The project starts requiring us to set up a virtual machine and install an operating system, either Debian (recommended for beginners) or CentOS.

CentOS x Debian

I have installed VirtualBox and created a machine that will run Debian 64bits. I chose Debian since it seemed easy to configure, minimally intuitive in the basic install compared to CentOS.

[In this blog](https://www.openlogic.com/blog/centos-vs-debian), i found the following text: *"CentOS is a free downstream rebuild of the commercial Red Hat Enterprise Linux distribution where, in contrast, Debian is the free upstream distribution that is the base for other distributions, including the Ubuntu Linux distribution*".

By looking further for the difference between downstream rebuild and upstream distribution. According to the [Red Hat Blog](https://www.redhat.com/en/blog/what-open-source-upstream), the "upstream in open source is the source repository and project where contributions happen and releases are made". In upstream distributions, usually it regards the precursor to other projects and products and the example it gives is Linux kernel, which is a base for many other Linux distributions. Upstreams are important because that's where the source contribution comes and is the focal point where collaborators do the work. The upstream is also a fixed place where developers can report bugs and security vulnerabilities. On the other hand, downstream builds are usually certified products that meet the market's needs.

One feature that gives Debian an advantage is that of major version upgrades between stable versions, making it evolve with the area over the years, including support. Finally, Ubuntu, which is a distro I am familiar with, derived from debian and that is why I chose to install debian in this project.

Unix

Unix is a multitasking multiuser operating system and serves as a basis of most servers in the internet. It works basically in three levels:

* Kernel - is the system core, not visible to the user, and responsible for the internal functions of the system;
* Shell - is the interface that connects the OS' user and the core (kernel). The first process, always automatically executed upon start, is the shell;
* Applications - most os unix commands.

Each task or command in unix is a process and receives an identification number.

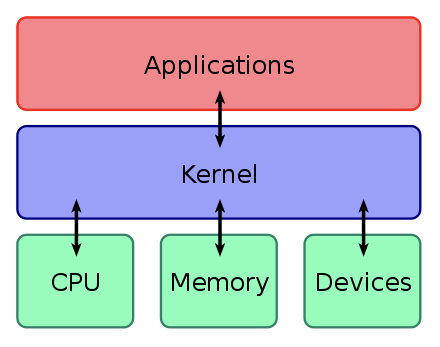
Kernel

The kernel is a computer program at the core of a computer's operating system and has complete control over everything in the system. It is responsible for connecting the hardware (physical parts) and the software (logic) of the computer. Its main objective is to manage the computer and allow applications to be executed and to use the computer's physical resources.

It is started upon turning on the computer and it detects all hardware devices and loads the operating system, when it starts to manage the processes, the files, the memory and peripheral devices.

The critical code of the kernel is usually loaded into a separate area of memory, which is protected from access by application software or other, less critical parts of the operating system. The kernel performs its tasks, such as running processes, managing hardware devices such as the hard disk, and handling interrupts, in this protected kernel space. In contrast, application programs like browsers, word processors, or audio or video players use a separate area of memory, user space. This separation prevents user data and kernel data from interfering with each other and causing instability and slowness, as well as preventing malfunctioning applications from affecting other applications or crashing the entire operating system.

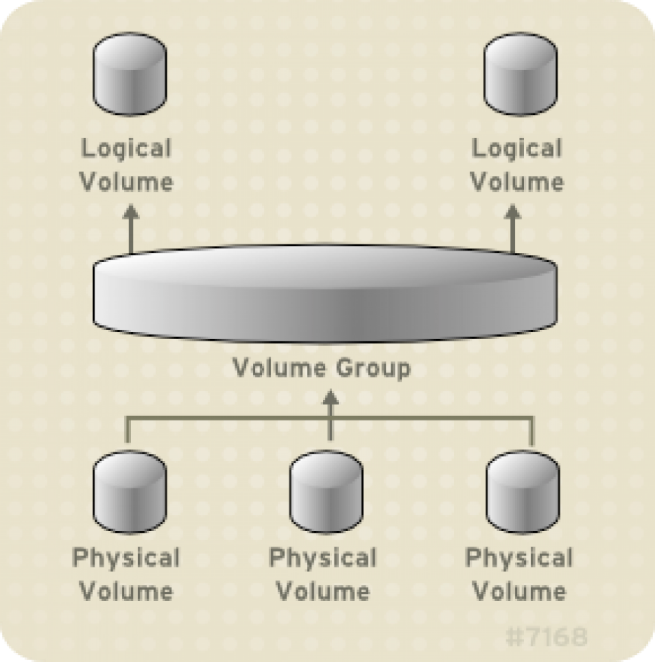
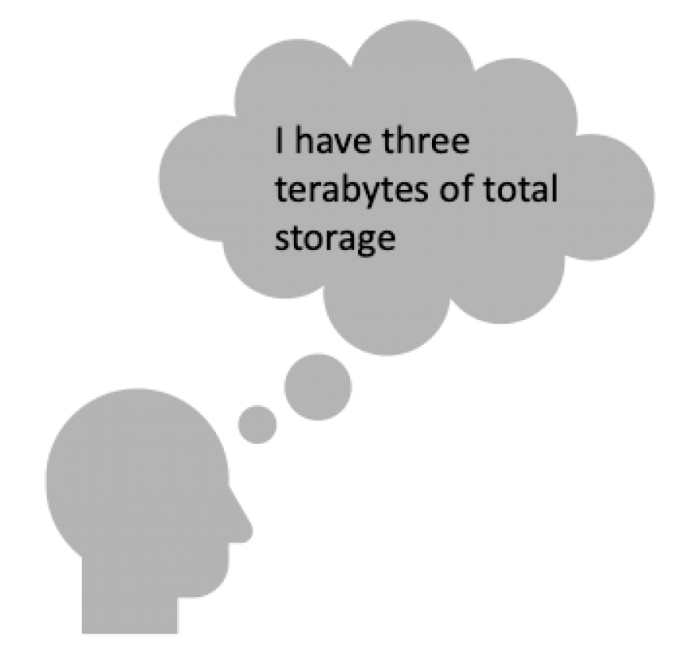
The kernel's interface is a low-level abstraction layer. When a process requests a service from the kernel, it must invoke a system call, usually through a wrapper function.



LVM – Logical Volume Manager

LVM allows for very flexible disk space management. It provides features like the ability to add disk space to a logical volume and its file system while that filesystem is mounted and active and it allows for the collection of multiple physical hard drives and partitions into a single volume group which can then be divided into logical volumes.

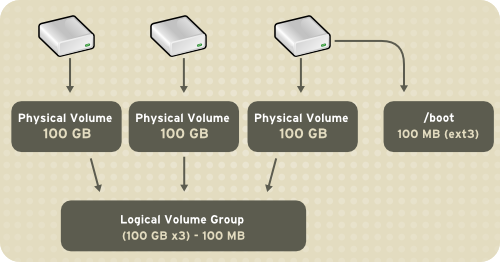
Traditional storage capacity is based on individual disk capacity. LVM uses a different concept. Storage space is managed by combining or pooling the capacity of the available drives. With traditional storage, three 1 TB disks are handled individually. With LVM, those same three disks are considered to be 3 TB of aggregated storage capacity. This is accomplished by designating the storage disks as Physical Volumes (PV), or storage capacity usable by LVM. The PVs are then added to one or more Volume Groups (VGs). The VGs are carved into one or more Logical Volumes (LVs), which then are treated as traditional partitions.



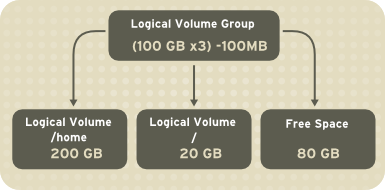
<https://www.redhat.com/sysadmin/lvm-vs-partitioning>

With LVM, a hard drive or set of hard drives is allocated to one or more *physical volumes*. LVM physical volumes can be placed on other block devices which might span two or more disks.

The physical volumes are combined into logical volumes, with the exception of the /boot partition. The /boot partition cannot be on a logical volume group because the boot loader cannot read it. If the root (/) partition is on a logical volume, create a separate /boot partition which is not a part of a volume group.



The volume groups can be divided into *logical volumes*, which are assigned mount points, such as /home and / and file system types, such as ext2 or ext3. When "partitions" reach their full capacity, free space from the volume group can be added to the logical volume to increase the size of the partition. When a new hard drive is added to the system, it can be added to the volume group, and partitions that are logical volumes can be increased in size.



<https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/ch-lvm>

* Physical Volume (PV) - Each Physical Volume can be a disk partition, whole disk, meta-device, or a loopback file. Use the command pvcreate to initialize storage for use by LVM. Initializing a block device as physical volume places a label at the start of the device.
* Volume Group (VG) - A Volume Group gathers together a collection of Logical Volumes and Physical Volumes into one administrative unit. Volume group is divided into fixed size physical extents. The command vgcreate creates a new volume group using the block special device Physical Volume path previously configured for LVM with pvcreate. VGs are made up of PVs, which in turn are made up of physical extents (PEs).
* Logical volume (LV) - is the conceptual equivalent of a disk partition in a non-LVM system. Logical volumes are block devices which are created from the physical extents present in the same volume group. You can use command lvcreate to create a logical volume in an existing volume group.

<https://www.thegeekdiary.com/redhat-centos-a-beginners-guide-to-lvm-logical-volume-manager/>

VM configuration

While configuring the VM, I have set up a RAM base of 1GB and a total space of 13GB, which should be enough for the whole project.

During the installation, it is already possible to set up one of the project requirements: have two encrypted partitions using LVM. This will give us one encrypted partition. All beginning configuration is done during the installation.

Next, we need to install sudo. To enter the root, type su, then do apt-get install sudo.

Sudo

Sudo is a program for Unix-like computer operating systems that allows users to run programs with the security privileges of another user, by default the superuser.

The project demands us to also install AppArmor in Debian.

Problems with adding the user to group sudo. None of the following commands worked:

* $ (sudo) adduser newuser sudo
* $ (sudo) usermod -aG sudo username

-aG: append to Group

The only one that worked was:

* $ sudo gpasswd -a username sudo

Adding user XXX to group sudo

To verify whether the user was successfully added to *sudo* group via getent group sudo.

$ getent group sudo

Then, **reboot** for changes to take effect, then login and verify sudopowers via sudo -v.

<https://www.digitalocean.com/community/tutorials/how-to-edit-the-sudoers-file-pt>

After giving my user sudo access, I have removed the unnecessary groups with [commands for administering groups on linux](https://man7.org/linux/man-pages/man1/gpasswd.1.html).

I also update the sudo rules in order to

* Prevent the program from forwarding the user to a remote window (through ssh) with a graphical interface (X11 forwarding);
* Configure port 4242;
* Inhibit remote root login.

Aptitude vs Apt

Aptitude - It is an interface to Apt that allows a user to interactively search for a package and install or remove it. It can emulate most command line arguments. It is considered high-level as apt is considered low level, but can be used by high-level package managers.

[Debian website](https://www.debian.org/doc/manuals/aptitude/rn01re01.en.html):

Aptitude is a text-based interface to the Debian GNU/Linux package system. It allows the user to view the list of packages and to perform package management tasks such as installing, upgrading, and removing packages. Actions may be performed from a visual interface or from the command-line.

Apt - Advanced Packaging Tool - free and open source software that handles software installation and removal, designed for Debian’s packages. Works in command lines and installs dependencies for a specific package with it. Highly flexible to configure.

Source: <https://www.tecmint.com/difference-between-apt-and-aptitude/>

[Debian website](https://wiki.debian.org/Apt):

Advanced Package Tool (or APT), the main command-line package manager for Debian and its derivatives. It provides command-line tools for searching, managing and querying information about packages, as well as low-level access to all features provided by the libapt-pkg and libapt-inst libraries which higher-level package managers can depend upon.

Apparmor

AppArmor ("Application Armor") is a Linux kernel security module that allows the system administrator to restrict programs' capabilities with per-program profiles. Profiles can allow capabilities like network access, raw socket access, and the permission to read, write, or execute files on matching paths. AppArmor supplements the traditional Unix discretionary access control (DAC) model by providing mandatory access control (MAC).

In addition to manually creating profiles, AppArmor includes a learning mode, in which profile violations are logged, but not prevented. This log can then be used for generating an AppArmor profile, based on the program's typical behavior.

AppArmor is implemented using the Linux Security Modules (LSM) kernel interface.

<https://wiki.debian.org/AppArmor/HowToUse>

SELinux

Security-Enhanced Linux (SELinux) is a security architecture for Linux systems that allows admins to have more control over who can or can't access the system.

Quando uma aplicação ou processo, também conhecidos como entidade, solicita acesso a um objeto, por exemplo, um arquivo, o SELinux executa uma verificação com um cache de vetor de acesso (AVC), local onde as permissões para entidades e objetos ficam armazenadas. Se a permissão for negada, a mensagem "avc: denied" aparecerá em /var/log.messages.

Há várias maneiras de configurar o SELinux para proteger seu sistema. As mais comuns são política direcionada (targeted policy) ou segurança multinível (MLS), geralmente mais complicada e só aplicada em organizações governamentais. É possível saber o que o sistema deveria estar executando olhando o arquivo /etc/sysconfig/selinux. O arquivo terá uma seção que exibe se o SELinux está no modo de permissão, modo de imposição ou desativado e qual política deveria ser carregada.

Controle de acesso opcional (DAC) x controle de acesso obrigatório (MAC)

Tradicionalmente, os sistemas Linux e UNIX usam o DAC. O SELinux é um exemplo de sistema MAC para o Linux.

No DAC, arquivos e processos têm proprietários. Um usuário, um grupo ou qualquer outra pessoa pode ser o proprietário de um arquivo. Os usuários podem mudar as permissões em seus próprios arquivos.

O usuário raiz tem controle de acesso completo com um sistema DAC. Se você tiver acesso irrestrito, poderá entrar nos arquivos de qualquer outro usuário ou fazer o que quiser no sistema.

No entanto, em sistemas MAC, como o SELinux, há uma política definida administrativamente para o acesso. Mesmo que as configurações DAC no seu diretório home estejam diferentes, uma política do SELinux criada para prevenir que outro usuário ou processo acesse o diretório irá proteger o sistema.

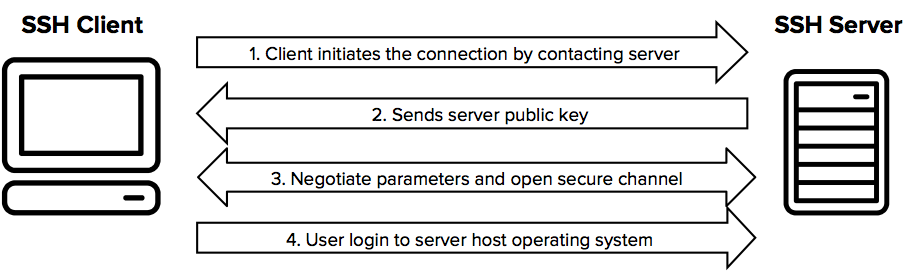
As políticas do SELinux permitem que você seja específico e contemple um grande número de processos. Você pode fazer alterações com o SELinux para limitar o acesso entre usuários, arquivos, diretórios e muito mais.

<https://www.redhat.com/pt-br/topics/linux/what-is-selinux>

SSH / Secure Shell

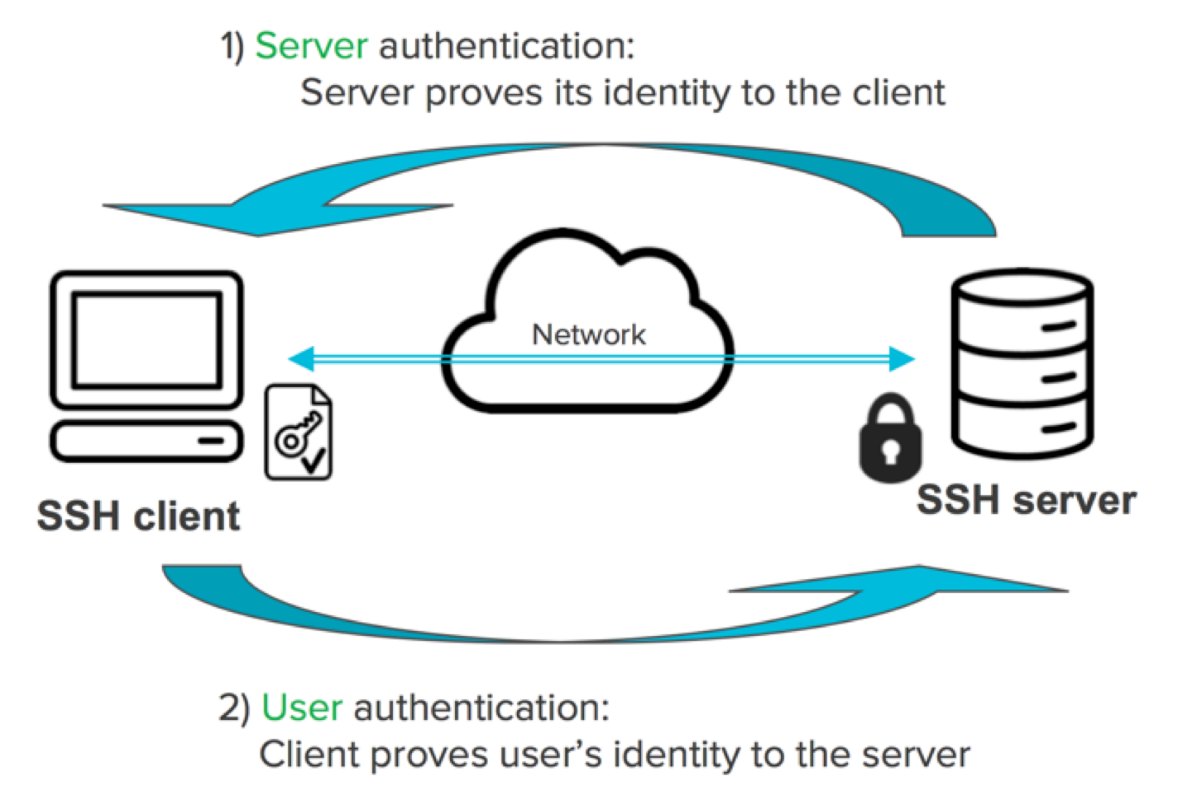
SSH is a software package that enables secure system administration and file transfers over insecure networks. It is used in nearly every data center and in every large enterprise.

The SSH protocol uses encryption to secure the connection between a client and a server. All user authentication, commands, output, and file transfers are encrypted to protect against attacks in the network.



The public key authentication method is primarily used for automation and sometimes by system administrators for single sign-on. Essentially, some session-specific data is signed using the private identity key. The signature is then sent to the server that checks if the key used for signing is configured as an authorized key. The server then verifies the digital signature using the public key in the authorized key. The identity key is never sent to the server.

The essential thing in public key authentication is that it allows one server to access another server without having to type in a password. This powerful feature is why it is so widely used for file transfers (using the SFTP protocol) and configuration management. It is also commonly used by system administrators for single sign-on.



<https://www.ssh.com/academy/ssh/protocol>

**The use of SSH will be tested during the defense by setting up a new account.**

To connect to the ssh server, write in the terminal: ssh julcarva@IPADDRESS –p4242

UFW - Uncomplicated Firewall

[PAM (Pluggable Authentication Modules)](https://www.debian.org/doc/manuals/securing-debian-manual/ch04s11.en.html) allows system administrators to choose how applications authenticate users.

Each application with PAM support provides a configuration file in /etc/pam.d/ which can be used to modify its behavior:

* what backend is used for authentication.
* what backend is used for sessions.
* how do password checks behave.

PAM offers you the possibility to go through several authentication steps at once, without the user's knowledge.

To get your IP address for configuring PAM, use the following code:

* The hostname command is actually used to display the hostname of a system. However, we can also use it to display the IP address of our system. To do so, open the Terminal and type hostname followed by –I character as shown below:
  + $ hostname –I



Then, to configure the port, follow the commands:

* sudo ufw enable
* sudo ufw allow <port>/<optional: protocol>

Sources for this section:

<https://wiki.debian.org/Uncomplicated%20Firewall%20%28ufw%29>

<https://help.ubuntu.com/community/UFW>

<https://www.tecmint.com/setup-ufw-firewall-on-ubuntu-and-debian/>

This stage allows to configure both TCP and UDP protocols.

* The **TCP/IP** (**Transmission Control Protocol)** belongs to the Internet protocol suite. TCP is connection-oriented, and a connection between client and server is established before data can be sent. The server must be listening (passive open) for connection requests from clients before a connection is established. It is generally more reliable than UDP.
* The **UDP (User Datagram Protocol)** also belongs to the Internet protocol suite. With UDP, computer applications can send messages, in this case referred to as datagrams, to other hosts on an Internet Protocol (IP) network. Prior communications are not required in order to set up communication channels or data paths.

Strong password policy

The subject requires the following:

* Your password has to expire every 30 days.
* The minimum number of days allowed before the modification of a password will

be set to 2.

* The user has to receive a warning message 7 days before their password expires.

For the subjects above, I followed these steps:

* sudo vi /etc/login.defs

[Libpam-pwquality](https://linux.die.net/man/8/pam_pwquality)

This module is used to provide some plug-in strength-checking for passwords. Pam\_cracklib is also an alternative, but this one is newer and with extra checks that will be helpful for this project, such as:

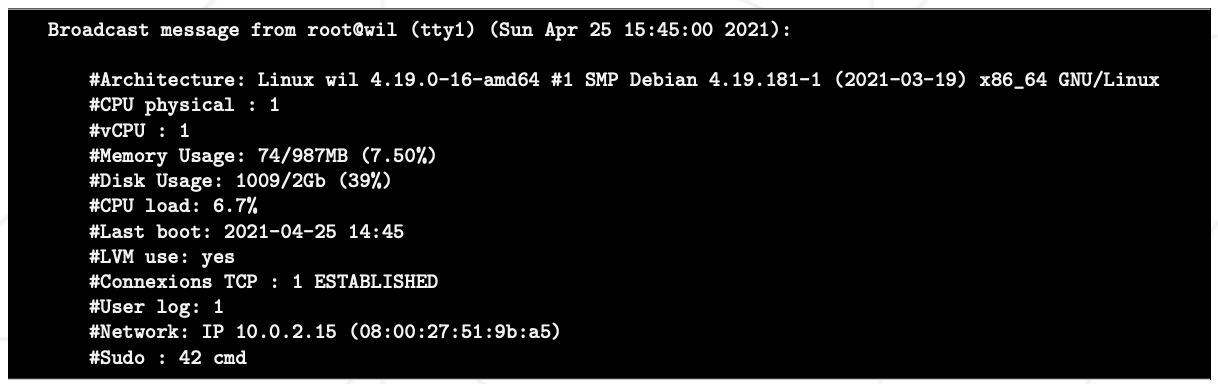
* Minimum length of 10 characters;
* Maximum repetition of characters 3 (Ex: no aaaa password);
* Dcredit = (N < 0) This is the minimum number of digits that must be met for a new password.
* UCredit = (N < 0) This is the minimum number of uppercase letters that must be met for a new password.
* Difok = This argument will change the default of 5 for the number of changes in the new password from the old password. Therefore, the new password has to have a minimum of 7 changes when compared to the old password;
* usercheck = whether to check if the password contains the user name in some form (enabled if the value is not 0)
* Enforce\_for\_root - The module will return an error on failed check even if the user changing the password is root. This option is off by default which means that just the message about the failed check is printed but root can change the password anyway.

After setting up the conditions for password strength, the subject requires us to implement them on our own and the root users. Given that I have just created a user for this project with a password that meets all the conditions demanded (Julia101010), I decided to implement the changes manually through the command [chage](https://geek-university.com/linux/manage-passwords/).

* Chage -m sets the minimum number of days between password changes (to 2);
* Chage -M changes the number of days the password is valid (to 30)
* Chage -W sets the number of days before account expiration that the system will warn the user (to 7).

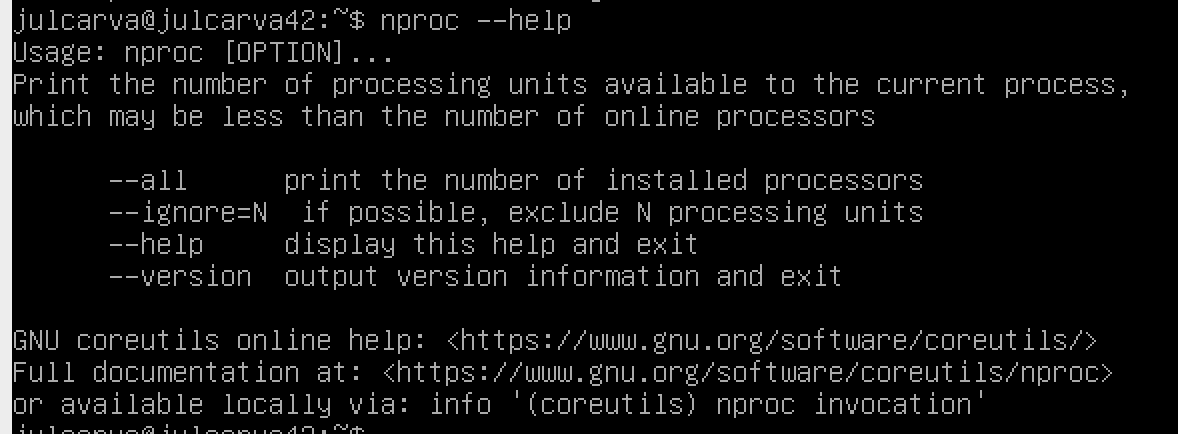
Monitoring.sh

Finally, we need to code a script that will show information on our VM like this:

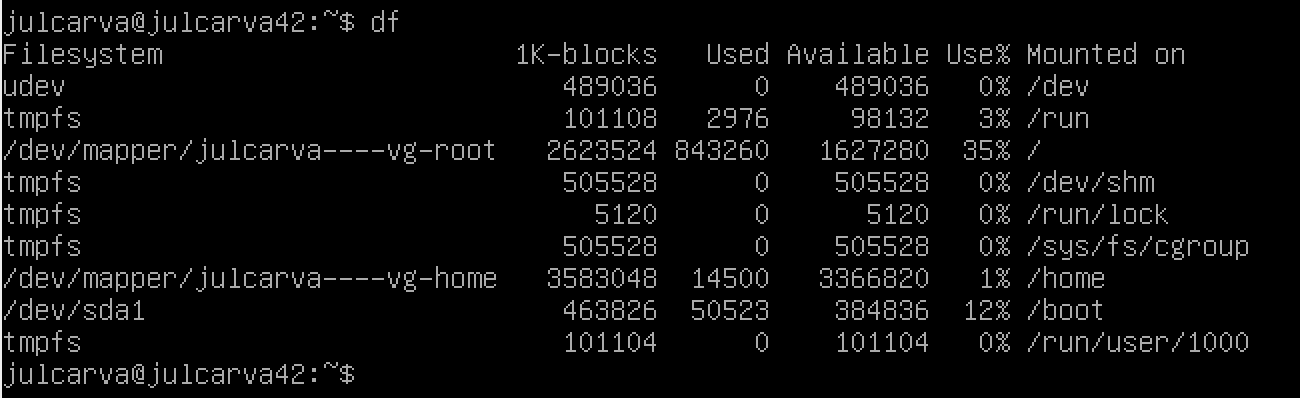


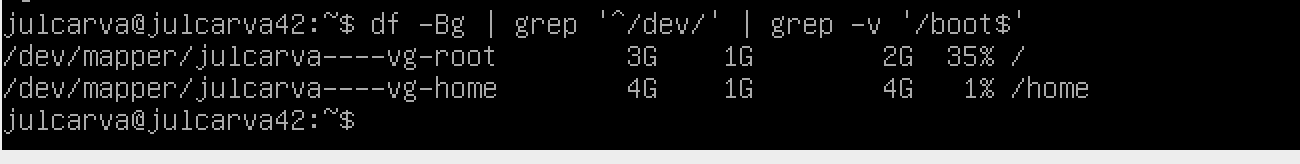
Source: <https://linuxize.com/post/uname-command-in-linux/>

* uname -a: shows all system information. When the -a option is used, uname behaves the same as if the -snrvmo options have been given and will display, respectively, kernel name, hostname, kernel release, kernel version, machine’s hardware name, and operating system.
* Check CPU Physical: cat /proc/cpuinfo to find the number of physical CPU cores, then grep the physical id, sort eliminating duplicates (-u), counting how many lines it will show (one per physical cpu shown) (wc -l);
* Virtual processors: I saw some people using nproc to get the virtual processors. I chose to use another approach after seeing this below. Note that the nproc command may exclude some online processors. To be safe, I chose to grep the processor directly and count the lines it has.

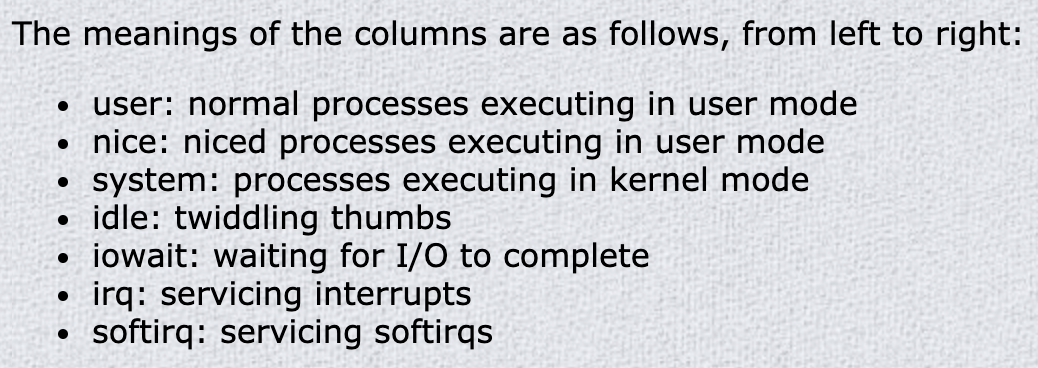


* Current available RAM on server and use rate as percentage:
  + get the memory information with the free -m command, then use the [awk](https://www.geeksforgeeks.org/awk-command-unixlinux-examples/) command to print it. We store information in variables starting with the $ sign. So $1 is “Mem:”, $2 is the value of total available ram for later. The numbers follow the order of variables shown.
  + using the same commands we get the used ram ($3).
  + Finally, we create a variable with the division of the total by the used and further divide it by 100 to get the %.
* Current available memory and its use rate as percentage:
  + The subject itself translates this as disk usage. The [df command](https://www.geeksforgeeks.org/df-command-in-linux-with-examples/) serves this purpose. The flags -B+(k,m,g,t…) resizes the display according to what we want. Since the example demands to show the data in Gb, we will use -Bg for the total size available in dev and all other than boot. We store the values in the variables to make the percentage calculation later. The used disk is shown as MB, so the flag changes to -Bm





* The current utilization rate of your processors as a percentage: grep ‘cpu from [/proc/stat](https://www.linuxhowtos.org/System/procstat.htm), then we sum the second argument of both $2, which will result in the total free space.



Other useful sources:

* <https://jjjaeu.tistory.com/33>
* <https://github-dotcom.gateway.web.tr/HEADLIGHTER/Born2BeRoot-42/blob/main/monitoring.sh>
* <https://www.notion.so/Born2BeRoot-6a10c2b772a74c20981c1c16b961b404>
* <https://giters.com/hanshazairi/42-born2beroot>
* <https://github-dotcom.gateway.web.tr/HEADLIGHTER/Born2BeRoot-42>
* <https://opensource.com/article/17/11/how-use-cron-linux>
* <https://www.howtogeek.com/101288/how-to-schedule-tasks-on-linux-an-introduction-to-crontab-files/>
* <https://www.cyberciti.biz/faq/how-do-i-add-jobs-to-cron-under-linux-or-unix-oses/>
* <https://githubmemory.com/repo/hanshazairi/42-born2beroot#setting-up-a-cron-job>
* <https://github.com/pgomez-a/born2beroot>
* <https://velog.io/@sehhong/born2beroot-cmd>
* <https://tbonelee.tistory.com/m/16?category=1000323>
* <https://stackoverflow.com/questions/8967902/why-do-you-need-to-put-bin-bash-at-the-beginning-of-a-script-file>
* <https://medium.com/@codingmaths/bin-bash-what-exactly-is-this-95fc8db817bf>