Red Hat Advanced Cluster Management for Kubernetes 2.3

サービス

サポートされるサービスをクラスターに追加する方法については以下を参照してください。

Last Updated: 2023-06-02

Red Hat Advanced Cluster Management for Kubernetes 2.3 サービス

サポートされるサービスをクラスターに追加する方法については以下を参照してください。

法律上の通知

概要

サポートされるサービスをクラスターに追加する方法については以下を参照してください。

目次

第1章 サービスの概要

Red Hat Advanced Cluster Management for Kubernetes で使用するサービスを追加して、一部のパフォーマンスエリアを向上できます。サービスは、マネージドクラスター 1 台または複数台で実行されます。

以下のセクションでは、Red Hat Advanced Cluster Management で利用可能なサービスを概説しています。

- コンソールを使用した Submariner のデプロイ
- API を使用した Submariner のデプロイ
- Submariner のサービス検出の有効化

1.1. SUBMARINER ネットワークサービス (テクノロジープレビュー)

submariner-addon コンポーネントは **テクノロジープレビュー** 機能です。

Submariner は、Red Hat Advanced Cluster Management for Kubernetes で使用可能なオープンソースツールで、お使いの環境 (オンプレミスまたはクラウドのいずれか) 内の 2 つ以上の Kubernetes クラスターにわたる Pod 間で直接ネットワークを提供できます。Submariner の詳細は、Submariner を参照してください。

以下の環境でホストされる OpenShift Container Platform クラスターで Submariner を有効にできます。

- Amazon Web Services (AWS)
- Google Cloud Platform
- Microsoft Azure
- IBM Cloud
- Red Hat OpenShift Dedicated
- VMware vSphere

Red Hat Advanced Cluster Management for Kubernetes は、ハブクラスターを使用してお使いの環境にデプロイできる Submariner コンポーネントを提供します。

- 前提条件
- Submariner をデプロイする一部のホストの準備
  - Microsoft Azure で Submariner をデプロイする準備
  - IBM Cloud で Submariner をデプロイする準備
  - Red Hat OpenShift Dedicated で Submariner をデプロイする準備

1.1.1. 前提条件

Submariner を使用する前に、以下の前提条件があることを確認します。

- Red Hat OpenShift Container Platform バージョン 4.6 以降および Kubernetes バージョン 1.19 以降で実行している Red Hat Advanced Cluster Management ハブクラスター。
- **cluster-admin** のパーミッションを使用してハブクラスターにアクセスするための認証情報。
- (Kubernetes バージョン 1.17 以降を使用する) OpenShift Container Platform バージョン 4.4 以降で実行している 2 つ以上の OpenShift Container Platform マネージドクラスターは、Red Hat Advanced Cluster Management ハブクラスターによって管理されます。
- 重複しないクラスター間の Pod および Service Classless Inter-Domain Routing (CIDR)。
- ゲートウェイノード間で IP 接続を設定している。2 つのクラスターを接続する場合に、最低でも 1 つのクラスターには、ゲートウェイノード専用のパブリックまたはプライベート IP アドレスを使用してゲートウェイノードにアクセスできる必要があります。詳細は、Submariner Nat Traversal を参照してください。
- 各マネージドクラスターのすべてのノードにおけるファイアウォール設定は、両方の方向で 4800/UDP が許可されている。
- ゲートウェイノードのファイアウォール設定では、入力 8080/TCP が許可されているため、クラスター内の他のノードがアクセスできます。
- UDP/4500、およびゲートウェイノード上の IPsec トラフィックに使用されるその他のポート用にファイアウォール設定が開いています。
  注記: これは、クラスターが AWS または Google Cloud Platform 環境にデプロイされる際に自動的に設定されますが、他の環境のクラスター用に手動で設定し、プライベートクラウドを保護するファイアウォール用に設定する必要があります。

表1.1 Submariner の必須ポート

Name	デフォルト値	カスタマイズ可能
IPsec NATT	4500/UDP	はい
VXLAN	4800/UDP	いいえ
Submariner メトリクスポート	8080/UDP	いいえ

前提条件の詳細は、Submariner の前提条件 を参照してください。

1.1.2. Submariner をデプロイする一部のホストの準備

Microsoft Azure、IBM Cloud または Red Hat OpenShift Dedicated に Red Hat Advanced Cluster Management for Kubernetes で Submariner をデプロイする前に、接続用にホスト環境でクラスターを準備する必要があります。要件はホスティング環境によって異なるため、ホスティング環境の手順に従います。

1.1.2.1. Microsoft Azure で Submariner をデプロイする準備

Submariner コンポーネントをデプロイするように Microsoft Azure でクラスターを準備するには、以下の手順を実行します。

1. Microsoft Azure 環境で受信および送信のファイアウォールルールを作成し、IP セキュリティー NAT トラバーサルポート (デフォルトでは 4500/UDP) を開き、Submariner 通信を有効にします。

```
# create inbound nat rule
$ az network lb inbound-nat-rule create --lb-name <lb-name> \
--resource-group <res-group> \
--name <name> \
--protocol Udp --frontend-port <ipsec-port> \
--backend-port <ipsec-port> \
--frontend-ip-name <frontend-ip-name>

# add your vm network interface to the created inbound nat rule
$ az network nic ip-config inbound-nat-rule add \
--lb-name <lb-name> --resource-group <res-group> \
--inbound-nat-rule <nat-name> \
--nic-name <nic-name> --ip-config-name <pipConfig>
```

lb-name は、ロードバランサー名に置き換えます。

res-group は、リソースグループ名に置き換えます。

nat-name は、ロードバランシングの受信 NAT ルール名に置き換えます。

ipsec-port は、実際の IPsec ポートに置き換えます。

pipConfig は、クラスターのフロントエンド IP 設定名に置き換えます。

nic-name は、実際のネットワークインターフェイスカード (NIC) 名に置き換えます。

2. Submariner ゲートウェイメトリクス要求を転送する負荷分散の受信 NAT ルールを 1 つ作成します。

```
# create inbound nat rule
$ az network lb inbound-nat-rule create --lb-name <lb-name> \
--resource-group <res-group> \
--name <name> \
--protocol Tcp --frontend-port 8080 --backend-port 8080 \
--frontend-ip-name <frontend-ip-name>

# add your vm network interface to the created inbound nat rule
$ az network nic ip-config inbound-nat-rule add \
--lb-name <lb-name> --resource-group <res-group> \
--inbound-nat-rule <nat-name> \
--nic-name <nic-name> --ip-config-name <pipConfig>
```

lb-name は、ロードバランサー名に置き換えます。

res-group は、リソースグループ名に置き換えます。

nat-name は、ロードバランシングの受信 NAT ルール名に置き換えます。

pipConfig は、クラスターのフロントエンド IP 設定名に置き換えます。

nic-name は、実際のネットワークインターフェイスカード (NIC) 名に置き換えます。

3. Azure でネットワークセキュリティーグループ {NSG) セキュリティールールを作成し、Submariner の NAT トラバーサルポート (デフォルトでは 4500/UDP) を開きます。

```
$ az network nsg rule create --resource-group <res-group> \
--nsg-name <nsg-name> --priority <priority> \
--name <name> --direction Inbound --access Allow \
--protocol Udp --destination-port-ranges <ipsec-port>

$ az network nsg rule create --resource-group <res-group> \
--nsg-name <nsg-name> --priority <priority> \
--name <name> --direction Outbound --access Allow \
--protocol Udp --destination-port-ranges <ipsec-port>
Replace `res-group` with your resource group name.
+
Replace `nsg-name` with your NSG name.
+
Replace `priority` with your rule priority.
+
Replace `name` with your rule name.
+
Replace `ipsec-port` with your IPsec port.
```

4. NSG ルールを作成し、4800/UDP ポートを開き、ワーカーノードおよびマスターノードから Submariner Gateway ノードに Pod トラフィックをカプセル化します。

```
$ az network nsg rule create --resource-group <res-group> \
--nsg-name <nsg-name> --priority <priority> \
--name <name> --direction Inbound --access Allow \
--protocol Udp --destination-port-ranges 4800 \

$ az network nsg rule create --resource-group <res-group> \
--nsg-name <nsg-name> --priority <priority> \
--name <name> --direction Outbound --access Allow \
--protocol Udp --destination-port-ranges 4800
```

res-group は、リソースグループ名に置き換えます。

nsg-name を、実際の NSG 名に置き換えます。

priority を、ルールの優先度に置き換えます。

name を、実際のルール名に置き換えます。

5. NSG ルールを作成して 8080/TCP ポートを開き、Submariner ゲートウェイノードからメトリクスサービスをエクスポートします。

```
$ az network nsg rule create --resource-group <res-group> \
--nsg-name <nsg-name> --priority <priority> \
--name <name> --direction Inbound --access Allow \
--protocol Tcp --destination-port-ranges 8080 \

$ az network nsg rule create --resource-group <res-group> \
--nsg-name <nsg-name> --priority <priority> \
--name <name> --direction Outbound --access Allow \
--protocol Udp --destination-port-ranges 8080
```

res-group は、リソースグループ名に置き換えます。

nsg-name を、実際の NSG 名に置き換えます。

priority を、ルールの優先度に置き換えます。

name を、実際のルール名に置き換えます。

6. クラスターのワーカーノードに **submariner.io/gateway=true** というラベルを付けます。

1.1.2.2. IBM Cloud で Submariner をデプロイする準備

IBM Cloud には、従来のクラスターと、仮想プライベートクラウド (VPC) での 2 世代のコンピュートインフラストラクチャー (VPC) の 2 種類の Red Hat OpenShift Kubernetes Service (ROKS) があります。従来のクラスターの IPSec ポートを設定できないため、従来の ROKS クラスターでは Submariner を実行できません。

VPC で、Submariner を使用するように ROKS クラスターを設定するには、以下のリンクの手順を実行します。

1. クラスターを作成する前に、Pod およびサービスのサブネットを指定します。これにより、他のクラスターと CIDR が重複しないようにします。既存のクラスターを使用している場合は、クラスター間で Pod およびサービス CIDR が重複していないことを確認します。手順は、VPC サブネット を参照してください。

2. パブリックゲートウェイを、クラスターで使用されるサブネットに割り当てます。この手順は、パブリック・ゲートウェイ を参照してください。

3. セキュリティーグループ の手順を実行して、クラスターのデフォルトのセキュリティーグループに受信ルールを作成します。ファイアウォールが、ゲートウェイノードの 4500/UDP および 500/UDP ポートでの受信トラフィックおよび送信トラフィックを許可し、他のすべてのノードについては受信および送信の UDP/4800 を許可するようにしてください。

4. クラスター内で、パブリックゲートウェイを持つノードに **submariner.io/gateway=true** とラベルを付けます。

5. クラスターに IPPools を作成して Calico CNI を設定するには、Calico を参照してください。

1.1.2.3. Red Hat OpenShift Dedicated で Submariner をデプロイする準備

Red Hat OpenShift Dedicated は、AWS および Google Cloud Platform によってプロビジョニングされたクラスターをサポートします。

1.1.2.3.1. Red Hat OpenShift Dedicated で AWS に Submariner をデプロイする準備

Red Hat OpenShift Dedicated に AWS クラスターを設定するには、以下の手順を実行します。

1. Red Hat OpenShift Hosted SRE サポートチーム対して サポートチケット を作成し、**cluster-admin** グループに Red Hat OpenShift Dedicated クラスターへのアクセスを許可します。**dedicated-admin** のデフォルトアクセスには、**MachineSet** の作成に必要なパーミッションがありません。

2. グループが作成されたら、Red Hat OpenShift Dedicated ドキュメントの ユーザーへの cluster-admin ロールの付与 の手順を実行して作成した **cluster-admin** グループにユーザー名を追加します。

3. ユーザーの **osdCcsAdmin** の認証情報を設定し、それをサービスアカウントとして使用することができます。

4. クラスターを Red Hat Advanced Cluster Management にインポートして、コンソールを使用した Submariner のデプロイ の手順に従います。

1.1.2.3.2. Red Hat OpenShift Dedicated で Google Cloud Platform に Submariner をデプロイする準備

Red Hat OpenShift Dedicated で Google Cloud Platform クラスターを設定するには、以下の手順を実行します。

1. デプロイメントの管理に使用できる **osd-ccs-admin** という名前のサービスアカウントを設定します。

2. クラスターを Red Hat Advanced Cluster Management にインポートして、コンソールを使用した Submariner のデプロイ の手順に従います。

1.1.3. コンソールを使用した Submariner のデプロイ

submariner-addon コンポーネントは **テクノロジープレビュー** 機能です。

Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、Amazon Web Services、Google Cloud Platform、および VMware vSphere にデプロイされた Red Hat OpenShift Container Platform マネージドクラスターに Submariner をデプロイできます。他のプロバイダーに Submariner をデプロイするには、API を使用した Submariner のデプロイ を参照してください。Red Hat Advanced Cluster Management for Kubernetes コンソールで Submariner をデプロイするには、以下の手順を実行します。

必要なアクセス権限: クラスターの管理者

1. コンソールナビゲーションメニューから **Infrastructure** > **Clusters** を選択します。

2. **Clusters** ページで、**Cluster sets** タブを選択します。Submariner で有効にするクラスターは、同じクラスターセットにある必要があります。

3. Submariner をデプロイするクラスターがすでに同じクラスターセットにある場合は、手順 5 を省略して Submariner をデプロイします。

4. Submariner をデプロイするクラスターが同じクラスターセットにない場合は、以下の手順に従ってクラスターセットを作成します。

    a. **Create cluster set**を選択します。

    b. クラスターセットに名前を付け、**Create** を選択します。

    c. **Manage resource assignments** を選択して、クラスターセットに割り当てます。

    d. Submariner で接続するマネージドクラスターを選択して、クラスターセットに追加します。

    e. **Review** を選択して、選択したクラスターを表示し、確認します。

    f. **Save** を選択してクラスターセットを保存し、作成されるクラスターセットページを表示します。

5. クラスターセットページで、**Submariner add-on** タブを選択します。

6. **Install Submariner add-ons** を選択します。

7. Submariner をデプロイするクラスターを選択します。

8. **Install Submariner add-on** エディターに以下の情報を入力します。

   - **AWS Access Key ID** - このフィールドは、AWS クラスターをインポートする場合にのみ表示されます。

   - **AWS Secret Access Key**: このフィールドは、AWS クラスターをインポートする場合にのみ表示されます。

   - **Google Cloud Platform service account JSON key**: このフィールドは、Google Cloud Platform クラスターをインポートする場合にのみ表示されます。

   - **インスタンスタイプ**: マネージドクラスターで作成されたゲートウェイノードの Amazon Web Services EC2 インスタンスタイプ。デフォルト値は **m5n.large** です。このフィールドは、マネージドクラスター環境が AWS の場合のみ表示されます。

   - **IPsec NAT-T ポート**: IPsec NAT トラバーサルポートのデフォルト値はポート **4500** です。マネージドクラスター環境が VMware vSphere の場合は、ファイアウォールでこのポートが開いていることを確認してください。

   - **ゲートウェイ数**: マネージドクラスターへの Submariner ゲートウェイコンポーネントのデプロイに使用されるワーカーノードの数。デフォルト値は **1** です。値が 1 を超える場合、Submariner ゲートウェイの High Availability (HA) は自動的に有効になります。

   - **ケーブルドライバー**: クラスター間トンネルを維持する Submariner ゲートウェイケーブルエンジンのコンポーネントです。デフォルト値は **Libreswan IPsec** です。

9. エディターの末尾で **Next** を選択して、次のクラスターのエディターに移動し、選択した残りのクラスターごとに、エディターを完了します。

10. 各マネージドクラスターの設定を確認します。

11. **Install** をクリックして、選択したマネージドクラスターに Submariner をデプロイします。インストールと設定が完了するまで数分かかる場合があります。**Submariner add-on** タブの一覧で Submariner ステータスを確認できます。

    - **Connection status** は、マネージドクラスターで確立される Submariner 接続の数を示します。

    - **Agent status** は、Submariner がマネージドクラスターに正常にデプロイされるかどうかを示します。コンソールでは、インストールと設定が完了するまで **Degraded** のステータスをレポートする場合があります。

    - **Gateway nodes labeled** は、マネージドクラスターの Submariner ゲートウェイラベル **submariner.io/gateway=true** が付いたワーカーノードの数を示します。

Submariner がクラスターにデプロイされました。

1.1.4. API を使用した Submariner のデプロイ

Red Hat Advanced Cluster Management for Kubernetes に Submariner をデプロイする前に、接続用にホスト環境でクラスターを準備する必要があります。現時点で、**SubmarinerConfig** API を使用して、Amazon Web Services、Google Cloud Platform、および VMware vSphere のクラスターを自動的に準

備できます。他のプラットフォームの場合、手動で準備する必要があります。手順については、Preparing the hosts to deploy Submariner を参照してください。

1.1.4.1. Submariner をデプロイするホストの準備

Red Hat Advanced Cluster Management に Submariner をデプロイする前に、接続用にホスト環境でクラスターを準備する必要があります。要件はホスティング環境によって異なるため、ホスティング環境の手順に従います。

1.1.4.1.1. Amazon Web Services で Submariner をデプロイする準備

SubmarinerConfig API を使用して、AWS クラスターを Submariner デプロイメントと統合するように設定できます。状況に適した手順を使用して、AWS で Submariner をインストールする準備を行います。

1. Red Hat Advanced Cluster Management でマネージドクラスターを作成していない場合は、AWS 認証情報シークレットが含まれるマネージドクラスターの namespace で、ハブクラスターにシークレットを手動で作成する必要があります。Red Hat Advanced Cluster Management でクラスターを作成した場合は、手順 2 に進みます。
シークレットを作成するには、以下の例のような情報が含まれるコマンドを入力します。

```
export AWS_ACCESS_KEY_ID=<aws-access-key-id>
export AWS_SECRET_ACCESS_KEY=<aws-secret-access-key>

cat << EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: <managed-cluster-name>-aws-creds
  namespace: <managed-cluster-namespace>
type: Opaque
data:
  aws_access_key_id: $(echo -n ${AWS_ACCESS_KEY_ID} | base64 -w0)
  aws_secret_access_key: $(echo -n ${AWS_SECRET_ACCESS_KEY} | base64 -w0)
EOF
```

managed-cluster-name は、マネージドクラスターの名前に置き換えます。

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

aws-access-key-id は、AWS アクセスキー ID に置き換えます。

aws-secret-access-key は、AWS アクセスキーに置き換えます。

2. Red Hat Advanced Cluster Management でマネージドクラスターを作成した場合、または直前の手順でシークレットを作成した後に、以下の例のようなコマンドを入力してクラスターを準備します。

```
cat << EOF | oc apply -f -
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
```

```
  credentialsSecret:
    name: <managed-cluster-name>-aws-creds
EOF
```

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

managed-cluster-name は、マネージドクラスターの名前に置き換えます。**managed-cluster-name-aws-creds** の値は、AWS の認証情報シークレット名で、この情報はハブクラスターの cluster namespace にあります。

注記: 以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

この設定では、Submariner で必要となる、AWS インスタンス上のポートを開きます (ネットワークアドレス変換 (NATT) ポート (4500/UDP)、仮想拡張可能 LAN (VXLAN) ポート (4800/UCP)、および Submariner メトリクスポート (8080/TCP)。また、AWS インスタンスタイプ **m5n.large** を使用して Submariner ゲートウェイとして AWS インスタンスを 1 つ作成します。

- NATT ポートをカスタマイズする場合は、以下の例のような情報が含まれるコマンドを入力します。

  ```
  cat << EOF | oc apply -f -
  apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
  kind: SubmarinerConfig
  metadata:
    name: submariner
    namespace: <managed-cluster-namespace>
  spec:
    credentialsSecret:
      name: <managed-cluster-name>-aws-creds
    IPSecNATTPort: <NATTPort>
  EOF
  ```

  managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

  managed-cluster-name は、マネージドクラスターの名前に置き換えます。**managed-cluster-name-aws-creds** の値は、AWS の認証情報シークレット名で、この情報はハブクラスターの cluster namespace にあります。

  NATTPort は、使用する NATT ポートに置き換えます。

  注記: 以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

- ゲートウェイノードの AWS インスタンスタイプをカスタマイズする場合は、以下の例のような情報が含まれるコマンドを入力します。

  ```
  cat << EOF | oc apply -f -
  apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
  kind: SubmarinerConfig
  metadata:
    name: submariner
    namespace: <managed-cluster-namespace>
  spec:
    credentialsSecret:
      name: <managed-cluster-name>-aws-creds
  ```

```
  gatewayConfig:
     instanceType: <instance-type>
EOF
```

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

managed-cluster-name は、マネージドクラスターの名前に置き換えます。**managed-cluster-name-aws-creds** の値は、AWS の認証情報シークレット名で、この情報はハブクラスターの cluster namespace にあります。

instance-type は、使用する AWS インスタンスタイプに置き換えます。

注記: 以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

- ゲートウェイノードの数をカスタマイズする場合は、以下の例のような情報が含まれるコマンドを入力します。

```
cat << EOF | oc apply -f -
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  credentialsSecret:
    name: <managed-cluster-name>-aws-creds
 gatewayConfig:
     gateways: <gateways>
EOF
```

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

managed-cluster-name は、マネージドクラスターの名前に置き換えます。**managed-cluster-name-aws-creds** の値は、AWS の認証情報シークレット名で、この情報はハブクラスターの cluster namespace にあります。

gateways は、使用するゲートウェイ数に置き換えます。値が 1 より大きい場合には、Submariner ゲートウェイは高可用性を自動的に有効にします。

注記: 以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

1.1.4.1.2. Google Cloud Platform で Submariner をデプロイする準備

SubmarinerConfig API を使用して、Google Cloud Platform クラスターを設定して Submariner デプロイメントと統合できます。状況に適した手順を使用して、Google Cloud Platform を準備し、Submariner をインストールします。

1. Red Hat Advanced Cluster Management でマネージドクラスターを作成していない場合は、Google Cloud Platform 認証情報シークレットが含まれるマネージドクラスターの namespace で、ハブクラスターにシークレットを手動で作成する必要があります。Red Hat Advanced Cluster Management でクラスターを作成した場合は、手順 2 に進みます。
   シークレットを作成するには、以下の例のような情報が含まれるコマンドを入力します。

```
cat << EOF | oc apply -f -
```

```
apiVersion: v1
kind: Secret
metadata:
   name: <managed-cluster-name>-gcp-creds
   namespace: <managed-cluster-namespace>
type: Opaque
data:
   osServiceAccount.json: <gcp-os-service-account-json-file-content>
EOF
```

managed-cluster-name は、マネージドクラスターの名前に置き換えます。**managed-cluster-name-aws-creds** の値は、Google Cloud Platform 認証情報シークレット名を指し、この情報はハブクラスターのクラスター namespace で見つけることができます。

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

gcp-os-service-account-json-file-content は、Google Cloud Platform **osServiceAccount.json** ファイルの内容に置き換えます。

2. Red Hat Advanced Cluster Management でマネージドクラスターを作成した場合、または直前の手順でシークレットをすでに作成している場合は、以下の例のようなコマンドを入力してクラスターを準備します。

```
cat << EOF | oc apply -f -
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
   name: submariner
   namespace: <managed-cluster-namespace>
spec:
   credentialsSecret:
     name: <managed-cluster-name>-gcp-creds
EOF
```

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

managed-cluster-name は、マネージドクラスターの名前に置き換えます。**managed-cluster-name-gcp-creds** の値は、Google Cloud Platform 認証情報シークレット名を指し、ハブクラスターのクラスター namespace で見つけることができます。

注記: 以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

この設定では、Submariner で必要となる、Google Cloud Platform インスタンス上のポートを開きます (ネットワークアドレス変換トラバーサル (NATT) ポート (4500/UDP)、仮想拡張可能 LAN (VXLAN) ポート (4800/UCP)、および Submariner メトリクスポート (8080/TCP))。また、1 つのワーカーノードに Submariner ゲートウェイとしてラベルを付け、Google Cloud Platform クラスターのこのノードのパブリック IP アドレスを有効にします。

- NATT ポートをカスタマイズする場合は、以下の例のような情報が含まれるコマンドを入力します。

```
cat << EOF | oc apply -f -
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
   name: submariner
```

```
    namespace: <managed-cluster-namespace>
spec:
    credentialsSecret:
      name: <managed-cluster-name>-gcp-creds
    IPSecNATTPort: <NATTPort>
EOF
```

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

managed-cluster-name は、マネージドクラスターの名前に置き換えます。**managed-cluster-name-gcp-creds** の値は、Google Cloud Platform 認証情報シークレット名を指し、ハブクラスターのクラスター namespace で見つけることができます。

NATTPort は、使用する NATT ポートに置き換えます。

注記: 以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

- ゲートウェイノードの数をカスタマイズする場合は、以下の例のような情報が含まれるコマンドを入力します。

```
cat << EOF | oc apply -f -
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  credentialsSecret:
    name: <managed-cluster-name>-gcp-creds
 gatewayConfig:
    gateways: <gateways>
EOF
```

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

managed-cluster-name は、マネージドクラスターの名前に置き換えます。**managed-cluster-name-aws-creds** の値は、Google Cloud Platform 認証情報シークレット名を指し、この情報はハブクラスターのクラスター namespace で見つけることができます。

gateways は、使用するゲートウェイ数に置き換えます。値が 1 より大きい場合には、Submariner ゲートウェイは高可用性を自動的に有効にします。

1.1.4.1.3. VMware vSphere で Submariner をデプロイする準備

Submariner は IPsec を使用して、ゲートウェイノード上のクラスター間でセキュアなトンネルを確立します。デフォルトのポートを使用するか、カスタムポートを指定できます。IPsec NATT ポートを指定せずにこの手順を実行すると、通信にはデフォルトのポートが自動的に使用されます。デフォルトのポートは 4500/UDP です。

Submariner は、仮想拡張可能な LAN (VXLAN) を使用して、ワーカーノードおよびマスターノードからゲートウェイノードに移行するときにトラフィックをカプセル化します。VXLAN ポートはカスタマイズできず、常にポート 4800/UDP を使用します。

Submariner は 8080/TCP を使用してクラスターのノード間でメトリクス情報を送信します。このポートはカスタマイズできません。

Submariner を有効にするには、VMWare vSphere 管理者が以下のポートを開放する必要があります。

表1.2 VMware vSphere および Submariner ポート

Name	デフォルト値	カスタマイズ可能
IPsec NATT	4500/UDP	はい
VXLAN	4800/UDP	いいえ
Submariner メトリクス	8080/TCP	いいえ

Submariner をデプロイするように VMware vSphere を準備するには、以下の手順を実行します。

1. IPsec NATT、VXLAN、およびメトリクスポートが開放されていることを確認します。
2. 以下の例のような情報が含まれるコマンドを入力します。

```
cat << EOF | oc apply -f -
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
   name: submariner
   namespace: <managed-cluster-namespace>
spec:{}
EOF
```

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

注記: 以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

この設定は、Submariner にデフォルトの NATT (Network Address Translation-Traversal) ポート (4500/UDP) を使用し、1 つのワーカーノードは vSphere クラスターの Submariner ゲートウェイとしてラベルが付けられています。

Submariner は IP セキュリティー (IPsec) を使用して、ゲートウェイノード上のクラスター間でセキュアなトンネルを確立します。デフォルトの IPsec NATT ポートを使用するか、設定した別のポートを指定できます。IPsec NATT を指定せずにこの手順を実行すると、4500/UDP のポートが通信に自動的に使用されます。

- NATT ポートをカスタマイズする場合は、以下の例のような情報が含まれるコマンドを入力します。

```
cat << EOF | oc apply -f -
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
   name: submariner
   namespace: <managed-cluster-namespace>
spec:
   IPSecNATTPort: <NATTPort>
EOF
```

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

NATTPort は、使用する NATT ポートに置き換えます。

注記: 以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

- ゲートウェイノードの数をカスタマイズする場合は、以下の例のような情報が含まれるコマンドを入力します。

```
cat << EOF | oc apply -f -
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
 gatewayConfig:
    gateways: <gateways>
EOF
```

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

gateways は、使用するゲートウェイ数に置き換えます。値が 1 より大きい場合には、Submariner ゲートウェイは高可用性を自動的に有効にします。

1.1.4.2. ManagedClusterAddOn API を使用した Submariner のデプロイ

ManagedClusterAddOn API を使用して Submariner をデプロイするには、以下の手順を実行します。

1. ManagedClusterSets の手順に従って、ハブクラスターに **ManagedClusterSet の作成および管理** を作成します。**ManagedClusterSet** のエントリーは以下のような内容になります。

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: ManagedClusterSet
metadata:
 name: <managed-cluster-set-name>
```

managed-cluster-set-name は、作成する **ManagedClusterSet** の名前に置き換えます。

注記: Kubernetes namespace の名前の最大長は 63 文字であるため、 **<managed-cluster-set-name>** の最大長さは 56 文字です。 **<managed-cluster-set-name>** の長さが 56 を超える場合には、**<managed-cluster-set-name>** は、先頭から 56 文字で省略されます。

ManagedClusterSet が作成されたら、**submariner-addon** は **<managed-cluster-set-name>-broker** と呼ばれる namespace を作成し、その namespace に Submariner ブローカーをデプロイします。

2. 以下のコマンドを入力して、マネージドクラスターを 1 つ **ManagedClusterSet** に追加します。

```
oc label managedclusters <managed-cluster-name> "cluster.open-cluster-management.io/clusterset=<managed-cluster-set-name>" --overwrite
```

managedcluster-name は、**ManagedClusterSet** に追加するマネージドクラスターの名前に置き換えます。

ManagedClusterSet-name は、マネージドクラスターを追加する **ManagedClusterSet** の名前に置き換えます。

3. 以下のコマンドを入力してマネージドクラスターで Submariner をデプロイします。

```
cat << EOF | oc apply -f -
apiVersion: addon.open-cluster-management.io/v1alpha1
kind: ManagedClusterAddOn
metadata:
    name: submariner
    namespace: <managed-cluster-name>
spec:
    installNamespace: submariner-operator
```

managedcluster-name は、Submariner で使用するマネージドクラスターの名前に置き換えます。

ManagedClusterAddOn の仕様の **installNamespace** フィールドは、Submariner をインストールするマネージドクラスター上の namespace に置き換えます。現在、**Submariner-operator** namespace に Submariner をインストールする必要があります。

ManagedClusterAddOn の作成後に、**submariner-addon** は Submariner をマネージドクラスターの **submariner-operator** namespace にデプロイします。この **ManagedClusterAddOn** のステータスから Submariner のデプロイメントステータスを表示できます。

注記: **ManagedClusterAddOn** の名前は **submariner** である必要があります。

4. Submariner を有効にするすべてのマネージドクラスターで、手順 2 と 3 を繰り返します。

5. マネージドクラスターに Submariner をデプロイしたら、以下のコマンドを入力して、Submariner **ManagedClusterAddOn** のステータスを確認して Submariner のデプロイメントのステータスを確認できます。

```
oc -n <managed-cluster-name> get managedclusteraddons submariner -oyaml
```

cluster-name は、マネージドクラスターの名前に置き換えます。

Submariner **ManagedClusterAddOn** のステータスの 3 つの条件により、Submariner のデプロイメントステータスが分かります。

- **SubmarinerGatewayNodesLabeled** の条件は、マネージドクラスターに Submariner ゲートウェイノードにラベル付けされているかどうかを示します。
- **SubmarinerAgentDegraded** の条件は、Submariner がマネージドクラスターに正常にデプロイされるかどうかを示します。
- **SubmarinerConnectionDegraded** の条件は、Submariner でマネージドクラスターで確立される接続の数を示します。

1.1.5. Submariner のサービス検出の有効化

submariner-addon コンポーネントは **テクノロジープレビュー** 機能です。

Submariner がマネージドクラスターと同じ環境にデプロイされると、**ManagedClusterSet** のクラスター全体の Pod とサービスとの間でセキュアな IP ルーティング用にルートが設定されます。**ManagedClusterSet** の他のクラスターにサービスを表示し、検出できるようにするに

は、**ServiceExport** オブジェクトを作成する必要があります。**ServiceExport** オブジェクトでサービスをエクスポートすると、**<service>.<namespace>.svc.clusterset.local** 形式でサービスにアクセスできます。複数のクラスターが同じ名前で、同じ namespace からサービスをエクスポートすると、他のクラスターは、その複数のクラスターを 1 つの論理サービスとして認識されます。

この例では、**default** の namespace で **nginx** サービスを使用しますが、Kubernetes の **ClusterIP** サービスまたはヘッドレスサービスを検出できます。

1. 以下のコマンドを入力して、**ManagedClusterSet** のマネージドクラスターに **nginx** サービスのインスタンスを適用します。

```
oc -n default create deployment nginx --image=nginxinc/nginx-unprivileged:stable-alpine
oc -n default expose deployment nginx --port=8080
```

2. YAML ファイルに以下の内容の **ServiceExport** エントリーを作成して、サービスをエクスポートします。

```
apiVersion: multicluster.x-k8s.io/v1alpha1
kind: ServiceExport
metadata:
  name: <service-name>
  namespace: <service-namespace>
```

   service-name を、エクスポートするサービスの名前に置き換えます。この例では、**nginx** になります。**service-namespace** を、サービスが置かれた namespace の名前に置き換えます。この例では、**default** になります。

3. 別のマネージドクラスターから以下のコマンドを実行して、**nginx** サービスにアクセスできることを確認します。

```
oc -n default run --generator=run-pod/v1 tmp-shell --rm -i --tty --image
quay.io/submariner/nettest -- /bin/bash curl nginx.default.svc.clusterset.local:8080
```

これで、**nginx** サービス検出が Submariner に対して設定されました。