.devcontainer	feat: Improve Github...	last week
.github	build: Add code quali...	5 days ago
assets	feat: Disable Networ...	3 weeks ago
src	fix: Remove unneces...	3 days ago
.dockerignore	build: Validate JSON ...	last week
.gitignore	build: Initial Dockerfi...	last year
Dockerfile	build: Update QEMU ...	last week
compose.yml	fix: Remove port 80 (...	7 months ago
kubernetes.yml	fix: Remove port 80 (...	7 months ago
license.md	Create license.md	last year
readme.md	docs: Update docker...	last week

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
48k stars
229 watching
3.6k forks
Report repository

Packages 1

windows

Contributors 26

+ 12 contributors

Languages

- Shell 99.1%
- Dockerfile 0.9%

README | MIT license

Windows

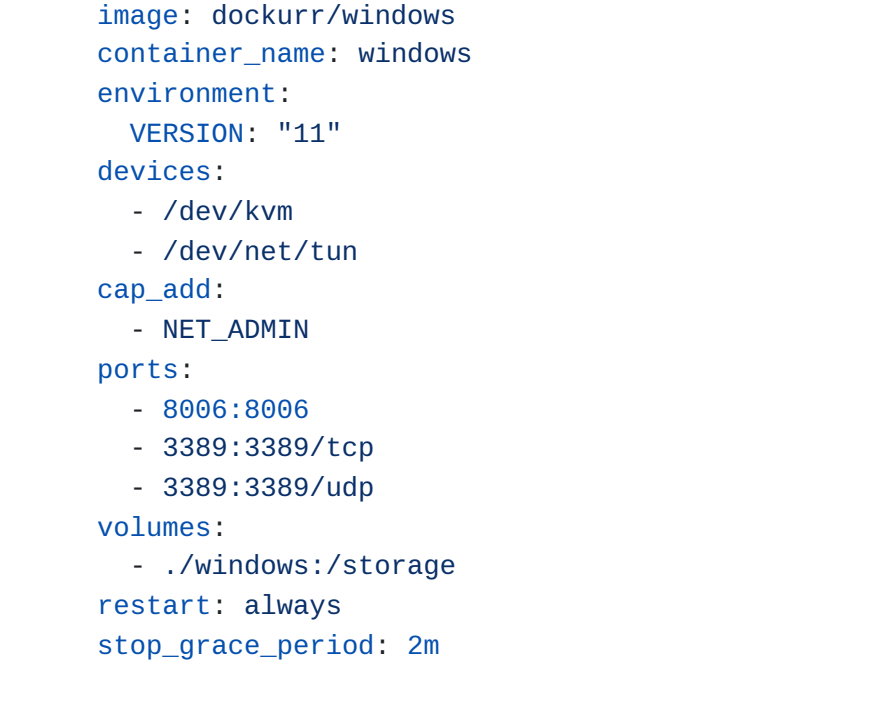
Build passing | version v5.12 | size 128.8 MiB | pulls 177.5k | pulls 1.5M

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"
    devices:
      - /dev/kvm
      - /dev/net/tun
    cap_add:
      - NET_ADMIN
    ports:
      - 8006:8006
      - 3389:3389/tcp
      - 3389:3389/udp
    volumes:
      - ./windows:/storage
    restart: always
    stop_grace_period: 2m
```

Via Docker CLI:

```
docker run -it --rm --name windows -e "VERSION:
```

Via Kubernetes:

```
kubectl apply -f https://raw.githubusercontent
```

Via Github Codespaces:

Open in GitHub Codespaces

Via a graphical installer:

Download BETA

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 Pro 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	7.2 GB
11l	Windows 11 LTSC	4.7 GB
11e	Windows 11 Enterprise	6.6 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
7u	Windows 7 Ultimate	3.1 GB
vu	Windows Vista Ultimate	3.0 GB
xp	Windows XP Professional	0.6 GB
2k	Windows 2000 Professional	0.4 GB
2025	Windows Server 2025	6.7 GB
2022	Windows Server 2022	6.0 GB
2019	Windows Server 2019	5.3 GB
2016	Windows Server 2016	6.5 GB
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:
  - ./windows:/storage
```

Replace the example path `./windows` with the desired storage folder or named volume.

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. However you will need to manually extend the disk partition since the added disk space will appear as unallocated.

How do I share files with the host?

After installation there will be a folder called `Shared` on your desktop, which can be used to exchange files with the host machine.

To select a folder on the host for this purpose, include the following bind mount in your compose file:

```
volumes:
  -  ./example:/shared
```

Replace the example path `./example` with your desired shared folder, which then will become visible as `Shared`.

How do I change the amount of CPU or RAM?

By default, Windows will be allowed to use 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 and its password is `admin`.

If you want to use different credentials during installation, you can configure 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 to be downloaded:

```
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 `KEYBOARD` and `REGION` variables like this:

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

How do I install a custom image?

In order to download an unsupported ISO image, specify its URL in the `VERSION` environment variable:

```
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:
  - ./example.iso:/boot.iso
```

Replace the example path `./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:
  -  ./example:/oem
```

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

How do I perform a manual installation?

It's recommended 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 at your own risk, add the following environment variable to your compose file:

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

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 password `admin`.

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, in which the container and Windows will have separate IP addresses, add the following lines to your compose file:

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

How do I add multiple disks?

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

```
environment:
  DISK2_SIZE: "32G"
  DISK3_SIZE: "64G"
volumes:
  - ./example2:/storage2
  - ./example3:/storage3
```

How do I pass-through a disk?

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

```
devices:
  - /dev/sdb:/disk1
  - /dev/sdc1:/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,
devices:
  - /dev/bus/usb
```

If the device is a USB disk drive, please wait until after the installation is fully 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?

First check if your software is compatible using this chart:

Product	Linux	Win11	Win10	macOS
Docker CLI	✅	✅	❌	❌
Docker Desktop	❌	✅	❌	❌
Podman CLI	✅	✅	❌	❌
Podman Desktop	✅	✅	❌	❌

After that 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 did not receive any error from `kvm-ok` but the container still complains about a missing KVM device, it could help to add `privileged: true` to your compose file (or `sudo` to your `docker` 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.

How do I run a Linux desktop in a container?

You can use qemus/qemu in that case.

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.

Disclaimer ⚖️

The product names, logos, brands, and other trademarks referred to within this project are the property of their respective trademark holders. This project is not affiliated, sponsored, or endorsed by Microsoft Corporation.

Terms Privacy Security Status Community Docs Contact Manage cookies
Do not share my personal information

© 2025 GitHub, Inc.