

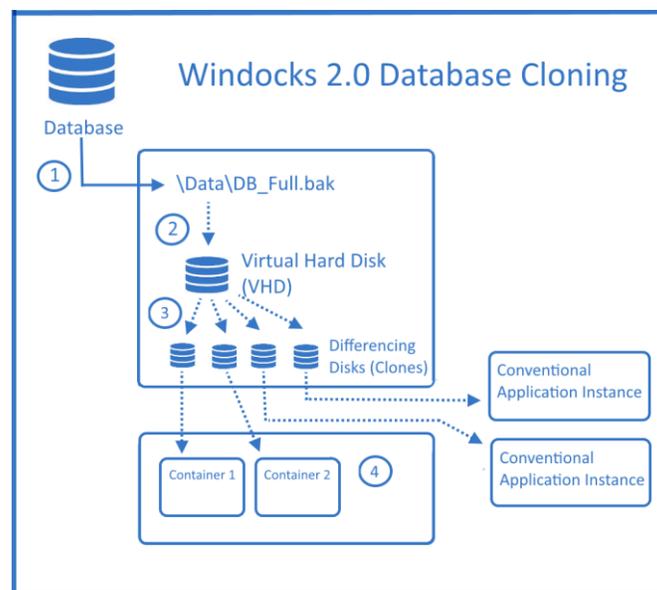


Windocks: a Modern, Open, Data Delivery Platform

SQL Server Cloned Databases and Containers

SQL Server cloned databases support delivery of large, writable SQL Server environments, in seconds, with minimal storage requirements. A companion article describes the use of smaller databases run in the container's private file system, see [Getting Started with SQL Server containers and in-container databases](#). This article is focused on Windows based SQL Server cloning. Please see separate articles for details in working with Storage Array snapshots.

SQL Server clones are based on Windows Virtual Hard Drives (VHD), and support creation of "differencing disks" which are writable "cloned" database environments. The VHD is created with databases defined by a Dockerfile (a plain text configuration file), using Full and Differential backups, or snapshots, and SQL Server scripts. The VHD is a full byte copy of the data, and is included in the SQL Server image. Clones are created from the parent VHD, and mounted to either SQL Server containers or instances.



Refer to <https://windocks.com/lps/sqlall#share> for details on Windocks use with network file shares.

Database clones are created in seconds, occupy <40 MB, and are writable. Storage requirement for each clone grows dynamically using a Copy on Write design. Clones share read-only access to the parent VHD, and are well suited for Development and Test use, but are not recommended for performance and database stress testing.

SQL Server clone images are defined with a Dockerfile that includes one or more **SETUPCLONING** commands that build the VHD either on the Windocks host or network based file share, and normally includes **RUN TargetAttach** with details defining the delivery target. The default Windocks targets are local SQL Server containers, so the dockerfile below takes a simplified form. SQL Server scripts can also



Windocks: a Modern, Open, Data Delivery Platform

be applied during the image build. The example below is located in the \windocks\samples directory. Network located files would use a universal network path.

```
Dockerfile - Notepad
File Edit Format View Help
FROM mssql-2012
SETUPCLONING FULL customers C:\windocks\dbbackups\customerdatafull.bak
COPY cleanseData.sql .
RUN cleanseData.sql
```

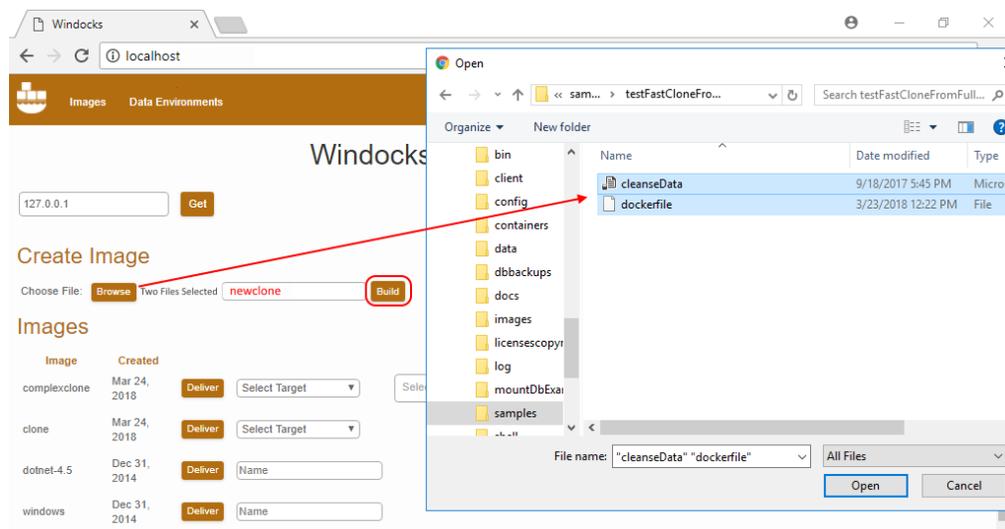
The dockerfile shown is built with the Docker command:

>docker build -t <imagename> c:\windocks\samples\testfastclonefromfullbackup

Build a Clonable SQL Server image using the web UI

Open a Firefox or Chrome browser on the Windocks host, and direct the browser to localhost. Once the Windocks web UI resolves enter the local loopback address: **127.0.0.1** in the Windocks Server IP address box, and select “Get.” Remote clients can access the server (if firewalls are properly configured) using the Windocks server IP.

To build a SQL Server image click the “choose files” tool and navigate to the Dockerfile and scripts (\windocks\samples\testfastclonefromfullbackup). Highlight the dockerfile and script, right-click and “select,” enter an image name and “build.” The Dockerfile and scripts are copied to the Windocks service for execution. The SETUPCLONING commands are executed on the host, and create the VHD on the host or on a network attached file share alongside the backup files.

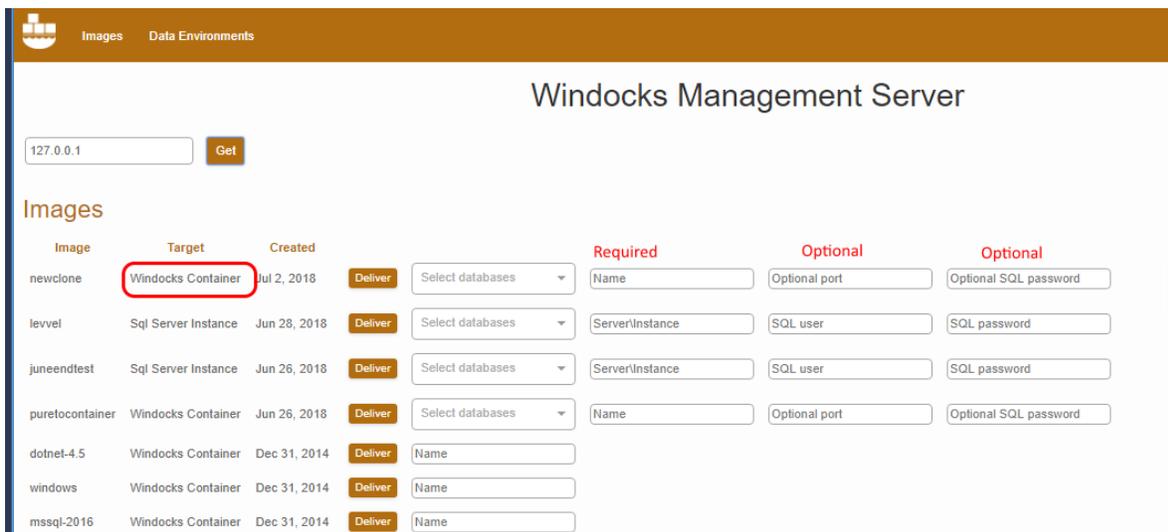




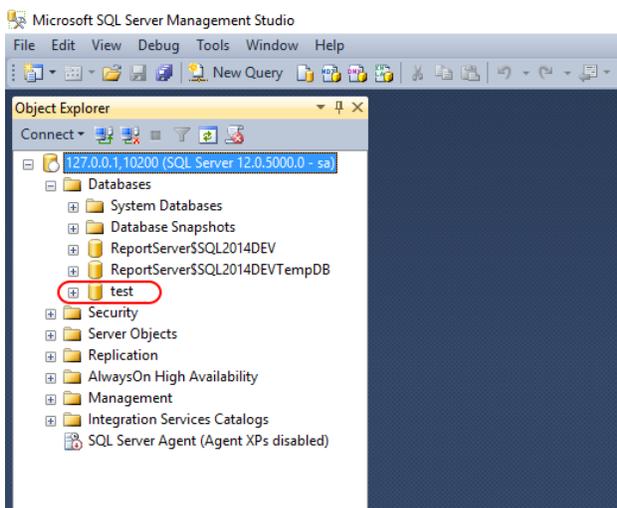
Windocks: a Modern, Open, Data Delivery Platform

Images can include scores of databases, with each included in the Dockerfile. Building the image takes time normally associated with restoring the backup, as the image is a full byte copy of the database.

On completion the UI updates to show the new image. Clonable images support delivery to SQL Server instances or containers, based on the Dockerfile. In the Dockerfile above the default target are local SQL Server containers. Assign a container name, and optional port and SQL sa password. The database selector tool allows for delivery of a subset of the image (leave blank for delivery of the complete image). Selecting the “Deliver” button initiates the delivery.



New containers and clones are delivered in seconds, regardless of the VHD size. Starting the container attaches the cloned databases. Remote clients use the host IP address (port separated by a comma), and the SQL sa credentials to access.

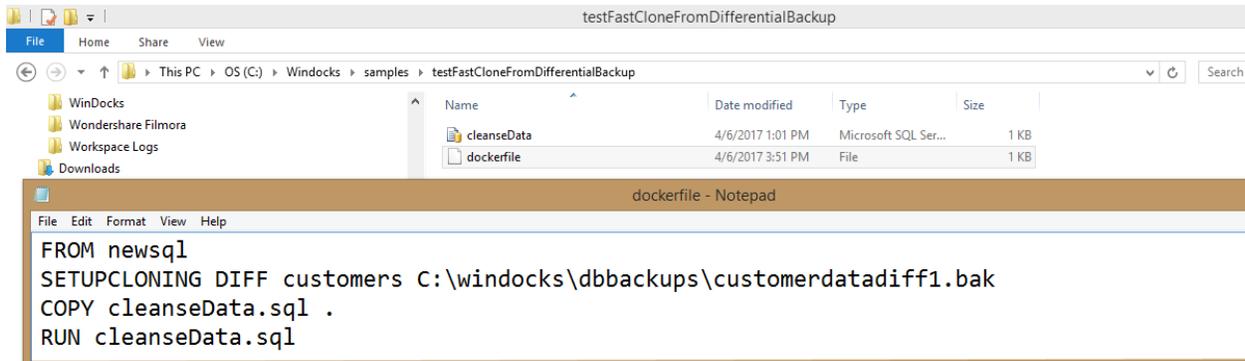




Windocks: a Modern, Open, Data Delivery Platform

Using the Web UI to create updated SQL Server images:

Images that include cloned databases are updated with Differential backups, or scripts, through another image build. In this case we use a Differential backup that is included in the \windocks\dbbackups folder.



The update is applied to the parent image (**newsql** in our example), and applies a differential backup with the **SETUPCLONING DIFF** command. SQL Server scripts are copied into the container during the build. Any files that need to be copied are located in the same folder as the Dockerfile. Note: image names are case sensitive!

Updated images based on backups depends on the Full backup parent image, and becomes another full byte copy of the environment. Updated images can be deleted and replaced, but require access to the original parent image built from the Full backup. To build the updated image using the web UI, select the “choose files” tool and navigate to the Dockerfile and script, highlight both files, right-click and “select.” Assign a name to the image, and click “create.”

Working with the Command Line

The docker client on the Windocks host is included in the system path, and is available via a standard command prompt window or PowerShell on any directory path. Additional Windows clients can be copied from the \windocks\client directory and installed on other machines.

The Windocks CLI supports standard Docker commands. When working from a remote client, the syntax takes on an added complexity, and requires the host firewall to be configured to allow inbound traffic on ports 10000 – 10200, 2375, and SQL Server port 1433.

>docker -H tcp://ip.address.of.host:2375 images Remote client call for >docker images

Docker commands used locally on the Windocks host use the simpler syntax shown below:

>docker images Presents a list of available images on the host



Windocks: a Modern, Open, Data Delivery Platform

```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\windows\system32>docker run -d -p 10050 --name pauls clone
ContainerId = d70a186e5a64c9a03b3ec705ac7e14572ee3325e58f58f8d7f779c34284a2cad & ContainerPort =
ontainerUserName = prison_oo_jHL9o3S & ContainerPassword = Pr!5uM60Ph7gDg & MSSQLPort = 10050 &
rUserName = sa & MSSQLServerSaPassword = Pr!5sjhr9ZW6he

C:\windows\system32>
```

The container is accessible via SQL Management Studio and other tools, for remote access use the SQL sa credentials (use either the local loopback address, with comma separator: 127.0.0.1,10050 or the machine IP address,port).

To update the image, build a new image including a SQL Server change script to the Full backup. Or, apply a Differential backup with or without a script. In the example below the image refers to the parent VHD (clone) that is updated with a differential backup, and includes the same data masking scripts used earlier.

```
dockerfile - Notepad
File Edit Format View Help
FROM clone
SETUPCLONING DIFF customers C:\windocks\dbbackups\customerdatadiff1.bak
COPY cleanseData.sql .
RUN cleanseData.sql
```

The build uses the **SETUPCLONING DIFF** command. The container delivered below is from the source clone image, and can be disregarded. The newclone image is confirmed with a **>docker images** command.

```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\windows\system32>docker build -t newclone c:\windocks\samples\testfastclonefromdifferentialbackup
Sending build context to Docker daemon 3.072 kB
Sending build context to Docker daemon
Step 0 : FROM clone
Step 1 : SETUPCLONING DIFF customers C:\windocks\dbbackups\customerdatadiff1.bak
Step 2 : COPY cleanseData.sql
Step 3 : RUN cleanseData.sql
ContainerId = 8ca0f9dd00a58c9e9cca7da42669b3e1339b2948431bf5be1fd96a46529b739a & ContainerPort = 10001 & C
ontainerUserName = prison_oo_uDK34jc & ContainerPassword = Pr!58jU36FQb08 & MSSQLPort = 10001 & MSSQLServe
rUserName = sa & MSSQLServerSaPassword = Pr!53cc2acgWmd & DockerFile output:
C:\windows\system32>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
dotnet-4.5          none               dotnet-4.5         2 years ago        0 B
windows             none               windows            2 years ago        0 B
mssql-2012          none               mssql-2012        2 years ago        0 B
newsq1              none               662f5e70-7d0      3 days ago         0 B
newclone            none               f01d6504-cb7      4 seconds ago     0 B
clone               none               3a88e15e-2d3      3 days ago         0 B

C:\windows\system32>
```



Windocks: a Modern, Open, Data Delivery Platform

SQL Server containers with Cloned databases

The processes outlined in this article are ideal for delivery of Dev and Test environments of any size. Each container is delivered in approximately 30 seconds, irrespective of the size of the image, is writable, and only occupies <40 MB on delivery. A team can work with scores of identical environments on a single shared server, and save hours each week in VM maintenance (and on costs of VM infrastructure). Containers are created quickly, and is ideal for short-lived instances needed for Dev and Test. The web UI provides Developers an easy to use self-service environment.

An additional advantage is that the cloned databases can be used with conventional SQL Server instances and Microsoft SQL Server containers. So, Windocks is both a container engine as well as general purpose SQL Server cloning solution.

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

