



Big Data Performance Management

MANUAL

Copyright © 2016 EXEM Inc. All rights reserved.

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. 설치전 준비사항	5
2.1. 고객사 확인 사항	5
2.2. TODO	5
3. Hortonworks HDP 배포판 설치 가이드	6
3.1. JDK 8 설치하기	6
3.2. 패키지 설치	7
3.3. MQ Sink 배포하기	7
3.4. Hadoop 모니터링 설정하기	8
3.5. Oozie 모니터링 설정하기	14
3.6. MapReduce 모니터링 설정하기	15
3.7. HDFS 모니터링 설정하기	18
4. Apache Hive 모니터링 설정하기	21
4.1. Hive JMX 설정 하기	21
4.2. Hive Agent	21
4.3. Hive CLI 모니터링	22
5. Oozie Event Listener 설정 하기	24
6. Resource Manager Agent 설정 하기	27
7. Namenode Agent 설정 하기	29
8. 시스템 리소스 모니터링 설정하기	30
8.1. System Agent의 MQ URL 변경하기	30
9. 서비스 구동 하기	31
9.1. Apache ActiveMQ	31
9.2. Hadoop 서비스 재시작 하기	31
9.3. System Agent 시작하기	32
9.4. Flamingo Web	33
9.5. Flamingo Collector	34
10. 설치 확인	35
10.1. MQ 확인	35
10.2. Resource Manage Agent 확인	36
10.3. Namenode Agent 확인	36
10.4. System Agent 확인	36
10.5. Oozie JPA Service Extension 확인	38
10.6. Collector 확인	39
10.7. Web 확인	39

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 프로젝트로써 초기에 오픈소스인 GPL v3 라이선스로 출발하였습니다.

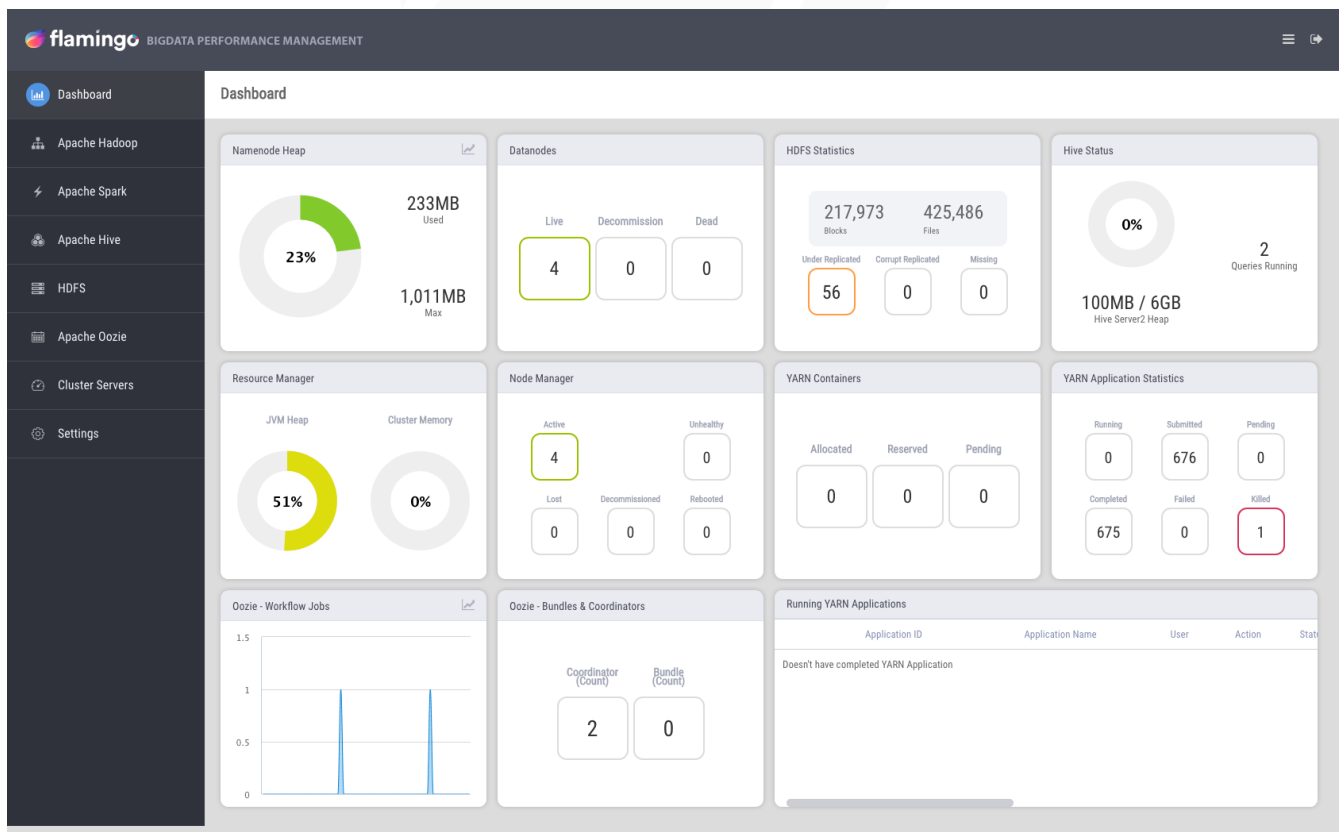
현재 Flamingo는 통합 브랜드로 발전하였고, 다음과 같이 총 3개의 프로젝트 및 제품으로 구성되어 있습니다. 총 3개의 프로젝트중 여전히 오픈소스로 개발되고 있는 프로젝트(Flamingo Oozie Workflow Designer, Flamingo Analytics)는 2개이며, 나머지 1개의 프로젝트(Flamingo Big Data Performance Management)는 On-Premise 및 Cloud SaaS 형태로만 제공하는 Commercial 제품입니다

제품 구성	라이선스	개요
Flamingo Oozie Workflow Designer	Apache 2	Apache Oozie 워크플로우 디자이너 프로젝트 : http://www.github.com/EXEM-OSS/flamingo
Flamingo Analytics	Apache 2 Subscription	빅데이터 기술을 이용한 데이터 분석 도구 프로젝트 : http://www.github.com/EXEM-OSS/flamingo-analytics
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.

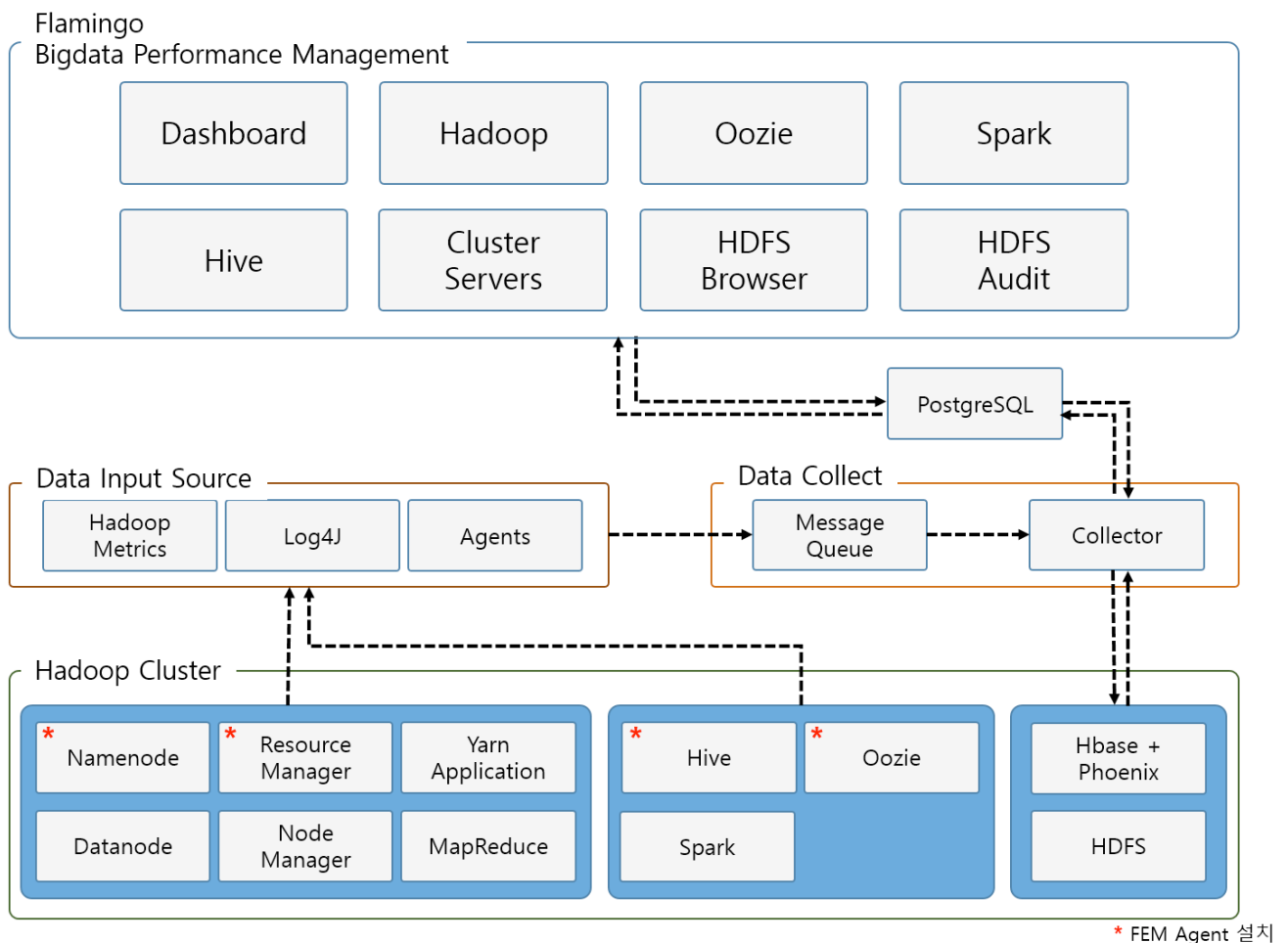


Flamingo Big Data Performance Management는 다음의 기능을 제공합니다.

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

구성요소	개요
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	/opt/exem/fem/3.0.1	flamingo	flamingo	규칙 : /opt/exem/fem/<VERSION>
링크 경로	/opt/exem/fem/current	flamingo	flamingo	/opt/exem/fem/3.0.1 경로의 링크
Agent 설치 경로	/opt/exem/fem/current/agents	flamingo	flamingo	
MQ Sink 설치 경로	/opt/exem/fem/current/agents	flamingo	flamingo	
Web 설치 경로	/opt/exem/fem/current/web	flamingo	flamingo	
Collector 설치 경로	/opt/exem/fem/current/collector	flamingo	flamingo	
MQ 설치 경로	/opt/exem/fem/current/mq	flamingo	flamingo	

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

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



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. 설치전 준비사항

2.1. 고객사 확인 사항

기술지원 또는 시스템 엔지니어가 Flamingo를 설치하기에 앞서 다음의 내용을 고객사에 문의하여 확인하도록 합니다.

구분	질문 내용	확인 여부
네트워크	고객사의 내부 시스템에 대한 방화벽 정책이 있는지?	
	방화벽 정책이 있는 경우 Flamingo에서 사용하는 Port 등에 대한 정보를 사전에 제공해야 하는지?	
	해당 포트에 대해서 개방하는데 소요되는 시간?	
	사용 금지 포트 번호가 별도로 있는지?	
	서버 또는 가상머신에서 외부로 네트워크 접속이 가능한지?	
시스템 구성	Flamingo를 위해서 별도의 서버 또는 가상머신을 제공할 수 있는지?	
	Flamingo를 위한 Postgres DB로 사용할 수 있는 서버를 제공하는지? 아니면 별도로 설치를 해야하는지?	
보안	설치에 필요한 계정 생성시 별도 절차가 필요한지?	
	계정 생성시 고객사 승인이 필요한 경우 승인에 소요되는 시간?	
설치	고객사 방문시 Flamingo를 포함한 각종 파일을 전달할 때 사용할 수 있는 미디어의 종류는?	
	Flamingo 관련 패키지 설치 후 설치 관련 각종 로그 정보를 설치 완료 가이드에 포함해야 하는지?	
	설치 완료 보고서를 별도로 제출해야 하는지?	
	설치 완료 보고서의 고객사 양식이 별도로 있는지?	

2.2. TODO

Chapter 3. Hortonworks HDP 배포판 설치 가이드

Hortonworks HDP 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	flamingo 계정으로 실행
6	Apache ActiveMQ 구동	flamingo 계정으로 실행
7	Flamingo Collector 구동	flamingo 계정으로 실행
8	Flamingo Web 구동	flamingo 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	설치 완료 점검	각종 포트의 정상동작 및 수집 데이터의 정상 송수신 확인

3.1. JDK 8 설치하기

3.1.1. 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)
```



Hadoop Cluster에 속해 있는 노드에 Flamingo를 설치하는 경우 보통 HDP가 설치되어 있으며 이 경우 JDK가 설치되어 있습니다. 구 버전의 HDP가 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.

3.1.2. CentOS

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

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

3.1.3. Ubuntu

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


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

3.1.4. 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
```

3.2. 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.

3.2.1. RPM 패키지 설치

RPM을 복사한 모든 서버에서 아래와 같이 커맨드를 실행하여 설치 합니다.

```
$ rpm -Uvh flamingo-management-*3.0.0.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
lrwxrwxrwx. 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]#
```

3.3. 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-hm-2-3.x.x.jar** 이며 **/opt/exem/fem/current/agents** 에 있습니다. 이 파일을 모든 서버의 다음 디렉토리에 복사하도록 합니다.

- **/usr/hdp/2.5.0.0-1245/hadoop-yarn/lib**
- **/usr/hdp/2.5.0.0-1245/hadoop-hdfs/lib**



모든 설치에 HDP 2.5 버전을 기준으로 하며 또한 HDP의 설치 버전에 따라서 설치 경로가 다를 수 있습니다.

3.4. 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. **/etc/hadoop/conf/hadoop-metrics2.properties** 파일을 직접 수정하는 경우 하둡 서비스 재시작 시 원상복귀가 되므로, HDP에서 제공하는 템플릿 파일을 수정하도록 합니다. HDP에서 제공하는 템플릿 파일의 위치는 다음과 같습니다.

- **/var/lib/ambari-agent/cache/stacks/HDP/2.0.6/hooks/before-START/templates/hadoop-metrics2.properties.j2**



만약에 Ambari Server에 Namenode, Resource Manager, Node Manager, Datanode 등이 같이 설치되어 있는 경우 **/var/lib/ambari-server/resources/stacks/HDP/2.0.6/hooks/before-START/templates/hadoop-metrics2.properties.j2** 템플릿 파일을 수정하도록 합니다.

다음은 **hadoop-metrics2.properties.j2** 파일의 내용입니다. 다음의 파일에서 MQ Sink 설정 정보를 추가하고자 할 때 Ambari 설치시 Ganglia와 함께 설치한 경우와 Metric Collector를 함께 설치한 경우 두 가지 경우에 따라서 설정 정보를 추가하는 위치가 상이합니다. 알맞는 위치에 아래 서비스별 설정 정보를 추가합니다

... 생략

```
{% if has_ganglia_server %}
*.period=60

*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31
*.sink.ganglia.period=10

# default for supportsparse is false
*.sink.ganglia.supportsparse=true

.sink.ganglia.slope=jvm.metrics.gcCount=zero,jvm.metrics.memHeapUsedM=both
.sink.ganglia.dmax=jvm.metrics.threadsBlocked=70,jvm.metrics.memHeapUsedM=40
```

... 생략

resourcemanager.sink.ganglia.tagsForPrefix.yarn=Queue

①

{% endif %}

```
{% if has_metric_collector %}

*.period={{metrics_collection_period}}
*.sink.timeline.plugin.urls=file:///usr/lib/ambari-metrics-hadoop-sink/ambari-metrics-hadoop-sink.jar
*.sink.timeline.class=org.apache.hadoop.metrics2.sink.timeline.HadoopTimelineMetricsSink
*.sink.timeline.period={{metrics_collection_period}}
*.sink.timeline.sendInterval={{metrics_report_interval}}000
*.sink.timeline.slave.host.name = {{hostname}}
```

... 생략

resourcemanager.sink.timeline.tagsForPrefix.yarn=Queue

②

```
{% if is_nn_client_port_configured %}
# Namenode rpc ports customization
namenode.sink.timeline.metric.rpc.client.port={{nn_rpc_client_port}}
{% endif %}
{% if is_nn_dn_port_configured %}
namenode.sink.timeline.metric.rpc.datanode.port={{nn_rpc_dn_port}}
{% endif %}
{% if is_nn_healthcheck_port_configured %}
namenode.sink.timeline.metric.rpc.healthcheck.port={{nn_rpc_healthcheck_port}}
{% endif %}

{% endif %}
```

① Ambari 설치시 Ganglia를 함께 설치한 경우 추가할 위치

② Ambari 설치시 Metric Collector를 함께 설치한 경우 추가할 위치

3.4.1. 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

Namendode 모니터링 설정하기

Namenode는 Namenode HA의 Active, Standby Namenode를 포함하여 다음과 같이 **hadoop-metrics2.properties.j2** 템플릿 파일에 추가하도록 합니다

```
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://fem.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://fem.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://fem.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)



Ambari 2.5 이상부터는 HDFS > Advenced > Advenced Hadoop-Metircs2 properties를 제공해서 그 안에 위 설정 값들을 적용해주면 됩니다.

Secondary Namendode 모니터링 설정하기

Namenode는 Namenode HA의 Active, Standby Namenode를 포함하여 다음과 같이 **hadoop-metrics2.properties.j2** 템플릿 파일에 추가하도록 합니다

```

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://fem.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://fem.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://fem.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)



Ambari 2.5 이상부터는 HDFS > Advanced > Advanced Hadoop-Metircs2 properties를 제공해서 그 안에 위 설정 값들을 적용해주면 됩니다.

Datanode 모니터링 설정하기

Hadoop Cluster의 모든 Datanode에 다음과 같이 **hadoop-metrics2.properties.j2** 템플릿 파일에 추가하도록 합니다

```
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://fem.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=admin ②
datanode.sink.jvm.broker_url=tcp://fem.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://fem.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)



Ambari 2.5 이상부터는 HDFS > Advanced > Advanced Hadoop-Metircs2 properties를 제공해서 그 안에 위 설정 값들을 적용해주면 됩니다.

YARN Node Manager 모니터링 설정하기

Hadoop Cluster의 모든 YARN Node Manager에 다음과 같이 **hadoop-metrics2.properties.j2** 템플릿 파일에 추가하도록 합니다

```
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://fem.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://fem.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://fem.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)



Ambari 2.5 이상부터는 HDFS > Advanced > Advanced Hadoop-Metircs2 properties를 제공해서 그 안에 위 설정값들을 적용해주면 됩니다.

YARN Resource Manager 모니터링 설정하기

Hadoop Cluster의 모든 YARN Resource Manager에 다음과 같이 **hadoop-metrics2.properties.j2** 템플릿 파일에 추가하도록 합니다

```

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://fem.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://fem.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://fem.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)



Ambari 2.5 이상부터는 HDFS > Advanced > Advanced Hadoop-Metircs2 properties를 제공해서 그 안에 위 설정 값들을 적용해주면 됩니다.

3.5. 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 HDP의 Ambari로 설정하는 방법과 **oozie-site.xml** 파일을 직접 수정하는 방법이 있으며 Ambari로 설정하는 것을 권장합니다.

3.5.1. 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 **/usr/hdp/current/oozie-client/oozie-server/webapps/oozie/WEB-INF/lib** 입니다.

3.5.2. Apache Ambari로 설정하기

oozie-site.xml 파일은 Oozie의 메인 설정 파일이며 이 파일의 내용을 수정하기 위해서는 HDP의 Ambari를 이용해야 합니다. Oozie 설정을 변경하기 위해서 다음과 같이 'Ambari > Oozie > Config' 메뉴를 선택합니다.

Config 메뉴를 선택하면 Key Value 형식으로 **oozie-site.xml** 파일의 내용이 출력되며 다음과 같이 'oozie.services.ext' 항목을 찾아서 제일 마지막에

com.exem.flamingo.management.agent.oz.jpa.JpaNativeQueryExecutorService,org.apache.oozie.service.MetricsInstrumentationService 항목을 추가합니다.



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

3.6. 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 HDP의 Ambari로 설정하는 방법과 YARN의 **log4j.properties** 파일을 직접 수정하는 방법이 있으며 Ambari로 설정하는 것을 권장합니다. 본 설치 가이드에서는 직접 수정하는 방법은 설명하지 않습니다.

3.6.1. Apache Ambari로 설정하기

Ambari로 로그인하여 'Ambari > YARN > Configs > Advanced' 메뉴를 선택합니다.

The screenshot shows the Ambari web interface. In the top navigation bar, the 'Configs' tab is highlighted. On the left sidebar, the 'YARN' service is selected. The main content area shows the 'Manage Config Groups' section with a list of configuration groups (V3 to V8). Below this, the 'Settings' section is expanded, and the 'Advanced' sub-tab is selected. Under the 'Resource Manager' section, the following configuration parameters are visible:

- ResourceManager host: hdp02.exem.oss
- ResourceManager Java heap size: 1024 MB
- yarn.acl.enable: false
- yarn.admin.acl: yarn
- Enable Log Aggregation: ☒

다음과 같이 'Advanced yarn-log4j' 메뉴를 찾아서 파일의 내용을 수정합니다.

Advanced yarn-log4j

```
log4j.appender.CSVMA.maxUniqueMessages=${yarn.rwma.maxUniqueMessages}

# 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.ProviderURL=tcp://fem.exem.oss:61613?daemon=true
log4j.appender.RMAUDITJMS.userName=admin
log4j.appender.RMAUDITJMS.password=password
log4j.appender.RMAUDITJMS.TopicBindingName=mapredAuditTopic
log4j.appender.RMAUDITJMS.TopicConnectionFactoryBindingName=ConnectionFactory
```

MQ를 통해 MapReduce 실행 이벤트를 수집하기 위해서 다음과 같이 수정합니다.

... 생략

```
# Audit logging for ResourceManager
rm.audit.logger=${hadoop.root.logger}
#log4j.logger.org.apache.hadoop.yarn.server.resourcemanager.RMAuditLogger=${rm.audit.logger} ①
log4j.logger.org.apache.hadoop.yarn.server.resourcemanager.RMAuditLogger=INFO, RMAUDIT, RMAUDITJMS ②
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 ③
log4j.appender.RMAUDITJMS.InitialContextFactoryName=org.apache.activemq.jndi.ActiveMQInitialContextFactory ③
log4j.appender.RMAUDITJMS.userName=admin ③
log4j.appender.RMAUDITJMS.password=admin ③
log4j.appender.RMAUDITJMS.ProviderURL=tcp://fem.exem.oss:61616?daemon=true ③
log4j.appender.RMAUDITJMS.TopicBindingName=mapredAuditTopic ③
log4j.appender.RMAUDITJMS.TopicConnectionFactoryBindingName=ConnectionFactory ③
```

... 생략

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

3.6.2. 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.

- **/usr/hdp/current/hadoop-yarn-client/jndi.properties**

새로 생성한 상기 파일에 다음을 내용을 추가하고 저장합니다.

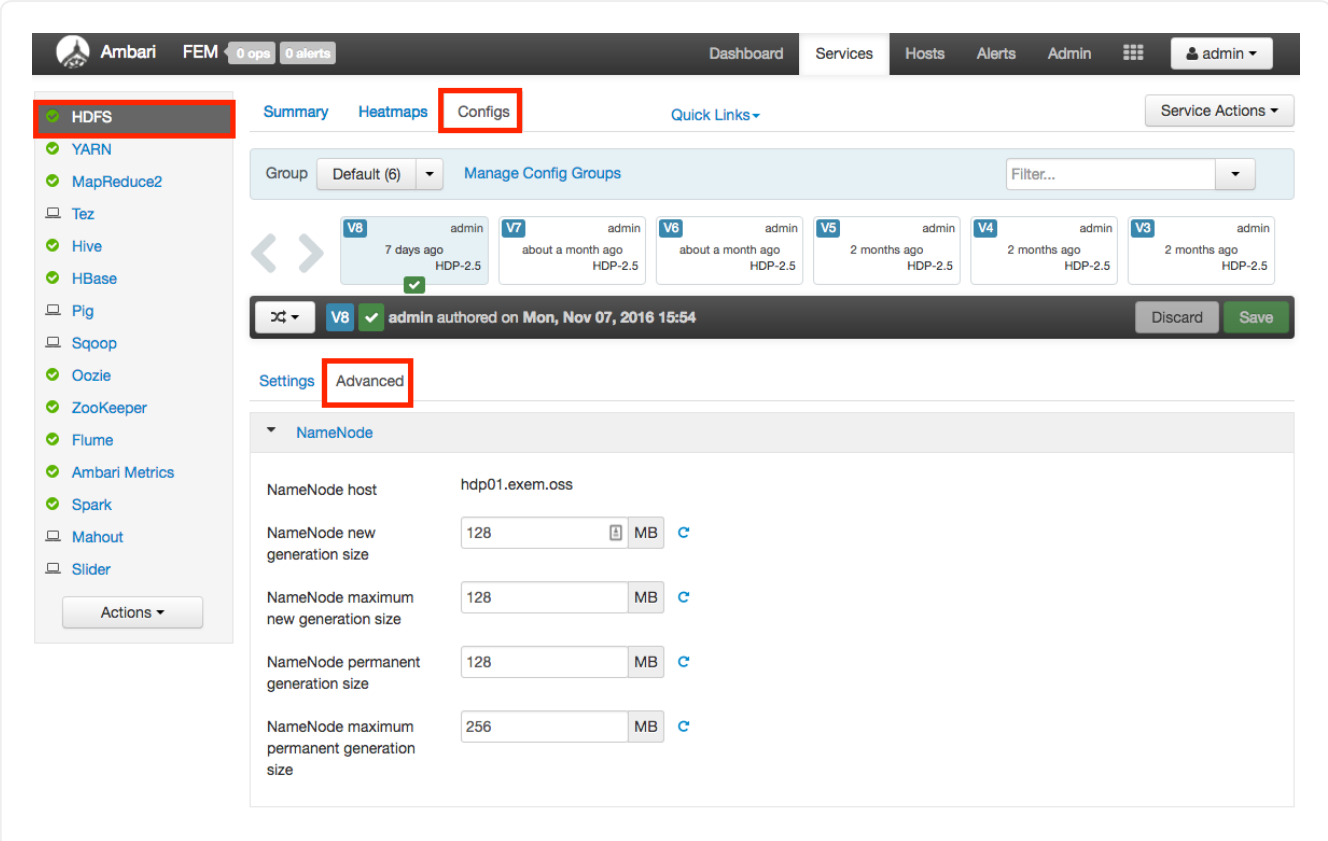
```
topic.mapredAuditTopic=mapredAuditTopic
topic.hdfsAuditTopic=hdfsAuditTopic
```

3.7. HDFS 모니터링 설정하기

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

3.7.1. Apache Ambari로 설정하기

Ambari로 로그인하여 'Ambari > HDFS > Configs > Advanced' 메뉴를 선택합니다.



The screenshot shows the Apache Ambari web interface. The top navigation bar includes 'Ambari', 'FEM', '0 ops', '0 alerts', and tabs for 'Dashboard', 'Services', 'Hosts', 'Alerts', and 'Admin'. The user is logged in as 'admin'. The left sidebar lists various services: HDFS (selected), YARN, MapReduce2, Tez, Hive, HBase, Pig, Sqoop, Oozie, ZooKeeper, Flume, Ambari Metrics, Spark, Mahout, and Slider. The main content area shows the 'HDFS' service configuration. The 'Configs' tab is selected, displaying a list of configuration groups. The 'Advanced' sub-tab is selected, showing the 'NameNode' configuration section. The configuration fields are as follows:

Configuration	Value	Unit	Action
NameNode host	hdp01.exem.oss		
NameNode new generation size	128	MB	C
NameNode maximum new generation size	128	MB	C
NameNode permanent generation size	128	MB	C
NameNode maximum permanent generation size	256	MB	C

다음과 같이 'Advanced hdfs-log4j' 메뉴를 찾아서 파일의 내용을 수정합니다.

Advanced hdfs-log4j

```
#
hdfs.audit.logger=INFO,console
#log4j.logger.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit=${hdfs.audit.logger}
log4j.logger.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit=INFO,DRFAAUDIT,DRFAAUDITJMS
log4j.additivity.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit=false
log4j.appender.DRFAAUDIT=org.apache.log4j.DailyRollingFileAppender
log4j.appender.DRFAAUDIT.File=${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
log4j.appender.DRFAAUDITJMS.InitialContextFactoryName=org.apache.activemq.jndi.ActiveMQInitialContextFactory
log4j.appender.DRFAAUDITJMS.ProviderURL=tcp://fem.exem.oss:61613?daemon=true
log4j.appender.DRFAAUDITJMS.userName=admin
log4j.appender.DRFAAUDITJMS.password=password
log4j.appender.DRFAAUDITJMS.TopicBindingName=hdfsAuditTopic
log4j.appender.DRFAAUDITJMS.TopicConnectionFactoryBindingName=ConnectionFactory
```

hdfs-log4j template
content

Custom log4j.properties

MQ를 통해 MapReduce 실행 이벤트를 수집하기 위해서 다음과 같이 수정합니다.

... 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:61613?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.



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

3.7.2. JNDI 설정하기

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

- `/usr/hdp/current/hadoop-hdfs-client/jndi.properties`

새로 생성한 상기 파일에 다음을 내용을 추가하고 저장합니다.

```
topic.mapredAuditTopic=mapredAuditTopic  
topic.hdfsAuditTopic=hdfsAuditTopic
```



Chapter 4. Apache Hive 모니터링 설정하기

4.1. Hive JMX 설정 하기

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

HDP 2.5 기준 `/usr/hdp/2.5.0.0-1245/hive/bin/ext/hiveserver2.sh` 파일을 열어 아래 번호표기 되어 추가 된 내용과 같이 추가합니다.

```
1 ... 생략
2
3 hiveserver2() {
4   echo "Starting Hive Server 2"
5   CLASS=org.apache.hive.service.server.HiveServer2
6   if $cygwin; then
7     HIVE_LIB=`cygpath -w "$HIVE_LIB"`
8   fi
9   JAR=${HIVE_LIB}/hive-service-*.jar
10
11   export HADOOP_OPTS="$HADOOP_OPTS -Dcom.sun.management.jmxremote" ①
12   export HADOOP_OPTS="$HADOOP_OPTS -Dcom.sun.management.jmxremote.authenticate=false" ①
13   export HADOOP_OPTS="$HADOOP_OPTS -Dcom.sun.management.jmxremote.ssl=false" ①
14   export HADOOP_OPTS="$HADOOP_OPTS -Dcom.sun.management.jmxremote.port=9004" ①
15   export HADOOP_OPTS="$HADOOP_OPTS -Djava.rmi.server.hostname=hdp05.exem.oss" ②
16   export HADOOP_OPTS="$HADOOP_OPTS -javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-hv-3.0.0-SNAPSHOT.jar" ③
17
18   echo "$HADOOP jar $JAR $CLASS $HIVE_OPTS" ④
19
20   exec $HADOOP jar $JAR $CLASS $HIVE_OPTS "$@"
21 }
22
23 hiveserver2_help() {
24   hiveserver2 -H
25 }
```

① 일부 버전에서는 이 부분이 생략되어 있을 수 있습니다. 없다면 다음과 같이 작성해줍니다.

② "~ hostname=" 부분은 반드시 Hive2가 설치되어 있는 hostname 으로 입력을 해주어야 합니다.

ex) hostname=hdp05.exem.oss

③ "~ javaagent:" 부분은 hive agent jar가 설치 되어 있는 경로로 입력해주어야 합니다

ex) javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-hv-3.0.0-SNAPSHOT.jar

④ 마지막으로 echo "\$HADOOP jar \$JAR \$CLASS \$HIVE_OPTS" 내용을 추가해줍니다.

4.2. Hive Agent

하나의 Agent로 Hive Server, Hive CLI를 동시에 모니터링 합니다.

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



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.

4.3. Hive CLI 모니터링

Hive CLI 모니터링 설정을 위해서 HDP 2.5 기준 다음과 같이 'Ambari > Hive > Configs > Advanced > Advanced hive-env > hive-env template' 메뉴를 선택합니다.

The screenshot shows the Ambari web interface. In the left sidebar, the 'Hive' service is selected. The main panel shows the 'Configs' tab for the 'Hive' service. Below the 'Configs' tab, there is a list of configuration groups (V1 to V6). The 'Advanced' settings tab is selected, showing the 'Hive Metastore' configuration. The 'Hive Database' option is set to 'Existing PostgreSQL Database'.

다음과 같이 'Advanced hive-env' 메뉴를 찾아서 hive-env-template을 아래와 같이 수정합니다.

Advanced hive-env

hive_ambari_database

MySQL

Hive Log Dir

/var/log/hive

Hive PID Dir

/var/run/hive

Use ATS Logging

true

hive_user_nofile_limit

32000

hive_user_nproc_limit

16000

hive-env template

The heap size of the jvm started by hive shell script can be controlled via:

if ["\$SERVICE" = "metastore"]; then

export HADOOP_HEAPSIZE={{hive_metastore_heapsize}} # Setting for HiveMetastore

else

export HADOOP_HEAPSIZE={{hive_heapsize}} # Setting for HiveServer2 and Client

fi

#export HADOOP_CLIENT_OPTS="\$HADOOP_CLIENT_OPTS -Xmx\${HADOOP_HEAPSIZE}m"

export HADOOP_CLIENT_OPTS="\$HADOOP_CLIENT_OPTS -javaagent:/usr/flamingo/flamingo-management-agent-hv-0.1-SNAPSHOT.jar"

Larger heap size may be required when running queries over large number of files or partitions.

By default hive shell scripts use a heap size of 256 (MB). Larger heap size would also be

appropriate for hive server (hwi etc).

Set HADOOP_HOME to point to a specific hadoop install directory

HADOOP_HOME=\${HADOOP_HOME:-{hadoop_home}}

```

1 ... 생략
2 if [ "$SERVICE" = "metastore" ]; then
3 export HADOOP_HEAPSIZE={{hive_metastore_heapsize}} # Setting for HiveMetastore
4 else
5 export HADOOP_HEAPSIZE={{hive_heapsize}} # Setting for HiveServer2 and Client
6
7 #export HADOOP_CLIENT_OPTS="$HADOOP_CLIENT_OPTS -Xmx${HADOOP_HEAPSIZE}m" ①
8 export HADOOP_CLIENT_OPTS="$HADOOP_CLIENT_OPTS -javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-hv-3.0.0-SNAPSHOT.jar" ②
9 # Larger heap size may be required when running queries over large number of files or partitions.
10 # By default hive shell scripts use a heap size of 256 (MB). Larger heap size would also be
11 # appropriate for hive server (hwi etc)
12 # Set HADOOP_HOME to point to a specific hadoop install directory
13 HADOOP_HOME=${HADOOP_HOME:-{hadoop_home}}
14 ... 생략

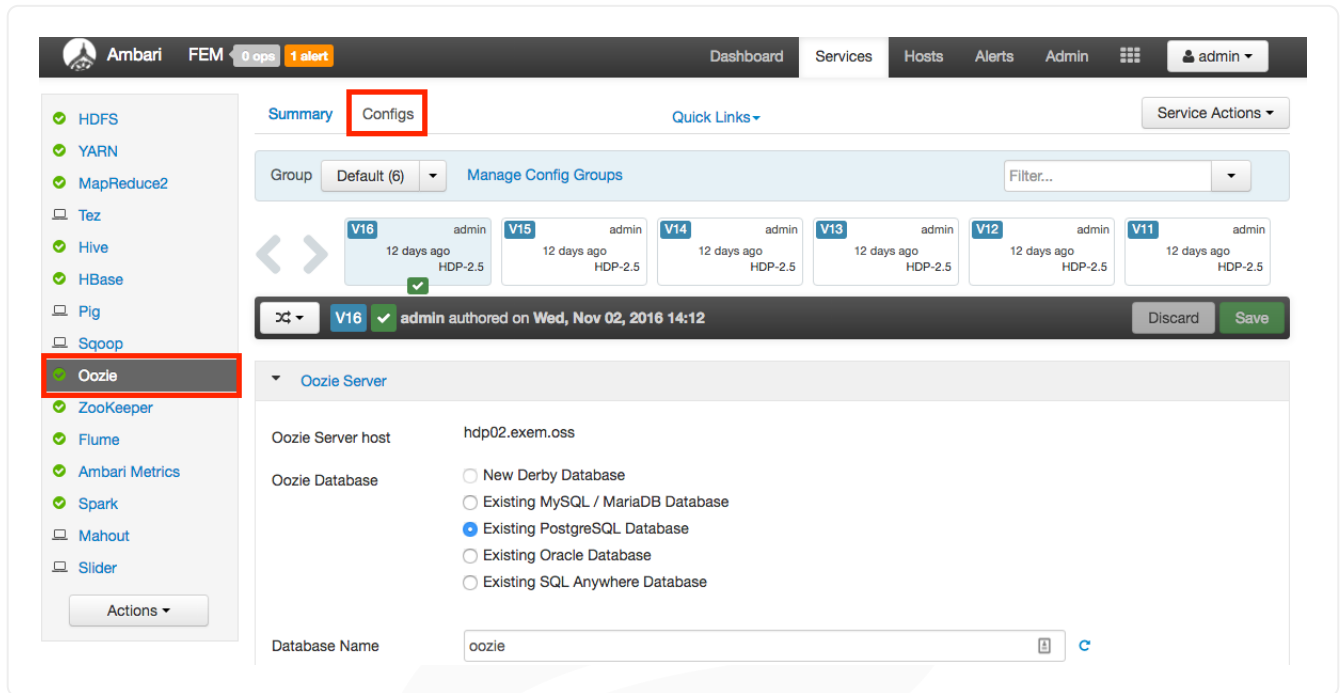
```

① 기존 설정되어 있는 값을 위와 같이 주석(#) 처리 해줍니다.

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

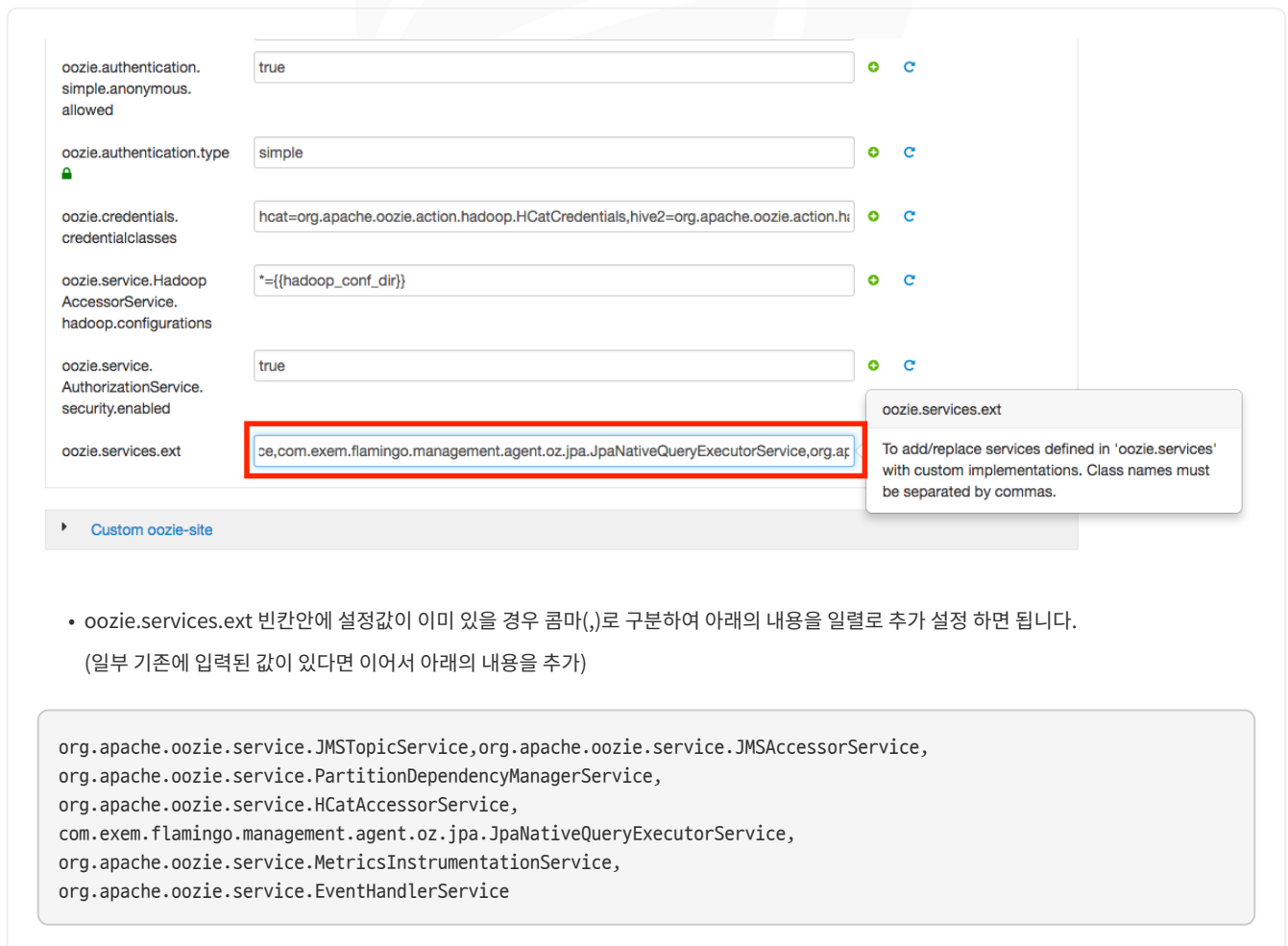
Chapter 5. Oozie Event Listener 설정 하기

Oozie Event Listener 설정을 위해 HDP 2.5는 Ambari > Oozie > Configs 로 이동합니다.



The screenshot shows the Ambari interface for configuring Oozie. The left sidebar lists various services, with 'Oozie' highlighted. The main area shows the 'Configs' tab for the Oozie Server. It displays a list of configuration groups (V11 to V16) and a detailed view for the selected group (V16). The 'Oozie Server' configuration details include the host (hdp02.exem.oss), database type (Existing PostgreSQL Database), and database name (oozie).

Config 화면 하단의 oozie.services.ext 메뉴를 통해 수정 합니다.



The screenshot shows the 'Oozie services' configuration page. It lists various configuration properties and their values. The 'oozie.services.ext' property is highlighted, showing a list of class names separated by commas. A tooltip explains that this property is used to add or replace services defined in 'oozie.services' with custom implementations. Below the configuration list, there is a 'Custom oozie-site' section.

oozie.services.ext

To add/replace services defined in 'oozie.services' with custom implementations. Class names must be separated by commas.

Custom oozie-site

- oozie.services.ext 빈칸안에 설정값이 이미 있을 경우 콤마(,)로 구분하여 아래의 내용을 일렬로 추가 설정 하면 됩니다.
(일부 기존에 입력된 값이 있다면 이어서 아래의 내용을 추가)

```
org.apache.oozie.service.JMSTopicService,org.apache.oozie.service.JMSAccessorService,  
org.apache.oozie.service.PartitionDependencyManagerService,  
org.apache.oozie.service.HCatAccessorService,  
com.exem.flamingo.management.agent.oz.jpa.JpaNativeQueryExecutorService,  
org.apache.oozie.service.MetricsInstrumentationService,  
org.apache.oozie.service.EventHandlerService
```

Config 화면 하단의 Custom oozie-site 메뉴를 통해 추가 설정을 합니다.

Custom oozie-site

Add Property ...

- 이동 시 현재 아무 추가 설정이 안된 있음을 확인 할 수 있습니다.

Add Property 를 눌러 설정 추가 창을 아래 화면과 같이 띄웁니다.

Add Property

X

Type

oozie-site.xml

Key

Value

Cancel

Add

Add Property 화면에서 해당 버튼을 누르면 Properties 입력 타입이 아래 화면과 같이 변경 됩니다.

Add Property

X

Type

oozie-site.xml

Properties

key=value (one per line)

Enter key=value (one per line)

Cancel

Add

App Property 의 Properties 값을 아래 와 같이 Key=value 형식으로 입력합니다.

Add Property

X

Type

oozie-site.xml



Properties

key=value (one per line)

```
oozie.service.EventHandlerService.event.listeners=Value : org.apache.oozie.jms.JMSJobEventListener
oozie.service.EventHandlerService.filter.app.types=workflow_job
oozie.jms.producer.connection.properties=java.naming.factory.initial#org.apache.activemq.jndi.ActiveMQInitialContextFactory;java.naming.provider.url#tcp://fem.exem.oss:61616;connectionFactoryNames#ConnectionFactory
oozie.service.JMSTopicService.topic.name=WORKFLOW=oozieWorkflow
```

Cancel

Add

```
oozie.service.EventHandlerService.event.listeners=org.apache.oozie.jms.JMSJobEventListener
```

```
oozie.service.EventHandlerService.filter.app.types=workflow_job
```

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

```
oozie.service.JMSTopicService.topic.name=WORKFLOW=oozieWorkflow
```

아래 화면과 같이 4가지로 Key=Value 입력이 잘 되었는지 확인합니다.

Custom oozie-site

oozie.service.
EventHandlerService.
event.listeners

org.apache.oozie.jms.JMSJobEventListener



oozie.service.
JMSTopicService.topic.
name

WORKFLOW=oozieWorkflow



oozie.service.
EventHandlerService.
filter.app.types

workflow_job



oozie.jms.producer.
connection.properties

java.naming.factory.initial#org.apache.activemq.jndi.ActiveMQInitialContextFactory;java.naming.provider.url#tcp://fem.exem.oss:61616;connectionFactoryNames#ConnectionFactory



[Add Property ...](#)

Chapter 6. Resource Manager Agent 설정 하기

Resource Manager Agent 설정을 위해 HDP2.5는 Ambari > YARN > Configs > Advanced 로 이동합니다.

The screenshot shows the Ambari web interface. On the left sidebar, 'YARN' is selected. The main content area has tabs for 'Summary', 'Heatmaps', and 'Configs', with 'Configs' being the active tab. Below the tabs, there's a 'Manage Config Groups' section showing a list of config groups (V3 to V8) with their creation dates and users. A confirmation bar at the bottom of this section indicates 'admin authored on Fri, Oct 07, 2016 14:21'. Below this, the 'Settings' section has a sub-tab 'Advanced' selected. Under 'Advanced', the 'Resource Manager' section is expanded, showing the following configuration fields:

- ResourceManager host: hdp02.exem.oss
- ResourceManager Java heap size: 1024 MB
- yarn.acl.enable: false
- yarn.admin.acl: yarn
- Enable Log Aggregation: checked

Config 화면 하단의 Advanced-yarn-env 메뉴를 통해 yarn-env-template을 아래와 같이 수정 합니다.

Advanced yarn-env

yarn_user_nproc_limit65536

yarn_user_nofile_limit32768

YARN PID Dir Prefix/var/run/hadoop-yarn

YARN Log Dir Prefix/var/log/hadoop-yarn

service_check.queue.namedefault

Minimum user ID for submitting job500

is_supported_yarn_rangertrue

yarn-env template

```

# This value will be overridden by an Xmx setting specified in either YARN_OPTS
# and/or YARN_RESOURCEMANAGER_OPTS.
# If not specified, the default value will be picked from either YARN_HEAPMAX
# or JAVA_HEAP_MAX with YARN_HEAPMAX as the preferred option of the two.
export YARN_RESOURCEMANAGER_HEAPSIZE={{(resourcemanager_heapsize)}}

# Specify the JVM options to be used when starting the ResourceManager.
# These options will be appended to the options specified as YARN_OPTS
# and therefore may override any similar flags set in YARN_OPTS

#export YARN_RESOURCEMANAGER_OPTS=-Xmx${HADOOP_HEAPSIZE}m"
export YARN_RESOURCEMANAGER_OPTS="-javaagent:/usr/local/flamingo/flamingo-
management-agent-rm-0.1-SNAPSHOT.jar"

# Node Manager specific parameters

# Specify the max Heapsize for the NodeManager using a numerical value
# in the scale of MB. For example, to specify an jvm option of -Xmx1000m, set
# the value to 1000.
# This value will be overridden by an Xmx setting specified in either YARN_OPTS
# and/or YARN_NODEMANAGER_OPTS.

```

```

1 ... 생략
2 # Specify the JVM options to be used when starting the ResourceManager.
3 # These options will be appended to the options specified as YARN_OPTS
4 # and therefore may override any similar flags set in YARN_OPTS
5
6 #export YARN_RESOURCEMANAGER_OPTS=-Xmx${HADOOP_HEAPSIZE}m" ①
7 export YARN_RESOURCEMANAGER_OPTS="-javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-rm-3.0.0-
SNAPSHOT.jar" ②
8
9 # Node Manager specific parameters
10
11 # Specify the max Heapsize for the NodeManager using a numerical value
12 # in the scale of MB. For example, to specify an jvm option of -Xmx1000m, set
13 # the value to 1000.
14 # This value will be overridden by an Xmx setting specified in either YARN_OPTS
15 # and/or YARN_NODEMANAGER_OPTS.
16 # If not specified, the default value will be picked from either YARN_HEAPMAX
17 # or JAVA_HEAP_MAX with YARN_HEAPMAX as the preferred option of the two.
18 export YARN_NODEMANAGER_HEAPSIZE={{(nodemanager_heapsize)}}
19 ... 생략

```

① 기존 설정되어 있는 값을 위와 같이 주석(#) 처리 해줍니다.

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

Chapter 7. Namenode Agent 설정 하기

hdfs가 실행 되는 쉘스크립트를 수정하기 위해 HDP2.5 기준 /usr/hdp/2.5.0.0-1245/hadoop-hdfs/bin/ 경로로 이동합니다.

```
[root@hdp01 bin]# pwd
/usr/hdp/2.5.0.0-1245/hadoop-hdfs/bin
[root@hdp01 bin]#
[root@hdp01 bin]# ls -la
합계 36
drwxr-xr-x 2 root root 4096 2016-12-20 10:52 .
drwxr-xr-x 7 root root 4096 2016-09-28 11:23 ..
-rwxr-xr-x 1 root root 468 2016-08-26 10:19 hdfs
-rwxr-xr-x 1 root root 12837 2016-12-20 10:52 hdfs.distro
-rwxr-xr-x 1 root root 830 2016-08-26 10:09 kill-name-node
-rwxr-xr-x 1 root root 888 2016-08-26 10:09 kill-secondary-name-node
-rw-r--r-- 1 root root 0 2016-12-07 09:25 rm-audit.log
[root@hdp01 bin]#
```

해당 경로로 이동하여 hdfs.distro 파일을 아래와 같이 수정 합니다.

```
1 ... 생략
2
3 if [ "$COMMAND" = "namenode" ] ; then
4     CLASS='org.apache.hadoop.hdfs.server.namenode.NameNode'
5     # HADOOP_OPTS="$HADOOP_OPTS $HADOOP_NAMENODE_OPTS" ①
6     HADOOP_OPTS="$HADOOP_OPTS -javaagent:/opt/exem/fem/current/agents/flamingo-management-agent-nn-3.0.0
-SNAPSHOT.jar $HADOOP_NAMENODE_OPTS" ②
7 elif [ "$COMMAND" = "zkfc" ] ; then
8     CLASS='org.apache.hadoop.hdfs.tools.DFSZKFailoverController'
9     HADOOP_OPTS="$HADOOP_OPTS $HADOOP_ZKFC_OPTS"
10 elif [ "$COMMAND" = "secondarynamenode" ] ; then
11
12 ... 생략
```

① 기존 설정되어 있는 값을 위와 같이 주석(#) 처리 해줍니다.

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

Chapter 8. 시스템 리소스 모니터링 설정하기

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.

8.1. System Agent의 MQ URL 변경하기

System Agent는 수집한 정보를 MQ로 송신합니다.

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



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 9. 서비스 구동 하기

9.1. Apache ActiveMQ

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

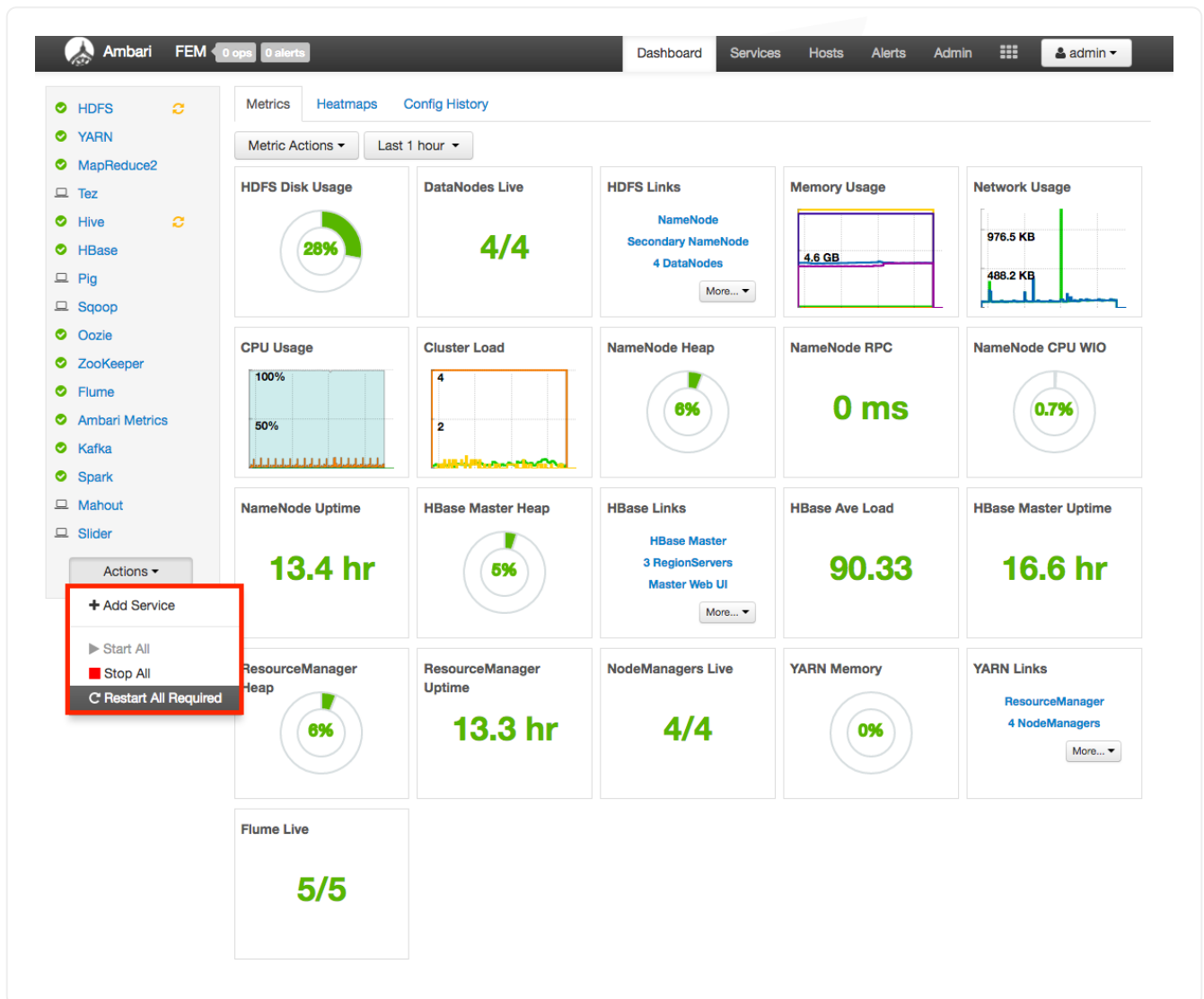
```
$ cd /opt/exem/fem/current/mq/bin/  
$ ./activemq start
```



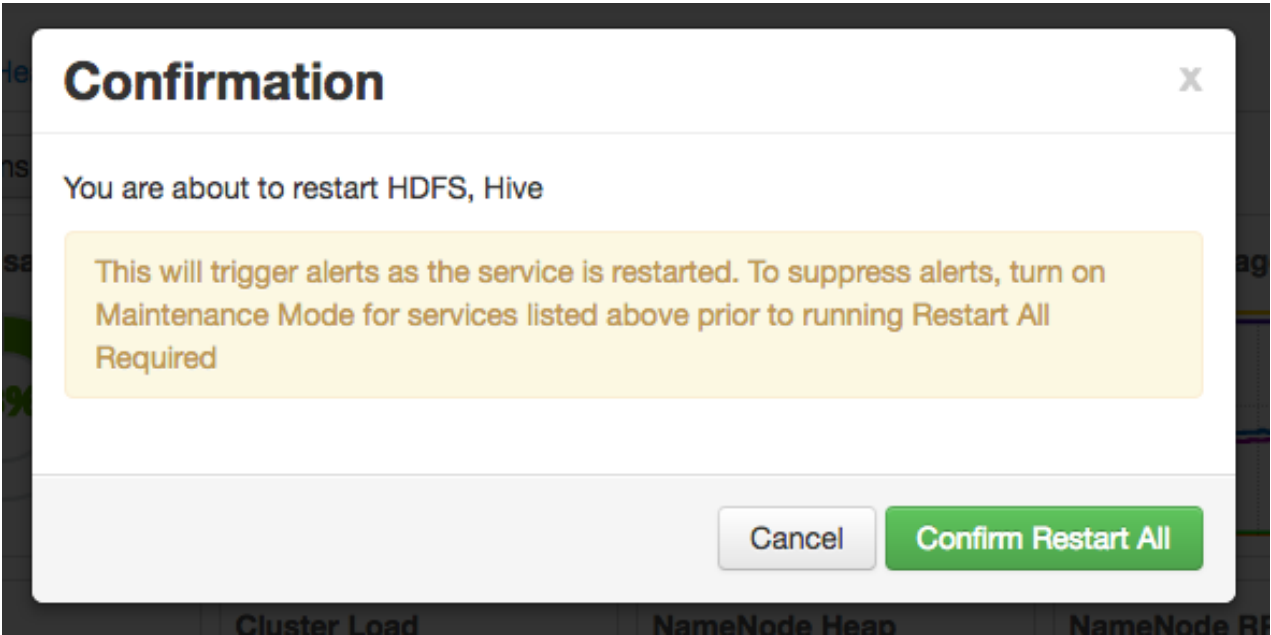
start, stop, restart, kill, status 등을 지원합니다.

9.2. Hadoop 서비스 재시작 하기

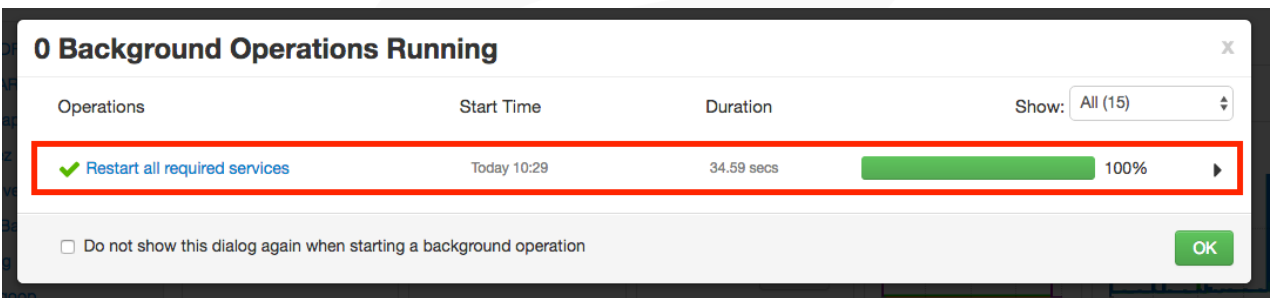
수정된 사항 적용을 위하여 Ambari에서 서비스들을 재시작 합니다.



수정된 서비스들을 확인 하고 Confirm 합니다.



설정을 정확히 했다면 문제 없이 서비스들이 재시작 됩니다.



9.3. System Agent 시작하기

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

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

9.3.1. 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
start	System Agent를 시작합니다.	
stop	System Agent를 중지합니다.	
restart	System Agent를 재시작합니다.	
force-reload	System Agent를 시작합니다.	
status	System Agent의 상태를 확인합니다.	

옵션	설명	비고
run	System Agent를 foreground로 실행합니다.	

9.4. 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
```



start, stop, restart, status 등을 지원합니다.

9.4.1. Flamingo 환경 설정 하기

Flamingo Web은 처음 구동시 GUI 환경 설정을 제공 합니다.

각 DB, Namenode, MQ, ResourceManager, Oozie, Hive 등의 정보를 설정 하고 테스트 완료 하고 저장을 하면 자동으로 재기동 됩니다.

Flamingo Web을 구동 후 브라우저를 통해 아래의 URL로 접속합니다.

```
http://web호스트명:WAS포트/
```

아래와 같은 설정 화면을 통해 각종 설정을 할 수 있습니다.

Flamingo

ActiveMQ

Namenode

ResourceManager

Oozie

Hive

ETC

Data Source

Phoenix JDBC:

jdbc:phoenix:localhost1,localhost2,localhost3:2181:/hbase-unsecure

● 성공

Test

Flamingo JDBC:

jdbc:postgresql://localhost:5432/flamingo

Username:

flamingo

Password:

flamingo

● 성공

Test

Collector

Host:

localhost

Port:

38081

Save



Test 버튼을 통해 설정이 올바른지 테스트 할 수 있으며 정상적인 정보를 입력했을 경우 초록색 아이콘으로 성공 여부를 표시해줍니다.

테스트를 완료하고 Save를 클릭하여 저장을 하면 서비스가 재기동 됩니다.

9.5. 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
```



start, stop, restart, status 등을 지원합니다.

Chapter 10. 설치 확인

10.1. MQ 확인

Agent와 Collector, MQ가 제대로 설치가 되었다면 <http://호스트명:8161/admin> 으로 접속해서 Queues와 Topics를 확인해보면 아래와 같이 나옵니다.

9개의 Queue와 3개의 Topic이 있고 Number Of Consumers가 1이어야 정상입니다.



일부 사용하지 않는 서비스가 있을 경우 9개가 아닐 수 있습니다.

Name ↑	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
collector.hdfs.dataNode	0	1	27816	27816	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
collector.hdfs.nameNode	0	1	9272	9272	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
collector.hdfs.secondaryNameNode	0	1	2304	2304	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
collector.hive.job	0	1	6	6	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
collector.hive.session	0	1	2466	2466	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
collector.system	0	1	52313	52313	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
collector.system.disk	0	1	2620	2620	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
collector.yarn.nodeManager	0	1	27543	27543	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
collector.yarn.resourceManager	0	1	6908	6908	Browse Active Consumers Active Producers atom rss	Send To Purge Delete

collector.hdfs.dataNode	Hadoop Metrics 2
collector.hdfs.nameNode	Hadoop Metrics 2
collector.hdfs.secondaryNameNode	Hadoop Metrics 2
collector.hive.job	Hive Agent
collector.hive.session	Hive Agent
collector.system	System Agent
collector.system.disk	System Agent
collector.yarn.nodeManager	Hadoop Metrics 2
collector.yarn.resourceManager	Hadoop Metrics 2



Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send

Topic Name Create

Topics

Name ↓	Number Of Consumers	Messages Enqueued	Messages Dequeued	Operations
oozieWorkflow	1	96	96	Send To Active Subscribers Active Producers Delete
mapredAuditTopic	1	8092	8092	Send To Active Subscribers Active Producers Delete
hdfsAuditTopic	1	51177	51177	Send To Active Subscribers Active Producers Delete

hdfsAuditTopic	HDFS Audit
mapredAuditTopic	MapReduce Audit
oozieWorkflow	Oozie Event Listener

10.2. Resource Manage Agent 확인

http://resourcemanager호스트명:18032

위의 URL로 접속 시 텍스트 메시지를 확인 할 수 있는데 이 메시지가 나온다면 에이전트가 설치된 것입니다.

10.3. Namenode Agent 확인

http://namenode호스트명:10070

위의 URL로 접속 시 텍스트 메시지를 확인 할 수 있는데 이 메시지가 나온다면 에이전트가 설치된 것입니다.

10.4. System Agent 확인

http://각서버호스트명:10179 또는 http://각서버호스트명:10179/test

위의 URL로 접속시 전자의 경우 텍스트 메시지, 후자의 경우 JSON타입으로 해당 서버의 프로세스 목록을 리턴 받을 수 있다.

```
[
  {
    "pid": "21106",
    "name": "(java)",
    "pgrpId": "21069",
    "ppid": "1",
    "sessionId": "21015",
    "vmem": "7245225984",
    "rssmemPage": "75022",
    "utime": "6119",
    "stime": "3073",
    "cpuusage": "7.554672",
    "cmd": "/usr/bin/java -Dsun.misc.URLClassPath.disableJarChecking=true -jar /root/flamingo-management-agent-system-3.0.0.jar "
  },
]
```

```

{
  "pid": "5909",
  "name": "(java)",
  "pgrpId": 5856,
  "ppid": "5895",
  "sessionId": 5856,
  "vmem": 4198150144,
  "rssmemPage": 477509,
  "utime": 65940,
  "stime": 20009,
  "cpuusage": 1.9880716,
  "cmd": "/usr/java/jdk1.8.0_102/bin/java -Dproc_regionserver -XX:OnOutOfMemoryError=kill -9 %p
-Dhdp.version=2.5.0.0-1245 -XX:+UseConcMarkSweepGC -XX:ErrorFile=/var/log/hbase/hs_err_pid%p.log -Djava.io.tmpdir=/tmp
-verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps -Xloggc:/var/log/hbase/gc.log-201611141036 -Xmn408m
-XX:CMSInitiatingOccupancyFraction=70 -Xms2048m -Xmx2048m -Dhbase.log.dir=/var/log/hbase -Dhbase.log.file=hbase-hbase
-regionserver-hdp06.exem.oss.log -Dhbase.home.dir=/usr/hdp/current/hbase-regionserver/bin/.. -Dhbase.id.str=hbase
-Dhbase.root.logger=INFO,RFA -Djava.library.path=/usr/hdp/2.5.0.0-1245/hadoop/lib/native/Linux-amd64
-64:/usr/hdp/2.5.0.0-1245/hadoop/lib/native -Dhbase.security.logger=INFO,RFA
org.apache.hadoop.hbase.regionserver.HRegionServer start "
},
{
  "pid": "5538",
  "name": "(java)",
  "pgrpId": 5483,
  "ppid": "1",
  "sessionId": 5483,
  "vmem": 2995466240,
  "rssmemPage": 95391,
  "utime": 17699,
  "stime": 22523,
  "cpuusage": 0.7952286,
  "cmd": "/usr/java/jdk1.8.0_102/bin/java -Dproc_datanode -Xmx1024m -Dhdp.version=2.5.0.0-1245
-Djava.net.preferIPv4Stack=true -Dhdp.version= -Djava.net.preferIPv4Stack=true -Dhdp.version=
-Djava.net.preferIPv4Stack=true -Dhadoop.log.dir=/var/log/hadoop/hdfs -Dhadoop.log.file=hadoop.log
-Dhadoop.home.dir=/usr/hdp/2.5.0.0-1245/hadoop -Dhadoop.id.str=hdfs -Dhadoop.root.logger=INFO,console
-Djava.library.path=/usr/hdp/2.5.0.0-1245/hadoop/lib/native/Linux-amd64-64:/usr/hdp/2.5.0.0-1245/hadoop/lib/native
-Dhadoop.policy.file=hadoop-policy.xml -Djava.net.preferIPv4Stack=true -Dhdp.version=2.5.0.0-1245
-Dhadoop.log.dir=/var/log/hadoop/hdfs -Dhadoop.log.file=hadoop-hdfs-datanode-hdp06.exem.oss.log
-Dhadoop.home.dir=/usr/hdp/2.5.0.0-1245/hadoop -Dhadoop.id.str=hdfs -Dhadoop.root.logger=INFO,RFA
-Djava.library.path=/usr/hdp/2.5.0.0-1245/hadoop/lib/native/Linux-amd64-64:/usr/hdp/2.5.0.0
-1245/hadoop/lib/native:/usr/hdp/2.5.0.0-1245/hadoop/lib/native/Linux-amd64-64:/usr/hdp/2.5.0.0-1245/hadoop/lib/native
-Dhadoop.policy.file=hadoop-policy.xml -Djava.net.preferIPv4Stack=true -server -server -XX:ParallelGCThreads=4
-XX:+UseConcMarkSweepGC -XX:ErrorFile=/var/log/hadoop/hdfs/hs_err_pid%p.log -XX:NewSize=200m -XX:MaxNewSize=200m
-Xloggc:/var/log/hadoop/hdfs/gc.log-201611141033 -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps
-XX:+PrintGCDateStamps -Xms1024m -Xmx1024m -Dhadoop.security.logger=INFO,DRFAS -Dhdfs.audit.logger=INFO,DRFAAUDIT
-server -XX:ParallelGCThreads=4 -XX:+UseConcMarkSweepGC -XX:ErrorFile=/var/log/hadoop/hdfs/hs_err_pid%p.log
-XX:NewSize=200m -XX:MaxNewSize=200m -Xloggc:/var/log/hadoop/hdfs/gc.log-201611141033 -verbose:gc -XX:+PrintGCDetails
-XX:+PrintGCDateStamps -XX:+PrintGCDateStamps -Xms1024m -Xmx1024m -Dhadoop.security.logger=INFO,DRFAS
-Dhdfs.audit.logger=INFO,DRFAAUDIT -server -XX:ParallelGCThreads=4 -XX:+UseConcMarkSweepGC
-XX:ErrorFile=/var/log/hadoop/hdfs/hs_err_pid%p.log -XX:NewSize=200m -XX:MaxNewSize=200m
-Xloggc:/var/log/hadoop/hdfs/gc.log-201611141033 -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps
-XX:+PrintGCDateStamps -Xms1024m -Xmx1024m -Dhadoop.security.logger=INFO,DRFAS -Dhdfs.audit.logger=INFO,DRFAAUDIT
-Dhadoop.security.logger=INFO,RFA org.apache.hadoop.hdfs.server.datanode.DataNode "
},
{
  "pid": "19992",
  "name": "(java)",
  "pgrpId": 19941,
  "ppid": "1",
  "sessionId": 19941,
  "vmem": 3085410304,
  "rssmemPage": 138250,
  "utime": 12021,
  "stime": 1941,
  "cpuusage": 0.3976143,
  "cmd": "/usr/java/jdk1.8.0_102/bin/java -Dproc_nodemanager -Xmx1024m -Dhdp.version=2.5.0.0-1245
-Dhadoop.log.dir=/var/log/hadoop-yarn/yarn -Dyarn.log.dir=/var/log/hadoop-yarn/yarn -Dhadoop.log.file=yarn-yarn

```

```

-nodemanager-hdp06.exem.oss.log -Dyarn.log.file=yarn-yarn-nodemanager-hdp06.exem.oss.log -Dyarn.home.dir=
-Dyarn.id.str=yarn -Dhadoop.root.logger=INFO,EWMA,RFA -Dyarn.root.logger=INFO,EWMA,RFA
-Djava.library.path=/usr/hdp/2.5.0.0-1245/hadoop/lib/native/Linux-amd64-64:/usr/hdp/2.5.0.0-1245/hadoop/lib/native:/var/lib/ambari-agent/tmp/hadoop_java_io_tmpdir:/usr/hdp/2.5.0.0-1245/hadoop/lib/native/Linux-amd64-64:/usr/hdp/2.5.0.0-1245/hadoop/lib/native:/var/lib/ambari-agent/tmp/hadoop_java_io_tmpdir
-Dyarn.policy.file=hadoop-policy.xml -Djava.io.tmpdir=/var/lib/ambari-agent/tmp/hadoop_java_io_tmpdir -server
-Dnm.audit.logger=INFO,NMAUDIT -Dnm.audit.logger=INFO,NMAUDIT -Dhadoop.log.dir=/var/log/hadoop-yarn/yarn
-Dyarn.log.dir=/var/log/hadoop-yarn/yarn -Dhadoop.log.file=yarn-yarn-nodemanager-hdp06.exem.oss.log -Dyarn.log.file=yarn
-yarn-nodemanager-hdp06.exem.oss.log -Dyarn.home.dir=/usr/hdp/current/hadoop-yarn-nodemanager
-Dhadoop.home.dir=/usr/hdp/2.5.0.0-1245/hadoop -Dhadoop.root.logger=INFO,EWMA,RFA -Dyarn.root.logger=INFO,EWMA,RFA
-Djava.library.path=/usr/hdp/2.5.0.0-1245/hadoop/lib/native/Linux-amd64-64:/usr/hdp/2.5.0.0-1245/hadoop/lib/native:/var/lib/ambari-agent/tmp/hadoop_java_io_tmpdir:/usr/hdp/2.5.0.0-1245/hadoop/lib/native/Linux-amd64-64:/usr/hdp/2.5.0.0-1245/hadoop/lib/native:/var/lib/ambari-agent/tmp/hadoop_java_io_tmpdir -classpath
/usr/hdp/current/hadoop-client/conf:/usr/hdp/current/hadoop-client/conf:/usr/hdp/current/hadoop-client/conf:/usr/hdp/2.5.0.0-1245/hadoop/lib/*:/usr/hdp/2.5.0.0-1245/hadoop/./*/usr/hdp/2.5.0.0-1245/hadoop-hdfs/./:/usr/hdp/2.5.0.0-1245/hadoop-hdfs/lib/*:/usr/hdp/2.5.0.0-1245/hadoop-hdfs/./*/usr/hdp/2.5.0.0-1245/hadoop-yarn/lib/*:/usr/hdp/2.5.0.0-1245/hadoop-yarn/./*/usr/hdp/2.5.0.0-1245/hadoop-mapreduce/lib/*:/usr/hdp/2.5.0.0-1245/hadoop-mapreduce/./*::mysql-connector-java-5.1.17.jar:mysql-connector-java.jar:/usr/hdp/2.5.0.0-1245/tez/*:/usr/hdp/2.5.0.0-1245/tez/lib/*:/usr/hdp/2.5.0.0-1245/tez/conf:mysql-connector-java-5.1.17.jar:mysql-connector-java.jar:/usr/hdp/2.5.0.0-1245/tez/*:/usr/hdp/2.5.0.0-1245/tez/lib/*:/usr/hdp/2.5.0.0-1245/tez/conf:/usr/hdp/current/hadoop-yarn-nodemanager/./*/usr/hdp/current/hadoop-yarn-nodemanager/lib/*:/usr/hdp/current/hadoop-client/conf/nm-config/log4j.properties
org.apache.hadoop.yarn.server.nodemanager.NodeManager "
    },
    {
        "pid": "1230",
        "name": "(python)",
        "pgrpId": 1175,
        "ppid": "1222",
        "sessionId": 685,
        "vmem": 1629884416,
        "rssmemPage": 8181,
        "utime": 1212980,
        "stime": 185595,
        "cpuusage": 0.3976143,
        "cmd": "/usr/bin/python /usr/lib/python2.6/site-packages/ambari_agent/main.py start "
    },
    {
        "pid": "20198",
        "name": "(java)",
        "pgrpId": 20196,
        "ppid": "20197",
        "sessionId": 20196,
        "vmem": 7400787968,
        "rssmemPage": 20257,
        "utime": 1676,
        "stime": 368,
        "cpuusage": 0.3976143,
        "cmd": "/usr/java/jdk1.8.0_102/bin/java -cp /usr/hdp/current/hbase-client/conf:/etc/hadoop/conf:/usr/hdp/2.5.0.0-1245/phoenix/bin/./phoenix-4.7.0.2.5.0.0-1245-client.jar:/usr/hdp/2.5.0.0-1245/phoenix/bin/./phoenix-4.7.0.2.5.0.0-1245-queryserver.jar -Dproc_phoenixserver
-Dlog4j.configuration=file:/usr/hdp/2.5.0.0-1245/phoenix/bin/log4j.properties -Dpsql.root.logger=INFO,DRFA
-Dpsql.log.dir=/var/log/hbase -Dpsql.log.file=phoenix-hbase-server.log org.apache.phoenix.queryserver.server.Main "
    }
    ...
]

```

10.5. Oozie JPA Service Extension 확인

<http://oozie호스트명:31000/query>

위의 URL로 접속시 아래와 같은 메시지가 나오면 정상입니다.


```
{
  "success": false,
  "message": "parameter 'query' is required"
}
```

10.6. Collector 확인

`http://collector호스트명:WAS포트/health`

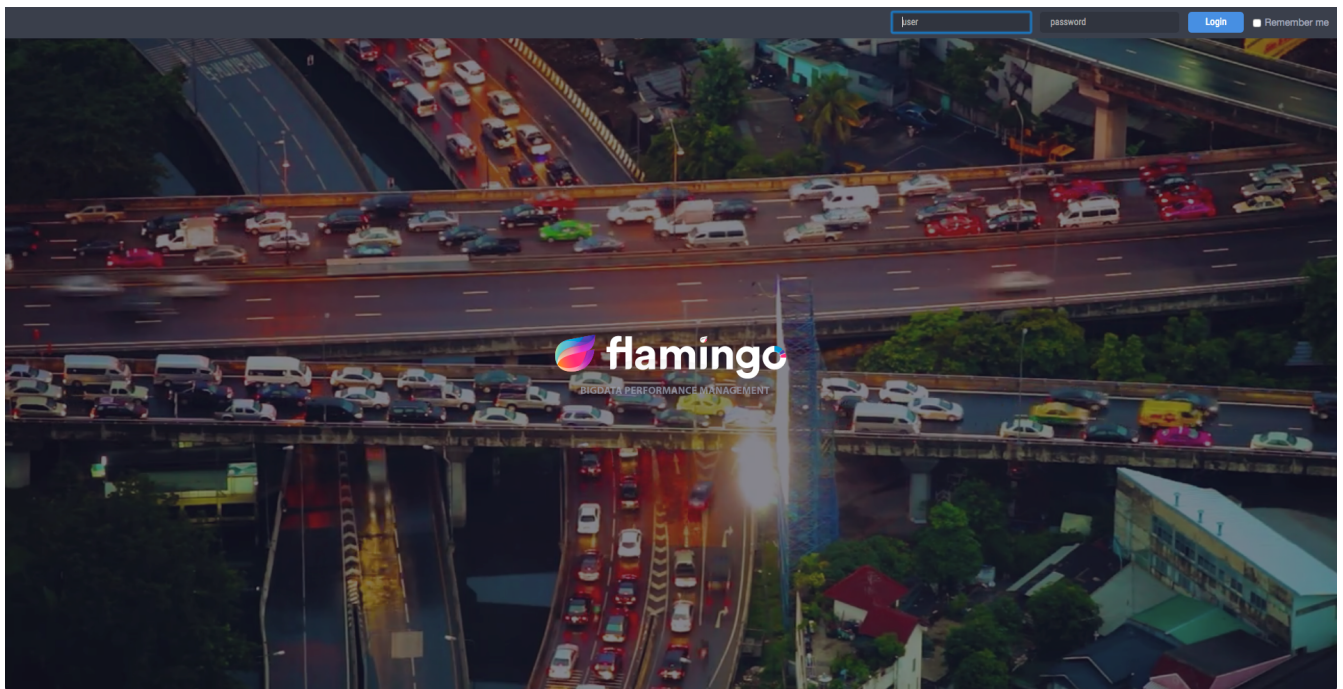
위의 URL로 접속 하면 아래 처럼 현재 Collector의 상태값이 JSON 형태로 리턴 됩니다.

```
{
  "status": "UP",
  "jms": {
    "status": "UP",
    "provider": "ActiveMQ"
  },
  "diskSpace": {
    "status": "UP",
    "total": 57993916416,
    "free": 41838891008,
    "threshold": 10485760
  },
  "db": {
    "status": "UP",
    "flamingoDataSource": {
      "status": "UP",
      "database": "PostgreSQL",
      "hello": 1
    },
    "phoenixDataSource": {
      "status": "UP",
      "database": "Phoenix",
      "hello": 1
    }
  }
}
```

10.7. Web 확인

`http://web호스트명:WAS포트/`

위 URL로 접속 시 아래와 같은 Flamingo Login 화면이 나옵니다. 초기 접속 계정은 admin, 패스워드는 1234 입니다.



로그인이 완료 후 Main Dashboard 화면이 나오는 것을 확인합니다.

