**CHAPTER-1**

1. **GitLab Installation and configuration**
2. **1 Install and configure the necessary dependencies**

**sudo apt-get update**

**sudo apt-get install -y curl openssh-server ca-certificates tzdata perl**

Next, install Postfix (or Sendmail) to send notification emails. If you want to use another solution to send emails please skip this step and [configure an external SMTP server after GitLab has been installed](https://docs.gitlab.com/omnibus/settings/smtp).

**Sudo apt-get install -y postfix**

During Postfix installation a configuration screen may appear. Select 'Internet Site' and press enter. Use your server's external DNS for 'mail name' and press enter. If additional screens appear, continue to press enter to accept the defaults.

* 1. **Add the GitLab package repository and install the package**

**curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ee/script.deb.sh | sudo bash**

Next, install the GitLab package. Make sure you have correctly set up your DNS, and change https://gitlab.example.com to the URL at which you want to access your GitLab instance. Installation will automatically configure and start GitLab at that URL.

For https:// URLs, GitLab will automatically request a certificate with Let's Encrypt, which requires inbound HTTP access and a valid hostname. You can also use your own certificate or just use http:// (without the s ).

If you would like to specify a custom password for the initial administrator user ( root ), check the documentation. If a password is not specified, a random password will be automatically generated.

**sudo EXTERNAL\_URL="https://gitlab.example.com" apt-get install gitlab-ee**

Above command we replaced with below command

sudo EXTERNAL\_URL="https://172.31.112.38" apt-get install gitlab-ee

**1.3 Browse to the hostname and login**

Unless you provided a custom password during installation, a password will be randomly generated and stored for 24 hours in /etc/gitlab/initial\_root\_password. Use this password with username root to login.

**CHAPTER-2**

**2. Generate SSH Key**

* 1. **How to Generate an SSH Key**

**Step 1: Open Terminal**

Launch your terminal application on your system.

**Step 2: Generate an SSH Key Pair**

Run the following command in the terminal, replacing the email with your own:

* **ssh-keygen -t rsa -b 4096 -C** [**your-email@example.com**](mailto:your-email@example.com)

****

**Step 3: Save the SSH Key**

* The terminal will prompt:

Enter file in which to save the key **(/home/your\_user/.ssh/id\_rsa):**

* Press Enter to save the key in the default location (/home/your\_user/.ssh/id\_rsa).
* If the key file already exists, you will be asked whether you want to overwrite it. Type “**y”** and press Enter to overwrite.

A black background with white text

Description automatically generated

**Step 4: Create a Passphrase (Optional)**

* You'll be prompted to enter a passphrase for additional security

**Enter passphrase (empty for no passphrase):**

* If you want extra protection for your key, enter a passphrase.

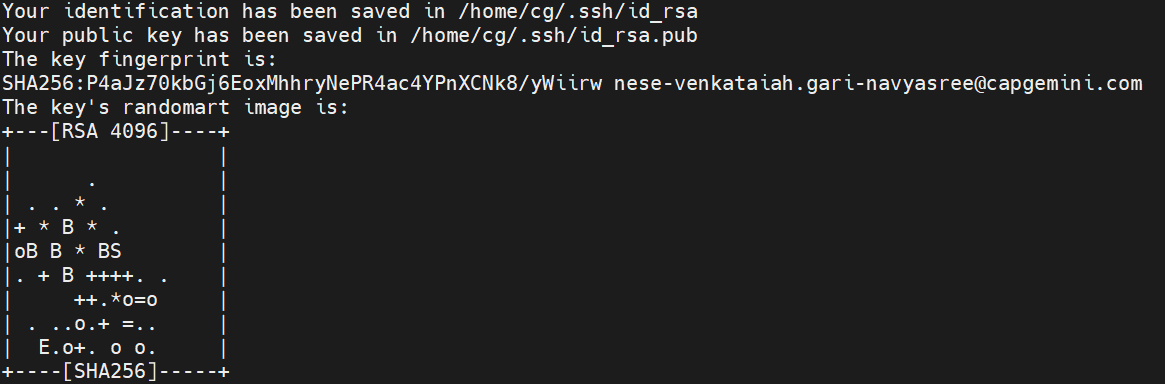
A computer screen with white text

Description automatically generated

* If you don’t want a passphrase, leave it empty and press Enter.
* Re-enter the passphrase if you set one.

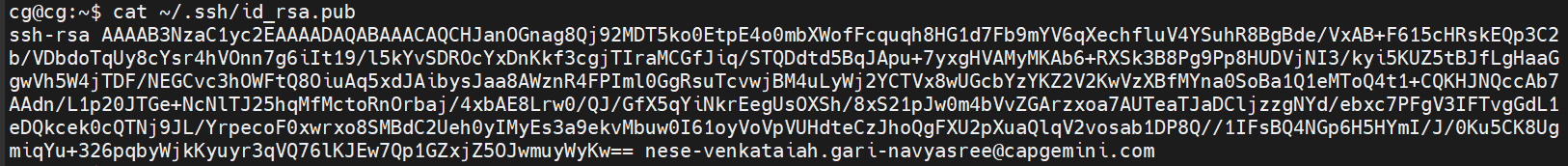
**Step 5: SSH Key Generation Complete**

* After successful generation, you will see messages like

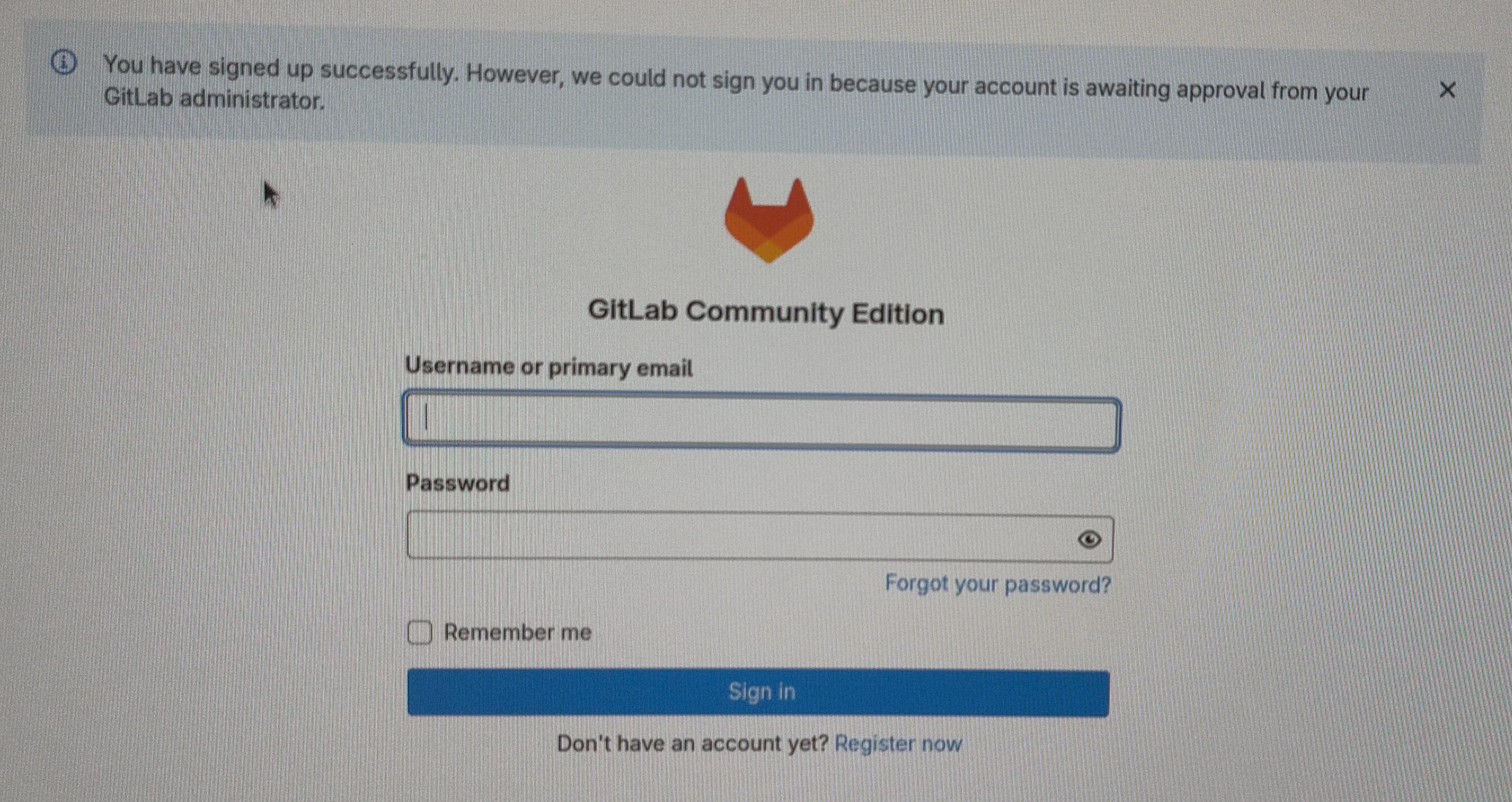


**Step 6: Copy the Public SSH Key**

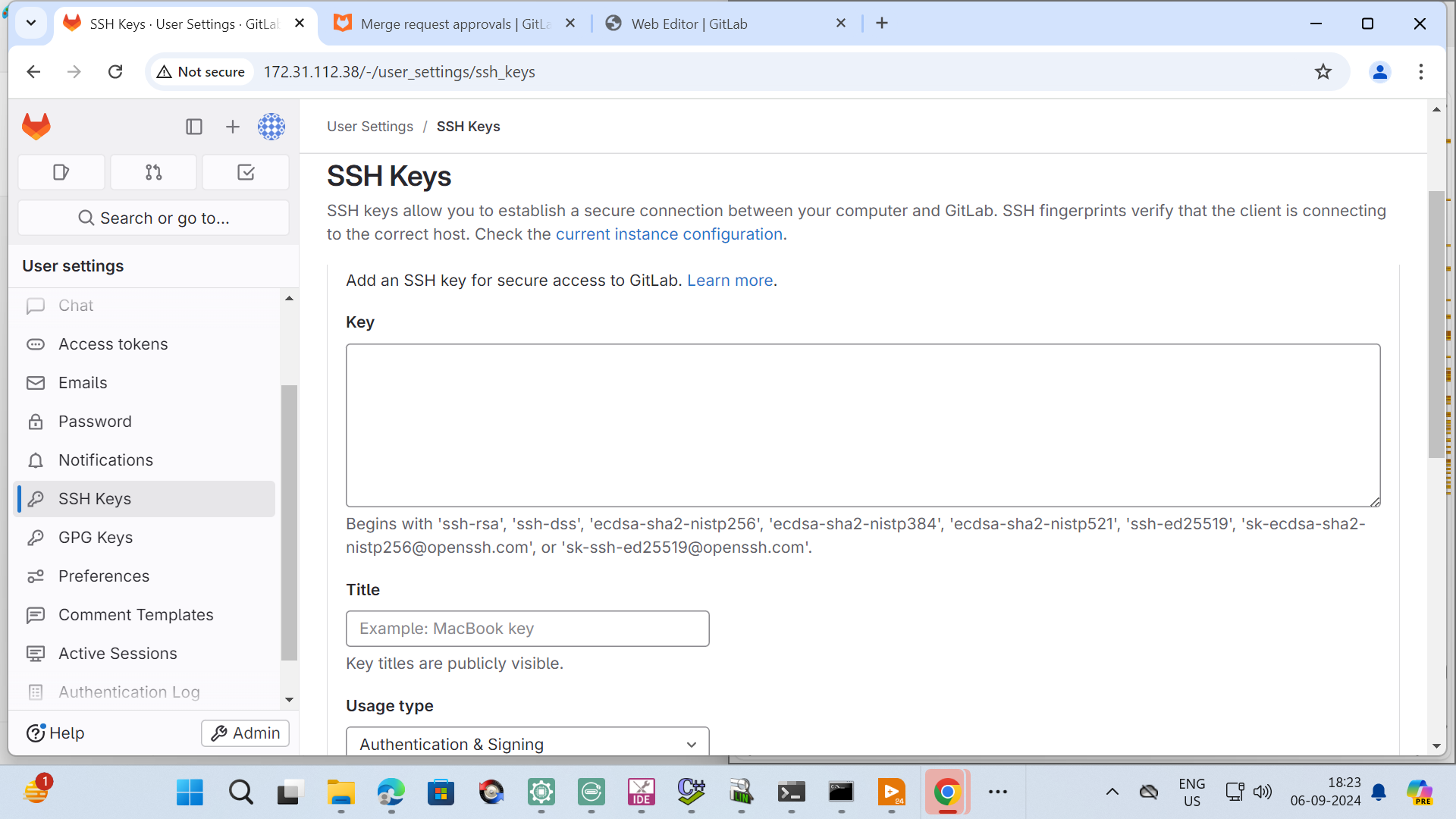
* To use the SSH key, you’ll need to copy the contents of the public key file.



* Copy the entire key to your GitLab clipboard.
* Enter http:172.31.112.38 in your web browser.
* You will be directed into Gitlab login page.
* Enter below credentials to login :



* **Username: airbus\_project**
* **Password: Cg@09871234**
* After successful verification, the following page will be displayed.
* Copied ssh key paste under key.



The outlined area above indicates where the generated SSH key should be pasted.

# **CHAPTER-3 3. Pushing code to gitlab**

**Step 1: Navigate to Your Local Repository**

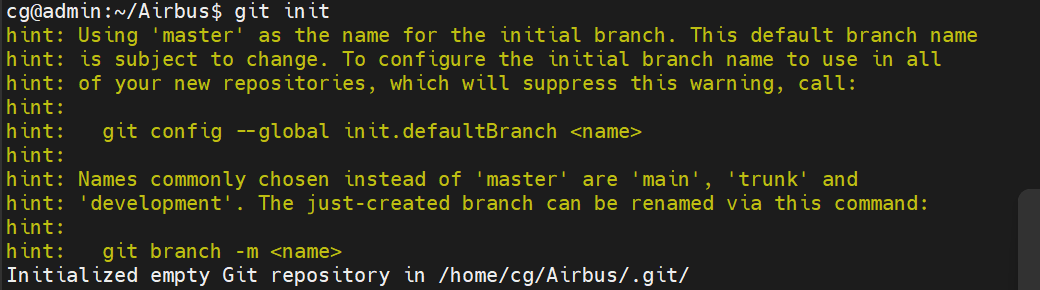
Open a terminal and change to your local directory that contains the code.

****

**Step 2: Initialize Git Repository (if not already done)**

If you haven’t already initialized a Git repository in your local directory, run:

**git init**

****

**Step 3: Add Remote Repository**

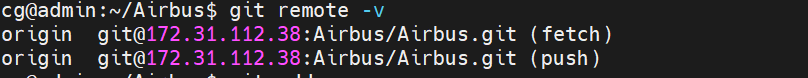
Add the URL of your GitLab repository as a remote.

git remote add origin git@172.31.112.38:Airbus/Airbus.git



**Step 4: Check Remote URL**

Verify that the remote URL has been added correctly.



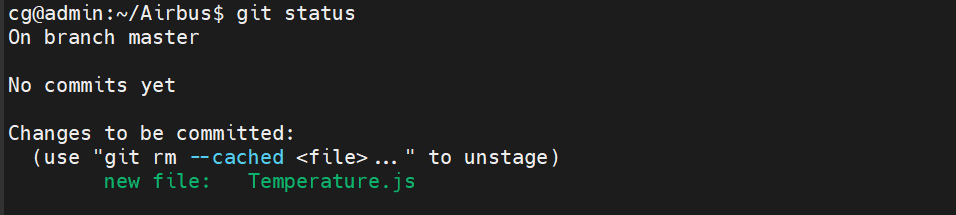
**Step 5: Add Files to Staging Area**

Add all the files in your local directory to the staging area.

**git add .**

****

**git status**

****

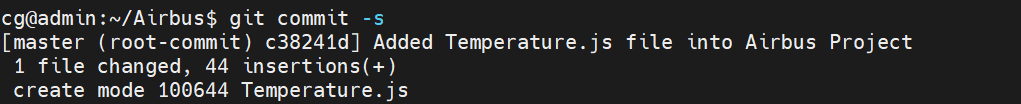
This status indicates that you have staged all the files in the Airbus directory for the initial commit.

**Step 6: Commit Changes**

Commit the added files with a descriptive message.

**git commit -s** and press enter

Add commit message in 1st line and save the message by clicking **ctrl+o** , **Enter**, **ctrl+x**.



**git branch**

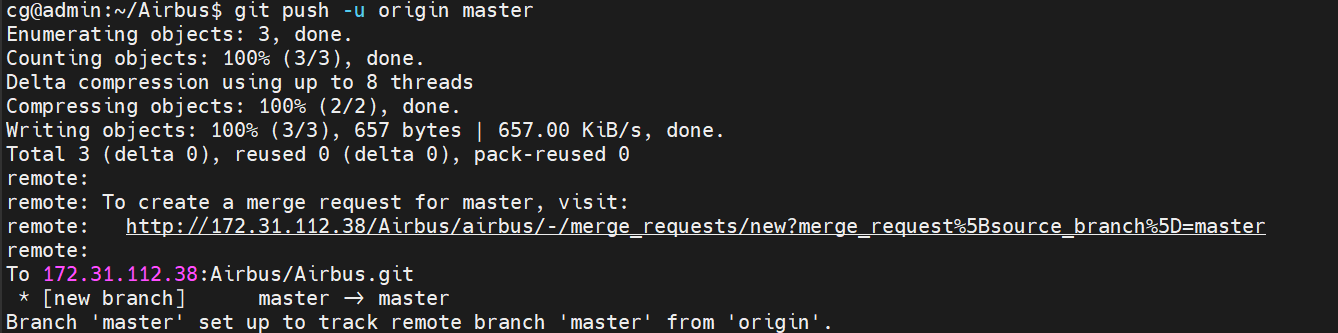
The branch you are currently on will be highlighted with an asterisk (\*) next to its name.

****

**Step 7: Push to GitLab**

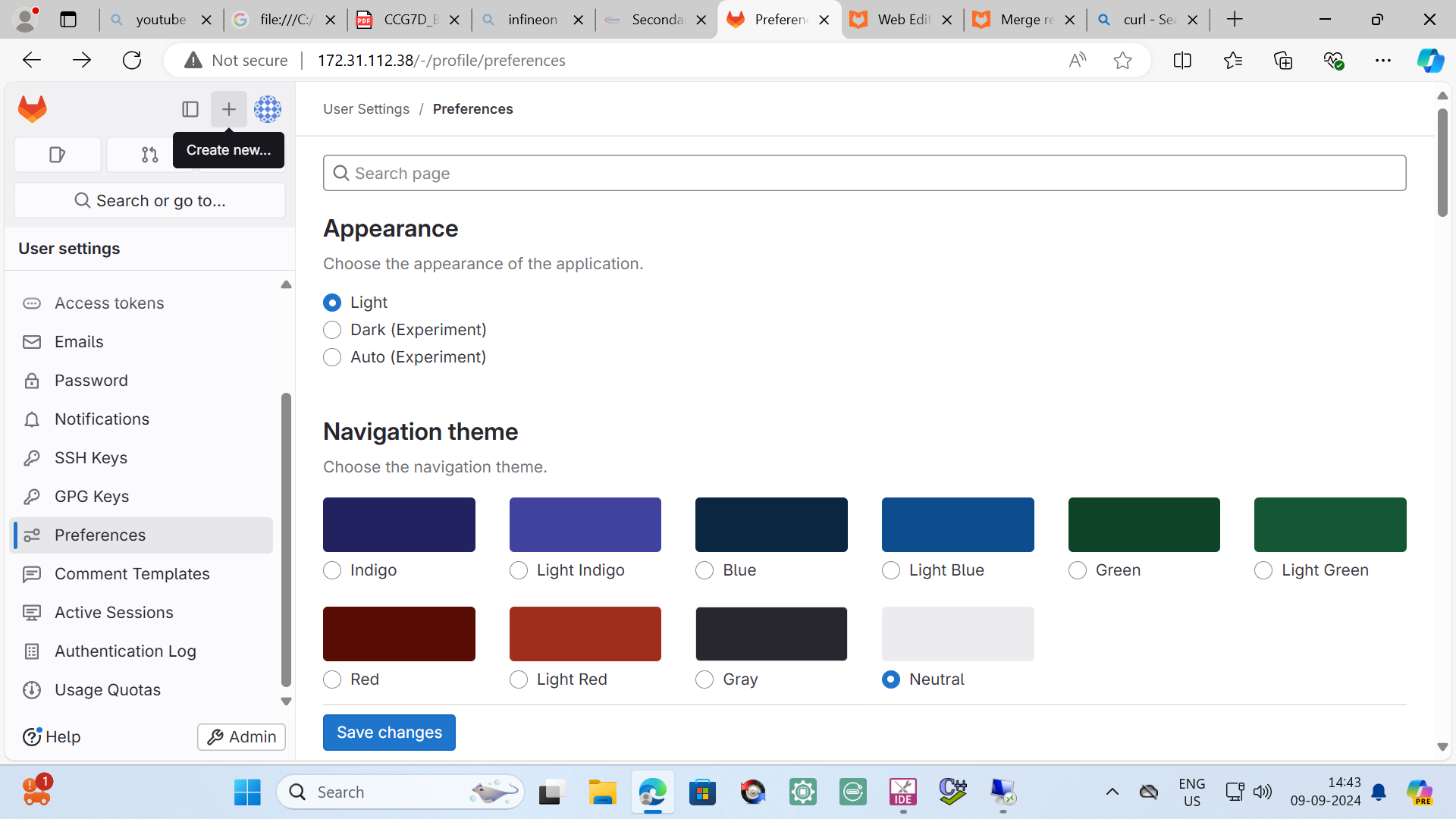
Push the committed changes to the main branch of your GitLab repository. If the main branch does not exist, you can push to master or create a new branch as needed.

**git push -u origin master**

****

**Next let's look at how to create and manage groups and sub-groups in Gitlab.**

Create a New Group by clicking “**+**” button on the top bar and choose the New group by taking below images as reference.



A screenshot of a computer

Description automatically generated

we can now add a new user to the group now because this user is a developer member of the group they automatically get developer access to all projects within that group.

A screenshot of a computer

Description automatically generated

Add Group name and make it as private and add team members.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

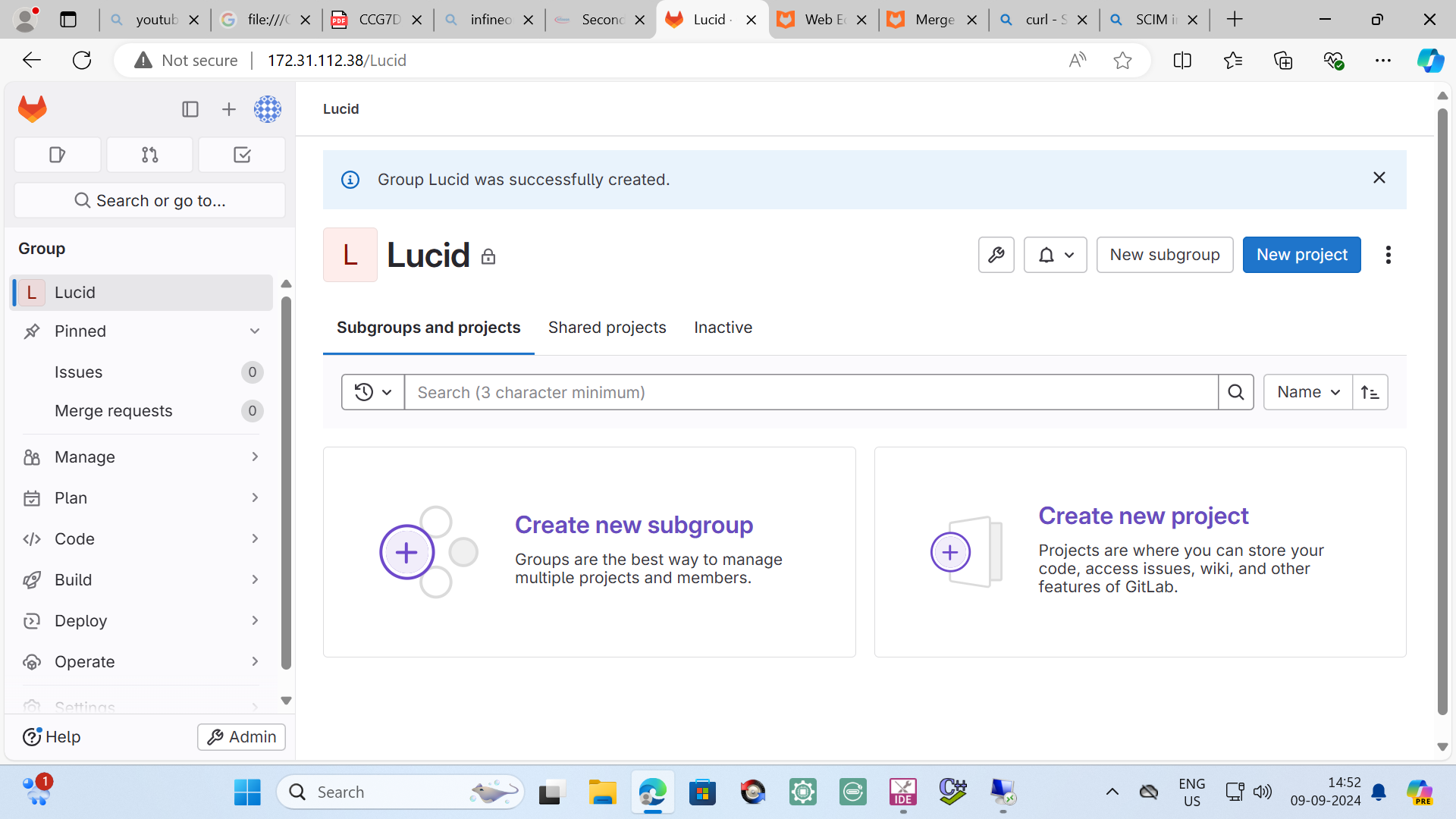
Description automatically generated

If required add sub-groups.

A screenshot of a computer

Description automatically generated

Likewise, create remaining Groups Lucid, Bose, NXP.



A screenshot of a computer

Description automatically generated

Created Groups will appear as shown below.

A screenshot of a computer

Description automatically generated

Create project for each user.

A screenshot of a computer

Description automatically generated

Select project.

A screenshot of a computer

Description automatically generated

Click on create project

A screenshot of a computer

Description automatically generated

Select create blank project and Enter Project name as NXP.

A screenshot of a computer

Description automatically generated

Follow below image and click on create project.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Follow the same and create project for remaining groups.