Red Hat Advanced Cluster Management for Kubernetes 2.5

トラブルシューティング

クラスターのトラブルシューティングトピックの一覧を確認します。また、must-gather コマンドを使用してログを収集することもできます。

Last Updated: 2023-07-14

Red Hat Advanced Cluster Management for Kubernetes 2.5 トラブルシューティング

クラスターのトラブルシューティングトピックの一覧を確認します。また、must-gather コマンドを使用してログを収集することもできます。

法律上の通知

概要

クラスターのトラブルシューティングトピックの一覧を確認します。また、must-gather コマンドを使用してログを収集することもできます。

目次

第1章 トラブルシューティング

トラブルシューティングガイドをご使用の前に **oc adm must-gather** コマンドを実行して、詳細およびログを収集し、問題のデバッグ手順を行います。詳細は、must-gather コマンドでのトラブルシューティング を参照してください。

また、ロールベースのアクセス権限を確認してください。詳細は、ロールベースのアクセス制御 を参照してください。

1.1. 文書化されたトラブルシューティング

以下に、Red Hat Advanced Cluster Management for Kubernetes のトラブルシューティングのトピックリストを表示します。

インストール

インストールタスクの主なドキュメントを表示するには、Installing を参照してください。

- インストールステータスがインストールまたは保留中の状態のトラブルシューティング
- 再インストールに失敗する場合のトラブルシューティング

クラスター管理

クラスターの管理に関する主なドキュメントを表示するには、クラスターの管理 を参照してください。

- オフラインクラスターのトラブルシューティング
- マネージドクラスターのインポート失敗に関するトラブルシューティング
- クラスターの再インポートが不明な権限エラーで失敗する
- Pending Import ステータスのクラスターのトラブルシューティング
- 証明書を変更した後のインポート済みクラスターのオフラインでのトラブルシューティング
- クラスターのステータスが offline から available に変わる場合のトラブルシューティング
- VMware vSphere でのクラスター作成のトラブルシューティング
- ステータスが Pending または Failed のクラスターのコンソールでのトラブルシューティング
- OpenShift Container Platform バージョン 3.11 クラスターのインポートの失敗時のトラブルシューティング
- 状態が Degraded の Klusterlet のトラブルシューティング
- マネージドクラスターでの Klusterlet アプリケーションマネージャーのトラブルシューティング
- オブジェクトストレージチャネルシークレットのトラブルシューティング
- クラスターの削除後も namespace が残る
- クラスターのインポート時の auto-import-secret-exists エラー

アプリケーション管理

アプリケーション管理に関する主なドキュメントを表示するには、Managing applications を参照してください。

- アプリケーションの Kubernetes デプロイメントバージョンのトラブルシューティング
- スタンドアロンのサブスクリプションメモリーの問題のトラブルシューティング
- アプリケーションの Git サーバー接続のトラブルシューティング
- ローカルクラスターが選択されていない問題のトラブルシューティング

ガバナンス

セキュリティーガイドを表示するには、Risk and compliance を参照してください。

コンソールの可観測性

コンソールの可観測性には、ヘッダーおよびナビゲーション機能、検索、および移動機能が含まれます。可観測性ガイドを表示するには、Observability in the console を参照してください。

- grafana のトラブルシューティング
- 可観測性のトラブルシューティング
- OpenShift モニターリングサービスのトラブルシューティング
- 検索アグリゲーター Pod のステータスのトラブルシューティング
- metrics-collector のトラブルシューティング

Submariner のネットワーキングとサービスディスカバリー

このセクションでは、Red Hat Advanced Cluster Management で Submariner を使用するときに発生する可能性のある Submariner のトラブルシューティング手順を示します。Submariner の一般的なトラブルシューティング情報については、Submariner のドキュメントの Troubleshooting を参照してください。

Submariner ネットワーキングサービスとサービスディスカバリの主なドキュメントを表示するには、Submariner multicluster networking and service discovery を参照してください。

- インストール後に Submariner が接続されない場合のトラブルシューティング - 一般情報
- Submariner アドオンのステータスが低下してい場合のトラブルシューティング

1.2. MUST-GATHER コマンドを実行したトラブルシューティング

トラブルシューティングを開始するには、問題のデバッグを行う **must-gather** コマンドを実行する場合のトラブルシューティングシナリオについて確認し、このコマンドの使用を開始する手順を参照してください。

必要なアクセス権限: クラスターの管理者

1.2.1. Must-gather のシナリオ

- **シナリオ 1:** 文書化されたトラブルシューティング セクションを使用して、問題の解決策がまとめられているかを確認します。本ガイドは、製品の主な機能別に設定されています。

このシナリオでは、解決策が本書にまとめられているかどうかを、本ガイドで確認します。たとえば、クラスターの作成に関するトラブルシューティングの場合は、**クラスターの管理** セクションの解決策を探します。

- **シナリオ 2:** 問題の解決策の手順が文書にまとめられていない場合は、**must-gather** コマンドを実行し、その出力を使用して問題をデバッグします。
- **シナリオ 3:** **must-gather** コマンドの出力を使用して問題をデバッグできない場合は、出力を Red Hat サポートに共有します。

1.2.2. Must-gather の手順

must-gather コマンドの使用を開始するには、以下の手順を参照してください。

1. **must-gather** コマンドについて確認し、Red Hat OpenShift Container Platform の クラスターに関するデータの収集 に必要な前提条件をインストールします。
2. クラスターにログインします。データおよびディレクトリーの収集に使用する Red Hat Advanced Cluster Management for Kubernetes イメージを追加します。以下のコマンドを実行して、出力用にイメージとディレクトリーを挿入します。

```
oc adm must-gather --image=registry.redhat.io/rhacm2/acm-must-gather-rhel8:v2.5.0 --dest-dir=<directory>
```

3. 通常のユースケースでは、ハブクラスターにログインして、**must-gather** を実行する必要があります。
注記: マネージドクラスターを確認する場合は、**cluster-scoped-resources** ディレクトリーにある **gather-managed.log** ファイルを検索します。

```
<your-directory>/cluster-scoped-resources/gather-managed.log>
```

JOINED および AVAILABLE 列に **True** が設定されていないマネージドクラスターがないかを確認します。**must-gather** コマンドは、ステータスが **True** として関連付けられていないクラスター上で、実行できます。

4. 指定したディレクトリーに移動し、以下のレベルに整理されている出力を確認します。
   - ピアレベル 2 つ: **cluster-scoped-resources** と **namespace** のリソース
   - それぞれに対するサブレベル: クラスタースコープおよび namespace スコープの両方のリソースに対するカスタムリソース定義の API グループ。
   - それぞれに対する次のレベル: **kind** でソートされた YAML ファイル

1.2.3. 非接続環境での must-gather

非接続環境で **must-gather** コマンドを実行するには、次の手順を実行します。

1. 非接続環境では、Red Hat Operator のカタログイメージをミラーレジストリーにミラーリングします。詳細は、ネットワーク切断状態でのインストール を参照してください。
2. 次のコマンドを実行して、ミラーレジストリーからイメージを参照するログを抽出します。

```
REGISTRY=registry.example.com:5000
IMAGE=$REGISTRY/rhacm2/acm-must-gather-
```

```
rhel8@sha256:ff9f37eb400dc1f7d07a9b6f2da9064992934b69847d17f59e385783c071b9d8

oc adm must-gather --image=$IMAGE --dest-dir=./data
```

1.3. インストールステータスがインストールまたは保留中の状態のトラブルシューティング

Red Hat Advanced Cluster Management のインストール時に、**MultiClusterHub** は **Installing** フェーズのままとなるか、または複数の Pod が **Pending** ステータスのままとなります。

1.3.1. 現象: Pending 状態で止まる

MultiClusterHub をインストールしてから、**MultiClusterHub** リソースの **status.components** フィールドからのコンポーネントの 1 つ以上で **ProgressDeadlineExceeded** と報告したまま 10 分以上経過しています。クラスターのリソース制約が問題となっている場合があります。

Multiclusterhub がインストールされている namespace の Pod を確認します。以下のようなステータスとともに **Pending** と表示される場合があります。

```
reason: Unschedulable
message: '0/6 nodes are available: 3 Insufficient cpu, 3 node(s) had taint {node-
role.kubernetes.io/master:
      }, that the pod didn't tolerate.'
```

このような場合には、ワーカーノードにはクラスターでの製品実行に十分なリソースがありません。

1.3.2. 問題の解決: ワーカーノードのサイズの調整

この問題が発生した場合は、大規模なワーカーノードまたは複数のワーカーノードでクラスターを更新する必要があります。クラスターのサイジングのガイドラインについては、クラスターのサイジング を参照してください。

1.4. 再インストールに失敗する場合のトラブルシューティング

Red Hat Advanced Cluster Management for Kubernetes を再インストールすると Pod が起動しません。

1.4.1. 現象: 再インストールの失敗

Red Hat Advanced Cluster Management のインストール後に Pod が起動しない場合には、Red Hat Advanced Cluster Management 以前にインストールされており、今回のインストールを試行する前にすべてのパーツが削除されていない可能性があります。

Pod はこのような場合に、インストールプロセスの完了後に起動しません。

1.4.2. 問題の解決: 再インストールの失敗

この問題が発生した場合は、以下の手順を実行します。

1. アンインストール の手順に従い、現在のコンポーネントを削除し、アンインストールプロセスを実行します。

2. Helm のインストール の手順に従い、Helm CLI バイナリーバージョン 3.2.0 以降をインストールします。

3. **oc** コマンドが実行できるように、Red Hat OpenShift Container Platform CLI が設定されていることを確認してください。**oc** コマンドの設定方法に関する詳細は、Red Hat OpenShift Container Platform ドキュメントの OpenShift CLI の使用方法 を参照してください。

4. 以下のスクリプトをファイルにコピーします。

```
#!/bin/bash
ACM_NAMESPACE=<namespace>
oc delete mch --all -n $ACM_NAMESPACE
helm ls --namespace $ACM_NAMESPACE | cut -f 1 | tail -n +2 | xargs -n 1 helm delete --
namespace $ACM_NAMESPACE
oc delete apiservice v1beta1.webhook.certmanager.k8s.io v1.admission.cluster.open-cluster-
management.io v1.admission.work.open-cluster-management.io
oc delete clusterimageset --all
oc delete configmap -n $ACM_NAMESPACE cert-manager-controller cert-manager-
cainjector-leader-election cert-manager-cainjector-leader-election-core
oc delete consolelink acm-console-link
oc delete crd klusterletaddonconfigs.agent.open-cluster-management.io
placementbindings.policy.open-cluster-management.io policies.policy.open-cluster-
management.io userpreferences.console.open-cluster-management.io
searchservices.search.acm.com discoveredclusters.discovery.open-cluster-management.io
discoveryconfigs.discovery.open-cluster-management.io
oc delete mutatingwebhookconfiguration cert-manager-webhook cert-manager-webhook-
v1alpha1 ocm-mutating-webhook managedclustermutators.admission.cluster.open-cluster-
management.io
oc delete oauthclient multicloudingress
oc delete rolebinding -n kube-system cert-manager-webhook-webhook-authentication-reader
oc delete scc kui-proxy-scc
oc delete validatingwebhookconfiguration cert-manager-webhook cert-manager-webhook-
v1alpha1 channels.apps.open.cluster.management.webhook.validator application-webhook-
validator multiclusterhub-operator-validating-webhook ocm-validating-webhook
```

スクリプトの **<namespace>** は、Red Hat Advanced Cluster Management がインストールされている namespace 名に置き換えます。namespace が消去され削除されるため、正しい namespace を指定するようにしてください。

5. スクリプトを実行して、アーティファクトを以前のインストールから削除します。

6. インストールを実行します。ネットワーク接続時のオンラインインストール を参照してください。

1.5. オフラインクラスターのトラブルシューティング

クラスターのステータスがオフラインと表示される一般的な原因がいくつかあります。

1.5.1. 現象: クラスターのステータスがオフライン状態である

クラスターの作成手順を完了したら、Red Hat Advanced Cluster Management コンソールからアクセスできず、クラスターのステータスが **offline** と表示されます。

1.5.2. 問題の解決: クラスターのステータスがオフライン状態になっている

1. マネージドクラスターが利用可能かどうかを確認します。これは、Red Hat Advanced Cluster Management コンソールの **Clusters** エリアで確認できます。
利用不可の場合は、マネージドクラスターの再起動を試行します。

2. マネージドクラスターのステータスがオフラインのままの場合は、以下の手順を実行します。

   a. ハブクラスターで **oc get managedcluster <cluster_name> -o yaml** コマンドを実行します。**<cluster_name>** は、クラスター名に置き換えます。

   b. **status.conditions** セクションを見つけます。

   c. **type: ManagedClusterConditionAvailable** のメッセージを確認して、問題を解決します。

1.6. マネージドクラスターのインポート失敗に関するトラブルシューティング

クラスターのインポートに失敗した場合は、クラスターのインポートが失敗した理由を判別するためにいくつかの手順を実行できます。

1.6.1. 現象: インポートされたクラスターを利用できない

クラスターのインポート手順を完了したら、Red Hat Advanced Cluster Management for Kubernetes コンソールからアクセスできません。

1.6.2. 問題の解決: インポートされたクラスターが利用できない

インポートの試行後にインポートクラスターが利用できない場合には、いくつかの理由があります。クラスターのインポートに失敗した場合は、インポートに失敗した理由が見つかるまで以下の手順を実行します。

1. Red Hat Advanced Cluster Management ハブクラスターで以下のコマンドを実行し、Red Hat Advanced Cluster Management インポートコントローラーが実行していることを確認します。

   ```
   kubectl -n multicluster-engine get pods -l app=managedcluster-import-controller-v2
   ```

   実行中の Pod が 2 つ表示されるはずです。Pod のいずれかが実行されていない場合には、以下のコマンドを実行してログを表示して理由を判別します。

   ```
   kubectl -n multicluster-engine logs -l app=managedcluster-import-controller-v2 --tail=-1
   ```

2. Red Hat Advanced Cluster Management ハブクラスターで以下のコマンドを実行し、Red Hat Advanced Cluster Management インポートコントローラーでマネージドクラスターのインポートシークレットが正常に生成されたかどうかを確認します。

   ```
   kubectl -n <managed_cluster_name> get secrets <managed_cluster_name>-import
   ```

   インポートシークレットが存在しない場合は、以下のコマンドを実行してインポートコントローラーのログエントリーを表示し、作成されていない理由を判断します。

   ```
   kubectl -n multicluster-engine logs -l app=managedcluster-import-controller-v2 --tail=-1 | grep importconfig-controller
   ```

3. Red Hat Advanced Cluster Management ハブクラスターで、マネージドクラスターが **local-**

cluster で、Hive によってプロビジョニングされるか、または自動インポートシークレットがある場合は、以下のコマンドを実行してマネージドクラスターのインポートステータスを確認します。

```
kubectl get managedcluster <managed_cluster_name> -o=jsonpath='{range
.status.conditions[*]}{.type}{"\t"}{.status}{"\t"}{.message}{"\n"}{end}' | grep
ManagedClusterImportSucceeded
```

ManagedClusterImportSucceeded が **true** でない場合には、コマンドの結果で失敗の理由が表示されます。

4. マネージドクラスターの Klusterlet ステータスが degraded 状態でないかを確認します。Klusterlet のパフォーマンスが低下した理由を特定するには、Troubleshooting Klusterlet with degraded conditions を参照してください。

1.7. クラスターの再インポートが不明な権限エラーで失敗する

管理対象クラスターを Red Hat Advanced Cluster Management ハブクラスターに再インポートする際に問題が発生した場合は、手順に従って問題をトラブルシューティングします。

1.7.1. 症状: クラスターの再インポートが不明な権限エラーで失敗する

Red Hat Advanced Cluster Management を使用して OpenShift Container Platform クラスターをプロビジョニングした後、API サーバー証明書を OpenShift Container Platform クラスターに変更または追加すると、**x509: certificate signed by unknown authority** エラーでクラスターの再インポートが失敗する場合があります。

1.7.2. 問題の特定: クラスターの再インポートが不明な権限エラーで失敗する

管理対象クラスターの再インポートに失敗した後、次のコマンドを実行して、Red Hat Advanced Cluster Management ハブクラスターのインポートコントローラーログを取得します。

```
kubectl -n multicluster-engine logs -l app=managedcluster-import-controller-v2 -f
```

次のエラーログが表示される場合は、マネージドクラスター API サーバーの証明書が変更されている可能性があります。

ERROR Reconciler error {"controller": "clusterdeployment-controller", "object": {"name":"awscluster1","namespace":"awscluster1"}, "namespace": "awscluster1", "name": "awscluster1", "reconcileID": "a2cccf24-2547-4e26-95fb-f258a6710d80", "error": "Get \"https://api.awscluster1.dev04.red-chesterfield.com:6443/api?timeout=32s\": x509: certificate signed by unknown authority"}

マネージドクラスター API サーバー証明書が変更されたかどうかを確認するには、次の手順を実行します。

1. 次のコマンドを実行して、**your-managed-cluster-name** をマネージドクラスターの名前に置き換えて、マネージドクラスターの名前を指定します。

```
cluster_name=<your-managed-cluster-name>
```

2. 次のコマンドを実行して、マネージドクラスター **kubeconfig** シークレット名を取得します。

```
kubeconfig_secret_name=$(oc -n ${cluster_name} get clusterdeployments ${cluster_name} -
ojsonpath='{.spec.clusterMetadata.adminKubeconfigSecretRef.name}')
```

3. 次のコマンドを実行して、**kubeconfig** を新しいファイルにエクスポートします。

```
oc -n ${cluster_name} get secret ${kubeconfig_secret_name} -ojsonpath={.data.kubeconfig} |
base64 -d > kubeconfig.old
```

```
export KUBECONFIG=kubeconfig.old
```

4. 次のコマンドを実行して、**kubeconfig** を使用してマネージドクラスターから namespace を取得します。

```
oc get ns
```

次のメッセージのようなエラーが表示された場合は、クラスター API サーバーの証明書が変更になっており、**kubeconfig** ファイルが無効です。

Unable to connect to the server: x509: certificate signed by unknown authority

1.7.3. 問題の解決: クラスターの再インポートが不明な権限エラーで失敗する

マネージドクラスター管理者は、マネージドクラスター用に新しい有効な **kubeconfig** ファイルを作成する必要があります。

新しい **kubeconfig** を作成したら、次の手順を実行して、マネージドクラスターの新しい **kubeconfig** を更新します。

1. 次のコマンドを実行して、**your-managed-cluster-name** をマネージドクラスターの名前に置き換えて、マネージドクラスターの名前を指定します。

```
cluster_name=<your-managed-cluster-name>
```

2. 次のコマンドを実行して、マネージドクラスターの新しい **kubeconfig** を更新します。

```
kubeconfig=$(cat <your-new-valid-kubeconfig-file-path> | base64 -w0)
kubeconfig_patch="[{\"op\":\"replace\", \"path\":\"/data/kubeconfig\",
\"value\":\"${kubeconfig}\"}]"
kubeconfig_secret_name=$(oc -n ${cluster_name} get clusterdeployments ${cluster_name} -
ojsonpath='{.spec.clusterMetadata.adminKubeconfigSecretRef.name}')
```

```
oc -n ${cluster_name} patch secrets ${kubeconfig_secret_name} --type='json' -
p=${kubeconfig_patch}
```

1.8. PENDING IMPORT ステータスのクラスターのトラブルシューティング

クラスターのコンソールで継続的に **Pending import** と表示される場合は、以下の手順を実行して問題をトラブルシューティングしてください。

1.8.1. 現象: ステータスが Pending Import クラスター

Red Hat Advanced Cluster Management コンソールを使用してクラスターをインポートした後に、コンソールで、クラスターのステータスが **Pending import** と表示されます。

1.8.2. 問題の特定: ステータスが Pending Import クラスター

1. マネージドクラスターで以下のコマンドを実行し、問題のある Kubernetes Pod 名を表示します。

```
kubectl get pod -n open-cluster-management-agent | grep klusterlet-registration-agent
```

2. マネージドクラスターで以下のコマンドを実行し、エラーのログエントリーを探します。

```
kubectl logs <registration_agent_pod> -n open-cluster-management-agent
```

   registration_agent_pod は、手順 1 で特定した Pod 名に置き換えます。

3. 返された結果に、ネットワーク接続の問題があったと示すテキストがないかどうかを検索します。たとえば、**no such host** です。

1.8.3. 問題の解決: ステータスが Pending Import クラスター

1. ハブクラスターで以下のコマンドを実行して、問題のあるポート番号を取得します。

```
oc get infrastructure cluster -o yaml | grep apiServerURL
```

2. マネージドクラスターのホスト名が解決でき、ホストおよびポートへの送信接続が機能していることを確認します。
   マネージドクラスターで通信が確立できない場合は、クラスターのインポートが完了していません。マネージドクラスターのクラスターステータスは、**Pending import** になります。

1.9. クラスターが既に存在するというエラーのトラブルシューティング

OpenShift Container Platform クラスターを Red Hat Advanced Cluster Management **MultiClusterHub** にインポートできず、**AlreadyExists** エラーを受け取る場合は、手順に従って問題をトラブルシューティングします。

1.9.1. 以前の動作: OpenShift Container Platform クラスターをインポートするときにエラーログが既に存在する

OpenShift Container Platform クラスターを Red Hat Advanced Cluster Management **MultiClusterHub** にインポートすると、エラーログが表示されます。

```
error log:
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+,
unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
Error from server (AlreadyExists): error when creating "STDIN":
customresourcedefinitions.apiextensions.k8s.io "klusterlets.operator.open-cluster-management.io"
already exists
The cluster cannot be imported because its Klusterlet CRD already exists.
Either the cluster was already imported, or it was not detached completely during a previous detach
process.
Detach the existing cluster before trying the import again."
```

1.9.2. 問題の特定: OpenShift Container Platform クラスターのインポート時に既に存在する

以下のコマンドを実行して、新しい Red Hat Advanced Cluster Management **MultiClusterHub** にインポートする Red Hat Advanced Cluster Management 関連のリソースがクラスター上にあるかどうかを確認します。

```
oc get all -n open-cluster-management-agent
oc get all -n open-cluster-management-agent-addon
```

1.9.3. 問題の解決: OpenShift Container Platform クラスターのインポート時に既に存在する

次のコマンドを実行して、既存のリソースを削除します。

```
oc delete namespaces open-cluster-management-agent open-cluster-management-agent-addon --wait=false
oc get crds | grep open-cluster-management.io | awk '{print $1}' | xargs oc delete crds --wait=false
oc get crds | grep open-cluster-management.io | awk '{print $1}' | xargs oc patch crds --type=merge -p '{"metadata":{"finalizers": []}}'
```

1.10. VMWARE VSPHERE でのクラスター作成のトラブルシューティング

VMware vSphere で Red Hat OpenShift Container Platform クラスターを作成する時に問題が発生した場合は、以下のトラブルシューティング情報を参照して、この情報のいずれかが問題に対応しているかどうかを確認します。

注記: VMware vSphere でクラスター作成プロセスが失敗した場合に、リンクが有効にならずログが表示されないことがあります。上記が発生する場合は、**hive-controllers** Pod のログを確認して問題を特定できます。**hive-controllers** ログは **hive** namespace にあります。

1.10.1. 証明書 の IP SAN エラーでマネージドクラスターの作成に失敗する

1.10.1.1. 現象: 証明書の IP SAN エラーでマネージドクラスターの作成に失敗する

VMware vSphere で新規の Red Hat OpenShift Container Platform クラスターを作成した後に、証明書 IP SAN エラーを示すエラーメッセージでクラスターに問題が発生します。

1.10.1.2. 問題の特定: 証明書の IP SAN エラーでマネージドクラスターの作成に失敗する

マネージドクラスターのデプロイメントに失敗して、デプロイメントログに以下のエラーが返されます。

```
time="2020-08-07T15:27:55Z" level=error msg="Error: error setting up new vSphere SOAP client: Post https://147.1.1.1/sdk: x509: cannot validate certificate for xx.xx.xx.xx because it doesn't contain any IP SANs"
time="2020-08-07T15:27:55Z" level=error
```

1.10.1.3. 問題の解決: 証明書の IP SAN エラーでマネージドクラスターの作成に失敗する

認証情報の IP アドレスではなく VMware vCenter サーバー完全修飾ホスト名を使用します。また、VMware vCenter CA 証明書を更新して、IP SAN を組み込むこともできます。

1.10.2. 不明な証明局のエラーでマネージドクラスターの作成に失敗する

1.10.2.1. 現象: 不明な証明局のエラーでマネージドクラスターの作成に失敗する

VMware vSphere で新規の Red Hat OpenShift Container Platform クラスターを作成した後に、証明書が不明な証明局により署名されているのでクラスターに問題が発生します。

1.10.2.2. 問題の特定: 不明な証明局のエラーでマネージドクラスターの作成に失敗する

マネージドクラスターのデプロイメントに失敗して、デプロイメントログに以下のエラーが返されます。

```
Error: error setting up new vSphere SOAP client: Post https://vspherehost.com/sdk: x509: certificate signed by unknown authority"
```

1.10.2.3. 問題の解決: 不明な証明局のエラーでマネージドクラスターの作成に失敗する

認証情報の作成時に認証局の正しい証明書が入力されていることを確認します。

1.10.3. 証明書の期限切れでマネージドクラスターの作成に失敗する

1.10.3.1. 現象: 証明書の期限切れでマネージドクラスターの作成に失敗する

VMware vSphere で新規の Red Hat OpenShift Container Platform クラスターを作成した後に、証明書の期限が切れているか、有効にしていないため、クラスターに問題が発生します。

1.10.3.2. 問題の特定: 証明書の期限切れでマネージドクラスターの作成に失敗する

マネージドクラスターのデプロイメントに失敗して、デプロイメントログに以下のエラーが返されます。

```
x509: certificate has expired or is not yet valid
```

1.10.3.3. 問題の解決: 証明書の期限切れでマネージドクラスターの作成に失敗する

ESXi ホストの時間が同期されていることを確認します。

1.10.4. タグ付けの権限が十分ではないためマネージドクラスターの作成に失敗する

1.10.4.1. 現象: タグ付けの権限が十分ではないためマネージドクラスターの作成に失敗する

VMware vSphere で新規の Red Hat OpenShift Container Platform クラスターを作成した後に、タグ付けの使用に十分な権限がないためクラスターに問題が発生します。

1.10.4.2. 問題の特定: タグ付けの権限が十分にないためにマネージドクラスターの作成に失敗する

マネージドクラスターのデプロイメントに失敗して、デプロイメントログに以下のエラーが返されます。

```
time="2020-08-07T19:41:58Z" level=debug msg="vsphere_tag_category.category: Creating..."
```

```
time="2020-08-07T19:41:58Z" level=error
time="2020-08-07T19:41:58Z" level=error msg="Error: could not create category: POST
https://vspherehost.com/rest/com/vmware/cis/tagging/category: 403 Forbidden"
time="2020-08-07T19:41:58Z" level=error
time="2020-08-07T19:41:58Z" level=error msg="  on ../tmp/openshift-install-436877649/main.tf line
54, in resource \"vsphere_tag_category\" \"category\":"
time="2020-08-07T19:41:58Z" level=error msg="  54: resource \"vsphere_tag_category\" \"category\"
{"
```

1.10.4.3. 問題の解決: タグ付けの権限が十分ではないためマネージドクラスターの作成に失敗する

VMware vCenter が必要とするアカウントの権限が正しいことを確認します。詳細は、インストール時に削除されたイメージレジストリー を参照してください。

1.10.5. 無効な dnsVIP でマネージドクラスターの作成に失敗する

1.10.5.1. 現象: 無効な dnsVIP でマネージドクラスターの作成に失敗する

VMware vSphere で新規の Red Hat OpenShift Container Platform クラスターを作成した後に、dnsVIP が無効であるため、クラスターに問題が発生します。

1.10.5.2. 問題の特定: 無効な dnsVIP でマネージドクラスターの作成に失敗する

VMware vSphere で新しいマネージドクラスターをデプロイしようとして以下のメッセージが表示されるのは、VMware Installer Provisioned Infrastructure (IPI) をサポートしない以前の OpenShift Container Platform リリースイメージを使用しているためです。

```
failed to fetch Master Machines: failed to load asset \\\"Install Config\\\": invalid \\\"install-
config.yaml\\\" file: platform.vsphere.dnsVIP: Invalid value: \\\"\\\": \\\"\\\" is not a valid IP
```

1.10.5.3. 問題の解決: 無効な dnsVIP でマネージドクラスターの作成に失敗する

VMware インストーラーでプロビジョニングされるインフラストラクチャーをサポートする OpenShift Container Platform で、新しいバージョンのリリースイメージを選択します。

1.10.6. ネットワークタイプが正しくないためマネージドクラスターの作成に失敗する

1.10.6.1. 現象: ネットワークタイプが正しくないためマネージドクラスターの作成に失敗する

VMware vSphere で新規の Red Hat OpenShift Container Platform クラスターを作成した後に、間違ったネットワークタイプが指定されているため、クラスターに問題が発生します。

1.10.6.2. 問題の特定: ネットワークタイプが正しくないためマネージドクラスターの作成に失敗する

VMware vSphere で新しいマネージドクラスターをデプロイしようとして以下のメッセージが表示されるのは、VMware Installer Provisioned Infrastructure (IPI) をサポートしない以前の OpenShift Container Platform イメージを使用しているためです。

```
time="2020-08-11T14:31:38-04:00" level=debug msg="vsphereprivate_import_ova.import:
Creating..."
```

```
time="2020-08-11T14:31:39-04:00" level=error
time="2020-08-11T14:31:39-04:00" level=error msg="Error: rpc error: code = Unavailable desc =
transport is closing"
time="2020-08-11T14:31:39-04:00" level=error
time="2020-08-11T14:31:39-04:00" level=error
time="2020-08-11T14:31:39-04:00" level=fatal msg="failed to fetch Cluster: failed to generate asset
\"Cluster\": failed to create cluster: failed to apply Terraform: failed to complete the change"
```

1.10.6.3. 問題の解決: ネットワークタイプが正しくないためマネージドクラスターの作成に失敗する

指定の VMware クラスターに対して有効な VMware vSphere ネットワークタイプを選択します。

1.10.7. ディスクの変更処理のエラーでマネージドクラスターの作成に失敗する

1.10.7.1. 現象: ディスクの変更処理のエラーが原因でマネージドクラスターの追加に失敗する

VMware vSphere で新規の Red Hat OpenShift Container Platform クラスターを作成した後に、ディスク変更処理時にエラーによりクラスターに問題が発生します。

1.10.7.2. 問題の特定: ディスクの変更処理のエラーが原因でマネージドクラスターの追加に失敗する

以下のようなメッセージがログに表示されます。

```
ERROR
ERROR Error: error reconfiguring virtual machine: error processing disk changes post-clone: disk.0:
ServerFaultCode: NoPermission: RESOURCE (vm-71:2000), ACTION (queryAssociatedProfile):
RESOURCE (vm-71), ACTION (PolicyIDByVirtualDisk)
```

1.10.7.3. 問題の解決: ディスクの変更処理のエラーが原因でマネージドクラスターの追加に失敗する

VMware vSphere クライアントを使用してユーザーに **プロファイル駆動型のストレージ権限** の **全権限** を割り当てます。

1.11. OPENSHIFT CONTAINER PLATFORM バージョン 3.11 クラスターのインポートの失敗時のトラブルシューティング

1.11.1. 現象: OpenShift Container Platform バージョン 3.11 クラスターのインポートに失敗する

Red Hat OpenShift Container Platform バージョン 3.11 クラスターのインポートを試行すると、以下の内容のようなログメッセージでインポートに失敗します。

```
customresourcedefinition.apiextensions.k8s.io/klusterlets.operator.open-cluster-management.io
configured
clusterrole.rbac.authorization.k8s.io/klusterlet configured
clusterrole.rbac.authorization.k8s.io/open-cluster-management:klusterlet-admin-aggregate-clusterrole
configured
clusterrolebinding.rbac.authorization.k8s.io/klusterlet configured
```

```
namespace/open-cluster-management-agent configured
secret/open-cluster-management-image-pull-credentials unchanged
serviceaccount/klusterlet configured
deployment.apps/klusterlet unchanged
klusterlet.operator.open-cluster-management.io/klusterlet configured
Error from server (BadRequest): error when creating "STDIN": Secret in version "v1" cannot be
handled as a Secret:
v1.Secret.ObjectMeta:
v1.ObjectMeta.TypeMeta: Kind: Data: decode base64: illegal base64 data at input byte 1313, error
found in #10 byte of ...|dhruy45="},"kind":"|..., bigger context
...|tye56u56u568yuo7i67i67i67o556574i"},"kind":"Secret","metadata":{"annotations":{"kube|...
```

1.11.2. 問題の特定: OpenShift Container Platform バージョン 3.11 クラスターのインポートに失敗する

この問題は多くの場合、インストールされている **kubectl** コマンドラインツールのバージョンが 1.11 以前であるために発生します。以下のコマンドを実行して、実行中の **kubectl** コマンドラインツールのバージョンを表示します。

```
kubectl version
```

返されたデータがバージョンが 1.11 以前の場合は、**問題の解決: OpenShift Container Platform バージョン 3.11 クラスターのインポートに失敗する** に記載される修正のいずれかを実行します。

1.11.3. 問題の解決: OpenShift Container Platform バージョン 3.11 クラスターのインポートに失敗する

この問題は、以下のいずれかの手順を実行して解決できます。

- 最新バージョンの **kubectl** コマンドラインツールをインストールします。
  1. **kubectl** ツールの最新バージョンを、Kubernetes ドキュメントの kubectl のインストールとセットアップ からダウンロードします。
  2. **kubectl** ツールのアップグレード後にクラスターを再度インポートします。
- import コマンドが含まれるファイルを実行します。
  1. CLI を使用したマネージドクラスターのインポート の手順を開始します。
  2. クラスターのインポートコマンドを作成する場合には、このインポートコマンドを **import.yaml** という名前の YAML ファイルにコピーします。
  3. 以下のコマンドを実行して、ファイルからクラスターを再度インポートします。

     ```
     oc apply -f import.yaml
     ```

1.12. 証明書を変更した後のインポート済みクラスターのオフラインでのトラブルシューティング

カスタムの **apiserver** 証明書のインストールはサポートされますが、証明書情報を変更する前にインポートされたクラスターの 1 つまたは複数でステータスが **offline** になる可能性があります。

1.12.1. 現象: 証明書の変更後にクラスターがオフラインになる

証明書シークレットの更新手順を完了すると、オンラインになっていたクラスターの 1 つまたは複数が、Red Hat Advanced Cluster Management for Kubernetes コンソールで **offline** ステータスと表示される可能性があります。

1.12.2. 問題の特定: 証明書の変更後にクラスターがオフラインになる

カスタムの API サーバー証明書の情報を更新すると、インポートされ、新しい証明書が追加される前に稼働していたクラスターのステータスが **offline** になります。

オフラインのマネージドクラスターの **open-cluster-management-agent** namespace にある Pod のログで、証明書に問題があるとのエラーが見つかります。以下の例のようなエラーがログに表示されます。

work-agent のログ:

```
E0917 03:04:05.874759       1 manifestwork_controller.go:179] Reconcile work test-1-klusterlet-addon-workmgr fails with err: Failed to update work status with err Get "https://api.aaa-ocp.dev02.location.com:6443/apis/cluster.management.io/v1/namespaces/test-1/manifestworks/test-1-klusterlet-addon-workmgr": x509: certificate signed by unknown authority
E0917 03:04:05.874887       1 base_controller.go:231] "ManifestWorkAgent" controller failed to sync "test-1-klusterlet-addon-workmgr", err: Failed to update work status with err Get "api.aaa-ocp.dev02.location.com:6443/apis/cluster.management.io/v1/namespaces/test-1/manifestworks/test-1-klusterlet-addon-workmgr": x509: certificate signed by unknown authority
E0917 03:04:37.245859       1 reflector.go:127] k8s.io/client-go@v0.19.0/tools/cache/reflector.go:156: Failed to watch *v1.ManifestWork: failed to list *v1.ManifestWork: Get "api.aaa-ocp.dev02.location.com:6443/apis/cluster.management.io/v1/namespaces/test-1/manifestworks?resourceVersion=607424": x509: certificate signed by unknown authority
```

registration-agent のログ:

```
I0917 02:27:41.525026       1 event.go:282] Event(v1.ObjectReference{Kind:"Namespace", Namespace:"open-cluster-management-agent", Name:"open-cluster-management-agent", UID:"", APIVersion:"v1", ResourceVersion:"", FieldPath:""}): type: 'Normal' reason: 'ManagedClusterAvailableConditionUpdated' update managed cluster "test-1" available condition to "True", due to "Managed cluster is available"
E0917 02:58:26.315984       1 reflector.go:127] k8s.io/client-go@v0.19.0/tools/cache/reflector.go:156: Failed to watch *v1beta1.CertificateSigningRequest: Get "https://api.aaa-ocp.dev02.location.com:6443/apis/cluster.management.io/v1/managedclusters?allowWatchBookmarks=true&fieldSelector=metadata.name%3Dtest-1&resourceVersion=607408&timeout=9m33s&timeoutSeconds=573&watch=true"": x509: certificate signed by unknown authority
E0917 02:58:26.598343       1 reflector.go:127] k8s.io/client-go@v0.19.0/tools/cache/reflector.go:156: Failed to watch *v1.ManagedCluster: Get "https://api.aaa-ocp.dev02.location.com:6443/apis/cluster.management.io/v1/managedclusters?allowWatchBookmarks=true&fieldSelector=metadata.name%3Dtest-1&resourceVersion=607408&timeout=9m33s&timeoutSeconds=573&watch=true": x509: certificate signed by unknown authority
E0917 02:58:27.613963       1 reflector.go:127] k8s.io/client-go@v0.19.0/tools/cache/reflector.go:156: Failed to watch *v1.ManagedCluster: failed to list *v1.ManagedCluster: Get "https://api.aaa-ocp.dev02.location.com:6443/apis/cluster.management.io/v1/managedclusters?allowWatchBookmarks=true&fieldSelector=metadata.name%3Dtest-1&resourceVersion=607408&timeout=9m33s&timeoutSeconds=573&watch=true"": x509: certificate signed by unknown authority
```

1.12.3. 問題の解決: 証明書の変更後にクラスターがオフラインになる

マネージドクラスターが **local-cluster** であるか、Red Hat Advanced Cluster Management for Kubernetes を使用して作成された場合、マネージドクラスターを再インポートするには、10 分以上待つ必要があります。

マネージドクラスターをすぐに再インポートするには、ハブクラスター上のマネージドクラスターのインポートシークレットを削除し、Red Hat Advanced Cluster Management を使用して再インポートします。以下のコマンドを実行します。

```
oc delete secret -n <cluster_name> <cluster_name>-import
```

<cluster_name> を、インポートするマネージドクラスターの名前に置き換えます。

Red Hat Advanced Cluster Management を使用してインポートされたマネージドクラスターを再インポートする場合は、以下の手順を実行してマネージドクラスターを再度インポートします。

1. ハブクラスターで、次のコマンドを実行してマネージドクラスターのインポートシークレットを再作成します。

   ```
   oc delete secret -n <cluster_name> <cluster_name>-import
   ```

   <cluster_name> を、インポートするマネージドクラスターの名前に置き換えます。

2. ハブクラスターで、次のコマンドを実行して、マネージドクラスターのインポートシークレットを YAML ファイルに公開します。

   ```
   oc get secret -n <cluster_name> <cluster_name>-import -ojsonpath='{.data.import\.yaml}' |
   base64 --decode  > import.yaml
   ```

   <cluster_name> を、インポートするマネージドクラスターの名前に置き換えます。

3. マネージドクラスターで、次のコマンドを実行して **import.yaml** ファイルを適用します。

   ```
   oc apply -f import.yaml
   ```

1.13. クラスターの削除後も NAMESPACE が残る

マネージドクラスターを削除すると、通常 namespace はクラスターの削除プロセスの一部として削除されます。まれに namespace は一部のアーティファクトが含まれた状態で残る場合があります。このような場合は、namaspace を手動で削除する必要があります。

1.13.1. 現象: クラスターの削除後も namespace が残る

マネージドクラスターの削除後に namespace が削除されません。

1.13.2. 問題の解決: クラスターの削除後も namespace が残る

namespace を手作業で削除するには、以下の手順を実行します。

1. 次のコマンドを実行して、<cluster_name> namespace に残っているリソースのリストを作成します。

   ```
   oc api-resources --verbs=list --namespaced -o name | grep -E
   ```

```
'^secrets|^serviceaccounts|^managedclusteraddons|^roles|^rolebindings|^manifestworks|^lease
|^managedclusterinfo|^appliedmanifestworks'|^clusteroauths' | xargs -n 1 oc get --show-kind -
-ignore-not-found -n <cluster_name>
```

cluster_name は、削除を試みたクラスターの namespace 名に置き換えます。

2. 以下のコマンドを入力してリストを編集し、ステータスが **Delete** ではないリストから特定したリソースを削除します。

```
oc edit <resource_kind> <resource_name> -n <namespace>
```

resource_kind は、リソースの種類に置き換えます。**resource_name** は、リソース名に置き換えます。**namespace** は、リソースの namespace に置き換えます。

3. メタデータで **finalizer** 属性の場所を特定します。
4. vi エディターの **dd** コマンドを使用して、Kubernetes 以外のファイナライザーを削除します。
5. **:wq** コマンドを入力し、リストを保存して **vi** エディターを終了します。
6. 以下のコマンドを入力して namespace を削除します。

```
oc delete ns <cluster-name>
```

cluster-name を、削除する namespace の名前に置き換えます。

1.14. クラスターのインポート時の AUTO-IMPORT-SECRET-EXISTS エラー

クラスターのインポートは、auto import secret exists というエラーメッセージで失敗します。

1.14.1. 現象: クラスターのインポート時の Auto-import-secret-exists エラー

管理用のハイブクラスターをインポートすると、**auto-import-secret already exists** というエラーが表示されます。

1.14.2. 問題の解決: クラスターのインポート時の Auto-import-secret-exists エラー

この問題は、Red Hat Advanced Cluster Management で以前に管理されていたクラスターをインポートしようとすると発生します。これが生じると、クラスターを再インポートしようとすると、シークレットは競合します。

この問題を回避するには、以下の手順を実行します。

1. 既存の **auto-import-secret** を手動で削除するには、ハブクラスターで以下のコマンドを実行します。

```
oc delete secret auto-import-secret -n <cluster-namespace>
```

namespace は、お使いのクラスターの namespace に置き換えます。

2. ハブクラスターへのターゲットのマネージドクラスターのインポート の手順を使用して、クラスターを再度インポートします。

クラスターがインポートされました。

1.15. クラスターのステータスが OFFLINE から AVAILABLE に変わる場合のトラブルシューティング

マネージドクラスターのステータスは、環境またはクラスターを手動で変更することなく、**offline** と **available** との間で切り替わります。

1.15.1. 現象: クラスターのステータスが offline から available に変わる

マネージドクラスターからハブクラスターへのネットワーク接続が不安定な場合に、マネージドクラスターのステータスが **offline** と **available** との間で順に切り替わると、ハブクラスターにより報告されます。

1.15.2. 問題の解決: クラスターのステータスが offline から available に変わる

この問題を解決するには、以下の手順を実行します。

1. 次のコマンドを入力して、ハブクラスターで **ManagedCluster** の仕様を編集します。

   ```
   oc edit managedcluster <cluster-name>
   ```

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

2. **ManagedCluster** 仕様の **leaseDurationSeconds** の値を増やします。デフォルト値は 5 分ですが、ネットワークの問題がある状態で接続を維持するには十分でない場合があります。リースの時間を長く指定します。たとえば、設定を 20 分に増やします。

1.16. ステータスが PENDING または FAILED のクラスターのコンソールでのトラブルシューティング

作成してたクラスターのステータスがコンソールで Pending または Failed と表示されている場合は、以下の手順を実行して問題のトラブルシューティングを実行します。

1.16.1. 現象: コンソールでステータスが Pending または Failed のクラスターのトラブルシューティング

Red Hat Advanced Cluster Management for Kubernetes コンソールで新規クラスターを作成した後に、コンソールでクラスターのステータスが **Pending** または **Failed** と表示されてそれ以上進みません。

1.16.2. 問題の特定: コンソールでステータスが Pending または Failed のクラスター

クラスターのステータスが **Failed** と表示される場合は、クラスターの詳細ページに移動して、提供されたログへのリンクに進みます。ログが見つからない場合や、クラスターのステータスが **Pending** と表示される場合は、以下の手順を実行してログを確認します。

- 手順 1

  1. ハブクラスターで以下のコマンドを実行し、新規クラスターの namespace に作成した Kubernetes Pod の名前を表示します。

     ```
     oc get pod -n <new_cluster_name>
     ```

new_cluster_name は、作成したクラスター名に置き換えます。

2. 名前に **provision** の文字列が含まれる Pod が表示されていない場合は、手順 2 に進みます。タイトルに **provision** が含まれる Pod があった場合は、ハブクラスターで以下のコマンドを実行して、その Pod のログを表示します。

```
oc logs <new_cluster_name_provision_pod_name> -n <new_cluster_name> -c hive
```

new_cluster_name_provision_pod_name は、作成したクラスター名の後に **provision** が含まれる Pod 名を指定するように置き換えます。

3. ログでエラーを検索してください。この問題の原因が解明する場合があります。

- 手順 2
名前に **provision** が含まれる Pod がない場合は、問題がプロセスの初期段階で発生しています。ログを表示するには、以下の手順を実行します。

1. ハブクラスターで以下のコマンドを実行してください。

```
oc describe clusterdeployments -n <new_cluster_name>
```

new_cluster_name は、作成したクラスター名に置き換えます。クラスターのインストールログの詳細は、Red Hat OpenShift ドキュメントの インストールログの収集 を参照してください。

2. リソースの **Status.Conditions.Message** と **Status.Conditions.Reason** のエントリーに問題に関する追加の情報があるかどうかを確認します。

1.16.3. 問題の解決: コンソールでステータスが Pending または Failed のクラスター

ログでエラーを特定した後に、エラーの解決方法を決定してから、クラスターを破棄して、作り直してください。

以下の例では、サポート対象外のゾーンを選択している可能性を示すログエラーと、解決に必要なアクションが提示されています。

```
No subnets provided for zones
```

クラスターの作成時に、サポートされていないリージョンにあるゾーンを 1 つ以上選択しています。問題解決用にクラスターを再作成する時に、以下のアクションの 1 つを実行します。

- リージョン内の異なるゾーンを選択します。
- 他のゾーンをリストしている場合は、サポートを提供しないゾーンを省略します。
- お使いのクラスターに、別のリージョンを選択します。

ログから問題を特定した後に、クラスターを破棄し、再作成します。

クラスターの作成に関する詳細は、クラスターの作成 を参照してください。

1.17. アプリケーションの GIT サーバー接続のトラブルシューティング

open-cluster-management namespace のログでは、Git リポジトリーのクローンの失敗について表示されます。

1.17.1. 現象: Git サーバー接続

open-cluster-management namespace にあるサブスクリプションコントローラー Pod **multicluster-operators-hub-subscription-<random-characters>** のログに、Git リポジトリーのクローン作成に失敗したことが示されています。**x509: certificate signed by unknown authority** エラーまたは **BadGateway** エラーが表示されます。

1.17.2. 問題の解決: Git サーバー接続

重要: 以前のバージョンを使用している場合は、アップグレードしてください。

1. apps.open-cluster-management.io_channels_crd.yaml を同じファイル名として保存します。
2. Red Hat Advanced Cluster Management クラスターで以下のコマンドを実行し、ファイルを適用します。

   ```
   oc apply -f apps.open-cluster-management.io_channels_crd.yaml
   ```

3. **open-cluster-management** namespace で以下のコマンドを実行して **advanced-cluster-management.<version, example 2.5.0>** CSV を編集します。

   ```
   oc edit csv advanced-cluster-management.<version, example 2.5.0> -n open-cluster-management
   ```

   以下のコンテナーを見つけます。

   - **multicluster-operators-standalone-subscription**
   - **multicluster-operators-hub-subscription**
     コンテナーイメージを、使用するコンテナーに置き換えます。

     ```
     quay.io/open-cluster-management/multicluster-operators-subscription:<your image tag>
     ```

   この更新では **open-cluster-management** namespace に以下の Pod を作成します。

   - **multicluster-operators-standalone-subscription-<random-characters>**
   - **multicluster-operators-hub-subscription-<random-characters>**

4. 新たに作成した Pod が新規 docker イメージで実行していることを確認します。以下のコマンドを実行して、新しい docker イメージを見つけます。

   ```
   oc get pod multicluster-operators-standalone-subscription-<random-characters> -n open-cluster-management -o yaml
   oc get pod multicluster-operators-hub-subscription-<random-characters> -n open-cluster-management -o yaml
   ```

5. マネージドクラスターのイメージを更新します。
   ハブクラスターで、次のコマンドを実行して、**multicluster_operators_subscription** キーのイメージ値を使用するイメージに更新します。

   ```
   oc edit configmap -n open-cluster-management mch-image-manifest-<version, example 2.5.0>
   ...
   ```

```
data:
multicluster_operators_subscription: <your image with tag>
```

6. 既存の **multicluster-operators-hub-subscription** Pod を再起動します。

```
oc delete pods -n open-cluster-management multicluster-operators-hub-subscription--
<random-characters>
```

これにより、マネージドクラスターの **open-cluster-management-agent-addon** namespace に **application-manager-<random-characters>** Pod が再作成されます。

7. 新たに作成した Pod が新規 docker イメージで実行していることを確認します。

8. コンソールまたは CLI を使用してアプリケーションを作成する場合は、チャネル仕様に insecureSkipVerify: true を手動で追加します。以下の例を参照してください。

```
apiVersion: apps.open-cluster-management.io/v1
kind: Channel
metadata:
labels:
  name: sample-channel
  namespace: sample
spec:
  type: GitHub
  pathname: <Git URL>
  insecureSkipVerify: true
```

1.18. GRAFANA のトラブルシューティング

Grafana エクスプローラーで時間のかかるメトリクスをクエリーすると、**Gateway Time-out** エラーが発生する可能性があります。

1.18.1. 現象: Grafana エクスプローラーゲートウェイのタイムアウト

Grafana エクスプローラーで時間のかかるメトリクスをクエリーして **Gateway Time-out** エラーが発生する場合は、**open-cluster-management** namespace の **multicloud-console** ルートが原因でタイムアウトが発生する可能性があります。

1.18.2. 問題の解決: multicloud-console ルートの設定

この問題が発生した場合は、以下の手順を実行します。

1. Grafana のデフォルト設定に想定のタイムアウト設定があることを確認します。

   a. Grafana のデフォルトタイムアウト設定を確認するには、以下のコマンドを実行します。

```
oc get secret grafana-config -n open-cluster-management-observability -o jsonpath="
{.data.grafana\.ini}" | base64 -d | grep dataproxy -A 4
```

以下のタイムアウト設定が表示されるはずです。

```
[dataproxy]
timeout = 300
dial_timeout = 30
```

```
keep_alive_seconds = 300
```

b. Grafana のデフォルトのデータソースクエリータイムアウトを確認するには、以下のコマンドを実行します。

```
oc get secret/grafana-datasources -n open-cluster-management-observability -o
jsonpath="{.data.datasources\.yaml}" | base64 -d | grep queryTimeout
```

以下のタイムアウト設定が表示されるはずです。

```
queryTimeout: 300s
```

2. Grafana のデフォルト設定にタイムアウト設定が想定される場合には、以下のコマンドを実行して **open-cluster-management** namespace に **multicloud-console** ルートを設定できます。

```
oc annotate route multicloud-console -n open-cluster-management --overwrite
haproxy.router.openshift.io/timeout=300s
```

Grafana ページを更新し、再度メトリクスのクエリーを試行します。**Gateway Time-out** エラーが表示されなくなりました。

1.19. 配置ルールでローカルクラスターが選択されていない場合のトラブルシューティング

マネージドクラスターは配置ルールで選択されますが、**local-cluster** (同じく管理されているハブクラスター) は選択されません。配置ルールユーザーには、**local-cluster** namespace の **managedcluster** リソースを取得するためのパーミッションは付与されません。

1.19.1. 現象: ローカルクラスターがマネージドクラスターとして選択されていない問題のトラブルシューティング

すべてのマネージドクラスターは配置ルールで選択されますが、**local-cluster** は選択されません。配置ルールユーザーには、**local-cluster** namespace の **managedcluster** リソースを取得するためのパーミッションは付与されません。

1.19.2. 問題の解決: マネージドクラスターとして選択されていないローカルクラスターのトラブルシューティング

この問題を解決するには、**local-cluster** namespace に **managedcluster** 管理パーミッションを付与する必要があります。以下の手順を実行します。

1. マネージドクラスターの一覧に **local-cluster** が含まれていること、配置ルールの **decisions** リストで **local-cluster** が表示されていないことを確認します。以下のコマンドを実行して結果を表示します。

```
% oc get managedclusters
```

出力例を確認すると、**local-cluster** が結合されているものの、**PlacementRule** の YAML にないことが分かります。

```
NAME            HUB ACCEPTED   MANAGED CLUSTER URLS   JOINED   AVAILABLE
AGE
```

```
local-cluster   true                      True     True       56d
cluster1        true                      True     True       16h
```

```
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: all-ready-clusters
  namespace: default
spec:
  clusterSelector: {}
status:
  decisions:
  - clusterName: cluster1
    clusterNamespace: cluster1
```

2. YAML ファイルに **Role** を作成し、**local-cluster** namespace に **managedcluster** 管理パーミッションを付与します。以下の例を参照してください。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: managedcluster-admin-user-zisis
  namespace: local-cluster
rules:
- apiGroups:
  - cluster.open-cluster-management.io
  resources:
  - managedclusters
  verbs:
  - get
```

3. **RoleBinding** リソースを作成し、配置ルールユーザーに **local-cluster** namespace へのアクセスを許可します。以下の例を参照してください。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: managedcluster-admin-user-zisis
  namespace: local-cluster
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: managedcluster-admin-user-zisis
  namespace: local-cluster
subjects:
- kind: User
  name: zisis
  apiGroup: rbac.authorization.k8s.io
```

1.20. アプリケーションの KUBERNETES デプロイメントバージョンのトラブルシューティング

非推奨の Kubernetes **apiVersion** を使用するマネージドクラスターはサポートされない可能性があります。非推奨の API バージョンの詳細は、Kubernetes issue を参照してください。

1.20.1. 現象: アプリケーションデプロイメントバージョン

サブスクリプションの YAML ファイルのアプリケーションリソースの 1 つまたは複数で、非推奨の API が使用されている場合に、以下のようなエラーが発生する可能性があります。

```
failed to install release: unable to build kubernetes objects from release manifest: unable to recognize
"": no matches for
kind "Deployment" in version "extensions/v1beta1"
```

または、**old.yaml** などの名前で YAML ファイルに新しい Kubernetes API バージョンを追加している場合は、以下のエラーが発生する可能性があります。

```
error: unable to recognize "old.yaml": no matches for kind "Deployment" in version
"deployment/v1beta1"
```

1.20.2. 解決: アプリケーションデプロイメントバージョン

1. リソースの **apiVersion** を更新します。たとえば、サブスクリプション YAML ファイルの **Deployment** の種類のエラーが表示された場合は、**apiVersion** を **extensions/v1beta1** から **apps/v1** に更新する必要があります。
   以下の例を参照してください。

   ```
   apiVersion: apps/v1
   kind: Deployment
   ```

2. マネージドクラスターで以下のコマンドを実行して、利用可能なバージョンを確認します。

   ```
   kubectl explain <resource>
   ```

3. **VERSION** を確認します。

1.21. スタンドアロンのサブスクリプションメモリーのトラブルシューティング

メモリーの問題が原因で **multicluster-operators-standalone-subscription** Pod が定期的に再起動します。

1.21.1. 現象: スタンドアロンのサブスクリプションメモリー

Operator Lifecycle Manager (OLM) が全 Operator をデプロイすると、スタンドアロンのサブスクリプションコンテナーに十分なメモリーが割り当てられていないため、multicluster-subscription-operator だけでなく、**multicluster-operators-standalone-subscription** Pod も再起動します。

マルチクラスターサブスクリプションコミュニティー Operator CSV で **multicluster-operators-standalone-subscription** Pod のメモリーの上限が 2GB に増えましたが、OLM はこのリソース制限の設定を無視します。

1.21.2. 問題の解決: スタンドアロンのサブスクリプションメモリー

1. インストール後に、マルチクラスターサブスクリプションコミュニティー Operator をサブスクライブする Operator サブスクリプション CR を検索します。以下のコマンドを実行します。

```
% oc get sub -n open-cluster-management acm-operator-subscription
```

2. Operator サブスクリプションカスタムリソースを編集し、リソース制限を定義する **spec.config.resources .yaml** ファイルを追加します。
注記: 同じマルチクラスターサブスクリプションコミュニティー Operator をサブスクライブする Operator サブスクリプションのカスタムリソースを新たに作成しないでください。2 つの Operator サブスクリプションが 1 つの Operator にリンクされているため、この 2 つの Operator サブスクリプションカスタムリソースは、Operator Pod を **kill** して再起動します。

   以下の更新後の **.yaml** ファイルの例を参照してください。

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: multicluster-operators-subscription-alpha-community-operators-openshift-
marketplace
  namespace: open-cluster-management
spec:
  channel: release-2.2
  config:
    resources:
      limits:
        cpu: 750m
        memory: 2Gi
      requests:
        cpu: 150m
        memory: 128Mi
  installPlanApproval: Automatic
  name: multicluster-operators-subscription
  source: community-operators
  sourceNamespace: openshift-marketplace
```

3. リソースの保存後に、スタンドアロンのサブスクリプション Pod が 2GB のメモリー上限で再起動されるようにします。以下のコマンドを実行します。

```
% oc get pods -n open-cluster-management multicluster-operators-standalone-subscription-
7c8cbf885f-c94kz -o yaml
```

```
apiVersion: v1
kind: Pod
...
spec:
  containers:
  - image: quay.io/open-cluster-management/multicluster-operators-subscription:community-
2.2
...
    resources:
      limits:
        cpu: 750m
        memory: 2Gi
      requests:
        cpu: 150m
        memory: 128Mi
```

```
...
status:
  qosClass: Burstable
```

1.22. 状態が DEGRADED の KLUSTERLET のトラブルシューティング

Klusterlet の状態が Degraded の場合は、マネージドクラスターの Klusterlet エージェントの状態を診断しやすくなります。Klusterlet の状態が Degraded になると、マネージドクラスターの Klusterlet エージェントで発生する可能性のあるエラーに対応する必要があります。Klusterlet の degraded の状態が **True** に設定されている場合は、以下の情報を参照します。

1.22.1. 現象: Klusterlet の状態が degraded である

マネージドクラスターで Klusterlet をデプロイした後に、**KlusterletRegistrationDegraded** または **KlusterletWorkDegraded** の状態が **True** と表示されます。

1.22.2. 問題の特定: Klusterlet の状態が degraded である

1. マネージドクラスターで以下のコマンドを実行して、Klusterlet のステータスを表示します

```
kubectl get klusterlets klusterlet -oyaml
```

2. **KlusterletRegistrationDegraded** または **KlusterletWorkDegraded** をチェックして、状態が **True** に設定されいるかどうかを確認します。記載されている Degraded の状態については、**問題の解決** に進みます。

1.22.3. 問題の解決: Klusterlet の状態が degraded である

ステータスが Degraded のリストおよびこれらの問題の解決方法を参照してください。

- **KlusterletRegistrationDegraded** の状態が **True** で、この状態の理由が **BootStrapSecretMissing** の場合は、**open-cluster-management-agent** namespace にブートストラップのシークレットを作成する必要があります。

- **KlusterletRegistrationDegraded** の状態が **True** と表示され、状態の理由が **BootstrapSecretError** または **BootstrapSecretUnauthorized** の場合は、現在のブートストラップシークレットが無効です。現在のブートストラップシークレットを削除して、**open-cluster-management-agent** namespace で有効なブートストラップシークレットをもう一度作成します。

- **KlusterletRegistrationDegraded** および **KlusterletWorkDegraded** が **True** と表示され、状態の理由が **HubKubeConfigSecretMissing** の場合は、Klusterlet を削除して作成し直します。

- **KlusterletRegistrationDegraded** および **KlusterletWorkDegraded** が **True** と表示され、状態の理由が **ClusterNameMissing**、**KubeConfigMissing**、**HubConfigSecretError**、または **HubConfigSecretUnauthorized** の場合は、**open-cluster-management-agent** namespace からハブクラスターの kubeconfig シークレットを削除します。登録エージェントは再度ブートストラップして、新しいハブクラスターの kubeconfig シークレットを取得します。

- **KlusterletRegistrationDegraded** が **True** と表示され、状態の理由が **GetRegistrationDeploymentFailed** または **UnavailableRegistrationPod** の場合は、状態のメッセージを確認して、問題の詳細を取得して解決してみてください。

- **KlusterletWorkDegraded** が **True** と表示され、状態の理由が **GetWorkDeploymentFailed** または **UnavailableWorkPod** の場合は、状態のメッセージを確認して、問題の詳細を取得し、解決してみてください。

1.23. マネージドクラスターでの KLUSTERLET アプリケーションマネージャーのトラブルシューティング

Red Hat Advanced Cluster Management for Kubernetes をアップグレードすると、Red Hat OpenShift Container Platform マネージドクラスターバージョン 4.5 および 4.6 上の **klusterlet-addon-appmgr** Pod が **OOMKilled** になります。

1.23.1. 現象: マネージドクラスター上の Klusterlet アプリケーションマネージャー

Red Hat OpenShift Container Platform マネージドクラスターのバージョン 4.5 および 4.6 上の **klusterlet-addon-appmgr** Pod で **OOMKilled** のエラーが発生します。

1.23.2. 問題の解決: マネージドクラスター上の Klusterlet アプリケーションマネージャー

Red Hat Advanced Cluster Management for Kubernetes 2.1.x および 2.2 の場合は、Pod のメモリー制限を手動で **8GB** に増やす必要があります。以下の手順を参照してください。

1. ハブクラスターで、 **klusterletaddonconfig** にアノテーションを付けてレプリケーションを一時停止します。以下のコマンドを使用します。

   ```
   oc annotate klusterletaddonconfig -n ${CLUSTER_NAME} ${CLUSTER_NAME}
   klusterletaddonconfig-pause=true --  overwrite=true
   ```

2. ハブクラスターで、 **klusterlet-addon-operator** をスケールダウンします。以下のコマンドを使用します。

   ```
   oc edit manifestwork ${CLUSTER_NAME}-klusterlet-addon-operator -n ${CLUSTER_NAME}
   ```

3. **klusterlet-addon-operator** デプロイメントを探し、その仕様に **replicas: 0** を追加してスケールダウンします。

   ```
   - apiVersion: apps/v1
     kind: Deployment
     metadata:
       labels:
         app: cluster1
       name: klusterlet-addon-operator
       namespace: open-cluster-management-agent-addon
       spec:
         replicas: 0
   ```

   マネージドクラスターでは、 **open-cluster-management-agent-addon** / **klusterlet-addon-operator** Pod が終了します。

4. マネージドクラスターにログインして、**appmgr** Pod のメモリー制限を手動で増やします。以下のコマンドを実行します。

   ```
   % oc edit deployments -n open-cluster-management-agent-addon klusterlet-addon-appmgr
   ```

たとえば、制限が 5G の場合は、8G に増やします。

```
resources:
  limits:
    memory: 2Gi  -> 8Gi
  requests:
    memory: 128Mi -> 256Mi
```

1.24. オブジェクトストレージチャネルシークレットのトラブルシューティング

SecretAccessKey を変更すると、オブジェクトストレージチャネルのサブスクリプションは、更新されたシークレットを自動的に取得できず、エラーが発生します。

1.24.1. 現象: オブジェクトストレージチャネルシークレット

オブジェクトストレージチャネルのサブスクリプションは、更新されたシークレットを自動的に取得できません。そのため、サブスクリプションオペレーターが調整できなくなり、オブジェクトストレージからマネージドクラスターにリソースがデプロイされなくなります。

1.24.2. 問題の解決: オブジェクトストレージチャネルシークレット

シークレットを作成するには、認証情報を手動で入力してから、チャネル内のシークレットを参照する必要があります。

1. サブスクリプションオペレーターに単一の調整を生成するために、サブスクリプション CR にアノテーションを付けます。以下の **data** 仕様を参照してください。

```
apiVersion: apps.open-cluster-management.io/v1
kind: Channel
metadata:
  name: deva
  namespace: ch-obj
  labels:
    name: obj-sub
spec:
  type: ObjectBucket
  pathname: http://ec2-100-26-232-156.compute-1.amazonaws.com:9000/deva
  sourceNamespaces:
    - default
  secretRef:
    name: dev
---
apiVersion: v1
kind: Secret
metadata:
  name: dev
  namespace: ch-obj
  labels:
    name: obj-sub
data:
  AccessKeyID: YWRtaW4=
  SecretAccessKey: cGFzc3dvcmRhZG1pbg==
```

2. **oc annotate** を実行してテストします。

```
oc annotate appsub -n <subscription-namespace> <subscription-name> test=true
```

コマンドの実行後に、アプリケーションコンソールに移動して、リソースがマネージドクラスターにデプロイされていることを確認してください。または、マネージドクラスターにログインして、アプリケーションリソースが特定の namespace で作成されているかどうかを確認できます。

1.25. 可観測性のトラブルシューティング

可観測性コンポーネントをインストールした後に、コンポーネントが停止し、**Installing** のステータスが表示されます。

1.25.1. 現象: MultiClusterObservability リソースの状態が停止する

可観測性のカスタムリソース定義 (CRD) をインストールして作成した後に可観測性のステータスが **Installing** で停止すると、**spec:storageConfig:storageClass** パラメーターに値が定義されていない場合があります。または、可観測性コンポーネントは自動的にデフォルトの **storageClass** を検索しますが、ストレージの値が指定されていないと、コンポーネントが **Installing** ステータスのままになります。

1.25.2. 問題の解決: MultiClusterObservability リソースの状態が停止する

この問題が発生した場合は、以下の手順を実行します。

1. 可観測性コンポーネントがインストールされていることを確認します。
   a. **multicluster-observability-operator** を確認するには、以下のコマンドを実行します。

```
kubectl get pods -n open-cluster-management|grep observability
```

   b. 適切な CRD が存在することを確認するには、以下のコマンドを実行します。

```
kubectl get crd|grep observ
```

   以下の CRD が表示されるまで、コンポーネントを有効化しないでください。

```
multiclusterobservabilities.observability.open-cluster-management.io
observabilityaddons.observability.open-cluster-management.io
observatoria.core.observatorium.io
```

2. Bare Metal クラスター用に独自の storageClass を作成する場合は、How to create an NFS provisioner in the cluster or out of the cluster を参照してください。

3. 可観測性コンポーネントがデフォルトの storageClass を検索できるようにするには、**multicluster-observability-operator** CRD の **storageClass** パラメーターを更新します。パラメーターは、以下のような値になります。

```
storageclass.kubernetes.io/is-default-class: "true"
```

インストールが完了すると、可観測性コンポーネントのステータスが **Ready** に更新されます。インストールの完了に失敗すると、ステータスが **Fail** と表示されます。

1.26. OPENSHIFT モニターリングサービスのトラブルシューティング

マネージドクラスターの可観測性サービスは OpenShift Container Platform モニターリングスタックからメトリクスを収集する必要があります。**metrics-collector** は、OpenShift Container Platform モニターリングスタックが準備状態にならないと、インストールされません。

1.26.1. 現象: OpenShift モニターリングサービスが準備状態にならない

endpoint-observability-operator-x Pod は、**prometheus-k8s** サービスが **openshift-monitoring** namespace で利用可能かどうかを確認します。このサービスが **openshift-monitoring** namespace に存在しない場合は、**metrics-collector** はデプロイされません。**Failed to get prometheus resource** というエラーメッセージが表示される可能性があります。

1.26.2. 問題の解決: OpenShift モニターリングサービスが準備状態にならない

この問題が発生した場合は、以下の手順を実行します。

1. OpenShift Container Platform クラスターにログインします。
2. **openshift-monitoring** namespace にアクセスし、**prometheus-k8s** サービスが利用可能であることを確認します。
3. マネージドクラスターの **open-cluster-management-addon-observability** namespace で、**endpoint-observability-operator-x** Pod を再起動します。

1.27. 検索アグリゲーター POD のステータスのトラブルシューティング

search-aggregator の実行に失敗します。

1.27.1. 現象 1: Not Ready 状態の検索アグリゲーター Pod

redisgraph-user-secret が更新される場合、検索アグリゲーター Pod は **Not Ready** 状態にあります。以下のエラーが返される場合があります。

```
E0113 15:04:42.427931       1 pool.go:93] Error authenticating Redis client. Original error: ERR invalid password
W0113 15:04:42.428100       1 healthProbes.go:36] Unable to reach Redis.
E0113 15:04:44.265777       1 pool.go:93] Error authenticating Redis client. Original error: ERR invalid password
W0113 15:04:44.266003       1 healthProbes.go:36] Unable to reach Redis.
E0113 15:04:46.316869       1 pool.go:93] Error authenticating Redis client. Original error: ERR invalid password
W0113 15:04:46.317029       1 healthProbes.go:36] Unable to reach Redis.
```

1.27.2. 問題の解決: Not Ready 状態の検索アグリゲーター Pod

この問題が発生した場合は、**search-aggregator** Pod および **search-api** Pod を削除して Pod を再起動します。以下のコマンドを実行して前述の Pod を削除します。

```
oc delete pod -n open-cluster-management <search-aggregator>

oc delete pod -n open-cluster-management <search-api>
```

1.27.3. 現象 2: Pending 状態の検索 redisgraph Pod

search-redisgraph Pod は、**Pending** 状態時では実行に失敗します。

1.27.4. 問題の解決: Pending 状態の検索 redisgraph Pod

この問題が発生した場合は、以下の手順を実行します。

1. 以下のコマンドを実行し、ハブクラスター namespace の Pod イベントを確認します。

```
oc describe pod search-redisgraph-0
```

2. **searchcustomization** CR を作成した場合は、ストレージクラスおよびストレージサイズが有効かどうかを確認し、PVC を作成できるかどうかを確認します。以下のコマンドを実行して PVC を一覧表示します。

```
oc get pvc  <storageclassname>-search-redisgraph-0
```

3. PVC を **search-redisgraph-0** Pod にバインドできることを確認します。問題が解決されない場合は、StatefulSet **search-redisgraph** を削除します。検索 Operator は StatefulSet を再作成します。以下のコマンドを実行します。

```
oc delete statefulset -n open-cluster-management search-redisgraph
```

1.28. METRICS-COLLECTOR のトラブルシューティング

マネージドクラスターで **observability-client-ca-certificate** シークレットが更新されないと、内部サーバーのエラーが発生する可能性があります。

1.28.1. 現象: metrics-collector が observability-client-ca-certificate を検証できない

メトリクスを利用できないマネージドクラスターが存在する可能性があります。この場合は、**metrics-collector** デプロイメントから以下のエラーが発生することがあります。

```
error: response status code is 500 Internal Server Error, response body is x509: certificate signed by unknown authority (possibly because of "crypto/rsa: verification error" while trying to verify candidate authority certificate "observability-client-ca-certificate")
```

1.28.2. 問題の解決: metrics-collector が observability-client-ca-certificate を検証できない

この問題が発生した場合は、以下の手順を実行します。

1. ターゲットのマネージドクラスターにログインします。

2. **open-cluster-management-addon-observability** namespace にある **observability-controller-open-cluster-management.io-observability-signer-client-cert** というシークレットを削除します。以下のコマンドを実行します。

```
oc delete observability-controller-open-cluster-management.io-observability-signer-client-cert -n open-cluster-management-addon-observability
```

注記: **observability-controller-open-cluster-management.io-observability-signer-client-cert** は、新しい証明書で自動的に再作成されます。

metrics-collector-deployment デプロイメントが再度作成され、**observability-controller-open-cluster-management.io-observability-signer-client-cert** シークレットが更新されます。

1.29. インストール後に SUBMARINER が接続されない場合のトラブルシューティング - 一般情報

Submariner を設定しても正しく実行されない場合は、問題を特定して解決するためにできることがいくつかあります。

1.29.1. 現象: インストール後に Submariner が接続されない - 一般情報

インストール後、Submariner ネットワークが通信していません。

1.29.2. 問題の特定: インストール後に Submariner が接続されない - 一般情報

Submariner のデプロイ後にネットワーク接続が確立されない場合は、トラブルシューティング手順を開始します。Submariner をデプロイすると、プロセスが完了するまでに数分かかる場合があることに注意してください。

1.29.3. 問題の解決: インストール後に Submariner が接続されない - 一般情報

デプロイメント後に Submariner が正しく実行されない場合、問題の診断に使用できるトラブルシューティングの手順とリソースがいくつかあります。

1. Submariner のコンポーネントが正しく展開されているかどうかを判断するには、次の要件を確認してください。
   - **submariner-addon** Pod は、ハブクラスターの **open-cluster-management** namespace で実行されています。
   - 次の Pod は、各マネージドクラスターの **submariner-operator** namespace で実行されています。
     - submariner-addon
     - submariner-gateway
     - submariner-routeagent
     - submariner-operator
     - submariner-globalnet (ClusterSet で Globalnet が有効になっている場合のみ)
     - submariner-lighthouse-agent
     - Submariner-lighthouse-coredns
2. **subctl diagnose** コマンドを実行して、**submariner-addon** Pod を除く必要な Pod のステータスを確認します。
3. マネージドクラスターで **subctl gather** コマンドを実行して、**submariner-addon** Pod を除くさまざまな Submariner Pod のログを収集します。

4. 問題を開きます。他の手順で問題が特定されない場合は、次の情報を使用して問題を開きます。

    a. subctl gather を実行して、関連するすべての Submariner ログを収集し、問題に追加します。

    b. ハブクラスター上の **ManagedCluster** namespace から、**ManagedClusterAddon** リソースタイプの **submariner** インスタンス、および、**SubmarinerConfig** リソースタイプの **submariner** インスタンスの情報を取得する。

    c. 問題に次の情報とその他のテンプレート情報を提供します。

        - なにが起きていますか？
        - 何が起こると想定していましたか？
        - どのようにそれを再現しますか (可能な限り最小限かつ正確に)?
        - 他に把握しておくべきことはありますか？
        - 環境情報:
            - Submariner バージョン (**subctl version** コマンドを使用)
            - Kubernetes バージョン (**kubectl version** コマンドを使用)
            - 収集された情報を診断します (**subctl diagnose all** コマンドを使用します)
            - 情報を収集します (**subctl gather** コマンドを使用します)
            - クラウドプロバイダーまたはハードウェア設定:
                - OS (**cat /etc/os-release** コマンドを使用)
                - カーネル (**uname -a** コマンドを使用)
            - ツールをインストールする
            - 役立つかもしれない他の環境情報

1.30. SUBMARINER アドオンのステータスが低下してい場合のトラブルシューティング

クラスターセット内のクラスターに Submariner アドオンを追加すると、**Connection status**、**Agent status**、および **Gateway nodes** のステータスにクラスターの予期しないステータスが表示されます。

1.30.1. 現象: Submariner のアドオンステータスが低下している

クラスターセット内のクラスターに Submariner アドオンを追加すると、クラスターの **Gateway nodes**、**Agent status**、および **Connection status** に次のステータスが表示されます。

- ラベルの付いたゲートウェイノード
    - **Progressing**: ゲートウェイノードにラベルを付けるプロセスが開始されました。
    - **Nodes not labeled**: ゲートウェイノードはラベル付けされていません。おそらく、それらにラベルを付けるプロセスが完了していないためです。

- **Nodes not labeled**: おそらくプロセスが別のプロセスの終了を待機しているため、ゲートウェイノードはまだラベル付けされていません。
- Nodes labeled: ゲートウェイノードにはラベルが付けられています。

- エージェントのステータス
  - Progressing: Submariner エージェントのインストールが開始されました。
  - Degraded: Submariner エージェントが正しく実行されていません。おそらく、まだ進行中です。
- 接続状態
  - Progressing: Submariner アドオンとの接続を確立するプロセスが開始されました。
  - Degraded: 接続の準備ができていません。アドオンをインストールしたばかりの場合は、プロセスがまだ進行中である可能性があります。接続がすでに確立されて実行された後である場合は、2 つのクラスターが相互に接続を失っています。複数のクラスターがある場合、いずれかのクラスターが切断状態にあると、すべてのクラスターに **Degraded** ステータスが表示されます。

また、接続されているクラスターと切断されているクラスターも表示されます。

1.30.2. 問題の解決: Submariner のアドオンステータスが低下している

- degraded ステータスは、プロセスが完了すると自動的に解決することがよくあります。表のステータスをクリックすると、プロセスの現在のステップを確認できます。その情報を使用して、プロセスが終了したかどうかを判断し、他のトラブルシューティング手順を実行する必要があります。
- それ自体で解決しない問題の場合は、次の手順を実行して問題のトラブルシューティングを行います。

1. 次の条件が存在する場合、**subctl** ユーティリティーで **diagnose** コマンドを使用して、Submariner 接続でいくつかのテストを実行できます。
   a. **Agent status** または **Connection status** は **Degraded** 状態です。**diagnose** コマンドは、問題に関する詳細な分析を提供します。
   b. コンソールではすべてが緑色ですが、ネットワーク接続が正しく機能していません。**diagnose** コマンドは、コンソールの外部に他の接続またはデプロイメントの問題がないことを確認するのに役立ちます。問題を特定するために、デプロイメント後に **diagnostics** コマンドを実行することをお勧めします。
   コマンドの実行方法の詳細は、Submariner の **diagnose** を参照してください。
2. **Connection status** で問題が続く場合は、**subctl** ユーティリティーツールの **diagnose** コマンドを実行して、2 つの Submariner クラスター間の接続のより詳細なステータスを取得することから始めることができます。コマンドの形式は次のとおりです。

```
subctl diagnose all --kubeconfig <path-to-kubeconfig-file>
```

   path-to-kubeconfig-file を **kubeconfig** ファイルへのパスに置き換えます。コマンドの詳細については、Submariner のドキュメントの **diagnose** を参照してください。
3. ファイアウォールの設定を確認してください。接続の問題は、クラスターの通信を妨げるファイアウォールのアクセス許可の問題が原因で発生する場合があります。これによ

り、**Connection status** が degraded として表示される可能性があります。次のコマンドを実行して、ファイアウォールの問題を確認します。

```
subctl diagnose firewall inter-cluster <path-to-local-kubeconfig> <path-to-remote-cluster-kubeconfig>
```

path-to-local-kubeconfig を、いずれかのクラスターの **kubeconfig** ファイルへのパスに置き換えます。

path-to-remote-kubeconfig を、他のクラスターの **kubeconfig** ファイルへのパスに置き換えます。**subctl** ユーティリティーツールで **verify** コマンドを実行して、2 つの Submariner クラスター間の接続をテストできます。コマンドの基本的な形式は次のとおりです。

4. **Connection status** で問題が続く場合は、**subctl** ユーティリティーツールで **verify** コマンドを実行して、2 つの Submariner クラスター間の接続をテストできます。コマンドの基本的な形式は次のとおりです。

```
subctl verify --kubecontexts <cluster1>,<cluster2> [flags]
```

cluster1 と **cluster2** を、テストしているクラスターの名前に置き換えます。コマンドの詳細については、Submariner のドキュメントの **verify** を参照してください。

5. トラブルシューティング手順で問題が解決したら、**subctl** ツールで **benchmark** コマンドを使用して、追加の診断を実行するときに比較する基準を確立します。
コマンドのオプションの詳細は、Submariner のドキュメントの **benchmark** を参照してください。