**DevOps**

**Miss first video**

**SDLC(software development life cycle): is a process normally follow three steps: design,developpemnt and test.**

**The aim of SDLC the high quality delivery project**

****

**The main focus as DevOps engineer, building(developpement),testing(quality assurance QA) and deployment(production server for customer) a software.**

**DevOps basically fasten this process and improve this process to deliver the code quickly by ensuring all of these method (building, testing,deployment).**

**As a devops we should focus on automation and improving efficiency of building, testing and deployment.**

**Virtual Machine:virtual environnement that function as virtual computer systems and this systems has own cpu, ram and own hardware.**

**super hypervisors are vmware, xen**

**Aws:Amazon Web Services, Inc. (Amazon Web Services)**

**Azure: Microsoft Azure est la plate-forme applicative en nuage de Microsoft**

**Connect with virtual machine AWS from local via mobixtreme :**

**By mobiextrem we can connect free with 10 virtual machine**

**Copie ip address from Aws and connect with key-gen was created during creating virtual machine**

**Commande:ssh -i C:/Users/moham/Downloads/test.pem [ubuntu@13.53.132.194](mailto:ubuntu@13.53.132.194) (-i for identity file)**

**To see tuto 6**

**Operating system: communicate between hardware and software**

**What does do kernel in operating system:1. kernel is heart of operating system , its has four primary responsibilities: device management,memory management,process management, handling system calls**

1. **System and libraries do Performing a task**
2. **System software,users process, compilers**

**Fundamental of shell scripting: shell: is way talk to operating system. Commande line is shell commande**

**Bash is a script-style command-line interpreter.**

**Commandes:**

**Commande to see property: free -g**

**Commande disk size: df -h**

**Commande for everything manage cpu, memory etc: top**

**Shell script:**

**For details : man touch/man mkdir**

**Open a file with editor: vi file, vim file (but vim need to install)**

**To write something ESC then i for insert,for save and exit ESC then:wq! , for close without save :q!**

**To write shell script at first write shebag #!/bin/bash ou #!/bin/sh**

**Bash, ksh,sh, dash etc work like as compiler java**

**Execute file: ./ means this file is executable(must give chmod) or write sh file with extensions(sh name.sh)**

**Chmod define for user, group, everyone**

**Linux chmod: 4 is read, 2 is for write, 1 is for execute.**

**Exemple: chmod 444 xyz**

**To see history commande: history**

**DevOps: architecture management, code, configure management all are doing by shell automatic.**

**In a word shell script is a tool for automatisation**

**Commande free for information ram: free**

**Top commande for what are processes are running in machine, see memory, cpu usages etc. (commande:top)**

**4 importants commands linux:df(all avaible space in virtaul machine),free(memory of the machine),nproc(display cpu of the current machine),top (print all infos process,slepping process, consuming memory) we can write: df | free | nproc |top**

**If any problem in virtual machine how to check: create a file script to check the problem.**

**It must be write meta-data,and debug mode(to)set -x (is echo state)**

**Another echo commande: set -e(exit the script wehn where there is an error),another echo commande: set -o(pipefail)(pipe fail is very important because if fail the script will execute otherwise not execute)**

**To see the process running in computer: ps -ef(p:process,-e:entire process)**

**To see particular process: ps -ef |grep ‘amazon’**

**Every virtuelle has 3 chanel: stdin, stdout,stderr so, if we write date | echo‘today is’ ,this case first commande not to send result to second commande beacause this is stdin.so result is today is.**

**For greping id instead of entire process or details we can use awk.**

**Awk: is pattern scanning and processing language.**

**Awk commande retrive information specific column.**

**Commande : ps -ef |grep amazon |awk -F’’ ‘‘ ‘{print$2}’**

**Grep something from file: grep grep\_target filename**

**Commande CURL: retrieve information from internet(api)**

**curl lien recover information then grep recover error :curl lien |grep ERROR**

**Another commande: curl -X Get api\_link**

**Wget commande: wget command as same as curl diffierncef download logfile and curl command display not download.**

**Linux commande su means (switch user)**

**Find commande: to find a file from your system: sudo find /pam.d**

**Trap commande: trap for use traffic signal,**

**Kill commande: kill commande uses for kill a process**

**Questions on shell scripting**

**I use shell scripting for listing file, to find something, etc.**

1. **List all the processes: ps -ef**
2. **Find me id a perticular process: ps -ef |awk -F’’ ’’ ‘{print$col\_num}’**
3. **Write all logs store in s3 backet or google storage: curl google.com | grep HREF**
4. **Write a shell script to print number numbers divided by 3 and 5 but no 15?**

**For n in {1..100}->do->if [$((n%3))==0] ->then ->echo «»->fi->done**

1. **Write a script to print number of s in mississippi**

**X=mississipi->grep -o ‘s’ <<<’$x’ | wc -l(here -o:only, ‘s’, <<<: ,wc:word count, l:length)**

1. **What is crontab in linux?can you provide an example?**

**Crontab : like alert , it will execute a certain time which is defined by devops**

**Report: node health**

1. **How to open read-only file?**

**Vim -r file\_name.txt**

1. **What is difference between soft link and hard link?**

**In Linux there is two link soft and hard link, hard link created a backup into memory so if original file deleted but hard link file will be rested.example:create and save file**

**Soft link: exam: python install in my machine so python file save into same point of memory if deleted the file il will be deleted no backup file exist.speacilly for create a alias**

1. **What is the difference between break and continue?**

**Break: breaking the execution file, Continue: continuing the execute->skip this and continue the next. Example: in for loop we can put break and continue.**

1. **What are disadvantages of the shell scripts.**

**-Errors are frequent and costly and a single error can alter the command.**

**-The execution speed is slow**

**-Bugs or inadequacies in the language’s syntax or implementation**

**-Every time a shell command is executed, a new process is lanuched.**

1. **What are different types of loops and when to use?**

**Such as while, do while, for etc. Purpose same as other languges**

1. **Is bash dynamically or statically typed and why?**

**Bash statically typed, example: x=5 (but dynamically declared var x string in other language)**

1. **Explain about a network troubleshooting utility?**

**Traceroute and tracepath**

1. **How will you sort list of names in a file?**

**Sort**

1. **How will you manage logs of a system that generate huge log files everyday?**

**Linux admin and administrateurs: if we save all logs it will consume memomry so we can use logrotate to manage linux machine logs.we can say every 24 hours save logs in different logs(gzip,tar,zip etc.) and can say after 30 days delete logs.**

Cloud infrastructure:

Why need to move cloud structure like, aws or azure,

1. Manageability(for exemple startup need to much teams)
2. Cost (pay as you go)

**EC2**: Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud.

**AWS Lambda** is a server-less, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers.

**Amazon S3** gives every user, its service access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of websites

**AWS Identity and Access Management (IAM)** roles are entities you create and assign specific permissions to that allow trusted identities such as workforce identities and applications to perform actions in AWS.

**Cron jobs, sh**ort for "chronos" in Greek, are scheduled tasks that automate repetitive processes and streamline operations on AWS.

**AWS CLI**: is an open source tool from AWS. You can use it to interact with AWS services using commands in your command line shell.

**Clé d'accès:AKIA6ODU4YRF4HMY7D47, Clé d'accès secrète:**1saeAeqQ3HvnaGhdImGJy0JBWg1wAeHBRpV4VK2L

Shell scripting project used GITHub Api integration

For access repository in collaboratives need to write shell scripts github integration.we can two integration application one is api and other is CLI.

API: application interface, CLI: commande line interface

Github APIs( or Github ReST APIs) are the APIs that you can use to interact with GitHub. They allow you to create and manage repositories, branches, issues, pull requests, and many more.

Multistage and destroy-less image: Creating images in different stage, ex: build stage and run stage, destroy-less: docker runtime image.

**Why need container instead of virtual machine:**

**In virtual machine we can create lot of machine or instances and can run our application. Each machine or instances allocate memory and disk but alls are not use that's why need container. In a word too much physical server solved virtual machine and too much virtual machine extends container.If you deploy application in virtual machine you never seen message out of memory or storage so all time reste extra memory or disk , this problem is also solve by container.**

**Docker is containerisation platform.**

**A container is a bundle of Application, Application libraries required to run your application and the minimum system dependencies.**

**Containers vs Virtualmachine:**

**1. Resource Utilization: Containers share the host operating system kernel, making them lighter and faster than VMs. VMs have a full-fledged OS and hypervisor, making them more resource-intensive.**

**2. Portability: Containers are designed to be portable and can run on any system with a compatible host operating system. VMs are less portable as they need a compatible hypervisor to run.**

**3. Security: VMs provide a higher level of security as each VM has its own operating system and can be isolated from the host and other VMs. Containers provide less isolation, as they share the host operating system.**

### What is Docker ?:Docker is a containerization platform that provides easy way to containerize your applications, which means, using Docker you can build container images, run the images to create containers .

Kernel: kernel is heart of operating system.



Docker daemon: is heart of docker A persistent background process that manages Docker images, containers, networks, and storage volumes, if daemon is down then docker will be down

### Docker LifeCycle

We can use the above Image as reference to understand the lifecycle of Docker.

There are three important things,

1. docker build -> builds docker images from Dockerfile
2. docker run -> runs container from docker images
3. docker push -> push the container image to public/private regestries to share the docker images.

#### Docker registrie

A Docker registry stores Docker images. Docker Hub is a public registry

#### Dockerfile

Dockerfile is a file where you provide the steps to build your Docker Image.

Difference between Github and Docker hub: both of versioning but github storing source code but dockerhub storing images.

Install docker in EC2: sudo apt update -y, install docker: sudo apt install docker.io -y, check docker: sudo systemctl status docker

If we run docker: docker run hello-world (it will not work because by default docker install everything in root user.so I need to add ubuntu user as docker group. Commande: sudo usermod -aG docker ubuntu and then must be restart if not will work.

Now it will be run: docker run hello-world

Now clone a project from github then enter into project,

Then build docker image :

docker build -t sohrab109/my-first-docker-image:latest **.** (here sohrab109 username, then create a repo like github, -t for tag latest, . is for create in same répo)

After build docker image run docker image:

docker run -it sohrab109/my-first-docker-image:latest

Now i want to share my docker image, I need to login and push:

Login:docker login success then:

Docker Volume and bind mount: if a container go down then log file also go down so admin cant monitor or may be lost information of all clients. To avoid this problem docker user Bind mount.

By default container does not have own file system thats why container very light weight by nature.It used of all ressources from host operation.Container can access only cpu, processeur,storage etc. from host.

Bind Mount: a directory inside into container which keep backup in host.

Volume: is offer better lifecycle,logical partition on host.

In a summary: in container raise three problems: 1. a container does not keep previous logs or auditing a company can get information of clients or others.

2.If two contaier front and back end try to share information each other if backend gone down there is problem persue means backup.

3.if app trying to read somes informations from host operating system it does not know how to read files from operating system.

Solve this problem docker propose solutions Bind Mounut and Volume,

Bind Mount: it allows to bind a directory inside into container with a folder on a host.(binding a specific directory on host with a contaienr)

Volumes: volume like as bind mount but also offer better life cycle.volume is logical partition on host.volume can manage by docker CLI.

Commande create volume: docker volume create volume\_name

To see list voulume: docker volume ls

Details where volume is: docker volume inspect volume\_name

For delete volume: docker volume rm volume\_name

Commande display first 5/6/7 things: docker images | head -5

Docker networking: is allows container to each other and core system.By default container can communicate with host or machine, container does not have complete operating machine. If we create a container by default container has different IP from host so normally form host to container not ping. But by default when create a container docker create a virtual network between host and container is called virtual bridge(called ethernet). generally when create a container it creates a network which difference from host but by virtual bridge container and host can communicate.

Docker offer three popular networking: virtual network and host network(container ip address machine itself) and third is overlay networking which is used by kubernetes.

Run a container: docker run -d --name container\_name image\_name:latest

Enter into docker container as root: docker exec -it contianer\_name /bin/bash

Now update : apt update then install ping

Commande : apt-get install iputils-ping -y

Check ping: ping -V

Now check ip address: go to terminal(out of root):docker inspect container\_name

This moment we can check network by ping command

Go to each container root then ping one to another :ping ip\_address

Check docker network: docker network ls

Create custom bridge/network: docker network create network\_name

Now create a container and assign network: docker run -d --name container\_name --network=creating\_network image\_name

Not create a container with host network: docker run -d --name container\_name --network=host image\_name (result: this network is null means it has network machine)

**Kubernetes**

Kubernetes container orchestration plateform.

Create container in docker have 3 problems:1. one container impact another(if one conatiner take more ram and process may be other container will die).

2.auto healing: container start withself without intervention.

3.

Entreprise standard support: load balancer, firewall,auto scaling, atuo healing,api gateway.

But docker can not solve of all this probleme , this problem is solve by kubernets.

First probleme single host, auto scaling, auto healing and enterprise level support. Who is solving this problem in a word kubernetes.

Kubernetes: by default kubernets is cluster(group of nodes), kubernetes is installed in cluster. If a container affected whole application may be affected kubernetes immediately il put application in different nodes.(first problemes solved).By default kubernetes is cluster image so problem one is solved by kluster behavious of kubernetes.second solution tell kubernetes in yaml file increase container 1 to 10 tomorow have fastivel.(this is auto scaling)

Third problem: whenever healing kubernetes control and fix the damage.If a container down kubernetes immidialtely rollout new container end user does not understand one container down or something happen. This is auto healing.

In docker have not support enterprise nature like load balancer, firewall, support for api gateway, black listing,white listing etc. And docker has no support enterpirse solution that’s why docker never use in production.(for emxeple enterprise needs to list some client in black or white list but docker nerver possible).

K8s Architecture

Kubernetes have components data plane:kubelet, proxy, container runtime.

Kubetnetes create multiple node master and multiple node worker.

Kublet is responsible for running pod, pod is a collection of smaller container which have some advance capabilities.

By default kubernetes has feature called auto healing, if one pod down auto healing tell do something.

Kubernetes can support container D, creo, docker shim and any other contianer run times environment and support kubelet.

In docker has default bridge called docker 0 and in kubernetes has bridge called Q proxy and in every container and pod I have creating il will allocate ip address.

In worker node in kubernetes have three components : kube proxy, kublet,container runtime.

Kubelet: is responsible for creating pod and that the ensure pod is running state if it not it takes necessary action using kebernetes control plane and then we have Q proxy(kube proxy) which is responsible for the networking like generating ip address or load balnacing basically it uses ip tables your linux machine.

The components running appliction: cubelet is deploying, Q proxy is providing ip address, container runtime is providing the execution environnement for container. In spite of everyting why we need control plane tools? Because of needs some specific enterprise standards.

Now who is decided to create a pod and decided in node 1 or node 2 etc and may be need some more instructions so it should be has component which basically access core components kubernetes and takes all incomming request is called API server.In kubernetes Api server takes all request from externel world.

Api server is component that exposes kubernetes and receive all of request from external world.

For exemple a user want to create a pod and send a request to API server, Api server tell ok node 1 is free, schedule is a component on node in kubernetes is called scheduler.

The responsibility of schedular is scheduling on pod or ressource on kubernetes.who decides the information Api server and who acts on the information that is scheduler.

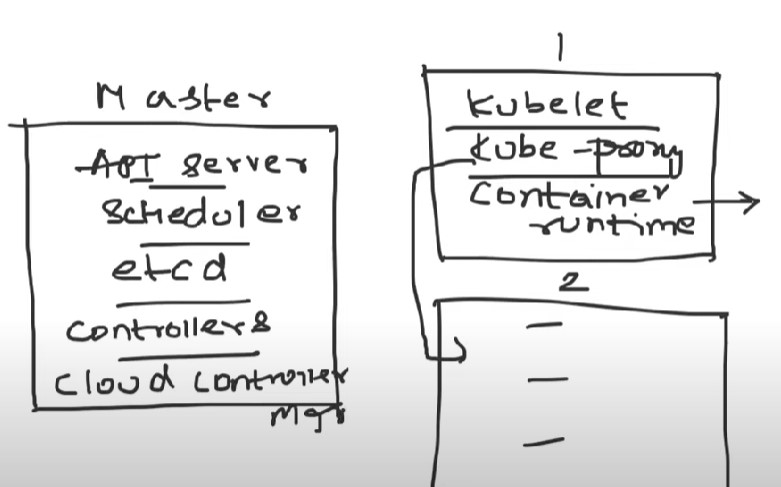
In production we need keep a bakcup entire cluster information there has to be a component in kubernetes basically act as backup service, basically act as a backing sotre of entire cluster information, etcd is basically key value store and entire information of kubernetes cluster information stored as object key vlaue pairs inside the etcd.

We have another components called controller manager and cloud controller manager Kubernetes. We know kubernetes subpport auto scalling for thats it has some controllers for exmple replica set, that is totaly controlled by controller manager.

Cloud control manager: we all know that kubernetes is run on cloud platform that is EKS/AKS/GKE etc.

In a summary: kubernetes has two component control plan, data plan.data plan have two worker node.

Control plan control the action and data plan executing the action.



**Docker(https://labs.play-with-docker.com/)**

**What is Docker(un ouitls qui simplement creer, depoloyer, executer application en utilisant container.)**

* **Docker is a containerization technology.**
* **Docker is a platform for packaging, deploying and running applications. Docker applications run in containers that can be used on any system: a developers laptop, systems on permises or in the cloud**
* **Containerization is technology that's been around for a long time, but its seen new life with Docker. It packages applications as images that contain everything needed to run them: code, runtime environment, libraires and configuration.**

**Why do you need Docker?**

* **Portablity**
* **Performance**
* **DevOps**
* **Cross Platform**

**What can it do?**

* **Containerize Applications**
* **Run each service with its dependencies in separate containers**

**Docker Commands**

1. **Docker --version**
2. **Docker images**
3. **Docker ps //list docker running container**
4. **Docker ps -a**
5. **Docker run -d -p 80:80 nginx**
6. **Docker stop container\_id**
7. **Docker rm container\_id**
8. **Docker rmi image\_id**
9. **Docker start container\_id**
10. **Docker logs -f container\_id**
11. **Docker logs --since=2h -f container\_id**
12. **Docker exec -it container\_id bash**

**Docker compose**

1. **Docker-compose up**
2. **Docker-compose up -d**
3. **Docker-compose rm**
4. **Docker-compose down**

**Run Applications as Docker Container**

**Multi-stage Docker: for reduce size of application so starting application run time will reduced.**

**Installing docker**

1. **Sudo apt-get update**
2. **Sudo apt-get install docker.io**
3. **Sudo systemctl enable docker**
4. **Sudo groupadd docker**
5. **Sudo usermod -aG docker $USER**
6. **Newgrp docker**
7. **Docker --version**

**Download your first Image**

**Docker pull nginx**

**Docker pull sohrab/appname**

**Docker pull sohrab/appname:latest**

**Note: Make docker file then create image and finally create container which is running portable.**

**To see docker container means which container is runnging: docker container ps**

**-stop container : docker container id**

**-to see images: docker images(after pull image)**

**-build images then it will be convert into container: Docker run -d -p 80:80 --name create\_container\_name shawon10/pythonapp**

**-stop service : docker stop container\_name**

**Git clone the project then enter the project.**

**Build docker as docker image: docker build -t imagename . (here t is target)**

**Check docker image: docker images**

**Run docker image as docker container: docker run -d -p 80:80 --name create\_containername imagename**

**Check docker container: docker container ps**

**To enter container: docker exec -it containername/id sh**

**Then enter: cd usr/share/nginx**

**(it interactive terminal)**

**Make docker image/Build docker image: docker build -t pythonapp . (here .means docker file in same file otherwise to show where is docker file like as ./foldername) (t means command run in termainal)**

**Run project or build container: docker run -d -p 4200:80 --name create\_containername imagename (here d means detach mode(run in background): I dont want to show log file, what is runing).P is port forwarding on which port want to run)**

**To enter docker image file: docker exec -it imagename**

**Show running container: docker ps**

**Show stop container: docker ps -a**

**Remove stop container : docker rm container\_name**

**Start container: docker start container\_name**

**Best practice note: multistage,naming conversion(image name=project name),**

**Enter into image: docker exec -ti 798bdc524756 /bin/bash**

**Without enter image and see details: docker exec -ti 798bdc524756 df -h**

**Without enter image: docker exec -ti 798bdc524756 find / -name nginx**

**Difference between copy and add:Docker use copy and add at same purpose they let you copy from specific location into a docker image.**

**Copy <src><dest> || Add<src><dest>**

**Little difference is: in ‘add’ compress file converted into uncompress then work like as ‘copy’**

**Installing Docker another train**

**Install docker engine: sudo apt-get update**

**Sudo apt-get install docker-ce docker-ce-cli container.io**

**Here ce=community edition, cli=docker engine,container.io=docker damen**

**(note first create virtual machine then give ip address remove docker command if exists)**

**Remove docker engine if exits: sudo apt-get remove docker docker-engine docker.io contianerd runc**

**Check dockers elements: sudo systemcli status docker**

**follow:https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository**

**Inspect docker image: docker image inspect docker\_image**

**To find image: find / -name image\_id**

**Running a basic web server: docker run -it --rm -d -p 8080:80 --name nginxcontain3 nginx (this command for running and create container)**

**Create a dockerfile and push into dockerhub**

**Create a Dockerfile and index.html then copy /usr/share/nginx/html/index.html**

**Then build docker file: docker build -t imagename .**

**Then run docker file: docker run -idt --name containername -p 8080:80 imagename**

**Push: docker login**

**docker tag firstimage:latest sohrab109/firstimage:latest (tag for image)**

**Docker push sohrab109/firstimage:latest (push)**

**Do link dockerhub with container: docker run -it -rm -d -p 8080:80 --name containername -v~/webcontent:/usr/share/nginx/html nginx**

**Link: https://www.stacksimplify.com/aws-eks/kubernetes-for-absolute-beginners/kubernetes-for-absolute-beginners/**

**Create docker network**

Why need to create network: allow containerized apps to function similarly to network services running directly on your host.

1. Docker network create chatbot
2. Docker network ls
3. Docker network inspect chatbot
4. Docker inpect --format = ’{{range.NetworkSettings}} {{ipaddress}} {{end}}’ chat\_app

****

**Docker 23**

**Linux**

**Linuxtricks.fr and quickref.me**

**sohrab@lenovo: ~$**

**Ici sohrab nom de utiilsatuer, lenovo nom de host, ~ dans le repartoire utilisateur, si c’etais root # au lieu de $, $ indicate utilisateur non privileger.**

**Super utilisateur: su -**

**Changer user: sudo**

**Open root fichier: nano /etc/sudoers (to see all rights and change)**

**To see all file even hide: ls -a /file nom**

**To install must be copy .bashrc user to root**

**Copy cp /home/user/.bashrc /root/.bashrc**

**Commande: /etc/.bashrc**

**To enter nano commande for configuartion nanorc: nano /etc/nanorc (nano is editeur)**

**Change color: to enter nano /root/.bashrc and line 60 to change color(**

**Test : echo $PS1 (majuscule ps)**

**Commande for give debian offciel source install of all application : nano /etc/apt/sources.list , then type: deb <http://deb.debian.org/debian> bookwrom main contrib non-free non-free-firmware ,and search documentation officel**

**Then commande : apt update && apt upgrade -y**

**For to see all privilizes: sudo /etc/sudoers**

**For restart machine: Init 0**

**Cloning a virtual machine**

**Create a snapshot then edit machine->option->advanced->enable template**

**Then click on machine: manage->next->select snapshot->clone link**

**Change usernmae: su - then hostname deb1**

**Then enter into machine : nano /etc/hostname**

****

**Deb1**

**Vm->setting->network adapter->lansegment**

**Then**

**Service network manager: nano /etc/network/interfaces (configuration)**

**Commande : systemctl stop NetworkManager**

**Commande: systemctl disabled NetworkManager**

**Commande: ip a flush networkcard**

**Commande: systemctl restart networking.service**

**Check adress: ip a**

**Config deb2: echo deb2> /etc/hostname**

**Service network manager: nano /etc/network/interfaces**

**Vm->setting->network adapter->add 2 and 3 network adapter and select lansegment 1 for network adapter, lansegment 2 network adapter 2, network adapter nat**

**Configure network card: ..pic**

**Commande : systemctl stop NetworkManager**

**Commande: systemctl disabled NetworkManager**

**Commande: ip a flush networkcard**

**Commande: systemctl restart networking.service**

**Check adress: ip a**

**Routage for diefferent network (ex: deb1 and deb3):**

**Nano /etc/sysctl.conf**

**Active line 28 , by remove # //now different network communicate established.**

**Then for access internet**

**Sysctl --system //for display**

**Update commande: apt update && upgrade**

**Commande : apt install iptables iptables-persistent -y (next config No, NO, NO)**

**Do Nat: commande: iptables -t nat -A POSTROUTING -o ens37 -j MASQUERADE**

**To see all tables: iptables -t nat -L**

**This commande take iptables Iptables-save> /etc/iptables/rules.v4**

**Deb3:**

**Vm->setting->network adapter->lansegment**

**Then**

**Service network manager: nano /etc/network/interfaces (configuration)**

**Commande : systemctl stop NetworkManager**

**Commande: systemctl disabled NetworkManager**

**Commande: ip a flush networkcard**

**Commande: systemctl restart networking.service**

**Check adress: ip a**

**Some commands linux:**

**Useradd -D -s /bin/bash (D:default, s:shell)**

**Change mot de passe direct: echo ‘solo:pricesse’ | chpasswd //auto change et enter user solo)**

**Commmande2: export PS1=’\u-\h:\’**

**Commande: bash pour reviens initials**

**Pour un nouveau bash il faut aller sur linux home->.bashrc->ecrit**

**export PS1=’\u-\h:\$’;enregistre et ouvrer nouveu bash**

**Commande echo $SHELL(majuscule) il affiche nom shell lorsque appel d’authentification**

**On peux verifier dans quelle shell on travaille on peux lancer cette command echo $SHELL**

**Commande: ls (lister repartoie)**

**Commande: ls -l(pour plus d’information)**

**Coomande: ls -l -t(trier par plus dsc)**

**Commnade: ls -l -t -r(trier par asc)**

**Commande: ls -lrt (tous contenu recurrent)**

**Commande: ls -lrth (comprendre humanin)**

**Commande: type echo/type bash (affiche qui situ ou)**

**Commande: file /usr/bin/bash (commande pour voir file)**

**Commande: id (affiche pas ou situe la id)**

**Commande: echo $PATH(majuscule) (pour voir ou situé id la arborescence)**

**Commande: help (pour aide) pour particulier: help echo (aide a echo)**

**Commande interne et externe (interne commande en meme fichier, externe commande un autre fichier exemple: type id )**

**Commmande interne: help echo , commande externe: id --help**

**Plus details aide: man id (plus details)**

**3 aide resume: commande interne: help cmd, externe:cmd --help,plus details: man cmd**

**Avantages du Bash:**

**Commnade: history(toutes les commande deja passe)then pour recupere le commande on peux saisie: !numero\_commande**

**Commande : CTRL+R then cherche le mot (pour recherche une commande)**

**On peux creer un alisa aller go : home->bashrc>ecirt alias pour exemple: alias ll: ‘ls -lrtha’**

**Commande: alias (lister tous les alias)**

**FHS Linux: /boot : noyau linux**

**/bin: commandes**

**/etc : fichier de configuration**

**Commnade: /root compte super utilisateur**

**Ne pas stoker dans le fichier root**

**Le répartoire /home (il permet stoker les fichiers compte perso d’utilisateur)**

**Le commande: /proc et /sys (ces sont vitrine virtuelle elles ne sont pas dans le disque dur, linux maintien en mémoire.**

**Le répartoire /home equivalent /var/www pour le serveur web**

**A partir du racine du systeme /**

**Commande: /root (pour stocker les donées personelle d l’administrateur)**

**Root est le compte privilégie de linux, il doit etre reservé exclusivement aux taches administratives sur le systeme.**

**Le repartoire /usr ,qui contiens deux fichier /usr/bin(ce répertoire contient majoritairement les commandes à destination de tous les utilisateurs du système, privilégiés ou non), /usr/sbin(ce sont des commandes à nouveau à destination unique de l'administrateur, mais non critiques pour le bon fonctionnement du système)**

**Repartoire /var, /usr en lecteur seule.**

**Afficher les contenu de fichier par cat(concaténer):**

**Commande : type cat (affiche**

**Commande: cat /etc/passwd**

**Commande cat: cat -n /etc/os-release (affiche le contenu de fichier avec le numero)**

**Deux commande meme temps: cat -n /etc/os-release /etc/passwd**

**Less: permet de visualiser les fichier sous linux.**

**Utiliser les canaux de Linux: stdin, numero 0(standard input),sdtout numero 1,(standard output),stderr, numero 2(stardard error)**

**Grep : chercher le ligne de mot adjectivement. Exempmle: grep debian /etc/os-release**

**Grep -rnio debian /etc/\* (o isoler le pattern dans la ligne, n le numero ligne, r la partie recursive, i le pattern censive la case)**

**L’editeur sous linux: vi**

**Vi nom de fichier: exemple: vi /etc/fichier1**

**Lettre I pour inserer /ecrire**

**Then Exc ou Echap pour passer en mode commande etpuis ajoter type A pour append.**

**Ajouter une ligne au dessous rester en mode commande + Shift+O**

**Supprimer une ligne rester en mode command etpuis DD**

**Pour copie rester en mode commande + YY etpuis bouge cursor et P**

**Replace une lettre rester en mode commande etpuis slectioner la lettre type R**

**Pour mettre le numéro type: :set number, save and exit :wq! /:x!**

**Principle command linux: des I nodes: ls -li**

**Copie et coller un fichier: cp ficheir1 fichier2 , voir i nodes: ls -li (different i node fichier1 et fichier2)**

**Move un fichier : mv fichier1 fichier2 (voir ls -li meme indice fichier2 et fichier1)**

**Faire lien entre deux fichier ln: ln fichier1 fichier(voir nodes meme node que fichier1 et fichier**

**Faire lien symbolique ln -s: ln -s fichier1 noveaunomfichier**

**Difference entre lien dur et symbolic (symbolic un lien comme raccourci)**

**Droits des fichiers: chmod u+rwx fichier**

**chmod u=rwx fichier**

**puis chmod g=rx fichier**

**puis chmod o=r fichier**

**Chmod calculate rwx=r:2²=4,w:2=2,x:2°=1(total droit 4+2+1=7)**

Commande linux

Bin:biniare un lien raccourcir

Boot: pour grab2

Dev: device/material

Etc:dossier configuration

Home: repartoire toutes les utilisateur

Lib: toutes les libraires pour system

Media: toutes media

Var: toutes les logs stokcer dans le var

Create new user: adduser nom\_user

Display user: cat /etc/group

Connection with Alma Lunux: ssh NomdeMachine@ipaddress

Ssh: chiffrement asymetric(demande de deux cle public et prive)

Les tache a faire:

. Bridge

. Créer les users

. Envoyer ses clés au serveur

. Modifier le server pour qu’il n’accepts quels personne qui ont envoyé la clé

Projet Answer:

Create user: adduser name

Give mot de passe: passwd name\_user

Then Active Bridge: on virtual box network adpater Bridge

Create key via SSH

Then create key :

Summary of connect with server from windows if already have created (reste in root then..): dir ->cd .\.ssh\->dir->rm .\known\_hosts

Create hidden file in linux: mkdir .ssh

To see hidden file: ls -a

Create key: ssh-keygen -t rsa -b 2048(b for sizz, t for type cryptage)

Phasephrasse: a key for authentification.

It will create two key public and private. Now send public key to server

Send public key to client: commande: scp C:\Users\

scp C:\Users\moham\.ssh\id\_alma.pub mohammad@192.168.1.31:/home/mohammad/.ssh

(scp:exchange between two, first part is source, last part destination)

Renmae key: enter .ssh/ ->mv id\_alma.pub authorized\_keys

The target of Authorized key is connect with server distance without password, but this case must be connected with private key:

Commande: ssh -i C:\Users\moham\.ssh\id\_alma [mohammad@192.168.1.31](mailto:mohammad@192.168.1.31)

When block from server we can not connect with server, ssh [mohammad@192.168.1.31](mailto:mohammad@192.168.1.31)

Permission denied. If we connect ssh -i C:\Users\moham\.ssh\id\_alma [mohammad@192.168.1.31](mailto:mohammad@192.168.1.31) by this (means with key) then connected.

Configure server: /etc/ssh/sshd\_config

After change port of server: restart server: systemctl restart sshd

Commande: ssh -i C:\Users\moham\.ssh\id\_alma [mohammad@192.168.1.31](mailto:mohammad@192.168.1.31) -p 60001

<https://fr.wikiversity.org/wiki/Certification_Linux_LPI/Administrateur_syst%C3%A8me_d%C3%A9butant/Examen_101>

Procedure Password recovery:

1. Redemarer de l’equipement
2. Interommpe process de boot(bios->bout louder(GRUB: linux, MBR/GPT:window)->noyaux->unit.d
3. En tanque bip choisir le deuxieme option apres swap ligne de commande: rd.break enforcing=0 , ensuite CTRL+X, apres ls voir sysroot,
4. Apres commande:mount -o remount, rw /sysroot
5. Apres commande: chroot /sysroot
6. Then automatically enter into noyux sh-5.1(5.1 vesion)
7. Then commande : passwd
8. Give strong password
9. Then commande: touch /.autorelabel
10. Now start linux , commande: exit, exit (2 times)

Informatoin /etc/passwd, /etc/shadow

MFA(double authetifiation), Clé PKI

Respecter RGPD

## **Configurez les cartes réseaux**

Configurer le nom réseau du serveur:

Regarde hostname: cat /proc/sys/kernel/hostname

Autre facon par noyux: sysctl kernel.hostname

Details de hostname: hostnamectl

Set hostname: hostnamectl set-hostname sohrab

Entre dans le shell: bash

### ****Détectez les interfaces réseaux de votre système****

Commande : dmesg (display message du noyaux pour essayer de détecter les cartes réseaux reconnues par le noyau lors du démarrage)

Liste des networks: dmesg | grep Network

Liete des différents des reseaux: ls -l /sys/class/net/

### ****Configurez les cartes réseaux de manière dynamique****

Ifconfig commande ne pas utilsé depuis 2009, e profitant pas non plus des nouvelles fonctionnalités des noyaux Linux.

Commande ip adresse toutes les configuration reseaux: ip a

Commande lister carte reseaux: ip link show

Ajouter une adresse ip: 192.128.1.24/24(exemple) eth0 (ethernet)

Delete adresse ip: 192.128.1.24/24 eth0

### ****Configurez les routes et les passerelles****

Voir toutes les routes: ip route list

Pour enleve le route par defaut: ip route del default via default\_route\_ip

Ajoute une route: ip route add default via new\_route\_ip nom\_carte\_reseaux(beacause of route add in same network card)

Ajouter sous reseaux en meme carte reseaux: ip route add 192.138.1.0/24 via ip\_carte\_resaux

Verifiy route list: ip route list

Renseigner les fichiers permettant d'utiliser des services de résolutions de noms de domaine:

## **Connectez-vous à distance avec SSH**

Commande install: sudo apt-get install openssh-server

Enable ssh service: sudo systemctl enable sshd

Voir le server port et ip: ss -lptun

Verifer la service running ssh: systemctl status sshd

Install ssh-client:

Commande installation ssh-client: sudo apt-get install ssh-client

Verify ssh client: ssh

Connexion avec serveur: ssh username@ip\_adresse

Récupéré ip address: ifconfig

Déconnexion: exit

Peut connecter également par nom de machine:ssh [nom\_de\_machine(sohrab@..)](mailto:nom_de_machine(sohrab@..))

Déconnexion: exit

Visauliser le cle public en tant que connecter avec serveur: cat .ssh/known\_hosts

Génération de cle ssh:

Commande genere la cle: ssh-keygen -t rsa (-t rsa:algorithm,

## **Transférez des fichiers par le réseau**

deux logiciels les plus utilisés pour télécharger des fichiers en HTTP sur le réseau, à savoir wget et curl.

Install wget: sudo apt-get install wget

Telecharger fichier: wget lien

Install curl: sudo apt-get install curl

Telecharger fichier: curl lien

Summary of linux:

3 layer of OS Linux: Noyux, Driver, Libraray

Redhat: alma Linux, centos, Rochez linux,Fedora

Debian:une version ubuntu

When start OS , the OS start by BIOS, BIOS is checked of all hardare ex: cpu,memory etc.

He will boot, first boot CD/ROM, Disque dur, Support externe, Reseaux,

Disque dur: Sequence d’amorsage (MBR/GPT)

For linux GRUB( noyux)

2 memoire exitw: meomire morte: Flash, HDD, SSD,NVME, Meomoire vivre: RAM, CACHE(3 niveau cahce l1,l2,l3)

Disk:material en totale

Volume: file system

Partition: il y debut en fin

Start of VMWire 1990

Exemple of hypervisor type 1: proxmox, ESXI(vmwire),Citrix, hyper -v-microsoft,KVM

Installation Alma Linux:

New machine->typical->selection iso->if not detect iso , slelect redhat entreprise linux9->single files->next et disk personalisé for fichier /12g, /var2g,/home2g,/tmp 1g,/boot 1g, swap2g

Configure Windows server-22

Mettre l’infrastructure system composer en server: router, server DHCP, client

Give a ip static ip address , via carte reseaux->TCP IP: 192.168.10.1

Then rename PC parametre->renommer pc->redemarrer

Then Clone server vm->manage->clone->next->clone link->terminer

Configure network card in differents server. Right click on server on tehn settings: Srv2 : create a network adapter and then network adapter 1 for lan segment 1, network adapter 2 for lan segment 2. Then configure srv1(main window server22) lan segment 1 and srv2 lan segment 2

Then rename and configure Adresse ip each server:

For ping must be desactive fire-wall: start->outil de addministration->fire -feu windows defendeur->propriete windows defandeur->desactive 3 profile : profile domaine,profile prive, profie public.

Now can ping in same reseau , for ping another network need to cinfigure with router:

Sur srv2 install routage: Start menu->Gestionare de serveur->ajouiter des roles et des fonctionlalites->suivant->suivant->Access a distance->routage->suivant .....->installation

Start menu->outils administration->routage et acces a distance->Right click on srv2->configaure et routage et access a distance->Configuation personalise->suivant -> routage reseau

Serv1: install role Serveur DHCP: start menu->gestionare de serveur->ajouter role et fonctions->suviant ->.....->click on server DHCP->ajoter des fontionalite->suivant ...->installtion

Si notification for configuration pour server DHCP , installer et vailider

Start menu->outils adminstration->click on DHCP

For dynmique DHCP must create nouvelle entendu

Creation d’entendu

click droite ipv4->Nouvelle entendu->suivant..->nom de lan ->........

Routeur jamais fait les request de Broadcast, Routeur fait toujour unicast

Configure en ligne de commande:

Serveur

Installation vagrant and virtualbix

vagrant cloud(search image corresponding needs)

Vagrant: un outil permet de deployer des outils provider(fournisseur) appel automatisation.

Provider: CPU, RAM, I/O(stockage), Réseaux

Install : vagrant et virtual box (https://developer.hashicorp.com/vagrant/install windows amd64)

Then: crate a file : mkdir ubuntu, after enter into ubuntu: cd ubuntu

-> commande: vagrant init ubuntu/trusty64 (verify fichier ls)

->commande: vagrant up

->it will create ubuntu on virtaul box

->for destroy machine: vagrant destroy

Comment mettre la machine en bridge directement.

Vagrant up->vagrant ssh->sudo su->ifconfig

Now add linge: config.vm.network "public\_network" ->vagrant reload->now check in network adapter it will add a new bridge.

Comment choisir la taille/sizing

vagrant plugin install vagrant-disksize

Add file in vagrantfile: config.disksize.size = '50GB' then vagrant reload

Install vagrant et systeme sous Windows

Install vagrant: download vagrant from: <https://developer.hashicorp.com/vagrant/docs/providers/vmware/installation> ->vagrant utility download->download and install

Install vagrant plugin desktop by powershell: vagrant plugin install vagrant-vmware-desktop

Then mkdir AlmaLinux9 ->entree Almalinux9->chercher image sur vagrant cloud ->vagrant init nom\_image

Then -> vagrant up

->connecnt with machine(AlmaLinux9)->vagrant ssh (note in vmwire no physical interface and connection must be stay in file AlmaLinux9

Declarer variable in vb in this line: Vagrant.configure("2") do |vb|

Plus detials: https://developer.hashicorp.com/vagrant/docs/providers/configuration

Then configure vagrantmakefile a, then->

Vagrant global-status

We can run file for virtual box

Vagrant up --provider=virtualbox (by default vmwire)

Display of all network: window+R then ncap.cpl