Getting Started with GitLab Repository

Table of Contents

[Introduction 2](#_Toc514248412)

[Pre-requisites 2](#_Toc514248413)

[Security 2](#_Toc514248414)

[URL 3](#_Toc514248415)

[What to put in GitLab Server 4](#_Toc514248416)

[Logon to GitLab Server 4](#_Toc514248417)

[Generating a new SSH key pair 5](#_Toc514248418)

[Windows 5](#_Toc514248419)

[Linux 5](#_Toc514248420)

[Adding SSH key to GitLab 6](#_Toc514248421)

[How to create a project in GitLab 7](#_Toc514248422)

[Create a new group in GitLab 7](#_Toc514248423)

[Add users to a group 8](#_Toc514248424)

[Create a new Repository (Linux) 10](#_Toc514248425)

[Things to Note: 11](#_Toc514248426)

Getting Started with GitLab Repository

# Introduction

This document describes how to create a git project in GitLab Server inside Discovery Platform, how to clone the git project from GitLab Server to local VM and how to commit new files into GitLab Server. GitLab Repository helps you to save your work (source codes and documents) when you are working on particular VM (either Linux or Windows) inside DP and it could improve your collaboration with other team members when you share your project to other users or when you are shared with their projects. GitLab Server makes sure that your works are safe because GitLab Server stores them securely and there is also everyday backup being done behind. From any VM inside Discovery Platform, you can reuse, redeploy, collaborate and share your work anytime from anywhere.

# Pre-requisites

* You can access Discovery Platform
* You are part of one group (mas\_xxxx:groupnamexxx) in Discovery Platform
* Your VM is either Windows or Linux with GUI
* If you don’t meet all the criteria mentioned above, stop here.

# Security

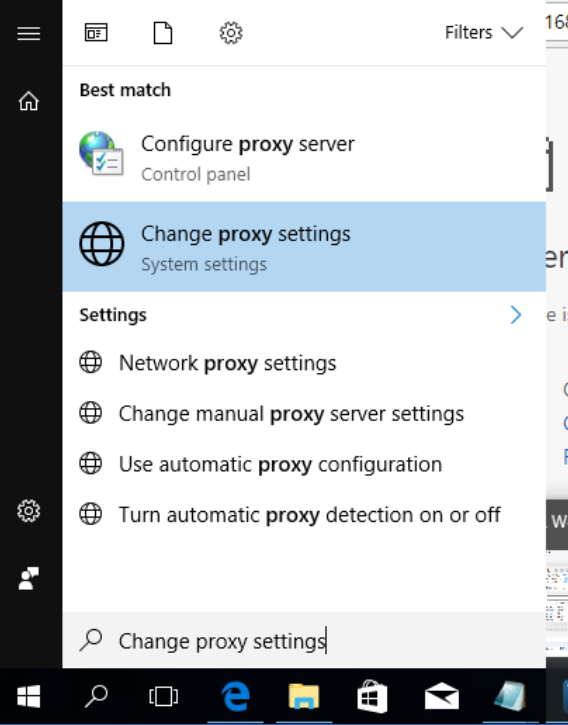
GitLab Server is secured with SSL encryption and domain level authentication. There is no man-in-middle attack or password unsecure or no password saving at GitLab Server because GitLab Server is using SSL and active directory authentication.

# URL

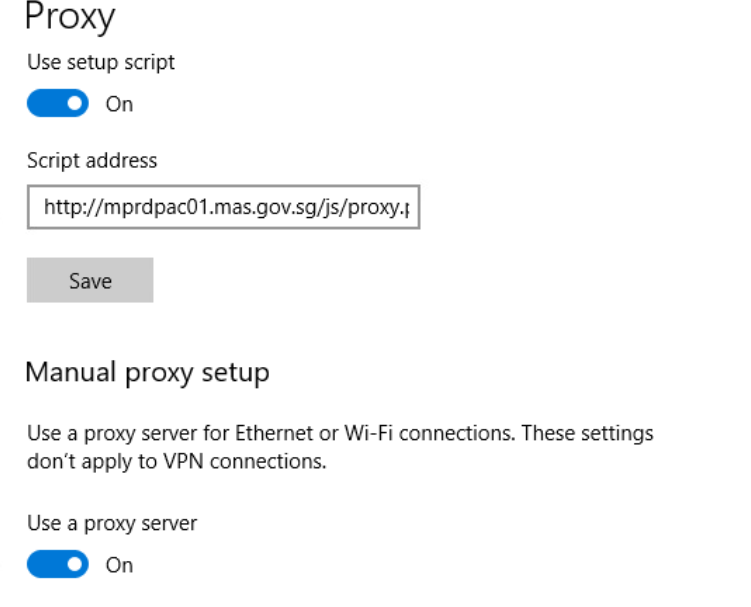
<https://192.168.11.23> is the url you need to use in your browser.

If you are using Windows VM, you may need to turn off proxy as mentioned below.

1. At Windows menu, type proxy and select **Change proxy settings.**



1. Turn off proxy by unchecking **Use setup script** to **Off** and by unchecking **Use a proxy server** to **Off**. Make sure that both options are off.



# What to put in GitLab Server

* Programming codes such as Python, Java, Scala, R, Java, Javascript, SQL, Conda, Shell in relevant extension formats
* Text Files or CSV files or Excel files or Microsoft Office files
* Data files
* Any Files which are not extraordinarily huge.

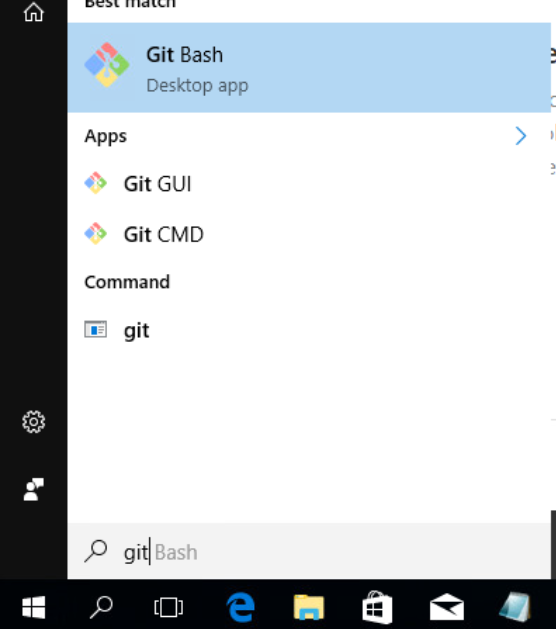
# Logon to GitLab Server

1. Open Firefox or Chrome Browser
2. Type in url <https://192.168.11.23> into URL bar. Click Enter.
3. Login Page appeared. Choose LDAP option, fill in LDAP Username and Password (LDAP Username is your MAS username and Password is your MASWORLD password. Click Sign in.
4. Now you are successfully logged on to GitLab Server.

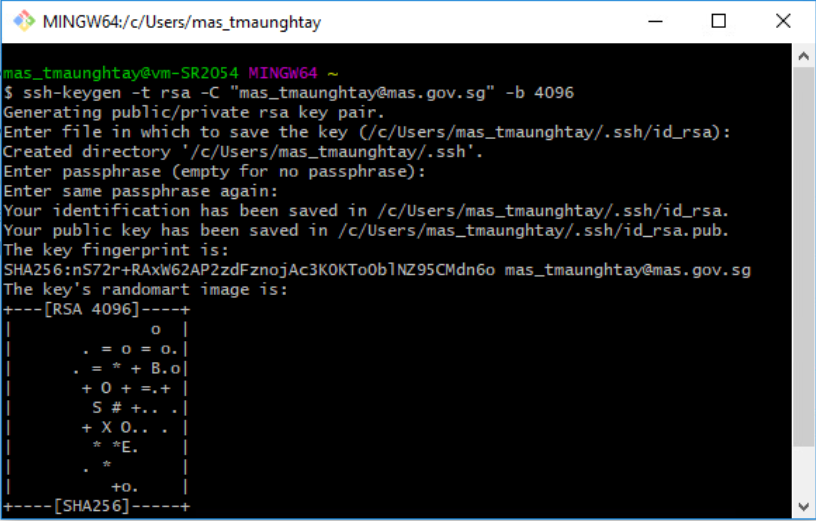
# Generating a new SSH key pair

## Windows

1. To generate a new SSH key pair, use the following command after launching Git Bash program in Windows. Click Enter when you are asked to give the input. You need to provide Enter when asked a passphrase so that there will be no passphrase.



$ ssh-keygen –t rsa –C “your preferred email or comment” –b 4096



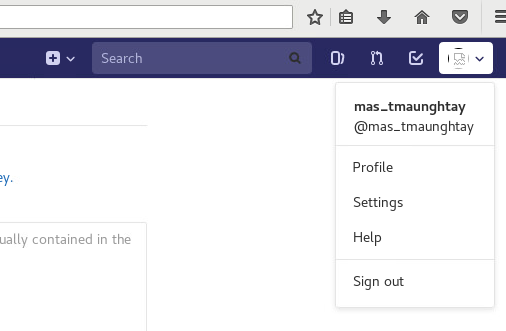
## Linux

1. Launch **Terminal**. Type below command and provide **Enter** when asked to input where to save, giving a passphrase, etc.

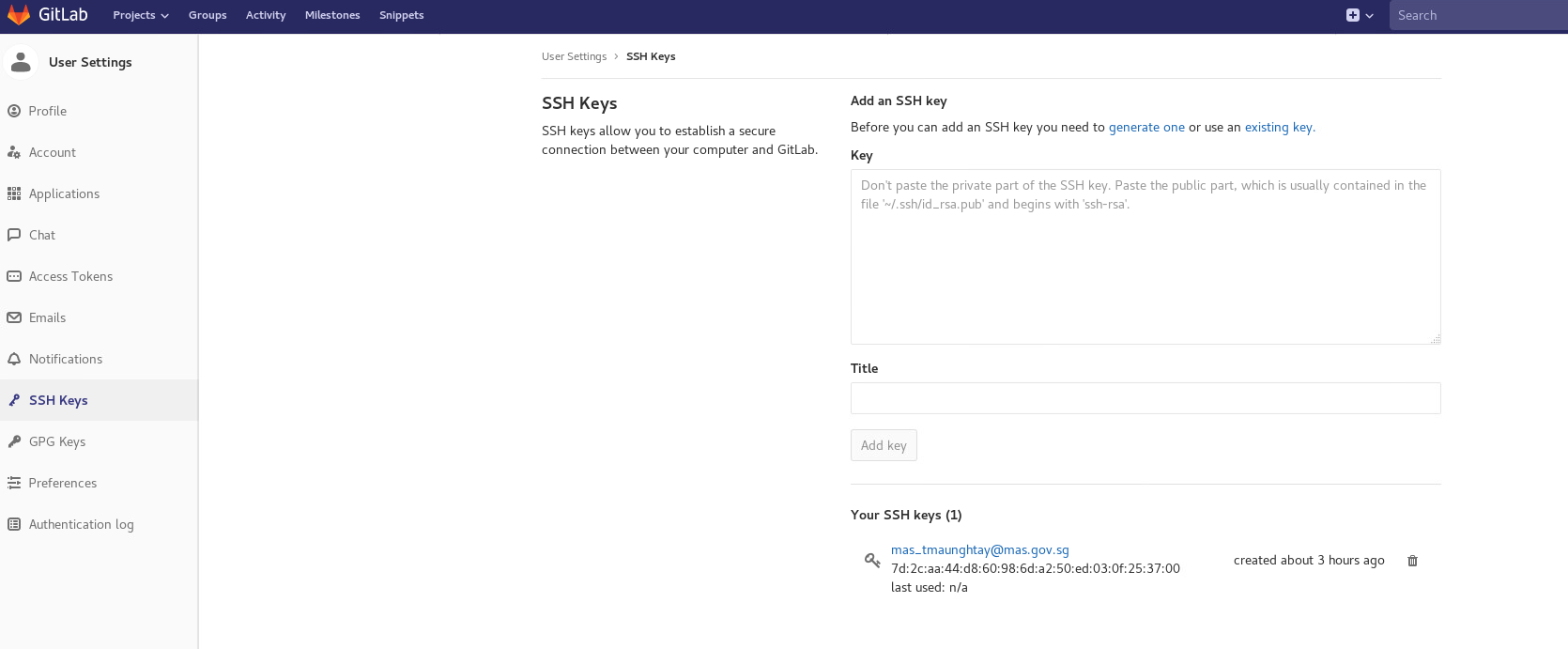
$ ssh-keygen –t rsa –C “mas\_tmaunghtay@mas.gov.sg” –b 4096

# Adding SSH key to GitLab

1. Before you proceed, make sure you already have SSH key generated. If you have not, please do so as mentioned in **Generating SSH key.**
2. Go to **your account profile** and select **Settings**.

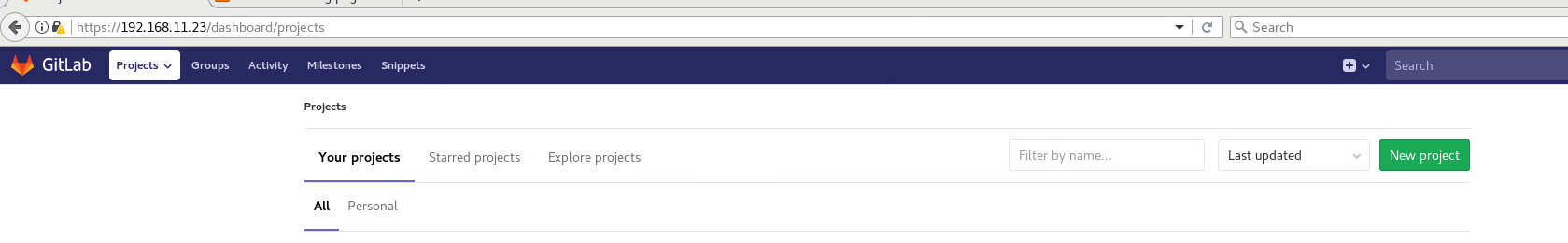


1. Click SSH Keys on left hand menu. **Copy your SSH key** from your local VM and paste it into **Key** field. Click **Add key**.

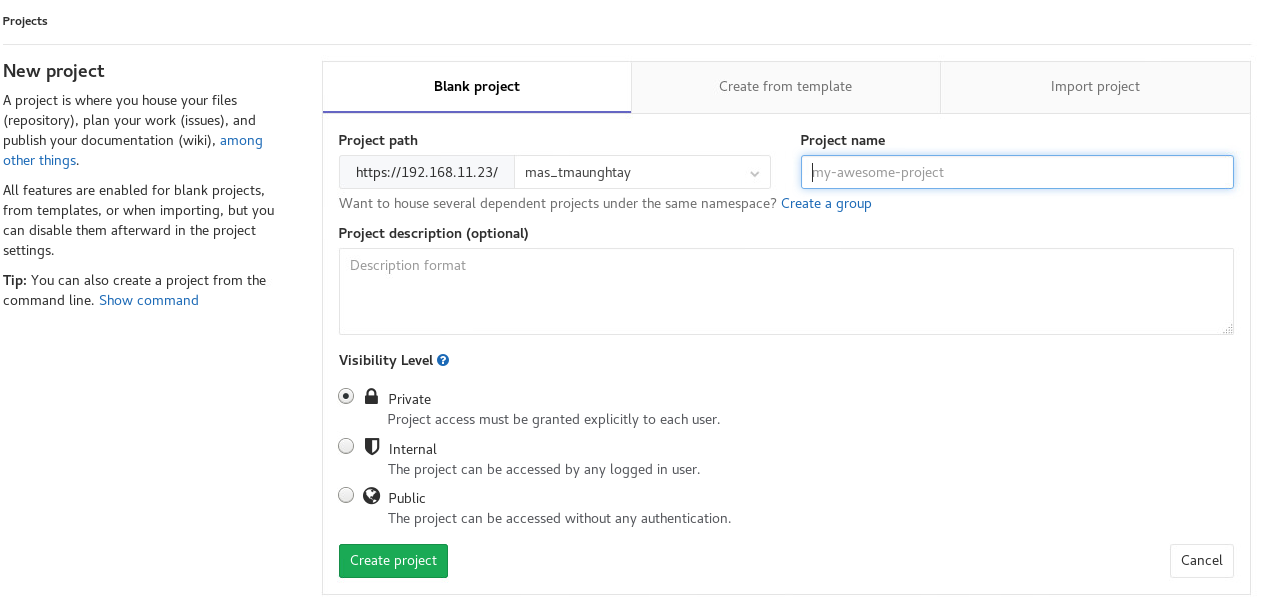


# How to create a project in GitLab

1. In your dashboard, click the green **New project** button or use the plus icon in the upper right corner of the navigation bar. This opens **New project** page.



1. Choose if you want start a blank project, Enter the **Project name** field. Project description (optional) field enables you to enter a description for your project dashboard, which will help other understand what your project is about. Changing the Visibility Level modifies the project’s viewing and access rights for users. Click **Create project** to get started with a new project.

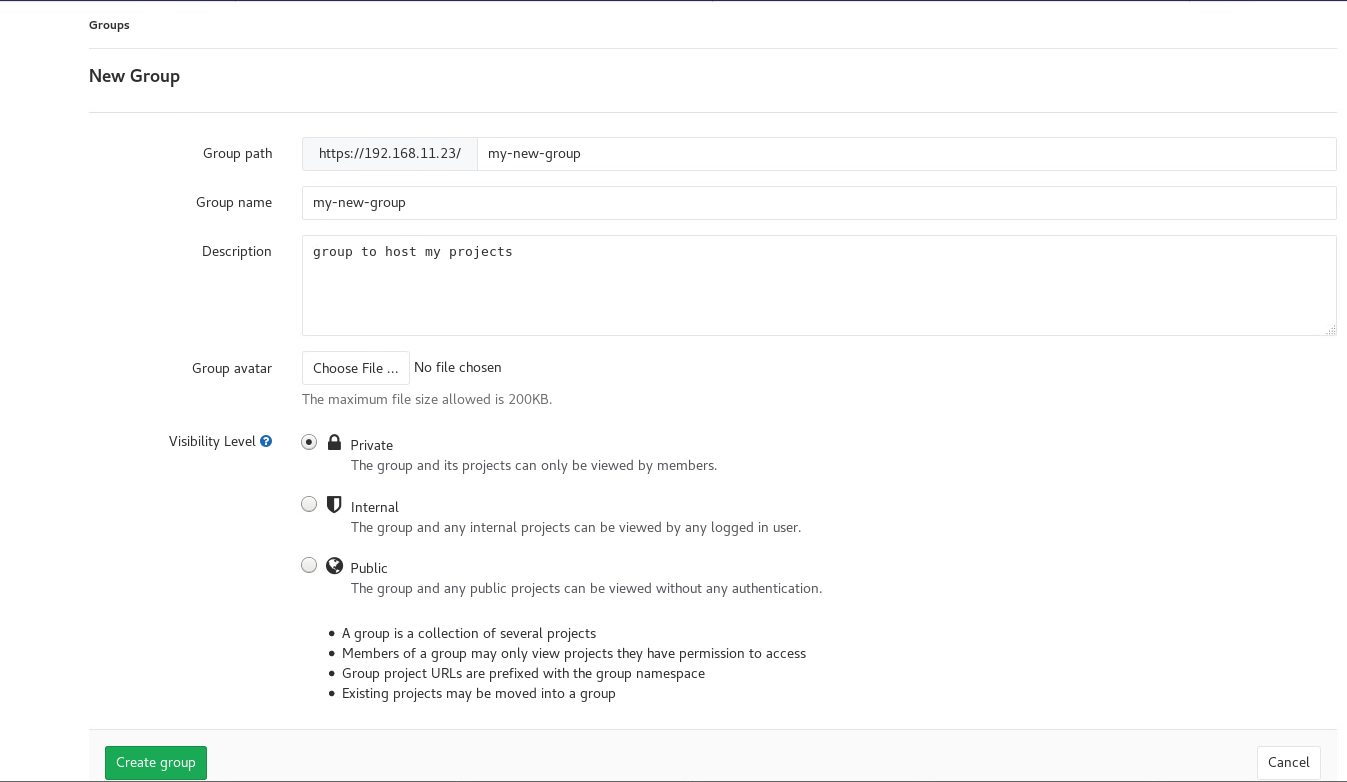


# Create a new group in GitLab

1. From the top menu, click **Groups**, and click the green button **New group**. You can also expand the plus sign button on the top navbar and choose **New group**.

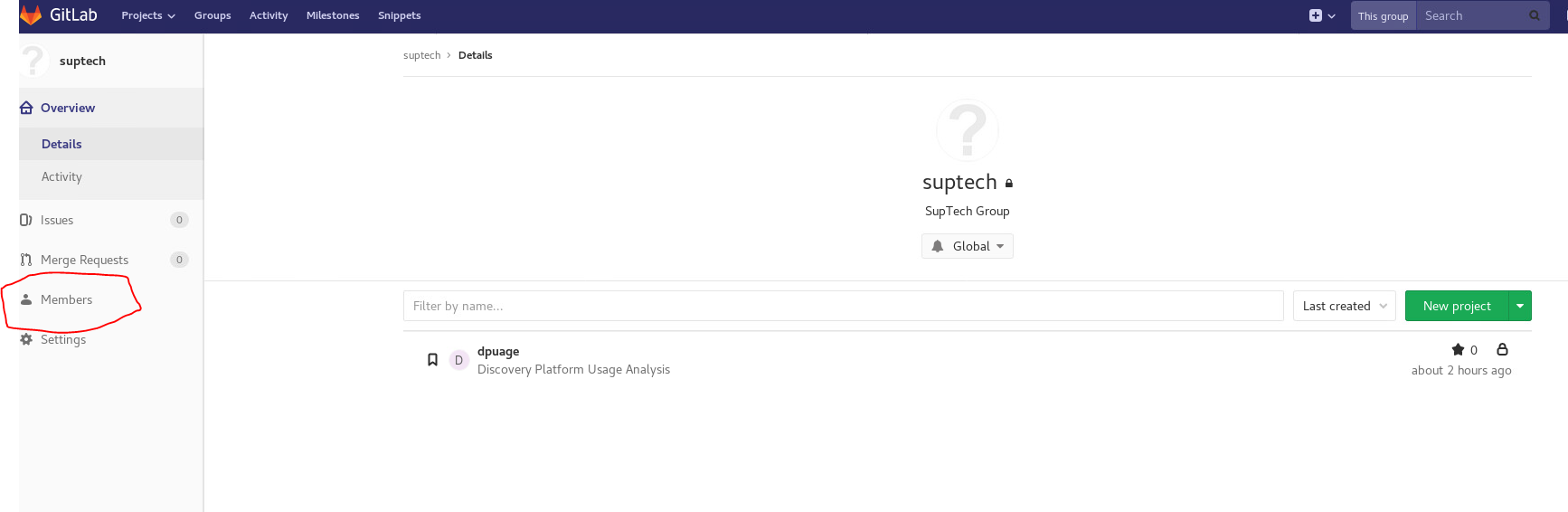


1. Add the following information as mentioned in picture.

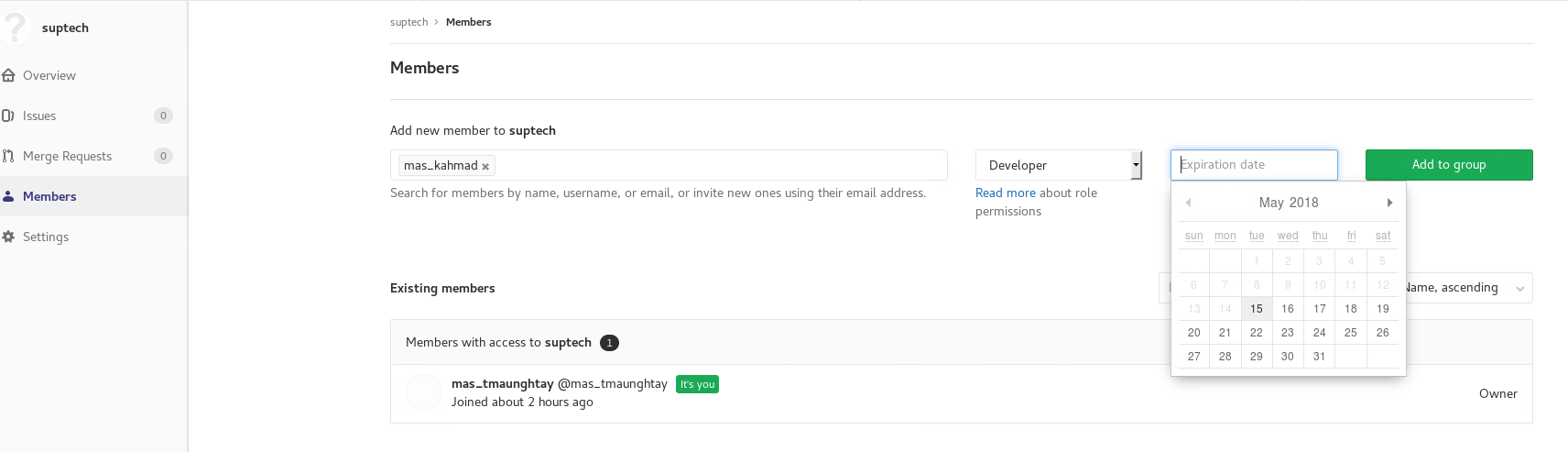


# Add users to a group

1. Click **Groups**, select one **existing group** from your groups.
2. Click **Members**.



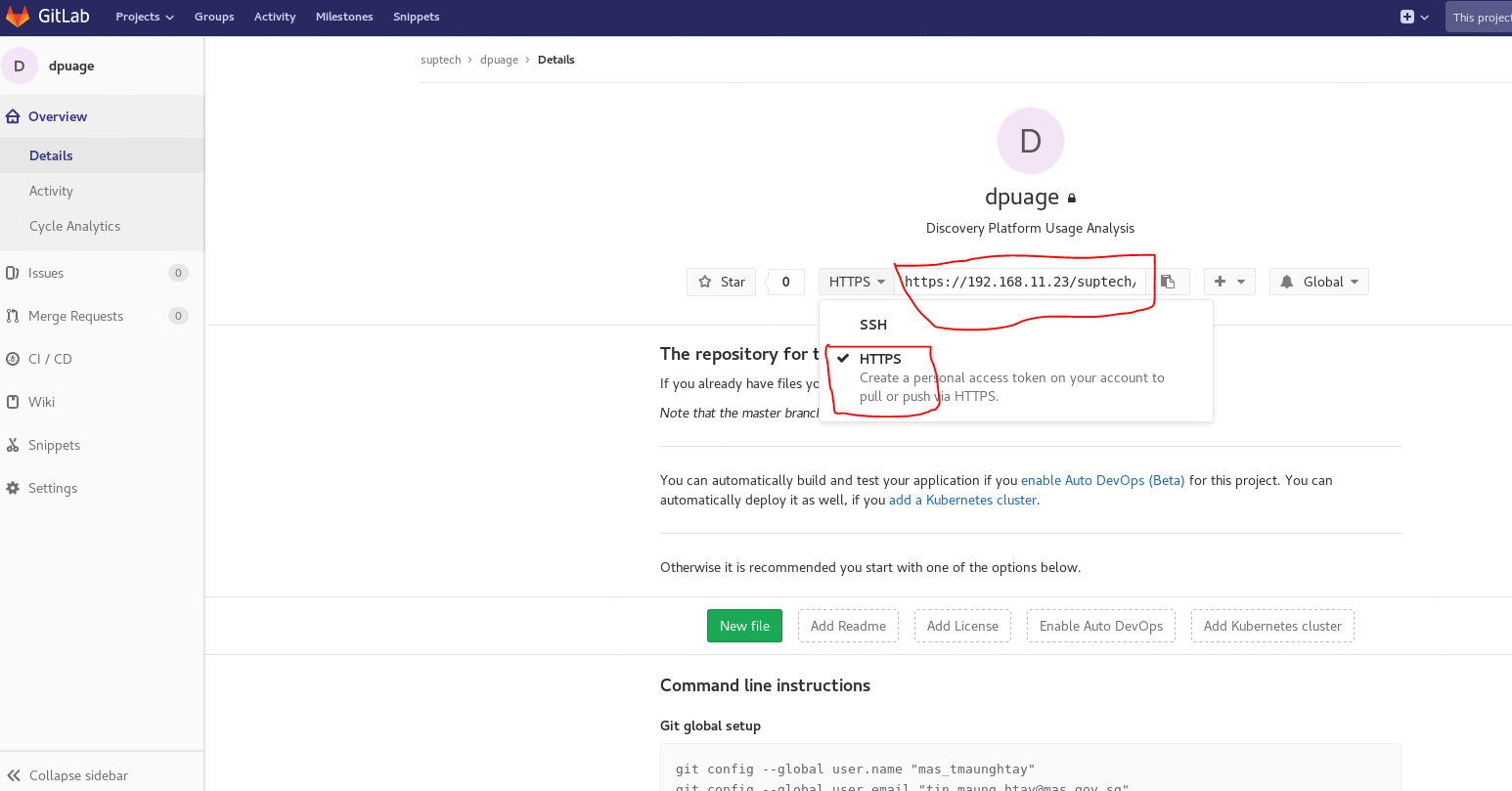
1. Add new member to your group by searching **user** and select him/her, give relevant **role** and set **expiration date** if you want to. Then click **Add to group.**



# Create a new Repository (Linux)

If you like to add or commit new files or updated files into existing project in GitLab, follow the instructions below.

1. Go to your project, select **HTTPS** option as shown and copy **url**.Launch a Linux **Terminal** to execute commands.



1. Clone the repository from GitLab to your local VM. Ignore the $ sign, replace the url with url which you had copied earlier.

**$ git clone https://192.168.11.23/suptech/dpusage.git**

1. Here is to create a **README.md** file and to commit the newly created file. Execute the followings.

$ cd dpusage

$ touch README.md

$ git add README.md

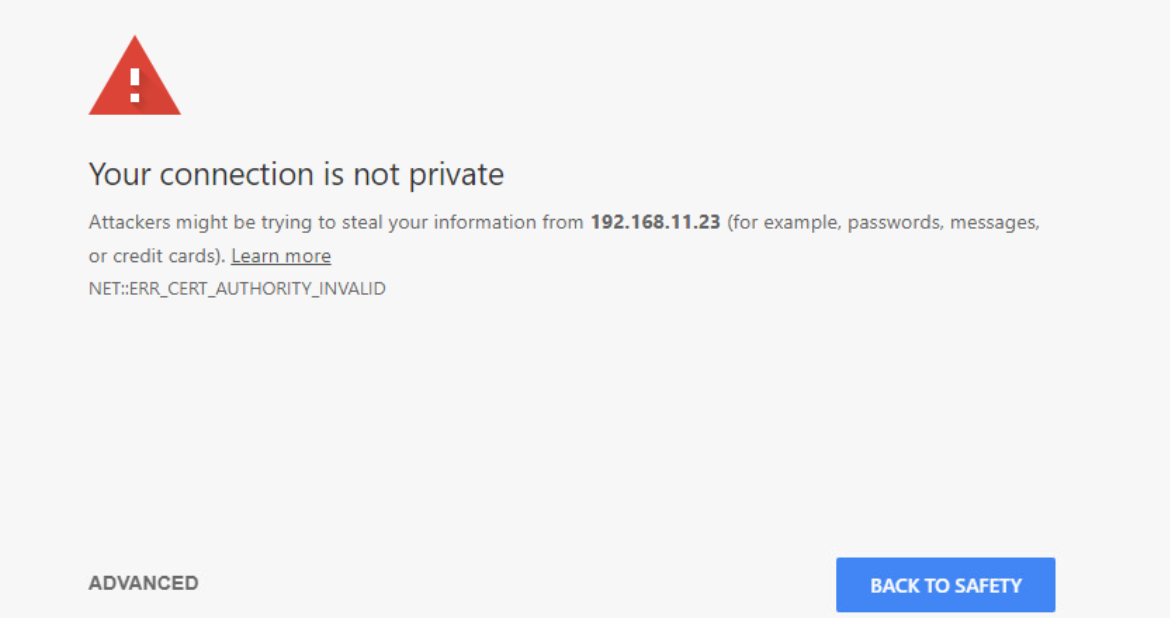
$ git commit –m “my first commit – adding README.md”

$ git push –u origin master

1. Go to Project in GitLab Server and verify if **README.md file** is committed.

# Things to Note:

1. If you have received **Your connection is not private** errorwhen browsing <https://192.168.11.23> on **Chrome**, do the following to fix that issue.
2. Click **ADVANCED**
3. Click **Proceed to 192.168.11.23**



1. If you have encountered **Certificate Issuer is invalid** at your command terminal while cloning project, execute the following.

$ git config --global http.sslVerify false