

DHP Architecture & Feature

Contents

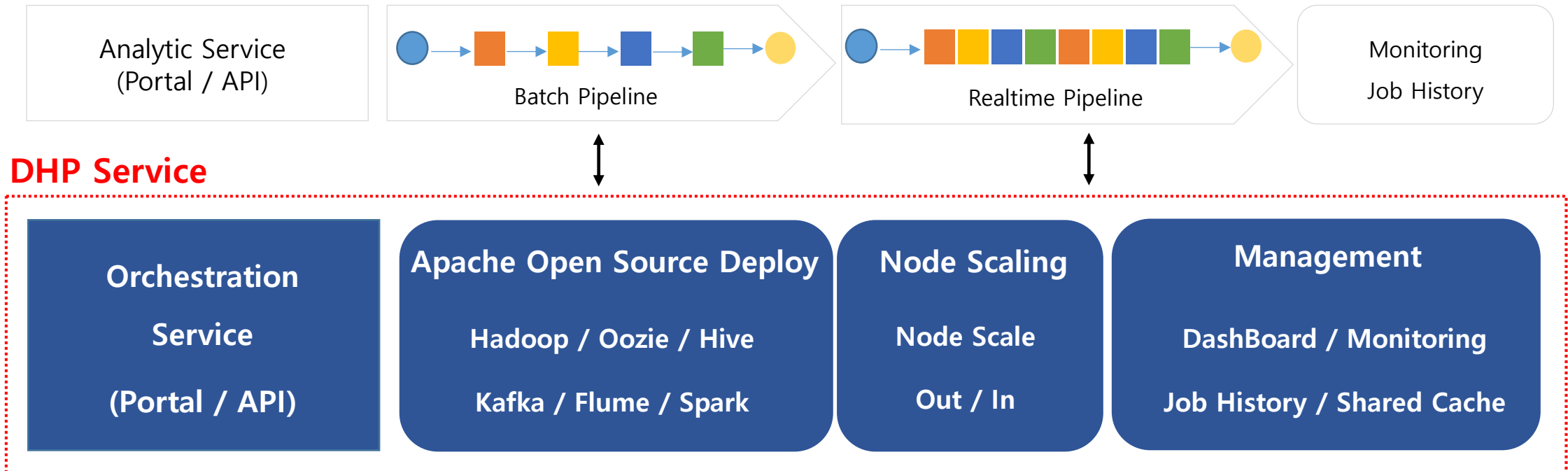
1. What is DHP?
 2. What is DHP Feature?
 3. Dev History (AS-IS, TO-BE)
 4. DHP Architecture (NW/System/Volume)
 5. Technology
 6. AWS EMR vs DHP (기능비교)
- # 별첨
1. Use Case
 2. Rancher 2.0

1. What is DHP (Dynamic Hadoop Provisioning)?

관리형 Hadoop 배포 서비스

① Batch processing(PipeLine), Streaming(Realtime Pipeline) 및 Machine Learning(ML Modeler) 등의 분석작업에 필요한 오픈 소스 데이터 도구를 쉽게 배포

② 이미지 기반의 하둡 클러스터를 자유롭게 생성/삭제 할 수 있으며, 이를 통해 관리에 소요되는 시간과 비용을 줄이고, 고객은 단지 데이터에 분석/처리에 집중

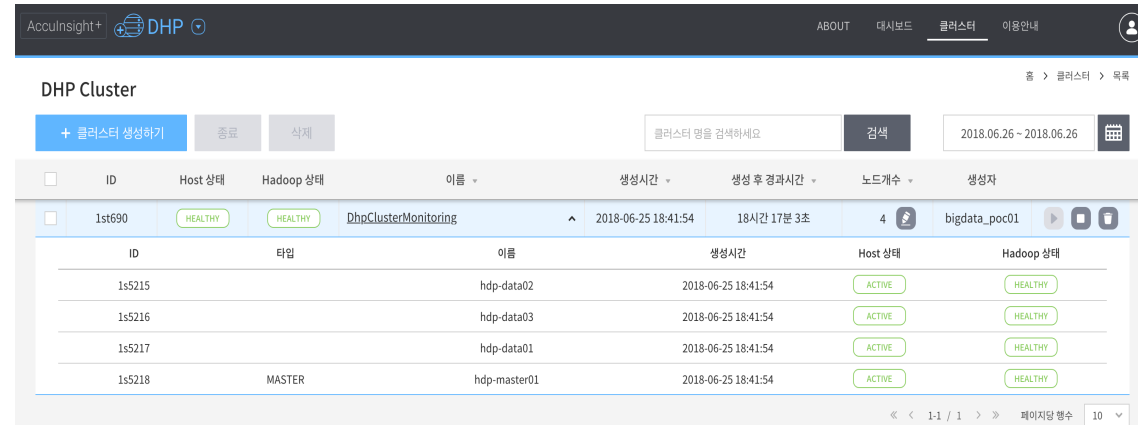


2. What is DHP Feature?

① 자동화된 클러스터 관리 및 사용자 정의 하둡 배포

- 편리한 클러스터 관리

자동 배포, 로그 기록이 관리되어 사용자는 하둡 클러스터 운영/관리가 아닌 데이터 분석에 집중



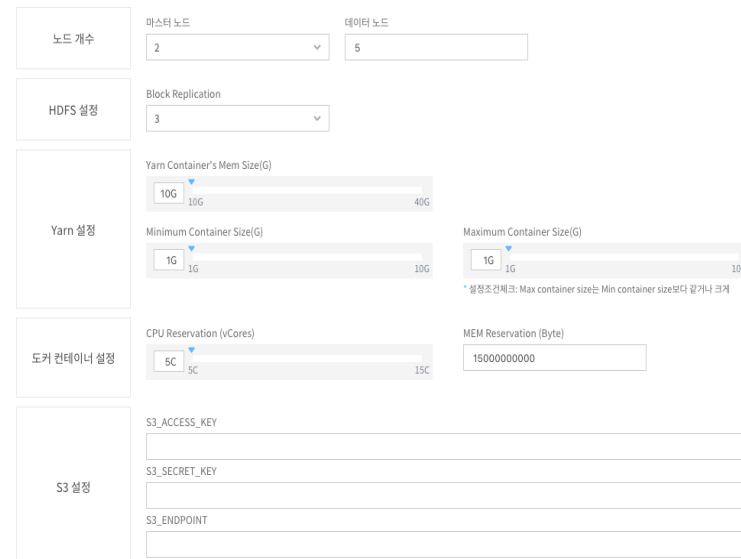
ID	Host 상태	Hadoop 상태	이름	생성시간	생성 후 경과시간	노드개수	생성자
1st690	HEALTHY	HEALTHY	DhpClusterMonitoring	2018-06-25 18:41:54	18시간 17분 3초	4	bigdata_poc01

ID	타입	이름	생성시간	Host 상태	Hadoop 상태
1s5215		hdp-data02	2018-06-25 18:41:54	ACTIVE	HEALTHY
1s5216		hdp-data03	2018-06-25 18:41:54	ACTIVE	HEALTHY
1s5217		hdp-data01	2018-06-25 18:41:54	ACTIVE	HEALTHY
1s5218	MASTER	hdp-master01	2018-06-25 18:41:54	ACTIVE	HEALTHY

- 선택 배포

하둡 클러스터 배포시 사용자 지정에 따라 다양한 유형의 Resource를 지정하여 배포 가능

하드웨어 구성



노드 개수: 2

마스터 노드: 2

데이터 노드: 5

HDFS 설정: Block Replication: 3

Yarn 설정: Yarn Container's Mem Size(G): 10G, Minimum Container Size(G): 1G, Maximum Container Size(G): 10G

도커 컨테이너 설정: CPU Reservation (vCores): 5C, MEM Reservation (Byte): 15000000000

S3 설정: S3_ACCESS_KEY, S3_SECRET_KEY, S3_ENDPOINT

소프트웨어 구성



소프트웨어 선택: DHP Ver.1.0

제공 애플리케이션:

- HDFS 2.7.3
- YARN 2.7.3
- MapReduce2 2.7.3
- Hive 1.2.1000
- Sqoop 1.4.6
- Oozie 4.2.0
- Zookeeper 3.4.6
- Kafka 0.10.1
- Spark 1.6
- Spark2 2.2.x
- Slider 0.92.0

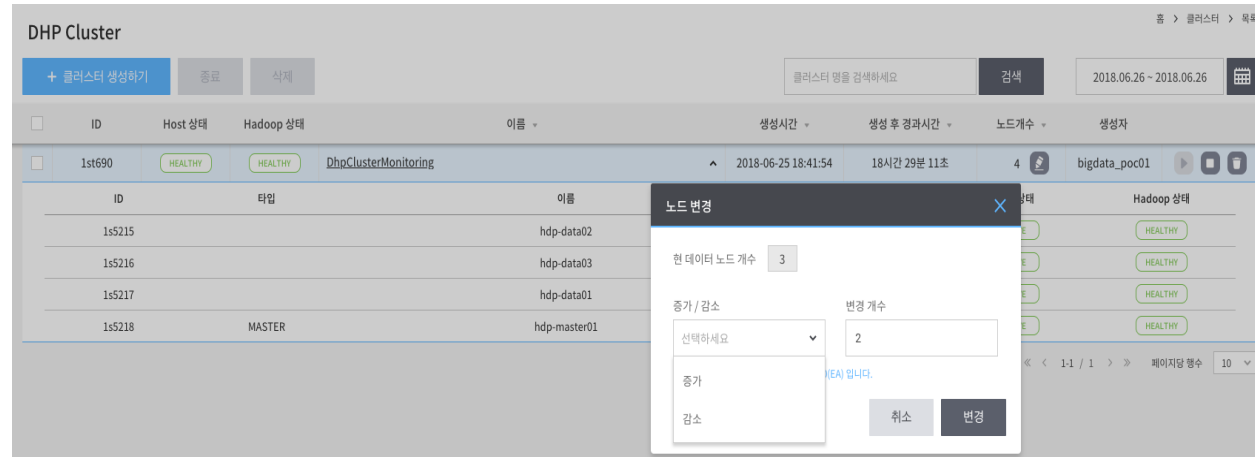
릴리스?

2. What is DHP Feature?

② On-Demand 형식의 자유로운 노드 증설/삭제

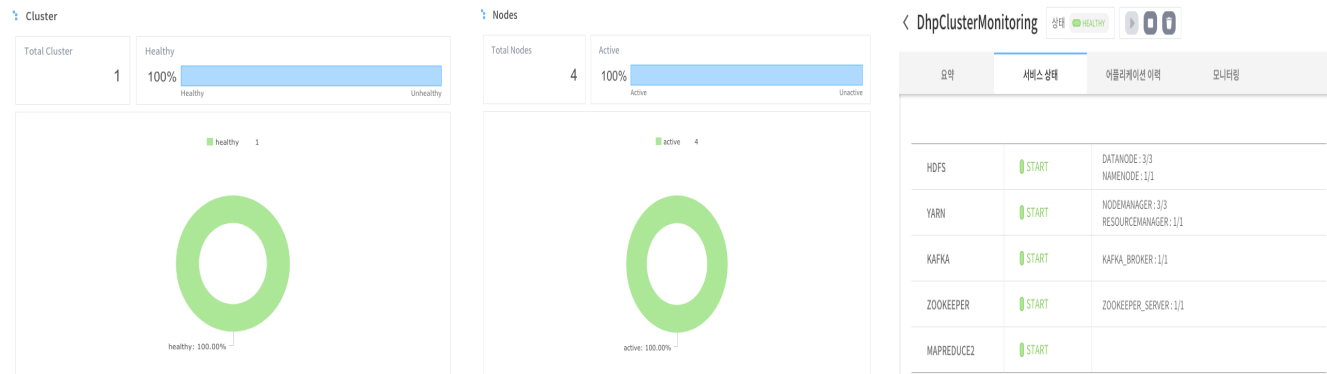
- 자유로운 노드 증설/삭제

Datanode 에 대한 탄력적인 증설/삭제
기능 제공



- 모니터링

클러스터 및 하둡 서비스 상태에 대한
모니터링 제공

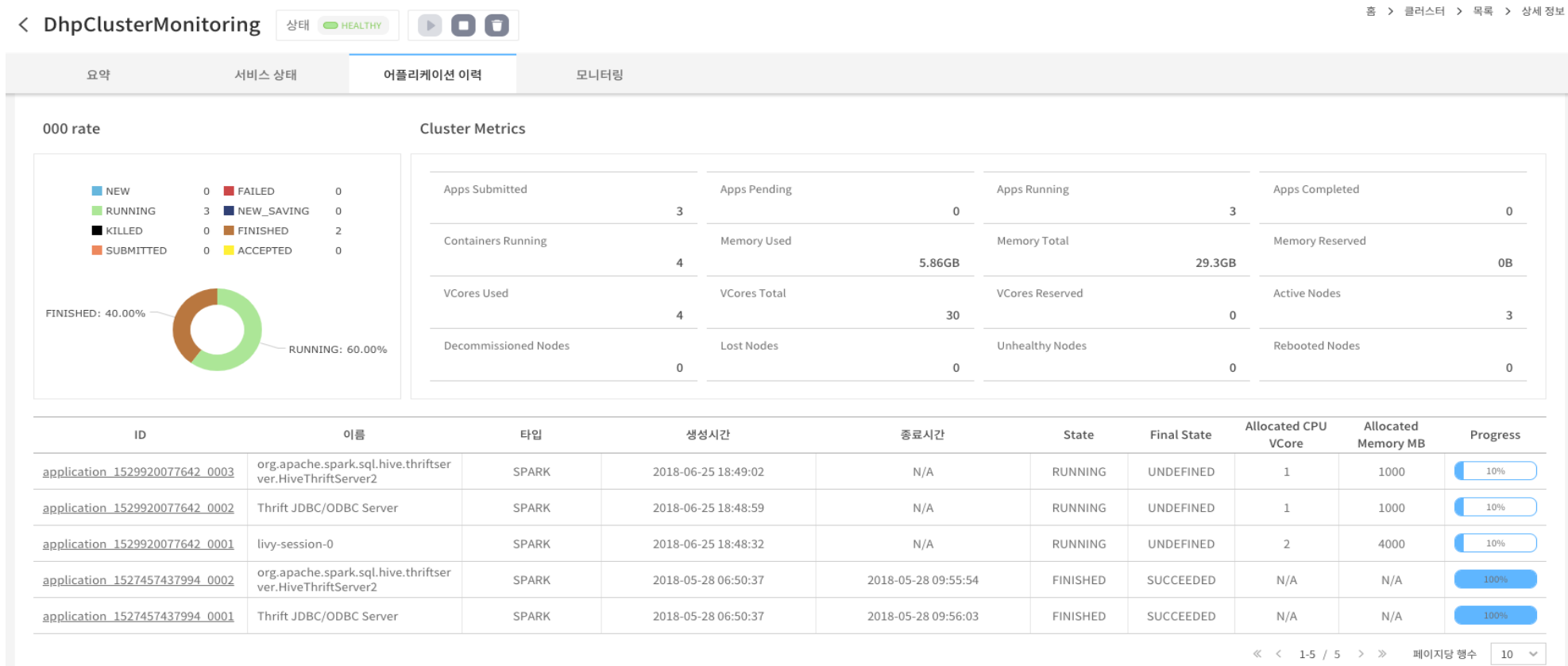


2. What is DHP Feature?

③ Job 이력 및 자원사용 내역 조회

- Job 이력 조회

클러스터를 통해 실행한 Job 이력/상태/자원사용 현황 등을 제공

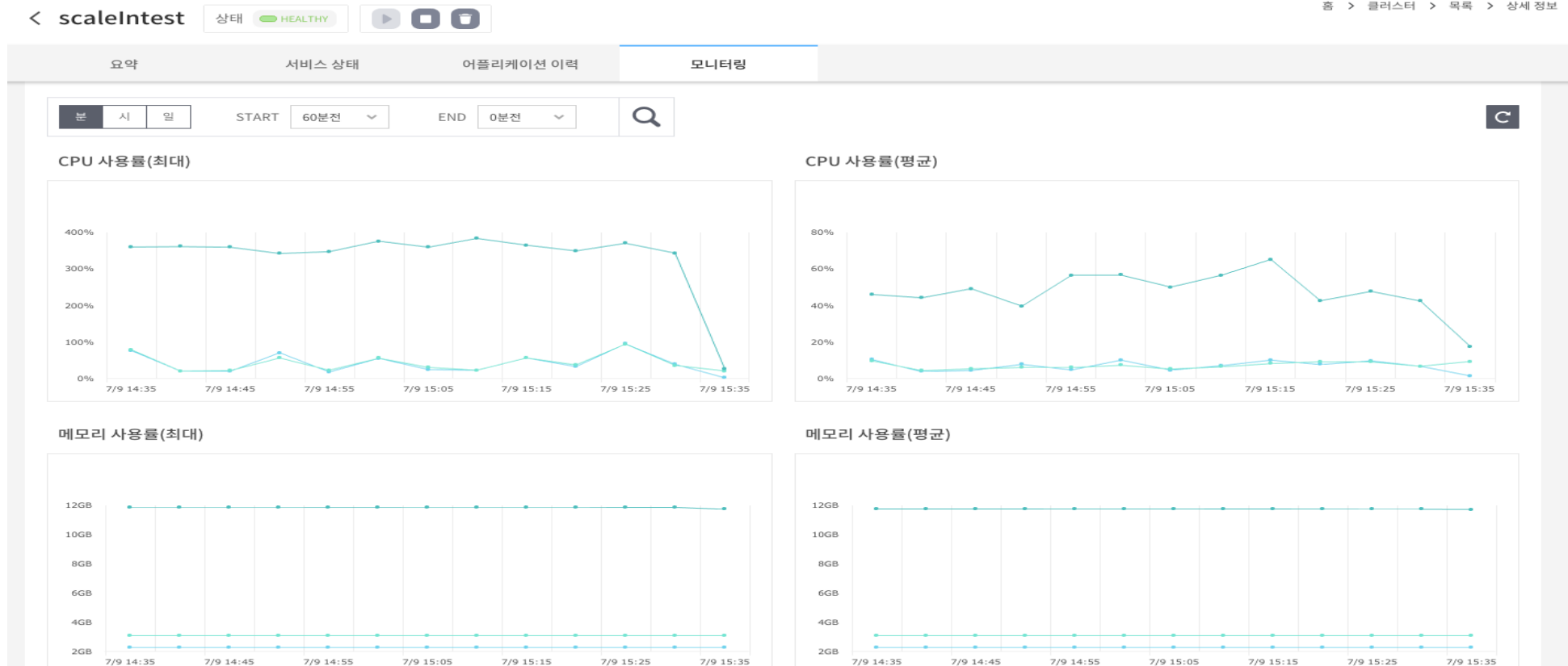


2. What is DHP Feature?

④ 노드(컨테이너) 모니터링

- **노드 모니터링**

배포된 노드에 대한 자원(vCore/Mem/BlockIO) 모니터링



3. Dev History (AS-IS)

항 목	'17년	'18년 6월말 기준	상태
멀티 클러스터 지원	1Cluster / 1User	nCluster / 1User	완료
배포방식 변경	Datanode 개수 지정 배포 (1/3/5/10)	Datanode 개수 임의 배포 (기본 1개 + 대수지정)	완료
Optional 배포	기능 없음	1) SW 구성 : SW 버전 선택 2) HW 구성 : 노드개수, HDFS/YARN, Container Size, S3 설정 등 3) Pyspark-jupyter 배포	완료
노드증설/삭제	기능 없음	데이터 노드 최대 40개까지 노드 증설 기본 노드(1개)를 제외한 노드 삭제	완료
Health 체크 (Host/Hadoop/Service)	기능 없음	1) DashBoard 기능 (Host) - 컨테이너 관점에서 cluster 및 node 에 대한 전체적인 상태 (대수에 따른 healthy/unhealthy 제공) 2) Host별 Hadoo 상태 (Hadoop) - 클러스터내 Host별 Hadoop 상태 체크 (클러스터 생성시 참고) 3) Hadoop 세부 서비스 상태 (Service) - 클러스터내 Hadoop 서비스 세부 상태 - HDFS/KAFKA/Oozie/Hive/YARN/Zookeeper/Spark1,2 등	완료
Application 이력조회	기능 없음	1) Cluster Metrics 제공 - Container Running, Mem Used/Total, Vcore Used/Total 등 2) APP ID 별 Job History 제공 - App ID에 따른 user, start/end time, State(SUCCEED/Fail/Running) 등 - Job Fail 시 Diagnostics	완료
모니터링	기능 없음	클러스터내 컨테이너 모니터링 제공 - 기본 Metric(CPU/MEM/BlockIO등) 제공	완료

3. Dev History (TO-BE)

Rancher2.0 

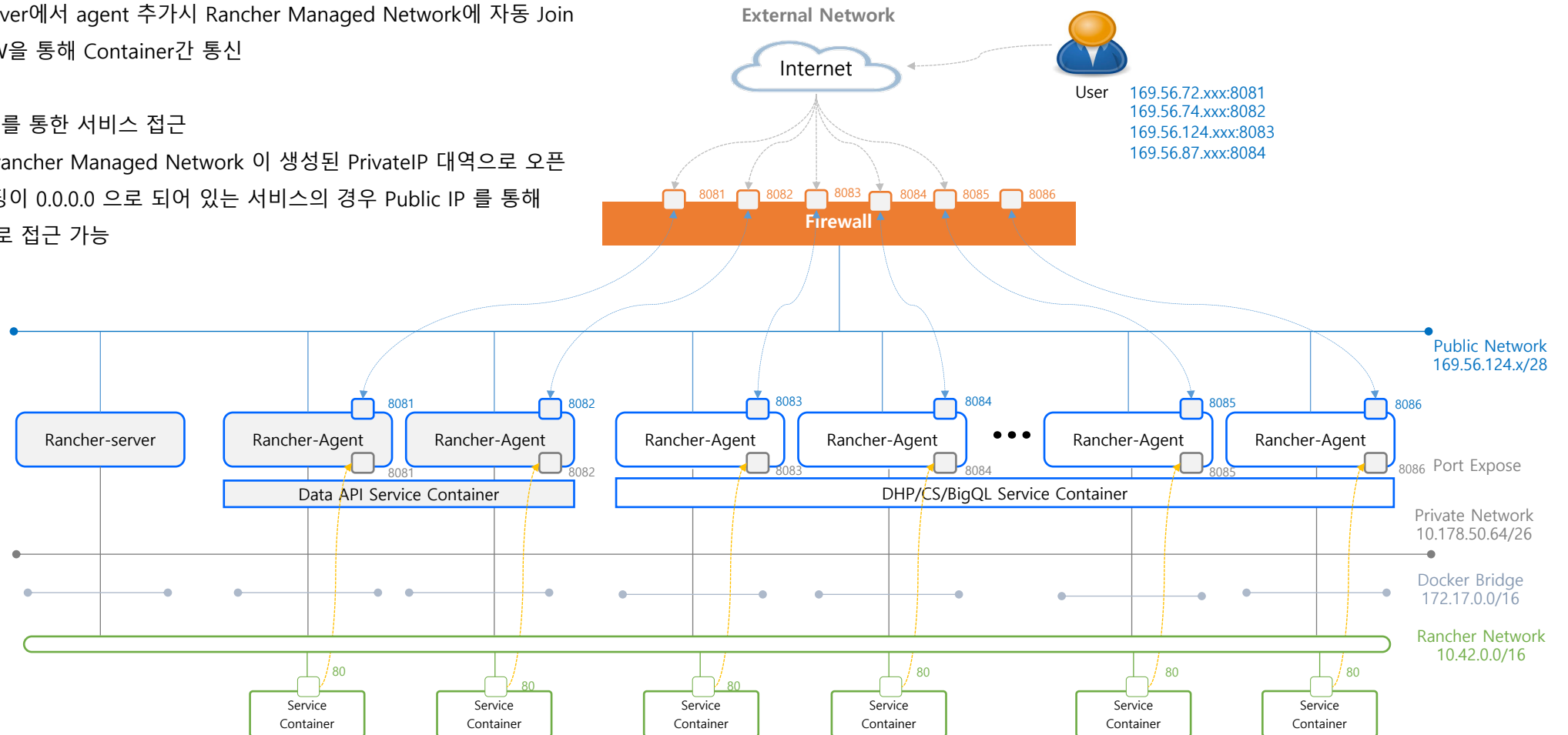
항 목	내 용	일 정
Kubernetes 마이그레이션 검토 & 실행	Kubernetes 전환을 위한 검토 및 계획 - ZCP / rancher2.0 / ICCS 검토 마이그레이션에 따른 API, Image, 배포방식, 화면 재개발	~'19.03 (단계별 수행)
단계실행 기능 추가	Application 작업유형에 따른 클러스터 생성 및 삭제 - Hive, Spark 등 사용자 job 선택후 실행결과에 따른 클러스터 삭제 기능	-
노드 증설 기능 추가 (Computing/Datanode 분리)	Computing node 만 증설 기능 - 컴퓨팅 노드 / 저장 노드 선택에 따른 증설 클러스터 노드 증설/삭제 API 추가 개발	-
배포 이미지 추가	HDP 이미지외 apache Hadoop 및 ECO 이미지 추가 - 추가 가능한 이미지 선정 후 클러스터 생성 API 추가 개발	-

4. DHP Architecture

- Network Architecture

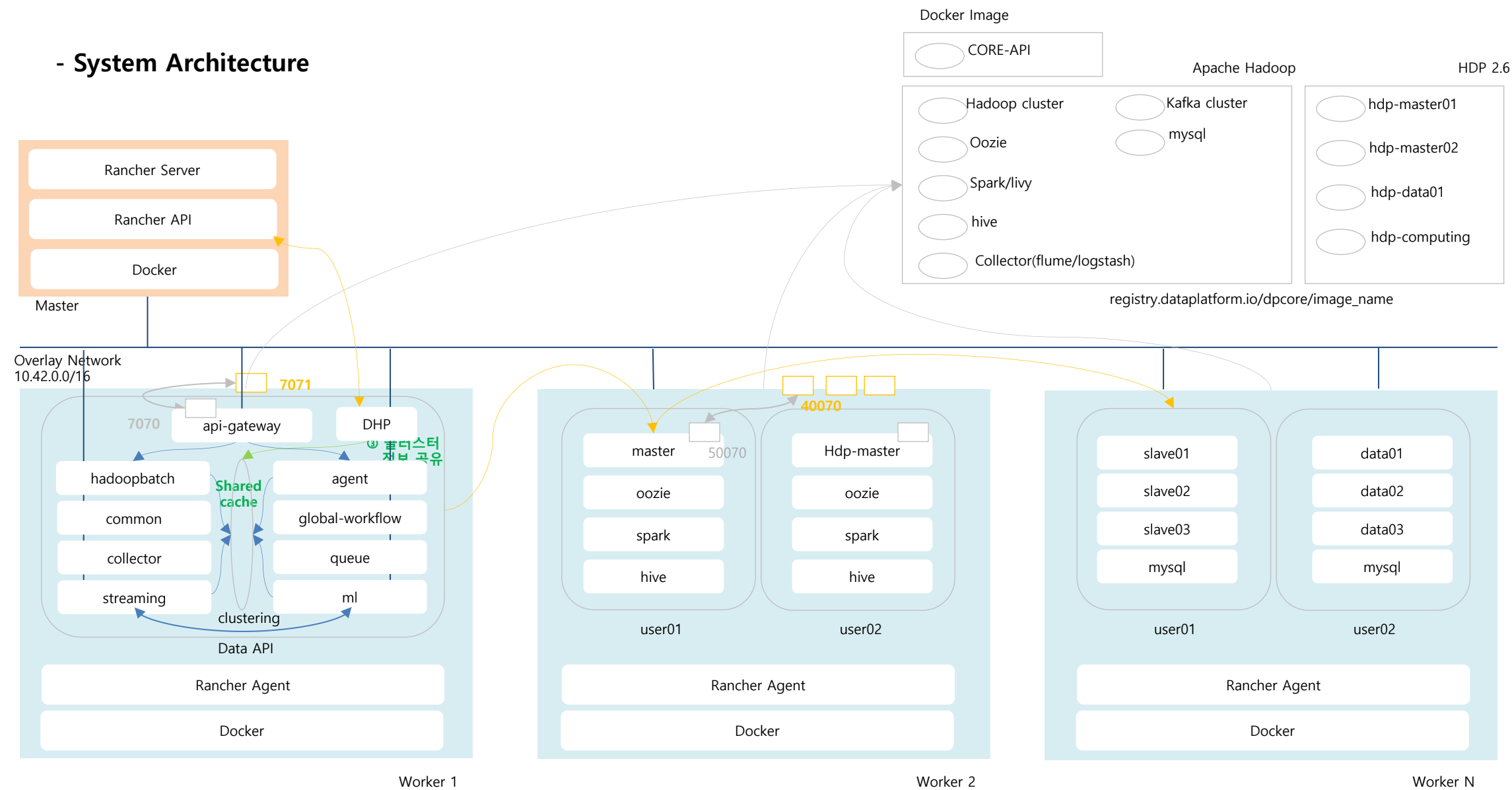
[기본 Infra 구성]

1. Rancher-server(Master) 가 Rancher-agent(Worker) 를 관리
 - Rancher server에서 agent 추가시 Rancher Managed Network에 자동 Join
 - Rancher NW를 통해 Container간 통신
2. 서비스 접근
 - Port Expose를 통한 서비스 접근
 - : Expose시 rancher Managed Network 이 생성된 PrivateIP 대역으로 오픈
 - : Port 바인딩이 0.0.0.0 으로 되어 있는 서비스의 경우 Public IP 를 통해 해당 포트로 접근 가능



4. DHP Architecture

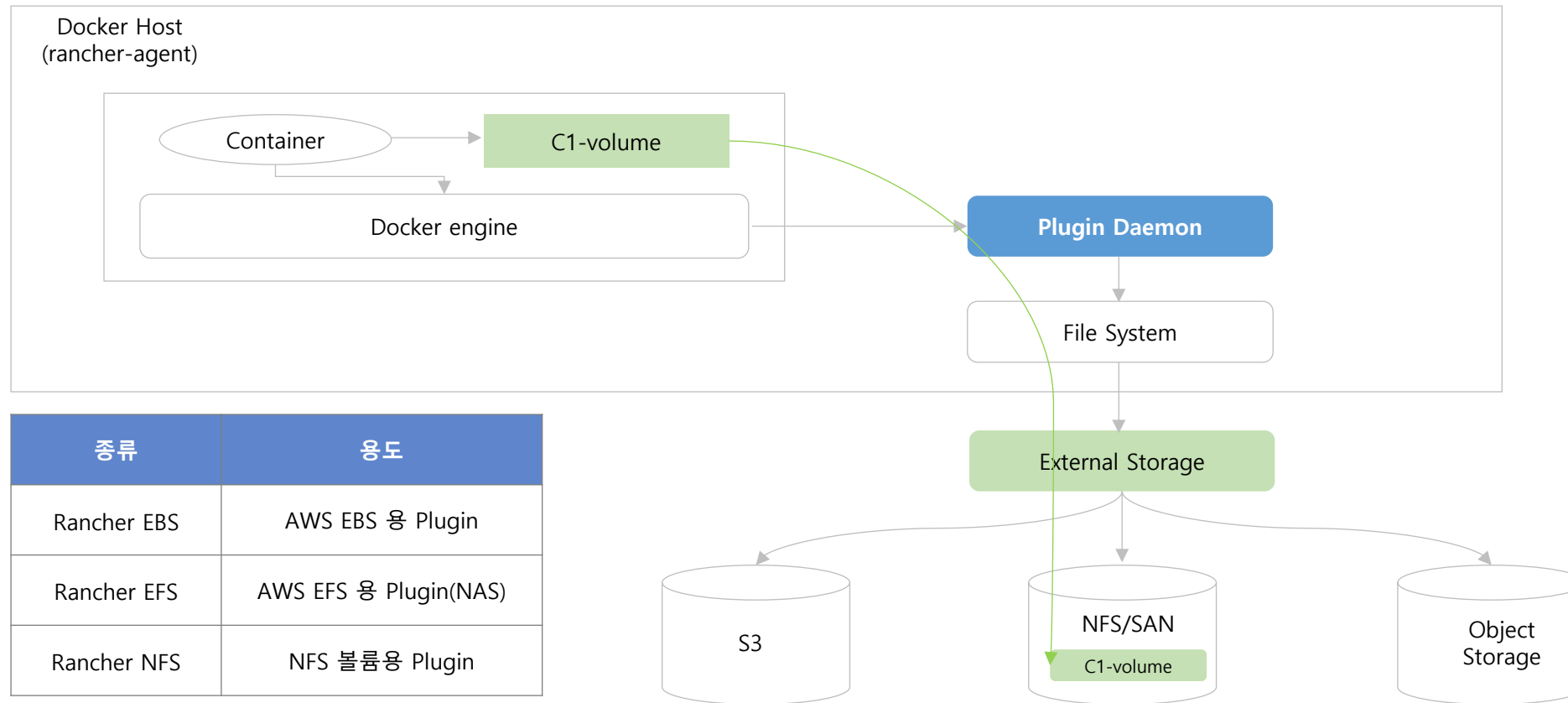
- System Architecture



4. DHP Architecture

- Volume Architecture

- Persistent Storage 구성을 위해 Rancher에서 제공하는 Plugin을 설치하여 사용 가능



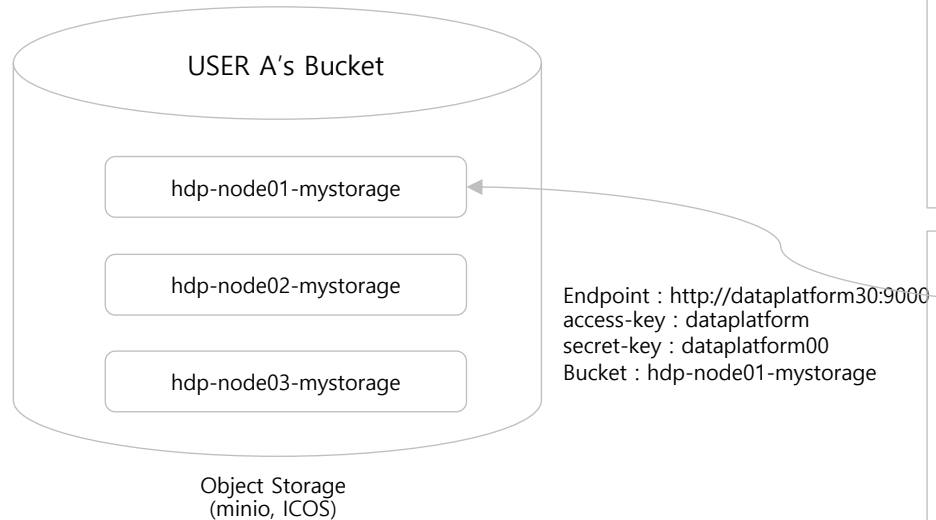
* DHP 는 컨테이너 특성상(Hadoop) External Storage를 CS(Container Storage)로 활용하지 않음

- Rancher-nfs 자체 테스트 결과 과부하시 Volume Read-only 현상 발생 (단, Storage NW 및 서버 상태에 따라 유동적이며 단위별 Test 필요)

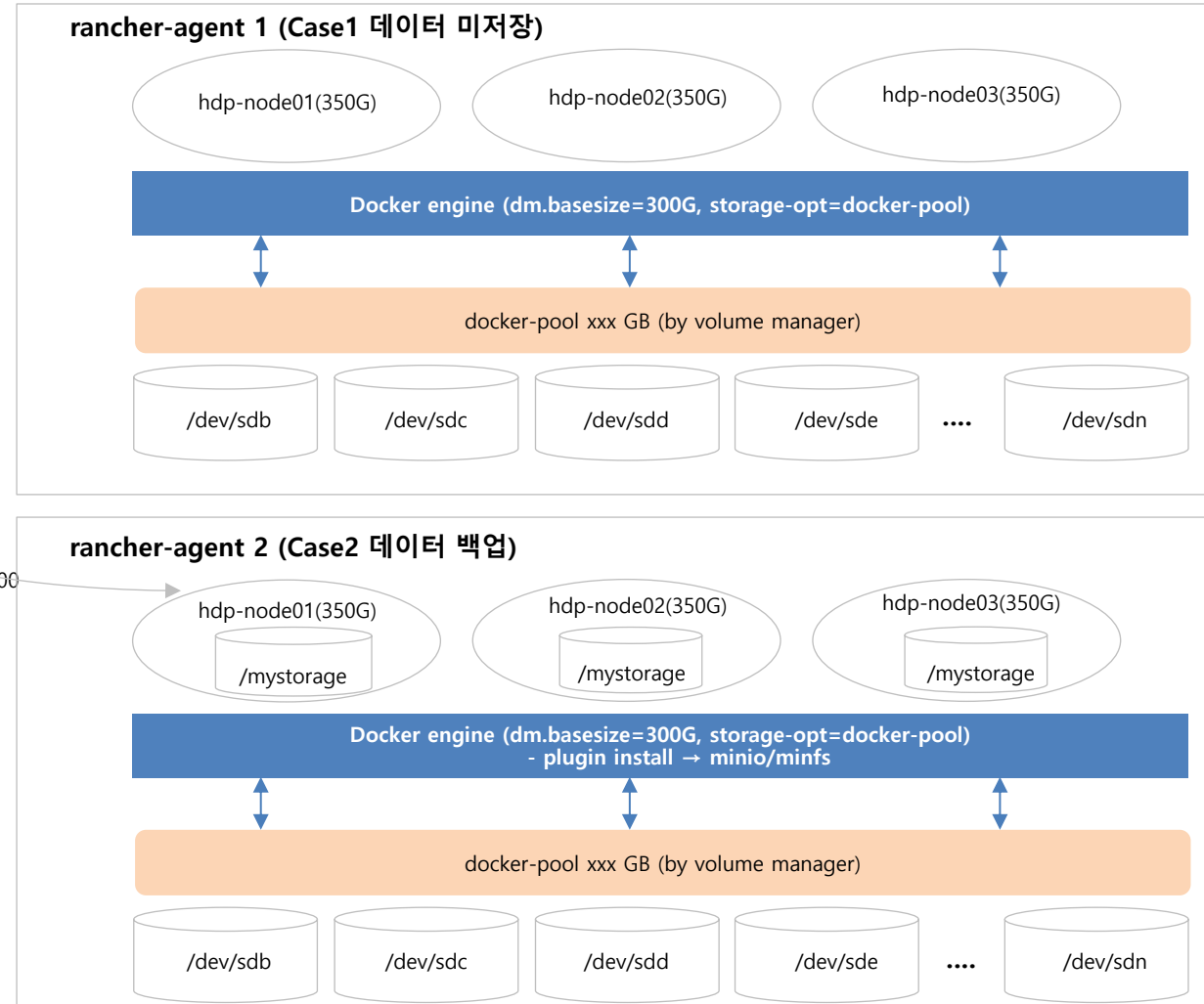
4. DHP Architecture

- Volume Architecture

- 별도 볼륨 Plugin 없이 서버 I-DISK 사용하고 백업 필요시 Object Storage 이용



※ PRD Service를 위한 Persistent Storage 는 S3 혹은 ICOS 등을 활용 예정
→ 멀티터넌시 필요 (사용자별 endpoint 를 통해 데이터 저장)



5. Technology

기술확보

진행예정

Container Service	<ul style="list-style-type: none">■ Image Deploy<ul style="list-style-type: none">- HDP 2.6- Apache Hadoop 2.7.3- Spark/Oozie/Hive/Kafka Cluster Collector(Flume/logstash)	<ul style="list-style-type: none">■ Node Scaling<ul style="list-style-type: none">- ambari API- rancher API	<ul style="list-style-type: none">■ Monitoring<ul style="list-style-type: none">- Container- Hadoop Status	<ul style="list-style-type: none">■ Metering/Billing■ 기능 고도화■ Orchestration Tool Migration 검토 (CATTLE→K8s)
Container Technology	<ul style="list-style-type: none">■ Container Platform<ul style="list-style-type: none">- Docker 17.06.2-ce	<ul style="list-style-type: none">■ Orchestration<ul style="list-style-type: none">- Rancher 1.6.x	<ul style="list-style-type: none">■ Container Defining Tool<ul style="list-style-type: none">- docker-compose 1.16.1- rancher-compose	<ul style="list-style-type: none">■ Volume Plugin<ul style="list-style-type: none">- devicemapper- rancher-nfs/ebs/efs- minFS, rClone
	<ul style="list-style-type: none">■ Service Discovery<ul style="list-style-type: none">- Consul 0.11.0	<ul style="list-style-type: none">■ Network<ul style="list-style-type: none">- IPSEC Overlay Network	<ul style="list-style-type: none">■ Registry<ul style="list-style-type: none">- Local Registry	<ul style="list-style-type: none">■ Dev Tools<ul style="list-style-type: none">- Vertx 3.4.2- spring-boot 1.5.2- hazelcast 3.6.3- GitHub
Infra	<ul style="list-style-type: none">■ Infrastructure<ul style="list-style-type: none">- On-promise (BM/VM)- CloudZ			

6. AWS EMR vs DHP (기능비교)

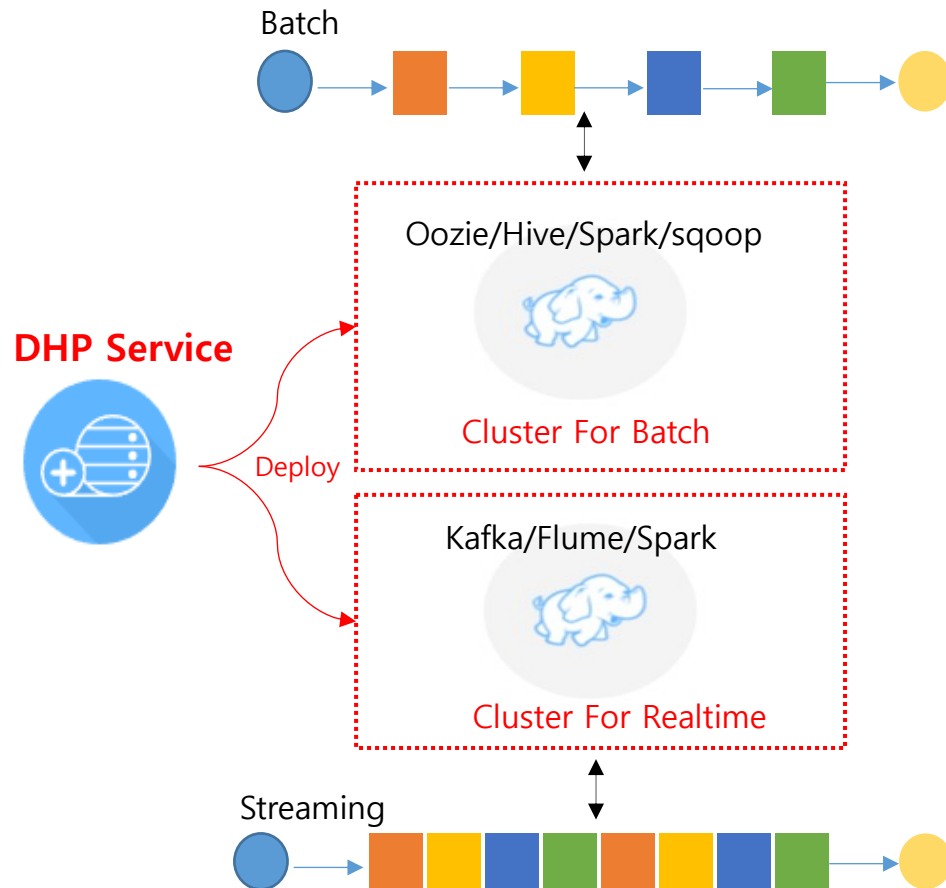
항 목	설 명	AWS EMR	DHP
클러스터 배포시 노드 Type	노드를 배포를 위한 Resource Type	VM	Docker Container
배포 노드 자원 지정	클러스터 노드생성을 위한 자원 선택	O (Instance Type 선택)	O (Docker vCore/Mem 지정)
클러스터 기본생성	노드 개수 선택에 따른 클러스터 생성	O	O
단계별 실행	사용자 어플리케이션 실행을 위한 노드 생성	O	X (개발 예정)
노드 증설 / 삭제	Running 중인 클러스터에 데이터노드 추가/삭제	O (VM)	O (Docker Container)
Task Node 생성	Master-Slave node 외 컴퓨팅 노드 추가 기능	O	X (개발 예정)
Application Job 이력조회	YARN 어플리케이션 정보를 수집/이력 제공	O	O
사용자 지정 배포	YARN Mem 및 Yarn Container Size 지정 배포	O	O
주요 API 제공	생성/삭제/증설/이력조회 등 주요 API 제공	O	O

#별첨1) Use Case (Integrated Product) – 탄력적이고 다양한 분석 환경

Batch/Realtime Pipeline 을 위한 탄력적인 분석환경 제공

Case Architecture

Batch/Streaming 과 함께 다양한 분석환경 제공



개념/절차

Batch/Streaming PipeLine 을 위한 분석환경 제공

- 1 Batch 및 Streaming Pipeline Job 수행을 위한 Processing Resource 제공
- 2 업무별 하둡 클러스터 선택에 따른 워크로드 분리
- 3 Job 수행을 위한 최적화된 자동 구성
 - Batch/Streaming 에 필요한 Jar 및 기본 패키지가 설치되어 배포
 - Oozie 사용을 위한 ShareLib 및 Job 수행을 위한 사용자 및 권한 자동 생성
 - Job 수행을 위한 Config 튜닝 등

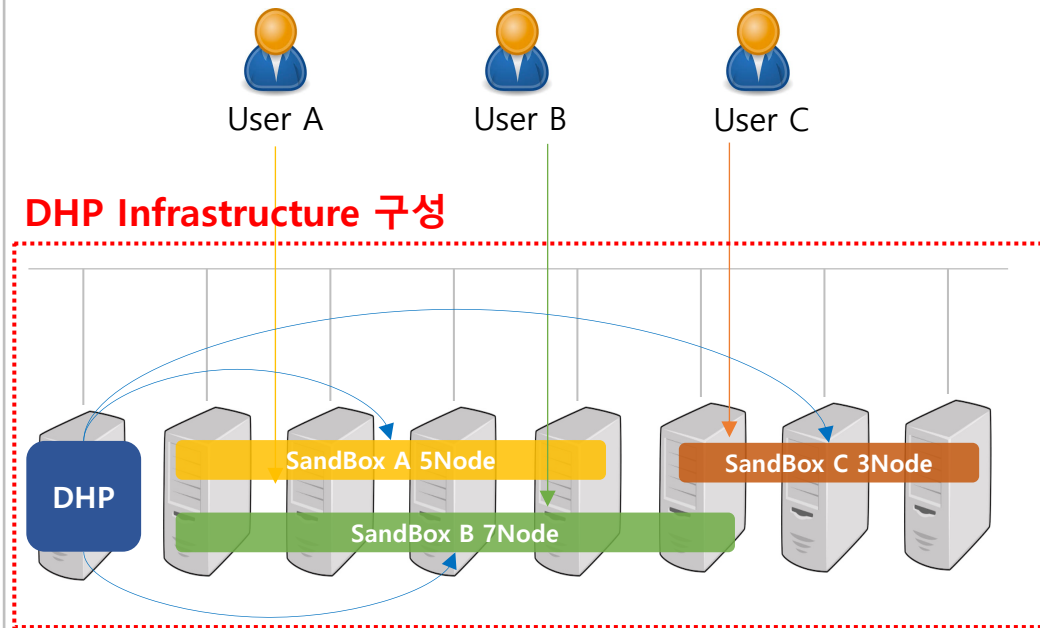
#별첨1) Use Case (Low Cost, Fast) – Sandbox Hadoop Cluster

기업내 Data Scientist 를 위한 sandbox 용 하둡 클러스터 사용

Case Architecture

On-premise 자원 활용 Multi-Hadoop 배포

- 버전별 다양한 이미지를 통해 원하는 서비스 자동 배포
- 기업내 서버 자원 활용하여 자원 효율적인 분석 실험환경 구성



* HDP 2.6.x, Apache Hadoop 2.7.x, Pyspark-Notebook, Kafka, Oozie, Hive 등

개념/절차

자원 효율화를 통한 분석가용 sandbox로 활용

- 1 On-premise 자원을 활용하여 DHP 구성
- 2 분석가는 개인용 sandbox 클러스터를 배포
- 3 Internal RDB 에서 DHP 클러스터로 데이터 마이그레이션 수행(Sqoop등 활용)
- 4 데이터 분석 시작 및 필요시 탄력적으로 노드 증설/삭제 수행

#별첨1) Use Case (Seamless) – 유연한 데이터 분석

Batch Pipeline을 통한 RDB 데이터 분석

Case Architecture

데이터 로딩 및 Batch Pipeline사용한 데이터 분석

→ 기업의 외부 혹은 내부 RDB 데이터를 하둡을 통해 분석

DHP Service



Deploy

AWS EC2

Mysql
Server

sqoop import Job Execute



Oozie Job Execute

RDB 분석용 클러스터

Batch PipeLine



개념/절차

Sqoop 활용 데이터 이관 및 분석

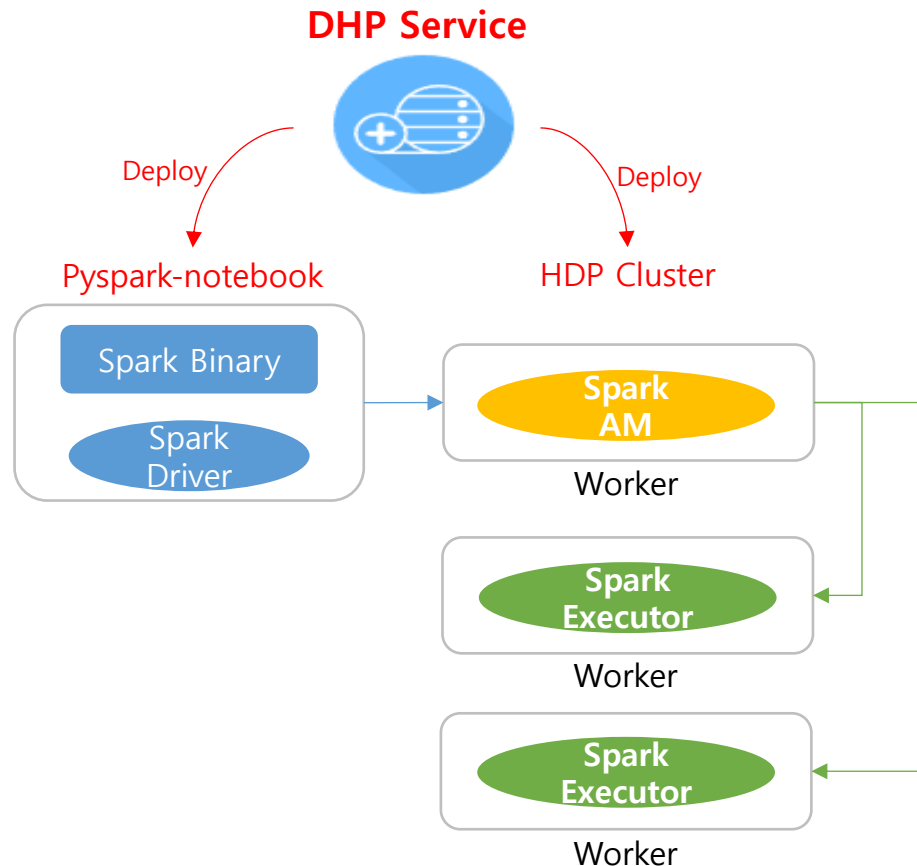
- 1 EC2 에서 Remote 접근을 위한 설정
- 2 DHP 클러스터 배포
 - jdbc driver는 클러스터 배포시 자동 설치
- 4 DHP 클러스터에서 remote로 DB에 접속 가능한지 확인
- 5 sqoop노드에서 import Job을 통한 데이터 마이그레이션
- 6 Pipeline 구성후 주기적으로 Job 수행 및 결과 저장

#별첨1) Use Case (Integrated, Seamless) – Interactive 데이터 분석

HDP와 Pyspark-jupyter notebook 을 활용한 Interactive 데이터 분석

Case Architecture

HDP / Pyspark 배포를 통한 분석환경 구성



개념/절차

DHP Yarn을 통한 Spark 실행

- 1 DHP 클러스터 배포시 pyspark-notebook 설치 (옵션 체크)
 - 2 pyspark-notebook endpoint 확인 및 접속
 - 3 notebook Terminal 활용 스크립트 upload
 - 4 HDFS로 Source Data Migration
 - 5 Spark yarn-client mode 활용 Interactive Data 분석
- * Jupyter notebook 자동 구성 내역
 - Spark1.6/2.2 binary, Anaconda, python등의 Lib 및 기본 패키지 설치
 - ssh를 통한 DHP 클러스터로 접속 가능

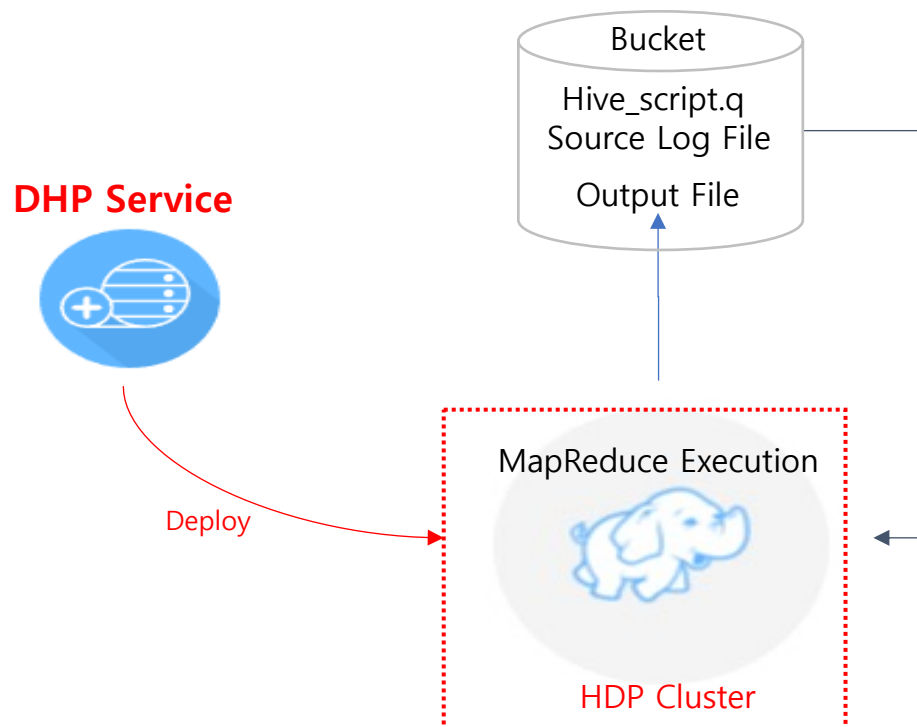
#별첨1) Use Case (User Specific) – 사용자 정의 Application 실행

사용자 정의 Hive 어플리케이션 실행 (추후 단계별 실행 기능을 통해 다양한 Application 실행 환경 지원 예정)

Case Architecture

Object Storage에 있는 사용자별 Application 실행

→ Object Storage 저장된 데이터를 사용자별 APP을 통한 실행

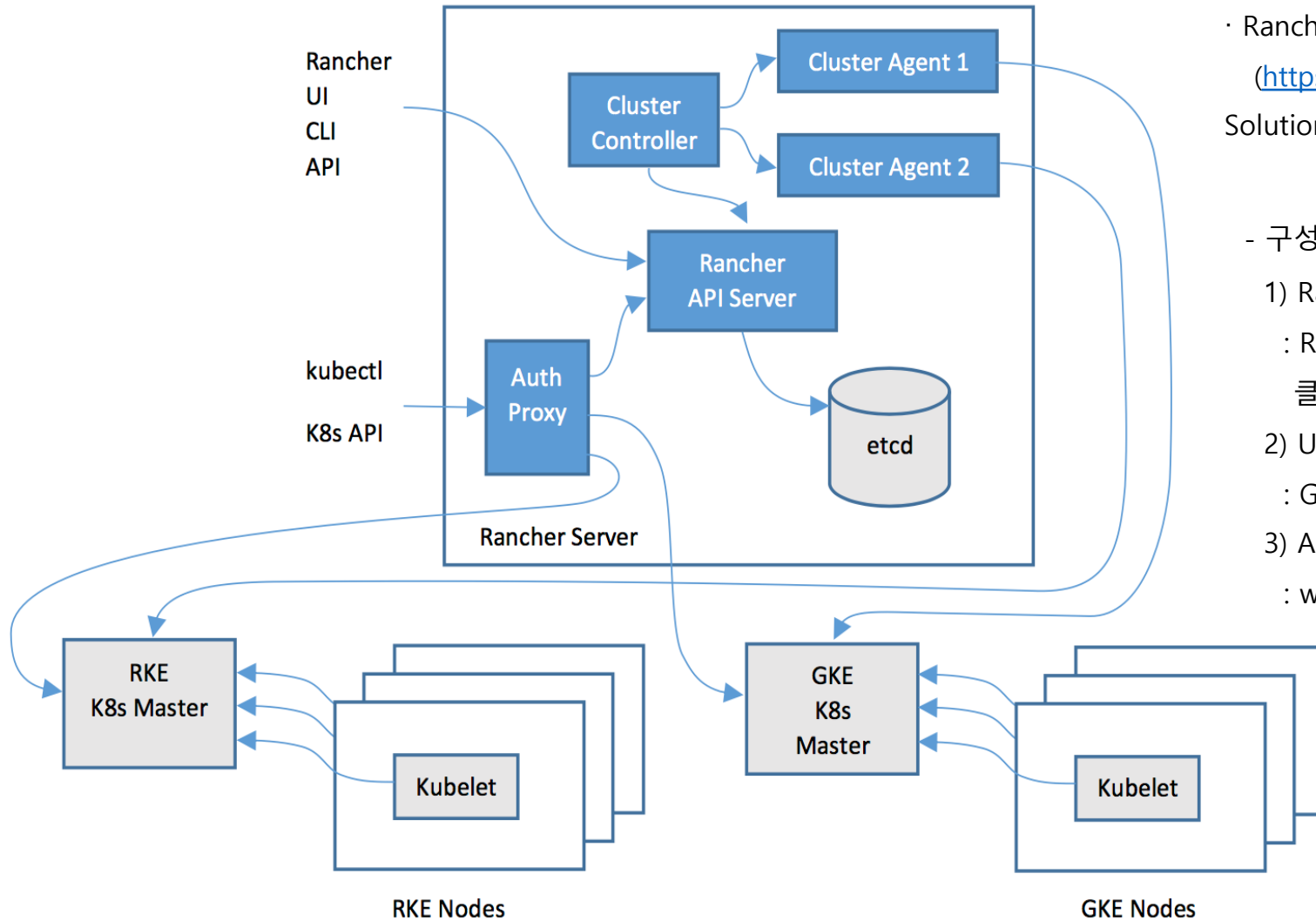


개념/절차

DHP를 활용한 Hive 어플리케이션 실행

- 1 DHP 클러스터 배포시 S3 정보 입력
(Access/Secret/Endpoint)
- 2 어플리케이션을 위한 스크립트 및 로그데이터 S3 upload
- 3 클러스터 내부에서 S3 endpoint를 활용한 Hive Job 스크립트 실행
- 4 Job 수행후 결과내역 S3 저장
- 5 클러스터 삭제

#별첨2) Rancher2.0



· Rancher 2.0은 Kubernetes를 기반으로하는 컨테이너 관리 플랫폼
(<https://kubernetes.io/docs/setup/pick-right-solution/> Turnkey Cloud Solutions 에 등록)

- 구성요소

1) Rancher Kubernetes Engine (RKE)

: RKE는 Kubernetes 는 VMware 클러스터, BM, VM 에서 Kubernetes 클러스터를 배포 가능

2) Unified Cluster Management

: GKE, EKS 및 AKS와 같은 관리 Kubernetes 서비스와 통합

3) Application Workload Management

: workload 관리 및 배포를 위한 직관적인 사용자 UI 제공

#별첨2) Rancher2.0


- 다양한 형태로 제공되는 Public Cloud에 클러스터 배포 가능


Rancher


← → ↻ ⚠ 안전하지 않음 | <https://169.56.87.59/g/clusters/add>


Global ▾ Clusters Node Drivers Catalogs Users Settings Security ▾


Add Cluster



Google Container Engine



Amazon EKS



Azure Container Service



Amazon EC2



Microsoft Azure


DigitalOcean


SoftLayer


vSphere


CUSTOM


IMPORT

Cluster Name * Add a Description

e.g. sandbox

Member Roles

Control who has access to the cluster and what permission they have to change it.

Service Account * Read from a file

Service account private key JSON file

Create a [Service Account](#) with a JSON private key and provide the JSON here. See [Google Cloud docs](#) for more info about creating a service account. Three IAM roles are required: `computer/viewer`, `project/viewer`, `kubernetes-engine/admin` and `service-account/user`, more info on roles can be found [here](#).

#별첨2) Rancher2.0

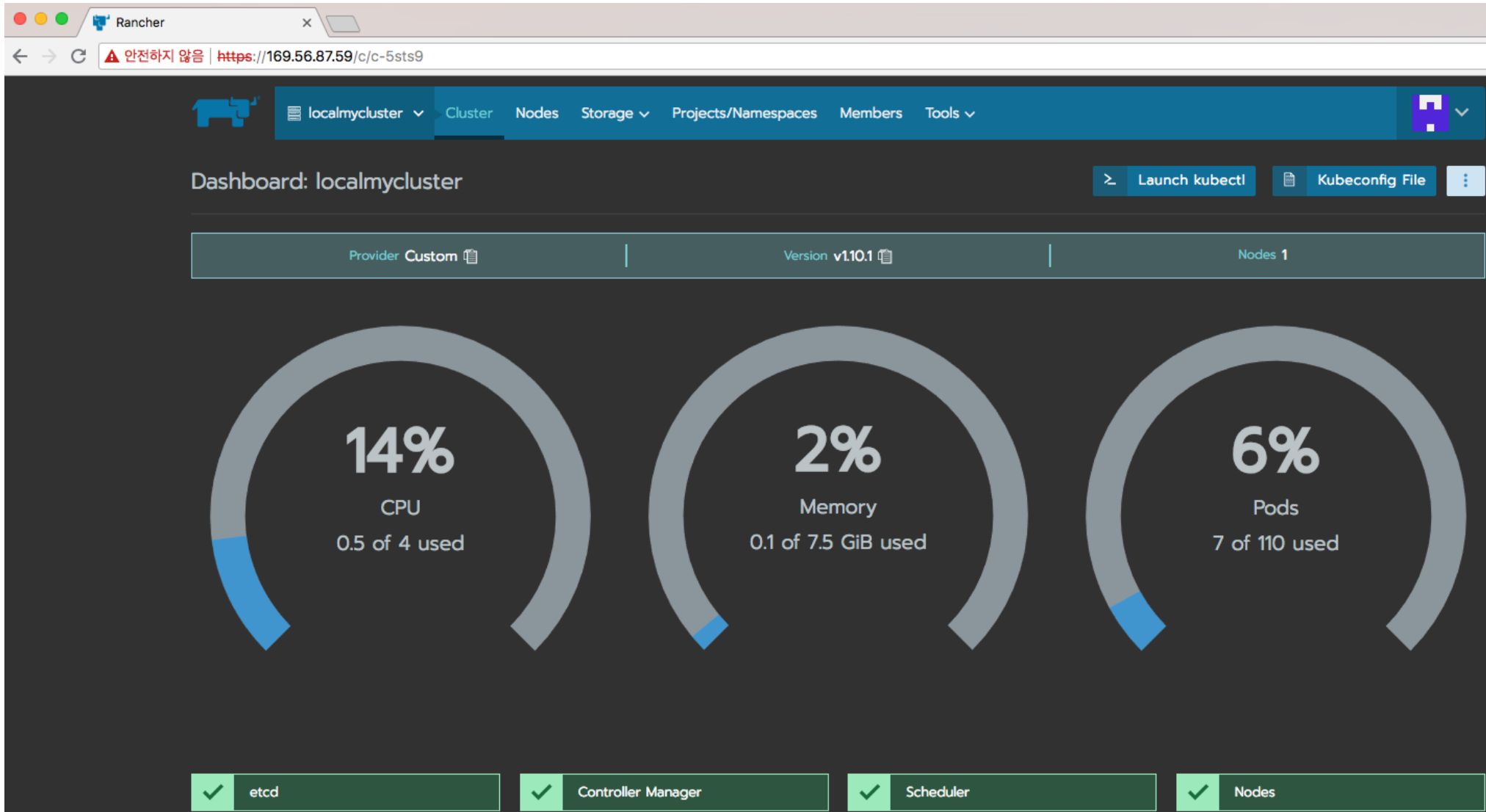
- 통합 클러스터 관리 (예시)

The screenshot displays the Rancher 2.0 web interface for managing clusters. The top navigation bar includes links for Global, Clusters, Node Drivers, Catalogs, Users, Settings, and Security. The main content area is titled 'Clusters' and features an 'Add Cluster' button. Below this, there is a table listing the clusters. The table has columns for State, Cluster Name, Provider, Nodes, CPU, and RAM. Two clusters are shown: 'icpmycluster' (Imported, 0 nodes) and 'localmycluster' (Custom, 1 node). Both clusters are in an 'Active' state.

State	Cluster Name	Provider	Nodes	CPU	RAM
Active	icpmycluster	Imported v17.3-7*154699da4767fd	0	n/a	n/a
Active	localmycluster	Custom v1.10.1	1	0.5/4 Cores 14%	0.1/7.5 GiB 2%

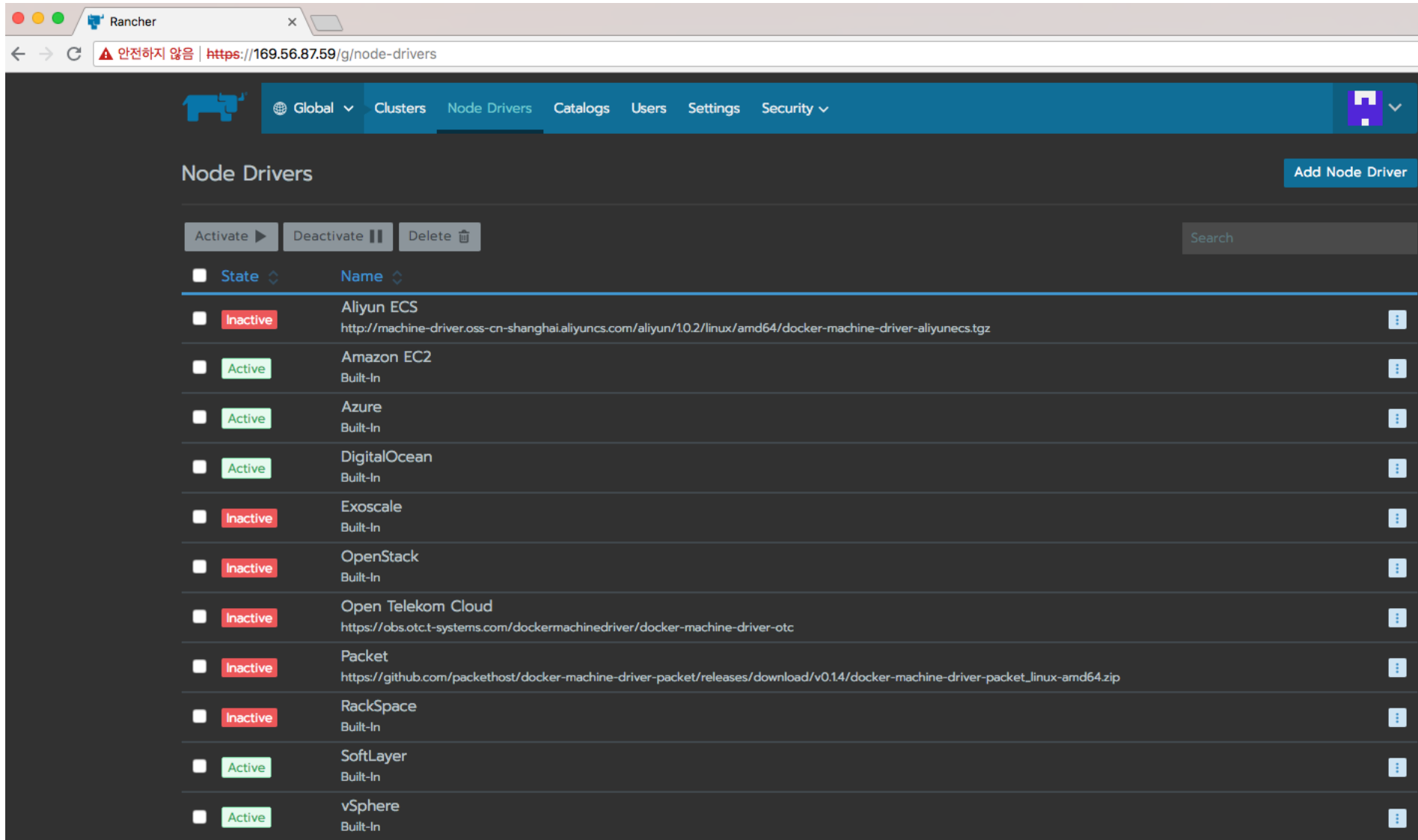
#별첨2) Rancher2.0

- 통합 클러스터 관리 (노드관리)



#별첨2) Rancher2.0

- 사용 가능한 다양한 드라이버 제공



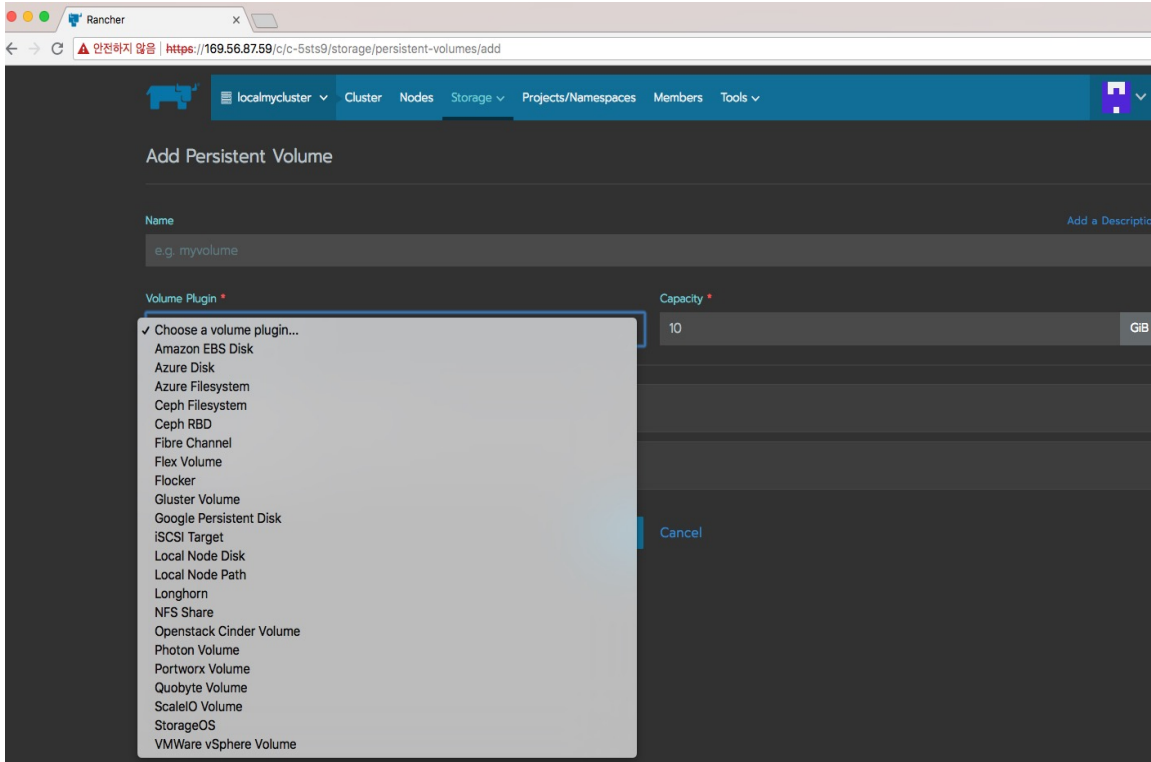
The screenshot displays the Rancher2.0 web interface, specifically the 'Node Drivers' section. The top navigation bar includes links for Global, Clusters, Node Drivers (selected), Catalogs, Users, Settings, and Security. A search bar is located on the right side of the navigation bar. Below the navigation bar, the 'Node Drivers' section is titled, and there is an 'Add Node Driver' button. The main content area shows a list of node drivers with columns for State, Name, and a search bar. The drivers listed are:

State	Name
Inactive	Aliyun ECS http://machine-driver.oss-cn-shanghai.aliyuncs.com/aliyun/10.2/linux/amd64/docker-machine-driver-aliyunecs.tgz
Active	Amazon EC2 Built-In
Active	Azure Built-In
Active	DigitalOcean Built-In
Inactive	Exoscale Built-In
Inactive	OpenStack Built-In
Inactive	Open Telekom Cloud https://obs.otc.t-systems.com/dockermachinedriver/docker-machine-driver-otc
Inactive	Packet https://github.com/packethost/docker-machine-driver-packet/releases/download/v0.14/docker-machine-driver-packet_linux-amd64.zip
Inactive	RackSpace Built-In
Active	SoftLayer Built-In
Active	vSphere Built-In

#별첨2) Rancher2.0

Dev History 

- Persistent Volume Plugin 제공



The screenshot shows the 'Add Persistent Volume' form in the Rancher UI. The 'Name' field contains 'e.g. myvolume'. The 'Volume Plugin' dropdown menu is open, displaying a list of available plugins. The 'Capacity' field is set to '10' with a 'GiB' unit selector. A 'Cancel' button is visible at the bottom right of the plugin list.

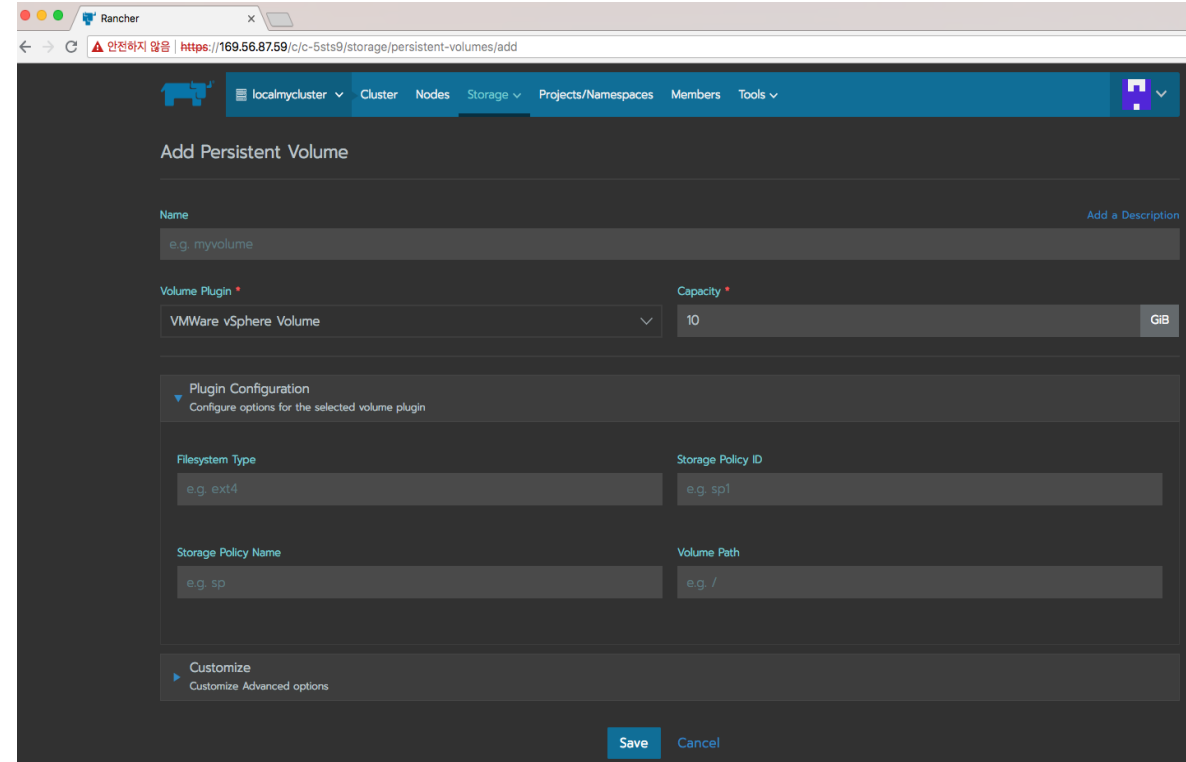
Add Persistent Volume

Name: e.g. myvolume Add a Description

Volume Plugin *
✓ Choose a volume plugin...
Amazon EBS Disk
Azure Disk
Azure Filesystem
Ceph Filesystem
Ceph RBD
Fibre Channel
Flex Volume
Flocker
Gluster Volume
Google Persistent Disk
iSCSI Target
Local Node Disk
Local Node Path
Longhorn
NFS Share
Openstack Cinder Volume
Photon Volume
Portworx Volume
Quobyte Volume
ScaleIO Volume
StorageOS
VMWare vSphere Volume

Capacity * 10 GiB

Cancel



The screenshot shows the 'Add Persistent Volume' form in the Rancher UI. The 'Name' field contains 'e.g. myvolume'. The 'Volume Plugin' dropdown menu is set to 'VMWare vSphere Volume'. The 'Capacity' field is set to '10' with a 'GiB' unit selector. The 'Plugin Configuration' section is expanded, showing fields for 'Filesystem Type' (e.g. ext4), 'Storage Policy ID' (e.g. sp1), 'Storage Policy Name' (e.g. sp), and 'Volume Path' (e.g. /). A 'Customize' section is also visible at the bottom. 'Save' and 'Cancel' buttons are at the bottom right.

Add Persistent Volume

Name: e.g. myvolume Add a Description

Volume Plugin * VMWare vSphere Volume Capacity * 10 GiB

Plugin Configuration
Configure options for the selected volume plugin

Filesystem Type: e.g. ext4 Storage Policy ID: e.g. sp1

Storage Policy Name: e.g. sp Volume Path: e.g. /

Customize
Customize Advanced options

Save Cancel