

Modern Storage Platforms for SQL Server



Anthony Nocentino

Principal Field Solution Architect @ Pure Storage

- Specializes in system architecture, performance, SQL Server, Kubernetes, Containers, Microsoft Azure, and VMware
- Master's degree in Computer Science

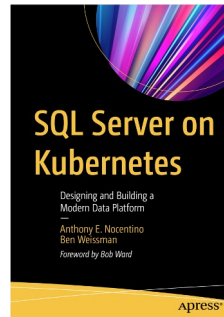
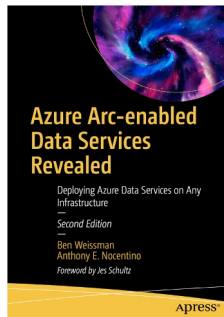
Email: anocentino@purestorage.com

Blog: www.nocentino.com

Twitter: @nocentino

GitHub: <https://github.com/nocentino/>

Pluralsight Author: www.pluralsight.com



Today's Agenda

- **Module 0** – Introduction and prerequisites 10 mins
- **Module 1** – Storage fundamentals for DBAs 45 mins
- **Module 2** – Storage-based snapshots and SQL Server 70 mins
- **Module 3** – SQL Server Object Integration: Backup and Restore 35 mins
- **Module 4** – SQL Server Object Integration: Data Virtualization 45 mins

Take a 15-minute break after each hands-on lab



Hands on Lab Information

SQL Server 2022 and Pure Storage

1

<https://purestorage.skytap-portal.com/>

2

Enter your passcode and open a desktop to **Windows1**

<https://github.com/PureStorage-OpenConnect/ModernStoragePlatformsForSqlServer>

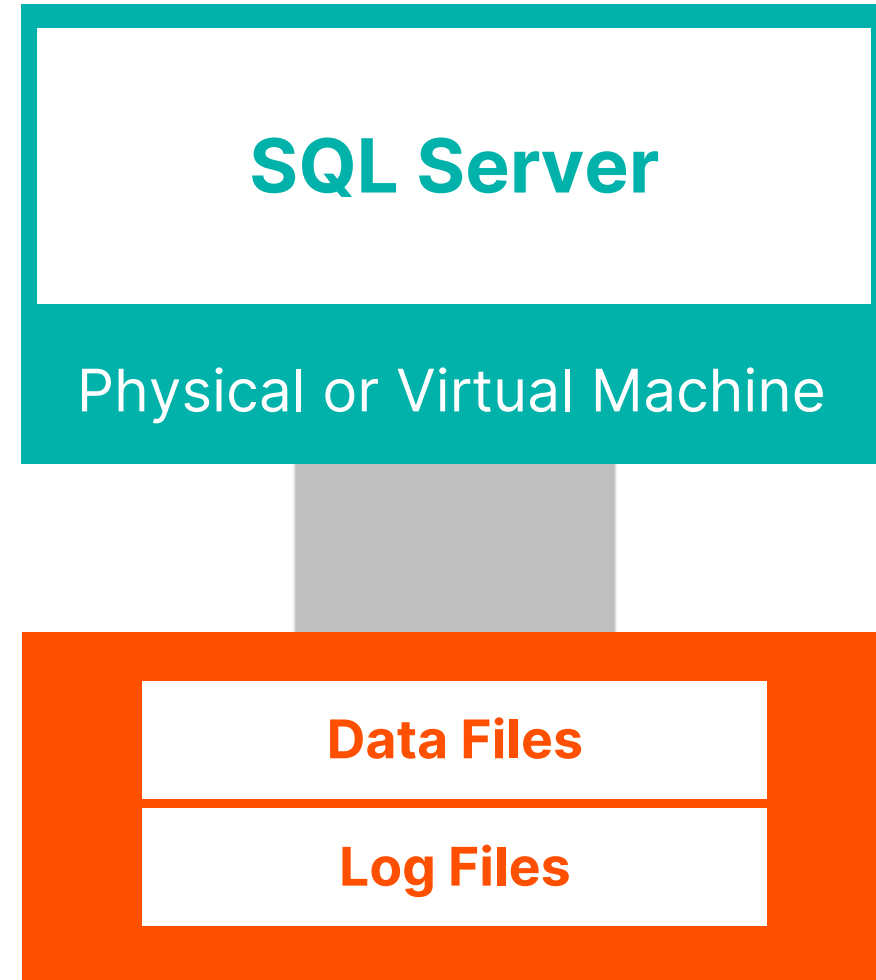


Storage Fundamentals for DBAs

Where data lives in a computer system...

Storage Is Where Data Lives

- SQL Server stores data...
- On disks
 - In files
 - Local
 - SCSI
 - PCIe
- **Remote**
 - Fibre Channel
 - iSCSI
 - NVMe-oF
 - S3 compatible object storage



Key Performance Metrics

Latency

How long a request takes

Transaction log I/Os

Saturation leads to queuing

Throughput

Amount of data moved

Function of the storage interconnect

Type of storage used

IOPs

Number of requests

Size depends on the application



Why Database Systems Care about These...

Latency

Does your system sell things?

Is your system time sensitive?

OLTP

Throughput

Does users need reports?

Backups?

OLAP

IOPs

Larger requests take longer

Often governed in cloud and on premises



Basics of SQL Server I/O

Operation	IO Block Size
Transaction log write	512 bytes - 60 KB
Checkpoint/Lazywriter	8KB - 1MB
Read-Ahead Scans	128KB - 512KB
Bulk Loads	256KB
Backup/Restore	1MB
ColumnStore Read-Ahead	8MB
File Initialization	8MB
In-Memory OLTP Checkpoint	1MB

<https://www.nocentino.com/posts/2021-12-10-sqlserver-io-size/>



Let's Look Closer...at Latency

- Monitor Latency
- **`sys.dm_io_virtual_file_stats`**
- Average per file since instance startup
- Have a monitoring tool

<https://www.nocentino.com/posts/2021-10-06-sql-server-file-latency/>



Let's Look Closer...at Latency

DBName	FileName	FileType	NumReads	NumWrites	ReadBytes	WriteBytes	PercentBytesRead	PercentBytesWrite	AvgReadLatency_(ms)	AvgReadSize_(KB)	AvgWriteLatency_(ms)	AvgWriteSize_(KB)	AvgLatency_(ms)	AvgIOSize_(KB)	PhysicalFileName
tempdb	temp3	ROWS	23	53261	1277952	530538496	0	99	7	54	41	9	41	9	T:\TEMPDB\tempdb_mss
tempdb	temp8	ROWS	43	53981	2129920	546938880	0	99	1	48	41	9	41	9	T:\TEMPDB\tempdb_mss
tempdb	tempdev	ROWS	503	53694	19636224	535355392	3	96	4	38	39	9	39	10	T:\TEMPDB\tempdb.mdf
tempdb	temp4	ROWS	26	53329	1417216	547823616	0	99	0	53	38	10	38	10	T:\TEMPDB\tempdb_mss
tempdb	temp5	ROWS	33	53974	1761280	536707072	0	99	0	52	38	9	38	9	T:\TEMPDB\tempdb_mss
tempdb	temp2	ROWS	25	53815	1351680	546734080	0	99	0	52	38	9	38	9	T:\TEMPDB\tempdb_mss
tempdb	temp7	ROWS	32	53307	1695744	531218432	0	99	0	51	35	9	35	9	T:\TEMPDB\tempdb_mss
tempdb	temp6	ROWS	531	53372	7135232	540041216	1	98	2	13	31	9	30	9	T:\TEMPDB\tempdb_mss
FT_Demo	FT_Demo	ROWS	564	2112	23461888	17309696	57	42	1	40	14	8	11	14	X:\FT_Demo.mdf
model	modellog	LOG	9	17	1015808	40960	96	3	17	110	0	2	6	39	S:\SYSTEM\MSSQL15.MS
tpch100	tpch100	ROWS	33850	1759903	971161600	999181459456	0	99	1	28	5	554	5	544	X:\TPCH\tpch100.mdf
TestDB1_LS	TestDB1_log	LOG	7	0	1007616	0	100	0	4	140	0	0	4	140	S:\SYSTEM\MSSQL15.MS
TestDB1	TestDB1_log	LOG	7	5	1007616	11264	98	1	4	140	0	2	2	82	L:\LOG\TestDB1_log.ldf
tpcc100	tpcc_log	LOG	11	5	1024000	14848	98	1	3	90	0	2	2	63	L:\LOG\tpcc_log100.ldf
TestDB1	TestDB1	ROWS	334	1	13926400	8192	99	0	1	40	0	8	1	40	D:\DATA\TestDB1.mdf
msdb	MSDBData	ROWS	7122	40371	233316352	423354368	35	64	1	31	1	10	1	13	S:\SYSTEM\MSSQL15.MS
msdb	MSDBLog	LOG	52	191943	350720	279290368	0	99	2	6	1	1	1	1	S:\SYSTEM\MSSQL15.MS
model	modeldev	ROWS	78	4	35872768	32768	99	0	1	449	0	8	1	427	S:\SYSTEM\MSSQL15.MS
TestDB1_LS	TestDB1	ROWS	29	0	1671168	0	100	0	1	56	0	0	1	56	S:\SYSTEM\MSSQL15.MS
tpcc100	tpcc	ROWS	665	1	37715968	8192	99	0	1	55	0	8	1	55	D:\DATA\tpcc100.mdf
tpch100	tpch100_log	LOG	15	47682006	1040384	2010226025984	99	99	0	67	1	41	1	41	X:\TPCH\tpch100_log.ldf
tempdb	templog	LOG	44	85605	1294336	5233082368	0	99	4	28	1	59	1	59	T:\TEMPDB\templog.ldf

<https://www.nocentino.com/posts/2021-10-06-sql-server-file-latency/>



Storage Devices

The list is long.



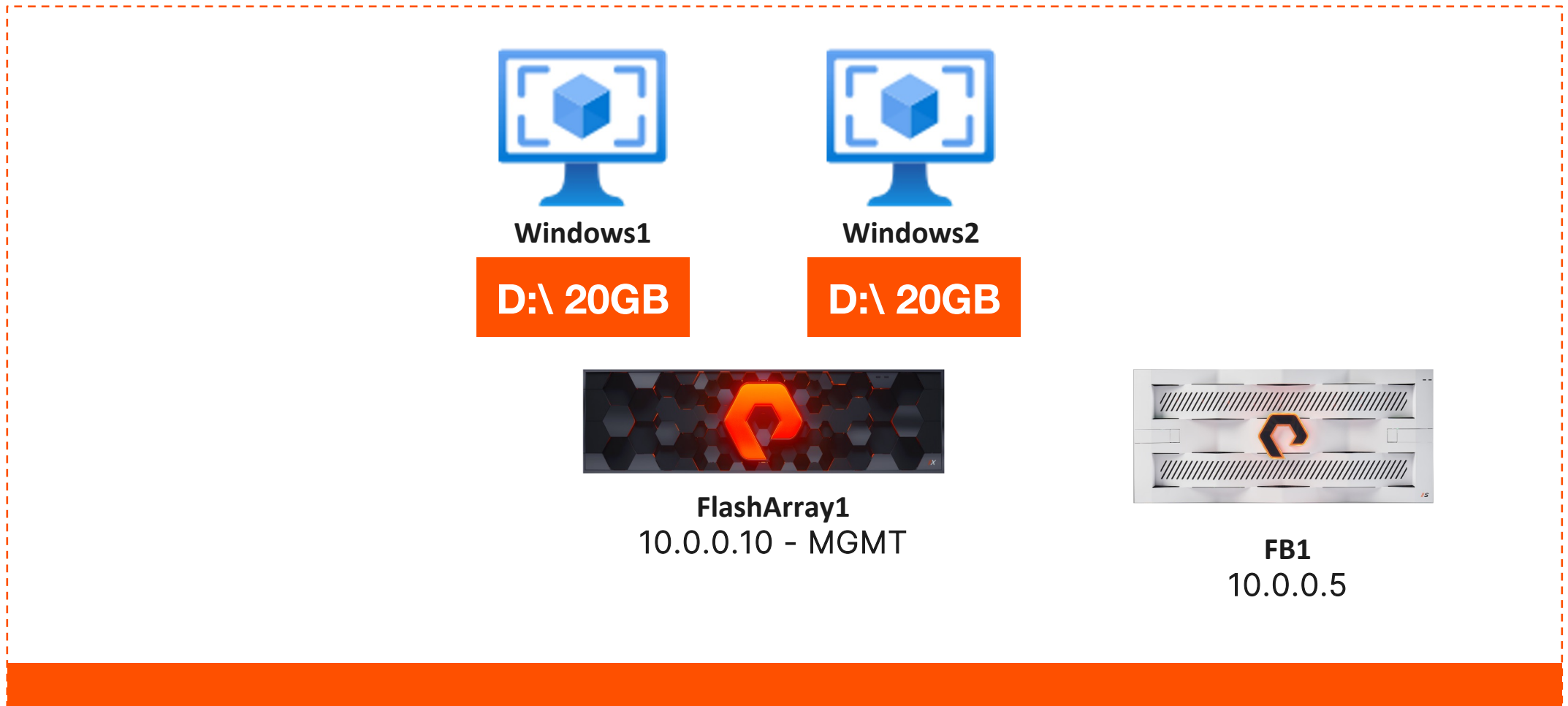
Block IO



Object Storage

Hands On Lab Architecture

SQL Server 2022 and Pure Storage



Hands On Lab Information

SQL Server 2022 and Pure Storage

1

Log into the lab.

2

Log into FlashArray™
web Interface.

3

Start up a database workload.

4

Working with
performance metrics.



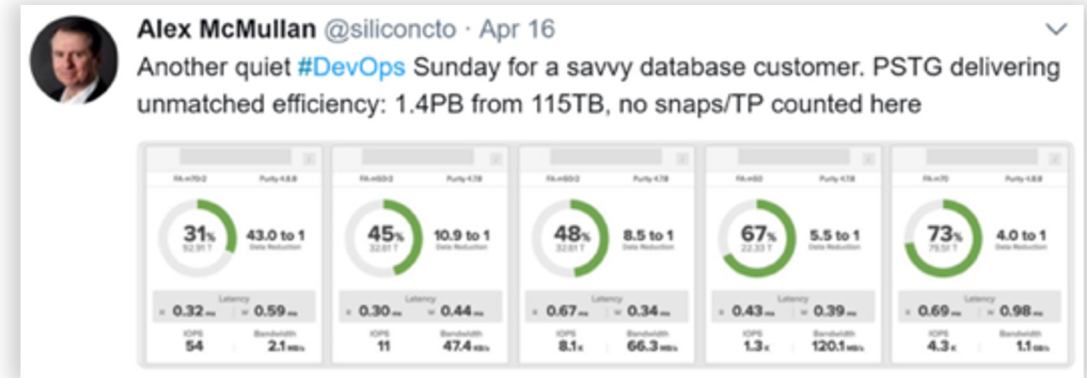


Storage-based Snapshots and SQL Server

Let's bend space and time when it comes to working with large databases

Data Reduction

- In-line deduplication and compression
- Performance over data reduction
- Deep background deduplication and compression
- Data reduction that's typically 2x better than retro-fit architecture-based arrays
- Deduplication down to 512 bytes block level
- Purity delivered by an engineering team that's constantly looking to improve data reduction
- But my database is encrypted with TDE!



Snapshots of Volumes

A Snapshot is a point in time representation of data on Volumes

Why leverage Snapshots for your databases:

- Instant data + ransomware protection + protecting audit files
- Dev/Test refreshes in seconds
- In-place application and database upgrades
- Intra-Instance ETL
- Offload database maintenance

Snapshots can significantly reduce compute, networking and storage, and overhead.

Snapshots consume very little space.

Clones will not impact performance of your source volumes.

Traditional restores are expensive for your company.

But my database is encrypted!

Choose your tooling – PowerShell, Python, REST, and more...



SQL Server 2022 – Database Engine Features

Protecting Large Databases

Crash consistent vs Application consistent snapshots

- Enables point in time recovery using VSS

TSQL Based Snapshot (Cross Platform Snapshot)

- Enables point in time recovery without VSS on FlashArray™
- Availability Group Replica Seeding from snapshot
 - Build Availability Groups faster
 - Get back into a high availability posture faster after a failure or failover



<https://www.nocentino.com/posts/2022-05-26-seed-ag-replica-from-snapshot/>



A Layered Approach to Backup and Recovery

- Replication of Snapshots to:
 - Another FlashArray™
 - FlashBlade™
 - NFS target
 - Public cloud like Amazon S3, Azure, GCP, or Cloud Block Store
- Replication of the reduced data (DR/Seeding)
- Ultra-fast backup and restore with FlashBlade™
- Use native or third-party backup tools
- Compliment your existing backup regime with snapshots of FlashArray™ Volumes



Hands on Lab Information

SQL Server 2022 and Pure Storage

1

In place restore a database from an array-based snapshot

2

Cloning a snapshot to a new volume and attaching the database

3

Cloning a database to another instance of SQL Server

4

Seed an Availability Group from an array-based snapshot (Optional)

Take a break after the lab – 15 minutes



SQL Server Object Integration: Backup and Restore

Backup and restore to s3 and performance tuning considerations

S3 – 101

Modern Object Storage and Data

- AWS Simple Storage Service (S3)
 - Storage service in the cloud
- API is open and available.
- Companies have built their own s3 compatible object storage platforms.
- Means you can get access to s3 anywhere.
 - Pure Storage FlashBlade
 - MinIO
 - Many others



S3 Object Integration – Backup and Restore

Modern Object Storage and Data

- Scale out rather than scale up
 - Single database high throughput
 - Concurrent backups
- Large environments
- Single Namespace
- Easy and native replication
- DBAs have one job
 - Get backups off the primary storage
 - Get them out of the data center as fast as possible...



<https://www.nocentino.com/posts/2022-06-06-backing-up-to-s3-storage-with-sqlserver/>



S3 Object Integration – Backup and Restore

Modern Object Storage and Data

Create a Bucket

Create a Credential

Backup Database

Restore Database

```
CREATE CREDENTIAL [s3://s3.example.com/sqlbackups]  
WITH IDENTITY = 'S3 Access Key', SECRET = ACCESSKEYID:SECRETKEY';
```

```
BACKUP DATABASE TestDB1  
TO URL = 's3://s3.example.com/sqlbackups/TestDB1.bak'  
WITH COMPRESSION, STATS = 10, FORMAT, INIT
```

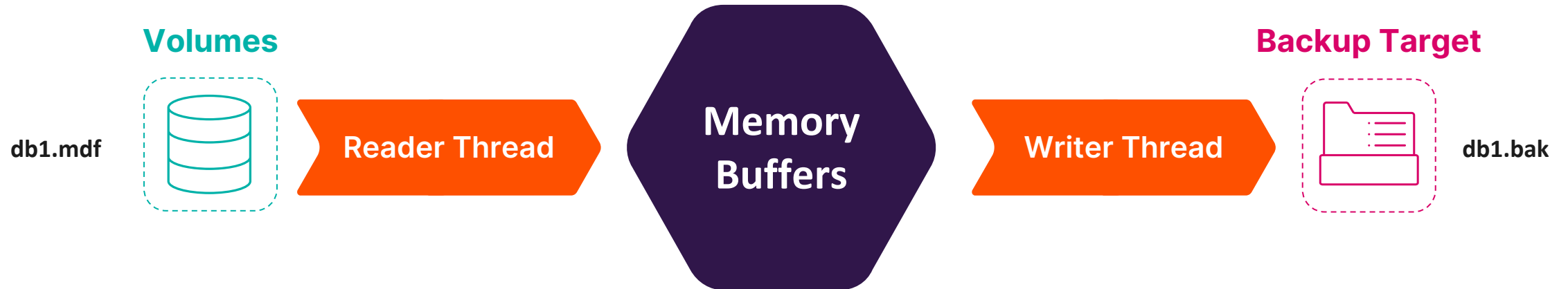
```
RESTORE DATABASE TestDB1  
FROM URL = 's3://s3.example.com/sqlbackups/TestDB1.bak'  
WITH STATS = 10
```



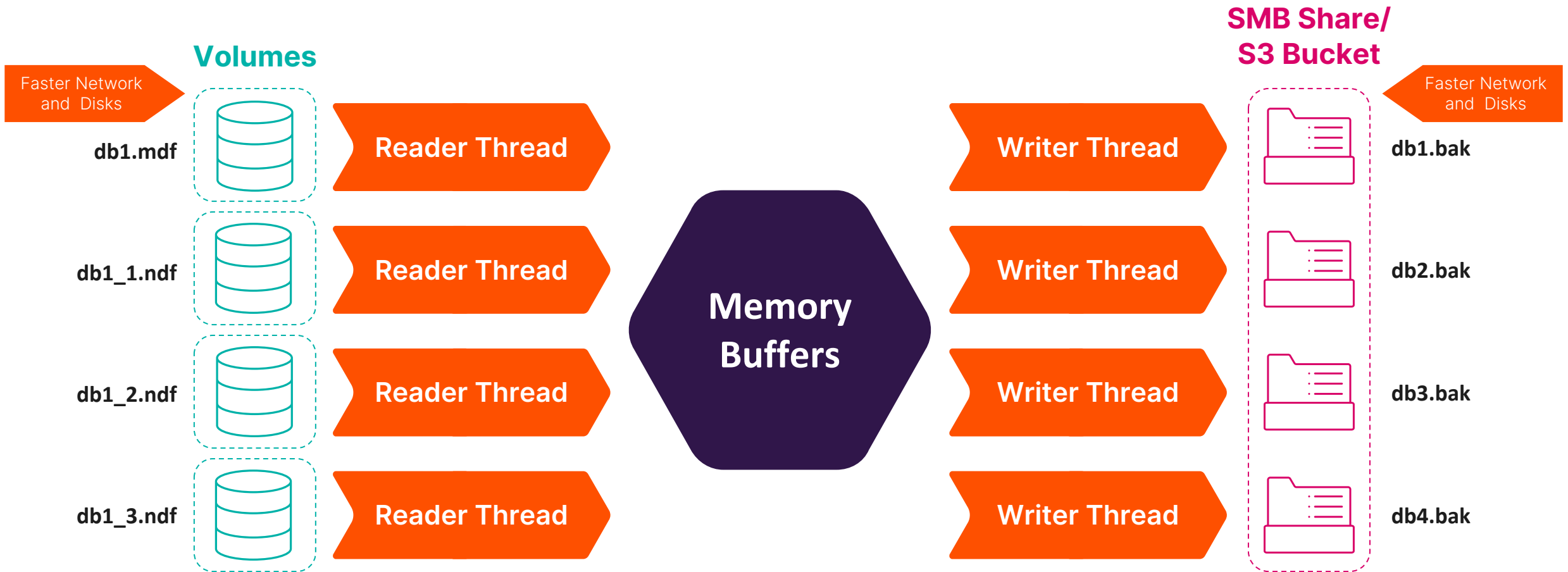
SQL Server Backup Architecture

Database Files are read by the **Reader Thread** into **Memory Backups Buffers**.

The **Writer Thread** reads from backup buffers and is written to the **Backup File**.



Performance Tuning Backups



How many database files and shares/buckets should you use?



S3 Object Integration – Backup and Restore

Modern Object Storage and Data

```
BACKUP DATABASE TestDB1
```

```
  TO URL = 's3://s3.example.com/sqlbackups/TestDB1.bak',  
     URL = 's3://s3.example.com/sqlbackups/TestDB2.bak',  
     URL = 's3://s3.example.com/sqlbackups/TestDB3.bak',  
     URL = 's3://s3.example.com/sqlbackups/TestDB4.bak',  
     URL = 's3://s3.example.com/sqlbackups/TestDB5.bak',  
     URL = 's3://s3.example.com/sqlbackups/TestDB6.bak'
```

```
WITH COMPRESSION, STATS = 10
```

- Each URL statement is a separate TCP stream.
- S3 has a max object size, so when working with backup files > 100GB, you must do a little tuning.



Hands-on Lab Information

SQL Server 2022 and Pure Storage

1

Backing up databases to S3 compatible object storage

2

Restoring databases from S3 compatible object storage

Take a break after the lab – 15 minutes



SQL Server Object Integration: Data Virtualization

Modern object storage data and SQL Server

S3 Object Integration – Data Virtualization

Modern Object Storage and Data

Why Data Virtualization?

- Access object storage directly from SQL Server engine
- Minimize overhead to get access to data
- Access data where it lives
- Backup restore / partitioning / index tuning not needed

Supported external file types

- Parquet/CSV/Delta

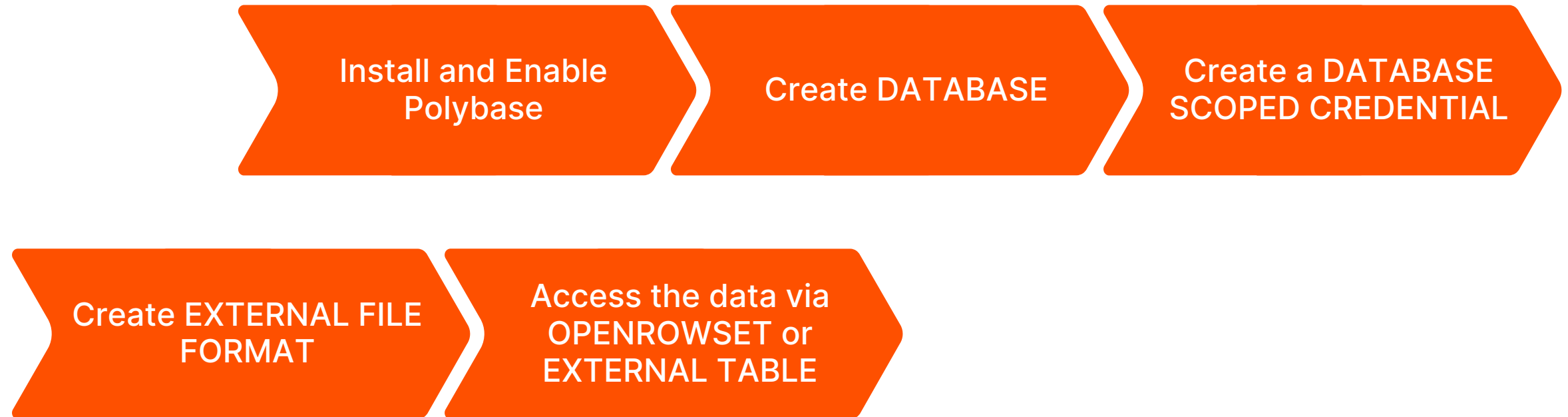
How to access external object data

- OPENROWSET
- EXTERNAL TABLE
- CREATE EXTERNAL TABLE AS SELECT



SQL Server 2022 – S3 Object Integration

Modern Object Storage and Data



Hands on Lab Information

SQL Server 2022 and Pure Storage

1

Query data on S3 compatible object storage with OPENROWSET

2

Query data on S3 compatible object storage with EXTERNAL TABLE

3

Wrap-up after the lab





PURESTORAGE®

Uncomplicate Data Storage, Forever