

# External Plugin Interfaces for OpenStack QA Projects

Matthew Treinish  
mtreinish@kortar.org  
mtreinish on Freenode

Andrea Frittoli  
andrea.frittoli@hp.com  
andreaf on Freenode

Oct 27, 2015

[https://github.com/mtreinish/external\\_plugins](https://github.com/mtreinish/external_plugins)

# What is OpenStack QA?

- ▶ Official Mission Statement:  
*Develop, maintain, and initiate tools and plans to ensure the upstream stability and quality of OpenStack, and its release readiness at any point during the release cycle.*

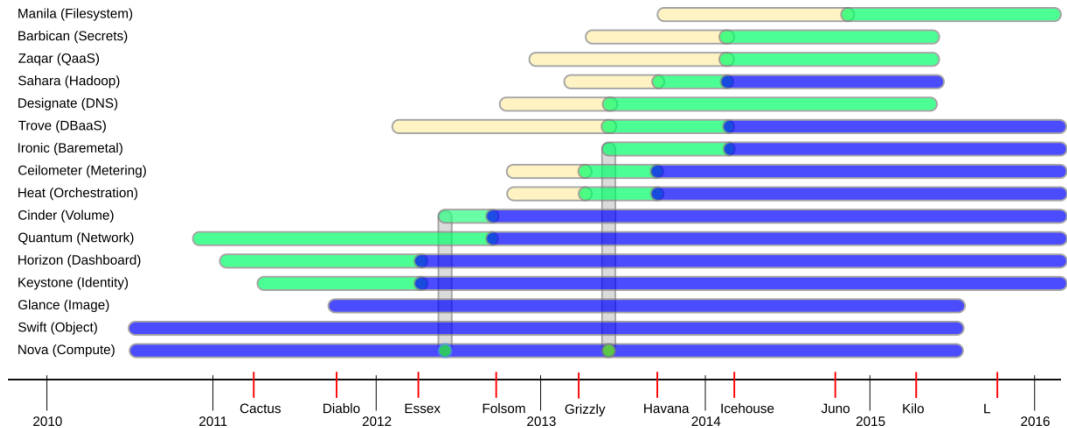
## Current QA Projects

- ▶ bashate
- ▶ devstack
- ▶ devstack-plugin-cookiecutter
- ▶ devstack-vagrant
- ▶ grenade
- ▶ tempest
- ▶ temest-lib
- ▶ tempest-plugin-cookiecutter
- ▶ stackviz
- ▶ hacking
- ▶ eslint-config-openstack
- ▶ os-testr
- ▶ openstack-health dashboard

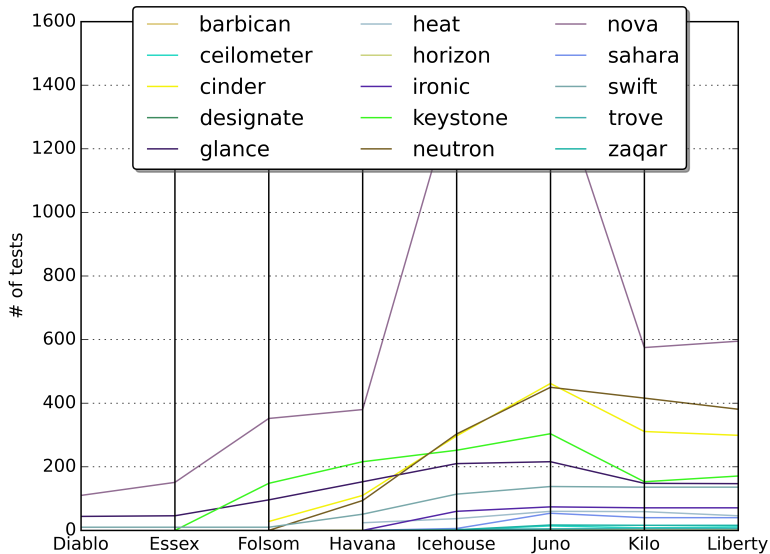
## Scaling Limitations

- ▶ Previously QA projects supported all incubated and integrated OpenStack projects
- ▶ This works with 5 projects
- ▶ At 10 things are stretched very thinly

# OpenStack Project Growth



# Tempest Tests per Project



## The Big Tent. . .



## QA in the Big Tent

- ▶ QA projects now directly support base
- ▶ Provide stable plugin interfaces to expand functionality
- ▶ Better fits with the growth in projects



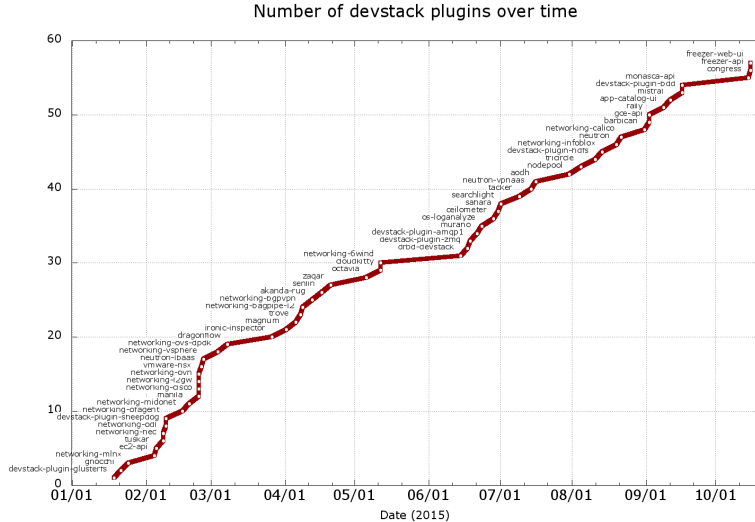
## Current Projects QA directly supports in-tree

- ▶ Keystone
- ▶ Nova
- ▶ Glance
- ▶ Cinder
- ▶ Neutron
- ▶ Swift

# Devstack Plugins

- ▶ Multiple functions
  - ▶ Enable deploying extra projects in devstack
  - ▶ Enable using custom drivers in devstack
  - ▶ Enable deploying extra services on top of a devstack cloud
- ▶ Bits of bash code that live outside of devstack tree
- ▶ Called via a strong contract
- ▶ Registry of plugins maintained by the community:  
<http://docs.openstack.org/developer/devstack/plugin-registry.html>

# Projects using Devstack Plugins



Source: `git log -diff-filter=A -format=%cd -date=short -1 - devstack/plugin.sh`

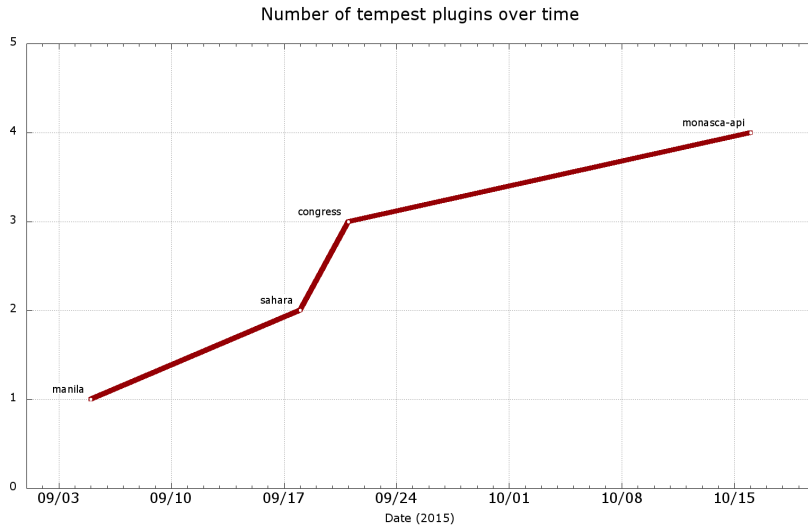
# Howto write a Devstack Plugin

- ▶ Devstack plugin cookie-cutter:  
`git://git.openstack.org/openstack-dev/devstack-plugin-cookiecutter`
- ▶ settings
  - ▶ Global variables sourced very early

# Tempest Plugins

- ▶ Integrates any external tests into a tempest run
- ▶ Unifies configuration between plugin(s) and in-tree
- ▶ Based on stevedore extension manager
- ▶ Automatically discovered when installed
- ▶ Test code built on top of tempest-lib

# Projects using tempest plugins



Source: `git blame -L '/^tempest/,+1' setup.cfg | awk '{print $1}' | xargs git log -1 --format=%cd --date=short`

# Howto Write a Tempest Plugin

- ▶ Tempest plugin cookie-cutter:  
`git://git.openstack.org/openstack/tempest-plugin-cookiecutter`
- ▶ Code can be hosted in existing or dedicated git project
- ▶ Extend `plugins.TempestPlugin` from `tempest` (or `tempest-lib`)
- ▶ Implement all abstract methods
  - ▶ Test discovery
  - ▶ Configuration options
- ▶ Is everything you need is `tempest-lib`?
- ▶ Setup CI

# Tempest Plugin Interface

```
@six.add_metaclass(abc.ABCMeta)
class TempestPlugin(object):
    """A TempestPlugin class provides the basic hooks for an external
    plugin to provide tempest the necessary information to run the plugin.
    """

    @abc.abstractmethod
    def load_tests(self):
        """Method to return the information necessary to load the tests in the
        plugin.

        :return: a tuple with the first value being the test_dir and the second
                 being the top_level
        :rtype: tuple
        """
        return
```



# Tempest Plugin Interface

```
@abc.abstractmethod
def register_opts(self, conf):
    """Method to add additional configuration options to tempest. This
    method will be run for the plugin during the register_opts() function
    in tempest.config

    :param ConfigOpts conf: The conf object that can be used to register
        additional options on.
    """
    return

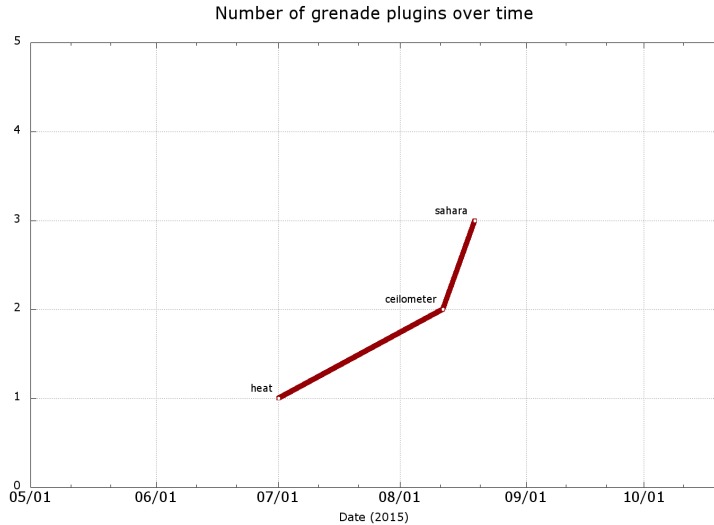
@abc.abstractmethod
def get_opt_lists(self):
    """Method to get a list of options for sample config generation

    :return option_list: A list of tuples with the group name and options
        in that group.
    :rtype: list
    """
    return []
```

# Grenade Plugins

- ▶ Enables running upgrades from out of tree
- ▶ Enables adding additional services to the old devstack
- ▶ Add additional resource creation and checks during the upgrade

# Projects using Grenade Plugins



Source: `git log -diff-filter=A -format=%cd -date=short -1 -- devstack/upgrade/upgrade.sh`

# Howto Write a Grenade Plugin

- ▶ `upgrade.sh`
  - ▶ The script to run the upgrade after the old has been shutdown
- ▶ `shutdown.sh`
  - ▶ The script to shutdown a service
- ▶ `settings`
  - ▶ register any devstack plugins
  - ▶ register any services for upgrade
- ▶ `resources.sh`
  - ▶ Create resources
  - ▶ Check resource survival across upgrade
  - ▶ Teardown resource after upgrade
- ▶ Per Release manual steps
  - ▶ Any manual steps to upgrade between releases

## Where to get more information

- ▶ Tempest Plugin Docs: <http://docs.openstack.org/developer/tempest/plugin.html>
- ▶ Devstack Plugin Docs:  
<http://docs.openstack.org/developer/devstack/plugins.html>
- ▶ Grenade Plugin Docs: <http://docs.openstack.org/developer/grenade/plugins.html>
- ▶ openstack-dev ML: [openstack-dev@lists.openstack.org](mailto:openstack-dev@lists.openstack.org)
- ▶ #openstack-qa on Freenode

Questions?