



Windocks: a Modern, Open, Data Delivery Platform

Windocks Command Line Reference

Windocks is a port of Docker's open source to Windows, and supports Windows 8.1 and 10 Pro and Enterprise editions, Windows Server 2012 R2, and Windows Server 2016. Windocks provides support for .NET, all editions of SQL Server 2008 onward, and Java and a variety of open source projects.

Docker clients for Mac, Linux, Windows

The WinDocks Windows client is included in the install package in the bin directory. Docker 1.7.0 clients for Mac, Linux and Windows can be downloaded using the following links:

[Docker Mac client](#)

[Docker Windows client](#)

[Docker Linux client](#)

Windocks as a Windows Service

Windocks installs as a Windows Service with automatic start. Windocks is accessed with standard Docker client software, or with the Windocks web UI. For remote access, ensure the Windows firewall is configured to allow inbound traffic on container ports 10,000 to 10,200, the Service port of 2375, and the default SQL Server port of 1433.

To access Windocks through the web UI, use either Chrome or Firefox browsers using the IP address of the Windocks server (127.0.0.1 for local use).

WinDocks Commands:

WinDocks supports the following Docker commands. Remote clients must use a more complex syntax:

```
docker -H=tcp://docks.host.ip.add:2375 <command> <options> <args>
```

The docker client can also be used locally on the host, without the **"-H=tcp:// . . . "** string:

```
docker <command> <options> <args>
```

Windocks installs the client on the system path of the host, so the following commands are accessed from the command line on any directory path. For use of these commands on other systems navigate to the directory where the client is located.



Windocks: a Modern, Open, Data Delivery Platform

>docker build -t <imagename> <directory>	<u>Builds an image only.</u> This command supports building SQL Server clonable images.
>docker build <directory>	Builds an image and container, with the image named with the container name. <u>Does not support building an image with cloned databases.</u>
>docker commit <option> <instruction><containerid> <imagename>	Commits a container to create a new image. <u>Does not support creating images from containers with cloned databases.</u>
>docker create <option> <image>	Creates a new stopped container.
>docker exec <containerid> <command>	Executes a command in a container. Note, Windocks lacks the ability to write output to a shell.
>docker images	Lists available images on the host.
>docker ps	Lists containers on the host.
>docker rm <containerid>	Removes (deletes) containers. Use 2-3 digits to achieve a unique match.
>docker rmi <imagename>	Removes image, use the full name.
>docker run -d <option> <image>	Creates a running container.
>docker start <containerid>	Starts the container
>docker stop <containerid>	Stops the container

Containers created with **create** or **run -d** can include assigned ports, SA_PASSWORD and names. A commit can include a Dockerfile instruction to **ADD, COPY, or RUN**, with the **-change** option.

-e SA_PASSWORD="sa_password" Note: include a suitably complex password.

-p <port>

--name <containername> Note: use lowercase in image names.

--change "instruction file"



Windocks: a Modern, Open, Data Delivery Platform

`--cidfile=<path to folder>`

`-e SQL_DB_NAME_OVERRIDES="db1, db2"`

Allows users to select a subset of databases from a larger image.

DockerFiles:

Dockerfiles are plain text configuration files that support the creation of containers and images. A number of examples are included in the `\windocks\samples` folders. The Docker client copies all files and folders that are located with the Dockerfile to the host. It is important to include only files that are desired in the container or image to be located with the Dockerfile.

Supported Dockerfile commands include:

`FROM<image>`

`ADD <file> .`

`ADDDDB <dbname> <mdf> <ndf> <ldf>`

`COPY <file> .`

`ENV <environment variable>` note: SA_PASSWORD not supported at this time.

`ENV USE DOCKERFILE_TO_CREATE_CONTAINER`

`EXPOSE <port>`

`MOUNTDB <dbname> \path\to\db.mdf \path\to\db_log.ldf` Universal paths for network located files.

`RUN <file>`

`RUN TargetAttach_SqlWindows <parameters>` Ids the target as a Windows SQL instance

`RUN SourceClone_San_Pure <parameters>` Ids a Pure Array as source

`SETUPCLONING FULL <dbname> <path to Full backup>`

`SETUPCLONING DIFF <dbname> <path to Differential backup>`

`SETUPCCLONING RAW <dbname> <path to DB files>`

Note, **SETUPCLONING** instructions must be used with the `>docker build -t` command. For network attached file shares use universal file paths.



Windocks: a Modern, Open, Data Delivery Platform

Database files referenced by the ADDDB should be located in the same directory as the Dockerfile.

Resources:

[Getting Started with SQL Server containers with in container data](#)

[Getting Started with SQL Server containers and database clones](#)

[Windocks Installation and Configuration](#)

<https://windocks.com/blog-2/docker-windows-containers-and-DNS>

https://windocks.com/files/WinDocks_Licensing_and_Support.pdf

For information on working with multi-tier environments, including .NET see:

<https://windocks.com/lps/gitbuildtest>

For technical support email: support@windocks.com

For information on how Windocks compares to Microsoft's new containers in Windows Server 2016:

<https://windocks.com/blog-2/Windows-Containers-Compared-Windocks-Microsoft>

About Windocks

Windocks combines Docker Windows containers with SQL Server database cloning, for a modern, open data delivery solution. Enterprises modernize application development, testing, reporting and BI with existing licenses and infrastructure, at a fraction the cost of alternatives.

For additional information, visit www.windocks.com, or contact Windocks at info@windocks.com

Windocks
Data Delivered



Microsoft Partner

