**Week One Tasks**

* Started understanding about Hypervisors
* Understood why they are used and what is the working of Hypervisor.

To define it in a definition, it would go like

A hypervisor is a thin control layer that lets multiple VMs share the same physical hardware while believing they each own a computer.

A Hypervisor works by intercepting privileged operations, memory translations, virtualizing I/O and scheduling VCPUs so that each VM believes it has its own full computer, while everything is guarded by the hypervisor.

**Task: Install hypervisor, create 2 Ubuntu Jammy VMs (2 cores, 512 MB RAM, no GUI).  
Host: MacBook M1, macOS.  
Hypervisor used: UTM (QEMU).  
ISO used: Ubuntu 22.04.5 LTS Live Server (ARM64).**

**Task: Once the installation is done, create all the users who are a part of this email ( siddharth.suvalka , aniruddha.joshi ). Create a group called \*intern\* and add the users to that group.**

In one of the created VMs(Ubuntu1), two users are created ( siddharth.suvalka , aniruddha.joshi ), with designated passwords, and a group intern is created where the two users are added and the made the secondary group.

Commands used:

Creating user: useradd user  
Creating password: sudo passwd user  
Creating group: groupadd group  
adding users to group: sudo gpasswd --add user group

**Task: Add 2 disks of 10GB each to this VM. Create a LVM using these two disks and create an XFS filesystem on it. Mount this on /data directory and make sure that the mount persists across reboots**

On Ubuntu1(VM) added to 10GBs of disk(i.e Physical Volume)  
-- pvcreate --name

, and created a Volumegroup(VG) of 20GB.  
-- vgcreate --name

and then from the volumegroup created a partition of 16GB(Logical Volume).  
-- lvcreate --size 16G --name lvname vgname

Formated it in a xfs format type  
-- mkfs.xfs --path

and then got the uuid  
-- blkid --path

and make the volume mount persistant  
-- made changes in fstab file( /etc/fstab)  
-- /path /mountPoint xfs defaults 0 0

**Task: Add another 10GB disk to the VM. Add it to the existing LVM .**

added another disk of 10GB and created a pv  
--pvcreate /dev/vdd

added the physical volume to the VG  
-- vgextend myvg /dev/vdd

then added the extra space of 12 GB to the lvm  
-- lvextend --size +12G /dev/myvg/mylv1

formated the remaining space

-- xfs\_growfs /dev/myvg/mylv1

**Task: Run HTTP server on VM1 and access it from VM2**

Installed nginx on VM1.

Placed a simple index.html under the default web directory./var/www/index.html

Enabled and started nginx service so it runs on boot.  
systemctl enable nginx --now

Allowed HTTP traffic through the firewall.  
ufw allow 'Nginx Full'

From VM2, used curl with VM1’s IP to confirm access.

Task: Mount /var/log on a separate mount point.

Created a partition(vm2)

Formated it using XFS format  
/dev/vdb1

Created a temporary mount point and mounted it  
mkdir /mnt/log  
mount /dev/vdb1 /mnt/log

Mount and remount  
umount /mnt/log

mount /dev/vdb1 /var/log

**Task: The http server should only listen on the VMs IP and not localhost.**

Change the file of nginx:  
In /etc/nginx/sites-enabled/default  
Here in this file add  
listen 192.168.64.12:80;  
Reload nginx  
systemctl reload nginx