

. What is work order / incident / task ?

In a Linux on Ubuntu environment, work orders, incidents, and tasks play vital roles in maintaining the system's stability, security, and overall performance. Here's how they are used:

• Work Order

When a system administrator needs to delegate specific responsibilities or tasks to team members, they can issue a work order outlining the required actions. The work order provides clear instructions, timelines, and any necessary resources, ensuring that everyone involved is aware of their responsibilities.

• Incident

In the event of unexpected system issues, such as a server crash or security breach, the incident management process is initiated. The incident is reported, documented, and assigned to relevant team members for investigation and resolution. Incident management aims to restore normal

operation as quickly as possible, following predefined procedures and best practices.

- **Task**

Tasks are the building blocks of everyday system administration and maintenance. As system administrators perform routine activities, they handle tasks like installing software updates, configuring firewalls, monitoring server performance, and managing user permissions.

- **Conclusion**

Linux on Ubuntu is a powerful and widely used operating system, but like any complex environment, it can face challenges and issues. Work orders are used to delegate responsibilities effectively, ensuring that tasks are completed in a coordinated manner. Incidents are unexpected disruptions that require a prompt and systematic response to minimise the impact on the system. Tasks, on the other hand, are routine activities that keep the system running smoothly and securely.

. What are P1 , P2 and P3 Tickets ?

In the context of Linux or any issue tracking system, P1, P2, and P3 are commonly used terms to prioritise and categorise tickets or issues.

- **P1 Ticket (Priority 1)**

P1 tickets are the highest priority issues or incidents. They represent critical problems that significantly impact the system's functionality or stability. These issues usually result in severe disruptions or complete system failures, affecting a large number of users. P1 tickets require immediate attention and are usually worked on with the highest urgency.

- **P2 Ticket (Priority 2)**

P2 tickets are of medium priority. They indicate important issues that might cause significant disruptions or functional limitations but are not as severe as P1 tickets. P2 issues still require attention and should be addressed as soon as possible, but they can sometimes be managed with a slightly longer timeframe.

- **P3 Ticket (Priority 3)**

P3 tickets are of lower priority compared to P1 and P2. These are minor issues or enhancements that do not have a major impact on the system's functionality or stability. P3 tickets can be attended to when the higher priority issues are resolved and the team has available resources.

- **How to use**

The use of P1, P2, and P3 tickets depends on the specific issue tracking or project management system being used. Generally, when an issue or bug is reported, it is assigned a priority level. The priority level helps the development team to organise their workflow and allocate resources efficiently. Here's how it typically works:

- 1. Report the Issue:** Users or testers report issues or bugs they encounter in the system.
- 2. Issue Assessment:** The development team reviews the reported issues and assigns

priority levels (P1, P2, P3, etc.) based on the severity and impact.

3. Prioritisation: P1 tickets are given the highest priority and require immediate action. P2 tickets come next, and P3 tickets are given lower priority.

4. Resource Allocation: Developers and teams focus on addressing higher priority tickets first, aiming to resolve critical issues before moving on to the lower priority ones.

● Conclusion

Using priority levels like P1, P2, and P3 in an issue tracking system allows development teams to efficiently manage their workload and address critical issues promptly. This helps in minimising system downtime, improving overall system stability, and ensuring a smoother user experience. By categorising and prioritising issues, teams can better organise their efforts and ensure that the most crucial problems are resolved in a timely manner.

. NTP SERVER AND CLIENT

Setting up an NTP (Network Time Protocol) server and client is a common practice to synchronise the system clocks across multiple machines in a network. The NTP server provides accurate time to the clients, ensuring that all devices have the same time reference. Here's a step-by-step guide to setting up an NTP server and client on Ubuntu and CentOS:

- Install NTP server package:**

Open a terminal and run the following command to install the NTP server package on Ubuntu:v

```
himanshu@123:~$ sudo apt update
[sudo] password for himanshu:
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [42.9 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [858 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [40.0 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [101 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted i386 Packages [30.4 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [668 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [278 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:13 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [7 996 B]
```

```
himanshu@123:~$ sudo apt install ntp
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ntp is already the newest version (1:4.2.8p15+dfsg-1ubuntu2).
The following packages were automatically installed and are no longer required:
  debugedit libfsverity0 librpmbuild9 librpmsign9
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

```
himanshu@123:~$ sntp --version
sntp 4.2.8p15@1.3728-o Wed Feb 16 17:13:02 UTC 2022 (1)
himanshu@123:~$
```

```
himanshu@123:~$ sudo service ntp status
● ntp.service - Network Time Service
  Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2023-08-01 14:45:49 IST; 16s ago
    Docs: man:ntpd(8)
  Process: 13581 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited, start=0ms)
 Main PID: 13587 (ntpd)
    Tasks: 2 (limit: 4600)
   Memory: 1.3M
      CPU: 35ms
     CGroup: /system.slice/ntp.service
             └─13587 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 129:137

Aug 01 14:45:53 123 ntpd[13587]: Soliciting pool server 129.154.46.154
Aug 01 14:45:53 123 ntpd[13587]: Soliciting pool server 139.84.137.53
Aug 01 14:45:53 123 ntpd[13587]: Soliciting pool server 164.100.255.122
Aug 01 14:45:53 123 ntpd[13587]: Soliciting pool server 192.46.210.39
Aug 01 14:45:54 123 ntpd[13587]: Soliciting pool server 95.216.144.226
Aug 01 14:45:54 123 ntpd[13587]: Soliciting pool server 13.126.27.131
Aug 01 14:45:54 123 ntpd[13587]: Soliciting pool server 91.189.94.4
Aug 01 14:45:55 123 ntpd[13587]: Soliciting pool server 185.125.190.56
Aug 01 14:45:55 123 ntpd[13587]: Soliciting pool server 192.46.215.60
Aug 01 14:45:55 123 ntpd[13587]: Soliciting pool server 192.46.212.117
```

● Configure NTP client:

The configuration file for the NTP client is also located at /etc/ntp.conf. Open the file in a text editor:

- **sudo nano /etc/ntp.conf**

```
# Use servers from the NTP Pool Project. Approved by Ubuntu Technical Board
# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/join.html for
# more information.
pool 0.ubuntu.pool.ntp.org iburst
pool 1.ubuntu.pool.ntp.org iburst
pool 2.ubuntu.pool.ntp.org iburst
pool 3.ubuntu.pool.ntp.org iburst

# Use Ubuntu's ntp server as a fallback.
pool ntp.ubuntu.com

# Access control configuration; see /usr/share/doc/ntp-doc/html/accpt.html for
# details. The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>
# might also be helpful.
#
# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
# up blocking replies from your own upstream servers.

# By default, exchange time with everybody, but don't allow configuration.
restrict -4 default kod notrap nomodify nopeer noquery limited
restrict -6 default kod notrap nomodify nopeer noquery limited

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1

# Needed for adding pool entries
restrict source notrap nomodify noquery

# Clients from this (example!) subnet have unlimited access, but only if
# cryptographically authenticated.
#restrict 192.168.123.0 mask 255.255.255.0 notrust

# If you want to provide time to your local subnet, change the next line.
# (Again, the address is an example only.)
#broadcast 192.168.123.255

# If you want to listen to time broadcasts on your local subnet, de-comment the
# next lines. Please do this only if you trust everybody on the network!
#disable auth
#broadcastclient
"/etc/ntp.conf" 60L, 2136B
```

- **sudo service ntp restart**

- **sudo systemctl restart ntpd**

- **Run command ntpq -p :-** The output will display a list of NTP servers and their synchronisation status. If the synchronisation is successful, the reach value should be 377, and the offset value should be close to zero.

```
himanshu@123:~$ ntpq -p
      remote          refid      st t when poll reach   delay    offset  jitter
===== 
0.ubuntu.pool.n .POOL.        16 p    -  64     0    0.000  +0.000  0.000
1.ubuntu.pool.n .POOL.        16 p    -  64     0    0.000  +0.000  0.000
2.ubuntu.pool.n .POOL.        16 p    -  64     0    0.000  +0.000  0.000
3.ubuntu.pool.n .POOL.        16 p    -  64     0    0.000  +0.000  0.000
ntp.ubuntu.com .POOL.        16 p    -  64     0    0.000  +0.000  0.000
-ntp6.mum-in.hos 124.216.164.14  2 u   24  64    17 130.375 -49.053 63.341
-ntp7.mum-in.hos 36.224.68.195  2 u   22  64    17 178.704 -73.391 87.073
+time.blr1.levi. 217.180.209.213 2 u   18  64    17  69.645 -7.140 32.255
+time.cloudflare 10.57.8.5       3 u   20  64    17   6.748 -0.381 78.126
#static.15.192.2 194.58.200.20  2 u   16  64    17 724.325 -277.17 251.785
+kyc2docs.x10cor 192.46.215.141 3 u   21  64    17 100.228 -15.869 33.844
#ec2-13-126-27-1 169.254.169.123 4 u   12  64    17 817.859 -382.46 359.310
-129.154.46.154 132.163.97.1   2 u   17  64    17 166.228 -53.338 56.206
+ntp.nic.in      14.139.60.107  2 u   13  64    17 194.089 +0.884 27.863
+ntp-pool.time.4 139.59.55.93   6 u   13  64    17  89.549 -21.581 32.689
#185.125.190.56 145.238.203.14  2 u   20  64    17 227.844 -37.673 26.632
-157.245.102.2  129.134.28.123  2 u   18  64    17 166.387 -59.439 61.597
#185.125.190.58 86.23.195.30   2 u   19  64    17 250.223 -33.344 265.192
*ntp.qntmnet.com 133.243.238.243 2 u   13  64    17  33.599 -6.697 22.148
#185.125.190.57 201.68.88.106  2 u   19  64    17 276.249 -65.952 38.875
himanshu@123:~$
```

. LAMP STAK

LAMP (Linux, Apache, MySQL, and PHP) stack is a widely used hosting stack made up of open-source software used for hosting both static and dynamic web pages. It comprises the Apache web server, MariaDB or MySQL database engine, and PHP which is a backend scripting language. In this guide, you will learn how to install the LAMP stack on Ubuntu 22.04.

Step 1: Install Apache

The apache web server is available in the Ubuntu repository. Therefore to install it, simply run the following command:

- `sudo apt install apache2`

```
himanshu@123:~$ sudo apt install apache2
[sudo] password for himanshu:
Sorry, try again.
[sudo] password for himanshu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  base58 bridge-utils containerd cppcheck cscope cython3 fonts-lato fonts-lyx
  gamin gdal-data graphviz hdf5-helpers i965-va-driver ibverbs-providers
  icu-devtools intel-media-va-driver libaacs0 libaec-dev libaec0 libann0
  libao3 libarmadillo10 libarpack2 libasio-dev libastyle3 libavcodec58
  libavformat58 libavutil56 libbdplus0 libblas-dev libblas3 libbblosc1
  libbluray2 libboost-date-time-dev libboost-date-time1.74-dev
  libboost-date-time1.74.0 libboost-dev libboost-regex-dev
  libboost-regex1.74-dev libboost-serialization1.74-dev
  libboost-serialization1.74.0 libboost1.74-dev libc6-i386 libcaf-openmpi-3
  libcdt5 libcfitsio9 libcgns-dev libcgns3.4 libcgraph6 libcharls2
  libchromaprint1 libclang-cpp13 libcoarrays-openmpi-dev libcodeblocks0
  libcodec2-1.0 libcrypto++8 libcurl4-openssl-dev libdav1d5 libde265-0
  libdraco4 libevent-dev libevent-extra-2.1-7 libevent-openssl-2.1-7
  libexpat1-dev libfabric1 libffii7 libftw3-double3 libfreexl1 libfyba0
  libgamin0 libgdal30 libgeos-c1v5 libgeos3.10.2 libgeotiff5 libgl2ps1.4
  libglew2.2 libgme0 libgsm1 libgts-0.7-5 libgts-bin libgvc6 libgvpr2
  libhdf4-0-alt libhdf5-103-1 libhdf5-cpp-103-1 libhdf5-dev
  libhdf5-fortran-102 libhdf5-hl-100 libhdf5-hl-cpp-100 libhdf5-hl-fortran-100
  libhdf5-mpi-dev libhdf5-openmpi-103-1 libhdf5-openmpi-cpp-103-1
  libhdf5-openmpi-dev libhdf5-openmpi-fortran-102 libhdf5-openmpi-hl-100
  libhdf5-openmpi-hl-cpp-100 libhdf5-openmpi-hl-fortran-100 libheif1
  libhwloc-dev libhwloc-plugins libhwloc15 libibverbs-dev libibverbs1
  libicu-dev libigdgmm12 libjpeg-dev libjpeg-turbo8-dev libjpeg8-dev
  libjs-highlight.js libjs-jquery libjs-jquery-ui libjs-sphinxdoc
  libjs-underscore libkmlbase1 libkmldom1 libkmlengine1 liblab-gamut1
  liblapack3 liblaszip8 liblbfsgsb0 liblldb-13 libllvm13 libmedc-dev libmedc11
  libmfx1 libminizip1 libmysqlclient21 libnetcdf19 libnl-3-dev
  libnl-route-3-dev libnorm1 libnuma-dev libodbc2 libodbcinst2 libogdi4.1
  libopenblas-dev libopenblas-pthread-dev libopenblas0 libopenblas0-pthread
```

- sudo systemctl status apache2

```
himanshu@123:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
   Active: active (running) since Tue 2023-08-01 15:43:04 IST; 5s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 5600 (apache2)
    Tasks: 55 (limit: 9124)
   Memory: 5.2M
      CPU: 35ms
    CGroup: /system.slice/apache2.service
            ├─5600 /usr/sbin/apache2 -k start
            ├─5601 /usr/sbin/apache2 -k start
            └─5602 /usr/sbin/apache2 -k start

Aug 01 15:43:04 123 systemd[1]: Starting The Apache HTTP Server...
Aug 01 15:43:04 123 apachectl[5599]: AH00558: apache2: Could not reliably deter>
Aug 01 15:43:04 123 systemd[1]: Started The Apache HTTP Server.
```

- sudo systemctl enable --now apache2
 - sudo ufw allow 'Apache'
 - <http://server-ip>



Ubuntu

Apache2 Default Page

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 for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2` and is managed using `systemd`, so to start/stop the service use `systemctl start apache2` and `systemctl stop apache2`, and use `systemctl status apache2` and `journalctl -u apache2` to check status. `system` and `apache2ctl` can also be used for service management if desired. **Calling `/usr/bin/apache2` directly will not work with the default configuration.**

Step 2: Install MariaDB database server

MariaDB is an open source relational database server that is a fork and a drop-in replacement of MySQL. It provides better performance, speed, and replication when compared to MySQL thanks

to its advanced features such as storage engines, Galera clustering, and many more.

To install mariaDB on Ubuntu 22.04 run the following command:

- sudo apt install mariadb-server

```
himanshu@123: $ sudo apt install mariadb-server
[sudo] password for himanshu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
base58 bridge-util containerd cppcheck cscope cython3 fonts-lato fonts-lyx gamin gdal-data graphviz hdf5-helpers i965-va-driver ibverbs-providers
icu-devtools intel-media-va-driver libaacs0 libbaec-dev libbaec0 libbam0 libarmadillo0 libbarpack2 libasio-dev libastyle3 libavcodec58
libavformat58 libavutil56 libbdlplus0 libblas-dev libblas3 libblosc1 libbltray2 libboost-date-time-dev libboost-date-time1.74-dev
libboost-date-time1.74.0 libboost-dev libboost-regex-dev libboost-regex1.74-dev libboost-serialization1.74-dev libboost-serialization1.74.0
libboost1.74-dev libc6-i386 libcaf-openmpi-3 libcdt libcfitsio libcgns-dev libcgns3.4 libcgraph6 libcharls2 libchromaprint libclang-cpp13
libcoarrays-openmpi-dev libcodeblocks0 libcodec2-1.0 libcrypto++8 libcurl4-openssl-dev libdav1d5 libde265-0 libdrac04 libevent-dev
libevent-extra-2.1-7 libevent-openssl-2.1.7 libexpat1-dev libfabric1 libffi7 libfftw3-double3 libfreeex1 libfyba0 libgamin0 libdral30 libgeos-c1v5
libgeos3.10.2 libgeotiff5 libgl2ps1.4 libglew2.2 libgme0 libgsml libgts-0.7.5 libgts-bin libgvc6 libgvpr2 libhdf4-0-alt libhdf5-103-1
libhdf5-cpp-103-1 libhdf5-dev libhdf5-fortran-102 libhdf5-hl-100 libhdf5-hl-cpp-100 libhdf5-hl-fortran-100 libhdf5-mpi-dev libhdf5-openmpi-103-1
libhdf5-openmpi-cpp-103-1 libhdf5-openmpi-dev libhdf5-openmpi-fortran-102 libhdf5-openmpi-hl-100 libhdf5-openmpi-hl-cpp-100
libhdf5-openmpi-hl-fortran-100 libheif1 libhwloc-dev libhwloc-plugins libhwloc15 libibverbs libibverbs-dev libicu-dev libigdgmm12 libjpeg-dev
libineq-turbo8-dev libineq8-dev libis-highlight is libis-inquery libis-jquery-ui libis-sphinxdoc libis-underscore libkmdbase1 libkmdom1 libkmengines
```

- sudo systemctl status mariadb

```
himanshu@123: $ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.6.12 database server
  Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2023-08-01 15:45:58 IST; 13s ago
    Docs: man:mariadb(8)
          https://mariadb.com/kb/en/library/systemd/
 Process: 6946 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld (code=exited, status=0/SUCCESS)
 Process: 6947 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
 Process: 6949 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= || VAR= cd /usr/bin/..; /usr/bin/galera_recovery ; [ $?
 Process: 6993 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
 Process: 6995 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
 Main PID: 6978 (mariadb)
   Status: "Taking your SQL requests now..."
     Tasks: 16 (limit: 9124)
    Memory: 61.5M
      CPU: 302ms
     CGroup: /system.slice/mariadb.service
             └─6978 /usr/sbin/mariadb

Aug 01 15:45:57 123 mariadb[6978]: 2023-08-01 15:45:57 [Note] /usr/sbin/mariadb: ready for connections.
Aug 01 15:45:57 123 mariadb[6978]: Version: '10.6.12-MariaDB-0ubuntu0.22.04.1' socket: '/run/mysqld/mysqld.sock' port: 3306 Ubuntu 22.04
Aug 01 15:45:58 123 systemd[1]: Started MariaDB 10.6.12 database server.
Aug 01 15:45:58 123 /etc/mysql/debian-start[6997]: Upgrading MySQL tables if necessary.
Aug 01 15:45:58 123 /etc/mysql/debian-start[7000]: Looking for 'mariadb' as: /usr/bin/mariadb
Aug 01 15:45:58 123 /etc/mysql/debian-start[7000]: Looking for 'mariadb-check' as: /usr/bin/mariadb-check
Aug 01 15:45:58 123 /etc/mysql/debian-start[7000]: This installation of MariaDB is already upgraded to 10.6.12-MariaDB.
Aug 01 15:45:58 123 /etc/mysql/debian-start[7000]: There is no need to run mysql_upgrade again for 10.6.12-MariaDB.
Aug 01 15:45:58 123 /etc/mysql/debian-start[7000]: You can use --force if you still want to run mysql_upgrade
Aug 01 15:45:58 123 /etc/mysql/debian-start[7012]: Triggering myisam-recover for all MyISAM tables and aria-recover for all Aria tables
himanshu@123: $
```

- sudo systemctl enable mariadb

```
himanshu@123:~$ sudo systemctl enable mariadb
Synchronizing state of mariadb.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mariadb
himanshu@123:~$
```

● sudo mysql_secure_installation

```
himanshu@123:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
... Success!

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB 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? [Y/n] y
... Success!
```

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

```
Disallow root login remotely? [Y/n] y
... Success!
```

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

```
Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!
```

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

```
Reload privilege tables now? [Y/n] y
... Success!
```

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

```
Thanks for using MariaDB!
```

```
himanshu@123:~$ mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 52
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

Step 3: Install PHP and the Required Extensions

By default, Ubuntu 22.04 provides PHP 8.1 from its repositories. As such, you can install it using the following command.

- `sudo apt install php8.1 libapache2-mod-php8.1`

```
himanshu@123: $ sudo apt install php8.1 libapache2-mod-php8.1
[sudo] password for himanshu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
base58 bridge-utils containerd cppcheck cscope cython3 fonts-lato fonts-lyx gamin gdal-data graphviz hdf5-helpers
i965-v-a-driver ibverbs-providers icu-devtools intel-media-v-a-driver libaacs0 libaec-dev libaec0 libann0 libaom3
libarmadillo0 libarpack2 libasio-dev libastyle3 libavcodec58 libavformat58 libavutil56 libbdplus0 libblas-dev
libblas3 libblosc1 libbluray2 libboost-date-time-dev libboost-date-time1.74-dev libboost-date-time1.74.0 libboost-dev
libboost-regex-dev libboost-regex1.74-dev libboost-serialization1.74-dev libboost-serialization1.74.0 libboost1.74-dev
libc6-i386 libcaf-openmpi-3 libcdt5 libcfitsio9 libcngns-dev libcngns3.4 libcgraph6 libcharls2 libchromaprint1
libclang-cpp13 libcoarrays-openmpi-dev libcodeblocks0 libcodec2-1.0 libcrypto++8 libcurl4-openssl-dev libdavid5
libde265-0 libdraco4 libevent-extra-2.1-7 libevent-openssl-2.1-7 libexpat1-dev libfabric1 libffi7
libfftw3-double3 libfreetype1 libfyba0 libgamin0 libgdal30 libgeos-c1v5 libgeos3-10.2 libgeotiff5 libgl2ps1.4 libglew2.1
libgme0 libgsmlibgts0.7-5 libgts-bin libgvc6 libgvpr2 libhdf4-0-alt libhdf5-103-1 libhdf5-cpp-103-1 libhdf5-dev
libhdf5-fortran-102 libhdf5-hl-100 libhdf5-hl-cpp-100 libhdf5-hl-fortran-100 libhdf5-mpi-dev libhdf5-openmpi-103-1
libhdf5-openmpi-cpp-103-1 libhdf5-openmpi-dev libhdf5-openmpi-fortran-102 libhdf5-openmpi-hl-100
libhdf5-openmpi-hl-cpp-100 libhdf5-openmpi-hl-fortran-100 libheif1 libhwloc-dev libhwloc-plugins libhwloc15
libibverbs-dev libibverbs1 libicu-dev libigdmm12 libjpeg-dev libjpeg-turbo8-dev libjpeg8-dev libjs-highlight.js
libjs-jquery libjs-jquery-ui libjs-sphinxdoc libjs-underscore libkmlbase1 libkmldom1 liblab-gamut1
liblapack3 liblaszip8 liblfgs0 liblldb-13 liblvm13 libmedc-dev libmedc11 libminizip1 libnetcdf19
libnl3-dev libnl-route-3-dev libnorm1 libnuma-dev libodbc2 libodbcinst2 libogdi4.1 libopenblas-dev
libopenblas-pthread-dev libopenblas0 libopenblas0-pthread libopenmpi-dev libopenmpi3 libopenmpt0 libpathplan4
libpdal-base13 libpdal-plugin-draco libpdal-plugin-e57 libpdal-plugin-faux libpdal-plugin-hdf libpdal-plugin-i3s
libpdal-plugin-icebridge libpdal-plugin-pgpointcloud libpdal-plugins libpdal-util13 libpgm-5.3.0 libpmix-dev libpmix2
libpq5 libproj2 libpsm-infinopath1 libpsm2-2 libpython3-dev libpython3.10-dev libqhull-r8.0 libqt5concurrent5
libqt5designer5 libqt5help5 libqt5sql5 libqt5sql5-sqlite libqt5test5 libqt5xml5 librabbitmq4 librdmacm1 librttopo1
```

- `php -v`

Next, you might also consider installing some PHP extensions. Here, we have installed a few PHP extensions

- sudo apt install php8.1-mysql php8.1-xml
 php8.1-xmlrpc php8.1-curl php8.1-gd
 php8.1-imagick php8.1-cli php8.1-imap
 php8.1-mbstring php8.1-opcache php8.1-soap
 php8.1-zip php8.1-intl -y

```
himanshu@123: $ sudo apt install php8.1-mysql php8.1-xml php8.1-xmlrpc php8.1-curl php8.1-gd php8.1-imagick php8.1-cli
p8.1-imap php8.1-mbstring php8.1-opcache php8.1-soap php8.1-zip php8.1-intl -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
php8.1-imagick is already the newest version (3.6.0-4ubuntu1).
php8.1-xmlrpc is already the newest version (3:1.0.0~rc3-2).
php8.1-cli is already the newest version (8.1.2-1ubuntu2.13).
php8.1-curl is already the newest version (8.1.2-1ubuntu2.13).
php8.1-gd is already the newest version (8.1.2-1ubuntu2.13).
php8.1-mbstring is already the newest version (8.1.2-1ubuntu2.13).
php8.1-mysql is already the newest version (8.1.2-1ubuntu2.13).
php8.1-opcache is already the newest version (8.1.2-1ubuntu2.13).
php8.1-xml is already the newest version (8.1.2-1ubuntu2.13).
php8.1-imap is already the newest version (8.1.2-1ubuntu2.13).
php8.1-intl is already the newest version (8.1.2-1ubuntu2.13).
php8.1-soap is already the newest version (8.1.2-1ubuntu2.13).
php8.1-zip is already the newest version (8.1.2-1ubuntu2.13).
The following packages were automatically installed and are no longer required:
base58 bridge-utils containerd cppcheck cscope cython3 fonts-lato fonts-lyx gamin gdal-data graphviz hdf5-helpers
i965-v4l2-drv ibverbs-providers icu-devtools intel-media-v4l2-drv libaacs0 libaec-dev libaec0 libann0 libarmadillo1
libarpack2 libasio-dev libastyle3 libavcodec58 libavformat58 libavutil56 libblasplus0 libblas-dev libblas3 libblosc1
libbluray2 libboost-date-time-dev libboost-date-time1.74-dev libboost-date-time1.74.0 libboost-dev libboost-regex-dev
libboost-regex1.74-dev libboost-serialization1.74-dev libboost-serialization1.74.0 libboost1.74-dev libc6-i386
libcaf-openmpi-3 libcdt5 libcfitsio9 libcgns-dev libcgns3.4 libcgraph6 libcharls2 libchromaprint1 libclang-cpp13
libcoarrays-openmpi-dev libcodeblocks0 libcodeblocks2-1.0 libcrypto++8 libcurl4-openssl-dev libdraco4 libevent-dev
libevent-extra-2.1.7 libevent-openssl-2.1.7 libexpat1-dev libfabric1 libffi7 libfreetype1 libfyba0 libgamin0 libgdal30
libgeos-c1v5 libgeos3.10.2 libgeotiff5 libgl2ps1.4 libglew2.2 libgme0 libgsm1 libqts-0.7-5 libqts-bin libqvc6 libqvpr
```

Next, let's create a PHP script to test out PHP integration with Apache. Run the following command:

- echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php

```
himanshu@123:~$ echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
<?php phpinfo(); ?>
himanshu@123:~$
```

- <http://server-ip/info.php>



System	Linux 123 5.19.0-50-Ubuntu SMP PREEMPT_DYNAMIC Mon Jul 10 18:24:29 UTC 2023 x86_64
Build Date	Jun 28 2023 14:01:49
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902,NTS
PHP Extension Build	API20210902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
 Zend Engine v4.1.2, Copyright (c) Zend Technologies
 with Zend OPcache v8.1.2-1ubuntu2.13, Copyright (c), by Zend Technologies

zendengine

. CRON JOB

Setting up a cron job in Ubuntu 22.04 (or any version) involves a few simple steps. A cron job is a scheduled task that runs automatically at specific intervals or times. Here's how you can set up a cron job:

- sudo apt update

- sudo apt install cron
- sudo systemctl start cron
- sudo systemctl enable cron
- sudo systemctl status cron.service

```
himanshu@123:~$ sudo systemctl status cron.service
[sudo] password for himanshu:
● cron.service - Regular background program processing daemon
  Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: >
  Active: active (running) since Tue 2023-08-01 15:39:18 IST; 49min ago
    Docs: man:cron(8)
   Main PID: 745 (cron)
     Tasks: 1 (limit: 9124)
    Memory: 468.0K
      CPU: 23ms
     CGroup: /system.slice/cron.service
             └─745 /usr/sbin/cron -f -P

Aug 01 15:39:18 123 cron[745]: (CRON) INFO (pidfile fd = 3)
Aug 01 15:39:18 123 cron[745]: (CRON) INFO (Running @reboot jobs)
Aug 01 15:39:18 123 systemd[1]: Started Regular background program processing d>
Aug 01 16:09:01 123 CRON[18789]: pam_unix(cron:session): session opened for user>
Aug 01 16:09:01 123 CRON[18790]: (root) CMD ( [ -x /usr/lib/php/sessionclean ]>
Aug 01 16:09:01 123 CRON[18789]: pam_unix(cron:session): session closed for user>
Aug 01 16:17:01 123 CRON[18949]: pam_unix(cron:session): session opened for user>
Aug 01 16:17:01 123 CRON[18950]: (root) CMD ( cd / && run-parts --report /etc/>
Aug 01 16:17:01 123 CRON[18949]: pam_unix(cron:session): session closed for user>
himanshu@123:~$
```

- Create a cron job: To create a cron job, you need to edit the crontab file. The crontab file contains the list of scheduled tasks. Each user has their own crontab file, so you'll need to edit the relevant one.

- crontab -e

```
GNU nano 6.2                                         /tmp/cr
# 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
10 18 * * * /bin/sh /home/himanshu/Desktop/abc.sh
```

```
himanshu@123:~$ crontab -e
crontab: installing new crontab
himanshu@123:~$
```

- Verify the cron job: After saving the crontab, the cron service will automatically pick up the changes. You can check the list of scheduled cron jobs for the current user with the command:

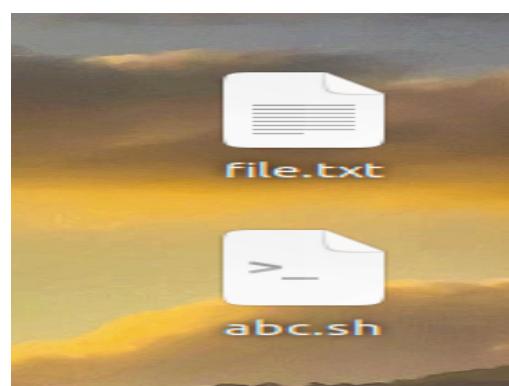
- `crontab -l`

```

himanshu@123:~/Desktop$ 
himanshu@123:~/Desktop$ 
himanshu@123:~/Desktop$ chmod u+x abc.sh
himanshu@123:~/Desktop$ 
himanshu@123:~/Desktop$ 
himanshu@123:~/Desktop$ 
himanshu@123:~/Desktop$ ls
abc.sh archive.tgz folder1 git grep Internship.AD JS
himanshu@123:~/Desktop$ 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

10 18 * * * /bin/sh /home/himanshu/Desktop/abc.sh
himanshu@123:~/Desktop$ crontab -e
No modification made
himanshu@123:~/Desktop$ 

```



. DNS

DNS stands for **Domain Name System (server)** . It is a crucial internet service that translates human-readable domain names (like www.example.com) into IP addresses (like 192.0.2.1) that computers use to identify each other on the network. Essentially, DNS acts as a directory service for the internet, allowing users to access websites and services by typing in easy-to-remember domain names rather than remembering complex IP addresses.

When you enter a domain name in your web browser, your computer sends a request to a DNS resolver (usually provided by your Internet Service Provider or a public DNS service like Google DNS or Cloudflare DNS). The DNS resolver then looks up the IP address associated with that domain name and returns it to your computer, enabling your browser to connect to the appropriate server hosting the website.

To install and configure a DNS server on Ubuntu 22.04, you can use the BIND (Berkeley Internet

Name Domain) software, which is a popular and widely used DNS server. Here's how to install it:

- **sudo apt update**

```
himanshu@123:~$ sudo apt update
[sudo] password for himanshu:
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadat
a [42.9 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [858
 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Met
adata [40.0 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metada
ta [101 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted i386 Packages
 [30.4 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packag
es [668 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 M
etadata [278 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11
 Metadata [940 B]
Get:13 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Met
adata [7 996 B]
```

- **sudo apt install bind9**

- sudo systemctl status bind9

```
himanshu@123:~$ sudo systemctl status bind9
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-08-02 12:52:10 IST; 9s ago
     Docs: man:named(8)
 Process: 29918 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 29919 (named)
    Tasks: 6 (limit: 9124)
   Memory: 7.8M
      CPU: 55ms
     CGroup: /system.slice/named.service
             └─29919 /usr/sbin/named -u bind

Aug 02 12:52:10 123 named[29919]: network unreachable resolving './DNSKEY/IN': 2001:500:2d::#53
Aug 02 12:52:10 123 named[29919]: network unreachable resolving './NS/IN': 2001:500:2d::#53
Aug 02 12:52:10 123 named[29919]: network unreachable resolving './DNSKEY/IN': 2001:500:12::d0d#53
Aug 02 12:52:10 123 named[29919]: network unreachable resolving './NS/IN': 2001:500:12::d0d#53
Aug 02 12:52:10 123 named[29919]: network unreachable resolving './DNSKEY/IN': 2001:500:a8::e#53
Aug 02 12:52:10 123 named[29919]: network unreachable resolving './NS/IN': 2001:500:a8::e#53
Aug 02 12:52:10 123 named[29919]: network unreachable resolving './DNSKEY/IN': 2001:dc3::35#53
Aug 02 12:52:10 123 named[29919]: network unreachable resolving './NS/IN': 2001:dc3::35#53
Aug 02 12:52:10 123 named[29919]: managed-keys-zone: Initializing automatic trust anchor management
Aug 02 12:52:10 123 named[29919]: resolver priming query complete: success
```

```
himanshu@123:~$ ping www.google.com
PING www.google.com (172.217.167.228) 56(84) bytes of data.
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=1 ttl=60 time=161 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=2 ttl=60 time=6.76 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=3 ttl=60 time=23.2 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=4 ttl=60 time=5.13 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=5 ttl=60 time=8.49 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=6 ttl=60 time=6.01 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=7 ttl=60 time=5.85 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=8 ttl=60 time=7.24 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=9 ttl=60 time=7.30 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=10 ttl=60 time=5.60 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=11 ttl=60 time=7.15 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=12 ttl=60 time=6.00 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=13 ttl=60 time=6.67 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=14 ttl=60 time=6.55 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=15 ttl=60 time=5.99 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=16 ttl=60 time=6.37 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=17 ttl=60 time=8.25 ms
64 bytes from del11s04-in-f4.1e100.net (172.217.167.228): icmp_seq=18 ttl=60 time=5.66 ms
```

- By default, BIND is configured to work as a caching DNS resolver. This means it will forward DNS requests to other DNS servers on the internet. You can modify its configuration file located at `/etc/bind/named.conf.options` to set up your own DNS zones and records.
- Please note that configuring a DNS server correctly requires a good understanding of DNS concepts and network administration. It's essential to ensure the security and proper functioning of your DNS server. If you're setting up a DNS server for a production environment or a critical application, it's advisable to consult with a network

administrator or a professional before proceeding. Incorrectly configured DNS servers can lead to issues and security vulnerabilities.

- If you don't have a specific need to run your own DNS server, you can use the default DNS resolver provided by your Internet Service Provider or public DNS servers like Google DNS (8.8.8.8 and 8.8.4.4) or Cloudflare DNS (1.1.1.1 and 1.0.0.1). These are reliable and well-maintained options for most users.