

kroese	build: Pin Alpine version (#133) ✓	6f83332 · 17 hours ago	🕒 340 Commits
📁 .github	build: Pin Alpine version (#133)		17 hours ago
📁 src	feat: Generate unique machine serials ...		17 hours ago
📄 .dockerignore	fix: Delete ISO when extraction fails (#...		7 months ago
📄 .gitignore	build: Initial Dockerfile (#3)		10 months ago
📄 Dockerfile	build: Pin Alpine version (#133)		17 hours ago
📄 compose.yml	docs: Use numeric version values (#102)		last month
📄 config.plist	feat: Removed bootmenu (#30)		5 months ago
📄 kubernetes.yml	docs: Removed Kubevirt dependancy ...		5 months ago
📄 license.md	Create license.md		10 months ago
📄 readme.md	feat: Generate unique machine serials ...		17 hours ago

About

OSX (macOS) inside a Docker container.

- #macos #docker #mac #osx #docker-container
- #virtualization #macosx #osx-virtual-machine
- #macos-vm #macos-virtual-machine

- 📖 Readme
- 📄 MIT license
- 📈 Activity
- 📋 Custom properties
- ⭐ 8.1k stars
- 👁 26 watching
- 🔗 277 forks

Report repository

Packages 1

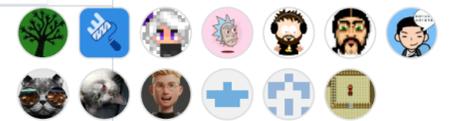
📦 macos

📖 README 📄 License



🔄 Build passing
📄 version v1.16
📄 size 110 MB
🔄 pulls 4.3k
👍 pulls 50k

👤 Contributors 13



Languages



OSX (macOS) inside a Docker container.

Features ✨

- KVM acceleration
- Web-based viewer
- Automatic download

Usage 🐳

Via Docker Compose:

```

services:
  macos:
    image: dockurr/macos
    container_name: macos
    environment:
      VERSION: "13"
    devices:
      - /dev/kvm
    cap_add:
      - NET_ADMIN
    ports:
      - 8006:8006
      - 5900:5900/tcp
      - 5900:5900/udp
    stop_grace_period: 2m

```

Via Docker CLI:

```
docker run -it --rm -p 8006:8006 --device=/dev/kvm --cap-add NET_ADMIN --stop-timeout 120 dockurr/macos
```

Via Kubernetes:

```
kubectl apply -f https://raw.githubusercontent.com/dockur/macos/refs/heads/master/kubernetes.yml
```

Compatibility

Product	Platform	
Docker Engine	Linux	✓
Docker Desktop	Linux	✗
Docker Desktop	macOS	✗
Docker Desktop	Windows 11	✓
Docker Desktop	Windows 10	✗

FAQ

How do I use it?

Very simple! These are the steps:

- Start the container and connect to [port 8006](#) using your web browser.
- Choose `Disk Utility` and then select the largest `Apple Inc. VirtIO Block Media` disk.
- Click the `Erase` button to format the disk to APFS, and give it any recognizable name you like.
- Close the current window and proceed the installation by clicking `Reinstall macOS`.
- When prompted where you want to install it, select the disk you just created previously.
- After all files are copied, select your region, language, and account settings.

Enjoy your brand new machine, and don't forget to star this repo!

How do I select the macOS version?

By default, macOS 13 (Ventura) will be installed, as it offers the best performance.

But you can add the `VERSION` environment variable to your compose file, in order to specify an alternative macOS version to be downloaded:

```
environment:  
  VERSION: "13"
```



Select from the values below:

Value	Version	Name
15	macOS 15	Sequoia
14	macOS 14	Sonoma
13	macOS 13	Ventura
12	macOS 12	Monterey
11	macOS 11	Big Sur

How do I change the storage location?

To change the storage location, include the following bind mount in your compose file:

```
volumes:  
  - /var/osx:/storage
```



Replace the example path `/var/osx` with the desired storage folder.

How do I change the size of the disk?

To expand the default size of 64 GB, add the `DISK_SIZE` setting to your compose file and set it to your preferred capacity:

```
environment:  
  DISK_SIZE: "256G"
```



Tip

This can also be used to resize the existing disk to a larger capacity without any data loss.

How do I change the amount of CPU or RAM?

By default, the container will be allowed to use a maximum of 2 CPU cores and 4 GB of RAM.

If you want to adjust this, you can specify the desired amount using the following environment variables:

```
environment:
  RAM_SIZE: "8G"
  CPU_CORES: "4"
```



How do I pass-through a USB device?

To pass-through a USB device, first lookup its vendor and product id via the `lsusb` command, then add them to your compose file like this:

```
environment:
  ARGUMENTS: "-device usb-host,vendorid=0x1234,productid=0x1234"
devices:
  - /dev/bus/usb
```



How do I verify if my system supports KVM?

Only Linux and Windows 11 support KVM virtualization, macOS and Windows 10 do not unfortunately.

You can run the following commands in Linux to check your system:

```
sudo apt install cpu-checker
sudo kvm-ok
```



If you receive an error from `kvm-ok` indicating that KVM cannot be used, please check whether:

- the virtualization extensions (`Intel VT-x` or `AMD SVM`) are enabled in your BIOS.
- you enabled "nested virtualization" if you are running the container inside a virtual machine.
- you are not using a cloud provider, as most of them do not allow nested virtualization for their VPS's.

If you do not receive any error from `kvm-ok` but the container still complains about KVM, please check whether:

- you are not using "Docker Desktop for Linux" as it does not support KVM, instead make use of Docker Engine directly.
- it could help to add `privileged: true` to your compose file (or `sudo` to your `docker run` command), to rule out any permission issue.

How do I run Windows in a container?

You can use [dockur/windows](#) for that. It shares many of the same features, and even has completely automatic installation.

Is this project legal?

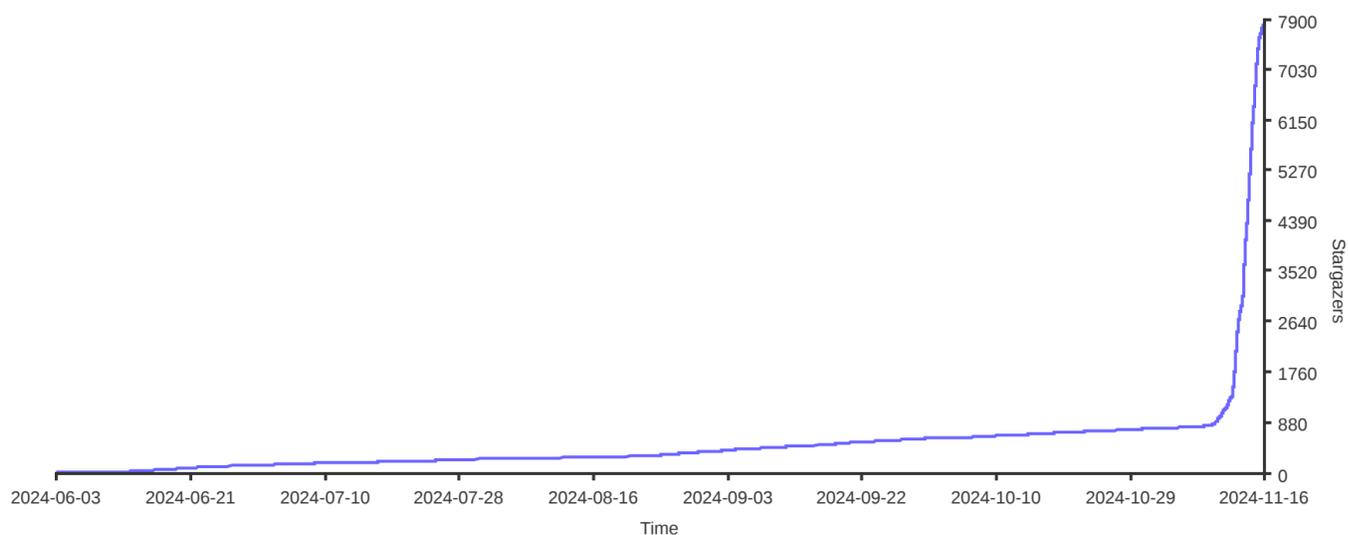
Yes, this project contains only open-source code and does not distribute any copyrighted material. Neither does it try to circumvent any copyright protection measures. So under all applicable laws, this project will be considered legal.

However, by installing Apple's macOS, you must accept their end-user license agreement, which does not permit installation on non-official hardware. So only run this container on hardware sold by Apple, as any other use will be a violation of their terms and conditions.

Acknowledgements 🙏

Special thanks to [seitenca](#), this project would not exist without her invaluable work.

Stars 🌟



Disclaimer ⚖️

Only run this container on Apple hardware, any other use is not permitted by their EULA. The product names, logos, brands, and