What We’re Going To Do

• Why Unix?
• Cloud Computing
• Connecting to AWS
• Introduction to Unix Commands
Etiquette

• PowerPoint interspersed with Challenges
• Ask me questions
• Ask demonstrators
• Work together
• Cheat!
### Unix/Linux Command Reference

#### File Commands
- `ls` - directory listing
- `ls -al` - formatted listing with hidden files
- `cd` - change directory to **dir**
- `cd -` - change to home
- `pwd` - show current directory
- `mkdir dir` - create a directory **dir**
- `rm file` - delete file
- `rm -r dir` - delete directory **dir**
- `rm -f file` - force remove file
- `cp file1 file2` - copy file1 to file2
- `mv file1 file2` - rename or move file1 to file2
- `ln -s file1 file2` - create symbolic link file1 to file2
- `touch file` - create or update file
- `find file` - places standard input into **file**
- `more file` - output the contents of **file**
- `head file` - output the first 10 lines of **file**
- `tail file` - output the last 10 lines of **file**
- `tail -f file` - output the contents of **file** as it grows, starting with the last 10 lines

#### Process Management
- `ps` - display your currently active processes
- `top` - display all running processes
- `kill pid` - kill process ID **pid**
- `killall proc` - kill all processes named **proc**
- `bg` - lists stopped or background jobs; resume a stopped job in the background
- `fg` - brings the most recent job to foreground
- `fg n` - brings job **n** to the foreground

#### File Permissions
- `chmod` - change the permissions of **file**
- `chmod octal file` - change the permissions of **file**
- `chmod -R pattern dir` - recursively set permissions for all files and directories in **dir**
- `chmod` - change the permissions of **file**
- `chmod -R pattern dir` - search for **pattern** in the output of **command**
- `locate file` - find all instances of **file**

#### System Info
- `date` - show the current date and time
- `cal` - show this month's calendar
- `uptime` - show current uptime
- `w` - display who is online
- `whoami` - who you are logged in as
- `finger user` - display information about **user**
- `uname -a` - show kernel information
- `cat /proc/cpuinfo` - cpu information
- `cat /proc/meminfo` - memory information
- `man command` - show the manual for **command**
- `df` - show disk usage
- `du` - show directory space usage
- `free` - show memory and swap usage
- `whereis app` - show possible locations of **app**
- `which app` - show which **app** will be run by default

#### Compression
- `tar cf file.tar files` - create a tar named **file.tar** containing **files**
- `tar xf file.tar` - extract the files from **file.tar**
- `tar czf file.tar.gz files` - create a tar with Gzip compression
- `tar xzf file.tar.gz` - extract a tar using Gzip
- `tar cjf file.tar.bz2` - create a tar with Bzip2 compression
- `tar xjf file.tar.bz2` - extract a tar using Bzip2
- `gzip file` - compresses **file** and renames it to **file.gz**
- `gzip -d file.gz` - decompresses **file.gz** back to **file**

#### Network
- `ping host` - ping **host** and output results
- `whois domain` - get whois information for **domain**
- `dig domain` - get DNS information for **domain**
- `dig -x host` - reverse lookup **host**
- `wget file` - download **file**
- `wget -c file` - continue a stopped download

#### Installation
- Install from source:
  - `./configure`
  - `make`
  - `make install`
- `dpkg -i pkg.deb` - install a package (Debian)
- `rpm -Uhv pkg.rpm` - install a package (RPM)

#### Shortcuts
- `Ctrl+C` - halts the current command
- `Ctrl+Z` - stops the current command, resume with `fg` in the foreground or `bg` in the background
- `Ctrl+D` - log out of current session, similar to `exit`
- `Ctrl+W` - erases one word in the current line
- `Ctrl+U` - erases the whole line
- `Ctrl+R` - type to bring up a recent command
- `!` - repeats the last command
- `exit` - log out of current session

*use with extreme caution.*
What is Unix?

• Operating System
Why Unix?

• Bioinformatics software designed to run on Unix platforms.
• Large amounts of data.
• Much faster than your Windows PC.
How Can We Use Unix?

• Linux computers or servers.
• Compute clusters.
• The cloud.
  – What we’re going to use this week
So What is Cloud Computing?

Gmail by Google

Dropbox

iTunes
Cloud Computing Solutions

Amazon Web Services

Google Compute Engine

Microsoft Azure
How it Works

Own copy of the AMI = Instance (Virtual Machine or VM)
Terminology

• Creating an instance – *buying a brand new computer with software already installed.*

• Starting an instance – *turning that computer on.*

• Stopping an instance – *turning that computer off.*

• Terminating an instance – *setting that computer on fire and throwing it out of the window.*
Connecting to Your Instance

Remote Desktop
Software
e.g. X2Go

Secure Shell – “SSH”
e.g. SSH or PuTTY
You’re now connected to your instance and you’re ready to learn some Unix!
Any Questions So Far?
The Terminal Window

The Command Line, The Shell, The Prompt

Where you see this “$” followed by text, I want you to type the text on your command line
### FastCGI Script

- **script-fastcgi.sh** - script for running FastCGI (locally installed as latest version)

### Variables

- **V-w**
- **V-c**
- **V-p**
- **V-cw**
- **V-cw**
- **V-un**
- **V-nc**
- **V-nc**
- **V-nun**
- **V-nun**
- **V-ncw**
- **V-ncw**
- **V-cw**
- **V-cw**
- **V-cw**

### Scripts

- **fastcgi.sh**
- **manage-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**

### Settings

- **readme-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**

### Environment

- **readme-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**

### Dependencies

- **readme-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**

### Notes

- **readme-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**
- **readme-fastcgi.sh**

### Examples

- **$ examples/SRC/\*.py**
- **$ examples/SRC/\*.py**
- **$ examples/SRC/\*.py**
- **$ examples/SRC/\*.py**
- **$ examples/SRC/\*.py**
Location is Important

First Task – Where am I?

```
$ pwd
/home/genomics
```

This is your “present working directory”
```
genomics@harvard_ami:~$ pwd
/home/genomics
genomics@harvard_ami:~$
```
This location is also known as your Home Directory

Tilde is shorthand for Home:

~
Now let’s create some directories and files

Make a directory

```
$ mkdir Data
```

Change into this directory

```
$ cd Data
```

Now what is your present working directory?

**NOTE!** Directory names (and file names for the matter) can not contain spaces. Underscores are often used instead if you want to separate words.
Now let’s create some directories and files

Make an empty file

$ touch rags

And another two

$ touch Earth Heaven

Now let’s list the contents of the current directory (Data)

$ ls

genomics@harvard ami:~/Data$ touch rags
genomics@harvard ami:~/Data$ touch Earth Heaven
genomics@harvard ami:~/Data$ ls
Earth  Heaven  rags
genomics@harvard ami:~/Data$
Now list ALL of the files

```bash
$ ls -a
```

genomics@harvardami:~/Data$ ls -a
. .. Earth Heaven rags

```bash
genomics@harvardami:~/Data$
```
Now list ALL of the files

```bash
$ ls -a
```

```
. .. Earth Heaven rags
```
These special files are in every directory

.. Points to one directory above
. Points to the current directory
These special files are in every directory:
- .. Points to one directory above
- . Points to the current directory