Testing



Why?

- Changes
- Regression
- Impact of outside events (Network, Storage)
- Monitoring (over time)
- Confidence (I have a working environment)
- Recognition of potential problems (in advance)



Challenges

- My specific configuration
- My specific implementation / deployment / infra
- Networking
- Changes outside of my environment (BGP, Storage, Object Storage, DNS)

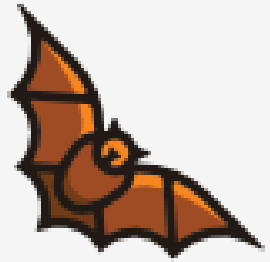


Tempest

Install

Release Notes Tempest v45

```
# Install Tempest on testing node
sudo apt-get update
sudo apt-get install build-essential python3.12-dev -y
python3 -m venv ~/venv/tempest
echo ""                                  >> ~/.profile
echo "source ~/venv/tempest/bin/activate" >> ~/.profile
source ~/.profile
pip install -U pip --proxy http://proxy.lab.nfgd.net:3128
pip install --proxy http://proxy.lab.nfgd.net:3128 -r \
  $LAB_RAW_BASE_PATH/testing/tempest/requirements.txt
```



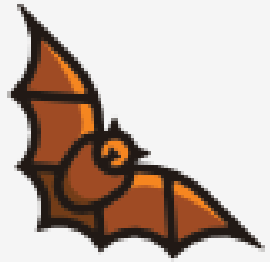
Tempest

Configuration

```
# Copy admin-openrc.sh file from deployment to testing node
scp ~/openstack/custom-config/lab/admin-openrc.sh testing.mgmt:~/
# On testing node source your openrc (credentials) file
source ~/admin-openrc.sh
# The configuration script
CONF_SCRIPT=$LAB_RAW_BASE_PATH/testing/tempest/configure.sh
# Review
curl --proxy http://proxy.lab.nfgd.net:3128 -L $CONF_SCRIPT
# Run
curl --proxy http://proxy.lab.nfgd.net:3128 -L $CONF_SCRIPT -sS | bash
# Review generated config
cd  ~/tempest-45
cat etc/tempest.conf
# Create a list of available tests
stestr list > tests-available.list
# Create a list of smoke tests
cat tests-available.list | grep smoke > smoke-tests.list


# How to run
tempest run --load-list smoke-tests.list  --regex \
  tempest.api.network.test_security_groups.SecGroupTest.test_list_security_groups


# Run full list
tempest run --load-list smoke-tests.list | tee -a $(date "+%Y%m%d%H%M%S")_smoke-tests.log
```



Scripts

openstack-health-monitor

```
# On testing node clone the repo with test script
cd ~
http_proxy=http://proxy.lab.nfgd.net:3128  \
https_proxy=http://proxy.lab.nfgd.net:3128 \
git clone https://github.com/SovereignCloudStack/openstack-health-monitor.git
cd openstack-health-monitor/
export IMG=ubuntu-24.04
export JHIMG=$IMG
export FLAVOR=m1.small
export JHFLAVOR=$FLAVOR
export ADDJHVOLSIZE=10
source ~/venv/ansible/bin/activate
source ~/automation-openrc.sh
```



Run

```
# -o    - translate nova/cinder/neutron/glance into OpenStack client commands
# -C    - full connectivity check: every VM pings every other
# -D    - create all VMs with one API call (implies -d -P)
# -n 1  - number of VMs to create
# -B    - measure TCP bandwidth between VMs (iperf3)
# -T    - assign tags to resources; use to clean up floating IPs
# -s 10 - send stats once per day (or every SH hours), not just alarms
# -W 45 - set error wait (VM only): 0-inf seconds or negative value for interactive wait
./api_monitor.sh -o -C -D -n 1 -B -T -s 10 -W 45
```