



Red Hat Enterprise Linux 9

在 Microsoft Azure 上部署 RHEL 9

获取 RHEL 系统镜像并在 Azure 上创建 RHEL 实例

Red Hat Enterprise Linux 9 在 Microsoft Azure 上部署 RHEL 9

获取 RHEL 系统镜像并在 Azure 上创建 RHEL 实例

法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

要在公有云环境中使用 Red Hat Enterprise Linux (RHEL)，您可以在各种云平台（包括 Microsoft Azure）上创建和部署 RHEL 系统镜像。您还可以在 Azure 上创建和配置红帽高可用性(HA)集群。以下章节提供了在 Azure 上创建云 RHEL 实例和 HA 集群的说明。这些进程包括安装所需的软件包和代理、配置隔离以及安装网络资源代理。

目录

对红帽文档提供反馈	3
第 1 章 公有云平台上的 RHEL 简介	4
1.1. 在公有云中使用 RHEL 的好处	4
1.2. RHEL 的公有云用例	4
1.3. 迁移到公有云时的常见关注	5
1.4. 为公有云部署获取 RHEL	6
1.5. 创建 RHEL 云实例的方法	6
第 2 章 将 VHD 镜像推送到 MICROSOFT AZURE 云	8
第 3 章 在 MICROSOFT AZURE 上将 RED HAT ENTERPRISE LINUX 镜像部署为虚拟机	11
3.1. AZURE 上的 RED HAT ENTERPRISE LINUX 镜像选项	11
3.2. 理解基础镜像	12
3.3. 为 MICROSOFT AZURE 配置自定义基础镜像	14
3.4. 将镜像转换为固定 VHD 格式	18
3.5. 安装 AZURE CLI	19
3.6. 在 AZURE 中创建资源	20
3.7. 上传并创建 AZURE 镜像	23
3.8. 在 AZURE 中创建并启动虚拟机	24
3.9. 其他验证方法	25
3.10. 附加红帽订阅	25
3.11. 对 AZURE 黄金镜像设置自动注册	26
3.12. 为 MICROSOFT AZURE 实例配置 KDUMP	27
3.13. 其他资源	29
第 4 章 在 MICROSOFT AZURE 上配置红帽高可用性集群	30
4.1. 在公有云平台上使用高可用性集群的好处	30
4.2. 在 AZURE 中创建资源	30
4.3. 高可用性所需的系统软件包	34
4.4. AZURE VM 配置设置	35
4.5. 安装 HYPER-V 设备驱动程序	35
4.6. 进行 MICROSOFT AZURE 部署所需的配置更改	36
4.7. 创建 AZURE ACTIVE DIRECTORY 应用程序	39
4.8. 将镜像转换为固定 VHD 格式	41
4.9. 上传并创建 AZURE 镜像	42
4.10. 安装 RED HAT HA 软件包和代理	43
4.11. 创建集群	45
4.12. 隔离 (FENCING) 概述	46
4.13. 创建隔离设备	46
4.14. 创建 AZURE 内部负载均衡器	48
4.15. 配置负载均衡资源代理	49
4.16. 配置共享块存储	50
4.17. 其他资源	54

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

通过 Jira 提交反馈（需要帐户）

1. 登录到 [Jira](#) 网站。
2. 点顶部导航栏中的 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您对改进的建议。包括文档相关部分的链接。
5. 点对话框底部的 **Create**。

第 1 章 公有云平台上的 RHEL 简介

公有云平台提供计算资源即服务。您可以运行您的 IT 工作负载，包括 Red Hat Enterprise Linux (RHEL) 系统，作为公共云实例，而不是使用内部硬件。

1.1. 在公有云中 使用 RHEL 的好处

RHEL 作为公共云平台上的云实例与内部物理系统或虚拟机(VM)的 RHEL 相比有以下优点：

- **灵活精细的资源分配**

RHEL 的云实例作为虚拟机在云平台上运行，这通常意味着由云服务提供商维护远程服务器集群。因此，给实例分配硬件资源（如特定类型的 CPU 或存储）发生在软件层面上，可轻松自定义。

与本地 RHEL 系统相比，您也不会受物理主机功能的限制。相反，您可以根据云提供商提供的选择，从各种功能中进行选择。

- **空间及成本效率**

您不需要拥有任何内部服务器来托管您的云工作负载。这可避免与物理硬件关联的空间、电源和维护要求。

相反，在公有云平台上，您可以直接向云提供商支付使用云实例的费用。成本通常基于分配给实例的硬件以及您使用的时间。因此，您可以根据要求优化成本。

- **软件控制的配置**

云实例的整个配置都作为数据保存在云平台上，并由软件控制。因此，您可以轻松地创建、删除、克隆或迁移实例。云实例也在云提供商控制台中远程操作，默认连接到远程存储。

另外，您可以随时将云实例的当前状态备份为快照。之后，您可以加载快照，将实例恢复到保存的状态。

- **与主机分离和软件兼容性**

与本地虚拟机类似，云实例上的 RHEL 客户机操作系统运行在虚拟化内核上。这个内核与主机操作系统以及用来连接实例的客户端系统分开。

因此，任何操作系统都可以安装在云实例上。这意味着，在 RHEL 公有云实例中，您可以运行无法在本地操作系统上使用的特定于 RHEL 的应用程序。

另外，即使实例的操作系统变得不稳定或被破坏，您的客户端系统也不会受到任何影响。

其他资源

- [什么是公有云？](#)
- [什么是 Hyperscaler？](#)
- [云计算的类型](#)
- [RHEL 的公有云用例](#)
- [为公有云部署获取 RHEL](#)

1.2. RHEL 的公有云用例

在公有云上部署会带来许多好处，但可能并非每种场景中最有效的解决方案。如果您要评估是否将 RHEL 部署迁移到公共云，请考虑您的用例是否将从公共云的好处中受益。

有益的用例

- 部署公有云实例对于灵活地增加和减少部署的活跃计算能力（也称为 *扩展* 和 *缩减*）非常有效。因此，在以下情况下，建议在公有云上使用 RHEL：
 - 具有峰值工作负载和一般性能要求的集群。在资源成本方面，根据您的需求扩展和缩减可能非常高效。
 - 快速设置或扩展集群。这可避免设置本地服务器的高前期成本。
- 云实例不受本地环境中发生的情况的影响。因此，您可以使用它们进行备份和恢复。

有潜在问题的用例

- 您正在运行一个无法调整的现有环境。与您当前的主机平台相比，自定义云实例以适应现有部署的特定需求可能并不划算。
- 您有预算方面的硬限制。在本地数据中心中维护您的部署通常具有更大的灵活性，但与公有云相比，您对最大资源成本有更多的控制。

后续步骤

- [为公有云部署获取 RHEL](#)

其他资源

- [我是否应该将应用程序迁移到云？以下是如何决定。](#)

1.3. 迁移到公有云时的常见关注

将 RHEL 工作负载从本地环境移到公有云平台可能会带来有关涉及的变化担忧。以下是最常见的问题。

作为云实例，我的 RHEL 是否与本地虚拟机工作不同？

在大部分方面，公有云平台上的 RHEL 实例的工作方式与本地主机上的 RHEL 虚拟机相同，如内部服务器。主要例外包括：

- 公有云实例使用特定于提供商的控制台接口，而不是私有编排接口，来管理云资源。
- 某些功能（如嵌套虚拟化）可能无法正常工作。如果特定功能对部署至关重要，请提前检查该功能与您选择的公有云提供商的兼容性。

与本地服务器相比，我的数据在公有云中是否保持安全？

RHEL 云实例中的数据归您所有，您的公共云提供商对齐没有任何访问权限。此外，主要的云提供商支持传输中的数据加密，这提高了将虚拟机迁移到公共云时数据的安全性。

RHEL 公共云实例的一般安全性如下：

- 您的公有云供应商负责云 hypervisor 的安全性
- 红帽在您的实例中提供 RHEL 客户机操作系统的安全功能
- 您可以在云基础架构中管理特定的安全设置和实践

我的地理区域对 RHEL 公有云实例的功能有何影响？

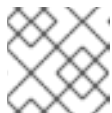
无论您所在的地理位置如何，您都可以在公有云平台上使用 RHEL 实例。因此，您可以在与内部服务器相同的区域中运行实例。

但是，在物理上较远的区域中托管您的实例可能会在操作它们时造成高延迟。此外，取决于公有云提供商，某些区域可能会提供额外的功能或更具成本效益。在创建 RHEL 实例前，请查看您选择的云提供商提供的托管区域的属性。

1.4. 为公有云部署获取 RHEL

要在公有云环境中部署 RHEL 系统，您需要：

1. 根据您的需求和当前市场提供的，为您的使用案例选择最佳云提供商。
当前认证的运行 RHEL 实例的云提供商有：
 - [Amazon Web Services \(AWS\)](#)
 - [Google Cloud Platform \(GCP\)](#)
 - [Microsoft Azure](#)



注意

本文档专门讨论在 Microsoft Azure 上部署 RHEL。

2. 在您选择的云平台上创建 RHEL 云实例。如需更多信息，请参阅 [创建 RHEL 云实例的方法](#)。
3. 要让您的 RHEL 部署保持最新状态，请使用 [红帽更新基础设施 \(RHUI\)](#)。

其他资源

- [RHUI 文档](#)
- [Red Hat Open Hybrid Cloud](#)

1.5. 创建 RHEL 云实例的方法

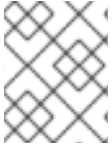
要在公有云上部署 RHEL 实例，您可以使用以下方法之一：

创建 RHEL 的系统镜像，并将其导入到云平台。

- 要创建系统镜像，您可以使用 [RHEL 镜像构建器](#)，也可以手动构建镜像。
- 此方法使用您现有的 RHEL 订阅，也称为 *自带订阅* (BYOS)。
- 您每年预付订阅费，您可以使用您的红帽客户折扣。
- 您的客户服务由红帽提供。
- 要有效地创建多个镜像，您可以使用 **cloud-init** 工具。

直接从云提供商市场购买 RHEL 实例。

- 您按小时为使用的服务后付费。因此，此方法也称为 *随用随付*(PAYG)。
- 您的客户服务由云平台提供商提供。



注意

有关使用各种方法在 Microsoft Azure 上部署 RHEL 实例的详细信息，请参阅本文档中的以下章节。

其他资源

- [什么是黄金镜像？](#)
- [为 RHEL 9 配置和管理 cloud-init](#)

第 2 章 将 VHD 镜像推送到 MICROSOFT AZURE 云

您可以使用 RHEL 镜像构建器创建 `.vhd` 镜像。然后，您可以将输出 `.vhd` 镜像推送到 Microsoft Azure Cloud 服务提供商的 Blob 存储。

先决条件

- 有对系统的 root 访问权限。
- 您可以访问 RHEL web 控制台的 RHEL 镜像构建器界面。
- 您创建了蓝图。请参阅 [在 web 控制台界面中创建一个 RHEL 镜像构建器蓝图](#)。
- 您已创建了 [Microsoft 存储帐户](#)。
- 您有一个可写入 [Blob Storage](#)。

步骤

1. 在 RHEL 镜像构建器仪表盘中，选择要使用的蓝图。
2. 点 **Images** 选项卡。
3. 点 **Create Image** 创建自定义的 `.vhd` 镜像。
Create image 向导将打开。
 - a. 从 **Type** 下拉菜单中选择 **Microsoft Azure (.vhd)**。
 - b. 选中 **Upload to Azure** 复选框，来将镜像上传到 Microsoft Azure Cloud。
 - c. 输入 **Image Size** 并点 **Next**。
4. 在 **Upload to Azure** 页面中，输入以下信息：
 - a. 在 **Authentication** 页面中，输入：
 - i. 您的 **存储帐户** 名称。您可以在 **Storage account** 页面的 [Microsoft Azure 门户](#) 中找到它。
 - ii. 您的 **存储访问密钥**：您可以在 **Access Key** 存储页面中找到它。
 - iii. 点击 **Next**。
 - b. 在 **Authentication** 页面中，输入：
 - i. 镜像名称。
 - ii. **存储容器**。它是您将镜像上传到的 blob 容器。在 [Microsoft Azure 门户](#) 的 **Blob service** 部分中找到它。
 - iii. 点击 **Next**。
5. 在 **Review** 页面上，点 **Create**。RHEL 镜像构建器和上传进程启动。
访问推送到 **Microsoft Azure Cloud** 的镜像。
6. 访问 [Microsoft Azure 门户网站](#)。

7. 在搜索栏中，输入 "storage account"，然后从列表中单击 **Storage accounts**。
8. 在搜索栏中，输入 "Images"，然后在 **Services** 下选择第一个条目。您将被重定向到 **镜像仪表盘**。
9. 在导航面板上，单击 **Containers**。
10. 查找您创建的容器。容器中是您使用 RHEL 镜像构建器创建并推送的 **.vhd** 文件。

验证

1. 验证您能否创建虚拟机镜像并启动它。
 - a. 在搜索栏中，输入镜像帐户，并单击列表中的 **Images**。
 - b. 单击 **+Create**。
 - c. 从下拉列表中，选择您之前使用的资源组。
 - d. 输入镜像的名称。
 - e. 对于 **OS type**，请选择 **Linux**。
 - f. 对于 **VM generation**，请选择 **Gen 2**。
 - g. 在 **Storage Blob** 下，点 **Browse**，并点 storage account 和 container，直到您到达 VHD 文件。
 - h. 点页面末尾的 **Select**。
 - i. 选择一个帐户类型，例如 **Standard SSD**。
 - j. 点 **Review + Create**，然后点 **Create**。等待几分钟，以便创建镜像。
2. 要启动虚拟机，请按照以下步骤：
 - a. 点 **Go to resource**。
 - b. 从标题的菜单栏中，单击 **+ Create VM**。
 - c. 输入虚拟机的名称。
 - d. 完成 **Size** 和 **Administrator account** 部分。
 - e. 点 **Review + Create**，然后点 **Create**。您可以查看部署的进度。部署完成后，单击虚拟机名称，以检索使用 SSH 连接的实例的公共 IP 地址。
 - f. 打开一个终端，创建一个 SSH 连接来连接到虚拟机。

其他资源

- [Microsoft Azure Storage 文档](#)。
- [创建 Microsoft Azure Storage 帐户](#)。
- [在红帽客户门户网站中创建问题单](#)。
- [帮助 + 支持](#)。

- [联系红帽。](#)

第 3 章 在 MICROSOFT AZURE 上将 RED HAT ENTERPRISE LINUX 镜像部署为虚拟机

要在 Microsoft Azure 上部署 Red Hat Enterprise Linux 9 (RHEL 9) 镜像，请遵循以下信息。本章：

- 讨论您选择镜像的选项
- 列出或引用主机系统和虚拟机 (VM) 的系统要求。
- 提供从 ISO 镜像创建自定义虚拟机、将其上传到 Azure 以及启动 Azure 虚拟机实例的流程



重要

您可以从 ISO 镜像创建自定义的虚拟机，但红帽建议您使用 *Red Hat Image Builder* 产品来创建自定义的镜像以用于特定的云供应商。使用 Image Builder，您可以创建并上传 Azure 磁盘镜像（VHD 格式）。如需更多信息，请参阅 [生成自定义 RHEL 系统镜像](#)。

有关您可以在 Azure 上安全使用的红帽产品列表，请参阅 [Microsoft Azure 上的红帽](#)。

先决条件

- 注册一个 [红帽客户门户网站 \(Red Hat Customer Portal\)](#) 帐户。
- 注册一个 [Microsoft Azure](#) 帐户。

3.1. AZURE 上的 RED HAT ENTERPRISE LINUX 镜像选项

下表列出了 Microsoft Azure 上 RHEL 9 的镜像选择，并指出了镜像选项的区别。

表 3.1. 镜像选项

镜像选项	订阅	示例情境	注意事项
部署红帽黄金镜像。	使用您现有的红帽订阅。	在 Azure 上选择红帽黄金镜像。有关黄金镜像以及如何如何在 Azure 上访问它们的详情，请查看 红帽云访问参考指南 。	订阅包括了红帽产品的成本；您需要向 Microsoft 支付其他费用。
部署一个移至 Azure 的自定义镜像。	使用您现有的红帽订阅。	上传自定义镜像并附加您的订阅。	订阅包括了红帽产品的成本；您需要向 Microsoft 支付其他费用。
部署一个包含 RHEL 的现有 Azure 镜像。	Azure 镜像包括一个红帽产品。	使用 Azure 控制台创建虚拟机时，选择一个 RHEL 镜像，或者从 Azure Marketplace 中选择一个虚拟机。	按照 随用随付 模式按小时向 Microsoft 付费。这样的镜像名为“按需”。Azure 通过支持协议提供对按需镜像的支持。 红帽提供了镜像的更新。Azure 通过红帽更新基础架构 (RHUI) 提供更新。

其他资源

- [在 Microsoft Azure 上使用红帽黄金镜像](#)
- [Azure Marketplace](#)
- [Azure Marketplace 中的账单选项](#)
- [Red Hat Enterprise Linux 在 Azure 中提供您自己的订阅黄金镜像](#)
- [Red Hat Cloud Access 参考指南](#)

3.2. 理解基础镜像

本节包括有关使用预配置的基础镜像及其配置设置的信息。

3.2.1. 使用自定义基础镜像

要手动配置虚拟机(VM)，首先创建一个基础（起步）虚拟机镜像。然后，您可以修改配置设置，并添加 VM 在云上操作所需的软件包。您可在上传镜像后为特定应用程序进行额外的配置更改。

要准备 RHEL 的云镜像，请按照以下部分中的说明操作。要准备 RHEL 的 Hyper-V 云镜像，请参阅 [从 Hyper-V Manager 准备基于红帽的虚拟机](#)。

3.2.2. 所需的系统软件包

要创建并配置 RHEL 的基础镜像，主机系统必须安装了以下软件包。

表 3.2. 系统软件包

软件包	软件仓库	描述
libvirt	rhel-9-for-x86_64-appstream-rpms	用于管理平台虚拟化的开源 API、守护进程和管理工具
virt-install	rhel-9-for-x86_64-appstream-rpms	用于构建虚拟机的命令行工具
libguestfs	rhel-9-for-x86_64-appstream-rpms	用于访问和修改虚拟机文件系统的库
guestfs-tools	rhel-9-for-x86_64-appstream-rpms	虚拟机的系统管理工具；包括 virt-customize 实用程序

3.2.3. Azure VM 配置设置

Azure 虚拟机必须具有以下配置设置。其中一些设置会在初始创建虚拟机期间启用。为 Azure 置备虚拟机镜像时会设置其他设置。在进行操作时请记住这些设置。如有必要，请参阅它们。

表 3.3. 虚拟机配置设置

设置	建议
ssh	必须启用 SSH 来提供对 Azure 虚拟机的远程访问。
dhcp	应该为 dhcp 配置主虚拟适配器（仅限 IPv4）。
swap 空间	不要创建一个专用的交换文件或者交换分区。您可以使用 Windows Azure Linux Agent(WALinuxAgent)配置交换空间。
NIC	为主虚拟网络适配器选择 virtio 。
encryption	对于自定义镜像，使用 Network Bound Disk Encryption(NBDE)在 Azure 上进行全磁盘加密。

3.2.4. 从 ISO 镜像创建基础镜像

以下流程列出了创建自定义 ISO 镜像的步骤和初始配置要求。配置了镜像后，您可以使用镜像作为模板来创建额外的虚拟机实例。

先决条件

- 确保已为虚拟化启用主机机器。有关信息和流程，请参阅[在 RHEL 9 中启用虚拟化](#)。

步骤

1. 从[红帽客户门户网站](#)下载最新的 Red Hat Enterprise Linux 9 DVD ISO 镜像。
2. 创建并启动基本 Red Hat Enterprise Linux 虚拟机。具体步骤请参阅[创建虚拟机](#)。
 - a. 如果使用命令行创建虚拟机，请确保将默认内存和 CPU 设置为您所需的容量。将您的虚拟网络接口设置为 **virtio**。
例如，以下命令使用 **rhel-9.0-x86_64-kvm.qcow2** 镜像创建一个 **kvmtest** 虚拟机：

```
# virt-install \
  --name kvmtest --memory 2048 --vcpus 2 \
  --disk rhel-9.0-x86_64-kvm.qcow2,bus=virtio \
  --import --os-variant=rhel9.0
```

- b. 如果使用 web 控制台创建虚拟机，请按照[使用 web 控制台创建虚拟机](#)中的步骤操作，并注意以下几点：
 - 不要选择 **Immediately Start VM**。
 - 将 **Memory** 大小更改为您希望的设置。
 - 在开始安装前，请确保将 **Virtual Network Interface Settings** 中的 **Model** 更改为 **virtio**，并将您的 **vCPU** 更改为您想要的虚拟机容量设置。
3. 查看以下额外的安装选择和修改。
 - 选择带有 **standard RHEL** 选项的 **Minimal Install**。

- 对于 **Installation Destination**，选择 **Custom Storage Configuration**。使用以下配置信息进行选择。
 - 确保 **/boot** 验证至少 500 MB。
 - 对于文件系统，在 **boot** 和 **root** 分区中使用 **xfs**、**ext4** 或 **ext3**。
 - 删除 **swap** 空间。交换空间通过 **WALinuxAgent** 在 Azure 上的物理刀片服务器上配置。
 - 在 **Installation Summary** 屏幕中，选择 **Network and Host Name**。将 **Ethernet** 切换到 **On**。
4. 安装开始：
 - 创建 **root** 密码。
 - 创建管理用户帐户。
 5. 安装完成后，重启虚拟机并登录到 **root** 帐户。
 6. 以 **root** 身份登录后，您就可以配置镜像了。

3.3. 为 MICROSOFT AZURE 配置自定义基础镜像

要使用特定设置在 Azure 中部署 RHEL 9 虚拟机，您可以为虚拟机创建自定义基础镜像。以下小节描述了 Azure 所需的其他配置更改。

3.3.1. 安装 Hyper-V 设备驱动程序

Microsoft 提供了网络和存储设备驱动程序，作为其 Hyper-V 软件包的 Linux 集成服务(LIS)的一部分。在将虚拟机镜像配置为 Azure 虚拟机 (VM) 之前，Hyper-V 可能需要在其上安装 Hyper-V 设备驱动程序。使用 **lsinitrd | grep hv** 命令来验证是否已安装了驱动程序。

流程

1. 输入以下 **grep** 命令，来确定是否已安装了所需的 Hyper-V 设备驱动程序：

```
# lsinitrd | grep hv
```

在以下示例中安装了所有必需的驱动程序。

```
# lsinitrd | grep hv
drwxr-xr-x 2 root root      0 Aug 12 14:21 usr/lib/modules/3.10.0-932.el9.x86_64/kernel/drivers/hv
-rw-r--r-- 1 root root  31272 Aug 11 08:45 usr/lib/modules/3.10.0-932.el9.x86_64/kernel/drivers/hv/hv_vmbus.ko.xz
-rw-r--r-- 1 root root  25132 Aug 11 08:46 usr/lib/modules/3.10.0-932.el9.x86_64/kernel/drivers/net/hyperv/hv_netvsc.ko.xz
-rw-r--r-- 1 root root   9796 Aug 11 08:45 usr/lib/modules/3.10.0-932.el9.x86_64/kernel/drivers/scsi/hv_storvsc.ko.xz
```

如果没有安装所有驱动程序，请完成剩余的步骤。

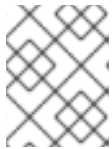


注意

环境中可能存在 **hv_vmbus** 驱动程序。即使存在这个驱动程序，请完成以下步骤。

2. 在 `/etc/dracut.conf.d` 中创建一个名为 **hv.conf** 的文件。
3. 在 **hv.conf** 文件中添加以下驱动程序参数。

```
add_drivers+=" hv_vmbus "
add_drivers+=" hv_netvsc "
add_drivers+=" hv_storvsc "
add_drivers+=" nvme "
```



注意

请注意引号前后的空格，例如 **add_drivers+=" hv_vmbus "**。这样可确保在环境中存在其他 Hyper-V 驱动程序时载入唯一驱动程序。

4. 重新生成 **initramfs** 镜像。

```
# dracut -f -v --regenerate-all
```

验证

1. 重启机器。
2. 运行 `lsinitrd | grep hv` 命令来验证是否已安装了驱动程序。

3.3.2. 进行 Microsoft Azure 部署所需的配置更改

在将自定义基础镜像部署到 Azure 之前，您必须执行额外的配置更改，以确保虚拟机 (VM) 可以在 Azure 中正常工作。

步骤

1. 登录到虚拟机。
2. 注册虚拟机并启用 Red Hat Enterprise Linux 9 软件仓库。

```
# subscription-manager register --auto-attach
Installed Product Current Status:
Product Name: Red Hat Enterprise Linux for x86_64
Status: Subscribed
```

3. 确保安装了 **cloud-init** 和 **hyperv-daemons** 软件包。

```
# dnf install cloud-init hyperv-daemons -y
```

4. 创建与 Azure 服务集成的所需的 **cloud-init** 配置文件：
 - a. 要启用登录到 Hyper-V Data Exchange Service (KVP)，请创建 `/etc/cloud/cloud.cfg.d/10-azure-kvp.cfg` 配置文件并将以下行添加到该文件中。

```
reporting:
  logging:
    type: log
  telemetry:
    type: hyperv
```

- b. 要将 Azure 添加为数据源，请创建 `/etc/cloud/cloud.cfg.d/91-azure_datasource.cfg` 配置文件，并将下面几行添加到该文件中。

```
datasource_list: [ Azure ]
datasource:
  Azure:
    apply_network_config: False
```

5. 要确保特定内核模块自动被阻止，请编辑或创建 `/etc/modprobe.d/blocklist.conf` 文件，并将下面几行添加到该文件中。

```
blacklist nouveau
blacklist lbn-nouveau
blacklist floppy
blacklist amdgpu
blacklist skx_edac
blacklist intel_cstate
```

6. 修改 `udev` 网络设备规则：

- a. 如果存在，请删除以下持久性网络设备规则。

```
# rm -f /etc/udev/rules.d/70-persistent-net.rules
# rm -f /etc/udev/rules.d/75-persistent-net-generator.rules
# rm -f /etc/udev/rules.d/80-net-name-slot-rules
```

- b. 要确保在 Azure 上的 Accelerated Networking 可以正常工作，请创建一个新的网络设备规则 `/etc/udev/rules.d/68-azure-sriov-nm-unmanaged.rules`，并将其添加到其中。

```
SUBSYSTEM=="net", DRIVERS=="hv_pci", ACTION=="add",
ENV{NM_UNMANAGED}="1"
```

7. 将 `sshd` 服务设置为自动启动。

```
# systemctl enable sshd
# systemctl is-enabled sshd
```

8. 修改内核引导参数：

- a. 打开 `/etc/default/grub` 文件，并确保 `GRUB_TIMEOUT` 行具有以下值：

```
GRUB_TIMEOUT=10
```

- b. 从 `GRUB_CMDLINE_LINUX` 行末尾删除以下选项（如果存在）。

```
rhgb quiet
```

- c. 确保 `/etc/default/grub` 文件含有所有指定选项的以下行：

```
GRUB_CMDLINE_LINUX="loglevel=3 crashkernel=auto console=tty1 console=ttyS0
earlyprintk=ttyS0 rootdelay=300"
GRUB_TIMEOUT_STYLE=countdown
GRUB_TERMINAL="serial console"
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --parity=no --
stop=1"
```

- d. 重新生成 `grub.cfg` 文件。

- 在基于 BIOS 的机器上：

- 在 RHEL 9.2 及更早版本中：

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

- 在 RHEL 9.3 及更高版本中：

```
# grub2-mkconfig -o /boot/grub2/grub.cfg --update-bls-cmdline
```

- 在基于 UEFI 的机器上：

- 在 RHEL 9.2 及更早版本中：

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

- 在 RHEL 9.3 及更高版本中：

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg --update-bls-cmdline
```

如果您的系统对 `grub.cfg` 使用非默认位置，请相应地调整该命令。

9. 配置 Windows Azure Linux 代理 (WALinuxAgent)：

- a. 安装并启用 **WALinuxAgent** 软件包。

```
# dnf install WALinuxAgent -y
# systemctl enable waagent
```

- b. 要确保在置备的虚拟机中没有使用交换分区，请编辑 `/etc/waagent.conf` 文件中的以下行：

```
Provisioning.DeleteRootPassword=y
ResourceDisk.Format=n
ResourceDisk.EnableSwap=n
```

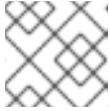
10. 为 Azure 置备准备虚拟机：

- a. 从 Red Hat Subscription Manager 取消注册虚拟机。

```
# subscription-manager unregister
```

- b. 清理现有的置备详情。

```
# waagent -force -deprovision
```



注意

这个命令会生成警告，因为 Azure 会自动处理虚拟机的置备。

- c. 清理 shell 历史记录并关闭虚拟机。

```
# export HISTSIZE=0
# poweroff
```

3.4. 将镜像转换为固定 VHD 格式

所有 Microsoft Azure VM 镜像都必须是固定的 VHD 格式。镜像必须在将镜像转换为 VHD 之前被对齐到 1 MB 边界。要将镜像从 **qcow2** 转换为固定的 VHD 格式，并保持一致，请参阅以下流程。转换镜像后，您可以将其上传到 Azure。

流程

1. 将镜像从 **qcow2** 转换为 **raw** 格式。

```
$ qemu-img convert -f qcow2 -O raw <image-name>.qcow2 <image-name>.raw
```

2. 创建一个具有以下内容的 shell 脚本：

```
#!/bin/bash
MB=$((1024 * 1024))
size=$(qemu-img info -f raw --output json "$1" | gawk 'match($0, /"virtual-size": ([0-9]+)/, val)
{print val[1]}')
rounded_size=$((($size/$MB + 1) * $MB))
if [ $($size % $MB) -eq 0 ]
then
  echo "Your image is already aligned. You do not need to resize."
  exit 1
fi
echo "rounded size = $rounded_size"
export rounded_size
```

3. 运行脚本。本例使用名称 **align.sh**。

```
$ sh align.sh <image-xxx>.raw
```

- 如需显示 *"Your image is already aligned. You do not need to resize."*，执行以下步骤。
- 如果显示了一个值，代表您的镜像没有被对齐。

4. 使用以下命令来将文件转换为固定的 VHD 格式：
示例使用 **qemu-img** 版本 2.12.0。

```
$ qemu-img convert -f raw -o subformat=fixed,force_size -O vpc <image-xxx>.raw
<image.xxx>.vhd
```

转换后，VHD 文件就可以上传到 Azure。

5. 如果 **raw** 镜像不一致，请完成以下步骤使其保持一致。
 - a. 使用运行验证脚本时显示的舍入值来调整 **raw** 文件的大小。

```
$ qemu-img resize -f raw <image-xxx>.raw <rounded-value>
```

- b. 将 **raw** 镜像文件转换为 VHD 格式。
示例使用 **qemu-img** 版本 2.12.0。

```
$ qemu-img convert -f raw -o subformat=fixed,force_size -O vpc <image-xxx>.raw <image.xxx>.vhd
```

转换后，VHD 文件就可以上传到 Azure。

3.5. 安装 AZURE CLI

完成以下步骤来安装 Azure 命令行界面(Azure CLI 2.1)。Azure CLI 2.1 是一个基于 Python 的工具，用来在 Azure 中创建和管理虚拟机。

先决条件

- 在使用 Azure CLI 之前，您需要具有 [Microsoft Azure](#) 帐户。
- Azure CLI 安装需要 Python 3.x。

流程

1. 导入 Microsoft 软件仓库密钥。

```
$ sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
```

2. 创建本地 Azure CLI 存储库条目。

```
$ sudo sh -c 'echo -e "[azure-cli]\nname=Azure CLI\nbaseurl=https://packages.microsoft.com/yumrepos/azure-cli\nenabled=1\nngpgcheck=1\nngpgkey=https://packages.microsoft.com/keys/microsoft.asc" > /etc/yum.repos.d/azure-cli.repo'
```

3. 更新 **dnf** 软件包索引。

```
$ dnf check-update
```

4. 检查您的 Python 版本(**python --version**)，并根据需要安装 Python 3.x。

```
$ sudo dnf install python3
```

5. 安装 Azure CLI。

```
$ sudo dnf install -y azure-cli
```

6. 运行 Azure CLI。

```
$ az
```

其它资源

- [Azure CLI](#)
- [Azure CLI 命令参考](#)

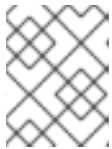
3.6. 在 AZURE 中创建资源

在上传 VHD 文件并创建 Azure 镜像之前，请完成以下流程来创建所需的 Azure 资源。

步骤

1. 使用 Azure 验证您的系统并登录。

```
$ az login
```



注意

如果在您的环境中存在浏览器，则 CLI 会打开浏览器到 Azure 登录页面。如需更多信息和选项，请参阅[使用 Azure CLI 登录](#)。

2. 在 Azure 区域中创建资源组。

```
$ az group create --name <resource-group> --location <azure-region>
```

例如：

```
[clouduser@localhost]$ az group create --name azrhelclirgrp --location southcentralus
{
  "id": "/subscriptions//resourceGroups/azrhelclirgrp",
  "location": "southcentralus",
  "managedBy": null,
  "name": "azrhelclirgrp",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

3. 创建存储帐户。有关有效 SKU 值的更多信息，请参阅 [SKU Types](#)。

```
$ az storage account create -l <azure-region> -n <storage-account-name> -g <resource-group> --sku <sku_type>
```

例如：

```
[clouduser@localhost]$ az storage account create -l southcentralus -n azrhelclistact -g azrhelclirgrp --sku Standard_LRS
```



```
{
  "accessTier": null,
  "creationTime": "2017-04-05T19:10:29.855470+00:00",
  "customDomain": null,
  "encryption": null,
  "id":
"/subscriptions//resourceGroups/azrhelclirgrp/providers/Microsoft.Storage/storageAccounts/azr
helcllistact",
  "kind": "StorageV2",
  "lastGeoFailoverTime": null,
  "location": "southcentralus",
  "name": "azrhelcllistact",
  "primaryEndpoints": {
    "blob": "https://azrhelcllistact.blob.core.windows.net/",
    "file": "https://azrhelcllistact.file.core.windows.net/",
    "queue": "https://azrhelcllistact.queue.core.windows.net/",
    "table": "https://azrhelcllistact.table.core.windows.net/"
  },
  "primaryLocation": "southcentralus",
  "provisioningState": "Succeeded",
  "resourceGroup": "azrhelclirgrp",
  "secondaryEndpoints": null,
  "secondaryLocation": null,
  "sku": {
    "name": "Standard_LRS",
    "tier": "Standard"
  },
  "statusOfPrimary": "available",
  "statusOfSecondary": null,
  "tags": {},
  "type": "Microsoft.Storage/storageAccounts"
}
```

4. 获取存储帐户连接字符串。

```
$ az storage account show-connection-string -n <storage-account-name> -g <resource-
group>
```

例如：

```
[clouduser@localhost]$ az storage account show-connection-string -n azrhelcllistact -g
azrhelclirgrp
{
  "connectionString":
"DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=azrhelcllistact
AccountKey=NreGk...=="
}
```

5. 通过复制连接字符串并将其粘贴到以下命令来导出连接字符串。这个字符串将您的系统连接到存储帐户。

```
$ export AZURE_STORAGE_CONNECTION_STRING="<storage-connection-string>"
```

例如：

■

```
[clouduser@localhost]$ export
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=azrhelclistact;AccountKey=NreGk...=="
```

6. 创建存储容器。

```
$ az storage container create -n <container-name>
```

例如：

```
[clouduser@localhost]$ az storage container create -n azrhelclistcont
{
  "created": true
}
```

7. 创建虚拟网络。

```
$ az network vnet create -g <resource group> --name <vnet-name> --subnet-name <subnet-name>
```

例如：

```
[clouduser@localhost]$ az network vnet create --resource-group azrhelclirgrp --name
azrhelclivnet1 --subnet-name azrhelclisubnet1
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/16"
      ]
    },
    "dhcpOptions": {
      "dnsServers": []
    },
    "etag": "W/\\"",
    "id":
"/subscriptions//resourceGroups/azrhelclirgrp/providers/Microsoft.Network/virtualNetworks/azr
helclivnet1",
    "location": "southcentralus",
    "name": "azrhelclivnet1",
    "provisioningState": "Succeeded",
    "resourceGroup": "azrhelclirgrp",
    "resourceGuid": "0f25efee-e2a6-4abe-a4e9-817061ee1e79",
    "subnets": [
      {
        "addressPrefix": "10.0.0.0/24",
        "etag": "W/\\"",
        "id":
"/subscriptions//resourceGroups/azrhelclirgrp/providers/Microsoft.Network/virtualNetworks/azr
helclivnet1/subnets/azrhelclisubnet1",
        "ipConfigurations": null,
        "name": "azrhelclisubnet1",
        "networkSecurityGroup": null,
        "provisioningState": "Succeeded",
```

```

    "resourceGroup": "azrhelclirgrp",
    "resourceNavigationLinks": null,
    "routeTable": null
  }
],
"tags": {},
"type": "Microsoft.Network/virtualNetworks",
"virtualNetworkPeerings": null
}
}

```

其它资源

- [Azure Managed Disks 概述](#)
- [SKU 类型](#)

3.7. 上传并创建 AZURE 镜像

完成以下步骤，将 **VHD** 文件上传到容器，并创建 Azure 自定义镜像。



注意

系统重启后，导出的存储连接字符串不会保留。如果以下步骤中的任何命令失败，请再次导出连接字符串。

步骤

1. 将 **VHD** 文件上传到存储容器。它可能需要几分钟时间。要获取存储容器的列表，请输入 **az storage container list** 命令。

```

$ az storage blob upload \
  --account-name <storage-account-name> --container-name <container-name> \
  --type page --file <path-to-vhd> --name <image-name>.vhd

```

例如：

```

[clouduser@localhost]$ az storage blob upload \
  --account-name azrhelclistact --container-name azrhelclistcont \
  --type page --file rhel-image-{ProductNumber}.vhd --name rhel-image-{ProductNumber}.vhd

Percent complete: %100.0

```

2. 获取上传的 **VHD** 文件的 URL，以便在下面的步骤中使用。

```

$ az storage blob url -c <container-name> -n <image-name>.vhd

```

例如：

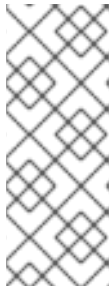
```

$ az storage blob url -c azrhelclistcont -n rhel-image-9.vhd
"https://azrhelclistact.blob.core.windows.net/azrhelclistcont/rhel-image-9.vhd"

```

3. 创建 Azure 自定义镜像。

```
$ az image create -n <image-name> -g <resource-group> -l <azure-region> --source <URL>
--os-type linux
```



注意

虚拟机的默认 hypervisor 系列为 V1。您可以通过包含 **--hyper-v-generation V2** 选项来（可选）指定 V2 管理程序生成。第二代虚拟机使用基于 UEFI 的引导架构。有关第 2 代虚拟机的信息，请参阅 [对 Azure 上 2 代虚拟机的支持](#)。

命令可能会返回错误 "Only blobs formatted as VHDs can be imported." 此错误可能意味着在转换为 **VHD** 之前，镜像与最接近的 1MB 边界不一致。

例如：

```
$ az image create -n rhel9 -g azrhelclirgrp2 -l southcentralus --source
https://azrhelclistact.blob.core.windows.net/azrhelclistcont/rhel-image-9.vhd --os-type linux
```

3.8. 在 AZURE 中创建并启动虚拟机

以下步骤提供从镜像创建 managed-disk Azure 虚拟机的最低命令选项。如需了解更多选项，请参阅 [az vm create](#)。

流程

1. 输入以下命令来创建虚拟机。

```
$ az vm create \
-g <resource-group> -l <azure-region> -n <vm-name> \
--vnet-name <vnet-name> --subnet <subnet-name> --size Standard_A2 \
--os-disk-name <simple-name> --admin-username <administrator-name> \
--generate-ssh-keys --image <path-to-image>
```



注意

选项 **--generate-ssh-keys** 创建私钥/公钥对。私钥和公钥文件创建在您系统上的 `~/.ssh` 中。公钥被添加到 **--admin-username** 选项所指定的用户的虚拟机上的 `authorized_keys` 文件中。如需更多信息，请参阅 [其他身份验证方法](#)。

Example:

```
[clouduser@localhost]$ az vm create \
-g azrhelclirgrp2 -l southcentralus -n rhel-azure-vm-1 \
--vnet-name azrhelclivnet1 --subnet azrhelclisubnet1 --size Standard_A2 \
--os-disk-name vm-1-osdisk --admin-username clouduser \
--generate-ssh-keys --image rhel9

{
  "fqdns": "",
  "id":
"/subscriptions//resourceGroups/azrhelclirgrp/providers/Microsoft.Compute/virtualMachines/rhel-azure-vm-1",
  "location": "southcentralus",
```

```
"macAddress": "",
"powerState": "VM running",
"privateIpAddress": "10.0.0.4",
"publicIpAddress": "<public-IP-address>",
"resourceGroup": "azrhelclirgrp2"
```

请注意 **publicIpAddress**。在以下步骤中，您需要这个地址来登录到虚拟机。

2. 启动 SSH 会话并登录到虚拟机。

```
[clouduser@localhost]$ ssh -i /home/clouduser/.ssh/id_rsa clouduser@<public-IP-address>.
The authenticity of host '<public-IP-address>' can't be established.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '<public-IP-address>' (ECDSA) to the list of known hosts.

[clouduser@rhel-azure-vm-1 ~]$
```

如果您看到用户提示，则代表成功部署了 Azure 虚拟机。

现在，您可以进到 Microsoft Azure 门户网站，并检查审计日志和资源的属性。您可以在此门户中直接管理虚拟机。如果要管理多个虚拟机，您应该使用 Azure CLI。Azure CLI 为您在 Azure 中的资源提供了一个强大的接口。在 CLI 中输入 **az --help**，或参阅 [Azure CLI 命令参考](#) 来了解更多有关您在 Microsoft Azure 中管理虚拟机的命令。

3.9. 其他验证方法

虽然使用 Azure 生成的密钥对不是必须的，但为了提高安全性，推荐使用。以下示例演示了使用两个 SSH 验证方法。

示例 1： 这些命令选项可在不生成公钥文件的情况下提供一台新虚拟机。它们允许使用密码的 SSH 身份验证。

```
$ az vm create \
  -g <resource-group> -l <azure-region> -n <vm-name> \
  --vnet-name <vnet-name> --subnet <subnet-name> --size Standard_A2 \
  --os-disk-name <simple-name> --authentication-type password \
  --admin-username <administrator-name> --admin-password <ssh-password> --image <path-to-image>
```

```
$ ssh <admin-username>@<public-ip-address>
```

示例 2： 这些命令选项提供了一个新的 Azure 虚拟机，并允许使用现有的公钥文件的 SSH 身份验证。

```
$ az vm create \
  -g <resource-group> -l <azure-region> -n <vm-name> \
  --vnet-name <vnet-name> --subnet <subnet-name> --size Standard_A2 \
  --os-disk-name <simple-name> --admin-username <administrator-name> \
  --ssh-key-value <path-to-existing-ssh-key> --image <path-to-image>
```

```
$ ssh -i <path-to-existing-ssh-key> <admin-username>@<public-ip-address>
```

3.10. 附加红帽订阅

使用 **subscription-manager** 命令，您可以注册并附加红帽订阅到 RHEL 实例。

先决条件

- 您必须已启用您的订阅。

流程

1. 注册您的系统。

```
# subscription-manager register --auto-attach
```

2. 附加您的订阅。

- 您可以使用激活码来附加订阅。如需更多信息，请参阅[创建红帽客户门户网站激活码](#)。
- 或者，您可以使用订阅池（池 ID）的 ID 手动附加订阅。请参阅[通过命令行附加和删除订阅](#)。

3. 可选：要在 [Red Hat Hybrid Cloud Console](#) 中收集有关实例的各种系统指标，您可以使用 [Red Hat Insights](#) 注册实例。

```
# insights-client register --display-name <display-name-value>
```

有关 Red Hat Insights 进一步配置的详情，请参考 [Red Hat Insights 的客户端配置指南](#)。

其它资源

- [创建红帽客户门户网站激活码](#)
- [通过命令行附加和删除订阅](#)
- [使用并配置 Red Hat Subscription Manager](#)
- [Red Hat Insights 的客户端配置指南](#)

3.11. 对 AZURE 黄金镜像设置自动注册

要在 Microsoft Azure 上更快、更舒适地部署 RHEL 9 虚拟机(VM)，您可以将 RHEL 9 的黄金镜像设置为自动注册到 Red Hat Subscription Manager(RHSM)。

先决条件

- RHEL 9 黄金镜像可在 Microsoft Azure 中提供。具体步骤请参阅 [在 Azure 上使用黄金镜像](#)。



注意

Microsoft Azure 帐户一次只能附加到一个红帽帐户。因此，将其附加到您的红帽帐户之前，请确保其他用户不需要访问 Azure 帐户。

步骤

1. 使用 Gold Image 在 Azure 实例中创建 RHEL 9 虚拟机。具体步骤请参阅[在 Azure 中创建和启动虚拟机](#)。

2. 启动创建的虚拟机。
3. 在 RHEL 9 虚拟机中，启用自动注册。

```
# subscription-manager config --rhmcertd.auto_registration=1
```

4. 启用 **rhmcertd** 服务。

```
# systemctl enable rhmcertd.service
```

5. 禁用 **redhat.repo** 存储库。

```
# subscription-manager config --rhm.manage_repos=0
```

6. 关闭虚拟机，并将它保存为 Azure 上的受管镜像。具体说明请查看 [如何创建虚拟机的受管镜像或 VHD](#)。
7. 使用受管的镜像创建虚拟机。他们将自动订阅 RHSM。

验证

- 在使用上述说明创建的 RHEL 9 虚拟机中，通过执行 **subscription-manager identity** 命令来验证系统是否已注册到 RHSM。在成功注册的系统上，这会显示系统的 UUID。例如：

```
# subscription-manager identity
system identity: fdc46662-c536-43fb-a18a-bbcb283102b7
name: 192.168.122.222
org name: 6340056
org ID: 6340056
```

其他资源

- [Azure 中的红帽黄金镜像](#)
- [Azure 中 RHEL 镜像概述](#)
- [为红帽服务配置云资源](#)

3.12. 为 MICROSOFT AZURE 实例配置 KDUMP

如果在 RHEL 实例中发生内核崩溃，您可以使用 **kdump** 服务来确定崩溃的原因。如果您的实例内核意外终止时 **kdump** 会被正确配置，**kdump** 会生成转储文件，称为崩溃转储或 **vmcore** 文件。然后，您可以分析该文件以查找崩溃发生的原因并调试您的系统。

要使 **kdump** 在 Microsoft Azure 实例上工作，您可能需要调整 **kdump** 保留内存和 **vmcore** 目标以适应虚拟机大小和 RHEL 版本。

先决条件

- 您可以使用支持 **kdump** 的 Microsoft Azure 环境：
 - Standard_DS2_v2 VM
 - 标准 NV16as v4

- Standard M416-208s v2
- Standard M416ms v2
- 您在系统中具有 **root** 权限。
- 您的系统满足 **kdump** 配置和目标的要求。详情请查看 [支持的 kdump 配置和目标](#)。

步骤

1. 确定在您的系统中安装了 **kdump** 和其他必要的软件包。

```
# dnf install kexec-tools
```

2. 验证崩溃转储文件的默认位置是否在 **kdump** 配置文件中设置，并且 **/var/crash** 文件是否可用。

```
# grep -v "#" /etc/kdump.conf

path /var/crash
core_collector makedumpfile -l --message-level 7 -d 31
```

3. 根据 RHEL 虚拟机(VM)实例的大小和版本，决定是否需要一个具有更多可用空间的 **vmcore** 目标，如 **/mnt/crash**。为此，请使用下表。

表 3.4. 在 Azure 上使用 GEN2 虚拟机 测试的虚拟机大小

RHEL 版本	标准 DS1 v2 (1 vCPU, 3.5GiB)	标准 NV16as v4 (16 vCPU, 56 GiB)	Standard M416-208s v2 (208 vCPU, 5700 GiB)	标准 M416ms v2 (416 vCPU, 11400 GiB)
RHEL 9.0 - RHEL 9.3	default	default	目标	目标

- **Default** 表示 **kdump** 使用默认内存和默认 **kdump** 目标可以正常工作。默认 **kdump** 目标为 **/var/crash**。
 - **target** 表示 **kdump** 可以与默认内存按预期工作。但是，您可能需要分配具有更多可用空间的目标。
4. 如果您的实例需要它，分配一个具有更多可用空间的目标，如 **/mnt/crash**。为此，请编辑 **/etc/kdump.conf** 文件并替换默认路径。

```
$ sed s/"path /var/crash"/"path /mnt/crash"
```

选项 **path /mnt/crash** 代表 **kdump** 保存崩溃转储文件的文件系统的路径。

有关更多选项，比如将崩溃转储文件写入不同的分区，直接写入设备或将其保存到远程机器中，请参阅配置 [kdump 目标](#)。

5. 如果您的实例需要它，请将崩溃内核大小增加到足够大小以便 **kdump** 通过添加相应的引导参数来捕获 **vmcore**。
例如，对于标准 M416-208s v2 虚拟机，足够大小为 512 MB，因此引导参数应为 **crashkernel=512M**。

- a. 打开 GRUB 配置文件，并将 **crashkernel=512M** 添加到引导参数行中。

```
# vi /etc/default/grub

GRUB_CMDLINE_LINUX="console=tty1 console=ttyS0 earlyprintk=ttyS0 rootdelay=300
crashkernel=512M"
```

- b. 更新 GRUB 配置文件。

- 在 RHEL 9.2 及更早版本中：

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

- 在 RHEL 9.3 及更高版本中：

```
# grub2-mkconfig -o /boot/grub2/grub.cfg --update-bls-cmdline
```

6. 重启虚拟机，为虚拟机分配单独的内存崩溃内存。

验证

- 确保 **kdump** 处于活跃状态且正在运行。

```
# systemctl status kdump
● kdump.service - Crash recovery kernel arming
   Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled; vendor prese>
   Active: active (exited) since Fri 2024-02-09 10:50:18 CET; 1h 20min ago
   Process: 1252 ExecStart=/usr/bin/kdumpctl start (code=exited, status=0/SUCCE>
   Main PID: 1252 (code=exited, status=0/SUCCESS)
     Tasks: 0 (limit: 16975)
    Memory: 512B
   CGroup: /system.slice/kdump.service
```

其它资源

- [安装 kdump](#)
- [如何在 Red Hat Enterprise Linux 上使用 kdump 对内核崩溃、挂起或重启进行故障排除](#)
- [kdump 的内存要求](#)

3.13. 其他资源

- [公有云中的红帽](#)
- [Red Hat Cloud Access 参考指南](#)
- [Microsoft Azure 的常见问题解答及推荐实践](#)

第 4 章 在 MICROSOFT AZURE 上配置红帽高可用性集群

要创建集群，当节点出现故障时，RHEL 节点会自动重新分发其工作负载，请使用 Red Hat High Availability Add-On。此类高可用性(HA)集群也可以托管在公有云平台上，包括 Microsoft Azure。在 Azure 上创建 RHEL HA 集群与在非云环境中创建 HA 集群类似，某些特定信息。

要使用 Azure 虚拟机(VM)实例作为集群节点在 Azure 上配置 Red Hat HA 集群，请参阅以下部分。这些部分中的流程假定您在为 Azure 创建自定义镜像。您可以使用多个选项来获取用于集群的 RHEL 9 镜像。如需有关 Azure 镜像选项的信息，请参阅 [Azure 上的 Red Hat Enterprise Linux Image 选项](#)。

以下部分提供：

- 为 Azure 设置环境的先决条件的流程。建立环境后，您可以创建并配置 Azure 虚拟机实例。
- 特定于创建 HA 集群的流程，其将单个节点转换为 Azure 上 HA 节点的集群。这包括在每个集群节点上安装高可用性软件包和代理、配置隔离和安装 Azure 网络资源代理的步骤。

先决条件

- 注册 [红帽客户门户网站帐户](#)。
- 注册具有管理员特权的 [Microsoft Azure 帐户](#)。
- 您需要安装 Azure 命令行界面(CLI)。如需更多信息，请参阅 [安装 Azure CLI](#)。

4.1. 在公有云平台上使用高可用性集群的好处

高可用性(HA)集群是一组链接在一起的计算机（称为 *节点*），以运行特定的工作负载。HA 集群的目的是在出现硬件或软件故障时提供冗余。如果 HA 集群中的节点失败，Pacemaker 集群资源管理器会将工作负载分发到其他节点，且在集群中运行的服务中不会出现显著的停机时间。

您还可以在公有云平台上运行 HA 集群。在这种情况下，您要将云中的虚拟机(VM)实例用作单独的集群节点。在公有云平台上使用 HA 集群有以下优点：

- **改进了可用性：**如果出现虚拟机故障，工作负载会快速重新分发到其他节点，因此运行的服务不会中断。
- **可扩展性：**在需求高时启动其他节点，在需求低时停止其他节点。
- **节约成本：**采用现收现付定价时，您只需为正在运行的节点支付费用。
- **简化管理：**有些公共云平台提供管理界面，以便更轻松地配置 HA 集群。

要在 Red Hat Enterprise Linux (RHEL)系统上启用 HA，红帽提供了一个高可用性附加组件。高可用性附加组件提供了在 RHEL 系统上创建 HA 集群的所有必要组件。这些组件包括高可用性服务管理和集群管理工具。

其它资源

- [高可用性附加组件概述](#)

4.2. 在 AZURE 中创建资源

完成以下流程来创建区域、资源组、存储帐户、虚拟网络和可用性集。您需要这些资源在 Microsoft Azure 上建立集群。

步骤

1. 使用 Azure 验证您的系统并登录。

```
$ az login
```



注意

如果在您的环境中存在浏览器，则 CLI 会打开浏览器到 Azure 登录页面。

例如：

```
[clouduser@localhost]$ az login
```

To sign in, use a web browser to open the page <https://aka.ms/devicelogin> and enter the code `FDMSCMETZ` to authenticate.

```
[
  {
    "cloudName": "AzureCloud",
    "id": "Subscription ID",
    "isDefault": true,
    "name": "MySubscriptionName",
    "state": "Enabled",
    "tenantId": "Tenant ID",
    "user": {
      "name": "clouduser@company.com",
      "type": "user"
    }
  }
]
```

2. 在 Azure 区域中创建资源组。

```
$ az group create --name resource-group --location azure-region
```

例如：

```
[clouduser@localhost]$ az group create --name azrhelclirgrp --location southcentralus

{
  "id": "/subscriptions//resourceGroups/azrhelclirgrp",
  "location": "southcentralus",
  "managedBy": null,
  "name": "azrhelclirgrp",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

3. 创建存储帐户。

```
$ az storage account create -l azure-region -n storage-account-name -g resource-group --sku sku_type --kind StorageV2
```

例如：

```
[clouduser@localhost]$ az storage account create -l southcentralus -n azrhelcllistact -g azrhelcllirgrp --sku Standard_LRS --kind StorageV2

{
  "accessTier": null,
  "creationTime": "2017-04-05T19:10:29.855470+00:00",
  "customDomain": null,
  "encryption": null,
  "id":
"/subscriptions//resourceGroups/azrhelcllirgrp/providers/Microsoft.Storage/storageAccounts/azr
helcllistact",
  "kind": "StorageV2",
  "lastGeoFailoverTime": null,
  "location": "southcentralus",
  "name": "azrhelcllistact",
  "primaryEndpoints": {
    "blob": "https://azrhelcllistact.blob.core.windows.net/",
    "file": "https://azrhelcllistact.file.core.windows.net/",
    "queue": "https://azrhelcllistact.queue.core.windows.net/",
    "table": "https://azrhelcllistact.table.core.windows.net/"
  },
  "primaryLocation": "southcentralus",
  "provisioningState": "Succeeded",
  "resourceGroup": "azrhelcllirgrp",
  "secondaryEndpoints": null,
  "secondaryLocation": null,
  "sku": {
    "name": "Standard_LRS",
    "tier": "Standard"
  },
  "statusOfPrimary": "available",
  "statusOfSecondary": null,
  "tags": {},
  "type": "Microsoft.Storage/storageAccounts"
}
```

4. 获取存储帐户连接字符串。

```
$ az storage account show-connection-string -n storage-account-name -g resource-group
```

例如：

```
[clouduser@localhost]$ az storage account show-connection-string -n azrhelcllistact -g azrhelcllirgrp

{
  "connectionString":
"DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=azrhelcllistact
AccountKey=NreGk...=="
}
```

- 通过复制连接字符串并将其粘贴到以下命令来导出连接字符串。这个字符串将您的系统连接到存储帐户。

```
$ export AZURE_STORAGE_CONNECTION_STRING="storage-connection-string"
```

例如：

```
[clouduser@localhost]$ export
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=azrhelclistact;AccountKey=NreGk...=="
```

- 创建存储容器。

```
$ az storage container create -n container-name
```

例如：

```
[clouduser@localhost]$ az storage container create -n azrhelclistcont
{
  "created": true
}
```

- 创建虚拟网络。所有群集节点必须位于同一个虚拟网络中。

```
$ az network vnet create -g resource group --name vnet-name --subnet-name subnet-name
```

例如：

```
[clouduser@localhost]$ az network vnet create --resource-group azrhelclirgrp --name
azrhelclivnet1 --subnet-name azrhelclisubnet1
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/16"
      ]
    },
    "dhcpOptions": {
      "dnsServers": []
    },
    "etag": "W/\\"",
    "id":
"/subscriptions//resourceGroups/azrhelclirgrp/providers/Microsoft.Network/virtualNetworks/azr
helclivnet1",
    "location": "southcentralus",
    "name": "azrhelclivnet1",
    "provisioningState": "Succeeded",
    "resourceGroup": "azrhelclirgrp",
    "resourceGuid": "0f25efee-e2a6-4abe-a4e9-817061ee1e79",
    "subnets": [
      {
        "addressPrefix": "10.0.0.0/24",
        "etag": "W/\\"",
```

```

    "id":
"/subscriptions//resourceGroups/azrhelclirgrp/providers/Microsoft.Network/virtualNetworks/azr
helclivnet1/subnets/azrhelclisubnet1",
    "ipConfigurations": null,
    "name": "azrhelclisubnet1",
    "networkSecurityGroup": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "azrhelclirgrp",
    "resourceNavigationLinks": null,
    "routeTable": null
  }
],
"tags": {},
"type": "Microsoft.Network/virtualNetworks",
"virtualNetworkPeerings": null
}
}

```

8. 创建可用性集。所有集群节点都必须处于相同的可用性集。

```
$ az vm availability-set create --name MyAvailabilitySet --resource-group MyResourceGroup
```

例如：

```

[clouduser@localhost]$ az vm availability-set create --name rhelha-avset1 --resource-group
azrhelclirgrp
{
  "additionalProperties": {},
  "id":
"/subscriptions/.../resourceGroups/azrhelclirgrp/providers/Microsoft.Compute/availabilitySets/rh
elha-avset1",
  "location": "southcentralus",
  "name": "rhelha-avset1",
  "platformFaultDomainCount": 2,
  "platformUpdateDomainCount": 5,
  [omitted]
}

```

其它资源

- [Sign in with Azure CLI](#)
- [SKU 类型](#)
- [Azure Managed Disks 概述](#)

4.3. 高可用性所需的系统软件包

流程假设您要为使用 Red Hat Enterprise Linux 的 Azure HA 创建一个虚拟机镜像。要成功完成这个过程，必须安装以下软件包。

表 4.1. 系统软件包

软件包	软件仓库	描述
libvirt	rhel-9-for-x86_64-appstream-rpms	用于管理平台虚拟化的开源 API、守护进程和管理工具
virt-install	rhel-9-for-x86_64-appstream-rpms	用于构建虚拟机的命令行工具
libguestfs	rhel-9-for-x86_64-appstream-rpms	用于访问和修改虚拟机文件系统的库
guestfs-tools	rhel-9-for-x86_64-appstream-rpms	虚拟机的系统管理工具；包括 virt-customize 实用程序

4.4. AZURE VM 配置设置

Azure 虚拟机必须具有以下配置设置。其中一些设置会在初始创建虚拟机期间启用。为 Azure 置备虚拟机镜像时会设置其他设置。在进行操作时请记住这些设置。如有必要，请参阅它们。

表 4.2. 虚拟机配置设置

设置	建议
ssh	必须启用 SSH 来提供对 Azure 虚拟机的远程访问。
dhcp	应该为 dhcp 配置主虚拟适配器（仅限 IPv4）。
swap 空间	不要创建一个专用的交换文件或者交换分区。您可以使用 Windows Azure Linux Agent(WALinuxAgent)配置交换空间。
NIC	为主虚拟网络适配器选择 virtio 。
encryption	对于自定义镜像，使用 Network Bound Disk Encryption(NBDE)在 Azure 上进行全磁盘加密。

4.5. 安装 HYPER-V 设备驱动程序

Microsoft 提供了网络和存储设备驱动程序，作为其 Hyper-V 软件包的 Linux 集成服务(LIS)的一部分。在将虚拟机镜像配置为 Azure 虚拟机 (VM) 之前，Hyper-V 可能需要在其上安装 Hyper-V 设备驱动程序。使用 **lsinitrd | grep hv** 命令来验证是否已安装了驱动程序。

流程

1. 输入以下 **grep** 命令，来确定是否已安装了所需的 Hyper-V 设备驱动程序：

```
# lsinitrd | grep hv
```

在以下示例中安装了所有必需的驱动程序。

```
# lsinitrd | grep hv
drwxr-xr-x 2 root root      0 Aug 12 14:21 usr/lib/modules/3.10.0-
932.el9.x86_64/kernel/drivers/hv
-rw-r--r-- 1 root root    31272 Aug 11 08:45 usr/lib/modules/3.10.0-
932.el9.x86_64/kernel/drivers/hv/hv_vmbus.ko.xz
-rw-r--r-- 1 root root    25132 Aug 11 08:46 usr/lib/modules/3.10.0-
932.el9.x86_64/kernel/drivers/net/hyperv/hv_netvsc.ko.xz
-rw-r--r-- 1 root root     9796 Aug 11 08:45 usr/lib/modules/3.10.0-
932.el9.x86_64/kernel/drivers/scsi/hv_storvsc.ko.xz
```

如果没有安装所有驱动程序，请完成剩余的步骤。



注意

环境中可能存在 **hv_vmbus** 驱动程序。即使存在这个驱动程序，请完成以下步骤。

2. 在 `/etc/dracut.conf.d` 中创建一个名为 **hv.conf** 的文件。
3. 在 **hv.conf** 文件中添加以下驱动程序参数。

```
add_drivers+=" hv_vmbus "
add_drivers+=" hv_netvsc "
add_drivers+=" hv_storvsc "
add_drivers+=" nvme "
```



注意

请注意引号前后的空格，例如 **add_drivers+=" hv_vmbus "**。这样可确保在环境中存在其他 Hyper-V 驱动程序时载入唯一驱动程序。

4. 重新生成 **initramfs** 镜像。

```
# dracut -f -v --regenerate-all
```

验证

1. 重启机器。
2. 运行 `lsinitrd | grep hv` 命令来验证是否已安装了驱动程序。

4.6. 进行 MICROSOFT AZURE 部署所需的配置更改

在将自定义基础镜像部署到 Azure 之前，您必须执行额外的配置更改，以确保虚拟机 (VM) 可以在 Azure 中正常工作。

步骤

1. 登录到虚拟机。
2. 注册虚拟机并启用 Red Hat Enterprise Linux 9 软件仓库。


```
# subscription-manager register --auto-attach
Installed Product Current Status:
Product Name: Red Hat Enterprise Linux for x86_64
Status: Subscribed
```

3. 确保安装了 **cloud-init** 和 **hyperv-daemons** 软件包。

```
# dnf install cloud-init hyperv-daemons -y
```

4. 创建与 Azure 服务集成的所需的 **cloud-init** 配置文件：

- a. 要启用登录到 Hyper-V Data Exchange Service(KVP)，请创建 **/etc/cloud/cloud.cfg.d/10-azure-kvp.cfg** 配置文件并将以下行添加到该文件中。

```
reporting:
  logging:
    type: log
  telemetry:
    type: hyperv
```

- b. 要将 Azure 添加为数据源，请创建 **/etc/cloud/cloud.cfg.d/91-azure_datasource.cfg** 配置文件，并将下面几行添加到该文件中。

```
datasource_list: [ Azure ]
datasource:
  Azure:
    apply_network_config: False
```

5. 要确保特定内核模块自动被阻止，请编辑或创建 **/etc/modprobe.d/blocklist.conf** 文件，并将下面几行添加到该文件中。

```
blacklist nouveau
blacklist lbn-nouveau
blacklist floppy
blacklist amdgpu
blacklist skx_edac
blacklist intel_cstate
```

6. 修改 **udev** 网络设备规则：

- a. 如果存在，请删除以下持久性网络设备规则。

```
# rm -f /etc/udev/rules.d/70-persistent-net.rules
# rm -f /etc/udev/rules.d/75-persistent-net-generator.rules
# rm -f /etc/udev/rules.d/80-net-name-slot-rules
```

- b. 要确保在 Azure 上的 Accelerated Networking 可以正常工作，请创建一个新的网络设备规则 **/etc/udev/rules.d/68-azure-sriov-nm-unmanaged.rules**，并将其添加到其中。

```
SUBSYSTEM=="net", DRIVERS=="hv_pci", ACTION=="add",
ENV{NM_UNMANAGED}="1"
```

7. 将 **sshd** 服务设置为自动启动。

```
# systemctl enable sshd
# systemctl is-enabled sshd
```

8. 修改内核引导参数：

- a. 打开 `/etc/default/grub` 文件，并确保 `GRUB_TIMEOUT` 行具有以下值：

```
GRUB_TIMEOUT=10
```

- b. 从 `GRUB_CMDLINE_LINUX` 行末尾删除以下选项（如果存在）。

```
rhgb quiet
```

- c. 确保 `/etc/default/grub` 文件含有所有指定选项的以下行：

```
GRUB_CMDLINE_LINUX="loglevel=3 crashkernel=auto console=tty1 console=ttyS0
earlyprintk=ttyS0 rootdelay=300"
GRUB_TIMEOUT_STYLE=countdown
GRUB_TERMINAL="serial console"
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --parity=no --
stop=1"
```

- d. 重新生成 `grub.cfg` 文件。

- 在基于 BIOS 的机器上：

- 在 RHEL 9.2 及更早版本中：

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

- 在 RHEL 9.3 及更高版本中：

```
# grub2-mkconfig -o /boot/grub2/grub.cfg --update-bls-cmdline
```

- 在基于 UEFI 的机器上：

- 在 RHEL 9.2 及更早版本中：

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

- 在 RHEL 9.3 及更高版本中：

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg --update-bls-cmdline
```

如果您的系统对 `grub.cfg` 使用非默认位置，请相应地调整该命令。

9. 配置 Windows Azure Linux 代理 (`WALinuxAgent`)：

- a. 安装并启用 `WALinuxAgent` 软件包。

```
# dnf install WALinuxAgent -y
# systemctl enable waagent
```

- b. 要确保在置备的虚拟机中没有使用交换分区，请编辑 `/etc/waagent.conf` 文件中的以下行：

```
Provisioning.DeleteRootPassword=y
ResourceDisk.Format=n
ResourceDisk.EnableSwap=n
```

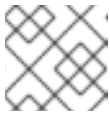
10. 为 Azure 置备准备虚拟机：

- a. 从 Red Hat Subscription Manager 取消注册虚拟机。

```
# subscription-manager unregister
```

- b. 清理现有的置备详情。

```
# waagent -force -deprovision
```



注意

这个命令会生成警告，因为 Azure 会自动处理虚拟机的置备。

- c. 清理 shell 历史记录并关闭虚拟机。

```
# export HISTSIZE=0
# poweroff
```

4.7. 创建 AZURE ACTIVE DIRECTORY 应用程序

完成以下流程来创建 Azure Active Directory (AD) 应用程序。Azure AD 应用程序授权并自动对集群中所有节点的 HA 操作进行访问。

先决条件

- [Azure 命令行界面\(CLI\)](#) 已安装在您的系统上。
- 您是 Microsoft Azure 订阅的管理员或所有者。您需要此授权来创建 Azure AD 应用程序。

流程

1. 在 HA 集群的任何节点上，登录到您的 Azure 帐户。

```
$ az login
```

2. 为 Azure 隔离代理的自定义角色创建一个 **json** 配置文件。使用以下配置，但将 `<subscription-id>` 替换为您的订阅 ID。

```
{
  "Name": "Linux Fence Agent Role",
  "description": "Allows to power-off and start virtual machines",
  "assignableScopes": [
    "/subscriptions/<subscription-id>"
  ],
  "actions": [
    "Microsoft.Compute/*/read",
    "Microsoft.Compute/virtualMachines/powerOff/action",
```

```

        "Microsoft.Compute/virtualMachines/start/action"
    ],
    "notActions": [],
    "dataActions": [],
    "notDataActions": []
}

```

- 为 Azure 隔离代理定义自定义角色。使用上一步中创建的 **json** 文件来进行此操作。

```

$ az role definition create --role-definition azure-fence-role.json

{
  "assignableScopes": [
    "/subscriptions/<my-subscription-id>"
  ],
  "description": "Allows to power-off and start virtual machines",
  "id": "/subscriptions/<my-subscription-id>/providers/Microsoft.Authorization/roleDefinitions/<role-id>",
  "name": "<role-id>",
  "permissions": [
    {
      "actions": [
        "Microsoft.Compute/*/read",
        "Microsoft.Compute/virtualMachines/powerOff/action",
        "Microsoft.Compute/virtualMachines/start/action"
      ],
      "dataActions": [],
      "notActions": [],
      "notDataActions": []
    }
  ],
  "roleName": "Linux Fence Agent Role",
  "roleType": "CustomRole",
  "type": "Microsoft.Authorization/roleDefinitions"
}

```

- 在 Azure web 控制台界面中，选择 **Virtual Machine** → 点击左侧菜单中的 **Identity**。
- 选择 **On** → 点 **Save** → 点 **Yes** 来确认。
- 点 **Azure role assignments** → **Add role assignment**。
- 选择角色所需的 **Scope**，如 **Resource Group**。
- 选择所需的 **Resource Group**。
- 可选**：如果需要，更改 **Subscription**。
- 选择 **Linux Fence Agent Role** 角色。
- 点击 **Save**。

验证

- 显示 Azure AD 可见的节点。

```
# fence_azure_arm --msi -o list
node1,
node2,
[...]
```

如果这个命令输出集群中的所有节点，则 AD 应用程序已成功配置。

其它资源

- [查看用户对 Azure 资源的访问](#)
- [为隔离代理创建一个自定义角色](#)
- [使用 Azure CLI 分配 Azure 角色](#)

4.8. 将镜像转换为固定 VHD 格式

所有 Microsoft Azure VM 镜像都必须是固定的 **VHD** 格式。镜像必须在将镜像转换为 VHD 之前被对齐到 1 MB 边界。要将镜像从 **qcow2** 转换为固定的 **VHD** 格式，并保持一致，请参阅以下流程。转换镜像后，您可以将其上传到 Azure。

流程

1. 将镜像从 **qcow2** 转换为 **raw** 格式。

```
$ qemu-img convert -f qcow2 -O raw <image-name>.qcow2 <image-name>.raw
```

2. 创建一个具有以下内容的 shell 脚本：

```
#!/bin/bash
MB=$((1024 * 1024))
size=$(qemu-img info -f raw --output json "$1" | gawk 'match($0, /"virtual-size": ([0-9]+)/, val)
{print val[1]}')
rounded_size=$((($size/$MB + 1) * $MB))
if [ $($size % $MB) -eq 0 ]
then
  echo "Your image is already aligned. You do not need to resize."
  exit 1
fi
echo "rounded size = $rounded_size"
export rounded_size
```

3. 运行脚本。本例使用名称 **align.sh**。

```
$ sh align.sh <image-xxx>.raw
```

- 如需显示 *"Your image is already aligned. You do not need to resize."*，执行以下步骤。
 - 如果显示了一个值，代表您的镜像没有被对齐。
4. 使用以下命令来将文件转换为固定的 **VHD** 格式：
示例使用 **qemu-img** 版本 2.12.0。

```
$ qemu-img convert -f raw -o subformat=fixed,force_size -O vpc <image-xxx>.raw
<image.xxx>.vhd
```

转换后，**VHD** 文件就可以上传到 Azure。

- 如果 **raw** 镜像不一致，请完成以下步骤使其保持一致。

- 使用运行验证脚本时显示的舍入值来调整 **raw** 文件的大小。

```
$ qemu-img resize -f raw <image-xxx>.raw <rounded-value>
```

- 将 **raw** 镜像文件转换为 **VHD** 格式。

示例使用 **qemu-img** 版本 2.12.0。

```
$ qemu-img convert -f raw -o subformat=fixed,force_size -O vpc <image-xxx>.raw
<image.xxx>.vhd
```

转换后，**VHD** 文件就可以上传到 Azure。

4.9. 上传并创建 AZURE 镜像

完成以下步骤，将 **VHD** 文件上传到容器，并创建 Azure 自定义镜像。



注意

系统重启后，导出的存储连接字符串不会保留。如果以下步骤中的任何命令失败，请再次导出连接字符串。

步骤

- 将 **VHD** 文件上传到存储容器。它可能需要几分钟时间。要获取存储容器的列表，请输入 **az storage container list** 命令。

```
$ az storage blob upload \
  --account-name <storage-account-name> --container-name <container-name> \
  --type page --file <path-to-vhd> --name <image-name>.vhd
```

Example:

```
[clouduser@localhost]$ az storage blob upload \
  --account-name azrhelcllistact --container-name azrhelcllistcont \
  --type page --file rhel-image-{ProductNumber}.vhd --name rhel-image-{ProductNumber}.vhd

Percent complete: %100.0
```

- 获取上传的 **VHD** 文件的 URL，以便在下面的步骤中使用。

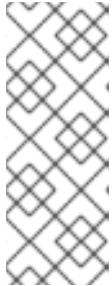
```
$ az storage blob url -c <container-name> -n <image-name>.vhd
```

Example:

```
$ az storage blob url -c azrhelcllistcont -n rhel-image-9.vhd
"https://azrhelcllistact.blob.core.windows.net/azrhelcllistcont/rhel-image-9.vhd"
```

3. 创建 Azure 自定义镜像。

```
$ az image create -n <image-name> -g <resource-group> -l <azure-region> --source <URL>
--os-type linux
```



注意

虚拟机的默认 hypervisor 系列为 V1。您可以通过包含 **--hyper-v-generation V2** 选项来（可选）指定 V2 管理程序生成。第二代虚拟机使用基于 UEFI 的引导架构。有关第 2 代虚拟机的信息，请参阅 [对 Azure 上 2 代虚拟机的支持](#)。

命令可能会返回错误 "Only blobs formatted as VHDs can be imported." 此错误可能意味着在转换为 **VHD** 之前，镜像与最接近的 1 MB 边界不一致。

Example:

```
$ az image create -n rhel9 -g azrhelclirgrp2 -l southcentralus --source
https://azrhelcllistact.blob.core.windows.net/azrhelcllistcont/rhel-image-9.vhd --os-type linux
```

4.10. 安装 RED HAT HA 软件包和代理

在所有节点上完成以下步骤。

流程

1. 启动一个 SSH 终端会话，并使用管理员名称和公共 IP 地址连接到虚拟机。

```
$ ssh administrator@PublicIP
```

要获取 Azure 虚拟机的公共 IP 地址，请在 Azure 门户网站中打开虚拟机属性，或者输入以下 Azure CLI 命令。

```
$ az vm list -g <resource-group> -d --output table
```

例如：

```
[clouduser@localhost ~] $ az vm list -g azrhelclirgrp -d --output table
Name ResourceGroup PowerState PublicIps Location
-----
node01 azrhelclirgrp VM running 192.98.152.251 southcentralus
```

2. 在红帽注册虚拟机。

```
$ sudo -i
# subscription-manager register --auto-attach
```

**注意**

如果 **--auto-attach** 命令失败，请手动将虚拟机注册到您的订阅。

- 禁用所有软件仓库。

```
# subscription-manager repos --disable=*
```

- 启用 RHEL 9 服务器 HA 软件仓库。

```
# subscription-manager repos --enable=rhel-9-for-x86_64-highavailability-rpms
```

- 更新所有软件包。

```
# dnf update -y
```

- 安装红帽高可用性附加组件软件包，以及来自高可用性频道的 Azure 隔离代理。

```
# dnf install pcs pacemaker fence-agents-azure-arm
```

- 用户 **hacluster** 是在上一步中的 pcs 和 pacemaker 安装过程中创建的。在所有群集节点上为 **hacluster** 创建密码。所有节点都使用相同的密码。

```
# passwd hacluster
```

- 如果安装了 **firewalld.service**，请在 RHEL Firewall 中添加 **high availability** 服务。

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --reload
```

- 启动 **pcs** 服务，并使其在引导时启动。

```
# systemctl start pcsd.service
# systemctl enable pcsd.service
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/pcsd.service to
/usr/lib/systemd/system/pcsd.service.
```

验证

- 确保 **pcs** 服务正在运行。

```
# systemctl status pcsd.service
pcsd.service - PCS GUI and remote configuration interface
Loaded: loaded (/usr/lib/systemd/system/pcsd.service; enabled; vendor preset: disabled)
Active: active (running) since Fri 2018-02-23 11:00:58 EST; 1min 23s ago
Docs: man:pcsd(8)
      man:pcs(8)
Main PID: 46235 (pcsd)
CGroup: /system.slice/pcsd.service
        └─46235 /usr/bin/ruby /usr/lib/pcsd/pcsd > /dev/null &
```


4.11. 创建集群

完成以下步骤以创建节点集群。

流程

1. 在其中一个节点上，输入以下命令来验证 pcs 用户 **hacluster**。在该命令中，指定集群中的每个节点的名称。

```
# pcs host auth <hostname1> <hostname2> <hostname3>
```

Example:

```
[root@node01 clouduser]# pcs host auth node01 node02 node03
Username: hacluster
Password:
node01: Authorized
node02: Authorized
node03: Authorized
```

2. 创建集群。

```
# pcs cluster setup <cluster_name> <hostname1> <hostname2> <hostname3>
```

Example:

```
[root@node01 clouduser]# pcs cluster setup new_cluster node01 node02 node03

[...]

Synchronizing pcsd certificates on nodes node01, node02, node03...
node02: Success
node03: Success
node01: Success
Restarting pcsd on the nodes in order to reload the certificates...
node02: Success
node03: Success
node01: Success
```

验证

1. 启用集群。

```
[root@node01 clouduser]# pcs cluster enable --all
node02: Cluster Enabled
node03: Cluster Enabled
node01: Cluster Enabled
```

2. 启动集群。

```
[root@node01 clouduser]# pcs cluster start --all
node02: Starting Cluster...
node03: Starting Cluster...
```

node01: Starting Cluster...

4.12. 隔离（FENCING）概述

如果与集群中某个节点通信失败，那么集群中的其他节点必须能够限制或释放对故障集群节点可访问的资源的访问。这无法通过通过联系集群节点本身来实现，因为集群节点可能没有响应。反之，必须提供一个外部的方法来实现。这个方法为称为隔离（fencing）。

不响应的节点可能仍然在访问数据。确定您的数据是安全的唯一方法是使用 STONITH 隔离节点。STONITH 是“Shoot The Other Node In The Head”的缩写，它保护您的数据不受有问题的节点或并发访问的影响。使用 STONITH 可以确保，在允许从另一个节点访问数据前确定节点真正离线。

其它资源

- [在 Red Hat High Availability Cluster 中进行隔离](#)

4.13. 创建隔离设备

完成以下步骤来配置隔离。在集群中的任何节点中完成这些命令

先决条件

您需要将集群属性 **stonith-enabled** 设为 **true**。

步骤

1. 识别每个 RHEL 虚拟机的 Azure 节点名称。您可以使用 Azure 节点名称来配置隔离设备。

```
# fence_azure_arm \
-l <AD-Application-ID> -p <AD-Password> \
--resourceGroup <MyResourceGroup> --tenantId <Tenant-ID> \
--subscriptionId <Subscription-ID> -o list
```

例如：

```
[root@node01 clouduser]# fence_azure_arm \
-l e04a6a49-9f00-xxxx-xxxx-a8bdda4af447 -p
z/a05AwCN0IzAjVwXXXXXXXXXEWIoeVp0xg7QT//JE=
--resourceGroup azrhelclirgrp --tenantId 77ecef66-cff0-XXXX-XXXX-757XXXX9485
--subscriptionId XXXXXXXX-38b4-4527-XXXX-012d49dfc02c -o list
```

```
node01,
node02,
node03,
```

2. 查看 Azure ARM STONITH 代理的选项。

```
# pcs stonith describe fence_azure_arm
```

例如：

```
# pcs stonith describe fence_apc
Stonith options:
```

```
password: Authentication key
password_script: Script to run to retrieve password
```



警告

对于提供方法选项的隔离代理，请不要指定循环值，因为它不被支持，并可能导致数据崩溃。

有些隔离设备只能隔离一个节点，其他设备则可能隔离多个节点。您创建隔离设备时指定的参数取决于您的隔离设备的支持和要求。

您可以在创建隔离设备时使用 `pcmk_host_list` 参数，以指定由该隔离设备控制的所有机器。

在创建隔离设备时，您可以使用 `pcmk_host_map` 参数将主机名映射到包含隔离设备的规范。

3. 创建隔离设备。

```
# pcs stonith create clusterfence fence_azure_arm
```

验证

1. 测试其他其中一个节点的隔离代理。

```
# pcs stonith fence azurenodename
```

例如：

```
[root@node01 clouduser]# pcs status
Cluster name: newcluster
Stack: corosync
Current DC: node01 (version 1.1.18-11.e17-2b07d5c5a9) - partition with quorum
Last updated: Fri Feb 23 11:44:35 2018
Last change: Fri Feb 23 11:21:01 2018 by root via cibadmin on node01

3 nodes configured
1 resource configured

Online: [ node01 node03 ]
OFFLINE: [ node02 ]

Full list of resources:

  clusterfence (stonith:fence_azure_arm): Started node01

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

2. 启动上一步中隔离的节点。

```
# pcs cluster start <hostname>
```

3. 检查状态以验证节点已启动。

```
# pcs status
```

例如：

```
[root@node01 clouduser]# pcs status
Cluster name: newcluster
Stack: corosync
Current DC: node01 (version 1.1.18-11.el7-2b07d5c5a9) - partition with quorum
Last updated: Fri Feb 23 11:34:59 2018
Last change: Fri Feb 23 11:21:01 2018 by root via cibadmin on node01

3 nodes configured
1 resource configured

Online: [ node01 node02 node03 ]

Full list of resources:

clusterfence (stonith:fence_azure_arm): Started node01

Daemon Status:
corosync: active/disabled
pacemaker: active/disabled
pcsd: active/enabled
```

其它资源

- [在 Red Hat High Availability Cluster 中进行隔离](#)
- [隔离设备的常规属性](#)

4.14. 创建 AZURE 内部负载均衡器

Azure 内部负载均衡器会删除对健康探测请求没有做出响应的集群节点。

执行以下步骤来创建 Azure 内部负载均衡器。每个步骤都引用特定的 Microsoft 流程，并包含用于自定义 HA 负载均衡器的设置。

先决条件

[Azure 控制面板](#)

流程

1. [创建基本负载均衡器](#)。为 IP 地址分配类型选择 **Internal load balancer**、**Basic SKU** 和 **Dynamic**。

2. **创建后端地址池。**将后端池与在 HA 中创建 Azure 资源时创建的可用性集关联。不要设置任何目标网络 IP 配置。
3. **创建健康探测。**对于健康探测，选择 **TCP** 并输入端口 **61000**。您可以使用不会影响到另一个服务的 TCP 端口号。对于某些 HA 产品应用（如 SAP HANA 和 SQL Server），您可能需要与 Microsoft 合作以确定要使用的正确端口。
4. **创建负载均衡规则。**要创建负载均衡规则，默认值会预先填充。确保将 **Floating IP (direct server return)** 设置为 **Enabled**。

4.15. 配置负载均衡资源代理

创建健康探测后，您必须配置 **负载均衡** 资源代理。此资源代理运行一个服务，它回答来自 Azure 负载均衡器的健康探测请求，并删除不回答请求的集群节点。

步骤

1. 在所有节点上安装 **nmap-ncat** 资源代理。

```
# dnf install nmap-ncat resource-agents-cloud
```

在单个节点上执行以下步骤。

2. 创建 **pcs** 资源和组。将负载均衡器 FrontendIP 用于 IPAddr2 地址。

```
# pcs resource create resource-name IPAddr2 ip="10.0.0.7" --group cluster-resources-group
```

3. 配置 **负载均衡** 资源代理。

```
# pcs resource create resource-loadbalancer-name azure-lb port=port-number --group cluster-resources-group
```

验证

- 运行 **pcs status** 来查看结果。

```
[root@node01 clouduser]# pcs status
```

输出示例：

```
Cluster name: clusterfence01
Stack: corosync
Current DC: node02 (version 1.1.16-12.el7_4.7-94ff4df) - partition with quorum
Last updated: Tue Jan 30 12:42:35 2018
Last change: Tue Jan 30 12:26:42 2018 by root via cibadmin on node01
```

```
3 nodes configured
3 resources configured
```

```
Online: [ node01 node02 node03 ]
```

```
Full list of resources:
```

```
clusterfence (stonith:fence_azure_arm): Started node01
```

```
Resource Group: g_azure
vip_azure (ocf::heartbeat:IPaddr2): Started node02
lb_azure (ocf::heartbeat:azure-lb): Started node02

Daemon Status:
corosync: active/disabled
pacemaker: active/disabled
pcsd: active/enabled
```

4.16. 配置共享块存储

要使用 Microsoft Azure Shared Disks 为 Red Hat High Availability 集群配置共享块存储，请使用以下流程。请注意，这个流程是可选的，以下步骤假设三个具有 1TB 共享磁盘的 Azure 虚拟机（三节点集群）。



注意

这是配置块存储的独立示例步骤。该流程假设您还没有创建集群。

先决条件

- 您必须已在主机系统中安装了 Azure CLI，并创建了 SSH 密钥。
- 您必须在 Azure 中创建了集群环境，其中包括创建以下资源。[Microsoft Azure 文档链接](#)。
 - [资源组](#)
 - [虚拟网络](#)
 - [网络安全组](#)
 - [网络安全组规则](#)
 - [子网](#)
 - [负载均衡器（可选）](#)
 - [存储帐户](#)
 - [代理放置组](#)
 - [可用性集](#)

流程

1. 使用 Azure 命令 **az disk create** 创建一个共享块卷。

```
$ az disk create -g <resource_group> -n <shared_block_volume_name> --size-gb
<disk_size> --max-shares <number_vms> -l <location>
```

例如，以下命令在 Azure Availability Zone **westcentralus** 的资源组 **sharedblock** 中创建名为 **shared-block-volume.vhd** 的共享块卷。

```
$ az disk create -g sharedblock-rg -n shared-block-volume.vhd --size-gb 1024 --max-shares
3 -l westcentralus
```

```

{
  "creationData": {
    "createOption": "Empty",
    "galleryImageReference": null,
    "imageReference": null,
    "sourceResourceId": null,
    "sourceUniqueId": null,
    "sourceUri": null,
    "storageAccountId": null,
    "uploadSizeBytes": null
  },
  "diskAccessId": null,
  "diskIopsReadOnly": null,
  "diskIopsReadWrite": 5000,
  "diskMbpsReadOnly": null,
  "diskMbpsReadWrite": 200,
  "diskSizeBytes": 1099511627776,
  "diskSizeGb": 1024,
  "diskState": "Unattached",
  "encryption": {
    "diskEncryptionSetId": null,
    "type": "EncryptionAtRestWithPlatformKey"
  },
  "encryptionSettingsCollection": null,
  "hyperVgeneration": "V1",
  "id": "/subscriptions/12345678910-12345678910/resourceGroups/sharedblock-rg/providers/Microsoft.Compute/disks/shared-block-volume.vhd",
  "location": "westcentralus",
  "managedBy": null,
  "managedByExtended": null,
  "maxShares": 3,
  "name": "shared-block-volume.vhd",
  "networkAccessPolicy": "AllowAll",
  "osType": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "sharedblock-rg",
  "shareInfo": null,
  "sku": {
    "name": "Premium_LRS",
    "tier": "Premium"
  },
  "tags": {},
  "timeCreated": "2020-08-27T15:36:56.263382+00:00",
  "type": "Microsoft.Compute/disks",
  "uniqueId": "cd8b0a25-6fbe-4779-9312-8d9cbb89b6f2",
  "zones": null
}

```

2. 验证您是否已使用 Azure 命令 **az disk show** 创建了共享块卷。

```
$ az disk show -g <resource_group> -n <shared_block_volume_name>
```

例如，以下命令显示资源组 **sharedblock-rg** 中共享块卷 **shared-block-volume.vhd** 的详细信息。

```

$ az disk show -g sharedblock-rg -n shared-block-volume.vhd

{
  "creationData": {
    "createOption": "Empty",
    "galleryImageReference": null,
    "imageReference": null,
    "sourceResourceId": null,
    "sourceUniqueId": null,
    "sourceUri": null,
    "storageAccountId": null,
    "uploadSizeBytes": null
  },
  "diskAccessId": null,
  "diskIopsReadOnly": null,
  "diskIopsReadWrite": 5000,
  "diskMbpsReadOnly": null,
  "diskMbpsReadWrite": 200,
  "diskSizeBytes": 1099511627776,
  "diskSizeGb": 1024,
  "diskState": "Unattached",
  "encryption": {
    "diskEncryptionSetId": null,
    "type": "EncryptionAtRestWithPlatformKey"
  },
  "encryptionSettingsCollection": null,
  "hyperVgeneration": "V1",
  "id": "/subscriptions/12345678910-12345678910/resourceGroups/sharedblock-rg/providers/Microsoft.Compute/disks/shared-block-volume.vhd",
  "location": "westcentralus",
  "managedBy": null,
  "managedByExtended": null,
  "maxShares": 3,
  "name": "shared-block-volume.vhd",
  "networkAccessPolicy": "AllowAll",
  "osType": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "sharedblock-rg",
  "shareInfo": null,
  "sku": {
    "name": "Premium_LRS",
    "tier": "Premium"
  },
  "tags": {},
  "timeCreated": "2020-08-27T15:36:56.263382+00:00",
  "type": "Microsoft.Compute/disks",
  "uniqueId": "cd8b0a25-6fbe-4779-9312-8d9cbb89b6f2",
  "zones": null
}

```

3. 使用 Azure 命令 **az network nic create** 创建三个网络接口。对每个网络接口使用不同的 **<nic_name>** 来运行以下命令三次。

```

$ az network nic create \
  -g <resource_group> -n <nic_name> --subnet <subnet_name> \

```



```
--vnet-name <virtual_network> --location <location> \
--network-security-group <network_security_group> --private-ip-address-version IPv4
```

例如，以下命令创建一个名为 **shareblock-nodea-vm-nic-protected** 的网络接口：

```
$ az network nic create \
-g shareblock-rg -n shareblock-nodea-vm-nic-protected --subnet shareblock-subnet-
protected \
--vnet-name shareblock-vn --location westcentralus \
--network-security-group shareblock-nsg --private-ip-address-version IPv4
```

4. 使用 Azure 命令 **az vm create** 创建三个虚拟机并附加共享块卷。每个虚拟机的选项值都相同，但每个虚拟机都有自己的 **<vm_name>**、**<new_vm_disk_name>** 和 **<nic_name>**。

```
$ az vm create \
-n <vm_name> -g <resource_group> --attach-data-disks <shared_block_volume_name> \
--data-disk-caching None --os-disk-caching ReadWrite --os-disk-name <new-vm-disk-
name> \
--os-disk-size-gb <disk_size> --location <location> --size <virtual_machine_size> \
--image <image_name> --admin-username <vm_username> --authentication-type ssh \
--ssh-key-values <ssh_key> --nics <nic_name> --availability-set <availability_set> --ppg
<proximity_placement_group>
```

例如，以下命令创建名为 **shareblock-nodea-vm** 的虚拟机：

```
$ az vm create \
-n shareblock-nodea-vm -g shareblock-rg --attach-data-disks shared-block-volume.vhd \
--data-disk-caching None --os-disk-caching ReadWrite --os-disk-name shareblock-nodea-
vm.vhd \
--os-disk-size-gb 64 --location westcentralus --size Standard_D2s_v3 \
--image /subscriptions/12345678910-12345678910/resourceGroups/sample-
azureimagesgroupwestcentralus/providers/Microsoft.Compute/images/sample-azure-rhel-
9.3.0-20200713.n.0.x86_64 --admin-username shareblock-user --authentication-type ssh \
--ssh-key-values @shareblock-key.pub --nics shareblock-nodea-vm-nic-protected --
availability-set shareblock-as --ppg shareblock-ppg

{
  "fqdns": "",
  "id": "/subscriptions/12345678910-12345678910/resourceGroups/shareblock-
rg/providers/Microsoft.Compute/virtualMachines/shareblock-nodea-vm",
  "location": "westcentralus",
  "macAddress": "00-22-48-5D-EE-FB",
  "powerState": "VM running",
  "privateIpAddress": "198.51.100.3",
  "publicIpAddress": "",
  "resourceGroup": "shareblock-rg",
  "zones": ""
}
```

验证

1. 对于集群中的每个虚拟机，使用 **ssh** 命令和虚拟机的 IP 地址验证块设备是否可用。

```
# ssh <ip_address> "hostname ; lsblk -d | grep ' 1T '"
```

例如，以下命令列出了虚拟机 IP **198.51.100.3** 的详细信息，其中包括主机名和块设备。

```
# ssh 198.51.100.3 "hostname ; lsblk -d | grep ' 1T '"  
  
nodea  
sdb 8:16 0 1T 0 disk
```

2. 使用 **ssh** 命令来验证集群中的每个虚拟机是否使用相同的共享磁盘。

```
# ssh <ip_address> "hostname ; lsblk -d | grep ' 1T ' | awk '{print \$1}' | xargs -i udevadm info  
--query=all --name=/dev/{} | grep '^E: ID_SERIAL='"
```

例如，以下命令列出了 IP 地址为 **198.51.100.3** 的实例的详细信息，其中包括主机名和共享磁盘卷 ID。

```
# ssh 198.51.100.3 "hostname ; lsblk -d | grep ' 1T ' | awk '{print \$1}' | xargs -i udevadm info -  
-query=all --name=/dev/{} | grep '^E: ID_SERIAL='"  
  
nodea  
E: ID_SERIAL=3600224808dd8eb102f6ffc5822c41d89
```

在确认将共享磁盘附加到每个虚拟机后，您可以为集群配置弹性存储。

其他资源

- [在集群中配置 GFS2 文件系统](#)
- [配置 GFS2 文件系统](#)

4.17. 其他资源

- [RHEL 高可用性集群的支持政策 - 作为集群成员的 Microsoft Azure 虚拟机](#)
- [配置和管理高可用性集群](#)