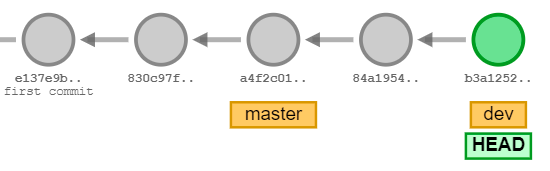
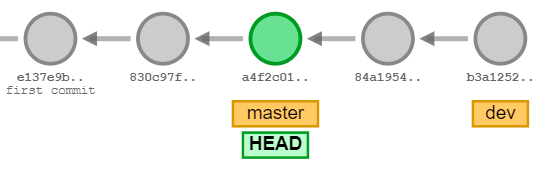
**Git fast-forward Merge:**

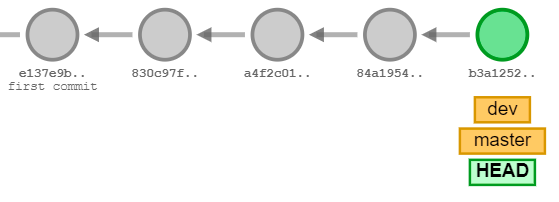
1. dev branch is created from the parent ‘master’ branch and current HEAD is at dev branch which has active development changes.

****

1. After dev branch changes are completed we are required to merge the changes back to parent branch ‘master’.

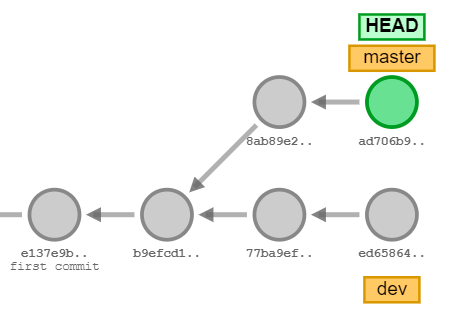
****

1. We run the command, $ git merge dev, allowing all changes from dev to be merged into master. Here no actual merge happens instead the HEAD pointer which is pointing to master (base code of dev), the HEAD is moved from earlier version to latest on tip of dev branch.

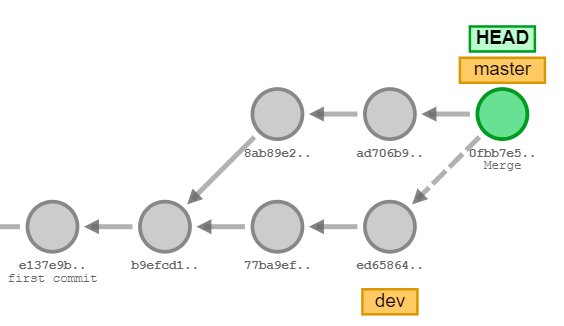
****

**Git Three-way Merge (Recursive Merge):**

1. There are two branches master and dev with dev as child branch of master. Here both dev and master branches have progressed ahead after the dev branch is created.

****

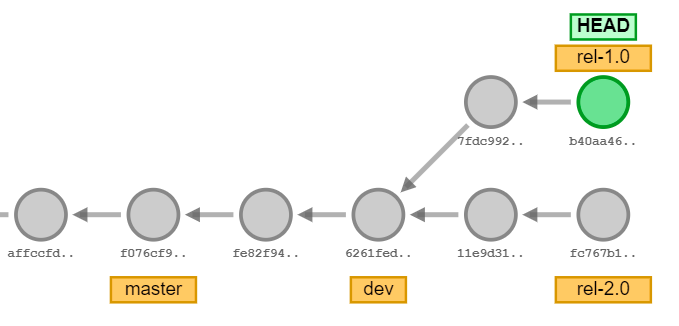
1. Now we want to merge dev with master, as the master has progressed ahead after the dev branch base code hash, fast-forward merge will no work, instead, git performs three-stage comparison and the results are as shown below.

****

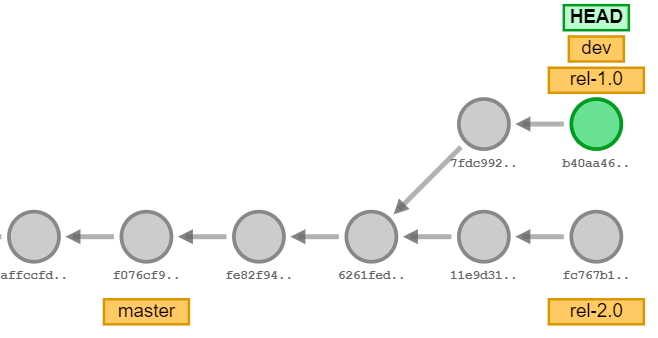
**testing**

**GIT Rebase operation:**

1. Consider we have below branch structure. Dev is the main development branch with rel-1.0 and rel-2.0 as parallel branches with same base branch.

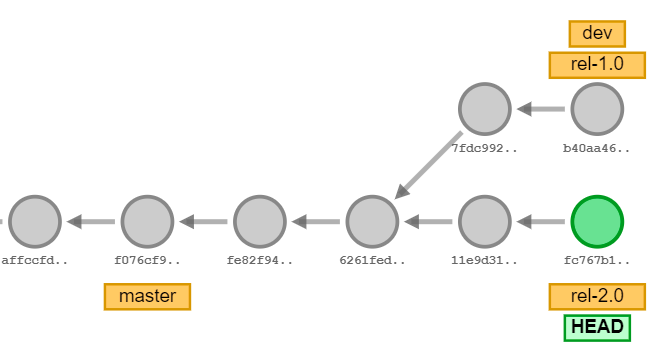
****

1. As the rel-1.0 approaches deployment (prod release), rel-1.0 branch is merged to Dev (parent branch). This is going to be a fast-forward merge.

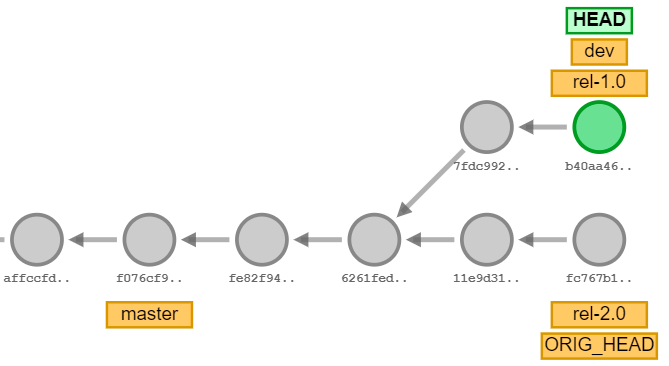
****

Version release to production

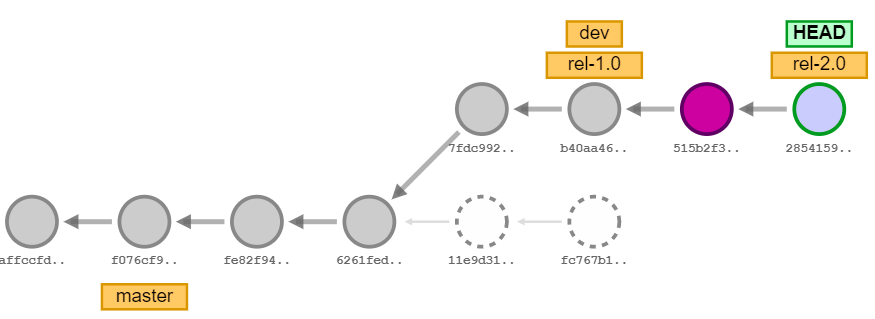
1. After rel-1.0 is complete, we can delete the branch rel-1.0 and the new code in production (dev/master branch) should become the new base for rel-2.0. Now this can be done using the rebase command, which works in a very different way than a Merge.
2. Move/ switch to rel-2.0 branch and run command $ git rebase dev

****

1. After we run the rebase command, HEAD moved to the tip of dev branch.

****

1. All commits in rel-2.0 branch after the common base is detached and moved on top of current tip of dev branch.



Detached commits hashes then deleted / garbage collected

New commits created as results of rebase

While working with rebasing, there’s a possibility of conflict. Now to resolve the conflict, we edit the file in conflict and run git add command and then use $ git rebase-continue command.

