



# Red Hat OpenStack Platform 17.1

## 使用 Red Hat OpenStack Platform Integration Test Suite 验证您的云

验证 Red Hat OpenStack Platform 的部署



# Red Hat OpenStack Platform 17.1 使用 Red Hat OpenStack Platform Integration Test Suite 验证您的云

---

验证 Red Hat OpenStack Platform 的部署

OpenStack Team  
rhos-docs@redhat.com

## 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<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

在 Red Hat OpenStack Platform 环境中安装、配置和管理 OpenStack Integration Test Suite (tempest)，以便您可以验证您的部署。

---

# 目录

使开源包含更多 .....	3
对红帽文档提供反馈 .....	4
第 1 章 OPENSTACK INTEGRATION TEST SUITE (TEMPEST)验证 .....	5
第 2 章 安装 INTEGRATION TEST SUITE (TEMPEST) .....	6
2.1. 先决条件 .....	6
2.2. 手动安装 INTEGRATION TEST SUITE .....	6
第 3 章 配置集成测试套件(TEMPEST) .....	9
3.1. 先决条件 .....	9
3.2. 创建工作区 .....	9
3.3. 手动配置 INTEGRATION TEST SUITE .....	10
3.4. 配置 INTEGRATION TEST SUITE 日志 .....	11
3.5. 配置 INTEGRATION TEST SUITE MICROVERSION 测试 .....	12
第 4 章 清理 INTEGRATION TEST SUITE (TEMPEST)资源 .....	13
4.1. 执行空运行 .....	13
4.2. 执行 TEMPEST 清理 .....	13
第 5 章 使用 INTEGRATION TEST SUITE (TEMPEST)验证 OPENSTACK 云 .....	15
5.1. 先决条件 .....	15
5.2. 列出可用的测试 .....	15
5.3. 运行烟雾测试 .....	15
5.4. 使用允许列表文件传递测试 .....	15
5.5. 使用 BLOCKLIST 文件跳过测试 .....	15
5.6. 在并行或系列中运行测试 .....	16
5.7. 运行特定的测试 .....	16
5.8. 删除 INTEGRATION TEST SUITE 对象 .....	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。

### 在 JIRA 中提供文档反馈

使用 [Create Issue](#) 表单对文档提供反馈。JIRA 问题将在 Red Hat OpenStack Platform Jira 项目中创建，您可以在其中跟踪您的反馈进度。

1. 确保您已登录到 JIRA。如果您没有 JIRA 帐户，请创建一个帐户来提交反馈。
2. 点击以下链接打开 **Create Issue** 页面：[Create Issue](#)
3. 完成 **Summary** 和 **Description** 字段。在 **Description** 字段中，包含文档 URL、章节或章节号以及问题的详细描述。不要修改表单中的任何其他字段。
4. 点 **Create**。



# 第1章 OPENSTACK INTEGRATION TEST SUITE (TEMPEST)验证

因为 Red Hat OpenStack Platform (RHOSP)由许多不同的项目组成，因此测试 RHOSP 集群中项目的互操作性非常重要。OpenStack Integration Test Suite 自动测试您的 RHOSP 部署。您可以运行测试以确保集群按预期工作。测试输出以提供潜在问题的早期警告，特别是在升级后。

Integration Test Suite 包含用于 OpenStack API 验证和场景测试的测试，以及用于自验证的单元测试。Integration Test Suite 使用 OpenStack 公共 API 和 tempest 作为测试运行程序来执行黑色测试。

OpenStack Integration Test Suite (tempest)充当向 Red Hat OpenStack Platform (RHOSP)核心项目的承诺，它可以压力测试以在云部署上生成负载，并且它可以执行 CLI 测试来检查命令行的响应格式。您可以针对 RHOSP 云部署运行 **场景测试** 和 **API 测试**。

## 场景测试

场景测试模拟典型的最终用户操作工作流，以测试服务之间的集成点。测试框架执行配置，测试服务之间的集成，然后自动删除。使用与其相关的服务标记测试，这些服务与测试所使用的客户端库相关。

以下情况基于用例：

- 将镜像上传到镜像服务
- 从镜像部署实例
- 将卷附加到实例
- 创建实例的快照
- 将卷从实例分离

## API 测试

API 测试验证 OpenStack API。测试使用 OpenStack API 的 OpenStack Integration Test Suite 实施。您可以使用有效和无效的 JSON 来确保错误响应有效。您可以独立运行测试，您不必依赖之前的测试状态。

## 第 2 章 安装 INTEGRATION TEST SUITE (TEMPEST)

要手动安装 Integration Test Suite, 请[手动安装 Integration Test Suite](#)。

### 2.1. 先决条件

- undercloud 安装。如需更多信息, 请[参阅安装 undercloud](#)。
- overcloud 部署。如需更多信息, 请[参阅使用 CLI 工具创建基本 overcloud](#)。

### 2.2. 手动安装 INTEGRATION TEST SUITE

如果您不想使用 director 自动安装 Integration Test Suite (tempest), 您可以稍后手动执行安装。您必须确保您有基本的网络配置, 安装 Integration Test Suite 软件包, 并创建一个包含 OpenStack 服务和其他测试的详情的配置文件。

#### 流程

1. 确保 Red Hat OpenStack Platform (RHOSP)环境中提供了以下网络：

- 提供浮动 IP 的外部网络。
- 专用网络。  
通过路由器连接这些网络。
  - a. 要创建专用网络, 请根据您的网络部署指定以下选项：

```
$ openstack network create <network_name> --share
$ openstack subnet create <subnet_name> --subnet-range <address/prefix> \
--network <network_name>
$ openstack router create <router_name>
$ openstack router add subnet <router_name> <subnet_name>
```

b. 要创建公共网络, 请根据您的网络部署指定以下选项：

```
$ openstack network create <network_name> --external \
--provider-network-type flat \
--provider-physical-network datacentre
$ openstack subnet create <subnet_name> --subnet-range <address/prefix> \
--gateway <default_gateway> --no-dhcp --network <network_name>
$ openstack router set <router_name> --external-gateway <public_network_name>
```

2. 安装与 Integration Test Suite 相关的软件包：

```
$ sudo dnf -y install openstack-tempest
```

此命令不安装任何 tempest 插件。您必须根据 RHOSP 安装手动安装插件。

3. 为您的环境中的每个组件安装适当的 tempest 插件。例如, 输入以下命令安装 keystone、neutron、cinder 和 telemetry 插件：

```
$ sudo dnf install python3-keystone-tests-tempest python3-neutron-tests-tempest python3-
cinder-tests-tempest python3-telemetry-tests-tempest
```

有关软件包的完整列表，请参阅 [Integration Test Suite 软件包](#)。



### 注意

您还可以安装 **openstack-tempest-all** 软件包。此软件包包含所有 tempest 插件。

## 2.2.1. Integration Test Suite 软件包

使用 **dnf search** 检索 tempest 测试软件包列表：

```
$ sudo dnf search $(openstack service list -c Name -f value) 2>/dev/null | grep test | awk '{print $1}'
```

组件	软件包名称
barbican	python3-barbican-tests-tempest
cinder	python3-cinder-tests-tempest
designate	python3-designate-tests-tempest
ec2-api	python3-ec2api-tests-tempest
Heat	python3-heat-tests-tempest
ironic	python3-ironic-tests-tempest
Keystone	python3-keystone-tests-tempest
Kuryr	python3-kuryr-tests-tempest
manila	python3-manila-tests-tempest
mistral	python3-mistral-tests-tempest
networking-bgpvpn	python3-networking-bgpvpn-tests-tempest
networking-l2gw	python3-networking-l2gw-tests-tempest
neutron	python3-neutron-tests-tempest
nova-join	python3-novajoin-tests-tempest
octavia	python3-octavia-tests-tempest
patrole	python3-patrole-tests-tempest
Telemetry	python3-telemetry-tests-tempest

组件	软件包名称
tripleo-common	python3-tripleo-common-tests-tempest
zaqar	python3-zaqar-tests-tempest



### 注意

**python3-telemetry-tests-tempest** 软件包包含 aodh、panko、gnocchi 和 ceilometer 测试的插件。**python3-ironic-tests-tempest** 软件包包含 ironic 和 ironic-inspector 的插件。

## 第 3 章 配置集成测试套件(TEMPEST)

在开始使用 Integration Test Suite 验证环境前，您必须创建一个工作区并生成 `/etc/tempest.conf` 配置文件。

### 3.1. 先决条件

- 包含 Integration Test Suite 软件包的 OpenStack 环境。

### 3.2. 创建工作区

为您的 Integration Test Suite (tempest)配置和输出创建一个工作区。

#### 流程

1. 提供目标部署的凭证：

- 如果目标在 undercloud 中，提供 undercloud 的凭证：

```
# source stackrc
```

- 如果目标在 overcloud 中，提供 overcloud 的凭证：

```
# source overcloudrc
```

2. 初始化 **tempest**：

```
# tempest init mytempest
# cd mytempest
```

此命令创建一个名为 **mytempest** 的临时工作区。

3. 可选：输入以下命令查看现有工作区列表：

```
# tempest workspace list
```

4. 生成 **etc/tempest.conf** 文件：

```
# discover-tempest-config --deployer-input ~/tempest-deployer-input.conf \
--debug --create --network-id <UUID>
```

使用外部网络的 **UUID** 替换 UUID。

**discover-tempest-config** 以前称为 **config\_tempest.py**，并使用相同的参数。**python-tempestconf** 作为 **openstack-tempest** 的依赖项，提供 **discover-tempest-config**。



#### 注意

要为 undercloud 生成 **etc/tempest.conf** 文件，请确保 **tempest-deployer-input.conf** 文件中的区域名称与 undercloud 部署中的名称相同。如果这些名称不匹配，请更新 **tempest-deployer-input.conf** 文件中的区域名称，以匹配 undercloud 的区域名称。

- 要检查 undercloud 的区域名称，请输入以下命令：

```
$ source stackrc
$ openstack region list
```

- 要检查 overcloud 的区域名称，请输入以下命令：

```
$ source overcloudrc
$ openstack region list
```

您可能需要修改默认的 **tempest.conf** 文件以适合您的环境。如需更多信息，[请参阅配置扩展列表和配置 heat\\_plugin。](#)

## 验证

- 验证您当前的 tempest 配置：

```
# tempest verify-config -o <output>
```

输出的值是 Integration Test Suite 写入更新的配置的输出文件。这与原始配置文件不同。

## 3.3. 手动配置 INTEGRATION TEST SUITE

**discover-tempest-config** 命令会自动生成 **tempest.conf** 文件。但是，您必须确保 **tempest.conf** 文件与环境的配置对应。

### 3.3.1. 手动配置 Integration Test Suite 扩展列表

默认 **tempest.conf** 文件包含每个组件的扩展列表。检查 **tempest.conf** 文件中每个组件的 **api\_extensions** 属性，并验证扩展列表是否与您的部署对应。

如果您的部署中提供的扩展与 **tempest.conf** 文件的 **api\_extensions** 属性中的扩展列表不匹配，则组件会失败。要防止此失败，您必须识别部署中可用的扩展，并将其包含在 **api\_extensions** 参数中。要获取部署中的网络、计算、卷或身份扩展列表，请运行以下命令：

## 流程

- 要在部署中检索网络、计算、卷或身份扩展列表，请输入以下命令：

```
$ openstack extension list [--network] [--compute] [--volume] [--identity]
```

### 3.3.2. 手动配置 heat\_plugin

您可以在 **tempest.conf** 文件中手动配置 **heat\_plugin**。

## 流程

- 使用以下示例配置 **heat\_plugin**：

```
[service_available]
heat = True
```

```
[heat_plugin]
username = demo
password = ***
project_name = demo
admin_username = admin
admin_password = ****
admin_project_name = admin
auth_url = http://10.0.0.110:5000/v3
auth_version = 3
user_domain_id = default
project_domain_id = default
user_domain_name = Default
project_domain_name = Default
region = regionOne
fixed_network_name = demo_project_network
network_for_ssh = public
floating_network_name = nova
instance_type = m1.nano
minimal_instance_type = m1.micro
image_ref = 7faed41e-a56c-4971-bf48-24e4e23e69a5
minimal_image_ref = 7faed41e-a56c-4971-bf48-24e4e23e69a5
```

使用 `openstack network list` 命令识别 `fixed_network_name`、`network_for_ssh` 和 `floating_network_name` 参数的网络。



### 注意

您必须在 `tempest.conf` 文件的 `[service_available]` 部分中将 `heat` 设置为 `True`，`[heat_plugin]` 部分的 `username` 属性中的用户必须具有角色 `成员`。例如，输入以下命令将 `member` 角色添加到 `demo` 用户：

```
$ openstack role add --user demo --project demo member
```

## 3.4. 配置 INTEGRATION TEST SUITE 日志

您可以在 `tempest` 工作区中的 `logs` 目录中更改日志文件的默认位置。

### 流程

1. 在 `tempest.conf` 中，在 `[DEFAULT]` 部分下，将 `log_dir` 设置为所需的目录：

```
[DEFAULT]
log_dir = <directory>
```

2. 如果您有自己的日志配置文件，请在 `[DEFAULT]` 部分下的 `tempest.conf` 中，将 `log_config_append` 设置为您的文件：

```
[DEFAULT]
log_config_append = <file>
```

如果您设置了 `log_config_append` 属性，则 Integration Test 套件会忽略 `tempest.conf` 中的所有其他日志记录配置，包括 `log_dir` 属性。

### 3.5. 配置 INTEGRATION TEST SUITE MICROVERSION 测试

Integration Test Suite (tempest) 提供稳定的接口来测试 API 微版本。要使用这些接口实施微版本测试，请完成以下步骤。

#### 流程

1. 在 **tempest.conf** 配置文件中配置选项，以指定目标微版本。配置这些选项，以确保支持的微版本与 OpenStack 云中的微版本对应。
2. 您可以指定一系列目标微版本，以在单个 Integration Test Suite 操作中运行多个微版本测试。例如，若要限制 **compute** 服务的微版本范围，在配置文件的 **[compute]** 部分中，将值分配给 **min\_microversion** 和 **max\_microversion** 参数：

```
[compute]
min_microversion = 2.14
max_microversion = latest
```



## 第 4 章 清理 INTEGRATION TEST SUITE (TEMPEST)资源

在使用 OpenStack Integration Test Suite (tempest)验证部署之前，请使用 `--init-saved-state` 标志运行 `cleanup` 命令。此命令扫描您的环境以发现资源，如网络、卷、镜像、类别、项目和用户。发现的资源保存在名为 `saved_state.json` 的文件中。当执行 `tempest cleanup` 命令时，未记录在 `saved_state.json` 文件中的所有资源会被删除。

### 先决条件

- 包含 Integration Test Suite 软件包的 OpenStack 环境。
- 与您的 OpenStack 环境对应的 Integration Test Suite 配置。如需更多信息，[请参阅创建工作区](#)。
- 一个或多个已完成的 Integration Test Suite 验证测试。

### 4.1. 执行空运行

执行清理前执行空运行。空运行列出了 Integration Test Suite 将通过清理删除的文件，而无需实际删除任何文件。`dry_run.json` 文件包含清理删除的文件列表。

#### 流程

1. 完成空运行：

```
# tempest cleanup --dry-run
```

2. 检查 `dry_run.json` 文件，以确保清理不会删除您的环境所需的任何文件。

### 4.2. 执行 TEMPEST 清理

在运行任何 `tempest` 测试前，您必须初始化保存的状态。这将创建 `saved_state.json` 文件，该文件可防止清理删除必须保留的对象。



#### 警告

如果您没有使用 `--init-saved-state` 标志运行 `cleanup` 命令，RHOSP 对象会被删除。

如果您使用 `--init-saved-state` 运行 `cleanup` 命令后创建对象，则后续 `tempest` 命令可以删除这些对象。

#### 流程

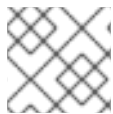
1. 初始化保存的状态以创建 `saved_state.json` 文件：

```
# tempest cleanup --init-saved-state
```

2. 执行清理：

```
# tempest cleanup
```

**tempest cleanup** 命令会删除 tempest 资源，但不删除项目或 tempest 管理员帐户。



### 注意

您可以修改 **saved\_state.json** 文件，使其包含或排除您要保留或删除的对象。

## 第 5 章 使用 INTEGRATION TEST SUITE (TEMPEST)验证 OPENSTACK 云

您可以使用 **tempest run** 命令以多种方式运行 Integration Test Suite 验证。您还可以组合单个 **tempest run** 命令中的多个选项。

### 5.1. 先决条件

- 包含 Integration Test Suite 软件包的 OpenStack 环境。
- 与您的 OpenStack 环境对应的 Integration Test Suite 配置。如需更多信息，[请参阅创建工作区](#)。

### 5.2. 列出可用的测试

使用 **--list-tests** 选项列出所有可用的测试。

#### 流程

- 输入带有 **--list-tests** 或 **-l** 选项的 **tempest run** 命令，以获取可用 tempest 测试列表：

```
# tempest run -l
```

### 5.3. 运行烟雾测试

烟雾测试是初始测试的一种类型，仅涵盖最重要的功能。虽然这些测试不全面，但运行烟雾测试可以在找出问题时节省时间。

#### 流程

- 使用 **--smoke** 选项输入 **tempest run** 命令：

```
# tempest run --smoke
```

### 5.4. 使用允许列表文件传递测试

allowlist 文件是一个文件，其中包含用于选择您要包含的测试的正则表达式。如果您使用一个或多个正则表达式，请在单独的行中指定每个表达式。

#### 流程

- 输入带有 **--whitelist-file** 或 **-w** 选项的 **tempest run** 命令，以使用允许列表文件：

```
# tempest run -w <whitelist_file>
```

### 5.5. 使用 BLOCKLIST 文件跳过测试

blocklist 文件是一个文件，其中包含用于选择您要排除的测试的正则表达式。如果您使用一个或多个正则表达式，请在单独的行中指定每个表达式。

流程

## 流程

- 输入带有 **--blacklist-file** 或 **-b** 选项的 **tempest run** 命令，以使用 blacklist 文件：

```
# tempest run -b <blacklist_file>
```

## 5.6. 在并行或系列中运行测试

您可以并行运行测试，也可以按系列运行测试。您还可以定义在运行并行测试时要使用的 worker 数量。默认情况下，Integration Test Suite 为每个可用 CPU 使用一个 worker。

选择按顺序或并行运行测试：

- 按顺序运行测试：

```
# tempest run --serial
```

- 并行运行测试（默认）：

```
# tempest run --parallel
```

- 使用 **--concurrency** 或 **-c** 选项指定并行运行测试时要使用的 worker 数量：

```
# tempest run --concurrency <workers>
```

## 5.7. 运行特定的测试

使用 **--regex** 选项运行特定的测试。正则表达式必须是 Python 正则表达式：

### 流程

- 使用以下命令：

```
# tempest run --regex <regex>
```

- 例如，使用以下示例命令运行具有以 **tempest.scenario** 开头的所有测试：

```
# tempest run --regex ^tempest.scenario
```

## 5.8. 删除 INTEGRATION TEST SUITE 对象

输入 **tempest cleanup** 命令，以删除所有 Integration Test Suite (tempest) 资源。这个命令还会删除项目，但命令不会删除管理员帐户：

### 流程

- 删除 tempest 资源：

```
# tempest cleanup --delete-tempest-conf-objects
```

