Red Hat Advanced Cluster Management for Kubernetes 2.10

문제 해결

문제 해결

Last Updated: 2024-09-05

Red Hat Advanced Cluster Management for Kubernetes 2.10 문제 해결

문제 해결

법적 공지

초록

클러스터의 문제 해결 주제 목록을 확인합니다. must-gather 명령을 사용하여 로그를 수집할 수도 있습니다.

차례

1장. 문제 해결

문제 해결 가이드를 사용하기 전에 **oc adm must-gather** 명령을 실행하여 세부 정보, 로그 및 디버깅 문제 단계를 수행할 수 있습니다. 자세한 내용은 must-gather 명령 실행을 참조하십시오.

또한 역할 기반 액세스를 확인합니다. 자세한 내용은 역할 기반 액세스 제어를 참조하십시오.

1.1. 문서화된 문제 해결

Red Hat Advanced Cluster Management for Kubernetes의 문제 해결 주제 목록을 확인하십시오.

설치

설치 작업의 기본 문서를 보려면 설치 및 업그레이드 를 참조하십시오.

- 설치 또는 보류 중인 설치 상태 문제 해결
- 설치 실패 문제 해결
- Red Hat Advanced Cluster Management 업그레이드 후 ocm-controller 오류 문제 해결

백업 및 복원

백업 및 복원에 대한 주요 문서를 보려면 백업 및 복원을 참조하십시오.

- 복원 상태 문제 해결 오류와 함께 완료
- hub 클러스터 백업을 복원할 때 일반 리소스가 제거됨

클러스터 관리

클러스터 관리에 대한 주요 문서를 보려면 다중 클러스터 엔진 Operator 클러스터 라이프사이클 개요 를 참조하십시오.

- 오프라인 클러스터 문제 해결
- 관리형 클러스터 가져오기 실패 문제 해결
- 가져오기 보류 중 상태의 클러스터 문제 해결
- 인증서 변경 후 가져온 클러스터 오프라인 문제 해결
- 클러스터 상태가 오프라인에서 사용 가능으로 변경 문제 해결
- VMware vSphere에서 클러스터 생성 문제 해결
- 보류 중 또는 실패 상태의 콘솔에서 클러스터 문제 해결
- OpenShift Container Platform 버전 3.11 클러스터 가져오기 실패 문제 해결
- 성능이 저하된 조건으로 Klusterlet 문제 해결
- 오브젝트 스토리지 채널 시크릿 문제 해결
- 클러스터를 삭제한 후에도 네임스페이스가 남아 있음
- 클러스터를 가져올 때 auto-import-secret-exists 오류

- volSync용 CSI(Container Storage Interface) 드라이버 문제 해결
- 애드온 허용 오차를 무시하고 클러스터 프록시 애드온 문제 해결

다중 클러스터 글로벌 허브

다중 클러스터 글로벌 허브에 대한 주요 문서를 보려면 {global-hub.

- must-gather 명령을 실행하여 문제 해결
- 문제 해결을 위해 프로비저닝된 postgres 데이터베이스에 액세스
- 문제 해결을 위해 데이터베이스 덤프 및 복원 사용
- 규정 준수 데이터 복원

애플리케이션 관리

애플리케이션 관리에 대한 기본 문서를 보려면 애플리케이션 관리를 참조하십시오. ../../html-single/applications#managing-applications

- 애플리케이션 Kubernetes 배포 버전 문제 해결
- 로컬 클러스터가 선택되지 않은 문제 해결

거버넌스

- 여러 줄 YAML 구문 분석 문제 해결

보안 가이드를 보려면 위험 및 규정 준수를 참조하십시오.

콘솔 관찰 기능

콘솔 관찰 기능에는 헤더 및 탐색 기능과 함께 검색이 포함됩니다. 관찰 기능 가이드를 보려면 콘솔의 Observability를 참조하십시오.

- grafana 문제 해결
- 관찰 기능 문제 해결
- OpenShift 모니터링 서비스 문제 해결
- metrics-collector 문제 해결
- PostgreSQL 공유 메모리 오류 문제 해결
- Thanos compactor의 블록 오류 문제 해결

Submariner 네트워킹 및 서비스 검색

이 섹션에는 Red Hat Advanced Cluster Management 또는 다중 클러스터 엔진 Operator와 함께 Submariner를 사용할 때 발생할 수 있는 하위 문제 해결 절차가 나열되어 있습니다. 일반적인 Submariner 문제 해결 정보는 Submariner 문서의 문제 해결을 참조하십시오.

Submariner 네트워킹 서비스 및 서비스 검색에 대한 기본 문서를 보려면 Submariner 다중 클러스터 네트워킹 및 서비스 검색을 참조하십시오.

- 설치 후 Submariner가 연결되지 않음 문제 해결 - 일반 정보

- Submariner 애드온 상태 문제 해결

1.2. MUST-GATHER 명령을 실행하여 문제 해결

문제 해결을 시작하려면 사용자가 **must-gather** 명령을 실행하여 문제를 디버깅하는 데 필요한 문제 해결 시나리오에 대해 확인한 다음 명령 사용을 시작하는 절차를 참조하십시오.

필수 액세스: 클러스터 관리자

1.2.1. must-gather 시나리오

- **시나리오 1: 문서화된 문제 해결 섹션을 사용하여 문제에 대한 해결 방법이 문서화되어 있는지 확인합니다.** 이 가이드는 제품의 주요 기능에 의해 구성됩니다.
  이 시나리오에서는 가이드가 설명서에 있는지 확인합니다. 예를 들어 클러스터 생성에 문제가 있는 경우 *클러스터 관리* 섹션에서 솔루션을 찾을 수 있습니다.

- **시나리오 2:** 해결 단계에 문제가 문서화되지 않은 경우 **must-gather** 명령을 실행하고 출력을 사용하여 문제를 디버깅합니다.

- **시나리오 3: must-gather** 명령의 출력을 사용하여 문제를 디버깅할 수 없는 경우 Red Hat 지원과 출력을 공유하십시오.

1.2.2. must-gather 절차

must-gather 명령을 사용하려면 다음 절차를 참조하십시오.

1. **must-gather** 명령에 대해 알아보고 Red Hat OpenShift Container Platform 설명서에서 클러스터에 대한 데이터 가져오기에서 필요한 사전 요구 사항을 설치합니다.

2. 클러스터에 로그인합니다. 데이터 및 디렉터리 수집에 사용되는 Red Hat Advanced Cluster Management for Kubernetes 이미지를 추가합니다. 다음 명령을 실행하여 출력 이미지와 디렉터리를 삽입합니다.

   ```
   oc adm must-gather --image=registry.redhat.io/rhacm2/acm-must-gather-rhel9:v2.10 --dest-dir=<directory>
   ```

3. 일반적인 사용 사례의 경우 *hub* 클러스터에 로그인하는 동안 **must-gather** 를 실행해야 합니다. **참고:** 관리 클러스터를 확인하려면 **cluster-scoped-resources** 디렉터리에 있는 **gather-managed.log** 파일을 찾습니다.

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

   JOINED 및 AVAILABLE 열에 **True** 가 설정되지 않은 관리형 클러스터를 확인합니다. **True** 상태와 연결되지 않은 클러스터에서 **must-gather** 명령을 실행할 수 있습니다.

4. 지정된 디렉터리로 이동하여 다음 수준에서 구성된 출력을 확인합니다.

   - 두 개의 피어 수준: **cluster-scoped-resources** 및 **namespace** resources.

   - 각 하위 수준: 클러스터 범위 및 네임스페이스 범위 리소스 모두에 대한 사용자 정의 리소스 정의에 대한 API 그룹입니다.

   - 유형별로 정렬된 YAML 파일 각의 다음 수준 .

1.2.3. 연결이 끊긴 환경의 must-gather

연결이 끊긴 환경에서 **must-gather** 명령을 실행하려면 다음 단계를 완료합니다.

1. 연결이 끊긴 환경에서 **Red Hat Operator** 카탈로그 이미지를 미러 레지스트리에 미러링합니다. 자세한 내용은 연결이 끊긴 네트워크에 설치를 참조하십시오.

2. 다음 명령을 실행하여 모든 정보를 수집하여 **< 2.x >**를 **< acm-must-gather> (예: 2.10)** 및 **< multicluster- engine/must-gather > (예: 2.5)** 모두에 대해 지원되는 버전으로 바꿉니다.

```
REGISTRY=<internal.repo.address:port>
IMAGE1=$REGISTRY/rhacm2/acm-must-gather-rhel9:v<2.x>
oc adm must-gather --image=$IMAGE1 --dest-dir=<directory>
```

현재 지원되는 릴리스 중 하나 또는 제품 문서에 문제가 발생하는 경우 **Red Hat** 지원팀으로 이동하여 추가 문제를 해결하거나 기술 자료 문서를 보거나 지원 팀과 연결하거나 케이스를 열 수 있습니다. **Red Hat** 인증 정보를 사용하여 로그인해야 합니다.

1.2.4. 호스트 클러스터의 must-gather

호스팅된 컨트롤 플레인 클러스터에 문제가 발생하는 경우 **must-gather** 명령을 실행하여 문제 해결에 도움이 되는 정보를 수집할 수 있습니다.

1.2.4.1. 호스팅된 클러스터의 must-gather 명령 정보

명령은 관리 클러스터 및 호스팅된 클러스터에 대한 출력을 생성합니다.

- 다중 클러스터 엔진 **Operator** 허브 클러스터의 데이터:
  - 클러스터 범위 리소스: 이러한 리소스는 관리 클러스터의 노드 정의입니다.
  - **hypershift-dump** 압축 파일: 이 파일은 다른 사용자와 콘텐츠를 공유해야 하는 경우에 유용합니다.
  - 네임스페이스 리소스: 이러한 리소스에는 구성 맵, 서비스, 이벤트 및 로그와 같은 관련 네임스페이스의 모든 오브젝트가 포함됩니다.

- 네트워크 로그: 이 로그에는 OVN northbound 및 southbound 데이터베이스와 각각에 대한 상태가 포함됩니다.
- 호스트 클러스터: 이 수준의 출력에는 호스팅된 클러스터 내부의 모든 리소스가 포함됩니다.

- 호스트 클러스터의 데이터:
  - 클러스터 범위 리소스: 이러한 리소스에는 노드 및 CRD와 같은 모든 클러스터 전체 오브젝트가 포함됩니다.
  - 네임스페이스 리소스: 이러한 리소스에는 구성 맵, 서비스, 이벤트 및 로그와 같은 관련 네임스페이스의 모든 오브젝트가 포함됩니다.

출력에 클러스터의 보안 오브젝트가 포함되어 있지 않지만 시크릿 이름에 대한 참조를 포함할 수 있습니다.

1.2.4.2. 사전 요구 사항

must-gather 명령을 실행하여 정보를 수집하려면 다음 사전 요구 사항을 충족해야 합니다.

- kubeconfig 파일이 로드되고 다중 클러스터 엔진 Operator 허브 클러스터를 가리키는지 확인해야 합니다.
- 다중 클러스터 엔진 Operator 허브 클러스터에 대한 cluster-admin 액세스 권한이 있어야 합니다.
- HostedCluster 리소스의 name 값과 사용자 정의 리소스가 배포된 네임스페이스가 있어야 합니다.

1.2.4.3. 호스트된 클러스터에 대한 must-gather 명령 입력

1. 호스트된 클러스터에 대한 정보를 수집하려면 다음 명령을 입력합니다. 명령에서 hosted-

cluster-namespace=HOSTEDCLUSTERNAMESPACE 매개변수는 선택 사항입니다. 포함하지 않으면 호스트 클러스터가 기본 네임스페이스인 것처럼 명령이 실행됩니다.

```
oc adm must-gather --image=quay.io/stolostron/backplane-must-gather:SNAPSHOTNAME
/usr/bin/gather hosted-cluster-namespace=HOSTEDCLUSTERNAMESPACE hosted-cluster-
name=HOSTEDCLUSTERNAME
```

2. 명령 결과를 압축 파일에 저장하려면 **NAME** 을 결과를 저장하려는 디렉터리 이름으로 교체하여 **--dest-dir=NAME** 매개변수를 포함합니다.

```
oc adm must-gather --image=quay.io/stolostron/backplane-must-gather:SNAPSHOTNAME
/usr/bin/gather hosted-cluster-namespace=HOSTEDCLUSTERNAMESPACE hosted-cluster-
name=HOSTEDCLUSTERNAME --dest-dir=NAME ; tar -cvzf NAME.tgz NAME
```

1.2.4.4. 연결이 끊긴 환경에서 must-gather 명령 입력

연결이 끊긴 환경에서 **must-gather** 명령을 실행하려면 다음 단계를 완료합니다.

1. 연결이 끊긴 환경에서 **Red Hat Operator** 카탈로그 이미지를 미러 레지스트리에 미러링합니다. 자세한 내용은 연결이 끊긴 네트워크에 설치를 참조하십시오.

2. 다음 명령을 실행하여 미러 레지스트리에서 이미지를 참조하는 로그를 추출합니다.

```
REGISTRY=registry.example.com:5000
IMAGE=$REGISTRY/multicluster-engine/must-gather-
rhel8@sha256:ff9f37eb400dc1f7d07a9b6f2da9064992934b69847d17f59e385783c071b9d8

oc adm must-gather --image=$IMAGE /usr/bin/gather hosted-cluster-
namespace=HOSTEDCLUSTERNAMESPACE hosted-cluster-
name=HOSTEDCLUSTERNAME --dest-dir=./data
```

1.2.4.5. 추가 리소스

- 호스트된 컨트롤 플레인 문제 해결에 대한 자세한 내용은 **OpenShift Container Platform** 설명서의 호스트된 컨트롤 플레인 문제 해결을 참조하십시오.

1.3. 설치 또는 보류 중인 설치 상태 문제 해결

Red Hat Advanced Cluster Management를 설치할 때 **MultiClusterHub** 는 설치 단계에 남아 있거나 여러 **Pod**가 **Pending** 상태를 유지합니다.

1.3.1. 증상: 보류 중 상태 발생

MultiClusterHub를 설치한 후 **10** 분 이상 전달되었으며 **MultiClusterHub** 리소스의 **status.components** 필드에서 하나 이상의 구성 요소가 **ProgressDeadlineExceeded.** 클러스터의 리소스 제약 조건이 문제가 될 수 있습니다.

Multiclusterhub 가 설치된 네임스페이스에서 **Pod**를 확인합니다. 다음과 유사한 상태로 보류 중이 표시될 수 있습니다.

```
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.'
```

이 경우 제품을 실행하기 위해 작업자 노드 리소스가 클러스터에 충분하지 않습니다.

1.3.2. 문제 해결: 작업자 노드 크기 조정

이 문제가 있는 경우 더 큰 작업자 노드로 클러스터를 업데이트해야 합니다. 클러스터 크기 지정에 대한 지침은 클러스터 크기 조정을 참조하십시오.

1.4. 설치 실패 문제 해결

Kubernetes용 **Red Hat Advanced Cluster Management**를 다시 설치할 때 **Pod**가 시작되지 않습니다.

1.4.1. 증상: 재설치 실패

Red Hat Advanced Cluster Management를 설치한 후 **Pod**가 시작되지 않으면 이 설치를 시도하기 전에 **Red Hat Advanced Cluster Management**가 이전에 설치되었을 가능성이 있으며 일부 항목은 제거되지 않았습니다.

이 경우 설치 프로세스를 완료한 후 **Pod**가 시작되지 않습니다.

1.4.2. 문제 해결: 설치 실패

이 문제가 있는 경우 다음 단계를 완료합니다.

1. 설치 제거 프로세스를 실행하여 설치 제거의 단계에 따라 현재 구성 요소를 제거합니다. ../../html-single/install#uninstalling

2. Helm 설치 지침에 따라 Helm CLI 바이너리 버전 3.2.0 이상을 설치합니다. https://helm.sh/docs/intro/install/

3. Red Hat OpenShift Container Platform CLI가 oc 명령을 실행하도록 구성되어 있는지 확인합니다. oc 명령 구성에 대한 자세한 내용은 OpenShift Container Platform 설명서에서 OpenShift CLI 시작하기 를 참조하십시오.

4. 다음 스크립트를 파일에 복사합니다.

```
#!/bin/bash
ACM_NAMESPACE=<namespace>
oc delete mch --all -n $ACM_NAMESPACE
oc delete apiservice v1.admission.cluster.open-cluster-management.io
v1.admission.work.open-cluster-management.io
oc delete clusterimageset --all
oc delete clusterrole multiclusterengines.multicluster.openshift.io-v1-admin
multiclusterengines.multicluster.openshift.io-v1-crdview
multiclusterengines.multicluster.openshift.io-v1-edit
multiclusterengines.multicluster.openshift.io-v1-view open-cluster-
management:addons:application-manager open-cluster-management:admin-aggregate open-
cluster-management:cert-policy-controller-hub open-cluster-management:cluster-manager-
admin-aggregate open-cluster-management:config-policy-controller-hub open-cluster-
management:edit-aggregate open-cluster-management:iam-policy-controller-hub open-
cluster-management:policy-framework-hub open-cluster-management:view-aggregate
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
discoveredclusters.discovery.open-cluster-management.io discoveryconfigs.discovery.open-
cluster-management.io
oc delete mutatingwebhookconfiguration ocm-mutating-webhook
managedclustermutators.admission.cluster.open-cluster-management.io multicluster-
observability-operator
oc delete validatingwebhookconfiguration
channels.apps.open.cluster.management.webhook.validator application-webhook-validator
multiclusterhub-operator-validating-webhook ocm-validating-webhook multicluster-
observability-operator multiclusterengines.multicluster.openshift.io
```

스크립트에서 <namespace >를 Red Hat Advanced Cluster Management가 설치된 네임스페이스 이름으로 바꿉니다. 네임스페이스가 정리 및 삭제되므로 올바른 네임스페이스를 지정해야 합니다.

-

5. 스크립트를 실행하여 이전 설치에서 아티팩트를 제거합니다.

6. 설치를 실행합니다. 온라인으로 연결하는 동안 설치를 참조하십시오.

1.5. RED HAT ADVANCED CLUSTER MANAGEMENT 업그레이드 후 OCM-CONTROLLER 오류 문제 해결

2.7.x에서 2.8.x로 업그레이드한 후 multicluster-engine 네임스페이스의 ocm-controller 가 충돌합니다.

1.5.1. 증상: Red Hat Advanced Cluster Management 업그레이드 후 ocm-controller 오류 문제 해결

ManagedClusterSet 및 ManagedClusterSetBinding 사용자 정의 리소스 정의를 나열하려는 후 다음 오류 메시지가 표시됩니다.

```
Error from server: request to convert CR from an invalid group/version: cluster.open-cluster-management.io/v1beta1
```

이전 메시지는 ManagedClusterSets 및 ManagedClusterSetBindings 사용자 정의 리소스 정의 v1beta1 에서 v1beta2 로의 사용자 정의 리소스 정의가 실패했음을 나타냅니다.

1.5.2. 문제 해결: Red Hat Advanced Cluster Management 업그레이드 후 ocm-controller 오류 문제 해결

이 오류를 해결하려면 API 마이그레이션을 수동으로 시작해야 합니다. 다음 단계를 완료하십시오.

1. cluster-manager 를 이전 릴리스로 되돌립니다.

   a. 다음 명령을 사용하여 다중 클러스터 엔진 을 일시 중지합니다.

   ```
   oc annotate mce multiclusterengine pause=true
   ```

   b. 다음 명령을 실행하여 cluster-manager 배포의 이미지를 이전 버전으로 교체합니다.

   ```
   oc patch deployment cluster-manager -n multicluster-engine -p \  '{"spec":
   ```

```
{"template":{"spec":{"containers":[{"name":"registration-
operator","image":"registry.redhat.io/multicluster-engine/registration-operator-
rhel8@sha256:35999c3a1022d908b6fe30aa9b85878e666392dbbd685e9f3edcb83e33
36d19f"}]}}}}'
export ORIGIN_REGISTRATION_IMAGE=$(oc get clustermanager cluster-manager
-o jsonpath='{.spec.registrationImagePullSpec}')
```

c. **ClusterManager** 리소스의 등록 이미지 참조를 이전 버전으로 교체합니다. 다음 명령을 실행합니다.

```
oc patch clustermanager cluster-manager --type='json' -p='[{"op": "replace",
"path": "/spec/registrationImagePullSpec", "value":
"registry.redhat.io/multicluster-engine/registration-
rhel8@sha256:a3c22aa4326859d75986bf24322068f0aff2103cccc06e1001faaf79b939
0515"}]'
```

2. 다음 명령을 실행하여 **ManagedClusterSets** 및 **ManagedClusterSetBindings** 사용자 정의 리소스 정의를 이전 릴리스로 되돌립니다.

```
oc annotate crds managedclustersets.cluster.open-cluster-management.io
operator.open-cluster-management.io/version-
oc annotate crds  managedclustersetbindings.cluster.open-cluster-management.io
operator.open-cluster-management.io/version-
```

3. **cluster-manager** 를 다시 시작하고 사용자 정의 리소스 정의가 다시 생성될 때까지 기다립니다. 다음 명령을 실행합니다.

```
oc -n multicluster-engine delete pods -l app=cluster-manager
oc wait crds managedclustersets.cluster.open-cluster-management.io --
for=jsonpath="{.metadata.annotations['operator\.open-cluster-
management\.io/version']}"="2.3.3" --timeout=120s
oc wait crds managedclustersetbindings.cluster.open-cluster-management.io --
for=jsonpath="{.metadata.annotations['operator\.open-cluster-
management\.io/version']}"="2.3.3" --timeout=120s
```

4. 다음 명령을 사용하여 스토리지 버전 마이그레이션을 시작합니다.

```
oc patch StorageVersionMigration managedclustersets.cluster.open-cluster-
management.io --type='json' -p='[{"op":"replace", "path":"/spec/resource/version",
"value":"v1beta1"}]'
oc patch StorageVersionMigration managedclustersets.cluster.open-cluster-
management.io --type='json' --subresource status -p='[{"op":"remove",
"path":"/status/conditions"}]'
oc patch StorageVersionMigration managedclustersetbindings.cluster.open-cluster-
management.io --type='json' -p='[{"op":"replace", "path":"/spec/resource/version",
"value":"v1beta1"}]'
```

```
oc patch StorageVersionMigration managedclustersetbindings.cluster.open-cluster-management.io --type='json' --subresource status -p='[{"op":"remove", "path":"/status/conditions"}]'
```

5. 다음 명령을 실행하여 마이그레이션이 완료될 때까지 기다립니다.

```
oc wait storageversionmigration managedclustersets.cluster.open-cluster-management.io --for=condition=Succeeded --timeout=120s
oc wait storageversionmigration managedclustersetbindings.cluster.open-cluster-management.io --for=condition=Succeeded --timeout=120s
```

6. cluster-manager 를 Red Hat Advanced Cluster Management 2.10으로 복원합니다. 몇 분 정도 걸릴 수 있습니다. 다음 명령을 실행합니다.

```
oc annotate mce multiclusterengine pause-
oc patch clustermanager cluster-manager --type='json' -p='[{"op": "replace", "path": "/spec/registrationImagePullSpec", "value": "'$ORIGIN_REGISTRATION_IMAGE'"}]'
```

1.5.2.1. 검증

Red Hat Advanced Cluster Management가 복구되었는지 확인하려면 다음 명령을 실행합니다.

```
oc get managedclusterset
oc get managedclustersetbinding -A
```

명령을 실행하면 ManagedClusterSets 및 ManagedClusterSetBindings 리소스가 오류 메시지 없이 나열됩니다.

1.6. 오프라인 클러스터 문제 해결

오프라인 상태를 표시하는 클러스터에는 몇 가지 일반적인 원인이 있습니다.

1.6.1. 증상: 클러스터 상태가 오프라인 상태

클러스터 생성 절차를 완료한 후에는 Red Hat Advanced Cluster Management 콘솔에서 액세스할 수 없으며 오프라인 상태가 표시됩니다.

1.6.2. 문제 해결: 클러스터 상태가 오프라인 상태입니다.

1.

관리 클러스터를 사용할 수 있는지 확인합니다. **Red Hat Advanced Cluster Management** 콘솔의 ***Clusters*** 영역에서 확인할 수 있습니다.

사용할 수 없는 경우 관리 클러스터를 다시 시작하십시오.

2. 관리 클러스터 상태가 여전히 오프라인 상태인 경우 다음 단계를 완료합니다.

   a. **hub** 클러스터에서 **oc get managedcluster <cluster_name> -o yaml** 명령을 실행합니다. **& lt;cluster_name >**을 클러스터 이름으로 바꿉니다.

   b. **status.conditions** 섹션을 찾습니다.

   c. **ManagedClusterConditionAvailable** 유형의 메시지를 확인하고 모든 문제를 해결합니다.

1.7. 관리형 클러스터 가져오기 실패 문제 해결

클러스터 가져오기에 실패하면 클러스터 가져오기에 실패한 이유를 확인하기 위해 수행할 수 있는 몇 가지 단계가 있습니다.

1.7.1. 증상: 가져온 클러스터를 사용할 수 없음

클러스터 가져오기 절차를 완료한 후에는 **Kubernetes** 콘솔용 **Red Hat Advanced Cluster Management**에서 액세스할 수 없습니다.

1.7.2. 문제 해결: 가져온 클러스터를 사용할 수 없음

가져오기를 시도한 후에는 가져온 클러스터를 사용할 수 없는 몇 가지 이유가 있을 수 있습니다. 클러스터 가져오기에 실패하면 가져오기에 실패한 이유를 찾을 때까지 다음 단계를 완료합니다.

1. **Red Hat Advanced Cluster Management Hub** 클러스터에서 다음 명령을 실행하여 **Red Hat Advanced Cluster Management** 가져오기 컨트롤러가 실행 중인지 확인합니다.

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

실행 중인 **Pod** 두 개가 표시되어야 합니다. **Pod** 중 하나가 실행 중이 아닌 경우 다음 명령을

실행하여 로그를 확인하여 이유를 확인합니다.

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

2. **Red Hat Advanced Cluster Management Hub** 클러스터에서 다음 명령을 실행하여 **Red Hat Advanced Cluster Management** 가져오기 컨트롤러에서 관리 클러스터 가져오기 보안이 성공적으로 생성되었는지 확인합니다.

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

가져오기 보안이 없는 경우 다음 명령을 실행하여 가져오기 컨트롤러의 로그 항목을 보고 생성되지 않은 이유를 확인합니다.

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

3. **Red Hat Advanced Cluster Management Hub** 클러스터에서 관리 클러스터가 로컬 클러스터 이거나 **Hive**에서 프로비저닝하거나 자동 가져오기 시크릿이 있는 경우 다음 명령을 실행하여 관리 클러스터의 가져오기 상태를 확인합니다.

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

ManagedClusterImportSucceeded 조건이 **true** 가 아닌 경우 명령의 결과는 실패 이유를 나타냅니다.

4. 성능이 저하된 상태에 대해 관리 클러스터의 **Klusterlet** 상태를 확인합니다. **Klusterlet**의 성능이 저하된 이유를 찾으려면 성능이 저하된 **Klusterlet** 문제 해결을 참조하십시오.

1.8. 가져오기 보류 중 상태의 클러스터 문제 해결

클러스터 콘솔에서 *보류 중 가져오기* 가 계속되는 경우 절차에 따라 문제를 해결합니다.

1.8.1. 증상: 가져오기 상태가 보류 중인 클러스터

Red Hat Advanced Cluster Management 콘솔을 사용하여 클러스터를 가져온 후 클러스터에 ***Pending import*** 상태로 콘솔에 표시됩니다.

1.8.2. 문제 확인: 가져오기 보류 중인 클러스터

1. 관리 클러스터에서 다음 명령을 실행하여 문제가 있는 **Kubernetes Pod** 이름을 확인합니다.

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

2. 관리 클러스터에서 다음 명령을 실행하여 오류의 로그 항목을 찾습니다.

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

   registration_agent_pod 를 **1**단계에서 확인한 포드 이름으로 교체합니다.

3. 반환된 결과에 네트워킹 연결 문제가 있음을 나타내는 텍스트를 검색합니다. 예제에는 이러한 호스트가 없습니다.

1.8.3. 문제 해결: 가져오기 보류 중인 클러스터

1. **hub** 클러스터에 다음 명령을 입력하여 문제가 있는 포트 번호를 검색합니다.

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

2. 관리 클러스터의 호스트 이름을 확인할 수 있고 호스트와 포트에 대한 아웃바운드 연결이 발생하는지 확인합니다.

   관리 클러스터에서 통신을 설정할 수 없는 경우 클러스터 가져오기가 완료되지 않습니다. 관리 클러스터의 클러스터 상태는 *가져오기 보류* 중입니다.

1.9. 이미 존재하는 클러스터 문제 해결 오류

OpenShift Container Platform 클러스터를 **Red Hat Advanced Cluster Management MultiClusterHub** 로 가져올 수 없고 **AlreadyExists** 오류가 발생하는 경우 다음 절차에 따라 문제를 해결합니다.

1.9.1. 증상: OpenShift Container Platform 클러스터를 가져올 때 오류 로그가 준비됨

OpenShift Container Platform 클러스터를 **Red Hat Advanced Cluster Management MultiClusterHub** 로 가져올 때 오류 로그가 표시됩니다.

```
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. 문제 확인: OpenShift Container Platform 클러스터를 가져올 때 이미 존재합니다.

다음 명령을 실행하여 새 **Red Hat Advanced Cluster Management MultiClusterHub** 로 가져올 **Red Hat Advanced Cluster Management** 관련 리소스가 있는지 확인합니다.

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

1.9.3. 문제 해결: OpenShift Container Platform 클러스터를 가져올 때 이미 존재합니다.

다음 명령을 사용하여 **klusterlet** 사용자 정의 리소스를 제거합니다.

```
oc get klusterlet | grep klusterlet | awk '{print $1}' | xargs oc patch klusterlet --type=merge -p
'{"metadata":{"finalizers": []}}'
```

다음 명령을 실행하여 기존 리소스를 제거합니다.

```
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 VSPHERE에서 클러스터 생성 문제 해결

VMware vSphere에서 **Red Hat OpenShift Container Platform** 클러스터를 생성할 때 문제가 발생하는 경우 다음 문제 해결 정보를 참조하여 문제가 해결되었는지 확인하십시오.

참고: **VMware vSphere**에서 클러스터 생성 프로세스가 실패하면 로그를 볼 수 있는 링크가 활성화되어 있지 않을 수 있습니다. 이 경우 **hive-controllers Pod**의 로그를 확인하여 문제를 식별할 수 있습니다.

hive-controllers 로그는 **hive** 네임스페이스에 있습니다.

1.10.1. 인증서 IP SAN 오류와 함께 관리되는 클러스터 생성 실패

1.10.1.1. 증상: 관리형 클러스터 생성이 인증서 IP SAN 오류로 인해 실패합니다.

VMware vSphere에서 새 **Red Hat OpenShift Container Platform** 클러스터를 생성한 후 클러스터에 인증서 **IP SAN** 오류를 나타내는 오류 메시지와 함께 실패합니다.

1.10.1.2. 문제 식별: 관리형 클러스터 생성이 인증서 IP SAN 오류로 인해 실패합니다.

관리 클러스터의 배포가 실패하고 배포 로그에 다음 오류를 반환합니다.

```
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. 문제 해결: 인증서 IP SAN 오류로 관리되는 클러스터 생성에 실패합니다.

자격 증명의 **IP** 주소 대신 **VMware vCenter** 서버 정규화된 호스트 이름을 사용합니다. **VMware vCenter CA** 인증서를 업데이트하여 **IP SAN**을 포함할 수도 있습니다.

1.10.2. 알 수 없는 인증 기관으로 관리 클러스터 생성 실패

1.10.2.1. 증상: 관리형 클러스터 생성이 알 수 없는 인증 기관으로 인해 실패함

VMware vSphere에서 새 **Red Hat OpenShift Container Platform** 클러스터를 생성하면 인증서가 알 수 없는 기관에서 서명했기 때문에 클러스터가 실패합니다.

1.10.2.2. 문제 식별: Managed cluster creation fails with unknown certificate authority

관리 클러스터의 배포가 실패하고 배포 로그에 다음 오류를 반환합니다.

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

1.10.2.3. 문제 해결: 알 수 없는 인증 기관을 사용하면 관리형 클러스터 생성이 실패합니다.

인증 정보를 생성할 때 인증 기관에서 올바른 인증서를 입력했는지 확인합니다.

1.10.3. 만료된 인증서로 관리되는 클러스터 생성 실패

1.10.3.1. 증상: 만료된 인증서로 관리 클러스터 생성에 실패합니다.

VMware vSphere에서 새 **Red Hat OpenShift Container Platform** 클러스터를 생성한 후 인증서가 만료되거나 유효하지 않기 때문에 클러스터가 실패합니다.

1.10.3.2. 문제 식별: 만료된 인증서로 관리 클러스터 생성이 실패합니다.

관리 클러스터의 배포가 실패하고 배포 로그에 다음 오류를 반환합니다.

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

1.10.3.3. 문제 해결: 만료된 인증서로 관리 클러스터 생성이 실패합니다.

ESXi 호스트의 시간이 동기화되었는지 확인합니다.

1.10.4. 관리 클러스터 생성에 실패하여 태그 지정 권한이 충분하지 않음

1.10.4.1. 증상: 관리형 클러스터 생성이 실패하고 태그 지정 권한이 충분하지 않음

VMware vSphere에서 새 **Red Hat OpenShift Container Platform** 클러스터를 생성한 후 태그 지정을 사용할 수 있는 권한이 부족하기 때문에 클러스터가 실패합니다.

1.10.4.2. 문제 식별: 관리형 클러스터 생성이 실패하여 태그 지정 권한이 부족하지 않습니다.

관리 클러스터의 배포가 실패하고 배포 로그에 다음 오류를 반환합니다.

```
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. 문제 해결: 태그 지정에 대한 권한이 부족하여 관리되는 클러스터 생성이 실패합니다.

VMware vCenter 필수 계정 권한이 올바른지 확인합니다. 자세한 내용은 이미지 레지스트리 를 참조하십시오.

1.10.5. 관리 클러스터 생성이 유효하지 않은 dnsVIP와 함께 실패합니다.

1.10.5.1. 증상: 관리형 클러스터 생성이 유효하지 않은 dnsVIP와 함께 실패합니다.

VMware vSphere에서 새 **Red Hat OpenShift Container Platform** 클러스터를 생성한 후 잘못된 **dnsVIP**가 있으므로 클러스터가 실패합니다.

1.10.5.2. 문제 식별: Managed 클러스터 생성이 유효하지 않은 dnsVIP와 함께 실패합니다.

VMware vSphere를 사용하여 새 관리 클러스터를 배포하려고 할 때 다음 메시지가 표시되면 **VMware** 설치 관리자 프로비저닝 인프라(**IPI**)를 지원하지 않는 이전 **OpenShift Container Platform** 릴리스 이미지가 있기 때문입니다.

```
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. 문제 해결: 잘못된 dnsVIP와 함께 관리 클러스터 생성이 실패합니다.

VMware 설치 관리자 프로비저닝 인프라를 지원하는 이후 버전의 **OpenShift Container Platform**에서 릴리스 이미지를 선택합니다.

1.10.6. 관리형 클러스터 생성이 잘못된 네트워크 유형과 함께 실패합니다.

1.10.6.1. 증상: 관리형 클러스터 생성이 잘못된 네트워크 유형과 함께 실패합니다.

VMware vSphere에서 새 **Red Hat OpenShift Container Platform** 클러스터를 생성한 후 잘못된 네트워크 유형이 지정되어 있으므로 클러스터가 실패합니다.

1.10.6.2. 문제 식별: Managed cluster creation fails with incorrect network type

VMware vSphere를 사용하여 새 관리 클러스터를 배포하려고 할 때 다음 메시지가 표시되면 **VMware Installer Provisioned Infrastructure (IPI)**를 지원하지 않는 이전 **OpenShift Container Platform** 이미지가 있기 때문입니다.

```
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. 문제 해결: 관리형 클러스터 생성이 잘못된 네트워크 유형으로 인해 실패합니다.

지정된 **VMware** 클러스터에 유효한 **VMware vSphere** 네트워크 유형을 선택합니다.

1.10.7. 디스크 처리 디스크 변경 오류와 함께 관리되는 클러스터 생성 실패

1.10.7.1. 증상: 오류 처리 디스크 변경으로 인해 VMware vSphere 관리 클러스터 추가 실패

VMware vSphere에서 새 **Red Hat OpenShift Container Platform** 클러스터를 생성한 후 디스크 변경 사항을 처리할 때 오류가 있기 때문에 클러스터가 실패합니다.

1.10.7.2. 문제 식별: 오류 처리 디스크 변경으로 인해 VMware vSphere 관리 클러스터를 추가할 수 없습니다

다음과 유사한 메시지가 로그에 표시됩니다.

```
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. 문제 해결: 오류 처리 디스크 변경으로 인해 VMware vSphere 관리 클러스터를 추가할 수 없습니다

VMware vSphere 클라이언트를 사용하여 *프로파일 중심 스토리지 권한에 대한 모든 권한을 사용자에게 부여합니다*.

1.11. 알 수 없는 권한 오류가 있는 RED HAT OPENSTACK PLATFORM에서 관리형 클러스터 생성 실패

Red Hat OpenStack Platform에서 **Red Hat OpenShift Container Platform** 클러스터를 생성할 때 문제가 발생하는 경우 다음 문제 해결 정보를 참조하여 문제가 해결되었는지 확인하십시오.

1.11.1. 증상: 관리형 클러스터 생성에 알 수 없는 권한 오류로 인해 실패합니다.

자체 서명된 인증서를 사용하여 **Red Hat OpenStack Platform**에서 새 **Red Hat OpenShift Container Platform** 클러스터를 생성하면 알 수 없는 권한 오류를 나타내는 오류 메시지와 함께 클러스터가 실패합니다.

1.11.2. 문제 식별: 관리 클러스터 생성이 알 수 없는 권한 오류로 인해 실패합니다.

관리 클러스터의 배포가 실패하고 다음 오류 메시지를 반환합니다.

x509: 알 수 없는 기관에서 서명한 인증서

1.11.3. 문제 해결: 관리 클러스터 생성이 알 수 없는 권한 오류로 인해 실패합니다.

다음 파일이 올바르게 구성되었는지 확인합니다.

1. **clouds.yaml** 파일은 **cacert** 매개변수의 **ca.crt** 파일의 경로를 지정해야 합니다. **cacert** 매개변수는 **ignition shim**을 생성할 때 **OpenShift** 설치 프로그램에 전달됩니다. 다음 예제를 참조하십시오.

```
clouds:
  openstack:
    cacert: "/etc/pki/ca-trust/source/anchors/ca.crt"
```

2. **certificatesSecretRef** 매개변수는 **ca.crt** 파일과 일치하는 파일 이름이 있는 시크릿을 참조해야 합니다. 다음 예제를 참조하십시오.

```
spec:
  baseDomain: dev09.red-chesterfield.com
  clusterName: txue-osspoke
  platform:
    openstack:
      cloud: openstack
      credentialsSecretRef:
        name: txue-osspoke-openstack-creds
      certificatesSecretRef:
        name: txue-osspoke-openstack-certificatebundle
```

일치하는 파일 이름으로 보안을 생성하려면 다음 명령을 실행합니다.

```
oc create secret generic txue-osspoke-openstack-certificatebundle --from-
file=ca.crt=ca.crt.pem -n $CLUSTERNAME
```

3. **ca.cert** 파일의 크기는 **631,000**바이트 미만이어야 합니다.

1.12. OPENSHIFT CONTAINER PLATFORM 버전 3.11 클러스터 가져오기 실패 문제 해결

1.12.1. 증상: OpenShift Container Platform 버전 3.11 클러스터 가져오기 실패

Red Hat OpenShift Container Platform 버전 **3.11** 클러스터를 가져오려고 하면 다음 콘텐츠와 유사한 로그 메시지와 함께 가져오기가 실패합니다.

```
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|...
```

1.12.2. 문제 식별: OpenShift Container Platform 버전 3.11 클러스터 가져오기 실패

이는 설치된 **kubectl** 명령줄 툴이 **1.11** 이하이기 때문에 발생하는 경우가 많습니다. 다음 명령을 실행하여 실행 중인 **kubectl** 명령줄 툴 버전을 확인합니다.

```
kubectl version
```

반환된 데이터에 버전 **1.11** 또는 이전 버전이 나열된 경우 *문제 해결 방법 중 하나를 완료하십시오: **OpenShift Container Platform** 버전 **3.11** 클러스터 가져오기 실패*.

1.12.3. 문제 해결: OpenShift Container Platform 버전 3.11 클러스터 가져오기 실패

다음 절차 중 하나를 완료하여 이 문제를 해결할 수 있습니다.

- **kubectl** 명령줄 툴의 최신 버전을 설치합니다.
  1. Kubernetes 문서에서 Install and Set Up kubectl 툴의 최신 버전을 다운로드합니다.
  2. **kubectl** 툴을 업그레이드한 후 클러스터를 다시 가져옵니다.
- **import** 명령이 포함된 파일을 실행합니다.
  1. CLI를 사용하여 관리 클러스터 가져오기 절차를 시작합니다.
  2. 클러스터를 가져오는 명령을 생성할 때 해당 명령을 **import.yaml** 이라는 **YAML** 파일에 복사합니다.
  3. 다음 명령을 실행하여 파일에서 클러스터를 다시 가져옵니다.

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

1.13. 인증서 변경 후 가져온 클러스터 오프라인 문제 해결

사용자 정의 **apiserver** 인증서 설치가 지원되지만 인증서 정보를 변경하기 전에 가져온 하나 이상의 클러스터는 오프라인 상태입니다.

1.13.1. 증상: 인증서 변경 후 오프라인 클러스터

인증서 보안 업데이트 절차를 완료한 후 온라인 클러스터 중 하나 이상에 이제 콘솔에 오프라인 상태가 표시됩니다.

1.13.2. 문제 식별: 인증서 변경 후 오프라인 클러스터

사용자 정의 **API** 서버 인증서에 대한 정보를 업데이트한 후 새 인증서가 이제 오프라인 상태가 되기 전에 가져온 클러스터입니다.

인증서가 문제가 있음을 나타내는 오류는 오프라인 관리 클러스터의 **open-cluster-management-agent** 네임스페이스에 있는 **Pod** 로그에서 확인할 수 있습니다. 다음 예제는 로그에 표시되는 오류와 유사합니다.

다음 **work-agent** 로그를 참조하십시오.

```
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
```

다음 **registration-agent** 로그를 참조하십시오.

```
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
```

1.13.3. 문제 해결: 인증서 변경 후 오프라인 클러스터

관리 클러스터가 로컬 클러스터 이거나 **Red Hat Advanced Cluster Management for Kubernetes**를

사용하여 관리되는 클러스터를 생성한 경우 관리 클러스터를 다시 가져오는 데 **10**분 이상 기다려야 합니다.

관리 클러스터를 즉시 다시 가져오려면 **hub** 클러스터에서 관리 클러스터 가져오기 보안을 삭제하고 **Red Hat Advanced Cluster Management**를 사용하여 다시 가져올 수 있습니다. 다음 명령을 실행합니다.

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

& lt;cluster_name >을 가져올 관리 클러스터의 이름으로 바꿉니다.

Red Hat Advanced Cluster Management를 사용하여 가져온 관리형 클러스터를 다시 가져오려면 다음 단계를 완료하여 관리 클러스터를 다시 가져옵니다.

1. **hub** 클러스터에서 다음 명령을 실행하여 관리 클러스터 가져오기 보안을 다시 생성합니다.

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

   & lt;cluster_name >을 가져올 관리 클러스터의 이름으로 바꿉니다.

2. 허브 클러스터에서 다음 명령을 실행하여 관리 클러스터 가져오기 보안을 **YAML** 파일에 노출합니다.

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

   & lt;cluster_name >을 가져올 관리 클러스터의 이름으로 바꿉니다.

3. 관리 클러스터에서 다음 명령을 실행하여 **import.yaml** 파일을 적용합니다.

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

참고**:** 이전 단계에서는 **hub** 클러스터에서 관리 클러스터를 분리하지 않습니다. 이 단계에서는 새 인증서 정보를 포함하여 관리 클러스터의 현재 설정으로 필요한 매니페스트를 업데이트합니다.

1.14. 클러스터를 삭제한 후에도 네임스페이스가 남아 있음

관리 클러스터를 제거하면 일반적으로 네임스페이스가 클러스터 제거 프로세스의 일부로 제거됩니다. 드문 경우지만 네임스페이스에 일부 아티팩트가 남아 있습니다. 이 경우 네임스페이스를 수동으로 제거해야 합니다.

1.14.1. 증상: 클러스터를 삭제한 후에도 네임스페이스가 유지됩니다.

관리 클러스터를 제거한 후 네임스페이스는 제거되지 않습니다.

1.14.2. 문제 해결: 클러스터를 삭제한 후에도 네임스페이스가 남아 있습니다.

네임스페이스를 수동으로 제거하려면 다음 단계를 완료합니다.

1. 다음 명령을 실행하여 **<cluster_name>** 네임스페이스에 남아 있는 리소스 목록을 생성합니다.

```
oc api-resources --verbs=list --namespaced -o name | grep -E
'^secrets|^serviceaccounts|^managedclusteraddons|^roles|^rolebindings|^manifestworks|^leases
|^managedclusterinfo|^appliedmanifestworks'|^clusteroauths' | xargs -n 1 oc get --show-kind -
-ignore-not-found -n <cluster_name>
```

cluster_name 을 제거하려는 클러스터의 네임스페이스 이름으로 교체합니다.

2. 다음 명령을 입력하여 목록을 편집하여 **Delete** 상태가 없는 목록에서 식별된 각 리소스를 삭제합니다.

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

resource_kind 를 리소스 종류로 바꿉니다. **resource_name** 을 리소스 이름으로 교체합니다. **namespace** 를 리소스의 네임스페이스 이름으로 교체합니다.

3. 메타데이터에서 종료자 속성을 찾습니다.

4. **vi** 편집기 **dd** 명령을 사용하여 **Kubernetes**가 아닌 종료자를 삭제합니다.

5. 목록을 저장하고 **:wq** 명령을 입력하여 **vi** 편집기를 종료합니다.

6. 다음 명령을 입력하여 네임스페이스를 삭제합니다.

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

cluster-name 을 삭제하려는 네임스페이스 이름으로 교체합니다.

1.15. 클러스터를 가져올 때 AUTO-IMPORT-SECRET-EXISTS 오류

자동 가져오기 보안이라는 오류 메시지와 함께 클러스터 가져오기가 실패합니다.

1.15.1. 증상: 클러스터를 가져올 때 자동 가져오기 보안 오류가 발생했습니다

관리를 위해 하이브 클러스터를 가져올 때 자동 가져오기 보안 오류가 이미 표시됩니다.

1.15.2. 문제 해결: 클러스터를 가져올 때 Auto-import-secret-exists 오류

이 문제는 **Red Hat Advanced Cluster Management**에서 이전에 관리하는 클러스터를 가져오려고 할 때 발생합니다. 이 경우 클러스터를 다시 가져오려고 할 때 보안이 충돌합니다.

이 문제를 해결하려면 다음 단계를 완료하십시오.

1. 기존 **auto-import-secret** 을 수동으로 삭제하려면 **hub** 클러스터에서 다음 명령을 실행합니다.

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

cluster-namespace 를 클러스터의 네임스페이스로 바꿉니다.

2. 클러스터 가져오기 소개의 절차를 사용하여 클러스터를 다시 가져옵니다.

1.16. VOLSYNC용 CSI(CONTAINER STORAGE INTERFACE) 드라이버 문제 해결

reflectSync를 사용하거나 CSI(cinder Container Storage Interface) 드라이버에서 기본 설정을 사용하는 경우 사용 중인 PVC에 오류가 발생할 수 있습니다.

1.16.1. 증상: Volumesnapshot 오류 상태

스냅샷을 사용하도록 EgressSync ReplicationSource 또는 ReplicationDestination 을 구성할 수 있습니다. 또한 ReplicationSource 및 ReplicationDestination 에서 storageclass 및 volumesnapshotclass 를 구성할 수 있습니다. 값이 false 인 force-create 라는 cinder volumesnapshotclass 에 매개 변수가 있습니다. volumesnapshotclass 의 이 force-create 매개변수는 cinder가 volumesnapshot 을 사용 중인 PVC를 사용하도록 허용하지 않음을 의미합니다. 결과적으로 volumesnapshot 이 오류 상태입니다.

1.16.2. 문제 해결: 매개변수를 true로 설정

1. cinder CSI 드라이버의 새 volumesnapshotclass 를 만듭니다.

2. paramater, force-create 를 true 로 변경합니다. 다음 샘플 YAML을 참조하십시오.

```
apiVersion: snapshot.storage.k8s.io/v1
deletionPolicy: Delete
driver: cinder.csi.openstack.org
kind: VolumeSnapshotClass
metadata:
  annotations:
    snapshot.storage.kubernetes.io/is-default-class: 'true'
  name: standard-csi
parameters:
  force-create: 'true'
```

1.17. MUST-GATHER 명령을 실행하여 문제 해결

must-gather 명령을 실행하여 세부 정보, 로그 및 디버깅 문제 단계를 수집합니다. 이 디버깅 정보는 지원 요청을 열 때 유용합니다. oc adm must-gather CLI 명령은 다음을 포함하여 문제를 디버깅하는 데 종종 필요한 정보를 클러스터에서 수집합니다.

- 리소스 정의
- 서비스 로그

1.17.1. 사전 요구 사항

must-gather 명령을 실행하려면 다음 사전 요구 사항을 충족해야 합니다.

- **cluster-admin** 역할의 사용자로 글로벌 허브 및 관리 허브 클러스터에 액세스할 수 있습니다.
- **OpenShift Container Platform CLI(oc)**가 설치되어 있어야 합니다.

1.17.2. must-gather 명령 실행

must-gather 명령을 사용하여 정보를 수집하려면 다음 절차를 완료합니다.

1. **must-gather** 명령에 대해 알아보고 **OpenShift Container Platform** 설명서에서 클러스터에 대한 데이터 수집 데이터를 읽고 필요한 사전 요구 사항을 설치합니다.
2. 글로벌 허브 클러스터에 로그인합니다. 일반적인 사용 사례는 글로벌 허브 클러스터에 로그인하는 동안 다음 명령을 실행합니다.

```
oc adm must-gather --image=quay.io/stolostron/must-gather:SNAPSHOTNAME
```

관리 허브 클러스터를 확인하려면 해당 클러스터에서 **must-gather** 명령을 실행합니다.

3. 선택 사항: **SOMENAME** 디렉터리에 결과를 저장하려면 이전 단계의 명령 대신 다음 명령을 실행할 수 있습니다.

```
oc adm must-gather --image=quay.io/stolostron/must-gather:SNAPSHOTNAME --dest-dir=
<SOMENAME> ; tar -cvzf <SOMENAME>.tgz <SOMENAME>
```

디렉터리의 다른 이름을 지정할 수 있습니다.

참고: 이 명령에는 **gzipped tarball** 파일을 만드는 데 필요한 추가 기능이 포함되어 있습니다.

must-gather 명령에서 다음 정보가 수집됩니다.

- 두 개의 피어 수준: **cluster-scoped-resources** 및 **namespace resources.**

- 각 하위 수준: 클러스터 범위 및 네임스페이스 범위 리소스 모두에 대한 사용자 정의 리소스 정의에 대한 **API** 그룹입니다.

- 각 **YAML** 파일의 다음 수준: 종류별로 정렬됩니다.

- 글로벌 허브 클러스터의 경우 네임스페이스 리소스에서 **PostgresCluster** 및 **Kafka** 를 확인할 수 있습니다.

- 글로벌 허브 클러스터의 경우 다중 클러스터 글로벌 허브 관련 **Pod**를 확인하고 네임스페이스 리소스의 **Pod** 에서 로그를 확인할 수 있습니다.

- 관리 허브 클러스터의 경우 다중 클러스터 글로벌 허브 에이전트 **Pod**를 확인하고 네임스페이스 리소스의 **Pod** 에 로그인할 수 있습니다.

1.18. 문제 해결을 위해 프로비저닝된 POSTGRESQL 데이터베이스에 액세스

프로비저닝된 **PostgreSQL** 데이터베이스에 액세스하여 **multicluster** 글로벌 허브의 문제를 해결하는 데 도움이 될 수 있는 메시지를 볼 수 있습니다. 서비스 유형에 따라 프로비저닝된 **PostgreSQL** 데이터베이스에 액세스하는 세 가지 방법이 있습니다.

- **ClusterIP** 서비스 사용

  1. 다음 명령을 실행하여 **postgres** 연결 **URI**를 확인합니다.

     ```
     oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "uri" | base64decode}}'
     ```

  2. 다음 명령을 실행하여 데이터베이스에 액세스합니다.

     ```
     oc exec -it $(kubectl get pods -n multicluster-global-hub-postgres -l postgres-operator.crunchydata.com/role=master -o jsonpath='{.items..metadata.name}') -c database -n multicluster-global-hub-postgres -- psql -U postgres -d hoh -c "SELECT 1"
     ```

- **NodePort 서비스 사용**

1. 다음 명령을 실행하여 서비스를 **NodePort**로 수정하고, 호스트를 노드 **IP**로 설정하고, 포트를 **32432**로 설정합니다.

```
oc patch postgrescluster hoh -n multicluster-global-hub-postgres -p '{"spec":{"service":{"type":"NodePort", "nodePort": 32432}}}'  --type merge
```

2. 다음 명령을 실행하여 사용자 이름을 추가합니다.

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "user" | base64decode}}'
```

3. 다음 명령을 실행하여 암호를 추가합니다.

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "password" | base64decode}}'
```

4. 다음 명령을 실행하여 데이터베이스 이름을 추가합니다.

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "dbname" | base64decode}}'
```

- **LoadBalancer**

1. 다음 명령을 실행하여 서비스 유형을 **LoadBalancer** 로 설정합니다.

```
oc patch postgrescluster hoh -n multicluster-global-hub-postgres -p '{"spec":{"service":{"type":"LoadBalancer"}}}'  --type merge
```

기본 포트는 **5432**입니다.

2. 다음 명령을 실행하여 호스트 이름을 설정합니다.

```
kubectl get svc -n multicluster-global-hub-postgres hoh-ha -ojsonpath='{.status.loadBalancer.ingress[0].hostname}'
```

3. 다음 명령을 실행하여 사용자 이름을 추가합니다.

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "user" | base64decode}}'
```

4. 다음 명령을 실행하여 암호를 추가합니다.

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "password" | base64decode}}'
```

5. 다음 명령을 실행하여 데이터베이스 이름을 추가합니다.

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "dbname" | base64decode}}'
```

1.19. 문제 해결을 위해 데이터베이스 덤프 및 복원 사용

프로덕션 환경에서는 **PostgreSQL** 데이터베이스를 정기적으로 데이터베이스 관리 작업으로 백업합니다. 백업을 사용하여 다중 클러스터 글로벌 허브를 디버깅할 수도 있습니다.

1.19.1. 중복을 위해 데이터베이스의 출력 덤프

multicluster 글로벌 허브 데이터베이스의 출력을 덤프하여 문제를 디버깅하는 경우가 있습니다. **PostgreSQL** 데이터베이스는 데이터베이스 콘텐츠를 덤프하는 **pg_dump** 명령줄 툴을 제공합니다. **localhost** 데이터베이스 서버에서 데이터를 덤프하려면 다음 명령을 실행합니다.

```
pg_dump hoh > hoh.sql
```

압축 형식이 있는 원격 서버에 있는 다중 클러스터 글로벌 허브 데이터베이스를 덤프하려면 다음 예와 같이 명령줄 옵션을 사용하여 연결 세부 정보를 제어합니다.

```
pg_dump -h my.host.com -p 5432 -U postgres -F t hoh -f hoh-$(date +%d-%m-%y_%H-%M).tar
```

1.19.2. 덤프에서 데이터베이스 복원

PostgreSQL 데이터베이스를 복원하려면 **psql** 또는 **pg_restore** 명령줄 툴을 사용할 수 있습니다. **psql** 툴은 **pg_dump** 로 생성된 일반 텍스트 파일을 복원하는 데 사용됩니다.

```
psql -h another.host.com -p 5432 -U postgres -d hoh < hoh.sql
```

pg_restore 툴은 **pg_dump** 가 아닌 텍스트 형식**(custom, tar** 또는 **directory)** 중 하나로 생성된 아카이브에서 **PostgreSQL** 데이터베이스를 복원하는 데 사용됩니다.

```
pg_restore -h another.host.com -p 5432 -U postgres -d hoh hoh-$(date +%d-%m-%y_%H-%M).tar
```

1.20. 규정 준수 데이터 복원

Grafana Datasource는 주로 **history.local_compliance** 라는 테이블에서 사용됩니다. 해당 레코드는 **Nightly 00:00:00**부터 시작하는 요약 루틴에 의해 생성됩니다. 일반적으로 요약 프로세스를 수동으로 실행할 필요가 없습니다. 경우에 따라 규정 준수 작업을 실행할 때 예기치 않은 오류가 발생할 수 있으므로 생성되지 않은 데이터를 복구하기 위해 전체 요약 프로세스를 실행하려면 데이터베이스에 수동으로 로그인해야 합니다. **Running the summarization** 프로세스를 수동으로 실행하여 데이터를 복구할 수 있습니다.

1.20.1. 선택 사항: 기존 테이블을 파티션 테이블로 수동으로 업그레이드

GA 전에 멀티 클러스터 글로벌 허브의 초기 버전을 설치한 경우 현재 다중 클러스터 글로벌 허브 **Operator**와 호환되도록 테이블을 업그레이드해야 합니다. 업그레이드의 주요 목적은 **event.local_policies,event.local_root_policies, history.local_compliance** 테이블을 분할된 테이블로 변환하는 것입니다.

다음 예제에서는 **2023-08** 에 대한 날짜가 설정된 **event.local_policies** 테이블의 변환을 보여줍니다. 다른 두 테이블의 업그레이드 단계는 비슷합니다.

1. 대상이 분할되었는지 확인합니다.

```
SELECT relname, relkind FROM pg_class WHERE relname = 'local_policies';
```

표 출력은 다음 예와 유사합니다.

relname	relkind	local_policies	r

relkind 가 **p** 이면 현재 테이블이 분할됩니다. 이 경우 나머지 단계를 건너뛰고 다른 테이블을 업그레이드할 수 있습니다.

2. 일반 테이블을 분할된 테이블로 변환합니다.

```
-- start a transaction
BEGIN;
-- Rename the legacy TABLE_NAME
ALTER TABLE event.local_policies RENAME TO local_policies_old;
-- Partition tables: https://github.com/stolostron/multicluster-global-hub/blob/main/operator/pkg/controllers/hubofhubs/database/2.tables.sql#L283-L318
CREATE TABLE IF NOT EXISTS event.local_policies (
   event_name character varying(63) NOT NULL,
   policy_id uuid NOT NULL,
   cluster_id uuid NOT NULL,
   leaf_hub_name character varying(63) NOT NULL,
   message text,
   reason text,
   count integer NOT NULL DEFAULT 0,
   source jsonb,
   created_at timestamp without time zone DEFAULT now() NOT NULL,
   compliance local_status.compliance_type NOT NULL,
   -- Rename the constraint to avoid conflicts
   CONSTRAINT local_policies_unique_partition_constraint UNIQUE (event_name, count, created_at)
) PARTITION BY RANGE (created_at);
-- Create partitions, load the old data to the previous partition table
CREATE TABLE IF NOT EXISTS event.local_policies_2023_08 PARTITION OF event.local_policies FOR VALUES FROM ('2023-08-01') TO ('2023-09-01');
CREATE TABLE IF NOT EXISTS event.local_policies_2023_07 PARTITION OF event.local_policies FOR VALUES FROM ('2000-01-01') TO ('2023-08-01');

-- Move the records from regular table to partition table
INSERT INTO event.local_policies SELECT * FROM event.local_polcies_old;
DROP TABLE IF EXISTS event.local_policies_old;
-- commit the transaction
COMMIT;
```

테이블 이름 및 현재 날짜에 따라 다음 값을 교체할 수 있습니다.

- event.local_policies_2023_08 은 8월 예를 사용하여 현재 달의 접미사가 있는 파티션 이름입니다.
- '2023-08-01' 및 '2023-09-01' 은 현재 달 파티션의 최소 및 최대 경계입니다.
- event.local_policies_2023_07 은 이전 월의 접미사가 있는 파티션 이름입니다(July)

- '2000-01-01' 및 '2023-08-01' 은 이전 달 파티션의 최소 및 최대 경계입니다.

1.21. 클러스터 상태가 오프라인에서 사용 가능으로 변경 문제 해결

관리 클러스터의 상태는 환경 또는 클러스터를 수동으로 변경하지 않고 오프라인에서 사용 가능한 상태로 변경됩니다.

1.21.1. 증상: 클러스터 상태가 오프라인에서 사용 가능으로 변경

관리 클러스터를 허브 클러스터에 연결하는 네트워크가 불안정한 경우 허브 클러스터 사이클에 의해 오프라인 과 사용 가능 으로 보고되는 관리 클러스터의 상태가 불안정합니다.

허브 클러스터와 관리형 클러스터 간의 연결은 **leaseDurationSeconds** 간격 값에 유효한 리스를 통해 유지 관리됩니다. **leaseDurationSeconds** 값의 5번 연속 시도 내에서 리스를 검증하지 않으면 클러스터가 오프라인으로 표시됩니다.

예를 들어, 리스**DurationSeconds** 간격이 **60**초 인 **5**분 후에 클러스터가 오프라인 상태로 표시됩니다. 이 구성은 연결 문제 또는 대기 시간과 같은 이유로 부적절하여 불안정성을 유발할 수 있습니다.

1.21.2. 문제 해결: 클러스터 상태가 오프라인에서 사용 가능으로 변경

5개의 검증 시도는 기본값이며 변경할 수 없지만 **leaseDurationSeconds** 간격을 변경할 수 있습니다.

클러스터를 오프라인으로 표시할 시간(분)을 확인한 다음 해당 값을 **60**으로 곱하여 초로 변환합니다. 그런 다음 기본 **5**개의 시도로 나눕니다. 그 결과 리스**DurationSeconds** 값이 됩니다.

1. 다음 명령을 입력하여 **hub** 클러스터에서 **ManagedCluster** 사양을 편집하지만 **cluster-name** 을 관리 클러스터 이름으로 교체합니다.

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

2. 다음 샘플 **YAML**에 표시된 대로 **ManagedCluster** 사양에서 **leaseDurationSeconds** 값을 늘립니다.

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
```

```
metadata:
  name: <cluster-name>
spec:
  hubAcceptsClient: true
  leaseDurationSeconds: 60
```

3. 파일을 저장하고 적용합니다.

1.22. 보류 중 또는 실패 상태의 콘솔에서 클러스터 문제 해결

생성한 클러스터의 콘솔에서 ***Pending*** 상태 또는 *실패* 상태를 모니터링하는 경우 절차를 수행하여 문제를 해결합니다.

1.22.1. 증상: 보류 중이거나 실패한 콘솔의 클러스터

Red Hat Advanced Cluster Management for Kubernetes 콘솔을 사용하여 새 클러스터를 생성한 후 클러스터는 ***Pending*** 상태를 초과하지 않거나 ***Failed*** 상태를 표시합니다.

1.22.2. 문제 식별: 보류 중이거나 실패한 콘솔의 클러스터

클러스터에 ***Failed*** 상태가 표시되면 클러스터의 세부 정보 페이지로 이동하여 제공된 로그 링크를 따릅니다. 로그를 찾을 수 없거나 클러스터에 ***Pending*** 상태가 표시되면 다음 절차를 계속 실행하여 로그를 확인합니다.

- 절차 1

  1. 허브 클러스터에서 다음 명령을 실행하여 새 클러스터의 네임스페이스에 생성된 **Kubernetes Pod**의 이름을 확인합니다.

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

     new_cluster_name을 생성한 클러스터 이름으로 교체합니다.

  2. 이름에 **provision** 문자열이 포함된 **Pod**가 나열되지 않은 경우 **Procedure 2**를 계속합니다. 제목에 프로비저닝 이 있는 **Pod**가 있는 경우 허브 클러스터에서 다음 명령을 실행하여 해당 **Pod**의 로그를 확인합니다.

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

new_cluster_name_provision_pod_name 을 생성한 클러스터 이름 및 프로비저닝이 포함된 포드 이름으로 교체 합니다.

3. 문제의 원인을 설명할 수 있는 로그에서 오류를 검색합니다.

- 절차 **2**

이름이 **provision** 인 **Pod**가 없는 경우 프로세스 초기에 문제가 발생했습니다. 로그를 보려면 다음 절차를 완료합니다.

1. **hub** 클러스터에서 다음 명령을 실행합니다.

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

new_cluster_name 을 생성한 클러스터 이름으로 교체합니다. 클러스터 설치 로그에 대한 자세한 내용은 **Red Hat OpenShift** 설명서의 설치 로그 수집을 참조하십시오.

2. 리소스의 ***Status.Conditions.Message*** 및 ***Status.Conditions.Reason*** 항목에서 문제에 대한 추가 정보가 있는지 확인하십시오.

1.22.3. 문제 해결: 보류 중이거나 실패한 콘솔의 클러스터

로그에서 오류를 확인한 후 클러스터를 제거하고 다시 생성하기 전에 오류를 해결하는 방법을 확인합니다.

다음 예제에서는 지원되지 않는 영역을 선택할 때 발생할 수 있는 로그 오류와 이를 해결하는 데 필요한 작업을 제공합니다.

```
No subnets provided for zones
```

클러스터를 생성할 때 지원되지 않는 리전 내에서 하나 이상의 영역을 선택했습니다. 클러스터를 재생성하여 문제를 해결할 때 다음 작업 중 하나를 완료합니다.

- 지역 내에서 다른 영역을 선택합니다.

- 다른 영역이 나열된 경우 지원을 제공하지 않는 영역을 생략합니다.
- 클러스터의 다른 리전을 선택합니다.

로그에서 문제를 확인한 후 클러스터를 제거하고 다시 생성합니다.

클러스터 생성에 대한 자세한 내용은 클러스터 생성 소개 를 참조하십시오.

1.23. GRAFANA 문제 해결

Grafana 탐색기에서 시간이 많이 걸리는 메트릭을 쿼리할 때 게이트웨이 시간 제한 오류가 발생할 수 있습니다.

1.23.1. 증상: Grafana explorer 게이트웨이 시간 초과

Grafana explorer에서 시간이 많이 걸리는 메트릭을 쿼리할 때 게이트웨이 시간 제한 오류가 발생하면 open-cluster-management-observability 네임스페이스의 Grafana로 인해 시간 초과가 발생할 수 있습니다.

1.23.2. 문제 해결: Grafana 구성

이 문제가 있는 경우 다음 단계를 완료합니다.

1. Grafana의 기본 구성에 예상 시간 제한 설정이 있는지 확인합니다.
   a. Grafana의 기본 시간 초과 설정을 확인하려면 다음 명령을 실행합니다.

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

   다음 시간 초과 설정이 표시되어야 합니다.

```
[dataproxy]
timeout = 300
```

```
dial_timeout = 30
keep_alive_seconds = 300
```

b. **Grafana에 대한 기본 데이터 소스 쿼리 타임아웃을 확인하려면 다음 명령을 실행합니다.**

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

다음 시간 초과 설정이 표시되어야 합니다.

```
queryTimeout: 300s
```

2. **Grafana의 기본 구성에 예상 시간 제한 설정이 있는 경우 다음 명령을 실행하여 open-cluster-management-observability 네임스페이스에서 Grafana를 구성할 수 있습니다.**

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

Grafana 페이지를 새로 고치고 메트릭을 다시 쿼리합니다. 게이트웨이 시간 제한 오류가 더 이상 표시되지 않습니다.

1.24. 배치 규칙을 사용하여 선택하지 않은 로컬 클러스터 문제 해결

관리 클러스터는 배치 규칙으로 선택되지만, 또한 관리하는 허브 클러스터인 local-cluster 는 선택되지 않습니다. 배치 규칙 사용자에게는 local-cluster 네임스페이스에서 관리 클러스터 리소스를 가져올 수 있는 권한이 부여되지 않습니다.

1.24.1. 증상: 관리형 클러스터로 선택되지 않은 로컬 클러스터 문제 해결

모든 관리 클러스터는 배치 규칙으로 선택되지만 local-cluster 는 그렇지 않습니다. 배치 규칙 사용자에게는 local-cluster 네임스페이스에서 관리 클러스터 리소스를 가져올 수 있는 권한이 부여되지 않습니다.

1.24.2. 문제 해결: 관리형 클러스터로 선택되지 않은 로컬 클러스터 문제 해결

더 이상 사용되지 않음: PlacementRule

이 문제를 해결하려면 **local-cluster** 네임스페이스에서 **managedcluster** 관리 권한을 부여해야 합니다. 다음 단계를 완료합니다.

1. 관리 클러스터 목록에 **local-cluster** 가 포함되어 있고 배치 규칙 결정 목록에 **local-cluster** 가 표시되지 않는지 확인합니다. 다음 명령을 실행하여 결과를 확인합니다.

```
% oc get managedclusters
```

local-cluster 가 결합되었지만 **PlacementRule** 의 **YAML**에 없는 샘플 출력에서 참조하십시오.

```
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** 파일에 역할을 생성하여 **local-cluster** 네임스페이스에서 **managedcluster** 관리 권한을 부여합니다. 다음 예제를 참조하십시오.

```
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** 리소스를 생성하여 배치 규칙 사용자에게 **local-cluster** 네임스페이스에 대한 액세스 권한을 부여합니다. 다음 예제를 참조하십시오.

```
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
```

1.25. 애플리케이션 KUBERNETES 배포 버전 문제 해결

더 이상 사용되지 않는 **Kubernetes apiVersion** 이 있는 관리형 클러스터는 지원되지 않을 수 있습니다. 더 이상 사용되지 않는 **API** 버전에 대한 자세한 내용은 **Kubernetes** 문제를 참조하십시오.

1.25.1. 증상: 애플리케이션 배포 버전

Subscription YAML 파일에서 하나 이상의 애플리케이션 리소스가 더 이상 사용되지 않는 **API**를 사용하는 경우 다음 오류와 유사한 오류가 표시될 수 있습니다.

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

또는 인스턴스의 경우 이름이 **old.yaml** 인 **YAML** 파일의 새 **Kubernetes API** 버전이 있으면 다음과 같은 오류가 발생할 수 있습니다.

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

1.25.2. 문제 해결: 애플리케이션 배포 버전

1. 리소스에서 **apiVersion** 을 업데이트합니다. 예를 들어 서브스크립션 **YAML** 파일에 *배포* 유형에 대한 오류가 표시되면 **extensions/v1beta1** 에서 **apps/v1** 로 **apiVersion** 을 업데이트해야 합니다.

   다음 예제를 참조하십시오.

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

2. 관리 클러스터에서 다음 명령을 실행하여 사용 가능한 버전을 확인합니다.

```
kubectl explain <resource>
```

3. **VERSION** 을 확인합니다.

1.26. 성능이 저하된 조건으로 KLUSTERLET 문제 해결

Klusterlet 성능이 저하된 조건은 관리 클러스터에서 **Klusterlet** 에이전트의 상태를 진단하는 데 도움이 될 수 있습니다. **Klusterlet**이 성능 저하된 상태에 있는 경우 관리 클러스터의 **Klusterlet** 에이전트에 문제를 해결해야 하는 오류가 있을 수 있습니다. **Klusterlet degraded conditions that are set to True** 를 참조하십시오.

1.26.1. 증상: Klusterlet은 성능 저하 상태에 있습니다.

관리 클러스터에 **Klusterlet**을 배포한 후 **KlusterletRegistrationDegraded** 또는 **KlusterletWorkDegraded** 상태가 ***True*** 로 표시됩니다.

1.26.2. 문제 식별: Klusterlet은 성능 저하된 상태에 있습니다.

1. 관리 클러스터에서 다음 명령을 실행하여 **Klusterlet** 상태를 확인합니다.

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

2. **KlusterletRegistrationDegraded** 또는 **KlusterletWorkDegraded** 를 선택하여 조건이 **True** 로 설정되어 있는지 확인합니다. 나열된 모든 성능이 저하된 조건에 대한 *문제*를 복구합니다.

1.26.3. 문제 해결: Klusterlet은 성능 저하 상태에 있습니다.

다음 성능 저하 상태의 목록과 이러한 문제를 해결하는 방법을 참조하십시오.

- 상태가 ***True*** 이고 조건 이유가 있는 **KlusterletRegistrationDegraded** 조건이 ***BootStrapSecretMissing*** 인 경우 **open-cluster-management-agent** 네임스페이스에 부트스트랩 시크릿을 생성해야 합니다.

- **KlusterletRegistrationDegraded** 조건이 ***True*** 로 표시되고 조건 이유가 ***BootstrapSecretError*** 또는 ***BootstrapSecretUnauthorized*** 이면 현재 부트스트랩 보안이 유효하지 않습니다. 현재 부트스트랩 시크릿을 삭제하고 **open-cluster-management-agent** 네임스페이스에 유효한 부트스트랩 시크릿을 다시 생성합니다.

- **KlusterletRegistrationDegraded** 및 **KlusterletWorkDegraded** 가 ***True*** 로 표시되고 조건 이유가 ***HubKubeConfigSecretMissing*** 인 경우 **Klusterlet**을 삭제하고 다시 생성합니다.

- **KlusterletRegistrationDegraded** 및 **KlusterletWorkDegraded** 가 ***True*** 로 표시되고 조건 이유가 ***ClusterNameMissing***,***KubeConfig*** **Mising** ,***HubConfigSecretError***, 또는 ***HubConfigSecretUnauthorized***, **open-cluster-management-agent** 네임스페이스에서 **hub cluster kubeconfig** 시크릿을 삭제합니다. 등록 에이전트는 다시 부팅되어 새 **hub** 클러스터 **kubeconfig** 시크릿을 가져옵니다.

- **KlusterletRegistrationDegraded** 가 ***True*** 를 표시하고 조건 이유가 ***GetRegistrationDeploymentFailed*** 또는 ***UnavailableRegistrationPod*** 인 경우 상태 메시지를 확인하여 문제 세부 정보를 가져오고 해결하려고 할 수 있습니다.

- **KlusterletWorkDegraded** 에 ***True*** 가 표시되고 조건 이유가 ***GetWorkDeploymentFailed*** , 또는 ***UnavailableWorkPod*** 인 경우 조건 메시지를 확인하여 문제 세부 정보를 가져오고 해결하려고 할 수 있습니다.

1.27. 오브젝트 스토리지 채널 시크릿 문제 해결

SecretAccessKey 를 변경하면 **Object** 스토리지 채널의 구독에서 업데이트된 보안을 자동으로 선택할 수 없으며 오류가 발생합니다.

1.27.1. 증상: 오브젝트 스토리지 채널 시크릿

오브젝트 스토리지 채널의 서브스크립션은 업데이트된 보안을 자동으로 선택할 수 없습니다. 이렇게 하면 서브스크립션 **Operator**가 오브젝트 스토리지에서 관리 클러스터로 리소스를 조정하지 못하도록 합니다.

1.27.2. 문제 해결: 오브젝트 스토리지 채널 시크릿

시크릿을 생성하기 위해 인증 정보를 수동으로 입력한 다음 채널 내의 시크릿을 참조해야 합니다.

1. 단일 서브스크립션 **Operator**를 조정하려면 서브스크립션 **CR**에 주석을 답니다. 다음 데이터

사양을 참조하십시오.

```
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** 를 실행하여 테스트합니다.

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

명령을 실행한 후 애플리케이션 콘솔로 이동하여 리소스가 관리되는 클러스터에 배포되었는지 확인할 수 있습니다. 또는 관리 클러스터에 로그인하여 애플리케이션 리소스가 지정된 네임스페이스에서 생성되었는지 확인할 수 있습니다.

1.28. 관찰 기능 문제 해결

관찰 기능 구성 요소를 설치하면 구성 요소가 중단되고 설치 상태가 표시됩니다.

1.28.1. 증상: MultiClusterObservability 리소스 상태가 중단됨

설치 후 **Observability CRD**(사용자 정의 리소스 정의)를 생성한 후 **observability** 상태가 **Installing** 상태에 있는 경우 **spec:storageConfig:storageClass** 매개변수에 대해 정의된 값이 없을 수 있습니다. 또

는 관찰 기능 구성 요소에서 기본 **storageClass** 를 자동으로 찾지만 스토리지 값이 없는 경우 구성 요소는 **Installing** 상태로 유지됩니다.

1.28.2. 문제 해결: MultiClusterObservability 리소스 상태가 중단됨

이 문제가 있는 경우 다음 단계를 완료합니다.

1. 관찰 기능 구성 요소가 설치되었는지 확인합니다.
   a. **multicluster-observability-operator** 를 확인하려면 다음 명령을 실행합니다.

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

   b. 적절한 **CRD**가 있는지 확인하려면 다음 명령을 실행합니다.

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

   구성 요소를 활성화하기 전에 다음 **CRD**를 표시해야합니다.

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

2. 베어 메탈 클러스터에 대한 자체 **storageClass**를 생성하는 경우 **NFS**를 사용하여 영구 스토리지를 참조하십시오.

3. 관찰 기능 구성 요소가 기본 **storageClass**를 찾을 수 있도록 **multicluster-observability-operator** 사용자 정의 리소스 정의에서 **storageClass** 매개변수를 업데이트합니다. 매개변수는 다음 값과 유사할 수 있습니다.

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

설치가 완료되면 관찰 기능 구성 요소 상태가 ***Ready*** 상태로 업데이트됩니다. 설치가 실패하면 ***Fail*** 상태가 표시됩니다.

1.29. OPENSHIFT 모니터링 서비스 문제 해결

관리 클러스터의 관찰 기능 서비스는 **OpenShift Container Platform** 모니터링 스택에서 메트릭을 스크랩해야 합니다. **OpenShift Container Platform** 모니터링 스택이 준비되지 않은 경우 **metrics-collector** 가 설치되지 않습니다.

1.29.1. 증상: OpenShift 모니터링 서비스가 준비되지 않음

endpoint-observability-operator-x Pod는 **openshift-monitoring** 네임스페이스에서 **prometheus-k8s** 서비스를 사용할 수 있는지 확인합니다. 서비스가 **openshift-monitoring** 네임스페이스에 없으면 **metrics-collector** 가 배포되지 않습니다. 다음과 같은 오류 메시지가 표시될 수 있습니다. **prometheus** 리소스를 가져오지 못했습니다.

1.29.2. 문제 해결: OpenShift 모니터링 서비스가 준비되지 않음

이 문제가 있는 경우 다음 단계를 완료합니다.

1. **OpenShift Container Platform** 클러스터에 로그인합니다.
2. **openshift-monitoring** 네임스페이스에 액세스하여 **prometheus-k8s** 서비스를 사용할 수 있는지 확인합니다.
3. 관리 클러스터의 **open-cluster-management-addon-observability** 네임스페이스에서 **endpoint-observability-operator-x Pod**를 다시 시작합니다.

1.30. METRICS-COLLECTOR 문제 해결

관리 클러스터에서 **observability-client-ca-certificate** 시크릿이 새로 고쳐지지 않으면 내부 서버 오류가 발생할 수 있습니다.

1.30.1. 증상: metrics-collector에서 observability-client-ca-certificate를 확인할 수 없습니다

메트릭을 사용할 수 없는 관리형 클러스터가 있을 수 있습니다. 이 경우 **metrics-collector** 배포에서 다음 오류가 발생할 수 있습니다.

```
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")
```

1.30.2. 문제 해결: metrics-collector에서 observability-client-ca-certificate를 확인할 수 없습니다

이 문제가 있는 경우 다음 단계를 완료합니다.

1. 관리 클러스터에 로그인합니다.

2. **open-cluster-management-addon-observability** 네임스페이스에 있는, **observability-controller-open-cluster-management.io-observability-signer-client-cert** 라는 시크릿을 삭제합니다. 다음 명령을 실행합니다.

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

참고: **observability-controller-open-cluster-management.io-observability-signer-client-cert** 는 새 인증서로 자동으로 다시 생성됩니다.

metrics-collector 배포가 다시 생성되고 **observability-controller-open-cluster-management.io-observability-signer-client-cert** 시크릿이 업데이트됩니다.

1.31. POSTGRESQL 공유 메모리 오류 문제 해결

대규모 환경이 있는 경우 검색 결과 및 애플리케이션의 토폴로지 보기에 영향을 주는 **PostgreSQL** 공유 메모리 오류가 발생할 수 있습니다.

1.31.1. 증상: PostgreSQL 공유 메모리 오류

검색**-api** 로그에 다음과 같은 오류 메시지가 표시됩니다**. ERROR: could not resize shared memory segment "/PostgreSQL.1083654800" to 25031264 bytes: No space left on device (SQLSTATE 53100)**

1.31.2. 문제 해결: PostgreSQL 공유 메모리 오류

문제를 해결하려면 **search-postgres ConfigMap**에 있는 **PostgreSQL** 리소스를 업데이트합니다. 리소스를 업데이트하려면 다음 단계를 완료합니다.

1. 다음 명령을 실행하여 **open-cluster-management** 프로젝트로 전환합니다.

-

```
oc project open-cluster-management
```

2. **search-postgres Pod** 메모리를 늘립니다. 다음 명령은 메모리를 **16Gi** 로 늘립니다.

```
oc patch search -n open-cluster-management search-v2-operator --type json -p '[{"op": "add",
"path": "/spec/deployments/database/resources", "value": {"limits": {"memory": "16Gi"},
"requests": {"memory": "32Mi", "cpu": "25m"}}}]'
```

3. 다음 명령을 실행하여 검색 **Operator**가 변경 사항을 덮어쓰지 않도록 합니다.

```
oc annotate search search-v2-operator search-pause=true
```

4. 다음 명령을 실행하여 **search-postgres YAML** 파일에서 리소스를 업데이트합니다.

```
oc edit cm search-postgres -n open-cluster-management
```

리소스 증가는 다음 예제를 참조하십시오.

```
postgresql.conf: |-
  work_mem = '128MB' # Higher values allocate more memory
  max_parallel_workers_per_gather = '0' # Disables parallel queries
  shared_buffers = '1GB' # Higher values allocate more memory
```

종료하기 전에 변경 사항을 저장해야 합니다.

5. 다음 명령을 실행하여 **postgres** 및 **api pod**를 다시 시작합니다.

```
oc delete pod search-postgres-xyz search-api-xzy
```

6. 변경 사항을 확인하려면 **search-postgres YAML** 파일을 열고 다음 명령을 실행하여 **postgresql.conf:** 에 변경한 사항이 있는지 확인합니다.

```
oc get cm search-postgres -n open-cluster-management -o yaml
```

환경 변수 추가에 대한 자세한 내용은 사용자 지정 및 구성 검색을 참조하십시오.

1.32. THANOS COMPACTOR의 블록 오류 문제 해결

Thanos compactor의 블록이 손상되었음을 나타내는 블록 오류 메시지가 표시될 수 있습니다.

1.32.1. 증상: Thanos compactor의 블록 오류

Kubernetes용 **Red Hat Advanced Cluster Management**를 업그레이드한 후 **oc logs observability-thanos-compact-0** 명령을 사용하여 **Thanos compactor**의 로그를 확인한 후 로그에 다음 오류 메시지가 표시됩니다.

```
ts=2024-01-24T15:34:51.948653839Z caller=compact.go:491 level=error msg="critical error detected; halting" err="compaction: group 0@15699422364132557315: compact blocks [/var/thanos/compact/compact/0@15699422364132557315/01HKZGQGJCKQWF3XMA8EXAMPLE /var/thanos/compact/compact/0@15699422364132557315/01HKZQK7TD06J2XWGR5EXAMPLE /var/thanos/compact/compact/0@15699422364132557315/01HKZYEZ2DVDQXF1STVEXAMPLE /var/thanos/compact/compact/0@15699422364132557315/01HM05APAHXBQSNC0N5EXAMPLE]: populate block: chunk iter: cannot populate chunk 8 from block 01HKZYEZ2DVDQXF1STVEXAMPLE: segment index 0 out of range"
```

1.32.2. 문제 해결: *thanos bucket verify* 명령 추가

오브젝트 스토리지 구성에 **thanos bucket verify** 명령을 추가합니다. 다음 단계를 완료합니다.

1. 오브젝트 스토리지 구성에 **thanos bucket verify** 명령을 추가하여 블록 오류를 해결합니다. 다음 명령을 사용하여 **observability-thanos-compact Pod**에서 구성을 설정합니다.

```
oc rsh observability-thanos-compact-0
[..]
thanos tools bucket verify -r --objstore.config="$OBJSTORE_CONFIG" --objstore-backup.config="$OBJSTORE_CONFIG" --id=01HKZYEZ2DVDQXF1STVEXAMPLE
```

2. 이전 명령이 작동하지 않으면 차단이 손상될 수 있으므로 삭제 블록을 표시해야 합니다. 다음 명령을 실행합니다.

```
thanos tools bucket mark --id "01HKZYEZ2DVDQXF1STVEXAMPLE" --objstore.config="$OBJSTORE_CONFIG" --marker=deletion-mark.json --details=DELETE
```

3. 삭제를 차단한 경우 다음 명령을 실행하여 표시된 블록을 정리합니다.

```
thanos tools bucket cleanup --objstore.config="$OBJSTORE_CONFIG"
```

1.33. 설치 후 SUBMARINER가 연결되지 않음 문제 해결

구성 후 Submariner가 올바르게 실행되지 않으면 다음 단계를 완료하여 문제를 진단합니다.

1.33.1. 증상: 설치 후 하위 시스템이 연결되지 않음

설치 후 하위 네트워크가 통신하지 않습니다.

1.33.2. 문제 식별: 설치 후 Submariner가 연결되지 않음

Submariner 배포 후 네트워크 연결이 설정되지 않은 경우 문제 해결 단계를 시작합니다. Submariner를 배포할 때 프로세스가 완료될 때까지 몇 분이 걸릴 수 있습니다.

1.33.3. 문제 해결: 설치 후 Submariner가 연결되지 않음

배포 후 Submariner가 올바르게 실행되지 않으면 다음 단계를 완료합니다.

1. Submariner의 구성 요소가 올바르게 배포되었는지 확인하려면 다음 요구 사항을 확인하십시오.
   - 하위 mariner-addon 포드는 hub 클러스터의 open-cluster-management 네임스페이스에서 실행되고 있습니다.
   - 다음 Pod는 각 관리 클러스터의 submariner-operator 네임스페이스에서 실행됩니다.
     - submariner-addon
     - submariner-gateway
     - submariner-routeagent
     - submariner-operator

- **Submariner-globalnet (ClusterSet에서 Globalnet이 활성화된 경우에만)**
- **submariner-lighthouse-agent**
- **submariner-lighthouse-coredns**
- **Submariner-networkplugin-syncer (지정된 CNI 값이 OVNKubernetes인 경우에만)**
- **submariner-metrics-proxy**

2. **submariner-addon Pod**를 제외하고 **subctl diagnose all** 명령을 실행하여 필요한 **Pod**의 상태를 확인합니다.

3. **must-gather** 명령을 실행하여 문제 디버깅에 도움이 될 수 있는 로그를 수집하십시오.

1.34. SUBMARINER 애드온 상태 문제 해결

Submariner 애드온을 클러스터 세트의 클러스터에 추가한 후 *연결* 상태,*에이전트 상태 및 게이트웨이 노드 의 상태가* 클러스터에 대한 예기치 않은 상태를 표시합니다.

1.34.1. 증상: 하위 요약 애드온 상태가 저하됨

Submariner 애드온을 클러스터 세트의 클러스터에 추가하면 *게이트웨이 노드,에이전트 상태 및 클러스터의 연결 상태에 다음 상태가* 표시됩니다.

- 레이블이 지정된 게이트웨이 노드
  - **progress:** 게이트웨이 노드의 레이블을 지정하는 프로세스입니다.
  - **nodes not labeled:** 게이트웨이 노드는 레이블이 지정되지 않았기 때문에 레이블이 지정되지 않을 수 있습니다.

- **nodes not labeled:** 게이트웨이 노드는 아직 레이블이 지정되지 않았습니다. 다른 프로세스가 완료될 때까지 프로세스가 대기 중이기 때문일 수 있습니다.
- 레이블이 지정된 노드: 게이트웨이 노드에 레이블이 지정되었습니다.

- 에이전트 상태
  - 진행 중: **Submariner** 에이전트 설치가 시작되었습니다.
  - **degraded: Submariner** 에이전트가 아직 진행 중이므로 올바르게 실행되지 않을 수 있습니다.

- 연결 상태
  - 진행 상황: **Submariner** 애드온을 사용한 연결을 설정하는 프로세스입니다.
  - **degraded:** 연결이 준비되지 않았습니다. 애드온을 설치한 경우에도 프로세스가 여전히 진행 중일 수 있습니다. 연결이 이미 설정되어 실행된 후 발생한 경우 두 클러스터가 서로 연결이 끊어졌습니다. 클러스터가 여러 개인 경우 클러스터의 연결이 끊어진 상태인 경우 모든 클러스터에 **Degraded** 상태가 표시됩니다.

또한 연결된 클러스터와 연결이 끊긴 클러스터도 표시됩니다.

1.34.2. 문제 해결: 잠수함 애드온 상태가 저하됨

- 프로세스가 완료되면 성능이 저하된 상태가 자동으로 확인되는 경우가 많습니다. 표의 상태를 클릭하여 프로세스의 현재 단계를 볼 수 있습니다. 해당 정보를 사용하여 프로세스가 완료되었는지 여부를 확인할 수 있으며 다른 문제 해결 단계를 수행해야 합니다.

- 자체적으로 해결되지 않는 문제의 경우 다음 단계를 완료하여 문제를 해결합니다.

1. **subctl** 유틸리티와 함께 **diagnose** 명령을 사용하여 다음 조건이 있는 경우 **Submariner** 연결에서 일부 테스트를 실행할 수 있습니다.

a. *에이전트 상태* 또는 *연결 상태*는 **Degraded** 상태입니다. **diagnose** 명령은 문제에 대한 자세한 분석을 제공합니다.

b. 모든 것이 콘솔에서 녹색이지만 네트워킹 연결이 제대로 작동하지 않습니다. 진단 명령은 콘솔 외부에 다른 연결 또는 배포 문제가 없는지 확인하는 데 도움이 됩니다. 배포 후 **diagnostics** 명령을 실행하여 문제를 식별하는 것이 좋습니다.

명령을 실행하는 방법에 대한 자세한 내용은 **Submariner**의 진단을 참조하십시오.

2. **Connection status** 에서는 문제가 계속되는 경우 **subctl** 유틸리티 툴의 **diagnose** 명령을 실행하여 두 **Submariner** 클러스터 간의 연결에 대한 자세한 상태를 얻을 수 있습니다. 명령의 형식은 다음과 같습니다.

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

path-to-kubeconfig-file 을 **kubeconfig** 파일의 경로로 바꿉니다. 명령에 대한 자세한 내용은 **Submariner** 설명서의 진단을 참조하십시오.

3. 방화벽 설정을 확인합니다. 경우에 따라 연결 문제는 클러스터가 통신하지 못하도록 방화벽 권한 문제로 인해 발생합니다. 이로 인해 연결 상태가 **degraded**로 표시될 수 있습니다. 다음 명령을 실행하여 방화벽 문제를 확인합니다.

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

path-to-local-kubeconfig 를 클러스터 중 하나의 **kubeconfig** 파일로 교체합니다.

path-to-remote-kubeconfig 를 다른 클러스터의 **kubeconfig** 파일 경로로 교체합니다. **verify** 명령을 **subctl** 유틸리티 툴로 실행하여 두 하위 클러스터 간의 연결을 테스트할 수 있습니다. 명령의 기본 형식은 다음과 같습니다.

4. **Connection status** 에서 문제가 계속되면 **subctl** 유틸리티 툴로 **verify** 명령을 실행하여 두 하위 클러스터 간의 연결을 테스트할 수 있습니다. 명령의 기본 형식은 다음과 같습니다.

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

cluster1 및 **cluster2** 를 테스트 중인 클러스터 이름으로 교체합니다. 명령에 대한 자세

한 내용은 **Submariner** 설명서의 확인을 참조하십시오.

5. 문제 해결 단계에서 문제를 해결한 후 **subctl** 툴과 함께 **benchmark** 명령을 사용하여 추가 진단을 실행할 때 비교할 기반을 설정합니다.

명령의 옵션에 대한 자세한 내용은 **Submariner** 문서의 벤치마크 를 참조하십시오.

1.35. 복원 상태 문제 해결 오류와 함께 완료

백업을 복원하면 리소스가 올바르게 복원되지만 **Red Hat Advanced Cluster Management** 복원 리소스에 **FinishedWithErrors** 상태가 표시됩니다.

1.35.1. 증상: 복원 상태 문제 해결이 오류로 완료됨

Red Hat Advanced Cluster Management에는 **FinishedWithErrors** 상태가 표시되고 **Red Hat Advanced Cluster Management** 복원에서 생성한 하나 이상의 **Velero** 복원 리소스에는 **PartiallyFailed** 상태가 표시됩니다.

1.35.2. 문제 해결: 복원 상태 문제 해결이 오류와 함께 완료

비어 있는 백업에서 복원하는 경우 **FinishedWithErrors** 상태를 무시해도 됩니다.

Kubernetes 복원용 **Red Hat Advanced Cluster Management**에는 모든 **Velero** 복원 리소스의 누적 상태가 표시됩니다. 하나의 상태가 **PartiallyFailed** 이고 다른 하나는 **Completed** 인 경우 표시되는 누적 상태는 하나 이상의 문제가 있음을 알리는 **PartiallyFailed** 입니다.

이 문제를 해결하려면 **PartiallyFailed** 상태로 모든 개별 **Velero** 복원 리소스의 상태를 확인하고 자세한 내용은 로그를 확인합니다. 오브젝트 스토리지에서 직접 로그를 가져오거나 **DownloadRequest** 사용자 정의 리소스를 사용하여 **OADP Operator**에서 로그를 다운로드할 수 있습니다.

콘솔에서 **DownloadRequest** 를 생성하려면 다음 단계를 완료합니다.

1. **Operators > Installed Operators > Create DownloadRequest** 로 이동합니다.
2. **BackupLog** 를 종류로 선택하고 콘솔 지침에 따라 **DownloadRequest** 생성을 완료합니다.

1.36. HUB 클러스터 백업을 복원할 때 일반 리소스가 제거됨

hub 클러스터 백업을 복원하고 Restore.cluster.open-cluster-management.io 리소스에서 생성한 cleanupBeforeRestore:CleanupRestored paramater를 사용하면 acm-resources-generic-schedule 백업에 의해 생성된 리소스가 제거될 수 있습니다.

1.36.1. 증상: 허브 클러스터 백업을 복원할 때 일반 리소스가 제거됩니다.

acm-resources-generic-schedule 백업에 백업된 리소스는 복원된 허브 클러스터에 표시되지 않습니다. 백업 Operator 로그를 확인하는 경우 다음과 유사한 메시지가 표시됩니다.

```
_2023-06-08T13:42:48.066572033Z 2023-06-08T13:42:48.066Z    INFO    Deleting resource
DRPlacementControl [c1-helloworld-placement-1-drpc.c1-helloworld]    {"controller": "restore",
"controllerGroup": "cluster.open-cluster-management.io", "controllerKind": "Restore", "restore":
{"name":"restore-acm","namespace":"open-cluster-management-backup"}
```

1.36.2. 문제 해결: 허브 클러스터 백업을 복원할 때 일반 리소스가 제거됩니다.

다음 조건이 발생하면 리소스가 제거됩니다.

- Secret 또는 ConfigMap 리소스 유형과 cluster.open-cluster-management.io/backup 레이블과 일치하지 않는 acm-resources-generic-schedule 백업에서 지원하는 리소스가 있습니다.
- Restore.cluster.open-cluster-management.io 리소스를 사용하는 복원을 실행하고 cleanupBeforeRestore: 값을 cleanup Restored 로 설정합니다.
- 최신 Red Hat Advanced Cluster Management 백업 세트에는 acm-resources-schedule 백업이 포함되어 있지 않으므로 이전 버전의 백업이 선택됩니다. 결과적으로 acm-resources-schedule 백업에는 acm-resources-generic-schedule 백업과 다른 타임스탬프가 있습니다. 복원 후 작업 중에 cleanRestore 옵션이 처리되면 acm-resources-schedule 백업과 동일한 타임스탬프가 없기 때문에 모든 일반 리소스가 정리됩니다.

문제를 해결하려면 복원 작업을 다시 실행하고 cleanupBeforeRestore: 값을 None 으로 설정합니다.'

1.37. 여러 줄 YAML 구문 분석 문제 해결

fromSecret 함수를 사용하여 Secret 리소스의 콘텐츠를 Route 리소스에 추가하려는 경우 콘텐츠가 잘못 표시됩니다.

1.37.1. 증상: 여러 줄 YAML 구문 분석 문제 해결

관리 클러스터 및 허브 클러스터가 동일한 클러스터인 경우 인증서 데이터가 수정되므로 내용이 템플릿 JSON 문자열로 구문 분석되지 않습니다. 다음과 같은 오류 메시지가 표시될 수 있습니다.

```
message: >-
        [spec.tls.caCertificate: Invalid value: "redacted ca certificate
        data": failed to parse CA certificate: data does not contain any
        valid RSA or ECDSA certificates, spec.tls.certificate: Invalid
        value: "redacted certificate data": data does not contain any valid
        RSA or ECDSA certificates, spec.tls.key: Invalid value: "": no key specified]
```

1.37.2. 문제 해결: 여러 줄 YAML 구문 분석 문제 해결

hub 클러스터 및 관리 클러스터 fromSecret 값을 검색하도록 인증서 정책을 구성합니다. autoindent 함수를 사용하여 다음 콘텐츠로 인증서 정책을 업데이트합니다.

```
            tls:
              certificate: |
                {{ print "{{hub fromSecret "open-cluster-management" "minio-cert" "tls.crt"
hub}}" | base64dec | autoindent }}
```