



Red Hat OpenStack Platform 17.1

하이퍼컨버지드 인프라 배포

Red Hat OpenStack Platform 오버클라우드에서 Hyperconverged Infrastructure 이해
및 구성

Red Hat OpenStack Platform 17.1 하이퍼컨버지드 인프라 배포

Red Hat OpenStack Platform 오버클라우드에서 Hyperconverged Infrastructure 이해 및 구성

OpenStack Team
rhos-docs@redhat.com

법적 공지

Copyright © 2024 Red Hat, Inc.

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 동일한 호스트에 Compute 및 Ceph Storage 서비스를 공동 배치하는 하이퍼컨버지드의 Red Hat OpenStack Platform 구현에 대해 설명합니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체	4
RED HAT 문서에 관한 피드백 제공	5
1장. RED HAT OPENSTACK PLATFORM 하이퍼컨버지드 인프라 구성 및 배포	6
1.1. HYPERCONVERGED 인프라 개요	6
2장. HCI 하드웨어 배포	7
2.1. CEPH STORAGE 노드 디스크 정리	7
2.2. 노드 등록	7
2.3. 사용 가능한 RED HAT CEPH STORAGE 패키지 확인	10
2.4. HCI 환경의 소프트웨어 이미지 배포	10
2.5. HCI용 노드 지정	11
2.6. 다중 디스크 CEPH 클러스터의 루트 디스크 정의	14
3장. HCI용 RED HAT CEPH STORAGE 클러스터 구성	17
3.1. 배포 사전 요구 사항	17
3.2. OPENSTACK OVERCLOUD CEPH DEPLOY 명령	17
3.3. HCI의 CEPH 구성 덮어쓰기	17
3.4. 시간 동기화 구성	18
3.5. 최상위 도메인 접미사 구성	20
3.6. RED HAT CEPH STORAGE 클러스터 이름 구성	20
3.7. 네트워크 데이터 파일을 사용하여 네트워크 옵션 구성	21
3.8. 구성 파일을 사용하여 네트워크 옵션 구성	22
3.9. OSD의 CRUSH 계층 구조 구성	23
3.10. CEPH 서비스 배치 옵션 구성	24
3.11. CEPH 노드에 대한 SSH 사용자 옵션 구성	24
3.12. CEPH STORAGE 컨테이너에 액세스	26
4장. HCI용 RED HAT CEPH STORAGE 클러스터 사용자 지정	28
4.1. 구성 옵션	28
4.2. 서비스 사양 생성 (선택 사항)	28
4.3. RED HAT CEPH STORAGE를 사용하는 RED HAT OPENSTACK PLATFORM용 CEPH 컨테이너	29
4.4. 고급 OSD 사양 구성	29
4.5. 노드별 덮어쓰기에서 마이그레이션	30
4.6. CEPH ON-WIRE 암호화 활성화	30
5장. HCI용 스토리지 서비스 사용자 정의	31
5.1. HCI의 COMPUTE 서비스 리소스 구성	31
5.2. 사용자 정의 환경 파일 구성	31
5.3. CEPH 메타데이터 서버 활성화	32
5.4. CEPH OBJECT GATEWAY 오브젝트 스토리지	33
5.5. RED HAT OPENSTACK PLATFORM 오브젝트 스토리지에 대한 배포 옵션	33
5.6. CEPH를 사용하도록 블록 스토리지 백업 서비스 구성	34
5.7. CEPH 노드에 대해 여러 개의 결합된 인터페이스 구성	34
5.8. HCI용 오버클라우드 배포 시작	35
6장. HCI 구성 확인	37
6.1. HCI 구성 확인	37
7장. 하이퍼컨버지드 노드 확장	38
7.1. HCI 환경에서 하이퍼컨버지드 노드 확장	38
7.2. HCI 환경에서 하이퍼컨버지드 노드 축소	38

부록 A. 추가 정보	39
A.1. 구성 지침	39
A.2. 하이퍼컨버지드 인프라 환경 구성을 위한 가이드 및 리소스	40

보다 포괄적 수용을 위한 오픈 소스 용어 교체

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 메시지](#)를 참조하십시오.

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 어떻게 더 나은지 알려주십시오.

Jira에서 문서 피드백 제공

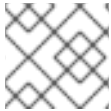
문제 생성 양식을 사용하여 문서에 대한 피드백을 제공합니다. Jira 문제는 Red Hat OpenStack Platform Jira 프로젝트에서 생성되어 피드백의 진행 상황을 추적할 수 있습니다.

1. Jira에 로그인했는지 확인합니다. Jira 계정이 없는 경우 피드백을 제출할 계정을 생성합니다.
2. 다음 링크를 클릭하여 **문제 생성** 페이지를 엽니다.
<https://issues.redhat.com/secure/CreateIssueDetails!init.jspx?pid=12336920&summary=Documentation%20feedback:%20%3CAdd%20summary%20here%3E&include=the+documentation+URL,+the%20chapter+or+section+number,+and+a+detailed+description>
3. **요약** 및 **설명** 필드를 작성합니다. **설명** 필드에 문서 URL, 장 또는 섹션 번호, 문제에 대한 자세한 설명을 포함합니다. 양식의 다른 필드를 수정하지 마십시오.
4. **생성**을 클릭합니다.

1장. RED HAT OPENSTACK PLATFORM 하이퍼컨버지드 인프라 구성 및 배포

1.1. HYPERCONVERGED 인프라 개요

RHOSP(Red Hat OpenStack Platform) HCI(하이퍼 컨버지드 인프라)는 하이퍼컨버지드 노드로 구성됩니다. RHOSP HCI에서 컴퓨팅 및 스토리지 서비스는 최적화된 리소스 사용을 위해 이러한 하이퍼컨버지드 노드에 배치됩니다. 하이퍼컨버지드 노드만 사용하거나 일반 컴퓨팅 및 Red Hat Ceph Storage 노드가 있는 하이퍼컨버지드 노드를 혼합하여 오버클라우드를 배포할 수 있습니다.



참고

Red Hat Ceph Storage를 스토리지 공급자로 사용해야 합니다.

작은 정보

BlueStore 메모리 처리 기능을 사용하려면 HCI 배포의 백엔드로 BlueStore를 사용합니다.

하이퍼컨버지드 인프라는 [Red Hat Ceph 및 OpenStack 배포에 설명된 배포 프로세스의 변형을 사용하여 director와 함께](#) 구축됩니다. 이 배포 시나리오에서 RHOSP director는 director가 오버클라우드 및 Red Hat Ceph Storage를 호출하는 클라우드 환경을 배포합니다. 오버클라우드 구성과는 별도로 Ceph 클러스터 자체를 관리하고 확장합니다.



중요

RHOSP HCI 환경에서는 인스턴스 HA가 지원되지 않습니다. RHOSP HCI 환경에서 인스턴스 HA를 사용하려면 인스턴스 HA를 사용하려면 Compute 노드 하위 집합을 ComputeInstanceHA 역할로 지정해야 합니다. Red Hat Ceph Storage 서비스는 인스턴스 HA를 호스팅하는 컴퓨팅 노드에서 호스팅해서는 안 됩니다.

Red Hat OpenStack Platform 17.1은 새 배포에 대해 Red Hat Ceph Storage 6만 지원합니다. Red Hat Ceph Storage 5는 새로운 배포 시나리오에서는 지원되지 않습니다.



중요

지원되는 Hyperconverged Infrastructure 환경의 모든 HCI 노드는 Red Hat OpenStack Platform 컨트롤러에서 사용하는 버전과 동일한 Red Hat Enterprise Linux 버전을 사용해야 합니다. 동일한 Hyperconverged Infrastructure 환경에서 HCI 노드의 하이브리드 상태에서 여러 Red Hat Enterprise 버전을 사용하려면 [Red Hat Customer Experience and Engagement](#) 팀에 문의하여 지원 예외에 대해 알아보십시오.

HCI 구성 지침은 [구성 지침](#)을 참조하십시오.

2장. HCI 하드웨어 배포

이 섹션에는 하이퍼컨버지드 노드의 준비 및 구성에 대한 절차 및 정보가 포함되어 있습니다.

사전 요구 사항

- *director와 함께 Red Hat Ceph 및 OpenStack 배포에서 오버클라우드 및 Red Hat Ceph Storage 배포를 읽었습니다.*

2.1. CEPH STORAGE 노드 디스크 정리

Ceph Storage OSD 및 저널 파티션에는 공장 정리 디스크가 필요합니다. Ceph OSD 서비스를 설치하기 전에 모든 데이터와 메타데이터는 해당 디스크에서 Bare Metal Provisioning 서비스(ironic)에서 지워야 합니다.

베어 메탈 프로비저닝 서비스를 사용하여 기본적으로 모든 디스크 데이터 및 메타데이터를 삭제하도록 director를 구성할 수 있습니다. 이 작업을 수행하도록 director가 구성되면 Bare Metal Provisioning 서비스는 노드를 **available**로 설정할 때마다 노드를 부팅하기 위해 추가 단계를 수행합니다.



주의

베어 메탈 프로비저닝 서비스는 **wipefs --force --all** 명령을 사용합니다. 이 명령은 디스크에서 모든 데이터와 메타데이터를 삭제하지만 보안 삭제는 수행하지 않습니다. 안전한 삭제는 훨씬 더 오래 걸립니다.

프로세스

1. **/home/stack/undercloud.conf**를 열고 다음 매개변수를 추가합니다.

```
clean_nodes=true
```

2. **/home/stack/undercloud.conf**를 저장합니다.
3. 언더클라우드 설정을 업데이트합니다.

```
openstack undercloud install
```

2.2. 노드 등록

director와의 통신을 활성화하려면 노드를 등록합니다.

프로세스

1. **/home/stack**에 노드 인벤토리 JSON 파일을 생성합니다.
2. 각 노드의 하드웨어 및 전원 관리 세부 정보를 입력합니다.
예를 들면 다음과 같습니다.

```

{
  "nodes":[
    {
      "mac":[
        "b1:b1:b1:b1:b1:b1"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.205"
    },
    {
      "mac":[
        "b2:b2:b2:b2:b2:b2"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.206"
    },
    {
      "mac":[
        "b3:b3:b3:b3:b3:b3"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.207"
    },
    {
      "mac":[
        "c1:c1:c1:c1:c1:c1"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.208"
    },
    {
      "mac":[

```

```

        "c2:c2:c2:c2:c2:c2"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "ipmi",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.209"
},
{
    "mac": [
        "c3:c3:c3:c3:c3:c3"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "ipmi",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.210"
},
{
    "mac": [
        "d1:d1:d1:d1:d1:d1"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "ipmi",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.211"
},
{
    "mac": [
        "d2:d2:d2:d2:d2:d2"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "ipmi",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.212"
},
{
    "mac": [
        "d3:d3:d3:d3:d3:d3"
    ],
    "cpu": "4",
    "memory": "6144",

```

```

        "disk": "40",
        "arch": "x86_64",
        "pm_type": "ipmi",
        "pm_user": "admin",
        "pm_password": "p@55w0rd!",
        "pm_addr": "192.0.2.213"
    }
}

```

3. 새 파일을 저장합니다.
4. stack 사용자를 초기화합니다.

```
$ source ~/stackrc
```

5. JSON 인벤토리 파일을 director로 가져오고 노드를 등록합니다.

```
$ openstack overcloud node import <inventory_file>
```

<inventory_file>을 첫 번째 단계에서 생성된 파일의 이름으로 바꿉니다.

6. 커널 및 램디스크 이미지를 각 노드에 할당합니다.

```
$ openstack overcloud node configure <node>
```

2.3. 사용 가능한 RED HAT CEPH STORAGE 패키지 확인

오버클라우드 배포가 실패하지 않도록 모든 필수 패키지를 사용할 수 있는지 확인합니다.

2.3.1. cephadm 패키지 설치 확인

cephadm 패키지가 하나 이상의 오버클라우드 노드에 설치되었는지 확인합니다. **cephadm** 패키지는 Ceph Storage 클러스터의 첫 번째 노드를 부트스트랩하는 데 사용됩니다.

cephadm 패키지는 **overcloud-hardened-uefi-full.qcow2** 이미지에 포함되어 있습니다. **tripleo_cephadm** 역할은 Ansible 패키지 모듈을 사용하여 이미지에 있는지 확인합니다.

2.4. HCI 환경의 소프트웨어 이미지 배포

HCI 환경에 구성된 노드는 **overcloud-hardened-uefi-full.qcow2** 소프트웨어 이미지를 사용해야 합니다. 이 소프트웨어 이미지를 사용하려면 RHOSP(Red Hat OpenStack Platform) 서브스크립션이 필요합니다.

프로세스

1. **/home/stack/templates/overcloud-baremetal-deploy.yaml** 파일을 엽니다.
2. **overcloud-hardened-uefi-full** 이미지가 필요한 노드의 이미지 속성을 추가하거나 업데이트합니다. 특정 노드에서 또는 특정 역할을 사용하는 모든 노드에 사용할 이미지를 설정할 수 있습니다.

특정 노드

```
- name: Ceph
```

```

count: 3
instances:
- hostname: overcloud-ceph-0
  name: node00
  image:
    href: file:///var/lib/ironic/images/overcloud-minimal.qcow2
- hostname: overcloud-ceph-1
  name: node01
  image:
    href: file:///var/lib/ironic/images/overcloud-hardened-uefi-full.qcow2
- hostname: overcloud-ceph-2
  name: node02
  image:
    href: file:///var/lib/ironic/images/overcloud-hardened-uefi-full.qcow2

```

특정 역할에 대해 구성된 모든 노드

```

- name: ComputeHCI
count: 3
defaults:
  image:
    href: file:///var/lib/ironic/images/overcloud-hardened-uefi-full.qcow2
instances:
- hostname: overcloud-ceph-0
  name: node00
- hostname: overcloud-ceph-1
  name: node01
- hostname: overcloud-ceph-2
  name: node02

```

3. **roles_data.yaml** 역할 정의 파일에서 **rhsm_enforce** 매개변수를 **False**로 설정합니다.

```
rhsm_enforce: False
```

4. 프로비저닝 명령을 실행합니다.

```

(undercloud)$ openstack overcloud node provision \
--stack overcloud \
--output /home/stack/templates/overcloud-baremetal-deployed.yaml \
/home/stack/templates/overcloud-baremetal-deploy.yaml

```

5. **overcloud-baremetal-deployed.yaml** 환경 파일을 **openstack overcloud deploy** 명령에 전달합니다.

2.5. HCI용 노드 지정

HCI에 대한 노드를 지정하려면 **ComputeHCI** 역할을 구성하기 위해 새 역할 파일을 생성하고 **ComputeHCI**의 리소스 클래스로 베어 메탈 노드를 구성해야 합니다.

프로세스

1. **stack** 사용자로 언더클라우드에 로그인합니다.
2. **stackrc** 인증 정보 파일을 소싱합니다.

```
[stack@director ~]$ source ~/stackrc
```

3. **Controller** 및 **ComputeHCI** 역할을 포함하는 **roles_data.yaml** 이라는 새 역할 데이터 파일을 생성합니다.

```
(undercloud)$ openstack overcloud roles generate Controller ComputeHCI -o  
~/roles_data.yaml
```

4. **roles_data.yaml** 을 열고 다음 매개변수 및 섹션이 있는지 확인합니다.

섹션/Parameter	현재의
역할 주석	역할: ComputeHCI
역할 이름	이름: ComputeHCI
description	HCI 역할
HostnameFormatDefault	%stackname%-novaceph-%index%
deprecated_nic_config_name	ceph.yaml

5. 노드 정의 템플릿 **node.json** 또는 **node.yaml** 에 추가하여 오버클라우드의 ComputeHCI 노드를 등록합니다.
6. 노드 하드웨어를 검사합니다.

```
(undercloud)$ openstack overcloud node introspect --all-manageable --provide
```

7. 사용자 지정 HCI 리소스 클래스를 사용하여 HCI를 지정할 각 베어 메탈 노드에 태그를 지정합니다.

```
(undercloud)$ openstack baremetal node set \  
--resource-class baremetal.HCI <node>
```

<node>를 베어 메탈 노드의 ID로 바꿉니다.

8. **ComputeHCI** 역할을 **/home/stack/templates/overcloud-baremetal-deploy.yaml** 파일에 추가하고 노드에 할당할 예측 노드 배치, 리소스 클래스 또는 기타 속성을 정의합니다.

```
- name: Controller  
  count: 3  
- name: ComputeHCI  
  count: 1  
  defaults:  
    resource_class: baremetal.HCI
```

9. **baremetal.yaml** 파일을 열고 HCI에 필요한 네트워크 구성이 포함되어 있는지 확인합니다. 다음은 구성 예제입니다.

```
- name: ComputeHCI
```

```
count: 3
hostname_format: compute-hci-%index%
defaults:
  profile: ComputeHCI
  network_config:
    template: /home/stack/templates/three-nics-vlans/compute-hci.j2
networks:
- network: ctlplane
  vif: true
- network: external
  subnet: external_subnet
- network: internalapi
  subnet: internal_api_subnet01
- network: storage
  subnet: storage_subnet01
- network: storage_mgmt
  subnet: storage_mgmt_subnet01
- network: tenant
  subnet: tenant_subnet01
```



참고

ComputeHCI 역할의 네트워크 구성에는 **storage_mgmt** 네트워크가 포함되어 있습니다. CephOSD 노드는 이 네트워크를 사용하여 중복 데이터 사본을 만듭니다. **Compute** 역할의 네트워크 구성에는 이 네트워크가 포함되어 있지 않습니다.

자세한 내용은 [베어 메탈 프로비저닝 서비스 구성](#) 을 참조하십시오.

10. 프로비저닝 명령을 실행합니다.

```
(undercloud)$ openstack overcloud node provision \
--stack overcloud \
--output /home/stack/templates/overcloud-baremetal-deployed.yaml \
/home/stack/templates/overcloud-baremetal-deploy.yaml
```

11. 별도의 터미널에서 프로비저닝 진행 상황을 모니터링합니다.

```
(undercloud)$ watch openstack baremetal node list
```



참고

watch 명령은 기본적으로 2초마다 갱신됩니다. **-n** 옵션은 갱신 타이머를 다른 값으로 설정합니다.

12. **감시** 프로세스를 중지하려면 **Ctrl-c** 를 입력합니다.
13. 확인: 프로비저닝이 성공하면 노드 상태가 **available** 에서 **active** 로 변경됩니다.

추가 리소스

- 노드 등록에 대한 자세한 내용은 *director 가이드를 사용하여 Red Hat OpenStack Platform 설치 및 관리*에서 [오버클라우드용 노드](#) 등록을 참조하십시오.

- 노드 하드웨어 검사에 대한 자세한 내용은 *director 가이드를 사용하여 Red Hat OpenStack Platform 설치 및 관리에서 베어 메탈 노드 하드웨어 인벤토리 생성* 을 참조하십시오.
- **ComputeHCI** 역할 및 **storage_mgmt** 네트워크의 네트워크 구성에 대한 자세한 내용은 *베어 메탈 프로비저닝 서비스 구성* 을 참조하십시오.

2.6. 다중 디스크 CEPH 클러스터의 루트 디스크 정의

Ceph Storage 노드는 일반적으로 여러 디스크를 사용합니다. director는 여러 디스크 구성에서 root 디스크를 식별해야 합니다. 오버클라우드 이미지는 프로비저닝 프로세스 중에 root 디스크에 작성됩니다.

하드웨어 속성은 루트 디스크를 식별하는 데 사용됩니다. 루트 디스크를 식별하는 데 사용할 수 있는 속성에 대한 자세한 내용은 루트 디스크 *를 식별하는 속성* 을 참조하십시오.

프로세스

1. 각 노드의 하드웨어 인트로스펙션에서 디스크 정보를 확인합니다.

```
(undercloud)$ openstack baremetal introspection data save <node_uuid> --file
<output_file_name>
```

- **<node_uuid>**를 노드의 UUID로 바꿉니다.
- **<output_file_name>**을 노드 인트로스펙션 출력이 포함된 파일 이름으로 바꿉니다.
예를 들어 노드 1개의 데이터에서 디스크 3개가 표시될 수 있습니다.

```
[
  {
    "size": 299439751168,
    "rotational": true,
    "vendor": "DELL",
    "name": "/dev/sda",
    "wwn_vendor_extension": "0x1ea4dcc412a9632b",
    "wwn_with_extension": "0x61866da04f3807001ea4dcc412a9632b",
    "model": "PERC H330 Mini",
    "wwn": "0x61866da04f380700",
    "serial": "61866da04f3807001ea4dcc412a9632b"
  },
  {
    "size": 299439751168,
    "rotational": true,
    "vendor": "DELL",
    "name": "/dev/sdb",
    "wwn_vendor_extension": "0x1ea4e13c12e36ad6",
    "wwn_with_extension": "0x61866da04f380d001ea4e13c12e36ad6",
    "model": "PERC H330 Mini",
    "wwn": "0x61866da04f380d00",
    "serial": "61866da04f380d001ea4e13c12e36ad6"
  },
  {
    "size": 299439751168,
    "rotational": true,
    "vendor": "DELL",
    "name": "/dev/sdc",
    "wwn_vendor_extension": "0x1ea4e31e121cfb45",
```

```

    "wwn_with_extension": "0x61866da04f37fc001ea4e31e121cfb45",
    "model": "PERC H330 Mini",
    "wwn": "0x61866da04f37fc00",
    "serial": "61866da04f37fc001ea4e31e121cfb45"
  }
]

```

2. 고유한 하드웨어 속성을 사용하여 노드의 root 디스크를 설정합니다.

```
(undercloud)$ openstack baremetal node set --property root_device='{<property_value>}'
<node-uuid>
```

- **<property_value>**를 루트 디스크를 설정하는 데 사용할 인트로스펙션 데이터의 고유한 하드웨어 속성 값으로 바꿉니다.
- **<node-uuid>**를 노드의 UUID로 바꿉니다.



참고

고유한 하드웨어 속성은 디스크를 고유하게 식별하는 하드웨어 인트로스펙션 단계의 모든 속성입니다. 예를 들어 다음 명령은 디스크 일련 번호를 사용하여 root 디스크를 설정합니다.

```
(undercloud)$ openstack baremetal node set --property
root_device='{ "serial": "61866da04f380d001ea4e13c12e36ad6" }'
1a4e30da-b6dc-499d-ba87-0bd8a3819bc0
```

3. 각 노드의 BIOS를 먼저 네트워크에서 부팅한 다음 root 디스크로 구성하십시오.

director가 root 디스크로 사용할 특정 디스크를 식별합니다. **openstack overcloud node provision** 명령을 실행하면 director가 오버클라우드 이미지를 프로비저닝하고 root 디스크에 씁니다.

2.6.1. 루트 디스크를 식별하는 속성

director가 root 디스크를 쉽게 식별할 수 있도록 다음과 같은 속성을 정의할 수 있습니다.

- **model**(문자열): 장치 식별자
- **vendor**(문자열): 장치 벤더
- **serial**(문자열): 디스크 일련번호
- **hctl**(문자열): Host:Channel:Target:Lun (SCSI 용)
- **size**(정수): 장치의 크기(GB 단위)
- **wwn**(문자열): 고유한 스토리지 식별자
- **wwn_with_extension**(문자열): 벤더 확장이 첨부된 고유한 스토리지 식별자
- **wwn_vendor_extension**(문자열): 고유한 벤더 스토리지 식별자
- **rotational**(부울): 회전 장치인 경우(HDD) True, 그렇지 않은 경우 false(SSD)
- **name**(문자열): 장치의 이름(예: /dev/sdb1)



중요

영구 이름이 있는 장치에 대해 **name** 속성을 사용합니다. 노드가 부팅될 때 값이 변경될 수 있으므로 **name** 속성을 사용하여 영구 이름이 없는 장치에 대해 루트 디스크를 설정하지 마십시오.

3장. HCI용 RED HAT CEPH STORAGE 클러스터 구성

이 장에서는 HCI 환경에 대해 Red Hat Ceph Storage 클러스터를 구성하고 배포하는 방법을 설명합니다.

3.1. 배포 사전 요구 사항

Red Hat Ceph Storage 클러스터를 구성하고 배포하기 전에 다음 작업이 수행되었는지 확인합니다.

- Bare Metal Provisioning 서비스(ironic)를 사용하여 베어 메탈 인스턴스 및 해당 네트워크를 프로비저닝합니다. 베어 메탈 인스턴스의 프로비저닝에 대한 자세한 내용은 [베어 메탈 프로비저닝 서비스 구성](#)을 참조하십시오.

3.2. OPENSTACK OVERCLOUD CEPH DEPLOY 명령

director를 사용하여 Ceph 클러스터를 배포하는 경우 **openstack overcloud ceph deploy** 명령을 사용해야 합니다. 명령 옵션 및 매개변수의 전체 목록은 [명령줄 인터페이스 참조](#)의 **openstack overcloud ceph deploy**를 참조하십시오.

openstack overcloud ceph deploy --help 명령은 현재 사용 가능한 옵션과 매개변수를 제공합니다.

3.3. HCI의 CEPH 구성 덮어쓰기

표준 형식 초기화 파일은 Ceph 클러스터 구성의 옵션입니다. 그런 다음 이 초기화 파일은 **cephadm bootstrap --config <file_name>** 또는 **openstack overcloud ceph deploy --config <file_name>** 명령을 사용하여 Ceph 클러스터를 구성하는 데 사용됩니다.

하이퍼컨버지드 노드에서 Ceph OSD 및 Compute 서비스를 공동 배치하면 Red Hat Ceph Storage와 Compute 서비스 간의 리소스 경합이 발생할 수 있습니다. 이는 서비스가 공동 배치를 인식하지 못하기 때문에 발생합니다. 리소스 경합으로 인해 서비스 성능이 저하될 수 있으므로 하이퍼컨버지드의 이점이 오히려 사라집니다.

리소스 할당은 리소스 경합을 관리하기 위해 초기화 파일을 사용하여 조정할 수 있습니다. 다음에서는 **initial-ceph.conf**라는 초기화 파일을 생성한 다음 **openstack overcloud ceph deploy** 명령을 사용하여 HCI 배포를 구성합니다.

```
$ cat <<EOF > initial-ceph.conf
[osd]
osd_memory_target_autotune = true
osd_numa_auto_affinity = true
[mgr]
mgr/cephadm/autotune_memory_target_ratio = 0.2
EOF
$ openstack overcloud ceph deploy --config initial-ceph.conf
```

OSD 때문에 osd_memory_target 구성 옵션을 기반으로 메모리 사용을 조정하도록 **osd_memory_target_autotune** 옵션이 **true**로 설정됩니다. **autotune_memory_target_ratio**의 기본값은 **Cryostat**입니다. 이는 시스템의 전체 RAM의 70%가 자동 조정되지 않은 Ceph 데몬에서 사용하는 메모리를 제거하는 시작점임을 나타냅니다. 그런 다음 나머지 메모리는 모든 OSD에 **osd_memory_target_autotune**이 **true**로 설정되어 있다고 가정합니다. HCI 배포의 경우 Compute 서비스에 더 많은 메모리를 사용할 수 있도록 **mgr/cephadm/autotune_memory_target_ratio**를 0.2로 설정합니다. **0.2** 값은 신중하게 시작 지점입니다. 배포 후 **ceph** 명령을 사용하여 필요한 경우 이 값을 변경합니다.

두 NUMA 노드 시스템은 하나의 NUMA 노드에서 대기 시간에 민감한 Nova 워크로드와 다른 NUMA 노드의 Ceph OSD 워크로드를 호스팅할 수 있습니다. 컴퓨팅 워크로드에서 사용하지 않는 특정 NUMA 노드를 사용하도록 Ceph OSD를 구성하려면 다음 Ceph OSD 구성 중 하나를 사용합니다.

- **osd_numa_node** 는 유사성을 numa 노드로 설정
- **osd_numa_auto_affinity** 는 스토리지 및 네트워크가 일치하는 NUMA 노드에 유사성을 자동으로 설정합니다.

NUMA 노드와 디스크 컨트롤러 모두에 네트워크 인터페이스가 있고 디스크 컨트롤러가 NUMA 노드 0인 경우 스토리지 네트워크에 대해 NUMA 노드 0의 네트워크 인터페이스를 사용하고 NUMA 노드 0에서 Ceph OSD 워크로드를 호스팅합니다. NUMA 노드 1에서 Nova 워크로드를 호스팅하고 NUMA 노드 1에서 네트워크 인터페이스를 사용하도록 합니다. 이 구성을 위해 **osd_numa_auto_affinity** 를 **true** 로 설정합니다. 또는 **osd_numa_node** 를 직접 **0** 으로 설정할 수 있으며 기본값은 **false** 로 설정되도록 **osd_numa_auto_affinity** 에 대해 값을 설정하지 않습니다.

OSD가 오프라인 상태가 되면 하이퍼컨버지드 클러스터 백필이 발생하면 백필 프로세스가 느려질 수 있습니다. 느린 복구를 위해 백필 작업은 배치된 컴퓨팅 워크로드에 미치는 영향이 줄어듭니다. Red Hat Ceph Storage에는 백필 활동 속도를 제어하는 다음과 같은 기본값이 있습니다.

- **osd_recovery_op_priority = 3**
- **osd_max_backfills = 1**
- **osd_recovery_max_active_hdd = 3**
- **osd_recovery_max_active_ssd = 10**



참고

이러한 기본값을 기본값이므로 초기화 파일에 전달할 필요가 없습니다. initial 구성에 기본값 이외의 값이 필요한 경우 배포 전에 필요한 값을 사용하여 초기화 파일에 추가합니다. 배포 후 'ceph config set osd' 명령을 사용합니다.

3.4. 시간 동기화 구성

시간 동기화 서비스(chrony)는 기본적으로 시간 동기화에 사용됩니다. 다음 작업을 수행하여 서비스를 구성할 수 있습니다.

- [구분된 목록을 사용하여 시간 동기화 구성](#)
- [환경 파일과의 시간 동기화 구성](#)
- [시간 동기화 비활성화](#)



참고

시간 동기화는 구분된 목록 또는 환경 파일을 사용하여 구성됩니다. 관리 관행에 가장 적합한 절차를 사용하십시오.

3.4.1. 구분된 목록을 사용하여 시간 동기화 구성

구분된 목록을 사용하여 NTP 서버를 구성하도록 시간 동기화 서비스 (chrony)를 구성할 수 있습니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.
2. 구분된 목록을 사용하여 NTP 서버를 구성합니다.

```
openstack overcloud ceph deploy \
  --ntp-server "<ntp_server_list>"
```

<ntp_server_list>를 쉼표로 구분된 서버 목록으로 바꿉니다.

```
openstack overcloud ceph deploy \
  --ntp-server "0.pool.ntp.org,1.pool.ntp.org"
```

3.4.2. 환경 파일과의 시간 동기화 구성

NTP 서버를 정의하는 환경 파일을 사용하도록 시간 동기화 서비스 (chrony)를 구성할 수 있습니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.
2. NTP 서버 구성을 포함하도록 **/home/stack/templates/ntp-parameters.yaml** 과 같은 환경 파일을 생성합니다.
3. **NtpServer** 매개변수를 추가합니다. **NtpServer** 매개변수에는 쉼표로 구분된 NTP 서버 목록이 포함되어 있습니다.

```
parameter_defaults:
  NtpServer: 0.pool.ntp.org,1.pool.ntp.org
```

4. 환경 파일을 사용하여 NTP 서버를 구성합니다.

```
openstack overcloud ceph deploy \
  --ntp-heat-env-file "<ntp_file_name>"
```

<ntp_file_name>을 생성한 환경 파일의 이름으로 바꿉니다.

```
openstack overcloud ceph deploy \
  --ntp-heat-env-file "/home/stack/templates/ntp-parameters.yaml"
```

3.4.3. 시간 동기화 비활성화

시간 동기화 서비스(chrony)는 기본적으로 활성화되어 있습니다. 사용하지 않으려면 서비스를 비활성화할 수 있습니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.
2. 시간 동기화 서비스(chrony)를 비활성화합니다.

```
openstack overcloud ceph deploy \
  --skip-ntp
```

3.5. 최상위 도메인 접미사 구성

최상위 도메인(TLD) 접미사를 구성할 수 있습니다. 이 접미사는 짧은 호스트 이름에 추가되어 오버클라우드 노드에 대해 정규화된 도메인 이름을 생성합니다.



참고

TLS-e 구성에 정규화된 도메인 이름이 필요합니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.
2. 최상위 도메인 접미사를 구성합니다.

```
openstack overcloud ceph deploy \
  --tld "<domain_name>"
```

& lt;domain_name& gt;을 필수 도메인 이름으로 바꿉니다.

```
openstack overcloud ceph deploy \
  --tld "example.local"
```

3.6. RED HAT CEPH STORAGE 클러스터 이름 구성

구성하는 이름으로 Red Hat Ceph Storage 클러스터를 배포할 수 있습니다. 기본 이름은 **ceph** 입니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.
2. 다음 명령을 사용하여 Ceph Storage 클러스터의 이름을 구성합니다.
OpenStack 오버클라우드 ceph deploy \ --cluster <cluster_name>

**\$ OpenStack 오버클라우드 ceph deploy \ --cluster central **



참고

현재 키 링 파일이 생성되지 않습니다. 인증 키는 오버클라우드 배포 중에 생성됩니다. 인증 키 파일은 이 절차 중에 구성된 클러스터 이름을 상속합니다. 오버클라우드 배포에 대한 자세한 내용은 다음을 참조하십시오. [5.8절. "HCI용 오버클라우드 배포 시작"](#)

위의 예에서 Ceph 클러스터의 이름은 **central** 입니다. 중앙 Ceph 클러스터의 구성 및 인증 키는 배포 프로세스 중에 **/etc/ceph** 에 생성됩니다.

```
[root@oc0-controller-0 ~]# ls -l /etc/ceph/
total 16
-rw-----. 1 root root 63 Mar 26 21:49 central.client.admin.keyring
-rw-----. 1 167 167 201 Mar 26 22:17 central.client.openstack.keyring
-rw-----. 1 167 167 134 Mar 26 22:17 central.client.radosgw.keyring
-rw-r--r--. 1 root root 177 Mar 26 21:49 central.conf
```

문제 해결

Ceph Storage 클러스터에 대한 사용자 정의 이름을 구성하면 다음 오류가 표시될 수 있습니다.

monclient: get_monmap_and_config에서 연결할 모니터를 식별할 수 없기 때문에

이 오류가 표시되면 Ceph 배포 후 다음 명령을 사용하십시오.

cephadm shell --config <configuration_file> --keyring <keyring_file>

예를 들어 클러스터 이름을 **central** 으로 구성할 때 이 오류가 표시되면 다음 명령을 사용합니다.

```
cephadm shell --config /etc/ceph/central.conf \
--keyring /etc/ceph/central.client.admin.keyring
```

다음 명령을 대안으로 사용할 수도 있습니다.

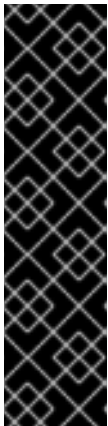
```
cephadm shell --mount /etc/ceph:/etc/ceph
export CEPH_ARGS='--cluster central'
```

3.7. 네트워크 데이터 파일을 사용하여 네트워크 옵션 구성

네트워크 데이터 파일은 Red Hat Ceph Storage 클러스터에서 사용하는 네트워크를 설명합니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.
2. **network_data.yaml** 이라는 사용자 지정 네트워크 속성을 정의하는 YAML 형식 파일을 생성합니다.



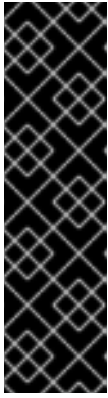
중요

네트워크 격리를 사용하면 표준 네트워크 배포는 두 개의 Ceph 네트워크에 매핑되는 두 개의 스토리지 네트워크로 구성됩니다.

- 스토리지 네트워크인 **storage** 는 Ceph 네트워크 **public_network** 에 매핑됩니다. 이 네트워크는 컴퓨팅 노드에서 Ceph 클러스터로 RBD 트래픽과 같은 스토리지 트래픽을 처리합니다.
- 스토리지 네트워크 **storage_mgmt** 는 Ceph 네트워크 **cluster_network** 에 매핑됩니다. 이 네트워크는 Ceph OSD 간의 데이터 복제와 같은 스토리지 관리 트래픽을 처리합니다.

3. **openstack overcloud ceph deploy** 명령을 **--crush-hierarchy** 옵션과 함께 사용하여 구성을 배포합니다.

```
openstack overcloud ceph deploy \
  deployed_metal.yaml \
  -o deployed_ceph.yaml \
  --network-data network_data.yaml
```



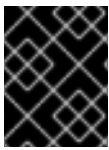
중요

openstack overcloud ceph deploy 명령은 **--network-data** 옵션으로 지정된 네트워크 데이터 파일을 사용하여 **public_network** 및 **cluster_network** 로 사용할 네트워크를 결정합니다. 이 명령은 **--public-network-name** 및 **--cluster-network-name** 옵션으로 다른 이름을 지정하지 않는 한 네트워크 데이터 파일에서 이러한 네트워크의 이름이 **storage** 및 **storage_mgmt** 라고 가정합니다.

네트워크 격리로 배포할 때 **--network-data** 옵션을 사용해야 합니다. 이 옵션을 사용하지 않는 경우 기본 언더클라우드(192.168.24.0/24)는 **public_network** 및 **cluster_network** 모두에 사용됩니다.

3.8. 구성 파일을 사용하여 네트워크 옵션 구성

네트워크 옵션은 네트워크 데이터 파일의 대안으로 구성 파일을 사용하여 지정할 수 있습니다.



중요

이 방법을 사용하여 네트워크 옵션을 구성하면 **network_data.yaml** 에서 자동으로 생성된 값을 덮어씁니다. 이 네트워크 구성 방법을 사용할 때 4개의 값을 모두 설정해야 합니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.
2. 표준 형식 초기화 파일을 생성하여 Ceph 클러스터를 구성합니다. 다른 구성 옵션을 포함하도록 파일을 이미 생성한 경우 네트워크 구성을 추가할 수 있습니다.
3. 파일의 **[global]** 섹션에 다음 매개변수를 추가합니다.

- **public_network**
- **cluster_network**
- **ms_bind_ipv4**



중요

public_network 및 **cluster_network** 가 **storage** 및 **storage_mgmt** 와 동일한 네트워크에 매핑되는지 확인합니다.

다음은 여러 서브넷 및 사용자 지정 네트워킹 이름이 있는 네트워크 구성에 대한 구성 파일 항목의 예입니다.

```
[global]
public_network = 172.16.14.0/24,172.16.15.0/24
cluster_network = 172.16.12.0/24,172.16.13.0/24
ms_bind_ipv4 = True
ms_bind_ipv6 = False
```

4. 구성 파일을 배포하려면 **openstack overcloud ceph deploy** 명령을 **--config** 옵션과 함께 사용합니다.

```
$ openstack overcloud ceph deploy \
--config initial-ceph.conf --network-data network_data.yaml
```

3.9. OSD의 CRUSH 계층 구조 구성

OSD 위치 속성을 Ceph Storage 클러스터 호스트 사양에 추가하도록 OSD 위치 속성을 OSD 배포 중에 CRUSH(Custom Controlled Replication Under Scalable Hashing) 계층 구조를 구성할 수 있습니다.

location 속성은 OSD가 CRUSH 계층 구조 내에 배치되는 위치를 구성합니다.



참고

location 속성은 초기 CRUSH 위치만 설정합니다. 속성의 후속 변경 사항은 무시됩니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.
2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.
\$ 소스 ~/stackrc
3. 구성 파일을 생성하여 사용자 지정 CRUSH 계층 구조(예: **crush_hierarchy.yaml**)를 정의합니다.
4. 파일에 다음 구성을 추가합니다.

```
<osd_host>:
  root: default
  rack: <rack_num>
<osd_host>:
  root: default
  rack: <rack_num>
<osd_host>:
  root: default
  rack: <rack_num>
```

- <osd_host>를 OSD가 배포된 노드의 호스트 이름(예: **ceph-0**)으로 바꿉니다.
 - <rack_num>을 OSD가 배포된 랙 수(예: **r0**)로 바꿉니다.
5. 사용자 지정 OSD 레이아웃을 사용하여 Ceph 클러스터를 배포합니다.

```
openstack overcloud ceph deploy \
  deployed_metal.yaml \
  -o deployed_ceph.yaml \
  --osd-spec osd_spec.yaml \
  --crush-hierarchy crush_hierarchy.yaml
```

Ceph 클러스터는 사용자 지정 OSD 레이아웃을 사용하여 생성됩니다.

위의 예제 파일은 다음과 같은 OSD 레이아웃을 생성합니다.

ID	CLASS	WEIGHT	TYPE NAME	STATUS	REWEIGHT	PRI-AFF
-1	0.02939	root default				
-3	0.00980	rack r0				
-2	0.00980	host ceph-node-00				

```

0 hdd 0.00980      osd.0          up 1.00000 1.00000
-5      0.00980      rack r1
-4      0.00980      host ceph-node-01
1 hdd 0.00980      osd.1          up 1.00000 1.00000
-7      0.00980      rack r2
-6      0.00980      host ceph-node-02
2 hdd 0.00980      osd.2          up 1.00000 1.00000

```



참고

장치 클래스는 Ceph에서 자동으로 감지되지만 CRUSH 규칙은 풀과 연결됩니다. 풀은 오버클라우드 배포 중에 **CephCrushRules** 매개변수를 사용하여 정의 및 생성됩니다.

추가 리소스

자세한 내용은 [Red Hat Ceph Storage 설치 가이드의 Red Hat Ceph Storage 워크로드 고려 사항](#)을 참조하십시오.

3.10. CEPH 서비스 배치 옵션 구성

사용자 지정 역할 파일을 사용하여 Ceph 서비스를 실행하는 노드를 정의할 수 있습니다. 사용자 지정 역할 파일은 환경으로 인해 기본 역할 할당을 사용하지 않는 경우에만 필요합니다. 예를 들어 하이퍼컨버지드 노드를 배포할 때 사전 배포된 컴퓨팅 노드는 컴퓨팅 인스턴스 목록을 포함하는 배치 목록을 포함하도록 **osd** 서비스 유형으로 **osd** 로 레이블이 지정되어야 합니다.

roles_data.yaml 파일의 서비스 정의에 따라 어떤 서비스가 실행되는 베어 메탈 인스턴스가 결정됩니다. CephStorage 역할에는 CephOSD 서비스가 있는 반면 Controller 역할에는 CephMon 및 CephMgr 서비스가 있습니다. 대부분의 구성 가능 서비스와 달리 Ceph 서비스는 서비스 구성 방법을 확인하기 위해 heat 출력이 필요하지 않습니다. **roles_data.yaml** 파일은 Heat가 실행되기 전에 배포된 Ceph 프로세스가 발생하는 경우에도 항상 Ceph 서비스 배치를 결정합니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.
2. 사용자 지정 역할을 정의하는 YAML 형식 파일을 생성합니다.
3. 구성 파일을 배포합니다.

```

$ openstack overcloud ceph deploy \
  deployed_metal.yaml \
  -o deployed_ceph.yaml \
  --roles-data custom_roles.yaml

```

3.11. CEPH 노드에 대한 SSH 사용자 옵션 구성

openstack overcloud ceph deploy 명령은 사용자와 키를 생성하여 호스트에 배포하므로 이 섹션의 절차를 수행할 필요가 없습니다. 그러나 이는 지원되는 옵션입니다.

cephadm 은 SSH를 사용하여 모든 관리형 원격 Ceph 노드에 연결합니다. Red Hat Ceph Storage 클러스터 배포 프로세스는 모든 오버클라우드 Ceph 노드에 계정 및 SSH 키 쌍을 생성합니다. 그런 다음 키 쌍이 Cephadm에 제공되어 노드와 통신할 수 있습니다.

3.11.1. Red Hat Ceph Storage 클러스터 생성 전에 SSH 사용자 생성

openstack overcloud ceph user enable 명령을 사용하여 Ceph 클러스터를 생성하기 전에 SSH 사용자를 생성할 수 있습니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.

2. SSH 사용자를 생성합니다.

\$ OpenStack 오버클라우드 ceph user enable <specification_file>

- **<specification_file>**을 사용자가 생성되고 공용 SSH 키가 설치된 클러스터를 설명하는 Ceph 사양 파일의 경로 및 이름으로 바꿉니다. 사양 파일은 수정할 노드와 개인 키가 필요한지 결정하는 정보를 제공합니다.

사양 파일 생성에 대한 자세한 내용은 [서비스 사양 생성](#)을 참조하십시오.



참고

기본 사용자 이름은 **ceph-admin** 입니다. 다른 사용자 이름을 지정하려면 **--cephadm-ssh-user** 옵션을 사용하여 다른 사용자 이름을 지정합니다.

openstack overcloud ceph user enable --cephadm-ssh-user <custom_user_name>

기본 이름을 사용하고 **--cephadm-ssh-user** 매개변수를 사용하지 않는 것이 좋습니다.

사용자가 미리 생성된 경우 **openstack overcloud ceph deploy** 를 실행할 때 **--skip-user-create** 매개변수를 사용합니다.

3.11.2. SSH 사용자 비활성화

SSH 사용자를 비활성화하면 **cephadm** 을 비활성화합니다. **cephadm** 을 비활성화하면 Ceph 클러스터를 관리하는 서비스 기능이 제거되고 연결된 명령이 작동하지 않습니다. Ceph 노드 오버클라우드 확장 작업도 차단합니다. 또한 공개 및 개인 SSH 키를 모두 제거합니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.

2. **openstack overcloud ceph user disable --fsid <FSID> <specification_file>** 명령을 사용하여 SSH 사용자를 비활성화합니다.

- **<FSID>**를 클러스터의 파일 시스템 ID로 바꿉니다. FSID는 클러스터의 고유 식별자입니다. FSID는 **deployed_ceph.yaml** 환경 파일에 있습니다.
- **<specification_file>**을 사용자가 생성된 클러스터를 설명하는 Ceph 사양 파일의 경로 및 이름으로 바꿉니다.



중요

cephadm 을 비활성화할 필요가 없는 경우 **openstack overcloud ceph user disable** 명령을 사용하지 않는 것이 좋습니다.



중요

비활성화된 후 SSH 사용자 및 Ceph 오케스트레이터 서비스를 활성화하려면 **openstack overcloud ceph user enable --fsid <FSID> <specification_file>** 명령을 사용합니다.



참고

이 명령을 실행하려면 다음을 확인하려면 Ceph 사양 파일의 경로가 필요합니다.

- SSH 사용자가 필요한 호스트는 무엇입니까.
- **_admin** 레이블이 있고 개인 SSH 키가 필요한 호스트는 무엇입니까.
- 공개 SSH 키가 필요한 호스트는 무엇입니까.

사양 파일 및 생성 방법에 대한 자세한 내용은 서비스 사양 생성을 참조하십시오.

3.12. CEPH STORAGE 컨테이너에 액세스

director 가이드를 사용하여 Red Hat OpenStack Platform 설치 및 관리에서 [컨테이너 이미지를 준비](#) 하려면 레지스트리와 언더클라우드 및 오버클라우드 설정을 컨테이너 이미지를 사용하도록 준비하는 방법에 대한 절차와 정보가 포함되어 있습니다. 이 섹션의 정보를 사용하여 Ceph Storage 컨테이너에 액세스하도록 이러한 절차를 조정합니다.

오버클라우드에서 Ceph Storage 컨테이너에 액세스하는 방법은 다음 두 가지가 있습니다.

- [원격 레지스트리에서 직접 컨테이너 다운로드](#)
- [언더클라우드의 컨테이너 캐시](#)

3.12.1. 언더클라우드의 컨테이너 캐시

[준비 과정에서 이미지 수정](#) 절차는 다음 명령을 사용하여 설명합니다.

```
sudo openstack tripleo container image prepare \
-e ~/containers-prepare-parameter.yaml \
```

원격 레지스트리에서 컨테이너 다운로드에 설명된 대로 **openstack overcloud ceph deploy** 명령에 인증 자격 증명을 제공하고 원격 레지스트리에서 Ceph 컨테이너를 직접 다운로드 하는 데 **--container-image-prepare** 옵션을 사용하지 않는 경우 Ceph를 배포하기 전에 **sudo openstack tripleo container image prepare** 명령을 실행해야 합니다.

3.12.2. 원격 레지스트리에서 직접 컨테이너 다운로드

원격 레지스트리에서 직접 컨테이너를 다운로드하도록 Ceph를 구성할 수 있습니다.

cephadm 명령은 **containers-prepare-parameter.yaml** 파일에 구성된 인증 정보를 사용하여 원격 레지스트리에 인증하고 Red Hat Ceph Storage 컨테이너를 다운로드합니다.

프로세스

1. *director 가이드를 사용하여 Red Hat OpenStack Platform 설치 및 관리의 컨테이너 이미지 준비* 절차를 사용하여 **containers-prepare-parameter.yaml** 파일을 생성합니다.
2. *개인 레지스트리에서 컨테이너 이미지 가져오기* 에 설명된 대로 **ContainerImageRegistryCredentials** 매개변수를 사용하여 **containers-prepare-parameter.yaml** 파일에 원격 레지스트리 인증 정보를 추가합니다.
3. Ceph를 배포할 때 **openstack overcloud ceph deploy** 명령을 사용하여 **containers-prepare-parameter.yaml** 파일을 전달합니다.

```
openstack overcloud ceph deploy \
  --container-image-prepare containers-prepare-parameter.yaml
```



참고

언더클라우드의 컨테이너 캐시에 설명된 대로 언더클라우드의 컨테이너 *를 캐시하지 않으면 Ceph를 배포* 할 때 동일한 **containers-prepare-parameter.yaml** 파일을 **openstack overcloud ceph deploy** 명령에 전달해야 합니다. 언더클라우드의 컨테이너가 캐시됩니다.

4장. HCI용 RED HAT CEPH STORAGE 클러스터 사용자 지정

RHOSP(Red Hat OpenStack Platform) director는 기본 설정을 사용하여 컨테이너화된 Red Hat Ceph Storage를 배포합니다. 기본 설정을 재정의하여 Ceph Storage를 사용자 지정할 수 있습니다.

사전 요구 사항

- 서버가 배포되고 해당 스토리지 네트워크가 구성됩니다.
- **openstack overcloud node provision -o ~/deployed_metal.yaml ...**에서 배포된 베어 메탈 파일을 출력으로 표시합니다.

4.1. 구성 옵션

Red Hat Ceph Storage 클러스터를 구성하는 방법은 여러 가지가 있습니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.
2. 선택 사항: 표준 형식 초기화(ini) 파일을 사용하여 Ceph 클러스터를 구성합니다.

- a. 구성 옵션을 사용하여 파일을 생성합니다.
다음은 간단한 구성 파일의 예입니다.

```
[global]
osd_crush_chooseleaf_type = 0
log_file = /var/log/ceph/$cluster-$type.$id.log

[mon]
mon_cluster_log_to_syslog = true
```

- b. 구성 파일을 저장합니다.
- c. **openstack overcloud ceph deploy --config <configuration_file_name>** 명령을 사용하여 설정을 배포합니다.
& **lt;configuration_file_name** >을 생성한 파일 이름으로 바꿉니다.

```
$ openstack overcloud ceph deploy --config initial-ceph.conf
```

3. 선택 사항: 구성 값을 **cephadm bootstrap** 명령에 보냅니다.
**openstack overcloud ceph deploy --force \ --cephadm-extra-args '<optional_arguments>' **
& **lt;optional_arguments** >를 기본 명령에 제공할 구성 값으로 바꿉니다.



참고

--log-to-file 및 **--skip-prepare-host** 인수를 사용하는 경우 **openstack overcloud ceph deploy --force \ --cephadm-extra-args '--log-to-file --skip-prepare-host'** 명령이 사용됩니다.

4.2. 서비스 사양 생성 (선택 사항)

Red Hat Ceph Storage 클러스터 서비스 사양은 Ceph Storage 서비스 배포를 설명하는 YAML 파일입니다. Ceph Storage 클러스터를 배포하기 전에 **tripleo**에 의해 자동으로 생성됩니다. 일반적으로 별도로 생성할 필요는 없습니다.

사용자 지정 서비스 사양을 생성하여 Red Hat Ceph Storage 클러스터를 사용자 지정할 수 있습니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.

2. 사양 파일을 생성합니다.

OpenStack 오버클라우드 **ceph spec deployed_metal.yaml -o <specification_file>**

- **<specification_file>**을 현재 서비스 사양으로 생성할 파일 이름으로 바꿉니다.



참고

deployed_metal.yaml은 **openstack overcloud node provision** 명령의 출력에서 제공됩니다.

3. 필요한 구성으로 생성된 파일을 편집합니다.

4. 사용자 정의 서비스 사양을 배포합니다.

```
openstack overcloud ceph deploy \
  deployed_metal.yaml \
  -o deployed_ceph.yaml \
  --ceph-spec <specification_file>
```

- **<specification_file>**을 사용자 정의 서비스 사양 파일의 이름으로 바꿉니다.

4.3. RED HAT CEPH STORAGE를 사용하는 RED HAT OPENSTACK PLATFORM용 CEPH 컨테이너

NFS Ganesha에서 Red Hat Ceph Storage를 사용하도록 RHOSP(Red Hat Openstack Platform)를 구성하려면 Ceph Storage 컨테이너가 있어야 합니다. 외부 Ceph Storage 클러스터에서 블록(RBD), Object(RGW) 또는 File(native CephFS) 스토리지만 제공하는 경우에만 Ceph Storage 컨테이너가 필요하지 않습니다.

RHOSP 17.1은 Red Hat Ceph Storage 6.x(Ceph 패키지 17.x)를 배포합니다. Ceph Storage 6.x 컨테이너는 인증이 필요한 레지스트리인 **registry.redhat.io**에서 호스팅됩니다. 자세한 내용은 [컨테이너 이미지 준비 매개변수](#)를 참조하십시오.

4.4. 고급 OSD 사양 구성

기본 사양에서 Ceph Storage 클러스터에 필요한 기능을 제공하지 않는 경우 고급 OSD 사양을 구성합니다.

프로세스

1. **stack** 사용자로 언더클라우드 노드에 로그인합니다.

2. 고급 OSD 사양을 정의하는 YAML 형식 파일을 생성합니다.

다음은 사용자 지정 OSD 사양의 예입니다.

```
data_devices:
  rotational: 1
db_devices:
  rotational: 0
```

이 예제에서는 모든 회전 장치가 데이터 장치이고 모든 순환되지 않은 장치가 공유 장치로 사용되는 OSD 사양을 생성합니다. 동적 Ceph 서비스 사양이 빌드되면 **service_type** 이 **osd** 인 경우 사양 파일의 모든 항목이 사양 섹션에 추가됩니다.

3. 사양 파일을 저장합니다.

4. 사양을 배포합니다.

```
openstack overcloud ceph deploy \ --osd-spec <osd_specification_file>
```

<osd_specification_file>을 생성한 사양 파일의 이름으로 바꿉니다.

```
$ openstack overcloud ceph deploy \ --osd-spec osd_spec.yaml \
```

추가 리소스

서비스 사양에서 OSD를 구성하는 데 사용되는 OSD 관련 속성 목록은 *Red Hat Ceph Storage Operations Guide* 에서 [OSD를 배포하기 위한 고급 서비스 사양 및 필터](#) 를 참조하십시오.

4.5. 노드별 덮어쓰기에서 마이그레이션

노드별 덮어쓰기는 Red Hat OpenStack Platform 17.0 이전에 동종이 아닌 서버 하드웨어를 관리하는 데 사용되었습니다. 이제 사용자 지정 OSD 사양 파일을 사용하여 수행합니다. 사용자 정의 [OSD 사양 파일을 생성하는 방법에 대한 자세한 내용은 고급 OSD 사양 구성](#)에서 참조하십시오.

4.6. CEPH ON-WIRE 암호화 활성화

enger 버전 2 프로토콜의 **보안 모드**를 사용하여 모든 Ceph Storage 트래픽에 대해 암호화를 활성화합니다. Red Hat Ceph Storage *Data Hardening Red Hat OpenStack Platform* 의 [암호화 및 키 관리](#)에 설명된 대로 Ceph Storage를 구성하여 Ceph on-wire 암호화를 활성화합니다.

추가 리소스

Ceph on-wire 암호화에 대한 자세한 내용은 Red Hat [Ceph Storage Architecture Guide](#) 의 Ceph on-wire [암호화](#)를 참조하십시오.

5장. HCI용 스토리지 서비스 사용자 정의

RHOSP(Red Hat OpenStack Platform) director는 기본 Ceph Storage 구성을 활성화하는 데 필요한 heat 템플릿 및 환경 파일을 제공합니다.

director는 `/usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml` 환경 파일을 사용하여 **openstack overcloud ceph deploy** 에서 배포한 Ceph 클러스터에 구성을 추가합니다.

RHOSP의 컨테이너화된 서비스에 대한 자세한 내용은 *director* 를 사용하여 Red Hat OpenStack Platform 설치 및 관리의 CLI 툴을 사용하여 기본 오버클라우드 구성 을 참조하십시오.

5.1. HCI의 COMPUTE 서비스 리소스 구성

하이퍼컨버지드 노드에서 Ceph OSD 및 Compute 서비스를 공동 배치하면 Red Hat Ceph Storage와 Compute 서비스 간의 리소스 경합이 발생할 수 있습니다. 이는 서비스가 콜렉션을 인식하지 못하기 때문에 발생합니다. 리소스 경합으로 인해 서비스 성능이 저하될 수 있으므로 하이퍼컨버지드의 이점이 오프셋됩니다.

Compute 서비스에서 사용하는 리소스를 구성하면 리소스 경합이 완화되고 HCI 성능이 향상됩니다.

프로세스

1. 언더클라우드 호스트에 stack 사용자로 로그인합니다.
2. stackrc 언더클라우드 인증 정보 파일을 소싱합니다.

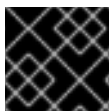
```
$ source ~/stackrc
```

3. **NovaReservedHostMemory** 매개변수를 **ceph-overrides.yaml** 파일에 추가합니다. 다음은 사용 예입니다.

```
parameter_defaults:
  ComputeHCIParameters:
    NovaReservedHostMemory: 75000
```

NovaReservedHostMemory 매개변수는 `/etc/nova/nova.conf` 의 **reserved_host_memory_mb** 기본값을 재정의합니다. 이 매개 변수는 Ceph OSD에 필요한 메모리를 가상 머신에 제공하는 Nova 스케줄러 중지로 설정됩니다.

위의 예제에서는 하이퍼바이저의 기본 예약 메모리 외에 호스트당 OSD당 5GB를 예약합니다. IOPS에 최적화된 클러스터에서 OSD당 더 많은 메모리를 예약하여 성능을 향상시킬 수 있습니다. 5GB 번호는 필요에 따라 구체화할 수 있는 시작점으로 제공됩니다.



중요

openstack overcloud deploy 명령을 사용할 때 이 파일을 포함합니다.

5.2. 사용자 정의 환경 파일 구성

director는 배포된 Red Hat Ceph Storage 클러스터에 기본 설정을 적용합니다. 사용자 지정 환경 파일에 추가 구성을 정의해야 합니다.

프로세스

— 114 —

1. **stack** 사용자로 언더클라우드에 로그인합니다.
2. 파일을 생성하여 사용자 지정 구성을 정의합니다.
vi /home/stack/templates/storage-config.yaml
3. 매개 변수 **_defaults** 섹션을 파일에 추가합니다.
4. 사용자 지정 구성 매개 변수를 추가합니다. 매개 변수 정의에 대한 자세한 내용은 [Overcloud 매개 변수](#)를 참조하십시오.

```
parameter_defaults:
  CinderEnableIscsiBackend: false
  CinderEnableRbdBackend: true
  CinderBackupBackend: ceph
  NovaEnableRbdBackend: true
  GlanceBackend: rbd
```



참고

사용자 지정 구성 파일에 정의된 매개 변수는 **/usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml**의 해당 기본 설정을 재정의합니다.

5. 파일을 저장합니다.

추가 리소스

사용자 지정 구성은 오버클라우드 배포 중에 적용됩니다.

5.3. CEPH 메타데이터 서버 활성화

Ceph Metadata Server(MDS)는 **ceph-mds** 데몬을 실행합니다. 이 데몬은 CephFS에 저장된 파일과 관련된 메타데이터를 관리합니다. CephFS는 기본적으로 또는 NFS 프로토콜을 통해 사용할 수 있습니다.



참고

Red Hat은 Shared File Systems 서비스(manila)의 기본 CephFS 및 CephFS NFS 백엔드를 사용하여 Ceph MDS 배포를 지원합니다.

프로세스

- 오버클라우드를 배포할 때 Ceph MDS를 활성화하려면 다음 환경 파일을 사용합니다.

```
/usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml
```



참고

기본적으로 Ceph MDS는 컨트롤러 노드에 배포됩니다. 자체 전용 노드에 Ceph MDS를 배포할 수 있습니다.

추가 리소스

- [Red Hat Ceph Storage 파일 시스템 가이드](#)

5.4. CEPH OBJECT GATEWAY 오브젝트 스토리지

Ceph Object Gateway(RGW)는 Red Hat Ceph Storage 클러스터 내의 오브젝트 스토리지 기능에 액세스하는 인터페이스를 제공합니다.

director를 사용하여 Ceph를 배포하면 director에서 RGW를 자동으로 활성화합니다. Object Storage 서비스(swift)를 직접 대체합니다. 일반적으로 Object Storage 서비스를 사용하는 서비스는 추가 구성없이 RGW를 대신 사용할 수 있습니다. Object Storage 서비스는 업그레이드된 Ceph 클러스터에 대한 오브젝트 스토리지 옵션으로 계속 사용할 수 있습니다.

이를 활성화하기 위해 별도의 RGW 환경 파일이 필요하지 않습니다. 다른 오브젝트 스토리지 옵션의 환경 파일에 대한 자세한 내용은 [5.5절. "Red Hat OpenStack Platform 오브젝트 스토리지에 대한 배포 옵션"](#)을 참조하십시오.

기본적으로 Ceph Storage는 OSD(오브젝트 스토리지 데몬)당 250개의 배치 그룹을 허용합니다. RGW를 활성화하면 Ceph Storage는 RGW에 필요한 다음과 같은 6개의 추가 풀을 생성합니다.

- `.rgw.root`
- `<zone_name>.rgw.control`
- `<zone_name>.rgw.meta`
- `<zone_name>.rgw.log`
- `<zone_name>.rgw.buckets.index`
- `<zone_name>.rgw.buckets.data`



참고

배포에서 `<zone_name>`은 풀이 속한 영역의 이름으로 교체됩니다.

추가 리소스

- RGW에 대한 자세한 내용은 Red Hat Ceph Storage [Object Gateway Guide](#)를 참조하십시오.
- Swift 대신 RGW를 사용하는 방법에 대한 자세한 내용은 [BLock Storage 볼륨 백업 가이드](#)를 참조하십시오.

5.5. RED HAT OPENSTACK PLATFORM 오브젝트 스토리지에 대한 배포 옵션

오버클라우드 오브젝트 스토리지를 배포하는 방법은 다음 세 가지가 있습니다.

- Ceph Object Gateway(RGW)
[5.4절. "Ceph Object Gateway 오브젝트 스토리지"](#)에 설명된 대로 RGW를 배포하려면 오버클라우드 배포 중에 다음 환경 파일을 포함합니다.

```
-e environments/cephadm/cephadm.yaml
```

이 환경 파일은 Ceph 블록 스토리지(RBD) 및 RGW를 모두 구성합니다.

- Object Storage 서비스(swift)
RGW 대신 Object Storage 서비스(swift)를 배포하려면 오버클라우드 배포 중에 다음 환경 파일을 포함합니다.

```
-e environments/cephadm/cephadm-rbd-only.yaml
```

cephadm-rbd-only.yaml 파일은 Ceph RBD를 구성하지만 RGW는 구성하지 않습니다.



참고

Red Hat Ceph Storage 클러스터를 업그레이드하기 전에 Object Storage 서비스(swift)를 사용한 경우 환경/ceph-ansible/**ceph-ansible.yaml** 파일을 환경/**cephadm** /**cephadm-rbd-only.yaml** 로 교체하여 RGW 대신 Object Storage 서비스(swift)를 계속 사용할 수 있습니다. 자세한 내용은 [Red Hat OpenStack Platform의 마이너 업데이트 수행](#)을 참조하십시오.

Red Hat OpenStack Platform은 Object Storage 서비스(swift)에서 Ceph Object Gateway(RGW)로의 마이그레이션을 지원하지 않습니다.

- 오브젝트 스토리지 없음
RGW 또는 Object Storage 서비스(swift)를 사용하여 Ceph를 배포하려면 오버클라우드 배포 중에 다음 환경 파일을 포함합니다.

```
-e environments/cephadm/cephadm-rbd-only.yaml
-e environments/disable-swift.yaml
```

cephadm-rbd-only.yaml 파일은 RGW가 아닌 RBD를 구성합니다. **disable-swift.yaml** 파일을 사용하면 Object Storage 서비스(swift)가 배포되지 않습니다.

5.6. CEPH를 사용하도록 블록 스토리지 백업 서비스 구성

블록 스토리지 백업 서비스(cinder-backup)는 기본적으로 비활성화되어 있습니다. Ceph와 함께 사용하려면 활성화해야 합니다.

프로세스

오버클라우드를 배포할 때 Block Storage Backup 서비스(cinder-backup)를 활성화하려면 다음 환경 파일을 사용합니다.

```
`/usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml`.
```

5.7. CEPH 노드에 대해 여러 개의 결합된 인터페이스 구성

결합된 인터페이스를 사용하여 여러 NIC를 결합하고 네트워크 연결에 중복성을 추가합니다. Ceph 노드에 NIC가 충분한 경우 각 노드에 결합된 인터페이스를 여러 개 생성하여 중복 기능을 확장할 수 있습니다.

노드에 필요한 각 네트워크 연결에 대해 결합된 인터페이스를 사용합니다. 이렇게 하면 중복과 각 네트워크에 대한 전용 연결이 모두 제공됩니다.

자세한 내용 및 절차는 *director 가이드*를 사용하여 Red Hat OpenStack Platform 설치 및 관리에서 [오버클라우드 네트워크 프로비저닝](#)을 참조하십시오.

5.8. HCI용 오버클라우드 배포 시작

RHOSP(Red Hat OpenStack Platform) 환경에 대한 변경 사항을 구현하려면 오버클라우드를 배포해야 합니다.

사전 요구 사항

- 언더클라우드를 설치하기 전에 **undercloud.conf** 파일에 **generate_service_certificate=false**를 설정합니다. 그렇지 않으면 *Hardening Red Hat OpenStack Platform* 의 오버클라우드 [공용 끝점](#)에서 [SSL/TLS 활성화](#)에 설명된 대로 오버클라우드에 SSL/TLS를 구성해야 합니다.



참고

오버클라우드 배포 중에 Ceph 대시보드를 추가하려면 *director*와 함께 [Red Hat Ceph Storage](#) 및 [Red Hat OpenStack Platform](#) 배포의 오버클라우드 배포에 [Red Hat Ceph Storage](#) 대시보드 추가를 참조하십시오.

프로세스

- 오버클라우드를 배포합니다. 배포 명령에는 추가 인수가 필요합니다. 예를 들면 다음과 같습니다.

```
$ openstack overcloud deploy --templates -r /home/stack/templates/roles_data_custom.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml \
-e /home/stack/templates/storage-config.yaml \
-e /home/stack/templates/deployed-ceph.yaml \
--ntp-server pool.ntp.org
```

예제 명령은 다음 옵션을 사용합니다.

- **--templates** - 기본 heat 템플릿 컬렉션인 **/usr/share/openstack-tripleo-heat-templates/**에서 오버클라우드를 생성합니다.
- **-r /home/stack/templates/roles_data_custom.yaml** - 사용자 지정 역할 정의 파일을 지정합니다.
- **-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml** - 이전에 배포한 Ceph Storage 클러스터를 종료하도록 director를 설정합니다. 이 환경 파일은 기본적으로 RGW를 배포합니다. 또한 풀, 키 및 데몬을 생성합니다.
- **-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml** - Ceph 메타데이터 서버를 활성화합니다.
- **-e /usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml** - Block Storage Backup 서비스를 활성화합니다.
- **-e /home/stack/templates/storage-config.yaml** - 사용자 지정 Ceph Storage 구성이 포함된 환경 파일을 추가합니다.
- **-e /home/stack/templates/deployed-ceph.yaml** - 이전에 실행한 **openstack overcloud ceph deploy** 명령의 출력으로 Ceph 클러스터 설정이 포함된 환경 파일을 추가합니다.
- **--ntp-server pool.ntp.org** - NTP 서버를 설정합니다.



참고

전체 옵션 목록은 **openstack help overcloud deploy** 명령을 실행합니다.

추가 리소스

- 자세한 내용은 *director 가이드*를 [사용하여 Red Hat OpenStack Platform 설치 및 관리의 CLI 툴을 사용하여](#) 기본 오버클라우드 구성을 참조하십시오.

6장. HCI 구성 확인

배포가 완료되면 HCI 환경이 올바르게 구성되었는지 확인합니다.

6.1. HCI 구성 확인

HCI 환경을 배포한 후 구성이 지정된 상태로 배포가 완료되었는지 확인합니다.

프로세스

1. ceph 셸을 시작합니다.
2. NUMA 및 메모리 대상 구성을 확인합니다.

```
[ceph: root@oc0-controller-0 /]# ceph config dump | grep numa
osd                                advanced osd_numa_auto_affinity      true
[ceph: root@oc0-controller-0 /]# ceph config dump | grep autotune
osd                                advanced osd_memory_target_autotune  true
[ceph: root@oc0-controller-0 /]# ceph config get mgr
mgr/cephadm/autotune_memory_target_ratio
0.200000
```

3. 특정 OSD 구성을 확인합니다.

```
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_memory_target
4294967296
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_memory_target_autotune
true
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_numa_auto_affinity
true
```

4. 특정 OSD 백필 구성을 확인합니다.

```
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_recovery_op_priority
3
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_max_backfills
1
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_recovery_max_active_hdd
3
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_recovery_max_active_ssd
10
```

5. 컴퓨팅 노드에서 **reserved_host_memory_mb** 구성을 확인합니다.

```
$ sudo podman exec -ti nova_compute /bin/bash
bash-5.1$ grep reserved_host_memory_mb /etc/nova/nova.conf
```

7장. 하이퍼컨버지드 노드 확장

HCI 노드를 확장하거나 축소하려면 컴퓨팅 노드 또는 Red Hat Ceph Storage 노드를 확장하는 것과 동일한 원칙과 방법이 적용됩니다.

7.1. HCI 환경에서 하이퍼컨버지드 노드 확장

HCI 환경에서 하이퍼 컨버지드 노드를 확장하려면 하이퍼 컨버지드 노드를 확장하는 것과 동일한 절차를 따르십시오. 자세한 내용은 [오버클라우드에 노드 추가](#)를 참조하십시오.



참고

새 노드를 태그하는 경우 올바른 플레이버를 사용해야 합니다.

7.2. HCI 환경에서 하이퍼컨버지드 노드 축소

HCI 환경에서 하이퍼 컨버지드 노드를 축소하려면 HCI 노드에서 Ceph OSD 서비스를 재조정하고 HCI 노드에서 인스턴스를 마이그레이션한 후 오버클라우드에서 컴퓨팅 노드를 제거해야 합니다.

절차

1. HCI 노드에서 Ceph OSD 서비스를 비활성화하고 재조정합니다. 이 단계는 HCI 또는 Red Hat Ceph Storage 노드를 제거할 때 director가 Red Hat Ceph Storage 클러스터를 자동으로 재조정하지 않기 때문에 필요합니다. 자세한 내용은 [Red Hat Ceph Storage 및 Red Hat OpenStack Platform 배포를 통해 Ceph Storage 클러스터 스케일링](#)을 참조하십시오.
2. HCI 노드에서 인스턴스를 마이그레이션합니다. 자세한 내용은 [인스턴스 생성 가이드를 위한 컴퓨팅 서비스 구성 가이드의 컴퓨팅 노드 간에 가상 머신 마이그레이션](#)을 참조하십시오.
3. 오버클라우드에서 컴퓨팅 노드를 삭제합니다. 자세한 내용은 [컴퓨팅 노드 제거](#)를 참조하십시오.

부록 A. 추가 정보

A.1. 구성 지침

다음 구성 지침은 Hyperconverged Infrastructure 환경을 만들기 위한 프레임워크를 제공하기 위한 것입니다. 이 지침은 모든 Red Hat OpenStack Platform 설치에 대해 고유한 구성 매개변수를 제공하는 것은 아닙니다. 특정 환경에 맞는 특정 지침 및 제안 사항은 [Red Hat Customer Experience and Engagement 팀](#)에 문의하십시오.

- [클러스터 크기 조정 및 확장](#)
- [용량 계획 및 크기 조정](#)

A.1.1. 클러스터 크기 조정 및 확장

[Red Hat Ceph Storage 하드웨어 가이드](#)에서는 IOPS 최적화, 처리량 최적화, 비용 및 용량 최적화된 Ceph 배포 시나리오에 대한 권장 사항을 제공합니다. 배포 시나리오를 가장 잘 나타내는 권장 사항에 따라 컴퓨팅 워크로드를 지원하는 데 필요한 NIC, CPU 및 RAM을 추가합니다.

최적의 작은 설치 공간 구성은 7개의 노드로 구성됩니다. 사용자 환경에 IOPS 최적화 성능에 대한 요구 사항이 있고 모든 플래시 스토리지를 사용하는 경우 최적화된 처리량을 사용해야 합니다.

3개의 노드 Ceph Storage 클러스터 구성이 가능합니다. 이 구성에서는 다음을 수행해야 합니다.

- 모든 플래시 스토리지를 사용합니다.
- **ceph.conf** 파일에서 **replica_count** 매개변수를 3으로 설정합니다.
- **ceph.conf** 파일에서 **min_size** 매개변수를 2로 설정합니다.

노드가 이 구성에 서비스를 벗어나면 IOPS가 계속됩니다. 데이터의 복사본을 3개 유지하기 위해 3번째 노드로의 복제는 서비스로 돌아올 때까지 대기됩니다. 그런 다음 데이터가 세 번째 노드로 백필됩니다.



참고

최대 64개의 노드의 HCI 구성이 테스트되었습니다. 일부 HCI 환경 예제는 최대 128개의 노드로 문서화되었습니다. 이러한 대규모 클러스터는 Support Exception 및 Consulting Services 참여를 통해 고려할 수 있습니다. 자세한 내용은 [Red Hat Customer Experience and Engagement 팀](#)에 문의하십시오.

두 개의 NUMA 노드가 있는 배포는 하나의 NUMA 노드와 Ceph OSD 서비스에서 대기 시간에 민감한 컴퓨팅 워크로드를 호스팅할 수 있습니다. 노드 모두에 네트워크 인터페이스가 있고 디스크 컨트롤러가 노드 0에 있는 경우 Storage 네트워크에 대해 노드 0의 네트워크 인터페이스를 사용하고 노드 0에서 Ceph OSD 워크로드를 호스팅합니다. 노드 1에서 컴퓨팅 워크로드를 호스팅하고 노드 1에서 네트워크 인터페이스를 사용하도록 구성합니다. 배포를 위해 하드웨어를 인수할 때는 어떤 NIC가 어떤 노드를 사용하고 스토리지와 워크로드 간에 분할하려고 하는지 유의하십시오.

A.1.2. 용량 계획 및 크기 조정

[Red Hat Ceph Storage 하드웨어 가이드](#)에 정의된 처리량 최적화된 Ceph 솔루션은 IOPS에 대한 최적화가 필요하지 않은 대부분의 배포에 대해 균형 잡힌 솔루션을 제공합니다. 솔루션에 제공된 구성 지침 외에도 환경을 생성할 때 다음 사항에 유의하십시오.

- OSD당 5GB의 RAM을 할당하면 OSD에 충분한 작동 메모리가 있는지 확인합니다. 하드웨어가 이 요구 사항을 지원할 수 있는지 확인합니다.
- CPU 속도는 사용 중인 스토리지 미디어와 일치해야 합니다. SSD와 같은 더 빠른 스토리지 매체의 장점은 CPU에 의해 너무 느려져 지원할 수 있습니다. 마찬가지로 빠른 CPU를 더 빠른 스토리지 매체에서 더 효율적으로 사용할 수 있습니다. CPU 및 스토리지 미디어 속도를 조정하여 다른 쪽의 병목 현상이 되지 않도록 합니다.

A.2. 하이퍼컨버지드 인프라 환경 구성을 위한 가이드 및 리소스

다음 가이드에는 하이퍼컨버지드 인프라 환경 구성에 도움이 되는 추가 정보 및 절차가 포함되어 있습니다.

- [director와 함께 Red Hat Ceph 및 OpenStack 배포](#)
 - 이 가이드에서는 Red Hat OpenStack Platform director를 사용하여 Red Hat Ceph Storage 클러스터로 오버클라우드를 생성하는 방법에 대한 정보를 제공합니다. 여기에는 director를 통해 Ceph 클러스터를 사용자 정의하는 지침이 포함됩니다.
- [director를 사용하여 Red Hat OpenStack Platform 설치 및 관리](#)
 - 이 가이드에서는 Red Hat OpenStack Platform 환경의 엔드 투 엔드 배포에 대한 지침을 제공합니다. 여기에는 director 설치, 환경 계획, director를 사용한 OpenStack 환경 구축 작업이 포함됩니다.
- [Red Hat OpenStack Platform 네트워킹 구성](#)
 - 이 가이드에서는 Red Hat OpenStack Platform 네트워킹 작업에 대해 자세히 설명합니다.
- [영구 스토리지 구성](#)
 - 이 가이드에서는 Red Hat OpenStack Platform 환경에서 영구 스토리지를 사용하고 관리하기 위한 다양한 절차에 대해 자세히 설명합니다. 또한 각 영구 스토리지 유형의 해당 OpenStack 서비스를 구성하고 관리하는 절차도 포함되어 있습니다.
- [베어 메탈 프로비저닝 서비스 구성](#)
 - 이 가이드에서는 클라우드 사용자를 위한 물리적 시스템을 프로비저닝 및 관리하기 위해 Red Hat OpenStack Platform 환경의 오버클라우드에서 Bare Metal Provisioning 서비스 설치 및 구성에 대해 자세히 설명합니다.
- [Red Hat OpenStack Platform 강화](#)
 - 이 가이드에서는 Red Hat OpenStack Platform 환경의 보안을 강화하는 방법에 대한 모범 사례 및 개념 정보를 제공합니다.
- [릴리스 노트](#)
 - 이 문서에서는 Red Hat OpenStack Platform의 주요 기능, 기능 향상 및 알려진 문제에 대해 간단히 설명합니다.