Red Hat Advanced Cluster Management for Kubernetes 2.10

发行注记

发行注记

Last Updated: 2024-07-01

Red Hat Advanced Cluster Management for Kubernetes 2.10 发行注记

发行注记

法律通告

摘要

参阅更多与发行注记相关的信息，了解新的、勘误更新、已知问题、弃用和删除以及 GDPR 和 FIPS 就绪的产品注意事项。

目录

第 1 章 发行注记

了解当前版本。

注： Red Hat Advanced Cluster Management 的 2.6 和更早的版本已 *从服务中删除*，且不再被支持。2.6 及更早的版本文档没有更新。其文档可能仍然可用，但不再有任何新的勘误或其他更新。

- Red Hat Advanced Cluster Management for Kubernetes 的新内容
- 勘误更新
- 限制和已知问题
- 弃用和删除
- Red Hat Advanced Cluster Management for Kubernetes 针对 GDPR 的注意事项
- FIPS 就绪性
- Observability 支持

如果您在当前支持的某个版本或产品文档时遇到问题，请访问 红帽支持，您可以在其中进行故障排除、查看知识库文章、与支持团队连接，或者创建一个问题单。您必须使用您的凭证登录。您还可以访问红帽客户门户文档，Red Hat Customer Portal FAQ 。

1.1. RED HAT ADVANCED CLUSTER MANAGEMENT FOR KUBERNETES 的新内容

Red Hat Advanced Cluster Management for Kubernetes 为您提供了整个 Kubernetes 域的可见性，以及内置监管、集群生命周期管理和应用程序生命周期管理功能。在这个版本中，您可以在更多环境中移至管理集群，应用程序的 GitOps 集成等等。

访问 支持列表，了解 hub 集群和受管集群要求和支持。

重要：一些功能和组件作为技术预览发布。

- Clusters
- 多集群全局 hub
- 应用程序
- Observability（可观察性）
- 监管
- 备份和恢复
- 网络

1.1.1. Cluster

集群生命周期组件和功能位于 multicluster engine operator 中，它是一个增强集群团队管理的软件操作员。multicluster engine operator 支持跨云和数据中心的 OpenShift Container Platform 和 Kubernetes 集群生命周期管理。OpenShift Container Platform 是此技术的先决条件。

- 多集群引擎 operator （集群）的文档位于产品文档的 Cluster Lifecycle 部分中。
- 从 *Cluster Lifecycle* 查看 multicluster engine operator 2.5 ../../html-single/clusters 的新内容。
- 查看 集群生命周期概述 中的任务和支持信息。

1.1.2. 多集群全局 hub

您可以在 Red Hat Advanced Cluster Management 备份和恢复功能中使用多集群全局 hub。这些功能可让您访问恢复解决方案和基本资源。如需更多信息，请参阅多集群全局 hub 备份（技术预览）。

有关其他多集群全局 hub 主题，请参阅 多集群全局 hub。

1.1.3. 应用程序

使用新的 **.status.subscription** 字段，您可以看到整个订阅状态，而不是只查看单个软件包的软件包状态。

有关其他应用程序主题，请参阅管理应用程序。

1.1.4. Observability（可观察性）

- 现在，hub 收集器指标总是被收集并发送到 Red Hat Advanced Cluster Management Thanos 实例。当您启用 Observability 时，该服务会在 hub 集群上的 **open-cluster-management-observability** 命名空间中启动一个 **endpoint-operator** 和 **metrics-collector** pod。**MultiClusterObservability** Operator 启动并管理 **endpoint-operator** 和 **metrics-collector** pod。Observability 附加组件不再控制 pod。请参阅 Observability 架构 以了解更多信息。
- 您可以使用 Grafana 仪表板查看托管的 control plane 集群容量估算以及现有的托管的 control plane 资源使用率。托管 control plane Observability 是集群生命周期的一部分，或 multicluster engine operator，以及 Red Hat Advanced Cluster Management 集成 。

请参阅 Observability 服务简介。

1.1.5. 监管

- **技术预览** 启用策略合规历史记录 API，以存储和查询 hub 集群的合规性历史记录事件。请参阅策略合规历史记录 API （技术预览）。要启用 API 请查看 ，策略合规历史记录（技术预览）。
- 配置 Gatekeeper operator Webhook 的操作，以管理准入事件。详情请参阅 管理 Gatekeeper operator 策略。
- 启用 Policy Generator 来处理 Helm chart，并在策略中添加描述。请参阅 Policy Generator 配置参考表 中的 **policyDefaults.policyLabels** 和 **policies.policyLabels** 可选规格等。
- 您可以使用 **ConfigurationPolicy** 资源中的 **recordDiff** 参数为 **ConfigurationPolicy** 资源启用 *diff 日志记录*。**object-template** 和受管集群上的对象之间的区别记录在受管集群的 **config-policy-controller** pod 中。详情请参阅 配置调试日志。
- 启用 Policy Generator 来处理 Helm chart，并在策略中添加描述。如需了解更多详细信息，请参阅 策略生成器配置 参考表。
- 现在，您可以配置监管框架的并发性。如需了解更多详细信息 ，请参阅策略控制器高级配置。

- Gatekeeper operator 在 **auditFromCache** 审计中的自定义资源定义中公开设置，该设置默认为禁用。您可以启用 **auditFromCache**，然后为同步详情设置 **config.gatekeeper.sh**。详情请参阅管理 Gatekeeper operator 策略。
- 启用 **auditEventsInvolvedNamespace** 来管理要创建的命名空间审计事件，并使用 **admissionEventsInvolvedNamespace** 来管理要创建的命名空间准入事件。请参阅管理 Gatekeeper operator 策略。
- **技术预览：** 您可以使用 Operator 策略控制器监控和安装集群中的 Operator Lifecycle Manager (OLM) Operator。如需更多信息，请参阅 Operator 策略控制器（技术预览）。
- 使用 **放置资源** 定义您要放置策略的位置。如需了解更多详细信息，请参阅策略概述。

如需了解更多有关仪表板和策略框架的信息，请参阅监管。

1.1.6. 备份和恢复

- **backup-restore-enabled** 策略包括一个名为 **OADP-channel** 的新模板。使用 **OADP-channel** 模板来防止备份和恢复 Operator 使用错误的自定义资源定义运行。如需了解更多详细信息，请参阅验证备份或恢复配置。
- 当您在 **MultiClusterHub** 中启用备份组件时，集群备份和恢复 Operator Helm Chart 安装策略。新的 **backup-restore-auto-import** 会告知您自动受管集群导入功能的问题。如需了解更多详细信息，请参阅验证备份或恢复配置。

请参阅备份和恢复以了解 hub 集群的灾难恢复解决方案。

1.1.7. 网络

- 您可以在 IBM Power Systems Virtual Server 上部署 Submariner。请参阅 使用控制台部署 Submariner 以了解更多信息。
- **技术预览：** 现在，您还可以在 IBM Cloud 上的 Red Hat OpenShift 上部署 Submariner。请参阅 使用控制台部署 Submariner 以了解更多信息。

请参阅 网络。

1.1.8. 了解有关此发行版本的更多信息

- 欢迎使用 Red Hat Advanced Cluster Management for Kubernetes 包括了 Red Hat Advanced Cluster Management for Kubernetes 的概述。
- 请参阅 Red Hat Advanced Cluster Management 发行注记中的*已知问题和限制*。
- 多集群架构包括了与该产品主要组件相关的详细信息。
- 请参阅 Red Hat Advanced Cluster Management 故障排除指南中的支持信息和更多信息。
- 访问开源的 *Open Cluster Management* 存储库，以获取开源社区的交互、增长和贡献。要参与，请参阅 open-cluster-management.io。如需更多信息，请访问 GitHub 存储库。

1.2. 勘误更新

默认情况下，勘误更新会在发布时自动应用。当发行版本可用时，会在此处发布详情。

重要：为了参考，勘误 链接和 Jira 号可能会添加到内容中并在内部使用。用户可能不能使用访问的链接。

有关升级的更多信息，请参阅使用 operator 升级。

1.2.1. Errata 2.10.3

- 当 **ConfigurationPolicy** 控制器在将策略与集群中的对象进行比较时，添加缺少的日志消息来报告错误。(ACM-10612)
- 修复了在删除后快速重新创建策略时的问题，有时会导致合规状态不会在受管集群上填充。(ACM-10664)
- 修复了在 **governance-policy-framework** pod 中造成一些不必要的日志的问题，并带有默认的日志详细程度设置。(ACM-10693)
- 修复了在安装或卸载 Gatekeeper Operator 时需要重启 **governance-policy-framework** pod 的问题。Red Hat Advanced Cluster Management for Kubernetes 与 Gatekeeper 集成现在在没有重启 pod 的情况下启用或禁用。(ACM-10966)
- 修复了在将对象与策略定义进行比较时，带有 **MustOnlyHave** 合规类型的 **ConfigurationPolicy** 资源没有因素根级别密钥。(ACM-10877)
- 修复了 Policy Generator 中存在一个问题，其中 **policyDefaults** parameter 部分外的一些放置覆盖没有正确地覆盖默认设置。(ACM-11075)
- 修复了当一些云供应商在 Red Hat OpenShift Container Platform 4.15 中置备名为 **application-manager** 的应用程序附加组件服务帐户时，阻止 Red Hat Advanced Cluster Management **gitopsCluster** 控制器为 Argo CD push 模型自动生成受管集群 secret 的问题。(ACM-11149)
- 修复了 **OperatorPolicy** 合规消息重复显示同一消息的问题，但在 Operator 安装失败时，带有多个错误的结果会有所不同。(ACM-11204)
- 为一个或多个产品容器镜像提供更新。

1.2.2. Errata 2.10.2

- 修复了在更新或删除 **AddOnDeploymentConfig** 附加组件后 **multicluster-observability-controller** 没有协调的问题。(ACM-10406)
- 修复了 **multicluster-observability-controller** 没有改为其 **AddOnDeploymentConfig** add-on 的 **nodePlacement** 字段中设置的配置的问题。(ACM-10811)
- 修复了 **multicluster-observability-controller** 中的升级问题，导致 **ServiceAccount** 的持续更新。持续更新会导致多次生成多个 **Secret** 对象。(ACM-10967)
- 为一个或多个产品容器镜像提供更新。

1.2.3. Errata 2.10.1

- 修复了在不使用 **cluster.open-cluster-management.io/backup: cluster-activation** 标签的情况下，使用 Red Hat Advanced Cluster Management for Kubernetes 备份和恢复功能以及备份 **managedcluster** 命名空间的用户可能会出现问题。问题导致受管集群命名空间在恢复后处于 **Terminating** 状态。(ACM-9780)

- 修复了当策略被更新时，策略可能会临时设置为 **不合规** 的问题，而 **governance-policy-framework** pod 在受管集群中关闭。(ACM-10402)

- 修复了在刷新策略详情前，导致控制台简要显示未找到的新策略的问题。(ACM-10416)

- 为一个或多个产品容器镜像提供更新。

1.3. 已知问题

检查应用程序管理的已知问题。以下列表包含本发行版本的已知问题，或从上一版本中继承的问题。

对于 Red Hat OpenShift Container Platform 集群，请参阅 OpenShift Container Platform 已知问题。

有关弃用和删除的更多信息，请参阅弃用和删除。

集群管理或 *集群生命周期* 由带有或没有 Red Hat Advanced Cluster Management 的多集群引擎 operator 提供。有关仅适用于 Red Hat Advanced Cluster Management 的集群管理，请参阅以下已知问题和限制。大多数已知的与集群管理相关的问题包括在 集群生命周期文档中。

- 已知的与安装相关的问题

- 已知的业务连续问题

- 已知的控制台问题

- 已知的应用程序问题

- 已知的可观察性问题

- 已知的监管问题

- 已知的已知问题

1.3.1. 已知的与安装相关的问题

查看安装和升级的已知问题。以下列表包含本发行版本的已知问题，或从上一版本中继承的问题。

对于 Red Hat OpenShift Container Platform 集群，请参阅 OpenShift Container Platform 已知问题。

有关弃用和删除的更多信息，请参阅弃用和删除。

1.3.1.1. 使用升级卸载和重新安装早期版本可能会失败

如果稍后要安装早期版本，然后从 OpenShift Container Platform 卸载 Red Hat Advanced Cluster Management 可能会导致问题。例如，从 OpenShift Container Platform 卸载 Red Hat Advanced Cluster Management 时，安装早期版本的 Red Hat Advanced Cluster Management 并升级该版本，升级可能会失败。如果没有删除 StorageVersionMigration 自定义资源，升级会失败。

卸载 Red Hat Advanced Cluster Management 时，您需要在重新安装和升级前手动删除以前的 StorageVersionMigration。

例如，如果您从 OpenShift Container Platform 卸载 Red Hat Advanced Cluster Management 2.10，以使用早期版本的 Red Hat Advanced Cluster Management，然后尝试再次升级到 2.10，除非删除 StorageVersionMigration 资源，否则升级会失败。

1.3.1.2. 带有 ARM 聚合流的基础架构 operator 错误

安装 infrastructure-operator 时，与 ARM 聚合流无法正常工作。将 ALLOW_CONVERGED_FLOW 设置为 false 以解决这个问题。

1. 运行以下命令来创建 ConfigMap 资源：

```
oc create -f
```

2. 运行 oc apply -f 以应用您的文件。请参阅以下文件示例，并将 ALLOW_CONVERGED_FLOW 设置为 false ：

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-assisted-service-config
  namespace: assisted-installer
data:
  ALLOW_CONVERGED_FLOW: false
```

3. 使用以下命令注解 agentserviceconfig ：

```
oc annotate --overwrite AgentServiceConfig agent unsupported.agent-install.openshift.io/assisted-service-configmap=my-assisted-service-config
```

当问题解决时，代理会出现在清单中。

1.3.1.3. 在升级到勘误发行版本后，已弃用的资源会保留

从 2.4.x 升级到 2.5.x，然后再升级到 2.6.x，受管集群命名空间中的已弃用资源可能会被保留。如果版本 2.6.x 从 2.4.x 升级，则需要手动删除这些已弃用的资源：

注： 在从 2.5.x 升级到 2.6.x 版本前，您需要等待 30 分钟或更长时间。

您可以从控制台中删除，也可以运行类似以下示例的命令，用于您要删除的资源：

```
oc delete -n <managed cluster namespace> managedclusteraddons.addon.open-cluster-management.io <resource-name>
```

查看可能保留的已弃用资源列表：

```
managedclusteraddons.addon.open-cluster-management.io:
policy-controller
manifestworks.work.open-cluster-management.io:
-klusterlet-addon-appmgr
-klusterlet-addon-certpolicyctrl
-klusterlet-addon-crds
-klusterlet-addon-iampolicyctrl
-klusterlet-addon-operator
-klusterlet-addon-policyctrl
-klusterlet-addon-workmgr
```

1.3.1.4. 升级 Red Hat Advanced Cluster Management 后一些 Pod 可能会处于不正常的状态

将 Red Hat Advanced Cluster Management 升级到新版本后，属于 StatefulSet 的少数 pod 可能会处于 failed 状态。这个问题不经常出现，是由一个已知的 Kubernetes 问题造成的。

这个问题的一个临时解决方案是删除失败的 pod。Kubernetes 会自动使用正确的设置重新启动它。

1.3.1.5. OpenShift Container Platform 集群升级失败的状态

当 OpenShift Container Platform 集群处于升级阶段时，集群 Pod 会被重启，并且集群可能在大约 1 到 5 分钟之内会处于升级失败状态。这个行为是正常的，在几分钟后自动解决。

1.3.1.6. Create MultiClusterEngine 按钮无法正常工作

在 Red Hat OpenShift Container Platform 控制台中安装 Red Hat Advanced Cluster Management for Kubernetes 后，会出现一个带有以下信息的弹出窗口：

MultiClusterEngine required

创建一个 MultiClusterEngine 实例来使用这个 Operator。

弹出窗口中的 Create MultiClusterEngine 按钮可能无法正常工作。要临时解决这个问题，在 Provided APIs 部分的 MultiClusterEngine 标题中选择 Create instance。

1.3.2. 已知的业务连续问题

查看 Red Hat Advanced Cluster Management for Kubernetes 中的已知问题。以下列表包含本发行版本的已知问题，或从上一版本中继承的问题。

对于 Red Hat OpenShift Container Platform 集群，请参阅 OpenShift Container Platform 已知问题。

有关弃用和删除的更多信息，请参阅弃用和删除。

1.3.2.1. 备份和恢复已知问题

此处列出了备份和恢复已知问题和限制，如果可用，以及临时解决方案。

1.3.2.1.1. *open-cluster-management-backup* 命名空间处于 *Terminating* 状态

当在 MultiClusterHub 资源中禁用 cluster-backup 组件时，如果您有一个由 Red Hat Advanced Cluster Management 恢复操作创建的 Velero 恢复资源，open-cluster-management-backup 命名空间将处于 Terminating 状态。

Terminating 状态是 Velero 恢复资源等待 restore. velero.io/external-resources-finalizer 的结果。要解决这个问题，请完成以下步骤：

1. 删除所有 Red Hat Advanced Cluster Management 恢复资源，并等待在 MultiClusterHub 资源中禁用集群备份选项前清理 Velero 恢复。

2. 如果您的 open-cluster-management-backup 命名空间已处于 Terminating 状态，请编辑所有 Velero 恢复资源并删除终结器。

3. 允许 Velero 资源删除命名空间和资源。

1.3.2.1.2. 使用 ZTP 流执行重新安装节点，使用 Infrastructure Operator 部署的裸机受管集群

如果使用 Red Hat Advanced Cluster Management 备份和恢复功能，则裸机集群会备份并恢复到辅助 hub 集群，则受管集群会在节点上重新安装，这会销毁现有受管集群。

注： 这只会影响使用零接触置备部署的裸机集群，这意味着它们有 BareMetalHost 资源来管理打开和关闭裸机节点，并附加虚拟介质引导。

如果受管集群的部署中没有使用 BareMetalHost 资源，则没有负面影响。

要临时解决这个问题，请排除主 hub 集群上的受管 BareMetalHost 资源，执行备份和恢复到辅助 hub 集群。

在主 hub 集群上的 BareMetalHost 资源中添加以下标签： velero.io/exclude-from-backup: "true"。

该标签排除备份过程中的任何资源。

当您从恢复中排除 BareMetalHost 资源时，使用零接触置备删除集群不会完全正常工作，因为 BareMetalHost 管理裸机节点的电源。

1.3.2.1.3. *BackupSchedule* 在使用 OADP 1.1.2 或更高版本时显示 *FailedValidation* 状态

启用 Red Hat Advanced Cluster Management 备份和恢复组件并成功创建 DataProtectionApplication 资源后，会创建一个 BackupStorageLocation 资源，状态为 Available。当您使用 OADP 版本 1.1.2 或更高版本时，您可能会在创建 BackupSchedule 资源后收到以下信息，其状态为 FailedValidation ：

```
oc get backupschedule -n open-cluster-management-backup
NAME PHASE MESSAGE
rosa-backup-schedule FailedValidation Backup storage location is not available. Check
velero.io.BackupStorageLocation and validate storage credentials.
```

此错误是由 BackupStorageLocation 资源中的 ownerReference 缺少的值造成的。DataProtectionApplication 资源的值应用作 ownerReference 的值。

要临时解决这个问题，请手动将 ownerReference 添加到 BackupStorageLocation 中：

1. 运行以下命令，打开 oadp-operator.v1.1.2 文件：

```
oc edit csv -n open-cluster-management-backup oadp-operator.v1.1.2
```

2. 通过将 1 替换为 OADP operator CSV 中的 0 来编辑 spec.deployments.label.spec.replicas 的值。

3. 对 YAML 脚本中的 ownerReference 注解进行补丁，如下例所示：

```
metadata:
resourceVersion: '273482'
name: dpa-sample-1
uid: 4701599a-cdf5-48ac-9264-695a95b935a0
namespace: open-cluster-management-backup
ownerReferences: <<

apiVersion: oadp.openshift.io/v1alpha1
```

```
blockOwnerDeletion: true
controller: true
kind: DataProtectionApplication
name: dpa-sample
uid: 52acd151-52fd-440a-a846-95a0d7368ff7
```

4. 将 spec.deployments.label.spec.replicas 的值改回到 1，以使用新设置启动数据保护应用进程。

1.3.2.1.4. Velero 恢复限制

如果在其中恢复数据的新 hub 集群有用户创建的资源，则这个新的 hub 集群可能会有与活跃的 hub 集群不同的配置。例如，在将备份的数据恢复到新的 hub 集群中之前，在这个新的 hub 集群上可能已包括了一个现存的策略。

如果不是恢复的备份的一部分，Velero 会跳过现存的资源，因此新 hub 集群上的策略不会改变，这会导致新 hub 集群和活跃 hub 集群之间的不同配置。

为解决这个问题，集群备份和恢复 Operator 可以运行一个恢复后的操作以清理由用户创建的资源，或在 restore.cluster.open-cluster-management.io 资源时执行不同的恢复操作。

如需更多信息，请参阅安装备份和恢复 Operator 主题。

1.3.2.1.5. 被动配置不显示受管集群

只有在被动 hub 集群上恢复激活数据时，才会显示受管集群。

1.3.2.1.6. 未恢复受管集群资源

当您恢复 local-cluster 受管集群资源的设置并覆盖新 hub 集群中的 local-cluster 数据时，设置会被错误配置。上一个 hub 集群 local-cluster 的内容没有备份，因为资源包含 local-cluster 特定信息，如集群 URL 详情。

您必须在恢复集群中手动应用与 local-cluster 资源相关的配置更改。请参阅 安装备份和恢复 Operator 主题 中的 *准备新的 hub 集群*。

1.3.2.1.7. 恢复的 Hive 受管集群可能无法与新的 hub 集群连接

当您为 Hive 受管集群恢复更改或轮转颁发机构 (CA) 的备份时，受管集群将无法连接到新的 hub 集群。连接会失败，因为此受管集群的 admin kubeconfig secret 通过备份提供，所以不再有效。

您必须在新 hub 集群中手动更新受管集群的恢复的 admin kubeconfig secret。

1.3.2.1.8. 导入的受管集群显示 *Pending Import* 状态

在主 hub 集群上手动导入的受管集群会在被动 hub 集群上恢复激活数据时显示一个 Pending Import 状态。如需更多信息，请参阅使用受管服务帐户连接集群。

1.3.2.1.9. 恢复 hub 集群后，*appliedmanifestwork* 不会被从受管集群中删除

当在新 hub 集群上恢复 hub 集群数据时，appliedmanifestwork 不会从没有固定集群集的应用程序订阅的放置规则的受管集群中删除。

有关不是固定集群集的应用程序订阅，请参阅以下放置规则示例：

```
spec:
  clusterReplicas: 1
  clusterSelector:
    matchLabels:
      environment: dev
```

因此，当受管集群从恢复的 hub 集群分离时，应用程序会被孤立。

要避免这个问题，请在放置规则中指定固定的集群集。请参见以下示例：

```
spec:
  clusterSelector:
    matchLabels:
      environment: dev
```

您还可以通过运行以下命令来手动删除剩余的 appliedmanifestwork ：

```
oc delete appliedmanifestwork <the-left-appliedmanifestwork-name>
```

1.3.2.1.10. *应用的manifestwork* 不会被删除，规格中缺少 *agentID*

当您将 Red Hat Advanced Cluster Management 2.6 用作主 hub 集群时，但您的恢复 hub 集群位于 2.7 或更高版本的版本时，appliedmanifestworks 规格中缺少 agentID，因为此字段在 2.7 发行版本中引入。这会为受管集群上的主 hub 生成额外的 appliedmanifestworks。

要避免这个问题，请将主 hub 集群升级到 Red Hat Advanced Cluster Management 2.7，然后在新的 hub 集群中恢复备份。

通过为每个 appliedmanifestwork 手动设置 spec.agentID 来修复受管集群。

1. 运行以下命令来获取 agentID ：

```
oc get klusterlet klusterlet -o jsonpath='{.metadata.uid}'
```

2. 运行以下命令，为每个 appliedmanifestwork 设置 spec.agentID ：

```
oc patch appliedmanifestwork <appliedmanifestwork_name> --type=merge -p '{"spec":
{"agentID": "'$AGENT_ID'"}}'
```

1.3.2.1.11. *managed-serviceaccount* add-on 状态显示 *Unknown*

如果您使用 Managed Service Account，则受管集群 appliedmanifestwork addon-managed-serviceaccount-deploy 会从导入的受管集群中删除，而无需在新 hub 集群的 multicluster engine for Kubernetes operator 资源中启用它。

受管集群仍然导入到新的 hub 集群，但 managed-serviceaccount add-on 状态显示 Unknown。

在 multicluster engine operator 资源中启用 Managed Service Account 后，您可以恢复 managed-serviceaccount 附加组件。请参阅启用自动导入以了解如何启用受管服务帐户。

1.3.3. 已知的控制台问题

查看控制台的已知问题。以下列表包含本发行版本的已知问题，或从上一版本中继承的问题。

对于 Red Hat OpenShift Container Platform 集群，请参阅 OpenShift Container Platform 已知问题。

有关弃用和删除的更多信息，请参阅弃用和删除。

1.3.3.1. 无法在控制台中升级 OpenShift Dedicated

从控制台中，您可以请求 OpenShift Dedicated 集群的升级，但升级会失败，并显示 Cannot upgrade non openshift cluster 错误信息。目前没有临时解决方案。

1.3.3.2. 搜索 PostgreSQL pod 处于 CrashLoopBackoff 状态

search-postgres pod 处于 CrashLoopBackoff 状态。如果 Red Hat Advanced Cluster Management 部署到启用了 hugepages 参数的节点，并且 search-postgres pod 调度到这些节点中，则 pod 不会启动。

完成以下步骤以增加 search-postgres pod 的内存：

1. 使用以下命令暂停 search-operator pod：

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

2. 使用对 hugepages 参数的限制更新 search-postgres 部署。运行以下命令，将 hugepages 参数设置为 512Mi ：

```
oc patch deployment search-postgres --type json -p '[{"op": "add", "path": "/spec/template/spec/containers/0/resources/limits/hugepages-2Mi", "value":"512Mi"}]'
```

3. 在验证 pod 的内存用量前，请确保您的 search-postgres pod 处于 Running 状态。运行以下命令：

```
oc get pod <your-postgres-pod-name>  -o jsonpath="Status: {.status.phase}"
```

4. 运行以下命令，以验证 search-postgres pod 的内存用量：

```
oc get pod <your-postgres-pod-name> -o jsonpath='{.spec.containers[0].resources.limits.hugepages-2Mi}'
```

出现以下值 ：512Mi。

1.3.3.3. 无法编辑集群集的命名空间绑定

当使用 admin 角色或 bind 角色编辑集群集的命名空间绑定时，您可能会遇到类似以下消息的错误：

ResourceError: managedclustersetbindings.cluster.open-cluster-management.io "<cluster-set>" is forbidden: User "<user>" cannot create/delete resource "managedclustersetbindings" in API group "cluster.open-cluster-management.io" in the namespace "<namespace>".

要解决这个问题，请确保还有权在您要绑定的命名空间中创建或删除 ManagedClusterSetBinding 资源。角色绑定只允许将集群集绑定到命名空间。

1.3.3.4. 在置备托管的 control plane 集群后，水平滚动无法正常工作

置备托管的 control plane 集群后，如果 ClusterVersionUpgradeable 参数太长，您可能无法在 Red Hat Advanced Cluster Management 控制台的集群概述中水平滚动。因此，您无法查看隐藏的数据。

要临时解决这个问题，请使用浏览器缩放控制来缩放，增加 Red Hat Advanced Cluster Management 控制台窗口大小，或者复制文本并将其粘贴到不同的位置。

1.3.3.5. *EditApplicationSet* 扩展功能重复

当您添加多个标签表达式或尝试为 ApplicationSet 输入集群选择器时，您可能会重复收到以下信息，"Expand to enter expression"。尽管出现这个问题，您可以输入集群选择。

1.3.4. 已知的与应用程序相关的问题和限制

检查应用程序管理的已知问题。以下列表包含本发行版本的已知问题，或从上一版本中继承的问题。

对于 Red Hat OpenShift Container Platform 集群，请参阅 OpenShift Container Platform 已知问题。

有关弃用和删除的更多信息，请参阅弃用和删除。

请参阅以下对应用程序生命周期组件的已知问题。

1.3.4.1. 订阅部署的 OpenShift Container Platform 3.11 的应用程序拓扑错误

注： Red Hat Advanced Cluster Management 支持（OpenShift Container Platform 3.11 已被弃用。

创建以 OpenShift Container Platform 3.11 集群为目标的订阅应用程序后，应用程序拓扑会显示错误用于 ReplicaSet 和 Pod 资源，因为 Kubernetes 中的一个缺陷。这个缺陷是 pod-template-hash 与 ReplicaSet 或 Pod 资源名称中的哈希不匹配的位置。更新的 Kubernetes 版本已被修正，但 OpenShift Container Platform 3.11 不会被修复。详情请参阅 Kubernetes 错误参考。

由于这个程序错误，拓扑可能无法反映资源的状态。例如，pod 和 replicaset 不会反映，但这些资源存在。

- **请参阅以下受管集群命令和 pod 的输出：**

```
oc get pod -n test-helloworld
```

```
NAME                                READY   STATUS    RESTARTS   AGE
helloworld-app-deploy-596765ff66-ndrv8   1/1     Running   0          20m
```

- **请参阅以下受管集群命令和 replicaset 的输出：**

```
oc get replicaset -n test-helloworld
```

```
NAME                             DESIRED   CURRENT   READY   AGE
helloworld-app-deploy-596765ff66   1         1         1       20m
```

1.3.4.2. OpenShift Container Platform 3.11 受管集群缺少应用程序 Kubernetes Lease API

应用程序附加组件组件使用 *Kubernetes Lease API* ,leases.coordination.k8s.io，OpenShift Container Platform 3.11 用户缺少这个组件。Kubernetes 1.14 中引入了 Kubernetes Lease API，但 OpenShift Container Platform 3.11 捆绑包 Kubernetes 版本 1.11。

要解决这个问题，手动将以下 Kubernetes Lease API CustomResourceDefinition 应用到 OpenShift Container Platform 3.11 受管集群：

```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: leases.coordination.k8s.io
```

```
spec:
  group: coordination.k8s.io
  names:
    kind: Lease
    listKind: LeaseList
    plural: leases
    singular: lease
    shortNames:
    - ls
  scope: Namespaced
  versions:
  - name: v1
    served: true    storage: true    schema:
      openAPIV3Schema:
        description: Lease defines a lease concept.
        type: object
        properties:
          apiVersion:
            type: string
          kind:
            type: string
          metadata:
            type: object
          spec:
            type: object
            properties:
              acquireTime:
                format: date-time
                type: string
              holderIdentity:
                type: string
              leaseDurationSeconds:
                format: int64
                type: integer
              leaseTransitions:
                format: int64
                type: integer
              renewTime:
                format: date-time
                type: string
            required:
            - holderIdentity
            - leaseDurationSeconds
            - renewTime
        required:
        - kind
        - metadata
        - spec
  additionalPrinterColumns:
  - JSONPath: .metadata.creationTimestamp
    name: Age
    type: date
  subresources:
    status: {}
```

注： Red Hat Advanced Cluster Management 支持（OpenShift Container Platform 3.11 已被弃用。

1.3.4.3. 服务帐户没有自动 secret

当您在 Red Hat OpenShift Container Platform 4.15 中创建服务帐户时，一些云供应商（如 IBM VMware 和 Bare Metal）置备的服务帐户时，帐户不会自动创建 secret。因此，Red Hat Advanced Cluster Management gitopsCluster 控制器无法为 Argo CD push 模型生成受管集群 secret。

此问题不会在 AWS 置备的 Red Hat OpenShift Container Platform 4.15 中发生。但是，这个问题可能会在由其他云供应商置备的 Red Hat OpenShift Container Platform 4.15 中发生。这个问题在 Red Hat Advanced Cluster Management 2.10.3 中以及 Red Hat Advanced Cluster Management 2.9.4 中提供。

要解决这个问题，您必须手动创建一个 secret，并将其附加到服务帐户 open-cluster-management-agent-addon/application-manager。要做到这一点，请完成以下步骤：

1. 登录到受管集群。

2. 运行以下 secret 模板来创建 secret：

```
apiVersion: v1
kind: Secret
metadata:
  name: application-manager-dockercfg
  namespace: open-cluster-management-agent-addon
  annotations:
    kubernetes.io/service-account.name: application-manager
    openshift.io/token-secret.name: application-manager-dockercfg
    openshift.io/token-secret.value: application-manager-dockercfg
type: kubernetes.io/service-account-token
```

3. 运行以下命令，从创建的 secret 检索令牌：

```
% oc get secrets -n open-cluster-management-agent-addon application-manager-dockercfg -o yaml
data:
  token: <token1>
```

4. 运行以下命令来解码 data.token ：

```
echo <token1 copied from data.token> |base64 -d
```

5. 运行以下命令，将令牌更新至创建的 secret 注解：

```
% oc edit secrets -n open-cluster-management-agent-addon application-manager-
dockercfg
metadata:
  annotations:
    openshift.io/token-secret.value: <paste the decoded token>
```

6. 运行以下命令，将修改后的 secret 链接到您的服务帐户：

```
% oc edit sa -n open-cluster-management-agent-addon application-manager
....
secrets:
- name: application-manager-dockercfg
```

要验证您是否已成功创建了 secret 并将其附加到服务帐户，请完成以下步骤：

1. 进入 hub 集群中的集群命名空间。

2. 运行以下命令验证集群 secret 是否生成：

```
% oc get secrets -n perf5 perf5-cluster-secret
NAME                TYPE    DATA   AGE
perf5-cluster-secret   Opaque   3     7m40s
```

1.3.4.4. 使用 *PlacementRule* 编辑订阅应用程序不会在编辑器中显示订阅 YAML

创建引用 PlacementRule 资源的订阅应用程序后，控制台的 YAML 不会在 YAML 编辑器中显示。使用您的终端编辑订阅 YAML 文件。

1.3.4.5. 带有 secret 依赖项的 Helm Chart 无法由 Red Hat Advanced Cluster Management 订阅部署

使用 Helm Chart，您可以在 Kubernetes secret 中定义隐私数据，并在 Helm Chart 的 value.yaml 文件中引用此 secret。

用户名和密码由引用的 Kubernetes secret 资源 dbsecret 提供。例如，请参阅以下 value.yaml 文件示例：

```
credentials:
  secretName: dbsecret
  usernameSecretKey: username
  passwordSecretKey: password
```

带有 secret 依赖项的 Helm Chart 只在 Helm 二进制 CLI 中被支持。operator SDK Helm 库不支持它。Red Hat Advanced Cluster Management 订阅控制器应用 operator SDK Helm 库来安装和升级 Helm Chart。因此，Red Hat Advanced Cluster Management 订阅无法使用 secret 依赖项部署 Helm Chart。

1.3.4.6. 不支持为 Argo CD Push 模型创建集群 secret

无法为 OpenShift Container Platform 3.11 受管集群上的 Argo CD Push 模型创建自定义集群 secret。这是因为 OpenShift Container Platform 3.11 受管集群不支持受管服务帐户附加组件。

1.3.4.7. 拓扑无法正确显示 Argo CD pull 模型 ApplicationSet 应用程序

当您使用 Argo CD pull 模型来部署 ApplicationSet 应用程序，且应用程序资源名称会被自定义时，每个集群的资源名称可能会有所不同。当发生这种情况时，拓扑无法正确显示您的应用程序。

1.3.4.8. 本地集群被排除为拉取模型的受管集群

hub 集群应用程序集部署到目标受管集群，但本地集群（一个受管 hub 集群）作为目标受管集群排除。

因此，如果 Argo CD 应用程序由 Argo CD pull 模型传播到本地集群，则不会清理本地集群 Argo CD 应用程序，即使本地集群已从 Argo CD ApplicationSet 资源的放置决定中删除。

要临时解决这个问题并清理本地集群 Argo CD 应用程序，请从本地集群 Argo CD 应用程序中删除 skip-reconcile 注解。请参阅以下注解：

```
annotations:
   argocd.argoproj.io/skip-reconcile: "true"
```

另外，如果您在 Argo CD 控制台的 Applications 部分手动刷新 pull model Argo CD 应用程序，则不会处理刷新，Argo CD 控制台中的 REFRESH 按钮被禁用。

要临时解决这个问题，请从 Argo CD 应用程序中删除 refresh 注解。请参阅以下注解：

```
annotations:
   argocd.argoproj.io/refresh: normal
```

1.3.4.9. Argo CD 控制器和传播控制器可能会同时协调

Argo CD 控制器和传播控制器可能会在同一应用程序资源上协调，并导致受管集群中应用程序部署的重复实例，但来自不同部署模型。

对于使用 pull 模型部署应用程序，当 Argo CD argocd.argoproj.io/skip-reconcile 注解添加到 ApplicationSet 的 template 部分时，Argo CD 控制器会忽略这些应用程序资源。

argocd.argoproj.io/skip-reconcile 注解仅适用于 GitOps operator 版本 1.9.0 或更高版本。为防止冲突，请等待 hub 集群和所有受管集群都升级到 GitOps operator 版本 1.9.0，然后再实施 pull 模型。

1.3.4.10. 资源无法部署

MulticlusterApplicationSetReport 中列出的所有资源实际上都部署到受管集群中。如果资源无法部署，则资源不包含在资源列表中，但原因会在错误消息中列出。

1.3.4.11. 资源分配可能需要几分钟时间

对于超过 1000 个受管集群和 Argo CD 应用程序集的大型环境，部署到数百个受管集群，hub 集群上的 Argo CD 应用程序创建可能需要几分钟时间。您可以在应用程序集的 clusterDecisionResource 生成器中将 requeueAfterSeconds 设置为 zero，如下例所示：

```
apiVersion: argoproj.io/v1alpha1
kind: ApplicationSet
metadata:
  name: cm-allclusters-app-set
  namespace: openshift-gitops
spec:
  generators:
  - clusterDecisionResource:
      configMapRef: ocm-placement-generator
      labelSelector:
        matchLabels:
          cluster.open-cluster-management.io/placement: app-placement
      requeueAfterSeconds: 0
```

1.3.4.12. 应用程序 ObjectBucket 频道类型无法使用 allow 和 deny 列表

您不能在 subscription-admin 角色中使用 ObjectBucket 频道类型指定 allow 和 deny 列表。在其他频道类型中，订阅中的 allow 和 deny 列表表示可以部署哪些 Kubernetes 资源，以及不应部署哪些 Kubernetes 资源。

1.3.4.12.1. Argo Application 无法部署到 3.x OpenShift Container Platform 受管集群

控制台中的 Argo ApplicationSet 无法部署到 3.x OpenShift Container Platform 受管集群，因为 Infrastructure.config.openshift.io API 在 3.x 上不可用。

1.3.4.13. 对 multicluster_operators_subscription 镜像的更改不会自动生效

在受管集群中运行的 application-manager 附加组件现在由 subscription operator 处理，后者之前由 klusterlet operator 处理。订阅 operator 没有管理 multicluster-hub，因此对 multicluster-hub 镜像清单 ConfigMap 中的 multicluster_operators_subscription 镜像的更改不会自动生效。

如果订阅 operator 使用的镜像通过更改 multicluster-hub 镜像清单 ConfigMap 中的 multicluster_operators_subscription 镜像覆盖，则受管集群中的 application-manager add-on 不会使用新镜像，直到订阅 operator pod 重启为止。您需要重启 pod。

1.3.4.14. 除非根据订阅管理员部署策略资源

对于 Red Hat Advanced Cluster Management 版本 2.4，默认情况下，policy.open-cluster-management.io/v1 资源不再被应用程序订阅部署。

订阅管理员需要部署应用程序订阅以更改此默认行为。

如需更多信息，请参阅以订阅管理员身份创建允许和拒绝列表。在之前的 Red Hat Advanced Cluster Management 版本中，由现有应用程序订阅部署的 policy.open-cluster-management.io/v1 资源仍然保留，除非应用程序订阅由订阅管理员部署。

1.3.4.15. 应用程序 Ansible hook 独立模式

不支持 Ansible hook 独立模式。要使用订阅在 hub 集群上部署 Ansible hook，您可以使用以下订阅 YAML：

```
apiVersion: apps.open-cluster-management.io/v1
kind: Subscription
metadata:
  name: sub-rhacm-gitops-demo
```

```
  namespace: hello-openshift
annotations:
  apps.open-cluster-management.io/github-path: myapp
  apps.open-cluster-management.io/github-branch: master
spec:
  hooksecretref:
     name: toweraccess
  channel: rhacm-gitops-demo/ch-rhacm-gitops-demo
  placement:
    local: true
```

但是，此配置可能永远不会创建 Ansible 实例，因为 spec.placement.local:true 有以 standalone 模式运行的订阅。您需要在 hub 模式中创建订阅。

1. 创建部署到 local-cluster 的放置规则。请参阅以下示例，其中 local-cluster: "true" 代表 hub 集群：

```
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: <towhichcluster>
  namespace: hello-openshift
spec:
  clusterSelector:
    matchLabels:
      local-cluster: "true"
```

2. 在您的订阅中引用该放置规则。请参见以下示例：

```
apiVersion: apps.open-cluster-management.io/v1
kind: Subscription
metadata:
  name: sub-rhacm-gitops-demo
  namespace: hello-openshift
annotations:
  apps.open-cluster-management.io/github-path: myapp
  apps.open-cluster-management.io/github-branch: master
spec:
  hooksecretref:
     name: toweraccess
  channel: rhacm-gitops-demo/ch-rhacm-gitops-demo
  placement:
    placementRef:
      name: <towhichcluster>
      kind: PlacementRule
```

应用两者后，您应该看到 hub 集群中创建的 Ansible 实例。

1.3.4.16. 在更新的放置规则后没有部署应用程序

如果应用程序在更新放置规则后没有部署，请验证 application-manager pod 是否正在运行。application-manager 是需要在受管集群上运行的订阅容器。

您可以运行 oc get pods -n open-cluster-management-agent-addon |grep application-manager 来验证。

您还可以在控制台中搜索 kind:pod cluster:yourcluster 来查看 application-manager 是否在运行。

如果无法验证，请尝试再次导入集群并重新验证。

1.3.4.17. Subscription operator 不会创建一个 SCC

如需了解更多与 Red Hat OpenShift Container Platform SCC 相关的信息，请参阅 管理 Security Context Constraints (SCC)。它是受管集群所需的一个额外的配置。

不同的部署有不同的安全性上下文和不同的服务帐户。订阅 operator 无法自动创建 SCC CR。pod 的管理员控制权限。需要一个安全性上下文约束（SCC）CR，以便为相关服务帐户启用适当的权限，以便在非默认命名空间中创建 pod。要手动在命名空间中创建 SCC CR，完成以下操作：

1. 找到在部署中定义的服务帐户。例如，查看以下 nginx 部署：

```
nginx-ingress-52edb
nginx-ingress-52edb-backend
```

2. 在命名空间中创建 SCC CR 为服务帐户或帐户分配所需的权限。请参见以下示例，其中添加了 kind: SecurityContextConstraints ：

```
apiVersion: security.openshift.io/v1
 defaultAddCapabilities:
 kind: SecurityContextConstraints
 metadata:
   name: ingress-nginx
   namespace: ns-sub-1
 priority: null
 readOnlyRootFilesystem: false
 requiredDropCapabilities:
 fsGroup:
   type: RunAsAny
```

```
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
users:
- system:serviceaccount:my-operator:nginx-ingress-52edb
- system:serviceaccount:my-operator:nginx-ingress-52edb-backend
```

1.3.4.18. 应用程序频道需要唯一的命名空间

在同一命名空间中创建多个频道可能会导致 hub 集群出现错误。

例如，安装程序将命名空间 charts-v1 作为 Helm 类型频道使用，因此不要在 charts-v1 中创建任何其他频道。确保您在唯一命名空间中创建频道。所有频道需要单独的命名空间，但 GitHub 频道除外，它们可与另一个 GitHub 频道共享命名空间。

1.3.4.19. Ansible Automation Platform 作业失败

当您选择不兼容的选项时，Ansible 作业无法运行。只有选择了 -cluster 范围内的频道选项时，Ansible Automation Platform 才起作用。这会影响需要执行 Ansible 作业的所有组件。

1.3.4.20. Ansible Automation Platform operator 在代理外访问 Ansible Automation Platform

Red Hat Ansible Automation Platform Operator 无法访问启用了代理的 OpenShift Container Platform 集群之外的 Ansible Automation Platform。要解决这个问题，您可以在代理中安装 Ansible Automation Platform。请参阅 Ansible Automation Platform 提供的安装步骤。

1.3.4.21. 应用程序名称要求

应用程序名称不能超过 37 个字符。如果字符超过这个数量，应用部署将显示以下错误。

```
status:
  phase: PropagationFailed
  reason: 'Deployable.apps.open-cluster-management.io "_long_lengthy_name_" is invalid:
metadata.labels: Invalid value: "_long_lengthy_name_": must be no more than 63
characters/n'
```

1.3.4.22. 应用程序控制台表限制

参阅控制台中不同 *Application* 表的限制：

- 在 *Overview* 页面的 *Applications* 表和 *Advanced 配置*页面上的 *Subscriptions* 表中，*Clusters* 列会显示部署应用程序资源的集群计数。因为应用程序是由本地集群上的资源定义的，所以本地集群会包含在搜索结果中，无论实际的应用程序资源是否在本地集群中部署。

- 在 *Subscriptions* 的 *Advanced configuration* 列表中，*Applications* 栏显示使用该订阅的应用程序总数，如果订阅部署了子应用程序，它们也会包含在搜索结果中。

- *Channels* 的 *Advanced configuration* 列表中，*Subscriptions* 栏显示使用该频道的本地集群中的订阅总数，但这不包括由其他订阅部署的订阅，这些订阅包含在搜索结果中。

1.3.4.23. 没有应用程序控制台拓扑过滤

2.10 的应用程序的 *Console* 和 *Topology* 已更改。*控制台 Topology 页面中没有过滤功能。*

1.3.4.24. 允许和拒绝列表在对象存储应用程序中无法正常工作

允许和决绝列表功能无法在对象存储应用程序订阅中工作。

1.3.5. 已知的可观察性问题

查看 Red Hat Advanced Cluster Management for Kubernetes 中的已知问题。以下列表包含本发行版本的已知问题，或从上一版本中继承的问题。

对于 Red Hat OpenShift Container Platform 集群，请参阅 OpenShift Container Platform 已知问题。

有关弃用和删除的更多信息，请参阅弃用和删除。

1.3.5.1. 配置默认值

MultiClusterObservability 使用早期版本的 Cluster Monitoring Operator。之前的版本可防止 cluster-monitoring-config 配置映射更新至可观察性中的其他配置。然后，配置会重置为默认值。

1.3.5.2. 恢复的 hub 集群中的 observatorium API 网关 pod 可能具有过时的租户数据

恢复的 hub 集群中的 Observatorium API 网关 pod 在备份和恢复过程后可能包含过时的租户数据，

因为 Kubernetes 限制。有关限制的更多信息，请参阅挂载 ConfigMap 自动更新。

因此，Observatorium API 和 Thanos 网关拒绝收集器的指标，Red Hat Advanced Cluster Management Grafana 仪表板不会显示数据。

请参阅 Observatorium API 网关 pod 日志中的以下错误：

```
level=error name=observatorium caller=logchannel.go:129 msg="failed to forward metrics" returncode="500 Internal Server Error" response="no matching hashring to handle tenant\n"
```

Thanos 接收带有以下错误的 pod 日志：

```
caller=handler.go:551 level=error component=receive component=receive-handler tenant=xxxx err="no matching hashring to handle tenant" msg="internal server error"
```

请参阅以下流程来解决这个问题：

1. 将 observability-observatorium-api 部署实例从 N 缩减到 0。
2. 将 observability-observatorium-api 部署实例从 0 扩展到 N。

注意：默认情况下 N = 2，但在某些自定义配置环境中可能大于 2。

这会重启所有 Observatorium API 网关 pod，收集器中的数据会在 5-10 分钟之间显示在 Grafana 中。

1.3.5.3. 在 openshift-monitoring 命名空间中添加 PrometheusRules 和 ServiceMonitor 的权限

从 Red Hat Advanced Cluster Management 2.9 开始，您必须使用定义的 Red Hat Advanced Cluster Management hub 集群命名空间中的标签。标签 openshift.io/cluster-monitoring: "true" 会导致 Cluster Monitoring Operator 提取指标的命名空间。

当部署 Red Hat Advanced Cluster Management 2.9 或安装升级到 2.9 时，Red Hat Advanced Cluster Management Observability ServiceMonitor 和 PrometheusRule 资源不再存在于 openshift-

monitoring 命名空间中。

1.3.5.4. 缺少对代理设置的支持

observability 附加组件的 Prometheus AdditionalAlertManagerConfig 资源不支持代理设置。您必须禁用 observability 警报转发功能。

完成以下步骤以禁用警报转发：

1. *进入 MultiClusterObservability 资源。*
2. *将 mco-disabling-alerting 参数值更新为 true*

不支持使用自签名 CA 证书的 HTTPS 代理。

1.3.5.5. Service-level Overview 仪表板上重复的 local-clusters

当各种 hub 集群使用相同的 S3 存储部署 Red Hat Advanced Cluster Management observability 时，可以在 Kubernetes/Service-Level Overview/API Server 仪表板中检测并显示重复的 local-clusters。重复的集群在以下面板中影响结果： Top Clusters、超过 SLO 的集群数，以及满足 SLO 的集群数量。local-clusters 是与共享 S3 存储关联的唯一集群。要防止多个 local-clusters 显示在仪表板中，建议每个唯一的 hub 集群使用针对 hub 集群的 S3 存储桶来部署可观察性。

1.3.5.6. Observability endpoint operator 无法拉取镜像

如果您创建一个 pull-secret 用于部署到 MultiClusterObservability CustomResource（CR），且 open-cluster-management-observability 命名空间中没有 pull-secret，则 observability endpoint operator 会失败。当您导入新集群或导入使用 Red Hat Advanced Cluster Management 创建的 Hive 集群时，需要在受管集群上手动创建 pull-image secret。

如需更多信息，请参阅启用可观察性。

1.3.5.7. 没有来自 ROKS 集群的数据

Red Hat Advanced Cluster Management observability 不会在内置仪表板中显示 ROKS 集群中的数据。这是因为 ROKS 不会从它们管理的服务器公开任何 API 服务器指标。以下 Grafana 仪表板包含不

支持 ROKS 集群的面板：Kubernetes/API server、Kubernetes/Compute Resources/Workload、Kubernetes/Compute Resources/Namespace(Workload)

1.3.5.8. ROKS 集群没有 etcd 数据

对于 ROKS 集群，Red Hat Advanced Cluster Management observability 不会在仪表板的 etcd 面板中显示数据。

1.3.5.9. Grafana 控制台中没有指标数据

- 注解查询在 Grafana 控制台中会失败：

  当在 Grafana 控制台中搜索特定注解时，您可能会因为已过期的令牌收到以下错误消息：

  "Annotation Query Failed"

  重新刷新浏览器，验证您是否已登录到 hub 集群。

- rbac-query-proxy pod 中的错误：

  由于未授权访问 managedcluster 资源，您可能会在查询集群或项目时收到以下错误：

  no project or cluster found

  检查角色权限并进行相应的更新。如需更多信息，请参阅基于角色的访问控制。

1.3.5.10. 受管集群上的 Prometheus 数据丢失

默认情况下，OpenShift 上的 Prometheus 使用临时存储。Prometheus 会在重启时丢失所有指标数据。

如果在由 Red Hat Advanced Cluster Management 管理的 OpenShift Container Platform 受管集群上启用或禁用了可观察性，observability 端点 Operator 会添加额外的 alertmanager 配置来自动重启本地 Prometheus，以此更新 cluster-monitoring-config ConfigMap。

1.3.5.11. Error ingesting out-of-order samples

Observability receive pod 报告以下出错信息：

```
Error on ingesting out-of-order samples
```

错误消息表示，在指标收集间隔期间，由受管集群发送的时间序列数据比在之前的集合间隔发送的时间序列数据旧。当出现这个问题时，Thanos 接收器会丢弃数据，这可能会在 Grafana 仪表板中显示的数据中造成差距。如果经常看到这个错误，建议将指标收集间隔增加到一个更高的值。例如，您可以将间隔增加到 60 秒。

只有在时间序列间隔被设置为较低值（如 30 秒）时，才会注意到这个问题。请注意，当指标收集间隔被设置为默认值 300 秒时，不会看到这个问题。

1.3.5.12. 升级后 Grafana 部署失败

如果您在 2.6 之前的系统中部署了 grafana-dev 实例，并将环境升级到 2.6，grafana-dev 无法正常工作。您必须运行以下命令来删除现有 grafana-dev 实例：

```
./setup-grafana-dev.sh --clean
```

使用以下命令重新创建实例：

```
./setup-grafana-dev.sh --deploy
```

1.3.5.13. klusterlet-addon-search pod 失败

klusterlet-addon-search pod 失败，因为达到内存限制。您必须通过自定义受管集群中的 klusterlet-addon-search 部署来更新内存请求和限制。在 hub 集群中编辑名为 search-collector 的 ManagedclusterAddon 自定义资源。在 search-collector 中添加以下注解并更新内存 addon.open-cluster-management.io/search_memory_request=512Mi 和 addon.open-cluster-management.io/search_memory_limit=1024Mi。

例如，如果您有一个名为 foobar 的受管集群，请运行以下命令将内存请求更改为 512Mi，内存限值为 1024Mi ：

```
oc annotate managedclusteraddon search-collector -n foobar \
addon.open-cluster-management.io/search_memory_request=512Mi \
addon.open-cluster-management.io/search_memory_limit=1024Mi
```

1.3.5.14. 启用 disableHubSelfManagement 在 Grafana 仪表板中会导致空列表

如果在 mulitclusterengine 自定义资源中将 disableHubSelfManagement 参数设置为 true 时，Grafana 仪表板会显示一个空标签列表。您必须将参数设置为 false 或删除参数来查看标签列表。如需了解更多详细信息，请参阅 disableHubSelfManagement。

1.3.5.14.1. 端点 URL 无法具有完全限定域名 (FQDN)

当您将 FQDN 或协议用于 endpoint 参数时，您的可观察性 pod 不会被启用。此时会显示以下出错信息：

> Endpoint url cannot have fully qualified paths

输入没有协议部分的 URL。您的 endpoint 值必须类似您的 secret 的以下 URL：

> endpoint: example.com:443

1.3.5.14.2. Grafana downsampled 数据不匹配

当您尝试查询历史数据时，计算的步骤值和 downsampled 数据之间存在差异，结果为空。例如，如果计算的步骤值为 5m，并且 downsampled 数据处于一小时的间隔，则数据不会出现在 Grafana 中。

此差异发生，因为 URL 查询参数必须通过 Thanos Query 前端数据源进行传递。之后，当数据缺失时，URL 查询可以对其他降级级别执行额外的查询。

您必须手动更新 Thanos Query 前端数据源配置。完成以下步骤：

1. 进入 Query 前端数据源。
2. 要更新您的查询参数，请点击 Misc 部分。
3. 在 Custom query parameters 字段中，选择 max_source_resolution=auto。
4. 要验证是否显示数据，请刷新 Grafana 页面。

您的查询数据会出现在 Grafana 仪表板中。

1.3.5.15. 指标收集器不会检测代理配置

指标收集器不会检测到您使用 addonDeploymentConfig 配置的受管集群中的代理配置。作为临时解决方案，您可以通过删除受管集群清单工作来启用 代理。删除 ManifestWork 会强制应用 addonDeploymentConfig 中的更改。

1.3.5.16. 不支持使用自定义 CA 捆绑包的 HTTPS 代理

当需要自定义 CA 捆绑包时，受管集群中的代理配置无法正常工作。

1.3.6. 已知的监管问题

查看监管的已知问题。以下列表包含本发行版本的已知问题，或从上一版本中继承的问题。

对于 Red Hat OpenShift Container Platform 集群，请参阅 OpenShift Container Platform 已知问题。

有关弃用和删除的更多信息，请参阅弃用和删除。

1.3.6.1. OpenShift Container Platform 3.11 不提供容器安全 Operator。

OpenShift Container Platform 3.11 不提供容器安全 Operator。因此，您无法在 ImageManifestVuln 策略中使用 policy-imagemanifestvuln-sub 的策略模板，并将其应用到 OpenShift Container Platform 3.11 集群。

如果您尝试应用 ImageManifestVuln 策略，您会收到以下违反消息：

> violation - couldn't find mapping resource with kind Subscription, please check if you have CRD deployed.

1.3.6.2. 当组件被禁用时，不会正确清理监管资源

监管资源不会被正确清理。当组件被设置为 false 或在 MultiClusterHub operator 中被禁用时，会在可以清理它管理的附加组件前删除监管组件。

1.3.6.3. 无法从 Red Hat Advanced Cluster Management 注销

当您使用外部身份提供程序登录到 Red Hat Advanced Cluster Management 时，您可能无法从 Red Hat Advanced Cluster Management 注销。当您使用与 IBM Cloud 和 Keycloak 作为身份提供程序一起安装的 Red Hat Advanced Cluster Management 时会出现这种情况。

在尝试从 Red Hat Advanced Cluster Management 注销前，您必须从外部身份提供程序注销。

1.3.6.4. 当命名空间处于 Terminating 状态时，配置策略列出了 complaint

当您有一个为 complianceType 参数配置且为 remediationAction 参数带有 mustnothave 的配置策略时，当向 Kubernetes API 发出删除请求时，策略会被列为合规。因此，在策略列为合规时，Kubernetes 对象可能会一直处于 Terminating 状态。

1.3.6.5. 使用策略部署的 Operator 不支持 ARM

虽然支持安装到 ARM 环境中，但使用策略部署的 operator 可能不支持 ARM 环境。安装 Operator 的以下策略不支持 ARM 环境：

- Red Hat Advanced Cluster Management for Quay Container Security Operator
- Red Hat Advanced Cluster Management for Compliance Operator

1.3.6.6. ConfigurationPolicy 自定义资源定义处于终止状态

当您通过在 KlusterletAddonConfig 或分离集群中禁用策略控制器从受管集群中删除 config-policy-controller 附加组件时，ConfigurationPolicy 自定义资源定义可能会处于终止状态。如果 ConfigurationPolicy 自定义资源定义处于终止状态，如果稍后重新安装附加组件，则可能不会添加新策略。您还可以收到以下错误：

```
template-error; Failed to create policy template: create not allowed while custom resource definition is terminating
```

使用以下命令检查自定义资源定义是否已卡住：

```
oc get crd configurationpolicies.policy.open-cluster-management.io -o=jsonpath='{.metadata.deletionTimestamp}'
```

如果删除时间戳位于资源中，则自定义资源定义会卡住。要解决这个问题，从集群中保留的配置策略中删除所有终结器。在受管集群中使用以下命令，将 <cluster-namespace> 替换为受管集群命名空间：

```
oc get configurationpolicy -n <cluster-namespace> -o name | xargs oc patch -n <cluster-namespace> --type=merge -p '{"metadata":{"finalizers": []}}'
```

配置策略资源会自动从集群中移除，自定义资源定义会退出其终止状态。如果已经重新安装了附加组件，则在没有删除时间戳的情况下自动重新创建自定义资源定义。

1.3.6.7. 在修改现有配置策略时， pruneObjectBehavior 无法正常工作

当您修改现有配置策略时， pruneObjectBehavior 无法正常工作。查看 pruneObjectBehavior 可能无法正常工作的原因：

- 如果您在配置策略中将 pruneObjectBehavior 设置为 DeleteAll 或 DeleteIfCreated，则不会正确清理修改前创建的旧资源。当您删除配置策略时，只有策略创建和策略更新中的新资源才会被跟踪和删除。

- 如果将 pruneObjectBehavior 设置为 None 或没有设置参数值，则可能会在受管集群上意外删除旧对象。具体来说，当用户更改模板中的 name, namespace, kind, or apiversion 时这会发生。当 object-templates-raw 或 namespaceSelector 参数更改时，参数字段可以动态更改。

1.3.6.8. 强制时策略状态显示重复的更新

如果策略被设置为 remediationAction: enforce 并重复更新， Red Hat Advanced Cluster Management 控制台会显示重复违反情况，并成功更新。请参阅以下可能的原因和错误解决方案：

- 另一个控制器或进程也使用不同的值更新对象。

  要解决这个问题，请禁用策略并比较策略和受管集群上的 objectDefinition 之间的不同。如果值不同，则可能会更新另一个控制器或进程。检查对象的元数据，以帮助识别值的不同原因。

- ConfigurationPolicy 中的 objectDefinition 不匹配，因为 Kubernetes 在应用策略时处理对象。

  要解决这个问题，请禁用策略并比较策略和受管集群上的 objectDefinition 之间的不同。如果键不同或缺失， Kubernetes 可能会在将密钥应用到对象之前处理密钥，如删除包含默认值或

空值的键。

1.3.6.9. OpenShift Container Platform 4.12 及更新的版本不支持 Pod 安全策略

对 Pod 安全策略的支持已从 OpenShift Container Platform 4.12 及更新的版本中删除，并从 Kubernetes v1.25 及之后的版本中删除。如果应用 PodSecurityPolicy 资源，您可能会收到以下不合规的信息：

> violation - couldn't find mapping resource with kind PodSecurityPolicy, please check if you have CRD deployed

1.3.6.10. 重复策略模板名称会创建 inconstistent 结果

当您创建具有相同策略模板名称的策略时，您会收到不一致的结果，但您可能不知道原因。例如，使用名为 create-pod 的多个配置策略定义策略会导致结果不一致。最佳实践：对策略模板避免使用重复名称。

1.3.6.11. 当禁用时，监管部署不会关闭且无错误

当您在 MultiClusterHub 对象中禁用监管部署时，不会在没有错误的情况下清理部署。完成以下步骤以禁用监管，以便同时清理部署：

1. 在受管集群的 KlusterletAddonConfig 中禁用 policyController。如果对所有受管集群执行此操作，请运行以下命令：

```
for CLUSTER in $(oc get managedclusters -o jsonpath='{.items[].metadata.name}'); do
  oc patch -n ${CLUSTER} klusterletaddonconfig ${CLUSTER} --type=merge --
patch='{"spec":{"policyController":{"enabled":false}}}'
done
```

2. 仅限本地集群：删除本地集群的 ManifestWork，并在 CrashLoopBackOffCrashLoopBackOff 中的 governance-policy-framework-uninstall pod 处于 CrashLoopBackOff 时删除 ManagedClusterAddon 上的终结器。运行以下命令：

```
oc delete manifestwork -n local-cluster -l open-cluster-management.io/addon-
name=governance-policy-framework
oc patch managedclusteraddon -n local-cluster governance-policy-framework --
type=merge --patch='{"metadata":{"finalizers":[]}}'
```

3. 如果需要，通过在 MultiClusterHub 对象中将 spec.overrides 部分中的 grc 元素设置为 false 来禁用全局监管。运行以下命令：

```
oc edit multiclusterhub <name> -n <namespace>
```

4. 仅限本地集群：如果有任何本地集群策略，您可以通过运行以下命令来删除策略：

```
oc delete policies -n local-cluster --all
```

5. 要在 KlusterletAddonConfig 中重新启用监管，请重新启用 MultiClusterHub 中的 spec.overrides 部分的 grc 元素。运行以下命令：

```
for CLUSTER in $(oc get managedclusters -o jsonpath='{.items[].metadata.name}'); do
  oc patch -n ${CLUSTER} klusterletaddonconfig ${CLUSTER} --type=merge --
patch='{"spec":{"policyController":{"enabled":true}}}'
done
```

6. 如果部署不成功，则 governance-policy-addon-controller 可能会具有过时的租期。使用以下命令删除租期：

```
oc delete lease governance-policy-addon-controller-lock -n <namespace>
```

1.3.6.12. 数据库和策略合规历史记录 API 中断

对数据库和策略合规历史记录 API 中断提供了内置弹性，但任何无法由受管集群记录的合规性事件都会在内存中排队，直到它们成功记录为止。这意味着，如果受管集群上有中断并且 managed -policy-framework pod 重启，则所有排队的合规性事件都会丢失。

如果您在数据库中断期间创建或更新新策略，则无法记录为这个新策略发送的任何合规事件，因为无法更新策略到数据库 ID 的映射。当数据库恢复在线时，映射会自动更新，并从这些策略中记录将来的合规事件。

1.3.6.13. PostgreSQL 数据丢失

如果数据丢失到 PostgreSQL 服务器，如在没有最新数据的情况下恢复到备份，则必须重启 Red Hat Advanced Cluster Management hub 集群上的监管策略传播器，以便它可以将策略映射到数据库 ID。在重启监管策略传播器前，不再记录与数据库中存在的策略关联的新合规事件。

要重启监管策略传播器，请在 Red Hat Advanced Cluster Management hub 集群中运行以下命令：

```
oc -n open-cluster-management rollout restart deployment/grc-policy-propagator
```

1.3.7. 已知的与网络相关的问题

查看 Submariner 的已知问题。以下列表包含本发行版本的已知问题，或从上一版本中继承的问题。

对于 Red Hat OpenShift Container Platform 集群，请参阅 OpenShift Container Platform 已知问题。

有关弃用和删除的更多信息，请参阅弃用和删除。

1.3.7.1. Submariner 已知问题

请查看以下在使用网络功能时可能会出现的已知问题和限制。

1.3.7.1.1. 没有 ClusterManagementAddon submariner 附加组件失败

对于版本 2.8 及更早版本，在安装 Red Hat Advanced Cluster Management 时，您还可以使用 Operator Lifecycle Manager 部署 submariner-addon 组件。如果您没有创建 MultiClusterHub 自定义资源，submariner-addon pod 会发送错误，并阻止 Operator 安装。

发生以下通知，因为缺少 ClusterManagementAddon 自定义资源定义：

> graceful termination failed, controllers failed with error: the server could not find the requested resource (post clustermanagementaddons.addon.open-cluster-management.io)

ClusterManagementAddon 资源由 cluster-manager 部署创建，但当集群中安装 MultiClusterEngine 组件时，此部署将可用。

如果在创建 MultiClusterHub 自定义资源时没有在集群中可用的 MultiClusterEngine 资源，MultiClusterHub operator 会部署 MultiClusterEngine 实例，以及所需的 Operator，用于解决前面的错误。

1.3.7.1.2. 当导入受管集群时，Submariner 附加组件资源没有正确清理

如果 MultiClusterHub (MCH) operator 中的 submariner-addon 组件被设置为 false，则不会为受管集群资源正确清理 submariner-addon 终结器。因为没有正确清理终结器，这可以防止 submariner-addon 组件在 hub 集群中被禁用。

1.3.7.1.3. 不是 Red Hat Advanced Cluster Management 可以管理的所有基础架构供应商都被支持

在 Red Hat Advanced Cluster Management 的所有基础架构供应商不支持 Submariner。如需支持的供应商列表，请参阅 Red Hat Advanced Cluster Management 支持列表。

1.3.7.1.4. Submariner 安装计划限制

Submariner 安装计划不会遵循整个安装计划设置。因此，Operator 管理屏幕无法控制 Submariner 安装计划。默认情况下，Submariner 安装计划会被自动应用，Submariner addon 总是更新至与已安装的 Red Hat Advanced Cluster Management 版本对应的最新可用版本。要更改此行为，您必须使用自定义 Submariner 订阅。

1.3.7.1.5. 有限的无头服务支持

在使用 Globalnet 时，在没有选择器的情况下的无头服务不支持服务发现。

1.3.7.1.6. 不支持在启用 NAT 时使用 VXLAN 的部署

只有非 NAT 部署支持使用 VXLAN 电缆驱动程序的 Submariner 部署。

1.3.7.1.7. OVN Kubernetes 需要 OCP 4.11 及更新的版本

如果使用 OVN Kubernetes CNI 网络，则需要 Red Hat OpenShift 4.11 或更高版本。

1.3.7.1.8. 自签名证书可能会阻止到代理的连接

代理上的自签名证书可能会阻止加入集群连接到代理。连接失败并显示证书验证错误。您可以通过在相关 SubmarinerConfig 对象中将 InsecureBrokerConnection 设置为 true 来禁用代理证书验证。请参见以下示例：

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  insecureBrokerConnection: true
```

1.3.7.1.9. Submariner 只支持 OpenShift SDN 或 OVN Kubernetes

Submariner 只支持使用 OpenShift SDN 或 OVN-Kubernetes Container Network Interface (CNI)

网络供应商的 Red Hat OpenShift Container Platform 集群。

1.3.7.1.10. Microsoft Azure 集群的命令限制

subctl diagnose firewall inter-cluster 命令无法在 Microsoft Azure 集群中工作。

1.3.7.1.11. 自动升级无法使用自定义 CatalogSource 或 Subscription

当 Red Hat Advanced Cluster Management for Kubernetes 升级时，Submariner 会被自动升级。如果您使用自定义 CatalogSource 或 Subscription，则自动升级可能会失败。

为确保在受管集群上安装 Submariner 时自动升级可以正常工作，您必须在每个受管集群的 SubmarinerConfig 自定义资源中将 spec.subscriptionConfig.channel 字段设置为 stable-0.15。

1.3.7.1.12. Submariner 与启用了 IPsec 的 OVN-Kubernetes 部署冲突

由启用了 IPsec 的 OVN-Kubernetes 部署创建的 IPsec 隧道可能会与 Submariner 创建的 IPsec 隧道冲突。不要在 Submariner 的 IPsec 模式中使用 OVN-Kubernetes。

1.3.7.1.13. 在从 ManageClusterSet中删除 ManagedCluster 前卸载 Submariner

如果您从 ClusterSet 中删除集群，或者将集群移到不同的 ClusterSet，则 Submariner 安装不再有效。

您必须在从 ManageClusterSet 移动或移除 ManagedCluster 前卸载 Submariner。如果没有卸载 Submariner，则无法卸载或重新安装 Submariner，Submariner 会停止在 ManagedCluster 上工作。

1.3.7.1.14. Submariner 安装在带有 OpenShift Container Platform 4.15 及更新版本的 VMware vSphere 上失败

因为 hub 集群上的受管集群缺少代理 secret，所以 Submariner 附加组件在 VMware vSphere 上运行 OpenShift Container Platform 4.15 及更新版本的 hub 集群会失败。只有在受管集群上的 submariner-operator 命名空间中创建 submariner-addon pod，控制台会显示没有标记的网关。

您可以通过为 ClusterSet 代理命名空间中的每个受管集群手动创建 secret 来解决此问题。要手动创建 secret，请完成以下步骤：

1

I. 登录到 hub 集群。

2. 创建 YAML 文件并添加以下模板。根据需要替换值：

```
apiVersion: v1
kind: Secret
metadata:
  name: <ManagedClusterName>-broker
  namespace: <ClustersetName>-broker
  annotations:
    kubernetes.io/service-account.name: <ManagedClusterName>
type: kubernetes.io/service-account-token
```

3. 运行以下命令来应用 YAML 文件：

```
oc apply
```

1.3.8. multicluster global hub Operator 已知问题

查看 multicluster global hub Operator 的已知问题。以下列表包含本发行版本的已知问题，或从上一版本中继承的问题。对于 OpenShift Container Platform 集群，请参阅 OpenShift Container Platform 已知问题。

1.3.8.1. Kafka operator 会保持重启

在联邦信息处理标准(FIPS)环境中，Kafka operator 会因为内存不足(OOM)状态保持重启。要修复此问题，请将资源限值设置为至少 512M。有关如何设置此限制的详细步骤，请参阅 amq 流文档。

1.3.8.2. 备份和恢复已知问题

如果您的原始多集群全局 hub 集群崩溃，则多集群全局 hub 会丢失其生成的事件和 cron 作业。即使恢复新的多集群全局 hub 集群，也不会恢复事件和 cron 作业。要解决这个问题，您可以手动运行 cron 作业，请参阅 手动运行总结过程。

1.3.8.3. 受管集群显示但不计算

没有成功创建的受管集群，这意味着受管集群中不存在 clusterclaim id.k8s.io，不会在策略合规仪表板中计算，但在策略合规仪表板中显示。

1.3.8.4. multicluster global hub 安装在 OpenShift Container Platform 4.13 超链接上，可能会重定向家

如果在 OpenShift Container Platform 4.13 上安装 multicluster global hub Operator，则链接到受管集群列表的所有超链接，仪表板中的详情页面可能会重定向到 Red Hat Advanced Cluster Management 主页。

您需要手动进入目标页面。

1.3.8.5. 标准组过滤器无法传递给新页面

在 Global Hub Policy Group Compliancy Overview hub 仪表板中，您可以通过单击 View Offending Policies for Standard group 来检查一个数据点，但在点此链接进入关闭页面后，标准组过滤器无法传递给新页面。

另外，Cluster Group Compliancy Overview 也是一个问题。

1.3.8.6. 无法重定向到 OpenShift Container Platform 3.11 集群 Observability 页面

如果受管 hub 集群导入 OpenShift Container Platform 3.11 集群（已弃用）作为受管集群，则无法重定向到 Global Hub > Overview 仪表板中的 Observability 页面。

您需要手动导航到目标页面。

1.4. 弃用和删除

了解产品将在什么时候被弃用，或从 Red Hat Advanced Cluster Management for Kubernetes 中删除。考虑推荐操作中的备选操作和详细信息，它们显示在当前版本的表中和之前两个版本。

重要： Red Hat Advanced Cluster Management 的 2.6 和更早的版本 已被删除，并不再被支持。2.6 及更早的版本文档没有更新。其文档可能仍然可用，但不再有任何新的勘误或其他更新。

最佳实践：升级到 Red Hat Advanced Cluster Management 的最新版本。

1.4.1. API 弃用和删除

Red Hat Advanced Cluster Management 的 API 会遵循 Kubernetes 弃用指南。有关相关策略的详情，请参阅 Kubernetes 弃用策略。Red Hat Advanced Cluster Management API 只在以下时间线外才会被弃用或删除：

- *所有 V1 API 已正式发布（GA），提供 12 个月或跨三个发行版本（以更长的时间为准）的支持。V1 API 没有被删除，但可能会在这个时间限制外被弃用。*
- *所有 beta API 通常在九个月或跨三个发行版本（以更长的时间为准）内可用。Beta API 不会在这个时间限制外被删除。*
- *所有 alpha API 都不是必需的，但如果对用户有好处，则可能会被列为已弃用或删除。*

1.4.1.1. API 删除

产品或类别	受影响的项	Version	推荐的操作	详情和链接
ManagedClusterSets	**v1beta1** API 已被删除。	2.9	使用 **v1beta2** 替代。	ManagedClusterSets.cluster.open-cluster-management.io
ManagedClusterSetBindings	**v1beta1** API 已被删除。	2.9	使用 **v1beta2** 替代。	ManagedClusterSetBindings.cluster.open-cluster-management.io
HypershiftDeployment	**HypershiftDeployment** API 已被删除。	2.7	不要使用这个 API。	
BareMetalAssets	**v1alpha1** API 被删除。	2.7	不要使用这个 API。	Baremetalassets.inventory.open-cluster-management.io
放置	**v1alpha1** API 被删除。	2.7	使用 **v1beta1** 替代。	Placements.cluster.open-cluster-management.io
PlacementDecisions	**v1alpha1** API 被删除。	2.7	使用 **v1beta1** 替代。	PlacementDecisions.cluster.open-cluster-management.io

产品或类别	受影响的项	Version	推荐的操作	详情和链接
ClusterManagementAddOn	字段 **addOnConfiguration** 在 **ClusterManagementAddOn** spec 中已弃用。	2.7	使用 **supportedConfigs** 字段。	None
ManagedClusterAddOn	字段 **addOnConfiguration** 在 **ManagedClusterAddOn** spec 中已弃用。	2.7	使用 **supportedConfigs** 字段。	None

1.4.2. Red Hat Advanced Cluster Management 弃用

弃用（deprecated）组件、功能或服务会被支持，但不推荐使用，并可能在以后的版本中被删除。考虑使用推荐操作中的相应的替代操作，详情在下表中提供：

产品或类别	受影响的项	Version	推荐的操作	详情和链接
OpenShift Container Platform 3.11 支持的功能	各种组件	2.9	None	生命周期政策
监管	IAM 策略控制器	2.9	None	
监管	容器安全 Operator	OpenShift Container Platform 3.11	None	请参阅 OpenShift Container Platform 3.11 不提供容器安全Operator

产品或类别	受影响的项	Version	推荐的操作	详情和链接
安装程序	**operator.open-cluster-management.io_multiclusterhubs_crd.yaml** 中的 **ingress.sslCiphers** 字段	2.9	None	请参阅高级配置来配置安装。如果您降级了 Red Hat Advanced Cluster Management for Kubernetes 版本，并且最初有一个带有 **spec.ingress.sslCiphers** 字段的 **MultiClusterHub** 自定义资源，则该字段仍可以被识别，但已被弃用，且没有效果。
应用程序和管理	**PlacementRule**	2.8	在使用 **PlacementRule** 的位置使用 **Placement**。	虽然 **PlacementRule** 仍然可用，但它不被支持，控制台默认会显示 **Placement**。
安装程序	**operator.open-cluster-management.io_multiclusterhubs_crd.yaml** 中的 **customCAConfigmap** 字段	2.7	None	请参阅高级配置来配置安装。

1.4.3. 删除

一个删除（removed）的项通常是在之前的版本中被弃用的功能，在该产品中不再可用。您必须将 alternatives 用于删除的功能。考虑使用推荐操作中的相应的替代操作，详情在下表中提供：

产品或类别	受影响的项	Version	推荐的操作	详情和链接
搜索	**SearchCustomizations.open-cluster-management.io** 自定义资源定义已被删除。	2.7	使用 **search.open-cluster-management.io/v1alpha1** 自定义搜索。	None

产品或类别	受影响的项	Version	推荐的操作	详情和链接
搜索	RedisGraph 被 PostgreSQL 替代作为内部数据库。	2.7	不需要更改。	使用 PostgreSQL 作为内部数据库重新实施搜索组件。
控制台（Console）	独立 Web 控制台	2.7	使用集成的 Web 控制台。	如需更多信息，请参阅访问您的控制台。

1.5. RED HAT ADVANCED CLUSTER MANAGEMENT FOR KUBERNETES 平台针对 GDPR 的注意事项

1.5.1. 备注

本文档旨在帮助您准备 General Data Protection Regulation（GDPR）就绪。它提供有关您可以配置的 Red Hat Advanced Cluster Management for Kubernetes 平台的功能信息，以及产品的使用情况，以满足 GDPR 就绪的要求。因为用户可以选择不同的方式来配置功能，并且产品的使用方式及第三方集群和系统都会有所不同，所以这里介绍的信息可能并没有覆盖所有情况。

客户需要负责确保自己遵守各种法律及条例，包括欧盟的 GDPR 条例。获取法律法规建议，确定并解释可能影响客户业务的相关法律及规范，以及客户可能需要为遵守此类法律及规范而可能需要执行的任何行动完全由客户自己负责。

这里描述的产品、服务和其他功能不适用于所有客户情况，且适用性可能有限制。红帽不提供法律、会计、审计方面的建议，也不代表或者认为其服务或产品会确保客户遵守任何法律和规范。

1.5.2. 内容表

- 数据存储
- 数据访问
- 数据处理
- 数据删除
- 限制使用个人数据的能力
- 附录

1.5.3. GDPR

欧盟（"EU"）已采用了 General Data Protection Regulation（GDPR）并从 2018 年 5 月 25 日起生效。

1.5.3.1. 为什么 GDPR 很重要？

GDPR 为处理个人数据建立了更强大的数据保护框架。GDPR 可以带来：

- 新的和增强的个人权利
- 扩展了个人数据的定义
- 数据处理方新的责任
- 非遵守方可能在经济上会受到大量处罚
- 强制数据违反通知

1.5.3.2. 更多关于 GDPR 的信息

- EU GDPR Information Portal
- Red Hat GDPR website

1.5.4. 针对 GDPR 的产品配置

以下小节描述了 Red Hat Advanced Cluster Management for Kubernetes 平台的数据管理的各个方面，并提供了有关帮助客户端满足 GDPR 要求的能力信息。

1.5.5. 数据生命周期

Red Hat Advanced Cluster Management for Kubernetes 是一个应用程序平台，用于开发并管理内部、容器化的应用程序。它是一个用于管理容器的集成环境，包括容器编配器 Kubernetes、集群生命周期、应用程序生命周期以及安全框架（监管、风险和合规）。

因此，Red Hat Advanced Cluster Management for Kubernetes 平台主要处理与平台的配置和管理相关的技术数据，其中的一些数据可能会涉及到受 GDPR 影响的数据。Red Hat Advanced Cluster Management for Kubernetes 平台还处理管理平台的用户的信息。在这个文档中会介绍这些数据，以使负责满足 GDPR 要求的用户了解这些内容。

这些数据会在本地或者远程文件系统中，以配置文件或数据库的形式存在。在 Red Hat Advanced Cluster Management for Kubernetes 平台上运行的应用程序可能会涉及到其它形式的、受 GDPR 影响的个人数据。用于保护和管理平台数据的机制也可用于平台上运行的应用程序。对于在 Red Hat Advanced Cluster Management for Kubernetes 平台上运行的应用程序所收集个人数据，可能还需要额外的机制来进行管理和保护。

为了更好了解 Red Hat Advanced Cluster Management for Kubernetes 平台及其数据流，您需要对 Kubernetes、Docker 和 Operator 的工作原理有所了解。这些开源组件是 Red Hat Advanced Cluster Management for Kubernetes 平台的基础。您使用 Kubernetes 部署来放置应用程序实例，这些实例会被内置到引用 Docker 镜像的 Operator 中。Operator 包含应用程序的详细信息，Docker 镜像包含应用程序需要运行的所有软件包。

1.5.5.1. Red Hat Advanced Cluster Management for Kubernetes 平台的数据流类型

作为一个平台，Red Hat Advanced Cluster Management for Kubernetes 需要不同类别的技术数据，这些数据可能会被视为个人数据，如管理员用户 ID 和密码、服务用户 ID 和密码、IP 地址以及

Kubernetes 节点名称。Red Hat Advanced Cluster Management for Kubernetes 平台还处理管理平台的用户的信息。在平台中运行的应用程序可能会使用与平台无关的其他类别的个人数据。

本文档后续部分将介绍如何收集/创建这些技术数据、存储、访问、安全、日志和删除。

1.5.5.2. 用于在线联系的个人数据

用户可以以各种方式提交在线评论/反馈/请求，主要有：

- 如果使用 Slack 频道，公共的 Slack 社区
- 产品文档中的公共注释或问题单
- 技术社区中的公共对话

通常，只使用客户名称和电子邮件地址，以便可以进行回复，对个人数据的使用符合 红帽在线隐私声明。

1.5.6. 数据收集

Red Hat Advanced Cluster Management for Kubernetes 平台不会收集敏感的个人数据。它会创建和管理技术数据，如管理员用户 ID 和密码、服务用户 ID 和密码、IP 地址以及 Kubernetes 节点名称。这些数据可能会被视为个人数据。Red Hat Advanced Cluster Management for Kubernetes 平台还处理管理平台的用户的信息。只有系统管理员才可以通过使用基于角色的访问控制的管理控制台访问此类信息，或者系统管理员登陆到一个 Red Hat Advanced Cluster Management for Kubernetes 平台节点才可以访问。

在 Red Hat Advanced Cluster Management for Kubernetes 平台上运行应用程序可能会收集个人数据。

当您在评估 Red Hat Advanced Cluster Management for Kubernetes 运行容器化应用程序，并需要符合 GDPR 要求时，您必须考虑应用程序收集的个人数据类型以及是如何管理这些数据的，例如：

- 当数据流向应用程序或从应用程序流出时，数据是如何被保护的？数据是否在传输中加密？

- 数据是如何被应用程序存储的？数据在不用时是否被加密？
- 用于访问应用程序的凭证是如何被收集和存储的？
- 应用程序用于访问数据源所使用的凭证是如何被收集和存储的？
- 如何根据需要删除应用程序收集的数据？

这不是 Red Hat Advanced Cluster Management for Kubernetes 平台所收集的数据类型的完整列表。它只作为一个示例以供考虑。如果您对数据类型有任何疑问，请联络红帽。

1.5.7. 数据存储

对于与配置和管理平台相关的技术数据，Red Hat Advanced Cluster Management for Kubernetes 平台会把它们以配置文件或数据库的形式保存在本地或远程文件系统中。对于存储的数据，必须考虑它们的安全性。Red Hat Advanced Cluster Management for Kubernetes 平台支持使用 dm-crypt 对存储的数据进行加密。

下面是主要的数据存储形式，您可能需要进行与 GDPR 相关的考虑。

- 平台配置数据： 通过更新带有常规设置、Kubernetes、日志、网络、Docker 和其他设置属性的配置 YAML 文件，可以自定义 Red Hat Advanced Cluster Management for Kubernetes 平台的配置。这些数据会作为 Red Hat Advanced Cluster Management for Kubernetes 平台的安装程序的输入被使用来部署节点。这些属性还包括用于 bootstrap 的管理员用户 ID 和密码。
- Kubernetes 配置数据： Kubernetes 集群状态数据保存在分布式“键-值”存储 etcd 中。
- 用户身份验证数据，包括用户 ID 和密码： 通过客户端企业级 LDAP 目录处理用户 ID 和密码管理。在 LDAP 中定义的用户和组可添加到 Red Hat Advanced Cluster Management for Kubernetes 平台的团队中，并分配访问角色。Red Hat Advanced Cluster Management for Kubernetes 平台会储存来自 LDAP 的电子邮件地址和用户 ID，但不保存密码。Red Hat Advanced Cluster Management for Kubernetes 平台会存储组名称，并在登录时缓存用户所属的可用组。组成员不会以长期形式有效。必须考虑在企业级 LDAP 中保护用户和组数据。Red Hat Advanced Cluster Management for Kubernetes 平台也包括了一个身份认证服务 Open ID

Connect（OIDC），它与企业目录服务进行交互并维护访问令牌。此服务使用 ETCD 作为后端存储。

- 服务身份验证数据，包括用户 ID 和密码： Red Hat Advanced Cluster Management for Kubernetes 平台组件使用的、用于在组件间进行访问的凭证被定义为 Kubernetes Secret。所有 Kubernetes 资源定义都保留在 etcd 键-值形式的数据存储中。初始凭证值在平台配置数据中定义，作为 Kubernetes Secret 配置 YAML 文件。如需更多信息，请参阅 Kubernetes 文档中的 Secret。

1.5.8. 数据访问

您可以通过以下定义的产品接口集合访问 Red Hat Advanced Cluster Management for Kubernetes 平台数据。

- Web 用户界面（控制台）
- Kubernetes kubectl CLI
- Red Hat Advanced Cluster Management for Kubernetes CLI
- oc CLI

这些接口可用于对 Red Hat Advanced Cluster Management for Kubernetes 集群进行管理级别的更改。当发出一个请求时，安全使用 Red Hat Advanced Cluster Management for Kubernetes 的管理访问权限涉及三个逻辑的、有特定顺序的阶段： 身份验证、角色映射和授权。

1.5.8.1. 身份验证

Red Hat Advanced Cluster Management for Kubernetes 平台的身份验证管理程序接受来自控制台的用户凭证，并将凭证转发到后端的 OIDC 供应商，后者根据企业目录验证用户凭证。然后，OIDC 供应商会向身份验证程序返回一个带有 JSON Web Token（JWT）内容的身份验证 cookie（auth-cookie）。JWT 令牌包括了身份验证请求时的组成员信息，以及用户 ID 和电子邮件地址等信息。然后，这个身份验证 cookie 会发送到控制台。在会话存在期间，cookie 会被刷新。在退出控制台或关闭浏览器后，这个 cookie 会在 12 小时内有效。

对于所有来自控制台的验证请求，前端 NGINX 服务器对请求中的可用身份验证 cookie 进行解码，并通过调用验证管理程序来验证请求。

Red Hat Advanced Cluster Management for Kubernetes 平台的 CLI 需要用户在登陆时提供凭证。

kubectl 和 oc CLI 也需要凭证来访问集群。这些凭证可以从管理控制台获得，并在 12 小时后过期。支持通过服务帐户访问。

1.5.8.2. 角色映射

Red Hat Advanced Cluster Management for Kubernetes 平台支持的基于角色的控制访问（RBAC）。在角色映射阶段，身份验证阶段提供的用户名映射到用户或组角色。在授权哪些管理操作可由经过身份验证的用户执行时使用角色。

1.5.8.3. 授权

Red Hat Advanced Cluster Management for Kubernetes 平台对集群配置操作的角色控制访问，适用于 catalog 和 Helm 资源，以及 Kubernetes 资源。提供了几个 IAM（Identity and Access Management）角色，包括 Cluster Administrator、Administrator、Operator、Editor、Viewer。在将用户或用户组添加到一个团队时，会为用户或用户组分配一个角色。对资源的团队访问可以由命名空间控制。

1.5.8.4. Pod 安全性

Pod 安全策略用于设置集群级别的控制，控制 pod 可以做什么或可以访问什么。

1.5.9. 数据处理

Red Hat Advanced Cluster Management for Kubernetes 的用户可以通过系统配置，来处理和保护与配置和管理相关的技术数据。

基于角色的访问控制 （RBAC）可控制用户可访问哪些数据和功能。

Data-in-transit 通过使用 TLS 加以保护。HTTP （TLS 底层）是用来在用户客户端和后端服务间进行安全的数据传输。用户可以指定在安装过程中要使用的 root 证书。

Data-at-rest 的保护是通过使用 dm-crypt 加密数据来实现的。

那些用来管理和保护 Red Hat Advanced Cluster Management for Kubernetes 平台的技术数据的机制，同样可用于对用户开发的或用户提供的应用程序的个人数据进行管理和保护。客户可以开发自己的功能进行进一步的控制。

1.5.10. 数据删除

Red Hat Advanced Cluster Management for Kubernetes 平台提供了命令、API 和用户界面操作以删除由产品创建或收集的数据。用户可以使用这些功能删除技术数据，如服务用户 ID 和密码、IP 地址、Kubernetes 节点名称或其他平台配置数据，并可以管理平台的用户的信息。

Red Hat Advanced Cluster Management for Kubernetes 平台中可用来进行数据删除的方法：

- 与平台配置相关的所有技术数据，都可通过管理控制台或 Kubernetes kubectl API 删除。

Red Hat Advanced Cluster Management for Kubernetes 平台中用于删除帐户数据的方法：

- 与平台配置相关的所有技术数据，都可通过 Red Hat Advanced Cluster Management for Kubernetes 或 Kubernetes kubectl API 删除。

删除通过企业级 LDAP 目录管理的用户 ID 和密码数据的功能，需要由与 Red Hat Advanced Cluster Management for Kubernetes 平台集成的 LDAP 产品提供。

1.5.11. 限制使用个人数据的能力

通过本文档中介绍的工具，Red Hat Advanced Cluster Management for Kubernetes 平台可以对最终用户对个人数据的使用加以限制。

根据 GDPR，用户的访问、修改和处理权限都需要被加以限制。请参考本文档的其它部分来控制以下内容：

- 访问权限
  - Red Hat Advanced Cluster Management for Kubernetes 平台管理员可以使用 Red Hat Advanced Cluster Management for Kubernetes 平台的功能提供个人对他们的数据的独立访问。

- Red Hat Advanced Cluster Management for Kubernetes 平台管理员可以使用 Red Hat Advanced Cluster Management for Kubernetes 平台的功能，可以提供 Red Hat Advanced Cluster Management for Kubernetes 平台为某个个人保存的什么个人数据的信息。

- 修改权限
  - Red Hat Advanced Cluster Management for Kubernetes 平台管理员可以使用 Red Hat Advanced Cluster Management for Kubernetes 平台的功能来允许一个个人修改自己的数据。
  - Red Hat Advanced Cluster Management for Kubernetes 平台管理员可以使用 Red Hat Advanced Cluster Management for Kubernetes 平台的功能为一个个人修改其个人数据。
- 限制处理的权利
  - Red Hat Advanced Cluster Management for Kubernetes 平台管理员可以使用 Red Hat Advanced Cluster Management for Kubernetes 平台的功能停止处理一个个人的数据。

1.5.12. 附录

作为一个平台，Red Hat Advanced Cluster Management for Kubernetes 需要不同类别的技术数据，这些数据可能会被视为个人数据，如管理员用户 ID 和密码、服务用户 ID 和密码、IP 地址以及 Kubernetes 节点名称。Red Hat Advanced Cluster Management for Kubernetes 平台也会处理管理平台的人员的信息。在平台中运行的应用程序可能会引入其它在平台中未知的个人数据类别。

本附录包含平台服务日志记录的数据详情。

1.6. FIPS 就绪性

Red Hat Advanced Cluster Management for Kubernetes 为 FIPS 设计。当以 FIPS 模式在 Red Hat OpenShift Container Platform 上运行时，OpenShift Container Platform 会使用 Red Hat Enterprise Linux 加密库提交给 NIST 进行 OpenShift Container Platform 支持的架构。有关 NIST 验证程序的更多信息，请参阅加密模块验证程序。有关为验证提交的 RHEL 加密库的单独版本的最新 NIST 状态，请参阅 Compliance Activities 和 Government Standards。

如果您计划管理启用了 FIPS 的集群，则必须在配置为以 FIPS 模式操作的 OpenShift Container Platform 集群上安装 Red Hat Advanced Cluster Management。hub 集群必须处于 FIPS 模式，因为在受管集群中使用在 hub 集群中创建的加密。

要在受管集群中启用 FIPS 模式，请在置备 OpenShift Container Platform 受管集群时设置 fips: true。置备集群后您无法启用 FIPS。如需更多信息，请参阅 OpenShift Container Platform 文档，是否需要额外的安全集群？

1.6.1. 限制

阅读 Red Hat Advanced Cluster Management 和 FIPS 中的以下限制。

- 在配置提供的存储时，必须对搜索和可观察组件使用的持久性卷声明(PVC)和 S3 存储进行加密。Red Hat Advanced Cluster Management 不提供存储加密，请参阅 OpenShift Container Platform 文档 配置持久性存储。

- 当使用 Red Hat Advanced Cluster Management 控制台置备受管集群时，在受管集群创建的 Cluster details 部分中选中以下复选框来启用 FIPS 标准：

> *FIPS with information text: Use the Federal Information Processing Standards (FIPS) modules provided with Red Hat Enterprise Linux CoreOS instead of the default Kubernetes cryptography suite file before you deploy the new managed cluster.*

1.7. OBSERVABILITY 支持

- Red Hat Advanced Cluster Management 使用 Red Hat OpenShift Data Foundation （以前称为 Red Hat OpenShift Container Platform）进行了测试并被完全支持。

- Red Hat Advanced Cluster Management 支持在用户提供的兼容 S3 API 的第三方对象存储中多集群可观察 Operator 的功能。Observability 服务使用 Thanos 支持的、稳定的对象存储。

- Red Hat Advanced Cluster Management 支持工作包括合理的努力来识别根本原因。如果您创建一个支持问题单，且根本原因是您提供的 S3 兼容对象存储，则必须使用客户支持频道来创建一个问题。