Red Hat OpenShift Service on AWS 4

概要

OpenShift Service on AWS ドキュメント

Last Updated: 2024-11-03

Red Hat OpenShift Service on AWS 4 概要

OpenShift Service on AWS ドキュメント

法律上の通知

概要

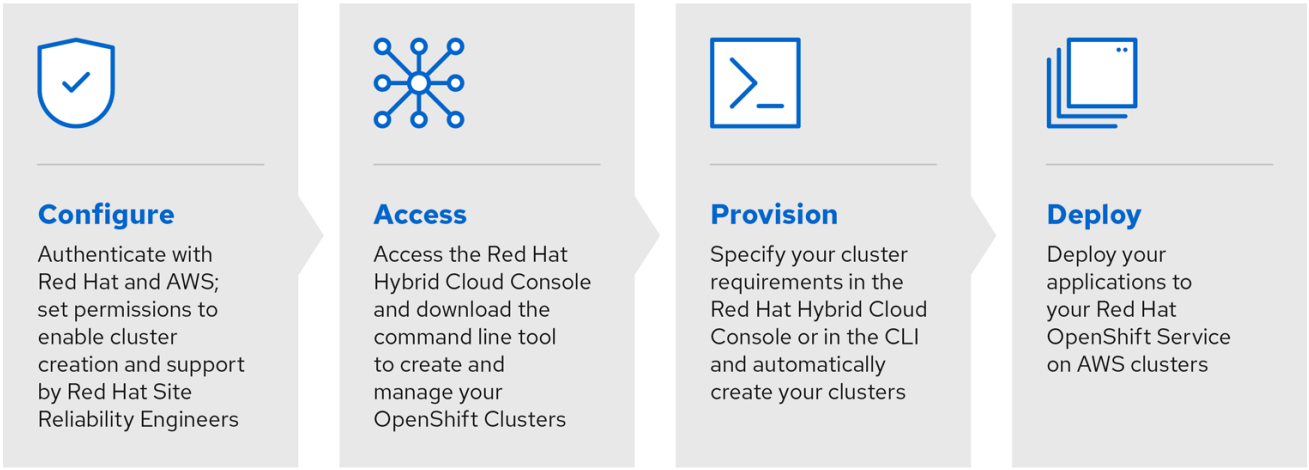
公式の OpenShift Service on AWS ドキュメントへようこそ。ここでは、OpenShift Service on AWS について学び、その機能について確認できます。

目次

第1章 RED HAT OPENSHIFT SERVICE ON AWS 4 ドキュメント

目次

公式の Red Hat OpenShift Service on AWS (ROSA) ドキュメントへようこそ。ここでは、ROSA について学び、その機能について確認できます。ROSA の概要、Red Hat OpenShift Cluster Manager およびコマンドラインインターフェイス (CLI) ツールを使用して ROSA を操作する方法、使用エクスペリエンス、Amazon Web Services (AWS) サービスとの統合について学習するには、ROSA の概要ドキュメント から始めてください。



ROSA ドキュメント内の移動には、左側のナビゲーションバーを使用します。

第2章 ROSA WITH HCP の詳細

Red Hat OpenShift Service on AWS (ROSA) with hosted control planes (HCP) は、効率性を重視したマネージド ROSA クラスターを作成するための低コストのソリューションを提供します。新しいクラスターをすばやく作成し、数分でアプリケーションをデプロイできます。

2.1. ROSA WITH HCP の主な機能

- ROSA with HCP は、最小限の 2 つのノードしか必要としないため、小規模なプロジェクトに最適でありながら、大規模なプロジェクトやエンタープライズをサポートするために拡張することもできます。
- 基盤となるコントロールプレーンインフラストラクチャーが完全に管理されている。API サーバー etcd データベースなどのコントロールプレーンコンポーネントは、Red Hat が所有する AWS アカウントでホストされます。
- プロビジョニング時間は約 10 分です。
- お客様はコントロールプレーンとマシンプールを個別にアップグレードできるため、アップグレード中にクラスター全体をシャットダウンする必要がありません。

2.2. ROSA WITH HCP の使用

次のセクションを使用して、ROSA with HCP の学習と使用に役立つコンテンツを見つけてください。

2.2.1. アーキテクト

ROSA with HCP について	ROSA with HCP のデプロイメントの計画	関連情報
アーキテクチャーの概要	バックアップと復元	ROSA with HCP ライフサイクル
ROSA with HCP アーキテクチャー		ROSA with HCP サービス定義
		サポート

2.2.2. クラスター管理者

ROSA with HCP について	ROSA with HCP のデプロイ	ROSA with HCP の管理	関連情報
ROSA with HCP アーキテクチャー	ROSA with HCP のインストール	サポート	OpenShift インタラクティブラーニングポータル
ストレージ	モニタリングの概要	ROSA with HCP ライフサイクル	
バックアップと復元			

2.2.3. 開発者

ROSA with HCP でのアプリケーション開発について	アプリケーションのデプロイ	関連情報
Red Hat Developer Web サイト	アプリケーションのビルドの概要	サポート
Red Hat OpenShift Dev Spaces (旧 Red Hat CodeReady Workspaces)	Operator の概要	
	イメージ	
	開発者向け CLI	

第3章 AWS STS と ROSA WITH HCP の説明

Hosted Control Plane (HCP) を備えた Red Hat OpenShift Service on AWS (ROSA) は、AWS Identity Access Management (IAM) 用の AWS (Amazon Web Services) Security Token Service (STS) を使用して、AWS アカウント内のリソースとやり取りするために必要な認証情報を取得します。

3.1. AWS STS 認証方法

ROSA with HCP の一環として、AWS アカウント内のインフラストラクチャーリソースを管理するために必要な権限を Red Hat に付与する必要があります。ROSA with HCP は、クラスターの自動化ソフトウェアに、AWS アカウント内のリソースへの限定的な短期アクセスを許可します。

STS メソッドでは、事前定義されたロールとポリシーを使用して、IAM ロールに一時的な最小限の権限を付与します。通常、認証情報は要求されてから 1 時間後に期限切れになります。有効期限が切れると、認証情報は AWS によって認識されなくなり、その認証情報を使用して実行される API 要求からアカウントにアクセスできなくなります。詳細は、AWS のドキュメント を参照してください。

ROSA with HCP クラスターごとに AWS IAM STS ロールを作成する必要があります。ROSA コマンドラインインターフェイス (CLI) (**rosa**) は STS ロールを管理し、ROSA 固有の AWS 管理ポリシーを各ロールにアタッチするのに役立ちます。CLI は、ロールを作成し、AWS 管理ポリシーをアタッチするためのコマンドとファイルを提供し、CLI が自動的にロールを作成してポリシーをアタッチできるようにするオプションも提供します。

3.2. AWS STS セキュリティー

AWS STS のセキュリティー機能には以下が含まれます。

- ユーザーが事前に作成する明示的かつ限定的なポリシーセット。
  - ユーザーは、プラットフォームに必要なすべての要求された権限を確認できます。
- サービスは、これらの権限以外の操作を一切行うことができません。
- 認証情報をローテーションしたり取り消したりする必要はありません。サービスは、操作を実行する必要が生じるたびに、1 時間以内に期限切れになる認証情報を取得します。
- 認証情報の有効期限により、認証情報の漏洩や再利用のリスクが軽減されます。

ROSA with HCP は、特定の分離された IAM ロールに、短期的なセキュリティー認証情報を使用して、クラスターソフトウェアコンポーネントに最小限の権限を付与します。認証情報は、AWS の API 呼び出しを実行する各コンポーネントおよびクラスターに固有の IAM ロールに関連付けられます。この方法は、クラウドサービスのリソース管理における最小権限と安全なプラクティスの原則に沿ったものです。

3.3. ROSA WITH HCP のコンポーネント

- **AWS インフラストラクチャー** - Amazon EC2 インスタンス、Amazon EBS ストレージ、ネットワークコンポーネントなど、クラスターに必要なインフラストラクチャー。クラウドリソース設定の詳細は、コンピュートノードでサポートされているインスタンスタイプと プロビジョニングされた AWS インフラストラクチャー を確認するには、AWS コンピュートタイプ を参照してください。
- **AWS STS** - 短期間の動的トークンを付与して、ユーザーに AWS アカウントのリソースを一時的に操作するために必要な権限を付与する方法。

- **OpenID Connect (OIDC)** - クラスター Operator が AWS で認証し、信頼ポリシーを通じてクラスターのロールを引き受け、必要な API 呼び出しを実行するために STS から一時的な認証情報を取得するためのメカニズム。

- **ロールとポリシー** - ROSA with HCP で使用するロールとポリシーは、アカウント全体のロールとポリシーと、Operator のロールとポリシーに分けられます。
ポリシーは、各ロールに対して許可されるアクションを決定します。個々のロールとポリシーの詳細は、STS を使用する ROSA クラスターの IAM リソースについて を参照してください。
信頼ポリシーの詳細は、ROSA IAM ロールリソース を参照してください。

  - アカウント全体のロールは次のとおりです。
    - ManagedOpenShift-Installer-Role
    - ManagedOpenShift-Worker-Role
    - ManagedOpenShift-Support-Role
  - アカウント全体の AWS 管理ポリシーは次のとおりです。
    - ROSAInstallerPolicy
    - ROSAWorkerInstancePolicy
    - ROSASRESupportPolicy
    - ROSAIngressOperatorPolicy
    - ROSAAmazonEBSCSIDriverOperatorPolicy
    - ROSACloudNetworkConfigOperatorPolicy
    - ROSAControlPlaneOperatorPolicy
    - ROSAImageRegistryOperatorPolicy
    - ROSAKMSProviderPolicy
    - ROSAKubeControllerPolicy
    - ROSAManageSubscription
    - ROSANodePoolManagementPolicy

注記

以下にリストされている特定のポリシーは、クラスター Operator ロールによって使用されます。Operator ロールは既存のクラスター名に依存しており、アカウント全体のロールと同時に作成できないため、2 番目のステップで作成されます。

  - Operator のロールは次のとおりです。
    - <operator_role_prefix>-openshift-cluster-csi-drivers-ebs-cloud-credentials
    - <operator_role_prefix>-openshift-cloud-network-config-controller-cloud-credentials

- <operator_role_prefix>-openshift-machine-api-aws-cloud-credentials
- <operator_role_prefix>-openshift-cloud-credential-operator-cloud-credentials
- <operator_role_prefix>-openshift-image-registry-installer-cloud-credentials
- <operator_role_prefix>-openshift-ingress-operator-cloud-credentials

- 信頼ポリシーは、アカウント全体のロールと Operator のロールごとに作成されます。

3.4. ROSA WITH HCP クラスターのデプロイ

ROSA with HCP クラスターをデプロイするには、次の手順を実行します。

1. アカウント全体のロールを作成します。
2. Operator ロールを作成します。
3. Red Hat は AWS STS を使用して、AWS が対応する AWS 管理 Operator ポリシーを作成してアタッチできるようにするために必要な権限を AWS に送信します。
4. OIDC プロバイダーを作成します。
5. クラスターを作成します。

クラスターの作成プロセス中に、ROSA CLI は必要な JSON ファイルを作成し、必要なコマンドを出力します。必要に応じて、ROSA CLI でコマンドを実行することもできます。

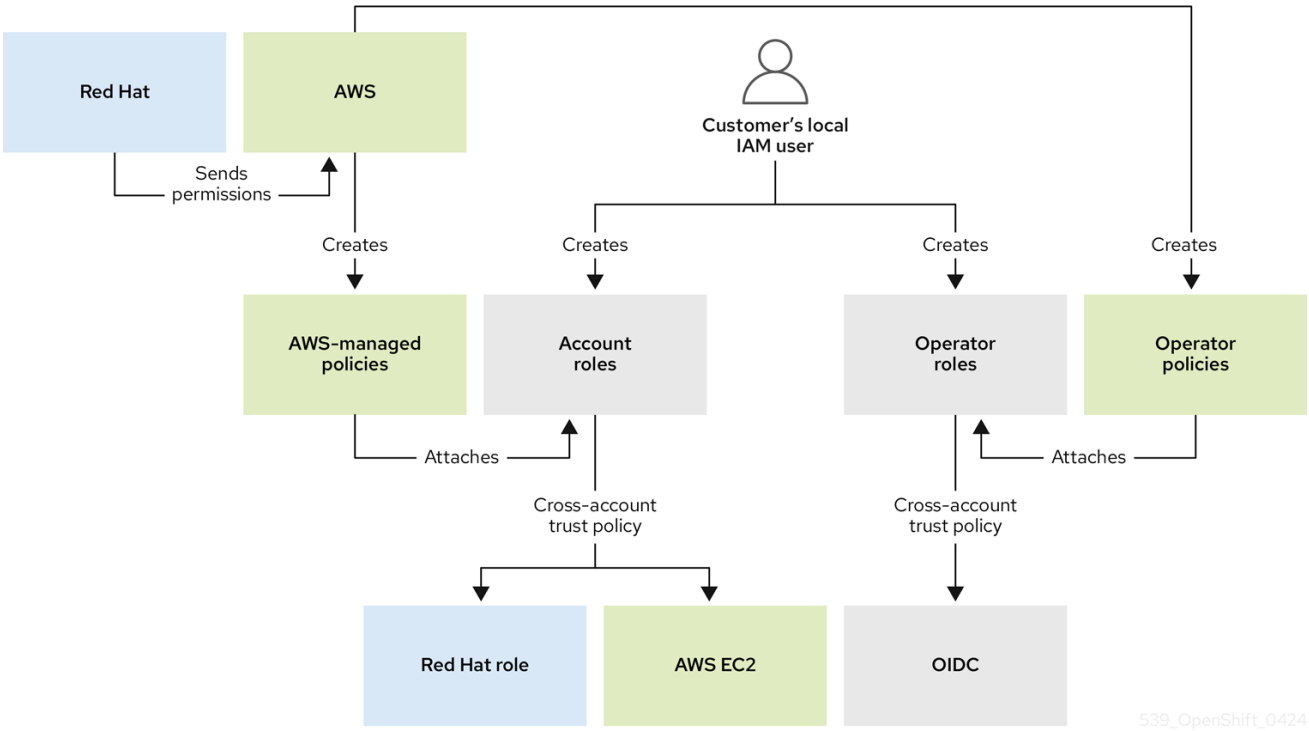
ROSA CLI は自動的にロールを作成することも、**--mode manual** または **--mode auto** フラグを使用して手動で作成することもできます。デプロイメントの詳細は、カスタマイズによるクラスターの作成 を参照してください。

3.5. ROSA WITH HCP ワークフロー

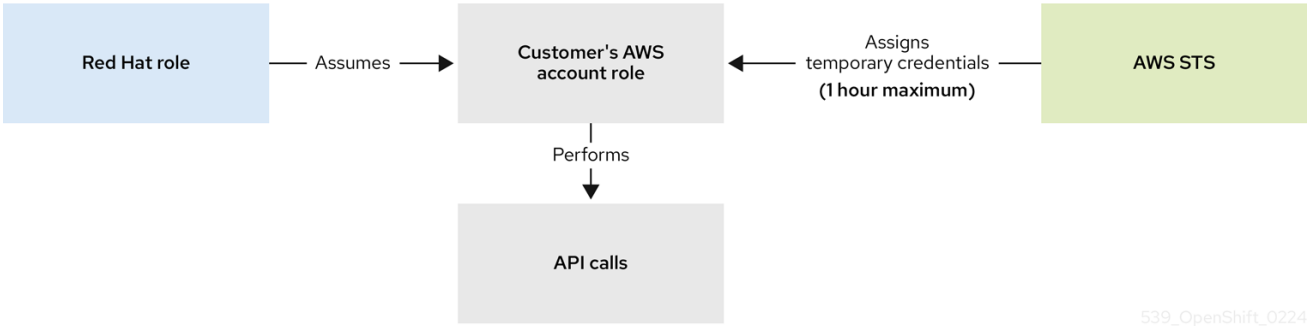
ユーザーは、必要なアカウント全体のロールを作成します。ロールの作成時に、クロスアカウント信頼ポリシーという信頼ポリシーが作成されます。このポリシーは、Red Hat 所有のロールがこのロールを引き受けることを許可するものです。また、EC2 サービス用の信頼ポリシーも作成されます。このポリシーは、EC2 インスタンス上のワークロードがロールを引き受けて認証情報を取得することを許可するものです。AWS は各ロールに対応するアクセス許可ポリシーを割り当てます。

アカウント全体のロールとポリシーを作成した後、ユーザーはクラスターを作成できます。クラスターの作成が開始すると、ユーザーは Operator のロールを作成し、クラスター Operator が AWS API 呼び出しを行えるようにします。これらのロールを、以前に作成された対応する権限ポリシーと、OIDC プロバイダーの信頼ポリシーに割り当てます。Operator ロールは、AWS リソースへのアクセスが必要な Pod を最終的に表すという点で、アカウント全体のロールとは異なります。ユーザーは IAM ロールを Pod に割り当てることができないため、Operator (ひいては Pod) が必要なロールにアクセスできるように、OIDC プロバイダーを使用して信頼ポリシーを作成する必要があります。

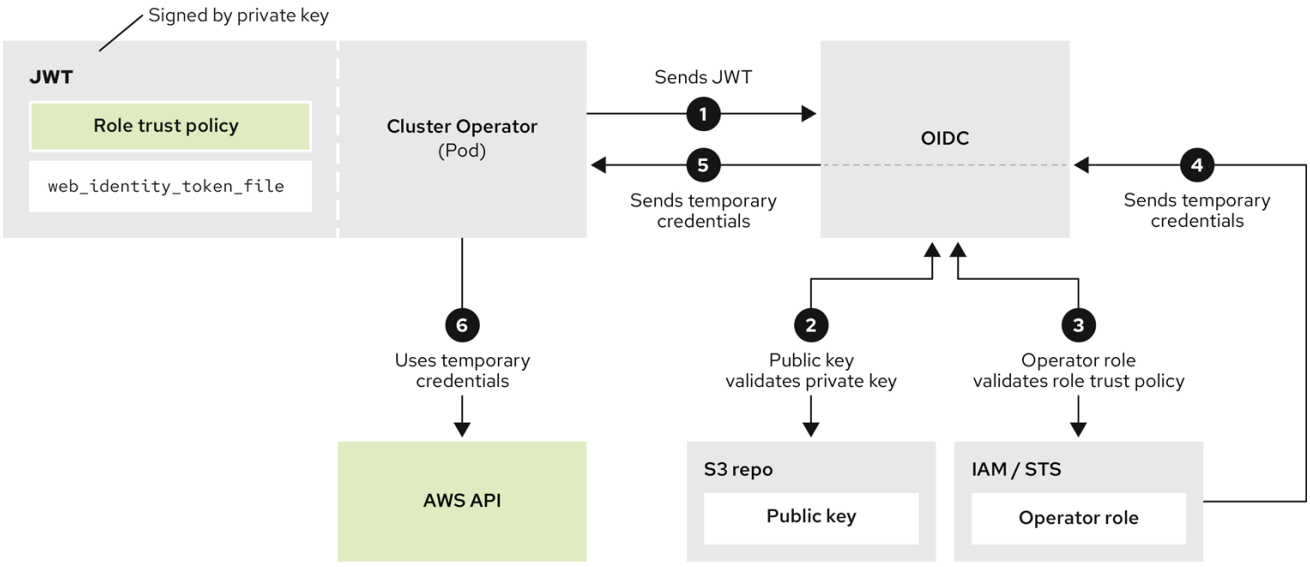
対応する権限ポリシーにロールを割り当てたら、最後のステップとして OIDC プロバイダーを作成します。



新しいロールが必要な場合、現在 Red Hat のロールを使用しているワークロードが AWS アカウントのロールを引き受け、AWS STS から一時的な認証情報を取得し、引き受けたロールの権限ポリシーに従ってユーザーの AWS アカウント内の API 呼び出しを使用してアクションの実行を開始します。認証情報は一時的なもので、有効期間は最大 1 時間です。



Operator は、次のプロセスを使用して、タスクを実行するために必要な認証情報を取得します。各 Operator には、Operator のロール、権限ポリシー、および OIDC プロバイダーの信頼ポリシーが割り当てられます。Operator は、ロールとトークンファイル (**web_identity_token_file**) を含む JSON Web トークンを OIDC プロバイダーに渡すことによってロールを引き受けます。OIDC プロバイダーは署名された鍵を公開鍵で認証します。公開鍵はクラスターの作成時に作成され、S3 バケットに保存されます。次に、Operator は、署名されたトークンファイル内のサブジェクトがロール信頼ポリシー内のロールと一致することを確認します。このロールは、OIDC プロバイダーが許可されたロールのみを取得できるようにするためのものです。その後、OIDC プロバイダーが一時的な認証情報を Operator に返し、Operator が AWS API 呼び出しを実行できるようにします。視覚的な説明は、次の図を参照してください。


Signed by private key
JWT
Role trust policy
web_identity_token_file
Cluster Operator
(Pod)
Sends JWT
1
5
Sends temporary
credentials
OIDC
4
Sends temporary
credentials
6
Uses temporary
credentials
2
Public key
validates private key
3
Operator role
validates role trust policy
AWS API
S3 repo
Public key
IAM / STS
Operator role
629_OpenShift_0424

第4章 法的通知