



Red Hat Enterprise Linux 9

RHEL for Edge 이미지 설치, 구성, 관리

Red Hat Enterprise Linux 9의 RHEL for Edge 이미지 설치, 구성 및 관리

Red Hat Enterprise Linux 9 RHEL for Edge 이미지 설치, 구성, 관리

Red Hat Enterprise Linux 9의 RHEL for Edge 이미지 설치, 구성 및 관리

Enter your first name here. Enter your surname here.

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

Enter your email address here.

법적 공지

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

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

초록

이 문서는 이미지 빌더를 사용하여 사용자 지정 RHEL(rpm-ostree) 이미지를 작성한 다음 예지 서버에서 이미지를 원격으로 설치 및 관리하려는 사용자를 위한 것입니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체	4
RED HAT 문서에 관한 피드백 제공	5
1장. RHEL FOR EDGE 이미지 소개	6
1.1. RHEL FOR EDGE - 지원 아키텍처	6
1.2. RHEL FOR EDGE 이미지를 작성하고 배포하는 방법	6
1.3. 비 네트워크 기반 배포	8
1.4. 네트워크 기반 배포	9
1.5. RHEL RPM 이미지와 RHEL FOR EDGE 이미지 간의 차이점	10
2장. 이미지 빌더 설정	12
2.1. 이미지 빌더 시스템 요구 사항	12
2.2. 이미지 빌더 설치	12
3장. 리포지토리 관리	14
3.1. 이미지 빌더 기본 시스템 리포지토리	14
3.2. 시스템 리포지토리 덮어쓰기	14
3.3. 서브스크립션에 대한 지원을 사용하여 시스템 리포지토리 덮어쓰기	15
4장. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 이미지 구성	17
4.1. RHEL 웹 콘솔에서 이미지 빌더에 액세스	17
4.2. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 커밋 이미지의 블루프린트 생성	18
4.3. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 컨테이너 이미지용 블루프린트 생성	19
4.4. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 설치 프로그램 이미지용 블루프린트 생성	20
4.5. RHEL FOR EDGE 이미지 블루프린트에 소스 추가	20
4.6. RHEL FOR EDGE 이미지 블루프린트의 관리자 계정 생성	21
4.7. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 커밋 이미지 생성	22
4.8. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 컨테이너 이미지 생성	23
4.9. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 설치 프로그램 이미지 생성	24
4.10. 에지 이미지 RHEL 다운로드	25
4.11. 추가 리소스	25
5장. IMAGE BUILDER 명령줄을 사용하여 에지 이미지용 RHEL 구성	26
5.1. 네트워크 기반 배포 워크플로	26
5.1.1. 이미지 빌더 명령줄 인터페이스를 사용하여 Edge Commit 이미지 청사진용 RHEL 생성	26
5.1.2. 이미지 빌더 명령줄 인터페이스를 사용하여 엣지 커밋 이미지용 RHEL 생성	28
5.1.3. 이미지 빌더 명령줄 인터페이스를 사용하여 상위 커밋으로 에지 이미지 업데이트 생성	29
5.1.4. Image Builder 명령줄 인터페이스를 사용하여 에지 이미지의 RHEL 다운로드	30
5.2. 비 네트워크 기반 배포 워크플로	31
5.2.1. 이미지 빌더 CLI를 사용하여 에지 컨테이너 이미지 청사진용 RHEL 생성	31
5.2.2. 이미지 빌더 CLI를 사용하여 엣지 컨테이너 이미지용 RHEL 생성	32
5.2.3. 이미지 빌더 CLI를 사용하여 에지 설치 프로그램 이미지 청사진용 RHEL 생성	33
5.2.4. 네트워크 기반 배포에 명령줄 인터페이스를 사용하여 에지 설치 프로그램 이미지 생성	34
5.2.5. 이미지 빌더 CLI를 사용하여 에지 설치 프로그램 이미지용 RHEL 다운로드	35
5.3. 지원되는 이미지 사용자 지정	35
5.4. 추가 리소스	40
6장. 에지 이미지용 RHEL을 프로비저닝하기 위해 간소화된 설치 프로그램 이미지 빌드	41
6.1. 설치 프로그램 이미지 빌드 및 배포 단순화	41
6.2. 이미지 빌더 CLI를 사용하여 단순화된 이미지에 대한 청사진 생성	42
6.3. 이미지 빌더 CLI를 사용하여 에지 SIMPLIFIED 설치 프로그램 이미지용 RHEL 생성	43
6.4. 이미지 빌더 명령줄 인터페이스를 사용하여 에지 이미지용 간소화된 RHEL 다운로드	44

6.5. UEFI HTTP BOOT 서버 설정	45
6.6. 가상 머신에 단순화된 ISO 이미지 배포	46
6.7. USB 플래쉬 드라이브에서 SIMPLIFIED ISO 이미지 배포	47
7장. FDO를 사용하여 에지 장치용 RHEL 자동 프로비저닝 및 온보딩	48
7.1. FDO 장치 온보딩 프로세스	48
7.2. 에지 장치용 RHEL 자동 프로비저닝 및 온보딩	50
7.3. 키 및 인증서 생성	50
7.4. 제조 서버 패키지 설치	51
7.5. FDO 인증을 사용하여 에지 장치용 RHEL 자동 온보딩	52
8장. 네트워크 기반 환경에서 에지 이미지에 RHEL 배포	54
8.1. 에지 이미지 커밋을 위해 RHEL 추출	54
8.2. 에지 이미지용 RHEL을 설치하기 위해 웹 서버 설정	56
8.3. RHEL BOOT.ISO 이미지 다운로드	57
8.4. KICKSTART 파일을 사용하여 에지 이미지용 RHEL 설치	58
9장. 네트워크 기반이 아닌 환경에서 EDGE 이미지용 RHEL 배포	61
9.1. 네트워크 기반이 아닌 배포를 위한 EDGE 컨테이너 이미지용 RHEL 생성	61
9.2. 네트워크 기반이 아닌 배포에 대한 엣지 설치 프로그램 이미지용 RHEL 생성	62
9.3. 네트워크 기반이 아닌 배포를 위한 EDGE 이미지용 RHEL 설치	63
10장. 에지 이미지 RHEL 관리	65
10.1. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 에지 이미지 승격을 위한 RHEL 편집	65
10.1.1. RHEL 웹 콘솔에서 Image Builder를 사용하여 에지 이미지 options를 위한 RHEL에 구성 요소 추가	65
10.1.2. RHEL 웹 콘솔을 사용하여 에지 이미지 승격을 위한 RHEL의 기존 구성 요소 버전 변경	66
10.1.3. RHEL 웹 콘솔에서 Image Builder를 사용하여 에지 이미지 options용 RHEL에서 구성 요소 제거	66
10.1.4. 명령줄 인터페이스를 사용하여 에지 이미지 options를 위한 RHEL 편집	67
10.2. 에지 이미지 RHEL 업데이트	68
10.2.1. RHEL for Edge 이미지 업데이트 배포 방법	68
10.2.2. 에지 이미지 업데이트를 위해 수동으로 RHEL 배포	69
10.2.3. 명령줄을 사용하여 에지 이미지 업데이트를 수동으로 위한 RHEL 배포	71
10.2.4. 네트워크 기반이 아닌 배포를 위해 RHEL for Edge 이미지 업데이트를 수동으로 배포	73
10.3. 에지 시스템을 위한 RHEL 업그레이드	75
10.3.1. RHEL 8 시스템을 RHEL 9로 업그레이드	75
10.4. 에지 자동 이미지 업데이트를 위한 RHEL 배포	77
10.4.1. 에지 이미지 업데이트 정책의 RHEL 업데이트 정책 업데이트	78
10.4.2. RHEL for Edge 자동 다운로드 및 업데이트 스테이징 활성화	79
10.5. 에지 이미지용 RHEL 롤백	80
10.5.1. RHEL에서 에지 이미지 롤백 방법	80
10.5.2. 에지 이미지에 대해 RHEL을 수동으로 롤백	80
10.5.3. 자동화된 프로세스를 사용하여 에지 이미지에 대해 RHEL 롤백	81
부록 A. 용어 및 명령	83
A.1. OSTREE 및 RPM-OSTREE 용어	83
A.2. OSTREE 명령	83
A.3. RPM-OSTREE 명령	84
A.4. FDO 자동 온보딩 용어	85
A.5. FDO 자동 온보딩 기술	86

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

문서에 대한 피드백에 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. **Summary** (요약) 필드에 설명 제목을 입력합니다.
4. **Description** (설명) 필드에 개선을 위한 제안을 입력합니다. 문서의 관련 부분에 대한 링크를 포함합니다.
5. **버그 제출**을 클릭합니다.

1장. RHEL FOR EDGE 이미지 소개

RHEL for Edge 이미지는 RHEL for Edge 서버를 원격으로 설치하는 시스템 패키지가 포함된 **rpm-ostree** 이미지입니다.

시스템 패키지는 다음과 같습니다.

- 기본 OS 패키지
- Podman 컨테이너 엔진
- 추가 RPM 콘텐츠

이미지를 사용자 지정하여 요구 사항에 따라 OS 콘텐츠를 구성하고 온프레미스, 클라우드 또는 컨테이너 환경에서 물리적 및 가상 머신에 배포할 수 있습니다.

베어 메탈, 어플라이언스 및 에지 서버에 RHEL for Edge 이미지를 배포할 수 있습니다.

RHEL for Edge 이미지를 사용하면 다음을 수행할 수 있습니다.

Atomic upgrades State of each update is known / no changes are seen until reboot	Custom health checks and intelligent rollbacks Resiliency in case of failed upgrades	Container-focused workflow Separate core OS updates	Optimized OTA payloads Transfer only delta updates
--	--	---	--

125_RHEL_1020

1.1. RHEL FOR EDGE - 지원 아키텍처

현재 AMD 및 Intel 64비트 시스템에 RHEL for Edge 이미지를 배포할 수 있습니다.



참고

현재 RHEL for Edge는 ARM 시스템을 지원하지 않습니다.

1.2. RHEL FOR EDGE 이미지를 작성하고 배포하는 방법

RHEL for Edge 이미지를 작성하고 배포하는 전체 프로세스는 배포 환경이 네트워크 기반인지 **네트워크 기반** 또는 **비 네트워크 기반**인지에 따라 다릅니다.

RHEL for Edge 이미지에 대한 구성 및 배포에는 다음 두 단계가 포함됩니다.

1. 이미지 빌더를 사용하여 RHEL **rpm-ostree** 이미지 구성. **composer-cli** 툴의 명령줄 인터페이스를 통해 Image Builder에 액세스하거나 RHEL 웹 콘솔에서 그래픽 사용자 인터페이스를 사용할 수 있습니다.
2. RHEL 설치 프로그램을 사용하여 이미지를 배포합니다.

RHEL for Edge 이미지를 빌드하는 동안 다음 이미지 유형을 선택할 수 있습니다.

표 1.1. RHEL for Edge 이미지 유형

이미지 유형	설명	네트워크 기반 배포에 적합	비 네트워크 기반 배포에 적합
RHEL for Edge Commit (.tar)	커밋 이미지는 전체 운영 체제가 포함되어 있어도 직접 부팅할 수 없습니다. 커밋 이미지 유형을 부팅하려면 이 이미지 유형을 배포해야 합니다.	있음	없음
RHEL for Edge Container (.tar)	컨테이너는 OSTree 커밋을 생성하고 웹 서버를 사용하여 OCI 컨테이너에 포함합니다. 컨테이너가 시작되면 웹 서버에서 해당 커밋을 OSTree 리포지토리로 제공합니다.	없음	있음
RHEL for Edge Installer (.iso)	설치 프로그램 이미지 유형은 실행 중인 컨테이너에서 커밋을 가져오고 포함된 OSTree 커밋을 사용하도록 구성된 kickstart 파일로 설치 가능한 부팅 ISO를 생성합니다.	없음	있음
RHEL for Edge Raw image (.raw.xz)	압축된 원시 이미지는 기존의 배포된 OSTree 커밋이 있는 파티션 레이아웃을 포함하는 파일로 구성됩니다. 하드 드라이브에서 RHEL 원시 이미지를 플래시하거나 가상 머신에서 부팅할 수 있습니다.	있음	있음
RHEL for Edge Simplified Installer (.iso)	Simplified Installer 이미지 유형은 실행 중인 컨테이너에서 커밋을 가져오고 포함된 OSTree 커밋을 사용하도록 구성된 kickstart 파일로 설치 가능한 부팅 ISO를 생성합니다.	있음	있음

이미지 유형은 콘텐츠 측면에서 다르며, 따라서 다양한 유형의 배포 환경에 적합합니다.

추가 리소스

- 표준 RHEL 9 설치 수행

1.3. 비 네트워크 기반 배포

이미지 빌더를 사용하여 요구 사항에 맞게 유연한 RHEL **rpm-ostree** 이미지를 생성한 다음 Anaconda를 사용하여 환경에 배포합니다.

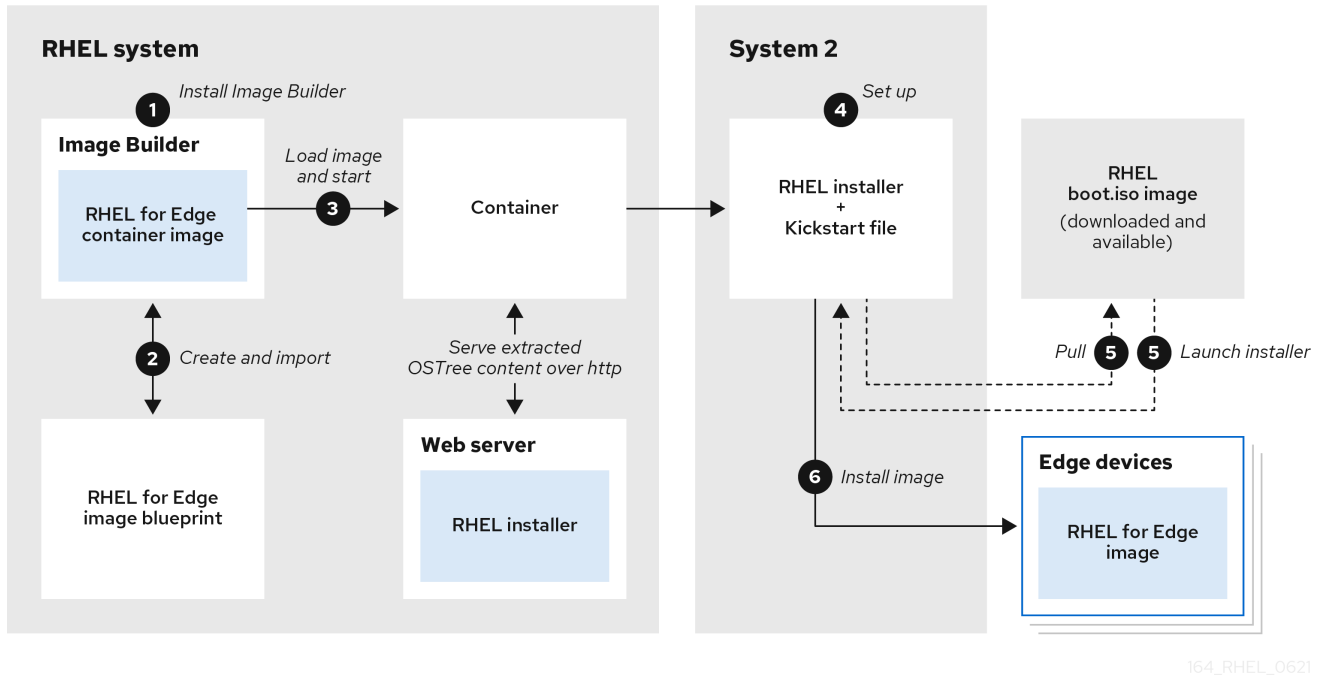
composer-cli 툴의 명령줄 인터페이스를 통해 Image Builder에 액세스하거나 RHEL 웹 콘솔에서 그래픽 사용자 인터페이스를 사용할 수 있습니다.

네트워크 기반이 아닌 배포에서 RHEL for Edge 이미지를 구성 및 배포하려면 다음과 같은 상위 수준 단계를 수행해야 합니다.

1. RHEL 시스템 설치 및 등록
2. 이미지 빌더 설치
3. 이미지 빌더를 사용하여 RHEL for Edge Container 이미지에 대한 사용자 지정으로 블루프린트 (Blueprint) 생성
4. 이미지 빌더에서 RHEL for Edge 블루프린트 가져오기
5. OSTree 리포지토리로 커밋을 배포할 준비가 된 웹 서버와 함께 OCI 컨테이너에 포함된 RHEL for Edge 이미지를 생성합니다.
6. RHEL for Edge Container 이미지 파일 다운로드
7. RHEL for Edge Container 커밋을 위해 리포지토리를 제공하는 컨테이너 배포
8. 이미지 빌더를 사용하여 RHEL for Edge 설치 프로그램 이미지의 블루프린트 생성
9. RHEL for Edge Container 이미지가 포함된 실행 중인 컨테이너에서 커밋을 가져오도록 구성된 RHEL for Edge 설치 관리자 이미지 생성
10. RHEL for Edge 설치 관리자 이미지 다운로드
11. 설치 실행
12. 이미지 생성

다음 다이어그램은 RHEL for Edge 이미지 비 네트워크 배포 워크플로를 나타냅니다.

그림 1.1. 네트워크 이외의 환경에서 RHEL for Edge 배포



1.4. 네트워크 기반 배포

이미지 빌더를 사용하여 요구 사항에 맞게 유연한 RHEL **rpm-ostree** 이미지를 생성한 다음 Anaconda를 사용하여 환경에 배포합니다. Image Builder는 배포 설정의 세부 정보를 자동으로 식별하고 이미지 출력을 **.tar** 파일로 **edge-commit**로 생성합니다.

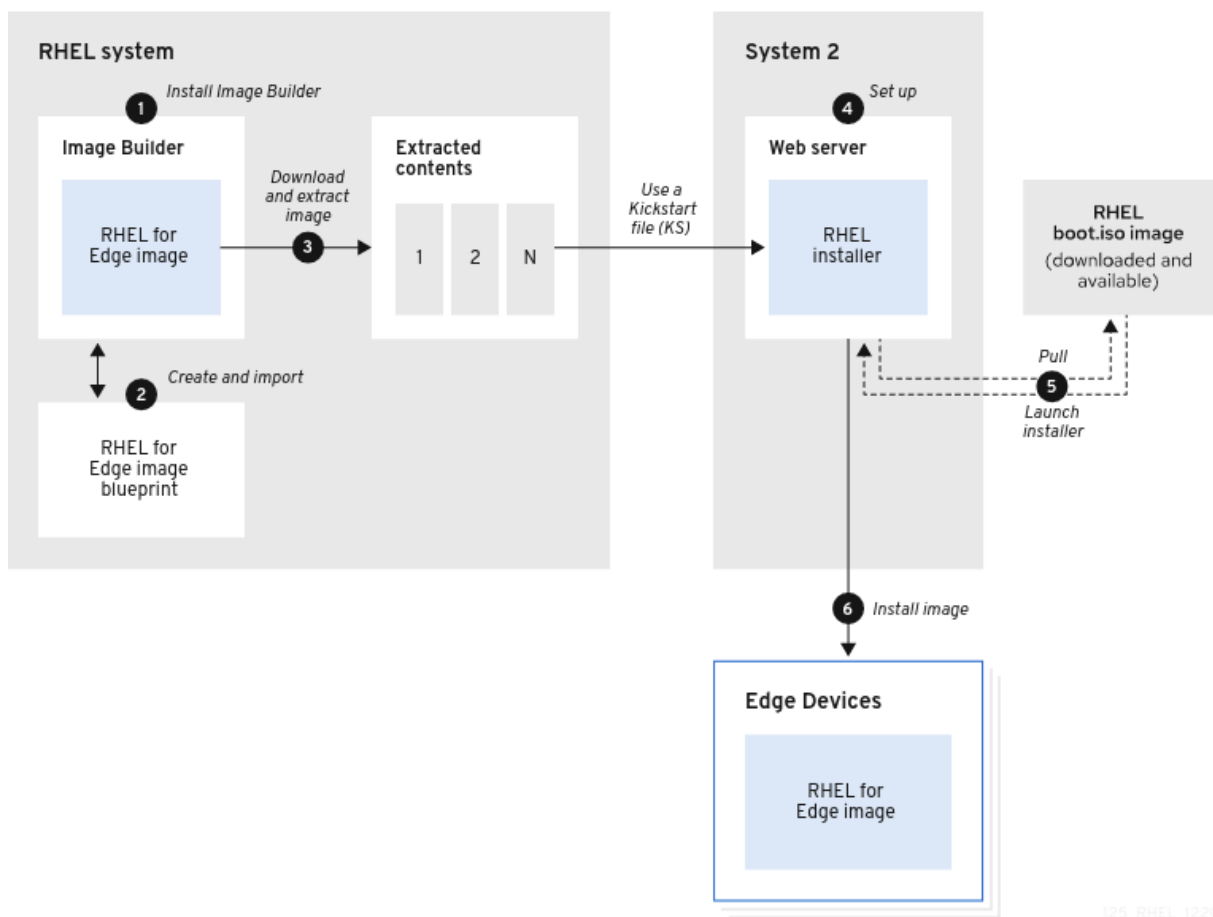
composer-cli 툴의 명령줄 인터페이스를 통해 Image Builder에 액세스하거나 RHEL 웹 콘솔에서 그래픽 사용자 인터페이스를 사용할 수 있습니다.

다음 고급 단계를 수행하여 RHEL for Edge 이미지를 작성하고 배포할 수 있습니다.

1. RHEL 시스템 설치 및 등록
2. 이미지 빌더 설치
3. 이미지 빌더를 사용하여 RHEL for Edge 이미지용 블루프린트 만들기
4. 이미지 빌더에서 RHEL for Edge 블루프린트 가져오기
5. RHEL for Edge 이미지 생성
6. RHEL for Edge 이미지 파일 다운로드
7. 웹 서버 설정
8. RHEL boot.iso 이미지 다운로드
9. 웹 서버를 사용하여 에지 장치에 RHEL for Edge 이미지를 설치합니다.

다음 다이어그램은 RHEL for Edge 네트워크 이미지 배포 워크플로를 나타냅니다.

그림 1.2. 네트워크 기반 환경에서 RHEL for Edge 배포



1.5. RHEL RPM 이미지와 RHEL FOR EDGE 이미지 간의 차이점

기존 패키지 기반 RPM 형식으로 RHEL 시스템 이미지를 생성하고 RHEL for Edge (rpm-ostree) 이미지를 생성할 수 있습니다.

기존 패키지 기반 RPM을 사용하여 기존 데이터 센터에 RHEL을 배포할 수 있습니다. 그러나 RHEL for Edge 이미지를 사용하면 기존 데이터센터 이외의 서버에 RHEL을 배포할 수 있습니다. 이러한 서버에는 데이터가 생성된 소스에 가장 근접하게 많은 양의 데이터를 처리하는 시스템이 포함됩니다.

다음 표를 참조하여 RHEL for Edge 이미지의 패키지 기반 RHEL RPM 이미지와 다른지 확인하십시오.

표 1.2. RHEL RPM 이미지와 RHEL for Edge 이미지 간의 차이점

키 속성	RHEL RPM 이미지	RHEL for Edge 이미지
OS 어셈블리	패키지를 로컬로 어셈블하여 이미지를 구성할 수 있습니다.	패키지는 시스템에 설치할 수 있는 ostree에서 어셈블됩니다.

OS 업데이트	dnf update 를 사용하여 활성화된 리포지토리에서 사용 가능한 업데이트를 적용할 수 있습니다.	/etc/ostree/remotes.d/ 의 원격 ostree에서 새 커밋을 사용할 수 있는 경우 rpm-ostree upgrade 를 사용하여 업데이트를 준비할 수 있습니다. 업데이트는 시스템 재부팅 시 적용됩니다.
리포지터리	패키지에 DNF 리포지터리가 포함되어 있습니다	패키지에는 Ostree 원격 리포지토리가 포함되어 있습니다.
사용자 액세스 권한	읽기 쓰기	읽기 전용 (/usr)
데이터 지속성	이미지가 아닌 tmpfs 마운트 지점에 마운트할 수 있습니다.	/etc 및 /var 는 읽기-쓰기가 활성화되어 있으며 데이터 유지를 포함합니다.

2장. 이미지 빌더 설정

이미지 빌더를 사용하여 에지 이미지에 대해 사용자 지정된 RHEL을 생성합니다. RHEL 시스템에 이미지 빌더를 설치한 후 RHEL 웹 콘솔에서 Image Builder를 애플리케이션으로 사용할 수 있습니다. **composer-cli** 툴의 명령줄 인터페이스를 통해 Image Builder에 액세스할 수도 있습니다.



참고

가상 머신에 이미지 빌더를 설치하는 것이 좋습니다.

이미지 빌더를 설치하려는 환경에서 먼저 시스템 요구 사항을 충족한 후 설치합니다.

2.1. 이미지 빌더 시스템 요구 사항

이미지 빌더가 실행되는 환경(예: 가상 머신)은 다음 표에 나열된 요구 사항을 충족해야 합니다.

표 2.1. 이미지 빌더 시스템 요구 사항

매개변수	최소 필수 값
시스템 유형	전용 가상 머신
프로세서	2개의 코어
메모리	4GiB
디스크 공간	20GiB
액세스 권한	관리자 수준(root)
네트워크	인터넷 연결



참고

20GiB 디스크 공간 요구 사항은 호스트에 Image Builder를 설치하고 실행하는 데 충분합니다. 이미지 빌드를 빌드하고 배포하려면 추가 전용 디스크 공간을 할당해야 합니다.

2.2. 이미지 빌더 설치

전용 가상 머신에 이미지 빌더를 설치하려면 다음 단계를 따르십시오.

사전 요구 사항

- 가상 시스템이 생성되어 전원이 켜집니다.
- RHEL을 설치하고 RHSM 또는 Red Hat Satellite에 등록되어 있습니다.

절차

- 가상 시스템에 다음 패키지를 설치합니다.

- osbuild-composer
- composer-cli
- cockpit-composer
- bash-completion
- firewalld

```
# dnf install osbuild-composer composer-cli cockpit-composer bash-completion firewalld
```

이미지 빌더는 RHEL 웹 콘솔에 애플리케이션으로 설치됩니다.

2. 가상 머신 재부팅

3. 웹 콘솔에 대한 액세스를 허용하도록 시스템 방화벽을 구성합니다.

```
# firewall-cmd --add-service=cockpit && firewall-cmd --add-service=cockpit --permanent
```

4. 이미지 빌더를 활성화합니다.

```
# systemctl enable osbuild-composer.socket --now
# systemctl enable cockpit.socket --now
```

osbuild-composer 및 cockpit 서비스는 첫 번째 액세스 시 자동으로 시작됩니다.

5. 재부팅하지 않고 **composer-cli** 명령의 자동 완성 기능이 즉시 작동하도록 셸 설정 스크립트를 로드합니다.

```
$ source /etc/bash_completion.d/composer-cli
```

추가 리소스

- [리포지토리 관리](#)

3장. 리포지토리 관리

3.1. 이미지 빌더 기본 시스템 리포지토리

osbuild-composer 백엔드는 **/etc/yum.repos.d/** 디렉터리에 있는 시스템 리포지토리를 상속하지 않습니다. 대신 **/usr/share/osbuild-composer/repositories** 디렉터리에 정의된 고유한 공식 리포지토리 세트가 있습니다. 공식 리포지토리를 재정의하려면 **/etc/osbuild-composer/repositories**에서 덮어쓰기를 정의해야 합니다. 이 디렉토리는 사용자 정의 덮어쓰기를 위한 것이며 여기에 있는 파일이 **/usr** 디렉토리의 파일보다 우선합니다.

구성 파일은 **/etc/yum.repos.d/**에 있는 파일에서 알려진 일반적인 DNF 리포지터리 형식이 아닙니다. 대신 간단한 JSON 파일입니다.

3.2. 시스템 리포지토리 덮어쓰기

다음 단계를 사용하여 **/etc/osbuild-composer/repositories** 디렉토리에서 리포지토리 재정의 구성할 수 있습니다.

사전 요구 사항

- 호스트 시스템에서 액세스할 수 있는 사용자 지정 리포지토리가 있습니다.

절차

1. 사용할 리포지토리 덮어쓰기가 포함된 디렉토리를 생성합니다.

```
$ sudo mkdir -p /etc/osbuild-composer/repositories
```

2. 다음 구조를 사용하여 JSON 파일을 생성합니다. 예를 들면 다음과 같습니다.

```
{
  "<ARCH>": [
    {
      "name": "baseos",
      "metalink": "",
      "baseurl": "http://mirror.example.com/composes/released/RHEL-9/9.0/BaseOS/x86_64/os/",
      "mirrorlist": "",
      "gpgkey": "-----BEGIN PGP PUBLIC KEY BLOCK-----\n\n (...)",
      "check_gpg": true,
      "metadata_expire": ""
    }
  ]
}
```

metalink, **mirrorlist** 또는 **baseurl** 중 하나만 지정합니다. 나머지 필드는 선택 사항입니다.

3. RHEL 버전에 해당하는 이름을 사용하여 JSON 파일을 저장합니다. 예를 들면 다음과 같습니다.

```
/etc/osbuild-composer/repositories/rhel-90.json
```

- a. 또는 **/usr/share/osbuild-composer/**에서 배포에 대한 JSON 파일을 복사하고 해당 콘텐츠를 수정할 수 있습니다.

- i. 리포지토리 파일을 생성한 디렉터리에 복사합니다.

```
$ cp /usr/share/osbuild-composer/repositories/rhel-version.json /etc/osbuild-composer/repositories/
```

rhel-version.json을 RHEL 버전으로 교체합니다(예: rhel-9.json).

4. 텍스트 편집기를 사용하여 **rhel-9.json** 파일의 **baseurl** 경로를 편집합니다. 예를 들어 다음과 같습니다.

```
$ vi /etc/osbuild-composer/repositories/rhel-9.json
```

5. **osbuild-composer.service** 를 다시 시작하십시오.

```
$ sudo systemctl restart osbuild-composer.service
```

결과적으로 리포지토리는 **/etc/yum.repos.d/redhat.repo** 파일에서 복사되는 올바른 URL을 가리킵니다.

추가 리소스

- **osbuild-composer** 에 표시되지 않는 리포지토리에서 사용 가능한 최신 RPM입니다.

3.3. 서브스크립션에 대한 지원을 사용하여 시스템 리포지토리 덮어쓰기

osbuild-composer 서비스는 **/etc/yum.repos.d/redhat.repo** 파일에 정의된 시스템 서브스크립션을 사용할 수 있습니다. **osbuild-composer** 에서 시스템 서브스크립션을 사용하려면 다음과 같은 리포지토리 재정의가 정의해야 합니다.

- **/etc/yum.repos.d/redhat.repo** 에 정의된 리포지토리과 동일한 **baseurl**입니다.
- JSON 오브젝트에 정의된 **"rhsm": true** 의 값입니다.

사전 요구 사항

- **/etc/yum.repos.d/redhat.repo**에 정의된 서브스크립션이 있는 시스템
- 리포지토리 덮어쓰기가 생성되어 있습니다. [시스템 리포지토리 덮어쓰기](#) 를 참조하십시오.

절차

1. **/etc/yum.repos.d/redhat.repo** 파일에서 **baseurl**을 가져옵니다.

```
[AppStream]
name = AppStream mirror example
baseurl = https://mirror.example.com/RHEL-9/9.0/AppStream/x86_64/os/
enabled = 1
gpgcheck = 0
sslverify = 1
sslcacert = /etc/pki/ca1/ca.crt
sslclientkey = /etc/pki/ca1/client.key
sslclientcert = /etc/pki/ca1/client.crt
metadata_expire = 86400
enabled_metadata = 0
```

2. 동일한 **baseurl**을 사용하고 **rhsm**을 true로 설정하도록 리포지토리 덮어쓰기를 구성합니다.

```
{
  "x86_64": [
    {
      "name": "AppStream mirror example",
      "baseurl": "https://mirror.example.com/RHEL-9/9.0/AppStream/x86_64/os/",
      "gpgkey": "-----BEGIN PGP PUBLIC KEY BLOCK-----\n\n (...)",
      "check_gpg": true,
      "rhsm": true
    }
  ]
}
```



참고

osbuild-composer는 **/etc/yum.repos.d/**에 정의된 리포지토리를 자동으로 사용하지 않습니다. 수동으로 시스템 리포지토리 덮어쓰기 또는 **composer-cli**를 사용하여 추가 **source**로 지정해야 합니다. 시스템 리포지토리 덮어쓰기는 일반적으로 "BaseOS" 및 "AppStream" 리포지토리에 사용되지만 **composer-cli** 소스는 다른 모든 리포지토리에 사용됩니다.

추가 리소스

- 이미지 빌더는 호스트가 Satellite 6에 등록되어 있을 때 CDN 리포지토리를 사용합니다.

4장. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 이미지 구성

이미지 빌더를 사용하여 사용자 지정 (OSTree 커밋) RHEL for Edge 이미지를 생성합니다.

이미지 빌더에 액세스하고 사용자 지정 RHEL for Edge 이미지를 생성하려면 RHEL 웹 콘솔 인터페이스 또는 명령줄 인터페이스를 사용할 수 있습니다. 이 섹션에서는 RHEL 웹 콘솔을 사용하여 이미지 빌더에 액세스하고 RHEL for Edge 이미지 생성에 대한 정보를 제공합니다.

다음 고급 단계를 수행하여 RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL for Edge 이미지를 구성할 수 있습니다.

1. RHEL 웹 콘솔에서 이미지 빌더 액세스
2. RHEL for Edge 이미지에 대한 블루프린트를 생성합니다. 다음 블루프린트를 생성할 수 있습니다.
 - "RHEL for Edge Commit (.tar)" 또는 "RHEL for Edge Container (.tar)" 이미지에 대한 사용자 지정이 있는 블루프린트
 - "RHEL for Edge Installer(.iso)" 이미지에 대한 블루프린트
3. RHEL for Edge 이미지 블루프린트의 사용자 계정 생성
4. RHEL for Edge 이미지를 만듭니다. 다음 이미지를 생성할 수 있습니다.
 - RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL for Edge 커밋 이미지 생성
 - RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL for Edge 컨테이너 이미지 생성
 - RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL for Edge 설치 프로그램 이미지 생성
5. RHEL for Edge 이미지 다운로드

4.1. RHEL 웹 콘솔에서 이미지 빌더에 액세스

RHEL 웹 콘솔에서 이미지 빌더에 액세스하려면 다음 사전 요구 사항을 충족했는지 확인한 후 절차를 따르십시오.

사전 요구 사항

- RHEL 시스템이 설치되어 있어야 합니다.
- 시스템에 대한 관리 권한이 있습니다.
- RHEL 시스템을 RHSM(Red Hat Subscription Manager) 또는 Red Hat Satellite 서버에 등록했습니다.
- 시스템 전원이 켜져 있고 네트워크를 통해 액세스할 수 있습니다.
- 시스템에 이미지 빌더를 설치했습니다.

절차

1. RHEL 시스템의 웹 브라우저에서 <https://localhost:9090/>에 액세스합니다.

2. 이미지 빌더에 원격으로 액세스하는 방법에 대한 자세한 내용은 [RHEL 9 웹 콘솔을 사용하여 시스템 관리](#) 문서를 참조하십시오.
3. 관리 사용자 계정을 사용하여 웹 콘솔에 로그인합니다.
4. 웹 콘솔의 왼쪽 메뉴에서 **앱**을 클릭합니다.
5. **이미지 빌더**를 클릭합니다.
오른쪽 창에서 이미지 빌더 대시보드가 열립니다. 이제 RHEL for Edge 이미지에 대한 블루프린트를 생성할 수 있습니다.

4.2. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 커밋 이미지의 블루프린트 생성

RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL for Edge 커밋 이미지에 대한 블루프린트를 생성하려면 다음 사전 요구 사항을 충족했는지 확인한 후 절차를 따르십시오.

사전 요구 사항

- RHEL 시스템에서 이미지 빌더 대시보드를 엽니다.

절차

1. 이미지 빌더 대시보드에서 **블루프린트 생성**을 클릭합니다.
블루프린트 생성 대화 상자가 열립니다.
2. 생성하려는 tekton에 대한 이름 및 설명을 지정합니다.
3. **생성**을 클릭합니다.
대시보드에는 **사용 가능한 구성 요소가 포함된 목록**이 표시됩니다.



참고

시스템이 RHSM에 등록해야 합니다. 그렇지 않으면 사용 가능한 구성 요소 목록이 "로드 중"으로 표시됩니다.

4. 특정 구성 요소를 검색하려면 **이름으로 필터링** 텍스트 상자에 구성 요소 이름을 입력하고 **Enter** 키를 누릅니다. **구성 요소 세부 정보** 창에 구성 요소 세부 정보 및 해당 종속 구성 요소가 표시됩니다.
5. **구성 요소 세부 정보** 창에서 **추가**를 클릭합니다.
웹 콘솔은 기본적으로 최신 버전을 선택합니다. 필요한 구성 요소 버전을 선택하려면 다음을 수행합니다.
 - a. 추가할 구성 요소를 클릭합니다.
 - b. 구성 요소 이름 옆에 있는 tekton 버튼을 클릭하고 **View**를 선택합니다.
 - c. **구성 요소 옵션** 아래의 **Version** 드롭다운 메뉴에서 필요한 버전을 선택합니다.
 - d. **적용 변경**을 클릭합니다.
사용 가능한 구성 요소 창에서 구성 요소를 제거하려면 구성 요소 이름에 대해 **-**를 클릭합니다.
6. **커밋**을 클릭하여 블루프린트를 저장합니다.

블루프린트 요약이 포함된 대화 상자가 열립니다.

7. 커밋을 클릭합니다.

8. 블루프린트로 돌아가기를 클릭합니다.

이미지 빌더 대시보드에는 사용자가 생성한 블루프린트가 나열됩니다.

4.3. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 컨테이너 이미지용 블루프린트 생성

RHEL 웹 콘솔의 Image Builder를 사용하여 에지 컨테이너 이미지에 대한 RHEL의 경고를 생성하고 패키지 및 같은 사용자 정의를 추가할 수 있습니다. 사용자 지정 기능이 있는 NetNamespace는 에지 컨테이너 이미지를 위한 RHEL을 생성하는 데 사용됩니다.

사전 요구 사항

- RHEL 시스템에서 이미지 빌더 대시보드를 엽니다.

절차

1. 이미지 빌더 대시보드에서 **블루프린트 생성**을 클릭합니다.

블루프린트 생성 대화 상자가 열립니다.

2. 생성하려는 블루프린트의 이름 및 설명을 지정합니다.

3. **생성**을 클릭합니다.

대시보드에 사용 가능한 구성 요소 목록이 표시됩니다.



참고

시스템이 RHSM에 등록해야 합니다. 그렇지 않으면 사용 가능한 구성 요소 목록이 "로드 중"으로 표시됩니다.

4. 선택 사항: 구성 요소를 사용하여 롤아웃을 사용자 지정합니다.

a. 특정 구성 요소를 검색하려면 **이름으로 필터링** 텍스트 상자에 구성 요소 이름을 입력하고

Enter 키를 누릅니다. **구성 요소 세부 정보** 페이지에는 구성 요소 세부 정보 및 해당 종속 구성 요소가 표시됩니다.

b. **구성 요소 세부 정보** 창에서 **추가**를 클릭합니다.

웹 콘솔은 기본적으로 최신 버전을 선택합니다. 필요한 구성 요소 버전을 선택하려면 다음을 수행합니다.

c. 추가할 구성 요소를 클릭합니다.

d. 구성 요소 이름 옆에 있는 **tekton** 버튼을 클릭하고 **View**를 선택합니다.

e. **구성 요소 옵션** 아래의 **Version** 드롭다운 메뉴에서 필요한 버전을 선택합니다.

f. **적용 변경**을 클릭합니다.

사용 가능한 구성 요소 창에서 구성 요소를 제거하려면 구성 요소 이름에 대해 **-**를 클릭합니다.

5. **Commit** (커밋)을 클릭하여 번호판에 보류 중인 변경 사항을 저장합니다.

선택한 구성 요소 변경 사항이 있는 대화 상자가 나타납니다.

6. 커밋을 클릭합니다.

7. **블루프린트로 돌아가기**를 클릭합니다.

이미지 빌더 대시보드에는 사용자가 생성한 블루프린트가 나열됩니다.

4.4. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 설치 프로그램 이미지용 블루프린트 생성

scalability를 생성하여 **Edge Installer(.iso)** 이미지에 대한 RHEL 을 빌드하고 설치 시 시스템에 하나 이상의 사용자를 자동으로 생성하도록 사용자 계정을 지정할 수 있습니다. [RHEL for Edge 이미지 블루프린트의 관리자 계정 생성](#)을 참조하십시오.

사전 요구 사항

- RHEL 시스템에서 이미지 빌더 대시보드를 엽니다.

절차

1. 이미지 빌더 대시보드에서 **블루프린트 생성**을 클릭합니다.
블루프린트 생성 대화 상자가 열립니다.
2. 생성하려는 블루프린트의 이름 및 설명을 지정합니다.
3. **생성**을 클릭합니다.
4. **블루프린트로 돌아가기**를 클릭합니다.
이미지 빌더 대시보드에는 사용자가 생성한 블루프린트가 나열됩니다.
5. 에지 설치 프로그램 이미지에 대해 RHEL에 대해 생성한 NetNamespace를 클릭합니다.
6. **Customization** (사용자 지정) 탭에서 **Create user account** (사용자 계정 생성)를 클릭합니다.
마법사 창이 열립니다.
7. 사용자 계정 세부 정보를 입력하고 **생성**을 클릭합니다.
Image Builder는 사용자가 생성한 사용자 계정을 나열합니다.

4.5. RHEL FOR EDGE 이미지 블루프린트에 소스 추가

기본적으로 **appstream** 및 **baseos dnf** 공식 소스를 사용할 수 있습니다. Image Builder, **dnf repository**, **mirrorlist** 및 **metalink** 에 정의된 소스를 사용하여 사용자 지정 타사 리포지토리의 RPM 패키지를 포함하도록 이를 청사진에 추가할 수 있습니다. 이러한 소스는 전역적으로 사용되므로 모든 블루프린트에서 사용할 수 있습니다.

시스템 소스 리포지토리는 컴퓨터에 로컬로 설정되며 이미지 빌더에서 비활성화하거나 제거할 수 없습니다. 추가 사용자 지정 소스를 추가할 수 있으므로 시스템에서 사용 가능한 시스템 소스 이외의 다른 콘텐츠에 액세스할 수 있습니다. 호스트 시스템에서 액세스할 수 있는 RPM 리포지토리는 소스로 유효합니다. 시스템 소스를 비활성화할 수 없습니다.

다음 단계를 수행하여 블루프린트에 소스를 추가합니다.

사전 요구 사항

- 브라우저에서 RHEL 웹 콘솔의 이미지 빌더 인터페이스를 열었습니다.

절차

1. 오른쪽 상단에 있는 **소스 관리** 버튼을 클릭합니다.
사용 가능한 소스, 이름 및 설명이 포함된 팝업 창이 표시됩니다.
2. 팝업 창 오른쪽에서 **소스 추가** 버튼을 클릭합니다.
3. 원하는 **소스 이름**, **소스 경로**, **소스 유형**을 추가합니다.
선택적으로 **보안 필드**와 관련된 확인란을 선택합니다.
 - a. **SSL 인증서** - 리포지토리의 ID를 확인하고 암호화된 연결을 활성화합니다.
 - b. **GPG 키** - 이 리포지토리에서 사용할 수 있는 RPM 패키지 서명의 유효성을 검사합니다.
4. **소스 추가**를 클릭합니다. 화면에 사용 가능한 소스 창이 표시되고 추가한 소스 목록이 표시됩니다.
따라서 새 시스템 소스를 사용하거나 원하는 변경 사항을 사용할 수 있습니다.

4.6. RHEL FOR EDGE 이미지 블루프린트의 관리자 계정 생성

Image Builder를 사용하여 생성한 Edge 이미지의 RHEL에는 root 계정이 잠겼고 다른 계정이 포함되어 있지 않습니다. 이미지 빌더를 사용하면 이미지 빌더를 사용하여 에지용 RHEL에 대한 암호를 사용하여 사용자 계정을 생성할 수 있으므로, 해당 이미지를 위해 RHEL for Edge 이미지에 로그인할 수 있습니다. 관리 사용자 계정의 경우 암호 기반 액세스 또는 SSH-키 기반 액세스 권한이 있을 수 있습니다.



참고

Kickstart를 사용하는 네트워크 기반 설치의 경우 Kickstart를 사용하여 사용자 계정을 만들 수도 있습니다.

사전 요구 사항

- 사용자 계정을 만드는 데 사용할 SSH 키를 생성했습니다.
- RHEL 웹 콘솔에서 이미지 빌더 대시보드에 액세스했습니다.
- RHEL for Edge 이미지용 블루프린트를 생성했습니다.

절차

1. 이미지 빌더 대시보드에서 RHEL for Edge 이미지의 블루프린트를 찾습니다.
필요한 블루프린트를 검색하려면 이름으로 필터링 텍스트 상자에서 블루프린트 이름을 지정 한 다음 **Enter**를 누릅니다.
2. 블루프린트 이름을 클릭합니다.
이미지 빌더에 블루프린트 세부 정보가 표시됩니다.
3. 사용자 지정 탭에서 **사용자 계정 생성**을 클릭합니다.
4. 사용자 계정 만들기 대화 상자에서 사용자 계정에 필요한 세부 정보와 암호를 지정합니다.
암호 기반 액세스의 경우 사용자 계정의 암호를 지정합니다.

SSH 기반 액세스의 경우 사용자 계정에 대한 SSH 키를 지정합니다.



참고

생성 중인 사용자 계정에 관리자 권한을 제공하려면 **서버 관리자** 확인란을 선택해야 합니다.

이미지 빌더에서 지정된 사용자 계정을 생성하고 세부 정보를 표시합니다.

필요한 경우 추가 사용자 계정을 생성할 수 있습니다.

4.7. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 커밋 이미지 생성

"**RHEL for Edge Commit(.tar)**"을 선택하여 네트워크 기반 배포를 위해 RHEL for Edge Commit을 생성할 수 있습니다. 전체 운영 체제가 포함되어 있어도 "**Edge Commit(.tar)**" 커밋 이미지 유형은 직접 부팅할 수 없습니다. 커밋 이미지 유형을 부팅하려면 실행 중인 컨테이너에 배포해야 합니다.

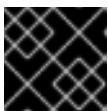
RHEL 웹 콘솔에서 이미지 빌더를 사용하여 에지 Commit 이미지의 RHEL을 생성하려면 다음 단계를 따르십시오.

사전 요구 사항

- RHEL 시스템에서 이미지 빌더 대시보드에 액세스했습니다.
- RHEL for Edge 커밋 이미지의 블루프린트를 생성했습니다.

절차

1. 이미지 빌더 대시보드에서 **에지 Commit 이미지용 RHEL**에 대해 생성한 options에 대해 **Create Image**를 클릭합니다.
특정 요건을 검색하려면 **Filter By Name** (이름으로 필터) 텍스트 상자에 Quarkus 이름을 입력한 다음 **Enter**를 누릅니다.
2. **이미지 생성** 창에서 다음 단계를 수행합니다.
3. **유형** 드롭다운 목록에서 네트워크 기반 배포에 대해 "**RHEL for Edge Commit(.tar)**"을 선택합니다.
4. 다음 인수 중 **하나**를 지정합니다. **리포지토리 URL** 또는 **상위 커밋**.



중요

리포지토리 URL과 상위 커밋 인수를 함께 사용하면 오류가 발생합니다.

- a. **리포지토리 URL**: 이미지에 포함할 커밋의 OSTree 리포지토리에 대한 URL을 지정합니다. 예: <http://10.0.2.2:8080/repository/>.
 - b. **parent commit**: 이전 커밋을 지정하거나 현재 커밋이 없는 경우 비워 둡니다.
5. **Ref** textbox에서 커밋이 생성되는 위치에 대한 참조 경로를 지정합니다. 기본적으로 웹 콘솔은 **rhel/9/\$ARCH/edge**를 지정합니다. "\$ARCH" 값은 호스트 시스템에 의해 결정됩니다.
 6. **생성**을 클릭합니다.
이미지 빌더는 생성한 리더십에 대한 RHEL for Edge Commit 이미지를 생성하기 시작합니다.

7. RHEL에서 에지 Commit 이미지 생성 진행 상황을 확인하려면 다음을 수행합니다.

- a. 이동 경로의 **ceilometer** 이름을 클릭합니다.
- b. **이미지** 탭을 클릭합니다.



참고

이미지 생성 프로세스를 완료하는 데 최대 20분이 걸립니다. 이미지 생성 프로세스를 중단하려면 More Options 메뉴에서 **Stop** (중지)을 클릭합니다.

이미지 생성 프로세스가 완료되면 결과 **"RHEL for Edge Commit(.tar)"** 이미지를 다운로드 할 수 있습니다.

추가 리소스

- [Edge 이미지용 RHEL 다운로드](#)

4.8. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 컨테이너 이미지 생성

"**RHEL for Edge Container(.tar)**"를 선택하여 네트워크 기반 배포가 아닌 이미지의 RHEL을 생성할 수 있습니다. **에지 컨테이너(.tar) 이미지 유형의 RHEL**은 OSTree 커밋을 생성하여 웹 서버가 있는 OCI 컨테이너에 포함합니다. 컨테이너를 시작하면 웹 서버에서 OSTree 리포지토리로 커밋을 제공합니다.

다음 절차의 단계에 따라 RHEL 웹 콘솔의 이미지 빌더를 사용하여 에지 컨테이너 이미지에 대한 RHEL을 생성합니다.

사전 요구 사항

- RHEL 시스템에서 이미지 빌더 대시보드에 액세스했습니다.
- 에지 컨테이너 이미지용 RHEL에 대한 devfile을 생성했습니다. [RHEL 웹 콘솔의 이미지 빌더를 사용하여 에지 컨테이너 이미지에 대한 RHEL 생성](#)을 참조하십시오.

절차

1. 이미지 빌더 대시보드에서 에지 이미지용으로 생성한 RHEL에 대해 **Create Image** (이미지 만들기)를 클릭합니다. 특정 요건을 검색하려면 **Filter By Name** (이름으로 필터) 텍스트 상자에 Quarkus 이름을 입력한 다음 **Enter** 를 누릅니다.
2. **이미지 생성** 창에서 다음 단계를 수행합니다.
3. **유형** 드롭다운 목록에서 **에지 컨테이너 (.tar)의 RHEL (RHEL)**을 선택합니다.
4. **Repository URL** textbox에서 이미지에 포함할 커밋의 OSTree 리포지토리에 대한 URL을 지정합니다. 예: <http://10.0.2.2:8080/repository/>.
5. **Parent commit** textbox에서 이전 커밋을 지정하거나, 이 커밋이 없는 경우 비워 둡니다.
6. **Ref** textbox에서 커밋이 생성되는 위치에 대한 참조 경로를 지정합니다. 기본적으로 웹 콘솔은 **rhel/9/\$ARCH/edge** 를 지정합니다. "\$ARCH" 값은 호스트 시스템에 의해 결정됩니다.
7. **생성**을 클릭합니다.

이미지 빌더는 생성한 credential에 대한 RHEL for Edge 컨테이너 이미지를 생성하기 시작합니다.

8. RHEL에서 에지 컨테이너 이미지 생성 진행 상황을 확인하려면 다음을 수행합니다.

- a. 이동 경로의 **ceilometer** 이름을 클릭합니다.
- b. **이미지** 탭을 클릭합니다.



참고

이미지 생성 프로세스를 완료하는 데 최대 20분이 걸립니다. 이미지 생성 프로세스를 중단하려면 More Options 메뉴에서 **Stop** (중지)을 클릭합니다.

이미지 생성 프로세스가 완료되면 결과 **"RHEL for Edge Container (.tar)"** 이미지를 다운로드할 수 있습니다.

추가 리소스

- [Edge 이미지용 RHEL 다운로드](#)

4.9. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 RHEL FOR EDGE 설치 프로그램 이미지 생성

"**RHEL for Edge Installer(.iso)**"를 선택하여 네트워크 기반 배포가 아닌 배포를 위해 RHEL for Edge Installer 이미지를 생성할 수 있습니다. **Edge 설치 프로그램(.iso) 이미지 유형은 RHEL for Edge Container (.tar)**에서 제공하는 실행 중인 컨테이너에서 OSTree 커밋 리포지토리를 가져오고 포함 된 OSTree 커밋을 사용하도록 구성된 kickstart 파일을 사용하여 설치 가능한 부팅 ISO 이미지를 생성합니다.

다음 절차의 단계에 따라 RHEL 웹 콘솔의 이미지 빌더를 사용하여 에지 이미지에 대한 RHEL을 생성합니다.

사전 요구 사항

- RHEL 시스템에서 이미지 빌더 대시보드에 액세스했습니다.
- 에지 설치 프로그램 이미지용 RHEL에 대한 청사진을 생성했습니다. [RHEL 웹 콘솔의 이미지 빌더를 사용하여 에지 설치 프로그램 이미지 생성](#)을 참조하십시오.
- 에지 컨테이너 이미지를 위한 RHEL을 생성하여 실행 중인 컨테이너에 로드했습니다. [비 네트워크 기반 배포의 경우 에지 컨테이너 이미지의 RHEL 생성](#)을 참조하십시오.

절차

1. 이미지 빌더 대시보드에서 **에지 설치 프로그램 이미지용**으로 RHEL을 만들었으며, **이미지 생성**을 클릭합니다.
특정 요건을 검색하려면 **Filter By Name** (이름으로 필터) 텍스트 상자에 Quarkus 이름을 입력한 다음 **Enter**를 누릅니다.
2. **이미지 생성** 창에서 다음 단계를 수행합니다.
 - a. **유형** 드롭다운 목록에서 **"RHEL for Edge Installer (.iso)"**를 선택합니다.

- b. **Repository URL** textbox에서 이미지에 포함할 커밋의 실행 중인 컨테이너 OSTree 리포지토리에 대한 URL을 지정합니다. 예: <http://10.0.2.2:8080/repository/>.
 - c. **Parent** 커밋 텍스트 상자, 이전 커밋을 지정하거나 이 커밋이 없는 경우 비워 둘 수 있습니다.
 - d. **Ref** 텍스트 상자의 참조 경로는 RHEL for Edge Container image compose의 **Ref**와 일치해야 합니다.
3. **생성**을 클릭합니다.
이미지 빌더는 사용자가 생성한 Credential에 대한 **RHEL for Edge Installer** 이미지를 생성하기 시작합니다.
 4. RHEL에서 에지 설치 프로그램 이미지 생성 진행 상황을 확인하려면 다음을 수행합니다.
 - a. 이동 경로의 **ceilometer** 이름을 클릭합니다.
 - b. **이미지** 탭을 클릭합니다.



참고

이미지 생성 프로세스를 완료하는 데 최대 20분이 걸립니다. 이미지 생성 프로세스를 중단하려면 **More Options** 메뉴에서 **Stop** (중지)을 클릭합니다.

이미지 생성 프로세스가 완료되면 결과 **RHEL for Edge Installer (.iso)** 이미지를 다운로드하여 ISO 이미지를 장치로 부팅할 수 있습니다.

추가 리소스

- [에지 이미지를 위한 RHEL 다운로드](#).

4.10. 에지 이미지 RHEL 다운로드

이미지 빌더에서 에지 이미지용 RHEL을 성공적으로 생성한 후 로컬 호스트에서 이미지를 다운로드합니다.

절차

이미지를 다운로드하려면 다음을 수행합니다.

1. **추가 옵션** 메뉴에서 **다운로드**를 클릭합니다.
이미지 빌더는 기본 다운로드 위치에 파일을 다운로드합니다.

다운로드한 파일은 에지 Commit 및 에지 컨테이너 이미지용 RHEL용 RHEL용 OSTree 리포지토리가 있는 **.tar** 파일 또는 OSTree 리포지토리가 있는 RHEL용 RHEL용 **.iso** 파일로 구성됩니다. 이 리포지토리에 있는 리포지토리 콘텐츠에 대한 정보 메타데이터가 포함된 커밋 및 **json** 파일이 포함되어 있습니다.

4.11. 추가 리소스

- [이미지 빌더 명령줄을 사용하여 에지 이미지용 RHEL 작성](#).

5장. IMAGE BUILDER 명령줄을 사용하여 에지 이미지용 RHEL 구성

이미지 빌더를 사용하여 에지 이미지용 사용자 지정 RHEL(OSTree 커밋)을 만들 수 있습니다.

이미지 빌더에 액세스하고 사용자 지정 RHEL for Edge 이미지를 생성하려면 RHEL 웹 콘솔 인터페이스 또는 명령줄 인터페이스를 사용할 수 있습니다. 이 장에서는 CLI를 사용하여 에지 이미지에 대한 RHEL을 만드는 방법을 설명합니다.

네트워크 기반 배포의 경우 CLI를 사용하여 에지 이미지에 대한 RHEL을 구성하는 워크플로우에는 다음과 같은 상위 수준 단계가 포함됩니다.

1. 에지 이미지 RHEL에 대한 devfile 생성
2. Edge Commit 이미지에 대한 RHEL 생성
3. Edge Commit 이미지용 RHEL 다운로드

비네트워크 기반 배포의 경우 CLI를 사용하여 에지 이미지에 대한 RHEL을 구성하는 워크플로우에는 다음과 같은 상위 수준 단계가 포함됩니다.

1. 에지 이미지 RHEL에 대한 devfile 생성
2. 에지 컨테이너 이미지용 RHEL 생성
3. 에지 설치 프로그램 이미지용 RHEL에 대한 청사진 생성
4. 엣지 설치 프로그램 이미지용 RHEL 생성
5. RHEL for Edge 이미지 다운로드

단계를 수행하려면 **composer-cli** 패키지를 사용합니다.



참고

root가 아닌 경우 **composer-cli** 명령을 실행하려면 **weldr** 그룹의 일부이거나 시스템에 대한 관리자 액세스 권한이 있어야 합니다.

5.1. 네트워크 기반 배포 워크플로

5.1.1. 이미지 빌더 명령줄 인터페이스를 사용하여 Edge Commit 이미지 청사진용 RHEL 생성

CLI를 사용하여 에지 커밋 이미지에 대한 RHEL을 생성합니다.

사전 요구 사항

- 기존 이메일은 없습니다. 이를 확인하기 위해 기존 tekton를 나열하십시오.

```
$ sudo composer-cli blueprints list
```

절차

1. 다음 콘텐츠를 사용하여 TOML 형식으로 일반 텍스트 파일을 생성합니다.

```
name = "blueprint-name"
description = "blueprint-text-description"
version = "0.0.1"
modules = [ ]
groups = [ ]
```

여기서,

- `openjdk-name` 은 `name`이며, `director-text-description`은 귀하의 인식에 대한 설명입니다.
 - `0.0.1` 은 Semantic Versioning scheme에 따른 버전 번호입니다.
 - 모듈은 이미지에 설치할 패키지 이름 및 일치하는 버전 글러스트를 설명합니다. 예를 들어 패키지 이름 = "tmux"이고 일치하는 버전은 `version = "2.9a"`입니다. 현재는 패키지와 모듈간에 차이가 없습니다.
 - 그룹은 이미지에 설치할 패키지 그룹입니다(예: 그룹 패키지 `anaconda-tools`). 이 시점에서 모듈과 그룹을 모르는 경우 비워 둡니다.
2. 필요한 패키지를 포함시키고 귀하의 요구 사항에 맞게 `director`에 다른 세부 사항을 사용자 정의 하십시오.

`Makefile`에 포함하려는 모든 패키지에 대해 파일에 다음 행을 추가합니다.

```
[[packages]]
name = "package-name"
version = "package-version"
```

여기서,

- `package-name`은 `httpd`, `gdb-doc` 또는 `coreutils`와 같은 패키지의 이름입니다.
 - `package-version`은 사용하려는 패키지의 버전 번호입니다. `package-version`은 다음 `dnf` 버전 사양을 지원합니다.
 - 특정 버전의 경우 `9.0`과 같은 정확한 버전 번호를 사용하십시오.
 - 사용 가능한 최신 버전의 경우 별표 `*`를 사용하십시오.
 - 최신 마이너 버전의 경우 `9.*`와 같은 형식을 사용하십시오.
3. 이미지 빌더 서버에 대한 `devfile`을 푸시(가져오기)합니다.

```
# composer-cli blueprints push blueprint-name.toml
```

4. 기존 `Makefile`을 나열하여 생성된 `inventory`가 성공적으로 푸시되고 있는지 확인합니다.

```
# composer-cli blueprints show BLUEPRINT-NAME
```

5. `Makefile` 및 해당 종속 항목에 나열된 구성 요소 및 버전이 유효한지 확인합니다.

```
# composer-cli blueprints depsolve blueprint-name
```

추가 리소스

- [지원되는 이미지 사용자 지정.](#)

5.1.2. 이미지 빌더 명령줄 인터페이스를 사용하여 엣지 커밋 이미지용 RHEL 생성

이미지 빌더 명령줄 인터페이스를 사용하여 엣지 Commit 이미지에 대한 RHEL을 생성하려면 다음 사전 요구 사항을 충족하고 절차를 따르십시오.

사전 요구 사항

- RHEL for Edge 커밋 이미지의 블루프린트를 생성했습니다.

절차

1. Edge Commit 이미지에 대한 RHEL을 만듭니다.

```
# composer-cli compose start blueprint-name image-type
```

여기서,

- Quarkus-name 은 엣지 opportunity 이름을 위한 RHEL입니다.
- image-type 은 **네트워크 기반 배포의 edge-commit**입니다.
쓰기 프로세스가 큐에 추가되었음을 확인합니다. 또한 생성된 이미지에 대한 UUID(Universally Unique Identifier) 번호를 표시합니다. UUID 번호를 사용하여 빌드를 추적합니다. 또한 추가 작업을 위해 UUID 번호를 편리하게 유지합니다.

2. 이미지 compose 상태를 확인합니다.

```
# composer-cli compose status
```

출력에는 다음 형식으로 상태가 표시됩니다.

```
<UUID> RUNNING date blueprint-name blueprint-version image-type
```



참고

이미지 생성 프로세스를 완료하는 데 최대 20분이 걸립니다.

이미지 생성 프로세스를 중단하려면 다음을 실행합니다.

```
# composer-cli compose cancel <UUID>
```

기존 이미지를 삭제하려면 다음을 실행합니다.

```
# composer-cli compose delete <UUID>
```

이미지가 준비되면 이미지를 다운로드하여 **네트워크 배포에서 사용할 수 있습니다.**

추가 리소스

- [이미지 빌더 명령줄을 사용하여 엣지 이미지용 RHEL 작성](#)

5.1.3. 이미지 빌더 명령줄 인터페이스를 사용하여 상위 커밋으로 에지 이미지 업데이트 생성

예를 들어 새 패키지를 추가한 후 이 새 패키지로 기존 RHEL for Edge 이미지를 업데이트하려면 상위 커밋 ID를 사용하여 에지 Commit(.tar) 이미지에 업데이트된 RHEL을 생성할 수 있습니다.

Image Builder 명령줄 인터페이스를 사용하여 상위 커밋으로 에지 이미지의 RHEL을 생성하려면 다음 사전 요구 사항을 충족한 다음 절차를 따르십시오.

사전 요구 사항

- 에지 이미지용 RHEL의 기존 credential을 업데이트했습니다.
- 기존 RHEL for Edge 이미지(OSTree 커밋)가 있습니다. [에지 이미지 커밋용 RHEL 추출](#)을 참조하십시오.

절차

1. 에지 이미지에 대한 RHEL을 만듭니다.

```
# composer-cli compose start-ostree --ref rhel/9/x86_64/edge --parent parent-OSTree-commit-id blueprint-name image-type
```

여기서,

- --ref 는 ostree 리포지토리를 빌드하는 데 사용하는 것과 동일한 값입니다.
- --parent 는 OSTree 상위 커밋입니다.
- Quarkus-name 은 에지 opportunity 이름을 위한 RHEL입니다.
- image-type 은 네트워크 기반 배포를 위한 **edge-commit**입니다.



참고

--parent 인수는 RHEL for Edge Commit(.tar) 이미지 유형에서만 사용할 수 있습니다. **--url** 및 **--parent** 인수를 함께 사용하면 오류가 발생합니다.

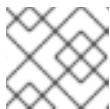
쓰기 프로세스가 큐에 추가되었음을 확인합니다. 또한 생성된 이미지에 대한 UUID(Universally Unique Identifier) 번호를 표시합니다. UUID 번호를 사용하여 빌드를 추적합니다. 또한 추가 작업을 위해 UUID 번호를 편리하게 유지합니다.

2. 이미지 compose 상태를 확인합니다.

```
# composer-cli compose status
```

출력에는 다음 형식으로 상태가 표시됩니다.

```
<UUID> RUNNING date blueprint-name blueprint-version image-type
```



참고

이미지 생성 프로세스를 완료하는 데 몇 분이 걸립니다.

(선택 사항) 이미지 생성 프로세스를 중단하려면 다음을 실행합니다.

```
# composer-cli compose cancel <UUID>
```

(선택 사항) 기존 이미지를 삭제하려면 다음을 실행합니다.

```
# composer-cli compose delete <UUID>
```

이미지 생성이 완료되면 기존 ostree 배포를 업그레이드하려면 다음이 필요합니다.

- 리포지토리를 설정합니다. [예지 이미지용 RHEL](#) 배포를 참조하십시오.
- 이 리포지토리를 원격, 즉 ostree 콘텐츠를 호스팅하는 http 또는 https 엔드포인트로 추가합니다.
- 기존 실행 중인 인스턴스에 새 OSTree 커밋을 가져옵니다. [예지 이미지 업데이트용 RHEL](#) 배포를 [수동으로](#) 참조하십시오.

추가 리소스

- [명령줄 인터페이스에서 이미지 빌더를 사용하여 시스템 이미지 생성](#)
- [이미지 빌더 명령줄 인터페이스를 사용하여 Edge 이미지용 RHEL 다운로드](#)

5.1.4. Image Builder 명령줄 인터페이스를 사용하여 예지 이미지의 RHEL 다운로드

Image Builder 명령줄 인터페이스를 사용하여 예지 이미지의 RHEL 을 다운로드하려면 다음 사전 요구 사항을 충족했는지 확인한 후 절차를 따르십시오.

사전 요구 사항

- 예지 이미지용 RHEL 을 생성했습니다.

절차

1. RHEL for Edge 이미지 상태를 검토합니다.

```
# composer-cli compose status
```

출력에 다음이 표시되어야 합니다.

```
$ <UUID> FINISHED date blueprint-name blueprint-version image-type
```

2. 이미지를 다운로드합니다.

```
# composer-cli compose image <UUID>
```

이미지 빌더는 이미지를 **tar** 파일로 현재 디렉터리에 다운로드합니다.

UUID 번호와 이미지 크기가 함께 표시됩니다.

```
$ <UUID>-commit.tar: size MB
```

이미지에는 리포지토리 콘텐츠에 대한 정보 메타데이터가 포함된 커밋 및 **json** 파일이 포함되어 있습니다.

5.2. 비 네트워크 기반 배포 워크플로

5.2.1. 이미지 빌더 CLI를 사용하여 예지 컨테이너 이미지 청사진용 RHEL 생성

예지 컨테이너 이미지에 대한 RHEL에 대한 청사진을 생성하려면 다음 단계를 수행합니다.

절차

1. 다음 콘텐츠를 사용하여 TOML 형식으로 일반 텍스트 파일을 생성합니다.

```
name = "blueprint-name"
description = "blueprint-text-description"
version = "0.0.1"
modules = [ ]
groups = [ ]
```

여기서,

- `openjdk-name` 은 `name` 이며, `director-text-description` 은 귀하의 인식에 대한 설명입니다.
 - `0.0.1` 은 Semantic Versioning scheme에 따른 버전 번호입니다.
 - 모듈은 이미지에 설치할 패키지 이름 및 일치하는 버전 글러스트를 설명합니다. 예를 들어 패키지 이름 = `"tmux"` 이고 일치하는 버전은 `version = "2.9a"` 입니다.
현재는 패키지와 모듈간에 차이가 없습니다.
 - 그룹은 이미지에 설치할 패키지 그룹입니다(예: 그룹 패키지 `anaconda-tools`).
이 시점에서 모듈과 그룹을 모르는 경우 비워 둡니다.
2. 필요한 패키지를 포함시키고 귀하의 요구 사항에 맞게 `director`에 다른 세부 사항을 사용자 정의 하십시오.
`Makefile`에 포함하려는 모든 패키지에 대해 파일에 다음 행을 추가합니다.

```
[[packages]]
name = "package-name"
version = "package-version"
```

여기서,

- `package-name`은 `httpd`, `gdb-doc` 또는 `coreutils`와 같은 패키지의 이름입니다.
 - `package-version`은 사용하려는 패키지의 버전 번호입니다.
`package-version`은 다음 `dnf` 버전 사양을 지원합니다.
 - 특정 버전의 경우 `9.0`과 같은 정확한 버전 번호를 사용하십시오.
 - 사용 가능한 최신 버전의 경우 별표 `*`를 사용하십시오.
 - 최신 마이너 버전의 경우 `9.*`와 같은 형식을 사용하십시오.
3. 이미지 빌더 서버에 대한 `devfile`을 푸시(가져오기)합니다.

```
# composer-cli blueprints push blueprint-name.toml
```

4. 기존 Makefile 을 나열하여 생성된 inventory가 성공적으로 푸시되고 있는지 확인합니다.

```
# composer-cli blueprints show BLUEPRINT-NAME
```

5. Makefile 및 해당 종속 항목에 나열된 구성 요소 및 버전이 유효한지 확인합니다.

```
# composer-cli blueprints depsolve blueprint-name
```

추가 리소스

- [지원되는 이미지 사용자 지정.](#)

5.2.2. 이미지 빌더 CLI를 사용하여 엣지 컨테이너 이미지용 RHEL 생성

이미지 빌더 명령줄 인터페이스를 사용하여 엣지 컨테이너 이미지에 대한 RHEL을 생성하려면 다음 사전 요구 사항을 충족하고 절차를 따르십시오.

사전 요구 사항

- 엣지 컨테이너 이미지용 RHEL에 대한 devfile을 생성했습니다.

절차

1. Edge 컨테이너 이미지에 대한 RHEL을 생성합니다.

```
# composer-cli compose start-ostree --ref rhel/9/x86_64/edge --url URL-OSTree-repository  
blueprint-name image-type
```

여기서,

- --ref 는 고객이 ostree 저장소를 구축하는 데 사용한 것과 동일한 값입니다.
 - -- URL 은 이미지에 포함할 커밋의 OSTree 리포지토리에 대한 URL입니다. 예:
<http://10.0.2.2:8080/repository/>. [Edge 이미지용 RHEL을 설치할 웹 서버 설정을](#) 참조하십시오.
 - Quarkus-name 은 엣지 opportunity 이름을 위한 RHEL입니다.
 - image-type 은 네트워크 기반이 아닌 배포의 **edge-container**입니다.
쓰기 프로세스가 큐에 추가되었음을 확인합니다. 또한 생성된 이미지에 대한
UUID(Universally Unique Identifier) 번호를 표시합니다. UUID 번호를 사용하여 빌드를 추적합
니다. 또한 추가 작업을 위해 UUID 번호를 편리하게 유지합니다.
2. 이미지 compose 상태를 확인합니다.

```
# composer-cli compose status
```

출력에는 다음 형식으로 상태가 표시됩니다.

```
<UUID> RUNNING date blueprint-name blueprint-version image-type
```



참고

이미지 생성 프로세스를 완료하는 데 최대 20분이 걸립니다.

이미지 생성 프로세스를 중단하려면 다음을 실행합니다.

```
# composer-cli compose cancel <UUID>
```

기존 이미지를 삭제하려면 다음을 실행합니다.

```
# composer-cli compose delete <UUID>
```

이미지가 준비되면 비 네트워크 배포에 사용할 수 있습니다

추가 리소스

- [이미지 빌더 명령줄을 사용하여 에지 이미지용 RHEL 작성](#)

5.2.3. 이미지 빌더 CLI를 사용하여 에지 설치 프로그램 이미지 청사진용 RHEL 생성

이제 에지 설치 관리자의 RHEL에서 사용자 계정을 지정할 수 있습니다. 설치 시 시스템에 사용자를 생성합니다. 에지 설치 프로그램 이미지용 RHEL에 대한 청사진을 생성하려면 다음 단계를 수행합니다.

절차

1. 다음 콘텐츠를 사용하여 TOML 형식으로 일반 텍스트 파일을 생성합니다.

```
name = "blueprint-installer"
description = "blueprint-for-installer-image"
version = "0.0.1"

[[customizations.user]]
name = "user"
description = "account"
password = "user-password"
key = "user-ssh-key "
home = "path"
groups = ["user-groups"]
```

여기서,

- openjdk-name 은 name이며, director-text-description은 귀하의 인식에 대한 설명입니다.
 - 0.0.1 은 Semantic Versioning scheme에 따른 버전 번호입니다.
2. 이미지 빌더 서버에 대한 devfile을 푸시(가져오기)합니다.

```
# composer-cli blueprints push blueprint-name.toml
```

3. 기존 Makefile을 나열하여 생성된 inventory가 성공적으로 푸시되고 있는지 확인합니다.

```
# composer-cli blueprints show blueprint-name
```

4. Makefile 및 해당 종속 항목에 나열된 구성 요소 및 버전이 유효한지 확인합니다.

```
# composer-cli blueprints depsolve blueprint-name
```

5.2.4. 네트워크 기반 배포에 명령줄 인터페이스를 사용하여 에지 설치 프로그램 이미지 생성

이미지 빌더 명령줄 인터페이스를 사용하여 **OSTree** 커밋을 포함하는 에지 설치 프로그램 이미지용 RHEL을 생성하려면 다음 사전 요구 사항을 충족한 다음 절차를 따르십시오.

사전 요구 사항

- 에지 설치 프로그램 이미지용 RHEL에 대한 청사진을 생성했습니다.
- Edge Edge Container 이미지를 위한 RHEL을 생성하고 웹 서버를 사용하여 배포했습니다.

절차

1. 에지 설치 프로그램 이미지용 RHEL 생성을 시작합니다.

```
# composer-cli compose start-ostree --ref rhel/9/x86_64/edge --url URL-OSTree-repository  
blueprint-name image-type
```

여기서,

- `ref` 는 ostree 리포지토리를 빌드하는 데 사용한 것과 동일한 값입니다.
 - `URL-OSTree-repository` 는 이미지에 포함할 커밋의 OSTree 리포지토리에 대한 URL입니다.
예: <http://10.0.2.2:8080/repository/>. Edge 이미지용 RHEL을 설치할 웹 서버 설정을 참조하십시오.
 - `Blueprint-name` 은 에지 설치 프로그램 청사진 이름에 대한 RHEL입니다.
 - `image-type` 은 **edge-installer** 입니다.
쓰기 프로세스가 큐에 추가되었음을 확인합니다. 또한 생성된 이미지에 대한 UUID(Universally Unique Identifier) 번호를 표시합니다. UUID 번호를 사용하여 빌드를 추적합니다. 또한 추가 작업을 위해 UUID 번호를 편리하게 유지합니다.
2. 이미지 compose 상태를 확인합니다.

```
# composer-cli compose status
```

명령 출력은 다음 형식으로 상태를 표시합니다.

```
<UUID> RUNNING date blueprint-name blueprint-version image-type
```



참고

이미지 생성 프로세스를 완료하는 데 몇 분이 걸립니다.

이미지 생성 프로세스를 중단하려면 다음을 실행합니다.

```
# composer-cli compose cancel <UUID>
```

-

기존 이미지를 삭제하려면 다음을 실행합니다.

```
# composer-cli compose delete <UUID>
```

이미지가 준비되면 **비 네트워크 배포에 사용할 수 있습니다**

5.2.5. 이미지 빌더 CLI를 사용하여 에지 설치 프로그램 이미지용 RHEL 다운로드

이미지 빌더 명령줄 인터페이스를 사용하여 에지 설치 프로그램 이미지의 RHEL을 다운로드하려면 다음 사전 요구 사항을 충족한 다음 절차를 따르십시오.

사전 요구 사항

- 에지 설치 프로그램 이미지용 RHEL을 생성했습니다.

절차

1. RHEL for Edge 이미지 상태를 검토합니다.

```
# composer-cli compose status
```

출력에 다음이 표시되어야 합니다.

```
$ <UUID> FINISHED date blueprint-name blueprint-version image-type
```

2. 이미지를 다운로드합니다.

```
# composer-cli compose image <UUID>
```

이미지 빌더는 이미지를 **.iso** 파일로 현재 디렉터리에 다운로드합니다.

UUID 번호와 이미지 크기가 함께 표시됩니다.

```
$ <UUID>-boot.iso: size MB
```

결과 이미지는 부팅 가능한 ISO 이미지입니다.

5.3. 지원되는 이미지 사용자 지정

assemblies에서는 여러 이미지 사용자 지정이 지원됩니다. 이러한 옵션을 사용하려면 처음에 탄력성에서 사용자 지정을 구성하고 이미지 빌더로 가져오기(push)해야 합니다.



참고

이러한 사용자 지정은 현재 웹 콘솔 내에서 지원되지 않습니다.

이미지 호스트 이름 설정

```
[customizations]
hostname = "baseimage"
```

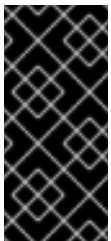
결과 시스템 이미지에 대한 사용자 사양

```
[[customizations.user]]
name = "USER-NAME"
description = "USER-DESCRIPTION"
password = "PASSWORD-HASH"
key = "PUBLIC-SSH-KEY"
home = "/home/USER-NAME/"
shell = "/usr/bin/bash"
groups = ["users", "wheel"]
uid = NUMBER
gid = NUMBER
```



참고

GID는 선택 사항이며 이미지에 이미 존재해야하거나, 패키지에 의해 생성되거나, locally **[[customizations.group]]** 항목에 의해 생성되어야 합니다.



중요

해시를 생성하려면 시스템에 **python3** 을 설치해야 합니다. 다음 명령은 **python3** 패키지를 설치합니다.

```
# dnf install python3
```

passphrase -HASH 를 실제 암호 해시로 바꿉니다. 해시를 생성하려면 다음과 같은 명령을 사용합니다.

```
$ python3 -c 'import crypt,getpass;pw=getpass.getpass();print(crypt.crypt(pw) if
(pw==getpass.getpass("Confirm: ")) else exit())'
```

PUBLIC-SSH-KEY 를 실제 공개 키로 교체합니다.

다른 자리 표시자를 적절한 값으로 바꿉니다.

필요에 따라 행을 벗어나 사용자 이름만 있으면 됩니다.

포함할 모든 사용자에게 대해 이 블록을 반복합니다.

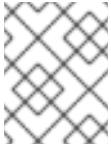
결과 시스템 이미지의 그룹 사양

```
[[customizations.group]]
name = "GROUP-NAME"
gid = NUMBER
```

포함할 모든 그룹에 대해 이 블록을 반복합니다.

기존 사용자 ssh 키 설정

```
[[customizations.sshkey]]
user = "root"
key = "PUBLIC-SSH-KEY"
```



참고

이 옵션은 기존 사용자에게만 적용됩니다. 사용자를 생성하고 ssh 키를 설정하려면 이 섹션의 결과 시스템 이미지 사용자 지정을 참조하십시오.

기본값에 커널 부팅 매개변수 옵션 추가

```
[customizations.kernel]
append = "KERNEL-OPTION"
```

기본적으로 이미지 빌더는 이미지에 기본 커널을 빌드합니다. 그러나 NetNamespace에서 다음과 같은 구성으로 커널을 사용자 지정할 수 있습니다.

```
[customizations.kernel]
name = "KERNEL-rt"
```

이미지에 사용할 커널 이름 정의

```
[customizations.kernel.name]
name = "KERNEL-NAME"
```

결과 시스템 이미지에 대해 시간대 및 NTP(Network Time Protocol) 서버 설정

```
[customizations.timezone]
timezone = "TIMEZONE"
ntpservers = "NTP_SERVER"
```

시간대를 설정하지 않으면 시스템은 Universal Time, Coordinated (UTC) 를 기본값으로 사용합니다. NTP 서버 설정은 선택 사항입니다.

결과 시스템 이미지의 로케일 설정 설정

```
[customizations.locale]
languages = ["LANGUAGE"]
keyboard = "KEYBOARD"
```

언어 및 키보드 옵션을 모두 설정해야 합니다. 여러 언어를 추가할 수 있습니다. 추가하는 첫 번째 언어는 기본 언어이고 다른 언어는 보조가 됩니다.

결과 시스템 이미지의 방화벽 설정

```
[customizations.firewall]
port = ["PORTS"]
```

숫자 포트 또는 **/etc/services** 파일의 해당 이름을 사용하여 목록을 활성화할 수 있습니다.

방화벽 서비스 사용자 지정

사용 가능한 방화벽 서비스를 검토합니다.

```
$ firewall-cmd --get-services
```

NetNamespace의 section **customizations.firewall.service** 에서 사용자 지정할 방화벽 서비스를 지정합니다.

```
[customizations.firewall.services]
enabled = ["SERVICES"]
disabled = ["SERVICES"]
```

firewall.services 에 나열된 서비스는 **/etc/services** 파일에서 사용할 수 있는 이름과 다릅니다.

선택적으로 생성하려는 시스템 이미지의 방화벽 서비스를 사용자 지정할 수 있습니다.



참고

방화벽 서비스를 사용자 지정하지 않으려면 NetNamespace에서 **[customizations.firewall]** 및 **[customizations.firewall.services]** 섹션을 생략합니다.

부팅 시 활성화할 서비스 설정

```
[customizations.services]
enabled = ["SERVICES"]
disabled = ["SERVICES"]
```

부팅 시 활성화할 서비스를 제어할 수 있습니다. 일부 이미지 유형에는 이미 서비스가 활성화 또는 비활성화되어 이미지가 올바르게 작동하고 이 설정을 재정의할 수 없습니다.



참고

빌드가 시작될 때마다 리포지토리를 복제합니다. 기록이 많은 리포지토리를 참조하는 경우 상당한 양의 디스크 공간을 복제하고 사용하는 데 시간이 걸릴 수 있습니다. 또한 복제본은 일시적인 것이며 RPM 패키지가 생성된 후 제거됩니다.

사용자 정의 파일 시스템 구성 지정

용량에서 사용자 지정 파일 시스템 구성을 지정하여 기본 레이아웃 구성을 사용하는 대신 특정 디스크 레이아웃으로 이미지를 생성할 수 있습니다. **implicationss**에서 기본이 아닌 레이아웃 구성을 사용하면 다음과 같은 이점을 얻을 수 있습니다.

- 보안 벤치마크 컴플라이언스
 - 디스크 부족 오류 보호
 - performance
 - 기존 설정과 일관성
- bastion**에서 파일 시스템 구성을 사용자 지정합니다.

```
[[customizations.filesystem]]
mountpoint = "MOUNTPOINT"
size = MINIMUM-PARTITION-SIZE
```

다음 **마운트 지점** 과 해당 하위 디렉터리가 지원됩니다.

- /- 루트 마운트 지점

- `/var`
- `/home`
- `/opt`
- `/srv`
- `/usr`
- `/app`
- `/data`



참고

CLI를 사용하여 마운트 지점 사용자 지정은 RHEL 8.5 및 RHEL 9.0 배포에서만 지원됩니다. 초기 배포에서는 루트 파티션만 마운트 지점으로 지정하고 **size** 인수를 이미지 크기의 별칭으로 지정할 수 있습니다.

파티션이 두 개 이상 있는 경우 LVM에서 사용자 지정 파일 시스템 파티션이 있는 이미지를 만들고 런타임에 해당 파티션의 크기를 조정할 수 있습니다. 이를 위해 용량에서 사용자 지정 파일 시스템 구성을 지정한 다음 필요한 디스크 레이아웃으로 이미지를 생성할 수 있습니다. 기본 파일 시스템 레이아웃은 변경되지 않고 파일 시스템 사용자 지정 없이 일반 이미지를 사용하는 경우 **cloud-init**에 의해 루트 파티션의 크기를 조정합니다.

파일 시스템 사용자 지정을 용량에 추가한 후 파일 시스템이 LVM 파티션으로 변환됩니다.

MINIMUM-PARTITION-SIZE를 정의할 때 기본 크기 형식이 없습니다. 다음 값 및 단위가 지원됩니다. kB to TB 및 KiB는 TiB로 지원됩니다. 예를 들어 마운트 지점 크기를 바이트 단위로 정의할 수 있습니다.

```
[[customizations.filesystem]]
mountpoint = "/var"
size = 1073741824
```

단위를 사용하여 마운트 지점 크기를 정의할 수도 있습니다.



참고

RHEL 8.6 및 RHEL 9.0 배포에 제공된 패키지 버전의 단위를 사용하여 마운트 지점 크기를 정의할 수 있습니다.

예를 들어 다음과 같습니다.

```
[[customizations.filesystem]]
mountpoint = "/opt"
size = "20 GiB"
```

추가 리소스

- 파일 시스템 사용자 지정 "size"를 추가한 후 PKCS 가져오기가 실패합니다.

5.4. 추가 리소스

- [RHEL 웹 콘솔의 이미지 빌더를 사용하여 에지 이미지용 RHEL](#) 작성.

6장. 에지 이미지용 RHEL을 프로비저닝하기 위해 간소화된 설치 프로그램 이미지 빌드

장치에 역추적 설치에 최적화된 에지 Simplified 설치 프로그램 이미지용 RHEL을 빌드하고 Edge 이미지용 RHEL에 이미지를 프로비저닝할 수 있습니다.

6.1. 설치 프로그램 이미지 빌드 및 배포 단순화

새로운 이미지 유형 **edge-simplified-installer**를 사용하여 에지 Simplified 설치 프로그램 이미지용 RHEL을 빌드합니다.

에지 간소화 설치 프로그램 이미지용 RHEL을 빌드하려면 기존 **OSTree** 커밋을 제공합니다. 결과 간소화된 이미지에는 OSTree 커밋이 배포된 원시 이미지가 포함되어 있습니다. Simplified 설치 프로그램 ISO 이미지를 부팅한 후 하드 드라이브 또는 가상 머신에서 부팅 이미지로 사용할 수 있는 Edge 시스템의 RHEL을 프로비저닝합니다.

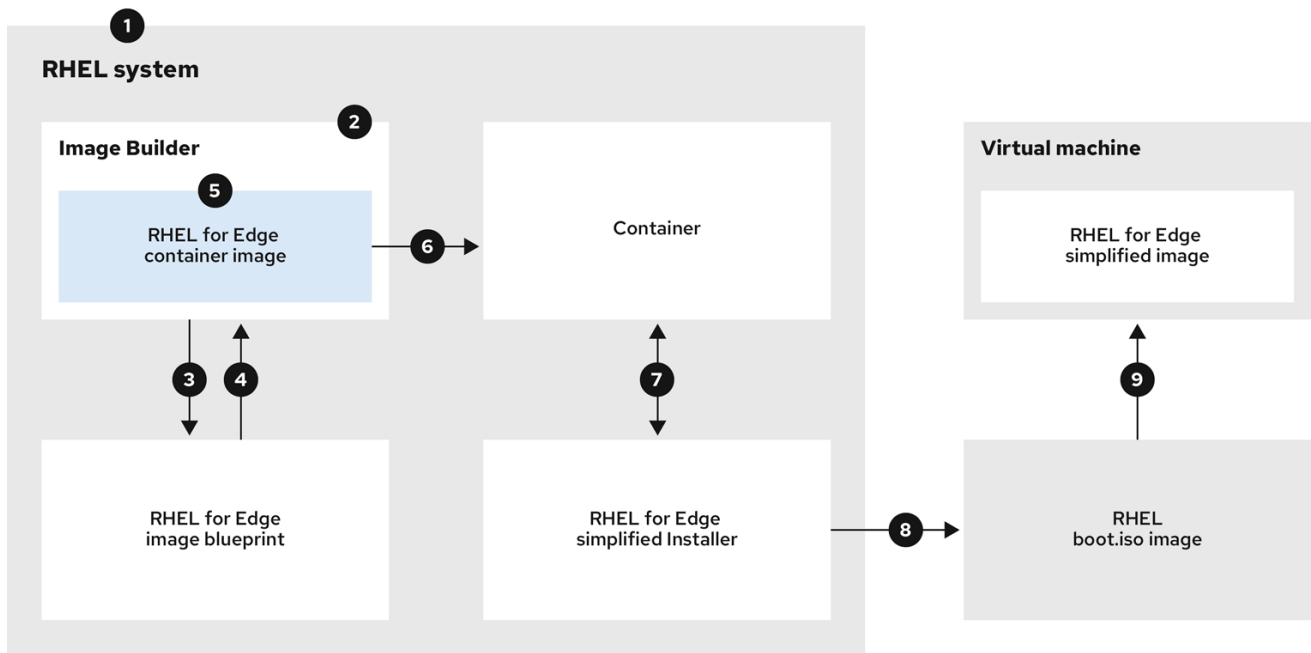
Edge Simplified 설치 프로그램 이미지는 장치에 자동 설치할 수 있도록 최적화되어 네트워크 기반 배포 및 비 네트워크 기반 배포를 모두 지원합니다. 그러나 네트워크 기반 배포의 경우 UEFI HTTP 부팅만 지원합니다.

에지 이미지용 간소화된 RHEL을 작성 및 배포하려면 다음과 같은 고급 단계가 포함됩니다.

1. RHEL 시스템 설치 및 등록
2. 이미지 빌더 설치
3. 이미지 빌더를 사용하여 RHEL for Edge Container 이미지에 대한 사용자 지정으로 블루프린트 (Blueprint) 생성
4. 이미지 빌더에서 RHEL for Edge 블루프린트 가져오기
5. OSTree 리포지토리로 커밋을 배포할 준비가 된 웹 서버와 함께 OCI 컨테이너에 포함된 RHEL for Edge 이미지를 생성합니다.
6. **Edge-simplified-installer**를 위한 tracing 생성
7. Edge 이미지용으로 간소화된 RHEL 빌드
8. Edge 간소화된 이미지를 위한 RHEL 다운로드
9. virt-install을 사용하여 원시 이미지 설치

다음 다이어그램은 Edge Simplified 빌드 및 프로비저닝 워크플로용 RHEL을 나타냅니다.

그림 6.1. 네트워크베이스 환경에서 Edge용 RHEL 빌드 및 프로비저닝



243_RHEL_0422

6.2. 이미지 빌더 CLI를 사용하여 단순화된 이미지에 대한 청사진 생성

에지 이미지용 간소화된 RHEL에 대한 청사진을 생성하려면 다음 단계를 수행합니다.

절차

1. 다음 콘텐츠를 사용하여 Tom의 Obvious, Minimal Language(TOML) 형식으로 일반 텍스트 파일을 생성합니다.

```
name = "simplified-installer-blueprint"
description = "blueprint for the simplified installer image"
version = "0.0.1"
packages = []
modules = []
groups = []
distro = ""

[customizations]
installation_device = "/dev/vda"

[customizations.fdo]
manufacturing_server_url = "http://10.0.0.2:8080"
diun_pub_key_insecure = "true"
```

여기서,

- name 은 name 이고 description 은 distribution에 대한 설명입니다.
- 0.0.1 은 Semantic Versioning scheme에 따른 버전 번호입니다.

- 모듈은 이미지에 설치할 패키지 이름 및 일치하는 버전 글러스트를 설명합니다. 예를 들어 패키지 이름 = "tmux"이고 일치하는 버전은 version = "2.9a"입니다. 현재는 패키지와 모듈간에 차이가 없습니다.
 - 그룹은 이미지에 설치할 패키지 그룹입니다(예: **anaconda-tools** 그룹 패키지). 모듈과 그룹을 모르는 경우 비워 두십시오.
 - 설치 장치는 장치에 자동 설치를 활성화하기 위한 사용자 지정입니다.
 - manufacturing_server_url 은 초기 장치 자격 증명 교환 수행을 수행하는 URL 입니다.
2. 이미지 빌더 서버에 대한 devfile 을 푸시(가져오기) 합니다.

```
# composer-cli blueprints push blueprint-name.toml
```

3. 기존 Makefile 을 나열하여 생성된 inventory 가 성공적으로 푸시되고 있는지 확인합니다.

```
# composer-cli blueprints show blueprint-name
```

4. Makefile 및 해당 종속 항목에 나열된 구성 요소 및 버전이 유효한지 확인합니다.

```
# composer-cli blueprints depsolve blueprint-name
```

추가 리소스

- [이미지 빌더 명령줄을 사용하여 에지 이미지용 RHEL](#) 작성.

6.3. 이미지 빌더 CLI를 사용하여 에지 SIMPLIFIED 설치 프로그램 이미지용 RHEL 생성

이미지 빌더 명령줄 인터페이스를 사용하여 에지 Simplified 이미지에 RHEL 을 생성하려면 다음 사전 요구 사항을 충족한 다음 절차를 따르십시오.

사전 요구 사항

- Edge Simplified 이미지에 대한 RHEL 에 대한 청사진을 생성했습니다.
- 이미지에 포함할 커밋의 OSTree 리포지토리를 제공하셨습니다. 예: <http://10.0.2.2:8080/repo>. [Edge 이미지용 RHEL 을 설치할 웹 서버 설정](#)을 참조하십시오.

절차

1. 부팅 가능한 ISO 이미지를 생성합니다.

```
# composer-cli compose start-ostree \
  blueprint-name \
  edge-simplified-installer \
  --ref rhel/9/x86_64/edge \
  --url URL-OSTree-repository \
```

여기서,

- Quarkus-name 은 에지 opportunity 이름을 위한 RHEL 입니다.

- `edge-simplified-installer` 는 `image-type` 입니다.
- `--ref` 는 커밋을 생성할 위치에 대한 참조입니다.
- `--URL` 은 이미지에 포함할 커밋의 OSTree 리포지토리에 대한 URL 입니다. 예:
<http://10.0.2.2:8080/repository/>. Edge Container에 대한 RHEL 을 시작하거나 웹 서버를 설정할 수 있습니다. 비 네트워크 기반 배포의 경우 [Edge Container 이미지용 RHEL 생성 및 에지 이미지용 RHEL 을 설치할 웹 서버 설정을 참조하십시오.](#)
 쓰기 프로세스가 큐에 추가되었음을 확인합니다. 또한 생성된 이미지에 대한 UUID(Universally Unique Identifier) 번호를 표시합니다. UUID 번호를 사용하여 빌드를 추적합니다. 또한 추가 작업을 위해 UUID 번호를 편리하게 유지합니다.

2. 이미지 compose 상태를 확인합니다.

```
# composer-cli compose status
```

출력에는 다음 형식으로 상태가 표시됩니다.

```
<UUID> RUNNING date blueprint-name blueprint-version image-type
```



참고

이미지 생성 프로세스를 완료하는 데 최대 10분이 걸릴 수 있습니다.

이미지 생성 프로세스를 중단하려면 다음을 실행합니다.

```
# composer-cli compose cancel <UUID>
```

기존 이미지를 삭제하려면 다음을 실행합니다.

```
# composer-cli compose delete <UUID>
```

추가 리소스

- [이미지 빌더 명령줄을 사용하여 에지 이미지용 RHEL 작성.](#)

6.4. 이미지 빌더 명령줄 인터페이스를 사용하여 에지 이미지용 간소화된 RHEL 다운로드

Image Builder 명령줄 인터페이스를 사용하여 에지 이미지의 RHEL 을 다운로드하려면 다음 사전 요구 사항을 충족했는지 확인한 후 절차를 따르십시오.

사전 요구 사항

- 에지 이미지용 RHEL 을 생성했습니다.

절차

1. RHEL for Edge 이미지 상태를 검토합니다.

```
# composer-cli compose status
```

출력에 다음이 표시되어야 합니다.

```
$ <UUID> FINISHED date blueprint-name blueprint-version image-type
```

2. 이미지를 다운로드합니다.

```
# composer-cli compose image <UUID>
```

이미지 빌더는 명령을 실행하는 현재 디렉터리 경로에서 이미지를 **.iso** 파일로 다운로드합니다.

UUID 번호와 이미지 크기가 함께 표시됩니다.

```
$ <UUID>-simplified-installer.iso: size MB
```

그 결과 Edge Simplified Installer ISO 이미지에 대한 RHEL을 다운로드했습니다. 부팅 ISO로 직접 사용하여 Edge 시스템에 RHEL을 설치할 수 있습니다.

6.5. UEFI HTTP BOOT 서버 설정

이 섹션에서는 **UEFI HTTP Boot** 서버에 연결하여 네트워크를 통해 에지 가상 머신의 RHEL을 프로비저닝하기 시작할 수 있도록 UEFI HTTP Boot 서버를 설정하는 방법을 보여줍니다.

사전 요구 사항

- ISO 단순화된 설치 프로그램 이미지를 생성했습니다.
- ISO 콘텐츠를 제공하는 http 서버입니다.

절차

1. 선택한 디렉토리에 ISO 이미지를 마운트합니다.

```
# mkdir /mnt/rhel9-install/
# mount -o loop,ro -t iso9660 /path_directory/installer.iso /mnt/rhel9-install/
```

/path_directory/installer.iso를 에지 부팅 가능 ISO 이미지의 RHEL 경로로 교체합니다.

2. 마운트된 이미지의 파일을 HTTP 서버 루트에 복사합니다. 이 명령은 이미지 콘텐츠를 사용하여 **/var/www/html/rhel9-install/** 디렉토리를 생성합니다.

```
# cp -R /mnt/rhel9-install/* /var/www/html/
# chmod -R +r /var/www/html/httpboot/*
```



참고

일부 복사 방법은 유효한 설치 소스에 필요한 **.treeinfo** 파일을 건너뛸 수 있습니다. 이 절차에 표시된 대로 전체 디렉토리에 대해 **cp** 명령을 실행하면 **.treeinfo**가 올바르게 복사됩니다.

3. 다음을 교체하여 **/var/www/html/EFI/BOOT/grub.cfg** 파일을 업데이트합니다.

a. **coreos.inst.install_dev=/dev/sda** with **coreos.inst.install_dev=/dev/vda**

- b. **linux /images/pxeboot/vmlinuz** with **linuxefi /images/pxeboot/vmlinuz**
- c. **initrd /images/pxeboot/initrd.img** with **initrdefi /images/pxeboot/initrd.img**
- d. **coreos.inst.image_file=/run/media/iso/disk.img.xz** with
coreos.inst.image_url=http://{IP-ADDRESS}/disk.img.xz
 IP-ADDRESS 는 이 시스템의 IP 주소로 http 부팅 서버로 사용됩니다.

4. httpd 서비스를 시작합니다.

```
# systemctl start httpd.service
```

결과적으로 **UEFI HTTP** 부팅을 설정한 후 **UEFI HTTP** 부팅을 사용하여 에지 장치에 대해 RHEL을 설치할 수 있습니다.

6.6. 가상 머신에 단순화된 ISO 이미지 배포

다음 설치 소스를 사용하여 Edge Simplified 이미지용 RHEL을 생성하여 생성한 Edge ISO 이미지에 대한 RHEL을 배포합니다.

- **UEFI HTTP Boot**
- **virt-install**

이 예에서는 네트워크 기반 설치의 ISO 이미지에서 **virt-install** 설치 소스를 생성하는 방법을 보여줍니다.

사전 요구 사항

- ISO 이미지를 생성했습니다.
- UEFI HTTP 부팅을 지원하는 네트워크 구성을 설정했습니다.

절차

1. **UEFI HTTP** 부팅을 지원하기 위해 네트워크 구성을 설정합니다. [libvirt를 사용하여 UEFI HTTP 부팅 설정을 참조하십시오.](#)
2. **virt-install** 명령을 사용하여 UEFI HTTP Boot에서 Edge Virtual Machine용 RHEL을 생성합니다.

```
# virt-install \
  --name edge-install-image \
  --disk path=" ",format=qcow2 \
  --ram 3072 \
  --memory 4096 \
  --vcpus 2 \
  --network network=integration,mac=mac_address \
  --os-type linux \
  --os-variant rhel9 \
  --cdrom "/var/lib/libvirt/images/"ISO_FILENAME" \
  --boot \
  uefi,loader_ro=yes,loader_type=pflash,nvram_template=/usr/share/edk2/ovmf/OVMF_VARS.fd, \
  loader_secure=no \
  --virt-type kvm \
```

```
--graphics none \
--wait=-1
--noreboot
```

명령을 실행하면 가상 머신 설치가 시작됩니다.

검증

- 생성된 가상 머신에 로그인합니다.

6.7. USB 플래시 드라이브에서 SIMPLIFIED ISO 이미지 배포

USB 설치를 사용하여 Edge Simplified 이미지용 RHEL을 생성하여 생성한 에지 ISO 이미지용 RHEL을 배포합니다.

이 예에서는 ISO 이미지에서 **USB 설치** 소스를 생성하는 방법을 보여줍니다.

사전 요구 사항

- ISO 이미지인 간소화된 설치 프로그램 이미지를 생성했습니다.
- 8GB USB 플래시 드라이브가 있습니다.

절차

1. ISO 이미지 파일을 USB 플래시 드라이브에 복사합니다.
2. USB 플래시 드라이브를 부팅하려는 컴퓨터의 포트에 연결합니다.
3. USB 플래시 드라이브에서 ISO 이미지를 부팅합니다. 부팅 메뉴는 다음 옵션을 보여줍니다.

```
Install Red Hat Enterprise Linux 9
Test this media & install Red Hat Enterprise Linux 9
```

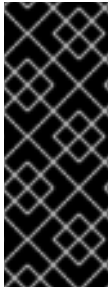
4. Choose Install Red Hat Enterprise Linux 9. 그러면 시스템 설치가 시작됩니다.

추가 리소스

- [설치를 부팅합니다.](#)

7장. FDO를 사용하여 에지 장치용 RHEL 자동 프로비저닝 및 온보딩

Edge Simplified 설치 프로그램 이미지용 RHEL을 빌드하고 Edge 이미지용 RHEL에 프로비저닝할 수 있습니다. FIDO 장치 온보딩(FDO) 프로세스는 자동으로 에지 장치를 프로비저닝하고 온보딩하고 네트워크에 연결된 다른 장치 및 시스템과 데이터를 교환합니다.



중요

Red Hat은 **FDO** 프로세스를 기술 프리뷰 기능으로 제공하며 보안 네트워크에서 실행해야 합니다. 기술 프리뷰 기능은 Red Hat 프로덕션 서비스 수준 계약(SLA)에서 지원되지 않으며 기능적으로 완전하지 않을 수 있습니다. 이러한 기능을 사용하면 향후 제품 기능을 조기에 이용할 수 있어 개발 과정에서 고객이 기능을 테스트하고 피드백을 제공할 수 있습니다. [기술 프리뷰 기능의 지원 범위에](#) 대한 정보는 Red Hat Customer Portal의 기술 프리뷰 기능 지원 범위를 참조하십시오.

7.1. FDO 장치 온보딩 프로세스

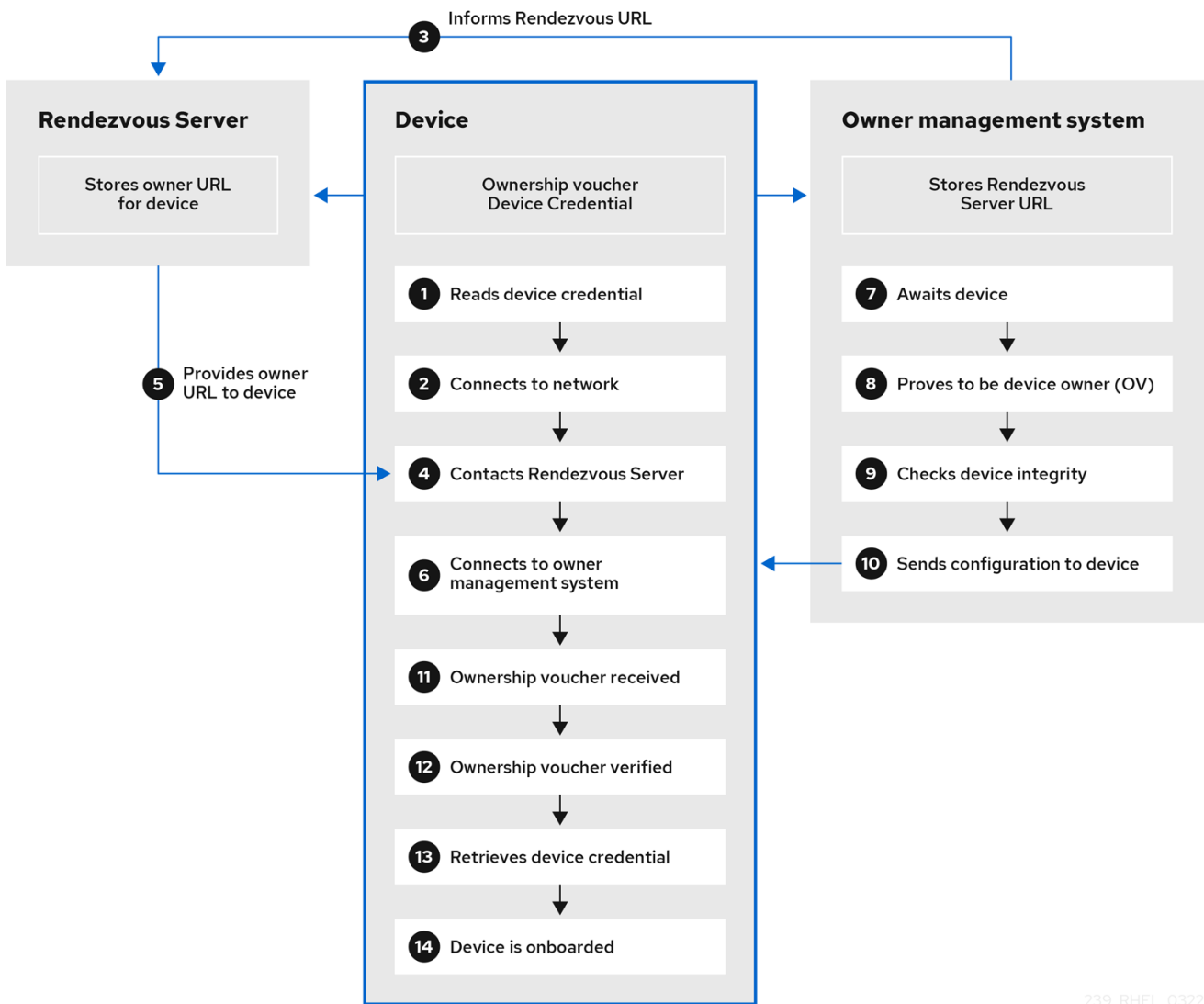
장치 온보딩은 물리적 장치를 프로비저닝 및 온보딩하고 이 장치에 대한 자격 증명을 자동으로 구성하고 네트워크에서 안전하게 연결 및 상호 작용하는 프로세스입니다. FIDO 장치 온보딩(FDO)은 실제 사용을 위해 제조 단계와 낮은 장치 바인딩에서 장치 초기화를 수행하는 프로토콜입니다. 즉, 장치를 관리 시스템에 실제로 바인딩하는 것은 장치에서 수동으로 구성하지 않고도 장치의 첫 번째 부팅 시 수행됩니다.

FDO 프로세스를 사용하면 자동 보안 장치 온보딩(즉, 제로 터치 설치 및 온보딩)을 지원합니다. 장치가 온보딩된 후 패치, 업데이트, 롤백 등을 적용할 수 있습니다. FDO 인증은 새 장치의 설치에 의해 트리거되는 자동 온보딩 프로세스입니다.

Edge Simplified Installer 이미지용 RHEL을 빌드하고 자동으로 커지려면 기존 OSTree 커밋을 제공합니다. 결과 간소화된 이미지에는 OSTree 커밋이 배포된 원시 이미지가 포함되어 있습니다. Simplified 설치 프로그램 ISO 이미지를 부팅한 후 하드 드라이브 또는 가상 머신에서 부팅 이미지로 사용할 수 있는 Edge 시스템의 RHEL을 프로비저닝합니다. Edge Simplified 설치 프로그램 이미지는 장치에 자동 설치할 수 있도록 최적화되어 네트워크 기반 배포 및 비 네트워크 기반 배포를 모두 지원합니다. 그러나 네트워크 기반 배포의 경우 UEFI HTTP 부팅만 지원합니다.

다음 다이어그램은 FIDO 장치 온보딩 워크플로를 나타냅니다.

그림 7.1. 네트워크 이외의 환경에서 RHEL for Edge 배포



239_RHEL_0322

1. 장치가 장치 인증 정보 읽기
2. 네트워크에 장치 연결
3. 초기 단계에서 소유자 관리 시스템은 제조업체가 소유자 관리 시스템의 위치에 대해 rendezvous 서버에 알립니다.
4. 네트워크에 연결한 후 장치는 Rendezvous 서버에 연결됩니다.
5. Rendezvous 서버는 장치 소유자 URL 을 보냅니다.
6. 장치는 소유자 관리 시스템에 연결하고, 장치 키로 문을 서명하여 올바른 장치임을 증명합니다.
7. 소유자 관리 시스템은 소유자 바의 마지막 키로 문을 서명하여 증명할 수 있습니다.
8. 소유자 관리 시스템은 장치가 ssh 키에서 저장하는 장치에 대한 구성을 제공합니다.
9. 장치는 소유권금을 수신하고 확인 할 수 있습니다.
10. 그런 다음 장치가 장치 인증 정보를 검색합니다.
11. 그 후, 소유자 관리 시스템은 장치를 온보딩(onboarded)으로 보고

전체 FDO 프로세스는 완료되었으며 이 장치에서 더 이상 사용되지 않습니다.

7.2. 에지 장치용 RHEL 자동 프로비저닝 및 온보딩

에지 장치에 대한 RHEL의 자동 프로비저닝 및 온보딩에는 다음과 같은 고급 단계가 포함됩니다.

1. RHEL 시스템 설치 및 등록
2. 이미지 빌더 설치
3. 이미지 빌더를 사용하여 RHEL for Edge Container 이미지에 대한 사용자 지정으로 블루프린트 (Blueprint) 생성
4. 이미지 빌더에서 RHEL for Edge 블루프린트 가져오기
5. OSTree 리포지토리로 커밋을 배포할 준비가 된 웹 서버와 함께 OCI 컨테이너에 포함된 RHEL for Edge 이미지를 생성합니다.
6. 스토리지 장치 경로 및 FDO 사용자 정의를 사용하여 edge-simplified-installer에 대한 청사진을 생성
7. 에지 이미지용 간소화된 설치 프로그램 RHEL 빌드
8. Edge용 RHEL 단순화 설치 프로그램 이미지 다운로드
9. virt-install을 사용하여 간소화된 설치 프로그램 ISO 이미지를 장치 또는 VM에 설치합니다. FIDO FDO 클라이언트는 Simplified Installer ISO에서 실행되며 UEFI 디렉터리 구조는 이미지를 부팅 가능하게 만듭니다.
10. 네트워크 구성을 사용하면 장치는 제조 서버에 연결하여 초기 장치 자격 증명 교환을 수행할 수 있습니다.
11. 시스템이 끝점에 도달하면 장치에 대한 장치 자격 증명이 생성됩니다.
12. 온보드 서버는 장치 인증 정보를 사용하여 온보딩 서버에 대해 인증합니다. 온보딩 서버는 구성을 장치/시스템에 전달합니다. 시스템에 연결한 후 온보드 서버에 연결하여 구성을 수신합니다.
13. 온보딩 서버는 장치에 SSH 키를 제공하고 시스템을 설치합니다.
14. 그런 다음 시스템을 재부팅하고 TPM에 저장된 강력한 키를 사용하여 암호화합니다.
15. 생성된 청사진에서 인증 정보를 사용하여 시스템에 로그인하고 Simplified Installer ISO 이미지에 생성된 구성을 확인할 수 있습니다.

추가 리소스

- [FDO 자동 온보딩 기술](#)

7.3. 키 및 인증서 생성

FDO 인프라를 실행하려면 키와 인증서를 생성해야 합니다. 이러한 키와 인증서는 제조 서버를 구성하는 데 사용됩니다.

중요

Red Hat은 **fdo-admin-tool generate-key-and-cert** 툴을 기술 프리뷰 기능으로 제공하며 보안 네트워크에서 실행해야 합니다. 기술 프리뷰 기능은 Red Hat 프로덕션 서비스 수준 계약(SLA)에서 지원되지 않으며 기능적으로 완전하지 않을 수 있습니다. 이러한 기능을 사용하면 향후 제품 기능을 조기에 이용할 수 있어 개발 과정에서 고객이 기능을 테스트하고 피드백을 제공할 수 있습니다. [기술 프리뷰 기능의 지원 범위에](#) 대한 정보는 Red Hat Customer Portal의 기술 프리뷰 기능 지원 범위를 참조하십시오.

절차

1. 키와 인증서로 폴더를 생성합니다. 이를 위해 다음 명령을 실행합니다.

```
# fdo-admin-tool generate-key-and-cert
```

2. 생성된 키와 인증서를 확인합니다.

```
$ tree keys
```

다음 출력을 볼 수 있습니다.

```
-device_ca_cert.pem
-device_ca_key.der
-diun_cert.pem
-diun_key.dre
-manufacturer_cert.pem
-manufacturer_key.der
-owner_cert.pem
-owner_key.pem
```

추가 리소스

- **fdo-admin-tool generate-key-and-cert -help**

7.4. 제조 서버 패키지 설치

manufacturing server RPM 패키지는 장치 온보드에 안전하게 인증 정보를 제공합니다. 장치 설치 중에 제조 서버는 Rendezvous 서버를 요청하여 서버에 대한 장치 자격 증명 인증을 제공하고 장치 자격 증명을 설치된 시스템에 설치합니다.

중요

Red Hat은 **F do-manufacturing-server** 툴을 기술 프리뷰 기능으로 제공하며 보안 네트워크에서 실행해야 합니다. 기술 프리뷰 기능은 Red Hat 프로덕션 서비스 수준 계약(SLA)에서 지원되지 않으며 기능적으로 완전하지 않을 수 있습니다. 이러한 기능을 사용하면 향후 제품 기능을 조기에 이용할 수 있어 개발 과정에서 고객이 기능을 테스트하고 피드백을 제공할 수 있습니다. [기술 프리뷰 기능의 지원 범위에](#) 대한 정보는 Red Hat Customer Portal의 기술 프리뷰 기능 지원 범위를 참조하십시오.

manufacturing server RPM 패키지를 설치하려면 다음 단계를 완료합니다.

절차

1. 다음 명령을 실행합니다.

```
# dnf install fdo-manufacturing-server --refresh
```

2. 설치된 파일을 확인합니다.

```
$ ls /usr/share/doc/fdo
```

다음 출력을 볼 수 있습니다.

```
Output:
manufacturing server.yml
Owner-onboarding-server.yml
rendezvous-server.yml
```

3. 선택 사항: 각 파일의 내용을 확인합니다. 예를 들면 다음과 같습니다.

```
$ cat /usr/share/doc/fdo/manufacturing-server.yml
```

4. manufacturing 서버를 구성합니다. 다음을 제공해야 합니다.

- 제조 서버 URL
- rendezvous 서버의 IP 주소 또는 DNS 이름
- 생성된 키와 인증서의 경로입니다. 키 및 인증서 생성 섹션을 참조하십시오.

5. RHEL for Edge 네트워크 단순화 이미지를 장치에 설치한 후 제조업체 서버가 Podman 컨테이너에서 실행되고 있는지 확인합니다. 제조 서버는 새 장치에서 장치 자격 증명 생성 및 활성화를 처리합니다.

```
$ cat /usr/share/doc/fdo/manufacturing-server.yml
```

추가 리소스

- [manufacturing-server.yml 예제](#)
- [FDO 자동 온보딩 용어](#)

7.5. FDO 인증을 사용하여 에지 장치용 RHEL 자동 온보딩

Edge 장치용 RHEL 을 자동으로 켜도록 장치를 준비하려면 다음 단계를 완료합니다.

사전 요구 사항

- ostree 컨테이너를 빌드하고 제공하셨습니다.
- 장치가 조립 및 프로비저닝되었습니다. 이 예에서는 VM 시스템을 사용하지만 실제 장치에서 사용할 수 있습니다.
- UEFI HTTP 부팅 서버를 실행하고 있습니다.
- **fdo-manufacturing-server** rpm 패키지를 설치했습니다. 다음을 실행합니다.

■

```
# dnf install fdo-manufacturing-server
```

절차

1. ISO 간소화 이미지를 사용하여 설치를 실행합니다. 예를 들어 CD-ROM 또는 USB 플래시 드라이브에서 설치할 수 있습니다.
설치에서는 ISO Simplified 설치 프로그램 이미지를 실행합니다. 여기서 FDO 클라이언트가 실행되고 UEFI 디렉터리 구조는 이미지를 부팅하여 ISO에서 원시 이미지를 구울 수 있습니다.
2. VM 터미널을 통해 장치가 초기 장치 인증 정보 교환을 수행하고 소유권 바우더를 생성하기 위해 제조 서비스에 도달했는지 확인합니다.

```
$ ls directory-path/ownership_voucher/
```

출력에 올바른 장치 자격 증명이 장치에 추가되었음을 나타내기 위해 **ownership_voucher** ID가 표시되어야 합니다.

온보딩 서버는 장치 자격 증명을 사용하여 온보딩 서버에 대해 인증합니다. 그런 다음 구성을 장치에 전달합니다. 장치가 온보드 서버에서 구성을 수신한 후 SSH 키를 수신하고 장치에 운영 체제를 설치합니다. 마지막으로 시스템은 자동으로 재부팅되며 TPM에 저장된 강력한 키를 사용하여 암호화합니다.

장치가 자동으로 재부팅되면 장치는 온보딩 서버에 연결하고 FDO에서 사용자 인증 정보를 자동으로 프로비저닝합니다.

검증

장치가 자동으로 재부팅된 후 드라이브용으로 생성한 자격 증명을 사용하여 장치 또는 VM에 로그인할 수 있습니다.

1. 청사진에 대해 생성한 사용자 이름과 암호를 제공하여 장치에 로그인합니다.
2. 선택 사항: 원시 이미지에 생성된 구성을 확인합니다.

추가 리소스

- [UEFI HTTP Boot 서버 설정](#)
- [USB 플래시 드라이브에서 Simplified ISO 이미지 배포](#)

8장. 네트워크 기반 환경에서 에지 이미지에 RHEL 배포

RHEL 설치 프로그램 그래픽 사용자 인터페이스 또는 Kickstart 파일을 사용하여 에지용 RHEL을 배포할 수 있습니다. Edge 이미지에 RHEL을 배포하는 전체 프로세스는 배포 환경이 네트워크 기반인지 또는 비네트워크 기반인지에 따라 다릅니다.



참고

베어 메탈에 이미지를 배포하려면 Kickstart 파일을 사용합니다.

네트워크 기반 배포

네트워크 기반 환경에서 에지 이미지에 RHEL을 배포하려면 다음과 같은 상위 수준 단계를 수행해야 합니다.

- a. 이미지 파일 콘텐츠를 추출합니다.
- b. 웹 서버 설정
- c. 이미지 설치

8.1. 에지 이미지 커밋을 위해 RHEL 추출

커밋을 다운로드한 후 .tar 파일을 추출하고 ref 이름과 커밋 ID를 기록해 둡니다.

다운로드한 커밋 파일은 OSTree 리포지토리가 있는 .tar 파일로 구성됩니다. OSTree 리포지토리에는 커밋과 **compose.json** 파일이 포함되어 있습니다.

compose.json 파일에는 "Ref"(참조 ID) 및 커밋 ID와 같은 정보가 포함된 커밋 메타데이터가 포함되어 있습니다. 커밋에는 rpm 패키지가 포함되어 있습니다.

패키지 콘텐츠를 추출하려면 다음 단계를 수행합니다.

사전 요구 사항

- Kickstart 파일을 생성하거나 기존 파일을 사용합니다.

절차

1. 다운로드한 이미지 tar 파일을 추출합니다.

```
# tar xvf <UUID>-commit.tar
```

2. .tar 파일을 추출한 디렉터리로 이동합니다.

compose.json 파일과 OSTree 디렉터리가 포함되어 있습니다. **compose.json** 파일에는 커밋 번호가 포함되어 있고 OSTree 디렉터리에는 rpm 패키지가 포함되어 있습니다.

3. **compose.json** 파일을 열고 커밋 ID 번호를 기록해 둡니다. 웹 서버를 설정하려면 이 숫자가 필요합니다.

jq JSON 프로세서가 설치된 경우 **jq** 툴을 사용하여 커밋 해시 ID를 검색할 수도 있습니다.

```
# jq '["ostree-commit"]' < compose.json
```

4. 커밋에 **rpm** 패키지를 나열합니다.

```
# rpm-ostree db list rhel/9/x86_64/edge --repo=repo
```

5. Kickstart 파일을 사용하여 RHEL 설치 프로그램을 실행합니다. 기존 파일을 사용하거나 Kickstart 생성기 도구를 사용하여 만들 수 있습니다. Kickstart 파일에서 파일 시스템을 프로비저닝하고, 사용자를 만들고, 에지 이미지용 RHEL 을 가져 오고 배포하는 방법에 대한 세부 정보를 포함해야 합니다. RHEL 설치 프로그램은 설치 프로세스 중에 이 정보를 사용합니다.

다음은 Kickstart 파일 예제입니다.

```
lang en_US.UTF-8
keyboard us
timezone Etc/UTC --isUtc
text
zerombr
clearpart --all --initlabel
autopart
reboot
user --name=core --group=wheel
sshkey --username=core "ssh-rsa AAAA3Nza...."

ostreesetup --nogpg --osname=rhel --remote=edge --url=https://mirror.example.com/repo/ --
ref=rhel/9/x86_64/edge
```

ostreesetup 명령은 OSTree 기반 설치를 설정하는 데 사용됩니다. 다음 플래그를 사용하여 OSTree 커밋을 가져옵니다.

- **--nogpg** - GPG 키 확인을 비활성화합니다.
- **--osname** - OS 설치를 위한 관리 루트입니다.
- **--remote** - OS 설치를 위한 관리 루트
- **--URL** - 설치할 리포지토리의 URL 입니다.
- **--ref** - 설치에 사용할 저장소의 분기 이름입니다.
- **--URL=https://mirror.example.com/repo/** 은 에지 커밋을 추출하고 nginx를 통해 제공하는 호스트 시스템의 주소입니다. 주소를 사용하여 게스트 시스템에서 호스트 시스템에 연결할 수 있습니다.
예를 들어 **/var/www/html** 디렉터리에서 커밋 이미지를 추출하고 호스트 이름이 **www.example.com** 인 시스템에서 nginx를 통해 커밋을 제공하는 경우 **--url** 매개 변수의 값은 **http://www.example.com/repo** 입니다.



참고

HTTP 프로토콜을 사용하여 Apache HTTP 서버에서 https가 활성화되어 있지 않기 때문에 커밋을 제공할 서비스를 시작합니다.

추가 리소스

- [Edge 이미지용 RHEL 다운로드](#)
- [Kickstart 파일 생성](#)

8.2. 에지 이미지용 **RHEL**을 설치하기 위해 웹 서버 설정

에지 이미지 콘텐츠를 위해 RHEL을 추출한 후 HTTP를 통해 RHEL 설치 프로그램에 이미지 커밋 세부 정보를 제공하도록 웹 서버를 설정합니다.

다음 예제에서는 컨테이너를 사용하여 웹 서버를 설정하는 단계를 보여줍니다.

사전 요구 사항

- 시스템에 Podman이 설치되어 있어야 합니다. [RHEL에 podman을 설치하는 방법](#)

절차

- 다음 지침을 사용하여 **nginx** 구성 파일을 생성합니다.

```
events {
}

http {
    server{
        listen 8080;
        root /usr/share/nginx/html;
    }
}

pid /run/nginx.pid;
daemon off;
```

- nginx** 구성 파일을 **/etc/nginx.conf** 디렉터리에 복사합니다.
- 다음 명령을 사용하여 **dockerfile**을 생성합니다.

```
FROM registry.access.redhat.com/ubi8/ubi
RUN dnf -y install nginx && dnf clean all
COPY kickstart.ks /usr/share/nginx/html/
COPY repo /usr/share/nginx/html/
EXPOSE 8080
CMD ["/usr/sbin/nginx", "-c", "/etc/nginx.conf"]
ARG commit
ADD ${commit} /usr/share/nginx/html/
```

여기서,

- Kickstart.ks**는 RHEL for Edge 이미지의 Kickstart 파일 이름입니다. Kickstart 파일에는 지시문 정보가 포함되어 있습니다. 나중에 이미지를 관리하려면 **greenboot** 검사를 위한 검사 및 설정을 포함하는 것이 좋습니다. 이를 위해 다음 설정을 포함하도록 Kickstart 파일을 업데이트할 수 있습니다.

```
lang en_US.UTF-8
keyboard us
timezone Etc/UTC --isUtc
text
zerombr
clearpart --all --initlabel
```

```

autopart
reboot
user --name=core --group=wheel
sshkey --username=core "ssh-rsa AAAA3Nza...."

ostreesetup --nogpg --osname=rhel --remote=edge
--url=https://mirror.example.com/repo/
--ref=rhel/9/x86_64/edge

%post
cat << EOF > /etc/greenboot/check/required.d/check-dns.sh
#!/bin/bash

DNS_SERVER=$(grep nameserver /etc/resolv.conf | cut -f2 -d" ")
COUNT=0

# check DNS server is available
ping -c1 $DNS_SERVER
while [ $? != '0' ] && [ $COUNT -lt 10 ]; do

    COUNT++
    echo "Checking for DNS: Attempt $COUNT ."
    sleep 10
    ping -c 1 $DNS_SERVER
done
EOF
%end

```

OSTree 리포지토리는 모든 HTTP 서비스에서 호스팅할 수 있으며, 예를 들어 컨테이너를 사용하는 것은 이 작업을 수행하는 방법에 대한 옵션일 뿐입니다. docker 파일은 다음 작업을 수행합니다.

- a. 최신 Universal Base Image (ubi) 사용
- b. 웹 서버 설치(nginx)
- c. 서버에 Kickstart 파일 추가
- d. 서버에 에지 이미지 커밋을 위한 RHEL 추가

4. Docker 컨테이너 빌드

```
# podman build -t name-of-container-image --build-arg commit=uuid-commit.tar .
```

5. 컨테이너 실행

```
# podman run --rm -d -p port:8080 localhost/name-of-container-image
```

그 결과 서버는 **commit.tar** 리포지토리 및 Kickstart 파일을 사용하여 RHEL 설치 프로그램을 시작할 준비가 되었습니다.

8.3. RHEL BOOT.ISO 이미지 다운로드

Red Hat Customer Portal에서 Red Hat Boot ISO 이미지를 다운로드할 수 있습니다. Red Hat Boot ISO 이미지는 RHEL 설치 프로그램을 시작하는 데 사용됩니다. 설치 프로그램은 에지 이미지용 RHEL 설치를 위해 제공하는 Kickstart 파일을 가져옵니다.

사전 요구 사항

- 유효한 Red Hat 서브스크립션이 있습니다.
- <https://access.redhat.com/downloads> 에서 Red Hat 고객 포털의 제품 다운로드 섹션에 로그인되어 있습니다.

절차

1. 브라우저를 열고 <https://access.redhat.com/downloads> 에 액세스합니다.
2. Infrastructure Management에서 **Red Hat Enterprise Linux 9** 제품을 클릭합니다.
3. "Red Hat Enterprise Linux 9 Boot ISO" 옵션에 대해 **지금 다운로드** 버튼을 클릭합니다.

추가 리소스

- [RHEL 설치 ISO 이미지 다운로드](#)

8.4. KICKSTART 파일을 사용하여 에지 이미지용 RHEL 설치

Kickstart 파일을 사용하여 에지 이미지의 RHEL을 설치하려면 웹 서버를 사용합니다. 웹 서버는 RHEL for Edge image **commit.tar** 리포지토리와 Kickstart 파일을 사용하여 RHEL 설치 프로그램을 시작합니다.

사전 요구 사항

- 설치 프로그램에서 커밋을 가져올 서버를 사용할 수 있으며 실행 가능합니다.
- 생성한 커밋을 설치할 **.qcow2** 디스크 이미지입니다. [CLI에서 이미지 빌더를 사용하여 시스템 이미지 생성을 참조하십시오.](#)

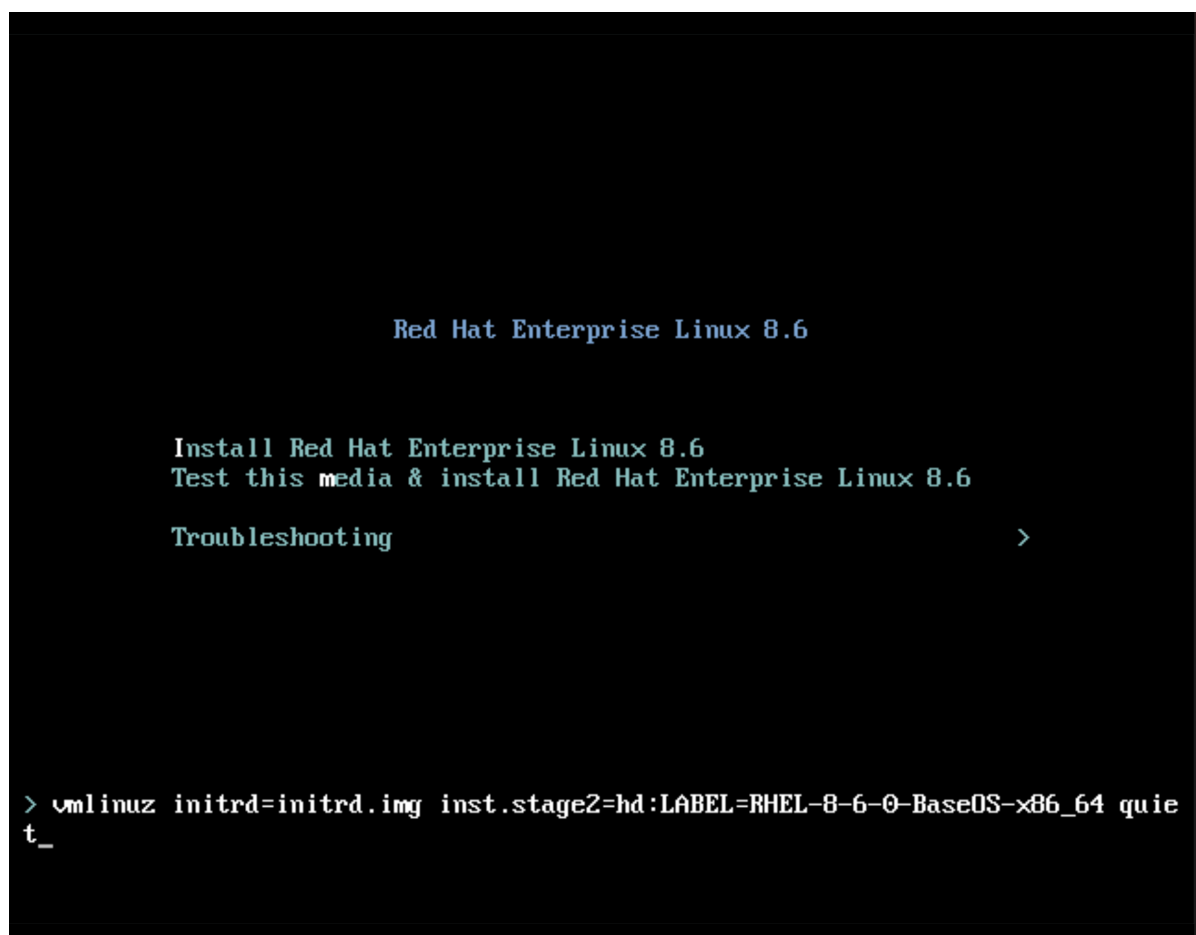
절차

1. **libvirt virt-install** 을 사용하여 Anaconda 설치 프로그램을 실행합니다.

```
virt-install \
--name rhel-edge-test-1
--memory 2048 \
--vcpus 2 \
--disk size=8 \
--disk path=prepared_disk_image.qcow2,format=qcow2 \
--os-variant rhel9
--cdrom /home/username/Downloads/rhel-9-x86_64-boot.iso
```

2. 설치 화면에서 다음을 수행합니다.

그림 8.1. Red Hat Enterprise Linux 부팅 메뉴



- a. **e** 키를 눌러 추가 커널 매개변수를 추가합니다.

```
inst.ks=http://edge_device_ip:port/kickstart.ks
```

kernel 매개변수는 RHEL 설치 프로그램에 포함된 RHEL 이미지가 아닌 Kickstart 파일을 사용하여 RHEL을 설치하도록 지정합니다.

- b. 커널 매개 변수를 추가한 후 **Ctrl+X**를 눌러 Kickstart 파일을 사용하여 RHEL 설치를 부팅합니다.

RHEL Installer가 시작되고 서버(HTTP) 끝점에서 Kickstart 파일을 가져오고 HTTP 끝점에서 에지 이미지 커밋용 RHEL을 설치하는 명령을 포함하여 명령을 실행합니다. 설치가 완료되면 설치 프로그램에서 로그인 세부 정보를 입력하라는 메시지를 표시합니다.

검증

- 로그인 화면에서 사용자 계정 자격 증명을 입력하고 **Enter**를 클릭합니다.
- 에지 이미지의 RHEL이 성공적으로 설치되었는지 확인합니다.

```
$ rpm-ostree status
```

명령 출력은 이미지 커밋 ID를 제공하며 설치에 성공했는지 표시됩니다.

다음은 샘플 출력입니다.

```
State: idle
Deployments:
```

```
*ostree://edge:rhel/9/x86_64/edge  
Timestamp: 2020-09-18T20:06:54Z  
Commit: 836e637095554e0b634a0a48ea05c75280519dd6576a392635e6fa7d4d5e96
```

추가 리소스

- [설치를 부팅합니다.](#)

9장. 네트워크 기반이 아닌 환경에서 **EDGE** 이미지용 **RHEL** 배포

비 네트워크 기반 환경에 엣지 이미지용 RHEL을 배포하려면 다음과 같은 상위 수준 단계가 포함됩니다.

1. Edge 컨테이너용 RHEL을 다운로드합니다. Edge 이미지용 RHEL을 다운로드하는 방법에 대한 자세한 내용은 [Edge 이미지용 RHEL 다운로드](#)에서 참조하십시오.
2. Edge 컨테이너 이미지의 RHEL을 Podman에 로드
3. Podman에서 Edge Container 이미지에 대한 RHEL을 실행합니다.
4. 설치 프로그램 화면 로드
5. Edge 설치 프로그램 이미지용 RHEL 빌드
6. qcow 디스크 준비
7. VM(가상 머신) 부팅
8. 이미지 설치

9.1. 네트워크 기반이 아닌 배포를 위한 **EDGE** 컨테이너 이미지용 **RHEL** 생성

Edge 컨테이너 OSTree 커밋을 위해 다운로드한 RHEL을 Podman에 로드하여 실행 중인 컨테이너를 빌드할 수 있습니다. 이를 위해 다음 단계를 수행합니다.

사전 요구 사항

- Edge 컨테이너 OSTree 커밋을 위한 RHEL을 생성하고 다운로드했습니다.
- 시스템에 **Podman**이 설치되어 있어야 합니다. [RHEL에 podman을 설치하는 방법](#)을 참조하십시오.

절차

1. 엣지 컨테이너 OSTree 커밋을 위해 RHEL을 다운로드한 디렉터리로 이동합니다.
2. 엣지 컨테이너 OSTree 커밋을 위한 RHEL을 **Podman**에 로드합니다.

```
$ sudo podman load -i UUID-container.tar
```

명령 출력은 이미지 ID를 제공합니다(예: @8e0d51f061ffa51d1578042bc2bc875b649b27f2ae66566a15e6530cec63).

3. 이전 단계에서 생성한 이미지 ID를 사용하여 엣지 컨테이너 이미지의 새 RHEL에 태그를 지정합니다.

```
$ sudo podman tag image-ID localhost/edge-container
```

podman tag 명령은 로컬 이미지에 추가 이름을 할당합니다.

4. **edge-container** 라는 컨테이너를 실행합니다.

```
$ sudo podman run -d --name=edge-container -p 8080:8080 localhost/edge-container
```

podman run -d --name=edge-container 명령은 **localhost/edge-container** 이미지를 기반으로 컨테이너에 이름을 할당합니다.

5. 컨테이너를 나열합니다.

```
$ sudo podman ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
2988198c4c4b	/localhost/edge-container	/bin/bash	3 seconds ago	Up 2 seconds ago
edge-container				

결과적으로 **Podman**은 에지 컨테이너 커밋을 위해 RHEL을 사용하여 OSTree 리포지토리를 제공하는 컨테이너를 실행합니다.

9.2. 네트워크 기반이 아닌 배포에 대한 엣지 설치 프로그램 이미지용 RHEL 생성

에지 컨테이너 커밋을 위해 RHEL을 사용하여 리포지토리를 제공하도록 실행 중인 컨테이너를 빌드한 후 **에지 설치 프로그램(.iso) 이미지를 위한 RHEL**을 생성합니다. 설치 프로그램(.iso)은 에지 컨테이너(.tar)용 RHEL에서 제공하는 커밋을 가져옵니다. Edge 컨테이너용 RHEL이 Podman에 로드되면 OSTree를 URL 형식으로 노출합니다.

웹 콘솔에서 에지 설치 프로그램 이미지에 RHEL을 생성하려면 다음 단계를 따르십시오.

사전 요구 사항

- 에지 이미지용 RHEL에 대한 청사진을 생성했습니다. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 **에지 설치 프로그램 이미지용 RHEL에 대한 청사진 생성**을 참조하십시오.
- RHEL 시스템에서 이미지 빌더 대시보드에 액세스했습니다.

절차

- Image Builder 대시보드에서 Edge 이미지용 RHEL을 위해 생성한 RHEL for Edge Installerakiate의 경우 **Create Image**를 클릭합니다.
- 이미지 생성** 창에서 다음 단계를 수행합니다.
 - Repository** textbox에서 이미지에 포함할 Edge 컨테이너 OSTree URL에 대한 RHEL을 지정합니다. 예: <http://localhost:8080/repo>
 - Ref** textbox에서 이미지에 삽입할 에지 컨테이너 커밋을 위한 RHEL 생성 중에 제공한 것과 동일한 참조를 지정합니다. 예를 들면 **rhel/edge/test**입니다.
 - 생성**을 클릭합니다.
이미지 빌더는 이미지 빌드 중에 실행 중인 컨테이너에서 제공하는 커밋을 가져옵니다.
이미지 빌드가 완료되면 **결과(.iso)** 이미지를 다운로드할 수 있습니다.
- 이미지를 다운로드합니다. [Edge 이미지용 RHEL 다운로드](#)를 참조하십시오.

추가 리소스

- [네트워크 이외의 배포의 명령줄 인터페이스를 사용하여 엣지 설치 프로그램 이미지용 RHEL 생성](#)

9.3. 네트워크 기반이 아닌 배포를 위한 **EDGE** 이미지용 **RHEL** 설치

Edge 이미지에 대한 RHEL 을 설치하려면 다음 단계를 따르십시오.

사전 요구 사항

- Edge Installer ISO 이미지를 위한 RHEL 을 생성했습니다.
- 실행 중인 컨테이너를 중지했습니다.
- 생성한 커밋을 설치할 디스크 이미지입니다.
- **edk2-ovmf** 패키지가 설치되어 있습니다.
- **virt-viewer** 패키지가 설치되어 있어야 합니다.
- 사용자 계정을 사용하여 롤아웃을 사용자 지정할 수 있습니다. [RHEL for Edge 이미지 블루프린트의 관리자 계정 생성](#)을 참조하십시오.



주의

용량에서 사용자 계정 사용자 지정을 정의하지 않으면 ISO 이미지에 로그인할 수 없습니다.

절차

1. (.iso) 이미지를 설치할 **qcow** VM 디스크 파일을 생성합니다. 이는 VM(가상 머신)의 하드 드라이브 이미지입니다. 예를 들어 다음과 같습니다.

```
$ qemu-img create -f qcow2 diskfile.qcow2 20G
```

2. **virt-install** 명령을 사용하여 디스크로 VM을 부팅하고 설치 프로그램 ISO를 CD-ROM으로 부팅합니다. 예를 들어 다음과 같습니다.

```
$ virt-install \
--boot uefi \
--name VM_NAME \
--memory 2048 \
--vcpus 2 \
--disk path=diskfile.qcow2 \
--cdrom /var/lib/libvirt/images/UUID-installer.iso \
--os-variant rhel9.0
```

이 명령은 **virt-install** 에 다음을 수행합니다.

- BIOS 대신 UEFI를 사용하여 부팅하도록 VM에 지시합니다.
- 설치 ISO를 마운트합니다.
- 첫 번째 단계에서 만든 하드 드라이브 이미지를 사용합니다.

Anaconda 설치 프로그램을 제공합니다. RHEL 설치 프로그램이 시작되고 ISO에서 Kickstart 파일을 로드하고 명령을 실행합니다(예: Edge 이미지 커밋용 RHEL 설치). 설치가 완료되면 설치 프로그램에서 로그인 세부 정보를 묻는 메시지를 표시합니다.



참고

Anaconda는 설치 중에 컨테이너 커밋을 사용하도록 사전 구성되어 있습니다. 그러나 디스크 파티션, 시간대 등의 시스템 구성을 설정해야 합니다.

3. **virt-viewer** 를 사용하여 Anaconda GUI에 연결하여 시스템 구성을 설정합니다.

```
$ virt-viewer --connect qemu:///system --wait VM_NAME
```

4. 시스템을 재부팅하여 설치를 완료합니다.
5. 로그인 화면에서 사용자 계정 자격 증명을 지정하고 **Enter** 를 클릭합니다.

검증 단계

1. 에지 이미지의 RHEL 이 성공적으로 설치되었는지 확인합니다.

```
$ rpm-ostree status
```

명령 출력은 이미지 커밋 ID를 제공하며 설치에 성공했는지 표시됩니다.

10장. 에지 이미지 RHEL 관리

에지 이미지의 RHEL을 관리하려면 다음 관리 작업을 수행할 수 있습니다.

- RHEL 웹 콘솔에서 Image Builder를 사용하여 에지 이미지 options를 위한 RHEL 편집
- Image Builder 명령줄을 사용하여 에지 이미지 options를 위한 RHEL 편집
- 에지 이미지의 RHEL 업데이트
- 노드에서 rpm-ostree 원격을 구성하여 노드 정책을 업데이트합니다.
- greenboot를 사용하여 에지 이미지에 대한 RHEL을 수동으로 복원하거나 자동으로 복원

10.1. RHEL 웹 콘솔에서 이미지 빌더를 사용하여 에지 이미지 승격을 위한 RHEL 편집

에지 이미지 options에 대한 RHEL을 다음과 같이 편집할 수 있습니다.

- 필요할 수 있는 구성 요소 추가
- 기존 구성 요소의 버전 수정
- 기존 구성 요소 제거

10.1.1. RHEL 웹 콘솔에서 Image Builder를 사용하여 에지 이미지 options를 위한 RHEL에 구성 요소 추가

RHEL for Edge 이미지 options에 구성 요소를 추가하려면 다음 사전 요구 사항을 충족했는지 확인한 후 절차에 따라 해당 devfile을 편집합니다.

사전 요구 사항

- RHEL 시스템에서 이미지 빌더 대시보드에 액세스했습니다.
- RHEL for Edge 이미지용 블루프린트를 생성했습니다.

절차

1. 이미지 빌더 대시보드에서 편집하려는 에지 이미지 options에 대한 RHEL을 클릭합니다.
특정 요건을 검색하려면 Filter By Name(이름으로 필터) 텍스트 상자에 Quarkus 이름을 입력한 다음 **Enter**를 누릅니다.
2. greater의 오른쪽 상단에서 **패키지 편집**을 클릭합니다.
패키지 편집 모드에 대한 보기 변경 사항.
3. Filter By Name(이름으로 필터) 텍스트 상자에 추가할 구성 요소 이름을 입력한 다음 Enter를 누릅니다.
구성 요소 이름이 있는 목록이 표시됩니다.
4. 구성 요소에 인접한 **+** 기호를 클릭합니다.
구성 요소가 Quarkus에 추가됩니다.
5. 커밋을 클릭합니다.

enhancements 업데이트가 저장되고 보류 중인 커밋 메시지가 표시됩니다.

- 요약 대화 상자에서 변경 사항을 검토한 다음 **Commit** 을 클릭합니다.
커밋에 성공했는지 확인하는 메시지가 표시됩니다.

그 결과 새로운 버전의 director가 생성되고 오른쪽 창에 최신 구성 요소가 나열됩니다.

10.1.2. RHEL 웹 콘솔을 사용하여 에지 이미지 승격을 위한 RHEL의 기존 구성 요소 버전 변경

기본(최신) 버전을 선택했거나 작업장에 포함된 구성 요소의 버전을 선택했습니다. 필요한 경우 이제 원하는 구성 요소의 버전을 변경할 수 있습니다.

이렇게 하려면 다음 사전 요구 사항을 충족했는지 확인한 다음 절차에 따라 해당 devfile의 구성 요소 버전을 변경합니다.

사전 요구 사항

- RHEL 시스템에서 이미지 빌더 대시보드에 액세스했습니다.
- RHEL for Edge 이미지용 블루프린트를 생성했습니다.
- Edge leadership를 위해 RHEL에 하나 이상의 구성 요소를 추가했습니다.

절차

- 이미지 빌더 대시보드에서 편집할 핸들러를 클릭합니다.
특정 요건을 검색하려면 Filter By Name(이름으로 필터) 텍스트 상자에 Quarkus 이름을 입력한 다음 **Enter** 를 누릅니다.
- greater의 오른쪽 상단에서 **패키지 편집** 을 클릭합니다.
Edit Packages 모드 보기에는 패키지 편집 모드가 변경되고 오른쪽 패널에는 현재 credential에 커밋된 구성 요소 이름이 나열됩니다.
- 구성 요소 이름을 클릭합니다.
- 구성 요소 옵션 버전 드롭다운 목록에서 원하는 버전을 선택합니다.
- 변경 사항 적용** 을 클릭합니다.
변경 사항이 저장되고 오른쪽 창에는 최신 변경 사항이 나열됩니다.
- 커밋** 을 클릭합니다.
새로운 버전은 options에 저장됩니다. 보류 중인 커밋이 포함된 메시지가 표시됩니다.
- 요약 대화 상자에서 변경 사항을 검토한 다음 **Commit** 을 클릭합니다.
커밋에 성공했는지 확인하는 메시지가 표시됩니다.

그 결과 새로운 버전의 Makefile이 생성되고 오른쪽 창에 최신 구성 요소가 나열됩니다.

10.1.3. RHEL 웹 콘솔에서 Image Builder를 사용하여 에지 이미지 options용 RHEL에서 구성 요소 제거

생성한 에지 이미지 inventory용 RHEL에서 원하지 않는 구성 요소를 하나 이상 제거하려면 다음 사전 요구 사항을 충족한 다음 절차를 따르십시오.

사전 요구 사항

- RHEL 시스템에서 이미지 빌더 대시보드에 액세스했습니다.
- RHEL for Edge 이미지용 블루프린트를 생성했습니다.
- Edge leadership를 위해 RHEL에 하나 이상의 구성 요소를 추가했습니다.

절차

1. 이미지 빌더 대시보드에서 편집할 핸들러를 클릭합니다.
특정 요건을 검색하려면 Filter By Name(이름으로 필터) 텍스트 상자에 stress 이름을 입력한 다음 Enter를 누릅니다.
2. greater의 오른쪽 상단에서 **패키지 편집**을 클릭합니다.
패키지 편집 모드에 대한 보기 변경 사항. 오른쪽 패널에는 현재 credential에 커밋된 구성 요소 이름이 나열됩니다.
3. 추가 옵션 메뉴에서 **제거**를 클릭합니다.
선택적으로 구성 요소 이름을 클릭한 다음 **제거**를 클릭합니다.
4. **커밋**을 클릭합니다.
보류 중인 커밋이 포함된 메시지가 표시됩니다.
5. 변경 사항을 검토한 다음 **Commit**을 클릭합니다.
커밋에 성공했는지 확인하는 메시지가 표시됩니다.

그 결과 새로운 버전의 director가 생성되고 오른쪽 창에 최신 구성 요소가 나열됩니다.

10.1.4. 명령줄 인터페이스를 사용하여 에지 이미지 options를 위한 RHEL 편집

Image Builder 명령줄을 사용하여 에지 이미지 panic에 대한 RHEL의 사양을 변경할 수 있습니다. 이렇게 하려면 다음 사전 요구 사항을 충족했는지 확인한 후 절차에 따라 해당)을 편집합니다.

사전 요구 사항

- Image Builder 명령줄에 액세스할 수 있습니다.
- 에지 이미지 options용 RHEL을 생성했습니다.

절차

1. 배치를 로컬 텍스트 파일에 저장(export) 합니다.

```
# composer-cli blueprints save BLUEPRINT-NAME
```
2. 선택한 텍스트 편집기로 **BLUEPRINT-NAME.toml** 파일을 편집하고 변경합니다.
편집을 완료하기 전에 파일이 유효한 rootfs인지 확인하십시오.
3. 버전 번호를 늘립니다.
Semantic Versioning scheme을 사용해야 합니다.



참고

버전을 변경하지 않으면 버전의 패키지 구성 요소가 자동으로 증가됩니다.

4. 콘텐츠가 유효한 TOML 사양인지 확인합니다. 자세한 내용은 TOML 설명서를 참조하십시오.



참고

TOML 문서는 커뮤니티 제품이며 Red Hat에서 지원하지 않습니다. 이 툴의 모든 문제는 <https://github.com/toml-lang/toml/issues> 에서 보고할 수 있습니다.

5. 파일을 저장하고 편집기를 종료합니다.
6. Images Builder 명령줄로 `back into Image Builder` 명령줄을 다시 푸시(가져오기) 합니다.

```
# composer-cli blueprints push BLUEPRINT-NAME.toml
```



참고

Makefile을 Image Builder 명령줄로 다시 푸시하는 경우 **.toml** 확장을 포함하여 파일 이름을 제공합니다.

7. 이미지 빌더에 업로드된 콘텐츠가 편집 내용과 일치하는지 확인합니다.

```
# composer-cli blueprints show BLUEPRINT-NAME
```

8. Makefile 및 해당 종속 항목에 나열된 구성 요소 및 버전이 유효한지 확인합니다.

```
# composer-cli blueprints depsolve BLUEPRINT-NAME
```

10.2. 예지 이미지 RHEL 업데이트

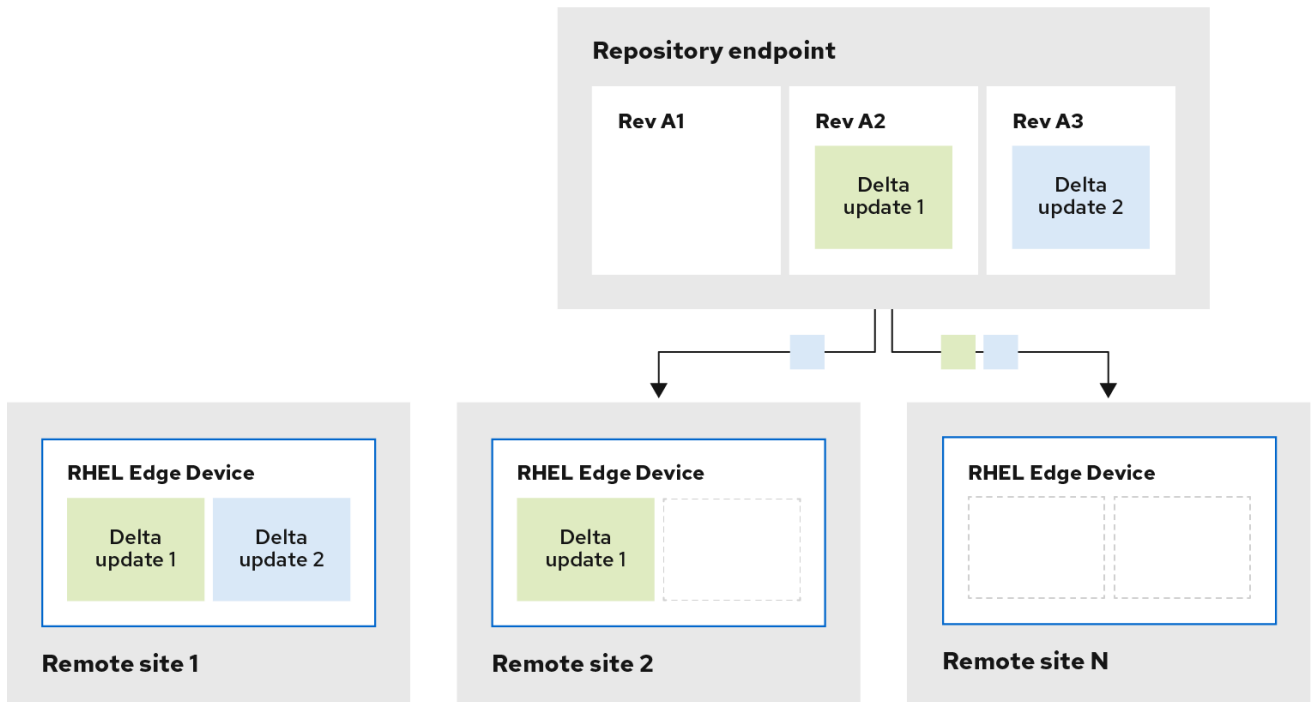
10.2.1. RHEL for Edge 이미지 업데이트 배포 방법

예지 이미지용 RHEL을 사용하면 수동으로 업데이트를 배포하거나 배포 프로세스를 자동화할 수 있습니다. 업데이트는 각 업데이트의 상태를 알 수 있는 원자성 방식으로 적용되며 업데이트는 준비되고 재부팅 시에만 적용됩니다. 장치를 재부팅할 때까지 변경 사항이 표시되지 않으므로 재부팅을 예약하여 최대한의 가동 시간을 유지할 수 있습니다.

업데이트된 OS 콘텐츠만 네트워크를 통해 전송되므로 이미지 업데이트 중에 전체 이미지를 전송하는 것보다 배포 프로세스가 더 효율적입니다. OS 바이너리 및 라이브러리(/usr)는 읽기 전용이며 읽기-쓰기 상태는 **/var** 및 **/etc** 디렉토리에서 유지됩니다.

델타 전송을 사용하면 간헐적이고 저 대역폭 연결이 되거나 로컬 미디어를 사용하여 연결이 끊긴 장치에도 업데이트를 배포할 수 있습니다. 또한 static-deltas를 생성하여 네트워크 사용량을 추가로 줄일 수도 있습니다. static-deltas는 모든 개별 업데이트를 단일 파일 아카이브로 가져오고, 여러 연결을 통해 각 업데이트를 개별적으로 전송하는 대신 단일 TCP 연결을 통해 OS 업데이트를 전송하는 TCP 네트워크 오버헤드를 크게 줄입니다.

다음 다이어그램은 RHEL for Edge 이미지 업데이트 배포 프로세스를 보여줍니다.



125_RHEL_1020

10.2.2. 에지 이미지 업데이트를 위해 수동으로 RHEL 배포

엣지용 RHEL을 편집한 후 이미지 커밋을 업데이트할 수 있습니다. 이미지 빌더는 에지 이미지에 대해 업데이트된 RHEL에 대한 새 커밋을 생성합니다. 이 새 커밋을 사용하여 최신 패키지 버전 또는 추가 패키지와 함께 이미지를 배포합니다.

에지 이미지 업데이트를 위해 RHEL을 배포하려면 사전 요구 사항을 충족해야 하며 절차를 따르십시오.

사전 요구 사항

- RHEL 시스템에서 이미지 빌더 대시보드에 액세스했습니다.
- 에지 이미지 options용 RHEL을 생성했습니다.
- Edge 이미지용 RHEL을 편집했습니다. [RHEL 웹 콘솔에서 이미지 빌더를 사용하여 에지 이미지 청사진용 RHEL 편집](#)을 참조하십시오.

절차

1. 편집한 시행에 대해 Image Builder(이미지 빌더) 대시보드에서 **Create Image** (이미지 만들기)를 클릭합니다.
2. 이미지 생성 창에서 다음 단계를 수행합니다.
 - a. **유형** 드롭다운 목록에서 "RHEL for Edge Commit (.tar)"을 선택합니다.
 - b. **Parent commit** textbox에서 이전에 생성된 상위 커밋 ID를 지정합니다. [에지 이미지 커밋용 RHEL 추출](#)을 참조하십시오.
 - c. Ref 텍스트 상자에 커밋의 이름을 지정하거나 비워 둘 수 있습니다. In the Ref text box, you can either specify a name for your commit or leave it empty. 기본적으로 웹 콘솔은 Ref를 **rhel/9/arch_name/edge**로 지정합니다.

- d. **생성**을 클릭합니다. 이미지 빌더는 업데이트된 credential에 대한 에지 이미지에 대한 RHEL을 생성합니다.
에지 이미지 생성 진행 상황을 보려면 이동 경로의 프로비전 이름을 클릭한 다음 **이미지** 탭을 클릭합니다.



참고

이미지 생성 프로세스를 완료하는 데 몇 분이 걸립니다.

결과 이미지에는 추가한 최신 패키지가 포함되어 있으며 원래 커밋 ID를 상위로 사용합니다.

3. 에지 이미지에 대한 결과 RHEL을 다운로드합니다. Edge 이미지용 RHEL을 다운로드하는 방법에 대한 자세한 내용은 Edge 이미지 [용 RHEL](#) 다운로드를 참조하십시오.
4. OSTree 커밋을 추출합니다. OSTree 커밋을 추출하는 방법에 대한 자세한 내용은 [에지 이미지 커밋용 RHEL 추출](#)을 참조하십시오.
5. 이번에는 Docker 컨테이너를 빌드하고 하위 커밋 ID를 제공합니다.

```
# podman build -t name-of-server --build-arg commit=uuid-child_commit.tar .
```

6. 컨테이너를 실행합니다.

```
# podman run --rm -p 8000:80 name-of-server
```

7. 프로비저닝된 RHEL 시스템의 원래 에지 이미지에서 현재 상태를 확인합니다.

```
$ rpm-ostree status
```

새 커밋 ID가 없는 경우 다음 명령을 실행하여 사용 가능한 업그레이드가 있는지 확인합니다.

```
$ rpm-ostree upgrade --check
```

명령 출력은 현재 활성 OSTree 커밋 ID를 제공합니다.

8. OSTree를 업데이트하여 새로운 OSTree 커밋 ID를 사용할 수 있도록 합니다.

```
$ rpm-ostree upgrade
```

ostree에서 리포지토리에 업데이트가 있는지 확인합니다. yes인 경우 이 새 커밋 업데이트 배포를 활성화할 수 있도록 업데이트를 가져와서 시스템을 재부팅하도록 요청합니다.

9. 현재 상태를 다시 확인합니다.

```
$ rpm-ostree status
```

이제 사용 가능한 두 개의 커밋이 있음을 확인할 수 있습니다.

- 활성 상위 커밋입니다.
- 활성 상태가 아니며 1개의 차이점이 포함된 새 커밋입니다.

10. 새 배포를 활성화하고 새 커밋을 활성화하려면 시스템을 재부팅합니다.

systemctl reboot

Anaconda 설치 프로그램이 새 배포로 재부팅됩니다. 로그인 화면에서 부팅할 수 있는 새 배포를 확인할 수 있습니다.

11. 최신 커밋/배포로 부팅하려면 `rpm-ostree upgrade` 명령에서 부팅 항목을 자동으로 정렬하여 새 배포가 목록에 먼저 표시되도록 합니다. 선택적으로 키보드의 화살표 키를 사용하여 GRUB 메뉴 항목을 선택하고 **Enter** 를 누릅니다.
12. 로그인 사용자 계정 자격 증명을 제공합니다.
13. OSTree 상태를 확인합니다.

\$ rpm-ostree status

명령 출력은 활성 커밋 ID를 제공합니다.

14. 변경된 패키지를 보려면 상위 커밋과 새 커밋 사이에 diff를 실행합니다.

\$ rpm-ostree db diff parent_commit new_commit

업데이트는 설치한 패키지가 사용 가능하고 사용할 준비가 되었음을 보여줍니다.

10.2.3. 명령줄을 사용하여 에지 이미지 업데이트를 수동으로 위한 RHEL 배포

엣지용 RHEL을 편집한 후 이미지 커밋을 업데이트할 수 있습니다. 이미지 빌더는 에지 이미지에 대해 업데이트된 RHEL에 대한 새 커밋을 생성합니다. 새 커밋을 사용하여 최신 패키지 버전 또는 CLI를 사용하는 추가 패키지와 함께 이미지를 배포합니다.

CLI를 사용하여 에지 이미지 업데이트를 위한 RHEL을 배포하려면 사전 요구 사항을 충족해야 하는 다음 절차를 따르십시오.

사전 요구 사항

- 에지 이미지 청사진용 RHEL을 만들었습니다.
- 에지 이미지 청사진을 위해 RHEL을 편집했습니다. [명령줄 인터페이스를 사용하여 에지 이미지 청사진용 RHEL 편집](#) 을 참조하십시오.

절차

1. 다음 인수를 사용하여 에지 커밋(.tar)의 RHEL 이미지를 만듭니다.

```
# composer-cli compose start-ostree --ref ostree_ref --parent parent_commitID_edge -
blueprint_name_ image-type
```

다음과 같습니다.

- 에지 컨테이너 커밋용 RHEL을 생성하는 동안 제공한 참조입니다. 예: **rhel/9/x86_64/edge**.
- **parent_commitID_edge** 는 이전에 생성된 상위 커밋 ID입니다.
- **image-type** 은 **edge-commit** 입니다.
이미지 빌더는 업데이트된 credential에 대한 에지 이미지에 대한 RHEL을 생성합니다.

2. RHEL에서 Edge 이미지 생성 진행 상황을 확인합니다.

```
# composer-cli compose status
```



참고

이미지 생성 프로세스를 완료하는 데 최대 10~30분이 걸릴 수 있습니다.

결과 이미지에는 추가한 최신 패키지가 포함되어 있으며 원래 커밋 ID가 상위 항목으로 포함되어 있습니다.

3. 에지 이미지에 대한 결과 RHEL을 다운로드합니다. 자세한 내용은 [Image Builder 명령줄 인터페이스를 사용하여 에지 이미지용 RHEL 다운로드](#)를 참조하십시오.
4. OSTree 커밋을 추출합니다. 자세한 내용은 [에지 이미지 커밋을 위한 RHEL 추출](#)을 참조하십시오.
5. httpd를 통해 OSTree 커밋을 제공합니다. [Edge 이미지용 RHEL을 설치할 웹 서버 설정](#)을 참조하십시오.
6. 원래 에지 이미지에서 프로비저닝된 RHEL 시스템에서 현재 상태를 확인합니다.

```
$ rpm-ostree status
```

새 커밋 ID가 없는 경우 다음 명령을 실행하여 사용 가능한 업그레이드가 있는지 확인합니다.

```
$ rpm-ostree upgrade --check
```

명령 출력은 현재 활성 OSTree 커밋 ID를 제공합니다.

7. OSTree를 업데이트하여 새 OSTree 커밋 ID를 사용할 수 있도록 합니다.

```
$ rpm-ostree upgrade
```

ostree에서 리포지토리에 업데이트가 있는지 확인합니다. 예, 새 커밋 업데이트의 배포를 활성화할 수 있도록 업데이트를 가져와서 시스템을 재부팅하도록 요청합니다.

8. 현재 상태를 다시 확인합니다.

```
$ rpm-ostree status
```

이제 사용 가능한 커밋이 두 개인지 확인해야 합니다.

- 활성 상위 커밋
- 새 커밋이 활성 상태가 아니며 하나의 추가 차이점이 포함되어 있습니다.

9. 새 배포를 활성화하고 새 커밋을 활성화하려면 시스템을 재부팅합니다.

```
# systemctl reboot
```

Anaconda 설치 프로그램이 새 배포로 재부팅됩니다. 로그인 화면에서 부팅할 수 있는 새 배포를 확인할 수 있습니다.

10. 최신 배포로 부팅하려는 경우 **rpm-ostree upgrade** 명령은 새 배포가 목록에 먼저 포함되도록 부팅 항목을 자동으로 주문합니다. 선택적으로 키보드의 화살표 키를 사용하여 GRUB 메뉴 항목을 선택하고 **Enter** 를 누릅니다.
11. 계정 자격 증명을 사용하여 로그인합니다.
12. OSTree 상태를 확인합니다.

```
$ rpm-ostree status
```

명령 출력은 활성 커밋 ID를 제공합니다.

13. 변경된 패키지를 보려면 상위 커밋과 새 커밋 사이에 diff를 실행합니다.

```
$ rpm-ostree db diff parent_commit new_commit
```

업데이트는 설치한 패키지가 사용 가능하고 사용할 준비가 되었음을 보여줍니다.

10.2.4. 네트워크 기반이 아닌 배포를 위해 RHEL for Edge 이미지 업데이트를 수동으로 배포

엣지용 RHEL을 편집한 후 이미지 커밋을 업데이트할 수 있습니다. 이미지 빌더는 에지 이미지에 대해 업데이트된 RHEL에 대한 새 커밋을 생성합니다. 이 새 커밋을 사용하여 최신 패키지 버전 또는 추가 패키지와 함께 이미지를 배포합니다.

에지 이미지 업데이트를 위해 RHEL을 배포하려면 사전 요구 사항을 충족해야 하며 절차를 따르십시오.

사전 요구 사항

- Edge 시스템의 RHEL이 실행 중입니다.
- OSTree 리포지토리는 HTTP를 통해 제공됩니다.
- 에지 이미지 options용 RHEL을 생성했습니다.
- Edge 이미지용 RHEL을 편집했습니다. [RHEL 웹 콘솔에서 Image Builder를 사용하여 에지 이미지로 RHEL 이미지 편집을 참조하십시오.](#)

절차

1. 편집한 시행에 대해 Image Builder(이미지 빌더) 대시보드에서 **Create Image** (이미지 만들기)를 클릭합니다.
2. **이미지 생성** 창에서 다음 단계를 수행합니다.
 - a. 유형 드롭다운 목록에서 **에지 컨테이너(.tar)의 RHEL**을 선택합니다.
 - b. **Parent** 커밋 텍스트 상자에서 이전에 생성된 상위 커밋 ID를 지정합니다. [에지 이미지 커밋을 위한 RHEL 추출](#)을 참조하십시오.
 - c. **Repository** textbox에서 이미지에 포함할 커밋의 OSTree 리포지토리에 대한 URL을 지정합니다. 예: <http://10.0.2.2:8080/repository/>
 - d. **Ref** textbox에서 이미지에 포함할 에지 컨테이너 커밋을 위한 RHEL을 생성하는 동안 제공한 것과 동일한 참조를 지정합니다. 예를 들면 'rhel/edge/test'입니다.

- e. **생성**을 클릭합니다. 이미지 빌더는 업데이트된 credential에 대한 에지 이미지에 대한 RHEL을 생성합니다.
에지 이미지 생성을 위한 RHEL의 진행 상황을 보려면 이동 경로의 인식 이름을 클릭한 다음 **이미지** 탭을 클릭합니다.



참고

이미지 생성 프로세스를 완료하는 데 몇 분이 걸립니다.

결과 이미지에는 추가한 최신 패키지가 포함되어 있으며 원래 커밋 ID가 상위 항목으로 포함되어 있습니다.

- 에지 이미지에 대한 결과 RHEL을 다운로드합니다. 에지 이미지용 RHEL을 다운로드하는 방법에 대한 자세한 내용은 Edge 이미지 [용 RHEL 이미지 다운로드를 참조하십시오](#).
- 에지 컨테이너 이미지의 RHEL을 Podman에 로드하여 이번에 하위 커밋 ID를 제공합니다.

```
$ cat ./child-commit_ID-container.tar | sudo podman load
```

- Podman**을 실행합니다.

```
# sudo podman run -p 8080:8080 localhost/edge-test
```

- 프로비저닝된 RHEL 시스템의 원래 에지 이미지에서 현재 상태를 확인합니다.

```
$ rpm-ostree status
```

새 커밋 ID가 없는 경우 다음 명령을 실행하여 사용 가능한 업그레이드가 있는지 확인합니다.

```
$ rpm-ostree upgrade --check
```

사용 가능한 업데이트가 있는 경우 명령 출력은 현재 활성 OSTree 커밋 ID와 같은 OSTree 리포지토리에서 사용 가능한 업데이트에 대한 정보를 제공합니다. 사용 가능한 업데이트가 없음을 알리는 메시지가 표시됩니다.

- OSTree를 업데이트하여 새로운 OSTree 커밋 ID를 사용할 수 있도록 합니다.

```
$ rpm-ostree upgrade
```

ostree에서 리포지토리에 업데이트가 있는지 확인합니다. yes인 경우 이 새 커밋 업데이트 배포를 활성화할 수 있도록 업데이트를 가져와서 시스템을 재부팅하도록 요청합니다.

- 현재 상태를 확인합니다.

```
$ rpm-ostree status
```

이제 사용 가능한 두 개의 커밋이 있음을 확인할 수 있습니다.

- 활성 상위 커밋입니다.
- 활성 상태가 아니며 1개의 차이점이 포함된 새 커밋입니다.

- 새 배포를 활성화하고 새 커밋을 활성화하려면 시스템을 재부팅합니다.

```
# systemctl reboot
```

Anaconda 설치 프로그램이 새 배포로 재부팅됩니다. 로그인 화면에서 부팅할 수 있는 새 배포를 확인할 수 있습니다.

10. 최신 커밋/배포로 부팅하려면 **rpm-ostree upgrade** 명령에서 부팅 항목을 자동으로 정렬하여 새 배포가 목록에 먼저 표시되도록 합니다. 선택적으로 키보드의 화살표 키를 사용하여 GRUB 메뉴 항목을 선택하고 **Enter** 를 누릅니다.
11. 로그인 사용자 계정 자격 증명을 제공합니다.
12. OSTree 상태를 확인합니다.

```
$ rpm-ostree status
```

명령 출력은 활성 커밋 ID를 제공합니다.

13. 변경된 패키지를 보려면 상위 커밋과 새 커밋 사이에 diff를 실행합니다.

```
$ rpm-ostree db diff parent_commit new_commit
```

업데이트는 설치한 패키지가 사용 가능하고 사용할 준비가 되었음을 보여줍니다.

10.3. 에지 시스템을 위한 RHEL 업그레이드

10.3.1. RHEL 8 시스템을 RHEL 9로 업그레이드

rpm-ostree rebase 명령을 사용하여 RHEL 8 시스템을 RHEL 9로 업그레이드할 수 있습니다. RHEL 8의 최신 업데이트에서 RHEL 9의 최신 업데이트까지 에지 업그레이드를 위해 기본 RHEL 패키지 세트를 완전히 지원합니다. RHEL 9 이미지가 다운로드되어 백그라운드에서 설치됩니다. 업그레이드가 완료되면 새 RHEL 9 이미지를 사용하려면 시스템을 재부팅해야 합니다.



참고

업그레이드는 가능한 모든 **rpm** 패키지 버전과 포함 기능을 지원하지 않습니다. 패키지 추가를 테스트하여 예상대로 작동하는지 확인해야 합니다.

사전 요구 사항

- Edge 8 시스템용 RHEL 이 실행 중입니다.
- HTTP를 통한 OSTree 리포지토리 서버
- 업그레이드할 Edge 9 이미지에 대해 RHEL 에 대한 SPL을 생성했습니다.

절차

1. Edge 9 이미지에 RHEL을 생성합니다.
 - a. 이미지 작성을 시작합니다.

```
$ sudo composer-cli compose start blueprint-name edge-commit
```

- b. compose 가 완료되면 이미지를 다운로드합니다.
- c. 다운로드한 이미지를 `/var/www/html/` 폴더에 추출합니다.

```
$ sudo tar -xf image_file -C /var/www/html
```

- d. httpd 서비스를 시작합니다.

```
$ systemctl start httpd.service
```

1. 현재 원격 리포지토리 구성을 확인합니다.

```
$ sudo cat /etc/ostree/remotes.d/edge.conf
```

2. 현재 url 저장소를 확인합니다.

```
$ sudo ostree remote show-url edge
```

3. 원격 참조 분기를 나열합니다.

```
$ ostree remote refs edge
```

다음 출력을 볼 수 있습니다.

```
Error: Remote refs not available; server has no summary file
```

4. 새 저장소를 추가하려면 다음을 수행합니다.

- a. 원격 리포지토리를 추가하도록 url 키를 구성합니다. 예를 들어 다음과 같습니다.

```
$ sudo ostree remote add \
--no-gpg-verify rhel9 http://192.168.122.1/repo/
```

- b. 업그레이드할 RHEL 9 커밋을 가리키도록 url 키를 구성합니다. 예를 들어 다음과 같습니다.

```
$ sudo cat /etc/ostree/remotes.d/edge.conf
[remote "edge"]
url=http://192.168.122.1/ostree/repo/
gpg-verify=false
```

- c. URL 이 새 원격 리포지토리로 설정되어 있는지 확인합니다.

```
$ sudo cat /etc/ostree/remotes.d/rhel9.conf
[remote "edge"]
url=http://192.168.122.1/repo/
gpg-verify=false
```

- d. 새 url 저장소를 확인합니다.

```
$ sudo ostree remote show-url rhel9 http://192.168.122.1/ostree-rhel9/repo/
```

- e. 현재 원격 목록 옵션을 나열합니다.

```
$ sudo ostree remote list
```

```
output:
edge
rhel9
```

5. 시스템을 RHEL 9 버전으로 리베이스하여 RHEL 9 버전의 참조 경로를 제공합니다.

```
$ rpm-ostree rebase rhel9:rhel/9/x86_64/edge
```

6. 시스템을 재부팅합니다.

```
$ systemctl reboot
```

7. 사용자 이름과 암호를 입력합니다.

8. 현재 시스템 상태를 확인합니다.

```
$ rpm-ostree status
```

검증

1. 현재 실행 중인 배포의 현재 상태를 확인합니다.

```
$ rpm-ostree status
```

2. 선택 사항: 커널이 관리하는 프로세서 및 작업을 실시간으로 표시합니다.

```
$ top
```

3. 업그레이드가 요구 사항을 지원하지 않는 경우 이전 안정 배포 RHEL 8 버전으로 수동으로 롤백할 수 있습니다.

```
$ sudo rpm-ostree rollback
```

4. 시스템을 재부팅합니다. 사용자 이름과 암호를 입력합니다.

```
$ systemctl reboot
```

재부팅 후 시스템에서 RHEL 9를 성공적으로 실행해야 합니다.



참고

업그레이드가 성공했으며 이전 배포 RHEL 8 버전을 사용하지 않으려면 이전 리포 지토리를 삭제할 수 있습니다.

```
$ sudo ostree remote delete edge
```

10.4. 에지 자동 이미지 업데이트를 위한 RHEL 배포

에지 장치에 에지 이미지의 RHEL을 설치한 후 사용 가능한 이미지 업데이트를 확인하고 자동 적용할 수 있습니다.

`rpm-ostreed-automatic.service`(systemd 서비스) 및 `rpm-ostreed-automatic.timer`(systemd timer)는 검사 및 업그레이드 빈도를 제어합니다. 사용 가능한 업데이트가 있는 경우 스테이징 배포로 표시됩니다.

자동 이미지 업데이트를 배포하려면 다음과 같은 상위 이미지 업데이트가 포함됩니다.

- 이미지 업데이트 정책 업데이트
- 자동 업데이트 다운로드 및 스테이징 활성화

10.4.1. 에지 이미지 업데이트 정책의 RHEL 업데이트 정책 업데이트

이미지 업데이트 정책을 업데이트하려면 Edge 장치의 `/etc/rpm-ostreed.conf` 파일의 `rpm-ostreed.conf` 파일에서 **AutomaticUpdatePolicy** 및 **IdleExitTimeout** 설정을 사용합니다.

AutomaticUpdatePolicy 설정은 자동 업데이트 정책을 제어하며 다음과 같은 옵션이 있습니다.

- **none**: 자동 업데이트를 비활성화합니다. 기본적으로 **AutomaticUpdatePolicy** 설정은 **none**으로 설정됩니다.
- **확인**: rpm-ostree 상태로 사용 가능한 업데이트를 표시하는 데 충분한 메타데이터를 다운로드합니다.
- **Stage**: 재부팅 시 적용되는 업데이트를 다운로드하고 압축을 풉니다.

IdleExitTimeout 설정은 데몬 종료 전의 비활성 시간(초)을 제어하며 다음과 같은 옵션을 갖습니다.

- **0**: auto-exit을 비활성화합니다.
- **60**: 기본적으로 **IdleExitTimeout** 설정은 **60**으로 설정됩니다.

자동 업데이트를 활성화하려면 다음 단계를 수행합니다.

절차

1. `/etc/rpm-ostreed.conf` 파일에서 다음을 업데이트합니다.
 - 확인하려면 **AutomaticUpdatePolicy** 값을 변경합니다.
 - 업데이트 검사를 실행하려면 **IdleExitTimeout**에 값을 초 단위로 지정합니다.
2. **rpm-ostreed** 서비스를 다시 로드하고 **systemd** 타이머를 활성화합니다.

```
# systemctl reload rpm-ostreed
# systemctl enable rpm-ostreed-automatic.timer --now
```

3. rpm-ostree 상태를 확인하여 자동 업데이트 정책이 구성되어 있고 시간이 활성화되어 있는지 확인합니다.

```
# rpm-ostree status
```

명령 출력은 다음을 보여줍니다.

```
State: idle; auto updates enabled (check; last run <minutes> ago)
```

또한 출력에 사용 가능한 업데이트에 대한 정보도 표시됩니다.

10.4.2. RHEL for Edge 자동 다운로드 및 업데이트 스테이징 활성화

이미지 업데이트 정책을 업데이트하여 이미지 업데이트를 확인한 후 업데이트 세부 정보와 함께 해당 업데이트가 표시되는지 확인합니다. 업데이트를 적용하기로 결정하면 정책을 활성화하여 업데이트를 자동으로 다운로드하고 스테이징하십시오. 그런 다음 사용 가능한 이미지 업데이트가 다운로드되어 배포를 위해 준비됩니다. 업데이트가 적용되고 에지 장치를 재부팅할 때 적용됩니다.

업데이트 자동 다운로드 및 스테이징 정책을 활성화하려면 다음 업데이트를 수행합니다.

절차

1. **/etc/rpm-ostreed.conf** 파일에서 'AutomaticUpdatePolicy'를 **스테이징**으로 업데이트합니다.
2. rpm-ostreed 서비스를 다시 로드합니다.

```
# systemctl enable rpm-ostreed-automatic.timer --now
```

3. rpm-ostree 상태 확인

```
# rpm-ostree status
```

명령 출력은 다음을 보여줍니다.

```
State: idle
AutomaticUpdates: stage; rpm-ostreed-automatic.timer: last run <time> ago
```

4. 업데이트를 시작하려면 타이머가 업데이트를 시작하도록 대기하거나 서비스를 수동으로 시작할 수 있습니다.

```
# systemctl start rpm-ostreed-automatic.service
```

업데이트가 시작되면 rpm-ostree 상태에 다음이 표시됩니다.

```
# rpm-ostree status
State: busy
AutomaticUpdates: stage; rpm-ostreed-automatic.service: running
Transaction: automatic (stage)
```

업데이트가 완료되면 배포 목록에 새 배포가 준비되고 원래 부팅된 배포는 그대로 유지됩니다. 새 배포를 사용하여 시스템을 부팅하거나 다음 업데이트를 기다릴 수 있는지 결정할 수 있습니다.

배포 목록을 보려면 **rpm-ostree status** 명령을 실행합니다.

다음은 샘플 출력입니다.

```
# rpm-ostree status
State: idle
AutomaticUpdates: stage; rpm-ostreed-automatic.timer: last run <time> ago
Deployments:
```

업데이트된 패키지 세부 정보를 사용하여 배포 목록을 보려면 **rpm-ostree status -v** 명령을 실행합니다.

10.5. 에지 이미지용 RHEL 롤백

업데이트된 이미지가 성공적으로 배포되었는지 확인할 수 있습니다. 배포에 실패하면 이전 버전(commit)으로 롤백할 수 있습니다. 이전 기능 상태로 롤백하려면 단계를 수동으로 수행하거나 자동화된 프로세스를 사용할 수 있습니다.

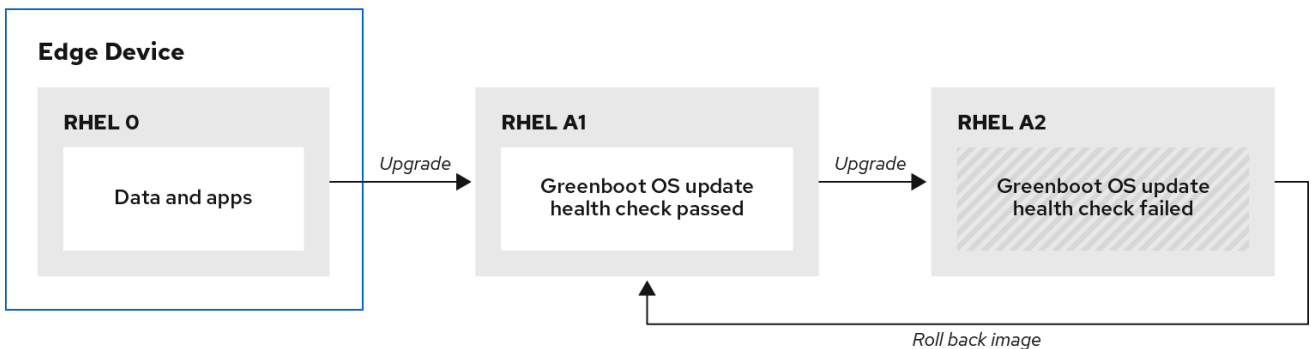
10.5.1. RHEL에서 에지 이미지 롤백 방법

에지 이미지용 RHEL을 사용하면 운영 체제에 트랜잭션 업데이트만 적용됩니다. 트랜잭션 업데이트를 사용하면 실패한 업데이트를 마지막 알려진 양호한 상태로 쉽게 롤백하여 업데이트 중에 시스템 오류를 방지할 수 있습니다.

그린 부팅과 함께 지능형 롤백을 사용하여 애플리케이션 안정성과 보안 업데이트 적용 중에서 선택하는 문제를 제거할 수 있습니다.

greenboot는 rpm-ostree를 활용하고 시스템 시작 시 실행되는 사용자 지정 상태 점검을 실행합니다. 문제가 발생하면 시스템은 변경 사항을 롤백하고 마지막 작업 상태를 유지합니다.

다음 다이어그램에서는 에지 이미지 롤백 프로세스를 위한 RHEL을 보여줍니다.



125_RHEL_1020

10.5.2. 에지 이미지에 대해 RHEL을 수동으로 롤백

RHEL for Edge 이미지 업데이트에 대한 배포가 실패하거나 업데이트가 성공적으로 작동하지 않으면 이전 배포 버전으로 수동으로 롤백할 수 있습니다.

이전 버전으로 롤백하려면 다음 단계를 수행합니다.

절차

1. **rollback** 명령을 실행합니다.

```
# rpm-ostree rollback
```

명령 출력은 이동 중인 커밋 ID에 대한 세부 정보를 제공하고 제거 중인 패키지의 세부 정보와 함께 완료된 트랜잭션을 나타냅니다.

2. 시스템을 재부팅합니다.

systemctl reboot

명령은 안정적인 콘텐츠를 사용하여 이전 커밋을 활성화합니다. 변경 사항이 적용되고 이전 버전이 복원됩니다.

10.5.3. 자동화된 프로세스를 사용하여 에지 이미지에 대해 RHEL 롤백

greenboot 검사는 부팅 프로세스에 통합되는 프레임워크를 제공하며 상태 점검이 실패할 때 **rpm-ostree** 롤백을 트리거할 수 있습니다. 상태 점검의 경우 상태 점검이 통과했는지 실패했는지 여부를 나타내는 사용자 정의 스크립트를 생성할 수 있습니다. 결과에 따라 롤백을 트리거해야 할 시기를 결정할 수 있습니다.

상태 점검 스크립트를 생성하려면 다음 단계를 수행합니다.

절차

- 표준 종료 코드 **0** 을 반환하는 스크립트를 생성합니다.
예를 들어 다음 스크립트는 구성된 DNS 서버를 사용할 수 있는지 확인합니다.

```
#!/bin/bash

DNS_SERVER=$(grep ^nameserver /etc/resolv.conf | head -n 1 | cut -f2 -d" ")
COUNT=0
# check DNS server is available
ping -c1 $DNS_SERVER
while [ $? != '0' ] && [ $COUNT -lt 10 ]; do
  ((COUNT++))
  echo "Checking for DNS: Attempt $COUNT ."
  sleep 10
  ping -c 1 $DNS_SERVER
done
```

- /etc/greenboot/check/required.d/** 에서 상태 점검을 위해 실행 가능한 파일을 포함합니다.

```
chmod +x check-dns.sh
```

다음 재부팅 중에 부팅 프로세스의 일부로 시스템이 **boot-complete.target** 에 진입하기 전에 스크립트가 실행됩니다. 상태 점검이 성공하면 작업이 수행되지 않습니다. 상태 점검에 실패하면 업데이트를 실패로 표시하고 이전 업데이트로 롤백하기 전에 시스템이 여러 번 재부팅됩니다.

검증 단계

기본 게이트웨이에 연결할 수 있는지 확인하려면 다음 상태 점검 스크립트를 실행합니다.

- 표준 종료 코드 **0** 을 반환하는 스크립트를 생성합니다.

```
#!/bin/bash

DEF_GW=$(ip r | awk '/^default/ {print $3}')
SCRIPT=$(basename $0)

count=10
connected=0
ping_timeout=5
interval=5
```

```
while [ $count -gt 0 -a $connected -eq 0 ]; do
    echo "$SCRIPT: Pinging default gateway $DEF_GW"
    ping -c 1 -q -W $ping_timeout $DEF_GW > /dev/null 2>&1 && connected=1 || sleep
    $interval
    ((--count))
done

if [ $connected -eq 1 ]; then
    echo "$SCRIPT: Default gateway $DEF_GW is reachable."
    exit 0
else
    echo "$SCRIPT: Failed to ping default gateway $DEF_GW!" 1>&2
    exit 1
fi
```

2. **/etc/greenboot/check/required.d/** 디렉터리에서 상태 점검을 위한 실행 가능한 파일을 포함합니다.

```
chmod +x check-gw.sh
```

부록 A. 용어 및 명령

이 섹션에서는 **rpm ostree** 용어 및 명령을 제공합니다.

A.1. OSTREE 및 RPM-OSTREE 용어

다음은 OSTree 및 **rpm-ostree** 이미지 컨텍스트에서 사용되는 몇 가지 유용한 용어입니다.

표 A.1. ostree 및 rpm-ostree 용어

용어	정의
OSTree	Linux 기반 운영 체제 버전을 관리하는 데 사용되는 틀입니다. OSTree 트리 보기는 Git과 유사하며 유사한 개념을 기반으로 합니다.
rpm-ostree	운영 체제 업데이트를 호스팅하는 하이브리드 이미지 또는 시스템 패키지.
commit	운영 체제의 릴리스 또는 이미지 버전입니다. 이미지 빌더는 에지 이미지용 RHEL에 대한 ostree 커밋을 생성합니다. 이러한 이미지를 사용하여 에지 서버에 RHEL을 설치하거나 업데이트할 수 있습니다.
Refs	ostree의 분기를 나타냅니다. 항상 최신 커밋을 확인합니다. 예: rhel/9/x86_64/edge .
버전 (Rev)	특정 커밋에 대한 SHA-256
원격	ostree 콘텐츠를 호스팅하는 http 또는 https 끝점입니다. 이는 dnf 리포지토리의 baseurl과 유사합니다.
static-delta	ostree 이미지에 대한 업데이트는 항상 델타 업데이트입니다. 에지 이미지용 RHEL의 경우 파일 수로 인해 TCP 오버헤드가 예상보다 클 수 있습니다. TCP 오버헤드를 방지하기 위해 특정 커밋 간에 static-delta를 생성하고 단일 연결로 업데이트를 보낼 수 있습니다. 이러한 최적화는 제한된 연결로 대규모 배포를 지원합니다.

A.2. OSTREE 명령

이 섹션에서는 ostree 이미지를 설치하거나 관리할 때 사용할 수 있는 몇 가지 ostree 명령을 제공합니다.

표 A.2. ostree 명령

ostree pull	ostree pull-local --repo [path] src ostree pull-local <path> <rev> --repo=<repo-path> ostree pull <URL> <rev> --repo=<repo-path>
-------------	---

ostree 요약	ostree summary -u --repo=<repo-path>
view refs	ostree refs --repo ~/Code/src/osbuild- iot/build/repo/ --list
리포지토리에 커밋 보기	ostree log --repo=/home/gicmo/src/osbuild- iot/build/repo/ <REV>
커밋 검사	ostree show --repo build/repo <REV>
리포지토리의 원격 나열	ostree remote list --repo <repo-path>
REV 해결	ostree rev-parse --repo ~/Code/src/osbuild- iot/build/repo fedora/x86_64/osbuild-demo ostree rev-parse --repo ~/Code/src/osbuild- iot/build/repo b3a008eceeddd0cfd
static-delta 생성	ostree static-delta generate --repo=[path] -- from=REV --to=REV
GPG 키를 사용하여 기존 ostree 커밋 서명	ostree gpg-sign --repo=<repo-path> --gpg- homedir <gpg_home> COMMIT KEY-ID...

A.3. RPM-OSTREE 명령

이 섹션에서는 ostree 이미지를 설치하거나 관리할 때 사용할 수 있는 몇 가지 **rpm-ostree** 명령을 제공합니다.

표 A.3. rpm-ostree 명령

명령	설명
rpm-ostree -- repo=/home/gicmo/Code/src/o sbuild-iot/build/repo/ db list <REV>	이 명령을 실행하면 <REV> 커밋에 있는 패키지가 리포지토리에 나열됩니다.
rpm-ostree rollback	ostree는 deployments 라는 정렬된 부트로더 항목 목록을 관리합니다. 인덱스 0의 항목은 기본 부트로더 항목입니다. 각 항목에는 별도의 /etc 디렉터리가 있지만 모든 항목은 단일 /var 디렉토리를 공유합니다. Tab 을 눌러 시작을 중단하고 부트로더를 사용하여 항목을 선택할 수 있습니다. 그러면 이전 상태로 롤백합니다. 즉, 기본 배포 변경사항은 기본값이 아닌 항목과 함께 배치됩니다.
rpm-ostree status	이 명령은 사용 중인 현재 배포에 대한 정보를 제공합니다. 목록의 첫 번째 배포가 부팅 시 기본값이 되도록 가능한 모든 배포의 이름 및 거부물 순서대로 나열합니다. *로 표시된 배포는 현재 부팅된 배포이며 'r'로 표시는 최신 업그레이드를 나타냅니다.

명령	설명
rpm-ostree db list	이 명령을 사용하여 커밋 또는 커밋 내에 있는 패키지를 확인합니다. 하나 이상의 커밋을 지정해야 하지만 하나 이상의 커밋 범위도 작동합니다.
rpm-ostree db diff	이 명령을 사용하여 두 개의 rev(revisions)의 트리 간에 패키지가 어떻게 다른지 보여줍니다. revs가 제공되지 않으면 부팅된 커밋이 보류 중인 커밋과 비교됩니다. 단일 rev만 제공되는 경우 부팅된 커밋을 해당 rev와 비교합니다.
rpm-ostree upgrade	이 명령은 현재 트리의 최신 버전을 다운로드하여 배포하고, 현재 트리를 다음 부팅의 기본값으로 설정합니다. 이는 실행 중인 파일 시스템 트리에는 영향을 미치지 않습니다. 변경 사항을 적용하려면 재부팅해야 합니다.

추가 리소스

- **rpm-ostree** 매뉴얼 페이지.

A.4. FDO 자동 온보딩 용어

이 섹션에서는 FDO 용어에 대한 정보를 제공합니다.

표 A.4. FDO 용어

명령	설명
FDO	FIDO 장치 온보딩.
장치	모든 하드웨어, 장치 또는 컴퓨터
소유자	장치의 최종 소유자 - 회사 또는 IT 부서.
제조업체	장치 제조업체입니다.
제조업체 서버	장치의 장치 자격 증명을 만듭니다.
제조업체 클라이언트	제조 서버의 위치를 알려줍니다.
ownership Boucher (OV)	개별 장치의 소유권 레코드. 다음 정보를 포함합니다. * 소유자 (fdo-owner-onboarding-service) * rendezvous Server - FIDO 서버(fdo-rendezvous-server) * 장치 (한 가지 조합) (fdo-manufacturing-service)
장치 인증 정보 (DC)	주요 인증 정보 및 rendezvous는 제조 장치에 저장됩니다.

명령	설명
키	제조 서버를 구성하는 키 * key_path * cert_path * key_type * mfg_string_type: 장치 일련 번호 * allowed_key_storage_types: 사용 중인 장치를 인증하는 데 사용되는 데이터를 보호하는 파일 시스템 및 신뢰할 수 있는 플랫폼 모듈(TPM)입니다.
rendezvous 서버	장치 소유자가 누구인지 확인하기 위해 프로세스에서 사용하는 서버에 대한 링크

추가 리소스

- [FIDO IoT 사양](#)

A.5. FDO 자동 온보딩 기술

다음은 FDO 자동 온보딩에 사용되는 기술입니다.

표 A.5. ostree 및 rpm-ostree 용어

기술	정의
UEFI	통합 확장 펌웨어 인터페이스.
RHEL	Red Hat® Enterprise Linux® 운영 체제(OS)
rpm-ostree	배경 이미지 기반 업그레이드.
Greenboot	rpm-ostree의 systemd에 대한 Healthcheck 프레임워크입니다.
Osbuild	운영 체제 아티팩트를 위한 파이프라인 기반 빌드 시스템입니다.
컨테이너	Linux® 컨테이너는 나머지 시스템과 격리된 1개 이상의 프로세스 집합입니다.
Coreos-installer	RHEL 이미지 설치를 지원하고 UEFI로 시스템을 부팅합니다.

기술	정의
FIDO FDO	구성 및 온보딩 장치를 프로비저닝하기 위한 규격 프로토콜입니다.