



Ansible을 사용한 NetApp SAP Landscape Management 통합

NetApp solutions for SAP

NetApp
October 30, 2025

목차

Ansible을 사용한 NetApp SAP Landscape Management 통합	1
TR-4953: Ansible을 사용한 NetApp SAP 환경 관리 통합	1
SAP 시스템 클론, 복제 및 업데이트 시나리오	1
시스템 업데이트, 복사 및 클론 복제 사용 사례	2
논리적 손상을 처리합니다	3
재해 복구 테스트	4
Ansible을 사용하여 NetApp SAP LaMa 통합	4
구현 예	5
검증된 구성 및 제한	6
랩 설정	6
SAP LaMa 구성	7
SAP LaMa가 워크플로우-클론 시스템을 프로비저닝합니다	10
SAP LaMa에서 워크플로우 프로비저닝 해제 - 시스템 폐기	18
SAP LaMa가 워크플로우-복사 시스템을 프로비저닝합니다	21
SAP LaMa 프로비저닝 워크플로우 - 시스템 업데이트	25
공급자 스크립트 구성 및 Ansible 플레이북	27
provider 구성 파일 netapp_clone.conf	28
공급자 스크립트 netapp_clone.sh	28
Ansible 플레이북 NetApp_LaMa_CloneVolumes.yml	36
Ansible 플레이북 NetApp_LaMa_ServiceConfigRemoval.yml	37
Ansible Playbook NetApp_LaMa_ClearMountConfig.yml	38
Ansible 재고 샘플	39
결론	40
추가 정보를 찾을 수 있는 위치	40
버전 기록	41

Ansible을 사용한 NetApp SAP Landscape Management 통합

TR-4953: Ansible을 사용한 NetApp SAP 환경 관리 통합

SAP LaMa(Landscape Management)를 사용하면 SAP 시스템 관리자가 전체 SAP 시스템의 클론 복제, 복사 및 업데이트 작업을 비롯한 SAP 시스템 작업을 자동화할 수 있습니다.

저자: 마이클 솔로스 서, 닐스 바우어, NetApp

NetApp은 SAP LaMa가 SAP LaMa Automation Studio를 통해 NetApp Snapshot 및 FlexClone과 같은 기술에 액세스할 수 있도록 다양한 Ansible 모듈을 제공합니다. 이러한 기술을 통해 SAP 시스템의 클론 복제, 복사 및 업데이트 작업을 간소화 및 가속화할 수 있습니다.

이러한 통합은 사내에서 NetApp 스토리지 솔루션을 실행하는 고객 또는 Amazon Web Services, Microsoft Azure 또는 Google Cloud Platform과 같은 퍼블릭 클라우드 공급자가 제공하는 NetApp 스토리지 서비스를 사용하는 고객이 사용할 수 있습니다.

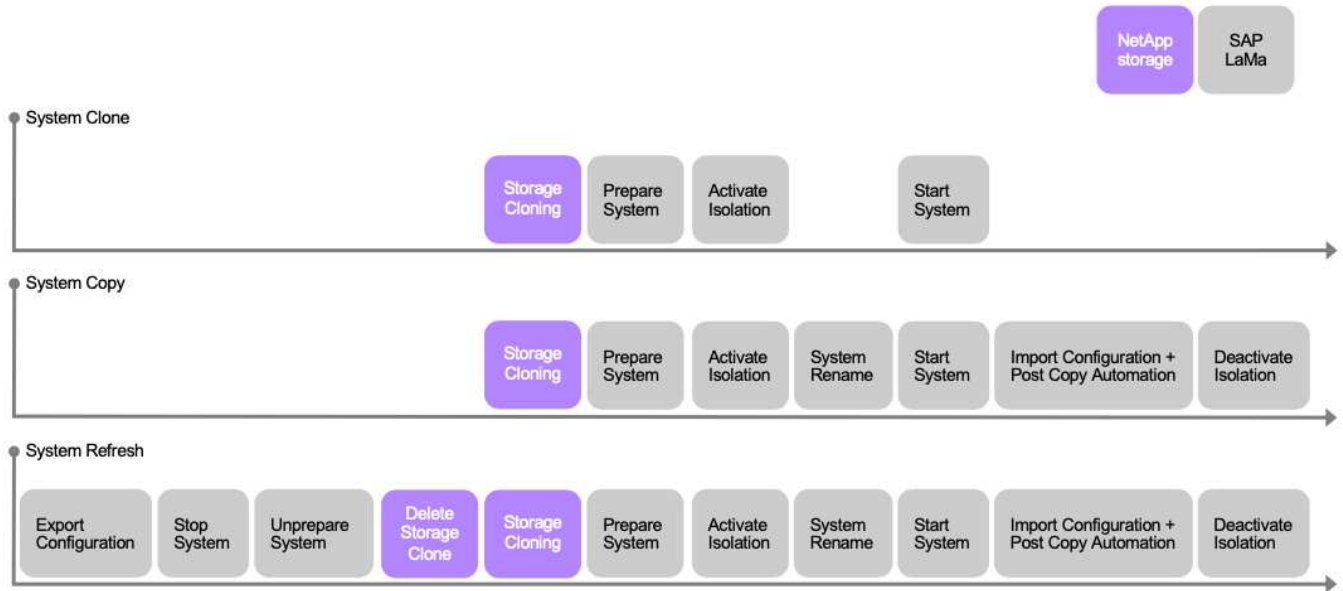
이 문서에서는 Ansible 자동화를 사용하여 SAP 시스템 복사, 클론 복제, 업데이트 작업을 위한 NetApp 스토리지 기능을 갖춘 SAP LaMa 구성을 설명합니다.

SAP 시스템 클론, 복제 및 업데이트 시나리오

SAP 시스템 복사라는 용어는 SAP 시스템 클론, SAP 시스템 복제 또는 SAP 시스템 교체와 같은 세 가지 프로세스의 동의어로 사용되는 경우가 많습니다. 워크플로와 사용 사례는 각각 다르므로 서로 다른 작업을 구분하는 것이 중요합니다.

- *** SAP 시스템 클론.** * SAP 시스템 클론은 소스 SAP 시스템의 동일한 클론입니다. SAP 시스템 클론은 일반적으로 논리적 손상을 다루거나 재해 복구 시나리오를 테스트하는 데 사용됩니다. 시스템 클론 작업에서 호스트 이름, 인스턴스 번호 및 SID는 동일하게 유지됩니다. 따라서 대상 시스템에 대해 적절한 네트워크 펜싱을 설정하여 운영 환경과 통신이 없는지 확인해야 합니다.
- *** SAP 시스템 복사.** * SAP 시스템 복사본은 소스 SAP 시스템의 데이터를 포함하는 새로운 타겟 SAP 시스템의 설정입니다. 예를 들어 새로운 대상 시스템은 운영 시스템의 데이터가 포함된 추가 테스트 시스템일 수 있습니다. 호스트 이름, 인스턴스 번호 및 SID는 소스 시스템과 타겟 시스템에 따라 다릅니다.
- *** SAP 시스템 업데이트.** * SAP 시스템 새로 고침은 기존 타겟 SAP 시스템을 소스 SAP 시스템의 데이터로 새로 고치는 것입니다. 타겟 시스템은 일반적으로 운영 시스템의 데이터로 업데이트되는 품질 보증 시스템과 같은 SAP 전송 환경의 일부입니다. 호스트 이름, 인스턴스 번호 및 SID는 소스 시스템과 타겟 시스템에 따라 다릅니다.

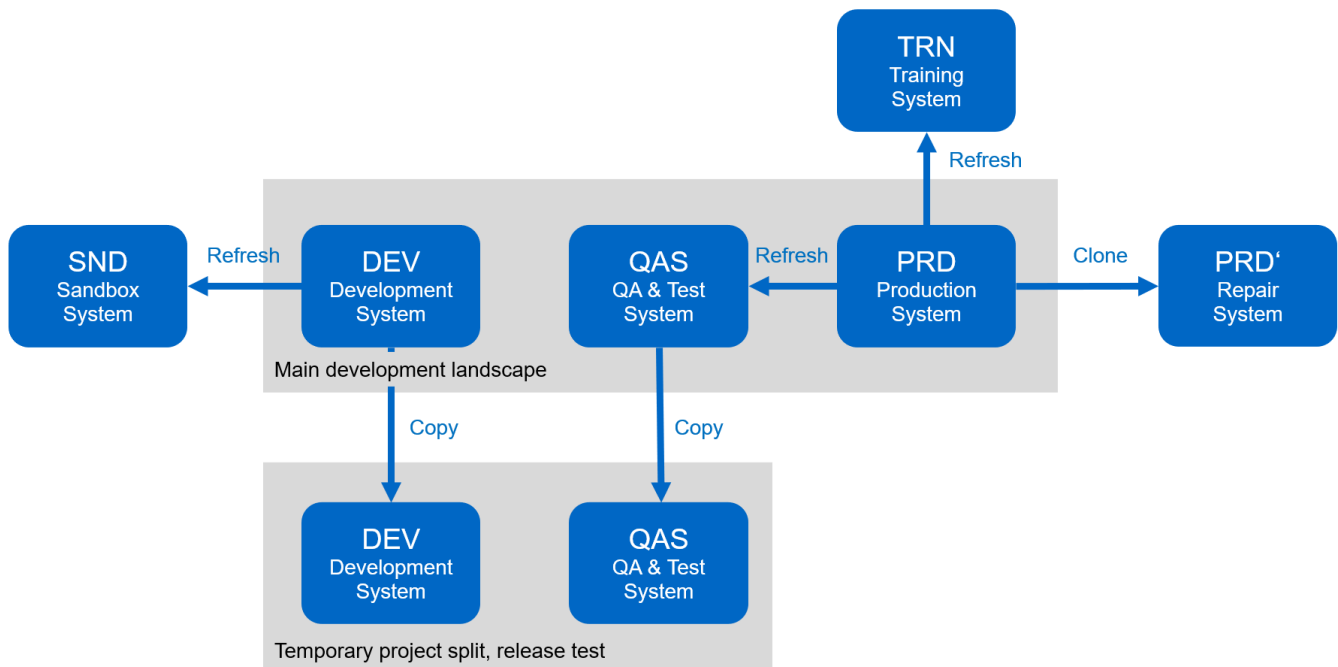
다음 그림에서는 시스템 클론, 시스템 복사본 또는 시스템 업데이트 작업 중에 수행해야 하는 주요 단계를 보여 줍니다. 보라색 상자는 NetApp 스토리지 기능을 통합할 수 있는 단계를 나타냅니다. SAP LaMa를 사용하면 세 가지 작업을 모두 완벽하게 자동화할 수 있습니다.



시스템 업데이트, 복사 및 클론 복제 사용 사례

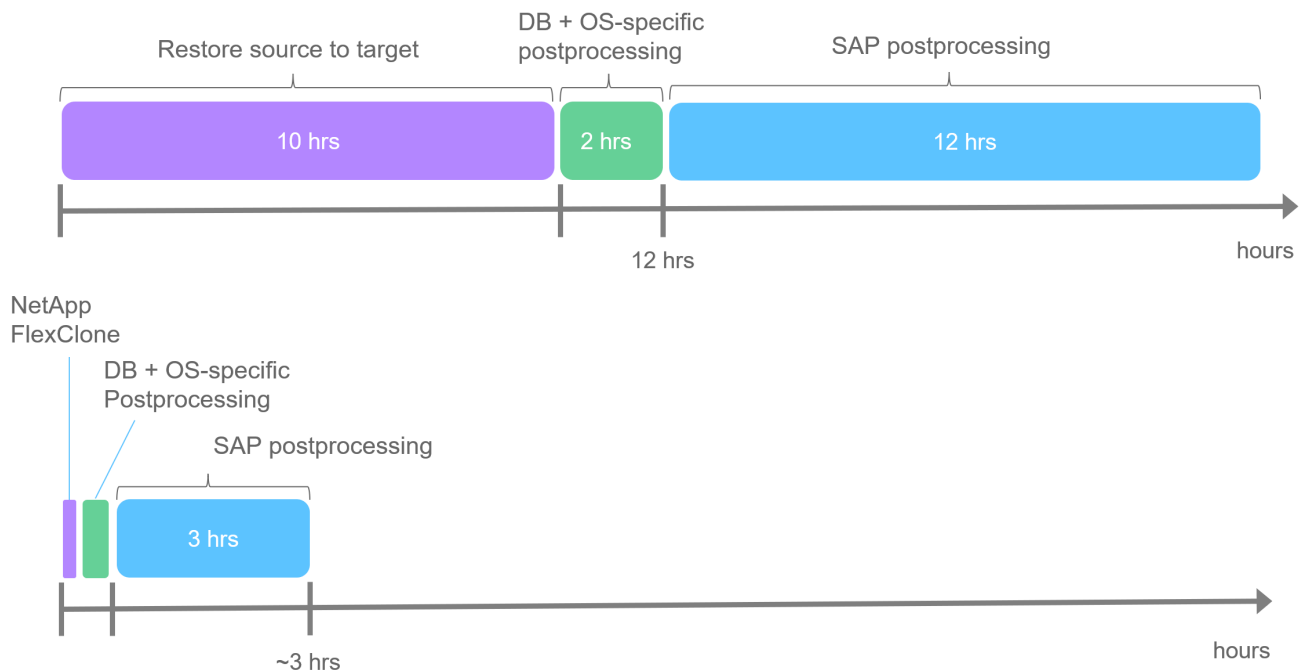
테스트 또는 교육 목적으로 대상 시스템에서 소스 시스템의 데이터를 사용할 수 있도록 여러 가지 시나리오가 있습니다. 이러한 테스트 및 교육 시스템은 정기적으로 소스 시스템의 데이터를 업데이트하여 테스트 및 교육이 현재 데이터 세트와 함께 수행되도록 해야 합니다.

이러한 시스템 업데이트 작업은 인프라, 데이터베이스 및 애플리케이션 계층에 있는 여러 작업으로 구성되며 자동화 수준에 따라 몇 일이 걸릴 수 있습니다.



SAP LaMa와 NetApp 클론 복제 워크플로우를 사용하면 인프라 및 데이터베이스 계층에서 필요한 작업을 더 빠르게 처리하고 자동화할 수 있습니다. SAP LaMa는 소스 시스템에서 타겟 시스템으로 백업을 복원하는 대신 NetApp

Snapshot 복사본과 NetApp FlexClone 기술을 사용하여 다음 그림과 같이 몇 시간이 아닌 몇 분 이내에 HANA 데이터베이스를 시작하는 데 필요한 작업을 수행할 수 있습니다. 클론 생성 프로세스에 필요한 시간은 데이터베이스의 크기와 무관합니다. 따라서 몇 분 안에 초대형 시스템도 만들 수 있습니다. 운영 체제 및 데이터베이스 계층뿐만 아니라 SAP 사후 처리 측면에서도 작업을 자동화하여 런타임을 더욱 줄일 수 있습니다.



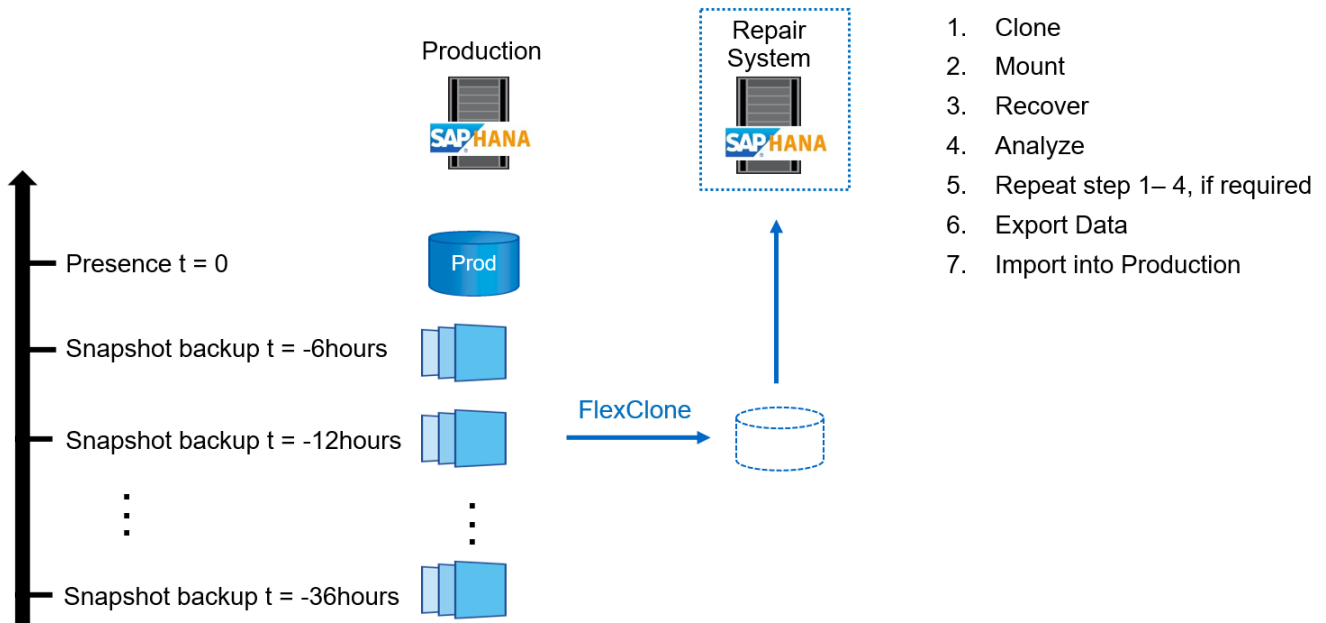
논리적 손상을 처리합니다

논리적 손상은 소프트웨어 오류, 인적 오류 또는 태업 때문에 발생할 수 있습니다. 하지만 논리적 손상은 표준고가용성 및 재해 복구 솔루션을 통해 해결할 수 없는 경우가 많습니다. 따라서 논리적 손상이 발생한 계층, 애플리케이션, 파일 시스템 또는 스토리지에 따라 가동 중지 시간을 최소화하고 허용되는 데이터 손실 요구 사항을 충족하지 못하는 경우가 있습니다.

최악의 경우는 SAP 애플리케이션의 논리적 손상입니다. SAP 애플리케이션은 서로 다른 애플리케이션이 서로 통신하고 데이터를 교환하는 환경에서 작동하는 경우가 많습니다. 따라서 논리적 손상이 발생한 SAP 시스템을 복원 및 복구하는 것은 권장되는 방법이 아닙니다. 손상이 발생하기 전의 시점으로 시스템을 복원하면 데이터가 손실됩니다. 또한 SAP 환경은 더 이상 동기화되지 않으며 추가 후처리 작업이 필요합니다.

SAP 시스템을 복원하는 대신 별도의 복구 시스템에서 문제를 분석하여 시스템 내의 논리적 오류를 해결하는 것이 더 좋습니다. 근본 원인 분석에는 비즈니스 프로세스 및 애플리케이션 소유자의 참여가 필요합니다. 이 시나리오에서는 논리적 손상이 발생하기 전에 저장된 데이터를 기반으로 복구 시스템(운영 시스템의 클론)을 생성합니다. 복구 시스템 내에서 필요한 데이터를 내보낸 후 운영 시스템으로 가져올 수 있습니다. 이 방법을 사용하면 운영 시스템을 중단할 필요가 없으며, 최상의 시나리오에서는 데이터가 없거나 극히 일부의 데이터만 손실됩니다.

수리 시스템을 설정할 때는 유연성과 속도가 매우 중요합니다. NetApp 스토리지 기반 Snapshot 백업을 사용하면 일관된 여러 데이터베이스 이미지를 사용하여 NetApp FlexClone 기술을 사용하여 운영 시스템의 클론을 생성할 수 있습니다. 파일 기반 백업에서 리디렉션된 복원을 사용하여 복구 시스템을 설정하는 경우 몇 시간이 아니라 몇 초 이내에 FlexClone 볼륨을 생성할 수 있습니다.

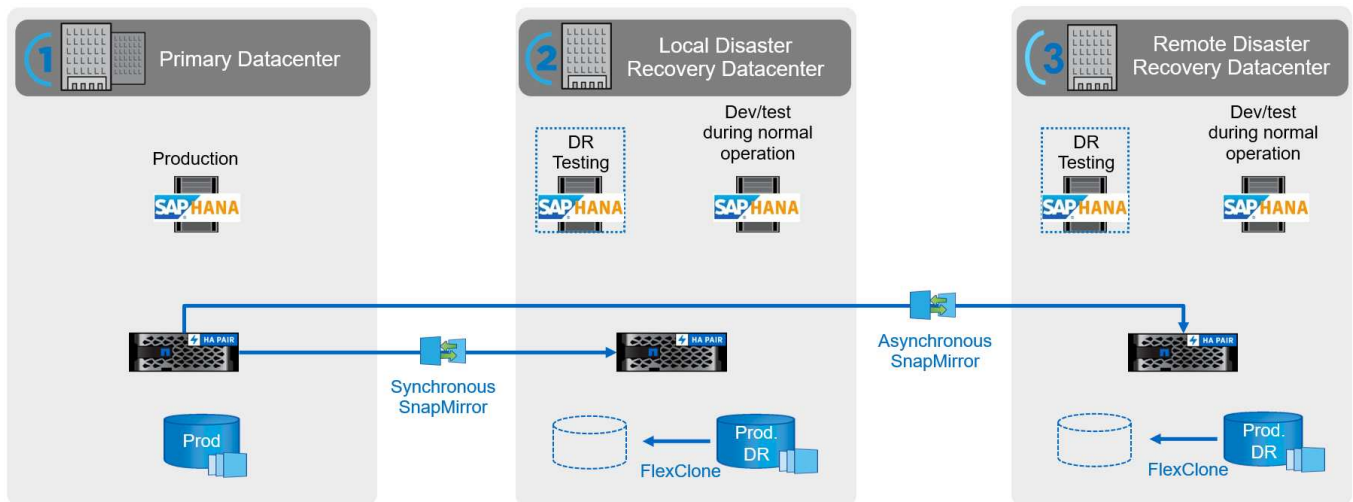


재해 복구 테스트

효과적인 재해 복구 전략을 위해서는 필요한 워크플로우를 테스트해야 합니다. 테스트는 전략이 제대로 작동하는지 여부와 내부 문서화가 충분한지 여부를 보여 줍니다. 또한 관리자가 필요한 절차를 교육할 수 있습니다.

SnapMirror를 통한 스토리지 복제를 사용하면 RTO 및 RPO를 위험에 빠뜨리지 않고 재해 복구 테스트를 실행할 수 있습니다. 데이터 복제를 중단하지 않고 재해 복구 테스트를 수행할 수 있습니다. 비동기식 및 동기식 SnapMirror의 재해 복구 테스트에서는 재해 복구 대상에 있는 Snapshot 백업과 FlexClone 볼륨을 사용합니다.

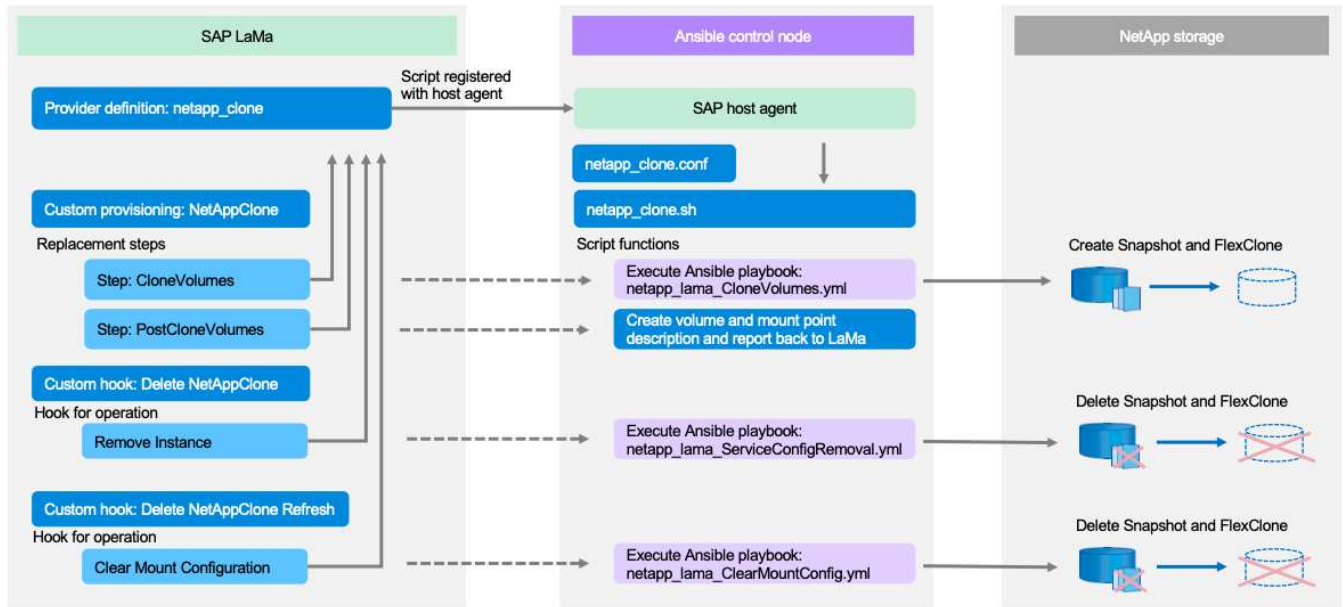
SAP LaMa는 전체 테스트 절차를 조율하는 데 사용할 수 있으며 네트워크 펜싱, 대상 호스트 유지 관리 등을 수행합니다.



Ansible을 사용하여 NetApp SAP LaMa 통합

이 통합 방식에서는 NetApp 스토리지 관리를 위한 Ansible 플레이북을 사용하여 SAP LaMa

맞춤형 프로비저닝 및 운영 후크를 사용합니다. 다음 그림에서는 LaMa 측 구성과 예제 구현의 해당 구성 요소를 개괄적으로 보여 줍니다.



Ansible 제어 노드 역할을 하는 중앙 호스트는 SAP LaMa의 요청을 실행하고 Ansible 플레이북을 사용하여 NetApp 스토리지 운영을 트리거하는 데 사용됩니다. SAP LaMa에 대한 통신 게이트웨이로 호스트를 사용할 수 있도록 SAP 호스트 에이전트 구성 요소가 이 호스트에 설치되어 있어야 합니다.

LaMa Automation Studio 내에서 Ansible 호스트의 SAP 호스트 에이전트에 등록된 공급자가 정의됩니다. Host Agent 구성 파일은 요청된 작업에 따라 명령줄 매개 변수 집합을 사용하여 SAP LaMa에서 호출되는 셸 스크립트를 가리킵니다.

LaMa Automation Studio 내에서 사용자 지정 프로비저닝과 사용자 지정 후크는 프로비저닝 중에 스토리지 클론 생성 작업을 실행하고, 시스템 프로비저닝 해제 시 정리 작업을 실행하도록 정의됩니다. 그런 다음 Ansible 제어 노드의 셸 스크립트가 해당하는 Ansible 플레이북을 실행하여 스냅샷 및 FlexClone 작업을 트리거하고 프로비저닝 해제 워크플로우에서 클론을 삭제합니다.

NetApp Ansible 모듈과 LaMa Provider 정의에 대한 자세한 내용은 다음 사이트에서 확인할 수 있습니다.

- ["NetApp Ansible 모듈"](#)
- ["SAP LaMa 설명서 – 공급자 정의"](#)

구현 예

시스템 및 스토리지 설정에 사용할 수 있는 옵션이 많기 때문에 예제 구현은 개별 시스템 설정 및 구성 요구 사항을 템플릿으로 사용해야 합니다.



예제 스크립트는 있는 그대로 제공되며 NetApp에서 지원하지 않습니다. <mailto:ng-sapcc@netapp.com> [ng-sapcc@netapp.com]으로 이메일을 통해 스크립트의 현재 버전을 요청할 수 있습니다.

검증된 구성 및 제한

예제 구현에 적용된 원리는 다음과 같습니다. 고객의 요구 사항에 맞게 조정해야 할 수 있습니다.

- 관리형 SAP 시스템은 NFS를 사용하여 NetApp 스토리지 볼륨에 액세스했으며 적응형 설계 원칙에 따라 설정되었습니다.
- NetApp Ansible 모듈(ZAPI 및 REST API)에서 지원하는 모든 ONTAP 릴리즈를 사용할 수 있습니다.
- 단일 NetApp 클러스터 및 SVM에 대한 자격 증명은 공급자 스크립트에서 변수로 하드 코딩되었습니다.
- 소스 SAP 시스템에서 사용한 것과 동일한 스토리지 시스템에서 스토리지 클론 생성이 수행되었습니다.
- 타겟 SAP 시스템의 스토리지 볼륨은 부록이 있는 소스와 이름이 동일합니다.
- 2차 스토리지(SV/SM)에서 클론이 구축되지 않았습니다.
- FlexClone 분할이 구현되지 않았습니다.
- 소스 및 타겟 SAP 시스템에 대한 인스턴스 번호가 동일합니다.

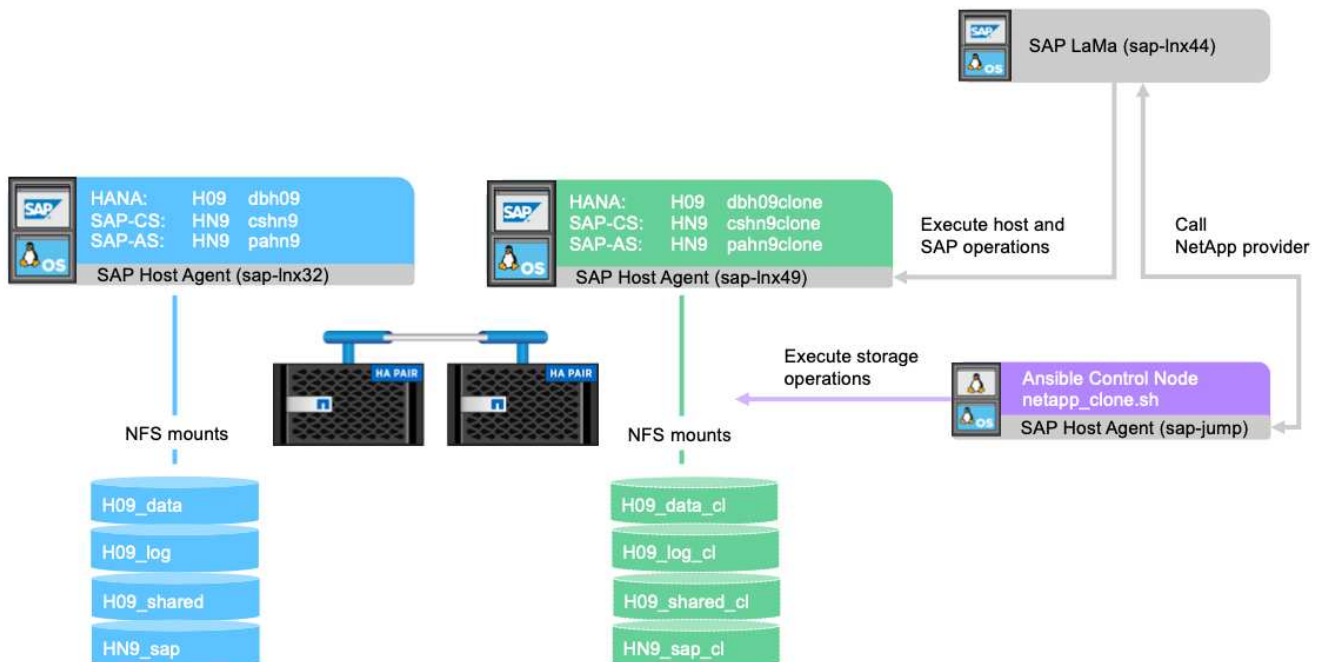
랩 설정

다음 그림은 사용된 랩 설정을 보여 줍니다. 시스템 클론 작업에 사용되는 소스 SAP 시스템 HN9는 데이터베이스 H09, SAP CS 및 SAP가 설치된 동일한 호스트(SAP-lnx32)에서 실행되는 서비스로 구성되었습니다 **"적응형 설계"** 활성화됨. 에 따라 Ansible 컨트롤 노드가 준비되었습니다 **"NetApp ONTAP용 Ansible 플레이북"** 문서화:

이 호스트에도 SAP 호스트 에이전트가 설치되었습니다. NetApp 공급자 스크립트와 Ansible 플레이북은 에 설명된 대로 Ansible 제어 노드에서 구성되었습니다 **"부록: 공급자 스크립트 구성"**

호스트 sap-lnx49 SAP LaMa 클론 복제 작업의 타겟으로 사용되었으며, 격리 지원 기능이 구성되었습니다.

PCA(Post Copy Automation)가 활성화되어 있기 때문에 시스템 복사 및 새로 고침 작업에 여러 SAP 시스템(HNA를 소스로, HN2 타겟을 대상으로)이 사용되었습니다.



실습 설치 시 사용된 소프트웨어 릴리즈는 다음과 같습니다.

- SAP LaMa Enterprise Edition 3.00 SP23_2
- SAP HANA 2.00.052.00.1599235305
- SAP 7.77 패치 27(S/4 HANA 1909)
- SAP 호스트 에이전트 7.22 패치 56
- SAPACEXT 7.22 패치 69
- Linux SLES 15 SP2
- Ansible 2. 13.7
- NetApp ONTAP 9.8P8

SAP LaMa 구성

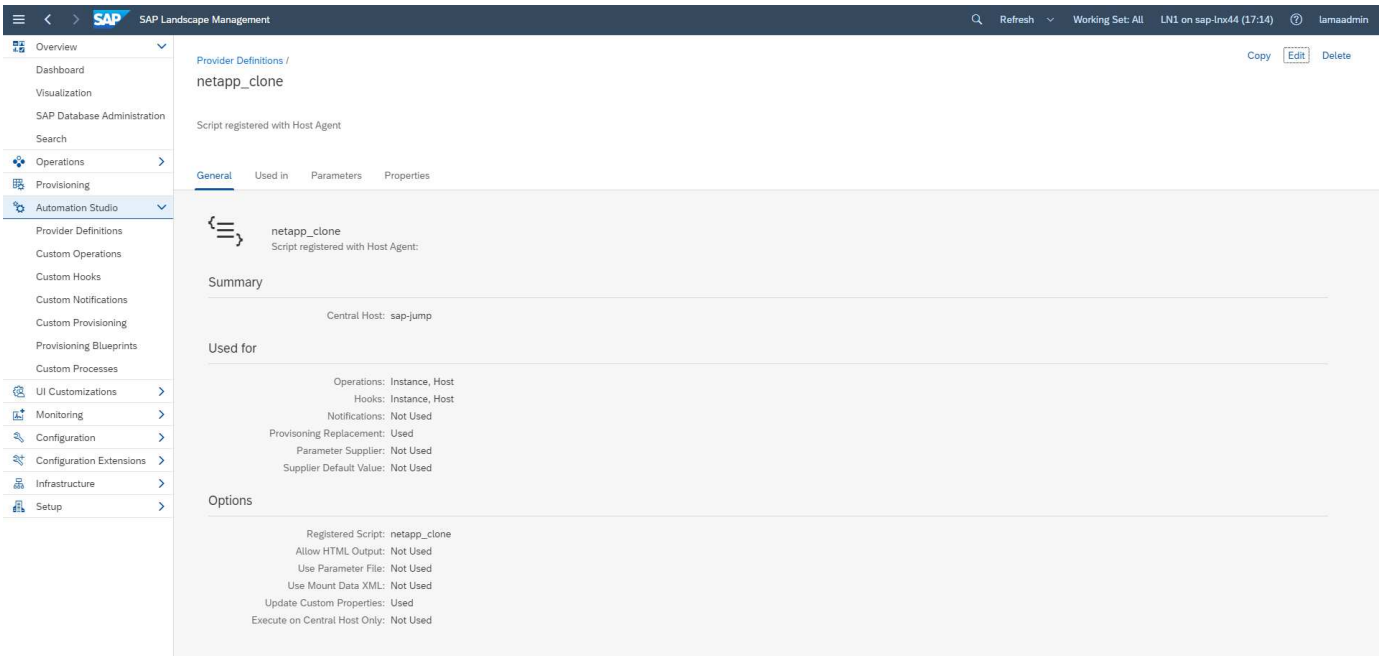
SAP LaMa Provider 정의

공급자 정의는 다음 스크린샷과 같이 SAP LaMa의 Automation Studio 내에서 수행됩니다. 예제 구현에서는 앞에서 설명한 것처럼 다양한 사용자 지정 프로비저닝 단계 및 작업 후크에 사용되는 단일 공급자 정의를 사용합니다.

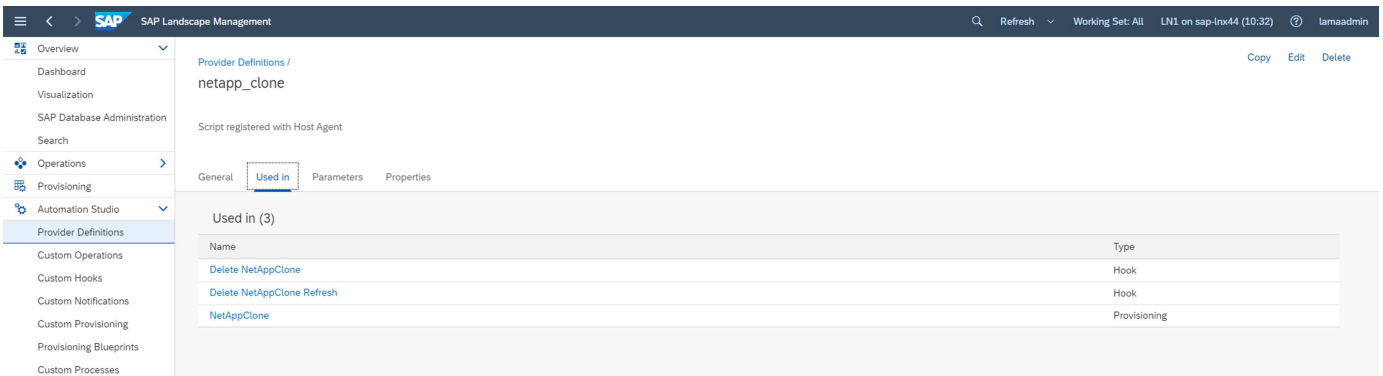
The screenshot shows the SAP Landscape Management (SLM) Automation Studio interface. The left sidebar contains navigation options: Overview, Dashboard, Visualization, SAP Database Administration, Search, Operations, Provisioning, and Automation Studio. The Automation Studio section is expanded, showing 'Provider Definitions'. The main area displays '1 Providers' and a table with the following data:

Name	Type	Target	Used
netapp_clone	Script registered with Host Agent	netapp_clone	<input checked="" type="checkbox"/>

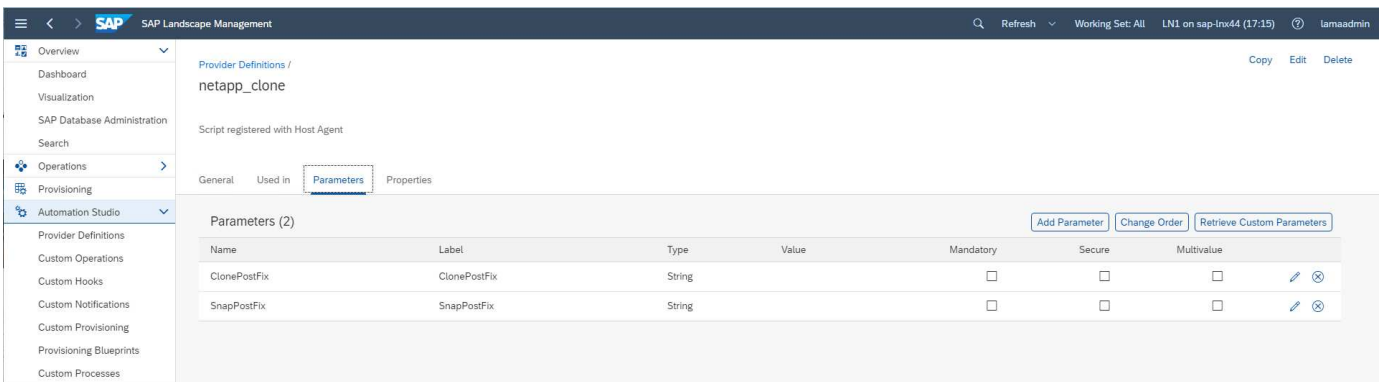
제공합니다 netapp_clone 스크립트로 정의됩니다 netapp_clone.sh SAP Host Agent에 등록되어 있습니다. SAP 호스트 에이전트는 중앙 호스트에서 실행됩니다 `sap-jump` Ansible 제어 노드 역할도 합니다.



[사용] 탭에 공급자가 사용하는 사용자 정의 작업이 표시됩니다. 사용자 지정 프로비저닝 * NetAppClone * 및 사용자 지정 후크 * NetAppClone 삭제 * 및 * NetAppClone 새로 고침 삭제 *에 대한 구성은 다음 장에 나와 있습니다.

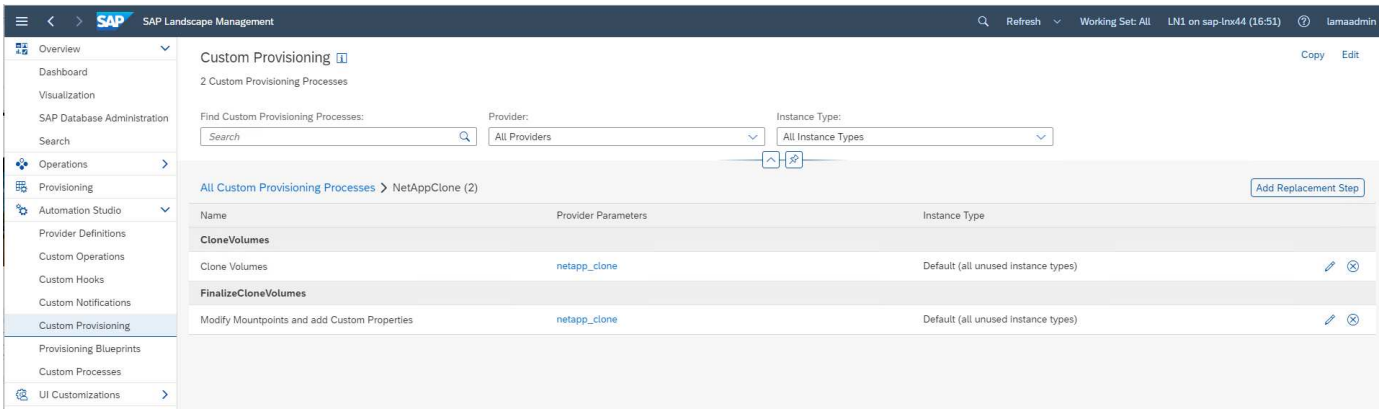


프로비저닝 워크플로를 실행하는 동안 * ClonePostFix * 및 * SnapPostFix * 매개 변수가 요청되며 스냅샷 및 FlexClone 볼륨 이름에 사용됩니다.



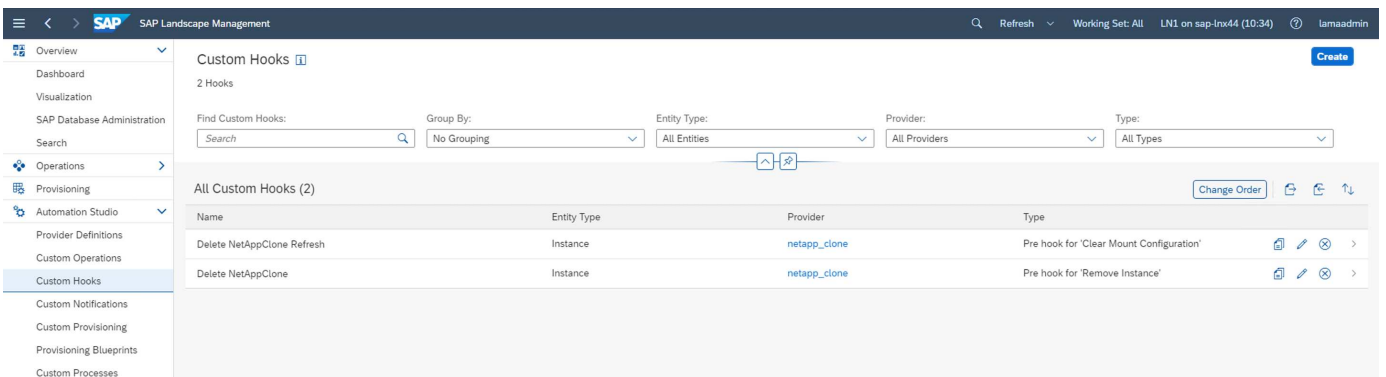
SAP LaMa 맞춤형 프로비저닝

SAP LaMa 사용자 지정 프로비저닝 구성에서 앞서 설명한 고객 공급자는 프로비저닝 워크플로우 단계 * 클론 볼륨 * 및 * PostCloneVolumes *를 대체하는 데 사용됩니다.

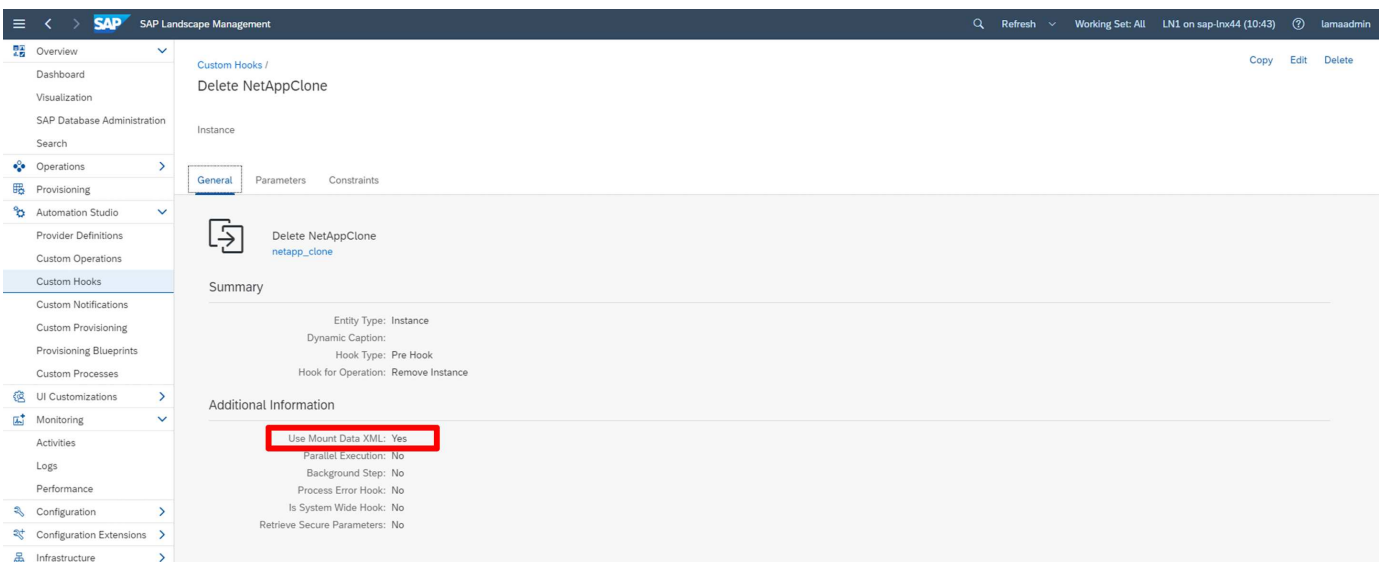


SAP LaMa 맞춤형 후크

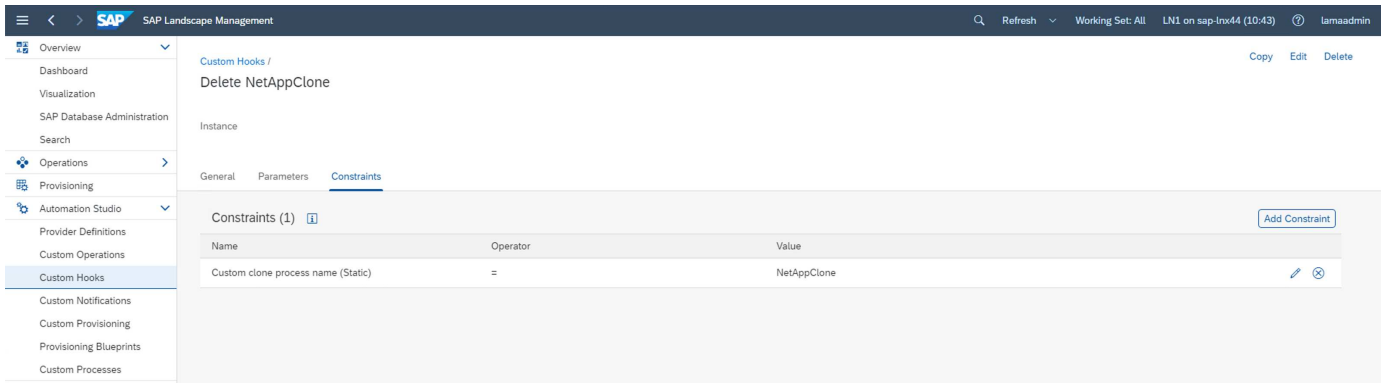
시스템이 시스템 제거 워크플로와 함께 삭제된 경우 hook * Delete NetAppClone * 은 공급자 정의를 호출하는 데 사용됩니다 netapp_clone. 실행 중에 인스턴스가 유지되므로 * NetApp 클론 업데이트 삭제 * 후크는 시스템 업데이트 워크플로우 중에 사용됩니다.



SAP LaMa가 공급업체에 마운트 지점 구성 정보를 제공할 수 있도록 사용자 지정 후크에 대해 * Mount Data XML * 사용 을 구성하는 것이 중요합니다.



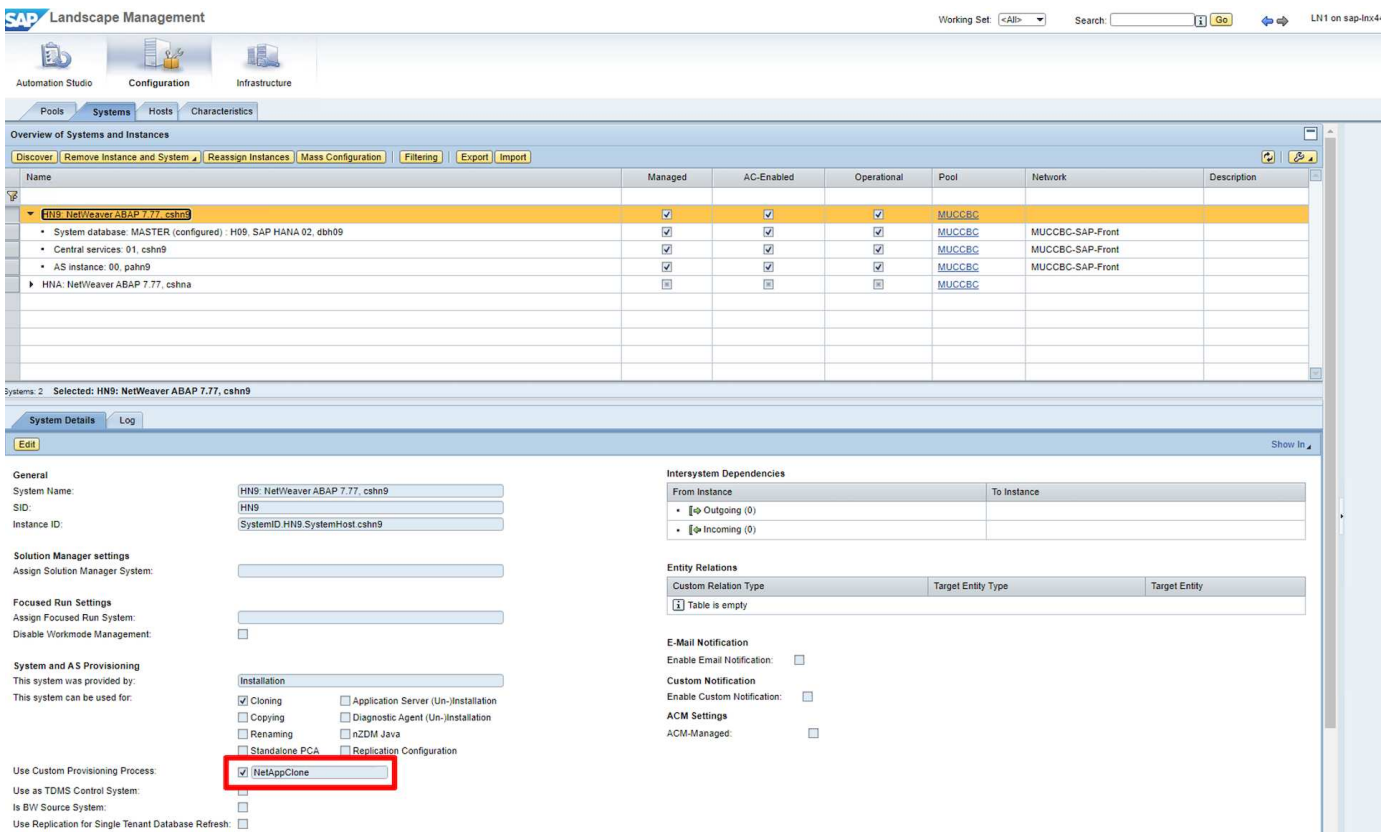
사용자 지정 후크를 사용자 지정 프로비저닝 워크플로로 시스템을 생성할 때만 사용하고 실행되도록 하려면 다음 제약 조건이 추가됩니다.



사용자 지정 후크 사용에 대한 자세한 내용은 에서 확인할 수 있습니다 "[SAP LaMa 설명서](#)".

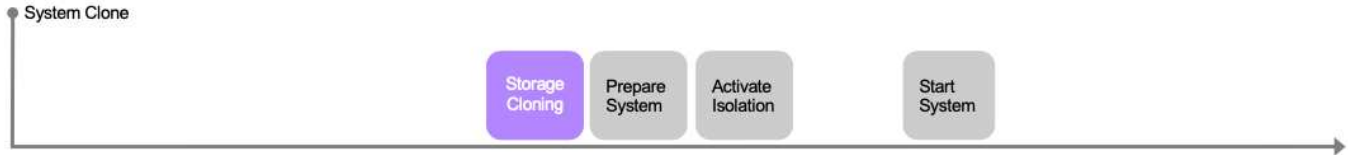
SAP 소스 시스템에 대한 사용자 지정 프로비저닝 워크플로우를 설정합니다

소스 시스템에 대한 사용자 지정 프로비저닝 워크플로우를 설정하려면 구성에서 해당 워크플로우를 적용해야 합니다. 해당 사용자 지정 프로비저닝 정의와 함께 * 사용자 지정 프로비저닝 프로세스 사용 * 확인란을 선택해야 합니다.

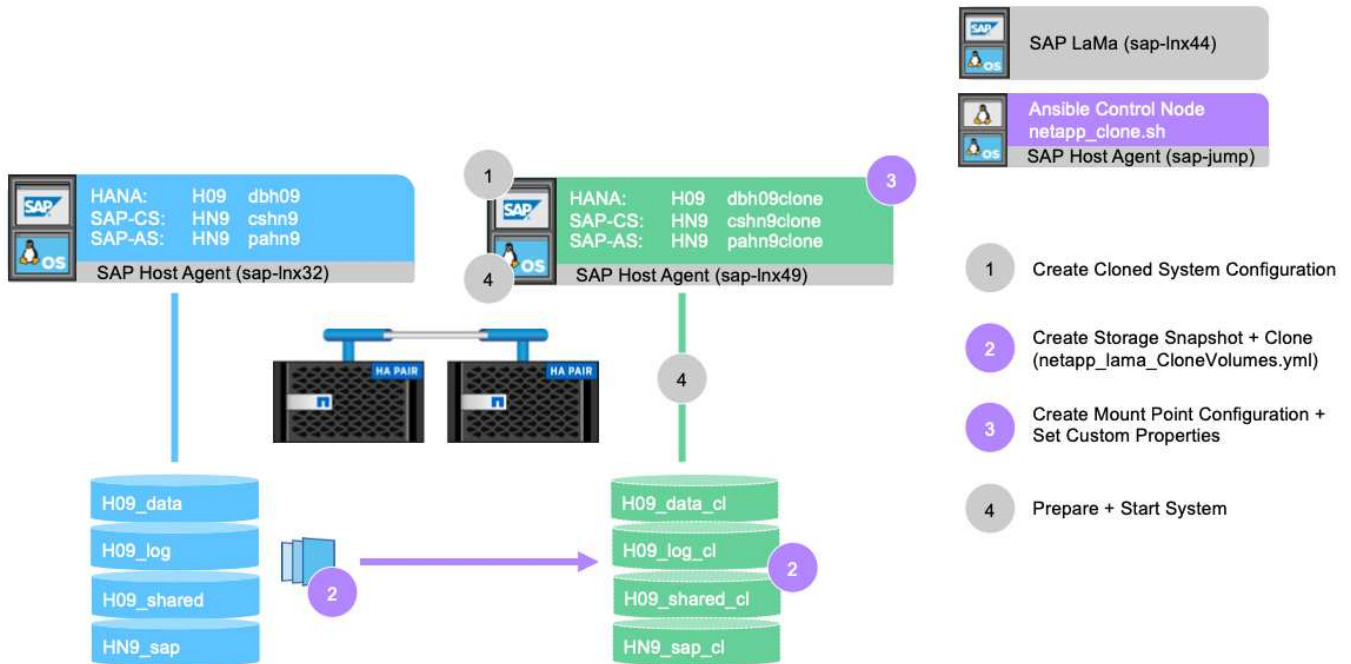


SAP LaMa가 워크플로우-클론 시스템을 프로비저닝합니다

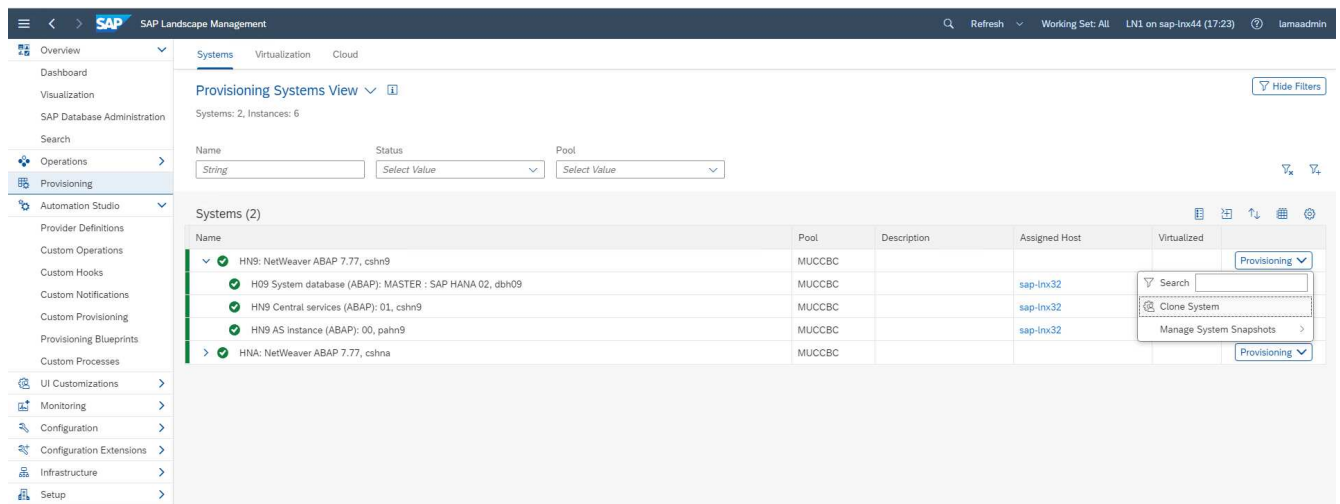
다음 그림에서는 시스템 클론 워크플로우에서 실행되는 주요 단계를 보여 줍니다.



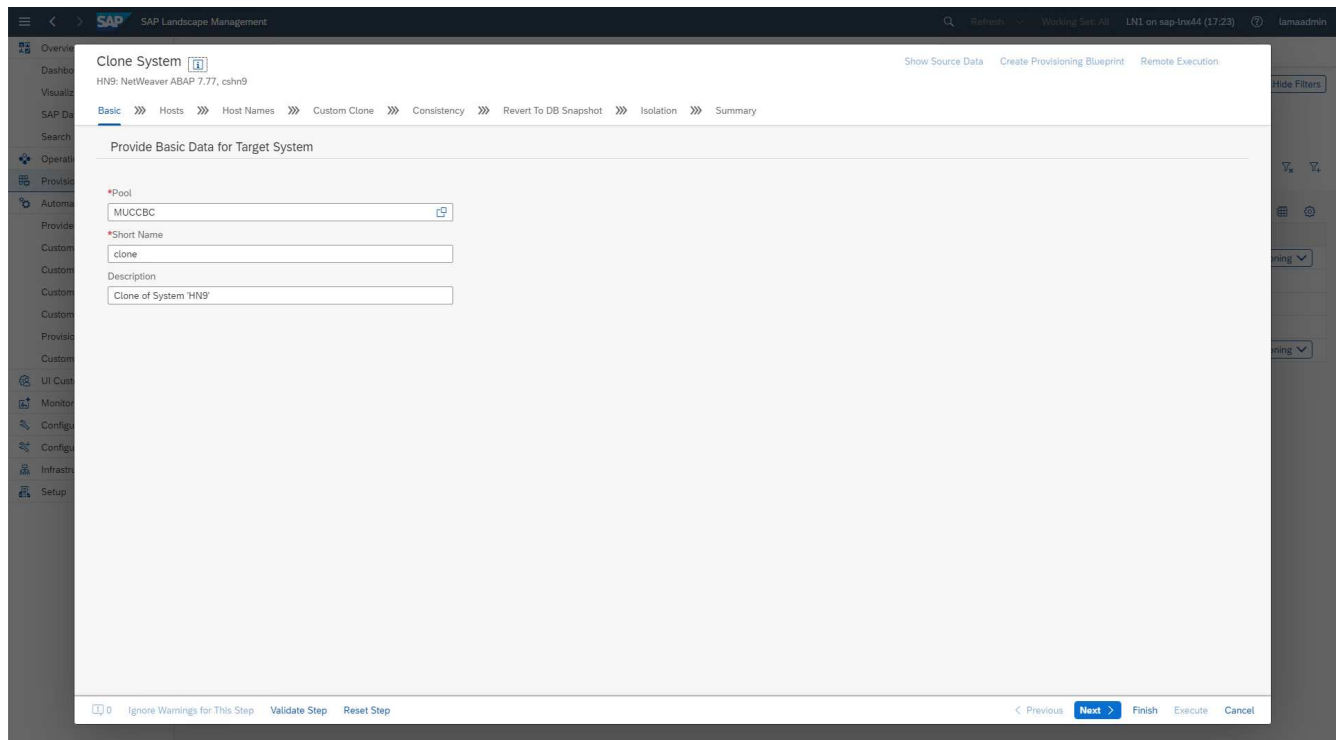
이 섹션에서는 HANA 데이터베이스 H09가 포함된 소스 SAP 시스템 HN9를 기반으로 하는 전체 SAP LaMa 시스템을 클론 복제 워크플로우를 살펴보겠습니다. 다음 그림은 워크플로 중에 실행된 단계에 대한 개요를 제공합니다.



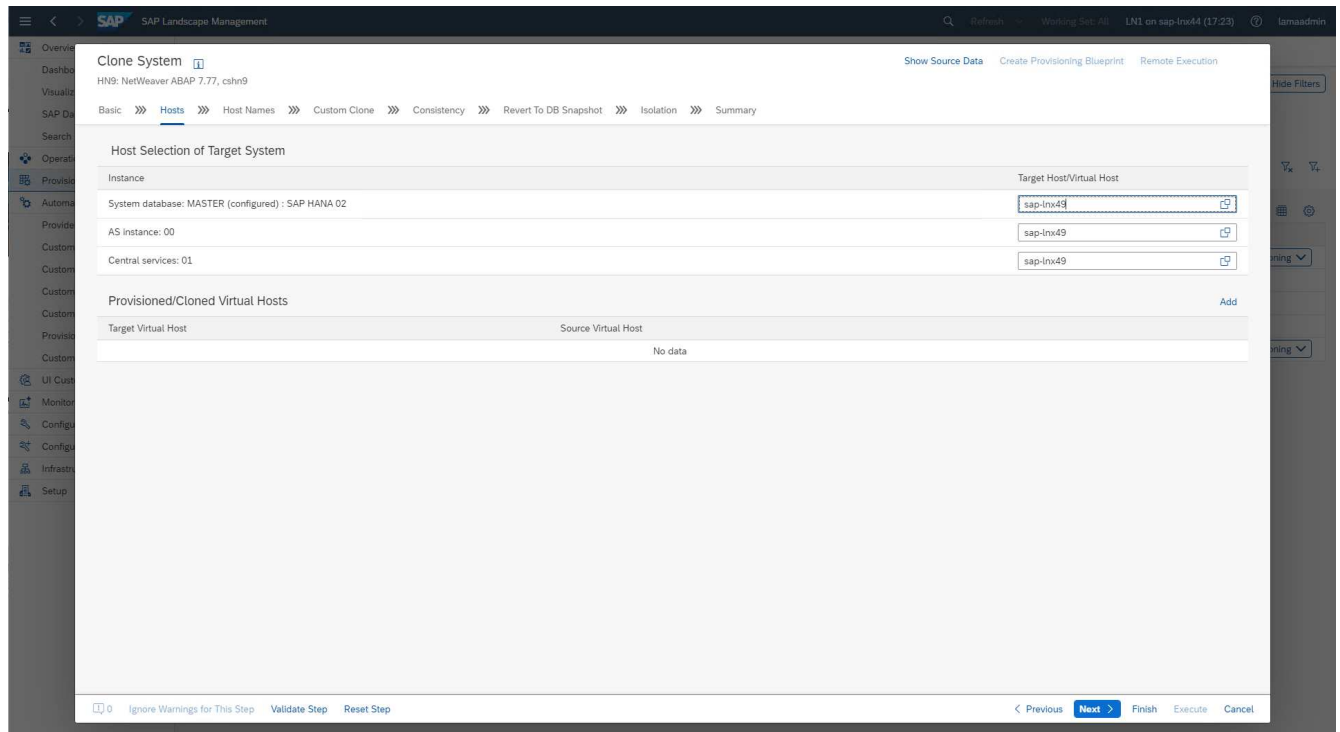
1. 클론 생성 워크플로우를 시작하려면 메뉴 트리에서 * Provisioning * 을 열고 소스 시스템(예: HN9)을 선택합니다. 그런 다음 * 클론 시스템 * 마법사를 시작합니다.



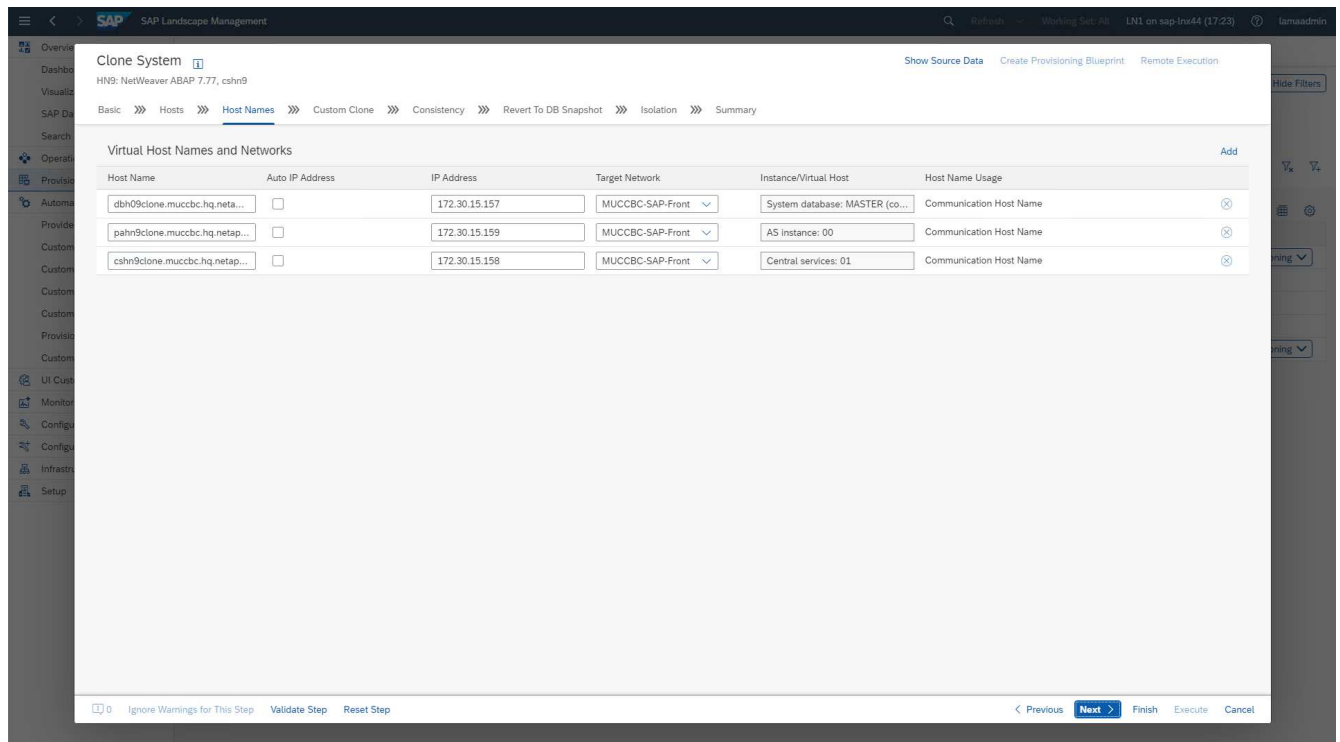
2. 요청된 값을 입력합니다. 마법사의 화면 1에서 클론 생성된 시스템의 풀 이름을 묻는 메시지가 표시됩니다. 이 단계에서는 복제된 시스템을 시작할 인스턴스(가상 또는 물리적)를 지정합니다. 기본값은 시스템을 타겟 시스템과 동일한 풀에 복제하는 것입니다.



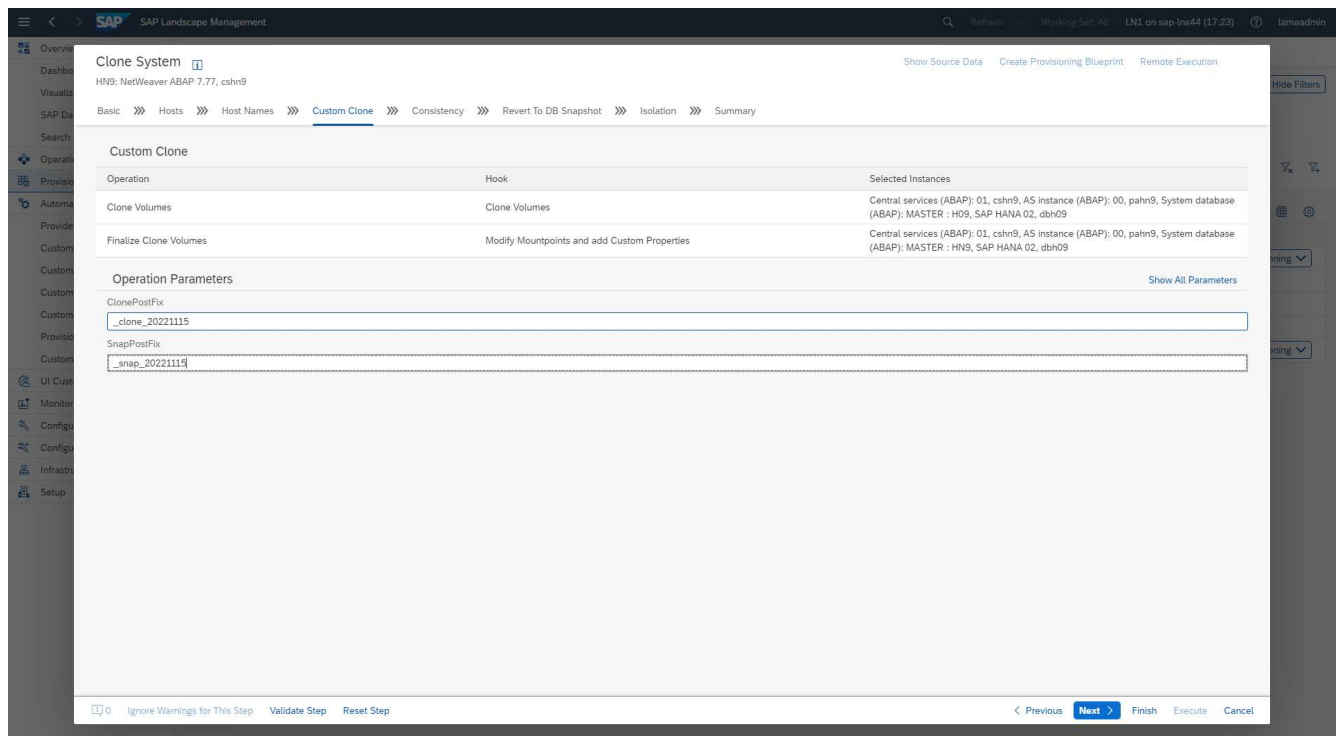
3. 마법사의 화면 2에서 새 SAP 인스턴스가 시작된 타겟 호스트를 요청합니다. 이전 화면에서 지정한 호스트 풀에서 이 인스턴스의 대상 호스트를 선택할 수 있습니다. 각 인스턴스 또는 서비스는 다른 호스트에서 시작할 수 있습니다. 이 예에서는 세 서비스가 모두 동일한 호스트에서 실행됩니다.



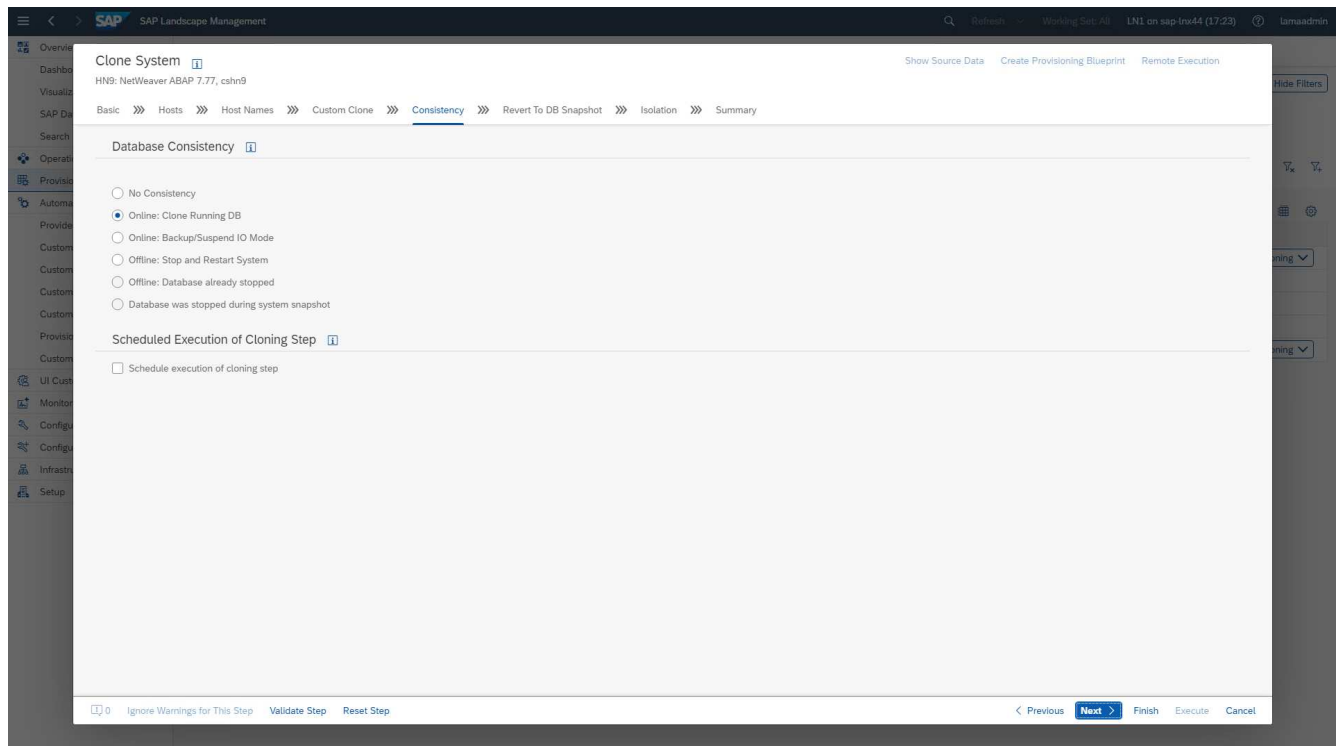
4. 화면 3에서 요청한 정보를 제공하여 가상 호스트 이름과 네트워크를 요청합니다. 일반적으로 호스트 이름은 DNS에서 유지되므로 IP 주소는 이에 따라 미리 채워집니다.



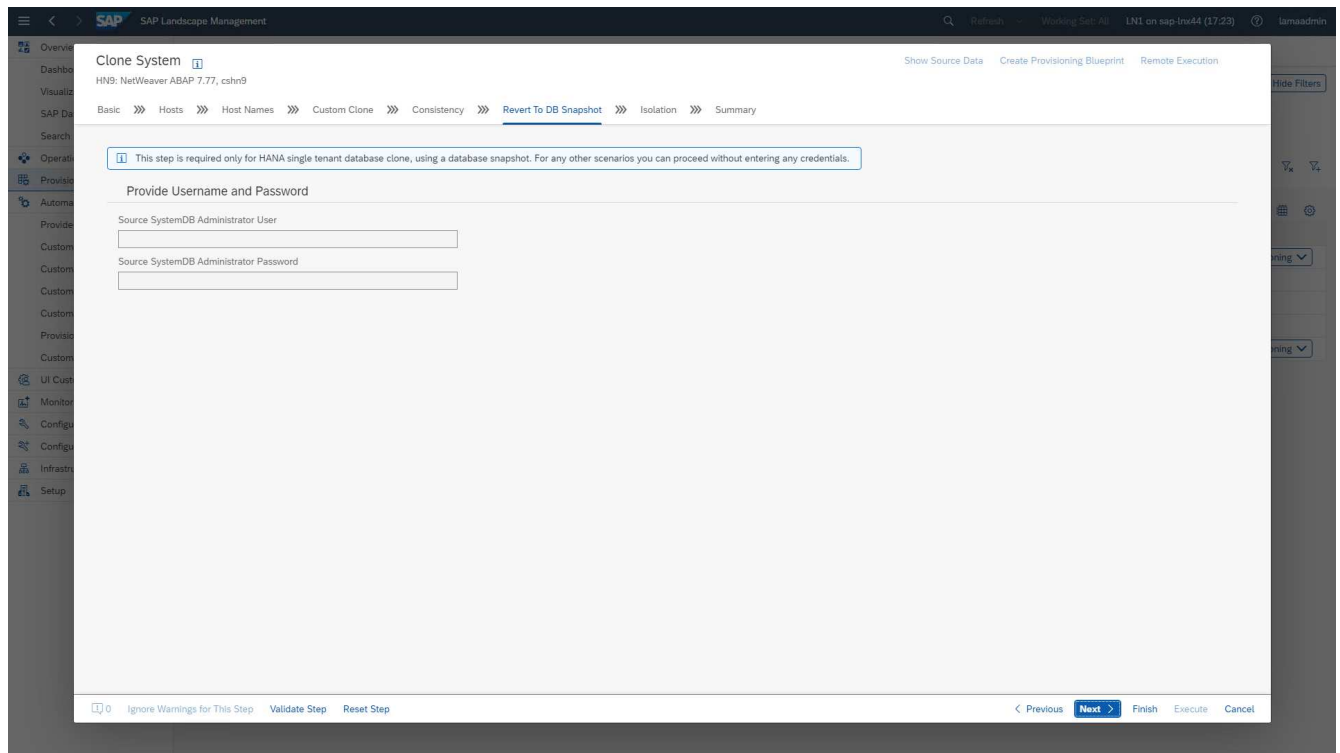
5. 화면 4에 사용자 지정 클론 작업이 나열됩니다. FlexClone 볼륨과 스냅샷 이름의 스토리지 클론 작업 중에 각각 사용되는 클론 및 * SnapPostfix * 이름이 제공됩니다. 이러한 필드를 비워 두면 공급자 스크립트의 변수 섹션에 구성된 기본값이 됩니다 netapp_clone.sh 사용됩니다.



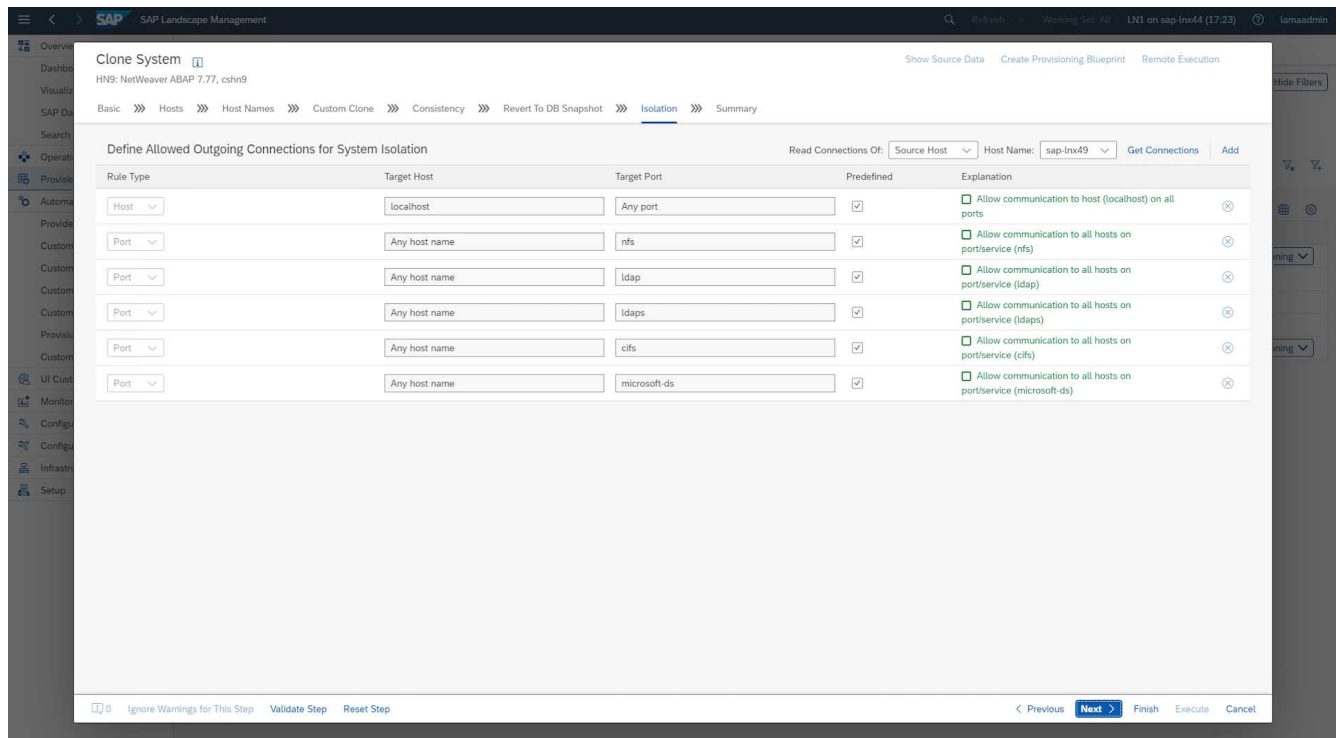
6. 화면 5에서 데이터베이스 정합성 옵션이 선택됩니다. 이 예에서는 * Online: Clone Running DB * 를 선택했습니다.



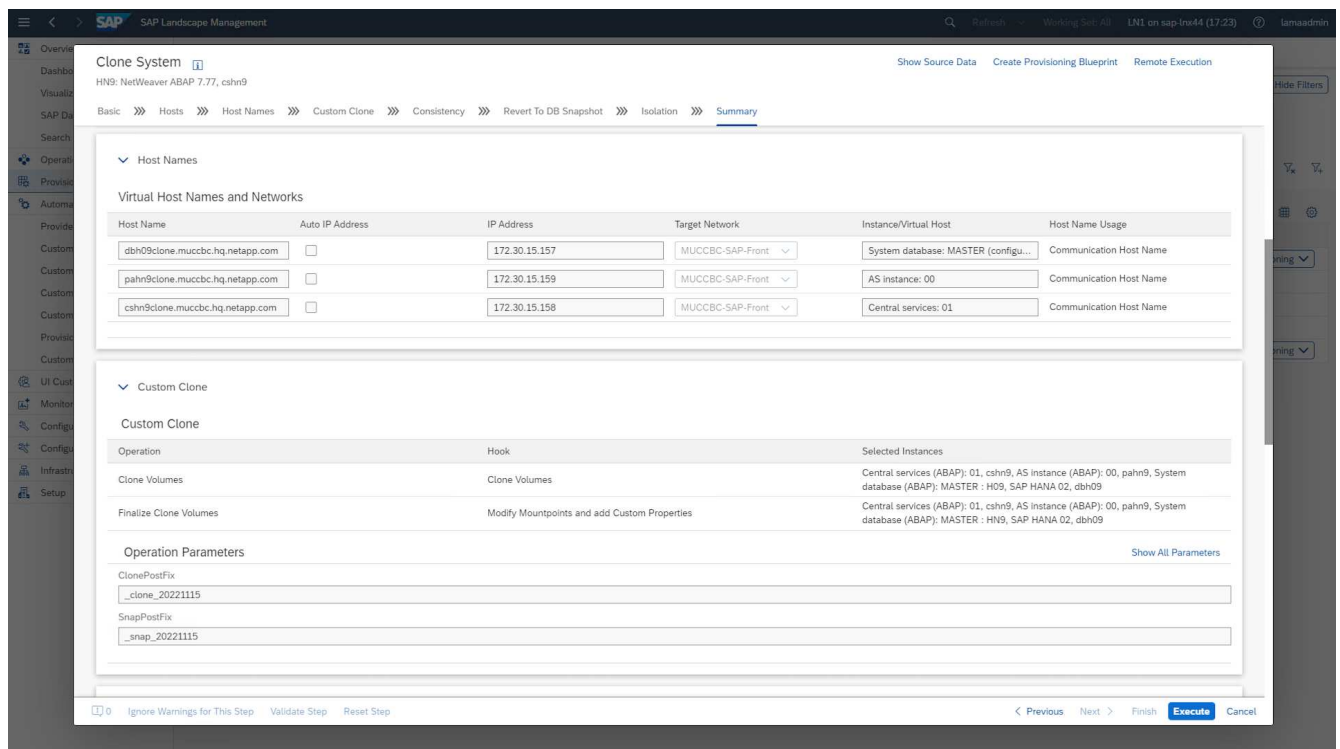
7. 화면 6에서는 테넌트 클론을 수행하는 경우에만 입력이 필요합니다.



8. 화면 7에서 시스템 격리를 구성할 수 있습니다.

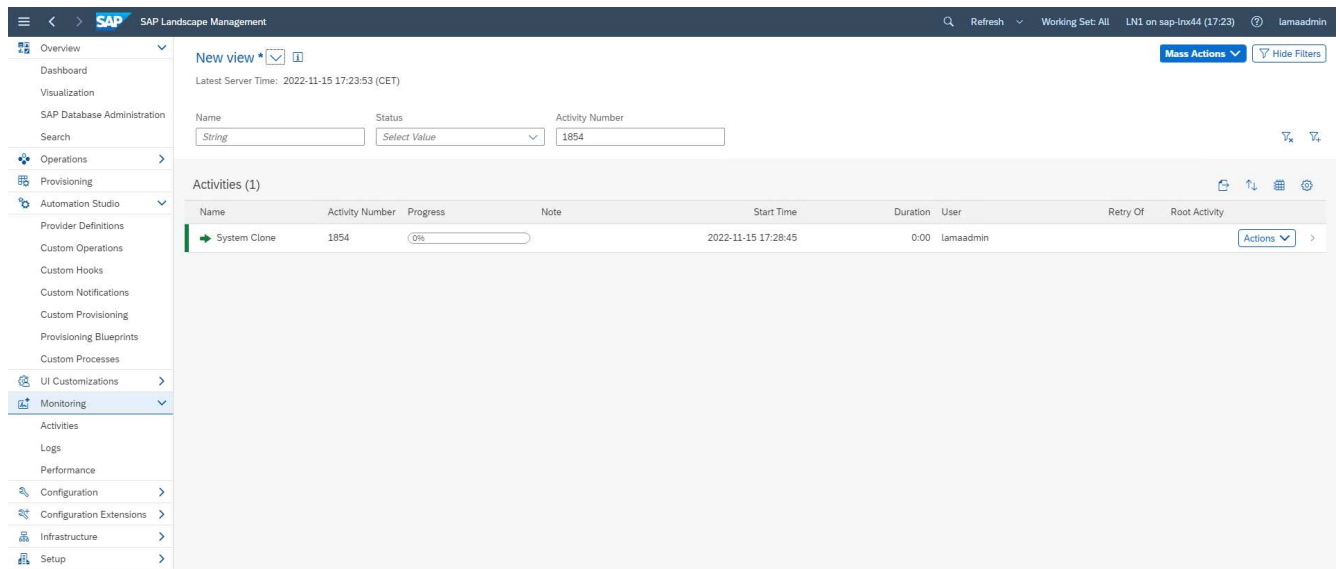


9. 화면 8에서 요약 페이지에는 워크플로가 시작되기 전에 최종 확인을 위한 모든 설정이 포함되어 있습니다. [실행]을 클릭하여 워크플로우를 시작합니다.

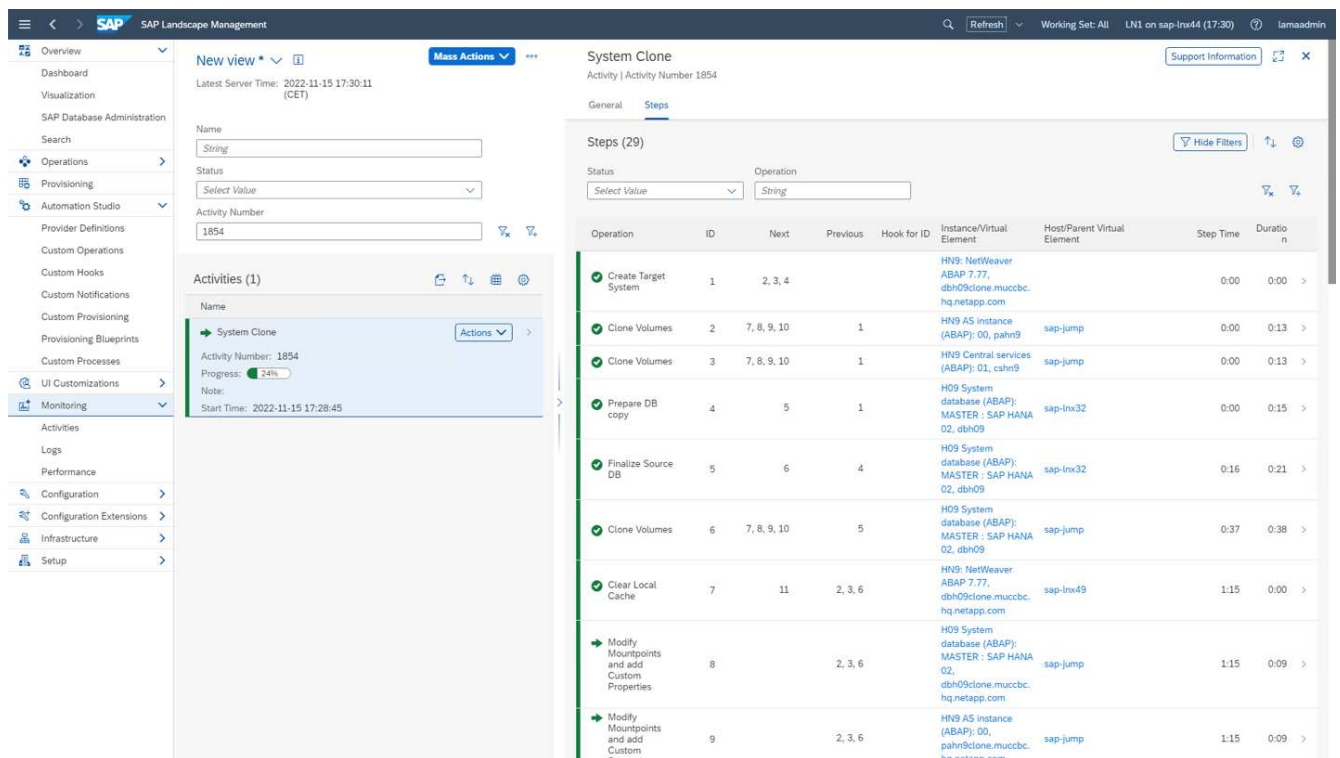


이제 SAP LaMa가 구성에 표시된 모든 작업을 수행합니다. 이러한 조치에는 스토리지 볼륨 클론 및 익스포트 생성, 타겟 호스트에 마운트, 격리를 위한 방화벽 규칙 추가, HANA 데이터베이스 및 SAP 서비스 시작 등이 포함됩니다.

10. 모니터링 * 메뉴에서 클론 워크플로우의 진행 상황을 모니터링할 수 있습니다.



상세 로그 내에서 * 클론 볼륨 * 및 * 마운트 지점 수정 및 사용자 정의 속성 추가 * 작업은 Ansible 노드인 에서 실행됩니다 sap-jump 호스트. 이러한 단계는 각 서비스, HANA 데이터베이스, SAP 중앙 서비스 및 SAP AS 서비스에 대해 실행됩니다.



11. Clone Volumes * 작업을 선택하면 해당 단계에 대한 상세 로그가 표시되며 Ansible Playbook의 실행이 여기에 표시됩니다. Ansible 플레이북이 보일 것입니다 netapp_lama_CloneVolumes.yml 각 HANA 데이터베이스 볼륨, 데이터, 로그 및 공유에 대해 실행됩니다.

The screenshot displays the SAP Landscape Management interface. The left sidebar shows the navigation menu with 'Monitoring' selected. The main area is divided into three panels: 'New view', 'System Clone', and 'Clone Volumes'. The 'System Clone' panel shows the 'Steps' tab with a list of steps including 'Prepare DB copy', 'Finalize Source DB', 'Clone Volumes', 'Clear Local Cache', and 'Modify Mountpoints and add Custom Properties'. The 'Clone Volumes' panel shows the 'Messages' tab with a list of messages. A red box highlights the messages related to 'NetApp Clone for Custom Provis'.

12. 단계 * 마운트 지점 수정 및 사용자 정의 속성 추가 * 의 세부 정보 보기에서 실행 스크립트에서 제공한 마운트 지점 및 사용자 정의 속성에 대한 정보를 찾을 수 있습니다.

The screenshot displays the SAP Landscape Management interface. The left sidebar shows the navigation menu with 'Monitoring' selected. The main area is divided into three panels: 'New view', 'System Clone', and 'Modify Mountpoints and add Custom Properties'. The 'System Clone' panel shows the 'Steps' tab with a list of steps including 'Finalize Source DB', 'Clone Volumes', 'Clear Local Cache', and 'Modify Mountpoints and add Custom Properties'. The 'Modify Mountpoints and add Custom Properties' panel shows the 'Messages' tab with a list of messages. A red box highlights the messages related to 'NetApp Clone for Custom Provis'.

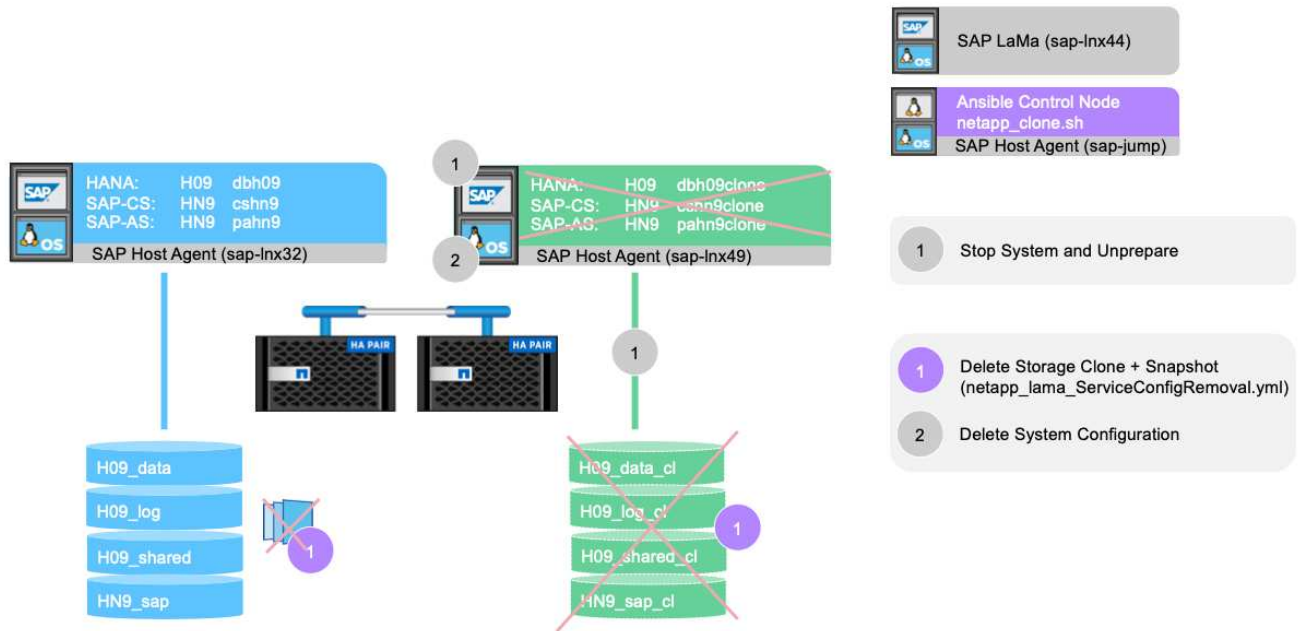
워크플로가 완료되면 복제된 SAP 시스템이 준비, 시작 및 사용할 준비가 됩니다.

SAP LaMa에서 워크플로우 프로비저닝 해제 - 시스템 폐기

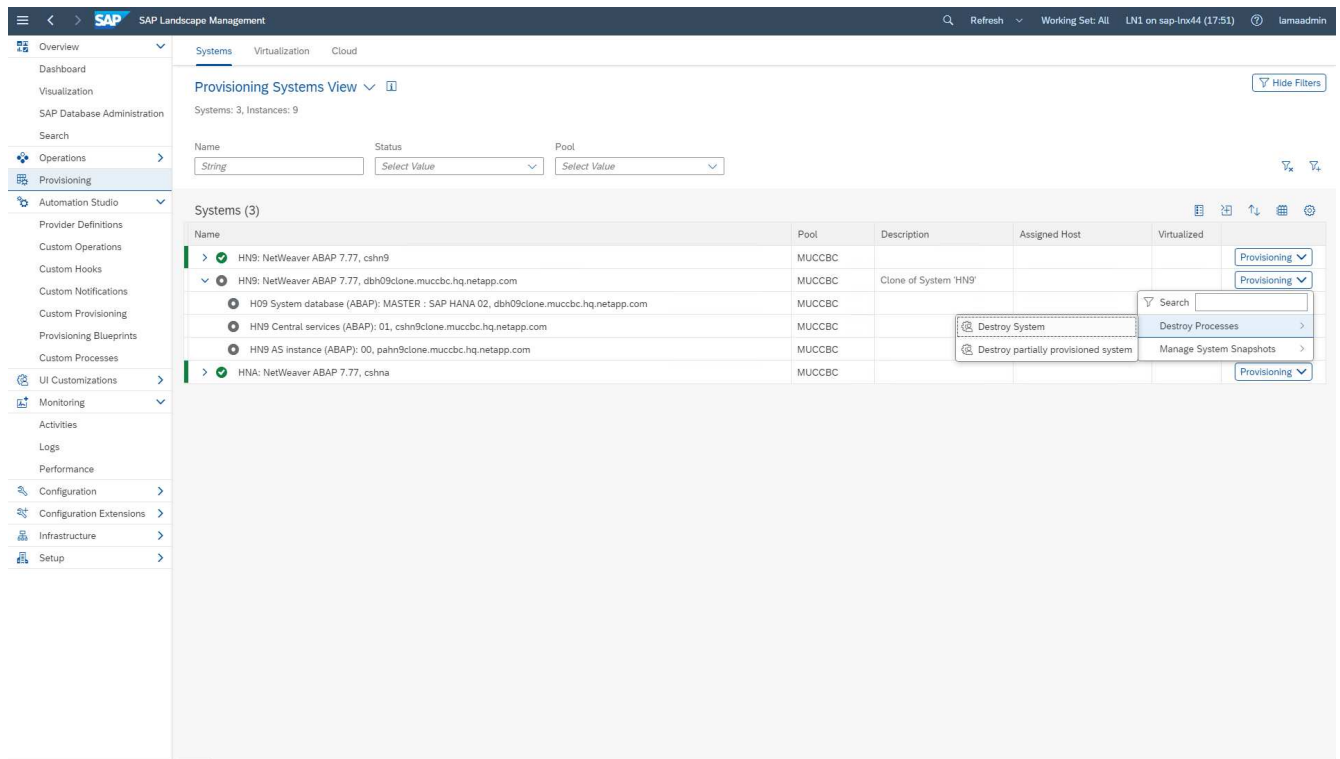
다음 그림에서는 시스템 제거 워크플로우와 함께 실행되는 주요 단계를 보여 줍니다.



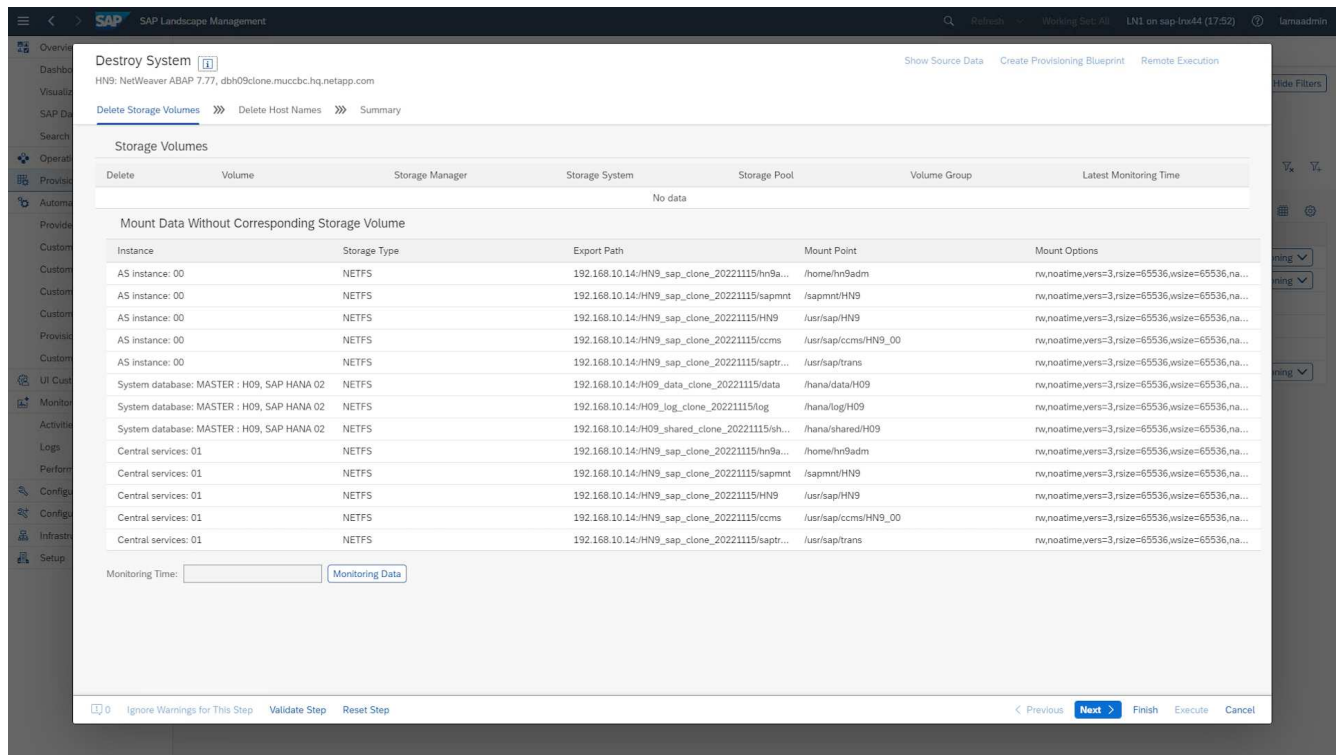
1. 클론 복제된 시스템을 해제하려면 먼저 중지하고 준비해야 합니다. 그런 다음 시스템 제거 워크플로우를 시작할 수 있습니다.



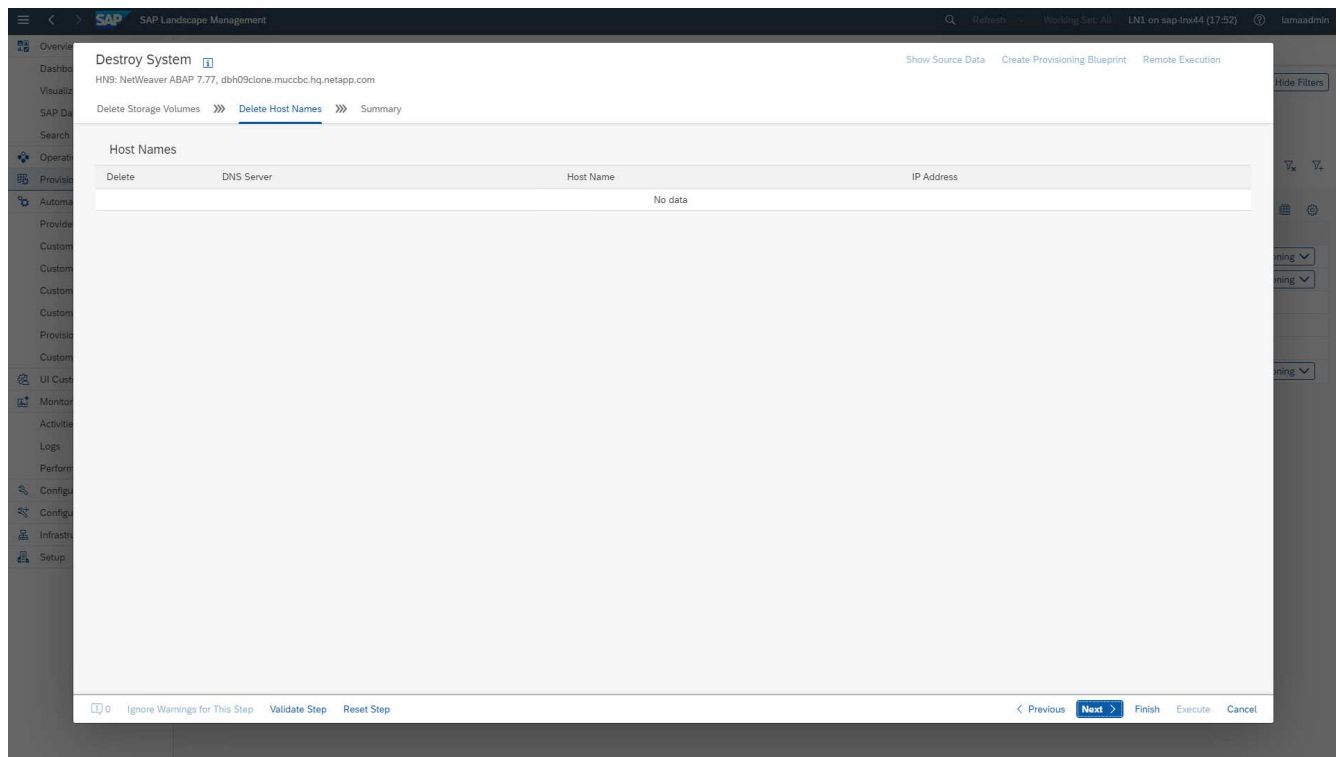
2. 이 예에서는 이전에 생성된 시스템에 대한 시스템 삭제 워크플로우를 실행합니다. System View * 화면에서 시스템을 선택하고 * Destroy Processes * 에서 시스템 삭제 워크플로우를 시작합니다.



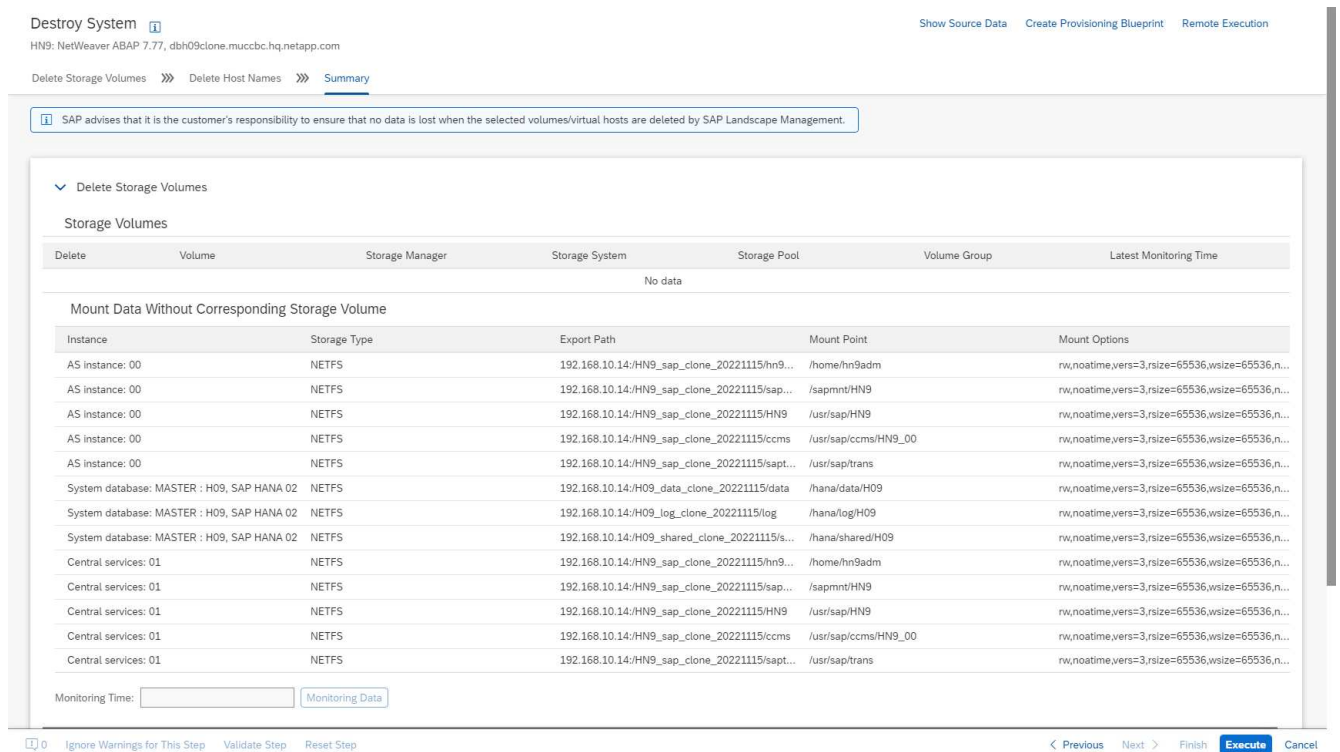
3. 프로비저닝 단계에서 유지 관리되는 모든 마운트 지점이 여기에 표시되며 시스템 제거 워크플로우 프로세스 중에 삭제됩니다.



가상 호스트 이름은 DNS를 통해 유지 관리되고 자동으로 할당되므로 삭제되지 않습니다.

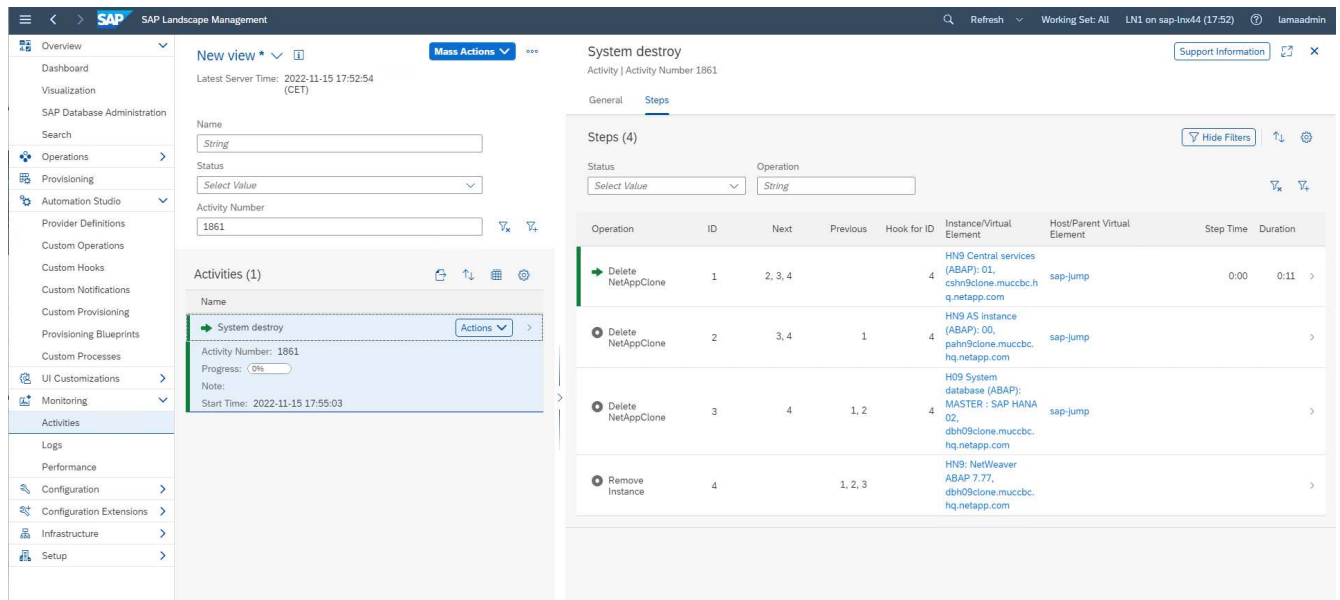


4. 실행 버튼을 클릭하면 작업이 시작됩니다.

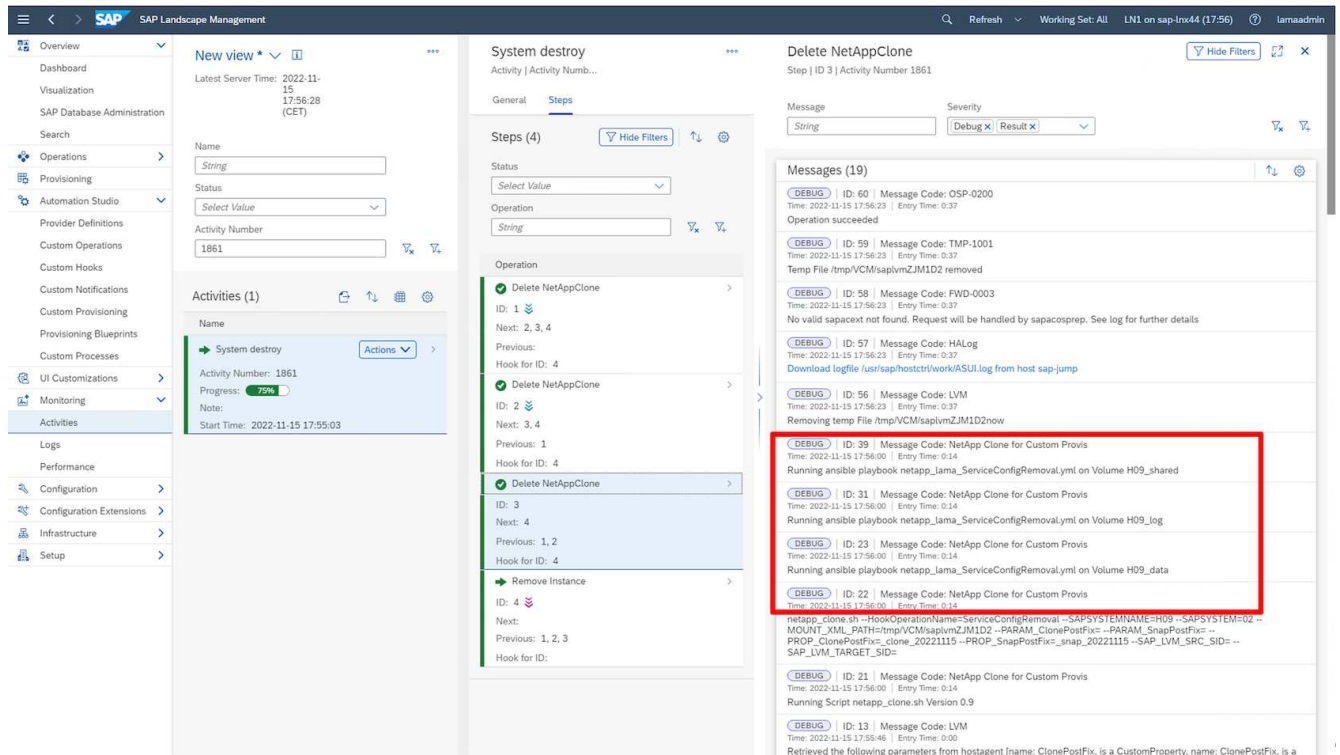


이제 SAP LaMa가 볼륨 클론을 삭제하고 복제된 시스템의 구성을 삭제합니다.

5. 모니터링 * 메뉴에서 클론 워크플로우의 진행 상황을 모니터링할 수 있습니다.

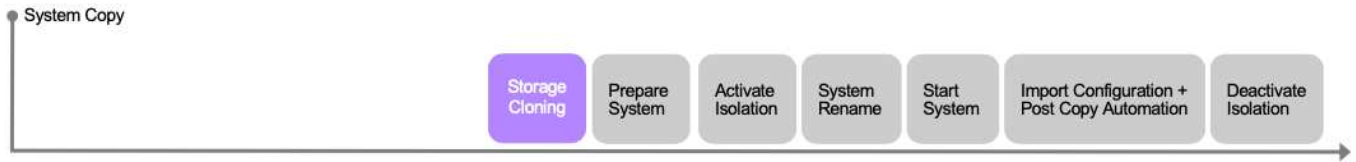


6. NetAppClone * 삭제 작업을 선택하면 해당 단계에 대한 자세한 로그가 표시됩니다. Ansible 플레이북을 실행하는 방법은 다음과 같습니다. 보시다시피 Ansible 플레이북도 있습니다
 netapp_lama_ServiceConfigRemoval.yml 각 HANA 데이터베이스 볼륨, 데이터, 로그 및 공유에 대해 실행됩니다.

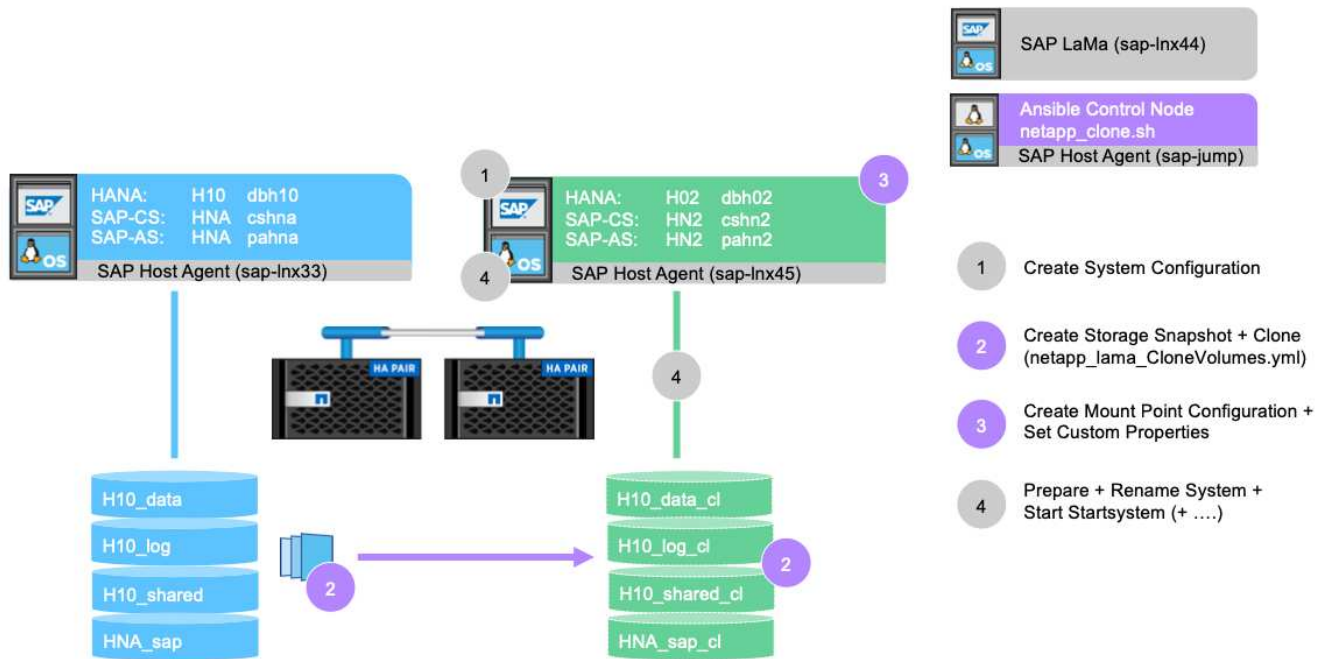


SAP LaMa가 워크플로우-복사 시스템을 프로비저닝합니다

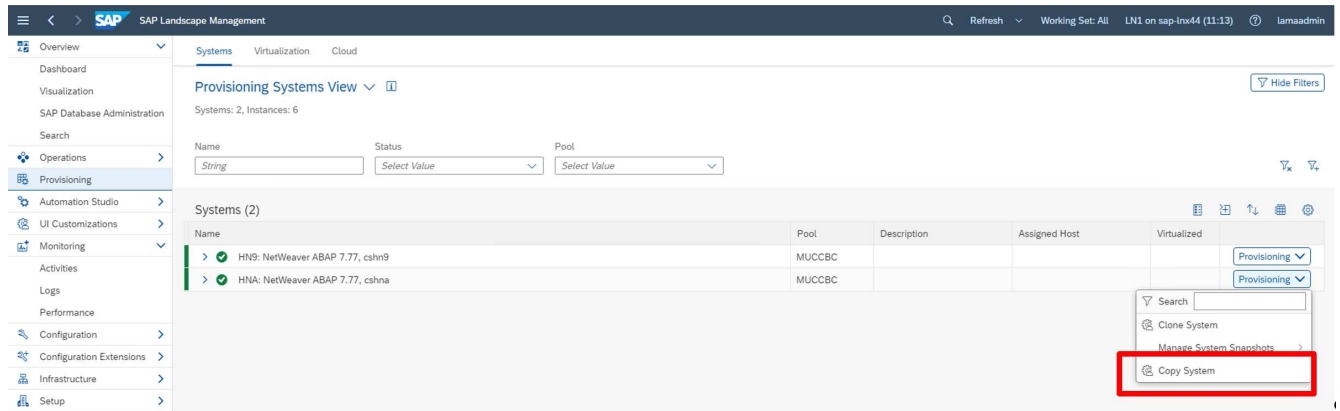
다음 그림에서는 시스템 복제 워크플로우에서 실행되는 기본 단계를 보여 줍니다.



이 장에서는 시스템 클론 워크플로우 및 입력 화면의 차이점에 대해 간략하게 설명합니다. 다음 이미지에서 볼 수 있듯이 스토리지 워크플로우에는 아무런 변화가 없습니다.



1. 시스템 복사 워크플로우는 시스템이 준비되면 시작할 수 있습니다. 이 작업은 이 구성에 대한 특정 작업이 아니며 자세한 내용은 설명하지 않습니다. 자세한 내용은 SAP LaMa 설명서를 참조하십시오.



2. 복사 작업 중에는 첫 번째 화면에서 지정해야 하는 것처럼 시스템 이름이 변경됩니다.

Copy System
HNA: NetWeaver ABAP 7.77, csna

Basic >>> Hosts >>> Host Names >>> Instance Number >>> Custom Clone >>> Consistency >>> Users >>> Rename >>> Isolation >>> ABAP PCA >>> Summary

Provide Basic Data for Target System

*System ID: HN2

☒ Use different Database Name

*HANA SID: H02

*Pool: MUCCBC

Description: Copy of System 'HNA'

Set Master Password for OS and DB Users

*Password: [Redacted]

*Confirm Password: [Redacted]

Ignore Warnings for This Step | Validate Step | Reset Step

< Previous | **Next** > | Finish | Execute | Cancel

3. 워크플로 중에 인스턴스 번호를 변경할 수 있습니다.

Copy System
HNA: NetWeaver ABAP 7.77, csna

Basic >>> Hosts >>> Host Names >>> **Instance Number** >>> Custom Clone >>> Consistency >>> Users >>> Rename >>> Isolation >>> ABAP PCA >>> Summary

SAP Instance Numbers

*System database: MASTER (configured) : SAP HANA 02
02

*AS instance: 00
00

*Central services: 01
01

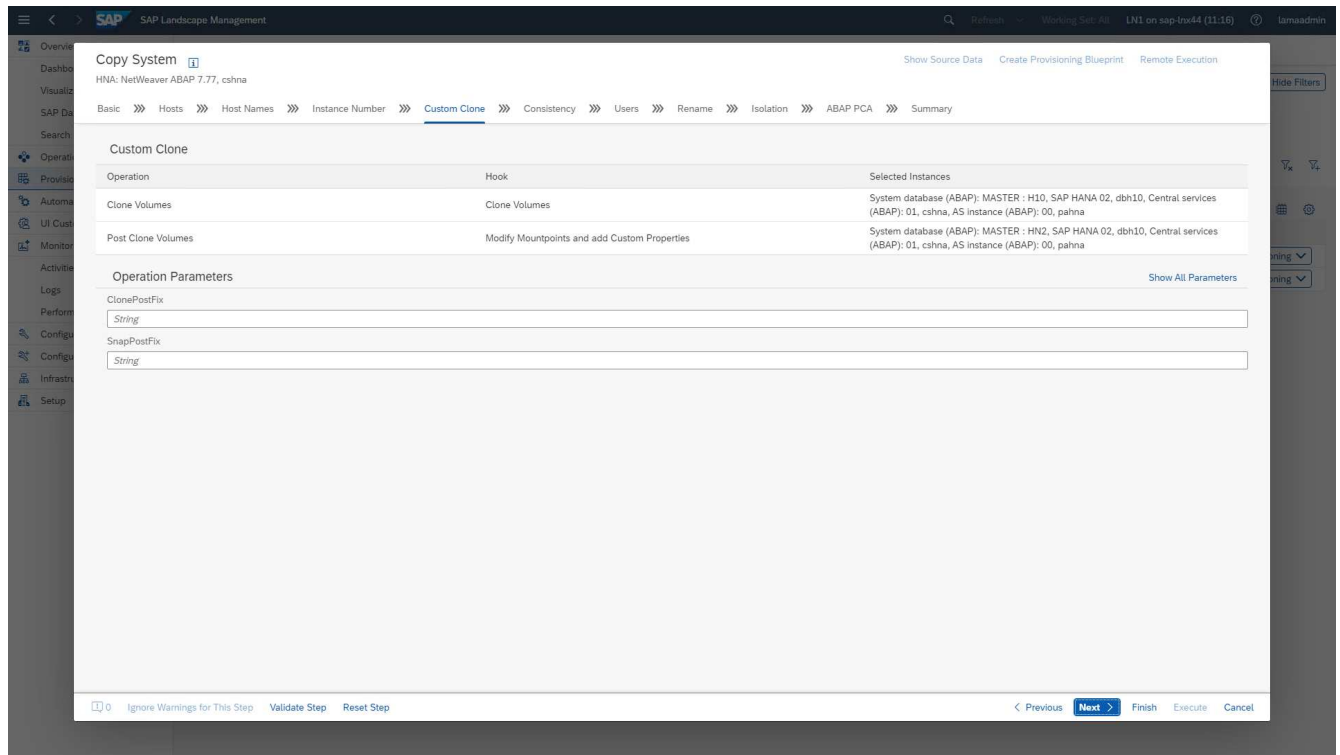
Ignore Warnings for This Step | Validate Step | Reset Step

< Previous | **Next** > | Finish | Execute | Cancel

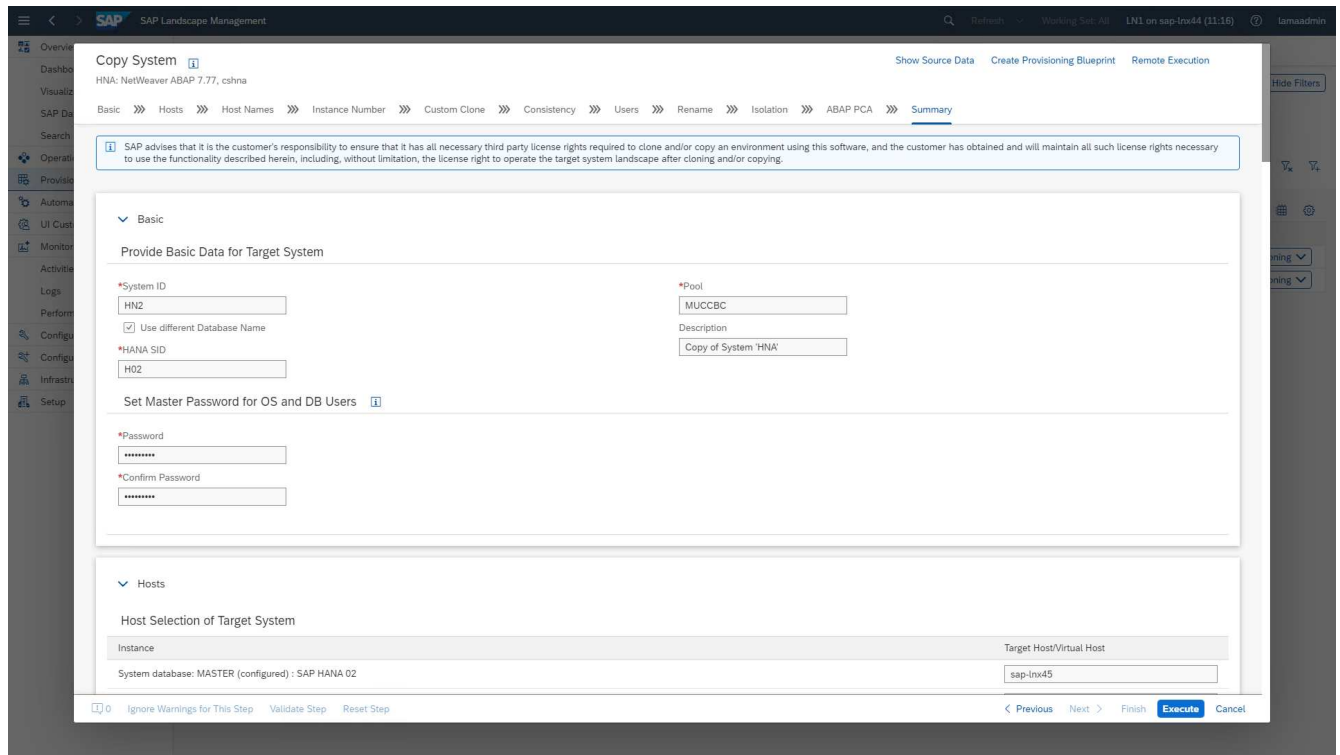


인스턴스 번호 변경은 테스트되지 않았으며 공급자 스크립트의 변경이 필요할 수 있습니다.

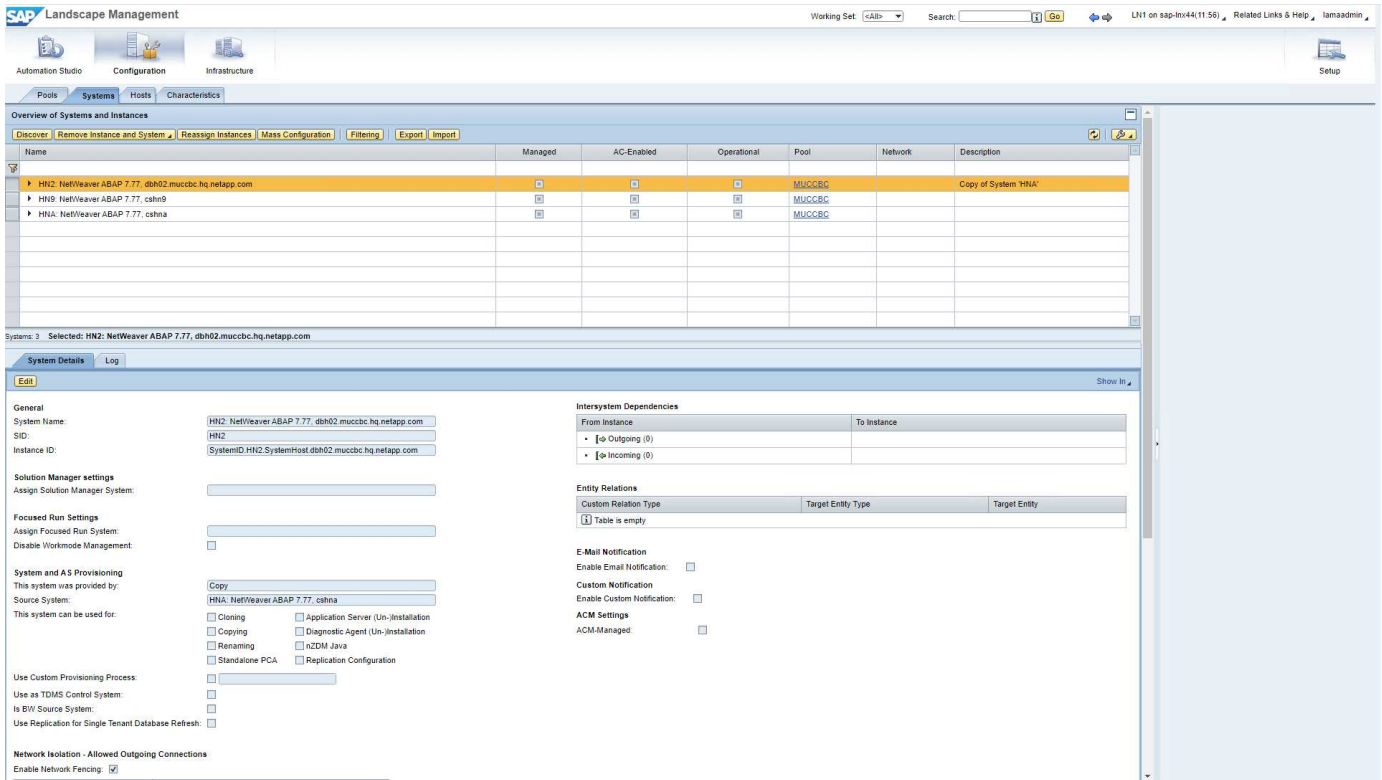
4. 설명과 같이 * 사용자 지정 클론 * 화면은 클론 생성 워크플로와 다르지 않습니다.



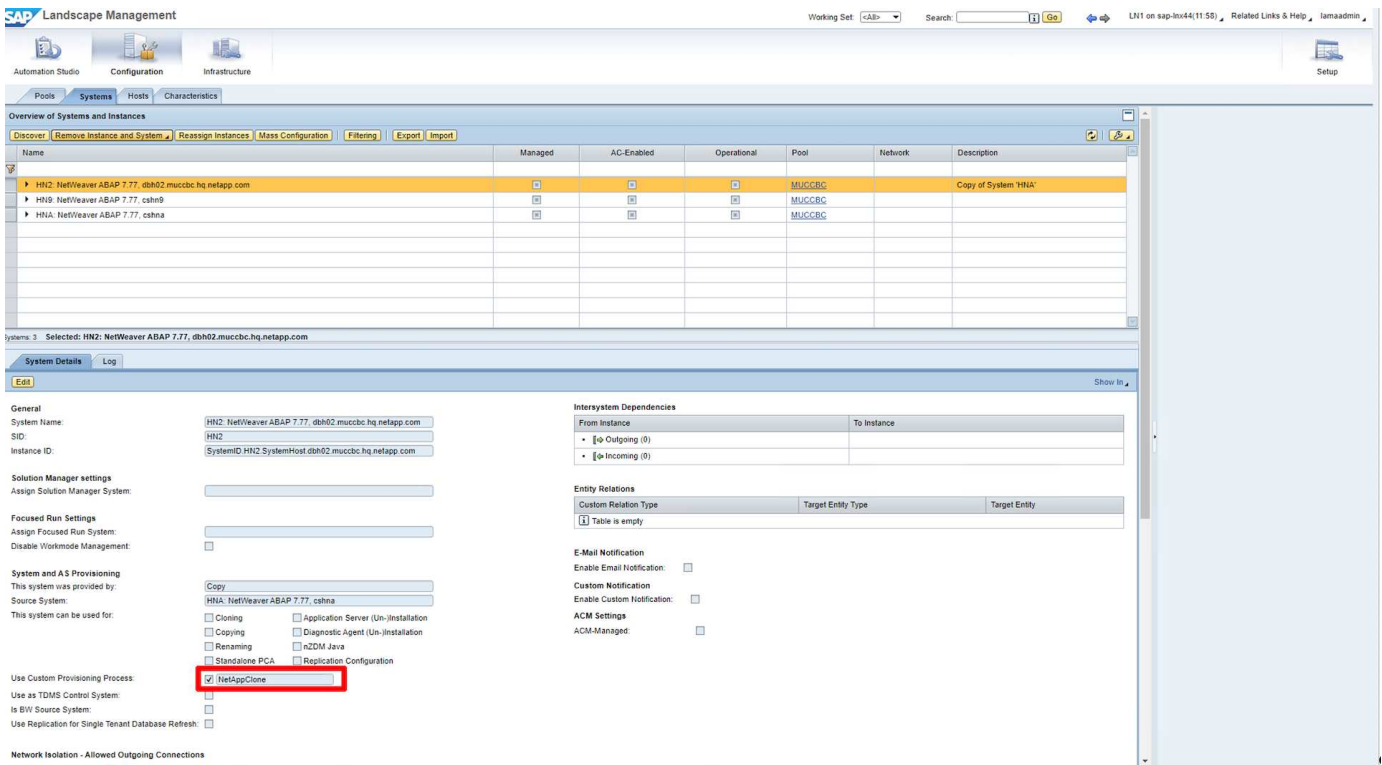
5. 이미 설명했듯이 나머지 입력 마스크는 표준에서 벗어나지 않으며 더 이상 여기에 넣지 않습니다. 최종 화면에는 요약이 표시되며 이제 실행을 시작할 수 있습니다.



복제 프로세스 후에는 타겟 인스턴스가 사용자 지정 클론 생성 프로세스에 대해 설정되지 않습니다.

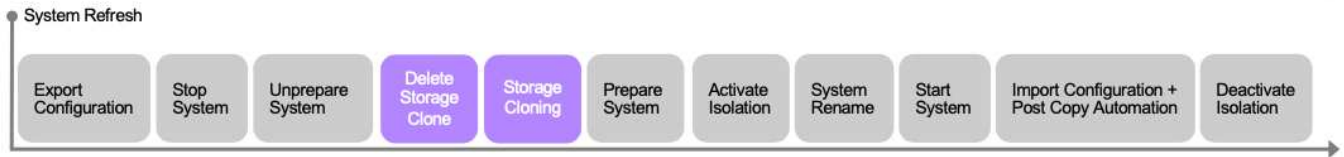


제약 조건이 설정되어 실행을 방해할 수 있으므로 시스템 폐기 프로세스 중에 사전 후크 단계를 실행하려면 수동으로 적용해야 합니다.

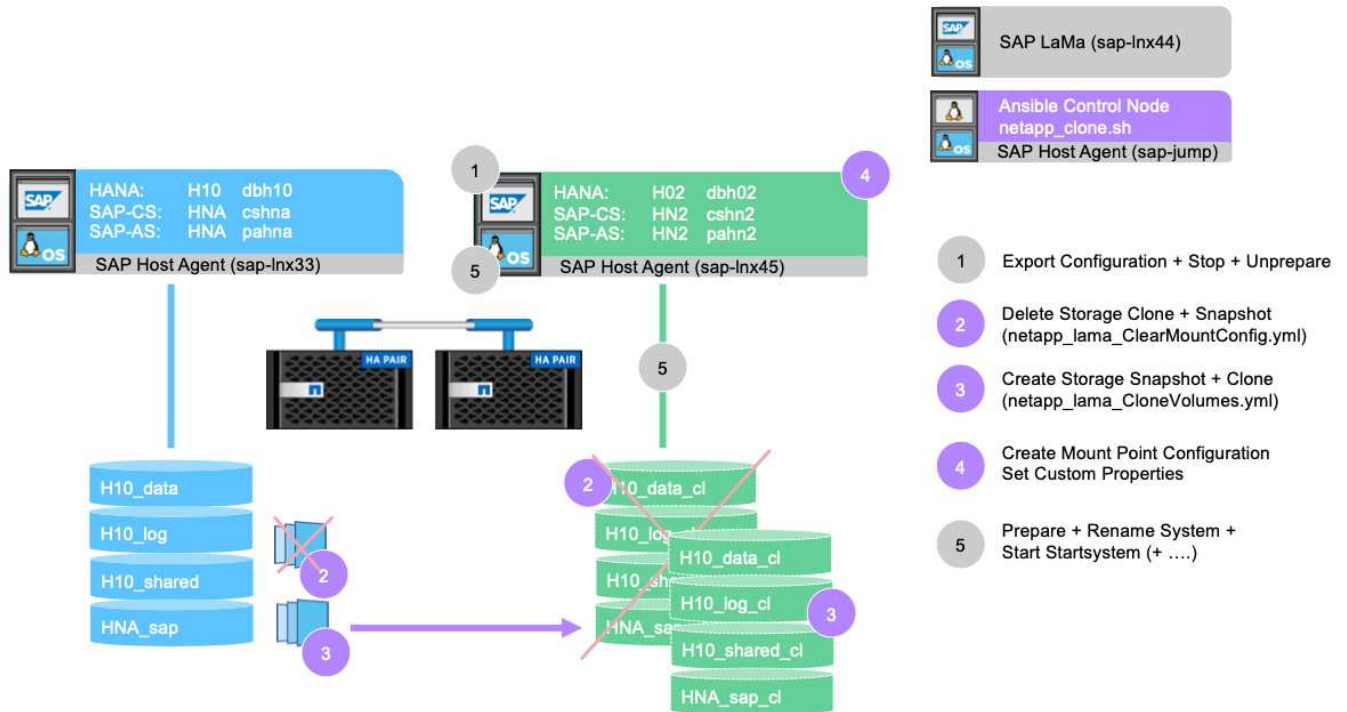


SAP LaMa 프로비저닝 워크플로우 - 시스템 업데이트

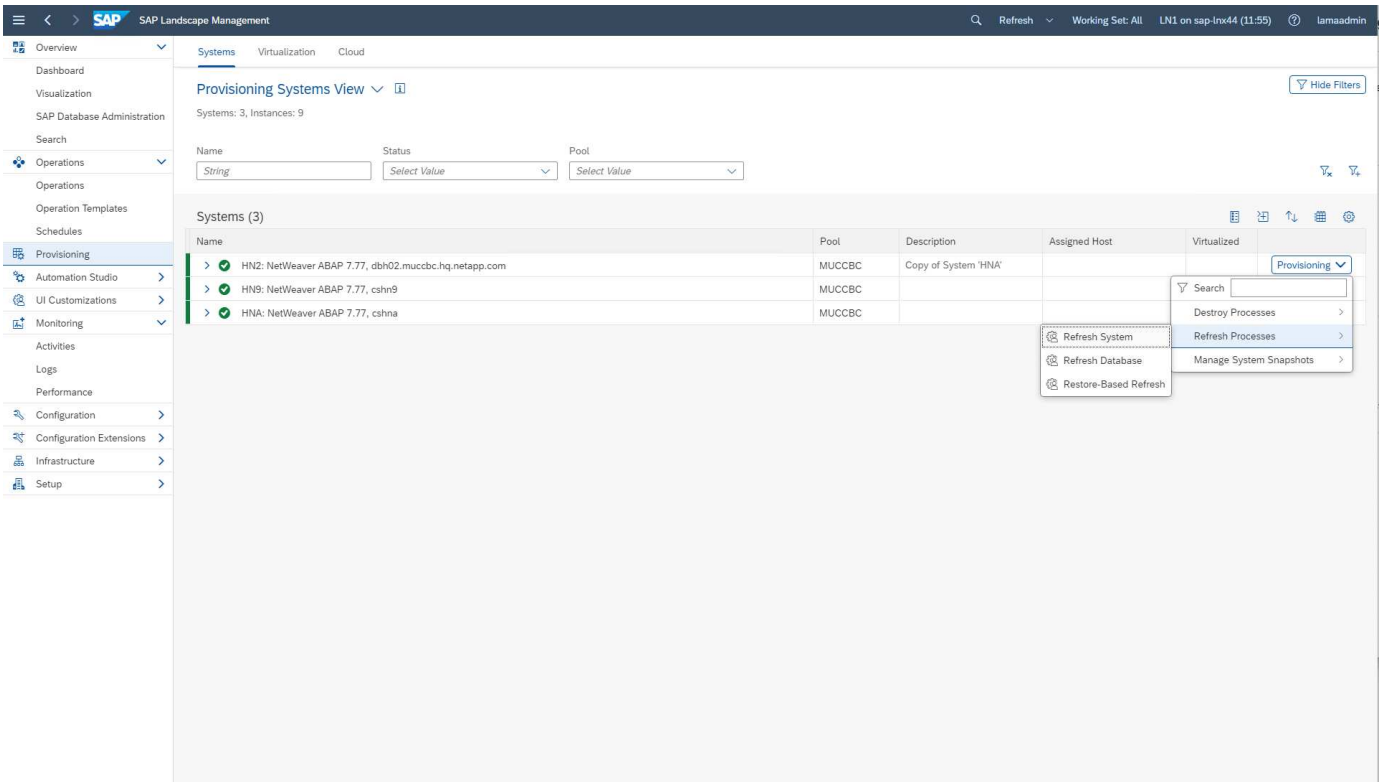
다음 그림에서는 시스템 새로 고침 워크플로우로 실행되는 주요 단계를 보여 줍니다.



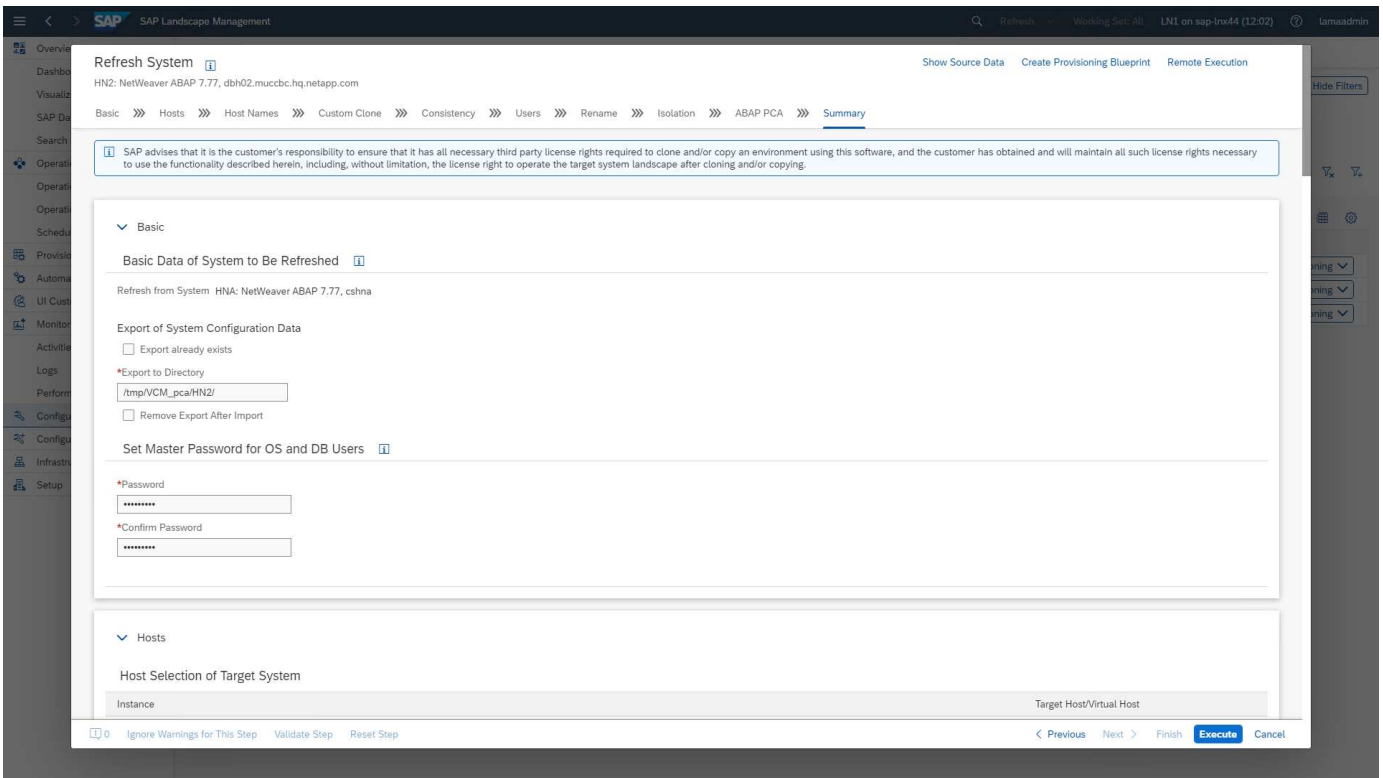
새로 고침 워크플로우 중에 스토리지 클론을 삭제해야 합니다. 시스템 폐기 워크플로우와 동일한 Ansible 플레이북을 사용할 수 있습니다. 그러나 맞춤형 후크는 다른 단계로 정의되므로 이에 따라 플레이북의 이름이 지정됩니다. 클론의 프로세스 단계가 다르게 적용되지 않습니다.



새로 고침 워크플로는 복제된 시스템에 대한 프로비저닝 화면을 통해 트리거될 수 있습니다.



다시 한 번 강조하지만 입력 화면과 표준에는 차이가 없으며, 요약 화면에서 워크플로우 실행을 시작할 수 있습니다.



공급자 스크립트 구성 및 Ansible 플레이북

이 문서에서는 샘플 배포 및 워크플로우 실행 중에 다음과 같은 공급자 구성 파일, 실행 스크립트

및 Ansible 플레이북을 사용합니다.



예제 스크립트는 있는 그대로 제공되며 NetApp에서 지원하지 않습니다. <mailto:ng-sapcc@netapp.com> [ng-sapcc@netapp.com]으로 이메일을 통해 스크립트의 현재 버전을 요청할 수 있습니다.

provider 구성 파일 **netapp_clone.conf**

구성 파일은 에 설명된 대로 생성됩니다 "[SAP LaMa 설명서 - SAP Host Agent 등록 스크립트 구성](#)". 이 구성 파일은 SAP 호스트 에이전트가 설치된 Ansible 제어 노드에 위치해야 합니다.

구성된 OS-사용자입니다 `sapuser` 스크립트와 소위 Ansible 플레이북을 실행할 수 있는 적절한 권한이 있어야 합니다. 스크립트를 일반 스크립트 디렉토리에 배치할 수 있습니다. SAP LaMa는 스크립트를 호출할 때 여러 매개 변수를 제공할 수 있습니다.

사용자 정의 매개변수 외에 `PARAM_ClonePostFix`, `PROP_ClonePostFix`, `PARAM_ClonePostFix`, 및 `'PROP_ClonePostFix'`에 나와 있는 것처럼 많은 다른 사람들이 인도될 수 있습니다 "[SAP LaMa 설명서](#)".

```
root@sap-jump:~# cat /usr/sap/hostctrl/exe/operations.d/netapp_clone.conf
Name: netapp_clone
Username: sapuser
Description: NetApp Clone for Custom Provisioning
Command: /usr/sap/scripts/netapp_clone.sh
--HookOperationName=${HookOperationName} --SAPSYSTEMNAME=${SAPSYSTEMNAME}
--SAPSYSTEM=${SAPSYSTEM} --MOUNT_XML_PATH=${MOUNT_XML_PATH}
--PARAM_ClonePostFix=${PARAM_ClonePostFix} --PARAM_SnapPostFix=${PARAM_SnapPostFix}
--PROP_ClonePostFix=${PROP_ClonePostFix}
--PROP_SnapPostFix=${PROP_SnapPostFix}
--SAP_LVM_SRC_SID=${SAP_LVM_SRC_SID}
--SAP_LVM_TARGET_SID=${SAP_LVM_TARGET_SID}
ResulConverter: hook
Platform: Unix
```

공급자 스크립트 **netapp_clone.sh**

공급자 스크립트는 에 저장해야 합니다 `/usr/sap/scripts` 공급자 구성 파일에 구성되어 있습니다.

변수

다음 변수는 스크립트에 하드 코딩되어 있으며 그에 따라 수정해야 합니다.

- `PRIMARY_CLUSTER=<hostname of netapp cluster>`
- `PRIMARY_SVM=<SVM name where source system volumes are stored>`

인증서 파일입니다 `PRIMARY_KEYFILE=/usr/sap/scripts/ansible/certs/ontap.key` 및 `PRIMARY_CERTFILE=/usr/sap/scripts/ansible/certs/ontap.pem` 에 설명된 대로 제공해야 합니다 "[NetApp Ansible 모듈 - ONTAP을 준비합니다](#)".



서로 다른 SAP 시스템에 서로 다른 클러스터 또는 SVM이 필요한 경우 이러한 변수를 SAP LaMa 공급자 정의의 매개 변수로 추가할 수 있습니다.

Function(기능): 재고 파일을 생성합니다

Ansible 플레이북을 더 동적이며 `inventory.yml` 파일이 즉시 생성됩니다. 일부 정적 값은 변수 섹션에서 구성되며, 일부는 실행 중에 동적으로 생성됩니다.

기능: Ansible 플레이북을 실행하십시오

이 기능은 동적으로 생성된 과 함께 Ansible 플레이북을 실행하는 데 사용됩니다 `inventory.yml` 파일. Playbook의 명령 규칙은입니다 `netapp_lama_{{HookOperationName}}.yml`. 의 값 `{{HookOperationName}}` LaMa 운용에 의존하며 LaMa에 의해 명령행 파라미터로 전달된다.

섹션 기본

이 섹션에서는 기본 실행 계획을 다룹니다. 변수 `{{HookOperationName}}` LAMA 교체 단계의 이름이 포함되어 있으며 스크립트를 호출할 때 LaMa에서 제공됩니다.

- 시스템 클론 및 시스템 복사본 프로비저닝 워크플로와 관련된 값:
 - 볼륨
 - `PostCloneVolumes`
- 시스템이 워크플로를 소멸시키는 값:
 - `ServiceConfigRemoval`
- 시스템 새로 고침 워크플로의 값:
 - `ClearMountConfig`를 클릭합니다

HookOperationName = CloneVolumes입니다

이 단계에서는 Ansible 플레이북을 실행하여 스냅샷 복사본 및 클론 복제 작업을 트리거합니다. 볼륨 이름 및 마운트 구성은 변수에 정의된 XML 파일을 통해 SAP LaMa에 의해 전달됩니다 `$MOUNT_XML_PATH`. 이 파일은 나중에 단계에서 사용되므로 저장됩니다 `FinalizeCloneVolumes` 를 클릭하여 새 마운트 지점 구성을 생성합니다. 볼륨 이름은 XML 파일에서 추출되고 Ansible 클론 플레이북은 각 볼륨에 대해 실행됩니다.



이 예에서 AS 인스턴스와 중앙 서비스는 동일한 볼륨을 공유합니다. 따라서 볼륨 클론 복제는 SAP 인스턴스 번호에서만 실행됩니다 (`$SAPSYSTEM`)은(는) 아닙니다 01. 이는 다른 환경에 따라 다를 수 있으므로 그에 따라 변경해야 합니다.

HookOperationName = PostCloneVolumes

이 단계에서 사용자 정의 속성이 표시됩니다 `ClonePostFix` 및 `SnapPostFix` 타겟 시스템의 마운트 지점 구성이 유지됩니다.

사용자 정의 속성은 나중에 이 과정에서 시스템을 해체할 때 입력으로 사용됩니다 `ServiceConfigRemoval` 또는 `ClearMountConfig` 단계. 시스템은 시스템 프로비저닝 워크플로우 중에 지정된 사용자 지정 매개 변수의 설정을 보존하도록 설계되었습니다.

이 예제에서 사용되는 값은입니다 `ClonePostFix=_clone_20221115` 및 `SnapPostFix=_snap_20221115`.

를 선택합니다 HN9_sap`동적으로 생성된 Ansible 파일에는 다음 값이 포함됩니다. `datavolumename: HN9_sap, snapshotpostfix: _snap_20221115, 및 clonepostfix: _clone_20221115.

이 경우 볼륨 HN9_SAP의 스냅샷 이름으로 연결됩니다 HN9_sap_snap_20221115 생성된 볼륨 클론 이름입니다 HN9_sap_clone_20221115.



사용자 지정 속성은 프로비저닝 프로세스 중에 사용되는 매개 변수를 보존하기 위해 어떤 방법으로든 사용할 수 있습니다.

마운트 지점 구성은 에서 LaMa가 넘겨준 XML 파일에서 추출됩니다 CloneVolume 단계. 를 클릭합니다 ClonePostFix 볼륨 이름에 추가되고 기본 스크립트 출력을 통해 LaMa로 다시 전송됩니다. 기능에 대한 설명은 에 나와 있습니다 "[SAP 참고 1889590](#)".



이 예에서는 스토리지 시스템의 qtree가 단일 볼륨에 여러 데이터를 배치하는 일반적인 방법으로 사용됩니다. 예를 들면, 다음과 같습니다. HN9_sap 의 마운트 지점을 보유합니다 /usr/sap/HN9, /sapmnt/HN9, 및 /home/hn9adm. 하위 디렉터리는 같은 방식으로 작동합니다. 이는 다른 환경에 따라 다를 수 있으므로 그에 따라 변경해야 합니다.

HookOperationName = ServiceConfigRemoval

이 단계에서는 볼륨 클론을 삭제하는 역할을 하는 Ansible 플레이북을 실행합니다.

볼륨 이름은 마운트 구성 파일 및 사용자 지정 속성을 통해 SAP LaMa에서 전달됩니다 ClonePostFix 및 SnapPostFix 시스템 프로비저닝 워크플로우 중에 원래 지정된 매개 변수의 값을 넘기는 데 사용됩니다(의 참고 사항 참조) HookOperationName = PostCloneVolumes)를 클릭합니다.

볼륨 이름은 XML 파일에서 추출되고, Ansible 클론 플레이북은 각 볼륨에 대해 실행됩니다.



이 예에서 AS 인스턴스와 중앙 서비스는 동일한 볼륨을 공유합니다. 따라서 볼륨 삭제는 SAP 인스턴스 번호에서만 실행됩니다 (\$SAPSYSTEM)은(는) 아닙니다 01. 이는 다른 환경에 따라 다를 수 있으므로 그에 따라 변경해야 합니다.

HookOperationName = ClearMountConfig입니다

이 단계에서는 시스템 업데이트 워크플로우 중에 볼륨 클론을 삭제하는 역할을 하는 Ansible 플레이북을 실행합니다.

볼륨 이름은 마운트 구성 파일 및 사용자 지정 속성을 통해 SAP LaMa에서 전달됩니다 ClonePostFix 및 SnapPostFix 시스템 프로비저닝 워크플로우 중에 원래 지정된 매개 변수의 값을 넘기는 데 사용됩니다.

볼륨 이름은 XML 파일에서 추출되고 Ansible 클론 플레이북은 각 볼륨에 대해 실행됩니다.



이 예에서 AS 인스턴스와 중앙 서비스는 동일한 볼륨을 공유합니다. 따라서 볼륨 삭제는 SAP 인스턴스 번호에서만 실행됩니다 (\$SAPSYSTEM)은(는) 아닙니다 01. 이는 다른 환경에 따라 다를 수 있으므로 그에 따라 변경해야 합니다.

```
root@sap-jump:~# cat /usr/sap/scripts/netapp_clone.sh
#!/bin/bash
#Section - Variables
#####
```



```

VERSION="Version 0.9"
#Path for ansible play-books
ANSIBLE_PATH=/usr/sap/scripts/ansible
#Values for Ansible Inventory File
PRIMARY_CLUSTER=grenada
PRIMARY_SVM=svm-sap01
PRIMARY_KEYFILE=/usr/sap/scripts/ansible/certs/ontap.key
PRIMARY_CERTFILE=/usr/sap/scripts/ansible/certs/ontap.pem
#Default Variable if PARAM ClonePostFix / SnapPostFix is not maintained in
LaMa
DefaultPostFix=_clone_1
#TMP Files - used during execution
YAML_TMP=/tmp/inventory_ansible_clone_tmp_$$$.yaml
TMPFILE=/tmp/tmpfile. $$
MY_NAME="`basename $0`"
BASE_SCRIPT_DIR="`dirname $0`"
#Sendig Script Version and run options to LaMa Log
echo "[DEBUG]: Running Script $MY_NAME $VERSION"
echo "[DEBUG]: $MY_NAME @$"
#Command declared in the netapp_clone.conf Provider definition
#Command: /usr/sap/scripts/netapp_clone.sh
--HookOperationName=${HookOperationName} --SAPSYSTEMNAME=${SAPSYSTEMNAME}
--SAPSYSTEM=${SAPSYSTEM} --MOUNT_XML_PATH=${MOUNT_XML_PATH}
--PARAM_ClonePostFix=${PARAM_ClonePostFix} --PARAM_SnapPostFix=${PARAM
-SnapPostFix} --PROP_ClonePostFix=${PROP_ClonePostFix}
--PROP_SnapPostFix=${PROP_SnapPostFix}
--SAP_LVM_SRC_SID=${SAP_LVM_SRC_SID}
--SAP_LVM_TARGET_SID=${SAP_LVM_TARGET_SID}
#Reading Input Variables hand over by LaMa
for i in "$@"
do
case $i in
--HookOperationName=*)
HookOperationName="${i#*=}";shift;;
--SAPSYSTEMNAME=*)
SAPSYSTEMNAME="${i#*=}";shift;;
--SAPSYSTEM=*)
SAPSYSTEM="${i#*=}";shift;;
--MOUNT_XML_PATH=*)
MOUNT_XML_PATH="${i#*=}";shift;;
--PARAM_ClonePostFix=*)
PARAM_ClonePostFix="${i#*=}";shift;;
--PARAM_SnapPostFix=*)
PARAM_SnapPostFix="${i#*=}";shift;;
--PROP_ClonePostFix=*)
PROP_ClonePostFix="${i#*=}";shift;;

```

```

--PROP_SnapPostFix=*)
PROP_SnapPostFix="${i#*=}";shift;;
--SAP_LVM_SRC_SID=*)
SAP_LVM_SRC_SID="${i#*=}";shift;;
--SAP_LVM_TARGET_SID=*)
SAP_LVM_TARGET_SID="${i#*=}";shift;;
*)
# unknown option
;;
esac
done
#If Parameters not provided by the User - defaulting to DefaultPostFix
if [ -z $PARAM_ClonePostFix ]; then PARAM_ClonePostFix=$DefaultPostFix;fi
if [ -z $PARAM_SnapPostFix ]; then PARAM_SnapPostFix=$DefaultPostFix;fi
#Section - Functions
#####
#Function Create (Inventory) YAML File
#####
create_yaml_file()
{
echo "ontapservers:">$YAML_TMP
echo " hosts:">>$YAML_TMP
echo "   ${PRIMARY_CLUSTER}:">>$YAML_TMP
echo "   ansible_host: "'"${PRIMARY_CLUSTER}"'">>$YAML_TMP
echo "   keyfile: "'"${PRIMARY_KEYFILE}"'">>$YAML_TMP
echo "   certfile: "'"${PRIMARY_CERTFILE}"'">>$YAML_TMP
echo "   svmname: "'"${PRIMARY_SVM}"'">>$YAML_TMP
echo "   datavolumename: "'"${datavolumename}"'">>$YAML_TMP
echo "   snapshotpostfix: "'"${snapshotpostfix}"'">>$YAML_TMP
echo "   clonepostfix: "'"${clonepostfix}"'">>$YAML_TMP
}
#Function run ansible-playbook
#####
run_ansible_playbook()
{
echo "[DEBUG]: Running ansible playbook
netapp_lama_${HookOperationName}.yaml on Volume $datavolumename"
ansible-playbook -i $YAML_TMP
$ANSIBLE_PATH/netapp_lama_${HookOperationName}.yaml
}
#Section - Main
#####
#HookOperationName - CloneVolumes
#####
if [ $HookOperationName = CloneVolumes ] ;then
#save mount xml for later usage - used in Section FinalizeCloneVolumes to

```

```

generate the mountpoints
echo "[DEBUG]: saving mount config...."
cp $MOUNT_XML_PATH /tmp/mount_config_${SAPSYSTEMNAME}_${SAPSYSTEM}.xml
#Instance 00 + 01 share the same volumes - clone needs to be done once
if [ $SAPSYSTEM != 01 ]; then
#generating Volume List - assuming usage of qtrees - "IP-
Adress:/VolumeName/mtree"
xmlFile=/tmp/mount_config_${SAPSYSTEMNAME}_${SAPSYSTEM}.xml
if [ -e $TMPFILE ];then rm $TMPFILE;fi
numMounts=`xml_grep --count "/mountconfig/mount" $xmlFile | grep "total: "
| awk '{ print $2 }'`
i=1
while [ $i -le $numMounts ]; do
    xmllint --xpath "/mountconfig/mount[$i]/exportpath/text()" $xmlFile
|awk -F"/" '{print $2}' >>$TMPFILE
i=$((i + 1))
done
DATAVOLUMES=`cat $TMPFILE |sort -u`
#Create yml file and rund playbook for each volume
for I in $DATAVOLUMES; do
datavolumename="$I"
snapshotpostfix="$PARAM_SnapPostFix"
clonepostfix="$PARAM_ClonePostFix"
create_yml_file
run_ansible_playbook
done
else
echo "[DEBUG]: Doing nothing .... Volume cloned in different Task"
fi
fi
#HookOperationName - PostCloneVolumes
#####
if [ $HookOperationName = PostCloneVolumes] ;then
#Reporting Properties back to LaMa Config for Cloned System
echo "[RESULT]:Property:ClonePostFix=$PARAM_ClonePostFix"
echo "[RESULT]:Property:SnapPostFix=$PARAM_SnapPostFix"
#Create MountPoint Config for Cloned Instances and report back to LaMa
according to SAP Note: https://launchpad.support.sap.com/#/notes/1889590
echo "MountDataBegin"
echo '<?xml version="1.0" encoding="UTF-8"?>'
echo "<mountconfig>"
xmlFile=/tmp/mount_config_${SAPSYSTEMNAME}_${SAPSYSTEM}.xml
numMounts=`xml_grep --count "/mountconfig/mount" $xmlFile | grep "total: "
| awk '{ print $2 }'`
i=1
while [ $i -le $numMounts ]; do

```

```

MOUNTPPOINT=`xmllint --xpath "/mountconfig/mount[$i]/mountpoint/text()"
$xmlFile`;
    EXPORTPATH=`xmllint --xpath
"/mountconfig/mount[$i]/exportpath/text()" $xmlFile`;
    OPTIONS=`xmllint --xpath "/mountconfig/mount[$i]/options/text()"
$xmlFile`;
#Adopt Exportpath and add Clonepostfix - assuming usage of qtrees - "IP-
Adress:/VolumeName/qtrees"
TMPFIELD1=`echo $EXPORTPATH|awk -F"/" '{print $1}'`
TMPFIELD2=`echo $EXPORTPATH|awk -F"/" '{print $2}'`
TMPFIELD3=`echo $EXPORTPATH|awk -F"/" '{print $3}'`
EXPORTPATH=$TMPFIELD1":/`${TMPFIELD2}$PARAM_ClonePostFix"/`${TMPFIELD3
echo -e '\t<mount fstype="nfs" storagetype="NETFS">'
echo -e "\t\t<mountpoint>${MOUNTPPOINT}</mountpoint>"
echo -e "\t\t<exportpath>${EXPORTPATH}</exportpath>"
echo -e "\t\t<options>${OPTIONS}</options>"
echo -e "\t</mount>"
i=$((i + 1))
done
echo "</mountconfig>"
echo "MountDataEnd"
#Finished MountPoint Config
#Cleanup Temporary Files
rm $xmlFile
fi
#HookOperationName - ServiceConfigRemoval
#####
if [ $HookOperationName = ServiceConfigRemoval ] ;then
#Assure that Properties ClonePostFix and SnapPostfix has been configured
through the provisioning process
if [ -z $PROP_ClonePostFix ]; then echo "[ERROR]: Propertiy ClonePostFix
is not handed over - please investigate";exit 5;fi
if [ -z $PROP_SnapPostFix ]; then echo "[ERROR]: Propertiy SnapPostFix is
not handed over - please investigate";exit 5;fi
#Instance 00 + 01 share the same volumes - clone delete needs to be done
once
if [ $SAPSYSTEM != 01 ]; then
#generating Volume List - assuming usage of qtrees - "IP-
Adress:/VolumeName/qtrees"
xmlFile=$MOUNT_XML_PATH
if [ -e $TMPFILE ];then rm $TMPFILE;fi
numMounts=`xml_grep --count "/mountconfig/mount" $xmlFile | grep "total: "
| awk '{ print $2 }'`
i=1
while [ $i -le $numMounts ]; do
    xmllint --xpath "/mountconfig/mount[$i]/exportpath/text()" $xmlFile

```

```

|awk -F"/" '{print $2}' >>$TMPFILE
i=$((i + 1))
done
DATAVOLUMES=`cat $TMPFILE |sort -u| awk -F $PROP_ClonePostFix '{ print $1
}'`
#Create yaml file and rund playbook for each volume
for I in $DATAVOLUMES; do
datavolumename="$I"
snapshotpostfix="$PROP_SnapPostFix"
clonepostfix="$PROP_ClonePostFix"
create_yaml_file
run_ansible_playbook
done
else
echo "[DEBUG]: Doing nothing .... Volume deleted in different Task"
fi
#Cleanup Temporary Files
rm $xmlFile
fi
#HookOperationName - ClearMountConfig
#####
if [ $HookOperationName = ClearMountConfig ] ;then
    #Assure that Properties ClonePostFix and SnapPostfix has been
    configured through the provisioning process
    if [ -z $PROP_ClonePostFix ]; then echo "[ERROR]: Property
ClonePostFix is not handed over - please investigate";exit 5;fi
    if [ -z $PROP_SnapPostFix ]; then echo "[ERROR]: Property
SnapPostFix is not handed over - please investigate";exit 5;fi
    #Instance 00 + 01 share the same volumes - clone delete needs to
    be done once
    if [ $SAPSYSTEM != 01 ]; then
        #generating Volume List - assuming usage of qtrees - "IP-
        Adress:/VolumeName/qtrees"
        xmlFile=$MOUNT_XML_PATH
        if [ -e $TMPFILE ];then rm $TMPFILE;fi
        numMounts=`xml_grep --count "/mountconfig/mount" $xmlFile
| grep "total: " | awk '{ print $2 }'`
        i=1
        while [ $i -le $numMounts ]; do
            xmllint --xpath
            "/mountconfig/mount[$i]/exportpath/text()" $xmlFile |awk -F"/" '{print
            $2}' >>$TMPFILE
            i=$((i + 1))
        done
        DATAVOLUMES=`cat $TMPFILE |sort -u| awk -F
        $PROP_ClonePostFix '{ print $1 }'`

```

```

        #Create yml file and rund playbook for each volume
        for I in $DATAVOLUMES; do
            datavolumename="$I"
            snapshotpostfix="$PROP_SnapPostFix"
            clonepostfix="$PROP_ClonePostFix"
            create_yml_file
            run_ansible_playbook

        done

    else
        echo "[DEBUG]: Doing nothing .... Volume deleted in
different Task"
    fi
    #Cleanup Temporary Files
    rm $xmlFile
fi
#Cleanup
#####
#Cleanup Temporary Files
if [ -e $TMPFILE ];then rm $TMPFILE;fi
if [ -e $YAML_TMP ];then rm $YAML_TMP;fi
exit 0

```

Ansible 플레이북 NetApp_LaMa_CloneVolumes.yml

LaMa 시스템 클론 워크플로우의 CloneVolumes 단계에서 실행되는 플레이북은 의 조합입니다 create_snapshot.yml 및 create_clone.yml (을 참조하십시오 ["NetApp Ansible 모듈 - YAML 파일"](#))를 클릭합니다. 이 플레이북은 보조 및 클론 분할 작업의 클론 복제와 같은 추가 사용 사례를 쉽게 포함할 수 있습니다.

```

root@sap-jump:~# cat /usr/sap/scripts/ansible/netapp_lama_CloneVolumes.yml
---
- hosts: ontapservers
  connection: local
  collections:
    - netapp.ontap
  gather_facts: false
  name: netapp_lama_CloneVolumes
  tasks:
    - name: Create SnapShot
      na_ontap_snapshot:
        state: present
        snapshot: "{{ datavolumename }}{{ snapshotpostfix }}"
        use_rest: always
        volume: "{{ datavolumename }}"
        vserver: "{{ svmname }}"
        hostname: "{{ inventory_hostname }}"
        cert_filepath: "{{ certfile }}"
        key_filepath: "{{ keyfile }}"
        https: true
        validate_certs: false
    - name: Clone Volume
      na_ontap_volume_clone:
        state: present
        name: "{{ datavolumename }}{{ clonepostfix }}"
        use_rest: always
        vserver: "{{ svmname }}"
        junction_path: '/{{ datavolumename }}{{ clonepostfix }}'
        parent_volume: "{{ datavolumename }}"
        parent_snapshot: "{{ datavolumename }}{{ snapshotpostfix }}"
        hostname: "{{ inventory_hostname }}"
        cert_filepath: "{{ certfile }}"
        key_filepath: "{{ keyfile }}"
        https: true
        validate_certs: false

```

Ansible 플레이북 NetApp_LaMa_ServiceConfigRemoval.yml

동안 실행되는 플레이북 ServiceConfigRemoval LaMa 시스템 제거 워크플로의 단계는 의 조합입니다 delete_clone.yml 및 delete_snapshot.yml (을 참조하십시오 ["NetApp Ansible 모듈 - YAML 파일"](#))를 클릭합니다. 의 실행 단계와 정렬되어야 합니다 netapp_lama_CloneVolumes 플레이북.

```

root@sap-jump:~# cat
/usr/sap/scripts/ansible/netapp_lama_ServiceConfigRemoval.yml
---
- hosts: ontapservers
  connection: local
  collections:
    - netapp.ontap
  gather_facts: false
  name: netapp_lama_ServiceConfigRemoval
  tasks:
    - name: Delete Clone
      na_ontap_volume:
        state: absent
        name: "{{ datavolumename }}{{ clonepostfix }}"
        use_rest: always
        vserver: "{{ svmname }}"
        wait_for_completion: True
        hostname: "{{ inventory_hostname }}"
        cert_filepath: "{{ certfile }}"
        key_filepath: "{{ keyfile }}"
        https: true
        validate_certs: false
    - name: Delete SnapShot
      na_ontap_snapshot:
        state: absent
        snapshot: "{{ datavolumename }}{{ snapshotpostfix }}"
        use_rest: always
        volume: "{{ datavolumename }}"
        vserver: "{{ svmname }}"
        hostname: "{{ inventory_hostname }}"
        cert_filepath: "{{ certfile }}"
        key_filepath: "{{ keyfile }}"
        https: true
        validate_certs: false
root@sap-jump:~#

```

Ansible Playbook NetApp_LaMa_ClearMountConfig.yml

Playbook은 에서 실행됩니다 netapp_lama_ClearMountConfig LaMa 시스템 업데이트 워크플로의 단계는 의 조합입니다 delete_clone.yml 및 delete_snapshot.yml (을 참조하십시오 ["NetApp Ansible 모듈 - YAML 파일"](#))를 클릭합니다. 의 실행 단계와 정렬되어야 합니다 netapp_lama_CloneVolumes 플레이북.


```

root@sap-jump:~# cat
/usr/sap/scripts/ansible/netapp_lama_ServiceConfigRemoval.yml
---
- hosts: ontapservers
  connection: local
  collections:
    - netapp.ontap
  gather_facts: false
  name: netapp_lama_ServiceConfigRemoval
  tasks:
    - name: Delete Clone
      na_ontap_volume:
        state: absent
        name: "{{ datavolumename }}{{ clonepostfix }}"
        use_rest: always
        vserver: "{{ svmname }}"
        wait_for_completion: True
        hostname: "{{ inventory_hostname }}"
        cert_filepath: "{{ certfile }}"
        key_filepath: "{{ keyfile }}"
        https: true
        validate_certs: false
    - name: Delete SnapShot
      na_ontap_snapshot:
        state: absent
        snapshot: "{{ datavolumename }}{{ snapshotpostfix }}"
        use_rest: always
        volume: "{{ datavolumename }}"
        vserver: "{{ svmname }}"
        hostname: "{{ inventory_hostname }}"
        cert_filepath: "{{ certfile }}"
        key_filepath: "{{ keyfile }}"
        https: true
        validate_certs: false
root@sap-jump:~#

```

Ansible 재고 샘플

이 재고 파일은 워크플로 실행 중에 동적으로 구축되며, 여기에는 설명을 위한 용도로만 표시됩니다.

```
ontapservers:
  hosts:
    grenada:
      ansible_host: "grenada"
      keyfile: "/usr/sap/scripts/ansible/certs/ontap.key"
      certfile: "/usr/sap/scripts/ansible/certs/ontap.pem"
      svmname: "svm-sap01"
      datavolumename: "HN9_sap"
      snapshotpostfix: " _snap_20221115"
      clonepostfix: " _clone_20221115"
```

결론

Ansible과 같은 최신 자동화 프레임워크를 SAP LaMa 프로비저닝 워크플로우에 통합하면 고객은 표준 또는 더 복잡한 인프라 요구사항을 해결할 수 있는 유연한 솔루션을 얻게 됩니다.

추가 정보를 찾을 수 있는 위치

이 문서에 설명된 정보에 대해 자세히 알아보려면 다음 문서 및/또는 웹 사이트를 검토하십시오.

- 컬렉션 in the NetApp 네임스페이스

["https://docs.ansible.com/ansible/latest/collections/netapp/index.html"](https://docs.ansible.com/ansible/latest/collections/netapp/index.html)

- Ansible 통합 및 샘플 Ansible Playbook에 대한 문서

["https://github.com/sap-linuxlab/demo.netapp_ontap"](https://github.com/sap-linuxlab/demo.netapp_ontap)

- 일반 Ansible 및 NetApp 통합

["https://www.ansible.com/integrations/infrastructure/netapp"](https://www.ansible.com/integrations/infrastructure/netapp)

- SAP LaMa를 Ansible과 통합하는 방법에 대한 블로그

["https://blogs.sap.com/2020/06/08/outgoing-api-calls-from-sap-landscape-management-lama-with-automation-studio/"](https://blogs.sap.com/2020/06/08/outgoing-api-calls-from-sap-landscape-management-lama-with-automation-studio/)

- SAP Landscape Management 3.0, Enterprise Edition 설명서

["https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/4df88a8f418c5059e10000000a42189c.html#loio4df88a8f418c5059e10000000a42189c"](https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/4df88a8f418c5059e10000000a42189c.html#loio4df88a8f418c5059e10000000a42189c)

- SAP LaMa 설명서 – 공급자 정의

["https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/bf6b3e43340a4cbcb0c0f3089715c068.html"](https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/bf6b3e43340a4cbcb0c0f3089715c068.html)

- SAP LaMa 설명서 - 맞춤형 후크

["https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/139eca2f925e48738a20dbf0b56674c5.html"](https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/139eca2f925e48738a20dbf0b56674c5.html)

- SAP LaMa 설명서 - SAP Host Agent 등록 스크립트 구성

["https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/250dfc5eef4047a38bab466c295d3a49.html"](https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/250dfc5eef4047a38bab466c295d3a49.html)

- SAP LaMa Documentation - Custom Operations 및 Custom Hook에 대한 매개 변수입니다

["https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/0148e495174943de8c1c3ee1b7c9cc65.html"](https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/0148e495174943de8c1c3ee1b7c9cc65.html)

- SAP LaMa 설명서 - Adaptive Design

["https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/737a99e86f8743bdb8d1f6cf4b862c79.html"](https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/737a99e86f8743bdb8d1f6cf4b862c79.html)

- NetApp 제품 설명서

["https://www.netapp.com/support-and-training/documentation/"](https://www.netapp.com/support-and-training/documentation/)

버전 기록

버전	날짜	문서 버전 기록
버전 1.0	2023년 1월	최초 릴리스

저작권 정보

Copyright © 2025 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.