Linux Essentials - Final Challenge

This final challenge consists of setting up a Raspberry Pi implementing different requirements. All of the required commands and actions are discussed in the course materials or have been part of a challenge.

This challenge has to be completed within a time span of 4 hours in the LAB. To prepare for this challenge the following should be a good guide:

- Good understanding of the course
- Complete all the course challenges and document them well
- Revise the preparation checklist and prepare the topics mentioned

The full LAB challenge is to be completed on a HEADless system. In other words no display, mouse or keyboard attached to the Raspberry Pi.

Make sure to bring your Raspberry Pi kit. The teacher will provide clean SD-cards. Make sure you can flash the Micro SD-cards using your laptop and RPi imager.

Make sure that you know the MAC address of your wired Ethernet interface !

The actual assignment will consist of:

- a number of primary requirements which will need to be accomplished in order to pass the challenge.
- a number of secondary requirements which are graded separately

The requirements will be clearly marked as primary or secondary.

Grading will be fully automated. In other words if the primary requirements are not met, this process will fail and no grade can be achieved. Requirements which are not fully operational will also be seen as not implemented.

Preparation Checklist

- Flashing an RPi image to an SD-card
- Setup a HEADless system and enabling SSH
- Determining the IP address based on a MAC address (`nmap`, `dhcpdump`, `tcpdump`, ...)
- Configure static IP addresses for network interfaces
- Setup a hostname
- Creating accounts
- Creating groups and adding accounts to that group
- Locking accounts
- Setup SSH authentication using keys
- Setup default shell for accounts
- Installing the ZSH shell
- Connecting to a Wireless network via command line
- Configuring UFW firewall
- Install an Apache Web server and modify standard `index.html` file a bit
- Install docker and docker-compose

- Run a basic docker image as a service when the docker-compose file is provided
- Setup a cron job that posts a message to a url using the `curl` command line tool (can be scripted)
- Creating directories and files, setting up permissions and ownership on both
- Cloning git repo's
- Installing basic packages using `apt`
- Setup an MQTT broker
- Create regular backups of a directory (`tar.gz`) using cron