



**FACULTY OF COMPUTER SCIENCE
AND MATHEMATICS**

UNIVERSITY OF TECHNOLOGY MARA (UiTM)

**BACHELOR OF COMPUTER SCIENCE (HONS.) NETCENTRIC
COMPUTING (CDCS251)**

**NETWORK AND SYSTEM ADMINISTRATION
(ITT420)**

LAB 3 REPORT

BY:

**NURUL AINA FIRZANAH BINTI ROSNAN
(2023388953)**

**PREPARED FOR:
DR. NOORHAYATI BINTI MOHAMED NOOR**

1.0 INTRODUCTION

Background services are essential to support core system and network functions of Linux Operating System. These services handle important tasks such as managing network connections, hosting web applications, and maintaining databases. For this reason, it required a proper effective service management to maintain system stability, performance, and security. Through this service management, it allows administrators to control which services are running, when they start, and how they behave during system boot. However, if these service configured incorrectly, it can consume system resources or it might introduce potential security risks. Therefore, understanding how to monitor and manage services is an important skill for anyone that involved in system administration.

The goal of this lab exercise is to focus on exploring and increase understanding of Linux service management concepts. The lab manual given by lecturer demonstrates service management using the chkconfig command and SysVinit-based systems. However, the latest Kali Linux uses the systemd framework, which replaces SysVinit and introduces a different approach to service management. As a result, the systemctl command is used as an equivalent tool to perform service management tasks. Common service management operations that carried out in this lab include listing services, checking service status, starting and stopping services, and configuring services to start automatically during system boot.

2.0 OBJECTIVE

The objectives of this lab exercise are:

- To understand the concept of Linux services and daemons.
- To identify common web-related services running on a Linux system.
- To learn how to check the status of services.
- To start and stop services manually.
- To enable and disable services at system boot using systemd.

3.0 TASK

The following steps were performed to complete this lab exercise:

Task 1: Gaining Administrative Priviledge	
Procedure & Output	<p>The system was accessed using administrative privileges (root user) through command ‘sudo -i’</p> <pre>(ainazanahnn㉿kali)-[~] \$ sudo -i (Message from Kali developers) This is a minimal installation of Kali Linux, you likely want to install supplementary tools. Learn how: → https://www.kali.org/docs/troubleshooting/common-minimum-setup/ (Run: "touch ~/.hushlogin" to hide this message)</pre>
	<p>Then, it followed by the command ‘whoami’ command to verify the root user</p> <pre>(root㉿kali)-[~] # whoami root</pre>
Discussion	<p>For this lab, the administrative privilege are required because the operation involved in this lab to manage Linux service can affect the entire system and these operations include starting, stopping and configuring the service. All of these operations are restricted for normal users unlike root user that have privilege to manage these operations. The output of the ‘whoami’ command displayed root, this confirming that administrative access was successfully obtained.</p>

Task 2: Listing Available Linux Services			
Procedure & Output	<p>All available Linux services were listed using the ‘systemctl list-unit-files --type=service’ command. This command displays services managed by systemd along with their startup status.</p> <pre>(root㉿kali)-[~] # systemctl list-unit-files --type=service UNIT FILE STATE PRESET accounts-daemon.service enabled enabled apache-htcacheclean.service disabled disabled apache-htcacheclean@.service disabled disabled apache2.service disabled disabled apache2@.service disabled disabled apparmor.service disabled disabled apt-daily-upgrade.service static - apt-daily.service static - atftpd.service indirect disabled auth-rpcgss-module.service static - autovt@.service alias - avahi-daemon.service disabled disabled bluetooth.service disabled disabled bolt.service static - capsule@.service static - colord.service static - configure-printer@.service static - console-getty.service disabled disabled console-setup.service enabled enabled container-getty@.service static - containerd.service disabled disabled cron.service enabled enabled cryptdisks-early.service masked disabled cryptdisks.service masked disabled dbus-org.freedesktop.hostname1.service alias - dbus-org.freedesktop.locale1.service alias - dbus-org.freedesktop.login1.service alias - dbus-org.freedesktop.ModemManager1.service alias -</pre>		
Discussion	<p>The command output displayed a list of installed services and their respective states, such as enabled or disabled. The enabled means that the service starts automatically, while disabled means it must be started manually. The ‘list-unit-files’ option in command will lists all the service unit files, while ‘--type=service’ option in command help to filters the output to show only services. By listing all the available services, this help administrator to identify which services are installed and which service that are either start automatically during system boot or not. This type of information is important for system monitoring and service management</p>		

Task 3: Checking the Status of a Web Related Service	
Procedure & Output	<p>The status of selected Linux service was checked using ‘systemctl status’ command. For this lab the Apache and Mariadb was chosen.</p> <pre>(root㉿kali)-[~] └─# systemctl status apache2 ○ apache2.service - The Apache HTTP Server Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled) Active: inactive (dead) Docs: https://httpd.apache.org/docs/2.4/ Dec 19 21:21:56 kali systemd[1]: Starting apache2.service - The Apache HTTP Server... Dec 19 21:21:56 kali systemd[1]: Started apache2.service - The Apache HTTP Server. Dec 19 21:22:59 kali systemd[1]: Stopping apache2.service - The Apache HTTP Server... Dec 19 21:22:59 kali systemd[1]: apache2.service: Deactivated successfully. Dec 19 21:22:59 kali systemd[1]: Stopped apache2.service - The Apache HTTP Server. (root㉿kali)-[~] └─# systemctl status mariadb ○ mariadb.service - MariaDB 11.4.3 database server Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; preset: disabled) Active: inactive (dead) Docs: man:mariadb(8) https://mariadb.com/kb/en/library/systemd/</pre>
Discussion	<p>The output showed that the both Apache and Mariadb was installed but currently inactive and disabled, this indicate that the service was not running and will not start automatically when the system boot . By checking this status, it help administrators gain insights whether a service is active, inactive or disabled. This steps is really important before performing further service management operations.</p>

Task 4: Starting a Linux Service	
Procedure & Output	<p>Then the Apache and Mariadb was started using ‘systemctl start’ command</p> <pre>(root㉿kali)-[~] └─# systemctl start apache2 (root㉿kali)-[~] └─# systemctl start mariadb</pre> <p>After that, the Apache and Mariadb service status were checked again to verify that the service was indeed running.</p>

	<pre>[root@kali] ~] # systemctl status apache2 ● apache2.service - The Apache HTTP Server Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled) Active: active (running) since Fri 2025-12-19 21:32:33 +08; 17s ago Invocation: 201acd9205cd4c5386ffffac0fffa2ba8 Docs: https://httpd.apache.org/docs/2.4/ Process: 7809 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS) Main PID: 7812 (apache2) Tasks: 6 (limit: 9389) Memory: 14.7M (peak: 15.4M) CPU: 148ms CGroup: /system.slice/apache2.service ├─7812 /usr/sbin/apache2 -k start ├─7815 /usr/sbin/apache2 -k start ├─7816 /usr/sbin/apache2 -k start ├─7817 /usr/sbin/apache2 -k start ├─7818 /usr/sbin/apache2 -k start └─7819 /usr/sbin/apache2 -k start Dec 19 21:32:33 kali systemd[1]: Starting apache2.service - The Apache HTTP Server... Dec 19 21:32:33 kali systemd[1]: Started apache2.service - The Apache HTTP Server. </pre> <pre>[root@kali] ~] # systemctl status mariadb ● mariadb.service - MariaDB 11.4.3 database server Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; preset: disabled) Active: active (running) since Fri 2025-12-19 21:32:45 +08; 9s ago Invocation: 1af376547b0249c59c69b01e3dac87e2 Docs: man:mariadb(8) https://mariadb.com/kb/en/library/systemd/ Process: 7839 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysql Process: 7842 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSIT> Process: 7846 ExecStartPre=/bin/sh -c [! -e /usr/bin/galera_recovery] && VAR= > Process: 7871 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSI> Process: 7874 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS) Main PID: 7857 (mariadb) Status: "Taking your SQL requests now..." Tasks: 14 (limit: 61972) Memory: 227.4M (peak: 232.6M) CPU: 3.777s CGroup: /system.slice/mariadb.service └─7857 /usr/sbin/mariadb Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] InnoDB: Loading buffer> Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] Plugin 'FEEDBACK' is d> Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] Plugin 'wsrep-provider'> Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] InnoDB: Buffer pool(s)> Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] Server socket created > Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] Server socket created > Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] mariadb: Event Schedu> Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] /usr/sbin/mariadb: re> Dec 19 21:32:45 kali mariadb[7857]: Version: '11.4.3-MariaDB-1' socket: '/run/mysqld/m> Dec 19 21:32:45 kali systemd[1]: Started mariadb.service - MariaDB 11.4.3 database serve></pre>
Discussion	The output shows that the service status changed to active and running, this confirmed that the service had started and running successfully. By starting service like Apache and Mariadb , it allow the services to run immediately without requiring a system reboot. This is very useful when need services to be active on demand.

Task 5: Enabling a Linux Service at System Boot

Procedure & Output	<p>Next, the Apache and Mariadb services was enabled using ‘systemctl enable’ command.</p> <pre>(root㉿kali)-[~] # systemctl is-enabled apache2 enabled (root㉿kali)-[~] # systemctl is-enabled mariadb enabled</pre>
Procedure & Output	<p>Then startup of Apache and Mariadb status was verified through ‘systemctl is-enabled’ command</p> <pre>(root㉿kali)-[~] # systemctl status apache2 ● apache2.service - The Apache HTTP Server Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: disabled) Active: active (running) since Fri 2025-12-19 21:32:33 +08; 9min ago Invocation: 201acd9205cd4c5386ffffac0ffa2ba8 Docs: https://httpd.apache.org/docs/2.4/ Main PID: 7812 (apache2) Tasks: 6 (limit: 9389) Memory: 14.7M (peak: 15.4M) CPU: 188ms CGroup: /system.slice/apache2.service └─7812 /usr/sbin/apache2 -k start ├─7815 /usr/sbin/apache2 -k start ├─7816 /usr/sbin/apache2 -k start ├─7817 /usr/sbin/apache2 -k start ├─7818 /usr/sbin/apache2 -k start └─7819 /usr/sbin/apache2 -k start Dec 19 21:32:33 kali systemd[1]: Starting apache2.service - The Apache HTTP Server... Dec 19 21:32:33 kali systemd[1]: Started apache2.service - The Apache HTTP Server. </pre> <pre>(root㉿kali)-[~] # systemctl status mariadb ● mariadb.service - MariaDB 11.4.3 database server Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled) Active: active (running) since Fri 2025-12-19 21:32:45 +08; 9min ago Invocation: 1af376547b0249c59c69b01e3dac87e2 Docs: man:mariadb(8) https://mariadb.com/kb/en/library/systemd/ Main PID: 7857 (mariadb) Status: "Taking your SQL requests now..." Tasks: 9 (limit: 61972) Memory: 227.3M (peak: 232.6M) CPU: 3.902s CGroup: /system.slice/mariadb.service └─7857 /usr/sbin/mariadb Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] InnoDB: Loading buffer> Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] Plugin 'FEEDBACK' is d> Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] Plugin 'wsrep-provider'> Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] InnoDB: Buffer pool(s)> Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] Server socket created > Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] Server socket created > Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] mariadb: Event Schedu> Dec 19 21:32:45 kali mariadb[7857]: 2025-12-19 21:32:45 0 [Note] /usr/sbin/mariadb: re> Dec 19 21:32:45 kali mariadb[7857]: Version: '11.4.3-MariaDB-1' socket: '/run/mysqld/m-> Dec 19 21:32:45 kali systemd[1]: Started mariadb.service - MariaDB 11.4.3 database serve></pre>
Discussion	<p>The output displayed enabled and this verified that the services was configured to start automatically during the system boots. By enabling this services at boot, it make sure that the service are available immediately after system startup.</p>

Task 6: Stopping and Disabling a Linux Service

Procedure & Output	<p>After that. Apache and Mariadb service were stopped using ‘systemctl stop’ command.</p>
-------------------------------	--

```
(root㉿kali)-[~]
└─# systemctl stop apache2

[root@kali ~]#
└─# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: disabled)
   Active: inactive (dead) since Fri 2025-12-19 21:45:13 +08; 11s ago
     Duration: 12min 39.652s
   Invocation: 201acd9205cd4c5386fffa2ba8
      Docs: https://httpd.apache.org/docs/2.4/
    Process: 8983 ExecStop=/usr/sbin/apachectl graceful-stop (code=exited, status=0/SUCCESS)
   Main PID: 7812 (code=exited, status=0/SUCCESS)
     Mem peak: 26.8M
        CPU: 339ms

Dec 19 21:32:33 kali systemd[1]: Starting apache2.service - The Apache HTTP Server...
Dec 19 21:32:33 kali systemd[1]: Started apache2.service - The Apache HTTP Server.
Dec 19 21:45:13 kali systemd[1]: Stopping apache2.service - The Apache HTTP Server...
Dec 19 21:45:13 kali systemd[1]: apache2.service: Deactivated successfully.
Dec 19 21:45:13 kali systemd[1]: Stopped apache2.service - The Apache HTTP Server.
```

```
(root㉿kali)-[~]
└─# systemctl stop mariadb

[root@kali ~]#
└─# systemctl status mariadb
● mariadb.service - MariaDB 11.4.3 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
   Active: inactive (dead) since Fri 2025-12-19 21:45:46 +08; 14s ago
     Duration: 13min 151ms
   Invocation: 1af376547b0249c59c69b01e3dac87e2
      Docs: man:mariadb(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 7857 ExecStart=/usr/sbin/mariadb $MYSQLD_OPTS $_WSREP_NEW_CLUSTER $_WSREP_
   Main PID: 7857 (code=exited, status=0/SUCCESS)
     Status: "MariaDB server is down"
    Mem peak: 232.6M
        CPU: 3.971s

Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] InnoDB: FTS optimize t>
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] InnoDB: Starting shudt>
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] InnoDB: Dumping buffer>
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] InnoDB: Buffer pool(s)>
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] InnoDB: Removed tempor>
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] InnoDB: Shutdown compl>
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] /usr/sbin/mariadb: Sh>
Dec 19 21:45:46 kali systemd[1]: mariadb.service: Deactivated successfully.
Dec 19 21:45:46 kali systemd[1]: Stopped mariadb.service - MariaDB 11.4.3 database serve>
Dec 19 21:45:46 kali systemd[1]: mariadb.service: Consumed 3.971s CPU time, 232.6M memor>
```

The Apache and Mariadb then disabled using ‘systemctl disable’ command

```
(root㉿kali)-[~]
└─# systemctl disable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-insta
Executing: /usr/lib/systemd/systemd-sysv-install disable apache2
Removed '/etc/systemd/system/multi-user.target.wants/apache2.service'.

[root@kali ~]#
└─# systemctl disable mariadb
Synchronizing state of mariadb.service with SysV service script with /usr/lib/systemd/systemd-sysv-insta
Executing: /usr/lib/systemd/systemd-sysv-install disable mariadb
Removed '/etc/systemd/system/multi-user.target.wants/mariadb.service'.
```

```
[root@kali] ~]
# systemctl status apache2
○ apache2.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
  Active: inactive (dead)
    Docs: https://httpd.apache.org/docs/2.4/
    
Dec 19 21:21:56 kali systemd[1]: Starting apache2.service - The Apache HTTP Server...
Dec 19 21:21:56 kali systemd[1]: Started apache2.service - The Apache HTTP Server.
Dec 19 21:22:59 kali systemd[1]: Stopping apache2.service - The Apache HTTP Server...
Dec 19 21:22:59 kali systemd[1]: apache2.service: Deactivated successfully.
Dec 19 21:22:59 kali systemd[1]: Stopped apache2.service - The Apache HTTP Server.
Dec 19 21:32:33 kali systemd[1]: Starting apache2.service - The Apache HTTP Server...
Dec 19 21:32:33 kali systemd[1]: Started apache2.service - The Apache HTTP Server.
Dec 19 21:45:13 kali systemd[1]: Stopping apache2.service - The Apache HTTP Server...
Dec 19 21:45:13 kali systemd[1]: apache2.service: Deactivated successfully.
Dec 19 21:45:13 kali systemd[1]: Stopped apache2.service - The Apache HTTP Server.

[~]
# systemctl status mariadb
○ mariadb.service - MariaDB 11.4.3 database server
  Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; preset: disabled)
  Active: inactive (dead)
    Docs: man:mariadb(8)
          https://mariadb.com/kb/en/library/systemd/
    
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] InnoDB: FTS optimize
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] InnoDB: Starting shutd
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] InnoDB: Dumping bufer
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] InnoDB: Buffer pool(s)
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] InnoDB: Removed tempor
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] InnoDB: Shutdown comp
Dec 19 21:45:46 kali mariadb[7857]: 2025-12-19 21:45:46 0 [Note] /usr/sbin/mariadb: S
Dec 19 21:45:46 kali systemd[1]: mariadb.service: Deactivated successfully.
Dec 19 21:45:46 kali systemd[1]: Stopped mariadb.service - MariaDB 11.4.3 database serv
Dec 19 21:45:46 kali systemd[1]: mariadb.service: Consumed 3.971s CPU time, 232.6M mem

lines 1-16/16 (END)
```

Discussion	The output shows that both the Apache and Mariadb status showed inactive, and disabled. This mean the services are not running and will not start automatically. By stopping and disabling a services, this can help the system to conserve resources and improve their security by ensuring that unnecessary services are not running or starting automatically.
-------------------	---

4.0 FINDING/DISCUSSION

Based on the lab activities performed, it was found that the latest Kali Linux uses systemd framework for service management instead of SysVinit system tha described in the lab manual. As a result, ‘systemctl’ command was used as an equivalent replacement for ‘chkconfig’ to manage the system services.Moreover, it was found that the administrative access is required to perform Linux management services operations as the operations done in this lab involved operation that can affect the entire operating system and these operations includes starting, stopping, enabling and disabling the services. Only user that have administrative privilege are allowed to perform these operations for security reason.

Other that that, another findings found during lab activities is that listing available services can help identify which service are installed on the system including their startup status. The service that marked as enabled indicate the it will start automatically during boot system while service that marked as disabled indicate that it must be started manually.Additionally, it was found that a service can be installed but remain inactive and disabled. This shows that the service is currently not running and will not start automatically during system boot. After starting and enabling the service, its status changed to active and enabled. This changes shows that both the running state and startup behavior of a service can be controlled by administrators using the systemctl command. Conversely, stopping and disabling the service will returned the service back to an inactive state and prevented it from starting automatically at system boot.

The lab manual given by the lecturer demonstrates service management using the mysqld service on SysVinit-based systems. However, the latest version of Kali Linux uses systemd framework and does not include MySQL by default. Therefore, the equivalent service management tasks were demonstrated using Apache and Mariadb services, which are commonly available and widely used in Kali Linux. In older Linux systems, services are managed using run levels and the chkconfig command. But the latest Kali Linux replaces this mechanism with systemd, which uses systemctl and targets to manage service startup behavior. Although the commands and underlying systems differ, the core service management concepts remain the same.

5.0 CONCLUSION

In conclusion, this lab provided a hands-on practical in managing some common web related services that run on a Kali Linux server like Apache and Mariadb. This lab covered important tasks such as listing available services in Kali Linux, checking their status, start and stop the services, as well as control service startup behavior. All of these tasks demonstrate how administrators manage system resources for all the services. Although the lab manual uses chkconfig, run levels and SysVinit, the same service management concepts were applied using systemctl in a systemd-based environment. Overall, this lab helped improve my understanding of Linux service management and the importance of controlling system services for security and system stability.

REFERENCES

- Ellingwood, J. (2025, April 19). *Manage systemd services with systemctl on linux*. DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-use-systemctl-to-manage-systemd-services-and-units>
- Garn, D. (2022, May 17). How to use systemctl to manage linux services. Red Hat - We make open source technologies for the enterprise. https://www.redhat.com/en/blog/linux-systemctl-manage-services?utm_source=chatgpt.com
- GeeksforGeeks. (2025, November 13). Managing system services in linux: Systemctl command. <https://www.geeksforgeeks.org/linux-unix/systemctl-in-unix/>