



Big Data Performance Management

**MANUAL**

Copyright © 2016 EXEM Inc. All rights reserved. EXEM and the Flamingo Logo are registered trademarks of EXEM Inc.

## 목차

1. Flamingo 소개	1
1.1. Flamingo Big Data Performance Management 소개	1
1.2. 아키텍처	2
1.3. 구성요소	2
1.4. 시스템 요구사항	3
1.5. 기본 경로 및 구성 정보	3
1.6. 사용하는 포트 현황	4
2. Cloudera CDH 배포판 설치 가이드	5
2.1. JDK 8 설치하기	5
3. 패키지 설치	7
3.1. RPM 패키지 설치	7
3.2. MQ Sink 배포하기	7
4. Hadoop 모니터링 설정하기	8
4.1. 서비스별 MQ Sink 설정 정보 추가하기	8
4.2. HDFS Hadoop Metrics2 설정하기	8
4.3. YARN Hadoop Metrics2 설정하기	12
5. Oozie 모니터링 설정하기	16
5.1. Oozie JPA 라이브러리 추가하기	16
5.2. Oozie-site.xml 설정하기	16
5.3. Oozie Event Listener 설정하기	19
6. MapReduce 모니터링 설정하기	21
6.1. Resource Manager log4j.properties 설정하기	22
6.2. JNDI 설정하기	23
7. HDFS 모니터링 설정하기	25
7.1. NameNode log4j.properties 설정하기	26
7.2. JNDI 설정하기	27
8. Apache Hive 모니터링 설정하기	29
8.1. Hive JMX(Hiveserver2) 설정 하기	29
8.2. Hive Agent	29
8.3. Hive CLI 모니터링 설정하기	31
9. 서비스 재시작 하기	34
10. 시스템 리소스 모니터링	35
10.1. System Agent 시작하기	35
10.2. System Agent 옵션	35
10.3. System Agent의 MQ URL 변경하기	35
11. Resource Manager Agent 설정 하기	37
12. Namenode Agent 설정 하기	38
13. Flamingo Collector 설정 하기	39
13.1. application.properties 파일 설정 하기	39
13.2. collector.properties 파일 설정 하기	40
14. Flamingo Web 설정 하기	44
14.1. config.properties 설정 하기	44
14.2. hadoop.properties 설정 하기	45

15. 서비스 구동 하기 .....	47
15.1. Apache ActiveMQ .....	47
15.2. Flamingo Collector .....	47
15.3. Flamingo Web .....	47
16. 릴리즈 노트 .....	48
16.1. Release 3.0.0 .....	48
17. 제품 및 기술지원 정보 .....	49



## Chapter 1. Flamingo 소개

Flamingo는 Apache Hadoop EcoSystem을 구성하는 다양한 컴포넌트 및 서비스의 관리 및 분석에 있어서 많은 사용자의 어려운 점(통합의 어려움, 관리 기능 미비, 낮은 사용성 등)을 해소하기 위해서 시작된 프로젝트로서 초기에 오픈소스인 GPL v3 라이선스로 출발하였습니다.

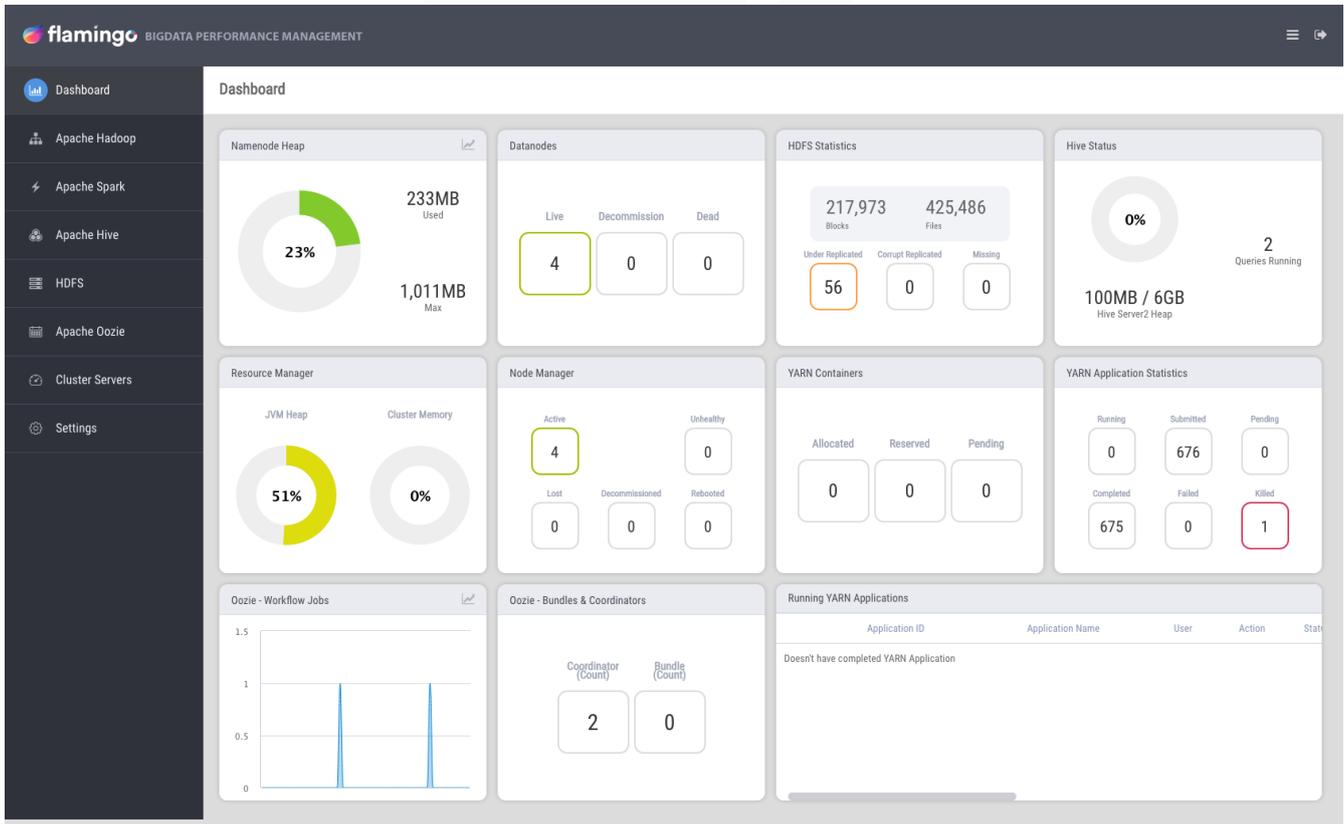
현재 Flamingo는 통합 브랜드로 발전하였고, 다음과 같이 총 3개의 프로젝트 및 제품으로 구성되어 있습니다. 총 3개의 프로젝트중 여전히 오픈소스로 개발되고 있는 프로젝트(Flamingo Oozie Workflow Designer, Flamingo Analytics)는 2개이며, 나머지 1개의 프로젝트(Flamingo Big Data Performance Management)는 On-Premise 및 Cloud SaaS 형태로만 제공하는 Commercial 제품입니다

제품 구성	라이선스	개요
Flamingo Oozie Workflow Designer	Apache 2	Apache Oozie 워크플로우 디자이너 프로젝트 : <a href="http://www.github.com/EXEM-OSS/flamingo">http://www.github.com/EXEM-OSS/flamingo</a> [github]
Flamingo Analytics	Apache 2 Subscription	빅데이터 기술을 이용한 데이터 분석 도구 프로젝트 : <a href="http://www.github.com/EXEM-OSS/flamingo-analytics">http://www.github.com/EXEM-OSS/flamingo-analytics</a> [github]
Flamingo Big Data Performance Management	On-Premise SaaS	빅데이터 성능 관리 도구

본 문서는 다음 제품 및 프로젝트에서 Flamingo Big Data Performance Management의 설치 및 사용법에 대한 내용만 다루며, 본 문서에 포함되어 있는 설치 가이드는 기술지원 및 Flamingo 엔지니어를 위한 문서이고 사용자 가이드는 Flamingo Big Data Performance Management 사용자를 위한 문서입니다.

### 1.1. Flamingo Big Data Performance Management 소개

Flamingo Big Data Performance Management(이하 Flamingo)는 Apache Hadoop EcoSystem을 구성하는 주요 서비스에 대한 성능을 모니터링하는 기능을 제공합니다.

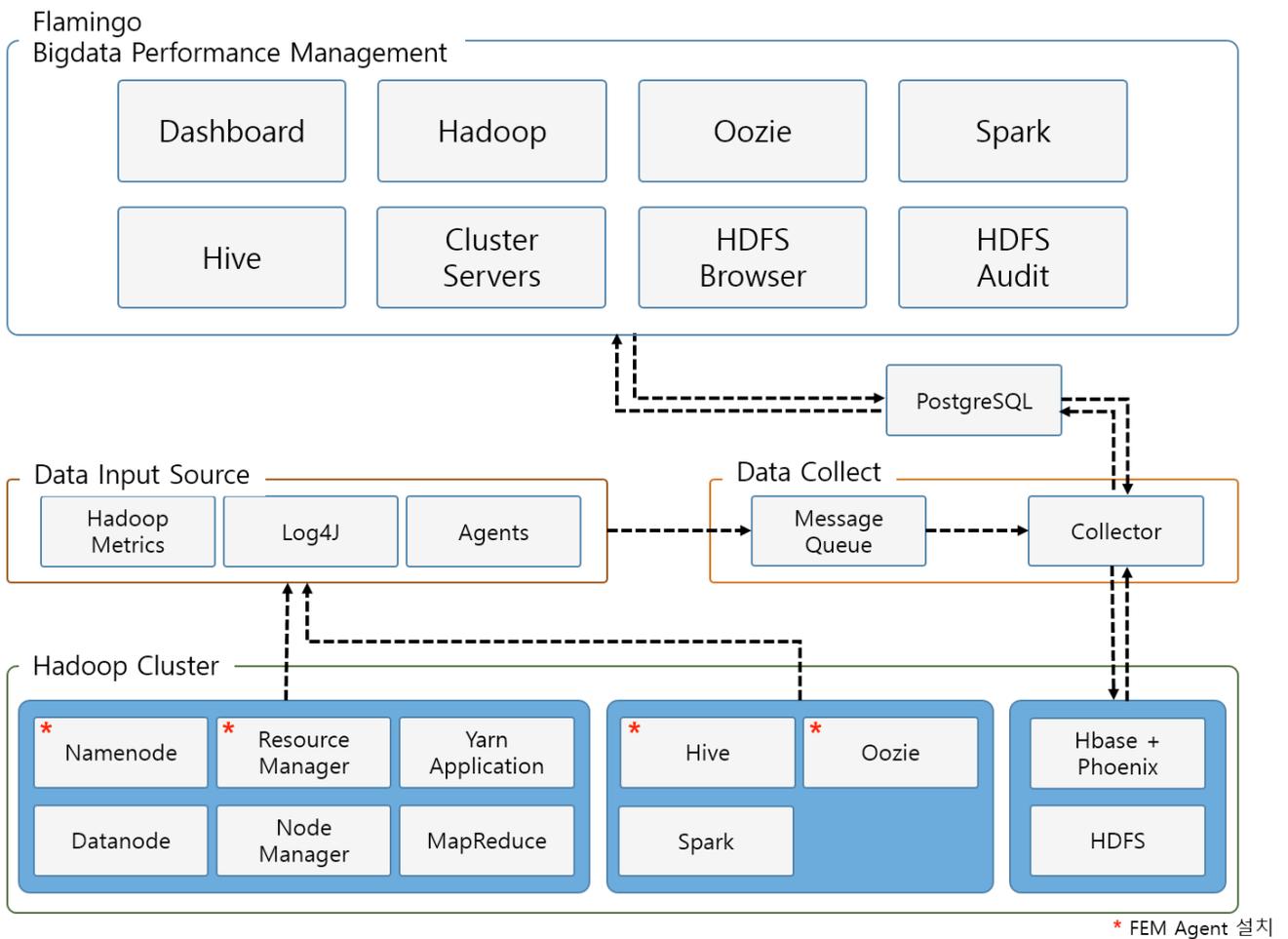


Flamingo Big Data Performance Management는 다음의 기능을 제공합니다.

주요 기능	기능 설명
Dashboard	Apache Hadoop EcoSystem의 핵심 서비스에 대한 핵심 성능 관리 지표를 시각화
Apache Hadoop	Namenode, Resource Manager, Datanode, Node Manager, YARN, MapReduce 등의 주요 핵심 서비스에 대한 성능 정보를 수집하여 시각화
Apache Spark	Apache Spark Application의 성능을 모니터링하고 시각화
Apache Hive	Apache Hive의 Metastore 관리, Session, Query 등을 추적하며 Topology View를 통해 Hive Query의 실행 상황을 시각화
HDFS	HDFS의 관리를 위한 브라우저 및 Audit 정보를 시각화
Apache Oozie	Apache Oozie의 Workflow, Coordinator, Bundle을 관리하고 시각화
Server 모니터링	Hadoop Cluster를 구성하는 서버의 Core, RAM, Disk, Network, Process 정보를 수집하여 실시간으로 시각화
설정	Flamingo의 각종 정보를 설정

## 1.2. 아키텍처

Flamingo의 아키텍처는 각 구성요소를 포함하여 다음과 같이 구성되어 있습니다.



## 1.3. 구성요소

구성요소	개요
Flamingo Web	Flamingo 웹 애플리케이션
Flamingo Collector	Flamingo의 각종 Agent에서 실시간으로 정보를 수신하여 관리하는 수집기
Flamingo System Agent	리눅스 시스템의 CPU, Core, RAM, Disk, Network 정보를 실시간으로 수집
Flamingo Namenode Agent	Namenode의 주요 정보를 수집하고 관리하는 에이전트
Flamingo Datanode Agent	Datanode의 주요 정보를 수집하고 관리하는 에이전트
Flamingo Resource Manager Agent	Resource Manager의 주요 정보를 수집하고 관리하는 에이전트
Flamingo Node Manager Agent	Node Manager의 주요 정보를 수집하고 관리하는 에이전트
Flamingo Hive Agent	Apache Hive의 주요 정보를 수집하고 쿼리를 관리하는 에이전트
Flamingo Oozie Agent	Apache Oozie의 주요 정보를 수집하고 관리하는 에이전트

#### 1.4. 시스템 요구사항

구분	내용	비고
OS	Linux Kernel 2.6 이상	CentOS 6 이상
Database	PostgreSQL 9.2 이상	UTF-8 Character Set Oozie 모니터링은 현재 Postgres만 지원
Memory	4G 이상	
Java	JDK 1.8 이상	
Hortonworks HDP	Hortonworks HDP 2.4 이상	
Cloudera CDH	Cloudera CDH 5.4 이상	
Apache Hadoop	Apache Hadoop 2.3 이상	
Web Browser	Internet Explorer 10+, Google Chrome, Safari, Firefox	

#### 1.5. 기본 경로 및 구성 정보

Flamingo의 기본 설치 경로와 Linux User/Group 정보는 다음과 같습니다. 또한 모든 디렉토리는 타 계정에서 접근할 수 있도록 실행(x) 권한이 필요하며 기본 권한은 755(rwxr-xr-x)로 설정되어야 합니다.

구분	PATH	User	Group	비고
기본설치 경로	<b>/opt/exem/fem/3.0.1</b>	flamingo	flamingo	규칙 : <b>/opt/exem/fem/&lt;VERSION&gt;</b>
링크 경로	<b>/opt/exem/fem/current</b>	flamingo	flamingo	<b>/opt/exem/fem/3.0.1</b> 경로의 링크
Agent 설치 경로	<b>/opt/exem/fem/current/agents</b>	flamingo	flamingo	
MQ Sink 설치 경로	<b>/opt/exem/fem/current/agents</b>	flamingo	flamingo	
Web 설치 경로	<b>/opt/exem/fem/current/web</b>	flamingo	flamingo	
Collector 설치 경로	<b>/opt/exem/fem/current/collector</b>	flamingo	flamingo	
MQ 설치 경로	<b>/opt/exem/fem/current/mq</b>	flamingo	flamingo	

Flamingo Agent는 다음과 같이 구성되어 있으며 다음의 User와 Group으로 설정되어야 합니다.

Agent	PATH	User	Group
Namenode Agent	<code>/opt/exem/fem/current/agents/flamingo-management-nn-agent-3.0.1.jar</code>	hdfs	hadoop
Resource Manager Agent	<code>/opt/exem/fem/current/agents/flamingo-management-rm-agent-3.0.1.jar</code>	yarn	hadoop
Node Manager Agent	<code>/opt/exem/fem/current/agents/flamingo-management-nm-agent-3.0.1.jar</code>	yarn	hadoop
Datanode Agent	<code>/opt/exem/fem/current/agents/flamingo-management-dn-agent-3.0.1.jar</code>	hdfs	hadoop
Oozie Agent	<code>/opt/exem/fem/current/agents/flamingo-management-oz-agent-3.0.1.jar</code>	oozie	hadoop
Hive Agent	<code>/opt/exem/fem/current/agents/flamingo-management-hive-agent-3.0.1.jar</code>	hive	hadoop
System Agent	<code>/opt/exem/fem/current/agents/flamingo-management-system-agent-3.0.1.jar</code>	flamingo	flamingo
MQ Sink	<code>/opt/exem/fem/current/agents/flamingo-management-mq-sink-3.0.1.jar</code>	flamingo	flamingo

NOTE Flamingo가 동작하기 위한 Linux User 및 Group은 배포판에 따라 다를 수 있습니다.

## 1.6. 사용하는 포트 현황

Flamingo Agent, Collector, Web은 다음의 포트를 사용합니다. 따라서 고객사 내부의 방화벽 및 보안 정책이 있는 경우 다음의 포트에 대해서 사전 승인을 받거나 또는 사전 검토가 필요합니다.

구분	포트	접속 대상 서버
Namenode Agent	10070	MQ
Resource Manager Agent	18032	MQ
Node Manager Agent	N/A	MQ
Datanode Agent	N/A	MQ
Oozie Agent	31000	MQ
Hive Agent	10179	MQ
System Agent	N/A	MQ
MQ Sink	N/A	MQ
MQ (Message Queue)	N/A	Collector
Collector	38081	Hadoop, Agents
Web	38080	HBase, PostgreSQL, Hadoop, Agents

## Chapter 2. Cloudera CDH 배포판 설치 가이드

Cloudera CDH 배포판에서 Flamingo를 설치하기 위한 순서는 다음과 같습니다.

순서	내용	주의사항
1	JDK 8 설치	OpenJDK 8 또는 Oracle JDK 8으로 설치
2	Flamingo RPM 패키지 설치	Hadoop Cluster 및 Flamingo 서버에 모두 설치
3	Flamingo JDK 설정	Flamingo Web, Collector, Agent에 JDK 설정
4	Flamingo 관련 포트 변경	Flamingo Web, Collector, Agent 등의 포트 변경
5	Apache HBase 구동	<b>flamingo</b> 계정으로 실행
6	Apache ActiveMQ 구동	<b>flamingo</b> 계정으로 실행
7	Flamingo Collector 구동	<b>flamingo</b> 계정으로 실행
8	Flamingo Web 구동	<b>flamingo</b> 계정으로 실행
9	Flamingo Agent 설정	Flamingo Agent의 Linux User, Group 확인 (배포판에 따라 다름)
10	Hadoop 주요 서비스 재시작	재시작시 반드시 해당 서비스의 로그를 확인하여 Flamingo Agent의 권한 문제로 비정상 종료되었는지 확인 (배포판에 따라 다름)
11	<a href="#">설치 완료 점검</a>	각종 포트의 정상동작 및 수집 데이터의 정상 송수신 확인

### 2.1. JDK 8 설치하기

#### 2.1.1. 설치 확인하기

Java 8 또는 JDK 8이 정상적으로 설치되었는지 확인하기 위해서 다음의 커맨드를 실행합니다.

```
$ java -version
java version "1.8.0_65"
Java(TM) SE Runtime Environment (build 1.8.0_65-b17)
Java HotSpot(TM) 64-Bit Server VM (build 25.65-b01, mixed mode)
```

#### NOTE

Hadoop Cluster에 속해 있는 노드에 Flamingo를 설치하는 경우 보통 CDH가 설치되어 있으며 이 경우 JDK가 설치되어 있습니다. 구 버전의 CDH가 설치되어 있는 경우는 JDK 1.7이 설치되어 있을 수 있으며 이 경우 JDK 1.8을 반드시 설치해야 합니다.

JDK 8이 설치되어 있지 않은 경우 Ubuntu, CentOS 등의 OS 유형에 따라서 설치를 진행하도록 합니다.

#### 2.1.2. CentOS

CentOS에서 JDK 8을 설치하기 위해서 root 계정으로 다음과 같이 커맨드를 실행합니다.

```
$ yum install -y java-1.8.0-openjdk-devel.x86_64
```

#### 2.1.3. Ubuntu

Ubuntu에서 JDK 8을 설치하기 위해서 root 계정으로 다음과 같이 커맨드를 실행합니다.

```
$ apt-get install openjdk-8-jdk
```

#### 2.1.4. 수동 설치하기

Oracle JDK는 라이선스 문제로 인하여 수동으로 설치해야 하며 수동으로 Oracle JDK 8을 설치하기 위해서 <http://www.oracle.com/technetwork/java/javase/downloads/index.html> 에서 다운로드합니다.

```
$ tar zxvf jdk-8u111-linux-x64.tar.gz
$ mkdir -p /usr/lib/jvm/jdk1.8.0
$ mv jdk1.8.0_111/* /usr/lib/jvm/jdk1.8.0/
$ update-alternatives --install "/usr/bin/java" "java" "/usr/lib/jvm/jdk1.8.0/bin/java" 1
$ update-alternatives --install "/usr/bin/javac" "javac" "/usr/lib/jvm/jdk1.8.0/bin/javac" 1
```



## Chapter 3. 패키지 설치

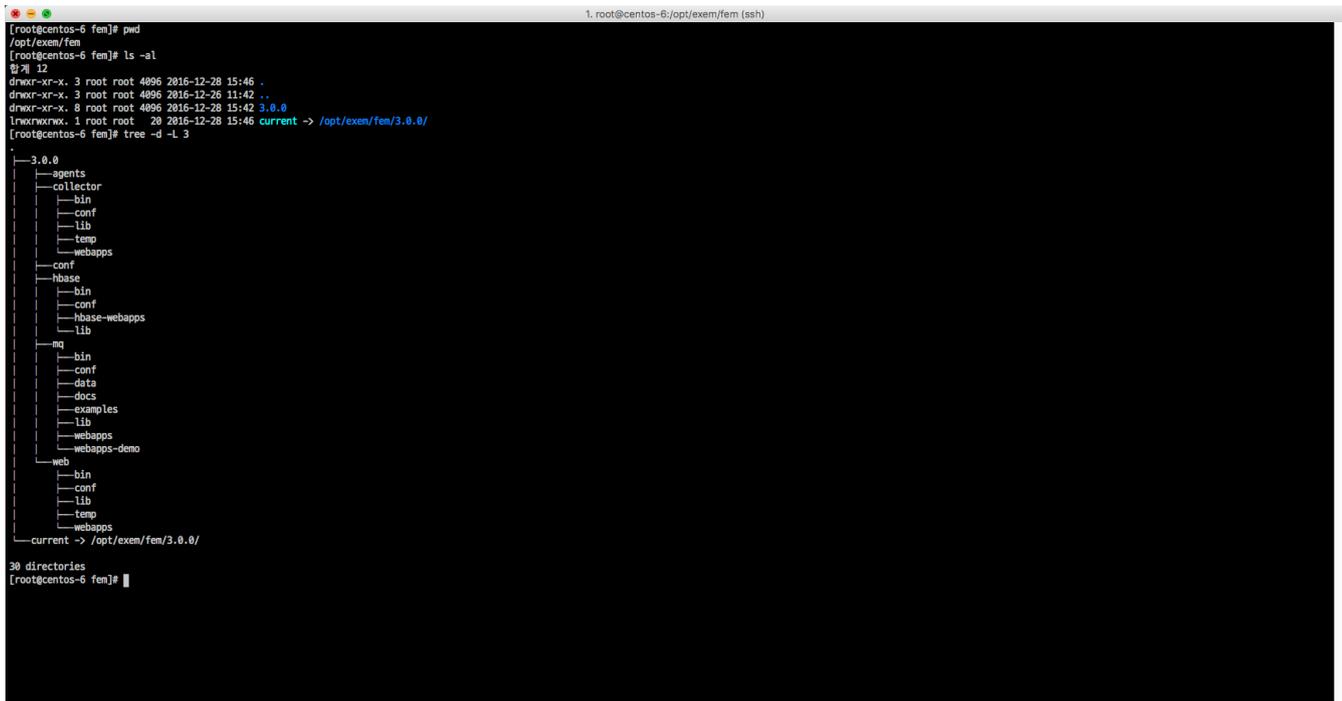
Flamingo 는 RPM 패키지 형태로 제공됩니다. 제공되는 패키지는 agent, collector, web 으로 나누어져 있습니다. agent 패키지는 모든 서버에 복사하고 web, collector는 설치할 서버에 복사합니다. 복사할 경로는 `/opt/exem/fem/install` 디렉토리입니다.

### 3.1. RPM 패키지 설치

RPM을 복사한 모든 서버에서 아래와 같이 커맨드를 실행하여 설치 합니다.

```
$ rpm -Uvh flamingo-management-*--3.0.1.rpm
```

설치 하게 되면 아래와 같이 `/opt/exem/fem` 의 경로에 설치가 되며 디렉토리 구조는 다음과 같습니다



```
[root@centos-6 fem]# pwd
/opt/exem/fem
[root@centos-6 fem]# ls -al
합계 12
drwxr-xr-x. 3 root root 4096 2016-12-28 15:46 .
drwxr-xr-x. 3 root root 4096 2016-12-26 11:42 ..
drwxr-xr-x. 8 root root 4096 2016-12-28 15:42 3.0.0
lnsrxrwx. 1 root root 20 2016-12-28 15:46 current -> /opt/exem/fem/3.0.0/
[root@centos-6 fem]# tree -d -L 3
- 3.0.0
|-- agents
|   |-- collector
|   |   |-- bin
|   |   |-- conf
|   |   |-- lib
|   |   |-- temp
|   |   |-- webapps
|   |-- conf
|   |   |-- hbase
|   |   |   |-- bin
|   |   |   |-- conf
|   |   |   |-- hbase-webapps
|   |   |   |-- lib
|   |-- mq
|   |   |-- bin
|   |   |-- conf
|   |   |-- data
|   |   |-- docs
|   |   |-- examples
|   |   |-- lib
|   |   |-- webapps
|   |   |-- webapps-demo
|   |-- web
|   |   |-- bin
|   |   |-- conf
|   |   |-- lib
|   |   |-- temp
|   |   |-- webapps
|-- current -> /opt/exem/fem/3.0.0/
30 directories
[root@centos-6 fem]#
```

### 3.2. MQ Sink 배포하기

최신 Hadoop 버전에서는 Hadoop의 관련 정보를 수집하기 위해서 Hadoop에서 자체적으로 제공하는 기능인 Hadoop Metrics 2가 있으며 Flamingo에서는 Hadoop의 성능 및 모니터링 정보를 수집하기 위해서 Hadoop Metrics 2를 사용합니다.

Hadoop Metrics 2는 정보를 Asynchronous Pushing하므로 Flamingo는 비동기로 메시지를 수신하기 위해서 MQ(Message Queue)를 이용합니다. 따라서 MQ Sink는 Hadoop의 각 컴포넌트 또는 서비스(Resource Manager, Node Manager, Namenode, Datanode 등에서 비동기로 메시지를 송신할 수 있는 기능을 제공하며 MQ Sink는 모든 Hadoop Cluster를 구성하는 노드에 설치되어야 합니다.

MQ Sink의 파일명은 `flamingo-management-hm2-3.x.x.jar` 이며 `/opt/exem/fem/current/agents` 에 있습니다. 이 파일을 모든 서버의 다음 디렉토리에 복사하도록 합니다.

- `/opt/cloudera/parcels/CDH/lib/hadoop-yarn/lib`
- `/opt/cloudera/parcels/CDH/lib/hadoop-hdfs/lib`

NOTE 모든 설치는 CDH 5.x 버전을 기준으로 하며 또한 CDH의 설치 버전에 따라서 설치 경로가 다를 수 있습니다.

## Chapter 4. Hadoop 모니터링 설정하기

Hadoop에서 성능 및 모니터링 정보를 수집하기 위해서는 MQ Sink가 MQ로 메시지를 송신해야 합니다. 이를 위해서 Hadoop의 **hadoop-metrics2.properties** 파일을 모두 수정해야 합니다. CDH 5.10.0에서는 **hadoop-metrics2.properties** 파일을 직접 수정하지 않고, CDH Manager를 통해 간편하게 설정 할 수 있습니다. 이 경우 하둡 서비스 재시작 시 원상복귀가 되므로, CDH에서 제공하는 템플릿 파일을 알맞게 수정하도록 합니다.

### 4.1. 서비스별 MQ Sink 설정 정보 추가하기

서비스별 MQ Sink를 설정해야 하는 Hadoop 서비스는 다음과 같습니다.

- (Active) Namenode 및 (Standby) Namenode
- Secondary Namenode
- Datanode
- Resource Manager
- Node Manager

### 4.2. HDFS Hadoop Metrics2 설정하기

Hadoop Metrics 2를 사용하기 위해서는 flamingo-management-hm2를 빌드해야 합니다.

빌드 후 각 노드에 아래 경로에 넣어 줍니다. (hadoop lib 경로)

broker\_url에는 설치된 activemq의 url를 넣고 username에는 아이디를 password는 비밀번호를 넣어줍니다.

별도로 설정하지 않으면 기본값은 admin/admin입니다.

CDH 5.x 기준

```
/opt/cloudera/parcels/CDH/lib/hadoop-hdfs/lib
```

CDH 5.x 버전에서는 CDH Manager(Cloudera Manager)에서 **hadoop-metrics2.properties** 파일을 직접 수정 가능합니다. 아래 화면과 같이 CDH Manager에서 **HDFS** 버튼을 클릭하고 **구성** 버튼을 누릅니다.

The screenshot displays the Cloudera Manager interface. On the left, a list of services for 'Cluster 1 (CDH 5.10.0, Parcel)' is shown, with 'HDFS' highlighted in a red box. Below this is the 'Cloudera Management Service' section. On the right, two charts are visible: '클러스터 CPU' (Cluster CPU) showing low usage (1.2%) and 'HDFS IO' showing a peak in data transfer rate (2.8b/s). The main navigation bar includes 'cloudera MANAGER' and various menu items like '클러스터', '호스트', '진단', '감사', '차트', and '관리'. Below the navigation bar, the 'HDFS (Cluster 1)' page is shown, with the '구성' (Configuration) tab highlighted in a red box. The 'HDFS 요약' (HDFS Summary) section shows a progress bar for '구성 용량' (Configuration Capacity) at 3.0GiB/483.1GiB. The '상태 테스트' (Status Test) section shows 'NameNode 상태' (NameNode Status) as '정상' (Normal) and '6 양호 표시' (6 Good Indicators).

HDFS 구성 페이지까지 이동 후 아래 화면과 같이 검색창에 **metric** 이라고 검색 하면 hadoop-metricrs2설정을 위한 입력창이 나타나게 됩니다.

metricj

Hadoop 메트릭2 고급 구성 스니펫(안전 밸브)  
동일한 값 편집

DataNode Default Group C

NameNode Default Group C

SecondaryNameNode Default Group C

#### 4.2.1. DataNode Default Group 설정하기

```

datanode.sink.dfs.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
datanode.sink.dfs.period=10
datanode.sink.dfs.username=admin ①
datanode.sink.dfs.password=admin ②
datanode.sink.dfs.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
datanode.sink.dfs.queue_name=collector.hdfs.dataNode
datanode.sink.dfs.context=dfs
datanode.sink.jvm.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
datanode.sink.jvm.period=10
datanode.sink.jvm.username=admin ①
datanode.sink.jvm.password=password ②
datanode.sink.jvm.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
datanode.sink.jvm.queue_name=collector.hdfs.dataNode
datanode.sink.jvm.context=jvm
datanode.sink.rpc.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
datanode.sink.rpc.period=10
datanode.sink.rpc.username=admin ①
datanode.sink.rpc.password=admin ②
datanode.sink.rpc.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
datanode.sink.rpc.queue_name=collector.hdfs.dataNode
datanode.sink.rpc.context=rpc

```

- ① MQ의 username (기본값 admin)
- ② MQ의 password (기본값 admin)
- ③ MQ의 URL (MQ의 IP 주소 및 Port; 기본값 61616)

#### 4.2.2. NameNode Default Group 설정하기

```

namenode.sink.dfs.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
namenode.sink.dfs.period=10
namenode.sink.dfs.username=admin ①
namenode.sink.dfs.password=admin ②
namenode.sink.dfs.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
namenode.sink.dfs.queue_name=collector.hdfs.nameNode
namenode.sink.dfs.context=dfs
namenode.sink.jvm.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
namenode.sink.jvm.period=10
namenode.sink.jvm.username=admin ①
namenode.sink.jvm.password=admin ②
namenode.sink.jvm.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
namenode.sink.jvm.queue_name=collector.hdfs.nameNode
namenode.sink.jvm.context=jvm
namenode.sink.rpc.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
namenode.sink.rpc.period=10
namenode.sink.rpc.username=admin ①
namenode.sink.rpc.password=admin ②
namenode.sink.rpc.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
namenode.sink.rpc.queue_name=collector.hdfs.nameNode
namenode.sink.rpc.context=rpc

```

- ① MQ의 username (기본값 admin)
- ② MQ의 password (기본값 admin)
- ③ MQ의 URL (MQ의 IP 주소 및 Port; 기본값 61616)

#### 4.2.3. SecondaryNameNode Default Group 설정하기

```

secondarynamenode.sink.dfs.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
secondarynamenode.sink.dfs.period=10
secondarynamenode.sink.dfs.username=admin ①
secondarynamenode.sink.dfs.password=admin ②
secondarynamenode.sink.dfs.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
secondarynamenode.sink.dfs.queue_name=collector.hdfs.secondaryNameNode
secondarynamenode.sink.dfs.context=dfs
secondarynamenode.sink.jvm.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
secondarynamenode.sink.jvm.period=10
secondarynamenode.sink.jvm.username=admin ①
secondarynamenode.sink.jvm.password=admin ②
secondarynamenode.sink.jvm.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
secondarynamenode.sink.jvm.queue_name=collector.hdfs.secondaryNameNode
secondarynamenode.sink.jvm.context=jvm
secondarynamenode.sink.rpc.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
secondarynamenode.sink.rpc.period=10
secondarynamenode.sink.rpc.username=admin ①
secondarynamenode.sink.rpc.password=admin ②
secondarynamenode.sink.rpc.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
secondarynamenode.sink.rpc.queue_name=collector.hdfs.secondaryNameNode
secondarynamenode.sink.rpc.context=rpc

```

- ① MQ의 username (기본값 admin)
- ② MQ의 password (기본값 admin)
- ③ MQ의 URL (MQ의 IP 주소 및 Port; 기본값 61616)

위 내용을 아래 화면 과 같이 입력된 것을 확인 후 `변경내용저장`을 합니다.

```

metric

Hadoop 메트릭2 고급 구성 스크립트(안전 벨트)

DataNode Default Group
datanode.sink.dfs.class=com.exem.flemingo.management.hm.hadoop.metrics.MoSink
datanode.sink.dfs.period=10
datanode.sink.dfs.username=admin
datanode.sink.dfs.password=password
datanode.sink.dfs.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true

NameNode Default Group
namenode.sink.dfs.class=com.exem.flemingo.management.hm.hadoop.metrics.MoSink
namenode.sink.dfs.period=10
namenode.sink.dfs.username=admin
namenode.sink.dfs.password=password
namenode.sink.dfs.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true

SecondaryNameNode Default Group
secondarynamenode.sink.dfs.class=com.exem.flemingo.management.hm.hadoop.metrics.MoSink
secondarynamenode.sink.dfs.period=10
secondarynamenode.sink.dfs.username=admin
secondarynamenode.sink.dfs.password=password
secondarynamenode.sink.dfs.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true
secondarynamenode.sink.queue_name=collector_hdfs_secondaryNameNode
secondarynamenode.sink.rpc.context=rpc
  
```

### 4.3. YARN Hadoop Metrics2 설정하기

Hadoop Metrics 2를 사용하기 위해서는 flamingo-management-hm2를 빌드해야 합니다.

빌드 후 각 노드에 아래 경로에 넣어 줍니다. (hadoop lib 경로)

broker\_url 에는 설치된 activemq의 url을 넣고 username에는 아이디를 password는 비밀번호를 넣어줍니다.

별도로 설정하지 않으면 기본값은 admin/admin 입니다.

CDH 5.x 기준

```
/opt/cloudera/parcels/CDH/lib/hadoop-yarn/lib
```

CDH Manager(Cloudera Manager)에서 **hadoop-metrics2.properties** 파일을 직접 수정 가능합니다. 아래 화면과 같이 CDH Manager에서 **HDFS** 버튼을 클릭하고 **구성** 버튼을 누릅니다.

**Cluster 1** (CDH 5.10.0, Parcel) ▼

- ☰ 호스트
- H HBase
- H HDFS ✖ 1
- H Hive 📄
- H Hue
- O Oozie
- S Spark
- Y YARN (MR2 Inc...)
- Z ZooKeeper ✖ 1

**Cloudera Management Service**

- C Cloudera Mana... ! 4 ✖ 4

**차트**

**클러스터 CPU**

Cluster 1, 전체 호스트의 호스트 CPU 사용량 1.2%

**HDFS IO**

모든 DataNodes... 2.8b/s | 모든 DataNodes의... 1

**YARN (MR2 Included)** (Cluster 1) 작업 ▼

---

상태
인스턴스
구성
모니터링
애플리케이션
리소스 풀
차트 라이브러리
감사
웹 UI ▼
쿼리 링크 ▼

**상태 테스트** 트리거 생성

● **ResourceManager 상태** 표시 안 함...

ResourceManager 요약: cdh1.exem.oss(가용성: 활성, 상태: 주의). 이 상태 테스트는 활성 ResourceManager의 상태를 반영합니다.

● **JobHistory Server 상태** 표시 안 함...

JobHistory Server의 상태가 주의 상태입니다. 다음 상태 테스트 결과가 주의입니다. 호스트 상태.

● **NodeManager 상태** 표시 안 함...

양호 상태의 NodeManager: 4개. 주의 상태의 NodeManager: 0개. 총 NodeManager: 4개. 양호(%): 100.00%. 양호 또는 주의(%): 100.00%.

⊗ **YARN Container Usage Aggregation** 표시 안 함...

This health test is disabled because container usage metric collection is disabled for YARN.

**상태 요약**

HDFS 구성 페이지까지 이동 후 아래 화면과 같이 검색창에 **metric** 이라고 검색 하면 hadoop-metrics2설정을 위한 입력창이 나타나게 됩니다.

metric

Hadoop 메트릭2 고급 구성 스니펫(안전 벨브)  
동일한 값 편집

JobHistory Server Default Group

NodeManager Default Group C

ResourceManager Default Group C

#### 4.3.1. NodeManager Default Group 설정하기

```
nodemanager.sink.yarn.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
nodemanager.sink.yarn.period=10
nodemanager.sink.yarn.username=admin ①
nodemanager.sink.yarn.password=admin ②
nodemanager.sink.yarn.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
nodemanager.sink.yarn.queue_name=collector.yarn.nodeManager
nodemanager.sink.yarn.context=yarn
nodemanager.sink.jvm.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
nodemanager.sink.jvm.period=10
nodemanager.sink.jvm.username=admin ①
nodemanager.sink.jvm.password=admin ②
nodemanager.sink.jvm.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
nodemanager.sink.jvm.queue_name=collector.yarn.nodeManager
nodemanager.sink.jvm.context=jvm
nodemanager.sink.rpc.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
nodemanager.sink.rpc.period=10
nodemanager.sink.rpc.username=admin ①
nodemanager.sink.rpc.password=admin ②
nodemanager.sink.rpc.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
nodemanager.sink.rpc.queue_name=collector.yarn.nodeManager
nodemanager.sink.rpc.context=rpc
```

- ① MQ의 username (기본값 admin)
- ② MQ의 password (기본값 admin)
- ③ MQ의 URL (MQ의 IP 주소 및 Port; 기본값 61616)

#### 4.3.2. ResourceManager Default Group 설정하기

```

resourcemanager.sink.yarn.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
resourcemanager.sink.yarn.period=10
resourcemanager.sink.yarn.username=admin ①
resourcemanager.sink.yarn.password=admin ②
resourcemanager.sink.yarn.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
resourcemanager.sink.yarn.queue_name=collector.yarn.resourceManager
resourcemanager.sink.yarn.context=yarn
resourcemanager.sink.jvm.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
resourcemanager.sink.jvm.period=10
resourcemanager.sink.jvm.username=admin ①
resourcemanager.sink.jvm.password=admin ②
resourcemanager.sink.jvm.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
resourcemanager.sink.jvm.queue_name=collector.yarn.resourceManager
resourcemanager.sink.jvm.context=jvm
resourcemanager.sink.rpc.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
resourcemanager.sink.rpc.period=10
resourcemanager.sink.rpc.username=admin ①
resourcemanager.sink.rpc.password=admin ②
resourcemanager.sink.rpc.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true ③
resourcemanager.sink.rpc.queue_name=collector.yarn.resourceManager
resourcemanager.sink.rpc.context=rpc

```

- ① MQ의 username (기본값 admin)
- ② MQ의 password (기본값 admin)
- ③ MQ의 URL (MQ의 IP 주소 및 Port; 기본값 61616)

위 내용들을 아래 화면 과 같이 입력된 것을 확인 후 `변경내용저장`을 합니다.

클래스 2

metric

**Hadoop 메트릭2 고급 구성 스크린(안전 밸브)**

JobHistory Server Default Group

NodeManager Default Group ←

```

nodemanager.sink.yarn.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
nodemanager.sink.yarn.period=10
nodemanager.sink.yarn.username=admin
nodemanager.sink.yarn.password=password
nodemanager.sink.yarn.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true

```

ResourceManager Default Group C

```

resourcemanager.sink.yarn.class=com.exem.flamingo.management.hm.hadoop.metrics.MqSink
resourcemanager.sink.yarn.period=10
resourcemanager.sink.yarn.username=admin
resourcemanager.sink.yarn.password=password
resourcemanager.sink.yarn.broker_url=tcp://cdhfem.exem.oss:61616?daemon=true

```

## Chapter 5. Oozie 모니터링 설정하기

Flamingo에서 Apache Oozie를 모니터링하기 위해서 Flamingo는 Oozie의 JPA를 직접 처리합니다. 이를 위해서 Oozie의 Service Extension에 Flamingo의 Oozie JPA Service를 **oozie-site.xml** 파일에 등록해야 합니다. 설정하는 방법은 CDH Manager에서 설정하는 방법과 **oozie-site.xml** 파일을 직접 수정하는 방법이 있으며 Cloudera Manager에서 설정하는 것을 권장합니다.

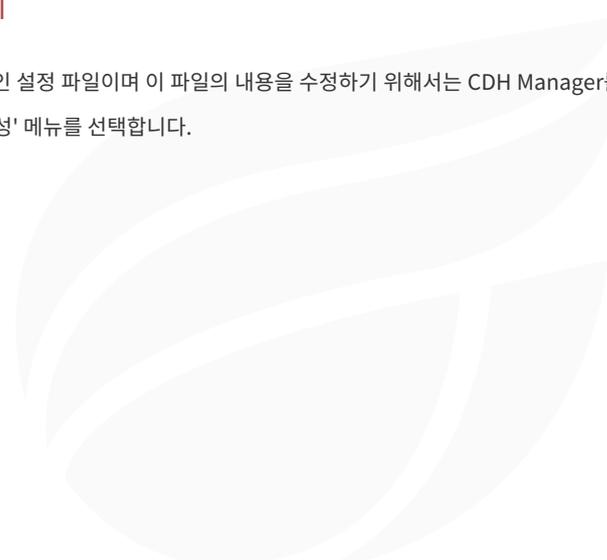
### 5.1. Oozie JPA 라이브러리 추가하기

Flamingo에서 Oozie의 JPA를 처리하기 위한 라이브러리를 제공합니다. 파일명은 **flamingo-management-agent-oz-3.x.x.jar** 이며 **/opt/exem/fem/current/agents** 에 있습니다. 이 파일을 oozie 웹콘솔 라이브러리에 추가합니다. 경로는 우지 웹콘솔이 설치된 서버의 **/opt/cloudera/parcels/CDH/lib/oozie/webapps/oozie/WEB-INF/lib** 입니다.

NOTE | 서버의 `/opt/cloudera/parcels/CDH/lib/oozie/webapps/oozie/WEB-INF`에서 lib 디렉토리가 없다면 만들어 줍니다.

### 5.2. Oozie-site.xml 설정하기

**oozie-site.xml** 파일은 Oozie의 메인 설정 파일이며 이 파일의 내용을 수정하기 위해서는 CDH Manager를 이용해야 합니다. Oozie 설정을 변경하기 위해서 다음과 같이 'CDH > Oozie > 구성' 메뉴를 선택합니다.



**Cluster 1** (CDH 5.10.0, Parcel) ▼

- ☰ 호스트
- H HBase
- H HDFS ✂ 1
- H Hive 📄
- H Hue
- O Oozie
- S Spark
- Y YARN (MR2 Inc...)
- Z ZooKeeper ✂ 1

Cloudera Management Service

- C Cloudera Mana... ! 4 ✂ 4

**차트**

**클러스터 CPU**

Cluster 1, 전체 호스트의 호스트 CPU 사용량 **1.2%**

**HDFS IO**

모든 DataNodes... **2.8b/s** | 모든 DataNodes의... **1**

**Oozie** (Cluster 1) 작업 ▼

---

상태
인스턴스
구성
차트 라이브러리
감사
Oozie 웹 UI
쿼리 링크 ▼

**상태 테스트** 트리거 생성

● **Oozie Server 상태** 표시 안 함...

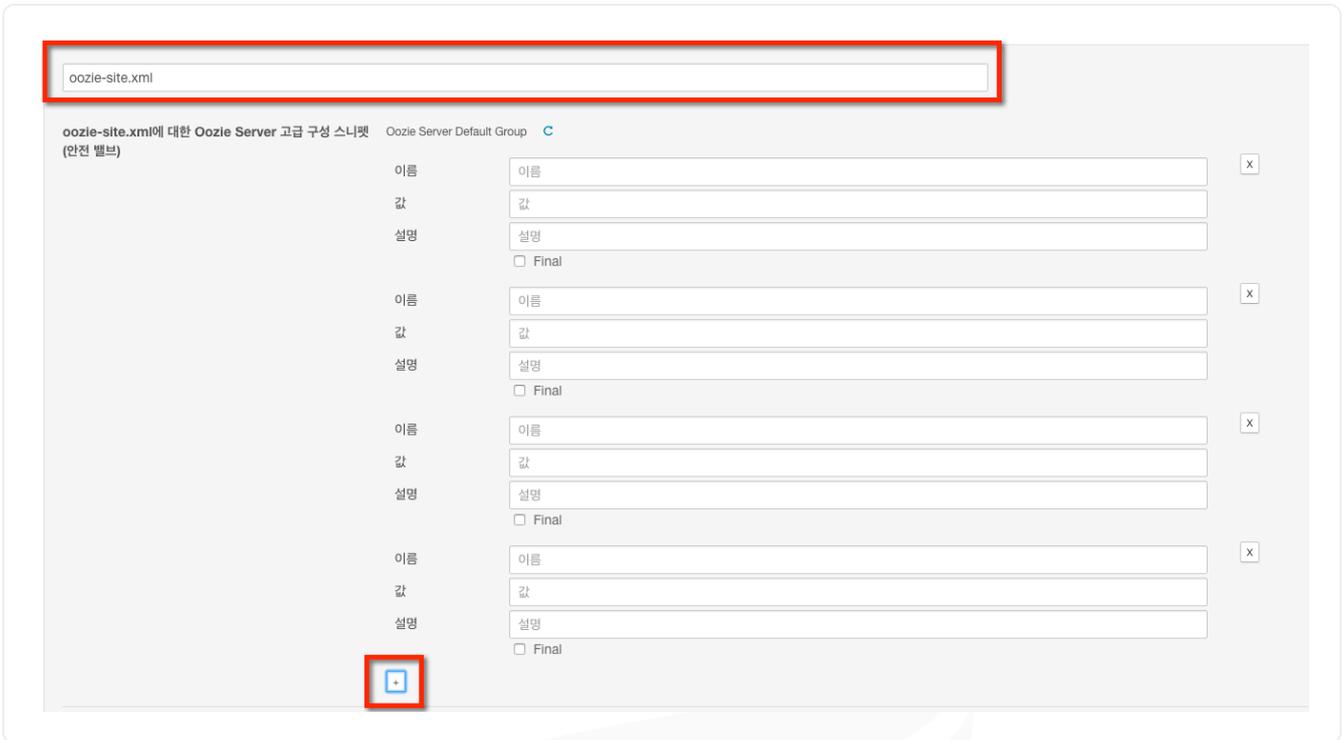
양호 상태의 Oozie Server: 0개. 주의 상태의 Oozie Server: 1개. 총 Oozie Server: 1개. 양호(%): 0.00%. 양호 또는 주의(%): 100.00%. 경고 임계값: 99.00%.

**상태 요약**

Oozie Server	● 1 주의 상태
<b>호스트</b>	● 1 주의 상태
스왑하는 중	! 1

**상태 기록**

Config 메뉴를 선택하면 Key Value 형식으로 `oozie-site.xml` 파일의 내용이 출력되며 다음과 같이



아래 내용을 `oozie-site.xml`에 대한 `Oozie Server 고급 구성 스프레드(안전 밸브)`에 Key:Value 형식으로 입력합니다.

```

key: oozie.service.EventHandlerService.event.listeners
Value : org.apache.oozie.jms.JMSJobEventListener

key: oozie.service.EventHandlerService.filter.app.types
Value: workflow_job

Key: oozie.jms.producer.connection.properties
Value:
java.naming.factory.initial#org.apache.activemq.jndi.ActiveMQInitialContextFactory;java.naming.provider.url#tcp://cdhfem
.exem.oss:61616;connectionFactoryNames#ConnectionFactory

key: oozie.service.JMSTopicService.topic.name
Value: WORKFLOW=oozieWorkflow

```

설정들이 아래의 화면과 같이 잘 반영이 되었다면 `변경내용저장`을 눌러줍니다.

oozie-site.xml

oozie-site.xml에 대한 Oozie Server 고급 구성 스니펫 (안전 탭) Oozie Server Default Group C

이름	oozie.service.EventHandlerService.event.listeners	X
값	org.apache.oozie.jms.JMSJobEventListener	
설명	설명 <input type="checkbox"/> Final	
이름	oozie.service.EventHandlerService.filter.app.types	X
값	workflow_job	
설명	설명 <input type="checkbox"/> Final	
이름	oozie.jms.producer.connection.properties	X
값	java.naming.factory.initial#org.apache.activemq.jndi.ActiveMQInitialContextFactory;java.naming.provider.url#tcp://	
설명	설명 <input type="checkbox"/> Final	
이름	oozie.service.JMSTopicService.topic.name	X
값	WORKFLOW=oozieWorkflow	
설명	설명 <input type="checkbox"/> Final	

### 5.3. Oozie Event Listener 설정하기

`oozie.services.ext` 설정을 위해 아래와 같이 `Oozie Server 플러그인`으로 검색을 합니다.

클래식 레C

Oozie Server 플러그인

Oozie Server 플러그인  
oozie.services.ext

Oozie Server Default Group C

표시 25 페이지 기준

`com.exem.flamingo.management.agent.oz.jpa.JpaNativeQueryExecutorService`  
`org.apache.oozie.service.MetricsInstrumentationService`

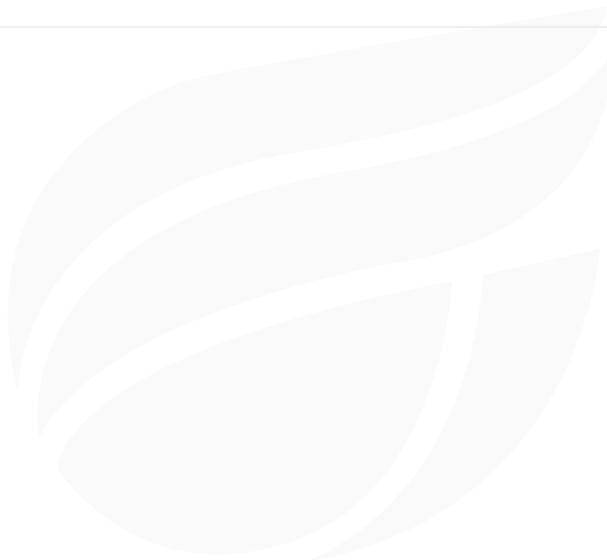
NOTE | oozie.services.ext에 항목이 여러개 일경우 콤마(,)로 구분하며 공백문자가 포함되지 않도록 합니다.

설정들이 아래의 화면과 같이 잘 반영이 되었다면 `변경내용저장`을 눌러줍니다.

oozie-site.xml

oozie-site.xml에 대한 Oozie Server 고급 구성 스니펫 (안전 밸브) Oozie Server Default Group C

이름	<input type="text" value="oozie.service.EventHandlerService.event.listeners"/>	X
값	<input type="text" value="org.apache.oozie.jms.JMSJobEventListener"/>	
설명	<input type="text" value="설명"/> <input type="checkbox"/> Final	
이름	<input type="text" value="oozie.service.EventHandlerService.filter.app.types"/>	X
값	<input type="text" value="workflow_job"/>	
설명	<input type="text" value="설명"/> <input type="checkbox"/> Final	
이름	<input type="text" value="oozie.jms.producer.connection.properties"/>	X
값	<input type="text" value="java.naming.factory.initial#org.apache.activemq.jndi.ActiveMQInitialContextFactory;java.naming.provider.url#tcp://"/>	
설명	<input type="text" value="설명"/> <input type="checkbox"/> Final	
이름	<input type="text" value="oozie.service.JMSTopicService.topic.name"/>	X
값	<input type="text" value="WORKFLOW=oozieWorkflow"/>	
설명	<input type="text" value="설명"/> <input type="checkbox"/> Final	



## Chapter 6. MapReduce 모니터링 설정하기

Flamingo는 MapReduce가 실행되거나 종료될 때 이벤트를 수신하여 Topology View 및 기타 정보를 표출하고 있습니다. MapReduce의 실행 이벤트는 YARN Resource Manager의 로그 정보를 통해서 수집이 가능합니다. 설정하는 방법은 CDH Manager 설정하는 방법과 YARN의 `log4j.properties` 파일을 직접 수정하는 방법이 있으며 CDH Manager로 설정하는 것을 권장합니다. 본 설치 가이드에서는 직접 수정하는 방법은 설명하지 않습니다.

CDH Manager로 로그인하여 'CDH > YARN > Configs' 메뉴를 선택합니다.

The screenshot displays the CDH Manager interface for Cluster 1 (CDH 5.10.0, Parcel). The left sidebar lists various services, with 'YARN (MR2 Inc...)' highlighted in a red box. Below the sidebar, the 'Cloudera Management Service' section shows 'Cloudera Mana...' with 4 warnings and 4 errors. The main content area shows the 'YARN (MR2 Included) (Cluster 1)' configuration page. The navigation bar includes '상태', '인스턴스', '구성', '모니터링', '애플리케이션', '리소스 풀', '차트 라이브러리', '감사', '웹 UI', and '쿼리 링크'. The '상태 테스트' (Status Test) section is visible, with a red box around the '구성' menu item and a mouse cursor pointing to it. The 'ResourceManager 상태' (ResourceManager Status) test result is highlighted in a red box. The test result shows: 'ResourceManager 요약: cdh1.exem.oss(가용성: 활성, 상태: 주의). 이 상태 테스트는 활성 ResourceManager의 상태를 반영합니다.' (ResourceManager Summary: cdh1.exem.oss(Availability: Active, Status: Warning). This status test reflects the status of the active ResourceManager.)

차트

클러스터 CPU

percent

100%

50%

0%

01:45

Cluster 1, 전체 호스트의 호스트 CPU 사용량 1.2%

HDFS IO

bytes / second

40b/s

20b/s

0

01:45

모든 DataNodes... 2.8b/s

모든 DataNodes의... 1

● YARN (MR2 Included) (Cluster 1) 작업 ▾

상태 인스턴스 구성 **모니터링** 애플리케이션 리소스 풀 차트 라이브러리 감사 웹 UI ▾ 쿼리 링크 ▾

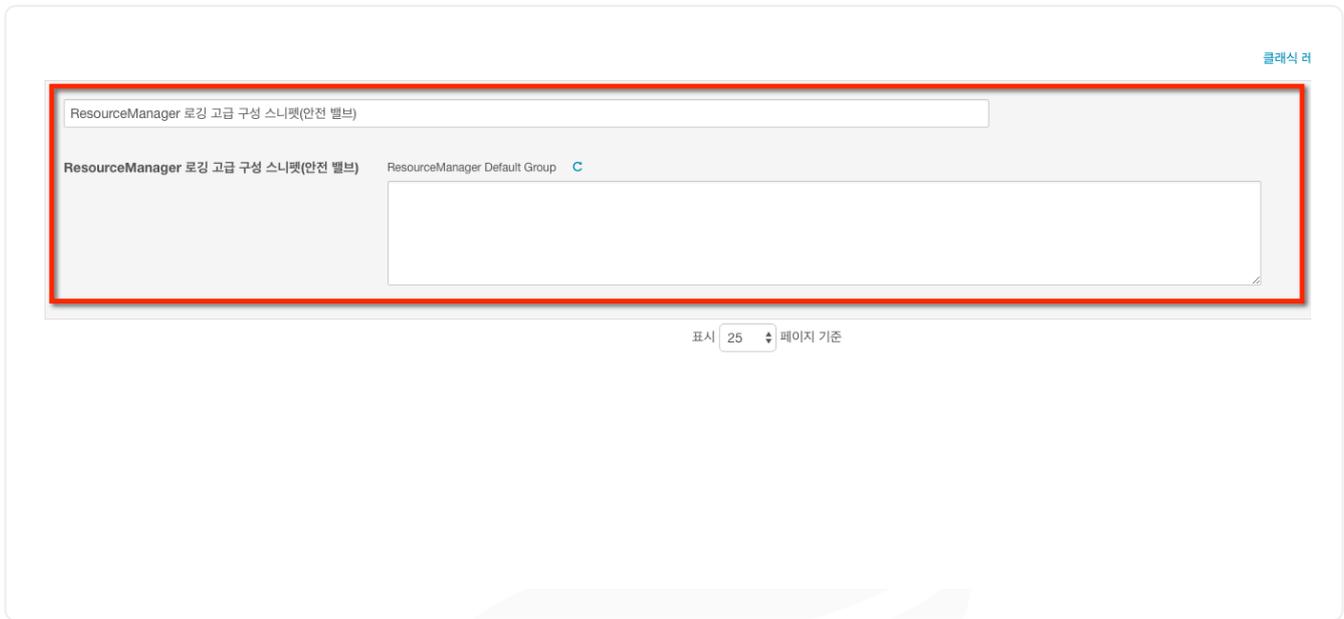
상태 테스트 트리거 생성

- ResourceManager 상태 표시 안 함...  
ResourceManager 요약: cdh1.exem.oss(가용성: 활성, 상태: 주의). 이 상태 테스트는 활성 ResourceManager의 상태를 반영합니다.
- JobHistory Server 상태 표시 안 함...  
JobHistory Server의 상태가 주의 상태입니다. 다음 상태 테스트 결과가 주의입니다. 호스트 상태.
- NodeManager 상태 표시 안 함...  
양호 상태의 NodeManager: 4개. 주의 상태의 NodeManager: 0개. 총 NodeManager: 4개. 양호(%): 100.00%. 양호 또는 주의(%): 100.00%.
- ⊗ YARN Container Usage Aggregation 표시 안 함...  
This health test is disabled because container usage metric collection is disabled for YARN.

상태 요약

## 6.1. Resource Manager log4j.properties 설정하기

log4j.properties 설정을 위해 아래 화면과 같이 ResourceManager 로깅 고급 구성 스크립트(안전 밸브) 키워드 검색을 합니다.



```
# Audit logging for ResourceManager
rm.audit.logger=${hadoop.root.logger}
#log4j.logger.org.apache.hadoop.yarn.server.resourcemanager.RMAuditLogger=${rm.audit.logger}
log4j.logger.org.apache.hadoop.yarn.server.resourcemanager.RMAuditLogger=INFO, RMAUDIT, RMAUDITJMS
log4j.additivity.org.apache.hadoop.yarn.server.resourcemanager.RMAuditLogger=false
log4j.appender.RMAUDIT=org.apache.log4j.DailyRollingFileAppender
log4j.appender.RMAUDIT.File=${yarn.log.dir}/rm-audit.log
log4j.appender.RMAUDIT.layout=org.apache.log4j.PatternLayout
log4j.appender.RMAUDIT.layout.ConversionPattern=%d{ISO8601} %p %c{2}: %m%n
log4j.appender.RMAUDIT.DatePattern=.yyyy-MM-dd
log4j.appender.RMAUDITJMS=org.apache.log4j.net.JMSAppender
log4j.appender.RMAUDITJMS.InitialContextFactoryName=org.apache.activemq.jndi.ActiveMQInitialContextFactory
log4j.appender.RMAUDITJMS.userName=admin ①
log4j.appender.RMAUDITJMS.password=admin ②
log4j.appender.RMAUDITJMS.ProviderURL=tcp://cdhfem.exem.oss:61613?daemon=true ③
log4j.appender.RMAUDITJMS.TopicBindingName=mapredAuditTopic
log4j.appender.RMAUDITJMS.TopicConnectionFactoryBindingName=ConnectionFactory
```

NOTE | 3가지 설정 값들을 잘 확인하여 수정 및 입력해줍니다.

- ① MQ의 username (기본값 admin)
- ② MQ의 password (기본값 admin)
- ③ MQ의 URL (MQ의 IP 주소 및 Port; 기본값 61616)

설정들이 아래의 화면과 같이 잘 반영이 되었다면 `변경내용저장`을 눌러줍니다.

ResourceManager 로깅 고급 구성 스크린샷(안전 벨트)

ResourceManager 로깅 고급 구성 스크린샷(안전 벨트)

ResourceManager Default Group

```
# Audit logging for ResourceManager
rm.audit.logger=${hadoop.root.logger}
#log4j.logger.org.apache.hadoop.yarn.server.resourcemanager.RMAuditLogger=${rm.audit.logger}
log4j.logger.org.apache.hadoop.yarn.server.resourcemanager.RMAuditLogger=INFO, RMAUDIT, RMAUDITJMS
log4j.additivity.org.apache.hadoop.yarn.server.resourcemanager.RMAuditLogger=false
```

표시 25 페이지 기준

## 6.2. JNDI 설정하기

MQ에 메시지를 송신하기 위해서 JNDI Lookup 정보를 추가해야 하며 이를 위해서 \*YARN Resource Manager, Node Manager 서버\*로 접속하여 다음의 파일을 신규로 생성합니다.

- `/opt/cloudera/parcels/CDH/lib/hadoop-yarn/jndi.properties`

```
[root@cdh1 hadoop-yarn]#
[root@cdh1 hadoop-yarn]# ls -la
합계 164
drwxr-xr-x 7 root root 4096 2017-02-15 16:01 .
drwxr-xr-x 37 root root 4096 2017-01-21 06:46 ..
-rw-r--r-- 1 root root 85063 2017-01-21 06:38 LICENSE.txt
-rw-r--r-- 1 root root 14978 2017-01-21 06:38 NOTICE.txt
drwxr-xr-x 2 root root 4096 2017-01-21 06:38 bin
drwxr-xr-x 2 root root 4096 2017-01-21 06:38 cloudera
drwxr-xr-x 2 root root 4096 2017-01-21 06:38 etc
lrwxrwxrwx 1 root root 46 2017-02-03 15:46 hadoop-yarn-api-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-yarn-api-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 35 2017-02-03 15:46 hadoop-yarn-api.jar -> hadoop-yarn-api-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 72 2017-02-03 15:46 hadoop-yarn-applications-distributedshell-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-yarn-applications-distributedshell-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 61 2017-02-03 15:46 hadoop-yarn-applications-distributedshell.jar -> hadoop-yarn-applications-distributedshell-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 77 2017-02-03 15:46 hadoop-yarn-applications-unmanaged-am-launcher-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-yarn-applications-unmanaged-am-launcher-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 66 2017-02-03 15:46 hadoop-yarn-applications-unmanaged-am-launcher.jar -> hadoop-yarn-applications-unmanaged-am-launcher-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 49 2017-02-03 15:46 hadoop-yarn-client-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-yarn-client-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 38 2017-02-03 15:46 hadoop-yarn-client.jar -> hadoop-yarn-client-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 49 2017-02-03 15:46 hadoop-yarn-common-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-yarn-common-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 38 2017-02-03 15:46 hadoop-yarn-common.jar -> hadoop-yarn-common-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 51 2017-02-03 15:46 hadoop-yarn-registry-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-yarn-registry-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 40 2017-02-03 15:46 hadoop-yarn-registry.jar -> hadoop-yarn-registry-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 75 2017-02-03 15:46 hadoop-yarn-server-applicationhistoryservice-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-yarn-server-applicationhistoryservice-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 64 2017-02-03 15:46 hadoop-yarn-server-applicationhistoryservice.jar -> hadoop-yarn-server-applicationhistoryservice-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 56 2017-02-03 15:46 hadoop-yarn-server-common-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-yarn-server-common-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 45 2017-02-03 15:46 hadoop-yarn-server-common.jar -> hadoop-yarn-server-common-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 61 2017-02-03 15:46 hadoop-yarn-server-nodemanager-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-yarn-server-nodemanager-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 50 2017-02-03 15:46 hadoop-yarn-server-nodemanager.jar -> hadoop-yarn-server-nodemanager-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 65 2017-02-03 15:46 hadoop-yarn-server-resourcemanager-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-yarn-server-resourcemanager-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 54 2017-02-03 15:46 hadoop-yarn-server-resourcemanager.jar -> hadoop-yarn-server-resourcemanager-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 55 2017-02-03 15:46 hadoop-yarn-server-tests-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-yarn-server-tests-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 44 2017-02-03 15:46 hadoop-yarn-server-tests.jar -> hadoop-yarn-server-tests-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 59 2017-02-03 15:46 hadoop-yarn-server-web-proxy-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-yarn-server-web-proxy-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 48 2017-02-03 15:46 hadoop-yarn-server-web-proxy.jar -> hadoop-yarn-server-web-proxy-2.6.0-cdh5.10.0.jar
-rw-r--r-- 1 root root 76 2017-02-07 14:05 jndi.properties
drwxr-xr-x 2 root root 4096 2017-02-06 17:58 lib
drwxr-xr-x 2 root root 4096 2017-01-21 06:38 sbin
[root@cdh1 hadoop-yarn]#
[root@cdh1 hadoop-yarn]# pwd
/opt/cloudera/parcels/CDH/lib/hadoop-yarn
[root@cdh1 hadoop-yarn]#
[root@cdh1 hadoop-yarn]#
```

새로 생성한 상기 파일에 다음 내용을 추가하고 저장합니다.



## Chapter 7. HDFS 모니터링 설정하기

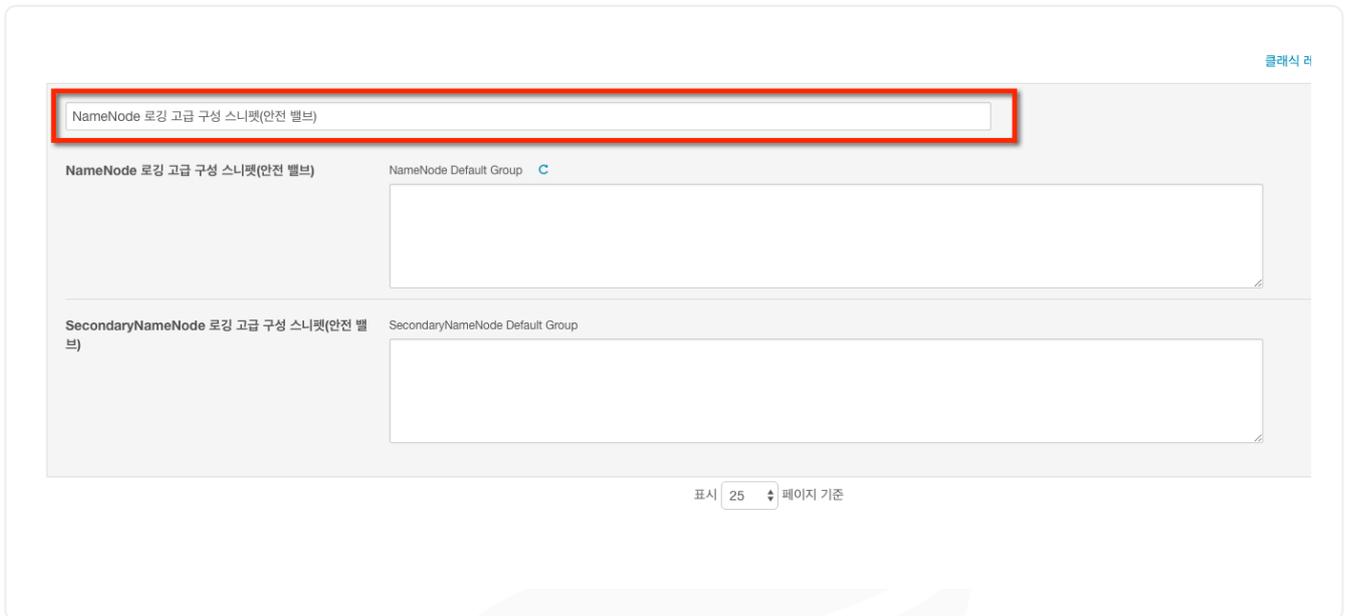
HDFS 모니터링은 HDFS의 각종 이벤트를 수집해서 취합하여 Flamingo에서 정보를 표시합니다. MapReduce 모니터링과 마찬가지로 HDFS 모니터링을 설정하는 방법은 CDH Manager로 설정하는 방법과 Namenode의 `log4j.properties` 파일을 직접 수정하는 방법이 있으며 CDH Manager로 설정하는 것을 권장합니다. 본 설치 가이드에서는 직접 수정하는 방법은 설명하지 않습니다.

CDH Manager로 로그인하여 'CDH Manager > HDFS > 구성' 메뉴를 선택합니다.

The screenshot displays the Cloudera Manager interface. On the left, a sidebar lists services for 'Cluster 1 (CDH 5.10.0, Parcel)'. The 'HDFS' service is highlighted with a red box. Below this, the 'Cloudera Management Service' section shows 'Cloudera Mana...' with 4 alerts and 4 warnings. On the right, two charts are visible: '클러스터 CPU' (Cluster CPU) showing 1.2% usage and 'HDFS IO' showing 2.8b/s. The main content area shows the 'HDFS (Cluster 1)' configuration page. The '구성' (Configuration) tab is selected and highlighted with a red box. Below the navigation bar, the 'HDFS 요약' (HDFS Summary) section shows a progress bar for '구성 용량' (Configuration Capacity) at 3.0GiB/483.1GiB. The '상태 테스트' (Status Test) section includes a '트리거 생성' (Generate Trigger) button and a 'NameNode 상태' (NameNode Status) section with a '표시 안 함...' (Hide...) link. The '6 양호 표시' (6 Good Indicators) section is also visible.

## 7.1. NameNode log4j.properties 설정하기

다음과 같이 NameNode 로깅 고급 구성 스니펫(안전 밸브) 검색을 합니다.



MQ를 통해 MapReduce 실행 이벤트를 수집하기 위해서 다음과 같이 `NameNode Default Group`에 아래 내용을 추가합니다.

```
... 생략
#
# hdfs audit logging
#
hdfs.audit.logger=INFO,console
#log4j.logger.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit=${hdfs.audit.logger} ①
log4j.logger.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit=INFO,DRFAAUDIT,DRFAAUDITJMS ②
log4j.additivity.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit=false
log4j.appender.DRFAAUDIT=org.apache.log4j.DailyRollingFileAppender
log4j.appender.DRFAAUDIT.File=${hadoop.log.dir}/hdfs-audit.log
log4j.appender.DRFAAUDIT.layout=org.apache.log4j.PatternLayout
log4j.appender.DRFAAUDIT.layout.ConversionPattern=%d{ISO8601} %p %c{2}: %m%n
log4j.appender.DRFAAUDIT.DatePattern=.yyyy-MM-dd

log4j.appender.DRFAAUDITJMS=org.apache.log4j.net.JMSAppender ③
log4j.appender.DRFAAUDITJMS.InitialContextFactoryName=org.apache.activemq.jndi.ActiveMQInitialContextFactory ③
log4j.appender.DRFAAUDITJMS.userName=admin ③
log4j.appender.DRFAAUDITJMS.password=admin ③
log4j.appender.DRFAAUDITJMS.ProviderURL=tcp://fem.exem.oss:61616?daemon=true ③
log4j.appender.DRFAAUDITJMS.TopicBindingName=hdfsAuditTopic ③
log4j.appender.DRFAAUDITJMS.TopicConnectionFactoryBindingName=ConnectionFactory ③
... 생략
```

- ① 기존의 **audit** 설정을 주석처리합니다.
- ② 새로운 **audit** 설정을 추가합니다.
- ③ MQ의 username, password, URL을 추가합니다.

### NOTE

HDFS 모니터링 설정은 Namenode HA가 설정되어 있는 경우 Active Namenode, Standby Namenode 모두 설정하도록 하며, Secondary Namenode에는 설정하지 않습니다.

설정들이 아래의 화면과 같이 잘 반영이 되었다면 `변경내용저장`을 눌러줍니다.

클래식 러

NameNode 로깅 고급 구성 스크린샷(안전 벨트)

NameNode 로깅 고급 구성 스크린샷(안전 벨트)

SecondaryNameNode 로깅 고급 구성 스크린샷(안전 벨트)

NameNode Default Group C

SecondaryNameNode Default Group

표시 25 페이지 기준

## 7.2. JNDI 설정하기

MQ에 메시지를 송신하기 위해서 JNDI Lookup 정보를 추가해야 하며 이를 위해서 \*NameNode, DataNode 서버\*로 접속하여 다음의 파일을 신규로 생성합니다.

- `/opt/cloudera/parcels/CDH/lib/hadoop-hdfs/jndi.properties`

```
[root@cdh1 hadoop-hdfs]#
[root@cdh1 hadoop-hdfs]#
[root@cdh1 hadoop-hdfs]# ls -la
합계 132
drwxr-xr-x 7 root root 4096 2017-02-15 17:55 .
drwxr-xr-x 37 root root 4096 2017-01-21 06:46 ..
-rw-r--r-- 1 root root 85063 2017-01-21 06:38 LICENSE.txt
-rw-r--r-- 1 root root 14978 2017-01-21 06:38 NOTICE.txt
drwxr-xr-x 2 root root 4096 2017-02-08 18:22 bin
drwxr-xr-x 2 root root 4096 2017-01-21 06:38 cloudera
lrwxrwxrwx 1 root root 48 2017-02-03 15:46 hadoop-hdfs-2.6.0-cdh5.10.0-tests.jar -> ../../jars/hadoop-hdfs-2.6.0-cdh5.10.0-tests.jar
lrwxrwxrwx 1 root root 42 2017-02-03 15:46 hadoop-hdfs-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-hdfs-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 46 2017-02-03 15:46 hadoop-hdfs-nfs-2.6.0-cdh5.10.0.jar -> ../../jars/hadoop-hdfs-nfs-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 35 2017-02-03 15:46 hadoop-hdfs-nfs.jar -> hadoop-hdfs-nfs-2.6.0-cdh5.10.0.jar
lrwxrwxrwx 1 root root 37 2017-02-03 15:46 hadoop-hdfs-tests.jar -> hadoop-hdfs-2.6.0-cdh5.10.0-tests.jar
lrwxrwxrwx 1 root root 31 2017-02-03 15:46 hadoop-hdfs.jar -> hadoop-hdfs-2.6.0-cdh5.10.0.jar
-rw-r--r-- 1 root root 76 2017-02-07 14:28 jndi.properties
drwxr-xr-x 2 root root 4096 2017-02-08 18:01 lib
drwxr-xr-x 2 root root 4096 2017-01-21 06:38 sbin
drwxr-xr-x 8 root root 4096 2017-01-21 06:38 webapps
[root@cdh1 hadoop-hdfs]#
[root@cdh1 hadoop-hdfs]#
[root@cdh1 hadoop-hdfs]# pwd
/opt/cloudera/parcels/CDH/lib/hadoop-hdfs
[root@cdh1 hadoop-hdfs]#
```

새로 생성한 상기 파일에 다음 내용을 추가하고 저장합니다.



## Chapter 8. Apache Hive 모니터링 설정하기

### 8.1. Hive JMX(Hiveserver2) 설정 하기

Hive JMX를 통해 Hive Server2의 메모리를 모니터링 하며 Hive Server 2의 JMX를 활성화하도록 합니다.

CDH 5.x 기준 `/opt/cloudera/parcels/CDH/lib/hive/bin/ext/hiveserver2.sh` 파일을 열어 아래 번호표기 되어 추가 된 내용과 같이 추가합니다.

```
... 생략
hiveserver2() {
  echo "Starting Hive Server 2"
  CLASS=org.apache.hive.service.server.HiveServer2
  if $cygwin; then
    HIVE_LIB=`cygpath -w "$HIVE_LIB"`
  fi
  JAR=${HIVE_LIB}/hive-service-*.jar

  export HADOOP_OPTS="$HADOOP_OPTS -Dcom.sun.management.jmxremote"
  export HADOOP_OPTS="$HADOOP_OPTS -Dcom.sun.management.jmxremote.authenticate=false"
  export HADOOP_OPTS="$HADOOP_OPTS -Dcom.sun.management.jmxremote.ssl=false"
  export HADOOP_OPTS="$HADOOP_OPTS -Dcom.sun.management.jmxremote.port=9004"
  export HADOOP_OPTS="$HADOOP_OPTS -Djava.rmi.server.hostname=cdh1.exem.oss" ①
  export HADOOP_OPTS="$HADOOP_OPTS -javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-hv-3.0.3
-SNAPSHOT.jar" ②

  echo "$HADOOP jar $JAR $CLASS $HIVE_OPTS" ③

  exec $HADOOP jar $JAR $CLASS $HIVE_OPTS "$@" ④
}

hiveserver2_help() {
  hiveserver2 -H
}
```

① "~ hostname=" 부분은 반드시 Hive2가 설치되어 있는 hostname 으로 입력을 해주어야 합니다.

ex) hostname=cdh01.exem.oss

② "~ javaagent:" 부분은 hive agent jar가 설치 되어 있는 경로로 입력해주어야 합니다

ex) javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-hv-3.0.3-SNAPSHOT.jar

③ 마지막으로 echo "\$HADOOP jar \$JAR \$CLASS \$HIVE\_OPTS" 내용을 추가해줍니다.

### 8.2. Hive Agent

하나의 Agent로 Hive Server, Hive CLI를 동시에 모니터링 합니다.

`/etc/fem/agent.conf` 파일을 열어서 설치된 activemq 정보에 맞게 수정합니다.

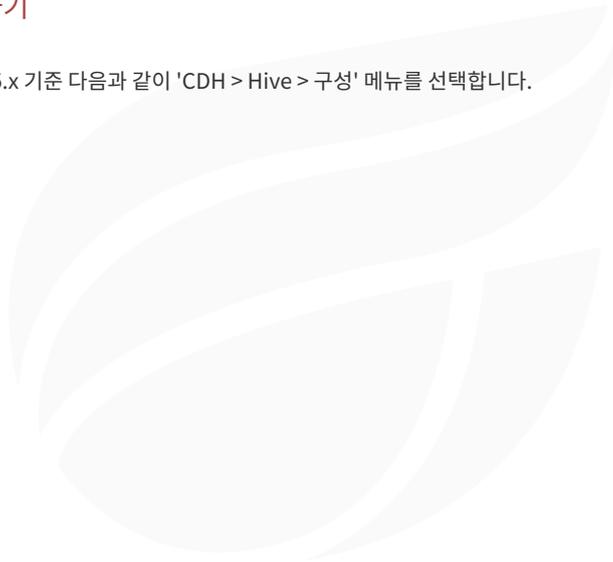


```
activemq.broker-url=tcp://fem.exem.oss:61616 ①  
activemq.user=admin ②  
activemq.password=admin ③  
  
collector.host=fem.exem.oss ④  
collector.port=38081 ⑤
```

- ① activemq 가 설치된 주소와 포트 정보를 입력합니다.
- ② activemq 의 사용자명을 입력합니다.
- ③ activemq 의 패스워드를 입력합니다.
- ④ FEM collector의 host명을 입력합니다.
- ⑤ FEM port를 입력합니다.

### 8.3. Hive CLI 모니터링 설정하기

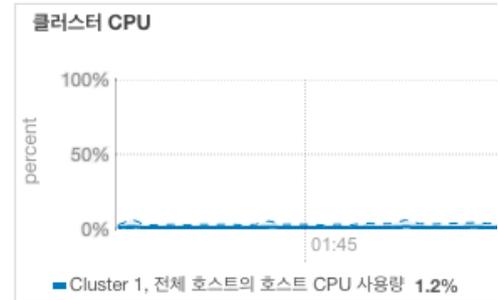
Hive CLI 모니터링 설정을 위해서 CDH 5.x 기준 다음과 같이 'CDH > Hive > 구성' 메뉴를 선택합니다.



You are running Cloudera Manager in non-production mode, which uses an embedded PostgreSQL database. Switch to production mode.

- Cluster 1 (CDH 5.10.0, Parcel)
- 호스트
- HBase
- HDFS 1
- Hive**
- Hue
- Oozie
- Spark

차트



Hive (Cluster 1) 작업

상태 테스트

트리거 생성

- HiveServer2 상태** 표시 안 함...  
양호 상태의 HiveServer2: 0개. 주의 상태의 HiveServer2: 1개. 총 HiveServer2: 1개. 양호(%): 0.00%. 양호 또는 주의(%): 100.00%. 경고 임계값: 99.00%.
- Hive Metastore Server 상태** 표시 안 함...  
양호 상태의 Hive Metastore Server: 0개. 주의 상태의 Hive Metastore Server: 1개. 총 Hive Metastore Server: 1개. 양호(%): 0.00%. 양호 또는 주의(%): 100.00%. 경고 임계값: 99.00%.

상태 요약

Gateway 5 없음

hive-env.sh 를 수정하기 위해 다음과 같이 hive-env.sh에 대한 Gateway 클라이언트 환경 고급 구성 스크립트(안전 밸브) 검색합니다.

hive-env.sh에 대한 Gateway 클라이언트 환경 고급 구성 스니펫(안전 밸브)

hive-env.sh에 대한 Gateway 클라이언트 환경 고급 구성 스니펫(안전 밸브)

Gateway Default Group C

표시 25 페이지 기준

### 8.3.1. hive-env.sh에 대한 Gateway 클라이언트 환경 고급 구성 스니펫(안전 밸브) 설정하기

```
HADOOP_CLIENT_OPTS="$HADOOP_CLIENT_OPTS -javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-hv-3.0.3-SNAPSHOT.jar"
```

NOTE **-javaagent:/opt** ~로 시작하는 Hive Agent 경로를 유의하여 입력해 줍니다.

설정들이 아래의 화면과 같이 잘 반영이 되었다면 `변경내용저장`을 눌러줍니다.

hive-env.sh에 대한 Gateway 클라이언트 환경 고급 구성 스니펫(안전 밸브)

hive-env.sh에 대한 Gateway 클라이언트 환경 고급 구성 스니펫(안전 밸브)

Gateway Default Group C

```
HADOOP_CLIENT_OPTS="$HADOOP_CLIENT_OPTS -javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-hv-3.0.1-SNAPSHOT.jar"
```

표시 25 페이지 기준

## Chapter 9. 서비스 재시작 하기

수정된 사항 적용을 위하여 CDH Manager에서 서비스들을 재시작 합니다.

The screenshot shows the Cloudera Manager interface for 'Cluster 1 (CDH 5.10.0, Parcel)'. A dropdown menu is open, listing various actions. The '재시작' (Restart) option is highlighted with a red box and a mouse cursor. Other options include '서비스 추가', '시작', '중지', '클러스터 재시작', '클라이언트 구성 배포', 'Kerberos 클라이언트 구성 배포', '클러스터 업그레이드', '클러스터 새로고침', '동적 리소스 풀 새로고침', 'Inspect Hosts in Cluster', 'Kerberos 설정', 'HDFS 데이터 저장 암호화 설정', and '클라이언트 구성 URL 보기'. To the right, there are two charts: '클러스터 CPU' showing 1.1% usage and 'HDFS IO' showing a spike in activity.

설정을 정확히 했다면 문제 없이 서비스들이 재시작 됩니다.

The screenshot shows a confirmation dialog box titled '재시작 명령' (Restart Command). The status is '완료됨' (Completed) for '클러스터: Cluster 1'. The start time is 'Feb 16, 4:39:13 PM' and the duration is '3.2m'. The message states 'All services successfully restarted.' Below this, there is a table with the following data:

단계	컨텍스트	시작 시간	소요 시간	작업
<ul style="list-style-type: none"> <li>클러스터의 모든 서비스를 중지합니다. All services successfully stopped.</li> </ul>	Cluster 1	Feb 16, 4:39:13 PM	34.13s	
<ul style="list-style-type: none"> <li>클러스터의 모든 서비스를 시작합니다. All services successfully started.</li> </ul>	Cluster 1	Feb 16, 4:39:47 PM	2.7m	

The dialog also includes a '닫기' (Close) button at the bottom right.

## Chapter 10. 시스템 리소스 모니터링

Flamingo System Agent는 모니터링할 모든 노드에 설치해야 하며 CPU, Core, RAM, Disk, Network 등의 정보를 실시간으로 수집하는 역할을 합니다. 따라서 원활하게 정보를 수집하기 위해서는 root 계정으로 실행할 것을 권고합니다.

### 10.1. System Agent 시작하기

리눅스의 서비스로 등록한 후 다음의 커맨드로 서비스를 실행합니다. 다음의 커맨드는 root로 실행하도록 합니다.

```
$ service flamingo-system-agent start
```

### 10.2. System Agent 옵션

`/etc/init.d` 경로에 등록된 System Agent의 관리 옵션은 다음과 같습니다. 이 옵션은 `service flamingo-system-agent <옵션>` 커맨드를 실행할 때 사용하는 옵션입니다.

옵션	설명	비고
<code>start</code>	System Agent를 시작합니다.	
<code>stop</code>	System Agent를 중지합니다.	
<code>restart</code>	System Agent를 재시작합니다.	
<code>force-reload</code>	System Agent를 시작합니다.	
<code>status</code>	System Agent의 상태를 확인합니다.	
<code>run</code>	System Agent를 foreground로 실행합니다.	

### 10.3. System Agent의 MQ URL 변경하기

System Agent는 수집한 정보를 MQ로 송신합니다.

이때 환경설정 파일을 `/etc/fem/agent.conf` 열어 아래와 같이 변경하도록 합니다.

NOTE | 파일이 없다면 만듭니다.

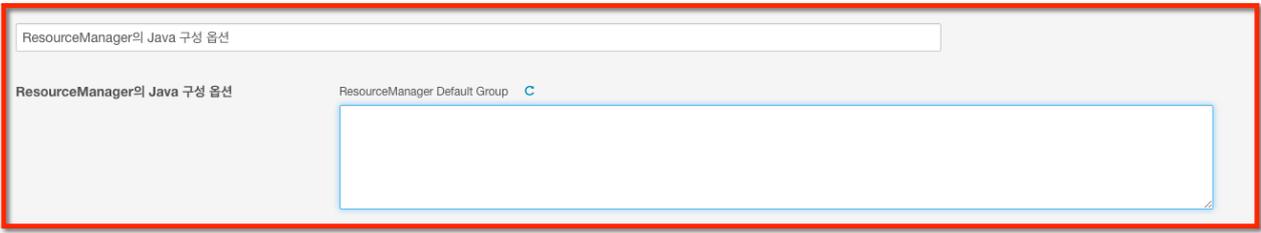
```
activemq.broker-url=tcp://fem.exem.oss:61616 ①  
activemq.user=admin ②  
activemq.password=admin ③  
  
collector.host=fem.exem.oss ④  
collector.port=38081 ⑤
```

- ① activemq 가 설치된 주소와 포트 정보를 입력합니다.
- ② activemq 의 사용자명을 입력합니다.
- ③ activemq 의 패스워드를 입력합니다.
- ④ FEM collector의 host명을 입력합니다.
- ⑤ FEM port를 입력합니다.



## Chapter 11. Resource Manager Agent 설정 하기

아래 화면과 같이 Resource Manager의 Java 구성 옵션 키워드 검색을 통해 Resource Manager의 Java 구성옵션 설정을 찾습니다.



클래식 리

ResourceManager의 Java 구성 옵션

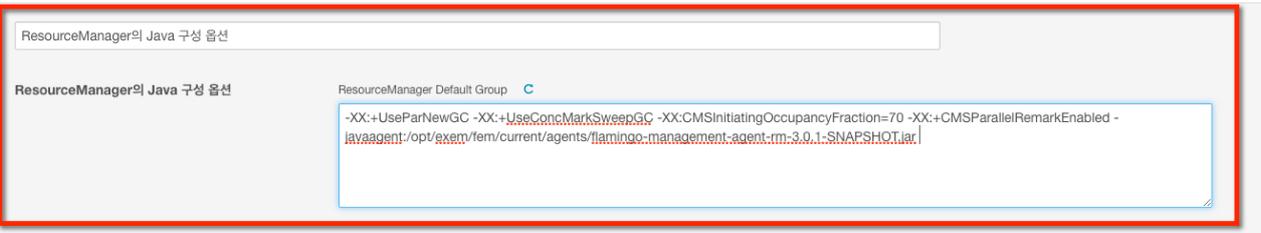
ResourceManager Default Group C

표시 25 페이지 기준

```
-XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=70 -XX:+CMSParallelRemarkEnabled  
-javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-rm-3.0.3-SNAPSHOT.jar
```

NOTE **-javaagent:/opt** ~로 시작하는 Resource Manager Agent 경로를 유의하여 입력해 줍니다.

위의 설정이 아래와 같이 잘 반영이 되었다면 `변경내용저장`을 눌러줍니다.



클래식 리

ResourceManager의 Java 구성 옵션

ResourceManager Default Group C

```
-XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=70 -XX:+CMSParallelRemarkEnabled -  
javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-rm-3.0.1-SNAPSHOT.jar
```

표시 25 페이지 기준

## Chapter 12. Namenode Agent 설정 하기

hdfs가 실행 되는 쉘스크립트를 수정하기 위해 CDH 5.x 기준 `/opt/cloudera/parcels/CDH/lib/hadoop-hdfs/bin` 경로로 이동합니다.

```
[root@cdh1 bin]#  
[root@cdh1 bin]#  
[root@cdh1 bin]# pwd  
/opt/cloudera/parcels/CDH/lib/hadoop-hdfs/bin  
[root@cdh1 bin]#  
[root@cdh1 bin]# ls -la  
합계 24  
drwxr-xr-x 2 root root 4096 2017-02-16 16:16 .  
drwxr-xr-x 7 root root 4096 2017-02-15 17:55 ..  
-rwxr-xr-x 1 root root 12497 2017-02-16 16:16 hdfs
```

해당 경로로 이동하여 `hdfs` 파일을 아래와 같이 수정 합니다.

```
1 ... 생략  
2 if [ "$COMMAND" = "namenode" ] ; then  
3   CLASS='org.apache.hadoop.hdfs.server.namenode.NameNode'  
4   # HADOOP_OPTS="$HADOOP_OPTS $HADOOP_NAMENODE_OPTS" ①  
5   HADOOP_OPTS="$HADOOP_OPTS -javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-nn-3.0.3  
-SNAPSHOT.jar $HADOOP_NAMENODE_OPTS" ②  
6 elif [ "$COMMAND" = "zkfc" ] ; then  
7   CLASS='org.apache.hadoop.hdfs.tools.DFSZKFailoverController'  
8   HADOOP_OPTS="$HADOOP_OPTS $HADOOP_ZKFC_OPTS"  
9 elif [ "$COMMAND" = "secondarynamenode" ] ; then  
10 ... 생략
```

① 기존 설정되어 있는 값을 위와 같이 주석(#) 처리 해줍니다.

② 위의 원본 파라미터를 복사하여 위와 같이 javaagent가 설치된 실제 경로를 추가 합니다.

위의 설정이 아래와 같이 잘 반영이 되었다면 `변경내용저장`을 눌러줍니다.

```
then  
  export SENTRY_HOME=`readlink -m ${HADOOP_HDFS_HOME}/../sentry`  
fi  
fi  
if [[ -n "$SENTRY_HOME" ]]  
then  
  for f in ${SENTRY_HOME}/lib/plugins/*.jar; do  
    CLASSPATH=${CLASSPATH}:${f}  
  done  
fi  
#HADOOP_OPTS="$HADOOP_OPTS $HADOOP_NAMENODE_OPTS"  
HADOOP_OPTS="$HADOOP_OPTS -javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-nn-3.0.1-SNAPSHOT.jar $HADOOP_NAMENODE_OPTS"  
elif [ "$COMMAND" = "zkfc" ] ; then  
  CLASS='org.apache.hadoop.hdfs.tools.DFSZKFailoverController'  
  HADOOP_OPTS="$HADOOP_OPTS $HADOOP_ZKFC_OPTS"  
elif [ "$COMMAND" = "secondarynamenode" ] ; then  
  CLASS='org.apache.hadoop.hdfs.server.namenode.SecondaryNameNode'  
  # Set SENTRY_HOME if possible and add Sentry plugin jars to classpath  
  if [[ -z "$SENTRY_HOME" ]]  
  then  
    # HADOOP_HDFS_HOME should have been set by hadoop-config.sh  
    if [[ -d ${HADOOP_HDFS_HOME}/../sentry ]]
```

## Chapter 13. Flamingo Collector 설정 하기

Collector 설치되어 있는 서버에 접속해서 `/opt/exem/fem/current/collector/webapps/ROOT/WEB-INF/classes` 경로로 이동합니다.

```
/opt/exem/fem/current/collector/webapps/ROOT/WEB-INF/classes
[root@cdhfem classes]#
[root@cdhfem classes]# ls -la
합계 180
drwxr-xr-x 4 exem exem 4096 2017-02-09 18:32 .
drwxr-xr-x 6 exem exem 4096 2017-02-06 16:48 ..
drwxr-xr-x 2 exem exem 4096 2017-02-06 16:48 META-INF
-rw-r--r-- 1 exem exem 1304 2017-02-09 16:39 application.properties
-rw-r--r-- 1 exem exem 0 2017-02-09 10:32 collector-dev.properties
-rw-r--r-- 1 root root 0 2017-02-09 10:18 collector-prd.properties
-rw-r--r-- 1 exem exem 4718 2017-02-09 17:16 collector.properties
drwxr-xr-x 3 exem exem 4096 2017-02-03 10:28 com
-rw-r--r-- 1 exem exem 6752 2017-02-03 10:28 flamingo-logo.jpg
-rw-r--r-- 1 exem exem 74984 2017-02-03 10:28 flamingo-logo.png
-rw-r--r-- 1 exem exem 28955 2017-02-03 10:32 flamingo-logo.txt
-rw-r--r-- 1 exem exem 28943 2017-02-03 10:32 flamingo-logo_invert.txt
-rw-r--r-- 1 exem exem 1921 2017-02-03 10:32 logback.xml
```

### 13.1. application.properties 파일 설정 하기

...생략

```
#####
# banner config #
#####

# 배너를 설정 한다.
# 리소스안에 flamingo-logo.jpg를 로고로 하여 최초 구동시 로고가 텍스트로 변환되어 표시된다.
spring.main.banner-mode=log
banner.location=classpath:flamingo-logo_invert.txt
#banner.image.location=flamingo-logo.png
#banner.image.invert=true
#banner.image.width=115
#banner.image.margin=0

#####
# tomcat config #
#####

# 내장 톰캣 사용시 설정
# 포트와 컨텍스트 패스를 지정한다.
server.port=38081 ①
server.context-path=/

#####
# actuator #
#####

# 모니터링 설정
# 기본 제공 모니터링은 /autoconfig, /info, /mappings, /beans, /env, /metrics, /trace, /heapdump, /configgroups,
/health, /dump
# reference http://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#production-ready-endpoints
endpoints.actuator.enabled=true
info.app.encoding=UTF-8
info.app.java.source=1.8.0_91
info.app.java.target=1.8.0_91

... 생략
```

① Collector에서 사용할 포트 번호를 입력 합니다.

collector.properties 수정하기 위해 /opt/exem/fem/current/collector/webapps/ROOT/WEB-INF/classes 경로로 이동합니다.

```
/opt/exem/fem/current/collector/webapps/ROOT/WEB-INF/classes
[root@cdhfem classes]#
[root@cdhfem classes]# ls -la
합계 180
drwxr-xr-x 4 exem exem 4096 2017-02-09 18:32 .
drwxr-xr-x 6 exem exem 4096 2017-02-06 16:48 ..
drwxr-xr-x 2 exem exem 4096 2017-02-06 16:48 META-INF
-rw-r--r-- 1 exem exem 1304 2017-02-09 16:39 application.properties
-rw-r--r-- 1 exem exem 0 2017-02-03 10:32 collector-dev.properties
-rw-r--r-- 1 root root 0 2017-02-09 10:18 collector-pwd.properties
-rw-r--r-- 1 exem exem 4718 2017-02-09 17:16 collector.properties
drwxr-xr-x 3 exem exem 4096 2017-02-03 10:28 com
-rw-r--r-- 1 exem exem 6752 2017-02-03 10:28 flamingo-logo.jpg
-rw-r--r-- 1 exem exem 74984 2017-02-03 10:28 flamingo-logo.png
-rw-r--r-- 1 exem exem 28955 2017-02-03 10:32 flamingo-logo.txt
-rw-r--r-- 1 exem exem 28943 2017-02-03 10:32 flamingo-logo_invert.txt
-rw-r--r-- 1 exem exem 1921 2017-02-03 10:32 logback.xml
```

### 13.2. collector.properties 파일 설정 하기

```
#####
# embedded zookeeper #
#####
#Deprecated
# 내장된 zookeeper를 사용할때 설정.
# 내장 hbase가 활성화 되어있으면 자동으로 사용한다.
collector.embedded.zookeeper.port=2181
collector.embedded.zookeeper.maxConnection=10
collector.embedded.zookeeper.snap.dir=${user.home}/flamingo/zookeeper/snap
collector.embedded.zookeeper.log.dir=${user.home}/flamingo/zookeeper/log
collector.embedded.zookeeper.tickTime=1000

#####
# embedded hbase #
#####
#Deprecated
# 내장된 hbase를 사용할때 true. (true | false )
# hbase의 데이터가 저장될 경로 및 zookeeper 설정, znode를 설정 한다.
collector.embedded.hbase.enable=false
collector.embedded.hbase.zookeeper.quorum=127.0.0.1
collector.embedded.hbase.zookeeper.port=2181
collector.embedded.hbase.zookeeper.znode.parent=/hbase
collector.embedded.hbase.data.dir=${user.home}/flamingo/hbase

#####
# embedded apollo #
#####
#Deprecated
# 내장된 mq apollo를 사용할때 true. (true | false )
# 포트를 지정하고 levelDB의 데이터가 저장될 경로를 지정.
collector.embedded.apollo.enable=false
collector.embedded.apollo.port=61616
collector.embedded.apollo.leveldb.dir=${user.home}/flamingo/leveldb

#####
# activemq #
#####
# activemq 관련 설정
# broker-url과 사용자 계정 정보를 설정 한다. activemq 기본값은 admin / admin
activemq.broker-url=tcp://cdhfem.exem.oss:61616?daemon=true ①
activemq.user=admin ②
activemq.password=admin ③

#####
# hdfs api #
```

```

#####
#Deprecated
# hdfs api를 호출 하기 위한 host, port, username을 설정.
collector.hdfs.api.host=cdh01.exem.oss ④
collector.hdfs.api.port=50070 ⑤
collector.hdfs.api.defaultUrl=/webhdfs/v1
collector.hdfs.api.usage.op=GETCONTENTSUMMARY
collector.hdfs.api.username=hdfs

#####
# oozie api #
#####
# oozie metrics를 수집 한다. (true | false)
# metrics 를 수집하기 위한 host, port, context, path등을 설정 한다.
oozie.metrics.enable=false
oozie.server.url=http://cdh01.exem.oss:11000/oozie ⑥
oozie.server.metrics=/v2/admin/metrics ⑦
oozie.server.jobinfo=/v2/job/

#####
# spark log #
#####
# spark 히스토리 로그를 이용해 수집하기 위한 spark history log 경로를 지정 한다. (hdfs 경로)
spark.log.dir=/spark-history ⑧

#####
# hive jmx #
#####
# Hive Metastore Agent
hive.metastore.url=thrift://cdh2.exem.oss:9083

# hive jmx 셋팅 시 지정한 jmxremote port와 hostname에 맞게 설정 한다.
hive.hiveserver2.host=cdh01.exem.oss ⑨
hive.hiveserver2.jmx.url=service:jmx:rmi://cdh01.exem.oss:9004/jndi/rmi://cdh01.exem.oss:9004/jmxrmi ⑩

#####
# namenode #
#####

# namenode ha 사용 여부에 따라 zookeeper 또는 host,port 를 설정 한다.
# 설치되어 있는 namenode agent의 port 정보를 입력한다.
namenode.ha=false ⑪
namenode.host=cdh01.exem.oss ⑫
namenode.port=8020 ⑬
namenode.znode=/hadoop-ha/namenode-ha/ActiveBreadCrumb ⑭
namenode.agent.port=10070 ⑮

#####
# resource manager #
#####

# resource manager 의 host명, port
resourcemanager.ha=false
resourcemanager.host=cdh01.exem.oss ⑯
resourcemanager.port=8088 ⑰
resourcemanager.znode=/yarn-leader-election/yarn-cluster/ActiveBreadCrumb
#####
# phoenix datasource #
#####

# phoenix-hbase 의 datasource.
# jdbcurl 정보를 입력한다.
datasource.phoenix.driver-class-name=org.apache.phoenix.jdbc.PhoenixDriver
datasource.phoenix.jdbcUrl=jdbc:phoenix:cdhfem.exem.oss:2181:/fem-hbase ⑱
datasource.phoenix.type=com.zaxxer.hikari.HikariDataSource
datasource.phoenix.connection-test-query=SELECT 1 FROM SYSTEM.CATALOG LIMIT 1

```

```
#####
# flamingo datasource #
#####

# 플라밍고와 WEB과 함께 쓰는 datasource.
# jdbc 정보와 classname, 사용자명, 비밀번호를 입력한다.
datasource.flamingo.driver-class-name=org.postgresql.Driver
datasource.flamingo.jdbcUrl=jdbc:postgresql://cdhfem.exem.oss:5432/flamingo ⑱
datasource.flamingo.type=com.zaxxer.hikari.HikariDataSource
datasource.flamingo.username=flamingo
datasource.flamingo.password=flamingo
datasource.flamingo.min.pool=3
datasource.flamingo.max.pool=10
datasource.flamingo.connection-test-query=select 1

#####
# h2 datasource #
#####

# System Metrics 정보를 임시로 저장하기 위해 사용하는 Cache DB
datasource.system.driver-class-name=org.h2.Driver
datasource.system.jdbcUrl=jdbc:h2:tcp:cdhfem.exem.oss:9092/mem:flamingo
datasource.system.type=com.zaxxer.hikari.HikariDataSource
datasource.system.username=sa
datasource.system.password=sa

#####
# Zookeeper #
#####
zookeeper.quorum=cdh1.exem.oss
```

- ① 설치된 Activemq의 broker-url 정보를 입력합니다.
- ② 설치된 Activemq의 사용자 아이디를 입력합니다.
- ③ 설치된 Activemq의 사용자 비밀번호를 입력합니다.
- ④ namenode 호스트명을 입력합니다.
- ⑤ namenode 포트를 입력합니다.
- ⑥ oozie-server의 호스트명을 입력합니다.
- ⑦ oozie-server의 포트를 입력합니다.
- ⑧ spark 설정값 중 spark.history.fs.logDirectory 값으로 지정된 디렉토리를 입력합니다.
- ⑨ hive jmx 세팅 시 지정한 jmx remote url 을 입력합니다.
- ⑩ hive-sever2 의 호스트명을 입력합니다.
- ⑪ namenode ha 구성여부에 따라 true/false 로 입력합니다.
- ⑫ namenode 구성여부가 false 일 경우 namenode의 호스트명을 입력합니다.
- ⑬ namenode 구성여부가 false 일 경우 namenode의 포트를 입력합니다.
- ⑭ namenode ha 구성이 true 일 경우 zookeeper의 namenode znode 를 입력합니다.
- ⑮ namenode agent 의 포트를 입력합니다.
- ⑯ resource manager 의 호스트명을 입력합니다.
- ⑰ resource manager 의 포트를 입력합니다.

⑱ peonix jdbc url 정보를 입력합니다.

⑲ flamingo DB 가 설치된 jdbc경로를 입력합니다.



## Chapter 14. Flamingo Web 설정 하기

Flamingo Web이 설치되어 있는 서버에 접속해서 `/opt/exem/fem/current/web/webapps/ROOT/WEB-INF` 경로로 이동합니다.

```
[root@cdhfem WEB-INF]# pwd
/opt/exem/fem/current/web/webapps/ROOT/WEB-INF
[root@cdhfem WEB-INF]#
[root@cdhfem WEB-INF]# ls -la
합계 128
drwxr-xr-x 5 exem exem 4096 2017-02-16 16:54 .
drwxr-xr-x 5 exem exem 4096 2017-02-06 16:47 ..
-rw-r--r-- 1 exem exem 274 2017-02-03 10:33 app.properties
-rw-r--r-- 1 exem exem 1618 2017-02-03 10:26 applicationContext-password.xml
-rw-r--r-- 1 exem exem 1165 2017-01-04 09:52 applicationContext-reactor.xml
-rw-r--r-- 1 exem exem 7226 2017-02-03 10:26 applicationContext-repository-fake.xml
-rw-r--r-- 1 exem exem 7796 2017-01-13 11:13 applicationContext-repository.xml
-rw-r--r-- 1 exem exem 3979 2017-02-03 10:26 applicationContext-security.xml
-rw-r--r-- 1 exem exem 1722 2017-02-03 10:26 applicationContext.xml
drwxr-xr-x 4 exem exem 4096 2017-02-06 16:48 classes
-rw-r--r-- 1 exem exem 6864 2017-02-09 10:12 config.properties
-rw-r--r-- 1 exem exem 8004 2017-01-04 09:52 dispatcher-service.xml
-rw-r--r-- 1 exem exem 2008 2017-02-10 14:25 hadoop.properties
drwxr-xr-x 2 exem exem 20480 2017-02-06 16:48 lib
-rw-r--r-- 1 exem exem 4933 2017-01-04 09:52 logback-dev.xml
-rw-r--r-- 1 exem exem 4179 2017-01-04 09:52 logback-prd.xml
-rw-r--r-- 1 exem exem 1258 2017-01-04 09:52 mybatis-config.xml
drwxr-xr-x 2 exem exem 4096 2017-02-06 16:48 pages
-rw-r--r-- 1 exem exem 1681 2017-01-04 09:52 quartz.properties
-rw-r--r-- 1 exem exem 4648 2017-02-03 10:26 web.xml
[root@cdhfem WEB-INF]#
```

### 14.1. config.properties 설정 하기

... 생략

```
#####
## DataSource Configuration
#####

# Flamingo가 사용하는 PostgreSQL 정보.
# PostgreSQL connection info for Flamingo
datasource.flamingo.driver-class-name=org.postgresql.Driver
datasource.flamingo.jdbcUrl=jdbc:postgresql://cdhfem.exem.oss:5432/flamingo ①
datasource.flamingo.username=flamingo ②
datasource.flamingo.password=flamingo ②
datasource.flamingo.min.pool=3
datasource.flamingo.max.pool=10

# HBASE + Phoenix 접속정보
datasource.phoenix.driver-class-name=org.apache.phoenix.jdbc.PhoenixDriver
datasource.phoenix.jdbcUrl=jdbc:phoenix:cdhfem.exem.oss:2181:/fem-hbase ③
datasource.phoenix.max.pool=10

datasource.system.driver-class-name=org.h2.Driver
datasource.system.jdbcUrl=jdbc:h2:tcp:cdhfem.exem.oss:9092/mem:flamingo ④
datasource.system.username=sa
datasource.system.password=sa
```

... 생략

- ① Flamingo WEB에서 사용하는 PostgreSQL의 JDBC URL 정보를 입력합니다
- ② 설치된 PostgreSQL의 username / password를 입력합니다
- ③ HBase + Phoenix가 설치된 JDBC URL 정보를 입력합니다.
- ④ Collector가 설치된 Hostname으로 변경합니다. Collector 내부에서 사용하는 In-memory Cache DB입니다 Collector가 구동될때 자동으로 DB가 활성화 됩니다. 만약 다른 DB로 사용한다면 해당 정보를 설치된 DB정보에 맞게 변경합니다

## 14.2. hadoop.properties 설정 하기

```
#####
## MapReduce Configuration
#####

# History Server
history.server.url=http://cdh1.exem.oss:19888 ①

#####
## Resource Manager Configuration
#####

# Yarn Application
resourcemanager.host=cdh1.exem.oss ②
resourcemanager.port=8088
#ha값이 true인 경우 반드시 zookeeper quorum 설정값이 필요하다
resourcemanager.ha=false
resourcemanager.agent.port=18032

#####
## Node Manager Configuration
#####
nodemanager.port=8042

#####
## Agent Configuration
#####

# Namenode Agent
#namenode.agent.ha값이 true인 경우 반드시 zookeeper quorum 설정값이 필요하다
namenode.ha=false ③
namenode.host=cdh1.exem.oss ④
namenode.znode=/hadoop-ha/namenode-ha/ActiveBreadCrumb ⑤
namenode.agent.port=10070

#####
## Hive Configuration
#####

# Hive Metastore Agent
hive.metastore.url=thrift://cdh2.exem.oss:9083 ⑥

# Hive Server 2 Agent
hive.server2.url=jdbc:hive2://cdh2.exem.oss:10000 ⑦
hive.server2.username=hive

# Hive 쿼리 실행시 Flamingo의 사용자를 적용할지 여부.
# 이 값이 false인 경우 hive.username 의 값을 적용한다.
hive.apply.flamingo.username=true

# Hive 쿼리 실행시 적용할 사용자
hive.username=yarn

# Hive 0.13.0 이전 버전을 사용할 경우 해당 값을 true로 변경한다.
hive.legacy=false

#####
## Oozie Configuration
#####

# Oozie Server 의 Url
oozie.server.url=http://cdh1.exem.oss:11000/oozie ⑧
oozie.query.url=http://cdh1.exem.oss:31000

#####
```

```
## Zookeeper Configuration
#####
zookeeper.quorum=cdh1.exem.oss ⑨
```

- ① MapReduce History Server의 URL, Port를 입력한다.
- ② 설치한 Resource Manager의 Agent Address, Port를 입력한다.
- ③ Namenode HA구성을 한 경우 true, Single Namenode를 사용하는경우 False로 설정한다.
- ④ namenode.agent.ha=false 일 경우 설치한 Namenode Agent의 Address, Port를 입력한다.
- ⑤ namenode.agent.ha=true 일 경우 Namenode HA의 zookeeper znode 값을 입력한다.
- ⑥ Hive Metastore가 설치된 Hostname Port를 입력한다.
- ⑦ Hive Server2가 설치된 Hostname Port를 입력한다.
- ⑧ Oozie Server의 URL을 입력한다.
- ⑨ 설치된 Zookeeper의 quorum 정보를 입력한다.



## Chapter 15. 서비스 구동 하기

### 15.1. Apache ActiveMQ

Apache ActiveMQ는 `/opt/exem/fem/current/mq` 에 있습니다. 아래와 같이 커맨드를 실행하여 구동합니다.

```
$ cd /opt/exem/fem/current/mq/bin/  
$ ./activemq start
```

NOTE | start, stop, restart, kill, status 등을 지원합니다

### 15.2. Flamingo Collector

Flamingo Collector는 `/opt/exem/fem/current/collector` 에 있지만 서비스 등록이 되어 있어 `service` 커맨드로 구동 가능합니다. 아래와 같이 커맨드를 실행하여 구동합니다.

```
$ service flamingo-collector start
```

NOTE | start, stop, restart, status 등을 지원합니다.

### 15.3. Flamingo Web

Flamingo Web은 `/opt/exem/fem/current/web` 에 있지만 서비스 등록이 되어 있어 `service` 커맨드로 구동 가능합니다. 아래와 같이 커맨드를 실행하여 구동합니다.

```
$ service flamingo-web start
```

NOTE | start, stop, restart, status 등을 지원합니다.

## Chapter 16. 릴리즈 노트

### 16.1. Release 3.0.0

- 모니터링 대시보드
- 실시간 정보 수집 및 표시 기능 강화
- HDFS Browser & Audit
- HDFS File System Audit
- Namenode 모니터링
- Datanode 모니터링
- Node Manager 모니터링
- Resource Manager 모니터링
- Apache Spark 모니터링
- Apache Oozie 모니터링
- 각 노드의 CPU Core, RAM, Network, Disk 실시간 모니터링



## Chapter 17. 제품 및 기술지원 정보

다음은 Flamingo와 관련된 사이트 정보입니다.

개발사	<a href="#">EXEM Inc.</a>
회사소개서	<a href="#">EXEM 회사소개서</a>
기술지원	<a href="#">Flamingo Technical Support Team</a>
제품 홈페이지	<a href="#">Flamingo Live</a>

