

今すぐ始める

Red Hat Ansible Automation Platform の
使用を開始

目次

はじめに

第1章： 基本を学ぶ

- 1.1 プラットフォームのインストール
- 1.2 プラットフォームの主なコンポーネント
- 1.3 Ansible Playbook
- 1.4 Ansible Role
- 1.5 インベントリ
- 1.6 Ansible Content Collections

第2章： 一般的なユースケースから始める

- 2.1 クラウドリソースのプロビジョニング
- 2.2 仮想マシンを大規模にデプロイ
- 2.3 サービスの管理
- 2.4 コンプライアンス監査の実施
- 2.5 システム構成の管理
- 2.6 アプリケーションのデプロイ
- 2.7 ネットワークデバイスの構成
- 2.8 オペレーティングシステムのアップグレード

第3章： 高度な自動化のユースケースを見る

- 3.1 イベント駆動型の自動化
- 3.2 エッジでの自動化
- 3.3 エンドツーエンドのクラウドワークフロー
- 3.4 組織規模の自動化

関連資料および情報



はじめに

自動化は、複雑化が進む現在の IT 環境を管理するための重要なテクノロジーです。

自動化により、スタッフの時間を節約でき、製品の品質が改善され、チーム間のコラボレーションが促進されます。また、従業員の満足度が上昇し、組織全体でコストを削減できます。

自動化サービスの広範な構築と運用の基盤となる **Red Hat® Ansible® Automation Platform** は、組織規模での自動化の導入に必要なすべてのツールと機能を提供します。このプラットフォームは、IT プロセスからハイブリッドクラウドやエッジロケーションまで、エンドツーエンドの自動化ワークフローをデプロイするための、柔軟で安定したセキュリティ重視の基盤を提供します。Ansible Automation Platform によって、組織全体での自動化の作成、管理、スケーリングが可能になります。

この e ブックでは、Ansible Automation Platform を使い始めるためのガイダンスを提供します。自動化の主要な概念、製品の特長と機能、迅速に価値を得るためのユースケース、組織規模の自動化を実装するための戦略について確認します。

自動化の価値に関するデータ

Red Hat Ansible Automation Platform は、人材とプロセスを柔軟な基盤上に統合し、組織全体に価値をもたらします。

36%

開発チームの生産性向上¹

68%

新しいコンピューティング・リソースのデプロイメントが迅速化¹

23%

新しい製品およびサービスの市場投入時間を短縮¹

61%

予定外のダウンタイムの削減¹

668%

3 年間の投資対効果 (ROI)¹

854 万米ドル

組織当たりの年間収益が増加¹

第1章：

基本を学ぶ

Red Hat Ansible Automation Platform を使い始める前に、このプラットフォームに含まれるもの、コンポーネントがどのように連携するか、および自動化に関する主要な概念 (Playbook、ロール、インベントリーなど) を理解することが重要です。

この章の内容

- 1.1 プラットフォームのインストール
- 1.2 プラットフォームの主なコンポーネント
- 1.3 Ansible Playbook
- 1.4 Ansible Role
- 1.5 インベントリー
- 1.6 Ansible Content Collections

Red Hat Ansible Automation Platform のインストール

Red Hat Ansible Automation Platform のインストールとセットアップは、すばやく簡単にできます。インストールプログラムでは複数のインストールシナリオがサポートされており、Ansible Automation Platform を柔軟にインストールできます (任意で Event-Driven Ansible コントローラーを併せてインストールすることも可能)。各シナリオの詳細については、以下のインストールガイドを参照してください。

- ▶ [従来の仮想マシン](#)
- ▶ [Red Hat OpenShift®](#)
- ▶ [クラウド環境](#)
- ▶ [コンテナ化された環境](#)

いずれのインストールシナリオを選択しても、Ansible Automation Platform のインストールは次の手順で行います。

1. **Ansible Automation Platform インストーラー・インベントリー・ファイル**を編集して、インストールシナリオを指定し、Ansible へのホストデプロイメントを記述する
2. **Ansible Automation Platform インストーラー・セットアップ・スクリプト**を実行し、インストーラー・インベントリー・ファイルで定義されたパラメーターを使用してプライベート Automation Hub をインストールする
3. インストールが成功したことを確認するため、automation controller、Automation Hub、任意で Event-Driven Ansible コントローラーにログインする

プラットフォームの主要コンポーネントを理解する

Red Hat Ansible Automation Platform にはいくつかの主要コンポーネントが含まれています。これらのコンポーネントは、組織全体で自動化を作成、管理、スケーリングする際に使用します。

Ansible Automation Hub

Ansible Automation Platform サブスクリプションで利用できる **Ansible Automation Hub** は、**Ansible Content Collections** と **検証済みコンテンツ**を検出、ダウンロード、管理するための中央リポジトリです。**プライベート Automation Hub** はオンサイトのリポジトリで、分離された環境を持つ企業が独自の自動化コンテンツを管理、共有、厳選し、Red Hat やパートナーが作成した認定済みコンテンツおよび検証済みコンテンツへのアクセスを制御できるようにします。



Ansible Content Collections の詳細は **10 ページ**をご覧ください。

automation controller

automation controller は Ansible Automation Platform のコントロールプレーンです。組織全体での自動化のデプロイ、開始、権限委任、監査を可能にします。automation controller には、ユーザー・インターフェース、ロールベースのアクセス制御 (RBAC)、ワークフローが含まれており、自動化を効率的かつ柔軟に拡張できるようにします。インベントリーの管理、ワークフローの起動とスケジュール、変更の追跡、レポートの生成のすべてを、一元化されたユーザー・インターフェースと **RESTful アプリケーション・プログラミング・インターフェース (API)** から実行します。



automation controller の構成および使用方法については、**スタートガイド**、**管理ガイド**、**ユーザーガイド**を参照してください。

Automation mesh

Automation mesh は、既存の接続性を使用してノードのコレクション全体への自動化タスクの分散を単純化するオーバーレイネットワークです。実行ノードは、Ansible Playbook で定義されたタスクを実行する automation execution environment を実行します。Automation mesh は、これらの実行ノード間でピアツーピア接続を確立し、ネットワークのレイテンシーや接続の中断に対する自動化ワークロードのレジリエンシー (回復力) を高めます。これによってアーキテクチャの柔軟性も向上し、制御および実行能力の迅速かつ独立したスケーリングが可能になります。

Event-Driven Ansible

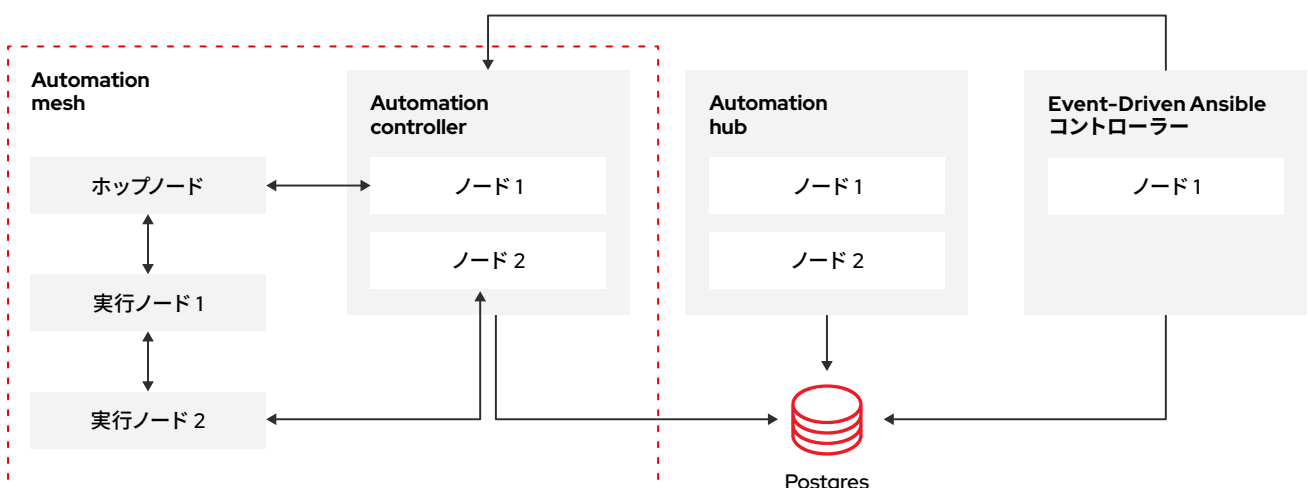
Event-Driven Ansible は、IT ドメインにおける IT 状態の変化に自動応答するためのイベント処理機能を提供します。IT 環境での条件に関する個別のインテリジェンスを含むイベントを処理し、イベントに対する適切な対応を判断し、イベントに対処または修復するための自動化されたアクションを実行できます。IT サービス管理タスクにはチケット強化、修復、ユーザー管理などが含まれ、最初に取り扱う対象として最適です。しかし、Event-Driven Ansible は IT 環境上のさまざまなタスクを自動化するに足りる柔軟性を備えています。



Event-Drive Ansible
の詳細は [23 ページ](#)
をご覧ください。

Red Hat Ansible Automation Platform 2.4 のアーキテクチャ

これらの各コンポーネントは、Ansible Automation Platform のアーキテクチャ内で連携します。



Ansible Playbook を理解する

Playbook は、Red Hat Ansible Automation Platform を介して IT 資産を設定し、デプロイし、オーケストレーションするための手順を記述したものです。コマンドのセット（「play」）で構成され、ホストのインベントリーに対する自動化を定義します。それぞれの play にはインベントリー内の 1 つ、複数、またはすべてのホストを対象とする、1 つ以上のタスクが含まれています。各タスクは**モジュール**を呼び出し、情報の収集、設定の管理、接続性の検証などの固有の機能を実行します。Playbook は複数のチームで共有し、再利用できるので、反復可能な自動化を作成できます。

この例は、Ansible Playbook の共通部分を示しています。

```
1 ---
2 - name: Add VLANs
3   hosts: arista
4   gather_facts: false
5
6   vars:
7     vlans:
8       - name: desktops
9         vlan_id: 20
10      - name: servers
11        vlan_id: 30
12      - name: DMZ
13        vlan_id: 50
14
15   tasks:
16     - name: Add VLAN configuration
17       arista.eos.eos_vlans:
18         state: merged
19         config: "{{ vlans }}"
```

Playbook の開始を示します。

arista という名前のデバイスまたはデバイスのグループを呼び出します。
指定したデバイスに関するファクトを取得するための任意のパラメーター。

変数の定義
この Playbook では、変数の値を直接定義します。
automation controller により、**Survey を作成**して、Playbook を実行する際にユーザーが変数の値を入力できるようにすることもできます。

タスク
タスクと Ansible モジュールは 1 対 1 の関係にあります。このセクションではモジュールを呼び出して、vars セクションで定義された 3 つの変数それぞれに対して仮想ローカルエリア・ネットワーク (VLAN) を設定します。

YAML とは

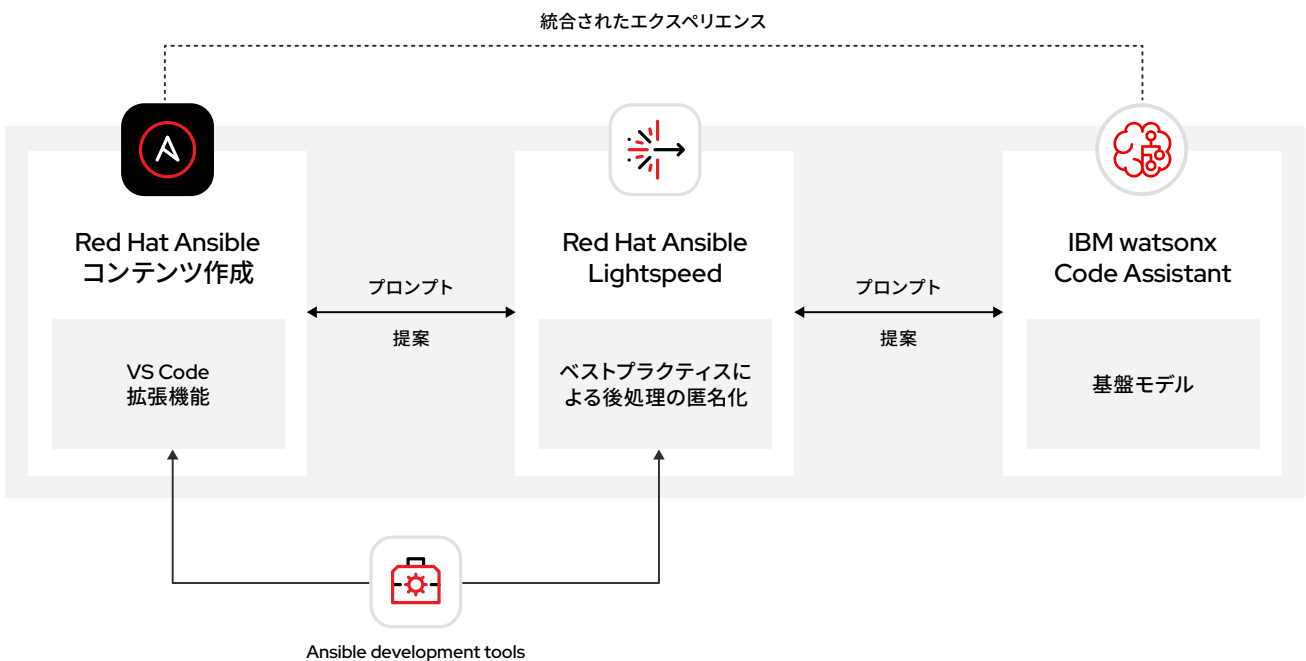
YAML は、容易に読み書きができ、理解しやすいように設計された、人に優しいデータシリアル化言語です。多くのプログラミング言語とは異なり、中括弧、角括弧、閉じタグ、引用符などの通常の書式記号はありません。YAML ファイルではスペース文字によるインデントを使用して構造を決定し、ネストを示すので、読みやすくなっています。柔軟で利用しやすいことから、Ansible Playbook、Rulebook、およびインベントリーは YAML を使用してすべての自動化コンテンツを完全に定義します。

Playbook の作成を単純化

Playbook は YAML で簡単に記述できますが、Playbook 作成の単純化と効率化のために利用できるツールが 2 つあります。

Ansible VS Code 拡張機能 は、`ansible` と `ansible-lint` をサポートするオペレーティングシステム上で稼働する Visual Studio Code および OpenVSX 互換エディターに Ansible の言語サポートを追加します。Ansible のキーワード、モジュール名、モジュールオプション、および標準の YAML 要素が認識され、強調表示されます。入力中に Ansible スクリプトの構文が検証され、即座にフィードバックが提供されます。ドキュメントを開いて保存すると、バックグラウンドで `ansible-lint` が実行され、潜在的な問題が見つければ報告されます。スマート入力補助機能は、カーソルが `play`、ブロック、またはタスクのいずれにあるのかを検出し、関連する提案を提供します。

Red Hat Ansible Lightspeed with IBM watsonx Code Assistant は、開発者が Ansible コンテンツをより効率的に作成できるようにする生成 AI (人工知能) サービスです。ユーザーによって入力されたプロンプトを基に、IBM watsonx 基盤モデルと対話して、Ansible のベストプラクティスに基づき、コード候補を生成します。Ansible Lightspeed with watsonx Code Assistant はコンテキストを認識するので、`play` 名やその他のモジュールなど、Playbook 内にある他の単語に基づいて提案を提供することもできます。



Ansible VS Code 拡張機能で Ansible Lightspeed with watsonx Code Assistant を有効化する方法については、[こちらの動画](#)をご覧ください。

ロールで再利用可能な自動化コンテンツを作成する

自動化コンテンツを共有して再利用することで、チームの作業効率が向上し、作業の重複を回避できます。Red Hat Ansible Automation Platform は、再利用可能なコンテンツを作成し、それを組織全体に迅速に配信できるようにする技術を提供します。

Ansible Role を使用すると、構成ファイル、テンプレート、タスク、ハンドラーなどの関連する自動化アーティファクトをグループ化およびカプセル化することで、再利用可能なコンポーネントを構築できます。これらのコンポーネントはロールによって分離されるため、コンポーネントの再利用や他のユーザーとの共有が容易になります。また、ユーザーがロールを呼び出すときに設定できる変数を公開することでロールの構成を可能にし、特定の要件に従ってシステムを構成できるようにすることもできます。



ロールによって再利用可能な Playbook を作成する方法については、[こちらの記事](#)をお読みください。



インベントリーとは

インベントリーは、Ansible コマンドと Playbook を使用して実行されるホストの集合です。インベントリーファイルはホストをグループに分類し、IT 資産の信頼ソースとして機能します。これらのファイルの形式としてはシンプルな INI または YAML を使用できます。多くの組織では、Playbook との一貫性のために YAML でインベントリーを作成します。インベントリーファイルを使用すると、1 つの Playbook から数百のデバイスを、コマンド 1 つで保守することもできます。

このセクションでは、インベントリーファイルを構築する方法を説明します。[インベントリーレポートのサンプル](#)を GitHub で参照することもできます。

INI 形式の基本インベントリーの作成

まず、インベントリーを論理的にグループ化します。ベストプラクティスとして、**何を**（アプリケーション、スタック、またはマイクロサービス）、**どこで**（データセンターまたはリージョン）、**いつ**（開発段階）という基準で、サーバーとデバイスをグループ化します。その例をいくつかご紹介します。

- ▶ **何を**: db、web、leaf、spine
- ▶ **どこで**: east、west、floor_19、building_A
- ▶ **いつ**: dev、test、staging、prod

この INI 形式のサンプルコードでは、極めて小規模なデータセンターの基本グループ構造を示しています。`[metagroupname:children]` という構文を使用して、メタグループのメンバーとしてグループを登録することで、グループをグループ化することもできます。

ここでは、グループ `network` にはすべてのリーフとすべてのスパインが含まれています。グループ `datacenter` には、すべてのネットワークデバイスと、すべての Web サーバーが含まれています。

詳細については、Ansible ドキュメントの「[Build your inventory \(インベントリーの構築\)](#)」のセクションを参照してください。

```
1  [leafs]
2  leaf01
3  leaf02
4
5  [spines]
6  spine01
7  spine02
8
9  [network:children]
10 leafs
11 spines
12
13 [webservers]
14 webserver01
15 webserver02
16
17 [datacenter:children]
18 network
19 webservers
```

YAML 形式のインベントリーの構造

```

1  ---
2  all:
3    vars:
4      ansible_user: admin
5      ansible_password: password123
6      ansible_become_pass: password123
7      ansible_become: True
8      ansible_become_method: enable
9      ansible_network_cli_ssh_type: libssh
10 children:
11   routers:
12     children:
13       arista:
14       cisco:
15       juniper:
16   arista:
17     hosts:
18       rtr2:
19         ansible_host: 172.16.100.2
20       rtr4:
21         ansible_host: 172.16.100.4
22     vars:
23       ansible_network_os: arista.eos.eos
24       ansible_connection: ansible.netcommon.network_cli
25   cisco:
26     hosts:
27       rtr1:
28         ansible_host: 172.16.100.1
29     vars:
30       ansible_network_os: cisco.ios.ios
31       ansible_connection: ansible.netcommon.network_cli
32   juniper:
33     hosts:
34       rtr3:
35         ansible_host: 172.16.100.3
36     vars:
37       ansible_network_os: junipernetworks.junos.junos
38       ansible_connection: ansible.netcommon.netconf

```

Playbook の開始を示します。

グループにかかわらず、インベントリー内のすべてのホストに適用される変数を定義します。

グループ階層

行 10 - 15 はこのインベントリー内のホストグループを識別します。この場合、グループ `routers` には、`arista`、`cisco`、`juniper` の 3 つのサブグループが含まれています。

グループ定義

`hosts` コマンドは、各グループにどのホストが属するかを定義します。このサンプルでは、グループ `arista` には IP アドレスで識別される 2 つのホストが含まれています。

グループ変数

各グループには固有の変数セットを指定できます。このインベントリーでは、各グループにオペレーティングシステムと接続タイプを定義しています。これらのグループ変数はすべて、Content Collections に含まれるアイテムを指しています。

バージョン管理によってフラストレーションを軽減

Playbook、ロール、インベントリ、変数ファイルを有益なコミットメッセージやコメントとともにバージョン管理システムに保存しておく、自動化コンテンツの作成、レビュー、および修正方法を単純化することができます。バージョン管理ツールは、ファイルに加えられた変更を追跡することでコンテンツ開発の長期にわたる監視と監査を可能にし、問題が発生した場合にはすぐに以前のバージョンに戻せます。また、ブランチの作成とマージに対応しているため、複数のチームメンバーが互いに影響を与えることなく同じ自動化コンテンツに取り組むことができます。

Git バージョン管理の初心者ガイドで、広く普及しているこのバージョン管理システムの詳細をご覧ください。

Ansible Content Collections を使用してすばやく開始する

Ansible Content Collections は、自動化コンテンツの標準化された配布形式です。コレクションには **Playbook**、**ルールブック**、**ロール**、**モジュール**、その他の **プラグイン**、インベントリ、ドキュメントが含まれます。

すぐに利用できるコレクションは以下の 2 種類です。

- ▶ **Red Hat Ansible Certified Content** には、Red Hat および当社のテクノロジーパートナーによって構築、サポート、保守されたコレクションが含まれています。これらのコレクションは Ansible Automation Hub から直接利用でき、Red Hat とパートナーのプラットフォーム間の統合に重点を置いているため、さまざまな IT ドメインおよびテクノロジーで自動化できます。
- ▶ **Ansible 検証済みコンテンツ** は、Ansible Automation Platform を使用して Red Hat とパートナーのプラットフォームで操作やタスクを実行する方法について、エキスパートによるガイダンスを提供します。通常、ロールやドキュメントとして提供される検証済みコンテンツは、カスタマイズ可能な独自のユースケースを含み、Red Hat Ansible Certified Content に基づいている場合があります。検証済みコンテンツは、automation controller および Automation Hub の構成や実行環境の作成など、Ansible Automation Platform 全体に関連する追加の運用機能を提供することができます。

また、**内部向けコレクション**を開発し、プライベート Automation Hub にアップロードして組織内で配信することもできます。コレクションが公開されると、内部ユーザーはそれをダウンロードして使用することができます。



Ansible Automation Platform の認定済みコンテンツと検証済みコンテンツの詳細は、Red Hat カスタマーポータルをご覧ください。

第 2 章：

一般的なユースケースから始める

自動化の導入は「0 か 100 か」といった二択ではありません。1 つのユースケースから始めて、組織に適したペースで拡張できます。

この章では、組織内での自動化のメリットを探索する上で役立てただけよう、一般的な自動化のユースケースの概要を説明します。これらのユースケースの自動化に必要な時間と作業量はそれぞれ異なります。自動化プロジェクトの複雑さとこれらのユースケースの最適な適用方法を理解するために、現在のプロセスのフローチャートを作成することをお勧めします。成功を収めた自動化導入は多くの場合、段階的に進展しています。チームは小さく始めて価値を示し、作業の範囲と複雑さを徐々に増していきます。

タスクを手動で実行する場合と、同じタスクを自動化する場合の時間とコストを測定して分析することで、組織全体で自動化することでどの程度節約できるかを判断できます。

Automation Calculator は、自動化投資の合計節約額を算出するために役立つグラフ、指標、計算方法を提供します。

Red Hat Ansible Automation Platform 向けの Automation Analytics と Red Hat Insights を使用すると、実用的なデータに基づいて自動化を計画、測定、管理、拡張できます。サブスクリプションに含まれているこれら 2 つのツールは、自動化の取り組みを追跡するのに役立つ充実したレポート作成機能と可観測性メトリクスを提供します。Automation Analytics ツールを使用して、投資対効果 (ROI) や自動化のパフォーマンスなど、Ansible Automation Platform によるビジネスへの影響を測定できます。Red Hat Insights を使用すると、システムの構成や健全性の状態などを含む、自動化インフラストラクチャを監視できます。これによりインフラストラクチャのパフォーマンスの問題、システムの可用性、セキュリティの脆弱性などの問題を発見し、解決することができます。Red Hat Insights からの通知をデータソースとして使用して、Event-Driven Ansible による自動修復を開始することもできます。

この章の内容

- 2.1 クラウドリソースのプロビジョニング
- 2.2 仮想マシンを大規模にデプロイ
- 2.3 サービスの管理
- 2.4 コンプライアンス監査の実施
- 2.5 システム構成の管理
- 2.6 アプリケーションのデプロイ
- 2.7 ネットワークデバイスの構成
- 2.8 オペレーティングシステムのアップグレード

ユースケース: クラウドリソースのプロビジョニング

多くの組織にとって、クラウドリソースのプロビジョニングは時間がかかり、エラーが発生しやすいプロセスです。Red Hat Ansible Automation Platform を使用すると、パブリッククラウド環境での仮想マシンのプロビジョニングを単純化できます。Ansible Certified Content を使用して Playbook を作成し、ストレージの割り当て、ネットワークとサブネットワークのセットアップ、仮想マシンインスタンスのプロビジョニングを行います。インスタンスタイプ、ゾーン、セキュリティグループなどの構成オプションの変数を追加して、Playbook を再利用できるようにすると、どこにでも仮想マシンをデプロイできます。

ここでは、パブリッククラウド環境でクラウドリソースをプロビジョニングするためのタスクの例を 3 つご紹介します。

例: ディスクとネットワーク・インタフェースを備えた Google Cloud インスタンスの作成

```
1 ---
2 - name: create a instance
3   google.cloud.gcp_compute_instance:
4     name: test_object
5     machine_type: n1-standard-1
6     disks:
7       - auto_delete: 'true'
8         boot: 'true'
9         source: "{{ disk }}"
10      - auto_delete: 'true'
11        interface: NVME
12        type: SCRATCH
13        initialize_params:
14          disk_type: local-ssd
15        labels:
16          environment: production
17      network_interfaces:
18        - network: "{{ network }}"
19          access_configs:
20            - name: External NAT
21              nat_ip: "{{ address }}"
22              type: ONE_TO_ONE_NAT
23      zone: us-centrall-a
24      project: test_project
25      auth_kind: serviceaccount
26      state: present
```



Google Cloud 向けの Red Hat Certified Content の詳細は、Red Hat Ecosystem Catalog をご覧ください。

例：パブリック IP アドレスを備えた AWS EC2 インスタンスの作成

```
1 ---
2 - name: Provision AWS EC2 instance
3   amazon.aws.ec2_instance:
4     name: "public-compute-instance"
5     key_name: "prod-ssh-key"
6     vpc_subnet_id: subnet-5calable
7     instance_type: c5.large
8     security_group: default
9     network:
10      assign_public_ip: true
11     image_id: ami-123456
12     tags:
13       Environment: Testing
```



AWS 向けの Red Hat Certified Content の詳細は、Red Hat Ecosystem Catalog をご覧ください。

例：マネージドディスクを備えた Microsoft Azure 仮想マシンの作成

```
1 ---
2 - name: Provision Microsoft Azure instance
3   azure_rm_virtualmachine:
4     name: vm-managed-disk
5     vm_size: Standard_D4
6     resource_group: myResourceGroup
7     admin_username: "{{ username }}"
8     availability_set: avs-managed-disk
9     managed_disk_type: Standard_LRS
10    image:
11      offer: RHEL
12      publisher: RedHat
13      sku: '8-lvm-gen2'
14      version: latest
```



Microsoft Azure 向けの Red Hat Certified Content の詳細は、Red Hat Ecosystem Catalog をご覧ください。

ユースケース：仮想マシンを大規模にデプロイ

手作業によるプロセスで仮想マシンをデプロイすると、構成ミスや予期せぬ問題が発生し、ダウンタイムやサービスの中断につながる可能性があります。Red Hat Ansible Automation Platform を使用すると、標準化された仮想マシンテンプレートを自動的に作成し維持できるため、VMware vSphere 環境全体で一貫して仮想マシンをプロビジョニングできます。VMware コレクションの `vmware.vmware_rest.vcenter_vmtemplate_libraryitems` モジュールを使用して、静的仮想マシンイメージに基づくテンプレートを作成します。そして同じモジュールを使用して、そのテンプレートに基づいて新しい仮想マシンを一貫してデプロイします。

例：テンプレートを使用して VMware で仮想マシンをデプロイ

```
1 ---
2 - name: Deploy a new VM based on the template
3   vmware.vmware_rest.vcenter_vmtemplate_libraryitems:
4     name: vm-from-template
5     library: "{{ nfs_lib.id }}"
6     template_library_item: "{{ my_template_item.id }}"
7     placement:
8       cluster: "{{ lookup('vmware.vmware_rest.cluster_moid',
9                            '/my_dc/host/my_cluster') }}"
10    folder: "{{ lookup('vmware.vmware_rest.folder_moid',
11                       '/my_dc/vm') }}"
12    resource_pool: "{{ lookup('vmware.vmware_rest.resource_pool_moid',
13                              '/my_dc/host/my_cluster/Resources') }}"
14    state: deploy
```



詳細は [Ansible による VMware テンプレートのライフサイクルの管理](#) を参照してください。VMware コレクションは Red Hat Ecosystem Catalog でご覧いただけます。

ユースケース：サービスの管理

頻繁な再起動を要する既知の問題があるサービスの管理は困難が伴います。Red Hat Ansible Automation Platform を使用すれば、アプリケーションやサービスで繰り返し発生する問題に迅速に対応できます。`ansible.builtin.systemd` や `ansible.builtin.sysvinit` などの組み込みモジュールにより、サービスマネージャーを選択してリモートホスト上のサービスを制御できます。`ansible.builtin.service` モジュールはサービスマネージャーモジュールのプロキシとして機能するため、サービスマネージャーごとに特定のタスクを作成せずにさまざまな環境を管理できます。その結果、影響を受けるシステムとアプリケーション層に関する情報を自動的に収集し、問題が報告されるとすぐにサービスを再起動する簡単な Playbook を作成できます。

例：サービスの開始

```
1 ---
2 - name: Start service httpd, if not started
3   ansible.builtin.service:
4     name: httpd
5     state: started
```

例：サービスの停止

```
1 ---
2 - name: Stop service httpd, if started
3   ansible.builtin.service:
4     name: httpd
5     state: stopped
```

例：サービスの再起動

```
1 ---
2 - name: Restart service httpd
3   ansible.builtin.service:
4     name: httpd
5     sleep: 60
6     state: restarted
```



`ansible.builtin.service` モジュールでサービスを管理する方法については、Ansible Automation Platform のドキュメントを参照してください。

ユースケース：コンプライアンス監査の実施

ほとんどの環境にはさまざまなプラットフォームやデバイスが多数含まれているため、手動によるコンプライアンス監査は困難で時間がかかります。Red Hat Ansible Automation Platform は、IT 環境全体のリソースの監査方法を単純化し、標準化します。Ansible Certified Content を使用して Playbook を作成し、少ない手作業でシステム構成のクエリ、保存、レポート作成を実行します。また、システム構成があるべき状態と異なっている場合、Ansible Automation Platform はサービスチケットを自動的にログに記録し、必要に応じて構成を修正できます。

例：ネットワークファクトの収集

```
1 ---
2 - name: Use Cisco IOS facts module
3   hosts: cisco
4   gather_facts: false # this is not the cisco facts module
5
6   tasks:
7     - name: retrieve facts
8       cisco.ios.ios_facts:
9
10    - name: display version
11      ansible.builtin.debug:
12        msg: "{{ ansible_net_version }}"
13
14    - name: display serial number
15      ansible.builtin.debug:
16        msg: "{{ ansible_net_serialnum }}"
```

例: ネットワークリソース情報の取得

```
1 ---
2 - name: Retrieve interface information
3   hosts: cisco
4   gather_facts: false # this is not the cisco facts module
5
6   tasks:
7     - name: use state gathered
8       cisco.ios.ios_interfaces:
9         state: gathered
10      register: interfaces_info
11
12     - name: print interface information
13       ansible.builtin.debug:
14         msg: "{{ interfaces_info }}"
```

例: ネットワーク構成のバックアップ

```
1 ---
2 - hosts: cisco
3   gather_facts: false
4
5   tasks:
6     - name: Back up config
7       cisco.ios.ios_config:
8         backup: yes
```

ユースケース：システム構成の管理

最新のセキュリティ基準に従ってリソースを最新の状態に保つことで、システムを保護し、脆弱性を軽減することができます。Red Hat Enterprise Linux® System Roles は、Red Hat Enterprise Linux の複数のリリースを自動化し管理するための、安定かつ一貫した構成インタフェースを提供する Ansible Certified Content のコレクションです。これらのロールを使用して Playbook を作成およびレビューし、セキュリティ基準が変更されるたびにシステム構成を自動的に更新します。



詳細は、Ansible Automation Platform ドキュメントの **Red Hat Enterprise Linux System Roles** の概要を参照してください。システムロールとドキュメントは Red Hat Hybrid Cloud Console (console.redhat.com) でご覧いただけます。

例：カーネル設定のアップデート

```
1 ---
2 - name: Manage kernel settings
3   hosts: all
4   vars:
5     kernel_settings_sysctl:
6       - name: fs.epoll.max_user_watches
7         value: 785592
8       - name: fs.file-max
9         value: 379724
10      - name: kernel.threads-max
11        state: absent
12
13     kernel_settings_sysfs:
14       - name: /sys/kernel/debug/x86/pti_enabled
15         value: 0
16       - name: /sys/kernel/debug/x86/retp_enabled
17         value: 0
18       - name: /sys/kernel/debug/x86/ibrs_enabled
19         value: 0
20
21     kernel_settings_systemd_cpu_affinity: "1,3,5,7"
22     kernel_settings_transparent_hugepages: madvise
23     kernel_settings_transparent_hugepages_defrag: defer
24   roles:
25     - linux-system-roles.kernel_settings
```

ユースケース: アプリケーションのデプロイ

アプリケーションを手作業でデプロイするとエラーが発生しやすく、セキュリティリスクが増大したり、アプリケーションのパフォーマンスが低下したりすることがあります。Red Hat Ansible Automation Platform には、環境全体でアプリケーションのインストールと構成を簡単に、一貫して行うための再利用可能な Playbook を作成できる組み込みモジュールが含まれています。認証済みモジュールを使って、YUM (Yellowdog Updater Modified) または DNF (Dandified YUM) を用いた Web サーバーのインストール、デフォルトホームページの設定、サーバーの起動、ファイアウォールの設定などを、すべて単一の読みやすい Playbook で行うことができます。

例: Web サーバーのデプロイ

```
1 ---
2 - name: Setup the web server
3   hosts: "{{ hosts }}"
4   become: true
5   tasks:
6     - name: httpd installed
7       ansible.builtin.yum:
8         name: httpd
9         state: latest
10
11    - name: custom index.html
12      ansible.builtin.copy:
13        dest: /var/www/html/index.html
14        content: | Custom Web Page
15
16    - name: httpd service enabled
17      ansible.builtin.service:
18        name: httpd
19        enabled: true
20        state: started
21
22    - name: open firewall
23      ansible.posix.firewalld:
24        service: http
25        state: enabled
26        immediate: true
27        permanent: true
```

ユースケース：ネットワークデバイスの構成

ネットワークの設定とアップデートを手動で行うとかなりの時間を必要とするため、先進的なアプリケーションやデータ転送の要件に効果的に対応することができません。Red Hat Ansible Certified Content により、ハイブリッドクラウドの多くの一般的なネットワークタスクを自動化できます。ルーターのホスト名とドメインネームシステム (DNS) サーバーを構成する Playbook を記述し、仮想ローカルエリアネットワーク (VLAN) 構成を作成して環境全体に伝達します。

例：ルーターの構成

```
1 ---
2 - name: configure cisco routers
3   hosts: routers
4   connection: ansible.netcommon.network_cli
5   gather_facts: false
6   vars:
7     dns: "8.8.8.8 8.8.4.4"
8
9   tasks:
10    - name: configure hostname
11      cisco.ios.ios_config:
12        lines: hostname {{ inventory_hostname }}
13
14    - name: configure DNS
15      cisco.ios.ios_config:
16        lines: ip name-server {{dns}}
```



ネットワークを自動化する方法について詳しくは、Ansible Automation Platform ドキュメントの [Ansible ネットワークロールを使用する](#) を参照してください。

例: VLAN の追加

```
1 ---
2 - name: add vlans
3   hosts: arista
4   gather_facts: false
5
6   vars:
7     vlans:
8       - name: desktops
9         vlan_id: 20
10      - name: servers
11        vlan_id: 30
12      - name: DMZ
13        vlan_id: 50
14
15   tasks:
16     - name: add VLAN configuration
17       arista.eos.eos_vlans:
18         state: merged
19         config: "{{ vlans }}"
```

ユースケース: オペレーティングシステムのアップグレード

オペレーティングシステムのアップグレードのようなインフラストラクチャ保守のタスクでは、多くの場合、さまざまなチームの大勢の IT スタッフが通常の営業時間外に作業する必要があります。Red Hat Ansible Automation Platform により、環境全体で Red Hat Enterprise Linux オペレーティングシステムのアップグレードを実行するための複雑な自動化ワークフローを作成できます。新しいオペレーティングシステムのバージョンをダウンロードおよびインストールし、仮想マシンを条件付きで再起動し、インストールされたサービスとパッケージについて説明するレポートを自動的に作成する Playbook を作成します。

例: Red Hat Enterprise Linux のインストールへのパッチ適用

```
1 ---
2 - name: Upgrade all packages (yum)
3   ansible.builtin.yum:
4     name: '*'
5     state: latest
6     update_only: true
7   when: ansible_pkg_mgr == "yum"
8   register: patchingresult_yum
9
10
11 - name: Upgrade all packages (dnf)
12   ansible.builtin.dnf:
13     name: '*'
14     state: latest
15     update_only: true
16   when: ansible_pkg_mgr == "dnf"
17   register: patchingresult_dnf
18
19
20 - name: Check to see if we need a reboot
21   ansible.builtin.command: needs-restarting -r
22   register: result
23   changed_when: result.rc == 1
24   failed_when: result.rc > 1
25   check_mode: false
26
27
28 - name: Reboot Server if Necessary
29   ansible.builtin.reboot:
30     when:
31       - result.rc == 1
32       - allow_reboot == true
```

第3章：

高度な自動化の ユースケースを見る

一般的なユースケースを自動化し、自動化による価値を提供できるようになったら、その経験を通じて習得した知識とプロセスを組織規模のより高度なユースケースで活用することができます。

この章では、Red Hat Ansible Automation Platform によって自動化のプロジェクトおよび目標を推進するための次のステップについて説明します。

この章の内容

- 3.1 イベント駆動型の自動化
- 3.2 エッジでの自動化
- 3.3 エンドツーエンドのクラウドワークフロー
- 3.4 組織規模の自動化

イベント駆動型の自動化で時間と労力を節約

イベント駆動型の自動化は、エンドツーエンドの IT 自動化への次のステップです。イベント駆動型の自動化では、IT 環境で特定のイベントや条件が発生すると自動的に応答します。可観測性ツールやその他のツールから情報を受け取ると、どのアクションを取るべきかを決定し、条件付きルールに基づいて事前定義されたアクションを開始します。ネットワークやシステムの色度低下、構成ドリフト、インフラストラクチャ状況の変化、新たなサービスチケットのエントリーなどのイベントへの対応を自動化することで、環境全体で革新的で複雑なワークフローを作成する柔軟性が得られます。

Red Hat Ansible Automation Platform には、イベント駆動型のパワフルな自動化機能が搭載されています。**Event-Driven Ansible** なら、手作業を行うことなく、IT 環境でイベントや状況が検出されたときに事前に決められた方法で対応できます。**Ansible Rulebook** で **if-then** ルール、イベントソース、自動化アクションを定義するだけで、プラットフォームがサードパーティの監視および可観測性ツールから受け取ったイベントを適切なルールブックと照合し、適切なアクションを決定してアクションを実行します。

ルールブックとは

ルールブックは、Event-Driven Ansible がアクションを実行するために使用する条件付きルールのセットです。ルールブックは、1 つ以上のイベントソース、条件付きルール、対応するアクションを定義します。ルールブックは YAML で記述され、**if-then** ルールを使用して特定のイベントを自動化されたアクションにリンクします。

Event-Driven Ansible の 5 つの用途

1. **問題をプロアクティブに修正する**：パフォーマンスの低下、構成ドリフト、セキュリティ脆弱性などの潜在的な問題を、運用やユーザーに影響が及ぶ前に自動的に特定して修正します。
2. **トラブルシューティングを迅速化する**：インシデントの種類と重大度、類似インシデントの頻度、適用される企業ポリシーなどの要素に基づく初期対応アクションを自動化することで、トラブルシューティングのアクティビティを単純化し、迅速化します。
3. **ユーザー管理要求に対応する**：ユーザーのロールや要求の種類などの情報に基づいて、パスワードの回復やアクセス管理などのユーザー管理要求を自動的に評価し、応答します。
4. **システムをプロアクティブに管理する**：構成ドリフトを監視して自動的にアップデートを適用し、インフラストラクチャ全体にわたって IT システムの期待される状態を維持します。
5. **システムをスケーリングおよび調整する**：インフラストラクチャ監視ツールによって報告されたデータ（ネットワーク帯域幅とレイテンシー、プロセッサとストレージの使用状況など）に基づき、ユーザーとアプリケーションの需要に応じてインフラストラクチャを自動的にスケーリングおよび調整します。



Event-Driven Ansible の利用法について詳しくは、以下のチェックリストを参照してください。

- ▶ **IT 戦略にイベント駆動型の自動化を含めるべき、その 5 つの理由**
- ▶ **Event-Driven Ansible の自動化でより多くの成果を生み出す 5 つの方法**

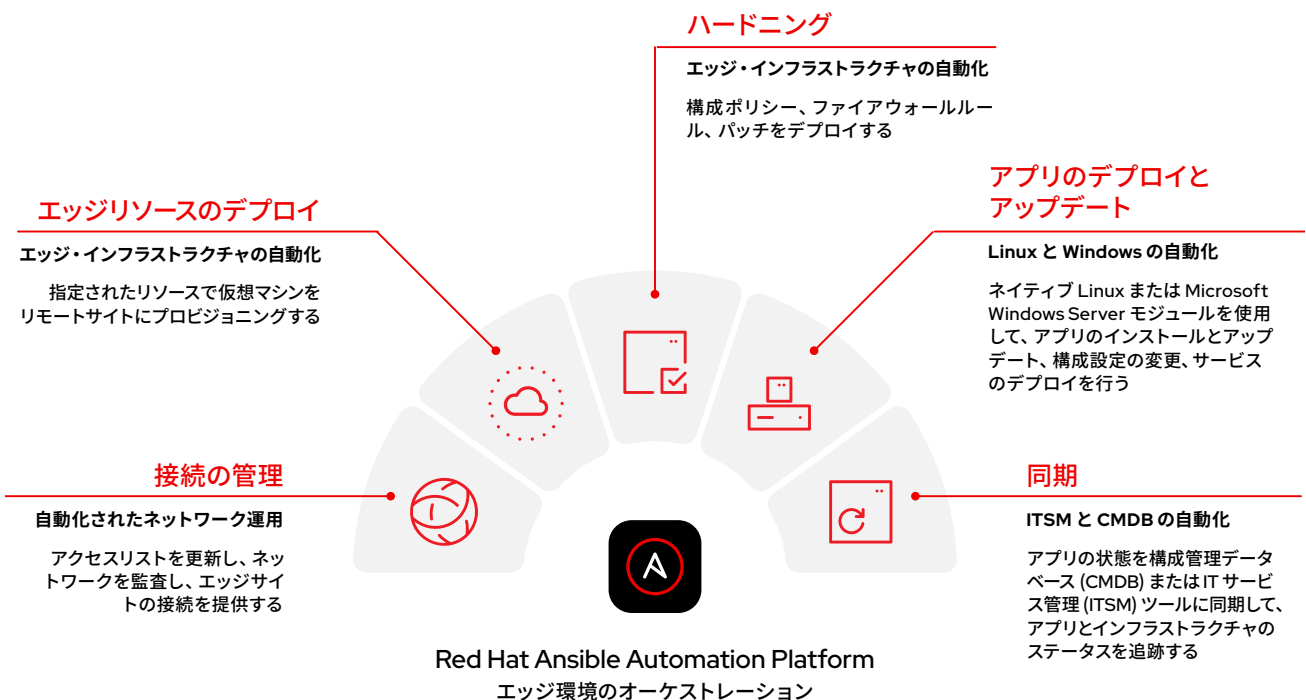
エッジで自動化をデプロイ

エッジコンピューティングは処理能力をデータソースに近づけるため、レイテンシーの影響を受けやすいアプリケーションのデプロイ、多数のデバイスからのデータ収集、データセンターやクラウドへの接続が失われた場合も動作できるサイトの構築が可能になります。**エッジでの自動化**は、一貫性のある標準化された方法で IT 環境全体の管理、オーケストレーション、保守を行うために役立ちます。

Red Hat Ansible Automation Platform は、コンテナ化を使用して環境全体に自動化を分散および実行し、運用チームがデータセンターからクラウド、エッジロケーションに至るまでの構成とデプロイを標準化できるよう支援します。IT ランドスケープが単一の一貫したビューで表示されるため、数千ものサイト、ネットワークデバイス、クラスタを管理できます。自動化へのこの統合アプローチにより、リソースが限られているエッジサイトで運用コストを削減し、よりスムーズなカスタマーエクスペリエンスを実現するのに役立ちます。



エッジ自動化の業界固有のユースケースについては、**エッジでの自動化：7 つの業界ユースケースと事例**を参照してください。



エッジでの自動化の6つのメリット

1. **セキュリティと安全性の重視**: 現場に技術者を派遣せずに、更新、パッチ、必要なメンテナンスを自動的に実行します。
2. **ダウンタイムの短縮**: ネットワーク管理を単純化し、ネットワーク障害を削減して、収益性を向上させます。
3. **効率性の向上**: パフォーマンスを向上させ、自動化された分析、監視、アラート機能で人的ミスを削減します。
4. **スケーラビリティの向上**: 構成をインフラストラクチャに一貫して適用し、エッジデバイスのスケーリングをスピードアップします。
5. **迅速な応答**: リアルタイムのデータとイベントに基づいて自動化されたワークフローにより、最適化されたユーザーエクスペリエンスを提供します。
6. **コンプライアンスの確保**: システムとアプリケーションがセキュリティおよび監査評価の定義どおりに動作し、規制要件と内部基準の両方を満たすようにします。

完全なクラウドワークフローのオーケストレーション

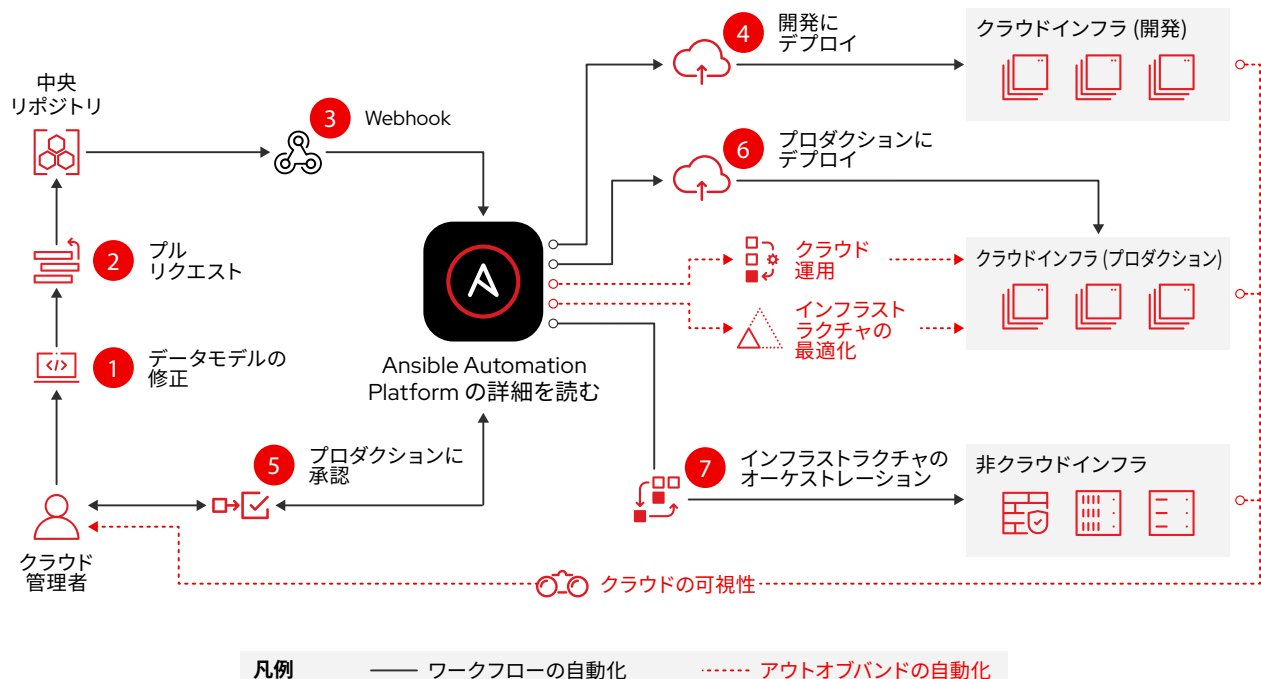
自動化によってオーケストレーションとワークフローが最適化され、オンサイトデータセンターからパブリッククラウド・インフラストラクチャまで、ハイブリッドおよびマルチクラウド環境全体の運用化を促します。クラウド自動化を使用すると、タスクを文書化、評価、体系化でき、ワークフローに繰り返し確実に組み合わせて、予測可能なビジネス成果を達成できます。クラウド自動化は、すべての IT およびクラウドドメインに対して一貫した運用フレームワークを作成する際にも役立ちます。

Red Hat Ansible Automation Platform により、クラウドリソースやサービスから、オペレーティングシステム、アプリケーション、セキュリティまで、ハイブリッドクラウド環境のすべての側面を自動化し、オーケストレーションできます。既存の自動化ツール、構成ツール、クラウドツールとプロセスを共通の言語で接続します。この結果、すべてのクラウドドメイン、プロセス、役割に対して一貫した運用フレームワークが作成され、自動化をターゲットエンドポイントの近くに配置できます。



クラウド環境を自動化する方法について詳しくは、**ハイブリッドクラウドを大規模に自動化**を参照してください。

この完全なクラウド自動化ワークフローの例では、Ansible Automation Platform を使用して、GitOps アプローチによってクラウドリソースとアプリケーション・ライフサイクルをオーケストレーションする方法を示します。



ワークフローの自動化

1. クラウド管理者がリソース定義または Playbook を修正します。
2. クラウド管理者が変更した定義または Playbook を中央リポジトリにコミットします。
3. Ansible Automation Platform Webhook 統合により、変更が検知され、必要な自動化があれば開始されます。
4. Ansible Automation Platform がクラウドリソースを開発環境に再デプロイします。
5. クラウド管理者が自動化されたプロダクションリクエストを承認します。
6. Ansible Automation Platform がクラウドリソースをプロダクション環境にデプロイします。
7. Ansible Automation Platform が、プロダクションデプロイに必要なその他のオフクラウドリソースをセットアップし、オーケストレーションします。

アウトオブバンドの自動化

- ▶ **クラウド運用**：Ansible Automation Platform は、修正やアップデートなど、必要に応じて Day 1 および Day 2 オペレーションを実行します。
- ▶ **インフラストラクチャの最適化**：Ansible Automation Platform は必要に応じてインフラストラクチャおよびリソースを最適化します。
- ▶ **クラウドの可視性**：Ansible Automation Platform はインフラストラクチャのスナップショットと取得し、必要に応じて可視性と情報を提供します。

組織全体で自動化を促進

組織規模の自動化はすぐに実現できるものではありません。また、自動化とは 0 か 100 かという問題でもありません。自動化を進めるには持続的な戦略が必要です。そして戦略の策定には、評価、計画、調整が必要です。

ビジネス目標を特定する

自動化の取り組みを、ビジネスの課題と目標に結びつけます。どの領域を自動化するか特定し、成功に対するトップダウンの要件の策定に役立ちます。たとえば、パッチ適用を自動化すると、システムのセキュリティと安定性が向上し、アップタイムを延長するというビジネスニーズを満たすことができます。

チーム間のコラボレーションを促進する

インセンティブを使用して組織でのコラボレーションを促進します。チーム間で調整することにより、全体を網羅した自動化ワークフローを作成でき、より高い価値を提供できます。他チームと連携すると、自動化に対する当事者意識や責任感を共有できるようにもなります。

組織全体で信頼を築く

信頼された自動化コンテンツを格納する一元的なリポジトリを構築します。各チームはそれぞれが専門とする領域の自動化コンテンツを作成し、他のチームが利用できるようにリポジトリに提供します。他のユーザーがコンテンツを迷わず使用できるように、スタッフは境界を付加できます。

知識と成功を共有する

中心的な関係者からなるチームを結成します。これは一般にコミュニティ・オブ・プラクティス (CoP) またはセンター・オブ・エクセレンス (CoE) などと呼ばれ、自動化のベストプラクティス、経験、達成を組織内で共有します。これらのチームは、自身の自動化を進めながら、他の人々の支援も行います。

自動化コンテンツを一元化する

組織全体のコラボレーション、ツール、コンテンツ向けに統一された基盤を提供する自動化プラットフォームを選択します。信頼できる単一の場所でツールとコンテンツを共有すると、チームは自動化を効率的に進められ、作業の重複を回避できます。

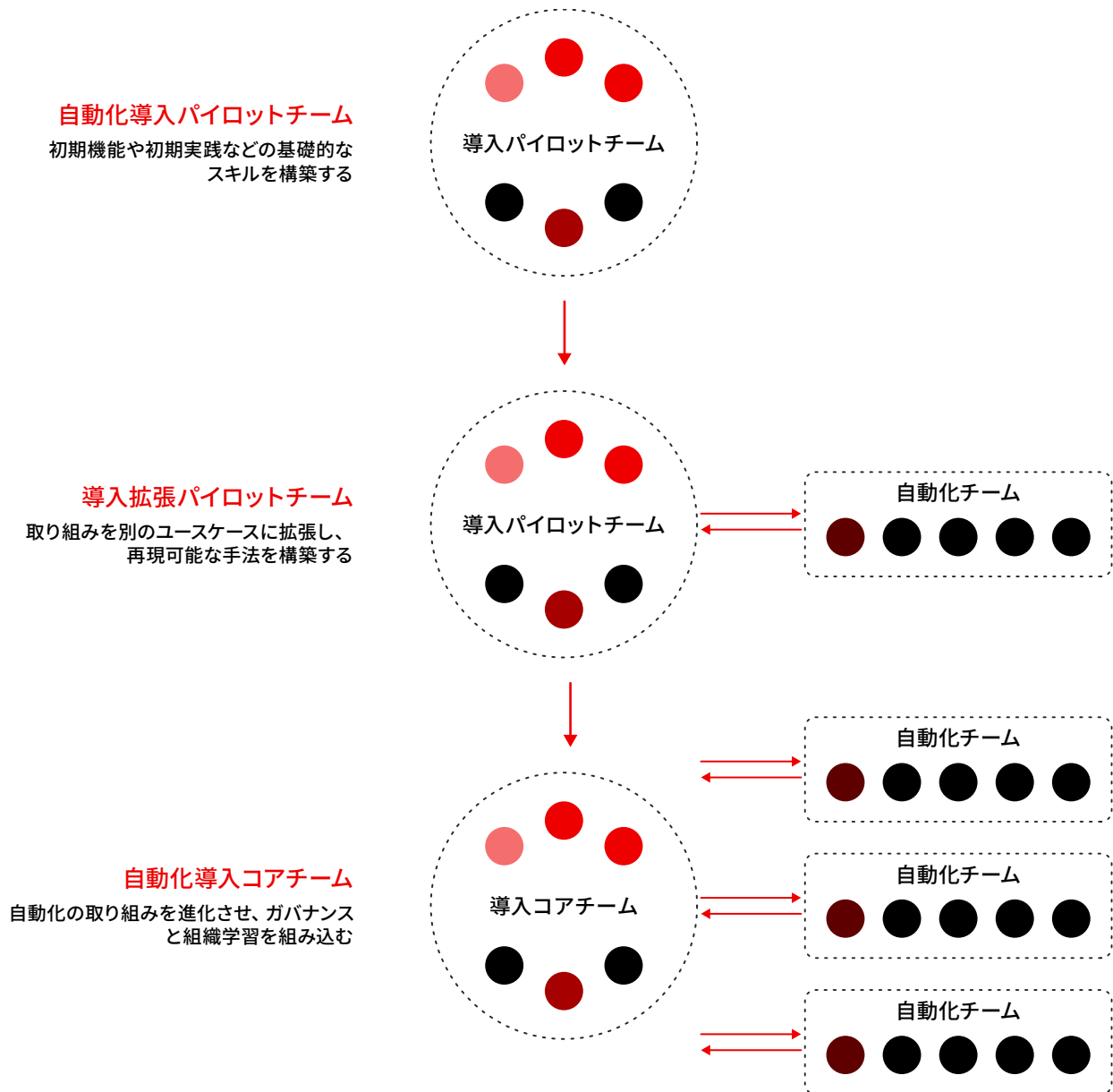
成功の定義

自動化の成功を測定する方法は一つではありません。各チームにそれぞれの特徴と展望があるからです。チームが持つ現行のスキルに見合った、それでいてスタッフが能力を伸ばせるような、現実的な目標を立てます。自動化の長期的な成功の例を示します。

- ▶ **導入**：自動化のビジョンから実施に至るまで組織全体で取り入れる。その際、シンプルさと知識の共有を重視する
- ▶ **アカウントビリティ**：組織の各メンバーが、各自の目標に対して責任を持つ
- ▶ **ガバナンス**：規定のプロセスで自動化の目標を達成し、繰り返し実現可能な結果を出す
- ▶ **セキュリティ**：シンプルなパイプライン、反復可能かつ再利用可能な実践、プロアクティブな脆弱性解決策、インシデントに対する自動化された調査と対応で実現する
- ▶ **標準**：組織およびチームの目標の達成に必要な、基盤と拡張性を提供する

詳細は、[組織を自動化する](#)を参照してください。

人材は、どのような全社的な取り組みでも中心的な存在で、自動化も例外ではありません。組織に自動化を導入するためには、ビジネス、ネットワーク、セキュリティ、運用、開発、インフラストラクチャを含むすべてのチームが協力し、新しい概念やスキルを学ぶ準備ができてい必要があります。自動化のベストプラクティスを組織規模で導入できるようサポートする COP または COE チームを作成しましょう。



凡例

● リードアーキテクト

● エンタープライズアーキテクト

● 他のチームメンバー：開発、ビジネス分析、セキュリティ、ネットワーキング、インフラストラクチャなど

● リードコンサルタント

● チームリード

関連資料および情報

Red Hat は、自動化への取り組みを開始し推進するために役立つ多くのリソースを提供しています。

無料のインタラクティブラボを試みる

実践的な学習シナリオで事前構成済みの環境が提供され、自動化がどのように機能するかを実験、練習、確認することができます。

製品ドキュメントを読む

リリースノート、インストールガイド、運用情報など、Ansible Automation Platform のドキュメントを確認できます。

グッドプラクティスを学ぶ

Ansible のグッドプラクティスガイドには、Red Hat の Ansible 実践者、コンサルタント、開発者などからの推奨事項がまとめられています。

無料のオンラインコースを受講する

Ansible Basics: Automation Technical Overview コースは、Ansible Automation Platform の概要を学ぶためのオンデマンド動画シリーズです。

必要なリソースを見つける

Red Hat は、学習の開始と継続に役立つ多くのリソースを提供しています。これらは「Ansible を始める」の Web ページにまとめられています。

Ansible コミュニティに参加する

Ansible コミュニティでは、IT の課題に関する議論やさまざまなソリューションでの共同作業が行われています。また、業界の仲間と交流したり、ソーシャルネットワークでつながったりすることもできます。