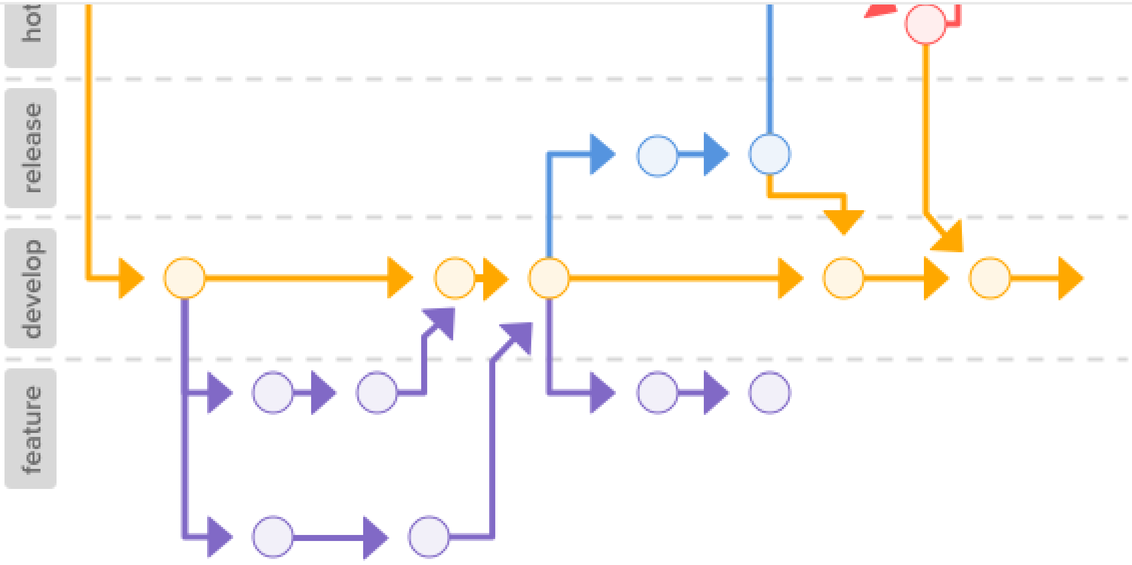
Branching workflows

Let's take a look at the **Gitflow Workflow** as outlined in A successful Git branching model.

This workflow consists of five types of branches, each with different roles:

- Master
- Feature branch (aka Topic branch)
- Release branch
- Hotfix branch
- Develop branch (aka Integration branch)

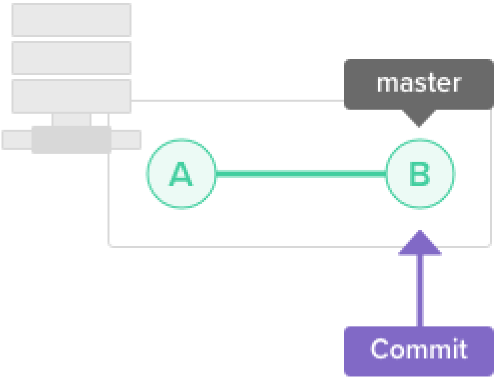


Basic Git branching workflow with master, topic, release, and hotfix branches.

Master

Upon making the first commit in a repository, Git will automatically create a master branch by default. Subsequent commits will go under the master branch until you decide to create and switch over to another branch.

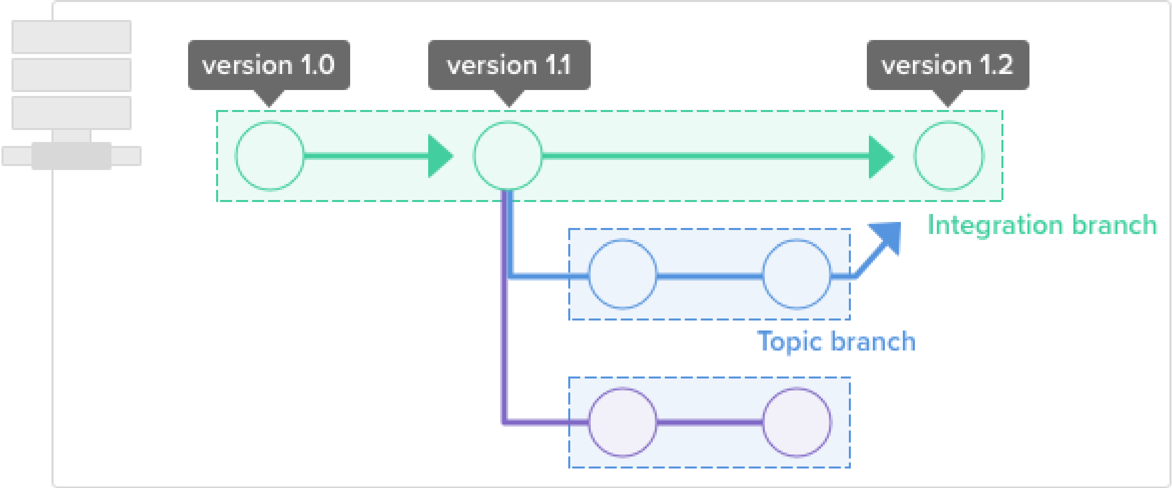
Codebase residing in the master branch is considered to be production-ready. When it is ready for a specific release, the latest commit will be given a release tag.



Changes are committed to the master branch.

Feature/Topic branch

You will push this branch to the remote repository whenever you are ready to merge the change set with the develop/integration branch.



Release branch

When you roll out a new release, you create a release branch. A release branch helps you to ensure that the new features are running correctly.

By convention, release branch names normally start with the prefix "release- ".

The release branch is typically created off the develop/integration branch when it's close to being production-ready.

Only bug fixes and release related issues should be addressed on this branch. Having this branch will allow other team members to continue pushing new features to the develop/integration branch without interrupting the release workflow.

When you are ready to release, merge the release branch with the master branch and tag a release number to the newly created merge commit.

You should also merge the release branch with the develop/integration branch so that both the master and develop/integration branches receive you can create a Hotfix branch off the master branch.

By convention, hotfix branch names normally start with the prefix "hotfix-".

The advantage of a hotfix branch is that it allows you to quickly issue a patch and have the change merged with the master branch without having to wait for the next release.

A hotfix branch should be merged with the develop/integration branch as well.

Develop/Integration branch

A develop/integration branch should be kept stable at all times. This is