How to avoid restarting the server with Ubuntu Livepatch

If you administer your own server (s), sooner or later you will encounter this problem. You must restart the operating system, but the machine that is providing an important service cannot be interrupted.

If you administer your own server (s), sooner or later you will encounter this problem. You must restart the operating system, but the machine that is providing an important service cannot be interrupted.

But why restart the server? Everything seems to work well after the `apt-get upgrade` command. However, the truth is not always the same. Although the system continues to run after each upgrade and is not required to restart like Windows, you may still need to do this.

For example, when a vulnerability in the kernel of the kernel is detected, it will be patched and pushed to your server as a new package. After you install the patched kernel, some files are written to the drive, but it is still the old kernel, because it is the file that is loaded into memory (RAM).

This means that your server is still vulnerable to previously discovered security vulnerabilities. Other processes, daemons, and services can be reloaded without restarting the operating system. However, the kernel is at the center of the system and can only be reloaded at the next boot.

Ubuntu Livepatch solves this by allowing you to close kernel security holes without rebooting. This way, you can avoid or delay rebooting for weeks or months without compromising security.

The core idea behind the Live Patching feature is simple: When a function is vulnerable to 'write', rewrite it, remove the vulnerability and load the new function somewhere into memory. When the function is called, instead of running the code in the kernel, redirect it to use the rewritten code.

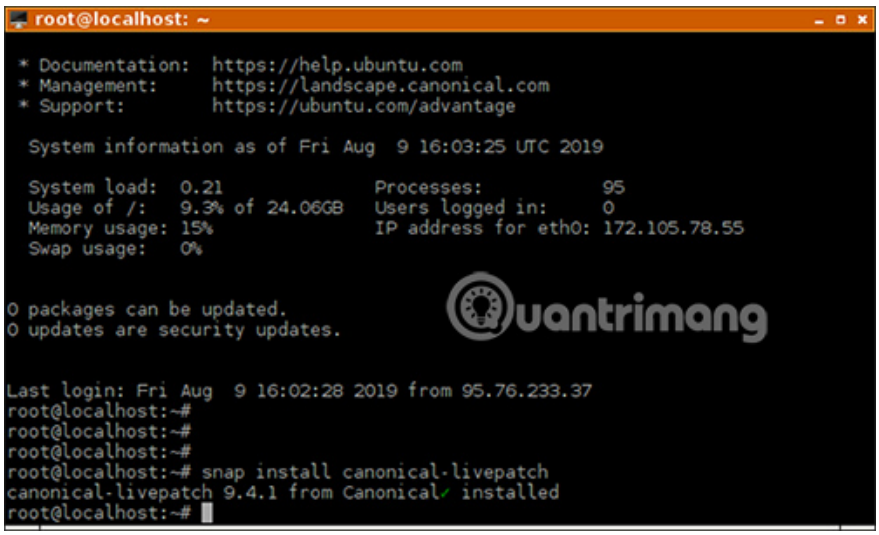
But, as with most things, implementation and technical details are not so simple.

How to set up Livepatch on Ubuntu

Go to this page and create an Ubuntu One account (or just log in if you already have an account). Check your email and click the account confirmation link later. Next, visit the Canonical Livepatch Service page. Select the option that indicates you are **'Ubuntu user'** and click the button to create a token. The next page will show you the exact commands you must enter on your server. After the first command, enter:

```
sudo snap install canonical-livepatch
```

Wait a few seconds until the snap package is fully installed. When you're done, you'll get a result similar to what is shown in the following image.



Finally, with the following command from the Canonical page `sudo canonical-livepatch enable #PASTE_YOUR_TOKEN_HERE` , the service will work and automatically apply security patches to the kernel,

whenever necessary, with no input required from the side. user.

Install the daemon snap if necessary

In rare cases, the first command in the previous section may not succeed, with the following error message: `-bash: /usr/bin/snap: No such file or directory`. In this case, that means your server provider has an Ubuntu operating system image that does not include the daemon service snap by default. Install it with the command:

```
sudo apt update && sudo apt install snapd
```

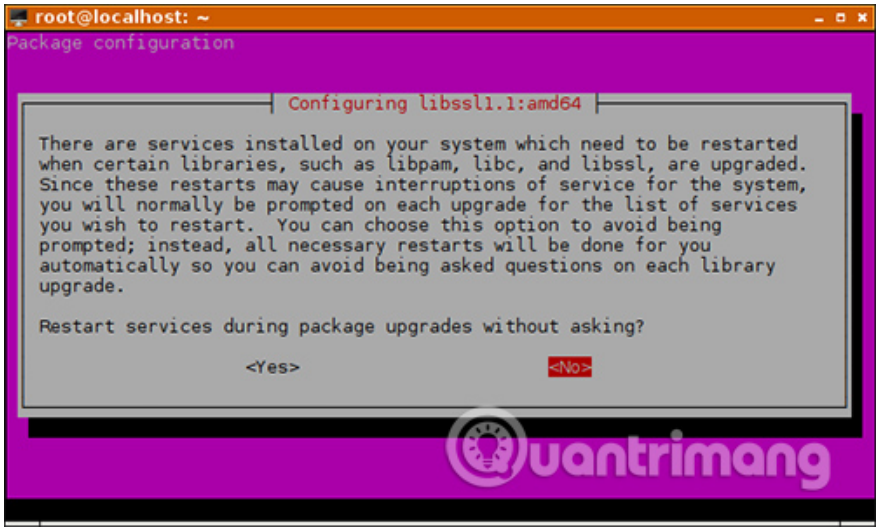
Now run the two commands from the previous section again.

Keep your server up to date

Livepatch will apply all necessary security updates to your kernel. However, you should still upgrade the rest of the system regularly with a command like:

```
sudo apt update && sudo apt upgrade
```

You should do this weekly, or even more often, if you can. Important system packages may prompt you that they need to be rebooted to apply the latest security fixes.



These restart operations do not break any service in the process. For example, in this case, the SSH daemon was restarted without interrupting the active SSH session.

In other situations, you can restart the service yourself to make sure that the new, patched code is reloaded and the security fixes are applied. For example, if you notice the nginx package has been upgraded, you can run: `systemctl restart nginx.service` to reload the nginx daemon into memory.

On the other hand, although a package is upgraded, it can still run with old, problem-prone code, which puts your server at risk. Some package upgrades do this for you, but some other upgrades do not. That's why paying

attention to what **'apt upgrade'** does and restarting some services, if necessary, is a good habit. You can also look at the log to see if this has been done automatically.

As you can see, Canonical makes it easy to avoid rebooting on the server. Regarding the kernel part, there is a part without maintenance. The only thing you can do is run the command: `canonical-livepatch status` to check everything.

Hope you are succesful.