Red Hat Enterprise Linux 9

기본 시스템 설정 구성

시스템의 필수 기능을 설정하고 시스템 환경을 사용자 정의

Last Updated: 2024-04-02

Red Hat Enterprise Linux 9 기본 시스템 설정 구성

시스템의 필수 기능을 설정하고 시스템 환경을 사용자 정의

법적 공지

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

초록

기본 시스템 관리 작업을 수행하고, 환경 설정을 구성하고, 시스템을 등록하고, 네트워크 액세스 및 시스템 보안을 구성합니다. 사용자, 그룹 및 파일 권한을 관리합니다. 여러 RHEL 시스템에서 시스템 구성 인터페이스를 관리하려면 시스템 역할을 사용합니다. 효율적인 서비스 관리를 위해 systemd를 사용합니다. chrony로 NTP(Network Time Protocol)를 구성합니다. ReaR을 사용하여 시스템을 백업 및 복원합니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체 4

RED HAT 문서에 관한 피드백 제공 5

1장. 기본 네트워크 액세스 구성 및 관리 6
1.1. 그래픽 설치 모드에서 네트워크 및 호스트 이름 구성 6
1.2. NMCLI를 사용하여 이더넷 연결 구성 7
1.3. NMTUI를 사용하여 이더넷 연결 구성 10
1.4. RHEL 웹 콘솔에서 네트워킹 관리 13
1.5. RHEL 시스템 역할을 사용하여 네트워킹 관리 14
1.6. 추가 리소스 15

2장. 시스템 등록 및 서브스크립션 관리 16
2.1. 설치 후 시스템 등록 16
2.2. 웹 콘솔에서 자격 증명을 사용하여 서브스크립션 등록 17
2.3. GNOME에 RED HAT 계정을 사용하여 시스템 등록 18
2.4. GNOME에서 활성화 키를 사용하여 시스템 등록 19

3장. RED HAT 지원 액세스 21
3.1. RED HAT 고객 포털을 통해 RED HAT 지원 받기 21
3.2. SOSREPORT를 사용하여 문제 해결 21

4장. 기본 환경 설정 변경 23
4.1. 날짜 및 시간 구성 23
4.2. 시스템 로케일 구성 23
4.3. 키보드 레이아웃 구성 24
4.4. 데스크탑 GUI를 사용하여 언어 변경 25
4.5. 추가 리소스 27

5장. OPENSSH로 두 시스템 간의 보안 통신 사용 28
5.1. SSH 및 OPENSSH 28
5.2. OPENSSH 서버 구성 및 시작 29
5.3. 키 기반 인증을 위한 OPENSSH 서버 설정 31
5.4. SSH 키 쌍 생성 31
5.5. 스마트 카드에 저장된 SSH 키 사용 33
5.6. OPENSSH의 보안 강화 34
5.7. SSH 건너뛰기 호스트를 사용하여 원격 서버에 연결 37
5.8. SSH-AGENT를 사용하여 SSH 키가 있는 원격 시스템에 연결 39
5.9. 추가 리소스 39

6장. 기본 시스템 보안 구성 41
6.1. FIREWALLD 서비스 활성화 41
6.2. 기본 SELINUX 설정 관리 41
6.3. SELINUX의 필수 상태 확인 42
6.4. 추가 리소스 43

7장. RHEL 시스템 역할 소개 44

8장. 로그 파일을 사용하여 문제 해결 46
8.1. SYSLOG 메시지를 처리하는 서비스 46
8.2. SYSLOG 메시지를 저장하는 하위 디렉터리 46
8.3. 웹 콘솔을 사용하여 로그 파일 검사 46
8.4. 명령줄을 사용하여 로그 보기 47
8.5. 추가 리소스 48

9장. 사용자 및 그룹 관리 **49**
9.1. 사용자 및 그룹 계정 관리 소개 49
9.2. 사용자 계정 관리 시작하기 50
9.3. 명령줄에서 사용자 관리 53
9.4. 웹 콘솔에서 사용자 계정 관리 58
9.5. 명령줄을 사용하여 사용자 그룹 편집 63
9.6. 루트 암호 변경 및 재설정 69

10장. SUDO 액세스 관리 **73**
10.1. SUDOERS의 사용자 권한 부여 73
10.2. 사용자에게 SUDO 액세스 권한 부여 75
10.3. 권한이 없는 사용자가 특정 명령을 실행하도록 활성화 76

11장. 파일 시스템 권한 관리 **80**
11.1. 파일 권한 관리 80
11.2. 액세스 제어 목록 관리 89
11.3. RECEIVER 관리 91

12장. SYSTEMD 관리 **99**
12.1. SYSTEMD 장치 파일 위치 100
12.2. SYSTEMCTL을 사용하여 시스템 서비스 관리 100
12.3. 대상 시스템 상태로 부팅 110
12.4. 시스템 종료, 일시 중지 및 절전 관리 115

13장. 시간 동기화 구성 **123**
13.1. CHRONY 모음을 사용하여 NTP 구성 123
13.2. CHRONY 사용 125
13.3. HW 타임스탬프링이 있는 CHRONY 133
13.4. CHRONY의 NTS(NETWORK TIME SECURITY) 개요 138

14장. 시스템 복구 및 복원 **143**
14.1. REAR 설정 143
14.2. 64비트 IBM Z 아키텍처에서 REAR RESCUE 이미지 사용 145

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.

Jira를 통해 피드백 제출 (등록 필요)

1. Jira 웹 사이트에 로그인합니다.
2. 상단 탐색 모음에서 **생성** 을 클릭합니다.
3. **Summary** (요약) 필드에 설명 제목을 입력합니다.
4. **Description** (설명) 필드에 개선을 위한 제안을 입력합니다. 문서의 관련 부분에 대한 링크를 포함합니다.
5. 대화 상자 하단에서 **생성** 을 클릭합니다.

1장. 기본 네트워크 액세스 구성 및 관리

이 섹션에서는 Red Hat Enterprise Linux에서 네트워크 설정을 구성하는 방법에 대한 기본 옵션만 설명합니다.

1.1. 그래픽 설치 모드에서 네트워크 및 호스트 이름 구성

이 절차의 단계에 따라 네트워크 및 호스트 이름을 구성합니다.

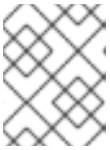
절차

1. **설치 요약** 창에서 **네트워크 및 호스트** 이름을 클릭합니다.
2. 왼쪽 창의 목록에서 인터페이스를 선택합니다. 자세한 내용은 오른쪽 창에 표시됩니다.

참고

영구적인 이름으로 네트워크 장치를 식별하는 데 사용되는 네트워크 장치 이름 지정 표준 유형은 여러 가지가 있습니다(예: **em1** 및 **wl3sp0**). 이러한 표준에 대한 자세한 내용은 *네트워킹 구성 및 관리를* 참조하십시오.

3. 선택한 인터페이스를 활성화하거나 비활성화하려면 **ON/OFF** 스위치를 전환합니다.



참고

설치 프로그램은 로컬 액세스 가능한 인터페이스를 자동으로 감지하므로 수동으로 추가하거나 제거할 수 없습니다.

4. **+** 를 클릭하여 다음 중 하나일 수 있는 가상 네트워크 인터페이스를 추가합니다. 팀(폐기됨), 본딩, 브리지 또는 VLAN.
5. 가상 인터페이스를 제거하려면 **-** 를 클릭합니다.
6. **Configure** (구성)를 클릭하여 기존 인터페이스(가상 및 물리적)의 IP 주소, DNS 서버 또는 라우팅 구성과 같은 설정을 변경합니다.
7. 시스템의 호스트 이름을 **호스트 이름** 필드에 입력합니다.

참고

- 호스트 이름은 ***hostname.domainname*** 형식의 FQDN(정규화된 도메인 이름) 또는 도메인이 없는 짧은 호스트 이름일 수 있습니다. 대부분의 네트워크에는 연결된 시스템을 도메인 이름으로 자동으로 공급하는 DHCP(Dynamic Host Configuration Protocol) 서비스가 있습니다. DHCP 서비스에서 이 시스템에 도메인 이름을 할당할 수 있도록 하려면 짧은 호스트 이름만 지정합니다.

- 고정 IP 및 호스트 이름 구성을 사용하는 경우 계획된 시스템 사용 사례에 따라 짧은 이름 또는 FQDN을 사용할지 여부에 따라 달라집니다. Red Hat Identity Management는 프로비저닝 중에 FQDN을 구성하지만 일부 타사 소프트웨어 제품에는 짧은 이름이 필요할 수 있습니다. 두 경우 모두 모든 상황에서 두 양식의 가용성을 보장하기 위해 **IP FQDN short-alias** 형식으로 **/etc/hosts** 의 호스트에 대한 항목을 추가합니다.

- **localhost** 값은 대상 시스템에 대한 특정 정적 호스트 이름이 구성되어 있지 않으며, 설치된 시스템의 실제 호스트 이름은 DHCP 또는 DNS를 사용하는 NetworkManager를 사용하여 네트워크 구성을 처리하는 동안 구성됩니다.

- 호스트 이름은 영숫자 문자만 포함할 수 있으며 **-** 또는 **.** .. 호스트 이름은 64자 이상이어야 합니다. 호스트 이름은 **-** 및 로 시작하거나 종료할 수 없습니다 **.** DNS를 준수하려면 FQDN의 각 부분이 63자 미만이어야 하며 점을 포함한 FQDN 길이에 255자를 초과해서는 안 됩니다.

8. **Apply(적용)**를 클릭하여 설치 프로그램 환경에 호스트 이름을 적용합니다.

9. 또는 **네트워크 및 호스트 이름** 창에서 무선 옵션을 선택할 수 있습니다. 오른쪽 창에서 **네트워크 선택을** 클릭하여eo 연결을 선택하고 필요한 경우 암호를 입력한 다음 **완료** 를 클릭합니다.

추가 리소스

- 고급 RHEL 9 설치 수행

1.2. NMCLI를 사용하여 이더넷 연결 구성

이더넷을 통해 호스트를 네트워크에 연결하는 경우 **nmcli** 유틸리티를 사용하여 명령줄에서 연결의 설정을 관리할 수 있습니다.

사전 요구 사항

- 물리적 또는 가상 이더넷 NIC(네트워크 인터페이스 컨트롤러)가 서버 구성에 있습니다.

절차

1. NetworkManager 연결 프로필을 나열합니다.

```
# nmcli connection show
NAME                UUID                                  TYPE      DEVICE
Wired connection 1  a5eb6490-cc20-3668-81f8-0314a27f3f75  ethernet  enp1s0
```

기본적으로 NetworkManager는 호스트의 각 NIC에 대한 프로필을 생성합니다. 이 NIC를 특정 네트워크에만 연결하려는 경우 자동으로 생성된 프로필을 조정합니다. 이 NIC를 다른 설정으로 네트워크에 연결하려는 경우 각 네트워크에 대한 개별 프로필을 생성합니다.

2. 추가 연결 프로필을 생성하려면 다음을 입력합니다.

```
# nmcli connection add con-name <connection-name> ifname <device-name> type ethernet
```

기존 프로필을 수정하려면 이 단계를 건너뜁니다.

3. 선택 사항: 연결 프로필의 이름을 변경합니다.

```
# nmcli connection modify "Wired connection 1" connection.id "Internal-LAN"
```

프로필이 여러 개인 호스트에서 의미 있는 이름을 사용하면 프로필의 용도를 쉽게 식별할 수 있습니다.

4. 연결 프로필의 현재 설정을 표시합니다.

```
# nmcli connection show Internal-LAN
...
connection.interface-name:    enp1s0
connection.autoconnect:       yes
ipv4.method:                  auto
ipv6.method:                  auto
...
```

5. IPv4 설정을 구성합니다.

- DHCP를 사용하려면 다음을 입력합니다.

```
# nmcli connection modify Internal-LAN ipv4.method auto
```

ipv4.method 가 이미 **auto** (기본값)로 설정된 경우 이 단계를 건너뜁니다.

- 정적 IPv4 주소, 네트워크 마스크, 기본 게이트웨이, DNS 서버 및 검색 도메인을 설정하려면 다음을 입력합니다.

```
# nmcli connection modify Internal-LAN ipv4.method manual ipv4.addresses 192.0.2.1/24 ipv4.gateway 192.0.2.254 ipv4.dns 192.0.2.200 ipv4.dns-search example.com
```

6. IPv6 설정을 구성합니다.

- SLAAC(상태 비저장 주소 자동 구성)를 사용하려면 다음을 입력합니다.

```
# nmcli connection modify Internal-LAN ipv6.method auto
```

ipv6.method 가 이미 **auto** (기본값)로 설정된 경우 이 단계를 건너뜁니다.

- 정적 IPv6 주소, 네트워크 마스크, 기본 게이트웨이, DNS 서버 및 검색 도메인을 설정하려면 다음을 입력합니다.

```
# nmcli connection modify Internal-LAN ipv6.method manual ipv6.addresses 2001:db8:1::fffe/64 ipv6.gateway 2001:db8:1::fffe ipv6.dns 2001:db8:1::ffbb ipv6.dns-search example.com
```

7. 프로필의 다른 설정을 사용자 지정하려면 다음 명령을 사용합니다.

```
# nmcli connection modify <connection-name> <setting> <value>
```

값을 따옴표로 묶거나 spaces로 묶습니다.

8. 프로필을 활성화합니다.

```
# nmcli connection up Internal-LAN
```

검증

1. NIC의 IP 설정을 표시합니다.

```
# ip address show enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 52:54:00:17:b8:b6 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute enp1s0
       valid_lft forever preferred_lft forever
    inet6 2001:db8:1::fffe/64 scope global noprefixroute
       valid_lft forever preferred_lft forever
```

2. IPv4 기본 게이트웨이를 표시합니다.

```
# ip route show default
default via 192.0.2.254 dev enp1s0 proto static metric 102
```

3. IPv6 기본 게이트웨이를 표시합니다.

```
# ip -6 route show default
default via 2001:db8:1::ffee dev enp1s0 proto static metric 102 pref medium
```

4. DNS 설정을 표시합니다.

```
# cat /etc/resolv.conf
search example.com
nameserver 192.0.2.200
nameserver 2001:db8:1::ffbb
```

여러 연결 프로필이 동시에 활성화된 경우 **이름 서버** 항목의 순서는 이러한 프로필의 DNS 우선 순위 값과 연결 유형에 따라 달라집니다.

5. **ping** 유틸리티를 사용하여 이 호스트가 다른 호스트에 패킷을 보낼 수 있는지 확인합니다.

```
# ping <host-name-or-IP-address>
```

문제 해결

- 네트워크 커넥터가 호스트와 스위치에 연결되어 있는지 확인합니다.
- 링크 실패가 이 호스트에만 있는지 또는 동일한 스위치에 연결된 다른 호스트에 있는지 확인합니다.

- 네트워크케이블 및 네트워크 인터페이스가 예상대로 작동하는지 확인합니다. 하드웨어 진단 단계를 수행하고 결함 variables 및 네트워크 인터페이스 카드를 교체합니다.
- 디스크의 구성이 장치의 구성과 일치하지 않는 경우 NetworkManager를 시작하거나 다시 시작하면 장치의 구성을 반영하는 메모리 내 연결이 생성됩니다. 자세한 내용과 이 문제를 방지하는 방법은 NetworkManager 서비스 솔루션을 다시 시작한 후 NetworkManager 연결 중복을 참조하십시오.

추가 리소스

- **nm-settings(5)** 매뉴얼 페이지
- 기본 게이트웨이를 제공하기 위해 특정 프로필을 사용하지 않도록 NetworkManager 구성
- DNS 서버 순서 구성

1.3. NMTUI를 사용하여 이더넷 연결 구성

이더넷을 통해 호스트를 네트워크에 연결하는 경우 **nmtui** 애플리케이션을 사용하여 텍스트 기반 사용자 인터페이스에서 연결의 설정을 관리할 수 있습니다. **nmtui** 를 사용하여 새 프로필을 만들고 그래픽 인터페이스 없이 호스트에서 기존 프로필을 업데이트합니다.

참고

nmtui:

- 커서 키를 사용하여 이동합니다.
- 버튼을 선택하고 **Enter** 를 누릅니다.
- **스페이스** 를 사용하여 확인란을 선택하고 선택 취소합니다.

사전 요구 사항

- 물리적 또는 가상 이더넷 NIC(네트워크 인터페이스 컨트롤러)가 서버 구성에 있습니다.

절차

1. 연결에 사용하려는 네트워크 장치 이름을 모르는 경우 사용 가능한 장치를 표시합니다.

```
# nmcli device status
DEVICE     TYPE      STATE                CONNECTION
enp1s0     ethernet  unavailable          --
...
```

2. nmtui **시작**:

```
# nmtui
```

3. **Edit a connection** 을 선택하고 **Enter** 를 누릅니다.
4. 새 연결 프로필을 추가하거나 기존 프로필을 수정할지 선택합니다.
   - 새 프로필을 생성하려면 다음을 수행합니다.

i. **Add** (추가) 버튼을 누릅니다.

ii. 네트워크 유형 목록에서 **이더넷** 을 선택하고 **Enter** 키를 누릅니다.

- 기존 프로필을 수정하려면 목록에서 프로필을 선택하고 **Enter** 키를 누릅니다.

5. 선택 사항: 연결 프로필의 이름을 업데이트합니다.
   프로필이 여러 개인 호스트에서 의미 있는 이름을 사용하면 프로필의 용도를 쉽게 식별할 수 있습니다.

6. 새 연결 프로필을 생성하는 경우 장치 필드에 네트워크 **장치** 이름을 입력합니다.

7. 환경에 따라 그에 따라 **IPv4 구성 및 IPv6 구성** 영역에서 IP 주소 설정을 구성합니다. 이를 위해 다음 영역 옆에 있는 버튼을 누른 후 다음을 선택합니다.

   - **비활성화됨** (이 연결에 IP 주소가 필요하지 않은 경우).
   - **자동으로** DHCP 서버가 이 NIC에 IP 주소를 동적으로 할당하는 경우입니다.
   - **수동**: 네트워크에 고정 IP 주소 설정이 필요한 경우입니다. 이 경우 추가 필드를 채워야 합니다.

     i. 추가 필드를 표시하도록 구성할 프로토콜 옆에 있는 **Show** 버튼을 누릅니다.

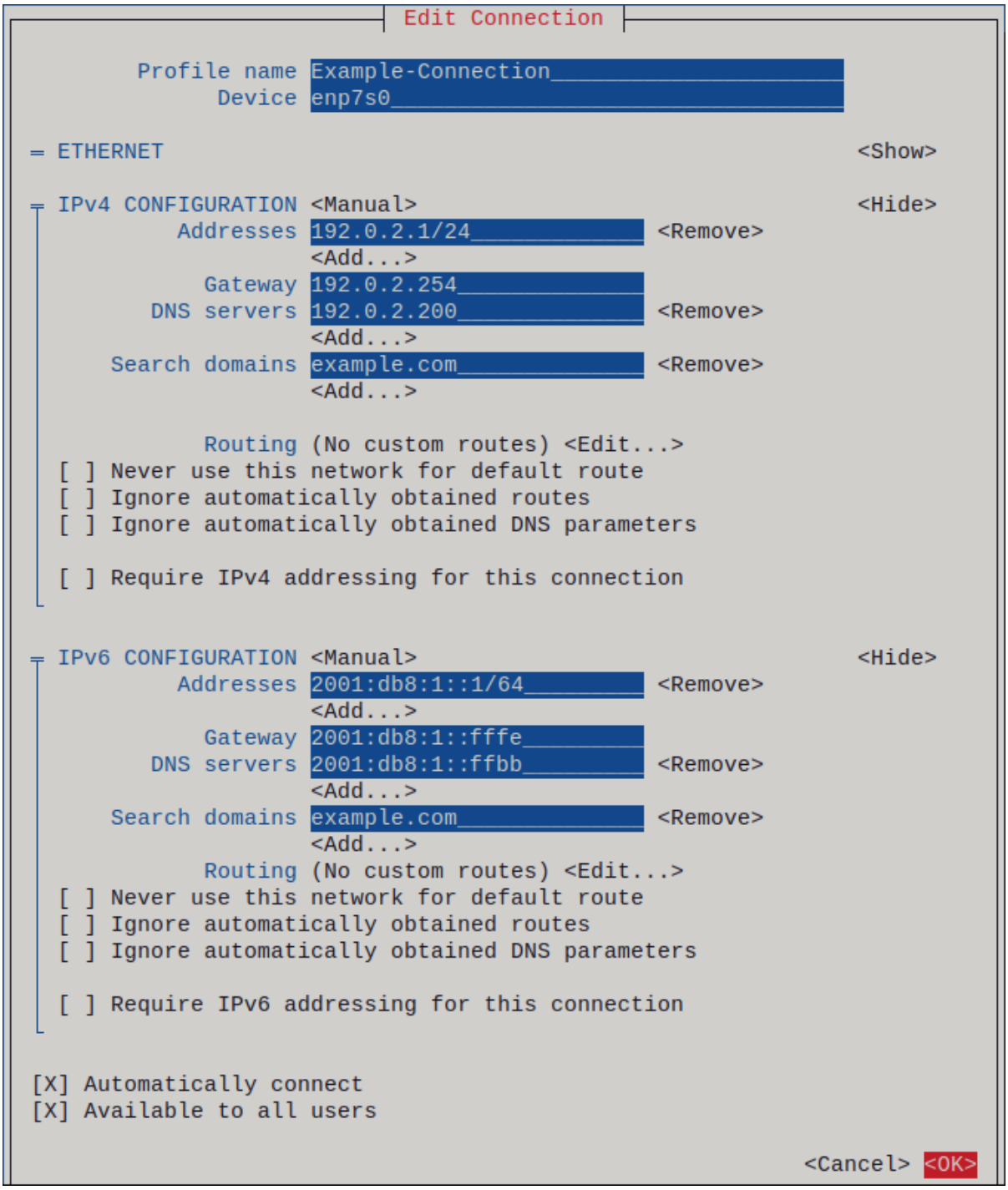
     ii. **주소** 옆에 있는 **추가** 버튼을 누른 다음 CIDR(Classless Inter-Domain Routing) 형식으로 IP 주소와 서브넷 마스크를 입력합니다.
     서브넷 마스크를 지정하지 않으면 NetworkManager는 IPv4 주소에 대해 **/32** 서브넷 마스크를 설정하고 IPv6 주소에 대해 **/64** 를 설정합니다.

     iii. 기본 게이트웨이의 주소를 입력합니다.

     iv. **DNS 서버** 옆에 있는 **추가** 버튼을 누른 후 DNS 서버 주소를 입력합니다.

     v. **검색 도메인 옆에 있는추가 버튼을 누른** 후 DNS 검색 도메인을 입력합니다.

그림 1.1. 고정 IP 주소 설정이 포함된 이더넷 연결 예



8. **OK** 버튼을 눌러 새 연결을 만들고 자동으로 활성화합니다.
9. **뒤로** 버튼을 눌러 메인 메뉴로 돌아갑니다.
10. **Quit** 를 선택하고 **Enter** 를 눌러 **nmtui** 애플리케이션을 종료합니다.

검증

1. NIC의 IP 설정을 표시합니다.

```
# ip address show enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 52:54:00:17:b8:b6 brd ff:ff:ff:ff:ff:ff
```

```
    inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute enp1s0
       valid_lft forever preferred_lft forever
    inet6 2001:db8:1::fffe/64 scope global noprefixroute
       valid_lft forever preferred_lft forever
```

2. IPv4 기본 게이트웨이를 표시합니다.

```
# ip route show default
default via 192.0.2.254 dev enp1s0 proto static metric 102
```

3. IPv6 기본 게이트웨이를 표시합니다.

```
# ip -6 route show default
default via 2001:db8:1::ffee dev enp1s0 proto static metric 102 pref medium
```

4. DNS 설정을 표시합니다.

```
# cat /etc/resolv.conf
search example.com
nameserver 192.0.2.200
nameserver 2001:db8:1::ffbb
```

   여러 연결 프로필이 동시에 활성화된 경우 **이름 서버** 항목의 순서는 이러한 프로필의 DNS 우선 순위 값과 연결 유형에 따라 달라집니다.

5. **ping** 유틸리티를 사용하여 이 호스트가 다른 호스트에 패킷을 보낼 수 있는지 확인합니다.

```
# ping <host-name-or-IP-address>
```

문제 해결

- 네트워크 커넥터가 호스트와 스위치에 연결되어 있는지 확인합니다.
- 링크 실패가 이 호스트에만 있는지 또는 동일한 스위치에 연결된 다른 호스트에 있는지 확인합니다.
- 네트워크케이블 및 네트워크 인터페이스가 예상대로 작동하는지 확인합니다. 하드웨어 진단 단계를 수행하고 결함 variables 및 네트워크 인터페이스 카드를 교체합니다.
- 디스크의 구성이 장치의 구성과 일치하지 않는 경우 NetworkManager를 시작하거나 다시 시작하면 장치의 구성을 반영하는 메모리 내 연결이 생성됩니다. 자세한 내용과 이 문제를 방지하는 방법은 NetworkManager 서비스 솔루션을 다시 시작한 후 NetworkManager 연결 중복을 참조하십시오.

추가 리소스

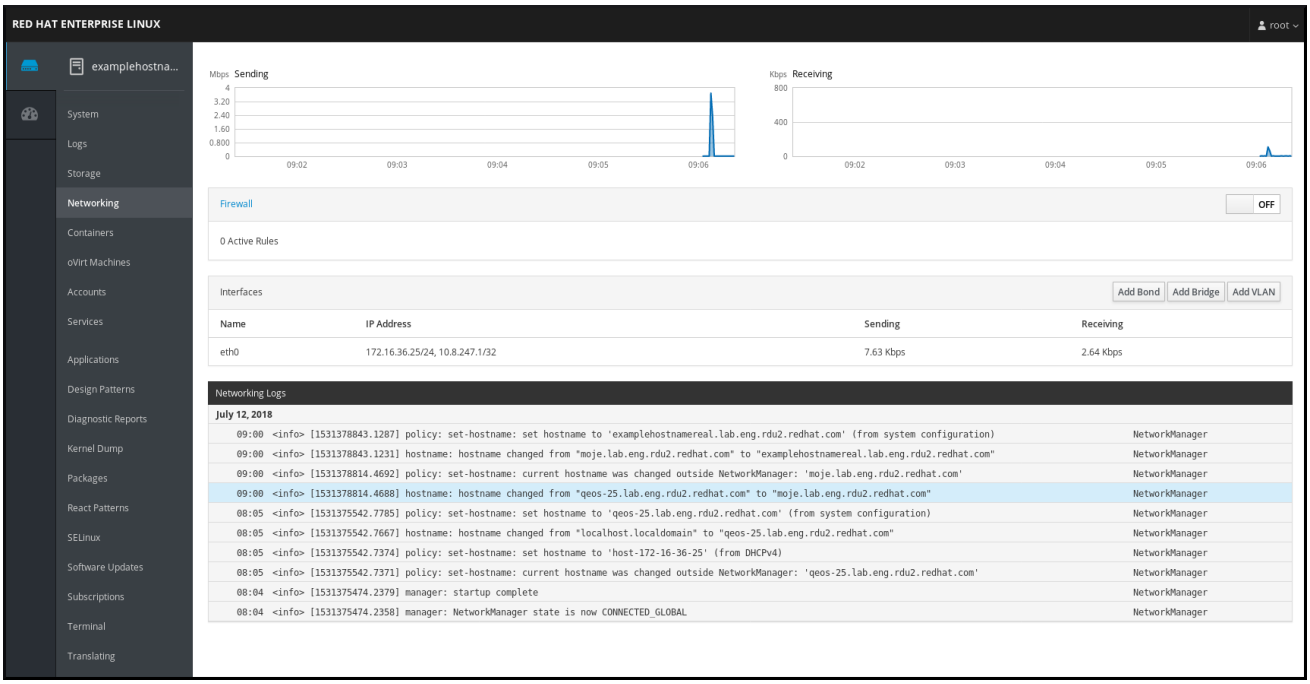
- 기본 게이트웨이를 제공하기 위해 특정 프로필을 사용하지 않도록 NetworkManager 구성
- DNS 서버 순서 구성

1.4. RHEL 웹 콘솔에서 네트워킹 관리

웹 콘솔에서 **네트워킹** 메뉴를 사용하면 다음을 수행할 수 있습니다.

- 현재 수신 및 전송된 패킷을 표시하려면 다음을 수행합니다.
- 사용 가능한 네트워크 인터페이스의 가장 중요한 특성 표시
- 네트워킹 로그의 콘텐츠를 표시하려면 다음을 수행합니다.
- 다양한 유형의 네트워크 인터페이스 추가 (bond, team, bridge, VLAN)

그림 1.2. RHEL 웹 콘솔에서 네트워킹 관리



1.5. RHEL 시스템 역할을 사용하여 네트워킹 관리

네트워크 역할을 사용하여 여러 대상 시스템에서 네트워킹 연결을 구성할 수 있습니다.

네트워크 역할을 사용하면 다음 유형의 인터페이스를 구성할 수 있습니다.

- 이더넷
- Bridge
- 본딩
- VLAN
- MacVLAN
- InfiniBand

각 호스트에 필요한 네트워킹 연결은 **network_connections** 변수 내에서 목록으로 제공됩니다.

주의

네트워크 역할은 **network _connections** 변수에 지정된 대로 정확하게 대상 시스템의 모든 연결 프로필을 업데이트하거나 생성합니다. 따라서 옵션이 시스템에만 존재하지만 **network _connections 변수에 없는 경우 네트워크** 역할은 지정된 프로필에서 옵션을 제거합니다.

다음 예제에서는 **네트워크** 역할을 적용하여 필수 매개 변수를 사용한 이더넷 연결이 있는지 확인하는 방법을 보여줍니다.

네트워크 역할을 적용하여 필요한 매개 변수를 사용하여 이더넷 연결을 설정하는 예제 Playbook

```
# SPDX-License-Identifier: BSD-3-Clause
---
- hosts: managed-node-01.example.com
  vars:
    network_connections:

      # Create one Ethernet profile and activate it.
      # The profile uses automatic IP addressing
      # and is tied to the interface by MAC address.
      - name: prod1
        state: up
        type: ethernet
        autoconnect: yes
        mac: "00:00:5e:00:53:00"
        mtu: 1450

  roles:
    - rhel-system-roles.network
```

추가 리소스

- RHEL 시스템 역할을 사용하도록 제어 노드 및 관리형 노드 준비

1.6. 추가 리소스

- 네트워킹 구성 및 관리

2장. 시스템 등록 및 서브스크립션 관리

서브스크립션은 운영 체제 자체를 포함하여 Red Hat Enterprise Linux에 설치된 제품에 적용됩니다.

Red Hat Content Delivery Network 서브스크립션을 사용하여 다음을 추적할 수 있습니다.

- 등록된 시스템
- 시스템에 설치된 제품
- 설치된 제품에 연결된 서브스크립션

2.1. 설치 후 시스템 등록

이미 설치 프로세스 중에 등록하지 않은 경우 다음 절차를 사용하여 시스템을 등록합니다.

사전 요구 사항

- Red Hat 고객 포털의 유효한 사용자 계정.
- Red Hat 로그인 생성 페이지를 참조하십시오.
- RHEL 시스템의 활성 서브스크립션입니다.
- 설치 프로세스에 대한 자세한 내용은 표준 RHEL 9 설치 수행을 참조하십시오.

절차

1. 시스템을 한 단계로 등록하고 자동으로 등록합니다.

```
# subscription-manager register --username <username> --password <password> --auto-attach
Registering to: subscription.rhsm.redhat.com:443/subscription
The system has been registered with ID: 37to907c-ece6-49ea-9174-20b87ajk9ee7
The registered system name is: client1.idm.example.com
Installed Product Current Status:
Product Name: Red Hat Enterprise Linux for x86_64
Status:       Subscribed
```

이 명령은 Red Hat Customer Portal 사용자 이름과 암호를 입력하라는 메시지를 표시합니다.

등록 프로세스가 실패하면 시스템을 특정 풀로 등록할 수 있습니다. 이 작업을 수행하는 방법에 대한 지침은 다음 단계를 진행합니다.

a. 필요한 서브스크립션의 풀 ID를 확인합니다.

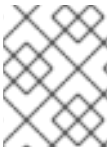
```
# subscription-manager list --available
```

이 명령은 Red Hat 계정에 사용 가능한 모든 서브스크립션을 표시합니다. 모든 서브스크립션에 대해 풀 ID를 포함하여 다양한 특성이 표시됩니다.

b. *pool_id* 를 이전 단계에서 확인한 풀 ID로 교체하여 시스템에 적절한 서브스크립션을 연결합니다.

```
# subscription-manager attach --pool=pool_id
```

-



참고

Red Hat Insights에 시스템을 등록하려면 **rhc connect** 유틸리티를 사용할 수 있습니다. 원격 호스트 구성 설정을 참조하십시오.

추가 리소스

- 고객 포털에서 자동 연결 서브스크립션 이해
- 고객 포털에서 수동 등록 및 서브스크립션 이해

2.2. 웹 콘솔에서 자격 증명을 사용하여 서브스크립션 등록

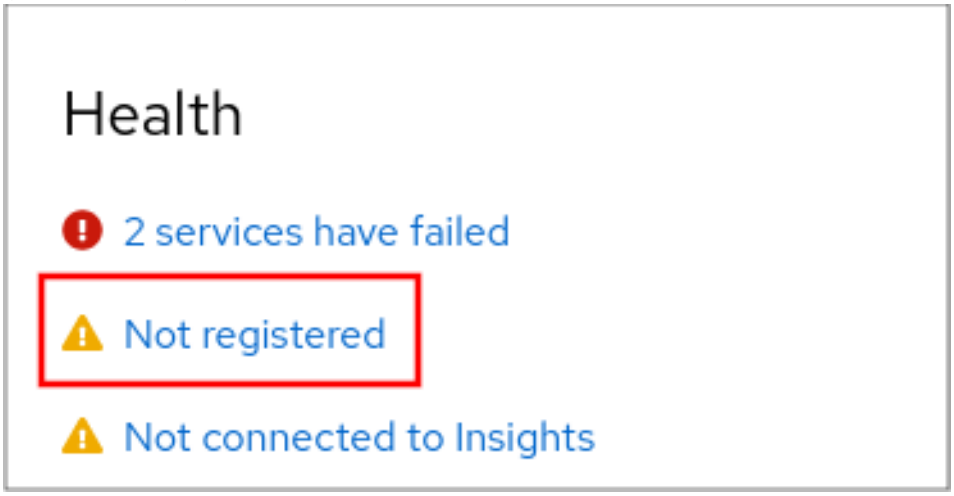
RHEL 웹 콘솔을 사용하여 계정 자격 증명에 새로 설치된 Red Hat Enterprise Linux를 등록하려면 다음 단계를 사용하십시오.

사전 요구 사항

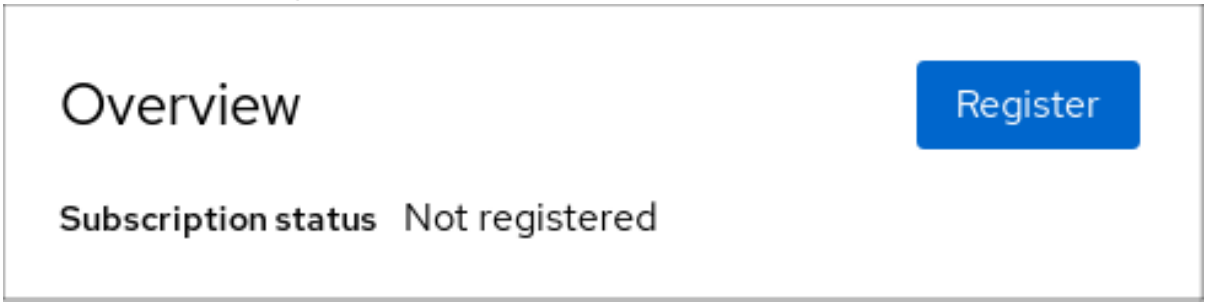
- Red Hat 고객 포털에서 유효한 사용자 계정.
  Red Hat 로그인 생성 페이지를 참조하십시오.
- RHEL 시스템용 활성 서브스크립션.

절차

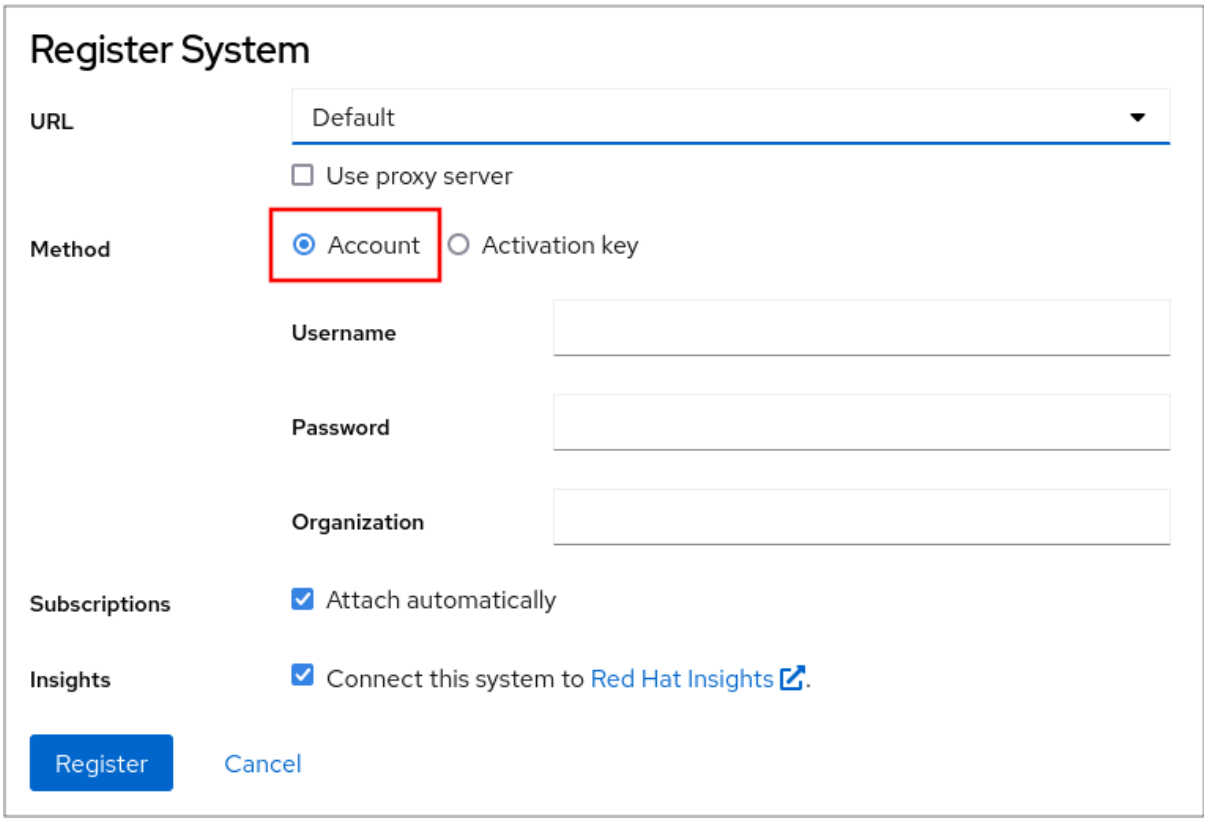
1. RHEL 웹 콘솔에 로그인합니다. 자세한 내용은 웹 콘솔에 로그인을 참조하십시오.
2. **개요** 페이지의 **Health** filed에서 **Not registered** 경고를 클릭하거나 메인 메뉴에서 **Subscriptions** (서브스크립션)를 클릭하여 서브스크립션 정보를 페이지로 이동합니다.



3. **Overview** filed에서 **Register** 를 클릭합니다.



4. 시스템 등록 대화 상자에서 계정 자격 증명을 사용하여 등록할 항목을 선택합니다.In the **Register system** dialog box, select that you want to register using your account credentials.



5. 사용자 이름을 입력합니다.

6. 암호를 입력합니다.

7. 필요한 경우 조직의 이름 또는 ID를 입력합니다.
Red Hat 고객 포털에서 두 개 이상의 조직에 속해 있는 경우 조직 이름 또는 조직 ID를 추가해야 합니다. org ID를 가져오려면 Red Hat 연락처로 이동하십시오.

   - 시스템을 Red Hat Insights에 연결하지 않으려면 **Insights** 확인란을 지웁니다.

8. **Register** 버튼을 클릭합니다.

이 시점에서 Red Hat Enterprise Linux Enterprise Linux 시스템이 성공적으로 등록되었습니다.

2.3. GNOME에 RED HAT 계정을 사용하여 시스템 등록

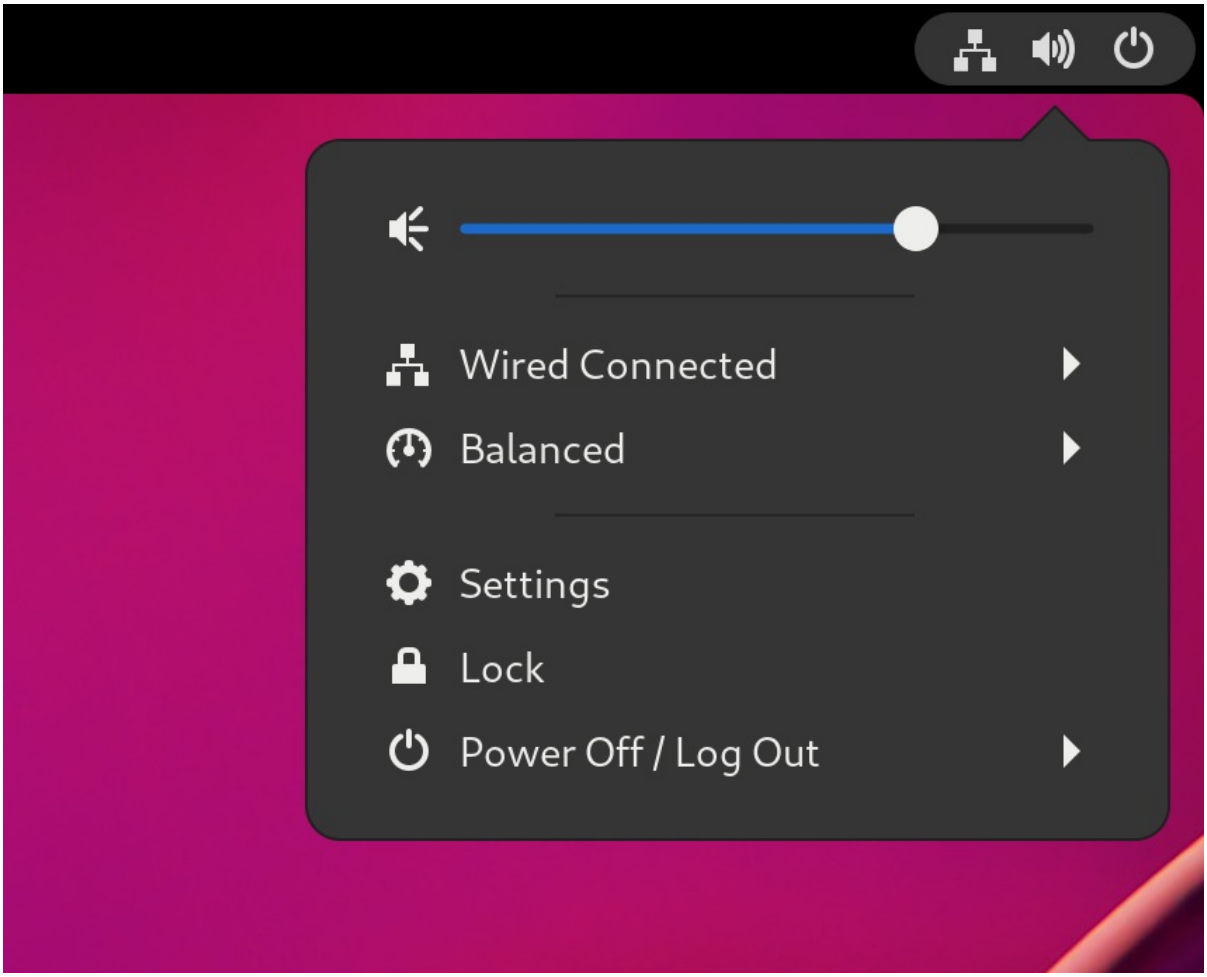
다음 절차에 따라 Red Hat 계정에 시스템을 등록합니다.

사전 요구 사항

- Red Hat 고객 포털에서 유효한 계정.
새 사용자 등록은 Create a Red Hat 로그인 페이지에서 참조하십시오.

절차

1. 오른쪽 상단에 있는 **시스템 메뉴**를 열고 **설정**을 클릭합니다.



2. **서브스크립션** 정보로 이동합니다.
3. Red Hat 서버를 사용하지 않는 경우:
   a. **등록 서버** 섹션에서 **사용자 지정 주소를 선택합니다**.
   b. **URL** 필드에 서버 주소를 입력합니다.
4. **등록 유형** 섹션에서 **Red Hat 계정을** 선택합니다.
5. **등록 세부 정보** 섹션에서 다음을 수행합니다.
   - **로그인** 필드에 Red Hat 계정 사용자 이름을 입력합니다.
   - **암호** 필드에 Red Hat 계정 암호를 입력합니다.
   - 조직 필드에 **조직** 이름을 입력합니다.
6. **등록을** 클릭합니다.

2.4. GNOME에서 활성화 키를 사용하여 시스템 등록

시스템을 활성화 키로 등록하려면 다음 절차의 단계를 수행합니다. 조직 관리자에서 활성화 키를 가져올 수 있습니다.

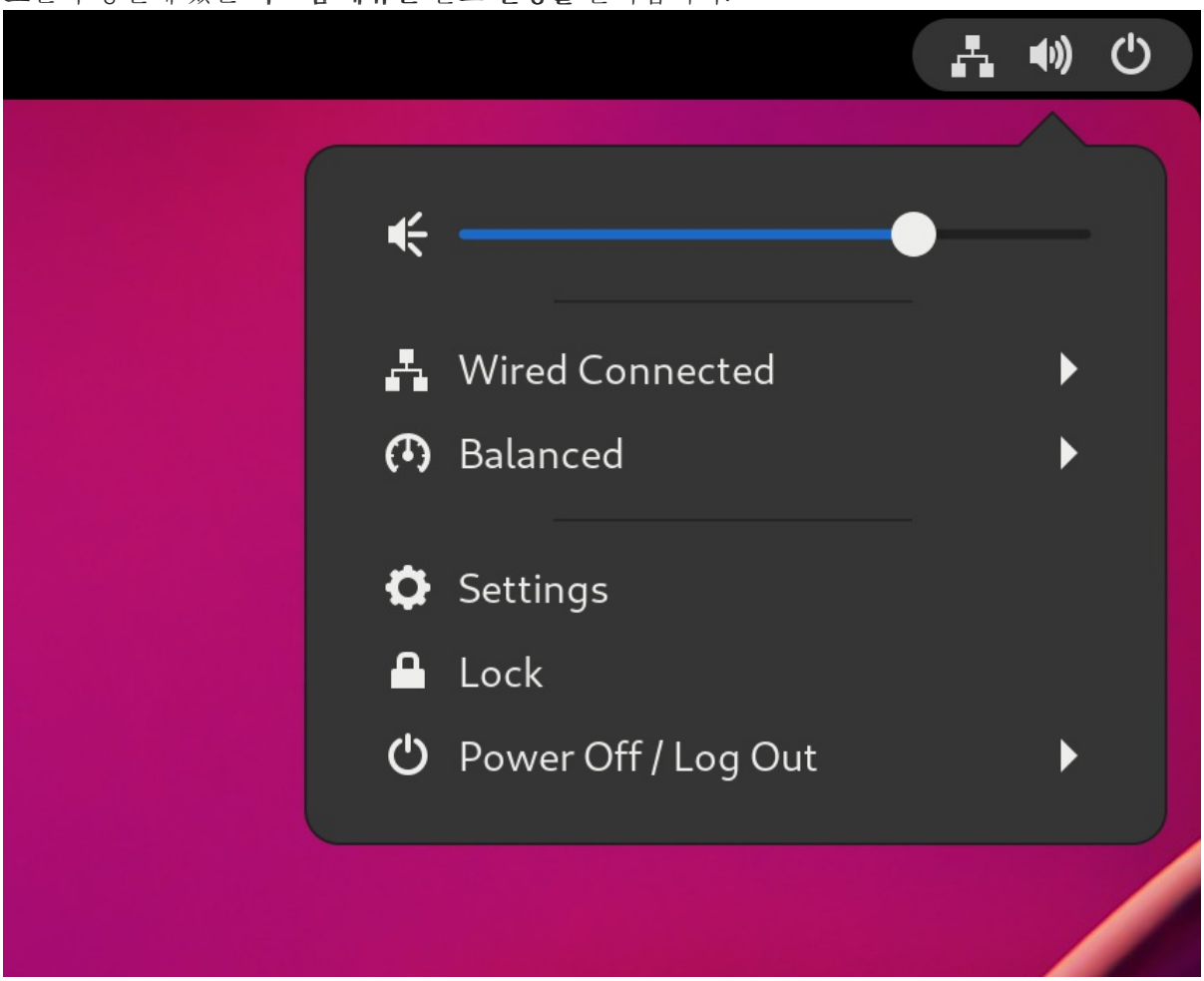
사전 요구 사항

- 활성화 키 또는 키.

새 활성화 키를 생성하려면 활성화 키 페이지를 참조하십시오.

절차

1. 오른쪽 상단에 있는 **시스템 메뉴**를 열고 **설정**을 클릭합니다.



2. **서브스크립션** 정보로 이동합니다.
3. Red Hat 서버를 사용하지 않는 경우:
   a. **등록 서버** 섹션에서 **사용자 지정 주소를 선택합니다**.
   b. **URL** 필드에 서버 주소를 입력합니다.
4. **등록 유형** 섹션에서 **활성화 키** 를 선택합니다.
5. **등록 세부 정보**:
   - 활성화 키 필드에 **활성화 키**를 입력합니다.
     키를 쉼표(,)로 구분합니다.
   - **조직** 필드에 조직의 이름 또는 ID를 입력합니다.
6. **등록**을 클릭합니다.

3장. RED HAT 지원 액세스

이 섹션에서는 Red Hat 지원 및 **sosreport** 를 사용하여 문제를 효과적으로 해결하는 방법을 설명합니다.

Red Hat의 지원을 받으려면 서브스크립션과 함께 제공되는 모든 항목에 액세스할 수 있는 Red Hat 고객 포털 을 사용하십시오.

3.1. RED HAT 고객 포털을 통해 RED HAT 지원 받기

다음 섹션에서는 Red Hat 고객 포털 사용 방법에 대해 설명합니다.

사전 요구 사항

- Red Hat 고객 포털에서 유효한 사용자 계정. Red Hat 로그인 생성을 참조하십시오.
- RHEL 시스템의 활성 서브스크립션입니다.

절차

1. Red Hat 지원에 액세스:
   a. 새 지원 케이스를 엽니다.
   b. Red Hat 전문가와의 실시간 채팅 시작.
   c. 이메일을 보내거나 전화하여 Red Hat 전문가에게 문의하십시오.

3.2. SOSREPORT를 사용하여 문제 해결

sosreport 명령은 Red Hat Enterprise Linux 시스템에서 구성 세부 정보, 시스템 정보 및 진단 정보를 수집합니다.

다음 섹션에서는 **sosreport** 명령을 사용하여 지원 사례에 대한 보고서를 생성하는 방법을 설명합니다.

사전 요구 사항

- Red Hat 고객 포털에서 유효한 사용자 계정. Red Hat 로그인 생성을 참조하십시오.
- RHEL 시스템의 활성 서브스크립션입니다.
- 지원 케이스 번호입니다.

절차

1. **sos** 패키지를 설치합니다.

```
# dnf install sos
```

참고

Red Hat Enterprise Linux의 기본 최소 설치에는 **sosreport** 명령을 제공하는 **sos** 패키지가 포함되어 있지 않습니다.

2. 보고서를 생성합니다.

```
# sosreport
```

3. 지원 케이스에 보고서를 첨부합니다.
   Red Hat 지원 케이스에 파일을 첨부하려면 어떻게 해야 합니까? 자세한 내용은 Red Hat Knowledgebase 문서.

   보고서를 첨부할 때 관련 지원 케이스 수를 입력하라는 메시지가 표시됩니다.

추가 리소스

- sosreport는 무엇이며 Red Hat Enterprise Linux에서 하나를 만드는 방법은 무엇입니까?

4장. 기본 환경 설정 변경

기본 환경 설정 구성은 설치 프로세스의 일부입니다. 다음 섹션에서는 나중에 변경할 때 안내합니다. 환경의 기본 구성은 다음과 같습니다.

- 날짜 및 시간
- 시스템 로케일
- 키보드 레이아웃
- 언어

4.1. 날짜 및 시간 구성

정확한 시간 보관은 여러 가지 이유로 중요합니다. Red Hat Enterprise Linux에서 시간 유지는 **NTP** 프로토콜에 의해 확인되며, 이는 사용자 공간에서 실행되는 데몬에 의해 구현됩니다. 사용자 공간 데몬은 커널에서 실행되는 시스템 클럭을 업데이트합니다. 시스템 클럭은 다양한 클럭 소스를 사용하여 시간을 유지할 수 있습니다.

Red Hat Enterprise Linux 9 이상 버전에서는 **chronyd** 데몬을 사용하여 **NTP** 를 구현합니다.**chronyd** 는 **chrony** 패키지에서 사용할 수 있습니다. 자세한 내용은 chrony 모음을 사용하여 NTP 구성을 참조하십시오.

4.1.1. 현재 날짜 및 시간 표시

현재 날짜 및 시간을 표시하려면 다음 단계 중 하나를 사용합니다.

절차

1. **date** 명령을 입력합니다.

```
$ date
Mon Mar 30 16:02:59 CEST 2020
```

2. 자세한 내용을 보려면 **timedatectl** 명령을 사용하십시오.

```
$ timedatectl
Local time: Mon 2020-03-30 16:04:42 CEST
Universal time: Mon 2020-03-30 14:04:42 UTC
  RTC time: Mon 2020-03-30 14:04:41
 Time zone: Europe/Prague (CEST, +0200)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

추가 리소스

- 웹 콘솔을 사용하여 시간 설정 구성
- **man date(1)** 및 **man timedatectl(1)**

4.2. 시스템 로케일 구성

시스템 전체 로케일 설정은 **systemd** 데몬에서 초기 부팅 시 읽을 수 있는 **/etc/locale.conf** 파일에 저장됩니다. 모든 서비스 또는 사용자는 개별 프로그램 또는 개별 사용자가 재정의하지 않는 한 **/etc/locale.conf** 에 구성된 로케일 설정을 상속합니다.

절차

- 사용 가능한 시스템 로케일 설정을 나열하려면 다음을 수행합니다.

```
$ localectl list-locales
C.utf8
aa_DJ
aa_DJ.iso88591
aa_DJ.utf8
...
```

- 시스템 로케일 설정의 현재 상태를 표시하려면 다음을 수행합니다.

```
$ localectl status
```

- 기본 시스템 로케일 설정을 설정하거나 변경하려면 **root** 사용자로 **localectl set-locale** 하위 명령을 사용합니다. 예를 들면 다음과 같습니다.

```
# localectl set-locale LANG=en_US
```

추가 리소스

- **man localectl(1), man locale(7)** 및 **man locale.conf(5)**

4.3. 키보드 레이아웃 구성

키보드 레이아웃 설정은 텍스트 콘솔 및 그래픽 사용자 인터페이스에서 사용되는 레이아웃을 제어합니다.

절차

- 사용 가능한 키맵을 나열하려면 다음을 수행합니다.

```
$ localectl list-keymaps
ANSI-dvorak
al
al-plisi
amiga-de
amiga-us
...
```

- keymaps 설정의 현재 상태를 표시하려면 다음을 수행합니다.

```
$ localectl status
...
VC Keymap: us
...
```

- 기본 시스템 키 맵을 설정하거나 변경하려면 다음을 수행합니다. 예를 들면 다음과 같습니다.

```
# localectl set-keymap us
```

추가 리소스

- **man localectl(1), man locale(7)** 및 **man locale.conf(5)**

4.4. 데스크탑 GUI를 사용하여 언어 변경

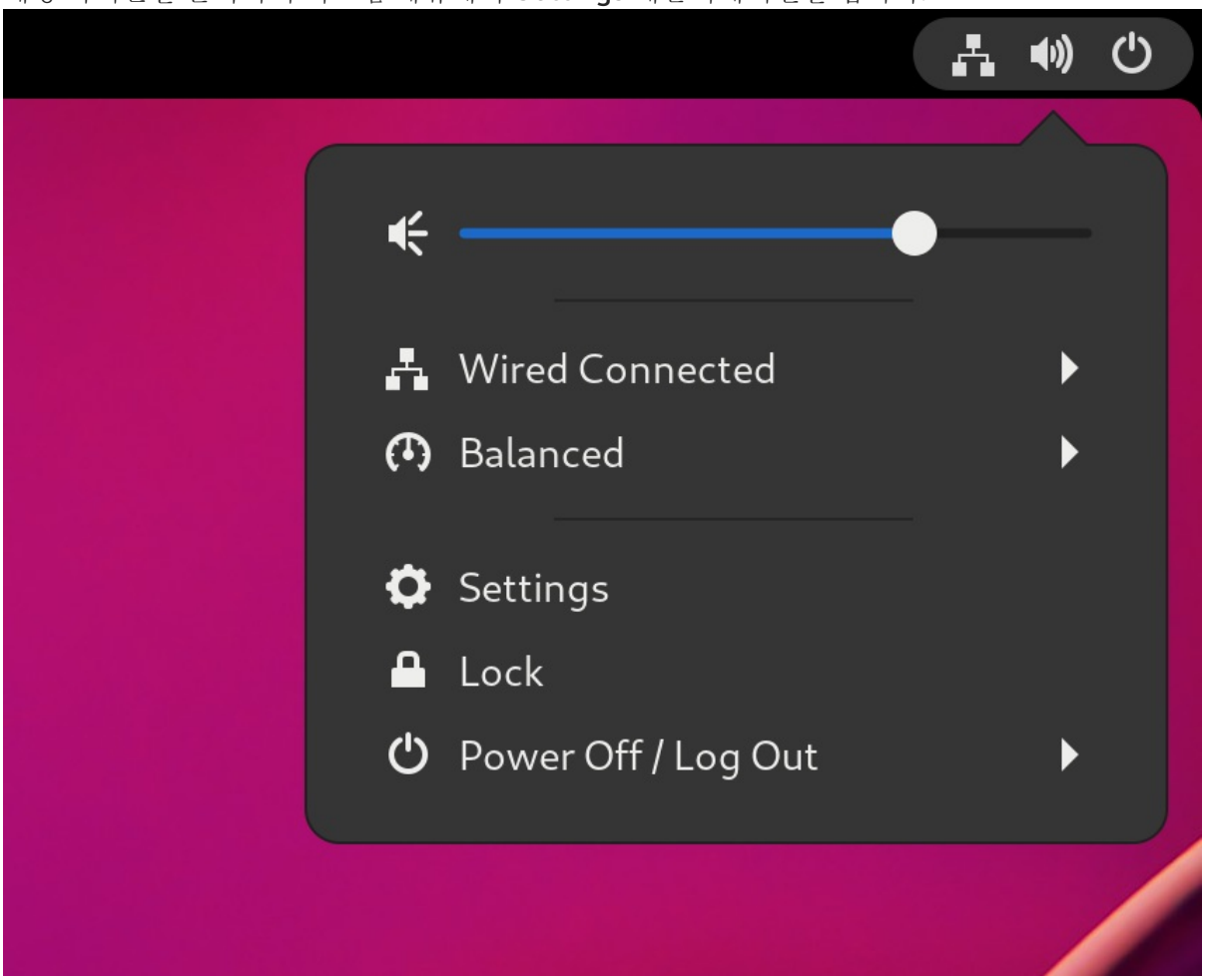
데스크탑 GUI를 사용하여 시스템 언어를 변경할 수 있습니다.

사전 요구 사항

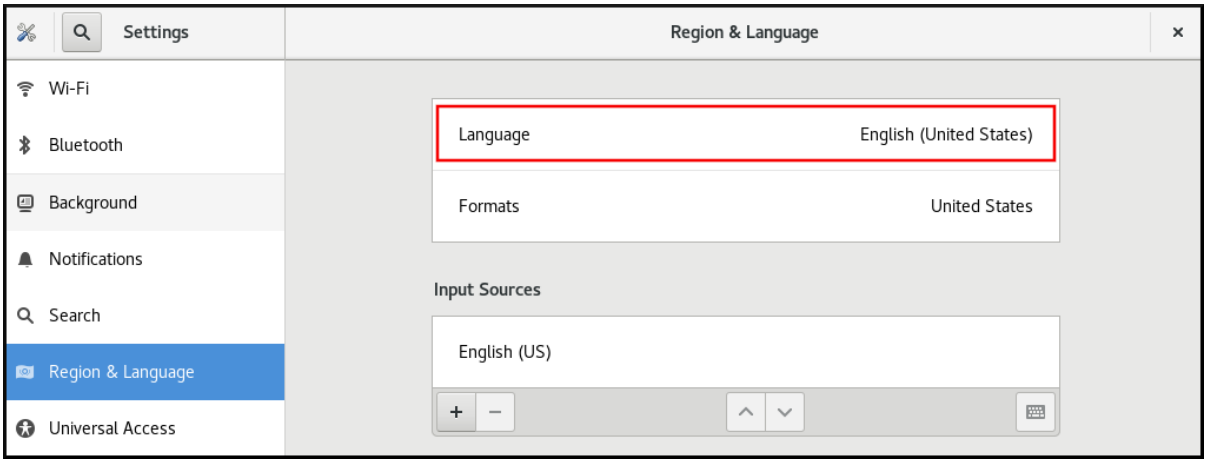
- 필수 언어 패키지가 시스템에 설치되어 있습니다.

절차

1. 해당 아이콘을 클릭하여 시스템 메뉴에서 **Settings** 애플리케이션을 엽니다.

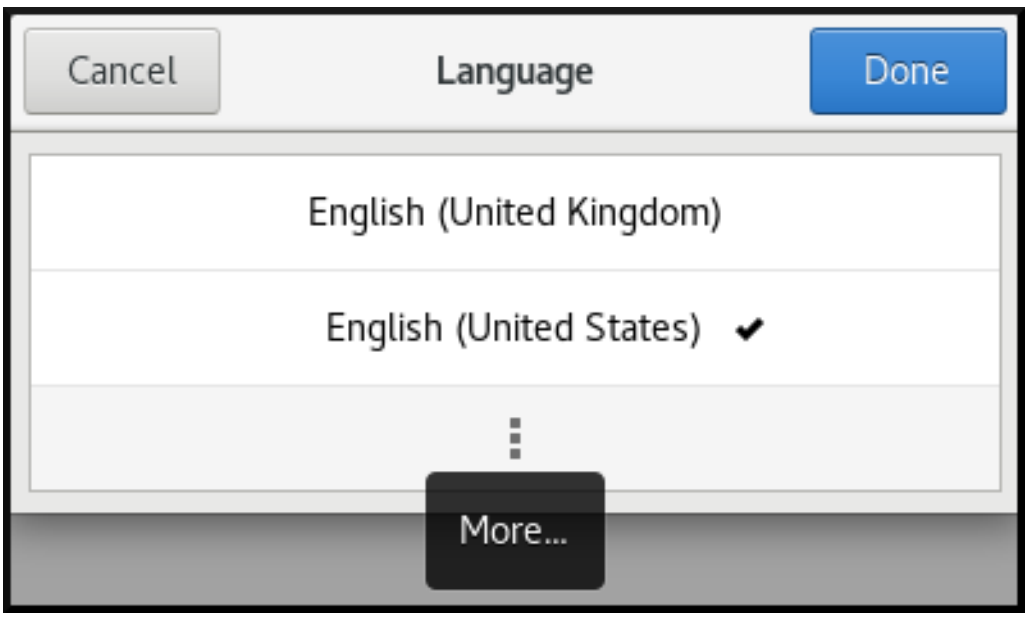


2. **설정 의 왼쪽 세로 막대에서 지역 및 언어**를 선택합니다.
3. **Language** 메뉴를 클릭합니다.



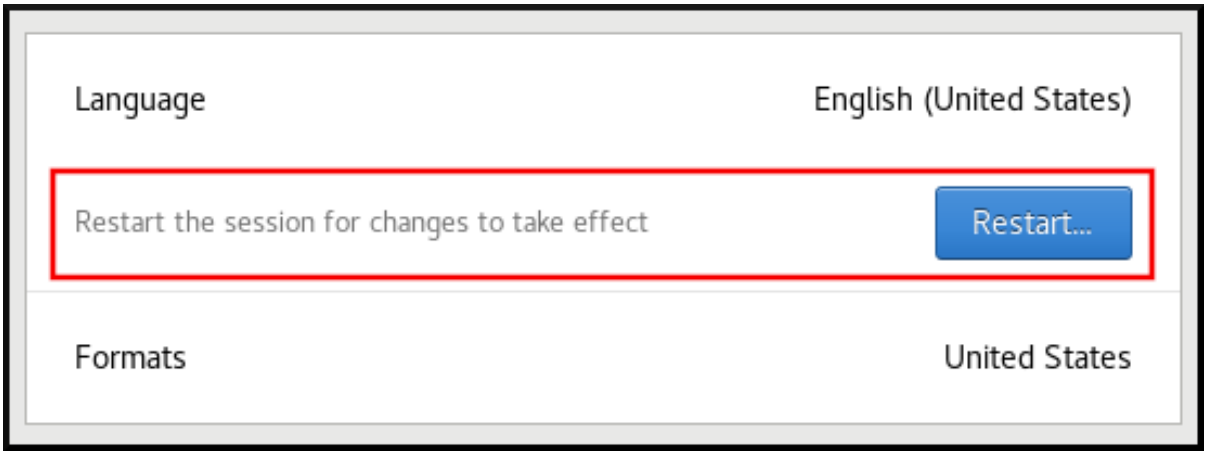
4. 메뉴에서 필요한 지역 및 언어를 선택합니다.

지역 및 언어가 나열되지 않은 경우 아래로 스크롤하여 **More** 를 클릭하여 사용 가능한 지역 및 언어를 선택합니다.



5. **Done(완료)을** 클릭합니다.

6. 변경 사항을 적용하려면 **Restart** 를 클릭합니다.



참고

일부 애플리케이션은 특정 언어를 지원하지 않습니다. 선택한 언어로 번역할 수 없는 애플리케이션의 텍스트는 미국 영어로 유지됩니다.

추가 리소스

- GNOME에서 애플리케이션 시작

4.5. 추가 리소스

- 표준 RHEL 9 설치 수행

5장. OPENSSH로 두 시스템 간의 보안 통신 사용

SSH(Secure Shell)는 클라이언트-서버 아키텍처를 사용하여 두 시스템 간에 보안 통신을 제공하고 사용자가 서버 호스트 시스템에 원격으로 로그인할 수 있는 프로토콜입니다. FTP 또는 Telnet과 같은 다른 원격 통신 프로토콜과 달리 SSH는 로그인 세션을 암호화하여 침입자가 연결에서 암호화되지 않은 암호를 수집하지 못하게 합니다.

Red Hat Enterprise Linux에는 일반 **openssh** 패키지, **openssh-server** 패키지, **openssh-clients** 패키지 등 기본 **OpenSSH** 패키지가 포함되어 있습니다. **OpenSSH** 패키지에는 **OpenSSH**가 암호화된 통신을 제공할 수 있는 몇 가지 중요한 암호화 라이브러리를 설치하는 **OpenSSL** 패키지 **openssl-libs**가 있어야 합니다.

5.1. SSH 및 OPENSSH

SSH(Secure Shell)는 원격 시스템에 로그인하고 해당 시스템에서 명령을 실행하는 프로그램입니다. SSH 프로토콜은 비보안 네트워크를 통해 신뢰할 수 없는 두 호스트 간에 안전한 암호화된 통신을 제공합니다. 보안 채널을 통해 X11 연결 및 임의의 TCP/IP 포트를 전달할 수도 있습니다.

SSH 프로토콜은 원격 쉘 로그인 또는 파일 복사에 사용할 때 두 시스템 간 통신 가로채기 및 특정 호스트의 가장과 같은 보안 위협을 완화합니다. 이는 SSH 클라이언트와 서버가 디지털 서명을 사용하여 신원을 확인하기 때문입니다. 또한 클라이언트와 서버 시스템 간의 모든 통신이 암호화됩니다.

호스트 키는 SSH 프로토콜에서 호스트를 인증합니다. 호스트 키는 OpenSSH가 처음 설치될 때 또는 호스트가 처음 부팅될 때 자동으로 생성되는 암호화 키입니다.

OpenSSH는 Linux, UNIX 및 유사한 운영 체제에서 지원하는 SSH 프로토콜 구현입니다. OpenSSH 클라이언트와 서버 모두에 필요한 코어 파일을 포함합니다. OpenSSH 제품군은 다음 사용자 공간 툴로 구성됩니다.

- **SSH** 는 원격 로그인 프로그램(SSH 클라이언트)입니다.
- **sshd** 는 OpenSSH SSH 데몬입니다.
- **SCP** 는 안전한 파일 복사 프로그램입니다.
- **SFTP** 는 안전한 파일 전송 프로그램입니다.
- **SSH-agent** 는 개인 키를 캐싱하기 위한 인증 에이전트입니다.
- **ssh-add** 는 **ssh-agent** 에 개인 키 ID를 추가합니다.
- **SSH-keygen은 ssh** 에 대한 인증 키를 생성, 관리 및 변환합니다.
- **ssh-copy-id** 는 원격 SSH 서버의 **authorized_keys** 파일에 로컬 공개 키를 추가하는 스크립트입니다.
- **SSH-keyscan** 은 SSH 공개 호스트 키를 수집합니다.

참고

RHEL 9에서는 기본적으로 SFTP(Secure File Transfer Protocol)가 SSH 파일 전송 프로토콜(SCP)으로 교체됩니다. 이는 SCP가 이미 보안 문제를 발생했기 때문입니다(예: CVE-2020-15778).

사용자의 시나리오에서 SFTP를 사용할 수 없거나 호환되지 않는 경우 **-O** 옵션을 사용하여 원래 SCP/RCP 프로토콜을 강제로 사용할 수 있습니다.

자세한 내용은 Red Hat Enterprise Linux 9에서 OpenSSH SCP 프로토콜 사용 중단 문서를 참조하십시오.

SSH에는 현재 버전 1과 최신 버전 2의 두 가지 버전이 있습니다. RHEL의 OpenSSH 제품군은 SSH 버전 2만 지원합니다. 버전 1에서 알려진 악용에 취약하지 않은 향상된 key-exchange 알고리즘을 가지고 있습니다.

OpenSSH는 RHEL의 핵심 암호화 하위 시스템 중 하나로 시스템 전체 암호화 정책을 사용합니다. 이렇게 하면 기본 구성에서 약한 암호 제품군 및 암호화 알고리즘이 비활성화됩니다. 정책을 수정하려면 관리자는 **update-crypto-policies** 명령을 사용하여 설정을 조정하거나 시스템 전체 암호화 정책을 수동으로 비활성화해야 합니다.

OpenSSH 제품군은 클라이언트 프로그램(즉, **ssh**,**scp**, **sftp**)과 서버(**sshd** 데몬)에 대한 두 가지 구성 파일 세트를 사용합니다.

시스템 전체 SSH 구성 정보는 **/etc/ssh/** 디렉토리에 저장됩니다. 사용자별 SSH 구성 정보는 사용자 홈 디렉터리의 **~/.ssh/**에 저장됩니다. OpenSSH 구성 파일의 자세한 목록은 **sshd(8)** 도움말 페이지의 **FILES** 섹션을 참조하십시오.

추가 리소스

- **man -k ssh** 명령을 사용하여 나열된 도움말 페이지
- 시스템 전체 암호화 정책 사용

5.2. OPENSSH 서버 구성 및 시작

환경 및 OpenSSH 서버를 시작하는 데 필요할 수 있는 기본 구성에는 다음 절차를 사용합니다. 기본 RHEL 설치 후에는 **sshd** 데몬이 이미 시작되고 서버 호스트 키가 자동으로 생성됩니다.

사전 요구 사항

- **openssh-server** 패키지가 설치되어 있어야 합니다.

절차

1. 현재 세션에서 **sshd** 데몬을 시작하고 부팅 시 자동으로 시작되도록 설정합니다.

```
# systemctl start sshd
# systemctl enable sshd
```

2. /etc/ssh/sshd_config 구성 파일의 ListenAddress 지시문의 경우 기본값 **0.0.0.0** (IPv4) 또는 **::** (IPv6) 이외의 다른 주소를 지정하고 느린 동적 네트워크 구성을 사용하려면 **network-online.target** 대상 장치에 대한 종속성을 **sshd.service** 장치 파일에 추가합니다. 이를 위해 다음 내용으로 사용하여 **/etc/systemd/system/sshd.service.d/local.conf** 파일을 만듭니다.

```
[Unit]
Wants=network-online.target
After=network-online.target
```

3. **/etc/ssh/sshd_config** 구성 파일의 OpenSSH 서버 설정이 시나리오 요구 사항을 충족하는지 검토합니다.

4. 선택적으로 **/etc/issue** 파일을 편집하여 클라이언트를 인증하기 전에 OpenSSH 서버가 표시하는 시작 메시지를 변경합니다. 예를 들면 다음과 같습니다.

```
Welcome to ssh-server.example.com
Warning: By accessing this server, you agree to the referenced terms and conditions.
```

Banner 옵션이 **/etc/ssh/sshd_config**에서 주석 처리되지 않고 해당 값에 **/etc/issue**가 포함되어 있는지 확인하십시오.

```
# less /etc/ssh/sshd_config | grep Banner
Banner /etc/issue
```

로그인에 성공한 후 표시되는 메시지를 변경하려면 서버에서 **/etc/motd** 파일을 편집해야 합니다. 자세한 내용은 **pam_motd** 도움말 페이지를 참조하십시오.

5. **systemd** 구성을 다시 로드하고 **sshd** 를 다시 시작하여 변경 사항을 적용합니다.

```
# systemctl daemon-reload
# systemctl restart sshd
```

검증

1. **sshd** 데몬이 실행 중인지 확인합니다.

```
# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2019-11-18 14:59:58 CET; 6min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1149 (sshd)
    Tasks: 1 (limit: 11491)
   Memory: 1.9M
   CGroup: /system.slice/sshd.service
           └─1149 /usr/sbin/sshd -D -oCiphers=aes128-ctr,aes256-ctr,aes128-cbc,aes256-cbc -
oMACs=hmac-sha2-256,>

Nov 18 14:59:58 ssh-server-example.com systemd[1]: Starting OpenSSH server daemon...
Nov 18 14:59:58 ssh-server-example.com sshd[1149]: Server listening on 0.0.0.0 port 22.
Nov 18 14:59:58 ssh-server-example.com sshd[1149]: Server listening on :: port 22.
Nov 18 14:59:58 ssh-server-example.com systemd[1]: Started OpenSSH server daemon.
```

2. SSH 클라이언트를 사용하여 SSH 서버에 연결합니다.

```
# ssh user@ssh-server-example.com
ECDSA key fingerprint is SHA256:dXbaS0RG/UzlTTku8GtXSz0S1++lPegSy31v3L/FAEc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
Warning: Permanently added 'ssh-server-example.com' (ECDSA) to the list of known hosts.

user@ssh-server-example.com's password:
```

추가 리소스

- **sshd(8)** 및 **sshd_config(5)** 도움말 페이지

5.3. 키 기반 인증을 위한 OPENSSH 서버 설정

시스템 보안을 강화하려면 OpenSSH 서버에서 암호 인증을 비활성화하여 키 기반 인증을 시행합니다.

사전 요구 사항

- **openssh-server** 패키지가 설치되어 있어야 합니다.
- **sshd** 데몬이 서버에서 실행되고 있어야 합니다.

절차

1. 텍스트 편집기에서 **/etc/ssh/sshd_config** 구성을 엽니다. 예를 들면 다음과 같습니다.

```
# vi /etc/ssh/sshd_config
```

2. **PasswordAuthentication** 옵션을 **no**로 변경합니다.

```
PasswordAuthentication no
```

새로운 기본 설치 이외의 시스템에서 **PubkeyAuthentication no** 가 설정되지 않았으며 **Kbd interactiveAuthentication** 지시문이 **no** 로 설정되어 있는지 확인합니다. 콘솔 또는 대역 외 액세스를 사용하지 않고 원격으로 연결하는 경우 암호 인증을 비활성화하기 전에 키 기반 로그인 프로세스를 테스트합니다.

3. NFS로 마운트된 홈 디렉토리에서 키 기반 인증을 사용하려면 **use_nfs_home_dirs** SELinux 부울을 활성화합니다.

```
# setsebool -P use_nfs_home_dirs 1
```

4. **sshd** 데몬을 다시 로드하여 변경 사항을 적용합니다.

```
# systemctl reload sshd
```

추가 리소스

- **sshd(8)**, **sshd_config(5)** 및 **setsebool(8)** 도움말 페이지

5.4. SSH 키 쌍 생성

이 절차를 사용하여 로컬 시스템에 SSH 키 쌍을 생성하고 생성된 공개 키를 OpenSSH 서버에 복사합니다. 서버가 적절하게 구성된 경우 암호를 제공하지 않고 OpenSSH 서버에 로그인할 수 있습니다.

중요

다음 단계를 **root**로 완료하면 **root**만 키를 사용할 수 있습니다.

절차

1. SSH 프로토콜의 버전 2에 대한 ECDSA 키 쌍을 생성하려면 다음을 수행합니다.

```
$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/joesec/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/joesec/.ssh/id_ecdsa.
Your public key has been saved in /home/joesec/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:Q/x+qms4j7PCQ0qFd09iZEFHA+SqwBKRNaU72oZfaCI
joesec@localhost.example.com
The key's randomart image is:
+---[ECDSA 256]---+
|.oo..o=++        |
|.. o .oo .       |
|. .. o. o        |
|....o.+...       |
|o.oo.o +S .      |
|.=.+.   .o       |
|E.*+.  .  . .    |
|.=..+ +..  o     |
|  .  oo*+o.      |
+----[SHA256]-----+
```

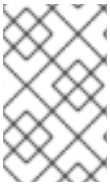
ssh-keygen -t ed25519 명령을 입력하여 **ssh-keygen** 명령 또는 Ed25519 키 쌍과 함께 **-t rsa** 옵션을 사용하여 RSA 키 쌍을 생성할 수도 있습니다.

2. 공개 키를 원격 머신에 복사하려면 다음을 수행합니다.

```
$ ssh-copy-id joesec@ssh-server-example.com
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are
already installed
joesec@ssh-server-example.com's password:
...
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'joesec@ssh-server-example.com'" and check to
make sure that only the key(s) you wanted were added.
```

세션에서 **ssh-agent** 프로그램을 사용하지 않는 경우 이전 명령은 가장 최근에 수정된 **~/.ssh/id*.pub** 공개 키가 아직 설치되지 않은 경우 공개 키를 복사합니다. 다른 공개 키 파일을 지정하거나 **ssh-agent**로 메모리에 캐시된 키보다 파일의 키 우선 순위를 지정하려면 **ssh-copy-id** 명령을 **-i** 옵션과 함께 사용합니다.



참고

시스템을 다시 설치하고 이전에 생성된 키 쌍을 유지하려면 **~/.ssh/** 디렉터리를 백업합니다. 다시 설치한 후 홈 디렉터리로 복사합니다. **root**를 포함하여 시스템의 모든 사용자에 대해 이 작업을 수행할 수 있습니다.

검증

1. 암호를 제공하지 않고 OpenSSH 서버에 로그인합니다.

```
$ ssh joesec@ssh-server-example.com
Welcome message.
...
Last login: Mon Nov 18 18:28:42 2019 from ::1
```

추가 리소스

- **ssh-keygen(1)** 및 **ssh-copy-id(1)** 도움말 페이지

5.5. 스마트 카드에 저장된 SSH 키 사용

Red Hat Enterprise Linux를 사용하면 OpenSSH 클라이언트의 스마트 카드에 저장된 RSA 및 ECDSA 키를 사용할 수 있습니다. 다음 절차에 따라 암호 대신 스마트 카드로 인증을 활성화합니다.

사전 요구 사항

- 클라이언트 측에서 **opensc** 패키지가 설치되고 **pcscd** 서비스가 실행 중입니다.

절차

1. PKCS #11 URI를 포함하여 OpenSC PKCS #11 모듈에서 제공하는 모든 키를 나열하고 출력을 *keys.pub* 파일에 저장합니다.

```
$ ssh-keygen -D pkcs11: > keys.pub
$ ssh-keygen -D pkcs11:
ssh-rsa AAAAB3NzaC1yc2E...KKZMzcQZzx
pkcs11:id=%02;object=SIGN%20pubkey;token=SSH%20key;manufacturer=piv_II?module-
path=/usr/lib64/pkcs11/opensc-pkcs11.so
ecdsa-sha2-nistp256 AAA...J0hkYnnsM=
pkcs11:id=%01;object=PIV%20AUTH%20pubkey;token=SSH%20key;manufacturer=piv_II?
module-path=/usr/lib64/pkcs11/opensc-pkcs11.so
```

2. 원격 서버(*example.com*)에서 스마트 카드를 사용하여 인증을 활성화하려면 공개 키를 원격 서버로 전송합니다. 이전 단계에서 만든 *keys.pub*와 함께 **ssh-copy-id** 명령을 사용하십시오.

```
$ ssh-copy-id -f -i keys.pub username@example.com
```

3. 1단계에서 **ssh-keygen -D** 명령의 출력에서 ECDSA 키를 사용하여 *example.com* 에 연결하려면 다음과 같이 키를 고유하게 참조하는 URI의 하위 집합만 사용할 수 있습니다.

```
$ ssh -i "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so" example.com
Enter PIN for 'SSH key':
[example.com] $
```

4. **~/.ssh/config** 파일에서 동일한 URI 문자열을 사용하여 구성을 영구적으로 만들 수 있습니다.

```
$ cat ~/.ssh/config
IdentityFile "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so"
$ ssh example.com
Enter PIN for 'SSH key':
[example.com] $
```

OpenSSH는 **p11-kit-proxy** 래퍼를 사용하고 OpenSC PKCS #11 모듈은 PKCS#11 Kit에 등록되므로 이전 명령을 간소화할 수 있습니다.

```
$ ssh -i "pkcs11:id=%01" example.com
Enter PIN for 'SSH key':
[example.com] $
```

PKCS #11 URI의 **id=** 부분을 건너뛰면 OpenSSH는 proxy 모듈에서 사용할 수 있는 모든 키를 로드합니다. 이렇게 하면 필요한 입력 횟수가 줄어듭니다.

```
$ ssh -i pkcs11: example.com
Enter PIN for 'SSH key':
[example.com] $
```

추가 리소스

- Fedora 28: OpenSSH에서 스마트 카드 지원 개선
- **p11-kit(8)**, **opensc.conf(5)**, **pcscd(8)**, **ssh(1)** 및 **ssh-keygen(1)** 매뉴얼 페이지

5.6. OPENSSH의 보안 강화

다음 팁은 OpenSSH를 사용할 때 보안을 강화하는 데 도움이 됩니다. **/etc/ssh/sshd_config** OpenSSH 구성 파일의 변경 사항을 적용하려면 **sshd** 데몬을 다시 로드해야 합니다.

```
# systemctl reload sshd
```

중요

대부분의 보안 강화 구성 변경으로 최신 알고리즘 또는 암호 제품군을 지원하지 않는 클라이언트와의 호환성이 줄어듭니다.

비보안 연결 프로토콜 비활성화

- SSH를 효과적으로 사용하려면 OpenSSH 제품군으로 대체되는 안전하지 않은 연결 프로토콜을 사용하지 않도록 합니다. 그렇지 않으면 Telnet을 사용하여 로그인할 때 나중에 하나의 세션이 캡처되도록 SSH를 사용하여 사용자 암호를 보호할 수 있습니다. 이러한 이유로 telnet, rsh, rlogin 및 ftp와 같은 비보안 프로토콜을 비활성화하는 것이 좋습니다.

키 기반 인증 활성화 및 암호 기반 인증 비활성화

- 인증에 대한 암호 비활성화 및 키 쌍만 허용하면 공격 면적이 줄어들고 사용자의 시간도 절약할 수 있습니다. 클라이언트에서 **ssh-keygen** 툴을 사용하여 키 쌍을 생성하고 **ssh-copy-id** 유틸리티를 사용하여 OpenSSH 서버의 클라이언트에서 공개 키를 복사합니다. OpenSSH 서버에서 암

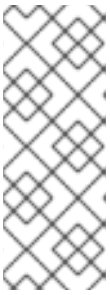
호 기반 인증을 비활성화하려면 **/etc/ssh/sshd_config**를 편집하고 **PasswordAuthentication** 옵션을 **no**로 변경합니다.

```
PasswordAuthentication no
```

키 유형

- **ssh-keygen** 명령은 기본적으로 RSA 키 쌍을 생성하지만 **-t** 옵션을 사용하여 ECDSA 또는 Ed25519 키를 생성하도록 지시할 수 있습니다. ECDSA(Elliptic Curve Digital Signature Algorithm)는 동등한 대칭 키 힘에서 RSA보다 더 나은 성능을 제공합니다. 또한 더 짧은 키를 생성합니다. Ed25519 공개 키 알고리즘은 RSA, DSA 및 ECDSA보다 더 빠르고 안전하며 더 빠릅니다.
  OpenSSH는 RSA, ECDSA 및 Ed25519 서버 호스트 키가 누락된 경우 자동으로 생성합니다. RHEL에서 호스트 키 생성을 구성하려면 **sshd-keygen@.service** 인스턴스화 서비스를 사용합니다. 예를 들어, RSA 키 유형의 자동 생성을 비활성화하려면 다음을 실행합니다.

```
# systemctl mask sshd-keygen@rsa.service
```



참고

cloud-init 가 활성화된 이미지에서 **ssh-keygen** 단위가 자동으로 비활성화됩니다. 이는 **ssh-keygen 템플릿** 서비스가 **cloud-init** 툴을 방해하고 호스트 키 생성에 문제가 발생할 수 있기 때문입니다. 이러한 문제를 방지하려면 **cloud-init** 가 실행 중인 경우 **etc/systemd/system/sshd-keygen@.service.d/disable-sshd-keygen-if-cloud-init-active.conf** 드롭인 구성 파일에서 **ssh-keygen** 장치를 비활성화합니다.

- SSH 연결에 대한 특정 키 유형을 제외하려면 **/etc/ssh/sshd_config**에서 관련 행을 주석 처리하고 **sshd** 서비스를 다시 로드합니다. 예를 들어 Ed25519 호스트 키만 허용하려면 다음을 수행합니다.

```
# HostKey /etc/ssh/ssh_host_rsa_key
# HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
```

중요

Ed25519 알고리즘은 FIPS-140과 호환되지 않으며 OpenSSH는 FIPS 모드에서 Ed25519 키와 작동하지 않습니다.

기본값 이외의 포트

- 기본적으로 **sshd** 데몬은 TCP 포트 22에서 수신 대기합니다. 포트를 변경하면 자동화된 네트워크 스캔을 기반으로 시스템이 공격에 노출되는 것을 줄이며 비만에 의해 보안이 향상됩니다.
  /etc/ssh/sshd_config 구성 파일에서 **Port** 지시문을 사용하여 포트를 지정할 수 있습니다.
  또한 기본이 아닌 포트를 사용할 수 있도록 기본 SELinux 정책을 업데이트해야 합니다. 이렇게 하려면 **policycoreutils-python-utils** 패키지에서 **semanage** 툴을 사용합니다.

```
# semanage port -a -t ssh_port_t -p tcp <port_number>
```

또한 **firewalld** 구성을 업데이트합니다.

```
# firewall-cmd --add-port <port_number>/tcp
# firewall-cmd --remove-port=22/tcp
# firewall-cmd --runtime-to-permanent
```

이전 명령에서 < *port_number*> 를 **Port** 지시문을 사용하여 지정된 새 포트 번호로 바꿉니다.

Root 로그인

- **PermitRootLogin**은 기본적으로 **prohibit-password** 로 설정됩니다. 이렇게 하면 root로 로그인하는 데 암호를 사용하는 대신 키 기반 인증을 사용하고 무차별 강제 공격을 방지하여 위험을 줄입니다.

주의

관리자가 권한이 있는 명령을 실행하는 사용자를 감사할 수 없기 때문에 root 사용자로 로그인 활성화는 안전한 방법이 아닙니다. 관리 명령을 사용하려면 로그인하고 **sudo**를 대신 사용합니다.

X 보안 확장 사용

- Red Hat Enterprise Linux 클라이언트의 X 서버는 X 보안 확장을 제공하지 않습니다. 따라서 클라이언트는 X11 전달을 사용하여 신뢰할 수 없는 SSH 서버에 연결할 때 다른 보안 계층을 요청할 수 없습니다. 대부분의 애플리케이션은 이 확장 기능을 사용하여 실행할 수 없습니다.
기본적으로 **/etc/ssh/ssh_config.d/50-redhat.conf** 파일의 **ForwardX11Trusted** 옵션은 **yes** 로 설정되고 **ssh -X remote_machine** (신뢰할 수 없는 호스트)과 **ssh -Y remote_machine** (신뢰할 수 있는 호스트) 명령 사이에 차이가 없습니다.

  시나리오에 X11 전달 기능이 전혀 필요하지 않은 경우 **/etc/ssh/sshd_config** 구성 파일의 **X11Forwarding** 지시문을 **no**로 설정합니다.

특정 사용자, 그룹 또는 도메인에 대한 액세스 제한

- **/etc/ssh/sshd_config** 구성 파일 서버의 **AllowUsers** 및 **AllowGroups** 지시문을 사용하면 특정 사용자, 도메인 또는 그룹만 OpenSSH 서버에 연결할 수 있습니다. **AllowUsers** 및 **AllowGroups**를 결합하여 보다 정확하게 액세스를 제한할 수 있습니다. 예를 들면 다음과 같습니다.

```
AllowUsers *@192.168.1.* *@10.0.0.* !*@192.168.1.2
AllowGroups example-group
```

  이전 구성 행은 192.168.1.2 주소가 있는 시스템을 제외하고 192.168.1.* 및 10.0.0.* 서브넷의 모든 사용자로부터의 연결을 허용합니다. 모든 사용자는 **example-group** 그룹에 있어야 합니다. OpenSSH 서버는 다른 모든 연결을 거부합니다.

  OpenSSH 서버는 **/etc/ssh/sshd_config** 의 모든 Allow 및 Deny 지시문을 전달하는 연결만 허용합니다. 예를 들어 **AllowUsers** 지시문이 **AllowGroups** 지시문에 나열된 그룹에 포함되지 않은 사용자를 나열하는 경우 사용자는 로그인할 수 없습니다.

허용 목록(허용으로 시작하는 디렉터리)을 사용하는 것은 차단 목록(거부로 시작하는 옵션)을 사용하는 것보다 더 안전합니다. allowlists는 새로운 권한 없는 사용자 또는 그룹도 차단하기 때문입니다.

시스템 전체 암호화 정책 변경

- OpenSSH는 RHEL 시스템 전체 암호화 정책을 사용하며 기본 시스템 전체 암호화 정책 수준은 현재 위협 모델에 대한 보안 설정을 제공합니다. 암호화 설정을 보다 엄격하게 수행하려면 현재 정책 수준을 변경합니다.

```
# update-crypto-policies --set FUTURE
Setting system policy to FUTURE
```

주의

시스템이 인터넷에서 통신할 경우 **FUTURE** 정책의 엄격한 설정으로 인해 상호 운용성 문제에 직면할 수 있습니다.

시스템 전체 암호화 정책을 통해 SSH 프로토콜의 특정 암호만 비활성화할 수도 있습니다. 자세한 내용은 보안 강화 문서의 하위 정책을 사용하여 시스템 전체 암호화 정책 사용자 지정 섹션을 참조하십시오.

OpenSSH 서버에 대한 시스템 전체 암호화 정책을 비활성화하려면 /etc/ssh/sshd_config.d/ 디렉터리에 있는 드롭인 구성 파일에 암호화 정책을 지정하려면 **/etc/ssh/sshd_config.d/** 디렉터리에 있는 **암호화 정책을 지정하여 두 자리 숫자 접두사가 50-redhat.conf 파일 앞에 있고 .conf 접미사를 .conf** 접미사로 지정합니다.

자세한 내용은 **sshd_config(5)** 도움말 페이지를 참조하십시오.

OpenSSH 클라이언트에 대한 시스템 전체 암호화 정책을 비활성화하려면 다음 작업 중 하나를 수행합니다.

- 지정된 사용자의 경우 **~/.ssh/ config 파일의 사용자별 구성으로 글로벌 ssh_ config** 를 재정의합니다.
- 전체 시스템의 경우 **/etc/ssh/ssh_config.d/** 디렉터리에 있는 드롭인 구성 파일에 암호화 정책을 지정하고 두 자리 숫자 접두사가 **50-redhat.conf** 파일 앞에 있고 **.conf** 접미사(예: **49-crypto-policy-override.conf**)를 사용합니다.

추가 리소스

- **sshd_config(5)**, **ssh-keygen(1)**, **crypto-policies(7)** 및 **update-crypto-policies(8)** 도움말 페이지
- 보안 강화 문서에서 시스템 전체 암호화 정책 사용 .
- ssh 서비스에 대한 특정 알고리즘 및 암호를 비활성화하는 방법만 문서입니다.

5.7. SSH 건너뛰기 호스트를 사용하여 원격 서버에 연결

건너뛰기 호스트라고도 하는 중간 서버를 통해 로컬 시스템을 원격 서버에 연결하려면 다음 절차를 사용하십시오.

사전 요구 사항

- 건너뛰기 호스트는 로컬 시스템의 SSH 연결을 허용합니다.
- 원격 서버는 건너뛰기 호스트에서만 SSH 연결을 허용합니다.

절차

1. 로컬 시스템에서 **~/.ssh/config** 파일을 편집하여 건너뛰기를 정의합니다. 예를 들면 다음과 같습니다.

```
Host jump-server1
  HostName jump1.example.com
```

   - **Host** 매개 변수는 **ssh** 명령에서 사용할 수 있는 호스트의 이름 또는 별칭을 정의합니다. 이 값은 실제 호스트 이름과 일치할 수 있지만 임의의 문자열일 수도 있습니다.
   - **HostName** 매개 변수는 건너뛰기 호스트의 실제 호스트 이름 또는 IP 주소를 설정합니다.

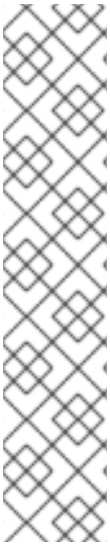
2. **ProxyJump** 지시문을 사용하여 원격 서버 건너뛰기를 로컬 시스템의 **~/.ssh/config** 파일에 추가합니다. 예를 들면 다음과 같습니다.

```
Host remote-server
  HostName remote1.example.com
  ProxyJump jump-server1
```

3. 로컬 시스템을 사용하여 이동 서버를 통해 원격 서버에 연결합니다.

```
$ ssh remote-server
```

   이전 명령은 구성 단계 1과 2를 생략하면 **ssh -J jump-server1 remote-server** 명령과 동일합니다.



참고

더 많은 건너뛰기 서버를 지정할 수 있으며 전체 호스트 이름을 제공할 때 구성 파일에 호스트 정의 추가를 건너뛸 수도 있습니다. 예를 들면 다음과 같습니다.

```
$ ssh -J jump1.example.com,jump2.example.com,jump3.example.com
remote1.example.com
```

건너뛰기 서버의 사용자 이름 또는 SSH 포트가 원격 서버의 이름과 포트와 다른 경우 이전 명령에서 호스트 이름 전용 표기법을 변경합니다. 예를 들면 다음과 같습니다.

```
$ ssh -J
johndoe@jump1.example.com:75,johndoe@jump2.example.com:75,johndoe@jump3.example.com:75 joesec@remote1.example.com:220
```

추가 리소스

- **ssh_config(5)** 및 **ssh(1)** 도움말 페이지

5.8. SSH-AGENT를 사용하여 SSH 키가 있는 원격 시스템에 연결

SSH 연결을 시작할 때마다 암호를 입력하지 않으려면 **ssh-agent** 유틸리티를 사용하여 개인 SSH 키를 캐시할 수 있습니다. 개인 키와 암호는 안전하게 유지됩니다.

사전 요구 사항

- SSH 데몬이 실행되고 네트워크를 통해 연결할 수 있는 원격 호스트가 있습니다.
- IP 주소 또는 호스트 이름 및 인증 정보를 통해 원격 호스트에 로그인합니다.
- 암호를 사용하여 SSH 키 쌍을 생성하고 공개 키를 원격 시스템으로 전송했습니다.

자세한 내용은 SSH 키 쌍 생성을 참조하십시오.

절차

1. 선택 사항: 키를 사용하여 원격 호스트에 인증할 수 있는지 확인합니다.

   a. SSH를 사용하여 원격 호스트에 연결합니다.

   ```
   $ ssh example.user1@198.51.100.1 hostname
   ```

   b. 개인 키에 대한 액세스 권한을 부여할 키를 만드는 동안 설정한 암호를 입력합니다.

   ```
   $ ssh example.user1@198.51.100.1 hostname
    host.example.com
   ```

2. **ssh-agent**를 시작합니다.

   ```
   $ eval $(ssh-agent)
   Agent pid 20062
   ```

3. **ssh-agent**에 키를 추가합니다.

   ```
   $ ssh-add ~/.ssh/id_rsa
   Enter passphrase for ~/.ssh/id_rsa:
   Identity added: ~/.ssh/id_rsa (example.user0@198.51.100.12)
   ```

검증

- 선택 사항: SSH를 사용하여 호스트 시스템에 로그인합니다.

  ```
  $ ssh example.user1@198.51.100.1

  Last login: Mon Sep 14 12:56:37 2020
  ```

  암호를 입력할 필요가 없습니다.

5.9. 추가 리소스

- **sshd(8)**, **ssh(1)**, **scp(1)**, **sftp(1)**, **ssh-keygen(1)**, **ssh-copy-id(1)**, **ssh_config(5)**, **sshd_config(5)**, **update-crypto-policies(8)** 및 **crypto-policies(7)** 도움말 페이지
- OpenSSH Home Page
- 비표준 구성을 사용하여 애플리케이션 및 서비스에 대한 SELinux 구성
- firewalld를 사용하여 네트워크 트래픽 제어

6장. 기본 시스템 보안 구성

컴퓨터 보안은 컴퓨터 시스템과 하드웨어, 소프트웨어, 정보, 서비스를 도용, 손상, 중단 및 잘못으로부터 보호하는 것입니다. 특히 중요한 데이터를 처리하고 비즈니스 트랜잭션을 처리하는 기업이 컴퓨터 보안을 보장하는 것이 중요합니다.

이 섹션에서는 운영 체제 설치 후 구성할 수 있는 기본 보안 기능만 다룹니다.

6.1. FIREWALLD 서비스 활성화

방화벽은 구성된 보안 규칙에 따라 들어오고 나가는 네트워크 트래픽을 모니터링하고 제어하는 네트워크 보안 시스템입니다. 방화벽은 일반적으로 신뢰할 수 있는 보안 내부 네트워크와 다른 외부 네트워크 간의 장벽을 설정합니다.

Red Hat Enterprise Linux에서 방화벽을 제공하는 **firewalld** 서비스는 설치 중에 자동으로 활성화됩니다.

firewalld 서비스를 활성화하려면 다음 절차를 따르십시오.

절차

- **firewalld** 의 현재 상태 표시:

```
$ systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset:
enabled)
   Active: inactive (dead)
...
```

- **firewalld** 가 활성화되어 실행되지 않으면 **root** 사용자로 전환하고 **firewalld** 서비스를 시작하고 시스템을 다시 시작한 후 자동으로 시작합니다.

```
# systemctl enable --now firewalld
```

검증 단계

- **firewalld** 가 실행 중이고 활성화되어 있는지 확인합니다.

```
$ systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset:
enabled)
   Active: active (running)
...
```

추가 리소스

- firewalld 사용 및 구성
- **man firewalld(1)**

6.2. 기본 SELINUX 설정 관리

SELinux(Security-hardened Linux)는 어떤 프로세스가 어떤 파일, 디렉터리 및 포트에 액세스할 수 있는지 결정하는 추가 시스템 보안 계층입니다. 이러한 권한은 SELinux 정책에 정의되어 있습니다. 정책은 SELinux 보안 엔진을 안내하는 일련의 규칙입니다.

SELinux에는 다음 두 가지 상태가 있습니다.

- disabled
- enabled

SELinux가 활성화되면 다음 모드 중 하나에서 실행됩니다.

- enabled
  - enforcing
  - 허용

강제 모드에서 SELinux는 로드된 정책을 적용합니다. SELinux는 SELinux 정책 규칙에 따라 액세스를 거부하고 명시적으로 허용되는 상호 작용만 활성화합니다. 강제 모드는 가장 안전한 SELinux 모드이며 설치 후 기본 모드입니다.

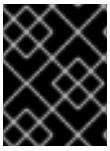
허용 모드에서 SELinux는 로드된 정책을 적용하지 않습니다. SELinux는 액세스를 거부하지 않지만 규칙을 중단하는 작업을 **/var/log/audit/audit.log** 로그에 보고합니다. 허용 모드는 설치 중에 기본 모드입니다. Permissive 모드는 예를 들어 문제 해결과 같은 특정 경우에 유용합니다.

추가 리소스

- SELinux 사용

6.3. SELINUX의 필수 상태 확인

기본적으로 SELinux는 강제 모드로 작동합니다. 그러나 특정 시나리오에서는 SELinux를 허용 모드로 설정하거나 비활성화할 수도 있습니다.



중요

Red Hat은 시스템을 강제 모드로 유지할 것을 권장합니다. 디버깅 목적으로 SELinux를 허용 모드로 설정할 수 있습니다.

시스템에서 SELinux의 상태 및 모드를 변경하려면 다음 절차를 따르십시오.

절차

1. 현재 SELinux 모드를 표시합니다.

```
$ getenforce
```

2. SELinux를 일시적으로 설정하려면 다음을 수행합니다.
   a. 강제 모드로 전환하려면 다음을 수행합니다.

```
# setenforce Enforcing
```

b. 허용 모드의 경우 다음을 수행합니다.

```
# setenforce Permissive
```

참고

재부팅 후 SELinux 모드는 **/etc/selinux/config** 구성 파일에 지정된 값으로 설정됩니다.

3. 재부팅 시 지속되도록 SELinux 모드를 설정하려면 **/etc/selinux/config** 구성 파일에서 **SELINUX** 변수를 수정합니다.
예를 들어 SELinux를 강제 모드로 전환하려면 다음을 수행합니다.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
...
```

주의

SELinux를 비활성화하면 시스템 보안이 줄어듭니다. 메모리 누수 및 경쟁 조건으로 인해 커널 패닉이 발생할 수 있으므로 **/etc/selinux/config** 파일에서 **SELINUX=disabled** 옵션을 사용하여 SELinux를 비활성화하지 마십시오. 대신 커널 명령줄에 **selinux=0** 매개 변수를 추가하여 SELinux를 비활성화합니다. 자세한 내용은 부팅 시 SELinux 모드 변경을 참조하십시오.

추가 리소스

- SELinux 상태 및 모드 변경

6.4. 추가 리소스

- SSH 키 쌍 생성
- 키 기반 인증을 위한 OpenSSH 서버 설정
- 보안 강화
- SELinux 사용
- 네트워크 보안

7장. RHEL 시스템 역할 소개

RHEL 시스템 역할을 사용하여 여러 시스템에서 시스템 관리를 자동화할 수 있습니다.

RHEL 시스템 역할은 Ansible 역할 및 모듈의 컬렉션입니다. RHEL 시스템 역할을 사용하면 RHEL의 주요 버전에서 여러 RHEL 시스템의 시스템 구성을 원격으로 관리할 수 있습니다. 시스템을 구성하는 데 사용하려면 다음 구성 요소를 사용해야 합니다.

제어 노드

제어 노드는 Ansible 명령 및 플레이북을 실행하는 시스템입니다. 제어 노드는 Ansible Automation Platform, Red Hat Satellite 또는 RHEL 9, 8 또는 7 호스트일 수 있습니다. 자세한 내용은 RHEL 9에서 제어 노드 준비를 참조하십시오.

관리형 노드

관리형 노드는 Ansible로 관리하는 서버 및 네트워크 장치입니다. 관리 노드를 호스트라고도 합니다. Ansible을 관리형 노드에 설치할 필요가 없습니다. 자세한 내용은 관리 노드 준비를 참조하십시오.

Ansible 플레이북

플레이북에서는 관리형 노드에서 수행할 구성을 정의하거나 수행할 관리 노드의 시스템의 단계 집합을 정의합니다. 플레이북은 Ansible의 구성, 배포 및 오케스트레이션 언어입니다.

인벤토리

인벤토리 파일에서 관리 노드를 나열하고 각 관리 노드의 IP 주소와 같은 정보를 지정합니다. 인벤토리에서는 더 쉽게 스케일링할 수 있도록 그룹을 생성하고 중첩하여 관리 노드를 구성할 수도 있습니다. 인벤토리 파일을 호스트 파일이라고도 합니다.

Red Hat Enterprise Linux 9에서는 **AppStream** 리포지토리에서 사용할 수 있는 **rhel-system-roles** 패키지에서 제공하는 다음 역할을 사용할 수 있습니다.

역할 이름	역할 설명	장 제목
certificate	인증서 발행 및 갱신	RHEL 시스템 역할을 사용하여 인증서 요청
cockpit	웹 콘솔	cockpit RHEL 시스템 역할을 사용하여 웹 콘솔 설치 및 구성
crypto_policies	시스템 전체 암호화 정책	시스템 전체에서 사용자 정의 암호화 정책 설정
방화벽	firewalld	시스템 역할을 사용하여 firewalld 구성
ha_cluster	HA 클러스터	시스템 역할을 사용하여 고가용성 클러스터 구성
kdump	커널 덤프	RHEL 시스템 역할을 사용하여 kdump 구성
kernel_settings	커널 설정	Ansible 역할을 사용하여 커널 매개변수를 영구적으로 구성
logging	로깅	로깅 시스템 역할 사용
메트릭	메트릭 (PCP)	RHEL 시스템 역할을 사용하여 성능 모니터링

역할 이름	역할 설명	장 제목
network	네트워킹	네트워크 RHEL 시스템 역할을 사용하여 InfiniBand 연결 관리
nbde_client	네트워크 Bound 디스크 암호화 클라이언트	nbde_client 및 nbde_server 시스템 역할 사용
nbde_server	네트워크 Bound 디스크 암호화 서버	nbde_client 및 nbde_server 시스템 역할 사용
postfix	Postfix	시스템 역할에서 postfix 역할의 변수
postgresql	PostgreSQL	postgresql RHEL 시스템 역할을 사용하여 PostgreSQL 설치 및 구성
selinux	SELinux	시스템 역할을 사용하여 SELinux 구성
ssh	SSH 클라이언트	ssh 시스템 역할을 사용하여 보안 통신 구성
sshd	SSH 서버	ssh 시스템 역할을 사용하여 보안 통신 구성
storage	스토리지	RHEL 시스템 역할을 사용하여 로컬 스토리지 관리
tlog	터미널 세션 레코딩	tlog RHEL 시스템 역할을 사용하여 세션 레코딩 시스템 구성
timesync	시간 동기화	RHEL 시스템 역할을 사용하여 시간 동기화 구성
vpn	VPN	vpn RHEL 시스템 역할을 사용하여 IPsec으로 VPN 연결 구성

추가 리소스

- RHEL 시스템 역할을 사용하여 시스템 관리 자동화
- Red Hat Enterprise Linux (RHEL) System Roles
- **rhel-system-roles** 패키지에서 제공하는 **/usr/share/doc/ rhel-system-roles /**

8장. 로그 파일을 사용하여 문제 해결

로그 파일에는 커널, 서비스 및 커널에서 실행되는 애플리케이션을 포함하여 시스템에 대한 메시지가 포함되어 있습니다. 여기에는 문제를 해결하거나 시스템 기능을 모니터링하는 데 도움이 되는 정보가 포함되어 있습니다. Red Hat Enterprise Linux의 로깅 시스템은 내장 **syslog** 프로토콜을 기반으로 합니다. 특정 프로그램은 이 시스템을 사용하여 이벤트를 기록하고 이를 로그 파일로 구성하며 운영 체제를 감사하고 다양한 문제를 해결할 때 유용합니다.

8.1. SYSLOG 메시지를 처리하는 서비스

다음 두 서비스는 **syslog** 메시지를 처리합니다.

- **systemd-journald** 데몬
- **Rsyslog** 서비스

systemd-journald 데몬은 다양한 소스에서 메시지를 수집하고 추가 처리를 위해 **Rsyslog** 에 전달합니다. **systemd-journald** 데몬은 다음 소스에서 메시지를 수집합니다.

- 커널
- 부팅 과정의 초기 단계
- 데몬의 표준 및 오류 출력 시작 및 실행
- **syslog**

Rsyslog 서비스는 **syslog** 메시지를 유형 및 우선 순위에 따라 정렬하고 **/var/log** 디렉터리의 파일에 씁니다. **/var/log** 디렉터리는 로그 메시지를 영구적으로 저장합니다.

8.2. SYSLOG 메시지를 저장하는 하위 디렉터리

/var/log 디렉터리 아래의 다음 하위 디렉터리에 **syslog** 메시지를 저장합니다.

- **/var/log/message** - 다음을 제외한 모든 **syslog** 메시지
- **/var/log/secure** - 보안 및 인증 관련 메시지 및 오류
- **/var/log/maillog** - 메일 서버 관련 메시지 및 오류
- **/var/log/cron** - 주기적으로 실행한 작업과 관련된 로그 파일
- **/var/log/boot.log** - 시스템 시작과 관련된 로그 파일

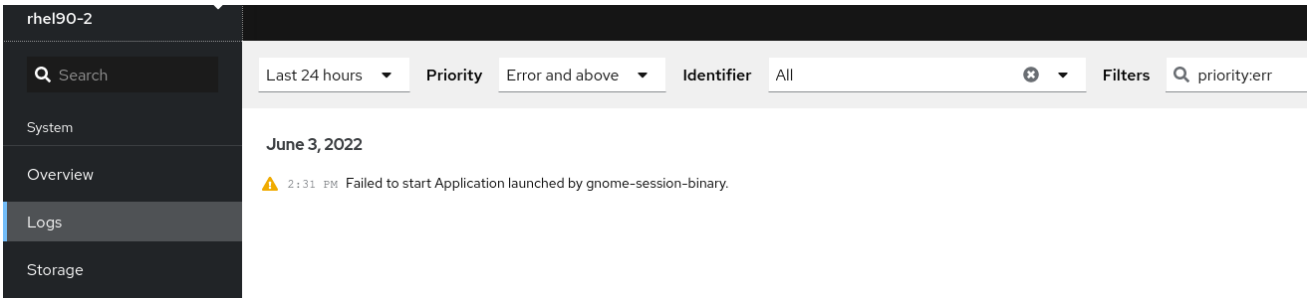
8.3. 웹 콘솔을 사용하여 로그 파일 검사

이 절차의 단계에 따라 RHEL 웹 콘솔을 사용하여 로그 파일을 검사합니다.

절차

1. RHEL 웹 콘솔에 로그인합니다. 자세한 내용은 웹 콘솔로의 로깅 을 참조하십시오.
2. **로그**를 클릭합니다.

그림 8.1. RHEL 9 웹 콘솔에서 로그 파일 검사



8.4. 명령줄을 사용하여 로그 보기

journal는 로그 파일을 보고 관리하는 데 도움이 되는 systemd의 구성 요소입니다. 다른 시스템과 밀접하게 통합되어 기존 로깅과 연결된 문제를 해결하고 로그 파일에 대한 다양한 로깅 기술 및 액세스 관리를 지원합니다.

journalctl 명령을 사용하여 명령줄을 사용하여 시스템 저널의 메시지를 볼 수 있습니다. 예를 들면 다음과 같습니다.

```
$ journalctl -b | grep kvm
May 15 11:31:41 localhost.localdomain kernel: kvm-clock: Using msrs 4b564d01 and 4b564d00
May 15 11:31:41 localhost.localdomain kernel: kvm-clock: cpu 0, msr 76401001, primary cpu clock
...
```

표 8.1. 시스템 정보 보기

명령	설명
journalctl	수집된 모든 저널 항목을 표시합니다.
journalctl FILEPATH	특정 파일과 관련된 로그를 표시합니다. 예를 들어 **journalctl /dev/sda** 명령은 **/dev/sda** 파일 시스템과 관련된 로그를 표시합니다.
journalctl -b	현재 부팅에 대한 로그를 표시합니다.
journalctl -k -b -1	현재 부팅에 대한 커널 로그를 표시합니다.

표 8.2. 특정 서비스에 대한 정보 보기

명령	설명
journalctl -b _SYSTEMD_UNIT=*<name.service>*	로그를 필터링하여 **systemd** 서비스와 일치하는 항목을 표시합니다.
journalctl -b _SYSTEMD_UNIT=*<name.service>* _PID=*<number>*	일치 항목이 결합됩니다. 예를 들어 이 명령은 < ***name.service*** > 및 PID < ***number*** >와 일치하는 **systemd-units** 에 대한 로그를 표시합니다.

명령	설명
journalctl -b _SYSTEMD_UNIT= <*name.service* > _PID= <*number* > + _SYSTEMD_UNIT= <*name2.service*>	더하기 기호(+) 구분 기호는 논리 OR로 두 표현식을 결합합니다. 예를 들어 이 명령은 < ***name.service*** > 서비스 프로세스의 모든 메시지와 < ***name2.service*** > 서비스의 모든 메시지(프로세스 중 하나)를 표시합니다.
journalctl -b _SYSTEMD_UNIT=<*name.service*> _SYSTEMD_UNIT=<*name2.service*>	이 명령은 동일한 필드를 참조하는 표현식과 일치하는 모든 항목을 표시합니다. 여기에서 이 명령은 systemd-unit < ***name.service*** > 또는 systemd-unit < ***name2.service*** >와 일치하는 로그를 표시합니다.

표 8.3. 특정 부팅과 관련된 로그 보기

명령	설명
journalctl --list-boots	부팅과 관련된 첫 번째 및 마지막 메시지의 테이블 형식 목록, 해당 ID 및 타임스탬프를 표시합니다. 다음 명령에서 ID를 사용하여 세부 정보를 볼 수 있습니다.
journalctl --boot=ID _SYSTEMD_UNIT=<*name.service*>	지정된 부팅 ID에 대한 정보를 표시합니다.

8.5. 추가 리소스

- **man journalctl(1)**
- 원격 로깅 솔루션 구성

9장. 사용자 및 그룹 관리

파일 및 프로세스에 대한 무단 액세스를 방지하려면 정확한 사용자 및 그룹 관리가 필요합니다. 계정을 중앙에서 관리하지 않거나 특정 시스템에서만 사용자 계정 또는 그룹이 필요한 경우 이 호스트에서 로컬로 생성할 수 있습니다.

9.1. 사용자 및 그룹 계정 관리 소개

사용자 및 그룹 제어는 RHEL(Red Hat Enterprise Linux) 시스템 관리의 핵심 요소입니다. 각 RHEL 사용자에게는 고유한 로그인 자격 증명이 있으며 다양한 그룹에 할당하여 시스템 권한을 사용자 지정할 수 있습니다.

9.1.1. 사용자 및 그룹 소개

파일을 생성하는 사용자는 해당 파일의 소유자와 해당 파일의 그룹 소유자입니다. *파일에는 소유자, 그룹 및 해당 그룹 외부의 사용자에 대한 별도의 읽기, 쓰기, 실행 권한이 할당됩니다. 파일 소유자는* ***root*** *사용자만 변경할 수 있습니다. 파일에 대한 액세스 권한은* ***root*** *사용자와 파일 소유자 모두에서 변경할 수 있습니다. 일반 사용자는 자신이 소유한 파일의 그룹 소유권을 멤버로 변경할 수 있습니다.*

*각 사용자는 사용자 ID(****UID****)라는 고유한 숫자 ID 번호와 연결됩니다. 각 그룹은 그룹 ID (****GID****)와 연결됩니다. 그룹 내의 사용자는 해당 그룹이 소유한 파일을 읽고, 쓰고, 실행할 수 있는 동일한 권한을 공유합니다.*

9.1.2. 예약된 사용자 및 그룹 ID 구성

RHEL은 시스템 사용자 및 그룹에 대해 1000 미만의 사용자 및 그룹 ID를 예약합니다. 예약된 사용자 및 그룹 ID는 ***setup*** *패키지에서 찾을 수 있습니다. 예약된 사용자 및 그룹 ID를 보려면 다음을 사용합니다.*

```
cat /usr/share/doc/setup*/uidgid
```

예약된 범위가 나중에 증가할 수 있으므로 5000에서 시작하는 새 사용자 및 그룹에 ID를 할당하는 것이 좋습니다.

기본적으로 새 사용자에 할당된 ID를 5000에서 시작하도록 하려면 ***/etc/login.defs*** *파일에서* ***UID_MIN*** *및* ***GID_MIN*** *매개변수를 수정합니다.*

절차

새 사용자에게 할당된 ID를 수정하려면 기본적으로 5000에서 시작합니다.

1. *선택한 편집기에서* ***/etc/login.defs*** *파일을 엽니다.*
2. *자동 UID 선택의 최소 값을 정의하는 행을 찾습니다.*

```
# Min/max values for automatic uid selection in useradd
#
UID_MIN                  1000
```

3. ***UID_MIN*** *값을 5000에서 시작하도록 수정합니다.*

```
# Min/max values for automatic uid selection in useradd
#
UID_MIN                  5000
```

4. *자동 GID 선택을 위한 최소 값을 정의하는 행을 찾습니다.*

```
# Min/max values for automatic gid selection in groupadd
#
GID_MIN                  1000
```

5. ***GID_MIN*** *값을 수정하여 5000 에서 시작합니다.*

```
# Min/max values for automatic gid selection in groupadd
#
GID_MIN                  5000
```

일반 사용자에 대해 동적으로 할당된 UID 및 GID 는 이제 5000 에서 시작합니다.

참고

UID_MIN 및 GID_MIN 값을 변경하기 전에 생성된 UID 및 GID 의 사용자 및 그룹은 변경되지 않습니다.

이렇게 하면 새 사용자 그룹이 UID 및 GID 와 5000+ ID 를 가질 수 있습니다.

주의

1000 제한을 유지하는 시스템과 충돌하지 않도록 ***SYS_UID_MAX*** *를 변경하여 시스템에서 예약된 ID 를 생성하지 마십시오.*

9.1.3. 사용자 개인 그룹

*RHEL 은 사용자 개인 그룹 (**UPG**) 시스템 구성을 사용하므로 UNIX 그룹을 보다 쉽게 관리할 수 있습니다. 새 사용자가 시스템에 추가될 때마다 사용자 개인 그룹이 생성됩니다. 사용자 개인 그룹은 생성된 사용자와 동일한 이름을 가지며, 해당 사용자는 사용자 개인 그룹의 유일한 멤버입니다.*

UPG 는 여러 사용자 간의 프로젝트에서의 협업을 단순화합니다. 또한 UPG 시스템 구성을 사용하면 사용자 및 이 사용자가 파일 또는 디렉터리를 모두 수정할 수 있으므로 새로 생성된 파일 또는 디렉터리에 대한 기본 권한을 안전하게 설정할 수 있습니다.

모든 그룹 목록은 ***/etc/group*** *구성 파일에 저장됩니다.*

9.2. 사용자 계정 관리 시작하기

Red Hat Enterprise Linux 는 다중 사용자 운영 체제로, 다른 컴퓨터에서 여러 사용자가 한 시스템에 설치된 단일 시스템에 액세스할 수 있습니다. 모든 사용자는 자체 계정으로 운영되며 사용자 계정을 관리하여 Red Hat Enterprise Linux 시스템 관리의 핵심 요소를 나타냅니다.

다음은 다양한 유형의 사용자 계정입니다.

- ***일반 사용자 계정:***
  특정 시스템의 사용자를 위해 일반 계정이 생성됩니다. 이러한 계정은 일반 시스템 관리 중에 추가, 제거 및 수정할 수 있습니다.

- ***시스템 사용자 계정:***

시스템 사용자 계정은 시스템의 특정 애플리케이션 식별자를 나타냅니다. 이러한 계정은 일반적으로 소프트웨어 설치 시에만 추가되거나 조작되며 나중에 수정되지 않습니다.

주의

시스템에서 로컬로 시스템 계정을 사용할 수 있다고 가정합니다. LDAP 구성 인스턴스에서와 같이 이러한 계정이 원격으로 구성되고 제공되는 경우 시스템 중단 및 서비스 시작 오류가 발생할 수 있습니다.

*시스템 계정의 경우 1000 미만의 사용자 ID가 예약됩니다. 일반 계정의 경우 1000부터 시작하는 ID를 사용할 수 있습니다. 그러나 권장 사례는 5000에서 시작하는 ID를 할당하는 것입니다. ID 할당은 **/etc/login.defs** 파일을 참조하십시오.*

- ***group:***
  그룹은 특정 파일에 대한 액세스 권한 부여와 같은 공통 목적을 위해 여러 사용자 계정을 함께 연결하는 엔티티입니다.

9.2.1. 명령줄 도구를 사용하여 계정 및 그룹 관리

다음 기본 명령줄 툴을 사용하여 사용자 계정 및 그룹을 관리합니다.

- *사용자 및 그룹 ID를 표시하려면 다음을 수행합니다.*

```
$ id
uid=1000(example.user) gid=1000(example.user) groups=1000(example.user),10(wheel)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

- *새 사용자 계정을 생성하려면 다음을 수행합니다.*

```
# useradd example.user
```

- *example.user 에 속하는 사용자 계정에 새 암호를 할당하려면 다음을 수행합니다.*

```
# passwd example.user
```

- *그룹에 사용자를 추가하려면 다음을 수행합니다.*

```
# usermod -a -G example.group example.user
```

추가 리소스

- ***man useradd(8)*, *man passwd(1)*, *man usermod(8)***

9.2.2. 웹 콘솔에서 관리되는 시스템 사용자 계정

RHEL 웹 콘솔에 사용자 계정이 표시되는 경우 다음을 수행할 수 있습니다.

- *시스템에 액세스할 때 사용자를 인증합니다.*

- 시스템에 대한 액세스 권한을 설정합니다.

RHEL 웹 콘솔은 시스템에 있는 모든 사용자 계정을 표시합니다. 따라서 웹 콘솔에 처음 로그인한 직후 적어도 하나의 사용자 계정을 볼 수 있습니다.

RHEL 웹 콘솔에 로그인한 후 다음 작업을 수행할 수 있습니다.

- 새 사용자 계정을 생성합니다.
- 매개 변수를 변경합니다.
- 계정을 잠급니다.
- 사용자 세션을 종료합니다.

9.2.3. 웹 콘솔을 사용하여 새 계정 추가

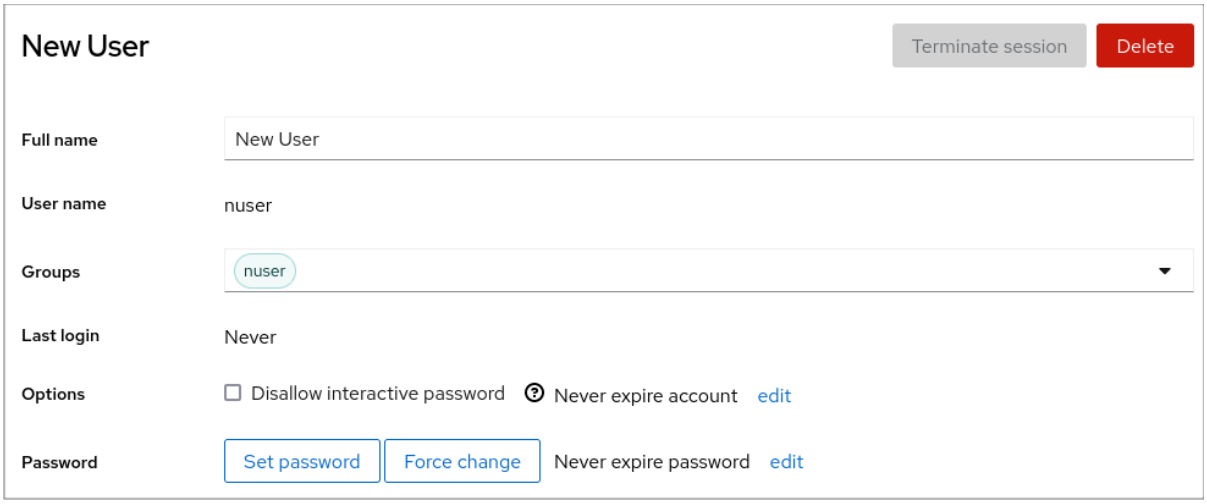
다음 단계를 사용하여 시스템에 사용자 계정을 추가하고 RHEL 웹 콘솔을 통해 계정에 관리 권한을 설정합니다.

사전 요구 사항

- RHEL 웹 콘솔을 설치하고 액세스할 수 있어야 합니다. 자세한 내용은 *웹 콘솔 설치*를 참조하십시오.

절차

1. RHEL 웹 콘솔에 로그인합니다.
2. **계정**을 클릭합니다.
3. **새 계정 만들기**를 클릭합니다.
4. **전체 이름** 필드에 사용자의 전체 이름을 입력합니다.
   RHEL 웹 콘솔은 전체 이름에서 사용자 이름을 자동으로 권장하고 **사용자 이름** 필드에 입력합니다. 첫 번째 이름의 첫 글자와 전체 성으로 구성된 원래 이름 지정 규칙을 사용하지 않으려면 제안을 업데이트합니다.
5. **Password/Confirm** 필드에 암호를 입력하고 암호가 올바른지 확인하도록 다시 입력합니다.
   필드 아래의 색상 표시줄은 입력한 암호의 보안 수준을 보여주므로 암호가 약한 사용자를 만들 수 없습니다.
6. **만들기**를 클릭하여 설정을 저장하고 대화 상자를 종료합니다.
7. 새로 생성된 계정을 선택합니다.
8. **그룹** 드롭다운 메뉴에서 새 계정에 추가할 그룹을 선택합니다.



*이제 **계정** 설정에서 새 계정을 볼 수 있으며 해당 자격 증명을 사용하여 시스템에 연결할 수 있습니다.*

9.3. 명령줄에서 사용자 관리

CLI(명령줄 인터페이스)를 사용하여 사용자 및 그룹을 관리할 수 있습니다. 이를 통해 Red Hat Enterprise Linux 환경에서 사용자 및 사용자 그룹을 추가, 제거 및 수정할 수 있습니다.

9.3.1. 명령줄에서 새 사용자 추가

***useradd** 유틸리티를 사용하여 새 사용자를 추가할 수 있습니다.*

사전 요구 사항

- *루트 액세스*

절차

- *새 사용자를 추가하려면 다음을 사용합니다.*

```
# useradd options username
```

*옵션을 **useradd** 명령의 명령줄 옵션으로 바꾸고 username 을 사용자 이름으로 교체합니다.*

> **예 9.1. 새 사용자 추가**
>
> *사용자 ID **5000** 을 사용하여 사용자 **sarah** 를 추가하려면 다음을 사용하십시오.*
>
> ```
> # useradd -u 5000 sarah
> ```

검증 단계

- *새 사용자가 추가되었는지 확인하려면 **id** 유틸리티를 사용합니다.*

```
# id sarah
```

출력이 반환됩니다.

```
uid=5000(sarah) gid=5000(sarah) groups=5000(sarah)
```

추가 리소스

- **useradd** man 페이지

9.3.2. 명령줄에서 새 그룹 추가

groupadd 유틸리티를 사용하여 새 그룹을 추가할 수 있습니다.

사전 요구 사항

- 루트 액세스

절차

- 새 그룹을 추가하려면 다음을 사용합니다.

```
# groupadd options group-name
```

옵션을 **groupadd** 명령의 명령줄 옵션으로 바꾸고 group-name 을 그룹 이름으로 교체합니다.

예 9.2. 새 그룹 추가

그룹 ID가 **5000** 으로 그룹 installers를 추가하려면 다음을 사용합니다.

```
# groupadd -g 5000 sysadmins
```

검증 단계

- 새 그룹이 추가되었는지 확인하려면 **tail** 유틸리티를 사용합니다.

```
# tail /etc/group
```

출력이 반환됩니다.

```
sysadmins:x:5000:
```

추가 리소스

- **groupadd man** 페이지

9.3.3. 명령줄에서 사용자를 보조 그룹에 추가

보조 그룹에 사용자를 추가하여 권한을 관리하거나 특정 파일 또는 장치에 대한 액세스를 활성화할 수 있습니다.

사전 요구 사항

- 루트 액세스

절차

- 사용자 보조 그룹에 그룹을 추가하려면 다음을 사용합니다.

```
# usermod --append -G group-name username
```

group-name 을 그룹 이름으로 바꾸고 **username** 을 사용자 이름으로 교체합니다.

예 **9.3.** 보조 그룹에 사용자 추가

사용자 **administrator**를 **system-administrators** 그룹에 추가하려면 다음을 사용합니다.

```
# usermod --append -G system-administrators sysadmin
```

검증 단계

- 새 그룹이 사용자 **administrator**의 보조 그룹에 추가되었는지 확인하려면 다음을 사용합니다.

```
# groups sysadmin
```

출력이 표시됩니다.

```
sysadmin : sysadmin system-administrators
```

9.3.4. 그룹 디렉터리 생성

UPG 시스템 구성에서 **set-group** 식별 권한**(set gid bit)**을 디렉터리에 적용할 수 있습니다. **setgid** 비트를 사용하면 디렉터리를 더 간단하게 공유하는 그룹 프로젝트를 관리할 수 있습니다. **setgid** 비트를 디렉터리에 적용하면 해당 디렉터리 내에서 생성된 파일이 디렉터리를 소유하는 그룹에 자동으로 할당됩니

다. 이 그룹 내에서 쓰기 및 실행할 권한이 있는 사용자는 이제 디렉터리에서 파일을 생성, 수정 및 삭제할 수 있습니다.

다음 섹션에서는 그룹 디렉터리를 만드는 방법을 설명합니다.

사전 요구 사항

- *루트 액세스*

절차

1. *디렉터리를 생성합니다.*

```
# mkdir directory-name
```

   ***directory-name** 을 디렉터리 이름으로 바꿉니다.*

2. *그룹을 생성합니다.*

```
# groupadd group-name
```

   ***group-name** 을 그룹 이름으로 바꿉니다.*

3. *그룹에 사용자를 추가합니다.*

```
# usermod --append -G group-name username
```

   ***group-name** 을 그룹 이름으로 바꾸고 **username** 을 사용자 이름으로 교체합니다.*

4. *디렉터리의 사용자 및 그룹 소유권을 **group-name** 그룹과 연결합니다.*

```
# chgrp group-name directory-name
```

   ***group-name** 을 그룹 이름으로 바꾸고 **directory-name** 을 디렉터리 이름으로 바꿉니다.*

5.

*사용자가 파일 및 디렉터리를 생성 및 수정하고 **setgid** 비트를 설정하여 **directory-name** 디렉터리 내에 이 권한을 적용할 수 있도록 쓰기 권한을 설정합니다.*

```
# chmod g+rwxs directory-name
```

***directory-name** 을 디렉터리 이름으로 바꿉니다.*

*이제 **group-name** 그룹의 모든 멤버는 **directory-name** 디렉터리에서 파일을 생성하고 편집할 수 있습니다. 새로 생성된 파일은 **group-name** 그룹의 그룹 소유권을 유지합니다.*

검증 단계

- *설정된 권한의 정확성을 확인하려면 다음을 사용합니다.*

```
# ls -ld directory-name
```

***directory-name** 을 디렉터리 이름으로 바꿉니다.*

출력이 반환됩니다.

```
drwxrwsr-x. 2 root group-name 6 Nov 25 08:45 directory-name
```

9.3.5. 명령줄에서 사용자 제거

명령줄을 사용하여 사용자 계정을 제거할 수 있습니다. 사용자 계정을 제거하는 것 외에도 선택적으로 홈 디렉터리 및 구성 파일과 같은 사용자 데이터 및 메타데이터를 제거할 수 있습니다.

사전 요구 사항

- ***root** 액세스 권한이 있습니다.*
- *사용자가 현재 존재합니다.*
- *사용자가 로그아웃되었는지 확인합니다.*

```
# loginctl terminate-user user-name
```

절차

- *사용자 데이터가 아닌 사용자 계정만 제거하려면 다음을 수행합니다.*

```
# userdel user-name
```

- *사용자, 데이터 및 메타데이터를 제거하려면 다음을 수행합니다.*

a. *사용자, 해당 홈 디렉터리, 메일 스풀 및 해당 **SELinux** 사용자 매핑을 제거합니다.*

```
# userdel --remove --selinux-user user-name
```

b. *추가 사용자 메타데이터 제거:*

```
# rm -rf /var/lib/AccountsService/users/user-name
```

이 디렉터리는 홈 디렉터리를 사용할 수 있기 전에 시스템에 필요한 정보를 사용자 에 저장합니다. 시스템 구성에 따라 사용자가 로그인 화면에서 인증할 때까지 홈 디렉터리를 사용할 수 없을 수 있습니다.

중요

이 디렉터리를 제거하지 않고 나중에 동일한 사용자를 다시 생성하는 경우 다시 생성한 사용자는 삭제된 사용자에게 상속된 특정 설정을 계속 사용합니다.

추가 리소스

- ***userdel(8)** 도움말 페이지.*

9.4. 웹 콘솔에서 사용자 계정 관리

***RHEL** 웹 콘솔은 터미널에 직접 액세스하지 않고도 광범위한 관리 작업을 실행할 수 있는 그래픽 인터페이스를 제공합니다. 예를 들어 시스템 사용자 계정을 추가, 편집 또는 제거할 수 있습니다.*

이 섹션을 읽으면 다음을 알 수 있습니다.

- 기존 계정이 있는 위치에서.
- 새 계정을 추가하는 방법
- 암호 만료를 설정하는 방법
- 사용자 세션을 종료하는 방법 및 시기

사전 요구 사항

- **RHEL** 웹 콘솔을 설정합니다. 자세한 내용은 *RHEL 웹 콘솔 사용하기 시작을* 참조하십시오.
- 관리자 권한이 할당된 계정으로 **RHEL** 웹 콘솔에 로그인합니다. 자세한 내용은 *RHEL 웹 콘솔에 로그인을* 참조하십시오.

9.4.1. 웹 콘솔에서 관리되는 시스템 사용자 계정

RHEL 웹 콘솔에 사용자 계정이 표시되는 경우 다음을 수행할 수 있습니다.

- 시스템에 액세스할 때 사용자를 인증합니다.
- 시스템에 대한 액세스 권한을 설정합니다.

RHEL 웹 콘솔은 시스템에 있는 모든 사용자 계정을 표시합니다. 따라서 웹 콘솔에 처음 로그인한 직후 적어도 하나의 사용자 계정을 볼 수 있습니다.

RHEL 웹 콘솔에 로그인한 후 다음 작업을 수행할 수 있습니다.

- *새 사용자 계정을 생성합니다.*
- *매개 변수를 변경합니다.*
- *계정을 잠급니다.*
- *사용자 세션을 종료합니다.*

9.4.2. 웹 콘솔을 사용하여 새 계정 추가

*다음 단계를 사용하여 시스템에 사용자 계정을 추가하고 **RHEL** 웹 콘솔을 통해 계정에 관리 권한을 설정합니다.*

사전 요구 사항

- ***RHEL** 웹 콘솔을 설치하고 액세스할 수 있어야 합니다. 자세한 내용은 웹 콘솔 설치를 참조하십시오.*

절차

1. ***RHEL** 웹 콘솔에 로그인합니다.*
2. *계정을 클릭합니다.*
3. *새 계정 만들기를 클릭합니다.*
4. *전체 이름 필드에 사용자의 전체 이름을 입력합니다.*

   ***RHEL** 웹 콘솔은 전체 이름에서 사용자 이름을 자동으로 권장하고 사용자 이름 필드에 입력합니다. 첫 번째 이름의 첫 글자와 전체 성으로 구성된 원래 이름 지정 규칙을 사용하지 않으려면 제안을 업데이트합니다.*

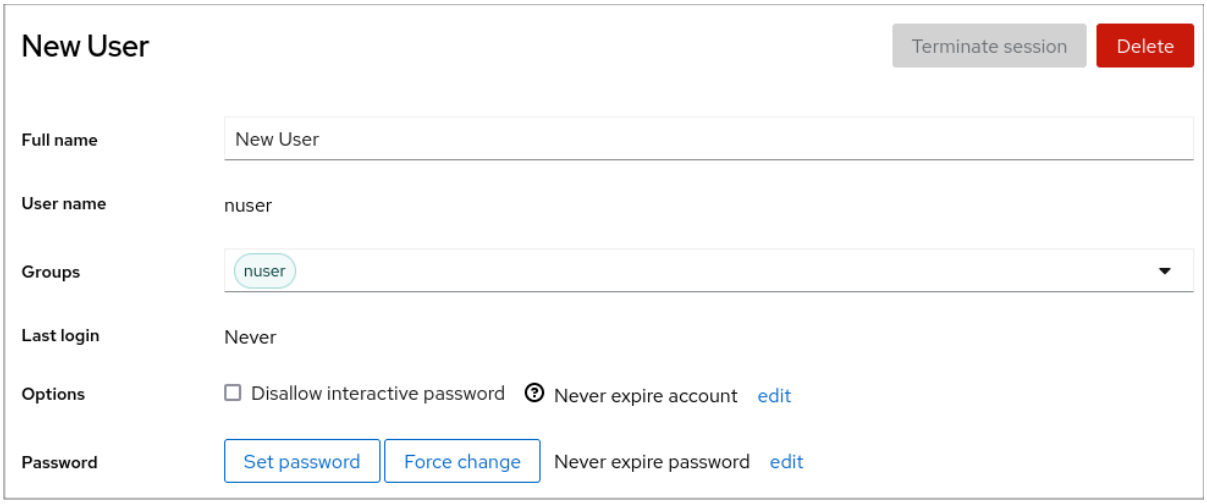
5. ***Password/Confirm** 필드에 암호를 입력하고 암호가 올바른지 확인하도록 다시 입력합니다.*

필드 아래의 색상 표시줄은 입력한 암호의 보안 수준을 보여주므로 암호가 약한 사용자를 만들 수 없습니다.

6. *만들기를 클릭하여 설정을 저장하고 대화 상자를 종료합니다.*

7. *새로 생성된 계정을 선택합니다.*

8. *그룹 드롭다운 메뉴에서 새 계정에 추가할 그룹을 선택합니다.*



이제 계정 설정에서 새 계정을 볼 수 있으며 해당 자격 증명을 사용하여 시스템에 연결할 수 있습니다.

9.4.3. 웹 콘솔에서 암호 만료 강제 적용

기본적으로 사용자 계정은 만료되지 않도록 암호를 설정합니다. 정의된 일 수 후에 만료되도록 시스템 암호를 설정할 수 있습니다. 암호가 만료되면 다음 로그인 시도에서 암호 변경을 묻는 메시지가 표시됩니다.

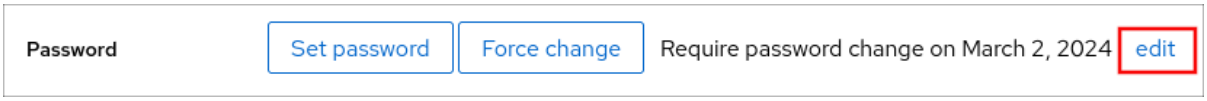
절차

1. ***RHEL 9** 웹 콘솔에 로그인합니다.*

2. *계정을 클릭합니다.*

3. *암호 만료를 적용할 사용자 계정을 선택합니다.*

4. *암호 행에서 **edit** 를 클릭합니다.*



5. *암호 만료 대화 상자에서 암호 변경 시 모든 ...을 선택합니다. **days** 을(를) 입력하고 암호가 만료된 날 수를 나타내는 양의 정수 숫자를 입력합니다.*

6. *변경을 클릭합니다.*

웹 콘솔은 암호 줄에 향후 암호 변경 요청 날짜를 즉시 표시합니다.

9.4.4. 웹 콘솔에서 사용자 세션 종료

사용자는 시스템에 로그인할 때 사용자 세션을 생성합니다. 사용자 세션을 종료하는 것은 사용자가 시스템에서 로그아웃하는 것을 의미합니다. 구성 변경에 민감한 관리 작업을 수행해야 하는 경우(예: 시스템 업그레이드) 도움이 될 수 있습니다.

***RHEL 9web** 콘솔의 각 사용자 계정에서 현재 사용 중인 웹 콘솔 세션을 제외하고 계정의 모든 세션을 종료할 수 있습니다. 이렇게 하면 시스템에 대한 액세스를 차단할 수 없습니다.*

절차

1. ***RHEL 9** 웹 콘솔에 로그인합니다.*

2. *계정을 클릭합니다.*

3. 세션을 종료할 사용자 계정을 클릭합니다.

4. **Terminate Session** 을 클릭합니다.

   Terminate Session 버튼이 비활성화된 경우 사용자는 시스템에 로그인되지 않습니다.

   RHEL 웹 콘솔은 세션을 종료합니다.

9.5. 명령줄을 사용하여 사용자 그룹 편집

사용자는 파일 및 폴더에 대한 유사한 액세스 권한을 가진 사용자의 논리적 컬렉션을 허용하는 특정 그룹 세트에 속합니다. 명령줄에서 기본 및 보조 사용자 그룹을 편집하여 사용자의 권한을 변경할 수 있습니다.

9.5.1. 기본 및 보조 사용자 그룹

그룹은 특정 파일에 대한 액세스 권한 부여와 같은 공통 목적을 위해 여러 사용자 계정을 함께 연결하는 엔티티입니다.

Linux에서 사용자 그룹은 기본 또는 보조 역할을 할 수 있습니다. 기본 및 보조 그룹에는 다음 속성이 있습니다.

기본 그룹

- 모든 사용자에게는 항상 하나의 기본 그룹만 있습니다.
- 사용자의 기본 그룹을 변경할 수 있습니다.

보조 그룹

- 기존 사용자를 기존 보조 그룹에 추가하여 그룹 내에서 동일한 보안 및 액세스 권한으로 사용자를 관리할 수 있습니다.
- 사용자는 **0**개 또는 여러 개의 보조 그룹의 구성원이 될 수 있습니다.

9.5.2. 사용자의 기본 및 보조 그룹 나열

사용자 그룹을 나열하여 자신이 속한 기본 및 보조 그룹을 확인할 수 있습니다.

절차

- 사용자의 기본 그룹과 보조 그룹의 이름을 표시합니다.

```
$ groups user-name
```

user-name 을 사용자 이름으로 변경합니다. 사용자 이름을 제공하지 않으면 명령은 현재 사용자의 그룹 멤버십을 표시합니다. 첫 번째 그룹은 기본 그룹 다음에 선택적 보조 그룹이 옵니다.

예 9.4. 사용자 **sarah**의 그룹 목록:

```
$ groups sarah
```

출력이 표시됩니다.

```
sarah : sarah wheel developer
```

User sarah 에는 기본 **group sarah** 가 있으며, 보조 그룹 **wheel** 및 **developer** 의 구성원입니다.

예 9.5. 사용자 **marc**의 그룹 목록:

```
$ groups marc
```

출력이 표시됩니다.

```
marc : marc
```

사용자 **marc** 에는 기본 그룹 **marc** 및 보조 그룹이 없습니다.

9.5.3. 사용자의 기본 그룹 변경

기존 사용자의 기본 그룹을 새 그룹으로 변경할 수 있습니다.

사전 요구 사항

1. 루트 액세스

2. 새 그룹이 존재해야 합니다.

절차

- 사용자의 기본 그룹을 변경합니다.

```
# usermod -g group-name user-name
```

group-name 을 새 기본 그룹의 이름으로 바꾸고 **user-name** 을 사용자 이름으로 교체합니다.

참고

사용자의 기본 그룹을 변경하면 명령은 사용자 홈 디렉터리에 있는 모든 파일의 그룹 소유권을 새 기본 그룹으로 자동 변경합니다. 사용자의 홈 디렉터리 외부에서 파일의 그룹 소유권을 수동으로 수정해야 합니다.

예 9.6. 사용자의 기본 그룹을 변경하는 예:

user sarah가 기본 **group sarah1** 에 속하고 사용자의 기본 그룹을 **sarah2** 로 변경하려면 다음을 사용합니다.

```
# usermod -g sarah2 sarah
```

검증 단계

- 사용자의 기본 그룹을 변경했는지 확인합니다.

```
$ groups sarah
```

출력이 표시됩니다.

```
sarah : sarah2
```

9.5.4. 명령줄에서 사용자를 보조 그룹에 추가

보조 그룹에 사용자를 추가하여 권한을 관리하거나 특정 파일 또는 장치에 대한 액세스를 활성화할 수 있습니다.

사전 요구 사항

- 루트 액세스

절차

- 사용자 보조 그룹에 그룹을 추가하려면 다음을 사용합니다.

```
# usermod --append -G group-name username
```

group-name 을 그룹 이름으로 바꾸고 **username** 을 사용자 이름으로 교체합니다.

예 **9.7.** 보조 그룹에 사용자 추가

사용자 **administrator**를 **system-administrators** 그룹에 추가하려면 다음을 사용합니다.

```
# usermod --append -G system-administrators sysadmin
```

검증 단계

- 새 그룹이 사용자 **administrator**의 보조 그룹에 추가되었는지 확인하려면 다음을 사용합니다.

```
# groups sysadmin
```

출력이 표시됩니다.

```
sysadmin : sysadmin system-administrators
```

9.5.5. 보조 그룹에서 사용자 제거

보조 그룹에서 기존 사용자를 제거하여 파일 및 장치에 대한 권한을 제한할 수 있습니다.

사전 요구 사항

- 루트 액세스

절차

- 보조 그룹에서 사용자 제거:

```
# gpasswd -d user-name group-name
```

user-name 을 사용자 이름으로 바꾸고 **group-name** 을 보조 그룹 이름으로 교체합니다.

예 9.8. 보조 그룹에서 사용자 제거

사용자 **sarah**에 기본 **group sarah2** 가 있고 보조 그룹 **wheel** 및 **developers** 에 속하고 그룹 개발자에서 해당 사용자를 제거하려면 다음을 사용합니다.

```
# gpasswd -d sarah developers
```

검증 단계

- 보조 그룹 **developers**에서 사용자 **sarah**가 제거되었는지 확인합니다.

```
$ groups sarah
```

출력이 표시됩니다.

```
sarah : sarah2 wheel
```

9.5.6. 사용자의 모든 보조 그룹 변경

사용자가 멤버로 유지하려는 보조 그룹 목록을 덮어쓸 수 있습니다.

사전 요구 사항

- *루트 액세스*
- *보조 그룹이 있어야 합니다.*

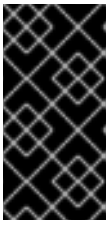
절차

- *사용자의 보조 그룹 목록을 덮어씁니다.*

```
# usermod -G group-names username
```

group-names** 를 하나 이상의 보조 그룹의 이름으로 바꿉니다. 사용자를 한 번에 여러 보조 그룹에 추가하려면 그룹 이름을 쉼표와 중간 공백을 사용하여 구분합니다. 예: **wheel,developer.

***user-name** 을 사용자 이름으로 변경합니다.*



중요

사용자가 현재 지정하지 않는 그룹의 멤버인 경우 명령은 그룹에서 사용자를 제거합니다.

*예 **9.9.** 사용자의 보조 그룹 목록 변경*

***user sarah** 에 기본 **group sarah2** 가 있고 보조 그룹 **wheel** 에 속하고 사용자가 **3**개의 보조 그룹 **developer,sysadmin** 및 **security** 에 속하게 하려면 다음을 사용합니다.*

```
# usermod -G wheel,developer,sysadmin,security sarah
```

검증 단계

- *보조 그룹 목록을 올바르게 설정했는지 확인합니다.*

```
# groups sarah
```

출력이 표시됩니다.

```
sarah : sarah2 wheel developer sysadmin security
```

9.6. 루트 암호 변경 및 재설정

기존 **root** 암호가 더 이상 적합하지 않거나 잊혀진 경우 **root** 사용자와 비 루트 사용자 모두 변경 또는 재설정할 수 있습니다.

9.6.1. root 사용자로 root 암호 변경

passwd 명령을 사용하여 **root** 사용자로 **root** 암호를 변경할 수 있습니다.

사전 요구 사항

- 루트 액세스

절차

- 루트 암호를 변경하려면 다음을 사용합니다.

```
# passwd
```

암호를 변경하기 전에 현재 암호를 입력하라는 메시지가 표시됩니다.

9.6.2. 루트가 아닌 사용자로 잊혀진 루트 암호 변경 또는 재설정

passwd 명령을 사용하여 루트가 아닌 사용자로 알 수 있는 **root** 암호를 변경하거나 재설정할 수 있습니다.

사전 요구 사항

- 루트가 아닌 사용자로 로그인할 수 있습니다.

- 당신은 관리 워치 그룹의 멤버입니다.

절차

- **wheel** 그룹에 속하는 루트가 아닌 사용자로 **root** 암호를 변경하거나 재설정하려면 다음을 사용합니다.

```
$ sudo passwd root
```

root 암호를 변경하기 전에 현재 **root** 가 아닌 암호를 입력하라는 메시지가 표시됩니다.

9.6.3. 부팅 시 루트 암호 재설정

루트가 아닌 사용자로 로그인할 수 없거나 관리 **wheel** 그룹에 속하지 않는 경우 특수 **chroot would** 환경으로 전환하여 부팅 시 루트 암호를 재설정할 수 있습니다.

절차

1. 시스템을 재부팅하고 **GRUB 2** 부팅 화면에서 **e** 키를 눌러 부팅 프로세스를 중단합니다.

   커널 부팅 매개변수가 나타납니다.

```
load_video
set gfx_payload=keep
insmod gzio
linux ($root)/vmlinuz-5.14.0-70.22.1.e19_0.x86_64 root=/dev/mapper/rhel-root ro crash\
kernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv/swap rhgb quiet
initrd ($root)/initramfs-5.14.0-70.22.1.e19_0.x86_64.img $tuned_initrd
```

2. **linux** 로 시작하는 행 끝으로 이동하십시오.

```
linux ($root)/vmlinuz-5.14.0-70.22.1.e19_0.x86_64 root=/dev/mapper/rhel-root ro crash\
kernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv/swap rhgb quiet
```

   Ctrl+e 를 눌러 행 끝으로 건너뜁니다.

3. **linux** 로 시작하는 행 끝에 **rd.break** 를 추가합니다.

```
linux ($root)/vmlinuz-5.14.0-70.22.1.e19_0.x86_64 root=/dev/mapper/rhel-root ro crash\
kernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv/swap rhgb quiet rd.break
```

4. **Ctrl+x** 를 눌러 변경된 매개 변수를 사용하여 시스템을 시작합니다.

   switch_root 프롬프트가 나타납니다.

5. 파일 시스템을 쓰기 가능으로 다시 마운트합니다.

   ```
   mount -o remount,rw /sysroot
   ```

   파일 시스템은 **/sysroot** 디렉토리에서 읽기 전용으로 마운트됩니다. 파일 시스템을 쓰기 가능으로 다시 마운트하면 암호를 변경할 수 있습니다.

6. **chroot** 환경을 입력합니다.

   ```
   chroot /sysroot
   ```

   sh-4.4# 프롬프트가 표시됩니다.

7. 루트 암호를 재설정합니다.

   ```
   passwd
   ```

   명령줄에 표시된 지침에 따라 **root** 암호 변경을 완료합니다.

8. 다음 시스템 부팅 시 **SELinux** 재지정 프로세스를 활성화합니다.

   ```
   touch /.autorelabel
   ```

9. **chroot** 환경을 종료합니다.

   ```
   exit
   ```

10. *switch_root 프롬프트를 종료합니다.*

```
exit
```

11. *SELinux 레이블 지정 프로세스가 완료될 때까지 기다립니다. 큰 디스크의 레이블을 다시 지정하는 데 시간이 오래 걸릴 수 있습니다. 프로세스가 완료되면 시스템이 자동으로 재부팅됩니다.*

검증 단계

1. *루트 암호가 변경되었는지 확인하려면 일반 사용자로 로그인하여 터미널을 엽니다.*

2. *대화형 셸을 root로 실행합니다.*

```
$ su
```

3. *새 루트 암호를 입력합니다.*

4. *현재 유효한 사용자 ID와 관련된 사용자 이름을 인쇄합니다.*

```
# whoami
```

출력이 반환됩니다.

```
root
```

10장. SUDO 액세스 관리

시스템 관리자는 루트가 아닌 사용자가 일반적으로 **root** 사용자로 예약된 관리 명령을 실행할 수 있도록 **sudo** 액세스 권한을 부여할 수 있습니다. 결과적으로 루트가 아닌 사용자는 **root** 사용자 계정에 로그인하지 않고 이러한 명령을 실행할 수 있습니다.

10.1. SUDOERS의 사용자 권한 부여

/etc/sudoers 파일은 **sudo** 명령을 사용하여 다른 명령을 실행할 수 있는 사용자를 지정합니다. 규칙은 개별 사용자 및 사용자 그룹에 적용할 수 있습니다. 별칭을 사용하여 호스트 그룹, 명령 및 사용자도 더 쉽게 정의할 수도 있습니다. 기본 별칭은 **/etc/sudoers** 파일의 첫 번째 부분에 정의됩니다.

사용자가 권한이 없는 **sudo** 를 사용하여 명령을 입력하면 시스템은 **< username > :** 사용자가 저널 로그에 대한 **sudoers**에 없는 메시지를 기록합니다.

기본 **/etc/sudoers** 파일은 권한 부여의 정보와 예를 제공합니다. 해당 행의 주석을 제거하여 특정 예제 규칙을 활성화할 수 있습니다. 사용자 권한 부여 섹션은 다음 도입으로 표시됩니다.

```
## Next comes the main part: which users can run what software on
## which machines  (the sudoers file can be shared between multiple
## systems).
```

다음 형식을 사용하여 새 **sudoers** 권한을 생성하고 기존 권한 부여를 수정할 수 있습니다.

```
<username> <hostname.example.com>=(<run_as_user>:<run_as_group>) <path/to/command>
```

다음과 같습니다.

- **<username >**은 명령을 입력하는 사용자입니다(예: **user1). 값이 % 로 시작되면 그룹을 정의합니다(예: %group1).**
- **<hostname.example.com >**은 규칙이 적용되는 호스트의 이름입니다.
- 섹션 **(<run_as_user > : <run_as_group>)** 은 명령이 실행되는 사용자 또는 그룹을 정의합니다. 이 섹션을 생략하면 **< username>** 에서 **root**로 명령을 실행할 수 있습니다.

- **<path/to/command >**는 명령의 전체 절대 경로입니다. 명령 경로 뒤에 해당 옵션을 추가하여 특정 옵션 및 인수가 있는 명령만 실행하도록 사용자를 제한할 수도 있습니다. 옵션을 지정하지 않으면 사용자는 모든 옵션과 함께 명령을 사용할 수 있습니다.

이러한 변수를 **ALL** 로 교체하여 모든 사용자, 호스트 또는 명령에 규칙을 적용할 수 있습니다.

주의

ALL =(ALL) ALL 과 같은 과도하게 허용 가능한 규칙을 사용하면 모든 사용자가 모든 호스트에서 모든 사용자로 모든 명령을 실행할 수 있습니다. 이는 심각한 보안 위험을 초래합니다.

! 연산자를 사용하여 인수를 부정적인 방식으로 지정할 수 있습니다. 예를 들어 **!root** 는 **root**를 제외한 모든 사용자를 지정합니다. 특정 사용자, 그룹 및 명령을 허용하는 것은 특정 사용자, 그룹 및 명령을 허용하지 않는 것보다 안전합니다. 이는 규칙이 권한이 없는 새 사용자 또는 그룹도 차단하기 때문입니다.

주의

사용자가 **alias** 명령으로 명령 이름을 변경하여 이러한 규칙을 극복할 수 있으므로 명령에 음수 규칙을 사용하지 마십시오.

시스템은 **/etc/sudoers** 파일을 처음부터 끝까지 읽습니다. 따라서 파일에 사용자에 대한 여러 항목이 포함된 경우 항목이 순서대로 적용됩니다. 충돌하는 값이 있는 경우 시스템이 가장 구체적인 일치 항목이 아닌 경우에도 마지막 일치 항목을 사용합니다.

시스템 업데이트 중 규칙을 유지하고 오류를 보다 쉽게 수정하려면 **/etc/sudoers.d/** 디렉터리에 새 파일을 **/etc/sudoers** 파일에 직접 입력하는 대신 새 규칙을 입력합니다. **/etc/sudoers.d** 디렉터리에서 **/etc/sudoers** 파일의 다음 행에 도달하면 시스템이 **/etc/sudoers.d** 디렉토리의 파일을 읽습니다.

```
#includedir /etc/sudoers.d
```

이 줄의 시작 부분에 있는 숫자 기호(#)는 구문의 일부이며 행이 주석임을 의미하지는 않습니다. 해당 디렉터리의 파일 이름은 마침표를 포함하지 않아야 하며 틸드(~)로 끝나지 않아야 합니다.

추가 리소스

- ***sudo(8)** 및 **sudoers(5)** 매뉴얼 페이지*

10.2. 사용자에게 SUDO 액세스 권한 부여

*시스템 관리자는 루트가 아닌 사용자가 **sudo** 액세스 권한을 부여하여 관리 명령을 실행할 수 있도록 허용할 수 있습니다. **sudo** 명령은 **root** 사용자의 암호를 사용하지 않고 사용자에게 관리 액세스 권한을 제공합니다.*

*사용자가 관리 명령을 수행해야 하는 경우 **sudo** 를 사용하여 해당 명령 앞에 . 사용자에게 명령에 대한 권한 부여가 있는 경우 **root**인 것처럼 명령이 실행됩니다.*

다음과 같은 제한 사항에 유의하십시오.

- */etc/sudoers** 구성 파일에 나열된 사용자만 **sudo** 명령을 사용할 수 있습니다.*
- *명령은 **root** 쉘이 아닌 사용자의 쉘에서 실행됩니다.*

사전 요구 사항

- *시스템에 대한 **root** 액세스 권한이 있습니다.*

절차

1. ***root**로 **/etc/sudoers** 파일을 엽니다.*

```
# visudo
```

   */etc/sudoers** 파일은 **sudo** 명령으로 적용되는 정책을 정의합니다.*

2.

/etc/sudoers 파일에서 관리 wheel 그룹의 사용자에게 sudo 액세스 권한을 부여하는 행을 찾습니다.

```
## Allows people in group wheel to run all commands
%wheel        ALL=(ALL)       ALL
```

3. *%wheel 으로 시작하는 행이 숫자 기호(#)를 사용하여 주석 처리되지 않았는지 확인합니다.*

4. *변경 사항을 저장하고 편집기를 종료합니다.*

5. *관리 wheel 그룹에 sudo 액세스 권한을 부여하려는 사용자를 추가합니다.*

```
# usermod --append -G wheel <username>
```

& lt;username >을 사용자 이름으로 바꿉니다.

검증 단계

- *사용자가 관리 wheel 그룹에 있는지 확인합니다.*

```
# id <username>
uid=5000(<username>) gid=5000(<username>) groups=5000(<username>),10(wheel)
```

추가 리소스

- *sudo(8), visudo(8), sudoers(5) 매뉴얼 페이지*

10.3. 권한이 없는 사용자가 특정 명령을 실행하도록 활성화

관리자는 /etc/sudoers.d/ 디렉터리에 정책을 구성하여 권한이 없는 사용자가 특정 워크스테이션에서 특정 명령을 실행하도록 허용할 수 있습니다.

사전 요구 사항

- *시스템에 대한 root 액세스 권한이 있습니다.*

절차

1. ***root** 로서 **/etc/**: 아래에 새 **sudoers.d** 디렉토리를 만듭니다.*

```
# mkdir -p /etc/sudoers.d/
```

2. ***/etc/sudoers.d** 디렉토리에 새 파일을 만듭니다.*

```
# visudo -f /etc/sudoers.d/<filename>
```

파일이 자동으로 열립니다.

3. ***/etc/sudoers.d/ <filename >** 파일에 다음 행을 추가합니다.*

```
<username> <hostname.example.com> = (<run_as_user>:<run_as_group>)
<path/to/command>
```

- ***& lt;username >**을 사용자 이름으로 바꿉니다.*
- ***& lt;hostname.example.com& gt;**을 호스트의 **URL**로 바꿉니다.*
- ***(<run_as_user > : <run_as_group>)** 명령을 실행할 수 있는 사용자 또는 그룹으로 바꿉니다. 이 섹션을 생략하면 **< username>** 에서 **root**로 명령을 실행할 수 있습니다.*
- ***& lt;path/to/command& gt;**를 명령의 전체 절대 경로로 바꿉니다. 명령 경로 뒤에 해당 옵션을 추가하여 특정 옵션 및 인수가 있는 명령만 실행하도록 사용자를 제한할 수도 있습니다. 옵션을 지정하지 않으면 사용자는 모든 옵션과 함께 명령을 사용할 수 있습니다.*
- *한 줄에 동일한 호스트에서 두 개 이상의 명령을 허용하려면 쉼표로 구분한 뒤에 공백으로 구분하여 나열할 수 있습니다.*

*예를 들어 **user1** 이 **dnf** 를 실행하고 **host1.example.com** 에서 **reboot** 명령을 실행할 수 있도록 하려면 **user1 host1.example.com = /bin/dnf, /sbin/reboot** 를 입력합니다.*

4. *선택 사항: 사용자가 **sudo** 권한을 사용하려고 할 때마다 이메일 알림을 받으려면 파일에 다음 행을 추가합니다.*

```
Defaults    mail_always
Defaults    mailto="<email@example.com>"
```

5. 변경 사항을 저장하고 편집기를 종료합니다.

검증

1. 사용자가 **sudo** 권한으로 명령을 실행할 수 있는지 확인하려면 계정을 전환합니다.

```
# su <username> -
```

2. 사용자로 **sudo** 명령을 사용하여 명령을 입력합니다.

```
$ sudo <command>
[sudo] password for <username>:
```

사용자의 **sudo** 암호를 입력합니다.

3. 권한이 올바르게 구성된 경우 시스템은 명령 및 옵션 목록을 표시합니다. 예를 들어 **dnf** 명령을 사용하면 다음 출력이 표시됩니다.

```
...
usage: dnf [options] COMMAND
...
```

시스템이 오류 메시지 **<username>**이 **sudoers** 파일에 없는 경우. 이 문제는 **/etc/sudoers.d/** 의 **< username >**에 대한 파일인 것으로 보고됩니다.

시스템이 **< username >** 오류 메시지를 반환하는 경우 **< host.example.com>**에서 **sudo**를 실행할 수 없는 경우 구성이 올바르게 완료되지 않았습니다. **root**로 로그인하고 구성이 올바르게 수행되었는지 확인합니다.

시스템이 오류 메시지 **Sorry**를 반환하는 경우 **user < username >**은 **<path/to/command> ' ' <path/to/command>'**를 **< host.example.com >**에서 **root**로 실행할 수 없습니다.. 에서는 사용자의 규칙에 명령이 올바르게 정의되지 않습니다.

추가 리소스

- ***sudo(8), visudo(8), sudoers(5) 매뉴얼 페이지***

11장. 파일 시스템 권한 관리

파일 시스템 권한은 사용자 및 그룹 계정에서 파일의 내용을 읽고 수정 및 실행하고 디렉토리를 입력할 수 있는 기능을 제어합니다. 무단 액세스로부터 데이터를 보호하려면 권한을 신중하게 설정합니다.

11.1. 파일 권한 관리

모든 파일 또는 디렉터리에는 세 가지 수준의 소유권이 있습니다.

- 사용자 소유자(**u**).
- 그룹 소유자(**g**).
- 기타(**o**).

각 수준의 소유권에는 다음 권한이 할당될 수 있습니다.

- 읽기(**r**).
- 쓰기(**W**).
- 실행(**x**).

파일에 대한 실행 권한을 사용하면 해당 파일을 실행할 수 있습니다. 디렉터리에 대한 실행 권한을 사용하면 디렉터리의 콘텐츠에 액세스할 수 있지만 실행할 수는 없습니다.

새 파일 또는 디렉터리가 생성되면 기본 권한 집합이 자동으로 할당됩니다. 파일 또는 디렉터리에 대한 기본 권한은 다음 두 가지 요인을 기반으로 합니다.

- 기본 권한.

- *사용자 파일 생성 모드 마스크(**mask**)입니다.*

11.1.1. 기본 파일 권한

*새 파일이나 디렉터리가 생성될 때마다 기본 권한이 자동으로 할당됩니다. 파일 또는 디렉터리에 대한 기본 권한을 기호 또는 **8**진수 값으로 표시할 수 있습니다.*

권한	심볼릭 값	8진수 값
권한 없음	---	0
execute	--x	1
write	-w-	2
쓰기 및 실행	-wx	3
읽기	r--	4
읽기 및 실행	r-x	5
읽기 및 쓰기	rw-	6
읽기, 쓰기, 실행	rwx	7

*디렉터리에 대한 기본 권한은 **777 (drwxrwxrwx)**이며 모든 사용자에게 읽기, 쓰기, 실행 권한을 부여합니다. 즉, 디렉터리 소유자, 그룹 및 기타 사용자가 디렉터리의 콘텐츠를 나열하고, 디렉터리 내의 항목을 만들고, 삭제하고, 편집하고, 편집할 수 있습니다.*

디렉토리에 있는 개별 파일에는 디렉터리에 대한 무제한 액세스 권한이 있더라도 편집할 수 있는 자체 권한이 있을 수 있습니다.

*파일에 대한 기본 권한은 **666 (-rw-rw-rw-)**이며 모든 사용자에게 읽기 및 쓰기 권한을 부여합니다. 즉, 파일 소유자, 그룹 및 다른 사용자가 파일을 읽고 편집할 수 있습니다.*

> *예 **11.1.** 파일에 대한 권한*
>
> *파일에 다음 권한이 있는 경우:*

```
$ ls -l
-rwxrw----. 1 sysadmins sysadmins 2 Mar 2 08:43 file
```

- *- 파일이 있음을 나타냅니다.*
- *rwx 는 파일 소유자가 파일을 읽고, 쓰고, 실행할 수 있는 권한이 있음을 나타냅니다.*
- *RW- 는 그룹에 읽기 및 쓰기 권한이 있지만 파일을 실행하지는 않음을 나타냅니다.*
- *--- 는 다른 사용자가 파일을 읽고, 쓰고, 실행할 수 있는 권한이 없음을 나타냅니다.*
- *. 는 SELinux 보안 컨텍스트가 파일에 설정되어 있음을 나타냅니다.*

예 11.2. 디렉터리에 대한 권한

디렉터리에 다음 권한이 있는 경우:

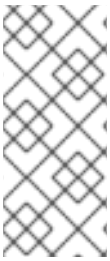
```
$ ls -dl directory
drwxr-----. 1 sysadmins sysadmins 2 Mar 2 08:43 directory
```

- *D 는 디렉터리임을 나타냅니다.*
- *rwx 는 디렉터리 소유자가 디렉터리의 콘텐츠를 읽고, 쓰고, 액세스할 수 있는 권한이 있음을 나타냅니다.*

  디렉터리 소유자는 디렉터리 내의 항목(파일, 하위 디렉터리)을 나열하고 해당 항목의 콘텐츠에 액세스한 다음 수정할 수 있습니다.
- *R -x 는 그룹에 디렉터리의 내용을 읽을 수 있는 권한이 있지만 쓸 수는 없음을 나타냅니다. 새 항목을 생성하거나 파일을 삭제할 수 없습니다. x 권한은 cd 명령을 사용하여 디렉터리에 액세스할 수도 있음을 의미합니다.*
- *--- 는 다른 사용자가 디렉터리의 콘텐츠를 읽고, 쓰거나, 액세스할 수 있는 권한이 없음을 나타냅니다.*

사용자 소유자가 아니거나 디렉토리의 그룹 소유자로서는 디렉터리 내의 항목을 나열하거나, 해당 항목에 대한 정보에 액세스하거나, 수정할 수 없습니다.

- . 는 SELinux 보안 컨텍스트가 디렉터리에 설정되어 있음을 나타냅니다.



참고

파일 또는 디렉터리에 자동으로 할당된 기본 권한은 파일 또는 디렉터리가 끝나는 기본 권한이 아닙니다. 파일 또는 디렉토리를 생성하면 기본 권한이 permissions에 의해 변경됩니다. 기본 권한과 mTLS의 조합은 파일 및 디렉터리에 대한 기본 권한을 생성합니다.

11.1.2. 사용자 파일 생성 모드 마스크

사용자 파일 생성 모드 마스크(balancer)는 새로 생성된 파일 및 디렉터리에 대해 파일 권한이 설정되는 방법을 제어하는 변수입니다. CloudEvent 는 Linux 시스템의 전체 보안을 높이기 위해 기본 권한 값에서 권한을 자동으로 제거합니다. jaeger 는 심볼릭 또는 8진수 값으로 표현할 수 있습니다.

권한	심볼릭 값	8진수 값
읽기, 쓰기 및 실행	rwx	0
읽기 및 쓰기	rw-	1
읽기 및 실행	r-x	2
읽기	r--	3
쓰기 및 실행	-wx	4
write	-w-	5
execute	--x	6
권한 없음	---	7

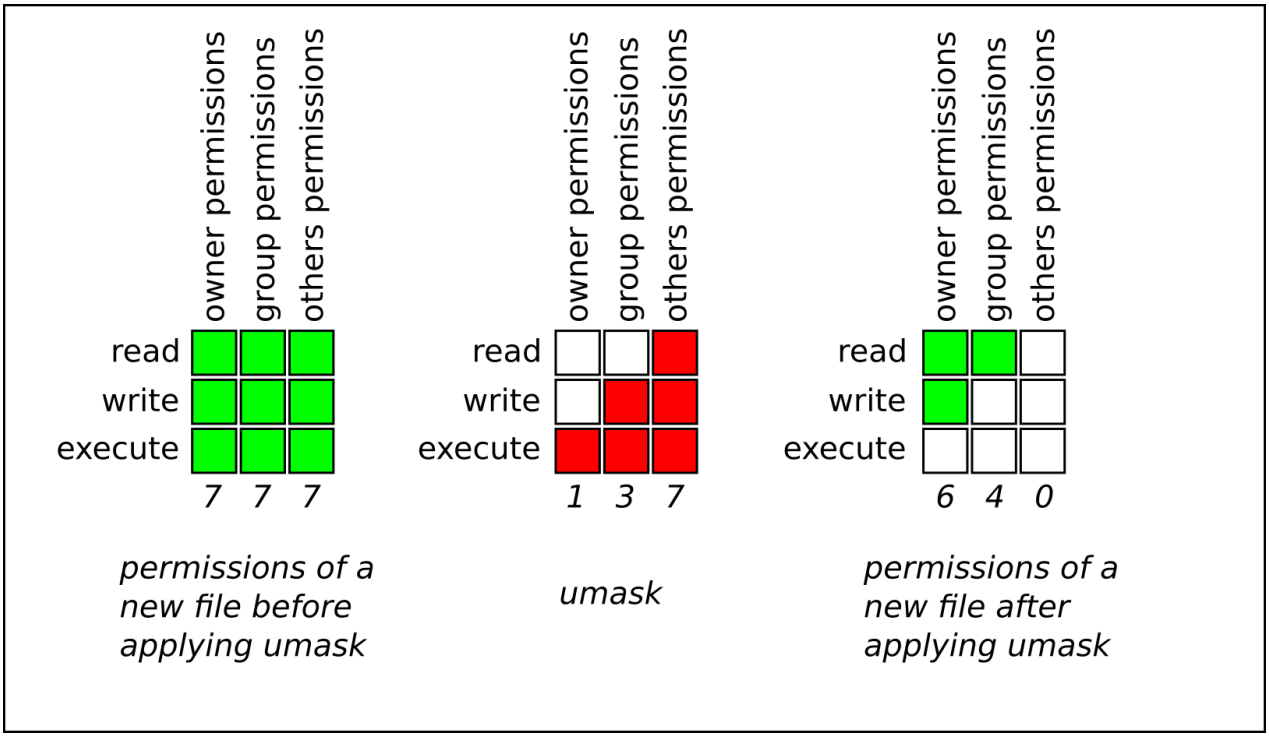
표준 사용자와 root 사용자의 기본 CloudEvent는 0022 입니다.

messages의 첫 번째 숫자는 특수 권한(sticky bit,)을 나타냅니다. PATH의 마지막 세 자리는 사용자

소유자(u), 그룹 소유자(g) 및 기타(o)에서 각각 제거된 권한을 나타냅니다.

예 11.3. 파일을 생성할 때 mTLS 적용

다음 예제에서는 기본 권한이 777 인 파일에 8진수 값이 0137 인 umask 를 적용하여 기본 권한이 640 인 파일을 생성하는 방법을 보여줍니다.



11.1.3. 기본 파일 권한

새로 생성된 모든 파일 및 디렉터리에 대해 기본 권한이 자동으로 설정됩니다. 기본 권한의 값은 기본 권한에 umask 를 적용하여 결정됩니다.

예 11.4. 디렉터리에 대한 기본 권한

표준 사용자 또는 루트 사용자가 새 디렉터리를 생성하는 경우 ScanSetting은 022 (rwxr-xr-x)로 설정되고 디렉터리에 대한 기본 권한이 777 (rwxrwxrwx)으로 설정됩니다. 기본 권한을 755 (rwxr-xr-x)로 가져옵니다.

	심볼릭 값	8진수 값
기본 권한	rwxrwxrwx	777
jaeger	rwxr-xr-x	022

기본 권한	rwxr-xr-x	755

즉, 디렉터리 소유자는 디렉터리의 콘텐츠를 나열하고, 디렉터리 내의 항목을 생성, 삭제, 편집할 수 있으며, 디렉터리의 콘텐츠를 내림차순할 수 있습니다. 그룹 및 기타 그룹은 디렉터리의 콘텐츠만 나열하고 그 내용을 추측할 수 있습니다.

***예 11.5.** 파일에 대한 기본 권한*

*표준 사용자 또는 루트 사용자가 새 파일을 생성하는 경우 **ScanSetting**은 **022 (rwxr-xr-x)**로 설정되고 파일의 기본 권한은 **666 (rw-rw-rw-)**으로 설정됩니다. **644 (-rw-r-**의 기본 권한을 가져옵니다.**r--).***

	심볼릭 값	8진수 값
기본 권한	rw-rw-rw-	666
jaeger	rwxr-xr-x	022
기본 권한	rw-r—r--	644

즉, 파일 소유자는 파일을 읽고 편집할 수 있으며, 그룹 및 다른 사용자는 파일을 읽을 수 있습니다.

참고

*보안상의 이유로 **permissions**가 **000 (rwxrwx)**으로 설정되어 있어도 일반 파일은 기본적으로 실행 권한을 가질 수 없습니다. 그러나 실행 권한을 사용하여 디렉터리를 만들 수 있습니다.*

11.1.4. 심볼릭 값을 사용하여 파일 권한 변경

***tekton** 유틸리티를 심볼릭 값(조합 문자 및 기호)과 함께 사용하여 파일 또는 디렉토리에 대한 파일 권한을 변경할 수 있습니다.*

다음 권한을 할당할 수 있습니다.

- *읽기(r)*
- *쓰기(W)*
- *실행 (x)*

권한은 다음 수준의 소유권 에 할당할 수 있습니다.

- *사용자 소유자 (u)*
- *그룹 소유자(g)*
- *기타 (O)*
- *모두 (a)*

사용 권한 추가 또는 제거 하려면 다음 표시를 사용 합니다.To add or remove permissions you can use the following signs:

- *+ 기존 권한 상단에 대한 권한을 추가하려면 다음을 수행합니다.*
- *기존 권한에서 권한을 제거하려면To remove the permissions from the existing permission*
- *기존 권한을 제거하고 새 권한을 명시적으로 정의하려면 =*

절차

- *파일 또는 디렉터리에 대한 권한을 변경하려면 다음을 사용합니다.*

```
$ chmod <level><operation><permission> file-name
```

-

<level> 을 권한을 설정하려는 소유권 수준으로 바꿉니다. <operation> 를 부호 중 하나로 바꿉니다. <permission> 를 할당할 권한으로 바꿉니다. file-name 을 파일 또는 디렉터리의 이름으로 바꿉니다. 예를 들어 모든 사용자에게 읽기, 쓰기, 실행(rwx) my-script.sh 를 부여하려면 etcdctl a=rwx my-script.sh 명령을 사용합니다.

자세한 내용은 기본 파일 권한을 참조하십시오.

검증 단계

- *특정 파일에 대한 권한을 보려면 다음을 사용합니다.*

```
$ ls -l file-name
```

file-name 을 파일 이름으로 바꿉니다.

- *특정 디렉터리에 대한 권한을 보려면 다음을 사용합니다.*

```
$ ls -dl directory-name
```

directory-name 을 디렉터리 이름으로 바꿉니다.

- *특정 디렉터리 내의 모든 파일에 대한 권한을 보려면 다음을 사용합니다.*

```
$ ls -l directory-name
```

directory-name 을 디렉터리 이름으로 바꿉니다.

예 11.6. 파일 및 디렉터리에 대한 권한 변경

- *my-file.txt 의 파일 권한을 -rw-rw-r--- 에서 -rw----- 으로 변경하려면 다음을 사용합니다.*

1. *my-file.txt 에 대한 현재 권한을 표시합니다.*

```
$ ls -l my-file.txt
-rw-rw-r--. 1 username username 0 Feb 24 17:56 my-file.txt
```

2. 그룹 소유자(g) 및 기타(o)에서 파일을 읽고 쓰고 실행할 수 있는 권한을 제거합니다.

```
$ chmod go= my-file.txt
```

등호(=) 후에 지정되지 않은 권한은 자동으로 금지됩니다.

3. my-file.txt 에 대한 권한이 올바르게 설정되었는지 확인합니다.

```
$ ls -l my-file.txt
-rw-------. 1 username username 0 Feb 24 17:56 my-file.txt
```

- my-directory 의 파일 권한을 drwxrwx--- 에서 drwxrwxr-x 로 변경하려면 다음을 사용합니다.

1. my-directory 에 대한 현재 권한을 표시합니다.

```
$ ls -dl my-directory
drwxrwx---. 2 username username 4096 Feb 24 18:12 my-directory
```

2. 모든 사용자(a)에 대해 읽기 및 실행(r-x) 액세스를 추가합니다.

```
$ chmod o+rx my-directory
```

3. my-directory 및 해당 콘텐츠에 대한 권한이 올바르게 설정되었는지 확인합니다.

```
$ ls -dl my-directory
drwxrwxr-x. 2 username username 4096 Feb 24 18:12 my-directory
```

11.1.5. 8진수 값을 사용하여 파일 권한 변경

8진수 값(numbers)과 함께 tekton 유틸리티를 사용하여 파일 또는 디렉터리에 대한 파일 권한을 변경할 수 있습니다.

절차

- 기존 파일 또는 디렉터리의 파일 권한을 변경하려면 다음을 사용합니다.

```
$ chmod octal_value file-name
```

file-name 을 파일 또는 디렉터리의 이름으로 바꿉니다. 8진수_value 를 8진수 값으로 바꿉니다. 자세한 내용은 기본 파일 권한을 참조하십시오.

11.2. 액세스 제어 목록 관리

각 파일과 디렉토리는 한 번에 하나의 사용자 소유자와 그룹 소유자만 가질 수 있습니다. 다른 파일 및 디렉터리를 비공개로 유지하면서 다른 사용자 또는 그룹에 속하는 특정 파일 또는 디렉터리에 액세스할 수 있는 권한을 사용자에게 부여하려면 Linux ACL(액세스 제어 목록)을 사용할 수 있습니다.

11.2.1. 현재 액세스 제어 목록 표시

getfacl 유틸리티를 사용하여 현재 ACL을 표시할 수 있습니다.

절차

- 특정 파일 또는 디렉터리에 대한 현재 ACL을 표시하려면 다음을 사용합니다.

```
$ getfacl file-name
```

file-name 을 파일 또는 디렉터리의 이름으로 바꿉니다.

11.2.2. 액세스 제어 목록 설정

setfacl 유틸리티를 사용하여 파일 또는 디렉터리에 대한 ACL을 설정할 수 있습니다.

사전 요구 사항

- 루트 액세스입니다.

절차

- 파일 또는 디렉터리에 대해 **ACL**을 설정하려면 다음을 사용합니다.

```
# setfacl -m u:username:symbolic_value file-name
```

username 을 사용자 이름으로, **symbolic_value** 를 심볼릭 값으로 바꾸고 **file-name** 을 파일 또는 디렉터리 이름으로 바꿉니다. 자세한 내용은 **setfacl man** 페이지를 참조하십시오.

예 11.7. 그룹 프로젝트에 대한 권한 수정

다음 예제에서는 이 파일이 **root** 그룹에 속하는 **root** 사용자가 소유한 **group-project** 파일에 대한 권한을 수정하는 방법을 설명합니다.

- 모든 사람이 실행할 수 없습니다.
- 사용자 및 **re w**에는 **rw-** 권한이 있습니다.
- 사용자 **susan** 에는 **---** 권한이 있습니다.
- 다른 사용자에게는 **r--** 권한이 있습니다.

절차

```
# setfacl -m u:andrew:rw- group-project
# setfacl -m u:susan:--- group-project
```

검증 단계

- 사용자 및**rew**에 **rw-** 권한이 있는지 확인하려면 사용자 **susan** 에 **---** 권한이 있으며 기타 사용자에게는 **r--** 권한이 있습니다.

```
$ getfacl group-project
```

출력이 반환됩니다.

```
# file: group-project
# owner: root
# group: root
user:andrew:rw-
user:susan:---
group::r--
mask::rw-
other::r--
```

11.3. RECEIVER 관리

umask 유틸리티를 사용하여 **umask** 의 현재 또는 기본값을 표시, 설정 또는 변경할 수 있습니다.

11.3.1. mTLS의 현재 값 표시

dependencies 유틸리티를 사용하여 현재 데이터 값을 심볼릭 또는 **8**진수 모드로 표시할 수 있습니다.

절차

- **symbolic mode**로 **topology**의 현재 값을 표시하려면 다음을 사용합니다.

```
$ umask -S
```

- **8**진수 모드에서 **topology**의 현재 값을 표시하려면 다음을 사용합니다.

```
$ umask
```

참고

8진수 모드로 **umask** 를 표시할 때 **4**자리 숫자(**0002** 또는 **0022**)로 표시될 수 있습니다. **messages**의 첫 번째 숫자는 특수 비트 (**sticky bit, SGID** 비트 또는 **SUID** 비트)를 나타냅니다. 첫 번째 숫자가 **0** 으로 설정되면 특수 비트가 설정되지 않습니다.

11.3.2. 기본 bash credentials 표시

bash,ksh,zsh 및 **tcsh**와 같이 사용할 수 있는 셸은 여러 가지가 있습니다. 이러한 셸은 로그인 또는 비로그인 셸로 작동할 수 있습니다. 네이티브 또는 **GUI** 터미널을 열어 로그인 셸을 호출할 수 있습니다.

로그인 또는 비로그인 셸에서 명령을 실행 중인지 확인하려면 **echo $0** 명령을 사용합니다.

예 11.8. 로그인 또는 비로그인 **bash** 셸에서 작동하는지 확인

- **echo $0** 명령의 출력이 **bash** 를 반환하면 비로그인 셸에서 명령을 실행하고 있습니다.

```
$ echo $0
bash
```

로그인이 아닌 셸의 기본 **umask** 는 **/etc/bashrc** 구성 파일에 설정됩니다.

- **echo $0** 명령의 출력이 **-bash** 를 반환하는 경우 로그인 셸에서 명령을 실행합니다.

```
# echo $0
-bash
```

로그인 셸의 기본 **umask** 는 **/etc/login.defs** 구성 파일에 설정되어 있습니다.

절차

- 로그인이 아닌 셸에 대한 기본 **bash umask** 를 표시하려면 다음을 사용합니다.

```
$ grep umask /etc/bashrc
```

출력이 반환됩니다.

```
# By default, we want umask to get set. This sets it for non-login shell.
    umask 002
    umask 022
```

- 로그인 셸에 대한 기본 **bash umask** 를 표시하려면 다음을 사용합니다.

```
$ grep "UMASK" /etc/login.defs
```

출력이 반환됩니다.

```
# UMASK is also used by useradd(8) and newusers(8) to set the mode for new
UMASK        022
# If HOME_MODE is not set, the value of UMASK is used to create the mode.
```

11.3.3. 심볼릭 값을 사용하여 mTLS 설정

***symlink** 유틸리티의 심볼릭 값(스크립 문자 및 기호)을 사용하여 현재 쉘 세션에 대한 **umask** 를 설정할 수 있습니다.*

다음 권한을 할당할 수 있습니다.

- *읽기(**r**)*
- *쓰기(**W**)*
- *실행 (**x**)*

권한은 다음 수준의 소유권 에 할당할 수 있습니다.

- *사용자 소유자 (**u**)*
- *그룹 소유자(**g**)*
- *기타 (**O**)*
- *모두 (**a**)*

*사용 권한 추가 또는 제거 하려면 다음 표시를 사용 합니다.**To add or remove permissions you can use the following signs:***

- *+ 기존 권한 상단에 대한 권한을 추가하려면 다음을 수행합니다.*

- *기존 권한에서 권한을 제거하려면To remove the permissions from the existing permission*

- *기존 권한을 제거하고 새 권한을 명시적으로 정의하려면 =*

참고

등호(=) 이후에 지정되지 않은 모든 권한은 자동으로 금지됩니다.

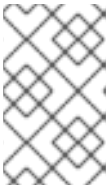
절차

- *현재 쉘 세션에 대해 mTLS를 설정하려면 다음을 사용합니다.*

```
$ umask -S <level><operation><permission>
```

<level> 을 에 대해 설정하려는 소유권 레벨로 바꿉니다. <operation> 를 부호 중 하나로 바꿉니다. <permission> 를 할당할 권한으로 바꿉니다. 예를 들어, permissions를 u=rwx,g=rwx,o=rwx 으로 설정하려면 umask -S a=rwx 를 사용합니다.

자세한 내용은 사용자 파일 생성 모드를 참조하십시오.



참고

jaeger 는 현재 쉘 세션에만 유효합니다.

11.3.4. 8진수 값을 사용하여 receiver 설정

8진수 값(number)과 함께 wizard 유틸리티를 사용하여 현재 쉘 세션에 대한 umask 를 설정할 수 있습니다.

절차

- 현재 셸 세션에 대해 **mTLS**를 설정하려면 다음을 사용합니다.

```
$ umask octal_value
```

8진수_**value** 를 **8**진수 값으로 바꿉니다. 자세한 내용은 **User file-creation mode mask** 를 참조하십시오.

참고

jaeger 는 현재 셸 세션에만 유효합니다.

11.3.5. 로그인이 아닌 셸의 기본 mTLS 변경

/etc/bashrc 파일을 수정하여 표준 사용자의 기본 **bash permissions**를 변경할 수 있습니다.

사전 요구 사항

- 루트 액세스

절차

1. 루트 로서 편집기에서 **/etc/bashrc** 파일을 엽니다.
2. 다음 섹션을 수정하여 새로운 기본 **bash umask** 를 설정합니다.

```
if [ $UID -gt 199 ] && [ “id -gn” = “id -un” ]; then
   umask 002
else
   umask 022
fi
```

errata (002)의 기본 **8**진수 값을 다른 **8**진수 값으로 바꿉니다. 자세한 내용은 **User file-creation mode mask** 를 참조하십시오.

3. 변경 사항을 저장하고 편집기를 종료합니다.

11.3.6. 로그인 쉘의 기본 mTLS 변경

/etc/login.defs 파일을 수정하여 root 사용자의 기본 bash umask 를 변경할 수 있습니다.

사전 요구 사항

- 루트 액세스

절차

1. 루트 권한으로 편집기에서 /etc/login.defs 파일을 엽니다.
2. 다음 섹션을 수정하여 새로운 기본 bash umask 를 설정합니다.

```
# Default initial "umask" value used by login(1) on non-PAM enabled systems.
# Default "umask" value for pam_umask(8) on PAM enabled systems.
# UMASK is also used by useradd(8) and newusers(8) to set the mode for new
# home directories if HOME_MODE is not set.
# 022 is the default value, but 027, or even 077, could be considered
# for increased privacy. There is no One True Answer here: each sysadmin
# must make up their mind.

UMASK           022
```

umask (022)의 기본 8진수 값을 다른 8진수 값으로 교체합니다. 자세한 내용은 User file-creation mode mask 를 참조하십시오.

3. 변경 사항을 저장하고 편집기를 종료합니다.

11.3.7. 특정 사용자의 기본 mTLS 변경

해당 사용자의 . bashrc 를 수정하여 특정 사용자의 기본 mTLS를 변경할 수 있습니다.

절차

- 특정 사용자의 .bashrc 파일에") 의 8진수 값을 지정하는 행을 추가합니다.

```
$ echo 'umask octal_value' >> /home/username/.bashrc
```

8진수_value 를 8진수 값으로 바꾸고 username 을 사용자 이름으로 교체합니다. 자세한 내용은 User file-creation mode mask 를 참조하십시오.

11.3.8. 새로 생성된 홈 디렉터리에 대한 기본 권한 설정

/etc/login.defs 파일을 수정하여 새로 생성된 사용자의 홈 디렉토리에 대한 권한 모드를 변경할 수 있습니다.

절차

1. *루트 권한으로 편집기에서 /etc/login.defs 파일을 엽니다.*

2. *다음 섹션을 수정하여 새로운 기본 HOME_MODE 를 설정합니다.*

```
# HOME_MODE is used by useradd(8) and newusers(8) to set the mode for new
# home directories.
# If HOME_MODE is not set, the value of UMASK is used to create the mode.
HOME_MODE       0700
```

기본 8진수 값(0700)을 다른 8진수 값으로 교체합니다. 선택한 모드는 홈 디렉토리에 대한 권한을 생성하는 데 사용됩니다.

3. *HOME_MODE 가 설정되어 있으면 변경 사항을 저장하고 편집기를 종료합니다.*

4. *HOME_MODE 가 설정되지 않은 경우 UMASK 를 수정하여 새로 생성된 홈 디렉터리의 모드를 설정합니다.*

```
# Default initial "umask" value used by login(1) on non-PAM enabled systems.
# Default "umask" value for pam_umask(8) on PAM enabled systems.
# UMASK is also used by useradd(8) and newusers(8) to set the mode for new
# home directories if HOME_MODE is not set.
# 022 is the default value, but 027, or even 077, could be considered
# for increased privacy. There is no One True Answer here: each sysadmin
# must make up their mind.

UMASK           022
```

기본 8진수 값(022)을 다른 8진수 값으로 바꿉니다. 자세한 내용은 User file-creation mode mask 를 참조하십시오.

5. *변경 사항을 저장하고 편집기를 종료합니다.*

12장. SYSTEMD 관리

시스템 관리자는 **systemd** 를 사용하여 시스템의 중요한 측면을 관리할 수 있습니다. **Linux** 운영 체제의 시스템 및 서비스 관리자 역할을 하는 **systemd** 소프트웨어 제품군은 제어, 보고 및 시스템 초기화를 위한 툴과 서비스를 제공합니다. **systemd** 의 주요 기능은 다음과 같습니다.

- 부팅 중 시스템 서비스의 병렬 시작
- 데몬의 온 디맨드 활성화
- 종속성 기반 서비스 제어 논리

systemd 가 관리하는 기본 오브젝트는 시스템 리소스 및 서비스를 나타내는 **systemd** 장치입니다. **systemd** 장치는 특정 작업을 정의하고 관리하는 이름, 유형 및 구성 파일로 구성됩니다. 단위 파일을 사용하여 시스템 동작을 구성할 수 있습니다. 다양한 **systemd** 장치 유형의 다음 예제를 참조하십시오.

Service

개별 시스템 서비스를 제어 및 관리합니다.

대상

시스템 상태를 정의하는 단위 그룹을 나타냅니다.

장치

하드웨어 장치 및 가용성을 관리합니다.

Mount

파일 시스템 마운트를 처리합니다.

타이머

특정 간격으로 실행되도록 작업을 예약합니다.

참고

사용 가능한 유닛 유형을 모두 표시하려면 다음을 수행합니다.

```
# systemctl -t help
```

12.1. SYSTEMD 장치 파일 위치

다음 디렉토리 중 하나에서 단위 구성 파일을 찾을 수 있습니다.

표 12.1. Systemd 유닛 파일 위치

디렉터리	설명
/usr/lib/systemd/system/	설치된 RPM 패키지와 함께 배포된 **systemd** 장치 파일.
/run/systemd/system/	런타임에 생성된 **systemd** 장치 파일입니다. 이 디렉터리는 설치된 서비스 장치 파일이 있는 디렉터리보다 우선합니다.
/etc/systemd/system/	**systemctl enable** 명령과 서비스 확장을 위해 추가된 유닛 파일을 사용하여 생성된 **systemd** 장치 파일입니다. 이 디렉터리는 런타임 장치 파일이 있는 디렉터리보다 우선합니다.

systemd 의 기본 구성은 컴파일 중에 정의되며 **/etc/systemd/system.conf** 파일에서 구성을 찾을 수 있습니다. 이 파일을 편집하여 전역적으로 **systemd** 단위의 값을 재정의하여 기본 구성을 수정할 수 있습니다.

예를 들어 **90**초로 설정된 제한 시간 제한의 기본값을 재정의하려면 **DefaultTimeoutStartSec** 매개변수를 사용하여 필요한 값을 초 단위로 입력합니다.

```
DefaultTimeoutStartSec=required value
```

12.2. SYSTEMCTL을 사용하여 시스템 서비스 관리

시스템 관리자는 **systemctl** 유틸리티를 사용하여 시스템 서비스를 관리할 수 있습니다. 실행 중인 서비스 시작, 중지, 다시 시작, 서비스 활성화 및 비활성화, 사용 가능한 서비스 나열, 시스템 서비스 상태 표시 등 다양한 작업을 수행할 수 있습니다.

12.2.1. 시스템 서비스 나열

현재 로드된 모든 서비스 단위를 나열하고 사용 가능한 모든 서비스 단위의 상태를 표시할 수 있습니다.

절차

systemctl 명령을 사용하여 다음 작업을 수행합니다.

- 현재 로드된 모든 서비스 단위를 나열합니다.

```
$ systemctl list-units --type service
UNIT                    LOAD   ACTIVE SUB     DESCRIPTION
abrt-ccpp.service       loaded active exited  Install ABRT coredump hook
abrt-oops.service       loaded active running ABRT kernel log watcher
abrtd.service           loaded active running ABRT Automated Bug Reporting Tool
...
systemd-vconsole-setup.service loaded active exited  Setup Virtual Console
tog-pegasus.service     loaded active running OpenPegasus CIM Server

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, or a generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

46 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'
```

기본적으로 **systemctl list-units** 명령은 활성 유닛만 표시합니다. 각 서비스 단위 파일에 대해 명령은 다음 매개변수에 대한 개요를 제공합니다.

단위

서비스 유닛의 전체 이름

LOAD

구성 파일의 로드 상태

ACTIVE 또는 **SUB**

현재 고급 및 낮은 수준의 장치 파일 활성화 상태

DESCRIPTION

유닛의 목적과 기능에 대한 간략한 설명

- **all** 또는 **-a** 명령줄 옵션과 함께 다음 명령을 사용하여 상태에 관계없이 로드된 모든 유닛 을 나열합니다.

```
$ systemctl list-units --type service --all
```

- 사용 가능한 모든 서비스 단위의 상태(활성화 또는 비활성화)를 나열합니다.

```
$ systemctl list-unit-files --type service
UNIT FILE                              STATE
abrt-ccpp.service                        enabled
abrt-oops.service                        enabled
abrtd.service                          enabled
...
wpa_supplicant.service                       disabled
ypbind.service                         disabled

208 unit files listed.
```

각 서비스 유닛에 대해 이 명령은 다음을 표시합니다.

단위 파일

서비스 유닛의 전체 이름

상태

부팅 중에 서비스 단위가 활성화되거나 비활성화되었는지 여부에 대한 정보

추가 리소스

- 시스템 서비스 상태 표시

12.2.2. 시스템 서비스 상태 표시

서비스 단위를 검사하여 자세한 정보를 가져오고 부팅 중에 시작되도록 활성화되었는지 또는 현재 실행 중인 서비스 상태인지 확인할 수 있습니다. 또한 특정 서비스 유닛 이후 또는 이전에 주문된 서비스를 볼 수도 있습니다.

절차

systemctl 명령을 사용하여 다음 작업을 수행합니다.

- *시스템 서비스에 해당하는 서비스 유닛에 대한 자세한 정보를 표시합니다.*

  $ systemctl status <name>.service

  & lt;name >을 검사하려는 서비스 단위의 이름으로 바꿉니다(예: gdm).

  이 명령은 다음 정보를 표시합니다.

  - *선택한 서비스 단위의 이름 뒤에 간단한 설명*
  - *사용 가능한 서비스 단위 정보에설명된 하나 이상의 필드*
  - *서비스 유닛의 실행: 루트 사용자가 유닛을 실행하는 경우*
  - *최신 로그 항목*

  표 12.2. 사용 가능한 서비스 단위 정보

필드	설명
loaded	서비스 유닛이 로드되었는지 여부, 단위 파일의 절대 경로, 부팅 중에 장치를 시작할 수 있는지 여부를 확인합니다.
active	서비스 유닛이 실행 중인지 여부 및 타임스탬프를 제공합니다.
Main PID	프로세스 ID 및 해당 시스템 서비스의 이름입니다.
상태	해당 시스템 서비스에 대한 추가 정보입니다.
process	관련 프로세스에 대한 추가 정보.
cgroup	관련 제어 그룹(**cgroup**)에 대한 추가 정보.

  예 12.1. 서비스 상태 표시

*GNOME Display Manager의 서비스 단위는 **gdm.service** 라고 합니다. 이 서비스 장치의 현재 상태를 확인하려면 셸 프롬프트에서 다음을 입력합니다.*

```
# systemctl status gdm.service
gdm.service - GNOME Display Manager
   Loaded: loaded (/usr/lib/systemd/system/gdm.service; enabled)
   Active: active (running) since Thu 2013-10-17 17:31:23 CEST; 5min ago
 Main PID: 1029 (gdm)
   CGroup: /system.slice/gdm.service
           ├─1029 /usr/sbin/gdm
           └─1047 /usr/bin/Xorg :0 -background none -verbose -auth /r...

Oct 17 17:31:23 localhost systemd[1]: Started GNOME Display Manager.
```

- *특정 서비스 장치가 실행 중인지 확인합니다.*

```
$ systemctl is-active <name>.service
```

- *부팅 중에 특정 서비스 단위가 시작되도록 활성화되어 있는지 확인합니다.*

```
$ systemctl is-enabled <name>.service
```

참고

*지정된 서비스 장치가 실행 중이거나 활성화된 경우 **systemctl is-active** 및 **systemctl is-enabled** 명령은 종료 상태 **0** 을 반환합니다.*

- *지정된 서비스 단위 전에 **systemd** 주문을 시작할 서비스 확인*

```
# systemctl list-dependencies --after <name>.service
```

*예를 들어 **gdm** 이전에 시작되도록 순서가 지정된 서비스 목록을 보려면 다음을 입력합니다.*

```
# systemctl list-dependencies --after gdm.service
gdm.service
├─dbus.socket
├─getty@tty1.service
├─livesys.service
├─plymouth-quit.service
├─system.slice
├─systemd-journald.socket
```

```
├─systemd-user-sessions.service
└─basic.target
[output truncated]
```

- 지정된 서비스 단위 후에 시작할 서비스 **systemd** 주문을 확인합니다.

```
# systemctl list-dependencies --before <name>.service
```

예를 들어 **gdm** 후에 시작할 서비스 **systemd** 주문 목록을 보려면 다음을 입력합니다.

```
# systemctl list-dependencies --before gdm.service
gdm.service
├─dracut-shutdown.service
├─graphical.target
│ ├─systemd-readahead-done.service
│ ├─systemd-readahead-done.timer
│ └─systemd-update-utmp-runlevel.service
└─shutdown.target
  ├─systemd-reboot.service
  └─final.target
    └─systemd-reboot.service
```

추가 리소스

- 시스템 서비스 나열

12.2.3. 시스템 서비스 시작

start 명령을 사용하여 현재 세션에서 시스템 서비스를 시작할 수 있습니다.

사전 요구 사항

- 루트 액세스

절차

- 현재 세션에서 시스템 서비스를 시작합니다.

```
# systemctl start <name>.service
```

& lt;name >을 시작하려는 서비스 단위의 이름으로 바꿉니다(예: httpd.service).

참고

systemd 에서 서비스 간 양수 및 음수 종속성이 있습니다. 특정 서비스를 시작하려면 하나 이상의 다른 서비스(긍정 종속성)를 시작하거나 하나 이상의 서비스(negative dependency)를 중지해야 할 수 있습니다.

새 서비스를 시작하려고 하면 systemd 는 사용자에게 명시적 통지 없이 모든 종속 항목을 자동으로 해결합니다. 즉, 이미 서비스를 실행하고 음수 종속성으로 다른 서비스를 시작하려고 하면 첫 번째 서비스가 자동으로 중지됩니다.

예를 들어 postfix 서비스를 실행 중이고 sendmail 서비스를 시작하려고 하는 경우 systemd 는 먼저 postfix 를 자동으로 중지합니다. 이 두 서비스는 충돌하고 동일한 포트에서 실행할 수 없기 때문입니다.

추가 리소스

- *systemctl(1) 도움말 페이지*
- *부팅 시 시스템 서비스가 시작되도록 활성화*
- *시스템 서비스 상태 표시*

12.2.4. 시스템 서비스 중지

현재 세션에서 시스템 서비스를 중지하려면 stop 명령을 사용합니다.

사전 요구 사항

- *루트 액세스*

절차

- *시스템 서비스를 중지합니다.*

```
# systemctl stop <name>.service
```

& lt;name >을 중지하려는 서비스 단위의 이름으로 바꿉니다(예: bluetooth).

추가 리소스

- systemctl(1) 도움말 페이지
- 부팅할 때 시작할 시스템 서비스 비활성화
- 시스템 서비스 상태 표시

12.2.5. 시스템 서비스 재시작

restart 명령을 사용하여 현재 세션에서 시스템 서비스를 다시 시작하여 다음 작업을 수행할 수 있습니다.

- 현재 세션에서 선택한 서비스 장치를 중지하고 즉시 다시 시작합니다.
- 해당 서비스가 이미 실행 중인 경우에만 서비스 장치를 다시 시작합니다.
- 실행을 중단하지 않고 시스템 서비스의 구성을 다시 로드합니다.

사전 요구 사항

- 루트 액세스

절차

- 시스템 서비스를 다시 시작하십시오.

```
# systemctl restart <name>.service
```

& lt;name >을 재시작할 서비스 단위의 이름으로 바꿉니다(예: httpd).

참고

선택한 서비스 장치가 실행되고 있지 않으면 이 명령도 시작합니다.

- *선택 사항: 해당 서비스가 이미 실행 중인 경우에만 서비스 장치를 다시 시작하십시오.*

```
# systemctl try-restart <name>.service
```

- *선택 사항: 서비스 실행을 중단하지 않고 구성을 다시 로드합니다.*

```
# systemctl reload <name>.service
```

참고

*이 기능을 지원하지 않는 시스템 서비스는 이 명령을 무시합니다. 이러한 서비스를 다시 시작하려면 대신 **reload-or-restart** 및 **reload-or-try-restart** 명령을 사용합니다.*

추가 리소스

- ***systemctl** 도움말 페이지*
- *시스템 서비스 상태 표시*

12.2.6. 부팅 시 시스템 서비스가 시작되도록 활성화

부팅 시 서비스가 자동으로 시작되도록 활성화할 수 있습니다. 이러한 변경 사항은 다음 재부팅 시 적용됩니다.

사전 요구 사항

- *루트 액세스*

- 사용하려는 서비스는 마스킹할 수 없습니다. 마스킹된 서비스가 있는 경우 먼저 마스킹을 해제합니다.

```
# systemctl unmask <name>.service
```

절차

- 부팅 시 시작할 서비스를 활성화합니다.

```
# systemctl enable <name>.service
```

& lt;name >을 활성화하려는 서비스 단위의 이름으로 바꿉니다(예: httpd).

- 선택 사항: 단일 명령을 사용하여 서비스를 활성화하고 시작할 수도 있습니다.

```
# systemctl enable --now <name>.service
```

추가 리소스

- systemctl(1) 도움말 페이지
- 시스템 서비스 상태 표시
- 시스템 서비스 시작

12.2.7. 부팅할 때 시작할 시스템 서비스 비활성화

부팅 시 서비스 장치가 자동으로 시작되지 않도록 할 수 있습니다. 서비스를 비활성화하면 부팅 시 시작되지 않지만 수동으로 시작할 수 있습니다. 수동으로 시작할 수 없도록 서비스를 마스킹할 수도 있습니다. 마스킹은 서비스를 다시 마스크 해제할 때까지 영구적으로 사용할 수 없게 하는 서비스를 비활성화하는 방법입니다.

사전 요구 사항

- 루트 액세스

절차

- 부팅 시 시작할 서비스를 비활성화합니다.

```
# systemctl disable <name>.service
```

& lt;name >을 비활성화하려는 서비스 단위의 이름으로 바꿉니다(예: bluetooth).

- 선택 사항: 서비스를 영구적으로 사용할 수 없게 하려면 서비스를 마스크하십시오.

```
# systemctl mask <name>.service
```

이 명령은 /etc/systemd/system/name.service 파일을 /dev/null 에 대한 심볼릭 링크로 대체하여 systemd 에 액세스할 수 없는 실제 장치 파일을 렌더링합니다.

추가 리소스

- systemctl(1) 도움말 페이지
- 시스템 서비스 상태 표시
- 시스템 서비스 중지

12.3. 대상 시스템 상태로 부팅

시스템 관리자는 시스템의 부팅 프로세스를 제어하고 시스템을 부팅할 상태를 정의할 수 있습니다. 이를 systemd 대상이라고 하며 시스템이 특정 수준의 기능에 도달하기 시작하는 systemd 장치 세트입니다. systemd 대상으로 작업하는 동안 기본 대상을 보고, 런타임에 대상을 선택하고, 기본 부팅 대상을 변경하고, 긴급 또는 복구 대상으로 부팅할 수 있습니다.

12.3.1. 대상 단위 파일

systemd 의 대상은 시스템을 시작하는 동안 동기화 지점 역할을 하는 관련 장치의 그룹입니다. .target 파일 확장자로 끝나는 대상 장치 파일은 systemd 대상을 나타냅니다. 대상 장치의 목적은 종속성 체인을 통해 다양한 systemd 장치를 그룹화하는 것입니다.

다음 예제를 고려하십시오.

- *그래픽 세션을 시작하기 위한 **graphical.target** 단위는 **GNOME Display Manager(gdm.service)** 또는 계정 서비스(**accounts-daemon.service**)와 같은 시스템 서비스를 시작하고, **multi-user.target** 단위 도 활성화합니다.*

- *마찬가지로, **multi-user.target** 단위는 **NetworkManager(NetworkManager.service)** 또는 **D-Bus(dbus.service)**와 같은 다른 필수 시스템 서비스를 시작하고 **basic.target** 이라는 다른 대상 장치를 활성화합니다.*

*다음 **systemd** 대상을 기본값 또는 현재 대상으로 설정할 수 있습니다.*

*표 12.3. 일반적인 **systemd** 대상*

rescue	기본 시스템에서 가져오고 복구 쉘을 생성하는 단위 대상
multi-user	다중 사용자 시스템을 설정하기 위한 단위 대상
graphical	그래픽 로그인 화면을 설정하는 단위 대상
emergency	기본 콘솔에서 긴급 쉘을 시작하는 단위 대상

추가 리소스

- ***systemd.special(7)** 도움말 페이지*

- ***systemd.target(5)** 도움말 페이지*

12.3.2. 부팅할 기본 대상 변경

*시스템이 시작되면 **systemd** 는 실제 대상 장치를 가리키는 **default.target** 심볼릭 링크를 활성화합니다. **/etc/systemd/system/default.target** 파일에서 현재 선택한 기본 대상 장치를 찾을 수 있습니다. 각 대상은 특정 수준의 기능을 나타내며 다른 단위를 그룹화하는 데 사용됩니다. 또한 대상 단위는 부팅 중에 동기화 지점 역할을 합니다. 시스템이 부팅되는 기본 대상을 변경할 수 있습니다. 기본 대상 장치를 설정하면 다음에 다시 부팅할 때까지 현재 대상이 변경되지 않은 상태로 유지됩니다.*

사전 요구 사항

- 루트 액세스

절차

1. systemd 가 시스템을 시작하는 데 사용하는 현재 기본 대상 장치를 결정합니다.

```
# systemctl get-default
graphical.target
```

2. 현재 로드된 대상을 나열합니다.

```
# systemctl list-units --type target
```

3. 기본적으로 다른 대상 단위를 사용하도록 시스템을 구성합니다.

```
# systemctl set-default <name>.target
```

& lt;name >을 기본적으로 사용하려는 대상 단위의 이름으로 바꿉니다.

```
Example:
# systemctl set-default multi-user.target
Removed /etc/systemd/system/default.target
Created symlink /etc/systemd/system/default.target -> /usr/lib/systemd/system/multi-user.target
```

4. 기본 대상 단위를 확인합니다.

```
# systemctl get-default
multi-user.target
```

5. 재부팅하여 변경 사항을 적용합니다.

```
# reboot
```

추가 리소스

- systemctl(1) 도움말 페이지

- *systemd.special(7) 도움말 페이지*

- *bootup(7) 도움말 페이지*

12.3.3. 현재 대상 변경

*실행 중인 시스템에서 재부팅하지 않고 현재 부팅의 대상 장치를 변경할 수 있습니다. 다른 대상으로 전환하면 **systemd** 는 이 대상에 필요한 모든 서비스와 해당 종속 항목을 시작하고 새 대상이 활성화하지 않는 모든 서비스를 중지합니다. 다른 대상을 격리하면 현재 부팅에만 영향을 미칩니다.*

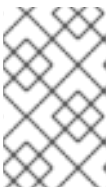
절차

1. *선택 사항: 현재 대상을 확인합니다.*

```
# systemctl get-default
graphical.target
```

2. *선택 사항: 선택할 수 있는 대상 목록을 표시합니다.*

```
# systemctl list-units --type target
```



참고

*단위 파일에 **AllowIsolate=yes** 옵션이 설정된 대상만 격리할 수 있습니다.*

3. *현재 부팅 시 다른 대상 단위로 변경합니다.*

```
# systemctl isolate <name>.target
```

& lt;name >을 현재 부팅 시 사용할 대상 단위의 이름으로 바꿉니다.

```
Example:
# systemctl isolate multi-user.target
```

*이 명령은 **multi-user** 및 모든 종속 단위라는 대상 장치를 시작하고 다른 모든 장치를 즉시 중지합니다.*

추가 리소스

- **systemctl(1)** 도움말 페이지

12.3.4. 복구 모드로 부팅

시스템이 이후 대상에 도달할 수 없는 경우 문제 해결 또는 복구를 위해 단일 사용자 환경을 제공하는 복구 모드로 부팅하고 일반 부팅 프로세스가 실패합니다. 복구 모드에서는 시스템이 모든 로컬 파일 시스템을 마운트하고 특정 중요한 시스템 서비스를 시작하려고 하지만 네트워크 인터페이스를 활성화하지는 않습니다.

사전 요구 사항

- 루트 액세스

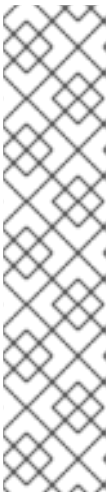
절차

- 복구 모드로 들어가려면 현재 세션의 현재 대상을 변경합니다.

```
# systemctl rescue

Broadcast message from root@localhost on pts/0 (Fri 2023-03-24 18:23:15 CEST):

The system is going down to rescue mode NOW!
```



참고

이 명령은 **systemctl isolate rescue.target** 과 유사하지만 현재 시스템에 로그인한 모든 사용자에게 정보 메시지를 보냅니다.

systemd 가 메시지를 보내지 않도록 하려면 **--no-wall** 명령줄 옵션을 사용하여 다음 명령을 입력합니다.

```
# systemctl --no-wall rescue
```

문제 해결 단계

시스템이 복구 모드로 전환할 수 없는 경우 가능한 최소한의 환경을 제공하는 긴급 모드로 부팅할 수

있습니다. 긴급 모드에서는 읽기용으로만 루트 파일 시스템을 마운트하고 다른 로컬 파일 시스템을 마운트하지 않고 네트워크 인터페이스를 활성화하지 않으며 몇 가지 필수 서비스만 시작합니다.

12.3.5. 부팅 프로세스 문제 해결

시스템 관리자는 부팅 시 기본값이 아닌 대상을 선택하여 부팅 프로세스의 문제를 해결할 수 있습니다. 부팅 시 대상을 변경하면 단일 부팅에만 영향을 미칩니다. 가능한 가장 최소한의 환경을 제공하는 긴급 모드로 부팅할 수 있습니다.

절차

1. 시스템을 재부팅하고 일반 부팅을 시작하는 **Enter** 키를 제외한 임의의 키를 눌러 부트 로더 메뉴 카운트다운을 중단합니다.

2. 시작할 커널 항목으로 커서를 이동합니다.

3. **E** 키를 눌러 현재 항목을 편집합니다.

4. **linux** 로 시작하는 행 끝으로 이동하고 **Ctrl+E**를 눌러 행 끝으로 이동합니다.

```
linux ($root)/vmlinuz-5.14.0-70.22.1.e19_0.x86_64 root=/dev/mapper/rhel-root ro crash\
kernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv/swap rhgb quiet
```

5. 대체 부팅 대상을 선택하려면 **linux** 로 시작하는 행의 끝에 **systemd.unit=** 매개 변수를 추가합니다.

```
linux ($root)/vmlinuz-5.14.0-70.22.1.e19_0.x86_64 root=/dev/mapper/rhel-root ro crash\
kernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv/swap rhgb quiet
systemd.unit=<name>.target
```

& lt;name >을 사용하려는 대상 단위의 이름으로 바꿉니다. 예: **systemd.unit=emergency.target**

6. **Ctrl+X**를 눌러 이러한 설정으로 부팅합니다.

12.4. 시스템 종료, 일시 중지 및 절전 관리

시스템 관리자는 다양한 전원 관리 옵션을 사용하여 전력 소비를 관리하고, 적절한 종료를 수행하여 모든 데이터가 저장되도록 하거나, 시스템을 다시 시작하여 변경 및 업데이트를 적용할 수 있습니다.

12.4.1. 시스템 종료

시스템을 종료하려면 **systemctl** 유틸리티를 직접 사용하거나 **shutdown** 명령을 통해 이 유틸리티를 호출할 수 있습니다.

종료 를 사용하면 다음과 같은 이점이 있습니다.

- **time** 인수를 사용하여 종료를 예약할 수 있습니다. 또한 시스템 종료가 예약되었음을 사용자에게 경고합니다.
- 종료를 취소할 수 있습니다.

추가 리소스

- **systemctl**을 사용한 전원 관리 명령 개요

12.4.2. 시스템 종료 예약

시스템 관리자는 지연된 종료를 예약하여 사용자가 작업을 저장하고 시스템을 로그아웃하는 시간을 제공할 수 있습니다. **shutdown** 명령을 사용하여 다음 작업을 수행합니다.

- 시스템을 종료하고 특정 시간에 시스템 전원을 끕니다.
- 시스템의 전원을 끄지 않고 시스템 종료 및 중지
- 보류 중인 종료 취소

사전 요구 사항

- 루트 액세스

절차

***shutdown** 명령을 사용하여 다음 작업을 수행합니다.*

- *시스템을 종료할 시간을 지정하고 머신의 전원을 끕니다.*

```
# shutdown --poweroff hh:mm
```

*여기서 **hh:mm** 는 **24** 시간 표기법의 시간입니다. 새 로그인을 방지하기 위해 시스템을 종료하기 전에 **/run/nologin** 파일이 **5**분 전에 생성됩니다.*

*시간 인수를 사용하는 경우 선택적 월 메시지 (예: **shutdown --poweroff 13:59 "Attention)** 를 지정하여 계획된 종료의 시스템에 로그인한 사용자에게 알릴 수 있습니다. 시스템이 **13:59**에서 종료됩니다..*

- *머신의 전원을 끄지 않고 지연 후 시스템을 종료하고 중지합니다.*

```
# shutdown --halt +m
```

*여기서 **+m** 은 지연 시간(분)입니다. **now** 키워드를 **+0** 의 별칭으로 사용할 수 있습니다.*

- *보류 중인 종료를 취소합니다.*

```
# shutdown -c
```

추가 리소스

- ***shutdown(8)** 매뉴얼 페이지*
- ***systemctl** 명령을 사용하여 시스템 종료*

12.4.3. systemctl 명령을 사용하여 시스템 종료

*시스템 관리자는 시스템을 종료하고 시스템 전원을 끄거나 **systemctl** 명령을 사용하여 시스템의 전원을 끄지 않고 시스템을 중지할 수 있습니다.*

사전 요구 사항

- *루트 액세스*

절차

***systemctl** 명령을 사용하여 다음 작업을 수행합니다.*

- *시스템을 종료하고 시스템 전원을 끕니다.*

```
# systemctl poweroff
```

- *시스템 전원을 끄지 않고 시스템을 종료하고 중지합니다.*

```
# systemctl halt
```

참고

*기본적으로 이러한 명령 중 하나를 실행하면 **systemd** 에서 현재 시스템에 로그인한 모든 사용자에게 정보 메시지를 보냅니다. **systemd** 가 이 메시지를 보내지 않도록 하려면 **--no-wall** 명령줄 옵션을 사용하여 선택한 명령을 실행합니다.*

12.4.4. 시스템을 다시 시작

*시스템을 다시 시작하면 **systemd** 는 실행 중인 모든 프로그램 및 서비스를 중지하고 시스템이 종료되고 즉시 다시 시작합니다. 시스템을 다시 시작하면 다음과 같은 상황에서 도움이 될 수 있습니다.*

- *새 소프트웨어 또는 업데이트를 설치한 후*
- *시스템 설정을 변경한 후*
- *시스템 문제 해결 시*

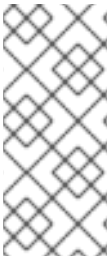
사전 요구 사항

- 루트 액세스

절차

- 시스템을 다시 시작하십시오.

```
# systemctl reboot
```



참고

기본적으로 이 명령을 사용하면 **systemd** 는 현재 시스템에 로그인한 모든 사용자에게 정보 메시지를 보냅니다. **systemd** 가 이 메시지를 보내지 않도록 하려면 **--no-wall** 옵션을 사용하여 이 명령을 실행합니다.

12.4.5. 시스템을 일시 중단하고 완화하여 전력 소비 최적화

시스템 관리자는 전력 소비를 관리하고, 시스템에 전력을 절약하고, 시스템의 현재 상태를 유지할 수 있습니다. 이렇게 하려면 다음 모드 중 하나를 적용합니다.

일시 중단

일시 중지는 **RAM**에 시스템 상태를 저장하고 **RAM** 모듈을 제외하고 시스템의 대부분의 장치의 전원을 끕니다. 시스템을 다시 켜면 시스템을 다시 부팅할 필요 없이 **RAM**에서 해당 상태를 복원합니다. 시스템 상태가 하드 디스크가 아닌 **RAM**에 저장되기 때문에 시스템을 일시 중지 모드로 복원하는 것은 절전 모드보다 훨씬 빠릅니다. 그러나 일시 중지된 시스템 상태도 정전에 취약합니다.

Hibernate

hibernating은 시스템 상태를 하드 디스크 드라이브에 저장하고 시스템의 전원을 컵니다. 시스템을 다시 켜면 시스템을 다시 부팅하지 않고도 저장된 데이터에서 해당 상태를 복원합니다. 시스템 상태가 하드 디스크에 저장되고 **RAM**에 저장되지 않으므로 시스템은 **RAM** 모듈에 전력을 공급할 필요가 없습니다. 그러나 결과적으로 시스템 장애 조치**(hibernation)**를 복원하는 것은 일시 중지 모드에서 복원하는 것보다 훨씬 느려집니다.

하이브리드 절전

이는 하이버네이션 및 일시 중단의 요소를 결합합니다. 시스템은 먼저 하드 디스크 드라이브에 현재 상태를 저장하고 일시 중단과 유사한 저전력 상태를 입력하면 시스템이 더 빨리 다시 시작할 수 있습니다. 하이브리드 절전의 이점은 시스템이 절전 상태 중에 전원을 끊는 경우에도 하이버네이션과 유사하게 하드 디스크에 저장된 이미지에서 이전 상태를 복구할 수 있다는 것입니다.

suspend-ECDHE-hibernate

이 모드는 먼저 시스템을 일시 중단하여 현재 시스템 상태를 **RAM**에 저장하고 시스템을 저전력

*모드로 전환합니다. **HibernateDelaySec** 매개변수에 정의할 수 있는 특정 기간 동안 일시 중지된 경우 시스템 **hibernates**입니다. 하이버네이션은 시스템 상태를 하드 디스크 드라이브에 저장하고 시스템을 완전히 종료합니다. **suspend-hibernate** 모드는 여전히 작업을 재개할 수 있는 동안 배터리 전원을 예약할 수 있는 이점을 제공합니다. 또한 이 모드를 사용하면 정전 시 데이터가 저장됩니다.*

사전 요구 사항

- *루트 액세스*

절차

절전을 위한 적절한 방법을 선택합니다.

- *시스템을 일시 중지합니다.*

```
# systemctl suspend
```

- ***Hibernate** 시스템:*

```
# systemctl hibernate
```

- ***Hibernate** 및 시스템 일시 중단:*

```
# systemctl hybrid-sleep
```

- *시스템을 일시 중지한 다음 **hibernate**를 실행합니다.*

```
# systemctl suspend-then-hibernate
```

12.4.6. systemctl을 사용하여 전원 관리 명령 개요

*다음 **systemctl** 명령 목록을 사용하여 시스템의 전원 관리를 제어할 수 있습니다.*

*표 **12.4. systemctl** 전원 관리 명령 개요*

systemctl 명령	설명
systemctl halt	시스템을 중지합니다.
systemctl poweroff	시스템의 전원을 끕니다.
systemctl reboot	시스템을 다시 시작합니다.
systemctl suspend	시스템을 일시 중지합니다.
systemctlloadbalancing	Hibernates the system을 실행합니다.
systemctl hybrid-sleep	Hibernates를 사용하여 시스템을 일시중지합니다.

12.4.7. 전원 버튼을 누를 때 시스템이 작동하는 방식 변경

컴퓨터의 전원 버튼을 누르면 기본적으로 시스템이 종료됩니다. 기본 설정에 따라 이 동작을 사용자 지정할 수 있습니다.

사전 요구 사항

- *관리 액세스.*

절차

1. ***/etc/systemd/logind.conf** 구성 파일을 엽니다.*
2. ***HandlePowerKey=poweroff** 라는 행을 찾습니다.*
3. *행이 **#** 기호로 시작하는 경우 이를 제거하여 설정을 활성화합니다.*
4. ***poweroff** 를 다음 옵션 중 하나로 교체합니다.*
   - *전원 끄기 는 컴퓨터를 종료합니다.*
   - *시스템을 재부팅 하려면 재부팅합니다.*

- *시스템 중지 를 시작하는 것을 중지합니다.*
- ***kexec** 기반 재부팅의 경우 **kexec**.*
- *시스템을 일시 중단하려면 일시 중지합니다.*
- ***Hibernate**: 시스템 하이버네이션을 시작합니다.*
- *아무것도 하지 않는 것을 무시 하십시오.*

예를 들어 전원 버튼을 누를 때 시스템을 재부팅하려면 다음 설정을 사용합니다.

```
HandlePowerKey=reboot
```

5. *변경 사항을 저장하고 편집기를 종료합니다.*

13장. 시간 동기화 구성

IT 환경에서 정확한 시간 유지가 중요합니다. 모든 네트워크 장치의 일관된 시간은 로그 파일 및 특정 프로토콜의 추적 가능성을 향상시킵니다. 예를 들어 Kerberos는 타임스탬프를 사용하여 재생 공격을 방지합니다.

13.1. CHRONY 모음을 사용하여 NTP 구성

IT의 여러 가지 이유로 정확한 시간 유지는 중요합니다. 예를 들어 네트워킹에서는 패킷 및 로그의 정확한 타임스탬프가 필요합니다. Linux 시스템에서 NTP 프로토콜은 사용자 공간으로 실행되는 데몬에 의해 구현됩니다.

사용자 공간 데몬은 커널에서 실행되는 시스템 클럭을 업데이트합니다. 시스템 클럭은 다양한 클럭 소스를 사용하여 시간을 유지할 수 있습니다. 일반적으로TSC(Time Stamp TSC)가 사용됩니다. TSC는 마지막 재설정 이후 사이클 수를 계산하는 CPU 레지스터입니다. 매우 빠르고 높은 해상도가 있으며 중단이 없습니다.

Red Hat Enterprise Linux 8부터 NTP 프로토콜은 chronyd 데몬에서 구현되며 chrony 패키지의 리포지토리에서 사용할 수 있습니다.

다음 섹션에서는 chrony 제품군을 사용하여 NTP를 설정하는 방법을 설명합니다.

13.1.1. chrony 제품군 소개

Chrony 는 NTP(Network Time Protocol) 구현입니다. chrony 를 사용할 수 있습니다:

- 시스템 클럭을 NTP 서버와 동기화하려면
- 시스템 클럭을 참조 클럭과 동기화하기 위해, 예를 들면 GPS 수신기
- 시스템 클럭을 수동 시간 입력과 동기화하려면
- NTPv4(RFC 5905) 서버 또는 피어로서 네트워크의 다른 컴퓨터에 시간 서비스를 제공합니다.

chrony 는 간헐적인 네트워크 연결, 과도한 정체된 네트워크, 온도 변경(또는 기존 컴퓨터 클럭이 온도에 민감함) 및 가상 머신에서 실행되지 않는 시스템을 포함한 다양한 조건에서 잘 작동합니다.

인터넷을 통해 동기화된 두 시스템 간의 일반적인 정확도는 몇 밀리초 내에 있으며 10초 이내에 LAN 의 경우 입니다. 하드웨어 타임스탬프 또는 하드웨어 참조 클럭은 마이크로초 수준에 동기화된 두 시스템 간의 정확도를 향상시킬 수 있습니다.

chrony 는 사용자 공간으로 실행되는 데몬인 chronyd, chronyc, chronyd 의 성능을 모니터링하고 실행 시 다양한 운영 매개 변수를 변경하는 데 사용할 수 있는 명령행 프로그램으로 구성됩니다.

chrony 데몬인 chronyd 는 명령행 유틸리티 chronyc 에서 모니터링하고 제어할 수 있습니다. 이 유틸리티는 여러 명령을 입력하여 chronyd 의 현재 상태를 쿼리하고 구성을 변경할 수 있는 명령 프롬프트를 제공합니다. 기본적으로 chronyd 는 chronyc 의 로컬 인스턴스의 명령만 허용하지만 원격 호스트에서 모니터링 명령도 허용하도록 구성할 수 있습니다. 원격 액세스는 제한되어야 합니다.

13.1.2. chronyc를 사용하여 chronyd 제어

chronyc 명령줄 유틸리티를 사용하여 chronyd 를 제어할 수 있습니다.

절차

1. 대화형 모드에서 명령행 유틸리티 chronyc 를 사용하여 chronyd 의 로컬 인스턴스를 변경하려면 root 로 다음 명령을 입력합니다.

```
# chronyc
```

restricted 명령 중 일부를 사용할 경우 chronyc 를 root 로 실행해야 합니다.

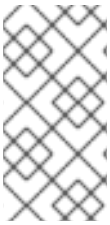
chronyc 명령 프롬프트가 다음과 같이 표시됩니다.

```
chronyc>
```

2. 모든 명령을 나열하려면 help 를 입력합니다.

3. 또는 다음과 같이 명령과 함께 호출되는 경우 비대화형 명령 모드에서 유틸리티를 호출할 수도 있습니다.

```
chronyc command
```



참고

chronyc 를 사용한 변경 사항은 영구적이 아니며 chronyd 를 다시 시작한 후 손실됩니다. 영구 변경 사항은 /etc/chrony.conf 를 수정합니다.

13.2. CHRONY 사용

다음 섹션에서는 chronyd 를 설치, 시작 및 중지하고 chrony 가 동기화되었는지 확인하는 방법에 대해 설명합니다. 섹션은 시스템 Clock을 수동으로 조정하는 방법도 설명합니다.

13.2.1. chrony 관리

다음 절차에서는 chronyd 의 상태를 설치, 시작, 중지 및 확인하는 방법을 설명합니다.

절차

1. chrony 제품군은 기본적으로 Red Hat Enterprise Linux에 설치됩니다. 이 파일을 확인하려면 root 로 다음 명령을 실행하십시오.

```
# dnf install chrony
```

chrony 데몬의 기본 위치는 /usr/sbin/chronyd 입니다. 명령행 유틸리티는 /usr/bin/chronyc 에 설치됩니다.

2. chronyd 의 상태를 확인하려면 다음 명령을 실행합니다.

```
$  systemctl status chronyd
chronyd.service - NTP client/server
  Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled)
  Active: active (running) since Wed 2013-06-12 22:23:16 CEST; 11h ago
```

3. chronyd 를 시작하려면 root 로 다음 명령을 실행합니다.

```
# systemctl start chronyd
```

*시스템을 시작할 때 **chronyd** 가 자동으로 시작되도록 하려면 **root** 로 다음 명령을 실행합니다.*

```
# systemctl enable chronyd
```

4. *chronyd 를 중지하려면 **root** 로 다음 명령을 실행합니다.*

```
# systemctl stop chronyd
```

*시스템 시작 시 **chronyd** 가 자동으로 시작되지 않도록 하려면 **root** 로 다음 명령을 실행합니다.*

```
# systemctl disable chronyd
```

13.2.2. chrony가 동기화되었는지 확인

*다음 절차에서는 **chrony** 가 **tracking,sources** 및 **sourcestats** 명령을 사용하는 것과 동기화되었는지 확인하는 방법을 설명합니다.*

절차

1. *chrony 추적을 확인하려면 다음 명령을 실행합니다.*

```
$  chronyc tracking
Reference ID    : CB00710F (ntp-server.example.net)
Stratum         : 3
Ref time (UTC)  : Fri Jan 27 09:49:17 2017
System time     :  0.000006523 seconds slow of NTP time
Last offset     : -0.000006747 seconds
RMS offset      : 0.000035822 seconds
Frequency       : 3.225 ppm slow
Residual freq   : 0.000 ppm
Skew            : 0.129 ppm
Root delay      : 0.013639022 seconds
Root dispersion : 0.001100737 seconds
Update interval : 64.2 seconds
Leap status     : Normal
```

2. *source 명령은 **chronyd** 가 액세스 중인 현재 시간 소스에 대한 정보를 표시합니다. **chrony** 소스를 확인하려면 다음 명령을 실행합니다.*

```
$ chronyc sources
 210 Number of sources = 3
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
====
#* GPS0                      0   4   377    11   -479ns[ -621ns] /-  134ns
^? a.b.c                     2   6   377    23   -923us[ -924us] +/-   43ms
^ d.e.f                    1   6   377    21  -2629us[-2619us] +/-   86ms
```

선택 사항 **-v** 인수를 지정하여 자세한 정보를 출력할 수 있습니다. 이 경우 추가 주석 줄은 열의 의미를 상기시키는 것으로 표시됩니다.

3.

sourcestats 명령은 **chronyd** 에서 현재 검사 중인 각 소스에 대한 드리프트 비율 및 오프셋 추정 프로세스에 대한 정보를 표시합니다. **chrony** 소스 통계를 확인하려면 다음 명령을 실행합니다.

```
$  chronyc sourcestats
210 Number of sources = 1
Name/IP Address            NP  NR  Span  Frequency  Freq Skew  Offset  Std Dev
===============================================================================
====
abc.def.ghi                11   5   46m     -0.001      0.045     1us    25us
```

선택적 인수 **-v** 를 지정할 수 있습니다. 즉 자세한 내용은 다음과 같습니다. 이 경우 추가 주석 줄은 열의 의미를 상기시키는 것으로 표시됩니다.

추가 리소스

- **chronyc(1)** 도움말 페이지

13.2.3. 시스템 시계 수동 조정

다음 절차에서는 시스템 **Clock**을 수동으로 조정하는 방법을 설명합니다.

절차

1.

시스템 클럭을 즉시 단계적으로 조정하려면 슬래핑하여 진행 중인 조정을 바이패스하고 **root** 로 다음 명령을 실행합니다.

```
# chronyc makestep
```

rtcfile 지시문을 사용하는 경우 실시간 클럭을 수동으로 조정할 수 없습니다. 임의의 조정으로 chrony의 실시간 클럭 드리프트가 발생하는 비율을 측정해야 합니다.

13.2.4. chrony 디스패치 스크립트 비활성화

chrony 디스패치 스크립트는 NTP 서버의 온라인 및 오프라인 상태를 관리합니다. 시스템 관리자는 디스패치 스크립트를 비활성화하여 chronyd 가 서버를 지속적으로 폴링하는 상태를 유지할 수 있습니다.

시스템에서 NetworkManager를 활성화하여 네트워킹 구성을 관리하는 경우 NetworkManager는 인터페이스 재구성, 중지 또는 시작 작업 중에 chrony 디스패처 스크립트를 실행합니다. 그러나 NetworkManager 이외의 특정 인터페이스 또는 경로를 구성하는 경우 다음과 같은 상황이 발생할 수 있습니다.

1. *NTP 서버에 대한 경로가 없는 경우 디스패치 스크립트가 실행되어 NTP 서버가 오프라인 상태로 전환될 수 있습니다.*
2. *나중에 경로를 설정하면 스크립트는 기본적으로 다시 실행되지 않으며 NTP 서버는 오프라인 상태로 유지됩니다.*

chronyd 가 별도의 관리 인터페이스가 있는 NTP 서버와 동기화할 수 있도록 하려면 디스패치 스크립트를 비활성화합니다.

사전 요구 사항

- *시스템에 NetworkManager를 설치하고 활성화합니다.*
- *루트 액세스*

절차

1. *chrony 디스패치 스크립트를 비활성화하려면 /dev/null 에 대한 심볼릭 링크를 만듭니다.*

```
# ln -s /dev/null /etc/NetworkManager/dispatcher.d/20-chrony-onoffline
```

참고

이 변경 후 **NetworkManager**는 디스패치 스크립트를 실행할 수 없으며 **NTP** 서버는 항상 온라인 상태로 유지됩니다.

13.2.5. 격리된 네트워크에서 시스템의 chrony 설정

인터넷에 연결되지 않은 네트워크의 경우 하나의 컴퓨터가 기본 타이머로 선택됩니다. 다른 컴퓨터는 서버 또는 클라이언트의 직접 클라이언트입니다. 서버에서 드리프트 파일을 시스템 클럭의 평균 드리프트 비율로 수동으로 설정해야 합니다. 서버가 재부팅되면 주변 시스템에서 시간을 확보하고 평균을 계산하여 시스템 시계를 설정합니다. 그런 다음 드리프트 파일에 따라 조정을 다시 시작합니다. **settime** 명령을 사용하면 드리프트 파일이 자동으로 업데이트됩니다.

다음 절차에서는 격리된 네트워크에서 시스템의 **chrony** 를 설정하는 방법을 설명합니다.

절차

1. 시스템에서 **root** 로 실행되는 텍스트 편집기를 사용하여 **/etc/chrony.conf** 를 다음과 같이 편집합니다.

```
driftfile /var/lib/chrony/drift
commandkey 1
keyfile /etc/chrony.keys
initstepslew 10 client1 client3 client6
local stratum 8
manual
allow 192.0.2.0
```

여기서 **192.0.2.0** 은 클라이언트가 연결할 수 있는 네트워크 또는 서브넷 주소입니다.

2. **root** 로 실행되는 텍스트 편집기를 사용하여 서버의 직접 클라이언트가 되도록 선택된 시스템에서 다음과 같이 **/etc/chrony.conf** 를 편집합니다.

```
server ntp1.example.net
driftfile /var/lib/chrony/drift
logdir /var/log/chrony
log measurements statistics tracking
keyfile /etc/chrony.keys
commandkey 24
local stratum 10
initstepslew 20 ntp1.example.net
allow 192.0.2.123
```

여기서 192.0.2.123 은 서버의 주소이며 ntp1.example.net 은 서버의 호스트 이름입니다. 이 구성을 사용하는 클라이언트는 재시작 시 서버와 다시 동기화됩니다.

서버의 직접 클라이언트가 아니어야 하는 클라이언트 시스템에서 /etc/chrony.conf 파일은 local 및 allow 지시문을 생략해야 한다는 점을 제외하고 동일해야 합니다.

격리된 네트워크에서 로컬 참조 모드를 활성화하는 local 지시문을 사용하면 로컬 참조 모드를 활성화하여 NTP 서버로 작동하는 chronyd 가 동기화되지 않았거나 시계의 마지막 업데이트가 오래 전에 발생했습니다.

네트워크에 있는 여러 서버가 동일한 로컬 구성을 사용하고 두 서버를 폴링하는 클라이언트의 혼동 없이 서로 동기화되도록 하려면 고립 모드를 활성화하는 로컬 지시문의 분리 옵션을 사용합니다. 다른 모든 서버를 로컬로 폴링하도록 각 서버를 구성해야 합니다. 이렇게 하면 참조 ID가 가장 작은 서버만 로컬 참조가 활성 상태이고 다른 서버가 동기화됩니다. 서버가 실패하면 다른 서버가 대신합니다.

13.2.6. 원격 모니터링 액세스 구성

Chronyc 는 다음 두 가지 방법으로 chronyd 에 액세스할 수 있습니다.

- 인터넷 프로토콜, IPv4 또는 IPv6.
- UNIX 도메인 소켓 - 루트 또는 chrony 사용자가 로컬로 액세스할 수 있습니다.

기본적으로 chronyc 는 Unix 도메인 소켓에 연결됩니다. 기본 경로는 /var/run/chrony/chronyd.sock 입니다. 이 연결에 실패하면 예를 들어 chronyc 가 루트가 아닌 사용자에서 실행되는 경우 chronyc 는 127.0.0.1에 연결을 시도한 다음 ::1을 시도합니다.

chronyd 의 동작에 영향을 미치지 않는 다음 모니터링 명령만 네트워크에서 허용됩니다.

- activity
- 수동 목록

- ***rtcdata***
- ***smoothing***
- *소스*
- ***sourcestats***
- ***tracking***
- ***waitsync***

***chronyd** 가 이러한 명령을 허용하는 호스트 세트는 **chronyd** 의 설정 파일에서 **cmdallow** 지시문 또는 **chronyc** 의 **cmdallow** 명령을 사용하여 구성할 수 있습니다. 기본적으로 이 명령은 **localhost(127.0.0.1** 또는 **::1)**에서만 사용할 수 있습니다.*

*다른 모든 명령은 **Unix** 도메인 소켓을 통해서만 허용됩니다. 네트워크를 통해 전송되면 **chronyd** 는 **localhost**에서 가져온 경우에도 **Not authorized error**로 응답합니다.*

*다음 절차에서는 **chronyc** 를 사용하여 **chronyd**에 원격으로 액세스하는 방법을 설명합니다.*

절차

1. */etc/chrony.conf* 파일에 다음을 추가하여 **IPv4** 및 **IPv6** 주소 모두에서 액세스를 허용합니다.

```
bindcmdaddress 0.0.0.0
```

또는

```
bindcmdaddress ::
```

2. *cmdallow 지시문을 사용하여 원격 IP 주소, 네트워크 또는 서브넷의 명령을 허용합니다.*

   /etc/chrony.conf 파일에 다음 내용을 추가합니다.

   ```
   cmdallow 192.168.1.0/24
   ```

3. *방화벽에서 포트 323을 열어 원격 시스템에서 연결합니다.*

   ```
   # firewall-cmd --zone=public --add-port=323/udp
   ```

   선택적으로 --permanent 옵션을 사용하여 포트 323을 영구적으로 열 수 있습니다.

   ```
   # firewall-cmd --permanent --zone=public --add-port=323/udp
   ```

4. *323 포트를 영구적으로 여는 경우 방화벽 구성을 다시 로드합니다.*

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

추가 리소스

- *chrony.conf(5) 도움말 페이지*

13.2.7. RHEL 시스템 역할을 사용하여 시간 동기화 관리

timesync 역할을 사용하여 여러 대상 시스템에서 시간 동기화를 관리할 수 있습니다. timesync 역할은 시스템 클럭을 동기화하도록 NTP 또는 PTP 클라이언트로 작동하도록 NTP 또는 PTP 구현을 설치하고 구성합니다.

주의

timesync 역할은 관리 호스트에서 지정된 또는 감지된 공급자 서비스의 구성을 대체합니다. 이전 설정은 역할 변수에 지정되지 않은 경우에도 손실됩니다. timesync_ntp_provider 변수가 정의되지 않은 경우 유일한 보존 설정은 공급자를 선택하는 것입니다.

다음 예제에서는 서버 풀 하나만 있는 상황에서 **timesync** 역할을 적용하는 방법을 보여줍니다.

예 13.1. 예제 Playbook은 단일 서버 풀에 timesync 역할을 적용

```
---
- hosts: timesync-test
  vars:
    timesync_ntp_servers:
      - hostname: 2.rhel.pool.ntp.org
        pool: yes
        iburst: yes
  roles:
    - rhel-system-roles.timesync
```

timesync 역할 변수에 대한 자세한 참조를 보려면 **rhel-system-roles** 패키지를 설치하고 **/usr/share/doc/rhel-system-roles/timesync** 디렉터리의 **README.md** 또는 **README.html** 파일을 참조하십시오.

추가 리소스

- **RHEL** 시스템 역할을 사용하도록 제어 노드 및 관리형 노드 준비

13.2.8. 추가 리소스

- **chronyc(1)** 도움말 페이지
- **chronyd(8)** 도움말 페이지
- 자주하는 질문

13.3. HW 타임스탬프링이 있는 CHRONY

하드웨어 타임스탬프는 일부 **NIC(Network Interface Controller)**에서 지원되는 기능으로, 수신 및 발신 패킷의 정확한 타임스탬프를 제공합니다. **NTP** 타임스탬프는 일반적으로 시스템 클럭을 사용하여 커널 및 **chronyd** 에 의해 생성됩니다. 그러나 **HW** 타임스탬프ing이 활성화되면 **NIC**는 자체 클럭을 사용하여 패킷이 링크 계층 또는 물리적 계층을 입력하거나 나가는 경우 타임스탬프를 생성합니다. **NTP** 와 함께 사용하면 하드웨어 타임스탬프가 동기화의 정확도를 크게 향상시킬 수 있습니다. 최상의 정확도를 위해 **NTP** 서버와 **NTP** 클라이언트는 하드웨어 타임스탬프를 사용해야 합니다. 이상적인 조건에서는 마이크로초의 정확도가 가능할 수 있습니다.

*하드웨어 타임스탬프를 사용하는 시간 동기화를 위한 또 다른 프로토콜은 **PTP** 입니다.*

***NTP** 와 달리**PTP** 는 네트워크 스위치 및 라우터의 지원에 의존합니다. 동기화의 최상의 정확도에 도달하려면 **PTP** 지원 기능이 포함된 스위치 및 라우터가 있는 네트워크에서 **PTP** 를 사용하고 이러한 스위치 및 라우터가 없는 네트워크에서 **NTP** 를 선호합니다.*

다음 섹션에서는 다음 방법을 설명합니다.

- *하드웨어 타임 스탬프에 대한 지원 확인*
- *하드웨어 타임스탬프 활성화*
- *클라이언트 폴링 간격 구성*
- ***interleaved** 모드 활성화*
- *다수의 클라이언트에 대해 서버 구성*
- *하드웨어 타임스탬프 확인*
- ***PTP-NTP** 브리지 구성*

13.3.1. 하드웨어 타임스탬프에 대한 지원 확인

***NTP** 를 사용한 하드웨어 타임스탬프를 인터페이스에서 확인하려면 **ethtool -T** 명령을 사용합니다. **ethtool** 이 **SOF_TIMESTAMPING_TX_HARDWARE** 및 **SOF_TIMESTAMPING_TX_SOFTWARE** 기능 및 **HWTSTAMP_FILTER_ALL** 필터 모드를 나열하는 경우 **NTP** 를 사용한 하드웨어 타임 스탬프링에 사용할 수 있습니다.*

> *예 **13.2.** 특정 인터페이스에서 하드웨어 타임스탬프 지원 확인*
>
> ```
> # ethtool -T eth0
> ```

출력:

```
Timestamping parameters for eth0:
Capabilities:
        hardware-transmit     (SOF_TIMESTAMPING_TX_HARDWARE)
        software-transmit     (SOF_TIMESTAMPING_TX_SOFTWARE)
        hardware-receive      (SOF_TIMESTAMPING_RX_HARDWARE)
        software-receive      (SOF_TIMESTAMPING_RX_SOFTWARE)
        software-system-clock (SOF_TIMESTAMPING_SOFTWARE)
        hardware-raw-clock    (SOF_TIMESTAMPING_RAW_HARDWARE)
PTP Hardware Clock: 0
Hardware Transmit Timestamp Modes:
        off                   (HWTSTAMP_TX_OFF)
        on                    (HWTSTAMP_TX_ON)
Hardware Receive Filter Modes:
        none                  (HWTSTAMP_FILTER_NONE)
        all                   (HWTSTAMP_FILTER_ALL)
        ptpv1-l4-sync         (HWTSTAMP_FILTER_PTP_V1_L4_SYNC)
        ptpv1-l4-delay-req    (HWTSTAMP_FILTER_PTP_V1_L4_DELAY_REQ)
        ptpv2-l4-sync         (HWTSTAMP_FILTER_PTP_V2_L4_SYNC)
        ptpv2-l4-delay-req    (HWTSTAMP_FILTER_PTP_V2_L4_DELAY_REQ)
        ptpv2-l2-sync         (HWTSTAMP_FILTER_PTP_V2_L2_SYNC)
        ptpv2-l2-delay-req    (HWTSTAMP_FILTER_PTP_V2_L2_DELAY_REQ)
        ptpv2-event           (HWTSTAMP_FILTER_PTP_V2_EVENT)
        ptpv2-sync            (HWTSTAMP_FILTER_PTP_V2_SYNC)
        ptpv2-delay-req       (HWTSTAMP_FILTER_PTP_V2_DELAY_REQ)
```

13.3.2. 하드웨어 타임스탬프 활성화

하드웨어 타임스탬프를 활성화하려면 **/etc/chrony.conf** 파일에 **hwtimestamp** 지시문을 사용합니다. 지시문은 단일 인터페이스를 지정하거나 와일드카드 문자를 사용하여 이를 지원하는 모든 인터페이스에서 하드웨어 타임스탬프를 활성화할 수 있습니다. **linuxptp** 패키지의 **ptp4l** 과 같은 다른 애플리케이션이 없는 경우 인터페이스에서 하드웨어 타임스탬프를 사용하는 경우 와일드카드 사양을 사용합니다. **chrony** 구성 파일에서 여러 **hwtimestamp** 지시문이 허용됩니다.

예 **13.3. hwtimestamp** 지시문을 사용하여 하드웨어 타임스탬프 활성화

```
hwtimestamp eth0
hwtimestamp eth1
hwtimestamp *
```

13.3.3. 클라이언트 폴링 간격 구성

폴링 간격(**64-1024**초)의 기본 범위는 인터넷 서버에 권장됩니다. 로컬 서버 및 하드웨어 타임스탬프의 경우 시스템 클럭의 오프셋을 최소화하기 위해 더 짧은 폴링 간격을 구성해야 합니다.

/etc/chrony.conf 의 다음 지시문은 1초 폴링 간격을 사용하여 로컬 NTP 서버를 지정합니다.

```
server ntp.local minpoll 0 maxpoll 0
```

13.3.4. interleaved 모드 활성화

하드웨어 NTP 어플라이언스가 아니라 일반적인 용도의 컴퓨터에서 chrony 와 같은 소프트웨어 NTP 구현을 실행하는 NTP 서버는 패킷을 보낸 후에만 하드웨어 전송 타임스탬프를 가져옵니다. 이 동작은 서버가 해당하는 패킷에 타임스탬프를 저장하지 않도록 합니다. 전송 후 생성된 전송 타임스탬프를 수신하는 NTP 클라이언트가 /etc/chrony.conf 의 server 지시문에 xleave 옵션을 추가하여 NTP 인터리빙 모드를 사용하도록 클라이언트를 구성합니다.

```
server ntp.local minpoll 0 maxpoll 0 xleave
```

13.3.5. 다수의 클라이언트에 대한 서버 구성

기본 서버 구성을 사용하면 대부분 수천 개의 클라이언트가 동시에 interleaved 모드를 사용할 수 있습니다. 더 많은 수의 클라이언트에 대해 서버를 구성하려면 /etc/chrony.conf 에서 clientloglimit 지시문을 늘립니다. 이 지시문은 서버에 클라이언트 액세스 로깅을 위해 할당된 최대 메모리 크기를 지정합니다.

```
clientloglimit 100000000
```

13.3.6. 하드웨어 타임스탬프 확인

인터페이스에서 하드웨어 타임스탬프가 성공적으로 활성화되었는지 확인하려면 시스템 로그를 확인합니다. 로그에 성공적으로 활성화된 하드웨어 타임스탬프가 있는 각 인터페이스에 대해 chronyd 의 메시지가 포함되어야 합니다.

예 13.4. 하드웨어 타임스탬프가 활성화된 인터페이스에 대한 로그 메시지

```
chronyd[4081]: Enabled HW timestamping on eth0
chronyd[4081]: Enabled HW timestamping on eth1
```

chronyd 가 NTP 클라이언트 또는 피어로 구성된 경우 chronyc ntpdata 명령을 통해 각 NTP 소스에 대해 보고된 전송 및 타임스탬프 지정 모드와 시간 초과 모드를 사용할 수 있습니다.

예 13.5. 각 NTP 소스에 대한 전송, 타임스탬프 지정 및 인터리빙 모드를 보고합니다.

```
# chronyc ntpdata
```

출력:

```
Remote address  : 203.0.113.15 (CB00710F)
Remote port     : 123
Local address   : 203.0.113.74 (CB00714A)
Leap status     : Normal
Version         : 4
Mode            : Server
Stratum         : 1
Poll interval   : 0 (1 seconds)
Precision       : -24 (0.000000060 seconds)
Root delay      : 0.000015 seconds
Root dispersion : 0.000015 seconds
Reference ID    : 47505300 (GPS)
Reference time  : Wed May 03 13:47:45 2017
Offset          : -0.000000134 seconds
Peer delay      : 0.000005396 seconds
Peer dispersion : 0.000002329 seconds
Response time   : 0.000152073 seconds
Jitter asymmetry: +0.00
NTP tests       : 111 111 1111
Interleaved     : Yes
Authenticated   : No
TX timestamping : Hardware
RX timestamping : Hardware
Total TX        : 27
Total RX        : 27
Total valid RX  : 27
```

예 13.6. NTP 측정의 안정성 보고

```
# chronyc sourcestats
```

하드웨어 타임스탬프를 활성화하면 **NTP** 측정의 안정성이 정상적인 부하에 따라 수십 또는 수백 나노초 내에 있어야 합니다. 이 안정성은 **chronyc sourcestats** 명령 출력의 **Std Dev** 열에 보고됩니다.

출력:

```
210 Number of sources = 1
Name/IP Address            NP  NR  Span  Frequency  Freq Skew  Offset  Std Dev
ntp.local                  12   7    11     +0.000      0.019     +0ns    49ns
```

13.3.7. PTP-NTP 브리지 구성

PTP(Precision Time Protocol) 기본 timeserver를 PTP를 지원하는 스위치 또는 라우터가 없는 네트워크에서 사용할 수 있는 경우 컴퓨터는 PTP 클라이언트 및 stratum-1 NTP 서버로만 작동할 수 있습니다. 이러한 컴퓨터에는 두 개 이상의 네트워크 인터페이스가 있어야 하며 기본 시간 서버에 가 있거나 직접 연결할 수 있어야 합니다. 이렇게 하면 네트워크에서 매우 정확한 동기화가 수행됩니다.

linuxptp 패키지에서 ptp4l 및 phc2sys 프로그램을 구성하여 PTP 를 사용하여 시스템 클럭을 동기화하도록 하나의 인터페이스를 사용합니다.

다른 인터페이스를 사용하여 시스템 시간을 제공하도록 chronyd 를 구성합니다.

> 예 13.7. 다른 인터페이스를 사용하여 시스템 시간을 제공하도록 chronyd 구성
>
> ```
> bindaddress 203.0.113.74
> hwtimestamp eth1
> local stratum 1
> ```

13.4. CHRONY의 NTS(NETWORK TIME SECURITY) 개요

NTS(Network Time Security)는 주요 클라이언트를 확장하도록 설계된 NTP(Network Time Protocol) 인증 메커니즘입니다. 클라이언트 시스템으로 이동하는 동안 서버 시스템에서 수신한 패킷이 변경되지 않은지 확인합니다. NTS(Network Time Security)에는 서버와 클라이언트 간에 사용되는 암호화 키를 자동으로 생성하는 NTS-KE(Key Establishment) 프로토콜이 포함되어 있습니다.

주의

NTS는 FIPS 및 OSPP 프로필과 호환되지 않습니다. FIPS 및 OSPP 프로필을 활성화하면 NTS로 구성된 chronyd 가 치명적 메시지와 함께 중단될 수 있습니다. GNUTLS_FORCE_FIPS_MODE=0 을 /etc/sysconfig/chronyd 파일에 추가하여 chronyd 서비스의 OSPP 프로필 및 FIPS 모드를 비활성화할 수 있습니다.

13.4.1. 클라이언트 구성 파일에서 NTS(Network Time Security) 활성화

기본적으로 NTS(Network Time Security)는 활성화되어 있지 않습니다. /etc/chrony.conf 에서 NTS를 활성화할 수 있습니다. 이를 위해 다음 단계를 수행하십시오.

사전 요구 사항

- ***NTS**를 지원하는 서버*

절차

클라이언트 구성 파일에서 다음을 수행합니다.

1. *권장되는 **iburst** 옵션 외에 **nts** 옵션을 사용하여 서버를 지정합니다.*

```
For example:
server time.example.com iburst nts
server nts.netnod.se iburst nts
server ptbtime1.ptb.de iburst nts
```

2. *시스템 부팅 중에 **NTS-KE(Network Time Security-Key Establishment)** 세션을 반복하지 않으려면 **chrony.conf**에 다음 행을 추가합니다.*

```
ntsdumpdir /var/lib/chrony
```

3. ***DHCP** 에서 제공하는 **NTP(Network Time Protocol)** 서버와의 동기화를 비활성화하려면 **chrony.conf** 에서 다음 행을 주석 처리하거나 제거합니다(있는 경우).*

```
sourcedir /run/chrony-dhcp
```

4. *변경 사항을 저장하십시오.*

5. ***chronyd** 서비스를 다시 시작하십시오.*

```
systemctl restart chronyd
```

검증

- ***NTS** 키가 성공적으로 설정되었는지 확인합니다.*

```
# chronyc -N authdata

Name/IP address  Mode KeyID Type KLen Last Atmp  NAK Cook CLen
==============================================================
```

```
time.example.com  NTS     1   15  256  33m    0    0    8  100
nts.sth1.ntp.se   NTS     1   15  256  33m    0    0    8  100
nts.sth2.ntp.se   NTS     1   15  256  33m    0    0    8  100
```

KeyID,Type 및 *KLen* 의 값은 *0*이 아니어야 합니다. 값이 *0*이면 *chronyd* 에서 시스템 로그에 오류 메시지가 있는지 확인합니다.

- 클라이언트가 *NTP*를 측정하고 있는지 확인합니다.

```
# chronyc -N sources

MS Name/IP address Stratum Poll Reach LastRx Last sample
===============================================================
time.example.com   3       6   377    45   +355us[ +375us] +/-   11ms
nts.sth1.ntp.se    1       6   377    44   +237us[ +237us] +/-   23ms
nts.sth2.ntp.se    1       6   377    44   -170us[ -170us] +/-   22ms
```

Reach 열에는 *0*이 아닌 값이 있어야 합니다. 이상적인 *377*. 값이 *377*를 거의 얻지 못하거나 *377*를 얻지 못하면 네트워크에서 *NTP* 요청 또는 응답이 손실되고 있음을 나타냅니다.

추가 리소스

- *chrony.conf(5)* 도움말 페이지

13.4.2. 서버에서 NTS(Network Time Security) 활성화

자체 *NTP(Network Time Protocol)* 서버를 실행하는 경우 서버 *NTS(Network Time Security)* 지원을 활성화하여 클라이언트가 안전하게 동기화하도록 할 수 있습니다.

NTP 서버가 다른 서버의 클라이언트인 경우 동기화에 대해 *NTS* 또는 대칭 키를 사용해야 합니다.

사전 요구 사항

- *PEM* 형식의 서버 개인 키
- 필요한 중간 인증서가 있는 서버 인증서 *PEM* 형식

절차

1. *chrony.conf 에서 개인 키와 인증서 파일을 지정합니다. 예를 들면 다음과 같습니다.*

```
ntsserverkey /etc/pki/tls/private/<ntp-server.example.net>.key
ntsservercert /etc/pki/tls/certs/<ntp-server.example.net>.crt
```

2. *그룹 소유권을 설정하여 chrony 시스템 사용자가 키와 인증서 파일을 모두 읽을 수 있는지 확인합니다. 예를 들면 다음과 같습니다.*

```
# chown :chrony /etc/pki/tls//<ntp-server.example.net>.
```

3. *the ntsdumpdir /var/lib/chrony 지시문이 chrony.conf 에 있는지 확인합니다.*

4. *chronyd 서비스를 다시 시작하십시오.*

```
# systemctl restart chronyd
```

중요

서버에 방화벽이 있는 경우 NTP 및 NTS-KE(Network Time Security-Key Establishment)에 대한 UDP 123 및 TCP 4460 포트를 모두 허용해야 합니다.

검증

- *다음 명령을 사용하여 클라이언트 시스템에서 빠른 테스트를 수행합니다.*

```
$ chronyd -Q -t 3 'server

ntp-server.example.net iburst nts maxsamples 1'
2021-09-15T13:45:26Z chronyd version 4.1 starting (+CMDMON +NTP +REFCLOCK
+RTC +PRIVDROP +SCFILTER +SIGND +ASYNCDNS +NTS +SECHASH +IPV6
+DEBUG)
2021-09-15T13:45:26Z Disabled control of system clock
2021-09-15T13:45:28Z System clock wrong by 0.002205 seconds (ignored)
2021-09-15T13:45:28Z chronyd exiting
```

시스템 클록 잘못된 메시지는 NTP 서버가 NTS-KE 연결을 수락하고 NTS 보호 NTP 메시지로 응답하고 있음을 나타냅니다.

- *서버에서 관찰된 **NTS-KE** 연결 및 인증된 **NTP** 패킷을 확인합니다.*

```
# chronyc serverstats

NTP packets received       : 7
NTP packets dropped        : 0
Command packets received   : 22
Command packets dropped    : 0
Client log records dropped : 0
NTS-KE connections accepted: 1
NTS-KE connections dropped : 0
Authenticated NTP packets: 7
```

__NTS-KE__ 연결 값이 __0__이 아닌 __NTP__ 패킷 필드인 경우 클라이언트가 __NTS-KE__ 포트에 연결하고 인증된 __NTP__ 요청을 보낼 수 있었습니다.

14장. 시스템 복구 및 복원

기존 백업을 사용하여 시스템을 복구 및 복구하기 위해 **Red Hat Enterprise Linux**는 **Relax-and-Recover(ReaR)** 유틸리티를 제공합니다.

유틸리티를 재해 복구 솔루션으로 사용하고 시스템 마이그레이션에도 사용할 수 있습니다.

유틸리티를 사용하면 다음 작업을 수행할 수 있습니다.

- 이미지를 사용하여 부팅 가능한 이미지를 생성하고 기존 백업에서 시스템을 복원합니다.
- 원래 스토리지 레이아웃을 복제합니다.
- 사용자 및 시스템 파일을 복원합니다.
- 시스템을 다른 하드웨어로 복원합니다.

또한 재해 복구를 위해 특정 백업 소프트웨어를 **ReaR**과 통합할 수도 있습니다.

ReaR 설정에는 다음과 같은 상위 수준 단계가 포함됩니다.

1. **ReaR**을 설치합니다.
2. **ReaR** 구성 파일을 수정하여 백업 방법 세부 정보를 추가합니다.
3. 복구 시스템을 생성합니다.
4. 백업 파일을 생성합니다.

14.1. REAR 설정

*다음 단계에 따라 **Relax-and-Recover(ReaR)** 유틸리티를 사용하는 패키지를 설치하고 복구 시스템을 생성하고 백업을 구성 및 생성합니다.*

사전 요구 사항

- *백업 복원 계획에 따라 필요한 구성이 준비되었습니다.*

  ***ReaR**과 함께 완전 통합 및 기본 제공 방법인 **NETFS** 백업 메서드를 사용할 수 있습니다.*

절차

1. *다음 명령을 실행하여 **ReaR** 유틸리티를 설치합니다.*

```
# dnf install rear
```

2. *선택한 편집기에서 **ReaR** 구성 파일을 수정합니다. 예를 들면 다음과 같습니다.*

```
# vi /etc/rear/local.conf
```

3. */etc/rear/local.conf* *에 백업 설정 세부 정보를 추가합니다. 예를 들어 **NETFS** 백업 방법의 경우 다음 행을 추가합니다.*

```
BACKUP=NETFS
BACKUP_URL=backup.location
```

   ***backup.location** 을 백업 위치의 **URL**로 바꿉니다.*

4. *새 파일이 생성될 때 이전 백업 아카이브를 유지하도록 **ReaR**을 구성하려면 구성 파일에 다음 행도 추가합니다.*

```
NETFS_KEEP_OLD_BACKUP_COPY=y
```

5. *백업을 증분 방식으로 만들려면 변경된 파일만 각 실행 시 백업됩니다.*

```
BACKUP_TYPE=incremental
```

6. 복구 시스템을 생성합니다.

```
# rear mkrescue
```

7. 복원 계획에 따라 백업을 수행합니다. 예를 들어 NETFS 백업 방법의 경우 다음 명령을 실행합니다.

```
# rear mkbackuponly
```

또는 다음 명령을 실행하여 단일 단계에서 복구 시스템 및 백업을 생성할 수 있습니다.

```
# rear mkbackup
```

이 명령은 rear mkrescue 및 rear mk backuponly 명령의 기능을 결합합니다.

14.2. 64비트 IBM Z 아키텍처에서 REAR RESCUE 이미지 사용

Basic Relax and Recover (ReaR) 기능은 이제 64비트 IBM Z 아키텍처에서 기술 프리뷰로 제공됩니다. IBM Z에서 z/VM 환경에서만 복구 이미지를 생성할 수 있습니다. LPAR(Logical partitions) 백업 및 복구는 테스트되지 않았습니다.

중요

64비트 IBM Z 아키텍처에서는 리어 패키지 버전 2.6-17.el9 이상에서만 지원됩니다. 이전 버전은 기술 프리뷰 기능으로만 사용할 수 있습니다. Red Hat 기술 프리뷰 기능의 지원 범위에 대한 자세한 내용은 https://access.redhat.com/support/offerings/techpreview 을 참조하십시오.

현재 사용 가능한 출력 방법은 초기 프로그램 로드(IPL)입니다. IPL은 zIPL 부트 로더와 함께 사용할 수 있는 커널 및 초기 RAM 디스크(initrd)를 생성합니다.

사전 요구 사항

- Rear가 설치되어 있습니다.
  - ReaR을 설치하려면 dnf install rear 명령을 실행합니다.

절차

다음 변수를 /etc/rear/local.conf 에 추가하여 64비트 IBM Z 아키텍처에서 복구 이미지를 생성하도록 ReaR을 구성합니다.

1. *IPL 출력 방법을 구성하려면 OUTPUT=IPL 을 추가합니다.*

2. *백업 방법 및 대상을 구성하려면 RuntimeClass 및 octets _URL 변수를 추가합니다. 예를 들면 다음과 같습니다.*

```
BACKUP=NETFS

BACKUP_URL=nfs://<nfsserver name>/<share path>
```

중요

로컬 백업 스토리지는 현재 64비트 IBM Z 아키텍처에서 지원되지 않습니다.

3. *선택적으로 커널 및 initrd 파일을 저장할 수 있도록 OUTPUT_URL 변수를 구성할 수도 있습니다. 기본적으로 OUTPUT_URL 은 etcdctl_URL에 맞게 조정됩니다.*

4. *백업 및 복구 이미지 생성을 수행하려면 다음을 수행합니다.*

```
# rear mkbackup
```

5. *이렇게 하면 EgressIP _URL 또는 OUTPUT_URL (설정된 경우) 변수에서 지정한 위치에 커널 및 initrd 파일이 생성되고 지정된 백업 방법을 사용하여 백업을 생성합니다.*

6. *시스템을 복구하려면 3단계에서 만든 ReaR 커널 및 initrd 파일을 사용하고, zipl 부트 로더, 커널, initrd를 사용하여 준비한 zipl boot loader, kernel 및 initrd를 사용하여 준비한 FCP(Direct Attached Storage Device) 또는 FCP(Fibre Channel Protocol)에서 부팅하십시오. 자세한 내용은 Prepared DASD 사용을 참조하십시오.*

7. *rescue 커널 및 initrd 가 부팅되면 ReaR rescue 환경을 시작합니다. 시스템 복구를 진행합니다.*

주의

*현재 복구 프로세스는 시스템에 연결된 모든 **DASD(Direct Attached Storage 장치)**를 다시 포맷합니다. 시스템 스토리지 장치에 중요한 데이터가 있는 경우 시스템 복구를 시도하지 마십시오. 여기에는 복구 환경에서 부팅하는 데 사용된 **zipl** 부트 로더, **ReaR** 커널 및 **initrd**로 준비된 장치도 포함됩니다. 복사본을 유지해야 합니다.*

추가 리소스

- *z/VM에 설치*
- *준비 DASD 사용*