AWS EKS クラスタの管理

Cisco Container Platform と Amazon Web Services（AWS）を統合することによって、シスコベースのオンプレミス環境と AWS クラウドの両方にコンテナ化されたアプリケーションを導入し、実行することができます。

この章は、次の内容で構成されています。

AWS EKS クラスタを設定するための前提条件

AWS EKS クラスタを設定するための前提条件は次のとおりです。

Amazon リソースの要件

次の表に、Cisco Container Platform の導入要件によっては引き上げる必要がある Amazon リソースのデフォルトの制限について説明します。

（注） 特定のリソースの制限を引き上げるには、Amazon のサポートに連絡する必要があります。

Amazon のリソース	デフォルトの制限	説明
各 AWS アカウントのネットワーク アドレス変換（NAT）	14	各 EKS クラスタは 3 つの NAT ゲートウェイを使用します。デフォルトの設定では、クラスタは 4 つまでに制限されています。

Amazon のリソース	デフォルトの制限	説明
各 AWS アカウントの Amazon 仮想プライベート クラウド（Amazon VPC）	3	各テナント クラスタには個別の Amazon VPC が必要です。
各 AWS アカウントの Amazon Elastic Container Service for Kubernetes（Amazon EKS）クラスタ	3	（注） Amazon EKS クラスタの制限に加えた変更が更新されるのは木曜日のみです。
各地域の Elastic IP アドレス	5	各 EKS クラスタは 3 つの Elastic IP アドレスを使用します。詳細については、「Amazon VPC の制限」を参照してください。
各地域のインターネット ゲートウェイ	5	各 EKS クラスタは 1 つのインターネット ゲートウェイを使用します。

Amazon アカウントへの AMI ファイルの追加

Cisco Container Platform は特定の AMI（Amazon マシン イメージ）ファイルを製品のリリースごとに生成します。AMI ファイルは、互換性のあるパッケージを使用してテナント クラスタを作成できるようにします。

AMI ファイルを Amazon アカウントで使用できるようにするには、12 桁の Amazon アカウント ID を含めてサポート ケースを提出する必要があります。AMI が Amazon アカウントで使用可能になると、通知が届きます。

AWS ロールの作成

ステップ 1 https://console.aws.amazon.com/iam/ で AWS 管理コンソールにログインし、IAM コンソールを開きます。

ステップ 2 IAM コンソールのナビゲーション ウィンドウで、[Roles] をクリックした後、[Create role] をクリックします。

ステップ 3 [Select type of trusted entity] の下にある [Another AWS account] をクリックします。

ステップ 4 [Account ID] フィールドに **AWS のアカウント ID** を入力した後、[Next] をクリックします。

Cisco Container Platform が EKS クラスタ作成時にロール ARN を使用できるように、AWS アカウント番号は信頼できるエンティティである必要があります。

ステップ 5 許可ポリシーと許可境界を選択する画面をスキップし、[Next] をクリックします。

ステップ 6 キーと値のペアとして選択したタグを追加することでロールにメタデータを追加し、[Next] をクリックします。

ステップ 7 [Role name] フィールドに、ルールの名前として `k8s-ccp-user` を入力するか、または任意の別の名前を入力します。

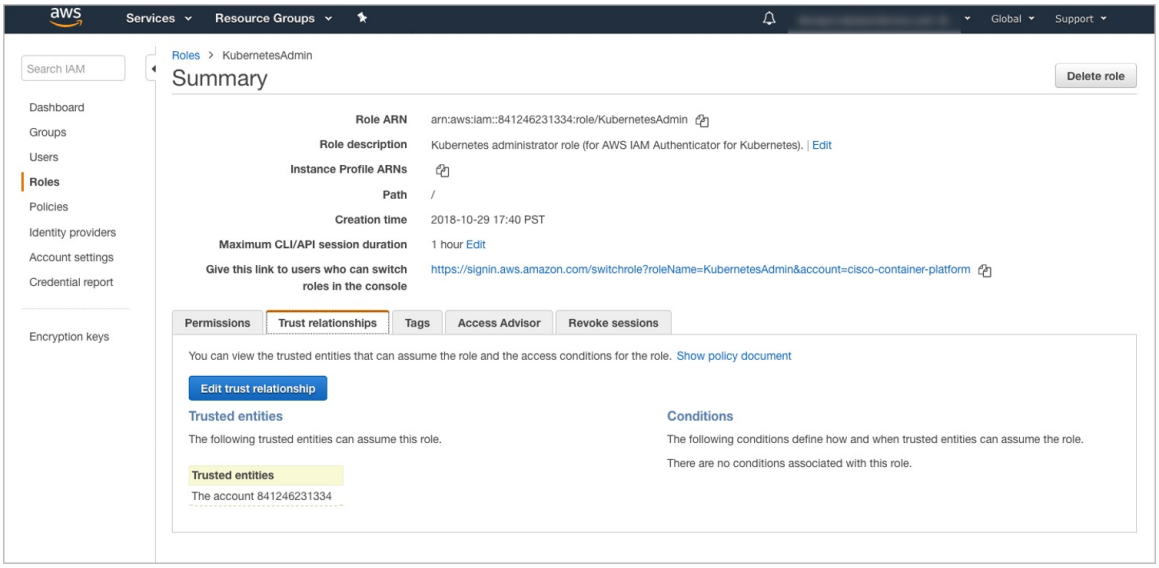
ステップ 8 [Description] フィールドに任意の説明を入力し、[Create role] をクリックします。

ステップ 9 ロールを作成したら、作成したロールまで移動し、そのロールの次の詳細情報を確認します。

a) [Permissions] タブをクリックして権限が設定されていないことを確認します。

b) [Trust Relationships] タブをクリックして、ロール ARN の作成時に入力した AWS アカウントに信頼関係があることを確認します。

図 *1 : AWS* 管理コンソール：*[Trust Relationships]* タブ



AWS アカウント ポリシーの要件

プロバイダーの権限

AWS プロバイダー アカウントがルート アカウントでない場合は、EKS および EC2 のリソースの作成に必要な権限がそのアカウントにあることを確認する必要があります。

必要な最小権限は、aws-provider-policy.json サンプル ファイルに含まれています。このファイルを作成してインポートし、必要な権限を設定することができます。

aws-provider-policy.json サンプル ファイル

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:*",
                "elasticloadbalancing:*",
```

```
                "autoscaling:*",
                "ec2:*",
                "eks:*",
                "ecr:*",
                "ecs:*",
                "s3:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:List*",
                "iam:Get*",
                "iam:PassRole",
                "iam:AddRoleToInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:CreateRole",
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:AttachRolePolicy",
                "iam:DetachRolePolicy",
                "iam:PutRolePolicy",
                "iam:*AccessKey*",
                "iam:*MFA*"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon IAM 認証

デフォルトでは AWS IAM のアイデンティティを使用して EKS クラスタを認証し、接続を確立します。Cisco Container Platform は AWS IAM アイデンティティを使用してオンプレミス クラスタを認証するのに AWS IAM 認証を使用します。この認証は、一貫性があり、統一されたアイデンティティ スキームをオンプレミス クラスタと AWS EKS クラスタの両方に提供します。

AWS IAM オーセンティケータは、クライアントとサーバの両方の機能を実現します。クライアント側では、オーセンティケータが署名前の URL を生成し、トークン化し、アイデンティティを検証するためにサーバ側に送信します。クライアントはワークステーションにインストールされた Go バイナリであり、Kubernetes クラスタとやり取りするたびに kubectl によって透過的に呼び出されます。サーバ側は Kubernetes マスター ノード上で DaemonSet として実行する AWS IAM オーセンティケータのコンテナ化されたインスタンスです。これによって、検証を実行するための AWS Secure Token Service（STS）とのやり取りが実行されます。Cisco Container Platform は最初のサーバ側の設定に対応し、管理者ユーザがダウンロードするための事前設定済みの `Kubeconfig` ファイルを提供します。

（注） kubectl を使用してクラスタとやり取りしているときに `$PATH` 内で AWS IAM オーセンティケータが使用可能であることを確認する必要があります。

共通アイデンティティの有効化

Cisco Container Platform の Web インターフェイス内で、ユーザはクラスタに対して共通のアイデンティティ スキームを選択することができます。クラスタがプロビジョニングされた後に、共通 RBAC ポリシーを適用できます。

（注） IAM 認証が EKS クラスタに明示的に使用できるようになります。Cisco Container Platform はユーザに提供された IAM ロールを EKS クラスタにマッピングすることができ、オンプレミス クラスタに IAM 認証を設定します。

AWS EKS クラスタの作成

始める前に

- Amazon プロバイダーのプロファイルが設定されていることを確認します。詳細については、Amazon プロバイダー プロファイルの追加を参照してください。
- 必要な AMI ファイルがアカウントに追加されていることを確認します。詳細については、Amazon アカウントへの AMI ファイルの追加 （2 ページ）を参照してください。
- AWS EKS クラスタの作成に Cisco Container Platform を使用するための AWS IAM ロールが作成されていることを確認します。詳細については、AWS ロールの作成 （2 ページ）を参照してください。

ステップ 1 左側のペインで、[Clusters] をクリックした後、[AWS] タブをクリックします。

ステップ 2 [NEW CLUSTER] をクリックします。

ステップ 3 [Basic Information] 画面で、次の情報を入力します。

a) [INFRASTUCTURE PROVIDER] ドロップダウン リストで、適切な Amazon アカウントに関連するプロバイダーを選択します。

b) [AWS REGION] ドロップダウン リストで、適切な AWS の地域を選択します。

（注） すべての地域で EKS がサポートされているわけではありません。サポートされている地域を選択してください。現時点では、**us-west-2** 地域と**us-east-1** 地域のみが Cisco Container Platform でサポートされています。

c) [KUBERNETES CLUSTER NAME] フィールドにクラスタの名前を入力します。

d) [NEXT] をクリックします。

ステップ 4 [Node Configuration] 画面で、次の情報を入力します。

a) [INSTANCE TYPE] ドロップダウン リストで、クラスタのインスタンス タイプを選択します。

b) [MACHINE IMAGE] ドロップダウン リストで、適切な Cisco Container Platform Amazon マシン イメージ（AMI）ファイルを選択します。

Amazon アカウントに AMI ファイルを追加するには、Amazon アカウントへの AMI ファイルの追加 （2 ページ）を参照します。

c) [WORKER COUNT] フィールドに、ワーカー ノードの適切な数を入力します。

d) [SSH PUBLIC KEY] ドロップダウン リストで、適切な認証キーを選択します。
このフィールドは任意です。トラブルシューティングのためにワーカー ノードへの ssh が必要な場合に必要です。公開キーに Ed25519 または ECDSA 形式が使用されていることを確認します。

注：RSA と DSA は安全性が低い形式であり、シスコではこれらの形式を使用しないようにしています。

e) [IAM ACCESS ROLE ARN] フィールドに、Amazon リソース名（ARN）情報を入力します。

（注） デフォルトでは、Amazon EKS クラスタの作成時に指定した AWS クレデンシャル、つまり、インフラストラクチャ プロバイダーに設定されているクレデンシャルが `Kubernetes cluster-admin ClusterRole` にマップされます。デフォルトの `ClusterRoleBinding` はクレデンシャルを `system:masters` グループにバインドします。そのため、IAM アイデンティティのホルダーへのスーパーユーザ アクセス権が付与されます。[IAM ACCESS ROLE ARN] フィールドでは、追加の AWS IAM ロール、またはクラスタの管理制御も付与されている IAM ユーザの ARN を指定できます。

f) [NEXT] をクリックします。

ステップ 5 [VPC Configuration] 画面で、次の情報を入力します。

a) [SUBNET CIDR] フィールドに、クラスタのサブネット CIDR 全体の値を入力します。

b) [PUBLIC SUBNET CIDR] フィールドの別の行にクラスタの値を入力します。

c) [PRIVATE SUBNET CIDR] フィールドの別の行にクラスタの値を入力します。

ステップ 6 [Summary] 画面でクラスタ情報を確認した後、[FINISH] をクリックします。

クラスタの作成に最大 20 分かかることがあります。クラスタの作成ステータスは [Clusters] 画面で監視できます。

（注） 「Could not get token: AccessDenied」というエラー メッセージが表示された場合は、AWS アカウントがロール ARN にとって信頼できるエンティティでないことを示しています。

信頼できるエンティティとしての AWS アカウントの追加については、AWS ロールの作成 （2 ページ）を参照してください。

AWS EKS クラスタのスケーリング

EKS クラスタは、実行するワークロードの需要に基づいてワーカー ノードを追加または削除することでスケーリングできます。

ステップ 1 右側のペインで [EDIT] をクリックします。
[Edit Cluster] ダイアログボックスが表示されます。

ステップ 2 [INSTANCE TYPE] ドロップダウン リストで、クラスタのインスタンス タイプを選択します。

ステップ 3 [MACHINE IMAGE] ドロップダウン リストで、適切な Cisco Container Platform Amazon マシン イメージ（AMI）ファイルを選択します。
Amazon アカウントに AMI ファイルを追加するには、Amazon アカウントへの AMI ファイルの追加 （2 ページ）を参照します。

ステップ 4 必要に応じて、[WORKER COUNT] フィールドで作業ノードの数を変更します。

ステップ 5 [UPDATE] をクリックします。

AWS EKS クラスタの削除

始める前に

AWS EKS クラスタを削除すると、コンテナとそれに関連付けられているデータが削除されるため、削除するクラスタが使用中でないことを確認します。

ステップ 1 左側のペインで、[Clusters] をクリックした後、[EKS Clusters] タブをクリックします。

ステップ 2 [ACTIONS] 列の下に表示されたドロップダウン リストで、削除するクラスタに対して [Delete] を選択します。

ステップ 3 確認ダイアログボックスで [DELETE] をクリックします。
AWS EKS クラスタを削除した後、クラスタ リソースが解放されるまでに約 10 分かかります。