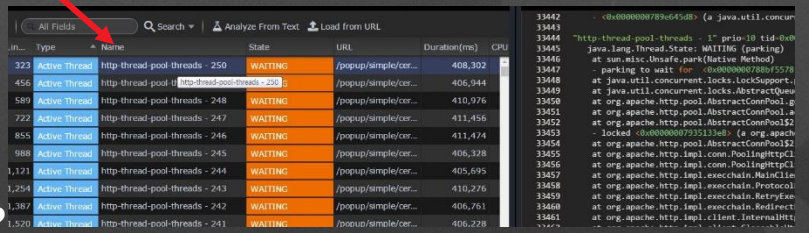






스레드 덤프 받는 방법은?

- **Linux / UNIX 시스템에서 스레드 덤프**
  - kill -3 <*PID*>
  - 일반적을 nohup 명령을 이용하여 시작하기 때문에 nohup.out파일에 stdout 로그가 남게 됨
  - 윈도우에서는 Ctrl + Break
  - **KHAN [apm] 화면에서 버튼 클릭(장애시 자동 생성)**



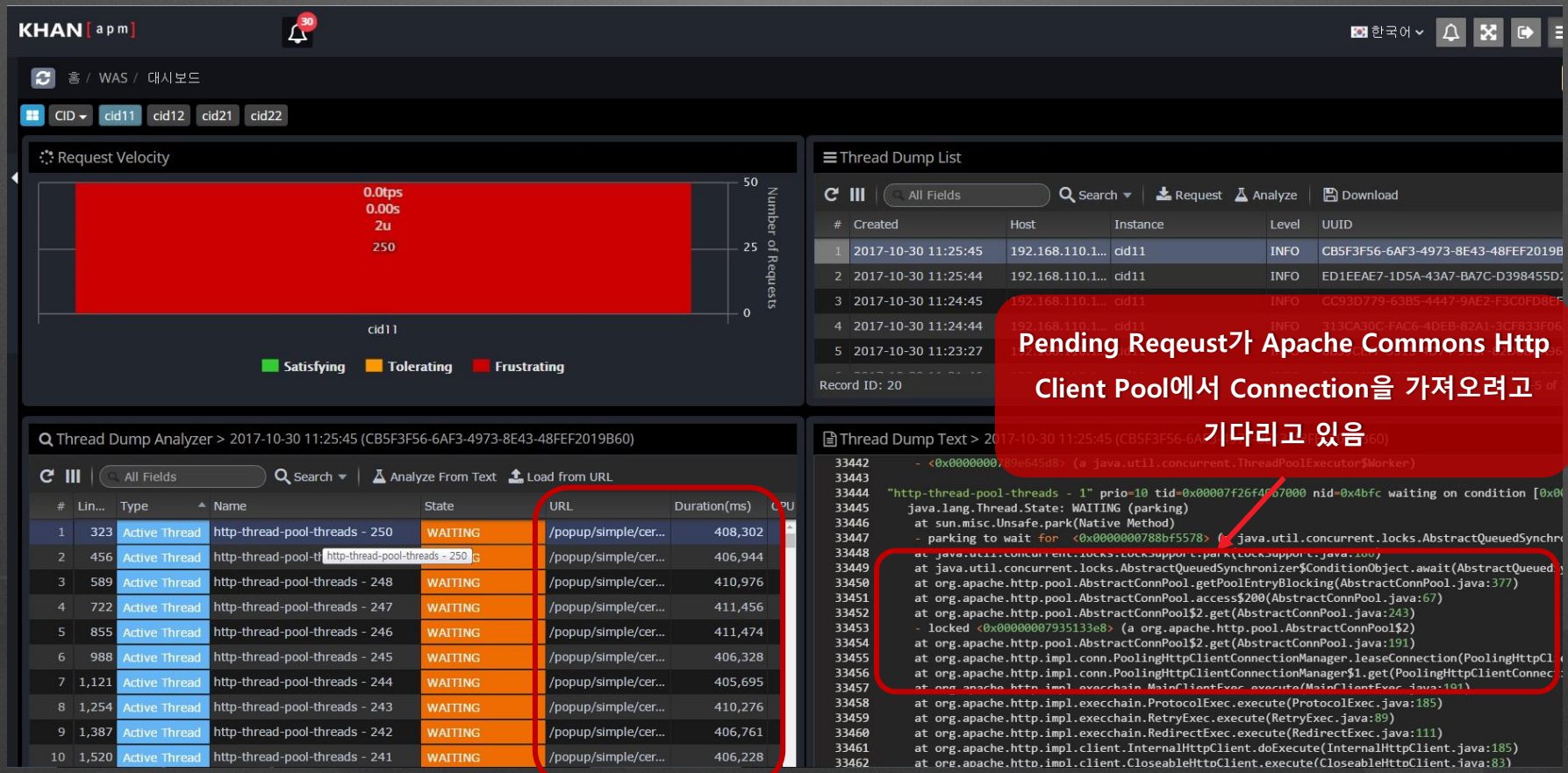
- **Docker와 Openshift 같은 환경**
  - **스레드 덤프을 어떻게 받나?**
    - docker exec -it c4343003c1 /bin/bash
      ~# kill -3 <PID>
  - **내용을 어떻게 복사? nohup 로그는 남기나?**
    - docker cp c4343003c1:/file/path/nohup.out /host/path/target
  - KHAN [apm]은 Openshift, Docker를 지원 → 버튼 클릭 & 분석 UI 제공

Server가 느려 지는 상황에서 Dashboard

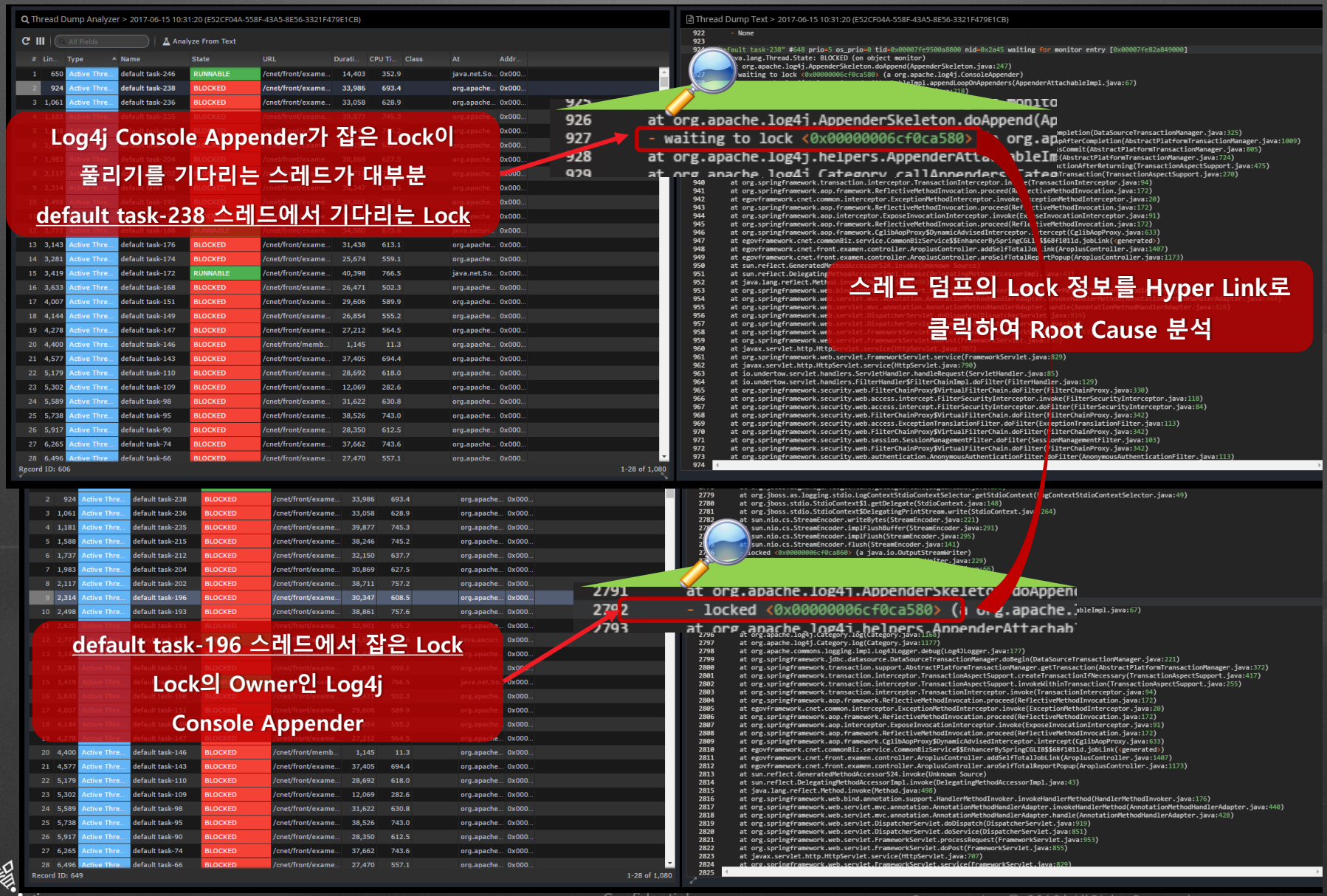
opennaru

KHAN [apm] 스레드 덤프 분석 예시

opennaru



KHAN [apm] 스레드 덤프 분석 예시



Opennaru Quick Service 란?

Opennaru Quick Service 란?

opennaru

고객사

1 시스템 이슈발생

2 쓰레드 덤프와 모니터링 화면 전달

3 오픈나루 서비스 데스크 접수

4 장애 분석

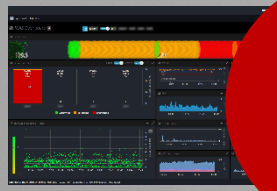
5 장애 분석 보고서 전달

오픈나루

KNOWLEDGE

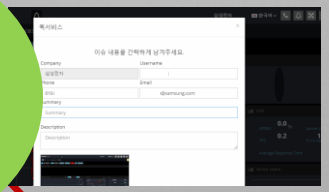
Quick Service ON

opennaru



문제발생

문제보고



고객사

문제해결

Quick Service ON

QuickService
ON Desk
자동 접수

오픈나루
Knowledge
Base

문제분석

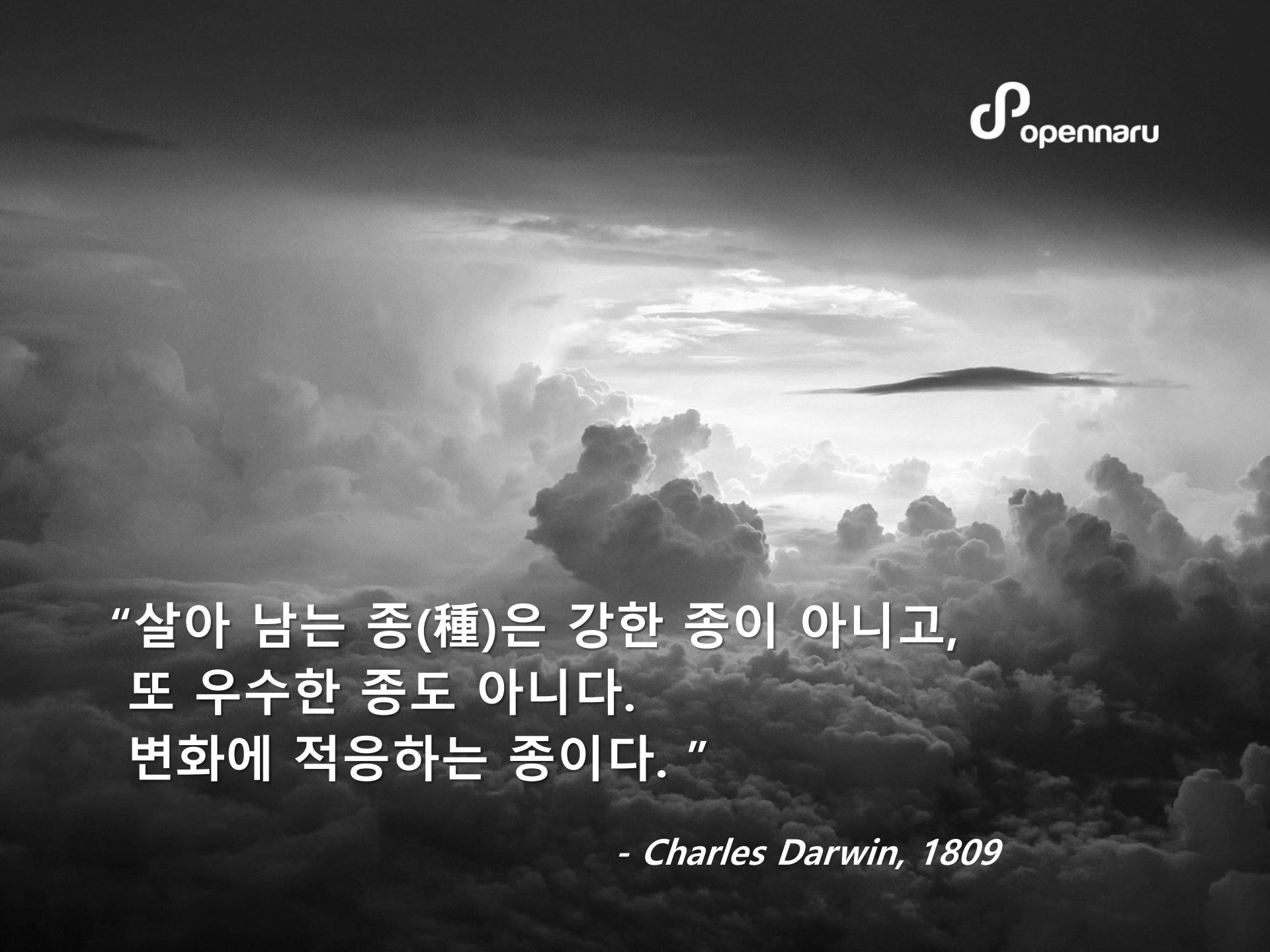
기술지원
엔지니어

opennaru

KHAN [apm] = Ah ! Provisioning + Monitoring

opennaru

설치	구성	튜닝	모니터링	진단/조치
•리눅스만 설치되어 있으면 WAS 와 웹서버 등 웹시스템에 필요한 소프트웨어를 자동설치	•WAS 클러스터링, 데이터그리드 , 필수 운영 스크립트 등 난이도 높은 구성을 지원 •개발자/관리자를 위한 운영 가이드 자동 생성	•전문가 수준의 튜닝 (리눅스 커널, 웹서버/JVM/WAS 튜닝)	•시스템과 OS 에 대한 모니터링 뿐만 아니라 Java 나 WAS 까지 포괄적으로 지원	•Threaddump , 네트워크 상태, 웹서버 상태 등을 진단할 수 있는 도구 제공


opennaru
“살아 남는 종(種)은 강한 종이 아니고,
또 우수한 종도 아니다.
변화에 적응하는 종이다. ”
- Charles Darwin, 1809

opennaru

감사합니다.

제품이나 서비스에 관한 문의

콜 센터 : 02-469-5426 (휴대폰 : 010-2243-3394)

전자 메일 : sales@opennaru.com