Red Hat Enterprise Linux 9

기본 시스템 설정 구성

시스템의 필수 기능을 설정하고 시스템 환경을 사용자 정의

Last Updated: 2025-08-04

Red Hat Enterprise Linux 9 기본 시스템 설정 구성

시스템의 필수 기능을 설정하고 시스템 환경을 사용자 정의

Legal Notice

Abstract

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.

Table of Contents

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

NetworkManager는 호스트에 설치된 각 이더넷 어댑터에 대한 연결 프로필을 생성합니다. 기본적으로 이 프로필은 IPv4 및 IPv6 연결에 DHCP를 사용합니다. 자동으로 생성된 이 프로필을 수정하거나 다음 경우 새 프로필을 추가합니다.

- 네트워크에는 고정 IP 주소 구성과 같은 사용자 지정 설정이 필요합니다.
- 서로 다른 네트워크 간에 호스트가 순환되므로 여러 프로필이 필요합니다.

Red Hat Enterprise Linux는 관리자에게 이더넷 연결을 구성하는 다양한 옵션을 제공합니다. 예를 들면 다음과 같습니다.

- nmcli를 사용하여 명령줄에서 연결을 구성합니다.
- nmtui를 사용하여 텍스트 기반 사용자 인터페이스에서 연결을 구성합니다.
- GNOME 설정 메뉴 또는 nm-connection-editor 애플리케이션을 사용하여 그래픽 인터페이스에서 연결을 구성합니다.
- nmstatectl을 사용하여 Nmstate API를 통해 연결을 구성합니다.
- RHEL 시스템 역할을 사용하여 하나 이상의 호스트에서 연결 구성을 자동화합니다.

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

   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.

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.

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.

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

- RHEL 자동 설치
- 네트워크 장치 이름 지정 표준에 대한 자세한 내용은 *네트워킹 구성 및 관리를* 참조하십시오.

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::fffe 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 결함이 있는 케이블 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 서비스를 다시 시작한 후 Red Hat Knowledgebase 솔루션 NetworkManager가 연결 중복을 참조하십시오.

추가 리소스

- 시스템의 **nm-settings(5)** 도움말 페이지

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.
- **Space** 를 사용하여 확인란을 선택하고 지웁니다.
- 이전 화면으로 돌아가려면 **ESC** 를 사용합니다.

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. **추가를** 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 **Show** 를 누릅니다.

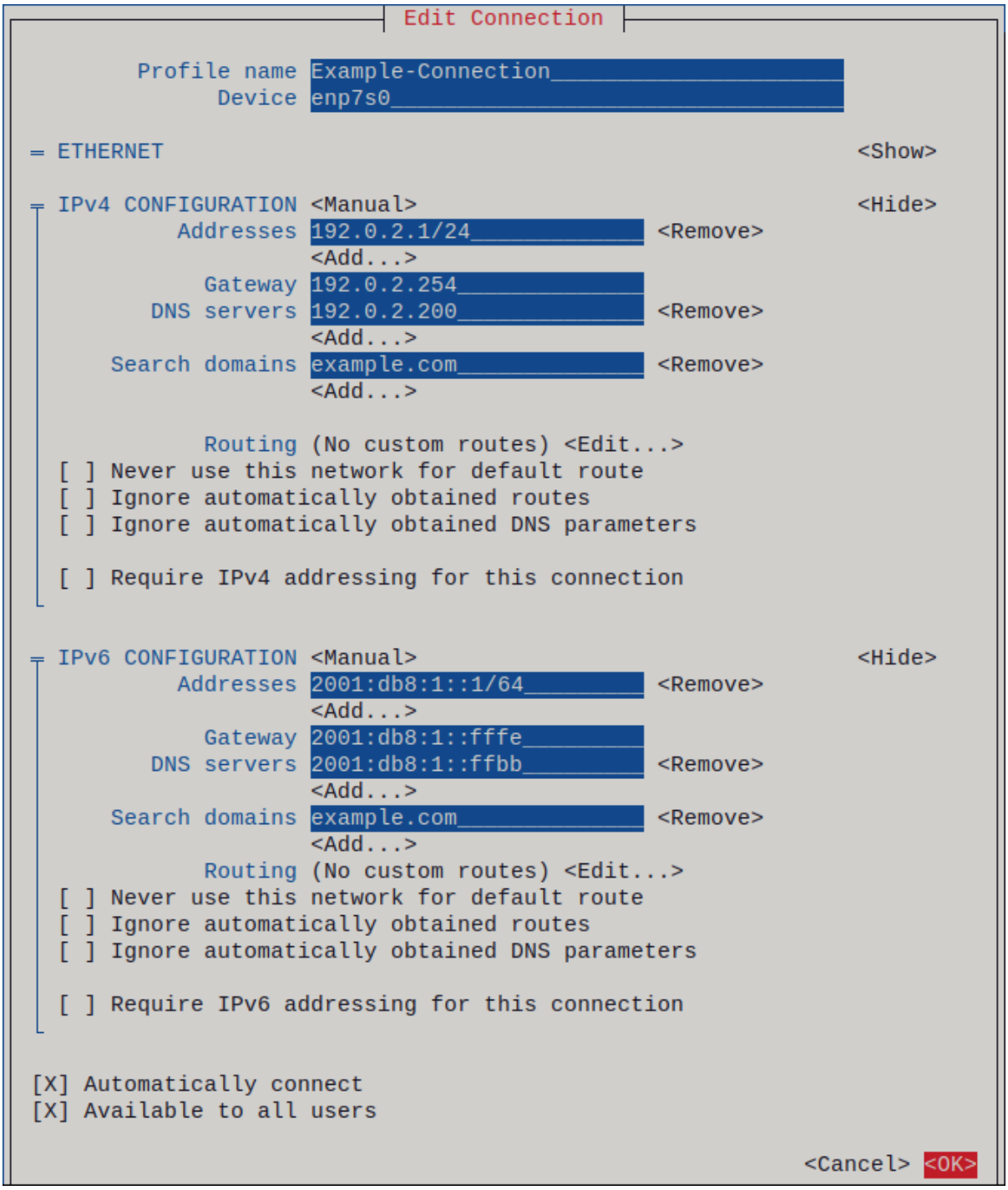
     ii. **주소** 옆에 있는 **추가** 를 클릭하고 CIDR(Classless Inter-Domain Routing) 형식으로 IP 주소와 서브넷 마스크를 입력합니다.
     서브넷 마스크를 지정하지 않으면 NetworkManager는 IPv4 주소에 대해 **/32** 서브넷 마스크를 설정하고 IPv6 주소에 대해 **/64** 를 설정합니다.

     iii. 기본 게이트웨이의 주소를 입력합니다.

     iv. **DNS 서버** 옆에 있는 **추가** 를 클릭하고 DNS 서버 주소를 입력합니다.

     v. **검색 도메인** 옆에 있는 **추가** 를 클릭하고 DNS 검색 도메인을 입력합니다.

그림 1.1. 고정 IP 주소 설정이 포함된 이더넷 연결 예



8. **OK** 를 눌러 새 연결을 만들고 자동으로 활성화합니다.
9. **다시** 키를 눌러 기본 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::fffe 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 결함이 있는 케이블 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 서비스를 다시 시작한 후 Red Hat Knowledgebase 솔루션 NetworkManager가 연결 중복을 참조하십시오.

추가 리소스

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

1.4. 인터페이스 이름으로 네트워크 RHEL 시스템 역할을 사용하여 동적 IP 주소로 이더넷 연결 구성

Red Hat Enterprise Linux 호스트를 이더넷 네트워크에 연결하려면 네트워크 장치의 NetworkManager 연결 프로필을 만듭니다. Ansible 및 **네트워크** RHEL 시스템 역할을 사용하면 이 프로세스를 자동화하고 플레이북에 정의된 호스트에서 연결 프로필을 원격으로 구성할 수 있습니다.

네트워크 RHEL 시스템 역할을 사용하여 DHCP 서버 및 IPv6 SLAAC(stateless address autoconfiguration)에서 IP 주소, 게이트웨이 및 DNS 설정을 검색하는 이더넷 연결을 구성할 수 있습니다. 이 역할을 사용하면 지정된 인터페이스 이름에 연결 프로필을 할당할 수 있습니다.

사전 요구 사항

- 컨트롤 노드 및 관리형 노드를 준비했습니다.
- 관리 노드에서 플레이북을 실행할 수 있는 사용자로 제어 노드에 로그인되어 있습니다.
- 관리 노드에 연결하는 데 사용하는 계정에는 **sudo** 권한이 있습니다.
- 서버의 구성에 물리적 또는 가상 이더넷 장치가 있습니다.
- DHCP 서버 및 SLAAC는 네트워크에서 사용할 수 있습니다.
- 관리형 노드는 NetworkManager 서비스를 사용하여 네트워크를 구성합니다.

절차

1. 다음 콘텐츠를 사용하여 플레이북 파일(예: **~/playbook.yml**)을 생성합니다.

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
    - name: Ethernet connection profile with dynamic IP address settings
      ansible.builtin.include_role:
        name: redhat.rhel_system_roles.network
      vars:
        network_connections:
          - name: enp1s0
            interface_name: enp1s0
            type: ethernet
            autoconnect: yes
            ip:
              dhcp4: yes
              auto6: yes
            state: up
```

예제 플레이북에 지정된 설정은 다음과 같습니다.

dhcp4: yes

DHCP, PPP 또는 유사한 서비스에서 자동 IPv4 주소 할당을 활성화합니다.

auto6: yes

IPv6 자동 구성을 활성화합니다. 기본적으로 NetworkManager는 라우터 알림을 사용합니다. 라우터에서 **관리** 플래그를 알릴 경우 NetworkManager는 DHCPv6 서버에서 IPv6 주소 및 접두사를 요청합니다.

플레이북에 사용되는 모든 변수에 대한 자세한 내용은 제어 노드의 **/usr/share/ansible/roles/rhel-system-roles.network/README.md** 파일을 참조하십시오.

2. 플레이북 구문을 확인합니다.

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

이 명령은 구문만 검증하고 잘못되었지만 유효한 구성으로부터 보호하지 않습니다.

3. Playbook을 실행합니다.

```
$ ansible-playbook ~/playbook.yml
```

검증

- 관리 노드의 Ansible 사실을 쿼리하고 인터페이스에서 IP 주소 및 DNS 설정을 수신했는지 확인합니다.

```
# ansible managed-node-01.example.com -m ansible.builtin.setup
...
        "ansible_default_ipv4": {
            "address": "192.0.2.1",
            "alias": "enp1s0",
            "broadcast": "192.0.2.255",
            "gateway": "192.0.2.254",
            "interface": "enp1s0",
            "macaddress": "52:54:00:17:b8:b6",
            "mtu": 1500,
            "netmask": "255.255.255.0",
            "network": "192.0.2.0",
            "prefix": "24",
            "type": "ether"
        },
        "ansible_default_ipv6": {
            "address": "2001:db8:1::1",
            "gateway": "2001:db8:1::fffe",
            "interface": "enp1s0",
            "macaddress": "52:54:00:17:b8:b6",
            "mtu": 1500,
            "prefix": "64",
            "scope": "global",
            "type": "ether"
        },
        ...
        "ansible_dns": {
            "nameservers": [
                "192.0.2.1",
                "2001:db8:1::ffbb"
            ],
            "search": [
                "example.com"
            ]
        },
...
```

추가 리소스

- **/usr/share/ansible/roles/rhel-system-roles.network/README.md** file
- **/usr/share/doc/rhel-system-roles/network/** 디렉터리

1.5. 추가 리소스

- 네트워킹 구성 및 관리

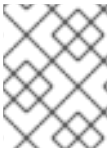
2장. 시스템 등록 및 서브스크립션 관리

서브스크립션은 운영 체제 자체를 포함하여 Red Hat Enterprise Linux에 설치된 제품에 적용됩니다. 시스템을 등록하지 않은 경우 RHEL 리포지토리에 액세스할 수 없습니다. 보안, 버그 수정과 같은 소프트웨어 업데이트는 설치할 수 없습니다. 자체 지원 서브스크립션이 있는 경우에도 서브스크립션 부족에서 더 많은 리소스를 사용할 수 없는 상태로 기술 자료에 대한 액세스 권한을 부여합니다. 서브스크립션을 구매하고 Red Hat CDN(Content Delivery Network)을 사용하면 다음을 추적할 수 있습니다.

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

2.1. 명령줄을 사용하여 시스템 등록

서브스크립션은 운영 체제 자체를 포함하여 Red Hat Enterprise Linux에 설치된 제품에 적용됩니다. 시스템을 등록하지 않은 경우 RHEL 리포지토리에 액세스할 수 없습니다. 보안, 버그 수정과 같은 소프트웨어 업데이트는 설치할 수 없습니다. 자체 지원 서브스크립션이 있는 경우에도 서브스크립션 부족에서 더 많은 리소스를 사용할 수 없는 상태로 기술 자료에 대한 액세스 권한을 부여합니다. Red Hat 계정에 대한 Red Hat Enterprise Linux 서브스크립션을 활성화하고 관리하려면 시스템을 등록해야 합니다.



참고

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

사전 요구 사항

- Red Hat Enterprise Linux 시스템에 대한 서브스크립션이 있습니다.

절차

- 시스템을 등록하고 구독하십시오.

```
# subscription-manager register
Registering to:                 subscription.rhsm.redhat.com:443/subscription
Username: <example_username>
Password: <example_password>
The system has been registered with ID: 37to907c-ece6-49ea-9174-20b87ajk9ee7
The registered system name is:          client1.example.com
```

이 명령을 실행하면 Red Hat Customer Portal 계정의 사용자 이름과 암호를 입력하라는 메시지가 표시됩니다.

등록 프로세스가 실패하면 시스템을 특정 풀에 등록할 수 있습니다. 자세한 내용은 다음 단계를 수행합니다.

  - 서브스크립션의 풀 ID를 확인합니다.

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

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

- 이전 단계에서 확인한 풀 ID로 교체하여 *<example_pool_id>* 시스템에 적절한 서브스크립션을 연결합니다.

```
# subscription-manager attach --pool=<example_pool_id>
```

검증

- 하이브리드 클라우드 콘솔의 **Inventory → Systems** 에서 시스템을 확인합니다.

추가 리소스

- Red Hat Subscription Management 이해
- Red Hat 제품 가입을 위한 워크플로우 이해
- 하이브리드 클라우드 콘솔에서 서브스크립션 인벤토리 보기

2.2. 웹 콘솔을 사용하여 시스템 등록

서브스크립션은 운영 체제 자체를 포함하여 Red Hat Enterprise Linux에 설치된 제품에 적용됩니다. 시스템을 등록하지 않은 경우 RHEL 리포지토리에 액세스할 수 없습니다. 보안, 버그 수정과 같은 소프트웨어 업데이트는 설치할 수 없습니다. 자체 지원 서브스크립션이 있는 경우에도 서브스크립션 부족에서 더 많은 리소스를 사용할 수 없는 상태로 기술 자료에 대한 액세스 권한을 부여합니다. Red Hat 웹 콘솔에서 계정 자격 증명으로 새로 설치된 Red Hat Enterprise Linux를 등록할 수 있습니다.

사전 요구 사항

- RHEL 시스템의 활성 서브스크립션이 있습니다.
- RHEL 9 웹 콘솔을 설치했습니다.
- cockpit 서비스를 활성화했습니다.
- 사용자 계정이 웹 콘솔에 로그인할 수 있습니다.
  자세한 내용은 웹 콘솔 설치 및 활성화를 참조하십시오.

절차

1. 브라우저에서 *https://<ip_address_or_hostname>:9090* 을 열고 웹 콘솔에 로그인합니다.
2. **개요** 페이지의 **상태** 필드에서 **등록된** 경고를 클릭하거나 기본 메뉴에서 **서브스크립션** 을 클릭하여 서브스크립션 정보가 포함된 페이지로 이동합니다.
3. **개요** 필드에서 **등록을** 클릭합니다.
4. **등록 시스템** 대화 상자에서 등록 방법을 선택합니다.
   선택 사항: 조직의 이름 또는 ID를 입력합니다. 계정이 Red Hat 고객 포털에서 두 개 이상의 조직에 속하는 경우 조직 이름 또는 ID를 추가해야 합니다. 조직 ID를 얻으려면 Red Hat에서 기술 계정 관리자에게 확인하십시오.
5. 시스템을 Red Hat Insights에 연결하지 않으려면 **Insights** 확인란을 지웁니다.
6. **등록을** 클릭합니다.

검증

- 하이브리드 클라우드 콘솔에서 서브스크립션 세부 정보를 확인하십시오.

2.3. GNOME 데스크탑 환경에 시스템 등록

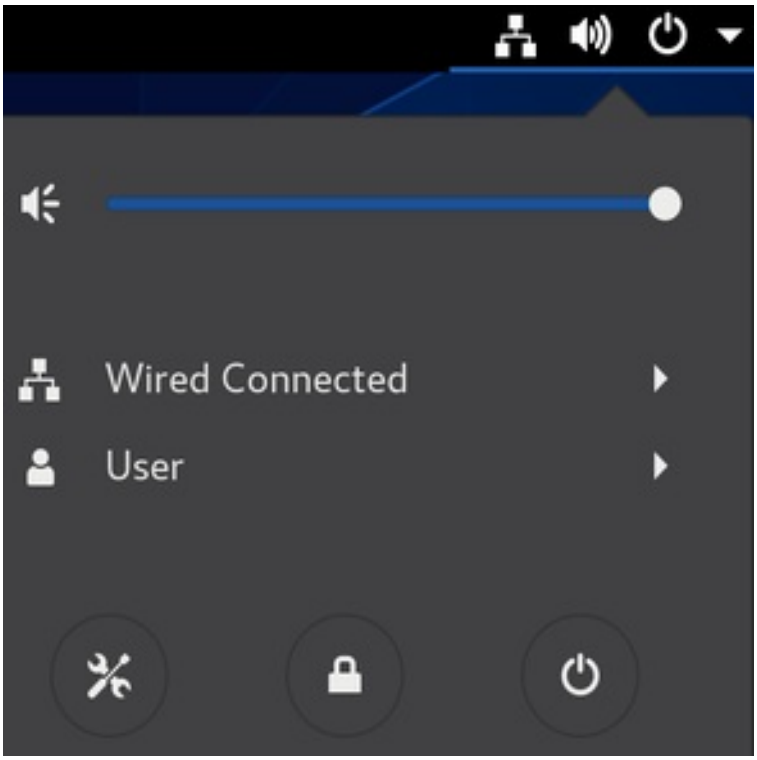
서브스크립션은 운영 체제 자체를 포함하여 Red Hat Enterprise Linux에 설치된 제품에 적용됩니다. 시스템을 등록하지 않은 경우 RHEL 리포지토리에 액세스할 수 없습니다. 보안, 버그 수정과 같은 소프트웨어 업데이트를 설치할 수 없습니다. 자체 지원 서브스크립션이 있는 경우에도 서브스크립션 부족에서 더 많은 리소스를 사용할 수 없는 상태로 기술 자료에 대한 액세스 권한을 부여합니다. 이 절차의 단계에 따라 시스템을 Red Hat 계정에 등록합니다.

사전 요구 사항

- Red Hat 계정을 생성했습니다.
- root 사용자이며 GNOME 데스크탑 환경에 로그인했습니다. 자세한 내용은 RHEL 시스템 등록 및 Red Hat Subscription Manager를 참조하십시오.

절차

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



2. **서브스크립션** 정보로 이동합니다.
3. Red Hat Satellite를 통해 시스템을 등록하려는 경우:
   a. **등록 서버** 섹션에서 **사용자 지정 주소를 선택합니다**.
   b. **URL** 필드에 서버 주소를 입력합니다.
4. **등록 유형** 섹션에서 원하는 등록 방법을 선택합니다.
5. **등록 세부 정보** 섹션을 작성합니다.

6. **등록을** 클릭합니다.

3장. RED HAT 지원에 액세스

문제 해결에 도움이 필요한 경우 Red Hat 지원팀에 문의하십시오.

절차

- Red Hat 지원 웹 사이트에 로그인하고 다음 옵션 중 하나를 선택합니다.
  - 새 지원 케이스를 엽니다.
  - Red Hat 전문가와의 실시간 채팅 시작.
  - 이메일을 보내거나 전화하여 Red Hat 전문가에게 문의하십시오.

3.1. SOSREPORT 유틸리티를 사용하여 시스템에 대한 DAIGNOSTIC 정보를 수집하여 지원 티켓에 연결

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
```

2. 보고서를 생성합니다.

```
# sosreport
```

   선택적으로 명령에 **--upload** 옵션을 전달하여 자동으로 보고서를 업로드하고 지원 케이스에 연결합니다. 이를 위해서는 인터넷 액세스 및 고객 포털 인증 정보가 필요합니다.

3. 선택 사항: 지원 케이스에 보고서를 수동으로 첨부합니다.
   자세한 내용은 Red Hat Knowledgebase 솔루션에서 Red Hat 지원 케이스에 파일을 첨부하는 방법 How can I attach a file to a Red Hat support case? 에서 참조하십시오.

추가 리소스

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

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. 날짜, 시간, 시간대 설정 수동 구성

현재 날짜 및 시간을 표시하려면 다음 단계 중 하나를 사용합니다.

절차

1. 선택 사항: 시간대를 나열합니다.

```
# timedatectl list-timezones

 Europe/Berlin
```

2. 시간대를 설정합니다.

```
# timedatectl set-timezone <time_zone>
```

3. 날짜 및 시간을 설정합니다.

```
# timedatectl set-time <YYYY-mm-dd HH:MM:SS>
```

검증

1. 날짜, 시간, 시간대를 표시합니다.

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

2. 자세한 내용은 timedatectl 명령을 사용합니다.

```
# 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

- **date(1)** 및 **timedatectl(1)** 도움말 페이지

4.2. 웹 콘솔을 사용하여 시간 설정 구성

시간대를 설정하고 RHEL 웹 콘솔에서 NTP(Network Time Protocol) 서버와 시스템 시간을 동기화할 수 있습니다.

사전 요구 사항

- RHEL 9 웹 콘솔을 설치했습니다.
- cockpit 서비스를 활성화했습니다.
- 사용자 계정이 웹 콘솔에 로그인할 수 있습니다.
  자세한 내용은 웹 콘솔 설치 및 활성화를 참조하십시오.

절차

1. RHEL 9 웹 콘솔에 로그인합니다.
   자세한 내용은 웹 콘솔에 로그인 을 참조하십시오.
2. **개요** 에서 현재 시스템 시간을 클릭합니다.
3. **시스템 시간을** 클릭합니다.
4. **시스템 시간 변경** 대화 상자에서 필요한 경우 시간대를 변경합니다.
5. **설정 시간** 드롭다운 메뉴에서 다음 중 하나를 선택합니다.

   수동

   NTP 서버 없이 수동으로 시간을 설정해야 하는 경우 이 옵션을 사용합니다.

   NTP 서버 자동 사용

   이 옵션은 시간을 사전 설정된 NTP 서버와 자동으로 동기화하는 기본 옵션입니다.

   특정 NTP 서버 자동 사용

   시스템을 특정 NTP 서버와 동기화해야 하는 경우에만 이 옵션을 사용합니다. DNS 이름 또는 서버의 IP 주소를 지정합니다.
6. **변경을** 클릭합니다.

검증

- 시스템 탭에 표시된 **시스템** 시간을 확인합니다.

추가 리소스

- Chrony 모음을 사용하여 NTP 구성

4.3. 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

1. 선택 사항: 현재 시스템 로케일 설정을 표시합니다.

```
# localectl status
System Locale: LANG=en_US.UTF-8
 VC Keymap: de-nodeadkeys
   X11 Layout: de
  X11 Variant: nodeadkeys
```

2. 사용 가능한 시스템 로케일 설정을 나열합니다.

```
$ localectl list-locales
C.UTF-8
...
en_US.UTF-8
en_ZA.UTF-8
en_ZW.UTF-8
...
```

3. syste 로케일 설정을 업데이트합니다.

예를 들면 다음과 같습니다.

+

```
# localectl set-locale LANG=en_US.UTF-8
```

참고

GNOME 터미널은 UTF8 이외의 시스템 로케일을 지원하지 않습니다. 자세한 내용은 시스템 로케일이 non-UTF8로 설정된 경우 Red Hat Knowledgebase 솔루션 The gnome-terminal application fails to start when the system locale is set to non-UTF8 에서 참조하십시오.

추가 리소스

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

4.4. 키보드 레이아웃 구성

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

절차

1. 사용 가능한 키맵을 나열하려면 다음을 수행합니다.

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

2. keymaps 설정의 현재 상태를 표시하려면 다음을 수행합니다.

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

3. 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)** 도움말 페이지

4.5. 텍스트 콘솔 모드에서 글꼴 크기 변경

가상 콘솔에서 글꼴 크기를 변경할 수 있습니다.

절차

1. 현재 사용 중인 글꼴 파일을 표시합니다.

```
# cat /etc/vconsole.conf

FONT="eurlatgr"
```

2. 사용 가능한 글꼴 파일을 나열합니다.

```
# ls -1 /usr/lib/kbd/consolefonts/*.psfu.gz

/usr/lib/kbd/consolefonts/eurlatgr.psfu.gz
/usr/lib/kbd/consolefonts/LatArCyrHeb-08.psfu.gz
/usr/lib/kbd/consolefonts/LatArCyrHeb-14.psfu.gz
/usr/lib/kbd/consolefonts/LatArCyrHeb-16.psfu.gz
/usr/lib/kbd/consolefonts/LatArCyrHeb-16+.psfu.gz
/usr/lib/kbd/consolefonts/LatArCyrHeb-19.psfu.gz
```

문자 세트 및 코드 페이지를 지원하는 글꼴 파일을 선택합니다.

3. 선택 사항: 글꼴 파일을 테스트하려면 임시로 로드합니다.

```
# setfont LatArCyrHeb-16.psfu.gz
```

setfont 유틸리티는 글꼴 파일을 즉시 적용하고 터미널에서 다른 글꼴 파일을 재부팅하거나 적용할 때까지 새 글꼴 크기를 사용합니다.

4. **/etc/vconsole.conf** 에 정의된 글꼴 파일로 돌아가려면 매개 변수 없이 **setfont** 을 입력합니다.

5. **/etc/vconsole.conf** 파일을 편집하고 **FONT** 변수를 부팅 시 로드해야 하는 글꼴 파일 RHEL로 설정합니다. 예를 들면 다음과 같습니다.

```
FONT=LatArCyrHeb-16
```

6. 호스트 재부팅

```
# reboot
```

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 못하도록 합니다.

5.1. SSH 키 쌍 생성

로컬 시스템에서 SSH 키 쌍을 생성하고 생성된 공개 키를 OpenSSH 서버에 복사하여 암호를 입력하지 않고 OpenSSH 서버에 로그인할 수 있습니다. 키를 생성하려는 각 사용자는 이 절차를 실행해야 합니다.

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

사전 요구 사항

- 키를 사용하여 OpenSSH 서버에 연결하려는 사용자로 로그인했습니다.
- OpenSSH 서버는 키 기반 인증을 허용하도록 구성됩니다.

절차

1. ECDSA 키 쌍을 생성합니다.

```
$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/<username>/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase): <password>
Enter same passphrase again: <password>
Your identification has been saved in /home/<username>/.ssh/id_ecdsa.
Your public key has been saved in /home/<username>/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:Q/x+qms4j7PCQ0qFd09iZEFHA+SqwBKRNaU72oZfaCI
<username>@<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 매개 변수 또는 Ed25519 키 쌍 없이 ssh-keygen 명령을 사용하여 RSA 키 쌍을 생성할 수도 있습니다. Ed25519 알고리즘은 FIPS-140과 호환되지 않으며 OpenSSH는 FIPS 모드에서 Ed25519 키와 함께 작동하지 않습니다.

2. 공개 키를 원격 머신에 복사합니다.

```
$ ssh-copy-id <username>@<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
<username>@<ssh-server-example.com>'s password:
…
Number of key(s) added: 1

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

&lt ***;username> @ <ssh-server-example.com*** >을 사용자 인증 정보로 바꿉니다.

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.

검증

- 키 파일을 사용하여 OpenSSH 서버에 로그인합니다.

```
$ ssh -o PreferredAuthentications=publickey <username>@<ssh-server-example.com>
```

추가 리소스

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

5.2. OPENSSH 서버에서 유일한 방법으로 키 기반 인증 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.
- 키를 사용하여 OpenSSH 서버에 이미 연결할 수 있습니다.
  자세한 내용은 SSH 키 쌍 생성 섹션을 참조하십시오.

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
```

3. 새 기본 설치 이외의 시스템에서 **PubkeyAuthentication** 매개변수가 설정되지 않았거나 **yes** 로 설정되어 있는지 확인합니다.

4. **Kbd interactiveAuthentication 지시문** 을 **no** 로 설정합니다.
   해당 항목은 구성 파일에서 주석 처리되며 기본값은 **yes** 입니다.

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

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

6. 콘솔 또는 대역 외 액세스를 사용하지 않고 원격으로 연결하는 경우 암호 인증을 비활성화하기 전에 키 기반 로그인 프로세스를 테스트합니다.

7. **sshd** 데몬을 다시 로드하여 변경 사항을 적용합니다.

```
# systemctl reload sshd
```

추가 리소스

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

5.3. SSH-AGENT를 사용하여 SSH 인증 정보 캐싱

SSH 연결을 시작할 때마다 암호를 입력하지 않으려면 **ssh-agent** 유틸리티를 사용하여 로그인 세션의 개인 SSH 키를 캐시할 수 있습니다. 에이전트가 실행 중이고 키가 잠금 해제되면 키의 암호를 다시 입력하지 않고도 이러한 키를 사용하여 SSH 서버에 로그인할 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 생성 섹션을 참조하십시오.

절차

1. 세션에서 **ssh-agent** 를 자동으로 시작하는 명령을 **~/.bashrc** 파일에 추가합니다.

   a. 선택한 텍스트 편집기에서 **~/.bashrc** 를 엽니다. 예를 들면 다음과 같습니다.

```
$ vi ~/.bashrc
```

   b. 파일에 다음 행을 추가합니다.

```
eval $(ssh-agent)
```

   c. 변경 사항을 저장하고 편집기를 종료합니다.

2. **~/.ssh/config** 파일에 다음 행을 추가합니다.

```
AddKeysToAgent yes
```

이 옵션과 **ssh-agent** 가 세션에서 시작되면 에이전트는 호스트에 처음 연결할 때만 암호를 입력하라는 메시지를 표시합니다.

검증

- 에이전트에서 캐시된 개인 키의 해당 공개 키를 사용하는 호스트에 로그인합니다. 예를 들면 다음과 같습니다.

```
$ ssh <example.user>@<ssh-server@example.com>
```

암호를 입력할 필요가 없습니다.

5.4. 스마트 카드에 저장된 SSH 키로 인증

스마트 카드에 ECDSA 및 RSA 키를 생성 및 저장하고 OpenSSH 클라이언트의 스마트 카드로 인증할 수 있습니다. 스마트 카드 인증은 기본 암호 인증을 대체합니다.

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
```

2. 공개 키를 원격 서버로 전송합니다. 이전 단계에서 만든 **keys.pub** 파일과 함께 **ssh-copy-id** 명령을 사용합니다.

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

3. ECDSA 키를 사용하여 < *ssh-server-example.com* >에 연결합니다. 키를 고유하게 참조하는 URI의 하위 집합만 사용할 수 있습니다. 예를 들면 다음과 같습니다.

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

OpenSSH는 **p11-kit-proxy** 래퍼를 사용하고 OpenSC PKCS #11 모듈이 **p11-kit** 툴에 등록되므로 이전 명령을 단순화할 수 있습니다.

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

```
[ssh-server-example.com] $
```

4. 선택 사항: **~/.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 <ssh-server-example.com>
Enter PIN for 'SSH key':
[ssh-server-example.com] $
```

이제 **ssh** 클라이언트 유틸리티에서 이 URI와 스마트 카드의 키를 자동으로 사용합니다.

추가 리소스

- **p11-kit(8)**, **opensc.conf(5)**, **pcscd(8)**, **ssh(1)**, 및 **ssh-keygen(1)** 도움말 페이지

5.5. 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
- 비표준 구성을 사용하여 애플리케이션 및 서비스에 대한 SELinux 구성
- firewalld를 사용하여 네트워크 트래픽 제어

6장. 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.

6.1. SYSLOG 메시지를 처리하는 서비스

다음 두 서비스는 **syslog** 메시지를 처리합니다.

- **systemd-journald** 데몬

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** 서비스

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

6.2. SYSLOG 메시지를 저장하는 로그 파일

/var/log 디렉터리에 있는 다음 로그 파일은 **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

참고

위에서 언급한 목록에는 일부 파일만 포함되어 있으며 **/var/log/** 디렉터리의 실제 파일 목록은 이 디렉터리에 로그인하는 서비스와 애플리케이션에 따라 다릅니다.

6.3. 명령줄을 사용하여 로그 보기

journal는 로그 파일을 보고 관리하는 데 도움이 되는 systemd의 구성 요소입니다. 기존 로깅과 관련된 문제를 해결하고 나머지 시스템과 밀접하게 통합되며 로그 항목에 대한 다양한 로깅 기술 및 액세스 관리를 지원합니다.

journalctl 명령을 사용하여 명령줄을 사용하여 시스템 저널의 메시지를 볼 수 있습니다.

표 6.1. 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 6.2. 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.
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 6.3. 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.

6.4. 웹 콘솔에서 로그 검토

RHEL 웹 콘솔에서 로그에 액세스, 검토 및 필터링하는 방법을 알아봅니다.

6.4.1. 웹 콘솔에서 로그 검토

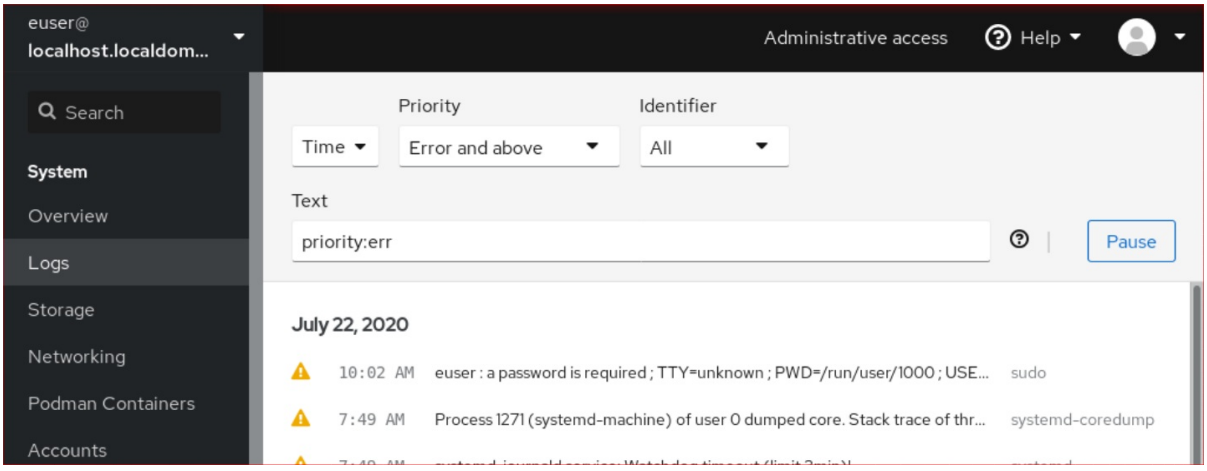
RHEL 9 웹 콘솔 로그 섹션은 **journalctl** 유틸리티의 UI입니다. 웹 콘솔 인터페이스에서 시스템 로그에 액세스할 수 있습니다.

사전 요구 사항

- RHEL 9 웹 콘솔을 설치했습니다.
- cockpit 서비스를 활성화했습니다.
- 사용자 계정이 웹 콘솔에 로그인할 수 있습니다.
  자세한 내용은 웹 콘솔 설치 및 활성화를 참조하십시오.

절차

1. RHEL 9 웹 콘솔에 로그인합니다.
   자세한 내용은 웹 콘솔에 로그인 을 참조하십시오.
2. **로그**를 클릭합니다.



3. 목록에서 선택한 로그 항목을 클릭하여 로그 항목 세부 정보를 엽니다.

참고

일시 중지 버튼을 사용하여 새 로그 항목이 표시되지 않도록 일시 중지할 수 있습니다. 새 로그 항목을 다시 시작하면 웹 콘솔은 **일시 정지** 버튼을 사용한 후 보고된 모든 로그 항목을 로드합니다.

로그는 시간, 우선 순위 또는 식별자별로 필터링할 수 있습니다. 자세한 내용은 웹 콘솔에서 로그 필터링을 참조하십시오.

6.4.2. 웹 콘솔에서 로그 필터링

웹 콘솔에서 로그 항목을 필터링할 수 있습니다.

사전 요구 사항

- RHEL 9 웹 콘솔을 설치했습니다.
- cockpit 서비스를 활성화했습니다.
- 사용자 계정이 웹 콘솔에 로그인할 수 있습니다.
  자세한 내용은 웹 콘솔 설치 및 활성화를 참조하십시오.

절차

1. RHEL 9 웹 콘솔에 로그인합니다.
   자세한 내용은 웹 콘솔에 로그인 을 참조하십시오.
2. **로그**를 클릭합니다.
3. 기본적으로 웹 콘솔은 최신 로그 항목을 표시합니다. 특정 시간 범위별로 필터링하려면 **시간** 드롭다운 메뉴를 클릭하고 선호하는 옵션을 선택합니다.
4. **오류 및** 심각도 로그 목록이 기본적으로 표시됩니다. 다른 우선 순위로 필터링하려면 **Error and above** 드롭다운 메뉴를 클릭하고 선호하는 우선 순위를 선택합니다.
5. 기본적으로 웹 콘솔은 모든 식별자에 대한 로그를 표시합니다. 특정 식별자에 대한 로그를 필터링하려면 **모두** 드롭다운 메뉴를 클릭하고 식별자를 선택합니다.
6. 로그 항목을 열려면 선택한 로그를 클릭합니다.

6.4.3. 웹 콘솔에서 로그 필터링을 위한 텍스트 검색 옵션

텍스트 검색 옵션 기능은 로그 필터링을 위한 다양한 옵션을 제공합니다. 텍스트 검색을 사용하여 로그를 필터링하려면 세 드롭다운 메뉴에 정의된 사전 정의된 옵션을 사용하거나 전체 검색을 직접 입력할 수 있습니다.

드롭다운 메뉴

검색의 기본 매개변수를 지정하는 데 사용할 수 있는 세 가지 드롭다운 메뉴가 있습니다.

- **시간**: 이 드롭다운 메뉴에는 다양한 검색 시간 범위에 대한 사전 정의된 검색이 포함되어 있습니다.

- **priority**: 이 드롭다운 메뉴에서는 다양한 우선 순위 수준에 대한 옵션을 제공합니다. **journalctl --priority** 옵션에 해당합니다. 기본 우선순위 값은 **Error 이상입니다**. 다른 우선 순위를 지정하지 않을 때마다 설정됩니다.

- **식별자**: 이 드롭다운 메뉴에서 필터링할 식별자를 선택할 수 있습니다. **journalctl --identifier** 옵션에 해당합니다.

Cryostatifiers

검색을 지정하는 데 사용할 수 있는 6가지 한정자가 있습니다. 로그 필터링 테이블에는 옵션이 포함되어 있습니다.

로그 필드

특정 로그 필드를 검색하려면 해당 콘텐츠와 함께 필드를 지정할 수 있습니다.

로그 메시지에서 자유형 텍스트 검색

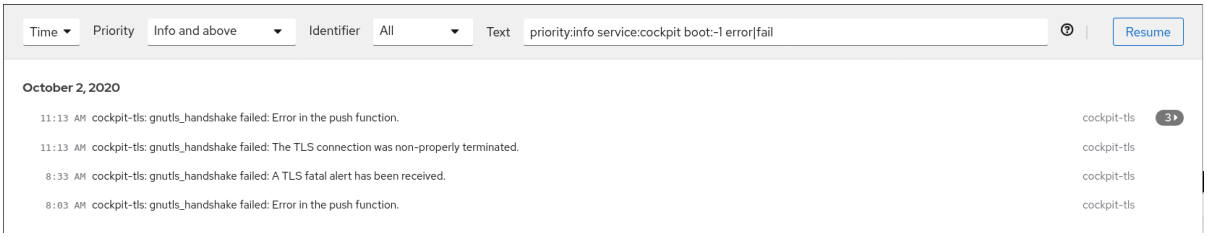
로그 메시지에서 선택한 텍스트 문자열을 필터링할 수 있습니다. 문자열은 정규식 형식일 수도 있습니다.

고급 로그 필터링 I

2020년 10월 22일 자정 이후 발생한 'systemd'로 식별된 모든 로그 메시지를 필터링하고 'JOB_TYPE' 필드는 'start' 또는 'restart'입니다.

1. type **identifier:systemd since:2020-10-22 JOB_TYPE=start,restart** to search field.

2. 결과를 확인하십시오.

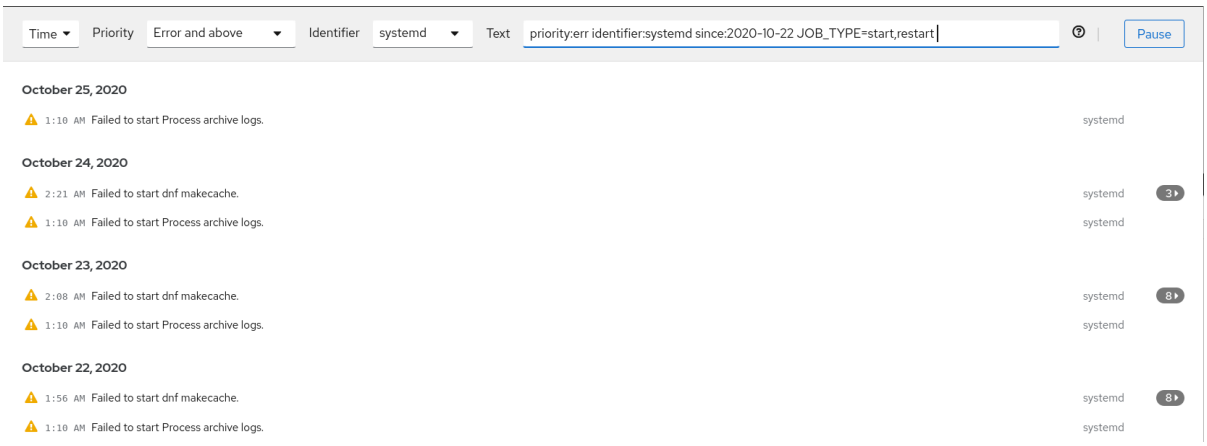


고급 로그 필터링 II

마지막 전에 부팅에 발생한 'cockpit.service' systemd 장치에서 제공되는 모든 로그 메시지를 필터링하고 메시지 본문에는 "error" 또는 "fail"가 포함됩니다.

1. **service:cockpit boot:-1 error|fail** to the search field를 입력합니다.

2. 결과를 확인하십시오.



6.4.4. 텍스트 검색 상자를 사용하여 웹 콘솔에서 로그를 필터링

웹 콘솔의 텍스트 검색 상자를 사용하여 다양한 매개변수에 따라 로그를 필터링할 수 있습니다. 검색에서는 필터링 드롭다운 메뉴, 정량자, 로그 필드 및 자유 형식 문자열 검색의 사용을 결합합니다.

사전 요구 사항

- RHEL 9 웹 콘솔을 설치했습니다.
- cockpit 서비스를 활성화했습니다.
- 사용자 계정이 웹 콘솔에 로그인할 수 있습니다.
  자세한 내용은 웹 콘솔 설치 및 활성화를 참조하십시오.

절차

1. RHEL 웹 콘솔에 로그인합니다.
   자세한 내용은 웹 콘솔에 로그인을 참조하십시오.
2. **로그**를 클릭합니다.
3. 드롭다운 메뉴를 사용하여 필터링하려는 세 가지 기본 한정자(시간 범위, 우선 순위 및 식별자)를 지정합니다.
   우선 순위 한정자는 항상 값이 있어야 합니다. 이를 지정하지 않으면 **Error 및 above** 우선 순위를 자동으로 필터링합니다. 설정한 옵션은 텍스트 검색 상자에 반영됩니다.
4. 필터링할 로그 필드를 지정합니다.
   여러 로그 필드를 추가할 수 있습니다.
5. 자유 형식 문자열을 사용하여 다른 항목을 검색할 수 있습니다. 검색 상자에는 정규식도 사용할 수 있습니다.

6.4.5. 로그 필터링 옵션

웹 콘솔에서 로그를 필터링하는 데 사용할 수 있는 여러 **journalctl** 옵션이 있습니다. 이 옵션은 유용할 수 있습니다. 이 중 일부는 웹 콘솔 인터페이스에서 드롭다운 메뉴의 일부로 이미 포함되어 있습니다.

표 6.4. 표

옵션 이름	사용법	참고
priority	메시지 우선 순위에 따라 출력을 필터링합니다. 단일 숫자 또는 텍스트 로그 수준을 사용합니다. 로그 수준은 일반적인 syslog 로그 수준입니다. 단일 로그 수준이 지정되면 이 로그 수준이 있는 모든 메시지 또는 하위(가장 중요함) 로그 수준이 표시됩니다.	**우선 순위** 드롭다운 메뉴에서 다룹니다.
identifier	지정된 syslog 식별자 SYSLOG_IDENTIFIER에 대한 메시지를 표시합니다. 여러 번 지정할 수 있습니다.	**ID** 드롭다운 메뉴에서 다룹니다.

옵션 이름	사용법	참고
팔로우	가장 최근의 저널 항목만 표시하고 저널에 추가되는 대로 새 항목을 지속적으로 출력합니다.	드롭다운에는 적용되지 않습니다.
서비스	지정된 **systemd** 장치에 대한 메시지를 표시합니다. 여러 번 지정할 수 있습니다.	드롭다운에 포함되지 않습니다. **journalctl --unit** 매개변수에 해당합니다.
부팅	특정 부팅의 메시지를 표시합니다. 양의 정수는 저널의 시작부터 부팅을 조회하고, 동일하거나 무의미한 0 정수는 저널이 끝날 때부터 부팅을 조회합니다. 따라서 1은 저널에 있는 첫 번째 부팅을 시간순으로, 2번 이상 두 번째 부팅을 의미합니다. -0은 마지막 부팅이고, 마지막 부팅은 -1이 됩니다.	**현재 부팅** 또는 **시간** 드롭다운 메뉴에서 **이전 부팅** 으로만 적용됩니다. 다른 옵션은 수동으로 작성해야 합니다.
이후	지정된 날짜보다 크거나 지정된 날짜보다 항목을 각각 표시하거나 그 이전 날짜를 표시합니다. 날짜 사양은 "2012-10-30 18:17:16" 형식이어야 합니다. 시간 부분이 생략되면 "00:00:00"이 사용됩니다. 초 구성 요소만 생략하면 ":00"이 사용됩니다. 날짜 구성 요소를 생략하면 현재 날짜로 가정합니다. 또는 문자열 "yesterday", "today", "tomorrow"가 이해되며, 이는 현재 날 전날, 현재 날짜 또는 하루 전의 00:00:00을 나타냅니다. "현재"는 현재 시간을 나타냅니다. 마지막으로, 현재 시간 전이나 후에 각각 "-" 또는 "+" 접두사를 지정하여 상대 시간을 지정할 수 있습니다.	드롭다운에는 적용되지 않습니다.

6.5. 추가 리소스

- **journalctl(1)** 시스템의 도움말 페이지
- 원격 로깅 솔루션 구성

7장. 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 호스트에서 로컬로 생성할 수 있습니다.

7.1. 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.

7.1.1. 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.*

7.1.2. 예약된 사용자 및 그룹 ID 구성

기본적으로 RHEL은 시스템 사용자 및 그룹에 대해 1000 미만의 사용자 및 그룹 ID를 예약합니다. 예약된 사용자 및 그룹 ID는 ***setup*** *패키지에서 찾을 수 있습니다.* ***UID_MIN*** *및* ***GID_MIN*** *값을 변경하기 전에 생성된 사용자 및 GID의 UID 및 GID는 변경되지 않습니다. 예약된 사용자 및 그룹 ID는 다음 사항에 설명되어 있습니다.*

```
/usr/share/doc/setup/uidgid
```

예약된 범위가 나중에 증가할 수 있으므로 새 사용자 및 그룹에 ID를 할당하려면 5000에서 시작합니다.

/etc/login.defs *파일에서* ***UID_MIN*** *및* ***GID_MIN*** *매개변수를 수정하여 기본값(1000) 이외의 시작 ID를 정의합니다.*

주의

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

절차

1. *편집기에서* ***/etc/login.defs*** *파일을 엽니다.*
2. ***UID_MIN*** *변수를 설정합니다. 예를 들면 다음과 같습니다.*

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

```
UID_MIN                  5000
```

3. ***GID_MIN*** *변수를 설정합니다. 예를 들면 다음과 같습니다.*

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

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

7.1.3. 사용자 개인 그룹

*RHEL은 사용자 개인 그룹 (**UPG**) 시스템 구성을 사용하므로 Linux 그룹을 더 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.*

UPG는 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 로컬 그룹 목록은 **/etc/group** 구성 파일에 저장됩니다.*

7.2. 사용자 계정 관리 시작하기

Red Hat Enterprise Linux는 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. 사용자 및 그룹, 시스템 사용자 및 시스템 그룹의 min/max ID를 정의하려면 **/etc/login.defs** 파일을 참조하십시오.*

- ***group:***

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

7.2.1. 명령줄 도구를 사용하여 계정 및 그룹 관리

다음 기본 명령줄 툴을 사용하여 사용자 계정 및 그룹을 관리합니다.

절차

- 새 사용자 계정을 생성합니다.

```
# useradd example.user
```

- example.user 에 속하는 사용자 계정에 새 암호를 할당합니다.

```
# passwd example.user
```

- 그룹에 사용자를 추가합니다.

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

추가 리소스

- **useradd(8)**, **passwd(1)**, **usermod(8)** 도움말 페이지

7.3. 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.

7.3.1. 명령줄에서 새 사용자 추가

useradd 유틸리티를 사용하여 새 사용자를 추가할 수 있습니다.

사전 요구 사항

- **루트** 액세스 권한이 있음

절차

- 새 사용자를 추가하고 다음을 사용합니다.

```
# useradd <options> <username>
```

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

예 7.1. 새 사용자 추가

사용자 ID **5000** 을 사용하여 사용자 **sarah** 를 추가하려면 다음을 사용하십시오.

```
# useradd -u 5000 sarah
```

검증

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

```
# id sarah
```

이 명령은 다음을 반환합니다.

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

추가 리소스

- **useradd** man 페이지

7.3.2. 명령줄에서 새 그룹 추가

groupadd 유틸리티를 사용하여 새 그룹을 추가할 수 있습니다.

사전 요구 사항

- 루트 액세스 권한이 있음

절차

- 새 그룹을 추가하려면 다음을 사용합니다.

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

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

예 7.2. 새 그룹 추가

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

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

검증

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

```
# getent group sysadmin
```

이 명령은 다음을 반환합니다.

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

추가 리소스

- **groupadd** 도움말 페이지

7.3.3. 명령줄에서 사용자를 보조 그룹에 추가

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

사전 요구 사항

- **root** 액세스 권한이 있음

절차

- 사용자 보조 그룹에 그룹을 추가하려면 다음을 사용합니다.

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

검증

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

```
# groups <username>
```

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

사전 요구 사항

- *루트 액세스 권한이 있음*

절차

1. *디렉터리를 생성합니다.*

```
# mkdir <directory-name>
```

2. *그룹을 생성합니다.*

```
# groupadd <group-name>
```

3. *그룹에 사용자를 추가합니다.*

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

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

```
# chgrp <group_name> <directory>
```

5. *사용자가 파일 및 디렉터리를 생성 및 수정하고 이 권한을 디렉터리에 적용하도록 **setgid** 비트를 설정하려면 쓰기 권한을 설정합니다.*

```
# chmod g+rwxs <directory>
```

검증

- *설정된 권한의 정확성을 확인하려면 다음을 사용합니다.*

```
# ls -ld <directory>
```

이 명령은 다음을 반환합니다.

```
*drwx__rws__r-x.* 2 root _group-name_ 6 Nov 25 08:45 _directory-name_
```

7.3.5. 명령줄에서 사용자 제거

명령줄을 사용하여 사용자 계정을 제거할 수 있습니다. 또한 아래에 언급된 명령은 사용자 계정을 제거하고 선택적으로 홈 디렉터리 및 구성 파일과 같은 사용자 데이터 및 메타데이터를 제거합니다.

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

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

- 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

7.4. 웹 콘솔에서 사용자 계정 관리

RHEL 웹 콘솔은 시스템 사용자 계정을 추가, 편집 및 제거하기 위한 그래픽 인터페이스를 제공합니다.

웹 콘솔에서 암호 만료를 설정하고 사용자 세션을 종료할 수도 있습니다.

7.4.1. 웹 콘솔을 사용하여 새 계정 추가

RHEL 웹 콘솔을 통해 사용자 계정을 시스템에 추가하고 계정에 대한 관리 권한을 설정할 수 있습니다.

사전 요구 사항

- **RHEL 9** 웹 콘솔을 설치했습니다.
- **cockpit** 서비스를 활성화했습니다.
- 사용자 계정이 웹 콘솔에 로그인할 수 있습니다.

  자세한 내용은 웹 콘솔 설치 및 활성화를 참조하십시오.

절차

1. **RHEL 9** 웹 콘솔에 로그인합니다.

자세한 내용은 웹 콘솔에 로그인 을 참조하십시오.

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.*

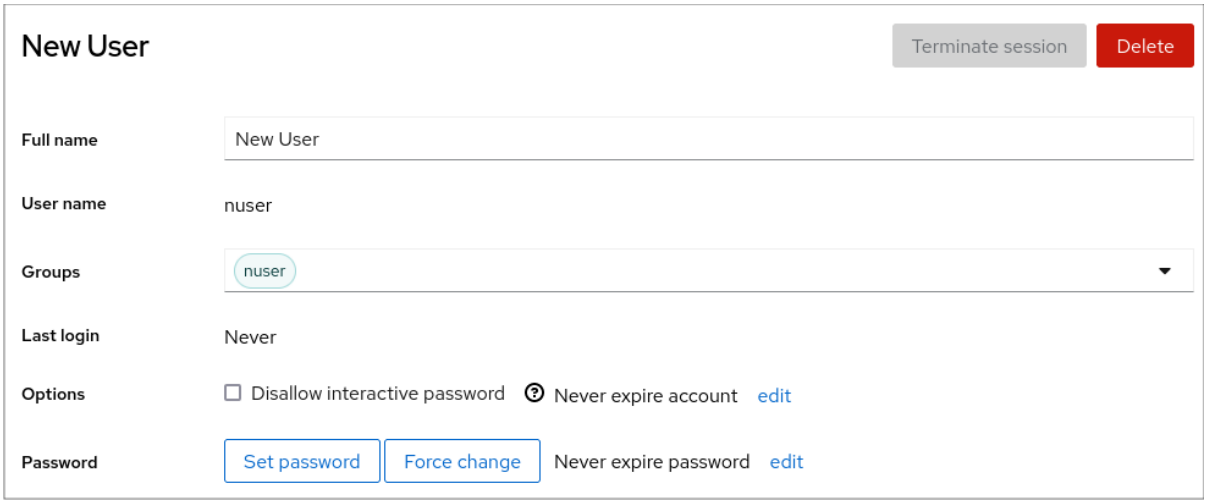
5. ***Password/Confirm** 필드에 암호를 입력하고 암호가 올바른지 확인하도록 다시 입력합니다.*

   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.*



검증

- *계정 설정에서 새 계정을 볼 수 있으며 해당 인증 정보를 사용하여 시스템에 연결할 수 있습니다.*

7.4.2. 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 사용자는 사용자가 시스템에 액세스하기 전에 다음 로그인 시도 시 암호를 변경해야 합니다.

사전 요구 사항

- ***RHEL 9** 웹 콘솔을 설치했습니다.*
- ***cockpit** 서비스를 활성화했습니다.*
- *사용자 계정이 웹 콘솔에 로그인할 수 있습니다.*

  자세한 내용은 웹 콘솔 설치 및 활성화를 참조하십시오.

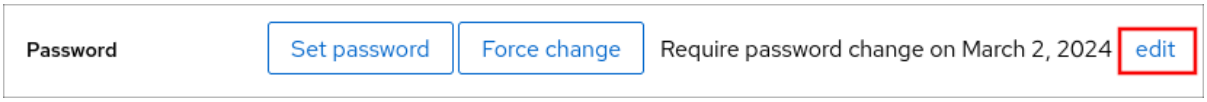
절차

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.

7.5. 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.

7.5.1. 기본 및 보조 사용자 그룹

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

RHEL에서 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, 1** 또는 여러 보조 그룹의 멤버일 수 있습니다.*

7.5.2. 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
```

사용자 이름을 제공하지 않으면 명령은 현재 사용자의 그룹 멤버십을 표시합니다. 첫 번째 그룹은 기본 그룹 다음에 선택적 보조 그룹이 옵니다.

*예 **7.3.** 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.

7.5.3. 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>
```

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.

- 사용자의 기본 그룹을 변경했는지 확인합니다.

```
$ groups <username>
```

7.5.4. 명령줄에서 사용자를 보조 그룹에 추가

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

사전 요구 사항

- ***root** 액세스 권한이 있음*

절차

- *사용자 보조 그룹에 그룹을 추가하려면 다음을 사용합니다.*

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

검증

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

```
# groups <username>
```

7.5.5. 보조 그룹에서 사용자 제거

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

사전 요구 사항

- ***root** 액세스 권한이 있음*

절차

- *보조 그룹에서 사용자 제거:*

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

검증

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

```
$ groups <username>
```

7.5.6. 사용자의 모든 보조 그룹 변경

사용자가 멤버로 유지하려는 보조 그룹 목록을 덮어쓸 수 있습니다.

사전 요구 사항

- root 액세스 권한이 있습니다.
- 보조 그룹이 있어야 합니다.

절차

- 사용자의 보조 그룹 목록을 덮어씁니다.

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

사용자를 한 번에 여러 보조 그룹에 추가하려면 그룹 이름을 쉼표와 중간 공백을 사용하여 구분합니다. 예: **wheel,developer**.

중요

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

검증

- 보조 그룹 목록을 올바르게 설정했는지 확인합니다.

```
# groups <username>
```

7.6. 루트 암호 변경 및 재설정

기존 **root** 암호가 더 이상 부적합하지 않으면 **root** 사용자와 루트가 아닌 사용자로 둘 다 변경할 수 있습니다.

7.6.1. root 사용자로 root 암호 변경

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

사전 요구 사항

- 루트 액세스 권한이 있음

절차

- 루트 암호를 변경하려면 다음을 사용합니다.

```
# passwd
```

암호를 변경하기 전에 현재 암호를 입력하라는 메시지가 표시됩니다.

7.6.2. 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.
- **sudo** 를 사용하여 **root**로 명령을 실행할 수 있는 권한이 있습니다.

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.

7.6.3. root 암호 재설정

root 사용자로 로그인할 수 없고 **sudo** 권한이 있는 **root**가 아닌 사용자가 없는 경우 **root** 암호를 재설정하거나 관리 **wheel** 그룹에 속하지 않는 경우 시스템을 특수 모드로 부팅하여 **root** 암호를 재설정할 수 있습니다. 이 모드에서 부팅 프로세스는 시스템이 **initramfs** 에서 실제 시스템으로 제어를 수행하기 전에 중지됩니다.

절차

1. 시스템을 재부팅하고 **GRUB** 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. 커서를 **linux** 로 시작하는 행 끝으로 설정합니다.

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

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
   ```

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

6. *chroot 환경을 입력합니다.*

```
# chroot /sysroot
```

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 종료하여 시스템을 재부팅합니다.*

```
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

1. *새 root 암호를 사용하여 root 사용자로 로그인합니다.*

2. *선택 사항: 현재 유효한 사용자 ID와 연결된 사용자 이름을 표시합니다.*

```
# whoami
```

8장. 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.

8.1. SUDOERS의 사용자 권한 부여

/etc/ sudo ers 파일 및 기본적으로 **/etc/sudoers.d/** 디렉터리에 있는 드롭인 파일은 **sudo** 명령을 사용하여 다른 사용자로 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.

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

규칙의 일부 또는 여러 세그먼트에서 **ALL**을 사용하면 심각한 보안 위험이 발생할 수 있습니다.

! 연산자를 사용하여 인수를 무효화할 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

- ***sudoers(5)** 도움말 페이지*

8.2. 그룹 멤버가 ROOT로 명령을 실행할 수 있도록 허용하는 SUDO 규칙 추가

*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.

- ***sudoers** 구성 파일에 나열된 사용자만 **sudo** 명령을 사용할 수 있습니다.*
- *명령은 **root** 쉘이 아닌 사용자의 쉘에서 실행됩니다. 그러나 모든 사용자에게 전체 **sudo** 권한이 부여된 경우와 같은 몇 가지 예외가 있습니다. 이러한 경우 사용자는 **root** 쉘에서 로 전환하고 명령을 실행할 수 있습니다. 예를 들면 다음과 같습니다.*
- ***sudo -i***
- ***sudo su -***

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

- **wheel** 그룹의 멤버로 로그인하여 다음을 실행합니다.

```
# sudo whoami
root
```

추가 리소스

- ***sudo(8), sudoers(5)*** *및* ***visudo(8)*** *도움말 페이지*

8.3. 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. 이는 사용자에게 전체* ***sudo*** *액세스 권한을 부여하거나 다음과 같은 이유로 사용자에게* ***root*** *암호를 제공하는 것보다 더 안전합니다.*

- *권한 있는 작업을 보다 세밀하게 제어합니다. 사용자가 전체 관리 액세스 권한을 부여하는 대신 특정 호스트에서 특정 작업을 수행하도록 허용할 수 있습니다.*
- *더 나은 로깅. 사용자가* ***sudo*** *를 통해 작업을 수행하면* ***root***뿐만 아니라 사용자 이름으로 작업이 기록됩니다.
- *투명한 제어. 사용자가* ***sudo*** *권한을 사용하려고 할 때마다 이메일 알림을 설정할 수 있습니다.*

사전 요구 사항

- *시스템에 대한* ***root*** *액세스 권한이 있습니다.*

절차

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

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

파일이 편집기에서 자동으로 열립니다.

2. ***/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 URL로 바꿉니다.*

- *(<run_as_user > : <run_as_group>) 명령을 실행할 수 있는 사용자 또는 그룹으로 바꿉니다. 이 섹션을 생략하면 < username> 에서 root로 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 재부팅하도록 허용하려면 다음을 입력합니다.

```
user1 host1.example.com = /bin/dnf, /sbin/reboot
```

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

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

2. *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 whoami
[sudo] password for <username>:
```

사용자의 **sudo** 암호를 입력합니다.

3. 권한이 올바르게 구성된 경우 **sudo**는 구성된 사용자로 명령을 실행합니다. 예를 들어 **dnf** 명령을 사용하면 다음 출력이 표시됩니다.

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

시스템에서 다음 오류 메시지를 반환하는 경우 **sudo**를 사용하여 명령을 실행할 수 없습니다.

```
<username> is not in the sudoers file. This incident will be reported.
```

+ 시스템에서 다음 오류 메시지를 반환하는 경우 구성이 올바르게 완료되지 않았습니다.

```
<username> is not allowed to run sudo on <host.example.com>.
```

+ 시스템에서 다음 오류 메시지를 반환하는 경우 명령은 사용자의 규칙에 올바르게 정의되지 않습니다.

```
`Sorry, user _<username>_ is not allowed to execute '_<path/to/command>_' as root on
_<host.example.com>_.`
```

추가 리소스

- **visudo(8)** 및 **sudoers(5)** 도움말 페이지

8.4. RHEL 시스템 역할을 사용하여 사용자 지정 SUDOERS 구성 적용

sudo RHEL 시스템 역할을 사용하여 관리 노드에 사용자 지정 **sudoers** 구성을 적용할 수 있습니다. 이렇게 하면 구성 효율성을 개선하고 보다 세분화된 제어를 통해 어떤 호스트에서 어떤 명령을 실행할 수 있

는지 정의할 수 있습니다.

사전 요구 사항

- 컨트롤 노드 및 관리형 노드를 준비했습니다.
- 관리 노드에서 플레이북을 실행할 수 있는 사용자로 제어 노드에 로그인되어 있습니다.
- 관리 노드에 연결하는 데 사용하는 계정에는 **sudo** 권한이 있습니다.

절차

1. 다음 콘텐츠를 사용하여 플레이북 파일(예: **~/playbook.yml**)을 생성합니다.

```
---
- name: "Configure sudo"
  hosts: managed-node-01.example.com
  tasks:
    - name: "Apply custom /etc/sudoers configuration"
      ansible.builtin.include_role:
        name: redhat.rhel_system_roles.sudo
      vars:
        sudo_sudoers_files:
          - path: "/etc/sudoers"
            user_specifications:
              - users:
                  - <user_name>
                hosts:
                  - <host_name>
                commands:
                  - <path_to_command_binary>
```

플레이북에 지정된 설정은 다음과 같습니다.

사용자

규칙이 적용되는 사용자 목록입니다.

호스트

규칙이 적용되는 호스트 목록입니다. 모든 호스트에 대해 모두 사용할 수 있습니다.

명령

규칙이 적용되는 명령 목록입니다. 모든 명령에는 모두 사용할 수 있습니다.

*플레이북에 사용되는 모든 변수에 대한 자세한 내용은 제어 노드의 **/usr/share/ansible/roles/rhel-system-roles.sudo/README.md** 파일을 참조하십시오.*

2. *플레이북 구문을 확인합니다.*

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

이 명령은 구문만 검증하고 잘못되었지만 유효한 구성으로부터 보호하지 않습니다.

3. ***Playbook**을 실행합니다.*

```
$ ansible-playbook ~/playbook.yml
```

검증

1. *관리 노드에서 플레이북이 새 규칙을 적용했는지 확인합니다.*

```
# cat /etc/sudoers | tail -n1
<user_name> <host_name>= <path_to_command_binary>
```

추가 리소스

- ***/usr/share/ansible/roles/rhel-system-roles.sudo/README.md file***
- ***/usr/share/doc/rhel-system-roles.sudo/sudo/ directory***

9장. 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.

9.1. 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.*

9.1.1. 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 **9.1.** 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 9.2. 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 디렉터리*

에 액세스할 수도 있음을 의미합니다.

- ***---** 는 다른 사용자가 디렉터리의 콘텐츠를 읽고, 쓰거나, 액세스할 수 있는 권한이 없음을 나타냅니다.*

  사용자 소유자가 아니거나 그룹으로, 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 **permissions**에 의해 변경됩니다. 기본 권한과 **mTLS**의 조합은 파일 및 디렉터리에 대한 기본 권한을 생성합니다.*

9.1.2. 사용자 파일 생성 모드 마스크

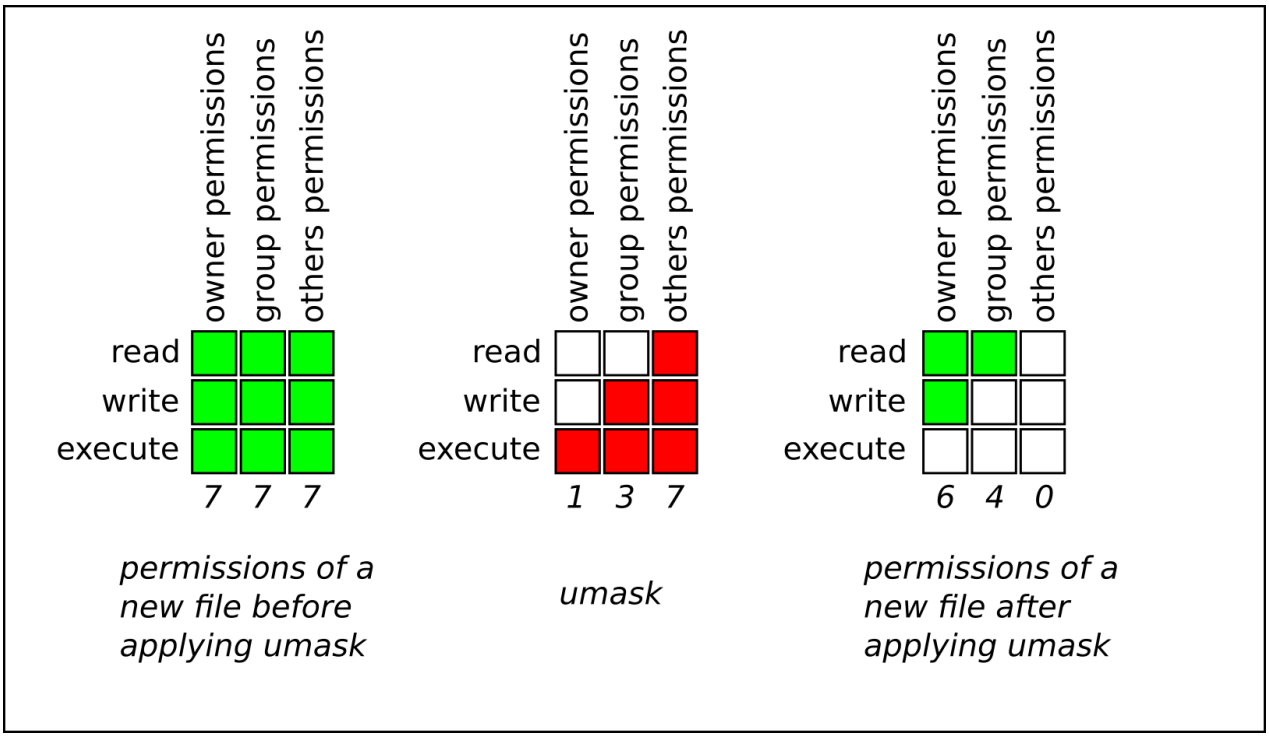
*사용자 파일 생성 모드 마스크(**balancer**)는 새로 생성된 파일 및 디렉터리에 대해 파일 권한이 설정되는 방법을 제어하는 변수입니다. **CloudEvent** 는 **Linux** 시스템의 전체 보안을 높이기 위해 기본 권한 값에서 권한을 자동으로 제거합니다. **jaeger** 는 심볼릭 또는 **8**진수 값으로 표현할 수 있습니다.*

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 **CloudEvent**는 **0022** 입니다.*

***messages**의 첫 번째 숫자는 특수 권한(**sticky bit, **)을 나타냅니다. **PATH**의 마지막 세 자리는 사용자 소유자(**u**), 그룹 소유자(**g**) 및 기타(**o**)에서 각각 제거된 권한을 나타냅니다.*

> ***예 9.3.** 파일을 생성할 때 **mTLS** 적용*
>
> *다음 예제에서는 기본 권한이 **777** 인 파일에 **8**진수 값이 **0137** 인 **umask** 를 적용하여 기본 권한이 **640** 인 파일을 생성하는 방법을 보여줍니다.*
>
> 
>

9.1.3. 기본 파일 권한

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

> ***예 9.4.** 디렉터리에 대한 기본 권한*
>
> *표준 사용자 또는 루트 사용자가 새 디렉터리를 생성하는 경우 **ScanSetting**은 **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

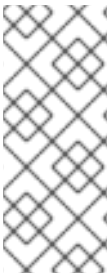
즉, 디렉터리 소유자는 디렉터리의 콘텐츠를 나열하고, 디렉터리 내의 항목을 생성, 삭제, 편집할 수 있으며, 디렉터리의 콘텐츠를 내림차순할 수 있습니다. 그룹 및 기타 그룹은 디렉터리의 콘텐츠만 나열하고 그 내용을 추측할 수 있습니다.

예 9.5. 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.*

9.1.4. 심볼릭 값을 사용하여 파일 권한 변경

chmod 유틸리티를 문자 및 기호 조합과 함께 사용하여 파일 또는 디렉터리에 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

- 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 9.6. 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
```

9.1.5. 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.

9.2. 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.

9.2.1. 액세스 제어 목록 설정

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

사전 요구 사항

- 루트 액세스 권한이 있습니다.

절차

- 특정 파일 또는 디렉터리에 대한 현재 **ACL**을 표시하려면 다음을 실행합니다.

```
$ getfacl file-name
```

file-name 을 파일 또는 디렉터리의 이름으로 바꿉니다.

- 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 시스템의 **setfacl** 도움말 페이지를 참조하십시오.

예 9.7. 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

- *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--
```

9.3. RECEIVER 관리

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

9.3.1. mTLS의 현재 값 표시

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

절차

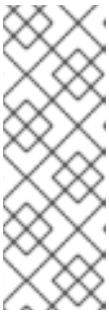
- ***symbolic mode**로 **topology**의 현재 값을 표시하려면 다음을 사용합니다.*

```
$ umask -S
```

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

```
$ umask
```

-



참고

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

9.3.2. 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 제거하려면**To 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 셸 세션에 대해 **mTLS**를 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, **permissions**를 **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 셸 세션에만 유효합니다.

9.3.3. 8진수 값을 사용하여 receiver 설정

8진수 값**(number)**과 함께 **wizard** 유틸리티를 사용하여 현재 셸 세션에 대한 **umask** 를 설정할 수 있

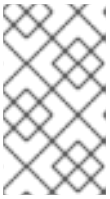
습니다.

절차

- 현재 쉘 세션에 대해 mTLS를 설정하려면 다음을 사용합니다.

```
$ umask octal_value
```

8진수_value 를 8진수 값으로 바꿉니다. 자세한 내용은 User file-creation mode mask 를 참조하십시오.



참고

jaeger 는 현재 쉘 세션에만 유효합니다.

9.3.4. 로그인이 아닌 쉘의 기본 mTLS 변경

/etc/bashrc 파일을 수정하여 표준 사용자의 기본 bash Cryostat를 변경할 수 있습니다.

사전 요구 사항

- 루트 액세스 권한이 있습니다.

절차

1. 편집기에서 /etc/bashrc 파일을 엽니다.

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

1. 변경 사항을 저장하고 편집기를 종료합니다.

9.3.5. 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.*

9.3.6. 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.

9.3.7. 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.*

10장. 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.

10.1. SYSTEMD 장치 파일 위치

다음 디렉토리 중 하나에서 단위 구성 파일을 찾을 수 있습니다.

*표 **10.1. 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
```

10.2. 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.

10.2.1. 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

10.2.2. 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

- 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 10.2. 사용 가능한 서비스 단위 정보

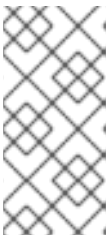
필드	설명
loaded	서비스 유닛이 로드되었는지 여부, 단위 파일의 절대 경로, 부팅 중에 장치를 시작할 수 있는지 여부를 확인합니다.
active	서비스 유닛이 실행 중인지 여부 및 타임스탬프를 제공합니다.
Main PID	프로세스 ID 및 해당 시스템 서비스의 이름입니다.
상태	해당 시스템 서비스에 대한 추가 정보입니다.
process	관련 프로세스에 대한 추가 정보.
cgroup	관련 제어 그룹(**cgroup**)에 대한 추가 정보.

- *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

10.2.3. systemd 장치 시작 및 중지

systemctl start 명령을 사용하여 현재 세션에서 시스템 서비스를 시작할 수 있습니다.

사전 요구 사항

- 루트 액세스 권한이 있습니다.

절차

- 현재 세션에서 시스템 서비스를 시작합니다.

```
# *systemctl start <systemd_unit> *
```

& lt;systemd_unit >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 **sendmail** 서비스를 실행 중이고 **postfix** 서비스를 시작하려고 하는 경우 **systemd** 는 먼저 **sendmail** 을 자동으로 중지합니다. 이 두 서비스는 충돌하고 동일한 포트에서 실행할 수 없기 때문입니다.

추가 리소스

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

10.2.4. 시스템 서비스 중지

현재 세션에서 시스템 서비스를 중지하려면 **systemctl 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

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

10.2.5. 시스템 서비스 다시 시작 및 다시 로드

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 권한이 있습니다.

절차

- 시스템 서비스를 다시 시작하십시오.

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

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

선택한 서비스 장치가 실행 중이 아닌 경우 이 명령이 시작됩니다.

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

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

- *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

- *시스템의 **systemctl man page***
- *시스템 서비스 상태 표시*

10.2.6. 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 권한이 있습니다.*

절차

- *단위가 마스크되었는지 확인합니다.*

```
# systemctl status <systemd_unit>
```

- *장치가 마스크된 경우 먼저 마스킹을 해제합니다.*

```
# systemctl unmask <systemd_unit>
```

- *부팅 시 서비스가 시작되도록 활성화합니다.*

```
# systemctl enable <systemd_unit>
```

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

선택적으로 명령에 --now 옵션을 전달하여 지금 장치를 시작합니다.

추가 리소스

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

10.2.7. 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 권한이 있습니다.

절차

- 부팅 시 시작할 서비스를 비활성화합니다.

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

& lt;name >을 비활성화하려는 서비스 단위의 이름으로 바꿉니다(예: bluetooth). 선택적으로 --now 명령을 전달하여 현재 실행 중인 서비스도 중지합니다.

- 선택 사항: 관리자가 실수로 장치를 시작하거나 다른 단위의 종속성으로 장치를 차단하려면 서비스를 마스크하십시오.

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

추가 리소스

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

10.3. 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.

10.3.1. 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.

- *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 10.3. 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

- *시스템의 systemd.special(7) 및 systemd.target(5) 도움말 페이지*

10.3.2. 부팅할 기본 대상 변경

default.target 심볼릭 링크는 시스템이 부팅해야 하는 systemd 대상을 나타냅니다. 시스템이 시작되면 systemd는 이 링크를 확인하고 정의된 대상으로 부팅됩니다. /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 권한이 있습니다.*

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. *선택 사항: 새 기본 대상으로 전환합니다.*

```
# systemctl isolate default.target
```

또는 시스템을 재부팅합니다.

추가 리소스

- *systemctl(1), systemd.special(7), bootup(7) 도움말 페이지*

10.3.3. 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. 수동으로 다른 대상으로 전환하는 것은 일시적인 작업일 뿐입니다. 호스트를 재부팅하면 **systemd**가 기본 대상으로 다시 부팅됩니다.*

절차

1. *선택 사항: 선택할 수 있는 대상 목록을 표시합니다.*

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

참고

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

2. *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

- *시스템의 **systemctl(1)** 도움말 페이지*

10.3.4. 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

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.

10.3.5. 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.

10.4. 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.

10.4.1. 시스템 종료

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

shutdown 유틸리티를 사용하면 다음과 같은 이점이 있습니다.

- **RHEL 8**에서는 **time** 인수를 사용하여 종료를 예약할 수 있습니다. 또한 시스템 종료가 예약되었음을 사용자에게 경고합니다.

10.4.2. 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 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 종료 취소

```
# shutdown -c
```

추가 리소스

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

10.4.3. 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.

10.4.4. 시스템을 다시 시작

시스템을 다시 시작하면 **systemd** 는 실행 중인 모든 프로그램 및 서비스를 중지하고 시스템이 종료되고 즉시 다시 시작합니다.

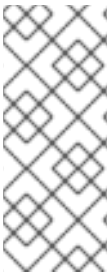
사전 요구 사항

- 루트 액세스 권한이 있습니다.

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.

10.4.5. 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
- **Hibernate**
- 하이브리드 **Sleep**
- **suspend-ECDHE-hibernate**

사전 요구 사항

- *루트 액세스 권한이 있습니다.*

절차

절전을 위한 적절한 방법을 선택합니다.

- ***Suspend ing**은 **RAM**에 시스템 상태를 저장하고 **RAM** 모듈을 제외한 대부분의 장치의 전원을 켭니다. 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. 시스템을 일시 중지하려면 다음을 실행합니다.*

```
# systemctl suspend
```

- ***Hibernate Hibernating**은 시스템 상태를 하드 디스크 드라이브에 저장하고 시스템의 전원을 켭니다. 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. 시스템을 **hibernate**하려면 다음을 실행합니다.*

```
# systemctl hibernate
```

- *하이브리드 절전 은 하이버네이션과 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. **hibernate** 및 시스템을 일시 중지하려면 다음을 실행합니다.*

```
# systemctl hybrid-sleep
```

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

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

10.4.6. 전원 버튼 동작 변경

컴퓨터의 전원 버튼을 누르면 기본적으로 시스템이 일시 중지되거나 종료됩니다. 기본 설정에 따라 이 동작을 사용자 지정할 수 있습니다.

10.4.6.1. 버튼을 누를 때 전원 버튼의 동작 변경 및 GNOME이 실행되지 않음

그래픽이 아닌 **systemd** 대상에서 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 편집하고 **HandlePowerKey=poweroff** 변수를 다음 옵션 중 하나로 설정합니다.

poweroff

컴퓨터를 종료합니다.

reboot

시스템을 재부팅합니다.

halt

시스템 중단을 시작합니다.

kexec

kexec 재부팅을 시작합니다.

일시 중단

시스템을 일시 중지합니다.

hibernate

*시스템 **hibernation**을 시작합니다.*

무시

아무것도 하지 마십시오.

예를 들어 전원 버튼을 누를 때 시스템을 재부팅하려면 다음 설정을 사용합니다.

```
HandlePowerKey=reboot
```

10.4.6.2. 버튼을 누를 때 전원 버튼의 동작 변경 및 GNOME이 실행 중

그래픽 로그인 화면 또는 그래픽 사용자 세션에서 전원 버튼을 누르면 기본적으로 시스템이 일시 중지됩니다. 이는 사용자가 전원 버튼을 물리적으로 누르거나 원격 콘솔에서 가상 전원 버튼을 누를 때 두 경우 모두 발생합니다. 다른 전원 버튼 동작을 선택할 수 있습니다.

절차

1. *다음 콘텐츠를 사용하여 **/etc/dconf/db/local.d/01-power** 파일에서 시스템 전체 설정에 대한 로컬 데이터베이스를 생성합니다.*

```
[org/gnome/settings-daemon/plugins/power]
power-button-action=<value>
```

& lt;value& gt;를 다음 전원 버튼 작업 중 하나로 바꿉니다.

없음

아무것도 하지 않습니다.

일시 중단

시스템을 일시 중지합니다.

hibernate

***Hibernates the system**을 실행합니다.*

대화형

사용자에게 수행할 작업을 요청하는 팝업 쿼리를 표시합니다.

*대화형 모드를 사용하면 전원 버튼을 누를 때 **60**초 후에 시스템이 자동으로 꺼집니다. 그러나 팝업 쿼리에서 다른 동작을 선택할 수 있습니다.*

2. *선택 사항: 사용자 설정을 재정의하고 사용자가 변경하지 못하도록 합니다. **/etc/dconf/db/local.d/locks/01-power** 파일에 다음 구성을 입력합니다.*

```
/org/gnome/settings-daemon/plugins/power/power-button-action
```

3. *시스템 데이터베이스를 업데이트합니다.*

```
# dconf update
```

4. *시스템 전체 설정이 적용되려면 로그아웃한 후 다시 로그인합니다.*

11장. 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. 사용자 공간 데몬은 커널에서 실행되는 시스템 클럭을 업데이트합니다. Red Hat Enterprise Linux 8부터 NTP 프로토콜은 chronyd 데몬에서 구현되며 chrony 패키지의 리포지토리에서 사용할 수 있습니다.

11.1. CHRONY 제품군 소개

NTP(Network Time Protocol) 의 구현은 chrony 입니다. 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 다양한 조건에서 잘 작동합니다.

- 간헐적인 네트워크 연결 포함
- 고도로 혼잡한 네트워크
- 온도 변경 (일반 컴퓨터 클럭은 온도에 민감합니다)
- 지속적으로 실행되지 않거나 가상 머신에서 실행되는 시스템입니다.

인터넷을 통해 동기화된 두 시스템 간의 일반적인 정확도는 몇 밀리초 내에 있으며 **10**초 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.

chronyd 데몬은 명령줄 유틸리티 **chronyc** 를 통해 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 인스턴스에서만 명령을 허용하지만 원격 호스트의 모니터링 명령도 수락하도록 구성할 수 있습니다. 원격 액세스는 제한되어야 합니다.

11.2. 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.

11.3. CHRONY 사용

다음 섹션에서는 chronyd 를 시작하고 중지하는 방법과 chrony 가 동기화되었는지 확인하는 방법을 설명합니다. 섹션은 시스템 Clock을 수동으로 조정하는 방법도 설명합니다.

11.3.1. chrony 관리

chronyd 를 시작, 중지 및 확인할 수 있습니다.

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
```

11.3.2. chrony가 동기화되었는지 확인

chrony 가 tracking,sources 및 sourcestats 명령을 사용하여 동기화되었는지 확인할 수 있습니다.

절차

1. *chrony 추적을 확인하려면 다음을 입력합니다.*

```
$  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. *chronyc sources 명령은 chronyd 가 액세스하는 현재 시간 소스에 대한 정보를 표시합니다.*

```
$ 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

- 시스템의 **chronyc(1)** 도움말 페이지

11.3.3. 시스템 시계 수동 조정

시스템 시계를 수동으로 조정할 수 있습니다.

절차

- 시스템 시계를 즉시 단계화하려면 슬루를 통해 진행 중인 조정 사항을 바이패스하려면 다음을 입력합니다.

```
# chronyc makestep
```

중요

rtcfile 지시문을 사용하는 경우 실시간 클럭을 수동으로 조정할 수 없습니다. 임의의 조정으로 **chrony**의 실시간 클럭 드리프트가 발생하는 비율을 측정해야 합니다.

11.3.4. 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.

NetworkManager는 인터페이스 재구성, 중지 또는 시작 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.

절차

- **chrony** 디스패치 스크립트를 비활성화하려면 **/dev/null** 에 대한 심볼릭 링크를 만듭니다.

```
# ln -f -s /dev/null /etc/NetworkManager/dispatcher.d/20-chrony-onoffline
```

참고

이러한 변경 후 **NTP** 서버는 항상 온라인 상태로 유지됩니다.

11.3.5. 격리된 네트워크에서 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.*

*격리된 네트워크에서 시스템에 대한 **chrony** 를 설정하려면 아래 설명된 단계를 따르십시오.*

절차

1. *서버가 되도록 선택한 시스템에서 **/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 <subnet>
```

*여기서 **<subnet >**은 클라이언트가 연결할 수 있는 네트워크입니다. **CIDR(Classless Inter-Domain Routing)** 표기법을 사용하여 서브넷을 지정합니다.*

2. *서버의 직접 클라이언트가 되도록 선택한 시스템에서 다음과 같이 **/etc/chrony.conf** 를 편집합니다.*

```
server <server_fqdn>
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 <server_ip_address>
```

*여기서 **<server_fqdn >**은 서버의 호스트 이름이며 **< server_ip_address >**는 서버의 주소입니다. 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 **ID**가 가장 작은 서버만 로컬 참조가 활성 상태이고 다른 서버가 동기화됩니다. 서버가 실패하면 다른 서버가 대신합니다.*

11.3.6. 원격 모니터링 액세스 구성

***chronyc** 유틸리티는 다음 방법을 사용하여 **chronyd** 에 액세스할 수 있습니다.*

- ***IPv4** 또는 **IPv6**.*
- *루트 및 **chrony** 사용자가 로컬로 액세스할 수 있는 도메인 소켓.*

*기본적으로 **chronyc** 는 **Unix** 도메인 소켓에 연결됩니다. 기본 경로는 **/var/run/chrony/chronyd.sock** 입니다. 이 연결에 실패하면 **chronyc** 는 **127.0.0.1**에 연결을 시도한 다음 **::1***

***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 다음 방법을 사용하여 구성할 수 있습니다.

- **chronyd** 의 구성 파일에서 **cmdallow** 지시문을 사용할 수 있습니다.
- **chronyc** 에서 **cmdallow** 명령을 실행합니다.

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 로컬 인터페이스에서 수신 대기하도록 **chrony**를 구성합니다.

```
bindcmdaddress 0.0.0.0
```

및

-

```
bindcmdaddress ::
```

2. 원격 IP 주소, 네트워크 및 서브넷의 명령을 허용합니다.

   /etc/chrony.conf 파일에 다음 내용을 추가합니다.

```
cmdallow 192.168.1.0/24
```

```
cmdallow 2001:db8::/64
```

3. 방화벽에서 포트 323을 열어 원격 시스템에서 연결을 허용합니다.

```
# firewall-cmd --permanent --add-port=323/udp
```

4. 방화벽 구성을 다시 로드합니다.

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

추가 리소스

- 시스템의 chrony.conf(5) 도움말 페이지

11.3.7. 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 11.1. 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 컨트롤 노드 및 관리형 노드 준비

11.3.8. 추가 리소스

- 시스템의 **chronyc(1)** 및 **chronyd(8)** 도움말 페이지
- 자주하는 질문

11.4. HW 타임스탬프링이 있는 CHRONY

일부 NIC(네트워크 인터페이스 컨트롤러)의 하드웨어 타임스탬프는 들어오고 나가는 패킷의 정확한 타임스탬프를 제공합니다. NTP 타임스탬프는 일반적으로 시스템 클럭을 사용하여 커널 및 chronyd 에 의해 생성됩니다. 그러나 HW 타임스탬프ing이 활성화되면 NIC는 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 마이크로 초의 정확도가 가능할 수 있습니다.

하드웨어 타임스탬프를 사용하는 시간 동기화를 위한 또 다른 프로토콜은 PTP 입니다.

NTP 와 달리PTP 는 네트워크 스위치 및 라우터의 지원에 의존합니다. 동기화의 최상의 정확성을 달성하려면 PTP 가 지원되는 스위치 및 라우터가 있는 네트워크에서 PTP 를 사용하고 이러한 스위치 및 라우터가 없는 네트워크에서 NTP 를 선호합니다.

11.4.1. 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.

절차

- 장치의 타임스탬프 기능 및 관련 PTP 하드웨어 클록을 표시합니다.

```
# ethtool -T enp1s0
```

11.4.2. 하드웨어 타임스탬프 활성화

/etc/chrony.conf 파일의 hwtimestamp 지시문을 사용하여 하나 이상의 인터페이스에서 하드웨어 타임스탬프를 활성화할 수 있습니다. 지시문은 단일 인터페이스를 지정하거나 와일드카드 문자를 사용하여 이를 지원하는 모든 인터페이스에서 하드웨어 타임스탬프를 활성화할 수 있습니다.

절차

1.

/etc/chrony.conf 파일을 편집하고 다음과 같이 변경합니다.

a.

하드웨어 타임스탬프를 지원하는 인터페이스에 대한 hwtimestamp 설정을 추가합니다. 예를 들면 다음과 같습니다.

```
hwtimestamp enp1s0
hwtimestamp eno*
```

*ptp4l 과 같은 다른 애플리케이션이 없는 경우 * 와일드카드를 사용할 수 있습니다.*

b.

서버 설정에 minpoll 및 maxpoll 옵션을 추가하여 짧은 클라이언트 폴링 간격을 구성합니다. 예를 들면 다음과 같습니다.

```
server ntp.example.comlocal minpoll 0 maxpoll 0
```

하드웨어 타임스탬프의 경우 시스템 클럭의 오프셋을 최소화하려면 기본 범위(641024 초)보다 짧은 폴링 간격을 구성해야 합니다.

c.

xleave 옵션을 서버 설정에 추가하여 NTP 인터리브 모드를 활성화합니다.

```
server ntp.example.comlocal minpoll 0 maxpoll 0 xleave
```

이 설정을 사용하면 chrony가 패킷을 보낸 후에만 하드웨어 전송 타임스탬프를 가져옵니다. 이 동작은 서버가 응답하는 패킷에 타임스탬프를 저장하지 못하도록 합니다. xleave 옵션을 사용하면 chrony가 전송 후 생성된 전송 타임스탬프를 수신할 수 있습니다.

d.

선택 사항: 서버에 대한 클라이언트 액세스 로깅에 할당된 최대 메모리 크기를 늘립니다. 예를 들면 다음과 같습니다.

```
clientloglimit 100000000
```

기본 서버 구성을 사용하면 몇 수천 개의 클라이언트가 인터리브 모드를 동시에 사용할 수 있습니다. clientloglimit 설정의 값을 늘리면 많은 클라이언트에 대해 서버를 구성할 수 있습니다.

2. *chronyd 서비스를 다시 시작합니다.*

```
# systemctrl restart chronyd
```

검증

1. *선택 사항: /var/log/messages 로그 파일에서 하드웨어 시간 샘플링이 활성화되었는지 확인합니다.*

```
chronyd[4081]: Enabled HW timestamping on enp1s0
chronyd[4081]: Enabled HW timestamping on eno1
```

2. *chronyd가 NTP 클라이언트 또는 피어로 구성된 경우 전송 및 수신 타임스탬프 모드 및 임시 모드를 표시합니다.*

```
# chronyc ntpdata

Output:

[literal,subs="+quotes,verbatim,normal"]


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
```

3.

NTP 측정의 안정성을 보고합니다.

```
# chronyc sourcestats

Output:

[literal,subs="+quotes,verbatim,normal"]
....

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
....
```

*이 안정성은 **Std Dev** 열에 보고됩니다. 하드웨어 타임스탬프가 활성화된 경우 **NTP** 측정의 안정성은 일반 부하에서 수십 또는 수백 나노초이어야 합니다.*

11.4.3. PTP-NTP 브리지 구성

***PTP(Precision Time Protocol)** 기본 **timeserver**를 **PTP**를 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.*

절차

1.

***linuxptp** 패키지에서 **ptp4l** 및 **phc2sys** 프로그램을 구성하여 **PTP** 를 사용하여 시스템 클럭을 동기화하도록 하나의 인터페이스를 사용합니다.*

1.

*다른 인터페이스를 사용하여 시스템 시간을 제공하도록 **chronyd** 를 구성합니다.*

```
bindaddress 203.0.113.74
hwtimestamp enp1s0
local stratum 1
```

2.

***chronyd** 서비스를 다시 시작합니다.*

```
# systemctrl restart chronyd
```

11.5. 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 설정을 /etc/sysconfig/chronyd 파일에 추가하여 chronyd 서비스의 OSPP 프로필 및 FIPS 모드를 비활성화할 수 있습니다.

11.5.1. 클라이언트에서 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

- 시간 서버는 NTS를 지원합니다.

절차

/etc/crony.conf 파일을 편집하고 다음과 같이 변경합니다.

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 않도록 다음 설정을 추가합니다.

```
ntsdumpdir /var/lib/chrony
```

3. **DHCP** 에서 제공하는 **NTP(Network Time Protocol)** 서버와의 동기화를 비활성화하도록 다음 설정을 주석 처리하거나 제거하십시오.

```
sourcedir /run/chrony-dhcp
```

4. **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.netnod.se   NTS     1   15  256  33m    0    0    8  100
ptbtime1.ptb.de   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.netnod.se   1       6   377    44   +237us[ +237us] +/-   23ms
ptbtime1.ptb.de    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

- 시스템의 **chrony.conf(5)** 도움말 페이지

11.5.2. 시간 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. **/etc/chrony.conf** 파일을 편집하고 다음과 같이 변경합니다.

```
ntsserverkey /etc/pki/tls/private/<ntp-server.example.net>.key
ntsservercert /etc/pki/tls/certs/<ntp-server.example.net>.crt
```

2. **chrony** 사용자가 파일을 읽을 수 있도록 개인 키와 인증서 파일에 대한 권한을 설정합니다. 예를 들면

```
# chown root:chrony /etc/pki/tls/private/<ntp-server.example.net>.key
/etc/pki/tls/certs/<ntp-server.example.net>.crt

# chmod 644 /etc/pki/tls/private/<ntp-server.example.net>.key /etc/pki/tls/certs/<ntp-
server.example.net>.crt
```

3. **ntsdumpdir /var/lib/chrony** 설정이 있는지 확인합니다.

4. *firewalld에서 필요한 포트를 엽니다.*

```
# firewall-cmd --permannent --add-port={323/udp,4460/tcp}
# firewall-cmd --reload
```

5. *chronyd 서비스를 다시 시작하십시오.*

```
# systemctl restart chronyd
```

검증

1. *클라이언트 시스템에서 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.

2. *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 0이 아닌 NTP 패킷 필드인 경우 클라이언트가 NTS-KE 포트에 연결하고 인증된 NTP 요청을 보낼 수 있었습니다.

12장. 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.

12.1. REAR 설정 및 수동으로 백업 생성

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.

  ReaR과 함께 완전 통합 및 기본 제공 방법인 **NETFS** 백업 메서드를 사용할 수 있습니다.

절차

1. **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 생성합니다. 예를 들어 **NETFS** 백업 방법의 경우 다음 명령을 실행합니다.

```
# rear mkbackuponly
```

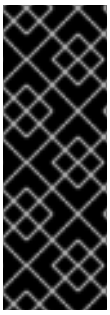
또는 다음 명령을 실행하여 단일 단계에서 복구 시스템 및 백업을 생성할 수 있습니다.

```
# rear mkbackup
```

이 명령은 **rear mkrescue** 및 **rear mk backuponly** 명령의 기능을 결합합니다.

12.2. 64비트 IBM Z 아키텍처에서 REAR RESCUE 이미지 사용

기본 **Relax** 및 **Recover(ReaR)** 기능은 이제 **64**비트 **IBM Z** 아키텍처에서 사용할 수 있으며 **RHEL 9.2** 이후 완전히 지원됩니다. **IBM Z**에서 **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

- **Rear**가 설치되어 있습니다.
  - **ReaR**을 설치하려면 **dnf install rear** 명령을 실행합니다.

절차

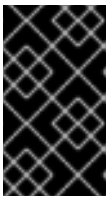
다음 변수를 **/etc/rear/local.conf** 에 추가하여 **64**비트 **IBM Z** 아키텍처에서 복구 이미지를 생성하도록 **ReaR**을 구성합니다.

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 **64**비트 **IBM Z** 아키텍처에서 지원되지 않습니다.

3. 선택 사항: 커널 및 **initrd** 파일을 저장하도록 **OUTPUT_URL** 변수를 구성할 수도 있습니다. 기본적으로 **OUTPUT_URL** 은 **etcdctl_URL**에 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 **3**단계에서 만든 **ReaR** 커널 및 **initrd** 파일을 사용하고, **zipl** 부트 로더, 커널, **initrd**를 사용하여 준비한 **zipl boot loader, kernel** 및 **initrd**를 사용하여 준비한 **FCP(Direct Attached Storage Device)** 또는 **FCP(Fibre Channel Protocol)**에서 부팅하십시오. 자세한 내용은 *준비 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/VM에서 설치*
- *준비 DASD 사용*.

12.3. REAR EXCLUSIONS

ReaR 유틸리티는 복구 프로세스 중 **/var/lib/rear/layout/disklayout.conf** 레이아웃 파일의 설명에 따라 복구된 시스템의 디스크에서 복구 이미지가 생성된 원래 시스템의 스토리지 레이아웃을 다시 생성합니다. 스토리지 레이아웃에는 파티션, 볼륨 그룹, 논리 볼륨, 파일 시스템 및 기타 스토리지 구성 요소가 포함됩니다.

Rear는 복구 이미지를 생성할 때 레이아웃 파일을 생성하고 이 파일을 이미지에 포함합니다. **rear savelayout** 명령을 사용하여 레이아웃 파일을 생성할 수도 있습니다. 이렇게 하면 전체 복구 이미지를 생성하지 않고도 레이아웃 파일을 빠르게 생성하고 검사할 수 있습니다.

레이아웃 파일은 특정 예외를 제외하고 원래 시스템의 전체 스토리지 레이아웃을 설명합니다. **ReaR**은 레이아웃 파일에서 일부 스토리지 구성 요소를 제외하고 복구 중에 재생성되지 않습니다. 레이아웃에서 스토리지 구성 요소를 제외하는 것은 다음 구성 변수에 의해 제어됩니다.

- **AUTOEXCLUDE_DISKS**
- **AUTOEXCLUDE_MULTIPATH**

- *AUTOEXCLUDE_PATH*
- *EXCLUDE_RECREATE*

/usr/share/rear/conf/default.conf 파일에서 구성 변수의 기본값을 보고 로컬 /etc/rear/local.conf 구성 파일에서 이러한 값을 변경할 수 있습니다.

레이아웃 파일의 구문 및 일부 스토리지 구성 요소를 제외하는 데 사용할 수 있는 구성 변수에 대한 자세한 내용은 ReaR 사용자 가이드의 레이아웃 구성 장을 참조하십시오. 이 구성은 /usr/share/doc/rear/relax-and-recover-user-guide.html.

또한 내부 NETFS 및 RSYNC 백업 방법으로 백업되는 파일을 구성할 수도 있습니다. 파일 시스템이 레이아웃 파일에 포함된 경우 기본적으로 모든 마운트된 로컬(디스크 기반) 파일 시스템의 파일은 rear mkbackup 또는 rear mkbackuponly 명령으로 백업됩니다. AUTOEXCLUDE_DISKS,AUTOEXCLUDE_MULTIPATH,AUTOEXCLUDE_PATH, EXCLUDE_RECREATE 및 EXCLUDE_RECREATE 와 같은 변수에 의해 제어되는 레이아웃 파일에서 일부 파일 시스템도 제외합니다. Cryostat _PROG_EXCLUDE 구성 변수를 사용하여 레이아웃 파일에서 파일 시스템을 제외하지 않고 백업에서 일부 파일 또는 디렉터리 트리를 제외할 수도 있습니다. 파일 시스템의 모든 파일 및 디렉터리가 이러한 방식으로 제외되면 복구 중에 파일 시스템을 다시 생성하지만 백업에 복원할 데이터가 포함되어 있지 않기 때문에 비어 있습니다. 이는 임시 데이터가 포함되어 있지 않은 파일 시스템이나 ReaR과 무관한 방법을 사용하여 백업되는 데이터에 유용합니다.

Cryo stat_PROG_EXCLUDE 변수는 tar 또는 rsync로 전달되는 glob Cryostat 스타일 와일드카드 패턴의 배열입니다. 구성 파일을 읽을 때 쉘이 확장되지 않도록 패턴을 인용해야 합니다. 이 변수의 기본값은 /usr/share/rear/conf/default.conf 파일에 설정됩니다. 기본값에는 /tmp 디렉토리 아래의 모든 파일과 디렉터리를 제외하지만 /tmp 디렉토리 자체는 제외하는 /tmp/ 패턴이 포함됩니다.*

다른 파일 및 디렉터리를 제외해야 하는 경우 기본값을 유지하기 위해 변수를 재정의하는 대신 패턴을 변수에 추가합니다. 예를 들어 /data/temp 디렉터리 아래의 모든 파일 및 디렉터리를 제외하려면 다음을 사용합니다.

```
# BACKUP_PROG_EXCLUDE+=( '/data/temp/*' )
```

rear mkbackup 명령은 로그에 백업 제외 패턴을 나열합니다. 로그 파일은 /var/log/rear 디렉토리에서 찾을 수 있습니다. 이는 전체 시스템 복구를 수행하기 전에 제외된 규칙을 확인하는 데 사용할 수 있습니다. 예를 들어 로그에 다음 항목이 포함될 수 있습니다.

```
2025-04-29 10:17:41.312431050 Making backup (using backup method NETFS)
2025-04-29 10:17:41.314369109 Backup include list (backup-include.txt contents):
2025-04-29 10:17:41.316197323   /
2025-04-29 10:17:41.318052001 Backup exclude list (backup-exclude.txt contents):
2025-04-29 10:17:41.319857125   /tmp/*
2025-04-29 10:17:41.321644442   /dev/shm/*
2025-04-29 10:17:41.323436363   /var/lib/rear/output/*
```

여기에서 전체 루트 파일 시스템은 백업에 포함되어 있으며 /tmp,/dev/shm 및 /var/lib/rear/output 디렉터리 아래에 있는 모든 파일과 디렉터리를 제외하고 백업에 포함됩니다.