

master

1 Branch

73 Tags

Go to file

Go to file

<> Code

...

About

Windows inside a Docker container.

#windows #docker #docker-container #virtualization #windows-vm #windows-virtual-machines #windows-virtual-desktop #windows-virtual-machine

- Readme
- MIT license
- Activity
- Custom properties
- 28.8k stars
- 153 watching
- 2k forks

Report repository

Packages 1

windows

	kroese feat: Improved download method (#903) ✓	2c4094b · 2 days ago	330 Commits
📁	.github	build: Docker lint exception (#813)	last month
📁	assets	feat: Select LTSC edition based on ke...	last month
📁	src	feat: Improved download method (#903)	2 days ago
📄	.dockerignore	fix: Delete ISO when extraction fails (#...	7 months ago
📄	.gitignore	build: Initial Dockerfile (#3)	10 months ago
📄	Dockerfile	feat: Improved download method (#903)	2 days ago
📄	compose.yml	docs: Version alias (#859)	last week
📄	kubernetes.yml	docs: Version alias (#859)	last week
📄	license.md	Create license.md	10 months ago
📄	readme.md	feat: Improved download method (#903)	2 days ago

README MIT license

Windows



Build passing version v4.04 size 133 MB pulls 27.2k pulls 577k

Windows inside a Docker container.

Features ✨

- ISO downloader
- KVM acceleration
- Web-based viewer

Video 📺



Usage 🐳

Via Docker Compose:

```
services:
  windows:
    image: dockurr/windows
    container_name: windows
    environment:
      VERSION: "11"
```

Followers 14

Languages



```
devices:
  - /dev/kvm
cap_add:
  - NET_ADMIN
ports:
  - 8006:8006
  - 3389:3389/tcp
  - 3389:3389/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/windows
```

Via Kubernetes:

```
kubectl apply -f https://raw.githubusercontent.com/dockur/windows/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.
- Sit back and relax while the magic happens, the whole installation will be performed fully automatic.
- Once you see the desktop, your Windows installation is ready for use.

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

How do I select the Windows version?

By default, Windows 11 will be installed. But you can add the `VERSION` environment variable to your compose file, in order to specify an alternative Windows version to be downloaded:

```
environment:
  VERSION: "11"
```

Select from the values below:

Value	Version	Size
11	Windows 11 Pro	5.4 GB
11l	Windows 11 LTSC	4.2 GB
11e	Windows 11 Enterprise	5.8 GB
10	Windows 10 Pro	5.7 GB
10l	Windows 10 LTSC	4.6 GB
10e	Windows 10 Enterprise	5.2 GB
8e	Windows 8.1 Enterprise	3.7 GB
7e	Windows 7 Enterprise	3.0 GB
ve	Windows Vista Enterprise	3.0 GB
xp	Windows XP Professional	0.6 GB
2025	Windows Server 2025	5.0 GB
2022	Windows Server 2022	4.7 GB
2019	Windows Server 2019	5.3 GB
2016	Windows Server 2016	6.5 GB

Value	Version	Size
2012	Windows Server 2012	4.3 GB
2008	Windows Server 2008	3.0 GB
2003	Windows Server 2003	0.6 GB

Tip

To install ARM64 versions of Windows use [dockur/windows-arm](#).

How do I change the storage location?

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

```
volumes:  
  - /var/win:/storage
```

Replace the example path `/var/win` 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 share files with the host?

Open 'File Explorer' and click on the 'Network' section, you will see a computer called `host.lan`. Double-click it and it will show a folder called `data`, which can be bound to any folder on your host via the compose file:

```
volumes:  
  - /home/user/example:/data
```

The example folder `/home/user/example` will be available as `\\host.lan\Data`.

Tip

You can map this path to a drive letter in Windows, for easier access.

How do I install a custom image?

In order to download an unsupported ISO image that is not selectable from the list above, specify the URL of that ISO in the `VERSION` environment variable, for example:

```
environment:  
  VERSION: "https://example.com/win.iso"
```

Alternatively, you can also skip the download and use a local file instead, by binding it in your compose file in this way:

```
volumes:  
  - /home/user/example.iso:/custom.iso
```

Replace the example path `/home/user/example.iso` with the filename of your desired ISO file, the value of `VERSION` will be ignored in this case.

How do I run a script after installation?

To run your own script after installation, you can create a file called `install.bat` and place it in a folder together with any additional files it needs (software to be installed for example). Then bind that folder in your compose file like this:

```
volumes:  
  - /home/user/example:/oem
```

The example folder `/home/user/example` will be copied to `C:\OEM` during installation and the containing `install.bat` will be executed during the last step.

How do I perform a manual installation?

It's best to stick to the automatic installation, as it adjusts various settings to prevent common issues when running Windows inside a virtual environment.

However, if you insist on performing the installation manually, add the following environment variable to your compose file:

```
environment:
  MANUAL: "Y"
```



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 configure the username and password?

By default, a user called `Docker` is created during the installation, with an empty password.

If you want to use different credentials, you can change them in your compose file:

```
environment:
  USERNAME: "bill"
  PASSWORD: "gates"
```



How do I select the Windows language?

By default, the English version of Windows will be downloaded. But you can add the `LANGUAGE` environment variable to your compose file, in order to specify an alternative language:

```
environment:
  LANGUAGE: "French"
```



You can choose between: Arabic, Bulgarian, Chinese, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hebrew, Hungarian, Italian, Japanese, Korean, Latvian, Lithuanian, Norwegian, Polish, Portuguese, Romanian, Russian, Serbian, Slovak, Slovenian, Spanish, Swedish, Thai, Turkish and Ukrainian.

How do I select the keyboard layout?

If you want to use a keyboard layout or locale that is not the default for your selected language, you can add the `KEYBOARD` and `REGION` variables with a culture code, like this:

```
environment:
  REGION: "en-US"
  KEYBOARD: "en-US"
```



Note

Changing these values will have no effect after the installation has been performed already. Use the control panel inside Windows in that case.

How do I connect using RDP?

The web-viewer is mainly meant to be used during installation, as its picture quality is low, and it has no audio or clipboard for example.

So for a better experience you can connect using any Microsoft Remote Desktop client to the IP of the container, using the username `Docker` and by leaving the password empty.

There is a RDP client for [Android](#) available from the Play Store and one for [iOS](#) in the Apple Store. For Linux you can use [FreeRDP](#) and on Windows just type `mstsc` in the search box.

How do I assign an individual IP address to the container?

By default, the container uses bridge networking, which shares the IP address with the host.

If you want to assign an individual IP address to the container, you can create a macvlan network as follows:

```
docker network create -d macvlan \
  --subnet=192.168.0.0/24 \
  --gateway=192.168.0.1 \
  --ip-range=192.168.0.100/28 \
  -o parent=eth0 vlan
```



Be sure to modify these values to match your local subnet.

Once you have created the network, change your compose file to look as follows:

```
services:
  windows:
    container_name: windows
    ..<snip>..
```



```
networks:
  vlan:
    ipv4_address: 192.168.0.100

networks:
  vlan:
    external: true
```

An added benefit of this approach is that you won't have to perform any port mapping anymore, since all ports will be exposed by default.

Important

This IP address won't be accessible from the Docker host due to the design of macvlan, which doesn't permit communication between the two. If this is a concern, you need to create a [second macvlan](#) as a workaround.

How can Windows acquire an IP address from my router?

After configuring the container for [macvlan](#), it is possible for Windows to become part of your home network by requesting an IP from your router, just like a real PC.

To enable this mode, add the following lines to your compose file:

```
environment:
  DHCP: "Y"
devices:
  - /dev/vhost-net
device_cgroup_rules:
  - 'c *: * rwm'
```

Note

In this mode, the container and Windows will each have their own separate IPs.

How do I add multiple disks?

To create additional disks, modify your compose file like this:

```
environment:
  DISK2_SIZE: "32G"
  DISK3_SIZE: "64G"
volumes:
  - /home/example:/storage2
  - /mnt/data/example:/storage3
```

How do I pass-through a disk?

It is possible to pass-through disk devices directly by adding them to your compose file in this way:

```
devices:
  - /dev/sdb:/disk1
  - /dev/sdc:/disk2
```

Use `/disk1` if you want it to become your main drive (which will be formatted during installation), and use `/disk2` and higher to add them as secondary drives (which will stay untouched).

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
```

Important

If the device is a USB disk drive, please wait until after the installation is completed before connecting it. Otherwise the installation may fail, as the order of the disks can get rearranged.

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.

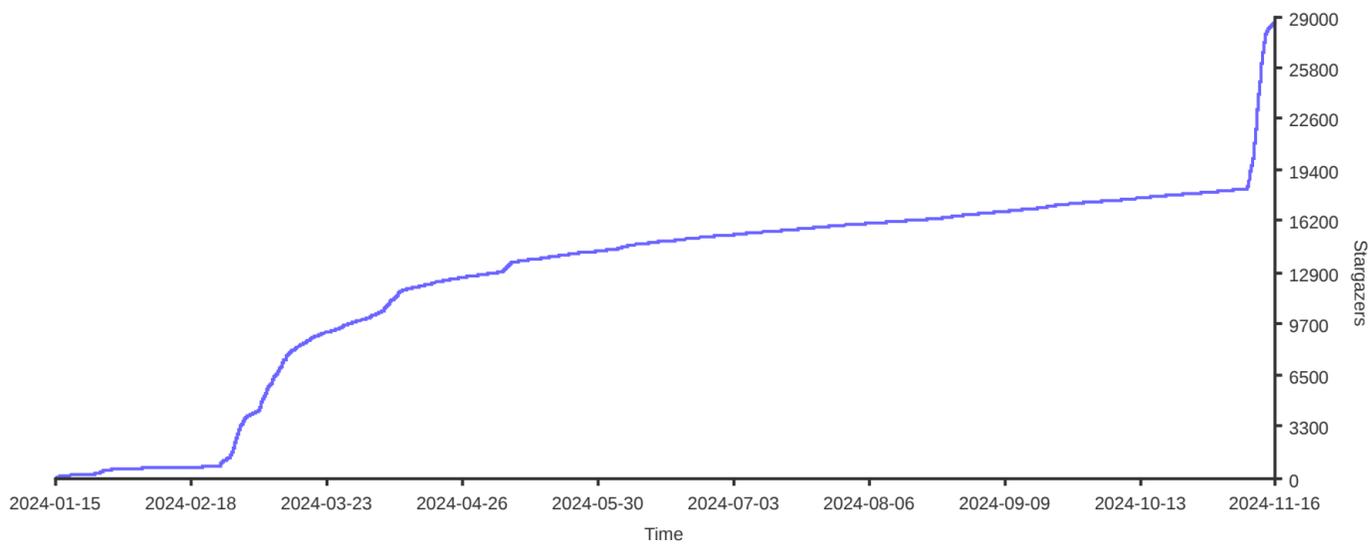
How do I run macOS in a container?

You can use [dockur/macOS](#) for that. It shares many of the same features, except for the automatic installation.

Is this project legal?

Yes, this project contains only open-source code and does not distribute any copyrighted material. Any product keys found in the code are just generic placeholders provided by Microsoft for trial purposes. So under all applicable laws, this project will be considered legal.

Stars



Disclaimer