



U Y U N I

Uyuni 2022.11

클라이언트 구성 가이드

2023년01월23일



차례

클라이언트 구성 안내서 개요	1
1. 지원되는 클라이언트 및 기능	2
1.1. 지원되는 클라이언트 시스템	2
1.2. 지원되는 도구 패키지	3
1.3. 지원하는 SUSE 및 openSUSE 클라이언트 기능	4
1.4. 지원하는 SLE Micro 및 openSUSE MicroOS 클라이언트 기능	6
1.5. 지원하는 Alibaba Cloud Linux 기능	8
1.6. 지원하는 AlmaLinux 기능	10
1.7. 지원되는 Amazon Linux 기능	12
1.8. 지원하는 CentOS 기능	14
1.9. 지원하는 Debian 기능	16
1.10. 지원하는 Oracle 기능	18
1.11. 지원하는 Red Hat Enterprise Linux 기능	20
1.12. 지원하는 Rocky Linux 기능	22
1.13. 지원하는 Ubuntu 기능	24
2. 구성 기본	27
2.1. 소프트웨어 채널	27
2.1.1. SUSE Package Hub가 제공하는 패키지	27
2.1.2. AppStream이 제공하는 패키지	28
2.1.3. EPEL이 제공하는 패키지	28
2.1.4. SUSE Linux Enterprise 클라이언트의 Unified Installer 업데이트 채널	28
2.1.5. 소프트웨어 리포지토리	29
2.1.6. 소프트웨어 제품	30
2.2. 부트스트랩 리포지토리	30
2.2.1. 부트스트랩 리포지토리 생성 준비	30
2.2.2. 자동 모드를 위한 옵션	31
2.2.3. 부트스트랩 리포지토리를 수동으로 생성	32
2.2.4. 부트스트랩 및 사용자 지정 채널	33
2.3. 활성화 키	33
2.3.1. 여러 활성화 키 결합	35
2.3.2. 활성화 키	35
2.3.3. 활성화 키 모범 사례	36
2.4. GPG 키	37
2.4.1. 클라이언트의 GPG 키 신뢰	38

클라이언트 구성 안내서 개요

업데이트 날짜: 2023-01-23

Uyuni 설치 후 수행하는 첫 단계는 클라이언트를 등록하는 일입니다. Uyuni에서 등록 클라이언트를 관리하는 작업에 대부분의 시간을 소요합니다.

Uyuni는 다양한 클라이언트 기술과 호환됩니다. 즉, 다양한 하드웨어 옵션으로 SUSE Linux Enterprise 또는 다른 Linux 운영 체제를 실행하는 기존 또는 Salt 클라이언트를 설치할 수 있습니다.

지원되는 클라이언트 및 기능의 전체 목록은 [Client-configuration > Supported-features](#)를 참조하십시오.

이 안내서에서는 다양한 클라이언트를 수동 및 자동으로 등록하고 구성하는 방법에 대해 설명합니다.

Chapter 1. 지원되는 클라이언트 및 기능

Uyuni는 다양한 클라이언트 기술과 호환됩니다. 다양한 하드웨어 옵션으로 SUSE Linux Enterprise 또는 다른 Linux 운영 체제를 실행하는 기존 또는 Salt 클라이언트를 설치할 수 있습니다.

이 섹션에는 지원되는 클라이언트 시스템에 관한 요약이 포함되어 있습니다. 각 클라이언트에서 사용할 수 있는 기능의 상세 목록은 다음 페이지를 참조하십시오.

1.1. 지원되는 클라이언트 시스템

다음 테이블에는 기존 및 Salt 클라이언트에 지원되는 운영 체제가 나열됩니다.

이 표의 아이콘에는 다음과 같은 의미가 있습니다.

- ✓ 이 운영 체제를 실행하는 클라이언트를 SUSE에서 지원합니다.
- ✗ 이 운영 체제를 실행하는 클라이언트를 SUSE에서 지원하지 않습니다.
- ? 이 클라이언트는 현재 지원 고려 중이며, 이후에 지원될 수도 있고 지원되지 않을 수도 있습니다.



클라이언트 운영 체제 버전 및 SP 수준에는 Uyuni로 지원될 일반 지원(일반 또는 LTSS)이 적용되어야 합니다. 지원되는 제품 버전에 대한 상세 정보는 <https://www.suse.com/lifecycle>에서 참조하십시오.



클라이언트에서 실행되는 운영 체제는 운영 체제를 제공하는 조직이 지원합니다.

표 1. 지원되는 클라이언트 시스템

Operating System	Architecture	Traditional Clients	Salt Clients
SUSE Linux Enterprise 15	x86-64, ppc64le, IBM Z, ARM	✓	✓
SUSE Linux Enterprise 12	x86-64, ppc64le, IBM Z, ARM	✓	✓
SUSE Linux Enterprise Server for SAP 15	x86-64, ppc64le	✓	✓
SUSE Linux Enterprise Server for SAP 12	x86-64, ppc64le	✓	✓
SLE Micro	x86-64, ppc64le, aarch64	✗	✓
openSUSE Leap 15	x86-64, aarch64	✓	✓
Alibaba Cloud Linux 2	x86-64, aarch64	✗	✓
AlmaLinux 9	x86-64, ppc64le, IBM Z, aarch64	✗	✓
AlmaLinux 8	x86-64, aarch64	✗	✓
Amazon Linux 2	x86-64, aarch64	✗	✓

Operating System	Architecture	Traditional Clients	Salt Clients
CentOS 7	x86-64, ppc64le, aarch64	✓	✓
Debian 11	x86-64	✗	✓
Debian 10	x86-64	✗	✓
Oracle Linux 9	x86-64, aarch64	✗	✓
Oracle Linux 8	x86-64, aarch64	✗	✓
Oracle Linux 7	x86-64, aarch64	✓	✓
Red Hat Enterprise Linux 9	x86-64	✗	✓
Red Hat Enterprise Linux 8	x86-64	✗	✓
Red Hat Enterprise Linux 7	x86-64	✓	✓
Rocky Linux 9	x86-64, aarch64, ppc64le, s390x	✗	✓
Rocky Linux 8	x86-64, aarch64	✗	✓
Ubuntu 22.04	amd64	✗	✓
Ubuntu 20.04	amd64	✗	✓
Ubuntu 18.04	amd64	✗	✓



Debian and Ubuntu list the x86-64 architecture as amd64.

배포의 수명이 만료하면 3개월의 유예 기간이 시작되며 지원은 중단한 것으로 간주됩니다. 이 기간이 경과하면 해당 제품은 지원하지 않는 것으로 간주됩니다. 모든 지원은 최선의 노력으로 제공합니다.

수명 종료에 대한 자세한 설명은 <https://endoflife.software/operating-systems>를 참조하십시오.

1.2. 지원되는 도구 패키지

The `spacewalk-utils` and `spacewalk-utils-extras` packages can provide additional services and features.

표 2. Spacewalk 유틸리티

도구 이름	설명	지원 여부
<code>spacewalk-common-channels</code>	SUSE Customer Center 가 제공하지 않는 채널 추가	✓
<code>spacewalk-hostname-rename</code>	Uyuni의 호스트 이름 변경	✓
<code>spacewalk-clone-by-date</code>	특정 날짜까지 채널 복제	✓
<code>spacewalk-sync-setup</code>	ISS 마스터 및 슬레이브 조직 매핑 설정	✓

도구 이름	설명	지원 여부
spacewalk-manage-channel-lifecycle	채널 라이프사이클 관리	✓

1.3. 지원하는 SUSE 및 openSUSE 클라이언트 기능

아래 표는 SUSE 및 openSUSE 클라이언트에서 다양한 기능의 사용 가능 여부를 정리한 것입니다. 이 표에서는 SAP용 SLES, SLED, SUSE Linux Enterprise Server 및 HPC용 SUSE Linux Enterprise Server를 포함한 SUSE Linux Enterprise 운영 체제의 모든 변형에 대해 설명합니다.



클라이언트에서 실행하는 운영 체제는 운영 체제를 제공하는 조직이 지원합니다. SUSE Linux Enterprise는 SUSE가 지원합니다. openSUSE는 SUSE 커뮤니티가 지원합니다.

이 표의 아이콘에는 다음과 같은 의미가 있습니다.

- ✓ 이 기능을 Salt 및 기존 클라이언트 모두에서 사용할 수 있습니다.
- ✗ 이 기능을 사용할 수 없습니다.
- ? 이 기능은 현재 고려 중이며, 이후에 사용할 수도 있고 사용하지 못할 수도 있습니다.
- Traditional 이 기능은 기존 클라이언트에서만 지원합니다.
- Salt 이 기능은 Salt 클라이언트에서만 지원합니다.

표 3. SUSE 및 openSUSE 운영 체제에서 지원하는 기능

Feature	SUSE Linux Enterprise 12	SUSE Linux Enterprise 15	openSUSE 15
Client	✓	✓	✓
System packages	SUSE	SUSE	openSUSE Community
Registration	✓	✓	Salt
Install packages	✓	✓	Salt
Apply patches	✓	✓	Salt
Remote commands	✓	✓	Salt
System package states	Salt	Salt	Salt
System custom states	Salt	Salt	Salt
Group custom states	Salt	Salt	Salt
Organization custom states	Salt	Salt	Salt
System set manager (SSM)	✓	✓	Salt
Product migration	✓	✓	Salt
Basic Virtual Guest Management *	✓	✓	Salt

Feature	SUSE Linux Enterprise 12	SUSE Linux Enterprise 15	openSUSE 15
Advanced Virtual Guest Management *	Salt	Salt	Salt
Virtual Guest Installation (AutoYaST), as Host OS	Traditional	Traditional	✗
Virtual Guest Installation (image template), as Host OS	Salt	Salt	Salt
Virtual Guest Management	Salt	Salt	Salt
System deployment (PXE/AutoYaST)	✓	✓	✓
System redeployment (AutoYaST)	✓	✓	Salt
Contact methods	Traditional: OSAD, RHNSD, SSH-push. Salt: ZeroMQ, Salt-SSH	Traditional: OSAD, RHNSD, SSH-push. Salt: ZeroMQ, Salt-SSH	Salt: ZeroMQ, Salt-SSH
Works with Uyuni Proxy	✓	✓	Salt
Action chains	✓	✓	Salt
Staging (pre-download of packages)	✓	✓	Salt
Duplicate package reporting	✓	✓	Salt
CVE auditing	✓	✓	Salt
SCAP auditing	✓	✓	Salt
Package verification	Traditional	Traditional	✗
Package locking	Salt	Salt	Salt
System locking	Traditional	Traditional	✗
Maintenance Windows	✓	✓	✓
System snapshot	Traditional	Traditional	✗
Configuration file management	✓	✓	Salt
Package profiles	Traditional. Salt: Profiles supported, Sync not supported	Traditional. Salt: Profiles supported, Sync not supported	Salt: Profiles supported, Sync not supported
Power management	✓	✓	✓
Monitoring server	Salt	Salt	Salt
Monitored clients	Salt	Salt	Salt

Feature	SUSE Linux Enterprise 12	SUSE Linux Enterprise 15	openSUSE 15
Docker buildhost	Salt	Salt	?
Build Docker image with OS	Salt	Salt	Salt
Kiwi buildhost	Salt	?	?
Build Kiwi image with OS	Salt	?	✗
Recurring Actions	Salt	Salt	Salt
AppStreams	N/A	N/A	N/A
Yomi	✗	✓	✓

* 가상 게스트 관리:

이 표에서 가상 게스트 관리는 기본과 고급으로 구분되어 있습니다.

기본 가상 게스트 관리에는 VM, 느린 새로 고침, VM 라이프사이클 작업(시작, 중지, 다시 시작, 일시 중지) 나열과 VM vCPU 및 메모리 수정이 포함됩니다.

고급 가상 게스트 관리에는 기본 가상 게스트 관리의 모든 기능과 함께 빠른 새로 고침, VM 라이프사이클 작업(삭제, 재설정, 전원 끄기), VM 디스크, 네트워크, 그래픽 표시 및 그래픽 표시 구성 수정이 포함됩니다.

1.4. 지원하는 SLE Micro 및 openSUSE MicroOS 클라이언트 기능



SLE Micro 및 openSUSE MicroOS 클라이언트에 대한 지원은 테스트 목적을 위해 기술 미리보기로 제공되며 이 단계에서는 모든 기능이 완전히 작동하지 않습니다. 이 기능은 Uyuni 이후 버전에서 완전하게 지원될 예정입니다. 프로덕션 시스템에서 이 기능을 사용하지 마십시오.



클라이언트에서 실행하는 운영 체제는 운영 체제를 제공하는 조직이 지원합니다. SUSE Linux Enterprise는 SUSE가 지원합니다. openSUSE MicroOS는 SUSE 커뮤니티가 지원합니다.

이 표의 아이콘에는 다음과 같은 의미가 있습니다.

- ✓ 이 기능을 Salt 및 기존 클라이언트 모두에서 사용할 수 있습니다.
- ✗ 이 기능을 사용할 수 없습니다.
- ? 이 기능은 현재 고려 중이며, 이후에 사용할 수도 있고 사용하지 못할 수도 있습니다.
- Traditional 이 기능은 기존 클라이언트에서만 지원합니다.
- Salt 이 기능은 Salt 클라이언트에서만 지원합니다.

표 4. SLE Micro 및 openSUSE MicroOS 운영 체제에서 지원하는 기능

Feature	SLE Micro	openSUSE MicroOS
Client	Salt	Salt
Operating system packages	Salt	Salt
Registration	Salt	Salt
Install packages	Salt	Salt
Apply patches (requires CVE ID)	Salt	Salt
Remote commands	Salt	Salt
System package states	Salt	Salt
System custom states	Salt	Salt
Group custom states	Salt	Salt
Organization custom states	Salt	Salt
System set manager (SSM)	Salt	Salt
Product migration	?	Salt
Basic Virtual Guest Management *	?	Salt
Advanced Virtual Guest Management *	?	Salt
Virtual Guest Installation (Kickstart), as Host OS	✗	Salt
Virtual Guest Installation (image template), as Host OS	?	Salt
System deployment (PXE/Kickstart)	?	Salt
System redeployment (Kickstart)	✗	Salt
Contact methods	Salt: ZeroMQ	Salt: ZeroMQ
Works with Uyuni Proxy	Salt	Salt
Action chains	?	?
Staging (pre-download of packages)	?	?
Duplicate package reporting	Salt	Salt
CVE auditing (requires CVE ID)	Salt	Salt
SCAP auditing	?	?
Package verification	?	?
Package locking	Salt	Salt
System locking	?	?
Maintenance Windows	?	?
System snapshot	✗	✗

Feature	SLE Micro	openSUSE MicroOS
Configuration file management	Salt	Salt
Snapshots and profiles	Salt: Profiles supported, Sync not supported	Salt: Profiles supported, Sync not supported
Power management	Salt	Salt
Monitoring server	✗	✗
Monitored clients	Salt	Salt
Docker buildhost	✗	✗
Build Docker image with OS	✗	✗
Kiwi buildhost	✗	✗
Build Kiwi image with OS	Salt	Salt
Recurring Actions	Salt	Salt
AppStreams	N/A	N/A
Yomi	?	?

* 가상 게스트 관리:

이 표에서 가상 게스트 관리는 기본과 고급으로 구분되어 있습니다.

기본 가상 게스트 관리에는 VM, 느린 새로 고침, VM 라이프싸이클 작업(시작, 중지, 다시 시작, 일시 중지) 나열과 VM vCPU 및 메모리 수정이 포함됩니다.

고급 가상 게스트 관리에는 기본 가상 게스트 관리의 모든 기능과 함께 빠른 새로 고침, VM 라이프싸이클 작업(삭제, 재설정, 전원 끄기), VM 디스크, 네트워크, 그래픽 표시 및 그래픽 표시 구성 수정이 포함됩니다.

1.5. 지원하는 Alibaba Cloud Linux 기능

아래 표는 Alibaba Cloud Linux 클라이언트에서 다양한 기능의 사용 가능 여부를 정리한 것입니다.



클라이언트에서 실행하는 운영 체제는 운영 체제를 제공하는 조직이 지원합니다. Alibaba Cloud Linux는 Alibaba Cloud에서 지원됩니다.

이 표의 아이콘에는 다음과 같은 의미가 있습니다.

- ✓ 이 기능을 Salt 및 기존 클라이언트 모두에서 사용할 수 있습니다.
- ✗ 이 기능을 사용할 수 없습니다.
- ? 이 기능은 현재 고려 중이며, 이후에 사용할 수도 있고 사용하지 못할 수도 있습니다.
- Traditional 이 기능은 기존 클라이언트에서만 지원합니다.
- Salt 이 기능은 Salt 클라이언트에서만 지원합니다.

표 5. Alibaba Cloud Linux 운영 체제에서 지원하는 기능

기능	Alibaba Cloud Linux 2
클라이언트	Salt
운영 체제 패키지	Salt
등록	Salt
패키지 설치	Salt
패치 적용(CVE ID 필요)	Salt
원격 명령	Salt
시스템 패키지 상태	Salt
시스템 사용자 정의 상태	Salt
그룹 사용자 정의 상태	Salt
조직 사용자 정의 상태	Salt
시스템 설정 관리자(SSM)	Salt
제품 마이그레이션	N/A
기본 가상 게스트 관리 *	?
고급 가상 게스트 관리 *	?
호스트 OS로 가상 게스트 설치(킥스타트)	✗
호스트 OS로 가상 게스트 설치(이미지 템플릿)	?
시스템 배포(PXE/킥스타트)	?
시스템 재배포(킥스타트)	?
연락 방법	Salt: ZeroMQ, Salt-SSH
Uyuni Proxy와 함께 작동	Salt
작업 체인	Salt
스테이징(패키지 사전 다운로드)	Salt
중복 패키지 보고	Salt
CVE 감사(CVE ID 필요)	Salt
SCAP 감사	Salt
패키지 검증	✗
패키지 잠금	✗
시스템 잠금	✗
유지보수 기간	✓
시스템 스냅샷	✗
구성 파일 관리	Salt
스냅샷 및 프로필	Salt: 프로필 지원됨, 동기화 지원되지 않음
전원 관리	?
모니터링 서버	✗

기능	Alibaba Cloud Linux 2
모니터링 클라이언트	Salt
Docker 빌드호스트	Salt
OS를 사용한 Docker 이미지 빌드	Salt
Kiwi 빌드호스트	Salt
OS를 사용한 Kiwi 이미지 빌드	Salt
반복 작업	Salt
AppStreams	N/A
Yomi	N/A

* 가상 게스트 관리:

이 표에서 가상 게스트 관리는 기본과 고급으로 구분되어 있습니다.

기본 가상 게스트 관리에는 VM, 느린 새로 고침, VM 라이프사이클 작업(시작, 중지, 다시 시작, 일시 중지) 나열과 VM vCPU 및 메모리 수정이 포함됩니다.

고급 가상 게스트 관리에는 기본 가상 게스트 관리의 모든 기능과 함께 빠른 새로 고침, VM 라이프사이클 작업(삭제, 재설정, 전원 끄기), VM 디스크, 네트워크, 그래픽 표시 및 그래픽 표시 구성 수정이 포함됩니다.

* 기존 스택은 Alibaba Cloud Linux에서 사용할 수 있지만 지원되지는 않습니다.

1.6. 지원하는 AlmaLinux 기능

아래 표는 AlmaLinux 클라이언트에서 다양한 기능의 사용 가능 여부를 정리한 것입니다.



클라이언트에서 실행하는 운영 체제는 운영 체제를 제공하는 조직이 지원합니다.
AlmaLinux는 AlmaLinux 커뮤니티에서 지원됩니다.

이 표의 아이콘에는 다음과 같은 의미가 있습니다.

- ✓ 이 기능을 Salt 및 기존 클라이언트 모두에서 사용할 수 있습니다.
- ✗ 이 기능을 사용할 수 없습니다.
- ? 이 기능은 현재 고려 중이며, 이후에 사용할 수도 있고 사용하지 못할 수도 있습니다.
- Traditional 이 기능은 기존 클라이언트에서만 지원합니다.
- Salt 이 기능은 Salt 클라이언트에서만 지원합니다.

표 6. AlmaLinux 운영 체제에서 지원하는 기능

기능	AlmaLinux 9	AlmaLinux 8
클라이언트	Salt (plain AlmaLinux)	Salt (plain AlmaLinux)
시스템 패키지	AlmaLinux Community	AlmaLinux Community

기능	AlmaLinux 9	AlmaLinux 8
등록	Salt	Salt
패키지 설치	Salt	Salt
패치 적용	Salt	Salt
원격 명령	Salt	Salt
시스템 패키지 상태	Salt	Salt
시스템 사용자 정의 상태	Salt	Salt
그룹 사용자 정의 상태	Salt	Salt
조직 사용자 정의 상태	Salt	Salt
시스템 설정 관리자(SSM)	Salt	Salt
제품 마이그레이션	N/A	N/A
기본 가상 게스트 관리 *	Salt	Salt
고급 가상 게스트 관리 *	Salt	Salt
호스트 OS로 가상 게스트 설치(킥스타트)	✗	✗
호스트 OS로 가상 게스트 설치(이미지 템플릿)	Salt	Salt
시스템 배포(PXE/킥스타트)	Salt	Salt
시스템 재배포(킥스타트)	Salt	Salt
연락 방법	Salt: ZeroMQ, Salt-SSH	Salt: ZeroMQ, Salt-SSH
Uyuni Proxy와 함께 작동	Salt	Salt
작업 체인	Salt	Salt
스테이징(패키지 사전 다운로드)	Salt	Salt
중복 패키지 보고	Salt	CVE 감사
Salt	Salt	SCAP 감사
Salt	Salt	패키지 검증
✗	✗	패키지 잠금
✗	✗	시스템 잠금
✗	✗	유지보수 기간
✓	✓	시스템 스냅샷
✗	✗	구성 파일 관리
Salt	Salt	스냅샷 및 프로필
Salt: 프로필 지원됨, 동기화 지원되지 않음	Salt: 프로필 지원됨, 동기화 지원되지 않음	전원 관리
Salt	Salt	모니터링 서버
✗	✗	모니터링되는 클라이언트

기능	AlmaLinux 9	AlmaLinux 8
Salt	Salt	Docker 빌드호스트
✗	✗	OS를 사용한 Kiwi 이미지 빌드
✗	✗	Kiwi 빌드호스트
✗	✗	OS를 사용한 Kiwi 이미지 빌드
✗	✗	반복 작업
Salt	Salt	AppStreams
✓	✓	Yomi

* 가상 게스트 관리:

이 표에서 가상 게스트 관리는 기본과 고급으로 구분되어 있습니다.

기본 가상 게스트 관리에는 VM, 느린 새로 고침, VM 라이프사이클 작업(시작, 중지, 다시 시작, 일시 중지) 나열과 VM vCPU 및 메모리 수정이 포함됩니다.

고급 가상 게스트 관리에는 기본 가상 게스트 관리의 모든 기능과 함께 빠른 새로 고침, VM 라이프사이클 작업(삭제, 재설정, 전원 끄기), VM 디스크, 네트워크, 그래픽 표시 및 그래픽 표시 구성 수정이 포함됩니다.

1.7. 지원되는 Amazon Linux 기능

아래 표는 Amazon Linux 클라이언트에서 다양한 기능의 사용 가능 여부를 정리한 것입니다.



클라이언트에서 실행하는 운영 체제는 운영 체제를 제공하는 조직이 지원합니다. Amazon Linux은 Amazon에서 지원됩니다.

이 표의 아이콘에는 다음과 같은 의미가 있습니다.

- ✓ 이 기능을 Salt 및 기존 클라이언트 모두에서 사용할 수 있습니다.
- ✗ 이 기능을 사용할 수 없습니다.
- ? 이 기능은 현재 고려 중이며, 이후에 사용할 수도 있고 사용하지 못할 수도 있습니다.
- Traditional 이 기능은 기존 클라이언트에서만 지원합니다.
- Salt 이 기능은 Salt 클라이언트에서만 지원합니다.

표 7. Amazon Linux 운영 체제에서 지원하는 기능

기능	Amazon Linux 2
클라이언트	Salt
운영 체제 패키지	Salt
등록	Salt
패키지 설치	Salt
패치 적용(CVE ID 필요)	Salt

기능	Amazon Linux 2
원격 명령	Salt
시스템 패키지 상태	Salt
시스템 사용자 정의 상태	Salt
그룹 사용자 정의 상태	Salt
조직 사용자 정의 상태	Salt
시스템 설정 관리자(SSM)	Salt
제품 마이그레이션	N/A
기본 가상 게스트 관리 *	?
고급 가상 게스트 관리 *	?
호스트 OS로 가상 게스트 설치(킥스타트)	✗
호스트 OS로 가상 게스트 설치(이미지 템플릿)	?
시스템 배포(PXE/킥스타트)	?
시스템 재배포(킥스타트)	?
연락 방법	Salt: ZeroMQ, Salt-SSH
Uyuni Proxy와 함께 작동	Salt
작업 체인	Salt
스테이징(패키지 사전 다운로드)	Salt
중복 패키지 보고	Salt
CVE 감사(CVE ID 필요)	Salt
SCAP 감사	Salt
패키지 검증	✗
패키지 잠금	✗
시스템 잠금	✗
유지보수 기간	✓
시스템 스냅샷	✗
구성 파일 관리	Salt
스냅샷 및 프로필	Salt: 프로필 지원됨, 동기화 지원되지 않음
전원 관리	?
모니터링 서버	✗
모니터링되는 클라이언트	Salt
Docker 빌드호스트	Salt
OS를 사용한 Docker 이미지 빌드	Salt
Kiwi 빌드호스트	Salt
OS를 사용한 Kiwi 이미지 빌드	Salt

기능	Amazon Linux 2
반복 작업	Salt
AppStreams	N/A
Yomi	N/A

*** 가상 게스트 관리:**

이 표에서 가상 게스트 관리는 기본과 고급으로 구분되어 있습니다.

기본 가상 게스트 관리에는 VM, 느린 새로 고침, VM 라이프사이클 작업(시작, 종지, 다시 시작, 일시 종지) 나열과 VM vCPU 및 메모리 수정이 포함됩니다.

고급 가상 게스트 관리에는 기본 가상 게스트 관리의 모든 기능과 함께 빠른 새로 고침, VM 라이프사이클 작업(삭제, 재설정, 전원 끄기), VM 디스크, 네트워크, 그래픽 표시 및 그래픽 표시 구성 수정이 포함됩니다.

*** 기존 스택은 Amazon Linux에서 사용할 수 있지만 지원되지는 않습니다.**

1.8. 지원하는 CentOS 기능

아래 표는 CentOS 클라이언트에서 다양한 기능의 사용 가능 여부를 정리한 것입니다.



클라이언트에서 실행하는 운영 체제는 운영 체제를 제공하는 조직이 지원합니다. CentOS는 CentOS 커뮤니티가 지원합니다.

이 표의 아이콘에는 다음과 같은 의미가 있습니다.

- ✓ 이 기능을 Salt 및 기존 클라이언트 모두에서 사용할 수 있습니다.
- ✗ 이 기능을 사용할 수 없습니다.
- ? 이 기능은 현재 고려 중이며, 이후에 사용할 수도 있고 사용하지 못할 수도 있습니다.
- Traditional 이 기능은 기존 클라이언트에서만 지원합니다.
- Salt 이 기능은 Salt 클라이언트에서만 지원합니다.

표 8. CentOS 운영 체제에서 지원하는 기능

기능	CentOS 7
클라이언트	✓ (plain CentOS)
시스템 패키지	CentOS Community
등록	✓
패키지 설치	✓
패치 적용(CVE ID 필요)	✓(오류를 위해 타사 서비스 필요)
원격 명령	✓
시스템 패키지 상태	Salt

기능	CentOS 7
시스템 사용자 정의 상태	Salt
그룹 사용자 정의 상태	Salt
조직 사용자 정의 상태	Salt
시스템 설정 관리자(SSM)	✓
제품 마이그레이션	N/A
기본 가상 게스트 관리 *	✓
고급 가상 게스트 관리 *	Salt
호스트 OS로 가상 게스트 설치(킥스타트)	Traditional
호스트 OS로 가상 게스트 설치(이미지 템플릿)	✓
시스템 배포(PXE/킥스타트)	✓
시스템 재배포(킥스타트)	✓
연락 방법	Traditional: OSAD, RHNSD, SSH-push. Salt: ZeroMQ, Salt-SSH
Uyuni Proxy와 함께 작동	✓
작업 체인	✓
스테이징(패키지 사전 다운로드)	✓
중복 패키지 보고	✓
CVE 감사(CVE ID 필요)	✓
SCAP 감사	✓
패키지 검증	Traditional
패키지 잠금	✓
시스템 잠금	Traditional
유지보수 기간	✓
시스템 스냅샷	Traditional
구성 파일 관리	✓
스냅샷 및 프로필	Traditional. Salt: 프로필 지원됨, 동기화 지원되지 않음
전원 관리	✓
모니터링 서버	✗
모니터링되는 클라이언트	Salt
Docker 빌드호스트	✗
OS를 사용한 Docker 이미지 빌드	✗
Kiwi 빌드호스트	✗
OS를 사용한 Kiwi 이미지 빌드	✗
반복 작업	Salt

기능	CentOS 7
AppStreams	N/A
Yomi	N/A

* 가상 게스트 관리:

이 표에서 가상 게스트 관리는 기본과 고급으로 구분되어 있습니다.

기본 가상 게스트 관리에는 VM, 느린 새로 고침, VM 라이프싸이클 작업(시작, 중지, 다시 시작, 일시 중지) 나열과 VM vCPU 및 메모리 수정이 포함됩니다.

고급 가상 게스트 관리에는 기본 가상 게스트 관리의 모든 기능과 함께 빠른 새로 고침, VM 라이프싸이클 작업(삭제, 재설정, 전원 끄기), VM 디스크, 네트워크, 그래픽 표시 및 그래픽 표시 구성 수정이 포함됩니다.

1.9. 지원하는 Debian 기능

아래 표는 Debian 클라이언트에서 다양한 기능의 사용 가능 여부를 정리한 것입니다.



클라이언트에서 실행하는 운영 체제는 운영 체제를 제공하는 조직이 지원합니다. Debian은 Debian 커뮤니티가 지원합니다.

이 표의 아이콘에는 다음과 같은 의미가 있습니다.

- ✓ 이 기능을 Salt 및 기존 클라이언트 모두에서 사용할 수 있습니다.
- ✗ 이 기능을 사용할 수 없습니다.
- ? 이 기능은 현재 고려 중이며, 이후에 사용할 수도 있고 사용하지 못할 수도 있습니다.
- Traditional 이 기능은 기존 클라이언트에서만 지원합니다.
- Salt 이 기능은 Salt 클라이언트에서만 지원합니다.

표 9. Debian 운영 체제에서 지원하는 기능

기능	Debian 10	Debian 11
클라이언트	✓	✓
시스템 패키지	Debian Community	Debian Community
등록	Salt	Salt
패키지 설치	Salt	Salt
패치 적용	?	?
원격 명령	Salt	Salt
시스템 패키지 상태	Salt	Salt
시스템 사용자 정의 상태	Salt	Salt
그룹 사용자 정의 상태	Salt	Salt
조직 사용자 정의 상태	Salt	Salt

기능	Debian 10	Debian 11
시스템 설정 관리자(SSM)	Salt	Salt
제품 마이그레이션	N/A	N/A
기본 가상 게스트 관리 *	Salt	Salt
고급 가상 게스트 관리 *	Salt	Salt
호스트 OS로 가상 게스트 설치(킥스타트)	✗	✗
호스트 OS로 가상 게스트 설치(이미지 템플릿)	Salt	Salt
시스템 배포(PXE/킥스타트)	✗	✗
시스템 재배포(킥스타트)	✗	✗
연락 방법	Salt: ZeroMQ, Salt-SSH	Salt: ZeroMQ, Salt-SSH
Uyuni Proxy와 함께 작동	Salt	Salt
작업 체인	Salt	Salt
스테이징(패키지 사전 다운로드)	Salt	Salt
중복 패키지 보고	Salt	Salt
CVE 감사	?	?
SCAP 감사	?	?
패키지 검증	✗	✗
패키지 잠금	✓	✓
시스템 잠금	✗	✗
유지보수 기간	✓	✓
시스템 스냅샷	✗	✗
구성 파일 관리	Salt	Salt
패키지 프로필	Salt: 프로필 지원됨, 동기화 지원되지 않음	Salt: 프로필 지원됨, 동기화 지원되지 않음
전원 관리	✓	✓
모니터링 서버	✗	✗
모니터링되는 클라이언트	Salt	Salt
Docker 빌드호스트	?	?
OS를 사용한 Docker 이미지 빌드	Salt	Salt
Kiwi 빌드호스트	✗	✗
OS를 사용한 Kiwi 이미지 빌드	✗	✗
반복 작업	Salt	Salt
AppStreams	N/A	N/A
Yomi	N/A	N/A

* 가상 게스트 관리:

이 표에서 가상 게스트 관리는 기본과 고급으로 구분되어 있습니다.

기본 가상 게스트 관리에는 VM, 느린 새로 고침, VM 라이프싸이클 작업(시작, 중지, 다시 시작, 일시 중지) 나열과 VM vCPU 및 메모리 수정이 포함됩니다.

고급 가상 게스트 관리에는 기본 가상 게스트 관리의 모든 기능과 함께 빠른 새로 고침, VM 라이프싸이클 작업(삭제, 재설정, 전원 끄기), VM 디스크, 네트워크, 그래픽 표시 및 그래픽 표시 구성 수정이 포함됩니다.

1.10. 지원하는 Oracle 기능

아래 표는 Oracle Linux 클라이언트에서 다양한 기능의 사용 가능 여부를 정리한 것입니다.



클라이언트에서 실행하는 운영 체제는 운영 체제를 제공하는 조직이 지원합니다. Oracle Linux는 Oracle이 지원합니다.

이 표의 아이콘에는 다음과 같은 의미가 있습니다.

- ✓ 이 기능을 Salt 및 기존 클라이언트 모두에서 사용할 수 있습니다.
- ✗ 이 기능을 사용할 수 없습니다.
- ? 이 기능은 현재 고려 중이며, 이후에 사용할 수도 있고 사용하지 못할 수도 있습니다.
- Traditional 이 기능은 기존 클라이언트에서만 지원합니다.
- Salt 이 기능은 Salt 클라이언트에서만 지원합니다.

표 10. Oracle Linux 운영 체제에서 지원하는 기능

기능	Oracle Linux 7	Oracle Linux 8	Oracle Linux 9
클라이언트	✓	Salt	Salt
운영 체제 패키지	✓	Salt	Salt
등록	✓	Salt	Salt
패키지 설치	✓	Salt	Salt
패치 적용(CVE ID 필요)	✓	Salt	Salt
원격 명령	✓	Salt	Salt
시스템 패키지 상태	Salt	Salt	Salt
시스템 사용자 정의 상태	Salt	Salt	Salt
그룹 사용자 정의 상태	Salt	Salt	Salt
조직 사용자 정의 상태	Salt	Salt	Salt
시스템 설정 관리자(SSM)	✓	Salt	Salt
제품 마이그레이션	N/A	N/A	N/A
기본 가상 게스트 관리 *	✓	Salt	Salt

기능	Oracle Linux 7	Oracle Linux 8	Oracle Linux 9
고급 가상 게스트 관리 *	Salt	Salt	Salt
호스트 OS로 가상 게스트 설치(킥스타트)	Traditional	✗	✗
호스트 OS로 가상 게스트 설치(이미지 템플릿)	✓	Salt	Salt
시스템 배포(PXE/킥스타트)	✓	Salt	Salt
시스템 재배포(킥스타트)	✓	Salt	Salt
연락 방법	Traditional: OSAD, RHNSD, SSH-push. Salt: ZeroMQ, Salt-SSH	Salt: ZeroMQ, Salt-SSH	Salt: ZeroMQ, Salt-SSH
Uyuni Proxy와 함께 작동	✓	Salt	Salt
작업 체인	✓	Salt	Salt
스테이징(패키지 사전 다운로드)	✓	Salt	Salt
중복 패키지 보고	✓	Salt	Salt
CVE 감사(CVE ID 필요)	✓	Salt	Salt
SCAP 감사	✓	Salt	Salt
패키지 검증	Traditional	✗	✗
패키지 잠금	✓	?	?
시스템 잠금	Traditional	✗	✗
유지보수 기간	✓	✓	✓
시스템 스냅샷	Traditional	✗	✗
구성 파일 관리	✓	Salt	Salt
스냅샷 및 프로필	Traditional. Salt: 프로필 지원됨, 동기화 지원되지 않음	Salt: 프로필 지원됨, 동기화 지원되지 않음	Salt: 프로필 지원됨, 동기화 지원되지 않음
전원 관리	✓	Salt	Salt
모니터링 서버	✗	✗	✗
모니터링되는 클라이언트	Salt	Salt	Salt
Docker 빌드호스트	✗	✗	✗
OS를 사용한 Docker 이미지 빌드	✗	✗	✗
Kiwi 빌드호스트	✗	✗	✗
OS를 사용한 Kiwi 이미지 빌드	✗	✗	✗
반복 작업	Salt	Salt	Salt

기능	Oracle Linux 7	Oracle Linux 8	Oracle Linux 9
AppStreams	N/A	✓	✓
Yomi	N/A	N/A	N/A

* 가상 게스트 관리:

이 표에서 가상 게스트 관리는 기본과 고급으로 구분되어 있습니다.

기본 가상 게스트 관리에는 VM, 느린 새로 고침, VM 라이프싸이클 작업(시작, 중지, 다시 시작, 일시 중지) 나열과 VM vCPU 및 메모리 수정이 포함됩니다.

고급 가상 게스트 관리에는 기본 가상 게스트 관리의 모든 기능과 함께 빠른 새로 고침, VM 라이프싸이클 작업(삭제, 재설정, 전원 끄기), VM 디스크, 네트워크, 그래픽 표시 및 그래픽 표시 구성 수정이 포함됩니다.

1.11. 지원하는 Red Hat Enterprise Linux 기능

This table lists the availability of various features on native Red Hat Enterprise Linux clients.



클라이언트에서 실행하는 운영 체제는 운영 체제를 제공하는 조직이 지원합니다. Red Hat Enterprise Linux는 Red Hat이 지원합니다.

이 표의 아이콘에는 다음과 같은 의미가 있습니다.

- ✓ 이 기능을 Salt 및 기존 클라이언트 모두에서 사용할 수 있습니다.
- ✗ 이 기능을 사용할 수 없습니다.
- ? 이 기능은 현재 고려 중이며, 이후에 사용할 수도 있고 사용하지 못할 수도 있습니다.
- Traditional 이 기능은 기존 클라이언트에서만 지원합니다.
- Salt 이 기능은 Salt 클라이언트에서만 지원합니다.

표 11. Red Hat Enterprise Linux 운영 체제에서 지원하는 기능

기능	RHEL 7	RHEL 8	RHEL 9
클라이언트	✓	Salt	Salt
시스템 패키지	Red Hat	Red Hat	Red Hat
등록	✓	Salt	Salt
패키지 설치	✓	Salt	Salt
패치 적용	✓	Salt	Salt
원격 명령	✓	Salt	Salt
시스템 패키지 상태	Salt	Salt	Salt
시스템 사용자 정의 상태	Salt	Salt	Salt
그룹 사용자 정의 상태	Salt	Salt	Salt

기능	RHEL 7	RHEL 8	RHEL 9
조직 사용자 정의 상태	Salt	Salt	Salt
시스템 설정 관리자(SSM)	Salt	Salt	Salt
제품 마이그레이션	N/A	N/A	N/A
기본 가상 게스트 관리 *	✓	Salt	Salt
고급 가상 게스트 관리 *	Salt	Salt	Salt
호스트 OS로 가상 게스트 설치(킥스타트)	Traditional	✗	✗
호스트 OS로 가상 게스트 설치(이미지 템플릿)	✓	Salt	Salt
시스템 배포(PXE/킥스타트)	✓	Salt	Salt
시스템 재배포(킥스타트)	✓	Salt	Salt
연락 방법	Traditional: OSAD, RHNSD, SSH-push. Salt: ZeroMQ, Salt-SSH	Salt: ZeroMQ, Salt-SSH	Salt: ZeroMQ, Salt-SSH
Uyuni Proxy와 함께 작동	✓	Salt	Salt
작업 체인	✓	Salt	Salt
스테이징(패키지 사전 다운로드)	✓	Salt	Salt
중복 패키지 보고	✓	Salt	Salt
CVE 감사	✓	Salt	Salt
SCAP 감사	✓	Salt	Salt
패키지 검증	Traditional	✗	✗
패키지 잠금	✓	?	?
시스템 잠금	Traditional	✗	✗
유지보수 기간	✓	✓	✓
시스템 스냅샷	Traditional	✗	✗
구성 파일 관리	✓	Salt	Salt
스냅샷 및 프로필	Traditional. Salt: 프로필 지원됨, 동기화 지원되지 않음	Salt: 프로필 지원됨, 동기화 지원되지 않음	Salt: 프로필 지원됨, 동기화 지원되지 않음
전원 관리	✓	Salt	Salt
모니터링 서버	✗	✗	✗
모니터링되는 클라이언트	Salt	Salt	Salt
Docker 빌드호스트	✗	✗	✗
OS를 사용한 Docker 이미지 빌드	?	?	?

기능	RHEL 7	RHEL 8	RHEL 9
Kiwi 빌드호스트	✗	✗	✗
OS를 사용한 Kiwi 이미지 빌드	✗	✗	✗
반복 작업	Salt	Salt	Salt
AppStreams	N/A	✓	✓
Yomi	N/A	N/A	N/A

* 가상 게스트 관리:

이 표에서 가상 게스트 관리는 기본과 고급으로 구분되어 있습니다.

기본 가상 게스트 관리에는 VM, 느린 새로 고침, VM 라이프싸이클 작업(시작, 중지, 다시 시작, 일시 중지) 나열과 VM vCPU 및 메모리 설정이 포함됩니다.

고급 가상 게스트 관리에는 기본 가상 게스트 관리의 모든 기능과 함께 빠른 새로 고침, VM 라이프싸이클 작업(삭제, 재설정, 전원 끄기), VM 디스크, 네트워크, 그래픽 표시 및 그래픽 표시 구성 설정이 포함됩니다.

1.12. 지원하는 Rocky Linux 기능

아래 표는 Rocky Linux 클라이언트에서 다양한 기능의 사용 가능 여부를 정리한 것입니다.



클라이언트에서 실행하는 운영 체제는 운영 체제를 제공하는 조직이 지원합니다. Rocky Linux는 Rocky Linux 커뮤니티가 지원합니다.

이 표의 아이콘에는 다음과 같은 의미가 있습니다.

- ✓ 이 기능을 Salt 및 기존 클라이언트 모두에서 사용할 수 있습니다.
- ✗ 이 기능을 사용할 수 없습니다.
- ? 이 기능은 현재 고려 중이며, 이후에 사용할 수도 있고 사용하지 못할 수도 있습니다.
- Traditional 이 기능은 기존 클라이언트에서만 지원합니다.
- Salt 이 기능은 Salt 클라이언트에서만 지원합니다.

표 12. Rocky Linux 운영 체제에서 지원하는 기능

기능	Rocky Linux 8	Rocky Linux 9
클라이언트	Salt (plain Rocky Linux)	Salt (plain Rocky Linux)
시스템 패키지	Rocky Linux Community	Rocky Linux Community
등록	Salt	Salt
패키지 설치	Salt	Salt
패치 적용	Salt	Salt
원격 명령	Salt	Salt

기능	Rocky Linux 8	Rocky Linux 9
시스템 패키지 상태	Salt	Salt
시스템 사용자 정의 상태	Salt	Salt
그룹 사용자 정의 상태	Salt	Salt
조직 사용자 정의 상태	Salt	Salt
시스템 설정 관리자(SSM)	Salt	Salt
제품 마이그레이션	N/A	N/A
기본 가상 게스트 관리 *	Salt	Salt
고급 가상 게스트 관리 *	Salt	Salt
호스트 OS로 가상 게스트 설치(킥스타트)	✗	✗
호스트 OS로 가상 게스트 설치(이미지 템플릿)	Salt	Salt
시스템 배포(PXE/킥스타트)	Salt	Salt
시스템 재배포(킥스타트)	Salt	Salt
연락 방법	Salt: ZeroMQ, Salt-SSH	Salt: ZeroMQ, Salt-SSH
Uyuni Proxy와 함께 작동	Salt	Salt
작업 체인	Salt	Salt
스테이징(패키지 사전 다운로드)	Salt	Salt
중복 패키지 보고	Salt	Salt
CVE 감사	Salt	Salt
SCAP 감사	Salt	Salt
패키지 검증	✗	✗
패키지 잠금	?	?
시스템 잠금	✗	✗
유지보수 기간	✓	✓
시스템 스냅샷	✗	✗
구성 파일 관리	Salt	Salt
스냅샷 및 프로필	Salt: 프로필 지원됨, 동기화 지원되지 않음	Salt: 프로필 지원됨, 동기화 지원되지 않음
전원 관리	Salt	Salt
모니터링 서버	✗	✗
모니터링되는 클라이언트	Salt	Salt
Docker 빌드호스트	✗	✗
OS를 사용한 Docker 이미지 빌드	✗	✗
Kiwi 빌드호스트	✗	✗

기능	Rocky Linux 8	Rocky Linux 9
OS를 사용한 Kiwi 이미지 빌드	✗	✗
반복 작업	Salt	Salt
AppStreams	✓	✓
Yomi	N/A	N/A

* 가상 게스트 관리:

이 표에서 가상 게스트 관리는 기본과 고급으로 구분되어 있습니다.

기본 가상 게스트 관리에는 VM, 느린 새로 고침, VM 라이프싸이클 작업(시작, 중지, 다시 시작, 일시 중지) 나열과 VM vCPU 및 메모리 수정이 포함됩니다.

고급 가상 게스트 관리에는 기본 가상 게스트 관리의 모든 기능과 함께 빠른 새로 고침, VM 라이프싸이클 작업(삭제, 재설정, 전원 끄기), VM 디스크, 네트워크, 그래픽 표시 및 그래픽 표시 구성 수정이 포함됩니다.

1.13. 지원하는 Ubuntu 기능

아래 표는 Ubuntu 클라이언트에서 다양한 기능의 사용 가능 여부를 정리한 것입니다.



클라이언트에서 실행하는 운영 체제는 운영 체제를 제공하는 조직이 지원합니다. Ubuntu는 Canonical이 지원합니다.

이 표의 아이콘에는 다음과 같은 의미가 있습니다.

- ✓ 이 기능을 Salt 및 기존 클라이언트 모두에서 사용할 수 있습니다.
- ✗ 이 기능을 사용할 수 없습니다.
- ? 이 기능은 현재 고려 중이며, 이후에 사용할 수도 있고 사용하지 못할 수도 있습니다.
- Traditional 이 기능은 기존 클라이언트에서만 지원합니다.
- Salt 이 기능은 Salt 클라이언트에서만 지원합니다.

표 13. Ubuntu 운영 체제에서 지원하는 기능

Feature	Ubuntu 18.04	Ubuntu 20.04	Ubuntu 22.04
Client	✓	✓	✓
System packages	Canonical	Canonical	Canonical
Registration	Salt	Salt	Salt
Install packages	Salt	Salt	Salt
Apply patches	✓	✓	✓
Remote commands	Salt	Salt	Salt
System package states	Salt	Salt	Salt
System custom states	Salt	Salt	Salt

Feature	Ubuntu 18.04	Ubuntu 20.04	Ubuntu 22.04
Group custom states	Salt	Salt	Salt
Organization custom states	Salt	Salt	Salt
System set manager (SSM)	Salt	Salt	Salt
Product migration	N/A	N/A	N/A
Basic Virtual Guest Management *	Salt	Salt	Salt
Advanced Virtual Guest Management *	Salt	Salt	Salt
Virtual Guest Installation (Kickstart), as Host OS	✗	✗	✗
Virtual Guest Installation (image template), as Host OS	Salt	Salt	Salt
System deployment (PXE/Kickstart)	✗	✗	✗
System redeployment (Kickstart)	✗	✗	✗
Contact methods	Salt: ZeroMQ, Salt-SSH	Salt: ZeroMQ, Salt-SSH	Salt: ZeroMQ, Salt-SSH
Works with Uyuni Proxy	Salt	Salt	Salt
Action chains	Salt	Salt	Salt
Staging (pre-download of packages)	Salt	Salt	Salt
Duplicate package reporting	Salt	Salt	Salt
CVE auditing	?	?	?
SCAP auditing	?	?	?
Package verification	✗	✗	✗
Package locking	✓	✓	✓
System locking	✗	✗	✗
System snapshot	✗	✗	✗
Configuration file management	Salt	Salt	Salt
Package profiles	Salt: Profiles supported, Sync not supported	Salt: Profiles supported, Sync not supported	Salt: Profiles supported, Sync not supported
Power management	✓	✓	✓
Monitoring	Salt	Salt	Salt

Feature	Ubuntu 18.04	Ubuntu 20.04	Ubuntu 22.04
Docker buildhost	?	?	?
Build Docker image with OS	Salt	Salt	Salt
Kiwi buildhost	✗	✗	✗
Build Kiwi image with OS	✗	✗	✗
Recurring Actions	Salt	Salt	Salt
AppStreams	N/A	N/A	N/A
Yomi	N/A	N/A	N/A

＊ 가상 게스트 관리:

이 표에서 가상 게스트 관리는 기본과 고급으로 구분되어 있습니다.

기본 가상 게스트 관리에는 VM, 느린 새로 고침, VM 라이프싸이클 작업(시작, 중지, 다시 시작, 일시 중지) 나열과 VM vCPU 및 메모리 수정이 포함됩니다.

고급 가상 게스트 관리에는 기본 가상 게스트 관리의 모든 기능과 함께 빠른 새로 고침, VM 라이프싸이클 작업(삭제, 재설정, 전원 끄기), VM 디스크, 네트워크, 그래픽 표시 및 그래픽 표시 구성 수정이 포함됩니다.

Chapter 2. 구성 기본

Uyuni에서는 클라이언트 등록을 위한 환경을 준비하기 위해 여러 단계를 거친 후 매우 다양한 작업을 활용해야 합니다.

이 섹션에는 성공적인 Uyuni 설치 및 설정 후 환경 작업을 지원하기 위해 필요한 초기 구성 단계에 대한 요약이 포함되어 있습니다.

- Uyuni 설치에 대한 자세한 정보는 [Installation-and-upgrade > Install-uyuni](#)에서 참조하십시오.
- Uyuni 설정에 대한 자세한 정보는 [Installation-and-upgrade > Uyuni-server-setup](#)에서 참조하십시오.

2.1. 소프트웨어 채널

채널은 소프트웨어 패키지를 그룹화하는 방법입니다. 소프트웨어 패키지는 리포지토리가 제공하며, 리포지토리는 채널과 연결되어 있습니다. 클라이언트를 소프트웨어 채널에 가입하면 클라이언트는 자신과 연결된 모든 소프트웨어를 설치하고 업데이트할 수 있습니다.

Uyuni에서 채널은 기본 채널과 하위 채널로 구분됩니다. 이런 식으로 채널을 구성하면 호환되는 패키지만 각 시스템에 설치됩니다. 클라이언트는 등록 중에 클라이언트 운영 체제 및 아키텍처에 따라 할당된 하나의 기본 채널에만 가입되어야 합니다. 벤더가 제공하는 유료 채널에 대해서는 연결된 구독이 있어야 합니다.

기본 채널은 특정 운영 체제 유형, 버전 및 아키텍처를 위해 구축된 패키지로 구성됩니다. 예를 들어 SUSE Linux Enterprise Server 15 x86-64 기본 채널에는 이 운영 체제 및 아키텍처와 호환되는 소프트웨어만 포함되어 있습니다.

하위 채널은 기본 채널과 연결되어 있으며 기본 채널과 호환되는 패키지만 제공합니다. 시스템은 기본 채널의 여러 하위 채널에 가입할 수 있습니다. 시스템이 기본 채널에 할당된 경우 해당 시스템은 관련 하위 채널만 설치할 수 있습니다. 예를 들어 시스템에 SUSE Linux Enterprise Server 15 **x86_64** 기본 채널이 할당된 경우 호환되는 기본 채널이나 기본 채널과 연결된 하위 채널 중 하나를 통해 사용할 수 있는 패키지만 설치하거나 업데이트할 수 있습니다.

Uyuni Web UI에서 [소프트웨어 > 채널 목록 > 전체](#)로 이동하여 사용 가능한 채널을 검색할 수 있습니다. [소프트웨어 > 관리 > 채널](#)로 이동하여 새 채널을 수정하거나 생성할 수 있습니다.

사용자 지정 채널을 포함한 채널 사용에 대한 자세한 내용은 [Administration > Channel-management](#)을 참조하십시오.

2.1.1. SUSE Package Hub가 제공하는 패키지

SUSE Package Hub는 SUSE Linux Enterprise 제품의 확장으로, openSUSE 커뮤니티가 제공하는 추가 오픈 소스 소프트웨어를 제공합니다.



- SUSE Package Hub의 패키지는 openSUSE 커뮤니티가 제공합니다. 이 패키지는 SUSE에서 지원되지 않습니다.

클라이언트에서 SUSE Linux Enterprise 운영 체제를 사용 중인 경우 SUSE Package Hub 확장을 활성화하여 이러한 추가 패키지에 액세스할 수 있습니다. 이렇게 하면 클라이언트를 가입할 수 있는 SUSE Package Hub 채널이 제공됩니다.

SUSE Package Hub는 다수의 패키지를 제공하므로 동기화하는 데 시간이 오래 걸리고 대량의 디스크 공간을 소비할 수 있습니다. SUSE Package Hub가 제공하는 패키지가 필요하지 않은 경우 SUSE Package Hub를 활성화하지

마십시오.

지원되지 않는 패키지를 의도치 않게 설치하거나 업데이트하지 않도록 하려면 처음에 모든 SUSE Package Hub 패키지를 거부하는 컨텐트 라이프싸이클 관리 전략을 구현하는 것이 좋습니다. 그러면 필요한 특정 패키지를 명시적으로 활성화할 수 있습니다. 컨텐트 라이프싸이클 관리에 대한 자세한 내용은 [Administration > Content-lifecycle](#)을 참조하십시오.

2.1.2. AppStream이 제공하는 패키지

Red Hat 기반 클라이언트의 경우 AppStream을 통해 추가 패키지가 제공됩니다. 대부분의 경우 AppStream 패키지가 있어야 필요한 소프트웨어를 모두 보유할 수 있습니다.

Uyuni Web UI에서 AppStream 패키지를 관리할 때 패키지 업데이트에 대한 모순되는 제안을 볼 수도 있습니다. 이는 Uyuni(가) 모듈 메타데이터를 올바르게 해석할 수 없기 때문입니다. 컨텐트 라이프싸이클 관리(CLM) AppStream 필터를 사용하여 AppStream 리포지토리를 일부 업그레이드 작업에서 사용할 비모듈형 리포지토리로 변환할 수 있습니다. CLM AppStream 필터에 대한 자세한 설명은 [Administration > Content-lifecycle-examples](#)에서 참조하십시오.

2.1.3. EPEL이 제공하는 패키지

Red Hat 기반 클라이언트의 경우 EPEL(엔터프라이즈 Linux용 추가 패키지)을 통해 추가 패키지가 제공됩니다. EPEL은 추가 소프트웨어를 제공하는 옵션 패키지 리포지토리입니다.



- EPEL의 패키지는 Fedora 커뮤니티가 제공합니다. 이 패키지는 SUSE에서 지원하지 않습니다.

클라이언트에서 Red Hat 운영 체제를 사용 중인 경우 EPEL 확장을 활성화하여 이러한 추가 패키지에 액세스할 수 있습니다. 이렇게 하면 클라이언트를 가입할 수 있는 EPEL 채널이 제공됩니다.

EPEL은 다수의 패키지를 제공하므로 동기화하는 데 시간이 오래 걸리고 대량의 디스크 공간을 소비할 수 있습니다. EPEL이 제공하는 패키지가 필요하지 않은 경우 EPEL 리포지토리를 활성화하지 마십시오.

지원되지 않는 패키지를 의도치 않게 설치하거나 업데이트하지 않도록 하려면 처음에 모든 EPEL 패키지를 거부하는 컨텐트 라이프싸이클 관리(CLM) 전략을 구현하는 것이 좋습니다. 그러면 필요한 특정 패키지를 명시적으로 활성화할 수 있습니다. 컨텐트 라이프싸이클 관리에 대한 자세한 내용은 [Administration > Content-lifecycle](#)을 참조하십시오.

2.1.4. SUSE Linux Enterprise 클라이언트의 Unified Installer 업데이트 채널

이 채널은 운영 체제를 설치하기 전에 최신 상태인지 확인하기 위해 Unified Installer에서 사용합니다. 모든 SUSE Linux Enterprise 제품은 설치 중에 설치 프로그램 업데이트 채널에 액세스할 수 있는 권한이 있어야 합니다.

SUSE Linux Enterprise Server 클라이언트의 경우 설치 프로그램 업데이트 채널은 이 채널을 포함하는 제품을 추가하면 기본적으로 동기화되고 이 제품 채널로 자동 설치 가능한 배포판을 생성하면 활성화됩니다.

SAP용 SUSE Linux Enterprise를 포함한 다른 모든 SUSE Linux Enterprise 변형의 경우 설치 프로그램 업데이트 채널을 수동으로 추가해야 합니다. 이 작업을 수행하려면 SUSE Linux Enterprise 변형의 기본 채널 아래에 적절한 SUSE Linux Enterprise Server 설치 프로그램 업데이트 채널을 복제해야 합니다. 채널을 복제한 후 SUSE Linux Enterprise 변형에 대해 자동 설치 가능한 배포판을 생성할 때 이 채널이 자동으로 사용됩니다.

2.1.5. 소프트웨어 리포지토리

리포지토리는 소프트웨어 패키지를 수집하는 데 사용됩니다. 소프트웨어 리포지토리에 액세스할 수 있으면 리포지토리가 제공하는 모든 소프트웨어를 설치할 수 있습니다. 클라이언트를 채널에 할당하고 클라이언트에서 패키지를 설치하고 업데이트하려면 Uyuni에 소프트웨어 채널과 연결된 리포지토리가 최소 한 개 있어야 합니다.

Uyuni의 기본 채널은 대부분 이미 올바른 리포지토리에 연결되어 있습니다. 사용자 정의 채널을 생성하려면 액세스할 수 있는 또는 사용자가 직접 생성한 리포지토리를 연결해야 합니다.

사용자 정의 리포지토리에 대한 자세한 내용은 [Administration > Custom-channels](#)에서 참조하십시오.

2.1.5.1. 로컬 리포지토리 위치

Salt 클라이언트에서 로컬 리포지토리를 구성하여 Uyuni 채널이 공급하지 않는 패키지를 제공할 수 있습니다.



대부분의 경우 클라이언트 시스템에는 로컬 리포지토리가 필요 없습니다. 로컬 리포지토리로 인해 클라이언트에서 어떤 패키지를 사용할 수 있는지 파악하는 데 문제가 생길 수 있습니다. 그러면 결국 예상치 않은 패키지를 설치하게 될 수 있습니다.

온보딩 중에는 로컬 리포지토리가 비활성화됩니다.

Salt 클라이언트의 경우 채널 상태가 실행될 때마다 로컬 리포지토리가 비활성화됩니다. 예를 들어, highstate를 적용하거나 패키지 작업을 수행하는 경우가 이에 해당합니다.

온보딩 후 로컬 리포지토리를 활성화 상태로 유지해야 하는 경우 영향을 받는 Salt 클라이언트에 대해 설정해야 하는 열은 다음과 같습니다.

`/srv/pillar/top.sls` 파일 편집:

```
base:
  'minionid':
    - localrepos
```

`/srv/pillar/localrepos.sls` 파일 편집:

```
mgr_disable_local_repos: False
```

클라이언트가 온보딩을 완료하고 나면 다음 위치에 로컬 리포지토리를 추가할 수 있습니다.

표 14. 로컬 리포지토리 위치

Client Operating System	Local Repository Directory
SUSE Linux Enterprise Server	<code>/etc/zypp/repos.d</code>
openSUSE	<code>/etc/zypp/repos.d</code>
Red Hat Enterprise Linux	<code>/etc/yum.repos.d/</code>

Client Operating System	Local Repository Directory
CentOS	/etc/yum.repos.d/
Ubuntu	/etc/apt/sources.list.d/
Debian	/etc/apt/sources.list.d/

2.1.6. 소프트웨어 제품

Uyuni에서는 소프트웨어를 제품 내에서 제공합니다. SUSE 구독을 통해 다양한 제품에 액세스할 수 있습니다. 이러한 제품은 Uyuni Web UI에서 관리 > 설치 마법사 > 제품으로 이동하여 검색하고 선택할 수 있습니다.

제품에는 소프트웨어 채널이 포함되어 있는데, 그 개수는 다양합니다. 제품 채널 표시 아이콘을 클릭하여 제품에 포함된 채널을 확인합니다. 제품을 추가하고 동기화를 완료하고 나면 제품이 제공하는 채널에 액세스할 수 있고 Uyuni 서버 및 클라이언트에서 제품에 있는 패키지를 사용할 수 있습니다.

프로시저: 소프트웨어 채널 추가

1. Uyuni Web UI에서 관리 > 설치 마법사 > 제품으로 이동합니다.
2. 검색 창을 사용하여 클라이언트 운영 체제 및 아키텍처에 적합한 제품을 찾고 해당 제품을 확인하십시오. 모든 필수 채널은 자동으로 확인됩니다. 또한, 추천 포함 토글이 켜져 있으면 모든 추천 채널이 확인됩니다. 관련 제품의 전체 목록을 보려면 화살표를 클릭하고 필요한 추가 제품이 선택되어 있는지 확인하십시오.
3. [제품 추가]를 클릭하고 제품이 동기화를 마칠 때까지 기다립니다.

자세한 설명은 Installation-and-upgrade > Setup-wizard를 참조하십시오.

2.2. 부트스트랩 리포지토리

부트스트랩 리포지토리에는 부트스트랩 중에 Salt 또는 기존 클라이언트를 등록하기 위한 필수 패키지와 클라이언트에 Salt를 설치하기 위한 패키지가 포함되어 있습니다. 제품이 동기화되면 부트스트랩 리포지토리가 자동으로 생성되고 Uyuni 서버에서 다시 생성됩니다.

2.2.1. 부트스트랩 리포지토리 생성 준비

동기화를 위해 제품을 선택하면 모든 필수 채널이 완전히 미러링되자마자 부트스트랩 리포지토리가 자동으로 생성됩니다.

프로시저: Web UI에서 동기화 진행 상태 확인

1. Uyuni Web UI에서 소프트웨어 > 관리 > 채널로 이동한 다음, 리포지토리에 연결된 채널을 클릭합니다.
2. 리포지토리 탭으로 이동한 다음, 동기화를 클릭하고 동기화 상태를 확인합니다.

프로시저: 명령 프롬프트에서 동기화 진행 상태 확인

1. Uyuni 서버의 명령 프롬프트에서 루트 권한으로 다음과 같이 tail 명령을 사용해 동기화 로그 파일을 확인합니다.

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 각 하위 채널은 동기화가 진행되는 동안 자체 로그를 생성합니다. 동기화가 완료되었는지 알아보려면 모든 기본 및 하위 채널 로그 파일을 확인해야 합니다.

2.2.2. 자동 모드를 위한 옵션

자동 부트스트랩 리포지토리가 생성되는 방식을 변경할 수 있습니다. 이 섹션에서는 다양한 설정에 대해 자세히 설명합니다.

플러시 모드::

플러시 모드

기본적으로 기존 리포지토리는 최신 패키지로만 업데이트됩니다. 대신, 빈 리포지토리로 항상 시작하도록 구성할 수 있습니다. 이 동작을 활성화하려면 다음과 같이 `/etc/rhn/rhn.conf`에서 이 값을 추가하거나 편집하십시오.

```
server.susemanager.bootstrap_repo_flush = 1
```

자동 모드::

자동 모드

기본적으로 부트스트랩 리포지토리 자동 재생성은 활성화되어 있습니다. 이 작업을 비활성화하려면 다음과 같이 `/etc/rhn/rhn.conf`에서 이 값을 추가하거나 편집하십시오.

```
server.susemanager.auto_generate_bootstrap_repo = 0
```

2.2.2.1. 부트스트랩 데이터 파일 구성

도구는 각 배포에 어떤 패키지가 필요한지에 대한 정보가 포함된 데이터 파일을 사용합니다. 이 데이터 파일은 `/usr/share/susemanager/mgr_bootstrap_data.py`에 저장되어 있습니다. SUSE는 이 파일을 정기적으로 업데이트합니다. 이 파일을 변경하고 싶은 경우 직접 편집하지 마십시오. 대신, 다음과 같이 동일한 디렉토리에 사본을 생성하고 이 사본을 편집하십시오.

```
cd /usr/share/susemanager/
cp mgr_bootstrap_data.py my_data.py
```

변경을 완료했으면 새 파일을 사용하도록 Uyuni(를) 구성하십시오. 다음과 같이 `/etc/rhn/rhn.conf`에서 이 값을 추가하거나 편집하십시오.

```
server.susemanager.bootstrap_repo_datamodule = my_data
```



- 다음 업데이트에서 SUSE의 새 데이터는 새 데이터 파일이 아닌 원본 데이터 파일을 덮어씁니다. SUSE에서 제공하는 변경 사항으로 새 파일을 최신 상태로 유지해야 합니다.

2.2.3. 부트스트랩 리포지토리를 수동으로 생성

기본적으로 부트스트랩 리포지토리는 매일 다시 생성됩니다. 명령 프롬프트에서 부트스트랩 리포지토리를 수동으로 생성할 수 있습니다.

프로시저: SUSE Linux Enterprise용 부트스트랩 리포지토리 생성

1. Uyuni 서버의 명령 프롬프트에서 루트 권한으로 다음과 같이 부트스트랩 리포지토리를 생성하는 데 사용 가능한 배포를 나열합니다.

```
mgr-create-bootstrap-repo -l
```

2. 다음과 같이 적절한 리포지토리 이름을 제품 레이블로 사용해 부트스트랩 리포지토리를 생성합니다.

```
mgr-create-bootstrap-repo -c SLE-version-x86_64
```

3. 또는 사용 가능한 배포 목록에서 배포 이름 옆에 표시된 번호를 사용하십시오.

클라이언트 리포지토리는 </srv/www/htdocs/pub/repositories/>에 있습니다.

제품을 두 개 이상 미러링한 경우(예: SLES, SAP용 SLES) 또는 사용자 지정 채널을 사용하는 경우 부트스트랩 리포지토리를 생성할 때 사용할 상위 채널을 지정해야 합니다. 이는 모든 상황에서 필수가 아닙니다. 예를 들어, 일부 SLES 15 버전에는 공통 코드 기반이 있으므로, 상위 채널을 지정할 필요가 없습니다. 환경에서 필요한 경우에만 이 프로시저를 사용하십시오.

선택 사항 프로시저: 부트스트랩 리포지토리에 상위 채널 지정

1. 사용 가능한 상위 채널을 확인하십시오.

```
mgr-create-bootstrap-repo -c SLE-15-x86_64
여러 개의 상위 채널용 옵션을 찾았습니다.
--with-parent-channel <label>옵션을 사용하고 다음 중 하나를 선택하십시오 .
- sle-product-sles15-pool-x86_64
- sle-product-sles_sap15-pool-x86_64
- sle-product-sled15-pool-x86_64
```

2. 다음과 같이 적절한 상위 채널을 지정합니다.

```
mgr-create-bootstrap-repo -c SLE-15-x86_64 --with-parent-channel sle-
product-sled15-pool-x86_64
```

2.2.3.1. 아키텍처가 여러 개인 리포지토리

아키텍처가 여러 개 포함된 부트스트랩 리포지토리를 생성하는 경우 모든 아키텍처가 올바르게 업데이트되는지 주의

깊게 살펴봐야 합니다. 예를 들어 SLE용 IBM Z 및 x86-64 아키텍처가 /srv/www/htdocs/pub/repositories/sle/15/2/bootstrap/에서 동일한 부트스트랩 리포지토리 URL을 사용합니다.

flush 옵션이 활성화된 상태에서 여러 아키텍처에 대해 부트스트랩 리포지토리를 생성하려고 하면 하나의 아키텍처만 생성됩니다. 이를 방지하려면 추가 아키텍처를 생성할 때 명령 프롬프트에서 **--no-flush** 옵션을 사용하십시오. 예:

```
mgr-create-bootstrap-repo -c SLE-15-SP2-x86_64
mgr-create-bootstrap-repo --no-flush -c SLE-15-SP2-s390x
```

2.2.4. 부트스트랩 및 사용자 지정 채널

사용자 지정 채널을 사용 중인 경우 **--with-custom-channels** 옵션을 **mgr-create-bootstrap-repo** 명령과 함께 사용할 수 있습니다. 이 경우 사용할 상위 채널도 지정해야 합니다.

사용자 지정 채널을 사용 중인 경우 부트스트랩 리포지토리 자동 생성에 실패할 수 있습니다. 이러한 경우에는 리포지토리를 수동으로 생성해야 합니다.

사용자 지정 채널에 대한 자세한 내용은 [Administration > Custom-channels](#)에서 참조하십시오.

2.3. 활성화 키

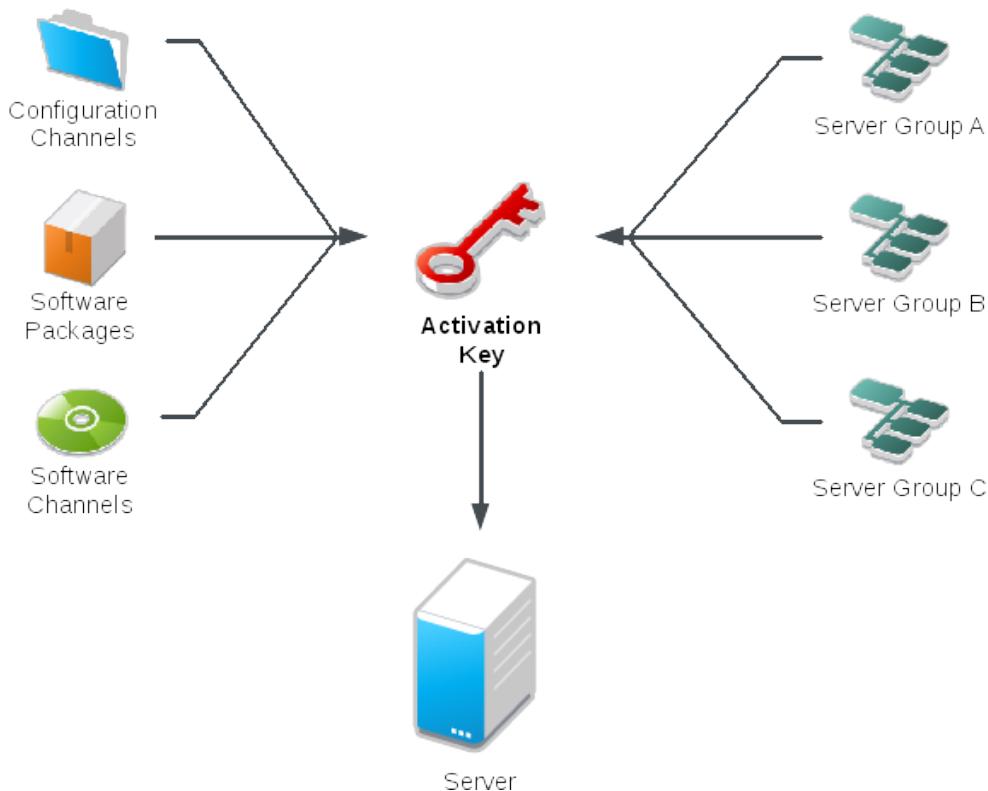
활성화 키는 기존 및 Salt 클라이언트에서 클라이언트가 올바른 소프트웨어 자격이 있는지, 적절한 채널에 연결하고 있는지, 관련 그룹에 가입되어 있는지 확인하는 데 사용됩니다. 각 활성화 키는 조직에 구속되며 키를 생성할 때 설정할 수 있습니다.

Uyuni에서 활성화 키는 레이블이 있는 구성 설정 그룹입니다. 레이블을 부트스트랩 스크립트에 파라미터로 추가하여 활성화 키와 연결된 모든 구성 설정을 적용할 수 있습니다. 활성화 키 레이블을 부트스트랩 스크립트와 함께 사용하는 것이 좋습니다. 부트스트랩 스크립트가 실행될 때 레이블에 연결된 모든 구성 설정은 스크립트가 실행되는 시스템에 적용됩니다.

활성화 키는 다음을 지정할 수 있습니다.

- 채널 할당
- 시스템 유형 또는 추가 자격
- 연락 방법
- 구성 파일
- 설치할 패키지
- 시스템 그룹 할당

활성화 키는 클라이언트 등록 시 사용되며 다시 사용되지 않습니다. 클라이언트는 등록된 후에 지정된 활성화 키와 관계없이 변경할 수 있습니다. 활성화 키와 클라이언트 사이의 관계는 기록 목적으로만 기록됩니다.



프로시저: 활성화 키 생성

1. Uyuni Web UI에서 관리자 권한으로 **시스템 > 활성화 키**로 이동합니다.
2. **[키 생성]** 버튼을 클릭합니다.
3. **활성화 키 상세 정보** 페이지의 **설명** 필드에 활성화 키의 설명을 입력합니다.
4. **키** 필드에 활성화 키의 이름을 입력합니다. 예를 들어, SUSE Linux Enterprise Server 15 SP4에 대해서는 **SLES15-SP4**를 입력합니다.



SUSE 제품인 경우 **키** 필드에 쉼표를 사용하지 마십시오. 하지만 Red Hat 제품인 경우 반드시 쉼표를 사용해야 합니다. 자세한 내용은 **Reference > Systems** 를 참조하십시오.

5. **기본 채널** 드롭다운 상자에서 적절한 기본 소프트웨어 채널을 선택하고, 관련 하위 채널이 작성되도록 허용합니다. 자세한 내용은 [reference:admin/setup-wizard.pdf](#) 및 **Administration > Custom-channels**를 참조하십시오.
6. 필요한 하위 채널을 선택합니다(예: 필수 SUSE Manager 도구 및 업데이트 채널).
7. 옵션을 활성화하려면 **추가 시스템 유형** 확인란을 선택합니다.
8. **연결 방법**은 **기본값** 으로 설정된 상태 그대로 두는 것이 좋습니다.
9. **범용 기본값** 설정은 확인란이 선택되지 않은 상태 그대로 두는 것이 좋습니다.
10. **[활성화 키 생성]**을 클릭하여 활성화 키를 생성합니다.
11. **구성 파일 배포** 확인란을 선택하여 이 키에 대한 구성 관리를 활성화하고, **[활성화 키 업데이트]**를 클릭하여 이 변경 사항을 저장합니다.



구성 파일 배포 확인란은 활성화 키를 생성한 후에야 표시됩니다. 구성 관리를 활성화해야 하는 경우 되돌아가 확인란을 선택하십시오.

2.3.1. 여러 활성화 키 결합

기존 클라이언트에서 부트스트랩 스크립트를 실행할 때 활성화 키를 결합할 수 있습니다. 키를 결합하면 시스템에서 설치된 것에 대해 더 많은 제어권을 확보할 수 있고 규모가 크거나 복잡한 환경에서 키가 중복되는 일이 줄어듭니다.



활성화 키 결합은 기존 클라이언트에서만 작동합니다. Salt 클라이언트는 결합된 활성화 키를 지원하지 않습니다. Salt 클라이언트에서 결합된 키를 사용하는 경우 첫 번째 키만 사용됩니다.

명령 프롬프트나 단일 자동 설치 프로파일에서 여러 개의 활성화 키를 지정할 수 있습니다.

Uyuni 서버의 명령 프롬프트에서 **rhnreg_ks** 명령을 사용하고, 키 이름을 쉼표로 구분하십시오. Kickstart 프로파일에서 여러 개의 키를 지정하려면 **시스템 > 자동 설치**로 이동하여 사용하려는 프로파일을 편집하십시오.

일부 값들 사이의 충돌로 인해 클라이언트 등록이 실패할 수 있으므로 활성화 키를 결합할 때 주의하십시오. 시작하기 전에 다음 값에 서로 충돌하는 정보가 있는지 확인하십시오.

- 소프트웨어 패키지
- 소프트웨어 하위 채널
- 구성 채널

충돌이 감지되면 다음과 같이 처리됩니다.

- 기본 소프트웨어 채널의 충돌: 등록에 실패합니다.
- 시스템 유형의 충돌: 등록에 실패합니다.
- **구성 활성화 플래그**의 충돌: 구성 관리가 활성화됩니다.
- 한 개의 키가 시스템 전용인 경우: 등록에 실패합니다.

2.3.2. 활성화 키

클라이언트를 다시 등록하고 모든 Uyuni 설정을 다시 확보하기 위한 목적으로 재활성화 키를 한 번만 사용할 수 있습니다. 재활성화 키는 클라이언트 전용이며 시스템 ID, 이력, 그룹 및 채널을 포함합니다.

재활성화 키를 생성하려면 **시스템**으로 이동한 후 클라이언트를 클릭하여 재활성화 키를 생성하고, **상세 정보 > 재활성 탭**으로 이동하십시오. **[새 키 생성]**을 클릭하여 재활성화 키를 생성합니다. 나중에 사용할 수 있도록 키의 상세 정보를 기록해 둡니다. 특정 시스템 ID와 연결되지 않은 일반적인 활성화 키와 달리 여기에서 생성한 키는 **시스템 > 활성화 키** 페이지에 표시되지 않습니다.

Salt 클라이언트의 경우 재활성화 키를 생성한 후 이 키를 **management_key** /etc/salt/minion.d/susemanager.conf에서 입자로 사용할 수 있습니다. 예:

```
grains:
  susemanager:
    management_key: "re-1-daf44db90c0853edbb5db03f2b37986e"
```

재활성화 키를 적용하려면 **salt-minion** 프로세스를 다시 시작하십시오.

재활성화 키를 부트스트랩 스크립트와 함께 사용할 수 있습니다. 부트스트랩 스크립트에 대한 자세한 내용은 [Client-configuration > Registration-bootstrap](#)을 참조하십시오.

기존 클라이언트의 경우 재활성화 키를 생성한 후 이 키를 **rhnreg_ks** 명령줄 유ти리티와 함께 사용할 수 있습니다. 이 명령은 클라이언트를 다시 등록하고 Uyuni 설정을 복원합니다. 기존 클라이언트에서는 재활성화 키를 활성화 키와 결합하여 단일 시스템 프로파일에 대해 여러 키의 설정을 집계할 수 있습니다. 예:

```
rhnreg_ks --server=<server-url>/XMLRPC \
--activationkey=<reactivation-key>,<activationkey> \
--force
```



클라이언트를 기존 Uyuni 프로파일과 함께 자동 설치하는 경우 프로파일은 재활성화 키를 사용해 시스템을 다시 등록하고 설정을 복원합니다. 프로파일 기반 자동 설치가 진행 중일 때는 이 키를 재생성, 삭제 또는 사용하지 마십시오. 그러면 자동 설치에 실패하게 됩니다.

2.3.3. 활성화 키 모범 사례

기본 상위 채널

SUSE Manager 기본값 상위 채널을 사용하지 마십시오. 이 설정은 Uyuni가 설치된 운영 체제와 가장 부합하는 상위 채널을 선택하도록 강제합니다. 이로 인해 때로 예기치 않은 동작이 발생할 수 있습니다. 대신에 각 배포 및 아키텍처별로 활성화 키를 생성하는 것이 좋습니다.

활성화 키를 이용한 부트스트래핑

부트스트랩 스크립트를 사용 중인 경우 각 스크립트에 대해 활성화 키를 생성하는 것을 고려하십시오. 이렇게 하면 채널 할당, 패키지 설치, 시스템 그룹 구성원, 구성 채널 할당을 조정하는 데 도움이 됩니다. 또한 등록 후 시스템에 대한 수동 개입이 덜 필요합니다.

대역폭 요구사항

활성화 키를 사용하면 등록 시점에 소프트웨어가 자동으로 다운로드될 수 있는데, 대역폭이 제한된 환경에서는 바람직하지 않을 수 있습니다.

이러한 옵션은 대역폭 사용량을 생성합니다.

- SUSE 제품 풀 채널을 할당하면 해당 제품 설명자 패키지가 자동으로 설치됩니다.
- 세션의 모든 패키지가 **Packages** 설치됩니다.
- **구성** 세션의 모든 Salt 상태가 그 내용에 따라 다운로드를 트리거할 수 있습니다.

키 레이블 명명

활성화 키에 대해 읽을 수 있는 이름을 입력하지 않으면 시스템이 수 스트링을 자동으로 생성하므로 키를 관리하기 어려울 수 있습니다.

키를 추적하는 데 도움이 되는 활성화 키 명명 스키마를 고려하십시오. 조직의 인프라와 관련이 있는 이름을 생성하면 더 복잡한 작업을 더 쉽게 수행할 수 있습니다.

키 레이블을 생성할 때 다음과 같은 팁을 참고하십시오.

- OS 명명(필수): 키는 항상 자신이 설정을 제공하는 대상인 OS를 참조해야 합니다.
- 아키텍처 명명(권장): 귀사가 단 하나의 아키텍처(예: x86_64)만 운영하고 있지 않다면 레이블에 아키텍처 유형을 제공하는 것이 좋습니다.
- 서버 유형 명명: 이 서버를 어떤 목적으로 사용 중입니까?
- 위치 명명: 서버가 어디에 있습니까? 서버룸, 건물 또는 부서 내?
- 날짜 명명: 유지보수 기간, 분기 등
- 사용자 정의 명명: 어떤 명명 스키마가 조직의 필요에 적합합니까?

활성화 키 레이블 이름의 예:

sles15-sp4-web_server-room_129-x86_64

sles15-sp4-test_packages-blg_502-room_21-ppc64le



SUSE 제품인 경우 키 필드에 쉼표를 사용하지 마십시오. 하지만 Red Hat 제품인 경우 반드시 쉼표를 사용해야 합니다. 자세한 내용은 [Reference > Systems](#)를 참조하십시오.

포함된 채널

활성화 키를 생성할 때는 어떤 소프트웨어 채널이 연결되어 있는지도 기억해 두어야 합니다. 키에는 할당된 특정 기본 채널이 있어야 합니다. 기본값으로 설정된 기본 채널은 사용하지 않는 것이 좋습니다. 자세한 내용은 [Client-configuration > Registration-overview](#)에서 설치하려는 클라이언트 운영 체제를 참조하십시오.

2.4. GPG 키

클라이언트는 소프트웨어 패키지가 설치되기 전에 GPG 키를 사용해 신뢰성을 확인합니다. 신뢰할 수 있는 소프트웨어만 클라이언트에 설치할 수 있습니다.

대부분의 경우 GPG 설정을 조정하지 않아도 클라이언트에 소프트웨어를 설치할 수 있습니다.

RPM 패키지에는 직접 서명할 수 있지만, Debian 기반 시스템은 메타데이터에만 서명하고 체크섬 체인을 사용하여 패키지를 보호합니다. 대부분의 RPM 기반 시스템은 서명된 패키지뿐만 아니라 서명된 메타데이터를 사용합니다.

2.4.1. 클라이언트의 GPG 키 신뢰

운영 체제는 자체 GPG 키를 직접 신뢰하거나 최소 시스템 이상과 함께 설치된 상태로 제공됩니다. 그러나 다른 GPG 키로 서명된 타사 패키지는 수동으로 처리해야 합니다. 클라이언트는 신뢰할 수 있는 GPG 키가 없어도 성공적으로 부트스트래핑할 수 있습니다. 하지만 키를 신뢰할 수 있을 때까지는 새 클라이언트 도구 패키지를 설치하거나 업데이트할 수 없습니다.

Salt clients use now GPG key information entered for a software channel to manage the trusted keys. When a software channel with GPG key information is assigned to a client, the key gets trusted as soon as the channel is refreshed or the first package gets installed from this channel.

소프트웨어 채널로 설정된 GPG 키 URL이 있어야 합니다. 파일 URL인 경우 클라이언트에 GPG 키 파일을 배포한 후 소프트웨어 채널을 사용해야 합니다.

The GPG keys for the Client Tools Channels of Red Hat based clients are deployed on the client into `/etc/pki/rpm-gpg/` and can be referenced with file URLs.