



Red Hat Ansible Automation Platform 2.4

OpenShift Container Platform에 Red Hat Ansible Automation Platform Operator 배포

OpenShift Container Platform에서 Ansible Automation Platform Operator 설치 및 구
성

Red Hat Ansible Automation Platform 2.4 OpenShift Container Platform 에 Red Hat Ansible Automation Platform Operator 배포

OpenShift Container Platform에서 Ansible Automation Platform Operator 설치 및 구성

법적 공지

Copyright © 2023 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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

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

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

All other trademarks are the property of their respective owners.

초록

이 가이드에서는 OpenShift Container Platform의 Red Hat Ansible Automation Platform Operator에 대해 지원되는 설치 시나리오에 대한 절차 및 참조 정보를 제공합니다.

차례

머리말	4
보다 포괄적 수용을 위한 오픈 소스 용어 교체	5
RED HAT 문서에 관한 피드백 제공	6
1장. RED HAT OPENSIFT CONTAINER PLATFORM에 RED HAT ANSIBLE AUTOMATION PLATFORM OPERATOR 설치 계획	7
1.1. ANSIBLE AUTOMATION PLATFORM OPERATOR 정보	7
1.2. OPENSIFT CONTAINER PLATFORM 버전 호환성	7
1.3. RED HAT OPENSIFT CONTAINER PLATFORM에서 지원되는 설치 시나리오	7
1.4. 사용자 정의 리소스	8
1.5. 추가 리소스	8
2장. RED HAT OPENSIFT CONTAINER PLATFORM에 RED HAT ANSIBLE AUTOMATION PLATFORM OPERATOR 설치	9
3장. RED HAT OPENSIFT CONTAINER PLATFORM 웹 콘솔에 자동화 컨트롤러 설치 및 구성	10
3.1. 사전 요구 사항	10
3.2. 자동화 컨트롤러 OPERATOR 설치	10
3.3. RED HAT ANSIBLE AUTOMATION PLATFORM OPERATOR에서 자동화 컨트롤러를 위한 외부 데이터베이스 구성	14
3.4. PVC 검색 및 삭제	15
3.5. 추가 리소스	16
4장. RED HAT OPENSIFT CONTAINER PLATFORM 웹 콘솔에 자동화 허브 설치 및 구성	17
4.1. 사전 요구 사항	17
4.2. 자동화 허브 OPERATOR 설치	17
4.3. 자동화 허브 사용자 인터페이스에 액세스	20
4.4. RED HAT ANSIBLE AUTOMATION PLATFORM OPERATOR에서 자동화 허브를 위한 외부 데이터베이스 구성	21
4.5. PVC 검색 및 삭제	22
4.6. 추가 리소스	23
5장. OPENSIFT CONTAINER PLATFORM CLI에서 ANSIBLE AUTOMATION PLATFORM OPERATOR 설치 ...	24
5.1. 사전 요구 사항	24
5.2. OPENSIFT CONTAINER PLATFORM CLI를 사용하여 OPERATOR에 네임스페이스 가입	24
5.3. OPENSIFT CONTAINER PLATFORM CLI에서 자동화 컨트롤러 로그인 세부 정보 가져오기	25
5.4. 추가 리소스	27
6장. OPENSIFT CONTAINER PLATFORM에서 ANSIBLE AUTOMATION PLATFORM OPERATOR를 사용하여 이벤트 기반 ANSIBLE 컨트롤러 배포	28
7장. 자동화 허브와 함께 RED HAT SINGLE SIGN-ON OPERATOR 사용	30
7.1. KEYCLOAK 인스턴스 생성	30
7.2. ANSIBLE AUTOMATION PLATFORM을 위한 KEYCLOAK 영역 생성	31
7.3. KEYCLOAK 클라이언트 생성	32
7.4. KEYCLOAK 사용자 생성	35
7.5. ANSIBLE AUTOMATION PLATFORM OPERATOR 설치	36
7.6. RED HAT SINGLE SIGN-ON 연결 시크릿 생성	36
7.7. OPERATOR를 사용하여 자동화 허브 설치	37
7.8. 자동화 허브 경로 확인	38
7.9. RED HAT SINGLE SIGN-ON 클라이언트 업데이트	38
7.10. 추가 리소스	39

8장. RED HAT ANSIBLE AUTOMATION PLATFORM을 ANSIBLE AUTOMATION PLATFORM OPERATOR로 마이그레이션	40
8.1. 마이그레이션 고려 사항	40
8.2. 마이그레이션 준비	40
8.3. ANSIBLE AUTOMATION PLATFORM OPERATOR로 데이터 마이그레이션	43
8.4. 마이그레이션 정리 후	43
9장. OPENSIFT CONTAINER PLATFORM에서 ANSIBLE AUTOMATION PLATFORM OPERATOR 업그레이드 .	45
9.1. 업그레이드 고려 사항	45
9.2. 사전 요구 사항	45
9.3. ANSIBLE AUTOMATION PLATFORM OPERATOR 업그레이드	45

머리말

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

이 가이드를 통해 OpenShift Container Platform에 Ansible Automation Platform Operator를 배포하기 위한 설치, 마이그레이션 및 업그레이드 요구 사항을 이해하는 데 도움이 됩니다.

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 [CTO Chris Wright의 메시지](#)를 참조하십시오.

RED HAT 문서에 관한 피드백 제공

Red Hat의 기술 콘텐츠에 대한 귀하의 피드백에 감사드리며, 귀하가 생각하는 것을 알려 주시기 바랍니다. 주석을 추가하거나, 인사이트를 제공하거나, 오타를 수정하거나, 질문을 하려면 문서에서 직접 이 작업을 수행할 수 있습니다.



참고

Red Hat 계정이 있어야 하며 고객 포털에 로그인해야 합니다.

고객 포털에서 문서 피드백을 제출하려면 다음을 수행하십시오.

1. **다중 페이지 HTML** 형식을 선택합니다.
2. 문서 오른쪽 상단에 있는 **피드백** 버튼을 클릭합니다.
3. 피드백을 제공하려는 텍스트 섹션을 강조 표시합니다.
4. 강조 표시된 텍스트 옆에 있는 **피드백 추가** 대화 상자를 클릭합니다.
5. 페이지 오른쪽에 있는 텍스트 상자에 피드백을 입력한 다음 **제출**을 클릭합니다.

피드백을 제출할 때마다 추적 문제가 자동으로 생성됩니다. **Submit** 을 클릭한 후 표시되는 링크를 열고 문제 모니터링을 시작하거나 의견을 더 추가합니다.

1장. RED HAT OPENSIFT CONTAINER PLATFORM에 RED HAT ANSIBLE AUTOMATION PLATFORM OPERATOR 설치 계획

Red Hat Ansible Automation Platform은 Red Hat Enterprise Linux 및 Red Hat OpenShift 모두에서 지원됩니다.

OpenShift Operator는 Red Hat OpenShift Container Platform에 복잡한 분산 소프트웨어의 Day-2 작업을 설치하고 자동화하는 데 도움이 됩니다. Ansible Automation Platform Operator를 사용하면 Red Hat OpenShift Container Platform에 Ansible Automation Platform 구성 요소를 배포하고 관리할 수 있습니다.

이 섹션을 사용하면 Red Hat OpenShift Container Platform 환경에 Red Hat Ansible Automation Platform 설치를 계획하는 데 도움이 될 수 있습니다. 설치하기 전에 지원되는 설치 시나리오를 검토하여 요구 사항을 충족하는지 확인합니다.

1.1. ANSIBLE AUTOMATION PLATFORM OPERATOR 정보

Ansible Automation Platform Operator는 OpenShift 환경에서 새로운 Ansible Automation Platform 인스턴스를 클라우드 네이티브 푸시 버튼을 배포합니다. Ansible Automation Platform Operator에는 Automation 컨트롤러 및 Private Automation 허브의 인스턴스를 배포하고 관리하는 리소스 유형이 포함되어 있습니다. 또한 자동화 컨트롤러 배포 내에서 작업을 정의하고 시작하기 위한 자동화 컨트롤러 작업 리소스가 포함되어 있습니다.

Kubernetes 네이티브 Operator를 사용하여 Ansible Automation Platform 인스턴스를 배포하면 Red Hat OpenShift Container Platform에 배포된 플레이북에서 인스턴스를 시작할 때 Red Hat Ansible Automation Platform 배포에 대한 업그레이드 및 전체 라이프사이클 지원을 포함하여 여러 가지 장점이 있습니다.

OperatorHub의 Red Hat Operator 카탈로그에서 Ansible Automation Platform Operator를 설치할 수 있습니다.

1.2. OPENSIFT CONTAINER PLATFORM 버전 호환성

Ansible Automation Platform 2.4를 설치하는 Ansible Automation Platform Operator는 OpenShift Container Platform 4.9 이상 버전에서 사용할 수 있습니다.

추가 리소스

- 최신 호환성 세부 사항은 [Red Hat Ansible Automation Platform 라이프 사이클](#)을 참조하십시오.

1.3. RED HAT OPENSIFT CONTAINER PLATFORM에서 지원되는 설치 시나리오

Red Hat OpenShift Container Platform 웹 콘솔에서 OperatorHub를 사용하여 Ansible Automation Platform Operator를 설치할 수 있습니다.

또는 OpenShift Container Platform CLI(명령줄 인터페이스), **oc** 에서 Ansible Automation Platform Operator를 설치할 수 있습니다.

아래 워크플로우 중 하나를 수행하여 Ansible Automation Platform Operator를 설치하고 이를 사용하여 필요한 Ansible Automation Platform의 구성 요소를 설치합니다.

- 자동화 컨트롤러 사용자 정의 리소스를 먼저 사용한 다음 허브 사용자 지정 리소스를 자동화합니다.
- 자동화 허브 사용자 정의 리소스를 먼저 사용한 다음 컨트롤러 사용자 정의 리소스를 자동화합니다.
- 자동화 컨트롤러 사용자 정의 리소스
- 자동화 허브 사용자 정의 리소스.

1.4. 사용자 정의 리소스

각 기본 설치 워크플로에 대한 사용자 정의 리소스를 정의할 수 있습니다.

1.5. 추가 리소스

- OpenShift Container Platform [OperatorHub](#)에 대한 자세한 내용은 [OperatorHub 이해](#) 를 참조하십시오.

2장. RED HAT OPENSIFT CONTAINER PLATFORM에 RED HAT ANSIBLE AUTOMATION PLATFORM OPERATOR 설치

사전 요구 사항

- OperatorHub에 Red Hat Ansible Automation Platform 카탈로그를 설치했습니다.
- 플랫폼에 대한 **StorageClass** 오브젝트와 **ReadWriteMany** 액세스 모드를 사용하여 PVC(영구 볼륨 클레임)를 생성했습니다. 자세한 내용은 [동적 프로비저닝](#) 을 참조하십시오.
- **ReadWriteMany** 액세스 모드를 사용하여 AWS(Amazon Web Services)에서 Red Hat OpenShift Container Platform 클러스터를 실행하려면 NFS 또는 기타 스토리지를 추가해야 합니다.
 - AWS EBS(Elastic Block Store)에 대한 정보 또는 **aws-efs** 스토리지 클래스를 사용하려면 [AWS Elastic Block Store를 사용한 영구 스토리지를](#) 참조하십시오.
 - AWS EBS에 다중 연결 **ReadWriteMany** 액세스 모드를 사용하려면 [Amazon EBS Multi-Attach를 사용하여 여러 인스턴스에 볼륨 연결을](#) 참조하십시오.

절차

1. Red Hat OpenShift Container Platform에 로그인합니다.
2. **Operators** → **OperatorHub** 로 이동합니다.
3. Red Hat Ansible Automation Platform Operator를 검색하고 **설치**를 클릭합니다.
4. **업데이트 채널**을 선택합니다.
 - **stable-2.x**: Operator가 설치된 네임스페이스에 자동화 허브 및 자동화 컨트롤러 인스턴스의 배포를 제한하는 네임스페이스 범위 Operator를 설치합니다. 이는 대부분의 경우에 적합합니다. **stable-2.x** 채널은 관리자 권한이 필요하지 않으며 단일 네임스페이스만 모니터링하므로 더 적은 리소스를 활용합니다.
 - **stable-2.x-cluster-scoped**: 클러스터의 여러 네임스페이스에 자동화 허브 및 자동화 컨트롤러를 배포하고 클러스터의 모든 네임스페이스에 대한 관리자 권한이 필요합니다.
5. **설치 모드, 설치된 네임스페이스, 승인 전략**을 선택합니다.
6. **설치**를 클릭합니다.

설치 프로세스가 시작됩니다. 설치가 완료되면 모달에서 Red Hat Ansible Automation Platform Operator가 지정된 네임스페이스에 설치되어 있음을 알리는 메시지가 표시됩니다.

- **Operator 보기** 를 클릭하여 새로 설치된 Red Hat Ansible Automation Platform Operator를 확인합니다.

3장. RED HAT OPENSIFT CONTAINER PLATFORM 웹 콘솔에 자동화 컨트롤러 설치 및 구성

이러한 지침을 사용하여 Red Hat OpenShift Container Platform에 자동화 컨트롤러 Operator를 설치하고, 사용자 정의 리소스를 지정하며, 외부 데이터베이스를 사용하여 Ansible Automation Platform을 배포할 수 있습니다.

자동화 컨트롤러 구성은 자동화 컨트롤러 `extra_settings`를 통해 수행하거나 배포 후 사용자 인터페이스에서 직접 수행할 수 있습니다. 그러나 `extra_settings`에서의 구성이 사용자 인터페이스의 설정보다 우선한다는 점에 유의해야 합니다.



참고

자동화 컨트롤러 인스턴스가 제거되면 연결된 PVC가 자동으로 삭제되지 않습니다. 이로 인해 새 배포의 이름이 이전 배포와 동일한 경우 마이그레이션 중에 문제가 발생할 수 있습니다. 따라서 동일한 네임스페이스에 새 자동화 컨트롤러 인스턴스를 배포하기 전에 이전 PVC를 수동으로 제거하는 것이 좋습니다. 자세한 내용은 [PVC 찾기 및 삭제](#) 를 참조하십시오.

3.1. 사전 요구 사항

- Operator Hub에 Red Hat Ansible Automation Platform 카탈로그를 설치했습니다.
- 컨트롤러의 경우 Operator가 필요한 PVC를 동적으로 생성할 수 있도록 클러스터에 기본 StorageClass를 구성해야 합니다. 외부 PostgreSQL 데이터베이스가 구성된 경우에는 이 작업이 필요하지 않습니다.
- Hub의 경우 ReadWriteMany를 지원하는 StorageClass를 클러스터에서 사용할 수 있어야 콘텐츠, redis 및 api Pod에 필요한 PVC를 동적으로 생성할 수 있습니다. 클러스터의 기본 StorageClass가 아닌 경우 AutomationHub 오브젝트를 생성할 때 지정할 수 있습니다.

3.2. 자동화 컨트롤러 OPERATOR 설치

자동화 컨트롤러 Operator를 설치하려면 다음 절차를 사용하십시오.

절차

1. Operators(운영자) **InstalledOperators** (설치된 Operator) 로 이동한 다음 **Ansible Automation Platform Operator**를 클릭합니다.
2. **Automation Controller** 탭을 찾은 다음 **Create instance** 를 클릭합니다.

양식 보기 또는 YAML 보기를 사용하여 인스턴스를 구성할 수 있습니다.

3.2.1. 자동화 컨트롤러 양식 보기 생성

뷰를 사용하여 자동화 컨트롤러를 생성하려면 다음 절차를 사용하십시오.

절차

1. **양식 보기** 가 선택되어 있는지 확인합니다. 기본적으로 선택해야 합니다.
2. 새 컨트롤러의 이름을 입력합니다.

3. 선택 사항: 필요한 레이블을 추가합니다.
4. 고급 구성을 클릭합니다.
5. 인스턴스의 호스트 이름을 입력합니다. 호스트 이름은 선택 사항입니다. 선택한 배포 이름에 따라 기본 호스트 이름이 생성됩니다.
6. Admin 계정 사용자 이름을 입력합니다.
7. 관리자 이메일 주소를 입력합니다.
8. Admin password secret 드롭다운 메뉴에서 시크릿을 선택합니다.
9. 데이터베이스 구성 시크릿 드롭다운 메뉴에서 시크릿을 선택합니다.
10. Old Database 구성 시크릿 드롭다운 메뉴에서 시크릿을 선택합니다.
11. Secret key secret 드롭다운 메뉴에서 시크릿을 선택합니다.
12. Broadcast Websocket Secret 드롭다운 메뉴에서 시크릿을 선택합니다.
13. 필요한 모든 서비스 계정 주석을 입력합니다.

3.2.2. 컨트롤러 이미지 가져오기 정책 구성

자동화 컨트롤러에서 이미지 가져오기 정책을 구성하려면 다음 절차를 사용하십시오.

절차

1. Image Pull Policy 에서 라디오 버튼을 클릭하여 선택합니다.
 - Always
 - Never
 - IfNotPresent
2. 이미지 가져오기 시크릿 아래에 옵션을 표시하려면 화살표를 클릭합니다.
 - a. + beside Image Pull Secret 을 클릭하고 값을 입력합니다.
3. Web 컨테이너 리소스 요구 사항 드롭다운 목록에 필드를 표시하려면 화살표를 클릭합니다.
 - a. Limits, Requests 아래에서 CPU 코어, Memory, Storage 값을 입력합니다.
4. Task 컨테이너 리소스 요구 사항 드롭다운 목록에 필드를 표시하려면 화살표를 클릭합니다.
 - a. Limits, Requests 아래에서 CPU 코어, Memory, Storage 값을 입력합니다.
5. EE 컨트롤 플레인 컨테이너 리소스 요구 사항 드롭다운 목록에 필드를 표시하려면 화살표를 클릭합니다.
 - a. Limits, Requests 아래에서 CPU 코어, Memory, Storage 값을 입력합니다.
6. PostgreSQL init 컨테이너 리소스 요구 사항(관리 서비스 사용 시) 드롭다운 목록에 필드를 표시하려면 화살표를 클릭합니다.
 - a. Limits, Requests 아래에서 CPU 코어, Memory, Storage 값을 입력합니다.

7. **Redis 컨테이너 리소스 요구 사항** 드롭다운 목록에 필드를 표시하려면 화살표를 클릭합니다.
 - a. **Limits, Requests** 아래에서 **CPU 코어, Memory, Storage** 값을 입력합니다.
8. **PostgreSQL 컨테이너 리소스 요구 사항(관리된 인스턴스를 사용하는 경우)*** 드롭다운 목록에 필드를 표시하려면 화살표를 클릭합니다.
 - a. **Limits, Requests** 아래에서 **CPU 코어, Memory, Storage** 값을 입력합니다.
9. **PostgreSQL 컨테이너 스토리지 요구 사항(관리 인스턴스 사용 시)** 드롭다운 목록을 표시하려면 화살표를 클릭합니다.
 - a. **Limits, Requests** 아래에서 **CPU 코어, Memory, Storage** 값을 입력합니다.
10. Replicas에서 인스턴스 복제본 수를 입력합니다.
11. 인스턴스 제거 시 사용된 시크릿 제거에서 **true** 또는 **false** 를 선택합니다. 기본값은 false입니다.
12. 생성 시 데이터가 있는 사전 로드 인스턴스에서 **true** 또는 **false** 를 선택합니다. 기본값은 true입니다.

3.2.3. 컨트롤러 LDAP 보안 구성

자동화 컨트롤러에 대한 LDAP 보안을 구성하려면 다음 절차를 사용하십시오.

절차

1. **LDAP 인증 기관 신뢰 번들**에서 드롭다운 메뉴를 클릭하고 시크릿을 선택합니다.
2. **LDAP Password Secret**에서 드롭다운 메뉴를 클릭하고 시크릿을 선택합니다.
3. **EE 이미지 가져오기 자격 증명 시크릿**에서 드롭다운 메뉴를 클릭하고 시크릿을 선택합니다.
4. **Bundle Cacert Secret**에서 드롭다운 메뉴를 클릭하고 시크릿을 선택합니다.
5. **서비스 유형**에서 드롭다운 메뉴를 클릭하고 선택합니다.
 - ClusterIP
 - LoadBalancer
 - NodePort

3.2.4. 자동화 컨트롤러 Operator 경로 옵션 구성

Red Hat Ansible Automation Platform Operator 설치 양식을 사용하면 **고급 구성**에서 자동화 컨트롤러 Operator 경로 옵션을 추가로 구성할 수 있습니다.

절차

1. **고급 구성**을 클릭합니다.
2. **Ingress 유형**에서 드롭다운 메뉴를 클릭하고 **경로** 를 선택합니다.
3. **Route DNS host**에서 경로에서 응답하는 공통 호스트 이름을 입력합니다.

4. **Route TLS 종료 메커니즘**에서 드롭다운 메뉴를 클릭하고 **Edge** 또는 **Passthrough** 를 선택합니다. 대부분의 경우 **Edge** 를 선택해야 합니다.
5. **Route TLS 인증 정보 시크릿**에서 드롭다운 메뉴를 클릭하고 목록에서 시크릿을 선택합니다.
6. **Enable persistence for /var/lib/projects** 디렉토리에 대한 지속성 아래에서 CloudEvent를 이동하여 true 또는 false를 선택합니다.

3.2.5. 자동화 컨트롤러 Operator의 Ingress 유형 구성

Red Hat Ansible Automation Platform Operator 설치 양식을 사용하면 **고급 구성**에서 자동화 컨트롤러 Operator Ingress를 추가로 구성할 수 있습니다.

절차

1. **고급 구성**을 클릭합니다.
2. **Ingress 유형**에서 드롭다운 메뉴를 클릭하고 **Ingress** 를 선택합니다.
3. **Ingress 주석**에서 Ingress에 추가할 주석을 입력합니다.
4. **Ingress TLS 시크릿**에서 드롭다운 메뉴를 클릭하고 목록에서 시크릿을 선택합니다.

자동화 컨트롤러 Operator를 구성한 후 양식 보기 하단에서 **생성**을 클릭합니다. Red Hat OpenShift Container Platform은 이제 Pod를 생성합니다. 이 작업은 몇 분 정도 걸릴 수 있습니다.

Workloads → **Pod** 로 이동하여 새로 생성된 인스턴스를 찾아 진행 상황을 확인할 수 있습니다.

검증

자동화 컨트롤러에서 Ansible Automation Platform Operator 설치에서 제공하는 다음 Operator Pod가 실행 중인지 확인합니다.

Operator 관리자 컨트롤러	자동화 컨트롤러	자동화 허브
<p>3 Operator 각각에 대한 Operator 관리자 컨트롤러는 다음을 포함합니다.</p> <ul style="list-style-type: none"> ● automation-controller-operator-controller-manager ● automation-hub-operator-controller-manager ● resource-operator-controller-manager 	<p>자동화 컨트롤러를 배포한 후 이러한 Pod가 추가된 것을 확인할 수 있습니다.</p> <ul style="list-style-type: none"> ● 컨트롤러 ● controller-postgres 	<p>자동화 허브를 배포한 후 다음 Pod가 추가된 것을 확인할 수 있습니다.</p> <ul style="list-style-type: none"> ● hub-api ● hub-content ● hub-postgres ● hub-redis ● hub-worker



참고

누락된 Pod는 풀 시크릿이 필요함을 나타낼 수 있습니다. 보호되는 이미지 레지스트리 또는 개인 이미지 레지스트리에는 pull secret이 필요합니다. 자세한 내용은 [이미지 풀 시크릿](#) 사용을 참조하십시오. **oc describe pod <pod-name>**을 실행하여 해당 Pod에 ImagePullBackOff 오류가 있는지 확인하여 이 문제를 추가로 진단할 수 있습니다.

3.3. RED HAT ANSIBLE AUTOMATION PLATFORM OPERATOR에서 자동화 컨트롤러를 위한 외부 데이터베이스 구성

외부 데이터베이스를 사용하여 Ansible Automation Platform을 배포하려는 사용자의 경우 인스턴스 자격 증명 및 연결 정보를 사용하여 시크릿을 구성한 다음 **oc create** 명령을 사용하여 클러스터에 적용할 수 있습니다.

기본적으로 Red Hat Ansible Automation Platform Operator는 Ansible Automation Platform 배포와 동일한 네임스페이스에서 관리되는 PostgreSQL Pod를 자동으로 생성하고 구성합니다. Red Hat Ansible Automation Platform Operator가 자동으로 생성하는 관리되는 PostgreSQL Pod 대신 외부 데이터베이스를 사용하여 Ansible Automation Platform을 배포할 수 있습니다.

외부 데이터베이스를 사용하면 리소스를 공유 및 재사용하고 백업, 업그레이드 및 성능 최적화를 수동으로 관리할 수 있습니다.



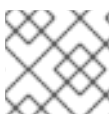
참고

데이터베이스 이름이 다른 경우 자동화 허브 및 자동화 컨트롤러에 동일한 외부 데이터베이스(PostgreSQL 인스턴스)를 사용할 수 있습니다. 즉, 단일 PostgreSQL 인스턴스 내에 이름이 다른 여러 개의 데이터베이스가 있을 수 있습니다.

다음 섹션에서는 Ansible Automation Platform Operator에서 자동화 컨트롤러에 대한 외부 데이터베이스를 구성하는 단계를 간략하게 설명합니다.

사전 요구 사항

외부 데이터베이스는 현재 Ansible Automation Platform 릴리스에서 지원하는 버전인 PostgreSQL 데이터베이스여야 합니다.



참고

Ansible Automation Platform 2.4에서는 PostgreSQL 13을 지원합니다.

절차

외부 postgres 인스턴스 자격 증명 및 연결 정보는 자동화 컨트롤러 사양에 설정되는 시크릿에 저장해야 합니다.

1. 아래 템플릿에 따라 **postgres_configuration_secret**.yaml 파일을 생성합니다.

```
apiVersion: v1
kind: Secret
metadata:
  name: external-postgres-configuration
  namespace: <target_namespace> 1
stringData:
  host: "<external_ip_or_url_resolvable_by_the_cluster>" 2
```

```
port: "<external_port>" 3
database: "<desired_database_name>"
username: "<username_to_connect_as>"
password: "<password_to_connect_with>" 4
sslmode: "prefer" 5
type: "unmanaged"
type: Opaque
```

- 1 보안을 생성할 네임스페이스입니다. 배포하려는 네임스페이스와 동일해야 합니다.
 - 2 데이터베이스 노드의 확인 가능한 호스트 이름입니다.
 - 3 외부 포트는 기본적으로 **5432**로 설정됩니다.
 - 4 변수 암호 값에는 배포, 백업 또는 복원 중에 문제가 발생하지 않도록 작은따옴표(' ') 또는 백슬래시(\)가 포함되지 않아야 합니다.
 - 5 변수 **sslmode** 는 외부 데이터베이스에만 유효합니다. 허용되는 값은 **prefer, disable, allow, require, verify-ca, verify-full** 입니다.
2. **oc create** 명령을 사용하여 **external-postgres-configuration-secret.yml** 을 클러스터에 적용합니다.

```
$ oc create -f external-postgres-configuration-secret.yml
```

3. **AutomationController** 사용자 정의 리소스 오브젝트를 생성할 때 사양의 보안을 지정합니다. 아래 예제를 따르십시오.

```
apiVersion: awx.ansible.com/v1beta1
kind: AutomationController
metadata:
  name: controller-dev
spec:
  postgres_configuration_secret: external-postgres-configuration
```

3.4. PVC 검색 및 삭제

PVC(영구 볼륨 클레임)는 자동화 허브 및 자동화 컨트롤러 애플리케이션에서 사용하는 데이터를 저장하는 데 사용되는 스토리지 볼륨입니다. 이러한 PVC는 애플리케이션과 독립적이며 애플리케이션이 삭제된 경우에도 유지됩니다. 더 이상 PVC가 필요하지 않거나 다른 위치에서 백업한 경우 수동으로 삭제할 수 있습니다.

절차

1. 배포 네임스페이스의 기존 PVC를 나열합니다.

```
oc get pvc -n <namespace>
```

2. 이전 배포 이름과 PVC 이름을 비교하여 이전 배포와 연결된 PVC를 확인합니다.
3. 이전 PVC를 삭제합니다.

```
oc delete pvc -n <namespace> <pvc-name>
```

3.5. 추가 리소스

- OpenShift Container Platform에서 Operator를 실행하는 방법에 대한 자세한 내용은 [OpenShift Container Platform 제품 설명서](#)로 이동하여 OpenShift Container Platform에서 Operator 작업 가이드를 클릭합니다.

4장. RED HAT OPENSIFT CONTAINER PLATFORM 웹 콘솔에 자동화 허브 설치 및 구성

이 지침을 사용하여 Red Hat OpenShift Container Platform에 자동화 허브 Operator를 설치하고, 사용자 정의 리소스를 지정하며, 외부 데이터베이스를 사용하여 Ansible Automation Platform을 배포할 수 있습니다.

자동화 허브 구성은 자동화 허브 pulp_settings를 통해 수행하거나 배포 후 사용자 인터페이스에서 직접 수행할 수 있습니다. 그러나 pulp_settings의 구성이 사용자 인터페이스의 설정보다 우선한다는 점에 유의해야 합니다. Hub 설정은 항상 Hub 사용자 정의 리소스 사양에서 소문자로 설정해야 합니다.



참고

자동화 허브 인스턴스가 제거되면 PVC가 자동으로 삭제되지 않습니다. 이로 인해 새 배포의 이름이 이전 배포와 동일한 경우 마이그레이션 중에 문제가 발생할 수 있습니다. 따라서 동일한 네임스페이스에 새 자동화 허브 인스턴스를 배포하기 전에 이전 PVC를 수동으로 제거하는 것이 좋습니다. 자세한 내용은 [PVC 찾기 및 삭제](#) 를 참조하십시오.

4.1. 사전 요구 사항

- Operator Hub에 Red Hat Ansible Automation Platform Operator가 설치되어 있습니다.

4.2. 자동화 허브 OPERATOR 설치

자동화 허브 Operator를 설치하려면 다음 절차를 사용하십시오.

절차

- Operators(운영자) **InstalledOperators** 로 이동합니다.
- Automation hub** 항목을 찾은 다음 **Create instance** 를 클릭합니다.

4.2.1. Red Hat OpenShift Container Platform에 Ansible Automation Platform Operator 설치를 위한 스토리지 옵션

자동화 허브에는 여러 Pod가 컬렉션과 같은 공유 콘텐츠에 액세스할 수 있도록 작업을 위해 **ReadWriteMany** 파일 기반 스토리지, Azure Blob 스토리지 또는 Amazon S3 호환 스토리지가 필요합니다.

AutomationHub CR에서 오브젝트 스토리지를 구성하는 프로세스는 Amazon S3 및 Azure Blob Storage와 유사합니다.

파일 기반 스토리지를 사용하고 있으며 설치 시나리오에 자동화 허브가 포함된 경우 Ansible Automation Platform Operator의 스토리지 옵션을 **ReadWriteMany** 로 설정해야 합니다. **ReadWriteMany** 는 기본 스토리지 옵션입니다.

또한 OpenShift Data Foundation은 **ReadWriteMany** 또는 S3 호환 구현을 제공합니다. 또한 **ReadWriteMany** 를 지원하도록 NFS 스토리지 구성을 설정할 수도 있습니다. 그러나 이로 인해 NFS 서버가 잠재적인 단일 장애 지점으로 도입되었습니다.

추가 리소스

- OpenShift Container Platform [스토리지 가이드의 NFS를 사용하는 영구저장장치](#)

- IBM의 OpenShift 환경에서 NFS 동적 스토리지 프로비저닝을 위한 스토리지 클래스를 생성하는 방법은 무엇입니까?

4.2.1.1. ReadWriteMany 액세스 모드로 OCP 스토리지 프로비저닝

Ansible Automation Platform Operator를 성공적으로 설치하려면 처음에 **ReadWriteMany** 액세스 모드로 자동화 허브를 위한 스토리지 유형을 프로비저닝해야 합니다.

절차

1. **Provisioning** (프로비저닝)을 클릭하여 액세스 모드를 업데이트합니다.
2. 첫 번째 단계에서 **accessModes**를 기본 **ReadWriteOnce**에서 **ReadWriteMany**로 업데이트합니다.
3. PVC(영구 볼륨 클레임)를 생성하려면 이 섹션의 추가 단계를 완료합니다.

4.2.1.2. Amazon S3에서 오브젝트 스토리지 구성

Red Hat은 자동화 허브를 위해 Amazon S3(Simple Storage Service)를 지원합니다. **AutomationHub** CR(사용자 정의 리소스)을 배포할 때 구성하거나 기존 인스턴스에 대해 구성할 수 있습니다.

사전 요구 사항

- Amazon S3 버킷을 생성하여 오브젝트를 저장합니다.
- S3 버킷의 이름을 기록해 둡니다.

절차

1. AWS 인증 정보 및 연결 세부 정보와 Amazon S3 버킷의 이름이 포함된 Kubernetes 시크릿을 생성합니다. 다음 예제에서는 **test-s3**라는 시크릿을 생성합니다.

```
$ oc -n $HUB_NAMESPACE apply -f- <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: 'test-s3'
stringData:
  s3-access-key-id: $S3_ACCESS_KEY_ID
  s3-secret-access-key: $S3_SECRET_ACCESS_KEY
  s3-bucket-name: $S3_BUCKET_NAME
  s3-region: $S3_REGION
EOF
```

2. 자동화 허브 CR(사용자 정의 리소스) 사양에 보안을 추가합니다.

```
spec:
  object_storage_s3_secret: test-s3
```

3. 이 보안을 기존 인스턴스에 적용하는 경우 API Pod를 재시작하여 변경 사항을 적용합니다. **<Hub-name>**은 hub 인스턴스의 이름입니다.

```
$ oc -n $HUB_NAMESPACE delete pod -l app.kubernetes.io/name=<hub-name>-api
```

4.2.1.3. Azure Blob 에서 오브젝트 스토리지 구성

Red Hat은 자동화 허브를 위한 Azure Blob Storage를 지원합니다. **AutomationHub** CR(사용자 정의 리소스)을 배포할 때 구성하거나 기존 인스턴스에 대해 구성할 수 있습니다.

사전 요구 사항

- 오브젝트를 저장할 Azure Storage Blob 컨테이너를 만듭니다.
- Blob 컨테이너의 이름을 기록해 둡니다.

절차

1. Azure 계정에 대한 자격 증명 및 연결 세부 정보와 Azure Storage Blob 컨테이너의 이름이 포함된 Kubernetes 시크릿을 생성합니다. 다음 예제에서는 **test-azure** 라는 시크릿을 생성합니다.

```
$ oc -n $HUB_NAMESPACE apply -f <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: 'test-azure'
stringData:
  azure-account-name: $AZURE_ACCOUNT_NAME
  azure-account-key: $AZURE_ACCOUNT_KEY
  azure-container: $AZURE_CONTAINER
  azure-container-path: $AZURE_CONTAINER_PATH
  azure-connection-string: $AZURE_CONNECTION_STRING
EOF
```

2. 자동화 허브 CR(사용자 정의 리소스) 사양에 보안을 추가합니다.

```
spec:
  object_storage_azure_secret: test-azure
```

3. 이 보안을 기존 인스턴스에 적용하는 경우 API Pod를 재시작하여 변경 사항을 적용합니다. **<Hub-name>**은 hub 인스턴스의 이름입니다.

```
$ oc -n $HUB_NAMESPACE delete pod -l app.kubernetes.io/name=<hub-name>-api
```

4.2.2. 자동화 허브 Operator 경로 옵션 구성

Red Hat Ansible Automation Platform Operator 설치 양식을 사용하면 고급 구성에서 자동화 허브 운영자 경로 옵션을 추가로 구성할 수 있습니다.

절차

1. 고급 구성을 클릭합니다.
2. Ingress 유형에서 드롭다운 메뉴를 클릭하고 경로를 선택합니다.
3. Route DNS host 에서 경로에서 응답하는 공통 호스트 이름을 입력합니다.
4. Route TLS 종료 메커니즘 에서 드롭다운 메뉴를 클릭하고 Edge 또는 Passthrough 를 선택합니다.

5. **Route TLS 인증 정보 시크릿** 에서 드롭다운 메뉴를 클릭하고 목록에서 시크릿을 선택합니다.

4.2.3. 자동화 허브 Operator의 Ingress 유형 구성

Red Hat Ansible Automation Platform Operator 설치 양식을 사용하면 **고급 구성에서** 자동화 허브 Operator Ingress를 추가로 구성할 수 있습니다.

절차

1. **고급 구성을** 클릭합니다.
2. **Ingress 유형** 에서 드롭다운 메뉴를 클릭하고 **Ingress** 를 선택합니다.
3. **Ingress 주석** 에서 Ingress에 추가할 주석을 입력합니다.
4. **Ingress TLS 시크릿** 에서 드롭다운 메뉴를 클릭하고 목록에서 시크릿을 선택합니다.

자동화 허브 Operator를 구성한 후 양식 보기 하단에서 **만들기** 를 클릭합니다. Red Hat OpenShift Container Platform은 이제 Pod를 생성합니다. 이 작업은 몇 분 정도 걸릴 수 있습니다.

Workloads → **Pod** 로 이동하여 새로 생성된 인스턴스를 찾아 진행 상황을 확인할 수 있습니다.

검증

자동화 허브에서 Ansible Automation Platform Operator 설치에서 제공하는 다음 Operator Pod가 실행 중인지 확인합니다.

Operator 관리자 컨트롤러	자동화 컨트롤러	자동화 허브
3 Operator 각각에 대한 Operator 관리자 컨트롤러는 다음을 포함합니다. <ul style="list-style-type: none"> ● automation-controller-operator-controller-manager ● automation-hub-operator-controller-manager ● resource-operator-controller-manager 	자동화 컨트롤러를 배포한 후 이러한 Pod가 추가된 것을 확인할 수 있습니다. <ul style="list-style-type: none"> ● 컨트롤러 ● controller-postgres 	자동화 허브를 배포한 후 다음 Pod가 추가된 것을 확인할 수 있습니다. <ul style="list-style-type: none"> ● hub-api ● hub-content ● hub-postgres ● hub-redis ● hub-worker



참고

누락된 Pod는 풀 시크릿이 필요함을 나타낼 수 있습니다. 보호되는 이미지 레지스트리 또는 개인 이미지 레지스트리에는 pull secret이 필요합니다. 자세한 내용은 **이미지 풀 시크릿** 사용을 참조하십시오. **oc describe pod <pod-name>**을 실행하여 해당 Pod에 ImagePullBackOff 오류가 있는지 확인하여 이 문제를 추가로 진단할 수 있습니다.

4.3. 자동화 허브 사용자 인터페이스에 액세스

모든 Pod가 성공적으로 시작되면 자동화 허브 인터페이스에 액세스할 수 있습니다.

절차

1. **Networking** → **Routes** (경로) 로 이동합니다.
2. **위치** 아래에서 자동화 허브 인스턴스의 URL 을 클릭합니다.

자동화 허브 사용자 인터페이스가 시작되어 운영자 구성 프로세스 중에 지정된 관리자 인증 정보로 로그인할 수 있습니다.



참고

구성 중에 관리자 암호를 지정하지 않으면 자동으로 생성됩니다. 이 암호를 찾으려면 프로젝트로 이동하여 **Workloads** → **Secrets** 를 선택하고 controller-admin-password를 엽니다. 여기에서 암호를 복사하여 Automation Hub 암호 필드에 붙여넣을 수 있습니다.

4.4. RED HAT ANSIBLE AUTOMATION PLATFORM OPERATOR에서 자동화 허브를 위한 외부 데이터베이스 구성

외부 데이터베이스를 사용하여 Ansible Automation Platform 을 배포하려는 사용자의 경우 인스턴스 자격 증명 및 연결 정보를 사용하여 시크릿을 구성한 다음 **oc create** 명령을 사용하여 클러스터에 적용할 수 있습니다.

기본적으로 Red Hat Ansible Automation Platform Operator 는 Ansible Automation Platform 배포와 동일한 네임스페이스에서 관리되는 PostgreSQL Pod 를 자동으로 생성하고 구성합니다.

전용 리소스를 보장하기 위해 전용 노드를 사용하거나 백업, 업그레이드 또는 성능을 수동으로 관리하려는 경우 외부 데이터베이스를 대신 사용하도록 선택할 수 있습니다.



참고

데이터베이스 이름이 다른 경우 자동화 허브 및 자동화 컨트롤러에 동일한 외부 데이터베이스(PostgreSQL 인스턴스)를 사용할 수 있습니다. 즉, 단일 PostgreSQL 인스턴스 내에 이름이 다른 여러 개의 데이터베이스가 있을 수 있습니다.

다음 섹션에서는 Ansible Automation Platform Operator에서 자동화 허브에 대한 외부 데이터베이스를 구성하는 단계를 간략하게 설명합니다.

사전 요구 사항

외부 데이터베이스는 현재 Ansible Automation Platform 릴리스에서 지원하는 버전인 PostgreSQL 데이터베이스여야 합니다.



참고

Ansible Automation Platform 2.4에서는 PostgreSQL 13을 지원합니다.

절차

외부 postgres 인스턴스 자격 증명 및 연결 정보는 자동화 허브 사양에 설정되는 시크릿에 저장해야 합니다.

1. 아래 템플릿에 따라 **postgres_configuration_secret.yaml** 파일을 생성합니다.

apiVersion: v1

```

kind: Secret
metadata:
  name: external-postgres-configuration
  namespace: <target_namespace> ❶
stringData:
  host: "<external_ip_or_url_resolvable_by_the_cluster>" ❷
  port: "<external_port>" ❸
  database: "<desired_database_name>"
  username: "<username_to_connect_as>"
  password: "<password_to_connect_with>" ❹
  sslmode: "prefer" ❺
  type: "unmanaged"
type: Opaque

```

- ❶ 보안을 생성할 네임스페이스입니다. 배포하려는 네임스페이스와 동일해야 합니다.
- ❷ 데이터베이스 노드의 확인 가능한 호스트 이름입니다.
- ❸ 외부 포트는 기본적으로 **5432**로 설정됩니다.
- ❹ 변수 **암호** 값에는 배포, 백업 또는 복원 중에 문제가 발생하지 않도록 작은따옴표(' ') 또는 백슬래시(\)가 포함되지 않아야 합니다.
- ❺ 변수 **sslmode** 는 외부 데이터베이스에만 유효합니다. 허용되는 값은 **prefer, disable, allow, require, verify-ca, verify-full** 입니다.

2. **oc create** 명령을 사용하여 **external-postgres-configuration-secret.yml** 을 클러스터에 적용합니다.

```
$ oc create -f external-postgres-configuration-secret.yml
```

3. **AutomationHub** 사용자 정의 리소스 오브젝트를 생성할 때 사양의 보안을 지정합니다. 아래 예제를 따르십시오.

```

apiVersion: awx.ansible.com/v1beta1
kind: AutomationHub
metadata:
  name: hub-dev
spec:
  postgres_configuration_secret: external-postgres-configuration

```

4.5. PVC 검색 및 삭제

PVC(영구 볼륨 클레임)는 자동화 허브 및 자동화 컨트롤러 애플리케이션에서 사용하는 데이터를 저장하는 데 사용되는 스토리지 볼륨입니다. 이러한 PVC는 애플리케이션과 독립적이며 애플리케이션이 삭제된 경우에도 유지됩니다. 더 이상 PVC가 필요하지 않거나 다른 위치에서 백업한 경우 수동으로 삭제할 수 있습니다.

절차

1. 배포 네임스페이스의 기존 PVC를 나열합니다.

```
oc get pvc -n <namespace>
```

- 이전 배포 이름과 PVC 이름을 비교하여 이전 배포와 연결된 PVC를 확인합니다.
- 이전 PVC를 삭제합니다.

```
oc delete pvc -n <namespace> <pvc-name>
```

4.6. 추가 리소스

- OpenShift Container Platform에서 Operator를 실행하는 방법에 대한 자세한 내용은 [OpenShift Container Platform 제품 설명서](#)로 이동하여 OpenShift Container Platform에서 Operator 작업 가이드를 클릭합니다.

5 장. OPENSIFT CONTAINER PLATFORM CLI에서 ANSIBLE AUTOMATION PLATFORM OPERATOR 설치

oc 명령을 사용하여 OpenShift Container Platform CLI(명령줄 인터페이스)에서 Red Hat OpenShift Container Platform에 Ansible Automation Platform Operator를 설치하려면 다음 지침을 사용합니다.

5.1. 사전 요구 사항

- Operator 설치 권한이 있는 계정을 사용하여 Red Hat OpenShift Container Platform에 액세스할 수 있습니다.
- OpenShift Container Platform CLI **oc** 명령이 로컬 시스템에 설치되어 있습니다. 자세한 내용은 Red Hat [OpenShift Container Platform 제품 설명서](#)에서 [OpenShift CLI 설치](#) 설명서를 참조하십시오.

5.2. OPENSIFT CONTAINER PLATFORM CLI를 사용하여 OPERATOR에 네임스페이스 가입

네임스페이스를 Operator에 가입하려면 다음 절차를 사용하십시오.

절차

1. Operator에 대한 프로젝트를 생성합니다.

```
oc new-project ansible-automation-platform
```

2. **sub.yaml** 이라는 파일을 생성합니다.
3. 다음 YAML 코드를 **sub.yaml** 파일에 추가합니다.

```
---
apiVersion: v1
kind: Namespace
metadata:
  labels:
    openshift.io/cluster-monitoring: "true"
  name: ansible-automation-platform
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ansible-automation-platform-operator
  namespace: ansible-automation-platform
spec:
  targetNamespaces:
    - ansible-automation-platform
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ansible-automation-platform
  namespace: ansible-automation-platform
spec:
```

```
channel: 'stable-2.4'
installPlanApproval: Automatic
name: ansible-automation-platform-operator
source: redhat-operators
sourceNamespace: openshift-marketplace
---
apiVersion: automationcontroller.ansible.com/v1beta1
kind: AutomationController
metadata:
  name: example
  namespace: ansible-automation-platform
spec:
  replicas: 1
```

이 파일은 `ansible-automation-platform` 네임스페이스를 **ansible-automation-platform operator**에 서브스크립션하는 **ansible-automation-platform -platform** 이라는 **Subscription** 오브젝트를 생성합니다.

그런 다음 **ansible-automation-platform** 네임스페이스에서 **example** 이라는 **AutomationController** 오브젝트를 생성합니다.

자동화 컨트롤러 이름을 예에서 변경하려면 `sub.yaml` 의 **kind: AutomationController** 섹션에서 `name` 필드를 편집하고 `<automation_controller_name>` 을 사용하려는 이름으로 바꿉니다.

```
apiVersion: automationcontroller.ansible.com/v1beta1
kind: AutomationController
metadata:
  name: <automation_controller_name>
  namespace: ansible-automation-platform
```

4. **oc apply** 명령을 실행하여 `sub.yaml` 파일에 지정된 오브젝트를 생성합니다.

```
oc apply -f sub.yaml
```

네임스페이스가 **ansible-automation-platform-operator** Operator에 성공적으로 구독되었는지 확인하려면 **oc get subs** 명령을 실행합니다.

```
$ oc get subs -n ansible-automation-platform
```

Operator에 네임스페이스 가입에 대한 자세한 내용은 Red Hat OpenShift Container Platform Operator 가이드의 [CLI를 사용하여 OperatorHub에서 설치를 참조하십시오](#).

OpenShift Container Platform CLI를 사용하여 사용자가 생성한 Automation 컨트롤러의 웹 주소와 암호를 가져올 수 있습니다.

5.3. OPENSIFT CONTAINER PLATFORM CLI에서 자동화 컨트롤러 로그인 세부 정보 가져오기

자동화 컨트롤러에 로그인하려면 웹 주소와 암호가 필요합니다.

5.3.1. 자동화 컨트롤러 웹 주소 가져오기


```
app.kubernetes.io/name: example
app.kubernetes.io/operator-version: ""
app.kubernetes.io/part-of: example
name: example-admin-password
namespace: ansible-automation-platform
resourceVersion: "185185"
uid: 39393939-5252-4242-b929-665f665f665f
```

이 예에서 암호는 **88TG88TG88TG88TG88TG88TG88TG88TG** 88TG88TG88TG88TG88TG 입니다.

5.4. 추가 리소스

- OpenShift Container Platform에서 Operator를 실행하는 방법에 대한 자세한 내용은 [OpenShift Container Platform 제품 설명서](#)로 이동하여 OpenShift Container Platform에서 Operator 작업 가이드를 클릭합니다.

6 장. OPENSIFT CONTAINER PLATFORM 에서 ANSIBLE AUTOMATION PLATFORM OPERATOR 를 사용하여 이벤트 기반 ANSIBLE 컨트롤러 배포

이벤트 기반 Ansible 컨트롤러는 이벤트 중심 자동화를 위한 인터페이스이며 IT 요청을 자동으로 해결합니다. 이 구성 요소를 사용하면 이벤트 소스에 연결하고 룰북을 사용하여 해당 이벤트에 대한 작업을 수행할 수 있습니다. 이벤트 기반 Ansible 컨트롤러를 배포할 때 의사 결정을 자동화하고, 다양한 이벤트 소스를 사용하고, 여러 IT 활용 사례 내에서 이벤트 중심 자동화를 구현하고, 서비스 제공을 보다 효율적으로 수행할 수 있습니다.

다음 지침을 사용하여 OpenShift Container Platform 에서 Ansible Automation Platform Operator 와 이벤트 기반 Ansible 을 설치합니다.

사전 요구 사항

- OpenShift Container Platform 에 Ansible Automation Platform Operator 를 설치했습니다.
- 자동화 컨트롤러를 설치 및 구성했습니다.

절차

1. Operator 설치된 Operator 를 선택합니다.
2. Ansible Automation Platform 설치를 찾아 선택합니다.
3. 제공된 API 아래에서 이벤트 기반 Ansible 모달을 찾고 인스턴스 생성 을 클릭합니다. 그러면 폼 뷰로 이동하여 설치를 사용자 지정할 수 있습니다.
4. 이름 필드에 새 이벤트 기반 Ansible 컨트롤러 배포에 필요한 이름을 입력합니다.



중요

현재 OpenShift Container Platform 네임스페이스에 다른 Ansible Automation Platform 구성 요소가 설치되어 있는 경우 이벤트 기반 Ansible 사용자 정의 리소스를 생성할 때 이벤트 기반 Ansible 컨트롤러에 고유한 이름을 제공해야 합니다. 그렇지 않으면 이름 지정 충돌이 발생하여 이벤트 기반 Ansible 컨트롤러 배포에 영향을 미칠 수 있습니다.

5. 컨트롤러 URL 을 지정합니다.
Openshift 에 자동화 컨트롤러를 배포한 경우 **Networking** → **경로** 의 탐색 패널에서 URL 도 찾을 수 있습니다.



참고

이는 유일한 필수 사용자 정의이지만 필요한 경우 UI 양식을 사용하거나 YAML 구성 탭에서 직접 다른 옵션을 사용자 지정할 수 있습니다.

6. 생성을 클릭합니다. 이렇게 하면 지정한 네임스페이스에 이벤트 기반 Ansible 컨트롤러가 배포됩니다.
설치가 **Successful** 로 표시되면 Openshift UI 의 **Routes** 페이지에서 Event-Driven Ansible UI 의 URL 을 찾을 수 있습니다.
7. 탐색 패널에서 **Networking** → **Routes** (경로) 를 선택하여 생성된 새 경로 URL 을 찾습니다.

경로는 사용자 정의 리소스의 이름에 따라 나열됩니다.

8. 새 URL 을 클릭하여 브라우저에서 이벤트 기반 Ansible 으로 이동합니다.
9. 사용자 정의 설정을 지정하지 않는 한 탐색 패널에서 **Workloads** → **Secrets** 시크릿) 를 선택하고 사용자 지정 암호를 지정하지 않는 한 사용자를 위해 생성된 Admin Password k8s 시크릿을 찾습니다.
시크릿은 사용자 정의 리소스의 이름에 따라 나열되고 **-admin-password**가 추가됩니다.



참고

시크릿의 password 값을 사용하여 이벤트 기반 Ansible 컨트롤러 UI 에 로그인할 수 있습니다. 기본 사용자는 **admin** 입니다.

7장. 자동화 허브와 함께 RED HAT SINGLE SIGN-ON OPERATOR 사용

프라이빗 자동화 허브에서는 Red Hat Single Sign-On 을 사용하여 인증합니다.

Red Hat Single Sign-On Operator 는 리소스를 생성하고 관리합니다. Openshift 에서 Red Hat Single Sign-On 관리 자동화를 위해 이 Operator 를 사용하여 사용자 지정 리소스를 생성합니다.

- VM(가상 머신) 에 Ansible Automation Platform 을 설치할 때 설치 프로그램은 프라이빗 자동화 허브에서 사용할 수 있도록 Red Hat Single Sign-On 을 자동으로 설치하고 구성할 수 있습니다.
- Red Hat OpenShift Container Platform 에 Ansible Automation Platform 을 설치할 때 Single Sign-On 을 별도로 설치해야 합니다.

이 장에서는 Ansible Automation Platform 이 OpenShift Container Platform 에 설치될 때 Red Hat Single Sign-On 을 구성하고 프라이빗 자동화 허브와 통합하는 프로세스를 설명합니다.

사전 요구 사항

- Operator 설치 권한이 있는 계정을 사용하여 Red Hat OpenShift Container Platform 에 액세스할 수 있습니다.
- Red Hat Ansible Automation Platform Operator 가 포함된 카탈로그가 설치되어 있습니다.
- Red Hat Single Sign-On Operator 가 설치되어 있습니다. Red Hat Single Sign-On Operator 를 설치하려면 [Red Hat Single Sign-On 설명서의 사용자 정의 리소스를 사용하여 Red Hat Single Sign-On 설치 절차를 따르십시오.](#)

7.1. KEYCLOAK 인스턴스 생성

Red Hat Single Sign-On Operator 가 설치되면 Ansible Automation Platform 에서 사용할 Keycloak 인스턴스를 생성할 수 있습니다.

여기에서 외부 Postgres 를 제공하거나 사용자를 위해 생성됩니다.

절차

1. Operator → 설치된 Operator 로 이동합니다.
2. **rh-ssso** 프로젝트를 선택합니다.
3. **Red Hat Single Sign-On Operator** 를 선택합니다.
4. Red Hat Single Sign-On Operator 세부 정보 페이지에서 **Keycloak** 을 선택합니다.
5. **인스턴스 생성** 을 클릭합니다.
6. **YAML 보기** 를 클릭합니다.
기본 Keycloak 사용자 정의 리소스는 다음과 같습니다.

```
apiVersion: keycloak.org/v1alpha1
kind: Keycloak
metadata:
  name: example-keycloak
  labels:
```

```

app: sso
namespace: aap
spec:
  externalAccess:
  enabled: true
  instances: 1

```

7. 생성을 클릭합니다.
8. 배포가 완료되면 이 인증 정보를 사용하여 관리 콘솔에 로그인할 수 있습니다.
9. 네임스페이스의 **credential-*<custom-resource>***(example keycloak) 시크릿에서 관리자의 인증 정보를 찾을 수 있습니다.

7.2. ANSIBLE AUTOMATION PLATFORM 을 위한 KEYCLOAK 영역 생성

사용자, 자격 증명, 역할 및 그룹 세트를 관리하는 영역을 생성합니다. 사용자는 영역에 속하고 로그인합니다. 영역은 서로 격리되며 제어하는 사용자만 관리하고 인증할 수 있습니다.

절차

1. Operator → 설치된 Operator로 이동합니다.
2. Red Hat Single Sign-On Operator 프로젝트를 선택합니다.
3. Keycloak>-< 탭을 선택하고 **Create Keycloak**>-< 를 클릭합니다.
4. Keycloak>-& It; 양식에서 **YAML 보기** 를 선택합니다. 다음과 같이 YAML 파일을 편집합니다.

```

kind: KeycloakRealm
apiVersion: keycloak.org/v1alpha1
metadata:
  name: ansible-automation-platform-keycloakrealm
  namespace: rh-sso
  labels:
    app: sso
    realm: ansible-automation-platform
spec:
  realm:
    id: ansible-automation-platform
    realm: ansible-automation-platform
    enabled: true
    displayName: Ansible Automation Platform
instanceSelector:
  matchLabels:
    app: sso

```

필드	description
metadata.name	구성 리소스(CR)의 이름에 대해 metadata에 고유값을 설정합니다.

metadata.namespace	구성 리소스(CR)의 이름에 대해 metadata에 고유 값을 설정합니다.
metadata.labels.app	라벨을 고유 값으로 설정합니다. 이는 클라이언트 CR을 생성할 때 사용됩니다.
metadata.labels.realm	라벨을 고유 값으로 설정합니다. 이는 클라이언트 CR을 생성할 때 사용됩니다.
spec.realm.id	영역 이름 및 ID를 설정합니다. 이 둘은 동일해야 합니다.
spec.realm.realm	영역 이름 및 ID를 설정합니다. 이 둘은 동일해야 합니다.
spec.realm.displayname	표시할 이름을 설정합니다.

5. 생성을 클릭하고 프로세스가 완료될 때까지 기다립니다.

7.3. KEYCLOAK 클라이언트 생성

Keycloak 클라이언트는 Red Hat Single Sign-On을 사용하여 허브 사용자를 인증합니다. 사용자가 인증하면 요청이 Keycloak 클라이언트를 통과합니다. Single Sign-On이 **OAuth** 토큰을 검증하거나 발행하면 클라이언트는 자동화 허브에 대한 응답을 제공하고 사용자가 로그인할 수 있습니다.

절차

1. Operator → 설치된 Operator로 이동합니다.
2. Red Hat Single Sign-On Operator 프로젝트를 선택합니다.
3. Keycloak Client 탭을 선택하고 **Create Keycloak Client** 를 클릭합니다.
4. Keycloak>< 양식에서 **YAML 보기** 를 선택합니다.
5. 기본 YAML 파일을 다음으로 바꿉니다.

```
kind: KeycloakClient
apiVersion: keycloak.org/v1alpha1
metadata:
  name: automation-hub-client-secret
  labels:
    app: sso
    realm: ansible-automation-platform
  namespace: rh-sso
spec:
  realmSelector:
    matchLabels:
      app: sso
      realm: ansible-automation-platform
  client:
    name: Automation Hub
```

```

clientId: automation-hub
secret: <client-secret>
clientAuthenticatorType: client-secret
description: Client for automation hub
attributes:
  user.info.response.signature.alg: RS256
  request.object.signature.alg: RS256
directAccessGrantsEnabled: true
publicClient: true
protocol: openid-connect
standardFlowEnabled: true
protocolMappers:
  - config:
    access.token.claim: "true"
    claim.name: "family_name"
    id.token.claim: "true"
    jsonType.label: String
    user.attribute: lastName
    userinfo.token.claim: "true"
    consentRequired: false
    name: family name
    protocol: openid-connect
    protocolMapper: oidc-usermodel-property-mapper
  - config:
    userinfo.token.claim: "true"
    user.attribute: email
    id.token.claim: "true"
    access.token.claim: "true"
    claim.name: email
    jsonType.label: String
    name: email
    protocol: openid-connect
    protocolMapper: oidc-usermodel-property-mapper
    consentRequired: false
  - config:
    multivalued: "true"
    access.token.claim: "true"
    claim.name: "resource_access.${client_id}.roles"
    jsonType.label: String
    name: client roles
    protocol: openid-connect
    protocolMapper: oidc-usermodel-client-role-mapper
    consentRequired: false
  - config:
    userinfo.token.claim: "true"
    user.attribute: firstName
    id.token.claim: "true"
    access.token.claim: "true"
    claim.name: given_name
    jsonType.label: String
    name: given name
    protocol: openid-connect
    protocolMapper: oidc-usermodel-property-mapper
    consentRequired: false
  - config:
    id.token.claim: "true"

```

1

```

    access.token.claim: "true"
    userinfo.token.claim: "true"
    name: full name
    protocol: openid-connect
    protocolMapper: oidc-full-name-mapper
    consentRequired: false
  - config:
    userinfo.token.claim: "true"
    user.attribute: username
    id.token.claim: "true"
    access.token.claim: "true"
    claim.name: preferred_username
    jsonType.label: String
    name: <username>
    protocol: openid-connect
    protocolMapper: oidc-usermodel-property-mapper
    consentRequired: false
  - config:
    access.token.claim: "true"
    claim.name: "group"
    full.path: "true"
    id.token.claim: "true"
    userinfo.token.claim: "true"
    consentRequired: false
    name: group
    protocol: openid-connect
    protocolMapper: oidc-group-membership-mapper
  - config:
    multivalued: 'true'
    id.token.claim: 'true'
    access.token.claim: 'true'
    userinfo.token.claim: 'true'
    usermodel.clientRoleMapping.clientId: 'automation-hub'
    claim.name: client_roles
    jsonType.label: String
    name: client_roles
    protocolMapper: oidc-usermodel-client-role-mapper
    protocol: openid-connect
  - config:
    id.token.claim: "true"
    access.token.claim: "true"
    included.client.audience: 'automation-hub'
    protocol: openid-connect
    name: audience mapper
    protocolMapper: oidc-audience-mapper
roles:
  - name: "hubadmin"
    description: "An administrator role for automation hub"

```

1 이 값을 고유한 값으로 바꿉니다.

6. **생성**을 클릭하고 프로세스가 완료될 때까지 기다립니다.

자동화 허브가 배포되면 [Red Hat Single Sign-On](#) 클라이언트 업데이트에 설명된 대로 "Valid Redirect URI" 및 "Web Origins"를 사용하여 클라이언트를 업데이트해야 합니다. 그러나 인증 공급자가 Red Hat SSO에

그룹 데이터를 제공하지 않으면 그룹 매핑을 업데이트하여 정보가 전달되는 방식을 반영해야 합니다. 이는 일반적으로 user 속성에서 사용됩니다.

7.4. KEYCLOAK 사용자 생성

이 절차에서는 Super Administration 권한으로 자동화 허브에 로그인할 수 있는 **hubadmin** 역할을 사용하여 Keycloak 사용자를 생성합니다.

절차

1. Operator → 설치된 Operator로 이동합니다.
2. Red Hat Single Sign-On Operator 프로젝트를 선택합니다.
3. Keycloak>< 탭을 선택하고 **Create Keycloak User** 를 클릭합니다.
4. Keycloak User 양식에서 **YAML 보기** 를 선택합니다.
5. 기본 YAML 파일을 다음으로 바꿉니다.

```
apiVersion: keycloak.org/v1alpha1
kind: KeycloakUser
metadata:
  name: hubadmin-user
  labels:
    app: sso
    realm: ansible-automation-platform
  namespace: rh-sso
spec:
  realmSelector:
    matchLabels:
      app: sso
      realm: ansible-automation-platform
  user:
    username: hub_admin
    firstName: Hub
    lastName: Admin
    email: hub_admin@example.com
    enabled: true
    emailVerified: false
    credentials:
      - type: password
        value: <ch8ngeme>
    clientRoles:
      automation-hub:
        - hubadmin
```

6. 생성을 클릭하고 프로세스가 완료될 때까지 기다립니다.

사용자가 생성되면 Operator는 `credentials-<realm name>-<username>-<namespace>; credentials-<realm name>-<namespace>` 를 사용하여 사용자 이름과 암호를 모두 포함하는 보안을 생성합니다. 이 예제에서 인증 정보는 `credentials -ansible-automation-platform-hub-admin-rh-sso` 라고 합니다. 사용자가 생성되면 Operator에서 사용자 암호를 업데이트하지 않습니다. 암호 변경 사항은 시크릿에 반영되지 않습니다.

7.5. ANSIBLE AUTOMATION PLATFORM OPERATOR 설치

절차

1. **Operator Hub** 로 이동하여 *Ansible Automation Platform Operator* 를 검색합니다.
2. *Ansible Automation Platform Operator* 프로젝트를 선택합니다.
3. Operator 타일을 클릭합니다.
4. 설치를 클릭합니다.
5. Operator 를 설치할 프로젝트를 선택합니다. Operator 권장 네임스페이스 이름을 사용하는 것이 좋습니다.
 - a. 권장 항목이 아닌 프로젝트에 Operator 를 설치하려면 드롭다운 메뉴에서 **Create Project** 를 선택합니다.
 - b. 프로젝트 이름을 입력합니다.
 - c. 생성을 클릭합니다.
6. 설치를 클릭합니다.
7. Operator 가 설치되면 **Operator** 보기를 클릭합니다.

7.6. RED HAT SINGLE SIGN-ON 연결 시크릿 생성

Red Hat Single Sign-On 에 대한 연결 시크릿을 생성하려면 다음 절차를 사용하십시오.

절차

1. `https://<sso_host>/auth/realms/ansible-automation-platform` 로 이동합니다.
2. `public_key` 값을 복사합니다.
3. OpenShift 웹 UI 에서 **Workloads** → **Secrets** (시크릿) 로 이동합니다.
4. `ansible-automation-platform` 프로젝트를 선택합니다.
5. 생성을 클릭하고 **YAML** 에서 을 선택합니다.
6. 다음 YAML 을 편집하여 보안을 생성합니다.

```

apiVersion: v1
kind: Secret
metadata:
  name: automation-hub-sso
  namespace: ansible-automation-platform
type: Opaque
stringData:
  keycloak_host: "keycloak-rh-sso.apps-crc.testing"
  keycloak_port: "443"
  keycloak_protocol: "https"
  keycloak_realm: "ansible-automation-platform"
  keycloak_admin_role: "hubadmin"
  
```

1


```
social_auth_keycloak_key: "automation-hub"
social_auth_keycloak_secret: "client-secret" 2
social_auth_keycloak_public_key: >- 3
```

- 1 이 이름은 자동화 허브 인스턴스를 생성할 때 다음 단계에서 사용됩니다.
- 2 자동화 허브에 대한 Keycloak 클라이언트를 생성할 때 시크릿이 변경된 경우 이 값이 일치하도록 변경해야 합니다.
- 3 **Ansible Automation Platform Operator** 설치에 복사된 **public_key** 값을 입력합니다.

7. 생성을 클릭하고 프로세스가 완료될 때까지 기다립니다.

7.7. OPERATOR를 사용하여 자동화 허브 설치

다음 절차에 따라 Operator를 사용하여 자동화 허브를 설치합니다.

절차

1. Operator → 설치된 Operator로 이동합니다.
2. Ansible Automation Platform을 선택합니다.
3. Automation hub 탭을 선택하고 **Create Automation hub**를 클릭합니다.
4. **YAML** 보기를 선택합니다. YAML은 다음과 유사해야 합니다.

```
apiVersion: automationhub.ansible.com/v1beta1
kind: AutomationHub
metadata:
  name: private-ah 1
  namespace: ansible-automation-platform
spec:
  sso_secret: automation-hub-sso 2
  pulp_settings:
    verify_ssl: false
  route_tls_termination_mechanism: Edge
  ingress_type: Route
  loadbalancer_port: 80
  file_storage_size: 100Gi
  image_pull_policy: IfNotPresent
  web:
    replicas: 1
  file_storage_access_mode: ReadWriteMany
  content:
    log_level: INFO
    replicas: 2
  postgres_storage_requirements:
    limits:
      storage: 50Gi
    requests:
      storage: 8Gi
  api:
    log_level: INFO
```

```

replicas: 1
postgres_resource_requirements:
  limits:
    cpu: 1000m
    memory: 8Gi
  requests:
    cpu: 500m
    memory: 2Gi
loadbalancer_protocol: http
resource_manager:
  replicas: 1
worker:
  replicas: 2
    
```

- 1 인스턴스에 사용할 `metadata.name` 을 이름으로 설정합니다.
- 2 `spec.sso_secret` 을 [Red Hat Single Sign On 연결 세부 정보를 유지하기 위해 보안 생성에서 생성된 시크릿](#) 이름으로 설정합니다.



참고

이 YAML 은 SSL 확인을 비활성화합니다(**`ssl_verify: false`**). OpenShift 에 자체 서명된 인증서를 사용하지 않는 경우 이 설정을 제거할 수 있습니다.

5. 생성을 클릭하고 프로세스가 완료될 때까지 기다립니다.

7.8. 자동화 허브 경로 확인

다음 절차에 따라 허브 경로를 결정합니다.

절차

1. **Networking** → **Routes** (경로) 로 이동합니다.
2. 설치에 사용한 프로젝트를 선택합니다.
3. **private-ah-web-svc** 서비스의 위치를 복사합니다. 자동화 허브 인스턴스를 만들 때 다른 이름을 사용한 경우 서비스 이름이 다릅니다. 나중에 Red Hat Single Sign-On 클라이언트를 업데이트하는 데 사용됩니다.

7.9. RED HAT SINGLE SIGN-ON 클라이언트 업데이트

자동화 허브가 설치되고 인스턴스의 URL 을 알고 있는 경우 Valid Redirect URI 및 Web Origins 설정을 설정하려면 Red Hat Single Sign-On 을 업데이트해야 합니다.

절차

1. **Operator** → 설치된 **Operator** 로 이동합니다.
2. RH-SSO 프로젝트를 선택합니다.
3. **Red Hat Single Sign-On Operator** 를 클릭합니다.

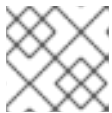
4. **Keycloak Client** 를 선택합니다.
5. `automation-hub-client-secret` 클라이언트를 클릭합니다.
6. **YAML** 을 선택합니다.
7. 클라이언트 YAML 을 업데이트하여 유효한 리디렉션 URI 및 Web Origins 설정을 추가합니다.

```

redirectUris:
- 'https://private-ah-ansible-automation-platform.apps-crc.testing/*'
webOrigins:
- 'https://private-ah-ansible-automation-platform.apps-crc.testing'

```

필드	description
redirectURIs	이는 Automation Hub 경로 확인 에서 결정된 위치입니다. redirectUris 설정 끝에 /*를 추가해야 합니다.
webOrigins	이는 Automation Hub 경로 확인 에서 결정된 위치입니다.



참고

이러한 설정을 입력할 때 들여쓰기가 올바른지 확인합니다.

8. **저장**을 클릭합니다.

연결 확인

1. 자동화 허브 경로로 이동합니다.
2. **hub_admin** 사용자 자격 증명을 입력하고 로그인합니다.
3. Red Hat Single Sign-On은 인증을 처리하고 자동화 허브로 다시 리디렉션합니다.

7.10. 추가 리소스

- OpenShift Container Platform에서 **Operator**를 실행하는 방법에 대한 자세한 내용은 [OpenShift Container Platform 제품 설명서의 OpenShift Container Platform에서 Operator 작업을 참조하십시오](#).

8 장. RED HAT ANSIBLE AUTOMATION PLATFORM 을 ANSIBLE AUTOMATION PLATFORM OPERATOR 로 마이그레이션

Red Hat Ansible Automation Platform 배포를 Ansible Automation Platform Operator로 마이그레이션하면 Red Hat Ansible Automation Platform 배포에 대한 업데이트 간소화 및 전체 라이프사이클 지원을 포함하여 Kubernetes 네이티브 Operator에서 제공하는 이점을 활용할 수 있습니다.

다음 절차를 사용하여 Ansible Automation Platform Operator로 다음 배포를 마이그레이션합니다.

- Ansible Tower 3.8.6, 자동화 컨트롤러 또는 자동화 허브의 VM 기반 설치
- Ansible Tower 3.8.6의 Openshift 인스턴스(Ansible Automation Platform 1.2)

8.1. 마이그레이션 고려 사항

OpenShift Container Platform 4의 OpenShift Container Platform 3에서 Ansible Automation Platform 1.2에서 Ansible Automation Platform 2.x로 업그레이드하는 경우 새 OpenShift Container Platform 버전 4 클러스터를 프로비저닝한 다음 Ansible Automation Platform을 새 클러스터로 마이그레이션해야 합니다.

8.2. 마이그레이션 준비

현재 Ansible Automation Platform 배포를 Ansible Automation Platform Operator로 마이그레이션하기 전에 기존 데이터를 백업하고 시크릿 키 및 postgresql 구성에 대한 k8s 시크릿을 생성해야 합니다.



참고

자동화 컨트롤러 및 자동화 허브 인스턴스를 모두 마이그레이션하는 경우 **시크릿 키 시크릿 생성** 단계를 반복하고 둘 다에 대한 **postgresql 구성 시크릿 생성** 단계를 반복한 다음 **Ansible Automation Platform Operator로 데이터를 마이그레이션** 합니다.

8.2.1. 사전 요구 사항

Ansible Automation Platform 배포를 Ansible Automation Platform Operator로 마이그레이션하려면 다음이 있어야 합니다.

- 보안 키 보안
- PostgreSQL 구성
- 새 OpenShift 클러스터의 네임스페이스에 대한 역할 기반 액세스 제어
- 새 OpenShift 클러스터는 이전 PostgreSQL 데이터베이스에 연결할 수 있어야 합니다.



참고

시크릿 키 정보는 초기 Red Hat Ansible Automation Platform 설치 중에 생성된 인벤토리 파일에 있을 수 있습니다. 시크릿 키를 기억하거나 인벤토리 파일을 찾는 데 문제가 없는 경우 Red Hat 고객 포털을 통해 **Ansible 지원에** 문의하십시오.

Ansible Automation Platform 2.x 또는 이전 버전에서 데이터를 마이그레이션하기 전에 손실 방지를 위해 데이터를 백업해야 합니다. 데이터를 백업하려면 다음을 수행합니다.

절차

1. 현재 배포 프로젝트에 로그인합니다.
2. `setup.sh`를 실행하여 현재 데이터/배포의 백업을 생성합니다.
버전 2.x 이하의 온-프레미스 배포의 경우:

```
┌ $ ./setup.sh -b
```

버전 2.0 이전의 OpenShift 배포의 경우 (operator 이외의 배포):

```
┌ ./setup_openshift.sh -b
```

8.2.2. 보안 키 시크릿 생성

OpenShift Container Platform의 Ansible Automation Platform Operator로 데이터를 마이그레이션하려면 초기 설치 중에 인벤토리 파일에 정의된 시크릿 키와 일치하는 시크릿 키를 생성해야 합니다. 그렇지 않으면 마이그레이션된 데이터는 마이그레이션 후에도 암호화되고 사용할 수 없습니다.

절차

1. 이전 설치에서 AAP를 배포하는 데 사용한 인벤토리 파일에서 이전 시크릿 키를 찾습니다.
2. 보안 키의 yml 파일을 생성합니다.

```
┌ apiVersion: v1
  kind: Secret
  metadata:
    name: <resourcename>-secret-key
    namespace: <target-namespace>
  stringData:
    secret_key: <old-secret-key>
  type: Opaque
```

3. 시크릿 키 yml을 클러스터에 적용합니다.

```
┌ oc apply -f <secret-key.yml>
```

8.2.3. postgresql 구성 시크릿 생성

마이그레이션이 성공하려면 기존 배포를 위해 데이터베이스에 대한 액세스 권한을 제공해야 합니다.

절차

1. postgresql 구성 시크릿의 yml 파일을 생성합니다.

```
┌ apiVersion: v1
  kind: Secret
  metadata:
    name: <resourcename>-old-postgres-configuration
    namespace: <target namespace>
  stringData:
    host: "<external ip or url resolvable by the cluster>"
    port: "<external port, this usually defaults to 5432>"
    database: "<desired database name>"
```

```
username: "<username to connect as>"
password: "<password to connect with>"
type: Opaque
```

2. postgresql 구성 yml 을 클러스터에 적용합니다.

```
oc apply -f <old-postgres-configuration.yml>
```

8.2.4. 네트워크 연결 확인

데이터를 성공적으로 마이그레이션하려면 새 Operator 배포에서 이전 배포 데이터베이스로 네트워크 연결이 있는지 확인합니다.

사전 요구 사항

기존 배포의 호스트 및 포트 정보를 기록해 둡니다. 이 정보는 conf.d 디렉터리에 있는 postgres.py 파일에 있습니다.

절차

1. yml 파일을 생성하여 새 배포 데이터베이스와 이전 배포 데이터베이스 간의 연결을 확인합니다.

```
apiVersion: v1
kind: Pod
metadata:
  name: dbchecker
spec:
  containers:
  - name: dbchecker
    image: registry.redhat.io/rhel8/postgresql-13:latest
    command: ["sleep"]
    args: ["600"]
```

2. 연결 검사기 yml 파일을 새 프로젝트 배포에 적용합니다.

```
oc project ansible-automation-platform
oc apply -f connection_checker.yml
```

3. 연결 검사기 Pod가 실행 중인지 확인합니다.

```
oc get pods
```

4. Pod 셸에 연결합니다.

```
oc rsh dbchecker
```

5. Pod에서 셸 세션을 열고 나면 새 프로젝트가 이전 프로젝트 클러스터에 연결할 수 있는지 확인합니다.

```
pg_isready -h <old-host-address> -p <old-port-number> -U awx
```

예제

`<old-host-address>:<old-port-number> - accepting connections`

8.3. ANSIBLE AUTOMATION PLATFORM OPERATOR로 데이터 마이그레이션

시크릿 키, postgresql 인증 정보, 네트워크 연결 확인 및 Ansible Automation Platform Operator를 설치한 후 데이터를 마이그레이션하기 전에 사용자 정의 리소스 컨트롤러 오브젝트를 생성해야 합니다.

8.3.1. AutomationController 오브젝트 생성

다음 단계를 사용하여 AutomationController 사용자 지정 리소스 오브젝트를 생성합니다.

절차

1. Red Hat OpenShift Container Platform에 로그인합니다.
2. Operators(운영자) InstalledOperators로 이동합니다.
3. 프로젝트 네임스페이스에 설치된 Ansible Automation Platform Operator를 선택합니다.
4. Automation Controller 탭을 선택합니다.
5. Create AutomationController를 클릭합니다.
6. 새 배포의 이름을 입력합니다.
7. 고급 구성에서 시크릿 키 시크릿 및 postgres 구성 시크릿을 선택합니다.
8. 생성을 클릭합니다.

8.3.2. AutomationHub 오브젝트 생성

다음 단계를 사용하여 AutomationHub 사용자 정의 리소스 오브젝트를 생성합니다.

절차

1. Red Hat OpenShift Container Platform에 로그인합니다.
2. Operators(운영자) InstalledOperators로 이동합니다.
3. 프로젝트 네임스페이스에 설치된 Ansible Automation Platform Operator를 선택합니다.
4. Automation Hub 탭을 선택합니다.
5. Create AutomationHub를 클릭합니다.
6. 새 배포의 이름을 입력합니다.
7. 고급 구성에서 시크릿 키 시크릿 및 postgres 구성 시크릿을 선택합니다.
8. 생성을 클릭합니다.

8.4. 마이그레이션 정리 후

데이터 마이그레이션이 완료되면 더 이상 필요하지 않은 인스턴스 그룹을 삭제해야 합니다.

절차

1. 마이그레이션 중에 생성한 암호를 사용하여 관리자로 Red Hat Ansible Automation Platform에 로그인합니다.



참고

참고: 마이그레이션 중에 관리자 암호를 생성하지 않으면 자동으로 생성된 암호가 생성됩니다. 이 암호를 찾으려면 프로젝트로 이동하여 **Workloads** → **Secrets** 를 선택하고 `controller-admin-password` 를 엽니다. 여기에서 암호를 복사하여 Red Hat Ansible Automation Platform 암호 필드에 붙여넣을 수 있습니다.

2. **Administration** → **InstanceGroups** 를 선택합니다.
3. `controlplane` 및 `default` 를 제외한 모든 `InstanceGroups` 를 선택합니다.
4. **삭제** 를 클릭합니다.

9장. OPENSIFT CONTAINER PLATFORM에서 ANSIBLE AUTOMATION PLATFORM OPERATOR 업그레이드

Ansible Automation Platform Operator는 OpenShift Container Platform 환경에서 새로운 Red Hat Ansible Automation Platform 인스턴스의 설치, 업그레이드 및 배포를 단순화합니다.

9.1. 업그레이드 고려 사항

Red Hat Ansible Automation Platform 버전 2.0은 Ansible Automation Platform Operator의 첫 번째 릴리스였습니다. 버전 2.0에서 업그레이드하는 경우 [Ansible Automation Platform Operator 업그레이드 프로시저](#)로 이동합니다.

업그레이드할 Red Hat Ansible Automation Platform 버전에서 지원하지 않는 OpenShift Container Platform 버전을 사용하는 경우 업그레이드하기 전에 OpenShift Container Platform 클러스터를 지원되는 버전으로 업그레이드해야 합니다.

필요한 OpenShift Container Platform 버전을 확인하려면 [Red Hat Ansible Automation Platform 라이프 사이클](#)을 참조하십시오.

클러스터 업그레이드에 대한 자세한 내용은 클러스터 업데이트를 [참조하십시오](#).

9.2. 사전 요구 사항

최신 버전의 Ansible Automation Platform Operator로 업그레이드하려면 다음을 수행하는 것이 좋습니다.

- AutomationControllerBackup 및 AutomationHubBackup 개체를 만듭니다.
- 업그레이드 중인 새로운 Ansible Automation Platform 버전과 중간 버전의 릴리스 노트를 검토하십시오.

9.3. ANSIBLE AUTOMATION PLATFORM OPERATOR 업그레이드

OpenShift Container Platform에서 Ansible Automation Platform Operator의 최신 버전으로 업그레이드하려면 다음을 수행하십시오.

Procedure

1. OpenShift Container Platform 4에 로그인합니다.
2. Operators(운영자) **InstalledOperators** 로 이동합니다.
3. **서브스크립션** 탭을 선택합니다.
4. **Upgrade status** 에서 **Upgrade Available** 을 클릭합니다.
5. **InstallPlan** 프리뷰 를 클릭합니다.
6. 승인을 클릭합니다.