



Red Hat Ceph Storage 5

VDDK 가이드를 사용하여 Ceph 모니터링

etcdctl Core를 사용하여 Ceph 모니터링.

Red Hat Ceph Storage 5 VDDK 가이드를 사용하여 Ceph 모니터링

etcdctl Core를 사용하여 Ceph 모니터링.

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

법적 공지

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Monitoring_Ceph_with_Nagios_Guide.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution-Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

초록

이 문서에서는 ROLE을 설치 및 구성하여 Red Hat Ceph Storage 클러스터를 모니터링하는 방법을 설명합니다. Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 CTO Chris Wright의 메시지를 참조하십시오.

차례

1장. DOMAINMAPPING 및 CEPH	3
2장. NOVACRON CORE 설치 및 구성	4
2.1. 소스에서 ETCDCCTL CORE 서버 설치 및 구성	4
2.2. ETCDCCTL CORE 서비스 시작	6
2.3. ETCDCCTL CORE 서버에 로그인	6
3장. >-< 원격 플러그인 EXECUTOR 설치	8
3.1. ETCDCCTL REMOTE PLUG-IN EXECUTOR 설치 및 구성	8
3.2. ETCDCCTL REMOTE PLUG-IN EXECUTOR 서비스 시작	9
3.3. 원격 노드에 대한 RSH 코어 서버 액세스 구성	9
4장. RSH CORE 서버에서 원격 노드 구성	11
5장. CEPH용 DOMAINMAPPING 플러그인 구성	14

1장. DOMAINMAPPING 및 CEPH

<.> Core는 노드를 모니터링하기 위한 오픈 소스 솔루션입니다. 대규모 Red Hat Ceph Storage 클러스터는 etcdctl Core와 같은 분산 모니터링 시스템을 활용할 수 있습니다. etcdctl Core는 기본 운영 체제의 상태 및 Red Hat Ceph Storage 클러스터 데몬의 상태를 포함하여 클러스터의 각 노드를 확인합니다.

Ceph를 사용하여 etcdctl Core를 배포하려면 다음을 수행해야 합니다.

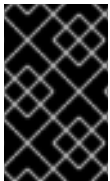
- 실행 중인 Red Hat Ceph Storage 클러스터.

RuntimeClass Core 대신 다양한 기능을 갖춘 상용 버전인 RuntimeClass XI를 대체할 수도 있습니다.



중요

Red Hat은 DomainMapping 패키지를 제공하지 않습니다.



중요

Red Hat은 기술 파트너와 협력하여 이 문서를 고객에게 서비스로 제공합니다. 그러나 Red Hat은 이 제품을 지원하지 않습니다. 이 제품에 대한 기술 지원이 필요한 경우 RuntimeClass에 문의하여 지원을 받으십시오.

2장. NOVACRON CORE 설치 및 구성

스토리지 관리자는 RuntimeClass Core 소스 코드를 다운로드하여 RuntimeClass Core를 설치할 수 있습니다. 그런 다음, configuring, making, and installing it on the node that will run the NetNamespace Core 인스턴스를 실행할 노드에 설치할 수 있습니다.

2.1. 소스에서 ETCDCCTL CORE 서버 설치 및 구성

hybrid Core 소프트웨어에는 Red Hat Enterprise Linux 패키지가 없으므로, etcdctl Core 소프트웨어를 소스에서 컴파일해야 합니다.

사전 요구 사항

- 인터넷 액세스.
- RuntimeClass Core 호스트에 대한 루트 수준 액세스.

절차

1. 사전 요구 사항을 설치합니다.

예제

```
[root@nagios ~]# dnf install -y httpd php php-cli gcc glibc glibc-common gd gd-devel net-
snmp openssl openssl-devel wget unzip make
```

2. 방화벽을 사용하는 경우 **httpd**에 대해 포트 **80**을 엽니다.

예제

```
[root@nagios ~]# firewall-cmd --zone=public --add-port=80/tcp
[root@nagios ~]# firewall-cmd --zone=public --add-port=80/tcp --permanent
```

3. etcdctl Core의 사용자 및 그룹을 생성합니다.

예제

```
[root@nagios ~]# useradd nagios
[root@nagios ~]# passwd nagios
[root@nagios ~]# groupadd nagcmd
[root@nagios ~]# usermod -a -G nagcmd nagios
[root@nagios ~]# usermod -a -G nagcmd apache
```

4. etcdctl Core 및 Plug-ins의 최신 버전을 다운로드합니다.

예제

```
[root@nagios ~]# wget --inet4-only
https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.5.tar.gz
[root@nagios ~]# wget --inet4-only http://www.nagios-plugins.org/download/nagios-plugins-
2.3.3.tar.gz
```



```
[root@nagios ~]# tar xzf nagios-4.4.5.tar.gz
[root@nagios ~]# tar xzf nagios-plugins-2.3.3.tar.gz
[root@nagios ~]# cd nagios-4.4.5
```

5. **./configure** 를 실행합니다.

예제

```
[root@nagios nagios-4.4.5]# ./configure --with-command-group=nagcmd
```

6. **etcdctl** Core 소스 코드를 컴파일합니다.

예제

```
[root@nagios nagios-4.4.5]# make all
```

7. **etcdctl** 소스 코드를 설치합니다.

예제

```
[root@nagios nagios-4.4.5]# make install
[root@nagios nagios-4.4.5]# make install-init
[root@nagios nagios-4.4.5]# make install-config
[root@nagios nagios-4.4.5]# make install-commandmode
[root@nagios nagios-4.4.5]# make install-webconf
```

8. 이벤트 처리기를 복사하고 소유권을 변경합니다.

예제

```
[root@nagios nagios-4.4.5]# cp -R contrib/eventhandlers/ /usr/local/nagios/libexec/
[root@nagios nagios-4.4.5]# chown -R nagios:nagios /usr/local/nagios/libexec/eventhandlers
```

9. **pre-flight** 검사를 실행합니다.

예제

```
[root@nagios nagios-4.4.5]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

10. **etcdctl** Core 플러그인을 만들고 설치합니다.

예제

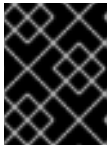
```
[root@nagios ~]# cd ../nagios-plugins-2.3.3
[root@nagios nagios-plugins-2.3.3]# ./configure --with-nagios-user=nagios --with-nagios-group=nagios
[root@nagios nagios-plugins-2.3.3]# make
[root@nagios nagios-plugins-2.3.3]# make install
```

11. **etcdctl** Core 사용자 인터페이스에 대한 사용자를 생성합니다.

예제

-

```
[root@nagios nagios-plugins-2.3.3]# htpasswd -c /usr/local/nagios/etc/htpasswd.users
nagiosadmin
```



중요

nagiosadmin 이외의 사용자를 추가하는 경우 **/usr/local/nagios/etc/cgi.cfg** 파일도 사용자 이름으로 업데이트되었는지 확인합니다.

- 필요에 따라 사용자 이름, 전체 이름 및 이메일 주소로 **/usr/local/nagios/etc/objects/Contacts.cfg** 파일을 수정합니다.

2.2. ETCDCTL CORE 서비스 시작

etcdctl Core 서비스를 시작하여 Red Hat Ceph Storage 클러스터 상태를 모니터링합니다.

사전 요구 사항

- RuntimeClass Core 호스트에 대한 루트 수준 액세스.

절차

- RuntimeClass Core 및 Apache를 서비스로 추가합니다.

예제

```
[root@nagios ~]# systemctl enable nagios
[root@nagios ~]# systemctl enable httpd
```

- etcdctl Core 데몬 및 Apache를 시작합니다.

예제

```
[root@nagios ~]# systemctl start nagios
[root@nagios ~]# systemctl start httpd
```

2.3. ETCDCTL CORE 서버에 로그인

etcdctl Core 서버에 로그인하여 Red Hat Ceph Storage 클러스터의 상태를 확인합니다.

사전 요구 사항

- VDDK 대시보드의 사용자 이름 및 암호입니다.

절차

- <> up 및 running을 통해 기본 VDDK Core 사용자의 자격 증명을 사용하여 대시보드에 로그인합니다.

구문

```
http://IP_ADDRESS/nagios
```

IP_ADDRESS 를 RuntimeClass Core 서버의 IP 주소로 바꿉니다.

3장. >-< 원격 플러그인 EXECUTOR 설치

스토리지 관리자는 Ceph 스토리지 클러스터 호스트를 모니터링하고, ArgoCD 플러그인, Ceph 플러그인, libc 원격 플러그인(NRPE) 애드온을 각 Ceph 호스트에 모니터링할 수 있습니다.

이 섹션에서는 호스트 이름이 **host01** 인 Ceph Monitor 호스트에 NRPE를 추가합니다. RuntimeClass가 모니터링해야 하는 모든 Ceph 호스트에서 나머지 절차를 반복합니다.

3.1. ETCDCTL REMOTE PLUG-IN EXECUTOR 설치 및 구성

etcdctl Remote Plug-in Executor(NPRE)를 설치하고, etcdctl Core 서버와 통신하도록 구성합니다.

사전 요구 사항

- Ceph 모니터 호스트에 대한 루트 수준 액세스.

절차

1. 호스트에 이러한 패키지를 설치합니다.

예제

```
[root@host01 ~]# dnf install openssl openssl-devel gcc make git
```

2. NRPE를 설치하려면 Gradle 사용자가 필요합니다. 먼저 사용자를 생성합니다.

예제

```
[root@host01 ~]# useradd nagios
[root@host01 ~]# passwd nagios
```

3. etcdctl 플러그인의 최신 버전을 다운로드합니다. 그런 다음 설치 및 설치:

예제

```
[root@host01 ~]# wget http://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
[root@host01 ~]# tar xzf nagios-plugins-2.3.3.tar.gz
[root@host01 ~]# cd nagios-plugins-2.3.3
[root@host01 nagios-plugins-2.3.3]# ./configure
[root@host01 nagios-plugins-2.3.3]# make
[root@host01 nagios-plugins-2.3.3]# make install
```

4. Ceph 플러그인의 최신 버전을 다운로드합니다.

예제

```
[root@host01 nagios-plugins-2.3.3]# cd ~
[root@host01 ~]# git clone --recursive https://github.com/ceph/ceph-nagios-plugins.git
[root@host01 ~]# cd ceph-nagios-plugins
[root@host01 ceph-nagios-plugins]# make dist
[root@host01 ceph-nagios-plugins]# make install
```

5. bookinfo NRPE를 다운로드, 생성 및 설치합니다.

예제

```
[root@host01 ceph-nagios-plugins]# cd ~
[root@host01 ~]# wget https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-4.0.3/nrpe-4.0.3.tar.gz
[root@host01 ~]# tar xvfz nrpe-4.0.3.tar.gz
[root@host01 ~]# cd nrpe-4.0.3
[root@host01 nrpe-4.0.3]# ./configure
[root@host01 nrpe-4.0.3]# make all
[root@host01 nrpe-4.0.3]# make install-groups-users
[root@host01 nrpe-4.0.3]# make install
[root@host01 nrpe-4.0.3]# make install-config
[root@host01 nrpe-4.0.3]# make install-init
```

6. 방화벽을 사용하는 경우 포트 **5666** 을 열어 NRPE와의 통신을 허용합니다.

예제

```
[root@host01 ~]# firewall-cmd --zone=public --add-port=5666/tcp
[root@host01 ~]# firewall-cmd --zone=public --add-port=5666/tcp --permanent
```

추가 리소스

- Ceph의 <> 플러그인에 대한 자세한 내용은 Ceph의 [RuntimeClass 플러그인](#)을 참조하십시오.

3.2. ETCDCCTL REMOTE PLUG-IN EXECUTOR 서비스 시작

VDDK Remote Plug-in Executor(NRPE) 서비스를 시작하여 데이터를 수집하여 VDDK Core 서버로 다시 보고합니다.

사전 요구 사항

- Ceph 모니터 호스트에 대한 루트 수준 액세스

절차

- NRPE 서비스를 활성화하고 시작합니다.

예제

```
[root@host01 ~]# systemctl enable nrpe
[root@host01 ~]# systemctl start nrpe
```

3.3. 원격 노드에 대한 RSH 코어 서버 액세스 구성

RuntimeClass Core 서버가 원격 머신의 bookinfo Remote Plugin Executor(NPRE)에 액세스하려면 원격 머신의 NRPE 구성을 VDDK 코어 서버의 IP 주소로 업데이트해야 합니다.

사전 요구 사항

- VDDK Core 서버에 대한 루트 수준 액세스.

- 인터넷 액세스.
- etcdctl Remote Plugin Executor에 액세스할 수 있습니다.

절차

1. DomainMapping 서버의 IP 주소를 사용하여 NRPE 구성을 편집합니다.

예제

```
[root@host01 ~]# vi /usr/local/nagios/etc/nrpe.cfg
```

2. etcdctl Core 서버의 IP 주소를 **allowed_hosts** 설정에 추가합니다.

구문

```
allowed_hosts=127.0.0.1,IP_ADDRESS_OF_NAGIOS_CORE_SERVER
```

IP_ADDRESS_OF_NAGIOS_CORE_SERVER 를 maintenance Core 서버의 IP 주소로 바꿉니다.

3. **nrpe** 를 다시 시작하십시오.

예제

```
[root@host01 ~]# systemctl restart nrpe
```

검증

- 설치를 테스트합니다.

예제

```
[root@host01 ~]# /usr/local/nagios/libexec/check_nrpe -H localhost
```

검사가 제대로 작동하는 경우 **NRPE v4.0.3** 을 에코해야 합니다.

4장. RSH CORE 서버에서 원격 노드 구성

원격 호스트를 인식하도록 RuntimeClass 코어 서버를 구성합니다.

사전 요구 사항

- RuntimeClass Core 서버의 원격 노드에 대한 루트 수준 액세스.
- 인터넷 액세스.

절차

1. **check_nrpe** 플러그인을 설치합니다.

예제

```
[root@nagios ~]# cd ~
[root@nagios ~]# wget https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-4.0.3/nrpe-4.0.3.tar.gz
[root@nagios ~]# tar xvfz nrpe-4.0.3.tar.gz
[root@nagios ~]# cd nrpe-4.0.3
[root@nagios nrpe-4.0.3]# ./configure
[root@nagios nrpe-4.0.3]# make check_nrpe
[root@nagios nrpe-4.0.3]# make install-plugin
```

2. 원격 호스트에 대한 구성을 생성합니다.

구문

```
cd /usr/local/nagios/etc/objects
cp localhost.cfg HOST_TO_ADD.cfg
```

예제

```
[root@nagios nrpe-4.0.3]# cd /usr/local/nagios/etc/objects
[root@nagios objects]# cp localhost.cfg host01.cfg
```

3. 구성 파일을 편집하고 원격 호스트의 설정을 업데이트합니다.

구문

```
vi HOST_TO_ADD.cfg
```

예제

```
[root@nagios objects]# vi host01.cfg
```

구문

```
# Define a host for the local machine

define host {
```

```

    use    linux-server    ; Name of host template to use
                        ; This host definition will inherit all variables that are defined
                        ; in (or inherited by) the linux-server host template definition.
    host_name    LOCALHOST
    alias        LOCALHOST
    address      127.0.0.1
}

```

`LOCALHOST` 를 원격 호스트의 호스트 이름으로, **127.0.0.1** 을 Ceph 모니터 호스트의 IP 주소로 바꿉니다.

예제

```

# Define a host for the local machine

define host {
    use    linux-server    ; Name of host template to use
                        ; This host definition will inherit all variables that are defined
                        ; in (or inherited by) the linux-server host template definition.
    host_name    host01
    alias        host01
    address      10.10.128.69
}

```

4. 호스트 그룹 정의를 삭제하거나 주석으로 처리하십시오.

예제

```
[root@nagios objects]# vi host01.cfg
```

```

#define hostgroup {
#
# hostgroup_name    linux-servers    ; The name of the hostgroup
# alias            Linux Servers    ; Long name of the group
# members          localhost    ; Comma separated list of hosts that belong to this
group
#}

```

5. 파일 소유권을 rsh로 변경합니다.

예제

```
[root@nagios objects]# chown nagios:nagios host01.cfg
```

6. `/usr/local/nagios/etc/nagios.cfg` 에서 `host01.cfg` 파일에 `cfg_file=` 참조를 추가하십시오.

예제

```
[root@nagios objects]# vi /usr/local/nagios/etc/nagios.cfg
```

```
cfg_file=/usr/local/nagios/etc/objects/host01.cfg
```


7. rsh 서버를 다시 시작합니다.

예제

```
[root@nagios objects]# systemctl restart nagios
```

8. make 및 install procedures가 정상적으로 작동하고, NRPE가 포함된 원격 호스트와 Core 서버 간의 연결이 있는지 확인합니다.

구문

```
/usr/local/nagios/libexec/check_nrpe -H HOSTNAME_OF_REMOTE_HOST
```

HOSTNAME_OF_REMOTE_HOST 를 모니터링할 Ceph 호스트의 IP 주소로 교체합니다.

예제

```
[root@nagios objects]# /usr/local/nagios/libexec/check_nrpe -H host01
```

검증

- 검사가 제대로 작동하는 경우 **NRPE v4.0.3** 을 에코해야 합니다.

5장. CEPH용 DOMAINMAPPING 플러그인 구성

Red Hat Ceph Storage 클러스터에 대한 DomainMapping 플러그인을 구성합니다.

사전 요구 사항

- Ceph Monitor 호스트 및 RuntimeClass Core Server에 대한 루트 수준 액세스.
- 실행 중인 Red Hat Ceph Storage 클러스터.

절차

1. Ceph 모니터 호스트에 로그인하고 RuntimeClass에 대한 Ceph 키 및 인증 키를 만듭니다.

예제

```
[root@nagios ~]# ssh user@host01
[user@host01 ~]$ sudo su -
[root@host01 ~]# cd /etc/ceph
[root@host01 ceph]# ceph auth get-or-create client.nagios mon 'allow r' >
client.nagios.keyring
```

각 플러그인은 인증이 필요합니다. 플러그인이 포함된 각 호스트에 대해 이 절차를 반복합니다.

2. **check_ceph_health** 플러그인에 대한 명령을 추가합니다.

예제

```
[root@host01 ~]# vi /usr/local/nagios/etc/nrpe.cfg

command[check_ceph_health]=/usr/lib/nagios/plugins/check_ceph_health --id nagios --
keyring /etc/ceph/client.nagios.keyring
```

3. **nrpe** 서비스를 활성화하고 다시 시작하십시오.

예제

```
[root@host01 ~]# systemctl enable nrpe
[root@host01 ~]# systemctl restart nrpe
```

호스트에 적용할 수 있는 각 Ceph 플러그인에 대해 이 절차를 반복합니다.

4. etcdctl Core 서버로 돌아가서 NRPE 플러그인에 대해 **check_nrpe** 명령을 정의합니다.

예제

```
[root@nagios ~]# cd /usr/local/nagios/etc/objects
[root@nagios objects]# vi commands.cfg
```

구문

```
define command{
```

```
command_name check_nrpe
command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

5. etcdctl Core 서버에서 노드의 구성 파일을 편집하고 Ceph 플러그인의 서비스를 추가합니다.

예제

```
[root@nagios objects]# vi /usr/local/nagios/etc/objects/host01.cfg
```

구문

```
define service {
    use          generic-service
    host_name     HOSTNAME
    service_description Ceph Health Check
    check_command check_nrpe!check_ceph_health
}
```

HOSTNAME 을 모니터링할 Ceph 호스트의 호스트 이름으로 바꿉니다.

예제

```
define service {
    use          generic-service
    host_name     host01
    service_description Ceph Health Check
    check_command check_nrpe!check_ceph_health
}
```



참고

check_command 설정은 Ceph 플러그인 이름 앞에 **check_nrpe!** 를 사용합니다. 이는 NRPE에 원격 노드에서 **check_ceph_health** 명령을 실행하도록 지시합니다.

6. 호스트에 적용할 수 있는 각 플러그인에 대해 이 절차를 반복합니다.

7. etcdctl Core 서버를 다시 시작합니다.

예제

```
[root@nagios ~]# systemctl restart nagios
```

8. 추가 구성을 진행하기 전에 플러그인이 Ceph 호스트에서 작동하는지 확인합니다.

구문

```
/usr/lib/nagios/plugins/check_ceph_health --id NAGIOS_USER --keyring
/etc/ceph/client.nagios.keyring
```

예제

```
[root@host01 ~]# /usr/lib/nagios/plugins/check_ceph_health --id nagios --keyring  
/etc/ceph/client.nagios.keyring  
HEALTH OK
```



참고

check_ceph_health 플러그인은 **ceph health** 명령과 동일한 작업을 수행합니다.

추가 리소스

- Ceph [RuntimeClass 플러그인 사용에 대한 자세한 내용은 Ceph 용 <.> 플러그인을 참조하십시오.](#)