

Installed_CDP (on ec2)

1. Instance Config

유형	상세 내역
OS	RHEL-7.9_HVM
CDP_VERSION	7.1.4
INSTANCE	t2.xlarge 이상 권고 드립니다.
STORAGE	100 GB

1. AMI 선택2. 인스턴스 유형 선택3. 인스턴스 구성4. 스토리지 추가5. 태그 추가6. 보안 그룹 구성7. 검토

단계 1: Amazon Machine Image(AMI) 선택

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버, 애플리케이션)이 포함된 템플릿입니다. AWS에, 사용자 커뮤니티 또는 AWS Marketplace에서 제공하는 AMI를 선택하거나, 자체 AMI 중 하나를 선택할 수도 있습니다.

RHEL-7.9_HVM

System Manager

1 ~ 39 AMI

AMI 시작 (0)

나의 AMI (0)

AWS Marketplace (252)

카탈로그 AMI (3)

▼ 운영 체제

☐ Amazon Linux

☐ CentOS

☐ Debian

☐ Fedora

☐ Gentoo

☐ openSUSE

☐ 기타 Linux

☐ Red Hat

☐ SUSE Linux

☐ Ubuntu

☐ Windows

☐ macOS

▼ 아키텍처

☐ 32비트(x86)

☐ 64비트(x86)

☐ 64비트(ARM)

☐ 64비트(Mac)

▼ 부트 디바이스 유형

☐ EBS

☐ 인스턴스 스토어

RHEL-7.9_HVM_GA-20200917.x86_64-0-Hourly2-GP2 - ami-02640e11b63aaf0e5

Provided by Red Hat, Inc.

최소 디바이스 유형 x86_64가 필요함

EBS 볼륨의 예

64비트(x86)

선택

RHEL-7.9_HVM-20210208.x86_64-0-Hourly2-GP2 - ami-0c5e102894167eccc

Provided by Red Hat, Inc.

최소 디바이스 유형 x86_64가 필요함

EBS 볼륨의 예

64비트(x86)

선택

RHEL-7.9_HVM_BETA-20200422.x86_64-0-Hourly2-GP2 - ami-0f3167cb74b5d52f

Provided by Red Hat, Inc.

최소 디바이스 유형 x86_64가 필요함

EBS 볼륨의 예

64비트(x86)

선택

다른 카탈로그에서 "RHEL-7.9_HVM"에 대해 다음 결과를 찾음:

252개 결과(AWS Marketplace)

AWS Marketplace는 AMI에 대한 운영 체제, 사전 구성된 패키지 스택의 제공

1. AMI 선택2. 인스턴스 유형 선택3. 인스턴스 구성4. 스토리지 추가5. 태그 추가6. 보안 그룹 구성7. 검토

단계 4: 스토리지 추가

인스턴스가 다음 소프트웨어 디바이스 설정으로 시작됩니다. 추가 EBS 볼륨 및 인스턴스 스토어 볼륨을 인스턴스에 연결하거나 부트 볼륨의 설정을 편집할 수 있습니다. 인스턴스를 시작할 때 추가 EBS 볼륨을 연결할 수도 있지만, 인스턴스 스토어 볼륨을 연결할 수 없습니다. Amazon EC2의 스토리지 옵션에 대해 자세히 알아보십시오.

볼륨 유형 (1)

디바이스 (1)

스냅샷 (1)

크기 (GiB) (1)

볼륨 유형 (1)

IOPS (1)

저리량(MB/초) (1)

종료 시 삭제 (1)

암호화 (1)

루트

idev1da1

snap-0207bdc7c1a2da

100

기본 SSB(gp2)

300/3000

백만 사할 없음

☒

암호화되지 않음

새 볼륨 추가

표의 최대 사용 가능 크기는 최대 3850GB의 EBS 볼륨(SD) 또는 마그네틱 스토리지를 사용할 수 있습니다. 표의 최대 사용 가능 크기는 최대 3850GB의 EBS 볼륨(SD) 또는 마그네틱 스토리지를 사용할 수 있습니다. 표의 최대 사용 가능 크기는 최대 3850GB의 EBS 볼륨(SD) 또는 마그네틱 스토리지를 사용할 수 있습니다.

2. Package install

1. yum install

```
[ec2-user@ip-10-0-1-235 ~]$ sudo yum update -y
Loaded plugins: amazon-id, search-disabled-repos
rhui-client-config-server-7
Resolving Dependencies
--> Running transaction check
--> Package NetworkManager.x86_64 1:1.18.8-1.el7 will be updated
--> Package NetworkManager.x86_64 1:1.18.8-2.el7_9 will be an update
--> Package NetworkManager-config-server.noarch 1:1.18.8-1.el7 will be updated
--> Package NetworkManager-config-server.noarch 1:1.18.8-2.el7_9 will be an update
--> Package NetworkManager-libnm.x86_64 1:1.18.8-1.el7 will be updated
--> Package NetworkManager-libnm.x86_64 1:1.18.8-2.el7_9 will be an update
--> Package NetworkManager-team.x86_64 1:1.18.8-1.el7 will be updated
--> Package NetworkManager-team.x86_64 1:1.18.8-2.el7_9 will be an update
--> Package NetworkManager-tui.x86_64 1:1.18.8-1.el7 will be updated
--> Package NetworkManager-tui.x86_64 1:1.18.8-2.el7_9 will be an update
--> Package bind-export-libs.x86_64 32:9.11.4-26.P2.el7 will be updated
--> Package bind-export-libs.x86_64 32:9.11.4-26.P2.el7_9.5 will be an update
```

```
sudo yum update -y
sudo yum install wget ntp iptables-services vim -y
```

2. ntpd 활성화

```
[ec2-user@ip-10-0-1-235 ~]$ # ntpd 활성화
[ec2-user@ip-10-0-1-235 ~]$ sudo systemctl start ntpd
[ec2-user@ip-10-0-1-235 ~]$ sudo systemctl enable ntpd
Created symlink from /etc/systemd/system/multi-user.target.wants/ntpd.service to /usr/lib/systemd/system/ntpd.service.
[ec2-user@ip-10-0-1-235 ~]$
[ec2-user@ip-10-0-1-235 ~]$ # ntpd 상시 활성화
[ec2-user@ip-10-0-1-235 ~]$ sudo chkconfig ntpd on
Note: Forwarding request to 'systemctl enable ntpd.service'.
[ec2-user@ip-10-0-1-235 ~]$
[ec2-user@ip-10-0-1-235 ~]$ # ntpd 활성화 내역 확인
[ec2-user@ip-10-0-1-235 ~]$ sudo chkconfig ntpd
Note: Forwarding request to 'systemctl is-enabled ntpd.service'.
enabled
```

```
# ntpd 활성화
sudo systemctl start ntpd
sudo systemctl enable ntpd

# ntpd 상시 활성화
sudo chkconfig ntpd on

# ntpd 활성화 내역 확인
sudo chkconfig ntpd
```

3. ntpd를 활용한 rpm download & install

```
[ec2-user@ip-10-0-1-235 ~]$ wget https://rpmfind.net/linux/centos/7.9.2009/os/x86_64/Packages/libtirpc-devel-0.2.4-0.16.el7.x86_64.rpm
--2021-06-08 02:39:14-- https://rpmfind.net/linux/centos/7.9.2009/os/x86_64/Packages/libtirpc-devel-0.2.4-0.16.el7.x86_64.rpm
Resolving rpmfind.net (rpmfind.net)... 195.220.108.100
Connecting to rpmfind.net (rpmfind.net)|195.220.108.100|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 93208 (91K) [application/x-rpm]
Saving to: 'libtirpc-devel-0.2.4-0.16.el7.x86_64.rpm'

100%[=====] 93,208 260KB/s in 0.4s

2021-06-08 02:39:16 (260 KB/s) - 'libtirpc-devel-0.2.4-0.16.el7.x86_64.rpm' saved [93208/93208]

[ec2-user@ip-10-0-1-235 ~]$ sudo yum install libtirpc-devel-0.2.4-0.16.el7.x86_64.rpm -y
Loaded plugins: amazon-id, search-disabled-repos
Examining libtirpc-devel-0.2.4-0.16.el7.x86_64.rpm: libtirpc-devel-0.2.4-0.16.el7.x86_64
Marking libtirpc-devel-0.2.4-0.16.el7.x86_64.rpm to be installed
Resolving Dependencies
--> Running transaction check
--> Package libtirpc-devel.x86_64 0:0.2.4-0.16.el7 will be installed
--> Processing Dependency: libtirpc = 0.2.4-0.16.el7 for package: libtirpc-devel-0.2.4-0.16.el7.x86_64
--> Processing Dependency: libtirpc.so.1()(64bit) for package: libtirpc-devel-0.2.4-0.16.el7.x86_64
--> Running transaction check
--> Package libtirpc.x86_64 0:0.2.4-0.16.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
libtirpc-devel x86_64 0.2.4-0.16.el7 /libtirpc-devel-0.2.4-0.16.el7.x86_64 214 k
Installing for dependencies:
libtirpc x86_64 0.2.4-0.16.el7 rhel-7-server-rhui-rpms 89 k
Transaction Summary
=====
Install 1 Package (+1 Dependent package)
```

```
wget https://rpmfind.net/linux/centos/7.9.2009/os/x86_64/Packages/libtirpc-
devel-0.2.4-0.16.el7.x86_64.rpm
sudo yum install libtirpc-devel-0.2.4-0.16.el7.x86_64.rpm -y
```

4. selinux disable

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
#SELINUX=enforcing
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
~
```

vi 수정을 통한 설정 변경

```
sudo vi /etc/selinux/config
```

```
SELINUX=disabled
```

linux 명령어 수행을 통한 값 변경

```
sed -i 's/SELINUX= enforcing/SELINUX=disabled/' /etc/selinux/config
```

5. key-gen

```
[ec2-user@ip-10-0-1-235 ~]$ # 공개 키 생성
[ec2-user@ip-10-0-1-235 ~]$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
Generating public/private dsa key pair.
Your identification has been saved in /home/ec2-user/.ssh/id_dsa.
Your public key has been saved in /home/ec2-user/.ssh/id_dsa.pub.
The key fingerprint is:
SHA256:Ri2JsZE7pJMIHAB7umYUjHU3WDsfA+tDuXR9QRc9DBU ec2-user@ip-10-0-1-235.us-west-2.compute.internal
The key's randomart image is:
+---[DSA 1024]-----+
|*.o .oB. .o.*E.|
|++ ...o@ + o o.|
|o+.. +X.B o . .|
|oo ++o* + .|
|.. .+.S|
|.. o|
|.o|
|o|
+---[SHA256]-----+
[ec2-user@ip-10-0-1-235 ~]$ # 공개 키 내용 변경
[ec2-user@ip-10-0-1-235 ~]$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
[ec2-user@ip-10-0-1-235 ~]$ # 권한 변경
[ec2-user@ip-10-0-1-235 ~]$ chmod 0600 ~/.ssh/authorized_keys
[ec2-user@ip-10-0-1-235 ~]$ # test
[ec2-user@ip-10-0-1-235 ~]$ ssh localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:d5Y4cxzl6DwF0oU9js4kBl1pK09/eMbnJ+GgjtUxN4.
ECDSA key fingerprint is MD5:44:18:04:91:55:73:cf:30:f1:7c:84:49:99:a0:9c:e4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Last login: Tue Jun 8 02:11:46 2021 from 58.151.93.2
```

```
# 공개 키 생성
ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
# 공개 키 내용 변경
cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
# 권한 변경
chmod 0600 ~/.ssh/authorized_keys
# test
ssh localhost
```

(Option) [Optimize performance]

(6) ~ (8) 항목은 option 사항이므로 미 수행 하여도 설치에 문제없습니다.

[linux] vm.swappiness 설정

1. 개요

- swappiness
- vm.swappiness

스왑 활용도, 스와핑 활용도, 스와피니스
리눅스 커널 속성 중 하나
스왑메모리 활용 수준 조절
스왑 사용의 적극성 수준

2. 값 설명

값의 범위: 0 ~ 100 (기본값: 60)

값	설명
vm.swappiness = 0	스왑 사용안함[1]
vm.swappiness = 1	스왑 사용 최소화
vm.swappiness = 60	기본값
vm.swappiness = 100	적극적으로 스왑 사용

6. vm.swappiness option

```
[ec2-user@ip-10-0-1-235 ~]$ sysctl vm.swappiness
vm.swappiness = 30
[ec2-user@ip-10-0-1-235 ~]$ sudo sysctl vm.swappiness=1
vm.swappiness = 1
```

```
sysctl vm.swappiness
sudo sysctl vm.swappiness=1
```

```
[ec2-user@ip-10-0-1-235 ~]$ sudo su
[root@ip-10-0-1-235 ec2-user]# echo "vm.swappiness=1" >> /etc/sysctl.conf
[root@ip-10-0-1-235 ec2-user]# sudo cat /etc/sysctl.conf
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
vm.swappiness=1
```

```
sudo su
echo "vm.swappiness=1" >> /etc/sysctl.conf
cat /etc/sysctl.conf
```

7. /etc/rc.d/rc.local 수정

rc.local

부팅시 자동 실행 명령어 스크립트 수행

일반적으로 서버 부팅시마다 매번 자동 실행되길 원하는 명령어는 /etc/rc.d/rc.local에 넣어 주시면됩니다.

[THP\(Transparent Huge Pages\)](#)

```
[root@ip-10-0-1-235 ec2-user]# chmod +x /etc/rc.d/rc.local
[root@ip-10-0-1-235 ec2-user]# cat /etc/rc.d/rc.local
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local
[root@ip-10-0-1-235 ec2-user]#
[root@ip-10-0-1-235 ec2-user]# echo never > /sys/kernel/mm/transparent_hugepage/enabled
[root@ip-10-0-1-235 ec2-user]# echo never > /sys/kernel/mm/transparent_hugepage/defrag
```

```
chmod +x /etc/rc.d/rc.local
vi /etc/rc.d/rc.local

# 하단의 내용 추가
echo never > /sys/kernel/mm/transparent_hugepage/enabled
echo never > /sys/kernel/mm/transparent_hugepage/defrag
```

THP 옵션이 활성화 확인 방법

```
cat /sys/kernel/mm/transparent_hugepage/enabled
```

[always] madvise never -> 출력된 결과에 [always] 에 대괄호가 되어있으면 THP가 활성화 된 상태입니다.

always madvise [never] -> 출력된 결과에 [never] 에 대괄호가 되어있으면 THP가 비 활성화 된 상태입니다.

8. /etc/default/grub 수정

```
[root@ip-10-0-1-235 ec2-user]# cat /etc/default/grub
GRUB_TIMEOUT=1
GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="console=ttyS0,115200n8 console=tty0 net.ifnames=0 rd.blacklist=nouveau nvme_core.io_timeout=4294967295 crashkernel=auto"
GRUB_DISABLE_RECOVERY="true"
[root@ip-10-0-1-235 ec2-user]# echo "transparent_hugepage=never" >> /etc/default/grub
[root@ip-10-0-1-235 ec2-user]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-1160.25.1.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-1160.25.1.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-1160.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-1160.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-31c0a74479864f71b2aa3bf592405578
Found initrd image: /boot/initramfs-0-rescue-31c0a74479864f71b2aa3bf592405578.img
Found linux image: /boot/vmlinuz-0-rescue-f5129602b89a4e4e930a60eeab2058c0
Found initrd image: /boot/initramfs-0-rescue-f5129602b89a4e4e930a60eeab2058c0.img
done
[root@ip-10-0-1-235 ec2-user]#
[root@ip-10-0-1-235 ec2-user]# systemctl start tuned
[root@ip-10-0-1-235 ec2-user]# tuned-adm off
[root@ip-10-0-1-235 ec2-user]# tuned-adm list
Available profiles:
- balanced                - General non-specialized tuned profile
- desktop                 - Optimize for the desktop use-case
- hpc-compute             - Optimize for HPC compute workloads
- latency-performance     - Optimize for deterministic performance at the cost of increased power consumption
- network-latency         - Optimize for deterministic performance at the cost of increased power consumption, focused on low latency network performance
- network-throughput      - Optimize for streaming network throughput, generally only necessary on older CPUs or 40G+ networks
- powersave              - Optimize for low power consumption
- throughput-performance - Broadly applicable tuning that provides excellent performance across a variety of common server workloads
- virtual-guest           - Optimize for running inside a virtual guest
- virtual-host            - Optimize for running KVM guests
No current active profile.
[root@ip-10-0-1-235 ec2-user]# systemctl stop tuned
[root@ip-10-0-1-235 ec2-user]# systemctl disable tuned
Removed symlink /etc/systemd/system/multi-user.target.wants/tuned.service.
```

```
cat /etc/default/grub
```

```
echo "transparent_hugepage=never" >> /etc/default/grub
```

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

```
systemctl start tuned
```

```
tuned-adm off
```

```
tuned-adm list
```

```
systemctl stop tuned
```

```
systemctl disable tuned
```

9. cloudera-manager-install

1. installer download

```
wget https://archive.cloudera.com/cm7/7.1.4/cloudera-manager-installer.bin
```

```
sudo chmod u+x cloudera-manager-installer.bin
```

```
sudo ./cloudera-manager-installer.bin
```

2. install

next next next yes

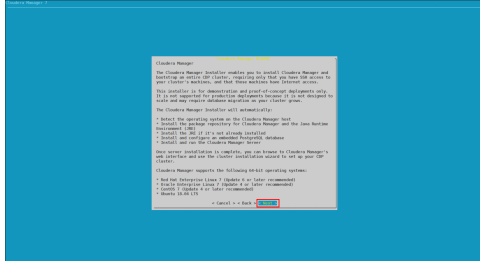
명령어 입력없이 화면에서 엔터만 반복적으로 입력시 자동으로 설치 진행

설치 완료 후 아래의 링크로 접속

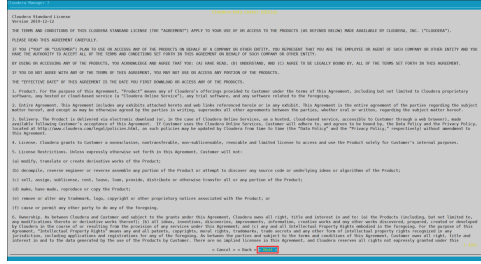
<https://publicdns:7180>

설치 이미지

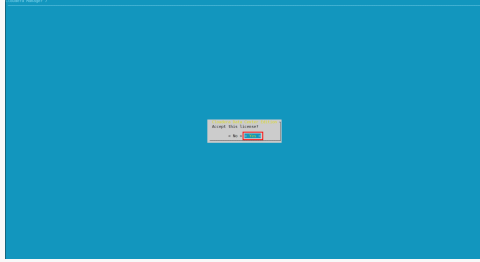
1



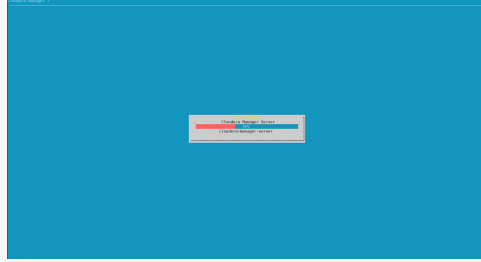
2



3



4



서버 부팅 시 자동실행 설정

```
sudo systemctl enable cloudera-scm-server.service
```

3. Multi node 설정

1. AMI 등록

- Name : `cdp_ami`
- Description : `Image for cdp cluster`
- Storage : `100`

- 이미지 생성 정보
이미지(AMI)라고 하는 EC2 인스턴스를 시작할 때 사용되는 프로그램 및 설정을 정의합니다. 기존 인스턴스의 구성에서 이미지를 생성할 수 있습니다.

인스턴스 ID
i-060883948ac60ac92 (cjm_cdp_manager)

이미지 이름

cdp_ami

최대 127자. 생성 후에는 수정할 수 없습니다.

이미지 설명 - 선택 사항

Image for cdp cluster

최대 255자

재부팅 안 함

☐ 활성화

인스턴스 볼륨

볼륨 유형

EBS

/dev/s...

볼륨에서 새 스냅샷 생성

100

볼륨 유형

EBS 전용 SSD - gp2

300

처리량

종료 시 삭제

☒ 활성화

암호화됨

☐ 활성화

볼륨 추가

이미지 생성 프로세스 중에 Amazon EC2는 위의 각 볼륨의 스냅샷을 생성합니다.

태그 - 선택 사항

태그는 사용자가 AWS 리소스에 할당하는 레이블입니다. 각 태그는 키와 값(선택 사항)으로 구성됩니다. 태그를 사용하여 리소스를 검색 및 필터링하거나 AWS 비용을 추적할 수 있습니다.

- ☒ 이미지와 스냅샷을 함께 태그 지정
이미지와 스냅샷에 동일한 태그를 지정합니다.

- ☐ 이미지와 스냅샷을 별도로 태그 지정
이미지와 스냅샷에 다른 태그를 지정합니다.

키

Owner

X

Name

X

태그 추가

태그를 48개 더 추가할 수 있습니다.

값 - 선택 사항

cjm

X

cdp_ami

X

태그 추가

2. Data node 생성

등록된 AMI로 3개의 Instance 생성

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 3: 인스턴스 세부 정보 구성

기본 VPC 없음. 다른 VPC 또는 새 기본 VPC 생성(를) 선택합니다.

요구 사항에 적합하게 인스턴스를 구성합니다. 동일한 AMI의 여러 인스턴스를 시작하고 스왑 인스턴스를 요청하여 보다 저렴한 요금을 활용하며 인스턴스에 액세스 관리 역할을 담당하는 등 다양한 기능을 사용할 수 있습니다.

인스턴스 개수 3 Auto Scaling 그룹 시작

애플리케이션 가용성을 유지 관리하고 향후 쉽게 확장할 수 있도록 이 인스턴스를 Auto Scaling 그룹으로 시작하는 것이 좋습니다. 애플리케이션의 상태를 정상으로...

구매 옵션 ☐ 스왑 인스턴스 요청

네트워크 vpc-02067642b982c2cf7 | cjm-vpc 새 VPC 생성
기본 VPC가 없습니다. 새 기본 VPC 생성.

서브넷 subnet-08f17a12a46396972 | cjm-subnet-public-a | 새 서브넷 생성
241개 IP 주소 사용 가능

퍼블릭 IP 자동 할당 서브넷 사용 설정(활성화)

배치 그룹 ☐ 배치 그룹에 인스턴스 추가

용량 예약 열기

도메인 조인 디렉터리 디렉터리 없음 새 디렉터리 생성

IAM 역할 없음 새 IAM 역할 생성

종료 방식 중지

최대 절전 중지 동작 ☐ 추가 종료 동작으로 최대 절전 모드를 활성화

종료 방지 기능 활성화 ☐ 우발적인 종료로부터 보호

모니터링 ☐ CloudWatch 세부 모니터링 활성화
추가 요금이 발생합니다.

태번서 공유됨 - 공유된 하드웨어 인스턴스 실행
전용 태번서에는 추가 요금이 적용됩니다.

Elastic Inference ☐ Elastic Inference 액셀러레이터 추가
추가 요금이 발생합니다.

크레딧 사양 ☐ 무제한
추가 요금이 적용될 수 있습니다

3. Cluster Network 설정

- 모든 node 설정

```
sudo vi /etc/hosts
```

```
# =====  
127.0.0.1    localhost localhost.localdomain localhost4  
localhost4.localhost4  
::1         localhost localhost.localdomain localhost6  
localhost6.localhost6  
  
10.0.1.235   cdp-manager  
10.0.1.22    cdp-node1  
10.0.1.228   cdp-node2  
10.0.1.112   cdp-node3
```

Cloudera 재시작

```
``sh  
sudo service cloudera-scm-server restart  
``
```

4. cloudera.manger 설정

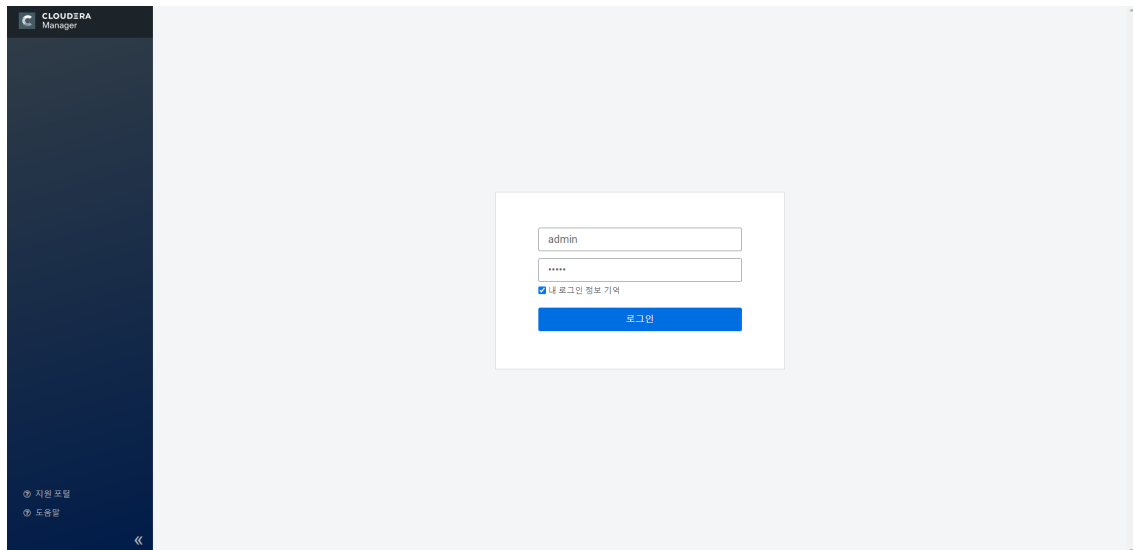
1. cloudera.manger web install

설치 완료 후 아래의 링크로 접속

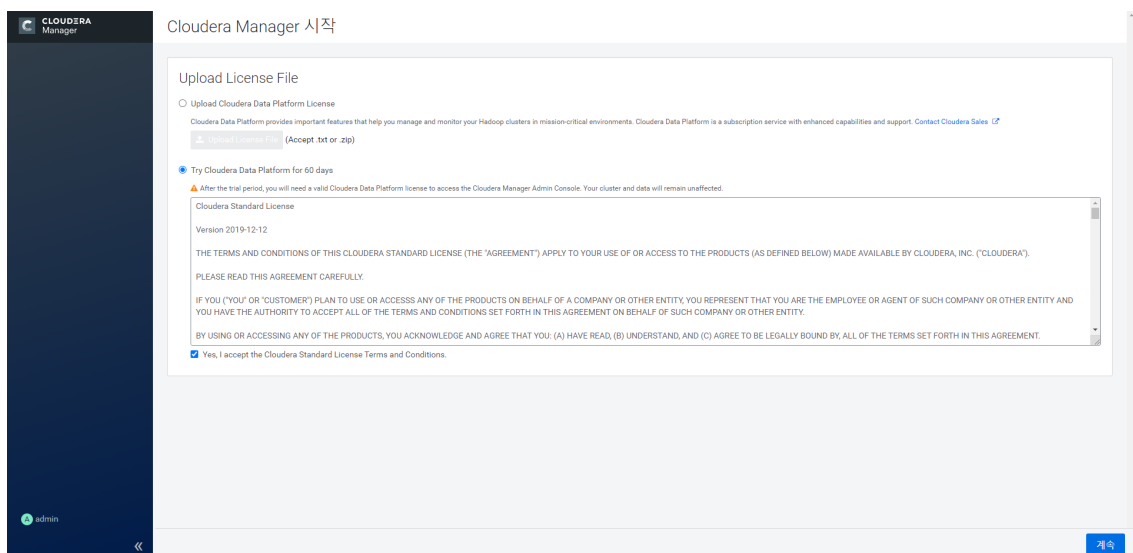
http://{manager_server_publicdns}:7180

ID : admin

PW : admin



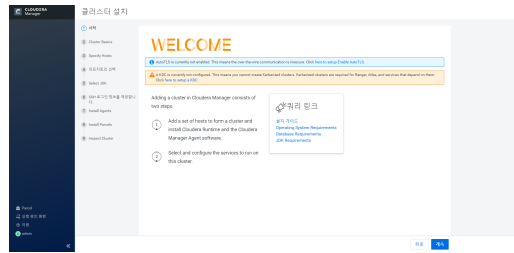
무료 라이선스 사용 60일



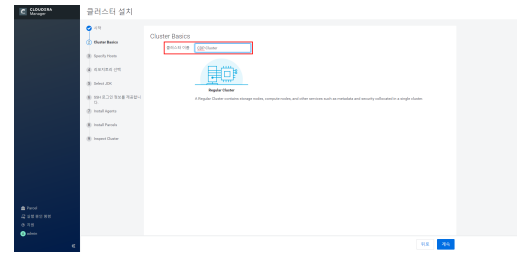
7-8 번 설치는 Issue가 있을 경우 1개 node 씩 수행

설치 이미지

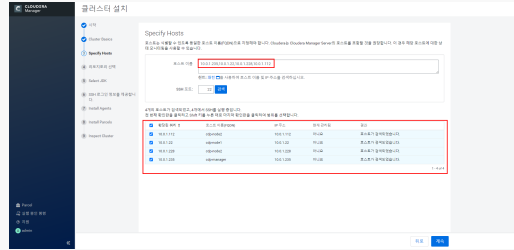
1



2



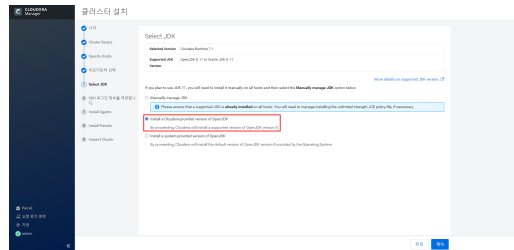
3



4



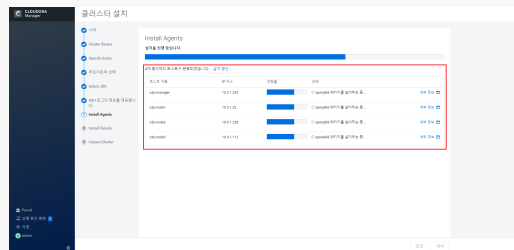
5



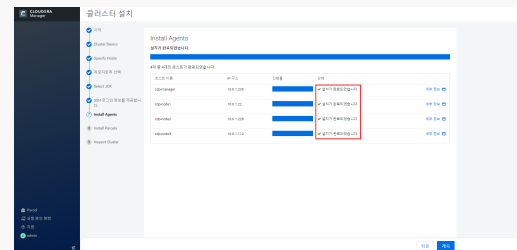
6



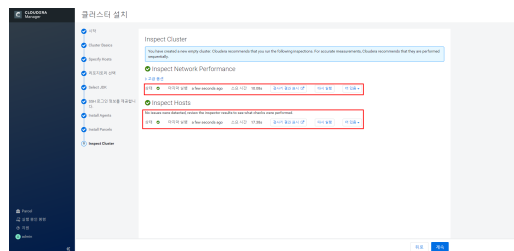
7



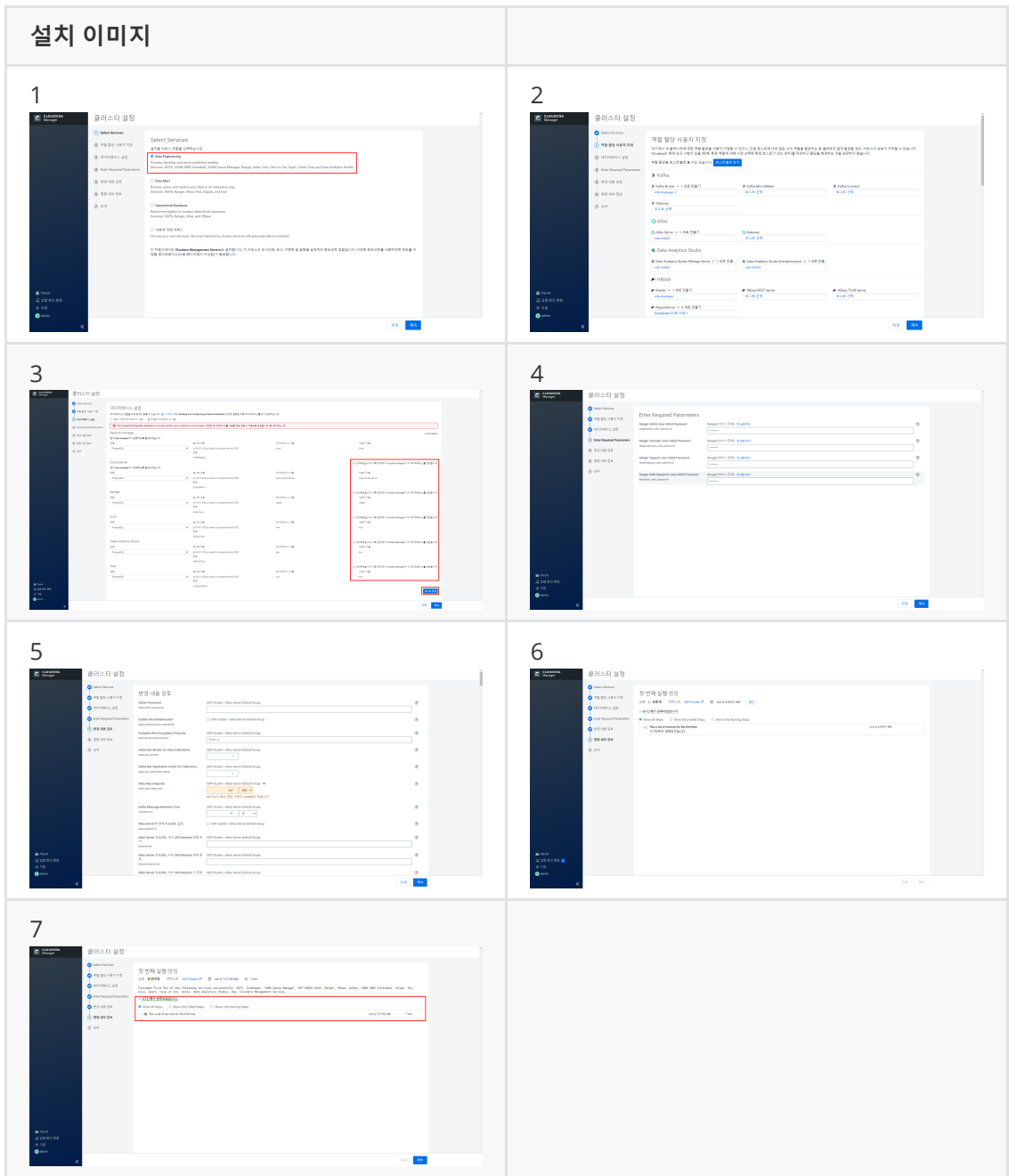
8



9



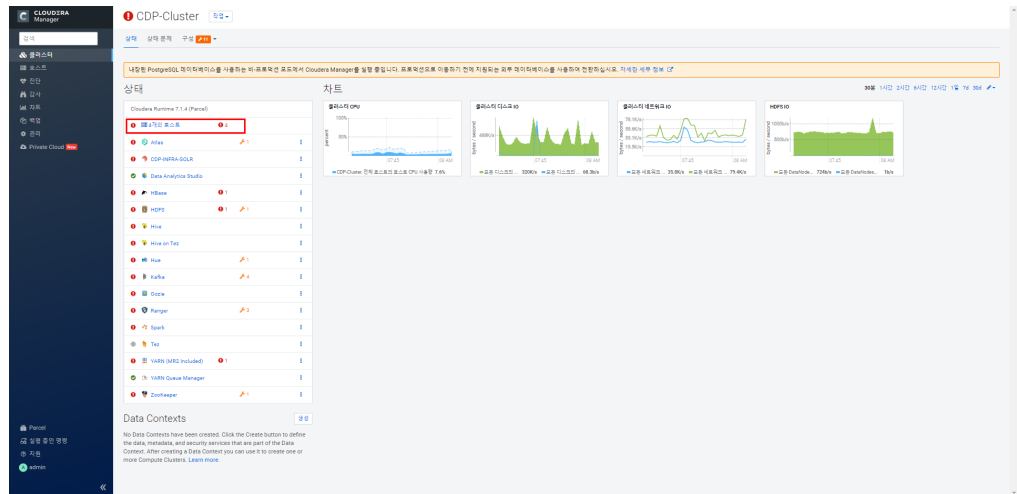
2. cloudera.manger web config
6-7 번 설치는 Issue가 있을 경우 재수행

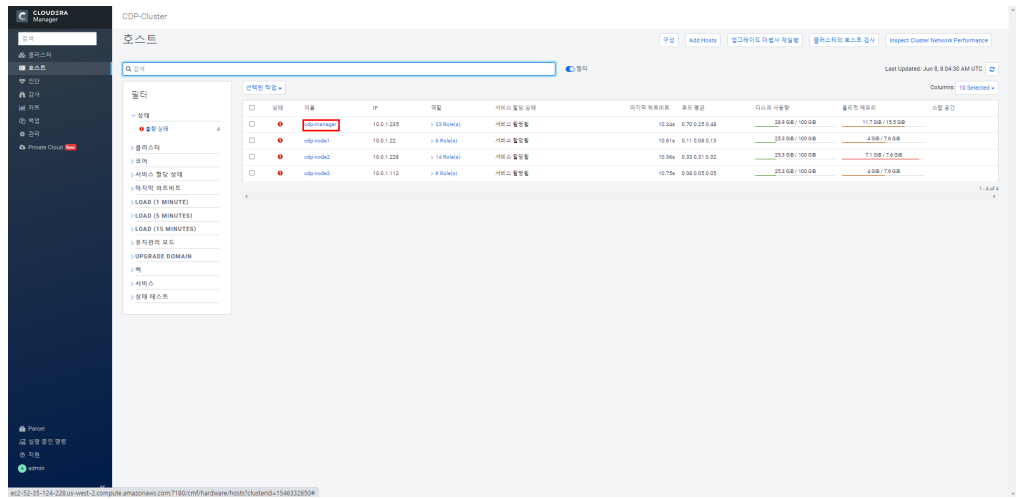


3. cloudera.manger web 설치 완료

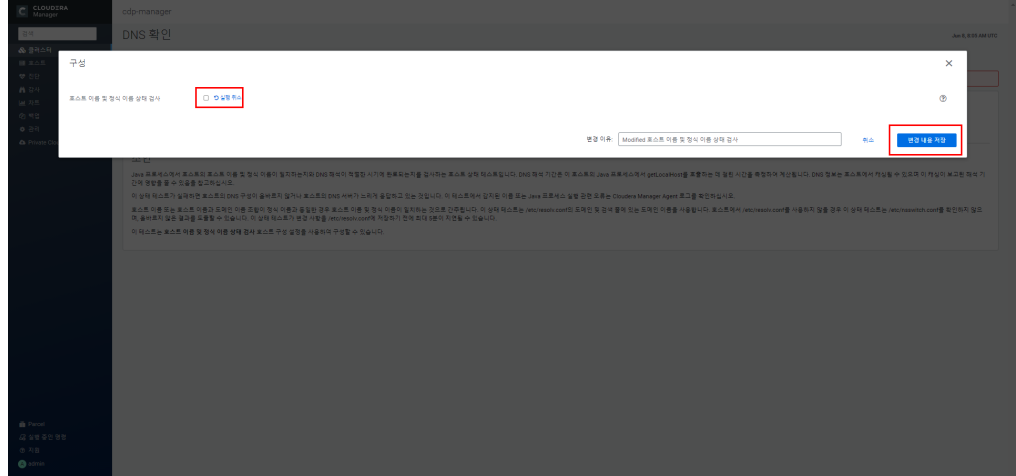
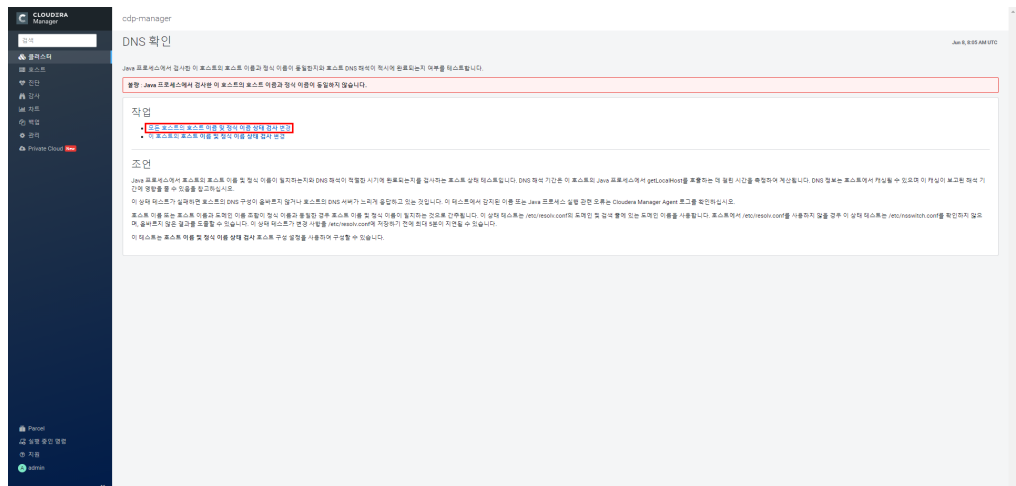
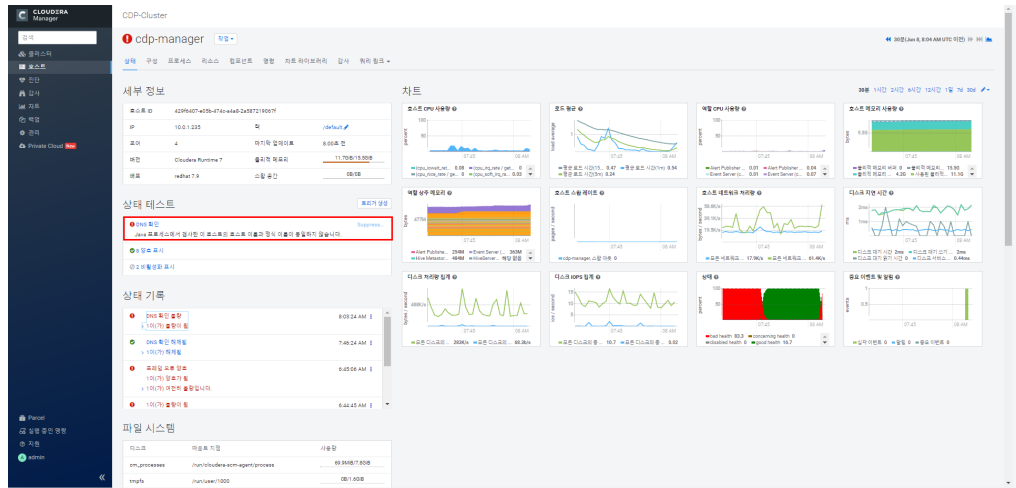
- 상태 체크 불량 시 해결 방법
해당 node들은 정식 DNS가 제공된 상태가 아니기 때문에 상태 검사 시 오류가 발생
DNS 상태 검사를 해제하면 해당 issue resolve 가능

1. 상태 불량 host 접근





2. DNS 상태 검사 해제



○ CDP-Cluster

