



# Red Hat Ansible Automation Platform 2.4

## Red Hat Ansible Automation Platform 계획 가이드

Ansible Automation Platform 설치 계획



# Red Hat Ansible Automation Platform 2.4 Red Hat Ansible Automation Platform 계획 가이드

---

Ansible Automation Platform 설치 계획

## 법적 공지

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 초록

이 가이드에서는 Red Hat Ansible Automation Platform 설치를 위한 요구 사항, 옵션 및 권장 사항을 제공합니다.

## 차례

머리말 .....	3
RED HAT 문서에 관한 피드백 제공 .....	4
1장. RED HAT ANSIBLE AUTOMATION PLATFORM 설치 계획 .....	5
2장. RED HAT ANSIBLE AUTOMATION PLATFORM 아키텍처 .....	6
2.1. ANSIBLE AUTOMATION PLATFORM 아키텍처의 예 .....	6
3장. RED HAT ANSIBLE AUTOMATION PLATFORM 구성 요소 .....	8
3.1. ANSIBLE 자동화 허브 .....	8
3.2. 프라이빗 자동화 허브 .....	8
3.3. 고가용성 자동화 허브 .....	8
3.4. 이벤트 기반 ANSIBLE 컨트롤러 .....	8
3.5. 자동화 메시 .....	8
3.6. 자동화 실행 환경 .....	9
3.7. ANSIBLE GALAXY .....	9
3.8. 자동화 콘텐츠 탐색기 .....	9
4장. 시스템 요구 사항 .....	10
4.1. RED HAT ANSIBLE AUTOMATION PLATFORM 시스템 요구 사항 .....	10
4.2. 자동화 컨트롤러 시스템 요구 사항 .....	11
4.3. 자동화 허브 시스템 요구 사항 .....	13
4.4. 이벤트 기반 ANSIBLE 컨트롤러 시스템 요구 사항 .....	15
4.5. POSTGRESQL 요구 사항 .....	16
5장. 네트워크 포트 및 프로토콜 .....	22
6장. RED HAT ANSIBLE AUTOMATION PLATFORM 서브스크립션 연결 .....	29
7장. RED HAT ANSIBLE AUTOMATION PLATFORM 설치 프로그램 선택 및 가져오기 .....	31
7.1. 인터넷으로 설치 .....	31
7.2. 인터넷 액세스없이 설치 .....	31
8장. 설치 프로그램 인벤토리 파일 정보 .....	33
8.1. 호스트 및 그룹 관련 지침 .....	34
8.2. 노드 또는 그룹 프로비저닝 해제 .....	35
8.3. 인벤토리 변수 .....	36
8.4. 인벤토리 파일에서 변수를 선언하는 규칙 .....	36
8.5. 인벤토리 파일의 보안 보안 .....	37
8.6. 추가 인벤토리 파일 변수 .....	37
9장. 지원되는 설치 시나리오 .....	39
9.1. 동일한 노드 또는 비INSTALLER 관리 데이터베이스의 데이터베이스가 있는 독립 실행형 자동화 컨트롤러 .....	39
9.2. 외부 관리 데이터베이스가 있는 독립형 자동화 컨트롤러 .....	39
9.3. 내부 데이터베이스가 있는 단일 이벤트 기반 ANSIBLE 컨트롤러 노드 .....	39
9.4. 동일한 노드 또는 비INSTALLER 관리 데이터베이스의 데이터베이스가 있는 독립 실행형 자동화 허브 .....	39
9.5. 외부 관리 데이터베이스가 있는 독립형 자동화 허브 .....	39
9.6. 자동화 컨트롤러 노드 또는 비INSTALLER 관리 데이터베이스에 데이터베이스를 사용한 플랫폼 설치 .....	40
9.7. 외부 관리 데이터베이스를 사용한 플랫폼 설치 .....	40
9.8. 외부 관리 데이터베이스를 사용한 다중 시스템 클러스터 설치 .....	40



## 머리말

Red Hat Ansible Automation Platform에 관심을 가져 주셔서 감사합니다. Ansible Automation Platform은 Ansible 기반 환경에 제어, 지식, 위임을 추가하여 팀이 복잡한 다중 계층 배포를 관리하는 데 도움이 되는 상용 서비스입니다.

이 가이드의 정보를 사용하여 Red Hat Ansible Automation Platform 설치를 계획합니다.

## RED HAT 문서에 관한 피드백 제공

이 문서를 개선하기 위한 제안이 있거나 오류를 찾을 수 있는 경우 <https://access.redhat.com> 에서 기술 지원에 문의하여 **docs-product** 구성 요소를 사용하여 Ansible Automation Platform Jira 프로젝트에 문제를 생성하십시오.



## 1장. RED HAT ANSIBLE AUTOMATION PLATFORM 설치 계획

Red Hat Ansible Automation Platform은 Red Hat Enterprise Linux 및 Red Hat OpenShift 모두에서 지원됩니다. 이 가이드를 사용하여 Red Hat Enterprise Linux에서 Red Hat Ansible Automation Platform 설치를 계획하십시오.

Red Hat OpenShift Container Platform 환경에 Red Hat Ansible Automation Platform을 설치하려면 OpenShift Container Platform [에 Red Hat Ansible Automation Platform Operator](#) 배포를 참조하십시오.

## 2장. RED HAT ANSIBLE AUTOMATION PLATFORM 아키텍처

모듈식 플랫폼으로 Ansible Automation Platform은 구성 요소를 쉽게 통합하고 자동화 요구 사항에 가장 적합한 배포를 사용자 정의할 수 있는 유연성을 제공합니다. 다음 섹션에서는 Ansible Automation Platform 배포의 포괄적인 아키텍처 예를 제공합니다.

### 2.1. ANSIBLE AUTOMATION PLATFORM 아키텍처의 예

Red Hat Ansible Automation Platform 2.4 참조 아키텍처에서는 Red Hat Enterprise Linux에서 자동화 메시지를 사용하여 Ansible Automation Platform의 표준 배포 설정을 보여줍니다. 표시된 배포에서는 자동화 워크로드를 처리하는 간단하고 안전하며 유연한 방법, 콘텐츠 컬렉션의 중앙 위치, IT 요청의 자동화된 해결을 제공하기 위해 다음 구성 요소를 활용합니다.

#### 자동화 컨트롤러

UI, Restful API, RBAC 워크플로 및 CI/CD 통합을 통해 자동화할 컨트롤 플레인을 제공합니다.

#### 자동화 메시

는 기존 네트워크를 사용하여 서로 피어 투 피어 연결을 설정하는 노드를 통해 대규모 작업자 컬렉션에서 작업을 쉽게 배포할 수 있는 기능을 제공하는 오버레이 네트워크입니다.

#### 프라이빗 자동화 허브

자동화 개발자에게 자체 자동화 콘텐츠를 협업 및 게시하고 조직 내에서 Ansible 코드 제공을 간소화할 수 있는 기능을 제공합니다.

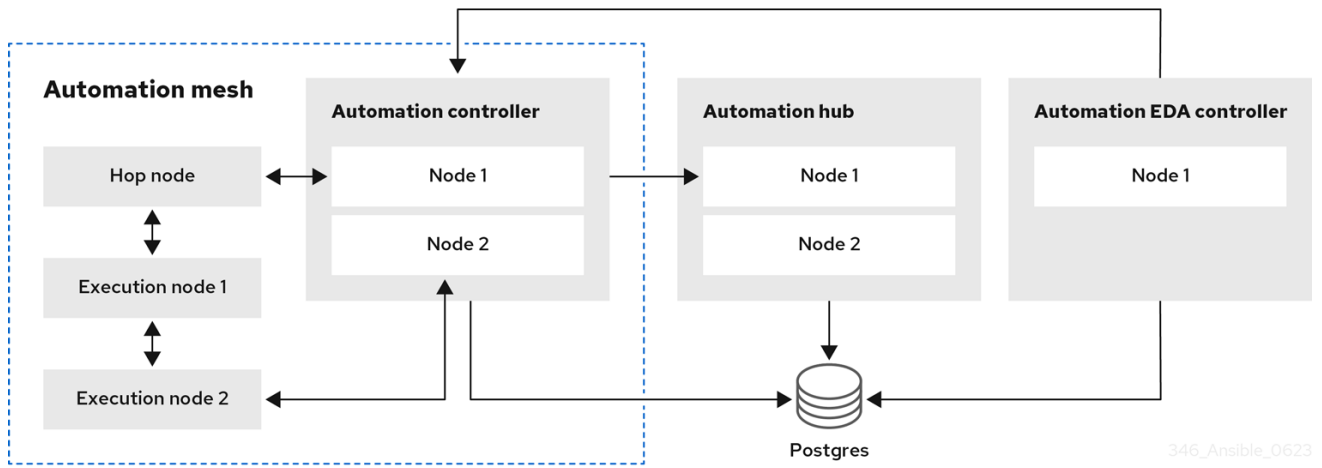
#### 이벤트 기반 Ansible

시간이 많이 걸리는 작업을 자동화하고 IT 도메인에서 변화하는 조건에 대응하는 데 필요한 이벤트 처리 기능을 제공합니다.

이 예제의 아키텍처는 다음으로 구성됩니다.

- 두 개의 노드 자동화 컨트롤러 클러스터
- 자동화 컨트롤러를 실행 노드에 연결하는 선택적 홉 노드
- 두 개의 노드 자동화 허브 클러스터
- 단일 노드 이벤트 기반 Ansible 컨트롤러 클러스터
- 자동화 컨트롤러, 자동화 허브 및 이벤트 기반 Ansible 컨트롤러 클러스터에 연결된 단일 PostgreSQL 데이터베이스
- 자동화 컨트롤러 클러스터당 두 개의 실행 노드

그림 2.1. Ansible Automation Platform 2.4 아키텍처의 예



## 3장. RED HAT ANSIBLE AUTOMATION PLATFORM 구성 요소

Ansible Automation Platform은 배포 요구 사항을 충족하기 위해 함께 연결할 수 있는 별도의 구성 요소로 구성된 모듈식 플랫폼입니다. Ansible Automation Platform 배포는 UI(사용자 인터페이스) 및 RESTful API(애플리케이션 프로그래밍 인터페이스)를 사용하여 Ansible 자동화를 제어, 보안 및 관리하기 위한 엔터프라이즈 프레임워크인 자동화 컨트롤러로 시작합니다. 그런 다음 다음 자동화 플랫폼 구성 요소의 모든 조합을 배포에 추가할 수 있습니다.

### 3.1. ANSIBLE 자동화 허브

Ansible 자동화 허브는 Ansible 콘텐츠 컬렉션의 인증된 콘텐츠를 위한 리포지토리입니다. Red Hat과 파트너는 중앙 집중식으로 콘텐츠를 게시하며 고객이 인증된 지원되는 Ansible 콘텐츠 컬렉션을 검색할 수 있습니다. Red Hat Ansible 인증 콘텐츠는 Red Hat에서 테스트 및 지원하는 콘텐츠를 사용자에게 제공합니다.

### 3.2. 프라이빗 자동화 허브

프라이빗 자동화 허브는 콘텐츠 동기화를 위한 연결이 끊긴 및 온프레미스 솔루션을 모두 제공합니다. Red Hat 클라우드 자동화 허브의 컬렉션 및 실행 환경 이미지를 동기화하여 사용자 정의 자동화 컬렉션 및 실행 이미지를 저장하고 제공할 수 있습니다. Ansible Galaxy 또는 기타 컨테이너 레지스트리와 같은 다른 소스를 사용하여 개인 자동화 허브에 콘텐츠를 제공할 수도 있습니다. 프라이빗 자동화 허브는 엔터프라이즈 디렉터리 및 CI/CD 파이프라인에 통합할 수 있습니다.

### 3.3. 고가용성 자동화 허브

HA(고가용성) 구성은 자동화 허브 배포를 위한 안정성과 스칼라성을 향상시킵니다.

자동화 허브의 HA 배포에는 워크로드를 배포하는 로드 밸런서를 사용하여 동일한 서비스를 동시에 실행하는 여러 노드가 있습니다(예: "활성" 구성). 이 구성을 사용하면 단일 장애 지점을 제거하여 서비스 다운타임을 최소화할 수 있으며 워크로드 요구 사항을 충족하기 위해 노드를 쉽게 추가하거나 제거할 수 있습니다.

### 3.4. 이벤트 기반 ANSIBLE 컨트롤러

이벤트 기반 Ansible 컨트롤러는 이벤트 중심 자동화를 위한 인터페이스이며 IT 요청을 자동으로 해결합니다. 이벤트 기반 Ansible 컨트롤러를 사용하면 룰북을 사용하여 이벤트 소스에 연결하고 해당 이벤트에서 작업할 수 있습니다. 이 기술은 IT 속도와 민첩성을 개선하고 일관성과 복원력을 가능하게 합니다. 이벤트 기반 Ansible을 사용하면 다음을 수행할 수 있습니다.

- 의사 결정 자동화
- 많은 이벤트 소스 사용
- 내부 및 많은 IT 사용 사례에서 이벤트 중심 자동화 구현

#### 추가 리소스

- [이벤트 기반 Ansible 가이드 시작하기](#).

### 3.5. 자동화 메시

자동화 메시는 기존 네트워크를 사용하여 서로 피어 간 연결을 설정하는 노드를 통해 대규모의 분산된 작업자 컬렉션에서 작업 분배를 용이하게 하기 위한 오버레이 네트워크입니다.

자동화 메시는 다음을 제공합니다.

- 독립적으로 확장되므로 다운타임을 최소화하여 노드를 생성, 등록, 그룹 해제 및 등록할 수 있습니다.
- 컨트롤 플레인 용량과 별도로 플레이북 실행 용량을 확장할 수 있는 컨트롤 및 실행 플레인 분리입니다.
- 대기 시간에 탄력적이고, 중단 없이 다시 구성할 수 있으며, 중단이 발생할 때 다른 경로를 동적으로 다시 라우팅하는 배포 선택
- 메시 라우팅 변경
- FIPS(Federal Information Processing Standards) 준수를 위한 양방향, 다중 연결 메시 통신 가능성이 포함됩니다.

### 3.6. 자동화 실행 환경

자동화 실행 환경은 Red Hat Ansible Automation Platform의 모든 자동화가 실행되는 컨테이너 이미지입니다. Ansible 실행 엔진과 수백 개의 모듈이 포함된 솔루션을 제공하여 사용자가 IT 환경 및 프로세스의 모든 측면을 자동화할 수 있도록 지원합니다. 자동화 실행 환경은 일반적으로 사용되는 운영 체제, 인프라 플랫폼, 네트워크 장치 및 클라우드를 자동화합니다.

### 3.7. ANSIBLE GALAXY

Ansible Galaxy는 Ansible 콘텐츠를 검색, 재사용 및 공유하기 위한 허브입니다. 사전 패키징된 역할 형태로 커뮤니티 제공 Galaxy 콘텐츠는 자동화 프로젝트를 시작하는 데 도움이 될 수 있습니다. 인프라 프로비저닝, 애플리케이션 배포 및 기타 작업을 완료하는 데 필요한 역할을 Ansible 플레이북에 드롭할 수 있으며 고객 환경에 즉시 적용할 수 있습니다.

### 3.8. 자동화 콘텐츠 탐색기

Automation Content navigator는 텍스트 사용자 인터페이스(TUI)로, 콘텐츠 빌드의 사용 사례를 소개하고, 실행 환경에서 로컬로 자동화 실행, Ansible Automation Platform에서 자동화 실행, 향후 IDE(통합 개발 환경)의 기반을 제공하는 텍스트 사용자 인터페이스(TUI)입니다.

## 4장. 시스템 요구 사항

Red Hat Ansible Automation Platform 설치를 계획하고 사용 사례에 맞는 자동화 메시 토폴로지를 설계할 때 이 정보를 사용하십시오.

### 사전 요구 사항

- **sudo** 명령을 통해 또는 권한 에스컬레이션을 통해 root 액세스 권한을 얻을 수 있습니다. 권한 에스컬레이션에 대한 자세한 내용은 권한 [에스컬레이션 이해](#) 를 참조하십시오.
- AWX, PostgreSQL, Event-Driven Ansible 또는 Pulp와 같은 권한을 root에서 사용자로 분리할 수 있습니다.
- 모든 노드에 NTP 클라이언트를 구성했습니다. 자세한 내용은 [Chrony를 사용하여 NTP 서버 구성](#) 을 참조하십시오.

### 4.1. RED HAT ANSIBLE AUTOMATION PLATFORM 시스템 요구 사항

Red Hat Ansible Automation Platform을 설치하고 실행하려면 시스템에서 다음과 같은 최소 시스템 요구 사항을 충족해야 합니다.

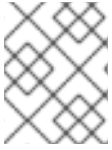
표 4.1. 기본 시스템

요구 사항	필수 항목	참고
서브스크립션	유효한 Red Hat Ansible Automation Platform	
OS	Red Hat Enterprise Linux 8.6 이상 64비트(x86, ppc64le, s390x, aarch64)	Red Hat Ansible Automation Platform은 OpenShift에서 지원됩니다. 자세한 내용은 <a href="#">OpenShift Container Platform에 Red Hat Ansible Automation Platform Operator</a> 배포를 참조하십시오.
ansible-core	ansible-core 버전 2.14 이상	Ansible Automation Platform에는 ansible-core 2.15가 포함된 실행 환경이 포함되어 있습니다.
Python	3.9 이상	
브라우저	현재 지원되는 Mozilla FireFox 버전 또는 Google Chrome	
데이터베이스	PostgreSQL 버전 13	

다음은 프로젝트 업데이트 및 컬렉션을 사용하려면 다음과 같습니다.

- *Table 5.9*에 나열된 [네트워크 포트 및 프로토콜](#)이 있는지 확인합니다. *Automation Hub* 를 사용하여 자동화 허브 또는 Ansible Galaxy 서버에서 컬렉션을 성공적으로 연결하고 다운로드할 수 있습니다.

- 자체 서명된 인증서를 사용하는 경우 또는 Red Hat 도메인을 사용하는 경우 SSL 검사를 비활성화합니다.

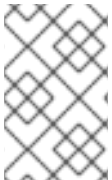


### 참고

Ansible Automation Platform에서 관리하는 시스템에 대한 요구 사항은 Ansible과 동일합니다. Ansible 커뮤니티 [설명서](#)에서 [Ansible 설치](#) 단원을 참조하십시오.

### Red Hat Ansible Automation Platform 요구 사항에 대한 추가 노트

- Red Hat Ansible Automation Platform은 Ansible 플레이북에 따라 다르며 ansible-core의 안정적인 최신 버전을 설치해야 합니다. ansible-core를 수동으로 다운로드하거나 Red Hat Ansible Automation Platform 설치의 일부로 자동으로 다운로드할 수 있습니다.
- 새로운 설치의 경우 자동화 컨트롤러는 ansible-core의 최신 릴리스 패키지를 설치합니다.
- 번들로 제공되는 Ansible Automation Platform 설치를 수행하는 경우 설치 setup.sh 스크립트에서 번들에서 ansible-core(및 해당 종속 항목)를 설치하려고 합니다.
- Ansible을 수동으로 설치한 경우 Ansible Automation Platform 설치 setup.sh 스크립트에서 Ansible이 설치되었음을 감지하고 다시 설치하지 않습니다.



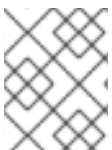
### 참고

Red Hat Ansible Automation Platform이 제대로 작동하려면 **dnf** 와 같은 패키지 관리자를 사용하여 Ansible을 설치하고 안정적인 최신 버전의 패키지 관리자를 설치해야 합니다. 버전 2.4 이상에는 Ansible 버전 2.14가 필요합니다.

## 4.2. 자동화 컨트롤러 시스템 요구 사항

자동화 컨트롤러는 여러 컴퓨팅 노드에 서로 다른 소프트웨어 구성 요소를 함께 배치하거나 배포할 수 있는 분산 시스템입니다. 설치 프로그램에서 사용 사례에 적합한 토폴로지를 컨트롤, 하이브리드, 실행 및 흡 노드에 적합한 토폴로지를 설계할 수 있도록 네 가지 노드 유형이 추상화로 제공됩니다.

노드 크기 조정에는 다음 권장 사항을 사용합니다.



### 참고

제어 및 하이브리드 노드에서 실행 환경 스토리지를 위해 최소 20GB를 `/var/lib/awx` 에 할당합니다.

### 실행 노드

실행 노드는 자동화를 실행합니다. 더 많은 포크를 실행하는 데 필요한 용량을 늘리려면 메모리 및 CPU를 늘립니다.



**참고**

- 명시된 RAM 및 CPU 리소스는 실행 노드에 설치된 패키지에 필요하지 않을 수 있지만 평균 작업 수를 동시에 실행하려면 노드의 작업 부하를 처리하는 데 권장되는 최소 수준입니다.
- 권장되는 RAM 및 CPU 노드 크기는 제공되지 않습니다. 필요한 RAM 또는 CPU는 해당 환경에서 실행 중인 작업 수에 따라 직접 달라집니다.

필요한 RAM 및 CPU 수준에 대한 자세한 내용은 [자동화 컨트롤러의 성능 튜닝](#)을 참조하십시오.

**표 4.2. 실행 노드**

요구 사항	최소 요구 사항
RAM	16GB
CPU	4
로컬 디스크	최소 40GB

**컨트롤 노드**

컨트롤 노드는 이벤트를 처리하고 프로젝트 업데이트 및 정리 작업을 포함하여 클러스터 작업을 실행합니다. CPU 및 메모리를 늘리면 작업 이벤트 처리에 도움이 될 수 있습니다.

**표 4.3. 컨트롤 노드**

요구 사항	최소 요구 사항
RAM	16GB
CPU	4
로컬 디스크	<ul style="list-style-type: none"> <li>• /var/lib/awx에서 최소 20GB를 사용할 수 있는 최소 40GB</li> <li>• 스토리지 볼륨은 최소 1500 IOPS 기준으로 평가되어야 합니다.</li> <li>• 프로젝트는 제어 및 하이브리드 노드에 저장되며 작업 기간 동안 실행 노드에도 저장됩니다. 클러스터에 많은 대규모 프로젝트가 있는 경우 디스크 공간 오류를 방지하기 위해 /var/lib/awx/projects에서 GB를 두 배로 늘리는 것이 좋습니다.</li> </ul>

**힙 노드**



홉 노드는 자동화 메시지의 한 부분에서 다른 부분으로 트래픽을 라우팅하는 역할을 합니다(예: 홉 노드는 다른 네트워크로 홉 호스트가 될 수 있음). RAM은 처리량에 영향을 미칠 수 있으며 CPU 활동이 적습니다. 네트워크 대역폭 및 대기 시간은 일반적으로 RAM 또는 CPU보다 더 중요한 요소입니다.

표 4.4. 홉 노드

요구 사항	최소 요구 사항
RAM	16GB
CPU	4
로컬 디스크	40GB

- 실제 RAM 요구 사항은 동시에 관리할 호스트 자동화 컨트롤러 수(작업 템플릿 또는 시스템 **ansible.cfg** 파일의 **forks** 매개변수에 의해 제어됨)에 따라 달라집니다. 가능한 리소스 충돌을 방지하기 위해 Ansible은 10 포크당 1GB의 메모리와 자동화 컨트롤러에 대해 2GB의 메모리를 권장합니다. 자세한 내용은 [자동화 컨트롤러 용량 결정 및 작업 영향](#)을 참조하십시오. 포크가 400으로 설정되면 42GB의 메모리가 권장됩니다.
- 자동화 컨트롤러 호스트는 **Cryostat**가 0022로 설정되어 있는지 확인합니다. 그렇지 않으면 설정이 실패합니다. 이 오류를 방지하려면 **Cryostat=0022**를 설정합니다.
- 더 많은 수의 호스트를 처리할 수 있지만 포크 번호가 총 호스트 수보다 작으면 호스트 간에 더 많은 패스가 필요합니다. 다음 접근 방법 중 하나를 사용하여 이러한 RAM 제한을 방지할 수 있습니다.
  - 롤링 업데이트를 사용합니다.
  - 자동화 컨트롤러에 빌드된 프로비저닝 콜백 시스템을 사용합니다. 각 시스템은 구성을 요청하는 각 시스템이 대기열로 전환하고 가능한 한 빨리 처리됩니다.
  - 자동화 컨트롤러에서 AMI와 같은 이미지를 생성하거나 배포하는 경우입니다.

#### 추가 리소스

- 자동화 컨트롤러 서브스크립션을 얻는 방법에 대한 자세한 내용은 서브스크립션 [가져오기](#)를 참조하십시오.
- 질문이 있는 경우 [Red Hat 고객 포털](#)을 통해 Ansible 지원에 문의하십시오.

### 4.3. 자동화 허브 시스템 요구 사항

자동화 허브를 사용하면 Red Hat Ansible 및 Certified Partners의 새로운 인증 자동화 콘텐츠를 검색하고 사용할 수 있습니다. Ansible 자동화 허브에서는 클라우드 자동화, 네트워크 자동화 및 보안 자동화와 같은 사용 사례를 위해 Red Hat과 파트너가 개발한 자동화 콘텐츠인 Ansible 컬렉션을 검색하고 관리할 수 있습니다.

Automation Hub에는 다음과 같은 시스템 요구 사항이 있습니다.

요구 사항	필수 항목	참고
RAM	최소 8GB	<ul style="list-style-type: none"> <li>8GB RAM (최소 및 gRPC 평가판 설치 시 권장)</li> <li>8GB RAM(외부 독립 실행형 PostgreSQL 데이터 베이스의 경우 최소 용량)</li> <li>구성의 포크를 기반으로 하는 용량은 <a href="#">자동화 컨트롤러 용량 결정 및 작업 영향</a>을 참조하십시오.</li> </ul>
CPU	최소 2개	구성의 포크를 기반으로 하는 용량은 <a href="#">자동화 컨트롤러 용량 결정 및 작업 영향</a> 을 참조하십시오.
로컬 디스크	60GB 디스크	컬렉션 저장을 위해 최소 40GB를 <b>/var</b> 에 전용으로 지정합니다.



### 참고

#### 프라이빗 자동화 허브

내부 주소에서 개인 자동화 허브를 설치하고 외부 주소만 포함하는 인증서가 있는 경우 인증서 문제가 없는 컨테이너 레지스트리로 사용할 수 없습니다.

이를 방지하려면 설치 인벤토리 파일의 프라이빗 자동화 허브 노드에 연결하는 <https://pah.example.com>와 같은 값과 함께 **automationhub\_main\_url** 인벤토리 변수를 사용합니다.

이렇게 하면 외부 주소가 **/etc/pulp/settings.py**에 추가됩니다. 즉, 외부 주소만 사용해야 합니다.

인벤토리 파일 변수에 대한 자세한 내용은 *Red Hat Ansible Automation Platform 설치 가이드*의 [인벤토리 파일 변수를 참조하십시오](#).

### 4.3.1. 고가용성 자동화 허브 요구사항

HA(고가용성) 자동화 허브를 배포하기 전에 환경에 공유 파일 시스템이 설치되어 있고 필요한 경우 네트워크 스토리지 시스템을 구성했는지 확인합니다.

#### 4.3.1.1. 필수 공유 파일 시스템

고가용성 자동화 허브를 사용하려면 NFS와 같은 공유 파일 시스템이 환경에 이미 설치되어 있어야 합니다. Red Hat Ansible Automation Platform 설치 프로그램을 실행하기 전에 클러스터 전체에 **/var/lib/pulp** 디렉터리를 공유 파일 시스템 설치의 일부로 설치했는지 확인합니다. Red Hat Ansible Automation Platform 설치 프로그램은 노드 중 하나에서 **/var/lib/pulp**이 탐지되지 않는 경우 오류를 반환하여 고가용성 자동화 허브 설정이 실패합니다.

노드 중 하나에서 **/var/lib/pulp**가 탐지되지 않는 오류가 발생하면 모든 서버에 **/var/lib/pulp**가 제대로 마운트되고 설치 프로그램을 다시 실행합니다.

### 4.3.1.2. 네트워크 스토리지를 위한 firewalld 설치

자동화 허브 노드 자체에 네트워크 스토리지를 사용하여 HA 자동화 허브를 설치하려면 먼저 **firewalld** 를 설치하고 사용하여 Ansible Automation Platform 설치 프로그램을 실행하기 전에 공유 스토리지 시스템에 필요한 포트를 열어야 합니다.

다음 명령을 실행하여 **firewalld** 를 설치하고 구성합니다.

1. **firewalld** 데몬을 설치합니다.

```
$ dnf install firewalld
```

2. 다음 명령을 사용하여 <service> 아래에 네트워크 스토리지를 추가합니다.

```
$ firewall-cmd --permanent --add-service=<service>
```



#### 참고

지원되는 서비스 목록은 **\$ firewall-cmd --get-services** 명령을 사용하십시오.

3. 구성을 적용하려면 다시 로드합니다.

```
$ firewall-cmd --reload
```

## 4.4. 이벤트 기반 ANSIBLE 컨트롤러 시스템 요구 사항

이벤트 기반 Ansible 컨트롤러는 CPU 코어 수에 따라 필요에 따라 다양한 장기 실행 프로세스(예: 룰북 활성화)를 처리할 수 있는 단일 노드 시스템입니다. 다음 최소 요구 사항을 사용하여 기본적으로 최대 12개의 동시 활성화를 실행합니다.

요구 사항	필수 항목
RAM	16GB
CPU	4
로컬 디스크	최소 40GB



중요

- Red Hat Enterprise Linux 8을 실행 중이고 메모리 제한을 설정하려면 이벤트 기반 Ansible을 설치하기 전에 cgroup v2가 활성화되어 있어야 합니다. 자세한 내용은 Knowledge-Centered Support (KCS) 문서를 참조하십시오. [Ansible Automation Platform Event-Driven Ansible Controller for Red Hat Enterprise Linux 8 requires cgroupv2.](#)
- 표준 조건에서 이벤트 기반 Ansible 롤북을 활성화하면 약 250MB의 메모리를 사용합니다. 그러나 실제 메모리 사용은 규칙의 복잡성과 처리된 이벤트의 볼륨 및 크기에 따라 크게 다를 수 있습니다. 많은 수의 이벤트가 예상되거나 롤북 복잡성이 높은 시나리오에서는 스테이징 환경에서 리소스 사용을 사전 평가하십시오. 이렇게 하면 최대 활성화 수가 리소스 용량을 기반으로 합니다. [이벤트 기반 Ansible 컨트롤러 최대 실행 가능한 활성화 설정에 대한 예는 단일 자동화 컨트롤러, 단일 자동화 허브 및 외부\(installer managed\) 데이터베이스가 포함된 단일 자동화 컨트롤러, 단일 이벤트 기반 Ansible 컨트롤러 노드를 참조하십시오.](#)

### 4.5. POSTGRESQL 요구 사항

Red Hat Ansible Automation Platform에서는 PostgreSQL 13을 사용합니다. PostgreSQL 사용자 암호는 데이터베이스에 저장하기 전에 SCRAM-SHA-256 보안 해시 알고리즘으로 해시됩니다.

자동화 컨트롤러 인스턴스가 데이터베이스에 액세스할 수 있는지 확인하려면 명령 **awx-manage check\_db** 명령을 사용하여 수행할 수 있습니다.

표 4.5. 데이터베이스

Service	필수 항목	참고
---------	-------	----

Service	필수 항목	참고
데이터베이스	<ul style="list-style-type: none"> <li>● 20GB 전용 하드 디스크 공간</li> <li>● 4개의 CPU</li> <li>● 16GB RAM</li> </ul>	<ul style="list-style-type: none"> <li>● 150GB 이상 권장</li> <li>● 스토리지 볼륨은 높은 기준 IOPS(1500 이상)로 평가되어야 합니다.</li> <li>● 모든 자동화 컨트롤러 데이터는 데이터베이스에 저장됩니다. 데이터베이스 스토리지는 관리되는 호스트 수, 실행 횟수, 사실 캐시에 저장된 팩트 수 및 개별 작업의 작업 수를 늘립니다. 예를 들어 플레이북이 250개 호스트에서 매시간(하루 24회) 실행되고 20개의 작업이 포함된 플레이북은 매주 데이터베이스에 800000개 이상의 이벤트를 저장합니다.</li> <li>● 데이터베이스에 공간이 충분하지 않은 경우 이전 작업 실행 및 팩트를 정기적으로 정리해야 합니다. 자세한 내용은 <a href="#">자동화 컨트롤러 관리 가이드</a>의 관리 작업을 참조하십시오.</li> </ul>

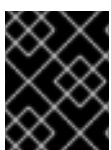
## PostgreSQL 구성

선택적으로 PostgreSQL 데이터베이스를 Red Hat Ansible Automation Platform 설치 프로그램에서 관리하지 않는 별도의 노드로 구성할 수 있습니다. Ansible Automation Platform 설치 프로그램에서 데이터베이스 서버를 관리할 때 대부분의 워크로드에 일반적으로 권장되는 기본값을 사용하여 서버를 구성합니다. 데이터베이스 성능을 개선하는 데 사용할 수 있는 설정에 대한 자세한 내용은 [데이터베이스 설정](#)을 참조하십시오.

### 추가 리소스

PostgreSQL 서버 튜닝에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

#### 4.5.1. 외부(고객 지원) 데이터베이스 설정



#### 중요

Red Hat은 외부(고객 지원) 데이터베이스를 지원하지 않지만 고객이 사용합니다. 제품 설치 관점에서의 초기 구성에 대한 다음 지침은 관련 지원 요청을 방지하기 위해 제공됩니다.

자동화 컨트롤러와 함께 사용할 외부 PostgreSQL 호환 데이터베이스에서 데이터베이스, 사용자 및 암호를 생성하려면 다음 절차를 사용하십시오.

#### 절차

1. 슈퍼유저 권한으로 PostgreSQL 호환 데이터베이스 서버에 연결합니다.

```
# psql -h <db.example.com> -U superuser -p 5432 -d postgres <Password for user
superuser>:
```

다음과 같습니다.

```
-h hostname
--host=hostname
```

서버가 실행 중인 시스템의 호스트 이름을 지정합니다. 값이 슬래시로 시작되면 Unix-domain 소켓의 디렉터리로 사용됩니다.

```
-d dbname
--dbname=dbname
```

연결할 데이터베이스의 이름을 지정합니다. 이는 명령줄에서 첫 번째 비옵션 인수로 **dbname** 을 지정하는 것과 동일합니다. **dbname** 은 연결 문자열일 수 있습니다. 이 경우 연결 문자열 매개 변수는 충돌하는 명령줄 옵션을 재정의합니다.

```
-U username
--username=username
```

기본값 대신 사용자 이름으로 데이터베이스에 연결합니다. (이 작업을 수행할 수 있는 권한이 있어야 합니다.)

2. **createDB** 또는 사용자에게 할당된 관리자 역할을 사용하여 사용자, 데이터베이스 및 암호를 만듭니다. 자세한 내용은 [데이터베이스 역할을 참조하십시오](#).
3. 데이터베이스 자격 증명 및 호스트 세부 정보를 자동화 컨트롤러 인벤토리 파일에 외부 데이터베이스로 추가합니다. 다음 예제에서는 기본값을 사용합니다.

```
[database]
pg_host='db.example.com'
pg_port=5432
pg_database='awx'
pg_username='awx'
pg_password='redhat'
```

4. 설치 프로그램을 실행합니다. 자동화 컨트롤러가 있는 PostgreSQL 데이터베이스를 사용하는 경우 연결된 사용자가 데이터베이스를 소유하며 **createDB** 또는 관리자 역할이 할당되어야 합니다.
5. 사용자, 암호 및 데이터베이스 이름을 사용하여 생성된 데이터베이스에 연결할 수 있는지 확인합니다.
6. 사용자 권한을 확인합니다. 사용자에게 **createDB** 또는 관리자 역할이 있어야 합니다.



**참고**

이 절차 중에 외부 데이터베이스 범위를 확인해야 합니다. 자세한 내용은 <https://access.redhat.com/articles/4010491>에서 참조하십시오.

## 4.5.2. 자동화 허브 PostgreSQL 데이터베이스의 hstore 확장 활성화

Ansible Automation Platform 2.4에서 데이터베이스 마이그레이션 스크립트는 **hstore** 필드를 사용하여 정보를 저장하므로 자동화 허브 PostgreSQL 데이터베이스에 대한 **hstore** 확장을 활성화해야 합니다.

이 프로세스는 Ansible Automation Platform 설치 프로그램 및 관리형 PostgreSQL 서버를 사용할 때 자동으로 수행됩니다.

PostgreSQL 데이터베이스가 외부인 경우 자동화 허브를 설치하기 전에 수동으로 **hstore** 확장을 자동화 허브 PostgreSQL 데이터베이스로 활성화해야 합니다.

자동화 허브 설치 전에 **hstore** 확장을 사용하지 않으면 데이터베이스 마이그레이션 중에 오류가 발생합니다.

### 절차

1. PostgreSQL 서버(자동화 허브 데이터베이스)에서 확장 기능을 사용할 수 있는지 확인합니다.

```
$ psql -d <automation hub database> -c "SELECT * FROM pg_available_extensions WHERE name='hstore'"
```

여기서 <automation hub database>의 기본값은 **automationhub**입니다.

#### hstore 를 사용할 수 있는 출력 예

```
name | default_version | installed_version | comment
-----+-----+-----+-----
hstore | 1.7 | | data type for storing sets of (key, value) pairs
(1 row)
```

#### hstore 를 사용할 수 없는 출력 예

```
name | default_version | installed_version | comment
-----+-----+-----+-----
(0 rows)
```

2. RHEL 기반 서버에서 **hstore** 확장 기능은 **postgresql-contrib** RPM 패키지에 포함되어 있으며 PostgreSQL 서버 RPM 패키지를 설치할 때 자동으로 설치되지 않습니다. RPM 패키지를 설치하려면 다음 명령을 사용하십시오.

```
dnf install postgresql-contrib
```

3. 다음 명령을 사용하여 자동화 허브 데이터베이스에 **hstore** PostgreSQL 확장을 생성합니다.

```
$ psql -d <automation hub database> -c "CREATE EXTENSION hstore;"
```

출력 결과는 다음과 같습니다.

```
CREATE EXTENSION
```

4. 다음 출력에서 **installed\_version** 필드에는 **hstore** 가 활성화되어 있음을 나타내는 사용되는 **hstore** 확장이 포함되어 있습니다.

```

name | default_version | installed_version | comment
-----+-----+-----+-----
hstore | 1.7 | 1.7 | data type for storing sets of (key, value) pairs
(1 row)

```

### 4.5.3. Ansible Automation Platform PostgreSQL 데이터베이스에 대한 스토리지 성능 벤치마킹

FIO(flexible I/O Tester) 툴을 사용하여 최소 Ansible Automation Platform PostgreSQL 데이터베이스 요구 사항을 충족하는지 확인합니다. FIO는 스토리지 시스템의 IOPS 성능을 작성하고 작성하는 데 사용되는 툴입니다.

#### 사전 요구 사항

- **Fio**(flexible I/O Tester) 스토리지 성능 벤치마킹 툴을 설치했습니다. **fio** 를 설치하려면 root 사용자로 다음 명령을 실행합니다.

```
# yum -y install fio
```

- **fio** 테스트 데이터 로그 파일을 저장할 충분한 디스크 공간이 있습니다. 절차에 표시된 예제에는 **/tmp** 디렉토리에 최소 60GB의 디스크 공간이 필요합니다.
  - **numjobs** 는 명령으로 실행하는 작업 수를 설정합니다.
  - **size=10G** 는 각 작업에서 생성한 파일 크기를 설정합니다.
- **size** 매개변수 값을 조정했습니다. 이 값을 조정하면 테스트 데이터 양이 줄어듭니다.

#### 절차

1. 임의의 쓰기 테스트를 실행합니다.

```

$ fio --name=write_iops --directory=/tmp --numjobs=3 --size=10G \
--time_based --runtime=60s --ramp_time=2s --ioengine=libaio --direct=1 \
--verify=0 --bs=4K --iodepth=64 --rw=randwrite \
--group_reporting=1 > /tmp/fio_benchmark_write_iops.log \
2>> /tmp/fio_write_iops_error.log

```

2. 임의의 읽기 테스트를 실행합니다.

```

$ fio --name=read_iops --directory=/tmp \
--numjobs=3 --size=10G --time_based --runtime=60s --ramp_time=2s \
--ioengine=libaio --direct=1 --verify=0 --bs=4K --iodepth=64 --rw=randread \
--group_reporting=1 > /tmp/fio_benchmark_read_iops.log \
2>> /tmp/fio_read_iops_error.log

```

3. 결과를 검토합니다.

벤치마크 명령으로 작성된 로그 파일에서 **iops** 로 시작하는 행을 검색합니다. 이 행은 테스트의 최소, 최대값 및 평균 값을 표시합니다.

다음 예제에서는 임의의 읽기 테스트에 대한 로그 파일의 행을 보여줍니다.

```
$ cat /tmp/fio_benchmark_read_iops.log
```



```
read_iops: (g=0): rw=randread, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B,  
ioengine=libaio, iodepth=64  
[...]  
iops      : min=50879, max=61603, avg=56221.33, stdev=679.97, samples=360  
[...]
```

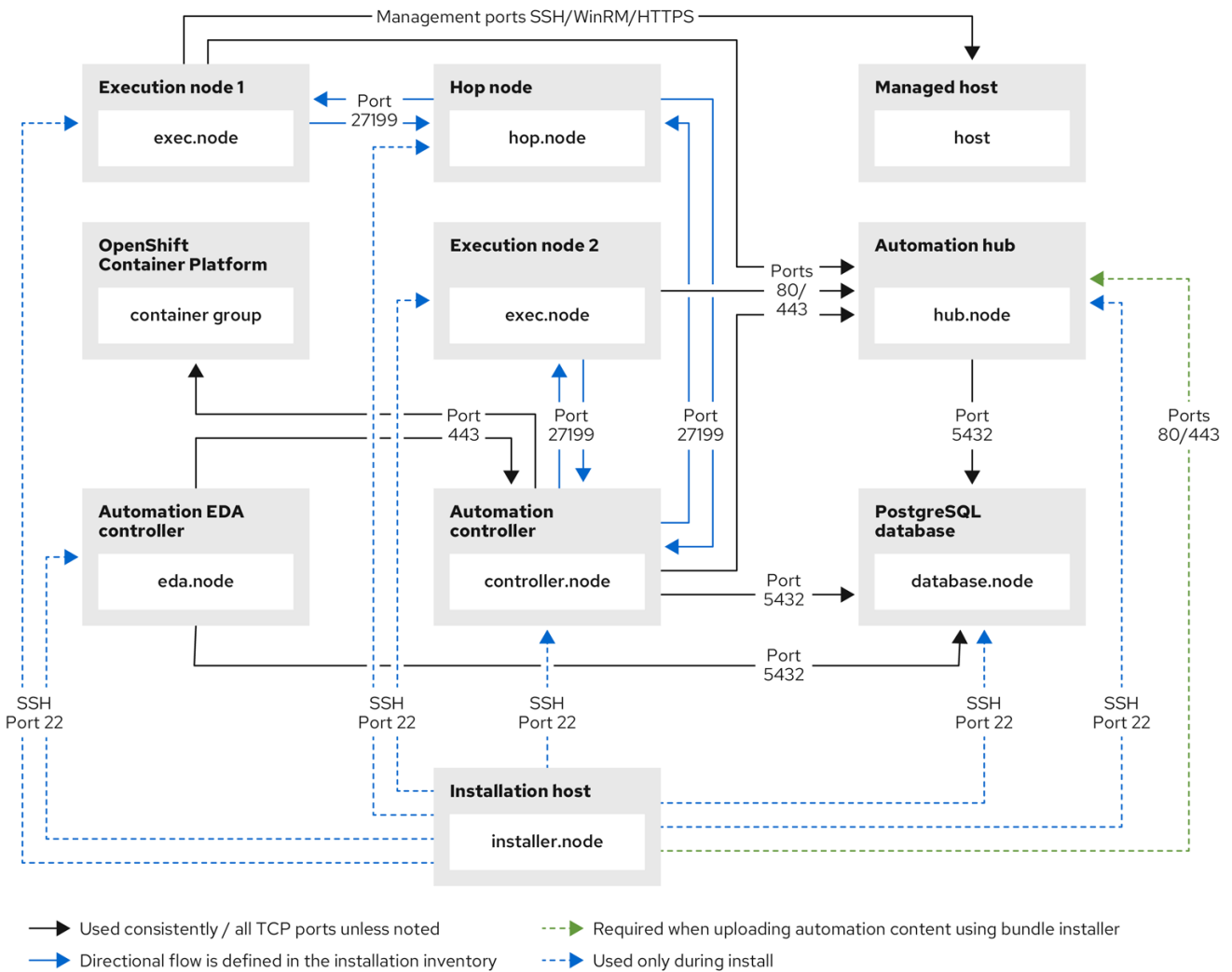
자체 비즈니스 요구 사항, 애플리케이션 워크로드 및 새로운 요구에 따라 로그 파일을 검토, 모니터링 및 다시 방문해야 합니다.

## 5장. 네트워크 포트 및 프로토콜

Red Hat Ansible Automation Platform은 여러 포트를 사용하여 서비스와 통신합니다. 이러한 포트가 열려 있어야 하며 Red Hat Ansible Automation Platform 서버 연결에서 사용할 수 있어야 합니다. 이러한 포트를 사용할 수 있고 서버 방화벽에 의해 차단되지 않았는지 확인합니다.

다음 아키텍처 다이어그램은 가능한 모든 구성 요소가 포함된 완전히 배포된 Ansible Automation Platform의 예입니다.

그림 5.1. Ansible Automation Platform 네트워크 포트 및 프로토콜



637\_Ansible\_0424

다음 표에는 각 애플리케이션에 필요한 기본 Red Hat Ansible Automation Platform 대상 포트가 표시되어 있습니다.



### 참고

나열된 다음 기본 대상 포트 및 설치 프로그램 인벤토리는 구성할 수 있습니다. 환경에 맞게 구성하도록 선택하는 경우 동작 변화가 발생할 수 있습니다.

표 5.1. PostgreSQL

포트	프로토콜	Service	direction	설치 프로그램 인벤토리 변수	필수 항목
22	TCP	SSH	인바운드 및 아웃 바운드	<b>ansible_port</b>	설치 중 원격 액세스
5432	TCP	Postgres	인바운드 및 아웃 바운드	<b>pg_port</b>	기본 포트  컨트롤러에서 데이터베이스 포트로의 ALLOW 연결

표 5.2. 자동화 컨트롤러

포트	프로토콜	Service	direction	설치 프로그램 인벤토리 변수	필수 항목
22	TCP	SSH	인바운드 및 아웃 바운드	<b>ansible_port</b>	설치
80	TCP	HTTP	인바운드	<b>nginx_http_port</b>	UI/API
443	TCP	HTTPS	인바운드	<b>nginx_https_port</b>	UI/API
5432	TCP	PostgreSQL	인바운드 및 아웃 바운드	<b>pg_port</b>	내부 데이터베이스를 다른 구성 요소와 함께 사용하는 경우에만 엽니다. 그렇지 않으면 이 포트가 열려서는 안 됩니다.  클러스터의 하이브리드 모드
27199	TCP	Receptor	인바운드 및 아웃 바운드	<b>receptor_listener_port</b>	필수 및 자동 컨트롤 플레인 클러스터링을 위해 모든 컨트롤러에서 ALLOW 수신기 리스너 포트

표 5.3. 홉 노드

포트	프로토콜	Service	direction	설치 프로그램 인벤토리 변수	필수 항목
22	TCP	SSH	인바운드 및 아웃 바운드	<b>ansible_port</b>	설치
27199	TCP	Receptor	인바운드 및 아웃 바운드	<b>receptor_listener_port</b>	메시  컨트롤러에서 Receptor 포트로의 ALLOW 연결

표 5.4. 실행 노드

포트	프로토콜	Service	direction	설치 프로그램 인벤토리 변수	필수 항목
22	TCP	SSH	인바운드 및 아웃 바운드	<b>ansible_port</b>	설치
80/443	TCP	SSH	인바운드 및 아웃 바운드	고정 값 (표 5.7 Automation Hub의 "사용자 인터페이스" 포트에 매핑)	실행 노드에서 자동화 허브에서 실행 환경 이미지를 가져올 수 있음
27199	TCP	Receptor	인바운드 및 아웃 바운드	<b>receptor_listener_port</b>	Mesh - 노드는 컨트롤러와 직접 피어링됩니다. 홉 노드가 포함되어 있지 않습니다. 27199는 실행 노드에 대한 양방향입니다.  컨트롤러에서 Receptor 포트로의 ALLOW 연결(연결되지 않은 노드)  Hop node(s)에서 Receptor 포트로의 ALLOW 연결 (홉 노드를 통해 중계된 경우)

표 5.5. 제어 노드

포트	프로토콜	Service	direction	설치 프로그램 인벤토리 변수	필수 항목
22	TCP	SSH	인바운드 및 아웃 바운드	<b>ansible_port</b>	설치
27199	TCP	Receptor	인바운드 및 아웃 바운드	<b>receptor_listener_port</b>	<p>Mesh - 노드는 컨트롤러와 직접 피어링됩니다. 직접 노드와 관련된 노드입니다. 27199는 실행 노드에 대한 양방향입니다.</p> <p>연결된 노드의 경우 controller에서 Receptor 포트로의 ENABLE 연결</p> <p>홉 노드를 통해 릴레이하는 경우 hop node에서 Receptor 포트로의 ENABLE 연결</p>
443	TCP	Podman	인바운드	<b>nginx_https_port</b>	UI/API

표 5.6. 하이브리드 노드

포트	프로토콜	Service	direction	설치 프로그램 인벤토리 변수	필수 항목
22	TCP	SSH	인바운드 및 아웃 바운드	<b>ansible_port</b>	설치

포트	프로토콜	Service	direction	설치 프로그램 인벤토리 변수	필수 항목
27199	TCP	Receptor	인바운드 및 아웃 바운드	<b>receptor_listener_port</b>	<p>Mesh - 노드는 컨트롤러와 직접 피어링됩니다. 홉 노드가 포함되어 있지 않습니다. 27199는 실행 노드에 대한 양방향입니다.</p> <p>연결된 노드의 경우 controller에서 Receptor 포트로의 ENABLE 연결</p> <p>홉 노드를 통해 릴레이하는 경우 hop node에서 Receptor 포트로의 ENABLE 연결</p>
443	TCP	Podman	인바운드	<b>nginx_https_port</b>	UI/API

표 5.7. 자동화 허브

포트	프로토콜	Service	direction	설치 프로그램 인벤토리 변수	필수 항목
22	TCP	SSH	인바운드 및 아웃 바운드	<b>ansible_port</b>	설치
80	TCP	HTTP	인바운드	고정 값	사용자 인터페이스
443	TCP	HTTPS	인바운드	고정 값	사용자 인터페이스
5432	TCP	PostgreSQL	인바운드 및 아웃 바운드	<b>automationhub_pg_port</b>	내부 데이터베이스를 다른 구성 요소와 함께 사용하는 경우에만 엽니다. 그렇지 않으면 이 포트가 열려서는 안 됩니다.

표 5.8. Red Hat Insights for Red Hat Ansible Automation Platform

URL	필수 항목
<a href="https://api.access.redhat.com:443">https://api.access.redhat.com:443</a>	일반 계정 서비스, 서브스크립션
<a href="https://cert-api.access.redhat.com:443">https://cert-api.access.redhat.com:443</a>	Insights 데이터 업로드
<a href="https://cert.console.redhat.com:443">https://cert.console.redhat.com:443</a>	인벤토리 업로드 및 Cloud Connector 연결
<a href="https://console.redhat.com:443">https://console.redhat.com:443</a>	Insights 대시보드에 액세스

표 5.9. Automation Hub

URL	필수 항목
<a href="https://console.redhat.com:443">https://console.redhat.com:443</a>	일반 계정 서비스, 서브스크립션
<a href="https://catalog.redhat.com:443">https://catalog.redhat.com:443</a>	실행 환경 인덱싱
<a href="https://sso.redhat.com:443">https://sso.redhat.com:443</a>	TCP
<a href="https://automation-hub-prd.s3.amazonaws.com:443">https://automation-hub-prd.s3.amazonaws.com:443</a> <a href="https://automation-hub-prd.s3.us-east-2.amazonaws.com:443">https://automation-hub-prd.s3.us-east-2.amazonaws.com:443</a>	방화벽 액세스
<a href="https://galaxy.ansible.com:443">https://galaxy.ansible.com:443</a>	Ansible 커뮤니티 큐레이션된 Ansible 콘텐츠
<a href="https://ansible-galaxy-ng.s3.dualstack.us-east-1.amazonaws.com:443">https://ansible-galaxy-ng.s3.dualstack.us-east-1.amazonaws.com:443</a>	커뮤니티 큐레이션 Ansible 콘텐츠 리포지토리를 위한 듀얼 스택 IPv6 끝점
<a href="https://registry.redhat.io:443">https://registry.redhat.io:443</a>	Red Hat 및 파트너사에서 제공하는 컨테이너 이미지에 액세스
<a href="https://cert.console.redhat.com:443">https://cert.console.redhat.com:443</a>	Red Hat 및 파트너 큐레이션 Ansible 컬렉션

표 5.10. 실행 환경(EE)

URL	필수 항목
<a href="https://registry.redhat.io:443">https://registry.redhat.io:443</a>	Red Hat 및 파트너사에서 제공하는 컨테이너 이미지에 액세스
<b>cdn.quay.io:443</b>	Red Hat 및 파트너사에서 제공하는 컨테이너 이미지에 액세스
<b>cdn01.quay.io:443</b>	Red Hat 및 파트너사에서 제공하는 컨테이너 이미지에 액세스

URL	필수 항목
<b>cdn02.quay.io:443</b>	Red Hat 및 파트너사에서 제공하는 컨테이너 이미지에 액세스
<b>cdn03.quay.io:443</b>	Red Hat 및 파트너사에서 제공하는 컨테이너 이미지에 액세스



**중요**

이미지 매니페스트 및 파일 시스템 Blob은 **registry.redhat.io** 에서 직접 제공됩니다. 그러나 2023년 5월 1일부터 파일 시스템 Blob은 **quay.io** 에서 대신 제공됩니다. 컨테이너 이미지를 가져오는 데 문제가 발생하지 않도록 하려면 나열된 **quay.io** 호스트 이름에 대한 아웃바운드 연결을 활성화해야 합니다.

**registry.redhat.io** 에 대한 아웃바운드 연결을 가능하게 하는 방화벽 구성에 대해 이 변경을 수행해야 합니다.

방화벽 규칙을 구성할 때 IP 주소 대신 호스트 이름을 사용합니다.

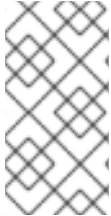
이 변경 후 **registry.redhat.io** 에서 이미지를 계속 가져올 수 있습니다. **quay.io** 로그인도 필요하지 않거나 Red Hat 컨테이너 이미지를 계속 가져오려면 **quay.io** 레지스트리와 직접 상호 작용해야 합니다.

자세한 내용은 [컨테이너 이미지 가져오기에 대한 방화벽 변경 사항](#)을 참조하십시오.



## 6장. RED HAT ANSIBLE AUTOMATION PLATFORM 서브스크립션 연결

Red Hat Ansible Automation Platform을 설치하기 전에 모든 노드에 유효한 서브스크립션이 연결되어 있어야 합니다. Ansible Automation Platform 서브스크립션을 연결하면 설치를 진행하는 데 필요한 서브스크립션 전용 리소스에 액세스할 수 있습니다.



### 참고

Red Hat 계정에서 Simple Content Access 모드를 활성화한 경우 서브스크립션을 연결할 필요가 없습니다. 활성화되면 Ansible Automation Platform을 설치하기 전에 RHSM(Red Hat Subscription Management) 또는 Satellite에 시스템을 등록해야 합니다. 자세한 내용은 [Simple Content Access](#) 를 참조하십시오.

### 절차

1. Red Hat Ansible Automation Platform 서브스크립션의 **pool\_id** 를 가져옵니다.

```
# subscription-manager list --available --all | grep "Ansible Automation Platform" -B 3 -A 6
```



### 참고

Ansible Automation Platform 서브스크립션 연결이 실패할 수 있으므로 서브스크립션에 대해 MCT4022를 **pool\_id** 로 사용하지 마십시오.

### 예제

**subscription-manager list** 명령의 출력 예. **Pool ID:** 섹션에 표시된 대로 **pool\_id** 를 가져옵니다.

```
Subscription Name: Red Hat Ansible Automation, Premium (5000 Managed Nodes)
Provides: Red Hat Ansible Engine
Red Hat Ansible Automation Platform
SKU: MCT3695
Contract: ````
Pool ID: <pool_id>
Provides Management: No
Available: 4999
Suggested: 1
```

2. 서브스크립션을 연결합니다.

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

이제 Red Hat Ansible Automation Platform 서브스크립션을 모든 노드에 연결했습니다.

### 검증

- 서브스크립션이 성공적으로 연결되었는지 확인합니다.

```
# subscription-manager list --consumed
```

### 문제 해결

- Ansible Automation Platform 설치 프로그램과 함께 번들된 특정 패키지를 찾을 수 없거나 **구성 메시지에 의해 비활성화된 리포지토리**가 표시되면 명령을 사용하여 리포지토리를 활성화하십시오.

Red Hat Ansible Automation Platform 2.4 for RHEL 8

```
subscription-manager repos --enable ansible-automation-platform-2.4-for-rhel-8-x86_64-rpms
```

Red Hat Ansible Automation Platform 2.4 for RHEL 9

```
subscription-manager repos --enable ansible-automation-platform-2.4-for-rhel-9-x86_64-rpms
```

## 7장. RED HAT ANSIBLE AUTOMATION PLATFORM 설치 프로그램 램 선택 및 가져오기

Red Hat Enterprise Linux 환경 인터넷 연결을 기반으로 필요한 Red Hat Ansible Automation Platform 설치 프로그램을 선택합니다. 다음 시나리오를 검토하여 요구 사항에 맞는 Red Hat Ansible Automation Platform 설치 프로그램을 결정합니다.

### 7.1. 인터넷으로 설치

Red Hat Enterprise Linux 환경이 인터넷에 연결된 경우 Red Hat Ansible Automation Platform 설치 프로그램을 선택합니다. 인터넷 액세스가 포함된 설치는 최신 필수 리포지토리, 패키지 및 종속성을 검색합니다. 다음 방법 중 하나를 선택하여 Ansible Automation Platform 설치 프로그램을 설정합니다.

#### tarball 설치

1. [Red Hat Ansible Automation Platform 다운로드](#) 페이지로 이동합니다.
2. **Ansible Automation Platform <latest-version> 설정에 대해 지금 다운로드를 클릭합니다.**
3. 파일을 추출합니다.

```
$ tar xvfz ansible-automation-platform-setup-<latest-version>.tar.gz
```

#### RPM 설치

1. Ansible Automation Platform 설치 프로그램 패키지 설치  
V.2.4 for RHEL 8 for x86\_64

```
$ sudo dnf install --enablerepo=ansible-automation-platform-2.4-for-rhel-8-x86_64-rpms  
ansible-automation-platform-installer
```

V.2.4 for RHEL 9 for x86-64

```
$ sudo dnf install --enablerepo=ansible-automation-platform-2.4-for-rhel-9-x86_64-rpms  
ansible-automation-platform-installer
```



#### 참고

리포지토리가 기본적으로 비활성화되어 있으므로 **dnf install** 을 사용하면 리포지토리를 사용할 수 있습니다.

RPM 설치 프로그램을 사용하면 파일이 **/opt/ansible-automation-platform/installer** 디렉터리에 배치됩니다.

### 7.2. 인터넷 액세스없이 설치

인터넷에 액세스할 수 없거나 온라인 리포지토리 및 별도의 구성 요소 및 종속 항목을 설치하지 않으려는 경우 Red Hat Ansible Automation Platform **Bundle** 설치 관리자를 사용합니다. Red Hat Enterprise Linux 리포지토리에 대한 액세스는 여전히 필요합니다. 기타 모든 종속 항목은 tar 아카이브에 포함되어 있습니다.

## 설치

1. [Red Hat Ansible Automation Platform 다운로드](#) 페이지로 이동합니다.
2. **Ansible Automation Platform <latest-version>** 설치 번들로 지금 **다운로드**를 클릭합니다.
3. 파일을 추출합니다.

```
$ tar xvzf ansible-automation-platform-setup-bundle-<latest-version>.tar.gz
```

## 8장. 설치 프로그램 인벤토리 파일 정보

Red Hat Ansible Automation Platform은 인벤토리 파일을 사용하여 논리적으로 구성된 인프라의 관리 노드 또는 호스트 목록과 대조됩니다. Red Hat Ansible Automation Platform 설치 프로그램 인벤토리 파일을 사용하여 설치 시나리오를 지정하고 Ansible에 대한 호스트 배포를 설명할 수 있습니다. Ansible은 인벤토리 파일을 사용하여 단일 명령으로 다수의 호스트를 관리할 수 있습니다. 또한 인벤토리는 지정해야 하는 명령줄 옵션 수를 줄임으로써 Ansible을 보다 효율적으로 사용할 수 있도록 지원합니다.

인벤토리 파일은 보유한 인벤토리 플러그인에 따라 여러 형식 중 하나일 수 있습니다. 가장 일반적인 형식은 **INI** 및 **YAML** 입니다. 이 문서에 나열된 인벤토리 파일은 INI 형식으로 표시됩니다.

인벤토리 파일의 위치는 사용한 설치 프로그램에 따라 다릅니다. 다음 표에서는 가능한 위치를 보여줍니다.

설치 프로그램	위치
번들 tar	<code>/ansible-automation-platform-setup-bundle-&lt;latest-version&gt;</code>
제공되지 않음 tar	<code>/ansible-automation-platform-setup-&lt;latest-version&gt;</code>
RPM	<code>/opt/ansible-automation-platform/installer</code>

다음 명령을 사용하여 인벤토리의 호스트를 확인할 수 있습니다.

```
ansible all -i <path-to-inventory-file> --list-hosts
```

### 인벤토리 파일 예

```
[automationcontroller]
host1.example.com
host2.example.com
Host4.example.com

[automationhub]
host3.example.com

[database]
Host5.example.com

[all:vars]
admin_password='<password>'

pg_host=""
pg_port=""

pg_database='awx'
pg_username='awx'
pg_password='<password>'

registry_url='registry.redhat.io'
registry_username='<registry username>'
registry_password='<registry password>'
```

인벤토리 파일의 첫 번째 부분에서는 Ansible에서 작업할 수 있는 호스트 또는 그룹을 지정합니다.

## 8.1. 호스트 및 그룹 관련 지침

### 데이터베이스

- 외부 데이터베이스를 사용하는 경우 인벤토리 파일의 **[database]** 섹션이 올바르게 설정되어 있는지 확인합니다.
- 성능 향상을 위해 데이터베이스 및 자동화 컨트롤러를 동일한 서버에 공동 배치하지 마십시오.

### 자동화 허브

- **[automationhub]** 그룹이 있는 경우 변수 `automationhub_pg_host` 및 `automationhub_pg_port` 를 포함해야 합니다.
- **[automationhub]** 그룹에 Ansible 자동화 허브 정보를 추가합니다.
- 동일한 노드에 Ansible 자동화 허브 및 자동화 컨트롤러를 설치하지 마십시오.
- **[automationhub]** 및 **[automationcontroller]** 호스트에 연결할 수 있는 IP 주소 또는 FQDN(정규화된 도메인 이름)을 제공하여 사용자가 다른 노드의 Ansible 자동화 허브 및 자동화 컨트롤러에서 콘텐츠를 동기화하고 설치할 수 있도록 합니다.  
FQDN은 올바르게 처리되지 않으므로 - 또는 \_ 기호를 포함하지 않아야 합니다.

**localhost** 를 사용하지 마십시오.

### 프라이빗 자동화 허브

- 동일한 노드에 프라이빗 자동화 허브 및 자동화 컨트롤러를 설치하지 마십시오.
- 동일한 PostgreSQL(데이터베이스) 인스턴스를 사용할 수 있지만 다른 (데이터베이스) 이름을 사용해야 합니다.
- 내부 주소에서 개인 자동화 허브를 설치하고 외부 주소만 포함하는 인증서가 있는 경우 설치 시 인증서 문제가 없는 컨테이너 레지스트리로 사용할 수 없습니다.



#### 중요

**[database]** 그룹이 동시에 설치된 경우 둘 다 구분하지 않으므로 자동화 컨트롤러와 Ansible 자동화 허브 설치를 분리해야 합니다.

**[database]** 에서 하나의 값을 사용하고 자동화 컨트롤러와 Ansible 자동화 허브가 둘 다 정의하는 경우 동일한 데이터베이스를 사용합니다.

### 자동화 컨트롤러

- 자동화 컨트롤러는 사용하는 데이터베이스에 대한 복제 또는 장애 조치를 구성하지 않습니다.
- 자동화 컨트롤러는 보유 중인 모든 복제에서 작동합니다.

### 이벤트 기반 Ansible 컨트롤러

- 이벤트 기반 Ansible 컨트롤러는 별도의 서버에 설치해야 하며 자동화 허브 및 자동화 컨트롤러와 동일한 호스트에 설치할 수 없습니다.

### 클러스터형 설치

- 기존 클러스터를 업그레이드할 때 기존 인스턴스 또는 인스턴스 그룹을 생략하도록 클러스터를 재구성할 수도 있습니다. 인벤토리 파일에서 인스턴스 또는 인스턴스 그룹을 생략하는 것만으로는 클러스터에서 제거하기에 충분하지 않습니다. 인벤토리 파일에서 인스턴스 또는 인스턴스 그룹을 생략하는 것 외에도 업그레이드를 시작하기 전에 인스턴스 또는 인스턴스 그룹을 프로비저닝 해제해야 합니다. 자세한 내용은 [노드 또는 그룹 프로비저닝](#) 을 참조하십시오. 그러지 않으면 생략된 인스턴스 또는 인스턴스 그룹이 클러스터와 계속 통신하므로 업그레이드 중에 자동화 컨트롤러 서비스에 문제가 발생할 수 있습니다.
- 클러스터형 설치 설정을 생성하는 경우 **[localhost]** 를 모든 인스턴스의 호스트 이름 또는 IP 주소로 교체해야 합니다. 자동화 컨트롤러 및 자동화 허브를 위한 설치 프로그램은 **[localhost]** 모든 노드와 인스턴스를 허용하지 않아야 합니다. 이 호스트 이름 또는 주소를 사용하여 다른 노드와 인스턴스에 연결할 수 있어야 합니다. 노드 중 하나에서 localhost **ansible\_connection=local** 을 사용할 수 없습니다. 모든 노드의 호스트 이름에 동일한 형식을 사용합니다. 따라서 이 작업은 작동하지 않습니다.

```
[automationhub]
localhost ansible_connection=local
hostA
hostB.example.com
172.27.0.4
```

대신 다음 형식을 사용하십시오.

```
[automationhub]
hostA
hostB
hostC
```

또는

```
[automationhub]
hostA.example.com
hostB.example.com
hostC.example.com
```

## 8.2. 노드 또는 그룹 프로비저닝 해제

Ansible Automation Platform 설치 프로그램을 사용하여 노드 및 인스턴스 그룹 프로비저닝을 해제할 수 있습니다. 설치 프로그램을 실행하면 모든 구성 파일과 해당 그룹의 노드에 연결된 로그가 제거됩니다.



### 참고

**[automationcontroller]** 그룹에 지정된 첫 번째 호스트를 제외하고 인벤토리의 모든 호스트를 프로비저닝 해제할 수 있습니다.

노드를 프로비저닝 해제하려면 인벤토리 파일 내의 노드 또는 그룹에 **node\_state=deprovision** 을 추가합니다.

예를 들면 다음과 같습니다.

배포에서 단일 노드를 제거하려면 다음을 수행합니다.

```
[automationcontroller]
host1.example.com
host2.example.com
host4.example.com node_state=deprovision
```

또는

배포에서 전체 인스턴스 그룹을 제거하려면 다음을 수행합니다.

```
[instance_group_restrictedzone]
host4.example.com
host5.example.com

[instance_group_restrictedzone:vars]
node_state=deprovision
```

### 8.3. 인벤토리 변수

예제 인벤토리 파일의 두 번째 부분인 **[all:vars]** 은 설치 프로그램에서 사용하는 변수 목록입니다. **all** 을 사용하면 변수가 모든 호스트에 적용됩니다.

특정 호스트에 변수를 적용하려면 **[hostname:vars]** 를 사용합니다. 예를 들면 **[automationhub:vars]** 입니다.

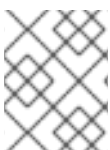
### 8.4. 인벤토리 파일에서 변수를 선언하는 규칙

문자열 변수의 값은 따옴표로 선언됩니다. 예를 들면 다음과 같습니다.

```
pg_database='awx'
pg_username='awx'
pg_password='<password>'
```

**:vars** 섹션에 선언되면 INI 값이 문자열로 해석됩니다. 예를 들어, **var=FALSE** 는 skopeo 와 같은 문자열을 만듭니다. 호스트 행과 달리 **:vars** 섹션은 해당 하나의 항목만 허용하므로 = 이후의 모든 항목은 항목의 값 이어야 합니다. 호스트 행은 해당 여러 **key=value** 매개변수를 허용합니다. 따라서 공백이 구분자가 아닌 값의 일부임을 나타내는 방법이 필요합니다. 공백이 포함된 값을 인용할 수 있습니다(단일 또는 이중). 자세한 내용은 [Python shlex 구문 분석 규칙](#) 을 참조하십시오.

INI 인벤토리에 설정된 변수 값이 특정 유형이어야 하는 경우(예: 문자열 또는 부울 값) 항상 작업에서 필터를 사용하여 유형을 지정합니다. 변수를 사용할 때 INI 인벤토리에 설정된 유형에 의존하지 마십시오.



#### 참고

인벤토리 소스에 YAML 형식을 사용하여 실제 변수 유형에 대한 혼동을 방지하는 것이 좋습니다. YAML 인벤토리 플러그인은 변수 값을 일관되고 올바르게 처리합니다.

Ansible 인벤토리 파일의 매개 변수 값에 #, { 또는 }와 같은 특수 문자가 포함된 경우 값을 두 번 이스케이프해야 합니다(단일 및 이중 인용 표시로 값을 묶습니다).



예를 들어 `mypasswordwith#hashsigns` 를 변수 `pg_password` 의 값으로 사용하려면 Ansible 호스트 인벤토리 파일에서 `pg_password='mypasswordwith#hashsigns'` 로 선언합니다.

## 8.5. 인벤토리 파일의 보안 보안

Ansible Vault를 사용하여 민감한 변수 또는 시크릿 변수를 암호화할 수 있습니다. 그러나 변수 이름과 변수 값을 암호화하면 값의 소스를 찾기가 어렵습니다. 이를 우회하려면 `ansible-vault encrypt_string` 을 사용하여 변수를 개별적으로 암호화하거나 변수가 포함된 파일을 암호화할 수 있습니다.

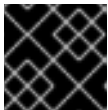
### 절차

1. `credentials.yml` 이라는 레이블이 지정된 파일을 만들어 암호화된 자격 증명을 저장합니다.

```
$ cat credentials.yml
admin_password: my_long_admin_pw
pg_password: my_long_pg_pw
registry_password: my_long_registry_pw
```

2. `ansible-vault` 를 사용하여 `credentials.yml` 파일을 암호화합니다.

```
$ ansible-vault encrypt credentials.yml
New Vault password:
Confirm New Vault password:
Encryption successful
```



### 중요

암호화된 자격 증명 모음 암호를 안전한 장소에 저장합니다.

3. `credentials.yml` 파일이 암호화되었는지 확인합니다.

```
$ cat credentials.yml
$ANSIBLE_VAULT;1.1;
AES256363836396535623865343163333339613833363064653364656138313534353135303
764646165393765393063303065323466663330646232363065316666310a37306230313337
633963383130303334313534383962613632303761636632623932653062343839613639653
6356433656162333133653636616639313864300a3532393734333133396134653263393130
356335653534643565386536316334643438353464323766386235336136663261363433323
131633436393939646132656164333634306335343039356462646330343839663362323033
65383763
```

4. Ansible Automation Platform 2.4 설치를 위해 `setup.sh` 를 실행하고 `credentials.yml` 과 `--ask-vault-pass` 옵션을 모두 전달합니다.

```
$ ANSIBLE_BECOME_METHOD='sudo' ANSIBLE_BECOME=True
ANSIBLE_HOST_KEY_CHECKING=False ./setup.sh -e @credentials.yml -- --ask-vault-pass
```

## 8.6. 추가 인벤토리 파일 변수

인벤토리 파일에 추가 변수를 포함하여 Red Hat Ansible Automation Platform 설치를 추가로 구성할 수 있습니다. 이러한 구성은 Red Hat Ansible Automation Platform을 관리하기 위한 선택적 기능을 추가합니다. 텍스트 편집기를 사용하여 인벤토리 파일을 편집하여 이러한 변수를 추가합니다.

인벤토리 파일 변수에 대해 사전 정의된 값 목록은 *Red Hat Ansible Automation Platform 설치 가이드* 의 [인벤토리 파일 변수](#) 에서 확인할 수 있습니다.

## 9장. 지원되는 설치 시나리오

Red Hat은 Red Hat Ansible Automation Platform에 대해 다음과 같은 설치 시나리오를 지원합니다.

### 추가 리소스

인벤토리 파일 매개변수를 편집하여 지원되는 설치 시나리오를 지정하려면 *Red Hat Ansible Automation Platform 설치 가이드의 설치 시나리오에 따라 인벤토리 파일 예제* 를 참조하십시오.

### 9.1. 동일한 노드 또는 비INSTALLER 관리 데이터베이스의 데이터베이스가 있는 독립 실행형 자동화 컨트롤러

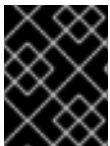
이 시나리오에는 단일 시스템에 웹 프론트 엔드, REST API 백엔드 및 데이터베이스를 포함한 자동화 컨트롤러 설치가 포함됩니다. PostgreSQL을 설치하고 이를 데이터베이스로 사용하도록 자동화 컨트롤러를 구성합니다. 이는 표준 자동화 컨트롤러 설치 시나리오로 간주됩니다.

### 9.2. 외부 관리 데이터베이스가 있는 독립형 자동화 컨트롤러

이 시나리오에는 단일 시스템에 자동화 컨트롤러 서버를 설치하는 작업과 원격 PostgreSQL 인스턴스와 의 통신을 해당 데이터베이스로 구성하는 작업이 포함됩니다. 이 원격 PostgreSQL은 사용자가 관리하는 서버이거나 Amazon RDS와 같은 클라우드 서비스에서 제공할 수 있습니다.

### 9.3. 내부 데이터베이스가 있는 단일 이벤트 기반 ANSIBLE 컨트롤러 노드

이 시나리오에는 내부 데이터베이스가 있는 단일 시스템에 이벤트 기반 Ansible 컨트롤러 설치가 포함됩니다. 자동화 컨트롤러 설치 시나리오와 유사한 설치 관리자 관리 PostgreSQL을 설치합니다.



#### 중요

적절한 Event-Driven Ansible 변수로 인벤토리 파일을 채우기 전에 자동화 컨트롤러를 설치해야 합니다.

### 추가 리소스

- [외부\(installer 관리\) 데이터베이스가 포함된 단일 자동화 컨트롤러, 단일 자동화 허브, 단일 이벤트 기반 Ansible 컨트롤러 노드](#)
- [부록 A.5. 이벤트 기반 Ansible 컨트롤러 변수](#)

### 9.4. 동일한 노드 또는 비INSTALLER 관리 데이터베이스의 데이터베이스가 있는 독립 실행형 자동화 허브

이 시나리오에는 단일 시스템에 웹 프론트 엔드, REST API 백엔드 및 데이터베이스를 포함한 자동화 허브의 설치가 포함됩니다. PostgreSQL을 설치하고 이를 데이터베이스로 사용하도록 자동화 허브를 구성합니다.

### 9.5. 외부 관리 데이터베이스가 있는 독립형 자동화 허브

이 시나리오에는 단일 시스템에 자동화 허브 서버를 설치하고 Red Hat Ansible Automation Platform 설치 프로그램에서 관리하는 원격 PostgreSQL 데이터베이스를 설치하는 작업이 포함됩니다.

## 9.6. 자동화 컨트롤러 노드 또는 비INSTALLER 관리 데이터베이스에 데이터베이스를 사용한 플랫폼 설치

이 시나리오에는 자동화 컨트롤러 노드 또는 설치 프로그램이 아닌 관리형 데이터베이스의 데이터베이스가 있는 자동화 컨트롤러 및 자동화 허브의 설치가 포함됩니다.

## 9.7. 외부 관리 데이터베이스를 사용한 플랫폼 설치

이 시나리오에는 자동화 컨트롤러 및 자동화 허브 설치를 포함하며 원격 PostgreSQL 인스턴스와의 통신을 해당 데이터베이스로 구성합니다. 이 원격 PostgreSQL은 사용자가 관리하는 서버이거나 Amazon RDS와 같은 클라우드 서비스에서 제공할 수 있습니다.

## 9.8. 외부 관리 데이터베이스를 사용한 다중 시스템 클러스터 설치

이 시나리오에는 여러 자동화 컨트롤러 노드 및 자동화 허브 인스턴스를 설치하는 작업이 포함되며, 데이터베이스로 원격 PostgreSQL 인스턴스와의 통신을 구성합니다. 이 원격 PostgreSQL은 사용자가 관리하는 서버이거나 Amazon RDS와 같은 클라우드 서비스에서 제공할 수 있습니다. 이 시나리오에서는 모든 자동화 컨트롤러가 활성 상태이며 작업을 실행할 수 있으며 모든 노드에서 HTTP 요청을 수신할 수 있습니다.



### 참고

- 클러스터 설정에서 실행하려면 자동화 컨트롤러가 외부 PostgreSQL이 되는 데 사용하는 모든 데이터베이스가 기본 또는 보조 Tower 노드 중 하나가 아닌 컴퓨터에 설치해야 합니다. 중복 설정에서 원격 PostgreSQL 버전 요구 사항은 **PostgreSQL 13**입니다.
  - 클러스터형 설정 구성에 대한 자세한 내용은 CloudEvent를 참조하십시오. <https://docs.ansible.com/automation-controller/latest/html/administration/clustering.html>
- **[automationhub]** 호스트의 액세스 가능한 IP 주소를 제공하여 사용자가 다른 노드에서 Private Automation Hub에서 콘텐츠를 동기화할 수 있도록 합니다.