

Setup Troubleshooting

This page contains known setup issues. If you run into issues not on this page or still have issues after following the instructions on this page, feel free to post on Ed!

Hive Setup Issues

Connection Issues

- Try connecting to a different Hive machine. Before switching to a different machine, be sure to run `~cs161/proj1/stop` on your original machine.
- If you're on campus and on the Berkeley-Visitor Wi-Fi network, then switch to eduroam or connect to the Berkeley VPN.
- Try turning your WiFi on and off, and reconnecting your computer.
- If you're able to connect but not open the Project 1 VM for whatever reason, try running the stop command (`ssh hive3 ~cs161/proj1/stop`). Then, SSH into the machine and run the start command again.
- If your ssh setup script is not working, you can still access the project by using the command `ssh cs161-XXX@hiveY.cs.berkeley.edu`, replacing `xxx` and `Y`, and using your instructional account password.
- If all else fails, switching to the local setup may be easier.

Stopping on a Broken Hive Machine

If you were working on a certain hive machine that's no longer working (e.g. the machine went down, or you were timed out of the machine for connecting too many times), then you could try running this command:

```
ssh -J hive2 hive3
```

which should let you temporarily bypass the timeout by routing your traffic through a different hive machine than the one you can't connect to. Replace `hive2` with a hive that's NOT the one you're currently using (e.g. a "good" Hive machine, and `hive3` with the one that you want to shut the project down on. If this command works, then you'll be able to run the stop command to kill the VM

on the original server that you were working on, or if you want to pick up where you left off, you may use the start or ssh commands as well.

```
~cs161/proj1/stop
~cs161/proj1/start
~cs161/proj1/ssh
```

Local Setup Issues

Host Identification Changed

If you see an error like this:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@  WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!  @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!

Someone could be eavesdropping on you right now (man-in-the-middle attack)!

It is also possible that a host key has just been changed.

The fingerprint for the ED25519 key sent by the remote host is
SHA256:IwINo1vgPF+zQvr9/68Xv9M1tscu5k6BADA4Cqg3rc0.
Please contact your system administrator.

Add correct host key in /Users/evanbot/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /Users/evanbot/.ssh/known_hosts:66
Host key for [127.0.0.1]:16122 has changed and you have requested strict checking.
Host key verification failed.
```

It means that you connected to the local VM once when it had a certain set of keys, but on your latest attempt to connect, those keys have changed. This is probably because you downloaded multiple versions of the VM, each one with different keys.

Look for filename and line number in the “offending key” section of the error message. In the example above, the filename is `/Users/evanbot/.ssh/known_hosts` and the line number is 66.

Open this file in a text editor (e.g. Vim), and delete the line number indicated. Then run the command again. If it warns you about an unknown host, type “yes”.

QEMU Issues

`-accel hvf: Unknown Error` on macOS

If you get an error if the form `qemu-system-x86_64: -accel hvf: Unknown Error` and QEMU immediately aborts and exits, this indicates that you have an issue with your Homebrew or QEMU installation. To confirm, run `file $(which qemu-system-x86_64)`. If the output of this command indicates that QEMU is an `x86_64` executable, then you have an issue with your Homebrew installation.

This usually happens because you migrated all the data from an Intel (x86-64) Mac to an Apple silicon (ARM) Mac. Homebrew functions differently on x86-64 and ARM devices, so copying over the Homebrew installation data from your old device causes the wrong version of QEMU to be installed—namely, the x86-64 version of QEMU rather than the ARM version of QEMU that can run natively on your Apple silicon Mac.

To fix this, you will need to reinstall your Homebrew installation. Follow the following instructions:

- 1 Ensure that you are not running on an x86-64 terminal or shell. If you are using the native macOS Terminal and default shells zsh or Bash, this should not be an issue (though it would be good to verify).

Run the `arch` command. If the output is `i386`, then you need to reinstall your custom terminal or custom shell. If the output is `arm64`, then you may proceed to the next step.

- 2 Export your current Homebrew installed packages state by running

```
$ brew bundle dump
```

This will create a `Brewfile` in your current directory.

- 3 Uninstall Homebrew by running the following command:

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/uninstall.sh)" /b
```

- 4 Reinstall Homebrew by running the following command:

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Make sure you follow the instructions for adding Homebrew to your PATH. The installation prompts will provide you with information on how to do this specific to your shell.

- 5 Reinstall your exported packages by running the following command:

```
$ brew bundle --file Brewfile
```
