H3ABioNet

Pan African Bioinformatics Network for H3Africa

LINUX: GETTING STARTED

A HOW TO GUIDE FOR Linux NEWBIES

Developed by

The H3ABionet Pipelines and Computing Work Package, Computing Infrastructure project team

Prepared for the greater

H3ABioNet and H3Africa Consortium communities

Document Control

Date	Version	Notes
	1.0	Initial guide development

Project Members

Last Name	First Name	Institution	Country
Ghanmi	Nidhal	Institute of Pasteur , Tunis	Tunisia
Lukyamuzi	Edward	Uganda Virus Research Institute	Uganda
Maslamoney	Suresh	Computational Biology Division, University of Cape Town	South Africa
Meintjes	Ayton	Computational Biology Division, University of Cape Town	South Africa
Oloyede	Emmnauel	National Biotechnology Development Agency	Nigeria
Wamala	Timothy	Uganda Virus Research Institute	Uganda

Preamble

This Linux “Getting Started” guide is intended for individuals new to Linux or Windows system administrators wanting to setup a basic Linux server. Linux comes in many flavors, for the proposes of this getting started guide, screenshots and commands are based on the Ubuntu 18.04 and Ubuntu 20.04 Operating Systems (OS).

No previous knowledge of Linux is assumed or required to follow this guide. The reader is able to follow the step by step instructions below to setup a basic Linux server running the Ubuntu 18.04 or 20.04 OS. By the end of this guide, the reader would have gained the knowledge to install the Ubuntu 18.04 or 20.04 OS, configure a RAID or LVM setup for physical disk redundancy, enabled ssh remote access, setup a basic file server, create user accounts and assign permission to directories and files. The reader will also get hands on experience with navigating the Linux file system.

It is recommended that the reader first reads this guide in its entirety before following the step by step instructions.

This guide makes the following assumptions:

- The reader has a copy of the Ubuntu 18.04 or Ubuntu 20.04 iso and has written it to a installation media like a USB thumb drive or CD.
- If this guide is being followed while setting up a physical server or desktop, it assumes the system has been configured to boot from the install medium.
- If using a virtual environment, it assumes the reader is familiar with the virtual environment and is able to create a virtual machine to be used for this guide.

How to read this guide

- General text describing each section and commands are written in the Arial font, size 11. This text is to be read to understand which skills will be gained in the following section and to describe the command operation.
- When noting a variation to a default command or to note a point or warning, the default text will be highlighted in yellow.
- When noting a tip, for example, a command that can be run in multiple ways producing the same output, this text will be highlighted in green.
- When giving examples of actual expected output, this text will be highlighted with a grey background.
- When listing a command that is to be run by the reader to produce an output, the command will be in light blue italics.

Table of contents

H3ABioNet

Pan African Bioinformatics Network for H3Africa

Abbreviations

Abbreviation	Description
OS	Operating System
cli	Command line interface
gui	Graphical user interface
dhcp	Dynamic host configuration protocol
IP	Internet Protocol

1. UBUNTU 18.04 or 20.04 OS installation instruction

The below section will guide the reader through the installation of the Ubuntu 18.04 / Ubuntu 20.04 server OS. For those new to Linux servers, the Ubuntu server OS does not come standard with a Graphical User Interface (GUI). All commands are run via the Command Line Interface (cli).

NOTE:

- This tutorial was completed using a virtual machine created on the VirtualBox application using a single virtual hard drive of 10GB and 8GB RAM.

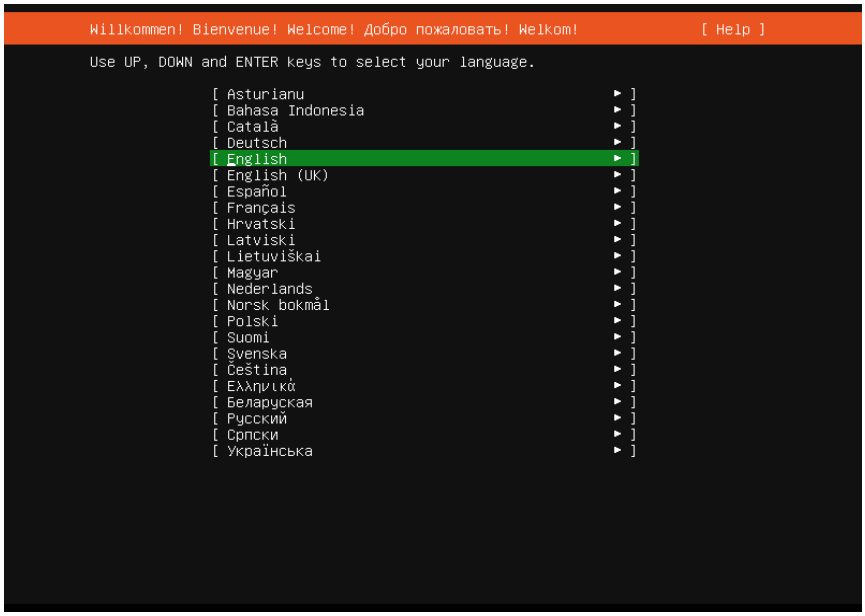
When installing the OS on a physical server, insert the Ubuntu installation medium into the server's USB port or DVD drive and power up the system. After booting from the Installer device , Select "Install Ubuntu server" from the installation menu.

If installing the OS into a virtual environment, mount the Ubuntu .iso file to your virtual DVD drive in your virtual environment.

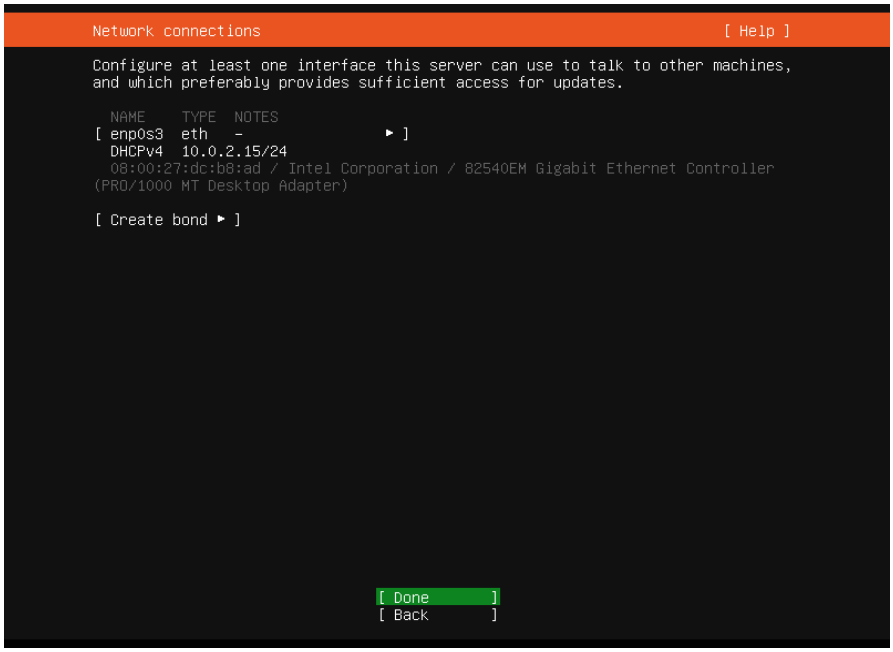
NOTE:

- If you do not see this installation menu, access the server's BIOS and ensure that "boot from removable media" has been selected as the first boot device.
- As an alternative, restart the server and as the POST (Power On Self-Test) runs, depress the function key to enter into the boot menu. On most system, this is either F8 or F2 keys.

Select your language, country and keyboard layout on the following screens



Configure your Network Card ports, if you have no preferences to use just hit enter to allow DHCP to automatically assign IP parameters.

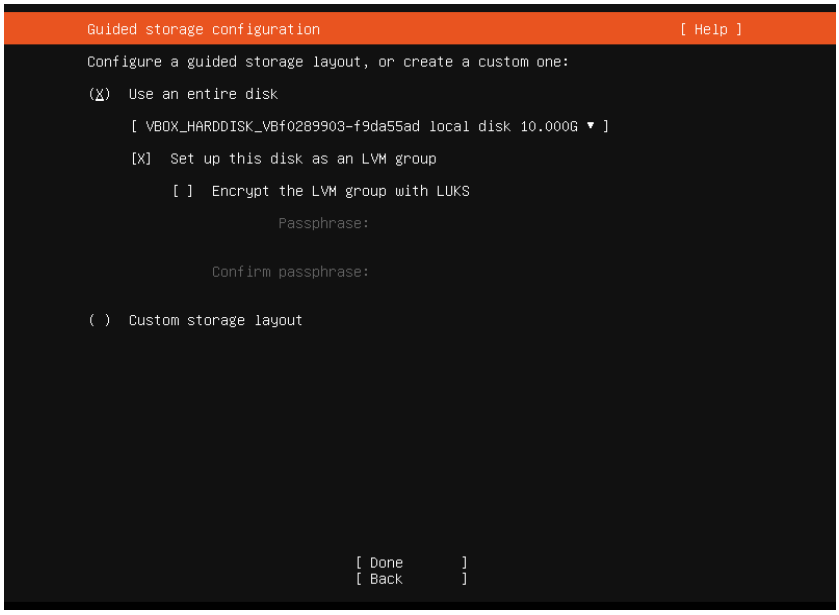


NOTE:

- Using the DHCP option assumes that there is a DHCP server already configured to supply this machine with an IP address. If no DHCP is available, you will still be able to proceed - the installation will take a few minutes to try and find an IP address. Once it fails to find a DHCP IP address, it will allow the reader to continue with the installation process. The IP address would need to be setup post OS installation.

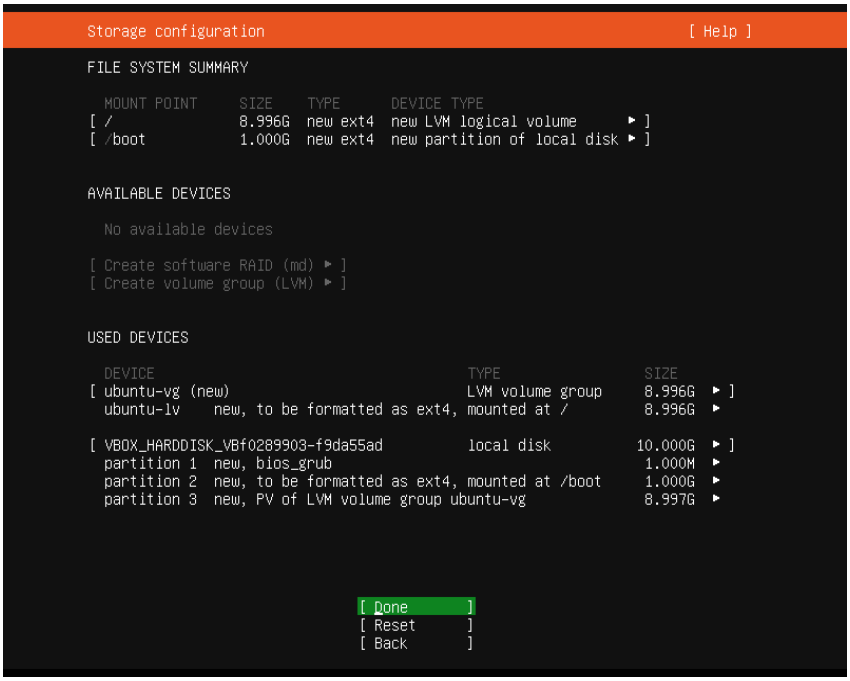
If your network requires a proxy to access the internet, supply the proxy details in the next screen , if not, just hit enter to proceed.

After selecting the proper mirror for your country, you will be asked to select the hard drive and asked how to partition it. One of the options on the hard drive selection screen is to enable LVM. Logical Volume Management (LVM) is a system of managing logical volumes. LVM is discussed later in this guide. If you are installing the OS to create a simple file server, it is easier to stick to the defaults. Click “Done” to continue.



The next screen will list all the partitions that will be created. You will be able to edit these partitions and their respective partition sizes should you prefer to do so. If you wish, you could at this stage configure a software based RAID configuration. RAID is discussed in the next section.

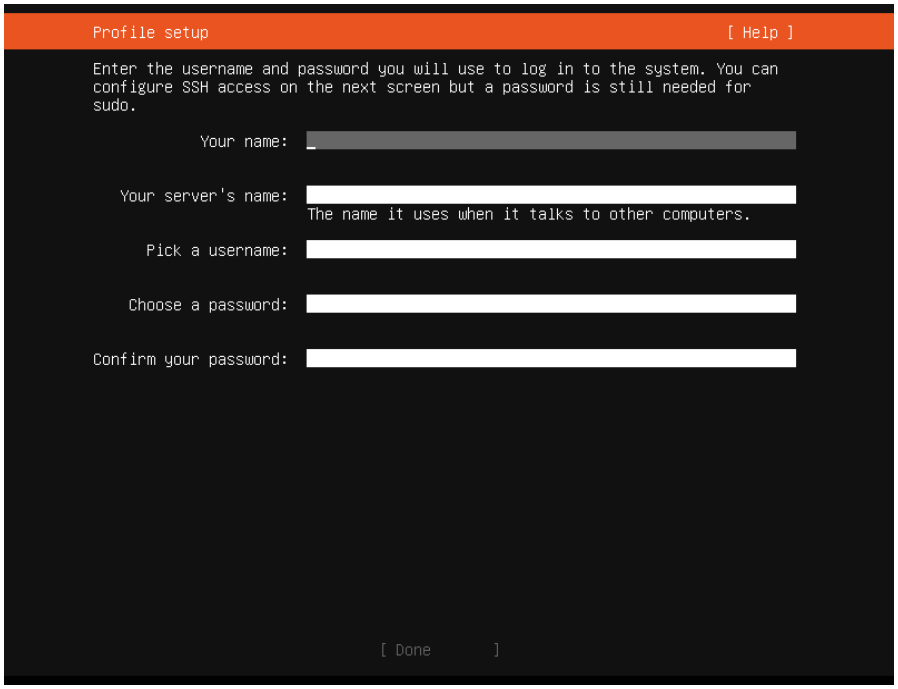
Unless you have a specific need, the default partition configuration would suffice. Review the disk partition summary and click on “Done” to proceed.



After confirming the file system and disk partitions, the reader is prompted to create a user account. This user will be the initial root or administrative account. Provide a username and password for the account, provide a host name for the new server.

TIP:

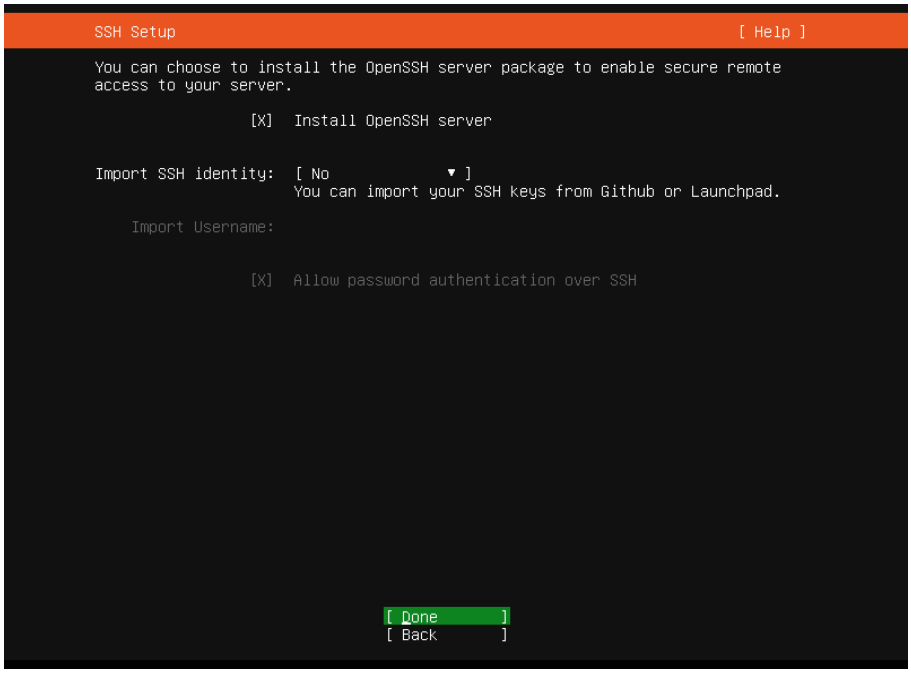
- Try not to exceed 8 characters for the servers name.
- Try to make the server name descriptive so that you can determine which server you are working on by the hostname in the terminal.



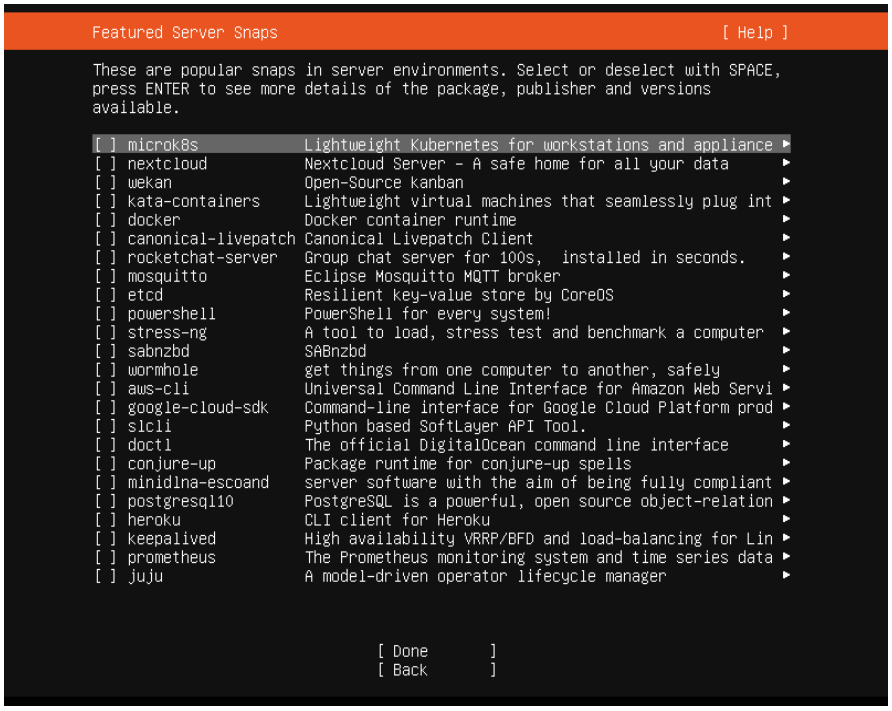
The next screen prompts the reader to enable the openSSH Server. SSH is an important function for a server as it is the primary method users will use to access this server remotely. Unless you have a specific reason not to enable SSH, select the option to install and enable SSH.

NOTE:

- If you did not opt to enable SSH during the OS installation, you can install the application at a later stage.

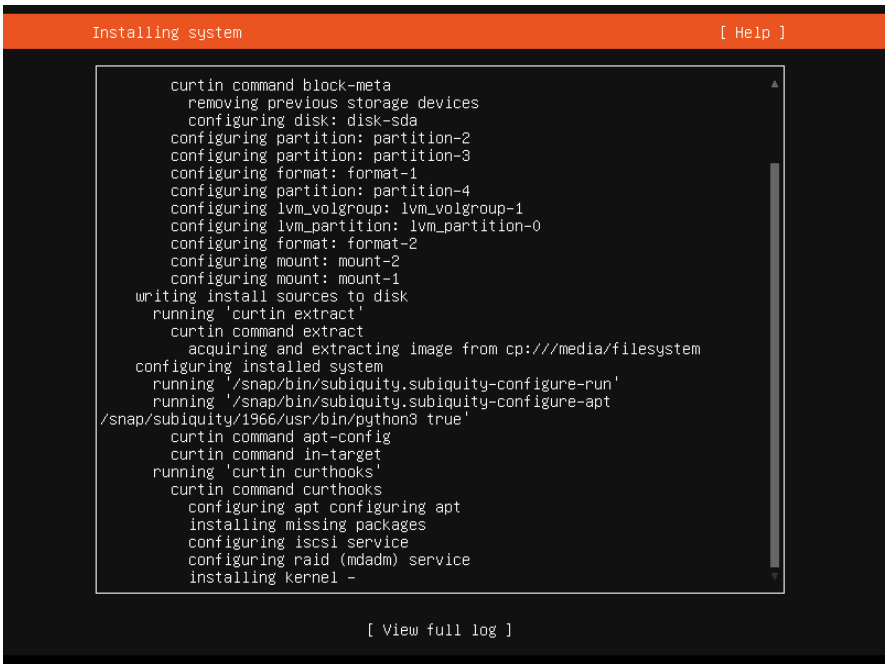


After clicking on “Done”, the reader is prompted with another screen listing additional software that might be useful, you can easily select them and they will be installed during system setup. If none of the additional software packages are needed, it is safe to skip this step by clicking on “Done” to proceed.



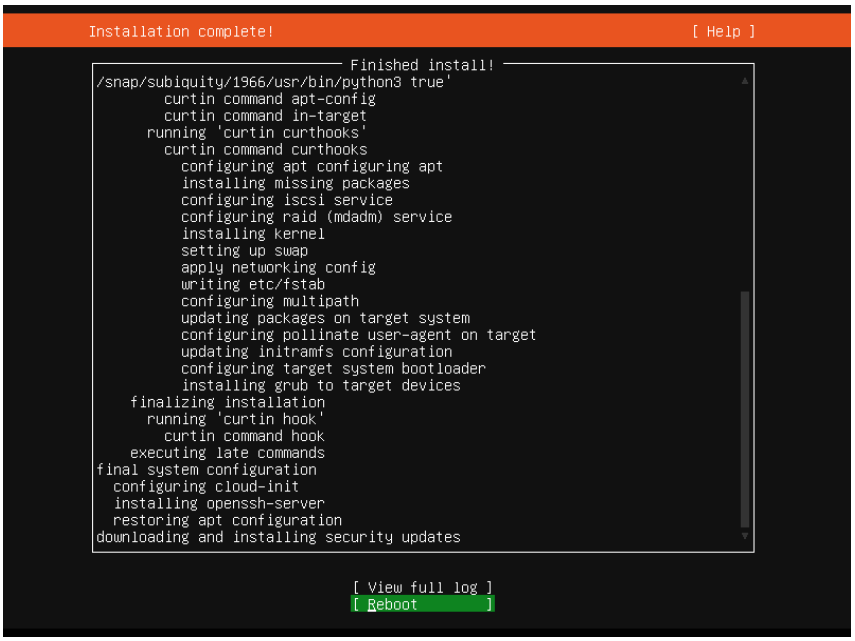
Clicking “Done” on the previous screen will begin the OS installation process using the choices made in previous steps. By default, once the OS has been installed, the installation process will attempt to update all software packages that were installed.

If an IP address was assigned and if the server has access to the internet, this step will proceed without issue. If no IP address was allocated, this step will fail but the OS installation will complete.

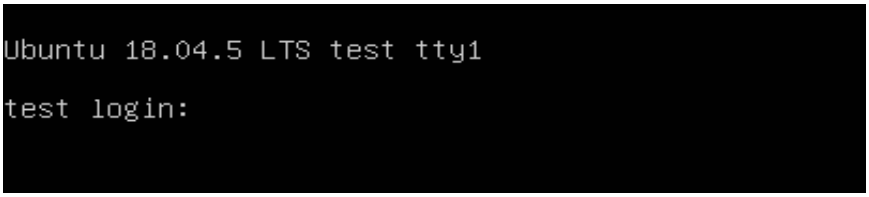


After setup and upgrades have run, the reader will see a “Reboot” option, remove the installation media

and hit enter to initiate the reboot.



Finally , the machine should reboot and a login screen will appear , type the login name and password you chose earlier in the installation process and start using your new server.



That's it, you now have a fresh installation of Ubuntu 18.04 or Ubuntu 20.4

2. Software RAID and LVM

2.1. What is RAID?

RAID (Redundant Array of Inexpensive Disks) is a way of introducing disk redundancy into a server or to expand the overall storage capacity. Essentially, RAID takes a number of physical hard disks and presents them as a unified logical storage space to the OS. The more common RAID configuration options are RAID 1 for the OS drives and RAID 5 or RAID 6 for disk storage. Depending on what RAID level you use, it will provide data redundancy, performance improvement, and reliability to your data.

An example would be configuring two physical hard drives in a server so that they mirror each other. This setup is referred to as a RAID 1 or "mirror" set. This configuration is typically used on the disks that house the OS and provides failover in the event one of the disks fail – the server will continue to operate while the faulty disk is replaced.

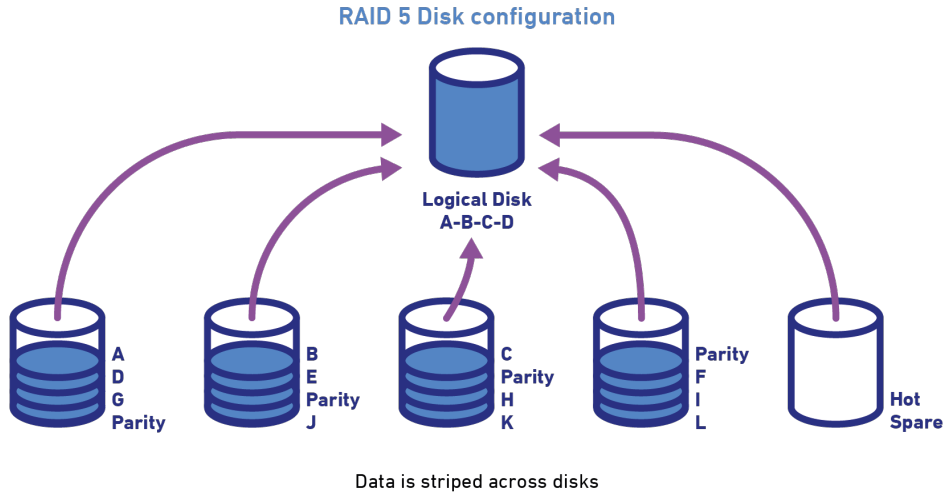
Likewise RAID 5 and 6 is used to expand the storage space of a single drive. RAID 5 and 6 require a minimum of 3 physical disks and are stitched together to present the combined storage of the 3 disks.

NOTE:

- RAID 5 uses one of the three disks while RAID 6 uses two physical disks to store parity information ultimately allowing the server to lose upto one or two disks respectively and continue to operate while the failed disk/s are replaced.
- RAID 6 is typically used when you have 5 disks or more.

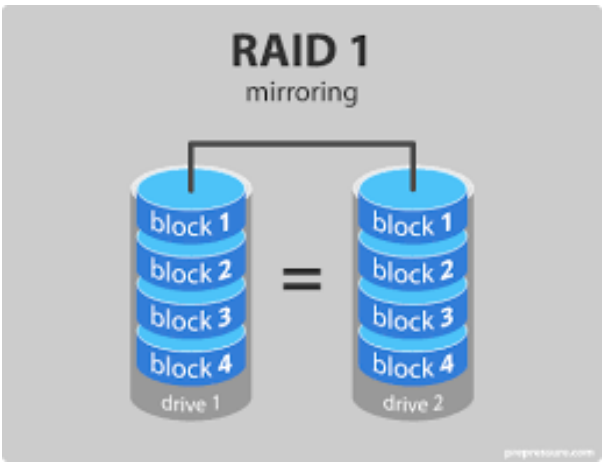
RAID can be managed by a physical RAID card in the server or via the OS, referred to as a "software RAID". Both options have their pros and cons, however for the purposes of this guide, we will only explore the software based RAID option.

A graphical representation of a RAID 5 configuration



Source: https://blog.leaseweb.com/2018/11/15/how-to-set-up-a-basic-storage-server/diagram-22x/

A graphical representation fo a RAID 1 configuration



Source: https://www.prepressure.com/library/technology/raid

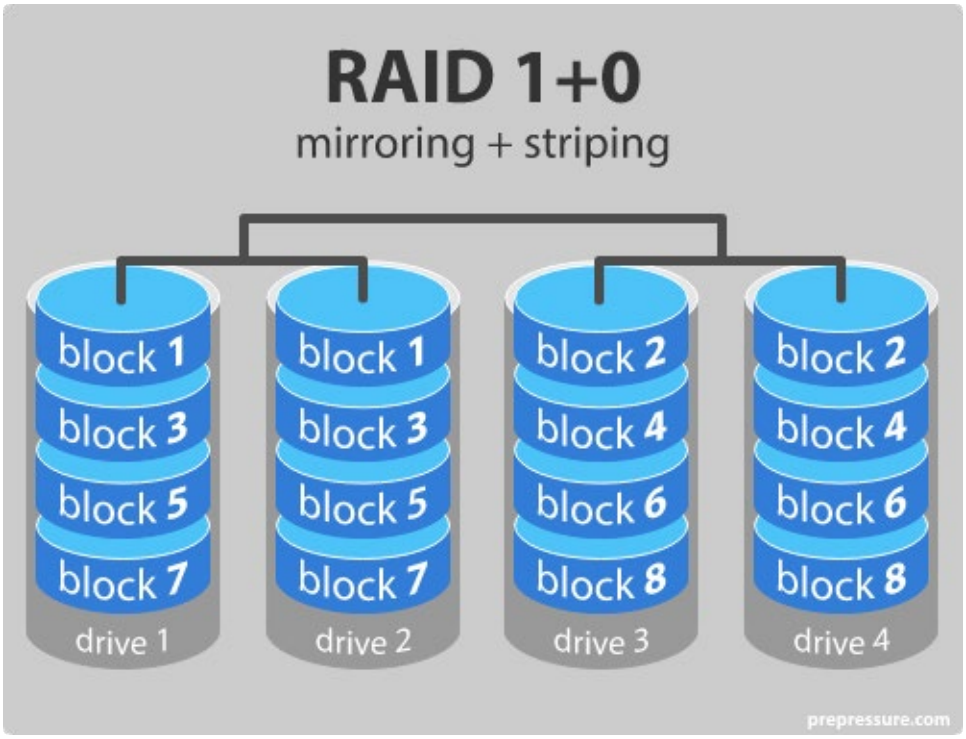
Linux Software RAID (often-called mdraid or MD/RAID) makes the use of RAID possible without a hardware RAID controller. For this purpose, the storage media used for this (hard disks, SSDs and so forth) are simply connected to the computer as individual drives, and then configured into a RAID via the software in the OS.

The current RAID drivers in Linux supports the following levels:

- Linear mode
  RAID-0 RAID-1 RAID-4 RAID-5

- Nested RAID levels
  RAID 10 which is a combination of RAID 1 + RAID 0

A graphical representation of a nested RAID 1 + 0 configuration



Source: https://www.prepressure.com/library/technology/raid

2.1.1. Installation of MDADM software RAID

Install mdadm.

$ sudo apt update

$ sudo apt upgrade -y

$ sudo apt install mdadm

Note:

- *mdadm* is used for administering pure software RAID using plain block devices: the underlying hardware does not provide any RAID logic, just a supply of disks. *mdadm* will work with any collection of block devices. Even if it's unusual. For example, one can thus make a RAID array from a collection of thumb drives.

With mdadm installed, we need to prepare the physical disk drives. If the device is being reused or repurposed from an existing array, erase any old RAID configuration information:

```
$ mdadm --misc --zero-superblock /dev/<drive>
```

or if a particular partition on a drive is to be deleted:

```
$ mdadm --misc --zero-superblock /dev/<partition>
```

After the physical disk drives have been presented to mdadm, we need to partition the devices

It is highly recommended to partition the disks to be used in the array. Since most RAID users are selecting disk drives larger than 2TB, GPT is required and recommended.

The next step is to build the array. Use mdadm to build the array. The following example shows building a 2-device RAID1 array:

```
$ mdadm --create --verbose --level=1 --metadata=1.2 --raid-devices=2 /dev/md/MyRAID1Array /dev/sdb1 /dev/sdc1
```

The following example shows building a RAID5 array with 4 active devices and 1 spare device:

```
$ mdadm --create --verbose --level=5 --metadata=1.2 --chunk=256 --raid-devices=4 /dev/md/MyRAID5Array /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1 --spare-devices=1 /dev/sdf1
```

The following example shows building a RAID10, far2 array with 2 devices:

```
$ mdadm --create --verbose --level=10 --metadata=1.2 --chunk=512 --raid-devices=2 --layout=f2 /dev/md/MyRAID10Array /dev/sdb1 /dev/sdc1
```

The array is created under the virtual device /dev/mdX, assembled and ready to use (in degraded mode). One can directly start using it while mdadm resyncs the array in the background. It can take a long time to restore parity. Check the progress with:

```
$ cat /proc/mdstat
```

By default, most of the mdadm.conf configuration file is commented out, it contains just the following content The reader needs to edit this configuration file to list the physical disk drives used in the madm RAID.

```
/etc/mdadm.conf
```

```
...
DEVICE partitions
...
```

Once the configuration file is updated, the array can be assembled using the below mdadm command.

```
$ mdadm --assemble --scan
```

2.2. Logical Volume Management (LVM)

2.2.1. What is LVM?

It is a system of managing the logical volumes created across the physical disk drives attached to the server. LVM provides significant improvement over the more traditional way of partitioning physical disk and then formatting them with a filesystem to present to the OS. It is a lot easier when it comes to managing logical volumes or when you want to add additional physical disk drives to the server to increase storage space.

2.2.2. Installation of LVM

To make use of LVM, the reader needs to first enroll all the physically attached disk drives to LVM, thereafter, the physical disk drives are grouped into a volume group, logical volumes are then created across the volume group. This logical volume is formatted with a file system and presented to the OS as storage space.

First, make sure the lvm2 package is installed. If not, install it using the below command.

NOTE:

- If you selected LVM during the OS installation process, this step is not required as LVM would have already been installed.
- It is assumed that the reader has already partitioned the physical hard drives that will be enrolled into LVM. If not, please partition the disks now.

```
$ sudo apt-get install lvm2
```

To partition the physical disks connected to the server, use the “parted” program.

```
sudo parted -a optimal /dev/sdb
```

Command break down

- Sudo instructs the command to run as the root or administrator account
- Parted invokes the parted program
- -a optimal instructs the parted program to utilize an optimal partition size
- /dev/sdb is the physical disk connected to the server. If a disk is not listed, parted will assume the first physical disk in the server. running this command on the wrong physical disk is destructive. Please make sure that you are using the correct disk to partition.

After the parted command has been initiated, the reader will need to create a label and partition type. For the purposes of this guide, we will use a primary partition.

```
mklabel gpt
mkpart primary 1 -1
```

Repeat the above process for all physical disks you plan to add to LVM. Once all the disks have been partitioned, the next step is to enroll the physical disk drives into LVM by creating Physical Volumes (PV) across the partitions on the physical disk drive. For example, to create a physical volume on /dev/sda1, run:

```
$ sudo pvcreate /dev/sda1
```

You can confirm that the physical volume has been created using the following command:

```
$ sudo pvdisplay
```

The above command will list all physical disks enrolled into LVM.

With all the physical disks enrolled into LVM and physical volumes successfully created, we can move to the next step which is generating a “Volume Group” (VG). The volume group will stitch together the various physical volumes.

To create a volume group named vg0 with an associated physical volume of /dev/sdb1, run.

```
$ sudo vgcreate vg0 /dev/sdb1
```

NOTE:

- “vg0” is a descriptive name given to the volume group. This name will be used to differentiate between the various volume groups managed by LVM on the server.
- “/dev/sdb1” is the partition on the physical disk drive.
- To include additional physical volumes, simply add them one after the other keeping a space in between each physical volume.

```
$ sudo vgcreate vg0 /dev/sdb1 /dev/sdb2 /dev/sdb3 /dev/sdb...n
```

After the vgcreate command, you can confirm that the new volume group has been created by running the below command

```
$ sudo vgdisplay
```

NOTE:

- Additional physical volumes can be added to an existing volume group. Physical volumes are added with the “vgextend” command
- Command example : vgextend vg0 /dev/sdb4 /dev/sdb...n

If you got to this stage without any errors, then we are almost done. The next step is to create a Logical Volume (LV) with in the Volume Group. The LV will be formatted with a files system and presented to the OS

as storage space.

To create a LV named “home_lv” in the VG “vg0” with 300GB of capacity, run:

```
$ sudo lvcreate -n home_lv -l 300G vg0
```

or, to create a LV home in VG vg0 using all available “free” storage capacity, run:

```
$ sudo lvcreate -n home_lv -l +100%FREE vg0
```

The new LV will appear as /dev/mapper/vg0/home_lv. Now you can format the LV with an appropriate file system. Before moving to the formatting step, confirm that the LV vg0-home was created by using the following command:

```
$ lvdisplay
```

With the successful creation of the logical volume, all we need to do now is to format the logical volume and mount it in the OS before the storage can be used.

To format the LV, run the below command

```
mkfs.xfs /dev/vg0/home_lv
```

NOTE:

- To use the default Linux files system of “ext4”. Simply replace the text “xfs” with “ext4”.

Now mount the storage to the OS. First make a directory to mount the new logical volume.

```
sudo mkdir /media/new_storage
sudo mount –t xfs /dev/vg0/home_lv /media/new_storage
```

Command break down

- The mkdir command creates the “new_storage” directory in the higher level “media” directory
- mount –t xfs tells the command to mount the storage with the xfs file system. if you used ext4, there is no need for the –t xfs option
- /dev/vg0/home-lv is the newly created logical volume
- /media/new_storage is the mount location in the OS

Now, to confirm that your new storage has been mounted and available for use, run the df command or navigate to the new storage location, at the Command Line Interface (cli), type either of the below

```
df –Th
```

Or

```
ls /meda/new_storage
```

The first command will print out the list of mount points on the server with the file system and total space available. The second command will do a directory or file listing of all the directories / files in the new storage location. If you have just created this mount point then the location should be empty and should not have any

directories or files in the mounted directory.

To unmount the storage, use the “umount” command.

NOTE:

- When running the umount command, it will disconnect the newly mounted storage from the OS. Any client accessing this space will be disconnected.
- Make sure that no one is using the storage location before unmounting the storage.
- Using the mount command will only mount the storage for the current session, if you restart the server, the storage will not automatically remount. To make this mount point persistent across server reboots, you need to add the mount information into the /etc/fstab file.

2.2.3. Additional LVM operations

After extending or prior to reducing the size of a device that has a physical volume on it, you need to grow or shrink the PV using the pvresize command.

To expand the PV on /dev/sda1 after enlarging the partition, run:

$ sudo pvresize /dev/sda1

This will automatically detect the new size of the device and extend the PV to its maximum.

To shrink a physical volume prior to reducing its underlying device, add the --setphysicalvolumesize *size* parameters to the command, *e.g.*:

$ sudo pvresize --setphysicalvolumesize 40G /dev/sda1

Move physical extents:

NOTE:

- Before moving free extents to the end of the volume, one must run pvdisplay -v -m to see physical segments. In the below example, there is one physical volume on /dev/sdd1, one volume group vg1 and one logical volume backup.

$ sudo pvdisplay -v –m

```
  Finding all volume groups.
  Using physical volume(s) on the command line.
--- Physical volume ---
PV Name               /dev/sdd1
VG Name               vg1
PV Size               1.52 TiB / not usable 1.97 MiB
Allocatable           yes
PE Size               4.00 MiB
Total PE              399669
Free PE               153600
Allocated PE          246069
PV UUID               MR9J0X-zQB4-wi3k-EnaV-5ksf-hN1P-Jkm5mW
```

```
--- Physical Segments ---
Physical extent 0 to 153600:
  FREE
Physical extent 153601 to 307199:
  Logical volume      /dev/vg1/backup
  Logical extents     1 to 153599
Physical extent 307200 to 307200:
  FREE
Physical extent 307201 to 399668:
  Logical volume      /dev/vg1/backup
  Logical extents     153601 to 246068
```

One can observe FREE spaces are split across the volume. To shrink the physical volume, we must first move all used segments to the beginning.

Here, the first free segment is from 0 to 153600 and leaves us with 153601 free extents. We can now move this segment number from the last physical extent to the first extent. The command will thus be:

$ sudo pvmove --alloc anywhere /dev/sdd1:307201-399668 /dev/sdd1:0-92467

```
/dev/sdd1: Moved: 0.1 %
/dev/sdd1: Moved: 0.2 %
...
/dev/sdd1: Moved: 99.9 %
/dev/sdd1: Moved: 100.0 %
```

Resize a physical volume:

Once all your free physical segments are on the last physical extents, run vgdisplay and see your free PE.

Then you can now run again the command:

$ sudo pvresize --setphysicalvolumesize size PhysicalVolume

See the result:

$ sudo pvdisplay

```
PV         VG   Fmt  Attr PSize    PFree
 /dev/sdd1  vg1  lvm2 a--     1t     500g
```

Resize partition

Last, you need to shrink the partition with your favorite partitioning tool.

Repairing a volume group:

To start the rebuilding process of the degraded mirror array in this example, you would run:

$ sudo lvconvert --repair /dev/vg0/mirror

You can monitor the rebuilding process with:

$ sudo lvs -a -o +devices

Deactivating a volume group:

Just invoke

$ sudo vgchange -a n my_volume_group

This will deactivate the volume group and allow you to unmount the container it is stored in.

Renaming a logical volume:

To rename an existing logical volume, use the lvrename command.

Either of the following commands rename logical volume lvold in volume group vg02 to lvnew.

$ sudo lvrename /dev/vg02/lvold /dev/vg02/lvnew
$ sudo lvrename vg02 lvold lvnew

Then unmount the filesystem on the logical volume:

$ sudo umount /<mountpoint>

Finally, remove the logical volume:

$ sudo lvremove <volume_group>/<logical_volume>

3. User management

Before a user can log into a Linux server and use its resources, the user first needs a login account. An initial account with root level access is create during the OS installation. However, for other users to access the server, they would need an account. It is recommended that users do not share login accounts. When creating a new user account, the account is created as a standard user – this user will only have access to the user owned data. For the user to have the ability to run system level commands or to access data that does not belong to the user, the user would need to have administrative level access rights.

In Linux, administrative access is referred to as sudo. To grant a standard users administrative level rights, you need to add the user account into the sudoers group.

The below list of commands are used to create, delete and modify user accounts, groups and permissions

- adduser: create a new user account on the server.
- userdel: delete a user account and related files.

- addgroup: add a group to the system.
- groupdel: remove a group from the system.
- usermod: modify a user account.
- chage: change user password expiry information.
- Relevant files: /etc/passwd (user information), /etc/shadow (encrypted passwords), /etc/group (group information) and /etc/sudoers (configuration for sudo).

It is easy to create a new Linux user account from the command line. For example, create the H3Admin user account, run:

$ sudo adduser h3admin

The above command will prompt you to supply a password for the new user. Once the password is supplied, the command will complete and the new user account will be created.

Creating or resetting a user's password, type the following passwd command:

$ sudo passwd h3admin

You can verify the new account by typing:

$ id h3admin

You can also delete a user account by typing :

$ sudo userdel {userName}

To add a user to the a group, use the below command swopping "sudo" with the desired group name.

$ sudo usermod -aG sudo {username} (Ubuntu)

or

$ sudo usermod -aG wheel {username} (RedHat)

A standard user is able to reset their password by running the passwd command. To reset the password of another user, the user needs to have sudo rights

$ passwd <username> : logged in user to change their password
$ sudo passwd <username>: admin user to change another user's password

Similar to creating user accounts, you can create a new group with the groupadd command. The below command will create a new group labeled "test".

$ sudo groupadd test

Deleting a group is equally achieved by running the groupdel command

$ sudo groupdel test

The chage command is used to modify password expiry times. When you initially created a new Linux user

account, the account does not have an expiry date. If you do not have a method of managing user access to a server, it is a good idea to set a user account expiration date. To see the h3admin user account expiry date, run the below command.

$ sudo chage -l h3admin

To expire a user account on the fly, run the below command. the command will not affect a currently logged in user. When the user next logs into the server, they will be prompted to change their password.

$ sudo passwd -e h3admin

To set a password expiration date, run either of the below commands.

This command will configre the h3admin account password to expire in 15 day's.

$ sudo chage –M 15 h3admin

While the below command specifies a specific date when the password will expire

$ sudo chage –E 2022-01-20 h3admin

Using the –w option will warn the user before their password expires. In the below example, the h3admin user will be notified 7 days prior to their password expiring.

$ sudo chage -W 7 h3admin

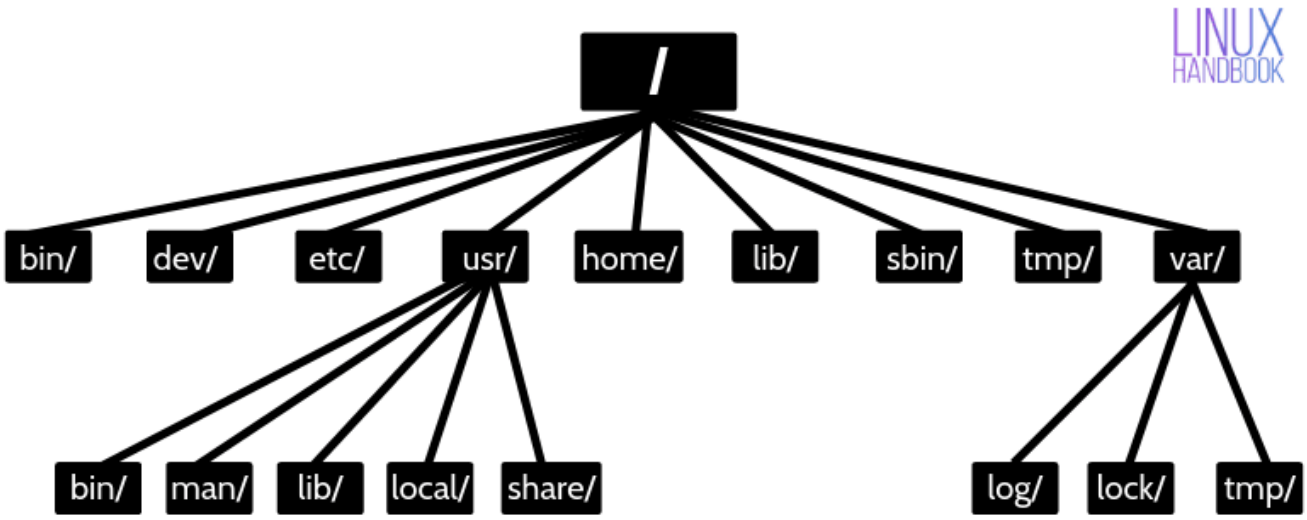
There are much more commands that you could use to manage user accounts but for now, the above commands will allow the Linux server administrator to add and remove user's and groups. Change a user's password or to set an password expiry date. Next, we will look at the Linux file system structure and how to navigate around the system.

4. Navigating the Linux Command Line Interface (cli)

As mentioned earlier in this guide, a Linux server OS installation, by default does not come with a GUI making the ability to efficiently navigate the Linux cli an essential skillset for all users of the system. Below are a list of commands with examples that will help get you started in moving around the cli and performing some routine tasks.

4.1. The Linux file system structure

Before we dive into navigating around the cli, lets first look at the Linux file system directory structure to give you an idea of where things are placed. Below is a graphical representation of the Linux file system structure.



Source: https://linuxhandbook.com/linux-directory-structure/

- / is the root directory. All directories branch off from root. The Linux root directory is like the C:\ in Microsoft Windows.
- The bin, usr and sbin directories hold all the binary files. The Linux binary files are similar to the Microsoft Windows .exe and .msi executable files. The bin directory typically holds are system based binary files, such as ping, grep, cp, excetra. The sbin directory typically holds system management type binaries such as iptables, fdisk, excetra; while the usr directory contains user based binary files.
- The etc directory usually holds all the configuration files for installed programs.
- The home directory is where all the user accounts personal data is stored. When you create a user account in Linux, a directory with the user accounts name is created in the /home/ directory.
- The lib directory contains all the system libraries that provide the helper files for the binaries. The Linux lib directory is similar to the Microsoft Windows .dll and .ini files.
- The temp directory is as the name implies, it's a temporary location for all user and system files. All data in this directory is typically deleted on system reboot.
- The var directory holds all the log, mail and spooler files. If you install a webserver on your server, the data files will typically be housed in the /var/www directory.

The above list of directories are some of the main Linux directories that a system administrator will use on a daily basis. There are additional directories that are not covered in this description but I recommend that you research these to get an idea of what they do. Examples of these extra directories would be the /media, /mnt and opt directories, to name but a few.

4.2. Useful commands to navigate the Linux file system

Now that you have an understanding of what the Linux file structure looks like, lets' look at some useful commands and examples of navigating around the Linux file system.

When you log into your server, you are typically dropped into your user account's home directory. A home directory is a directory set aside for your user to store personal files. It is the location in the filesystem where you have full dominion and do not require sudo access rights.

To find out where your home directory is in relationship to the rest of the filesystem, you can use the pwd command. This command displays the directory that we are currently in:

```
$ pwd
```

To view the contents of a directory, use the ls command:

```
$ ls
```

"ls" by itself will just list all the files and directories in a high-level directory. To list all of the contents in an extended form, including all hidden files, we can use the -la flag (for "long" output). the –l flag will display the file ownership and date it was created and modified. The –a flag will show the hidden filed. Hidden files in Linux have a period "." in front of the file or directory name.

```
$ ls -la
```

In Linux, every file and directory is under the top-most directory, which is called the "root" directory, but referred to by a single leading slash "/". An absolute path indicates the location of a directory in relation to this top-level directory. This allows us to refer to directories in an unambiguous way from any place in the filesystem. Every absolute path must begin with a slash.

While ls will allow you to look into a directory, the cd (change directory) command will allow you to change from the current directory to the directory listed in the cd command.

```
$ cd {Directory Path}
```

To move up one level, we can type:

```
$ cd ..
```

Likewise, too go back to your home directory, type the below command

```
$ cd ~/
```

The "~" sign represents the logged in user's home directory.

The ls and cd commands allow you to list directory contents and to move in and out of a directory. It does not however allow you to read the contents of a file. The ability to read the contents of a file at the cli, we use the "less" command. This is what we call a "pager", because it allows us to scroll through pages of a file. While the previous commands immediately executed and returned you to the command line, less is an application that will continue to run and occupy the screen until you exit.

We can use it for example to check a configuration file like services file that contains service information that the system knows about:

```
$ less /etc/services
```

To scroll, you can use the up and down arrow keys on your keyboard. To page down one whole screens-worth

of information, you can use either the space bar, the “Page Down” button on your keyboard, or the CTRL-f shortcut.

To scroll back up, you can use either the “Page Up” button, or the CTRL-b keyboard shortcut.

To search for some text in the document, you can type a forward slash “/” followed by the search term. For example, to search for “mail”, we would type:

/mail

This will search forward through the document and stop at the first result. To get to another result, you can type the lower-case n key. To move backwards to the previous result, use a capital N instead

When you wish to exit the less program, you can type q to quit.

To create a file, we use the touch command.

$ touch <path-to-new-file>/<new-file-name>

To create a directory, we use the “mkdir” command. Similar to the touch command, the mkdir command allows us to create empty directories. For instance, to create a directory within our home directory called test, we could type:

```
$ cd ~/
$ mkdir test
```

The “mv” command is used to move or rename a file or directory. We can move a file to a new location using the mv command. For instance, we can move file1 into the test directory by typing:

```
$ mv file1 test
```

The mv command is also used to *rename* files and directories, to rename the test directory to testing, we could type:

```
$ mv test testing
```

"cp” is used to make a new copy of an existing item. For example, we can copy file3 to a new file called file4:

```
$ cp file3 file4
```

To copy the a directory structure to a new structure , we could type:

```
$ cp -r DIR NEWDIR
```

To copy a file into a directory, just use cp with file name and directory path:

```
$ cp file1 {dir_path}
```

To delete a file, you can use the rm command.

Note:

- Be extremely careful when using any destructive command like rm. There is no “undo” command for these actions so it is possible to accidentally destroy important files permanently.

To remove a regular file, just pass it to the rm command:

$ rm file_name

Likewise, to remove *empty* directories, we can use the rmdir command, we can type:

$ rmdir directory_name

If you wish to remove a *non-empty* directory, you will have to use the rm command again. This time, you will have to pass the -r option, which removes all of the directory’s contents recursively, plus the directory itself, we can type:

$ rm -r directory_name

Once again, it is worth reiterating that these are permanent actions. Be entirely sure that the command you typed is the one that you wish to execute.

nano is one of the simplest command-line Linux text editors, and is a great starting point for beginners. We can open the “file1” file for editing by typing:

$ nano file1

To save our work after finishing, we can type: CTRL-O and then try to exit the program by typing:

CTRL-X

When you are unsure of what a command does or how best to use a command at the cli, you are able to invoke help using the --help option on a command or use the man command. The man command is short for “manual pages” and will list all the options you can use with the specified command. The --help option will give a quick summary of available options to use.

To see what options are available to use the rm command, type the below command

$ rm --help

Similarly, we can use the man command to print out the full capability of a command

$ man rm

Let's now look at how to add and remove software via the cli

5. Installing software in Linux

The process is fairly simple and can generally be accomplished with a one line command. In Linux, you typically install software in one of three ways.

a) Compile software from source

b) Install software directly from a repository, and,

c) You can use the following command to install any available software :

$ sudo apt install {Software Name}

To delete an installed software type :

$ sudo apt remove {Software Name}

The above command will install software from the apt repository. When using a Linux server you will most likely come across .tar.gz and .deb file. Instruction on using these types are files will be discussed in the intermediate Linux guide.

6. Installing and Enabling OpenSSH on Linux

SSH software packages are often installed during the OS installation, however, if the reader omitted to install these packages during the OS installation, they could easily be installed by completing Step 1, outlined below.

Step 1: Install OpenSSH Server Software Package
Enter the following command from your terminal to start the installation process:

$ sudo apt –y install openssh-server openssh-clients

This command installs both the OpenSSH client applications, as well as the OpenSSH server daemon, sshd.

Step 2: Starting SSH Service
To start the SSH daemon on the OpenSSH server:

$ sudo systemctl start sshd.

Step 3: Check sshd status
Check the status of the SSH daemon:

$ sudo systemctl status sshd

As we have previously started the service, the output confirms that it is active.

To stop the SSH daemon enter:

$ systemctl stop sshd

Step 4: Enable OpenSSH Service

Enable SSH to start automatically after each system reboot by using the systemctl command:

$ sudo systemctl enable sshd

To disable SSH after reboot enter:

$ sudo systemctl disable sshd

The most common settings to enhance security by changing the port number, disabling root logins, and limiting access to only certain users.

To edit these settings access the /etc/ssh/sshd_config file:

$ sudo nano /etc/ssh/sshd_config

To disable root login:

```
PermitRootLogin no
```

By default, ssh listens on port 22. It is a good idea to change this to run on a non-standard port for security reasons – especially when your server is exposed to the internet. For example, in the sshd_config file, change the port from 22 to 2002

```
Port 2002
```

NOTE:

- Remember to uncomment the lines that you edit by removing the hashtag.
- In linux

Save and close the file. Restart sshd:

```
$ service sshd restart
```

To copy files (big or small) from a one machine to another in the same network, you can use scp command. The overall format of the scp command is “scp <path-to-source> <path-to-destination>”. When either the source or destination is on a remote system, you would need to supply the username and IP or DNS name of the remote host. In the below example, File1 resides on the local server and is being copied over to the remote host “machine_IP”. As you can see, the remote host includes the user account on the remote server an includes the full path to the destination.

```
$ scp File1 user@machine_IP:/where/to/copy
```

You will be asked to enter the user password, type it and hit enter to continue.

For example , I want to copy a file name merged.txt to another machine with IP address 192.168.0.100 , where my username is “admin” and want to copy the file into my home directory on the remote machine or server :

```
$ scp ~/merged.txt admin@192.168.0.100:/home/admin
```

Login into another machine or server using SSH. In order to login into another machine where you have an active account on it without physically moving to it, you can simply use the ssh command :

$ ssh user@machine_IP

You will be asked to enter your password, type it and hit enter. To exit out of the remote server, type exit to close your session.

To restart or shutdown a local or remote server, type the below commands

$ ssh shutdown –r now

The above command will immediately reboot the server

$ ssh shutdown –h now

Running the above command will shut down the server. When executing the command on a remote server, you will need to have physical access to the server to switch it back on again.

Use the shutdown command with caution.

Next, let's take a quick look at configuring a firewall to implement basic security to the server.

7. Firewall Settings

A firewall forms part of your first line of defense against unauthorized access to the server. With the rapid increase in cyber related crime and the launch of government based data security policies such as POPI and GDPR, a solid security implementation is essential. At a technical level, security tools generally fall into two categories: Intrusion Prevention Systems (IPS)and Intrusion Detection Systems (IDS).

Intrusion Preventions Systems are tools and implementations designed and configured to limit the attack surface of a server. It's main aim is to prevent unauthorized access to a server.

Intrusion Detection Systems on the other hand are a set of tools and policies designed and configured to detect and track an intrusion if an attacker was able to get around the IPS systems.

The topic of security is vast and not within the scope of this beginners guide. For this section, we will concentrate on implementing a local firewall on the server. Look at how to open and close ports to limit and control access.

Iptables is the underlying firewall system on a most Linux distributions. iptables however is not "beginner" user friendly – as such, many distributions have provide a second firewall layer to simplify management and configuration of a firewall. On Ubuntu servers, this second layer is the Uncomplicated Firewall or more commonly referred to as ufw. Ufw forms part of the OS installation but is not enabled by default.

To see the status of ufw, at the cli, type

$ sudo ufw status

The output of the command will indicate if the firewall is active or inactive. If the system is inactive and you wish to activate it, run the below command.

NOTE:

- When working on a remote server, add the “allow ssh” rule before enabling the firewall as this might disconnect you from the remote server and you would need to physically access the server to add the rule.

$ sudo ufw enable

The expected output will be “*Firewall is active and enabled on system startup*”. This shows that the firewall has been enabled and will be activated at each system reboot. To confirm this, run the systemctl status command. Look for the “loaded” and “enabled” text. the loaded texts confirms that ufw has been loaded and enabled, ensures that ufw is loaded at each system reboot.

sudo systemctl status ufw.service

● ufw.service - Uncomplicated firewall

Loaded: **loaded** (/lib/systemd/system/ufw.service; **enabled**; vendor preset: enabled)

Active: active (exited) since Fri 2021-07-23 08:19:36 SAST; 4h 3min ago

Docs: man:ufw(8)

Main PID: 316 (code=exited, status=0/SUCCESS)

Tasks: 0 (limit: 38181)

Memory: 0B

CGroup: /system.slice/ufw.service

Warning: journal has been rotated since unit was started, output may be incomplete.

Now when you run the “sudo ufw status” command, you will notice that the output has been changed to “active”.

If for any reason ufw is not installed on your Ubuntu server, you are able to install the ufw firewall program with the below command.

$ sudo apt install ufw

The default policy configuration of the ufw firewall is to deny all incoming access. This is the reason you need to ensure that the ssh port is opened before enabling the firewall.

NOTE:

- When enabling the firewall, if you have changed the ssh port in the sshd_config file, be sure to use the custom port when allowing ssh access through the firewall.

To allow the ssh port through the firewall, run the below command

$ sudo ufw allow openssh

Now when you run the systemctl status command, the list of open ports will be listed.

To allow ssh access on a custom port, use the below command. In the below example, the command

instructions ufw to allow ssh through the firewall on port 2002. Any ssh connections made on port 22 will be dropped.

```
$ sudo ufw allow openssh 2002
```

It is also possible to restrict IP access to make the connection even more secure. To deny all connections into a machine, we can use the next command :

```
$ sudo ufw deny all
```

Always remember to restart the ufw service after each restriction implemented.

```
$ sudo systemctl restart ufw
```

There are many more variations to the ufw configuration, for example, you can grant or deny access to or from a specific host, IP address, subnet, or network interface. These configurations however are outside the scope of this guide.

Let's now move our attention over to the setup and configuration of the Linux NFS file server.

8. NFS Server and Client Installation on Linux

NFS, or Network File Server is the program and process of sharing storage on the server across a network to remote clients. To setup a NFS server, you need to configure the server to offer storage across the network and you need to install the NFS client software on the remove machine to be able to mount the shared storage.

At NFS server end

As the first step, we will install these packages on the Linux server with:

```
$ sudo apt install nfs-kernel-server nfs-common
```

Now create the directory that will be shared by NFS for example:

```
$ sudo mkdir /home/nfsshare
```

Change the permissions of the folder as follows:

```
$ sudo chmod -R 755 /home/nfsshare
$ sudo chown nfsnobody:nfsnobody /home/nfsshare
```

Next, we need to start the services and enable them to be started at boot time.

```
$ sudo systemctl enable rpcbind
$ sudo systemctl enable nfs-server
$ sudo systemctl enable nfs-lock
$ sudo systemctl enable nfs-idmap
$ sudo systemctl start rpcbind
$ sudo systemctl start nfs-server
$ sudo systemctl start nfs-lock
$ sudo systemctl start nfs-idmap
```

Now we will share the NFS directory over the network a follows:

```
$ sudo nano /etc/exports
```

We will make a sharing point /home/nfsshare. Edit the exports file as follows:

```
/home/nfsshare    192.168.0.101(rw,sync,no_root_squash,no_all_squash)
```

Note:

- 192.168.0.101 is the IP of the client machine, if you wish that any other client should access it you need to add it's IP otherwise you can add "*" instead of IP for all IP access.

Finally, start the NFS service:

```
$ sudo systemctl restart nfs-server
```

NOTE:

- Again we need to add the NFS service override in CentOS 7 firewall-cmd public zone service as:

```
$ sudo firewall-cmd --permanent --zone=public --add-service=nfs
$ sudo firewall-cmd --permanent --zone=public --add-service=mountd
$ sudo firewall-cmd --permanent --zone=public --add-service=rpc-bind
$ sudo firewall-cmd –reload
```

Note:

- If it will be not done, then it will give error for Connection Time Out at client side.

Now we are ready with the NFS server part.

NFS client end

Install the nfs-utild package as follows:

```
$ sudo apt-get install nfs-common
```

Now create the NFS directory mount point:

```
$ sudo mkdir -p /mnt/nfs/home/nfsshare
```

Next, we will mount the NFS shared home directory in the client machine as shown below:

```
$ sudo mount -t nfs 192.168.0.100:/home/nfsshare /mnt/nfs/home/nfsshare
```

It will mount /home/nfsshare of NFS server. Now we are connected with the NFS share.

Permanent NFS mounting

We have to re-mount the NFS share at the client after every reboot. Here are the steps to mount it permanently by adding the NFS-share in /etc/fstab file of client machine:

```
$ sudo nano /etc/fstab
```

Add the entries like this:

```
[...]
192.168.0.100:/home/nfsshare    /mnt/nfs/home/nfsshare    nfs defaults 0 0
```

This will make the permanent mount of the NFS-share. Now you can reboot the machine and mount points will be permanent even after the reboot.

9. LINUX: Network Card Configuration in Linux

The network configuration tools and network configuration files are different for Ubuntu/Debian vs Red Hat/Fedora based systems.

The following commands will start, stop or restart networking:

```
$ sudo /etc/init.d/networking start
$ sudo /etc/init.d/networking stop
$ sudo /etc/init.d/networking restart
```

Ubuntu 18.04 or 20.04 uses netplan as a default network manager. The configuration file for the netplan is stored in the */etc/netplan* directory. You can find this configuration file listed in the /etc/netplan directory the following command:

```
$ ls /etc/netplan
```

The above command will return the name of the configuration file with the .yaml extension, which in my case was 01-network-manager-all.yaml.

Before making any changes to this file, make sure to create a backup copy of it. Use the cp command to do so:

```
$ sudo cp /etc/netplan/01-network-manager-all.yaml 01-network-manager-all.yaml.bak
```

Note: You might have a configuration file with the name other than the 01-network-manager-all.yaml. So make sure you use the right configuration file name in the commands.

To view the current IP address of your machine and the network port name, you can use the following command: *$ ip a*

Usually, after the lo , you will find a second network name with a static ip address , it can be named like eth0 ..

You can edit the netplan configuration using any text editor. Here we are using the Nano text editor for this purpose.

```
$ sudo nano /etc/netplan/01-network-manager-all.yaml
```

Then add the following lines by replacing the interface name, IP address, gateway, and DNS information that fit your networking needs. (we took eth0 as a network card name, it can be different)

```
network:
version: 2
renderer: NetworkManager
ethernets:
 eth0:
  dhcp4: no
  addresses:
  - 192.168.0.140/24
  gateway4: 192.168.0.2
  nameservers:
   addresses: [8.8.8.8, 8.8.4.4]
```

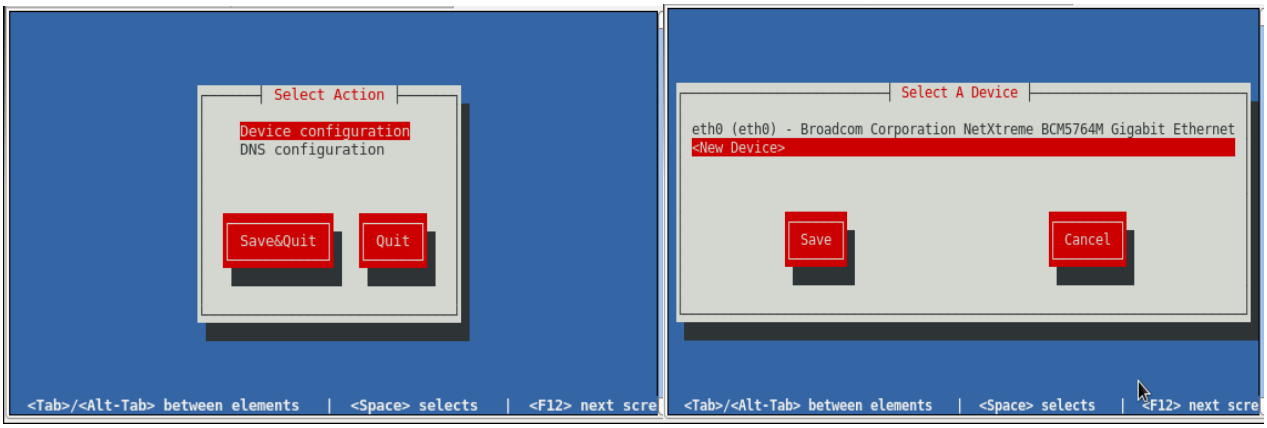
Once done, save and close the file.

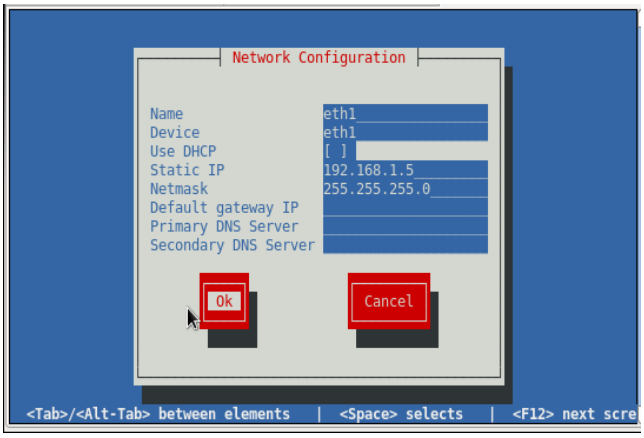
Apply the new settings with this command :

```
$ sudo netplan apply
```

For a RedHat based OS, The most simple method of configuring Network is by using the console based GUI tool:

```
$ system-config-network
```





Note:

- Both static and dynamic configurations can be set. If configuring for DHCP, go to that line and pres the space bar. An asterisk "*" will appear to show that this option has been selected.

10. Screen

SSH is great for connecting to remote Linux machines over the network or the internet. One of the downsides, however is that when you close the SSH connection or if the link to the internet or network is disrupted, the SSH connection will be severed and the active job or task that was running will be ended. This is especially true when performing long running tasks remotely and the user needs to either shutdown their client machine or push the operation to the background freeing up the terminal. In Linux, there are a few way to address these issues, one of the more commonly used solutions is a program called “Screen”.

Screen allows the user to create a screen session that can be detached, allowing the user to shut down their client machine while the task continues to run in the background on the remote server. Multiple virtual terminals can be created in a single screen session. Screen sessions can be detached and reattached. It

allows the user to move back and forth, between various virtual terminals in a screen session.

In the Ubuntu 18.04 and 20.04 server editions, screen is installed by default, to confirm that screen is installed, run the below command:

$ screen --version

If installed, screen should print the installed screen version to the screen. If not, it's possible that screen is not yet installed. Follow the below command to install screen from the Ubuntu apt repository.

$ sudo apt update && sudo apt upgrade -y

$ sudo apt -y install screen

Below are some tips for using the screen application.

NOTE:

- Screen needs to be installed on the remote machine
- SSH into the remote machine and run the screen command. Do not execute screen from your client machine if you want to keep the task running after you detach from screen and shut down your client machine.

From your client machine, SSH into your remote server and start a screen session with the below command

$ screen

Screen will open a terminal. You are now able to work as normal and perform your long running tasks. Once you the task has started, the user can simply detach from the screen terminal – the task will continue to run on the remote machine, to reconnect, you simply SSH into the remote server and attach to an active screen session.

NOTE:

- By default, users can only attach to screen sessions opened under their account. To access someone else's screen session, the user would need to enable multiuser support and then allow the specific user account access.

While the screen program can be started by simply typing the command "screen" at the cli, it's not very practical when you have multiple screen sessions running on the remote server as you won't be able to distinguish between them. It is therefore advisable to start the screen session with the "-S" option. With this option, the user is able to give a user-friendly name to the screen session, now, when you list all the open screen sessions on a remote server, you will be able to distinguish between.

To start screen with a user-friendly name attached to it, run the below command:

$ screen -S {screen name}

When executing a screen level command, always start with depressing the "control" key then the "a" key. An example

would be when you want to detach from an active screen session,

$ ctrl+a and then depress the "d" key

Running the above command will detach from the active screen session and drop you to the server's cli. From there, you could perform other tasks, terminate the SSH session, etc.

To reattach to a screen session, use the "-r" option or the "-ls" option to list all the active screen sessions

Once detached from the active screen session, you can list all active screen sessions with the below command

$ screen -ls

To attach a single screen , use

$ screen -r .

The above screen command will attach to the active screen session if there is only one active screen session running. If there are multiple screen sessions running, you will be prompted to select session you want to attach to. You do this by running the "screen -r" command and include the name of the specific screen session.

Once in the session, you can created new virtual terminal's in the screen session by using the below command combination

$ ctrl+a and then depress the "c" key

If you have multiple virtual terminals running in a single screen session, you can navigate between them using the "n" and "p" keys. "n" will move one virtual terminal forward, similarly, the "p" toggle will move one virtual terminal back from the current virtual terminal position.

$ ctrl+a and then depress the "n" key

$ ctrl+a and then depress the "p" key

To allow other users on the remote system to attach to a screen session you created under your account, create a screen session with a user-friendly name. While in the screen session, enable multiuser and then add the specific user. Example,

$ screen –S test_multiuser_option

$ ctrl+a and then :multiuser on

$ ctrl+a and then :acladd new_user

The "new_user" will be able to run the below command to attach to your screen session

$ screen –x user/test_multiuser_option

In the above command, "user" is the user that originally created the screen session, "test_multiuser_option" is the

name given to the screen session.

To list all of screen's capabilities, use the "man screen" command to list all command combinations and option.

This concludes the chapter on the "Ubuntu - getting started" started guide. Once you are comfortable with the basics, have a look at our level 2 guide which introduces containerization using Docker. Our level guide goes one step further and introduces a basic Beowulf HPC cluster with SLURM as a resource manager. If you intent to follow all the guides in this technical series, have a look at our supplement Globus Online guide to simplify your data transfer needs.

Should you have any comments or recommendations regarding this guide, please drop us a note on our helpdesk at helpdesk@h3abionet.org.

--oOo END oOo--