Github Lab 2: Add SSH key to authenticate with Github Server

Takeaways and Goals:

- Understand public/private key. This is being used everywhere for authentication.
- Github server uses SSH authentication as well, we would need to operate on github server with admin role. Assuming you are a devOps engineer, you would be one of the admin type of person, so your github account would have the permission. Remember that the Github server also have a similar account permission hierachy just like linux, windows operating systems, so it deals with account types (access permissions).
- Once we understand the public/private key we need to understand how to use the `ssh-keygen` tool to generate the public/private key pairs. You always keep the private key and you share the public key with someone. In our case, we add the public key to the github server.

Labs

Step 1 Validate our docker image is good

Once we work in the company, we are also preparing a good docker image, some engineer who works on upstream team would give you a docker image, but that does not mean that it will have everything you need.

In this lab, since it is a linux docker image of Jenkin, so we would need to at least need a couple of things,

- Java (JVM) – We need this to run Jenkin app itself, which is written in Java.
- Javac (Java compiler) – we need to build and compile maven project.
- ssh-keygen

First of all, we need to make sure our Jenkin docker image is pulled down (downloaded from docker.io) successfully. It might not be running, so we need to check with `ls -a` command.

```
bash-3.2$ docker container ls -a
CONTAINER ID   IMAGE                                  COMMAND
CREATED         STATUS                        PORTS
NAMES
bf043ccd8869   jenkins/jenkins:lts                    "/usr/bin/tini
-- /u…"   2 weeks ago     Exited (255) 8 days ago      0.0.0.0:8080-
>8080/tcp, 0.0.0.0:50000->50000/tcp
nervous_cori
```

We need to start the Jenkin container from the previous lab,

```
bash-3.2$ docker container start bf043ccd8869
bf043ccd8869
```

Now we check if our Jenkin container is running,

```
bash-3.2$ docker container ls
CONTAINER ID   IMAGE                 COMMAND
CREATED        STATUS          PORTS
NAMES
bf043ccd8869   jenkins/jenkins:lts   "/usr/bin/tini -- /u…"   2
weeks ago   Up 48 seconds   0.0.0.0:8080->8080/tcp, 0.0.0.0:50000-
>50000/tcp   nervous_cori
```

Next, we would `ssh` into the running docker container.

```
bash-3.2$ docker exec -it bf043ccd8869 /bin/bash
```

First we need to check we have enough binary tools in this docker image, here we would need to use the 'ssh-keygen', and we can check it by doing the following,

```
jenkins@bf043ccd8869:/$ which ssh-keygen
/usr/bin/ssh-keygen
```

Once you type the following, you see the following and it is ok.
Press “ctrol + c” to terminate it in your VM(ubuntu or Kali).

```
jenkins@bf043ccd8869:/$ ssh-keygen
```

```
Generating public/private rsa key pair.
Enter file in which to save the key (/var/jenkins_home/.ssh/id_rsa):
```

Now let us check Java and Javac,

```
jenkins@bf043ccd8869:/$ which java
/opt/java/openjdk/bin/java

jenkins@bf043ccd8869:/$ which javac
/opt/java/openjdk/bin/javac

jenkins@bf043ccd8869:/$ javac --version
javac 21.0.8
```

Now everything looks good.

Next step, let us verify that our login account is not root. It is better to use a dedicated account that runs Jenkin.

```
jenkins@bf043ccd8869:/$ whoami
jenkins

jenkins@bf043ccd8869:/$ ps axuw|grep java
jenkins      7  3.2 24.0 5222292 489560 ?      Sl   20:00   0:48
java -Duser.home=/var/jenkins_home
-Djenkins.model.Jenkins.slaveAgentPort=50000
-Dhudson.lifecycle=hudson.lifecycle.ExitLifecycle -jar
/usr/share/jenkins/jenkins.war
```

Step 2 Generating SSH key pairs

Normally we would need to do the following to generate a key pairs, but in our case, we need to check if our repo already had it.

So please do not generate a new pair of keys just yet. Scroll down to pass the following section and see how we handle this.

```
jenkins@bf043ccd8869:/$ ssh-keygen -t rsa -b 4096 -C
"xlics05@example.com"
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/var/jenkins_home/.ssh/id_rsa):
Created directory '/var/jenkins_home/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/jenkins_home/.ssh/id_rsa
Your public key has been saved in /var/jenkins_home/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:r9ymp6BU6SyFn+X0NrB4dc0lpWwSJbRwlnONS3f5nrc
xlics05@example.com
The key's randomart image is:
+---[RSA 4096]----+
|          ..*o.oo|
|           +o=+++|
|            o+=o+|
|      . .     =.o.|
|     . + S . . o..|
|      * * * .   .o|
|     o B + =     o|
|    . o + +o.   E |
|     .   ==.      |
+----[SHA256]-----+

jenkins@bf043ccd8869:/$ cd ~
jenkins@bf043ccd8869:~$ ls
caches             hudson.plugins.gradle.Gradle.xml
jenkins.model.JenkinsLocationConfiguration.xml
org.jenkinsci.plugins.gitclient.JGitApacheTool.xml
    secret.key.not-so-secret  war
config.xml         hudson.tasks.Ant.xml
jenkins.mvn.GlobalMavenConfig.xml
org.jenkinsci.plugins.gitclient.JGitTool.xml     secrets
workspace
copy_reference_file.log    hudson.tasks.Maven.xml
jenkins.telemetry.Correlator.xml
org.jenkinsci.plugins.workflow.flow.FlowExecutionList.xml  tools
fingerprints           identity.key.enc               jobs
             plugins                           updates
hudson.model.UpdateCenter.xml
    jenkins.install.InstallUtil.lastExecVersion  logs
         queue.xml.bak                     userContent
hudson.plugins.git.GitTool.xml jenkins.install.UpgradeWizard.state
nodeMonitors.xml                    secret.key
    users
jenkins@bf043ccd8869:~$ ls -al
total 168
drwxr-xr-x 20 jenkins jenkins  4096 Oct 29 20:36 .
drwxr-xr-x  1 root    root     4096 Sep 17 10:04 ..
drwxr-xr-x  3 jenkins jenkins  4096 Oct 15 20:10 .cache
```

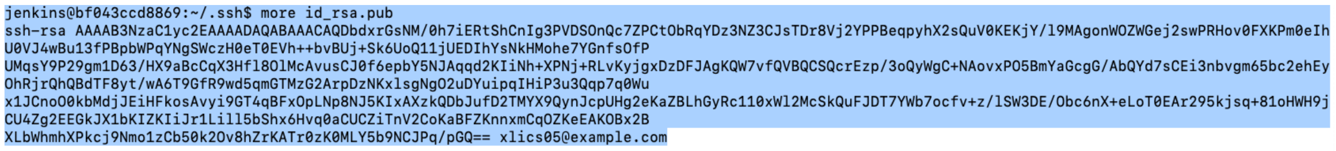
```
drwxr-xr-x  3 jenkins jenkins  4096 Oct 17 20:16 .config
drwxr-xr-x  3 jenkins jenkins  4096 Oct 15 20:13 .groovy
drwxr-xr-x  3 jenkins jenkins  4096 Oct 15 20:10 .java
-rw-r--r--  1 jenkins jenkins     0 Oct 29 20:00 .lastStarted
drwxr-xr-x  3 jenkins jenkins  4096 Oct 17 20:19 .m2
-rw-r--r--  1 jenkins jenkins     1 Oct 18 01:04 .owner
drwx------  2 jenkins jenkins  4096 Oct 29 20:36 .ssh
drwxr-xr-x  4 jenkins jenkins  4096 Oct 17 20:19 caches
-rw-r--r--  1 jenkins jenkins  1663 Oct 29 20:00 config.xml
-rw-r--r--  1 jenkins jenkins   324 Oct 29 20:00
copy_reference_file.log
drwxr-xr-x  3 jenkins jenkins  4096 Oct 17 20:19 fingerprints
-rw-r--r--  1 jenkins jenkins   156 Oct 29 20:00
hudson.model.UpdateCenter.xml
-rw-r--r--  1 jenkins jenkins   370 Oct 17 20:18
hudson.plugins.git.GitTool.xml
-rw-r--r--  1 jenkins jenkins   194 Oct 17 20:18
hudson.plugins.gradle.Gradle.xml
-rw-r--r--  1 jenkins jenkins   161 Oct 17 20:18
hudson.tasks.Ant.xml
-rw-r--r--  1 jenkins jenkins   571 Oct 17 20:18
hudson.tasks.Maven.xml
-rw-------  1 jenkins jenkins  1680 Oct 15 20:13 identity.key.enc
-rw-r--r--  1 jenkins jenkins     7 Oct 29 20:00
jenkins.install.InstallUtil.lastExecVersion
-rw-r--r--  1 jenkins jenkins     7 Oct 15 20:25
jenkins.install.UpgradeWizard.state
-rw-r--r--  1 jenkins jenkins   179 Oct 15 20:24
jenkins.model.JenkinsLocationConfiguration.xml
-rw-r--r--  1 jenkins jenkins   247 Oct 17 20:18
jenkins.mvn.GlobalMavenConfig.xml

-rw-r--r--  1 jenkins jenkins   171 Oct 15 20:10
jenkins.telemetry.Correlator.xml
drwxr-xr-x  4 jenkins jenkins  4096 Oct 15 20:26 jobs
drwxr-xr-x  2 jenkins jenkins  4096 Oct 15 20:14 logs
-rw-r--r--  1 jenkins jenkins  1037 Oct 29 20:00 nodeMonitors.xml
-rw-r--r--  1 jenkins jenkins   255 Oct 17 20:18
org.jenkinsci.plugins.gitclient.JGitApacheTool.xml
-rw-r--r--  1 jenkins jenkins   243 Oct 17 20:18
org.jenkinsci.plugins.gitclient.JGitTool.xml
-rw-r--r--  1 jenkins jenkins    46 Oct 17 20:19
org.jenkinsci.plugins.workflow.flow.FlowExecutionList.xml
drwxr-xr-x 91 jenkins jenkins 12288 Oct 15 20:13 plugins
-rw-r--r--  1 jenkins jenkins   258 Oct 17 20:20 queue.xml.bak
-rw-r--r--  1 jenkins jenkins    64 Oct 15 20:10 secret.key
-rw-r--r--  1 jenkins jenkins     0 Oct 15 20:10 secret.key.not-so-
secret
```

```
drwx------  2 jenkins jenkins  4096 Oct 17 20:16 secrets
drwxr-xr-x  3 jenkins jenkins  4096 Oct 17 20:19 tools
drwxr-xr-x  2 jenkins jenkins  4096 Oct 29 20:00 updates
drwxr-xr-x  2 jenkins jenkins  4096 Oct 15 20:10 userContent
drwxr-xr-x  3 jenkins jenkins  4096 Oct 15 20:10 users
drwxr-xr-x 10 jenkins jenkins  4096 Oct 15 20:10 war
drwxr-xr-x  5 jenkins jenkins  4096 Oct 17 20:19 workspace
```

Now we noticed we have a .ssh directory,

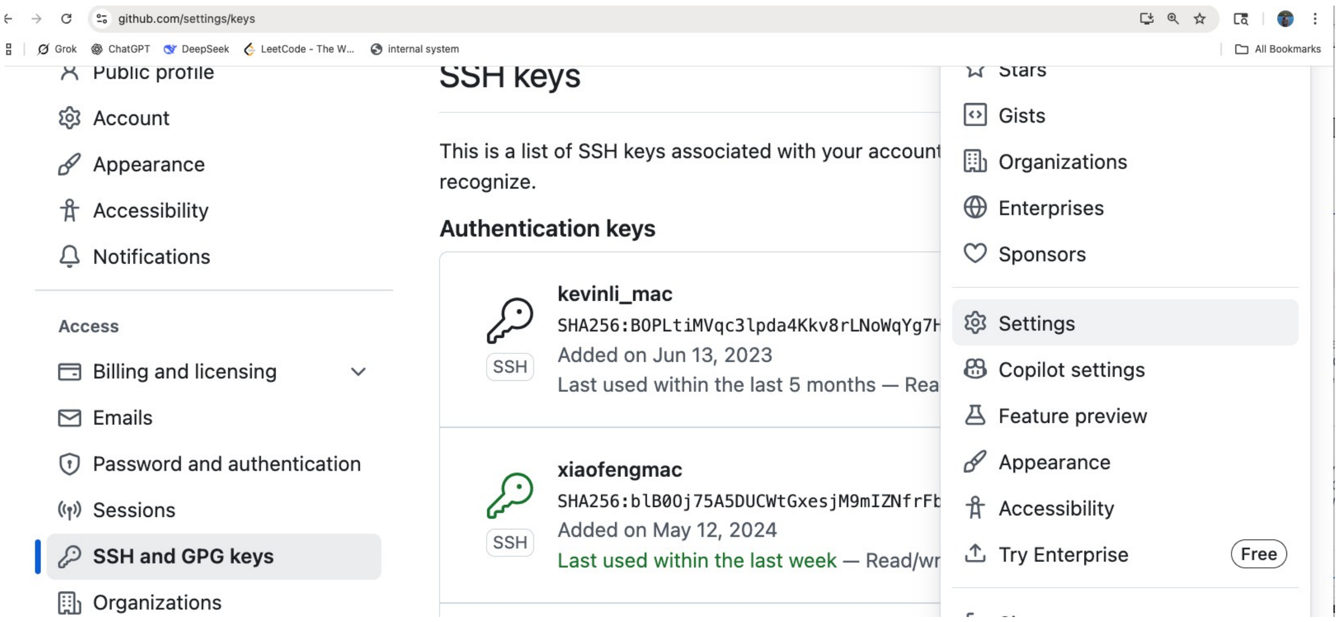
```
jenkins@bf043ccd8869:~/.ssh$ ls
id_rsa    id_rsa.pub
```



```
jenkins@bf043ccd8869:~/.ssh$ more id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDbdxrGsNM/0h7iERtShCnIg3PVDSOnQc7ZPCtObRqYDz3NZ3CJsTDr8Vj2YPPBeqpyhX2sQuV0KEKjY/l9MAgonWOZWGej2swPRHov0FXKPm0eIh
U0VJ4wBu13fPBpbWPqYNgSWczH0eT0EVh++bvBUj+Sk6UoQ11jUEDIhYsNkHMohe7YGnfsOfP
UMqsY9P29gm1D63/HX9aBcCqX3Hfl8OlMcAvusCJ0f6epbY5NJAqqd2KIiNh+XPNj+RLvKyjgxDzDFJAgKQW7vfQVBQCSQcrEzp/3oQyWgC+NAovxPO5BmYaGcgG/AbQYd7sCEi3nbvgm65bc2ehEy
OhRjrQhQBdTF8yt/wA6T9GfR9wd5qmGTMzG2ArpDzNKxlsgNgO2uDYuipqIHiP3u3Qqp7q0Wu
x1JCnoO0kbMdjJEiHFkosAvyi9GT4qBFxOpLNp8NJ5KIxAXzkQDbJufD2TMYX9QynJcpUHg2eKaZBLhGyRc110xWl2McSkQuFJDT7YWb7ocfv+z/lSW3DE/Obc6nX+eLoT0EAr295kjsq+81oHWH9j
CU4Zg2EEGkJX1bKIZKIiJr1Lill5bShx6Hvq0aCUCZiTnV2CoKaBFZKnnxmCqOZKeEAKOBx2B
XLbWhmhXPkcj9Nmo1zCb50k2Ov8hZrKATr0zK0MLY5b9NCJPq/pGQ== xlics05@example.com
```

Add SSH Key to GITHUB remote server

Let us click on your Avatar and click on "Settings",



SSH keys

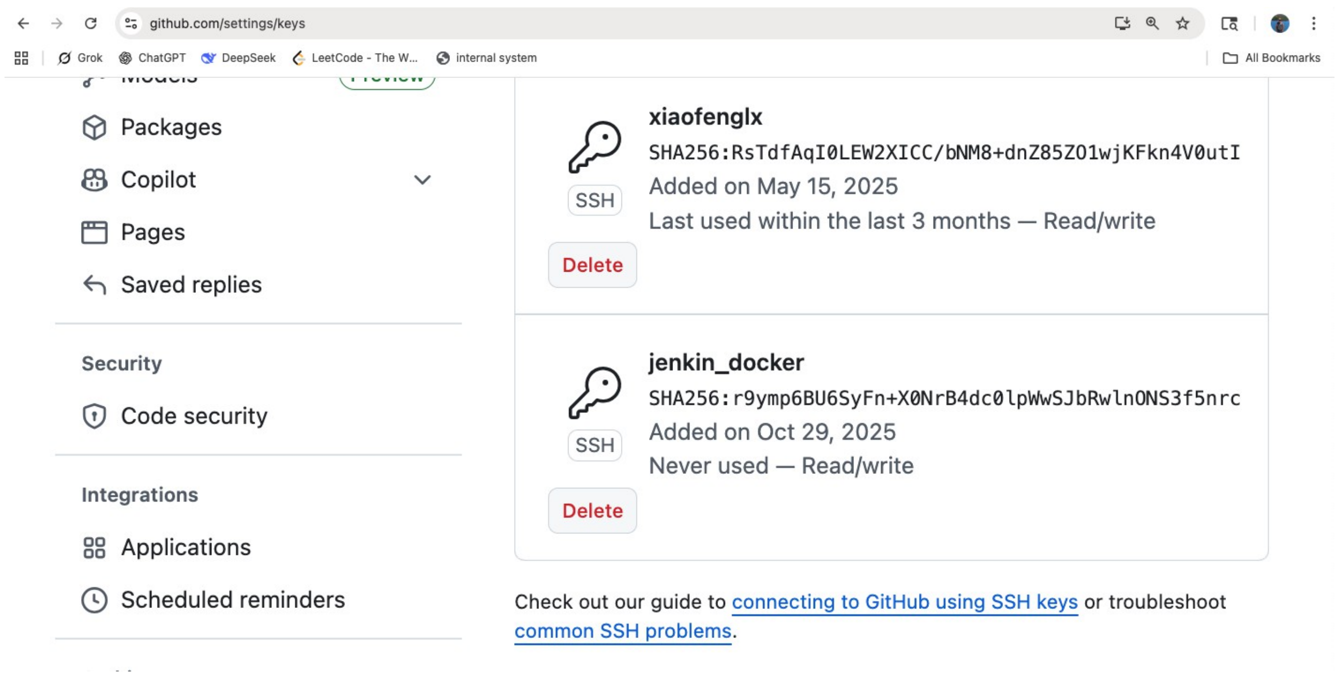
This is a list of SSH keys associated with your account
recognize.

Authentication keys

kevinli_mac
SHA256:BOPLtiMVqc3lpda4Kkv8rLNoWqYg7H
Added on Jun 13, 2023
Last used within the last 5 months — Rea

xiaofengmac
SHA256:blB00j75A5DUCWtGxesjM9mIZNfrFb
Added on May 12, 2024
Last used within the last week — Read/wr

Confirm that the new key was added like the following,



xiaofenglx
SHA256:RsTdfAqI0LEW2XICC/bNM8+dnZ85Z01wjKFkn4V0utI
Added on May 15, 2025
Last used within the last 3 months — Read/write

jenkin_docker
SHA256:r9ymp6BU6SyFn+X0NrB4dc0lpWwSJbRwln0NS3f5nrc
Added on Oct 29, 2025
Never used — Read/write

Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.

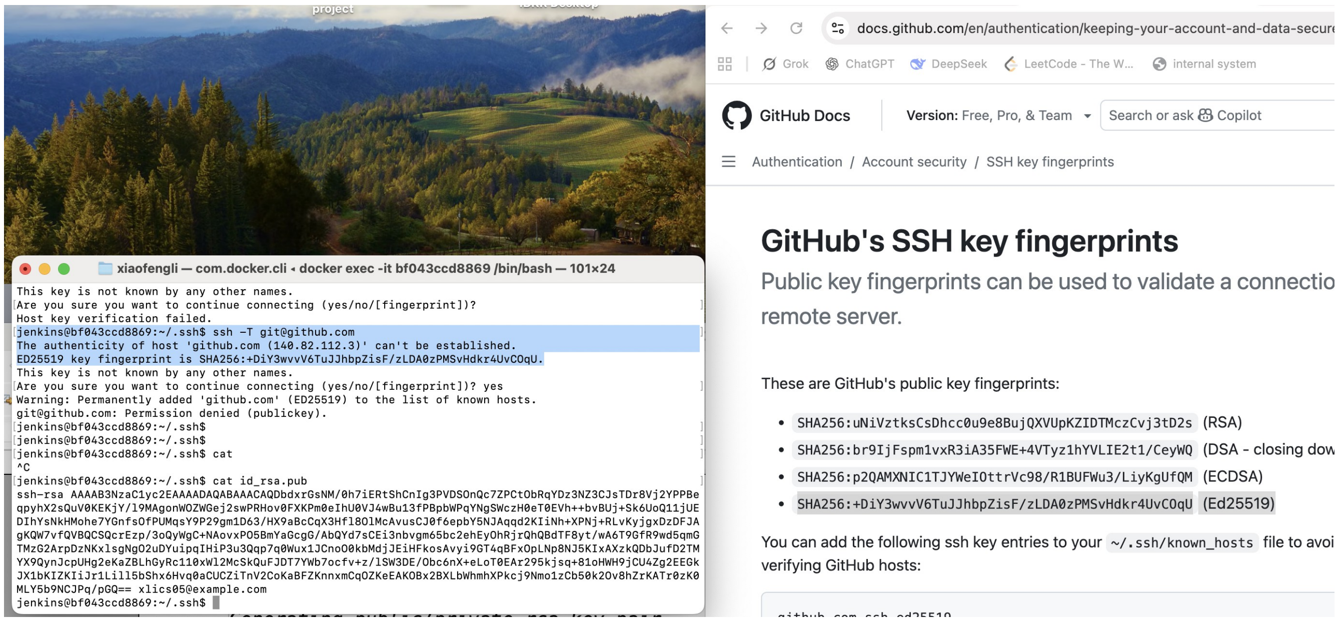
Now let us valid if this public/private key authentication can be established successfully.

For github, it actually provides the hash so it confirms with you after you added your public key in github server. The following is the link from github official documentation.

https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/githubs-ssh-key-fingerprints

From our side, all you need to do is to run the following commands,

```
jenkins@bf043ccd8869:~/.ssh$ ssh -T git@github.com
The authenticity of host 'github.com (140.82.112.3)' can't be established.
ED25519 key fingerprint is
SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
```


xiaofengli — com.docker.cli ◂ docker exec -it bf043ccd8869 /bin/bash — 101×24
docs.github.com/en/authentication/keeping-your-account-and-data-secure
Grok
ChatGPT
DeepSeek
LeetCode - The W...
internal system
GitHub Docs
Version: Free, Pro, & Team
Search or ask Copilot
Authentication / Account security / SSH key fingerprints

```
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Host key verification failed.
jenkins@bf043ccd8869:~/.ssh$ ssh -T git@github.com
The authenticity of host 'github.com (140.82.112.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
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.com' (ED25519) to the list of known hosts.
git@github.com: Permission denied (publickey).
jenkins@bf043ccd8869:~/.ssh$
jenkins@bf043ccd8869:~/.ssh$
jenkins@bf043ccd8869:~/.ssh$ cat
^C
jenkins@bf043ccd8869:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDbdxrGsNM/0h7iERtShCnIg3PVDSOnQc7ZPCtObRqYDz3NZ3CJsTDr8Vj2YPPBe
qpyhX2sQuV0KEKjY/l9MAgonWOZWGej2swPRHov0FXKPm0eIhU0VJ4wBu13fPBpbWPqYNgSWczH0eT0EVh++bvBUj+Sk6UoQ11jUE
DIhYsNkHMohe7YGnfsOfPUMqsY9P29gm1D63/HX9aBcCqX3Hfl8OlMcAvusCJ0f6epbY5NJAqqd2KIiNh+XPNj+RLvKyjgxDzDFJA
gKQW7vfQVBQCSQcrEzp/3oQyWgC+NAovxPO5BmYaGcgG/AbQYd7sCEi3nbvgm65bc2ehEyOhRjrQhQBdTF8yt/wA6T9GfR9wd5qmG
TMzG2ArpDzNKxlsgNgO2uDYuipqIHiP3u3Qqp7q0Wux1JCnoO0kbMdjJEiHFkosAvyi9GT4qBFxOpLNp8NJ5KIxAXzkQDbJufD2TM
YX9QynJcpUHg2eKaZBLhGyRc110xWl2McSkQuFJDT7YWb7ocfv+z/lSW3DE/Obc6nX+eLoT0EAr295kjsq+81oHWH9jCU4Zg2EEGk
JX1bKIZKIiJr1Lill5bShx6Hvq0aCUCZiTnV2CoKaBFZKnnxmCqOZKeEAKOBx2BXLbWhmhXPkcj9Nmo1zCb50k2Ov8hZrKATr0zK0
MLY5b9NCJPq/pGQ== xlics05@example.com
jenkins@bf043ccd8869:~/.ssh$
```

GitHub's SSH key fingerprints

Public key fingerprints can be used to validate a connectio
remote server.

These are GitHub's public key fingerprints:

- `SHA256:uNiVztksCsDhcc0u9e8BujQXVUpKZIDTMczCvj3tD2s` (RSA)
- `SHA256:br9IjFspm1vxR3iA35FWE+4VTyz1hYVLIE2t1/CeyWQ` (DSA - closing dow
- `SHA256:p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM` (ECDSA)
- `SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU` (Ed25519)

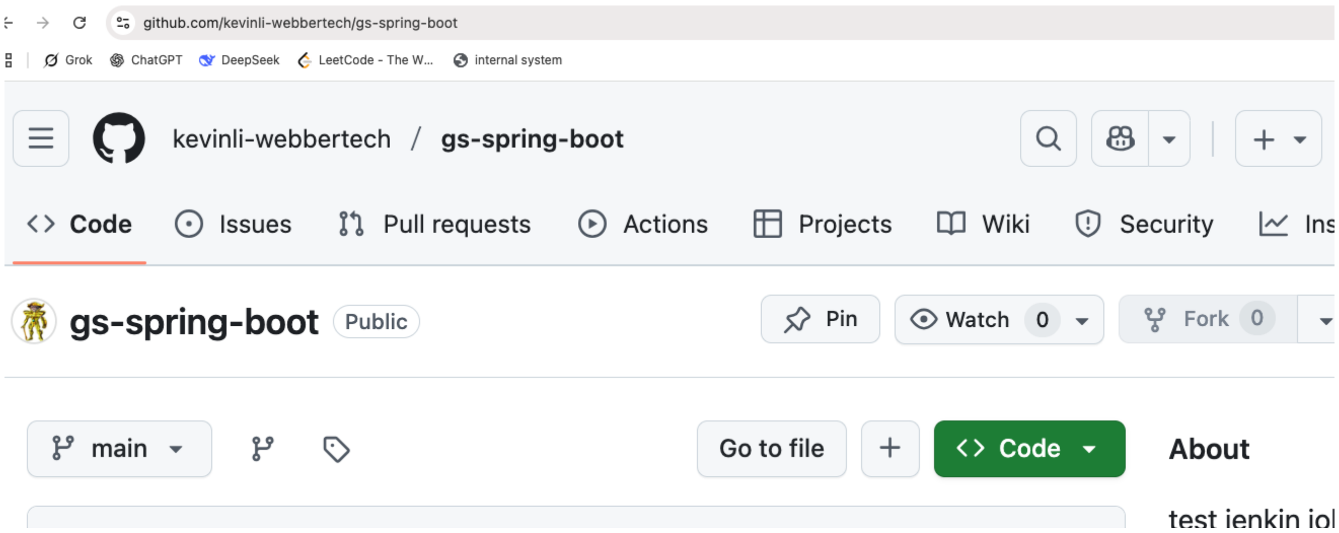
You can add the following ssh key entries to your `~/.ssh/known_hosts` file to avoi
verifying GitHub hosts:

The line above

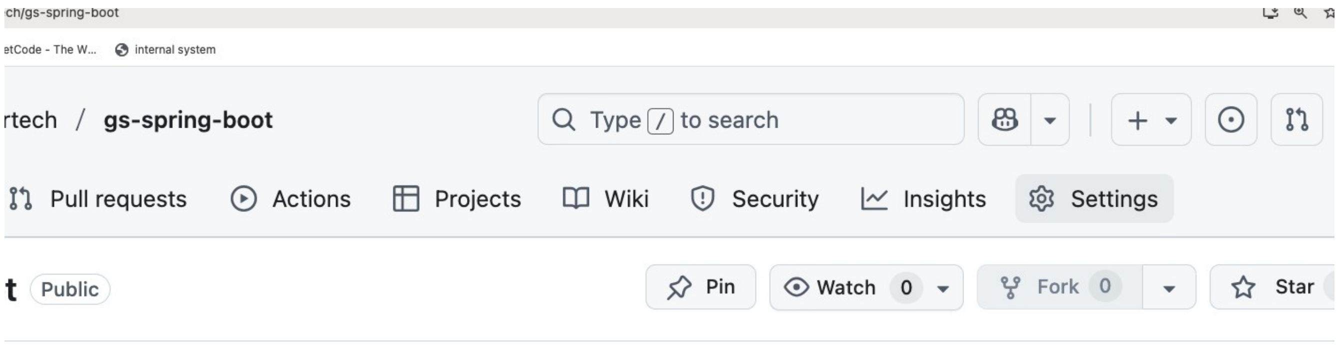
“*ED25519 key fingerprint is*
SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.”

Step 3 change our spring-boot repo from public to private

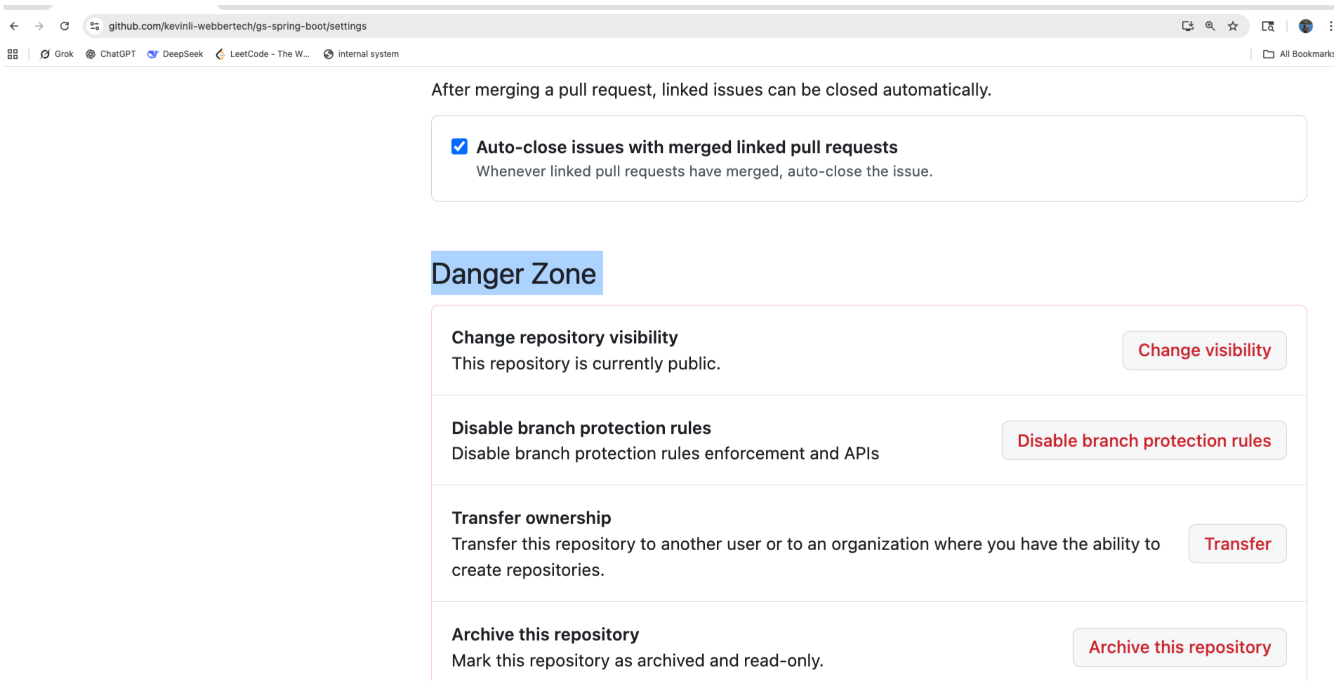
First, we click our public repo we created before shown in the following image,



Next, we click on “Settings”,



We scroll down to the bottom of this page,



github.com/kevinli-webbertech/gs-spring-boot/settings

After merging a pull request, linked issues can be closed automatically.

☑ **Auto-close issues with merged linked pull requests**
Whenever linked pull requests have merged, auto-close the issue.

Danger Zone

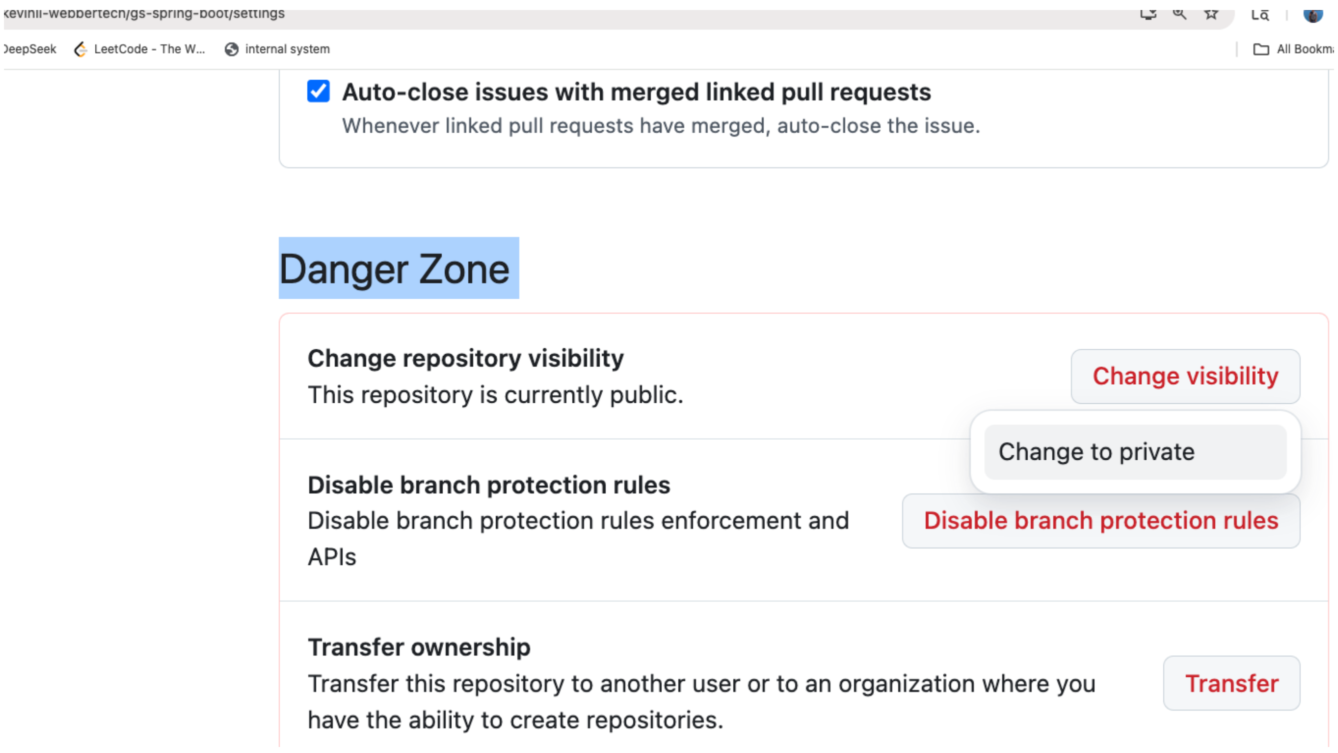
Change repository visibility
This repository is currently public.

Disable branch protection rules
Disable branch protection rules enforcement and APIs

Transfer ownership
Transfer this repository to another user or to an organization where you have the ability to create repositories.

Archive this repository
Mark this repository as archived and read-only.

Now if you click on the “Change visibility”, and you will see the “Change to private”,



☑ **Auto-close issues with merged linked pull requests**
Whenever linked pull requests have merged, auto-close the issue.

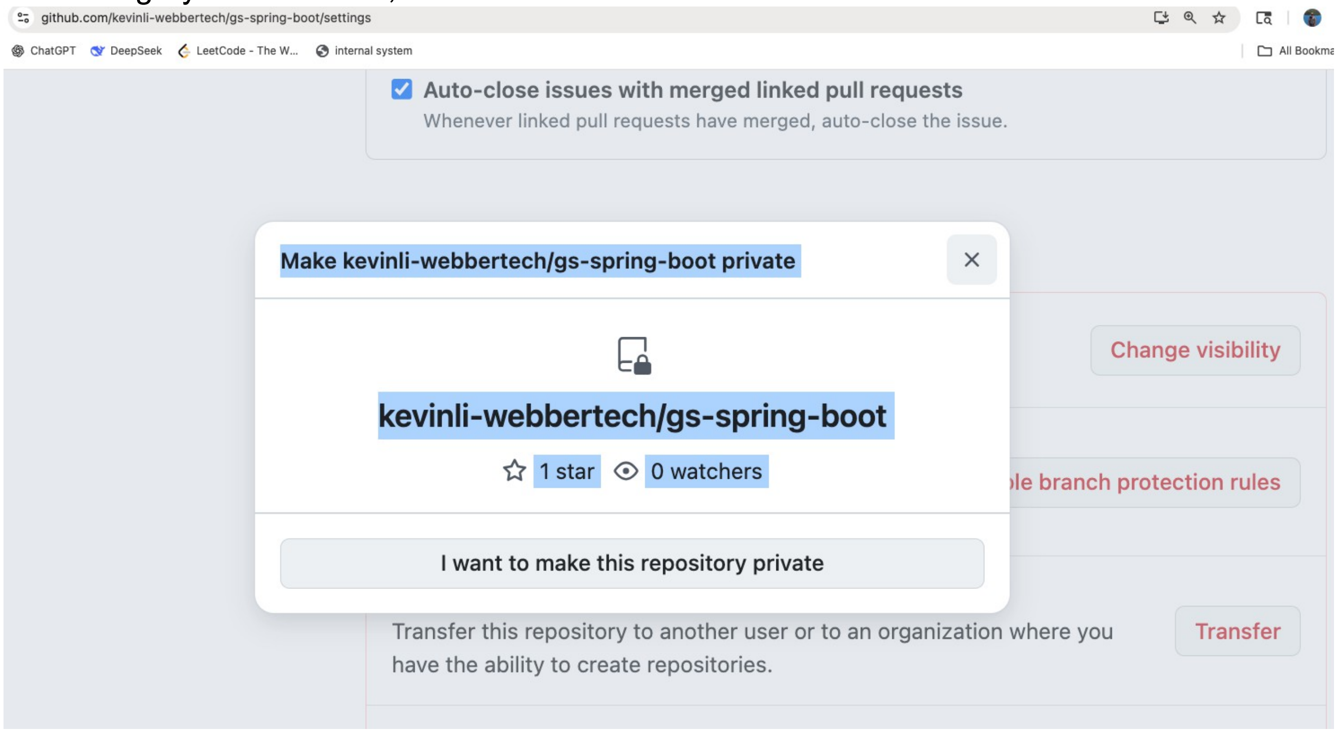
Danger Zone

Change repository visibility
This repository is currently public.

Disable branch protection rules
Disable branch protection rules enforcement and APIs

Transfer ownership
Transfer this repository to another user or to an organization where you have the ability to create repositories.

Click the grey button below,



github.com/kevinli-webbertech/gs-spring-boot/settings

ChatGPT DeepSeek LeetCode - The W... internal system

Auto-close issues with merged linked pull requests

Whenever linked pull requests have merged, auto-close the issue.

Make kevinli-webbertech/gs-spring-boot private

kevinli-webbertech/gs-spring-boot

1 star 0 watchers

I want to make this repository private

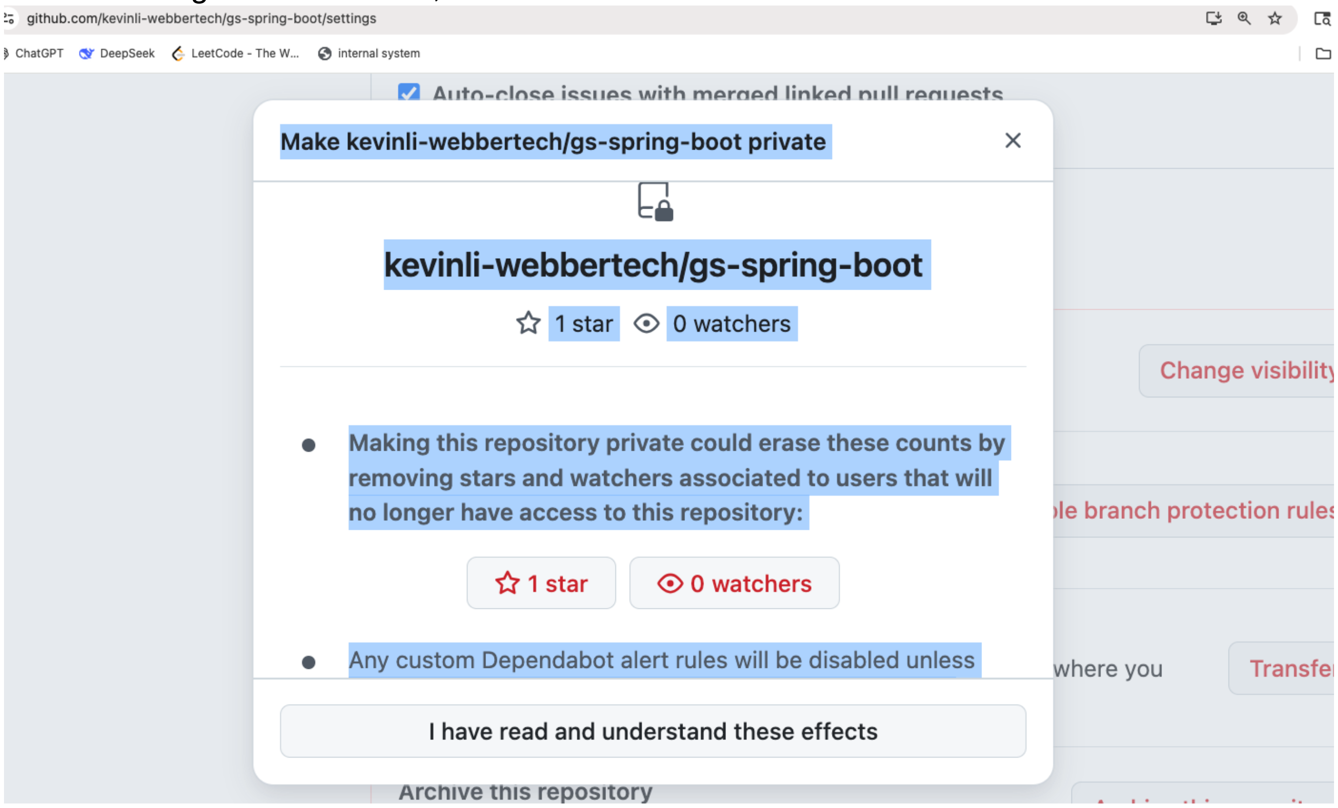
Change visibility

le branch protection rules

Transfer this repository to another user or to an organization where you have the ability to create repositories.

Transfer

There is an agreement to click,



github.com/kevinli-webbertech/gs-spring-boot/settings

ChatGPT DeepSeek LeetCode - The W... internal system

Auto-close issues with merged linked pull requests

Make kevinli-webbertech/gs-spring-boot private

kevinli-webbertech/gs-spring-boot

1 star 0 watchers

- Making this repository private could erase these counts by removing stars and watchers associated to users that will no longer have access to this repository:

1 star 0 watchers

- Any custom Dependabot alert rules will be disabled unless

I have read and understand these effects

Change visibility

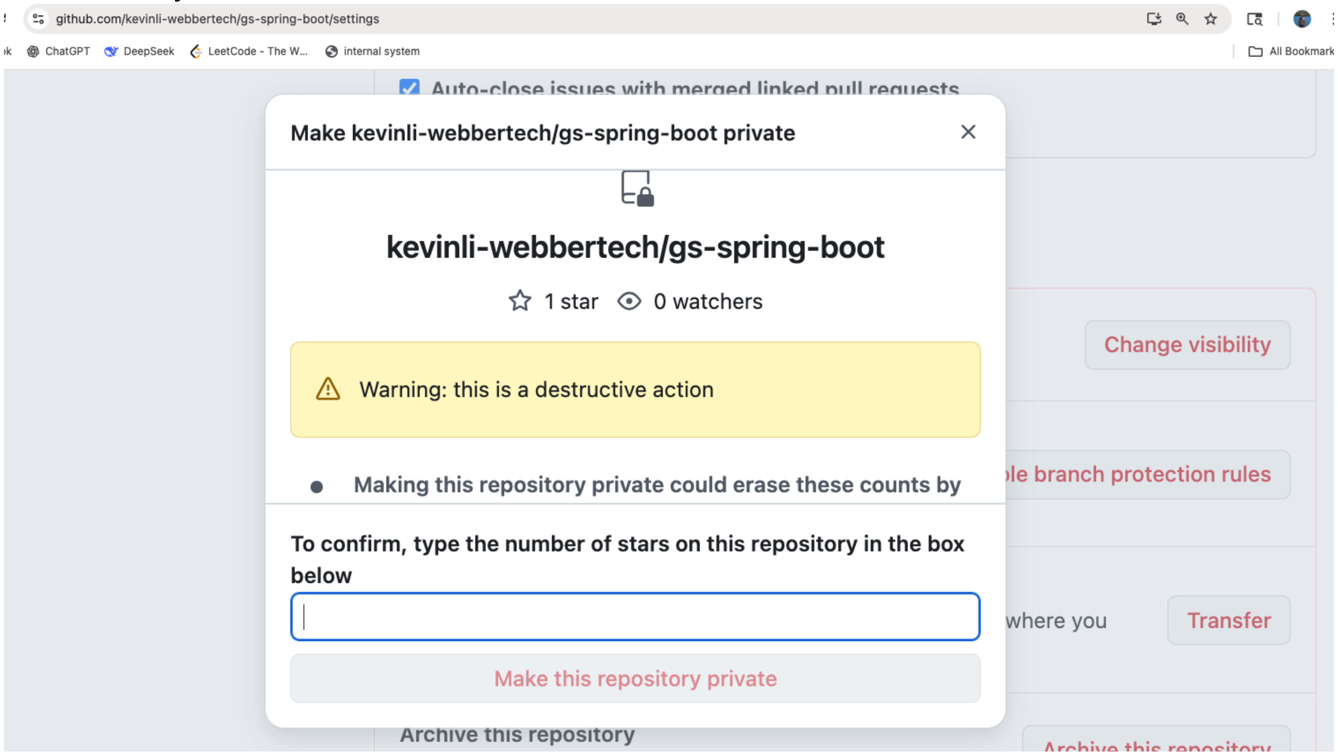
le branch protection rules

where you

Transfe

Archive this repository

This is a confirmation box, because this can be a dangerous operation. Without confirmation, we could accidentally delete a repo or make the scope unavailable to the developers who were actively work on it,



github.com/kevinli-webbertech/gs-spring-boot/settings

Make kevinli-webbertech/gs-spring-boot private

kevinli-webbertech/gs-spring-boot

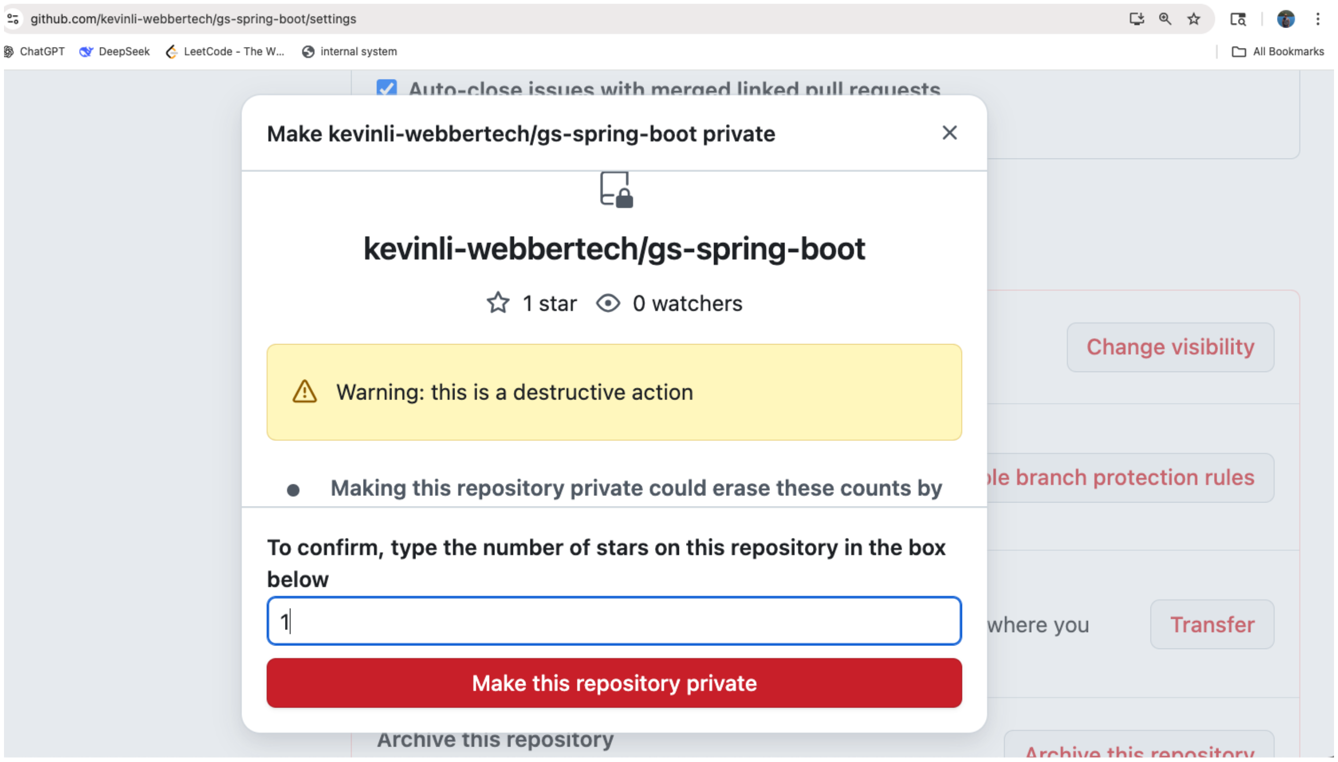
1 star 0 watchers

Warning: this is a destructive action

- Making this repository private could erase these counts by

To confirm, type the number of stars on this repository in the box below

Now the button is not grey anymore, and we can click on it,



github.com/kevinli-webbertech/gs-spring-boot/settings

Make kevinli-webbertech/gs-spring-boot private

kevinli-webbertech/gs-spring-boot

1 star 0 watchers

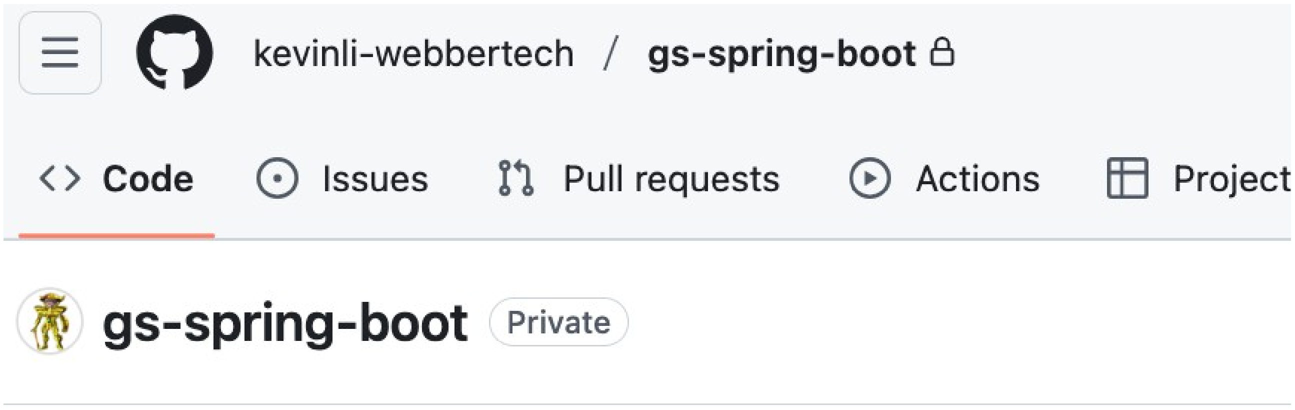
Warning: this is a destructive action

- Making this repository private could erase these counts by

To confirm, type the number of stars on this repository in the box below

1

Now it shows "private".



Now you are good to go. Please use this guide to prepare your container. Make sure once you set it up and you do not delete your container, since the .ssh directory would be there and just in case it changes every time so you don't need to update your ssh public key in the github server all the time.

Ref

https://docs.github.com/en/authentication/connecting-to-github-with-ssh/testing-your-ssh-connection (github official guide – how to test your ssh key after you add it)

https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account (how to add ssh key to the github)

https://medium.com/@mai.vly/setup-jenkins-for-your-application-from-the-private-github-repo-with-docker-bac79335ff27 (how to configure and jenkin job with the ssh)