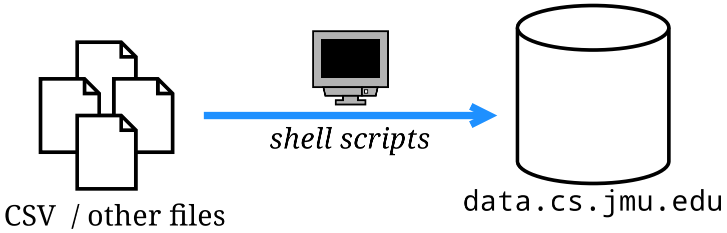
How to Run the Scripts for GP2

Dr. Chris Mayfield

Department of Computer Science
James Madison University

Feb 22, 2022



Part 1: SSH

SSH config

Are you tired of typing long ssh commands?

For example:

```
ssh -L 5432:data.cs.jmu.edu:5432 username@student.cs.jmu.edu
```

- Create a `~/.ssh/config` file with the following:

```
Host stu
Hostname student.cs.jmu.edu
User username  # CHANGE TO YOUR E-ID
LocalForward 5432 data.cs.jmu.edu:5432
```

- Now you can just type: `ssh stu`

SSH known_hosts

The authenticity of host ... can't be established.

- Add/edit these lines in your `~/.ssh/known_hosts` file:
  https://w3.cs.jmu.edu/mayfiecs/cs374/notes/known_hosts

SSH keys

Are you tired of typing your password?

1. Run `ssh-keygen` on your machine (one-time setup)
   - Pick a good passphrase to protect your identity (in case someone steals your laptop / private key)
2. Add your public key to `.ssh/authorized_keys`
   - On macOS, run:
     ```
     ssh-copy-id stu
     ```
   - On Windows, run:
     ```
     type id_rsa.pub | ssh stu "cat >> .ssh/authorized_keys"
     ```
3. Add your public key to your GitHub account
   - https://github.com/settings/keys

Part 2: GitHub

Check out a working copy

If you haven't done so already:

```
git clone git@github.com:cs374/group.git
```

(replace "group" with your group name)

This step is a **one-time setup**

You may clone as many copies as you like!

- For example, at school and at home
- GitHub will merge changes for you

Passwords in CS 374

Reminder

JMU e-ID $\neq$ Database password $\neq$ GitHub password

Which password do I use?

`ssh student.cs.jmu.edu`	JMU e-ID
`psql -h data.cs.jmu.edu`	Database
Password field of pgAdmin	Database
`git clone git@github...`	GitHub

Part 3: Scripts

Example: HW4

Original instructions:

-
  ```
  psql -q -h localhost -U mayfiecs postgres
        < hw4.sql 2>&1 | tee hw4.txt
  ```

Environment variables:

- `export PGHOST=localhost`
- `export PGUSER=mayfiecs`
- `export PGPASSWORD=123456789`
- `export PGDATABASE=postgres`
- `psql -q < hw4.sql 2>&1 | tee hw4.txt`

On Windows, use `set` instead of `export`:
https://phoenixnap.com/kb/windows-set-environment-variable

Running scripts

If working remotely and my group name is *absent*:

- `psql -h localhost absent < create.sql`
- `./copy.sh`
- `psql -h localhost absent < stats.sql`

Using environment variables *works until you exit the terminal*

- `export PGHOST=localhost`
- `export PGDATABASE=absent`
- `psql < create.sql`
- `./copy.sh`
- `psql < stats.sql`

COPY and \copy

- `COPY` – copy data between a file and a table from database server's point of view
- `\copy` – copy data between a file and a table from the psql client's point of view

https://www.postgresql.org/docs/11/sql-copy.html

- "`\copy` invokes `COPY FROM STDIN` or `COPY TO STDOUT`, and then fetches/stores the data in a file accessible to the psql client. Thus, file accessibility and access rights depend on the client rather than the server when `\copy` is used."

About copy.sh

Make it executable first:

- `chmod 755 copy.sh`
- `ls -l copy.sh`

Why is copy.sh so expensive?

-
  ```
  psql -c 'COPY (...) TO STDOUT;' vdoe | \
  psql -c 'COPY foo FROM STDIN;' absent
  ```

If slow, run copy.sh on stu

- `ssh student.cs.jmu.edu`
- `cd` *directory_of_copy.sh*