



Red Hat Enterprise Linux 9

Google Cloud Platform에 RHEL 9 배포

GCP에서 RHEL 시스템 이미지 가져오기 및 RHEL 인스턴스 생성

Red Hat Enterprise Linux 9 Google Cloud Platform에 RHEL 9 배포

GCP에서 RHEL 시스템 이미지 가져오기 및 RHEL 인스턴스 생성

법적 공지

Copyright © 2023 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

퍼블릭 클라우드 환경에서 RHEL(Red Hat Enterprise Linux)을 사용하려면 GCP(Google Cloud Platform)를 비롯한 다양한 클라우드 플랫폼에 RHEL 시스템 이미지를 생성하고 배포할 수 있습니다. GCP에서 Red Hat HA(고가용성) 클러스터를 생성하고 구성할 수도 있습니다. 다음 장에서는 GCP에서 클라우드 RHEL 인스턴스 및 HA 클러스터를 생성하는 방법을 설명합니다. 이러한 프로세스에는 필수 패키지 및 에이전트 설치, 펜싱 구성 및 네트워크 리소스 에이전트 설치가 포함됩니다.

차례

RHEL 베타 버전 릴리스	3
보다 포괄적 수용을 위한 오픈 소스 용어 교체	4
RED HAT 문서에 관한 피드백 제공	5
1장. 베타의 새로운 기능	6
2장. 이미지 빌더를 사용하여 GCP에 이미지 업로드	7
2.1. CLI를 사용하여 GCP에 GCE 이미지 업로드	7
2.2. GCP로 인증	8
3장. GOOGLE CLOUD PLATFORM에서 RED HAT ENTERPRISE LINUX 이미지를 GOOGLE COMPUTE ENGINE 인스턴스로 배포	10
3.1. GCP의 RED HAT ENTERPRISE LINUX 이미지 옵션	10
3.2. 기본 이미지 이해	12
3.3. ISO 이미지에서 기본 VM 생성	12
3.4. GCP에 RHEL 이미지 업로드	14
3.5. 추가 리소스	20
4장. GOOGLE CLOUD PLATFORM에서 RED HAT HIGH AVAILABILITY CLUSTER 구성	21
4.1. 필수 시스템 패키지	21
4.2. GCP의 RED HAT ENTERPRISE LINUX 이미지 옵션	22
4.3. GOOGLE CLOUD SDK 설치	23
4.4. GCP 이미지 버킷 생성	24
4.5. 사용자 지정 가상 프라이빗 클라우드 네트워크 및 서브넷 생성	24
4.6. 기본 GCP 이미지 준비 및 가져오기	25
4.7. 기본 GCP 인스턴스 생성 및 구성	25
4.8. 스냅샷 이미지 생성	28
4.9. HA 노드 템플릿 인스턴스 및 HA 노드 생성	28
4.10. HA 패키지 및 에이전트 설치	29
4.11. HA 서비스 구성	30
4.12. 클러스터 생성	30
4.13. 펜싱 장치 생성	31
4.14. GCP 노드 권한 부여 구성	32
4.15. GCP-VCP-MOVE-VIP 리소스 에이전트 구성	32
4.16. 추가 리소스	35

RHEL 베타 버전 릴리스

Red Hat은 Red Hat Enterprise Linux 베타 버전을 통해 등록된 모든 Red Hat 계정에 액세스할 수 있습니다. 베타 버전 액세스의 목적은 다음과 같습니다.

- 고객에게 일반 가용성 릴리스 전에 주요 기능을 테스트하고 피드백을 제공하거나 문제를 보고할 수 있는 기회를 제공합니다.
- 베타 제품 설명서를 미리 보기로 제공합니다. 베타 제품 문서는 개발 중이며 상당한 변경이 있을 수 있습니다.

Red Hat은 프로덕션 사용 사례에서는 RHEL 베타 버전 릴리스를 지원하지 않습니다. 자세한 내용은 [Red Hat Enterprise Linux에서 베타 버전의 의미는 무엇이며 RHEL 베타 설치를 GA\(General Availability\) 릴리스로 업그레이드할 수 있습니까?](#)에서 참조하십시오.

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 의견에 감사드립니다. 어떻게 개선할 수 있는지 알려주십시오.

특정 문구에 대한 의견 제출

1. **Multi-page HTML** 형식으로 설명서를 보고 페이지가 완전히 로드된 후 오른쪽 상단 모서리에 **피드백** 버튼이 표시되는지 확인합니다.
2. 커서를 사용하여 주석 처리할 텍스트 부분을 강조 표시합니다.
3. 강조 표시된 텍스트 옆에 표시되는 **피드백 추가** 버튼을 클릭합니다.
4. 의견을 추가하고 **제출**을 클릭합니다.

Bugzilla를 통해 피드백 제출(등록 필요)

1. [Bugzilla](#) 웹 사이트에 로그인합니다.
2. **버전** 메뉴에서 올바른 버전을 선택합니다.
3. **요약** 필드에 설명 제목을 입력합니다.
4. **설명** 필드에 개선을 위한 제안을 입력합니다. 문서의 관련 부분에 대한 링크를 포함합니다.
5. **버그 제출**을 클릭합니다.

1장. 베타의 새로운 기능

보다 집중된 정보 및 개선된 사용자 환경을 제공하기 위해 [공용 클라우드 플랫폼에 RHEL 9 배포](#) 문서는 [특정 클라우드 환경에 배포하는 데](#) 전하는 여러 개의 문서로 나뉘어져 있습니다.

이 변화가 좋은지 여부에 대한 귀하의 의견을 환영합니다. 자세한 내용은 [Red Hat 문서에 대한 피드백](#) 제 공을 참조하십시오.

2장. 이미지 빌더를 사용하여 GCP에 이미지 업로드

이미지 빌더를 사용하면 **gce** 이미지를 빌드하고 사용자 또는 GCP 서비스 계정의 인증 정보를 제공한 다음 **gce** 이미지를 GCP 환경에 직접 업로드할 수 있습니다.

2.1. CLI를 사용하여 GCP에 GCE 이미지 업로드

절차에 따라 자격 증명이 포함된 구성 파일을 설정하여 **gce** 이미지를 GCP에 업로드합니다.

사전 요구 사항

- GCP에 이미지를 업로드할 수 있는 사용자 또는 서비스 계정 Google 자격 증명이 있습니다. 인증 정보와 연결된 계정에는 최소한 다음 IAM 역할이 할당되어 있어야 합니다.
 - **roles/storage.admin** - 스토리지 오브젝트를 생성하고 삭제합니다.
 - **roles/compute.storageAdmin** - VM 이미지를 Compute Engine으로 가져오려면 다음을 수행합니다.
- 기존 GCP 버킷이 있습니다.

절차

1. 텍스트 편집기를 사용하여 다음 콘텐츠로 **gcp-config.toml** 구성 파일을 생성합니다.

```
provider = "gcp"

[settings]
bucket = "GCP_BUCKET"
region = "GCP_STORAGE_REGION"
object = "OBJECT_KEY"
credentials = "GCP_CREDENTIALS"
```

다음과 같습니다.

- **GCP_BUCKET** 은 기존 버킷을 가리킵니다. 업로드 중인 이미지의 중간 스토리지 오브젝트를 저장하는 데 사용됩니다.
- **GCP_STORAGE_REGION** 은 일반 Google 스토리지 리전과 듀얼 또는 다중 리전 모두입니다.
- **OBJECT_KEY** 는 중간 스토리지 오브젝트의 이름입니다. 업로드하기 전에 존재하지 않아야 하며 업로드 프로세스가 완료되면 삭제됩니다. 개체 이름이 **.tar.gz** 로 끝나지 않으면 확장 이 개체 이름에 자동으로 추가됩니다.
- **GCP_CREDENTIALS** 는 GCP에서 다운로드한 자격 증명 JSON 파일의 Base64 인코딩 스키마입니다. 인증 정보는 GCP가 이미지를 업로드하는 프로젝트를 결정합니다.



참고

다른 메커니즘을 사용하여 GCP로 인증하는 경우 **gcp-config.toml** 에 **GCP_CREDENTIALS** 를 지정하는 것은 선택 사항입니다. GCP로 인증하는 다양한 방법에 대한 자세한 내용은 [GCP로 인증](#) 을 참조하십시오.

2. 추가 이미지 이름 및 클라우드 공급자 프로필을 사용하여 작성을 생성합니다.

```
$ sudo composer-cli compose start BLUEPRINT-NAME gce IMAGE_KEY gcp-config.toml
```

참고: 이미지 빌드, 업로드 및 클라우드 등록 프로세스는 완료하는 데 최대 10분이 걸릴 수 있습니다.

검증

- 이미지 상태가 FINISHED인지 확인합니다.

```
$ sudo composer-cli compose status
```

추가 리소스

- [ID 및 액세스 관리](#).
- [스토리지 버킷을 생성합니다](#).

2.2. GCP로 인증

이미지 빌더와 함께 여러 가지 유형의 인증 정보를 사용하여 GCP로 인증할 수 있습니다. 이미지 빌더 구성이 여러 인증 정보를 사용하여 GCP에서 인증하도록 설정된 경우 다음과 같은 기본 설정 순서로 인증 정보를 사용합니다.

1. 구성 파일에서 **composer-cli** 명령으로 지정된 인증 정보입니다.
2. **osbuild-composer** 작업자 구성에 구성된 인증 정보
3. 다음 옵션을 사용하여 인증할 방법을 자동으로 찾으려고 하는 **Google GCP SDK** 라이브러리의 애플리케이션 기본 자격 증명:
 - a. **GOOGLE_APPLICATION_CREDENTIALS** 환경 변수가 설정된 경우 Application Default Credentials는 변수가 가리키는 파일에서 인증 정보를 로드하고 사용하려고 합니다.
 - b. 애플리케이션 기본 자격 증명은 코드를 실행하는 리소스에 연결된 서비스 계정을 사용하여 인증을 시도합니다. 예를 들면 Google Compute Engine VM입니다.



참고

GCP 인증 정보를 사용하여 이미지를 업로드할 GCP 프로젝트를 결정해야 합니다. 따라서 모든 이미지를 동일한 GCP 프로젝트에 업로드하지 않으려면 **Composer-cli** 명령으로 **gcp-config.toml** 구성 파일에 인증 정보를 지정해야 합니다.

2.2.1. composer-cli 명령으로 인증 정보 지정

제공된 업로드 대상 설정 **gcp-config.toml** 에서 GCP 인증 자격 증명을 지정할 수 있습니다. **Base64**로 인코딩된 Google 계정 자격 증명 JSON 파일의 스키마를 사용하여 시간을 절약합니다.

절차

- 제공된 업로드 대상 구성 **gcp-config.toml** 에서 자격 증명을 설정합니다.

```

provider = "gcp"

[settings]
provider = "gcp"

[settings]
...
credentials = "GCP_CREDENTIALS"

```

- **GOOGLE_APPLICATION_CREDENTIALS** 환경 변수에 저장된 경로를 사용하여 Google 계정 자격 증명 파일의 인코딩된 콘텐츠를 가져오려면 다음 명령을 실행합니다.

```
$ base64 -w 0 "${GOOGLE_APPLICATION_CREDENTIALS}"
```

2.2.2. osbuild-composer 작업자 구성에서 인증 정보 지정

모든 이미지 빌드에 전역적으로 GCP에 사용하도록 GCP 인증 자격 증명을 구성할 수 있습니다. 이렇게 하면 동일한 GCP 프로젝트로 이미지를 가져오려면 모든 이미지 업로드에 동일한 인증 정보를 사용할 수 있습니다.

절차

- **/etc/osbuild-worker/osbuild-worker.toml** 작업자 구성에서 다음 인증 정보 값을 설정합니다.

```

[gcp]
credentials = "PATH_TO_GCP_ACCOUNT_CREDENTIALS"

```

3장. GOOGLE CLOUD PLATFORM에서 RED HAT ENTERPRISE LINUX 이미지를 GOOGLE COMPUTE ENGINE 인스턴스로 배포

GCP(Google Cloud Platform)에서 GCE(Google Compute Engine) 인스턴스로 RHEL 9(Red Hat Enterprise Linux 9)를 배포하려면 다음 정보를 따르십시오. 이 장에서는 다음을 수행합니다.

- 이미지 선택 옵션에 대해 설명합니다.
- 호스트 시스템 및 VM(가상 머신)의 시스템 요구 사항을 나열하거나 참조합니다.
- ISO 이미지에서 사용자 지정 VM을 생성하고 GCE에 업로드 및 인스턴스 시작 절차를 제공합니다.



참고

GCP용 Red Hat 제품 인증 목록은 [Google Cloud Platform에서 Red Hat](#) 을 참조하십시오.



중요

ISO 이미지에서 사용자 지정 VM을 생성할 수 있지만, *Red Hat Image Builder* 제품을 사용하여 특정 클라우드 공급자에서 사용할 사용자 지정 이미지를 생성하는 것이 좋습니다. 자세한 내용은 [사용자 지정된 RHEL 시스템 이미지 구성](#) 을 참조하십시오.

사전 요구 사항

- 이 장의 절차를 완료하려면 [Red Hat 고객 포털](#) 계정이 필요합니다.
- GCP가 있는 계정을 생성하여 Google Cloud Platform 콘솔에 액세스합니다. 자세한 내용은 [Google Cloud](#) 를 참조하십시오.
- Red Hat [Cloud Access 프로그램](#) 을 통해 [Red Hat](#) 서브스크립션을 활성화합니다. Red Hat Cloud Access 프로그램을 사용하면 Red Hat 서브스크립션을 물리적 또는 온프레미스 시스템에서 GCP로 이전할 수 있습니다.

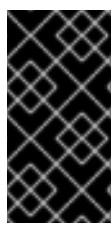
3.1. GCP의 RED HAT ENTERPRISE LINUX 이미지 옵션

다음 표에는 Google Cloud Platform의 RHEL 9에 대한 이미지 선택 사항 및 이미지 옵션의 차이점이 나열되어 있습니다.

표 3.1. 이미지 옵션

이미지 옵션	서브스크립션	샘플 시나리오	고려 사항
--------	--------	---------	-------

이미지 옵션	서브스크립션	샘플 시나리오	고려 사항
Red Hat Gold Image를 배포하도록 선택합니다.	기존 Red Hat 서브스크립션을 사용합니다.	Red Hat Cloud Access 프로그램 을 통해 서브스크립션을 활성화한 다음 Google Cloud Platform에서 Red Hat Gold Image를 선택합니다. 골드 이미지 및 Google Cloud Platform에서 액세스하는 방법에 대한 자세한 내용은 Red Hat Cloud Access 참조 가이드 를 참조하십시오.	서브스크립션에는 Red Hat 제품 비용이 포함되며 다른 모든 인스턴스 비용으로 Google에 비용을 지불하는 것이 좋습니다. Red Hat Gold Image는 기존 Red Hat 서브스크립션을 사용하므로 "Cloud Access" 이미지라고 합니다. Red Hat은 Cloud Access 이미지에 대한 직접 지원을 제공합니다.
GCP로 이동하는 사용자 정의 이미지를 배포하도록 선택합니다.	기존 Red Hat 서브스크립션을 사용합니다.	Red Hat Cloud Access 프로그램 을 통해 서브스크립션을 활성화하고 사용자 지정 이미지를 업로드하고 서브스크립션을 첨부하십시오.	서브스크립션에는 Red Hat 제품 비용이 포함되며 다른 모든 인스턴스 비용을 절감할 수 있습니다. 기존 Red Hat 서브스크립션을 사용하므로 GCP로 이동하는 사용자 지정 이미지를 "Cloud Access" 이미지라고 합니다. Red Hat은 Cloud Access 이미지에 대한 직접 지원을 제공합니다.
RHEL이 포함된 기존 GCP 이미지를 배포하도록 선택합니다.	GCP 이미지에는 Red Hat 제품이 포함되어 있습니다.	GCP Compute Engine 에서 인스턴스를 시작할 때 RHEL 이미지를 선택하거나 Google Cloud Platform Marketplace 에서 이미지를 선택합니다.	사용량에 따라 사용량 기준(Pay-as-you-go) 모델로 GCP를 매시간 결제합니다. 이러한 이미지를 "온 디맨드" 이미지라고 합니다. GCP는 지원 계약을 통해 주문형 이미지를 지원합니다.



중요

온디맨드 인스턴스를 Red Hat Cloud Access 인스턴스로 변환할 수 없습니다. 온 디맨드 이미지에서 BYOS(your-own-subscription) 이미지로 변경하려면 새 Red Hat Cloud Access 인스턴스를 생성하고 온 디맨드 인스턴스에서 데이터를 마이그레이션합니다. 데이터를 마이그레이션한 후 이중 청구를 방지하기 위해 온 디맨드 인스턴스를 취소합니다.

다음 섹션에서는 사용자 지정 이미지와 관련된 정보 및 절차를 제공합니다.

추가 리소스

- [퍼블릭 클라우드의 Red Hat](#)

- [컴퓨팅 엔진 이미지](#)
- [Red Hat Cloud Access 참조 가이드](#)
- [사용자 정의 이미지에서 인스턴스 생성](#)

3.2. 기본 이미지 이해

이 섹션에는 사전 구성된 기본 이미지 및 해당 구성 설정 사용에 대한 정보가 포함되어 있습니다.

3.2.1. 사용자 정의 기본 이미지 사용

VM(가상 시스템)을 수동으로 구성하려면 먼저 기본(시작자) VM 이미지를 생성합니다. 다음으로 구성 설정을 수정하고 VM이 클라우드에서 작동하는 데 필요한 패키지를 추가할 수 있습니다. 이미지를 업로드한 후 특정 애플리케이션에 대한 추가 구성을 변경할 수 있습니다.

추가 리소스

- [Red Hat Enterprise Linux](#)

3.2.2. 가상 머신 구성 설정

클라우드 VM에는 다음과 같은 구성 설정이 있어야 합니다.

표 3.2. VM 구성 설정

설정	권장 사항
ssh	VM에 대한 원격 액세스를 제공하려면 SSH를 활성화해야 합니다.
dhcp	기본 가상 어댑터는 dhcp에 대해 구성해야 합니다.

3.3. ISO 이미지에서 기본 VM 생성

이 섹션의 절차에 따라 ISO 이미지에서 RHEL 9 기본 이미지를 생성합니다.

사전 요구 사항

- 호스트 머신에서 [가상화가 활성화](#)되어 있습니다.
- Red Hat [고객 포털](#)에서 최신 Red Hat Enterprise Linux ISO 이미지를 다운로드하여 이미지를 `/var/lib/libvirt/images` 로 이동했습니다.

3.3.1. RHEL ISO 이미지에서 VM 생성

절차

1. 가상화를 위해 호스트 시스템을 활성화했는지 확인합니다. 자세한 내용은 [RHEL 9에서 가상화 활성화](#)를 참조하십시오.

2. 기본 Red Hat Enterprise Linux VM을 만들고 시작합니다. 자세한 내용은 [가상 머신 생성](#)을 참조하십시오.
 - a. 명령줄을 사용하여 VM을 생성하는 경우 기본 메모리 및 CPU를 VM에 필요한 용량으로 설정해야 합니다. 가상 네트워크 인터페이스를 **virtio**로 설정합니다.
예를 들어 다음 명령은 **/home/username/Downloads/rhel9.iso** 이미지를 사용하여 **kvmtest** VM을 생성합니다.

```
# virt-install \
--name kvmtest --memory 2048 --vcpus 2 \
--cdrom /home/username/Downloads/rhel9.iso,bus=virtio \
--os-variant=rhel9.0
```

- b. 웹 콘솔을 사용하여 VM을 생성하는 경우 다음 주의 사항과 함께 [웹 콘솔을 사용하여 가상 머신 생성](#) 절차를 따르십시오.
 - 즉시 **VM 시작**을 선택하지 마십시오.
 - **메모리 크기**를 원하는 설정으로 변경합니다.
 - 설치를 시작하기 전에 [가상 네트워크 인터페이스 설정](#)에서 **모델**을 **virtio**로 변경하고 **vCPU**를 VM에 필요한 용량 설정으로 변경했는지 확인합니다.

3.3.2. RHEL 설치 완료

다음 단계를 수행하여 설치를 완료하고 VM이 시작되면 루트 액세스를 활성화합니다.

절차

1. 설치 프로세스 중에 사용할 언어를 선택합니다.
2. **설치 요약** 보기에서 다음을 실행합니다.
 - a. **소프트웨어 선택** 사항을 클릭하고 **최소 설치**를 확인합니다.
 - b. **Done** 을 클릭합니다.
 - c. **Installation Destination** (설치 대상)을 클릭하고 스토리지 구성에서 **Custom** (사용자 지정)을 선택합니다.
 - **/boot**에 대해 최소 500MB인지 확인합니다. root에 남은 공간을 사용할 수 있습니다. **/**.
 - 표준 파티션은 권장되지만 LVM(Logical Volume Management)을 사용할 수 있습니다.
 - 파일 시스템에 xfs, ext4 또는 ext3을 사용할 수 있습니다.
 - 변경 사항이 완료되면 **완료**를 클릭합니다.
3. **설치 시작**을 클릭합니다.
4. **Root 암호**를 설정합니다. 해당되는 경우 다른 사용자를 만듭니다.
5. VM을 재부팅하고 설치가 완료되면 **root**로 로그인합니다.
6. 이미지를 구성합니다.
 - a. VM을 등록하고 Red Hat Enterprise Linux 9 리포지토리를 활성화합니다.

```
# subscription-manager register --auto-attach
```

b. **cloud-init** 패키지가 설치되어 활성화되어 있는지 확인합니다.

```
# dnf install cloud-init
# systemctl enable --now cloud-init.service
```

7. VM의 전원을 끕니다.

추가 리소스

- [고객 포털에서 자동 연결 서브스크립션 이해](#)
- [cloud-init 소개](#)

3.4. GCP에 RHEL 이미지 업로드

RHEL 9 이미지를 GCP(Google Cloud Platform)에 업로드하려면 이 섹션의 절차를 따르십시오.

3.4.1. GCP에서 새 프로젝트 생성

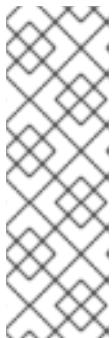
GCP(Google Cloud Platform)에 새 프로젝트를 생성하려면 다음 단계를 완료합니다.

사전 요구 사항

- GCP 계정이 있어야 합니다. 자세한 내용은 [Google Cloud](#) 를 참조하십시오.

절차

1. [GCP 콘솔](#) 을 시작합니다.
2. **Google Cloud Platform** 오른쪽에 있는 드롭다운 메뉴를 클릭합니다.
3. 팝업 메뉴에서 새 프로젝트를 **클릭**합니다.
4. **새 프로젝트** 창에서 새 프로젝트의 이름을 입력합니다.
5. 조직 확인 . 필요한 경우 드롭다운 메뉴를 클릭하여 조직을 변경합니다.
6. 상위 조직 또는 폴더의 위치를 확인합니다. 필요한 경우 **찾아보기**를 클릭하여 이 값을 검색하고 **변경**합니다.
7. **CREATE** 를 클릭하여 새 **GCP** 프로젝트를 생성합니다.



참고

Google Cloud SDK를 설치한 후에는 **gcloud** 프로젝트 **create** CLI 명령을 사용하여 프로젝트를 생성할 수 있습니다. 예를 들면 다음과 같습니다.

```
# gcloud projects create my-gcp-project3 --name project3
```

이 예제에서는 프로젝트 ID **my-gcp-project3** 및 프로젝트 이름 **project3** 을 사용하여 프로젝트를 생성합니다. 자세한 내용은 [gcloud project create](#) 를 참조하십시오.

추가 리소스

- [Google Cloud에서 리소스 생성 및 관리](#)

3.4.2. Google Cloud SDK 설치

Google Cloud SDK를 설치하려면 다음 단계를 완료하십시오.

절차

1. Google Cloud SDK 아카이브를 다운로드하고 추출하기 위한 GCP 지침을 따르십시오. 자세한 내용은 GCP 문서 [빠른 시작 Linux](#)를 참조하십시오.
2. Google Cloud SDK를 초기화하는 것과 동일한 지침을 따르십시오.



참고

Google Cloud SDK를 초기화한 후에는 **gcloud** CLI 명령을 사용하여 작업을 수행하고 프로젝트 및 인스턴스에 대한 정보를 얻을 수 있습니다. 예를 들어 **gcloud compute project-info describe --project <project-name>** 명령을 사용하여 프로젝트 정보를 표시할 수 있습니다.

추가 리소스

- [Linux 빠른 시작](#)
- [gcloud 명령 참조](#)
- [gcloud 명령줄 톨 개요](#)

3.4.3. Google Compute Engine용 SSH 키 생성

다음 절차에 따라 공용 IP 주소를 사용하여 인스턴스에 직접 SSH 연결을 시작할 수 있도록 GCE를 사용하여 SSH 키를 생성하고 등록합니다.

절차

1. **ssh-keygen** 명령을 사용하여 GCE와 함께 사용할 SSH 키 쌍을 생성합니다.

```
# ssh-keygen -t rsa -f ~/.ssh/google_compute_engine
```

2. [GCP 콘솔 대시보드 페이지에서 GoogleCloud Console 배너](#) 왼쪽에 있는 탐색 메뉴를 클릭하고 **Compute Engine** 을 선택한 다음 **메타데이터** 를 선택합니다.
3. SSH 키를 클릭한 다음 **편집** 을 클릭합니다.
4. **~/.ssh/google_compute_engine.pub** 파일에서 생성된 출력을 입력하고 **저장** 을 클릭합니다. 이제 표준 SSH를 사용하여 인스턴스에 연결할 수 있습니다.

```
# ssh -i ~/.ssh/google_compute_engine <username>@<instance_external_ip>
```



참고

gcloud compute config-ssh 명령을 실행하여 구성 파일을 인스턴스의 별칭으로 채울 수 있습니다. 별칭을 사용하면 인스턴스 이름으로 간단한 SSH 연결을 수행할 수 있습니다. **gcloud compute config-ssh** 명령에 대한 자세한 내용은 [gcloud compute config-ssh](#) 를 참조하십시오.

추가 리소스

- [gcloud compute config-ssh](#)
- [인스턴스에 연결](#)

3.4.4. GCP Storage에서 스토리지 버킷 생성

GCP로 가져오려면 GCP 스토리지 버킷이 필요합니다. 버킷을 생성하려면 다음 단계를 완료합니다.

절차

1. GCP에 아직 로그인하지 않은 경우 다음 명령을 사용하여 로그인합니다.

```
# gcloud auth login
```

2. 스토리지 버킷을 생성합니다.

```
# gsutil mb gs://bucket_name
```



참고

또는 [Google Cloud Console](#)을 사용하여 버킷을 만들 수도 있습니다. 자세한 내용은 [버킷 만들기](#) 를 참조하십시오.

추가 리소스

- [버킷 생성](#)

3.4.5. 이미지를 GCP Bucket에 변환 및 업로드

이미지를 변환하고 GCP Bucket에 업로드하려면 다음 절차를 완료합니다. 샘플은 대표적인 것입니다. **qcow2** 이미지를 원시 형식으로 변환한 다음 업로드를 위해 해당 이미지를 tar로 변환합니다.

절차

1. **qemu-img** 명령을 실행하여 이미지를 변환합니다. 변환된 이미지에는 **disk.raw** 이름이 있어야 합니다.

```
# qemu-img convert -f qcow2 -O raw rhel-{ProductNumber}.0-sample.qcow2 disk.raw
```

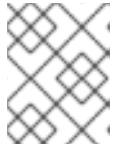
2. 이미지를 tar로 지정합니다.

```
# tar --format=oldgnu -Sczf disk.raw.tar.gz disk.raw
```

3. 이전에 만든 버킷에 이미지를 업로드합니다. 업로드에 몇 분 정도 걸릴 수 있습니다.

```
# gsutil cp disk.raw.tar.gz gs://bucket_name
```

4. Google Cloud Platform 홈 화면에서 접힌 메뉴 아이콘을 클릭하고 Storage 를 선택한 다음 CloudEvent를 선택합니다.
5. 버킷 이름을 클릭합니다.
tarred 이미지는 버킷 이름 아래에 나열됩니다.



참고

GCP 콘솔을 사용하여 이미지를 업로드할 수도 있습니다. 이렇게 하려면 버킷의 이름을 클릭한 다음 파일 업로드 를 클릭합니다.

추가 리소스

- [수동으로 가상 디스크 가져오기](#)
- [가져오기 방법 선택](#)

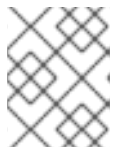
3.4.6. GCP 버킷의 오브젝트에서 이미지 생성

다음 절차에 따라 GCP 버킷의 오브젝트에서 이미지를 생성합니다.

절차

1. 다음 명령을 실행하여 GCE의 이미지를 생성합니다. 생성 중인 이미지의 이름, 버킷 이름 및 tarred 이미지의 이름을 지정합니다.

```
# gcloud compute images create my-image-name --source-uri gs://my-bucket-name/disk.raw.tar.gz
```



참고

또는 Google Cloud Console을 사용하여 이미지를 만들 수도 있습니다. 자세한 내용은 [사용자 정의 이미지 생성, 삭제 및 사용 중단](#)을 참조하십시오.

2. 필요한 경우 GCP 콘솔에서 이미지를 찾습니다.
 - a. Google Cloud Console 배너의 왼쪽에 있는 탐색 메뉴를 클릭합니다.
 - b. Compute Engine 을 선택한 다음 이미지를 선택합니다.

추가 리소스

- [사용자 정의 이미지 생성, 삭제 및 사용 중단](#)
- [gcloud 컴퓨팅 이미지 생성](#)

3.4.7. 이미지에서 Google Compute Engine 인스턴스 생성

GCP 콘솔을 사용하여 GCE VM 인스턴스를 구성하려면 다음 단계를 완료합니다.



참고

다음 절차에서는 GCP 콘솔을 사용하여 기본 VM 인스턴스를 생성하는 방법을 설명합니다. [GCE VM 인스턴스 및 구성 옵션에 대한 자세한 내용은 VM 인스턴스 생성 및 시작을 참조하십시오.](#)

절차

1. [GCP 콘솔 대시보드 페이지에서 GoogleCloud Console 배너](#) 왼쪽에 있는 탐색 메뉴를 클릭하고 **Compute Engine** 을 선택한 다음 이미지를 선택합니다.
2. 이미지를 선택합니다.
3. 인스턴스 생성을 클릭합니다.
4. 인스턴스 생성 페이지에서 인스턴스의 이름을 입력합니다.
5. 지역 및 영역을 선택합니다.
6. 워크로드 요구 사항을 충족하거나 초과하는 머신 구성을 선택합니다.
7. 부팅 디스크가 이미지 이름을 지정하는지 확인합니다.
8. 선택적으로 방화벽에서 HTTP 트래픽 허용 또는 HTTPS 트래픽 허용을 선택합니다.
9. 생성을 클릭합니다.



참고

기본 인스턴스를 생성하는 데 필요한 최소 구성 옵션입니다. 애플리케이션 요구 사항에 따라 추가 옵션을 검토합니다.

10. VM 인스턴스에서 이미지를 찾습니다.
11. [GCP 콘솔 대시보드에서 Google Cloud Console 배너](#) 왼쪽의 탐색 메뉴를 클릭하고 **Compute Engine** 을 선택한 다음 VM 인스턴스를 선택합니다.



참고

또는 `gcloud compute instances create` CLI 명령을 사용하여 이미지에서 GCE VM 인스턴스를 생성할 수도 있습니다. 간단한 예는 다음과 같습니다.

```
gcloud compute instances create myinstance3 --zone=us-central1-a --image test-iso2-image
```

이 예제에서는 기존 이미지 `test-iso2-image` 에 따라 `us-central1-a` 영역에 `myinstance3` 이라는 VM 인스턴스를 생성합니다. 자세한 내용은 [gcloud compute 인스턴스 생성](#) 을 참조하십시오.

3.4.8. 인스턴스에 연결

공용 IP 주소를 사용하여 GCE 인스턴스에 연결하려면 다음 절차를 수행합니다.

절차

1. 다음 명령을 실행하여 인스턴스가 실행 중인지 확인합니다. 이 명령은 인스턴스가 실행 중인지 여부 및 실행 중인 인스턴스의 공용 IP 주소를 포함하여 GCE 인스턴스에 대한 정보를 나열합니다.

```
# gcloud compute instances list
```

2. 표준 SSH를 사용하여 인스턴스에 연결합니다. 이 예제에서는 이전에 만든 `google_compute_engine` 키를 사용합니다.

```
# ssh -i ~/.ssh/google_compute_engine <user_name>@<instance_external_ip>
```



참고

GCP는 인스턴스에 SSH 연결을 수행하는 다양한 방법을 제공합니다. [자세한 내용은 인스턴스에 연결](#)을 참조하십시오. 이전에 설정한 암호와 루트 계정을 사용하여 인스턴스에 연결할 수도 있습니다.

추가 리소스

- [gcloud 컴퓨팅 인스턴스 목록](#)
- [인스턴스에 연결](#)

3.4.9. Red Hat 서브스크립션 첨부

Red Hat Cloud Access 프로그램을 통해 이전에 활성화한 서브스크립션을 연결하려면 다음 단계를 완료합니다.

사전 요구 사항

- 서브스크립션을 활성화해야 합니다.

절차

1. 시스템을 등록합니다.

```
# subscription-manager register --auto-attach
```

2. 서브스크립션을 첨부합니다.

- 활성화 키를 사용하여 서브스크립션을 연결할 수 있습니다. 자세한 내용은 [Red Hat 고객 포털 활성화 키 생성](#)을 참조하십시오.
- 또는 서브스크립션 풀 ID(Pool ID)를 사용하여 서브스크립션을 수동으로 연결할 수 있습니다. [명령줄을 통해 서브스크립션 연결 및 제거](#)를 참조하십시오.

추가 리소스

- [Red Hat 고객 포털 활성화 키 생성](#)
- [명령줄을 통해 서브스크립션 연결 및 제거](#)
- [Red Hat 서브스크립션 관리자 사용 및 구성](#)

3.5. 추가 리소스

- [퍼블릭 클라우드의 Red Hat](#)
- [Google Cloud](#)

4장. GOOGLE CLOUD PLATFORM에서 RED HAT HIGH AVAILABILITY CLUSTER 구성

GCE(Google Compute Engine) 가상 머신(VM) 인스턴스를 클러스터 노드로 사용하여 GCP(Google Cloud Platform)에서 HA(Red Hat High Availability) 클러스터를 구성하려면 다음 섹션을 참조하십시오.

이는 다음에 대한 정보를 제공합니다.

- GCP용 환경을 설정하기 위한 사전 요구 사항 절차 환경을 설정한 후에는 VM 인스턴스를 생성하고 구성할 수 있습니다.
- 개별 노드를 GCP의 HA 노드 클러스터로 변환하는 HA 클러스터 생성과 관련된 절차입니다. 여기에는 각 클러스터 노드에 High Availability 패키지 및 에이전트를 설치하고, 펜싱을 구성하고, 네트워크 리소스 에이전트를 설치하는 절차가 포함됩니다.

사전 요구 사항

- [Red Hat Cloud Access 프로그램](#)에 등록하고 사용되지 않은 RHEL 서브스크립션이 있어야 합니다. 연결된 서브스크립션에는 각 GCP 인스턴스에 대해 다음 리포지토리에 대한 액세스 권한이 포함되어야 합니다.
 - Red Hat Enterprise Linux 9 Server: rhel-9-server-rpms/8Server/x86_64
 - Red Hat Enterprise Linux 9 Server(High Availability): rhel-9-server-ha-rpms/8Server/x86_64
- 활성 GCP 프로젝트에 속하고 프로젝트에서 리소스를 생성할 수 있는 충분한 권한이 있어야 합니다.
- 프로젝트에는 개별 사용자가 아닌 VM 인스턴스에 속하는 **서비스 계정**이 있어야 합니다. 별도의 서비스 계정을 생성하는 대신 [기본 서비스 계정 사용에 대한 정보는 Compute Engine 기본 서비스 계정 사용을 참조하십시오.](#)

프로젝트 관리자가 사용자 지정 서비스 계정을 생성하는 경우 다음 역할에 맞게 서비스 계정을 구성해야 합니다.

- Cloud Trace 에이전트
- 컴퓨팅 관리자
- 컴퓨팅 네트워크 관리자
- Cloud Datastore 사용자
- 로깅 관리자
- 모니터링 편집기
- Monitoring Metric Writer
- 서비스 계정 관리자
- 스토리지 관리자

4.1. 필수 시스템 패키지

RHEL의 기본 이미지를 생성하고 구성하려면 호스트 시스템에 다음 패키지가 설치되어 있어야 합니다.

표 4.1. 시스템 패키지

패키지	리포지토리	설명
libvirt	rhel-9-for-x86_64-appstream-rpms	플랫폼 가상화 관리를 위한 오픈 소스 API, 데몬 및 관리 도구
virt-install	rhel-9-for-x86_64-appstream-rpms	가상 머신 구축을 위한 명령줄 유틸리티
libguestfs	rhel-9-for-x86_64-appstream-rpms	가상 머신 파일 시스템 액세스 및 수정을 위한 라이브러리
guestfs-tools	rhel-9-for-x86_64-appstream-rpms	VM용 시스템 관리 도구; virt-customize 유틸리티 포함

4.2. GCP의 RED HAT ENTERPRISE LINUX 이미지 옵션

다음 표에는 Google Cloud Platform의 RHEL 9에 대한 이미지 선택 사항 및 이미지 옵션의 차이점이 나열되어 있습니다.

표 4.2. 이미지 옵션

이미지 옵션	서브스크립션	샘플 시나리오	고려 사항
Red Hat Gold Image를 배포하도록 선택합니다.	기존 Red Hat 서브스크립션을 사용합니다.	Red Hat Cloud Access 프로그램 을 통해 서브스크립션을 활성화한 다음 Google Cloud Platform에서 Red Hat Gold Image를 선택합니다. 골드 이미지 및 Google Cloud Platform에서 액세스하는 방법에 대한 자세한 내용은 Red Hat Cloud Access 참조 가이드 를 참조하십시오.	서브스크립션에는 Red Hat 제품 비용이 포함되며 다른 모든 인스턴스 비용으로 Google에 비용을 지불하는 것이 좋습니다. Red Hat Gold Image는 기존 Red Hat 서브스크립션을 사용하므로 "Cloud Access" 이미지라고 합니다. Red Hat은 Cloud Access 이미지에 대한 직접 지원을 제공합니다.

이미지 옵션	서브스크립션	샘플 시나리오	고려 사항
GCP로 이동하는 사용자 정의 이미지를 배포하도록 선택합니다.	기존 Red Hat 서브스크립션을 사용합니다.	Red Hat Cloud Access 프로그램 을 통해 서브스크립션을 활성화하고 사용자 지정 이미지를 업로드하고 서브스크립션을 첨부하십시오.	서브스크립션에는 Red Hat 제품 비용이 포함되어 다른 모든 인스턴스 비용을 절감할 수 있습니다. 기존 Red Hat 서브스크립션을 사용하므로 GCP로 이동하는 사용자 지정 이미지를 "Cloud Access" 이미지라고 합니다. Red Hat은 Cloud Access 이미지에 대한 직접 지원을 제공합니다.
RHEL이 포함된 기존 GCP 이미지를 배포하도록 선택합니다.	GCP 이미지에는 Red Hat 제품이 포함되어 있습니다.	GCP Compute Engine 에서 인스턴스를 시작할 때 RHEL 이미지를 선택하거나 Google Cloud Platform Marketplace 에서 이미지를 선택합니다.	사용량에 따라 사용량 기준(Pay-as-you-go) 모델로 GCP를 매시간 결제합니다. 이러한 이미지를 "온 디맨드" 이미지라고 합니다. GCP는 지원 계약을 통해 주문형 이미지를 지원합니다.



중요

온디맨드 인스턴스를 Red Hat Cloud Access 인스턴스로 변환할 수 없습니다. 온 디맨드 이미지에서 BYOS(your-own-subscription) 이미지로 변경하려면 새 Red Hat Cloud Access 인스턴스를 생성하고 온 디맨드 인스턴스에서 데이터를 마이그레이션합니다. 데이터를 마이그레이션한 후 이중 청구를 방지하기 위해 온 디맨드 인스턴스를 취소합니다.

다음 섹션에서는 사용자 지정 이미지와 관련된 정보 및 절차를 제공합니다.

추가 리소스

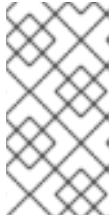
- [퍼블릭 클라우드의 Red Hat](#)
- [컴퓨팅 엔진 이미지](#)
- [Red Hat Cloud Access 참조 가이드](#)
- [사용자 정의 이미지에서 인스턴스 생성](#)

4.3. GOOGLE CLOUD SDK 설치

Google Cloud SDK를 설치하려면 다음 단계를 완료하십시오.

절차

1. Google Cloud SDK 아카이브를 다운로드하고 추출하기 위한 GCP 지침을 따르십시오. 자세한 내용은 GCP 문서 [빠른 시작 Linux](#)를 참조하십시오.
2. Google Cloud SDK를 초기화하는 것과 동일한 지침을 따르십시오.



참고

Google Cloud SDK를 초기화한 후에는 **gcloud** CLI 명령을 사용하여 작업을 수행하고 프로젝트 및 인스턴스에 대한 정보를 얻을 수 있습니다. 예를 들어 **gcloud compute project-info describe --project <project-name>** 명령을 사용하여 프로젝트 정보를 표시할 수 있습니다.

추가 리소스

- [Linux 빠른 시작](#)
- [gcloud 명령 참조](#)
- [gcloud 명령줄 툴 개요](#)

4.4. GCP 이미지 버킷 생성

다음 문서에는 기본 위치에 [다중 지역](#) 버킷을 생성하기 위한 최소 요구사항이 포함되어 있습니다.

사전 요구 사항

- GCP 스토리지 유틸리티(**gsutil**)

절차

1. Google Cloud Platform에 아직 로그인하지 않은 경우 다음 명령을 사용하여 로그인합니다.

```
# gcloud auth login
```

2. 스토리지 버킷을 생성합니다.

```
$ gsutil mb gs://BucketName
```

예제:

```
$ gsutil mb gs://rhel-ha-bucket
```

추가 리소스

- [버킷 만들기](#)

4.5. 사용자 지정 가상 프라이빗 클라우드 네트워크 및 서브넷 생성

사용자 지정 VPC(가상 프라이빗 클라우드) 네트워크 및 서브넷을 생성하려면 다음 단계를 완료합니다.

절차

1. GCP 콘솔을 시작합니다.

2. 왼쪽 탐색 창의 **Networking** 에서 VPC 네트워크를 선택합니다.
3. **Create VPC Network**를 클릭합니다.
4. VPC 네트워크의 이름을 입력합니다.
5. 새 서브넷 아래의 클러스터를 생성할 리전에서 사용자 지정 서브넷을 만듭니다.
6. 생성을 클릭합니다.

4.6. 기본 GCP 이미지 준비 및 가져오기

GCP용 Red Hat Enterprise Linux 9 이미지를 준비하려면 다음 단계를 완료합니다.

절차

1. 파일을 변환합니다. GCP에 업로드된 이미지는 원시 형식이어야 하며 **disk.raw** 라는 이름이 지정되어야 합니다.

```
$ qemu-img convert -f qcow2 ImageName.qcow2 -O raw disk.raw
```

2. 원시 파일을 압축합니다. GCP로 업로드된 이미지는 압축해야 합니다.

```
$ tar -Sczf ImageName.tar.gz disk.raw
```

3. 압축된 이미지를 이전에 생성한 버킷으로 가져옵니다.

```
$ gsutil cp ImageName.tar.gz gs://BucketName
```

4.7. 기본 GCP 인스턴스 생성 및 구성

GCP 운영 및 보안 요구 사항을 준수하는 GCP 인스턴스를 생성하고 구성하려면 다음 단계를 완료합니다.

절차

1. 버킷의 압축된 파일에서 이미지를 만듭니다.

```
$ gcloud compute images create BaseImageName --source-uri gs://BucketName/BaseImageName.tar.gz
```

예제:

```
[admin@localhost ~] $ gcloud compute images create rhel-76-server --source-uri gs://user-rhelha/rhel-server-76.tar.gz
Created
[https://www.googleapis.com/compute/v1/projects/MyProject/global/images/rhel-server-76].
NAME          PROJECT          FAMILY DEPRECATED STATUS
rhel-76-server rhel-ha-testing-on-gcp          READY
```

2. 이미지에서 템플릿 인스턴스를 생성합니다. 기본 RHEL 인스턴스에 필요한 최소 크기는 n1-standard-2입니다. 추가 구성 옵션은 [gcloud compute 인스턴스 생성](#)을 참조하십시오.

```
$ gcloud compute instances create BaseInstanceName --can-ip-forward --machine-type n1-standard-2 --image BaseImageName --service-account ServiceAccountEmail
```

예제:

```
[admin@localhost ~] $ gcloud compute instances create rhel-76-server-base-instance --can-ip-forward --machine-type n1-standard-2 --image rhel-76-server --service-account account@project-name-on-gcp.iam.gserviceaccount.com
Created [https://www.googleapis.com/compute/v1/projects/rhel-ha-testing-on-gcp/zones/us-east1-b/instances/rhel-76-server-base-instance].
NAME      ZONE      MACHINE_TYPE  PREEMPTIBLE  INTERNAL_IP  EXTERNAL_IP  STATUS
rhel-76-server-base-instance  us-east1-bn1-standard-2      10.10.10.3  192.227.54.211  RUNNING
```

3. SSH 터미널 세션을 사용하여 인스턴스에 연결합니다.

```
$ ssh root@PublicIPAddress
```

4. RHEL 소프트웨어를 업데이트합니다.

- a. Red Hat Subscription Manager (RHSM)에 등록합니다.
- b. 서브스크립션 풀 ID를 활성화하거나 `--auto-attach` 명령을 사용합니다.
- c. 모든 리포지토리를 비활성화합니다.

```
# subscription-manager repos --disable=*
```

- d. 다음 리포지토리를 활성화합니다.

```
# subscription-manager repos --enable=rhel-9-server-rpms
```

- e. `dnf update` 명령을 실행합니다.

```
# dnf update -y
```

5. 실행 중인 인스턴스에 GCP Linux 게스트 환경을 설치합니다(장기 설치). 자세한 내용은 [guest environment in-place](#) 를 참조하십시오.
6. CentOS/RHEL 옵션을 선택합니다.
7. 스크립트를 복사하여 명령 프롬프트에 붙여넣어 스크립트를 즉시 실행합니다.
8. 인스턴스를 다음과 같이 변경합니다. 이러한 변경 사항은 사용자 정의 이미지에 대한 GCP 권장 사항을 기반으로 합니다. 자세한 내용은 [gcloudcompute 이미지 목록](#) 을 참조하십시오.
 - a. `/etc/chrony.conf` 파일을 편집하고 모든 NTP 서버를 제거합니다.
 - b. 다음 NTP 서버를 추가합니다.

```
metadata.google.internal iburst Google NTP server
```
 - c. 영구 네트워크 장치 규칙을 제거합니다.

```
# rm -f /etc/udev/rules.d/70-persistent-net.rules
# rm -f /etc/udev/rules.d/75-persistent-net-generator.rules
```

- d. 자동으로 시작되도록 network 서비스를 설정합니다.

```
# chkconfig network on
```

- e. 자동으로 시작되도록 sshd 서비스를 설정합니다.

```
# systemctl enable sshd
# systemctl is-enabled sshd
```

- f. 시간대를 UTC로 설정합니다.

```
# ln -sf /usr/share/zoneinfo/UTC /etc/localtime
```

- g. (선택 사항) /etc/ssh/ssh_config 파일을 편집하고 다음 행을 파일 끝에 추가합니다. 이렇게 하면 비활성 기간 동안 SSH 세션이 활성 상태로 유지됩니다.

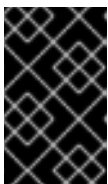
```
# Server times out connections after several minutes of inactivity.
# Keep alive ssh connections by sending a packet every 7 minutes.
ServerAliveInterval 420
```

- h. /etc/ssh/sshd_config 파일을 편집하고 필요한 경우 다음과 같이 변경합니다. ClientAliveInterval 420 설정은 선택 사항이며, 이렇게 하면 비활성 기간 동안 SSH 세션을 활성 상태로 유지합니다.

```
PermitRootLogin no
PasswordAuthentication no
AllowTcpForwarding yes
X11Forwarding no
PermitTunnel no
# Compute times out connections after 10 minutes of inactivity.
# Keep ssh connections alive by sending a packet every 7 minutes.
ClientAliveInterval 420
```

9. 암호 액세스를 비활성화합니다.

```
ssh_pwauth from 1 to 0.
ssh_pwauth: 0
```



중요

이전에는 SSH 세션 액세스를 허용하여 인스턴스를 구성하기 위해 암호 액세스를 활성화했습니다. 암호 액세스를 비활성화해야 합니다. 모든 SSH 세션 액세스는 암호가 없어야 합니다.

10. 서브스크립션 관리자에서 인스턴스를 등록 취소합니다.

```
# subscription-manager unregister
```

11. 셸 기록을 정리합니다. 다음 절차를 위해 인스턴스를 계속 실행하십시오.

```
# export HISTSIZE=0
```

4.8. 스냅샷 이미지 생성

인스턴스 구성 설정을 유지하고 스냅샷을 생성하려면 다음 단계를 완료합니다.

절차

1. 실행 중인 인스턴스에서 데이터를 디스크에 동기화합니다.

```
# sync
```

2. 호스트 시스템에서 스냅샷을 생성합니다.

```
$ gcloud compute disks snapshot InstanceName --snapshot-names SnapshotName
```

3. 호스트 시스템의 스냅샷에서 구성된 이미지를 생성합니다.

```
$ gcloud compute images create ConfiguredImageFromSnapshot --source-snapshot SnapshotName
```

추가 리소스

- [영구 디스크 스냅샷 생성](#)

4.9. HA 노드 템플릿 인스턴스 및 HA 노드 생성

스냅샷에서 이미지를 구성한 후 노드 템플릿을 생성할 수 있습니다. 이 템플릿을 사용하여 모든 HA 노드를 생성합니다. 템플릿과 HA 노드를 생성하려면 다음 단계를 완료합니다.

절차

1. 인스턴스 템플릿을 생성합니다.

```
$ gcloud compute instance-templates create InstanceTemplateName --can-ip-forward -  
-machine-type n1-standard-2 --image ConfiguredImageFromSnapshot --service-  
account ServiceAccountEmailAddress
```

예제:

```
[admin@localhost ~] $ gcloud compute instance-templates create rhel-91-instance-  
template --can-ip-forward --machine-type n1-standard-2 --image rhel-91-gcp-image --  
service-account account@project-name-on-gcp.iam.gserviceaccount.com  
Created [https://www.googleapis.com/compute/v1/projects/project-name-on-  
gcp/global/instanceTemplates/rhel-91-instance-template].  
NAME MACHINE_TYPE PREEMPTIBLE CREATION_TIMESTAMP  
rhel-91-instance-template n1-standard-2 2018-07-25T11:09:30.506-07:00
```

2. 하나의 영역에 여러 노드를 생성합니다.


```
# gcloud compute instances create NodeName01 NodeName02 --source-instance-
template InstanceTemplateName --zone RegionZone --network=NetworkName --
subnet=SubnetName
```

예제:

```
[admin@localhost ~] $ gcloud compute instances create rhel81-node-01 rhel81-node-
02 rhel81-node-03 --source-instance-template rhel-91-instance-template --zone us-
west1-b --network=projectVPC --subnet=range0
Created [https://www.googleapis.com/compute/v1/projects/project-name-on-
gcp/zones/us-west1-b/instances/rhel81-node-01].
Created [https://www.googleapis.com/compute/v1/projects/project-name-on-
gcp/zones/us-west1-b/instances/rhel81-node-02].
Created [https://www.googleapis.com/compute/v1/projects/project-name-on-
gcp/zones/us-west1-b/instances/rhel81-node-03].
NAME          ZONE          MACHINE_TYPE  PREEMPTIBLE  INTERNAL_IP
EXTERNAL_IP  STATUS
rhel81-node-01 us-west1-b    n1-standard-2      10.10.10.4   192.230.25.81
RUNNING
rhel81-node-02 us-west1-b    n1-standard-2      10.10.10.5   192.230.81.253
RUNNING
rhel81-node-03 us-east1-b    n1-standard-2      10.10.10.6   192.230.102.15
RUNNING
```

4.10. HA 패키지 및 에이전트 설치

모든 노드에서 다음 단계를 완료합니다.

절차

1. Google Cloud Console에서 Compute Engine 을 선택한 다음 VM 인스턴스를 선택합니다.
2. 인스턴스를 선택하고 SSH 옆에 있는 화살표를 클릭하고 gcloud보기 명령 옵션을 선택합니다.
3. 인스턴스에 암호 없이 액세스할 수 있도록 명령 프롬프트에 이 명령을 붙여 넣습니다.
4. sudo 계정 액세스를 활성화하고 Red Hat Subscription Manager에 등록합니다.
5. 서브스크립션 풀 ID를 활성화하거나 --auto-attach 명령을 사용합니다.
6. 모든 리포지토리를 비활성화합니다.

```
# subscription-manager repos --disable=*
```

7. 다음 리포지토리를 활성화합니다.

```
# subscription-manager repos --enable=rhel-9-server-rpms
# subscription-manager repos --enable=rhel-9-for-x86_64-highavailability-rpms
```

8. pcs pacemaker, fence 에이전트, 리소스 에이전트를 설치합니다.

```
# dnf install -y pcs pacemaker fence-agents-gce resource-agents-gcp
```

9. 모든 패키지를 업데이트합니다.

```
# dnf update -y
```

4.11. HA 서비스 구성

HA 서비스를 구성하려면 모든 노드에서 다음 단계를 완료합니다.

절차

1. 사용자 **hacluster** 는 이전 단계에서 **pcs** 및 **pacemaker** 설치 중에 생성되었습니다. 모든 클러스터 노드에서 **hacluster** 사용자의 암호를 생성합니다. 모든 노드에 동일한 암호를 사용합니다.

```
# passwd hacluster
```

2. **firewalld** 서비스가 설치된 경우 HA 서비스를 추가합니다.

```
# firewall-cmd --permanent --add-service=high-availability
```

```
# firewall-cmd --reload
```

3. **pcs** 서비스를 시작하고 부팅 시 시작되도록 활성화합니다.

```
# systemctl start pcsd.service
```

```
# systemctl enable pcsd.service
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/pcsd.service to /usr/lib/systemd/system/pcsd.service.
```

검증

1. **pcsd** 서비스가 실행 중인지 확인합니다.

```
# systemctl status pcsd.service
```

```
pcsd.service - PCS GUI and remote configuration interface
Loaded: loaded (/usr/lib/systemd/system/pcsd.service; enabled; vendor preset: disabled)
Active: active (running) since Mon 2018-06-25 19:21:42 UTC; 15s ago
Docs: man:pcsd(8)
man:pcs(8)
Main PID: 5901 (pcsd)
CGroup: /system.slice/pcsd.service
└─5901 /usr/bin/ruby /usr/lib/pcsd/pcsd > /dev/null &
```

2. **/etc/hosts** 파일을 편집합니다. 모든 노드의 RHEL 호스트 이름 및 내부 IP 주소를 추가합니다.

추가 리소스

- [RHEL 클러스터 노드에 /etc/hosts 파일을 설정하려면 어떻게 해야 하나요?](#)

4.12. 클러스터 생성

노드 클러스터를 생성하려면 다음 단계를 완료합니다.

절차

1. 노드 중 하나에서 **pcs** 사용자를 인증합니다. 명령에서 클러스터에 있는 각 노드의 이름을 지정합니다.

```
# pcs host auth hostname1 hostname2 hostname3
Username: hacluster
Password:
hostname1: Authorized
hostname2: Authorized
hostname3: Authorized
```

2. 클러스터를 생성합니다.

```
# pcs cluster setup cluster-name hostname1 hostname2 hostname3
```

검증

1. 다음 명령을 실행하여 노드가 시작될 때 자동으로 클러스터에 참여할 수 있도록 합니다.

```
# pcs cluster enable --all
```

2. 클러스터를 시작합니다.

```
# pcs cluster start --all
```

4.13. 펜싱 장치 생성

펜싱 장치를 생성하려면 다음 단계를 완료합니다.

대부분의 기본 구성의 경우 GCP 인스턴스 이름과 RHEL 호스트 이름이 동일합니다.

절차

1. GCP 인스턴스 이름을 가져옵니다. 다음 명령의 출력에는 인스턴스의 내부 ID도 표시되어 있습니다.

```
# fence_gce --zone us-west1-b --project=rhel-ha-on-gcp -o list
```

예제:

```
[root@rhel81-node-01 ~]# fence_gce --zone us-west1-b --project=rhel-ha-testing-on-gcp -o list
4435801234567893181,InstanceName-3
4081901234567896811,InstanceName-1
7173601234567893341,InstanceName-2
```

2. 차단 장치를 생성합니다.

```
# pcs stonith create FenceDeviceName fence_gce zone=Region-Zone
project=MyProject
```

검증

- 펜스 장치가 시작되었는지 확인합니다.

```
# pcs status
```

예제:

```
[root@rhel81-node-01 ~]# pcs status

Cluster name: gcp-cluster
Stack: corosync
Current DC: rhel81-node-02 (version 1.1.18-11.el7_5.3-2b07d5c5a9) - partition with quorum
Last updated: Fri Jul 27 12:53:25 2018
Last change: Fri Jul 27 12:51:43 2018 by root via cibadmin on rhel81-node-01

3 nodes configured
3 resources configured

Online: [ rhel81-node-01 rhel81-node-02 rhel81-node-03 ]

Full list of resources:

us-west1-b-fence (stonith:fence_gce): Started rhel81-node-01

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

4.14. GCP 노드 권한 부여 구성

계정 자격 증명을 사용하여 GCP에 액세스하도록 클라우드 SDK 툴을 구성합니다.

절차

각 노드에 다음 명령을 입력하여 각 노드를 프로젝트 ID 및 계정 자격 증명으로 초기화합니다.

```
# gcloud-ra init
```

4.15. GCP-VCP-MOVE-VIP 리소스 에이전트 구성

gcp-vpc-move-vip 리소스 에이전트는 실행 중인 인스턴스에 보조 IP 주소(alias IP)를 연결합니다. 클러스터의 다른 노드 간에 전달할 수 있는 유동 IP 주소입니다.

이 리소스에 대한 자세한 정보를 보려면 다음을 수행합니다.

```
# pcs resource describe gcp-vpc-move-vip
```

기본 서브넷 주소 범위 또는 보조 서브넷 주소 범위를 사용하도록 리소스 에이전트를 구성할 수 있습니다.

기본 서브넷 주소 범위

기본 VPC 서브넷에 대한 리소스를 구성하려면 다음 단계를 완료합니다.

절차

1. **aliasip** 리소스를 만듭니다. 사용되지 않는 내부 IP 주소를 포함합니다. 명령에 CIDR 블록을 포함합니다.

```
# pcs resource create aliasip gcp-vpc-move-vip alias_ip=UnusedIPAddress/CIDRblock
```

예제:

```
[root@rhel81-node-01 ~]# pcs resource create aliasip gcp-vpc-move-vip
alias_ip=10.10.10.200/32
```

2. 노드에서 IP를 관리할 **IPAddr2** 리소스를 생성합니다.

```
# pcs resource create vip IPAddr2 nic=interface ip=AliasIPAddress cidr_netmask=32
```

예제:

```
[root@rhel81-node-01 ~]# pcs resource create vip IPAddr2 nic=eth0 ip=10.10.10.200
cidr_netmask=32
```

3. **vipgrp** 에서 네트워크 리소스를 그룹화합니다.

```
# pcs resource group add vipgrp aliasip vip
```

검증

1. 리소스가 시작되었고 **vipgrp** 로 그룹화되었는지 확인합니다.

```
# pcs status
```

2. 리소스가 다른 노드로 이동할 수 있는지 확인합니다.

```
# pcs resource move vip Node
```

예제:

```
[root@rhel81-node-01 ~]# pcs resource move vip rhel81-node-03
```

3. 다른 노드에서 **vip** 가 성공적으로 시작되었는지 확인합니다.

```
# pcs status
```

보조 서브넷 주소 범위

보조 서브넷 주소 범위에 대한 리소스를 구성하려면 다음 단계를 완료합니다.

사전 요구 사항

- 사용자 지정 네트워크 및 서브넷 생성

절차

1. 보조 서브넷 주소 범위를 생성합니다.

```
# gcloud-ra compute networks subnets update SubnetName --region RegionName --add-secondary-ranges SecondarySubnetName=SecondarySubnetRange
```

예제:

```
# gcloud-ra compute networks subnets update range0 --region us-west1 --add-secondary-ranges range1=10.10.20.0/24
```

2. **aliasip** 리소스를 만듭니다. 보조 서브넷 주소 범위에 사용되지 않은 내부 IP 주소를 생성합니다. 명령에 CIDR 블록을 포함합니다.

```
# pcs resource create aliasip gcp-vpc-move-vip alias_ip=UnusedIPAddress/CIDRblock
```

예제:

```
[root@rhel81-node-01 ~]# pcs resource create aliasip gcp-vpc-move-vip alias_ip=10.10.20.200/32
```

3. 노드에서 IP를 관리할 **IPAddr2** 리소스를 생성합니다.

```
# pcs resource create vip IPAddr2 nic=interface ip=AliasIPAddress cidr_netmask=32
```

예제:

```
[root@rhel81-node-01 ~]# pcs resource create vip IPAddr2 nic=eth0 ip=10.10.20.200 cidr_netmask=32
```

4. **vipgrp** 에서 네트워크 리소스를 그룹화합니다.

```
# pcs resource group add vipgrp aliasip vip
```

검증

1. 리소스가 시작되었고 **vipgrp** 로 그룹화되었는지 확인합니다.

```
# pcs status
```

2. 리소스가 다른 노드로 이동할 수 있는지 확인합니다.

```
# pcs resource move vip Node
```

예제:

```
[root@rhel81-node-01 ~]# pcs resource move vip rhel81-node-03
```

3. 다른 노드에서 **vip** 가 성공적으로 시작되었는지 확인합니다.

```
# pcs status
```

■

4.16. 추가 리소스

- [RHEL 고가용성 클러스터에 대한 지원 정책 - 전송 프로토콜](#)
- [VPC 네트워크 개요](#)
- [RHEL 고가용성의 구성 요소, 개념 및 기능 탐색 - 전송 프로토콜 개요](#)
- [RHEL 고가용성 클러스터에 대한 설계 지침 - 전송 프로토콜 선택](#)