Red Hat Ansible Automation Platform 2.4

컨테이너화된 Ansible Automation Platform 설치 가이드

컨테이너화된 Ansible Automation Platform 설치 가이드

Last Updated: 2024-06-11

Red Hat Ansible Automation Platform 2.4 컨테이너화된 Ansible Automation Platform 설치 가이드

컨테이너화된 Ansible Automation Platform 설치 가이드

법적 공지

초록

컨테이너화된 Ansible Automation Platform 설치 가이드

차례

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

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. ANSIBLE AUTOMATION PLATFORM 컨테이너화된 설치

Ansible Automation Platform은 Ansible 기반 환경에 제어, 지식, 위임을 추가하여 팀이 복잡한 다중 계층 배포를 관리하는 데 도움이 되는 상용 서비스입니다.

이 가이드에서는 새로운 Ansible Automation Platform 버전의 설치 요구 사항 및 프로세스를 이해하는 데 도움이 됩니다. 이 초기 버전은 Ansible Automation Platform 2.4를 기반으로 하며 기술 프리뷰로 릴리스되고 있습니다. 기술 프리뷰 기능 지원 범위를 참조하여 기술 프리뷰에 대한 자세한 내용을 확인하십시오.

사전 요구 사항

- RHEL 9.2 기반 호스트 최소 OS 기본 설치가 권장됩니다.
- sudo 또는 기타 Ansible 지원 권한 에스컬레이션(sudo)이 있는 RHEL 호스트의 루트가 아닌 사용자입니다. 이 사용자는 컨테이너화된 Ansible Automation Platform 설치를 담당합니다.
- 루트가 아닌 사용자에 대해 **SSH 공개 키 인증을** 설정하는 것이 좋습니다. 루트가 아닌 사용자에 대한 SSH 공개 키 인증 설정에 대한 지침은 암호 없는 로그인을 위한 SSH 공개 키 인증을 구성하는 방법을 참조하십시오.
- SSH 키는 원격 호스트에 설치하는 경우에만 필요합니다. 자체 포함된 로컬 VM 기반 설치를 수행하는 경우 SSH가 필요하지 않은 예에 따라 **ansible_connection: local** 을 사용할 수 있습니다.
- 기본 온라인 설치 방법을 사용하는 경우 RHEL 호스트에서 인터넷에 액세스할 수 있습니다.

1.1. 시스템 요구 사항

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

메모리	16GB RAM
CPU	4 CPU
디스크 공간	40Gb
디스크 IOPs	1500

1.2. 컨테이너화된 설치를 위한 RHEL 호스트 준비

프로세스

컨테이너화된 Ansible Automation Platform은 RHEL 호스트 상단에 있는 podman 기반 컨테이너로 구성 요소 서비스를 실행합니다. 기본 호스트가 준비되면 설치 프로그램에서 이 문제를 처리합니다. 이를 위해 다음 지침을 사용하십시오.

1. 루트가 아닌 사용자로 RHEL 호스트에 로그인합니다.
2. **dnf repolist** 를 실행하여 BaseOS 및 appstream 리포지토리만 호스트에서 설정 및 활성화되어 있는지 확인합니다.

```
$ dnf repolist
```

```
Updating Subscription Management repositories.
repo id                                 repo name
rhel-9-for-x86_64-appstream-rpms        Red Hat Enterprise Linux 9 for x86_64 -
AppStream (RPMs)
rhel-9-for-x86_64-baseos-rpms           Red Hat Enterprise Linux 9 for x86_64 -
BaseOS (RPMs)
```

3. 이러한 리포지토리와 이러한 리포지토리만 호스트 OS에서 사용할 수 있는지 확인합니다. 이 가이드를 사용하는 방법을 알아야 하는 경우 Chapter 10. 사용자 지정 소프트웨어 리포지토리 관리 Red Hat Enterprise Linux

4. 호스트에 DNS가 구성되어 있으며 FQDN(정규화된 도메인 이름)을 사용하여 호스트 이름과 IP를 확인할 수 있는지 확인합니다. 이는 서비스가 서로 통신할 수 있도록 하는 데 중요합니다.

바인딩되지 않은 DNS 사용

바인딩되지 않은 DNS를 구성하려면 Chapter 2를 참조하십시오. 바인딩되지 않은 DNS 서버 Red Hat Enterprise Linux 9 설정.

BIND DNS 사용

BIND를 사용하여 DNS를 구성하려면 Chapter 1을 참조하십시오. BIND DNS 서버 Red Hat Enterprise Linux 9 설정 및 구성.

선택 사항

설치 프로그램이 Ansible Automation Platform 서브스크립션 매니페스트 라이센스를 자동으로 선택하고 적용하도록 하려면 이 가이드를 사용하여 설치 프로그램에 대해 다운로드할 수 있는 매니페스트 파일을 생성합니다. Chapter 2. 매니페스트 파일 Red Hat Ansible Automation Platform 2. 가져오기.

1.3. ANSIBLE-CORE 설치

프로세스

1. ansible-core 및 기타 툴을 설치합니다.

```
sudo dnf install -y ansible-core wget git rsync
```

2. 정규화된 호스트 이름을 설정합니다.

```
sudo hostnamectl set-hostname your-FQDN-hostname
```

1.4. ANSIBLE AUTOMATION PLATFORM 다운로드

프로세스

1. access.redhat.com 에서 최신 설치 프로그램 tarball을 다운로드합니다. 이 작업은 RHEL 호스트 내에서 직접 수행할 수 있으므로 시간을 절약할 수 있습니다.

2. tarball 및 선택적 매니페스트 zip 파일을 랩탑에 다운로드한 경우 해당 파일을 RHEL 호스트에 복사합니다.

설치 프로그램이 파일 시스템에 상주할 위치를 결정합니다. 설치 관련 파일은 이 위치에 생성되며 초기 설치를 위해 최소 10Gb가 필요합니다.

3. 설치 프로그램 tarball을 설치 디렉터리에 압축 해제하고 cd를 압축 해제한 디렉터리에 압축을 풉니다.

   a. 온라인 설치 관리자

   ```
   $ tar xfvz ansible-automation-platform-containerized-setup-2.4-2.tar.gz
   ```

   b. 번들 설치

   ```
   $ tar xfvz ansible-automation-platform-containerized-setup-bundle-2.4-2-<arch
   name>.tar.gz
   ```

1.5. 컨테이너화된 ANSIBLE AUTOMATION PLATFORM의 사후 설치 기능 사용

컨테이너화된 Ansible Automation Platform의 실험적인 postinstaller 기능을 사용하여 초기 설치 중에 구성을 정의하고 로드합니다. 이 경우 간단한 YAML 파일로 로드할 구성을 정의하기 위한 구성-코드 접근 방식을 사용합니다.

1. 이 선택적 기능을 사용하려면 인벤토리 파일에서 다음 변수의 주석을 제거해야 합니다.

   ```
   controller_postinstall=true
   ```

2. 기본값은 false이므로 postinstaller를 활성화하려면 이를 활성화해야 합니다. 이 기능에 대한 Ansible Automation Platform 라이센스가 있어야 자동으로 로드할 수 있도록 로컬 파일 시스템에 있어야 합니다.

   ```
   controller_license_file=/full_path_to/manifest_file.zip
   ```

3. Git 기반 리포지토리에서 구성 as 코드를 가져올 수 있습니다. 이 작업을 수행하려면 다음 변수를 설정하여 콘텐츠를 가져오는 위치와 Ansible Automation Platform 컨트롤러에 업로드할 위치를 지정합니다.

   ```
   controller_postinstall_repo_url=https://your_cac_scm_repo
   controller_postinstall_dir=/full_path_to_where_you_want_the pulled_content_to_reside
   ```

4. controller_postinstall_repo_url 변수를 사용하여 인증 정보를 포함해야 하는 postinstall 리포지토리 URL을 정의할 수 있습니다.

   ```
   http(s)://<host>/<repo>.git (public repository without http(s) authentication)
   http(s)://<user>:<password>@<host>:<repo>.git (private repository with http(s)
   authentication)
   git@<host>:<repo>.git (public/private repository with ssh authentication)
   ```

참고

ssh 기반 인증을 사용하는 경우 설치 프로그램이 아무것도 구성하지 않으므로 설치 프로그램 노드에서 모든 항목을 구성해야 합니다.

정의 파일은 infra 인증 컬렉션을 사용합니다. controller_configuration 컬렉션은 설치의 일부로 사전 설치되어 있으며 인벤토리 파일에 제공한 설치 컨트롤러 인증 정보를 사용하여 Ansible Automation Platform 컨트롤러에 액세스합니다. YAML 구성 파일을 지정하기만 하면 됩니다.

인증 정보, LDAP 설정, 사용자 및 팀, 조직, 프로젝트, 인벤토리 및 호스트, 작업 및 워크플로우 템플릿과 같은 Ansible Automation Platform 구성 속성을 설정할 수 있습니다.

다음 예제에서는 컨트롤러 작업 템플릿을 정의하고 로드하는 샘플 **your-config.yml** 파일을 보여줍니다. 이 예제에서는 Ansible Automation Platform 설치와 함께 제공된 사전 로드된 데모 예제에 대한 간단한 변경 사항을 보여줍니다.

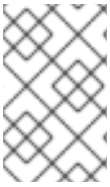
```
/full_path_to_your_configuration_as_code/
├── controller
    └── job_templates.yml
```

```
controller_templates:
 - name: Demo Job Template
   execution_environment: Default execution environment
   instance_groups:
  - default
   inventory: Demo Inventory
```

1.6. 컨테이너화된 ANSIBLE AUTOMATION PLATFORM 설치

Ansible Automation Platform은 인벤토리 파일을 사용하여 제어됩니다. 인벤토리 파일은 사용 및 생성된 호스트 및 구성 요소 변수, 설치를 사용자 지정하는 데 필요한 기타 정보를 정의합니다.

쉽게 시작할 수 있도록 복사 및 수정할 수 있는 예제 인벤토리 파일이 제공됩니다.



참고

인벤토리 파일에 지정된 기본 데이터베이스 선택이 없습니다. 인벤토리 파일의 지침에 따라 내부적으로 제공된 postgres를 적절히 선택하거나 외부에서 관리 및 지원되는 데이터베이스 옵션을 제공해야 합니다.

< > 자리 표시자를 특정 변수로 교체하고 요구 사항과 관련된 모든 행의 주석을 제거하여 인벤토리 파일을 편집합니다.

```
# This is the AAP installer inventory file
# Please consult the docs if you're unsure what to add
# For all optional variables please consult the included README.md

# This section is for your AAP Controller host(s)
# -----------------------------------------------
[automationcontroller]
fqdn_of_your_rhel_host ansible_connection=local

# This section is for your AAP Automation Hub host(s)
# -----------------------------------------------
[automationhub]
fqdn_of_your_rhel_host ansible_connection=local

# This section is for your AAP EDA Controller host(s)
```

```
# ----------------------------------------------------
[automationeda]
fqdn_of_your_rhel_host ansible_connection=local

# This section is for your AAP Execution host(s)
# -----------------------------------------------
#[execution_nodes]
#fqdn_of_your_rhel_host

# This section is for the AAP database(s)
# ---------------------------------------
# Uncomment the lines below and amend appropriately if you want AAP to install and manage the
postgres databases
# Leave commented out if you intend to use your own external database and just set appropriate
_pg_hosts vars
# see mandatory sections under each AAP component
#[database]
#fqdn_of_your_rhel_host ansible_connection=local

[all:vars]

# Common variables needed for installation
# ----------------------------------------
postgresql_admin_username=postgres
postgresql_admin_password=<set your own>
# If using the online (non-bundled) installer, you need to set RHN registry credentials
registry_username=<your RHN username>
registry_password=<your RHN password>
# If using the bundled installer, you need to alter defaults by using:
#bundle_install=true
# The bundle directory must include /bundle in the path
#bundle_dir=<full path to the bundle directory>
# To add more decision environment images you need to set the de_extra_images variable
#de_extra_images=[{'name': 'Custom decision environment', 'image':
'<registry>/<namespace>/<image>:<tag>'}]
# To add more execution environment images you need to set the ee_extra_images variable
#ee_extra_images=[{'name': 'Custom execution environment', 'image':
'<registry>/<namespace>/<image>:<tag>'}]
# To use custom TLS CA certificate/key you need to set these variables
#ca_tls_cert=<full path to your TLS CA certificate file>
#ca_tls_key=<full path to your TLS CA key file>

# AAP Database - optional
# -----------------------
# To use custom TLS certificate/key you need to set these variables
#postgresql_tls_cert=<full path to your TLS certificate file>
#postgresql_tls_key=<full path to your TLS key file>

# AAP Controller - mandatory
# --------------------------
controller_admin_password=<set your own>
controller_pg_host=fqdn_of_your_rhel_host
controller_pg_password=<set your own>

# AAP Controller - optional
# -------------------------
```

```
# To use the postinstall feature you need to set these variables
#controller_postinstall=true
#controller_license_file=<full path to your manifest .zip file>
#controller_postinstall_dir=<full path to your config-as-code directory>
# When using config-as-code in a git repository
#controller_postinstall_repo_url=<url to your config-as-code git repository>
#controller_postinstall_repo_ref=main
# To use custom TLS certificate/key you need to set these variables
#controller_tls_cert=<full path to your TLS certificate file>
#controller_tls_key=<full path to your TLS key file>

# AAP Automation Hub - mandatory
# -----------------------------
hub_admin_password=<set your own>
hub_pg_host=fqdn_of_your_rhel_host
hub_pg_password=<set your own>

# AAP Automation Hub - optional
# -----------------------------
# To use the postinstall feature you need to set these variables
#hub_postinstall=true
#hub_postinstall_dir=<full path to your config-as-code directory>
# When using config-as-code in a git repository
#hub_postinstall_repo_url=<url to your config-as-code git repository>
#hub_postinstall_repo_ref=main
# To customize the number of worker containers
#hub_workers=2
# To use the collection signing feature you need to set these variables
#hub_collection_signing=true
#hub_collection_signing_key=<full path to your gpg key file>
# To use the container signing feature you need to set these variables
#hub_container_signing=true
#hub_container_signing_key=<full path to your gpg key file>
# To use custom TLS certificate/key you need to set these variables
#hub_tls_cert=<full path to your TLS certificate file>
#hub_tls_key=<full path to your TLS key file>

# AAP EDA Controller - mandatory
# -----------------------------
eda_admin_password=<set your own>
eda_pg_host=fqdn_of_your_rhel_host
eda_pg_password=<set your own>

# AAP EDA Controller - optional
# -----------------------------
# When using an external controller node unmanaged by the installer.
#controller_main_url=https://fqdn_of_your_rhel_host
# To customize the number of default/activation worker containers
#eda_workers=2
#eda_activation_workers=2
# To use custom TLS certificate/key you need to set these variables
#eda_tls_cert=<full path to your TLS certificate file>
#eda_tls_key=<full path to your TLS key file>

# AAP Execution Nodes - optional
# ------------------------------
```

```
#receptor_port=27199
#receptor_protocol=tcp
# To use custom TLS certificate/key you need to set these variables
#receptor_tls_cert=<full path to your TLS certificate file>
#receptor_tls_key=<full path to your TLS key file>
# To use custom RSA key pair you need to set these variables
#receptor_signing_private_key=<full path to your RSA private key file>
#receptor_signing_public_key=<full path to your RSA public key file>
```

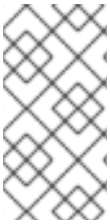
다음 명령을 사용하여 컨테이너화된 **Ansible Automation Platform**을 설치합니다.

```
ansible-playbook -i inventory ansible.containerized_installer.install
```

참고

If your privilege escalation requires a password to be entered, append *-K* to the command line. You will then be prompted for the *BECOME* password.

증가된 상세 정보 표시**(-vvvv)**를 사용하여 설치 프로세스의 세부 정보를 확인할 수 있습니다.



참고

이렇게 하면 설치 시간이 크게 증가할 수 있으므로 필요에 따라 사용하거나 **Red Hat** 지원에 의해 요청되는 경우에만 사용하는 것이 좋습니다.

1.7. 자동화 컨트롤러, 자동화 허브 및 이벤트 기반 ANSIBLE 컨트롤러에 액세스

설치가 완료되면 기본 프로토콜 및 사용되는 포트입니다.

- **HTTP/https** 프로토콜
- 자동화 컨트롤러용 포트 **8080/8443**
- 자동화 허브용 포트 **8081/8444**
- 이벤트 기반 **Ansible** 컨트롤러의 포트 **8082/8445**

이러한 내용은 변경될 수 있습니다. 자세한 내용은 **README.md** 를 참조하십시오. 포트 충돌 또는 기타 요인으로 인해 변경할 필요가 없는 한 기본값을 두는 것이 좋습니다.

자동화 컨트롤러 UI에 액세스

자동화 컨트롤러 **UI**는 기본적으로 다음에서 사용할 수 있습니다.

```
https://<your_rhel_host>:8443
```

controller_admin_password 용으로 생성한 암호를 사용하여 **admin** 사용자로 로그인합니다.

설치의 일부로 라이센스 매니페스트를 제공한 경우 **Ansible Automation Platform** 대시보드가 표시됩니다. 라이센스 파일을 제공하지 않은 경우 라이센스 세부 정보를 제공해야 하는 서브스크립션 화면이 표시됩니다. 이 문서는 여기로 문서화되어 있습니다: **Chapter 1. Red Hat Ansible Automation Platform 활성화**.

자동화 허브 UI에 액세스

자동화 허브 **UI**는 기본적으로 다음에서 사용할 수 있습니다.

```
https://<hub node>:8444
```

hub_admin_password 용으로 생성한 암호를 사용하여 **admin** 사용자로 로그인합니다.

이벤트 기반 Ansible UI에 액세스

이벤트 기반 **Ansible UI**는 기본적으로 다음에서 사용할 수 있습니다.

```
https://<eda node>:8445
```

eda_admin_password 에 대해 생성한 암호를 사용하여 **admin** 사용자로 로그인합니다.

1.8. 사용자 정의 TLS 인증서 사용

기본적으로 설치 프로그램은 사용자 정의 **CA**(인증 기관)에서 서명한 모든 서비스에 대한 **TLS** 인증서 및 키를 생성합니다. 각 서비스에 대해 사용자 정의 **TLS** 인증서/키를 제공할 수 있습니다. 해당 인증서를

사용자 정의 **CA**에서 서명한 경우 **CA TLS** 인증서 및 키를 제공해야 합니다.

- 인증 기관

```
ca_tls_cert=/full/path/to/tls/certificate
ca_tls_key=/full/path/to/tls/key
```

- **Automation Controller**

```
controller_tls_cert=/full/path/to/tls/certificate
controller_tls_key=/full/path/to/tls/key
```

- **Automation Hub**

```
hub_tls_cert=/full/path/to/tls/certificate
hub_tls_key=/full/path/to/tls/key
```

- **Automation EDA**

```
eda_tls_cert=/full/path/to/tls/certificate
eda_tls_key=/full/path/to/tls/key
```

- **Postgresql**

```
postgresql_tls_cert=/full/path/to/tls/certificate
postgresql_tls_key=/full/path/to/tls/key
```

- 수신기

```
receptor_tls_cert=/full/path/to/tls/certificate
receptor_tls_key=/full/path/to/tls/key
```

1.9. 사용자 정의 수신기 서명 키 사용

이제 **receiver _disable_signing=true** 가 설정되어 있고 설치 프로그램에서 **RSA** 키 쌍 **(public/private)**이 생성되지 않는 한 수신기 서명이 기본적으로 활성화됩니다. 그러나 경로 변수를 설정하여 사용자 지정 **RSA** 공개/개인 키를 제공할 수 있습니다.

-

```
receptor_signing_private_key=/full/path/to/private/key
receptor_signing_public_key=/full/path/to/public/key
```

1.10. 자동화 허브 컬렉션 및 컨테이너 서명 활성화

Automation Hub를 사용하면 **Ansible** 컬렉션 및 컨테이너 이미지에 서명할 수 있습니다. 이 기능은 기본적으로 활성화되어 있지 않으며 **GPG** 키를 제공해야 합니다.

```
hub_collection_signing=true
hub_collection_signing_key=/full/path/to/collections/gpg/key
hub_container_signing=true
hub_container_signing_key=/full/path/to/containers/gpg/key
```

GPG 키가 암호로 보호되는 경우 암호를 제공해야 합니다.

```
hub_collection_signing_pass=<collections gpg key passphrase>
hub_container_signing_pass=<containers gpg key passphrase>
```

1.11. 실행 노드 추가

컨테이너화된 설치 프로그램은 원격 실행 노드를 배포할 수 있습니다. 이 작업은 **ansible** 인벤토리 파일의 **execution_nodes** 그룹에 의해 처리됩니다.

```
[execution_nodes]
fqdn_of_your_execution_host
```

실행 노드는 기본적으로 포트 **27199(TCP)**에서 실행되는 실행 유형으로 구성됩니다. 이는 다음 변수로 변경할 수 있습니다.

- **receptor_port=27199**
- **receptor_protocol=tcp**
- **receptor_type=hop**

수신기 유형 값은 실행 또는 홉일 수 있지만 프로토콜은 **TCP** 또는 **UDP**입니다. 기본적으로 **execution_nodes** 그룹의 노드는 컨트롤러 노드의 피어로 추가됩니다. 그러나 **receiver _peers** 변수를

사용하여 피어 구성을 변경할 수 있습니다.

```
[execution_nodes]
fqdn_of_your_execution_host
fqdn_of_your_hop_host receptor_type=hop receptor_peers='["fqdn_of_your_execution_host"]'
```

1.12. 컨테이너화된 ANSIBLE AUTOMATION PLATFORM 설치 제거

컨테이너화된 배포를 설치 제거하려면 **uninstall.yml** 플레이북을 실행합니다.

```
$ ansible-playbook -i inventory ansible.containerized_installer.uninstall
```

이렇게 하면 모든 **systemd** 장치 및 컨테이너가 중지되고 다음과 같은 컨테이너화된 설치 프로그램에서 사용하는 모든 리소스가 삭제됩니다.

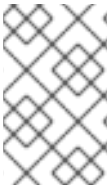
- 구성 및 데이터 디렉터리/파일
- **systemd** 장치 파일
- **Podman** 컨테이너 및 이미지
- **RPM** 패키지

컨테이너 이미지를 유지하려면 **container_keep_images** 변수를 **true**로 설정할 수 있습니다.

```
$ ansible-playbook -i inventory ansible.containerized_installer.uninstall -e
container_keep_images=true
```

postgresql 데이터베이스를 유지하려면 **postgresql_keep_databases** 변수를 **true**로 설정할 수 있습니다.

```
$ ansible-playbook -i </path/to/inventory> ansible.containerized_installer.uninstall -e
postgresql_keep_databases=true
```



참고

자동 생성된 항목이 아닌 동일한 **django secret** 키 값을 사용해야 합니다.