



Red Hat OpenStack Platform 16.2

OpenStack Integration Test Suite 指南

OpenStack Integration Test Suite 简介

Red Hat OpenStack Platform 16.2 OpenStack Integration Test Suite 指南

OpenStack Integration Test Suite 简介

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[®] 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

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. 使用 DIRECTOR 安装 INTEGRATION TEST SUITE	6
2.3. 手动安装 INTEGRATION TEST SUITE	6
第 3 章 配置 INTEGRATION TEST SUITE(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 日志记录	11
3.5. 配置集成测试套件微版本测试	12
第 4 章 清理 INTEGRATION TEST SUITE(TEMPEST)资源	13
4.1. 执行空运行	13
4.2. 执行临时清理	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 ALLOWLIST 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)包含许多不同的项目，所以务必要测试 RHOSP 集群中项目的互操作性。OpenStack Integration Test Suite 自动对 RHOSP 部署进行集成测试。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 包含对 OpenStack API 验证和场景测试的测试，以及自我验证的单元测试。Integration Test Suite 通过使用 OpenStack 公共 API 执行黑项测试，并将 tempest 作为测试运行。

OpenStack Integration Test Suite(tempest)充当向 Red Hat OpenStack Platform(RHOSP)核心项目的提交的最低要求，它可以压力测试来生成云部署的负载，并且可以执行 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。为测试标记相关的服务，以阐明测试所使用的客户端库。

以下场景基于用例：

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

API 测试

API 测试验证 OpenStack API。测试使用 OpenStack API 的 OpenStack Integration Test Suite 实现。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，且您不必依赖于之前的测试状态。

第 2 章 安装 INTEGRATION TEST SUITE(TEMPEST)

您可以使用 `director` 或手动安装 Integration Test Suite。

- 要使用 `director` 安装 Integration Test Suite，请参阅使用 `director` [安装 Integration Test Suite](#)。
- 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. 使用 DIRECTOR 安装 INTEGRATION TEST SUITE

使用 Red Hat OpenStack Platform(RHOSP)`director` 自动安装测试套件。

先决条件

- 已安装 `python3-tripleoclient` 软件包。如需更多信息，请参阅 Director 安装和使用指南中的 [安装 director 软件包](#)。

流程

1. 以 `stack` 用户身份登录 `undercloud` 主机。
2. 编辑位于 `stack` 用户的主目录中的 `undercloud.conf` 文件。
3. 将 `enable_tempest` 参数设置为 `true`。

```
enable_tempest = true
```

4. 运行 `openstack undercloud install` 命令，在 `undercloud` 中包含额外配置：

```
$ openstack undercloud install
```

2.3. 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。您必须确保具有基本网络配置，安装 Integration Test Suite 软件包，并创建一个配置文件，其中包含有关 OpenStack 服务和其他测试交换机的详细信息。

流程

1. 确保 Red Hat OpenStack Platform(RHOSP)环境中有以下网络：
 - 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：

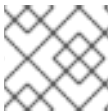
```
$ sudo dnf -y install openstack-tempest
```

此命令不会安装任何 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，输入以下命令来安装 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。这个软件包包含所有 tempest 插件。

2.3.1. 集成测试套件软件包

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

组件	软件包名称
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
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 INTEGRATION TEST SUITE(TEMPEST)

在使用 Integration Test Suite 验证您的环境前，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。如需更多信息，[请参阅使用 director 安装 Integration Test Suite。](#)

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 的凭据：

```
# source stackrc
```

- 如果目标位于 overcloud 中，则提供 overcloud 的凭据：

```
# 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** 是 **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 写入您更新的配置的输出文件。这与您的原始配置文件不同。

3.3. 手动配置 INTEGRATION TEST SUITE

discover-tempest-config 命令自动生成 **tempest.conf** 文件。但是，您必须确保 **tempest.conf** 文件对应于您的环境配置。

3.3.1. 手动配置 Integration Test Suite 扩展列表

默认的 **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，并验证扩展列表与您的部署对应。

如果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 tempest 测试。要防止此失败，您必须识别部署中存在的扩展，并将它们包含在 **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

- 根据您的部署，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`，并且 `[heat_plugin]` 部分的 `username` 属性中的用户必须具有角色 `member`。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 日志记录

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. 如果您的日志文件有自己的日志记录配置文件，在 `[DEFAULT]` 部分下的 `tempest.conf` 中，请将 `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，Integration Test Suite 会忽略 `tempest.conf` 中的所有其他日志配置，包括 `log_dir` 属性。

3.5. 配置集成测试套件微版本测试

Integration Test Suite(tempest)提供了稳定的接口来测试 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来指定目标微版本。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)验证部署前，请先使用 `--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 `tempest cleanup` 命令执行没有在 `saved_state.json` 文件中记录的所有资源时。

先决条件

- 包含 Integration Test Suite 软件包的 OpenStack 环境。如需更多信息，[请参阅使用 director 安装 Integration Test Suite](#)。
- 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](#)。
- 一个或多个完成的 Integration Test Suite 验证测试。

4.1. 执行空运行

执行清理前的空运行。空运行列出了 Integration Test Suite 将按清理删除的文件，而不实际删除任何文件。`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所有文件。

4.2. 执行临时清理

在运行任何 `tempest` 测试前，您必须初始化保存的状态。这会创建文件 `saved_state.json`，这样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，则 RHOSP 对象会被删除。

如果在运行带有 `--init-saved-state` 的 `cleanup` 命令后创建对象，则后续 `tempest` 命令可以删除这些对象。

流程

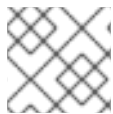
1. 初始化保存的状态以创建 `saved_state.json` 文件：

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

2. 执行清理：

```
# tempest cleanup
```

tempest cleanup 命令删除 tempest 资源，但不会删除项目或临时管理员帐户。



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。您还可以在单个 **tempest run** 命令中组合多个选项。

5.1. 先决条件

- 包含 Integration Test Suite 软件包的 OpenStack 环境。如需更多信息，[请参阅使用 director 安装 Integration Test Suite。](#)
- 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

- 使用 **--list-tests** 或 **-l** 选项输入 **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

烟雾测试是一些初步的测试，仅包含最重要的功能。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 ALLOWLIST 文件传递测试

allowlist 文件是一个包含正则表达式的文件，用于选择您要包含的测试。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

- 使用 **--whitelist-file** 或 **-w** 选项输入 **tempest run** 命令以使用 allowlist 文件：

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

5.5. 使用 BLOCKLIST 文件跳过测试

blocklist 文件是一个包含正则表达式的文件，用于选择您要排除的测试。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

- 使用 **--blacklist-file** 或 **-b** 选项输入 **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

您可以并行或系列运行测试。您还可以定义在运行并行测试时使用的 worker 数量。默认情况下，Integration Test Suite 会为每个可用的 CPU 使用一个 worker。

选择以串行方式或并行运行测试：

- 以串行方式运行测试：

```
# 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使用的 worker 数量：

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

5.7. 运行特定测试

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名称以 **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** 命令来删除所有 Integration Test Suite(tempest)资源。这个命令还会删除项目，但该命令不会删除管理员帐户：

流程

- 删除 tempest 资源：

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

