Red Hat OpenStack Platform 16.2

하이퍼컨버지드 인프라 가이드

Red Hat OpenStack Platform 오버클라우드에서 Hyperconverged Infrastructure 이해 및 구성

Last Updated: 2022-07-18

Red Hat OpenStack Platform 16.2 하이퍼컨버지드 인프라 가이드

Red Hat OpenStack Platform 오버클라우드에서 Hyperconverged Infrastructure 이해 및 구성

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

초록

이 문서에서는 동일한 호스트에 Compute 및 Ceph Storage 서비스를 공동 배치하는 하이퍼컨버지스에 대한 Red Hat OpenStack Platform 구현에 대해 설명합니다.

차례

PREFACE 3

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. 사전 요구 사항 6
1.2. 하이퍼컨버지드 노드에 대한 오버클라우드 역할 준비 6
1.2.1. 멀티 디스크 클러스터의 root 디스크 정의 9
1.3. 하이퍼컨버지드 노드에서 리소스 격리 구성 11
1.3.1. Compute 서비스를 예약하기 위해 CPU 및 메모리 리소스를 자동화하는 프로세스 12
1.3.2. Red Hat Ceph Storage 백필 및 복구 작업 13
1.4. 사용 가능한 RED HAT CEPH STORAGE 패키지 확인 14
1.4.1. ceph-ansible 패키지 버전 확인 14
1.4.2. 사전 프로비저닝된 노드의 패키지 확인 14
1.5. HCI 오버클라우드 배포 14
1.5.1. ceph-anible 이 실행되는 노드 제한 16
1.6. OPENSTACK WORKFLOW COMPUTE CPU 및 메모리 계산기 17
1.7. 추가 리소스 18

2장. 하이퍼컨버지드 노드 확장 19
2.1. HCI 환경에서 하이퍼컨버지드 노드 확장 19
2.2. HCI 환경에서 하이퍼컨버지드 노드 축소 19

PREFACE

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 이를 개선하는지 알려주십시오.

DDF(직접 문서 피드백) 기능 사용

특정 문장, 단락 또는 코드 블록에 대한 직접 주석은 **피드백 추가** DDF 기능을 사용하십시오.

1. *다중 페이지 HTML* 형식으로 설명서를 봅니다.
2. 문서 오른쪽 상단에 **Feedback** (피드백) 버튼이 표시되는지 확인합니다.
3. 주석 처리하려는 텍스트 부분을 강조 표시합니다.
4. **피드백 추가**를 클릭합니다.
5. 주석을 사용하여 **Add Feedback** (피드백 추가) 필드를 작성합니다.
6. 선택 사항: 설명서 팀이 문제에 대한 자세한 내용을 문의할 수 있도록 이메일 주소를 추가하십시오.
7. **Submit(제출)**을 클릭합니다.

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

RHOSP(Red Hat OpenStack Platform) HCI(하이퍼컨버지드 인프라)는 하이퍼컨버지드 노드로 구성됩니다. 서비스는 최적화된 리소스 사용을 위해 이러한 하이퍼컨버지드 노드에 공동 배치됩니다. RHOSP HCI에서는 계산 및 스토리지 서비스가 하이퍼컨버지드 노드에 함께 배치됩니다. 하이퍼컨버지드 노드만 사용하여 오버클라우드를 배포하거나 일반 Compute 및 Ceph Storage 노드가 혼합된 하이퍼컨버지드 노드를 배포할 수 있습니다.

참고

Red Hat Ceph Storage를 스토리지 공급자로 사용해야 합니다.

작은 정보

- Ceph 메모리 설정을 자동으로 조정하려면 ceph-ansible 3.2 이상을 사용합니다.
- BlueStore를 HCI 배포의 백엔드로 사용하여 BlueStore 메모리 처리 기능을 사용합니다.

오버클라우드에서 HCI를 생성 및 배포하려면 Network Function Virtualization과 같은 오버클라우드의 다른 기능과 통합되고 하이퍼컨버지드 노드에서 Compute 및 Red Hat Ceph Storage 서비스의 최적 성능을 확인해야 합니다.

1. 하이퍼컨버지드 노드인 **ComputeHCI** 에 대해 사전 정의된 사용자 지정 Overcloud 역할을 준비합니다.
2. 리소스 격리 구성.
3. 사용 가능한 Red Hat Ceph Storage 패키지를 확인합니다.
4. HCI Overcloud를 배포합니다.

1.1. 사전 요구 사항

- 언더클라우드를 배포했습니다. Undercloud 배포 방법에 대한 지침은 Director 설치 및 사용을 참조하십시오.
- RHOSP 컴퓨팅 및 Red Hat Ceph Storage 요구 사항을 충족하는 노드를 프로비저닝할 수 있습니다. 자세한 내용은 Basic Overcloud Deployment 를 참조하십시오.
- 환경에 있는 모든 노드를 등록했습니다. 자세한 내용은 노드 등록을 참조하십시오.
- 해당 환경의 모든 노드에 태그를 지정했습니다. 자세한 내용은 노드 태그 수동을 참조하십시오.
- Compute 및 Ceph OSD 서비스에 사용하려는 노드에서 디스크를 정리했습니다. 자세한 내용은 Ceph Storage 노드 디스크 정리를 참조하십시오.
- Red Hat Content Delivery Network 또는 Red Hat Satellite 서버에 등록할 오버클라우드 노드를 준비했습니다. 자세한 내용은 Ansible 기반 Overcloud 등록을 참조하십시오.

1.2. 하이퍼컨버지드 노드에 대한 오버클라우드 역할 준비

노드를 하이퍼컨버지드로 지정하려면 하이퍼컨버지드 역할을 정의해야 합니다. RHOSP(Red Hat OpenStack Platform)는 하이퍼컨버지드 노드에 대해 사전 정의된 역할 **ComputeHCI** 를 제공합니다. 이

역할은 Compute 및 Ceph 개체 스토리지 데몬(OSD) 서비스를 함께 배치하여 동일한 하이퍼컨버지드 노드에 함께 배포할 수 있습니다.

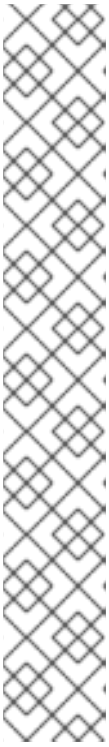
절차

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

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

3. **ComputeHCI** 역할이 포함된 새 사용자 지정 역할 데이터 파일을 오버클라우드에 사용할 다른 역할과 함께 생성합니다. 다음 예제에서는 **Controller**,**ComputeHCI**,**Compute** 및 **CephStorage** 역할을 포함하는 역할 데이터 파일 **roles_data_hci.yaml** 을 생성합니다.

```
(undercloud)$ openstack overcloud roles \
 generate -o /home/stack/templates/roles_data_hci.yaml \
  Controller ComputeHCI Compute CephStorage
```



참고

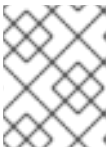
생성된 사용자 지정 역할 데이터 파일에 **ComputeHCI** 역할에 나열된 네트워크에는 계산 및 스토리지 서비스 모두에 필요한 네트워크가 포함됩니다. 예를 들면 다음과 같습니다.

```
- name: ComputeHCI
  description: |
    Compute node role hosting Ceph OSD
  tags:
    - compute
  networks:
    InternalApi:
      subnet: internal_api_subnet
    Tenant:
      subnet: tenant_subnet
    Storage:
      subnet: storage_subnet
    StorageMgmt:
      subnet: storage_mgmt_subnet
```

4. **network_data.yaml** 파일의 로컬 사본을 생성하여 구성 가능한 네트워크를 오버클라우드에 추가합니다. **network_data.yaml** 파일은 기본 네트워크 환경 파일 **/usr/share/openstack-tripleo-heat-templates/environments/*** 와 상호 작용하여 **ComputeHCI** 역할에 대해 정의한 네트워크를 하이퍼컨버지드 노드와 연결합니다. 자세한 내용은 *Advanced Overcloud Customization* 가이드 의 Adding a composable network 를 참조하십시오.
5. Red Hat Ceph Storage의 성능을 개선하려면 network **_data.yaml의 로컬 사본에서 점보 프레임의 스토리지 및 Storage Mgmt 네트워크** 모두에 대한 MTU 설정을 **9000** 으로 업데이트합니다. 자세한 내용은 Director에서 MTU 설정 구성 및 점보 프레임 구성을 참조하십시오.
6. 하이퍼컨버 **지드 노드의 computeHCI** 오버클라우드 플레이버를 생성합니다.

```
(undercloud)$ openstack flavor create --id auto \
 --ram <ram_size_mb> --disk <disk_size_gb> \
 --vcpus <no_vcpus> computeHCI
```

- **<ram_size_mb>** 를 베어 메탈 노드의 RAM(MB)으로 바꿉니다.
- **<disk_size_gb>** 를 베어 메탈 노드의 디스크 크기(GB)로 바꿉니다.
- **<no_vcpus>** 를 베어 메탈 노드의 CPU 수로 바꿉니다.



참고

이러한 속성은 인스턴스를 예약하는 데 사용되지 않습니다. 그러나 계산 스케줄러는 디스크 크기를 사용하여 루트 파티션 크기를 결정합니다.

7. 노드 목록을 검색하여 UUID를 확인합니다.

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

8. 사용자 정의 HCI 리소스 클래스를 사용하여 하이퍼컨버지드로 지정할 각 베어 메탈 노드에 태그를 지정합니다.

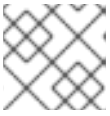
```
(undercloud)$ openstack baremetal node set \
 --resource-class baremetal.HCI <node>
```

<node> 를 베어 메탈 노드의 ID로 바꿉니다.

9. **computeHCI** 플레이버를 사용자 지정 HCI 리소스 클래스와 연결합니다.

```
(undercloud)$ openstack flavor set \
--property resources:CUSTOM_BAREMETAL_HCI=1 \
computeHCI
```

베어 메탈 서비스 노드의 리소스 클래스에 해당하는 사용자 지정 리소스 클래스의 이름을 확인하려면 리소스 클래스를 대문자로 변환하려면 모든 문장 부호를 밑줄로 바꾸고 접두사는 **CUSTOM_** 로 바꿉니다.



참고

플레이버는 베어 메탈 리소스 클래스의 인스턴스 하나만 요청할 수 있습니다.

10. Compute 스케줄러가 베어 메탈 플레이버 속성을 사용하여 인스턴스를 예약하지 못하도록 다음 플레이버 속성을 설정합니다.

```
(undercloud)$ openstack flavor set \
 --property resources:VCPU=0 \
 --property resources:MEMORY_MB=0 \
 --property resources:DISK_GB=0 computeHCI
```

11. 다음 매개변수를 **node-info.yaml** 파일에 추가하여 하이퍼컨버지드 및 컨트롤러 노드 수와 하이퍼컨버지드 및 컨트롤러 지정 노드에 사용할 플레이버를 지정합니다.

```
parameter_defaults:
```

```
OvercloudComputeHCIFlavor: computeHCI
ComputeHCICount: 3
OvercloudControlFlavor: baremetal
ControllerCount: 3
```

추가 리소스

- 구성 가능 서비스 및 사용자 지정 역할
- roles_data 파일 검사
- 역할에 노드 및 플레이버 할당

1.2.1. 멀티 디스크 클러스터의 root 디스크 정의

여러 디스크가 있는 노드의 경우 director가 프로비저닝 중에 root 디스크를 식별해야 합니다. 예를 들어 대부분의 Ceph Storage 노드는 여러 디스크를 사용합니다. 기본적으로 director는 프로비저닝 프로세스 중에 오버클라우드 이미지를 root 디스크에 씁니다.

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

- **모델** (문자열): 장치 식별자.
- **벤더** (문자열): 장치 벤더.
- **serial** (문자열): 디스크 일련 번호.
- **hctl** (문자열): host:Channel:Target: SCSI의 Lun.
- **크기** (정수): 장치 크기(GB)입니다.
- **WWN (** 문자열): 고유한 스토리지 식별자.
- **wwn_with_extension** (문자열): 공급업체 확장이 추가된 고유한 스토리지 식별자입니다.
- **wwn_vendor_extension** (문자열): 고유한 벤더 스토리지 식별자.
- **rotational** (부울): 회전 장치(HDD)의 경우 true이며 그렇지 않으면 false(SSD)입니다.
- **이름** (문자열): 장치 이름(예: /dev/sdb1)

중요

name 속성은 영구적인 이름이 있는 장치에만 사용합니다. 노드가 부팅될 때 값이 변경될 수 있으므로 **name**을 사용하여 다른 장치에 대해 root 디스크를 설정하지 마십시오.

일련 번호를 사용하여 root 장치를 지정할 수 있습니다.

절차

1. 각 노드의 하드웨어 인트로스펙션에서 디스크 정보를 확인합니다. 다음 명령을 실행하여 노드의 디스크 정보를 표시합니다.

```
(undercloud)$ openstack baremetal introspection data save 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0 | jq ".inventory.disks"
```

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. **openstack baremetal node set --property root_device=**를 입력하여 노드의 root 디스크를 설정합니다. root 디스크를 정의하는 데 가장 적절한 하드웨어 속성값을 포함시킵니다.

```
(undercloud)$ openstack baremetal node set --property root_device='{"serial":"
<serial_number>"}' <node-uuid>
```

예를 들어, root 장치를 일련 번호가 **61866da04f380d001ea4e13c12e36ad6**인 disk 2로 설정하려면 다음 명령을 입력합니다.

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

참고

각 노드의 BIOS를 설정하여 선택한 root 디스크로 부팅이 포함되도록 합니다. 먼저 네트워크에서 부팅한 다음 root 디스크에서 부팅하도록 부팅 순서를 구성합니다.

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

1.3. 하이퍼컨버지드 노드에서 리소스 격리 구성

하이퍼컨버지드 노드의 Ceph OSD 및 계산 서비스를 결합하면 Red Hat Ceph Storage와 Compute 서비스 간의 리소스 경합이 발생할 수 있습니다. 리소스 경합으로 인해 하이퍼컨버지스(hyconvergence)의 이점을 오프셋하는 서비스가 저하될 수 있습니다.

경합을 방지하려면 Ceph 및 Compute 서비스 모두에 대한 리소스 격리를 구성해야 합니다.

절차

1. 선택 사항: Compute 환경 파일에 다음 매개 변수를 추가하여 자동 생성된 Compute 설정을 재정의합니다.

```
parameter_defaults:
  ComputeHCIParameters:
    NovaReservedHostMemory: <ram>
    NovaCPUAllocationRatio: <ratio>
```

   - **<ram>** 을 Ceph OSD 서비스 및 하이퍼컨버지드 노드의 인스턴스 오버헤드(MB)에 예약할 RAM 양으로 바꿉니다.
   - **<ratio>** 를 인스턴스를 배포할 컴퓨팅 노드를 선택할 때 컴퓨팅 스케줄러에서 사용해야 하는 비율로 바꿉니다.
     자동 생성 Compute 설정에 대한 자세한 내용은 Compute 서비스에 대해 예약할 CPU 및 메모리 리소스를 자동 생성을 위한 프로세스를 참조하십시오.

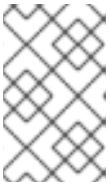
2. Red Hat Ceph Storage의 메모리 리소스를 예약하려면 **/home/stack/templates/storage-container-config.yaml에서 매개변수 is_hci 를 true 로 설정합니다.**

```
parameter_defaults:
  CephAnsibleExtraConfig:
    is_hci: true
```

   이를 통해 **ceph-ansible** 은 HCI 배포에 대한 **osd_memory_target** 매개변수 설정을 자동으로 조정하여 Red Hat Ceph Storage의 메모리 리소스를 예약하고 Ceph OSD에 의한 메모리 증가를 줄일 수 있습니다.

주의

Red Hat은 **ceph_osd_docker_memory_limit** 매개 변수를 직접 재정의하는 것을 권장하지 않습니다.



참고

ceph-ansible 3.2부터 **ceph_osd_docker_memory_limit** 는 파일 저장소 또는 BlueStore 백엔드 사용 여부와 관계없이 Ansible에서 검색한 대로 호스트의 최대 메모리로 자동 설정됩니다.

3. 선택 사항: 기본적으로 **ceph-ansible** 은 각 Ceph OSD에 대해 하나의 vCPU를 예약합니다. Ceph OSD당 두 개 이상의 CPU가 필요한 경우 **/home/stack/templates/storage-container-config.yaml에 다음 구성을 추가합니다.**

```
parameter_defaults:
  CephAnsibleExtraConfig:
    ceph_osd_docker_cpu_limit: <cpu_limit>
```

<cpu_limit> 를 각 Ceph OSD에 예약할 CPU 수로 바꿉니다.

하드웨어 및 워크로드를 기반으로 CPU 리소스를 조정하는 방법에 대한 자세한 내용은 Red Hat Ceph Storage 하드웨어 선택 가이드를 참조하십시오.

4. 선택 사항: Ceph 환경 파일에 다음 매개 변수를 추가하여 Ceph OSD가 제거될 때 Red Hat Ceph Storage 백필 및 복구 작업의 우선 순위를 줄입니다.

```
parameter_defaults:
  CephConfigOverrides:
    osd_recovery_op_priority: <priority_value>
    osd_recovery_max_active: <no_active_recovery_requests>
    osd_max_backfills: <max_no_backfills>
```

- OSD 클라이언트 OP 우선 순위를 기준으로 **<priority_value>** 를 복구 작업의 우선 순위로 바꿉니다.
- **<no_active_recovery_requests>** 를 한 번에 OSD당 활성 복구 요청 수로 바꿉니다.
- **<max_no_backfills>** 를 단일 OSD에 또는 단일 OSD에서 허용된 최대 백필 수로 바꿉니다. 기본 Red Hat Ceph Storage 백필 및 복구 옵션에 대한 자세한 내용은 Red Hat Ceph Storage 백필 및 복구 작업을 참조하십시오.

1.3.1. Compute 서비스를 예약하기 위해 CPU 및 메모리 리소스를 자동화하는 프로세스

director는 배포 중에 하이퍼컨버지드 노드에서 리소스 제약 조건을 구성하는 기본 계획 환경 파일을 제공합니다. 이 계획 환경 파일은 다음 프로세스를 완료하도록 OpenStack 워크플로우에 지시합니다.

1. 하드웨어 노드를 검사하는 동안 수집된 하드웨어 세부 검사 데이터를 검색합니다.
2. 해당 데이터를 기반으로 하이퍼컨버지드 노드에서 컴퓨팅에 대한 최적의 CPU 및 메모리 할당 워크로드를 계산합니다.
3. 이러한 제약 조건을 구성하고 Compute를 위해 CPU 및 메모리 리소스를 예약하는 데 필요한 매개 변수를 자동생성합니다. 이러한 매개 변수는 **plan-environment-derived-params.yaml** 파일의 the **hci_profile_config** 섹션에 정의되어 있습니다.

참고

각 워크로드 프로필의 **average_guest_memory_size_in _mb 및 average_guest_cpu_utilization_percentage** 매개 변수는 Compute의 **reserved_host_memory 및 cpu_ allocation_ratio** 설정 값을 계산하는 데 사용됩니다.

Compute 환경 파일에 다음 매개변수를 추가하여 자동 생성된 Compute 설정을 재정의할 수 있습니다.

nova.conf 매개변수 자동 생성	컴퓨팅 환경 파일 덮어쓰기	설명
reserved_host_memory	parameter_defaults: ComputeHCIParameters: NovaReservedHostMemory: 181000	Ceph OSD 서비스 및 하이퍼컨버지드 노드의 게스트당 인스턴스 오버헤드에 대해 예약해야 하는 RAM 양을 설정합니다.
cpu_allocation_ratio	parameter_defaults: ComputeHCIParameters: NovaCPUAllocationRatio: 8.2	인스턴스를 배포할 컴퓨팅 노드를 선택할 때 Compute 스케줄러에서 사용해야 하는 비율을 설정합니다.

이러한 재정의는 ComputeHCI 역할을 사용하는 모든 노드에 적용됩니다(즉, 모든 하이퍼컨버지드 노드). **NovaReservedHostMemory** 및 **NovaCPUAllocationRatio** 에 대한 최적 값을 수동으로 결정하는 방법에 대한 자세한 내용은 OpenStack Workflow Compute CPU 및 메모리 계산기 를 참조하십시오.

작은 정보

다음 스크립트를 사용하여 하이퍼컨버지드 노드에 적합한 **NovaReservedHostMemory** 및 **NovaCPUAllocationRatio** 값을 계산할 수 있습니다.

nova_mem_cpu_calc.py

추가 리소스

- 베어 메탈 노드 하드웨어의 인벤토리 생성

1.3.2. Red Hat Ceph Storage 백필 및 복구 작업

Ceph OSD가 제거되면 Red Hat Ceph Storage는 백필(backfill) 및 복구 작업을 사용하여 클러스터를 리밸런싱합니다. Red Hat Ceph Storage는 배치 그룹 정책에 따라 여러 데이터 복사본을 유지하기 위해 이 작업을 수행합니다. 이러한 작업은 시스템 리소스를 사용합니다. Red Hat Ceph Storage 클러스터가 로드 중인 경우 리소스가 백필(backfill) 및 복구에 따라 성능이 저하됩니다.

OSD를 제거하는 동안 이 성능 효과를 완화하려면 백필(backfill) 및 복구 작업의 우선 순위를 줄일 수 있습니다. 이에 대한 단점은 더 적은 데이터 복제본이 더 많다는 것입니다. 이로 인해 데이터가 약간 더 위험해집니다.

다음 표에 자세히 설명된 매개 변수는 백필(backfill) 및 복구 작업의 우선 순위를 구성하는 데 사용됩니다.

매개변수	설명	기본값
osd_recovery_op_priority	OSD 클라이언트 OP 우선 순위를 기준으로 복구 작업의 우선 순위를 설정합니다.	3
osd_recovery_max_active	한 번에 OSD당 활성 복구 요청 수를 설정합니다. 더 많은 요청으로 복구를 가속화하지만 요청에 따라 클러스터에서 부하가 증가합니다. 대기 시간을 줄이려면 이 값을 1로 설정합니다.	3

매개변수	설명	기본값
osd_max_backfills	단일 OSD에 또는 단일 OSD에서 허용된 최대 백필 수를 설정합니다.	1

1.4. 사용 가능한 RED HAT CEPH STORAGE 패키지 확인

오버클라우드 배포 실패를 방지하려면 서버에 필요한 패키지가 있는지 확인합니다.

1.4.1. ceph-ansible 패키지 버전 확인

언더클라우드에는 오버클라우드를 배포하기 전에 잠재적인 문제를 식별하기 위해 실행할 수 있는 Ansible 기반 검증이 포함되어 있습니다. 이러한 검증을 통해 일반적인 문제가 발생하기 전에 식별하여 오버클라우드 배포 실패를 방지할 수 있습니다.

절차

- 설치하려는 **ceph-ansible** 패키지 버전이 있는지 확인합니다.

```
$ ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/ansible/validation-playbooks/ceph-ansible-installed.yaml
```

1.4.2. 사전 프로비저닝된 노드의 패키지 확인

RHCS(Red Hat Ceph Storage)는 특정 패키지 세트가 있는 오버클라우드 노드만 서비스할 수 있습니다. 사전 프로비저닝된 노드를 사용하는 경우 이러한 패키지가 있는지 확인할 수 있습니다.

사전 프로비저닝된 노드에 대한 자세한 내용은 사전 프로비저닝된 노드 를 사용하여 기본 오버클라우드 구성을 참조하십시오.

절차

- 사전 프로비저닝된 노드에 필수 패키지가 포함되어 있는지 확인합니다.

```
ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/ansible/validation-playbooks/ceph-dependencies-installed.yaml
```

1.5. HCI 오버클라우드 배포

HCI 구성을 완료한 후 오버클라우드를 배포해야 합니다.

중요

RHOSP(Red Hat OpenStack Platform) HCI 환경을 배포할 때 인스턴스 HA를 활성화하지 마십시오. Red Hat Ceph Storage와 함께 하이퍼컨버지드 RHOSP 배포와 함께 인스턴스 HA를 사용하려면 Red Hat 담당자에게 문의하십시오.

사전 요구 사항

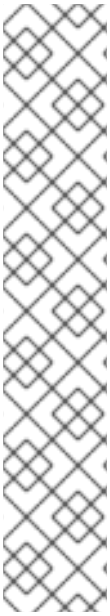
- 다른 모든 Red Hat Ceph Storage 설정(예: **/home/stack/templates/storage-config.yaml**)에 별도의 기본 환경 파일 또는 파일 집합을 사용하고 있습니다. 자세한 내용은 Customizing the Storage service and 부록 A. 샘플 환경 파일: Ceph Storage 클러스터 생성을 참조하십시오.
- 기본 환경 파일에서 각 역할에 할당할 노드 수를 정의했습니다. 자세한 내용은 노드 및 플레이버 역할에 할당을 참조하십시오.
- 언더클라우드를 설치하는 동안 **undercloud.conf** 파일에 **generate_service_certificate=false** 를 설정합니다. 그렇지 않으면 Overcloud Public Endpoint에서 SSL/TLS 활성화에 설명된 대로 오버클라우드를 배포할 때 신뢰 앵커를 삽입해야 합니다.

절차

- 다른 환경 파일을 사용하여 스택에 새 역할 및 환경 파일을 추가하고 HCI 오버클라우드를 배포합니다.

```
(undercloud)$ openstack overcloud deploy --templates \
  -e [your environment files] \
  -r /home/stack/templates/roles_data_hci.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
ansible.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/network-environment.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/net-single-nic-with-vlans.yaml
  -e /home/stack/templates/storage-config.yaml \
  -e /home/stack/templates/storage-container-config.yaml \
  -n /home/stack/templates/network_data.yaml \
  [-e /home/stack/templates/ceph-backfill-recovery.yaml \ ]
  --ntp-server pool.ntp.org
```

배포 명령에 **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml** 을 포함하여 컨테이너화된 Red Hat Ceph 클러스터를 배포하는 기본 환경 파일을 모든 기본 설정으로 추가합니다. 자세한 내용은 Deploying an Overcloud with Containerized Red Hat Ceph 를 참조하십시오.



참고

배포에서 단일 루트 입력/출력 가상화(SR-IOV)를 사용하는 경우 배포 명령에 다음 옵션을 포함합니다.

배포에서 ML2/OVS 메커니즘 드라이버를 사용하는 경우 다음 옵션을 지정합니다.

```
-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-
sriov.yaml
-e /home/stack/templates/network-environment.yaml
```

배포에서 ML2/OVN 메커니즘 드라이버를 사용하는 경우 다음 옵션을 지정합니다.

```
-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovn-
sriov.yaml
-e /home/stack/templates/network-environment.yaml
```

작은 정보

응답 파일을 사용하여 배포에 포함할 환경 파일을 지정할 수도 있습니다. 자세한 내용은 *Director 설치 및 사용* 가이드 의 오버클라우드 배포의 환경 파일 포함 을 참조하십시오.

1.5.1. ceph-anible 이 실행되는 노드 제한

ceph-anible 이 실행되는 노드를 제한하여 배포 업데이트 시간을 줄일 수 있습니다. RHOSP(Red Hat OpenStack Platform)에서 **config-download** 를 사용하여 Ceph를 구성하는 경우 전체 배포에서 **config-download 및 ceph- anible** 을 실행하는 대신 **--limit** 옵션을 사용하여 노드 목록을 지정할 수 있습니다. 이 기능은 예를 들어 오버클라우드를 확장하거나 실패한 디스크를 교체하는 데 유용합니다. 이러한 시나리오에서는 환경에 추가하는 새 노드에서만 배포를 실행할 수 있습니다.

실패한 디스크 교체에서 --limit 를 사용하는 시나리오의 예

다음 예제 절차에서 Ceph storage 노드 **oc0-cephstorage-0** 에는 디스크 오류가 있으므로 새 팩토리 클린 디스크를 받을 수 있습니다. 새 디스크를 OSD로 사용할 수 있도록 **oc0-cephstorage-0** 노드에서 Ansible을 실행해야 하지만 다른 모든 Ceph 스토리지 노드에서 실행할 필요는 없습니다. 환경 파일과 노드 이름을 환경에 적합한 이름으로 바꿉니다.

절차

1. **stack 사용자로 언더클라우드 노드에 로그인하고 stack rc** 자격 증명 파일을 가져옵니다.

```
# source stackrc
```

2. 새 디스크를 사용하여 누락된 OSD를 시작하도록 다음 단계 중 하나를 완료합니다.

- 스택 업데이트를 실행하고 **--limit** 옵션을 포함하여 **ceph-ansible** 을 실행할 노드를 지정합니다.

```
$ openstack overcloud deploy --templates \
  -r /home/stack/roles_data.yaml \
  -n /usr/share/openstack-tripleo-heat-templates/network_data_dashboard.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml \
  -e ~/my-ceph-settings.yaml \
  -e <other-environment_files> \
  --limit oc0-controller-0:oc0-controller-2:oc0-controller-1:oc0-cephstorage-0:undercloud
```

이 예에서는 Ceph mons에서 OSD 정의를 변경하는 데 Ansible이 필요하므로 컨트롤러가 포함됩니다.

- **config-download** 에서 **ansible-playbook-command.sh** 스크립트를 생성한 경우 **--limit** 옵션으로 스크립트를 실행하여 지정된 노드를 **ceph-ansible** 에 전달할 수도 있습니다.

```
./ansible-playbook-command.sh --limit oc0-controller-0:oc0-controller-2:oc0-controller-1:oc0-cephstorage-0:undercloud
```

경고

항상 제한 목록에 언더클라우드를 포함해야 합니다. 그러지 않으면 **--limit** 를 사용하는 경우 **ceph-anible** 을 실행할 수 없습니다. 이는 **ceph-ansible** 실행이 언더클라우드에서만 실행되는 **external_deploy_steps_tasks** 플레이북을 통해 발생하기 때문에 필요합니다.

1.6. OPENSTACK WORKFLOW COMPUTE CPU 및 메모리 계산기

OpenStack 워크플로는 CPU 및 메모리에 대한 최적의 설정을 계산하고 결과를 사용하여 **NovaReservedHostMemory 및 Nova CPUAllocationRatio** 매개 변수를 채웁니다.

NovaReservedHostMemory

NovaReservedHostMemory 매개 변수는 호스트 노드에 예약할 메모리 양(MB)을 설정합니다. 하이퍼 컨버지드 노드에 적절한 값을 결정하려면 각 OSD에서 3GB 메모리를 사용한다고 가정합니다. 256GB 메모리와 10개의 OSD가 있는 노드가 있는 경우 Ceph에 30GB 메모리를 할당할 수 있으며, Compute용으로 226GB를 남겨둘 수 있습니다. 이 크기의 메모리를 사용하면 노드에서 호스팅할 수 있습니다(예: 각각 2GB 메모리를 사용하는 인스턴스 113개).

그러나 여전히 *하이퍼바이저* 의 인스턴스당 추가 오버헤드를 고려해야 합니다. 이 오버헤드가 0.5GB라고 가정하면 동일한 노드에서 90개 인스턴스만 호스팅할 수 있으며, 이 인스턴스는 226GB를 2.5GB로 나눈 값입니다. 호스트 노드(즉, Compute 서비스에서 사용하지 않아야 하는 메모리)를 예약할 메모리 양은 다음과 같습니다.

(* Ov) + (Os * RA)

다음과 같습니다.

- **에서**: 인스턴스 수
- **ov**: 인스턴스당 필요한 오버헤드 메모리 양
- **OS**: 노드의 OSD 수
- **RA**: 각 OSD마다 보유해야 하는 RAM의 양

90개의 인스턴스를 사용할 경우 이 값은 (90*0.5) + (10*3) = 75GB입니다. 계산 서비스에서는 이 값이 MB, 즉 75000으로 예상합니다.

다음 Python 코드는 다음과 같은 계산을 제공합니다.

```
left_over_mem = mem - (GB_per_OSD * osds)
number_of_guests = int(left_over_mem /
    (average_guest_size + GB_overhead_per_guest))
nova_reserved_mem_MB = MB_per_GB * (
    (GB_per_OSD * osds) +
    (number_of_guests * GB_overhead_per_guest))
```

NovaCPUAllocationRatio

인스턴스를 배포할 컴퓨팅 노드를 선택할 때 계산 스케줄러는 **NovaCPUAllocationRatio** 를 사용합니다. 기본적으로 **16.0** (에서와 같이 16:1)입니다. 즉, 노드에 56개의 코어가 있으면 Compute 스케줄러는 노드에서 더 이상 호스팅할 수 없는 것으로 간주하기 전에 노드에서 896개의 vCPU를 사용할 수 있는 충분한 인스턴스를 예약합니다.

하이퍼 컨버지드 노드에 적합한 **NovaCPU AllocationRatio** 를 확인하려면 각 Ceph OSD에서 하나 이상의 코어를 사용한다고 가정합니다(워크로드가 I/O 사용량이 아닌 경우 SSD가 없는 노드에서). 56개의 코어와 10개의 OSD가 있는 노드에서는 Compute에 대해 46개의 코어가 남아 있습니다. 각 인스턴스에서 수신하는 CPU의 100%를 사용하는 경우, 이 비율은 인스턴스 vCPU 수를 코어 수로 나눈 값, 즉 46 / 56 = 이며. 그러나 인스턴스가 할당된 CPU의 100%를 사용하지 않으므로 필요한 게스트 vCPU 수를 결정할 때 예상되는 백분율을 고려하여 **NovaCPUAllocationRatio** 를 늘릴 수 있습니다.

따라서 인스턴스가 vCPU의 10%(또는 0.1)만 사용하도록 예측하면 인스턴스의 vCPU 수는 46/ 0.1 = 460으로 나타낼 수 있습니다. 이 값을 코어 수(56)로 나누면 비율은 약 8개로 증가합니다.

다음 Python 코드는 다음과 같은 계산을 제공합니다.

```
cores_per_OSD = 1.0
average_guest_util = 0.1 # 10%
nonceph_cores = cores - (cores_per_OSD * osds)
guest_vCPUs = nonceph_cores / average_guest_util
cpu_allocation_ratio = guest_vCPUs / cores
```

1.7. 추가 리소스

RHOSP(Red Hat OpenStack Platform)에 대한 자세한 내용은 다음 가이드를 참조하십시오.

- Director 설치 및 사용: 이 가이드에서는 Undercloud 및 Overcloud 둘 다의 RHOSP 환경의 포괄적인 배포에 대한 지침을 제공합니다.
- 고급 오버클라우드 사용자 지정: 이 가이드에서는 사용자 지정 역할 사용 방법과 같이 director를 통해 고급 RHOSP 기능을 구성하는 방법을 설명합니다.
- 컨테이너화된 Red Hat Ceph를 사용하여 오버클라우드 배포 : 이 가이드에서는 Red Hat Ceph Storage를 스토리지 프로바이더로 사용하는 오버클라우드를 배포하는 방법을 설명합니다.
- 네트워킹 가이드: 이 가이드에서는 RHOSP 네트워킹 작업에 대한 세부 정보를 제공합니다.

2장. 하이퍼컨버지드 노드 확장

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

2.1. HCI 환경에서 하이퍼컨버지드 노드 확장

HCI 환경에서 하이퍼컨버지드 노드를 확장하려면 하이퍼바이저가 아닌 노드를 확장하는 것과 동일한 절차를 따릅니다. 자세한 내용은 오버클라우드에 노드 추가를 참조하십시오.

참고

새 노드에 태그를 지정하는 경우 올바른 플레이버를 사용해야 합니다.

Red Hat Ceph Storage 클러스터에 OSD를 추가하여 HCI 노드를 확장하는 방법에 대한 자세한 내용은 *Deploying an Overcloud with Containerized Red Hat Ceph에서 Ceph Storage 노드에 OSD 추가를* 참조하십시오.

2.2. HCI 환경에서 하이퍼컨버지드 노드 축소

HCI 환경에서 하이퍼컨버지드 노드를 축소하려면 HCI 노드에서 Ceph OSD 서비스를 리밸런스하고, HCI 노드에서 인스턴스를 마이그레이션하고, 오버클라우드에서 Compute 노드를 제거해야 합니다.

절차

1. HCI 노드에서 Ceph OSD 서비스를 비활성화하고 리밸런스합니다. 이 단계는 HCI 또는 Red Hat Ceph Storage 노드를 제거할 때 director가 Red Hat Ceph Storage 클러스터를 자동으로 재조정하지 않기 때문에 필요합니다.

2. HCI 노드에서 인스턴스를 마이그레이션합니다. 자세한 내용은 *Configuring the Compute Service for Instance Creation* 가이드의 Compute 노드 간 가상 머신 마이그레이션 을 참조하십시오.

3. 오버클라우드에서 컴퓨팅 노드를 제거합니다. 자세한 내용은 Compute 노드 제거를 참조하십시오.