



Red Hat Satellite 6.1

프로비저닝 가이드

Red Hat Satellite 서버에서 물리적 호스트 및 가상 호스트 프로비저닝을 위한 가이드

역음 1

Red Hat Satellite 6.1 프로비저닝 가이드

Red Hat Satellite 서버에서 물리적 호스트 및 가상 호스트 프로비저닝을 위한 가이드
역음 1

Red Hat Satellite 문서화팀

법적 공지

Copyright © 2015 Red Hat.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

초록

Red Hat Satellite 프로비저닝 가이드는 물리적 호스트 및 가상 호스트 프로비저닝을 준비하는데 있어서 Red Hat Satellite를 설치 및 설정하기 위해 고안된 작업 기반 문서입니다. 이에는 네트워크 토폴로지 설정, 서비스 설정, 네트워크 상의 호스트를 프로비저닝하기 위해 필요한 기타 다른 모든 설정 정보가 들어 있습니다. 이 가이드는 네트워크 관련 숙련된 지식 및 기술을 갖고 있는 Satellite 관리자를 주요 대상으로 합니다.

차례

1장. RED HAT SATELLITE를 사용한 프로비저닝 방법 소개	3
1.1. RED HAT ENTERPRISE LINUX 호스트 생성	3
1.2. 네트워크 토폴로지 설정	3
1.3. 호스트 등록 및 설정	3
2장. RED HAT SATELLITE 서비스 설정	6
2.1. DNS, DHCP, TFTP 설정	6
2.2. 기본 조직 및 위치에 객체를 연결	8
3장. 서브스크립션 가져오기 및 콘텐츠 동기화	11
3.1. 매니페스트 생성	11
3.2. SATELLITE 서버에 매니페스트 업로드	11
3.3. RED HAT 리포지터리 활성화	11
3.4. 사용자 정의 제품 및 리포지터리 생성	12
3.5. 콘텐츠 동기화	13
4장. 콘텐츠 관리 및 승격	14
4.1. 애플리케이션 라이프 사이클 환경 생성	14
4.2. 콘텐츠 뷰 생성	14
4.3. RED HAT ENTERPRISE LINUX 리포지터리 추가	14
4.4. PUPPET 모듈 추가	15
4.5. 콘텐츠 뷰 공개	15
4.6. 활성화키 생성 및 편집	16
5장. 프로비저닝 설정 완료	17
5.1. 프로비저닝 템플릿 생성	17
5.2. 호스트 그룹 생성	17
6장. 호스트 프로비저닝	19
6.1. PXE를 사용하여 호스트 프로비저닝	19
6.2. 부팅 디스크를 사용하여 호스트 프로비저닝	19
6.3. 고정 IP 주소를 사용하여 호스트 프로비저닝	22
부록 A. 용어집	25
부록 B. 고친 과정	29

1장. RED HAT SATELLITE를 사용한 프로비저닝 방법 소개

다음 부분에서는 호스트를 프로비저닝하기 위해 Red Hat Satellite 서버를 설정하는 방법에 대해 설명합니다. 이에는 Red Hat Enterprise Linux 설치를 포함하여 전형적인 네트워크 토폴로지 및 사용 가능한 서비스, 호스트를 Red Hat Subscription Management에 등록하는 방법에 대해 설명합니다.

1.1. RED HAT ENTERPRISE LINUX 호스트 생성

@Core 또는 @Base 패키지 세트 중 하나를 사용하여 x86_64에 Red Hat Enterprise Linux 서버 버전 6.6 이상을 설치합니다. Red Hat Enterprise Linux 설치에 대한 보다 자세한 내용은 [Red Hat Enterprise Linux 6 설치 가이드](#)에서 참조하십시오.

1.2. 네트워크 토폴로지 설정

이 문서에서는 Satellite 6를 실행하고 있는 호스트가 DHCP, DNS, TFTP 서비스를 활성화할 수 있는 전용 서브넷이 배치되어 있다고 가정합니다. 예에서는 172.17.13.0/24 서브넷을 사용합니다. 또한 Satellite에 의해 관리되는 example.org DNS 도메인을 사용합니다.

또한 다음 예에서는 Satellite 호스트에 다음과 같은 네트워크 정보가 사용되고 있다고 가정합니다. 사용에 따라 이러한 매개 변수를 조정하십시오:

- 호스트 이름: satellite.example.org
- IP 주소 = 172.17.13.2
- 넷마스크: 255.255.255.0

1.3. 호스트 등록 및 설정

다음 부분에서는 호스트를 등록하는 방법, 서브스크립션을 확인하는 방법, 호스트가 콘텐츠를 사용할 수 있도록 이러한 서브스크립션을 할당하는 방법에 대해 설명합니다.

1.3.1. Red Hat Subscription Management에 등록

이 절차의 첫 번째 단계는 호스트를 Red Hat Subscription Management에 등록하는 것입니다. 이는 호스트를 등록하여 사용자가 사용할 수 있는 서브스크립션에 대한 콘텐츠를 사용할 수 있게 합니다. 이에는 Red Hat Enterprise Linux, Red Hat Software Collections (RHSC), Red Hat Satellite와 같은 콘텐츠가 포함되어 있습니다. `subscription_manager register` 명령을 사용하여 Satellite를 등록합니다:

```
# subscription-manager register
Username: demouser
Password:
The system has been registered with ID: 541084ff2-44cab-4eb1-9fa1-7683431bcf9a
```

1.3.2. Satellite 서브스크립션 확인

호스트를 등록한 후 Satellite 서브스크립션 풀 ID를 확인해야 합니다. 이 ID를 사용하여 호스트에 필요한 서브스크립션을 할당할 수 있습니다. Satellite 서브스크립션은 Satellite 콘텐츠 및 Red Hat Enterprise Linux, Red Hat Software Collections (RHSC), Red Hat Satellite로의 액세스를 제공합니다. 이에는 서브스크립션만 필요합니다.

다음 명령을 실행하여 Satellite 서브스크립션을 확인합니다:

```
# subscription-manager list --all --available
Subscription Name: Red Hat Satellite
Provides:          Oracle Java (for RHEL Server)
                  Red Hat Satellite 6 Beta
                  Red Hat Enterprise Linux Server
                  Red Hat Satellite
                  Red Hat Enterprise Linux Load Balancer (for RHEL
Server)
SKU:              MCT0370
Pool ID:          8a85f9874152663c0541943739717d11
Available:        3
Suggested:        1
Service Level:    Premium
Service Type:     L1-L3
Multi-Entitlement: No
Ends:             10/07/2014
System Type:      Physical
```



중요

풀 ID를 적어둡니다. 이 값을 사용하여 서브스크립션을 Satellite 호스트에 할당할 수 있습니다. 예에서 풀 ID는 8a85f9874152663c0541943739717d11입니다. 실제 서브스크립션의 풀 ID는 이와 다릅니다.

다음 명령을 실행하여 서브스크립션을 Satellite에 할당합니다. 실제 자신의 풀 ID로 바꿔야 합니다:

```
# subscription-manager attach --pool=8a85f9874152663c0541943739717d11
Successfully attached a subscription for: Red Hat Satellite
```

다음 명령을 실행하여 서브스크립션이 올바르게 할당되었는지 확인합니다:

```
# subscription-manager list --consumed
+-----+
Consumed Subscriptions
+-----+
Subscription Name: Red Hat Satellite
Provides:          Red Hat Satellite
                  Red Hat Enterprise Linux Server
                  Red Hat Software Collections (for RHEL Server)
                  Red Hat Satellite Beta
                  Red Hat Satellite 6 Beta
                  Red Hat Software Collections Beta (for RHEL Server)
                  Red Hat Satellite Capsule Beta
                  Red Hat Enterprise Linux Load Balancer (for RHEL
Server)
                  Red Hat Satellite with Embedded Oracle
                  Red Hat Satellite Capsule
                  Red Hat Enterprise Linux High Availability (for RHEL
Server)
SKU:              MCT0370
Contract:         10293569
Account:          5361051
```

```
Serial:          1653856191250699363
Pool ID:         8a85f9874152663c0541943739717d11
Active:          True
Quantity Used:   1
Service Level:   Premium
Service Type:    L1-L3
Status Details:
Starts:          10/08/2013
Ends:            10/07/2014
System Type:     Physical
```

1.3.3. 리포지터리 설정

기존 모든 리포지터리를 비활성화하고 Red Hat Enterprise Linux 6, Red Hat Software Collections, Satellite 6 리포지터리만 활성화합니다. 다음 명령을 실행합니다:

```
# subscription-manager repos --disable=*
# subscription-manager repos --enable rhel-6-server-rpms \
--enable rhel-server-rhsc1-6-rpms --enable rhel-6-server-satellite-6.0-
rpms
```

1.3.4. Satellite 6 설치

다음 명령을 실행하여 Satellite 6를 설치합니다:

```
# yum install katello
```

2장. RED HAT SATELLITE 서비스 설정

설정 예시에서 Satellite는 172.17.13.0/24 서브넷에 있는 호스트를 프로비저닝해야 합니다. 다음 부분에서는 서브넷에서 프로비저닝되는 클라이언트를 제공하기 위해 DNS, DHCP, TFTP를 설정하는 방법에 대해 설명합니다.

2.1. DNS, DHCP, TFTP 설정

다음 부분에서는 **example.org** 도메인 및 172.17.13.x 서브넷의 신뢰할 수 있는 DNS 서비스를 제공하기 위해 BIND (**named**)를 실행하도록 Satellite를 설정하는 방법에 대해 설명합니다. 이에는 **example.org** 영역 파일에 포함된 정방향 검색을 위한 DNS 영역 설정이 필요합니다. 또한 역방향 검색을 위한 DNS 영역은 172.17.13.x 서브넷에 생성되어 **13.17.172.in-addr.arpa** 역방향 영역 파일에 포함됩니다. 이는 Satellite에서 프로비저닝된 호스트가 올바른 이름 확인 매개 변수를 사용하게 합니다. 다음 부분에서는 PXE를 사용하여 호스트를 부팅할 수 있도록 TFTP 프록시를 설정하는 방법에 대해서도 설명합니다.

이 네트워크에 있는 클라이언트는 다음과 같은 특징이 있습니다:

- DHCP의 경우 172.17.13.100에서 172.17.13.150 범위에 있는 IP 주소에 액세스할 수 있습니다.
- DNS의 경우 Satellite (172.17.13.2의 **satellite.example.org**)를 사용합니다.
- PXE 부팅을 활성화하려면 Satellite (172.17.13.2의 **satellite.example.org**)에서 **pxelinux.0** 파일을 수신합니다.
- 호스트 이름은 **hostname.example.org**이며 여기서 **hostname**은 호스트 프로비저닝 시 설정됩니다.



중요

이 예에서는 Satellite 서버에서 DHCP 서비스를 사용합니다. 작업을 계속 진행하기 전 네트워크 관리자에게 문의하십시오.

2.1.1. Satellite 설정 옵션

다음 표에서는 Satellite 서버를 올바르게 설정하기 위해 필요한 다양한 옵션 및 값에 대해 설명합니다. **katello-installer** 명령은 Puppet을 사용합니다. 따라서 이는 추가 패키지 (**bind**, **dhcp**, **xinetd**, 등)를 설치 및 설정하여 필요한 기능을 추가합니다.

사용 가능한 옵션에 대한 전체 목록을 보려면 **katello-installer --help**를 실행합니다.

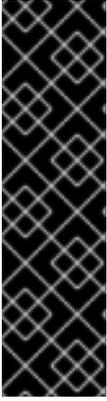
표 2.1. Satellite 설정 옵션

옵션	설명	값
<code>--foreman-admin-username</code>	초기 관리자의 사용자 이름입니다.	(사용자 지정)
<code>--foreman-admin-password</code>	초기 관리자의 암호입니다.	(사용자 지정)

옵션	설명	값
--capsule-dns	DNS 프록시 기능을 활성화합니다.	yes
--capsule-dns-interface	named 가 수신하는 인터페이스입니다.	eth0
--capsule-dns-zone	Satellite가 호스팅하는 정방향 DNS 영역입니다	example.org
--capsule-dns-forwarders	알 수 없는 쿼리가 전송되는 DNS 서버입니다.	172.17.13.1
--capsule-dns-reverse	Satellite가 호스트하는 역방향 DNS 영역입니다. 이는 일반적으로 역방향 IP 주소 (172.17.13)의 처음 세 개의 옥텟으로 ".in-addr.arpa"가 추가됩니다.	13.17.172.in-addr.arpa
--capsule-dhcp	DHCP 프록시 기능을 활성화합니다.	yes
--capsule-dhcp-interface	DHCP가 수신하는 인터페이스입니다.	eth0
--capsule-dhcp-range	클라이언트에 발급할 IP 주소 범위입니다.	172.17.13.100 172.17.13.150
--capsule-dhcp-gateway	클라이언트에 발급할 기본 게이트웨이 IP입니다.	172.17.13.1
--capsule-dhcp-nameservers	클라이언트가 이름 확인을 위해 사용해야 하는 호스트입니다. 이는 배포 모델에서 Satellite의 IP 주소로 설정해야 합니다.	172.17.13.2
--capsule-tftp	TFTP 프록시 기능을 활성화합니다. 이는 클라이언트를 PXE 부팅하는데 필요합니다.	yes
--capsule-tftp-servername	TFTP 호스트 이름을 설정합니다. 서버의 호스트 이름 (satellite.example.org)과 일치하도록 이를 설정합니다.	\$(hostname)
--capsule-puppet	Puppet 마스터를 사용합니다.	yes
--capsule-puppetca	Puppet CA를 사용합니다.	yes

2.1.2. Satellite 서비스 설정

Satellite 서버에서 필요한 서비스를 설정하기 위해 지정된 옵션을 사용하여 root로 다음의 **katello-installer** 명령을 실행합니다. 원하는 관리자 사용자 이름 및 암호로 바꿔야 합니다.



중요

- **설치 가이드**에 있는 지침을 따라 Satellite를 이미 설치한 경우 다음 명령에 **--foreman-admin-username** 및 **--foreman-admin-password** 옵션을 포함하지 마십시오.
- 관리자 사용자 이름 및 암호를 지정하지 않으면 기본 사용자 **admin**이 생성되어 암호가 자동으로 생성됩니다. 인증은 설치 프로세스의 마지막에 표시됩니다. 이 암호를 메모해 둡니다. **/etc/katello-installer/answers.katello-installer.yaml** 파일에 있는 **admin_password** 매개 변수에서 암호를 검색할 수도 있습니다.

```
katello-installer \
--foreman-admin-username admin-username \
--foreman-admin-password admin-password \
--capsule-dns true \
--capsule-dns-interface eth0 \
--capsule-dns-zone example.org \
--capsule-dns-forwarders 172.17.13.1 \
--capsule-dns-reverse 13.17.172.in-addr.arpa \
--capsule-dhcp true \
--capsule-dhcp-interface eth0 \
--capsule-dhcp-range "172.17.13.100 172.17.13.150" \
--capsule-dhcp-gateway 172.17.13.1 \
--capsule-dhcp-nameservers 172.17.13.2 \
--capsule-tftp true \
--capsule-tftp-servername $(hostname) \
--capsule-puppet true \
--capsule-puppetca true
```

설치 프로세스 마지막에 **katello-installer**는 설치 상태를 표시합니다.

```
Success!
* Katello is running at https://satellite.example.org
  Default credentials are 'admin:*****'
* Capsule is running at https://satellite.example.org:9090
* To install additional capsule on separate machine continue by
running:"

capsule-certs-generate --capsule-fqdn "$CAPSULE" --certs-tar
"~/ $CAPSULE-certs.tar"

The full log is at /var/log/katello-installer/katello-installer.log
```

웹 브라우저를 사용하여 <https://satellite.example.org>로 이동하면 Satellite 홈페이지가 나타납니다. 예에서는 기본 조직 (Default_Organization)과 기본 위치를 사용하고 있습니다.

2.2. 기본 조직 및 위치에 객체를 연결

Satellite 6는 여러 조직 (논리적 관리 부분) 및 위치 (컨텐츠 전달의 물리적 부분)를 지원하기 때문에 프로비저닝에 필요한 템플릿, 서브넷, 기타 다른 항목을 기본 조직 (Default_Organization) 및 기본 위치 (Default_Location)에 연결해야 합니다.

다음 절차에 따라 기본 조직 (Default_Organization)에서 모든 "pre-seeded" 콘텐츠를 사용할 수 있습니다:

절차 2.1. 기본 위치를 지정하려면 다음을 실행합니다:

1. 메인 메뉴에서 **관리** → **위치**를 클릭한 후 **이름** 칼럼에 있는 **Default_Location**을 클릭합니다.
2. **조직**을 클릭하여 조직 목록을 표시합니다.
3. **Default_Organization**을 클릭하여 이를 **선택한 항목** 목록에 추가한 후 **제출**을 클릭합니다.

절차 2.2. 기본 조직을 지정하려면 다음을 실행합니다:

1. 메인 메뉴에서 **관리** → **조직**를 클릭한 후 **이름** 칼럼에 있는 **Default_Organization**을 클릭합니다.
2. **위치**를 클릭하여 사용 가능한 위치 목록을 표시합니다.
3. **Default_Location**을 클릭하여 이를 **선택한 항목** 목록에 추가한 후 **제출**을 클릭합니다.

절차 2.3. 도메인을 기본 조직에 연결하려면 다음을 실행합니다:

1. 메인 메뉴에서 **인프라** → **도메인**을 클릭하여 **도메인** 화면을 엽니다.
2. **설명** 칼럼에서 **example.org**를 클릭하면 **도메인 편집** 화면이 열리고 도메인의 상세 정보를 업데이트할 수 있습니다.
3. **도메인** 탭에서 Satellite의 호스트 이름을 반영하도록 DNS 도메인을 변경합니다.
4. Satellite 서버에 **DNS Capsule** 값을 설정합니다.
5. **위치** 탭에서 **Default_Location**을 클릭하고 **선택한 항목** 목록에 추가합니다. 이는 도메인을 기본 위치에 연결합니다.
6. **조직** 탭에서 **Default_Organization**을 클릭하고 **선택한 항목** 목록에 추가합니다. 이는 도메인을 기본 조직에 연결합니다.
7. **제출**을 클릭하여 변경 사항을 적용합니다.

절차 2.4. 기본 서브넷을 선택하려면 다음을 실행합니다:

1. 메인 메뉴에서 **인프라** → **서브넷**을 클릭합니다.
2. **새 서브넷**을 클릭하고 다음 정보를 입력합니다. 사용자 배포에 맞게 상세 정보를 업데이트합니다:
 - 이름: Provisioning_Net
 - 네트워크 주소: 172.17.13.0
 - 넷마스크: 255.255.255.0
 - 게이트웨이 주소: 172.17.13.1
 - 1차 DNS 서버: 172.17.13.2
 - 2차 DNS 서버: 비워 둡

- IP 범위의 시작: 172.17.13.100
 - IP 범위의 끝: 172.17.13.150
 - VLAN ID: 비워 둠
3. **제출**을 클릭합니다.
 4. Provisioning_Net 클릭하여 서브넷을 편집합니다.
 5. **도메인** 탭에서 example.org를 선택합니다.
 6. **Capsules** 탭에서 Satellite의 호스트 이름을 반영하도록 DNS, DHCP, TFTP capsule을 변경합니다.
 7. **조직** 탭에서 **모든 항목** 아래에 있는 Default_Location을 선택하고 도메인을 기본 위치에 연결합니다.
 8. **조직** 탭에서 **모든 항목** 아래에 있는 Default_Organization을 선택하고 도메인을 기본 조직에 연결합니다.

절차 2.5. 설치 미디어의 조직 및 위치에 연결하려면 다음을 수행합니다:

1. 메인 메뉴에서 **호스트** → **설치 미디어**를 클릭합니다
2. **이름** 칼럼에서 사용하고자 하는 미디어 이름을 클릭합니다.
3. **위치** 탭에서 원하는 위치를 선택한 항목 목록에 추가합니다.
4. **조직** 탭에서 필요한 조직을 선택된 항목 목록에 추가하고 **제출**을 클릭합니다.

3장. 서브스크립션 가져오기 및 콘텐츠 동기화

다음 부분에서는 콘텐츠를 다운로드하고 관리하기 위해 Satellite를 설정하는 방법에 대해 설명합니다. 이에는 Satellite 서버에 매니페스트 파일 업로드, Red Hat 리포지터리 활성화, 사용자 정의 제품 생성 및 콘텐츠 동기화가 포함됩니다.

3.1. 매니페스트 생성

다음 부분에서는 Red Hat Satellite에 적합한 매니페스트를 생성하는 방법에 대해 설명합니다.

절차 3.1. Satellite 6에 매니페스트를 생성하려면 다음을 실행합니다:

1. access.redhat.com으로 이동하여 메인 메뉴에서 **서브스크립션**을 클릭합니다.
2. **Red Hat Subscription Management** 섹션에서 **서브스크립션 관리 애플리케이션** 하의 **Satellite**를 클릭합니다.
3. 매니페스트를 생성해야 하는 시스템을 배치합니다. 올바른 버전이 선택되어 있는지 확인합니다.
4. 할당하고자 하는 각각의 서브스크립션에 대해 서브스크립션의 체크 박스를 선택하고 할당할 서브스크립션 수량을 지정합니다.
5. **선택 사항 할당**을 클릭합니다.



참고

모든 서브스크립션을 할당하는데 몇 분 정도 걸릴 수 있습니다. 서브스크립션이 할당되었다는 확인을 받을 때 까지 몇 분 간격으로 화면을 새로고침합니다.

6. 서브스크립션을 할당한 후 **매니페스트 다운로드**를 클릭하여 알려진 위치에 매니페스트 파일을 저장합니다.

3.2. SATELLITE 서버에 매니페스트 업로드

절차 3.2. Satellite 서버에 매니페스트를 업로드하려면 다음을 수행합니다:

1. 올바른 조직을 아직 선택하지 않은 경우 **모든 컨텍스트** → **모든 조직** → **Default_Organization**을 클릭합니다.
2. **콘텐츠** → **Red Hat 서브스크립션**을 클릭합니다.
3. **매니페스트 관리**를 클릭하여 **서브스크립션** 페이지를 엽니다.
4. **검색**을 클릭하여 적절한 매니페스트를 선택한 후 **열기**를 클릭합니다.
5. **업로드**를 클릭하여 Satellite 서버에 매니페스트를 업로드합니다.

3.3. RED HAT 리포지터리 활성화

다음 부분에서는 Red Hat Enterprise Linux 6 호스트 프로비저닝을 지원하기 위해 필요한 Red Hat 리포지터리를 활성화하는 방법에 대해 설명합니다. 배포에 맞는 적절한 릴리즈를 선택합니다. 이 단계에서 다음과 같은 리포지터리를 사용합니다:

- Red Hat Enterprise Linux 6 Server (Kickstart): Satellite가 호스트의 킥스타트를 실행할 수 있습니다. 이는 Satellite 5 "kickstart tree"와 거의 유사합니다.
- Red Hat Enterprise Linux 6 Server (RPM): 이는 지속적인 콘텐츠 및 에라타를 제공합니다. 이는 Satellite 5 "rhel-x86_64-server-6" 채널과 거의 유사합니다.
- Red Hat Enterprise Linux 6 Server - Satellite Tools (RPMs): 이는 Puppet 및 katello-agent와 같은 소프트웨어를 지원합니다. 이는 Satellite 5 "rhn-tools-rhel-x86_64-server-6" 채널과 거의 유사합니다.

절차 3.3. Red Hat 리포지토리를 활성화하려면 다음을 실행합니다:

1. 콘텐츠 → **Red Hat** 리포지토리를 선택합니다.
2. **킥스타트** 탭에서 **Red Hat Enterprise Linux Server** → **Red Hat Enterprise Linux 6 Server (Kickstart)**로 이동하여 Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.5 (또는 이후 버전) 리포지토리를 선택합니다.



중요

6Server Kickstart 리포지토리를 선택하지 마십시오. 향후 Satellite 버전에서는 이러한 불필요한 Kickstart 리포지토리를 숨길 수 있습니다.

3. **RPM** 탭에서 **Red Hat Enterprise Linux Server** → **Red Hat Enterprise Linux 6 Server (RPMs)**로 이동하여 Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server 리포지토리를 선택합니다.
4. **RPM** 탭에서 **Red Hat Enterprise Linux Server** → **Red Hat Enterprise Linux 6 Server - Satellite Tools (RPMs)**로 이동하여 Red Hat Enterprise Linux 6 Server - Satellite Tools RPMs x86_64 리포지토리를 선택합니다.

3.4. 사용자 정의 제품 및 리포지터리 생성

다음 부분에서는 배포할 Puppet 모듈을 반영하도록 사용자 정의 제품을 생성하는 방법에 대해 설명합니다. 이 절차를 사용하여 Puppet 및 Yum 모두의 사용자 정의 리포지토리를 생성할 수 있습니다.

절차 3.4. 사용자 정의 제품을 생성하려면 다음을 수행합니다:

1. 콘텐츠 → 제품을 클릭하고 **새 제품**을 클릭합니다.
2. **이름** 필드에 **사용자 정의 제품**을 입력합니다. 레이블이 자동으로 생성됩니다. GPG 키, 동기와 계획 또는 설명을 입력할 필요가 없습니다.
3. **저장**을 클릭합니다.
4. 화면을 새로고침한 후 **리포지터리 생성**을 클릭합니다.
5. 이름에 **Puppet 모듈**을 입력합니다. 레이블이 자동으로 생성됩니다.
6. **유형** 필드에서 **puppet**을 선택합니다. **URL** 필드를 비워 둡니다.
7. **저장**을 클릭합니다.

다음 단계에서는 Puppet 모듈 리포지터리에 Puppet 모듈을 업로드합니다. Puppet Forge를 로컬로 미러링하기 위해 URL로 <https://forge.puppetlabs.com>을 사용할 수도 있습니다. 이는 Puppet Forge에

서의 모든 콘텐츠를 Satellite에서 사용할 수 있게 됨을 의미합니다. 하지만 이에는 2700 이상의 모듈을 다운로드해야 하며 사용 가능한 대역폭에 따라 오랜 시간이 걸릴 수 있습니다. 이 예제에서는 motd 모듈을 사용하고 있습니다. 이는 단순하고 다른 모듈에 종속성이 없기 때문입니다.

절차 3.5. 리포지터리에 Puppet 모듈을 업로드하려면 다음을 수행합니다:

1. <https://forge.puppetlabs.com/jeffmccune/motd>에서 motd puppet 모듈을 다운로드합니다. 다운로드한 파일에는 **.tar.gz** 확장자가 붙습니다.
2. 콘텐츠 → 제품을 클릭하고 이름 필드에서 **사용자 정의 제품**을 클릭합니다.
3. **리포지터리** 탭에서 **Puppet 모듈**을 클릭하여 Puppet 모듈 리포지터리를 변경합니다.
4. **Puppet 모듈 업로드** 섹션에서 **검색**을 클릭하여 다운로드한 motd 모듈로 이동합니다.
5. **업로드**를 클릭합니다.

3.5. 콘텐츠 동기화

다음 부분에서는 Red Hat 콘텐츠 전송 네트워크에서 Satellite로 리포지터리를 동기화하는 방법에 대해 설명합니다. 이 절차는 리포지터리 URL이 들어 있는 사용자 정의 리포지터리 (Yum 또는 Puppet) 동기화하는 방법에도 적용됩니다.

절차 3.6. Satellite에 리포지터리를 동기화하려면 다음을 수행합니다:

1. 콘텐츠 → 동기화 상태를 클릭하여 사용 가능한 제품 목록을 표시합니다.
2. **Red Hat Enterprise Linux Server → 6Server → x86_64**로 이동합니다.
3. 다음과 같은 제품을 선택합니다:
 - Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server.
 - Red Hat Enterprise Linux 6 Server - Satellite Tools RPMs x86_64.
4. **Red Hat Enterprise Linux Server → 6.6 → x86_64**로 이동하여 Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.6을 선택합니다.
5. **지금 동기화**를 클릭합니다.

리포지터리가 동기화될 때 까지 대기합니다. 사용 가능한 대역폭에 따라 몇 시간이 걸릴 수 있습니다.

4장. 콘텐츠 관리 및 승격

다음 부분에서는 애플리케이션 라이프 사이클 환경과 콘텐츠 뷰 설정 방법 및 Red Hat Enterprise Linux 리포지터리와 Puppet 모듈 추가 방법에 대해 설명합니다. 또한 콘텐츠 뷰 공개 방법과 활성화 생성 및 편집 방법에 대해서도 설명합니다.

이 장의 절차에서는 3장 *서브스크립션 가져오기 및 콘텐츠 동기화* 단계를 성공적으로 완료했다고 가정합니다.

4.1. 애플리케이션 라이프 사이클 환경 생성

애플리케이션 라이프 사이클 환경은 소프트웨어 개발 라이프 사이클 (SDLC)의 승격 경로에 있는 단계를 나타냅니다. 이 예제의 첫 번째 부분에서는 Dev;와 QA라는 두 개의 라이프 사이클 환경을 설정합니다. 두 번째 부분에서는 이러한 환경에서 사용하기 위한 콘텐츠 뷰를 생성합니다.

절차 4.1. 애플리케이션 라이프 사이클 환경을 생성하려면 다음을 실행합니다:

1. 콘텐츠 → 라이프 사이클 환경을 클릭하여 **라이프 사이클 환경 경로** 화면을 엽니다.
2. **새 환경 추가**를 클릭하여 **새 환경** 페이지를 표시합니다. **라이브러리**는 사용자 환경에서 사용할 수 있는 모든 콘텐츠의 기본이 됩니다.
3. **이름** 필드에 **Dev**를 입력합니다. 레이블에 동일한 이름이 자동 입력되지만 이는 필요에 따라 변경할 수 있습니다. 필요한 경우 사용자 환경에 대한 설명을 추가할 수 있습니다.
4. **저장**을 클릭하여 새 환경을 저장하고 이전 페이지로 돌아갑니다.
5. **새 환경 추가**를 다시 클릭한 후 **QA**라는 환경을 생성합니다.
6. **저장**을 클릭합니다.

4.2. 콘텐츠 뷰 생성

콘텐츠 뷰는 옵션 필터링과 함께 하나 이상의 리포지터리 (yum 또는 Puppet)에 들어 있는 관리되는 콘텐츠 선택 기능입니다. 이러한 필터는 포함 또는 제외될 수 있으며 라이프 사이클 관리를 위해 콘텐츠의 호스트 뷰를 조정할 수 있습니다. 필터를 사용하여 클라이언트 호스트에서 사용 가능하도록 콘텐츠를 사용자 정의할 수 있습니다.

절차 4.2. 콘텐츠 뷰를 생성하려면 다음을 수행합니다:

1. 콘텐츠 → 콘텐츠 뷰를 클릭하고 **새 뷰 생성**을 클릭합니다.
2. **이름** 필드에 RHEL6 x86_64를 입력합니다. 레이블에 자동으로 설정됩니다.
3. **복합 뷰** 체크박스가 선택되어 있는지 확인하고 **저장**을 클릭합니다.

콘텐츠 뷰가 성공적으로 생성되면 **리포지터리 선택** 화면이 자동으로 표시됩니다. 이 화면을 사용하여 선택된 리포지터리와 Puppet 모듈을 콘텐츠 뷰에 추가합니다.

4.3. RED HAT ENTERPRISE LINUX 리포지터리 추가

다음 단계에서는 이전 단계에서 생성된 콘텐츠 뷰에 Enterprise Linux 리포지터리를 추가하는 방법에 대해 설명합니다. 동일한 절차를 사용하여 Red Hat 또는 사용자 정의 리포지터리를 추가할 수 있습니다.

이 예제에서는 모든 콘텐츠가 공개되는 간단한 사례를 보여줍니다. 필터를 생성하여 공개된 콘텐츠 뷰에

포함 또는 제외하는 것과 같이 콘텐츠를 제어할 수 있습니다.

절차 4.3. Red Hat Enterprise Linux RPM 리포지토리를 추가하려면 다음을 수행합니다:

1. **콘텐츠 선택** 화면에 있는 **추가** 탭에서 다음의 각 리포지터리 옆에 있는 체크 박스를 선택합니다:
 - Red Hat Enterprise Linux 6 Server Kickstart x86_64 6Server
 - Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server
 - Red Hat Enterprise Linux 6 Server - Satellite Tools RPMs x86_64
2. **리포지터리 추가**를 클릭합니다. 페이지가 새로고침되면 **목록/삭제** 탭에서 콘텐츠 뷰의 리포지터리 목록을 확인할 수 있습니다.

4.4. PUPPET 모듈 추가

다음 절차에서는 [절차 3.5. “리포지터리에 Puppet 모듈을 업로드하려면 다음을 수행합니다:”](#)에 업로드된 motd Puppet 모듈을 추가하는 방법에 대해 설명합니다. Puppet 모듈은 사용자, 파일, 서비스와 같이 리소스를 관리하기 위해 사용할 수 있는 코드 및 데이터 자체에 포함된 번들입니다.

절차 4.4. Puppet 모듈을 추가하려면 다음을 수행합니다:

1. 아직 **콘텐츠 뷰** 페이지에 있는지 확인한 후 메인 메뉴에서 **콘텐츠** → **콘텐츠 뷰**를 클릭합니다.
2. **Puppet 모듈** 탭에서 **새 모듈 추가**를 클릭하여 사용 가능한 Puppet 모듈 목록을 표시합니다. **필터** 필드를 사용하여 필요한 모듈을 찾을 수 있습니다.
3. **버전 선택**을 클릭하여 motd 모듈을 선택합니다.
4. 추가하고자 하는 모듈 버전 옆에 있는 **버전 선택**을 클릭합니다.



참고

사용할 Puppet 모듈 버전을 선택할 때 "최근 사용"을 선택할 경우 콘텐츠 뷰의 새 버전이 공개될 때 마다 모듈의 최신 버전이 공개되는 뷰에 포함됩니다.

4.5. 콘텐츠 뷰 공개

모든 단계를 성공적으로 완료하면 Red Hat Satellite는 3 개의 Red Hat Enterprise Linux 리포지터리와 1 개의 Puppet 모듈이 포함된 하나의 콘텐츠 뷰를 갖습니다. 다음 단계에서는 이를 라이브러리에 게시합니다.

절차 4.5. 라이브러리에 콘텐츠 뷰를 공개하려면 다음을 수행합니다:

1. **콘텐츠** → **콘텐츠 뷰**를 클릭하여 **콘텐츠 뷰** 페이지를 표시합니다.
2. 공개하고자 하는 콘텐츠 뷰 이름을 클릭합니다.
3. **새 버전 공개**를 클릭하여 **새 버전 공개** 페이지를 표시합니다. 여기에서는 버전을 지정하여 콘텐츠 뷰에서의 모든 변경 사항을 반영하기 위한 코멘트를 입력할 수 있습니다.
4. **저장**을 클릭하여 라이브러리에 콘텐츠 뷰를 공개합니다. 나타나는 화면에서 공개 진행 상황을 모니터링할 수 있습니다.

5. 공개 프로세스가 완료되면 **승격**을 클릭하여 사용 가능한 승격 경로 목록 (Library -> Dev -> QA)을 표시합니다.
6. Dev 환경의 체크 박스를 선택하고 **버전 승격**을 클릭합니다.

4.6. 활성화키 생성 및 편집

컨텐츠 뷰를 성공적으로 공개한 후 활성화키를 생성해야 합니다. 나중 절차에서 활성화키는 호스트 그룹에 연결됩니다. 이를 통해 프로비저닝된 호스트가 Satellite에 등록되고 선택한 라이프 사이클 환경, 컨텐츠 뷰, 서브스크립션 등에 연결됩니다.

절차 4.6. 활성화키를 생성하려면 다음을 수행합니다:

1. 메인 메뉴에서 **컨텐츠** → **활성키**를 클릭하고 **새 활성화키**를 클릭합니다.
2. 이름 필드에 **ak-Reg_to_Dev**를 입력합니다.
3. 이 예제의 목적 상 **컨텐츠 호스트 제한** 체크박스를 삭제합니다.

이 필드를 사용하여 지정된 활성화키가 사용되는 횟 수를 제어할 수 있습니다. 예를 들어 활성화키를 수량 제한이 있는 서브스크립션에 연결할 경우 해당 수량을 초과하지 않도록 활성화키에 제한을 설정할 수 있습니다.

4. Dev 환경의 체크박스를 선택합니다.
5. **컨텐츠 뷰** 드롭다운 목록에서 RHEL 6 x86_64 컨텐츠 뷰를 선택하고 **저장**을 클릭합니다.

활성키를 생성한 후 키의 여러 매개 변수를 편집할 수 있습니다.

절차 4.7. 활성화키 매개변수를 편집하려면 다음을 수행합니다:

1. **활성키** 페이지에서 **서브스크립션** → **추가**를 클릭하여 사용 가능한 서브스크립션 목록을 표시합니다.
2. 이 활성화키를 사용하는 **각각의** 호스트에 첨부할 각 서브스크립션 옆에 있는 확인란을 선택합니다.
3. **선택 사항 추가**를 클릭합니다.

5장. 프로비저닝 설정 완료

컨텐츠 뷰 및 활성키를 성공적으로 생성한 후 호스트를 프로비저닝하기 위해 필요한 나머지 항목을 설정해야 합니다. 이에는 프로비저닝 템플릿 설정 및 호스트 그룹 생성이 포함됩니다.

5.1. 프로비저닝 템플릿 생성

다음 부분에서는 각각 동일한 설정을 갖고 있는 여러 호스트 프로비저닝에 사용할 수 있는 프로비저닝 템플릿을 설치하는 방법에 대해 설명합니다.

절차 5.1. 프로비저닝 템플릿을 생성하려면 다음을 수행합니다:

1. 메인 메뉴에서 **호스트** → **프로비저닝 템플릿**을 클릭합니다.
2. **이름** 칼럼에 있는 프로비저닝 템플릿 목록에서 **Satellite Kickstart 디폴트**를 클릭합니다. 이는 템플릿을 사용자 정의할 수 있는 설정 탭을 표시합니다.
3. **연결** 탭에서 적용 가능한 운영 체제 목록에서 RHEL Server 6.5를 선택한 **후제출**을 클릭합니다.
4. **이름** 칼럼에 있는 프로비저닝 템플릿 목록에서 **Kickstart 디폴트 PXELinux**를 클릭합니다.
5. **연결** 탭에서 적용 가능한 운영 체제 목록에서 RHEL Server 6.5를 선택한 **후제출**을 클릭합니다.
6. 메인 메뉴에서 **호스트** → **운영 체제**를 클릭한 후 RHEL Server 6.5를 클릭합니다. 이는 운영 체제를 사용자 정의할 수 있는 설정 탭을 표시합니다.
7. **파티션 테이블** 탭에서 **Kickstart 디폴트**를 선택합니다.
8. 설치 미디어 탭에서 Default_Organization/Library/Red_Hat_6_Server_Kickstart_x86_64_Server가 표시되어 선택되어 있는지 확인합니다.
9. 템플릿 탭에 있는 **PXELinux** 드롭 다운 목록에서 **Kickstart 디폴트 PXELinux**를 선택합니다.
10. **프로비저닝** 드롭 다운 목록에서 **Satellite Kickstart 디폴트**를 선택하고 **제출**을 클릭합니다.

5.2. 호스트 그룹 생성

다음 부분에서는 호스트 그룹을 생성 및 설정하는 방법에 대해 설명합니다. 호스트 그룹은 각 호스트에 동일한 등록 정보를 지정할 필요 없이 여러 호스트를 프로비저닝하여 다시 사용할 수 있는 효과적인 호스트 템플릿입니다.

절차 5.2. 호스트 그룹을 생성하려면 다음을 실행합니다:

1. 메인 메뉴에서 **설정** → **호스트 그룹**을 클릭한 후 새 **호스트 그룹**을 클릭합니다.
2. **호스트 그룹** 탭에서 다음 값을 입력합니다:
 - **이름**: RHEL6Server-x86_64
 - **라이프 사이클 환경**: Default_Organization/DEV
 - **컨텐츠 뷰**: RHEL_6_x86_64



참고

다음 필드는 **라이프 사이클 환경** 필드에 값을 입력한 후에만 표시됩니다.

- **컨텐츠 소스:** Capsule (Satellite Server가 될 수 있음)의 FQDN입니다.
 - **Puppet CA:** Satellite의 FQDN.
 - **Puppet 마스터:** Satellite의 FQDN.
3. **Puppet 클래스** 탭에 있는 사용 가능한 클래스 목록에서 motd puppet 모듈을 선택합니다.
4. **네트워크** 탭에서 다음 값을 선택합니다:
- **도메인:** example.org
 - **서브넷:** Provisioning_Net
 - **영역:** 이 예제의 목적 상 이 필드를 비워둡니다. IPA와 같은 영역 관리를 설정할 경우 여기서 해당하는 영역을 선택합니다.
5. **운영 체제** 탭에서 다음 값을 선택합니다:
- **아키텍처:** x86_64
 - **운영 체제:** RHEL Server 6.5
 - **미디어:** Default_Organization/Library Red Hat Server 6.5 x86_64. 이 예제에서는 자동으로 설정됩니다.



참고

필드가 자동으로 설정되지 않을 경우 조직 및 위치 연결이 올바르게 설정되지 않았음을 의미합니다. [절차 2.5. "설치 미디어의 조직 및 위치에 연결하려면 다음을 수행합니다:"](#)에서 참조하십시오.

- **파티션 테이블:** Kickstart default
 - **Root 암호:** changeme
6. **위치** 탭에서 Default_location을 선택합니다.
7. **조직** 탭에서 Default_Organization을 선택합니다.
8. **활성기** 탭에서 **활성기** 목록의 ak-Reg_To_Dev를 선택합니다.
9. **제출**을 클릭합니다.

6장. 호스트 프로비저닝

다음 부분에서는 Red Hat Satellite Server를 사용하여 새 호스트를 프로비저닝하는 방법에 대해 설명합니다. 이전 부분에서는 프로비저닝에 필요한 모든 사항을 설치 및 설정하는 방법을 살펴보았습니다. 호스트 프로비저닝을 실행하기 전 이전 장의 모든 작업을 완료했는지 확인하십시오.

Satellite는 호스트 프로비저닝에서 DHCP 및 TFTP 서비스를 필요로 하는 PXE 부팅과 PXE 서비스를 사용할 수 없는 경우 호스트 프로비저닝을 제공하는 부팅 디스크 프로비저닝이라는 두 가지 방식을 제공합니다.

6.1. PXE를 사용하여 호스트 프로비저닝

다음 부분에서는 Satellite 6 서버에서 호스트를 프로비저닝하는 방법에 대해 설명합니다.

절차 6.1. 호스트를 프로비저닝하려면 다음을 실행합니다:

1. 메인 메뉴에서 **호스트** → **새 호스트**를 클릭하여 **새 호스트** 페이지를 엽니다.
2. **호스트** 탭에서 다음 값을 입력합니다:
 - **이름**: 호스트에 적합한 이름을 선택합니다. 예: `host1.example.org`
 - **호스트 그룹**: RHEL6Server-x86_64를 선택합니다



참고

새 호스트가 호스트 그룹에 설정된 기본값을 상속합니다. 이는 이러한 값을 다시 입력하지 않고 호스트를 즉시 빌드할 수 있음을 의미합니다.

- **컨텐츠 소스**: Satellite의 \$FQDN입니다. 이는 호스트 그룹에 따라 자동으로 선택됩니다.
3. **네트워크** 탭에서 다음 값을 입력합니다:
 - **MAC 주소**: 새 호스트의 MAC 주소입니다. Satellite 서버는 이 값을 사용하여 DHCP 주소를 마련해 둡니다. 올바른 값을 입력하십시오.
 - **서브넷**: `Provisioning_Net` 이 값은 자동으로 설정됩니다.
 - **IP 주소**: 이 값은 자동으로 설정됩니다.

Puppet 클래스, 운영 체제, 매개 변수, 또는 **추가 정보** 탭에 어떠한 사항도 변경하지 마십시오.

4. **제출**을 클릭합니다.
5. 호스트 (물리적 호스트 또는 가상 머신 중 하나)의 전원을 켜면 PXE-boot가 실행되어 설치 프로세스를 시작합니다.

보다 자세한 내용은 <https://access.redhat.com/articles/1169613>에서 참조하십시오.

6.2. 부팅 디스크를 사용하여 호스트 프로비저닝

Satellite 네트워크 프로비저닝 모델은 일반적으로 DHCP 및 TFTP 서비스를 필요로 하는 PXE를 기반으로 합니다. 모든 Satellite 배치에서 이러한 서비스를 사용할 수 있는 것이 아니기 때문에 부팅 디스크 프로비저닝 기능은 이러한 배치에서 프로비저닝을 활성화하기 위해 호스트별, 전체 호스트, 일반 부팅 디스크 이미지 유형을 제공합니다.

각각의 부팅 디스크 이미지 유형에는 제각기 장점을 가지고 있지만 이 모두는 네트워크 인프라의 제어 없는 환경에 대해 고안되어 있습니다. 결과적으로 DHCP 예약 또는 TFTP 설정이 필요하지 않습니다.

부팅 이미지는 하이브리드 ISO 이미지 (ISO 파일 또는 USB 디스크로 사용 가능)로 작성되며 물리적 미디어 또는 가상 디스크나 CD 중 하나에서 부팅할 수 있습니다.

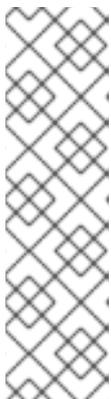
표 6.1. 부팅 이미지 유형의 특징 비교

유형	일반	DHCP 필요	DHCP 예약	사전 등록 호스트	운영 체제 별
호스트별 이미지	없음	없음	없음	있음	없음
전체 호스트 이미지	없음	있음	없음	있음	있음
일반 이미지	있음	있음	없음	있음	없음

6.2.1. 전제 조건

Satellite 부팅 디스크의 모든 전제 조건 패키지는 일반적으로 기본으로 설치되어 있습니다. 다음으로 진행하기 전 다음 조건을 확인합니다:

- 사용하는 이미지 유형과 상관 없이 호스트는 이미지에서 부팅하기 전 Satellite에 등록되어 있어야 합니다. 호스트가 빌드 모드에 있을 경우 올바른 프로비저닝 템플릿을 제공하기 위해 호스트는 MAC 또는 IP 주소에 따라 구분됩니다.
- 호스트별 이미지의 경우 호스트 IP 주소 및 서브넷이 이미 지정되어 있으며 서브넷의 게이트웨이, 서브넷 마스크, DNS 확인자가 올바르게 설정되어 있는지 확인합니다. **인프라** → **서브넷**으로 이동하여 이 값을 설정합니다.
- 관리자가 아닌 사용자에게 이미지에 대한 액세스를 허용하려면 사용자에게 "Boot disk access" 역할을 추가하거나 기존 역할에 "download_bootdisk" 권한을 추가합니다.
- 적절한 iPXE 프로비저닝 템플릿을 사용할 수 있는지 확인합니다. Red Hat Enterprise Linux는 "Kickstart default iPXE" 템플릿을 사용합니다.



참고

gPXE 이미지는 현재 지원되지 않습니다.

호스트 및 일반 이미지 유형은 iPXE 기술을 기반으로 하며 PXELinux에서 다른 하드웨어 드라이버 세트를 지원합니다. 지원되는 하드웨어 목록은 http://ipxe.org/appnote/hardware_drivers에서 참조하십시오.

iPXE 관련 문제가 발생할 경우 전체 호스트 이미지에는 내장 커널 및 RAM 디스크가 포함되어 PXE 지원되지 않는 카드를 포함한 모든 종류의 네트워크 카드를 읽어들일 수 있습니다.

- 적절한 표준 템플릿이 사용 가능한 지 확인합니다. Red Hat Enterprise Linux는 프로비저닝 템플릿으로 "Kickstart RHEL default"를 사용합니다.



중요

기본 Satellite 키스타트 프로비저닝 템플릿을 사용하지 않는 경우 사용하는 템플릿이 운영 체제 설정에 필요한 고정 IP 정보를 제공하고 있는지 확인합니다. 키스타트 파일의 경우 다음 설정을 사용할 수 있습니다:

```
network --bootproto <%= @static ? "static" : "dhcp" %> --
hostname <%= @host %> <%= " --ip=#{@host.ip} --netmask=#
{@host.subnet.mask} --gateway=#{@host.subnet.gateway} --
nameserver=#{@host.subnet.dns_primary},#
{@host.subnet.dns_secondary}" if @static %>
```

6.2.2. 부팅 디스크 이미지 생성

다음 부분에서는 호스트별, 전체 호스트, 일반 부팅 디스크 이미지를 생성하는 방법에 대해 설명합니다. web UI 또는 명령행을 사용하여 이미지를 생성할 수 있습니다. 두 가지 방법 모두에 대해 설명합니다.



참고

명령행을 사용하여 이미지를 생성하려면 ruby193-rubygem-foreman_bootdisk 패키지가 설치되어 있어야 합니다. 이는 일반적으로 기본값으로 설치되어 있습니다.

6.2.2.1. 호스트별 이미지 생성

고정 네트워크로 호스트별 이미지를 생성하기 위해 Satellite에 있는 호스트 및 서브넷 데이터를 사용할 수 있습니다. 동작은 동적입니다. Satellite에서 이미지 연쇄 로딩이 진행되어 결과적으로 현재 운영 체제와 빌드 상태는 이미지에 저장되는 대신 Satellite에 의해 제공됩니다.

절차 6.2. Web UI를 사용하여 호스트별 이미지를 생성하려면 다음을 수행합니다:

1. **호스트** → **전체 호스트**로 이동하여 해당 호스트 이름을 클릭합니다.
2. **부팅 디스크**를 클릭하고 **호스트 호스트이름 이미지**를 클릭합니다.

hammer CLI 도구를 사용하여 호스트별 이미지를 생성하려면 다음 명령을 실행합니다:

```
# hammer bootdisk host --host client.example.com
```

Satellite server 상의 명령행에서 호스트별 이미지를 생성하려면 다음 명령을 실행합니다:

```
# foreman-rake bootdisk:generate:host NAME=client.example.com OUTPUT=/tmp
```

OUTPUT 값을 디렉토리나 파일 중 하나로 적절한 대상 경로를 설정합니다. **foreman** 사용자는 지정한 대상에 대해 쓰기 권한을 갖고 있어야 합니다.

6.2.2.2. 전체 호스트 이미지 생성

전체 호스트 이미지는 호스트별 이미지와 유사하지만 Satellite에서 연쇄 로딩하는 대신 이러한 이미지에 초기 운영 체제 부트 로더가 포함되어 있습니다. 이는 연쇄 로딩에 실패한 호스트의 경우에 유용하지만 호스트 운영 체제, 부트로더 또는 템플릿 변경, 빌드 토큰이 필요하며 만료된 경우 이미지가 오래된 것일 수 있다는 단점이 있습니다.

절차 6.3. Web UI를 사용하여 전체 호스트 이미지를 생성하려면 다음을 수행합니다:

1. **호스트** → **전체 호스트**로 이동하여 해당 호스트 이름을 클릭합니다.
2. **부팅 디스크**를 클릭하고 **전체 호스트 호스트 이름 이미지**를 클릭합니다.

운영 체제 부트 로더를 다운로드하는 프로세스가 상당히 클 수 있기 때문에 전체 호스트 이미지를 생성하는 데는 오랜 시간이 소요될 수 있습니다.

hammer CLI 도구를 사용하여 전체 호스트 이미지를 생성하려면 다음 명령을 실행합니다:

```
# hammer bootdisk host --host client.example.com --full true
```

Satellite server 상의 명령행에서 전체 호스트 이미지를 생성하려면 다음 명령을 실행합니다:

```
# foreman-rake bootdisk:generate:full_host NAME=client.example.com
```

6.2.2.3. 일반 이미지 생성

일반 이미지는 등록된 모든 호스트가 사용할 수 있는 단일 ISO 파일을 제공합니다. IP 주소 정보는 이러한 이미지 내에 저장할 수 없습니다. 즉 네트워크는 DHCP 풀을 지정해야 합니다. 일반 이미지를 사용하여 호스트를 부팅하면 DHCP에 의해 호스트에 할당된 MAC 주소 또는 IP와 일치하는 등록된 호스트의 템플릿에 해당하는 Satellite에 연결합니다.

운영 체제 iPXE 템플릿이 설정된 방법에 따라 DHCP 할당 또는 고정 IP 주소를 사용하여 설치를 계속 진행할 수 있습니다. 킥스타트 파일을 사용하여 추가 네트워크 설정 옵션을 지정할 수 있습니다.

절차 6.4. Web UI에서 일반 이미지를 생성하려면 다음을 수행합니다:

1. **호스트** → **전체 호스트**로 이동하여 해당 호스트 이름을 클릭합니다.
2. **부팅 디스크**를 클릭하고 **일반 이미지**를 클릭합니다.

hammer CLI 도구를 사용하여 일반 이미지를 생성하려면 다음 명령을 실행합니다:

```
# hammer bootdisk generic
```

Satellite server 상의 명령행에서 일반 이미지를 생성하려면 다음 명령을 실행합니다:

```
# foreman-rake bootdisk:generate:generic
```

6.2.2.4. USB 이미지 생성

ISO 파일을 생성할 때 마다 이는 **isohybrid** 명령을 통해 전달됩니다. 즉 결과 파일은 디스크로 부팅 가능하며 USB 장치에 복사하는데 적합합니다.

ISO 파일을 USB 장치에 복사하려면 다음 명령을 실행합니다. 장치 이름 및 입력 파일이 사용자 환경에 적합한 지 확인합니다:

```
# dd if=fqdn.iso of=/dev/sdX
```

6.3. 고정 IP 주소를 사용하여 호스트 프로비저닝

Red Hat Satellite 6에서는 모든 시스템이 DHCP 용으로 설정된 것으로 예상합니다. 이는 지정된 MAC 주소의 DHCP 레코드가 예약되기 때문입니다. 또한 사용자 지정 프로비저닝 템플릿, 호스트 매개 변수, 서브넷 정보를 기반하는 것중 하나의 방법을 사용하여 고정 IP 주소로 호스트를 프로비저닝할 수 있습니다.

6.3.1. 고정 IP 주소를 할당하기 위해 사용자 지정 템플릿 사용

모든 프로비저닝된 호스트에 대해 고정 IP 지원을 제공하는 사용자 지정 프로비저닝 템플릿을 만들 수 있습니다. PXE 설정 템플릿을 복사하고 "RHEL 7.1 static"과 같이 다른 운영 체제에 연결할 수 있습니다. 즉 이 템플릿을 사용하여 시스템 킥스타트를 실행하면 고정 IP를 받게 됩니다. Satellite 6.0 및 6.1에서는 기존 "새 호스트 생성" 워크 플로우에서 동적 또는 고정 IP 설정 선택을 지원하지 않기 때문에 현재 이러한 방법이 필요합니다.

이 방법에서는 고정 네트워크를 활성화하기 위해 PXE 템플릿을 편집해야 합니다. 다음의 예에서 설명된 대로 PXE 템플릿을 편집하고 `&static=yes`를 `foreman_url('provision')`의 각 인스턴스 끝에 추가합니다.

절차 6.5. PXE 템플릿을 편집하려면 다음을 수행합니다:

1. **호스트** → 모든 호스트로 이동하여 편집하고자 하는 템플릿의 호스트 이름을 클릭합니다.
2. **템플릿** 탭을 클릭하여 사용 가능한 템플릿 유형 목록을 나열합니다.
3. **PXELinux 템플릿** 유형의 **편집**을 클릭합니다. 템플릿이 템플릿 편집기에 표시됩니다.
4. 필요한 사항을 변경한 후 **제출**을 클릭합니다. PXE 템플릿을 업데이트하는 방법에 대한 예는 [예 6.1. "고정 IP 주소를 할당하기 위해 PXE 템플릿 편집"](#)에서 참조하십시오.

예 6.1. 고정 IP 주소를 할당하기 위해 PXE 템플릿 편집

```
default linux
label linux
kernel <%= @kernel %>
<% if @host.operatingsystem.name == 'Fedora' and
@host.operatingsystem.major.to_i > 16 -%>
append initrd=<%= @initrd %> ks=<%=
foreman_url('provision')%>&static=yes ks.device=bootif network
ks.sendmac
<% elsif @host.operatingsystem.name != 'Fedora' and
@host.operatingsystem.major.to_i >= 7 -%>
append initrd=<%= @initrd %> ks=<%=
foreman_url('provision')%>&static=yes network ks.sendmac
<% else -%>
append initrd=<%= @initrd %> ks=<%=
foreman_url('provision')%>&static=yes ksdevice=bootif network kssendmac
<% end -%>
IPAPPEND 2
```

6.3.2. 고정 IP 주소를 할당하기 위해 호스트 매개 변수 사용

호스트에 고정 IP 주소를 설정하기 위해 호스트 매개 변수를 사용할 수 있습니다. 새 호스트를 만들려면 **호스트** → **새 호스트**로 이동한 후 **매개 변수** 탭에 필요한 매개 변수를 입력합니다.

[예 6.2. "고정 IP 주소를 할당하기 위해 호스트 매개 변수 사용"](#)에서는 고정 IP 주소를 설정하기 위해 호스트 매개 변수와 함께 **if** 문을 사용하는 방법을 보여주고 있습니다.

예 6.2. 고정 IP 주소를 할당하기 위해 호스트 매개 변수 사용

```
<% if @host.params['my_param_deploy_static'] == "true" -%>
  network --bootproto <%= "static --ip=#{@host.ip} --netmask=#
{@host.subnet.mask} --gateway=#{@host.subnet.gateway} --nameserver=#
[#{@host.subnet.dns_primary,@host.subnet.dns_secondary}].reject{|n|
n.blank?}.join(',')" %> --hostname <%= @host %>
<% else -%>
  network --bootproto 'dhcp' --hostname <%= @host %>
<% end -%>
```

보다 자세한 내용은 [Red Hat Satellite 6 사용자 가이드](#)에서 참조하십시오.

6.3.3. 고정 IP 주소를 설정하기 위해 서브넷 사용

호스트의 서브넷에 따라 고정 IP 주소를 사용하여 호스트를 프로비저닝하도록 Red Hat Satellite를 설정할 수 있습니다. 서브넷을 설정할 때 DHCP 또는 정적 부팅 모드 중 하나를 지정할 수 있습니다. Red Hat Enterprise Linux 설치 프로그램 (Anaconda)은 이 값을 사용하여 DHCP 플에서의 주소 또는 고정 IP 주소를 할당할 지에 대한 여부를 결정합니다. 이 서브넷에서 제공되는 모든 호스트가 고정 IP 주소를 받게 하려면 "Static"을 지정합니다.

절차 6.6. 기본값으로 고정 부팅 모드를 지정하려면 다음을 수행합니다:

1. **인프라** → **서브넷**으로 이동하여 설정하고자 하는 서브넷 이름을 클릭합니다. 새 서브넷을 만들려면 [절차 2.4. "기본 서브넷을 선택하려면 다음을 실행합니다."](#)에 있는 단계를 수행합니다.
2. **서브넷** 탭에 있는 **부팅 모드** 드롭 다운 목록에서 **Static**을 선택하고 **제출**을 클릭합니다.

새 호스트를 만들고 이를 서브넷에 할당할 때 마다 이는 기본값으로 고정 IP 주소를 사용합니다.

부록 A. 용어집

다음은 이 문서 전체에서 사용되는 용어입니다. 이러한 용어를 미리 알아두시면 Red Hat Satellite 6를 이해하는데 도움이 됩니다.

활성키

등록시 동작을 제어하기 위해 킥스타트 파일에 사용되는 등록 토큰입니다. 이는 Red Hat Satellite 5의 활성키와 비슷하지만 등록 후 Puppet이 패키지 및 설정 관리를 제어하기 때문에 이에 필요한 기능을 제공합니다.

애플리케이션 라이프 사이클 환경 (Application Life Cycle Environment)

*애플리케이션 라이프 사이클 환경*은 소프트웨어 개발 라이프 사이클 (SDLC)의 프로모션 경로에 있는 단계를 나타냅니다. 프로모션 경로는 개발 경로로도 알려져 있습니다. 패키지와 Puppet 모듈과 같은 콘텐츠는 콘텐츠 뷰를 공개 또는 승격하여 라이프 사이클 환경 사이를 이동합니다. 모든 콘텐츠 뷰에는 버전이 있어 일반적인 프로모션 경로를 통해 특정 버전을 승격할 수 있습니다. 예를 들어 개발 (development)에서 테스트 (test) 또는 프로덕션 (production)으로 전환할 수 있습니다. Red Hat Satellite 5에서는 채널 복제가 이러한 개념을 실행하고 있습니다.

할당 (Attach)

RPM 콘텐츠에 대한 액세스를 제공하는 호스트에 서브스크립션을 할당하는 과정입니다.

Capsule

*Capsule*은 Red Hat Satellite 6 배포에 사용할 수 있는 추가 서버로 로컬화된 다른 서비스 (Puppet Master, DHCP, DNS, TFTP 등) 이외에 콘텐츠 연합 및 배포를 용이하게 합니다.

카탈로그

*카탈로그*는 하나의 특정 컴퓨터에 필요한 시스템 상태를 설명하는 문서입니다. 이에 관리해야 하는 모든 리소스 및 이러한 리소스 간의 모든 종속성이 나열되어 있습니다.

컴퓨팅 프로파일 (Compute Profile)

*컴퓨팅 프로파일*은 컴퓨터 리소스에 있는 새 가상 머신의 기본 속성을 지정합니다.

컴퓨팅 리소스

*컴퓨팅 리소스*는 가상 또는 클라우드 인프라로 Red Hat Satellite 6는 호스트 및 시스템 배포에 사용합니다. 예를 들어 Red Hat Enterprise Virtualization Manager, OpenStack, EC2, VMWare가 있습니다.

컨텐츠

*컨텐츠*에는 소프트웨어 패키지 (RPM 파일) 및 Puppet 모듈이 포함되어 있습니다. 이는 라이브러리에 동기화된 후 콘텐츠 뷰를 사용하여 라이프 사이클 환경으로 승격되어 호스트가 이를 사용할 수 있게 됩니다.

CDN (Content Delivery Network)

*CDN (Content Delivery Network)*은 지리적으로 같은 위치에 Red Hat 콘텐츠를 전송하기 위해 사용되는 메커니즘입니다. 예를 들어, 유럽에 있는 Satellite로 콘텐츠를 동기화할 경우 유럽에 있는 소스에서 콘텐츠를 가져옵니다.

컨텐츠 호스트

*컨텐츠 호스트*는 콘텐츠 및 서브스크립션과 관련된 작업을 관리하는 호스트의 일부입니다.

컨텐츠 뷰

*컨텐츠 뷰*는 제품, 패키지, Puppet 모듈을 지능형 필터링 및 스냅샷 생성과 같은 기능과 결합된 컨텐츠 정의입니다. 컨텐츠 뷰는 Red Hat Satellite 5에서 채널과 복제의 조합을 재정의한 것입니다.

외부 노드 분류기

*외부 노드 분류기*는 호스트 설정 시 사용할 Puppet 마스터 용 추가 데이터를 제공하는 Puppet 구조입니다. Red Hat Satellite 6는 Satellite 배포에서 Puppet 마스터로 외부 노드 분류기 역할을 합니다.

Facter

*Facter*는 실행되고 있는 시스템에 대한 정보를 제공하는 프로그램입니다. 예를 들어 Facter는 총 메모리, 운영 버전, 아키텍처 등을 보고할 수 있습니다. Puppet 모듈은 Facter에 의해 수집된 호스트 데이터에 기반하여 특정 설정을 사용합니다.

Hammer

*Hammer*는 Red Hat Satellite 6 용 명령행 도구입니다. Hammer를 사용하여 표준 CLI로나 스크립트 또는 대화식 셸로 사용하도록 Red Hat Satellite 6를 관리할 수 있습니다.

Hiera

Hiera는 설정 데이터의 키/값 검색 도구로 puppet 매니페스트에서 사이트별 데이터를 보관할 수 있습니다.

호스트

호스트는 Red Hat Satellite 6가 관리하는 물리적 또는 가상 시스템입니다.

호스트 컬렉션 (Host Collection)

*호스트 컬렉션*은 Satellite 5 시스템 그룹에 해당하는 것으로 사용자가 정의하는 하나 이상의 호스트 그룹을 말합니다.

호스트 그룹 (Host Group)

*호스트 그룹*은 호스트를 빌드하기 위한 템플릿입니다. 이에는 컨텐츠 뷰 (사용 가능한 RPM 파일 및 Puppet 모듈을 정의) 및 적용할 Puppet 클래스 (궁극적으로 소프트웨어 및 설정 지정)가 포함되어 있습니다.

위치

*위치*는 실제 위치를 나타내는 기본 설정 모음입니다. 이는 중복될 수 있기 때문에 위치의 계층 모음을 설정할 수 있습니다. 예를 들어 "중동"을 기본값으로 설정하여 "텔아бив"로 다시 지정하고 "동쪽 데이터 센터"로 상세 지정한 후 마지막으로 "랙 22"로 설정할 수 있습니다.

라이브러리

*라이브러리*에는 최신 동기화된 버전을 포함하여 사용자가 배포할 소프트웨어의 모든 버전이 포함되어 있습니다. ITIL (Information Technology Infrastructure Library) [1] 조직 또는 부서의 경우 이는 확정판 미디어 라이브러리 [2]에 해당합니다 (이전에는 확정 버전 소프트웨어 라이브러리라고 부름).

매니페스트

*매니페스트*는 서브스크립션을 고객 포털에서 Red Hat Satellite 6로 전송합니다. 이는 Red Hat Satellite 5에서 사용된 인증서와 비슷합니다.

인증서 및 서브스크립션 유형에 대한 보다 자세한 내용은 다음에서 참조하십시오:

- [RHN Classic, Red Hat Satellite 및 채널 인타이틀먼트](#)^[3]
- [Satellite 인증 구조 \(클래식 스타일 인증서\)](#)^[4]

조직

조직은 Satellite 6 배포에서 시스템, 콘텐츠, 다른 기능으로 구성된 단일 모음입니다.

제품

콘텐츠 리포지터리 모음입니다. 제품은 Red Hat 제품과 소프트웨어 및 설정 콘텐츠로 구성된 새로 생성된 제품이 될 수 있습니다.

승격

개발에서 QA 또는 프로덕션으로 전환하는 것과 같이 소프트웨어 및 설정 콘텐츠로 구성된 콘텐츠 뷰를 하나의 애플리케이션 라이프 사이클 환경에서 다른 애플리케이션 라이프 사이클 환경으로 마이그레이션하는 작업을 말합니다.

프로비저닝 템플릿

*프로비저닝 템플릿*은 키스타트 파일, 스톱, 기타 프로비저닝 작업을 위한 사용자 정의 템플릿입니다. Satellite 6에서 이 템플릿은 Red Hat Satellite 5의 키스타트 프로파일과 cobbler 스톱과 유사한 기능을 제공합니다.

Pulp 노드

Pulp 노드는 콘텐츠를 미러링하는 Capsule Server 구성 요소입니다. 이는 Red Hat Satellite 5 Proxy와 유사합니다. 주요 차이점은 콘텐츠가 호스트에 의해 사용되기 전에 Pulp 노드에서 콘텐츠를 준비할 수 있다는 것입니다.

Puppet 에이전트

*Puppet 에이전트*는 호스트에서 실행되는 에이전트로 호스트에 설정 변경 사항을 적용합니다.

Puppet 마스터

*Puppet 마스터*는 Puppet 에이전트에 의한 실행을 위해 호스트에 Puppet 매니페스트를 제공하는 Capsule Server 구성 요소입니다.

Puppet 모듈

*Puppet 모듈*은 사용자, 파일, 서비스와 같은 리소스를 관리하는데 사용할 수 있는 코드 및 데이터 자체에 포함된 번들입니다.

리포지터리

*리포지터리*는 콘텐츠 모음에 대한 스토리지를 제공합니다. 예를 들어 YUM 리포지터리 또는 Puppet 리포지터리 등이 있습니다.

역할

*역할*은 호스트와 같은 리소스 모음에 적용되는 권한 모음을 지정합니다.

스마트 프록시

*스마트 프록시*는 DNS 또는 DHCP와 같은 외부 서비스와 통합할 수 있는 Capsule Server 구성 요소입니다.

스마트 변수

*스마트 변수*는 Puppet 클래스 동작을 제어하는 설정입니다. 이는 호스트, 호스트 그룹, 조직 또는 위치에 설정할 수 있습니다.

SOE (Standard Operating Environment)

SOE (*Standard Operating Environment*)는 애플리케이션이 배포되는 운영 체제의 제어된 버전입니다.

서브스크립션

*서브스크립션*은 Red Hat에서 콘텐츠 및 서비스를 받는 수단입니다.

동기화

*동기화*는 외부 리소스에서 Red Hat Satellite 6 라이브러리로 콘텐츠를 미러링하는 것을 말합니다.

동기화 계획

*동기화 계획*은 콘텐츠 동기화의 스케줄된 실행을 제공합니다.

사용자 그룹

*사용자 그룹*은 사용자 모음에 할당할 수 있는 역할 모음입니다. 이는 Red Hat Satellite 5의 역할과 비슷합니다.

사용자

사용자는 Red Hat Satellite를 사용할 수 있도록 등록된 모든 사용자를 말합니다. 인증 및 권한 부여는 내장된 로직, 외부 LDAP 리소스 또는 Kerberos를 통해 가능합니다.

[1] http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library

[2] http://en.wikipedia.org/wiki/Definitive_Media_Library

[3] https://access.redhat.com/site/documentation/en-US/Red_Hat_Subscription_Management/1/html/MigratingRHN/sat-certs.html

[4] https://access.redhat.com/site/documentation/en-US/Red_Hat_Subscription_Management/1/html/Subscription_Concepts_and_Workflows/index.html#subscription-legacy

부록 B. 고친 과정

고침 1.2-8 Removing beta watermark.	Wed August 12 2015	Hayley Hudgeons
고침 1.2-7 한국어 번역 완료	Tue Jun 23 2015	Eun-Ju Kim
고침 1.2-6.3 XML 소스 1.2-6 버전과 번역 파일을 동기화	Tue Jun 23 2015	Eun-Ju Kim
고침 1.2-6.2 한국어 번역 완료	Wed Jun 17 2015	Eun-Ju Kim
고침 1.2-6.1 XML 소스 1.2-6 버전과 번역 파일을 동기화	Wed Jun 17 2015	Eun-Ju Kim
고침 1.2-6 업데이트된 제품 버전을 사용하여 다시 빌드함.	Fri May 15 2015	David O'Brien
고침 1.2-5 기술 리뷰를 위해 빌드함.	Mon May 11 2015	David O'Brien
고침 1.2-4 Satellite 6.1에서 고정 IP를 사용하는 방법을 문서화함.	Thu Apr 16 2015	David O'Brien
고침 1.2-3 브랜드를 업데이트함. 머리말 제거함.	Wed Apr 8 2015	Megan Lewis
고침 1.2-2 새 포털 디자인에 맞게 매니페스트를 만드는 절차를 업데이트함.	Fri Apr 3 2015	David O'Brien
고침 1.2-1 BZ 1207429: "호스트 그룹 생성" 절차의 오류를 수정함. 콘텐츠 소스는 Capsule을 참조해야 함. "PTR 영역 파일"에 대한 참조를 명확히함. BZ 1203878: "RH Common" 리포지터리 이름을 "Satellite Tools"에 업데이트함. BZ 1168240: "도메인" 대 "영역"에 대한 부분을 명확히함. BZ 1176466: katello-installer 명령의 필요한 옵션을 명확히함. 호스트 그룹을 만드는 방법에 대한 절차를 명확히함. BZ 1183657: "puppet 모듈" 및 "카탈로그"를 용어집에 추가함. "버그 리포트" 링크에 있는 엔티티 수정.	Tue Mar 31 2015	David O'Brien
고침 1.2-0 BZ 1184671: katello-installer 옵션의 오류 수정.	Thu Feb 22 2015	David O'Brien
고침 1.1-1 올바른 버전을 참조하기 위해 Satellite의 호스트를 생성하는 섹션을 업데이트함. "사용자 정의 제품 및 리포지터리 생성" 부분에 있는 관련 링크를 수정.	Mon Dec 22 2014	David O'Brien
고침 1.1-0 상세 정보를 참조하기 위해 매니페스트 생성 부분을 업데이트.	Wed Dec 10 2014	David O'Brien
고침 1.0-0 QA 리뷰에서 업데이트 구현. 첫 번째 공개 릴리즈.	Wed Nov 26 2014	David O'Brien

고침 0.1-1

Thu Nov 20 2014

David O'Brien

- 용어집의 용어를 추가.
- 컨텐츠 관리 및 승격에 있는 컨텐츠 뷰 생성 부분에 컨텐츠 뷰 정의를 추가.
- 참조된 절차의 xref 추가.
- 관리자 사용자 이름 및 암호를 포함하도록 초기 설정 부분이 업데이트됨.
- 호스트 프로비저닝 부분을 업데이트함.
- 요약 추가.
- 프로비저닝 템플릿 설정 부분의 초안 작성.
- 컨텐츠 관리 및 승격 부분의 초안 작성.
- 서브스크립션 가져오기 및 컨텐츠 동기화에 대한 부분을 개정.

고침 0.1-0

Fri Oct 17 2014

David O'Brien

- "서비스 구성" 장의 초안을 추가.
- "프로비저닝 개요" 장의 초안을 추가.
- 기본 섹션으로 TOC를 추가.
- Publican에 의한 최초 문서 생성.