

Guide to SSH, SCP, and Git

SSH into umnak

For Lab 1, you must ssh into the CSE-managed Linux server `umnak.cs.washington.edu`. This uses the same credentials as `attu.cs.washington.edu`, i.e., your CSE NetID and password.

```
ssh <username>@umnak.cs.washington.edu
```

Optionally, rather than having to enter your password each time you do this, you may wish to set up a public/private key pair on your local machine, and copy your public key to umnak. Once set up correctly, you will no longer be prompted with a password when you ssh. More detailed instructions on how to do this can be found [below](#).

Using SCP to copy files from umnak

SCP (Secure CoPy) is a tool that copies files, and understands how to do so over ssh. Just like ssh, it can use either keys or a password to connect to remote machines.

```
scp <username>@umnak.cs.washington.edu:<path_to_your_file_on_umnak>  
<destination_on_your_local_machine>
```

This will prompt you for your umnak account password, and will then copy the file from umnak to `<destination>` on your local machine. (You can also copy from your local machine to umnak by switching the second and third arguments above.)

If you have passwordless ssh set up as explained below, you can do

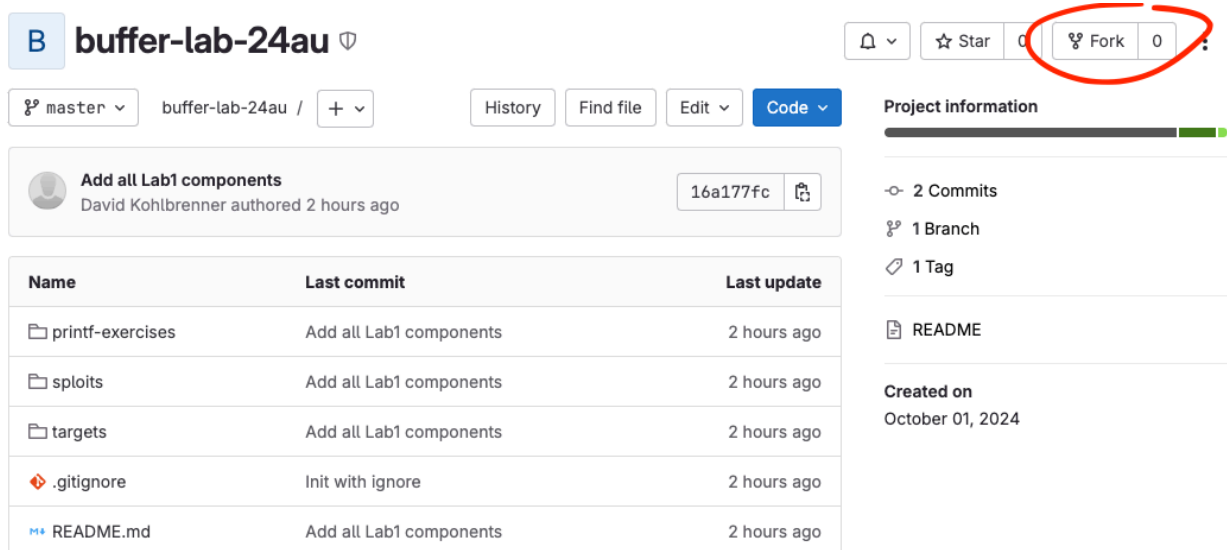
```
scp -i <path_to_private_key>  
<username>@umnak.cs.washington.edu:<path_to_your_file_on_umnak>  
<destination_on_your_local_machine>
```

Git Basics

Step 1: Fork repository

For Lab 1 and the Final Project, you will need to fork a git repository that we provide you. This essentially means that you will create your own copy of the repository, which you can then share with a partner.

From the main repository we share with you, fork the project by clicking the “Fork” button on the top right:






The screenshot shows the GitLab interface for a repository named 'buffer-lab-24au'. At the top right, the 'Fork' button is circled in red. Below the repository name, there are buttons for 'History', 'Find file', 'Edit', and 'Code'. A commit history table is displayed, showing the most recent commit 'Add all Lab1 components' by David Kohlbrenner. The table lists files like 'printf-exercises', 'spoits', 'targets', '.gitignore', and 'README.md'. On the right, the 'Project information' section shows '2 Commits', '1 Branch', '1 Tag', and 'README'. The repository was created on October 01, 2024.

Name	Last commit	Last update
printf-exercises	Add all Lab1 components	2 hours ago
spoits	Add all Lab1 components	2 hours ago
targets	Add all Lab1 components	2 hours ago
.gitignore	Init with ignore	2 hours ago
README.md	Add all Lab1 components	2 hours ago

Give the new repository whatever name you want. **Important!!! Make sure that you make this a private repository.**

Visibility level

-  Private
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.
-  Internal
The project can be accessed by any logged in user.
-  Public
The project can be accessed without any authentication.

Step 2: Share Repository

If you have a partner, you can add them to your new repository via Manage > Members in the gitlab web interface.

Step 3: Clone Repository

Next, you will want to clone your repository to umnak, as follows.

First, from the gitlab web view of your project, click the blue “Code” button and copy the “Clone with HTTPS” string.

Optional alternative, or necessary alternative if you have two-factor authentication on your CSE account: you can also “Clone with SSH”, in which case you must connect umnak and your gitlab account with a key pair. Instructions for how to do that are [below](#).

Second, SSH into umnak, navigate to whatever directory you want to be working in, and then:

```
git clone <string_from_previous_step>
```

Both partners can do this on their respective umnak accounts.

Step 4: Work on Code

When you have made changes that you'd like to save and/or share with your partner, you can commit it to your shared repository. We recommend you do this regularly! You may want to use branches and merge requests and other git features, but that is beyond the scope of this guide.

```
git add <changed files>
git commit -m "a useful message about what you did"
git push
```

To pull in changes that your partner has made:

```
git pull
```

Useful git commands for seeing what is different between the current main repository and your local copy:

```
git status
git diff <filename>
```

Optional: Generating and Using a Key Pair

There are two reasons you may need or want to generate a key pair as part of this process, one on your local machine, and one on umnak.

Step 1: Generate Key Pair

For password-less SSH to umnak: Do the following *on your local machine!*

For using SSH to clone your git repository on umnak: Do the following *on umnak!*

To generate a key-pair run the following in a terminal (it is strongly suggested to use a passphrase):

```
ssh-keygen -t ed25519 -f ~/.ssh/<file_name_for_key>
```

NOTE: If you are doing this for use with Gitlab, we recommend omitting the non-bolded parts above, so you get a default key name.

This will generate *two* files in the ~/.ssh/ directory: <file_name_for_key> and <file_name_for_key>.pub. If you omit the -f argument, it will create files with a default name, something like id_ed25519 and id_ed25519.pub. The .pub is your public key, the other is your private key. **Never share your private key.**

Step 2: Share Public Key

For password-less SSH to umnak: *This assumes you did Step 1 on your local machine!* You will need to copy the contents of your local <file_name_for_key>.pub file onto umnak.

That file should include a single line that starts with “ssh-ed25519”, something like:

```
ssh-ed25519  
AAAAC3NzaC1lZDI1NTE5AAAAIOHK0eNCjWCsVX/otyjCFadkLe89W9Ep4Mqk00R 0ODdp  
user@localhost.
```

SSH into umnak with a password and edit (or create) the file ~/.ssh/authorized_keys to include that line.

To then use ssh without a password:

```
ssh -i <path_to_private_key> <username>@umnak.cs.washington.edu
```

For using SSH to clone your git repository on umnak: *This assumes you did Step 1 on umnak!* Copy the contents of your ~/.ssh/<file_name_for_key>.pub file on umnak. That file should include a single line that starts with “ssh-ed25519”, something like: ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIOHK0eNCjWCsVX/otyjCFadkLe89W9Ep4Mqk00R 0ODdp
user@localhost.

In your browser, go to https://gitlab.cs.washington.edu/-/user_settings/ssh_keys. Click “Add new key”. Paste your public key from umnak there, and “Add key”. **If your key did not have a default name** (like id_ed25519.pub, i.e., you specified a different name in Step 1 above), you need to set up an SSH config file in .ssh/config so that Gitlab knows what file to find your key in. See an example and more details [here](#).

Then you can return to the “git clone” step in the Git instructions [above](#).

A note for Windows:

We’re assuming you are using the official “Windows Terminal” application (<https://apps.microsoft.com/store/detail/windows-terminal/9N0DX20HK701>)
You can also consider installing and using “[windows subsystem for linux](#)”

FAQ

- Do I have to set a passphrase for my ssh key?
 - Nope. Just a good idea in general. This is not the same as a password for logging in to umnak. The ssh key passphrase just protects the ssh private key locally, and is never sent to umnak.
- I want to use VSCode or JetBrains or some other tool to edit code on umnak directly, can I do that?
 - Sure! All those tools allow you to edit files over ssh, you’ll need to look up instructions for your specific tool for how to do that. It may need access to the ssh private key you generated.
 - A note that these tools accomplish this goal by automatically installing a (quite large, >1GB) server on the remote machine you are connecting to. This may cause them to run somewhat slowly.