Install and setup git

Install command: sudo apt install git -y

Setup Git

Set your username: git config --global user.name 'your username here'

Set your email address: git config --global user.email 'your email here'

Setup SSH authentication

Generate an ssh key: ssh-keygen -t ed25519 -C "your\_email@example.com" This creates a new SSH key, using the provided email as a label. When you’re prompted to: "Enter a file in which to save the key," press Enter. This accepts the default file location. At the prompt, to type a secure passphrase, press enter for no passphrase. If you want to use a passphrase, go ahead as it provides an extra layer of security. However, for this course you do not need to add a passphase.

Use the following command to see your public ssh key: cat ~/.ssh/id\_ed25519.pub copy the entire output of this command to the clipboard. See examle here:

In your github page, go to settings > SSH and GPG Keys Here add your ssh key that you copied to the clipboad in the step prior. Type yes (the whole word) and press enter and you are done

The error message indicates that the Git repository you are trying to access has disabled password authentication, and you need to use a different method for authentication.

GitHub, for example, has deprecated password authentication for Git operations since August 13, 2021. Instead, it now recommends using personal access tokens or SSH keys for authentication.

Here are the steps you can take to resolve this issue:

1. Personal Access Token

1. Generate a personal access token on GitHub:

- Go to your GitHub settings.

- Navigate to "Developer settings" > "Personal access tokens."

- Generate a new token with the "repo" scope.

2. Use the generated token for authentication:

bash

git clone https://<username>:<token>@github.com/santhu41/learning.git/

Replace `<username>` with your GitHub username and `<token>` with the generated personal access token.

2. SSH Key

1. If you don't have an SSH key set up, generate one:

bash

ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"

2. Add your SSH key to the SSH agent:

bash

eval "$(ssh-agent -s)"

ssh-add ~/.ssh/id\_rsa

3. Add your SSH key to your GitHub account:

- Copy the SSH key to the clipboard: `pbcopy < ~/.ssh/id\_rsa.pub`

- Go to your GitHub settings.

- Navigate to "SSH and GPG keys" > "New SSH key."

- Paste your key into the "Key" field.

4. Clone the repository using the SSH URL:

bash

git clone git@github.com:santhu41/learning.git

Ensure that you replace `<username>` with your GitHub username in the commands above.

Choose either the personal access token or SSH key method based on your preference. Both methods are secure, but some users may prefer one over the other.

**Error**

root@ceipal-VirtualBox:~/sample# git clone git@github.com:santhu41/learning.git

Cloning into 'learning'...

ssh: connect to host github.com port 22: Connection refused

fatal: Could not read from remote repository.

Please make sure you have the correct access rights

and the repository exists.

The error message "ssh: connect to host github.com port 22: Connection refused" indicates that your system is unable to connect to GitHub using the default SSH port (Port 22). This could be due to a variety of reasons.

Here are some steps you can take to troubleshoot and resolve the issue:

path is where the generated pub key create a file in the path ./ssh/config and add below

1. \*\*Check SSH Configuration:\*\*

Make sure that your SSH configuration is correct. Open your SSH config file (`~/.ssh/config`), or create it if it doesn't exist, and ensure it has the following:

Host github.com

Hostname ssh.github.com

Port 443

This configuration redirects the SSH traffic to port 443, which might help if you are behind a firewall that blocks the default SSH port.

2. \*\*Firewall or Network Issues:\*\*

Ensure that your firewall or network allows outgoing connections on port 443. Some networks or firewalls may block outgoing connections on non-standard ports.

3. \*\*GitHub Status:\*\*

Check the [GitHub Status](https://www.githubstatus.com/) page to see if there are any ongoing issues with GitHub services. Sometimes, service disruptions can cause connectivity issues.

4. \*\*SSH Key Permissions:\*\*

Confirm that your SSH private key (`~/.ssh/id\_rsa`) has the correct permissions. Run the following command to set the appropriate permissions:

bash

chmod 600 ~/.ssh/id\_rsa

5. \*\*Use HTTPS Instead:\*\*

If you encounter persistent issues with SSH, you can switch to HTTPS. Update your remote URL to use HTTPS:

bash

git remote set-url origin https://github.com/santhu41/learning.git

Then try cloning again:

bash

git clone https://github.com/santhu41/learning.git

Note: With HTTPS, you'll need to use your GitHub username and either a personal access token or your GitHub password for authentication.

Try these steps and see if any of them resolve the issue. If the problem persists, there may be a more specific issue with your network or system configuration, and further investigation might be needed.

**Error :-**

root@ceipal-VirtualBox:~/sample# git push -u origin master

Username for 'https://github.com': santhu41

Password for 'https://santhu41@github.com':

remote: Support for password authentication was removed on August 13, 2021.

remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.

fatal: Authentication failed for 'https://github.com/santhu41/learning.git/'

The error you're encountering indicates that you're trying to push to a Git repository using HTTPS with a username and password, but GitHub no longer supports password authentication for Git operations.

As mentioned in the previous responses, GitHub recommends using either a personal access token or SSH keys for authentication. Since you're using HTTPS, let's go with the personal access token method.

Here's how you can update the remote URL to use a personal access token:

1. \*\*Generate a Personal Access Token:\*\*

- Go to your GitHub account settings.

- Navigate to "Developer settings" > "Personal access tokens."

- Generate a new token with the "repo" scope.

2. \*\*Update the Remote URL:\*\*

After generating the token, update the remote URL of your Git repository. Open a terminal and navigate to your local repository:

bash

git remote set-url origin https://<username>:<token>@github.com/santhu41/learning.git

Replace `<username>` with your GitHub username and `<token>` with the personal access token you generated.

For example:

bash

git remote set-url origin https://santhu41:your-token@github.com/santhu41/learning.git

3. \*\*Push Again:\*\*

Now, try pushing your changes:

```bash

git push -u origin master

```

You should not be prompted for a password, and the push should succeed using the personal access token.

Remember to keep your personal access token secure and do not share it. If you ever need to revoke the token's access, you can do so in your GitHub settings.