



# Red Hat Ansible Automation Platform 2.4

## Red Hat Ansible Automation Platform 計画ガイド

Ansible Automation Platform のインストールを計画する



# Red Hat Ansible Automation Platform 2.4 Red Hat Ansible Automation Platform 計画ガイド

---

Ansible Automation Platform のインストールを計画する

## 法律上の通知

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.

## 概要

このガイドでは、Red Hat Ansible Automation Platform をインストールするための要件、オプション、および推奨事項を説明します。

## 目次

はじめに .....	3
RED HAT ドキュメントへのフィードバック (英語のみ) .....	4
第1章 RED HAT ANSIBLE AUTOMATION PLATFORM インストールの計画 .....	5
第2章 RED HAT ANSIBLE AUTOMATION PLATFORM アーキテクチャー .....	6
2.1. ANSIBLE AUTOMATION PLATFORM アーキテクチャーの例 .....	6
第3章 RED HAT ANSIBLE AUTOMATION PLATFORM のコンポーネント .....	8
3.1. ANSIBLE AUTOMATION HUB .....	8
3.2. PRIVATE AUTOMATION HUB .....	8
3.3. 高可用性 AUTOMATION HUB .....	8
3.4. EVENT-DRIVEN ANSIBLE CONTROLLER .....	8
3.5. 自動化メッシュ .....	9
3.6. 自動化実行環境 .....	9
3.7. ANSIBLE GALAXY .....	9
3.8. 自動化コンテンツナビゲーター .....	9
第4章 システム要件 .....	10
4.1. RED HAT ANSIBLE AUTOMATION PLATFORM のシステム要件 .....	10
4.2. AUTOMATION CONTROLLER のシステム要件 .....	11
4.3. AUTOMATION HUB のシステム要件 .....	13
4.4. EVENT-DRIVEN ANSIBLE CONTROLLER のシステム要件 .....	15
4.5. POSTGRESQL の要件 .....	16
第5章 ネットワークポートおよびプロトコル .....	22
第6章 RED HAT ANSIBLE AUTOMATION PLATFORM サブスクリプションの割り当て .....	28
第7章 RED HAT ANSIBLE AUTOMATION PLATFORM インストーラーの選択および取得 .....	30
7.1. インターネットアクセスを使用したインストール .....	30
7.2. インターネットアクセスなしでのインストール .....	30
第8章 インストーラーインベントリーファイルについて .....	32
8.1. ホストとグループのガイドライン .....	33
8.2. ノードまたはグループのプロビジョニング解除 .....	34
8.3. インベントリー変数 .....	35
8.4. インベントリーファイルで変数を宣言するためのルール .....	35
8.5. インベントリーファイルでシークレットを保護する .....	36
8.6. 追加のインベントリーファイル変数 .....	37
第9章 サポート対象のインストールシナリオ .....	38
9.1. 同一ノード上にあるデータベースを持つスタンドアロン AUTOMATION CONTROLLER またはインストーラー以外が管理するデータベース .....	38
9.2. 外部管理データベースが設定されたスタンドアロン AUTOMATION CONTROLLER .....	38
9.3. 内部データベースを備えた単一の EVENT-DRIVEN ANSIBLE CONTROLLER ノード .....	38
9.4. 同じノード上にあるデータベースまたはインストーラー以外が管理するデータベースを使用するスタンドアロン AUTOMATION HUB .....	38
9.5. 外部管理データベースを使用するスタンドアロン AUTOMATION HUB .....	39
9.6. AUTOMATION CONTROLLER ノードまたはインストーラー以外が管理するデータベースを使用したプラットフォームインストール .....	39
9.7. 外部管理データベースを使用したプラットフォームのインストール .....	39
9.8. 外部管理データベースを使用した複数マシンのクラスターのインストール .....	39



## はじめに

Red Hat Ansible Automation Platform に興味をお持ちいただきありがとうございます。Ansible Automation Platform は、Ansible を装備した環境に、制御、ナレッジ、委譲の機能を追加して、チームが複雑かつ複数層のデプロイメントを管理できるように支援する商用サービスです。

このガイドの情報を使用して、Red Hat Ansible Automation Platform のインストールを計画してください。

## RED HAT ドキュメントへのフィードバック (英語のみ)

このドキュメントを改善するための提案がある場合、またはエラーを見つけた場合は、テクニカルサポート (<https://access.redhat.com>) に連絡し、**docs-product** コンポーネントを使用して Ansible Automation Platform Jira プロジェクトで Issue を作成してください。



## 第1章 RED HAT ANSIBLE AUTOMATION PLATFORM インストールの計画

Red Hat Ansible Automation Platform は、Red Hat Enterprise Linux と Red Hat OpenShift の両方でサポートされます。このガイドを使用して、Red Hat Enterprise Linux への Red Hat Ansible Automation Platform のインストールを計画してください。

Red Hat OpenShift Container Platform 環境に Red Hat Ansible Automation Platform をインストールするには、[Red Hat Ansible Automation Platform Operator を OpenShift Container Platform 上にデプロイする](#) を参照してください。

## 第2章 RED HAT ANSIBLE AUTOMATION PLATFORM アーキテクチャー

Ansible Automation Platform は、モジュールプラットフォームとして、コンポーネントの統合およびデプロイメントのカスタマイズを簡単に実行するための柔軟性を提供し、自動化要件が最大限一致するようにします。このセクションでは、Ansible Automation Platform デプロイメントの包括的なアーキテクチャー例を示します。

### 2.1. ANSIBLE AUTOMATION PLATFORM アーキテクチャーの例

Red Hat Ansible Automation Platform 2.4 リファレンスアーキテクチャーは、Red Hat Enterprise Linux で自動化メッシュを使用した Ansible Automation Platform の標準デプロイメントのセットアップ例を提供します。表示されているデプロイでは、次のコンポーネントを利用して、自動化ワークロードを処理するシンプルかつ安全で柔軟な方法、コンテンツコレクションの中央ロケーション、および IT 要求の自動解決を提供します。

#### Automation Controller

UI、Restful API、RBAC ワークフロー、および CI/CD 統合を介して自動化用のコントロールプレーンを提供します。

#### 自動化メッシュ

自動化メッシュは、既存ネットワークを使用して互いにピアツーピア接続を確立しているノードを介して、大規模かつ分散したワーカーのコレクション全体に作業を容易に分散できる機能を提供するオーバーレイネットワークです。

#### Private Automation Hub

Private Automation Hub は、自動化開発者が共同作業で独自の自動化コンテンツを公開し、組織内での Ansible コードの配信を合理化できる機能を提供します。

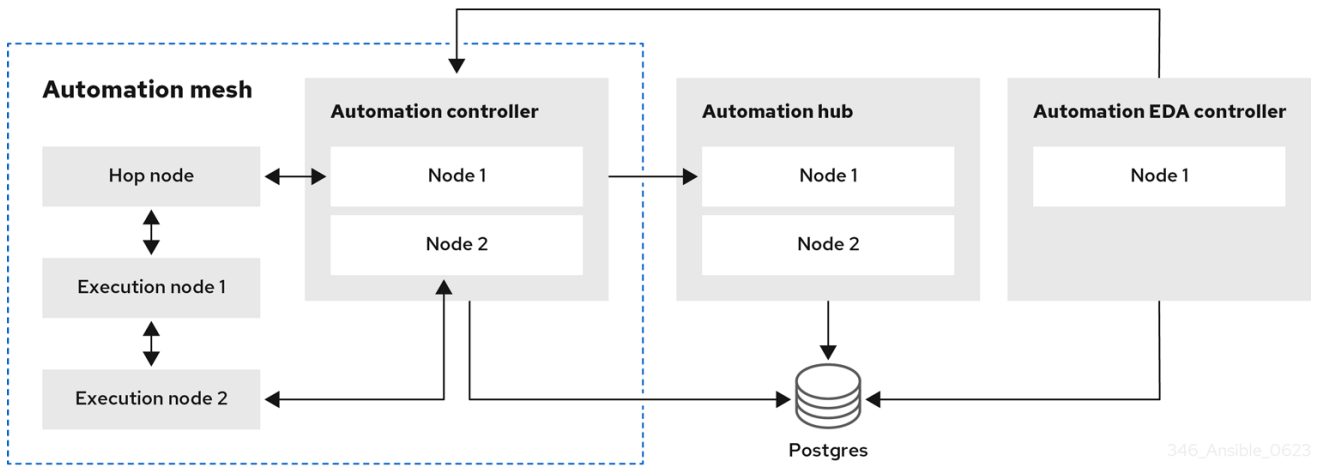
#### Event-Driven Ansible

時間のかかるタスクを自動化し、あらゆる IT ドメインの条件の変化に対応するために必要なイベント処理機能を提供します。

この例のアーキテクチャーは、以下で構成されます。

- 2 ノードの Automation Controller クラスタ
- Automation Controller を実行ノードに接続するためのオプションのホップノード
- 2 ノードの Automation Hub クラスタ
- シングルノードの Event-Driven Ansible Controller クラスタ
- Automation Controller、Automation Hub、および Event-Driven Ansible Controller クラスタに接続された単一の PostgreSQL データベース
- Automation Controller クラスタごとに 2 つの実行ノード

図2.1 Ansible Automation Platform 2.4 アーキテクチャーの例



## 第3章 RED HAT ANSIBLE AUTOMATION PLATFORM のコンポーネント

Ansible Automation Platform は、デプロイメントのニーズを満たすために相互に接続できる個別のコンポーネントで構成されるモジュラープラットフォームです。Ansible Automation Platform のデプロイメントは、ユーザーインターフェイス (UI) と RESTful アプリケーションプログラミングインターフェイス (API) を使用して Ansible 自動化を制御、保護、および管理するためのエンタープライズフレームワークである Automation Controller から始まります。続いて、以下の Automation Platform コンポーネントの任意の組み合わせをデプロイメントに追加できます。

### 3.1. ANSIBLE AUTOMATION HUB

Ansible Automation Hub は、Ansible コンテンツコレクションの認定済みコンテンツのリポジトリです。Ansible Automation Hub は、Red Hat とそのパートナーがコンテンツを公開し、お客様が認定済みでサポートされている Ansible Content Collection を発見するための一元化されたリポジトリです。Red Hat Ansible Certified Content は、Red Hat によってテストされ、サポートされているコンテンツをユーザーに提供します。

### 3.2. PRIVATE AUTOMATION HUB

Private Automation Hub は、コンテンツを同期するためのオフラインソリューションとオンプレミスソリューションの両方を提供します。Red Hat クラウド Automation Hub からコレクションと実行環境のイメージを同期し、独自のカスタム自動化コレクションと実行イメージを保存して提供できます。Ansible Galaxy や他のコンテナーレジストリーなどの他のソースを使用して、Private Automation Hub にコンテンツを提供することもできます。Private Automation Hub は、エンタープライズディレクトリーと CI/CD パイプラインに統合できます。

### 3.3. 高可用性 AUTOMATION HUB

高可用性 (HA) の設定により、Automation Hub デプロイメントの信頼性およびスケーラビリティが向上します。

Automation Hub の HA デプロイメントには、ワークロードを分散するロードバランサー (「アクティブ - アクティブ」設定) で同じサービスを同時に実行する複数のノードがあります。この設定により、サービスのダウンタイムを最小限に抑えるための単一障害点がなくなり、ワークロードの要件を満たすためのノードを簡単に追加または削除できます。

### 3.4. EVENT-DRIVEN ANSIBLE CONTROLLER

Event-Driven Ansible Controller は、イベント駆動型自動化のためのインターフェイスであり、IT リクエストの自動解決を導入するものです。Event-Driven Ansible Controller は、イベントのソースに接続し、ルールブックを使用してそれらのイベントに対処するのに役立ちます。このテクノロジーにより、IT の速度と俊敏性が向上し、一貫性と耐障害性が実現します。Event-Driven Ansible を使用すると、以下が可能になります。

- 意思決定を自動化する
- 多くのイベントソースを使用する
- 多くの IT ユースケース内で、また多くの IT ユースケースをまたいでイベント駆動型自動化を実装する

- [Event-Driven Ansible のスタートガイド](#)。

### 3.5. 自動化メッシュ

自動化メッシュは、既存ネットワークを使用して互いにピアツーピア接続を確立しているノードを介して、大規模な分散ワーカーのコレクション全体で作業分散を容易にするオーバーレイネットワークです。

自動化メッシュは以下を提供します。

- 個別にスケーリングする動的クラスター容量。これにより、ダウンタイムを最小限に抑えてノードを作成、登録、グループ化、グループ化解除、および登録解除できます。
- コントロールプレーンと実行プレーンの分離。コントロールプレーンの容量とは関係なく Playbook の実行容量をスケーリングできます。
- 遅延に対する回復力があり、停止することなく再設定可能であり、停止が存在する場合は動的に再ルーティングして別のパスを選択するデプロイメントの選択肢。
- メッシュルーティングの変更。
- FIPS (Federal Information Processing Standards) に準拠する双方向、マルチホップのメッシュ通信の可能性を含む接続性。

### 3.6. 自動化実行環境

自動化実行環境は、Red Hat Ansible Automation Platform のすべての自動化が実行されるコンテナイメージです。これらは、Ansible 実行エンジンと、ユーザーが IT 環境とプロセスのあらゆる側面を自動化するのに役立つ数百のモジュールを含むソリューションを提供します。自動化実行環境は、一般的に使用されるオペレーティングシステム、インフラストラクチャプラットフォーム、ネットワークデバイス、およびクラウドを自動化します。

### 3.7. ANSIBLE GALAXY

Ansible Galaxy は、Ansible コンテンツを検索、再利用、および共有するためのハブです。事前にパッケージ化されたロールの形式でコミュニティが提供する Galaxy コンテンツは、自動化プロジェクトの開始に役立ちます。インフラストラクチャーのプロビジョニング、アプリケーションのデプロイ、およびその他のタスクを完了するためのロールは、Ansible Playbook にドロップして、顧客の環境にすぐに適用できます。

### 3.8. 自動化コンテンツナビゲーター

自動化コンテンツナビゲーターは、自動化プラットフォームへの主要なコマンドラインインターフェイスとなる **テキストユーザーインターフェイス (TUI)** です。これは、コンテンツの構築、実行環境でのローカルでの自動化の実行、Ansible Automation Platform での自動化の実行、将来の **統合開発環境 (IDE)** の基盤の提供などのユースケースを扱います。

## 第4章 システム要件

この情報を使用して、Red Hat Ansible Automation Platform のインストールを計画し、ユースケースに適した自動化メッシュトポロジーを設計します。

### 前提条件

- **sudo** コマンドまたは権限昇格により、root アクセスを取得できる。権限昇格の詳細は、[Understanding privilege escalation](#) を参照してください。
- root から、AWX、PostgreSQL、Event-Driven Ansible、Pulp などのユーザーへ権限を降格できる。
- すべてのノードで NTP クライアントを設定している。詳細は、[Chrony を使用した NTP サーバーの設定](#) を参照してください。

### 4.1. RED HAT ANSIBLE AUTOMATION PLATFORM のシステム要件

お使いのシステムは、Red Hat Ansible Automation Platform をインストールして実行するために、以下の最小システム要件を満たしている必要があります。

表4.1 ベースシステム

要件	必須	備考
サブスクリプション	有効な Red Hat Ansible Automation Platform	
OS	Red Hat Enterprise Linux 8.8 以降 64 ビット (x86、ppc64le、s390x、aarch64)、または Red Hat Enterprise Linux 9.0 以降 64 ビット (x86、ppc64le、s390x、aarch64)	Red Hat Ansible Automation Platform は OpenShift でもサポートされています。詳細は、 <a href="#">Red Hat Ansible Automation Platform Operator を OpenShift Container Platform 上にデプロイする</a> を参照してください。
Ansible-core	Ansible-core バージョン 2.14 以降	Ansible Automation Platform には、ansible-core 2.15 を含む実行環境が含まれています。
Python	3.9 以降	
ブラウザ	Mozilla FireFox または Google Chrome の現行のサポートバージョン	
データベース	PostgreSQL バージョン 13	

プロジェクトの更新およびコレクションを使用するには、以下が必要です。

- ネットワークポートとプロトコル (表 5.3 Automation Hub に記載のもの) を使用して正常に接続し、Automation Hub または Ansible Galaxy サーバーからコレクションをダウンロードできることを確認します。
- 自己署名証明書または Red Hat ドメインを使用する場合に SSL インスペクションを無効にします。



### 注記

Ansible Automation Platform が管理するシステムの要件は Ansible と同じです。Ansible コミュニティドキュメントの [Installing Ansible](#) を参照してください。

## Red Hat Ansible Automation Platform 要件に関する注意点

- Red Hat Ansible Automation Platform は Ansible Playbook に依存しており、最新の安定したバージョンの `ansible-core` のインストールが必要です。`ansible-core` は手動でダウンロードすることも、Red Hat Ansible Automation Platform のインストール中に自動的にダウンロードすることもできます。
- 新規インストールの場合、Automation Controller は `ansible-core` の最新リリースパッケージをインストールします。
- バンドルの Ansible Automation Platform インストールを実行する場合、インストール `setup.sh` スクリプトにより、バンドルから `ansible-core` (およびその依存関係) のインストールが試行されます。
- Ansible を手動でインストールした場合、Ansible Automation Platform インストールの `setup.sh` スクリプトは、Ansible がインストールされていることを検出し、再インストールを試行しません。



### 注記

`dnf` などのパッケージマネージャーを使用して Ansible をインストールする必要があります。また、Red Hat Ansible Automation Platform が正常に動作するには、最新の安定したバージョンのパッケージマネージャーをインストールする必要があります。バージョン 2.4 以降には、Ansible バージョン 2.14 が必要です。

## 4.2. AUTOMATION CONTROLLER のシステム要件

Automation Controller は分散システムであり、このシステムでは、異なるソフトウェアコンポーネントを同じ場所に配置したり、複数のコンピュータードにデプロイしたりすることができます。インストーラーでは、ユースケースに適したトポロジーの設計に役立つ抽象化として、コントロールノード、ハイブリッドノード、実行ノード、ホップノードの4つのノードタイプが提供されます。

ノードのサイジングには、次の推奨事項を使用してください。

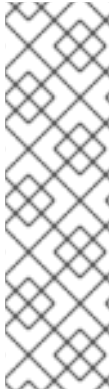


### 注記

コントロールノードとハイブリッドノードで、実行環境のストレージ用に、最小 20 GB を `/var/lib/awx` に割り当てます。

### 実行ノード

実行ノードは自動化を実行します。メモリーと CPU を増やし、フォークを多く実行できるように容量を増加します。



### 注記

- 記載されている RAM および CPU リソースは、実行ノードにインストールするパッケージには必要ない場合がありますが、ノードのジョブ負荷を処理して平均的な数のジョブを同時に実行するために推奨される最小リソースです。
- RAM および CPU ノードの推奨サイズはありません。必要な RAM または CPU は、その環境で実行しているジョブの数に直接依存します。

必要な RAM および CPU レベルの詳細は、[Automation Controller のパフォーマンスチューニング](#) を参照してください。

表4.2 実行ノード

要件	最小要件
RAM	16 GB
CPU	4
ローカルディスク	最小 40 GB

### コントロールノード

コントロールノードはイベントを処理し、プロジェクトの更新やクリーンアップジョブなどのクラスタージョブを実行します。CPU およびメモリーを増やすと、ジョブイベントの処理に役立ちます。

表4.3 コントロールノード

要件	最小要件
RAM	16 GB
CPU	4
ローカルディスク	<ul style="list-style-type: none"> <li>• 最小 40 GB。/var/lib/awx で少なくとも 20 GB が使用可能。</li> <li>• ストレージボリュームは、最小ベースライン 1500 IOPS で評価される必要があります。</li> <li>• プロジェクトは制御ノードとハイブリッドノードに保存され、ジョブの実行中は実行ノードにも保存されます。クラスターに大規模なプロジェクトが多数ある場合は、ディスク領域のエラーを回避するために、/var/lib/awx/projects に 2 倍の GB を追加することを検討してください。</li> </ul>



## ホップノード

ホップノードは、自動化メッシュの別の部分にトラフィックをルーティングする役割を果たします (たとえば、ホップノードは別のネットワークへの踏み台ホストにすることができます)。RAM はスループットに影響を与える可能性があり、CPU アクティビティは低くなります。一般に、ネットワーク帯域幅と遅延は、RAM や CPU よりも重要な要素です。

表4.4 ホップノード

要件	最小要件
RAM	16 GB
CPU	4
ローカルディスク	40 GB

- 実際の RAM 要件は、同時に管理するホストの Automation Controller の数により異なります (これはジョブテンプレートまたはシステムの **ansible.cfg** ファイルの **forks** パラメーターによって制御されます)。リソースの競合を回避するために、Ansible では、10 フォークあたり 1 GB のメモリーと、Automation Controller 用に 2 GB を予約することを推奨しています。詳細は、[Automation Controller の容量決定とジョブへの影響](#) を参照してください。forks が 400 に設定されている場合は、42 GB のメモリーが推奨されます。
- Automation Controller ホストは、**umask** が 0022 に設定されているかを確認します。設定されていない場合、セットアップが失敗します。このエラーを回避するには、**umask=0022** を設定します。
- より多くのホストにも対応できますが、フォーク数がホストの総数より少ない場合は、ホスト間でより多くのパスが必要になります。次のいずれかの方法を使用すると、このような RAM の制限を回避できます。
  - ローリング更新を使用します。
  - Automation Controller に組み込まれたプロビジョニングコールバックシステムを使用します。このシステムでは、設定を要求する各システムがキューに登録され、できるだけ早く処理されます。
  - Automation Controller が AMI などのイメージを作成またはデプロイしている場合。

## 関連情報

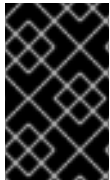
- Automation Controller サブスクリプションの取得の詳細は、[サブスクリプションのインポート](#) を参照してください。
- ご質問がある場合は、[Red Hat カスタマーポータル](#) から Ansible サポートにお問い合わせください。

## 4.3. AUTOMATION HUB のシステム要件

Automation Hub を使用すると、Red Hat Ansible および認定パートナーからの新しい認定自動化コンテンツを見つけ使用できます。Ansible Automation Hub では、クラウド自動化、ネットワーク自動化、セキュリティ自動化などのユースケースのために Red Hat とパートナーによって開発された、サポート対象自動化コンテンツである Ansible コレクションを検出して管理できます。

Automation Hub には、以下のシステム要件があります。

要件	必須	備考
RAM	最小 8 GB	<ul style="list-style-type: none"> <li>8 GB のメモリー (Vagrant 試用版のインストールに推奨される最小要件)</li> <li>8 GB メモリー (外部のスタンドアロン PostgreSQL データベースの最小要件)</li> <li>設定内のフォークに基づく容量については、<a href="#">Automation Controller の容量決定とジョブへの影響</a> を参照してください。</li> </ul>
CPU	最小 2 つ	設定内のフォークに基づく容量については、 <a href="#">Automation Controller の容量決定とジョブへの影響</a> を参照してください。
ローカルディスク	60 GB ディスク	少なくとも 40 GB をコレクションストレージ用に <code>/var</code> に割り当てます。



## 重要

Ansible Automation 実行ノードと Automation Hub のシステム要件は異なるため、ネットワークのニーズを満たさない可能性があります。必要なメモリー量を決定するための一般的な式は、 $\text{合計制御容量} = \text{合計メモリー (MB)} / \text{フォークサイズ (MB)}$  です。



## 注記

### Private Automation Hub

内部アドレスから Private Automation Hub をインストールし、外部アドレスしか記載されていない証明書を使用している場合は、インストールして証明書の問題がなくてもコンテナレジストリーとして使用できなくなる可能性があります。

これを回避するには、`automationhub_main_url` インベントリー変数を使用し、インストールインベントリーファイル内の Private Automation Hub ノードにリンクする値 (`https://pah.example.com` など) を指定します。

これにより、外部アドレスが `/etc/pulp/settings.py` に追加されます。これは、外部アドレスのみを使用することを意味します。

インベントリーファイル変数の詳細は、[Red Hat Ansible Automation Platform インストールガイド](#) の [インベントリーファイル変数](#) を参照してください。

### 4.3.1. 高可用性 Automation Hub の要件

高可用性 (HA) Automation Hub をデプロイする前に、環境に共有ファイルシステムがインストールされていること、および該当する場合はネットワークストレージシステムが設定されていることを確認してください。

#### 4.3.1.1. 必要な共有ファイルシステム

高可用性 Automation Hub では、お使いの環境に、NFS などの共有ファイルシステムを設定しておく必要があります。Red Hat Ansible Automation Platform インストーラーを実行する前に、共有ファイルシステムのインストールの一部としてクラスター全体に `/var/lib/pulp` ディレクトリーをインストールしたことを確認します。いずれかのノードで `/var/lib/pulp` が検出されないと、Red Hat Ansible Automation Platform インストーラーはエラーを返し、高可用性 Automation Hub のセットアップが失敗します。

`/var/lib/pulp` がいずれかのノードで検出されないことを示すエラーが表示された場合は、`/var/lib/pulp` がすべてのサーバーに正しくマウントされていることを確認し、インストーラーを再実行します。

#### 4.3.1.2. ネットワークストレージ用の firewalld のインストール

Automation Hub ノード自体にネットワークストレージを使用して HA Automation Hub をインストールする場合は、Ansible Automation Platform インストーラーを実行する前に、最初に `firewalld` をインストールして、共有ストレージシステムに必要なポートを開く必要があります。

以下のコマンドを実行して `firewalld` をインストールして設定します。

1. `firewalld` デーモンをインストールします。

```
$ dnf install firewalld
```

2. 以下のコマンドを使用して、<service> にネットワークストレージを追加します。

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



#### 注記

対応しているサービスの一覧は `$ firewall-cmd --get-services` コマンドを使用します。

3. リロードして設定を適用します。

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

## 4.4. EVENT-DRIVEN ANSIBLE CONTROLLER のシステム要件

Event-Driven Ansible コントローラーは、CPU コアの数に応じて、可変数の長時間実行プロセス (ルールブックアクティベーションなど) をオンデマンドで処理できるシングルノードシステムです。デフォルトで最大 12 個の同時アクティベーションを実行するには、次の最小要件を使用します。

要件	必須
RAM	16 GB

要件	必須
CPU	4
ローカルディスク	最小 40 GB



### 重要

- Red Hat Enterprise Linux 8 を実行していてメモリー制限を設定したい場合は、Event-Driven Ansible をインストールする前に cgroup v2 を有効にする必要があります。具体的な手順については、Knowledge-Centered Support (KCS) のソリューション記事 [Ansible Automation Platform Event-Driven Ansible controller for Red Hat Enterprise Linux 8 requires cgroupv2](#) を参照してください。
- 標準条件下で Event-Driven Ansible ルールブックをアクティブ化すると、約 250 MB のメモリーが使用されます。ただし、実際のメモリー消費量は、ルールの複雑さと処理されるイベントのボリュームおよびサイズによって大幅に異なる可能性があります。大量のイベントが予想される場合やルールブックの複雑さが高いシナリオでは、ステージング環境でのリソース使用量の事前評価を行います。これにより、アクティベーションの最大数はリソースの容量に基づいて行われます。Event-Driven Ansible コントローラーの最大実行アクティベーションの設定例については、[単一 Automation Controller](#)、[単一 Automation Hub](#)、および[外部 \(インストーラー管理\) データベースを備えた単一 Event-Driven Ansible Controller ノード](#) を参照してください。

## 4.5. POSTGRESQL の要件

Red Hat Ansible Automation Platform は PostgreSQL13 を使用します。PostgreSQL ユーザーパスワードは、データベースに保存する前に SCRAM-SHA-256 のセキュアハッシュアルゴリズムでハッシュ化されます。

Automation Controller インスタンスがデータベースにアクセスできるかどうかを確認するには、**awx-manage check\_db** コマンドを使用します。

表4.5 データベース

サービス	必須	備考
------	----	----

サービス	必須	備考
データベース	<ul style="list-style-type: none"> <li>● 20 GB の専用ハードディスクスペース</li> <li>● 4 CPU</li> <li>● 16 GB RAM</li> </ul>	<ul style="list-style-type: none"> <li>● 150 GB 以上を推奨</li> <li>● ストレージボリュームは、高いベースライン IOPS (1500 以上) に対応するように評価されている必要があります。</li> <li>● すべての Automation Controller データはデータベースに保存されます。データベースストレージは、マネージドホストの数、ジョブ実行数、ファクトキャッシュに保存されているファクトの数、および個別ジョブのタスク数と共に増加します。たとえば、ホスト 250 台で1時間ごと(1日に 24 回)に 20 個のタスクの Playbook を実行する場合は、毎週 800000 を超えるイベントを保存します。</li> <li>● データベースに十分な容量が確保されていない場合は、以前のジョブ実行やファクトを定期的に消去する必要があります。詳細は、<b>Automation Controller 管理ガイド</b>の<a href="#">管理ジョブ</a>を参照してください。</li> </ul>

## PostgreSQL の設定

必要に応じて、PostgreSQL データベースを、Red Hat Ansible Automation Platform インストーラーで管理されていない個別ノードとして設定できます。Ansible Automation Platform インストーラーがデータベースサーバーを管理する場合は、大半のワークロードで一般的に推奨されているデフォルト値を使用してサーバーを設定します。データベースのパフォーマンスを向上させるために使用できる設定の詳細は、[データベース設定](#)を参照してください。

### 関連情報

PostgreSQL サーバーのチューニングの詳細は、[PostgreSQL のドキュメント](#)を参照してください。

#### 4.5.1. 外部 (お客様がサポートする) データベースの設定



## 重要

Red Hat は外部 (お客様がサポートする) データベースの使用をサポートしていませんが、外部データベースはお客様によって使用されています。以下の初期設定に関するガイダンスは、関連するサポートリクエストを回避するために、製品インストールの観点からのみ提供されています。

Automation Controller で使用するために、外部の PostgreSQL 準拠データベースにデータベース、ユーザー、およびパスワードを作成するには、次の手順に従います。

## 手順

1. PostgreSQL 準拠のデータベースサーバーをインストールし、スーパーユーザー権限で接続します。

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

ここでは、以下のようになります。

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

サーバーが実行されているマシンのホスト名を指定します。値がスラッシュで始まる場合、その値は Unix ドメインソケットのディレクトリーとして使用されます。

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

接続するデータベースの名前を指定します。これは、コマンドラインで最初の非オプション引数として **dbname** を指定するのと同様です。**dbname** には接続文字列を指定できます。その場合、接続文字列パラメーターにより、競合するコマンドラインオプションがオーバーライドされます。

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

デフォルトではなく、ユーザー **username** としてデータベースに接続します。(これを行うには権限が必要です。)

2. ユーザーに割り当てられた **createDB** または管理者ロールを使用して、ユーザー、データベース、およびパスワードを作成します。詳細は、[Database Roles](#) を参照してください。
3. データベース認証情報とホストの詳細を、外部データベースとして Automation Controller インベントリーファイルに追加します。  
次の例ではデフォルト値が使用されています。

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

4. インストーラーを実行します。  
Automation Controller で PostgreSQL データベースを使用する場合、データベースは接続ユーザーが所有するものであり、**createDB** または管理者ロールがそのユーザーに割り当てられている必要があります。
5. 作成したデータベースにユーザー名、パスワード、データベース名で接続できることを確認します。
6. ユーザーの権限を確認します。ユーザーには **createDB** または管理者ロールが必要です。



### 注記

この手順の実行中、外部データベースの範囲を確認する必要があります。詳細は、<https://access.redhat.com/articles/4010491> を参照してください。

## 4.5.2. Automation Hub PostgreSQL データベースの hstore 拡張機能の有効化

Ansible Automation Platform 2.4 以降、データベース移行スクリプトは **hstore** フィールドを使用して情報を保存するため、Automation Hub PostgreSQL データベースの **hstore** 拡張機能を有効にする必要があります。

Ansible Automation Platform インストーラーとマネージド PostgreSQL サーバーを使用する場合、このプロセスは自動的に行われます。

PostgreSQL データベースが外部にある場合は、Automation Hub をインストールする前に、Automation Hub PostgreSQL データベースの **hstore** 拡張機能を手動で有効にする必要があります。

Automation Hub のインストール前に **hstore** 拡張機能が有効になっていない場合は、データベースの移行中にエラーが発生します。

### 手順

1. 拡張機能が PostgreSQL サーバー (Automation Hub データベース) で利用できるかどうかを確認します。

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

<automation hub database> のデフォルト値は **automationhub** です。

**hstore** が利用できる場合の出力例:

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

**hstore** が利用できない場合の出力例:

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

- RHEL ベースのサーバーでは、**hstore** 拡張機能は **postgresql-contrib** RPM パッケージに含まれていますが、PostgreSQL サーバー RPM パッケージのインストール時に自動的にインストールされません。  
RPM パッケージをインストールするには、次のコマンドを使用します。

```
dnf install postgresql-contrib
```

- 次のコマンドを使用して、Automation Hub データベースに **hstore** PostgreSQL 拡張機能を作成します。

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

その出力は次のとおりです。

```
CREATE EXTENSION
```

- 次の出力では、使用されている **hstore** 拡張子が **installed\_version** フィールドに含まれており、**hstore** が有効であることを示しています。

```
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 データベースのストレージパフォーマンスのベンチマーク

Flexible I/O Tester (FIO) ツールを使用して、Ansible Automation Platform PostgreSQL データベースの最小要件が満たされているかどうかを確認します。FIO は、ストレージシステムの読み取りおよび書き込み IOPS パフォーマンスをベンチマークするために使用されるツールです。

#### 前提条件

- Flexible I/O Tester (**fio**) ストレージパフォーマンスベンチマークツールがインストールされている。  
**fio** をインストールするには、root ユーザーとして次のコマンドを実行します。

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

- fio** テストデータログファイルを保存するのに十分なディスク容量がある。  
この手順に示す例では、**/tmp** ディレクトリーに少なくとも 60 GB のディスク領域が必要です。
  - numjobs** は、コマンドによって実行されるジョブの数を設定します。
  - size=10G** は、各ジョブによって生成されるファイルサイズを設定します。
- size** パラメーターの値を調整済みである。この値を調整すると、テストデータの量が減ります。

#### 手順

- ランダムな書き込みテストを実行します。



```
$ 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. ランダムな読み取りテストを実行します。

```
$ 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. 結果を確認します。

ベンチマークコマンドによって書き込まれたログファイルで、**iops** で始まる行を検索します。この行は、テストの最小値、最大値、および平均値を表示します。

次の例は、ランダム読み取りテストのログファイル内の行を表示しています。

```
$ 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  
[...]
```

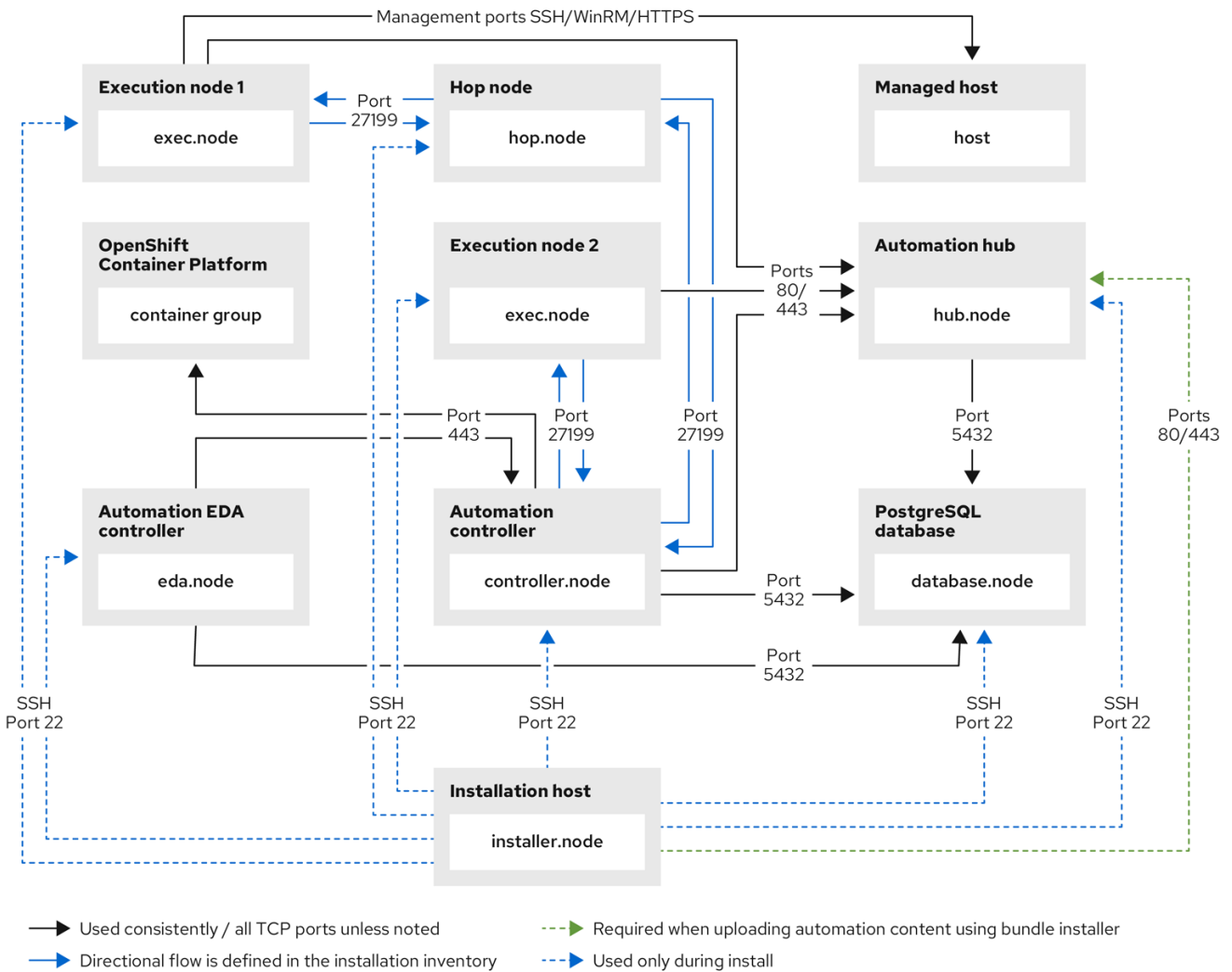
独自のビジネス要件、アプリケーションのワークロード、および新しい要求に応じて、ログファイルを確認、監視、再検討する必要があります。

## 第5章 ネットワークポートおよびプロトコル

Red Hat Ansible Automation Platform は、サービスとの通信に複数のポートを使用します。Red Hat Ansible Automation Platform サーバーへの着信接続を有効にするには、これらのポートを開いて利用できるようにする必要があります。これらのポートが利用可能で、サーバーのファイアウォールでブロックされていないことを確認してください。

以下のアーキテクチャー図は、すべての可能なコンポーネントと共に完全にデプロイされた Ansible Automation Platform の例です。

図5.1 Ansible Automation Platform のネットワークポートおよびプロトコル



637\_Ansible\_0424

以下の表は、宛先ポートとネットワークトラフィックの方向を示しています。



### 注記

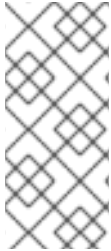
以下に示すデフォルトの宛先ポートとインストーラインベントリは設定可能です。使用している環境に合わせて設定すると、動作が変わる可能性があります。

表5.1 ネットワークポートおよびプロトコル

ポート	プロトコル	サービス	Source	宛先	用途	インストーラーのインベントリー変数
22	TCP	SSH	インストーラーノード	Automation Hub	インストール (一時的)	<b>ansible_port</b>
22	TCP	SSH	インストーラーノード	controller_node	インストール (一時的)	<b>ansible_port</b>
22	TCP	SSH	インストーラーノード	EDA ノード	インストール (一時的)	<b>ansible_port</b>
22	TCP	SSH	インストーラーノード	実行ノード	インストール (一時的)	<b>ansible_port</b>
22	TCP	SSH	インストーラーノード	ホップノード	インストール (一時的)	<b>ansible_port</b>
22	TCP	SSH	インストーラーノード	ハイブリッドノード	インストール (一時的)	<b>ansible_port</b>
22	TCP	SSH	インストーラーノード	PostgreSQL データベース	インストール中のリモートアクセス (一時的)	<b>pg_port</b>
80/443	TCP	HTTP/HTTPS	インストーラーノード	Automation Hub	バンドルインストーラーを使用するときに、インストーラーノードが実行環境イメージを Automation Hub にプッシュできるようにします。	固定値
80/443	TCP	HTTP/HTTPS	実行ノード	Automation Hub	実行ノードが Automation Hub から実行環境イメージをプルできるようにします。	固定値
443	TCP	HTTPS	controller_node	クライアント	Web UI/API	<b>nginx_https_port</b>
443	TCP	HTTPS	controller_node	OpenShift Container Platform	コンテナグループを使用してジョブを実行する場合にのみ必要です。	OpenShift API サーバーのホスト名

ポート	プロトコル	サービス	Source	宛先	用途	インストーラーのインベントリー変数
5432	TCP	PostgreSQL	controller_node	PostgreSQL データベース	内部データベースが別のコンポーネントとともに使用されている場合にのみ開放されます。そうでない場合は、このポートを開放しないでください。	<b>automationcontroller_pg_port</b>
5432	TCP	PostgreSQL	EDA ノード	PostgreSQL データベース	内部データベースが別のコンポーネントとともに使用されている場合にのみ開放されます。そうでない場合は、このポートを開放しないでください。	<b>automationedaccontroller_pg_port</b>
5432	TCP	PostgreSQL	Automation Hub	PostgreSQL データベース	内部データベースが別のコンポーネントとともに使用されている場合にのみ開放されます。そうでない場合は、このポートを開放しないでください。	<b>automationhub_pg_port</b>
27199	TCP	Receptor	controller_node	実行ノード	<p>設定可能</p> <p>メッシュノードは、コントローラーに直接ピア接続されます。</p> <p>関係するダイレクトノード。27199 通信は、(インベントリーに応じて) 実行ノードに対して双方向にすることができます</p>	<b>receptor_listener_peers</b>

ポート	プロトコル	サービス	Source	宛先	用途	インストーラーのインベントリー変数
27199	TCP	Receptor	controller_node	ホップノード	設定可能  ホップノードを介してリレーされる場合、ホップノードから Receptor ポートへの接続を有効にします。	<b>receptor_listen r_port</b>  <b>peers</b>
27199	TCP	Receptor	controller_node	ハイブリッドノード	設定可能  ホップ接続されていないノードを介してリレーされる場合、コントローラーから Receptor ポートへの接続を有効にします。	<b>receptor_listen r_port</b>  <b>peers</b>
27199	TCP	Receptor	実行ノード	ホップノード	設定可能  メッシュ 27199 通信は、(インストールインベントリーに応じて) 実行ノードに対して双方向にすることができます  コントローラーから Receptor ポートへの接続を許可します。	<b>receptor_listen r_port</b>  <b>peers</b>
27199	TCP	Receptor	実行ノード	controller_node	設定可能  メッシュ 27199 通信は、(インストールインベントリーに応じて) 実行ノードに対して双方向にすることができます  コントローラーから Receptor ポートへの接続を許可します。	<b>receptor_listen r_port</b>  <b>peers</b>



## 注記

- ハイブリッドノードは、制御ノードと実行ノードの組み合わせとして機能するため、ハイブリッドノードは両方の接続を共有します。
- **receptor\_listener\_port** が定義されている場合、マシンには、受信 TCP 接続を確立するために使用可能なオープンポート (27199 など) も必要です。

表5.2 Red Hat Insights for Red Hat Ansible Automation Platform

URL	用途
<a href="https://api.access.redhat.com:443">https://api.access.redhat.com:443</a>	一般的なアカウントサービス、サブスクリプション
<a href="https://cert-api.access.redhat.com:443">https://cert-api.access.redhat.com:443</a>	Insights データのアップロード
<a href="https://cert.console.redhat.com:443">https://cert.console.redhat.com:443</a>	インベントリーのアップロードおよびクラウドコネクター接続
<a href="https://console.redhat.com:443">https://console.redhat.com:443</a>	Insights ダッシュボードへのアクセス

表5.3 Automation Hub

URL	用途
<a href="https://console.redhat.com:443">https://console.redhat.com:443</a>	一般的なアカウントサービス、サブスクリプション
<a href="https://catalog.redhat.com:443">https://catalog.redhat.com:443</a>	実行環境のインデックス作成
<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>	ファイアウォールアクセス
<a href="https://galaxy.ansible.com:443">https://galaxy.ansible.com:443</a>	Ansible コミュニティーがキュレートされた Ansible コンテンツ
<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>	コミュニティがキュレーションした Ansible コンテンツリポジトリ用のデュアルスタック IPv6 エンドポイント
<a href="https://registry.redhat.io:443">https://registry.redhat.io:443</a>	Red Hat およびパートナーが提供するコンテナイメージへのアクセス
<a href="https://cert.console.redhat.com:443">https://cert.console.redhat.com:443</a>	Red Hat およびパートナーキュレートされた Ansible コレクション

表5.4 実行環境 (EE)

URL	用途
<a href="https://registry.redhat.io:443">https://registry.redhat.io:443</a>	Red Hat およびパートナーが提供するコンテナイメージへのアクセス
<b>cdn.quay.io:443</b>	Red Hat およびパートナーが提供するコンテナイメージへのアクセス
<b>cdn01.quay.io:443</b>	Red Hat およびパートナーが提供するコンテナイメージへのアクセス
<b>cdn02.quay.io:443</b>	Red Hat およびパートナーが提供するコンテナイメージへのアクセス
<b>cdn03.quay.io:443</b>	Red Hat およびパートナーが提供するコンテナイメージへのアクセス

### 重要

イメージマニフェストとファイルシステム Blob は、**registry.redhat.io** から直接提供されます。ただし、2023年5月1日以降、ファイルシステム Blob は代わりに **quay.io** から提供されます。コンテナイメージのプルに関する問題を回避するには、一覧表示された **quay.io** ホスト名への送信接続を有効にする必要があります。

この変更を、**registry.redhat.io** へのアウトバウンド接続を有効にするすべてのファイアウォール設定に変更を加えます。

ファイアウォールルールを設定するときは、IP アドレスの代わりにホスト名を使用します。

この変更を加えた後、引き続き **registry.redhat.io** からイメージをプルできます。Red Hat コンテナイメージのプルを続行するために、**quay.io** にログインする必要も、**quay.io** レジストリーと直接やりとりする必要もありません。

詳細は、[Firewall changes for container image pulls](#) を参照してください。

## 第6章 RED HAT ANSIBLE AUTOMATION PLATFORM サブスクリプションの割り当て

Red Hat Ansible Automation Platform をインストールする前に、全ノードに有効なサブスクリプションが割り当てられている **必要があります**。Ansible Automation Platform サブスクリプションを割り当てると、インストールを続行するために必要なサブスクリプション専用のリソースにアクセスできるようになります。



### 注記

Red Hat アカウントで Simple Content Access Mode を有効にしている場合は、サブスクリプションを割り当てる必要はありません。有効にした場合は、Ansible Automation Platform をインストールする前にシステムを Red Hat Subscription Management (RHSM) または Satellite に登録する必要があります。詳細は、[Simple Content Access](#) を参照してください。

### 手順

1. Red Hat Ansible Automation Platform サブスクリプションの **pool\_id** を取得します。

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



### 注記

Ansible Automation Platform サブスクリプションのアタッチが失敗する可能性があるため、MCT4022 をサブスクリプションの **pool\_id** として使用しないでください。

### 例

**subscription-manager list** コマンドの出力例。 **Pool ID:** セクションの説明に従って **pool\_id** を取得します。

```
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. サブスクリプションを割り当てます。

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

これで、Red Hat Ansible Automation Platform サブスクリプションがすべてのノードに割り当てられました。

### 検証



- サブスクリプションが正常に割り当てられたことを確認します。

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

### トラブルシューティング

- Ansible Automation Platform インストーラーにバンドルされている特定のパッケージが見つからない場合や、**Repositories disabled by configuration** というメッセージが表示される場合は、次のコマンドを使用してリポジトリを有効にしてみてください。

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
```

## 第7章 RED HAT ANSIBLE AUTOMATION PLATFORM インストーラーの選択および取得

Red Hat Enterprise Linux 環境のインターネット接続に基づいて、必要な Ansible Automation Platform インストーラーを選択します。以下のシナリオを確認し、ニーズを満たす Red Hat Ansible Automation Platform インストーラーを決定してください。

### 7.1. インターネットアクセスを使用したインストール

Red Hat Enterprise Linux 環境がインターネットに接続している場合は、Ansible Automation Platform インストーラーを選択します。インターネットアクセスを使用してインストールすると、必要な最新のリポジトリ、パッケージ、および依存関係を取得します。Ansible Automation Platform インストーラーをセットアップするには、次のいずれかの方法を選択します。

#### tarball インストール

1. [Red Hat Ansible Automation Platform のダウンロード](#) ページに移動します。
2. **Ansible Automation Platform <latest-version> Setup** の **Download Now** をクリックします。
3. ファイルをデプロイメントします。

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

#### RPM インストール

1. Ansible Automation Platform インストーラーパッケージをインストールします。  
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
```



#### 注記

**dnf install** は、リポジトリがデフォルトで無効になっているため、リポジトリを有効にします。

RPM インストーラーを使用すると、ファイルは `/opt/ansible-automation-platform/installer` ディレクトリに置かれます。

### 7.2. インターネットアクセスなしでのインストール

インターネットにアクセスできない場合や、オンラインリポジトリから個別のコンポーネントおよび依存関係をインストールしたくない場合は、Red Hat Ansible Automation Platform の **Bundle** インストーラーを使用します。Red Hat Enterprise Linux リポジトリへのアクセスは依然として必要です。その他の依存関係はすべて tar アーカイブに含まれます。

## 手順

1. [Red Hat Ansible Automation Platform のダウンロード](#) ページに移動します。
2. **Ansible Automation Platform <latest-version> Setup Bundle**の **Download Now** をクリックします。
3. ファイルをデプロイメントします。

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

## 第8章 インストーラーインベントリーファイルについて

Red Hat Ansible Automation Platform は、インベントリーファイルを使用して、論理的に編成されたインフラストラクチャー内の管理対象ノードまたはホストのリストに対して機能します。Red Hat Ansible Automation Platform インストーラーインベントリーファイルを使用して、インストールシナリオを指定し、Ansible へのホストのデプロイについて説明できます。インベントリーファイルを使用することで、Ansible は単一のコマンドで多数のホストを管理できます。インベントリーは、指定する必要があるコマンドラインオプションの数を減らすことで、Ansible をより効率的に使用するのにも役立ちます。

インベントリーファイルは、所有するインベントリープラグインに応じて、多数ある形式のいずれかになります。最も一般的な形式は **INI** と **YAML** です。このドキュメントに記載されているインベントリーファイルは、INI 形式で示されています。

インベントリーファイルの場所は、使用したインストーラーによって異なります。次の表に、可能な場所を示します。

インストーラー	ロケーション
Bundle tar	<code>/ansible-automation-platform-setup-bundle-&lt;latest-version&gt;</code>
Non-bundle tar	<code>/ansible-automation-platform-setup-&lt;latest-version&gt;</code>
RPM	<code>/opt/ansible-automation-platform/installer</code>

次のコマンドを使用して、インベントリー内のホストを確認できます。

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

### インベントリーファイルの例

```
[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>'
```

インベントリーファイルの最初の部分は、Ansible が使用できるホストまたはグループを指定します。

## 8.1. ホストとグループのガイドライン

### データベース

- 外部データベースを使用する場合は、インベントリーファイルの **[database]** セクションが正しく設定されていることを確認してください。
- パフォーマンスを向上させるために、データベースと Automation Controller を同じサーバーに配置しないでください。

### Automation Hub

- **[automationhub]** グループが存在する場合は、変数 **automationhub\_pg\_host** および **automationhub\_pg\_port** を含める必要があります。
- **[automationhub]** グループに Ansible Automation Hub 情報を追加します。
- Ansible Automation Hub と Automation Controller を同じノードにインストールしないでください。
- **[automationhub]** および **[automationcontroller]** ホストに到達可能な IP アドレスまたは完全修飾ドメイン名 (FQDN) を提供して、ユーザーが別のノードの Ansible Automation Hub および Automation Controller からコンテンツを同期してインストールできるようにします。FQDN には - 記号または \_ 記号を含めることはできません。正しく処理されません。

**localhost** は使用しないでください。

### Private Automation Hub

- Private Automation Hub と Automation Controller を同じノードにインストールしないでください。
- 同じ PostgreSQL (データベース) インスタンスを使用できますが、別の (データベース) 名を使用する必要があります。
- 内部アドレスから Private Automation Hub をインストールし、外部アドレスしか記載されていない証明書を使用している場合は、インストールして証明書の問題がなくてもコンテナレジストリーとして使用できなくなる可能性があります。



### 重要

Automation Controller と Ansible Automation Hub を別々にインストールする必要があります。両方が同時にインストールされている場合、**[database]** グループは 2 つを区別しないからです。

**[database]** で 1 つの値を使用し、Automation Controller と Ansible Automation Hub の両方がそれを定義する場合、それらは同じデータベースを使用します。

## Automation Controller

- Automation Controller は、使用するデータベースのレプリケーションやフェイルオーバーを設定しません。
- Automation Controller は、任意のレプリケーションで動作します。

## Event-Driven Ansible Controller

- Event-Driven Ansible Controller は別のサーバーにインストールする必要があります。Automation Hub および Automation Controller と同じホストにインストールすることはできません。

## クラスター化されたインストール

- 既存のクラスターをアップグレードする場合は、既存のインスタンスまたはインスタンスグループを省略するようにクラスターを再設定することもできます。インスタンスまたはインスタンスグループをインベントリーファイルから省略するだけでは、クラスターから削除するには不十分です。インベントリーファイルからインスタンスまたはインスタンスグループを除外するほかに、アップグレードを開始する前にインスタンスまたはインスタンスグループのプロビジョニングを解除する必要もあります。詳細は、[ノードまたはグループのプロビジョニング解除](#)を参照してください。そうしないと、省略されたインスタンスまたはインスタンスグループが引き続きクラスターと通信するため、アップグレード中に Automation Controller サービスで問題が発生する可能性があります。
- クラスター化されたインストールセットアップを作成している場合は、**[localhost]** をすべてのインスタンスのホスト名または IP アドレスに置き換える必要があります。Automation Controller および Automation Hub のインストーラーは、**[localhost]** を受け入れません。すべてのノードとインスタンスは、このホスト名またはアドレスを使用して他のノードに到達できるようにする必要があります。いずれかのノードで localhost **ansible\_connection=local** を使用することはできません。すべてのノードのホスト名に同じ形式を使用します。したがって、これは機能しません。

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

代わりに以下の形式を使用します。

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

または

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

## 8.2. ノードまたはグループのプロビジョニング解除

Ansible Automation Platform インストーラーを使用して、ノードとインスタンスグループのプロビジョニングを解除できます。インストーラーを実行すると、グループ内のノードに割り当てられたすべての設定ファイルおよびログが削除されます。



### 注記

**[automationcontroller]** グループで指定されている最初のホストを除き、インベントリーの任意のホストのプロビジョニングを解除することができます。

ノードのプロビジョニングを解除するには、インベントリーファイル内のノードまたはグループに **node\_state=deprovision** を追加します。

以下に例を示します。

デプロイメントから単一のノードを削除するには、以下を実行します。

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

または

デプロイからインスタンスグループ全体を削除するには、以下を実行します。

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

[instance_group_restrictedzone:vars]
node_state=deprovision
```

## 8.3. インベントリー変数

サンプルインベントリーファイルの **[all:vars]** に続く 2 番目の部分は、インストーラーによって使用される変数のリストです。**all** を使用すると、変数がすべてのホストに適用されます。

特定のホストに変数を適用するには、**[hostname:vars]** を使用します。たとえば、**[automationhub:vars]** です。

## 8.4. インベントリーファイルで変数を宣言するためのルール

文字列変数の値は、引用符で囲んで宣言します。以下に例を示します。

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

**:vars** セクションで宣言すると、INI 値は文字列として解釈されます。たとえば、**var=FALSE** は **FALSE** に等しい文字列を作成します。ホスト行とは異なり、**:vars** セクションは行ごとに1つのエントリーのみを受け入れるため、**=** の後のすべてがエントリーの値である必要があります。ホスト行は、行

ごとに複数の **key=value** パラメーターを受け入れます。したがって、スペースがセパレーターではなく値の一部であることを示す方法が必要です。空白を含む値は引用符で囲むことができます (一重または二重)。詳細は、[Python shlex の解析ルール](#) を参照してください。

INI インベントリに設定された変数値が特定の型 (文字列やブール値など) でなければならない場合は、常にタスクでフィルターを使用して型を指定します。変数を使用するときは、INI インベントリで設定されたタイプに依存しないでください。



### 注記

変数の実際の型に関する混乱を避けるために、インベントリソースに YAML 形式を使用することを検討してください。YAML インベントリプラグインは、変数値を一貫して正しく処理します。

Ansible インベントリファイルのパラメーター値に、#、{または}などの特殊文字が含まれている場合は、値をダブルエスケープ (double-escape) する必要があります (値を単一と二重引用符で囲みません)。

たとえば、**mypasswordwith#hashsigns** を変数 **pg\_password** の値として使用するには、Ansible ホストインベントリファイルで **pg\_password="mypasswordwith#hashsigns"** として宣言します。

## 8.5. インベントリファイルでシークレットを保護する

Ansible Vault を使用して機密変数または秘密変数を暗号化できます。ただし、変数名と変数値を暗号化すると、値のソースを見つけるのが難しくなります。これを回避するには、**ansible-vault encrypt\_string** を使用して変数を個別に暗号化するか、変数を含むファイルを暗号化します。

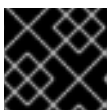
### 手順

1. 暗号化された認証情報を保存するために、**credentials.yml** というラベルの付いたファイルを作成します。

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

2. **ansible-vault** を使用して **credentials.yml** ファイルを暗号化します。

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



### 重要

暗号化された vault パスワードを安全な場所に保管します。

3. **credentials.yml** ファイルが暗号化されていることを確認します。

```
$ cat credentials.yml
$ANSIBLE_VAULT;1.1;
```



```
AES256363836396535623865343163333339613833363064653364656138313534353135303
764646165393765393063303065323466663330646232363065316666310a37306230313337
633963383130303334313534383962613632303761636632623932653062343839613639653
6356433656162333133653636616639313864300a3532393734333133396134653263393130
356335653534643565386536316334643438353464323766386235336136663261363433323
131633436393939646132656164333634306335343039356462646330343839663362323033
65383763
```

4. Ansible Automation Platform 2.4 のインストールのために **setup.sh** を実行し、**credentials.yml** と **--ask-vault-pass** オプションの両方を渡します。

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

## 8.6. 追加のインベントリーファイル変数

インベントリーファイルに追加変数を追加して、Red Hat Ansible Automation Platform インストールをさらに設定できます。これらの設定では、Red Hat Ansible Automation Platform 管理用のオプション機能を追加します。テキストエディターでインベントリーファイルを編集して、これらの変数を追加します。

インベントリーファイル変数の定義済み値の表が、Red Hat Ansible Automation Platform インストールガイドの [インベントリーファイル変数](#) に記載されています。

## 第9章 サポート対象のインストールシナリオ

Red Hat は、Red Hat Ansible Automation Platform 向けに以下のインストールシナリオをサポートします。

### 関連情報

インベントリーファイルのパラメーターを編集して、サポートされているインストールシナリオを指定するには、**Red Hat Ansible Automation Platform インストールガイド**の [インストールシナリオに基づくインベントリーファイルの例](#) を参照してください。

### 9.1. 同一ノード上にあるデータベースを持つスタンドアロン AUTOMATION CONTROLLER またはインストーラー以外が管理するデータベース

このシナリオでは、Web フロントエンド、REST API バックエンド、データベースを含む Automation Controller を1台のマシンにインストールします。PostgreSQL をインストールし、そのデータベースとして使用するように Automation Controller を設定します。これは、標準の Automation Controller のインストールシナリオとみなされます。

### 9.2. 外部管理データベースが設定されたスタンドアロン AUTOMATION CONTROLLER

このシナリオでは、単一のマシンに Automation Controller サーバーをインストールし、リモート PostgreSQL インスタンスとの通信をデータベースとして設定します。このリモート PostgreSQL は、管理するサーバーを使用することも、Amazon RDS などのクラウドサービスで提供することも可能です。

### 9.3. 内部データベースを備えた単一の EVENT-DRIVEN ANSIBLE CONTROLLER ノード

このシナリオには、内部データベースを備えた単一マシンへの Event-Driven Ansible コントローラーのインストールが含まれます。Automation Controller のインストールシナリオと同様の、インストーラー管理の PostgreSQL をインストールします。



#### 重要

インベントリーファイルに適切な Event-Driven Ansible 変数を設定する前に、Automation Controller をインストールする必要があります。

### 関連情報

- [単一の Automation Controller、単一の Automation Hub、および外部 \(インストーラー管理\) データベースを備えた単一の Event-Driven Ansible Controller ノード](#)
- [付録 A.5. Event-Driven Ansible Controller の変数](#)

### 9.4. 同じノード上にあるデータベースまたはインストーラー以外が管理するデータベースを使用するスタンドアロン AUTOMATION HUB

このシナリオでは、1台のマシンに Web フロントエンド、REST API バックエンド、データベースなど、Automation Hub のインストールが含まれます。PostgreSQL をインストールし、そのデータベースとして使用するよう Automation Hub を設定します。

## 9.5. 外部管理データベースを使用するスタンドアロン AUTOMATION HUB

このシナリオでは、1台のマシンに Automation Hub サーバーをインストールし、Red Hat Ansible Automation Platform インストーラーが管理するリモート PostgreSQL データベースをインストールします。

## 9.6. AUTOMATION CONTROLLER ノードまたはインストーラー以外が管理するデータベースを使用したプラットフォームインストール

このシナリオには、Automation Controller ノードにあるデータベース、またはインストーラー以外が管理するデータベースを使用した Automation Controller および Automation Hub のインストールが含まれます。

## 9.7. 外部管理データベースを使用したプラットフォームのインストール

このシナリオでは、Automation Controller と Automation Hub をインストールし、リモートの PostgreSQL インスタンスとの通信をデータベースとして設定します。このリモート PostgreSQL は、管理するサーバーを使用することも、Amazon RDS などのクラウドサービスで提供することも可能です。

## 9.8. 外部管理データベースを使用した複数マシンのクラスタのインストール

このシナリオでは、複数の Automation Controller ノードおよび Automation Hub インスタンスをインストールし、リモート PostgreSQL インスタンスとの通信をデータベースとして設定します。このリモート PostgreSQL は、管理するサーバーを使用することも、Amazon RDS などのクラウドサービスで提供することも可能です。このシナリオでは、すべての Automation Controller がアクティブでジョブを実行でき、すべてのノードが HTTP 要求を受信できます。

### 注記

- クラスタ設定で実行するには、Automation Controller が外部のものである必要があります。PostgreSQL はプライマリーまたはセカンダリーの Tower ノードの1つではないマシンにインストールする必要があります。冗長設定の場合、リモートの PostgreSQL バージョン要件は **PostgreSQL 13** です。
  - クラスタ化の設定に関する情報は、[クラスタリング](#) を参照してください。
- **[automationhub]** ホストの到達可能な IP アドレスを指定して、ユーザーが別のノードから Private Automation Hub のコンテンツを同期できるようにします。