Rebasing a pull request

You may find that your pull request (PR) is out-of-date and needs to be rebased. This can happen for several reasons:

- Files modified in your PR are in conflict with changes which have already been merged.
- Your PR is old enough that significant changes to automated test infrastructure have occurred.

Rebasing the branch used to create your PR will resolve both of these issues.

Configuring your remotes

Before you can rebase your PR, you need to make sure you have the proper remotes configured. These instructions apply to any repository on GitHub, including collections repositories. On other platforms (bitbucket, gitlab), the same principles and commands apply but the syntax may be different. We use the ansible/ansible repository here as an example. In other repositories, the branch names may be different. Assuming you cloned your fork in the usual fashion, the `origin` remote will point to your fork:

```
$ git remote -v
origin  git@github.com:YOUR_GITHUB_USERNAME/ansible.git (fetch)
origin  git@github.com:YOUR_GITHUB_USERNAME/ansible.git (push)
```

However, you also need to add a remote which points to the upstream repository:

```
$ git remote add upstream https://github.com/ansible/ansible.git
```

Which should leave you with the following remotes:

```
$ git remote -v
origin  git@github.com:YOUR_GITHUB_USERNAME/ansible.git (fetch)
origin  git@github.com:YOUR_GITHUB_USERNAME/ansible.git (push)
upstream        https://github.com/ansible/ansible.git (fetch)
upstream        https://github.com/ansible/ansible.git (push)
```

Checking the status of your branch should show your fork is up-to-date with the `origin` remote:

```
$ git status
On branch YOUR_BRANCH
Your branch is up-to-date with 'origin/YOUR_BRANCH'.
nothing to commit, working tree clean
```

Rebasing your branch

Once you have an `upstream` remote configured, you can rebase the branch for your PR:

```
$ git pull --rebase upstream devel
```

This will replay the changes in your branch on top of the changes made in the upstream `devel` branch. If there are merge conflicts, you will be prompted to resolve those before you can continue.

After you rebase, the status of your branch changes:

```
$ git status
On branch YOUR_BRANCH
Your branch and 'origin/YOUR_BRANCH' have diverged,
and have 4 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)
nothing to commit, working tree clean
```

Don't worry, this is normal after a rebase. You should ignore the `git status` instructions to use `git pull`. We'll cover what to do next in the following section.

Updating your pull request

Now that you've rebased your branch, you need to push your changes to GitHub to update your PR.

Since rebasing re-writes git history, you will need to use a force push:

```
$ git push --force-with-lease
```

Your PR on GitHub has now been updated. This will automatically trigger testing of your changes. You should check in on the status of your PR after tests have completed to see if further changes are required.

Getting help rebasing

For help with rebasing your PR, or other development related questions, join us on the #ansible-devel chat channel (using Matrix at ansible.im or using IRC at irc.libera.chat).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\(ansible-devel)(docs)(docsite)(rst)(dev_guide)developing_rebasing.rst`, line 94)

Unknown directive type "seealso".

```
.. seealso::

   :ref:`community_development_process`
       Information on roadmaps, opening PRs, Ansibullbot, and more
```