OpenStack Upgrade Testing

Tuesday, December 6, 2016 3:03 PM

OpenStack Upgrade Tests

Persistent Resources Testing

Repository

https://github.com/osic/persistent-resources-tests

Background

There are many resources that can be provisioned using an IaaS cloud, things like:

- Virtual Machines
- Volumes (Block Storage: The OpenStack Block Storage service provides persistent block storage resources that OpenStack Compute instances can consume.)
- Objects (Object Storage: The OpenStack Object Store project, offers cloud storage software so that you can store and retrieve lots of data with a simple API. It's built for scale and optimized for durability, availability, and concurrency across the entire data set.)
- Networks and subnets
- Etc.

These resources are created on demand by the user, and they persist until the user deletes them, for this reason will call these resources "persistent resources" from now on. Software updates in the OpenStack services should be transparent to this resources which should not be affected by them.

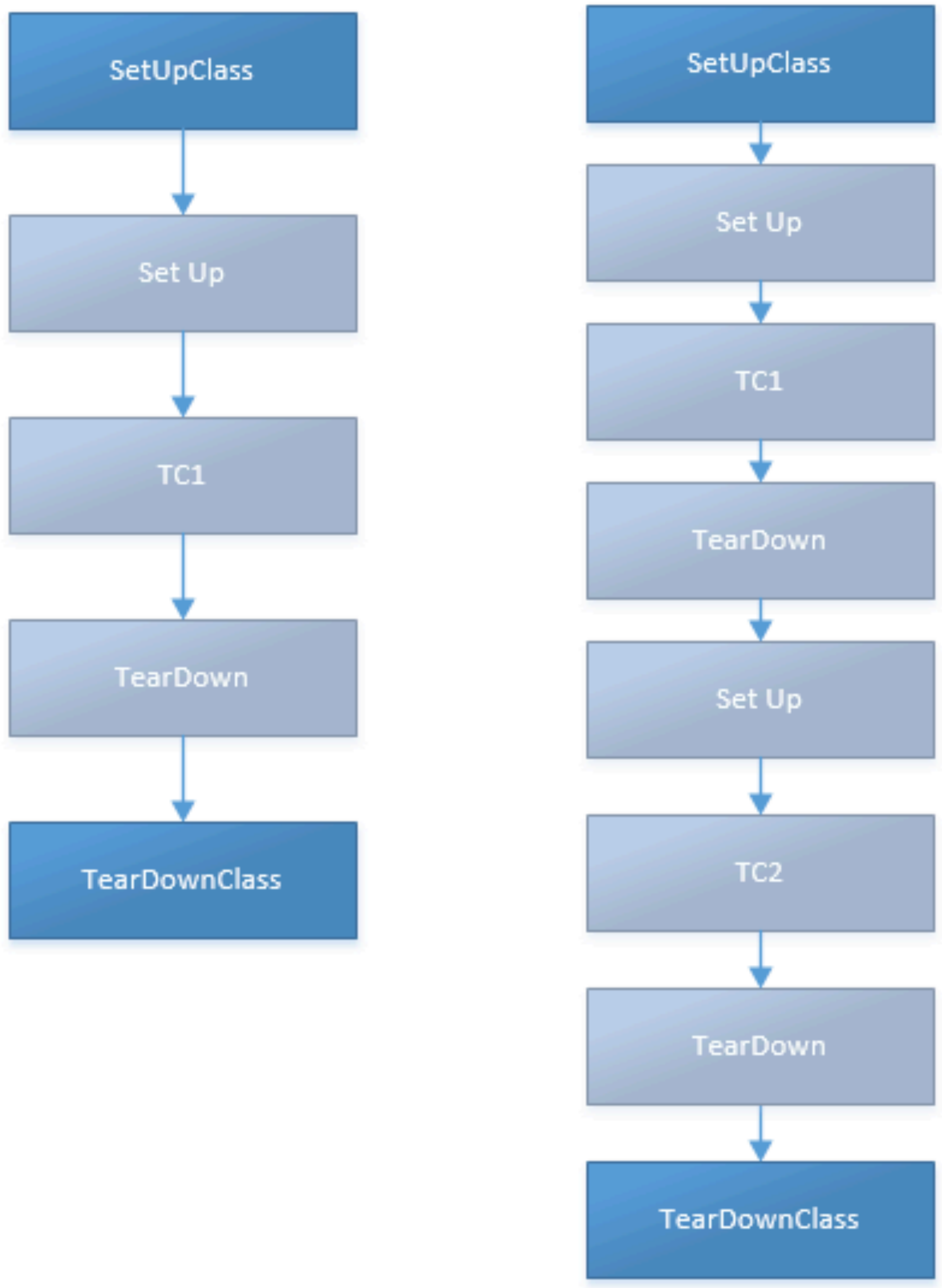
Assumption:

An existing persistent resource should not be affected and should remain

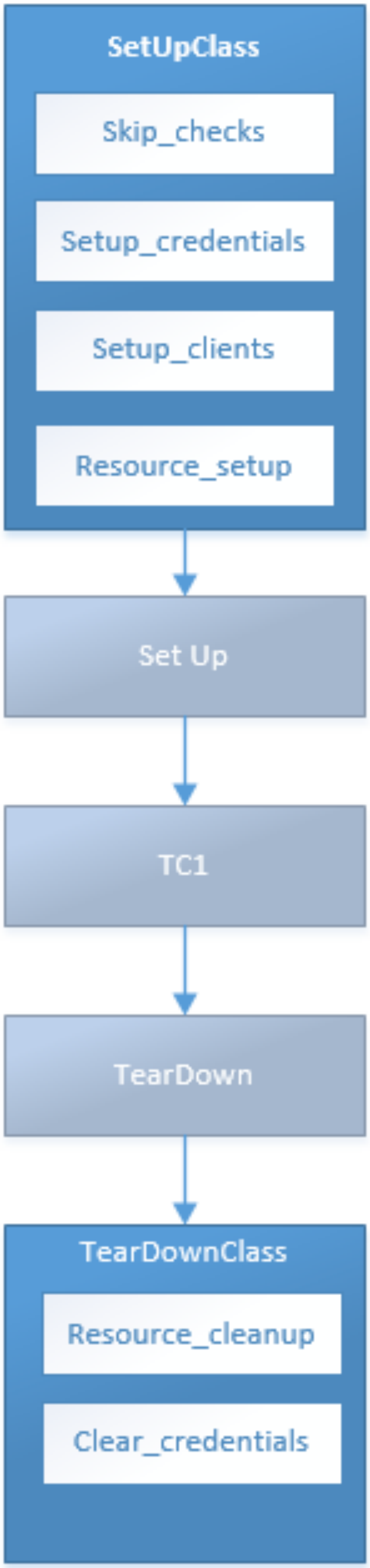
An existing persistent resource should not be affected and should remain accessible by the user during or after an OpenStack software upgrade.

Python Unittest Framework

The general structure of a unittest in Python is as follows:

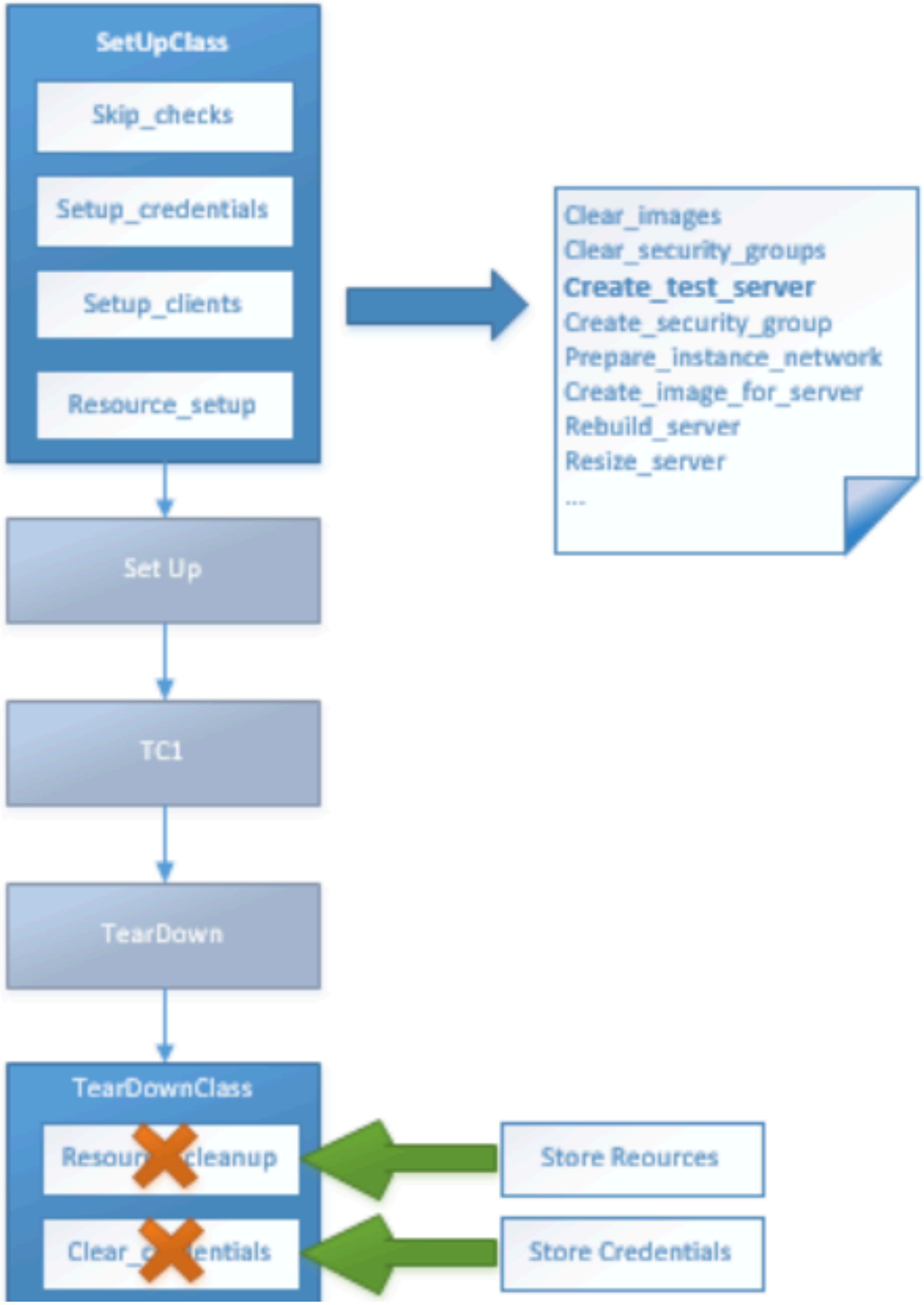


Tempest uses this framework for building their tests but it does some hacking:

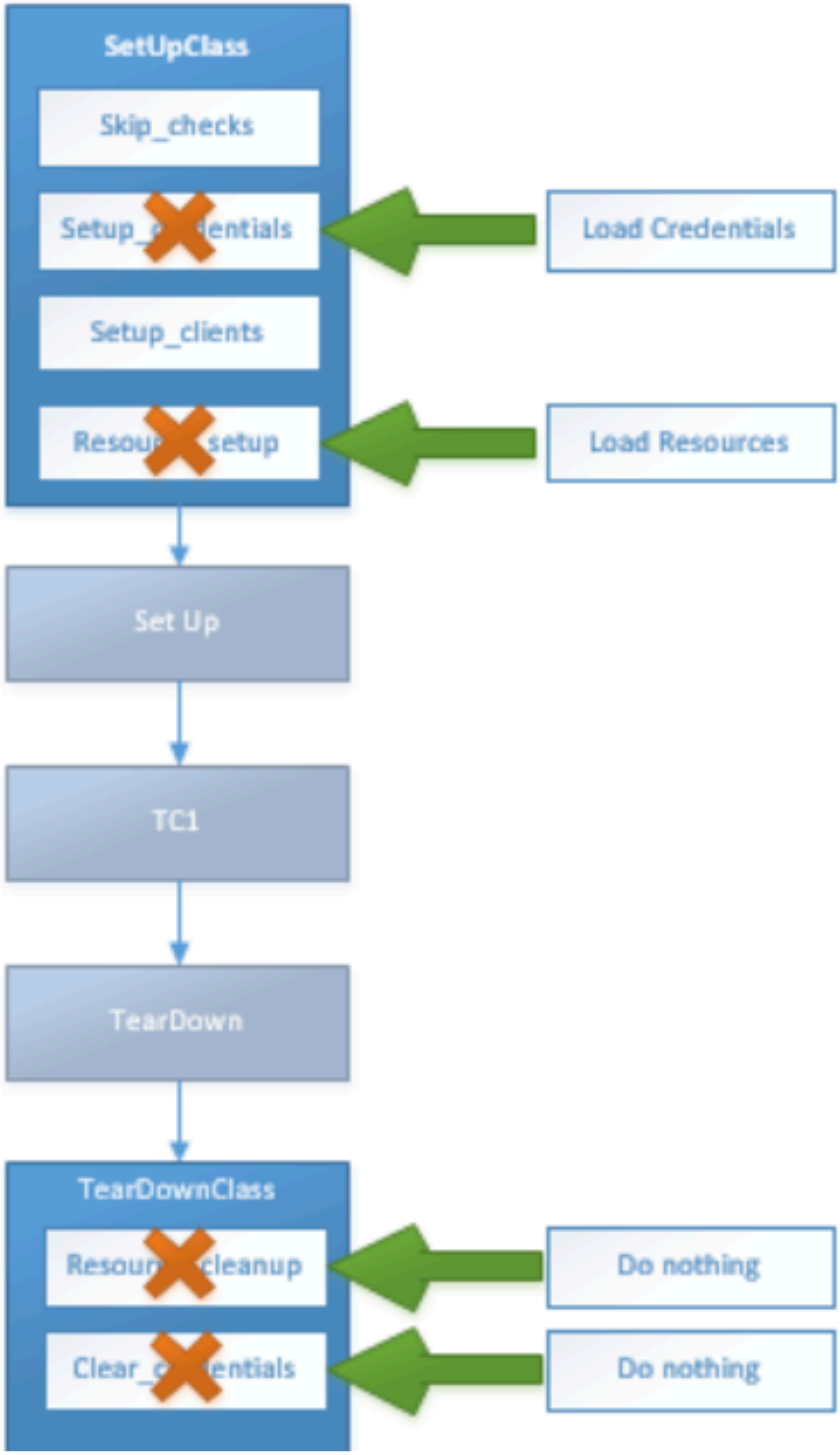


Reusing the same concept for the persistent resources tests, we need to overwrite some functions to change the default behavior of Tempest for our purpose:

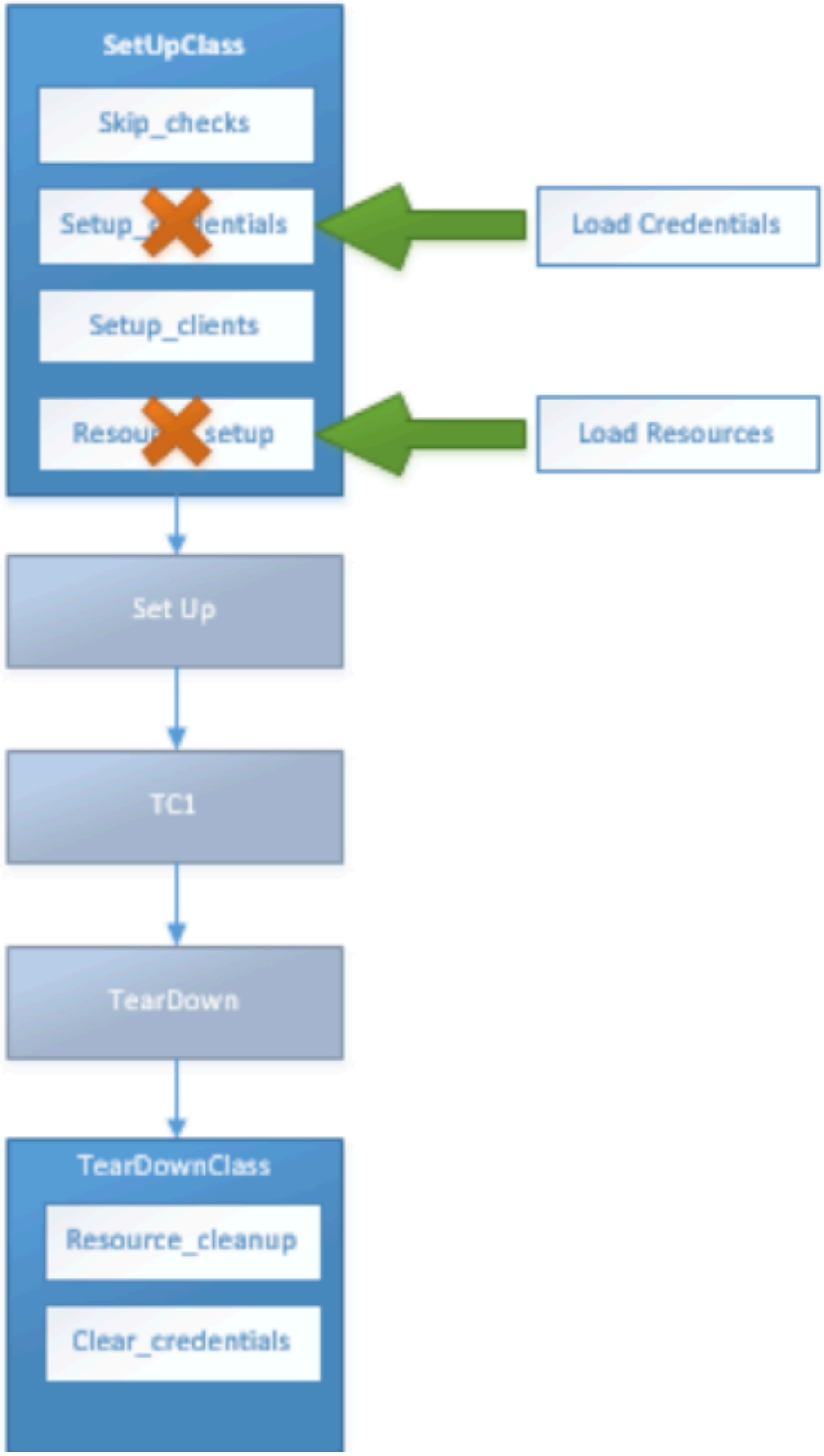
1. For creating the persistent resources.



2. For verifying persistent resources



3. For deleting persistent resources


SetUpClass
Skip_checks
Setup_ dentials
Setup_clients
Resou setup
Load Credentials
Load Resources
Set Up
TC1
TearDown
TearDownClass
Resource_cleanup
Clear_credentials