PASS MARATHON

Database Cloning for SQL Server Containers

SQL containers without database clones is like a rocket ship without a payload

Paul Stanton, VP & Co-founder, Windocks
Moderated By: Jan Mulkens

Immerse yourself

in the data community

Access deep-dive technical sessions, learn best practices, and discover new tips and tricks

Gain the technical skills and connections to advance your data career

Attend

PASS Summit

PASS

SUMMIT2018

NOV 6-9 | SEATTLE WA

PASS Summit is the largest conference for technical professionals who leverage the Microsoft Data Platform.

See everything PASS Summit has to offer at PASSsummit.com

More info:

Paul Stanton

VP, Co-founder, Windocks

https://windocks.com

in /paulstan

@paul_e_stanton

✉ pauls@windocks.com

Microsoft 1990 to 2000

Co-led Windows NT internetworking (Steelhead project), and development of the Microsoft/Cisco alliance. Prior to leaving I led Enterprise Marketing.

Early involvement in Containers

Involved in container startups focused on Cloud Foundry and OpenShift, and helped develop the first Windows support for both.

Windocks

Formed a team of former Microsoft engineers to deliver an independent port of Docker’s source to Windows, focusing on full stack support with SQL Server. Recently extended support to storage arrays and all SQL Server targets.

PASS
MARATHON

Database Cloning for SQL Server Containers

SQL containers without database clones is like a rocket ship without a payload

Paul Stanton, VP & Co-founder, Windocks
Moderated By: Jan Mulkens

Agenda

Understand SQL Server database clones delivered from storage arrays or Windows Virtual Hard Drives (VHDs), for SQL Server containers

- The motivation for clones
- What are database clones?
- Pros and Cons
- Alternatives
- Building and delivering clones to SQL Server containers

Why database cloning?

Fast, Secure Data Delivery for complete environments

- Speed: TB databases are delivered in seconds
- Functionality: Read/Write support for complex database environments, and can support enterprise EKM, TDE, and other enterprise infrastructure and processes
- Economy: each clone requires <40 MB, resulting in 99% storage reduction
- Security: built into Docker images, creating immutable, auditable artifacts that incorporate security policies of the org/team
- Freedom: use any source (storage arrays or backups), and deliver to any target (MS containers, instances, and K8), all managed by SQL DBAs or Developers, independent of the storage admins

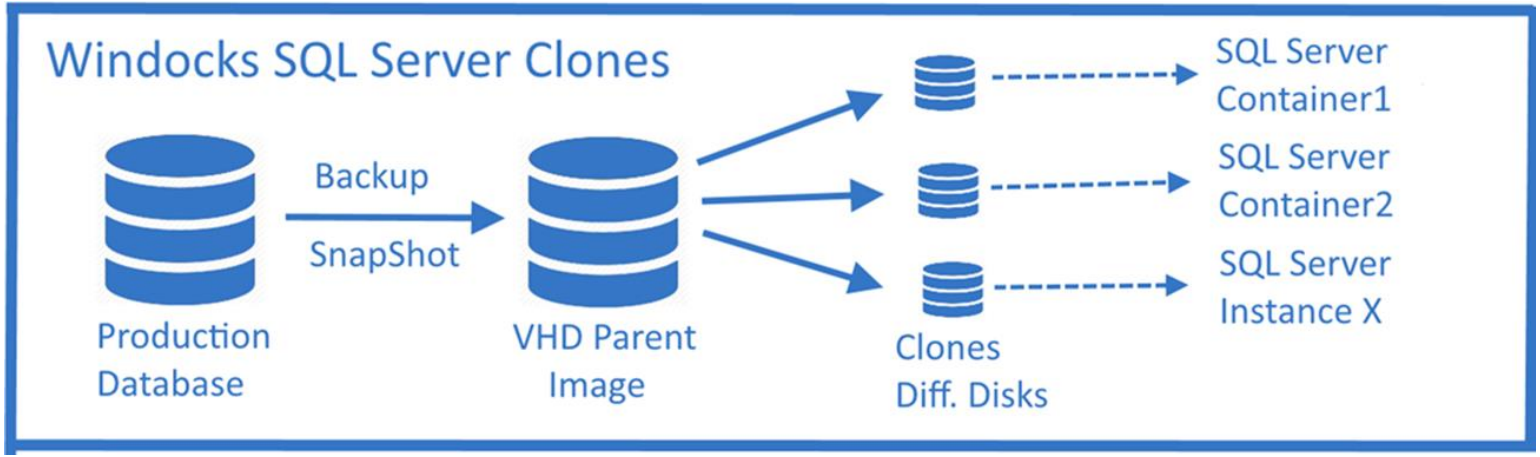
What is a database clone?

A Windows VHD is a file encapsulating a virtual disk. Differencing disks ("clones") are thin-provisioned copies, with read-only file pointers to the parent VHD

- The "Parent" VHD is a full byte copy of the environment, used as a read-only source
- Each clone (differencing disk) expands dynamically with Copy-on-Write
- VHDs are part of Windows, but associated with Hyper-V support of VMs. Windows VHDs run wherever Windows is supported, enabling database cloning on AWS or any on-premise infrastructure, along with Azure
- All storage arrays support volume snapshots or clones with file pointers to a source volume

Building a database clone with backups

A parent VHD is created and mounted to a SQL Server instance. Backups are restored and scripts are run. On completion the VHD is un-mounted, and saved as a read-only source for clones.



Pros and Cons of Database Clones

Pros:

- Fast, economical, scalable, and lightweight
- Ideal for Dev/Test, ETL, reporting
- Flexible: supports Docker, CI, and user-driven provisioning
- Independence from Storage Admins
- Support SQL Server instances, for a complete data delivery strategy

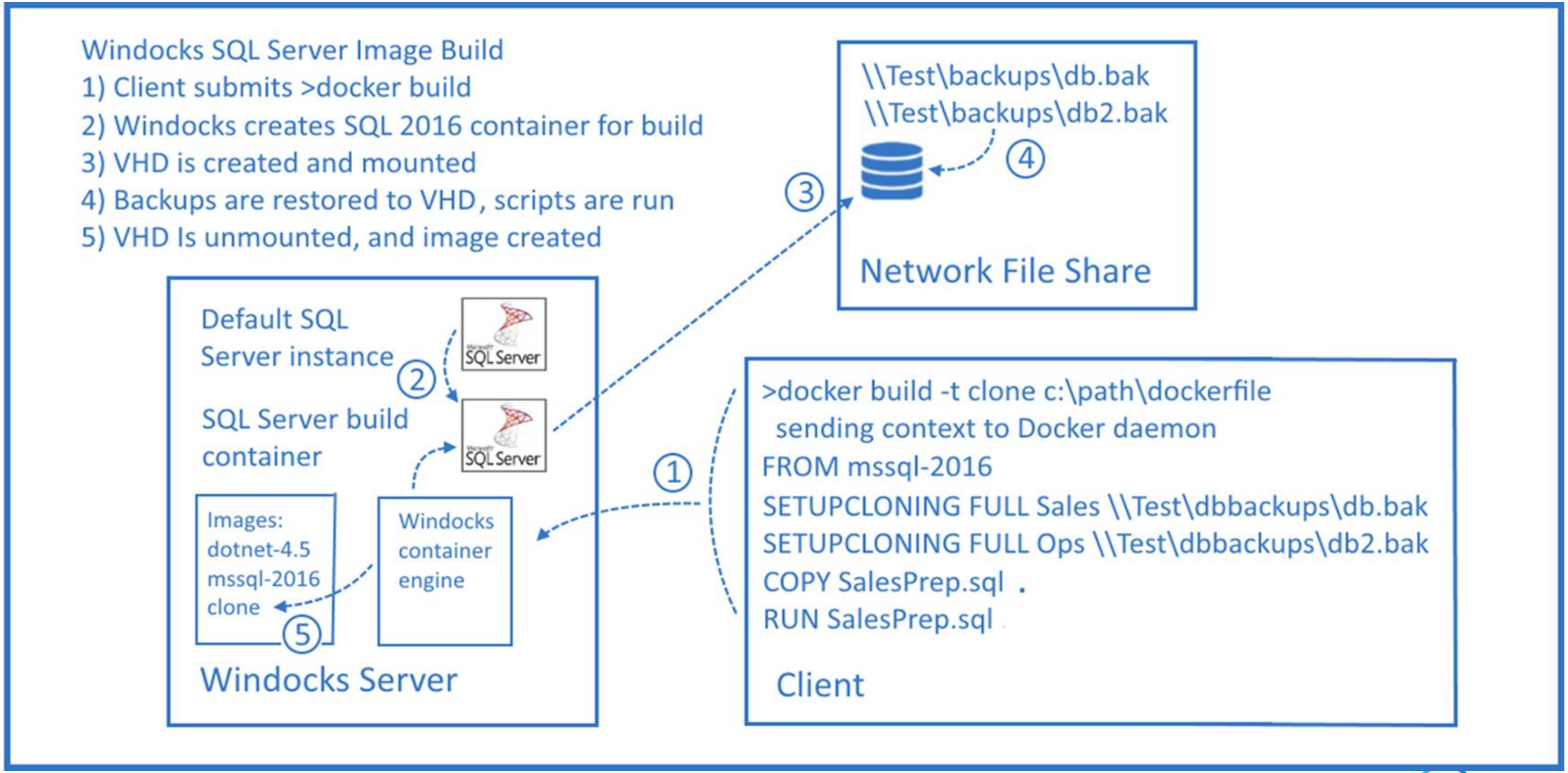
Cons:

- Windows VHD parent build is time consuming due to use of backups (storage arrays don't have this issue)
- Clones, with shared parent disk, are not suited for stress testing or production

Alternative Methods for Data Delivery

- Local Container File System: works great for small data environments, but becomes unworkable as data sets grow in size
- Mounting local or network based volumes.   This requires time and storage to create sufficient copies for users
- Database clones as we’re discussing . . . .

Building a SQL Server Clonable Image

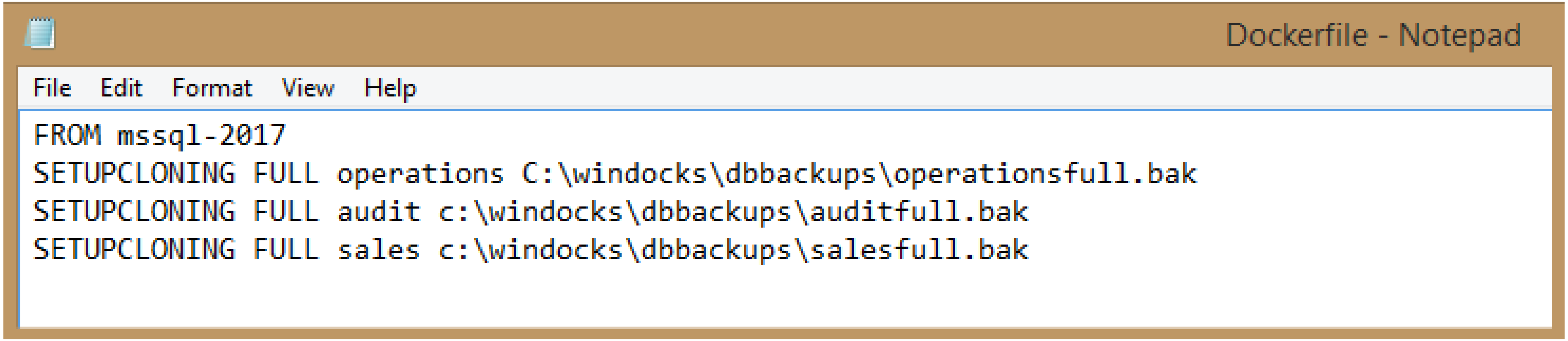


Windocks SQL Server Image Build

1) Client submits >docker build
2) Windocks creates SQL 2016 container for build
3) VHD is created and mounted
4) Backups are restored to VHD, scripts are run
5) VHD Is unmounted, and image created

```
>docker build -t clone c:\path\dockerfile
 sending context to Docker daemon
FROM mssql-2016
SETUPCLONING FULL Sales \\Test\dbbackups\db.bak
SETUPCLONING FULL Ops \\Test\dbbackups\db2.bak
COPY SalesPrep.sql .
RUN SalesPrep.sql
```

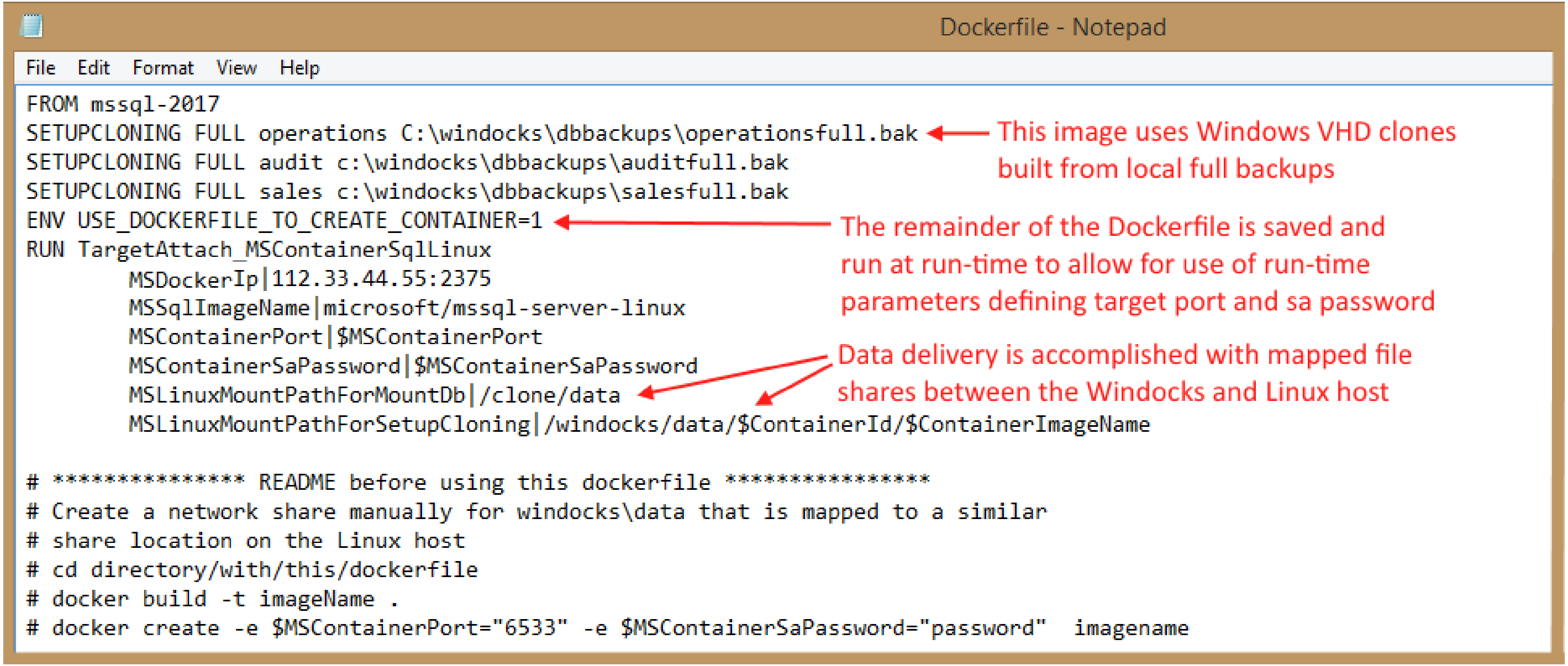
Dockerfile to build a Full Backup based VHD Image



```
FROM mssql-2017
SETUPCLONING FULL operations C:\windocks\dbbackups\operationsfull.bak
SETUPCLONING FULL audit c:\windocks\dbbackups\auditfull.bak
SETUPCLONING FULL sales c:\windocks\dbbackups\salesfull.bak
```

- The VHD will be built in the same location as the backup
- Network located backups would require Universal File Paths

Dockerfile for delivery to Linux SQL 2017 container



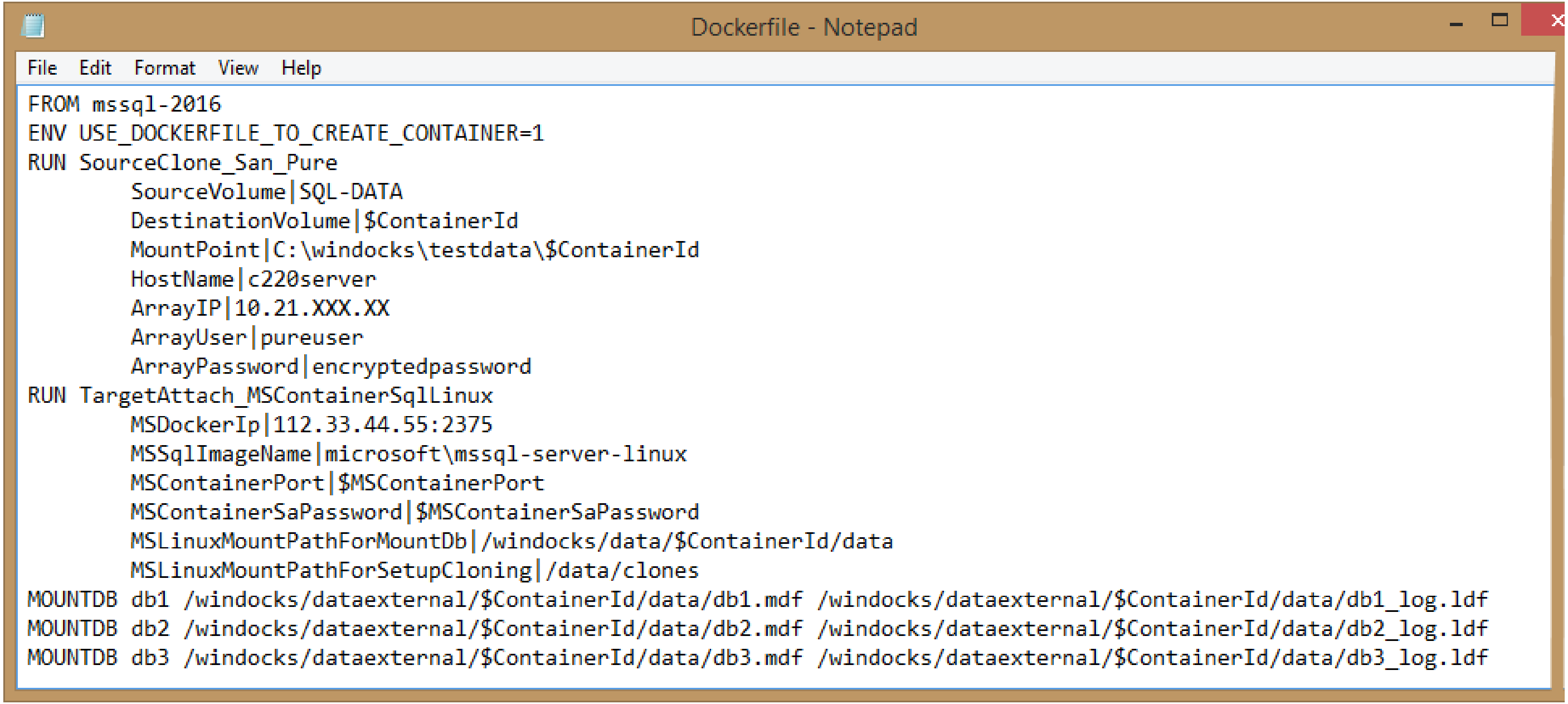
```
FROM mssql-2017
SETUPCLONING FULL operations C:\windocks\dbbackups\operationsfull.bak
SETUPCLONING FULL audit c:\windocks\dbbackups\auditfull.bak
SETUPCLONING FULL sales c:\windocks\dbbackups\salesfull.bak
ENV USE_DOCKERFILE_TO_CREATE_CONTAINER=1
RUN TargetAttach_MSContainerSqlLinux
        MSDockerIp|112.33.44.55:2375
        MSSqlImageName|microsoft/mssql-server-linux
        MSContainerPort|$MSContainerPort
        MSContainerSaPassword|$MSContainerSaPassword
        MSLinuxMountPathForMountDb|/clone/data
        MSLinuxMountPathForSetupCloning|/windocks/data/$ContainerId/$ContainerImageName

# *************** README before using this dockerfile ****************
# Create a network share manually for windocks\data that is mapped to a similar
# share location on the Linux host
# cd directory/with/this/dockerfile
# docker build -t imageName .
# docker create -e $MSContainerPort="6533" -e $MSContainerSaPassword="password"  imagename
```

This image uses Windows VHD clones built from local full backups

The remainder of the Dockerfile is saved and run at run-time to allow for use of run-time parameters defining target port and sa password

Data delivery is accomplished with mapped file shares between the Windocks and Linux host

Using a Pure Storage array w/delivery to SQL 2017 container



```
FROM mssql-2016
ENV USE_DOCKERFILE_TO_CREATE_CONTAINER=1
RUN SourceClone_San_Pure
        SourceVolume|SQL-DATA
        DestinationVolume|$ContainerId
        MountPoint|C:\windocks\testdata\$ContainerId
        HostName|c220server
        ArrayIP|10.21.XXX.XX
        ArrayUser|pureuser
        ArrayPassword|encryptedpassword
RUN TargetAttach_MSContainerSqlLinux
        MSDockerIp|112.33.44.55:2375
        MSSqlImageName|microsoft\mssql-server-linux
        MSContainerPort|$MSContainerPort
        MSContainerSaPassword|$MSContainerSaPassword
        MSLinuxMountPathForMountDb|/windocks/data/$ContainerId/data
        MSLinuxMountPathForSetupCloning|/data/clones
MOUNTDB db1 /windocks/dataexternal/$ContainerId/data/db1.mdf /windocks/dataexternal/$ContainerId/data/db1_log.ldf
MOUNTDB db2 /windocks/dataexternal/$ContainerId/data/db2.mdf /windocks/dataexternal/$ContainerId/data/db2_log.ldf
MOUNTDB db3 /windocks/dataexternal/$ContainerId/data/db3.mdf /windocks/dataexternal/$ContainerId/data/db3_log.ldf
```

Database Cloning Resources:

- MSDN on VHDs: https://msdn.microsoft.com/en-us/library/windows/desktop/dd323654%28v=vs.85%29.aspx
- Pure Storage: https://blog.purestorage.com/empowering-sql-server-dbas-via-snapshots-and-powershell/
- Red Gate provides similar Windows based VHD cloning, but lacks support for SQL Server containers currently. https://www.red-gate.com/hub/product-learning/sql-clone/getting-started-with-sql-clone
- Windocks supports database clones sourced from any storage array, or Windows VHD clones, with delivery to all SQL Server targets (SQL containers, instances, and Kubernetes). Download a free Community Edition at: https://windocks.com/community-docker-windows

PASS MARATHON

Thank you for attending

Learn more from Paul Stanton

@paul_e_stanton   pauls@windocks.com

@sqlpass #sqlpass   @PASScommunity

PASS
MARATHON