

vRealize Automation 7.3으로 마이그레이션

2018년 1월 18일

vRealize Automation 7.3



vmware®

VMware 웹 사이트 (<https://docs.vmware.com/kr/>) 에서 최신 기술 문서를 확인할 수 있습니다.
이 문서에 대한 의견이 있으면 docfeedback@vmware.com으로 사용자 의견을 보내주십시오.

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2008 – 2018 VMware, Inc. [판권 소유](#). [저작권 및 상표 정보](#).

목차

업데이트된 정보 5

1 vRealize Automation 마이그레이션 8

2 마이그레이션 사전 요구 사항 10

최소 환경으로 마이그레이션하기 위한 사전 요구 사항 10

고가용성 환경으로 마이그레이션하기 위한 사전 요구 사항 11

3 사전 마이그레이션 작업 14

vRealize Automation 6.2.x에서 7.x로의 마이그레이션에 따른 변경 내용 검토 14

vSphere 에이전트에서 DoDeletes 설정을 False로 변경 15

마이그레이션을 위해 vRealize Automation 가상 시스템 준비 15

마이그레이션에 필요한 정보 수집 16

소스 vRealize Automation 환경에서 암호화 키 가져오기 17

소스 vRealize Automation 6.2.x 환경의 테넌트 및 IaaS 관리자 나열 18

소스 vRealize Automation 환경에서 대상 환경으로 각 테넌트 추가 19

추가된 각 테넌트에 대해 관리자 생성 20

최소 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화 21

고가용성 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화 23

소스 vRealize Automation 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집 실행 25

수동으로 소스 vRealize Automation IaaS Microsoft SQL 데이터베이스 복제 25

대상 vRealize Automation 환경의 스냅샷 생성 26

4 마이그레이션 절차 27

vRealize Automation 소스 데이터를 vRealize Automation 7.3 최소 환경으로 마이그레이션 27

vRealize Automation 소스 데이터를 vRealize Automation 7.3 고가용성 환경으로 마이그레이션 29

5 사후 마이그레이션 작업 32

소스 vRealize Automation 6.2.x 환경의 테넌트 및 IaaS 관리자 추가 32

연결 테스트 실행 및 마이그레이션된 끝점 확인 33

대상 vRealize Automation 7.3 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집 실행 34

고가용성 환경으로 마이그레이션한 후 로드 밸런서 재구성 35

외부 vRealize Orchestrator 서버를 vRealize Automation 7.3 으로 마이그레이션 35

외부 및 내장된 Orchestrator 간의 제어 센터 차이점 36

- Windows용 외부 vRealize Orchestrator 6.x를 vRealize Automation 7.3으로 마이그레이션 36
- 외부 vRealize Orchestrator 6.x 가상 장치를 vRealize Automation 7.3으로 마이그레이션 39
- 외부 vRealize Orchestrator 7.x를 vRealize Automation 7.3으로 마이그레이션 42
- 내장된 vRealize Orchestrator 서버 구성 44
- vRealize Automation 7.x에서 7.3으로 포함된 vRealize Orchestrator 서버 마이그레이션 46
- 일시적으로 소스 vRealize Automation 장치의 구성 변경 47
- 소스 vRealize Automation 장치의 포함된 vRealize Orchestrator 에서 구성 내보내기 48
- 포함된 소스 vRealize Orchestrator 의 구성과 데이터베이스를 포함된 대상 vRealize Orchestrator 로 가져오기 49
- 고가용성을 지원하도록 포함된 대상 vRealize Orchestrator 재구성 51
- 소스 vRealize Automation 장치의 구성 복원 51
- 대상 vRealize Orchestrator 에서 vRealize Automation 끝점 재구성 52
- 대상 vRealize Orchestrator 에서 vRealize Automation 인프라 끝점 재구성 53
- vRealize Orchestrator 사용자 지정 설치 54
- 대상 vRealize Automation 에서 포함된 vRealize Orchestrator 인프라 끝점 재구성 54
- 대상 vRealize Automation 환경에서 Azure 끝점 재구성 55
- vRealize Automation 6.2.x Automation Application Services를 7.3으로 마이그레이션 56
- 기존 가상 시스템의 소프트웨어 에이전트 업데이트 56
- 원본 대상 vRealize Automation IaaS Microsoft SQL 데이터베이스 삭제 58
- 마이그레이션 후 데이터 센터 위치 메뉴 콘텐츠 업데이트 58
- 대상 vRealize Automation 7.3 환경 검증 58

6 마이그레이션 문제 해결 60

- PostgreSQL 버전에 따른 오류 발생 60
- 마이그레이션 시 일부 가상 시스템의 배포가 생성되지 않음 60
- 로드 밸런서 구성 때문에 장기 실행 작업에서 시간 초과 발생 61
- 마이그레이션 로그 위치 61
- 마이그레이션 후 카탈로그 항목이 서비스 카탈로그에 나타나지만 요청할 수 없음 62
- 마이그레이션 후 vRealize Automation 에 빈 배포가 표시됨 63
- 마이그레이션 이후 배포라는 이름의 XaaS 리소스 매핑이 누락됨 63

업데이트된 정보

이 vRealize Automation 7.3으로 마이그레이션는 제품의 각 릴리스에 따라 또는 필요할 때 업데이트됩니다.

이 표에는 vRealize Automation 7.3으로 마이그레이션 설명서의 업데이트 기록이 나와 있습니다.

개정	설명
2018년 1월 18일	<ul style="list-style-type: none"> ■ 마이그레이션 이후 배포라는 이름의 XaaS 리소스 매핑이 누락됨 항목을 수정했습니다. ■ vSphere 에이전트에서 DoDeletes 설정을 False로 변경을 추가했습니다. ■ vRealize Automation 7.x에서 7.3으로 포함된 vRealize Orchestrator 서버 마이그레이션 항목을 수정했습니다. ■ 일시적으로 소스 vRealize Automation 장치의 구성 변경 항목을 수정했습니다. ■ 소스 vRealize Automation 장치의 포함된 vRealize Orchestrator에서 구성 내보내기 항목을 수정했습니다. ■ 포함된 소스 vRealize Orchestrator의 구성과 데이터베이스를 포함한 대상 vRealize Orchestrator로 가져오기 항목을 수정했습니다. ■ 고가용성을 지원하도록 포함된 대상 vRealize Orchestrator 재구성 항목을 수정했습니다. ■ 소스 vRealize Automation 장치의 구성 복원 항목을 수정했습니다.
2017년 12월 4일	<ul style="list-style-type: none"> ■ 최소 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화 항목을 수정했습니다. ■ 고가용성 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화 항목을 수정했습니다. ■ 마이그레이션 후 vRealize Automation에 빈 배포가 표시됨을 추가했습니다. ■ 1장vRealize Automation 마이그레이션 항목을 수정했습니다. ■ 최소 환경으로 마이그레이션하기 위한 사전 요구 사항 항목을 수정했습니다. ■ 고가용성 환경으로 마이그레이션하기 위한 사전 요구 사항 항목을 수정했습니다. ■ vRealize Automation 6.2.x에서 7.x로의 마이그레이션에 따른 변경 내용 검토를 추가했습니다. ■ 마이그레이션을 위해 vRealize Automation 가상 시스템 준비 항목을 수정했습니다. ■ 마이그레이션에 필요한 정보 수집 항목을 수정했습니다. ■ 최소 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화 항목을 수정했습니다. ■ 고가용성 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화 항목을 수정했습니다. ■ vRealize Automation 소스 데이터를 vRealize Automation 7.3 최소 환경으로 마이그레이션 항목을 수정했습니다. ■ vRealize Automation 소스 데이터를 vRealize Automation 7.3 고가용성 환경으로 마이그레이션 항목을 수정했습니다. ■ 소스 vRealize Automation 6.2.x 환경의 테넌트 및 IaaS 관리자 추가 항목을 수정했습니다. ■ 마이그레이션 후 vRealize Automation에 빈 배포가 표시됨을 추가했습니다. ■ 마이그레이션 이후 배포라는 이름의 XaaS 리소스 매핑이 누락됨을 추가했습니다.
2017년 9월 12일	고가용성 환경으로 마이그레이션하기 위한 사전 요구 사항 항목을 수정했습니다.
2017년 8월 30일	<ul style="list-style-type: none"> ■ 1장vRealize Automation 마이그레이션 항목을 수정했습니다. ■ 소스 vRealize Automation 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집 실행 항목을 수정했습니다. ■ 대상 vRealize Automation 7.3 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집 실행 항목을 수정했습니다. ■ Windows용 외부 vRealize Orchestrator 6.x를 vRealize Automation 7.3으로 마이그레이션 항목을 수정했습니다. ■ 마이그레이션 후 카탈로그 항목이 서비스 카탈로그에 나타나지만 요청할 수 없음 항목을 추가했습니다.
2017년 8월 7일	<ul style="list-style-type: none"> ■ 마이그레이션을 위해 vRealize Automation 가상 시스템 준비를 추가했습니다.

개정	설명
002425-01	<ul style="list-style-type: none"> ■ 사소한 편집 업데이트를 적용했습니다. ■ 제목을 변경했고 연결 테스트 실행 및 마이그레이션된 끝점 확인 항목에 정보를 추가했습니다. ■ vRealize Orchestrator 마이그레이션 항목을 추가했습니다. <ul style="list-style-type: none"> ■ 외부 vRealize Orchestrator 서버를 vRealize Automation 7.3 으로 마이그레이션 ■ Windows용 외부 vRealize Orchestrator 6.x를 vRealize Automation 7.3으로 마이그레이션 ■ 외부 vRealize Orchestrator 6.x 가상 장치를 vRealize Automation 7.3으로 마이그레이션 ■ 외부 vRealize Orchestrator 7.x를 vRealize Automation 7.3으로 마이그레이션 ■ 내장된 vRealize Orchestrator 서버 구성 ■ 외부 및 내장된 Orchestrator 간의 제어 센터 차이점
002425-00	최초 릴리스

vRealize Automation 마이그레이션

1

마이그레이션을 사용하여 현재 vRealize Automation 환경의 병존 업그레이드를 수행할 수 있습니다.

마이그레이션은 테넌트와 ID 저장소를 제외한 모든 데이터를 현재 vRealize Automation 소스 환경에서 vRealize Automation 최신 버전이 있는 대상 배포로 이동합니다.

마이그레이션은 데이터를 수집하여 대상 환경에 안전하게 복사하는 데 필요한 시간 동안 vRealize Automation 서비스를 중지하는 것 외에는 소스 환경을 변경하지 않습니다. 소스 vRealize Automation 데이터베이스의 크기에 따라 마이그레이션에 몇 분에서 몇 시간까지 소요될 수 있습니다.

최소 배포로 또는 고가용성 배포로 소스 환경을 마이그레이션할 수 있습니다.

마이그레이션 후 대상 환경을 운영 환경에서 사용하려면 소스 환경을 다시 활성화하지 마십시오. 마이그레이션 이후의 소스 환경 변경 내용은 대상 환경에 동기화되지 않습니다.

소스 환경이 vCloud Air 또는 vCloud Director와 통합되거나 물리적 끝점을 포함하는 경우에는 마이그레이션을 사용하여 업그레이드를 수행해야 합니다. 마이그레이션은 대상 환경에서 이러한 끝점 및 끝점과 연결된 모든 항목을 제거합니다. 마이그레이션은 대상 환경에서 6.x VMware vRealize Application Services 통합도 제거합니다.

참고 마이그레이션하기 전에 vRealize Automation 가상 시스템을 준비하기 위한 추가적인 작업을 완료해야 합니다. 마이그레이션하기 전에 [기술 자료 문서 51531](#)을 검토하십시오.

vRealize Automation 6.2.x에서 최신 버전으로 마이그레이션할 경우 다음과 같은 문제가 발생할 수 있습니다.

문제	해결 방법
<p>vRealize Automation 6.2.x에서 최신 버전으로 마이그레이션한 후 이러한 속성 정의를 사용하는 카탈로그 항목이 서비스 카탈로그에 나타나지만 요청할 수 없습니다.</p> <ul style="list-style-type: none"> ■ 제어 유형: 확인란 또는 링크. ■ 특성: 관계, 정규식 또는 속성 레이아웃. <p>vRealize Automation 7.x에서, 속성 정의는 더 이상 이러한 요소를 사용하지 않습니다.</p>	<p>포함된 제어 유형이나 특성이 아니라 vRealize Orchestrator 스크립트 작업을 사용하도록 속성 정의를 다시 만들거나 속성 정의를 구성해야 합니다. 자세한 내용은 마이그레이션 후 카탈로그 항목이 서비스 카탈로그에 나타나지만 요청할 수 없음 항목을 참조하십시오.</p>
<p>vRealize Automation 6.2.x에서 드롭다운 목록으로 상위-하위 관계를 정의하는 데 사용되던 정규식이 마이그레이션 후 더 이상 작동하지 않습니다. 예를 들어 특정 컨텍스트에서만 사용할 수 있는 리소스를 하나 이상 정의한 경우, 마이그레이션 이후 해당 리소스가 메뉴 항목으로 표시되지 않습니다.</p>	<p>마이그레이션 이후 속성 정의를 다시 생성하여 드롭다운 메뉴 항목을 복원해야 합니다.</p>

마이그레이션 사전 요구 사항

마이그레이션 사전 요구 사항은 대상 환경에 따라 다릅니다.

최소 환경 또는 고가용성 환경으로 마이그레이션할 수 있습니다.

이 장에서는 다음 주제에 대해 설명합니다.

- [최소 환경으로 마이그레이션하기 위한 사전 요구 사항](#)
- [고가용성 환경으로 마이그레이션하기 위한 사전 요구 사항](#)

최소 환경으로 마이그레이션하기 위한 사전 요구 사항

최소 환경으로 성공적으로 마이그레이션하기 위해 다음 사전 요구 사항을 검토합니다.

필수 조건

- 새로운 vRealize Automation 대상 환경이 있는지 확인합니다.
- 다음 요구 사항에 따라 대상 환경에 관련 프록시 에이전트를 설치합니다.
 - vSphere, Hyper-V, Citrix XenServer 및 테스트 프록시 에이전트에 대해 대상 프록시 에이전트 이름과 소스 프록시 에이전트 이름이 일치해야 합니다.

참고 다음 단계를 완료하여 에이전트 이름을 가져옵니다.

- 1 IaaS 노드에서 에이전트 설치 디렉토리로 이동합니다.
- 2 VRMAgent.exe.config 파일을 엽니다.
- 3 serviceConfiguration 태그 아래에서 agentName 특성의 값을 찾습니다.

- vSphere, Hyper-V, Citrix XenServer 및 테스트 프록시 에이전트에 대해 대상 프록시 에이전트 끝점 이름과 소스 프록시 에이전트 끝점 이름이 일치해야 합니다.
- 대상 환경에서 vSphere, Hyper-V, Citrix XenServer 또는 테스트 프록시 에이전트에 대해 끝점을 생성하지 마십시오.
- vRealize Automation 구성 요소의 버전 번호를 검토합니다.
 - a 대상 vRealize Automation 7.3 환경에서 브라우저를 시작합니다. vRealize Automation 장치 관리 콘솔(<https://vra-va-hostname.domain.name:5480>)로 이동합니다.
 - b `root`라는 사용자 이름과 장치를 배포할 때 입력한 암호를 사용하여 로그인합니다.

c **vRA 설정 > 클러스터**를 선택합니다.

d 삼각형을 클릭하여 호스트/노드 이름 레코드를 확장합니다.

vRealize Automation IaaS 구성 요소의 버전 번호가 일치하는지 확인합니다.

- **기술 자료 문서 000051531**을 검토하고 마이그레이션하기 전에 환경에서 모든 관련 수정을 수행해야 합니다.
- vRealize Automation 대상 IaaS 데이터베이스의 대상 Microsoft SQL Server 버전이 2012, 2014 또는 2016인지 확인합니다.
- 소스 및 대상 vRealize Automation 환경 사이에 포트 22가 열려 있는지 확인합니다. 소스 가상 장치와 대상 가상 장치 사이에 SSH(보안 셸) 연결을 설정하려면 포트 22가 필요합니다.
- 대상 환경의 각 IaaS 서버 노드에 Java SE Runtime Environment(JRE) 8, 업데이트 111(64비트) 이상이 설치되어 있는지 확인합니다. JRE를 설치한 후 JAVA_HOME 시스템 변수가 각 IaaS 노드에 설치한 Java 버전을 가리키는지 확인하고, 필요한 경우 경로를 수정합니다.
- 각 IaaS 노드에 PowerShell 3.0 이상이 설치되어 있는지 확인합니다.
- 소스 및 대상 vRealize Automation 환경이 실행되고 있는지 확인합니다.
- 소스 vRealize Automation 환경에서 실행 중인 사용자 작업 및 프로비저닝 작업이 없는지 확인합니다.
- 보안 소프트웨어는 마이그레이션 동안 대상 vRealize Automation 환경의 IaaS 노드에서 실행 중인 운영 체제 및 해당 구성 요소와 상호 작용하면 안 됩니다. 바이러스 백신 또는 보안 소프트웨어가 설치되어 있는 경우 해당 소프트웨어가 마이그레이션에 대해 올바르게 구성되었거나 사용되지 않도록 설정되었는지 확인합니다.

후속 작업

3장사전 **마이그레이션 작업**.

고가용성 환경으로 마이그레이션하기 위한 사전 요구 사항

고가용성 환경으로 성공적으로 마이그레이션하기 위해 다음 사전 요구 사항을 검토합니다.

필수 조건

- 고가용성을 위해 구성된 마스터 장치와 복제 가상 장치가 있는 새로운 vRealize Automation 대상 설치가 있는지 확인합니다. 참조 아키텍처에서 vRealize Automation 고가용성 구성 고려 사항 항목을 참조하십시오.
- 모든 vRealize Automation 가상 장치가 루트 사용자에게 대해 동일한 암호를 사용하는지 확인합니다.

- 다음 요구 사항에 따라 대상 환경에 관련 프록시 에이전트를 설치합니다.
 - vSphere, Hyper-V, Citrix XenServer 및 테스트 프록시 에이전트에 대해 대상 프록시 에이전트 이름과 소스 프록시 에이전트 이름이 일치해야 합니다.

참고 다음 단계를 완료하여 에이전트 이름을 가져옵니다.

- 1 IaaS 노드에서 에이전트 설치 디렉토리로 이동합니다.
- 2 VRMAgent.exe.config 파일을 엽니다.
- 3 serviceConfiguration 태그 아래에서 agentName 특성의 값을 찾습니다.

-
- vSphere, Hyper-V, Citrix XenServer 및 테스트 프록시 에이전트에 대해 대상 프록시 에이전트 끝점 이름과 소스 프록시 에이전트 끝점 이름이 일치해야 합니다.
 - 대상 환경에서 vSphere, Hyper-V, Citrix XenServer 또는 테스트 프록시 에이전트에 대해 끝점을 생성하지 마십시오.
- vRealize Automation 구성 요소의 버전 번호를 확인합니다.
 - a 대상 vRealize Automation 7.3 환경에서 브라우저를 시작하고 vRealize Automation 장치 관리 콘솔(<https://vra-va-hostname.domain.name:5480>)로 이동합니다.
 - b root라는 사용자 이름과 장치를 배포할 때 입력한 암호를 사용하여 로그인합니다.
 - c **vRA 설정 > 클러스터**를 선택합니다.
 - d 호스트/노드 이름 레코드를 확장하여 구성 요소를 보려면 확장 버튼을 클릭합니다.

모든 가상 장치 시스템 노드에서 vRealize Automation 구성 요소의 버전 번호가 일치하는지 확인합니다.

모든 IaaS 노드에서 vRealize Automation IaaS 구성 요소의 버전 번호가 일치하는지 확인합니다.
 - e [기술 자료 문서 000051531](#)을 검토하고 마이그레이션하기 전에 환경에서 모든 관련 수정을 수행해야 합니다.
 - 이러한 단계를 수행하여 트래픽을 마스터 노드에만 연결합니다.
 - a 모든 중복 노드를 사용하지 않도록 설정합니다.
 - b 로드 밸런서 설명서에 따라 이러한 항목에 대한 상태 모니터를 제거합니다.
 - vRealize Automation 가상 장치
 - IaaS 웹 사이트
 - IaaS Manager Service
 - vRealize Automation 장치 마스터 노드가 마스터 모드의 PostgreSQL 데이터베이스에 연결되는지 확인합니다.
 - a 대상 vRealize Automation 7.3 환경에서 브라우저를 시작하고 마스터 vRealize Automation 장치 관리 콘솔(<https://vra-va-hostname.domain.name:5480>)로 이동합니다.

- b root라는 사용자 이름과 장치를 배포할 때 입력한 암호를 사용하여 로그인합니다.
- c vRA 설정 > 데이터베이스를 선택합니다.
- d 데이터베이스 노드 호스트 모드가 마스터인지 확인합니다.
- vRealize Automation 대상 IaaS 데이터베이스의 대상 Microsoft SQL Server 버전이 2012, 2014 또는 2016인지 확인합니다.
- 소스 및 대상 vRealize Automation 환경 사이에 포트 22가 열려 있는지 확인합니다. 소스 가상 장치와 대상 가상 장치 사이에 SSH(보안 셸) 연결을 설정하려면 포트 22가 필요합니다.
- 대상 환경의 IaaS Web Service 및 Model Manager Server 노드가 Java Runtime Environment 권한을 가지고 있는지 확인합니다. Java SE Runtime Environment(JRE) 8, 업데이트 111(64비트) 이상이 설치되어 있어야 합니다. JAVA_HOME 시스템 변수가 각 IaaS 노드에 설치한 Java 버전을 가리키는지 확인합니다. 필요한 경우 경로를 수정합니다.
- 각 IaaS 노드에 PowerShell 3.0 이상이 설치되어 있는지 확인합니다.
- 소스 및 대상 vRealize Automation 환경이 실행되고 있는지 확인합니다.
- 소스 vRealize Automation 환경에서 실행 중인 사용자 작업 및 프로비저닝 작업이 없는지 확인합니다.
- 대상 vRealize Automation 환경의 IaaS 노드에서 실행되는 운영 체제 및 해당 구성 요소와 상호 작용할 수 있는 모든 바이러스 백신 또는 보안 소프트웨어가 올바르게 구성되었거나 사용하지 않도록 설정되었는지 확인합니다.
- 보안 소프트웨어는 마이그레이션 동안 대상 vRealize Automation 환경의 IaaS 노드에서 실행 중인 운영 체제 및 해당 구성 요소와 상호 작용하면 안 됩니다. 바이러스 백신 또는 보안 소프트웨어가 설치되어 있는 경우 마이그레이션에 대해 올바르게 구성되었거나 사용되지 않도록 설정되었는지 확인합니다.

후속 작업

3장사전 마이그레이션 작업.

사전 마이그레이션 작업

마이그레이션하기 전에 몇 가지 사전 마이그레이션 작업을 수행해야 합니다.

소스 vRealize Automation 환경을 대상 vRealize Automation 7.3 환경에 마이그레이션하기 전에 수행하는 마이그레이션 전 작업은 소스 환경에 따라 다릅니다.

이 장에서는 다음 주제에 대해 설명합니다.

- vRealize Automation 6.2.x에서 7.x로의 마이그레이션에 따른 변경 내용 검토
- vSphere 에이전트에서 DoDeletes 설정을 False로 변경
- 마이그레이션을 위해 vRealize Automation 가상 시스템 준비
- 마이그레이션에 필요한 정보 수집
- 소스 vRealize Automation 환경에서 암호화 키 가져오기
- 소스 vRealize Automation 6.2.x 환경의 테넌트 및 IaaS 관리자 나열
- 소스 vRealize Automation 환경에서 대상 환경으로 각 테넌트 추가
- 추가된 각 테넌트에 대해 관리자 생성
- 최소 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화
- 고가용성 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화
- 소스 vRealize Automation 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집 실행
- 수동으로 소스 vRealize Automation IaaS Microsoft SQL 데이터베이스 복제
- 대상 vRealize Automation 환경의 스냅샷 생성

vRealize Automation 6.2.x에서 7.x로의 마이그레이션에 따른 변경 내용 검토

vRealize Automation 7 이상에서는 업그레이드 프로세스 진행 중과 완료 후에 다양한 기능이 변경됩니다. vRealize Automation 6.2.x 배포를 최신 버전으로 업그레이드하기 전에 이러한 변경 내용을 검토하십시오.

vRealize Automation 6.2.x 및 7.x 간의 차이점에 대한 자세한 내용은 vRealize Automation 6.2.5에서 7.4로 업그레이드에서 “이 vRealize Automation 버전으로의 업그레이드에 대한 고려 사항”을 참조하십시오.

참고 vRealize Production Test Upgrade Assist Tool은 vRealize Automation 6.2.x 환경을 분석하여 업그레이드 문제를 일으킬 수 있는 기능 구성을 파악하고 환경이 업그레이드할 준비가 되었는지 확인합니다. 이 도구 및 관련 설명서를 다운로드하려면 [VMware vRealize Production Test Tool](#) 제품 다운로드 페이지로 이동하십시오.

vRealize Automation 6.2.x에서 최신 버전으로 마이그레이션한 후 이러한 속성 정의를 사용하는 카탈로그 항목이 서비스 카탈로그에 나타나지만 요청할 수 없습니다.

- 제어 유형: 확인란 또는 링크.
- 특성: 관계, 정규식 또는 속성 레이아웃.

vRealize Automation 7.x에서, 속성 정의는 더 이상 이러한 요소를 사용하지 않습니다. 포함된 제어 유형이나 특성이 아니라 vRealize Orchestrator 스크립트 작업을 사용하도록 속성 정의를 다시 만들거나 속성 정의를 구성해야 합니다. 자세한 내용은 [마이그레이션 후 카탈로그 항목이 서비스 카탈로그에 나타나지만 요청할 수 없음](#) 항목을 참조하십시오.

vSphere 에이전트에서 DoDeletes 설정을 False로 변경

vRealize Automation 6.2.4 환경에서 마이그레이션할 경우 마이그레이션하기 전에 대상 vSphere 에이전트의 DoDeletes 설정을 변경해야 합니다. 이렇게 하면 마이그레이션 후 소스 환경에서 가상 시스템이 삭제되는 것을 방지할 수 있습니다.

vRealize Automation 설치에 나와 있는 vSphere 에이전트 구성 절차의 단계에 따라 DoDeletes를 false로 설정합니다.

필수 조건

마이그레이션 사전 요구 사항을 완료했습니다.

후속 작업

마이그레이션을 위해 [vRealize Automation 가상 시스템 준비](#).

마이그레이션을 위해 vRealize Automation 가상 시스템 준비

vRealize Automation 6.2.x 가상 시스템 마이그레이션에 대한 알려진 문제 때문에 마이그레이션 후에 문제가 발생할 수 있습니다.

[기술 자료 문서 000051531](#)을 검토하고 마이그레이션하기 전에 환경에서 모든 관련 수정을 수행해야 합니다.

후속 작업

마이그레이션에 필요한 정보 수집.

마이그레이션에 필요한 정보 수집

다음 테이블을 사용하면 소스 및 대상 환경에서 마이그레이션하는 데 필요한 정보를 기록할 수 있습니다.

필수 조건

상황에 맞는 사전 요구 사항 확인을 완료합니다.

- [최소 환경으로 마이그레이션하기 위한 사전 요구 사항](#).
- [고가용성 환경으로 마이그레이션하기 위한 사전 요구 사항](#).

참고 [기술 자료 문서 000051531](#)을 검토하고 마이그레이션하기 전에 환경에서 모든 관련 수정을 수행해야 합니다.

표 3-1. 소스 vRealize Automation 장치

옵션	설명	값
호스트 이름	소스 vRealize Automation 장치 관리 콘솔에 로그인합니다. 시스템 탭에서 호스트 이름을 찾습니다. 호스트 이름은 FQDN(정규화된 도메인 이름)이어야 합니다.	
루트 사용자 이름	root	
루트 암호	소스 vRealize Automation 장치를 배포할 때 입력한 루트 암호입니다.	

표 3-2. 대상 vRealize Automation 장치

옵션	설명	값
루트 사용자 이름	root	
루트 암호	대상 vRealize Automation 장치를 배포할 때 입력한 루트 암호입니다.	
기본 테넌트	vRealize Automation 설치 마법사 에서 Single Sign-On을 구성할 때 생성한 기본 테넌트(일반적으로 vsphere.local)입니다.	
관리자 사용자 이름	대상 vRealize Automation 환경을 배포할 때 입력한 기본 테넌트 관리자(보통 관리자) 사용자 이름입니다.	
관리자 암호	대상 vRealize Automation 환경을 배포할 때 입력한 기본 테넌트 관리자 사용자 암호입니다.	

표 3-3. 대상 IaaS 데이터베이스

옵션	설명	값
데이터베이스 서버	복원된 vRealize Automation IaaS Microsoft SQL 데이터베이스가 상주하는 Microsoft SQL Server의 위치입니다. 명명된 인스턴스와 기본이 아닌 포트를 사용하는 경우, SERVER,PORT\INSTANCE-NAME 형식으로 입력합니다.	
복제된 데이터베이스 이름	소스에서 백업하고 대상 환경에서 복원한 소스 vRealize Automation 6.2.x 또는 7.x IaaS Microsoft SQL 데이터베이스의 이름입니다.	
로그인 이름	대상 환경의 복제된 IaaS Microsoft SQL 데이터베이스에 대해 db_owner 역할을 가진 사용자의 로그인 이름입니다. Windows 인증의 경우 vCloud Automation Center 관리 에이전트 서비스에 대한 Windows 계정이 복제된 IaaS SQL 데이터베이스의 db_owner 여야 합니다.	
암호	복제된 IaaS Microsoft SQL 데이터베이스에 대해 db_owner 역할을 가진 SQL Server 사용자의 암호입니다.	
원본 암호화 키	소스 환경에서 검색한 원본 암호화 키입니다. 소스 vRealize Automation 환경에서 암호화 키 가져오기 항목을 참조하십시오.	
새 암호	새 암호화 키를 생성하는 데 사용되는 일련의 단어입니다. 대상 vRealize Automation 환경에서 새 IaaS 구성 요소를 설치할 때마다 이 암호를 사용합니다.	

후속 작업

[소스 vRealize Automation 환경에서 암호화 키 가져오기](#).

소스 vRealize Automation 환경에서 암호화 키 가져오기

마이그레이션 절차의 일부로 소스 vRealize Automation 환경의 암호화 키를 입력해야 합니다.

필수 조건

소스 환경에서 액티브 Manager Service 호스트 가상 시스템에 대해 관리자 권한을 갖고 있는지 확인합니다.

프로시저

- 1 소스 환경에서 액티브 Manager Service를 호스팅하는 가상 시스템에서 관리자 명령 프롬프트를 열고 다음 명령을 실행합니다.

```
"C:\Program Files  
(x86)\VMware\VCACServer\ConfigTool\EncryptionKeyTool\DynamicOps.Tools.EncryptionKeyTool.exe" key-read -c "C:\Program Files (x86)\VMware\VCACServer\ManagerService.exe.config" -v
```

설치 디렉토리가 기본 위치(C:\Program Files (x86)\VMware\VCAC)가 아닌 경우 실제 설치 디렉토리로 경로를 편집합니다.

- 2 명령을 실행한 후에 표시되는 키를 저장합니다.

이 키는 다음 예와 유사한 긴 문자열입니다.

```
NRH+f/BlncB6yvasLS3sxespgdkcFWAEuyV0g4lfryg=.
```

후속 작업

- vRealize Automation 6.2.x 환경에서 마이그레이션하는 경우: [소스 vRealize Automation 환경에서 대상 환경으로 각 테넌트 추가](#)
- vRealize Automation 7.x 환경에서 마이그레이션하는 경우: [소스 vRealize Automation 6.2.x 환경의 테넌트 및 IaaS 관리자 나열](#)

소스 vRealize Automation 6.2.x 환경의 테넌트 및 IaaS 관리자 나열

vRealize Automation 6.2.x 환경을 마이그레이션하려면 먼저 각 테넌트에 대해 테넌트 관리자와 IaaS 관리자의 목록을 만들어야 합니다.

소스 vRealize Automation 콘솔에서 각 테넌트에 대해 다음 절차를 수행합니다.

참고 vRealize Automation 7.x 환경에서 마이그레이션한 경우에는 이 절차를 수행하지 않아도 됩니다.

필수 조건

소스 vRealize Automation 콘솔에 로그인합니다.

- 1 소스 가상 장치의 정규화된 도메인 이름(<https://vra-va-hostname.domain.name/vcac>)을 사용하여 vRealize Automation 콘솔을 엽니다.

고가용성 환경인 경우, 소스 가상 장치 로드 밸런서의 정규화된 도메인 이름(<https://vra-va-lb-hostname.domain.name/vcac>)을 사용하여 콘솔을 엽니다.

- 2 `administrator@vsphere.local`이라는 사용자 이름과 소스 vRealize Automation을 배포할 때 입력한 암호를 사용하여 로그인합니다.

프로시저

- 1 **관리 > 테넌트**를 선택합니다.

- 2 테넌트 이름을 클릭합니다.
- 3 관리자를 클릭합니다.
- 4 각 테넌트 및 IaaS 관리자 사용자 이름이 포함된 목록을 만듭니다.
- 5 취소를 클릭합니다.

후속 작업

소스 vRealize Automation 환경에서 대상 환경으로 각 테넌트 추가.

소스 vRealize Automation 환경에서 대상 환경으로 각 테넌트 추가

소스 환경에 있는 각 테넌트의 이름을 사용하여 대상 환경에 테넌트를 추가해야 합니다.

마이그레이션이 성공하려면 소스 환경의 각 테넌트를 대상 환경에 생성해야 합니다. 또한 소스 환경의 테넌트 URL 이름을 사용하여 추가하는 각 테넌트에 대해 테넌트별 액세스 URL을 사용해야 합니다. 마이그레이션하지 않을 미사용 테넌트가 소스 환경에 있는 경우에는 마이그레이션 전에 해당 테넌트를 소스 환경에서 삭제해야 합니다.

소스 환경의 각 테넌트에 대해 이 절차를 수행합니다.

- vRealize Automation 6.2.x 환경에서 마이그레이션할 때는 소스 환경에 있는 기존 SSO2 테넌트와 ID 저장소를 대상 환경의 VMware Identity Manager로 마이그레이션합니다.
- vRealize Automation 7.x 환경에서 마이그레이션할 때는 소스 환경에 있는 기존 VMware Identity Manager 테넌트와 ID 저장소를 대상 환경의 VMware Identity Manager로 마이그레이션합니다.

필수 조건

- [마이그레이션에 필요한 정보 수집](#).
- 대상 vRealize Automation 콘솔에 로그인합니다.
 - a 대상 가상 장치의 정규화된 도메인 이름(https://vra-va-hostname.domain.name/vcac)을 사용하여 vRealize Automation 콘솔을 엽니다.
고가용성 환경인 경우, 대상 가상 장치 로드 밸런서의 정규화된 도메인 이름(https://vra-va-lb-hostname.domain.name/vcac)을 사용하여 콘솔을 엽니다.
 - b administrator@vsphere.local이라는 사용자 이름과 대상 vRealize Automation을 배포할 때 입력한 암호를 사용하여 로그인합니다.

프로시저

- 1 **관리 > 테넌트**를 선택합니다.
- 2 **새로 만들기** 아이콘(+)을 클릭합니다.
- 3 **이름** 텍스트 상자에 소스 환경의 테넌트 이름과 일치하는 테넌트 이름을 입력합니다.
예를 들어 소스 환경의 테넌트 이름이 DEVTenant이면 DEVTenant를 입력합니다.

- 4 (선택 사항) **설명** 텍스트 상자에 설명을 입력합니다.
- 5 **URL 이름** 텍스트 상자에 소스 환경의 테넌트 URL 이름과 일치하는 테넌트 URL 이름을 입력합니다.
이 URL 이름은 vRealize Automation 콘솔 URL에 테넌트 관련 식별자를 추가하는 데 사용됩니다.
예를 들어 소스 환경의 DEVTenant에 대한 URL 이름이 dev인 경우 **dev**를 입력하여 `https://vra-va-hostname.domain.name/vcac/org/dev` URL을 생성합니다.
- 6 (선택 사항) **연락처 이메일** 텍스트 상자에 이메일 주소를 입력합니다.
- 7 **제출하고 다음 단계로 진행**을 클릭합니다.

후속 작업

추가된 각 테넌트에 대해 관리자 생성.

추가된 각 테넌트에 대해 관리자 생성

대상 환경에 추가한 각 테넌트에 대해 관리자를 생성해야 합니다. 관리자를 생성하려면 로컬 사용자 계정을 생성한 후 이 계정에 테넌트 관리자 권한을 할당합니다.

대상 환경의 각 테넌트에 대해 이 절차를 수행합니다.

필수 조건

- 소스 vRealize Automation 환경에서 대상 환경으로 각 테넌트 추가.
- 대상 vRealize Automation 콘솔에 로그인합니다.
 - a 대상 가상 장치의 정규화된 도메인 이름(`https://vra-va-hostname.domain.name/vcac`)을 사용하여 vRealize Automation 콘솔을 엽니다.
고가용성 환경인 경우, 대상 가상 장치 로드 밸런서의 정규화된 도메인 이름(`https://vra-va-lb-hostname.domain.name/vcac`)을 사용하여 콘솔을 엽니다.
 - b `administrator@vsphere.local`이라는 사용자 이름과 대상 vRealize Automation을 배포할 때 입력한 암호를 사용하여 로그인합니다.

프로시저

- 1 **관리 > 테넌트**를 선택합니다.
- 2 추가한 테넌트를 클릭합니다.
예를 들어 DEVTenant의 경우 **DEVTenant**를 클릭합니다.
- 3 **로컬 사용자**를 클릭합니다.
- 4 **새로 만들기** 아이콘(+)을 클릭합니다.
- 5 **사용자 세부 정보**에 요청된 정보를 입력하여 테넌트 관리자 역할을 할당할 로컬 사용자 계정을 생성합니다.
로컬 사용자 이름은 기본 로컬 디렉토리인 `vsphere.local`에서 고유해야 합니다.

- 6 **확인**을 클릭합니다.
- 7 **관리자**를 클릭합니다.
- 8 **테넌트 관리자** 검색 상자에 로컬 사용자 이름을 입력하고 Enter 키를 누릅니다.
- 9 사용자를 테넌트 관리자 목록에 추가하려면 검색 결과에서 적절한 이름을 클릭합니다.
- 10 **완료**를 클릭합니다.
- 11 콘솔에서 로그아웃합니다.

후속 작업

- 최소 배포의 경우: **최소 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화.**
- 고가용성 배포의 경우: **고가용성 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화.**

최소 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화

사용자와 그룹을 vRealize Automation의 최소 배포로 가져오기 전에 대상 vRealize Automation을 Active Directory 링크에 연결해야 합니다.

각 테넌트에 대해 이 절차를 수행합니다. 테넌트에 Active Directory가 두 개 이상 있는 경우에는 테넌트에서 사용하는 Active Directory 각각에 대해 이 절차를 수행합니다.

필수 조건

- **추가된 각 테넌트에 대해 관리자 생성.**
- Active Directory에 대한 액세스 권한이 있는지 확인합니다.
- **테넌트 관리자**로 vRealize Automation 콘솔에 로그인합니다.

프로시저

- 1 **관리 > 디렉토리 관리 > 디렉토리**를 선택합니다.
- 2 **디렉토리 추가** 아이콘(+)을 클릭하고 **LDAP/IWA를 통한 Active Directory 추가**를 선택합니다.
- 3 Active Directory 계정 설정을 입력합니다.

◆ 비네이티브 Active Directory

옵션	샘플 입력
디렉토리 이름	고유한 디렉토리 이름을 입력합니다. 비네이티브 Active Directory를 사용하는 경우 LDAP를 통한 Active Directory 를 선택합니다.

이 디렉토리는 DNS 서비스 위치를 지원 합니다. 이 옵션을 선택 해제합니다.

옵션	샘플 입력
기본 DN	서버가 검색하는 디렉토리에 대한 시작점의 DN(고유 이름)을 입력합니다. 예를 들어 cn=users,dc=rainpole,dc=local 을 입력합니다.
Bind DN	사용자를 검색할 권한이 있는 Active Directory 사용자 계정의 CN(일반 이름)을 포함하여 전체 DN(고유 이름)을 입력합니다. 예를 들어 cn=config_admin infra,cn=users,dc=rainpole,dc=local 을 입력합니다.
Bind DN 암호	사용자를 검색할 수 있는 계정의 Active Directory 암호를 입력하고 연결 테스트 를 클릭하여, 구성된 디렉토리에 대한 연결을 테스트합니다.

◆ 네이티브 Active Directory

옵션	샘플 입력
디렉토리 이름	고유한 디렉토리 이름을 입력합니다. 네이티브 Active Directory를 사용하는 경우 Active Directory(Windows 통합 인증) 를 선택합니다.
도메인 이름	가입할 도메인의 이름을 입력합니다.
도메인 관리자 사용자 이름	도메인 관리자의 사용자 이름을 입력합니다.
도메인 관리자 암호	도메인 관리자의 암호를 입력합니다.
Bind 사용자 UPN	도메인에서 인증될 수 있는 사용자의 이름을 이메일 주소 형식을 사용하여 입력합니다.
Bind DN 암호	사용자를 검색할 수 있는 계정에 대한 Active Directory Bind 계정 암호를 입력합니다.

4 저장 및 다음을 클릭합니다.

도메인 선택에 도메인 목록이 표시됩니다.

5 기본 도메인 설정을 수락하고 다음을 클릭합니다.

6 특성 이름이 올바른 Active Directory 특성에 매핑되어 있는지 확인하고 다음을 클릭합니다.

7 동기화할 그룹과 사용자를 선택합니다.

a **새로 만들기** 아이콘(+)을 클릭합니다.

b 사용자 도메인을 입력하고 **그룹 찾기**를 클릭합니다.

예를 들어 **dc=vcac,dc=local**을 입력합니다.

c 동기화할 그룹을 선택하려면 **선택**을 클릭하고 **다음**을 클릭합니다.

d **사용자 선택**에서 동기화할 사용자를 선택하고 **다음**을 클릭합니다.

vRealize Automation을 사용해야 하는 사용자 및 그룹만 추가합니다. 중첩 구조의 모든 그룹에서 vRealize Automation을 사용해야 하는 경우가 아니면 **중첩된 그룹 동기화**를 선택하지 마십시오.

8 디렉토리에 동기화 중인 사용자 및 그룹을 검토하고 **디렉토리 동기화**를 클릭합니다.

디렉토리 동기화에 시간이 걸리며 백그라운드에서 실행됩니다.

후속 작업

소스 vRealize Automation 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집 실행

고가용성 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화

사용자와 그룹을 고가용성 vRealize Automation 환경으로 가져오기 전에 Active Directory 링크에 연결해야 합니다.

- 각 테넌트에 대해 1~8단계를 수행합니다. 테넌트에 Active Directory가 두 개 이상 있는 경우에는 테넌트에서 사용하는 Active Directory 각각에 대해 이 절차를 수행합니다.
- 테넌트에 연결된 각 ID 제공자에 대해 9~10단계를 반복합니다.

필수 조건

- 추가된 각 테넌트에 대해 관리자 생성.
- Active Directory에 대한 액세스 권한이 있는지 확인합니다.
- 테넌트 관리자 사용자 이름과 암호를 사용하여 테넌트 대상 vRealize Automation 콘솔 (<https://vra-va-lb-hostname.domain.name/vcac/org/tenant-URL-name>)에 로그인합니다.

프로시저

- 1 관리 > 디렉토리 관리 > 디렉토리를 선택합니다.
- 2 디렉토리 추가 아이콘(+)을 클릭하고 LDAP/IWA를 통한 Active Directory 추가를 선택합니다.
- 3 Active Directory 계정 설정을 입력합니다.
 - ◆ 비네이티브 Active Directory

옵션	샘플 입력
디렉토리 이름	고유한 디렉토리 이름을 입력합니다. 비네이티브 Active Directory를 사용하는 경우 LDAP를 통한 Active Directory를 선택합니다.
이 디렉토리는 DNS 서비스 위치를 지원 합니다.	이 옵션을 선택 해제합니다.
기본 DN	서버가 검색하는 디렉토리에 대한 시작점의 DN(고유 이름)을 입력합니다. 예를 들어 <code>cn=users,dc=rainpole,dc=local</code> 을 입력합니다.

옵션	샘플 입력
Bind DN	사용자를 검색할 권한이 있는 Active Directory 사용자 계정의 CN(일반 이름)을 포함하여 전체 DN(고유 이름)을 입력합니다. 예를 들어 cn=config_admin infra,cn=users,dc=rainpole,dc=local 을 입력합니다.
Bind DN 암호	사용자를 검색할 수 있는 계정의 Active Directory 암호를 입력하고 연결 테스트 를 클릭하여, 구성된 디렉토리에 대한 연결을 테스트합니다.

◆ 네이티브 Active Directory

옵션	샘플 입력
디렉토리 이름	고유한 디렉토리 이름을 입력합니다. 네이티브 Active Directory를 사용하는 경우 Active Directory(Windows 통합 인증) 를 선택합니다.
도메인 이름	가입할 도메인의 이름을 입력합니다.
도메인 관리자 사용자 이름	도메인 관리자의 사용자 이름을 입력합니다.
도메인 관리자 암호	도메인 관리자 계정의 암호를 입력합니다.
Bind 사용자 UPN	도메인에서 인증될 수 있는 사용자의 이름을 이메일 주소 형식을 사용하여 입력합니다.
Bind DN 암호	사용자를 검색할 수 있는 계정에 대한 Active Directory Bind 계정 암호를 입력합니다.


4 저장 및 다음을 클릭합니다.

도메인 선택 페이지에 도메인 목록이 표시됩니다.

5 기본 도메인 설정을 수락하고 다음을 클릭합니다.

6 특성 이름이 올바른 Active Directory 특성에 매핑되어 있는지 확인하고 다음을 클릭합니다.

7 동기화할 그룹과 사용자를 선택합니다.

a **새로 만들기** 아이콘  을 클릭합니다.

b 사용자 도메인을 입력하고 **그룹 찾기**를 클릭합니다.

예를 들어 **dc=vcac,dc=local**을 입력합니다.

c 동기화할 그룹을 선택하려면 **선택**을 클릭하고 **다음**을 클릭합니다.

d **사용자 선택** 페이지에서 동기화할 사용자를 선택하고 **다음**을 클릭합니다.

vRealize Automation을 사용해야 하는 사용자 및 그룹만 추가합니다. 중첩 구조의 모든 그룹에서 vRealize Automation을 사용해야 하는 경우가 아니면 **중첩된 그룹 동기화**를 선택하지 마십시오.

8 디렉토리에 동기화 중인 사용자 및 그룹을 검토하고 **디렉토리 동기화**를 클릭합니다.

디렉토리 동기화에 시간이 걸리며 백그라운드에서 실행됩니다.

9 **관리 > 디렉토리 관리 > ID 제공자**를 선택하고 새 ID 제공자를 클릭합니다.

예: **WorkspaceIDP_1**.

10 선택한 ID 제공자에 대한 페이지에서 각 노드에 커넥터를 추가합니다.

- a 커넥터 추가에 대한 지침을 따릅니다.
- b vRealize Automation 로드 밸런서의 FQDN(정규화된 도메인 이름)을 가리키도록 **IdP 호스트 이름** 속성의 값을 업데이트합니다.
- c **저장**을 클릭합니다.

후속 작업

[소스 vRealize Automation 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집 실행.](#)

소스 vRealize Automation 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집 실행

마이그레이션하기 전에 소스 vRealize Automation 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집을 실행해야 합니다.

이 데이터 수집은 7.1 및 7.2 배포에 대해 vRealize Automation 7.3에서 [로드 밸런서 재구성] 작업을 수행하는 데 필요합니다.

참고 vRealize Automation 6.2.x에서 마이그레이션할 때 소스 환경에서 이 데이터 수집을 실행할 필요가 없습니다. vRealize Automation 6.2.x는 [로드 밸런서 재구성] 작업을 지원하지 않습니다.

프로시저

- ◆ vRealize Automation 7.3으로 마이그레이션하기 전에 소스 vRealize Automation 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집을 실행합니다. 자세한 내용은 vRealize Automation 관리에서 끝점 데이터 수집 수동 시작 항목을 참조하십시오.

후속 작업

[수동으로 소스 vRealize Automation IaaS Microsoft SQL 데이터베이스 복제.](#)

수동으로 소스 vRealize Automation IaaS Microsoft SQL 데이터베이스 복제

마이그레이션을 시작하기 전에 vRealize Automation 소스 환경의 IaaS Microsoft SQL 데이터베이스를 백업한 후 vRealize Automation 대상 환경에 새로 생성된 빈 데이터베이스에 복원해야 합니다.

필수 조건

- [소스 vRealize Automation 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집 실행.](#)
- SQL Server 데이터베이스 백업 및 복원에 대한 정보를 가져옵니다. 전체 SQL Server 데이터베이스 백업을 생성하고 SQL Server 데이터베이스를 새 위치에 복원하는 데 대한 문서를 [Microsoft Developer Network](#)에서 검색합니다.

프로시저

- ◆ 소스 vRealize Automation 6.2.x 또는 7.x IaaS Microsoft SQL 데이터베이스의 전체 백업을 생성합니다. 이 백업을 사용하여 대상 환경의 새로 생성된 빈 데이터베이스에 SQL 데이터베이스를 복원합니다.

후속 작업

[대상 vRealize Automation 환경의 스냅샷 생성.](#)

대상 vRealize Automation 환경의 스냅샷 생성

각 대상 vRealize Automation 가상 시스템의 스냅샷을 생성합니다. 마이그레이션이 실패하면 가상 시스템 스냅샷을 사용하여 마이그레이션을 다시 시도할 수 있습니다.

자세한 내용은 vSphere 설명서를 참조하십시오.

필수 조건

[수동으로 소스 vRealize Automation IaaS Microsoft SQL 데이터베이스 복제.](#)

후속 작업

다음 절차 중 하나를 수행합니다.

- [vRealize Automation 소스 데이터를 vRealize Automation 7.3 최소 환경으로 마이그레이션.](#)
- [vRealize Automation 소스 데이터를 vRealize Automation 7.3 고가용성 환경으로 마이그레이션.](#)

마이그레이션 절차

소스 vRealize Automation 환경 데이터를 마이그레이션하기 위해 수행하는 절차는 최소 환경으로 마이그레이션하는지 아니면 고가용성 환경으로 마이그레이션하는지에 따라 다릅니다.

이 장에서는 다음 주제에 대해 설명합니다.

- vRealize Automation 소스 데이터를 vRealize Automation 7.3 최소 환경으로 마이그레이션
- vRealize Automation 소스 데이터를 vRealize Automation 7.3 고가용성 환경으로 마이그레이션

vRealize Automation 소스 데이터를 vRealize Automation 7.3 최소 환경으로 마이그레이션

현재의 vRealize Automation 환경을 새로운 vRealize Automation 7.3 설치로 마이그레이션할 수 있습니다.

필수 조건

- 마이그레이션에 필요한 정보 수집.
- 소스 vRealize Automation 환경에서 암호화 키 가져오기.
- 소스 vRealize Automation 환경에서 대상 환경으로 각 테넌트 추가.
- 추가된 각 테넌트에 대해 관리자 생성.
- 최소 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화.
- 수동으로 소스 vRealize Automation IaaS Microsoft SQL 데이터베이스 복제.
- 대상 vRealize Automation 환경의 스냅샷 생성.

프로시저

- 1 대상 vRealize Automation 7.3 환경에서 브라우저를 시작하고 vRealize Automation 장치 관리 콘솔(<https://vra-va-hostname.domain.name:5480>)로 이동합니다.
- 2 root라는 사용자 이름과 장치를 배포할 때 입력한 암호를 사용하여 로그인합니다.
- 3 vRA 설정 > 마이그레이션을 선택합니다.

4 소스 vRealize Automation 장치에 대해 정보를 입력합니다.

옵션	설명
호스트 이름	소스 vRealize Automation 장치의 호스트 이름입니다.
루트 사용자 이름	root
루트 암호	vRealize Automation 장치를 배포할 때 입력한 루트 암호입니다.

5 대상 vRealize Automation 장치에 대해 정보를 입력합니다.

옵션	설명
루트 사용자 이름	root
루트 암호	대상 vRealize Automation 장치를 배포할 때 입력한 루트 암호입니다.
기본 테넌트	설치 마법사에서 Single Sign-On을 구성할 때 생성한 기본 테넌트(일반적으로 vsphere.local)입니다.
관리자 사용자 이름	대상 vRealize Automation 장치를 배포할 때 입력한 테넌트 관리자 사용자 이름입니다. 필요한 경우 기존 값을 변경합니다.
관리자 암호	대상 vRealize Automation 장치를 배포할 때 기본 테넌트 관리자에 대해 입력한 암호입니다.

6 대상 IaaS 데이터베이스 서버에 대한 정보를 입력합니다.

옵션	설명
데이터베이스 서버	복제된 vRealize Automation IaaS Microsoft SQL 데이터베이스가 상주하는 Microsoft SQL Server의 위치입니다. 명명된 인스턴스와 기본이 아닌 포트가 사용되는 경우 SERVER,PORT\INSTANCE-NAME 형식으로 입력합니다. AAG(AlwaysOn 가용성 그룹)를 사용하도록 대상 Microsoft SQL Server를 구성하는 경우, 포트 또는 인스턴스 이름 없이 대상 SQL Server를 AAG 수신기 이름으로 입력해야 합니다.
복제된 데이터베이스 이름	소스에서 백업하고 대상 환경에서 복원한 소스 vRealize Automation 6.2.x 또는 7.x IaaS Microsoft SQL 데이터베이스의 이름입니다.
인증 모드	<ul style="list-style-type: none"> ■ Windows Windows 인증 모드를 사용하는 경우 IaaS 서비스 사용자에게는 SQL Server db_owner 역할이 있어야 합니다. SQL Server 인증 모드를 사용할 때 동일한 사용 권한이 적용됩니다. ■ SQL Server SQL Server에 로그인 이름 및 암호 텍스트 상자가 열립니다.
로그인 이름	복제된 IaaS Microsoft SQL 데이터베이스에 대해 db_owner 역할을 가진 SQL Server 사용자의 로그인 이름입니다.
암호	복제된 IaaS Microsoft SQL 데이터베이스에 대해 db_owner 역할을 가진 SQL Server 사용자의 암호입니다.
원본 암호화 키	소스 환경에서 검색한 원본 암호화 키입니다. 소스 vRealize Automation 환경에서 암호화 키 가져오기 항목을 참조하십시오.
새 암호	새 암호화 키를 생성하는 데 사용되는 일련의 단어입니다. 대상 vRealize Automation 환경에서 새 IaaS 구성 요소를 설치할 때마다 이 암호를 사용합니다.

7 검증을 클릭합니다.

이 페이지는 검증 진행률을 표시합니다.

- 모든 항목이 성공적으로 검증되는 경우 8단계로 이동합니다.
- 항목 검증에 실패할 경우 IaaS 노드에서 오류 메시지와 검증 로그 파일을 확인합니다. 로그 파일 위치는 [마이그레이션 로그 위치](#) 항목을 참조하십시오. **설정 편집**을 클릭하고 문제 항목을 편집합니다. 7단계로 이동합니다.

8 마이그레이션을 클릭합니다.

이 페이지는 마이그레이션 진행률을 표시합니다.

- 마이그레이션이 성공적으로 완료되면 소프트웨어 에이전트 마이그레이션 후 업데이트에 대한 정보가 페이지에 표시됩니다.
- 마이그레이션이 실패한 경우 가상 장치 및 IaaS 노드에서 마이그레이션 로그 파일을 검사합니다. 로그 파일 위치는 [마이그레이션 로그 위치](#) 항목을 참조하십시오.

마이그레이션을 다시 시작하기 전에 다음 단계를 완료합니다.

- a 마이그레이션 전에 스냅샷을 생성할 때 캡처한 상태로 대상 vRealize Automation 환경을 되돌립니다.
- b 소스 IaaS 데이터베이스의 백업을 사용하여 대상 IaaS Microsoft SQL 데이터베이스를 복원합니다.

후속 작업

[5장사후 마이그레이션 작업](#).

vRealize Automation 소스 데이터를 vRealize Automation 7.3 고가용성 환경으로 마이그레이션

현재 vRealize Automation 환경을 고가용성 환경으로 구성된 vRealize Automation 7.3의 새 설치로 마이그레이션할 수 있습니다.

필수 조건

- 마이그레이션에 필요한 정보 수집.
- 소스 vRealize Automation 환경에서 암호화 키 가져오기.
- 소스 vRealize Automation 환경에서 대상 환경으로 각 테넌트 추가.
- 추가된 각 테넌트에 대해 관리자 생성.
- 고가용성 환경으로 마이그레이션하기 전에 Active Directory 링크의 사용자와 그룹 동기화.
- 수동으로 소스 vRealize Automation IaaS Microsoft SQL 데이터베이스 복제.
- 대상 vRealize Automation 환경의 스냅샷 생성.

프로시저

- 1 대상 vRealize Automation 7.3 환경에서 브라우저를 열고 마스터 vRealize Automation 장치 관리 콘솔(<https://vra-va-hostname.domain.name:5480>)로 이동합니다.

- 2 root라는 사용자 이름과 장치를 배포할 때 입력한 암호를 사용하여 로그인합니다.
- 3 vRA 설정 > 마이그레이션을 선택합니다.
- 4 소스 vRealize Automation 장치에 대한 정보를 입력합니다.

옵션	설명
호스트 이름	소스 vRealize Automation 장치의 호스트 이름입니다.
루트 사용자 이름	root
루트 암호	소스 vRealize Automation 장치를 배포할 때 입력한 루트 암호입니다.

- 5 대상 vRealize Automation 장치에 대해 정보를 입력합니다.

옵션	설명
루트 사용자 이름	root
루트 암호	대상 vRealize Automation 장치를 배포할 때 입력한 루트 암호입니다.
기본 테넌트	설치 마법사에서 Single Sign-On을 구성할 때 생성한 기본 테넌트(일반적으로 vsphere.local)입니다.
관리자 사용자 이름	대상 vRealize Automation 장치를 배포할 때 입력한 테넌트 관리자 사용자 이름입니다. 필요한 경우 기존 값을 변경합니다.
관리자 암호	대상 vRealize Automation 장치를 배포할 때 기본 테넌트 관리자에 대해 입력한 암호입니다.

- 6 대상 IaaS 데이터베이스 서버에 대한 정보를 입력합니다.

옵션	설명
데이터베이스 서버	복원된 vRealize Automation IaaS Microsoft SQL 데이터베이스가 상주하는 Microsoft SQL Server 인스턴스의 위치입니다. 명명된 인스턴스와 기본이 아닌 포트가 사용되는 경우 SERVER,PORT\INSTANCE-NAME 형식으로 입력합니다. AAG(AlwaysOn 가용성 그룹)를 사용하도록 대상 Microsoft SQL Server를 구성하는 경우, 포트 또는 인스턴스 이름 없이 대상 SQL Server를 AAG 수신기 이름으로 입력해야 합니다.
복제된 데이터베이스 이름	소스에서 백업하고 대상 환경에서 복원한 소스 vRealize Automation 6.2.x 또는 7.x IaaS Microsoft SQL 데이터베이스의 이름입니다.
인증 모드	<ul style="list-style-type: none"> ■ Windows Windows 인증 모드를 사용하는 경우 IaaS 서비스 사용자에게는 SQL Server db_owner 역할이 있어야 합니다. SQL Server 인증 모드를 사용할 때 동일한 사용 권한이 적용됩니다. ■ SQL Server SQL Server에 로그인 이름 및 암호 텍스트 상자가 열립니다.
로그인 이름	복제된 IaaS Microsoft SQL 데이터베이스에 대해 db_owner 역할을 가진 SQL Server 사용자의 로그인 이름입니다.
암호	복제된 IaaS Microsoft SQL 데이터베이스에 대해 db_owner 역할을 가진 SQL Server 사용자의 암호입니다.

옵션	설명
원본 암호화 키	소스 환경에서 검색한 원본 암호화 키입니다. 소스 vRealize Automation 환경에서 암호화 키 가져오기 항목을 참조하십시오.
새 암호	새 암호화 키를 생성하는 데 사용되는 일련의 단어입니다. 대상 vRealize Automation 환경에서 새 IaaS 구성 요소를 설치할 때마다 이 암호를 사용합니다.

7 검증을 클릭합니다.

이 페이지는 검증 진행률을 표시합니다.

- 모든 항목이 성공적으로 검증되는 경우 8단계로 이동합니다.
- 항목 검증에 실패할 경우 IaaS 노드에서 오류 메시지와 검증 로그 파일을 확인합니다. 로그 파일 위치는 [마이그레이션 로그 위치](#) 항목을 참조하십시오. **설정 편집**을 클릭하고 문제 항목을 편집합니다. 7단계로 이동합니다.

8 마이그레이션을 클릭합니다.

이 페이지는 마이그레이션 진행률을 표시합니다.

- 마이그레이션이 성공적으로 완료되면 소프트웨어 에이전트 마이그레이션 후 업데이트에 대한 정보가 페이지에 표시됩니다.
- 마이그레이션이 실패한 경우 가상 장치 및 IaaS 노드에서 마이그레이션 로그 파일을 검사합니다. 로그 파일 위치는 [마이그레이션 로그 위치](#) 항목을 참조하십시오.

마이그레이션을 다시 시작하기 전에 다음 단계를 완료합니다.

- 마이그레이션 전에 스냅샷을 생성할 때 캡처한 상태로 대상 vRealize Automation 환경을 되돌립니다.
- 소스 IaaS 데이터베이스의 백업을 사용하여 대상 IaaS Microsoft SQL 데이터베이스를 복원합니다.

후속 작업

[5장사후 마이그레이션 작업.](#)

사후 마이그레이션 작업

vRealize Automation을 마이그레이션한 후에 상황과 관련된 마이그레이션 후 작업을 수행합니다.

참고 ID 저장소를 마이그레이션한 후 vRealize Code Stream의 사용자는 수동으로 vRealize Code Stream 역할을 재할당해야 합니다.

이 장에서는 다음 주제에 대해 설명합니다.

- 소스 vRealize Automation 6.2.x 환경의 테넌트 및 IaaS 관리자 추가
- 연결 테스트 실행 및 마이그레이션된 끝점 확인
- 대상 vRealize Automation 7.3 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집 실행
- 고가용성 환경으로 마이그레이션한 후 로드 밸런서 재구성
- 외부 vRealize Orchestrator 서버를 vRealize Automation 7.3 으로 마이그레이션
- vRealize Automation 7.x에서 7.3으로 포함된 vRealize Orchestrator 서버 마이그레이션
- 대상 vRealize Orchestrator에서 vRealize Automation 끝점 재구성
- 대상 vRealize Orchestrator에서 vRealize Automation 인프라 끝점 재구성
- vRealize Orchestrator 사용자 지정 설치
- 대상 vRealize Automation에서 포함된 vRealize Orchestrator 인프라 끝점 재구성
- 대상 vRealize Automation 환경에서 Azure 끝점 재구성
- vRealize Automation 6.2.x Automation Application Services를 7.3으로 마이그레이션
- 기존 가상 시스템의 소프트웨어 에이전트 업데이트
- 원본 대상 vRealize Automation IaaS Microsoft SQL 데이터베이스 삭제
- 마이그레이션 후 데이터 센터 위치 메뉴 콘텐츠 업데이트
- 대상 vRealize Automation 7.3 환경 검증

소스 vRealize Automation 6.2.x 환경의 테넌트 및 IaaS 관리자 추가

마이그레이션 후 각 테넌트의 vRealize Automation 6.2.x 테넌트 관리자를 삭제하고 복원해야 합니다.

대상 vRealize Automation 콘솔에서 각 테넌트에 대해 다음 절차를 수행합니다.

참고 vRealize Automation 7.x 환경에서 마이그레이션한 경우에는 이 절차를 수행하지 않아도 됩니다.

필수 조건

- vRealize Automation 7.3으로 마이그레이션을 완료했습니다.
- 대상 vRealize Automation 콘솔에 로그인합니다.

프로시저

- 1 **관리 > 테넌트**를 선택합니다.
- 2 테넌트 이름을 클릭합니다.
- 3 **관리자**를 클릭합니다.
- 4 각 테넌트 관리자 이름 및 사용자 이름이 포함된 목록을 만듭니다.
- 5 각 관리자를 가리킨 후 삭제 아이콘([삭제])을 클릭합니다. 모든 관리자를 삭제할 때까지 이 작업을 반복합니다.
- 6 **완료**를 클릭합니다.
- 7 [테넌트] 페이지에서 테넌트 이름을 다시 클릭합니다.
- 8 **관리자**를 클릭합니다.
- 9 앞서 삭제한 각 사용자의 이름을 적절한 검색 상자에 입력하고 Enter 키를 누릅니다.
- 10 검색 결과에서 적절한 사용자의 이름을 클릭하여 해당 사용자를 관리자로 다시 추가합니다.
작업을 마치면 테넌트 관리자 목록이 삭제했던 관리자 목록과 동일하게 됩니다.
- 11 **완료**를 클릭합니다.

연결 테스트 실행 및 마이그레이션된 끝점 확인

vRealize Automation 7.3으로 마이그레이션하면 대상 환경에서 끝점이 변경됩니다.

vRealize Automation 7.3으로 마이그레이션한 후에는 적용 가능한 모든 끝점에 대해 **연결 테스트** 작업을 사용해야 합니다. 마이그레이션된 일부 끝점을 수정해야 할 수도 있습니다. 자세한 내용은 vRealize Automation 구성에서 업그레이드 또는 마이그레이션된 끝점 사용 시 고려 사항을 참조하십시오.

업그레이드 또는 마이그레이션된 끝점의 기본 보안 설정은 신뢰할 수 없는 인증서를 허용하지 않는 것입니다.

vRealize Automation 7.3 이전 릴리스에서 업그레이드 또는 마이그레이션한 후 신뢰할 수 없는 인증서를 사용하고 있는 경우 인증서 검증을 사용하도록 모든 vSphere 및 NSX 끝점에 대해 다음 단계를 수행해야 합니다. 그렇지 않으면 인증서 오류가 발생하고 끝점 작업이 실패합니다. 자세한 내용은 <http://kb.vmware.com/kb/2150230>의 VMware 기술 자료 문서 vRA 7.3으로 업그레이드 후 끝점 통신이 끊김(2150230) 및 <http://kb.vmware.com/kb/2108294>의 웹 브라우저 인증서 주의를 방지하도록 vCenter Server 루트 인증서를 다운로드 및 설치하는 방법(2108294)을 참조하십시오.

- 1 업그레이드 또는 마이그레이션 후에 vRealize Automation vSphere 에이전트 시스템에 로그인하고 **서비스** 탭을 사용하여 vSphere 에이전트를 다시 시작합니다.

마이그레이션이 모든 에이전트를 다시 시작하지 못할 수 있으므로 필요한 경우 에이전트를 수동으로 다시 시작합니다.

- 2 적어도 하나 이상의 ping 보고가 완료될 때까지 기다립니다. ping 보고가 완료되려면 1~2분 정도가 소요됩니다.
- 3 vSphere 에이전트가 데이터 수집을 시작하면 vRealize Automation에 IaaS 관리자로 로그인합니다.
- 4 **인프라 > 끝점 > 끝점**을 클릭합니다.
- 5 vSphere 끝점을 편집하고 **연결 테스트**를 클릭합니다.
- 6 인증서 프롬프트가 표시되면 **확인**을 클릭하여 인증서를 수락합니다.

인증서 프롬프트가 표시되지 않으면 현재 끝점에 대한 Windows 시스템 호스팅 서비스의 신뢰할 수 있는 루트 인증 기관(예: 프록시 에이전트 시스템 또는 DEM 시스템)에 인증서가 올바르게 저장되어 있을 수 있습니다.

- 7 **확인**을 클릭하여 인증서 수락을 적용하고 끝점을 저장합니다.
- 8 각 vSphere 끝점에 대해 이 절차를 반복합니다.
- 9 각 NSX 끝점에 대해 이 절차를 반복합니다.

연결 테스트 작업이 성공해도 일부 데이터 수집 또는 프로비저닝 작업이 실패하면 끝점 역할을 하는 모든 에이전트 시스템과 모든 DEM 시스템에 동일한 인증서를 설치할 수 있습니다. 또는 기존 시스템에서 인증서를 제거하고 실패한 끝점에 대해 위의 절차를 반복할 수 있습니다.

대상 vRealize Automation 7.3 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집 실행

마이그레이션 후 대상 VMware vRealize™ Automation 7.3 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집을 실행해야 합니다.

이 데이터 수집은 7.1 및 7.2 배포에 대해 vRealize Automation 7.3에서 [로드 밸런서 재구성] 작업을 수행하는 데 필요합니다.

참고 vRealize Automation 6.2.x에서 7.3으로 마이그레이션한 경우에는 이 데이터 수집을 수행할 필요가 없습니다.

필수 조건

- [소스 vRealize Automation 환경에서 NSX 네트워크 및 보안 인벤토리 데이터 수집 실행](#) .
- vRealize Automation 7.3으로 성공적으로 마이그레이션합니다.

프로시저

- ◆ vRealize Automation 7.3으로 마이그레이션하기 전에 대상 vRealize Automation 환경에서 NSX 네트워크 및 보안 인벤토리 데이터를 수집을 실행합니다. 자세한 내용은 vRealize Automation 관리에서 끝점 데이터 수집 수동 시작 항목을 참조하십시오.

고가용성 환경으로 마이그레이션한 후 로드 밸런서 재구성

고가용성 환경으로 마이그레이션할 때는 마이그레이션을 완료한 후 각 로드 밸런서를 재구성해야 합니다.

필수 조건

[vRealize Automation 소스 데이터를 vRealize Automation 7.3 고가용성 환경으로 마이그레이션](#).

프로시저

- ◆ 복제 노드가 수신 트래픽을 수용하도록 원래 상태 점검 설정을 복원하려면 다음 항목에 대해 로드 밸런서를 구성합니다.
 - vRealize Automation 장치
 - Model Manager를 호스팅하는 IaaS 웹 서버
 - Manager Service

외부 vRealize Orchestrator 서버를 vRealize Automation 7.3 으로 마이그레이션

기존 외부 vRealize Orchestrator 서버에서 vRealize Automation에 포함된 vRealize Orchestrator 인스턴스로 마이그레이션할 수 있습니다.

vRealize Orchestrator를 외부 서버 인스턴스로 배포하고 vRealize Automation을 구성하여 해당 외부 인스턴스로 작업하거나 vRealize Automation 장치에 포함된 vRealize Orchestrator 서버를 구성하고 사용할 수 있습니다.

VMware는 외부 vRealize Orchestrator를 vRealize Automation에 내장된 Orchestrator 서버로 마이그레이션할 것을 권장합니다. 외부에서 포함된 Orchestrator로의 마이그레이션은 다음과 같은 이점을 제공합니다.

- 총 소유 비용을 절감합니다.
- 배포 모델을 간소화합니다.

- 운영 효율성을 개선합니다.

참고 다음 경우에 외부 vRealize Orchestrator의 사용을 고려하십시오.

- vRealize Automation 환경의 여러 테넌트
- 지리적으로 분산된 환경
- 워크로드 처리
- Site Recovery Manager 플러그인과 같은 특정 플러그인 사용

외부 및 내장된 Orchestrator 간의 제어 센터 차이점

외부 vRealize Orchestrator의 제어 센터에서 사용 가능한 메뉴 항목 중 일부는 내장된 Orchestrator 인스턴스의 기본 제어 센터 보기에 포함되어 있지 않습니다.

내장된 Orchestrator 서버의 제어 센터에 몇 가지 옵션은 기본적으로 숨겨져 있습니다.

메뉴 항목	세부 정보
라이센싱	내장된 Orchestrator는 vRealize Automation을 라이선스 제공자로 사용하도록 미리 구성되어 있습니다.
구성 내보내기/가져오기	내장된 Orchestrator 구성은 내보낸 vRealize Automation 구성 요소에 포함되어 있습니다.
데이터베이스 구성	내장된 Orchestrator는 vRealize Automation에서 사용하는 데이터베이스를 사용합니다.
고객 환경 항상 프로그램 (CEIP)	vRealize Automation 장치 관리 인터페이스에서 고객 환경 항상 프로그램(CEIP)에 가입할 수 있습니다. 고객 환경 항상 프로그램(vRealize Automation 관리에 있음)을 참조하십시오.

기본 제어 센터 보기에 숨겨진 다른 옵션은 **인증 제공자 구성** 페이지의 **호스트 주소** 텍스트 상자 및 **등록 취소** 버튼입니다.

참고 vRealize Automation에 내장된 vRealize Orchestrator에서 제어 센터 옵션의 전체 집합을 보려면 고급 Orchestrator 관리 페이지(https://vra-va-hostname.domain.name_or_load_balancer_address:8283/vco-controlcenter/#/?advanced)에 액세스하여 키보드의 F5 버튼을 클릭해 페이지를 새로 고쳐야 합니다.

Windows용 외부 vRealize Orchestrator 6.x를 vRealize Automation 7.3으로 마이그레이션

vRealize Automation을 6.x 버전에서 7.3 버전으로 업그레이드한 후에 Windows에 설치되어 있는 기존의 외부 Orchestrator 6.x를 vRealize Automation 7.3에 내장된 Orchestrator 서버로 마이그레이션할 수 있습니다.

참고 여러 개의 vRealize Automation 장치 노드가 있는 분산된 vRealize Automation 환경을 사용하는 경우, 기본 vRealize Automation 노드에서만 마이그레이션 절차를 수행하십시오.

필수 조건

- vRealize Automation 7.3으로 마이그레이션을 완료했습니다.

- 외부 Orchestrator의 Orchestrator 서버 서비스를 중지합니다.
- 외부 Orchestrator 서버의 데이터베이스를 백업합니다(데이터베이스 스키마 포함).

프로시저

- 1 대상 Orchestrator 서버에서 마이그레이션 도구를 다운로드합니다.
 - a SSH를 통해 vRealize Automation 장치에 **루트**로 로그인합니다.
 - b migration-tool.zip 아카이브(/var/lib/vco/downloads 디렉토리에 있음)를 다운로드합니다.
- 2 소스 Orchestrator 서버에서 Orchestrator 구성을 내보냅니다.
 - a PATH 환경 변수가 Orchestrator와 함께 설치된 Java JRE의 bin 폴더를 가리키도록 설정합니다.
 - b 마이그레이션 도구를 외부 Orchestrator가 설치되어 있는 Windows 서버에 업로드합니다.
 - c Orchestrator 설치 폴더에서 다운로드한 아카이브의 압축을 풉니다.
Windows 기반 설치에서 Orchestrator 설치 폴더의 기본 경로는 C:\Program Files\VMware\Orchestrator입니다.
 - d Windows 명령 프롬프트를 관리자로 실행하고 Orchestrator 설치 폴더의 bin 폴더로 이동합니다.
기본적으로 bin 폴더의 경로는 C:\Program Files\VMware\Orchestrator\migration-cli\bin입니다.
 - e 명령줄에서 export 명령을 실행합니다.

```
C:\Program Files\VMware\Orchestrator\migration-cli\bin\vro-migrate.bat export
```

이 명령은 VMware vRealize Orchestrator 구성 파일과 플러그인을 내보내기 아카이브로 합쳐 줍니다.

아카이브는 migration-cli 폴더와 동일한 폴더에 생성됩니다.

3 내보낸 구성을 vRealize Automation 7.3에 내장된 Orchestrator 서버로 마이그레이션합니다.

- a 내보낸 구성 파일을 vRealize Automation 장치의 `/usr/lib/vco/tools/configuration-cli/bin` 디렉토리에 업로드합니다.
- b `/usr/lib/vco/tools/configuration-cli/bin` 디렉토리에서 내보낸 Orchestrator 구성 파일의 소유권을 변경합니다.

```
chown vco:vco orchestrator-config-export-orchestrator_ip_address-date_hour.zip
```

- c `import` 명령으로 `vro-configure` 스크립트를 실행하여 Orchestrator 구성 파일을 내장된 vRealize Orchestrator 서버로 가져옵니다.

```
./vro-configure.sh import --skipDatabaseSettings --skipLicense --skipSettings --skipSslCertificate --notForceImportPlugins --notRemoveMissingPlugins --skipTrustStore --path orchestrator-config-export-orchestrator_appliance_ip-date_hour.zip
```

4 `db-migrate` 명령으로 `vro-configure` 스크립트를 실행하여 데이터베이스를 내부 PostgreSQL 데이터베이스로 마이그레이션합니다.

```
./vro-configure.sh db-migrate --sourceJdbcUrl JDBC_connection_URL --sourceDbUsername database_user --sourceDbPassword database_user_password
```

참고 특수 문자를 포함하는 암호는 작은따옴표로 묶어야 합니다.

JDBC_connection_URL은 사용하는 데이터베이스 유형에 따라 결정됩니다.

```
PostgreSQL: jdbc:postgresql://host:port/database_name
```

```
MSSQL: jdbc:jtds:sqlserver://host:port/database_name#; if using SQL authentication and MSSQL:
jdbc:jtds:sqlserver://host:port/database_name#;domain=domain#;useNTLMv2=TRUE if using Windows authentication.
```

```
Oracle: jdbc:oracle:thin:@host:port:database_name
```

기본 데이터베이스 로그인 정보는 다음과 같습니다.

database_name	vmware
database_user	vmware
database_user_password	vmware

5 vRealize Automation을 업그레이드 대신 마이그레이션한 경우 내장된 Orchestrator 인스턴스의 데이터베이스에서 신뢰할 수 있는 Single Sign-On 인증서를 삭제합니다.

```
sudo -u postgres -i -- /opt/vmware/vpostgres/current/bin/psql vcac -c "DELETE FROM vmo_keystore WHERE id='cakeystore-id';"
```

Windows에 설치되어 있는 외부 vRealize Orchestrator 6.x를 vRealize Orchestrator 인스턴스(vRealize Automation 7.3에 내장)로 성공적으로 마이그레이션했습니다.

후속 작업

내장된 vRealize Orchestrator 서버를 설정합니다. [내장된 vRealize Orchestrator 서버 구성](#) 항목을 참조하십시오.

외부 vRealize Orchestrator 6.x 가상 장치를 vRealize Automation 7.3으로 마이그레이션

vRealize Automation을 6.x 버전에서 7.3 버전으로 업그레이드한 후에 기존의 외부 Orchestrator 6.x 가상 장치를 vRealize Automation 7.3에 내장된 Orchestrator 서버로 마이그레이션할 수 있습니다.

참고 여러 개의 vRealize Automation 장치 노드가 있는 분산된 vRealize Automation 환경을 사용하는 경우, 기본 vRealize Automation 노드에서만 마이그레이션 절차를 수행하십시오.

필수 조건

- vRealize Automation 7.3으로 마이그레이션을 완료했습니다.
- 외부 Orchestrator의 Orchestrator 서버 서비스를 중지합니다.
- 외부 Orchestrator 서버의 데이터베이스를 백업합니다(데이터베이스 스키마 포함).

프로시저

- 1 대상 Orchestrator 서버에서 소스 Orchestrator로 마이그레이션 도구를 다운로드합니다.
 - a SSH를 통해 vRealize Orchestrator 6.x 가상 장치에 **루트**로 로그인합니다.
 - b `/var/lib/vco` 디렉토리에서 `scp` 명령을 실행하여 `migration-tool.zip` 아카이브를 다운로드합니다.

```
scp root@vra-va-hostname.domain.name:/var/lib/vco/downloads/migration-tool.zip ./
```

- c `unzip` 명령을 실행하여 마이그레이션 도구 아카이브의 압축을 풉니다.

```
unzip migration-tool.zipy7
```

2 소스 Orchestrator 서버에서 Orchestrator 구성을 내보냅니다.

- a /var/lib/vco/migration-cli/bin 디렉토리에서 export 명령을 실행합니다.

```
./vro-migrate.sh export
```

이 명령은 VMware vRealize Orchestrator 구성 파일과 플러그인을 내보내기 아카이브로 합쳐 줍니다.

파일 이름이 orchestrator-config-export-*orchestrator_ip_address-date_hour.zip*인 아카이브가 /var/lib/vco 폴더에 생성됩니다.

3 내보낸 구성을 vRealize Automation 7.3에 내장된 Orchestrator 서버로 마이그레이션합니다.

- a SSH를 통해 vRealize Automation 장치에 **루트**로 로그인합니다.
- b /usr/lib/vco/tools/configuration-cli/bin 디렉토리에서 scp 명령을 실행하여 내보낸 구성 아카이브를 다운로드합니다.

```
scp root@orchestrator_ip_or_DNS_name:/var/lib/vco/orchestrator-config-export-orchestrator_ip_address-date_hour.zip ./
```

- c 내보낸 Orchestrator 구성 파일의 소유권을 변경합니다.

```
chown vco:vco orchestrator-config-export-orchestrator_ip_address-date_hour.zip
```

- d Orchestrator 서버 서비스 및 내장된 vRealize Orchestrator 서버의 제어 센터 서비스를 중지합니다.

```
service vco-server stop && service vco-configurator stop
```

- e import 명령으로 vro-configure 스크립트를 실행하여 Orchestrator 구성 파일을 내장된 vRealize Orchestrator 서버로 가져옵니다.

```
./vro-configure.sh import --skipDatabaseSettings --skipLicense --skipSettings --skipSslCertificate --notForcelImportPlugins --notRemoveMissingPlugins --skipTrustStore --path orchestrator-config-export-orchestrator_appliance_ip-date_hour.zip
```

4 마이그레이션을 시작하려는 외부 Orchestrator 서버에서 내장된 PostgreSQL 데이터베이스를 사용하는 경우, 해당 데이터베이스 구성 파일을 편집합니다.

- a /var/vmware/vpostgres/current/pgdata/postgresql.conf 파일에서 listen_addresses 줄의 주석 처리를 제거합니다.
- b listen_addresses의 값을 와일드카드(*)로 설정합니다.

```
listen_addresses = '*'
```


- c 다음 줄을 `/var/vmware/vpostgres/current/pgdata/pg_hba.conf` 파일에 추가합니다.

```
host all all vra-va-ip-address/32 md5
```

참고 `pg_hba.conf` 파일은 IP 주소 및 서브넷 마스크에서 CIDR 접두사 형식을 대신 사용해야 합니다.

- d PostgreSQL 서버 서비스를 다시 시작합니다.

```
service vpostgres restart
```

- 5 `db-migrate` 명령으로 `vro-configure` 스크립트를 실행하여 데이터베이스를 내부 PostgreSQL 데이터베이스로 마이그레이션합니다.

```
./vro-configure.sh db-migrate --sourceJdbcUrl Jdbc_connection_URL --sourceDbUsername database_user --sourceDbPassword database_user_password
```

참고 특수 문자를 포함하는 암호는 작은따옴표로 묶어야 합니다.

Jdbc_connection_URL은 사용하는 데이터베이스 유형에 따라 결정됩니다.

```
PostgreSQL: jdbc:postgresql://host:port/database_name
```

```
MSSQL: jdbc:jtds:sqlserver://host:port/database_name#; if using SQL authentication and MSSQL:
jdbc:jtds:sqlserver://host:port/database_name#;domain=domain#;useNTLMv2=TRUE if using Windows authentication.
```

```
Oracle: jdbc:oracle:thin:@host:port:database_name
```

기본 데이터베이스 로그인 정보는 다음과 같습니다.

database_name	vmware
database_user	vmware
database_user_password	vmware

- 6 vRealize Automation을 업그레이드 대신 마이그레이션한 경우 내장된 Orchestrator 인스턴스의 데이터베이스에서 신뢰할 수 있는 Single Sign-On 인증서를 삭제합니다.

```
sudo -u postgres -i -- /opt/vmware/vpostgres/current/bin/psql vcac -c "DELETE FROM vmo_keystore WHERE id='cakeystore-id';"
```

- 7 `postgresql.conf` 및 `pg_hba.conf` 파일의 기본 구성으로 되돌립니다.

- a PostgreSQL 서버 서비스를 다시 시작합니다.

외부 vRealize Orchestrator 6.x 가상 장치를 vRealize Orchestrator 인스턴스 (vRealize Automation 7.3에 내장)로 성공적으로 마이그레이션했습니다.

후속 작업

내장된 vRealize Orchestrator 서버를 설정합니다. [내장된 vRealize Orchestrator 서버 구성 항목](#)을 참조하십시오.

외부 vRealize Orchestrator 7.x를 vRealize Automation 7.3으로 마이그레이션

기존의 외부 Orchestrator 인스턴스의 구성을 내보내고 vRealize Automation에 내장된 Orchestrator 서버로 가져올 수 있습니다.

참고 여러 개의 vRealize Automation 장치 노드가 있는 경우 기본 vRealize Automation 노드에서만 마이그레이션 절차를 수행하십시오.

필수 조건

- vRealize Automation 7.3으로 마이그레이션을 완료했습니다.
- 외부 Orchestrator의 Orchestrator 서버 서비스를 중지합니다.
- 외부 Orchestrator 서버의 데이터베이스를 백업합니다(데이터베이스 스키마 포함).

프로시저

- 1 외부 Orchestrator 서버의 구성을 내보냅니다.
 - a 소스 버전에 따라 외부 Orchestrator 서버의 제어 센터에 **루트** 또는 **관리자**로 로그인합니다.
 - b **시작 옵션** 페이지에서 Orchestrator 서버 서비스를 중지하여 원치 않는 변경이 데이터베이스에 발생하지 않도록 합니다.
 - c **구성 내보내기/가져오기** 페이지로 이동합니다.
 - d **구성 내보내기** 페이지에서 **서버 구성 내보내기**, **번들 플러그인**, **플러그인 구성 내보내기**를 선택합니다.
- 2 내보낸 구성을 내장된 Orchestrator 인스턴스로 마이그레이션합니다.
 - a 내보낸 Orchestrator 구성 파일을 vRealize Automation 장치의 `/usr/lib/vco/tools/configuration-cli/bin` 디렉토리로 업로드합니다.
 - b SSH를 통해 vRealize Automation 장치에 **루트**로 로그인합니다.
 - c Orchestrator 서버 서비스 및 내장된 vRealize Orchestrator 서버의 제어 센터 서비스를 중지합니다.

```
service vco-server stop && service vco-configurator stop
```
 - d `/usr/lib/vco/tools/configuration-cli/bin` 디렉토리로 이동합니다.

- e 내보낸 Orchestrator 구성 파일의 소유권을 변경합니다.

```
chown vco:vco orchestrator-config-export-orchestrator_appliance_ip-date_hour.zip
```

- f import 명령으로 vro-configure 스크립트를 실행하여 Orchestrator 구성 파일을 내장된 vRealize Orchestrator 서버로 가져옵니다.

```
./vro-configure.sh import --skipDatabaseSettings --skipLicense --skipSettings --skipSslCertificate --notForceImportPlugins --notRemoveMissingPlugins --skipTrustStore --path orchestrator-config-export-orchestrator_appliance_ip-date_hour.zip
```

- 3 마이그레이션을 시작하려는 외부 Orchestrator 서버에서 내장된 PostgreSQL 데이터베이스를 사용하는 경우, 해당 데이터베이스 구성 파일을 편집합니다.

- a /var/vmware/vpostgres/current/pgdata/postgresql.conf 파일에서 listen_addresses 줄의 주석 처리를 제거합니다.
- b listen_addresses의 값을 와일드카드(*)로 설정합니다.

```
listen_addresses = '*'
```

- c 다음 줄을 /var/vmware/vpostgres/current/pgdata/pg_hba.conf 파일에 추가합니다.

```
host all all vra-va-ip-address/32 md5
```

참고 pg_hba.conf 파일은 IP 주소 및 서브넷 마스크에서 CIDR 접두사 형식을 대신 사용해야 합니다.

- d PostgreSQL 서버 서비스를 다시 시작합니다.

```
service vpostgres restart
```

- 4 db-migrate 명령으로 vro-configure 스크립트를 실행하여 데이터베이스를 내부 PostgreSQL 데이터베이스로 마이그레이션합니다.

```
./vro-configure.sh db-migrate --sourceJdbcUrl JDBC_connection_URL --sourceDbUsername database_user --sourceDbPassword database_user_password
```

참고 특수 문자를 포함하는 암호는 작은따옴표로 묶어야 합니다.

JDBC_connection_URL은 사용하는 데이터베이스 유형에 따라 결정됩니다.

```
PostgreSQL: jdbc:postgresql://host:port/database_name
```

```
MSSQL: jdbc:jtds:sqlserver://host:port/database_name;; if using SQL authentication and MSSQL:
jdbc:jtds:sqlserver://host:port/database_name;;domain=domain;;useNTLMv2=TRUE if using Windows authentication.
```

```
Oracle: jdbc:oracle:thin:@host:port:database_name
```

기본 데이터베이스 로그인 정보는 다음과 같습니다.

database_name	vmware
database_user	vmware
database_user_password	vmware

- 5 vRealize Automation을 업그레이드 대신 마이그레이션한 경우 내장된 Orchestrator 인스턴스의 데이터베이스에서 신뢰할 수 있는 Single Sign-On 인증서를 삭제합니다.

```
sudo -u postgres -i -- /opt/vmware/vpostgres/current/bin/psql vcac -c "DELETE FROM vmo_keystore WHERE id='cakeystore-id';"
```

- 6 postgresql.conf 및 pg_hba.conf 파일의 기본 구성으로 되돌립니다.

- a PostgreSQL 서버 서비스를 다시 시작합니다.

외부 Orchestrator 서버 인스턴스를 vRealize Orchestrator 인스턴스(vRealize Automation에 내장)로 성공적으로 마이그레이션했습니다.

후속 작업

내장된 vRealize Orchestrator 서버를 설정합니다. [내장된 vRealize Orchestrator 서버 구성](#) 항목을 참조하십시오.

내장된 vRealize Orchestrator 서버 구성

외부 Orchestrator 서버의 구성을 내보내고 vRealize Automation 7.3으로 가져온 후에는 vRealize Automation에 내장된 Orchestrator 서버를 구성해야 합니다.

필수 조건

외부에서 내부 vRealize Orchestrator로 구성을 마이그레이션합니다.

프로시저

- 1 SSH를 통해 vRealize Automation 장치에 **루트**로 로그인합니다.
- 2 내장된 vRealize Orchestrator 서버의 제어 센터 서비스 및 Orchestrator 서버 서비스를 시작합니다.

```
service vco-configurator start && service vco-server start
```

- 3 내장된 Orchestrator 서버의 제어 센터에 **관리자**로 로그인합니다.

참고 외부 vRealize Orchestrator 7.3 인스턴스에서 마이그레이션하는 경우 5단계로 건너뛸 수 있습니다.

- 4 제어 센터의 **구성 검증** 페이지에서 Orchestrator가 올바르게 구성되었는지 확인합니다.
- 5 외부 Orchestrator가 클러스터 모드에서 작동하도록 구성된 경우 vRealize Automation에서 Orchestrator 클러스터를 재구성합니다.
 - a 고급 **Orchestrator 클러스터 관리** 페이지인 https://vra-va-hostname.domain.name_or_load_balancer_address:8283/vco-controlcenter/#/control-app/ha?remove-nodes로 이동합니다.

참고 클러스터에서 기존 노드 옆에 **제거** 확인란이 나타나지 않는 경우, 키보드에서 F5 버튼을 클릭하여 브라우저 페이지를 새로 고침해야 합니다.

- b 외부 Orchestrator 노드 옆에 있는 확인란을 선택하고 **제거**를 클릭하여 이 노드를 클러스터에서 제거합니다.
 - c 고급 클러스터 관리 페이지에서 나가려면 URL에서 &remove-nodes 문자열을 삭제하고 키보드에서 F5 버튼을 클릭하여 브라우저 페이지를 새로 고칩니다.
 - d 제어 센터의 **구성 검증** 페이지에서 Orchestrator가 제대로 구성되었는지 확인합니다.
- 6 (선택 사항) **패키지 서명 인증서** 탭(인증서 페이지에 있음) 아래에서 새 패키지 서명 인증서를 생성합니다.
- 7 (선택 사항) **기본 테넌트 및 기본 그룹(인증 제공자 구성** 페이지에 있음)의 값을 변경합니다.
- 8 vco-server 서비스가 등록됨으로 나타나는지 **서비스** 탭(vRealize Automation 장치 관리 콘솔에 있음) 아래에서 확인합니다.
- 9 외부 Orchestrator 서버의 vco 서비스를 선택하고 **등록 취소**를 클릭합니다.

후속 작업

- 외부 Orchestrator 서버에서 신뢰하는 모든 인증서를 내장된 Orchestrator의 신뢰 저장소로 가져옵니다.

- Orchestrator 구성을 동기화하려면 vRealize Automation 복제 노드를 vRealize Automation 클러스터에 결합합니다.

자세한 내용은 고가용성을 지원하도록 내장된 대상 vRealize Orchestrator 재구성(vRealize Automation 설치 또는 업그레이드에 있음)을 참조하십시오.

참고 vRealize Orchestrator 인스턴스는 자동으로 클러스터링되며 사용 가능하게 됩니다.

- 클러스터에 있는 모든 노드에서 vco-configurator 서비스를 다시 시작합니다.
- 마이그레이션한 내장된 Orchestrator 서버로 이동하도록 vRealize Orchestrator 끝점을 업데이트합니다.
- vRA 호스트 추가 및 vRA 호스트 워크플로의 IaaS 호스트 추가를 실행하여 vRealize Automation 호스트 및 IaaS 호스트를 vRealize Automation 플러그인의 인벤토리에 추가합니다.

vRealize Automation 7.x에서 7.3으로 포함된 vRealize Orchestrator 서버 마이그레이션

다음 절차를 수행하여 vRealize Orchestrator 서버를 vRealize Automation 7.x 소스 환경에서 vRealize Automation 7.3으로 마이그레이션할 수 있습니다.

필수 조건

vRealize Automation 7.3으로 마이그레이션을 완료했습니다.

프로시저

1 일시적으로 소스 vRealize Automation 장치의 구성 변경

vRealize Orchestrator 서버를 vRealize Automation 7.x 소스 환경에서 vRealize Automation 7.3으로 마이그레이션하기 전에 이 절차에 나와 있는 명령을 실행하여 소스 vRealize Automation 장치의 구성을 일시적으로 변경해야 합니다.

2 소스 vRealize Automation 장치의 포함된 vRealize Orchestrator에서 구성 내보내기

vRealize Orchestrator 서버를 vRealize Automation 7.x 소스 환경에서 vRealize Automation 7.3으로 마이그레이션하기 전에 이 절차에 나와 있는 명령을 실행하여, 포함된 소스 vRealize Orchestrator의 구성을 내보내야 합니다.

3 포함된 소스 vRealize Orchestrator의 구성과 데이터베이스를 포함된 대상 vRealize Orchestrator로 가져오기

이 절차에 나와 있는 명령을 실행하여 vRealize Automation 7.x 소스 환경에서 vRealize Automation 7.3으로 vRealize Orchestrator 서버를 마이그레이션합니다.

4 고가용성을 지원하도록 포함된 대상 vRealize Orchestrator 재구성

고가용성 배포의 경우, 포함된 vRealize Orchestrator에 대해 고가용성을 사용하려면 각 대상 복제 vRealize Automation 장치를 클러스터에 수동으로 다시 가입해야 합니다.

5 소스 vRealize Automation 장치의 구성 복원

소스 vRealize Automation 장치의 구성을 복원하려면 다음 절차를 사용합니다.

일시적으로 소스 vRealize Automation 장치의 구성 변경

vRealize Orchestrator 서버를 vRealize Automation 7.x 소스 환경에서 vRealize Automation 7.3으로 마이그레이션하기 전에 이 절차에 나와 있는 명령을 실행하여 소스 vRealize Automation 장치의 구성을 일시적으로 변경해야 합니다.

필수 조건

- 최소 배포의 경우 SSH를 사용하여 소스 vRealize Automation 장치에 **root**로 로그인합니다.
- 고가용성 배포의 경우 SSH를 사용하여 마스터 소스 vRealize Automation 장치에 **root**로 로그인합니다.

프로시저

1 소스 PostgreSQL 서버에 vro_migration 사용자를 생성합니다.

- a 명령을 실행하기 전에 VRO-MIGRATION-USER-PASSWORD를 vro_migration 사용자의 암호로 바꿉니다.

```
sudo -u postgres -i -- /opt/vmware/vpostgres/current/bin/psql vcac
-c "CREATE USER vro_migration WITH PASSWORD
'VRO-MIGRATION-USER-PASSWORD';"
```

- b vro_migration 사용자에게 vcac 데이터베이스의 테이블에 대한 액세스 권한을 부여합니다.

```
sudo -u postgres -i -- /opt/vmware/vpostgres/current/bin/psql vcac
-c "GRANT SELECT ON ALL TABLES IN SCHEMA public TO vro_migration;"
```

2 소스 PostgreSQL 클라이언트 인증 구성 파일(/storage/db/pgdata/pg_hba.conf)의 백업을 생성합니다.

```
cp /storage/db/pgdata/pg_hba.conf /storage/db/pgdata/pg_hba.conf.bak
```

3 대상 vRealize Automation 장치에서 vcac 데이터베이스에 원격으로 액세스할 수 있는 권한을 vro_migration 사용자에게 부여하도록 소스 PostgreSQL 클라이언트 인증 구성 파일을 수정합니다. 명령을 실행하기 전에 TARGET-VRA-APPLIANCE-IPV4-ADDRESS를 대상 vRealize Automation 장치의 IP v4 주소로 바꿉니다.

```
echo "host vcac vro_migration TARGET-VRA-APPLIANCE-IPV4-ADDRESS/32 md5" >> /storage/db/pgdata/pg_hba.conf
```

4 소스 PostgreSQL 서버를 다시 시작합니다.

```
service vpostgres restart
```

후속 작업

[소스 vRealize Automation 장치의 포함된 vRealize Orchestrator에서 구성 내보내기](#)

소스 vRealize Automation 장치의 포함된 vRealize Orchestrator 에서 구성 내보내기

vRealize Orchestrator 서버를 vRealize Automation 7.x 소스 환경에서 vRealize Automation 7.3으로 마이그레이션하기 전에 이 절차에 나와 있는 명령을 실행하여, 포함된 소스 vRealize Orchestrator의 구성을 내보내야 합니다.

필수 조건

- 최소 배포의 경우 SSH를 사용하여 소스 vRealize Automation 장치에 **root**로 로그인합니다.
- 고가용성 배포의 경우 SSH를 사용하여 마스터 소스 vRealize Automation 장치에 **root**로 로그인합니다.

프로시저

- 1 vRealize Orchestrator 마이그레이션 도구를 대상 vRealize Automation 7.3 환경에서 소스 vRealize Automation 7.x 장치로 복사합니다. 명령을 실행하기 전에 TARGET-VRA-APPLIANCE-HOST-NAME을 대상 vRealize Automation 장치 호스트의 정규화된 도메인 이름으로 바꿉니다.

```
scp root@TARGET-VRA-APPLIANCE-HOST-NAME:/var/lib/vco/downloads/migration-tool.zip /storage
```

- 2 vRealize Automation 소스 장치에서 마이그레이션 도구를 추출합니다.

```
unzip /storage/migration-tool.zip -d /var/lib/vco
```

- 3 마이그레이션 도구를 실행합니다.

```
/var/lib/vco/migration-cli/bin/vro-migrate.sh export
```

마이그레이션 도구가 /var/lib/vco/orchestrator-config-export-localhost-DATE-AND-TIMESTAMP.zip 파일을 생성합니다.

참고 이 명령을 실행할 때 표시되는 SLF4J 오류 메시지는 무시해도 됩니다.

- 4 생성된 zip 파일을 나중에 사용할 수 있게 /tmp/vro-config.zip에 복사합니다.

```
mv /var/lib/vco/orchestrator-config-export-localhost-DATE-AND-TIMESTAMP.zip /tmp/vro-config.zip
```

후속 작업

[포함된 소스 vRealize Orchestrator의 구성과 데이터베이스를 포함된 대상 vRealize Orchestrator로 가져오기](#)

포함된 소스 vRealize Orchestrator 의 구성과 데이터베이스를 포함 된 대상 vRealize Orchestrator 로 가져오기

이 절차에 나와 있는 명령을 실행하여 vRealize Automation 7.x 소스 환경에서 vRealize Automation 7.3으로 vRealize Orchestrator 서버를 마이그레이션합니다.

필수 조건

- 최소 배포의 경우 SSH를 사용하여 소스 vRealize Automation 장치에 **root**로 로그인합니다.
- 고가용성 배포의 경우 SSH를 사용하여 마스터 소스 vRealize Automation 장치에 **root**로 로그인합니다.

프로시저

- 1 vRealize Orchestrator 서버 서비스를 중지합니다.

```
service vco-server stop
```

- 2 vRealize Orchestrator 제어 센터 서비스를 중지합니다.

```
service vco-configurator stop
```

고가용성 배포의 경우 마스터 vRealize Automation 장치 및 각 복제 장치에서 vRealize Orchestrator 서버 서비스와 vRealize Orchestrator 제어 센터 서비스를 중지합니다.

- 3 소스 vRealize Automation 장치에서 vro-config.zip을 대상 vRealize Automation 장치의 /tmp 디렉토리로 복사합니다. 명령을 실행하기 전에 SOURCE-VRA-APPLIANCE-HOST-NAME을 소스 vRealize Automation 장치의 정규화된 도메인 이름으로 바꿉니다. 메시지가 표시되면 소스 vRealize Automation 장치 루트 사용자의 암호를 입력합니다.

```
scp root@SOURCE-VRA-APPLIANCE-HOST-NAME:/tmp/vro-config.zip /tmp/vro-config.zip
```

- 4 /tmp/vro-config.zip의 소유권을 변경합니다.

```
chown vco:vco /tmp/vro-config.zip
```

5 구성 파일을 포함한 대상 vRealize Orchestrator 서버로 가져옵니다.

```
/usr/lib/vco/tools/configuration-cli/bin/vro-configure.sh import --skipDatabaseSettings --skipLicense --
skipSettings --skipSslCertificate --skipTrustStore --notForceImportPlugins --notRemoveMissingPlugins --
path /tmp/vro-config.zip
```

구성 파일 가져오기가 완료되면 Orchestrator 구성 가져오기가 완료되었습니다. 메시지가 나타납니다.

참고 가져오기가 성공적으로 완료된 경우에는 이 명령을 실행할 때 표시되는 모든 [ERROR] 또는 [WARN] 메시지를 무시해도 됩니다.

6 소스 vRealize Orchestrator 데이터베이스를 대상 vRealize Automation 장치에서 실행 중인 PostgreSQL 서버로 마이그레이션합니다. 명령을 실행하기 전에 SOURCE-VRA-APPLIANCE-HOST-NAME을 소스 vRealize Automation 장치의 정규화된 도메인 이름으로 바꾸고, VRO-MIGRATION-USER-PASSWORD를 일시적으로 소스 vRealize Automation 장치의 구성 변경 절차에서 생성한 vro_migration 사용자 암호로 바꿉니다.

```
/usr/lib/vco/tools/configuration-cli/bin/vro-configure.sh db-migrate --sourceJdbcUrl jdbc:postgresql://SOURCE-
VRA-APPLIANCE-HOST-NAME:5432/vcac --sourceDbUsername vro_migration --sourceDbPassword VRO-MIGRATION-USER-
PASSWORD
```

7 마이그레이션된 데이터베이스에서 이전의 신뢰할 수 있는 인증서를 삭제합니다.

```
sudo -u postgres -i -- /opt/vmware/vpostgres/current/bin/psql vcac -c "DELETE FROM vmo_keystore WHERE
id='cakeystore-id';"
```

8 마이그레이션된 데이터베이스에서 이전 vRealize Orchestrator 노드를 삭제합니다.

```
sudo -u postgres -i -- /opt/vmware/vpostgres/current/bin/psql vcac -c "DELETE FROM vmo_clustermember;"
```

9 /tmp 디렉토리에서 vro-config.zip을 삭제합니다.

```
rm -rf /tmp/vro-config.zip
```

10 vRealize Orchestrator 서버 서비스를 시작합니다.

```
service vco-server start
```

고가용성 배포의 경우 마스터 vRealize Automation 장치에서만 vRealize Orchestrator 서버 서비스를 시작하십시오.

후속 작업

[고가용성을 지원하도록 포함된 대상 vRealize Orchestrator 재구성](#)

고가용성을 지원하도록 포함된 대상 vRealize Orchestrator 재구성

고가용성 배포의 경우, 포함된 vRealize Orchestrator에 대해 고가용성을 사용하려면 각 대상 복제 vRealize Automation 장치를 클러스터에 수동으로 다시 가입해야 합니다.

필수 조건

대상 복제 vRealize Automation 장치 관리에 로그인합니다.

- 1 브라우저를 시작하고, 대상 복제 가상 장치의 FQDN(정규화된 도메인 이름)인 `https://vra-va-hostname.domain.name:5480`을 사용하여 대상 복제 vRealize Automation 장치 관리를 엽니다.
- 2 `root`라는 사용자 이름과 대상 복제 vRealize Automation 장치를 배포할 때 입력한 암호를 사용하여 로그인합니다.

프로시저

- 1 **vRA 설정 > 클러스터**를 선택합니다.
- 2 **선행 클러스터 노드** 텍스트 상자에 대상 마스터 vRealize Automation 장치의 FQDN(정규화된 도메인 이름)을 입력합니다.
- 3 **암호** 텍스트 상자에 루트 사용자 암호를 입력합니다.
- 4 **클러스터에 가입**을 클릭합니다.
모든 인증서 경고를 무시하고 계속합니다. 클러스터의 서비스가 다시 시작됩니다.
- 5 서비스가 실행 중인지 확인합니다.
 - a 상단 탭 모음에서 **서비스**를 클릭합니다.
 - b **새로 고침**을 클릭하여 서비스 시작 진행률을 모니터링합니다.

후속 작업

[소스 vRealize Automation 장치의 구성 복원](#)

소스 vRealize Automation 장치의 구성 복원

소스 vRealize Automation 장치의 구성을 복원하려면 다음 절차를 사용합니다.

필수 조건

- 최소 배포의 경우 SSH를 사용하여 소스 vRealize Automation 장치에 `root`로 로그인합니다.
- 고가용성 배포의 경우 SSH를 사용하여 마스터 소스 vRealize Automation 장치에 `root`로 로그인합니다.

프로시저

- 1 `/tmp` 디렉토리에서 `vro-config.zip`을 삭제합니다.

```
rm -rf /tmp/vro-config.zip
```

- 이전에 추가한 줄을 소스 PostgreSQL 클라이언트 인증 구성 파일에서 제거하여 vcac 데이터베이스에 대한 vro_migration 사용자의 원격 액세스 권한을 취소합니다.

```
sed -i '/vro_migration/d' /storage/db/pgdata/pg_hba.conf
```

- PostgreSQL 서버를 다시 시작합니다.

```
service vpostgres restart
```

- 소스 PostgreSQL 데이터베이스에서 vro_migration 사용자를 삭제합니다.

- vcac 데이터베이스의 테이블에 대한 vro_migration 사용자 액세스 권한을 취소합니다.

```
sudo -u postgres -i -- /opt/vmware/vpostgres/current/bin/psql vcac -c "REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA public FROM vro_migration;"
```

- 소스 PostgreSQL 서버에서 vro_migration 사용자를 제거합니다.

```
sudo -u postgres -i -- /opt/vmware/vpostgres/current/bin/psql vcac -c "DROP USER vro_migration;"
```

대상 vRealize Orchestrator 에서 vRealize Automation 끝점 재구성

포함된 대상 vRealize Orchestrator에서 vRealize Automation 끝점을 재구성하려면 다음 절차를 따릅니다.

필수 조건

- vRealize Automation 7.3으로 마이그레이션을 완료했습니다.
- vRealize Orchestrator 클라이언트를 사용하여 대상 vRealize Orchestrator에 연결합니다. 자세한 내용은 [vRealize Orchestrator 설명서](#)에서 VMware vRealize Orchestrator Client 사용 항목을 참조하십시오.

프로시저

- 상단 드롭다운 메뉴에서 **설계**를 선택합니다.
- 인벤토리**를 클릭합니다.
- vRealize Automation**을 확장합니다.

- 4 소스 vRealize Automation 장치 호스트의 FQDN(정규화된 도메인 이름) 또는 고가용성 배포에서 마이그레이션한 경우 로드 밸런싱된 호스트의 FQDN(정규화된 도메인 이름)이 포함된 끝점을 식별합니다.

소스 vRealize Automation 장치 호스트의 FQDN 또는 고가용성 배포에서 마이그레이션한 경우 로드 밸런싱된 호스트의 FQDN이 포함된 끝점을 찾은 경우

- 1 워크플로를 클릭합니다.
- 2 확장 버튼을 클릭하고 **라이브러리 > vRealize Automation > 구성**을 선택합니다.
- 3 소스 장치 호스트의 FQDN이 포함된 모든 끝점에 대해 vRA 호스트 제거 vRealize Automation 워크플로를 실행합니다.

소스 vRealize Automation 장치 호스트의 FQDN 또는 고가용성 배포에서 마이그레이션한 경우 로드 밸런싱된 호스트의 FQDN이 포함된 끝점을 찾지 못한 경우

- 1 리소스를 클릭합니다.
- 2 상단 도구 모음에서 업데이트 아이콘을 클릭합니다.
- 3 확장 버튼을 클릭하고 **라이브러리 > vCACCAFE > 구성**을 선택합니다.
- 4 소스 vRealize Automation 장치 호스트의 FQDN 또는 고가용성 배포에서 마이그레이션한 경우 로드 밸런싱된 호스트의 FQDN이 포함된 URL 속성이 있는 각 리소스를 삭제합니다.

- 5 워크플로를 클릭합니다.
- 6 확장 버튼을 클릭하고 **라이브러리 > vRealize Automation > 구성**을 선택합니다.
- 7 대상 vRealize Automation 장치 호스트 또는 고가용성 배포로 마이그레이션한 경우 로드 밸런싱된 호스트를 추가하려면 **구성 요소 레지스트리를 사용하여 vRA 호스트 추가** 워크플로를 실행합니다.

대상 vRealize Orchestrator 에서 vRealize Automation 인프라 끝점 재구성

포함된 대상 vRealize Orchestrator에서 vRealize Automation 인프라 끝점을 재구성하려면 다음 절차를 따릅니다.

필수 조건

- vRealize Automation 7.3으로 마이그레이션을 완료했습니다.
- vRealize Orchestrator 클라이언트를 사용하여 대상 vRealize Orchestrator에 연결합니다. 자세한 내용은 [vRealize Orchestrator 설명서](#)에서 VMware vRealize Orchestrator Client 사용 항목을 참조하십시오.

프로시저

- 1 상단 드롭다운 메뉴에서 **설계**를 선택합니다.
- 2 **인벤토리**를 클릭합니다.
- 3 **vRealize Automation 인프라**를 확장합니다.

- 4 소스 vRealize Automation 인프라 호스트의 FQDN(정규화된 도메인 이름) 또는 고가용성 배포에서 마이그레이션한 경우 로드 밸런싱된 호스트의 FQDN(정규화된 도메인 이름)이 포함된 끝점을 식별합니다.

소스 vRealize Automation 인프라 호스트의 FQDN 또는 고가용성 배포에서 마이그레이션한 경우 로드 밸런싱된 호스트의 FQDN이 포함된 끝점을 찾은 경우	소스 vRealize Automation 인프라 호스트의 FQDN 또는 고가용성 배포에서 마이그레이션한 경우 로드 밸런싱된 호스트의 FQDN이 포함된 끝점을 찾지 못한 경우
<ol style="list-style-type: none"> 1 워크플로를 클릭합니다. 2 확장 버튼을 클릭하고 라이브러리 > vRealize Automation > 인프라 관리 > 구성을 선택합니다. 3 소스 인프라 호스트의 FQDN이 포함된 모든 끝점에 대해 IaaS 호스트 제거 vRealize Automation 워크플로를 실행합니다. 	<ol style="list-style-type: none"> 1 리소스를 클릭합니다. 2 상단 도구 모음에서 업데이트 아이콘을 클릭합니다. 3 확장 버튼을 클릭하고 라이브러리 > vCAC > 구성을 선택합니다. 4 소스 vRealize Automation 인프라 호스트의 FQDN 또는 고가용성 배포에서 마이그레이션한 경우 로드 밸런싱된 호스트의 FQDN이 포함된 host 속성이 있는 각 리소스를 삭제합니다.

- 5 워크플로를 클릭합니다.
- 6 확장 버튼을 클릭하고 라이브러리 > vRealize Automation > 구성을 선택합니다.
- 7 대상 vRealize Automation 인프라 호스트를 추가하려면 또는 고가용성 배포 로드 밸런싱된 호스트로 마이그레이션한 경우 vRA 호스트의 IaaS 호스트 추가 워크플로를 실행합니다.

vRealize Orchestrator 사용자 지정 설치

사용자 지정된 상태 변경 워크플로 스텝과 vRealize Orchestrator 메뉴 작업 워크플로를 설치하는 워크플로를 실행할 수 있습니다.

자세한 내용은 수명 주기 확장성에서 vRealize Orchestrator 사용자 지정 설치 항목을 참조하십시오.

필수 조건

vRealize Automation 7.3으로 마이그레이션을 완료했습니다.

대상 vRealize Automation 에서 포함된 vRealize Orchestrator 인프라 끝점 재구성

vRealize Automation 6.2.x 환경에서 마이그레이션하는 경우에는 포함된 대상 vRealize Orchestrator 서버를 가리키는 인프라 끝점의 URL을 업데이트해야 합니다.

필수 조건

- vRealize Automation 7.3으로 성공적으로 마이그레이션합니다.
- 대상 vRealize Automation 콘솔에 로그인합니다.
 - a 대상 가상 장치의 정규화된 도메인 이름(https://vra-va-hostname.domain.name/vcac)을 사용하여 vRealize Automation 콘솔을 엽니다.
고가용성 환경인 경우, 대상 가상 장치 로드 밸런서의 정규화된 도메인 이름(https://vra-va-lb-hostname.domain.name/vcac)을 사용하여 콘솔을 엽니다.

- b IaaS 관리자 사용자로 로그인합니다.

프로시저

- 1 **인프라 > 끝점 > 끝점**을 선택합니다.
- 2 [끝점] 페이지에서 vRealize Orchestrator 끝점을 선택하고 **편집**을 클릭합니다.
- 3 [주소] 텍스트 상자에서 vRealize Orchestrator 끝점 URL을 편집합니다.
 - 최소 환경으로 마이그레이션한 경우, vRealize Orchestrator 끝점 URL을 `https://vra-va-hostname.domain.name:443/vco`으로 바꿉니다.
 - 고가용성 환경으로 마이그레이션한 경우, vRealize Orchestrator 끝점 URL을 `https://vra-va-lb-hostname.domain.name:443/vco`으로 바꿉니다.
- 4 **확인**을 클릭합니다.
- 5 vRealize Orchestrator 끝점에서 데이터 수집을 수동으로 실행합니다.
 - a [끝점] 페이지에서 vRealize Orchestrator 끝점을 선택합니다.
 - b **작업 > 데이터 수집**을 선택합니다.
데이터 수집이 완료되었는지 확인합니다.

대상 vRealize Automation 환경에서 Azure 끝점 재구성

마이그레이션 이후 Microsoft Azure 끝점을 재구성해야 합니다.

각 Azure 끝점에 대해 이 절차를 수행합니다.

필수 조건

- vRealize Automation 7.3으로 성공적으로 마이그레이션합니다.
- 대상 vRealize Automation 콘솔에 로그인합니다.
 - a 대상 가상 장치의 정규화된 도메인 이름(`https://vra-va-hostname.domain.name/vcac`)을 사용하여 vRealize Automation 콘솔을 엽니다.
고가용성 환경인 경우, 대상 가상 장치 로드 밸런서의 정규화된 도메인 이름(`https://vra-va-lb-hostname.domain.name/vcac`)을 사용하여 콘솔을 엽니다.
 - b IaaS 관리자 사용자로 로그인합니다.

프로시저

- 1 **관리 > vRO 구성 > 끝점**을 선택합니다.
- 2 Azure 끝점을 선택합니다.
- 3 **편집**을 클릭합니다.
- 4 **세부 정보**를 클릭합니다.
- 5 **클라이언트 비밀** 텍스트 상자에 원래 클라이언트 비밀 키를 입력합니다.
- 6 **완료**를 클릭합니다.

7 각 Azure 끝점에 대해 반복합니다.

vRealize Automation 6.2.x Automation Application Services를 7.3으로 마이그레이션

VMware vRealize Application Services 마이그레이션 도구를 사용하여 기존 Application Services Blueprint 및 배포 프로파일을 VMware vRealize Application Services 6.2.x에서 vRealize Automation 7.3으로 마이그레이션할 수 있습니다.

필수 조건

vRealize Automation 7.3으로 마이그레이션을 완료했습니다.

프로시저

- ◆ VMware vRealize Application Services 마이그레이션 도구를 다운로드하려면 다음 단계를 완료하십시오.
 - a [VMware vRealize Automation 다운로드](#)를 클릭합니다.
 - b **드라이버 및 도구 > VMware vRealize Application Services 마이그레이션 도구**를 선택합니다.

기존 가상 시스템의 소프트웨어 에이전트 업데이트

vRealize Automation 7.2에서 7.3으로 마이그레이션한 후에는 대상 vRealize Automation 콘솔에서 기존 가상 시스템의 소프트웨어 구성 요소를 관리할 수 없습니다. 대상 콘솔에서 기존 가상 시스템의 소프트웨어 구성 요소를 관리할 수 있으려면 먼저 각 가상 시스템의 소프트웨어 에이전트를 업데이트해야 합니다.

vRealize Orchestrator 클라이언트를 사용하여 다음과 같은 작업을 수행합니다.

- 다운로드한 소프트웨어 에이전트 마이그레이션 후 업데이트 패키지를 소스 vRealize Orchestrator로 가져옵니다.
- 기존 가상 시스템의 소프트웨어 에이전트를 업데이트합니다.
- 대상 vRealize Automation 장치와의 통신을 재설정합니다.

참고 소프트웨어 에이전트 업데이트는 취소할 수 없는 작업입니다. 이 업데이트를 수행한 후에는 소스 vRealize Automation 콘솔에서 기존 가상 시스템의 소프트웨어 구성 요소를 더 이상 관리할 수 없습니다.

필수 조건

- 소스 vRealize Automation 7.2 환경에서 대상 vRealize Automation 7.3 환경으로 성공적인 마이그레이션을 수행합니다.

- 소프트웨어 에이전트 마이그레이션 후 업데이트 패키지를 다운로드합니다.
 - a 대상 장치의 정규화된 도메인 이름(<https://vra-va-hostname.domain.name/software/index.html>)을 사용하여 대상 vRealize Automation 장치 [게스트 및 소프트웨어 에이전트 설치 관리자] 페이지를 엽니다.
 - b **소프트웨어 에이전트 업데이트 워크플로**를 클릭합니다.
- vRealize Orchestrator 클라이언트를 사용하여 대상 vRealize Orchestrator에 연결합니다. 자세한 내용은 [vRealize Orchestrator 설명서](#)에서 VMware vRealize Orchestrator Client 사용 항목을 참조하십시오.

프로시저

- 1 vRealize Orchestrator 클라이언트의 상단 드롭다운 메뉴에서 **실행**을 선택합니다.
- 2 [내 Orchestrator] 페이지에서 **패키지 가져오기**를 클릭합니다.
- 3 소프트웨어 에이전트 마이그레이션 후 업데이트 패키지를 다운로드한 디렉토리 (`com.vmware.vra.sct.update.package`)로 이동합니다.
- 4 패키지 이름을 선택하고 **열기**를 클릭합니다.
- 5 **제공자 가져오기 및 신뢰**를 클릭합니다.
- 6 **선택한 요소 가져오기**를 클릭합니다.
가져온 패키지가 표시되어 있는 **패키지** 탭이 열립니다.
- 7 **워크플로** 탭을 클릭합니다.
- 8 확장 버튼을 클릭하고 **라이브러리 > vRealize Automation > 마이그레이션 > 소프트웨어 에이전트**를 선택합니다.
- 9 대상 **vRealize Automation**에서 **소프트웨어 에이전트 상위 항목 재지정**을 두 번 클릭합니다.
소스 vRealize Automation 환경의 각 테넌트에 대해 이 워크플로를 실행합니다.
- 10 마법사를 실행하려면 오른쪽 창 맨 위에 있는 녹색 **워크플로 시작** 버튼을 클릭합니다.
- 11 소스 vRealize Automation 환경에 대해 요청된 정보를 입력합니다.
- 12 대상 vRealize Automation 환경에 대해 요청된 정보를 입력합니다.
이 대상 환경 정보는 대상 vRealize Automation 관리 콘솔의 [마이그레이션 상태] 페이지에 나와 있습니다.
 - 가상 장치 IP 주소
 - 가상 장치 인증서
 - 소프트웨어 에이전트 JAR SHA256 체크섬
- 13 **제출**을 클릭합니다.
워크플로가 다음의 작업을 소스 vRealize Automation 환경에서 수행합니다.
 - 테넌트에서 사용자를 인증하여 API 토큰을 가져옵니다.

- 소프트웨어 에이전트 업데이트 스크립트를 소스 vRealize Automation 환경에 새 소프트웨어 구성 요소로 설치합니다. 시스템에서는 지원되는 운영 체제(Windows 또는 Linux) 각각에 대해 하나의 소프트웨어 구성 요소를 설치합니다.
- 소프트웨어 에이전트가 설치되어 있고 실행 중인 가상 시스템의 목록을 가져옵니다.
- 목록 내의 각 가상 시스템에서 적절한 소프트웨어 에이전트 업데이트 스크립트를 실행하여 소프트웨어 에이전트를 업데이트합니다.
- 이전에 추가한 소프트웨어 구성 요소를 소스 vRealize Automation 환경에서 제거합니다.

원본 대상 vRealize Automation IaaS Microsoft SQL 데이터베이스 삭제

마이그레이션이 완료되면 원본 IaaS 데이터베이스를 삭제할 수 있습니다.

필수 조건

vRealize Automation 7.3으로 마이그레이션을 완료했습니다.

마이그레이션된 환경에서는 대상 vRealize Automation 7.3 환경을 설치할 때 생성한 원본 vRealize Automation IaaS Microsoft SQL 데이터베이스를 사용하지 않습니다. 따라서 마이그레이션을 완료한 후에는 이 원본 IaaS 데이터베이스를 Microsoft SQL Server에서 안심하고 삭제할 수 있습니다.

마이그레이션 후 데이터 센터 위치 메뉴 콘텐츠 업데이트

마이그레이션 후에는 누락된 모든 사용자 지정 데이터 센터 위치를 위치 드롭다운 메뉴에 추가해야 합니다.

vRealize Automation 7.3으로 마이그레이션 후에는 [계산 리소스] 페이지의 위치 드롭다운 메뉴에 있는 데이터 센터 위치가 기본 목록으로 되돌려집니다. 사용자 지정 데이터 센터 위치는 누락되지만 모든 계산 리소스 구성은 성공적으로 마이그레이션되고 Vrm.DataCenter.Location 속성이 영향을 받지 않습니다. 위치 메뉴에 사용자 지정 데이터 센터 위치를 추가할 수 있습니다.

필수 조건

vRealize Automation 7.3으로 마이그레이션합니다.

프로시저

- ◆ 누락된 데이터 센터 위치를 위치 드롭다운 메뉴에 추가합니다. vRealize Automation 구성에서 시나리오: 영역 간 배포를 위한 데이터 센터 위치 추가 항목을 참조하십시오.

대상 vRealize Automation 7.3 환경 검증

모든 데이터가 대상 vRealize Automation 7.3 환경에 성공적으로 마이그레이션되었는지 확인할 수 있습니다.

필수 조건

- vRealize Automation 7.3으로 마이그레이션합니다.
- 대상 vRealize Automation 콘솔에 로그인합니다.
 - a 대상 가상 장치의 정규화된 도메인 이름(https://vra-va-hostname.domain.name/vcac)을 사용하여 vRealize Automation 콘솔을 엽니다.
고가용성 환경인 경우, 대상 가상 장치 로드 밸런서의 정규화된 도메인 이름(https://vra-va-lb-hostname.domain.name/vcac)을 사용하여 콘솔을 엽니다.
 - b 테넌트 관리자 사용자 이름과 암호를 사용하여 로그인합니다.

프로시저

- 1 **인프라 > 관리되는 시스템**을 선택하고 모든 관리되는 가상 시스템이 존재하는지 확인합니다.
- 2 **계산 리소스**를 클릭하고 각 끝점을 선택한 후 **데이터 수집, 지금 요청 및 새로 고침**을 클릭하여 끝점이 작동 중인지 확인합니다.
- 3 **설계**를 클릭하고 **Blueprint** 페이지에서 각 Blueprint의 요소를 확인합니다.
- 4 **XaaS**를 클릭하고 **사용자 지정 리소스, 리소스 매핑, XaaS Blueprint, 리소스 작업**의 내용을 확인합니다.
- 5 **관리 > 카탈로그 관리**를 선택하고 **서비스, 카탈로그 항목, 작업 및 사용 권한**의 내용을 확인합니다.
- 6 **항목 > 배포**를 선택하고 프로비저닝된 가상 시스템의 정보를 확인합니다.
- 7 [배포] 페이지에서 전원이 꺼진 프로비저닝된 가상 시스템을 선택하고 **작업 > 전원 켜기**를 선택한 다음 **제출**을 클릭하고 **확인**을 클릭합니다. 가상 시스템 전원이 제대로 켜졌는지 확인합니다.
- 8 **카탈로그**를 클릭하고 새 카탈로그 항목을 요청합니다.
- 9 **일반** 탭에서 요청 정보를 입력합니다.
- 10 시스템 아이콘을 클릭하고 모든 기본 설정을 그대로 선택한 다음 **제출과 확인**을 차례로 클릭합니다.
- 11 요청이 완료되는지 확인합니다.

마이그레이션 문제 해결

마이그레이션 문제 해결 항목은 vRealize Automation을 마이그레이션할 때 발생할 수 있는 문제에 대한 솔루션을 제공합니다.

이 장에서는 다음 주제에 대해 설명합니다.

- PostgreSQL 버전에 따른 오류 발생
- 마이그레이션 시 일부 가상 시스템의 배포가 생성되지 않음
- 로드 밸런서 구성 때문에 장기 실행 작업에서 시간 초과 발생
- 마이그레이션 로그 위치
- 마이그레이션 후 카탈로그 항목이 서비스 카탈로그에 나타나지만 요청할 수 없음
- 마이그레이션 후 vRealize Automation에 빈 배포가 표시됨
- 마이그레이션 이후 배포라는 이름의 XaaS 리소스 매핑이 누락됨

PostgreSQL 버전에 따른 오류 발생

업데이트된 PostgreSQL 데이터베이스가 포함된 소스 vRealize Automation 6.2.x 환경에서 관리자 액세스를 차단합니다.

문제점

vRealize Automation 6.2.x에서 업그레이드된 PostgreSQL 데이터베이스를 사용하는 경우 관리자는 vRealize Automation에서 이 데이터베이스로의 액세스를 제공하는 항목을 pg_hba.conf 파일에 추가해야 합니다.

해결 방법

- 1 pg_hba.conf 파일을 엽니다.
- 2 이 데이터베이스에 대한 액세스 권한을 부여하려면 다음 항목을 추가합니다.

```
host all vcac-database-user vra-va-ip trust-method
```

마이그레이션 시 일부 가상 시스템의 배포가 생성되지 않음

마이그레이션할 때 누락된 상태의 가상 시스템에 대해서는 해당하는 배포가 대상 환경에 생성되지 않습니다.

문제점

마이그레이션 시 소스 환경에서 가상 시스템이 누락된 상태인 경우, 해당하는 배포가 대상 환경에 생성되지 않습니다.

해결 방법

- ◆ 마이그레이션 이후에 가상 시스템이 더 이상 누락된 상태가 아니면 대량 가져오기를 사용하여 가상 시스템을 대상 배포로 가져올 수 있습니다.

로드 밸런서 구성 때문에 장기 실행 작업에서 시간 초과 발생

로드 밸런서 때문에 연결이 예기치 않게 종료될 수 있습니다.

문제점

일부 로드 밸런서는 HTTP/HTTPS 요청을 실행하는 동안 연결을 지속하기 위한 시간 제한이 매우 짧습니다. 짧은 시간 제한으로 인해 마이그레이션 시 장기 실행 작업을 수행할 때 연결이 예기치 않게 종료될 수 있습니다.

해결 방법

- ◆ 로드 밸런서의 시간 제한을 늘리거나 마이그레이션을 수행하는 동안 적절한 활성 노드를 가리키도록 로드 밸런서 DNS 기록을 업데이트합니다. 마이그레이션이 완료된 후에는 로드 밸런서 DNS 기록을 되돌립니다.

마이그레이션 로그 위치

마이그레이션 프로세스를 기록하는 로그를 확인하여 검증 또는 마이그레이션 문제를 해결할 수 있습니다.

표 6-1. 소스 vRealize Automation 장치

로그	위치
패키지 생성 로그	/var/log/vmware/vcac/migration-package.log

표 6-2. 대상 vRealize Automation 장치

로그	위치
마이그레이션 로그	/var/log/vmware/vcac/migrate.log
마이그레이션 실행 로그	/var/log/vmware/vcac/mseq.migration.log
마이그레이션 실행 출력 로그	/var/log/vmware/vcac/mseq.migration.out.log
검증 실행 로그	/var/log/vmware/vcac/mseq.validation.log
검증 실행 출력 로그	/var/log/vmware/vcac/mseq.validation.out.log

표 6-3. 대상 vRealize Automation 인프라 노드

로그	위치
마이그레이션 로그	C:\Program Files (x86)\VMware\WvCACW\InstallLogs-YYYYMMDDHHMMXXWmigrate.log
검증 로그	C:\Program Files (x86)\VMware\WvCACW\InstallLogs-YYYYMMDDHHMMXXWValidate.log

마이그레이션 후 카탈로그 항목이 서비스 카탈로그에 나타나지만 요청할 수 없음

최신 버전의 vRealize Automation으로 마이그레이션한 후 이전 버전의 특정 속성 정의를 사용하는 카탈로그 항목이 서비스 카탈로그에 나타나지만 요청할 수 없습니다.

문제점

6.2.x 이전 버전에서 마이그레이션했고 다음과 같은 제어 유형 또는 특성이 포함된 속성 정의가 있는 경우, 이러한 요소가 속성 정의에서 누락되고 해당 정의를 사용하는 모든 카탈로그 항목이 마이그레이션하기 전의 방식대로 작동하지 않습니다.

- 제어 유형. 확인란 또는 링크.
- 특성. 관계, 정규식 또는 속성 레이아웃.

원인

vRealize Automation 7.0 이상에서는 속성 정의에서 더 이상 이러한 요소를 사용하지 않습니다. 속성 정의에서 포함된 제어 유형 또는 특성을 사용하지 않고 대신 vRealize Orchestrator 스크립트 작업을 사용하도록 속성 정의를 구성하거나 속성 정의를 다시 생성해야 합니다.

스크립트 작업을 사용하여 제어 유형 또는 특성을 vRealize Automation 7.x로 마이그레이션합니다.

해결 방법

- 1 vRealize Orchestrator에서 속성 값을 반환하는 스크립트 작업을 생성합니다. 작업은 단순 유형을 반환해야 합니다. 예를 들어 문자열, 정수 또는 지원되는 다른 유형을 반환합니다. 작업은 해당 작업이 종속된 다른 속성을 입력 매개 변수로 사용할 수 있습니다.
- 2 vRealize Automation 콘솔에서 제품 정의를 구성합니다.
 - a **관리 > 속성 사전 > 속성 정의**를 선택합니다.
 - b 속성 정의를 선택하고 **편집**을 클릭합니다.
 - c [권장 사항 표시] 드롭다운 메뉴에서 **드롭다운**을 선택합니다.
 - d [값] 드롭다운 메뉴에서 **외부 값**을 선택합니다.
 - e 스크립트 작업을 선택합니다.
 - f **확인**을 클릭합니다.

- g 스크립트 작업에 포함된 입력 매개 변수를 구성합니다. 기존 관계를 유지하려면 매개 변수를 다른 속성에 바인딩합니다.
- h **확인**을 클릭합니다.

마이그레이션 후 vRealize Automation 에 빈 배포가 표시됨

프로비저닝 후 작업이 성공적으로 완료되는 것처럼 보이지만 실제로 아무것도 변경되지 않습니다.

문제점

마이그레이션 프로세스 때문에 일부 가상 시스템이 잘못된 배포에 할당됩니다. 자세한 내용은 기술 자료 문서 [2151400](#)을 참조하십시오.

마이그레이션 이후 배포라는 이름의 XaaS 리소스 매핑이 누락 됨

vRealize Automation 6.2.x에서 7.3으로 마이그레이션한 이후 배포라는 이름의 배포 XaaS 리소스가 누락됩니다.

문제점

vRealize Automation 6.2.x에서 7.3으로 마이그레이션한 이후 배포라는 이름의 XaaS 리소스가 누락됩니다. 자세한 내용은 [기술 자료 문서 1153](#)을 참조하십시오.