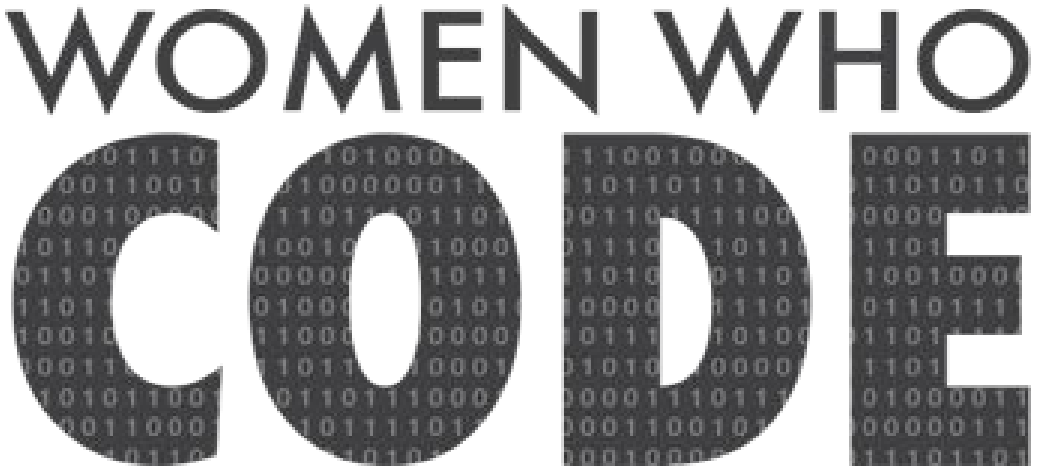
DevOps Study Night



@wwcodeportland

@jivesoftware

DEVOPS COMMAND LINE

Who are these people?

Tricia Radcliffe is DevOps Lead for Women Who Code Portland. She is a Sales Engineer at AppDynamics. She comes from a sysadmin/operations background, with experience as a developer. In her free time, Tricia is also a budding farmer, she's raising an army of hens and knows how to run a tight coop!

Cynthia Kwok is Co-lead for DevOps and Algorithms study nights for Women Who Code Portland. She has a BS in Computer Science from University of California, Riverside. Cynthia has recently moved to Portland, and is currently dabbling with automation and DevOps in her spare time.

How do I get to the Command Line?

- If you are on a mac, you can open ‘terminal’ or download ‘iterm’
  - https://www.iterm2.com/
- On Windows you can open a command line or powershell to access the command line.
- Many code editors have interactive terminals

Windows User

Our talk today will be using linux commands so we will provide a linux shell.
In order to access it, please download and install PUTTY

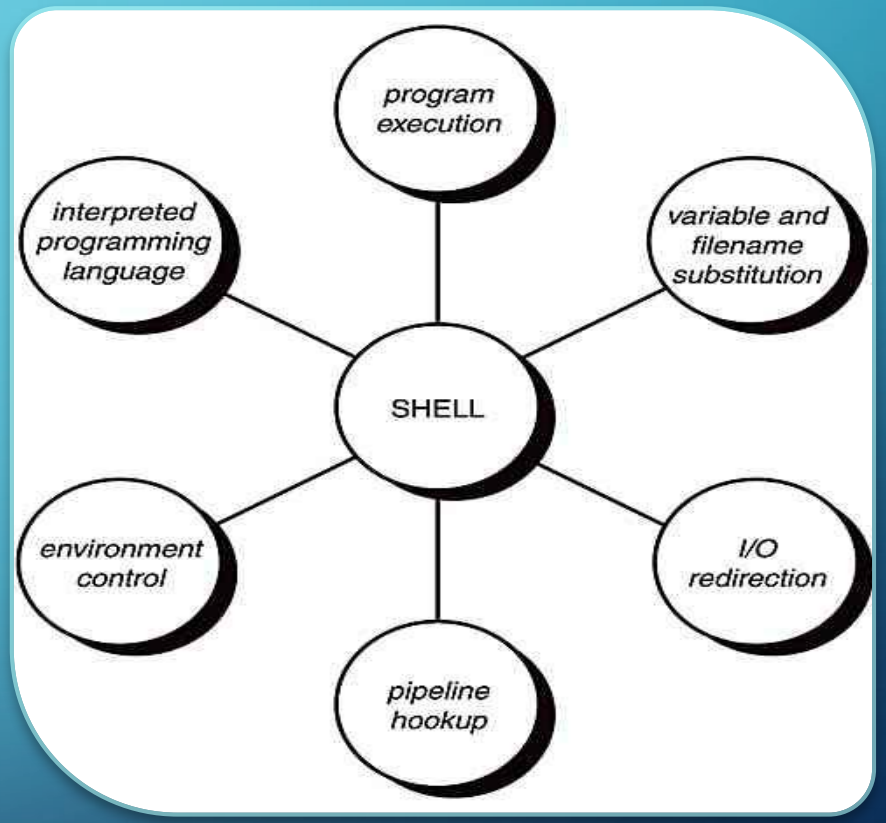
http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

Once installed, create a ssh session

host: 64.94.138.242 user: staff*N* password: abc123

What is a shell?

A **Unix shell** is a command-line interpreter or **shell** that provides a traditional **Unix**-like command line user interface. Users direct the operation of the computer by entering commands as text for a command line interpreter to execute, or by creating text scripts of one or more such commands.



There are MANY shells

sh

- The Bourne shell, called "sh," is one of the original shells, developed for Unix computers by Stephen Bourne at AT&T's Bell Labs in 1977. Its long history of use means many software developers are familiar with it. It offers features such as input and output redirection, shell scripting with string and integer variables, and condition testing and looping.

bash

- The popularity of sh motivated programmers to develop a shell that was compatible with it, but with several enhancements. Linux systems still offer the sh shell, but "bash" -- the "Bourne-again Shell," based on sh -- has become the new default standard. One attractive feature of bash is its ability to run sh shell scripts unchanged. Shell scripts are complex sets of commands that automate programming and maintenance chores; being able to reuse these scripts saves programmers time. Conveniences not present with the original Bourne shell include command completion and a command history.

csh and tcsh

- Developers have written large parts of the Linux operating system in the C and C++ languages. Using C syntax as a model, Bill Joy at Berkeley University developed the "C-shell," csh, in 1978. Ken Greer, working at Carnegie-Mellon University, took csh concepts a step forward with a new shell, tcsh, which Linux systems now offer. Tcsh fixed problems in csh and added command completion, in which the shell makes educated "guesses" as you type, based on your system's directory structure and files. Tcsh does not run bash scripts, as the two have substantial differences.

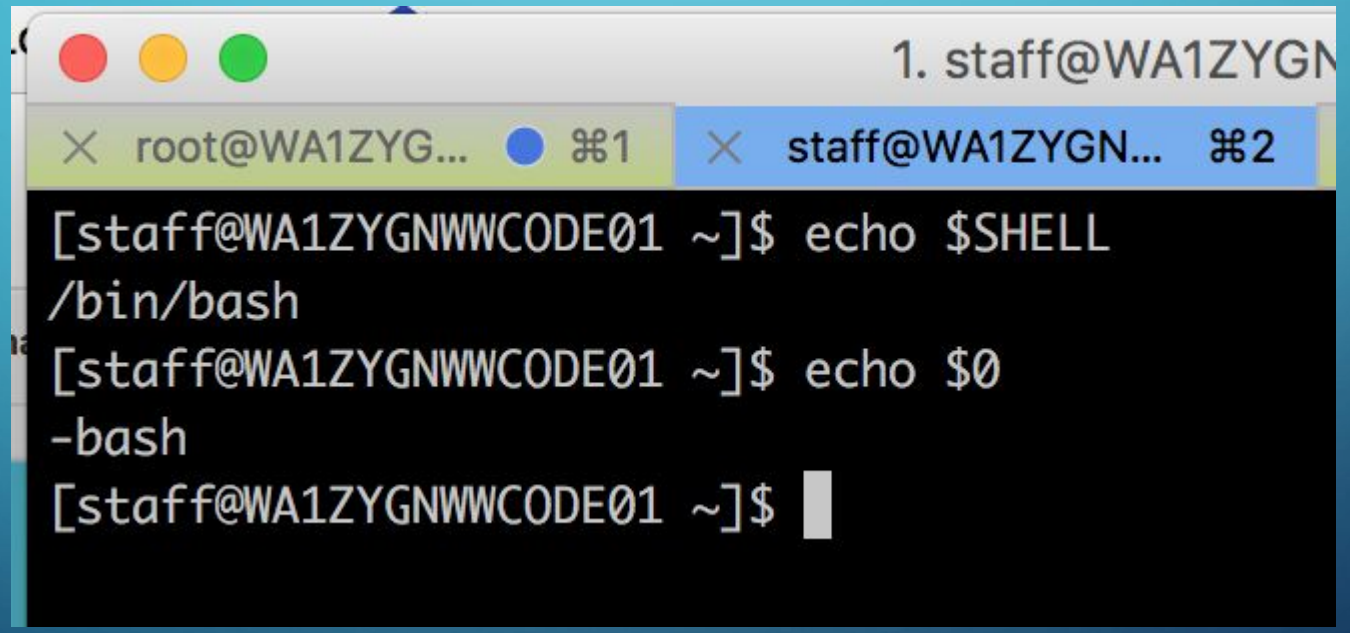
ksh

- David Korn developed the Korn shell, or ksh, about the time tcsh was introduced. Ksh is compatible with sh and bash. Ksh improves on the Bourne shell by adding floating-point arithmetic, job control, command aliasing and command completion. AT&T held proprietary rights to ksh until 2000, when it became open source.

HANDS ON: What shell am I in?

$ echo $0

$ echo $shell

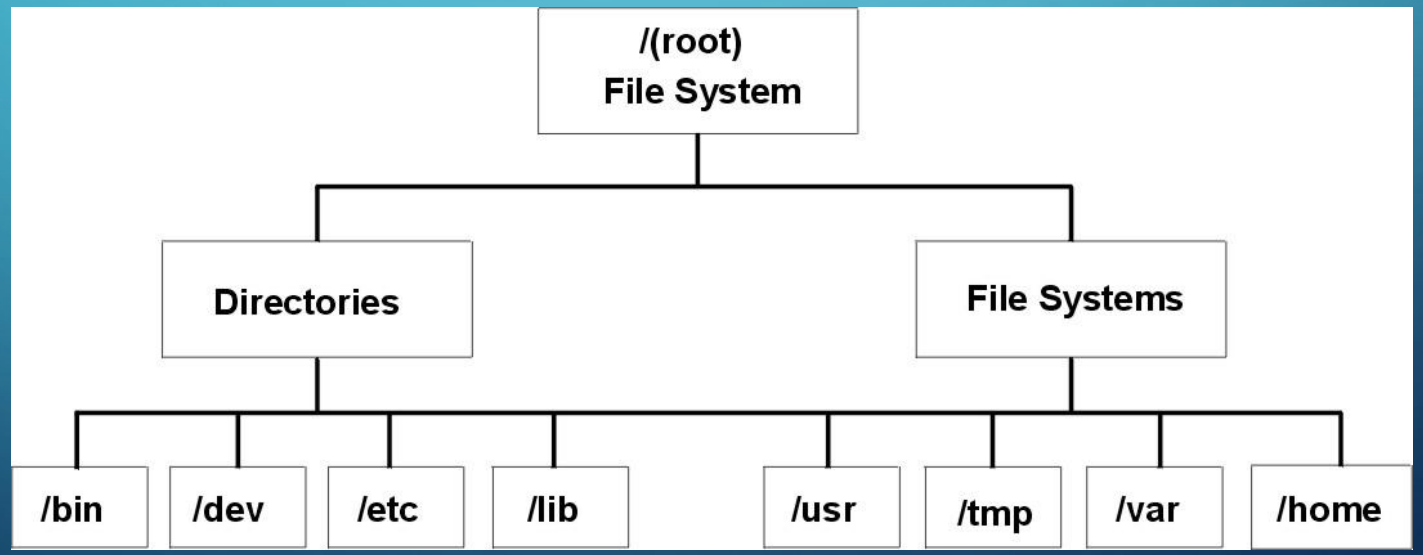


File System Standards

Is the method or structure by which files are stored on Disk, which varies per distribution

Important:

- Knowing how to navigate the structure
- Knowing what each folder's designation/purpose

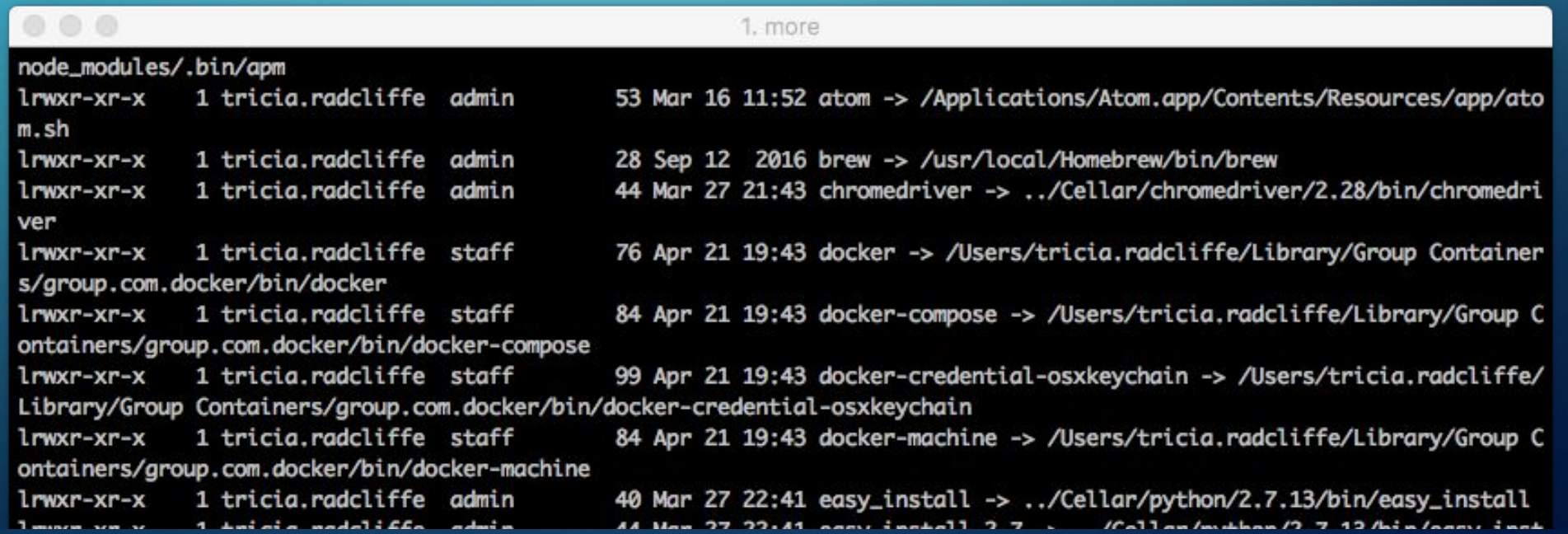


Symbolic Links

The **ln** command is a standard **Unix** command utility used to create a hard link or a symbolic link (symlink) to an existing file. The use of a hard link allows multiple filenames to be associated with the same file since a hard link points to the inode of a given file, the data of which is stored on disk.

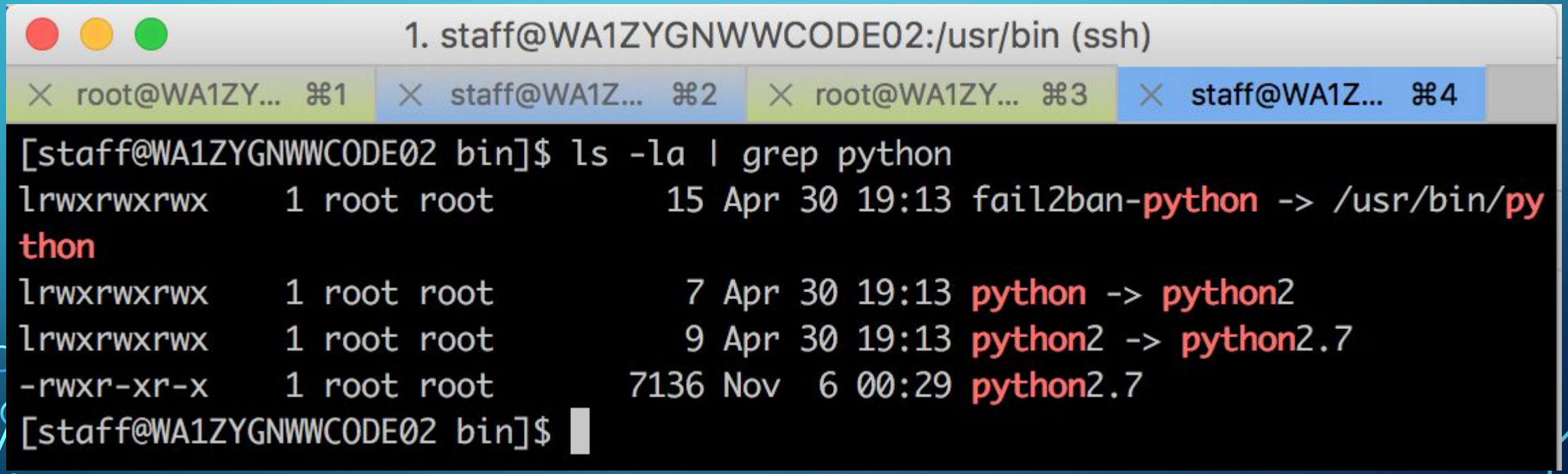
Most often you will run into symbolic links

$ ln –s <originlocation\file> <linklocation\file>



HANDS ON: Looking at directories & linked files

$ ls -la /usr/bin | grep python

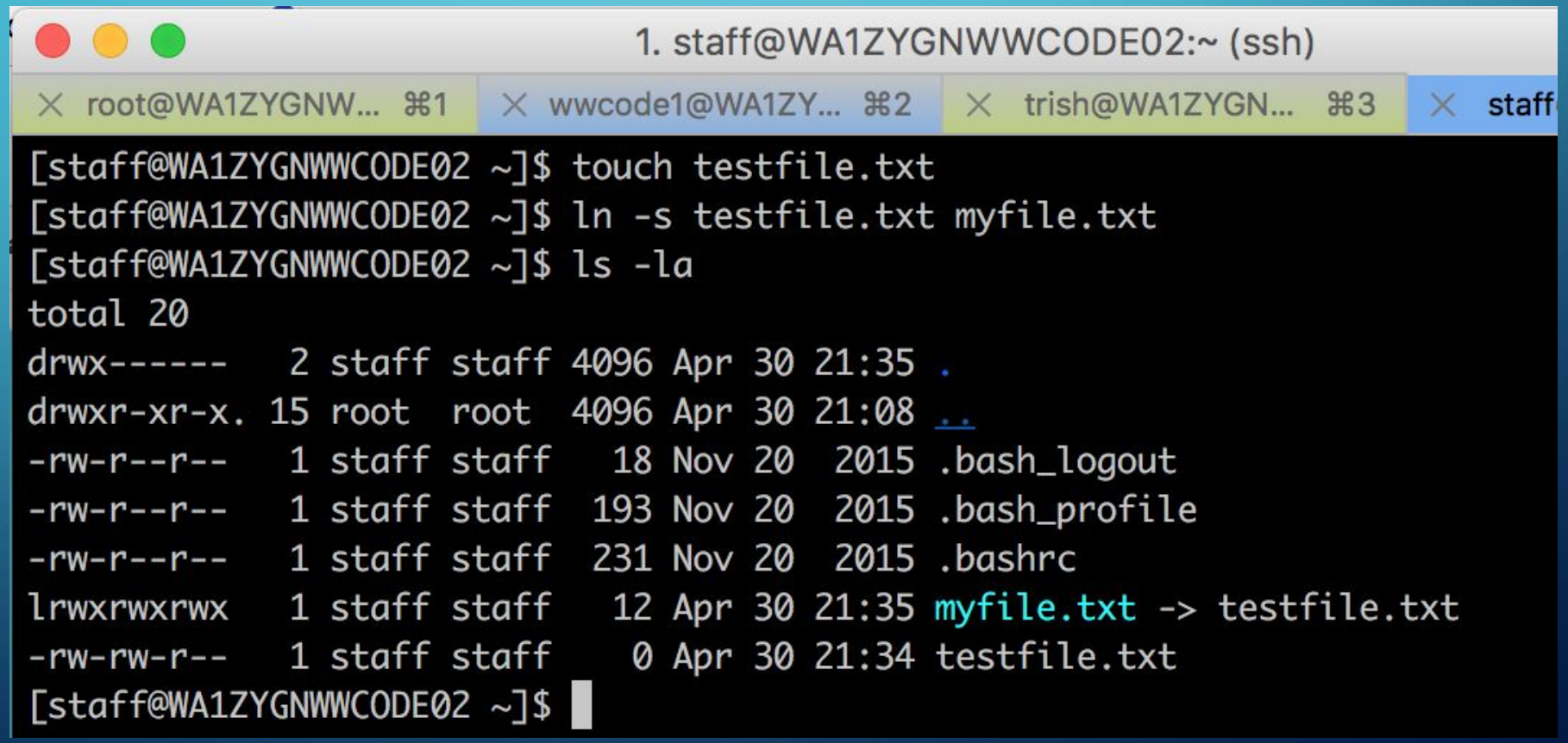


HANDS ON: Create a link

$ touch testfile.txt

$ ln -s testfile.txt myfile.txt

$ ls -la



What's going on?

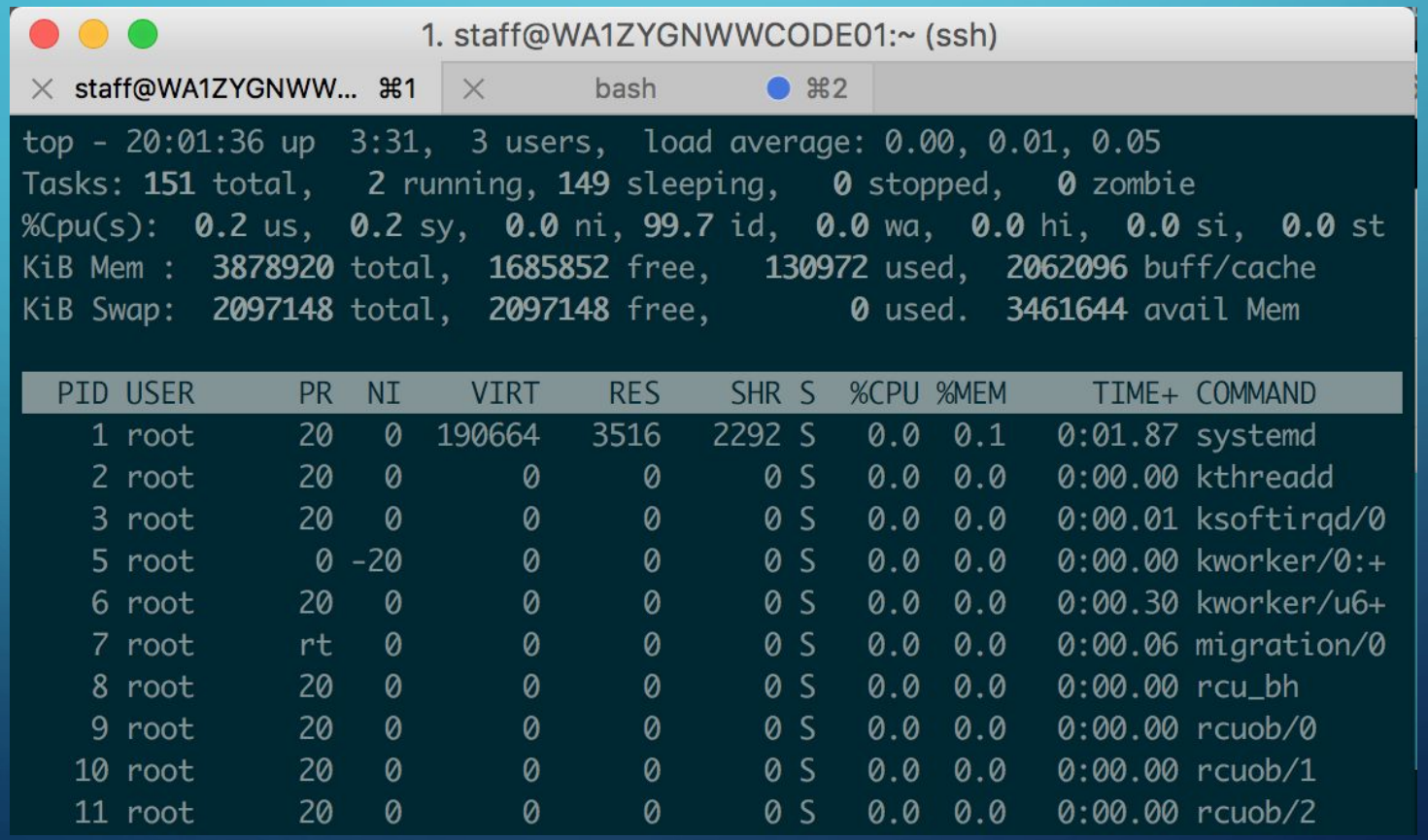
Knowing how to look up:

- what processes are running
- how much memory is being used (RAM and Disk)
- what's the ip address?
- how to kill a process

CPU: top or ps

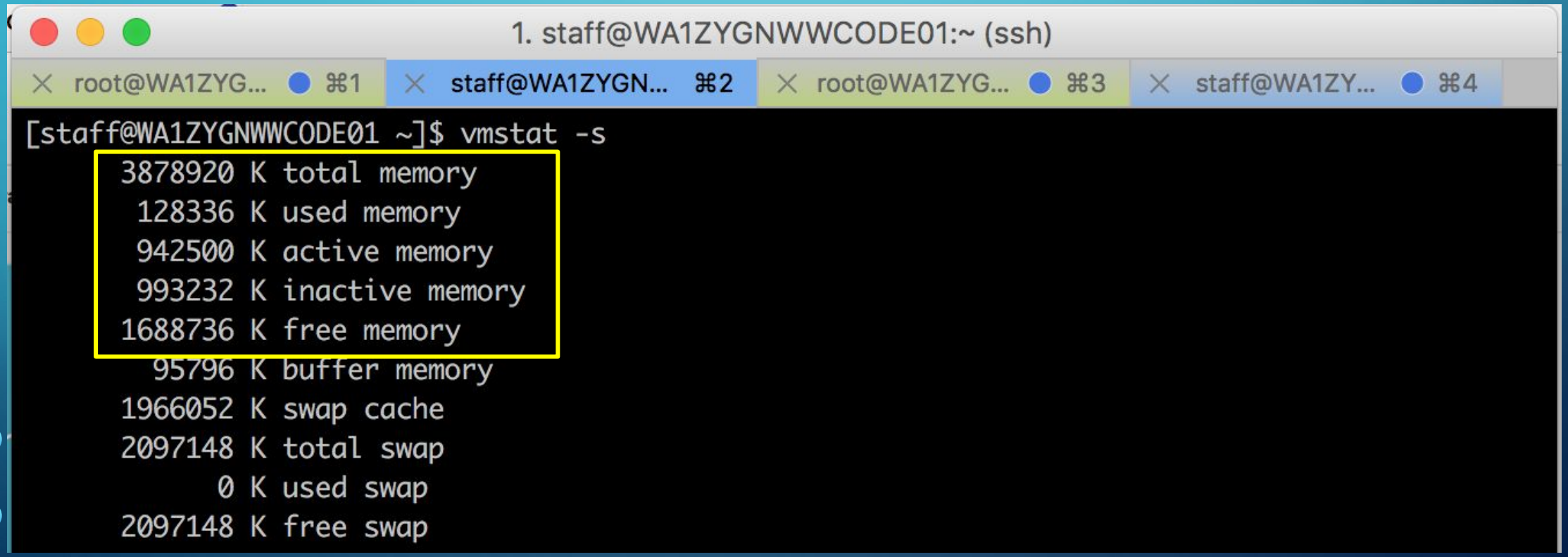
$ top

displays updated sorted info on processes



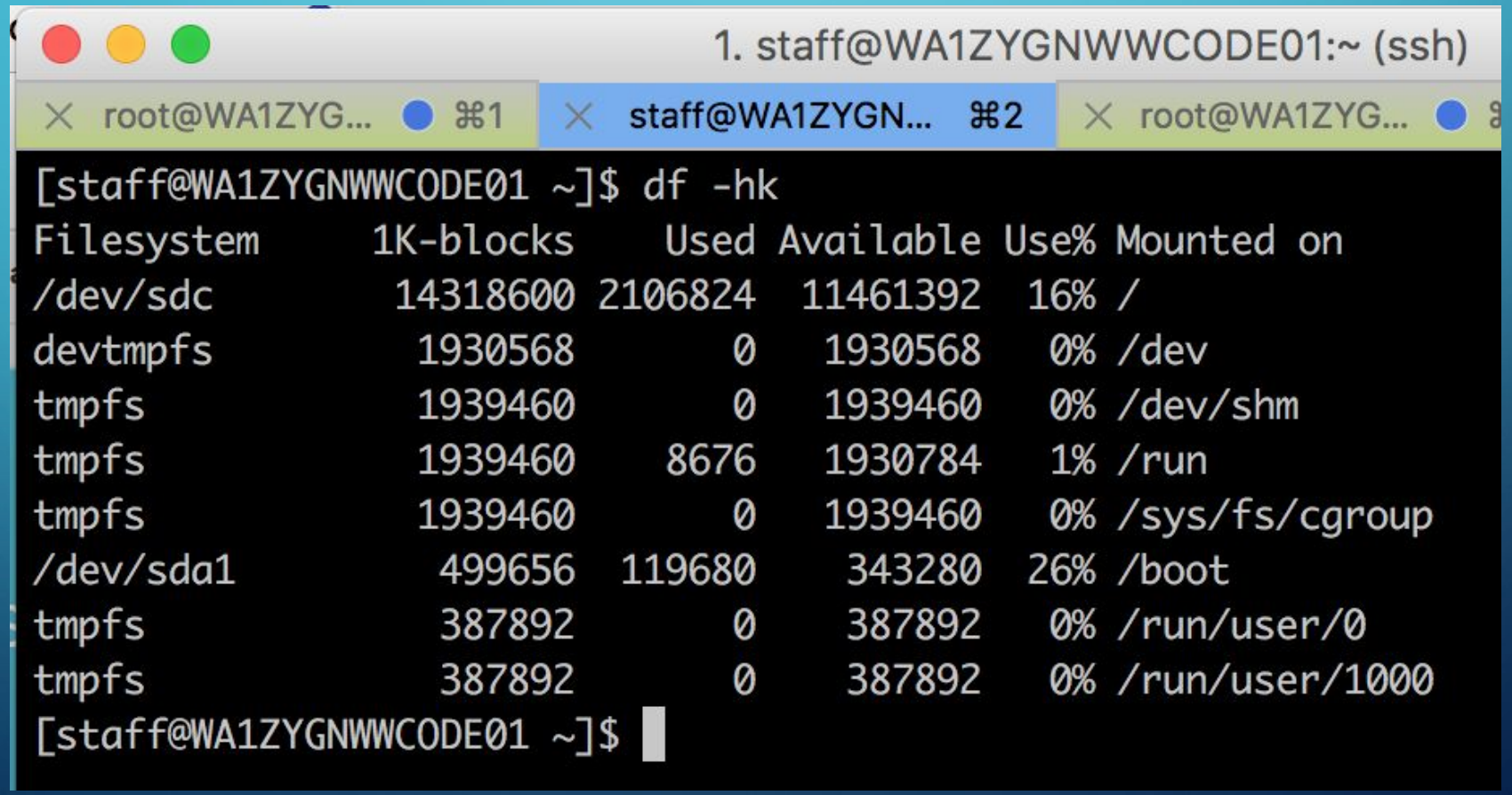
Memory: vmstat

$ vmstat -s



Disk: df (display free disk space)

$ df -hk



Network

$ ifconfig -a

1. staff@WA1ZYGNWWCODE01:~ (ssh)

root@WA1ZYG... ⌘1 | staff@WA1ZYGN... ⌘2 | root@WA1ZYG... ⌘3 | staff@WA1ZY... ⌘4

```
[staff@WA1ZYGNWWCODE01 ~]$ ifconfig -a
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.82.36.12  netmask 255.255.255.0  broadcast 10.82.36.255
        inet6 fe80::20c:29ff:fe57:8739  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:57:87:39  txqueuelen 1000  (Ethernet)
        RX packets 113456  bytes 435708432 (415.5 MiB)
        RX errors 0  dropped 40  overruns 0  frame 0
        TX packets 61698  bytes 5042060 (4.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 0  (Local Loopback)
        RX packets 18  bytes 936 (936.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18  bytes 936 (936.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

KILL! How to give a process the ax

$ kill <PID>

signal or terminate process by ID

$ kill -9 <PID>

terminates process and all dependent processes with extreme prejudice

$ killall <ProcessName>

kills all processes with process name

LIVE DEMO

Compressing files: tar or gzip

$ tar -czvf <tarfile.tar.gz> <file-to-be-compressed>

Create, Zip, Verbose, Filename

$ tar -xvf <tarfile.tar.gz>

EXtract, Verbose, Filename

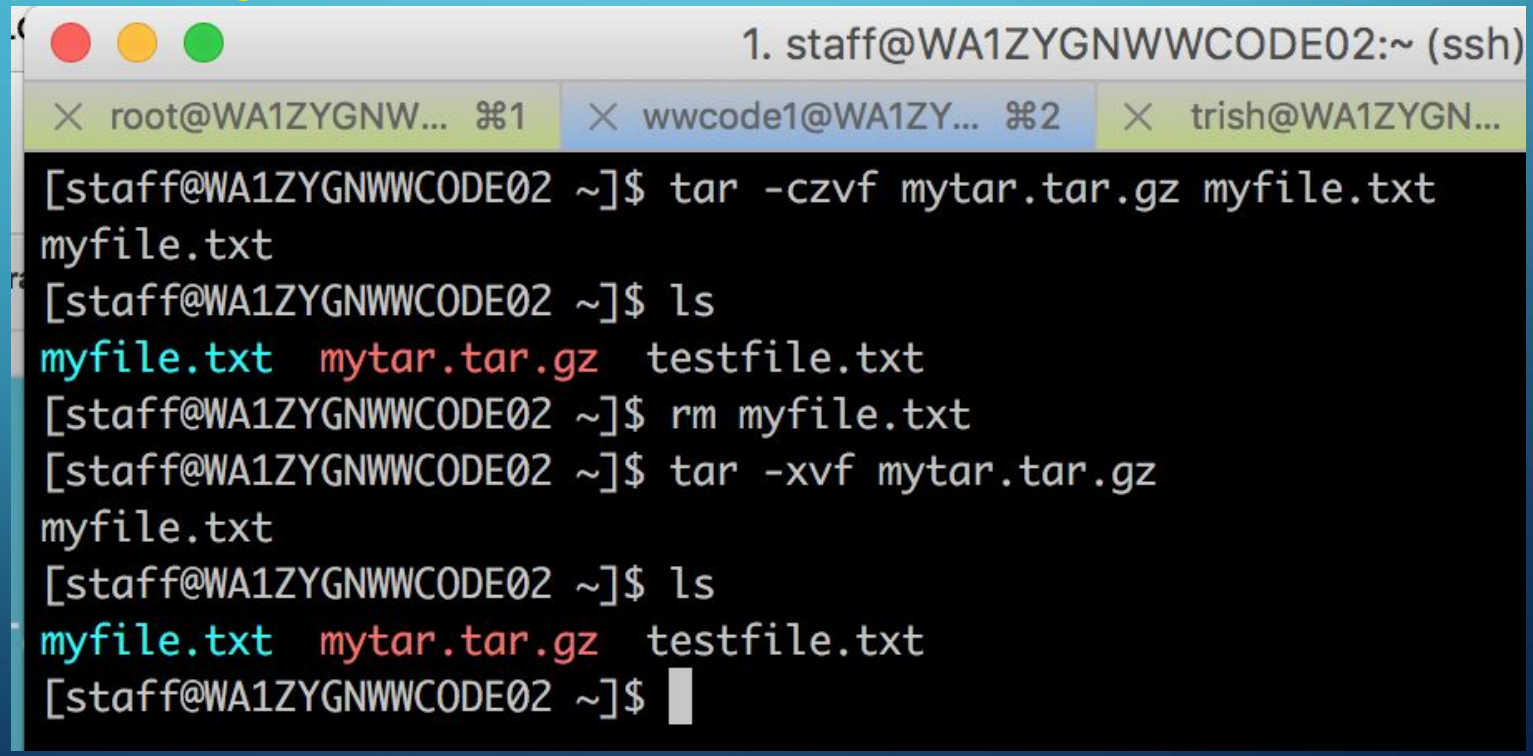
HANDS ON: Create a tar

$ tar -czvf mytar.tar.gz myfile.txt

$ ls; rm myfile.txt;ls

$ tar -xvf mytar.tar.gz

$ ls -la



Remote Shell Login - SSH

Login into a remote box - for example a server hosting your application.Secured with key

$ ssh <username>@<hostname>

type ‘exit’ to leave the session

Secure Copy - SCP

Securely copies a file from local to remote and vice versa.

Local to remote:

$ scp <local_file_path> <username>@<hostname>:<path>

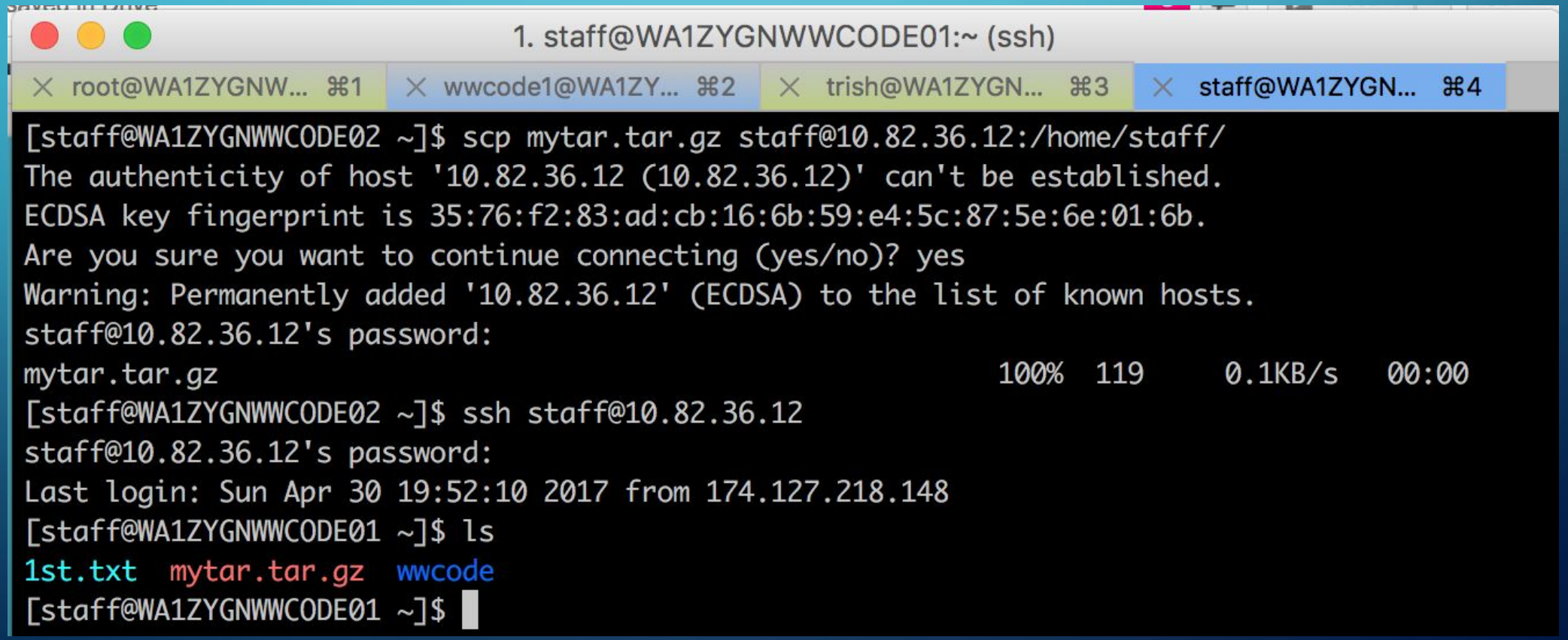
Remote to local:

$ scp <username>@host:path <local_file_path>

HANDS ON: SSH & SCP

$ scp mytar.tar.gz staff@10.82.36.12:/home/staff/

$ ssh staff@10.82.36.12



RBAC

Role-based access control (**RBAC**) is a method of regulating access to computer or network resources based on the roles of individual users within an enterprise. In this context, access is the ability of an individual user to perform a specific task, such as view, create, or modify a file.

SUDO: With Great Power Comes Great Responsibility

$ sudo <command>

executes command as a sudo user

“superuser do” (e.g. root, admin)

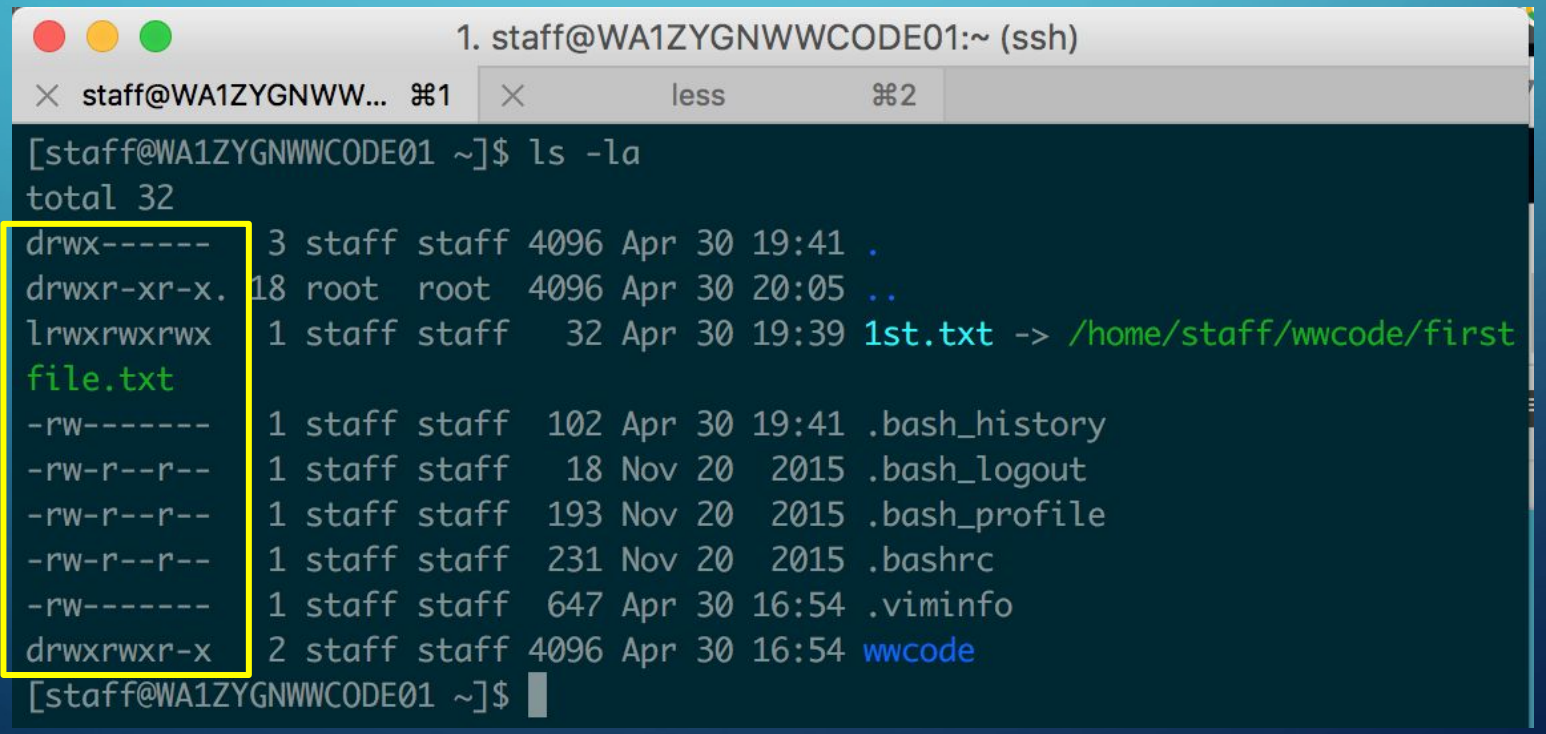
CHMOD

change file access control:

3 Groups: Owner, Group, Others

3 Types: Read, Write, Executable

$ chmod 777 <filename/foldername>



CHOWN

change file owner and group

$ chown <user>:<group> <filename>

$ chown -R <user>:<group> <foldername>

-R for recursive, only for folders

LIVE DEMO

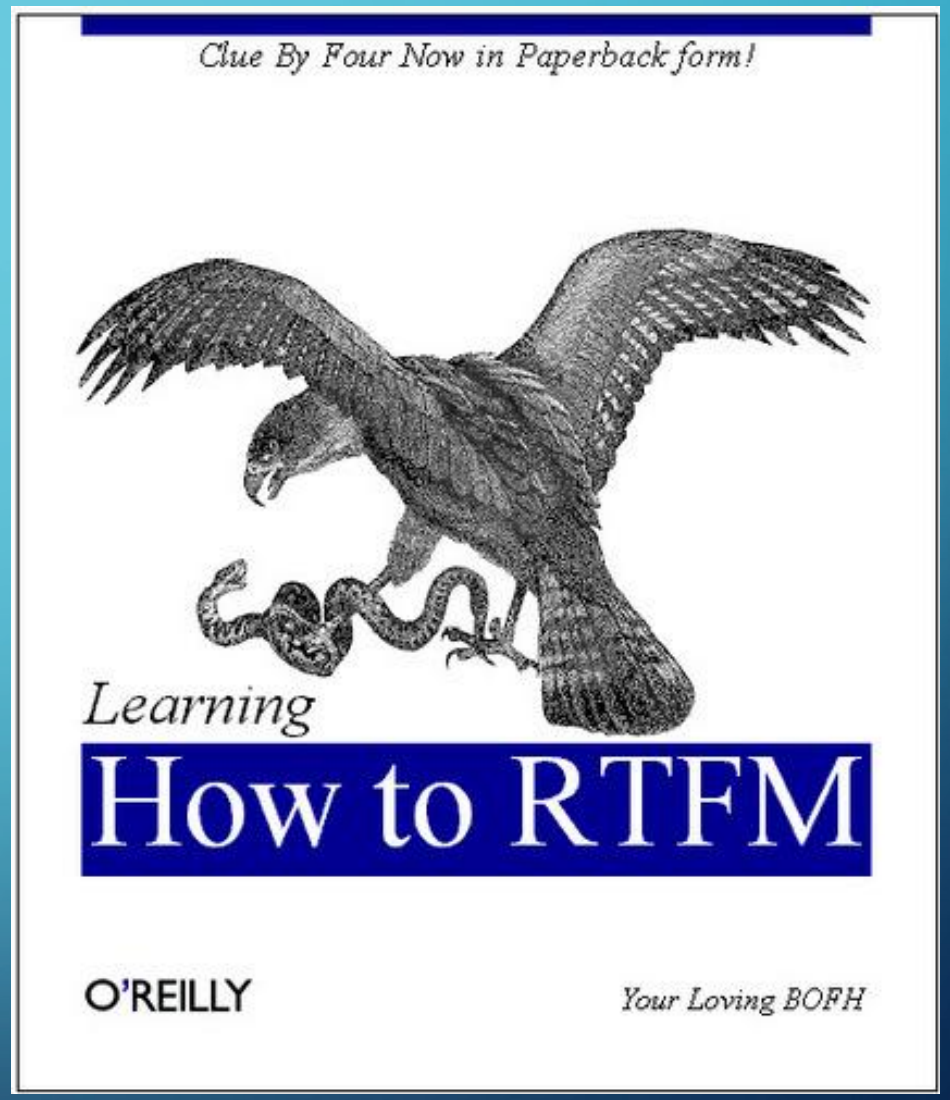
Google that Sh*t

When in doubt, search..

https://www.linux.com/

https://www.stackexchange.com

https://stackoverflow.com/



What's Next? Feedback?

- June Kubernettes Study Night, Whats this Container Stuff, How do I get started with MiniKube?

- More Command Line!!!
  - Terminals
  - find
  - Linking commands with pipe
  - Standard in and standard out
  - Running things in the background
  - Shell scripts
  - Customize your bash profile
  - awk
  - Sed