Winning at CI with feature branches and Bamboo

Branching makes it easier to release on a moment's notice – but there's a catch.

BY SEPIDEH SETAYESHFAR

It's no secret that developers need freedom to make changes and run tests without the fear of messing up the main code line. And they can, thanks to branching – the magical element that allows for parts of the code to be modified, developed, and tested in isolation *and in parallel.*

Thanks to Git and Mercurial, prolific branching has passed from the realm of mythical practices into the world of the practical, as we discussed at length in 5 tips for CI-friendly Git repos. Yet despite all the benefits, using branches in a continuous delivery pipeline can create two distinct problems:

Requirements

> **Time:**
>
> 30 minutes
>
> **Audience:**
>
> You are new to continuous deployment and/or Bitbucket Pipelines
>
> **Prerequisite:**
>
> - Create a Bitbucket Account
> - Follow the continuous integration tutorial
> - Follow the continuous delivery tutorial
>
> Try it free

Duplication and drift of build configuration

- When a developer creates a branch, a handful of build configurations on the CI server need to be cloned manually. Because it's a manual step, some developers don't bother with it, and thus no tests are executed as they develop the feature. Even when we do bother to copy CI configs for our feature branches, we often make little tweaks here n' there. Then we clone the tweaked configs for our next feature branch, make more tweaks, and eventually the build configuration drifts away from the original configuration it was cloned from, causing administration headaches (and unpleasant surprised when the branch is merged back to main).

Uncertainty of integration state

- Unless we're constantly merging changes from the main line (I'll refer to this as main) into our branch and run the tests, there is always uncertainty around whether our changes work or not – at least until the branch is merged into the main line. Inevitably, the changes don't work as expected and main is now polluted with broken code, defeating the whole point of feature branches and CI!

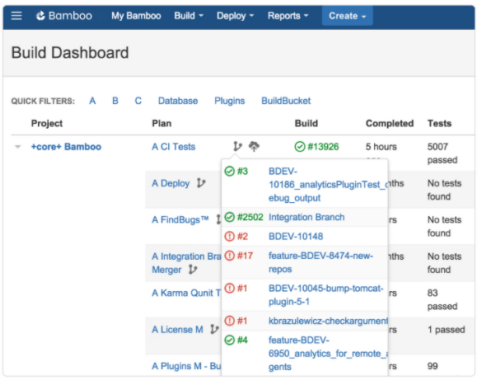
Bamboo has addressed and resolved both these issues with a feature we call plan branches, which will be explained in detail here. Read on!

Plan branches for feature branches

A plan branch is an element of your build plan in Bamboo that corresponds to a branch in your repository. They inherit all of the configuration defined by the parent plan, except that instead of building against main, they build against a branch. Then, when the branch build succeeds, changes can be merged automatically from the branch to main. Alternatively, successful branch builds can be deployed to a test environment using Bamboo's deployments features (more on that in another article).

As you can see in the screenshot below, our plan called "A CI Tests" is building not only main, but also the integration branch, and a handful of branches dedicated to single

Jira issues (BDEV-10045-bump-tomcat-plugin-5-1, for example). Bamboo makes it easy to spot which plans have branches by displaying the branch symbol.

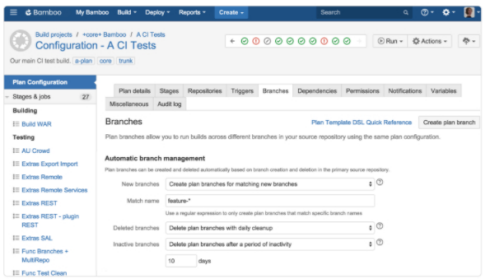


To set this up for an existing plan...

- Log in as a user with admin privileges for that plan.
- Go to the configuration screen by clicking the pencil icon on Bamboo's dashboard or using the **actions** menu on the built results screen.
- On the configuration screen, click the **branches** tab.

From here, we can configure Bamboo to automatically detect new branches in the repository, and create the corresponding plan branch. Note that automatic branch detection is currently available for Git, Mercurial, and Subversion repositories only. But don't feel left out if you use another version control system like TFS or Perforce – you can create a plan branch manually, too.

> **Pro Tip:** You probably won't want to build every single branch you create. If you use a naming convention where all branches that you want tested start with "feature-*", for example, then Bamboo can be configured to only create plan branches that match that prefix. (You can use regular expressions in these configs, too).



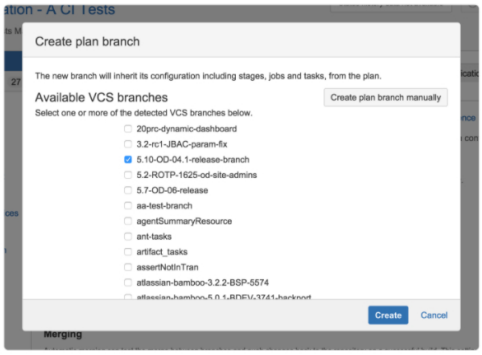
Since most feature branches are short-lived (3-4 days at most), you can configure the plan to remove the branch and its build results from Bamboo after a certain number of days of inactivity. This can be disabled on a per-branch basis if you want to keep it around for an indefinite amount of time, after which you will have to remove it manually. Don't worry: whether you remove a plan branch automatically or by hand, Bamboo won't remove the branch itself from your repository – we leave that decision up to you. That said, disabling automatic cleanup outright for some branches is a good idea for situations where you might maintain a long lived "stable" branch of your project.

Building every branch the easy way

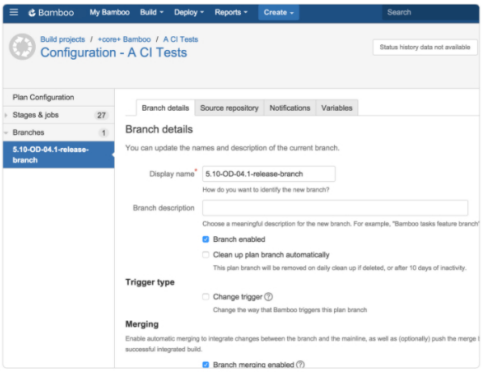
Although we highly recommend using Git, it's possible to create a plan branch for any version control system supported by Bamboo. To create a plan branch by hand...

- Go to the build results screen for your plan and select **configure plan** from the **actions** menu.
- Click on the **branches** tab, then the click the **create branch** button.
- For Git and Mercurial, a dialog will appear listing all the available branches in the repository which do not yet have a corresponding plan branch in Bamboo.
- Pick a branch and hit **create**.
- Users of centralized VCS repositories will be asked to

- simply choose a name, description and what branch they want to use (for Subversion this is the URL to the branch) before hitting **create**.



You can now disable the automatic cleanup for the current Plan Branch if you wish, and in the case of DVCS repos, configure a Merge Strategy. Merge Strategies allow Bamboo to automatically merge code between branches on successful builds (more on that later). It is also possible at this point to override repository information (if you are using Plan Branches with Subversion, you can change the SVN URL from trunk to branch here) or override build variables. For example, you might have a variable that specifies whether you are going to deploy to production that you may want to change for the Plan Branch.



Because the build steps for parent plan builds and plan branch builds work exactly the same, configs like triggers and notifications will be inherited from the parent plan. However, you can override those settings (and more) on a per-branch basis in each plan branch's configs.

Automatically merging branches

Since you don't really know if everything works until you've merged changes to (or from) main, Bamboo provides two ways to do that automatically in DVCS repositories. We call these options "Branch updater" and "Gatekeeper," and you can use either method on any plan branch. Before the build runs, Bamboo will merge the latest from main into the branch if you're using *branch updater*, or *checkout main* and merge changes up from your branch if you're using *gatekeeper*. (More on choosing a merge model in the