

How to Upgrade from Ubuntu 20.04 to Ubuntu 22.04

Learn how to safely upgrade your system from Ubuntu 20.04 LTS to Ubuntu 22.04 LTS with our step-by-step guide covering preparation, installation, and troubleshooting.

Contents

01	Introduction	3
02	Prerequisites	3
03	Check the Compatibility of Installed Applications	4
04	Prepare the Server for Upgrade	4
05	Upgrade From Ubuntu 20.04 to 22.04	7
06	Test the Upgraded Ubuntu 22.04 Server	11
07	Resolve Post-upgrade Issues	13
08	Conclusion	14

Introduction

Ubuntu is a modern open-source Linux distribution that's available in different versions. Ubuntu 22.04 is an LTS (Long Term Support) version with the latest packages, newer kernel features, improved application stability, and security improvements compared to Ubuntu 20.04. You can upgrade Ubuntu 20.04 to 22.04 to take advantage of these enhancements.

This article covers how to upgrade from Ubuntu 20.04 to 22.04.

Prerequisites

Before you begin:

- Have an [Ubuntu 20.04 server](#).
- Access the instance using SSH as a non-root user with sudo privileges.
- Ensure you've at least 5GB of free space.

CONSOLE

```
$ df -h
```

Output:

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	96M	2.4M	93M	3%	/run
/dev/vda2	23G	6.5G	16G	30%	/
tmpfs	476M	0	476M	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
/dev/vda1	511M	6.1M	505M	2%	/boot/efi
tmpfs	96M	4.0K	96M	1%	/run/user/0

Check the Compatibility of Installed Applications

Installed applications must be compatible with Ubuntu 22.04 before upgrade. Upgrading incompatible applications may lead to runtime errors or package conflicts on your server. Follow the steps below to view all the installed applications and check the compatibility of installed applications before upgrading Ubuntu 20.04 to 22.04.

1. Export the list of all installed application packages to a single file.

CONSOLE

```
$ sudo dpkg --get-selections > packages-list.txt
```

2. View the file contents and visit the official documentation of each package to verify that it's compatible with Ubuntu 22.04.

CONSOLE

```
$ cat packages-list.txt | more
```

3. Visit the [official Ubuntu 22.04 release notes](#) to verify all incompatible dependencies and known performance issues.

Prepare the Server for Upgrade

Follow the steps below to update the APT package manager repository sources and upgrade all existing packages to the latest versions on your server.

1. View the server's distribution version and verify that it's Ubuntu 20.04.

CONSOLE

```
$ cat /etc/os-release
```

2. Update your server's package information index.

```
CONSOLE
```

```
$ sudo apt update
```

3. Upgrade all existing packages to the latest versions.

```
CONSOLE
```

```
$ sudo apt upgrade -y
```

If you receive the following Kernel warning:

```
Newer kernel available
The currently running kernel version is 5.15.0-23-generic which is not the
expected kernel version 5.15.0-24-generic.
Restarting the system to load the new kernel will not be handled automatically,
so you should consider rebooting.
```

Restart your server to load the new Kernel version.

```
CONSOLE
```

```
$ sudo reboot
```

4. Remove all unwanted packages.

```
CONSOLE
```

```
$ sudo apt autoremove -y
```

5. Remove any cached packages.

```
CONSOLE
```

```
$ sudo apt autoclean
```

Enable the Temporary SSH Port

The Ubuntu 22.04 upgrade process uses the temporary SSH port `1022` as a fallback port in case your SSH session terminates before the upgrade process is complete. Follow the steps below to enable connections to the temporary SSH port `1022` on your server.

1. Allow port `1022` through the default `UFW` firewall.

CONSOLE

```
$ sudo ufw allow 1022/tcp
```

2. Reload UFW to apply the firewall changes.

CONSOLE

```
$ sudo ufw reload
```

3. View the UFW status and verify that the new rule is available.

CONSOLE

```
$ sudo ufw status
```

Output:

```
Status: active
```

To	Action	From
--	-----	----
22/tcp	ALLOW	Anywhere
1022/tcp	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)
1022/tcp (v6)	ALLOW	Anywhere (v6)

Upgrade From Ubuntu 20.04 to 22.04

Follow the steps below to upgrade your server to Ubuntu 22.04 from Ubuntu 20.04.

1. Run the following command to start the upgrade process.

CONSOLE

```
$ sudo do-release-upgrade
```

2. Enter Y and press Enter when prompted to upgrade to Ubuntu 22.04.

```
Checking for a new Ubuntu release

= Welcome to Ubuntu 22.04 LTS 'Noble Numbat' =

The Ubuntu team is proud to announce Ubuntu 22.04 LTS 'Noble Numbat'.

To see what's new in this release, visit:
  https://wiki.ubuntu.com/NobleNumbat/ReleaseNotes

.....

To sign up for future Ubuntu announcements, please subscribe to Ubuntu's
very low volume announcement list at:

  http://lists.ubuntu.com/mailman/listinfo/ubuntu-announce

Continue [yN]
```

3. Enter Y and press Enter to enable the temporary SSH port `1022`.

```
Reading cache

Checking package manager

Continue running under SSH?
```

```
This session appears to be running under ssh. It is not recommended
to perform a upgrade over ssh currently because in case of failure it
is harder to recover.
```

```
If you continue, an additional ssh daemon will be started at port
'1022'.
```

```
Do you want to continue?
```

```
Continue [yN]
```

4. Press Enter to accept the additional firewall rules and continue with the upgrade.

```
Starting additional sshd
```

```
To make recovery in case of failure easier, an additional sshd will
be started on port '1022'. If anything goes wrong with the running
ssh you can still connect to the additional one.
```

```
If you run a firewall, you may need to temporarily open this port. As
this is potentially dangerous it's not done automatically. You can
open the port with e.g.:
```

```
'iptables -I INPUT -p tcp --dport 1022 -j ACCEPT'
```

```
To continue please press [ENTER]
```

5. Enter Y and press Enter to start the upgrade.

```
Do you want to start the upgrade?
```

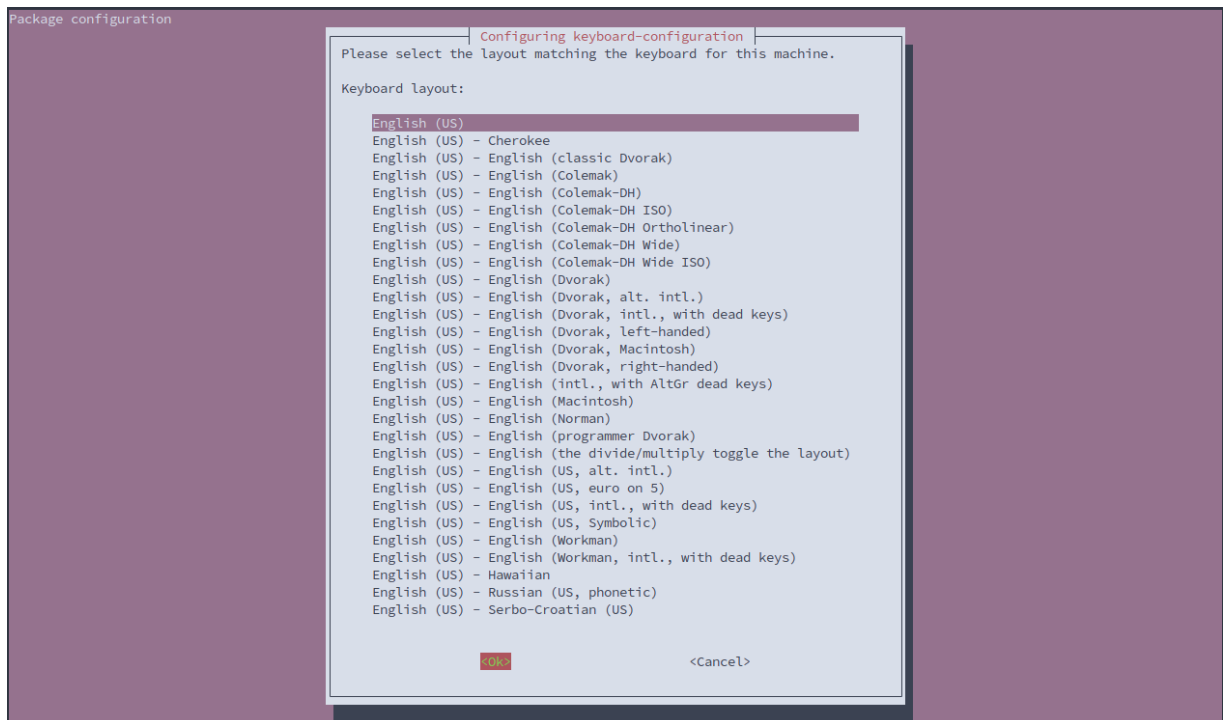
```
52 packages are going to be removed. 162 new packages are going to be
installed. 660 packages are going to be upgraded.
```

```
You have to download a total of 1,197 M. This download will take
about 2 minutes with your connection.
```

```
Installing the upgrade can take several hours. Once the download has
finished, the process cannot be canceled.
```

```
Continue [yN] Details [d]
```

6. Select your desired keyboard layout and press Enter when prompted.



7. Press Enter to choose **N** and keep the default `/etc/systemd/journald.conf` configuration version when prompted.

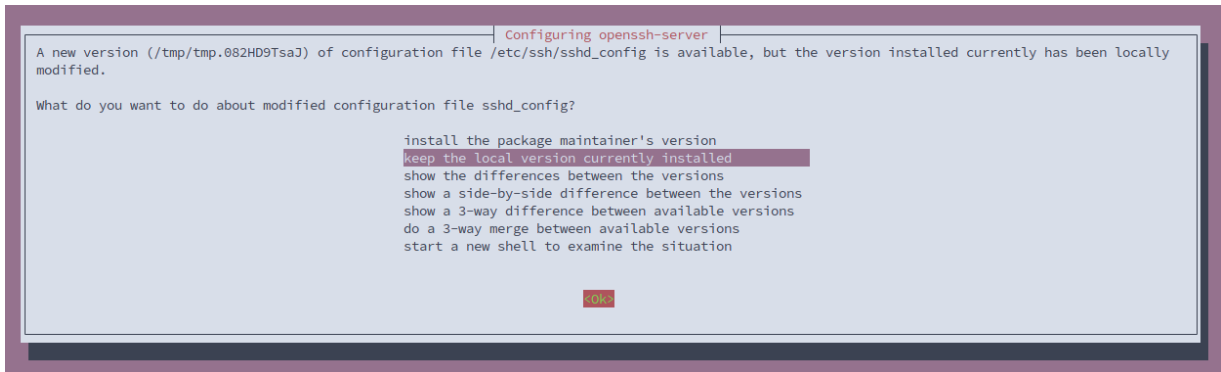
```
Configuration file '/etc/systemd/journald.conf'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it? Your options are:
Y or I : install the package maintainer's version
N or O : keep your currently-installed version
D      : show the differences between the versions
Z      : start a shell to examine the situation
The default action is to keep your current version.
*** journald.conf (Y/I/N/O/D/Z) [default=N] ?
```

8. Press Enter to choose **N** and keep the default `/etc/systemd/timesyncd.conf` configuration version when prompted.

```
Configuration file '/etc/systemd/timesyncd.conf'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it? Your options are:
Y or I : install the package maintainer's version
N or O : keep your currently-installed version
D      : show the differences between the versions
```

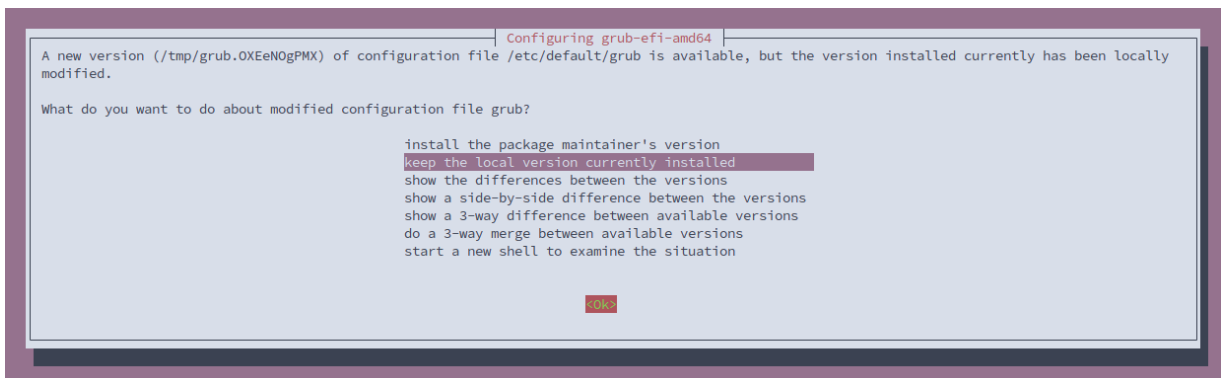
```
Z      : start a shell to examine the situation
The default action is to keep your current version.
*** timesyncd.conf (Y/I/N/O/D/Z) [default=N] ?
```

9. Select the `keep the local version currently installed` option and press Enter when prompted to modify the `/etc/ssh/sshd_config` file.



```
Configuring openssh-server
A new version (/tmp/tmp.082HD9TsaJ) of configuration file /etc/ssh/sshd_config is available, but the version installed currently has been locally modified.
What do you want to do about modified configuration file sshd_config?
install the package maintainer's version
keep the local version currently installed
show the differences between the versions
show a side-by-side difference between the versions
show a 3-way difference between available versions
do a 3-way merge between available versions
start a new shell to examine the situation
OK
```

10. Select the `keep the local version currently installed` option and press Enter when prompted to modify the `/etc/default/grub` file.



```
Configuring grub-efi-amd64
A new version (/tmp/grub.OXEeNOgPMX) of configuration file /etc/default/grub is available, but the version installed currently has been locally modified.
What do you want to do about modified configuration file grub?
install the package maintainer's version
keep the local version currently installed
show the differences between the versions
show a side-by-side difference between the versions
show a 3-way difference between available versions
do a 3-way merge between available versions
start a new shell to examine the situation
OK
```

11. Press Enter to select `N` and keep the existing `/etc/cloud/cloud.cfg` configuration file version.

```
Configuration file '/etc/cloud/cloud.cfg'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it? Your options are:
Y or I : install the package maintainer's version
N or O : keep your currently-installed version
D      : show the differences between the versions
Z      : start a shell to examine the situation
The default action is to keep your current version.
*** cloud.cfg (Y/I/N/O/D/Z) [default=N] ?
```

12. Enter D when prompted to view obsolete packages, enter Y, and press Enter to remove the packages.

```
Processing snap replacements

refreshing snap lxd

Searching for obsolete software
Reading state information... Done

Remove obsolete packages?

98 packages are going to be removed.

Continue [yN] Details [d]
```

13. Wait for at least **30** minutes for the upgrade process to complete depending on your server packages, enter Y and press Enter when prompted to restart the server.

```
System upgrade is complete.

Restart required

To finish the upgrade, a restart is required.
If you select 'y' the system will be restarted.

Continue [yN]
```

To further optimize your server setup after the upgrade, you may want to [install Docker on Ubuntu 22.04](#). Docker enables you to run applications in containers, providing improved efficiency and resource management.

Test the Upgraded Ubuntu 22.04 Server

Follow the steps below to test your server packages and verify that the system upgrades to Ubuntu 22.04.

1. View the active Ubuntu version on your server.

CONSOLE

```
$ cat /etc/os-release
```

Output:

```
PRETTY_NAME="Ubuntu 22.04.5 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.5 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
```

2. View the upgraded kernel version.

CONSOLE

```
$ uname -mrs
```

Output:

```
Linux 5.15.0-136-generic x86_64
```

3. Delete the temporary SSH port `1022` you created earlier.

CONSOLE

```
$ sudo ufw delete allow 1022/tcp
```

Resolve Post-upgrade Issues

Follow the steps below to verify the existing package repository sources and upgrade all packages on your server.

1. View all existing APT repository sources.

CONSOLE

```
$ ls /etc/apt/
```

Modify any third-party repository files to include the new Ubuntu 22.04 sources.

2. Update the server's package information index.

CONSOLE

```
$ sudo apt update
```

3. Upgrade the server packages.

CONSOLE

```
$ sudo apt upgrade
```

4. Remove all unused and obsolete packages.

CONSOLE

```
$ sudo apt autoremove -y && sudo apt autoclean
```

Conclusion

You have upgraded your server from Ubuntu 20.04 to 22.04. With this upgrade from Ubuntu 20.04 to 22.04, you can access new security improvements, improved application performance, and upgraded package dependencies on your server. For more information about the upgrade process, please visit the [Ubuntu server upgrade documentation](#).



VULTR

