**AWS VM DETAILS WHICH ARE USED FOR MIGRATION AND VERIFICATION -**  
  
**AWS VM used for SP migration -**  
  
IP - 10.220.117.84

User - ec2-user

Key -

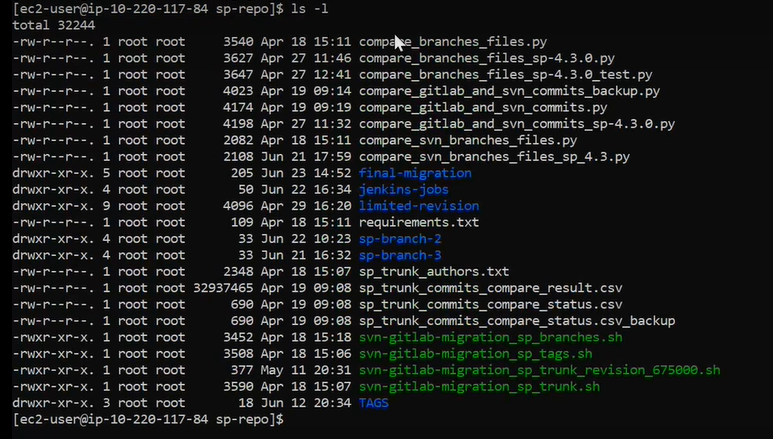
* SP repository data is present at - /repo/sp-repo
* And fmc\_infra repository data is present at - /repo/fmc-repo

So if you go inside /repo/sp-repo

Cd /repo/sp-repo

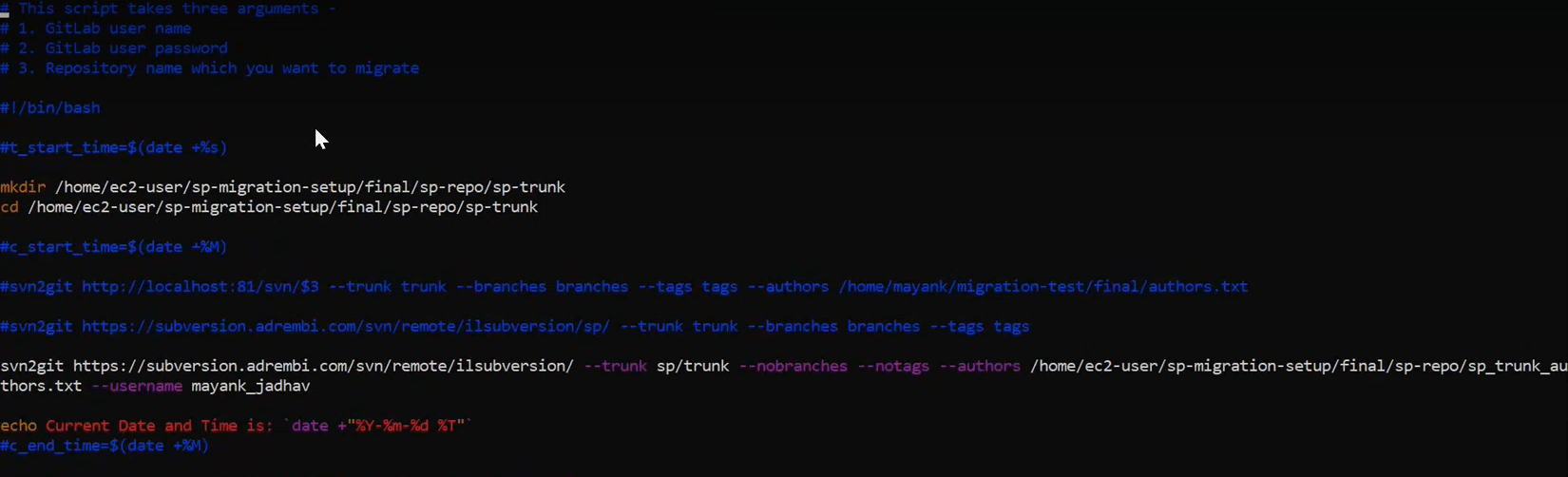
ls -la

you will see some scripts,. Trunk,branch and tags …etc .



So if see script for trunk migration u can see like

Cat svn-gitlab-migration\_sp\_trunk.sh



First it will create directory in the mentioned path,then it will navigate to that path,then it will execute svn to git convert command with the help of author.txt fil(on the above command only we are migrating trunk so mentioned trunk path and also mentioned wht not migrate like tag and branches)

After exexute the above command svn file will cloned to our machine under trunk directory which we created earlier.



below mentioned

author.txt file is used for the mapping of SVN and GIT user.

If you do cat author.txt ,its looks like below .



Then add you GIT to your migration machine using below command.

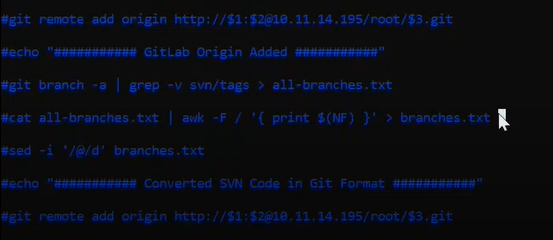
Git remote add origin master



$1 for GITLAB user name

$2 for GITLA PASSWORD

$3 for repository url



After u push the code to GIT ,its time to verify using below scipt.

**Compare\_branches\_files.py**

If u do cat here : **cat Compare\_branches\_files.py**

**In SP and fmc\_infra repository we have 3 components -**

1. Trunk (master in git)
2. Branches
3. Tags

**SP Repository SVN URL** - https://subversion.adrembi.com/svn/remote/ilsubversion/sp/

**SP Trunk (master branch on git and also known as 5.0.0.0 branch)-**

We have used “git svn” tool to convert svn code in git format and then pushed it on gitlab.

Git svn tool - <https://git-scm.com/docs/git-svn>

**SP Trunk gitlab url** - https://tlvgit03.nice.com/fmc-rd/sp/sp/-/tree/master

**SP Tags -**  
We haven’t converted the SP Tags in .git format, we have migrated the SP tags in .svn format to Gitlab and we have migrated 56 tags.  
Steps taken -

1. Clone the SP tags on VM from SVN
2. Add the remote repository of Gitlab and push the tags as branches

**SP Tags gitlab url** - https://tlvgit03.nice.com/fmc-rd/sp/sp-tags

**SP Branches -**

We have used “git svn” tool to convert svn code in git format and then pushed it on gitlab.

Git svn tool - <https://git-scm.com/docs/git-svn>

**SP Branches gitlab url** -  
**sp-4.2.0** - <https://tlvgit03.nice.com/fmc-rd/sp/sp/-/tree/sp-4.2.0>

**sp-4.3.0 -** https://tlvgit03.nice.com/fmc-rd/sp/sp/-/tree/sp-4.3.0

**SP Total** -  
Trunk – 1

Branches – 2

Tags – 56

**Fmc\_infra repository SVN url** - https://subversion.adrembi.com/svn/remote/ilsubversion/fmc\_infra/

**Fmc\_infra Trunk -**

We have used “git svn” tool to convert svn code in git format and then pushed it on gitlab.

Git svn tool - <https://git-scm.com/docs/git-svn>

**Fmc\_infra Tags -**  
We have used “git svn” tool to convert svn code in git format and then pushed it on gitlab.

Git svn tool - <https://git-scm.com/docs/git-svn>

**Fmc\_infra Branches -**

We have used “git svn” tool to convert svn code in git format and then pushed it on gitlab.

Git svn tool - <https://git-scm.com/docs/git-svn>

**Fmc\_infra Total** -  
Trunk – 1

Branches – 0

Tags – 3

**AWS VM used for CS migration -**  
  
IP - 10.220.117.203

User - ec2-user

Key - <https://nesstech-my.sharepoint.com/:u:/g/personal/v7500231_ness_com/ERWt33_5ATVMoB0fAshDawYBbXNNlWeUCDJcLJpwdOHEqA?e=9S4oeI>

* All the repositories data is present at - /home/ec2-user/CS-SVN

**In CS, we have 10 repositories which need to migrate on Gitlab and we have 4 components in approximately all the repositories -**

1. Trunk (master in git)
2. Branches
3. Tags
4. Builds -> As mentioned by Nice CS Team(Hitesh), we don’t need to migrate builds on gitlab.

**CS SVN Repository’s URL** - <https://nesstech-my.sharepoint.com/:x:/g/personal/v7500231_ness_com/EfNf3P6pMsFMoyIVu9a726ABDZP1j9RvtmKrRBINQQ7gBQ?e=0PUY9K>

I have used “git svn” tool for migrating the repositories from SVN to Gitlab.

I have executed the command with respect to components(Trunk, branches and tags).  
  
Command -  
sudo git svn clone CS\_SVN\_REPOSITORY\_URL --COMPONENT\_NAME=COMPONENT\_PATH --preserve-empty-dirs --username USERNAME

Example – 1. sudo git svn clone https://svn.cssrd.local/CyberTech/BuildServers/build-farm/ --trunk=trunk --preserve-empty-dirs --username bbro

In the above example I have added the url of CS SVN repository Build Farm, I’m cloning the trunk component that’s why mentioned trunk, another flag –preserver-empty-dirs to clone the empty folders and last one is –username which is bbro and it has access on CS SVN repository.

2. sudo git svn clone https://svn.cssrd.local/CyberTech/BuildServers/build-farm/ --branches=branches --preserve-empty-dirs --username bbro

You can find the verification scripts here - https://github.com/mayank-niyuj/verification-scripts

**STEPS TO PUSH THE CODE ON GITLAB ONCE THE REPOSITORY IS CLONED ON THE VM -**

1. Move to the directory where repository is cloned
2. Check the branches with command “git branch –a”
3. Add the gitlab remote origin with command “git add remote origin repository\_url”
4. If you want to push master branch then run the command “git push origin master” to push the code of master branch
5. If you want to push some other branch then first checkout to that branch with command “git checkout –b branch\_name” and then push that branch with command “git push origin branch\_name”
6. After pushing the code, check the branches, tags on gitlab ui
7. Atlast execute the verification scripts
8. In verification scripts you have to update the gitlab project id and svn repository url.

NOTE: After pushing the branches of SP repository on Gitlab, we have updated the root pom.xml, as suggested by the NICE DevOps team. In the pom.xml, we have updated svn to git.  
Below I have attached a snapshot from which you can understand this -



Apart from this I have updated the scm too -

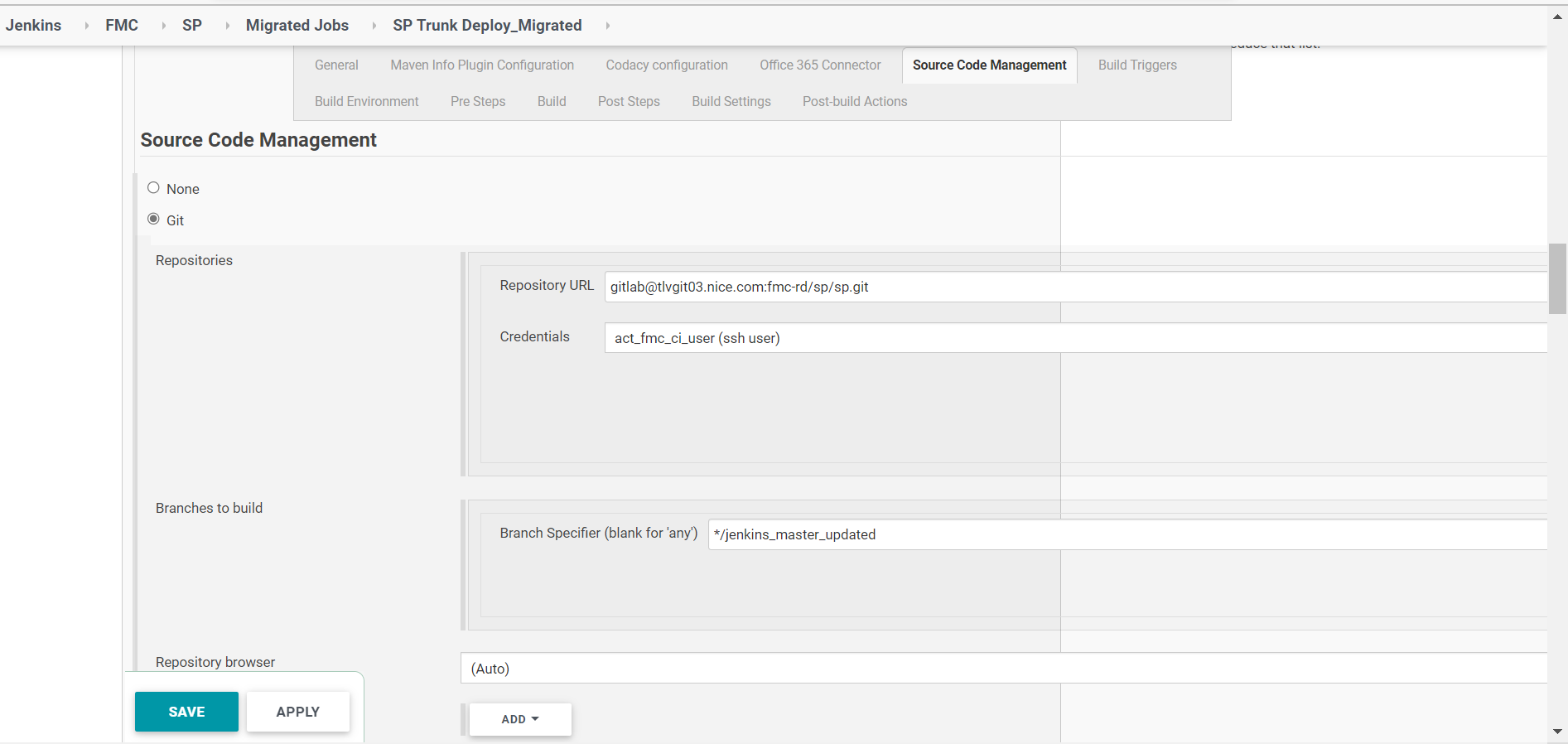


**SP JENKINS JOBS -**  
1. We have 16 Jenkins jobs - <https://nesstech-my.sharepoint.com/:x:/g/personal/v7500231_ness_com/EYo4GNyX3fRDvaB2UqEfGucBsXnkSCekcnBMHhRQHYGYPw?e=wAkaxR>

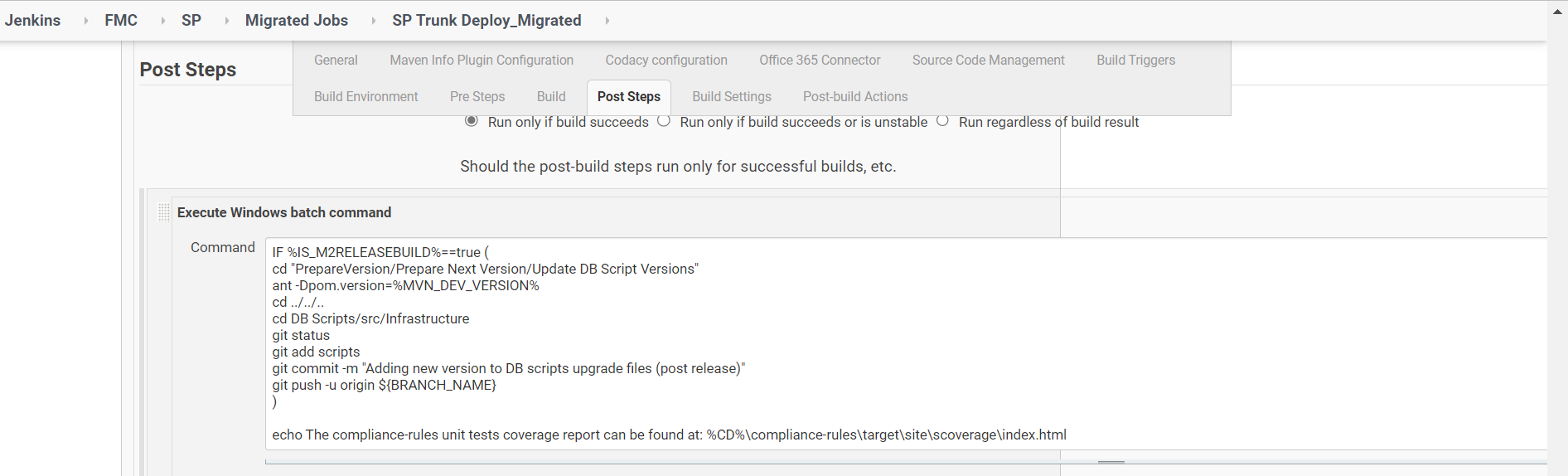
2. We have cloned these 16 Jenkins jobs from the existing jobs

3. In this jenkins jobs we have updated the “Source Code Management” from Subversion to Git

4. After changing the “Source Code Management”, we have updated the “Repository URL” and “Branch Specifier”



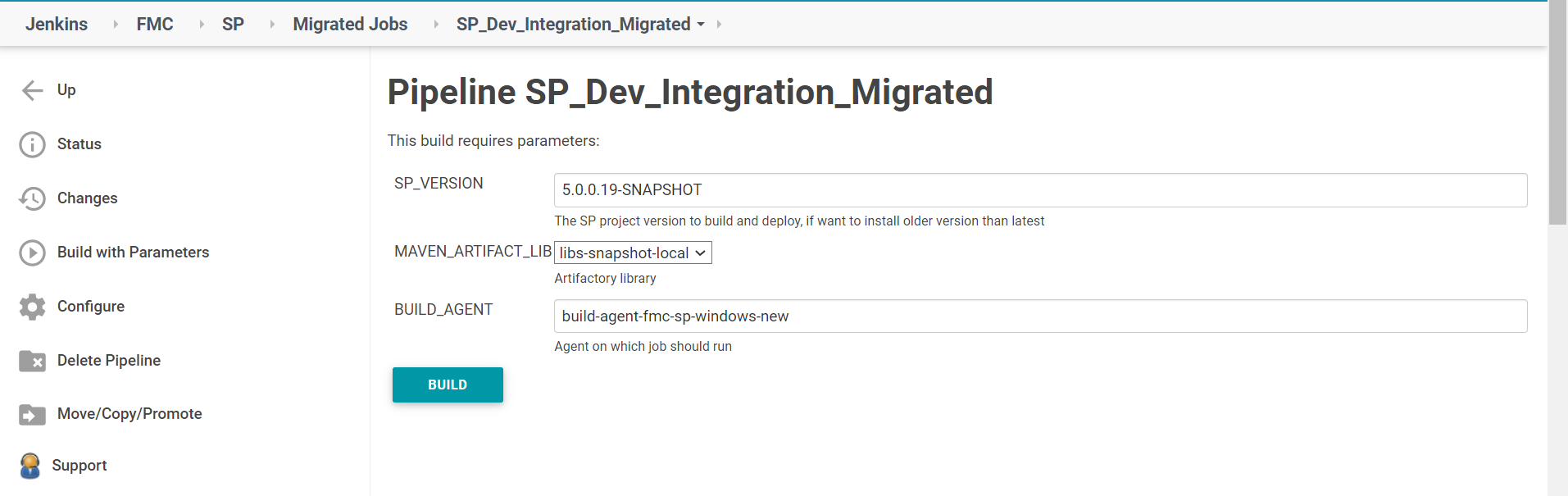
5. We have also updated the “Execute Windows Batch Command” section in Post steps, earlier svn reference is mentioned there but then we have updated it with respect to git.



6. Following jobs creates an Artifact -

SP Trunk Deploy\_Migrated, SP 4.2.0 Deploy\_Migrated and SP 4.3.0 Deploy\_Migrated

7. Following jobs consumes artifacts created from above jobs and executes automatically once the above jobs executes succesfully. You can check the post build actions of above jobs.



We have to pass the parameter while triggering these jobs, the parameter name is “SP\_VERSION” and you have to enter the name of the snapshot created from above jobs -

SP\_Dev\_Integration\_Migrated, MSSQL\_Dev\_Integration\_Migrated, SP 4.2.0 Oracle Integration Test\_Migrated, SP 4.3.0 Oracle Integration Test\_Migrated and SP\_4.3\_MSSQL\_Dev\_Migrated.