Land DA System Training

Connecting to an HPC Environment on Mac OS

By: Kristopher Booker, Gillian Petro, Edward Snyder

Introduction to SSH

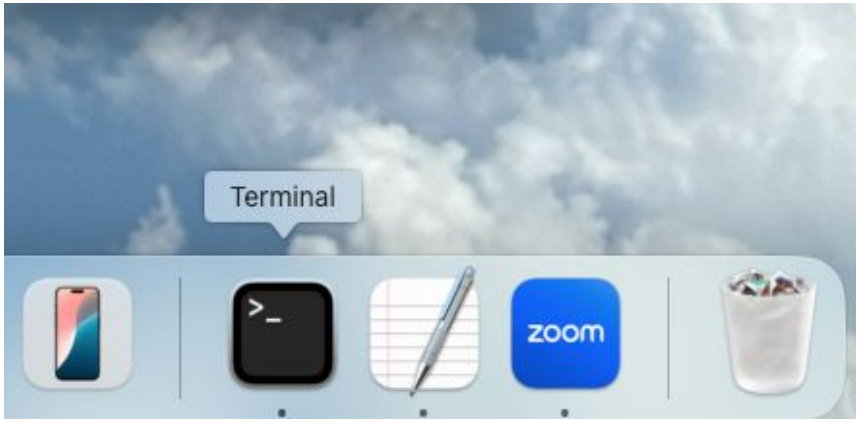
- A **S**ecure **SH**ell (SSH) tunnel creates an encrypted connection between two computer systems. This allows users to:
  - Access and use a remote system via the command line on their local machine.
  - Transfer data securely between two systems.
- Many HPC platforms are accessed via SSH from a user's computer.
  - NOAA RDHPCS
  - Academic HPCs
  - Commercial cloud platforms (e.g., AWS EC2s)

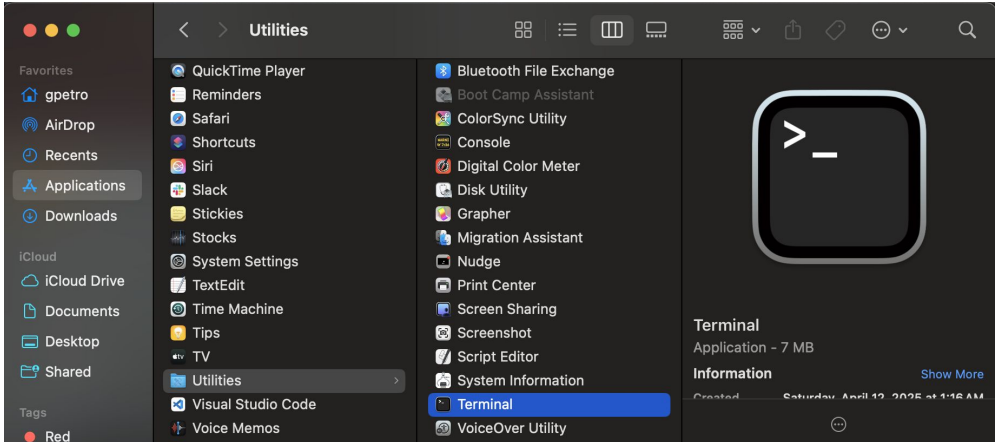
Introduction to SSH

- To use key-based authentication, you first need to generate public/private key pairs for your system. You can use **`ssh-keygen`** to generate key files, and you can specify the following key-generation algorithms:
  - Digital Signature Algorithm (DSA)
  - Rivest–Shamir–Adleman (RSA)
  - Elliptic Curve Digital Signature Algorithm (ECDSA)
  - Ed25519
- To see the available options, run the **`ssh-keygen`** command with the **`-t`** flag
- ed25519 is the default algorithm. A strong algorithm and key length should be used, such as ECDSA.

Open Terminal App

- Open the Mac OS Terminal Application:
  - In the menu bar OR
  - Under Finder → Applications → Utilities → Terminal





Generate public/private key pair

- Issue the following command in your Terminal window:

  ```
  ssh-keygen -t ed25519 -f /Users/<username>/.ssh/id_ed25519_student{1-60}
  ```

  where **`<username>`** is replaced with your actual username, and **`{1-60}`** is replaced with your assigned student number.

- When prompted for a passphrase, press **return/enter** twice and leave blank.

```
gpetro@gpetro-MacBook-Pro ~ % ssh-keygen -t ed25519 -f /Users/gpetro/.ssh/id_ed2
5519_student4
Generating public/private ed25519 key pair.
Enter passphrase for "/Users/gpetro/.ssh/id_ed25519_student4" (empty for no pass
phrase):
Enter same passphrase again:
```

Generate public/private key pair

- This should generate a public/private key pair in your home `.ssh` directory.

```
Your identification has been saved in /Users/gpetro/.ssh/id_ed25519_student4
Your public key has been saved in /Users/gpetro/.ssh/id_ed25519_student4.pub
The key fingerprint is:
SHA256:gxRGEXCo1ZDXQSPsVsg2FymeIMCt/z8L+s3dxY3znCI gpetro@gpetro-MacBook-Pro
The key's randomart image is:
+--[ED25519 256]--+
|o.. oXO*=+       |
| ...=oXo=.       |
|  .+ *.*         |
| ..  .=.         |
|  .  .. S        |
|   .     .  . o  |
|    ..       = . |
|    ..+.. .E..+ .|
|   ....=o. .. .+ |
+----[SHA256]-----+
gpetro@gpetro-MacBook-Pro ~ %
```

Generate public/private key pair

- Use a text editor of your choice to view the public key file (e.g., vim).
- For example:

```
vim /Users/<username>/.ssh/id_ed25519_student(n).pub
```

(when using vim, press `:q` to quit the editor)

- Copy-paste the public key contents to the workshop administrator via the Slack workspace channel **#cadre-publickeys** and inform them of your student number (i.e., student 5).

Generate public/private key pair

- NOTE: Two (2) keys are generated: a public and a private key. **DO NOT SEND THE PRIVATE KEY!** A **public key** will end in `.pub` and will look like this:

```
gpetro — vim ~/.ssh/id_ed25519_student4.pub — 80×24
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIAsbBmWYZJYoKMbZ3oeaQNdyg8qLc9RRTgQmbQu0ozhw
 gpetro@gpetro-MacBook-Pro
```

- And a **private key** will look like this:

```
-----BEGIN OPENSSH PRIVATE KEY-----
AAAAAAAAABAAAA
11111111==
-----END OPENSSH PRIVATE KEY-----
```

- Workshop administrators will add the public key to the authorization file on the bastion host, which will allow you to log in.

Connecting to an HPC Environment

- Add the newly generated **private** key to your laptop's identity:

  ```
  ssh-add /User/<username>/.ssh/id_ed25519_student(n)
  ```

  where **`<username>`** is replaced with your actual username, and **`(n)`** is replaced by your assigned student number.

- If successful you should see a message similar to:

  ```
  Identity added: /Users/<username>/.ssh/id_ed25519_student5
  (username@MacBook-Pro.local)
  ```

Connecting to an HPC Environment

- Users may access the HPC environment by issuing the command below in the terminal (again replacing **(n)** with their assigned student number):

```
ssh student(n)@jump.epic.noaa.gov
```

OR

```
ssh student(n)@137.75.93.46
```

- **NOTE:** This will only work during the training when the HPC system is active for the training!

Connecting to an HPC Environment

- The user may see a message such as:

```
The authenticity of host '137.75.93.46 (137.75.93.46)'
can't be established.
ED25519 key fingerprint is
SHA256:/QXZI9NOLMLEAPUMCgNSTQ7m36T+U9PueVLhQWvqBRI.
This key is not known by any other names.
Are you sure you want to continue connecting
(yes/no/[fingerprint])?
```

- Verify that you are connecting to the correct system and enter **`yes`** to continue.

Connecting to an HPC Environment

- This should automatically redirect users through the bastion proxy to the controller node of their HPC environment.
- If you run the **`ls`** command, you will see the Land DA container (**`.img`**) file, the **`inputs`** data directory, and a **`rocoto`** directory:

```
ubuntu@ip-10-29-82-122:~$ ls
Land-DA_v2.1_inputs.tar.gz  rocoto
inputs                      ubuntu22.04-intel-landda-daconsortium.img
```