Debian

Disable Unnecessary Services Debian Linux

2 years ago • by David Adams

This article aims to help you increase performance and decrease vulnerabilities by reducing the number of services at minimal level as possible. By reducing the services instructed in this tutorial, domestic users who need regular access to the internet shouldn't experience problems but only improvements, even if not visible.

This article is optimized for those looking for immediate changes or the syntax to manage services on Debian, additionally you can find descriptive information on Systemd at the end.

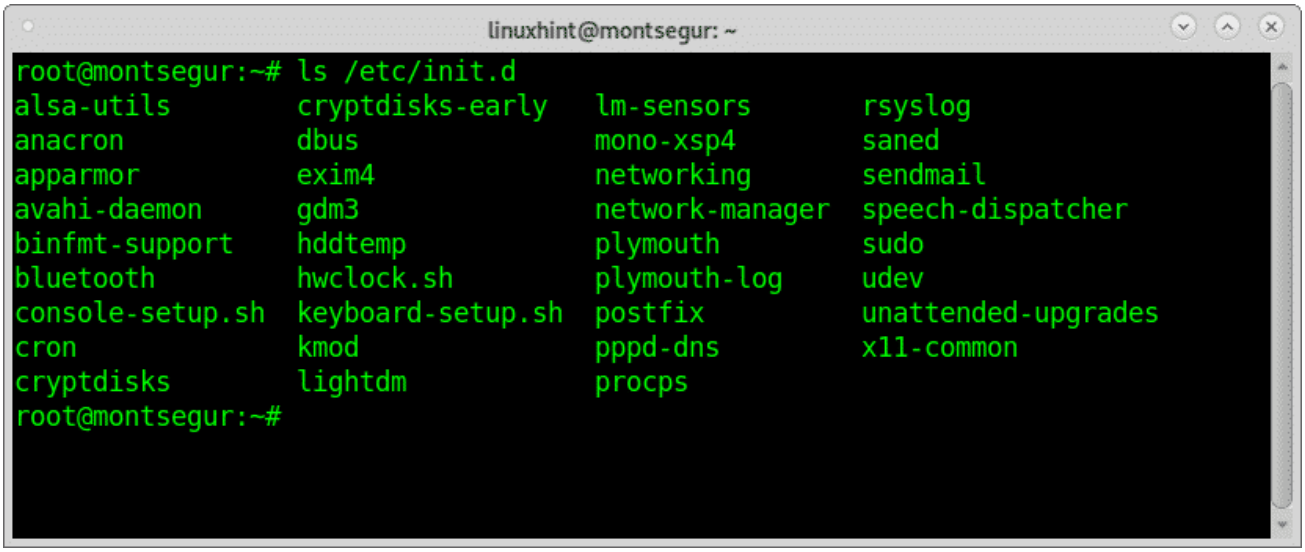
To begin let's see what services are running to decide what service to discard, to list all services run:

```
# sudo service --status-all
```

You will see many services with a plus symbol, these are the services running while the services with a minus symbol are not active.

As you can see all listed services are found at **/etc/init.d** which is where services are stored. You can run ls on /etc/init.d to check:

```
# ls /etc/init.d
```



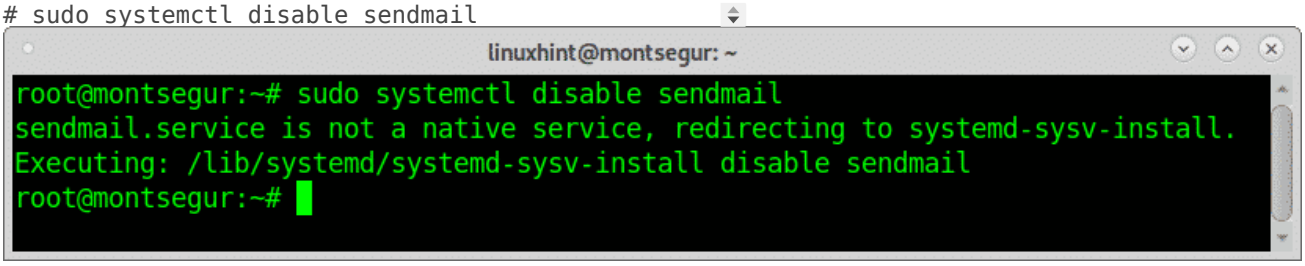
In my case, a desktop user with a dynamic IP I don't want the mail service because setting UP SPF, DKIM, etc. may not be impossible but messy, therefore I want to disable the active sendmail service. Some other services like ssh, apache, etc. should be removed if you don't use them.

The syntax to disable a service on Debian is:

```
# sudo systemctl disable <service>
```

The command to disable sendmail on Debian is

```
# sudo systemctl disable sendmail
```

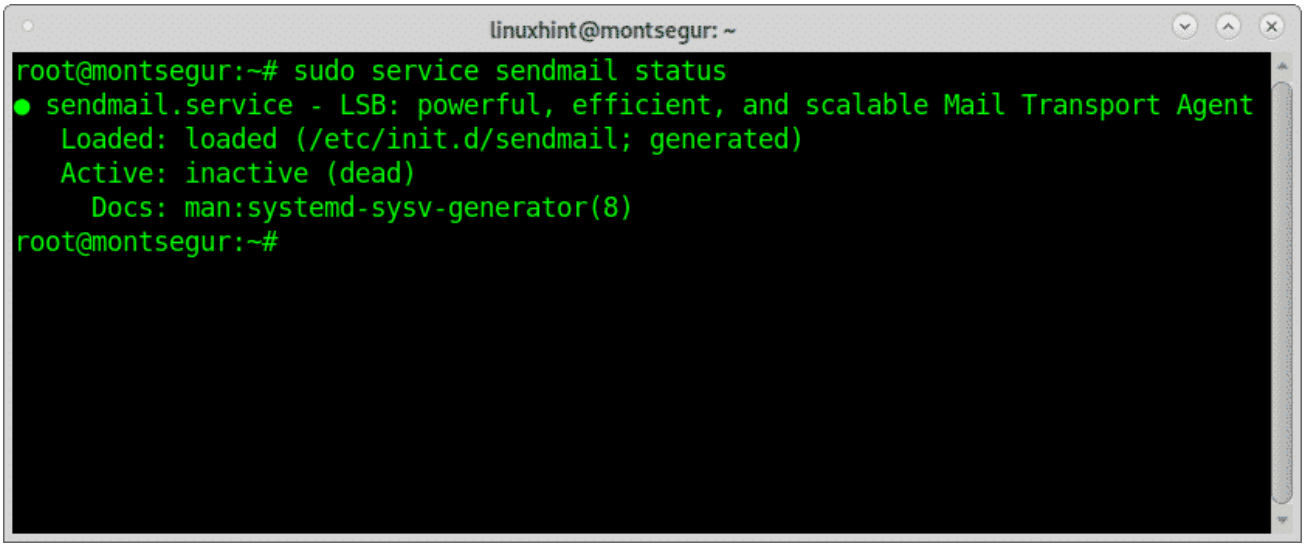


As you can see now the service is disabled

```
# sudo service --status-all
```

You can also check for the specific service

```
# sudo service sendmail status
```

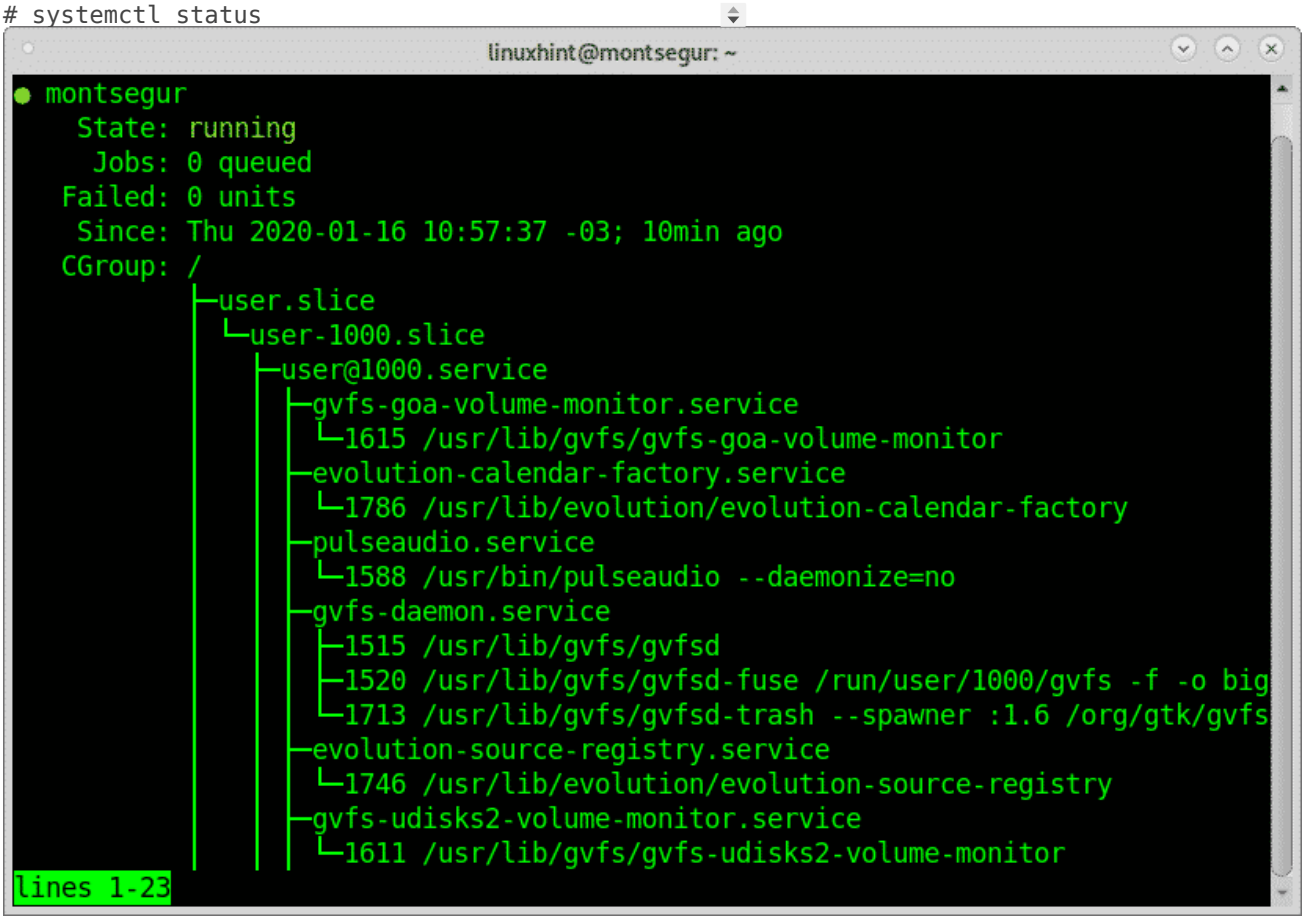


Currently most Linux distributions don't use System V anymore but use Systemd. Systemd is a service manager, it is the PID 1, assigns a control group (cgroup) to each service and can be also useful to track the processes.

Through it you can configure services with the system and service manage which initializes in the last boot process step and manages processes executed by the user.

Below there is a list of commands to manage services through SystemD:

```
# systemctl status
```

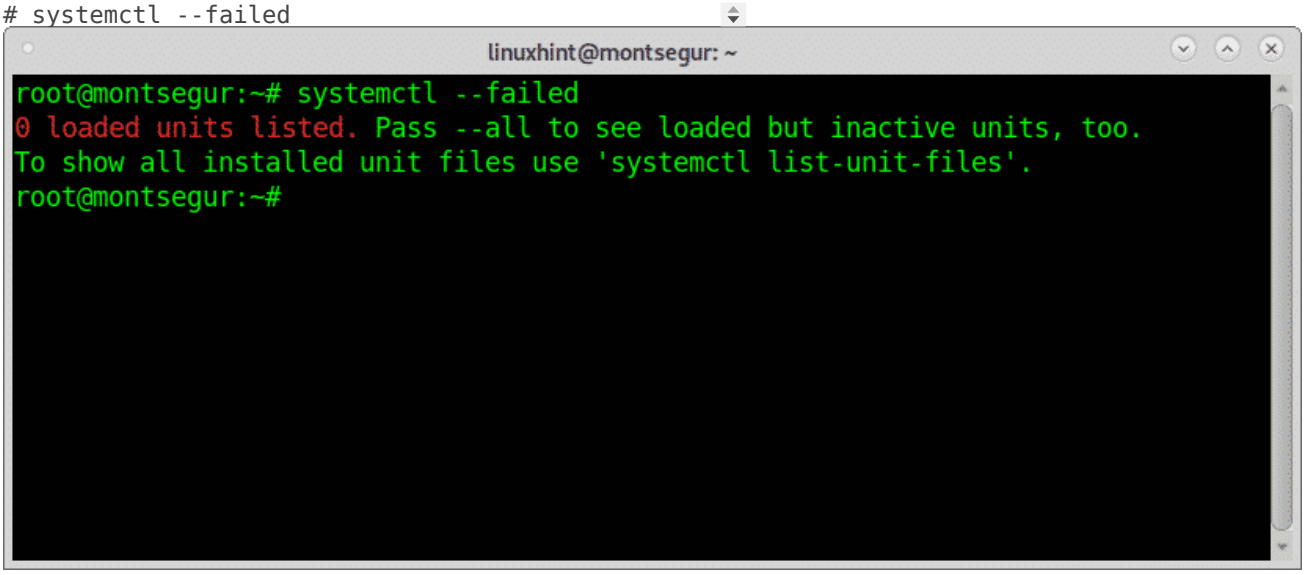


As you can see in the output above SystemD is running.

The following commands lists failed units, services or daemons which didn’t start properly due a misconfiguration, unmatched dependency, etc.

Running the following command is a way to audit the device’s health.

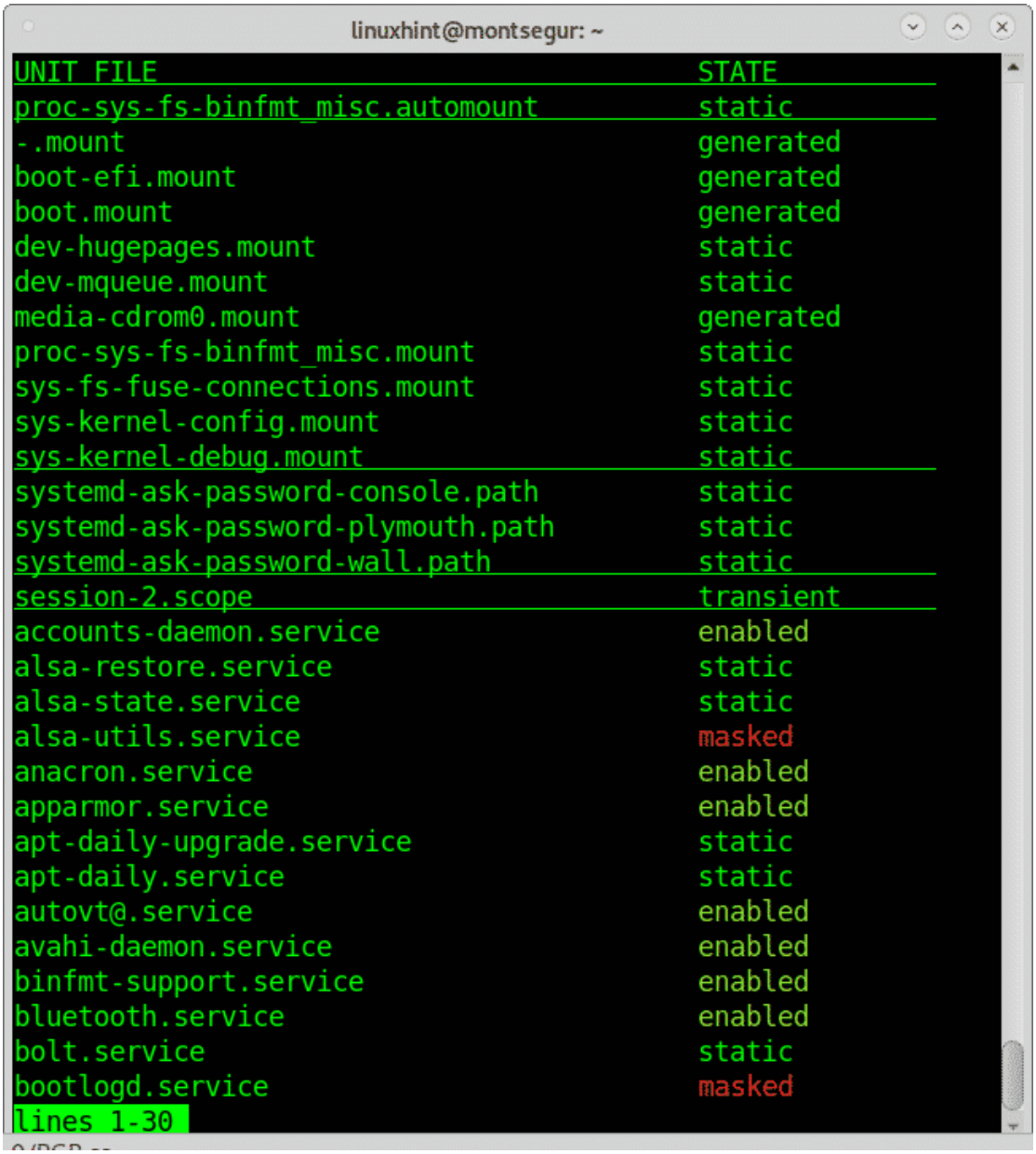
```
# systemctl --failed
```



Unit files contain information on sockets, devices, mount points, swap or partition (.service, .socket, .device, .mount, .automount, .swap, .target, .path, .timer, .slice, or .scope). They may contain information on more options. If Systemd fails to identify an option it will log warnings, options starting by X- are ignored.

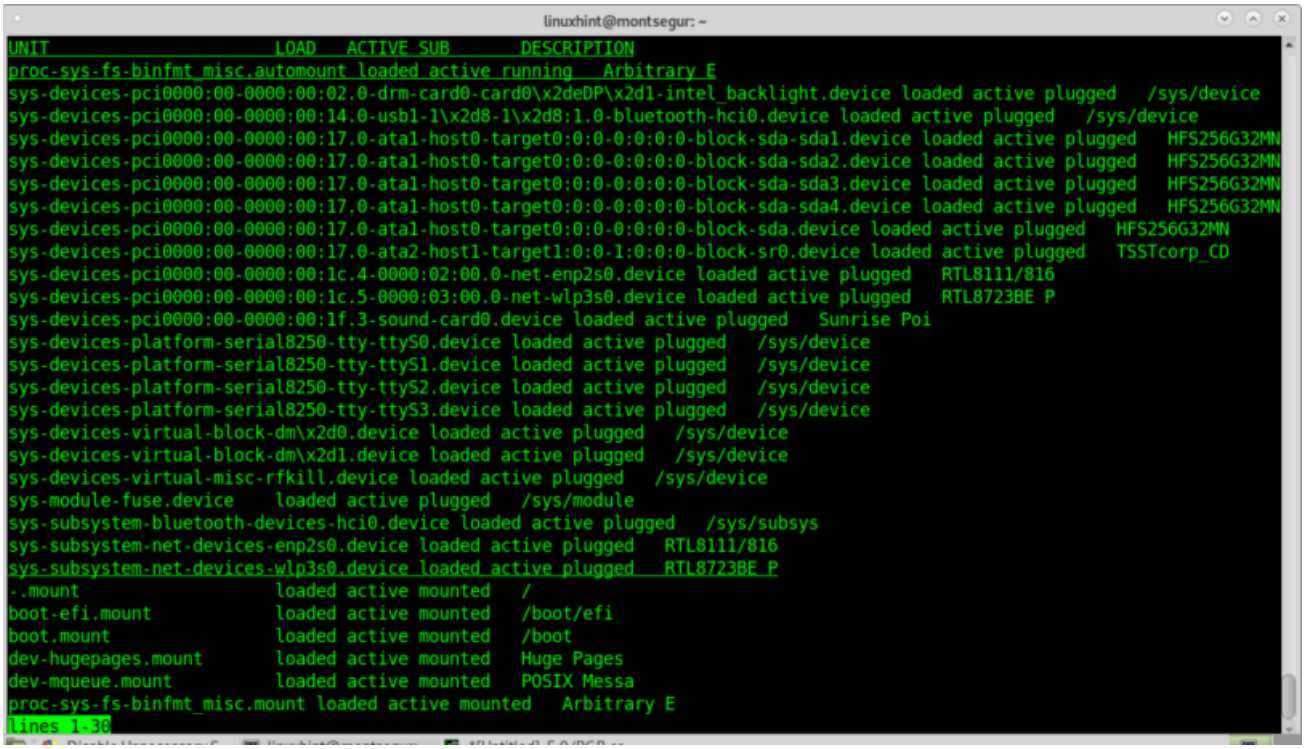
To list installed unit files run the following command:

```
# systemctl list-unit-files
```



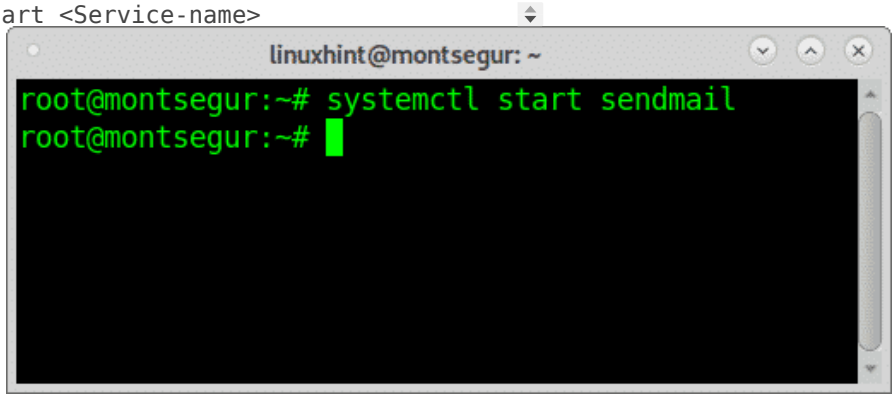
List all running services:

```
# systemctl
```



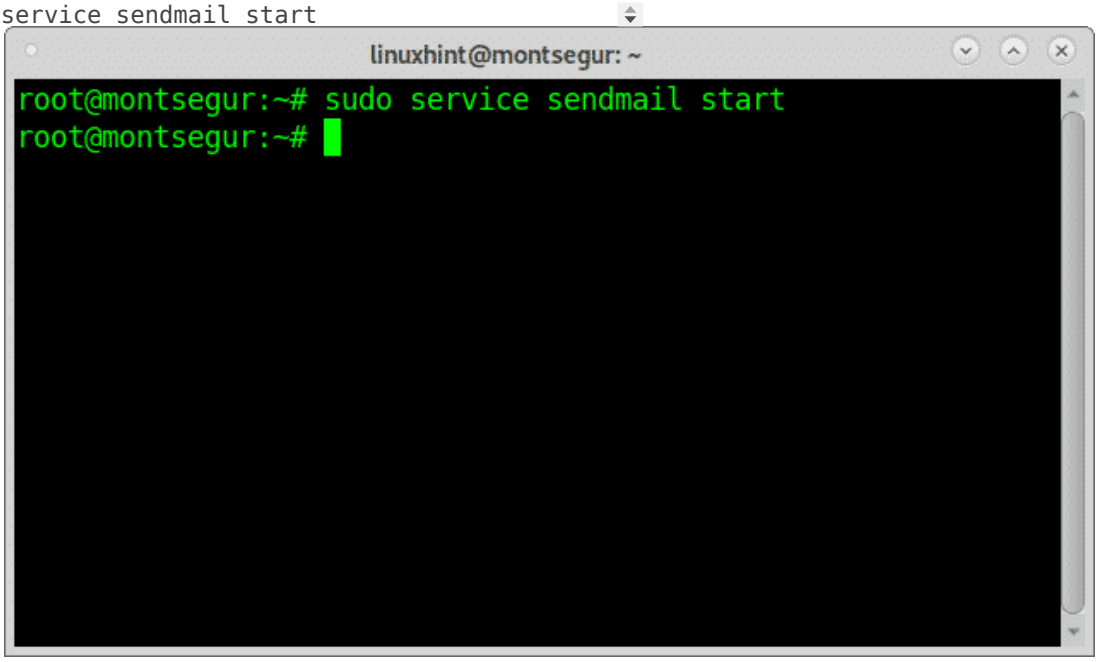
To start a service using systemd run:

```
# systemctl start <Service-name>
```



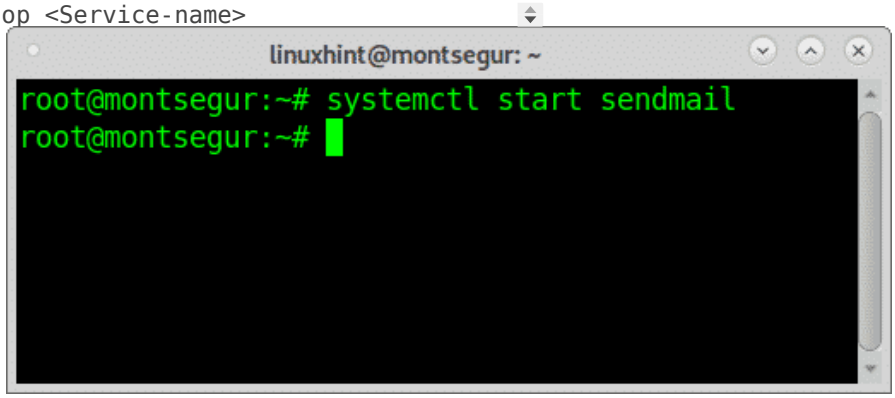
Alternatively you can run:

```
# sudo service sendmail start
```



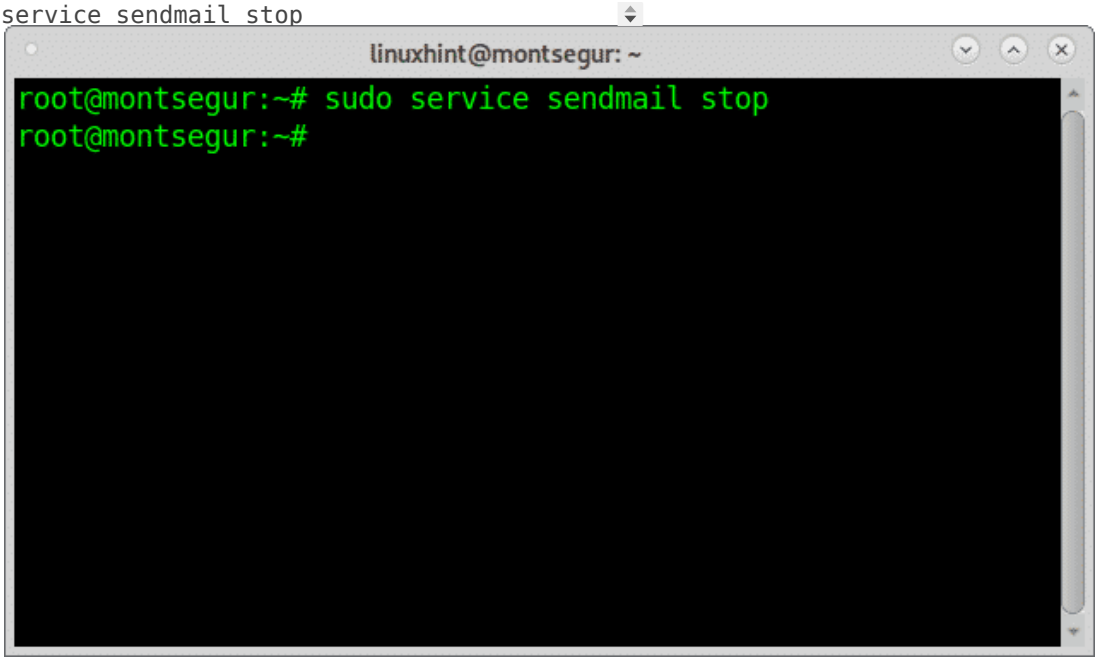
To stop services use the same syntax replacing “start” with “ stop”, to start the <Service-name> run:

```
# systemctl stop <Service-name>
```



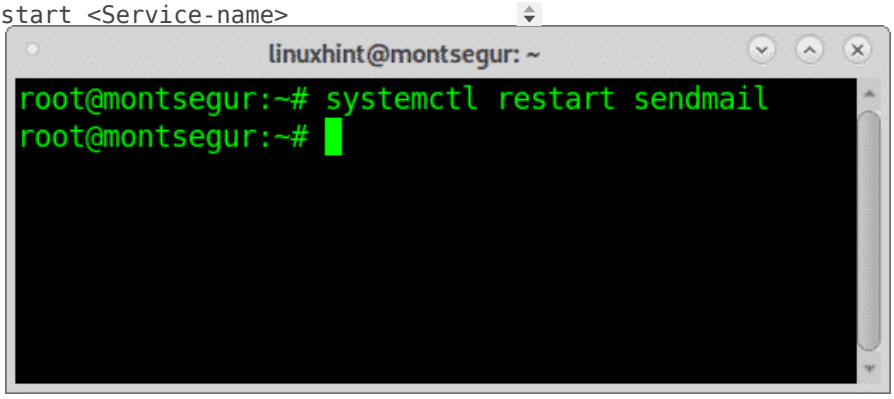
Similarly, you can use the command “service”:

```
# sudo service sendmail stop
```



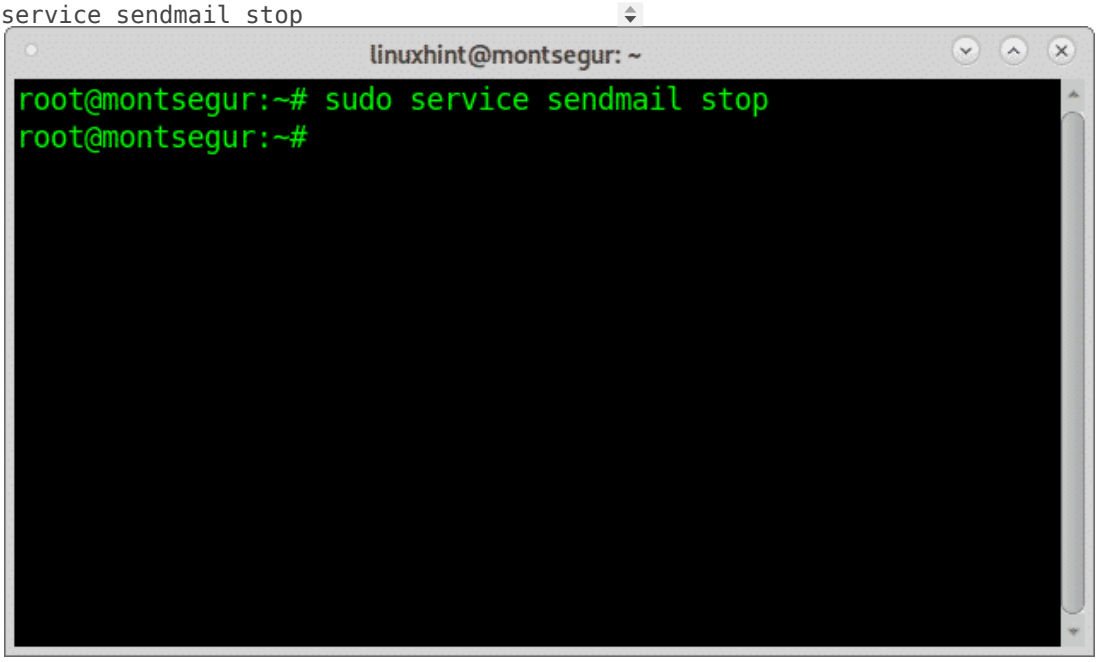
To restart a service replace “stop” or “start” for “restart”, the following example shows how to start the service “<Service-name>”:

```
# systemctl restart <Service-name>
```



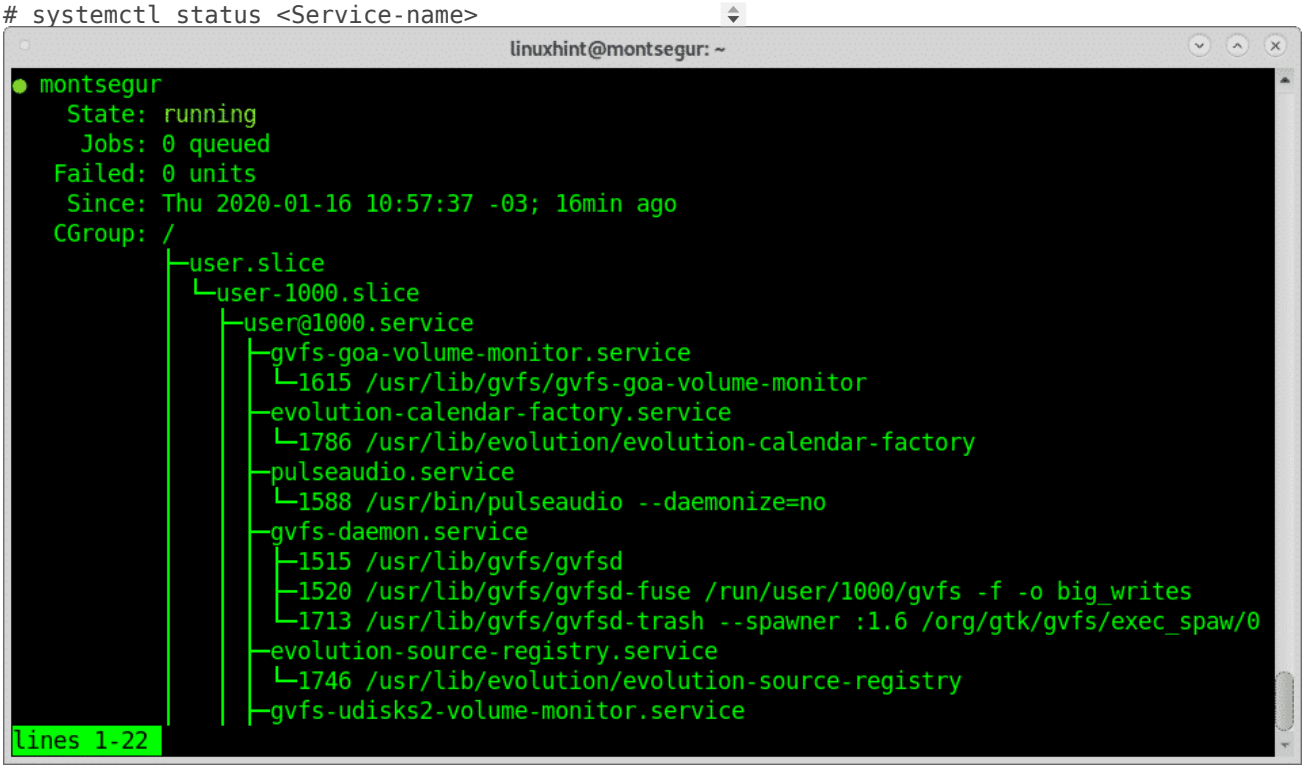
You can also command “service” to restart:

```
# sudo service sendmail stop
```



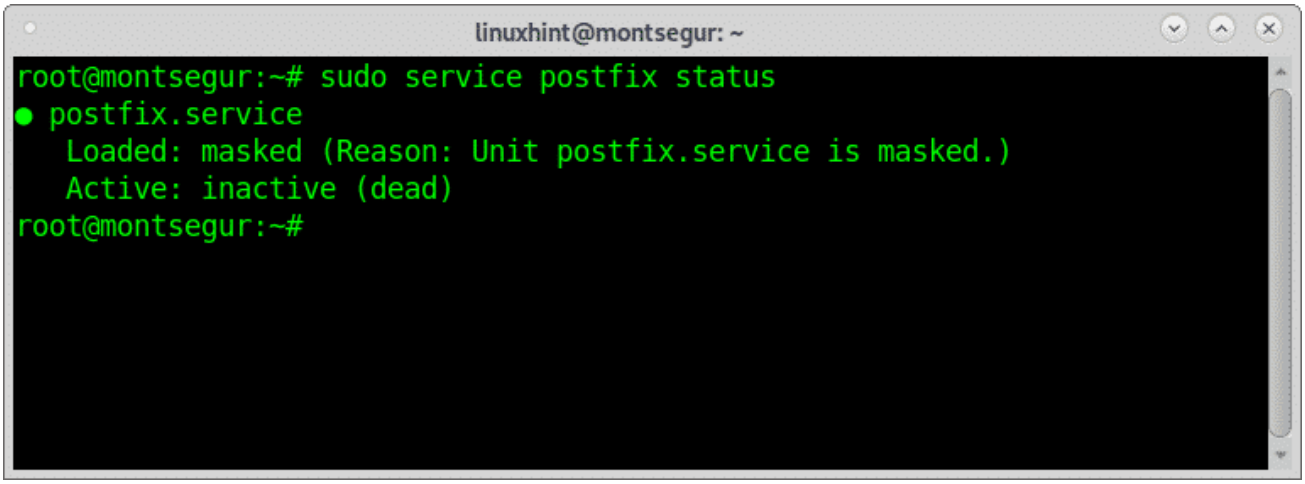
To show the status of a specific service use the option “status”, the following example shows how to start the service <Service-Name>:

```
# systemctl status <Service-name>
```



You can check a service status using the command “service”:

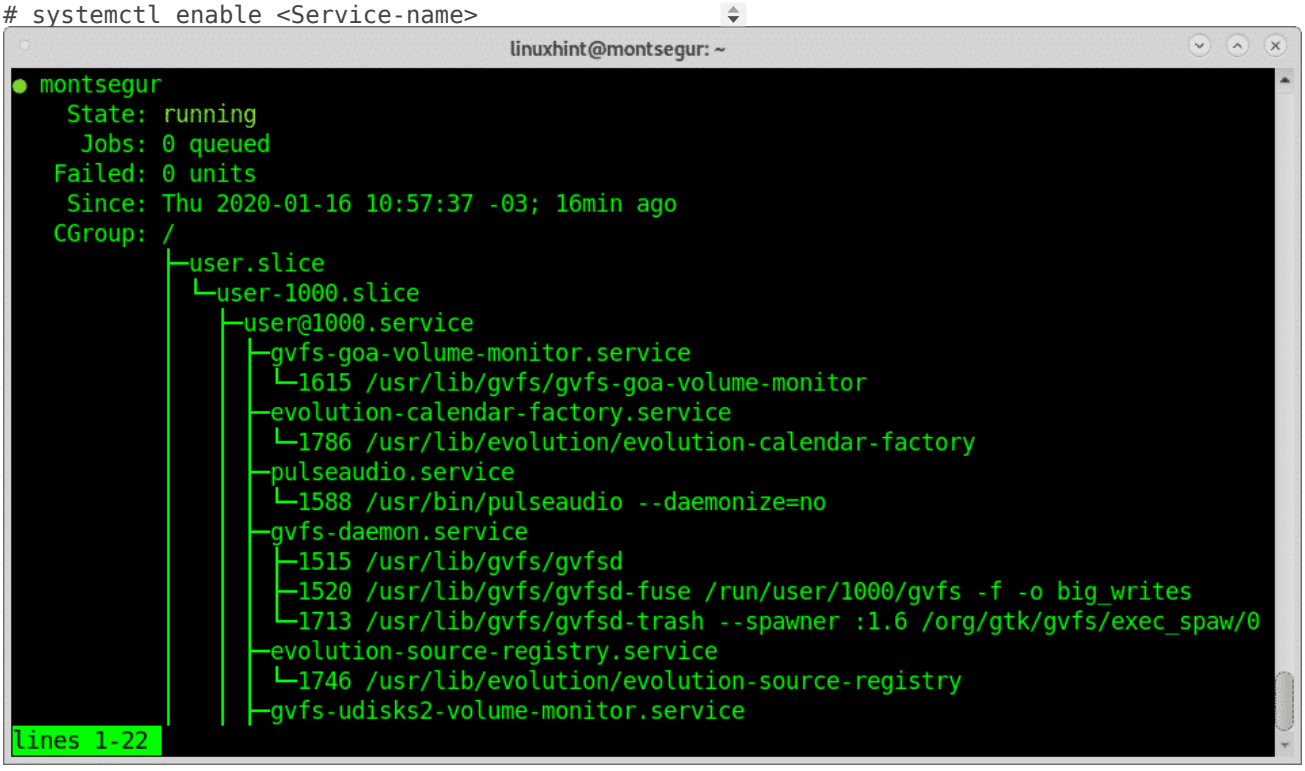
```
# sudo service postfix status
```



As you can see the service says to be masked which means it is strongly disabled and can't be enabled even manually (it can be unmasked but it won't be shown in this tutorial). The following instructions show how to enable services are meant for disabeld services, not for masked ones.

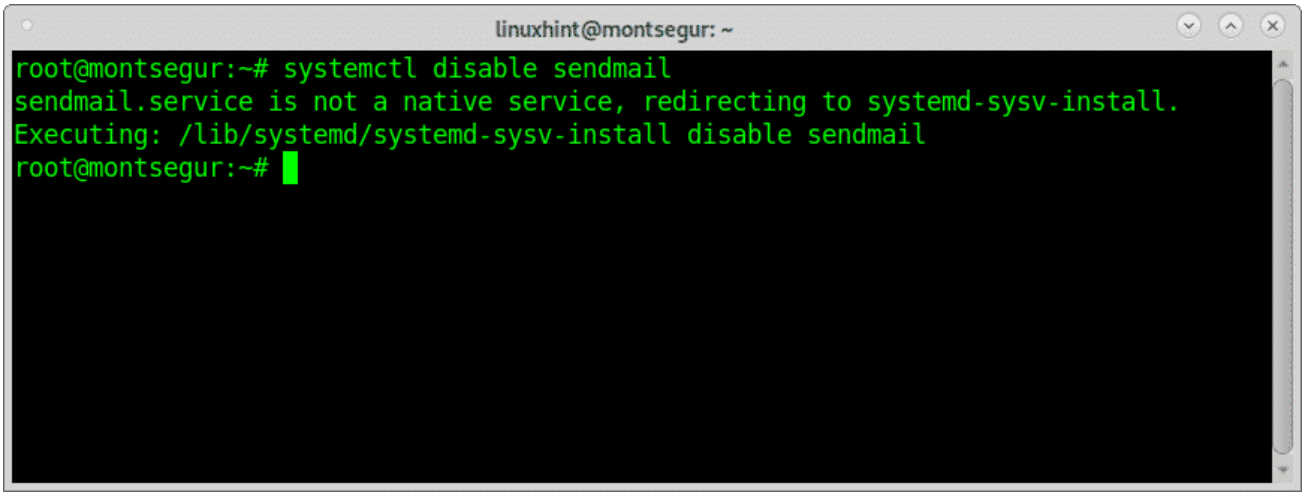
To enable a service to be activated every time the device boots use the option ***enable***, the following example shows how to enable the <Service-Name> to be started on boot:

```
# systemctl enable <Service-name>
```



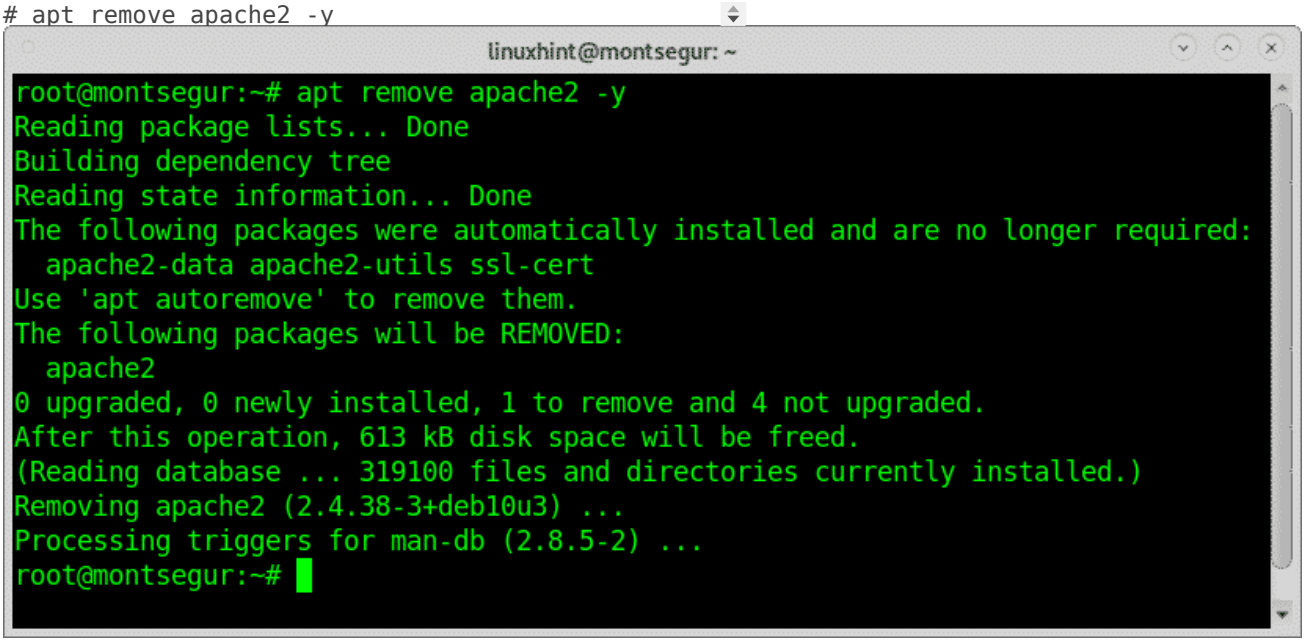
To disable a service to remain inactive even after the device boots use the option ***disable***, the following example shows how to enable the <Service-Name> **no** to be started on boot:

```
# systemctl disable <Service-name>
```



Additionally to disabling services you can also, and it is recommended, to uninstall any service you are sure you won't use, and even if you plan in the future you can install it, to remove a service, rather than disabling or stopping it, for example to uninstall Apache on Debian run:

```
# apt remove apache2 -y
```



You can do the same with all installed services you don't use such as ssh, cups, etc.

There are additional options for systemctl you can find on its man page or online at http://man7.org/linux/man-pages/man1/systemctl.1.html.

I hope you found this article on Disabling Unnecessary Services Debian Linux useful.

ABOUT THE AUTHOR

David Adams

David Adams is a System Admin and writer that is focused on open source technologies, security software, and computer systems.

View all posts

RELATED LINUX HINT POSTS

- How to Install Edge Browser on Debian
- How to Install and Configure OneDrive on Debian 11
- How to Install Debian 11 in Virtual Box
- Install PIP on Debian 11
- How do I Check Disk Space in Debian-LinuxHint
- How to use systemd in Debian
- 10 Best File Compression Tools in Debian

Linux Hint LLC, editor@linuxhint.com
1210 Kelly Park Cir, Morgan Hill, CA 95037