SYSTEM OPERATIONS LAB

Department of Computer Science and Engineering

Shri Govindram Seksaria Institute of Technology and Science, Indore

PREPARED BY:

0801CS201010 – Akshat Samdani
0801CS201048 – Keshav Bhattad
0801CS201050 – Keyush Parwal
0801CS201059 – Mohit Khatri
0801CS201080 – Samyak Jain

SUBMITTED TO:

Priyanka Bamne Ma’am

Table of Contents

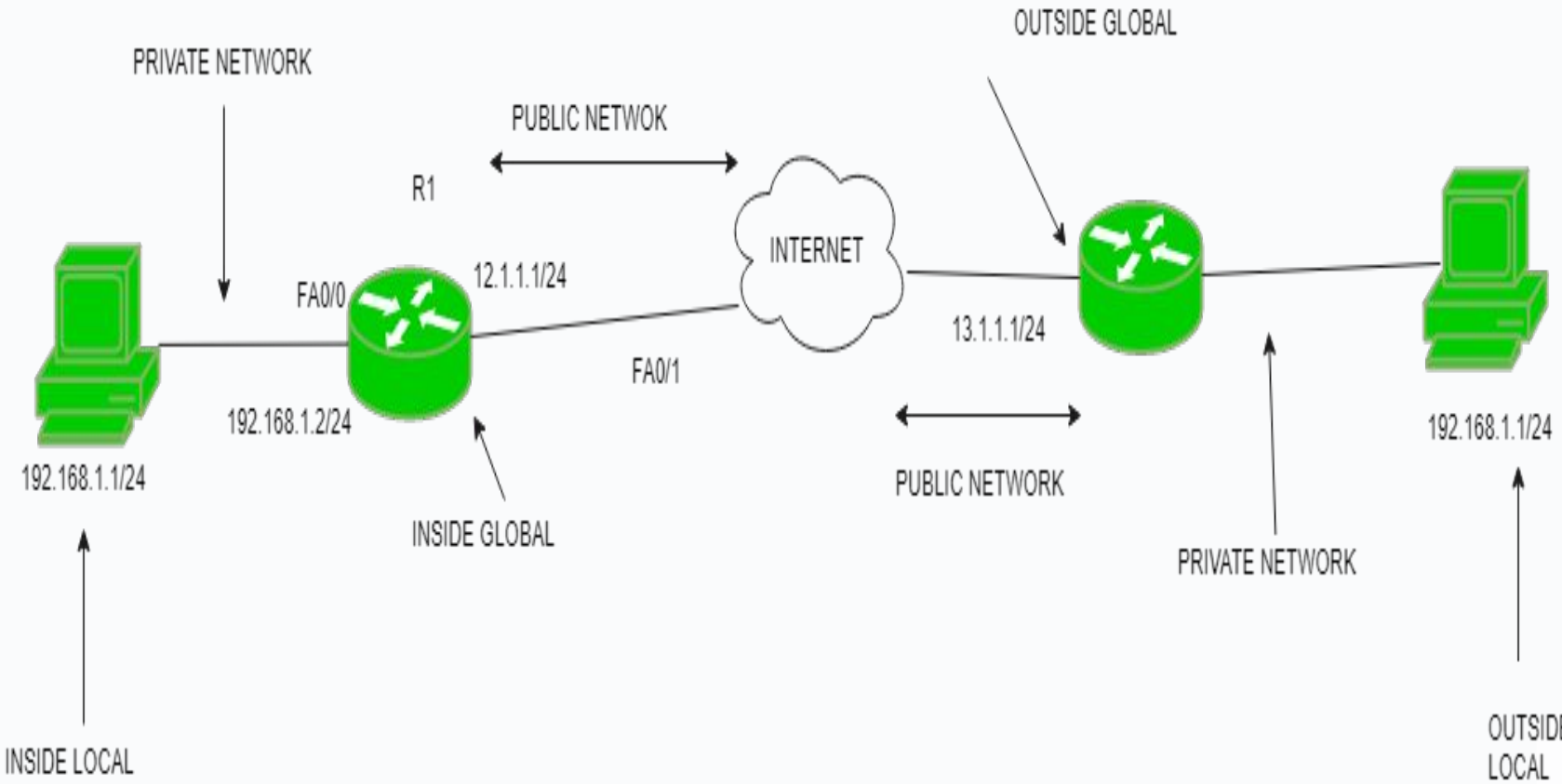
System Operation Lab (CO44401)

Network Address Translation (NAT)

Presented By:
Akshat Samdani (0801CS201010)
Keshav Bhattad (0801CS201048)
Keyush Parwal (0801CS201050)
Mohit Khatri (0801CS201059)
Samyak Jain(0801CS201080)

Network Address Translation(NAT)

Network Address Translation(NAT) is a process of assigning a unique public IP address to represent an entire group of computers. In Network Address Translation, a network device, typically a router which assigns a public address to one or more devices connected to a private network. Network address translation enables a single device to serve as an agent or intermediate between a local, private network and the internet, a public network. The basic objective of NAT is to reduce the number of public IP addresses in use for security and financial reasons.


OUTSIDE GLOBAL
PRIVATE NETWORK
PUBLIC NETWOK
R1
INTERNET
FA0/0
12.1.1.1/24
13.1.1.1/24
FA0/1
192.168.1.2/24
192.168.1.1/24
192.168.1.1/24
PUBLIC NETWORK
INSIDE GLOBAL
PRIVATE NETWORK
INSIDE LOCAL
OUTSIDE
LOCAL

Working of NAT

- **Private and Public IP Addresses:** In a typical network setup, devices within a private network (e.g., home or office network) are assigned private IP addresses. These private IP addresses are reserved for internal use and are not routable on the public internet. On the other hand, the router connecting the private network to the internet is assigned a public IP address that is unique and globally routable.

- **Address Translation:** When a device from the private network wants to communicate with a server on the internet (e.g., accessing a website), the NAT router plays a crucial role in translating the private IP address of the device into its own public IP address.

- **Outgoing Traffic:** When a device initiates an outgoing connection to an external server, the NAT router intercepts the outgoing packet and modifies the source IP address in the packet header. The source IP address is changed from the original private IP address of the device to the public IP address of the NAT router.

- **Port Number Mapping:** Additionally, the NAT router also assigns a unique port number to each connection in its translation table. This process is called "Port Address Translation" (PAT) or "Network Address and Port Translation" (NAPT). The combination of the public IP address and the assigned port number differentiates between multiple connections from various devices within the private network.

- **Routing on the Internet:** The modified packet with the public IP address (and assigned port number) is then forwarded to the destination server on the internet. The server sees the public IP address of the NAT router as the source of the request and sends the response back to that IP address.

- **Incoming Traffic:** When the response reaches the NAT router, it looks into its translation table to determine which internal device the response is intended for. The router uses the port number and the source IP address of the incoming packet to identify the corresponding private IP address of the device that initiated the outgoing request.

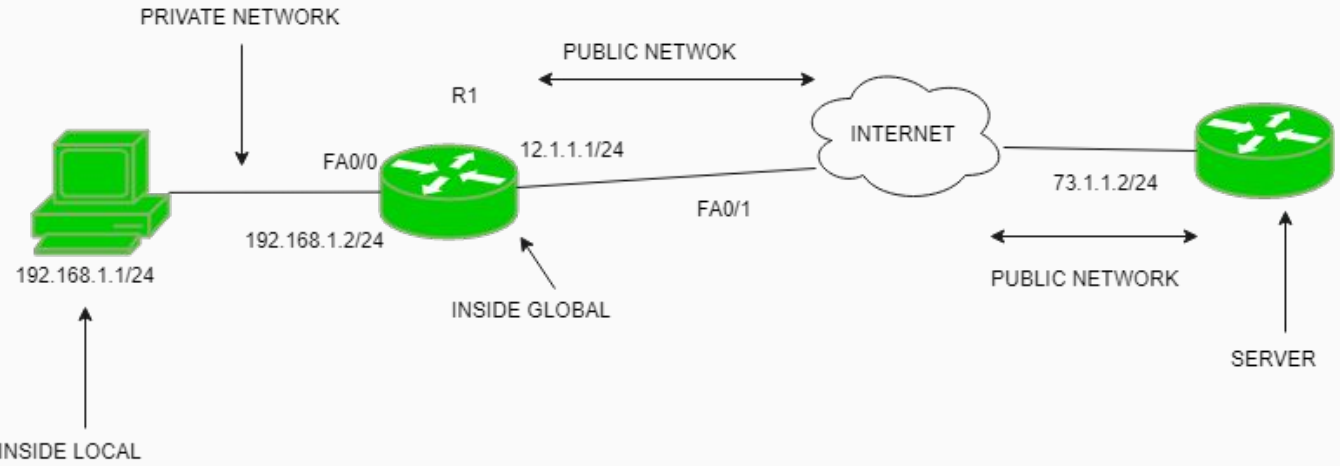
- **Stateful NAT:** The NAT router maintains a stateful connection table that keeps track of the active connections. This stateful approach ensures that only incoming packets belonging to established connections are allowed to pass through the NAT router, while any unsolicited traffic from the internet is blocked.

Types of Network Address Translation (NAT)

1.) Static NAT
2.) Dynamic NAT
3.) Port Address Translation (PAT)

Static NAT

In this, a single unregistered (Private) IP address is mapped with a legally registered (Public) IP address i.e one-to-one mapping between local and global addresses. This is generally used for Web hosting. These are not used in organizations as there are many devices that will need Internet access and to provide Internet access, a public IP address is needed.



Here is a small topology in which there is PC having IP address 192.168.1.1/24, Router R1 having IP address 192.168.1.2/24 on interface fa0/0, 12.1.1.1/24 on fa0/1, and server having IP address 73.1.1.2/24.

Now, inside local and inside global are shown in the figure. Configuring the static NAT through command IP nat inside source static INSIDE_LOCAL_IP_ADDRESS INSIDE_GLOBAL_IP_ADDRESS.

```
R1(config)# ip nat inside source static 192.168.1.1 12.1.1.1
```

Dynamic NAT

In this type of NAT, an unregistered IP address is translated into a registered (Public) IP address from a pool of public IP addresses. If the IP address of the pool is not free, then the packet will be dropped as only a fixed number of private IP addresses can be translated to public addresses.

Now, enabling Dynamic NAT:

```
R1(config)# ip nat inside source list 1 pool pool1
```

At last, we have to configure router interfaces as inside or outside.

```
R1(config)# int fa0/0
R1(config-if)# ip nat inside
R1(config)# int fa0/1
R1(config-if)# ip nat outside
```

Port Address Translation

This is also known as NAT overload. In this, many local (private) IP addresses can be translated to a single registered IP address. Port numbers are used to distinguish the traffic i.e., which traffic belongs to which IP address. This is most frequently used as it is cost-effective as thousands of users can be connected to the Internet by using only one real global (public) IP address.

Advantages of NAT

1.) NAT conserves legally registered IP addresses.

2.) It provides privacy as the device's IP address, sending and receiving the traffic, will be hidden.

3.) Eliminates address renumbering when a network evolves.

Disadvantages of NAT

1. Translation results in switching path delays.

2. Certain applications will not function while NAT is enabled.

3. Also, the router being a network layer device, should not tamper with port numbers(transport layer) but it has to do so because of NAT.

System Operation Lab (CO44401)

Dynamic Host Configuration Protocol (DHCP)

Presented By:
Akshat Samdani (0801CS201010)
Keshav Bhattad (0801CS201048)
Keyush Parwal (0801CS201050)
Mohit Khatri (0801CS201059)
Samyak Jain(0801CS201080)

Dynamic Host Configuration Protocol (DHCP)

DHCP is a network management protocol used in networks to dynamically assign IP addresses and other network configuration information like default gateway, mask, DNS server address, etc. It is an application layer protocol.

Dynamic Host Configuration Protocol (DHCP)

- **IP address:** This is the most important piece of information that DHCP provides. It is the unique address that the client will use to communicate on the network.
- **Subnet mask:** This is used to determine which part of the IP address is the network address and which part is the host address.
- **Default gateway:** This is the IP address of the router that the client should use to send traffic to other networks.
- **DNS servers:** These are the IP addresses of the DNS servers that the client should use to resolve hostnames to IP addresses.
- **Lease time:** This is the amount of time that the client is allowed to use the IP address before it has to renew the lease.

Components of DHCP

Components of DHCP

The main components of DHCP include:

- **DHCP Server:** DHCP Server is basically a server that holds IP Addresses and other information related to configuration.
- **DHCP Client:** It is basically a device that receives configuration information from the server. It can be a mobile, laptop, computer, or any other electronic device that requires a connection.
- **DHCP Relay:** DHCP relays basically work as a communication channel between DHCP Client and Server.
- **IP Address Pool:** It is the pool or container of IP Addresses possessed by the DHCP Server. It has a range of addresses that can be allocated to devices.

Components of DHCP

- **Subnets:** Subnets are smaller portions of the IP network partitioned to keep networks under control.
- **Lease:** It is simply the time that how long the information received from the server is valid, in case of expiration of the lease, the tenant must have to re-assign the lease.
- **DNS Servers:** DHCP servers can also provide DNS (Domain Name System) server information to DHCP clients, allowing them to resolve domain names to IP addresses.
- **Default Gateway:** DHCP servers can also provide information about the default gateway, which is the device that packets are sent to when the destination is outside the local network.

Components of DHCP

- **Options:** DHCP servers can provide additional configuration options to clients, such as the subnet mask, domain name, and time server information.
- **Renewal:** DHCP clients can request to renew their lease before it expires to ensure that they continue to have a valid IP address and configuration information.
- **Failover:** DHCP servers can be configured for failover, where two servers work together to provide redundancy and ensure that clients can always obtain an IP address and configuration information, even if one server goes down.
- **Dynamic Updates:** DHCP servers can also be configured to dynamically update DNS records with the IP address of DHCP clients, allowing for easier management of network resources.
- **Audit Logging:** DHCP servers can keep audit logs of all DHCP transactions, providing administrators with visibility into which devices are using which IP addresses and when leases are being assigned or renewed.

Working of Dynamic Host Configuration Protocol

Automatic IP address assignment process undergoes four message exchange. These messages are abbreviated as Discover, Offer, Request & Acknowledgement (DORA). The following table gives the detail of these four messages.

1.) Discover: This is message sent by DHCP client to discover a DHCP server.
2.) Offer: Sent by DHCP server to lease unique IP address and other parameters needed to client.
3.) Request: Sent by DHCP client asking server to lease parameters listed in Offer message.
4.) Acknowledgement: Sent by DHCP server to assign IP address, mask, default router & DNS server address to client.

Working of Dynamic Host Configuration Protocol

There are other 4 messages that are used in the process:

1.) **DHCP negative acknowledgment message** : Whenever a DHCP server receives a request for an IP address that is invalid according to the scopes that are configured, it sends a DHCP Nak message to the client.
2.) **DHCP decline:** If the DHCP client determines the offered configuration parameters are different or invalid, it sends a DHCP decline message to the server.
3.) **DHCP release:** A DHCP client sends a DHCP release packet to the server to release the IP address and cancel any remaining lease time.
4.) **DHCP inform:** If a client address has obtained an IP address manually then the client uses DHCP information to obtain other local configuration parameters,.

DHCP Related Commands

- **ipconfig /release:** This command releases the DHCP-assigned IP address for all network interfaces on a Windows computer. For example: `ipconfig /release`

- **ipconfig /renew:** This command requests a new DHCP lease and obtains a new IP address configuration for all network interfaces on a Windows computer. For example: `ipconfig /renew`

- **ipconfig /all:** This command displays detailed information about all network interfaces on a Windows computer, including the DHCP configuration. For example: `ipconfig /all`

DHCP Related Commands

- **netsh interface ip show config**: This command provides detailed information about the IP configuration for all network interfaces on a Windows computer, including DHCP-related settings. For example: netsh interface ip show config.
- **netsh interface ip set address "Wi-Fi" dhcp:** This command helps in enabling DHCP

Advantages of DHCP

1.) Centralized management of IP addresses.

2.) Reuse of IP addresses reduces the total number of IP addresses that are required.

3.) Simple reconfiguration of the IP address space on the DHCP server without needing to reconfigure each client.

4.) With the help of DHCP, easy handling of new users and the reuse of IP addresses can be achieved.

Command Screenshots (server side)

Updating

```
cndc-27@cndc27-OptiPlex-3050-AIO:~$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRe
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InF
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports I
Get:5 http://security.ubuntu.com/ubuntu jammy-security/mair
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/mai
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/mai
Get:8 http://security.ubuntu.com/ubuntu jammy-security/mair
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-updates/mai
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-updates/ma
```

Installing net-tools

```
cndc-27@cndc27-OptiPlex-3050-AIO:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 189 not upgraded.
Need to get 204 kB of archives.
After this operation, 819 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+
git20181103.0eebece-1ubuntu5 [204 kB]
Fetched 204 kB in 1s (154 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 203869 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ..
.
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Processing triggers for man-db (2.10.2-1) ...
```

Seeing the IP Address and Netmask

```
Processing triggers for man-db (2.10.2-1) ...
cndc-27@cndc27-OptiPlex-3050-AIO:~$ ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.100.27  netmask 255.255.0.0  broadcast 10.1.255.255
        inet6 fe80::e1ad:4021:71df:6602  prefixlen 64  scopeid 0x20<link>
        ether d8:9e:f3:95:93:43  txqueuelen 1000  (Ethernet)
        RX packets 12387  bytes 11690717 (11.6 MB)
        RX errors 0  dropped 58  overruns 0  frame 0
        TX packets 5935  bytes 463305 (463.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2899  bytes 236823 (236.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2899  bytes 236823 (236.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Installing isc-dhcp-server

```
cndc-27@cndc27-OptiPlex-3050-AIO:~$ sudo apt-get install isc-dhcp-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libirs-export161 libisccfg-export163
Suggested packages:
  isc-dhcp-server-ldap policycoreutils
The following NEW packages will be installed:
  isc-dhcp-server libirs-export161 libisccfg-export163
0 upgraded, 3 newly installed, 0 to remove and 189 not upgraded.
Need to get 529 kB of archives.
After this operation, 1,546 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libisccfg-export163 amd64 1:9.11.19+dfsg-2.1ub
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libirs-export161 amd64 1:9.11.19+dfsg-2.1ubunt
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 isc-dhcp-server amd64 4.4.1-2.3ubuntu2
Fetched 529 kB in 2s (315 kB/s)
Preconfiguring packages
```

Setting interface as enp1s0 int INTERFACESv4

```
cndc-27@cndc27-OptiPlex-3050-AIO:~$ ^C
cndc-27@cndc27-OptiPlex-3050-AIO:~$ sudo nano /etc/default/isc-dhcp-server
```

```
INTERFACESv4="ens1s0"
INTERFACESv6=""
```

Configuring the range of ip addresses that needs to be allotted

```
cndc-27@cndc27-OptiPlex-3050-AIO:/etc/dhcp$ ls
ddns-keys  dhclient.conf          dhclient-exit-hooks.d  dhcpd.conf
debug      dhclient-enter-hooks.d  dhcpd6.conf
cndc-27@cndc27-OptiPlex-3050-AIO:/etc/dhcp$ sudo nano /etc/dhcp/dhcpd.conf
```

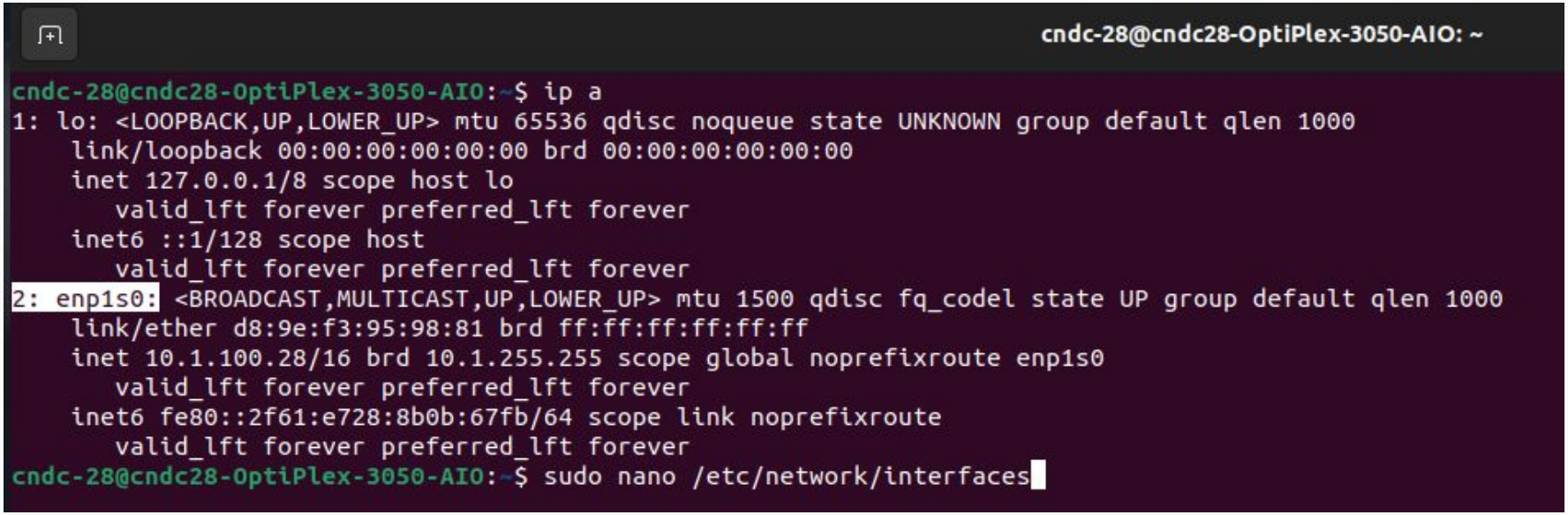
```
subnet 10.1.0.0 netmask 255.255.0.0 {
  range 10.1.0.150 10.1.0.200;
  #option domain-name-servers ns1.internal.example.org;
  #option domain-name "internal.example.org";
  option subnet-mask 255.255.255.224;
  option routers 10.5.5.1;
  option broadcast-address 10.5.5.31;
  default-lease-time 600;
  max-lease-time 7200;
}
```

Starting the server and checking the status

```
cndc-27@cndc27-OptiPlex-3050-AIO:/etc/dhcp$ sudo systemctl start isc-dhcp-server
cndc-27@cndc27-OptiPlex-3050-AIO:/etc/dhcp$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-09-12 17:17:15 IST; 11min ago
       Docs: man:dhcpd(8)
   Main PID: 6901 (dhcpd)
      Tasks: 4 (limit: 9093)
     Memory: 4.9M
        CPU: 51ms
     CGroup: /system.slice/isc-dhcp-server.service
             └─6901 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf enp1s0

Sep 12 17:24:03 cndc27-OptiPlex-3050-AIO dhcpd[6901]: DHCPREQUEST for 10.1.0.100 from d8:9e:f3:95:98:81 via enp1s0: unknown lease 10.1.0.100.
Sep 12 17:24:05 cndc27-OptiPlex-3050-AIO dhcpd[6901]: reuse_lease: lease age 52 (secs) under 25% threshold, reply with unaltered, existing lease for 10.1.0.150
Sep 12 17:24:05 cndc27-OptiPlex-3050-AIO dhcpd[6901]: DHCPREQUEST for 10.1.0.150 from d8:9e:f3:95:98:81 (cndc28-OptiPlex-3050-AIO) via enp1s0
Sep 12 17:24:05 cndc27-OptiPlex-3050-AIO dhcpd[6901]: DHCPACK on 10.1.0.150 to d8:9e:f3:95:98:81 (cndc28-OptiPlex-3050-AIO) via enp1s0
Sep 12 17:24:26 cndc27-OptiPlex-3050-AIO dhcpd[6901]: reuse_lease: lease age 73 (secs) under 25% threshold, reply with unaltered, existing lease for 10.1.0.150
Sep 12 17:24:26 cndc27-OptiPlex-3050-AIO dhcpd[6901]: DHCPDISCOVER from d8:9e:f3:95:98:81 (cndc28-OptiPlex-3050-AIO) via enp1s0
Sep 12 17:24:26 cndc27-OptiPlex-3050-AIO dhcpd[6901]: DHCPOFFER on 10.1.0.150 to d8:9e:f3:95:98:81 (cndc28-OptiPlex-3050-AIO) via enp1s0
Sep 12 17:24:26 cndc27-OptiPlex-3050-AIO dhcpd[6901]: reuse_lease: lease age 73 (secs) under 25% threshold, reply with unaltered, existing lease for 10.1.0.150
Sep 12 17:24:26 cndc27-OptiPlex-3050-AIO dhcpd[6901]: DHCPREQUEST for 10.1.0.150 (10.1.100.27) from d8:9e:f3:95:98:81 (cndc28-OptiPlex-3050-AIO) via enp1s0
Sep 12 17:24:26 cndc27-OptiPlex-3050-AIO dhcpd[6901]: DHCPACK on 10.1.0.150 to d8:9e:f3:95:98:81 (cndc28-OptiPlex-3050-AIO) via enp1s0
```

Command Screenshots (Client Side)

Identifying Interface



Command Screenshots (Client Side)

Setting up DHCP on interface

Dynamic Allocation of IP

```
cndc-28@cndc28-OptiPlex-3050-AIO:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether d8:9e:f3:95:98:81 brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.150/27 brd 10.5.5.31 scope global dynamic enp1s0
       valid_lft 500sec preferred_lft 500sec
    inet6 fe80::2f61:e728:8b0b:67fb/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

```
cndc-28@cndc28-OptiPlex-3050-AIO:~$ ip a show dev enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether d8:9e:f3:95:98:81 brd ff:ff:ff:ff:ff:ff
    inet 10.1.100.28/16 brd 10.1.255.255 scope global noprefixroute enp1s0
       valid_lft forever preferred_lft forever
    inet 10.1.0.100/27 brd 10.5.5.31 scope global dynamic enp1s0
       valid_lft 397sec preferred_lft 397sec
    inet6 fe80::2f61:e728:8b0b:67fb/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

System Operation Lab (CO44401)

Makefile

Presented By:
Akshat Samdani (0801CS201010)
Keshav Bhattad (0801CS201048)
Keyush Parwal (0801CS201050)
Mohit Khatri (0801CS201059)
Samyak Jain(0801CS201080)

Introduction

Makefile is a tool to simplify or to organize code for compilation. It is a set of commands (similar to terminal commands) with variable names and targets to create object file and to remove them. In a single make file we can create multiple targets to compile and to remove object, binary files.

Definition

A Makefile is a simple text file that contains a set of rules used by the make utility to build executable targets from source files and directories. The basic structure of a Makefile consists of a set of target rules, dependencies, and commands. Each rule defines a target, its dependencies, and the commands to build the target.

Makefile Basics

- **Target:** A target is the name of a file that is generated by a rule. A target rule has the following structure:

  target:dependencies
  commands

  The target is followed by a colon, and then a list of its dependencies separated by spaces. The commands to build the target are indented by a tab character.

Makefile Basics

- **Dependencies:** Dependencies are files that the target depends on. If any of the dependencies have been updated more recently than the target, the target is considered out of date and needs to be rebuilt. Dependencies can be other targets or source files.
- **Commands:** Commands are shell commands that are executed when the target needs to be built. The commands are indented by a tab character and must be preceded by a blank line.

Makefile Implementation in C

header.h

```
1    void hello(void);
2    void add(int a, int b);
```

hello.c

```c
#include <stdio.h>
#include "header.h"
#include <unistd.h>

void hello()
{
    printf("\nI am in hello.c\n");
    printf("PID of hello.c is %d\n", getpid());
}
```

add.c

```c
#include<stdio.h>
#include"header.h"

void add(int a, int b)
{
    int c;
    c = a+b;

    printf("\nAddition = %d", c);
}
```

```c
#include<stdio.h>
#include"header.h"

void main()
{
    printf("\n I am in main.c\n");
    printf("\nCalling hello function");
    hello();
    printf("\nCalling add function");
    add(3,5);
    printf("\nBack in main.c");
}
```

main.c

Printing Output

```
PS C:\Users\samyak\Desktop\SOL\Makefile_Presentation> gcc main.c hello.c add.c
PS C:\Users\samyak\Desktop\SOL\Makefile_Presentation> ./a.exe

 I am in main.c

Calling hello function
I am in hello.c
PID of hello.c is 17624

Calling add function
Addition = 8
Back in main.c
```

Creating Makefile

Makefile

```
1   final:
2       gcc main.c add.c hello.c -o final
```

Printing output using make command

```
PS C:\Users\samyak\Desktop\SOL\Makefile_Presentation> make
gcc main.c add.c hello.c -o final
PS C:\Users\samyak\Desktop\SOL\Makefile_Presentation> ./final

 I am in main.c

Calling hello function
I am in hello.c
PID of hello.c is 11160

Calling add function
Addition = 8
Back in main.c
```

Using variables

```
cc = gcc
final:
    $(cc) main.c add.c hello.c -o final
```

Using multiple targets

```
cc = gcc

final: main.o add.o hello.o
    $(cc) main.o hello.o add.o -o final

main.o: main.c header.h
    $(cc) -c main.c

hello.o: hello.c header.h
    $(cc) -c hello.c

add.o: add.c header.h
    $(cc) -c add.c
```

Printing Output

```
PS C:\Users\samyak\Desktop\SOL\Makefile_Presentation> make final
gcc main.o hello.o add.o -o final
PS C:\Users\samyak\Desktop\SOL\Makefile_Presentation> ./final

 I am in main.c

Calling hello function
I am in hello.c
PID of hello.c is 20732

Calling add function
Addition = 8
Back in main.c
PS C:\Users\samyak\Desktop\SOL\Makefile_Presentation>
```

Makefile Implementation in C++

header.h

```h
#ifndef HEADER_H
#define HEADER_H

class func
{
    public:
      int fibonacci(int n);
};

#endif
```

function.cpp

```cpp
#include <bits/stdc++.h>
#include "header.h"
using namespace std;

int func::fibonacci(int n)
{
    int a, b, c;
    a = 0;
    b = 1;
    if (n == 0)
        return 0;
    for (int i = 2; i <= n; i++)
    {
        c = a + b;
        a = b;
        b = c;
    }
    return b;
}
```

main.cpp

```cpp
#include <bits/stdc++.h>
#include "header.h"

using namespace std;
int main()
{
    int num;
    cin >> num;
    func obj;
    cout << obj.fibonacci(num);
}
```

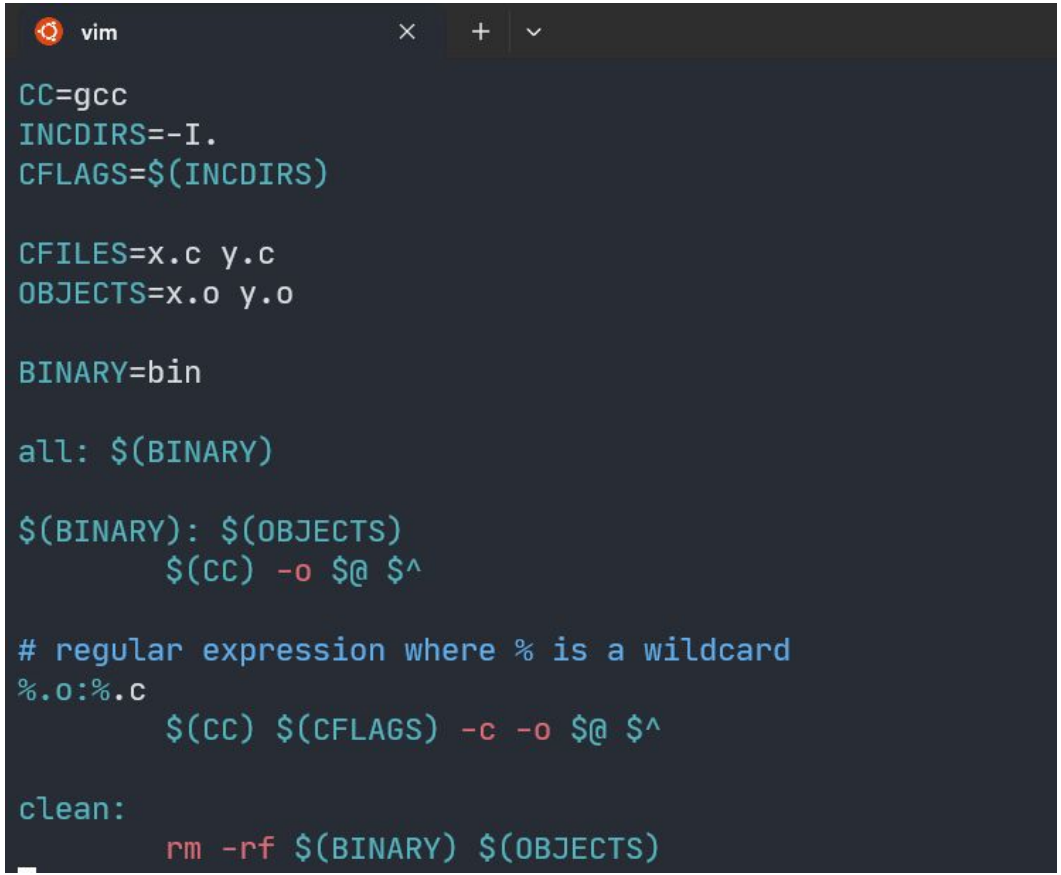
Makefile

```
output: main.o function.o
    g++ main.o function.o -o output

main.o: main.cpp header.h
    g++ -c main.cpp

function.o: function.cpp header.h
    g++ -c function.cpp
```

Output

```
PS C:\Users\samyak\Desktop\SOL\Makefile\Makefile_CPP> make
g++ -c main.cpp
g++ -c function.cpp
g++ main.o function.o -o output
PS C:\Users\samyak\Desktop\SOL\Makefile\Makefile_CPP> ./output
8
21
```

Makefile Implementation in Ubuntu


vim
CC=gcc
INCDIRS=-I.
CFLAGS=$(INCDIRS)
CFILES=x.c y.c
OBJECTS=x.o y.o
BINARY=bin
all: $(BINARY)
$(BINARY): $(OBJECTS)
$(CC) -o $@ $^
regular expression where % is a wildcard
%.o:%.c
$(CC) $(CFLAGS) -c -o $@ $^
clean:
rm -rf $(BINARY) $(OBJECTS)

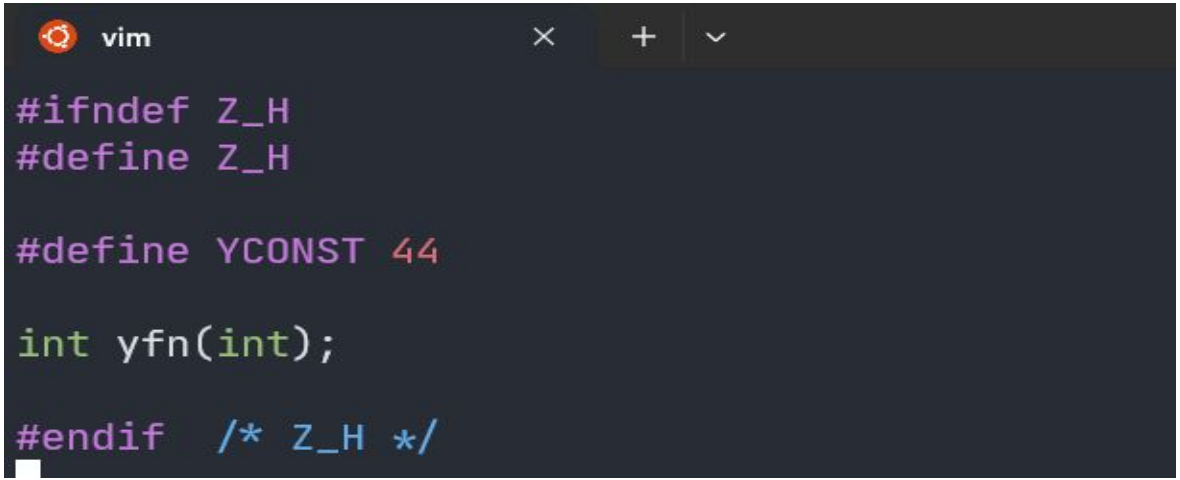
x.c


vim
#include <stdio.h>
#include <z.h>
int
xfn(void)
{
return yfn(1);
}
int
main(void)
{
printf("%d\n", xfn());
return 0;
}

y.c


vim
#include <z.h>
int
yfn(int a)
{
return (a + 1) * YCONST;
}

z.h


vim
#ifndef Z_H
#define Z_H
#define YCONST 44
int yfn(int);
#endif /* Z_H */

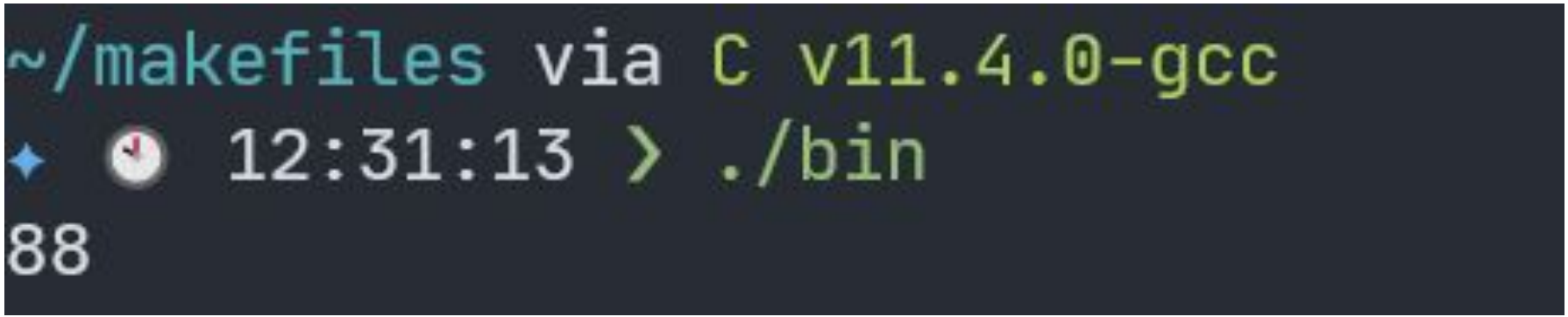
Output

```
~/makefiles via C v11.4.0-gcc
✦ 🕛 12:30:40 ❯ ls
Makefile  x.c  y.c  z.h
```

```
~/makefiles via C v11.4.0-gcc
✦ 🕛 12:30:42 ❯ make all
gcc -I. -c -o x.o x.c
gcc -I. -c -o y.o y.c
gcc -o bin x.o y.o
```

```
~/makefiles via C v11.4.0-gcc
✦ 🕛 12:30:56 ❯ ls
Makefile  bin  x.c  x.o  y.c  y.o  z.h
```

Ubuntu



Advantages

- It makes codes more concise and clear to read and debug.
- No need to compile entire program every time whenever you make a change to a functionality or a class. Makefile will automatically compile only those files where change has occurred.
- Generally, in long codes or projects, Makefile is widely used in order to present project in more systematic and efficient way.

System Operation Lab (CO44401)

Cron Job

Presented By:
Akshat Samdani (0801CS201010)
Keshav Bhattad (0801CS201048)
Keyush Parwal (0801CS201050)
Mohit Khatri (0801CS201059)
Samyak Jain(0801CS201080)

What is a cron?

Cron is a job scheduling utility present in Unix like systems. The cron reads the **crontab** (cron tables) for running predefined scripts.

By using a specific syntax, you can configure a cron job to schedule scripts or other commands to run automatically.

For individual users, the cron service checks the following file: **/var/spool/cron/crontabs**

Cron job example

```
* * * * * sh /path/to/script.sh
```

In the above example,

- * * * * * represents minute(s) hour(s) day(s) month(s) weekday(s), respectively.

	VALUE	DESCRIPTION
Minutes	0-59	Command would be executed at the specific minute.
Hours	0-23	Command would be executed at the specific hour.
Days	1-31	Commands would be executed in these days of the months.
Months	1-12	The month in which tasks need to be executed.
Weekdays	0-6	Days of the week where commands would run. Here, 0 is Sunday.

- `sh` represents that the script is a bash script and should be run from `/bin/bash`.

Application of cron jobs

- Scheduled backups or data synchronization between different systems.
- Regular system maintenance tasks like log rotation or database cleanup.
- Periodic data aggregation or reporting tasks.
- Automated scaling of resources based on time-specific requirements.

Cron Job Syntax

Cron job syntax

Crontabs use the following flags for adding and listing cron jobs.

- `crontab -e` : edits crontab entries to add, delete, or edit cron jobs.
- `crontab -l` : list all the cron jobs for the current user.
- `crontab -u username -l` : list another user's crons.
- `crontab -u username -e` : edit another user's crons.

Cron Job Syntax

```
*    *    *    *    *   sh /path/to/script/script.sh
|    |    |    |    |                    |
|    |    |    |    |        Command or Script to Execute
|    |    |    |    |
|    |    |    |    |
|    |    |    |    |
|    |    |    |    | Day of the Week(0-6)
|    |    |    |    |
|    |    |    | Month of the Year(1-12)
|    |    |    |
|    |    | Day of the Month(1-31)
|    |    |
|    | Hour(0-23)
|    |
Min(0-59)
```

Cron Job Examples

Below are some examples of scheduling cron jobs.

SCHEDULE	SCHEDULED VALUE
5 0 * 8 *	At 00:05 in August.
5 4 * * 6	At 04:05 on Saturday.
0 22 * * 1-5	At 22:00 on every day-of-week from Monday through Friday.

Cron Job Examples

Setting frequency

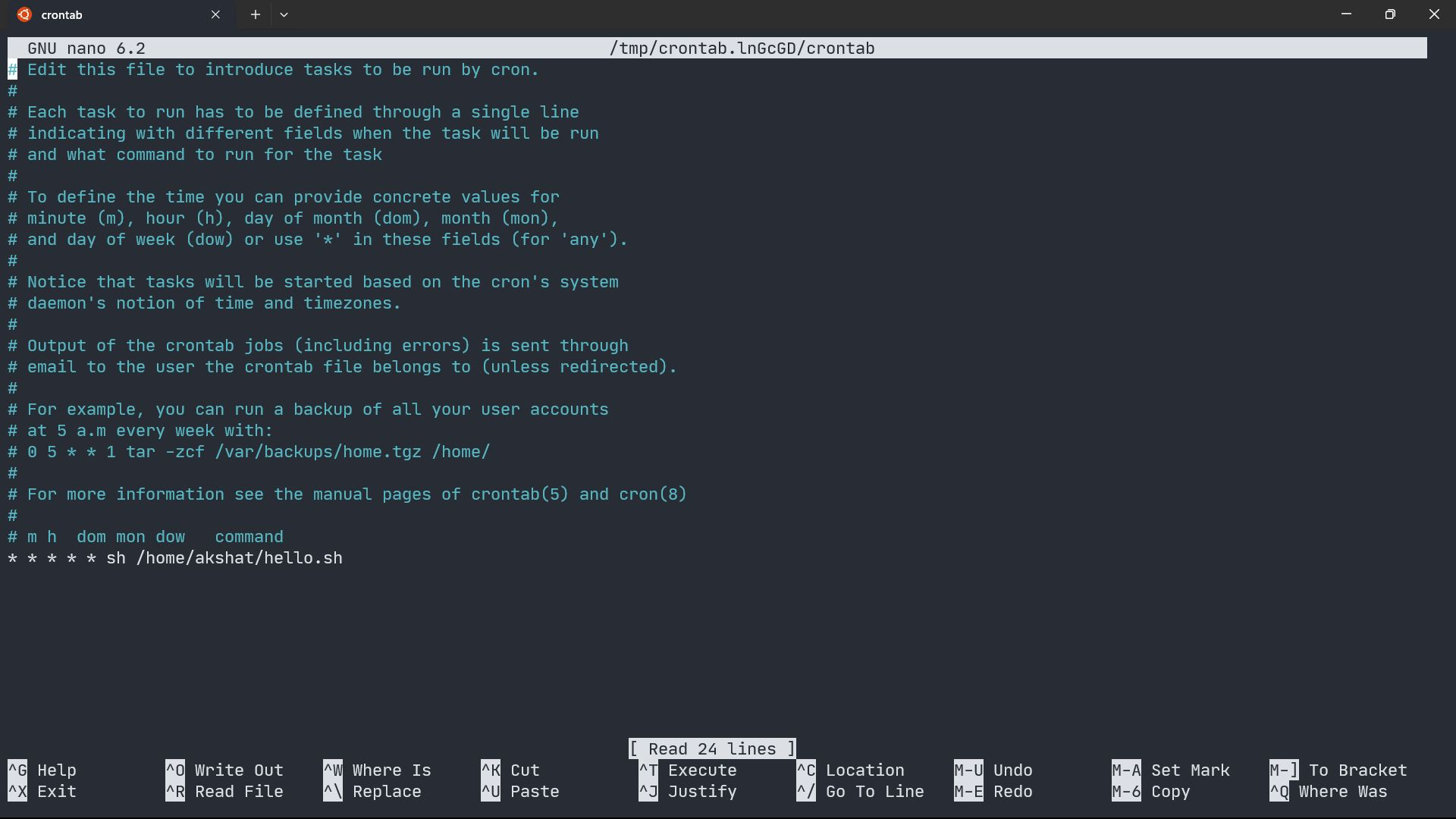
*/10 * * * * -> This will run every 10 minutes

Setting range

5-10 * * * * -> This will run every hour for 5-10 minutes

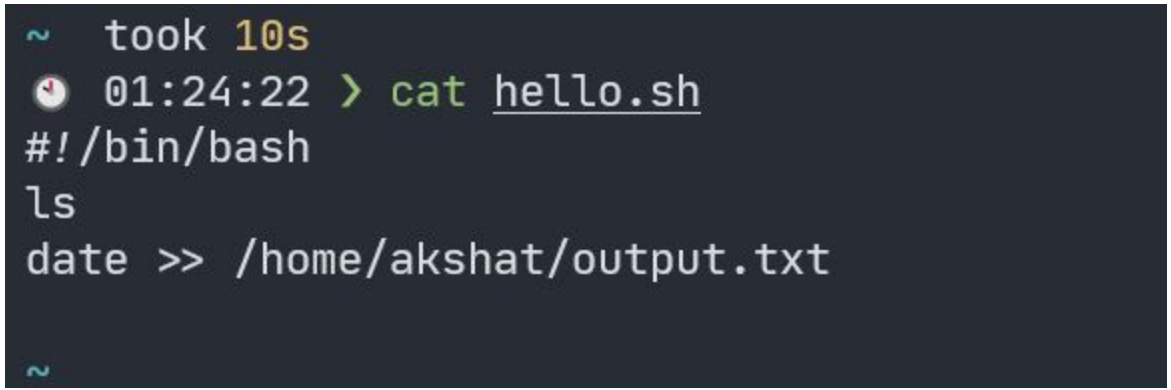
Cro

```
~
 01:22:14 ❯ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
* * * * * sh /home/akshat/hello.sh
```


crontab
GNU nano 6.2 /tmp/crontab.lnGcGD/crontab
Edit this file to introduce tasks to be run by cron.
#
Each task to run has to be defined through a single line
indicating with different fields when the task will be run
and what command to run for the task
#
To define the time you can provide concrete values for
minute (m), hour (h), day of month (dom), month (mon),
and day of week (dow) or use '*' in these fields (for 'any').
#
Notice that tasks will be started based on the cron's system
daemon's notion of time and timezones.
#
Output of the crontab jobs (including errors) is sent through
email to the user the crontab file belongs to (unless redirected).
#
For example, you can run a backup of all your user accounts
at 5 a.m every week with:
0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
For more information see the manual pages of crontab(5) and cron(8)
#
m h dom mon dow command
* * * * * sh /home/akshat/hello.sh
[Read 24 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^Q Where Was

```
~
🕙 01:22:48 ❯ crontab -e
No modification made

~  took 16s
🕙 01:23:33 ❯ cat output.txt
Wed Oct 11 14:30:01 IST 2023
Wed Oct 11 14:31:01 IST 2023
Wed Oct 11 14:32:01 IST 2023
Wed Oct 11 14:33:01 IST 2023
Wed Oct 11 14:34:01 IST 2023
Wed Oct 11 14:35:01 IST 2023
Wed Oct 11 14:36:01 IST 2023
Wed Oct 11 14:37:01 IST 2023
Wed Oct 11 14:38:01 IST 2023
Wed Oct 11 14:39:01 IST 2023
Wed Oct 11 14:40:01 IST 2023
Wed Oct 11 14:41:01 IST 2023
Wed Oct 11 14:42:01 IST 2023
Wed Oct 11 14:43:01 IST 2023
Wed Oct 11 14:44:01 IST 2023
Wed Oct 11 14:45:01 IST 2023
Wed Oct 11 14:46:01 IST 2023
Wed Oct 11 14:47:01 IST 2023
Wed Oct 11 14:48:01 IST 2023
Wed Oct 11 14:49:01 IST 2023
Wed Oct 11 14:50:01 IST 2023
Wed Oct 11 14:51:01 IST 2023
Wed Oct 11 14:52:01 IST 2023
Wed Oct 11 14:53:01 IST 2023
Wed Oct 11 14:54:01 IST 2023
Wed Oct 11 14:55:01 IST 2023
Wed Oct 11 14:56:01 IST 2023
Wed Oct 11 14:57:01 IST 2023
Wed Oct 11 14:58:01 IST 2023
```

Cronjob


~ took 10s
01:24:22 ❯ cat hello.sh
#!/bin/bash
ls
date >> /home/akshat/output.txt
~

Setup of MYSQL server with permission of remote login from a specific IP.

Installation Mysql Server and Mysql Client

```
puneet@puneet: $ sudo apt install  mysql-server mysql-client -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58 libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0
  libchromaprint1 libcodec2-1.0 libdav1d5 libflite1 libgme0 libgsm1 libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm15 libmfx1 libmysofa1 libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55
  librabbitmq4 librubberband2 libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 libsratom-0-0 libsrt1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5 libudfread0 libva-drm2 libva-wayland2 libva-x11-2
  libva2 libvdpau1 libvidstab1.1 libx265-199 libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0 mesa-va-drivers mesa-vdpau-drivers pocketsphinx-en-us systemd-hwe-hwdb va-driver-all vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7 libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi0ldbl libhtml-template-perl libmecab2 libprotobuf-lite23 mecab-ipadic
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libipc-sharedcache-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7 libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi0ldbl libhtml-template-perl libmecab2 libprotobuf-lite23 mecab-ipadic
  mecab-ipadic-utf8 mecab-utils mysql-client mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 21 newly installed, 0 to remove and 472 not upgraded.
Need to get 29.3 MB of archives.
After this operation, 242 MB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+1.0.8 [7,212 B]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-core-8.0 amd64 8.0.34-0ubuntu0.22.04.1 [2,754 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-8.0 amd64 8.0.34-0ubuntu0.22.04.1 [22.7 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libaio1 amd64 0.3.112-13build1 [7,176 B]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libevent-core-2.1-7 amd64 2.1.12-stable-1build3 [93.9 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libevent-pthreads-2.1-7 amd64 2.1.12-stable-1build3 [7,642 B]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libmecab2 amd64 0.996-14build9 [199 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libprotobuf-lite23 amd64 3.12.4-1ubuntu7.22.04.1 [209 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-server-core-8.0 amd64 8.0.34-0ubuntu0.22.04.1 [17.5 MB]
17% [9 mysql-server-core-8.0 299 kB/17.5 MB 2%]                                                                                                                3,388 B/s 2h 6min 33sS
18% [9 mysql-server-core-8.0 619 kB/17.5 MB 4%]                                                                                                                 12.6 kB/s 33min 41sS
```

Installation Complete

```
reading /usr/share/mecab/dic/ipadic/Noun.org.csv ... 16668
reading /usr/share/mecab/dic/ipadic/Conjunction.csv ... 171
reading /usr/share/mecab/dic/ipadic/Noun.name.csv ... 34202
reading /usr/share/mecab/dic/ipadic/Verb.csv ... 130750
reading /usr/share/mecab/dic/ipadic/Filler.csv ... 19
reading /usr/share/mecab/dic/ipadic/Noun.proper.csv ... 27328
reading /usr/share/mecab/dic/ipadic/Interjection.csv ... 252
reading /usr/share/mecab/dic/ipadic/Postp-col.csv ... 91
reading /usr/share/mecab/dic/ipadic/Suffix.csv ... 1393
reading /usr/share/mecab/dic/ipadic/Noun.csv ... 60477
reading /usr/share/mecab/dic/ipadic/Symbol.csv ... 208
reading /usr/share/mecab/dic/ipadic/Noun.others.csv ... 151
reading /usr/share/mecab/dic/ipadic/Others.csv ... 2
reading /usr/share/mecab/dic/ipadic/Noun.verbal.csv ... 12146
reading /usr/share/mecab/dic/ipadic/Noun.number.csv ... 42
reading /usr/share/mecab/dic/ipadic/Noun.adverbal.csv ... 795
reading /usr/share/mecab/dic/ipadic/Postp.csv ... 146
reading /usr/share/mecab/dic/ipadic/Adj.csv ... 27210
reading /usr/share/mecab/dic/ipadic/Prefix.csv ... 221
reading /usr/share/mecab/dic/ipadic/Adverb.csv ... 3032
reading /usr/share/mecab/dic/ipadic/Auxil.csv ... 199
reading /usr/share/mecab/dic/ipadic/Noun.nai.csv ... 42
emitting double-array: 100% |###########################################|
reading /usr/share/mecab/dic/ipadic/matrix.def ... 1316x1316
emitting matrix      : 100% |###########################################|

done!
update-alternatives: using /var/lib/mecab/dic/ipadic-utf8 to provide /var/lib/mecab/dic/debian (mecab-dictionary) in auto mode
Setting up mysql-server-8.0 (8.0.34-0ubuntu0.22.04.1) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Renaming removed key_buffer and myisam-recover options (if present)
mysqld will log errors to /var/log/mysql/error.log
mysqld is running as pid 39469
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib/systemd/system/mysql.service.
Setting up mysql-server (8.0.34-0ubuntu0.22.04.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
puneet@puneet:~$
```

Start : It tells systemctl to initiate or start a specific service.

Enable : the system will create symbolic links in the appropriate directories to ensure that the service starts automatically during system boot

```
puneet@puneet:~$ sudo systemctl start mysql
[sudo] password for puneet:
puneet@puneet:~$ sudo systemctl enable mysql
Synchronizing state of mysql.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mysql
puneet@puneet:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-09-05 22:36:28 IST; 7min ago
   Main PID: 39753 (mysqld)
     Status: "Server is operational"
      Tasks: 37 (limit: 3725)
     Memory: 362.6M
        CPU: 3.598s
     CGroup: /system.slice/mysql.service
             └─39753 /usr/sbin/mysqld

Sep 05 22:36:27 puneet systemd[1]: Starting MySQL Community Server...
Sep 05 22:36:28 puneet systemd[1]: Started MySQL Community Server.
puneet@puneet:~$
```

`mysql_secure_installation` is a shell script developed for securing the MySQL server installation on Unix systems. The script configures security settings and allows you to:

- Set a password for root accounts
- Remove the root accounts accessible from outside the localhost.
- Remove anonymous-user accounts.
- Delete the test database, accessible by anonymous users.
- Reload the user privileges tables.

```
puneet@puneet:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.
The 'validate_password' component is installed on the server.
The subsequent steps will run with the existing configuration
of the component.

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y
Success.
```

Open Mysql

```
puneet@puneet:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.34-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.01 sec)

mysql>
```

Comment Bind Address

```
puneet@puneet:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
[sudo] password for puneet:
puneet@puneet:~$
```

```
  GNU nano 6.2                                  /etc/mysql/mysql.conf.d/mysqld.cnf *
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user            = mysql
# pid-file      = /var/run/mysqld/mysqld.pid
# socket        = /var/run/mysqld/mysqld.sock
# port          = 3306
# datadir       = /var/lib/mysql


# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir                = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address           = 127.0.0.1
#mysqlx-bind-address    = 127.0.0.1
#
# * Fine Tuning
#
```

Create New User on Mysql Server

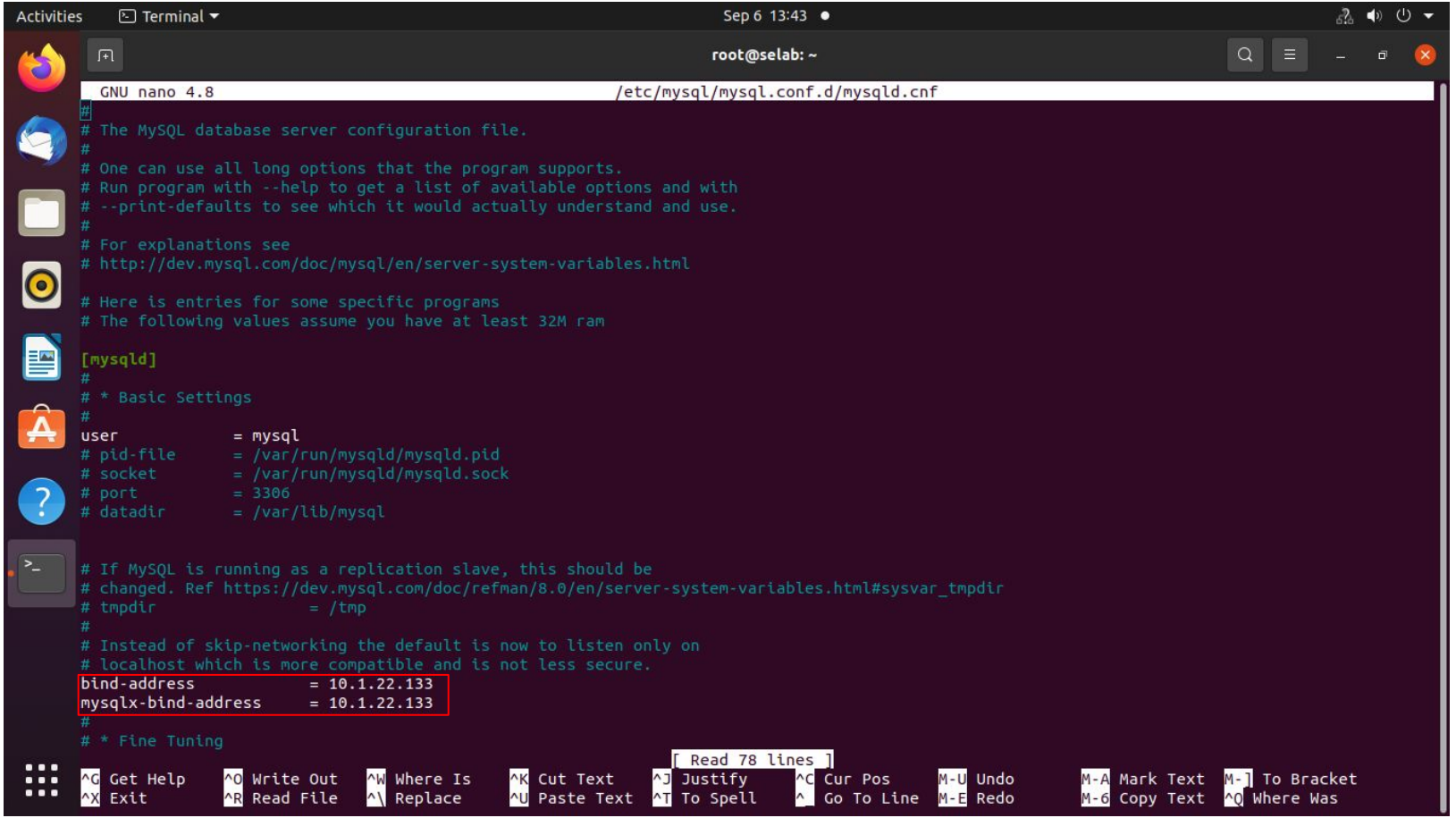
```
mysql> create user "sanyam"@"%" identified by "Sanyam$2222";
Query OK, 0 rows affected (0.16 sec)
```

```
mysql> create user "group1"@"%" identified by "Group$2222"
    -> ;
Query OK, 0 rows affected (0.13 sec)
```

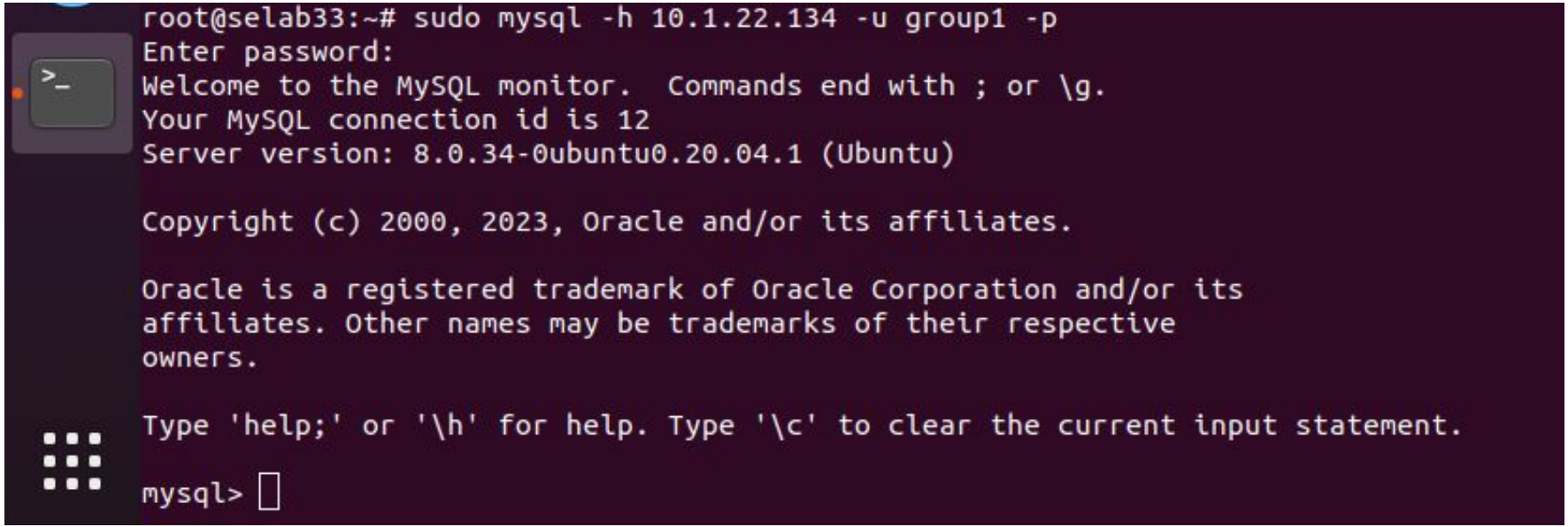
Remote access of database

```
root@selab33:~# sudo mysql -h 10.1.22.134 -u sanyam -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.34-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> exit
```

Set up for specific ip address



Remote access for specific ip address



Execution of jobs at remote host using scripts and without entering password from keyboard

- First we install ssh on server

```
root@selab:~# sudo apt install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-client openssh-server openssh-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 5 newly installed, 0 to remove and 108 not upgraded.
Need to get 1,365 kB of archives.
After this operation, 6,139 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssh-client amd64 1:8.2p
1-4ubuntu0.9 [671 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssh-sftp-server amd64 1
:8.2p1-4ubuntu0.9 [51.7 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssh-server amd64 1:8.2p
```

- Start SSH on Server
- Check SSH Status

```
root@selab:~# service ssh start
root@selab:~# service ssh status
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: e>
     Active: active (running) since Wed 2023-09-13 10:15:51 IST; 1min 42s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 5526 (sshd)
      Tasks: 1 (limit: 9318)
     Memory: 1.0M
     CGroup: /system.slice/ssh.service
             └─5526 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Sep 13 10:15:51 selab systemd[1]: Starting OpenBSD Secure Shell server...
Sep 13 10:15:51 selab sshd[5526]: Server listening on 0.0.0.0 port 22.
Sep 13 10:15:51 selab sshd[5526]: Server listening on :: port 22.
Sep 13 10:15:51 selab systemd[1]: Started OpenBSD Secure Shell server.
Sep 13 10:17:09 selab sshd[7179]: Connection closed by 127.0.0.1 port 55902 [pr>
lines 1-16/16 (END)...skipping...
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-09-13 10:15:51 IST; 1min 42s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 5526 (sshd)
```

- Check IP address of server

```
student@selab:~$ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.22.134  netmask 255.255.0.0  broadcast 10.1.255.255
        inet6 fe80::5de4:8e64:3548:ac7e  prefixlen 64  scopeid 0x20<link>
        ether 64:51:06:4e:c4:19  txqueuelen 1000  (Ethernet)
        RX packets 279917  bytes 414732705 (414.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 131594  bytes 9262675 (9.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xf7c00000-f7c20000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2632  bytes 192831 (192.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2632  bytes 192831 (192.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Generate Public and Private key

```
student@selab33:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/student/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/student/.ssh/id_rsa
Your public key has been saved in /home/student/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:by7xEZEpNm1rLnM+eb+2OvQgNvCiJTaj/Hcxgz8Sneg student@selab33
The key's randomart image is:
+---[RSA 3072]----+
|          . o    |
|         + *     |
|        . + o    |
|         . +     |
|         SO o    |
|       = X.# o   |
|    . o B %oX o  |
|     o . EoO o o |
|      ... +.+.=+o|
+----[SHA256]-----+
```

- Copy Public key on server

- Connect to Client

```
student@selab33:~$ ssh student@10.1.22.134
The authenticity of host '10.1.22.134 (10.1.22.134)' can't be established.
ECDSA key fingerprint is SHA256:ygxJoTwA8+rBcYwYYUoXt/Q8s4kybo2fN4BLMEOXCog.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.1.22.134' (ECDSA) to the list of known hosts.
student@10.1.22.134's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

97 updates can be applied immediately.
38 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Your Hardware Enablement Stack (HWE) is supported until April 2025.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

- Create Script on Server

```
student@selab:~$ cat script.sh
cat: script.sh: No such file or directory
student@selab:~$ nano script.sh
student@selab:~$ chmod 777 script.sh
student@selab:~$
```

```
  GNU nano 4.8                                  script.sh
#!bin/bash
date
ls
```

```
student@selab:~$ ls
a.txt  Desktop    Downloads  Music     Public     Videos
Chiya  Documents  Kushagra   Pictures  Templates
student@selab:~$
```

```
student@selab:~$ nano script.sh
student@selab:~$ chmod 777 script.sh
```

```
student@selab:~$ bash script.sh
Wednesday 13 September 2023 10:47:07 AM IST
a.txt  Chiya  Desktop  Documents  Downloads  Kushagra  Music  Pictures
```

- Exit From Server

```
student@selab:~$ exit
logout
Connection to 10.1.22.134 closed.
student@selab33:~$
```

CO44401- SYSTEM OPERATIONS LAB
LINUX BASED CLUSTERS

WHAT IS LINUX BASED CLUSTERS?

- A Linux cluster is a connected array of Linux computers or nodes that work together and can be viewed and managed as a single system. Nodes are usually connected by fast local area networks, with each node running its own instance of Linux. Nodes may be physical or virtual machines, and they may be separated geographically.

- Clusters are usually dedicated to specific functions, such as load balancing, high availability, high performance, s ge-scale processing.

IMPORTANCE

- Parallel Computing
- High Performance
- Scalability
- Open Source Software
- Research and Scientific Computing
- Containerization

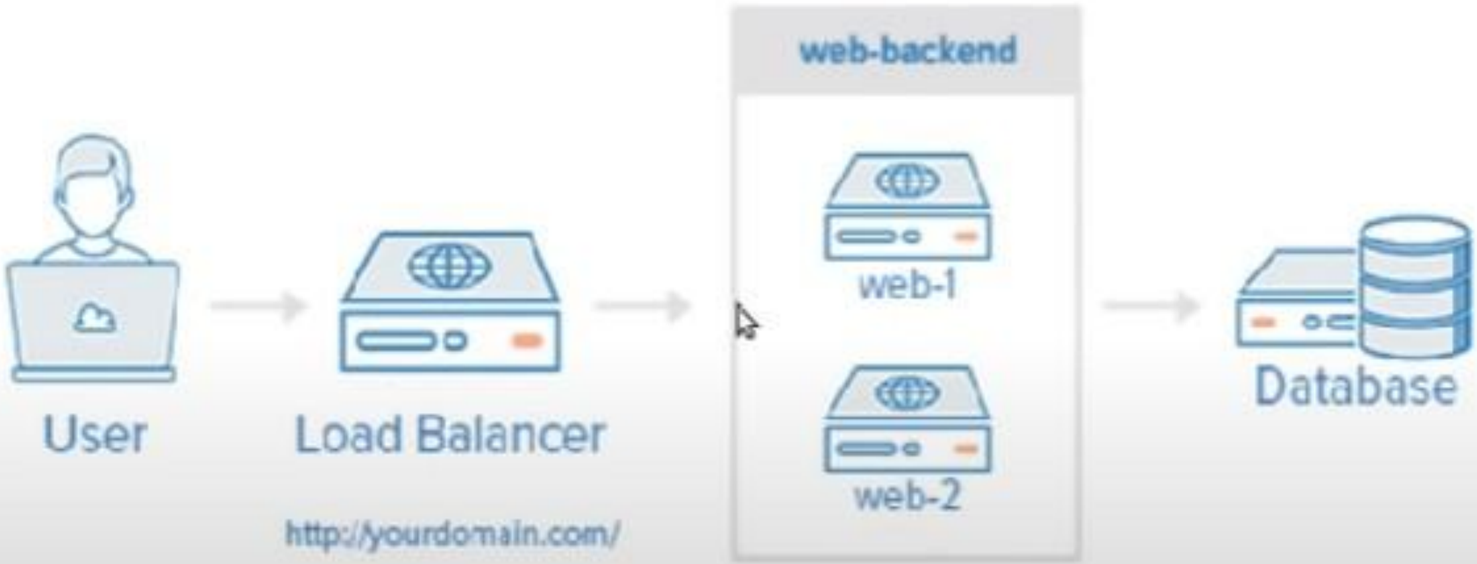
LOAD BALANCING

- Load balancing means splitting up network traffic so that you can distribute it evenly across a group of backend servers. For example, if you run two web servers, both hosting a copy of the same website, then you can balance the traffic across them, sending half to one and half to the other.

- The goal of load balancing is to increase the availability of your website or web-based application by routing a portion of requests to each server. If one of your load-balanced servers fails, then the other is still there to handle requests. It doesn’t have to be web-based traffic, either. Load balancing can work for any networked application, such as FTP servers, databases, and cache servers.

LOAD BALANCING

LOAD BALANCING



INSTALL APACHE SERVER

```
E: Invalid operation apache2
cndc-17@cndc17-OptiPlex-3050-AIO:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 upgraded, 8 newly installed, 0 to remove and 186 not upgraded.
Need to get 1,918 kB of archives.
After this operation, 7,706 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4.22.04.2 [92.8 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4.22.04.2 [11.3 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4.22.04.2 [9,170 B]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.6 [1,345 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-1ubuntu4.6 [165 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-1ubuntu4.6 [89.1 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-1ubuntu4.6 [97.8 kB]
```

```
GNU nano 6.2                                                    /var/www/html/index.html *
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Web Server 2: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }

  body, html {
    padding: 3px 3px 3px 3px;

    background-color: #D8DBE2;

    font-family: Ubuntu, Verdana, sans-serif;
    font-size: 11pt;
    text-align: center;
  }

  div.main_page {
    position: relative;
    display: table;

    width: 800px;

    margin-bottom: 3px;
    margin-left: auto;
    margin-right: auto;
    padding: 0px 0px 0px 0px;

    border-width: 2px;
    border-color: #212738;
    border-style: solid;
```

LINUX CLUSTERS FOR LOAD BALANCING USING HAPROXY AND KEEPALIVED

- Install HAProxy:
  - sudo apt-get install
- haproxy Configure HAProxy
- Start HAProxy:
  - sudo systemctl start haproxy
  - sudo systemctl enable
- haproxy

Test Load Balancing

LINUX CLUSTERS FOR LOAD BALANCING

```
cndc-17@cndc17-OptiPlex-3050-AIO:~$ sudo apt-get install haproxy
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  vim-haproxy haproxy-doc
The following NEW packages will be installed:
  haproxy
0 upgraded, 1 newly installed, 0 to remove and 186 not upgraded.
Need to get 1,645 kB of archives.
After this operation, 3,690 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 haproxy amd64 2.4.22-0ubuntu0.22.04.2 [1,645 kB]
Fetched 1,645 kB in 3s (632 kB/s)
Selecting previously unselected package haproxy.
(Reading database ... 205799 files and directories currently installed.)
Preparing to unpack .../haproxy_2.4.22-0ubuntu0.22.04.2_amd64.deb ...
Unpacking haproxy (2.4.22-0ubuntu0.22.04.2) ...
Setting up haproxy (2.4.22-0ubuntu0.22.04.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/haproxy.service → /lib/systemd/system/haproxy.service.
Processing triggers for rsyslog (8.2112.0-2ubuntu2.2) ...
Processing triggers for man-db (2.10.2-1) ...
```

```
 GNU nano 6.2                                  /etc/haproxy/haproxy.cfg *
        stats timeout 30s
        user haproxy
        group haproxy
        daemon

        # Default SSL material locations
        ca-base /etc/ssl/certs
        crt-base /etc/ssl/private

        # See: https://ssl-config.mozilla.org/#server=haproxy&server-version=2.0.3&config=intermediate
        ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA>
        ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
        ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

defaults
        log     global
        mode    http
        option  httplog
        option  dontlognull
        timeout connect 5000
        timeout client  50000
        timeout server  50000
        errorfile 400 /etc/haproxy/errors/400.http
        errorfile 403 /etc/haproxy/errors/403.http
        errorfile 408 /etc/haproxy/errors/408.http
        errorfile 500 /etc/haproxy/errors/500.http
        errorfile 502 /etc/haproxy/errors/502.http
        errorfile 503 /etc/haproxy/errors/503.http
        errorfile 504 /etc/haproxy/errors/504.http

frontend myweb
        bind *:80
        option tcplog
        mode tcp
        default_backend web-servers
backend web-servers
        more tcp
        balanced roundrobin
        option tcp-check
        server web1 10.1.100.16:80 check fall 3 rise 2
        server web2 10.1.100.17:80 check fall 3 rise 2
```

```
Processing triggers for man-db (2.10.2-1) ...
cndc-17@cndc17-OptiPlex-3050-AIO:~$ sudo nano /etc/haproxy/haproxy.cfg
cndc-17@cndc17-OptiPlex-3050-AIO:~$ sudo systemctl enable haproxy
Synchronizing state of haproxy.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable haproxy
cndc-17@cndc17-OptiPlex-3050-AIO:~$ sudo systemctl start haproxy
cndc-17@cndc17-OptiPlex-3050-AIO:~$ sudo apt install -y keepalived
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ipvsadm
Suggested packages:
  heartbeat ldirectord
The following NEW packages will be installed:
  ipvsadm keepalived
0 upgraded, 2 newly installed, 0 to remove and 186 not upgraded.
Need to get 495 kB of archives.
After this operation, 1,452 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 keepalived amd64 1:2.2.4-0.2build1 [453 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 ipvsadm amd64 1:1.31-1build2 [42.2 kB]
Fetched 495 kB in 2s (321 kB/s)
Selecting previously unselected package keepalived.
(Reading database ... 205889 files and directories currently installed.)
Preparing to unpack .../keepalived_1%3a2.2.4-0.2build1_amd64.deb ...
Unpacking keepalived (1:2.2.4-0.2build1) ...
Selecting previously unselected package ipvsadm.
Preparing to unpack .../ipvsadm_1%3a1.31-1build2_amd64.deb ...
Unpacking ipvsadm (1:1.31-1build2) ...
Setting up ipvsadm (1:1.31-1build2) ...
Setting up keepalived (1:2.2.4-0.2build1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/keepalived.service → /lib/systemd/system/keepalived.service.
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Processing triggers for man-db (2.10.2-1) ...
cndc-17@cndc17-OptiPlex-3050-AIO:~$
```

KEEPALIVED LOADBALANCING SETUP

```
cndc-16@cndc16-OptiPlex-3050-AIO:~/Desktop$ cat /etc/keepalived/keepalived.conf
    # Define the script used to check if haproxy is still working
    vrrp_script chk_haproxy {
        script "killall -0 haproxy"
        interval 2
        weight 2
    }
    # Configuation for the virtual interface
    vrrp_instance VI_1 {
        interface enp1s0
        state BACKUP # set this to BACKUP on the other machine
        priority 100        # set this to 100 on the other machine
        virtual_router_id 51
        smtp_alert          # Activate email notifications
        authentication {
            auth_type AH
            auth_pass myPassw0rd      # Set this to some secret phrase
        }
        # The virtual ip address shared between the two loadbalancers
        virtual_ipaddress {
    10.1.100.99
        }
        # Use the script above to check if we should fail over
        track_script {
            chk_haproxy
        }
    }
cndc-16@cndc16-OptiPlex-3050-AIO:~/Desktop$
```

```
cndc-16@cndc16-OptiPlex-3050-AIO:~/Desktop$ sudo systemctl status keepalived
● keepalived.service - Keepalive Daemon (LVS and VRRP)
     Loaded: loaded (/lib/systemd/system/keepalived.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-09-20 13:16:28 IST; 4min 13s ago
   Main PID: 9725 (keepalived)
      Tasks: 2 (limit: 9101)
     Memory: 2.3M
        CPU: 1.303s
     CGroup: /system.slice/keepalived.service
             ├─9725 /usr/sbin/keepalived --dont-fork
             └─9726 /usr/sbin/keepalived --dont-fork

Sep 20 13:16:28 cndc16-OptiPlex-3050-AIO systemd[1]: keepalived.service: Got notification message from PID 9726, but reception only permitted for main PID 9725
Sep 20 13:16:28 cndc16-OptiPlex-3050-AIO Keepalived_vrrp[9726]: WARNING - default user 'keepalived_script' for script execution does not exist - please create.
Sep 20 13:16:28 cndc16-OptiPlex-3050-AIO Keepalived_vrrp[9726]: (/etc/keepalived/keepalived.conf: Line 16) Truncating auth_pass to 8 characters
Sep 20 13:16:28 cndc16-OptiPlex-3050-AIO systemd[1]: Started Keepalive Daemon (LVS and VRRP).
Sep 20 13:16:28 cndc16-OptiPlex-3050-AIO Keepalived_vrrp[9726]: WARNING - script `killall` resolved by path search to `/usr/bin/killall`. Please specify full path.
Sep 20 13:16:28 cndc16-OptiPlex-3050-AIO Keepalived_vrrp[9726]: SECURITY VIOLATION - scripts are being executed but script_security not enabled.
Sep 20 13:16:28 cndc16-OptiPlex-3050-AIO Keepalived[9725]: Startup complete
Sep 20 13:16:28 cndc16-OptiPlex-3050-AIO Keepalived_vrrp[9726]: (VI_1) Entering BACKUP STATE (init)
Sep 20 13:16:28 cndc16-OptiPlex-3050-AIO Keepalived_vrrp[9726]: VRRP_Script(chk_haproxy) succeeded
Sep 20 13:16:28 cndc16-OptiPlex-3050-AIO Keepalived_vrrp[9726]: (VI_1) Changing effective priority from 100 to 102
```

10.1.100.99

Ubuntu Software

Main Server

Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

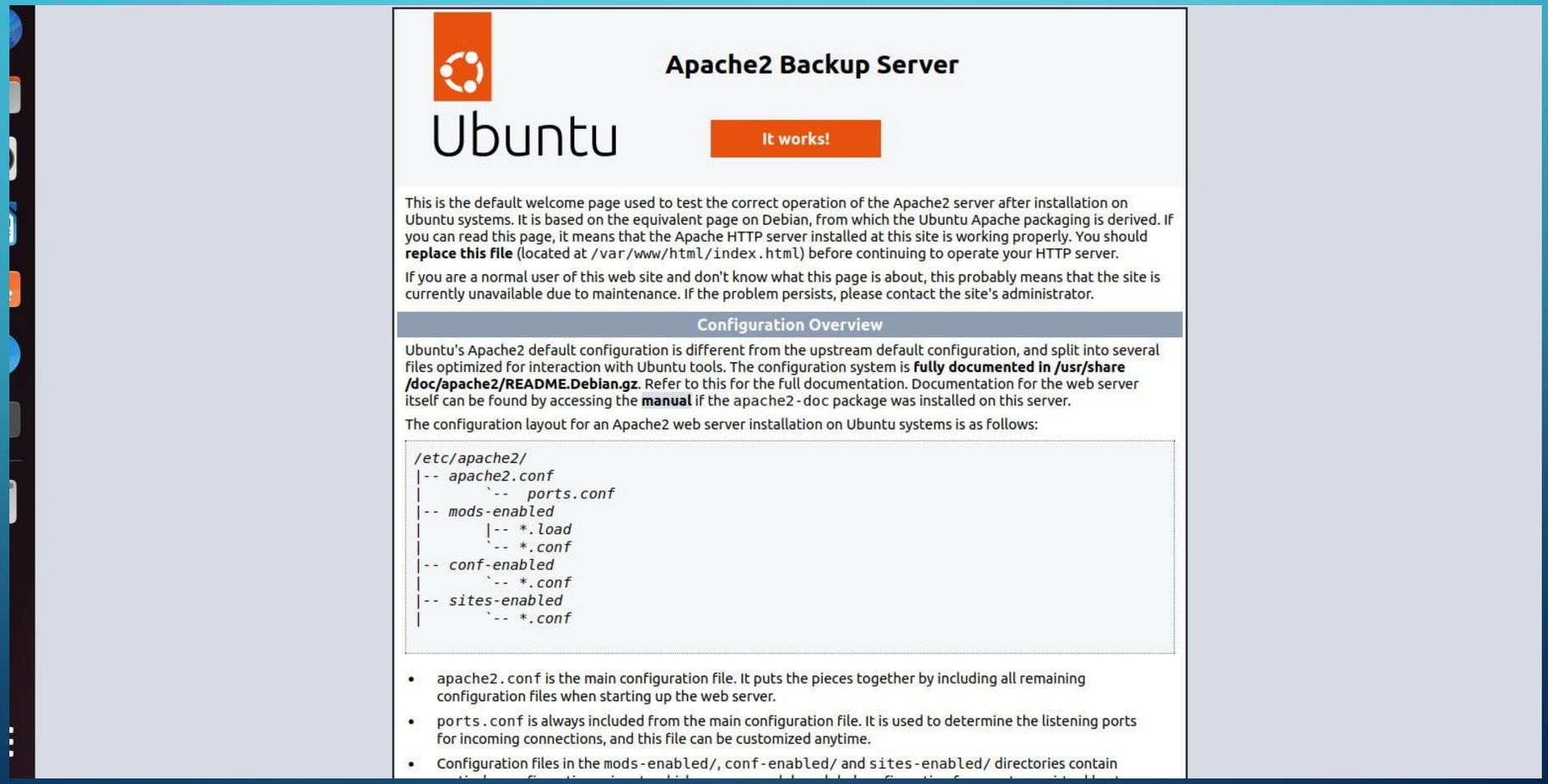
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share /doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports

```
ex-3050-AIO:~$ sudo systemctl start keepalived
ex-3050-AIO:~$ sudo systemctl stop keepalived
ex-3050-AIO:~$
```



SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE

DEPARTMENT OF COMPUTER ENGINEERING

TOPIC: SCRIPTING
SYSTEM OPERATION LAB: CO44401

SYSTEM OPERATIONS LAB

TASK - 3

Scripting

SED

SED is primarily used for performing basic text transformations on an input stream (a file or input from a pipeline).It reads the input line by line, applies specified operations, and outputs the result to the standard output.

COMMON SED COMMANDS

1. **Substitution:** Replace occurrences of a pattern with another string. sed 's/pattern/replacement/' file.txt

Delete: Remove lines that match a pattern.

sed '/pattern/d' file.txt

AWK

AWK is a versatile programming language designed for text processing. It processes input line by line, and you can define rules to process and manipulate data based on fields (columns) and patterns.

COMMON AWK COMMANDS

1. Print: Print specific fields or the entire line.

. awk '{ print $1 }' file.txt

awk '{ print $1, $3 }' file.txt

1.Conditional Statements: Process lines based on conditions.

2.awk '$3 > 50 { print $1 }' file.txt

3.Calculations: Perform arithmetic operations.

4.awk '{ total += $3 } END { print total }' file.txt

5.Built-in Variables: Access AWK's built-in variables.

6.awk '{ print NR, NF }' file.txt

TO MAKE 10 DIFFERENT SCRIPTS TO PERFORM OPERATIONS.

1. **Rename** Files Script(rename_files.sh):

```
#!/bin/bash
old_name = “old_files.txt”
new_name =”new_files.txt”


mv”$old_name””$new_name”
```

Note: This Script renames a file named as “old_files.txt” in the current directory.

2. **Make Folder Script** (make_folder.sh):

```
#!/bin/bash
folder_name="new_folder"

mkdir "$folder_name"
```

3. **Search and Replace Script** (search_replace.sh):

```
#!/bin/bash
search_string = “old_text”
replace_string = “new_text”
sed -i "s/$search_string/$replace_string/g" file.txt
```

Note: This script uses sed to replace all occurrences of "old_text" with "new_text" in the file "file.txt."

4. **Delete Empty Files Script** (delete_empty_files.sh):

```
#!/bin/bash
find . -type f -empty -delete
```

Note:This script finds and deletes all empty files in the current directory and its subdirectories.

5. **Extract ZIP Files Script** (extract_zip.sh):

```
#!/bin/bash


zip_file="file.zip"
unzip "$zip_file"
```

Note:This script extracts the contents of "file.zip" in the current directory.

6. **Count Lines in Files Script** (count_lines.sh):

```
#!/bin/bash
wc -l * > line_counts.txt
```

Note:This script uses wc to count the number of lines in each file in the current directory and saves the output in "line_counts.txt".

7.**Concatenate Files Script** (concat_files.sh):

```
#!/bin/bash
cat file1.txt file2.txt > combined_file.txt
```

Note:This script concatenates the contents of "file1.txt" and "file2.txt" and stores result in "combined_file.txt."

8.**Print Even Numbers Script** (print_even_numbers.sh):

```
#!/bin/bash

for((i=2;i<=10;i+=2));do
    echo$i
done
```

Note:This script prints even numbers from 2 to 10 (inclusive).

9.**Delete Lines Containing Pattern Script**

(delete_lines_pattern.sh):

```
 #!/bin/bash

pattern="delete_me"
sed -i "/$pattern/d" file.txt
```

Note:This script uses sed to delete all lines containing the specified pattern ("delete_me") in the file "file.txt."

10. **Find Largest File Script** (find_largest_file.sh):

```
#!/bin/bash
largest_file=$(find . -type f -exec du -h {} + | sort -rh | head

-n 1 | cut -f 2)

echo "The largest file is: $largest_file"
```

Note: This script finds the largest file in the current directory and its subdirectories and prints its name.

PRINT INFORMATION ABOUT FIRST 5 PROCESS IN MEMORY USING SED AND AWK.

To print information about the first 5 processes in memory using sed and awk, you can use the ps command to retrieve the process information and then use sed and awk to manipulate the output.

Command:

```
ps aux | sed -n '1,6p' | awk '{print "PID: "$2", User: "$1", CPU %: "$3", Memory %: "$4",
Command:  "$11}'
```

Explanation:

1. **Ps aux:** This command lists all processes along with the detailed information like user, CPU usage,memory usage, and command.
2. **Sed -n’1,6p’:** This command selects and prints the first 6 lines of the output. The first line contains headers,and the next 5 line contain information about the first 5 processes.
3. **awk '{print "PID: "$2", User: "$1", CPU %: "$3", Memory %: "$4", Command: "$11}':** This awk command formats the selected lines from sed to display the desired information. It extracts the relevant fields (PID, User, CPU %, Memory %, and Command) using their respective column numbers.

Task 5- Understanding the Network Files

NETWORK PARAMETERS

Steps:

Open the Terminal:

- Open the terminal on your Linux system. We find the terminal in your system's applications menu or by pressing Ctrl + Alt + T.

Edit the Configuration File:

- We use a text editor like nano, vim, or gedit to edit the network configuration file. We'll need administrative privileges, so prefix the command with sudo.

```
sudo nano /etc/netplan/01-netcfg.yaml
```

UPDATE THE PARAMETERS:

- Inside the configuration file, locate the section related to the network interface you want to configure
- Update the parameters such as IP address, gateway, subnet mask
- Save Changes

```
network:
  version: 2  renderer: networkd  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.1.10/24]  gateway4: 192.168.1.1  nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

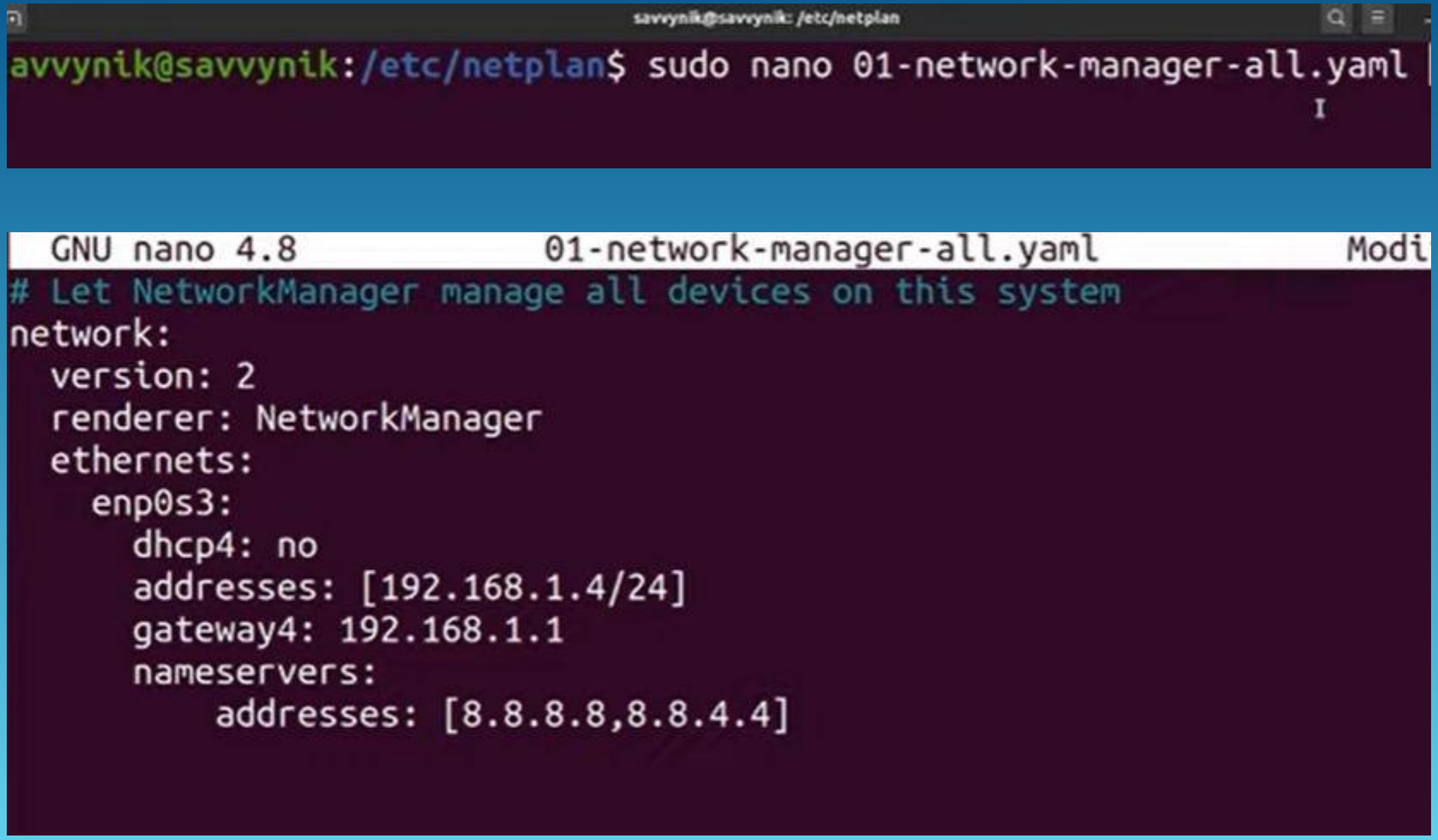
APPLY CHANGES

- After editing the file, save the changes and close the text editor.
- Apply the changes

```
sudo netplan apply
```

OUTPUTS


```
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether 08:00:27:b4:0c:27 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.114/24 brd 192.168.1.255 scope global dynamic noprefixro
ute enp0s3
       valid_lft 85774sec preferred_lft 85774sec
    inet6 2600:6c40:300:e1f:489f:5528:3277:aca8/64 scope global temporary d
ynamic
       valid_lft 433936sec preferred_lft 85191sec
    inet6 2600:6c40:300:e1f:5d5e:ae07:572:21f0/64 scope global dynamic mngt
mpaddr noprefixroute
       valid_lft 433936sec preferred_lft 433936sec
    inet6 fd0b:af42:fd74:0:489f:5528:3277:aca8/64 scope global temporary dy
namic
```


savvynik@savvynik: /etc/netplan
avvynik@savvynik:/etc/netplan$ sudo nano 01-network-manager-all.yaml
GNU nano 4.8 01-network-manager-all.yaml Modi
Let NetworkManager manage all devices on this system
network:
version: 2
renderer: NetworkManager
ethernets:
enp0s3:
dhcp4: no
addresses: [192.168.1.4/24]
gateway4: 192.168.1.1
nameservers:
addresses: [8.8.8.8,8.8.4.4]

```
savvynik@savvynik:/etc/netplan$ sudo netplan apply
savvynik@savvynik:/etc/netplan$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group d
efault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether 08:00:27:b4:0c:27 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.4/24 brd 192.168.1.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feb4:c27/64 scope link
```

Routes:

Routes can be added using the ip route command or by modifying the /etc/network/interfaces file.

```
sudo ip route add 10.0.0.0/24 via 192.168.1.1
```

```
$ nano /etc/network/interfaces
```

```
auto eth0
iface eth0 inet static
    address 192.168.1.10
    netmask 255.255.255.0
    gateway 192.168.1.1
    post-up route add - net 192.168.1.10 netmask 255.255.255.0 gw 192.168.1.1
```

Resolvers (DNS Servers):

DNS resolvers are configured in the /etc/resolv.conf file

```
sudo nano /etc/resolvconf/resolv.conf
```

Add the DNS server entries and then Save the file and update the resolvconf service:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

```
sudo resolvconf -u
```

XINETD SCRIPT

- Xinetd (Extended Internet Services Daemon) is a powerful tool that allows you to manage and control various network services on Unix-like systems such as Ubuntu.
- It acts as a "super-server" that listens for incoming network requests and can spawn individual service processes to handle those requests.
- Xinetd scripts, also known as service configuration files, play a crucial role in defining how Xinetd manages these services.

XINETD SCRIPT

- **Service Configuration**: Its configuration files are located in the

  /etc/xinetd.d/ directory.

- **Service Activation:** It acts as a gatekeeper for network services.
- **Resource Efficiency:** It is known for its resource efficiency. It allows you to run multiple services on a single port, reducing the overall resource usage.
- **Customization:** It allow you to customize various aspects of service behavior, such as setting service-specific environment variables or passing arguments to the service process..

Containers and Virtual Machines

SYSTEM OPERATION LAB

Virtual Machines

- A virtual machine is like a computer emulator that runs inside your actual computer. It lets you use different operating systems and run software as if they were on separate, independent computers, even though they're really just simulations running on your real computer. Simply virtual machines are like computer-in-a-computer setups that let you run multiple operating systems and software on a single physical machine, helping you do different tasks without needing multiple physical computers.

Advantages of Virtual Machine

Isolation
VMs isolate programs and protect against problems, letting different apps run safely on one computer.

Compatibility
VMs support diverse operating systems, making them ideal for legacy apps needing specific environments to run alongside modern software on shared hardware.

Snapshot and Rollback
VMs let you take snapshots for backups and recovery, helping system admins and developers easily go back to previous setups if something goes wrong.

Resourse Allocation
VMs let you control how much CPU, memory, and storage each virtual machine gets, making it easier to manage resources and make sure important apps have enough.

Mobility
VMs can move between computers smoothly, letting you balance work, fix hardware, and use server resources without causing downtime for your virtualized apps.

Types of Virtual Machine

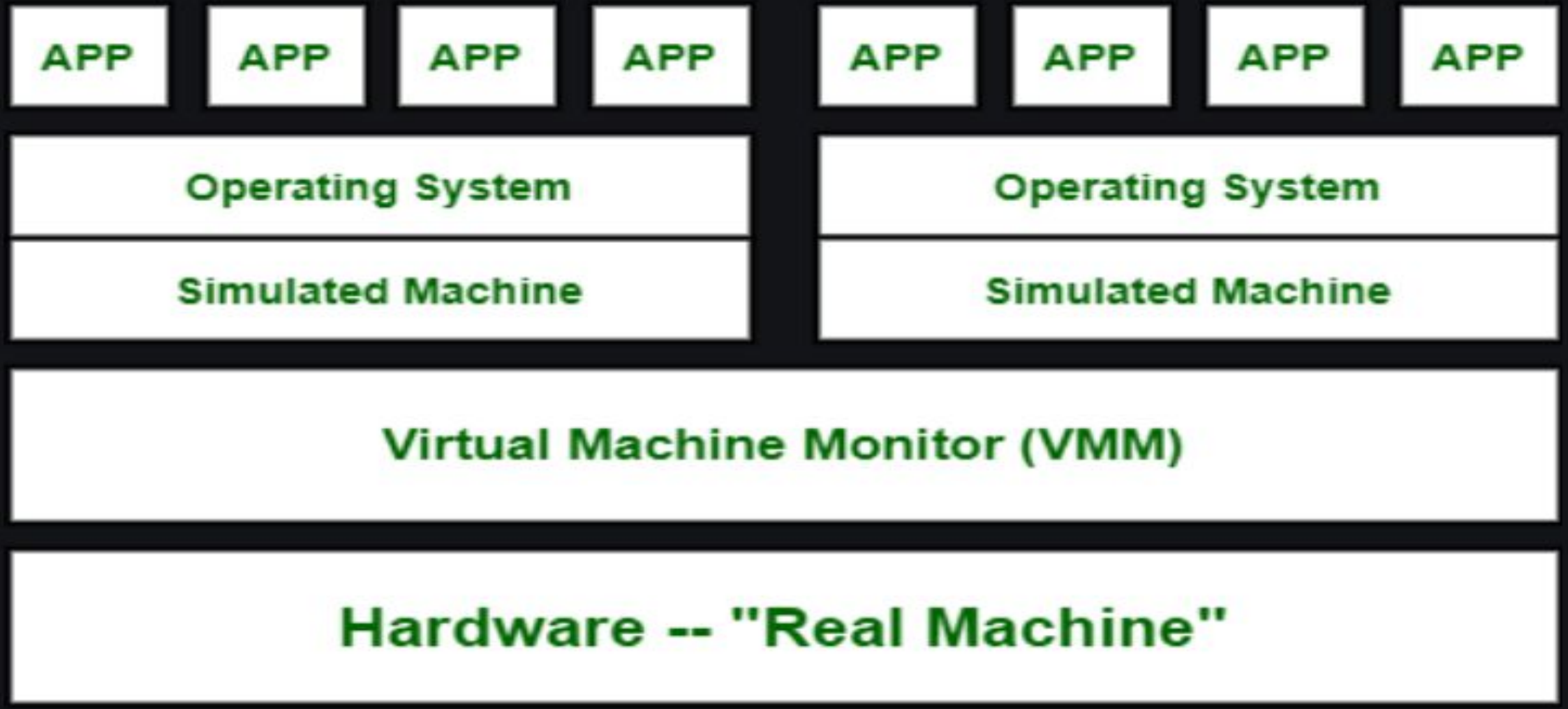
System Virtal Machine

These types of virtual machines gives us complete system p
execution of the complete virtual operating system. Just lik
virtual machine is providing an environment for an OS to b
Example:- Oracle VM.

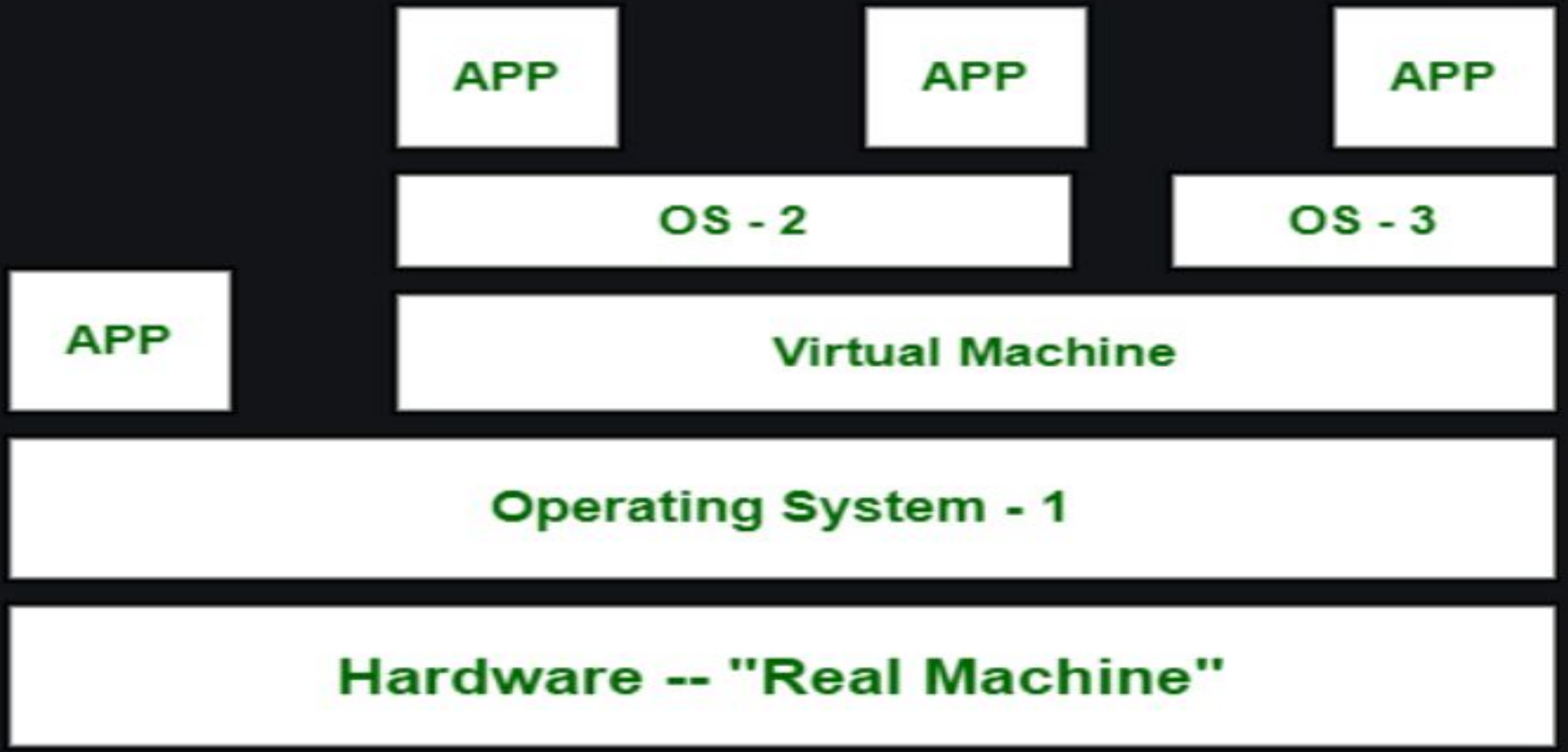
Process Virtual Machine

While process virtual machines, unlike system virtual mach
with the facility to install the virtual operating system comp
virtual environment of that OS while using some app or pro
environment will be destroyed as soon as we exit from that

System Virtual Machine



Process Virtual Machine



Containers

- A Docker container is a lightweight, standalone, and portable package that contains everything a piece of software needs to run, including the code, libraries, and settings. It's like a mini-computer that runs inside your computer, and it's isolated from the rest of your system, so it won't mess up anything else. It makes it easy to run software consistently, no matter where you are or what computer you're using.

Advantages of Docker Contianer

Less Overhead

All containers need fewer resources of the system compared to hardware or traditional VM environments due to they do not contain images of an operating system.

Application Development

Application Development
Every container support DevOps and Agile efforts for accelerating production, test, and development cycles.

Consistent operations

Every team of DevOps knows that applications will execute the same without having to care where they're deployed in the containers.

Greater Efficienty

Each container permits applications to be scaled, patched, and deployed more rapidly.

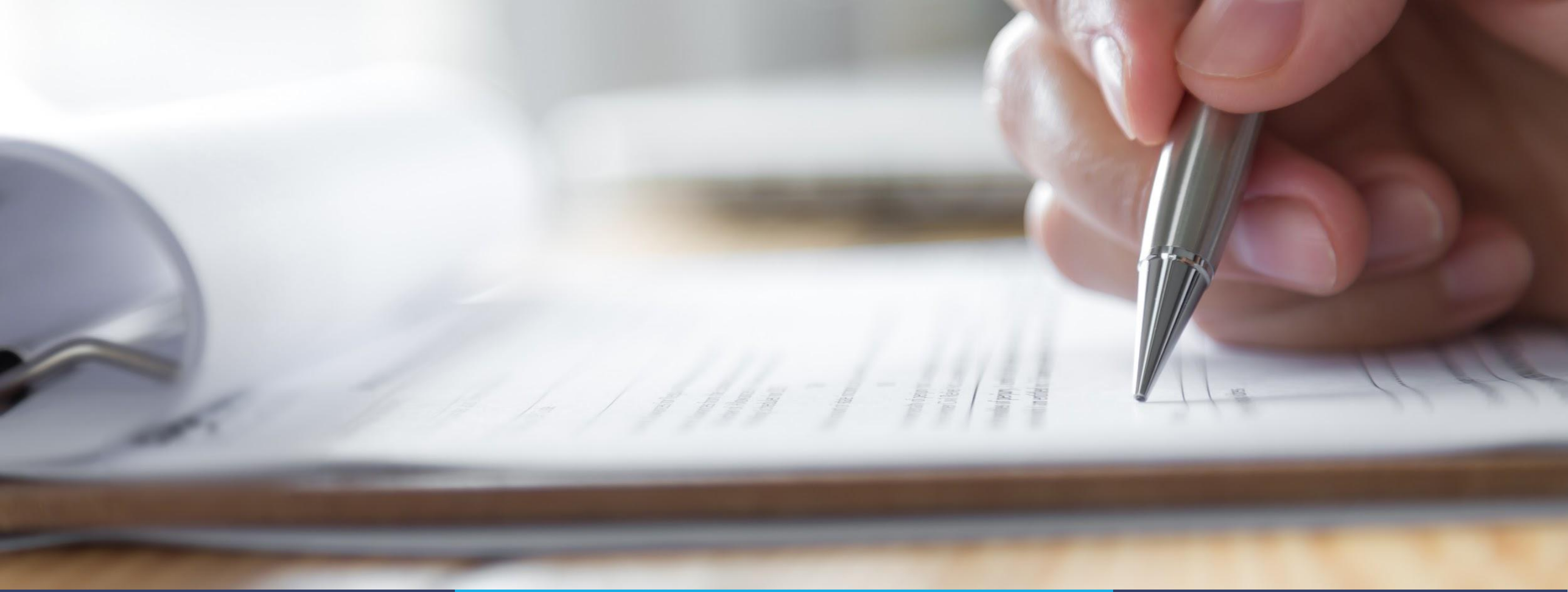
.

Difference Between VM and Containers

Aspect	Docker Container	Virtual Machines
Resource Efficiency	Lightweight, share OS kernel, efficient.	Heavier, each VM has its own OS, less efficient.
Isolation	Uses OS-level isolation, less isolated.	Provides strong isolation, more secure.
Start Time	Start quickly, in seconds.	Slower to start, in minutes.
Resource Overhead	Minimal overhead, smaller footprint.	Higher overhead, larger resource usage.

Difference Between VM and Containers

Aspect	Docker Container	Virtual Machines
Scaling	Scales easily with low overhead.	Scales with more resource requirements.
Security	Limited isolation but good for apps.	Strong isolation, good for critical services.
Portability	Highly portable across systems.	Less portable due to OS dependencies.
Use Cases	Ideal for microservices and DevOps.	Better for legacy apps and diverse workloads.



Containers

Docker Desktop | Upgrade plan | Search for images, containers, volumes, extensions and more... Ctrl+K | keshav...

Walkthroughs

- Multi-container applications — 6 mins
- Persist your data between containers — 3 mins
- Containerize your application — 3 mins
- Publish your image — 5 mins
- How do I run a container? — Completed
- What is a container? — Completed
- Run Docker Hub images — Completed

RAM 0.00 GB CPU 0.00% Signed in v4.23.0

Back to learning center

How do I run a container?

- Images are used to run containers
- Get the sample application
- Verify your Dockerfile
- **Build your first image**

You can build an image using the following **docker build** command via a CLI in your project folder.

```
docker build -t welcome-to-docker .
```

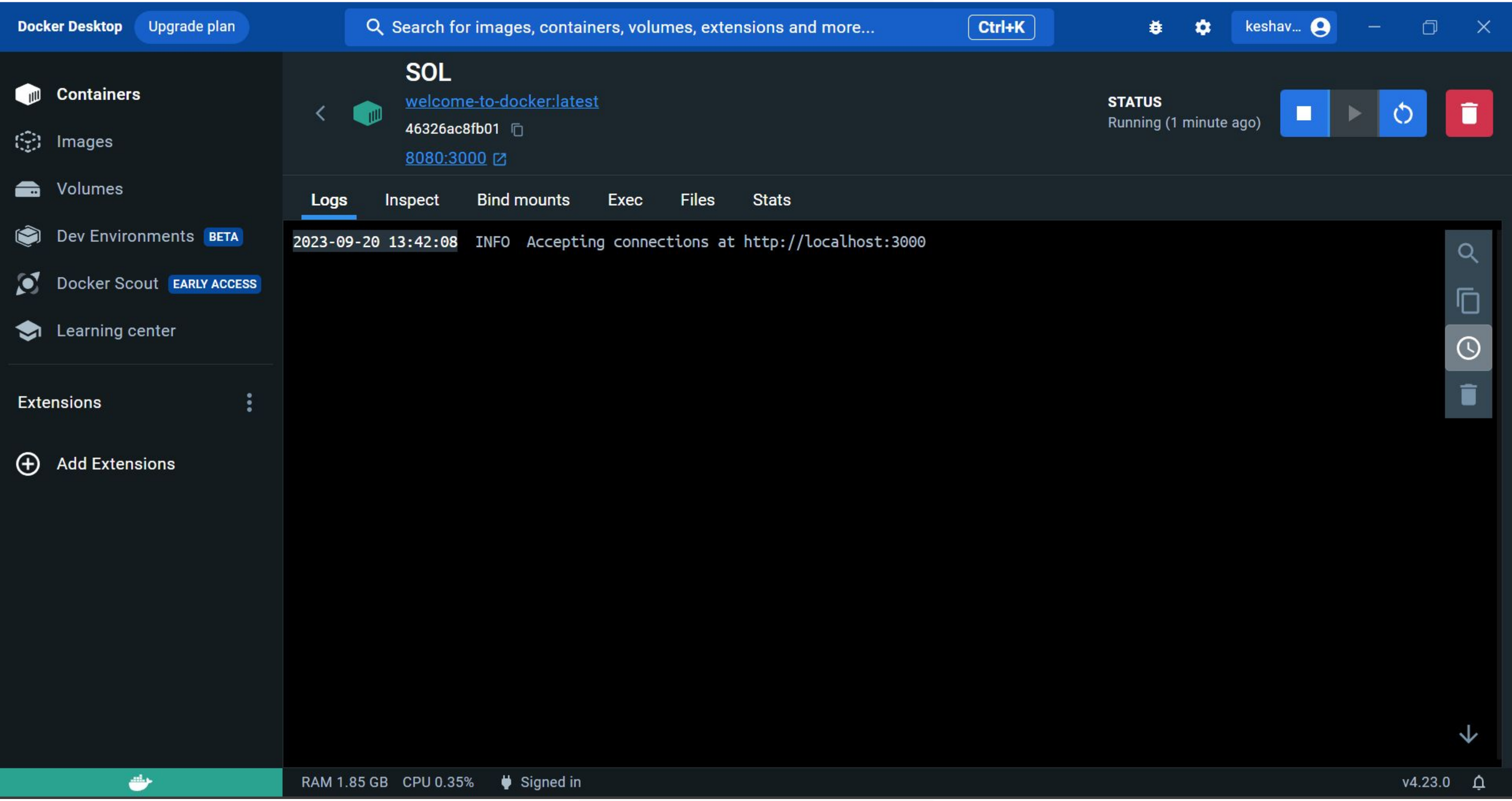
Breaking down this command

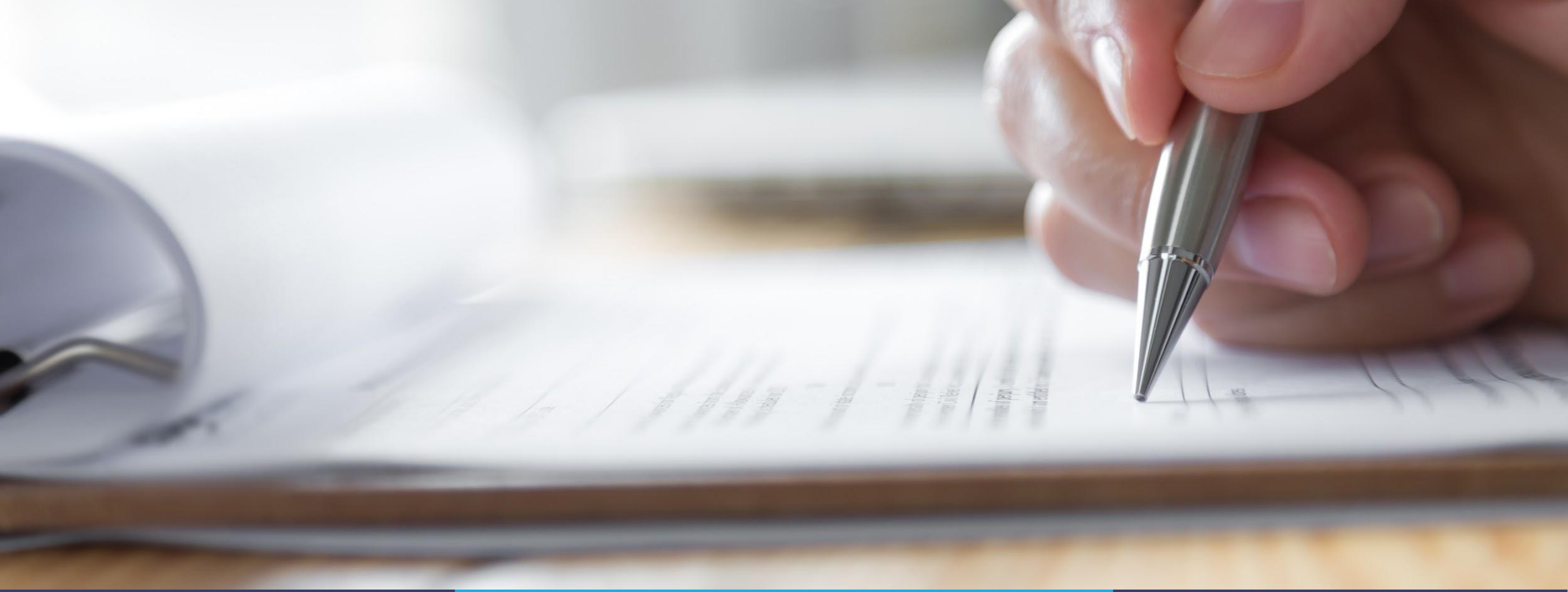
The **-t** flag tags your image with a name. (**welcome-to-docker** in this case). And the **.** lets Docker know where it can find the Dockerfile.

Back | Next

- Run your container

Search | ENG IN | 3:25 PM 9/20/2023


Docker Desktop
Upgrade plan
Search for images, containers, volumes, extensions and more...
Ctrl+K
keshav...
Containers
Images
Volumes
Dev Environments
BETA
Docker Scout
EARLY ACCESS
Learning center
Extensions
Add Extensions
SOL
welcome-to-docker:latest
46326ac8fb01
8080:3000
STATUS
Running (1 minute ago)
Logs
Inspect
Bind mounts
Exec
Files
Stats
2023-09-20 13:42:08 INFO Accepting connections at http://localhost:3000
RAM 1.85 GB CPU 0.35%
Signed in
v4.23.0



Virtual Machine

Oracle VM VirtualBox Manager

File Machine Help

Tools

New Add Settings Discard Start

Ubuntu
Powered Off

General

Name: Ubuntu
Operating System: Ubuntu (64-bit)

System

Base Memory: 4853 MB
Processors: 8
Boot Order: Hard Disk, Optical, Floppy
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Preview

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

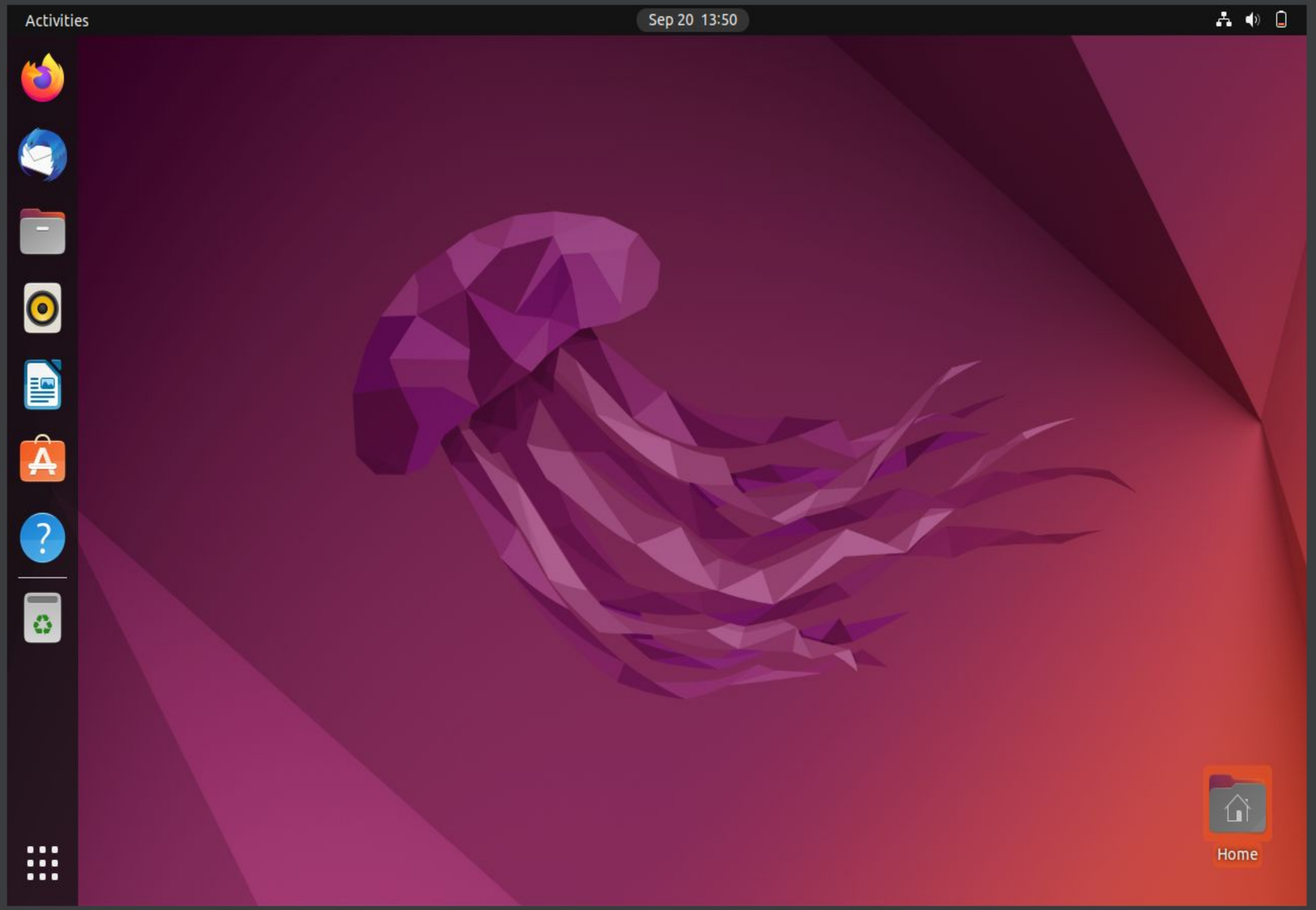
Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Ubuntu.vdi (Normal, 20.00 GB)

Audio

Host Driver: Default

Ubuntu [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help



Thank You

Installation and Setup

Download and Install

Download the Tomcat binary file and follow the installation steps.

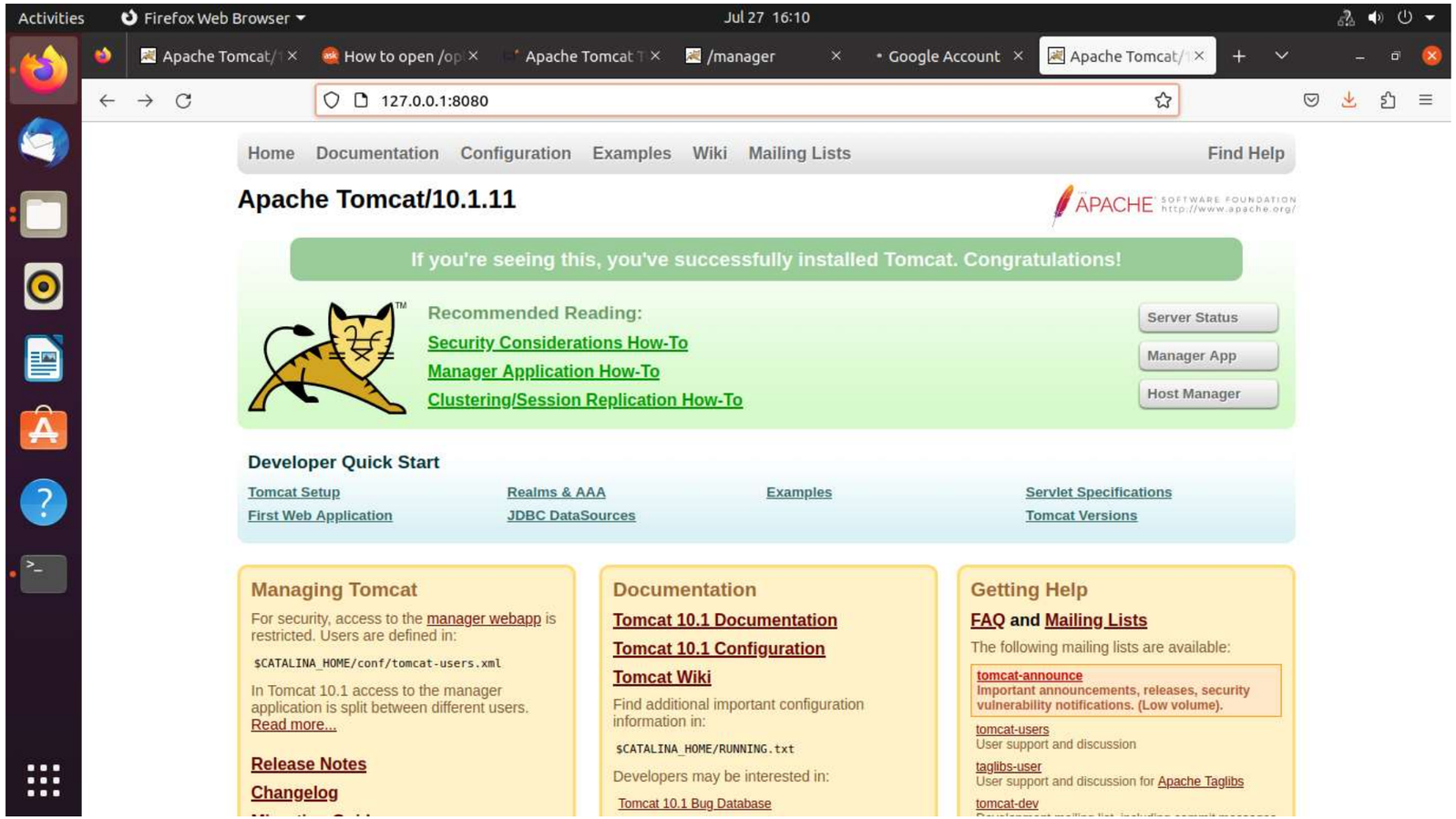
Configuration

Configure Tomcat by setting environment variables and editing configuration files.

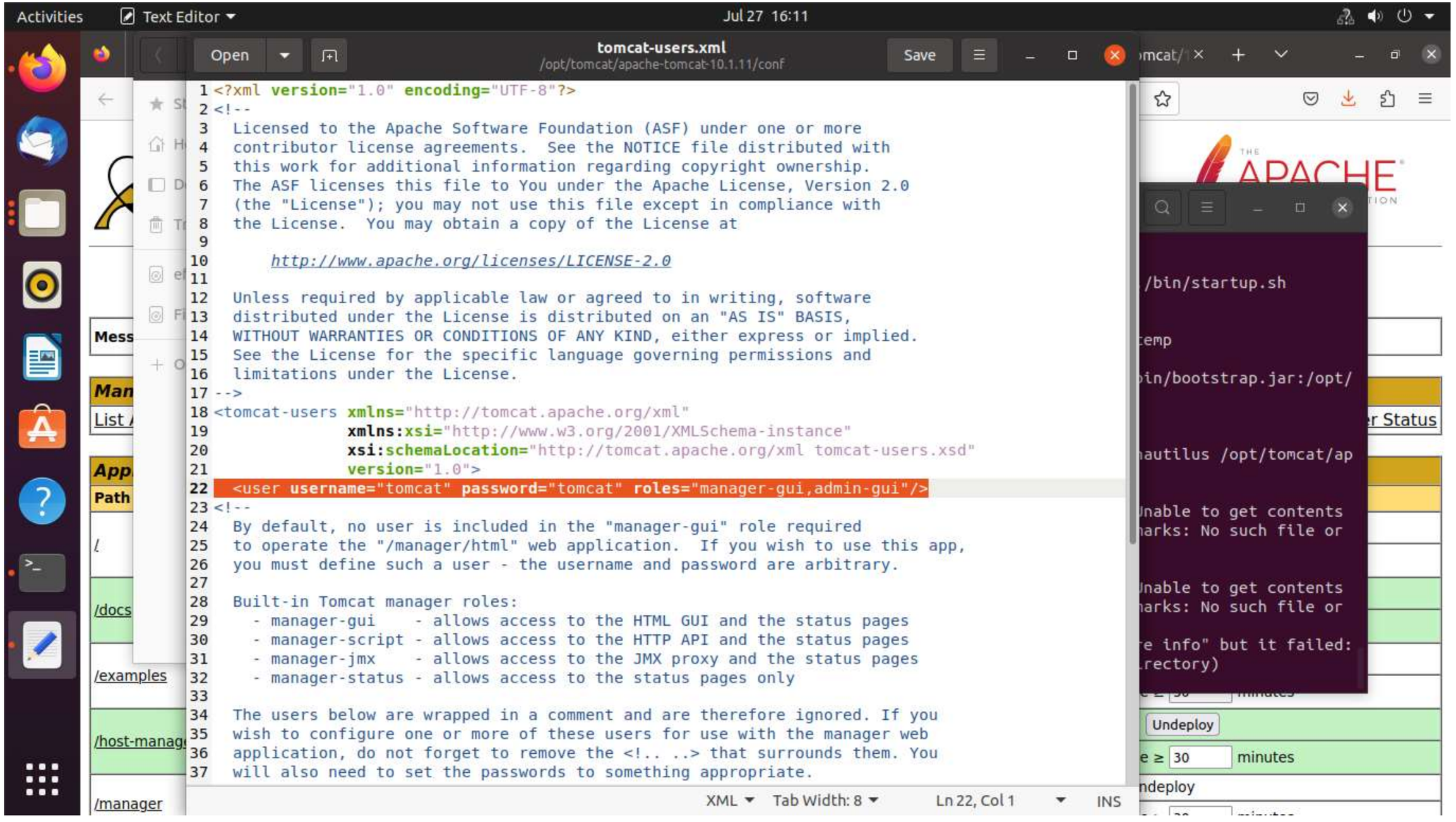
Setup Web Server

Integrate Tomcat with web servers like Apache or NGINX for a complete server setup.

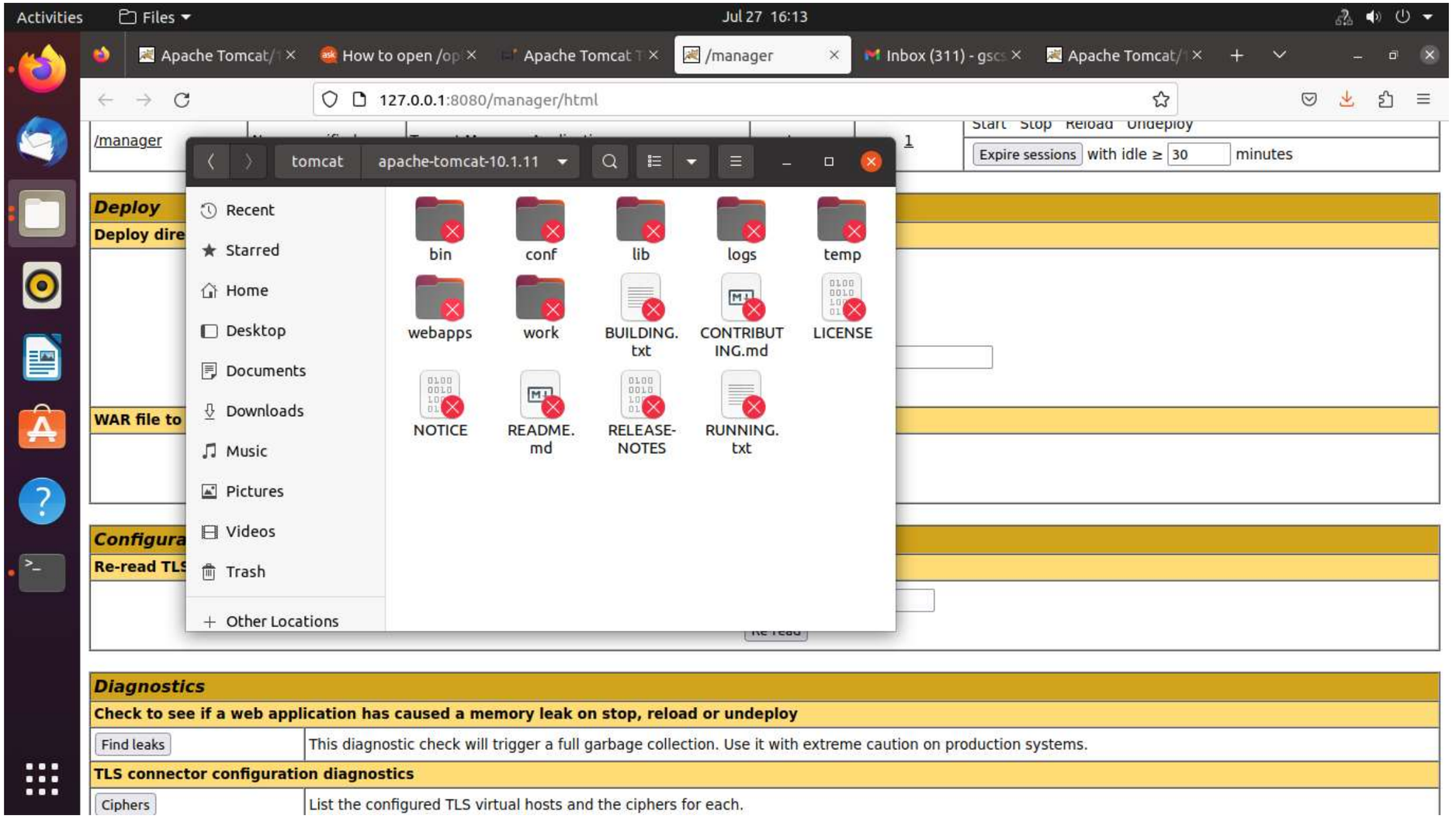
Installation and Setup



- Sudo wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.11/bin/apache-tomcat-10.1.11.tar.gz
- cd Downloads/
- Ls
- sudo tar -xvzf apache-tomcat-10.1.11.tar.gz -C /opt/tomcat


Activities
Text Editor
Jul 27 16:11
Open
tomcat-users.xml
/opt/tomcat/apache-tomcat-10.1.11/conf
Save
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<tomcat-users xmlns="http://tomcat.apache.org/xml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
version="1.0">
<user username="tomcat" password="tomcat" roles="manager-gui,admin-gui"/>
<!--
By default, no user is included in the "manager-gui" role required
to operate the "/manager/html" web application. If you wish to use this app,
you must define such a user - the username and password are arbitrary.
Built-in Tomcat manager roles:
- manager-gui - allows access to the HTML GUI and the status pages
- manager-script - allows access to the HTTP API and the status pages
- manager-jmx - allows access to the JMX proxy and the status pages
- manager-status - allows access to the status pages only
The users below are wrapped in a comment and are therefore ignored. If you
wish to configure one or more of these users for use with the manager web
application, do not forget to remove the <!.. ..> that surrounds them. You
will also need to set the passwords to something appropriate.
XML
Tab Width: 8
Ln 22, Col 1
INS
/bin/startup.sh
Undeploy
minutes
/docs
/examples
/host-manag
/manager

File Structure



Understanding Tomcat Architecture

Servlet Container

Tomcat is a powerful servlet container that manages the life cycle of servlets and JSP pages.

Core Components

Tomcat is composed of several core components.

Class Loaders

Tomcat uses customized class loaders to load web application classes, reducing the risk of conflicts.

Request Processing

Tomcat processes client requests through a series of steps, including encryption, authentication, and authorization.

Deploying Web Applications on Tomcat

WAR Files

Deploy applications packaged as WAR files, which can contain libraries, configuration files, and web resources.

1

Manual Deployment

Deploy web applications by copying them to the webapps directory or using the Tomcat Manager web interface.

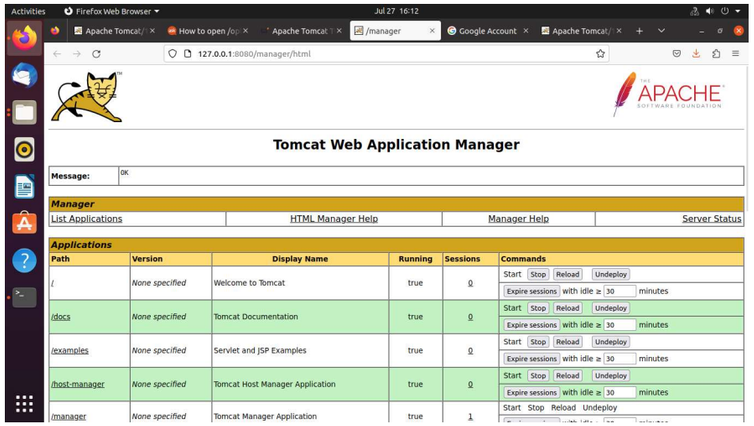
2

3

Deploy Tools

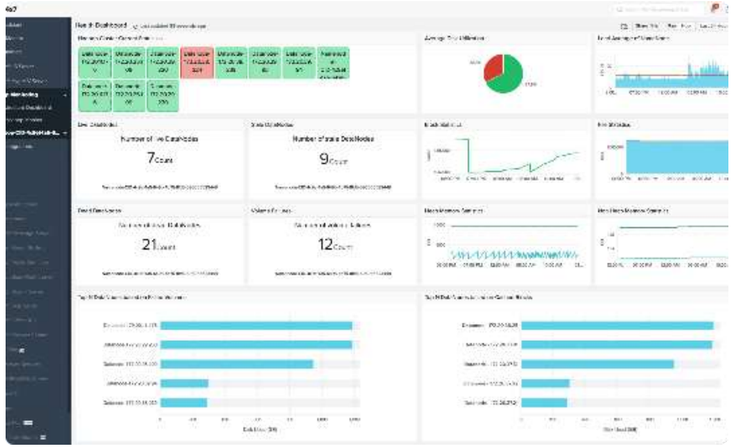
Use deployment tools to automate the application deployment process.

Managing Tomcat Server



Tomcat Manager

Monitor server performance, deploy and undeploy applications, and configure the server using the Tomcat Manager web interface.

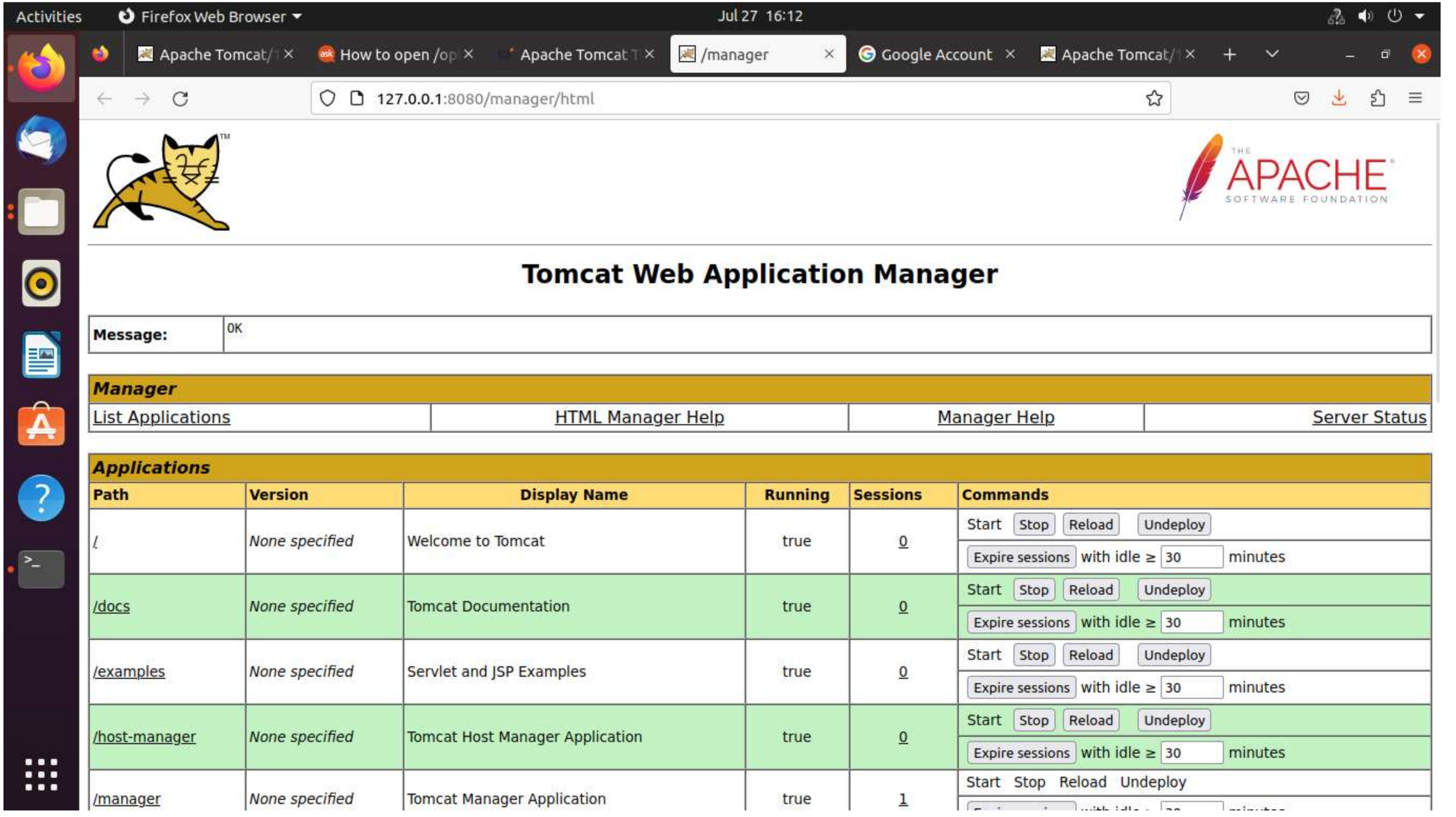


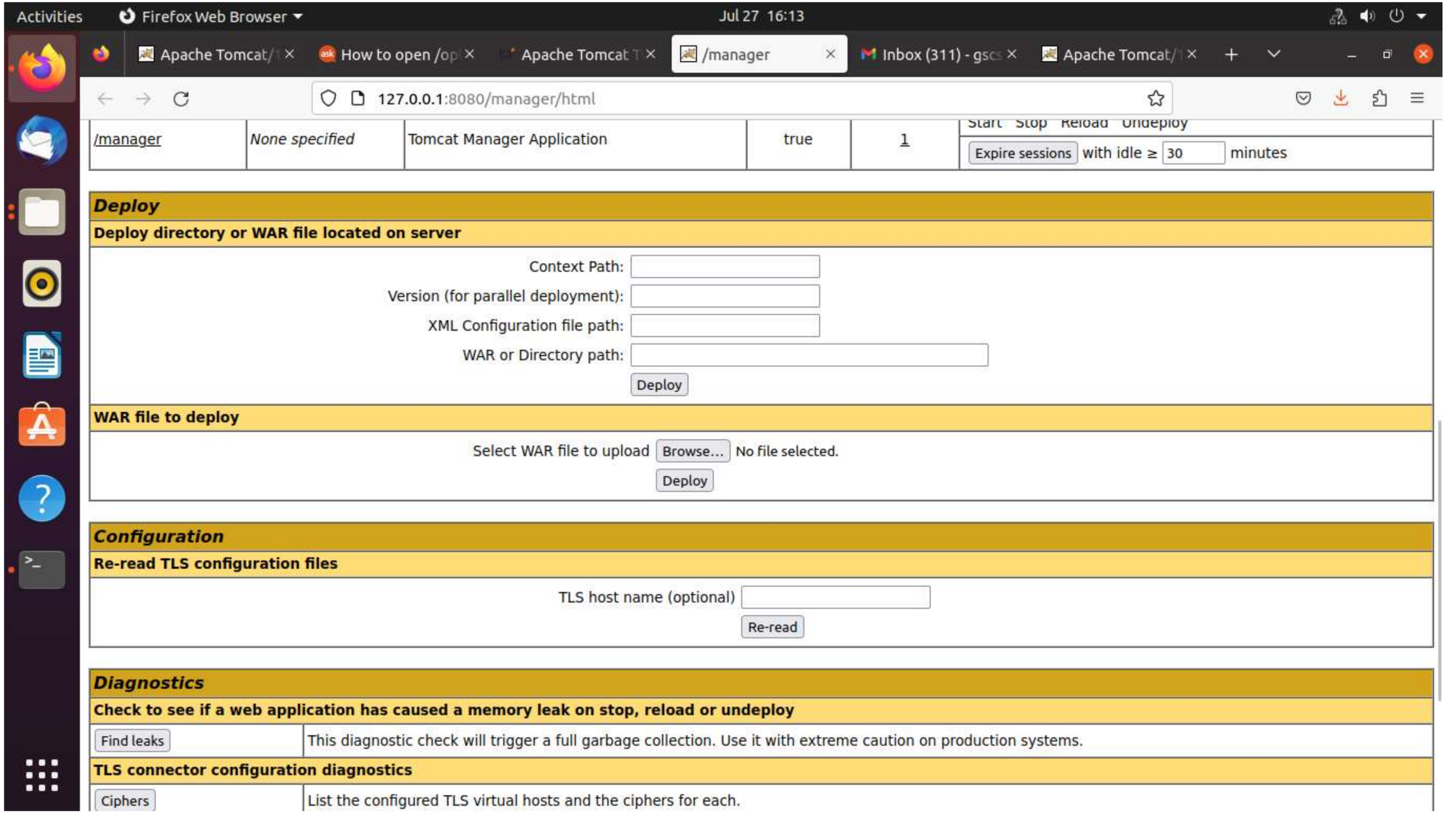
Monitoring and Metrics

Collect and analyze metrics like CPU usage, memory usage, and request processing time to optimize server performance.

Backup and Recovery

Create backups of server configuration and web applications for disaster recovery purposes.


Activities
Firefox Web Browser
Jul 27 16:12
Apache Tomcat/
How to open /op
Apache Tomcat T
/manager
Google Account
Apache Tomcat/
127.0.0.1:8080/manager/html
THE APACHE SOFTWARE FOUNDATION
Tomcat Web Application Manager
Message: OK
Manager
List Applications
HTML Manager Help
Manager Help
Server Status
Applications
Path
Version
Display Name
Running
Sessions
Commands
/
None specified
Welcome to Tomcat
true
0
Start Stop Reload Undeploy
Expire sessions with idle ≥ 30 minutes
/docs
None specified
Tomcat Documentation
true
0
Start Stop Reload Undeploy
Expire sessions with idle ≥ 30 minutes
/examples
None specified
Servlet and JSP Examples
true
0
Start Stop Reload Undeploy
Expire sessions with idle ≥ 30 minutes
/host-manager
None specified
Tomcat Host Manager Application
true
0
Start Stop Reload Undeploy
Expire sessions with idle ≥ 30 minutes
/manager
None specified
Tomcat Manager Application
true
1
Start Stop Reload Undeploy


Activities
Firefox Web Browser
Jul 27 16:13
Apache Tomcat/
How to open /op
Apache Tomcat T
/manager
Inbox (311) - gscs
Apache Tomcat/
127.0.0.1:8080/manager/html
/manager
None specified
Tomcat Manager Application
true
1
Start Stop Reload Undeploy
Expire sessions with idle ≥ 30 minutes
Deploy
Deploy directory or WAR file located on server
Context Path:
Version (for parallel deployment):
XML Configuration file path:
WAR or Directory path:
Deploy
WAR file to deploy
Select WAR file to upload
Browse...
No file selected.
Deploy
Configuration
Re-read TLS configuration files
TLS host name (optional)
Re-read
Diagnostics
Check to see if a web application has caused a memory leak on stop, reload or undeploy
Find leaks
This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.
TLS connector configuration diagnostics
Ciphers
List the configured TLS virtual hosts and the ciphers for each.

```java
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorldServlet extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println(
            "<html><head><title>Hello World Servlet</title></head>");
        out.println("<body>");
        out.println("<h1>Hello World!</h1>");
        out.println("</body></html>");
        out.close();
    }
}
```

```
<html>
    <head>
        <title>Hello World Servlet</title>
    </head>
    <body>
        <h1>Hello World!</h1>
    </body>
</html>
```

```
<servlet>
     <servlet-name>HelloWorld</servlet-name>
     <servlet-class>HelloWorldServlet</servlet-class>
</servlet>
<servlet-mapping>
     <servlet-name>HelloWorld</servlet-name>
     <url-pattern>/servlet/HelloWorld</url-pattern>
</servlet-mapping>
```

Hello World!

Security and Performance Tuning

1 Secure the Server

Use SSL/TLS encryption, set strong authentication and authorization policies, and secure sensitive configuration files.

2 Optimize Performance

Use connection pooling, enable caching, reduce request processing time, and use load balancing to optimize server performance.

3 Maintain the Server

Regularly apply security patches and updates, monitor log files, and use performance tuning tools to keep the server running smoothly.

Conclusion

Benefits of Apache Tomcat

1. Easy to use and configure
2. Open-source and free
3. Powerful and modular architecture
4. Great community support