

# USAP Assignment 1

---

s3668468 Cameron Tavelli

## Set up environment

---

xUbuntu (A linux distribution) is already installed on my system and the following steps are for linux users

### Install TigerVNC and NMAP

1. Update the system repos
  - i. `sudo apt update -y`
2. Install TigerVNC
  - i. `sudo apt install tigervnc-*`
  - ii. *Enter Yes*
  - iii. **SCREENSHOT**
3. Install NMAP
  - i. `sudo apt install nmap`
  - ii. *Enter Yes*
  - iii. **SCREENSHOT**

### Install Raspberry Pi OS

---

<https://www.raspberrypi.org/downloads/raspberry-pi-os/>

1. Insert micro SD card reader with SD card into the system
  - i. **SCREENSHOT**
2. Download Raspberry Pi OS (32-bit) with desktop and recommended software image from the Raspberry Pi website
3. Extract the .zip
  - i. `unzip 2020-08-20-raspbios-buster-armhf-full.zip`

#### 4. Unmount partitions

- i. `lsblk -p`
- ii. `sudo umount -f /dev/sdb1`
- iii. **SCREENSHOT**

#### 5. Write the Raspberry Pi image onto the SD Card

- i. `sudo dd if=2020-08-20-raspbian-buster-armhf-full.img  
of=/dev/sdb bs=4M status=progress`
- ii. **SCREENSHOT**

#### 6. Insert the SD Card into the Raspberry Pi

#### 7. Run the configurator on the device and enable SSH

#### 8. Use nmap to verify the IP of the Raspberry Pi

- i. `nmap -sn 192.168.0.0/24`
- ii. **SCREENSHOT**

#### 9. SSH into the Raspberry Pi

- i. `ssh pi@192.168.x.x`
- ii. **SCREENSHOT**
- iii. Update the system: `sudo apt update -y && sudo apt upgrade  
-y`

#### 10. Uninstall RealVNC and install TigerVNC

- i. `sudo apt remove realvnc-*`
- ii. `sudo apt install -y tigervnc-standalone-server`

#### 11. Create a service to run the VNC Server

- i. `sudo nano /etc/systemd/system/vncserver@.service`  
  
`[Unit]  
Description=Remote Desktop Service (VNC)  
After=syslog.target network.target  
  
[Service]  
Type=forking`

```
User=pi
WorkingDirectory=/home/pi
ExecStart=/usr/bin/tigervncserver %i -geometry 1440x900 -alw
ExecStop=/usr/bin/tigervncserver -kill %i

[Install]
WantedBy=multi-user.target
```

## 12. Start the VNC Server on the Pi and enable it to start on boot

- i. tigervncserver
  - a. *Enter a password*
  - b. *Verify password*
  - c. *No*
- ii. `sudo systemctl enable vncserver@:1`
- iii. `sudo systemctl restart vncserver@:1`

## 13. Connect to the Raspberry Pi from a client machine

- i. `vncviewer raspberrypi.local:1`
- ii. **SCREENSHOT**

## Setting up a RAID configuration

1. Prepare two identical USB drives
2. Insert a USB drive into the Raspberry Pi and format it for RAID
  - i. `lsblk -p`
  - ii. `gdisk /dev/sda` 3. n 4. *Enter* x3 5. fd00 6. p 7. w 8. Y
  - iii. Repeat this step for other USB drive ( `/dev/sdb` )
    - a. **SCREENSHOT**
3. Once both USB drives are partitioned for Linux RAID, install mdadm and create a RAID1 configuration
  - i. `sudo apt install mdadm`
  - ii. Verify the changes on the USB drives with `sudo mdadm -E /dev/sd[ab]1`
  - iii. Create a RAID1 configuration on `/dev/md0`
    - a. `sudo mdadm --create /dev/md0 --level=mirror --raid-`

```
devices=2 /dev/sd[ab]1
```

b. y

iv. Watch the RAID configuration being created

a. watch cat /proc/mdstat

b. **SCREENSHOT**

4. Once RAID is complete, create a file system on RAID /dev/md0

i. sudo mkfs -t ext4 /dev/md0

5. Create a mount point to attach new file system

i. sudo mkdir /mnt/md0

6. Mount file system

i. sudo mount /dev/md0 /mnt/md0

7. Check if the mount was successful

i. df -h /mnt/md0

ii. **SCREENSHOT**

8. Save the mdadm array layout (Can only be done in root)

i. sudo su

ii. mdadm --detail --scan /dev/md0 >> /etc/mdadm/mdadm.conf

iii. cat /etc/mdadm/mdadm.conf

iv. exit

9. Update initramfs

i. sudo update-initramfs -u

10. Add new file system mount options to the fstab file for automatic mounting at boot

i. sudo nano /etc/fstab

```
/dev/md0    /mnt/md0    ext4    defaults    0    0
```

11.

The RAID1 configuration is now complete and should be automatically mounted when the Raspberry Pi boots up. The next steps are to set up pushbullet notifications for when one of the USB drives fail

12. First, create a Push Bullet account and obtain the API Key

- i. <https://www.pushbullet.com/>

13. Enter your API key into a file called `pushbullet` in the `.config` directory

- i. `nano ~/.config/pushbullet`
- ii. `PB_API_KEY=<api_key>`

14. Download and install Pushbullet

- i. `sudo apt install -y git`
- ii. `sudo git clone https://github.com/Red5d/pushbullet-bash /opt/pb/`
- iii. `cd /opt/pb/`
- iv. `sudo git submodule init && sudo git submodule update`
- v. `cd`
- vi. `sudo ln -s /opt/pb/pushbullet /usr/local/bin/pushbullet`
- vii. `pushbullet list`

Pushbullet can now be used in the command line

15. Next we download a script ([https://github.com/hunleyd/mdadm\\_notify](https://github.com/hunleyd/mdadm_notify)) that triggers when mdadm reports an event. This event is then pushed to all pushbullet devices on the account.

- i. `sudo git clone https://github.com/hunleyd/mdadm_notify /opt/mn/`
- ii. `sudo nano /opt/mn/mdadm_notify`
- iii. Edit the final line so it looks like this  
`echo "$msg" | pushbullet push all note $msg`
- iv. Edit the mdadm configuration file so that it uses the newly created program `sudo nano /etc/mdadm/mdadm.conf`
- v. Replace the MAILADDR line with the following: `PROGRAM /opt/mn/mdadm_notify`
- vi. To simulate a fault we will be using the following commands

```
mdadm --manage --set-faulty /dev/md0 /dev/sda
```

**SCREENSHOT** Congratulations you have now implemented Pushbullet notifications! Mdaemon will automatically use this script whenever it detects an event.

## Installing Docker

---

1. Ensure you are logged into the Raspberry Pi
  - i. `ssh pi@192.168.x.x`
2. Download the Docker convenience script from <https://get.docker.com>
  - i. `wget -O get-docker.sh https://get.docker.com`
3. Make the Docker script an executable and run it
  - i. `chmod +x get-docker.sh`
4. Run Docker script
  - i. `sudo sh get-docker.sh`
  - ii. **SCREENSHOT**
5. After the script is installed, add the user to the Docker group
  - i. `sudo usermod -aG docker pi`
  - ii. `sudo reboot now`
6. Upgrade the Docker install by running a standard apt update and upgrade
  - i. `sudo apt update -y && sudo apt upgrade -y`

## Creating Docker Image

---

### Install Packages

1. Create a centos:centos7 Docker image and run it
  - i. `docker image pull centos:centos7`
  - ii. `docker run -i -t centos:centos7`
2. Download the centos install script from the github repository.
  - i. `yum makecache && yum install -y wget`
  - ii. `wget -N https://github.com/s3668468/usap-a1/blob/master/centos7_install.sh`
3. Make the script an executable and run it
  - i. `chmod +x centos7_install.sh`

- ii. `./centos7_install.sh`
- 4. This script should install all necessary applications for the Docker image.  
Please refer to the installation script for correct methods.
- 5. Commit and save the Docker image
  - i. Press `CTRL+P` and `CTRL+Q` to exit the Docker image without closing it
  - ii. Obtain the Docker ID with `docker ps`
  - iii. `docker commit <containerid> <name:tag>`

## Make configuration changes

### Expose nginx

- 1. Open the saved Docker container with the following commands and run nginx
  - i. `docker run -i -t -p 80:80 <name:tag>`
  - ii. `nginx`
- 2. Connect to it from the client machine
  - i. **SCREENSHOT**

### Create new users

- 1. Create a new user called 'fred' who has access to root
  - i. `yum install -y sudo`
  - ii. `useradd fred`
  - iii. `usermod -aG wheel fred`  
Wheel is used as installing sudo does not create the privileged group
  - iv. `passwd fred`
  - v. Password is `usap`
- 2. Create a new user called 'user' who has access to root, Berryconda and has zsh set by default.
  - i. `useradd user -s /bin/zsh`
  - ii. `usermod -aG wheel user`
  - iii. `passwd user`
  - iv. Password is `usap`
  - v. `su user`

- vi. ZSH set up is required
  - a. Enter `1`
  - b. Enter `0`
- vii. Berryconda was symbolically link during the installation script from earlier. To test user `conda --version`

## Install OhMyZSH and Auto Suggestions

<https://github.com/ohmyzsh/ohmyzsh> <https://github.com/zsh-users/zsh-autosuggestions>

### 1. Log into 'fred' and install OhMyZsh

- i. `su fred`
- ii. `sh -c "$(wget -O- https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"`
- iii. Enter `y`
- iv. `git clone https://github.com/zsh-users/zsh-autosuggestions ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-autosuggestions`
- v. Add the following line to the plugins in `.zshrc` `nano ~/.zshrc`  
`plugins=(zsh-autosuggestions)`
- vi. Add a custom theme by editing the `ZSH_THEME` variable  
`ZSH_THEME="afowler"`

### 2. Repeat these steps for 'user'

#### i. SCREENSHOT

## Enable SSH with redirected port

1. Exit the Docker image with `CTRL+P` and `CTRL+Q`
2. Commit and save the Docker image
  - i. `docker commit <containerid> <name:tag>`
  - ii. `docker stop <containerid>`
3. Start the Docker image with more port argument
  - i. `docker run -i -t -p 80:80 -p 1234:22 <name:tag>`
4. Create an ssh key and run the process to test connection
  - i. `ssh-keygen -A`



ii. sshd requires an absolute path `/usr/sbin/sshd -D`

### iii. SCREENSHOT

## Enable x11 over SSH and Disable Root Access

1. Edit the SSH configuration file and enable X11 and disable root access

- i. `nano /etc/ssh/sshd_config`
- ii. Uncomment `PermitRootLogin` and set it to `no`
- iii. Uncomment `X11Forwarding` yes , `X11DisplayOffset` 10 and `X11UseLocalhost` yes
- iv. Start sshd with `/sbin/sshd`

2. Install xeyes on the client machine

- i. `sudo apt install x11-apps`
- ii. `ssh -X user@raspberrypi.local -p 1234`

## Add zsh to the list of available shells on the system

1. Adding zsh to the list of shells can be done with one command

- i. `echo "/bin/zsh" >> /etc/shells`

## Write a script to start nginx and sshd

1. Create a script file and add in commands to start nginx and sshd

- i. `cd && nano nginx_sshd.sh`

ii. Enter the folling text:

```
#!/bin/bash
/sbin/nginx
/sbin/sshd
```

- iii. Make the script and executable and run it `chmod +x nginx_sshd.sh` `sh nginx_sshd.sh`

## Configure Git

1. Configure git user name

- i. `git config --global user.name "Cameron Tavelli"`

2. Configure git user email

```
i. git config --global user.email  
"s3668468@student.rmit.edu.au"
```

### Check versions for every program

1. Run the provided script to automatically check

- i. `wget -N https://github.com/s3668468/usap-a1/blob/master/version.sh`
- ii. `chmod +x version.sh`
- iii. `sh version.sh`

### Commit Docker image

<https://docs.docker.com/engine/reference/commandline/push/>

1. Create an account on <https://hub.docker.com>

2. Log into the account in terminal

- i. `docker login`
- ii. Username: <username>
- iii. Password: <password>

3. Check and tag the Docker image

- i. `docker images`
- ii. `docker tag centos7:latest s3668468/centos7:latest`
- iii. `docker push s3668468/centos7:latest`

4. Check the Docker image on the repository and set it to private

- i. Link to Docker image <https://hub.docker.com/repository/docker/s3668468/centos7/general>

<https://github.com/s3668468/usap-a1>

### References

- [1]"Download Raspberry Pi OS for Raspberry Pi", Raspberry Pi, 2020. [Online]. Available: <https://www.raspberrypi.org/downloads/raspberry-pi-os/>
- [2]"Pushbullet - Your devices working better together", Pushbullet.com, 2020. [Online]. Available: <https://www.pushbullet.com/>
- [3]Get.docker.com, 2020. [Online]. Available: <https://get.docker.com>

- [4]"ohmyzsh/ohmyzsh", GitHub, 2020. [Online]. Available: <https://github.com/ohmyzsh/ohmyzsh>
- [5]"zsh-users/zsh-autosuggestions", GitHub, 2020. [Online]. Available: <https://github.com/zsh-users/zsh-autosuggestions>
- [6]"docker push", Docker Documentation, 2020. [Online]. Available: <https://docs.docker.com/engine/reference/commandline/push/>
- [7]"CentOS Repositories", 2020. [Online]. Available: <https://centos.pkgs.org/>
- [8]"SpecialInterestGroup/AltArch/armhfp - CentOS Wiki", Wiki.centos.org, 2020. [Online]. Available: <https://wiki.centos.org/action/show/SpecialInterestGroup/AltArch/armhfp>
- [9]"jjhelmus/berryconda", GitHub, 2020. [Online]. Available: <https://github.com/jjhelmus/berryconda>
- [10]"mozilla/rr", GitHub, 2020. [Online]. Available: <https://github.com/mozilla/rr>