Bhanu Kiran Botta

SVN to Git Migration on AWS SYSTEM

Version 1.0

Contents

[Introduction 2](#_Toc478985878)

[Git’s features](#_Toc478985879) 2

[Benefits of GIT over SVN 3](#_Toc478985880)

[Migrating from SVN to GIT 3](#_Toc478985881)

[DEMO 3](#_Toc478985882)

**Introduction**

**Repositories**:

As SVN is CVCS (Centralized Version Control System) tool, for each project there is a single repository at a detached central place where all the history is and which we checkout and commit into.

Whereas GIT is a distributed source control management, each copy of the project tree (we call that the working copy) carries its own repository around (in the .git sub-directory in the project tree root).

**URL:**

In SVN the URL identifies the location of the repository and the path inside the repository, so we organize the layout of the repository as trunk/ , branches/ , tags/directories.

Whereas in GIT, the URL is just the location of the repository, and it always contains branches and tags. One of the branches is default and named as the ‘master’.

**Branches:**

In SVN, branches are just additional folders containing a copy of the code base. When an SVN branch is merged, the branch has to be deleted to prevent an erroneous merge back to trunk in future. Tracing merge history in SVN is a serious problem.

Where as in GIT branches, have their own history and revision tree giving us explicit information from where it was forked, what commits have happened and if it has been merged to another branch or not. This makes branch history management much easier. Hence GIT maintains the entire branch history and is completely traceable.

**Benefits of GIT over SVN**

* GIT uses a light weight branching model that allows developers to start working on a new feature or fix a bug very quickly and very efficiently.
* The local commit model of GIT helps implementation of agile development.
* The 'pull requests' encourage excellent code review experiences and allow teams to enforce policies before the code actually moves to production.

**Types of Migrations:**

“Tip” Migration:

* Only the latest/recent/tip version of the code is moved to Git and the history remains on the old server. This is the recommended way of migration from SVN to Git for Mathews, as we need to move one of the branch currently named as ‘Trunk’ to a master branch in Git. We may need to leave our existing SVN code base to read-only, so that no more changes are moved to it even accidently.

“History” Migration:

* Tries to mimic history that is present in SVN to GIT. Not recommended as history translation may not happen accurately. (As history and branches are stored differently with in SVN & GIT). Can still be done, if client insists on persisting existing SVN history of that branch to be moved to GIT.

**Importing an SNN repository to GIT:**

**Prerequisites:**

* Get read-only user access for SVN repository.
* Gitsvn utility is available in local environment.
* Download svn-migration-scripts.jar
* Setup remote github server
  + Get Org.name, Teaam name and members from the respective teams.
  + Setup organization.
  + Setup Team and assign team members.
  + Create repository and push the code from SVN repo.

Git plugin + screenshots.

Benefits of Git over SVN

Migrating from SVN to GIT – using Tip migration:

* Sign in to Github with an enterprise account.
* Create new project within the account (one time creation, which would be project name).
* Create new repository within the project – the first ever code base to this repository woud be considered as the ‘master’ branch.
* Goto the AWS machine and perform a ‘git clone’ with the repository url copied. This creates an empty repository on this machine.
* We may navigate to the SVN repository and copy code base from ‘Trunk’ folder (this will become the ‘master’ branch within GIT) of the SVN current code base and paste it in this empty repository created.









