Contents

Getting Started

For Fall 2020, most of your work in 6.004 will happen on the Athena machines. However, there are a few steps that you will need to do to setup your account: adding the 6.004 toolchain, and setting up an SSH key for MIT GitHub.

1 Login to Athena

To access athena, you have a few options. You can setup tools for your computer, access Athena via a web browser, or visit an Athena workstation.

1.1 Connect locally

To access Athena on your PC/Mac, you can use SSH. However, if you have a Windows or Mac computer, there are a few things you need to do first.

1.1.1 Windows

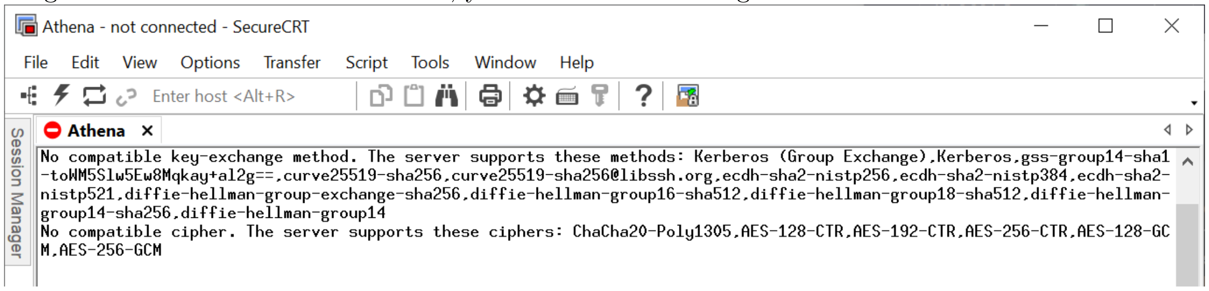
To access Athena on Windows, you can use SecureCRT that MIT students are licensed to use non-commercially.

SecureCRT

1. **Download and install SecureCRT.**
2. **Setup SecureCRT:** On the same page, follow the SecureCRT for Windows: Quick Start Guide instructions (certificates required).
3. **Access Athena** Open SecureCRT and click on **Athena** After providing your password and authenticating with duo, you will be on Athena.

Fixing SecureCRT connection errors

Sometimes, after installing and configuring SecureCRT, you still cannot connect to Athena. In this case, you might see the error below. To fix this, you can do the following:


Athena - not connected - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
Athena
Session Manager

```
No compatible key-exchange method. The server supports these methods: Kerberos (Group Exchange),Kerberos,gss-group14-sha1
-toWM5Slw5Ew8Mqkay+al2g==,curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-
nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-
group14-sha256,diffie-hellman-group14
No compatible cipher. The server supports these ciphers: ChaCha20-Poly1305,AES-128-CTR,AES-192-CTR,AES-256-CTR,AES-128-GC
M,AES-256-GCM
```

1. Right click the Athena session and click **Properties**.
2. Under **SSH2**, in the **Authentication** section, check **Keyboard Interactive** (for entering your password and Duo authentication), and in the Key exchange section, check **curve25519-sha256**.
3. Under **SSH2** / **Advanced**, check the **AES-256-CTR** cipher.

1.2 MacOS

MacOS comes with a terminal app and SSH installed:

1. **Open a terminal**. You can open any terminal application. For example, **Terminal.app** is pre-installed.
   At the terminal prompt, type: `ssh {kerberos}@athena.dialup.mit.edu` and replace `{kerberos}` with your Athena username. After authenticating with your password and duo, you will be on Athena.

1.3 Linux

For most Linux systems, no additional setup should be needed. You just need to open a terminal window, and type: `ssh {kerberos}@athena.dialup.mit.edu` and replace `{kerberos}` with your Athena username. After authenticating with your password and duo, you will be on Athena.

1.4 Via a web browser

Visit `https://athena.dialup.mit.edu`, where you can use a web-based SSH client to access the dialup servers. While this method requires no configuration, the web-based SSH client may not have some advanced terminal features.

1.5 From a workstation

At any unoccupied workstation that displays the words "Welcome to Athena", enter your username (kerberos) and password.

2 Setup on Athena

2.1 Setting up the 6.004 Toolchain

6.004 has an AFS locker containing binaries that it needs for you to test and compile your labs, however you need to add this manually.

1. **Access your Athena account.**
2. **Add Course Locker.** Run the following commands in your terminal. This will configure your current session to use the 6.004 toolchains required for labs.

   ```
   source /mit/6.004/setup.sh
   ```

3. **Add Course Locker Automatically.** To add the course locker automatically whenever you log in, you should add the line "`source /mit/6.004/setup.sh`" (without the quotes) to your `.bashrc`. You can do this through any editor (`emacs`, `nano`, `vim`, ...), or by running the following command:

   ```
   echo "source /mit/6.004/setup.sh" >> /mit/{kerberos}/.bashrc
   ```

   Note that the change to `.bashrc` is applied after you restart the current Athena session.

   If you prefer to not modify your `.bashrc`, you should run "`source /mit/6.004/setup.sh`" manually whenever you login.

2.2 Generating an SSH Key

To submit your code for grading, you will be pushing it to MIT's GitHub. However, to access GitHub, you will need to create an SSH key and add it to your account. (An SSH key is a piece of data that can be used to authenticate to many services on the command line; we will use it in 6.004 to authenticate to MIT GitHub so you can pull and submit code.)

On your Athena account (and locally, if you want to push to GitHub), you would run the command `ssh-keygen`. It will ask you where to store the file, and a password. For a good default, paste the following, replacing `{kerberos}` with your Athena username.

```
ssh-keygen -t rsa -b 4096 -C "{kerberos}@mit.edu"
```

- When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location (`.ssh/id_rsa`). Athena automatically loads this key name to the ssh-agent.
- When you're prompted to "Enter passphrase," you can create a passphrase to secure your SSH key. You will be prompted to type it a second time as a double-check, and will need to enter it again every time in the future you want to use this SSH key. Note that nothing will appear on the screen as you enter your passphrase; this is normal and is how many command-line applications accept passphrases. Also note that entering a passphrase is optional.

2.3 Link your MIT GitHub account

Once you have created your SSH key, you will need to link it with GitHub.

1. Output the content of the public key on your terminal using

   ```
   cat ~/.ssh/id_rsa.pub
   ```

   Then copy the output to your clipboard (manually).

2. Login to MIT GitHub or `https://github.mit.edu` with your MIT Kerberos and Password.

3. Go to **Settings** (click on your icon in the top right) → **SSH and GPG keys**.
4. Click **New SSH key**.
5. Paste the content on the Key field.
6. Click **Add SSH Key**.

3 Some basic git commands

In this class, Git is used to track all of your code submissions and automatically send your code to our graders every time you update it. You will only need to use a few simple git commands. After following the Create Repository link for the lab you are working on, a repo will be created for you with the initial contents of the lab. The website will also show you a `git clone` command that you need to run in your terminal to make a local copy of your lab repo. This command automatically configures your local repo to print a feedback message from Didit (our lab grading system) on your Terminal when you make a submission.

Alternatively, you can clone a local repo without the feedback message feature by running the `git clone` command in your terminal:

```
git clone git@github.mit.edu:6004-fa20/lab1-{YourMITUsername}.git
```

replacing lab1 with whichever lab you are working on. While this configuration does not print a Didit feedback message on your terminal on new submission, Didit will grade your new submissions. Refer to the end of this section for accessing Didit results.

The next instruction is `git commit`. This will take a set of files and package them up and create a point that you can come back to. Once any file tracked by Git is changed, you can commit all your changes by using the command:

```
git commit -am "Message here: commits all changes on files that are tracked"
```

This will take ALL the files you have edited and package them up together.

If you would like to control which files are included in the commit, you can use `git add` together with a `commit -m "Message"` (without -a option, to commit only the files that you added explicitly. For example,

```
git add File1 File2 File3
git commit -m "Message here: commits only changes in File1 File2 File3"
```

will add and commit only files `File1`, `File2`, and `File3`.

The last git command you will need to learn is `git push` which takes your code and submits it to github.mit.edu where we collect your labs.

```
git push
```

You may submit your code as many times as you like. We urge you to submit often so that your work is backed up in Git, and so that the course staff can see your latest version of your code when trying to answer your questions on piazza.

You can see the status of your latest lab submission by going to the course Didit website, clicking on your repo name, followed by clicking on your build. Each build shows you exactly which tests have passed and which have failed.

4 Editting files

The repos that you clone from github will have code templates that you will need to fill in in order to complete the lab. In order to modify these templates, you will need to use an editor. There are multiple editors available on athena including: vi, vim, emacs, and nano. If you are new to using text editors, then you will probably want to use nano which is a basic text editor which lists commonly used commands at the bottom of the editor. If you would like to use vi/m or emacs you may want to do a search for their most commonly used commands on the web.

You may also find our short bash reference guide helpful in manipulating your files on athena.

As you get into the later labs (lab 3 on), you will be writing minispec code. There are configuration files available to help properly highlight these files in your editors. See the course resources page (Minispec syntax setup section) for details.

5 Setup for local development (optional)

While most of the tools used for 6.004 are only available on Athena, it is possible to write code locally using your favorite text editor and then sync your files to Athena.

5.1 Enable syntax highlighting

If you want to enable syntax highlighting, you can acquire the files at the course resources pages (Minispec syntax setup) section.

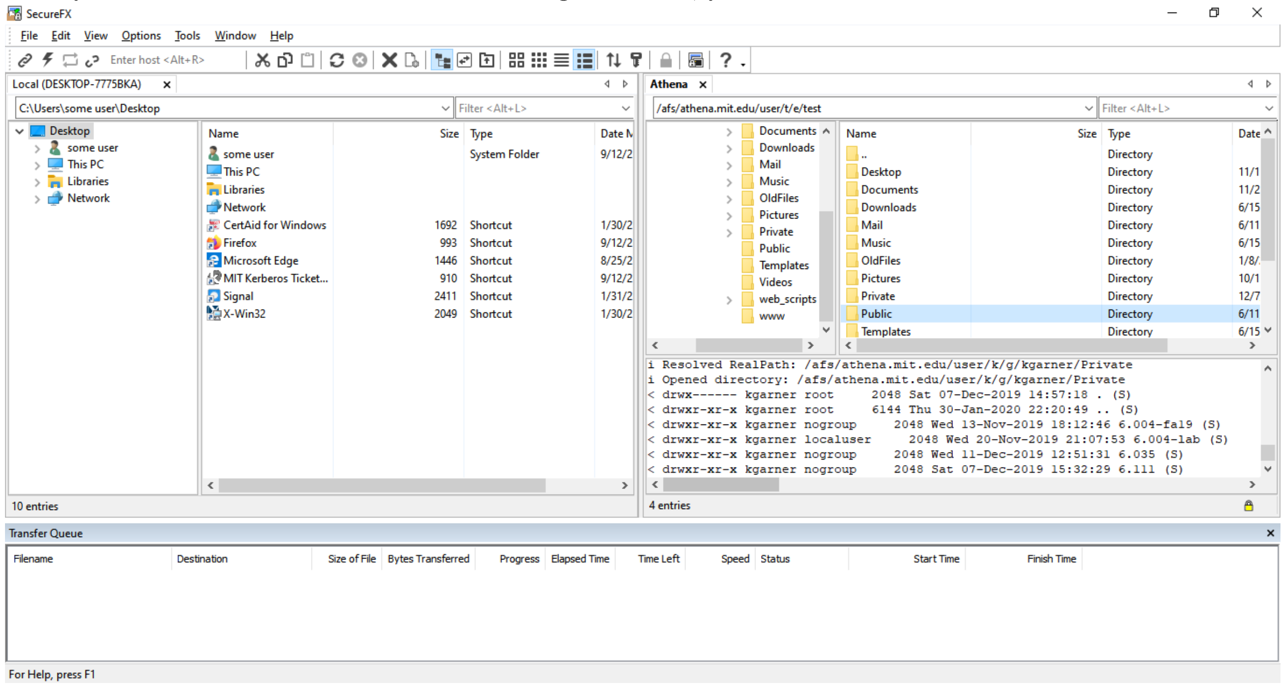
5.2 Syncing files with Athena

To sync files to Athena from your computer, you have a couple of options: graphically and through the command line.

5.2.1 Graphically (Windows)

One of the easiest tools you can use to sync files between Athena and your Windows computer is SecureFX. This comes installed with **SecureCRT**. You can read more about installation details in Section 1.1.1.

Once you have installed and authenticated using SecureFX, you should see a screen similar to below:

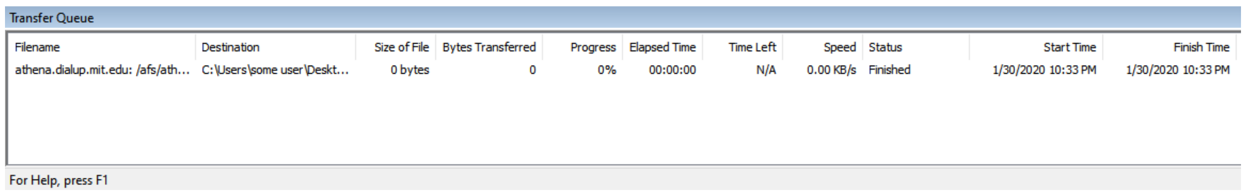


Name	Size	Type	Date M
some user		System Folder	9/12/2
This PC			
Libraries			
Network			
CertAid for Windows	1692	Shortcut	1/30/2
Firefox	993	Shortcut	9/12/2
Microsoft Edge	1446	Shortcut	8/25/2
MIT Kerberos Ticket...	910	Shortcut	9/12/2
Signal	2411	Shortcut	1/31/2
X-Win32	2049	Shortcut	1/30/2

Name	Size	Type	Date
..		Directory	
Desktop		Directory	11/1
Documents		Directory	11/2
Downloads		Directory	6/15
Mail		Directory	6/11
Music		Directory	6/15
OldFiles		Directory	1/8/
Pictures		Directory	10/1
Private		Directory	12/7
Public		Directory	6/11
Templates		Directory	6/15

```
i Resolved RealPath: /afs/athena.mit.edu/user/k/g/kgarner/Private
i Opened directory: /afs/athena.mit.edu/user/k/g/kgarner/Private
< drwx------ kgarner root     2048 Sat 07-Dec-2019 14:57:18 . (S)
< drwxr-xr-x kgarner root     6144 Thu 30-Jan-2020 22:20:49 .. (S)
< drwxr-xr-x kgarner nogroup     2048 Wed 13-Nov-2019 18:12:46 6.004-fa19 (S)
< drwxr-xr-x kgarner localuser     2048 Wed 20-Nov-2019 21:07:53 6.004-lab (S)
< drwxr-xr-x kgarner nogroup     2048 Wed 11-Dec-2019 12:51:31 6.035 (S)
< drwxr-xr-x kgarner nogroup     2048 Sat 07-Dec-2019 15:32:29 6.111 (S)
```

On the left, you will see the directories that are on your computer, and on the right you will see files and directories that are on Athena. To copy files from your computer to Athena or vice-versa, you can just drag files from one window to another. When copying files, you should see information about the progress in the Transfer Queue.



Transfer Queue

Filename	Destination	Size of File	Bytes Transferred	Progress	Elapsed Time	Time Left	Speed	Status	Start Time	Finish Time
athena.dialup.mit.edu: /afs/ath...	C:\Users\some user\Deskt...	0 bytes	0	0%	00:00:00	N/A	0.00 KB/s	Finished	1/30/2020 10:33 PM	1/30/2020 10:33 PM

For Help, press F1

5.2.2 Graphically (MacOS)

On MacOS, you can install Fetch to sync files between Athena and your computer:

1. **Download Fetch** (credentials required).
2. **Install Fetch**. Follow the Installation instructions (certificates required).
3. **Start Fetch**. You can follow Using Fetch at MIT for examples.

5.3 Command line

If you would prefer to use the command line, there are a few options: **SFTP** and **scp**

5.3.1 Using SFTP

SFTP (SSH File Transfer Protocol) is a tool for sharing files via SSH. It can be used in a similar fashion as SSH. The nice thing about SFTP is that it presreves the session (so long as you are connected).

1. **Open SFTP**. Where this is found depends on the operating system you run
   - **Windows:** Open the command prompt. This can be found by opening the start menu and searching for **command prompt**.
   - **MacOS:** Open the **Terminal** app. This can be found in **Applications**
   - **Linux:** open any Terminal application
2. Run the following command: `sftp {kerberos}@athena.dialup.mit.edu` and replace `{kerberos}` with your Athena username. You will be asked to provide a password and login with Duo.
3. When you see **sftp>**, you are in the SFTP shell and can run SFTP commands (described below).

SFTP Commands

- ? Get a description of all commands
- `ls {path}` lists files and directories on Athena. You can provide a file path to list the files in that directory
- `lls {path}` lists files and directories on your computer. You can provide a file path to list the files in that directory
- `get -r {path}` download a file from Athena to your computer. For directories, you should provide **-r** to download all files in a directory.
  Sample command: **get -r test.txt**
- `put -r {path}` upload a file to Athena from your computer. For directories, you should provide **-r** to download all files in a directory.
  Sample command: **put -r test.txt**
- `cd {path}` Change the current directory on Athena. For example, you can run **cd Desktop**
- `lcd {path}` Change the current directory on your computer. For example, you can run **lcd Desktop**

5.3.2 SCP

If you prefer to instead to copy files periodically, you can also use the **scp** command (native to Windows and Linux). The general syntax is as follows:

```
scp {local path} {kerberos}@athena.dialup.mit.edu:{remote path}
```

For example, to copy the file **temp.txt** in my current directory to **temp.txt** on my home directory in athena, run

```
scp temp.txt kerberos@athena.dialup.mit.edu:temp.txt
```

6 Tools for Unreliable Internet Connections (optional)

An unreliable internet connection can make it difficult to maintain an SSH connection to Athena. In this section, we will install a tool called Mosh (mobile shell) which will keep the SSH connection alive even if the network connection fails temporarily. Please follow the instructions in each section below to install, configure, and start a Mosh connection.

6.1 Installing Mosh

To use Mosh, it first needs to be installed on your local machine. Follow the instructions below that correspond to your operating system.

6.1.1 Windows 10

Sadly, there is no Window-specific mosh available yet (The Google chrome version does not support logging into our Athena server). Right now, our best solutions are to enable Linux subsystem for Window and install mosh on Linux.

You can enable Window subsystem for Linux by following these steps from here

1. Open PowerShell as Administrator and run this exact command:
   ```
   Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
   ```
2. Restart your computer when prompted.
3. Go to your Micorsoft Store and Download Ubuntu 18.04 LTS
4. Run Ubuntu 18.04 LTS app from Start Menu and Follow the Linux installation guideline from subsubsection 6.1.3

6.1.2 MacOS

If you have a package manager installed on your Mac such as Homebrew or MacPorts, you can simply use that to install Mosh.

- Homebrew: `brew install mosh`
- MacPorts: `sudo port install mosh`

Alternatively, if you do not have a package manager, you can download and install the .pkg file from the mosh installation website.

6.1.3 Linux

Use the package manager for your distribution of Linux. For example, if you are on Debian or Ubuntu, run

```
sudo apt-get install mosh
```

For a more extensive list of distribution options, check out the mosh installation website.

6.2 Configuration

Mosh needs to be enabled on Athena before you can connect to it. To do so, you will need to add the `mosh_project` locker. First SSH into Athena as usual:

```
ssh {kerberos}@athena.dialup.mit.edu
```

Then run the following command to update your .bashrc file:

```
echo "add mosh_project" >> /mit/{kerberos}/.bashrc
```

After doing so, you can close the SSH connection.

6.3 Staring a Mosh Session

The instructions in the previous two sections only need to be completed once. This section describes the steps you will take each time you want to start a Mosh session.

To use Mosh, simply run the following command on your local machine:

```
mosh {kerberos}@athena.dialup.mit.edu
```

You will be prompted to log in with your Kerberos password and 2-factor authentication. Afterwards, this session should remain persistent even if the network connection drops.

6.4 Limitations and Issues

There are a couple issues to be aware of when working with Mosh:

- If you use Mosh and leave an SSH connection open for a long time (over 10 hours), your Kerberos tickets will expire, and you will be unable to access any of your files. For example, `ls` will be unable to display information about the files, and you will not be able to write text to your files. To fix this, run the command `renew` to renew your Kerberos tickets. You will be prompted for your Kerberos password, but you will not need 2-factor authentication.