

Personal Photo Management Server with Immich

This will be a small "guide" to the installation of Tailscale and Immich.
After some time of paying for iCloud storage and recently running out of storage for Google Photos I decided to finally get rid of pay-to-play services by using Immich.

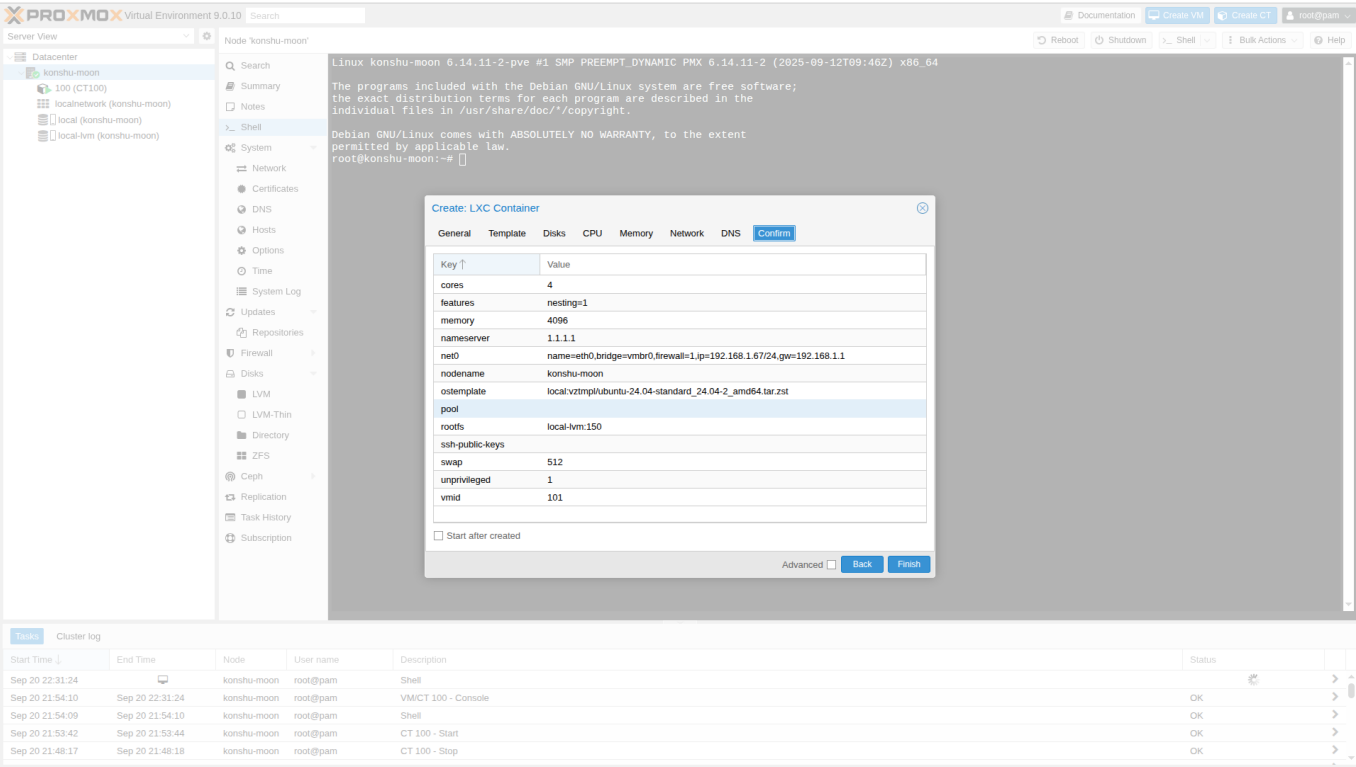
Immich is essentially a Google Photos clone but is free, open-source software that can be used on a home server to have your very own cloud with unlimited amounts of storage as long as you provide it.

Tailscale is an open-source WireGuard-based VPN that connects users, services, and devices directly. Tailscale can be used anywhere in the world as long as you have internet access — and the best part is that it's FREE (with paid tiers available, but the free tier is more than enough for a regular user).

Specs

- Intel Core i5-7500 @ 3.40GHz
- 8GB DDR3 RAM
- 1TB HDD (Seagate)
- Proxmox (OS)

LXC Container

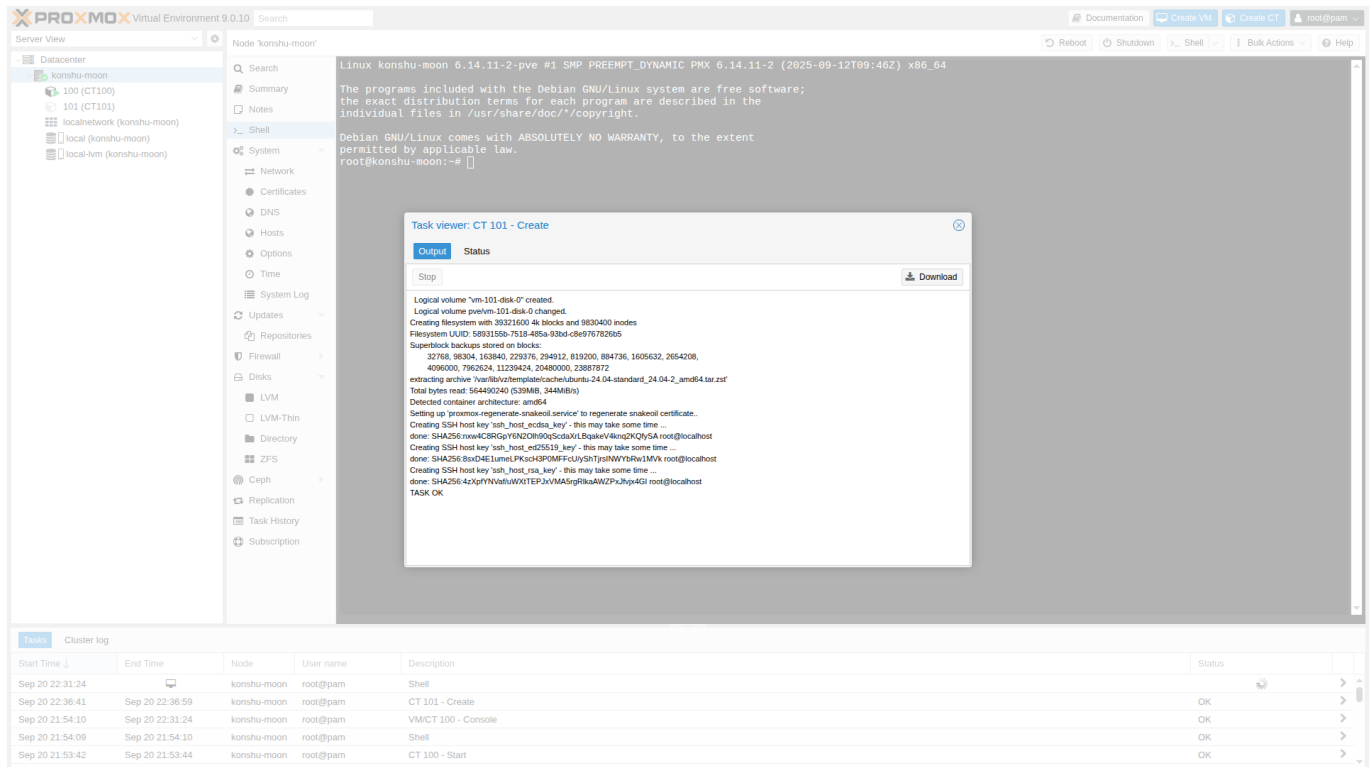


Inside of Proxmox I made an LXC container running Ubuntu so that it can house our Immich and Docker programs.

In my screenshot above you'll see exactly what settings I am using for this container to run Immich along with Tailscale.

Note: What I'm missing in the screenshot is the DNS and IP configuration.

- **DNS:** hostname set to host setting, DNS IP set to **1.1.1.1** (Cloudflare)
- **IP configuration:** can be automatically set, but I opted for a static IP (easier for the Tailscale installation).



The screenshot shows the Proxmox VE 9.0.10 interface. In the left sidebar, the 'Datacenter' view is active, showing a tree of nodes: 'konshu-moon', '100 (CT100)', and '101 (CT101)'. The '101 (CT101)' container is selected. The main panel displays the 'Task viewer: CT 101 - Create' window, which shows the output of the container creation process. The output includes the following text:

```
Linux konshu-moon 6.14.11-2-pve #1 SMP PREEMPT_DYNAMIC PMX 6.14.11-2 (2025-09-12T09:46Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@konshu-moon:~#
```

The task viewer window also shows a 'Status' tab with a 'Download' button. Below the task viewer, a 'Tasks' table is visible, showing the history of tasks performed on the cluster.

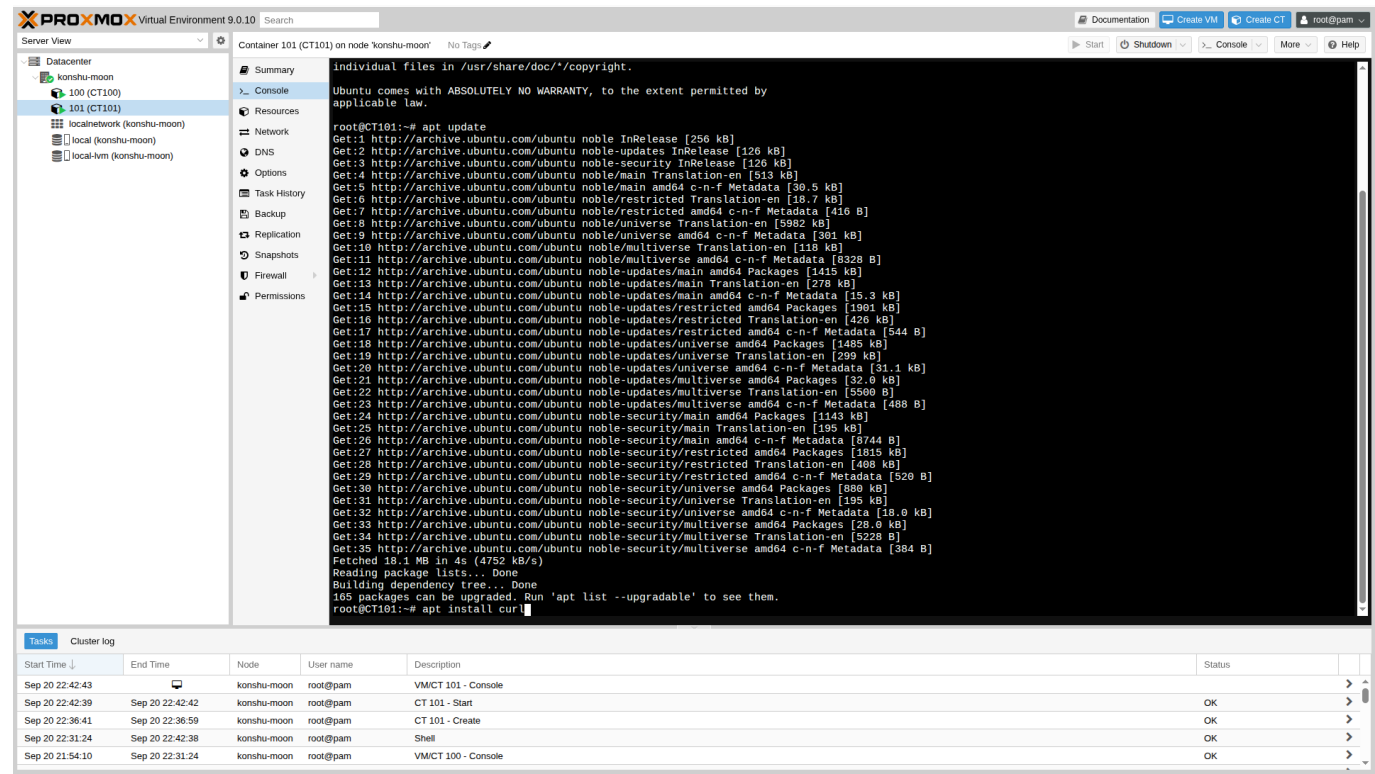
Start Time	End Time	Node	User name	Description	Status
Sep 20 22:31:24		konshu-moon	root@pam	Shell	
Sep 20 22:36:41	Sep 20 22:36:59	konshu-moon	root@pam	CT 101 - Create	OK
Sep 20 21:54:10	Sep 20 22:31:24	konshu-moon	root@pam	VM/CT 100 - Console	OK
Sep 20 21:54:09	Sep 20 21:54:10	konshu-moon	root@pam	Shell	OK
Sep 20 21:53:42	Sep 20 21:53:44	konshu-moon	root@pam	CT 100 - Start	OK

Once the fields were filled in with our appropriate settings, we click on **Create**.

When finished, you'll see what's shown above letting us know that our container is now ready to use.

Before starting installations, update the container:

```
apt update
apt install curl
```

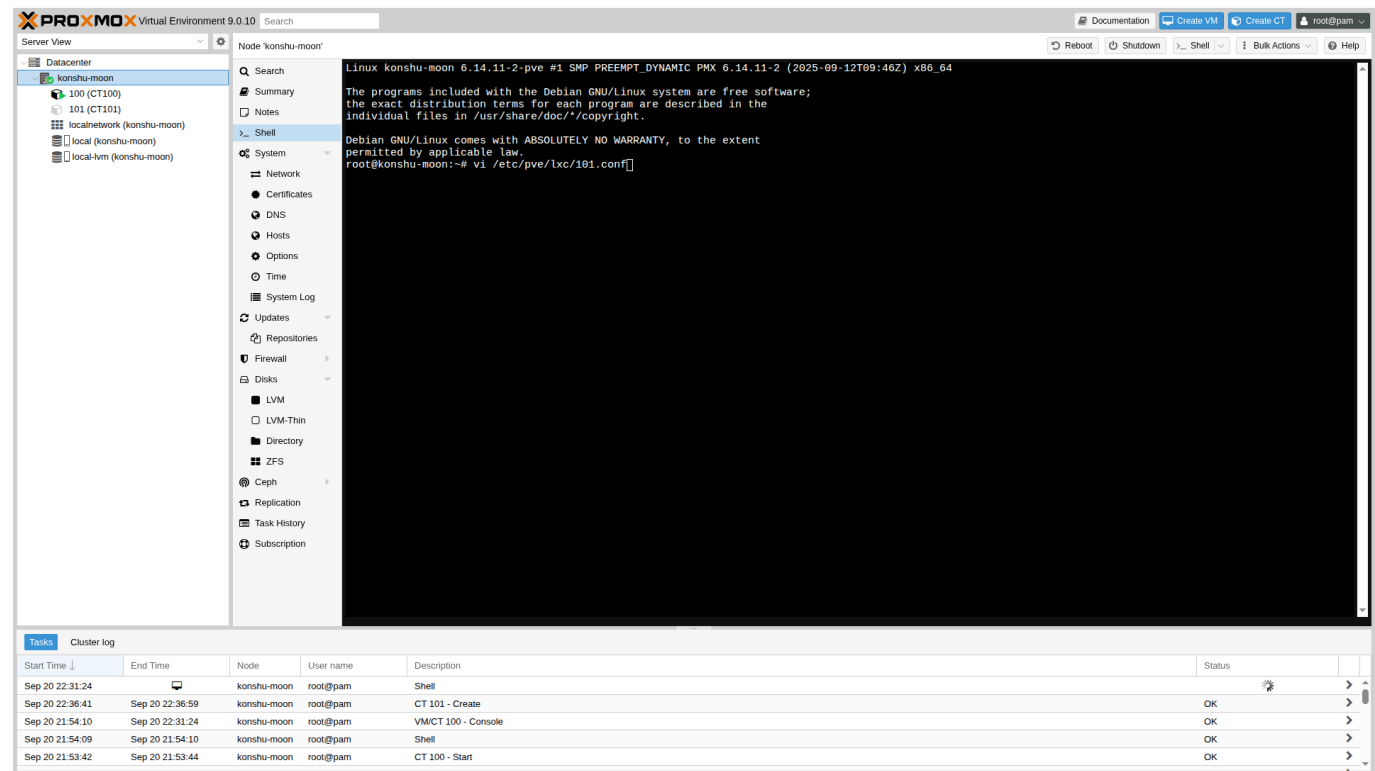


Tailscale Install

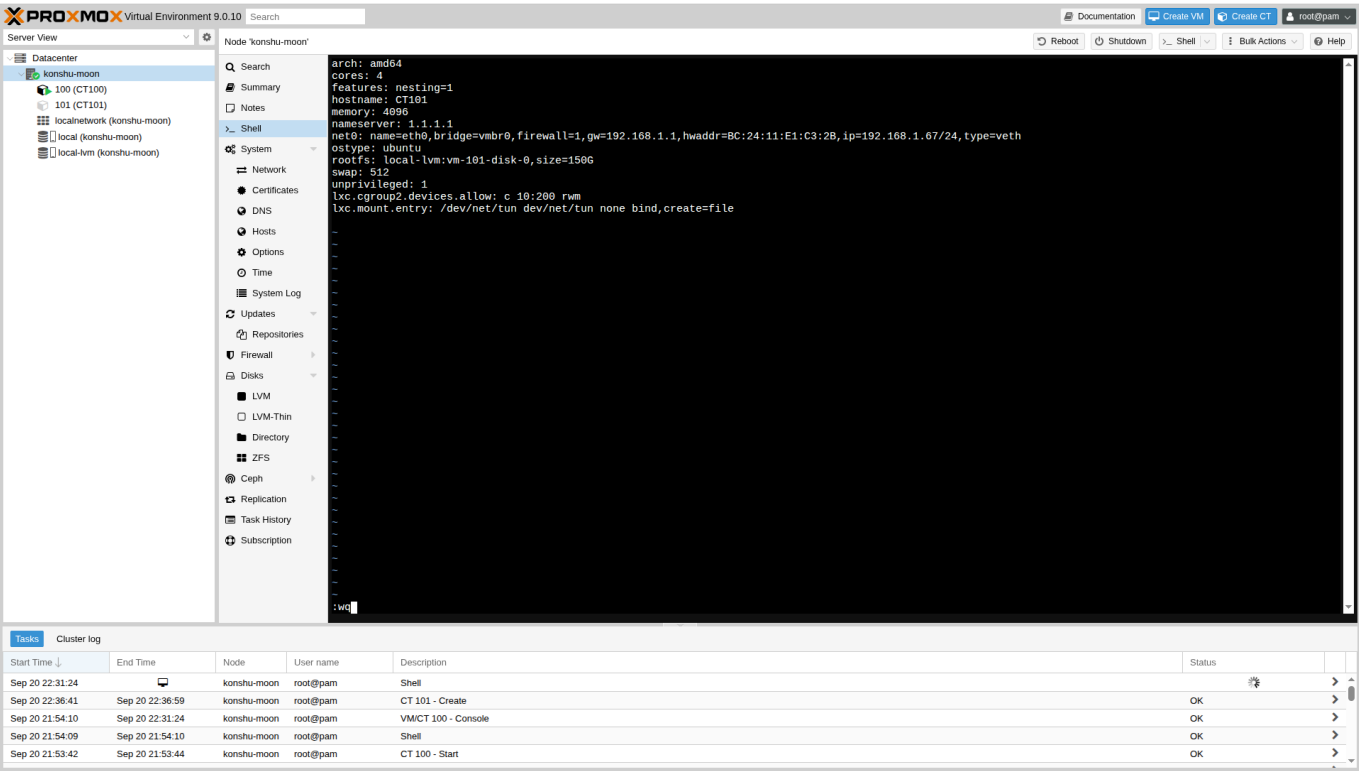
In our **host shell**, we need to edit a config file to let Tailscale use a tun device. Add these lines:

```
lxc.cgroup2.devices.allow: c 10:200 rwm
lxc.mount.entry: /dev/net/tun dev/net/tun none bind,create=file
```

To edit the file, run the command shown in the screenshot below:



Inside of the file, the configuration will look like this:



Now that our config file is ready, we can install Tailscale inside the LXC container:

```
curl -fsSL https://tailscale.com/install.sh | sh
```

After the installation completes, run:

```
tailscale up
```

Tailscale will then present a **login link** you can open in your browser to set up Tailscale with your email.

Screenshots below:

PROXMOX Virtual Environment 9.0.10

Search

Documentation Create VM Create CT root@pam

Server View

Container 101 (CT101) on node 'konshu-moon' No Tags

Start Shutdown Console More Help

Datacenter

konshu-moon

100 (CT100)

101 (CT101)

localnetwork (konshu-moon)

local (konshu-moon)

local-lvm (konshu-moon)

Summary

Console

Resources

Network

DNS

Options

Task History

Backup

Replication

Snapshots

Firewall

Permissions

Tailscale packages for ubuntu noble

deb [signed-by=/usr/share/keyrings/tailscale-archive-keyring.gpg] https://pkgs.tailscale.com/stable/ubuntu noble main

+ chmod 0644 /etc/apt/sources.list.d/tailscale.list

+ apt-get update

Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease

Hit:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease

Hit:3 http://archive.ubuntu.com/ubuntu noble-security InRelease

Get:4 https://pkgs.tailscale.com/stable/ubuntu noble InRelease

Get:5 https://pkgs.tailscale.com/stable/ubuntu noble/main all Packages [354 B]

Get:6 https://pkgs.tailscale.com/stable/ubuntu noble/main amd64 Packages [13.1 kB]

Fetched 20.0 kB in 0s (63.4 kB/s)

Reading package lists... Done

+ apt-get install -y tailscale tailscale-archive-keyring

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

The following NEW packages will be installed:

tailscale tailscale-archive-keyring

0 upgraded, 2 newly installed, 0 to remove and 165 not upgraded.

Need to get 34.1 MB of archives:

After this operation, 66.3 MB of additional disk space will be used.

Get:2 https://pkgs.tailscale.com/stable/ubuntu noble/main all tailscale-archive-keyring all 1.35.181 [3082 B]

Get:1 https://pkgs.tailscale.com/stable/ubuntu noble/main amd64 tailscale amd64 1.88.1 [34.1 MB]

Fetched 34.1 MB in 1s (36.7 MB/s)

Selecting previously unselected package tailscale.

(Reading database ... 17372 files and directories currently installed.)

Preparing to unpack .../tailscale-1.88.1-amd64.deb ...

Unpacking tailscale (1.88.1) ...

Selecting previously unselected package tailscale-archive-keyring.

Preparing to unpack .../tailscale-archive-keyring-1.35.181-all.deb ...

Unpacking tailscale-archive-keyring (1.35.181) ...

Setting up tailscale-archive-keyring (1.35.181) ...

Setting up tailscale (1.88.1) ...

Created symlink /etc/systemd/system/multi-user.target.wants/tailscaled.service -> /usr/lib/systemd/system/tailscaled.service.

+ [false = true]

+ set -x

Installation complete! Log in to start using Tailscale by running:

tailscale up

root@CT101:~# tailscale up

To authenticate, visit:

https://login.tailscale.com/a/fe7014d01c184

Tasks Cluster log

Start Time	End Time	Node	User name	Description	Status
Sep 20 22:42:43		konshu-moon	root@pam	VMCT 101 - Console	
Sep 20 22:42:39	Sep 20 22:42:42	konshu-moon	root@pam	CT 101 - Start	
Sep 20 22:36:41	Sep 20 22:36:59	konshu-moon	root@pam	CT 101 - Create	OK
Sep 20 22:31:24	Sep 20 22:42:38	konshu-moon	root@pam	Shell	OK
Sep 20 21:54:10	Sep 20 22:31:24	konshu-moon	root@pam	VMCT 100 - Console	OK

login.tailscale.com/a/fe7014d01c184?next_url=%252Fa%252Ffe7014d01c184&refreshed=true

tailscale

Login successful

Your device CT101 is logged in to the

If this is not what you meant to do, you can remove the device from your tailnet. If you need help, contact support.

You will be redirected to your console shortly.

Or, you can visit the console immediately.

Docker Installation

/

The screenshot shows the Docker documentation page for installing Docker Engine on Ubuntu using the apt repository. The page is titled "Install using the apt repository" and includes a table of contents on the right. The main content area shows the steps to install Docker, including adding the Docker repository and installing the Docker packages. The steps are as follows:

- Set up Docker's apt repository.

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download
  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```
- Install the Docker packages.

Latest Specific version

To install the latest version, run:

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compos
```

Note

The Docker service starts automatically after installation. To verify that Docker is running, use:

```
$ sudo systemctl status docker
```

To properly run Immich, we must first install Docker Compose.
We'll use the [Docker official apt repository instructions](#).

Installing Immich

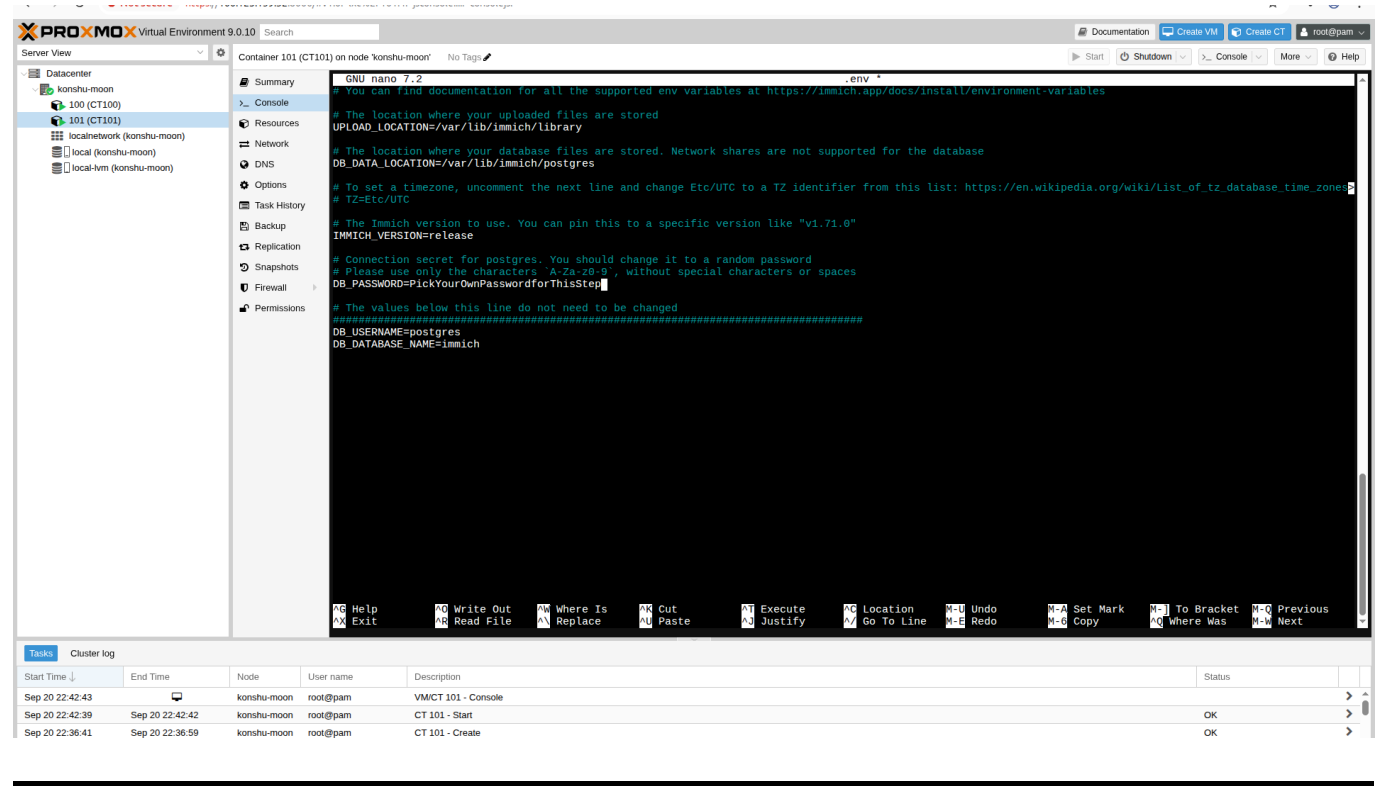
Now for the golden egg: **Immich**.
The installation steps are available on the [Immich website](#).

While following the steps, I ran into some confusion when editing the configuration files (not mentioned in the docs).

To access the files, run:

```
nano .env
```

Inside the editor, fill out the configuration values as shown in the screenshot below (remember to create your own password).



The Finished Product

After completing the installation of Docker, Tailscale (so we can access it anywhere), and Immich, the last step is the post-installation setup.

Follow [Immich post-install instructions](#) to download the mobile app.

This is what my finished product looks like:

