

# Project Team Lead (PTL) Responsibilities

So you want to be a PTL? Here is a list of items that you can expect to perform. This document is meant to serve as a general guide for current and future PTLs. It is not all encompassing, nor are all sections required, it is simply a guide that you may refer to. Each OpenStack project is different and will release, plan, and delegate differently, too.

## Weekly Tasks

1. Organize the team meeting agenda. Most teams will use a wiki or etherpad.
2. Follow the weekly *[release]* guidelines email to keep track of the development cycle tasks (unless you have an appointed release liaison to follow it for you). It is also very useful to subscribe into the [release calendar](#).
3. Attend Technical Committee and Cross-Project meetings when possible. For more information see [meetings](#).
4. If the team does not have an appointed bug czar, then perform a bug triage. Ensure all *new* bugs that have been reported are triaged. The below is a URL to view all *new* bugs for keystone.

```
https://bugs.launchpad.net/keystone/+bugs?orderby=status&start=0
```

5. If the team does not have an appointed bug czar, then remember to also tag bugs accordingly. The below is a URL to view all untagged bugs for keystone.

```
https://bugs.launchpad.net/keystone/+bugs?field.tag=-*
```

## Core member maintenance

This will vary greatly from team to team, but here is a general guide you may consider.

### Criteria when considering new cores

A successful appointment can be made solely by the PTL, however it is encouraged to hear the opinions of the existing core team. This can be done through private messages on IRC or publicly through the openstack-dev mailing list.

1. Are their review numbers (quantity and disagreements) similar to those of existing core members?
2. Does the person follow up on reviews?
3. Are the reviews helpful, critical, and nice to authors?
4. Does the person help in triaging bugs? (For example: reproducing the issue)
5. Does the person provide bug fixes?
6. Does the person create and implement new features?
7. Does the person review specifications?
8. Does the person participate in weekly meetings?
9. Does the person participate at the summit design sessions?
10. Does the person participate at the mid-cycle / project-team-gathering?
11. Does the person know the project direction and priorities?
12. Does the person help others on IRC?

## Actions to perform when adding new cores

---

Once a person is ready, perform the following:

1. Announce new core member on mailing list
2. Give them voice on IRC (if applicable):

```
/msg chanserv flags <room> <irc_handle> +v
```

3. Add them to the project drivers group in launchpad.
4. Update the core team in Gerrit.

## Criteria when removing new cores ¶

1. Is the core reviewer reviewing enough? Use data (i.e. stackalytics) to determine if the core reviewer is among the other core members in terms of overall review count.
2. Does the core reviewer have enough time to review? Job responsibilities may have changed.
3. Does the core reviewer no longer have knowledge of the code base and new features?
4. Does the core reviewer still provide useful feedback, even if infrequently?

## At the beginning of a new cycle

---

1. Add placeholder migrations, these should be the first things to merge once the n-1 stable branch has been created. *Do not merge any migrations before this is done*
2. Appoint [cross project liaisons](#) (Docs, release, QA, Oslo, etc.).
3. Check if the meeting time works for most active contributors.
4. If you keep release liaison responsibilities, join [#openstack-release](#) and be sure to follow [\[release\]](#) emails from the mailing list.
5. If necessary, squash database migrations. This is usually not necessary, but will reduce the amount of time necessary for upgrading. A sample squash can be seen [by the keystone team](#).
6. Track removed and deprecated features, for example using *deprecated-as-of-<series>* and *removed-as-of-<series>* blueprints.
7. Organize the specifications page.

## During the cycle

1. Help first time contributors, they will be struggling the most.
2. Lack of reviews? Reach out to the core team and remind them.
3. Release libraries as necessary but don't wait too long! Some teams will release after 4 weeks even if the changes are minor. *More often is better than less often.*

## At the end of the cycle

1. Clean up release notes.
2. Coordinate with the [release management](#) team for deliverables, unless a liaison has been appointed
3. Expect queries from the release marketing staff to name release highlights and major features
4. Perform a retrospective via an etherpad. Suggested sections include: *What went well?*, *What didn't go well*.
5. Analyze how *complete* each new feature is. Does it have DevStack support? Horizon support? Client bindings? CLI support? Documentation? Does the install guide need to be updated?

## Before the Summit

---

1. Start an etherpad for the design session proposals mail it out. For example: <https://www.mail-archive.com/openstack-dev@lists.openstack.org/msg49312.html>
2. Create an etherpad for every design session, prime the content. List these etherpads in the Wiki and send out a note to mailing list. For example:  
<http://lists.openstack.org/pipermail/openstack-dev/2015-October/076283.html>

## During the summit

---

1. Reach out to new contributors to the project at the design sessions.
2. Attend as many cross-project sessions as possible.
3. At the project design sessions.
  - Take notes on the etherpad (or delegate a scribe)
  - Act as a moderator rather than actively participate (or delegate a moderator)

## Stable

---

Alternatively, the responsibilities in this section can be delegated to a local stable maintenance czar.

1. Ensure the stable branches gates are not broken.
2. Co-ordinate with the stable release team to ensure releases are performed when a critical fix is backported, or sufficient smaller fixes have landed.

## One offs

---

When necessary, the following can be performed at unscheduled times.

1. Bug smashes
2. API sprints