



Red Hat OpenStack Platform 17.1

Red Hat OpenStack Platform Integration Test Suite로 클라우드 검증

Red Hat OpenStack Platform 배포 검증

Red Hat OpenStack Platform 17.1 Red Hat OpenStack Platform Integration Test Suite로 클라우드 검증

Red Hat OpenStack Platform 배포 검증

OpenStack Team
rhos-docs@redhat.com

법적 공지

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.

초록

배포를 검증할 수 있도록 Red Hat OpenStack Platform 환경에서 OpenStack Integration Test Suite(tempest)를 설치, 구성 및 관리합니다.

차례	
보다 포괄적 수용을 위한 오픈 소스 용어 교체	3
RED HAT 문서에 관한 피드백 제공	4
1장. OPENSTACK INTEGRATION TEST SUITE(TEMPEST) 검증	5
2장. INTEGRATION TEST SUITE 설치(TEMPEST)	6
2.1. 사전 요구 사항	6
2.2. 수동으로 INTEGRATION TEST SUITE 설치	6
3장. INTEGRATION TEST SUITE 구성 (TEMPEST)	9
3.1. 사전 요구 사항	9
3.2. 작업 공간 생성	9
3.3. 수동으로 INTEGRATION TEST SUITE 구성	10
3.4. INTEGRATION TEST SUITE 로깅 구성	11
3.5. INTEGRATION TEST SUITE 마이크로 버전 테스트 구성	12
4장. INTEGRATION TEST SUITE(TEMPEST) 리소스 정리	13
4.1. 시험 실행 수행	13
4.2. 임시 정리 수행	13
5장. INTEGRATION TEST SUITE를 사용하여 OPENSTACK 클라우드 검증(TEMPEST)	15
5.1. 사전 요구 사항	15
5.2. 사용 가능한 테스트 나열	15
5.3. 스모크 테스트 실행	15
5.4. 허용 목록 파일을 사용하여 테스트 전달	15
5.5. BLOCKLIST 파일을 사용하여 테스트 건너뛰기	15
5.6. 병렬로 또는 직렬로 테스트 실행	16
5.7. 특정 테스트 실행	16
5.8. INTEGRATION TEST SUITE 오브젝트 삭제	16

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 [CTO Chris Wright의 메시지](#)를 참조하십시오.

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 어떻게 더 나은지 알려주십시오.

Jira에서 문서 피드백 제공

[Create Issue](#) 양식을 사용하여 OpenShift (RHOSO) 또는 이전 Red Hat OpenStack Platform (RHOSP)의 Red Hat OpenStack Services 문서에 대한 피드백을 제공합니다. RHOSO 또는 RHOSP 문서에 대한 문제를 생성할 때 RHOSO Jira 프로젝트에 문제가 기록되어 피드백의 진행 상황을 추적할 수 있습니다.

[문제 생성](#) 양식을 완료하려면 Jira에 로그인해야 합니다. Red Hat Jira 계정이 없는 경우 <https://issues.redhat.com> 에서 계정을 생성할 수 있습니다.

1. 다음 링크를 클릭하여 **문제 생성** 페이지를 엽니다.
<https://issues.redhat.com/secure/CreateInfoDetails!init.jspx?pid=12336920&summary=Documentation%20feedback:%20%3CAdd%20summary%20here%3E&i<Include+the+documentation+URL,+the%20chapter+or+section+number,+and+a+detailed+descrip>
2. **요약** 및 **설명** 필드를 작성합니다. **설명** 필드에 문서 URL, 장 또는 섹션 번호, 문제에 대한 자세한 설명을 포함합니다. 양식의 다른 필드를 수정하지 마십시오.
3. **생성**을 클릭합니다.

1장. OPENSTACK INTEGRATION TEST SUITE(TEMPEST) 검증

RHOSP(Red Hat OpenStack Platform)는 다양한 프로젝트로 구성되므로 RHOSP 클러스터 내에서 프로젝트의 상호 운용성을 테스트하는 것이 중요합니다. OpenStack Integration Test Suite는 RHOSP 배포의 통합 테스트를 자동화합니다. 테스트를 실행하여 클러스터가 예상대로 작동하는지 확인할 수 있습니다. 테스트 출력은 특히 업그레이드 후 잠재적인 문제에 대한 조기 경고를 제공합니다.

Integration Test Suite에는 OpenStack API 검증 및 시나리오 테스트와 자체 유효성 검사에 대한 단위 테스트가 포함되어 있습니다. Integration Test Suite는 테스트 실행자로서 Tempest를 사용하여 OpenStack 공용 API를 사용하여 검정 상자 테스트를 수행합니다.

OpenStack Integration Test Suite(tempest)는 RHOSP(Red Hat OpenStack Platform) 코어 프로젝트에 대한 커밋을 위한 게이트 역할을 하며, 테스트로 클라우드 배포에 대한 부하를 생성할 수 있으며 CLI 테스트를 수행하여 명령줄의 응답 형식을 확인할 수 있습니다. RHOSP 클라우드 배포에 대해 **시나리오 테스트** 및 **API 테스트**를 실행할 수 있습니다.

시나리오 테스트

시나리오 테스트는 일반적인 최종 사용자 작업 워크플로를 시뮬레이션하여 서비스 간 통합 지점을 테스트합니다. 테스트 프레임워크는 구성을 수행하고 서비스 간 통합을 테스트한 다음 자동으로 제거됩니다. 테스트에 관련된 서비스에 태그를 지정하여 테스트에서 사용하는 클라이언트 라이브러리를 명확히 지정합니다.

다음 시나리오는 사용 사례를 기반으로 합니다.

- 이미지 서비스에 이미지 업로드
- 이미지에서 인스턴스 배포
- 인스턴스에 볼륨 연결
- 인스턴스의 스냅샷 생성
- 인스턴스에서 볼륨 분리

API 테스트

API 테스트에서는 OpenStack API를 검증합니다. 테스트에서는 OpenStack API의 OpenStack Integration Test Suite 구현을 사용합니다. 유효한 JSON과 유효하지 않은 JSON을 사용하여 오류 응답이 유효한지 확인할 수 있습니다. 테스트를 독립적으로 실행할 수 있으며 이전 테스트 상태에 의존할 필요가 없습니다.

2장. INTEGRATION TEST SUITE 설치(TEMPEST)

Integration Test Suite를 수동으로 설치하려면 [Integration Test Suite](#)를 수동으로 설치합니다.

2.1. 사전 요구 사항

- 언더클라우드 설치입니다. 자세한 내용은 [언더클라우드 설치](#)를 참조하십시오.
- 오버클라우드 배포입니다. 자세한 내용은 [CLI 툴을 사용하여 기본 오버클라우드 생성](#)을 참조하십시오.

2.2. 수동으로 INTEGRATION TEST SUITE 설치

director와 함께 Integration Test Suite(tempest)를 자동으로 설치하지 않으려면 나중에 수동으로 설치를 수행할 수 있습니다. 기본 네트워크 구성이 있는지 확인하고 Integration Test Suite 패키지를 설치하고, OpenStack 서비스 및 기타 테스트 동작 스위치에 대한 세부 정보가 포함된 구성 파일을 생성해야 합니다.

절차

1. RHOSP(Red Hat OpenStack Platform) 환경에서 다음 네트워크를 사용할 수 있는지 확인합니다.

- 유동 IP를 제공할 수 있는 외부 네트워크입니다.
- 사설 네트워크입니다.
라우터를 통해 이러한 네트워크를 연결합니다.
 - a. 사설 네트워크를 생성하려면 네트워크 배포에 따라 다음 옵션을 지정합니다.

```
$ 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. 공용 네트워크를 생성하려면 네트워크 배포에 따라 다음 옵션을 지정합니다.

```
$ 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. Integration Test Suite와 관련된 패키지를 설치합니다.

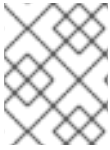
```
$ sudo dnf -y install openstack-tempest
```

이 명령은 템플릿 플러그인을 설치하지 않습니다. RHOSP 설치에 따라 플러그인을 수동으로 설치해야 합니다.

3. 환경의 각 구성 요소에 대해 적절한 tempest 플러그인을 설치합니다. 예를 들어 keystone, neutron, cinder 및 telemetry 플러그인을 설치하려면 다음 명령을 입력합니다.

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

전체 패키지 목록은 [Integration Test Suite 패키지](#)를 참조하십시오.



참고

openstack-tempest-all 패키지를 설치할 수도 있습니다. 이 패키지에는 모든 tempest 플러그인이 포함되어 있습니다.

2.2.1. 통합 테스트 모음 패키지

dnf 검색을 사용하여 tempest 테스트 패키지 목록을 검색합니다.

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

구성 요소	패키지 이름
Barbican	python3-barbican-tests-tempest
cinder	python3-cinder-tests-tempest
지정	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

구성 요소	패키지 이름
Evenrole	python3-patrole-tests-tempest
telemetry	python3-telemetry-tests-tempest
tripleo-common	python3-tripleo-common-tests-tempest
Zaqar	python3-zaqar-tests-tempest



참고

python3-telemetry-tests-tempest 패키지에는 aodh, panko, gnocchi 및 ceilometer 테스트용 플러그인이 포함되어 있습니다. **python3-ironic-tests-tempest** 패키지에는 ironic 및 ironic-inspector용 플러그인이 포함되어 있습니다.

3장. INTEGRATION TEST SUITE 구성 (TEMPEST)

Integration Test Suite로 환경 검증을 시작하기 전에 작업 영역을 생성하고 `/etc/tempest.conf` 구성 파일을 생성해야 합니다.

3.1. 사전 요구 사항

- Integration Test Suite 패키지가 포함된 OpenStack 환경.

3.2. 작업 공간 생성

Integration Test Suite(tempest) 구성 및 출력에 대한 작업 공간을 생성합니다.

절차

1. 대상 배포의 인증 정보를 가져옵니다.

- 대상이 언더클라우드에 있는 경우 언더클라우드의 인증 정보를 가져옵니다.

```
# source stackrc
```

- 대상이 오버클라우드에 있는 경우 오버클라우드의 인증 정보를 가져옵니다.

```
# source overcloudrc
```

2. **Tempest** 초기화:

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

이 명령은 **mytempest** 라는 임시 작업 공간을 생성합니다.

3. 선택 사항: 기존 작업 공간 목록을 보려면 다음 명령을 입력합니다.

```
# tempest workspace list
```

4. **etc/tempest.conf** 파일을 생성합니다.

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

UUID 를 외부 네트워크의 UUID로 바꿉니다.

discover-tempest-config 는 이전에 **config_tempest.py** 라고 하며 동일한 매개변수를 사용합니다. **python-tempestconf** 는 **openstack-tempest** 의 종속성으로, **discover-tempest-config** 를 제공합니다.



참고

언더클라우드의 **etc/tempest.conf** 파일을 생성하려면 **tempest-deployer-input.conf** 파일의 지역 이름이 언더클라우드 배포의 이름과 같은지 확인합니다. 이러한 이름이 일치하지 않으면 언더클라우드의 지역 이름과 일치하도록 **tempest-deployer-input.conf** 파일의 리전 이름을 업데이트합니다.

5. 언더클라우드의 지역 이름을 검사하려면 다음 명령을 입력합니다.

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

6. 오버클라우드의 지역 이름을 검사하려면 다음 명령을 입력합니다.

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

환경에 맞게 기본 **tempest.conf** 파일을 수정해야 할 수 있습니다. 자세한 내용은 [확장 목록 구성 및 heat_plugin 구성](#)을 참조하십시오.

검증

- 현재 임시 구성을 확인합니다.

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

출력 값은 Integration Test Suite가 업데이트된 구성을 작성하는 출력 파일입니다. 이는 원래 구성 파일과 다릅니다.

3.3. 수동으로 INTEGRATION TEST SUITE 구성

discover-tempest-config 명령은 **tempest.conf** 파일을 자동으로 생성합니다. 그러나 **tempest.conf** 파일이 환경 구성에 해당하는지 확인해야 합니다.

3.3.1. 수동으로 Integration Test Suite 확장 목록 구성

기본 **tempest.conf** 파일에는 각 구성 요소에 대한 확장 목록이 포함되어 있습니다. **tempest.conf** 파일의 각 구성 요소에 대한 **api_extensions** 속성을 검사하고 확장 목록이 배포에 해당하는지 확인합니다.

배포에서 사용할 수 있는 확장이 **tempest.conf** 파일의 **api_extensions** 속성의 확장 프로그램에 해당하지 않으면 구성 요소가 Tempest 테스트에 실패합니다. 이 실패를 방지하려면 배포에서 사용 가능한 확장을 식별하고 **api_extensions** 매개변수에 포함해야 합니다. 배포의 네트워크, 컴퓨팅, 볼륨 또는 ID 확장 목록을 가져오려면 다음 명령을 실행합니다.

절차

- 배포에서 네트워크, 컴퓨팅, 볼륨 또는 ID 확장 목록을 검색하려면 다음 명령을 입력합니다.

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

3.3.2. heat_plugin 수동 구성

tempest.conf 파일에서 **heat_plugin** 을 수동으로 구성할 수 있습니다.

절차

- 배포에 따라 **heat_plugin** 을 구성하려면 다음 예제를 사용합니다.

```
[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

```

openstack network list 명령을 사용하여 **fixed_network_name**, **network_for_ssh** 및 **floating_network_name** 매개변수의 네트워크를 식별합니다.



참고

tempest.conf 파일의 **[service_available]** 섹션에서 **heat** 를 **True** 로 설정해야 하며, **[heat_plugin]** 섹션의 **username** 속성에 있는 사용자에게 역할 **멤버**가 있어야 합니다. 예를 들어 다음 명령을 입력하여 **멤버** 역할을 **demo** 사용자에게 추가합니다.

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

3.4. INTEGRATION TEST SUITE 로깅 구성

tempest 작업 공간 내의 로그 디렉터리에 있는 로그 파일의 기본 위치를 변경할 수 있습니다.

절차

1. **tempest.conf** 의 **[DEFAULT]** 섹션에서 **log_dir** 을 원하는 디렉터리로 설정합니다.

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

2. 자체 로깅 구성 파일이 있는 경우 **tempest.conf** 에서 **[DEFAULT]** 섹션 아래에 **log_config_append** 를 파일로 설정합니다.

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

log_config_append 속성을 설정하면 Integration Test Suite에서 **log_dir** 특성을 포함하여 **tempest.conf**의 다른 모든 로깅 구성을 무시합니다.

3.5. INTEGRATION TEST SUITE 마이크로 버전 테스트 구성

Integration Test Suite(tempest)는 API 마이크로버전을 테스트하기 위한 안정적인 인터페이스를 제공합니다. 이러한 인터페이스를 사용하여 마이크로 버전 테스트를 구현하려면 다음 단계를 완료합니다.

절차

1. **tempest.conf** 구성 파일에서 옵션을 구성하여 대상 마이크로버전을 지정합니다. 지원되는 마이크로버전이 OpenStack 클라우드의 마이크로버전에 해당하는지 확인하도록 이러한 옵션을 구성합니다.
2. 대상 마이크로버전 범위를 지정하여 단일 Integration Test Suite 작업에서 여러 마이크로 버전 테스트를 실행할 수 있습니다.
예를 들어 컴퓨팅 서비스의 마이크로 버전 범위를 제한하려면 구성 파일의 **[compute]** 섹션에서 **min_microversion** 및 **max_microversion** 매개변수에 값을 할당합니다.

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


4장. INTEGRATION TEST SUITE(TEMPEST) 리소스 정리

OpenStack Integration Test Suite(tempest)를 사용하여 배포를 확인하기 전에 **--init-saved-state** 플래그를 사용하여 **cleanup** 명령을 실행합니다. 이 명령은 환경을 검사하여 네트워크, 볼륨, 이미지, 플레이버, 프로젝트, 사용자 등 리소스를 검색합니다. 검색된 리소스는 **saved_state.json** 이라는 파일에 저장됩니다. **tempest cleanup** 명령이 실행되면 **saved_state.json** 파일에 기록되지 않은 모든 리소스가 삭제됩니다.

사전 요구 사항

- Integration Test Suite 패키지가 포함된 OpenStack 환경.
- OpenStack 환경에 해당하는 Integration Test Suite 구성입니다. 자세한 내용은 [작업 공간 생성을](#) 참조하십시오.
- 하나 이상의 완료된 Integration Test Suite 검증 테스트

4.1. 시험 실행 수행

정리를 실행하기 전에 예행 실행을 수행합니다. 예행 실행에서는 실제로 파일을 삭제하지 않고 Integration Test Suite가 정리에서 삭제할 파일을 나열합니다. **dry_run.json** 파일에는 정리가 삭제하는 파일 목록이 포함되어 있습니다.

절차

1. 예행 실행을 완료합니다.

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

2. **dry_run.json** 파일을 검토하여 정리에서 환경에 필요한 파일을 삭제하지 않는지 확인합니다.

4.2. 임시 정리 수행

Tempest 테스트를 실행하기 전에 저장된 상태를 초기화해야 합니다. 이렇게 하면 **saved_state.json** 파일이 생성되어 정리에서 유지해야 하는 오브젝트를 삭제하지 않습니다.



주의

--init-saved-state 플래그를 사용하여 **cleanup** 명령을 실행하지 않으면 RHOSP 오브젝트가 삭제됩니다.

--init-saved-state 를 사용하여 **cleanup** 명령을 실행한 후 오브젝트를 생성하면 후속 **tempest** 명령으로 해당 오브젝트를 삭제할 수 있습니다.

절차

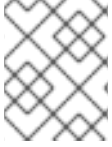
1. 저장된 상태를 초기화하여 **saved_state.json** 파일을 생성합니다.

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

2. 정리를 수행합니다.

```
# tempest cleanup
```

tempest cleanup 명령은 Tempest 리소스를 삭제하지만 프로젝트 또는 tempest 관리자 계정을 삭제하지 않습니다.



참고

유지 또는 제거하려는 오브젝트를 포함하거나 제외하도록 **saved_state.json** 파일을 수정할 수 있습니다.

5장. INTEGRATION TEST SUITE를 사용하여 OPENSTACK 클라우드 검증(TEMPEST)

tempest run 명령을 사용하여 여러 가지 방법으로 Integration Test Suite 검증을 실행할 수 있습니다. 단일 **tempest run** 명령에서 여러 옵션을 결합할 수도 있습니다.

5.1. 사전 요구 사항

- Integration Test Suite 패키지가 포함된 OpenStack 환경.
- OpenStack 환경에 해당하는 Integration Test Suite 구성입니다. 자세한 내용은 [작업 공간 생성을 참조하십시오](#).

5.2. 사용 가능한 테스트 나열

--list-tests 옵션을 사용하여 사용 가능한 모든 테스트를 나열합니다.

절차

- 사용 가능한 tempest 테스트 목록을 가져오려면 **--list-tests** 또는 **-l** 옵션을 사용하여 **tempest run** 명령을 입력합니다.

```
# tempest run -l
```

5.3. 스모크 테스트 실행

연기 테스트는 가장 중요한 기능 만 포함하는 예비 테스트 유형입니다. 이러한 테스트는 포괄적이지 않지만 스모크 테스트 실행으로 문제를 식별하는 경우 시간을 절약할 수 있습니다.

절차

- **--smoke** 옵션을 사용하여 **tempest run** 명령을 입력합니다.

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

5.4. 허용 목록 파일을 사용하여 테스트 전달

허용 목록 파일은 포함하려는 테스트를 선택하는 정규식이 포함된 파일입니다. 하나 이상의 정규식을 사용하는 경우 별도의 행에 각 표현식을 지정합니다.

절차

- 허용 목록 파일을 사용하려면 **--whitelist-file** 또는 **-w** 옵션을 사용하여 **tempest run** 명령을 입력합니다.

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

5.5. BLOCKLIST 파일을 사용하여 테스트 건너뛰기

blocklist 파일은 제외할 테스트를 선택할 정규식이 포함된 파일입니다. 하나 이상의 정규식을 사용하는 경우 별도의 행에 각 표현식을 지정합니다.

절차

- blocklist 파일을 사용하려면 **--blacklist-file** 또는 **-b** 옵션을 사용하여 **tempest run** 명령을 입력합니다.

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

5.6. 병렬로 또는 직렬로 테스트 실행

테스트를 병렬로 실행하거나 직렬로 실행할 수 있습니다. 병렬 테스트를 실행할 때 사용할 작업자 수를 정의할 수도 있습니다. 기본적으로 Integration Test Suite는 사용 가능한 각 CPU에 대해 하나의 작업자를 사용합니다.

직렬 또는 병렬로 테스트를 실행하도록 선택합니다.

- 테스트를 직렬로 실행합니다.

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

- 테스트를 병렬로 실행합니다(기본값):

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

- **--concurrency** 또는 **-c** 옵션을 사용하여 테스트를 병렬로 실행할 때 사용할 작업자 수를 지정합니다.

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

5.7. 특정 테스트 실행

--regex 옵션을 사용하여 특정 테스트를 실행합니다. 정규식은 Python 정규식이어야 합니다.

절차

- 다음 명령을 실행합니다.

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

- 예를 들어, 다음 예제 명령을 사용하여 **tempest.scenario** 로 시작하는 이름이 있는 모든 테스트를 실행합니다.

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

5.8. INTEGRATION TEST SUITE 오브젝트 삭제

tempest cleanup 명령을 입력하여 모든 Integration Test Suite(tempest) 리소스를 삭제합니다. 이 명령은 프로젝트도 삭제하지만 명령은 관리자 계정을 삭제하지 않습니다.

절차

- tempest 리소스를 삭제합니다.

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