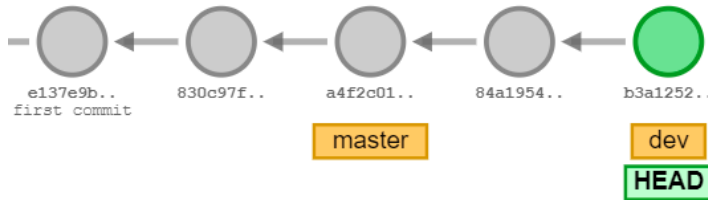


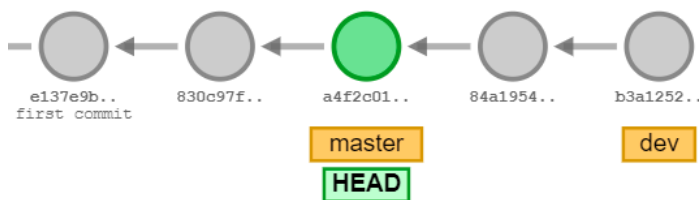


Git fast-forward Merge:

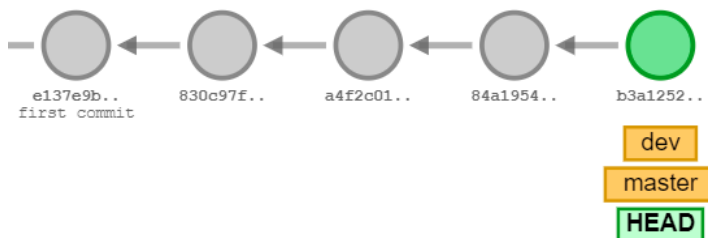
- 1) `dev` branch is created from the parent '`master`' branch and current HEAD is at `dev` branch which has active development changes.



- 2) After `dev` branch changes are completed we are required to merge the changes back to parent branch '`master`'.

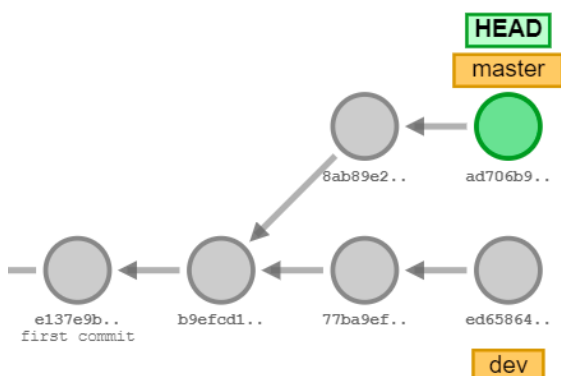


- 3) We run the command, `$ git merge dev`, allowing all changes from `dev` to be merged into `master`. Here no actual merge happens instead the HEAD pointer which is pointing to `master` (base code of `dev`), the HEAD is moved from earlier version to latest on tip of `dev` branch.



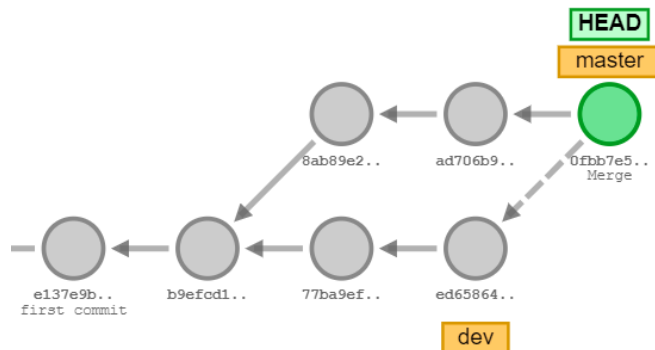
Git Three-way Merge (Recursive Merge):

- 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.



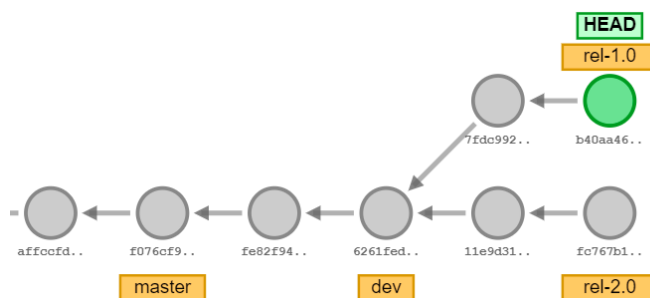


- 2) Now we want to merge `dev` with `master`, as the `master` has progressed ahead after the `dev` branch base code hash, fast-forward merge will no work, instead, git performs three-stage comparison and the results are as shown below.

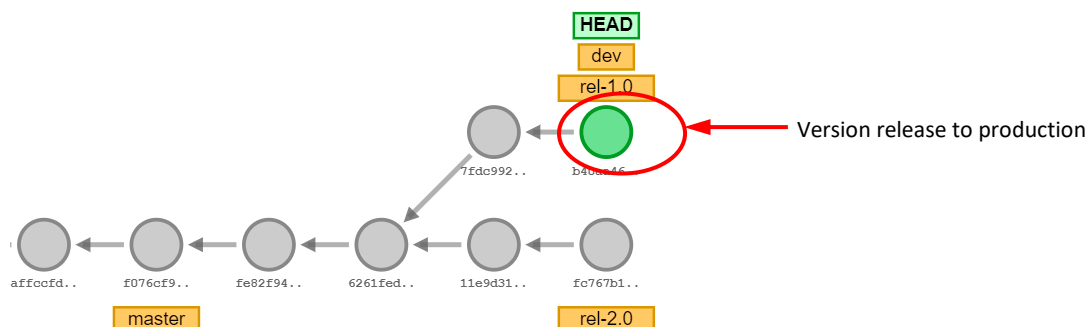


GIT Rebase operation:

- 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.



- 2) 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.

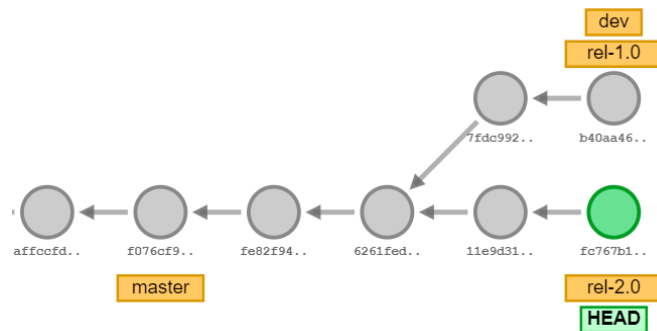


- 3) After `rel-1.0` is complete, we can delete the branch `rel-1.0` and the new code in production (`dev/master` branch) should become the new base for `rel-2.0`. Now this can be done using the rebase command, which works in a very different way than a Merge.

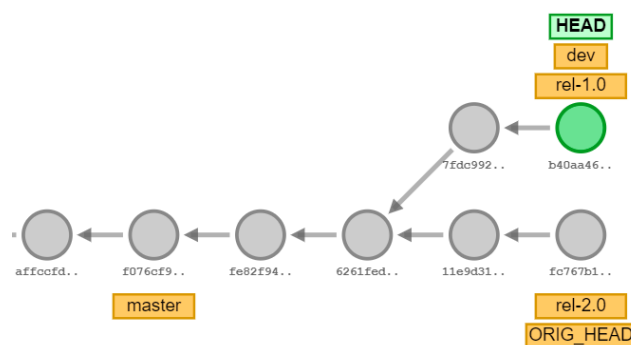
Working with git branch, Merge and Rebase commands



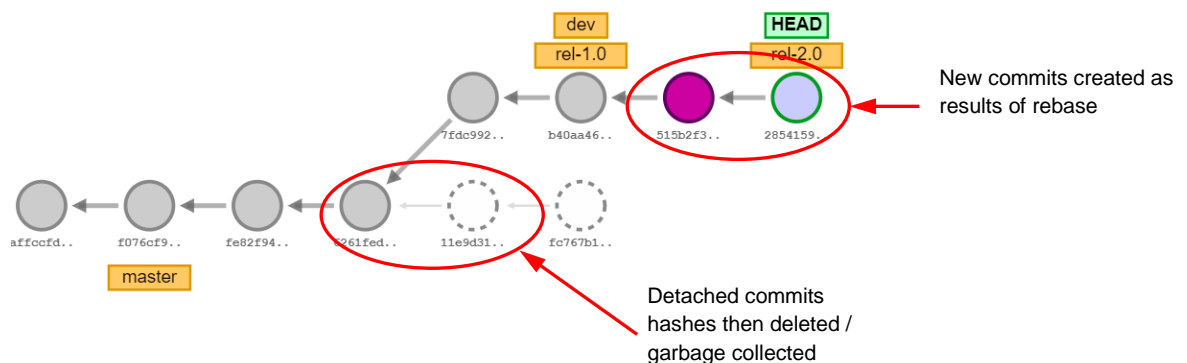
- 4) Move/ switch to rel-2.0 branch and run command `$ git rebase dev`



- 5) After we run the `rebase` command, HEAD moved to the tip of `dev` branch.



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



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.



1)

```
root@gitserver:~/project# git checkout dev
Switched to branch 'dev'
root@gitserver:~/project# git rebase master
First, rewinding head to replay your work on top of it...
Applying: initial commit
Applying: new change
Applying: one more change
Applying: one more change
Applying: added jenkins pipeline file
Applying: form updated with department field
Applying: form updated by root user
Using index info to reconstruct a base tree...
M       src/main/webapp/index.jsp
<stdin>:11: space before tab in indent.
        <input type="text" name="lastname"><br>
<stdin>:15: trailing whitespace.
        <input type="text" name="Gender"><br>
warning: 2 lines add whitespace errors.
Falling back to patching base and 3-way merge...
Auto-merging src/main/webapp/index.jsp
CONFLICT (content): Merge conflict in src/main/webapp/index.jsp
Failed to merge in the changes.
Patch failed at 0007 form updated by root user
The copy of the patch that failed is found in:
    /root/project/.git/rebase-apply/patch

When you have resolved this problem, run "git rebase --continue".
If you prefer to skip this patch, run "git rebase --skip" instead.
To check out the original branch and stop rebasing, run "git rebase --abort".
```

2)

```
root@gitserver:~/project# git branch
* (no branch, rebasing dev)
dev
master
root@gitserver:~/project# git rebase --skip
Applying: file updated to resolve conflict
Using index info to reconstruct a base tree...
M       src/main/webapp/index.jsp
Falling back to patching base and 3-way merge...
Auto-merging src/main/webapp/index.jsp
CONFLICT (content): Merge conflict in src/main/webapp/index.jsp
Failed to merge in the changes.
Patch failed at 0008 file updated to resolve conflict
The copy of the patch that failed is found in:
    /root/project/.git/rebase-apply/patch

When you have resolved this problem, run "git rebase --continue".
If you prefer to skip this patch, run "git rebase --skip" instead.
To check out the original branch and stop rebasing, run "git rebase --abort".

root@gitserver:~/project# git rebase --abort
root@gitserver:~/project#
```



3)

```
root@gitserver:~/project# git checkout master
Switched to branch 'master'
root@gitserver:~/project# git merge dev
CONFLICT (modify/delete): file2 deleted in dev and modified in HEAD. Version HEAD of file2 left in tree.
CONFLICT (modify/delete): file1 deleted in dev and modified in HEAD. Version HEAD of file1 left in tree.
Automatic merge failed; fix conflicts and then commit the result.
root@gitserver:~/project#
```