

Apache Tomcat 통합 관리 솔루션

# Athena Meerkat

공개 SW 개발자 Lab 오픈소스프론티어 2기 박상천

웹 애플리케이션 서버(Web Application Server, 약자 WAS)는 인터넷 상에서 HTTP를 통해 사용자 컴퓨터나 장치에 애플리케이션을 수행해 주는 미들웨어(소프트웨어 엔진)이다. 웹 애플리케이션 서버는 동적 서버 콘텐츠를 수행하는 것으로 일반적인 웹 서버와 구별이 되며 주로 데이터베이스 서버와 같이 수행된다. 국내에서는 일반적으로 "WAS"로 통칭하고 있다.

아파치 톰캣(Apache Tomcat)은 전세계적으로 가장 많이 사용하는 WAS(Web Application Server) 제품(Plumbr 2015년 조사 58.6%, RebelLabs 2014년 조사 50%)이다. 본 글에서 설명하고자 하는 Athena Meerkat은 이러한 아파치 톰캣의 통합 관리 솔루션으로써 설치에서부터 구성, 세션 클러스터링, 로그 분석 및 모니터링까지 모든 과정을 웹 UI를 통해 쉽게 관리할 수 있다.

이 글을 통하여 아파치 톰캣의 통합 관리 솔루션인 Athena Meerkat의 기능 및 사용 방법에 대해 설명하도록 하겠다.

## [ 목차 ]

- 1 웹 애플리케이션 서버(Web Application Server, WAS)
- 2 WAS 도입효과 및 기술표준
- 3 오픈소스 WAS의 강자 - 아파치 톰캣(Apache Tomcat)
- 4 아파치 톰캣의 약점
- 5 도메인 기반 아파치 톰캣 관리를 위한 도구 - Meerkat
  - 5.1 세션 클러스터링(clustering) 지원
  - 5.2 프로비저닝(provisioning) 및 설정 관리 기능
  - 5.3 실시간 모니터링 대시보드
- 6 요약

## 1 웹 애플리케이션 서버(Web Application Server, WAS)

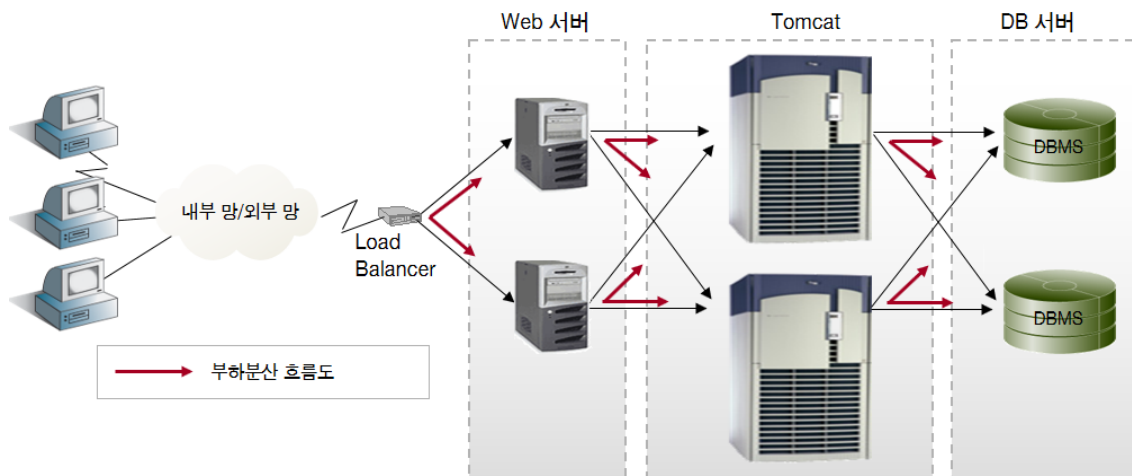
웹과 기업의 기간 시스템 사이에 위치하면서, 웹 기반 분산 시스템 개발을 쉽게 도와주고 안정적인 트랜잭션 처리를 보장해 주는 일종의 미들웨어 소프트웨어 서버다. 보통 3계층 웹 컴퓨팅 환경에서 기존 클라이언트/서버환경의 애플리케이션 서버와 같은 역할을 하며, 클라이언트와 서버 환경에서 트랜잭션 처리 및 관리와 다른 기종 시스템 간의 애플리케이션 연동 등을 주된 기능으로 한다.

WAS는 웹이 탄생한 이래, 주로 데이터베이스 조회나 일반적인 비즈니스 로직에 대한 처리를 위해 다양한 언어로 개발된 인터넷/인트라넷환경의 소프트웨어를 지칭한다. 자바스크립트나 JSP등과 같은 스크립트 및 서비스들은 대개 최신의 데이터를 검색하기 위해 데이터베이스에 접근하고, 브라우저 또는 클라이언트 프로그램을 통해 사용자들에게 검색 결과를 제공한다.

WAS를 비롯한 애플리케이션 서버들은 웹 서버, 즉 HTTP 서버와 같은 컴퓨터를 공유할 수도 있지만, 별개의 컴퓨터를 독립적으로 사용하는 경우도 많다. 대규모 사이트에서는 오히려 WAS와 웹 서버등을 위해 여러 대의 컴퓨터가 동원되기도 한다. 넷스케이프의 Netscape Application Server, 오라클의 웹로직, IBM의 웹스피어 애플리케이션 서버, 티맥스소프트의 Jeus 등이 국내에서 사용되는 대표적인 WAS이며, 가장 많이 사용되는 오픈 소스 기반의 애플리케이션 서버는 아파치 톰캣이 있다.

## 2 WAS 도입효과 및 기술표준

WAS를 도입하게 되는 경우 안정적인 시스템 구성, 애플리케이션에 대한 로드밸런싱과 클러스터링을 통한 고가용성(High Availability) 시스템을 구축할 수 있으며, DB 통신의 커넥션 풀 제공을 통해 성능 보장, 트랜잭션 관리, 스레드 조정을 수행할 수 있다.



[그림 1] 로드밸런싱/클러스터링을 통한 고가용성 웹 아키텍처의 구현

또한 자바 기반의 WAS 표준을 활용하여 어떠한 애플리케이션 서버 상에서도 작동이 될 수 있는 업무 시스템을 만들 수 있으며, 프레젠테이션 로직/비즈니스 로직을 분리하여 개발을 원

활하게 할 수 있는 기반을 제공한다.

### 3 오픈소스 WAS 의 강자 - 아파치 톰캣(Apache Tomcat)

아파치 톰캣은 아파치 소프트웨어 재단에서 개발된 서블릿 컨테이너(또는 웹 컨테이너)만 있는 웹 애플리케이션 서버이다.

톰캣은 웹 서버와 연동하여 실행할 수 있는 자바 환경을 제공하여 자바 서버 페이지(JSP)와 자바 서블릿이 실행할 수 있는 환경을 제공한다. 톰캣은 기본적으로 내장하고 있는 관리 툴을 통해 설정을 변경할 수 있지만, 이를 통한 작업은 상당히 단순하여 상용 웹 애플리케이션 서버들이 가지고 있는 관리자 기능과 상당히 많은 차이점을 보이고 있다. 대부분의 서버에 대한 설정은 XML 파일을 편집하여 작업을 하고 있다.

아파치 톰캣은 Apache Licence Version 2를 채용한 오픈소스 소프트웨어로서, 서블릿이나 JSP를 실행하기 위한 서블릿 컨테이너를 제공하며, 상용 웹 애플리케이션 서버에서도 서블릿 컨테이너로 사용하는 경우가 많다. 버전 5.5 이후는 기본적으로 Java SE 5.0 이후를 대응하는 형태로 발전되었다.

### 4 아파치 톰캣의 약점

아파치 톰캣은 전세계에서 많은 사용량을 보이는 서버로 클라우드가 대두된 요즘, 훨씬 더 각광을 받는 WAS가 되어가고 있다. 비용 효율을 고려하여 클라우드로 전환하는 시스템이 값비싼 상용 웹 애플리케이션 서버를 활용하기는 어렵기 때문이다.

점차 사용량이 많아지는 톰캣이지만 여전히 극복이 안 되는 부분들이 있는데 이는 아래와 같다.

1. 모든 설정을 server.xml, context.xml 등의 xml 설정 파일을 활용하여 수정해야 한다.
2. 세션 클러스터링에 대한 부분은 안정성에 대한 의심으로 많은 사이트에서 사용하지 않고 있다.
3. 한 서버에 여러 대의 톰캣 인스턴스를 구동하는 경우 포트 관리, 설정 관리, 엔진 관리 등이 어렵다.
4. 톰캣이 가진 JMX 기능을 통해 모니터링을 할 수 있는 부분이 한정적이다.

5. 관리자의 설정 변경 시 히스토리 관리가 제대로 되지 않아 문제를 일으킬 수 있는 소지가 있다.

위에서 나열한 사항들은 톰캣을 관리하는 관리자들이라면 한 번쯤은 부딪혀본 문제들이일 것이다.

## 5 도메인 기반 아파치 톰캣 관리를 위한 도구 - Meerkat

앞서 언급한 관리의 어려움, 잠재적인 문제점을 관리자가 손쉽게 해결할 수 있도록 돕기 위한 도구가 필요하게 되었으며 이를 해결할 수 있는 도구가 바로 아테나 톰캣 매니저(코드명 Meerkat)이다.

많은 수의 톰캣 서버들을 운영하는 관리자들은 운영 시스템 전체의 상태를 한 눈에 보고 싶어한다. Athena Meerkat은 톰캣 서버의 상태(스레드, 힙메모리, 데이터소스 등 JMX 기반 정보)는 물론 운영체제의 상태(CPU, 메모리, 디스크, 네트워크 등)에 대한 실시간 모니터링 정보를 ExtJS 기반의 다양한 차트로 표시한다.

또한, 톰캣이 제공하고 있지 않은 세션 클러스터링 기능을 Athena Dolly 제품을 통해 지원한다. Athena Dolly의 클러스터링 구성 과정도 웹 UI 기반으로 손쉽게 할 수 있다.

관리되는 모든 기능들 즉, 서버의 구성 변경, 시작, 중지 등의 작업은 히스토리로 관리되어 언제라도 확인할 수 있다.

데이터 소스 및 드라이버 등을 별도의 서버 수작업을 할 필요 없이 관리할 수 있으며, 자동으로 테스트용 페이지를 통해 확인이 가능하다.

아래에서는 내장하고 있는 기능에 대해 알아보겠다.

### 5.1 세션 클러스터링 지원

미어캣의 클러스터링 기능을 사용하게 되면 기존 웹 애플리케이션 서버가 가지고 있는 JVM 내장 클러스터링 처리로 인한 WAS 장애 전이, 인스턴스의 메모리 사용량 증가, WAN(Wide

Area Network) 기반 클러스터링의 확장 문제를 해결할 수 있다.

대량의 세션을 처리하기 위해서는 웹 애플리케이션 서버의 세션 부하를 하부의 별도 데이터 그리드 형태의 서버로 옮겨 고성능을 필요로 하는 시스템을 구성한다.

- **세션 메모리 부족 및 GC 문제에 대한 해결**

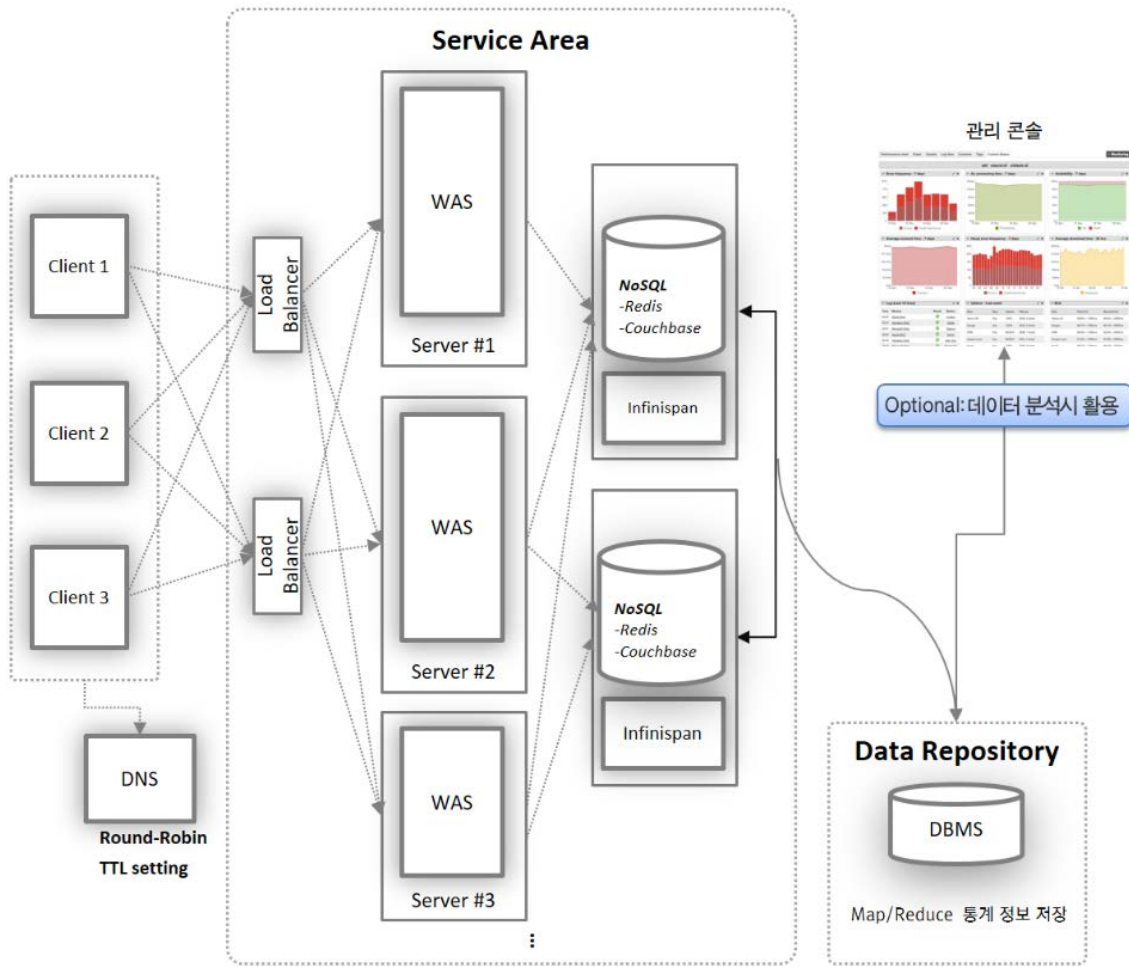
- ✓ WAS 에서 관리하던 세션 정보를 서버 노드로 옮김으로써 GC 발생을 최소화 하여 항상 일정한 서비스 응답을 제공
- ✓ WAS 의 메모리 부족에 의한 Out Of Memory 장애 발생 방지

- **안정성 및 고성능 보장**

- ✓ 웹 애플리케이션 서버의 세션 부하를 하부의 별도 서버로 옮김

분산 캐시 기능으로 병목구간의 해소와 애플리케이션 영역의 확장으로 고성능을 보장한다.

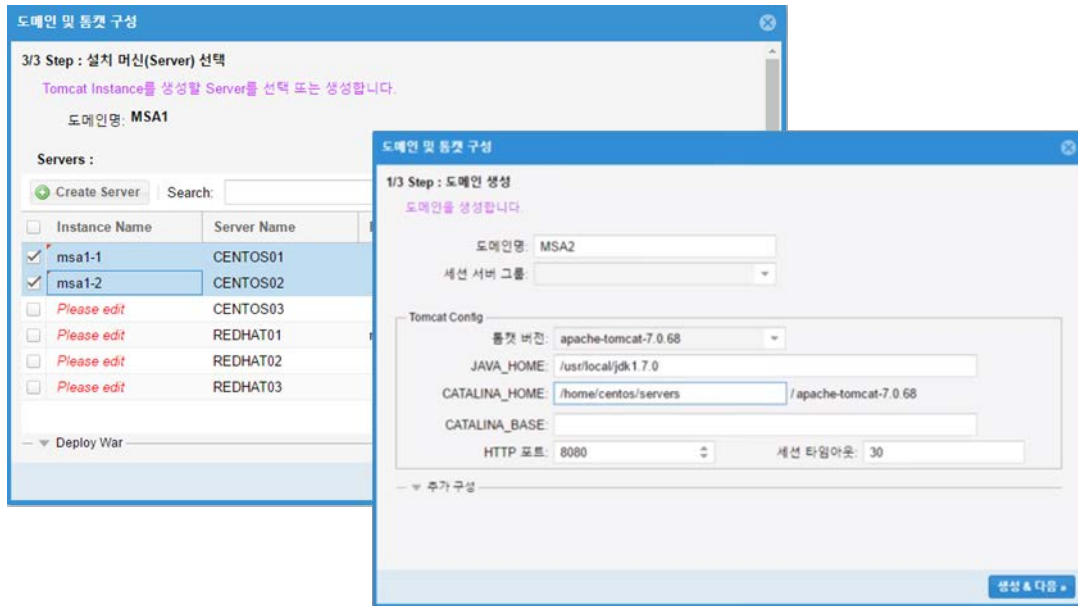
또한 웹 애플리케이션 서버의 세션 데이터 저장을 위해 데이터그리드, 캐시, NoSQL 솔루션 인 Infinispan, Redis, Couchbase를 지원하며 저장된 데이터에 대한 검색, 질의, 통계 정보를 위한 데이터 추출 기능을 제공한다. 대용량 데이터 처리 요청에 효과적으로 대응하기 위해 다음과 같은 아키텍처상의 특징을 가지고 있다.



[그림 2] 미어캣을 활용한 세션 클러스터링 아키텍처

## 5.2 프로비저닝 및 설정 관리 기능

프로비저닝의 의미는 특정 서비스를 제공받기 위하여 서비스 실행부터 시작해 서비스를 제공받기 전 단계까지 처리되는 일련의 절차를 말한다. 톱캣을 구성하기 위한 시스템 설정, 포트 환경 구성, 데이터 소스 등에 대한 설정이 그 작업 중의 일부이며 자동으로 엔진 설치, 환경 구성 등을 화면에서 제어할 수 있도록 기능을 제공한다.



[그림 3] 프로비저닝 및 설정 관리 기능

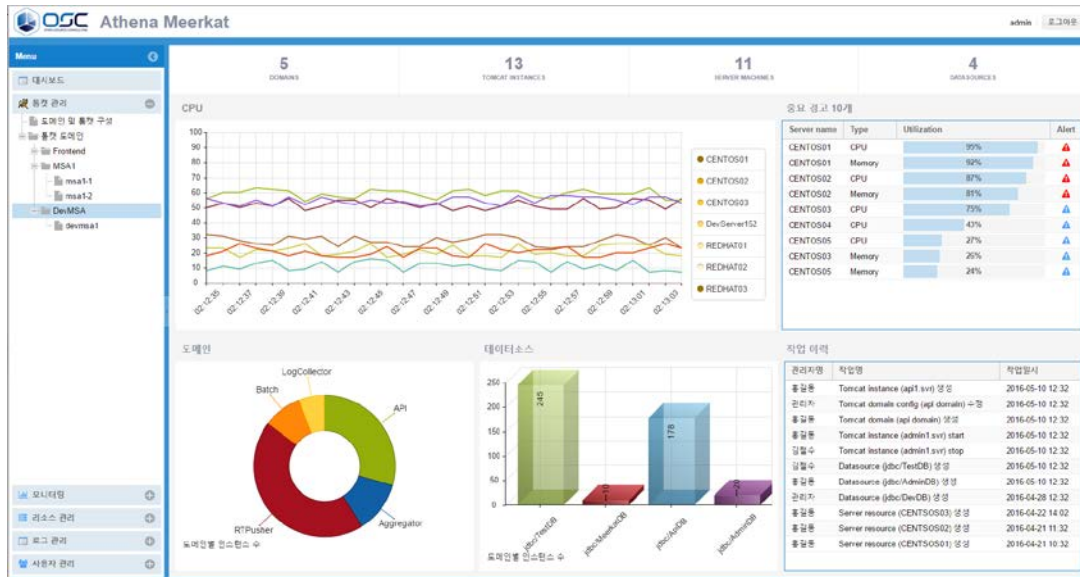
위의 기능을 이용하는 경우 복잡한 데이터 소스 설정, 스레드 설정, 드라이버에 대한 추가 등을 손쉽게 할 수 있도록 도와준다.

### 5.3 실시간 모니터링 대시보드

전체 관리 대상의 아파치 톰캣 시스템에 대한 실시간 상황 모니터링을 제공하며 경고 설정을 통해 문제가 될 수 있는 상태를 사전에 지정하고 해당 조건에 도달하는 문제 상황이 발생할 경우 화면에 표시는 물론 이메일, SMS 등을 통해 통지하는 기능을 수행한다

관리 대상 서버들의 각종 로그(톰캣 로그 및 시스템 로그)들을 중앙으로 수집하고 ElasticSearch를 이용하여 특정 메시지를 검색하거나 패턴을 분석하는 기능을 제공한다.





[그림 4] 톰캣에 대한 통합 모니터링 및 작업 이력 등의 정보

## 6 요약

데이터 센터의 관리 팀에서 웹 애플리케이션 서버를 관리하는 방법은 다양하다. 대부분의 엔터프라이즈에서는 상용WAS를 사용하고 문제가 생기면 이를 지원하는 벤더 혹은 파트너의 기술지원을 받는 것이 대부분이다. 하지만 현재 IT의 대부분은 클라우드로 점점 전환하는 추세이고, 이러한 클라우드는 유연성, 확장성, 비용절감 등을 기반으로 점차 그 사용량이 많아지고 있다.

TCO(Total Cost of Ownership, 총소유비용)적인 측면을 감안했을 때 미들웨어 역시 오픈 소스 기반으로 전환되고 있으며, 이를 위해서는 관리적인 도구가 필요한 시점이다. 이번 글에서 살펴보았던 Athena Meerkat은 많은 클라우드 기반의 WAS로 사용할 수 있는 아파치 톰캣에 대한 프로비저닝, 관리, 모니터링, 제어를 수행함으로써 관리자가 저지를 수 있는 실수를 최소화하고 안정적으로 시스템을 운영할 수 있는 환경을 제공한다.

### <Athena Meerkat 기능 요약>

#### 운영 환경을 빠르게 구축할 수 있는 프로비저닝(Provisioning)

프로비저닝 기능을 통해 동일한 구성을 가진 Tomcat 서버들을 손쉽게 여러 서버에 자동으로 설치하는 기능을 제공한다. 이를 통해 Tomcat 운영 환경을 빠르게 구성하는 것은 물론 구성을 일관되게 가져감으로써 관리의 어려움을 줄여준다. Tomcat 6부터 최신 버전까지 버전

별 설치 기능을 한다.

### **관리자들의 부담을 덜어주는 구성 관리**

많은 수의 Tomcat 서버들을 운영하는 것은 관리자들에게 큰 부담이 아닐 수 없다. Athena Meerkat은 웹 UI를 통해 Tomcat 서버와 그것이 동작하는 머신 및 애플리케이션에 대한 관리 기능을 제공한다. 자주 사용하는 설정에 대해서는 폼 기반 화면을 제공하여 관리자가 손쉽게 사용할 수 있고, 모든 설정 변경은 히스토리 저장으로 이전 버전과 비교 및 이전 설정으로 쉽게 되돌릴 수 있다.

### **세션 클러스터링을 통한 성능 확장**

다수의 사용자를 대상으로 한 서비스를 WAS에 운영하기 위해서는 클러스터링 기능이 필요하다. Athena Meerkat은 Tomcat뿐만 아니라 다양한 WAS 제품에 적용할 수 있는 세션 클러스터링 기능을 Athena Dolly 제품을 통해 지원한다. 이를 통해 Tomcat 간에 또는 Tomcat과 다른 WAS간의 세션 클러스터링을 구성할 수 있다. Athena Meerkat은 Athena Dolly의 클러스터링 구성 과정마저도 웹 UI 기반으로 손쉽게 할 수 있도록 만들었다.

### **서버의 상태를 한 눈에, 모니터링**

많은 수의 Tomcat 서버들을 운영하는 관리자들은 운영 시스템 전체의 상태를 한 눈에 보고 싶어한다. Athena Meerkat은 Tomcat 서버의 상태(스레드, 힙메모리, 데이터소스 등 JMX 기반 정보)는 물론 운영체제의 상태(CPU, 메모리, 디스크, 네트워크 등)에 대한 실시간 모니터링 정보를 ExtJS 기반의 다양한 차트로 표시한다.

### **시스템의 이상 상태를 알려 주는 경고(Alert) 기능**

Athena Meerkat은 경고 설정을 통해 문제가 될 수 있는 상태를 사전에 지정하고, 해당 조건에 도달하는 문제 상황이 발생할 경우 경고상황이 표시된다. API를 통해 업무 시스템과 연동하는 것도 가능하다.

### **누가 어떤 작업을 언제 했는지 알 수 있는 작업 이력 관리**

Athena Meerkat을 통해 관리 되는 모든 기능들 즉, 서버의 구성 변경, 시작, 중지 등의 작업은 히스토리로 관리 되어 언제라도 확인할 수 있다. 이를 통해 관리자의 오류를 줄이고 문제 발생 시에 원인을 빠르게 추적할 수 있다.

### **관리자 권한 관리**

Athena Meerkat을 통해 관리자 계정뿐만 아니라 메뉴별 읽기 및 쓰기 권한을 부여할 수 있다. 이러한 계정별 권한 관리를 통해 허용된 관리자에게 허용된 작업만 수행토록 할 수 있다.