

SVN to GIT migration

Table of Contents

- [Introduction](#)
- [Steps](#)
- [Integrating with Ivy](#)
- [How to make Jenkins build green](#)

Introduction

This document lists the steps to migrate code base from svn to git repository. Migration includes whole code base along with commit history, tags and branches.

Steps

Follow below steps in order to get the code base migrated from svn to git:

1. Check out code which is to be migrated.

```
$ svn co <svn_url> <temporary_dir>
```

Example

```
$ svn co https://svn.ultradns.net/svn/services/neuxfr-client temp_neuxfr_client
```

2. SVN simply lists the username for each commit. Git's commits have much richer data, name and email. Hence, we should have a file with list of all SVN users and their corresponding Git name and emails.
3. Inside the above checked out directory, run the following command to retrieve svn users for the code base:

```
svn log -q | awk -F '|' '/^r/ {sub("^ ", "", $2); sub(" $", "", $2); print $2 =  
"$2" <"$2">"}' | sort -u > authors_temp.txt
```

4. That will grab all the log messages, pluck out the usernames, eliminate any duplicate usernames, sort the usernames and place them into a "authors_temp.txt" file. Now edit each line in the file. For example, convert:

awilkins = awilkins <awilkins>

into

awilkins = Wilkins, Alan <johnalan@example.com>

5. Check out the script used for migration on the box.

```
svn co https://svn.ultradns.net/svn/svn2git/
```

6. Change working directory to svn2git.

```
cd svn2git/
```

7. Three files will be present inside this directory.

```
[root@Impetus-651 svn2git]# ls
authors.txt  svn2git.conf  svn2git.sh
```

8. svn2git.conf have following parameters:

- a. **SVN_REPO_URL** : This represents the absolute path of code base on svn.
Sample value: <https://svn.ultradns.net/svn/services/neuxfr-client>
- b. **GIT_REPO_URL** : This represents the path of project on git to which code will be migrated.
Sample value: <ssh://git@git.nexgen.neustar.biz:8022/ultradns/neuxfr-client.git>
- c. **SVN_REPO_NAME** : This represents svn repository name.
Sample value: neuxfr-client
- d. **GIT_REPO_NAME** : This represents git repository name.
Sample value: neuxfr-client

9. Ensure above parameters are accordingly set.

10. Create an empty project <GIT_REPO_NAME> in GIT.

11. Append the user details from authors_temp.txt (obtained in 2nd step) to authors.txt avoiding duplicate entries.

12. Ensure ssh to git is possible from working box:

```
[root@Impetus-651 svn2git]# ssh git@git.nexgen.neustar.biz -p8022 -T
Welcome to GitLab, Khan, Mushkooor!
```

13. Now, we are ready to execute migration script, svn2git.sh.

```
./svn2git.sh
```

14. Execution of script may take some time. After execution, code base will be visible on git repo with complete commit history.

Integrating with Ivy

If your project is using Ivy with SVN, it will need a couple of additions to make it work with Ivy with Git. This is due to our Ivy support being shared across multiple projects.

1. Create a file .gitmodules in the root of your git directory. It should have the following content:

```
[submodule "lib/ivy"]
    path = lib/ivy
    url = ssh://git@git.nexgen.neustar.biz:8022/ultradns/ivy.git
```

2. Create a lib directory, and within the lib directory, create an ivy directory:

```
mkdir lib
cd lib
mkdir ivy
cd ivy
```

3. In the ivy directory, create a .gitignore file. It should have the following content:

```
runtime_ivysettings.properties
```

4. Add, commit, and push the .gitignore and the .gitmodules files
5. When you clone your repository, be sure to include the --recursive flag (See Jenkins section)
 - a. If you forget to include --recursive, run the following command BEFORE you add the .gitignore file (if you added it, delete it and re-add it after the command completes):

```
git submodule update
```

How to make Jenkins build green

After successfully migrating code from svn to git, we have to modify corresponding jenkins job to pull code-base from git instead of svn.

Following points should be kept in mind while modifying Jenkins job:

- Update 'build.xml' in git to remove dependency on calculating svn revision.
- Ensure deploy keys for hudson user is enabled in migrated code in git.
- Ensure following command runs successfully on jenkins build box for hudson user:

```
-bash-3.2$ ssh git@git.nexgen.neustar.biz -p8022 -T
Welcome to GitLab, Anonymous!
```

- If the above command gives permission denied, check following two files.

```

$cat ~/.ssh/config
Host *
    IdentityFile ~/.ssh/jenkins.id_rsa

$cat ~/.ssh/jenkins.id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAuR+Z50LKfFfydRiQUUrlXq0VYg8rsf555jnk+Dml8PcVlsqZ
z2QX9YpFz0ZbWK5ZB6pGQ2FI/qCd2of2kDMhtSxxDsgLT5PqKdfoUxoT7cS/qDjJ
IXR65KFLAF1TtQvAjqz38gWVUZHg6wip1gYVQtyfDNQchXMfT8equRwT36gb+S4A
f/Hl4U5WJilI6IGpTrFDBdoadoaGU+mYKCX8fy2k3wIQfh09zMhTf7CUDjbympxx
nHnnDDMk6w3NCpE122se0XovfONAF145IqDWHHJmSYDEpfhogTQ9rkSfkHEvyz8D
b0exQKL9U9gNWw84aGMq4y84fXKzntlVivIEuQIBIwKCAQEAjs88Ng7z+YWsaPWx
KMS/ki3ACdEwVhU5M2TYGb8Ws08ikSFSDdFrSnA0xkEo38uvMUu44WQFOJN4xht
v7JqddHTjw9ZLsKQEaaOpoHU2/4tcyR95puvQqhj27WlmkrlAFmMENlV7nfZZNrM
KMLdM5RALne+Ow+qcL6ZpL3jcgQE9djKHlJ5zaOyUt7vh2JEr4FWRiHJn9g8Itl0
M/KQxBbxmSOSsj7QopcUwEqMDysIYYOjNLh0lN7DRdIoCPGJ1AxFb0lhgWccnF+
rb/hBASFGXkbheIcoTbycIEleLALoNkzZnM69sXXb8q0ipCdZ7WW0VnGOBRiJDwm
RfVhCwKBgQD2XZBPtmDBJW+r3Fkvq2zfOVuNJ0EVIDYe/X7n02Ok9uOWeqr2NDGe
ALLO70icV3DB1X7tXRE5V8LKJBUo2NIUbVcjI07dJS232xtrrOJxbI7vlyv9dZ3C
FN82skB769jnkZ69ggjEgQanPsUEVa6qNwfoig2+x+XJLx2LSeV1iQKBgQDAXOz2
pJwBqIrbJwlG85LHt1lcLks72eBrfZJGua4UTXLukVvFwJ+YZ2GG+FoFOO9nAdx
Yu8EO39l7uPwt33LOx3PA8RF2WJoGK+t/M8VyjxrWGvXOAldWydTbt2CCXaVifm
2jfGPVmbLtnhWzzLGp9F9Gq7IQc3EVrbAt55sQKBgQDhP5nietweThz05rfxEcKg
NHD2FULRfIlA58R8LvwEifSmlo2t5pPDqN3/BqjJdIRZcr0wyh5gUD0QlgS3oaoh
TgaGhq6BDAYKlvR/tAI70Pe2e5XvDHL6lr1zlixUCtTw+iq75JpNQsRBI3JNGyNZ
y+n5LcNsmYGH/zD0YNHkiwKBgDF2+xrZ3wL6ZDcNTLkVx1UlMaHbKEaylvV14D6v
JWKYi0SpCPhWgmKEpAzDhNzFf2OhZiVYghnEiGGkmJ4U4wH4xf5p3+s7Ws5jEKu
uOhYkzGLw/u4y/B9R9Ct1pqxo1VmnIvRtpIPxo7uyKQBhK1BXCfuZJaE1ExU63ol
TyabAoGBAJSM4RIndZlo2Up+Ljakc9jK9H7tr5thU96fiyHy+O8mf36u1BZScHy1
olFk697RFLJZQVYHhN8zOlHaoiGaIhqMG0PxW8L//kwuF2baM6/OLgVBRPMIdGj
08cFQyJy1NEm0lyVgASszmVLvD6PQvYnKyZzoim0ckJnHnY1f8nGn
-----END RSA PRIVATE KEY-----

```

- In order to get your repo from Git, we rely on a small shell script. In Jenkins build, go to configure.
- Under Source Code Management, select "None"

Source Code Management

- ☐ CVS
- ☒ Git
- ☐ Google Code (automatic configuration)
- ☐ Mercurial
- ☒ None
- ☐ Subversion
- ☐ Subversion Release
- ☐ URL Copy

- Under Build, add a build step "Execute Shell". If there are other items under Build, drag Execute Shell so that it is the first item. Put the following commands into the text area:

Execute shell

Command

```

set -e

rm -rf *;
rm -rf .??* ;

GIT_TRACE=2 git clone -b master --recursive ssh://git@git.nexgen.neustar.biz:8022/ultradns/signer.git .

```

See the list of available environment variables

Delete

- Use the URL for your repository instead of the URL for signer. As of this writing, the firewall ports were being opened so that the IP address for git.nexgen.neustar.biz will no longer be needed.
- In order to trigger Jenkins build automatically (on git push), we have to create a web hook for migrated project in git. Steps for creating web hook is as follows:
 - In Jenkins build, goto configure. Under "Build Triggers" section, select "Trigger builds remotely". Give an "Authentication Token".
 - Click here to view an example...

Build Triggers

☐ Build after other projects are built
 ☒ Trigger builds remotely (e.g., from scripts)

Authentication Token

Use the following URL to trigger build remotely: JENKINS_URL/job/ultra-dto-git/build?token=TOKEN_NAME or /buildWithParameters?token=TOKEN_NAME
 Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.

☐ Build periodically
 ☐ Build when a change is pushed to GitHub
 ☐ Build when another project is promoted
 ☐ Poll SCM

Build Environment

☐ Delete workspace before build starts
 ☐ Send files or execute commands over SSH before the build starts

- In git, goto project settings >> "Web Hooks". Specify a url and click "Add Web Hook". URL should be of following format:

```
jenkins_url/job/job-name/build?token=token_name_given_in_jenkins_job
```

For e.g:

```
http://stulhdebuild3.ultra.neustar.com:8080/job/ultra-dto-git/build?token=ultra-dto
```

where,

```
http://stulhdebuild3.ultra.neustar.com:8080 = jenkins_url
ultra-dto-git = job-name
ultra-dto = Token name given in jenkins job
```

Click here to view example...

ultradns / ultra-dto

Search in this project

Project

Files

Commits

Network

Graphs

Issues 0

Merge Requests 0

Wiki

Settings

Project

Members

Deploy Keys

Web Hooks

Services

Protected branches

Web hooks

Web hooks can be used for binding events when something is happening within the project.

URL

Trigger

☒ Push events
This url will be triggered by a push to the repository
 ☐ Tag push events
This url will be triggered when a new tag is pushed to the repository
 ☐ Issues events
This url will be triggered when an issue is created
 ☐ Merge Request events
This url will be triggered when a merge request is created

Add Web Hook

Web hooks (1)

http://stulhdebuild3.ultra.neustar.com:8080/job/ultra-dto-git/build?token=ultra-dto

Test Hook Remove

Push Events

- Click on "Test Hook". Build should be triggered. This implies that whenever code is pushed into git repository, build will be triggered automatically.