Red Hat OpenShift Service on AWS 4

集群管理

配置 Red Hat OpenShift Service on AWS clusters

Last Updated: 2024-06-28

Red Hat OpenShift Service on AWS 4 集群管理

配置 Red Hat OpenShift Service on AWS clusters

法律通告

摘要

本文档提供有关配置 Red Hat OpenShift Service on AWS 集群的安全信息。

目录

第 1 章 集群通知

集群通知是有关集群状态、健康或性能的信息。

集群通知是 Red Hat Site Reliability Engineering (SRE)与您有关受管集群健康状况的主要方法。SRE 也可能使用集群通知来提示您执行操作，以解决或防止集群出现问题。

集群所有者和管理员必须定期检查和操作集群通知，以确保集群保持健康且受支持。

您可以在集群的 **Cluster history** 选项卡中查看 Red Hat Hybrid Cloud Console 中的集群通知。默认情况下，只有集群所有者接收集群通知作为电子邮件。如果其他用户需要接收集群通知电子邮件，请将每个用户添加为集群的通知联系人。

1.1. 其他资源

- 客户职责：检查和操作集群通知
- 集群通知电子邮件
- 故障排除：集群通知

1.2. 从集群通知中预期的内容

作为集群管理员，您需要了解何时和为什么发送集群通知及其类型和严重性级别，以便有效地了解集群的健康和管理需求。

1.2.1. 集群通知策略

集群通知旨在让您了解集群的健康状况以及影响它的高影响事件。

大多数集群通知都会自动生成并自动发送，以确保您立即了解集群状态的问题或重要更改。

在某些情况下，Red Hat Site Reliability Engineering (SRE)创建并发送集群通知，以便为复杂的问题提供额外的上下文和指导。

集群通知不会针对低影响的事件、低风险安全更新、日常操作和维护，或由 SRE 快速解决的临时问题发送。

红帽服务在以下情况下自动发送通知：

- 远程健康监控或环境验证检查会检测集群中的问题，例如当 worker 节点有低磁盘空间时。
- 大量的集群生命周期事件（例如调度维护或升级时），或者集群操作会受到事件的影响，但不需要客户干预。
- 大量的集群管理更改，例如，当集群所有权或管理控制从一个用户转移到另一个用户时。
- 您的集群订阅会被更改或更新，例如，当红帽对集群进行订阅条款或功能的更新时。

SRE 在以下情况下创建和发送通知：

- 事件会导致降级或中断会影响集群的可用性或性能，例如，您的云供应商有区域中断。SRE 发送后续通知以告知您事件解析进度以及事件被解决的时间。
- 集群中检测到安全漏洞、安全漏洞或异常活动。

- 红帽检测到您所做的更改正在创建，或可能会导致集群不稳定。
- 红帽检测到您的工作负载会导致集群中的性能下降或不稳定。

1.2.2. 集群通知严重性级别

每个集群通知都有一个关联的严重性级别，可帮助您识别对您的业务有最大影响的通知。您可以在 Red Hat Hybrid Cloud Console 的 **Cluster history** 选项卡中根据 Red Hat Hybrid Cloud Console 中的这些严重性级别过滤集群通知。

红帽对集群通知使用以下严重性级别，从最严重到最严重：

Critical

需要立即操作。服务或集群的一个或多个关键功能无法正常工作，或者将很快停止工作。关键警报足以在调用人员上页面并中断常规工作流。

主

强烈建议立即采取行动。集群的一个或多个关键功能将很快停止工作。如果问题及时无法及时解决，则可能会造成关键问题。

Warning

尽快采取行动。集群的一个或多个关键功能在最佳上无法正常工作，可能会进一步降级，但不给集群运行造成即时的危险。

info

不需要操作。此严重性不描述需要解决的问题，只有有关有意义的或重要生命周期、服务或集群事件的重要信息。

Debug

不需要操作。调试通知提供有关不太重要的生命周期、服务或集群事件的低级别信息，以帮助调试意外行为。

1.2.3. 集群通知类型

每个集群通知都有一个关联的通知类型，可帮助您识别与您的角色和职责相关的通知。您可以根据 Red Hat Hybrid Cloud Console 中的 **Cluster history** 选项卡中的这些类型过滤集群通知。

红帽使用以下通知类型来指示通知相关性。

容量管理

通知与更新、创建或删除节点池、机器池、计算副本或配额（负载均衡器、存储等）相关的事件。

集群访问

有关添加或删除组、角色或身份提供程序相关的事件通知，例如当 SRE 无法访问集群时，因为 STS 凭证已过期，当 AWS 角色出现配置问题时，或者您添加或删除身份提供程序。

集群附加组件

有关附加组件的附加管理或升级维护相关的事件通知，例如当安装、升级或删除附加组件时，或者因为未满足要求而无法安装。

集群配置

集群调整事件、工作负载监控和动态检查的通知。

集群生命周期

集群或集群资源创建、删除和注册通知，或更改集群或资源状态（如就绪或休眠）。

集群网络

与集群网络相关的通知，包括 HTTP/S 代理、路由器和入口状态。

集群所有权

与集群所有权相关的通知，从一个用户传输到另一个用户。

集群扩展

与更新、创建或删除节点池、机器池、计算副本或配额相关的通知。

集群安全性

与集群安全性相关的事件（例如，增加访问尝试次数、信任捆绑包的更新或具有安全影响的软件更新）。

集群订阅

集群过期、试用集群通知或从免费切换到付费。

集群更新

与升级相关的任何内容，如升级维护或启用。

客户支持

支持问题单状态的更新。

常规通知

默认通知类型。这仅用于没有更特定类别的通知。

1.3. 使用 RED HAT HYBRID CLOUD CONSOLE 查看集群通知

集群通知提供有关集群健康状况的重要信息。您可以在 Red Hat Hybrid Cloud Console 上的 **Cluster history** 选项卡中查看发送到集群的通知。

前提条件

- 已登陆到 Hybrid Cloud Console。

流程

1. 进入到 Hybrid Cloud Console 的 Clusters 页面。
2. 点集群名称进入集群详情页面。
3. 点 **Cluster history** 选项卡。
   集群通知会出现在 Cluster history 标题下。
4. 可选：相关集群通知的过滤器
   使用过滤器控制来隐藏与您无关的集群通知，以便您可以专注于专业知识区域或解决关键问题。您可以根据通知描述、严重性级别、通知类型、通知类型以及哪个系统或个人触发通知的内容过滤通知。

1.4. 集群通知电子邮件

默认情况下，当向集群发送通知时，也会将其作为电子邮件发送到集群所有者。您可以为通知电子邮件配置额外的接收者，以确保所有合适的用户保持有关集群状态的了解。

1.4.1. 在集群中添加通知联系人

当集群通知发送到集群时，通知联系人会收到电子邮件。默认情况下，只有集群所有者接收集群通知电子邮件。您可以将其他集群用户配置为集群支持设置中的其他通知联系人。

前提条件

- 集群已部署并注册到 Red Hat Hybrid Cloud Console。
- 已登陆到 Hybrid Cloud Console。
- 预期的通知接收者在集群中有一个用户帐户。

流程

1. 进入到 Hybrid Cloud Console 的 Clusters 页面。
2. 点集群名称进入集群详情页面。
3. 点 **Support** 选项卡。
4. 在 **Support** 选项卡上，找到 **Notification contacts** 部分。
5. 点 **Add notification contact**。
6. 在 **Red Hat username or email** 字段中输入新接收者的电子邮件地址或用户名。
7. 点 **Add contact**。

验证步骤

- 此时会显示"Notification contact added successfully"消息。

1.4.2. 从集群中删除通知联系人

当集群通知发送到集群时，通知联系人会收到电子邮件。

您可以删除集群支持设置中的通知联系人，以防止它们接收通知电子邮件。

前提条件

- 集群已部署并注册到 Red Hat Hybrid Cloud Console。
- 已登陆到 Hybrid Cloud Console。

流程

1. 进入到 Hybrid Cloud Console 的 Clusters 页面。
2. 点集群名称进入集群详情页面。
3. 点 **Support** 选项卡。
4. 在 **Support** 选项卡上，找到 **Notification contacts** 部分。
5. 点您要删除的接收者旁的选项菜单(criu)。
6. 点击 **Delete**。

验证步骤

- 这时将显示"Notification contact deleted successfully"消息。

1.5. 故障排除

如果您没有收到集群通知电子邮件

- 确保从 **@redhat.com** 地址发送的电子邮件没有过滤掉您的电子邮件。
- 确保您的正确电子邮件地址被列为集群的通知联系人。
- 询问集群所有者或管理员以作为通知联系人添加： 集群 通知电子邮件。

如果您的集群没有收到通知

- 确保集群可以访问 **api.openshift.com** 的资源。
- 确保根据记录的先决条件配置了防火墙： AWS 防火墙先决条件

第 2 章 配置私有连接

2.1. 配置私有连接

可以实施私有集群访问权限，以满足 Red Hat OpenShift Service on AWS (ROSA) 环境的需求。

流程

1. 访问您的 ROSA AWS 帐户，并使用以下一个或多个方法建立与集群的私有连接：
   - 配置 AWS VPC 对等 ：启用 VPC 对等路由两个私有 IP 地址之间的网络流量。
   - 配置 AWS VPN ：建立虚拟专用网络，将私有网络安全地连接到您的 Amazon 虚拟私有云。
   - 配置 AWS Direct Connect ：配置 AWS Direct Connect 以在您的私有网络和 AWS Direct Connect 位置之间建立专用网络连接。
2. 在 ROSA 上配置私有集群。

2.2. 配置 AWS VPC 对等

这个示例过程配置一个 Amazon Web Services (AWS) VPC，其中包含一个 Red Hat OpenShift Service on AWS 集群作为另外一个 AWS VPC 网络的对等（peer）。有关创建 AWS VPC 交换连接或其它可能配置的更多信息，请参阅 AWS VPC Peering 指南。

2.2.1. VPC 对等术语

当在两个独立 AWS 帐户中的两个 VPC 之间设置 VPC 对等连接时，会使用以下术语：

Red Hat OpenShift Service on AWS AWS 账户	包括 Red Hat OpenShift Service on AWS 集群的 AWS 账户。
Red Hat OpenShift Service on AWS Cluster VPC	包括 Red Hat OpenShift Service on AWS 集群的 VPC。
客户 AWS 帐户	您要进行对等的非 Red Hat OpenShift Service on AWS AWS 账户。
客户 VPC	您要进行对等的 AWS 帐户中的 VPC。
客户 VPC 区域	客户 VPC 所在的区域。

注意

从 2018 年 7 月起，AWS 支持在所有商业区域 (不包括中国) 之间的 inter-region VPC。

2.2.2. 启动 VPC 对等请求

您可以将来自 Red Hat OpenShift Service on AWS 帐户的 VPC 对等连接请求发送到客户 AWS 帐户。

前提条件

- 收集启动对等请求所需的客户 VPC 的以下信息：
  - 客户 AWS 帐号
  - 客户 VPC ID
  - 客户 VPC 区域
  - 客户 VPC CIDR
- 检查 Red Hat OpenShift Service on AWS Cluster VPC 使用的 CIDR 块。如果与客户 VPC 的 CIDR 块重叠或匹配，则无法在这两个 VPC 之间对等。有关详细信息，请参阅 Amazon VPC 不支持的 VPC 仪表板配置文档。如果 CIDR 块没有重叠，您可以继续执行这个过程。

流程

1. 登录到 Red Hat OpenShift Service on AWS AWS 帐户的 Web 控制台，在托管集群的区域进入 **VPC 仪表板**。
2. 进入 **Peering Connections** 页面，点 **Create Peering Connection** 按钮。
3. 验证您登录到的帐户的详情，以及您要连接的帐户和 VPC 的详情：
   a. **对等连接名称标签** ：为 VPC Peering Connection 设置一个描述性名称。
   b. **VPC (Requester)** ：从 *list 下拉列表中选择 Red Hat OpenShift Service on AWS Cluster VPC ID。
   c. **帐户** ：选择 **另一个帐户** 并提供客户 AWS 帐户号 *（不带横线）。
   d. **区域** ：如果客户 VPC 区域与当前区域不同，请选择 **Another Region**，然后从下拉列表中选择客户 VPC 区域。
   e. **VPC (Accepter)** ：设置客户 VPC ID。
4. 点 **Create Peering Connection**。
5. 确认请求变为 **Pending** 状态。如果它变为 **Failed** 状态，请确认详情并重复此过程。

2.2.3. 接受 VPC 对等请求

创建 VPC 对等连接后，您必须接受客户 AWS 帐户中的请求。

前提条件

- 启动 VPC 对等请求。

流程

1. 登录到 AWS Web 控制台。
2. 进入 **VPC Service**。
3. 进入 **Peering Connections**。
4. 点 **Pending peering connection**。
5. 确认请求来自于的 AWS 帐户和 VPC ID。这应该来自 Red Hat OpenShift Service on AWS AWS 帐户和 Red Hat OpenShift Service on AWS 集群 VPC。
6. 点 **Accept Request**。

2.2.4. 配置路由表

接受 VPC 对等请求后，两个 VPC 必须将其路由配置为在对等连接间进行通信。

前提条件

- 启动并接受 VPC 对等请求。

流程

1. 登录到 Red Hat OpenShift Service on AWS AWS 帐户的 AWS Web 控制台。
2. 进入 **VPC Service**，然后使用 **Route Tables**。
3. 为 Red Hat OpenShift Service on AWS Cluster VPC 选择路由表。

注意

在一些集群中，特定 VPC 可能有多个路由表。选择具有多个显式关联子网的私有子网。

4. 选择 **Routes** 标签页，然后选择 **Edit**。
5. 在 **Destination** 文本框中输入 Customer VPC CIDR 块。
6. 在 **Target** 文本框中输入 Peering Connection ID。
7. 点 **Save**。
8. 您必须使用其他 VPC 的 CIDR 块完成相同的进程：
    a. 登录到 Customer AWS Web Console → **VPC Service** → **Route Tables**。
    b. 选择 VPC 的路由表。
    c. 选择 **Routes** 标签页，然后选择 **Edit**。
    d. 在 **Destination** 文本框中输入 Red Hat OpenShift Service on AWS Cluster VPC CIDR 块。
    e. 在 **Target** 文本框中输入 Peering Connection ID。

f. 点 **Save**。

VPC peering 连接现已完成。按照验证过程，确保对等连接间的连接正常工作。

2.2.5. 验证 VPC 对等的故障排除

设置 VPC 对等连接后，最好确认它已经配置且正常工作。

前提条件

- 启动并接受 VPC 对等请求。
- 配置路由表。

流程

- 在 AWS 控制台中，查看对等集群 VPC 的路由表。确保配置了路由表的步骤，并且有一个路由表条目，将 VPC CIDR 范围目的地指向对等连接目标。
  如果路由在 AWS Cluster VPC 上的 Red Hat OpenShift Service 路由表和客户 VPC 路由表上都正确，则应该使用 **netcat** 方法测试连接。如果测试调用成功，则 VPC 对等可以正常工作。
- 要测试到端点设备的网络连接，**nc** （或 **netcat**）是一个有用的故障排除工具。它包含在默认镜像中，如果可以建立连接，提供快速和清晰的输出：

  a. 使用 **busybox** 镜像创建一个临时 pod，它会自行清理：

  ```
  $ oc run netcat-test \
      --image=busybox -i -t \
      --restart=Never --rm \
      -- /bin/sh
  ```

  b. 使用 **nc** 检查连接。

  - 成功连接结果示例：

  ```
  / nc -zvv 192.168.1.1 8080
  10.181.3.180 (10.181.3.180:8080) open
  sent 0, rcvd 0
  ```

  - 连接结果示例：

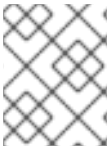
  ```
  / nc -zvv 192.168.1.2 8080
  nc: 10.181.3.180 (10.181.3.180:8081): Connection refused
  sent 0, rcvd 0
  ```

  c. 退出容器，该容器自动删除 Pod:

  ```
  / exit
  ```

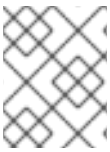
2.3. 配置 AWS VPN

这个示例流程配置 Amazon Web Services (AWS) Red Hat OpenShift Service on AWS 集群来使用客户的现场硬件 VPN 设备。



注意

AWS VPN 目前不提供将 NAT 应用到 VPN 流量的受管选项。如需了解更多详细信息，请参阅 AWS 知识库。



注意

不支持通过私有连接路由所有流量，如 **0.0.0.0/0**。这需要删除互联网网关，它将禁用 SRE 管理流量。

有关使用硬件 VPN 设备将 AWS VPC 连接到远程网络的更多信息，请参阅 Amazon VPC VPN 连接 文档。

2.3.1. 创建 VPN 连接

您可以按照以下过程，配置 Amazon Web Services (AWS) Red Hat OpenShift Service on AWS 集群来使用客户的现场硬件 VPN 设备。

前提条件

- 硬件 VPN 网关设备模型和软件版本，例如 Cisco ASA 运行版本 8.3。请参阅 Amazon VPC Network Administrator 指南，以确认 AWS 是否支持您的网关设备。
- VPN 网关设备的公共、静态 IP 地址。
- BGP 或静态路由：如果需要 BGP，则需要 ASN。如果静态路由，必须至少配置一个静态路由。
- 可选：一个可访问服务的 IP 和端口/Protocol 来测试 VPN 连接。

2.3.1.1. 配置 VPN 连接

流程

1. 登陆到 Red Hat OpenShift Service on AWS AWS Account Dashboard，然后进入 VPC 仪表板。
2. 点**您的 VPC**，找到包含 Red Hat OpenShift Service on AWS 集群上的 VPC 的名称和 VPC ID。
3. 在 VPC 仪表板中，点 **Customer Gateway**。
4. 点 **Create Customer Gateway** 并为它指定一个有意义的名称。
5. 选择路由方法：**Dynamic** 或 **Static**。
6. 如果 Dynamic，在出现的字段中输入 BGP ASN。
7. 粘贴 VPN 网关端点 IP 地址。
8. 点 **Create**。
9. 如果您还没有附加到预期的 VPC 的虚拟私有网关：
    a. 在 VPC 仪表板中点 **Virtual Private Gateway**。
    b. 点 **Create Virtual Private Gateway**，为它指定一个有意义的名称，然后点 **Create**。

c. 默认 Amazon 默认 ASN。

d. 选择新创建的网关，点 **Attach to VPC**，并将其附加到您之前标识的集群 VPC。

2.3.1.2. 建立 VPN 连接

流程

1. 在 VPC 仪表板中，点 **Site-to-Site VPN 连接**。
2. 点 **Create VPN 连接**。
   a. 为它指定有意义的名称标签。
   b. 选择之前创建的虚拟专用网关。
   c. 对于客户网关，请选择**现有**。
   d. 按名称选择客户网关设备。
   e. 如果 VPN 将使用 BGP，请选择 **Dynamic**，否则选择 **Static**。输入静态 IP CIDR。如果有多个 CIDR，请将每个 CIDR 添加为**另一个规则**。
   f. 点 **Create**。
   g. 等待 VPN 状态更改为 **Available**，大约需要 5 到 10 分钟。
3. 选择您刚才创建的 VPN 并点 **Download Configuration**。
   a. 从下拉列表中，选择客户网关设备的供应商、平台和版本，然后点**下载**。
   b. **Generic** 供应商配置也可用于以纯文本格式检索信息。

注意

建立 VPN 连接后，请确定设置路由传播，或者 VPN 可能无法正常工作。

注意

请注意，VPC 子网信息，您必须作为远程网络添加到配置中。

2.3.1.3. 启用 VPN 路由传播

设置 VPN 连接后，您必须确保启用了路由传播，以便将所需的路由添加到 VPC 的路由表中。

流程

1. 在 VPC 仪表板中点 **Route Tables**。
2. 选择与 VPC 关联的私有 Route 表，该表包含您的 Red Hat OpenShift Service on AWS。

注意

在一些集群中，特定 VPC 可能有多个路由表。选择具有多个显式关联子网的私有子网。

3. 点 **Route Propagation** 选项卡。
4. 在显示的表中，您应看到之前创建的虚拟专用网关。检查 **Propagate 列**中的值。
    a. 如果 Propagate 设为 **No**，点 **Edit route propagation**，选中虚拟私有网关名称旁边的 Propagate 复选框，再点 **Save**。

在配置 VPN 隧道和 AWS 检测到它为 **Up**后，您的静态或 BGP 路由会自动添加到路由表中。

2.3.2. 验证 VPN 连接

设置 VPN 隧道后，您可以验证隧道是否在 AWS 控制台中，并且通过隧道连接正常工作。

前提条件

- 创建 VPN 连接。

流程

1. **验证隧道是否在 AWS 中启动。**
    a. 在 VPC 仪表板中点 **VPN 连接**。
    b. 选择您之前创建的 VPN 连接并点 **Tunnel Details** 选项卡。
    c. 您应该可以看到至少一个 VPN 隧道为 **Up**。
2. **验证连接。**
   要测试到端点设备的网络连接，**nc** （或 **netcat**）是一个有用的故障排除工具。它包含在默认镜像中，如果可以建立连接，提供快速和清晰的输出：
    a. 使用 **busybox** 镜像创建一个临时 pod，它会自行清理：

```
$ oc run netcat-test \
    --image=busybox -i -t \
    --restart=Never --rm \
    -- /bin/sh
```

    b. 使用 **nc** 检查连接。
        - 成功连接结果示例：

```
/ nc -zvv 192.168.1.1 8080
10.181.3.180 (10.181.3.180:8080) open
sent 0, rcvd 0
```

        - 连接结果示例：

```
/ nc -zvv 192.168.1.2 8080
nc: 10.181.3.180 (10.181.3.180:8081): Connection refused
sent 0, rcvd 0
```

    c. 退出容器，该容器自动删除 Pod:

```
/ exit
```

2.3.3. VPN 连接故障排除

隧道没有连接
如果隧道连接仍然处于 **Down** 状态，您可以验证以下几个方面：

- AWS 隧道不会启动 VPN 连接。连接尝试必须从客户网关启动。
- 确保您的源流量来自与配置的客户网关相同的 IP。AWS 将静默丢弃到源 IP 地址不匹配的网关的流量。
- 确保您的配置与 AWS 支持的值匹配。这包括 IKE 版本、DH 组、IKE 生命周期等等。
- 再次检查 VPC 的路由表。确保启用传播，并在路由表中有您之前创建的虚拟专用网关作为目标的条目。
- 确认您没有可能会造成中断的防火墙规则。
- 检查您是否使用基于策略的 VPN，因为这可能导致配置情况复杂。
- 有关故障排除步骤，请参阅 AWS 知识库。

隧道没有保持连接
如果隧道连接无法稳定地保持为 **Up**，代表所有 AWS 隧道连接都必须从您的网关启动。AWS 隧道不会启动隧道。

红帽建议在持续发送"中断"流量的隧道端设置 SLA Monitor (Cisco ASA)或一些设备，例如 **ping**、**nc** 或 **telnet**，位于 VPC CIDR 范围内配置的任何 IP 地址。连接是否成功并不重要，只要流量被定向到隧道。

处于 Down 状态的二级隧道
创建 VPN 隧道时，AWS 会创建额外的故障转移隧道。根据网关设备，有时第二个隧道将显示为 **Down** 状态。

AWS 通知如下：

> You have new non-redundant VPN connections
>
> One or more of your vpn connections are not using both tunnels. This mode of operation is not highly available and we strongly recommend you configure your second tunnel. View your non-redundant VPN connections.

2.4. 配置 AWS DIRECT CONNECT

此过程描述了接受 Red Hat OpenShift Service on AWS 的 AWS Direct Connect 虚拟接口。有关 AWS Direct Connect 类型和配置的更多信息，请参阅 AWS Direct Connect 组件文档。

2.4.1. AWS Direct Connect 方法

Direct Connect 连接需要一个托管的、连接到一个 Direct Connect Gateway (DXGateway) 的虚拟接口 (VIF)，它与一个 Virtual Gateway (VGW) 或一个 Transit Gateway 进行关联，从而可以访问同一个账户或另外一个账户中的远程 VPC。

如果您没有现有的 DXGateway，则需要创建托管的 VIF，在 Red Hat OpenShift Service on AWS AWS 账户中创建 DXGateway 和 VGW。

如果您有一个现有的 DXGateway 连接到一个或多个现有 VGW，该过程涉及 AWS 帐户上的 Red Hat OpenShift Service，将 Association Proposal 发送到 DXGateway 所有者。DXGateway 所有者必须确保所提议的 CIDR 不会与他们关联的任何其他 VGW 冲突。

详情请查看以下 AWS 文档：

- 虚拟接口
- Direct Connect Gateways
- 在帐户间关联 VGW

重要

当连接到现有的 DXGateway 时，您需要负责相关的成本。

有两个可用的配置选项：

方法 1	创建托管的 VIF，然后创建 DXGateway 和 VGW。
方法 2	通过您拥有的现有的 Direct Connect Gateway 请求连接。

2.4.2. 创建托管的虚拟接口

前提条件

- 收集 Red Hat OpenShift Service on AWS AWS 账户 ID。

2.4.2.1. 确定直接连接连接的类型

查看 Direct Connect Virtual Interface 详情以确定连接的类型。

流程

1. 登陆到 Red Hat OpenShift Service on AWS AWS Account Dashboard 并选择正确的区域。
2. 从 **Services** 菜单选择 **Direct Connect**。
3. 将有一个或多个虚拟接口等待被接受，请选择其中一个来查看 **Summary**。
4. 查看虚拟接口类型：private 或 public。
5. 记录下 **Amazon side ASN** 的值。

如果 Direct Connect Virtual 接口类型是 Private，则会创建一个虚拟专用网关。如果直接连接虚拟接口是公共的，则会创建一个直接连接网关。

2.4.2.2. 创建私有直接连接

如果 Direct Connect Virtual Interface 类型为 Private，则会创建一个 Private Direct Connect Connect。

流程

1. 登陆到 Red Hat OpenShift Service on AWS AWS Account Dashboard 并选择正确的区域。
2. 在 AWS 区域中，从 **Services** 菜单中选择 **VPC**。
3. 从 **VPN Connections** 中选择 **Virtual Private Gateways**。
4. 点 **Create Virtual Private Gateway**。
5. 为 Virtual Private Gateway 指定一个合适的名称。
6. 选择 **Custom ASN**，并输入前面收集的 **Amazon side ASN** 的值。
7. 创建虚拟专用网关.
8. 点新创建的 Virtual Private Gateway，从 **Actions** 选项卡中选择 **Attach to VPC**。
9. 从列表中选择 **Red Hat OpenShift Service on AWS Cluster VPC**，并将 Virtual Private Gateway 附加到 VPC。
10. 在 **Services** 菜单中点 **Direct Connect**。从列表中选择一个直接连接虚拟接口。
11. 确认 **I understand that Direct Connect port charges apply once I click Accept Connection**信息，然后选 **Accept Connection**。
12. 选择 **接受** 虚拟专用网关连接并选择在前面的步骤中创建的虚拟专用网关。
13. 选择 **Accept** 接受连接。
14. 如果有多个虚拟接口，请重复前面的步骤。

2.4.2.3. 创建公共直接连接

如果 Direct Connect Virtual Interface 类型为 Public，则会创建一个公共直接连接。

流程

1. 登陆到 Red Hat OpenShift Service on AWS AWS Account Dashboard 并选择正确的区域。
2. 在 Red Hat OpenShift Service on AWS AWS Account 区域中，选择 **Direct Connect** from the **Services** 菜单。
3. 选择 **Direct Connect Gateways** 和 **Create Direct Connect Gateway**。
4. 为直接连接网关指定合适的名称。
5. 在 **Amazon side ASN** 中，输入前面收集的 Amazon side ASN 值。
6. 创建直接连接网关。
7. 从 **Services** 菜单选择 **Direct Connect**。
8. 从列表中选择一个 Direct Connect Virtual Interfaces。
9. 确认 **I understand that Direct Connect port charges apply once I click Accept Connection**信息，然后选 **Accept Connection**。
10. 选择 **接受** 直接连接网关连接并选择在前面的步骤中创建的直接连接网关。

11. 点 **Accept** 以接受连接。
12. 如果有多个虚拟接口，请重复前面的步骤。

2.4.2.4. 验证虚拟接口

在直接连接虚拟接口被接受后，等待一个简短的时间并查看接口的状态。

流程

1. 登陆到 Red Hat OpenShift Service on AWS AWS Account Dashboard 并选择正确的区域。
2. 在 Red Hat OpenShift Service on AWS AWS Account 区域中，选择 **Direct Connect** from the **Services** 菜单。
3. 从列表中选择一个 Direct Connect Virtual Interfaces。
4. 检查 Interface State 的状态为 **Available**
5. 检查 Interface BGP 状态是否为 **Up**。
6. 对剩余的直接连接接口重复此验证。

在 Direct Connect Virtual Interfaces 可用时，您可以在 AWS 帐户仪表板上登录到 Red Hat OpenShift Service，并下载您的侧配置的直接连接配置文件。

2.4.3. 连接到现有直接连接网关

前提条件

- 确认 AWS VPC 上的 Red Hat OpenShift Service 的 CIDR 范围不会与您关联的任何其他 VGW 冲突。
- 收集以下信息：
  - 直接连接网关 ID。
  - 与虚拟接口关联的 AWS 帐户 ID。
  - 为 DXGateway 分配的 BGP ASN。可选：也可以使用 Amazon 默认 ASN。

流程

1. 登陆到 Red Hat OpenShift Service on AWS AWS Account Dashboard 并选择正确的区域。
2. 在 AWS AWS 帐户区域中的 Red Hat OpenShift Service 中，从 **Services** 菜单中选择 **VPC**。
3. 从 **VPN 连接** 中，选择 **虚拟专用网关**。
4. 选择 **Create Virtual Private Gateway**。
5. 为 Virtual Private Gateway 指定一个合适的名称。
6. 点 **Custom ASN**，然后输入前面收集的 **Amazon side ASN** 值或使用 Amazon Provided ASN。
7. 创建虚拟专用网关.

8. 在 Red Hat OpenShift Service on AWS AWS Account Dashboard 的**导航**框中，选择 **Virtual private 网关** 并选择虚拟私有网关。选择 **View details**。
9. 选择 **Direct Connect 网关关联**，然后点**关联直接连接网关**。
10. 在 **关联** 帐户类型 下，对于帐户所有者，请选择"**另一个帐户**。
11. 对于 **直接连接网关所有者**，请输入拥有直接连接网关的 AWS 帐户的 ID。
12. 在**关联设置**下，用于直接连接网关 ID，输入 Direct Connect 网关的 ID。
13. 在**关联设置**下，对于 Virtual interface owner，输入拥有关联虚拟接口的 AWS 帐户的 ID。
14. 可选：添加前缀作为允许的前缀，使用逗号分隔它们。
15. 选择**关联直接连接网关**。
16. 在发送关联建议后，它会等待您接受的接受。AWS 文档中提供了您必须执行的最终步骤。

2.4.4. 直接连接故障排除

有关进一步的故障排除文档，请参阅故障排除 AWS 直接连接 文档。

第 3 章 集群自动扩展

将自动扩展应用到 Red Hat OpenShift Service on AWS 集群涉及配置集群自动扩展，然后为集群中的至少一个机器池配置机器自动扩展。

重要

您只能在机器 API 正常工作的集群中配置集群自动扩展。

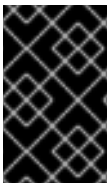
每个集群只能创建一个集群自动扩展。

3.1. 关于集群自动扩展

集群自动扩展会调整 Red Hat OpenShift Service on AWS 集群的大小，以满足其当前的部署需求。它使用 Kubernetes 样式的声明性参数来提供基础架构管理，而且这种管理不依赖于特定云提供商的对象。集群自动控制会在集群范围内有效，不与特定的命名空间相关联。

当由于资源不足而无法在任何当前 worker 节点上调度 pod 时，或者在需要另一个节点来满足部署需求时，集群自动扩展会增加集群的大小。集群自动扩展不会将集群资源增加到超过您指定的限制。

集群自动扩展会计算集群所有节点上的内存和 CPU 总量，即使它不管理 control plane 节点。这些值不是单计算机导向型。它们是整个集群中所有资源的聚合。例如，如果您设置最大内存资源限制，集群自动扩展在计算当前内存用量时包括集群中的所有节点。然后，该计算用于确定集群自动扩展是否具有添加更多 worker 资源的容量。



重要

确保您所创建的 **ClusterAutoscaler** 资源定义中的 **maxNodesTotal** 值足够大，足以满足计算集群中可能的机器总数。此值必须包含 control plane 机器的数量以及可扩展至的机器数量。

每隔 10 秒，集群自动扩展会检查集群中不需要哪些节点，并移除它们。如果满足以下条件，集群自动扩展会考虑要删除的节点：

- 节点使用率低于集群的节点 *利用率级别* 阈值。节点使用率级别是请求的资源的总和，由分配给节点的资源划分。如果您没有在 **ClusterAutoscaler** 自定义资源中指定值，集群自动扩展会使用默认值 **0.5**，它对应于 50% 的利用率。
- 集群自动扩展可以将节点上运行的所有 pod 移到其他节点。Kubernetes 调度程序负责在节点上调度 pod。
- 集群自动扩展没有缩减禁用注解。

如果节点上存在以下类型的 pod，集群自动扩展不会删除该节点：

- 具有限制性 pod 中断预算（PDB）的 Pod。
- 默认不在节点上运行的 Kube 系统 Pod。
- 没有 PDB 或 PDB 限制性太强的 Kube 系统 pod。
- 不受控制器对象支持的 Pod,如部署、副本集或有状态集。
- 具有本地存储的 Pod。

- 因为缺乏资源、节点选择器或关联性不兼容或有匹配的反关联性等原因而无法移至其他位置的 Pod。
- 具有 **"cluster-autoscaler.kubernetes.io/safe-to-evict": "false"** 注解的 Pod，除非同时也具有 **"cluster-autoscaler.kubernetes.io/safe-to-evict": "true"** 注解。

例如，您可以将最大 CPU 限值设置为 64 个内核，并将集群自动扩展配置为每个创建具有 8 个内核的机器。如果您的集群从 30 个内核开始，集群自动扩展可最多添加具有 32 个内核的 4 个节点，共 62 个。

如果配置集群自动扩展，则需要额外的使用限制：

- 不要直接修改位于自动扩展节点组中的节点。同一节点组中的所有节点具有相同的容量和标签，并且运行相同的系统 Pod。
- 指定适合您的 Pod 的请求。
- 如果需要防止 Pod 被过快删除，请配置适当的 PDB。
- 确认您的云提供商配额足够大，能够支持您配置的最大节点池。
- 不要运行其他节点组自动扩展器，特别是云提供商提供的自动扩展器。

pod 横向自动扩展（HPA）和集群自动扩展以不同的方式修改集群资源。HPA 根据当前的 CPU 负载更改部署或副本集的副本数。如果负载增加，HPA 会创建新的副本，不论集群可用的资源量如何。如果没有足够的资源，集群自动扩展会添加资源，以便 HPA 创建的 pod 可以运行。如果负载减少，HPA 会停止一些副本。如果此操作导致某些节点利用率低下或完全为空，集群自动扩展会删除不必要的节点。

集群自动扩展会考虑 pod 优先级。如果集群没有足够的资源，则"Pod 优先级和抢占"功能可根据优先级调度 Pod，但集群自动扩展会确保集群具有运行所有 Pod 需要的资源。为满足这两个功能，集群自动扩展包含一个优先级截止函数。您可以使用此截止函数来调度"尽力而为"的 Pod，它们不会使集群自动扩展增加资源，而是仅在有可用备用资源时运行。

优先级低于截止值的 Pod 不会导致集群扩展或阻止集群缩减。系统不会添加新节点来运行 Pod，并且可能会删除运行这些 Pod 的节点来释放资源。

集群自动扩展支持在其上有机器 API 的平台。

3.2. 使用 OPENSHIFT CLUSTER MANAGER 在集群创建过程中启用自动扩展

您可以在集群创建过程中使用 OpenShift Cluster Manager 来自动扩展。

流程

1. 在集群创建过程中，选中 **Enable autoscaling** 复选框。**Edit cluster autoscaling settings** 按钮变为可选择。
    a. 您还可以选择自动扩展的最小或最大节点数量。
2. 点 **Edit cluster autoscaling settings**。
3. 编辑您想要的任何设置，然后点 **Close**。

3.3. 使用 OPENSHIFT CLUSTER MANAGER 在集群创建后启用自动扩展

您可以使用 OpenShift Cluster Manager 在集群创建后自动扩展。

流程

1. 在 OpenShift Cluster Manager 中，点您要自动扩展的集群的名称。集群的 Overview 页面有一个 **自动扩展** 项，它指示它是否启用或禁用。
2. 点 **Machine Pools** 选项卡。
3. 点 **Edit cluster autoscaling** 按钮。此时会显示 **Edit cluster autoscaling** settings 窗口。
4. 点窗口顶部的 **Autoscale 集群** 切换。现在，所有设置都可以编辑。
5. 编辑您想要的任何设置，然后单击 **Save**。
6. 单击屏幕右上角的 **x**，以关闭设置窗口。

要将所有自动扩展设置恢复到默认值（如果已更改），请单击 **Revert all 为 defaults** 按钮。

3.4. 使用 OPENSHIFT CLUSTER MANAGER 的集群自动扩展设置

表解释了在使用带有 OpenShift Cluster Manager 的集群自动扩展时，所有可配置的 UI 设置。

3.4.1. 常规设置

表 3.1. 使用 OpenShift Cluster Manager 时为集群自动扩展配置常规设置

设置	描述	类型或范围	Default（默认）
log-verbosity	设置自动扩展日志级别。默认值为 1。建议使用 4 级进行调试。级别 6 可实现几乎所有的一切。	**整数**	1
skip-nodes-with-local-storage	如果为 **true**，集群自动扩展永远不会删除带有本地存储的 pod 的节点，如 EmptyDir 或 HostPath。	**布尔值**	true
max-pod-grace-period	为 pod 提供安全终止时间（以秒为单位）。	**整数**	600
max-node-provision-time	集群自动扩展等待置备节点的最长时间。	**字符串**	15m
pod-priority-threshold	允许用户调度 "best-effort" pod，这些 pod 不应该触发集群自动扩展操作。这些 pod 仅在备用资源可用时运行。	**整数**	-10
ignore-daemonsets-utilization	决定集群自动扩展在计算资源利用率时是否忽略守护进程集 pod。	**布尔值**	false

设置	描述	类型或范围	Default（默认）
balance-similar-node-groups	如果为 **true**，则此设置会自动识别具有相同实例类型和同一组标签的节点组，并尝试保持这些节点组的相应大小平衡。	**布尔值**	false
balancing-ignored-labels	这个选项指定集群自动扩展在考虑节点组相似时应该忽略的标签。这个选项不能包含空格。	**数组（字符串）**	格式应该是以逗号分隔的标签列表。

3.4.2. 资源限值

表 3.2. 使用 OpenShift Cluster Manager 时，为集群自动扩展配置资源限制设置

设置	描述	类型或范围	Default（默认）
cores-total-min	集群中的最低内核数。集群自动扩展不会扩展集群小于这个数字。	**object**	0
cores-total-max	集群中内核的最大数量。集群自动扩展不会扩展集群大于这个数字。	**object**	180 * 64 (11520)
memory-total-min	集群中最少的 GB 内存数。集群自动扩展不会扩展集群小于这个数字。	**object**	0
memory-total-max	集群中内存的最大数量。集群自动扩展不会扩展集群大于这个数字。	**object**	180 * 64 * 20 (230400)
max-nodes-total	所有节点组中的节点数。包括所有节点，而不只是自动扩展节点。集群自动扩展不会增加大于这个数字的集群。	**整数**	180
GPU	集群中不同 GPU 的最小和最大数量。集群自动扩展不会缩放集群小于这些数字。	**数组**	格式应是一个用逗号分开的 "{p>:<min>:<max>"。

3.4.3. 缩减配置

表 3.3. 使用 OpenShift Cluster Manager 时配置集群自动扩展设置

设置	描述	类型或范围	Default（默认）
scale-down-enabled	集群自动扩展是否应该缩减集群。	**布尔值**	true
scale-down-utilization-threshold	节点使用率级别被定义为请求的资源的总和（根据容量划分），可考虑节点缩减。	**浮点值**	0.5
scale-down-unneeded-time	在节点有资格缩减前，应该不需要多久。	**字符串**	10m
scale-down-delay-after-add	扩展后扩展该缩减评估的时长。	**字符串**	10m
scale-down-delay-after-delete	删除节点后扩展评估的时长。	**字符串**	0s
scale-down-delay-after-failure	缩减缩减故障后的时长，用于缩减评估恢复。	**字符串**	3m

3.5. 使用带有 ROSA CLI 的互动模式在集群创建过程中启用自动扩展

您可以使用终端的交互模式在集群创建过程中设置集群范围的自动扩展行为。

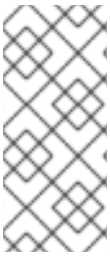
交互模式提供有关可用可配置参数的更多信息。交互模式也执行基本检查和 preflight 验证，这意味着如果提供的值无效，终端会输出一个有效输入的提示。

流程

- 在集群创建过程中，使用 **--enable-autoscaling** 和 **--interactive** 参数启用集群自动扩展：

  Example:

  ```
  $ rosa create cluster --cluster-name <cluster_name> --enable-autoscaling --interactive
  ```



注意

如果您的集群名称超过 15 个字符，它将包含自动生成的域前缀作为您 provisioned 集群 **的子域**。

要自定义子域，请使用 **--domain-prefix** 标志。域前缀不能超过 15 个字符，它必须是唯一的，且在集群创建后无法更改。

出现以下提示时，输入 **y** 以浏览所有可用的自动扩展选项。

互动提示示例：

```
? Configure cluster-autoscaler (optional): [? for help] (y/N) y <enter>
```

3.5.1. 使用带有 ROSA CLI 的互动模式在集群创建后启用自动扩展

您可以使用终端的交互模式在创建集群后设置集群范围的自动扩展行为。

流程

- 创建集群后，输入以下命令：

  Example:

  ```
  $ rosa create autoscaler --cluster=<mycluster> --interactive
  ```

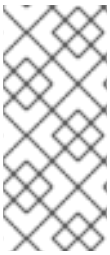
  然后，您可以设置所有可用的自动扩展参数。

3.6. 使用 ROSA CLI 在集群创建过程中启用自动扩展

您可以使用 ROSA CLI (**rosa**)在集群创建过程中设置集群范围的自动扩展行为。您可以在整个机器或只启用集群中启用自动扩展。

流程

- 在集群创建过程中，在集群名称后键入 **--enable autoscaling** 来启用机器自动扩展：



注意

如果您的集群名称超过 15 个字符，它将包含自动生成的域前缀作为您 provisioned 集群 **的子域**。

要自定义子域，请使用 **--domain-prefix** 标志。域前缀不能超过 15 个字符，它必须是唯一的，且在集群创建后无法更改。

Example:

```
$ rosa create cluster --cluster-name <cluster_name> --enable-autoscaling
```

运行以下命令，设置至少一个参数来启用集群自动扩展：

Example:

```
$ rosa create cluster --cluster-name <cluster_name> --enable-autoscaling <parameter>
```

3.6.1. 使用 ROSA CLI 在集群创建后启用自动扩展

您可以使用 ROSA CLI (**rosa**)在集群创建后设置集群范围的自动扩展。

流程

- 创建集群后，创建自动扩展：

  Example:

  ```
  $ rosa create autoscaler --cluster=<mycluster>
  ```

-

a. 您还可以使用以下命令使用特定参数创建自动扩展：

Example:

```
$ rosa create autoscaler --cluster=<mycluster> <parameter>
```

3.6.2. 使用 ROSA CLI 在集群创建后编辑自动扩展

您可以在创建自动扩展后编辑集群自动扩展的任何特定参数。

- 要编辑集群自动扩展，请运行以下命令：

Example:

```
$ rosa edit autoscaler --cluster=<mycluster>
```

a. 要编辑特定的参数，请运行以下命令：

Example:

```
$ rosa edit autoscaler --cluster=<mycluster> <parameter>
```

3.6.3. 使用 ROSA CLI 删除自动扩展

如果不再使用集群自动扩展，您可以删除它。

- 要删除集群自动扩展，请运行以下命令：

Example:

```
$ rosa delete autoscaler --cluster=<mycluster>
```

3.7. 使用 ROSA CLI 的集群自动扩展参数

您可以在集群创建命令中添加以下参数，以便在使用 ROSA CLI (**rosa**)时配置自动扩展参数。

表 3.4. 可以使用 ROSA CLI (rosa)的可配置自动扩展器参数

设置	描述	类型或范围	示例/构建
--autoscaler-balance-similar-node-groups	识别具有相同实例类型和标签的节点组，并尝试平衡这些节点组的相应大小。	**布尔值**	将它添加到 true，省略该选项设置为 false。
--autoscaler-skip-nodes-with-local-storage	如果设置，集群自动扩展不会删除带有本地存储的 pod 的节点，如 EmptyDir 或 HostPath。	**布尔值**	将它添加到 true，省略该选项设置为 false。

设置	描述	类型或范围	示例/构建
--autoscaler-log-verbosity *int*	自动缩放器日志级别。使用您要使用的数字替换命令中的 *int*。	**整数**	**--autoscaler-log-verbosity 4**
--autoscaler-max-pod-grace-period *int*	在缩减前提供 pod 安全终止时间，单位为秒。使用您要使用的秒数替换命令中的 *int*。	**整数**	**--autoscaler-max-pod-grace-period 0**
--autoscaler-pod-priority-threshold *int*	pod 必须超过这个值才能使集群自动扩展部署额外的节点。将 命令中的 *int* 替换为您要使用的数字，可以是负数。	**整数**	**--autoscaler-pod-priority-threshold -10**
--autoscaler-gpu-limit *stringArray*	集群中不同 GPU 的最小和最大数量。集群自动扩展不会缩放集群小于这些数字。格式必须是以逗号分隔的 "{p>,<min>,<max>"。	**数组**	**--autoscaler-gpu-limit nvidia.com/gpu,0,10 --autoscaler-gpu-limit amd.com/gpu,1,5**
--autoscaler-ignore-daemonsets-utilization	如果设置，cluster-autoscaler 会在计算用于缩减资源使用率时忽略守护进程集 pod。	**布尔值**	将它添加到 true，省略该选项设置为 false。
--autoscaler-max-node-provision-time *string*	集群自动扩展等待置备节点的最长时间。使用整数和时间单位 (ns,us,unmarshals,ms,s,m,h)替换命令中的 *字符串*。	**字符串**	**--autoscaler-max-node-provision-time 35m**
--autoscaler-balancing-ignored-labels *strings*	以逗号分隔的标签键列表，在比较节点组时应该忽略这些键。使用相关标签替换命令中的 *字符串*。	**字符串**	**--autoscaler-balancing-ignored-labels topology.ebs.csi.aws.com/zone,alpha.eksctl.io/instance-id**
--autoscaler-max-nodes-total *int*	集群中节点的最大数量，包括自动扩展节点。使用您要使用的数字替换 命令中的 *int*。	**整数**	**--autoscaler-max-nodes-total 180**

设置	描述	类型或范围	示例/构建
--autoscaler-min-cores *int*	在集群中部署的最小内核数。使用您要使用的数字替换 命令中的 *int*。	**整数**	**--autoscaler-min-cores 0**
--autoscaler-max-cores *int*	集群中要部署的最大内核数。使用您要使用的数字替换 命令中的 *int*。	**整数**	**--autoscaler-max-cores 100**
--autoscaler-min-memory *int*	集群中的最小内存量（以 GiB 为单位）。使用您要使用的数字替换 命令中的 *int*。	**整数**	**--autoscaler-min-memory 0**
--autoscaler-max-memory *int*	集群中的最大内存量（以 GiB 为单位）。使用您要使用的数字替换 命令中的 *int*。	**整数**	**--autoscaler-max-memory 4096**
--autoscaler-scale-down-enabled	如果设置，集群自动扩展应该缩减集群。	**布尔值**	将它添加到 true，省略该选项设置为 false。
--autoscaler-scale-down-unneeded-time *string*	在节点有资格缩减前，应该不需要多久。使用整数和时间单位 (ns,us,unmarshals,ms,s,m,h)替换命令中的 *字符串*。	**字符串**	**--autoscaler-scale-down-unneeded-time 1h**
--autoscaler-scale-down-utilization-threshold *float*	节点使用率级别被定义为请求资源的总和，以下将考虑节点进行缩减。值必须在 0 到 1 之间。	**浮点值**	**--autoscaler-scale-down-utilization-threshold 0.5**
--autoscaler-scale-down-delay-after-add *string*	扩展后扩展评估的时长。使用整数和时间单位 (ns,us,unmarshals,ms,s,m,h)替换命令中的 *字符串*。	**字符串**	**--autoscaler-scale-down-delay-after-add 1h**
--autoscaler-scale-down-delay-after-delete *string*	删除节点后缩减评估的时长。使用整数和时间单位 (ns,us,unmarshals,ms,s,m,h)替换命令中的 *字符串*。	**字符串**	**--autoscaler-scale-down-delay-after-delete 1h**

设置	描述	类型或范围	示例/构建
--autoscaler-scale-down-delay-after-failure *string*	缩减故障后缩减评估的时长。使用整数和时间单位(ns,us,unmarshals,ms,s,m,h)替换命令中的 *字符串*。	**字符串**	**--autoscaler-scale-down-delay-after-failure 1h**

第 4 章 使用机器池管理节点

4.1. 关于机器池

Red Hat OpenShift Service on AWS 使用机器池作为云基础架构之上的弹性动态置备方法。

主要资源包括机器、计算机器集和机器池。

4.1.1. Machines

机器是描述 worker 节点主机的基本单元。

4.1.2. 机器集

MachineSet 资源是计算机器组。如果需要更多机器或必须缩减机器，请更改计算机器集所属的机器池中的副本数量。

机器集不能在 ROSA 中直接修改。

4.1.3. 机器池

机器池是计算机器集的更高级别构造。

机器池创建计算机器集，它们是跨可用性区域相同的配置克隆。机器池在 worker 节点上执行所有主机节点置备管理操作。如果需要更多机器或必须缩减机器，请更改机器池中的副本数量以满足您的计算需求。您可以手动配置扩展或设置自动扩展。

默认情况下，集群有一个机器池。在集群安装过程中，您可以定义实例类型或大小，并为这个机器池添加标签。

集群安装后：

- 您可以删除或向任何机器池添加标签。
- 您可以将额外的机器池添加到现有集群中。
- 只要有一个没有污点的机器池，您可以给任何机器池添加污点。
- 只要有一个没有污点的机器池，您可以创建和删除机器池，并为 Single-AZ 集群至少两个副本，或为 Multi-AZ 集群三个副本。

注意

您无法更改机器池节点类型或大小。机器池节点类型或大小仅在创建期间指定。如果需要不同的节点类型或大小，您必须重新创建机器池并指定所需的节点类型或大小值。

- 您可以为每个添加的机器池添加标签。

单个集群中可以存在多个机器池，每个机器池可以包含唯一的节点类型和节点大小配置。

4.1.4. 多个区集群中的机器池

在跨多个可用区(AZ)创建的集群中，可在所有三个 AZ 或您选择的任何单个 AZ 中创建机器池。在集群创建时默认创建的机器池将使用所有三个 AZ 中的机器创建，并以三个 的倍数进行扩展。

如果您创建新的 Multi-AZ 集群，机器池会自动复制到这些区域。默认情况下，如果您将机器池添加到现有的 Multi-AZ 集群，则所有区域中会自动创建新的机器池。

注意

您可以覆盖此默认设置，并在您选择的 Single-AZ 中创建机器池。

同样，删除机器池将从所有区中删除。由于这种多副本的效果，在多AZ 集群中使用机器池可以在创建机器池时为特定区域消耗更多项目的配额。

4.1.5. 使用 HCP 集群的 ROSA 中的机器池

在带有 HCP 集群的 ROSA 中，托管的 control plane 跨越安装的云区域中的三个可用区(AZ)。在带有 HCP 的 ROSA 的每个机器池都会在单个 AZ 内部署单一子网中。每个 AZ 只能有一个机器池。

带有 HCP 集群升级的 ROSA 的每个机器池都独立进行。因为机器池独立升级，所以它们必须保持在托管 control plane 的 2 个次版本中。例如，如果您的托管的 control plane 是 4.16.z，您的机器池必须至少为 4.14.z。

下图描述了机器池在 ROSA 和使用 HCP 集群的 ROSA 中工作：

注意

ROSA with HCP 集群中的机器池每个都独立升级，机器池版本必须保留在 control plane 的两个次版本中。

4.1.6. 其他资源

- 管理计算节点
- 关于自动扩展

- 配置 PID 限制

4.2. 管理计算节点

本文档论述了如何通过 Red Hat OpenShift Service on AWS (ROSA) 管理计算（也称为 worker）节点。

机器池中配置了计算节点的大部分更改。机器池是集群中有相同配置的计算节点的一组计算节点，提供轻松管理。

您可以编辑机器池配置选项，如扩展、添加节点标签和添加污点。

4.2.1. 创建机器池

在 AWS (ROSA)集群上安装 Red Hat OpenShift Service 时会创建一个机器池。安装后，您可以使用 OpenShift Cluster Manager 或 ROSA CLI (**rosa**) 为集群创建额外的机器池。

注意

对于 ROSA CLI **rosa** 版本 1.2.25 及更早版本的用户，与集群创建的机器池被识别为 **默认值**。对于 ROSA CLI **rosa** 版本 1.2.26 及更高版本的用户，在集群中创建的机器池被识别为 **worker**。

4.2.1.1. 使用 OpenShift Cluster Manager 创建机器池

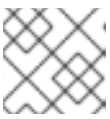
您可以使用 OpenShift Cluster Manager 为您的 Red Hat OpenShift Service on AWS (ROSA) 集群创建额外的机器池。

前提条件

- 您创建了 ROSA 集群。

流程

1. 进入到 OpenShift Cluster Manager 并选择您的集群。
2. 在 **Machine pool** 选项卡下，点 **Add machine pool**。
3. 添加**机器池名称**。
4. 从下拉菜单中选择 **Compute 节点实例类型**。实例类型定义机器池中各个计算节点的 vCPU 和内存分配。



注意

在创建池后，您无法更改机器池的实例类型。

5. 可选：为机器池配置自动扩展：
   a. 选择 **Enable autoscaling** 以自动扩展机器池中的机器数量，以满足部署需求。
   b. 设置自动扩展的最小和最大节点数限值。集群自动扩展不会减少或增加机器池节点数超过您指定的限制。

- 如果您使用一个可用区部署集群，请**设置最小和最大节点数**。这会在可用区中定义最小和最大计算节点限值。
- 如果您使用多个可用区部署集群，请为每个区设置 **Minimum nodes per zone**和 **Maximum nodes per zone**。它定义每个区的最小和最大计算节点限值。

注意

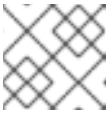
另外，您可以在创建机器池后为机器池设置自动扩展首选项。

6. 如果没有启用自动扩展，请选择计算节点计数：
   - 如果您使用一个可用区部署集群，请从下拉菜单中选择 **Compute 节点数**。这定义了置备到区域的机器池的计算节点数量。
   - 如果您使用多个可用区部署集群，请从下拉菜单中选择 **Compute 节点数（每个区域）**。这定义了每个区要置备到机器池的计算节点数量。
7. 可选：配置 **根磁盘大小**。
8. 可选：为您的机器池添加节点标签和污点：
   a. 展开 **Edit node labels and taints**菜单。
   b. 在 **Node labels** 下，为您的节点标签添加 **Key** 和 **Value** 项。
   c. 在 **Taints** 下，为您的污点添加 **Key** 和 **Value** 条目。

注意

只有集群已至少有一个没有污点的机器池时，才能创建带有污点的机器池。

   d. 对于每个污点，从下拉菜单中选择 **Effect**。可用选项包括 **NoSchedule**、**PreferNoSchedule** 和 **NoExecute**。



注意

另外，您可以在创建机器池后添加节点标签和污点。

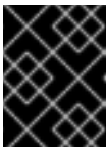
9. 可选：选择用于此机器池中节点的附加自定义安全组。您必须已创建了安全组，并将其与您为这个集群选择的 VPC 关联。您无法在创建机器池后添加或编辑安全组。如需更多信息，请参阅"添加资源"部分中的安全组的要求。

重要

您可以使用最多 10 个额外的安全组用于带有 HCP 集群的 ROSA 上的机器池。

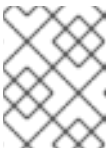
10. 可选：如果要配置机器池将机器部署为非保障的 AWS Spot 实例，请使用 Amazon EC2 Spot 实例：
    a. 选择 **使用 Amazon EC2 Spot 实例**。
    b. 选择 **Use On-Demand 实例价格** 即可使用按需实例价格。或者，选择 **Set maximum price** 来为 Spot 实例定义最大每小时价格。

有关 Amazon EC2 Spot 实例的更多信息，请参阅 AWS 文档。



重要

您的 Amazon EC2 Spot 实例可能随时中断。仅对可容许中断的工作负载使用 Amazon EC2 Spot 实例。



注意

如果为机器池选择了**使用 Amazon EC2 Spot 实例**，则在创建机器池后无法禁用该选项。

11. 点 **Add machine pool** 创建机器池。

验证

- 验证机器池页面中是否存在 **机器池**，配置是如预期的。

其他资源

- 其他自定义安全组

4.2.1.2. 使用 ROSA CLI 创建机器池

您可以使用 ROSA CLI (**rosa**) 为 Red Hat OpenShift Service 在 AWS (ROSA) 集群上创建额外的机器池。

前提条件

- 在您的工作站上安装和配置了最新的 Red Hat OpenShift Service on AWS (ROSA) CLI **rosa**。
- 您可以使用 ROSA CLI (**rosa**)登录到您的红帽帐户。
- 您创建了 ROSA 集群。

流程

- 要添加不使用自动扩展的机器池，请创建机器池，并定义实例类型、计算（也称为 worker）节点数和节点标签：

```
$ rosa create machinepool --cluster=<cluster-name> \
                          --name=<machine_pool_id> \ 1
                          --replicas=<replica_count> \ 2
                          --instance-type=<instance_type> \ 3
                          --labels=<key>=<value>,<key>=<value> \ 4
                          --taints=<key>=<value>:<effect>,<key>=<value>:<effect> \ 5
                          --use-spot-instances \ 6
                          --spot-max-price=0.5 \ 7
                          --disk-size=<disk_size> 8
                          --availability-zone=<availability_zone_name> 9
                          --additional-security-group-ids <sec_group_id> 10
                          --subnet string 11
```

1. 指定机器池的名称。将 **<machine_pool_id>** 替换为机器池的名称。

2. 指定要置备的计算节点数量。如果您使用单一可用区部署了 ROSA，这定义了要置备到区域的机器池的计算节点数量。如果您使用多个可用区部署集群，这会定义在所有区中要置备的计算节点数量，计数必须是 3。在没有配置自动扩展时，需要 **--replicas** 参数。

3. 可选：为您的机器池中的计算节点设置实例类型。实例类型定义池中各个计算节点的 vCPU 和内存分配。将 **<instance_type>** 替换为实例类型。默认值为 **m5.xlarge**。在创建池后，您无法更改机器池的实例类型。

4. 可选：定义机器池的标签。将 **<key>=<value>,<key>=<value>** 替换为以逗号分隔的键-值对，例如 **--labels=key1=value1,key2=value2**。

5. 可选：定义机器池的污点。使用每个污点的实际的 key, value, 和 effect 替换 Replace **<key>=<value>:<effect>,<key>=<value>:<effect>**，例如 **--taints=key1=value1:NoSchedule,key2=value2:NoExecute**。可用效果包括 **NoSchedule**、**PreferNoSchedule** 和 **NoExecute**。

6. 可选：配置机器池以将机器部署为非保障的 AWS Spot 实例。如需更多信息，请参阅 AWS 文档中的 Amazon EC2 Spot 实例。如果为机器池选择了**使用 Amazon EC2 Spot 实例**，则在创建机器池后无法禁用该选项。

7. 可选：如果选择使用 Spot 实例，您可以指定此参数来为 Spot 实例定义最大每小时价格。如果没有指定这个参数，则使用 on-demand 价格。

重要

您的 Amazon EC2 Spot 实例可能随时中断。仅对可容许中断的工作负载使用 Amazon EC2 Spot 实例。

8. 可选：指定 worker 节点磁盘大小。该值可以是 GB、GiB、TB 或 TiB。将 **<disk_size** > 替换为数字值和单位，如 **--disk-size=200GiB**。

9. 可选： 对于 Multi-AZ 集群，您可以在您选择的 Single-AZ 中创建机器池。将 **<az** > 替换为 Single-AZ 名称。

注意

Multi-AZ 集群保留一个 Multi-AZ control plane，且可以在 Single-AZ 或 Multi-AZ 之间具有 worker 机器池。机器池在可用区间平均分配机器（节点）。

警告

如果您使用 Single-AZ 选择 worker 机器池，则无论机器副本数如何，该机器池都没有容错功能。对于容错 worker 机器池，请选择 Multi-AZ 机器池将机器分发到 3 的倍数跨可用区。

- 具有三个可用区的 Multi-AZ 机器池只能有 3 的倍数，如 3、6、9 等。
- 带有一个可用区的 Single-AZ 机器池可以具有 1 的倍数，如 1,2,3,4 等。

10 可选：对于没有红帽受管 VPC 的集群中的机器池，您可以选择在机器池中使用的额外自定义安全组。您必须已创建了安全组，并将其与您为这个集群选择的 VPC 关联。您无法在创建机器池后添加或编辑安全组。如需更多信息，请参阅"添加资源"部分中的安全组的要求。

重要

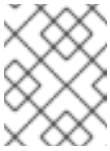
您可以使用最多 10 个额外的安全组用于带有 HCP 集群的 ROSA 上的机器池。

11 可选：对于 BYO VPC 集群，您可以选择创建 Single-AZ 机器池的子网。如果子网没有集群创建子网，则必须有一个带有键 **kubernetes.io/cluster/<infra-id** > 和值 **shared** 的标签。客户可使用以下命令获取 Infra ID：

```
$ rosa describe cluster -c <cluster name>|grep "Infra ID:"
```

输出示例

```
Infra ID:                   mycluster-xqvj7
```



注意

您不能同时设置 **--subnet** 和 **--availability-zone**，仅允许创建 Single-AZ 机器池。

以下示例创建一个名为 **mymachinepool** 的机器池，它使用 **m5.xlarge** 实例类型并具有 2 个计算节点副本。这个示例还添加了 2 个特定于工作负载的标签：

```
$ rosa create machinepool --cluster=mycluster --name=mymachinepool --replicas=2 --
instance-type=m5.xlarge --labels=app=db,tier=backend
```

输出示例

```
I: Machine pool 'mymachinepool' created successfully on cluster 'mycluster'
I: To view all machine pools, run 'rosa list machinepools -c mycluster'
```

- 要添加使用自动扩展的机器池，请创建机器池，并定义自动扩展配置、实例类型和节点标签：

```
$ rosa create machinepool --cluster=<cluster-name> \
                          --name=<machine_pool_id> \ 1
                          --enable-autoscaling \ 2
                          --min-replicas=<minimum_replica_count> \ 3
                          --max-replicas=<maximum_replica_count> \ 4
                          --instance-type=<instance_type> \ 5
                          --labels=<key>=<value>,<key>=<value> \ 6
                          --taints=<key>=<value>:<effect>,<key>=<value>:<effect> \ 7
                          --use-spot-instances \ 8
                          --spot-max-price=0.5 9
                          --availability-zone=<availability_zone_name> 10
```

1. 指定机器池的名称。将 **<machine_pool_id>** 替换为机器池的名称。
2. 在机器池中启用自动扩展来满足部署需求。
3. 4. 定义最小和最大计算节点限值。集群自动扩展不会减少或增加机器池节点数超过您指定的限制。如果您使用单个可用区部署 ROSA，**--min-replicas** 和 **--max-replicas** 参数在区的机器池中定义自动扩展限制。如果您使用多个可用区部署集群，则参数在所有区总数中定义自动扩展限制，计数必须为 3。
5. 可选：为您的机器池中的计算节点设置实例类型。实例类型定义池中各个计算节点的 vCPU 和内存分配。将 **<instance_type>** 替换为实例类型。默认值为 **m5.xlarge**。在创建池后，您无法更改机器池的实例类型。
6. 可选：定义机器池的标签。将 **<key>=<value>,<key>=<value>** 替换为以逗号分隔的键-值对，例如 **--labels=key1=value1,key2=value2**。
7. 可选：定义机器池的污点。使用每个污点的实际的 key, value, 和 effect 替换 Replace **<key>=<value>:<effect>,<key>=<value>:<effect>**，例如 **--taints=key1=value1:NoSchedule,key2=value2:NoExecute**。可用效果包括 **NoSchedule**、**PreferNoSchedule** 和 **NoExecute**。
8. 可选：配置机器池以将机器部署为非保障的 AWS Spot 实例。如需更多信息，请参阅 AWS 文档中的 Amazon EC2 Spot 实例。如果为机器池选择了**使用 Amazon EC2 Spot 实例**，则在创建机器池后无法禁用该选项。

重要

您的 Amazon EC2 Spot 实例可能随时中断。仅对可容许中断的工作负载使用 Amazon EC2 Spot 实例。

9. 可选：如果选择使用 Spot 实例，您可以指定此参数来为 Spot 实例定义最大每小时价格。如果没有指定这个参数，则使用 on-demand 价格。
10. 可选： 对于 Multi-AZ 集群，您可以在您选择的 Single-AZ 中创建机器池。将 **<az** > 替换为 Single-AZ 名称。

以下示例创建一个名为 **mymachinepool** 的机器池，它使用 **m5.xlarge** 实例类型并启用自动扩展。最少的计算节点限值为 3，最大为 6 个。这个示例还添加了 2 个特定于工作负载的标签：

```
$ rosa create machinepool --cluster=mycluster --name=mymachinepool --enable-autoscaling
--min-replicas=3 --max-replicas=6 --instance-type=m5.xlarge --labels=app=db,tier=backend
```

输出示例

```
I: Machine pool 'mymachinepool' created successfully on cluster 'mycluster'
I: To view all machine pools, run 'rosa list machinepools -c mycluster'
```

验证

您可以列出集群中的所有机器池，或描述单独的机器池。

1. 列出集群中的可用机器池：

```
$ rosa list machinepools --cluster=<cluster_name>
```

输出示例

```
ID           AUTOSCALING  REPLICAS  INSTANCE TYPE  LABELS              TAINTS
AVAILABILITY ZONES                SPOT INSTANCES
Default      No           3         m5.xlarge                          us-east-1a, us-east-1b, us-
east-1c    N/A
mymachinepool  Yes        3-6       m5.xlarge      app=db, tier=backend  us-east-1a,
us-east-1b, us-east-1c    No
```

2. 描述集群中特定机器池的信息：

```
$ rosa describe machinepool --cluster=<cluster_name> mymachinepool
```

输出示例

```
ID:                 mymachinepool
Cluster ID:         27iimopsg1mge0m81l0sqivkne2qu6dr
Autoscaling:        Yes
Replicas:           3-6
Instance type:      m5.xlarge
Labels:             app=db, tier=backend
Taints:
Availability zones:  us-east-1a, us-east-1b, us-east-1c
Subnets:
Spot instances:     No
Disk size:          300 GiB
Security Group IDs:
```

3. 验证机器池已包含在输出中，并且配置符合预期。

其他资源

- 其他自定义安全组

4.2.2. 配置机器池磁盘卷

可以配置机器池磁盘大小以提供额外的灵活性。默认磁盘大小为 300 GiB。对于集群版本 4.13 或更早版本，磁盘大小可以至少配置为 128 GiB，最大值为 1 TiB。对于版本 4.14 及更新的版本，磁盘大小可以至少配置为 128 GiB，最多为 16 TiB。

您可以使用 OpenShift Cluster Manager 或 ROSA CLI (**rosa**)为集群配置机器池磁盘大小。

注意

现有集群和机器池节点卷无法调整大小。

重要

默认磁盘大小为 300 GiB。对于集群版本 4.13 或更早版本，磁盘大小可以至少配置为 128 GiB，最大值为 1 TiB。对于版本 4.14 及更新的版本，磁盘大小可以至少配置为 128 GiB，最多为 16 TiB。

4.2.2.1. 使用 OpenShift Cluster Manager 配置机器池磁盘卷

集群创建的先决条件

- 您可以选择在集群安装过程中为默认机器池选择节点磁盘大小。

集群创建的步骤

1. 从 ROSA 集群向导中，进入到 Cluster settings。
2. 导航到 **Machine pool** 步骤。
3. 选择所需的 **Root 磁盘大小**。
4. 选择 **Next** 以继续创建集群。

机器池创建的先决条件

- 您可以选择安装集群后为新机器池选择节点磁盘大小。

机器池创建的步骤

1. 进入到 OpenShift Cluster Manager 并选择您的集群。
2. 导航到 **Machine pool** 选项卡。
3. 点 **Add machine pool**。
4. 选择所需的 **Root 磁盘大小**。
5. 选择 **Add machine pool** 来创建机器池。

4.2.2.2. 使用 ROSA CLI 配置机器池磁盘卷

集群创建的先决条件

- 您可以选择在集群安装过程中为默认机器池选择根磁盘大小。

集群创建的步骤

- 在为所需根磁盘大小创建 OpenShift 集群时运行以下命令：

```
$ rosa create cluster --worker-disk-size=<disk_size>
```

该值可以是 GB、GiB、TB 或 TiB。将 '<disk_size>' 替换为数字值和单元，如 '--worker-disk-size=200GiB'。您不能分隔数字和单位。不允许使用空格。

机器池创建的先决条件

- 您可以选择安装集群后为新机器池选择根磁盘大小。

机器池创建的步骤

1. 执行以下命令来扩展集群：

```
$ rosa create machinepool --cluster=<cluster_id> 1
                          --disk-size=<disk_size> 2
```

1 指定现有 OpenShift 集群的 ID 或名称

2 指定 worker 节点磁盘大小。该值可以是 GB、GiB、TB 或 TiB。将 '<disk_size>' 替换为数字值和单位，如 '--disk-size=200GiB'。您不能分隔数字和单位。不允许使用空格。

2. 登录到 AWS 控制台并找到 EC2 虚拟机根卷大小，以确认新的机器池磁盘大小。

其他资源

- 有关 **rosa create machinepool** 子命令可用的参数的详细列表，请参阅使用 ROSA CLI 管理对象。

4.2.3. 删除机器池

如果您的工作负载要求已更改，当前机器池已不再满足您的需要，则可以删除机器池。

您可以使用 OpenShift Cluster Manager 或 ROSA CLI (**rosa**)删除机器池。

4.2.3.1. 使用 OpenShift Cluster Manager 删除机器池

您可以使用 OpenShift Cluster Manager 删除 Red Hat OpenShift Service on AWS (ROSA)集群的机器池。

前提条件

- 您创建了 ROSA 集群。
- 集群处于 ready 状态。
- 您有一个没有污点的机器池，至少有两个用于单个 AZ 集群或用于多个 AZ 集群的三个实例。

流程

1. 在 OpenShift Cluster Manager 中，进入到 **Clusters** 页面，再选择包含您要删除的机器池的集群。
2. 在所选集群中，选择 **Machine pool** 选项卡。
3. 在 **Machine pool** 选项卡中，点您要删除的机器池的选项菜单  。
4. 点击 Delete。

所选机器池已删除。

4.2.3.2. 使用 ROSA CLI 删除机器池

您可以使用 ROSA CLI 删除 Red Hat OpenShift Service on AWS (ROSA)集群的机器池。

注意

对于 ROSA CLI **rosa** 版本 1.2.25 及更早版本的用户，无法删除与集群中创建的机器池 (ID='Default')。对于 ROSA CLI **rosa** 版本 1.2.26 及更新的版本的用户，只要集群中有一个没有污点的机器池，且至少有一个副本用于一个 Multi-AZ 集群或 Multi-AZ 集群的三个副本就可以删除。

前提条件

- 您创建了 ROSA 集群。
- 集群处于 ready 状态。
- 您有一个没有污点的现有机器池，至少有两个用于 Single-AZ 集群或 Multi-AZ 集群的实例。

流程

1. 在 ROSA CLI 中运行以下命令：

```
$ rosa delete machinepool -c=<cluster_name> <machine_pool_ID>
```

输出示例

```
? Are you sure you want to delete machine pool <machine_pool_ID> on cluster
<cluster_name>? (y/N)
```

2. 输入 'y' 以删除机器池。
所选机器池已删除。

4.2.4. 手动扩展计算节点

如果您还没有为机器池启用自动扩展，您可以手动扩展池中计算（也称为 worker）节点的数量来满足部署需求。

您必须单独扩展每个机器池。

前提条件

- 在您的工作站上安装和配置了最新的 Red Hat OpenShift Service on AWS (ROSA) CLI **rosa**。
- 您可以使用 ROSA CLI (**rosa**)登录到您的红帽帐户。
- 创建一个 Red Hat OpenShift Service on AWS (ROSA) 集群。
- 您有一个现有的机器池。

流程

1. 列出集群中的机器池：

```
$ rosa list machinepools --cluster=<cluster_name>
```

输出示例

```
ID       AUTOSCALING   REPLICAS    INSTANCE TYPE  LABELS    TAINTS
AVAILABILITY ZONES    DISK SIZE   SG IDs
default   No           2          m5.xlarge                  us-east-1a          300GiB     sg-
0e375ff0ec4a6cfa2
mp1       No           2          m5.xlarge                  us-east-1a          300GiB     sg-
0e375ff0ec4a6cfa2
```

2. 增加或减少机器池中的计算节点副本数量：

```
$ rosa edit machinepool --cluster=<cluster_name> \
                        --replicas=<replica_count> \ 1
                        <machine_pool_id> 2
```

1 如果您使用一个可用区部署 Red Hat OpenShift Service on AWS (ROSA)，则副本数定义为区的机器池置备的计算节点数量。如果您使用多个可用区部署集群，则计数定义所有区中机器池中的计算节点总数，且必须是 3 的倍数。

2 将 **<machine_pool_id>** 替换为机器池的 ID，如上一命令的输出中所示。

验证

1. 列出集群中可用的机器池：

```
$ rosa list machinepools --cluster=<cluster_name>
```

输出示例

```
ID       AUTOSCALING   REPLICAS    INSTANCE TYPE  LABELS    TAINTS
AVAILABILITY ZONES    DISK SIZE   SG IDs
default   No           2          m5.xlarge                  us-east-1a          300GiB     sg-
0e375ff0ec4a6cfa2
mp1       No           3          m5.xlarge                  us-east-1a          300GiB     sg-
0e375ff0ec4a6cfa2
```

2. 在上一命令的输出中，验证机器池的计算节点副本数是否如预期。在示例输出中，**mp1** 机器池的计算节点副本数已扩展到 3。

4.2.5. 节点标签

标签是应用于 **Node** 对象的键值对。您可以使用标签来组织一组对象，并控制 pod 的调度。

您可以在集群创建过程中或之后添加标签。标签可以随时修改或更新。

其他资源

- 有关标签的更多信息，请参阅 Kubernetes 标签和选择器概述。

4.2.5.1. 在机器池中添加节点标签

随时为计算（也称为 worker）节点添加或编辑标签，以便以与您的相关方式管理节点。例如，您可以将工作负载的类型分配给特定的节点。

标签以一个键值对的形式进行分配。对于其分配到的对象，每个键需要是唯一的。

前提条件

- 在您的工作站上安装和配置了最新的 Red Hat OpenShift Service on AWS (ROSA) CLI **rosa**。
- 您可以使用 ROSA CLI (**rosa**)登录到您的红帽帐户。
- 创建一个 Red Hat OpenShift Service on AWS (ROSA) 集群。
- 您有一个现有的机器池。

流程

1. 列出集群中的机器池：

```
$ rosa list machinepools --cluster=<cluster_name>
```

输出示例

```
ID          AUTOSCALING  REPLICAS  INSTANCE TYPE  LABELS    TAINTS
AVAILABILITY ZONES    SPOT INSTANCES
Default     No           2         m5.xlarge                  us-east-1a          N/A
db-nodes-mp  No          2         m5.xlarge                  us-east-1a          No
```

2. 为机器池添加或更新节点标签：

   - 要为不使用自动扩展的机器池添加或更新节点标签，请运行以下命令：

```
$ rosa edit machinepool --cluster=<cluster_name> \
                        --replicas=<replica_count> \ 1
                        --labels=<key>=<value>,<key>=<value> \ 2
                        <machine_pool_id>
```

1. 对于不使用自动扩展的机器池，必须在添加节点标签时提供副本数。如果没有指定 **--replicas** 参数，则在命令完成前会提示您输入副本数。如果您使用一个可用区部署 Red Hat OpenShift Service on AWS (ROSA)，则副本数定义为区的机器池置备的计算节点数量。如果您使用多个可用区部署集群，则计数定义所有区中机器池中的计算节点总数，且必须是 3 的倍数。

2 将 **<key>=<value>,<key>=<value>** 替换为以逗号分隔的键-值对，例如 **--labels=key1=value1,key2=value2**。此列表会持续覆盖对节点标签所做的任何修改。

以下示例将标签添加到 **db-nodes-mp** 机器池：

```
$ rosa edit machinepool --cluster=mycluster --replicas=2 --labels=app=db,tier=backend
db-nodes-mp
```

输出示例

```
I: Updated machine pool 'db-nodes-mp' on cluster 'mycluster'
```

- 要为使用自动扩展的机器池添加或更新节点标签，请运行以下命令：

```
$ rosa edit machinepool --cluster=<cluster_name> \
                        --min-replicas=<minimum_replica_count> \ 1
                        --max-replicas=<maximum_replica_count> \ 2
                        --labels=<key>=<value>,<key>=<value> \ 3
                        <machine_pool_id>
```

1 2 对于使用自动扩展的机器池，您必须提供最小和最大计算节点副本限制。如果没有指定参数，则在命令完成前会提示您输入值。集群自动扩展不会减少或增加机器池节点数超过您指定的限制。如果您使用单个可用区部署 ROSA，**--min-replicas** 和 **--max-replicas** 参数在区的机器池中定义自动扩展限制。如果您使用多个可用区部署集群，则参数在所有区总数中定义自动扩展限制，计数必须为 3。

3 将 **<key>=<value>,<key>=<value>** 替换为以逗号分隔的键-值对，例如 **--labels=key1=value1,key2=value2**。此列表会持续覆盖对节点标签所做的任何修改。

以下示例将标签添加到 **db-nodes-mp** 机器池：

```
$ rosa edit machinepool --cluster=mycluster --min-replicas=2 --max-replicas=3 --
labels=app=db,tier=backend db-nodes-mp
```

输出示例

```
I: Updated machine pool 'db-nodes-mp' on cluster 'mycluster'
```

验证

1. 使用新标签描述机器池的详情：

```
$ rosa describe machinepool --cluster=<cluster_name> <machine-pool-name>
```

输出示例

```
ID:                 db-nodes-mp
Cluster ID:           <ID_of_cluster>
Autoscaling:           No
Replicas:           2
Instance type:          m5.xlarge
```

```
Labels:                 app=db, tier=backend
Taints:
Availability zones:     us-east-1a
Subnets:
Spot instances:         No
Disk size:              300 GiB
Security Group IDs:
```

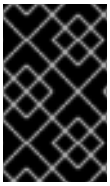
2. 验证您的机器池在输出中是否包含这些标签。

4.2.6. 为机器池添加标签

您可以在机器池中为计算节点添加标签（也称为 worker 节点），为置备机器池时生成的 AWS 资源引入自定义用户标签。

4.2.6.1. 使用 ROSA CLI 为机器池添加标签

您可以使用 ROSA 命令行界面(CLI)为 Red Hat OpenShift Service on AWS 集群添加标签。



重要

您必须确保标签键不是 **aws**、**red-hat-managed**、**red-hat-clustertype** 或 **Name**。另外，您不能设置以 **kubernetes.io/cluster/** 开头的标签键。标签的键不能超过 128 个字符，而标签的值不能超过 256 个字符。红帽保留在以后添加其他保留标签的权利。

前提条件

- 在您的工作站上安装和配置了最新的 AWS (**aws**)、ROSA (**rosa**) 和 OpenShift (**oc**) CLI。
- 您可以使用 **rosa** CLI 登录到您的红帽帐户。
- 创建一个 Red Hat OpenShift Service on AWS (ROSA) 集群。

流程

- 运行以下命令，创建带有自定义标签的机器池：

```
$ rosa create machinepools --cluster=<name> --replicas=<replica_count> \
    --name <mp_name> --tags='<key> <value>,<key> <value>' \ 1
```

1 将 **<key> <value>,<key&gt**; <value> 替换为每个标签的键和值。

输出示例

```
$ rosa create machinepools --cluster=mycluster --replicas 2 --tags='tagkey1
tagvalue1,tagkey2 tagvaluev2'

I: Checking available instance types for machine pool 'mp-1'
I: Machine pool 'mp-1' created successfully on cluster 'mycluster'
I: To view the machine pool details, run 'rosa describe machinepool --cluster mycluster --
machinepool mp-1'
I: To view all machine pools, run 'rosa list machinepools --cluster mycluster'
```

验证

- 使用 **describe** 命令查看带有标签的机器池的详情，并验证输出中您的机器池是否包含标签：

```
$ rosa describe machinepool --cluster=<cluster_name> <machine_pool_name>
```

输出示例

```
$ rosa describe machinepool --cluster classic-rosa --machinepool mp-1

ID:                          mp-1
Cluster ID:                  2baiirqa2141oreotoivp4sipq84vp5g
Autoscaling:                 No
Replicas:                    2
Instance type:               m5.xlarge
Labels:
Taints:
Availability zones:          us-east-1a
Subnets:
Spot instances:              No
Disk size:                   300 GiB
Additional Security Group IDs:
Tags:                        red-hat-clustertype=rosa, red-hat-managed=true,
tagkey1=tagvalue1, tagkey2=tagvaluev2
```

4.2.7. 为机器池添加污点

您可以为机器池中的计算（也称为 worker）节点添加污点，以控制哪些 pod 调度到它们。将污点应用到机器池时，调度程序无法将 pod 放置到池中节点上，除非 pod 规格包含污点的容限。可以使用 OpenShift Cluster Manager 或 Red Hat OpenShift Service on AWS (ROSA) CLI (**rosa**)将污点添加到机器池中。

注意

集群必须至少有一个不包含任何污点的机器池。

4.2.7.1. 使用 OpenShift Cluster Manager 为机器池添加污点

您可以使用 OpenShift Cluster Manager 为 Red Hat OpenShift Service on AWS (ROSA)集群添加污点。

前提条件

- 您在 AWS (ROSA)集群上创建了 Red Hat OpenShift Service。
- 您有一个现有的机器池，它不包含任何污点，至少包含两个实例。

流程

1. 进入到 OpenShift Cluster Manager 并选择您的集群。
2. 在 **Machine pool** 选项卡中，点您要向其添加污点的机器池的选项菜单  。

3. 选择 **Edit taint**。
4. 为您的污点添加 **Key** 和 **Value** 条目。
5. 从下拉菜单中选择污点的 **Effect**。可用选项包括 **NoSchedule**、**PreferNoSchedule** 和 **NoExecute**。
6. 可选： 如果要向机器池添加更多污点，请选择 **Add taint**。
7. 点 **Save** 将污点应用到机器池。

验证

1. 在 **Machine pool** 选项卡中，选择机器池旁边的 **>** 来扩展视图。
2. 验证您的污点是否在展开的视图中的 **Taints** 下列出。

4.2.7.2. 使用 ROSA CLI 为机器池添加污点

您可以使用 ROSA CLI 将污点添加到 Red Hat OpenShift Service on AWS (ROSA)集群的机器池中。

注意

对于 ROSA CLI **rosa** 版本 1.2.25 及之前的版本的用户，在与集群创建的机器池 (ID=**Default**)中无法更改污点数量。对于 ROSA CLI **rosa** 版本 1.2.26 及更高版本的用户，可以在与集群一同创建的机器池中更改污点数量(ID=**worker**)。必须至少有一个没有污点的机器池，以及 Single-AZ 集群至少有两个副本，或 Multi-AZ 集群的三个副本。

前提条件

- 在您的工作站上安装和配置了最新的 AWS (**aws**)、ROSA (**rosa**) 和 OpenShift (**oc**) CLI。
- 您可以使用 **rosa** CLI 登录到您的红帽帐户。
- 创建一个 Red Hat OpenShift Service on AWS (ROSA) 集群。
- 您有一个现有的机器池，它不包含任何污点，至少包含两个实例。

流程

1. 运行以下命令列出集群中的机器池：

```
$ rosa list machinepools --cluster=<cluster_name>
```

输出示例

```
ID          AUTOSCALING  REPLICAS  INSTANCE TYPE  LABELS    TAINTS
AVAILABILITY ZONES    SPOT INSTANCES     DISK SIZE   SG IDs
Default     No           2         m5.xlarge                 us-east-1a          N/A              300 GiB
sg-0e375ff0ec4a6cfa2
db-nodes-mp  No           2         m5.xlarge                 us-east-1a          No               300
GiB     sg-0e375ff0ec4a6cfa2
```

2. 为机器池添加或更新污点：

- 要为不使用自动扩展的机器池添加或更新污点，请运行以下命令：

```
$ rosa edit machinepool --cluster=<cluster_name> \
                        --replicas=<replica_count> \ 1
                        --taints=<key>=<value>:<effect>,<key>=<value>:<effect> \ 2
                        <machine_pool_id>
```

1. 对于不使用自动扩展的机器池，必须在添加污点时提供副本数。如果没有指定 **--replicas** 参数，则在命令完成前会提示您输入副本数。如果您使用一个可用区部署 Red Hat OpenShift Service on AWS (ROSA)，则副本数定义为区的机器池置备的计算节点数量。如果您使用多个可用区部署集群，则计数定义所有区中机器池中的计算节点总数，且必须是 3 的倍数。

2. 使用每个污点的实际的 key, value, 和 effect 替换 Replace **<key>=<value>:<effect>, <key>=<value>:<effect>**，例如 **--taints=key1=value1:NoSchedule,key2=value2:NoExecute**。可用效果包括 **NoSchedule**、**PreferNoSchedule** 和 **NoExecute**。此列表会持续覆盖对节点污点所做的任何修改。

以下示例将污点添加到 **db-nodes-mp** 机器池：

```
$ rosa edit machinepool --cluster=mycluster --replicas 2 --
taints=key1=value1:NoSchedule,key2=value2:NoExecute db-nodes-mp
```

输出示例

```
I: Updated machine pool 'db-nodes-mp' on cluster 'mycluster'
```

- 要为使用自动扩展的机器池添加或更新污点，请运行以下命令：

```
$ rosa edit machinepool --cluster=<cluster_name> \
                        --min-replicas=<minimum_replica_count> \ 1
                        --max-replicas=<maximum_replica_count> \ 2
                        --taints=<key>=<value>:<effect>,<key>=<value>:<effect> \ 3
                        <machine_pool_id>
```

1 2. 对于使用自动扩展的机器池，您必须提供最小和最大计算节点副本限制。如果没有指定参数，则在命令完成前会提示您输入值。集群自动扩展不会减少或增加机器池节点数超过您指定的限制。如果您使用单个可用区部署 ROSA，**--min-replicas** 和 **--max-replicas** 参数在区的机器池中定义自动扩展限制。如果您使用多个可用区部署集群，则参数在所有区总数中定义自动扩展限制，计数必须为 3。

3. 使用每个污点的实际的 key, value, 和 effect 替换 Replace **<key>=<value>:<effect>, <key>=<value>:<effect>**，例如 **--taints=key1=value1:NoSchedule,key2=value2:NoExecute**。可用效果包括 **NoSchedule**、**PreferNoSchedule** 和 **NoExecute**。此列表会持续覆盖对节点污点所做的任何修改。

以下示例将污点添加到 **db-nodes-mp** 机器池：

```
$ rosa edit machinepool --cluster=mycluster --min-replicas=2 --max-replicas=3 --
taints=key1=value1:NoSchedule,key2=value2:NoExecute db-nodes-mp
```

输出示例

```
I: Updated machine pool 'db-nodes-mp' on cluster 'mycluster'
```

验证

1. 描述带有新污点的机器池的详情：

```
$ rosa describe machinepool --cluster=<cluster_name> <machine-pool-name>
```

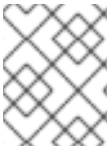
输出示例

```
ID:                 db-nodes-mp
Cluster ID:         <ID_of_cluster>
Autoscaling:        No
Replicas:           2
Instance type:      m5.xlarge
Labels:
Taints:             key1=value1:NoSchedule, key2=value2:NoExecute
Availability zones: us-east-1a
Subnets:
Spot instances:     No
Disk size:          300 GiB
Security Group IDs:
```

2. 验证您的机器池是否包含在输出中。

4.2.8. 在机器池中添加节点调整

您可以为机器池中的计算（也称为 worker）节点添加调整，以控制其在带有托管 control plane (HCP)集群的 Red Hat OpenShift Service on AWS (ROSA)上的配置。



注意

这个功能只在带有托管的 control plane (HCP)集群的 Red Hat OpenShift Service on AWS (ROSA)上被支持。

前提条件

- 在您的工作站上安装和配置了最新的 Red Hat OpenShift Service on AWS (ROSA) CLI **rosa**。
- 您可以使用 ROSA CLI 登录到您的红帽帐户。
- 已使用托管的 control plane (HCP)集群创建了 Red Hat OpenShift Service on AWS (ROSA)。
- 您有一个现有的机器池。
- 您有一个现有的调优配置。

流程

1. 列出集群中的所有机器池：

```
$ rosa list machinepools --cluster=<cluster_name>
```

输出示例

```
ID          AUTOSCALING  REPLICAS  INSTANCE TYPE [...] AVAILABILITY ZONES
SUBNET  VERSION  AUTOREPAIR  TUNING CONFIGS
workers     No           2         m5.xlarge    [...] us-east-1a          N/A     4.12.14  Yes
db-nodes-mp  No          2         m5.xlarge    [...] us-east-1a          No      4.12.14  Yes
```

2. 您可以将调优配置添加到现有或新机器池中。

   a. 在创建机器池时添加调整：

   ```
   $ rosa create machinepool -c <cluster-name> <machinepoolname> --tuning-configs
   <tuning_config_name>
   ```

   输出示例

   ```
   ? Tuning configs: sample-tuning
   I: Machine pool 'db-nodes-mp' created successfully on hosted cluster 'sample-cluster'
   I: To view all machine pools, run 'rosa list machinepools -c sample-cluster'
   ```

   b. 为机器池添加或更新调整：

   ```
   $ rosa edit machinepool -c <cluster-name> <machinepoolname> --tuning-configs
   <tuningconfigname>
   ```

   输出示例

   ```
   I: Updated machine pool 'db-nodes-mp' on cluster 'mycluster'
   ```

验证

1. 列出集群中可用的机器池：

```
$ rosa list machinepools --cluster=<cluster_name>
```

输出示例

```
ID          AUTOSCALING  REPLICAS  INSTANCE TYPE [...] AVAILABILITY ZONES
SUBNET  VERSION  AUTOREPAIR  TUNING CONFIGS
workers     No           2         m5.xlarge    [...] us-east-1a          N/A     4.12.14  Yes
db-nodes-mp  No          2         m5.xlarge    [...] us-east-1a          No      4.12.14  Yes
sample-tuning
```

2. 验证您的机器池是否包含在输出中。

4.2.9. 使用 HCP 集群在 ROSA 中配置节点排空宽限期

您可以为集群中的机器池配置节点排空宽限期。机器池的节点排空宽限期是集群在升级或替换机器池时遵守 Pod Disruption Budget 保护的工作负载的时间。在这个宽限期后，所有剩余的工作负载都会强制被驱

除。节点排空宽限期的值范围从 **0** 到 **1 周**。使用默认值 **0** 或空值时，机器池会排空而无需任何时间限制，直到完成为止。

前提条件

- 在您的工作站上安装和配置了最新的 Red Hat OpenShift Service on AWS (ROSA) CLI **rosa**。
- 已使用托管的 control plane (HCP)集群创建了 Red Hat OpenShift Service on AWS (ROSA)。
- 您有一个现有的机器池。

流程

1. 运行以下命令，列出集群中的所有机器池：

```
$ rosa list machinepools --cluster=<cluster_name>
```

输出示例

```
ID          AUTOSCALING  REPLICAS  INSTANCE TYPE [...] AVAILABILITY ZONES
SUBNET  VERSION  AUTOREPAIR  TUNING CONFIGS
workers     No           2         m5.xlarge    [...] us-east-1a         N/A     4.14.18  Yes
db-nodes-mp  No          2         m5.xlarge    [...] us-east-1a         No      4.14.18  Yes
```

2. 运行以下命令，检查机器池的节点排空宽限期：

```
$ rosa describe machinepool --cluster <cluster_name> <machinepool_name>
```

输出示例

```
ID:                          workers
Cluster ID:                  2a90jdl0i4p9r9k9956v5ocv40se1kqs
Node drain grace period:     (1)
```

(1) 如果这个值为空，机器池会排空而不有时间限制，直到完成为止。

3. 可选：运行以下命令来为机器池更新节点排空宽限期：

```
$ rosa edit machinepool --node-drain-grace-period="<node_drain_grace_period_value>" --
cluster=<cluster_name>  <machinepool_name>
```

注意

在机器池升级过程中更改节点排空宽限期适用于将来的升级，而不是进行升级。

验证

1. 运行以下命令，检查机器池的节点排空宽限期：

```
$ rosa describe machinepool --cluster <cluster_name> <machinepool_name>
```

输出示例

```
ID:                         workers
Cluster ID:                 2a90jdl0i4p9r9k9956v5ocv40se1kqs
Node drain grace period:    30 minutes
```

2. 在输出中验证您的机器池的正确 节**点排空宽限期**。

4.2.10. 其他资源

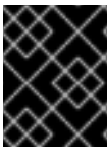
- 关于机器池
- 关于自动扩展
- 启用自动扩展
- 禁用自动扩展
- ROSA 服务定义

4.3. 在本地区域中配置机器池

本文档论述了如何在使用 AWS (ROSA)上的 Red Hat OpenShift Service 的机器池中配置 Local Zones。

4.3.1. 在本地区域中配置机器池

使用以下步骤在 Local Zones 中配置机器池。



重要

Red Hat OpenShift Service on AWS 4.12 支持 AWS Local Zones。有关如何启用本地区域的详情，请参阅红帽知识库文章。

前提条件

- Red Hat OpenShift Service on AWS (ROSA)通常在选择的父区域中提供。请参阅 AWS 通用可用位置列表，以确定特定 AWS 区域可用的 Local Zone。
- ROSA 集群最初以现有 Amazon VPC (BYO-VPC)构建。
- ROSA 集群的最大传输单元(MTU)设置为 1200。

重要

通常，本地区中的 Amazon EC2 实例和 Region 中的 Amazon EC2 实例之间的最大传输单元(MTU)为 1300。请参阅 AWS 文档中的 Local Zones 的工作原理。对于开销，集群网络 MTU 总是小于 EC2 MTU。具体开销由您的网络插件决定，例如：

- OVN-Kubernetes: **100 字节**
- OpenShift SDN: **50 字节**

网络插件可以提供可能也会减少 MTU 的额外功能。查看文档以了解更多信息。

- AWS 帐户启用了 Local Zones。
- AWS 帐户具有与集群相同的 VPC 的 Local Zone 子网。
- AWS 帐户有一个与路由表关联的子网，该路由表具有到 NAT 网关的路由。
- AWS 帐户在相关子网上具有标签 'kubernetes.io/cluster/<infra_id>: shared'。

流程

1. 运行以下 ROSA CLI (**rosa**)命令，在集群中创建机器池。

```
$ rosa create machinepool -c <cluster-name> -i
```

2. 在 ROSA CLI 中为机器池添加子网和实例类型。几分钟后，集群将置备节点。

```
I: Enabling interactive mode (1)
? Machine pool name: xx-lz-xx (2)
? Create multi-AZ machine pool: No (3)
? Select subnet for a single AZ machine pool (optional): Yes (4)
? Subnet ID: subnet-<a> (region-info) (5)
? Enable autoscaling (optional): No (6)
? Replicas: 2 (7)
I: Fetching instance types (8)
? disk-size (optional): (9)
```

(1) 启用交互模式。

(2) 将机器池命名为。这仅限于字母数字，最大长度为 30 个字符。

(3) 将这个选项设置为 no。

(4) 将这个选项设置为 yes。

(5) 从列表中选择子网 ID。

(6) 选择 yes 来启用自动扩展或 no 来禁用自动扩展。

(7) 为机器池选择机器数量。这个数字可以从 1 到 180 的任意位置。

(8) 从列表中选择实例类型。只有所选 Local Zone 支持的实例类型才会出现。

(9) 可选：指定 worker 节点磁盘大小。该值可以是 GB、GiB、TB 或 TiB。设置一个数字值和单位，如 '200GiB'。您不能分隔数字和单位。不允许使用空格。

3. 提供子网 ID 以在 Local Zone 中置备机器池。

有关正式发布和宣布的 AWS Local Zone 位置，请参阅 AWS 上的 AWS Local Zones 位置列表。

4.4. 关于集群中的自动扩展节点

自动扩展器选项可以配置为自动扩展集群中的机器数量。

当由于资源不足而无法在任何当前节点上调度 Pod 时，或者在需要另一个节点来满足部署需求时，集群自动扩展会增加集群的大小。集群自动扩展不会将集群资源增加到超过您指定的限制。

另外，如果相当长的一段时间内都不需要某些节点，例如集群资源使用率较低并且所有重要 Pod 都可以安置在其他节点上时，集群自动扩展会减小集群的大小。

启用自动扩展时，还必须设置 worker 节点的最小和最大数量。

注意

只有集群所有者和机构管理员才可以扩展或删除集群。

4.4.1. 在集群中启用自动扩展节点

您可以通过编辑现有集群的机器池定义，在 worker 节点上启用自动扩展来增加或减少可用节点的数量。

使用 Red Hat OpenShift Cluster Manager 在现有集群中启用自动扩展节点
从 OpenShift Cluster Manager 控制台，为机器池定义中启用 worker 节点的自动扩展。

流程

1. 在 OpenShift Cluster Manager 中，进入到 **Clusters** 页面，再选择您要为其启用自动扩展的集群。
2. 在所选集群中，选择 **Machine pool** 选项卡。
3. 点您要为其启用自动扩展的机器池的末尾的 Options 菜单  ，然后选择 **Scale**。
4. 在 **Edit node count** 对话框中，选中 **Enable autoscaling** 复选框。
5. 选择 **Apply** 以保存这些更改并为集群启用自动扩展。

注意

另外，在使用交互模式 创建集群时，您可以在默认机器池上配置自动扩展。

使用 ROSA CLI 在现有集群中启用自动扩展节点
配置自动扩展，以根据负载动态扩展 worker 节点的数量。

成功自动扩展取决于您的 AWS 帐户中的正确的 AWS 资源配额。验证来自 AWS 控制台的资源配额和请求配额的增加。

流程

1. 要识别集群中的机器池 ID，请输入以下命令：

```
$ rosa list machinepools --cluster=<cluster_name>
```

输出示例

```
ID       AUTOSCALING   REPLICAS    INSTANCE TYPE  LABELS    TAINTS
AVAILABILITY ZONES    DISK SIZE   SG IDs
```

```
default   No        2        m5.xlarge                  us-east-1a          300GiB     sg-
0e375ff0ec4a6cfa2
mp1       No        2        m5.xlarge                  us-east-1a          300GiB     sg-
0e375ff0ec4a6cfa2
```

2. 获取您要配置的机器池的 ID。

3. 要在机器池上启用自动扩展，请输入以下命令：

```
$ rosa edit machinepool --cluster=<cluster_name> <machinepool_ID> --enable-autoscaling -
-min-replicas=<number> --max-replicas=<number>
```

示例

在名为 **mycluster** 的集群上 ID 为 **mp1** 的机器池上启用自动扩展，其副本数为 2 到 5 个 worker 节点：

```
$ rosa edit machinepool --cluster=mycluster mp1 --enable-autoscaling --min-replicas=2 --
max-replicas=5
```

4.4.2. 禁用集群中的自动扩展节点

您可以通过编辑现有集群的机器池定义，在 worker 节点上禁用自动扩展来增加或减少可用节点的数量。

您可以使用 OpenShift Cluster Manager 控制台或 Red Hat OpenShift Service on AWS CLI 禁用集群的自动扩展功能。

注意

另外，在使用交互模式 创建集群时，您可以在默认机器池上配置自动扩展。

使用 Red Hat OpenShift Cluster Manager 禁用现有集群中的自动扩展节点
从 OpenShift Cluster Manager 控制台，禁用机器池定义中的 worker 节点的自动扩展。

流程

1. 在 OpenShift Cluster Manager 中，进入到 **Clusters** 页面，再选择必须禁用自动扩展的集群。
2. 在所选集群中，选择 **Machine pool** 选项卡。
3. 点启动了自动扩展的机器池的末尾的 Options 菜单  ，然后选择 **Scale**。
4. 在"编辑节点数"对话框中，取消选择**启用自动扩展**复选框。
5. 选择 **Apply** 以保存这些更改并从集群中禁用自动扩展。

使用 ROSA CLI 禁用现有集群中的自动扩展节点
使用 Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**，禁用机器池定义中的 worker 节点的自动扩展。

流程

1. 输入以下命令：

```
$ rosa edit machinepool --cluster=<cluster_name> <machinepool_ID> --enable-
autoscaling=false --replicas=<number>
```

示例

在名为 **mycluster** 的集群上，禁用 **default** 机器池上的自动扩展。

```
$ rosa edit machinepool --cluster=mycluster default --enable-autoscaling=false --replicas=3
```

4.4.3. 其他资源

- 故障排除： 自动扩展不会缩减节点
- 关于 machinepools
- 管理计算节点
- 使用 ROSA CLI 管理对象

4.5. 配置集群内存以满足容器内存和风险要求

作为集群管理员，您可以通过以下方式管理应用程序内存，从而帮助集群有效运作：

- 确定容器化应用程序组件的内存和风险要求，并配置容器内存参数以满足这些要求。
- 配置容器化应用程序运行时（如 OpenJDK），以最佳的方式遵守配置的容器内存参数。
- 诊断并解决与在容器中运行相关的内存错误情况。

4.5.1. 了解管理应用程序内存

在继续操作前，建议完全阅读 Red Hat OpenShift Service on AWS 如何管理计算资源的概述。

对于每种资源（内存、CPU、存储），Red Hat OpenShift Service on AWS 允许在 pod 中的每个容器上放置可选的 **请求和限制**值 。

注意以下关于内存请求和内存限制的信息：

- **内存请求**
  - 如果指定，内存请求值会影响 Red Hat OpenShift Service on AWS 调度程序。将容器调度到节点时，调度程序会考虑内存请求，然后在所选节点上隔离出请求的内存供该容器使用。
  - 如果节点的内存已用尽，Red Hat OpenShift Service on AWS 会优先驱除其内存用量超过其内存请求的容器。在严重的内存耗尽情形中，节点 OOM 终止程序可以根据类似的指标选择并终止容器中的一个进程。
  - 集群管理员可以分配配额，或者分配内存请求值的默认值。
  - 集群管理员可以覆盖开发人员指定的内存请求值，以便管理集群过量使用。
- **内存限制**

- 如果指定，内存限制值针对可在容器中所有进程间分配的内存提供硬性限制。
- 如果分配给容器中所有进程的内存超过内存限制，则节点超出内存（OOM）终止程序将立即选择并终止容器中的一个进程。
- 如果同时指定了内存请求和限制，则内存限制必须大于或等于内存请求量。
- 集群管理员可以分配配额，或者分配内存限制值的默认值。
- 最小内存限值为 12MB。如果容器因为一个 **Cannot allocate memory** pod 事件启动失败，这代表内存限制太低。增加或删除内存限制。删除限制可让 pod 消耗无限的节点资源。

4.5.1.1. 管理应用程序内存策略

在 Red Hat OpenShift Service on AWS 上调整应用程序内存大小的步骤如下：

1. **确定预期的容器内存用量**
   从经验判断（例如，通过独立的负载测试），根据需要确定容器内存用量的预期平均值和峰值。需要考虑容器中有可能并行运行的所有进程：例如，主应用程序是否生成任何辅助脚本？

2. **确定风险嗜好**
   确定用于驱除的风险嗜好。如果风险嗜好较低，则容器应根据预期的峰值用量加上一个安全裕度百分比来请求内存。如果风险嗜好较高，那么根据预期的平均用量请求内存可能更为妥当。

3. **设定容器内存请求**
   根据以上所述设定容器内存请求。请求越能准确表示应用程序内存用量越好。如果请求过高，集群和配额用量效率低下。如果请求过低，应用程序驱除的几率就会提高。

4. **根据需要设定容器内存限制**
   在必要时，设定容器内存限制。如果容器中所有进程的总内存用量超过限制，那么设置限制会立即终止容器进程，所以这既有利也有弊。一方面，可能会导致过早出现意料之外的过量内存使用（“快速失败”）；另一方面，也会突然终止进程。

   请注意，一些 Red Hat OpenShift Service on AWS 集群可能需要设置限制值；有些应用程序镜像可能会根据限制覆盖请求；有些应用程序镜像依赖于设置的限制，因为这比请求值更容易检测。

   如果设置内存限制，其大小不应小于预期峰值容器内存用量加上安全裕度百分比。

5. **确保应用程序经过性能优化**
   在适当时，确保应用程序已根据配置的请求和限制进行了性能优化。对于池化内存的应用程序（如 JVM），这一步尤为相关。本页的其余部分将介绍这方面的内容。

4.5.2. 了解 Red Hat OpenShift Service on AWS 的 OpenJDK 设置

默认的 OpenJDK 设置在容器化环境中效果不佳。因此在容器中运行 OpenJDK 时，务必要提供一些额外的 Java 内存设置。

JVM 内存布局比较复杂，并且视版本而异，因此本文不做详细讨论。但作为在容器中运行 OpenJDK 的起点，至少以下三个于内存相关的任务非常重要：

1. 覆盖 JVM 最大堆大小。
2. 在可能的情况下，促使 JVM 向操作系统释放未使用的内存。
3. 确保正确配置了容器中的所有 JVM 进程。

优化容器中运行的 JVM 工作负载已超出本文讨论范畴，并且可能涉及设置多个额外的 JVM 选项。

4.5.2.1. 了解如何覆盖 JVM 最大堆大小

对于许多 Java 工作负载，JVM 堆是最大的内存用户。目前，OpenJDK 默认允许将计算节点最多 1/4 (1/**-XX:MaxRAMFraction**) 的内存用于该堆，不论 OpenJDK 是否在容器内运行。因此，务**必要**覆盖此行为，特别是设置了容器内存限制时。

达成以上目标至少有两种方式：

- 如果设置了容器内存限制，并且 JVM 支持那些实验性选项，请设置 **-XX:+UnlockExperimentalVMOptions -XX:+UseCGroupMemoryLimitForHeap**。

  注意

  UseCGroupMemoryLimitForHeap 选项已在 JDK 11 中删除。使用 **-XX:+UseContainerSupport 替代**。

  这会将 **-XX:MaxRAM** 设置为容器内存限制，并将最大堆大小 (**-XX:MaxHeapSize** / **-Xmx**) 设置为 1/**-XX:MaxRAMFraction**（默认为 1/4）。

- 直接覆盖 **-XX:MaxRAM**、**-XX:MaxHeapSize** 或 **-Xmx**。
  这个选项涉及对值进行硬编码，但也有允许计算安全裕度的好处。

4.5.2.2. 了解如何促使 JVM 向操作系统释放未用的内存

默认情况下，OpenJDK 不会主动向操作系统退还未用的内存。这可能适合许多容器化的 Java 工作负载，但也有明显的例外，例如额外活跃进程与容器内 JVM 共存的工作负载，这些额外进程是原生或附加的 JVM，或者这两者的组合。

基于 Java 的代理可使用以下 JVM 参数来鼓励 JVM 向操作系统释放未使用的内存：

```
-XX:+UseParallelGC
-XX:MinHeapFreeRatio=5 -XX:MaxHeapFreeRatio=10 -XX:GCTimeRatio=4
-XX:AdaptiveSizePolicyWeight=90.
```

这些参数旨在当分配的内存超过 110% 使用中内存时 (**-XX:MaxHeapFreeRatio**) 将堆内存返还给操作系统，这将在垃圾回收器上最多花费 20% 的 CPU 时间 (**-XX:GCTimeRatio**)。应用程序堆分配一定不会小于初始堆分配（被 **-XX:InitialHeapSize** / **- Xms** 覆盖）。调节 Java 在 OpenShift 中的内存占用（第 1 部分）、调节 Java 在 OpenShift 中的内存占用（第 2 部分） 以及 OpenJDK 和容器提供了其他的详细信息。

4.5.2.3. 了解如何确保正确配置容器中的所有 JVM 进程

如果多个 JVM 在同一容器中运行，则必须保证它们的配置都正确无误。如果有许多工作负载，需要为每个 JVM 分配一个内存预算百分比，留出较大的额外安全裕度。

许多 Java 工具使用不同的环境变量(**JAVA_OPTS**、**GRADLE_OPTS** 等)来配置其 JVM，并确保将正确的设置传递给正确的 JVM。

OpenJDK 始终尊重 **JAVA_TOOL_OPTIONS** 环境变量，在 **JAVA_TOOL_OPTIONS** 中指定的值会被 JVM 命令行中指定的其他选项覆盖。默认情况下，为了确保这些选项默认用于在基于 Java 的代理镜像中运行的所有 JVM 工作负载，Red Hat OpenShift Service on AWS Jenkins Maven 代理镜像集：

```
JAVA_TOOL_OPTIONS="-XX:+UnlockExperimentalVMOptions
-XX:+UseCGroupMemoryLimitForHeap -Dsun.zip.disableMemoryMapping=true"
```

注意

UseCGroupMemoryLimitForHeap 选项已在 JDK 11 中删除。使用 **-XX:+UseContainerSupport 替代**。

这不能保证不需要额外选项，只是用作一个实用的起点。

4.5.3. 从 pod 中查找内存请求和限制

希望从 pod 中动态发现内存请求和限制的应用程序应该使用 Downward API。

流程

1. 配置 pod，以添加 **MEMORY_REQUEST** 和 **MEMORY_LIMIT** 小节：
   a. 创建一个类似以下示例的 YAML 文件：

```
apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  securityContext:
    runAsNonRoot: true
    seccompProfile:
      type: RuntimeDefault
  containers:
  - name: test
    image: fedora:latest
    command:
    - sleep
    - "3600"
    env:
    - name: MEMORY_REQUEST (1)
      valueFrom:
        resourceFieldRef:
          containerName: test
          resource: requests.memory
    - name: MEMORY_LIMIT (2)
      valueFrom:
        resourceFieldRef:
          containerName: test
          resource: limits.memory
    resources:
      requests:
        memory: 384Mi
      limits:
        memory: 512Mi
    securityContext:
```

```
      allowPrivilegeEscalation: false
      capabilities:
        drop: [ALL]
```

1. 添加此小节来发现应用程序内存请求值。
2. 添加此小节来发现应用程序内存限制值。

b. 运行以下命令来创建 pod：

```
$ oc create -f <file-name>.yaml
```

验证

1. 使用远程 shell 访问 pod：

```
$ oc rsh test
```

2. 检查是否应用了请求的值：

```
$ env | grep MEMORY | sort
```

输出示例

```
MEMORY_LIMIT=536870912
MEMORY_REQUEST=402653184
```

注意

内存限制值也可由 **/sys/fs/cgroup/memory/memory.limit_in_bytes** 文件从容器内部读取。

4.5.4. 了解 OOM 终止策略

如果容器中所有进程的总内存用量超过内存限制，或者在严重的节点内存耗尽情形下，AWS 上的 Red Hat OpenShift Service 可以终止容器中的进程。

当进程超出内存（OOM）终止时，这可能会导致容器立即退出。如果容器 PID 1 进程收到 **SIGKILL**，则容器会立即退出。否则，容器行为将取决于其他进程的行为。

例如，某个容器进程以代码 137 退出，这表示它收到了 SIGKILL 信号。

如果容器没有立即退出，则能够检测到 OOM 终止，如下所示：

1. 使用远程 shell 访问 pod：

```
# oc rsh test
```

2. 运行以下命令，查看 **/sys/fs/cgroup/memory/memory.oom_control** 中的当前 OOM 终止计数：

```
$ grep '^oom_kill ' /sys/fs/cgroup/memory/memory.oom_control
```

输出示例

```
oom_kill 0
```

3. 运行以下命令来引发一个 OOM kill：

```
$ sed -e '' </dev/zero
```

输出示例

```
Killed
```

4. 运行以下命令查看 **sed** 命令的退出状态：

```
$ echo $?
```

输出示例

```
137
```

例如，**137** 代表容器进程以代码 137 退出，这表示它收到了 SIGKILL 信号。

5. 运行以下命令，查看 **/sys/fs/cgroup/memory/memory.oom_control** 中的 OOM 终止计数器：

```
$ grep '^oom_kill ' /sys/fs/cgroup/memory/memory.oom_control
```

输出示例

```
oom_kill 1
```

如果 pod 中的一个或多个进程遭遇 OOM 终止，那么当 pod 随后退出时（不论是否立即发生），它都将会具有原因为 **OOMKilled** 的 **Failed** 阶段。被 OOM 终止的 pod 可能会根据 **restartPolicy** 的值重启。如果不重启，复制控制器等控制器会看到 pod 的失败状态，并创建一个新 pod 来替换旧 pod。

使用以下命令获取 pod 状态：

```
$ oc get pod test
```

输出示例

```
NAME      READY     STATUS      RESTARTS   AGE
test      0/1       OOMKilled   0          1m
```

- 如果 pod 没有重启，请运行以下命令来查看 pod:

```
$ oc get pod test -o yaml
```

输出示例

```
...
status:
  containerStatuses:
  - name: test
    ready: false
    restartCount: 0
    state:
      terminated:
        exitCode: 137
        reason: OOMKilled
  phase: Failed
```

- 如果重启，运行以下命令来查看 pod:

```
$ oc get pod test -o yaml
```

输出示例

```
...
status:
  containerStatuses:
  - name: test
    ready: true
    restartCount: 1
    lastState:
      terminated:
        exitCode: 137
        reason: OOMKilled
    state:
      running:
  phase: Running
```

4.5.5. 了解 pod 驱除

当节点的内存耗尽时，Red Hat OpenShift Service on AWS 可以从其节点中驱除 pod。根据内存耗尽的程度，驱除可能是安全操作，但也不一定。安全驱除表示，各个容器的主进程 (PID 1) 收到 SIGTERM 信号，稍等片刻后，如果进程还未退出，则会收到一个 SIGKILL 信号。非安全驱除暗示着各个容器的主进程会立即收到 SIGKILL 信号。

被驱除的 pod 具有 **Failed** 阶段，原因为 **Evicted**。无论 **restartPolicy** 的值是什么，该 pod 都不会重启。但是，复制控制器等控制器会看到 pod 的失败状态，并且创建一个新 pod 来取代旧 pod。

```
$ oc get pod test
```

输出示例

```
NAME     READY     STATUS    RESTARTS   AGE
test     0/1       Evicted   0          1m
```

```
$ oc get pod test -o yaml
```

输出示例

```
...
status:
  message: 'Pod The node was low on resource: [MemoryPressure].'
  phase: Failed
  reason: Evicted
```

第 5 章 配置 PID 限制

进程标识符(PID)是 Linux 内核分配给系统中当前运行的每个进程或线程的唯一标识符。在系统上可以同时运行的进程数量限制为 4、194、304、Linux 内核。这个数字也可能受对其他系统资源（如内存、CPU 和磁盘空间）的有限访问的影响。

在 Red Hat OpenShift Service on AWS 4.11 及更新的版本中，pod 最多可以有 4,096 PID。如果您的工作负载需要超过这个对象，您可以通过配置 **KubeletConfig** 对象来增加允许的最大 PID 数量。

运行早于 4.11 的 Red Hat OpenShift Service on AWS 集群使用默认的 PID 限制 **1024**。

5.1. 了解进程 ID 限制

在 Red Hat OpenShift Service on AWS 中，在调度集群中工作前，请考虑为进程 ID (PID)使用的两个支持限制：

- 每个 pod 的最大 PID 数量。
  在 Red Hat OpenShift Service on AWS 4.11 及更高版本中，默认值为 4,096。这个值由节点上设置的 **podPidsLimit** 参数控制。

- 每个节点的最大 PID 数量。
  默认值取决于 节点资源。在 Red Hat OpenShift Service on AWS 中，这个值由 **--system-reserved** 参数控制，该参数会根据节点的总资源在每个节点上保留 PID。

当 pod 超过每个 pod 允许的最大 PID 数量时，pod 可能会停止正常工作，并可能会从节点驱除。如需更多信息，请参阅 Kubernetes 文档中的驱除信号和阈值。

当节点超过每个节点允许的最大 PID 数量时，节点可能会变得不稳定，因为新进程无法分配 PID。如果在不创建额外进程的情况下无法完成现有进程，则整个节点都可以不可用，需要重启。这种情况可能会导致数据丢失，具体取决于运行进程和应用程序。当达到这个阈值时，客户管理员和 Red Hat Site Reliability Engineering 会收到通知，**Worker 节点遇到 PIDPressure** 警告将出现在集群日志中。

5.2. 为 RED HAT OPENSHIFT SERVICE ON AWS POD 设置更高进程 ID 限制的风险

pod 的 **podPidsLimit** 参数控制该 pod 中可以同时运行的最大进程和线程数。

您可以将 **podPidsLimit** 的值从默认值 4,096 增加到最多 16,384。更改此值可能会给应用程序造成停机，因为更改 **podPidsLimit** 需要重新引导受影响的节点。

如果您每个节点运行了大量 pod，且节点上具有较高的 **podPidsLimit** 值，则的风险超过节点的 PID 最大值。

要找到单个节点上可以同时运行的最大 pod 数量，而不超过节点的 PID，请将 3,650,000 个由 **podPidsLimit** 值划分。例如，如果您的 **podPidsLimit** 值为 16,384，并且您希望 pod 使用接近该进程 ID 的数量，您可以在单个节点上安全地运行 222 个 pod。

注意

内存、CPU 和可用存储也可以限制可同时运行的最大 pod 数量，即使正确设置 **podPidsLimit** 值也是如此。如需更多信息，请参阅"规划您的环境"和"Limits and scalability"。

其他资源

- 实例类型
- 规划您的环境
- 限制和可扩展性

5.3. 在 ROSA CLASSIC 集群中配置 PID 限制

5.3.1. 对现有 Red Hat OpenShift Service on AWS 集群设置更高的进程 ID 限制

您可以通过创建或编辑更改 **--pod-pids-limit** 参数的 **KubeletConfig** 对象，在 AWS (ROSA)集群上设置更高的 **podPidsLimit**。

重要

在现有集群中更改 **podPidsLimit** 将触发集群中的非 control plane 节点，以一次重启。在集群的峰值用量时间外进行这个更改，并避免升级或休眠集群，直到所有节点都重启为止。

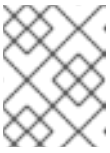
前提条件

- 您有一个 Red Hat OpenShift Service on AWS 集群。
- 已安装 ROSA CLI (**rosa**)。
- 已安装 OpenShift CLI(**oc**)。
- 已使用 ROSA CLI 登录到您的红帽帐户。

流程

1. 创建或编辑 **KubeletConfig** 对象以更改 PID 限制。
   - 如果这是您第一次更改默认 PID 限制，请运行以下命令创建 **KubeletConfig** 对象并设置 **--pod-pids-limit** 值：

     ```
     $ rosa create kubeletconfig -c <cluster_name> --name <kubeletconfig_name> --pod-pids-limit=<value>
     ```

     注意

     --name 参数在 ROSA Classic 集群中是可选的，因为每个 ROSA Classic 集群只支持一个 **KubeletConfig** 对象。

     例如，以下命令为集群 **my-cluster** 设置最多 16,384 PID：

     ```
     $ rosa create kubeletconfig -c my-cluster --name set-high-pids --pod-pids-limit=16384
     ```

   - 如果您之前创建了 **KubeletConfig** 对象，请运行以下命令编辑现有的 **KubeletConfig** 对象并设置 **--pod-pids-limit** 值：

     ```
     $ rosa edit kubeletconfig -c <cluster_name> --name <kubeletconfig_name> --pod-pids-limit=<value>
     ```

-

触发集群范围的 worker 节点的滚动重启。

2. 运行以下命令，验证所有 worker 节点是否重启：

```
$ oc get machineconfigpool
```

输出示例

```
NAME     CONFIG                 UPDATED  UPDATING   DEGRADED  MACHINECOUNT
READYMACHINECOUNT  UPDATEDMACHINECOUNT DEGRADEDMACHINECOUNT
AGE
master   rendered-master-06c9c4…  True     False      False     3          3              3
0                  4h42m
worker   rendered-worker-f4b64…   True     False      False     4          4              4
0                  4h42m
```

验证

当集群中的每个节点都重启时，您可以验证新设置是否就位。

- 检查 **KubeletConfig** 对象中的 Pod Pids 限制：

```
$ rosa describe kubeletconfig --cluster=<cluster_name>
```

新的 PID 限值会出现在输出中，如下例所示：

输出示例

```
Pod Pids Limit:                16384
```

5.3.2. 从集群中删除自定义配置

您可以通过删除包含配置详情的 **KubeletConfig** 对象从集群中删除自定义配置。

前提条件

- 您有一个现有的 Red Hat OpenShift Service on AWS 集群。
- 已安装 ROSA CLI (rosa)。
- 已使用 ROSA CLI 登录到您的红帽帐户。

流程

- 通过删除相关的自定义 **KubeletConfig** 对象从集群中删除自定义配置：

```
$ rosa delete kubeletconfig --cluster <cluster_name> --name <kubeletconfig_name>
```

验证步骤

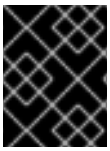
- 确认没有为集群列出自定义 **KubeletConfig** 对象：

```
$ rosa describe kubeletconfig --name <cluster_name>
```

5.4. 使用 HCP 集群在 ROSA 上配置 PID 限制

5.4.1. 在 Red Hat OpenShift Service on AWS 集群中为机器池设置更高的进程 ID 限制

您可以通过创建或编辑更改 **--pod-pids-limit** 参数的 **KubeletConfig** 对象，为现有 Red Hat OpenShift Service on AWS (ROSA)集群中的机器池设置更高的 **podPidsLimit**。



重要

更改现有机器池上的 **podPidsLimit** 会触发机器池中的节点一次重启。为机器池中的工作负载进行这个更改，并避免升级或休眠集群，直到所有节点都重启为止。

前提条件

- 您有一个 Red Hat OpenShift Service on AWS 集群。
- 已安装 ROSA CLI (**rosa**)。
- 已使用 ROSA CLI 登录到您的红帽帐户。

流程

1. 为集群创建新的 **KubeletConfig** 对象，用于指定新的 **--pod-pids-limit** ：

```
$ rosa create kubeletconfig -c <cluster_name> --name=<kubeletconfig_name>  --pod-pids-limit=<value>
```

例如，以下命令为 **my-cluster** 集群创建一个 **set-high-pids KubeletConfig** 对象，该集群设置最大 16,384 PIDs 每个 pod：

```
$ rosa create kubeletconfig -c my-cluster --name=set-high-pids --pod-pids-limit=16384
```

2. 将新的 **KubeletConfig** 对象与新或现有机器池关联。
   - 对于新机器池：

```
$ rosa create machinepool -c <cluster_name> --kubelet-configs=<kubeletconfig_name> --name <machinepool_name>
```

   - 对于现有的机器池：

```
$ rosa edit machinepool -c <cluster_name> --kubelet-configs=<kubeletconfig_name> --name <machinepool_name>
```

例如，以下命令将 **set-high-pids KubeletConfig** 对象与 **my-cluster** 集群中的 **high-pid-pool** 机器池相关联：

```
$ rosa edit machinepool -c my-cluster --kubelet-configs=set-high-pids --name=high-pid-pool
```

当新的 **KubeletConfig** 对象附加到现有机器池时，会触发 worker 节点的滚动重启。您可以在机器池描述中检查 rollout 的进度：

```
$ rosa describe machinepool --cluster <cluster_name> --name <machinepool_name>
```

验证

- 确认新设置已放在机器池中的节点上：

```
$ rosa describe kubeletconfig --cluster=<cluster_name> --name <kubeletconfig_name>
```

新的 PID 限值会出现在输出中，如下例所示：

输出示例

```
Pod Pids Limit:                 16384
```

5.4.2. 从机器池中删除自定义配置

您可以通过删除包含配置详情的 **KubeletConfig** 对象来删除机器池上的自定义配置。

前提条件

- 您有一个现有的 Red Hat OpenShift Service on AWS 集群。
- 已安装 ROSA CLI (rosa)。
- 已使用 ROSA CLI 登录到您的红帽帐户。

流程

- 编辑机器池并设置 **--kubeletconfigs** 参数，以便省略您要删除的 **KubeletConfig** 对象。要从机器池中删除所有 **KubeletConfig** 对象，请为 **--kubeletconfigs** 参数设置一个空值，例如：

```
$ rosa edit machinepool -c <cluster_name> --kubeletconfigs="" --name
<machinepool_name>
```

验证步骤

- 确认您删除的 **KubeletConfig** 对象在机器池描述中不可见：

```
$ rosa describe machinepool --cluster <cluster_name> --name <machinepool_name>
```