10718/94889
Tech Session 1

Agenda

1. Introduction to course infrastructure
2. CMU VPN
3. SSH access to course server
4. Access to course database
5. GUI access to course database (via SSH tunneling)
6. Access to course github

Quiz

Fill out this quiz as you complete steps, so that we can verify that you’re up to speed:

https://forms.gle/1PQYrEQjgFfDLxpw9

Intro to course infrastructure

Why remote dev?

- Confidentiality
- Computing power
- Collaboration

On confidentiality

- Do not try to download data
- Inform course instructors if you accidentally download data
- Get comfortable using the database!

Our stack

Course server

- AWS server shared by entire class
- Connect via ssh & CMU Full VPN
- Scale to meet demand throughout semester

Course Database

- AWS database shared by entire class
- Connect from within course server: psql command line tool, or local GUI tool tunneled through server
- Also scaled throughout semester

CMU VPN

CMU VPN

1. Download Cisco Anyconnect VPN client from here and install
2. Open Anyconnect client
3. Enter login credentials:
    - Connect to: vpn.cmu.edu
    - Group: “Full VPN”
    - Username: your andrewid
    - Password: your CMU password
4. Click connect

SSH to server

- Public key authentication (asymmetric cryptography) is used to verify your identity to the Amazon EC2 servers.
- Key pair is majority times generated via OpenSSH through the command

```
ssh-keygen -t rsa
```

  Remember to take a note of your passphrase.

- Public key of all users are added to an ‘authorized_keys_file’ on the server end.
- Your **private key** is your secret! **NEVER** *share*.
- On connecting the server you send over a session specific token which contains a digital signature encrypted with your private key. Only the corresponding public key can decrypt the token. On success the server then allows the incoming login request to be validated.

Platform based SSH nuances

- **Windows** - Terminal can be accessed via popular SSH clients like Putty, MobaXTerm, Powershell.
- MobaXTerm (Benefit - Allows you to open multiple terminal tabs, and view directory structure Windows style):
  1. On the session panel right click ‘User sessions’ and click on ‘New Session’.
  2. Click on the SSH tab on top left corner.
  3. Enter Remote Host: {andrewid}@mlpolicylab.dssg.io
  4. Click on ‘Advanced SSH Settings’.
  5. Make sure X11-Forwarding is enabled.
  6. Select checkbox ‘Use private key’ and provide the location the contains the private key file (some like ~/.ssh/id_rsa). Finally, click OK.
  7. The session will now appear on the left. Click it and provide your passphrase at the terminal

Windows (Contd.)

Putty Setup:

- Putty requires the private key format to be changed from the OpenSSH one to PuttyKeyGen that would have the .ppk extention
- Open PuttyGen and click on Conversion => Import Key
- Search for id_rsa key on your computer
- Click on “Save Private Key”.
- Choose a location and a name for the new .ppk key.
- Now go to putty and add path to key for the connection.

Putty Private Key Configuration:

- Enter remote server Host Name under Session
- Navigate to Connection > SSH > Auth
- Click Browse.. Under Authentication parameters/Private key file for authentication.
- Locate the {id_rsa_from_puttgyen}.ppk private key and click Open.
- Finally, click Open again to log in to the remote server with key pair authentication.

SSH for Linux/Mac

Quite straightforward:

-
  ```
  ssh {andrew-id}@mlpolicylab.dssg.io -i path/to/your/private/key/file
  ```
- Adding the key file to ssh-agent
  - Start the ssh-agent by running - eval `ssh-agent`
  - Add the following instructions to ~/.ssh/config

    ```
    Host *
     UseKeychain yes
     AddKeysToAgent yes
     IdentityFile ~/path/to your/private/key/file
    ```
  - Add your private key to the ssh-agent.

    ```
    $ ssh-add -K ~/path/to your/private/key/file
    ```
  - Verify by checking the output of `$ ssh-add -l`

Connect to course database (w/ psql)

.pgpass file

Your .pgpass file holds your database access credentials in the following format:

```
host:port:database:username:password
```

Connect to course database w/ local gui client

Connecting via DBeaver (1)

1. Install dbeaver (https://dbeaver.com/edition/community/)
2. Make sure you're connected to the VPN
3. Navigate to Database > New Database Connection
4. Select PostgreSQL, click next
5. In the Main panel, enter the database credentials from your .pgpass file.
    - For the database name, use `group_students_database`
    - remember: `run` `cat ~/.pgpass` `on class server` to print .pgpass file contents to the command line

Connecting via DBeaver (2)

6. Enter your SSH credentials:
   a. Confirm “Use SSH Tunnel”
   b. In host/ip, enter the course server hostname (mlpolicylab.dssg.io)
   c. In username, enter your username for the course server (your andrew id)
7. Set up private key authentication:
   a. Set Authentication Method to Public Key
   b. In the Private Key field, enter the full path to your private key file. Ex:
      - `C:\Users\Aaron\.ssh\id_rsa`
      - /home/aaron/.ssh/id_rsa
   c. Enter your SSH passphrase in the passphrase field
8. Click the Test Connection button to make sure everything is working
9. Paste the output for `SELECT * FROM test_gui_connect` into question 3 in your google form

DBeaver & SSHJ

If you’re having trouble with SSH tunneling: try installing SSHJ

1. Under help, choose “Install new software”
2. Search for SSHJ and install the software
3. Restart DBeaver and try setting up SSH tunneling.
4. Ask us if you’re still getting an error!

Accessing the course Github

1. Open this page: https://github.com/dssg/test-mlpolicylab-private
2. If you have access, paste the title line from README.md into the google form