Gavin Faught - May 22nd, 2020
##
Submission File for Unit 5: Archiving and Logging Data Homework

Please edit and save this file while updating it with the commands and file (configuration and rules) edits you used to solve your homework.

`tar`: Create, extract, compress, and manage tar backup archives

Command to **extract** the `TarDocs.tar` archive to the current directory:

tar -xf TarDocs.tar -C /home/sysadmin/Projects
ls -alh (refer to HW5 - Tar#1.png)

Command to **create** the `Javaless_Doc.tar` archive from the `TarDocs/` directory, while excluding the `TarDocs/Documents/Java` directroy:

sudo tar cvf Javaless_Doc.tar --exclude "TarDocs/Documents/Java" TarDocs/Document

Ensuring `Java/` is not in the new `Javaless_Docs.tar` archive:

tar -tvf Javaless_Docs.tar | grep "Java" ---> no results. (refer to HW5 - Tar#4.png)

Bonus: Command to create an incremental archive called `logs_backup_tar.gz` with only changed files to `snapshot.file` for the `/var/log` directory.

sudo tar -cvvzWf logs_backup_tar.gz snapshot.snar --level 1 /var/log

`tar` Critical Thinking

Why wouldn't you use the options `-x` and `-c` at the same with `tar`?
-x is to extract a tar file. -c is to create a tar file. The tar command has either one or the other, not both because you can't
extract and create a tar file at the same time.

`cron`: Create, manage and automate various cron jobs

Cron job for backing up the `/var/log/auth.log` file:

```
# Every Wednesday at 6AM
# Minute  Hour  Day of Month  Month  Day of Week
#-----------------------------------------------
   0     6       *          *       3        tar -cvzf /auth.backup.tgz /var/log/auth.log
```

`bash scripting`: Write basic bash scripts

Brace expansion command to create the four subdirectories:

mkdir -p /home/sysadmin/backups/{freemem,diskuse,openlist,freedisk} (refer ot HW5 - bashScripting1.png)

Command and file edit to create `system.sh` (you can copy and paste it here):

touch system.sh
nano -l system.sh

- Within the script, have the following:

  ```bash
  #!/bin/bash
  free -h > ~/backups/freemem/free_mem.txt
  du -h > ~/backups/diskuse/disk_usage.txt
  lsof > ~/backups/openlist/open_list.txt
  df -h > ~/backups/freedisk/free_disk.txt     (refer to HW5 - bashScripting-3.png)
  ```

Command to make the `system.sh` script executable:

chmod u+x system.sh (refer to HW5 - bashScripting-4.png)

Commands to to confirm script's execution:

First, run the file. At the command line, type "./system.sh" (refer to HW5 - bashScripting-5.png)
Then, make sure the output went to the right path:

nano ~/backups/freemem/free_mem.txt

nano ~/backups/diskuse/disk_usage.txt

nano ~/backups/openlist/open_list.txt

nano ~/backups/freedisk/free_disk.txt (as an example. refer to HW5 -bashScripting-6.png)

Command to copy `system` to system-wide cron directory:

sudo cp system.sh /etc/cron.weekly (refer to HW5 -bashScripting-7.png)

`journalctl`: Perform various log filtering techniques

Command to return `journalctl` messages with priorities from emergency to error:

sudo journalctl -b 0 -p "emerg.."err"

Command to return `systemd-journald` messages:

sudo journalctl -b 0 -u systemd-journald | df -h | less

Comand to prune archived journal files except the most recent 2:

sudo journalctl --vacuum-files=2

Bonus Command to filter all log messages with priority levels between 0 and 2, output to `/home/sysadmin/Priority_High.txt`

(make sure in root!) sudo journalctl -p "emerg.."crit" > /home/sysadmin/Priority_High.txt

Bonus 2 Command and file edit to automate the last command in a daily cronjob:

crontab -e

- Within the `crontab` file, add the following:

  ```bash
  #Daily cronjob - everyday at 2am.
  0 2 * * * journal -p "emerg.."crit" > /home/sysadmin/Priority_High.txt
  ```

`rsyslog`: Priority based log file creation

Command and file edit to record all `mail` log messages, except for `debug` to `/var/log/mail.log`:

nano -l /etc/rsyslog.conf

- Add within the configuration file:

  ```bash
  mail.!debug   /var/log/mail.log
  ```

(THIS IS A BONUS QUESTION)
Command and file edit to record all `boot` log messages, except for `info` and `debug` to `/var/log/boot.log`:

nano -l /etc/rsyslog.conf

- Add within the configuration file:

local7.!info /var/log/boot.log

`logrotate`: Manage log file sizes

Command and file edit that backs up authentication messages to `/var/log/auth.log`:

- Run `sudo nano -l /etc/logrotate.conf` to edit the `logrotate` configuration file.

- Add within the configuration file:

  ```bash
  /var/log/auth.log {
    weekly
    rotate 7
    notifempty
    delaycompress
    missingok
  }
  ```

BONUS ACTIVITY `auditd`: Check for policy and file violations.

Command to verify `auditd` is active:

systemctl status auditd

Command and file edit to set number of retained logs and maximum log file size:

sudo su

sudo nano -l /etc/audit/auditd.conf

- Add within the configuration file:

  ```bash
  # line 13
  num_logs=7

  # line 12
  max_log_file=35
  ```

Command and file edit using `auditd` itself to set rules for `/etc/shadow`, `/etc/passwd` and `/var/log/auth.log`:

sudo nano -l /etc/audit/rules.d/audit.rules

- Add within the `rules` file:

  ```bash
  -w /etc/shadow -p wra -k hashpass_audit
  -w /etc/passwd -p wra -k userpass_audit
  -w /var/log/auth.log -p wra -k authlog_audit

  ```

Command to restart `auditd`:

sudo systemctl restart auditd

Command to list all `auditd` rules:

sudo auditctl -l

Command to produce an audit report:

sudo aureport -au

Command to use `auditd` to watch `/var/log/cron`:

sudo auditctl -w /var/log/cron

Command to re-verify `auditd` rules:

sudo auditctl -l

Gavin's Corner
---
at command - like a cron job, but is executed only once. Make sure your std
is installed on your Linux machine.