



Red Hat OpenStack Platform 16.2

创建和管理实例

OpenStack Team
rhos-docs@redhat.com

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[®] 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

目录

前言	4
使开源包含更多	5
对红帽文档提供反馈	6
第 1 章	7
第 2 章	8
第 3 章	9
3.1.	9
3.2.	9
3.3.	9
3.4.	9
3.5.	9
第 4 章	10
第 5 章	11
5.1. 先决条件	11
5.2.	11
5.3.	11
5.4.	12
5.5. 其他资源	13
第 6 章	14
6.1.	15
第 7 章	16
7.1.	16
7.2.	16
7.3.	16
7.4.	17
7.5.	17
7.6.	18
7.7.	18
第 8 章	20
8.1. 先决条件	20
8.2.	20
8.3.	23
8.4.	23
8.5.	23
8.6. 其他资源	25
第 9 章 连接到实例	26
9.1.	26
9.2. 登录到一个实例	26
第 10 章 管理实例	28
10.1. 调整实例大小	28
10.2. 创建实例快照	29
10.3. 修正一个实例	29
10.4. SHELVING 一个实例	30

10.5. 实例管理操作	30
第 11 章 创建自定义实例	34
11.1. 使用用户数据自定义实例	35
11.2. 使用元数据自定义实例	36
11.3. 使用配置驱动器自定义实例	36

前言



注意

您无法在实例创建过程中将基于角色的访问控制 (RBAC) 共享安全组直接应用到实例。要将 RBAC 共享安全组应用到实例，您必须首先创建端口，将共享安全组应用到该端口，然后将该端口分配给实例。

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。

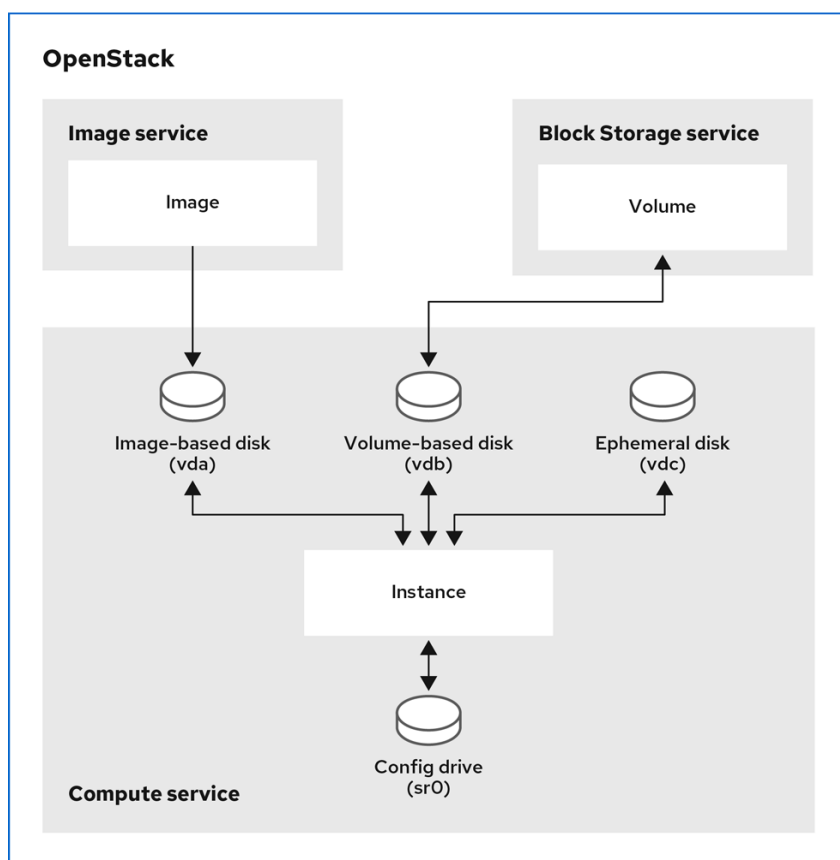
- 1.
- 2.
- 3.
4. **点 Create。**

第1章

计算服务是按需提供实例的中央组件。

第 2 章

您可以从同一基础镜像启动任意数量的实例。



145_OpenStack_0321

第 3 章

-
- 临时存储
-
-
-

3.1.

3.2.

⚠️ 注意

3.3.

3.4.

⚠️ 注意

3.5.

您可以在启动时将配置驱动器附加到实例。

第 4 章

第 5 章



注意

5.1. 先决条件

- ○
-
-
-
-

5.2.

流程

1. `$ openstack flavor list`



注意

2. `$ openstack image list`



注意

- `hw_scsi_model=virtio-scsi`
- `hw_disk_bus=scsi`

3. `$ openstack network list`

4. `$ openstack server create --flavor <flavor> \`
`--image <image> --network <network> \`
`--wait myInstanceFromImage`

-
-
-

5.3.



注意

流程

1. `$ openstack image list`



注意

- `hw_scsi_model=virtio-scsi`
- `hw_disk_bus=scsi`

2. `$ openstack volume create --image <image> \`
`--size <size_gb> --bootable myBootableVolume`

-
- 将 `<size_gb>` 替换为卷的大小（以 GB 为单位）。

3. `$ openstack flavor list`

4. `$ openstack network list`

5. `$ openstack server create --flavor <flavor> \`
`--volume myBootableVolume --network <network> \`
`--wait myInstanceFromVolume`

-
-

5.4.

流程

1. `$ openstack flavor list`



注意


提示

2. `$ openstack image list`

提示

3. `$ openstack network list`

4. `$ openstack port create --network <network> \`
`--vnic-type <vnic_type> mySriovPort`

- - ○
 -
 -
5.  \$ openstack server create --flavor <flavor> \
--image <image> --port <port> \
--wait mySriovInstance

-
-
-

5.5. 其他资源

-

第 6 章

-
- `$ openstack port show <port_name/port_id>`

先决条件

-

流程

1. `(overcloud)$ openstack network qos policy list`

```

-----+
| ID                | Name  | Shared | Default | Project          |
-----+
| 6d771447-3cf4-4ef1-b613-945e990fa59f | policy2 | True  | False  | ba4de51bf7694228a350dd22b7a3dc24 |
| 78a24462-e3c1-4e66-a042-71131a7daed5 | policy1 | True  | False  | ba4de51bf7694228a350dd22b7a3dc24 |
| b80acc64-4fc2-41f2-a346-520d7cfe0e2b | policy0 | True  | False  | ba4de51bf7694228a350dd22b7a3dc24 |
-----+

```

2. `(overcloud)$ openstack network qos policy show policy0`

```

-----+
| Field  | Value                                                                 |
-----+
| description |
| id       | b80acc64-4fc2-41f2-a346-520d7cfe0e2b
| is_default | False
| location  | cloud=', project.domain_id=', project.domain_name='Default,
project.id=ba4de51bf7694228a350dd22b7a3dc24, project.name=admin,
region_name=regionOne, zone=
| name     | policy0
| project_id | ba4de51bf7694228a350dd22b7a3dc24
| rules    | [{min_kbps: 100000, direction: egress, id: d46218fe-9218-4e96-952b-
9f45a5cb3b3c, qos_policy_id: b80acc64-4fc2-41f2-a346-520d7cfe0e2b, type:
minimum_bandwidth}, {min_kbps: 100000, direction: ingress, id: 1202c4e3-a03a-464c-80d5-
0bf90bb74c9d, qos_policy_id: b80acc64-4fc2-41f2-a346-520d7cfe0e2b, type:
minimum_bandwidth}] |
| shared   | True
|

```

```
| tags    | []
```

```
|
```

```
-----+
```

3. (overcloud)\$ openstack port create port-normal-qos --network net0 --qos-policy policy0
4. \$ openstack server create --flavor cirros256 --image cirros-0.3.5-x86_64-disk --nic port-id=port-normal-qos --wait qos_instance

6.1.

流程

- \$ openstack server remove port <vm_name|vm_id> <port_name|port_id>

第 7 章

7.1.

流程

1.

```
(overcloud)$ openstack network list
```

```
(overcloud)$ openstack network create <network>
```
2.

```
$ openstack server add network <instance> <network>
```

 -
 -

其他资源

-
-

7.2.

注意

流程

1.

```
(overcloud)$ openstack server show <instance>
```
2.

```
$ openstack server remove network <instance> <network>
```

 -
 -

7.3.

提示

注意

流程

1.

```
(overcloud)$ openstack port list
```

```
(overcloud)$ openstack port create --network <network> <port>
```

-
-
- 2. `$ openstack server add port <instance> <port>`

-
-

其他资源

-
-

7.4.

流程

1. `(overcloud)$ openstack server show <instance>`
2. `$ openstack server remove port <instance> <port>`

-
-

7.5.

先决条件

-
- - `hw_scsi_model=virtio-scsi`
 - `hw_disk_bus=scsi`

流程

1. `(overcloud)$ openstack volume list`
2. `$ openstack server add volume <instance> <volume>`

-



注意

Multiattach volumes are only supported starting with compute API version 2.60. (HTTP 400) (Request-ID: req-3a969c31-e360-4c79-a403-75cc6053c9e5)

\$ openstack --os-compute-api-version 2.72 server add volume <instance> <volume>

提示

3. \$ openstack volume show <volume>

输出示例：

```

+-----+-----+-----+-----+-----+
| ID                | Name          | Status | Size | Attached to |
+-----+-----+-----+-----+-----+
| f3fb92f6-c77b-429f-871d-65b1e3afa750 | volMultiattach | in-use | 50 | Attached to instance1 on /dev/vdb Attached to instance2 on /dev/vdb |
+-----+-----+-----+-----+-----+
    
```

7.6.

先决条件

-

流程

• \$ openstack server volume list <instance>

```

+-----+-----+-----+-----+
| ID                | Device | Server ID      | Volume ID      |
+-----+-----+-----+-----+
| 1f9dcb02-9a20-4a4b- | /dev/vda | ab96b635-1e63-4487- | 1f9dcb02-9a20-4a4b-9f |
| 9f25-c7846a1ce9e8 |         | a85c-854197cd537b | 25-c7846a1ce9e8    |
+-----+-----+-----+-----+
    
```

7.7.



注意

流程

1. (overcloud)\$ openstack server show <instance>

2. `$ openstack server remove volume <instance> <volume>`

-

-  注意

第 8 章

8.1. 先决条件

-
-

8.2.

安全组规则应用到项目中的所有实例。



注意

您无法在实例创建过程中将基于角色的访问控制 (RBAC) 共享安全组直接应用到实例。要将 RBAC 共享安全组应用到实例，您必须首先创建端口，将共享安全组应用到该端口，然后将该端口分配给实例。



注意

8.2.1.

流程

1.

```
$ openstack security group list
$ openstack security group rule list <sec_group>
```

 -
2.

```
$ openstack security group create mySecGroup
```
3.

```
$ openstack security group rule create --protocol <protocol> \
[--dst-port <port-range>] \
[--remote-ip <ip-address> | --remote-group <group>] \
[--ingress | --egress] mySecGroup
```

 -
 -
 -
 -
4.

```
$ openstack security group rule create --protocol tcp \
--dst-port 22 mySecGroup
```

8.2.2.

流程

1. `$ openstack security group list`
- 2.
3. `$ openstack security group rule create --protocol <protocol> \
 [--dst-port <port-range>] \
 [--remote-ip <ip-address> | --remote-group <group>] \
 [--ingress | --egress] <group_name>`
 -
 -
 -
 -
 -
4. `$ openstack security group rule create --protocol tcp \
 --dst-port 22 mySecGroup`

8.2.3.

流程

1. `$ openstack security group list`
2. `$ openstack security group show <sec-group>`
3. `$ openstack security group rule delete <rule> [<rule> ...]`

8.2.4.

流程

1. `$ openstack port list --server myInstancewithSSH`
2. `$ openstack port set --security-group <sec_group> <port>`

8.2.5.

流程

1. `$ openstack port show <port>`
2. `$ openstack port set --no-security-group <port>`

3.

```
$ openstack port set --security-group <sec_group> <port>
```

8.2.6.

流程

1.

```
$ openstack security group list
```
2.

```
$ openstack port list
```
3.

```
$ openstack port show <port-uuid> -c security_group_ids
```
4.

```
$ openstack security group delete <group> [<group> ...]
```

8.2.7.



注意

流程

1.

```
$ openstack keypair create <keypair> > ~/.ssh/<keypair>.pem
```
2.

```
$ chmod 600 ~/.ssh/<keypair>.pem
```

8.2.8.

流程

1.
 - ```
$ openstack keypair create --public-key ~/.ssh/<public_key>.pub \
<keypair> > ~/.ssh/<keypair>.pem
```

    - 
    -
  - ```
$ openstack keypair create --private-key ~/.ssh/<private_key> \  
<keypair> > ~/.ssh/<keypair>.pem
```

 -
 -
2.

```
$ chmod 600 ~/.ssh/<keypair>.pem
```

8.2.9. 其他资源

-

-

8.3.

先决条件

-
-

流程

1.

```
$ openstack floating ip list
```

```
$ openstack floating ip create <provider-network>
```

提示

2.

```
$ openstack server add floating ip [--fixed-ip-address <ip_address>] \  
<instance> <floating_ip>
```

 -
 -
 -
3.

```
$ openstack server show <instance>
```

其他资源

-

8.4.

流程

1.

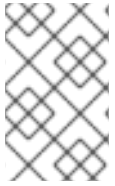
```
$ openstack server remove floating ip <instance> <ip_address>
```

 -
 -
2.

```
$ openstack floating ip delete <ip_address>
```
3.

```
$ openstack floating ip list
```

8.5.

**注意****注意**

您无法在实例创建过程中将基于角色的访问控制 (RBAC) 共享安全组直接应用到实例。要将 RBAC 共享安全组应用到实例，您必须首先创建端口，将共享安全组应用到该端口，然后将该端口分配给实例。

先决条件

-
-
-
-
-
-

流程

1. `$ openstack flavor list`

**注意**

2. `$ openstack image list`

3. `$ openstack network list`

4. `$ openstack keypair list`

5. `$ openstack server create --flavor <flavor> \
--image <image> --network <network> \
[--security-group <secgroup>] \
--key-name <keypair> --wait myInstancewithSSH`

-
-
-
-
-

6. `$ openstack server add floating ip myInstancewithSSH <floating_ip>`

7. `$ ssh -i ~/.ssh/<keypair>.pem cloud-user@<floatingIP>`
`[cloud-user@demo-server1 ~]$`

8.6. 其他资源

-
-

第 9 章 连接到实例



注意

9.1.

流程

1. 要显示实例的 VNC 控制台 URL，请输入以下命令：

```
$ openstack console url show <vm_name>
+-----+-----+
| Field | Value          |
+-----+-----+
| type  | novnc          |
| url   | http://172.25.250.50:6080/vnc_auto.html?token=      |
|       | 962dfd71-f047-43d3-89a5-13cb88261eb9              |
+-----+-----+
```

2. 要直接连接到 VNC 控制台，请在浏览器中输入显示的 URL。

9.2. 登录到一个实例

您可以远程登录到公共实例。

先决条件

- 您有实例的密钥对证书。证书在创建密钥对时下载。如果您没有自己创建密钥对，请咨询您的管理员。
- 实例配置为公共实例。有关公共实例要求的更多信息，请参阅 [向实例提供公共访问](#)。
- 您有一个云用户帐户。

流程

1. 检索您要登录的实例的浮动 IP 地址：

```
$ openstack server show <instance>
```

将 `<instance>` 替换为您要连接的实例的名称或 ID。

2. 使用自动创建的 `cloud-user` 帐户登录到您的实例：

```
$ ssh -i ~/.ssh/<keypair>.pem cloud-user@<floatingIP>
[cloud-user@demo-server1 ~]$
```

- 将 `<rhcs>` 替换为密钥对的名称。
- 将 `<floating_ip>` 替换为实例的浮动 IP 地址。

提示

您可以使用以下命令在没有浮动 IP 地址的情况下登录到实例：

```
$ openstack server ssh --login cloud-user \  
--identity ~/.ssh/<keypair>.pem --private <instance>
```

- 将 `< rhcs >` 替换为密钥对的名称。
- 将 `& It;instance >` 替换为您要连接的实例的名称或 ID。

第 10 章 管理实例

您可以对一个实例执行管理操作，如调整实例的大小或她实例的大小。有关管理操作的完整列表，请参阅[实例管理操作](#)。

10.1. 调整实例大小

如果需要增加或减少实例的内存或 CPU 计数，您可以调整实例的大小。若要调整实例大小，请为具有所需容量的实例选择一个新类别。重新定义实例重建并重启实例的大小。

流程

1. 检索您要调整大小的实例的名称或 ID：

```
$ openstack server list
```

2. 检索您要用来调整实例大小的类别的名称或 ID：

```
$ openstack flavor list
```

3. 重新定义实例大小：

```
$ openstack server resize --flavor <flavor> \  
--wait <instance>
```

- 将 **<flavor>** 替换为在第 2 步中检索的类别的名称或 ID。
- 将 **<instance>** 替换为您要调整大小的实例的名称或 ID。



注意

调整大小可能需要时间。实例上的操作系统在实例关闭前执行受控的关闭，并调整实例大小。在这个时间中，实例状态为 **RESIZE**：

```
$ openstack server list  
+-----+-----+-----+-----+  
| ID          | Name          | Status | Networks          |  
+-----+-----+-----+-----+  
| 67bc9a9a-5928-47c... | myCirrosServer | RESIZE | admin_internal_net=192.168.111.139 |  
+-----+-----+-----+-----+
```

4. 调整大小完成后，实例状态变为 **VERIFY_RESIZE**。现在，您必须确认或恢复大小：

- 要确认调整大小，请输入以下命令：

```
$ openstack server resize confirm <instance>
```

- 要恢复调整大小，请输入以下命令：

```
$ openstack server resize revert <instance>
```


该实例恢复到原始类别，状态更改为 **ACTIVE**。



注意

如果没有在配置的时间段内确认或恢复，云可能被配置为自动确认实例大小。

10.2. 创建实例快照

快照是捕获实例的运行磁盘状态的镜像。您可以对实例进行快照来创建可用作模板的镜像来创建新实例。快照允许您从另一个实例创建新实例，并恢复实例的状态。如果删除基于快照的实例，您可以使用快照镜像将新实例创建与快照相同的状态。

流程

1. 检索您要对其进行快照的实例的名称或 ID：

```
$ openstack server list
```

2. 创建快照：

```
$ openstack server image create --name <image_name> <instance>
```

- 将 `<image_name>` 替换为新快照镜像的名称。
- 将 `<instance>` 替换为您要从其中创建快照的实例的名称或 ID。

3. 可选：当您使用实例快照作为模板创建新实例时，要确保磁盘状态一致，请启用 QEMU 客户机代理并指定文件系统必须在快照处理过程中被静止：

```
$ openstack image set --property hw_qemu_guest_agent=yes \
  --property os_require_quiesce=yes <image_name>
```

QEMU 客户机代理是一个后台进程，可帮助管理应用程序执行实例操作系统级别命令。启用此代理向实例添加另一个设备，这会消耗 PCI 插槽，并限制您可以分配给实例的其他设备数量。它还会导致 Windows 实例显示有关未知硬件设备的警告信息。

10.3. 修正一个实例

在紧急情况下，如系统故障或访问失败，您可以将实例置于救援模式中。这会关闭实例，使用新实例磁盘重新启动它，并在重启的实例上将原始实例磁盘和配置驱动器挂载为卷。您可以连接到重启的实例，以查看原始实例磁盘来修复系统并恢复数据。



注意

您无法救援从卷引导的实例。

流程

1. 执行实例救援：

```
$ openstack server rescue [--image <image>] <instance>
```

- 可选：默认情况下，实例从云管理员提供的救援镜像引导，或原始实例镜像的新副本。使用 `-image` 选项指定在以救援模式重启实例时使用的替代镜像。
- 将 `<instance>` 替换为您要救援实例的名称或 ID。

2. 连接到救援实例来解决这个问题。

3. 从普通引导磁盘重启实例：

```
$ openstack server unrescue <instance>
```

10.4. SHELVING 一个实例

如果您有一个不使用的实例，则 shelving 很有用，但您不想删除。当您 shelve 实例时，您可以保留实例数据和资源分配，但清除实例内存。根据云配置，Mandlved 实例会立即移到 `SHELVED_OFFLOADED` 状态，或延迟时间后。当 `SHELVED_OFFLOADED` 时，实例数据和资源分配会被删除。

当您 shelve 实例时，计算服务会生成一个快照镜像，该镜像捕获实例的状态，并以以下格式为镜像分配一个名称：`<instance>-shelved`。当实例被取消删除时，会删除此快照镜像。

如果您不再需要 shelved 实例，您可以将其删除。您可以一次Mandlve 多个实例。

流程

1. 检索您要 shelve 的实例或实例的名称或 ID：

```
$ openstack server list
```

2. herlve 实例或实例：

```
$ openstack server shelve <instance> [<instance> ...]
```

将 `<instance>` 替换为您要 shelve 的实例的名称或 ID。您可以根据需要为 shelve 指定多个实例。

3. 验证实例是否已是 shelved：

```
$ openstack server list
```

shelved 实例的状态为 `SHELVED_OFFLOADED`。

10.5. 实例管理操作

创建实例后，您可以执行以下管理操作。

表 10.1. 管理操作

操作	描述	命令
停止一个实例	停止该实例。	OpenStack 服务器停止
启动实例	启动已停止的实例。	OpenStack 服务器启动

操作	描述	命令
暂停正在运行的实例	立即暂停正在运行的实例。实例的状态存储在内存中(RAM)中。暂停的实例继续以冻结状态运行。系统将提示您确认暂停操作。	OpenStack 服务器暂停
恢复暂停实例的运行	立即恢复暂停的实例。系统将提示您确认恢复操作。	OpenStack server unpause
挂起正在运行的实例	立即暂停正在运行的实例。实例的状态存储在实例磁盘上。系统将提示您确认挂起操作。	OpenStack 服务器挂起
恢复暂停实例的运行	立即恢复暂停的实例。实例的状态存储在实例磁盘上。系统将提示您确认恢复操作。	OpenStack 服务器恢复
删除实例	<p>永久销毁实例。系统将提示您确认 <code>destroy</code> 操作。除非云被配置为启用软删除，否则已删除的实例将无法恢复。</p> <div style="display: flex; align-items: center;">  <div> <p>注意</p> <p>删除实例不会删除其附加的卷。您必须单独删除附加的卷。如需更多信息，请参阅存储指南中的删除块存储服务卷。</p> </div> </div>	OpenStack server delete
编辑实例元数据	您可以使用实例元数据来指定实例的属性。如需更多信息， 请参阅创建自定义实例。	openstack server set --property <key=value> [--property <key=value>] <instance>
添加安全组	将指定的安全组添加到实例。	OpenStack 服务器添加安全组
删除安全组	从实例移除指定的安全组。	OpenStack 删除安全组

操作	描述	命令
救援实例	<p>在紧急情况下，如系统故障或访问失败，您可以将实例置于救援模式中。这会关闭实例，并将根磁盘挂载到临时服务器。您可以连接到临时服务器来修复系统并恢复数据。</p> <p>也可以将正在运行的实例重新引导为救援模式。例如，如果实例的文件系统损坏，则可能需要此操作。</p> <p> 注意</p> <p>您无法救援从卷引导的实例。</p>	OpenStack 服务器救援
恢复救援实例	重启救援实例。	OpenStack server unrescue
查看实例日志	查看实例控制台日志的最新部分。	OpenStack 控制台日志显示
shelve 一个实例	<p>当您 shelve 实例保留实例数据和资源分配时，但清除实例内存。根据云配置，Mandlved 实例会立即移到 SHELVED_OFFLOADED 状态，或延迟时间后。当一个实例处于 SHELVED_OFFLOADED 状态时，实例数据和资源分配会被删除。实例的状态存储在实例磁盘上。如果实例从卷引导，它会立即进入 SHELVED_OFFLOADED。系统将提示您确认 shelve 操作。</p>	OpenStack server shelve
取消提升一个实例	使用 shelved 实例的磁盘镜像恢复实例。	OpenStack server unshelve
锁定实例	锁定实例，以防止非管理员用户对实例执行操作。	OpenStack 服务器锁定 OpenStack 服务器解锁
软重启实例	正常停止并重启实例。软重启会在重启实例前尝试安全关闭所有进程。默认情况下，当您重启实例时，它是一个软重启。	openstack server reboot --soft <server>
硬重启实例	停止并重启实例。硬重启关闭实例电源，然后重新打开它。	openstack server reboot --hard <server>

操作	描述	命令
重建一个实例	使用新镜像和磁盘分区选项重建实例，实例涉及关闭、重新镜像和重启实例。如果您遇到操作系统问题，而不是终止实例并启动，则使用此选项。	OpenStack 服务器重建

第 11 章 创建自定义实例

云用户可以指定在启动实例时要使用的附加数据，如实例在引导时运行的 shell 脚本。云用户可以使用以下方法将数据传递给实例：

用户数据

用于在实例启动命令中包括要执行的 cloud-init 的说明。

实例元数据

创建或更新实例时可以指定的键值对列表。

您可以使用配置驱动器或元数据服务访问传递给实例的额外数据。

配置驱动器

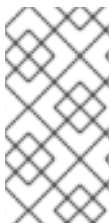
您可以在实例引导时将配置驱动器附加到实例。配置驱动器以只读驱动器形式呈现给实例。实例可以挂载此驱动器并从中读取文件。您可以使用配置驱动器作为 cloud-init 信息的源。当与 cloud-init 用于服务器引导时，配置驱动器很有用，当您要将大型文件传递给实例时。例如，您可以将 cloud-init 配置为自动挂载配置驱动器，并在初始实例引导期间运行设置脚本。配置驱动器使用 config-2 的卷标签创建，并在引导时附加到实例。传递给配置驱动器的任何其他文件内容都添加到配置驱动器的 `openstack/{version}/` 目录中的 `user_data` 文件中。cloud-init 从此文件检索用户数据。

元数据服务

提供 REST API 以检索特定于实例的数据。实例通过 `169.254.169.254` 或 `fe80::a9fe:a9fe` 访问此服务。

cloud-init 可以使用 config 驱动器和元数据服务来消耗额外的数据来自定义实例。cloud-init 软件包支持多个数据输入格式。shell 脚本和 cloud-config 格式是最常见的输入格式：

- shell 脚本：数据声明以 `192.168.1.0/24!` 或 `Content-Type: text/x-shellscript` 开头。shell 脚本在引导过程中最后一次调用。
- cloud-config 格式：数据声明以 `evince cloud-config` 或 `Content-Type: text/cloud-config` 开头。cloud-config 文件必须是有效的 YAML，才能由 cloud-init 解析并执行。



注意

对于传递给实例的数据，cloud-init 的最大用户数据大小为 16384 字节。您无法更改大小限制，因此当您需要超过大小限制时，使用 gzip 压缩。

特定于供应商的数据

RHOSP 管理员也可以在创建数据时将数据传递给实例。此数据可能无法以云用户身份看到，例如，将实例注册到 Active Directory 的加密令牌。

RHOSP 管理员使用 `vendordata` 功能将数据传递给实例。vendordata 配置是只读的，位于以下文件中：

- `/openstack/{version}/vendor_data.json`
- `/openstack/{version}/vendor_data2.json`

您可以使用元数据服务或实例中的配置驱动器查看这些文件。要使用元数据服务访问文件，请对 `http://169.254.169.254/openstack/{version}/vendor_data.json` 或 `http://169.254.169.254/openstack/{version}/vendor_data2.json` 发出 GET 请求。

11.1. 使用用户数据自定义实例

您可以使用用户数据在 `instance launch` 命令中包含指令。`cloud-init` 执行这些命令，将实例自定义为引导过程中的最后一个步骤。

流程

1. 使用 `cloud-init` 指令创建一个文件。例如，创建一个 `bash` 脚本，用于在实例上安装并启用 Web 服务器：

```
$ vim /home/scripts/install_httpd
#!/bin/bash

yum -y install httpd python-psycopg2
systemctl enable httpd --now
```

2. 使用 `--user-data` 选项启动实例，以传递 `bash` 脚本：

```
$ openstack server create \
--image rhel8 \
--flavor default \
--nic net-id=web-server-network \
--security-group default \
--key-name web-server-keypair \
--user-data /home/scripts/install_httpd \
--wait web-server-instance
```

3. 当实例状态为 `active` 时，附加一个浮动 IP 地址：

```
$ openstack floating ip create web-server-network
$ openstack server add floating ip web-server-instance 172.25.250.123
```

4.

使用 **SSH** 登录实例：

```
$ ssh -i ~/.ssh/web-server-keypair cloud-user@172.25.250.123
```

5.

检查自定义是否已成功执行。例如，要检查 **Web 服务器** 是否已安装并启用，请输入以下命令：

```
$ curl http://localhost | grep Test
<title>Test Page for the Apache HTTP Server on Red Hat Enterprise Linux</title>
<h1>Red Hat Enterprise Linux <strong>Test Page</strong></h1>
```

6.

查看 `/var/log/cloud-init.log` 文件以了解相关消息，如是否执行 `cloud-init`：

```
$ sudo less /var/log/cloud-init.log
...output omitted...
...util.py[DEBUG]: Cloud-init v. 0.7.9 finished at Sat, 23 Jun 2018 02:26:02 +0000.
Datasource DataSourceOpenStack [net,ver=2]. Up 21.25 seconds
```

11.2. 使用元数据自定义实例

您可以使用实例元数据在 `instance launch` 命令中指定实例的属性。

流程

1.

使用 `--property <key=value >` 选项启动实例。例如，要将实例标记为 `webserver`，请设置以下属性：

```
$ openstack server create \
--image rhel8 \
--flavor default \
--property role=webservers \
--wait web-server-instance
```

2.

可选：在创建实例后为实例添加一个附加属性，例如：

```
$ openstack server set \
--property region=emea \
--wait web-server-instance
```

11.3. 使用配置驱动器自定义实例

您可以为实例引导过程中附加的实例创建配置驱动器。您可以将内容传递给配置驱动器提供给实例的配置驱动器。

流程

1. 启用配置驱动器，并指定包含您要在配置驱动器中提供的内容的文件。例如，以下命令会创建一个名为 **config-drive-instance** 的新实例，并附加包含文件 **my-user-data.txt** 的配置驱动器：

```
(overcloud)$ openstack server create --flavor m1.tiny \
  --config-drive true \
  --user-data ./my-user-data.txt \
  --image cirros config-drive-instance
```

此命令使用 **config-2** 的卷标签创建配置驱动器，该标签在引导时附加到实例，并将 **my-user-data.txt** 的内容添加到配置驱动器的 **openstack/{version}/** 目录中的 **user_data** 文件中。

2. 登录该实例。

3. 挂载配置驱动器：

- 如果实例操作系统使用 **udev**：

```
# mkdir -p /mnt/config
# mount /dev/disk/by-label/config-2 /mnt/config
```

- 如果实例操作系统没有使用 **udev**，则首先需要识别与配置驱动器对应的块设备：

```
# blkid -t LABEL="config-2" -odevice
/dev/vdb
# mkdir -p /mnt/config
# mount /dev/vdb /mnt/config
```