

CASE STUDY

1. CHECK YOUR CURRENT WORKING DIRECTORY

```
Select juhi@05d861eca28c5c4: ~/my-working-copy  
juhi@05d861eca28c5c4:~$ pwd  
/home/juhi
```

2. BEFORE STATRING THE MIGRATION ENSURE YOU HAVE THE FOLLOWING INSTALLED:

GIT

SUBVERSION(SVN)

GIT-SVN

```
juhi@05d861eca28c5c4:~$ sudo apt-get install git-svn  
[sudo] password for juhi:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
git-svn is already the newest version (1:2.43.0-1ubuntu7.2).  
0 upgraded, 0 newly installed, 0 to remove and 59 not upgraded.
```

```
juhi@05d861eca28c5c4:~$ git svn --version  
git-svn version 2.43.0 (svn 1.14.3)
```

3. CREATING AN EMPTY DIRECTORY FOR YOUR NEW GIT REPOSITORY AND INITIALIZE IT:

```
juhi@05d861eca28c5c4:~$ mkdir new-git-repo1  
juhi@05d861eca28c5c4:~$ cd new-git-repo1/  
juhi@05d861eca28c5c4:~/new-git-repo1$ git init  
hint: Using 'master' as the name for the initial branch. This default branch name  
hint: is subject to change. To configure the initial branch name to use in all  
hint: of your new repositories, which will suppress this warning, call:  
hint:  
hint:   git config --global init.defaultBranch <name>  
hint:  
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and  
hint: 'development'. The just-created branch can be renamed via this command:  
hint:  
hint:   git branch -m <name>  
Initialized empty Git repository in /home/juhi/new-git-repo1/.git/
```

4. INSTALL SUBVERSION(SVN) TO CREATE A SVN REPOSITORY

```
juhi@05d861eca28c5c4:~/new-git-repo1$ sudo apt-get install subversion
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
subversion is already the newest version (1.14.3-1build4).
0 upgraded, 0 newly installed, 0 to remove and 59 not upgraded.
```

5. CREATE A SVN REPOSITORY AND REPOSITORY STRUCTURE

```
juhi@05d861eca28c5c4:~/new-git-repo1$ svnadmin create my-svn-repo1

juhi@05d861eca28c5c4:~/new-git-repo1$ ls /home/juhi/new-git-repo1/my-svn-repo1
README.txt  conf  db  format  hooks  locks
juhi@05d861eca28c5c4:~/new-git-repo1$ svn mkdir file:///home/juhi/new-git-repo1/my-svn-repo1/trunk -m "
Creating trunk"
file:///home/juhi/new-git-repo1/my-svn-repo1/branches -m "Creating branches"
svn mkdir file:///home/juhi/new-git-repo1/my-svn-repo1/tags -m "Creating tags"
Committing transaction...
Committed revision 1.
juhi@05d861eca28c5c4:~/new-git-repo1$ svn mkdir file:///home/juhi/new-git-repo1/my-svn-repo1/branches -
m "Creating branches"
Committing transaction...
Committed revision 2.
juhi@05d861eca28c5c4:~/new-git-repo1$ svn mkdir file:///home/juhi/new-git-repo1/my-svn-repo1/tags -m "C
reating tags"
Committing transaction...
Committed revision 3.
```

6. Create a test directory called my-working-copy and add files like test-file.txt and file.txt.

7. Add the new files to svn and commit the changes using svn add and svn commit commands.

8. Verify the changes using svn log command

9. Checkout the repository to work locally

```
juhi@05d861eca28c5c4:~/new-git-repo1$ mkdir /home/juhi/my-working-copy
juhi@05d861eca28c5c4:~/new-git-repo1$ cd /home/juhi/my-working-copy/
```

```
juhi@05d861eca28c5c4:~/my-working-copy$ ls /home/juhi/new-git-repo1/my-svn-repo1
README.txt  conf  db  format  hooks  locks
juhi@05d861eca28c5c4:~/my-working-copy$ svn list file:///home/juhi/new-git-repo1/my-svn-repo1
branches/
tags/
trunk/
juhi@05d861eca28c5c4:~/my-working-copy$ svn checkout file:///home/juhi/new-git-repo1/my-svn-repo1/trunk
/home/juhi/my-working-copy
Checked out revision 3.
juhi@05d861eca28c5c4:~/my-working-copy$ cd /home/juhi/my-working-copy
```

```
juhi@05d861eca28c5c4:~/my-working-copy$ echo "This is a test file" > test-file.txt
juhi@05d861eca28c5c4:~/my-working-copy$ svn add test-file.txt
A      test-file.txt
juhi@05d861eca28c5c4:~/my-working-copy$ svn status
A      test-file.txt
juhi@05d861eca28c5c4:~/my-working-copy$ svn commit -m "Added test-file.txt"
Adding      test-file.txt
Transmitting file data .done
Committing transaction...
Committed revision 4.
```

```

juhi@05d861eca28c5c4:~/my-working-copy$ svn log file:///home/juhi/new-git-repo1/my-svn-repo1/
-----
r4 | juhi | 2025-01-31 10:16:32 +0000 (Fri, 31 Jan 2025) | 1 line
Added test-file.txt
-----
r3 | juhi | 2025-01-31 10:09:04 +0000 (Fri, 31 Jan 2025) | 1 line
Creating tags
-----
r2 | juhi | 2025-01-31 10:09:04 +0000 (Fri, 31 Jan 2025) | 1 line
Creating branches
-----
r1 | juhi | 2025-01-31 10:09:04 +0000 (Fri, 31 Jan 2025) | 1 line
Creating trunk
-----
juhi@05d861eca28c5c4:~/my-working-copy$ echo "Another Change!" >> file.txt
juhi@05d861eca28c5c4:~/my-working-copy$ svn add file.txt
A      file.txt
juhi@05d861eca28c5c4:~/my-working-copy$ svn commit -m "Made changes to file.txt"
Adding      file.txt
Transmitting file data .done
Committing transaction...
Committed revision 5.

```

```

juhi@05d861eca28c5c4:~/my-working-copy$ svn log file:///home/juhi/new-git-repo1/my-svn-repo1
-----
r5 | juhi | 2025-01-31 10:19:29 +0000 (Fri, 31 Jan 2025) | 1 line
Made changes to file.txt
-----
r4 | juhi | 2025-01-31 10:16:32 +0000 (Fri, 31 Jan 2025) | 1 line
Added test-file.txt
-----
r3 | juhi | 2025-01-31 10:09:04 +0000 (Fri, 31 Jan 2025) | 1 line
Creating tags
-----
r2 | juhi | 2025-01-31 10:09:04 +0000 (Fri, 31 Jan 2025) | 1 line
Creating branches
-----
r1 | juhi | 2025-01-31 10:09:04 +0000 (Fri, 31 Jan 2025) | 1 line
Creating trunk
-----

```

10. IMPORT THE FILES FROM TEST DIRECTORY TO YOUR SVN REPOSITORY AND VERIFY THE CHANGES

```

juhi@05d861eca28c5c4:~/my-working-copy$ svn list file:///home/juhi/new-git-repo1/my-svn-repo1
branches/
tags/
trunk/

```

11. NOW YOU WILL USE SVN-GIT TO CLONE THE SVN REPOSTORY INTO YOUR GIT REPOSITORY NEW-GIT-REPO

```

juhi@05d861eca28c5c4:~/my-working-copy$ git svn clone file:///home/juhi/new-git-repo1/my-svn-repo1 --std
layout /home/juhi/new-git-repo1
r1 = 2d5a3dd9c410e1a4bb0a6b6c45116d0bd98a5362 (refs/remotes/origin/trunk)
A      test-file.txt
r4 = 1580cc1d4e06501e3ac200ca675c640a3580d82a (refs/remotes/origin/trunk)
A      file.txt
r5 = dd3628e7600cab80461bcf12cabd9ae9fa7d10c7 (refs/remotes/origin/trunk)
Checked out HEAD:
  file:///home/juhi/new-git-repo1/my-svn-repo1/trunk r5
juhi@05d861eca28c5c4:~/my-working-copy$

```

