**CS Migration to GITLAB and trigger the JOB using Jenkins**

**Step-1**

**Migrate Repos from Bitbucket to Gitlab**

Understanding: There are N number of Bitbucket repos in BITBUCKET. Migrate those repos from BITBUCKET to Gitlab.

Approach:

1. Gitlab provides an in-built functionality to migrate the repos from Bitbucket to Gitlab. We will use the import project functionality and setup the Gitlab changes to mirror/track the changes in the Bitbucket repo.
2. After every migration/mirroring is done, we validate the following in the Gitlab and BITBUCKET:

* Users
* Commits
* Repo structure
* Files
* Branches

To validate the above details, we have created multiple Python Scripts as the scale of the repos increases validating the repo integrity manually will become more and more error prone.

Phase 1:

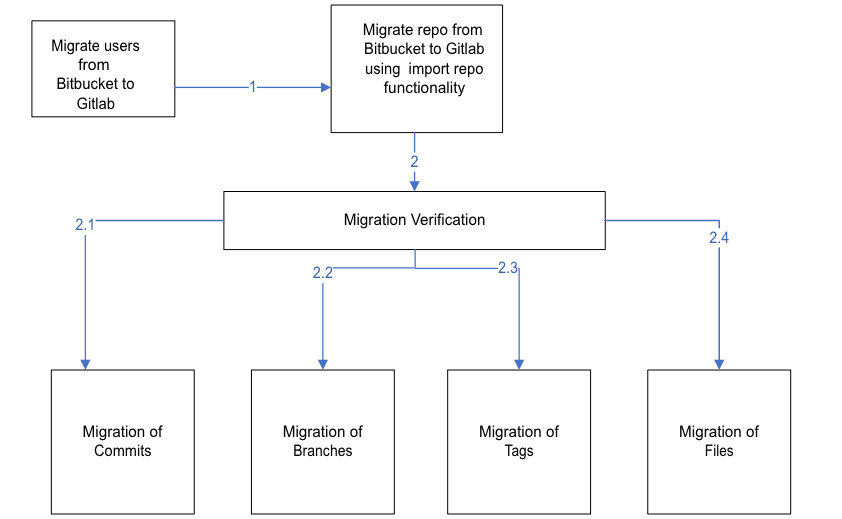
1. Migrate the users from Bitbucket to Gitlab. The migration of users' needs to be completed before the repo migration is started. If the users have performed actions on the Repo and if those users are not present in the Gitlab beforehand then there will be discrepancies in the Gitlab Repo
2. Migrate 20% of the available repos. Start with a sample repo. Migrate that repo successfully to Gitlab
3. Run the verification scripts to verify whether the components of the repos are migrated successfully or not.
4. Migrate the jobs corresponding to the repos. As part of job migration, only the pointing of the Jenkins job needs to be changed from BITBUCKET to Gitlab

Assumption:

1. The infrastructure setup like installation of Jenkins, Gitlab, JFrog will be done before hand
2. All the necessary access is provided
3. A backup/ DRR plan is put in place by the NICE IT team.

**Migration of repository from Bitbucket to GitLab -**

Steps of Migration -



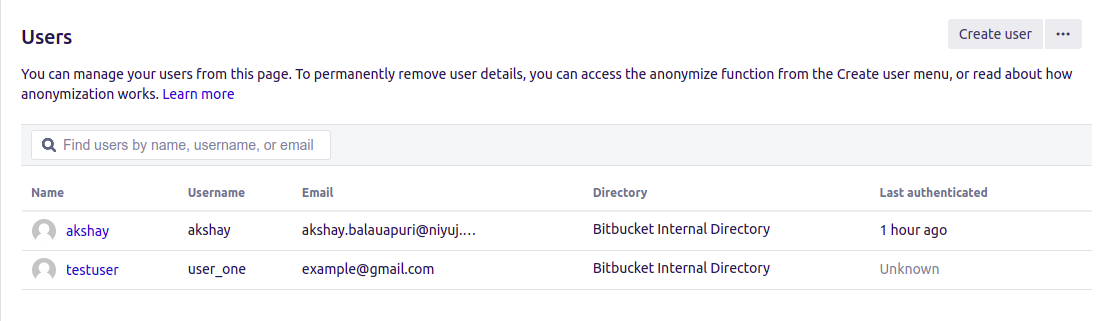
1. **Creating users on Gitlab:** Before migrating the project from Bitbucket to Gitlab, we need to ensure all the related users should be migrated from bitbucket to Gitlab. If any of the user is absent on Gitlab, there will be a discrepancy in the commit details, etc. Of the user in Gitlab.

We have created an automated python script to migrate the users ***migrate\_users.py.***

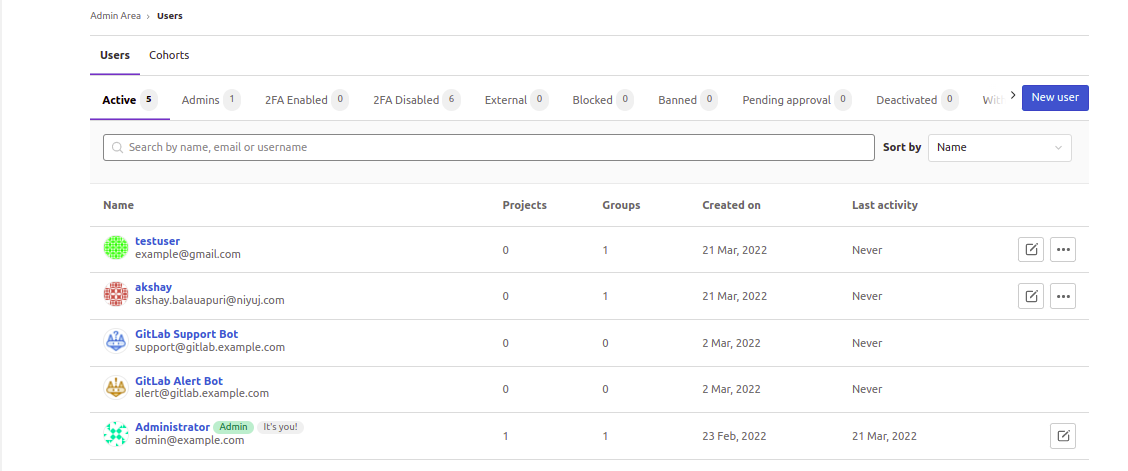
It will get the user details such as “Name”, “Username” & “Email Address” from Bitbucket & will create the same users over Gitlab.  
  
The user will also get created over the default instance group.  
  
If a user already exists over Gitlab, Script will give output as the user already exists & if a user doesn’t exist over GitLab server it will create user with all the details which script already fetched.  
   
As soon as the user gets created over Gitlab, the user’s will receive mail over their registered Email Address which is migrated from Bitbucket to Gitlab. The user can also use the same email address to reset their password.  
  
Note: SMTP server must be enabled in Gitlab, so that each user can create their own password.

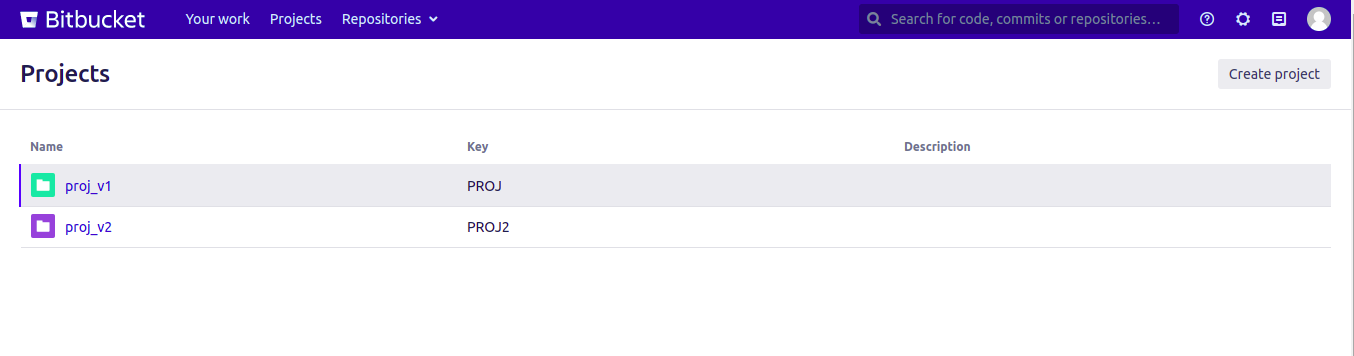
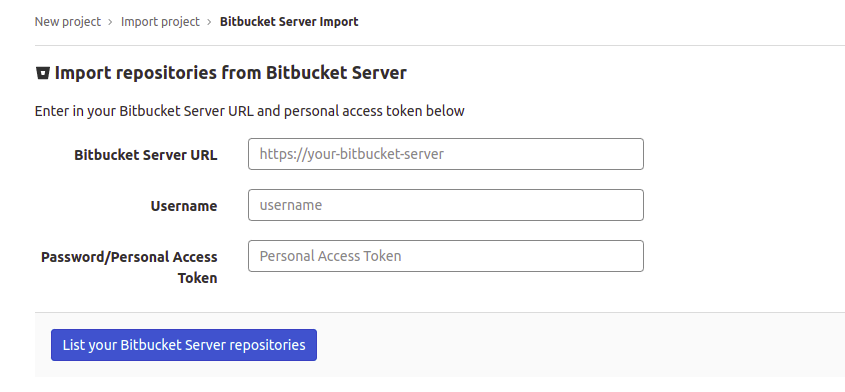
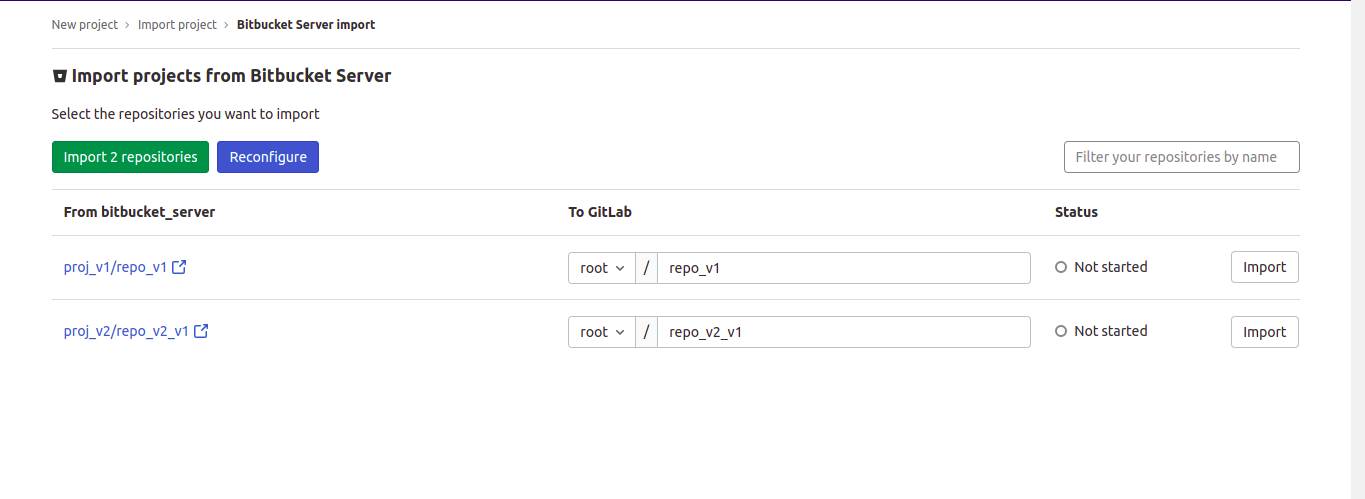
**NOTE: Later we found that all the user already exist in GITLAB ,So we did not used any script to migrate ,just we given them permission to our Existing repo.**

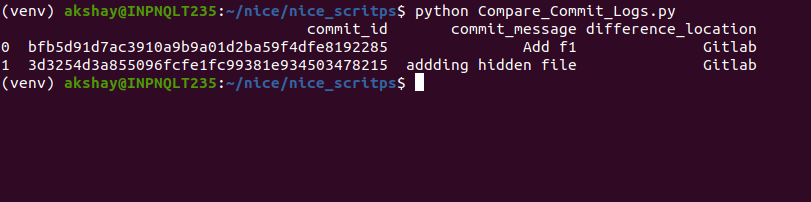
Bitbucket users: -

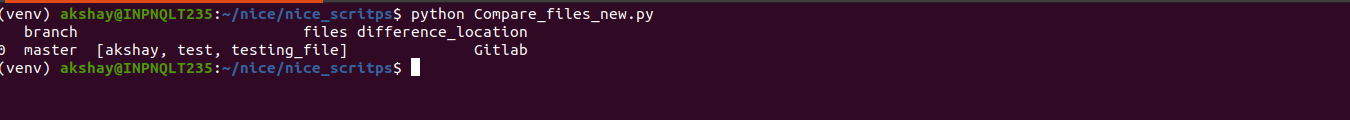
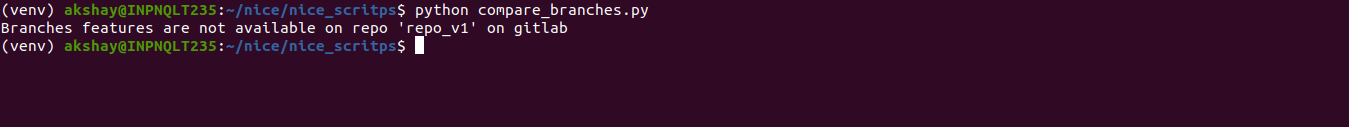


Gitlab users: -



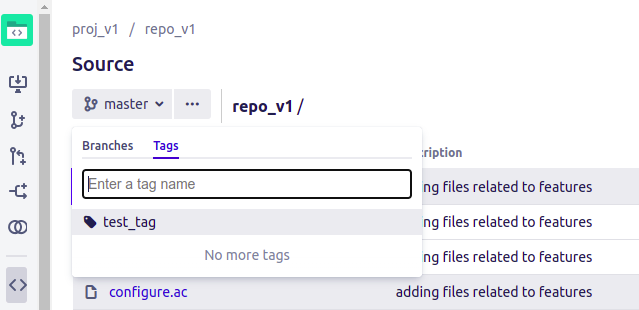
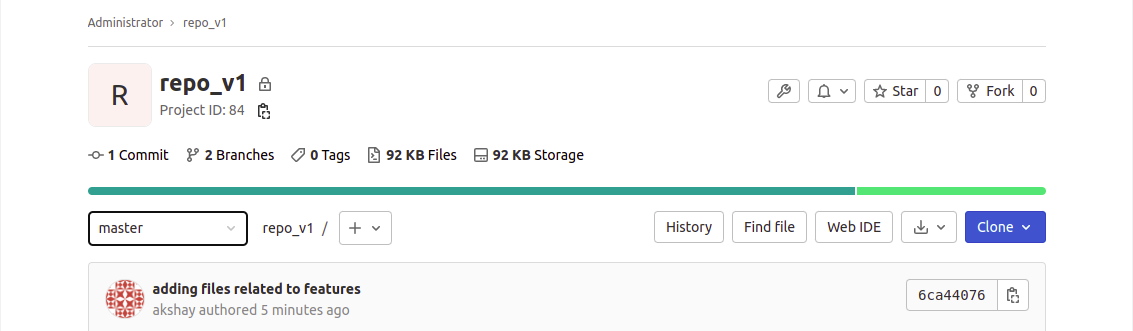
1. **Import Project from Bitbucket to Gitlab**:  
     
   2.1) These are the projects in Bitbucket.  
     
      
     
   2.2) We will use inbuilt functionality of Gitlab to Import Project from Bitbucket server: -  
     
     
     
   As Input, we are required with 3 parameters related to Bitbucket: -  
     
   A) Bitbucket Server URL  
   B) Username  
   C) PAT/Password  
     
     
   Once after providing all these parameters & clicking on ”List your Bitbucket Server repositories”. All the Repositories will get listed.  
     
     
   2.3) All project consists of “Import" button. Once clicking on Import button will import project from Bitbucket to Gitlab  
     
      
     
   Official Documentation: - <https://docs.gitlab.com/ee/user/project/import/bitbucket_server.html>NOTE: Import Bitbucket server project must be enabled. If it is disabled an administrator must have to enable the **Bitbucket Server** in **Admin > Settings > General > Visibility and access controls > Import sources**.  
     
   Note: Before Importing project make sure all the migrated users have activated their account.
2. **Migration Verification:**

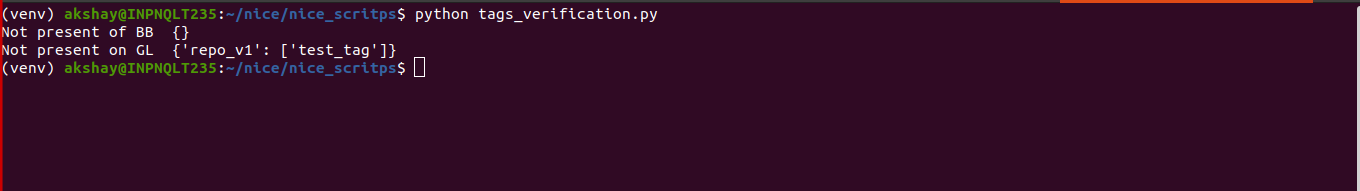
**3.1. compare\_commit\_logs.py**  
  
Once the project is imported successfully, we will verify whether there are any changes to either of the servers.  
  
For verification we will use the above script that is “Compare\_commit\_logs.py”.   
  
The script will verify the above content: -  
  
A) Commit ID  
B) Author name  
C) Email Address  
D) Timestamp of commit  
  
  
If there are any changes the output will reflect the changes and will also show the destination (where the changes are visible).  
  


**3.2. compare\_files.py**  
  
This script will compare the files over both the server.  
  
If Gitlab server is missing any file, the output will reflect the branch name & will showcase the data.  
  
  
**3.3. compare\_branches.py**  
  
This script will compare branches over both the sever. If any branch which exist over Bitbucket is not present on Gitlab, will show output as mention in above screenshot.  
  


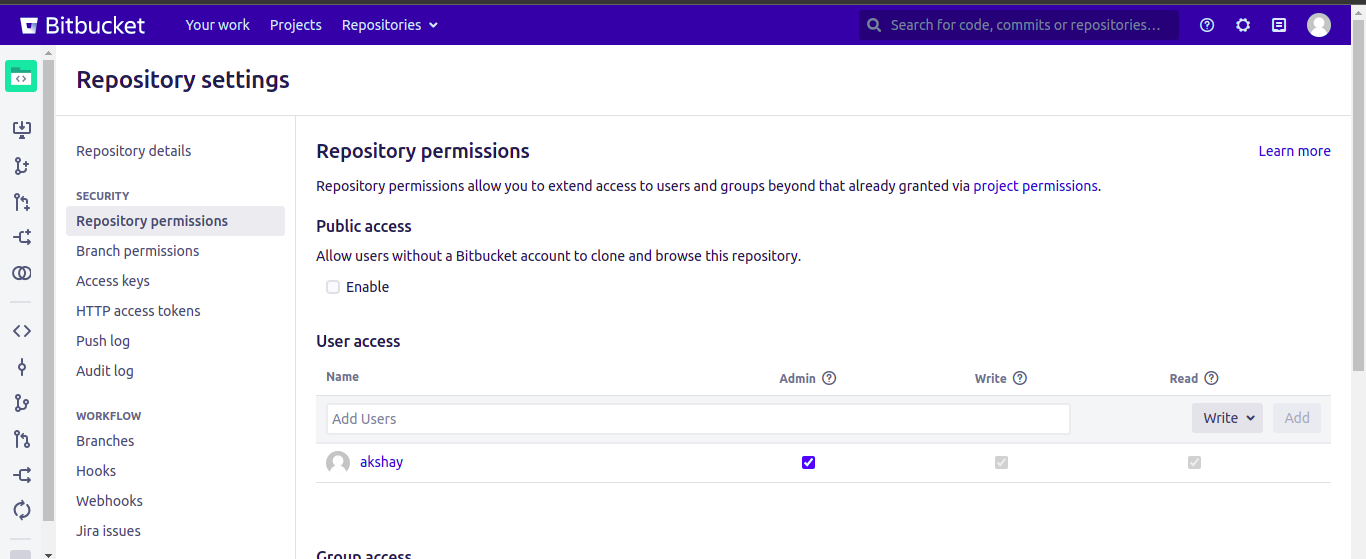
**3.4. tags\_verification.py**

As mentioned earlier tags are main feature which is being used for deployment.   
“tags\_verification.py” will compare the tags with respect to repository.  
If any tag of repository is present over Bitbucket and is not migrated to Gitlab then script will show the output and will list the repository of which tag is not migrated.  
  
 Bitbucket tag: -

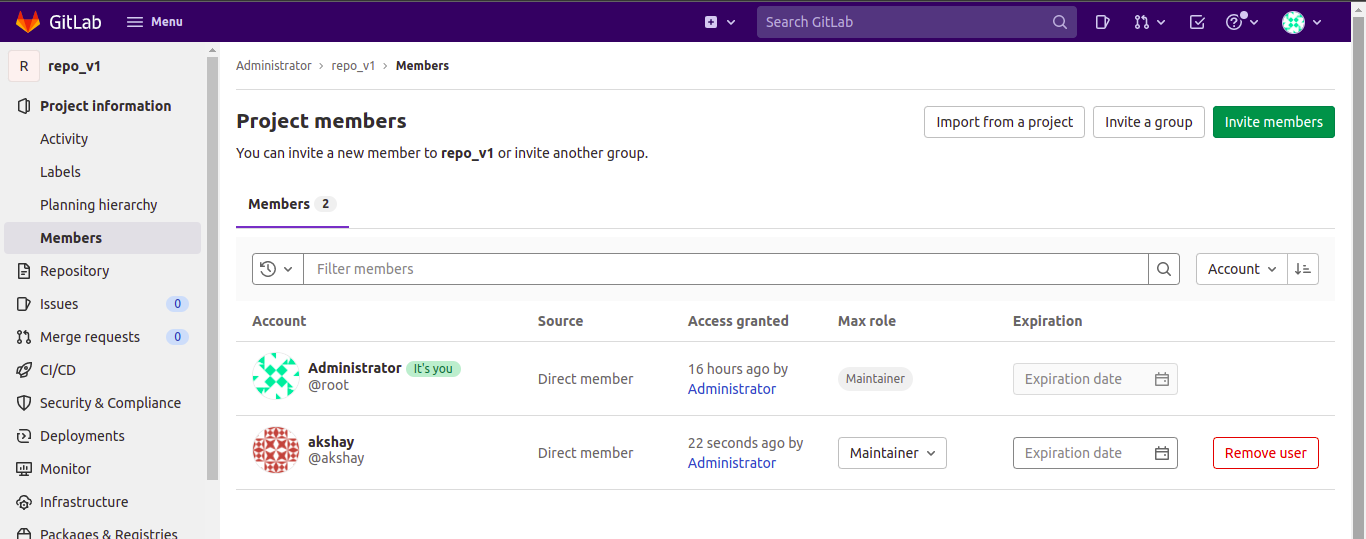
   
 GitLab tag: -  
  
  
Any tag which is not present over Gitlab will be shown as an output of tags\_verfication.py  
Example: As above screenshot Bitbucket repository consists of a single tag which is not present over Gitlab (shown in Gitlab Tag screenshot)  
  
Output: -



**3.5. user\_permission\_project.py**  
  
Bitbucket provides multiple types of access to the user. One of them is project-based access   
The user’s having access to multiple projects over the repository will get replicated same on Gitlab after running this script.  
  
For example: The user “akshay“ is having permission of admin over repo\_v1 repository. Same corresponding permission will get reflected over Gitlab repo\_v1.  
  
Any number of users having access to any repository on Bitbucket will get the same access over Gitlab.  
  
 Bitbucket Repo Access -



Gitlab Repo Access: -



**3.6 . groups.py**  
The script will check & get the names of groups inside each project. The same group will get created over GitLab in the same project.  
  
For example: - In Bitbucket repository “repo\_v1” consists of the group name “feature”. In Gitlab, once the project is imported & after running group.py the script will create same group “feature” for “repo\_v1”.