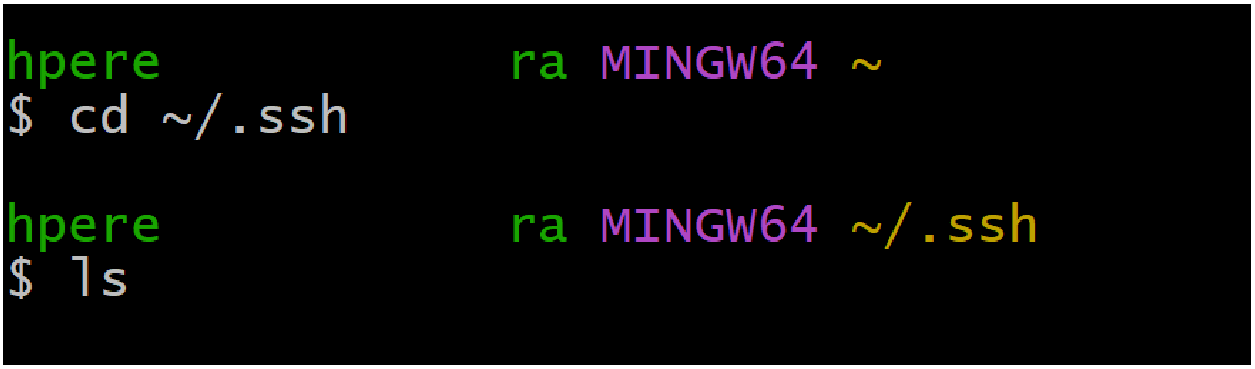
How to create an ssh key for your github account.

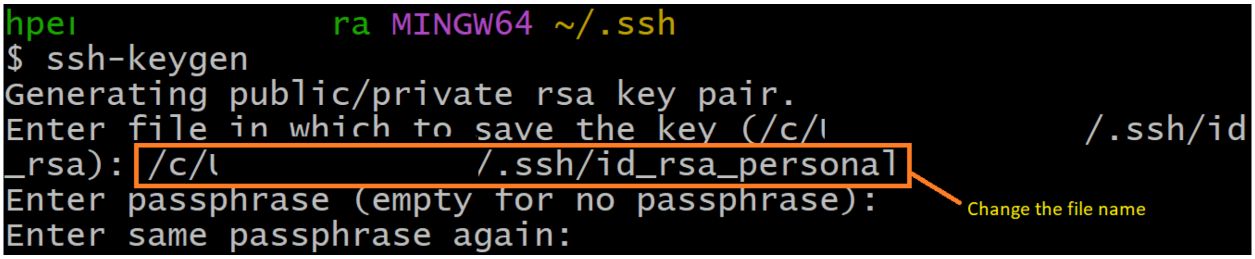
Navigate to .ssh location in windows as below

Type ls command and get all existing file names as below



```
hpere[redacted]ra MINGW64 ~
$ cd ~/.ssh

hpere[redacted]ra MINGW64 ~/.ssh
$ ls
```

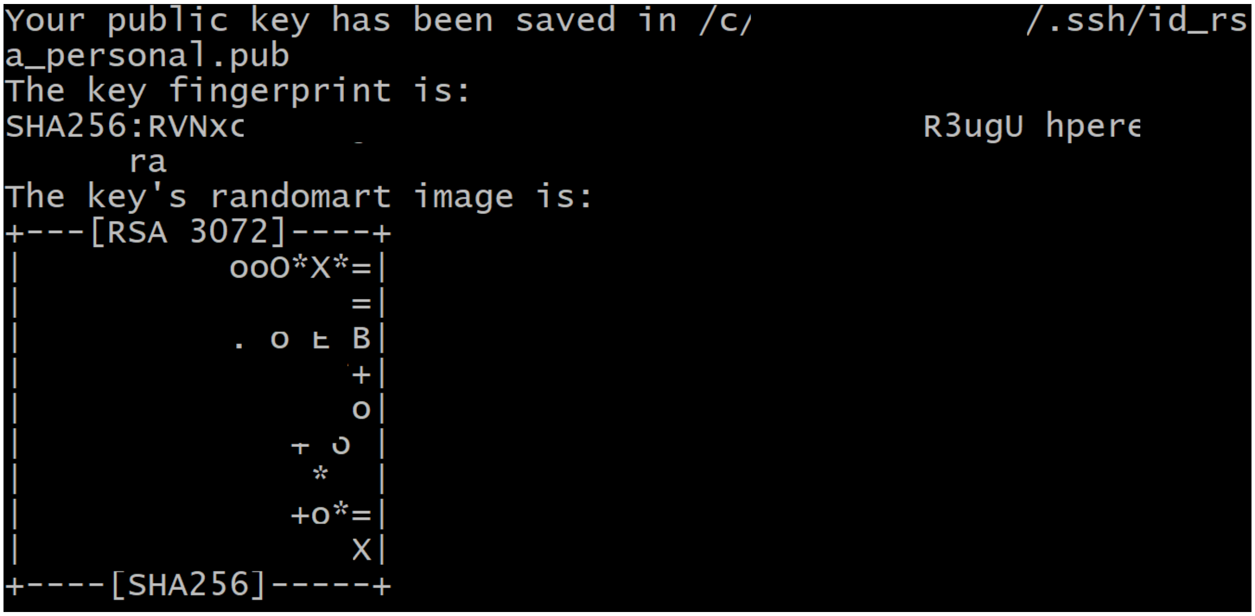
Create a ssh key using the ssh-keygen command. In this tutorial I skipped the passphrase



```
hpe[redacted]ra MINGW64 ~/.ssh
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/[redacted]/.ssh/id
_rsa): /c/[redacted]/.ssh/id_rsa_personal
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

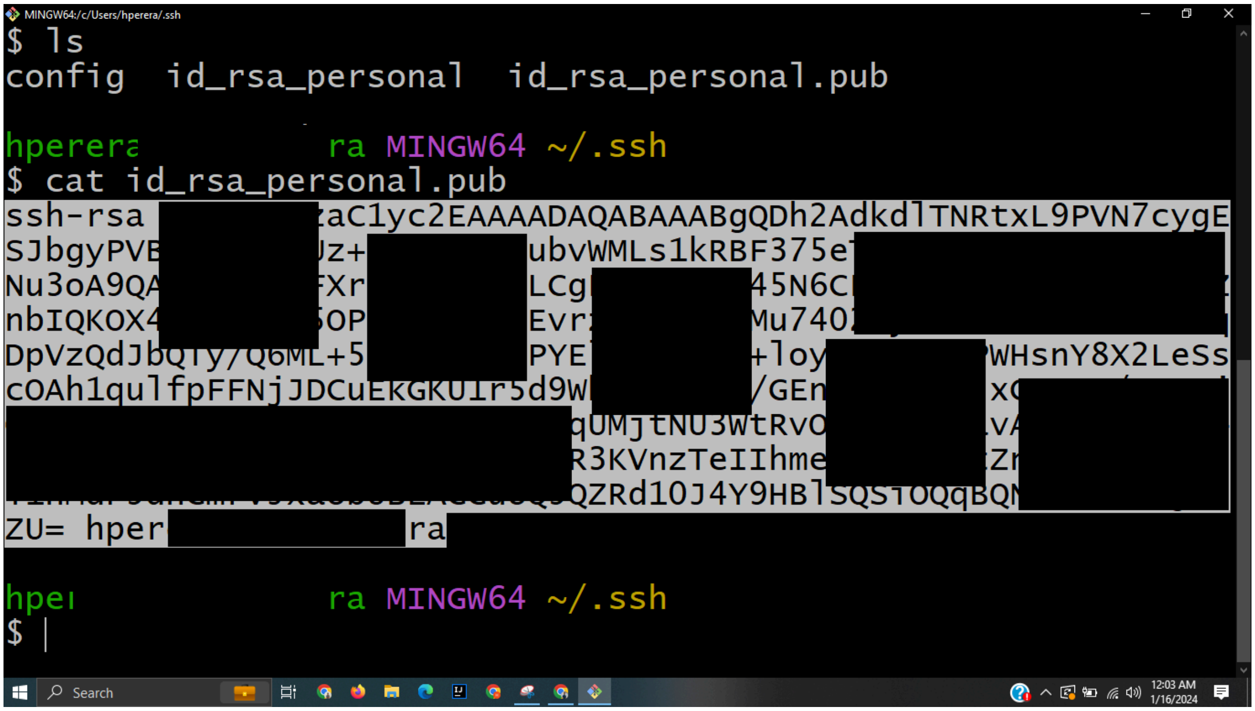
Change the file name

Once you hit “Enter”, you will get the following.



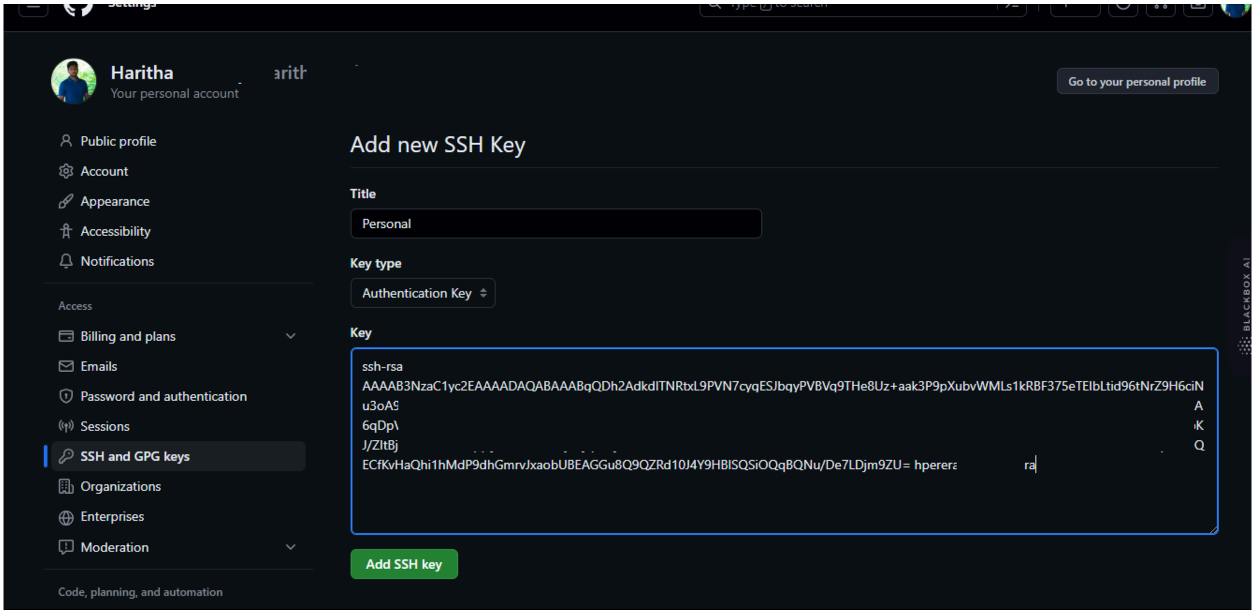
```
Your public key has been saved in /c/[redacted]/.ssh/id_rsa_personal.pub
The key fingerprint is:
SHA256:RVNxc[redacted]R3ugU hpere[redacted]
      ra
The key's randomart image is:
+---[RSA 3072]----+
|          ooO*X*=|
|                =|
|          . o E B|
|                +|
|                o|
|             + o |
|              *  |
|             +o*=|
|                X|
+----[SHA256]-----+
```

Print the public key using cat command on the screen and copy it by selecting.



```
$ ls
config  id_rsa_personal  id_rsa_personal.pub

hperera[redacted]ra MINGW64 ~/.ssh
$ cat id_rsa_personal.pub
ssh-rsa [redacted]aC1yc2EAAAADAQABAAABgQDh2AdklTNRtxL9PVN7cygE
SJbgyPVE[redacted]Jz+[redacted]ubvWMLs1kRBF375e[redacted]
Nu3oA9QA[redacted]FXr[redacted]LCg[redacted]45N6C[redacted]
nbIQKOX4[redacted]5OP[redacted]Evr[redacted]Mu740[redacted]
DpVzQdJb[redacted]Q6ML+5[redacted]PYE[redacted]+loy[redacted]WHsnY8X2LeSs
cOAh1qulfpFFNjJDCuEKGKUIr5d9W[redacted]/GEn[redacted]x[redacted]
[redacted]qUMjtNU3WtRvO[redacted]vA[redacted]
R3KVnzTeIIhme[redacted]Zr
[redacted]QZRd10J4Y9HBlSQSTOQqBQN[redacted]
ZU= hper[redacted]ra

hpe[redacted]ra MINGW64 ~/.ssh
$ |
```

Now go to your github account and go to settings page. Then click on the SSH and GPG keys section, and then press “new ssh key” button



Add new SSH Key

Title: Personal

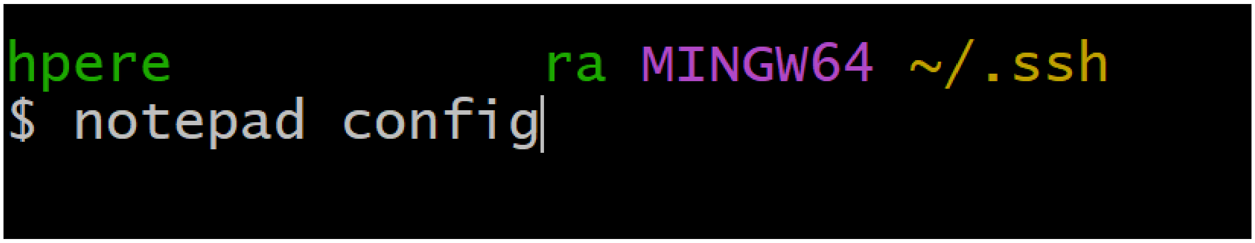
Key type: Authentication Key

Key:

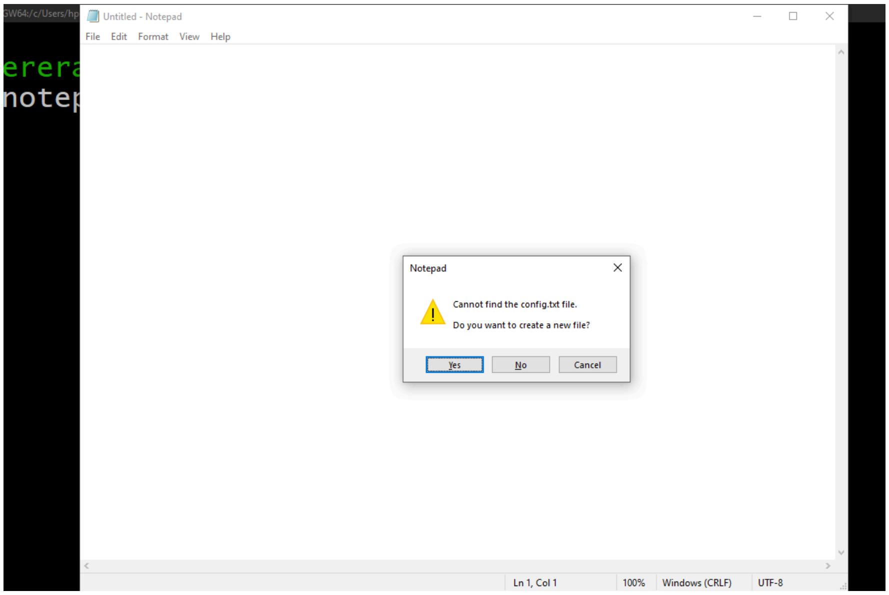
```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDh2AdkdITNRtxL9PVN7cyqESJbqyPVBVq9THe8Uz+aak3P9pXubvWMLs1kRBF375eTEIbLtid96tNrZ9H6ciN
u3oA9[redacted]A
6qDp\[redacted]K
J/ZItBj[redacted]Q
ECfKvHaQhi1hMdP9dhGmrvJxaobUBEAGGu8Q9QZRd10J4Y9HBISQSiOQqBQNu/De7LDjm9ZU= hperera[redacted]ra
```

Put a title and paste your key to the key field.

Create a file called config



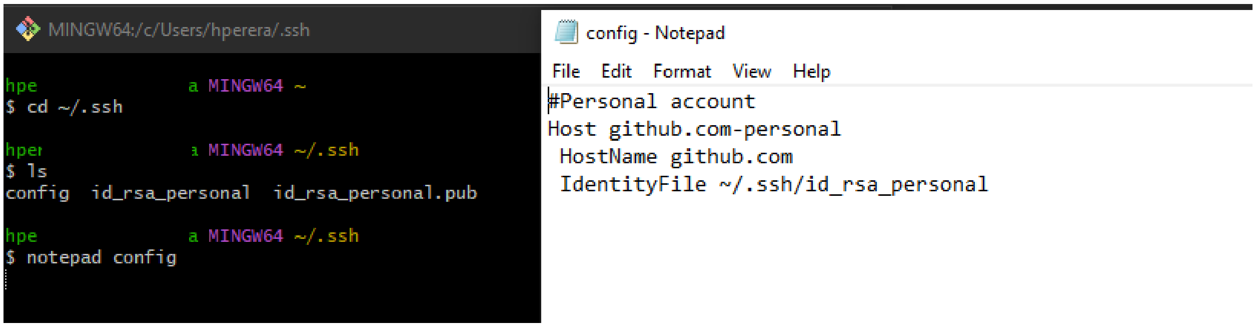
```
hpere[redacted]ra MINGW64 ~/.ssh
$ notepad config
```



Cannot find the config.txt file.

Do you want to create a new file?

You can make the following changes in your config file. Change as per your account configurations.



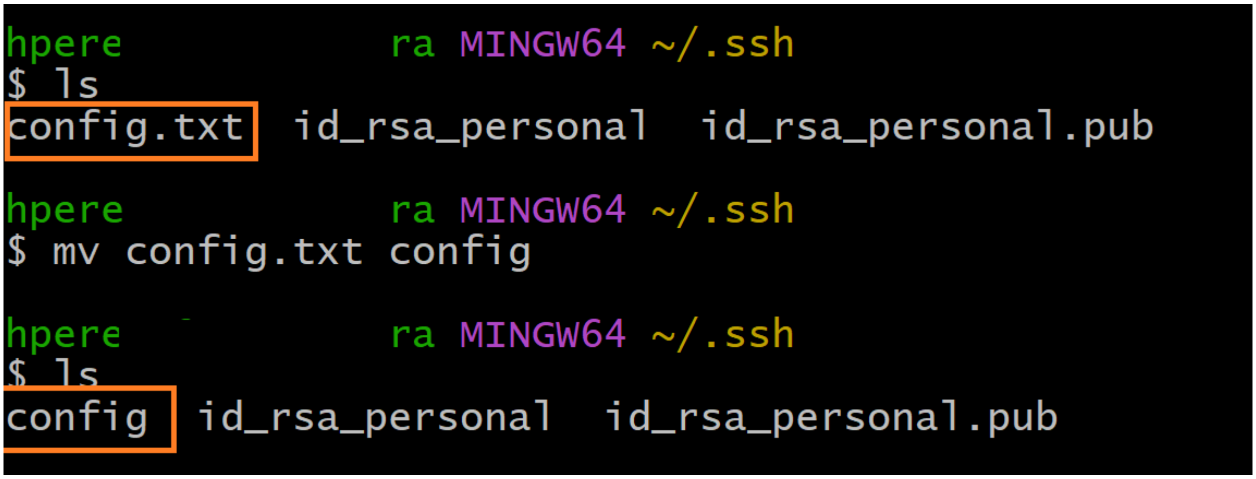
MINGW64:/c/Users/hperera/.ssh

```
hpe            a MINGW64 ~
$ cd ~/.ssh

hper           a MINGW64 ~/.ssh
$ ls
config  id_rsa_personal  id_rsa_personal.pub

hpe            a MINGW64 ~/.ssh
$ notepad config
```

config - Notepad

```
#Personal account
Host github.com-personal
 HostName github.com
 IdentityFile ~/.ssh/id_rsa_personal
```

If the file was created with the “.txt” extension you can remove it by “mv config.txt config” command.



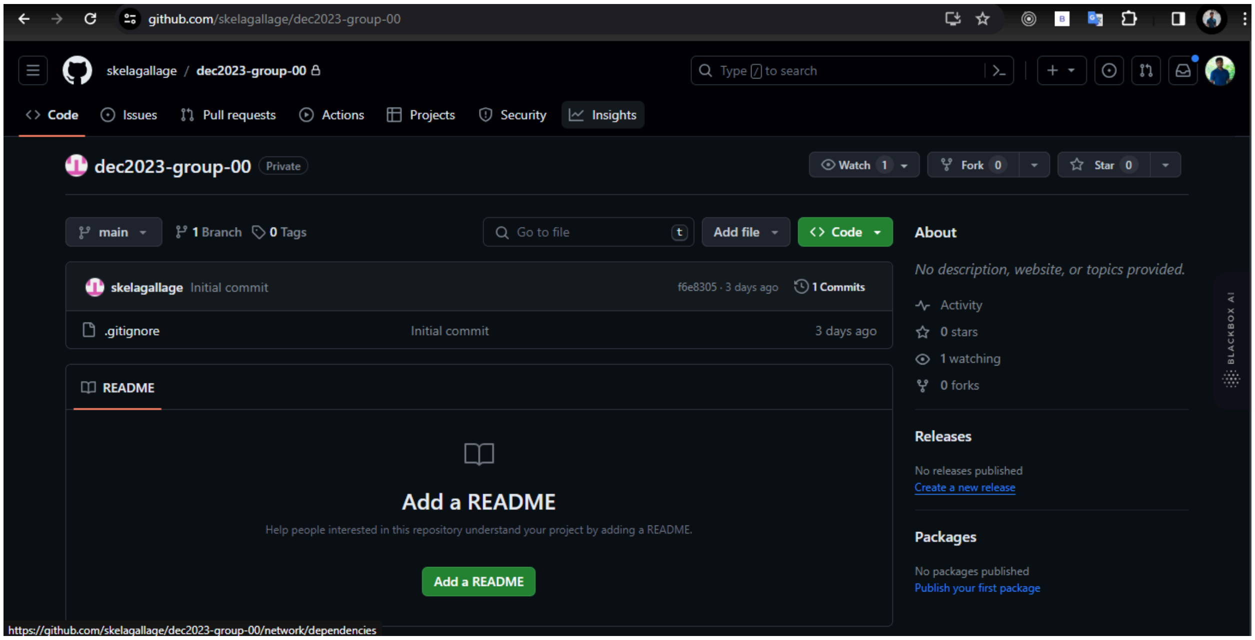
```
hpere               ra MINGW64 ~/.ssh
$ ls
config.txt  id_rsa_personal  id_rsa_personal.pub

hpere               ra MINGW64 ~/.ssh
$ mv config.txt config

hpere               ra MINGW64 ~/.ssh
$ ls
config  id_rsa_personal  id_rsa_personal.pub
```

Now you can use your github account through the ssh key that you have created.

How to use github/ssh to develop your group project

Below is the private github repository

Only one group member does the following.


github.com/skelagallage/dec2023-group-00

skelagallage / dec2023-group-00

dec2023-group-00 Private

main · 1 Branch · 0 Tags

skelagallage Initial commit · f6e8305 · 3 days ago · 1 Commits

.gitignore · Initial commit · 3 days ago

About

No description, website, or topics provided.

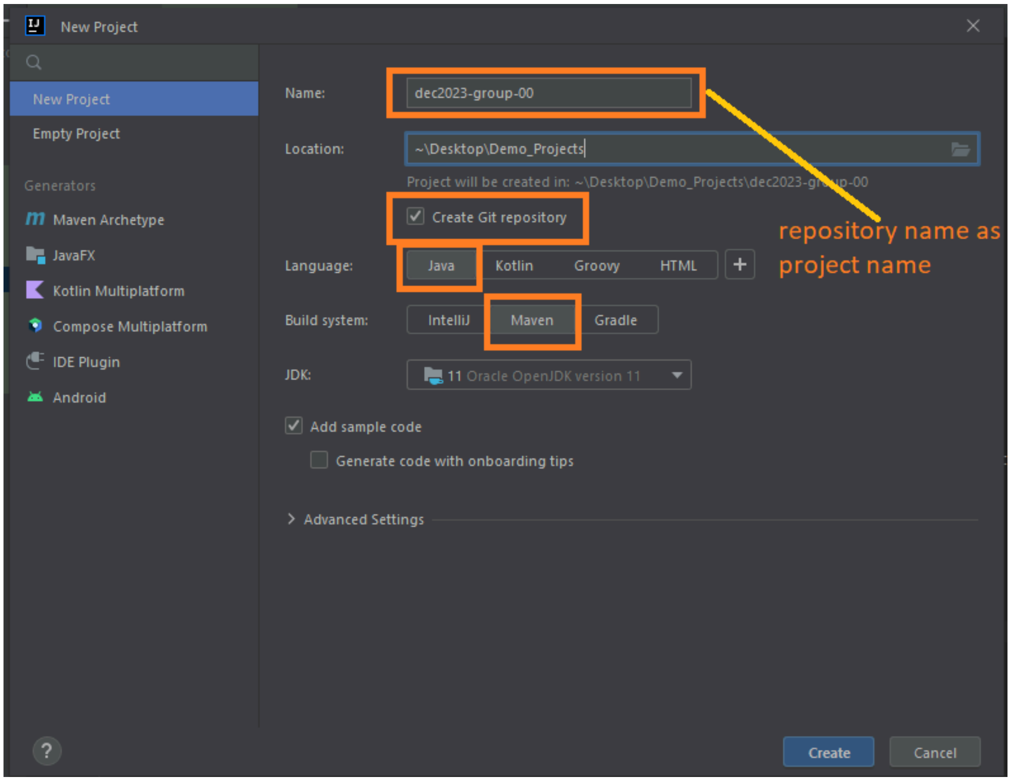
Activity · 0 stars · 1 watching · 0 forks

Releases: No releases published · Create a new release

Packages: No packages published · Publish your first package

https://github.com/skelagallage/dec2023-group-00/network/dependencies

You can create a project as follows.


New Project

Name: dec2023-group-00

Location: ~\Desktop\Demo_Projects

Project will be created in: ~\Desktop\Demo_Projects\dec2023-group-00

☑ Create Git repository

Language: Java

Build system: Maven

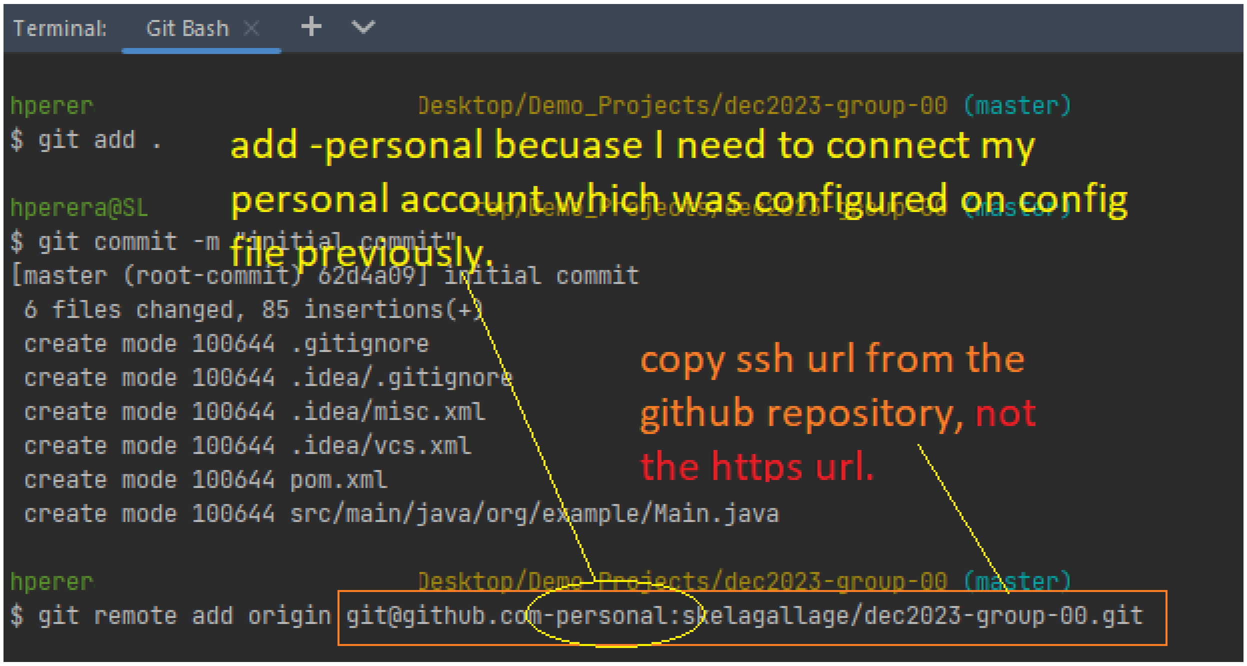
JDK: 11 Oracle OpenJDK version 11

☑ Add sample code

□ Generate code with onboarding tips

repository name as project name

Now you can connect your repository to your local project as follows.
(Do the followings in IDE terminal)

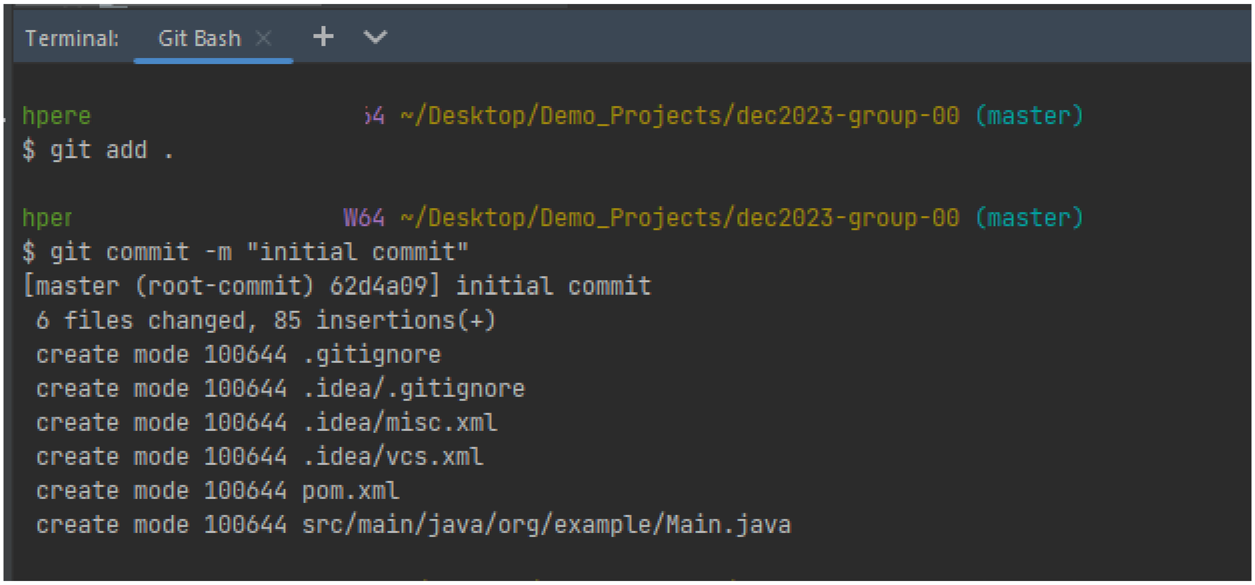


```
hperer                         Desktop/Demo_Projects/dec2023-group-00 (master)
$ git add .

hperera@SL                     top/Demo_Projects/dec2023-group-00 (master)
$ git commit -m "initial commit"
[master (root-commit) 62d4a09] initial commit
 6 files changed, 85 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 .idea/.gitignore
 create mode 100644 .idea/misc.xml
 create mode 100644 .idea/vcs.xml
 create mode 100644 pom.xml
 create mode 100644 src/main/java/org/example/Main.java

hperer                         Desktop/Demo_Projects/dec2023-group-00 (master)
$ git remote add origin git@github.com-personal:skelagallage/dec2023-group-00.git
```

add -personal becuase I need to connect my personal account which was configured on config file previously.

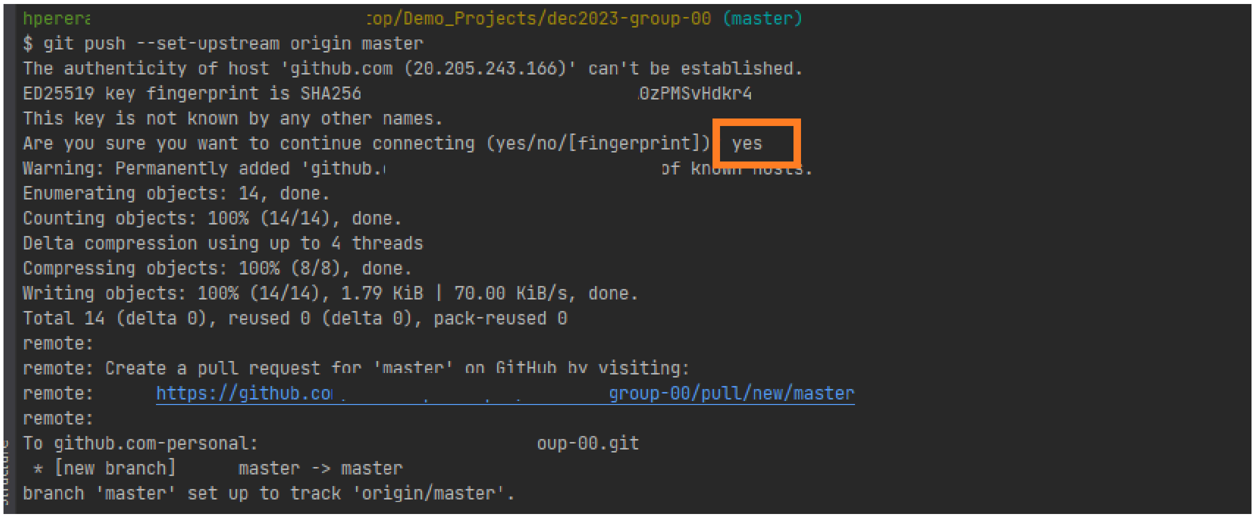
copy ssh url from the github repository, not the https url.

Commit the changes as follows



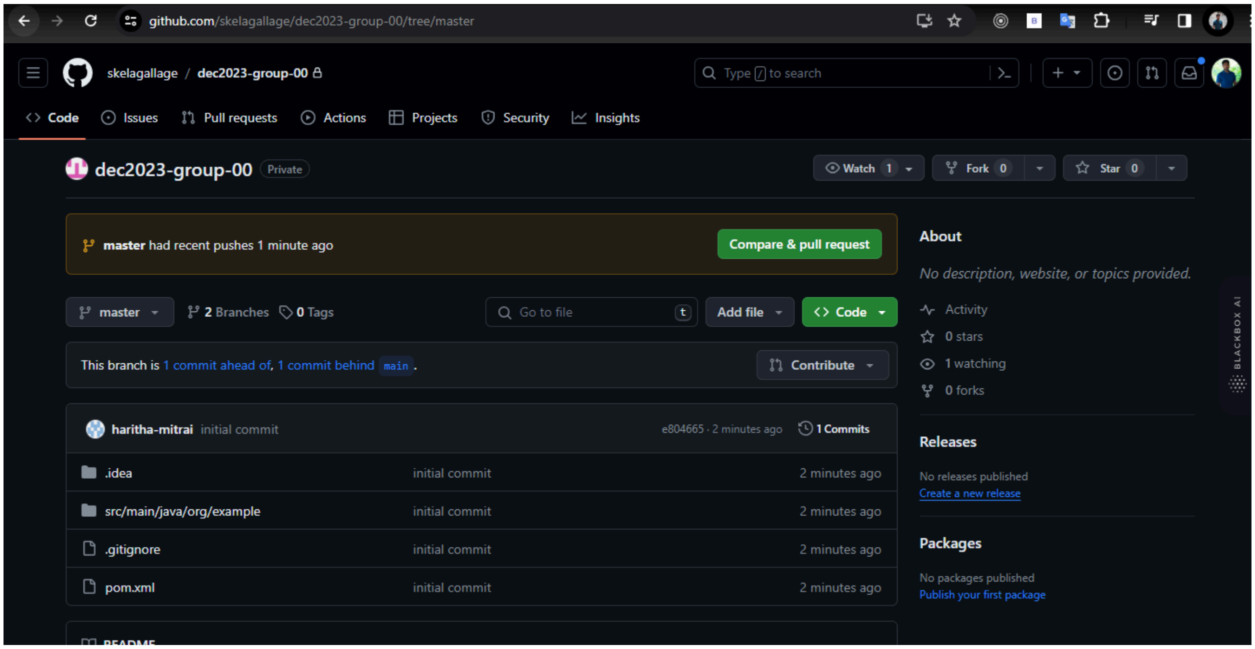
```
hpere                     4 ~/Desktop/Demo_Projects/dec2023-group-00 (master)
$ git add .

hper                    W64 ~/Desktop/Demo_Projects/dec2023-group-00 (master)
$ git commit -m "initial commit"
[master (root-commit) 62d4a09] initial commit
 6 files changed, 85 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 .idea/.gitignore
 create mode 100644 .idea/misc.xml
 create mode 100644 .idea/vcs.xml
 create mode 100644 pom.xml
 create mode 100644 src/main/java/org/example/Main.java
```

You can push the changes to the new master branch.



```
hperera[redacted]:op/Demo_Projects/dec2023-group-00 (master)
$ git push --set-upstream origin master
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256[redacted]0zPMSvHdkr4
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint]) yes
Warning: Permanently added 'github.[redacted]of known hosts.
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 4 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (14/14), 1.79 KiB | 70.00 KiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.co[redacted]group-00/pull/new/master
remote:
To github.com-personal:[redacted]oup-00.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Refresh the project repository and check whether the master branch is there.



master had recent pushes 1 minute ago

This branch is 1 commit ahead of, 1 commit behind main.

haritha-mitrai initial commit e804665 · 2 minutes ago 1 Commits

```
.idea                   initial commit   2 minutes ago
src/main/java/org/example   initial commit   2 minutes ago
.gitignore              initial commit   2 minutes ago
pom.xml                 initial commit   2 minutes ago
```

No description, website, or topics provided.

No releases published
Create a new release

No packages published
Publish your first package

It is better to maintain a development branch (as a stable develop branch) which can be taken from the existing master branch. (git checkout -b develop)

git push --set-upstream develop

Now all group members can take a pull using the “git pull” command and start to do the coding; If You have not yet cloned the project, you can use “git clone git@github.com-skelagallage:skelagallage/dec2023-group-00.git” command to clone the project. (you can change “-skelagalalge” as per your github account config file)

Now all can take a pull and start to do their work in feature branches.
“git checkout -b feature/login_scenarios”

And do the necessary changes.

Once you do the changes in your own feature branch, then push it using “git push --set-upstream origin feature/login_scenarios” at first time. If you do the modification again in the feature branch, you can just push by “git push” command.

Once you merge all the branches to the develop branch then you will face some difficulties. We can discuss those things in the next class.