



# Red Hat OpenStack Platform 17.1

## 备份和恢复 undercloud 和 control plane 节点

创建并恢复 undercloud 和 overcloud control plane 节点的备份



# Red Hat OpenStack Platform 17.1 备份和恢复 undercloud 和 control plane 节点

---

创建并恢复 undercloud 和 overcloud control plane 节点的备份

## 法律通告

Copyright © 2023 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

本指南介绍了如何备份和恢复 Red Hat Openstack Platform (RHOSP)。本指南还介绍了如何对备份和恢复问题进行故障排除。您必须在更新或升级 RHOSP 时创建备份。要备份 RHOSP，您可以使用以下工具之一：快照和 Revert 工具。在创建快照时，保留 RHOSP 集群的原始磁盘状态。根据更新或升级的结果，您可以删除或恢复快照。Relax-and-Recover (ReaR)工具。当您使用 ReaR 工具备份环境时，您可以创建 undercloud 节点和 control plane 节点的备份镜像。如果升级或更新过程中出现错误，您可以使用这些备份将这些 undercloud 节点和 control plane 节点恢复到之前的状态。另外，您可以使用 ReaR 工具定期创建环境备份，以便在出现问题时最小化停机时间。

---

## 目录

使开源包含更多 .....	3
对红帽文档提供反馈 .....	4
第 1 章 使用 SNAPSHOT 和 REVERT 工具备份 RED HAT OPENSTACK PLATFORM 集群 .....	5
1.1. 创建 CONTROLLER 和 COMPUTE 节点的快照 .....	5
第 2 章 使用 RELAX-AND-RECOVER 工具备份 UNDERCLOUD 和 CONTROL PLANE 节点 .....	7
2.1. 使用 RELAX-AND-RECOVER 工具备份 UNDERCLOUD 节点 .....	7
2.2. 使用 RELAX-AND-RECOVER 工具备份 CONTROL PLANE 节点 .....	12
2.3. 使用 RELAX-AND-RECOVER 工具恢复 UNDERCLOUD 节点和 CONTROL PLANE 节点 .....	16



---

## 使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [CTO Chris Wright 的信息](#)。

## 对红帽文档提供反馈

我们感谢您对文档提供反馈信息。与我们分享您的成功秘诀。

### 使用直接文档反馈(DDF)功能

使用 **添加反馈** DDF 功能，用于特定句子、段落或代码块上的直接注释。

1. 以 *Multi-page HTML* 格式查看文档。
2. 请确定您看到文档右上角的 **反馈** 按钮。
3. 用鼠标指针高亮显示您想评论的文本部分。
4. 点 **添加反馈**。
5. 在**添加反馈**项中输入您的意见。
6. 可选：添加您的电子邮件地址，以便文档团队可以联系您以讨论您的问题。
7. 点 **Submit**。

# 第1章 使用 SNAPSHOT 和 REVERT 工具备份 RED HAT OPENSTACK PLATFORM 集群

在执行升级或从 RHOSP 17.1 或更高版本更新前，快照会保留 Red Hat OpenStack Platform (RHOSP) 集群的原始磁盘状态。然后，您可以删除或恢复快照，具体取决于结果。例如，如果升级成功完成，且您不再需要快照，请将它们从节点中删除。如果升级失败，您可以恢复快照，评估任何错误，然后再次启动升级过程。恢复会使所有节点的磁盘与执行快照时完全相同。

RHOSP Snapshot 和 Revert 工具基于逻辑卷管理器(LVM)快照功能，它只旨在恢复升级或更新失败。



## 重要

快照存储在与磁盘上存储的数据相同的硬盘上。因此，快照和 Revert 工具不会阻止硬件故障、数据中心故障或无法访问的节点时出现数据丢失。

您可以对 Controller 节点和 Compute 节点进行快照。不支持为 undercloud 拍摄快照。

## 1.1. 创建 CONTROLLER 和 COMPUTE 节点的快照

在执行升级或更新前，为 Controller 和 Compute 节点创建快照。然后，您可以删除或恢复快照，具体取决于这些操作的结果。



## 注意

您只能为 Controller 和 Compute 节点创建一个快照。要创建另一个快照，您必须删除或恢复之前的快照。

### 先决条件

- 已在节点上启用了 LVM。
- RHOSP 安装定义的以下默认 LVM 逻辑卷集合如下：
  - /dev/vg/lv\_audit
  - /dev/vg/lv\_home
  - /dev/vg/lv\_log
  - /dev/vg/lv\_root
  - /dev/vg/lv\_srv
  - /dev/vg/lv\_var

您可以运行 `lvs`、`lvscan` 或 `lvdisplay` 命令，以确认您的环境在更改节点磁盘前是否包含这些先决条件。



## 注意

这些先决条件包含在 17.1 集群的默认安装中。但是，如果您从以前的 RHOSP 版本升级到 RHOSP 17.1，您的 control plane 不包括这些先决条件，因为它们需要重新格式化磁盘。

### 流程

1. 以 stack 用户身份登录 undercloud。
2. 查找 stackrc undercloud 凭证文件：

```
[stack@undercloud ~]$ source stackrc
(undercloud) [stack@undercloud ~]$
```

3. 如果您之前还没有这样做，请从安装过程中保存它的位置提取静态 Ansible 清单文件：

```
(undercloud) [stack@undercloud ~]$ cp ~/overcloud-deploy/<stack> /tripleo-ansible-
inventory.yaml ~/tripleo-inventory.yaml
```

- 将 <stack> 替换为您的堆栈的名称。默认情况下，堆栈的名称是 **overcloud**。

4. 创建快照：

```
(undercloud) [stack@undercloud ~]$ openstack overcloud backup snapshot --inventory
~/tripleo-inventory.yaml
```

5. 如果升级或更新成功，请删除快照：

```
(undercloud) [stack@undercloud ~]$ openstack overcloud backup snapshot --remove --
inventory ~/tripleo-inventory.yaml
```



### 重要

删除快照是一个关键操作。如果您不打算恢复节点，例如在升级成功完成后删除快照。如果您在节点上保留快照太长，则它们会降低磁盘 I/O 性能。

6. 如果升级或更新失败，请恢复快照：

```
(undercloud) [stack@undercloud ~]$ openstack overcloud backup snapshot --revert --
inventory ~/tripleo-inventory.yaml
```

- a. 重启每个节点，以便更改将应用到文件系统。恢复选项自动删除快照。

## 第 2 章 使用 RELAX-AND-RECOVER 工具备份 UNDERCLOUD 和 CONTROL PLANE 节点

升级或更新 Red Hat Openstack Platform (RHOSP)时，您必须备份 undercloud 节点和 control plane 节点。您可以使用 Relax-and-Recover (ReaR)工具备份 undercloud 节点和 control plane 节点。要使用 ReaR 工具备份和恢复 undercloud 和 control plane 节点，您必须完成以下步骤：

1. 备份 undercloud 节点
2. 备份 control plane 节点
3. 恢复 undercloud 和 control plane 节点

### 2.1. 使用 RELAX-AND-RECOVER 工具备份 UNDERCLOUD 节点

要备份 undercloud 节点，请配置备份节点，在 undercloud 节点上安装 Relax-and-Recover 工具，然后创建备份镜像。您可以将备份创建作为常规环境维护的一部分。

另外，您必须在执行更新或升级前备份 undercloud 节点。如果在更新或升级过程中出现错误，您可以使用备份将 undercloud 节点恢复到之前的状态。

#### 2.1.1. 支持的备份格式和协议

undercloud 和备份和恢复过程使用开源工具 Relax-and-Recover (ReaR)创建和恢复可引导备份镜像。ReaR 使用 Bash 编写，支持多个镜像格式和多个传输协议。

以下列表显示了 Red Hat OpenStack Platform 在使用 ReaR 备份和恢复 undercloud 和 control plane 时支持的备份格式和协议。

##### 可引导介质格式

- ISO

##### 文件传输协议

- SFTP
- NFS

#### 2.1.2. 配置备份存储位置

在创建 control plane 节点的备份前，在 **bar-vars.yaml** 环境文件中配置备份存储位置。此文件存储要传递给备份执行的键值参数。

##### 流程

1. 以 **stack** 用户的身份登录 undercloud。
2. Source **stackrc** 文件：

```
$ source ~/stackrc
```

### 3. 创建 **bar-vars.yaml** 文件：

```
touch /home/stack/bar-vars.yaml
```

### 4. 在 **bar-vars.yaml** 文件中，配置备份存储位置：

- a. 如果使用 NFS 服务器，请添加以下参数并设置 NFS 服务器的 IP 地址和备份存储文件夹：

```
tripleo_backup_and_restore_server: <ip_address>
tripleo_backup_and_restore_shared_storage_folder: <backup_dir>
```

- 将 <ip\_address> 和 <backup\_dir> 替换为应用到您的环境的值。默认情况下，**tripleo\_backup\_and\_restore\_server** 参数值为 **192.168.24.1**。

- b. 如果您使用 SFTP 服务器，请添加 **tripleo\_backup\_and\_restore\_output\_url** 参数并设置 SFTP 服务器的 URL 和凭证的值：

```
tripleo_backup_and_restore_output_url: sftp://<user>:<password>@<backup_node>/
tripleo_backup_and_restore_backup_url: iso:///backup/
```

将 **<user>**、**<password>** 和 **<backup\_node>** 替换为备份节点 URL 和凭证。

### 2.1.3. 可选：配置备份加密

您可以将备份加密为额外的安全措施，以保护敏感数据。

#### 流程

- 在 **bar-vars.yaml** 文件中，添加以下参数：

```
tripleo_backup_and_restore_crypt_backup_enabled: true
tripleo_backup_and_restore_crypt_backup_password: <password>
```

将 **<password>** 替换为您要用来加密备份的密码。

### 2.1.4. 在备份节点上安装和配置 NFS 服务器

您可以安装和配置新的 NFS 服务器来存储备份文件。要在备份节点上安装和配置 NFS 服务器，请创建一个清单文件，创建 SSH 密钥，并使用 NFS 服务器选项运行 **openstack undercloud backup** 命令。



#### 重要

- 如果您之前安装和配置了 NFS 或 SFTP 服务器，则不需要完成此流程。当您在要备份的节点上设置 ReaR 时，输入服务器信息。
- 默认情况下，NFS 服务器的 Relax 和 Recover (ReaR) IP 地址参数是 **192.168.24.1**。您必须添加参数 **tripleo\_backup\_and\_restore\_server** 来设置与您的环境匹配的 IP 地址值。

#### 流程

1. 在 undercloud 节点上，提供 undercloud 凭证：

```
[stack@undercloud-0 ~]$ source stackrc
(undercloud) [stack@undercloud ~]$
```

- 在 undercloud 节点上，为备份节点创建一个清单文件：

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF'> ~/nfs-inventory.yaml
[BackupNode]
<backup_node> ansible_host=<ip_address> ansible_user=<user>
EOF
```

将 `<backup_node>` , `<ip_address>` , 和 `<user>` 替换为应用到您的环境的值。

- 将公共 SSH 密钥从 undercloud 节点复制到备份节点。

```
(undercloud) [stack@undercloud ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub <backup_node>
```

将 `<backup_node>` 替换为备份节点的路径和名称。

- 在备份节点上配置 NFS 服务器：

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup --setup-nfs --extra-vars
/home/stack/bar-vars.yaml --inventory /home/stack/nfs-inventory.yaml
```

## 2.1.5. 在 undercloud 节点上安装 ReaR

在创建 undercloud 节点的备份前，在 undercloud 上安装和配置 Relax 和 Recover (ReaR)。

### 先决条件

- 您已在备份节点上安装并配置了 NFS 或 SFTP 服务器。有关创建新 NFS 服务器的详情请参考 [第 2.1.4 节“在备份节点上安装和配置 NFS 服务器”](#)。

### 流程

- 在 undercloud 节点上，提供 undercloud 凭证：

```
[stack@undercloud-0 ~]$ source stackrc
```

- 如果您之前还没有这样做，请从安装过程中保存它的位置提取静态 ansible 清单文件：

```
(undercloud) [stack@undercloud ~]$ cp ~/overcloud-deploy/<stack>/tripleo-ansible-
inventory.yaml ~/tripleo-inventory.yaml
```

- 将 `<stack>` 替换为您的堆栈的名称。默认情况下，堆栈的名称是 **overcloud**。

- 在 undercloud 节点上安装 ReaR：

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup --setup-rear --extra-vars
/home/stack/bar-vars.yaml --inventory /home/stack/tripleo-inventory.yaml
```

- 如果您的系统使用 UEFI 引导装载程序，请在 undercloud 节点上执行以下步骤：

- 安装以下工具：

```
$ sudo dnf install dosfstools efibootmgr
```

- b. 通过将 **USING\_UEFI\_BOOTLOADER** 参数值 **0** 替换为值 **1**，在 `/etc/rear/local.conf` 中的 ReaR 配置文件中启用 UEFI 备份。

## 2.1.6. 可选：创建 undercloud 节点的独立数据库备份

您可以在常规备份调度中包含独立 undercloud 数据库备份，以提供额外的数据安全性。undercloud 节点的完整备份包括 undercloud 节点的数据库备份。但是，如果完整 undercloud 恢复失败，您可能会丢失对完整 undercloud 备份的数据库部分的访问。在这种情况下，您可以从独立 undercloud 数据库备份中恢复数据库。

您可以与 ReaR 工具和 Snapshot 和 Revert 工具结合使用创建独立 undercloud 数据库备份。但是，建议您备份整个 undercloud。有关创建 undercloud 节点备份的更多信息，请参阅[创建 undercloud 节点的备份](#)。

### 流程

- 创建 undercloud 节点的数据库备份：

```
openstack undercloud backup --db-only
```

db 备份文件存储在 `/home/stack` 中，其名称为 `openstack-backup-mysql-<timestamp>.sql`。

### 其他资源

- [第 2.1.8 节 “创建 undercloud 节点的备份”](#)
- [第 2.3.4 节 “手动恢复 undercloud 节点数据库”](#)

## 2.1.7. 配置 Open vSwitch (OVS)接口以进行备份

如果您在环境中使用 Open vSwitch (OVS)网桥，则必须在创建 undercloud 或 control plane 节点的备份前手动配置 OVS 接口。恢复过程使用此信息来恢复网络接口。

### 流程

- 在 `/etc/rear/local.conf` 文件中，以以下格式添加 **NETWORKING\_PREPARATION\_COMMANDS** 参数：

```
NETWORKING_PREPARATION_COMMANDS=('<command_1>' '<command_2>' ...)
```

将 `<command_1>` 和 `<command_2>` 替换为配置网络接口名称或 IP 地址的命令。例如，您可以添加 `ip link add br-ctlplane type bridge` 命令来配置 control plane 网桥名称，或者添加 `ip link set eth0 up` 命令来设置接口名称。您可以根据网络配置在参数中添加更多命令。

## 2.1.8. 创建 undercloud 节点的备份

要创建 undercloud 节点的备份，请使用 `openstack undercloud backup` 命令。然后，当节点变得损坏或无法访问时，您可以使用备份将 undercloud 节点恢复到之前的状态。undercloud 节点的备份包括 undercloud 节点上运行的数据库的备份。



## 注意

建议您按照以下流程创建 undercloud 节点的备份。但是，如果您已完成 [创建 undercloud 节点的独立数据库备份](#)，您可以跳过这个过程。

### 先决条件

- 您已在备份节点上安装并配置了 NFS 或 SFTP 服务器。有关创建新 NFS 服务器的详情请参考 [第 2.1.4 节“在备份节点上安装和配置 NFS 服务器”](#)。
- 您已在 undercloud 节点上安装了 ReaR。更多信息请参阅 [第 2.1.5 节“在 undercloud 节点上安装 ReaR”](#)。
- 如果您的网络接口使用 OVS 网桥，您已配置 OVS 接口。更多信息请参阅 [第 2.1.7 节“配置 Open vSwitch \(OVS\) 接口以进行备份”](#)。

### 步骤

1. 以 **stack** 用户的身份登录 undercloud。

2. 检索 MySQL root 密码：

```
[stack@undercloud ~]$ PASSWORD=$(sudo /bin/hiera -c /etc/puppet/hiera.yaml
mysql::server::root_password)
```

3. 创建 undercloud 节点的数据库备份：

```
[stack@undercloud ~]$ sudo podman exec mysql bash -c "mysqldump -uroot -
p$PASSWORD --opt --all-databases" | sudo tee /root/undercloud-all-databases.sql
```

4. 在 undercloud 节点上，提供 undercloud 凭证：

```
[stack@undercloud-0 ~]$ source stackrc
```

5. 创建 undercloud 节点的备份：

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup
--inventory /home/stack/tripleo-inventory.yaml
```

## 2.1.9. 使用 cron 调度 undercloud 节点备份

您可以使用 Ansible **backup-and-restore** 角色使用 ReaR 调度 undercloud 节点的备份。您可以查看 `/var/log/rear-cron` 目录中的日志。

### 先决条件

- 您已在备份节点上安装并配置了 NFS 或 SFTP 服务器。有关创建新 NFS 服务器的详情请参考 [第 2.1.4 节“在备份节点上安装和配置 NFS 服务器”](#)。
- 您已在 undercloud 和 control plane 节点上安装 ReaR。更多信息请参阅 [第 2.2.3 节“在 control plane 节点上安装 ReaR”](#)。
- 在您的备份位置有足够的可用磁盘空间来存储备份。

## 流程

1. 要调度 control plane 节点的备份，请运行以下命令。默认调度为 Sundays 在午夜：

```
openstack undercloud backup --cron
```

2. 可选：根据您的部署自定义调度的备份：

- 要更改默认备份调度，请在 **tripleo\_backup\_and\_restore\_cron** 参数上传递不同的 cron 调度：

```
openstack undercloud backup --cron --extra-vars
'{"tripleo_backup_and_restore_cron": "0 0 * * 0"}
```

- 要定义在 cron 运行调度备份时添加到 backup 命令中的其他参数，请将 **tripleo\_backup\_and\_restore\_cron\_extra** 参数传递给 backup 命令，如下例所示：

```
openstack undercloud backup --cron --extra-vars
'{"tripleo_backup_and_restore_cron_extra": "--extra-vars bar-vars.yaml --inventory
/home/stack/tripleo-inventory.yaml"}'
```

- 要更改执行备份的默认用户，请将 **tripleo\_backup\_and\_restore\_cron\_user** 参数传递给 backup 命令，如下例所示：

```
openstack undercloud backup --cron --extra-vars
'{"tripleo_backup_and_restore_cron_user": "root"}
```

## 2.2. 使用 RELAX-AND-RECOVER 工具备份 CONTROL PLANE 节点

要备份 control plane 节点，请配置备份节点，在 control plane 节点上安装 Relax-and-Recover 工具，并创建备份镜像。您可以将备份创建作为常规环境维护的一部分。

另外，您必须在执行更新或升级前备份 control plane 节点。如果在更新或升级过程中出现错误，您可以使用备份将 control plane 节点恢复到之前的状态。

### 2.2.1. 支持的备份格式和协议

undercloud 和备份和恢复过程使用开源工具 Relax-and-Recover (ReaR) 创建和恢复可引导备份镜像。ReaR 使用 Bash 编写，支持多个镜像格式和多个传输协议。

以下列表显示了 Red Hat OpenStack Platform 在使用 ReaR 备份和恢复 undercloud 和 control plane 时支持的备份格式和协议。

#### 可引导介质格式

- ISO

#### 文件传输协议

- SFTP
- NFS

### 2.2.2. 在备份节点上安装和配置 NFS 服务器

您可以安装和配置新的 NFS 服务器来存储备份文件。要在备份节点上安装和配置 NFS 服务器，请创建一个清单文件，创建 SSH 密钥，并使用 NFS 服务器选项运行 **openstack undercloud backup** 命令。



#### 重要

- 如果您之前安装和配置了 NFS 或 SFTP 服务器，则不需要完成此流程。当您在要备份的节点上设置 ReaR 时，输入服务器信息。
- 默认情况下，NFS 服务器的 Relax 和 Recover (ReaR) IP 地址参数是 **192.168.24.1**。您必须添加参数 **tripleo\_backup\_and\_restore\_server** 来设置与您的环境匹配的 IP 地址值。

#### 流程

1. 在 undercloud 节点上，提供 undercloud 凭证：

```
[stack@undercloud-0 ~]$ source stackrc
(undercloud) [stack@undercloud ~]$
```

2. 在 undercloud 节点上，为备份节点创建一个清单文件：

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF'> ~/nfs-inventory.yaml
[BackupNode]
<backup_node> ansible_host=<ip_address> ansible_user=<user>
EOF
```

将 **<backup\_node>**，**<ip\_address>**，和 **<user>** 替换为应用到您的环境的值。

3. 将公共 SSH 密钥从 undercloud 节点复制到备份节点。

```
(undercloud) [stack@undercloud ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub <backup_node>
```

将 **<backup\_node>** 替换为备份节点的路径和名称。

4. 在备份节点上配置 NFS 服务器：

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup --setup-nfs --extra-vars
/home/stack/bar-vars.yaml --inventory /home/stack/nfs-inventory.yaml
```

### 2.2.3. 在 control plane 节点上安装 ReaR

在创建 control plane 节点的备份前，在每个 control plane 节点上安装和配置 Relax 和 Recover (ReaR)。



#### 重要

由于一个已知问题，即使 Controller 节点停机，overcloud 节点的 ReaR 备份也会继续。在运行 ReaR 备份前，请确保所有 Controller 节点都在运行。计划在以后的 Red Hat OpenStack Platform (RHOSP) 发行版本进行修复。有关更多信息，请参阅 [BZ#2077335](#) - 即使一个控制器无法访问，也会保留 overcloud ctlplane。

#### 先决条件

- 您已在备份节点上安装并配置了 NFS 或 SFTP 服务器。有关创建新 NFS 服务器的详情请参考第 2.2.2 节“在备份节点上安装和配置 NFS 服务器”。

## 流程

1. 在 undercloud 节点上，提供 undercloud 凭证：

```
[stack@undercloud-0 ~]$ source stackrc
```

2. 如果您之前还没有这样做，请从安装过程中保存它的位置提取静态 ansible 清单文件：

```
(undercloud) [stack@undercloud ~]$ cp ~/overcloud-deploy/<stack>/tripleo-ansible-inventory.yaml ~/tripleo-inventory.yaml
```

- 将 **<stack>** 替换为您的堆栈的名称。默认情况下，堆栈的名称是 **overcloud**。

3. 在 **bar-vars.yaml** 文件中，配置备份存储位置：

- 如果您安装并配置了自己的 NFS 服务器，请添加 **tripleo\_backup\_and\_restore\_server** 参数，并将值设为 NFS 服务器的 IP 地址：

```
tripleo_backup_and_restore_server: <ip_address>
tripleo_backup_and_restore_shared_storage_folder: <backup_dir>
```

- 将 **<ip\_address>** 和 **<backup\_dir>** 替换为应用到您的环境的值。默认情况下，**tripleo\_backup\_and\_restore\_server** 参数值为 **192.168.24.1** adtrust

- 如果您使用 SFTP 服务器，请添加 **tripleo\_backup\_and\_restore\_output\_url** 参数并设置 SFTP 服务器的 URL 和凭证的值：

```
tripleo_backup_and_restore_output_url: sftp://<user>:<password>@<backup_node>/
tripleo_backup_and_restore_backup_url: iso:///backup/
```

将 **<user>**、**<password>** 和 **<backup\_node>** 替换为备份节点 URL 和凭证。

4. 在 control plane 节点上安装 ReaR：

```
(undercloud) [stack@undercloud ~]$ openstack overcloud backup --setup-rear --extra-vars /home/stack/bar-vars.yaml --inventory /home/stack/tripleo-inventory.yaml
```

5. 如果您的系统使用 UEFI 引导装载程序，请在 control plane 节点上执行以下步骤：

- a. 安装以下工具：

```
$ sudo dnf install dosfstools efibootmgr
```

- b. 通过将 **USING\_UEFI\_BOOTLOADER** 参数值 **0** 替换为值 **1**，在 **/etc/rear/local.conf** 中的 ReaR 配置文件中启用 UEFI 备份。

### 2.2.4. 配置 Open vSwitch (OVS)接口以进行备份

如果您在环境中使用 Open vSwitch (OVS)网桥，则必须在创建 undercloud 或 control plane 节点的备份前手动配置 OVS 接口。恢复过程使用此信息来恢复网络接口。

## 流程

- 在 `/etc/rear/local.conf` 文件中，以以下格式添加 **NETWORKING\_PREPARATION\_COMMANDS** 参数：

```
NETWORKING_PREPARATION_COMMANDS=('<command_1>' '<command_2>' ...)
```

将 `<command_1>` 和 `<command_2>` 替换为配置网络接口名称或 IP 地址的命令。例如，您可以添加 `ip link add br-ctlplane type bridge` 命令来配置 control plane 网桥名称，或者添加 `ip link set eth0 up` 命令来设置接口名称。您可以根据网络配置在参数中添加更多命令。

### 2.2.5. 创建 control plane 节点的备份

要创建 control plane 节点的备份，请使用 **openstack overcloud backup** 命令。然后，如果节点被破坏或无法访问，您可以使用备份将 control plane 节点恢复到之前的状态。control plane 节点的备份包括在 control plane 节点上运行的数据库的备份。

#### 先决条件

- 您已在备份节点上安装并配置了 NFS 或 SFTP 服务器。有关创建新 NFS 服务器的详情请参考第 2.2.2 节“在备份节点上安装和配置 NFS 服务器”。
- 您已在 control plane 节点上安装 ReaR。更多信息请参阅第 2.2.3 节“在 control plane 节点上安装 ReaR”。
- 如果您的网络接口使用 OVS 网桥，您已配置 OVS 接口。更多信息请参阅第 2.2.4 节“配置 Open vSwitch (OVS)接口以进行备份”。

## 流程

- 在每个 control plane 节点上找到 **config-drive** 分区：

```
[stack@undercloud-0 ~]$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda 253:0 0 55G 0 disk
├─vda1 253:1 0 1M 0 part 1
├─vda2 253:2 0 100M 0 part /boot/efi
└─vda3 253:3 0 54.9G 0 part /
```

- 1** **config-drive** 分区是未挂载的 1M 分区。

- 在每个 control plane 节点上，以 **root** 用户身份备份每个节点的 **config-drive** 分区：

```
[root@controller-x ~]# dd if=<config_drive_partition> of=/mnt/config-drive
```

将 `<config_drive_partition>` 替换为在第 1 步中的 **config-drive** 分区的名称。

- 在 undercloud 节点上，提供 undercloud 凭证：

```
[stack@undercloud-0 ~]$ source stackrc
```

- 创建 control plane 节点的备份：

```
(undercloud) [stack@undercloud ~]$ openstack overcloud backup --inventory
/home/stack/tripleo-inventory.yaml
```

备份过程在每个 control plane 节点上按顺序运行，而不会破坏您的环境的服务。

## 2.2.6. 使用 cron 调度 control plane 节点备份

您可以使用 Ansible **backup-and-restore** 角色使用 ReaR 调度 control plane 节点的备份。您可以查看 `/var/log/rear-cron` 目录中的日志。

### 先决条件

- 您已在备份节点上安装并配置了 NFS 或 SFTP 服务器。有关创建新 NFS 服务器的详情请参考 [第 2.1.4 节 “在备份节点上安装和配置 NFS 服务器”](#)。
- 您已在 undercloud 和 control plane 节点上安装 ReaR。更多信息请参阅 [第 2.2.3 节 “在 control plane 节点上安装 ReaR”](#)。
- 在您的备份位置有足够的可用磁盘空间来存储备份。

### 流程

1. 要调度 control plane 节点的备份，请运行以下命令。默认调度为 Sundays 在午夜：

```
openstack overcloud backup --cron
```

2. 可选：根据您的部署自定义调度的备份：

- 要更改默认备份调度，请在 `tripleo_backup_and_restore_cron` 参数上传递不同的 cron 调度：

```
openstack overcloud backup --cron --extra-vars
'{"tripleo_backup_and_restore_cron": "0 0 * * 0"}
```

- 要定义在 cron 运行调度备份时添加到 backup 命令中的其他参数，请将 `tripleo_backup_and_restore_cron_extra` 参数传递给 backup 命令，如下例所示：

```
openstack overcloud backup --cron --extra-vars
'{"tripleo_backup_and_restore_cron_extra": "--extra-vars bar-vars.yaml --inventory
/home/stack/tripleo-inventory.yaml"}'
```

- 要更改执行备份的默认用户，请将 `tripleo_backup_and_restore_cron_user` 参数传递给 backup 命令，如下例所示：

```
openstack overcloud backup --cron --extra-vars
'{"tripleo_backup_and_restore_cron_user": "root"}
```

## 2.3. 使用 RELAX-AND-RECOVER 工具恢复 UNDERCLOUD 节点和 CONTROL PLANE 节点

如果您的 undercloud 或 control plane 节点被破坏，或者在更新或升级过程中发生错误，您可以将 undercloud 或 overcloud control plane 节点从备份恢复到之前的状态。如果恢复过程无法自动恢复具有 colocated Ceph 监视器的 Galera 集群或节点，您可以手动恢复这些组件。

### 2.3.1. 恢复 undercloud 节点

您可以使用 ReaR 创建的备份 ISO 镜像将 undercloud 节点恢复到之前的状态。您可以在备份节点中找到备份 ISO 镜像。将可引导 ISO 镜像刻录到 DVD，或通过 Integrated Lights-Out (iLO) 远程访问将其下载到 undercloud 节点。

#### 先决条件

- 您已创建了 undercloud 节点的备份。更多信息请参阅 [第 2.1.8 节“创建 undercloud 节点的备份”](#)。
- 您可以访问备份节点。
- 如果将 OVS 网桥用于网络接口，您可以访问您在 **NETWORKING\_PREPARATION\_COMMANDS** 参数中设置的网络配置信息。如需更多信息，请参阅 [第 2.1.7 节“配置 Open vSwitch \(OVS\) 接口以进行备份”](#)。
- 如果配置了备份加密，您必须在开始恢复过程前解密备份。在备份文件所在的系统中运行以下解密步骤：

```
$ dd if=backup.tar.gz | /usr/bin/openssl des3 -d -k "<encryption key>" | tar -C
<backup_location> -xzf - '*.conf'
```

- 将 **<encryption key>** 替换为您的加密密钥。
- 将 **<backup\_location>** 替换为您要保存 **backup.tar.gz** 文件的文件夹，如 **/ctl\_plane\_backups/undercloud-0/**。

#### 流程

1. 关闭 undercloud 节点。在继续操作前，请确保 undercloud 节点已完全关闭。
2. 使用备份 ISO 镜像引导 undercloud 节点。
3. 显示 **Relax-and-Recover** 引导菜单时，选择 **Recover <undercloud\_node>**。将 **<undercloud\_node>** 替换为 undercloud 节点的名称。



#### 注意

如果您的系统使用 UEFI，请选择 **Relax-and-Recover (no Secure Boot)** 选项。

4. 以 **root** 用户身份登录并恢复节点：  
显示以下信息：

```
Welcome to Relax-and-Recover. Run "rear recover" to restore your system!
RESCUE <undercloud_node>:~ # rear recover
```

当 undercloud 节点恢复过程完成后，控制台会显示以下信息：

```
Finished recovering your system
Exiting rear recover
Running exit tasks
```

5. 关闭该节点：

```
RESCUE <undercloud_node>:~ # poweroff
```

在引导时，节点会恢复其之前的状态。

### 2.3.2. 恢复 control plane 节点

如果在更新或升级过程中发生错误，您可以使用您使用 ReaR 创建的备份 ISO 镜像将 control plane 节点恢复到之前的状态。

要恢复 control plane，您必须恢复所有 control plane 节点以确保状态一致性。

您可以在备份节点中找到备份 ISO 镜像。将可引导 ISO 镜像刻录到 DVD，或通过 Integrated Lights-Out (iLO) 远程访问将其下载到 undercloud 节点。



#### 注意

红帽支持对带有原生 SDN 的 Red Hat OpenStack Platform 备份，如 Open vSwitch (OVS) 和默认的 Open Virtual Network (OVN)。有关第三方 SDN 的详情，请参考第三方 SDN 文档。

#### 先决条件

- 您已创建了 control plane 节点的备份。更多信息请参阅 [第 2.2.5 节“创建 control plane 节点的备份”](#)。
- 您可以访问备份节点。
- 如果将 OVS 网桥用于网络接口，您可以访问您在 **NETWORKING\_PREPARATION\_COMMANDS** 参数中设置的网络配置信息。如需更多信息，请参阅 [第 2.2.4 节“配置 Open vSwitch \(OVS\) 接口以进行备份”](#)。

#### 流程

1. 关闭每个 control plane 节点。在继续操作前，请确保 control plane 节点已完全关闭。
2. 使用对应的备份 ISO 镜像引导每个 control plane 节点。
3. 当显示 **Relax-and-Recover** 引导菜单时，在每个 control plane 节点上选择 **Recover <control\_plane\_node>**。将 **<control\_plane\_node>** 替换为对应 control plane 节点的名称。



#### 注意

如果您的系统使用 UEFI，请选择 **Relax-and-Recover (no Secure Boot)** 选项。

4. 在每个 control plane 节点上，以 **root** 用户身份登录并恢复节点：  
显示以下信息：

```
Welcome to Relax-and-Recover. Run "rear recover" to restore your system!  
RESCUE <control_plane_node>:~ # rear recover
```

当 control plane 节点恢复过程完成后，控制台会显示以下信息：

```
Finished recovering your system
Exiting rear recover
Running exit tasks
```

5. 当命令行控制台可用时，恢复每个 control plane 节点的 **config-drive** 分区：

```
# once completed, restore the config-drive partition (which is ISO9660)
RESCUE <control_plane_node>:~ $ dd if=/mnt/local/mnt/config-drive of=
<config_drive_partition>
```

6. 关闭该节点：

```
RESCUE <control_plane_node>:~ # poweroff
```

7. 将引导序列设置为普通引导设备。在引导时，节点会恢复其之前的状态。
8. 要确保服务正确运行，请检查 pacemaker 的状态。以 **root** 用户身份登录 Controller 节点，然后输入以下命令：

```
# pcs status
```

9. 要查看 overcloud 的状态，请使用 OpenStack Integration Test Suite (tempest)。如需更多信息，请参阅使用 [Integration Test Suite \(tempest\)](#) 验证您的 OpenStack 云。

## 故障排除

- 运行以下命令，清除 **pcs status** 显示的资源警报：

```
# pcs resource clean
```

- 运行以下命令，清除 **pcs status** 显示的 STONITH 隔离操作错误：

```
# pcs resource clean
# pcs stonith history cleanup
```

### 2.3.3. 手动恢复 Galera 集群

如果 Galera 集群没有作为恢复过程的一部分恢复，您必须手动恢复 Galera。



#### 注意

在此过程中，您必须在一个 Controller 节点上执行一些步骤。确保您在与完成流程相同的 Controller 节点上执行这些步骤。

#### 流程

1. 在 **Controller-0** 上，检索 Galera 集群虚拟 IP：

```
$ sudo hiera -c /etc/puppet/hiera.yaml mysql_vip
```

2. 通过所有 Controller 节点上的虚拟 IP 禁用数据库连接：

```
$ sudo iptables -I INPUT -p tcp --destination-port 3306 -d $MYSQL_VIP -j DROP
```

3. 在 **Controller-0** 上，检索 MySQL root 密码：

```
$ sudo hiera -c /etc/puppet/hiera.yaml mysql::server::root_password
```

4. 在 **Controller-0** 上，将 Galera 资源设置为 **非受管** 模式：

```
$ sudo pcs resource unmanage galera-bundle
```

5. 停止所有 Controller 节点上的 MySQL 容器：

```
$ sudo podman container stop $(sudo podman container ls --all --format "{{.Names}}" --filter=name=galera-bundle)
```

6. 移动所有 Controller 节点上的当前目录：

```
$ sudo mv /var/lib/mysql /var/lib/mysql-save
```

7. 在所有 Controller 节点上创建新目录 **/var/lib/mysql**：

```
$ sudo mkdir /var/lib/mysql
$ sudo chown 42434:42434 /var/lib/mysql
$ sudo chcon -t container_file_t /var/lib/mysql
$ sudo chmod 0755 /var/lib/mysql
$ sudo chcon -r object_r /var/lib/mysql
$ sudo chcon -u system_u /var/lib/mysql
```

8. 启动所有 Controller 节点上的 MySQL 容器：

```
$ sudo podman container start $(sudo podman container ls --all --format "{{.Names }}" --filter=name=galera-bundle)
```

9. 在所有 Controller 节点上创建 MySQL 数据库：

```
$ sudo podman exec -i $(sudo podman container ls --all --format "{{.Names }}" \
--filter=name=galera-bundle) bash -c "mysql_install_db --datadir=/var/lib/mysql --
user=mysql --log_error=/var/log/mysql/mysql_init.log"
```

10. 启动所有 Controller 节点上的数据库：

```
$ sudo podman exec $(sudo podman container ls --all --format "{{.Names }}" \
--filter=name=galera-bundle) bash -c "mysqld_safe --skip-networking --wsrep-on=OFF --
log-error=/var/log/mysql/mysql_safe.log" &
```

11. 在所有 Controller 节点上移动 **.my.cnf** Galera 配置文件：

```
$ sudo podman exec $(sudo podman container ls --all --format "{{.Names }}" \
--filter=name=galera-bundle) bash -c "mv /root/.my.cnf /root/.my.cnf.bck"
```

12. 重置所有 Controller 节点上的 Galera root 密码：

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysql -uroot -e'use mysql;set password for
root@localhost = password(\"$ROOTPASSWORD\");flush privileges;"
```

13. 在所有 Controller 节点上的 Galera 容器中恢复 **.my.cnf** Galera 配置文件：

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mv /root/.my.cnf.bck /root/.my.cnf"
```

14. 在 **Controller-0** 上，将备份数据库文件复制到 **/var/lib/MySQL**：

```
$ sudo cp $BACKUP_FILE /var/lib/mysql
$ sudo cp $BACKUP_GRANT_FILE /var/lib/mysql
```



### 注意

这些文件的路径为 `/home/tripleo-admin/`。

15. 在 **Controller-0** 上，恢复 MySQL 数据库：

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysql -u root -p$ROOT_PASSWORD <
/var/lib/mysql/$BACKUP_FILE" "
```

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysql -u root -p$ROOT_PASSWORD <
/var/lib/mysql/$BACKUP_GRANT_FILE" "
```

16. 关闭所有 Controller 节点上的数据库：

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysqladmin shutdown"
```

17. 在 **Controller-0** 上，启动 bootstrap 节点：

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) \
  /usr/bin/mysqld_safe --pid-file=/var/run/mysql/mysqld.pid --
socket=/var/lib/mysql/mysql.sock --datadir=/var/lib/mysql \
  --log-error=/var/log/mysql/mysql_cluster.log --user=mysql --open-files-limit=16384 \
  --wsrep-cluster-address=gcomm:// &
```

18. 验证：在 **Controller-0** 上，检查集群的状态：

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "clustercheck"
```

确保显示以下信息：“Galera 集群节点同步”，否则您必须重新创建该节点。

19. 在 **Controller-0** 上，从配置检索集群地址：

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) bash -c "grep wsrep_cluster_address /etc/my.cnf.d/galera.cnf |
awk '{print $3}'"
```

20. 在每个剩余的 Controller 节点上，启动数据库并验证集群：

a. 启动数据库：

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) /usr/bin/mysqld_safe --pid-
file=/var/run/mysql/mysqld.pid --socket=/var/lib/mysql/mysql.sock \
--datadir=/var/lib/mysql --log-error=/var/log/mysql/mysql_cluster.log --user=mysql --
open-files-limit=16384 \
--wsrep-cluster-address=$CLUSTER_ADDRESS &
```

b. 检查 MYSQL 集群的状态：

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) bash -c "clustercheck"
```

确保显示以下信息：“Galera 集群节点同步”，否则您必须重新创建该节点。

21. 停止所有 Controller 节点上的 MySQL 容器：

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) \
/usr/bin/mysqladmin -u root shutdown
```

22. 在所有 Controller 节点上，删除以下防火墙规则以允许通过虚拟 IP 地址进行数据库连接：

```
$ sudo iptables -D INPUT -p tcp --destination-port 3306 -d $MYSQL_VIP -j DROP
```

23. 重启所有 Controller 节点上的 MySQL 容器：

```
$ sudo podman container restart $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle)
```

24. 重启所有 Controller 节点上的 **clustercheck** 容器：

```
$ sudo podman container restart $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=clustercheck)
```

25. 在 **Controller-0** 上，将 Galera 资源设置为 **受管** 模式：

```
$ sudo pcs resource manage galera-bundle
```

## 验证

1. 要确保服务正确运行，请检查 pacemaker 的状态：

```
$ sudo pcs status
```

2. 要查看 overcloud 的状态，请使用 OpenStack Integration Test Suite (tempest)。如需更多信息，请参阅使用 [Integration Test Suite \(tempest\)](#) 验证您的 OpenStack 云。
3. 如果您怀疑特定节点有问题，请使用 **clustercheck** 检查集群的状态：

```
$ sudo podman exec clustercheck /usr/bin/clustercheck
```

### 2.3.4. 手动恢复 undercloud 节点数据库

如果 undercloud 数据库没有作为 undercloud 恢复过程的一部分恢复，您可以手动恢复数据库。只有之前创建了独立数据库备份时，才能恢复数据库。

#### 先决条件

- 您已创建了 undercloud 数据库的独立备份。更多信息请参阅 [第 2.1.6 节“可选：创建 undercloud 节点的独立数据库备份”](#)。

#### 流程

1. 以 **root** 用户身份登录 director undercloud 节点。
2. 停止所有 tripleo 服务：

```
[root@director ~]# systemctl stop tripleo_*
```

3. 输入以下命令确保没有容器在服务器上运行：

```
[root@director ~]# podman ps
```

如果有任何容器正在运行，请输入以下命令来停止容器：

```
[root@director ~]# podman stop <container_name>
```

4. 创建当前 **/var/lib/mysql** 目录的备份，然后删除该目录：

```
[root@director ~]# cp -a /var/lib/mysql /var/lib/mysql_bck
[root@director ~]# rm -rf /var/lib/mysql
```

5. 重新创建数据库目录并为新目录设置 SELinux 属性：

```
[root@director ~]# mkdir /var/lib/mysql
[root@director ~]# chown 42434:42434 /var/lib/mysql
[root@director ~]# chmod 0755 /var/lib/mysql
[root@director ~]# chcon -t container_file_t /var/lib/mysql
[root@director ~]# chcon -r object_r /var/lib/mysql
[root@director ~]# chcon -u system_u /var/lib/mysql
```

6. 为 **mariadb** 镜像创建一个本地标签。将 **<image\_id>** 和 **<undercloud.ctlplane.example.com>** 替换为环境中适用的值：

```
[root@director ~]# podman images | grep mariadb
<undercloud.ctlplane.example.com>:8787/rh-osbs/rhosp16-openstack-mariadb
16.2_20210322.1 <image_id> 3 weeks ago 718 MB
```

-

```
[root@director ~]# podman tag <image_id> mariadb
```

```
[root@director ~]# podman images | grep maria
localhost/mariadb          latest      <image_id> 3
weeks ago 718 MB
<undercloud.ctlplane.example.com>:8787/rh-osbs/rhosp16-openstack-mariadb
16.2_20210322.1 <image_id> 3 weeks ago 718 MB
```

7. 使用容器初始化 `/var/lib/mysql` 目录：

```
[root@director ~]# podman run --net=host -v /var/lib/mysql:/var/lib/mysql localhost/mariadb
mysql_install_db --datadir=/var/lib/mysql --user=mysql
```

8. 复制您要导入到数据库的数据库备份文件：

```
[root@director ~]# cp /root/undercloud-all-databases.sql /var/lib/mysql
```

9. 启动数据库服务来导入数据：

```
[root@director ~]# podman run --net=host -dt -v /var/lib/mysql:/var/lib/mysql
localhost/mariadb /usr/libexec/mysqld
```

10. 导入数据并配置 `max_allowed_packet` 参数：

- a. 登录到容器并进行配置：

```
[root@director ~]# podman exec -it <container_id> /bin/bash
()[mysql@5a4e429c6f40 /]$ mysql -u root -e "set global max_allowed_packet =
1073741824;"
()[mysql@5a4e429c6f40 /]$ mysql -u root < /var/lib/mysql/undercloud-all-
databases.sql
()[mysql@5a4e429c6f40 /]$ mysql -u root -e 'flush privileges'
()[mysql@5a4e429c6f40 /]$ exit
exit
```

- b. 停止容器：

```
[root@director ~]# podman stop <container_id>
```

- c. 检查没有容器是否正在运行：

```
[root@director ~]# podman ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[root@director ~]#
```

11. 重启所有 tripleo 服务：

```
[root@director ~]# systemctl start multi-user.target
```

