

# Standard Terraform Commit Procedure

Here's a walkthrough of the key points regarding merge requests, Atlantis, and Terraform:

Terminology:

- GitLab uses "merge request" while GitHub uses "pull request"

Atlantis workflow:

- When a merge request is created, Atlantis automatically runs a Terraform plan
- The plan output is posted as a comment on the merge request
- You can apply changes by commenting `atlantis apply` on the merge request
- Atlantis will then run terraform apply to make the approved changes

Locking mechanism:

- Atlantis uses Terraform's built-in state locking
- When a plan or apply starts, it locks the Terraform state
- This prevents concurrent operations that could cause conflicts
- The lock is held until the operation completes

Managing locks:

- Locks can be viewed and managed in the Atlantis UI
- Authorized users can manually remove locks if needed
- Access to the Atlantis UI is controlled by username/password

Best practices:

- Discourage users from manually unlocking/applying outside the normal workflow
- Limit access to the Atlantis UI to prevent unauthorized lock removal
- Educate users on the importance of respecting locks and the merge request process

Other notes:

- Plans run automatically on merge request creation without approval
- Apply requires manual approval via comment
- There's a separate Terraform setup managed by ML Ops team

The key benefits are automating Terraform operations, integrating with version control workflows, and providing locking to prevent conflicts. Proper education and access controls are important for using it effectively.

Making a commit->merge request

- A Jira ticket reference is required
- Taking the Jira ticket reference ie DOP-1234 prefix this to your branch name ie `git checkout -b DOP-1234_Change_to_terraform` (Ensure descriptive)
- On any change make a standard commit:
  - `git add <files>`
  - `git commit -m "Here are my changes"` (ensure this is descriptive)

- You will be able to add notes within gitlab but if you wish to add direct as a commit message use `git commit` and this will allow a longer more indepth commit message
- After all work is complete it may be good practice to squash the commit
  - `git rebase -i main`
    - Make changes as you wish
      - Common are:
        - `s` to squash a commit into the previous but retain the commit
        - `f` to squash the commit into the previous
        - `d` to remove that commit
    - Close the interactive text app
  - `git push -f`
- You can now create the merge request
- Atlantis will now run a plan
- Once the merge has been approved an apply can be made by commenting in gitlab comments `atlantis apply`

**i** If an issue occurs and the environment is locked or If you wish to abort an apply it is possible to do this via the Atlantis web GUI <https://atlantis.raft.ai/> (access granted from IT)