SHELL SCRIPTS

- Shell scripts and functions are both **interpreted** by the shell command interpreter
- Standardize and **automate the performance of routing admin tasks** that free up time for more interesting tasks
- Act as a **source of documentation** for the steps to be taken to complete a given task
- Linux is **casE SensiTive** and special characters have meanings in shell scripts \ / ; , . ~ # $? & * ( ) [ ] ‘ ’ " + - ! ^ = | < >
- Before the start of any shell script, we **declare the shell** that needs to be used **#!/bin/bash**
- Variables are defined as **varName=someValue** (It is important to not have spaces) and we display them as **echo “${varName}”** including the “ “ helps with variable values that have spaces in them

Quoting: Using “ or ‘ makes a difference

- echo "Path is $PATH" ## $PATH will be expanded
- echo 'I want to print $PATH' ## PATH will not be expanded

Arithmetic: ans=$((x + y))

Conditions : if [[-v name_of_var]] (-v checks if the variable is set)

```
.case $variable in
Match 1)
   commands for 1
   ;;
Match 2 )
   commands for 1
   ;;
*) # this means default value
   command for no match / default
.esac # case written in reverse
```

If block :

```
If command
Then
  do stuff here
Else
  do stuff
If
```

nested If block :

```
If command
Then
    If command
    Then
      do stuff here
    Else
      do stuff
    fi
Else
  do stuff
If
```

for block :

```
For loop_variable in list
Do
  commands
Done
```

While block :

```
while condition
Do
  commands
Done
```

Positional arguments:

$0, $1, $2, ... $9 refer to as the arguments to the command.

$0 refers to the actual command, program or the shell script that is being executed

Special Params:

$* and $@ are special Params that specify all command line arguments

Exit Status: 0 means success, [1 - 255) (255 excluded) anything means it was a failure

Linking > Commands:

; -> Runs command in sequence (num install ; npm build)

& -> Runs command in background (npm run &)

&& -> Second runs only if first is success (npm run && npm build)

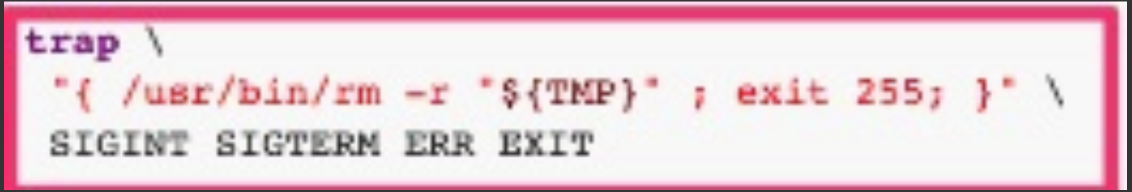
|| -> Second runs only if first fails (yarn install || npm install)

| -> Output of first command as input to second (ls -al | grep 3000)

> -> Output of first command concatenated with next (usually used for file writes)

TRAP command

This can be used to capture a Kill signal and perform clean up tasks before exiting form the script.



```
trap \
 "{ /usr/bin/rm -r "${TMP}" ; exit 255; }" \
 SIGINT SIGTERM ERR EXIT
```

User management

Add **an existing user** to a group:

- To one group: sudo usermod -a -G groupname username
- To multiple groups: sudo usermod -a -G group1,group2 username

Always use the -a (append) option when adding a user to a new group. If you omit the -a option, the user will be removed from any groups not listed after the -G option.

Create a group. : sudo **groupadd** groupname
Delete a group. : sudo **groupdel** groupname
Change user's primary group : sudo **usermod** -g groupname username
Create user and assign groups : sudo **useradd** -g users -G wheel,developers nathan

-g : refers to primary group
-G : refers to secondary groups