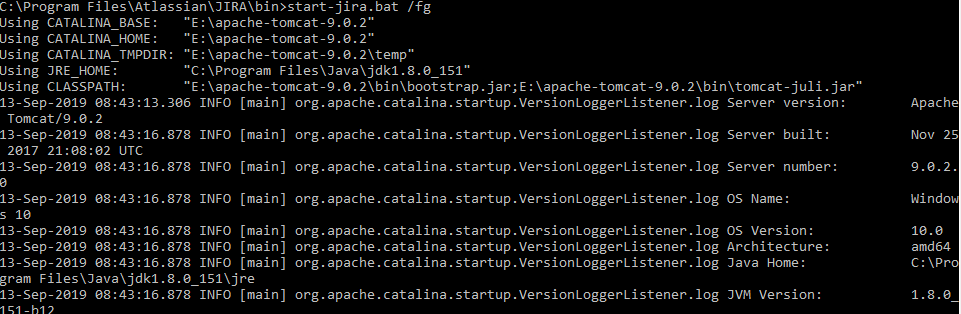
**GitLab & Jira Integration**

**Table of Contents**

|  |  |
| --- | --- |
| Topics | Page Number |
| Installing Jira, Login into Admin, creating the dashboard and user story creation | 01 -02 |
| Installing gitlab(AWS console), Login as Admin, Adding external url, Restarting gitlab | 03 -04 |
| Creating users in gitlab, SSH keys generation, Creating project and committing the code changes | 04-08 |
| Integrating jira in gitlab and creating service template(gitlab), Jiraservice setup(Jira) | 09-13 |
| Commit/Merge requests in Gitlab | 13-14 |
| Configuring Jira issue Tracker in Gitlab – Specific format | 15 |
| Jira userstory updated with Gitlab commit message | 15 |

**Installing Jira in local system**

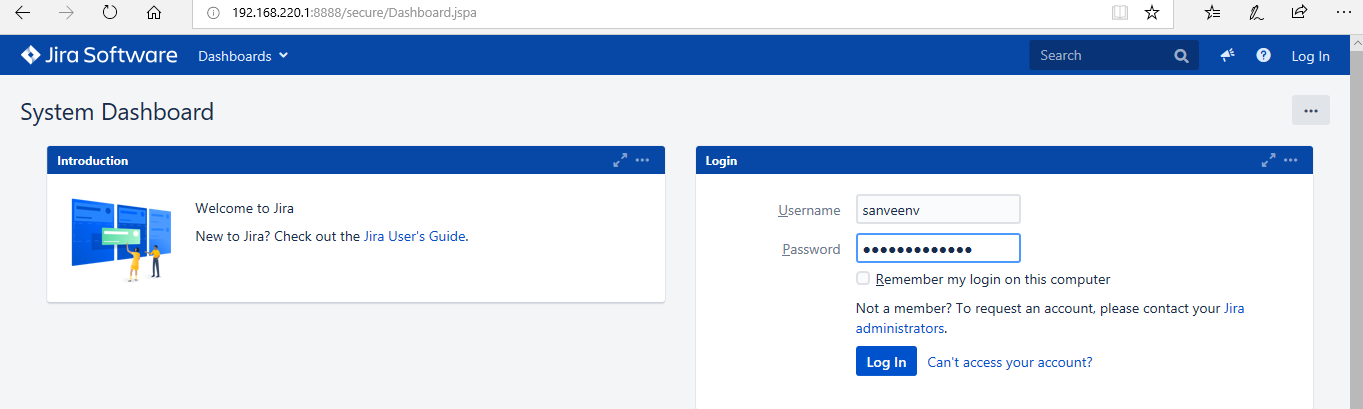
Download the Zip file and startup the batch file in foreground and used custom port 8888 for Jira.



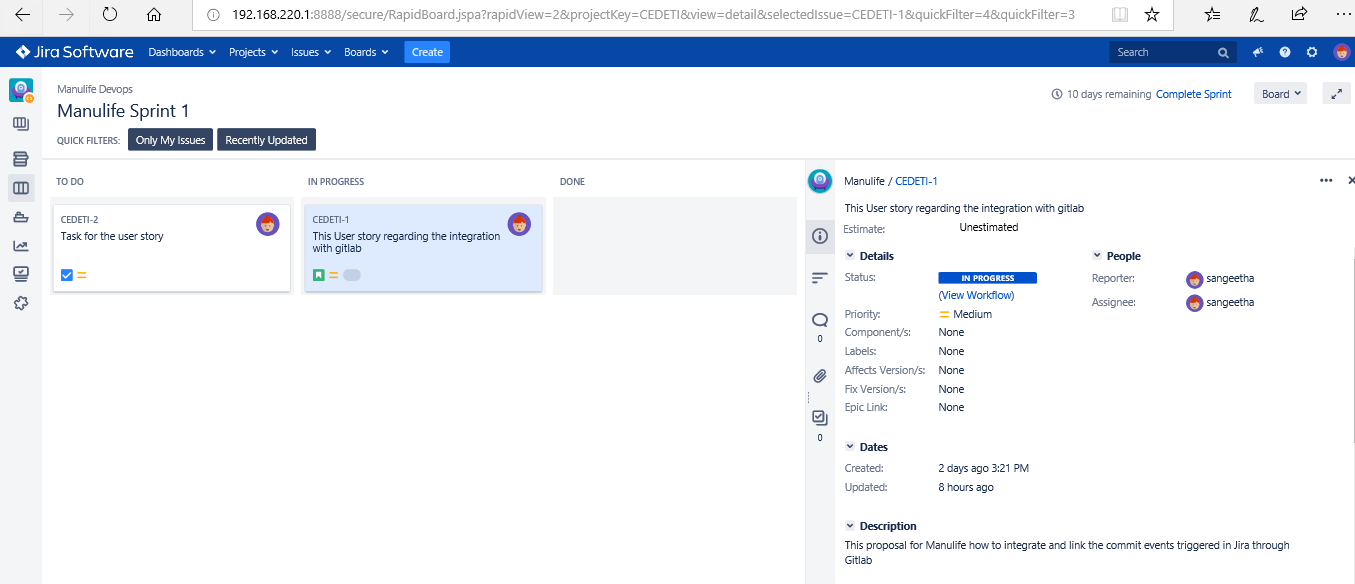
**Login into Jira**

[**http://192.168.220.1:8888/secure/Dashboard.jspa**](http://192.168.220.1:8888/secure/Dashboard.jspa)

create the Administrator account in Jira

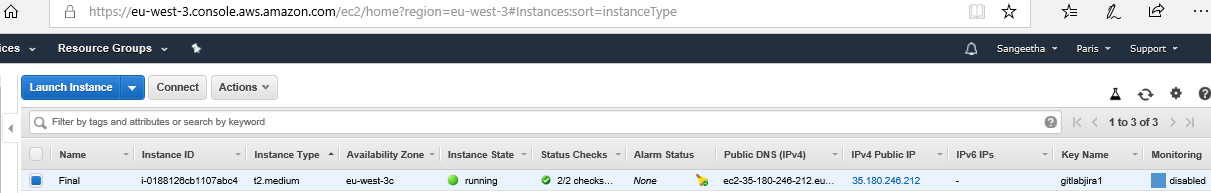


Created the Sprint and created the user story which matches with Manulife user story.

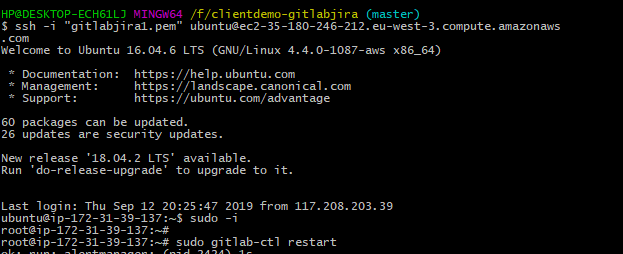


**GitLab Installation**

Taken AWS EC2 instance (Ubuntu) for installing Gitlab with Admin features.



**Connecting the Ubuntu instance through gitbash**



**Starting/Stopping the gitlab instance, below is the command.**

sudo gitlab-ctl stop

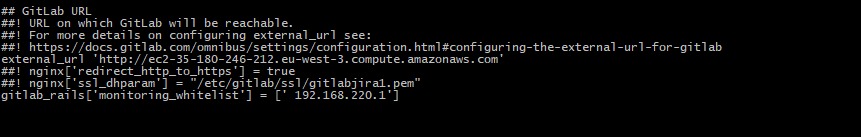
sudo gitlab-ctl start

For restarting the gitlab sudo gitlab-ctl restart

We need to do the following settings in /etc/gitlab/gitlab.rb

Make a changes by adding external url and add the local ip in the monitoring\_whitelist

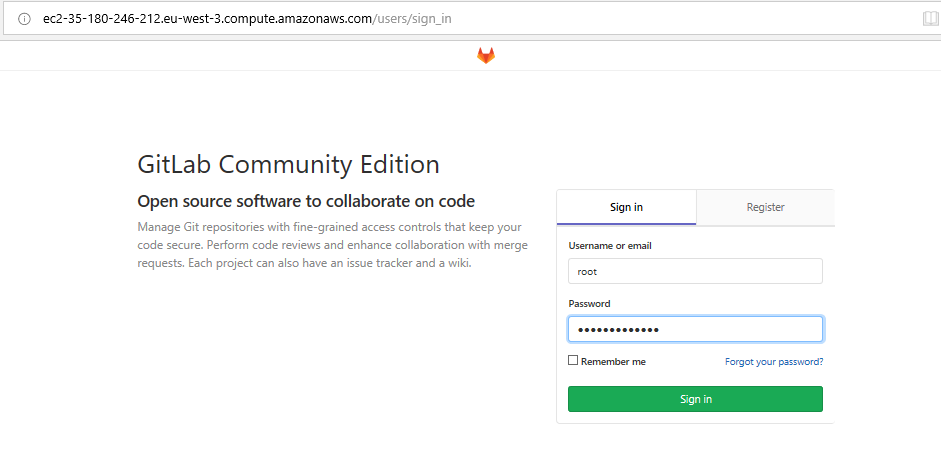
cat gitlab.rb



**Once make a changes , pleas reconfigure the gitlab to make in effective**

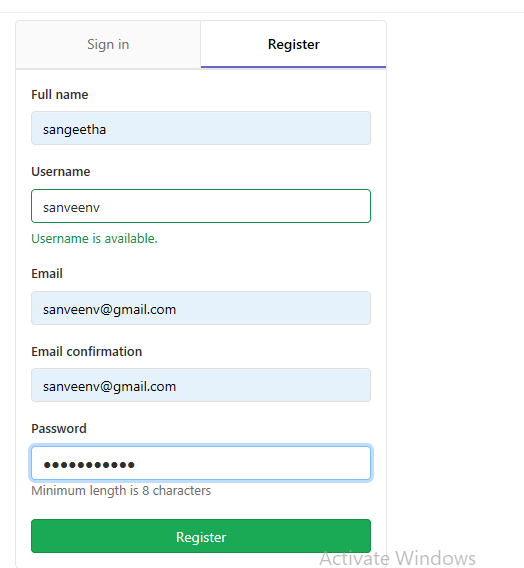
* **sudo gitlab-ctl reconfigure**

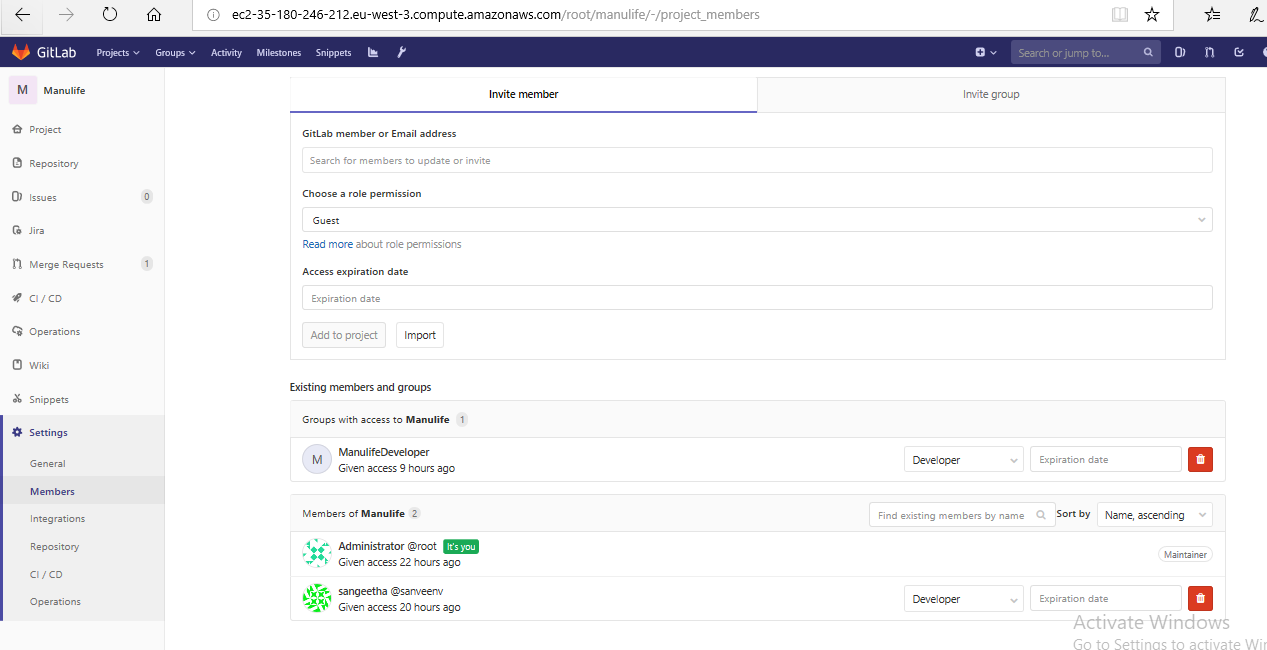
Then Login into Gitlab URL as mentioned in external\_url and login as Administrator



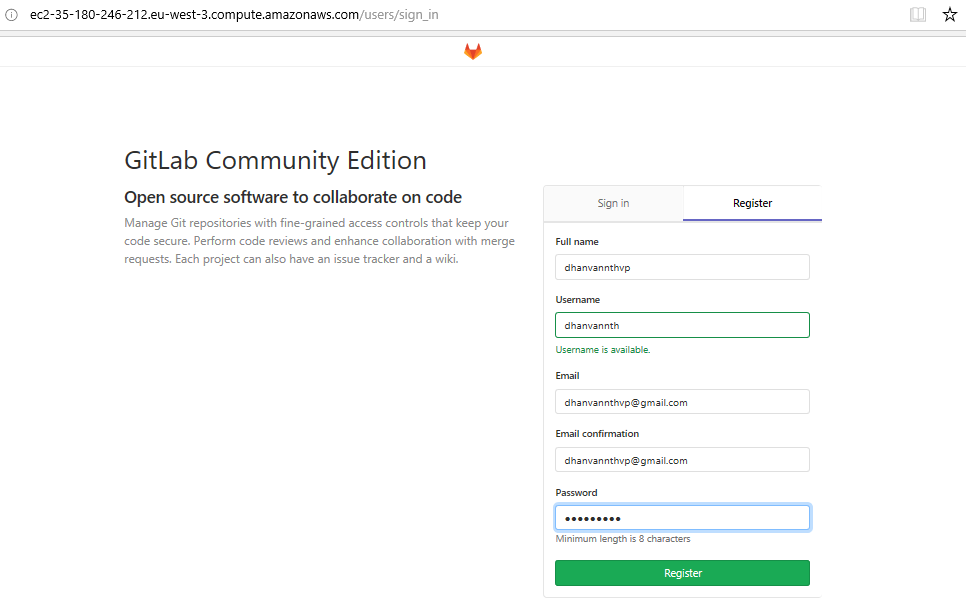
Then created members in setting under Admin access and assign to the specific group

1. created users as sangeetha and Dhanvannthvp





one more user in gitlab

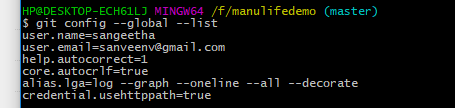


And then add the user dhanvannth in the specific group

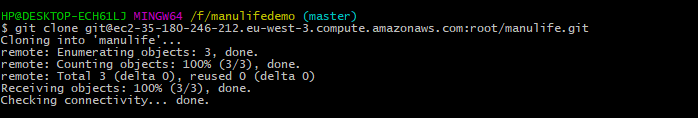
**Creating local repository and configuring changes**

Do the configuration in gitlab with username,email in local repository

**Git init** – will create the local repository.

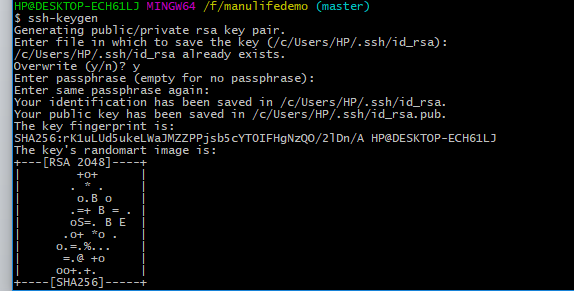


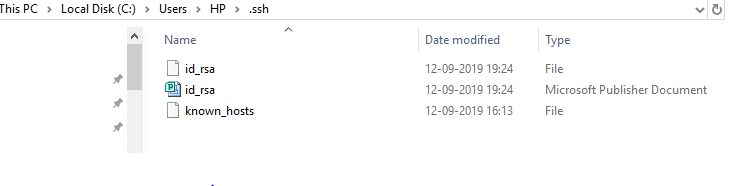
Clone your gitlab repository in local repository



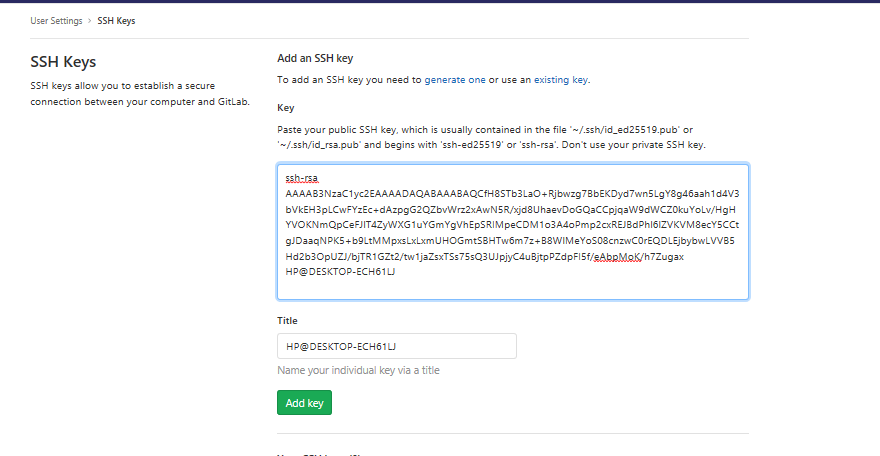
**SSH Keys Generation**

Generate the SSH keys to connect between your local system with Gitlab

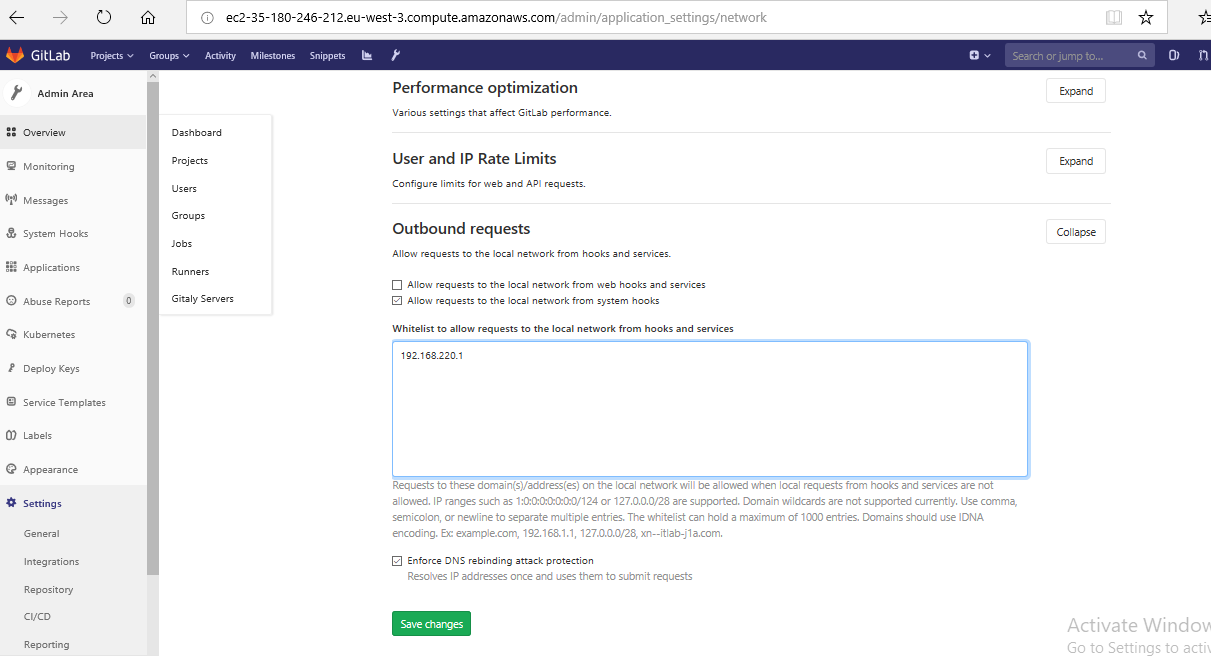




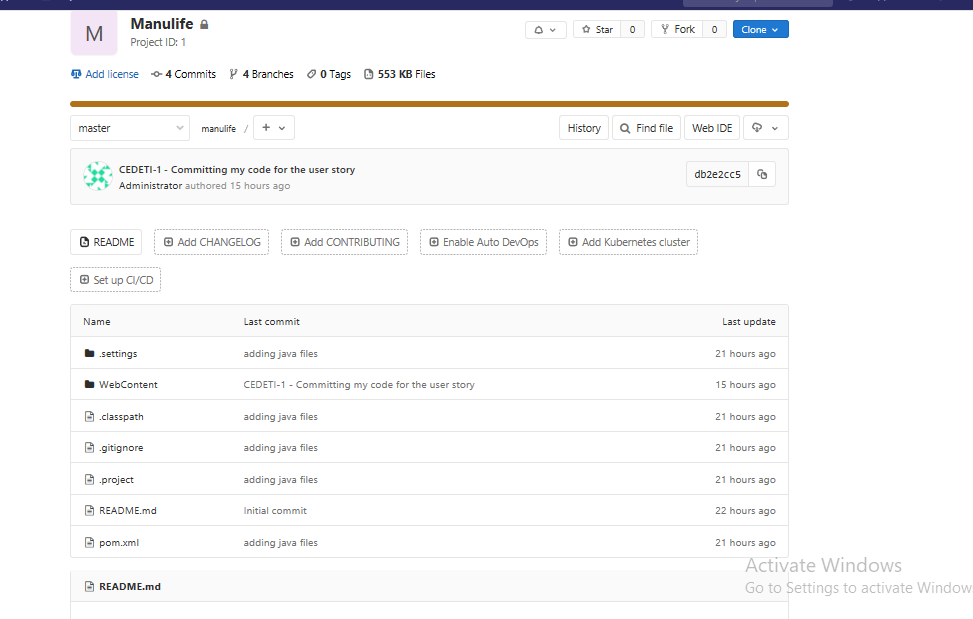
Copy the Public Key and Paste it under SSH keys. So created the connection between local system and gitlab

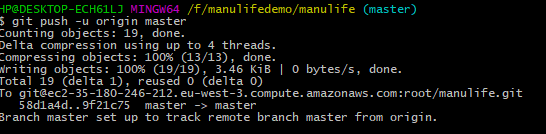


**Adding Jira local ip in whitelist**

And also please add the Jira local IP in gitlab network to accept it.

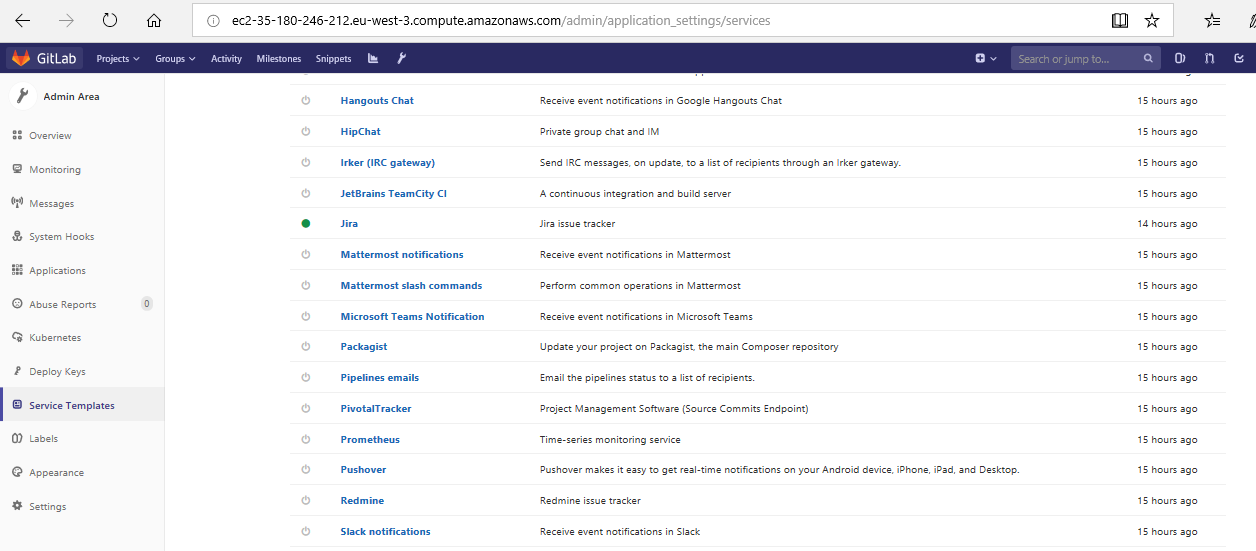
Created one Manulife project in gitlab and push your code from local repository



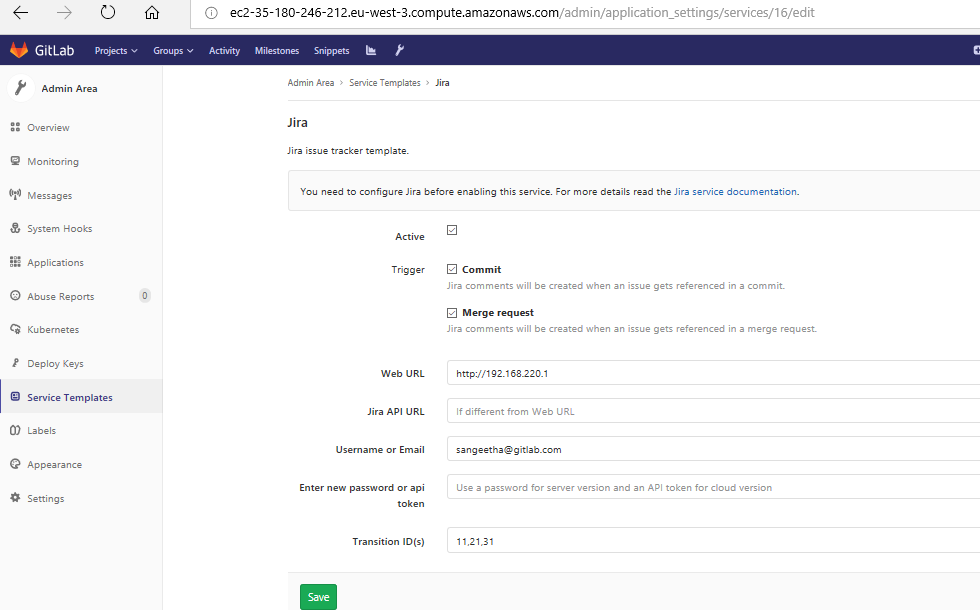


**Integrating jira in Gitlab**

And also Admin is having the rights to create Jira service template in Gitlab



And filling up with all required details from Jira

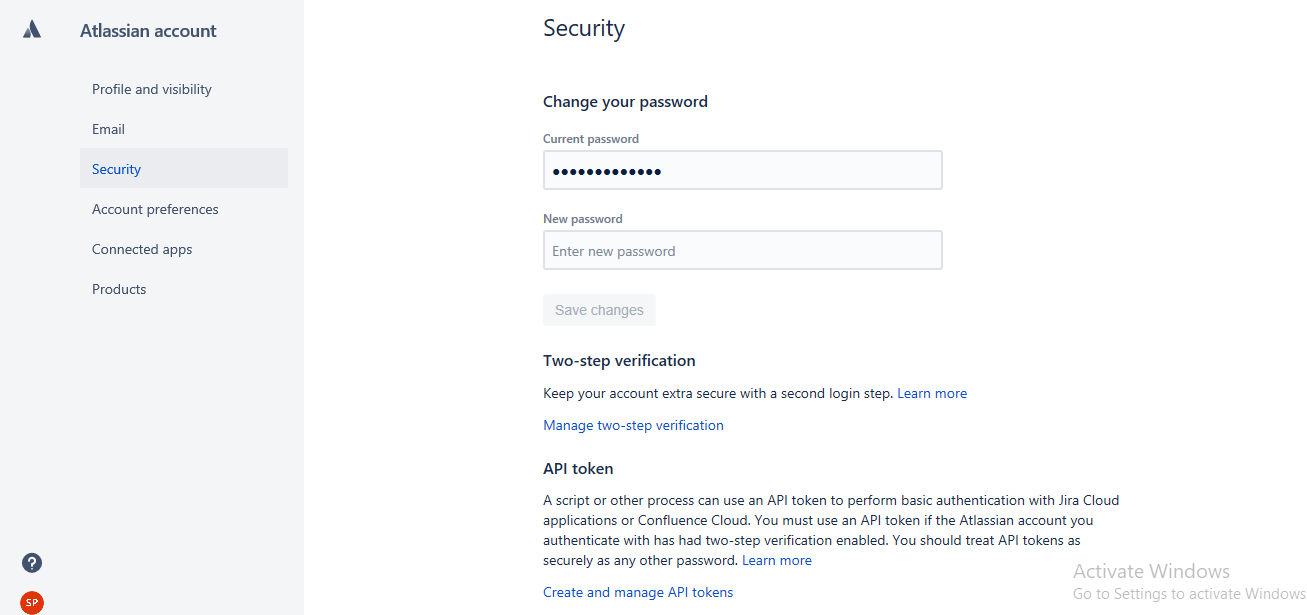


**Jira service setup**

Below is the screenshot to get API token, user name, Transition ID(s) from Jira end which required for service template in gitlab

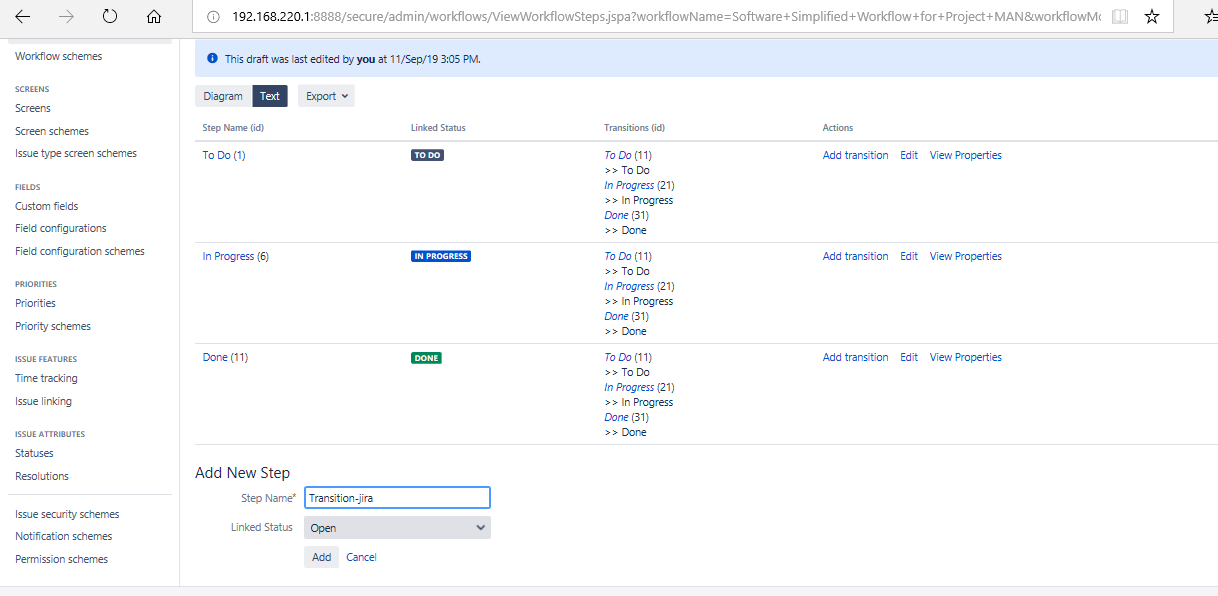
**To set API tokens,**

Click create and manage API tokens, generate the API token for your account. AxAuDW8INA5ae78diRjR39B7 is my jira API token.



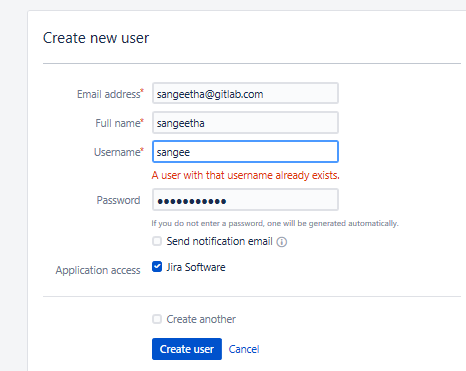
**To get the Transition IDs**

Go to Admin access, click workflows for the specific project, able to find the Transition ids for the user story. Copy the number and provide in service template under Transition ids in Gitlab



**User creation and Granting permission**

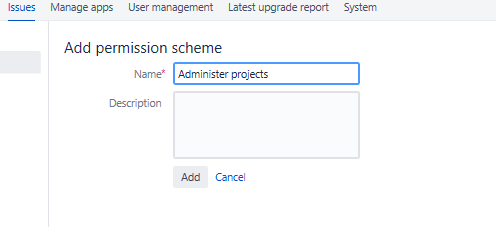
Create a user and assign to the group under Admin previleges

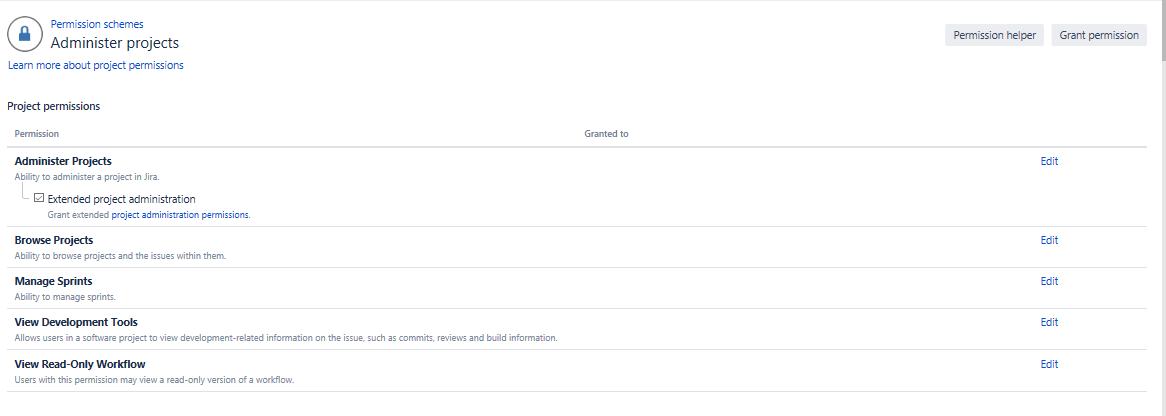


Step2: Creating group and add the user to the group

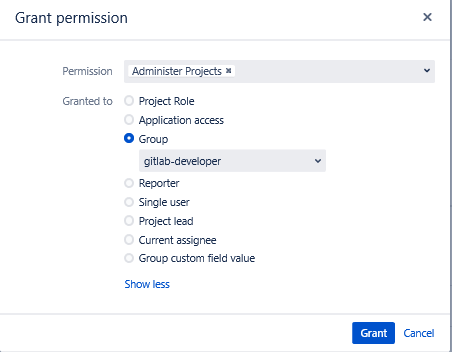


Click selected issues->Permission Schemes



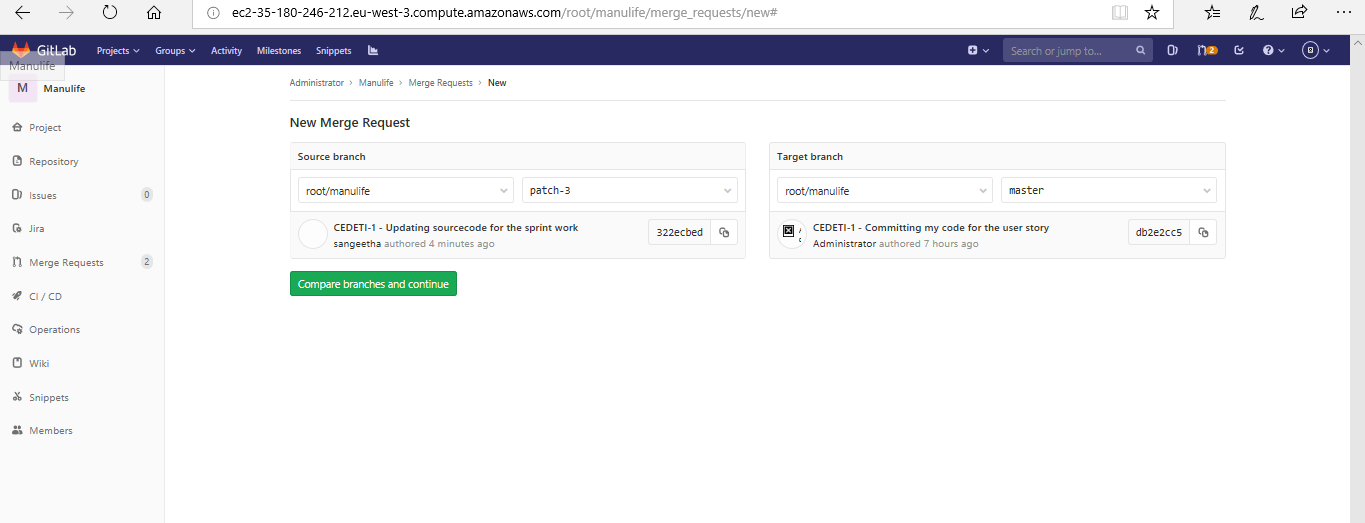


Grant permissions

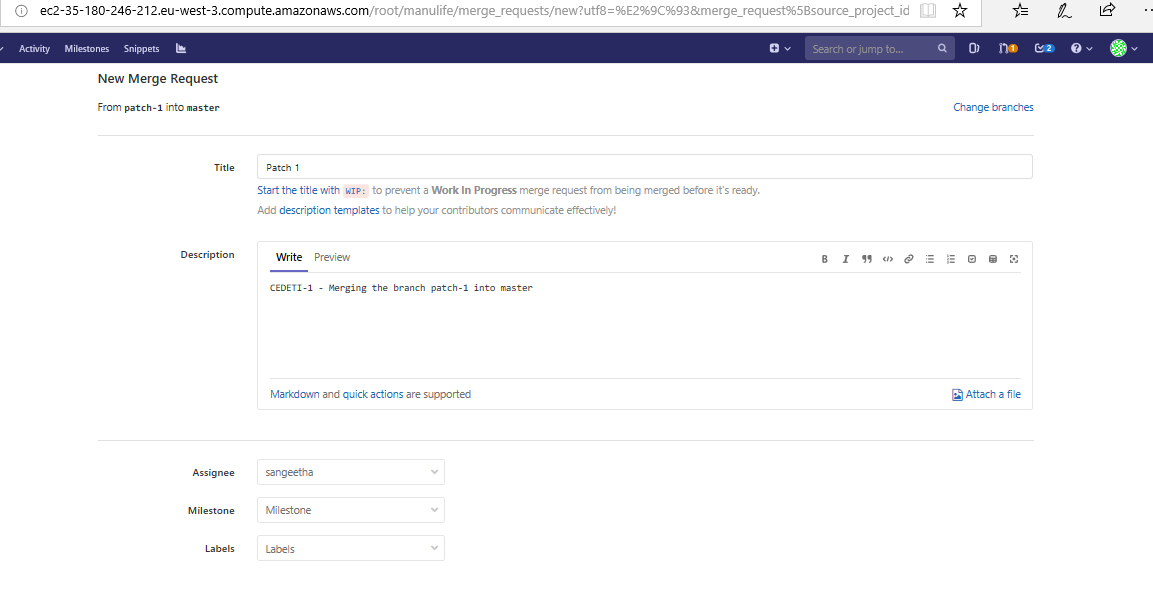


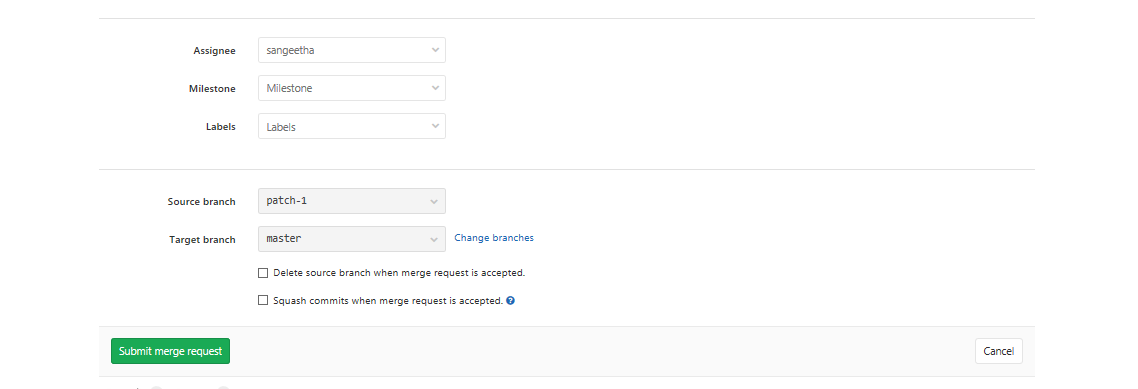
Both settings done in Gitlab and Jira. Now commit the message in gitlab and it will be reflected under comment section in jira.

**Commit/Merge requests in Gitlab**



Merge conflicts





When resolving merge conflicts, the commit message should be included with jira user story. Important set up is to configure Jira issue tracker in gitlab which provides the standard format for the commit message which will be reflected in Jira user story under comment section.

**Configuring Jira issue Tracker in Gitlab – Specific format**

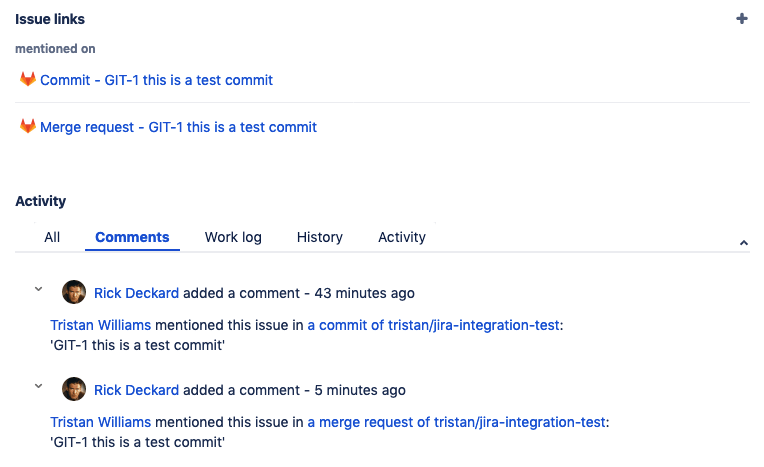
When GitLab project has Jira issue tracker configured and enabled, mentioning Jira issue in GitLab will automatically add a comment in Jira issue with the link back to GitLab. This means that in comments in merge requests and commits referencing an issue, e.g., PROJECT-7, will add a comment in Jira issue in the format:

USER mentioned this issue in RESOURCE\_NAME of [PROJECT\_NAME|LINK\_TO\_COMMENT]:

ENTITY\_TITLE

* USER A user that mentioned the issue. This is the link to the user profile in GitLab.
* LINK\_TO\_THE\_COMMENT Link to the origin of mention with a name of the entity where Jira issue was mentioned.
* RESOURCE\_NAME Kind of resource which referenced the issue. Can be a commit or merge request.
* PROJECT\_NAME GitLab project name.
* ENTITY\_TITLE Merge request title or commit message first line.

**Sample Output**



For example, the following commit will reference the Jira issue with PROJECT-1 as its ID:

git commit -m "PROJECT-1 Fix spelling and grammar"