EXPIRIMENT 10:

Aim: Analysing network packet stream using tcpdump and wireshark. Perform basic network service tests using nc.

Solution :-

Hypervisor:

A hypervisor, also known as a virtual machine monitor or VMM, is software that creates and runs virtual machines (VMs). A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, such as memory and processing.

Hypervisors make it possible to use more of a system's available resources and provide greater IT mobility since the guest VMs are independent of the host hardware. This means they can be easily moved between different servers. Because multiple virtual machines can run off of one physical server with a hypervisor, a hypervisor reduces:

- Space
- Energy
- Maintenance requirements

There are two main hypervisor types, referred to as "Type 1" (or "bare metal") and "Type 2" (or "hosted"). A type 1 hypervisor acts like a lightweight operating system and runs directly on the host's hardware, while a type 2 hypervisor runs as a software layer on an operating system, like other computer programs.

The most commonly deployed type of hypervisor is the type 1 or bare-metal hypervisor, where virtualization software is installed directly on the hardware where the operating system is normally installed. Because bare-metal hypervisors are isolated from the attack-prone operating system, they are extremely secure. In addition, they generally perform better and more efficiently than hosted hypervisors. For these reasons, most enterprise companies choose bare-metal hypervisors for data centre computing needs.

Benefits of hypervisors

There are several benefits to using a hypervisor that hosts multiple virtual machines:

- Speed: Hypervisors allow virtual machines to be created instantly, unlike bare-metal servers. This makes it easier to provision resources as needed for dynamic workloads.

- Efficiency: Hypervisors that run several virtual machines on one physical machine’s resources also allow for more efficient utilization of one physical server. It is more cost- and energy-efficient to run several virtual machines on one physical machine than to run multiple underutilized physical machines for the same task.
- Flexibility: Bare-metal hypervisors allow operating systems and their associated applications to run on a variety of hardware types because the hypervisor separates the OS from the underlying hardware, so the software no longer relies on specific hardware devices or drivers.
- Portability: Hypervisors allow multiple operating systems to reside on the same physical server (host machine). Because the virtual machines that the hypervisor runs are independent from the physical machine, they are portable. IT teams can shift workloads and allocate networking, memory, storage and processing resources across multiple servers as needed, moving from machine to machine or platform to platform. When an application needs more processing power, the virtualization software allows it to seamlessly access additional machines.

Virtual machine:

A Virtual Machine (VM) is a compute resource that uses software instead of a physical computer to run programs and deploy apps. One or more virtual “guest” machines run on a physical “host” machine. Each virtual machine runs its own operating system and functions separately from the other VMs, even when they are all running on the same host. This means that, for example, a virtual MacOS virtual machine can run on a physical PC.

Virtual machine technology is used for many use cases across on-premises and cloud environments. More recently, public cloud services are using virtual machines to provide virtual application resources to multiple users at once, for even more cost efficient and flexible compute.

What are virtual machines used for?

Virtual machines (VMs) allow a business to run an operating system that behaves like a completely separate computer in an app window on a desktop. VMs may be deployed to accommodate different levels of processing power needs, to run software that requires a different operating system, or to test applications in a safe, sandboxed environment.

Virtual machines have historically been used for server virtualization, which enables IT teams to consolidate their computing resources and improve efficiency. Additionally, virtual machines can perform specific tasks considered too risky to carry out in a host environment, such as accessing virus-infected data or testing operating systems. Since the virtual machine is separated from the rest of the system, the software inside the virtual machine cannot tamper with the host computer.

Advantages of virtual machines

Virtual machines are easy to manage and maintain, and they offer several advantages over physical machines:

- VMs can run multiple operating system environments on a single physical computer, saving physical space, time and management costs.
- Virtual machines support legacy applications, reducing the cost of migrating to a new operating system. For example, a Linux virtual machine running a distribution of Linux as the guest operating system can exist on a host server that is running a non-Linux operating system, such as Windows.
- VMs can also provide integrated disaster recovery and application provisioning options.

KVM:

KVM stands for **Kernel-based Virtual Machine**. It allows the kernel to operate as a hypervisor. Moreover, it requires a processor with hardware virtualization extensions such as Intel VT or AMD-V. KVM was initially designed for x86 processors. Later, it was ported to processors such as ARM, PowerPC etc. Operating systems such as FreeBSD and illumos contain KVM as loadable kernel modules. Also, KVM provides hardware-assisted virtualization for many guest operating systems such as Linux, Solaris, Windows, Haiku and OS X. Furthermore, Android 2.2, Darwin 8.-.1 etc. work with some limitations.

Furthermore, some graphical management tools having KVM are as follows.

Kimchi is KVM’s web-based virtualization management tool

Virtual Machine Manager allows creating, editing, starting and stopping KVM based virtual machine

OpenQRM allows managing and controlling various data centre infrastructures.

GNOME Boxes – Gnome interface for handling libvirt guests on Linux.

oVirt is an open-source virtualization management tool for KVM

Proxmox Virtual Environment is an open-source virtualization management package with KVM and LXC.

Xen:

Xen or Xen Project is a type 1 hypervisor. It provides services to allow multiple computer operating systems to execute on the same computer hardware simultaneously.

In brief, KVM and Xen are two hypervisors written in C language. The main difference between KVM and Xen is that KVM is a virtualization module in Linux kernel that works similar to a hypervisor while Xen is a type 1 hypervisor that allows multiple operating systems to execute on the same computer hardware simultaneously.

Docker Container:

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.
Container images become containers at runtime and in the case of Docker containers - images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Docker containers that run on Docker Engine:

- **Standard:** Docker created the industry standard for containers, so they could be portable anywhere
- **Lightweight:** Containers share the machine’s OS system kernel and therefore do not require an OS per application, driving higher server efficiencies and reducing server and licensing costs
- **Secure:** Applications are safer in containers and Docker provides the strongest default isolation capabilities in the industry

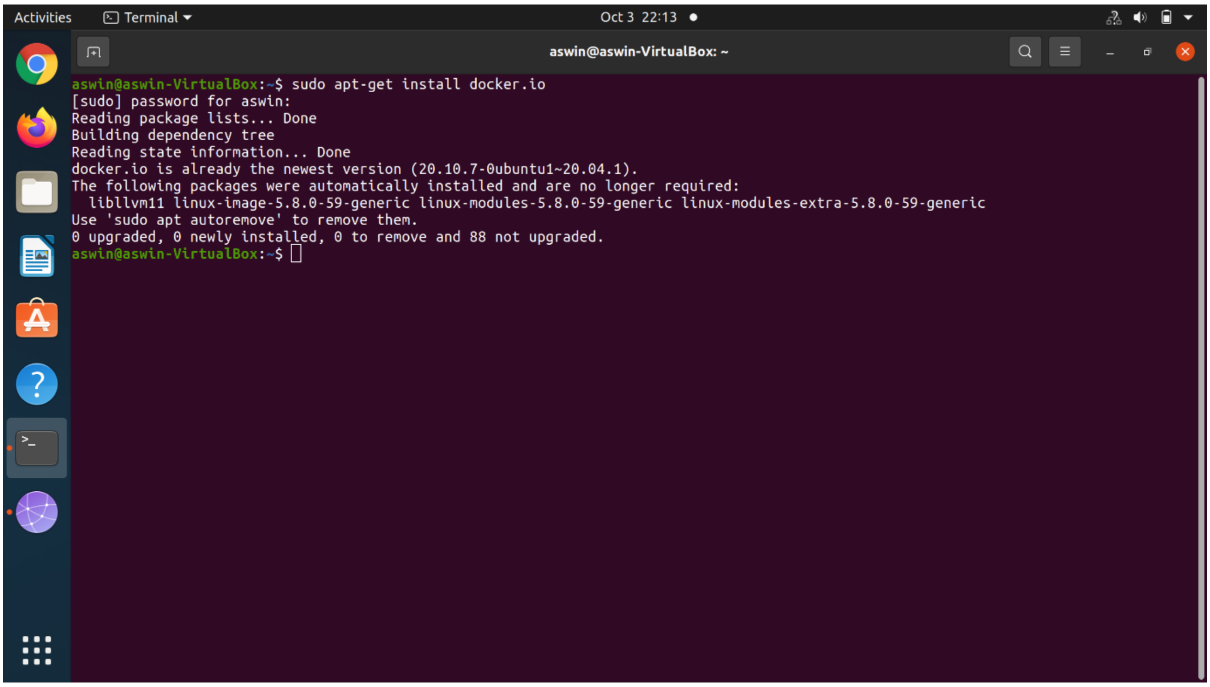
Installing Docker:

1) Updating the local repository

   sudo apt update

2) Installing Docker

   sudo apt install docker.io

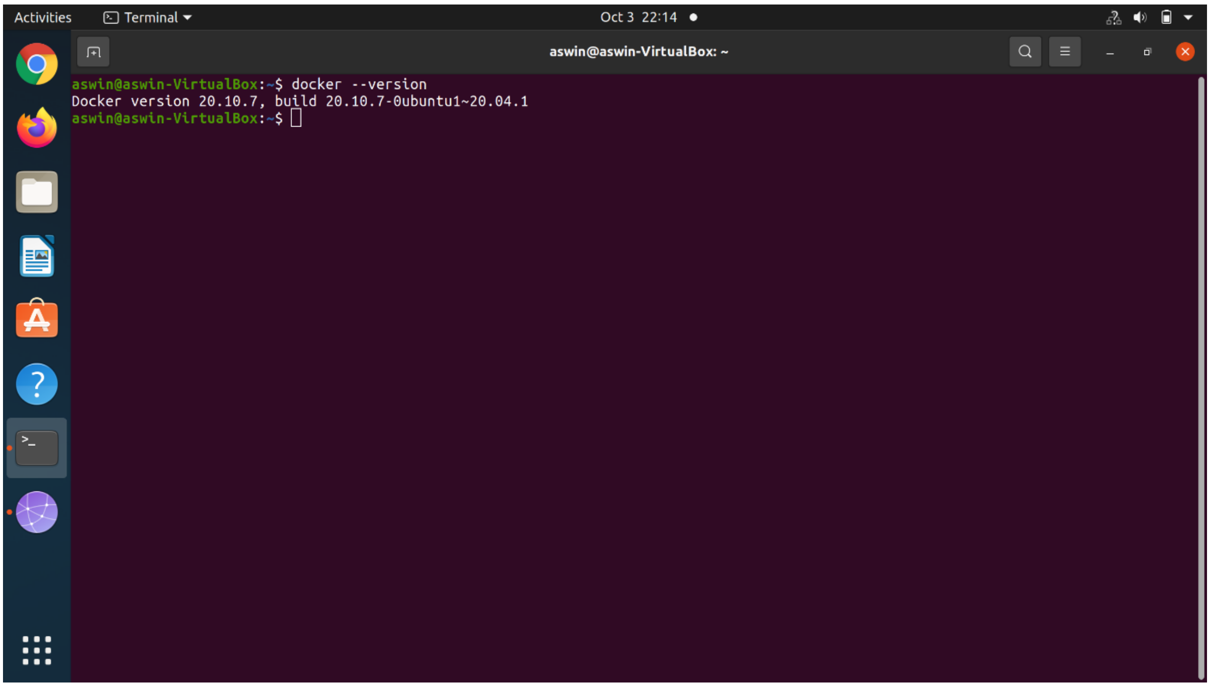


```
aswin@aswin-VirtualBox:~$ sudo apt-get install docker.io
[sudo] password for aswin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (20.10.7-0ubuntu1~20.04.1).
The following packages were automatically installed and are no longer required:
  libllvm11 linux-image-5.8.0-59-generic linux-modules-5.8.0-59-generic linux-modules-extra-5.8.0-59-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 88 not upgraded.
aswin@aswin-VirtualBox:~$ 
```

Type y and hit **Enter** to confirm the installation. Once the install is completed, the output notifies you Docker has been installed.

3) Checking docker installation

docker –version



```
aswin@aswin-VirtualBox:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu1~20.04.1
aswin@aswin-VirtualBox:~$ 
```

4) Starting docker service

* Start the Docker service by running:

sudo systemctl start docker

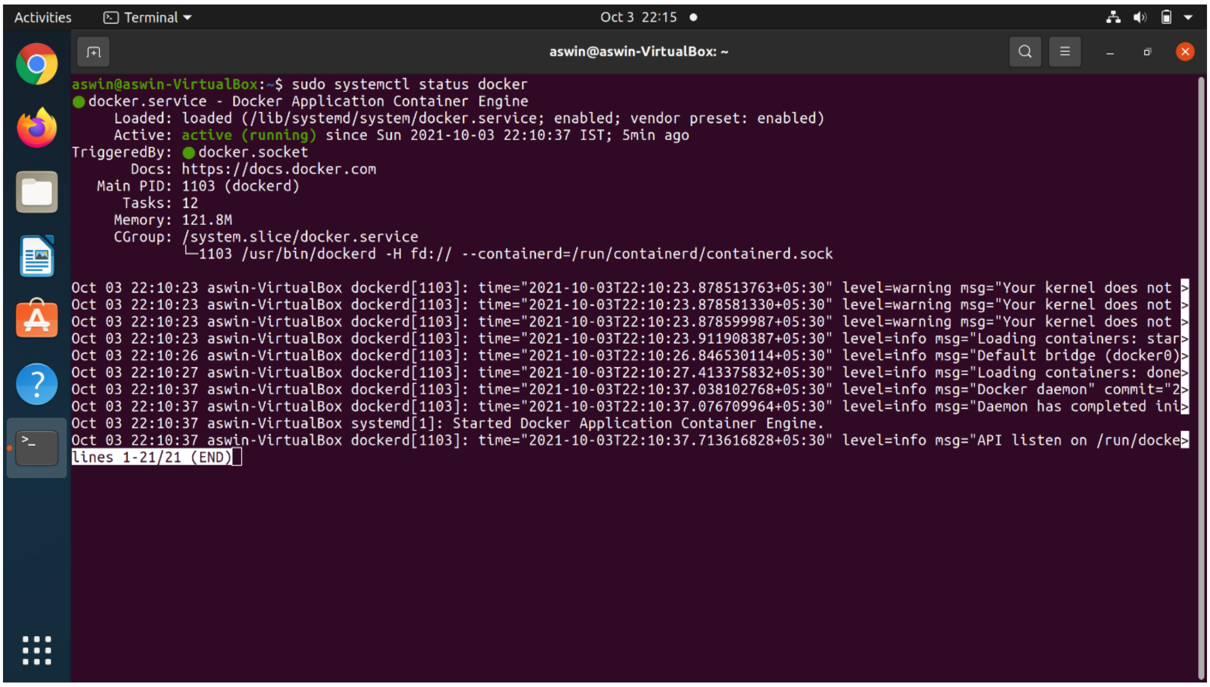
* Then, enable it to run at startup:

sudo systemctl enable docker

*To check the status of the service, run:

sudo systemctl status docker

The output should verify Docker is **active (running)**.



```
aswin@aswin-VirtualBox:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2021-10-03 22:10:37 IST; 5min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 1103 (dockerd)
      Tasks: 12
     Memory: 121.8M
     CGroup: /system.slice/docker.service
             └─1103 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Oct 03 22:10:23 aswin-VirtualBox dockerd[1103]: time="2021-10-03T22:10:23.878513763+05:30" level=warning msg="Your kernel does not >
Oct 03 22:10:23 aswin-VirtualBox dockerd[1103]: time="2021-10-03T22:10:23.878581330+05:30" level=warning msg="Your kernel does not >
Oct 03 22:10:23 aswin-VirtualBox dockerd[1103]: time="2021-10-03T22:10:23.878599987+05:30" level=warning msg="Your kernel does not >
Oct 03 22:10:23 aswin-VirtualBox dockerd[1103]: time="2021-10-03T22:10:23.911908387+05:30" level=info msg="Loading containers: star>
Oct 03 22:10:26 aswin-VirtualBox dockerd[1103]: time="2021-10-03T22:10:26.846530114+05:30" level=info msg="Default bridge (docker0)>
Oct 03 22:10:27 aswin-VirtualBox dockerd[1103]: time="2021-10-03T22:10:27.413375832+05:30" level=info msg="Loading containers: done>
Oct 03 22:10:37 aswin-VirtualBox dockerd[1103]: time="2021-10-03T22:10:37.038102768+05:30" level=info msg="Docker daemon" commit="2>
Oct 03 22:10:37 aswin-VirtualBox dockerd[1103]: time="2021-10-03T22:10:37.076709964+05:30" level=info msg="Daemon has completed ini>
Oct 03 22:10:37 aswin-VirtualBox systemd[1]: Started Docker Application Container Engine.
Oct 03 22:10:37 aswin-VirtualBox dockerd[1103]: time="2021-10-03T22:10:37.713616828+05:30" level=info msg="API listen on /run/docke>
lines 1-21/21 (END)
```

Use Docker on Ubuntu 20.04

The basic syntax for docker commands is:

sudo docker [option] [command] [argument]

Working With Docker Images

Docker images are files that contain the source code, libraries, dependencies, tools, and other files a container need. You can create Docker images with Dockerfiles or use existing ones available on Docker Hub.

To download a new Docker image, use the command:

```
docker pull [image_name]
```

If you don’t know the exact name of the image, search for it in Docker’s repository with:

```
docker search ubuntu
```

After working with Docker for some time, you will collect a local registry of images. Display a list of all Docker images on the system with:

```
docker images
```

Working With Docker Containers

Docker containers are isolated virtual environments that run based on the Docker image assigned to them.
To run a container based on an existing Docker image, use the command:

```
docker run [image_name]
```

Using the command above runs a container but doesn’t move you inside of it. To run a container in interactive mode and change to the container command prompt, run:

```
docker run -it [image_name]
```

Another useful docker command is listing all the containers on the system. To list all active containers, type:

```
docker container ps
```

To view all containers (active and inactive), run:

```
docker container ps -a
```