

Git and Linux SSH Script

💡 Version control systems are prevalent in today's IT environments. A firm knowledge of how to store different iterations of critical configurations and source code is therefore important.

You should already have a git repository, though it may only contain your wiki. We will add configurations, source files and scripts to your repository to make it far more useful in this class and beyond.

💣 Sensitive information such as passwords, SSH Private Keys are frequently exposed by inattentive repository owners. Anything other than sample, non-production passwords should not be used or documented. Private keys should also be left out of even private repositories. #SecOps

Part 1. GIT

Install git on docker01

If you haven't done so already, install git on docker01.

The clone

The following screenshot shows gmcyber's private repo being pulled down to docker01.

💡 Replace the example repo and email with your own!

```
hermione@docker01-hermione: ~  
hermione@docker01-hermione:~$ pwd  
/home/hermione  
hermione@docker01-hermione:~$ hostname  
docker01-hermione  
hermione@docker01-hermione:~$ git clone https://github.com/gmcyber/tech-journal-private  
Cloning into 'tech-journal-private'...  
Username for 'https://github.com': gmcyber  
Password for 'https://gmcyber@github.com':  
remote: Enumerating objects: 137, done.  
remote: Counting objects: 100% (137/137), done.  
remote: Compressing objects: 100% (105/105), done.  
remote: Total 137 (delta 34), reused 114 (delta 14), pack-reused 0  
Receiving objects: 100% (137/137), 27.10 KiB | 1.13 MiB/s, done.  
Resolving deltas: 100% (34/34), done.  
hermione@docker01-hermione:~$
```

Create a directory structure

If you haven't done so already, create a directory structure within your local repository that is organized to capture your configuration information.

```
hermione@docker01-hermione: ~/tech-journal-private/SYS265/docker01

hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ pwd
/home/hermione/tech-journal-private/SYS265/docker01
hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ ls
50-cloud-init.yaml  cloud.cfg  docker-compose.yml  docker.txt  hosts
hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$
```

Add, commit and push.

💡 Replace the example username and email with your own

```
hermione@docker01-hermione: ~/tech-journal-private/SYS265/docker01

hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ echo "docker01 configuration" >> README.md
hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ git add .
hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   README.md

hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ git config user.email gmcycber@greenmountaincyber.com
hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ git config user.name gmcycber
hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ git commit -m "added a readme"
[master a91a6d8] added a readme
 1 file changed, 4 insertions(+)
 create mode 100644 SYS265/docker01/README.md
hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ git push
```

Deliverable 1. A screenshot similar to the following that shows the configuration files (not your wiki) added to your github site. Note how the README.md is displayed

The screenshot shows a GitHub repository page for 'gmcyber / tech-journal-private'. The repository is private and has 1 watch, 0 stars, and 0 forks. The main navigation bar includes links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. The current branch is 'master'. The repository path is 'tech-journal-private / SYS265 / docker01 /'. There are buttons for 'Create new file', 'Upload files', 'Find file', and 'History'. A commit by 'gmcyber' is shown, titled 'added a readme', with the latest commit hash 'a91a6d8' made 3 minutes ago. Below the commit, a list of files is displayed:

File	Commit	Time
50-cloud-init.yaml	Docker Project	7 days ago
README.md	added a readme	3 minutes ago
cloud.cfg	Docker Project	7 days ago
docker-compose.yml	Docker Project	7 days ago
docker.txt	added docker install	6 days ago
hosts	Docker Project	7 days ago

Below the file list, the 'README.md' file is expanded, showing the content: 'docker01 configuration docker01 configuration docker01 configuration docker01 configuration'.

Git clone

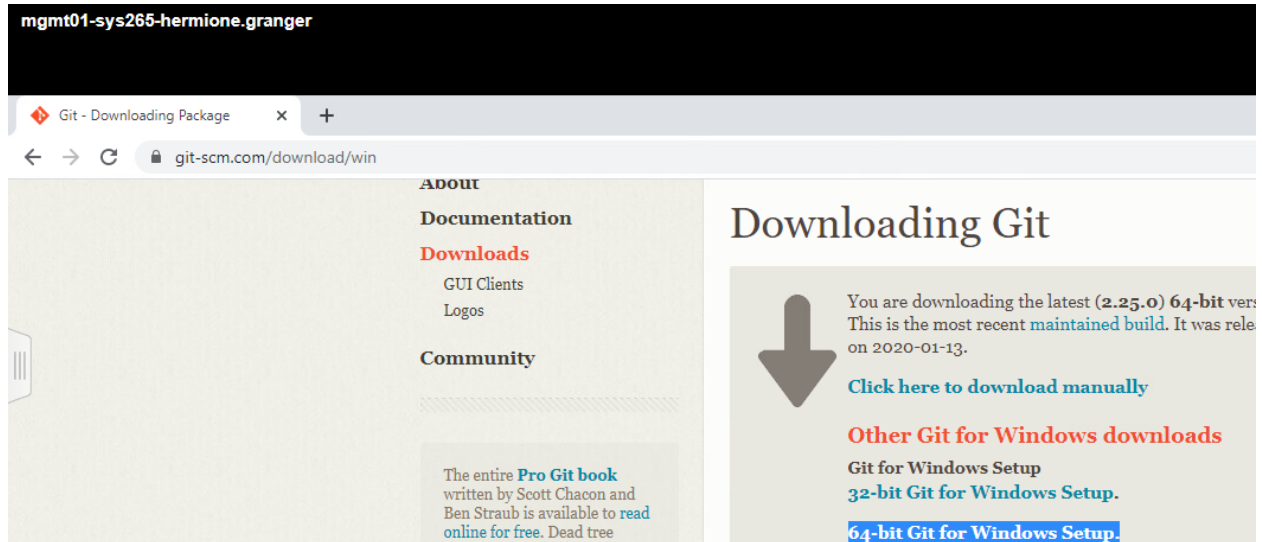
Once pushed, you can always recover files deleted locally by doing a git checkout. Delete the README.md file from the local repo on docker01.

```
hermione@docker01-hermione: ~/tech-journal-private/SYS265/docker01
hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ cat README.md
docker01 configuration
docker01 configuration
docker01 configuration
docker01 configuration
hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ rm README.md
hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ git checkout .
hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$ cat README.md
docker01 configuration
docker01 configuration
docker01 configuration
docker01 configuration
hermione@docker01-hermione:~/tech-journal-private/SYS265/docker01$
```

Deliverable 2. Provide a screenshot similar to the one above

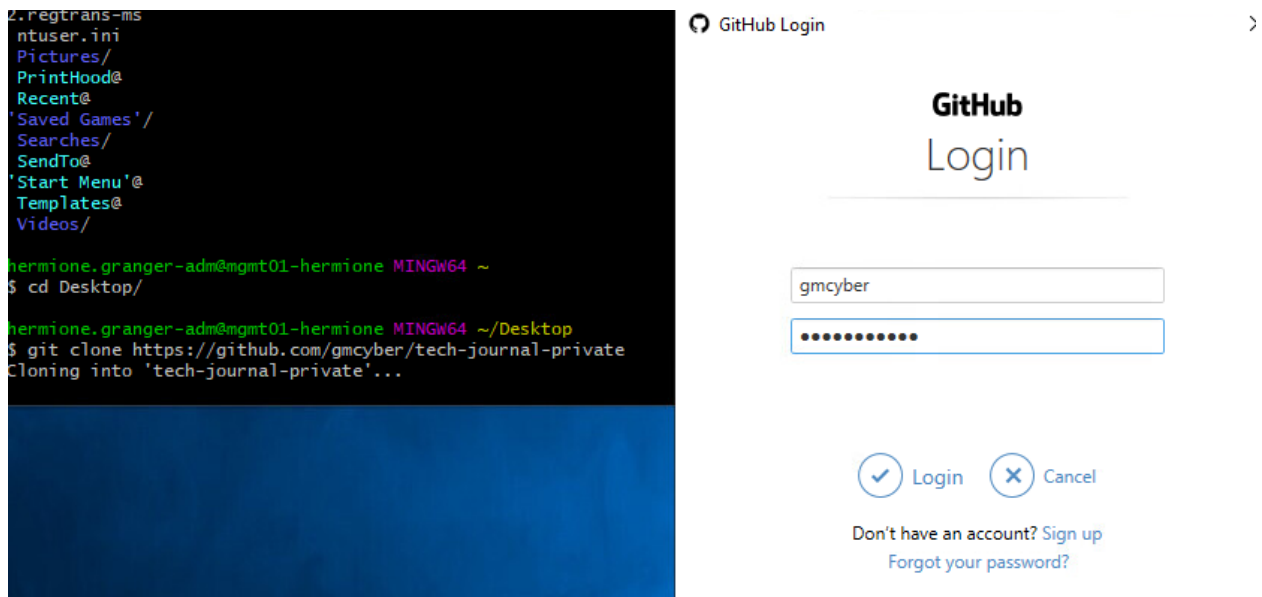
Git on Windows

Install the 64-bit version of Git on mgmt01 using defaults.



Clone your repo on mgmt01

Find and execute git-bash and then clone your repo in much the same way as you did on docker01.



Modify your repo

Create a mgmt01 directory with a README.md file with some arbitrary content.

```
MINGW64:/c/Users/hermione.granger-adm/Desktop/tech-journal-private/S...
remote: Total 142 (delta 36), reused 118 (delta 15), pack-reused 0
Receiving objects: 100% (142/142), 27.43 KiB | 319.00 KiB/s, done.
Resolving deltas: 100% (36/36), done.

hermione.granger-adm@mgmt01-hermione MINGW64 ~/Desktop
$ cd tech-journal-private/SYS265/mgmt01/

hermione.granger-adm@mgmt01-hermione MINGW64 ~/Desktop/tech-journal-private/SYS265/mgmt01 (master)
$ hostname
mgmt01-hermione

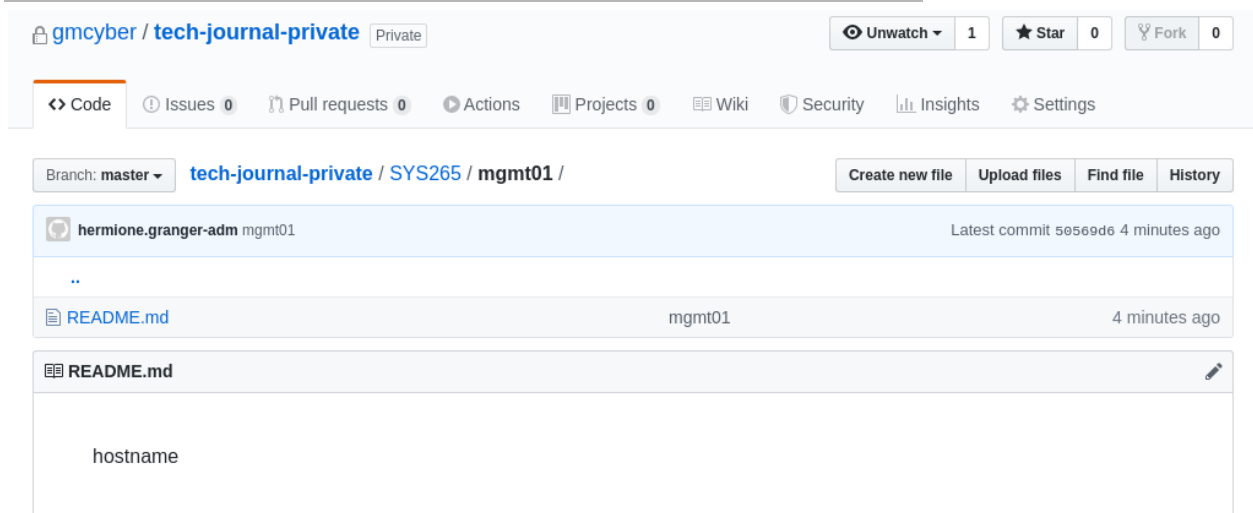
hermione.granger-adm@mgmt01-hermione MINGW64 ~/Desktop/tech-journal-private/SYS265/mgmt01 (master)
$ echo hostname >> README.md

hermione.granger-adm@mgmt01-hermione MINGW64 ~/Desktop/tech-journal-private/SYS265/mgmt01 (master)
$ cat README.md
hostname

hermione.granger-adm@mgmt01-hermione MINGW64 ~/Desktop/tech-journal-private/SYS265/mgmt01 (master)
$
```

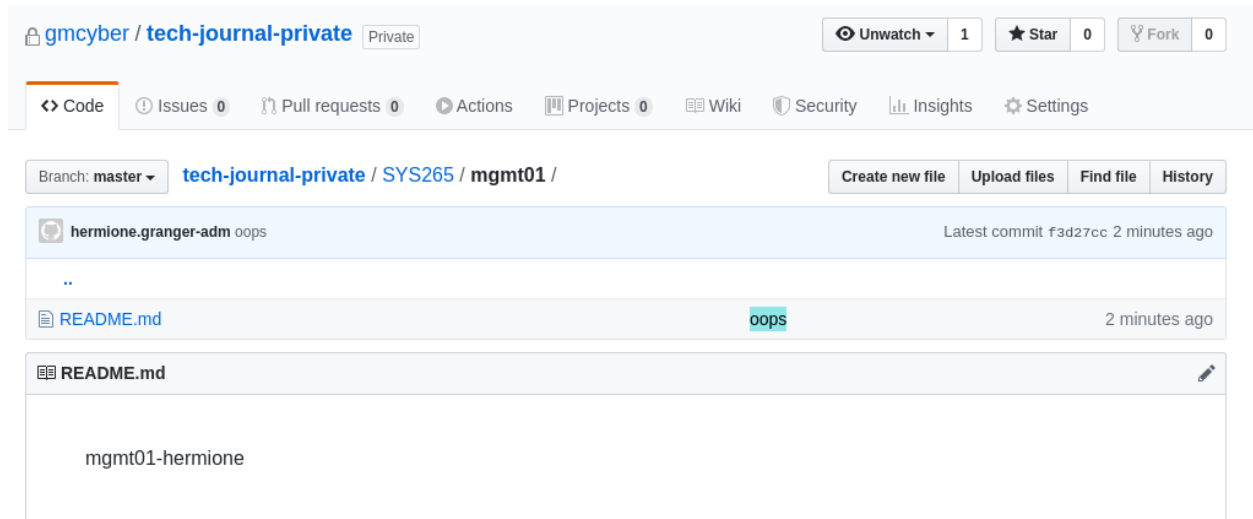
Add, commit, and push to github. You may notice that git is caching your credentials on Windows. It also grabs your username and assumes an email.

Deliverable 3. Screenshot similar to the one below



Oops ... the intent was to put the actual hostname in the README file. Figure out the correct use of echo to make that happen.

Deliverable 4. Re-commit with the comment, "oops", and push. Provide a screenshot similar to the one below.



Git pull

Now the local repository on docker01 is out of sync with the online version because of the push from mgmt01 that is not reflected in the local repo on docker01. Let's sort that out.

Deliverable 5. Provide a screenshot similar to the one below, that shows README.md being pulled.


```
hermione@docker01-hermione:~$ cd tech-journal-private/
hermione@docker01-hermione:~/tech-journal-private$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
hermione@docker01-hermione:~/tech-journal-private$ git pull
Username for 'https://github.com': gmcyber
Password for 'https://gmcyber@github.com':
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 10 (delta 2), reused 9 (delta 1), pack-reused 0
Unpacking objects: 100% (10/10), done.
From https://github.com/gmcyber/tech-journal-private
   a91a6d8..f3d27cc  master    -> origin/master
Updating a91a6d8..f3d27cc
Fast-forward
  SYS265/mgmt01/README.md | 1 +
  1 file changed, 1 insertion(+)
  create mode 100644 SYS265/mgmt01/README.md
hermione@docker01-hermione:~/tech-journal-private$
```

Part 2: Hardening SSH

Clone your tech journal to web01. You will need to install git.

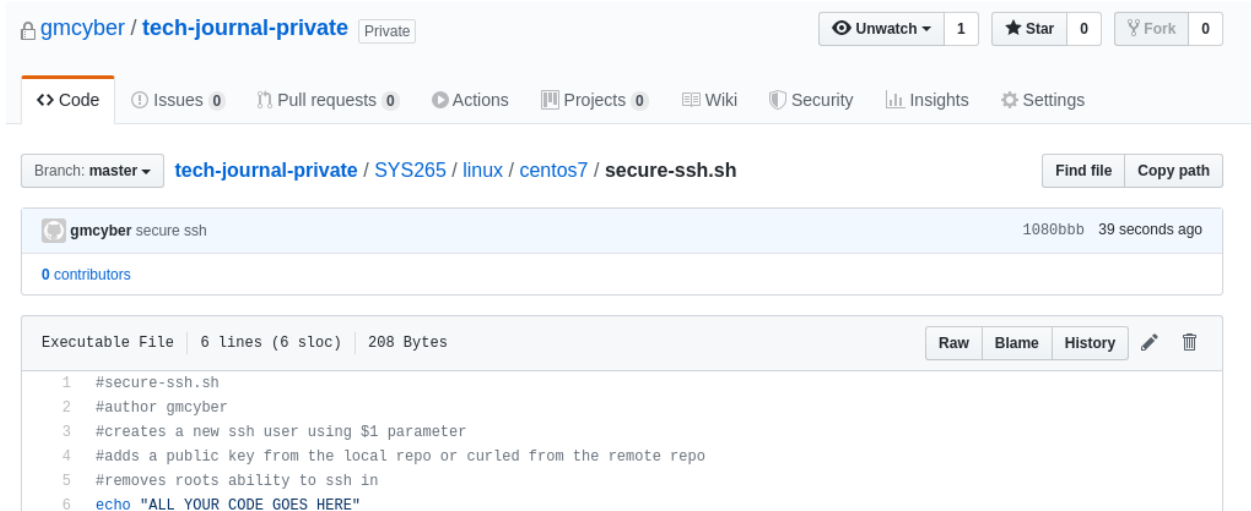
Let's organize our local repository a bit and then push the changes up to github. We are going to create a few directories and a shell script called secure-ssh.sh

 Note: Though the screenshot below refers to CentOS, docker01 is actually running on Ubuntu if you want to make the change in directory name. The script should work on both CentOS and Ubuntu; though on Ubuntu, you do not need to disable root ssh (that is done already).

```
hermione@web01-hermione:~/tech-journal-private/SYS265/linux
[hermione@web01-hermione SYS265]$ pwd
/home/hermione/tech-journal-private/SYS265
[hermione@web01-hermione SYS265]$ mkdir -p linux/{public-keys,centos7}
[hermione@web01-hermione SYS265]$ cd linux/
[hermione@web01-hermione linux]$ ls
centos7  public-keys
[hermione@web01-hermione linux]$ nano centos7/secure-ssh.sh
[hermione@web01-hermione linux]$ chmod +x centos7/secure-ssh.sh
[hermione@web01-hermione linux]$ cat centos7/secure-ssh.sh
#secure-ssh.sh
#author gmcyber
#creates a new ssh user using $1 parameter
#adds a public key from the local repo or curled from the remote repo
#removes roots ability to ssh in
echo "ALL YOUR CODE GOES HERE"
```

Push changes to github

Deliverable 6. Provide a screenshot that shows your submitted secure-ssh.sh file




RSA keypair

Create an RSA Keypair on web01, with no passphrase required. Copy the PUBLIC key to the local repo, see the last two lines.


```
hermione@web01-hermione:~/tech-journal-private/SYS265/linux/public-keys
[hermione@web01-hermione public-keys]$ ssh-keygen -t rsa -C "sys265"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hermione/.ssh/id_rsa):
Created directory '/home/hermione/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hermione/.ssh/id_rsa.
Your public key has been saved in /home/hermione/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Xi/gJlRG10Ft8YKrUHbKLiK8K3HcD/ihTs5Rlx0uk7Y sys265
The key's randomart image is:
+---[RSA 2048]---+
|      .+++. |
|      . ...O. |
|      * o... |
|      @ + . . |
|  . o. S * . |
| ..+.* O o   |
|  o=o.+E = . |
| .+.+.+.+. |
|  o*.        |
+---[SHA256]-----+
[hermione@web01-hermione public-keys]$ cp ~/.ssh/id_rsa.pub .
[hermione@web01-hermione public-keys]$
```

Now add, commit and push your web01 modifications.

 Note: if I see a private key in your repo, you will get a 0 on this assignment. #SecOps

Deliverable 7. Provide a screenshot similar to the one below that shows your public key on github.



Hardening Script

On docker01, pull to synchronize your repo.

```
hermione@docker01-hermione: ~/tech-journal-private
hermione@docker01-hermione:~$ cd tech-journal-private/
hermione@docker01-hermione:~/tech-journal-private$ git pull
Username for 'https://github.com': gmcyber
Password for 'https://gmcyber@github.com':
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 12 (delta 2), reused 11 (delta 1), pack-reused 0
Unpacking objects: 100% (12/12), done.
From https://github.com/gmcyber/tech-journal-private
   f3d27cc..57e76d7  master    -> origin/master
Updating f3d27cc..57e76d7
Fast-forward
  SYS265/linux/centos7/secure-ssh.sh | 6 ++++++
  SYS265/linux/public-keys/id_rsa.pub | 1 +
  2 files changed, 7 insertions(+)
  create mode 100755 SYS265/linux/centos7/secure-ssh.sh
  create mode 100644 SYS265/linux/public-keys/id_rsa.pub
hermione@docker01-hermione:~/tech-journal-private$
```

The following screenshot shows the manual creation of a user that can only login via RSA Private Key. You are going to need to figure out how to create such a user using a script. In this case, SYS265 is the created user.

```
hermione@docker01-hermione: ~/tech-journal-private
hermione@docker01-hermione:~/tech-journal-private$ sudo useradd -m -d /home/sys265 -s /bin/bash sys265
hermione@docker01-hermione:~/tech-journal-private$ sudo mkdir /home/sys265/.ssh
hermione@docker01-hermione:~/tech-journal-private$ sudo cp SYS265/linux/public-keys/id_rsa.pub /home/sys265/.ssh/authorized_keys
hermione@docker01-hermione:~/tech-journal-private$ sudo chmod 700 /home/sys265/.ssh
hermione@docker01-hermione:~/tech-journal-private$ sudo chmod 600 /home/sys265/.ssh/authorized_keys
hermione@docker01-hermione:~/tech-journal-private$ sudo chown -R sys265:sys265 /home/sys265/.ssh
hermione@docker01-hermione:~/tech-journal-private$
```

Now test your manual configuration web01->docker01

Deliverable 8. Provide a screenshot, showing the passwordless login.

```
sys265@docker01-hermione: ~
sys265@docker01-hermione:~$ exit
logout
Connection to docker01-hermione closed.
[hermione@web01-hermione ~]$ clear
[hermione@web01-hermione ~]$ ssh sys265@docker01-hermione
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-74-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Feb 16 18:00:48 UTC 2020

System load:  0.0          Processes:           165
Usage of /:   33.9% of 15.68GB Users logged in:        1
Memory usage: 19%         IP address for ens160: 10.0.5.12
Swap usage:   0%          IP address for docker0: 172.17.0.1

 * Multipass 1.0 is out! Get Ubuntu VMs on demand on your Linux, Windows or
   Mac. Supports cloud-init for fast, local, cloud devops simulation.

https://multipass.run/
```

Modify your secure-ssh.sh script to do the following:

- 1) Using a passed parameter for username such as ./secure-ssh.sh testuser12, create a passwordless user such that the user with the associated private key on web01 can login

without password as shown in the following example:

```
sys265@docker01-hermione:~  
[hermione@web01-hermione linux]$ ssh sys265@docker01-hermione  
[sys265@docker01-hermione ~]$
```

- 2) Root should not be able to login (see the "PermitRootLogin setting in sshd_conf). Note, docker01 is Ubuntu and this is already the case. For this lab, you get a pass on this. Were this a CentOS server, you would need to fix that.

Deliverable 9. Cat the script syntax, show a test running of your script on docker01, and the passwordless ssh login from web01 (similar to the one above).

Deliverable 10. Provide a direct link to the updated secure-ssh.sh file on github.

Deliverable 11. Provide a link to your tech journal. Make sure you spend some time on reviewing/explaining what you did with git, ssh and keys. We are raising the bar on tech journal entries. They should actually be useful, accessible and well formatted.