Red Hat build of MicroShift 4.16

故障排除

常见问题故障排除

Last Updated: 2024-07-02

Red Hat build of MicroShift 4.16 故障排除

常见问题故障排除

法律通告

摘要

有关对 MicroShift 常见红帽构建的故障排除的信息。

目录

第 1 章 检查您已安装了哪个版本

要开始故障排除，请确定您已安装了哪个红帽构建的 MicroShift 版本。

1.1. 使用命令行界面检查版本

在开始故障排除前，需要知道您的 MicroShift 版本。获取此信息的一种方法是使用 CLI。

流程

- 运行以下命令来检查版本信息：

```
$ microshift version
```

输出示例

```
Red Hat build of MicroShift Version: 4.16-0.microshift-e6980e25
Base OCP Version: 4.16
```

1.2. 使用 API 检查 MICROSHIFT 版本

在开始故障排除前，需要知道您的 MicroShift 版本。获取此信息的一种方法是使用 API。

流程

- 要使用 OpenShift CLI (**oc**) 获取版本号，请运行以下命令来查看 **kube-public/microshift-version** 配置映射：

```
$ oc get configmap -n kube-public microshift-version -o yaml
```

输出示例

```
apiVersion: v1
data:
  major: "4"
  minor: "13"
  version: 4.13.8-0.microshift-fa441af87431
kind: ConfigMap
metadata:
  creationTimestamp: "2023-08-03T21:06:11Z"
  name: microshift-version
  namespace: kube-public
```

1.3. 检查 ETCD 版本

您可以使用以下方法之一获取 MicroShift 中包含的 etcd 数据库的版本信息，具体取决于您需要的信息级别。

流程

- 要显示基本数据库版本信息，请运行以下命令：

```
$ microshift-etcd version
```

输出示例

```
microshift-etcd Version: 4.16.0
Base etcd Version: 3.5.13
```

- 要显示完整的数据库版本信息，请运行以下命令：

```
$ microshift-etcd version -o json
```

输出示例

```
{
  "major": "4",
  "minor": "16",
  "gitVersion": "4.16.0~rc.1",
  "gitCommit": "140777711962eb4e0b765c39dfd325fb0abb3622",
  "gitTreeState": "clean",
  "buildDate": "2024-05-10T16:37:53Z",
  "goVersion": "go1.21.9"
  "compiler": "gc",
  "platform": "linux/amd64",
  "patch": "",
  "etcdVersion": "3.5.13"
}
```

第 2 章 集群故障排除

要开始对 MicroShift 集群进行故障排除，首先访问集群状态。

2.1. 检查集群的状态

您可以检查 MicroShift 集群的状态，或查看活跃的 pod。以下流程中给出有三个不同的命令，可用于检查集群状态。您可以选择运行一个、两个或所有命令，以帮助您获取对集群进行故障排除所需的信息。

流程

- 运行以下命令，检查返回集群状态的系统状态：

```
$ sudo systemctl status microshift
```

如果 MicroShift 无法启动，这个命令会返回上一个运行的日志。

健康输出示例

```
● microshift.service - MicroShift
     Loaded: loaded (/usr/lib/systemd/system/microshift.service; enabled; preset: disabled)
     Active: active (running) since <day> <date> 12:39:06 UTC; 47min ago
   Main PID: 20926 (microshift)
      Tasks: 14 (limit: 48063)
     Memory: 542.9M
        CPU: 2min 41.185s
     CGroup: /system.slice/microshift.service
             └─20926 microshift run

<Month-Day> 13:23:06 i-06166fbb376f14a8b.<hostname> microshift[20926]: kube-apiserver
I0528 13:23:06.876001   20926 controll>
<Month-Day> 13:23:06 i-06166fbb376f14a8b.<hostname> microshift[20926]: kube-apiserver
I0528 13:23:06.876574   20926 controll>
# ...
```

- 可选：运行以下命令来获取全面的日志：

```
$ sudo journalctl -u microshift
```

注意

systemd 日志服务的默认配置会将数据存储在易失性目录中。要在系统启动和重启后保留系统日志，请启用日志持久性并设置最大日志数据大小的限制。

- 可选：如果 MicroShift 正在运行，请输入以下命令检查活跃 pod 的状态：

```
$ oc get pods -A
```

输出示例

```
NAMESPACE                  NAME                                  READY   STATUS
RESTARTS  AGE
```

```
default                 i-06166fbb376f14a8bus-west-2computeinternal-debug-qtwcr  1/1
Running  0     46m
kube-system             csi-snapshot-controller-5c6586d546-lprv4              1/1     Running  0
51m
kube-system             csi-snapshot-webhook-6bf8ddc7f5-kz6k9                 1/1     Running
0      51m
openshift-dns           dns-default-45jl7                                     2/2     Running  0      50m
openshift-dns           node-resolver-7wmzf                                   1/1     Running  0      51m
openshift-ingress       router-default-78b86fbf9d-qvj9s                       1/1     Running  0
51m
openshift-ovn-kubernetes    ovnkube-master-5rfhh                              4/4     Running  0
51m
openshift-ovn-kubernetes    ovnkube-node-gcnt6                                1/1     Running  0
51m
openshift-service-ca    service-ca-bf5b7c9f8-pn6rk                            1/1     Running  0
51m
openshift-storage       topolvm-controller-549f7fbdd5-7vrmv                   5/5     Running  0
51m
openshift-storage       topolvm-node-rht2m                                    3/3     Running  0
50m
```

注意

这个示例输出显示基本的 MicroShift。如果您安装了可选的 RPM，则您的输出中也会显示运行这些服务的 pod 状态。

第 3 章 数据备份和恢复故障排除

要排除失败的数据备份和恢复，请首先检查数据路径、存储配置和存储容量等基础知识。

3.1. 备份数据失败

数据备份在 **rpm-ostree** 系统上是自动的。如果您没有使用 **rpm-ostree** 系统，并尝试创建手动备份，原因可能会导致备份失败：

- 系统启动后不会等待几分钟，以成功停止 MicroShift。系统必须在备份成功前完成健康检查和任何其他后台进程。
- 如果 MicroShift 因为错误而停止运行，则无法备份数据。
  - 确保系统处于健康状态。
  - 在尝试备份前，将其处于健康状态。
- 如果您没有足够的数据存储，备份会失败。确保有足够的存储用于 MicroShift 数据。
- 如果您没有足够的权限，备份可能会失败。确保您有正确的用户权限来创建备份并执行所需的配置。

3.2. 备份日志

- 在手动备份过程中，日志会打印到终端控制台。
- 作为 MicroShift 日志的一部分，会自动为 **rpm-ostree** 系统自动备份生成日志。您可以运行以下命令来检查日志：

```
$ sudo journalctl -u microshift
```

3.3. 恢复数据失败

出于许多原因，数据恢复可能会失败，包括存储和权限问题。不匹配的数据版本可能会在 MicroShift 重启时造成失败。

3.3.1. 基于 RPM-OSTree 的系统数据恢复失败

数据恢复在 **rpm-ostree** 系统中是自动的，但可能会失败，例如：

- **rpm-ostree** 系统上恢复的唯一备份是从当前部署或回滚部署中备份。备份不会在不健康的系统中执行。
  - 只有具有对应部署的最新备份才会被保留。没有匹配部署的过时的备份会被自动删除。
  - 数据通常不会从较新的 MicroShift 版本中恢复。
  - 确保恢复的数据遵循与更新路径相同的版本模式。例如，如果 MicroShift 的目标版本比您当前使用的 MicroShift 数据版本旧的版本，则恢复可能会失败。

3.3.2. 基于 RPM 的手动数据恢复失败

如果您使用不是 **rpm-ostree** 的 RPM 系统，并尝试恢复手动备份，原因可能会导致恢复失败：

- 如果 MicroShift 因为错误而停止运行，则无法恢复数据。
  - 确保系统处于健康状态。
  - 在尝试恢复数据前，启动它处于健康状态。
- 如果您没有为传入的数据分配足够的存储空间，恢复会失败。
  - 请确定将您当前的系统存储配置为接受恢复的数据。
- 您试图从 MicroShift 的新版本恢复数据。
  - 确保恢复的数据遵循与更新路径相同的版本模式。例如，如果 MicroShift 的目标版本比您要使用的 MicroShift 数据版本旧的版本，则恢复可能会失败。

3.4. 存储迁移失败

存储迁移失败通常是由于自定义资源(CR)中的大量更改从一个 MicroShift 变为下一个。如果存储迁移失败，则通常会在版本之间无法解析，需要手动审核。

第 4 章 排除更新

要对 MicroShift 更新进行故障排除，请使用以下指南。

4.1. MICROSHIFT 更新故障排除

在某些情况下，MicroShift 可能无法更新。在这些事件中，了解失败类型以及如何对它们进行故障排除很有帮助。

4.1.1. 更新路径会因为版本不兼容而阻止

如果 MicroShift 更新与 Red Hat Enterprise Linux for Edge (RHEL for Edge)或 Red Hat Enterprise Linux (RHEL)的版本不兼容，则 RPM 依赖项错误会导致。

4.1.1.1. 兼容性表

检查以下兼容性表：

Red Hat Device Edge 发行版本兼容性列表

Red Hat Enterprise Linux (RHEL)和 MicroShift 可以一起工作，作为设备边缘计算的单一解决方案。您可以单独更新每个组件，但产品版本必须兼容。例如，MicroShift 从 4.14 更新至 4.16 需要 RHEL 更新。如下表所示，Red Hat Device Edge 的支持的配置为每个 Red Hat Device Edge 使用验证的版本：

RHEL for Edge 版本	MicroShift 版本	MicroShift 发行版本状态	支持的 MicroShift 版本→MicroShift 版本更新
9.4	4.16	正式发布	4.16.0→4.16.z, 4.14→4.16 和 4.15→4.16
9.2, 9.3	4.15	正式发布	4.15.0→4.15.z, 4.14→4.15 和 4.15→4.16
9.2, 9.3	4.14	正式发布	4.14.0→4.14.z, 4.14→4.15 和 4.14→4.16
9.2	4.13	技术预览	None
8.7	4.12	开发者预览	None

4.1.1.2. 版本兼容性

检查以下更新路径：

红帽构建的 MicroShift 更新路径

- RHEL for Edge 9.4 上正式发布版本 4.16.0 到 4.16.z
- RHEL 9.4 上正式发布版本 4.15.0 从 RHEL 9.2 升级到 4.16.0
- 通常 Available Version 4.14.0 从 RHEL 9.2 到 4.16.0 on RHEL 9.4

4.1.2. ostree 更新失败

如果您在 OSTree 系统上更新，Greenboot 健康检查会自动日志并处理系统健康状况。通过 Greenboot 的系统回滚可以指示失败。如果更新失败，但 Greenboot 没有完成系统回滚，您可以使用此内容"Additional resources"部分中的 RHEL for Edge 文档进行故障排除。

手动检查 Greenboot 日志

- 运行以下命令，手动检查 Greenboot 日志以验证系统健康状况：

```
$ sudo systemctl restart --no-block greenboot-healthcheck && sudo journalctl -fu greenboot-healthcheck
```

4.1.3. 手动 RPM 更新失败

如果您在非 OSTree 系统上使用 RPM 更新，则 Greenboot 可以指示更新失败，但健康检查仅以信息形式表示。检查系统日志是手动 RPM 更新故障故障排除的下一步。您可以使用 Greenboot 和 **sos report** 检查 MicroShift 更新和主机系统。

其他资源

- 启用 **systemd** 日志服务数据持久性
- 检查 MicroShift 版本
- 停止 MicroShift 服务
- 启动 MicroShift 服务
- 准备、安装和管理 RHEL for Edge 镜像
- RHEL for Edge 镜像的回滚

4.2. 在更新后检查日志

在某些情况下，MicroShift 可能无法更新。在这些事件中，了解失败类型以及如何对它们进行故障排除很有帮助。日志可帮助诊断更新失败。

注意

systemd 日志服务的默认配置会将数据存储在易失性目录中。要在系统启动和重启后保留系统日志，请启用日志持久性并设置最大日志数据大小的限制。

流程

- 运行以下命令来获取全面的 MicroShift 日志：

```
$ sudo journalctl -u microshift
```

- 运行以下命令检查 Greenboot 日志：

```
$ sudo journalctl -u greenboot-healthcheck
```

- 检查特定引导的综合日志使用了三个步骤：首先列出引导，然后从您获取的列表中选择您想要的：
  - 运行以下命令，列出日志日志中存在的引导：

```
$ sudo journalctl --list-boots
```

输出示例

```
IDX  BOOT ID                          FIRST ENTRY                 LAST ENTRY
 0   681ece6f5c3047e183e9d43268c5527f  <Day> <Date> 12:27:58 UTC  <Day>
<Date>> 13:39:41 UTC
#....
```

  - 运行以下命令，检查您想要的特定引导的日志：

```
$ sudo journalctl --boot <-my_boot_ID> 1
```

1 将 *HEKETImy-boot-ID* > 替换为您要检查的特定引导数。

  - 运行以下命令，检查特定服务的引导日志：

```
$ sudo journalctl --boot <-my_boot_ID> -u <service_name> 1 2
```

1 将 *HEKETImy-boot-ID* > 替换为您要检查的特定引导数。

2 将 < *service_name* > 替换为您要检查的服务的名称。

4.3. 检查 GREENBOOT 健康检查的状态

在对系统进行更改或故障排除期间，检查 Greenboot 健康检查的状态。您可以使用以下任一命令来帮助确保 Greenboot 脚本已运行。

流程

- 要查看健康检查状态的报告，请使用以下命令：

```
$ systemctl show --property=SubState --value greenboot-healthcheck.service
```

  - **start** 的输出表示 Greenboot 检查仍在运行。
  - **退出** 的输出表示检查已通过，Greenboot 已退出。当系统处于健康状态时，greenboot 在 **green.d** 目录中运行脚本。
  - **失败** 的输出意味着检查还没有通过。greenboot 在系统处于此状态时在 **red.d** 目录中运行脚本，并可能重启系统。
- 要查看显示服务的数字退出代码的报告，其中 **0** 表示成功，非零值表示发生失败，请使用以下命令：

```
$ systemctl show --property=ExecMainStatus --value greenboot-healthcheck.service
```

- 要查看显示引导状态的消息的报告，如 **Boot Status** 为 **GREEN - Health Check SUCCESS**，请使用以下命令：

```
$ cat /run/motd.d/boot-status
```

第 5 章 检查审计日志

您可以使用审计日志来识别 pod 安全违反情况。

5.1. 通过审计日志识别 POD 安全违反情况

您可以通过查看服务器审计日志来识别工作负载中的 pod 安全准入违反情况。以下流程演示了如何访问审计日志，并解析它们以在工作负载中查找 pod 安全准入违反情况。

先决条件

- 您已安装了 **jq**。
- 您可以使用具有 **cluster-admin** 角色的用户访问集群。

流程

1. 要检索节点名称，请运行以下命令：

```
$ <node_name>=$(oc get node -ojsonpath='{.items[0].metadata.name}')
```

2. 要查看审计日志，请运行以下命令：

```
$ oc adm node-logs <node_name> --path=kube-apiserver/
```

输出示例

```
rhel-92.lab.local audit-2023-08-18T18-25-41.663.log
rhel-92.lab.local audit-2023-08-19T11-21-29.225.log
rhel-92.lab.local audit-2023-08-20T04-16-09.622.log
rhel-92.lab.local audit-2023-08-20T21-11-41.163.log
rhel-92.lab.local audit-2023-08-21T14-06-10.402.log
rhel-92.lab.local audit-2023-08-22T06-35-10.392.log
rhel-92.lab.local audit-2023-08-22T23-26-27.667.log
rhel-92.lab.local audit-2023-08-23T16-52-15.456.log
rhel-92.lab.local audit-2023-08-24T07-31-55.238.log
```

3. 要解析受影响的审计日志，请输入以下命令：

```
$ oc adm node-logs <node_name> --path=kube-apiserver/audit.log \
  | jq -r 'select((.annotations["pod-security.kubernetes.io/audit-violations"] != null) and
(.objectRef.resource=="pods")) | .objectRef.namespace + " " + .objectRef.name + " " +
.objectRef.resource' \
  | sort | uniq -c
```

第 6 章 对 ETCD 进行故障排除

要对 etcd 进行故障排除并提高性能，请配置服务的内存允许。

6.1. 配置 MEMORYLIMITMB 值来为 ETCD 服务器设置参数

默认情况下，etcd 根据需要使用尽可能多的内存来处理系统上的负载。在内存有限制的系统中，您可能需要限制 etcd 使用的内存量。

流程

- 编辑 **/etc/microshift/config.yaml** 文件，以设置 **memoryLimitMB** 值。

```
etcd:
  memoryLimitMB: 128
```

注意

MicroShift 上 **memoryLimitMB** 所需的最小值为 128 MB。接近最小值的值可能会影响 etcd 性能。较小的限制，etcd 会对查询做出响应所需的时间。如果限制太小，或者 etcd 的使用量很高，查询会超时。

验证

1. 修改 **/etc/microshift/config.yaml** 中的 **memoryLimitMB** 值后，运行以下命令重启 MicroShift：

```
$ sudo systemctl restart microshift
```

2. 运行以下命令验证新的 **memoryLimitMB** 值是否在使用中：

```
$ systemctl show --property=MemoryHigh microshift-etcd.scope
```

第 7 章 响应重启和安全证书

红帽构建的 MicroShift 在检测到更改后会对系统配置的更改（包括 IP 地址更改、时钟调整和安全证书期限）进行响应并重启。

7.1. IP 地址更改或时钟调整

MicroShift 依赖于设备 IP 地址和系统范围的时钟设置，以便在运行时保持一致。但是，这些设置偶尔可能会在边缘设备上更改，如 DHCP 或网络时间协议 (NTP) 更新。

当进行此类更改时，一些 MicroShift 组件可能会停止正常运行。为缓解这种情况，MalShift 会监控 IP 地址和系统时间，并在检测到任一设置更改时重新启动。

触发时钟更改的阈值是，时间的调整超过 10 秒（早或晚）。网络时间协 议(NTP) 服务会定期执行小的时间调整，这不会造成重启。

7.2. 安全证书生命周期

MicroShift 证书被分为两个基本组：

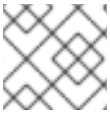
1. 短期证书的有效期为一年。
2. 长期证书的有效期为 10 年。

大多数服务器或叶证书都是短期的。

一个长期的证书示例是用于 **system:admin** 用户身份验证的客户端证书，或 **kube-apiserver** 外部服务证书的证书。

7.2.1. 证书轮转

过期或接近其过期日期的证书需要轮转，以确保继续 MicroShift 操作。当 MicroShift 因任何原因重启时，接近过期的证书将被轮转。当证书被设置为马上过期或已过期，可能会导致自动 MicroShift 重启执行轮转。



注意

如果轮转的证书是证书颁发机构（CA），则其签署的所有证书都会轮转。

7.2.1.1. 短期证书

以下情况描述了在短期证书生命周期内 MicroShift 的操作：

1. 无轮转：
   a. 当短期证书最多已存在 5 个月时，不会发生轮转。
2. 重启时轮转：
   a. 当短期证书已存在 5 到 8 个月时，它会在 MicroShift 启动或重启时进行轮转。
3. 自动重启进行轮转：
   a. 当短期证书存在超过 8 个月时，MicroShift 可以自动重启以轮转并应用新证书。

7.2.1.2. 长期证书

以下情况描述了在长期证书生命周期内 MicroShift 的操作：

1. 无轮转：
    a. 当长期证书存在最多 8.5 年时，不会发生轮转。
2. 重启时轮转：
    a. 当长期证书存在 8.5 到 9 年时，它会在 MicroShift 启动或重启时进行轮转。
3. 自动重启进行轮转：
    a. 当长期证书存在超过 9 年时，MicroShift 可以自动重启来轮转并应用新证书。