Gianluca Hotz

#SQLSAT895

PASS SQLSATURDAY PARMA

SQL Server Modernization

Who am I?

Gianluca Hotz | @glhotz | ghotz@ugiss.org

Independent consultant

- 20+ years on SQL Server (from 4.21 in 1996)
- Database modeling and development, sizing and administration, upgrade and migration, performance tuning

Community

- 20+ years Microsoft MVP SQL Server/Data Platform (since 1998)
- VMware Experts SQL Server
- Founder and president UGISS (PASS Chapter)
- Co-organizer DAMAG Meetup community

Organizers

PASS

ENGAGE IT SERVICES

daniela mavisi COMMUNICATION

UNIVERSITÀ DEGLI STUDI DI PARMA

GetLatestVersion.it

UGISS

Sponsors


vmware


IDERA
An Idera, Inc. Company


Bi FACTORY
FACTORY SOFTWARE
DATA
KNOWLEDGE
ADVISOR


WÜRTHPHOENIX


ENGAGE
IT SERVICES


Microsoft


tweppy


redgate


SSMSBoost
Add-in for SSMS


SolidQ
Think Big. Move Fast.


pcsnet


YOUR
IMPROVEMENT


SOFT
Jam
what's next.

Modernization
What, where & why

#SQLSAT895

Modernization?

Low Complexity

- Version upgrades (e.g. 2008 R2 -> 2017)
- Edition upgrades/downgrades (e.g. Standard -> Enterprise)

Medium Complexity

- Virtualization
- Consolidation
- Adding HA/DR

High Complexity

- Re-Platforming e.g. Oracle -> SQL Server or Azure SQL Database
- Re-Architecting e.g. SQL Server -> Azure SQL Datawarehouse/Data Lake

SQL Server flavors

On-Premises

- SQL Server (Database Engine, SSAS, SSRS, SSIS, DQS, MDS, MLS)
- Analytics Platform System (APS was SQL Server PDW)

IaaS

- SQL Server (Database Engine, SSAS, SSRS, SSIS, DQS, MDS, MLS)
  - VMs, Containers/Kubernetes

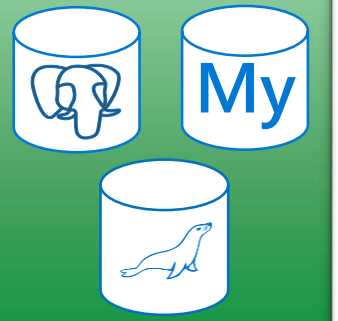
PaaS

- Azure SQL Database
- Azure Synapse Analytics (was Azure SQL Data Warehouse)
- Amazon RDS for SQL Server

Azure PaaS Operational Data Services

SQL-based

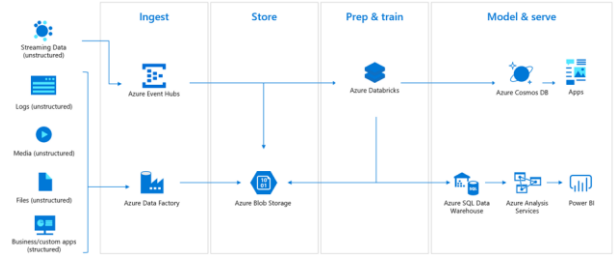
- Azure SQL Database
  - Traditional
  - Managed Instance
- Azure Database for PostgreSQL
- Azure Database for MySQL
- Azure Database for MariaDB



NoSQL-based

- Azure Cosmos DB
- Redis Cache

Azure Modern Datawarehouse



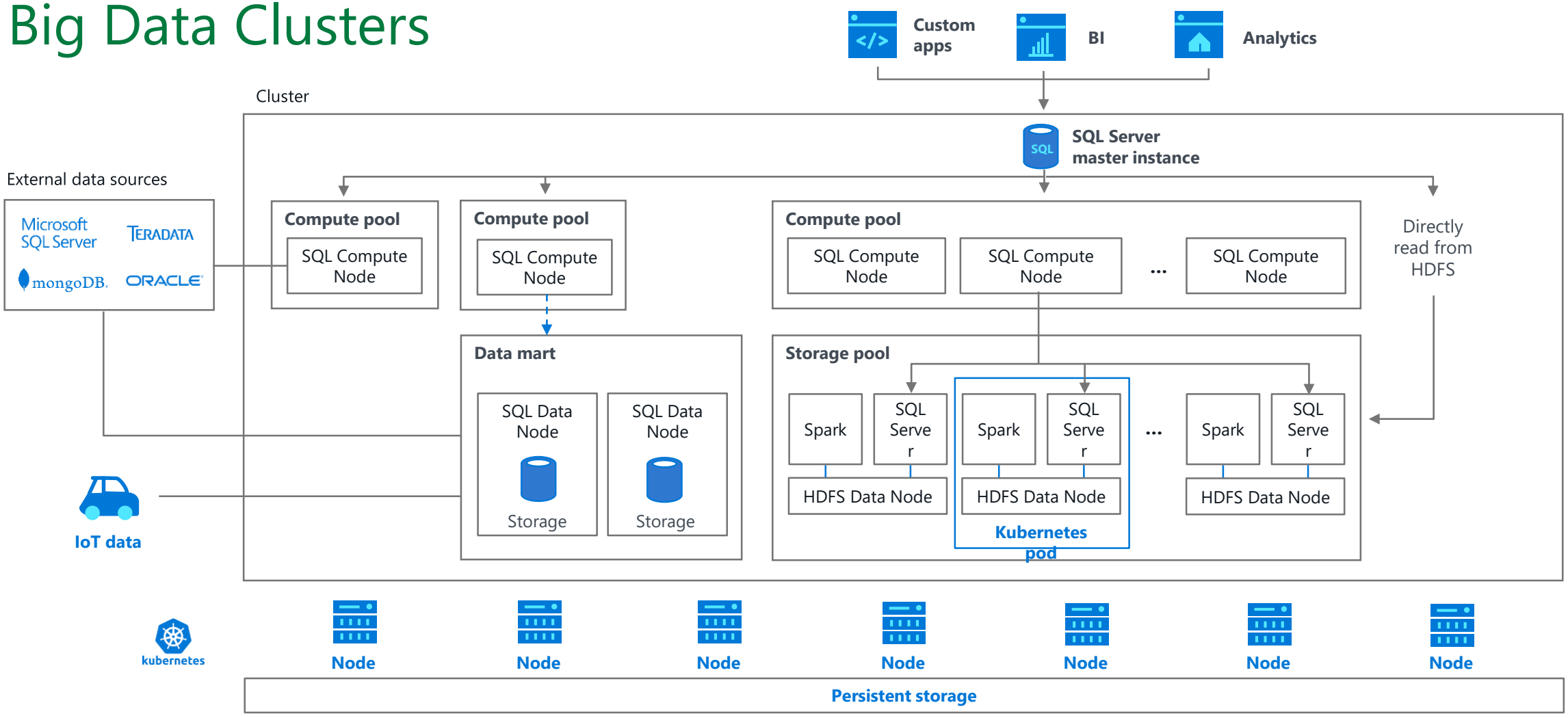
Ingestion
- Azure Event Hubs
- Azure Data Factory

Storage
- Azure Blob Storage
- Azure Data Lake Storage

Preparation
- Azure Databricks
- Azure Data Lake Analytics
- Azure HDInsight

Model/serve
- Azure Synapse Analytics (was Azure SQL Datawarehouse)
- Azure Analysis Services
- Power BI

SQL Server 2019 Big Data Clusters



#SQLSAT895

https://docs.microsoft.com/en-us/sql/big-data-cluster/big-data-cluster-overview

Changing platform

Can be easy

- Migrating databases to PaaS or containerized versions (but same RDBMS)
- or…

Hard: converting databases to other RDBMSs

- data must be physically migrated (different types, different representations,…)
- database code must be converted (constraints, triggers, functions, procedures,…)

Application code can be

- converted completely, to a new platform
- just adapted to run against the new RDBMS
- adapted to support multiple RDBMSs

Different kind of projects

- Migration
  - one time conversion
  - solution switch-over at a certain point
- Porting
  - side-by-side conversion & evolution
  - no switch-over, current solutions remain

Migration Projects

- Usually done by final customers (i.e. not ISVs)
- Driving factors
  - cost reduction
  - move away from vendor(s)
  - solution obsolescence
    - unreliability and/or supportability problems
- Application
  - may be converted to a new platform
  - may be adapted to run with new RDBMS

Porting Projects

- Usually done by ISVs
- Driving factors
  - get into new markets
    - cost reduction for customers (TCO)
    - more choice for customers
  - become less dependent from vendor(s)
- Application
  - may be ported to new platform
    - happens rarely, means re-write or 2 codebases
  - may be adapted to run with new RDBMS

Different kind of Databases to migrate/port

Simple schema

- all the logic is inside the application
- simple access methods (mostly by cursors)
- very limited use of SQL dialects
- database is used mostly for storage

Complex schema

- much of the logic is inside the database
- heavy usage of SQL dialects
- database used as a rule and inference engine

Origin of problems

Sometimes porting issues arise from

- misusage of functionalities
- bad decoupling of application/database code
- too much row oriented code in the database
- low technical skills of coders/architects

Never judge

- every application has it own story
- a lot of decisions may be compromises
- a lot of money was probably already invested

Application portability

- Code in the RDBMS
  - has been a best practice for years
  - still a very good solution for both
    - integrity
    - scalability
  - however… portability is an issue!
    - different SQL dialects
    - different type systems
    - different engine features

Changing Architecture

Can be much more harder!

More complex

- Low familiarity with non RBMS technologies
- Low familiarity with unstructured dta
- New implementation patterns
- ML & AI will stop to be buzzwords…

Edition upgrades/downgrades

- Upgrades
  - Motivations similar to version upgrades
  - Barriers similar to version upgrades
    - Complexity usually lower because setup covers the scenario
- Downgrades
  - Main motivation to save money
  - Barriers similar to version upgrades
    - Needs reinstallation, less impact for non production environment

Version Upgrades: motivations

Compliancy (e.g. end of support and/or security)

Better Performance

New functionalities

Winter ~~is coming~~ has come!

When will end of support happen?

SQL Server 2008 and 2008 R2 will no longer be supported starting on **July 9, 2019**.

	Current support level	End mainstream	End extended
SQL Server 2014	Currently supporting all versions	July 9, 2019	July 9, 2024
SQL Server 2012	SQL Server 2012 SP2+ is in extended support which includes security updates, paid support, and requires purchasing non-security hotfix support	July 11, 2017	July 12, 2022
⚠ **SQL Server 2008 and SQL Server 2008 R2**	SQL Server 2008 and 2008 R2 are in extended support which includes security updates, paid support, and requires purchasing non-security hotfix support	**July 8, 2014**	**July 9, 2019**
⚠ **SQL Server 2005**	SQL Server 2005 support ended on April 12, 2016	**April 12, 2011**	**April 12, 2016**

Learn more about the SQL Server support lifecycle: support.microsoft.com/lifecycle/

Extended Security Updates

Applies to

- SQL Server 2008 and 2008 R2 **Enterprise** and **Standard** editions
- Windows Server 2008 and 2008 R2 **Datacenter** and **Standard** editions

On-Premise

- Must have active SA (or equivalent e.g. EAS, SCE, EES)
- Price approximately 75% of latest version license cost

Azure

- 3 years for **free**

More info at https://www.microsoft.com/2008-eos

Editions: remember SQL Server 2016?

Feature	RTM				SP1			
	Standard	Web	Express	Local DB	Standard	Web	Express	Local DB
Row-level security	Yes	No	No	No	Yes	Yes	Yes	Yes
Dynamic Data Masking	Yes	No	No	No	Yes	Yes	Yes	Yes
Change data capture*	No	No	No	No	Yes	Yes	No*	No*
Database snapshot	No	No	No	No	Yes	Yes	Yes	Yes
Columnstore	No	No	No	No	Yes	Yes	Yes	Yes
Partitioning	No	No	No	No	Yes	Yes	Yes	Yes
Compression	No	No	No	No	Yes	Yes	Yes	Yes
In Memory OLTP	No	No	No	No	Yes	Yes	Yes	No**
Always Encrypted	No	No	No	No	Yes	Yes	Yes	Yes
PolyBase	No	No	No	No	Yes	Yes	Yes	No
Fine grained auditing	No	No	No	No	Yes	Yes	Yes	Yes
Multiple filestream containers	No	No	No	No	Yes	Yes	Yes	No**

SQL Server 2019 Standard Edition

Transparent Data Encryption!

Intelligent Query Processing

Accelerated Database Recovery

Many others

https://techcommunity.microsoft.com/t5/SQL-Server/SQL-Server-2019-Standard-Edition/ba-p/986121

https://docs.microsoft.com/sql/sql-server/editions-and-components-of-sql-server-version-15

IQP: Scalar UDF in-lining example

```
CREATE FUNCTION dbo.discount_price(@price DECIMAL(12,2), @discount DECIMAL(12,2))
RETURNS DECIMAL (12,2) AS BEGIN RETURN @price * (1 - @discount); END

SELECT L_SHIPDATE, O_SHIPPRIORITY
, SUM(dbo.discount_price(L_EXTENDEDPRICE, L_DISCOUNT))
FROM LINEITEM, ORDERS
WHERE O_ORDERKEY = L_ORDERKEY
GROUP BY L_SHIPDATE, O_SHIPPRIORITY
ORDER BY L_SHIPDATE
```

10GB CCI compressed TPC-H Schema, 2 x CPUs (12 cores), 96GB RAM, SSD storage

	Query without UDF	Query with UDF (no in-lining)	Query with UDF (in-lining)
Execution time	1.6 seconds	29 minutes 11 seconds	1.6 seconds

SQL Server 2019 Failover Server Benefit

New Software Assurance

- Enterprise/Standard Core/Server+CAL

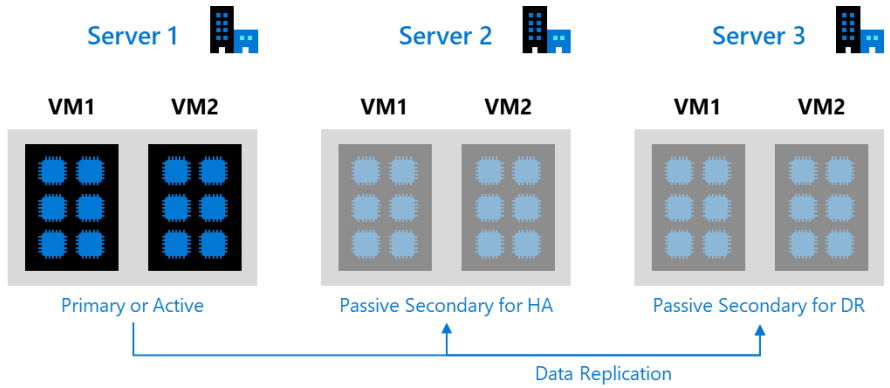
Not enough?

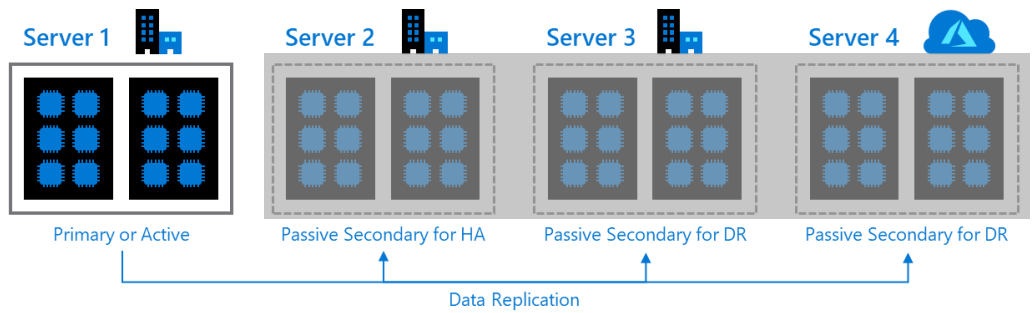
Allowed on passive

- Database consistency checks
- Log backups
- Full backups
- Monitoring resource usage data

More

- https://cloudblogs.microsoft.com/sqlserver/2019/10/30/new-high-availability-and-disaster-recovery-benefits-for-sql-server





Version Upgrades: barriers

- Licensing Costs
- Regressions risk
- Application changes costs
- Complexity
  - Testing changes (application or certification processes)
  - Planning for low downtime
- Lack of time/resources/competencies

Modernization Barriers

#SQLSAT895

Hardest problems

- Factors that require application changes
  - Breaking changes
  - Discontinued functionalities
- Plan affecting changes may cause performance regressions
  - New Cardinality Estimator
  - Hotfixes
  - New QP strategies/behavior (e.g. Intelligent Query Processing)
- Testing changes is hard!
  - Complete Workload coverage
  - Concurrent access

Database Compatibility Level

Sets behavior compatible with specified version of SQL Server

Version	Native Level	Supported Levels
SQL Server 2019	150	150, 140, 130, 120, 110, 100
SQL Server 2017	140	140, 130, 120, 110, 100
Azure SQL Database	**150**	150, 140, 130, 120, 110, 100
SQL Server 2016	130	130, 120, 110, 100
SQL Server 2014	120	120, 110, 100
SQL Server 2012	110	110, 100, 90
SQL Server 2008 R2	100	100, 90, 80
SQL Server 2008	100	100, 90, 80
SQL Server 2005	90	90, 80
SQL Server 2000	80	80

DB Compatibility Level based certification

Target for a database compatibility level

- Stop certifying for any given platform (On-Premises, Cloud, PaaS/IaaS)
- Stop certifying for a named SQL Server version

Microsoft Database Compatibility Level Protection

- Full functional protection (once assessment tools runs with no errors)
- Query Plan shape protection (on comparable hardware)
- Maintaining backward compatibility very important for SQL Server team…

DB Compatibility Level certification benefits

- Simplified certification across platforms
  - E.g. SQL Server On-Premises and Azure SQL Database Managed Instances
- Improved risk management
  - Decoupling application upgrade from database support upgrade cycles
- Allow customers to upgrade/migrate to new/different versions
  - Benefits from new features Without changing functionalities
  - Not everything is gated by Database Compatibility levels e.g.
    - Adaptive Joins: gated
    - Auto-tuning, new Query memory gateways, leveraging NUMA architecture, Columnstore indexes: non-gated

Functional Change Protection

Breaking Changes

- Behavior changes resulting in different outcomes!
- Not all protected under Database Compatibility

Protected changes examples

- Improved accuracy in compatibility level 130+
- Conversion behavior setting LANGUAGE context info
- Conversion behavior parsing dates

Non protected changes examples

- Column names changes in system objects

Demo

Functional Change Protection & Trace Flags

Some functional changes gated by Trace Flags

Example: truncation error 8152

- Gated by trace flag 460 in SQL 2017 CU12+
- Default in SQL 2019 with Compatibility Level 150
- https://blogs.msdn.microsoft.com/sql_server_team/string-or-binary-data-would-be-truncated-replacing-the-infamous-error-8152

Discontinued functionalities

Not protected by Compatibility Level!

- Removed from database engine
- Need to change queries using it

Examples

- **FASTFIRSTROW**
  - replaced by **OPTION (FAST n)** in SQL Server 2012
- **sp_dboption**
  - system stored procedure **removed** in SQL Server 2012

Query Plan shape protection

Plan-affecting changes

- Query Optimizer fixes (Trace Flag 4199)
- Changes to Cardinality Estimator

These will very likely cause regressions!

- Especially Cardinality Estimator
- Many queries better performance, other worse
- Are you sure you understood how fixes works?
- What about fixes for incorrect result causing regressions?

Query Optimizer Fixes with TF 4199

SQL Server 2016+ QO fixes from previous versions **enabled by default** under the **latest** level

- previously you had to enable them but you enabled also current version, post-RTM, fixes
- **zero reported errors in support cases after changing the default**

Fully Documented

- https://support.microsoft.com/en-us/help/974006/sql-server-query-optimizer-hotfix-trace-flag-4199-servicing-model

Compatibility Level	TF 4199	QO changes from previous level	QO changes for current version post-RTM
100-120	**Off**	**Disabled**	Disabled
	On	Enabled	Enabled
130	**Off**	**Enabled**	Disabled
	On	Enabled	Enabled
140	**Off**	**Enabled**	Disabled
	On	Enabled	Enabled
150/Azure	**Off**	**Enabled**	Disabled
	On	Enabled	Enabled

Query Optimizer fixes further control

SQL Server 2016+ database-level granularity

ALTER DATABASE SCOPED CONFIGURATION SET **QUERY_OPTIMIZER_HOTFIXES** = ON|OFF

SQL Server 2016 SP1+ query-level granularity

OPTION(USE HINT('**ENABLE_QUERY_OPTIMIZER_HOTFIXES**'))

Cardinality Estimator problems

Essential for query plan generation

Was largely based on SQL Server 7.0

Redesigned in SQL Server 2014+

- Ascending data estimation, correlation assumptions, join containment, fixed estimation values (e.g. MTVF)
- Will most probably cause regressions in performance!

Controlling Cardinality Estimation

SQL Server 2014

- Bound to Database Compatibility Level 😬
- Trace Flags (global, session and query levels)
  - Trace flag 2312 reverts to CE version 70
  - Trace flag 9481 use CE version 120

SQL Server 2016+ database-level configuration

- ALTER DATABASE SCOPED CONFIGURATION SET **LEGACY_CARDINALITY_ESTIMATION** = ON|OFF

SQL Server 2016 SP1+ query-level configuration

- OPTION(USE HINT('**FORCE_LEGACY_CARDINALITY_ESTIMATION**'))
- OPTION(USE HINT('**FORCE_DEFAULT_CARDINALITY_ESTIMATION**'))

Query Optimizer Hints

SQL Server 2016 SP1+

Control QO behavior

Mainly to replace trace flags

https://blogs.msdn.microsoft.com/sql_server_team/developers-choice-use-hint-query-hints

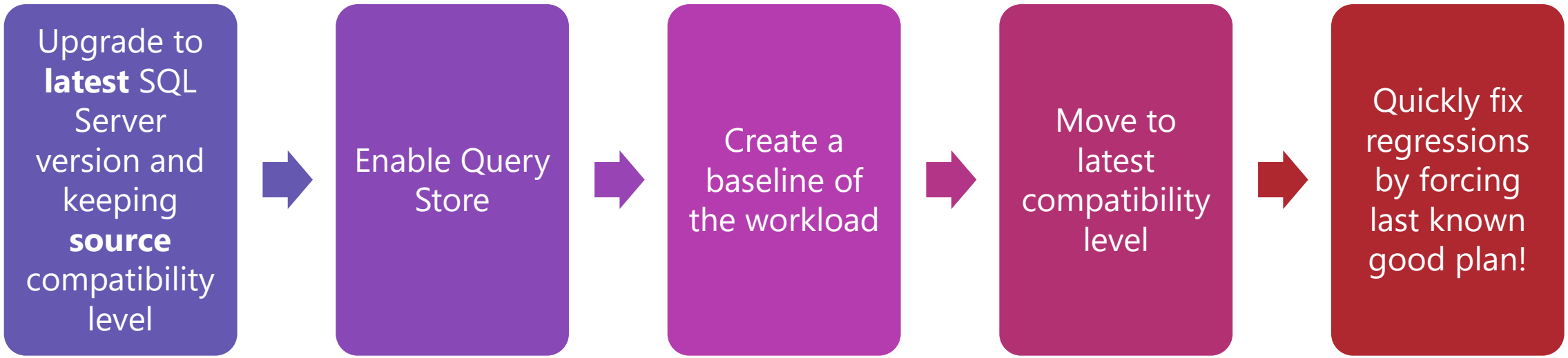
USE HINT	Trace Flag	DB option	Ver.
DISABLE_OPTIMIZED_NESTED_LOOP	2340		2016 SP1+
FORCE_LEGACY_CARDINALITY_ESTIMATION	9841	Yes	2016 SP1+
ENABLE_QUERY_OPTIMIZER_HOTFIXES	4199	Yes	2016 SP1+
DISABLE_PARAMETER_SNIFFING	4136	Yes	2016 SP1+
ASSUME_MIN_SELECTIVITY_FOR_FILTER_ESTIMATES	4137 old CE 9471 new CE		2016 SP1+
DISABLE_OPTIMIZER_ROWGOAL	4138		2016 SP1+
ENABLE_HIST_AMENDMENT_FOR_ASC_KEYS	4139		2016 SP1+
ASSUME_JOIN_PREDICATE_DEPENDS_ON_FILTERS	9476 new CE		2016 SP1+
FORCE_DEFAULT_CARDINALITY_ESTIMATION	2312		2016 SP1+
DISALLOW_BATCH_MODE			2017+
DISABLE_INTERLEAVED_EXECUTION_TVF			2017+
DISABLE_BATCH_MODE_MEMORY_GRANT_FEEDBACK			2017+
DISABLE_BATCH_MODE_ADAPTIVE_JOINS			2017+
DISABLE_ROW_MODE_MEMORY_GRANT_FEEDBACK			2019+
DISABLE_DEFERRED_COMPILATION_TV			2019+
DISABLE_TSQL_SCALAR_UDF_INLINING			2019+
QUERY_OPTIMIZER_COMPATIBILITY_LEVEL_n			2017 CU10+
QUERY_PLAN_PROFILE			2016 SP2 CU3+ 2017 CU11+

Query Optimizer Compatibility Level

SQL Server 2017 CU10 introduces query-level granularity

- OPTION(USE HINT('**QUERY_OPTIMIZER_COMPATIBILITY_LEVEL_n**'))
- Includes QO fixes enabled by default up to that level
- TF 4199 enabled globally will still enable all fixes
- https://blogs.msdn.microsoft.com/sql_server_team/developers-choice-hinting-query-execution-model

Microsoft recommended upgrade Plan



Modernization How

#SQLSAT895

Upgrade Strategies

Side-by-side

Allow OS Upgrade

Easier testing

Easier rollback strategy

Less downtime

In-Place

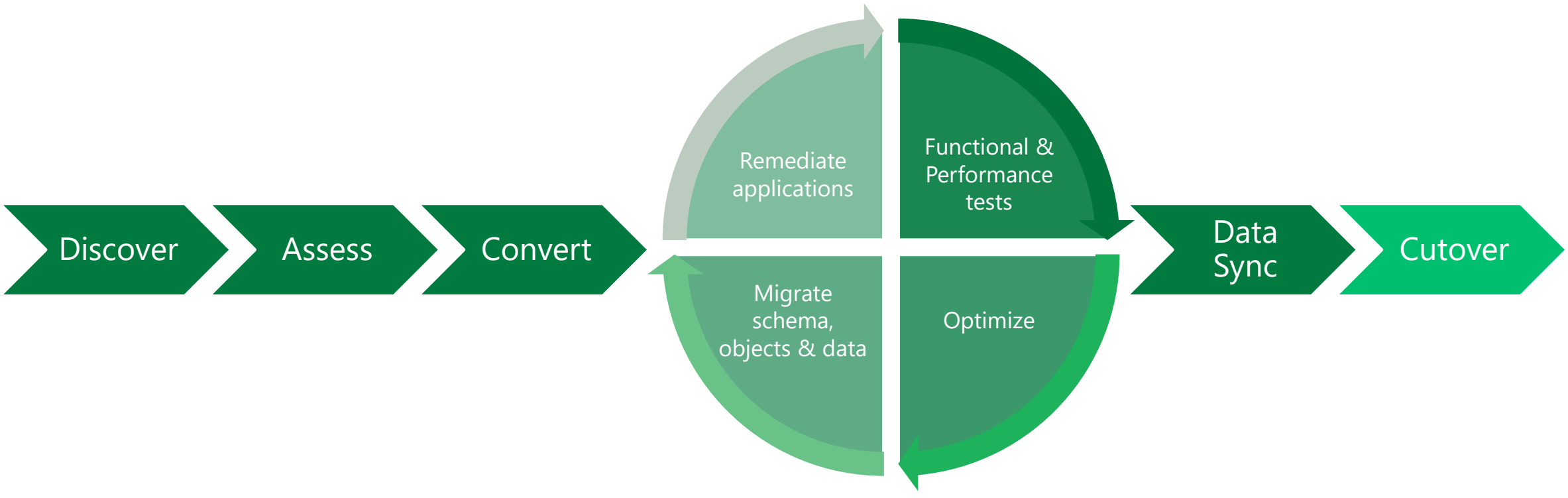
No additional Hardware

No data migration

Side-by-side Upgrade Service Availability

Synchronization Strategy	Notes
Log Shipping	Cutover typically in minutes
Replication	Cutover potentially in seconds
Backup/Restore	From minutes to hours (depends on strategy e.g. usage of differential and/or log backups)
Filesystem/SAN Copy	Dependent on technology
Database Mirroring/Availability Groups	Cutover typically in seconds

Upgrade process to minimize risks



Discover Phase

- Which SQL Server versions do I have?
- Which Editions am I running?
- Which SQL Server components are installed?
- How many cores are on each server?
- How many databases are in each instance?
- What are the sizes of all my databases?
- What are the settings for each instance and database?

Assessment Tools

Microsoft Assessment and Planning (MAP) Toolkit

SQL PowerDoc

- PowerShell Script gather data from WMI in XML
- Windows and SQL Server rich inventory in Excel
- Best Practices Reports in Excel
- Consolidates multiple server/instances
- Official https://github.com/kendalvandyke/sqlpowerdoc
- Newer https://github.com/bryan5989/SQLPowerDoc

Quest Enterprise Reporter for SQL Server

- Based on previously free Discovery Wizard for SQL Server
- Part of a suite for compliancy covering several other products (e.g. AD)

Microsoft Upgrade/Migration Tools

~~Upgrade Advisor~~

Data Migration Assistant (DMA)

Data Experimentation Assistant (DEA)

Data Migration Service (DMS)

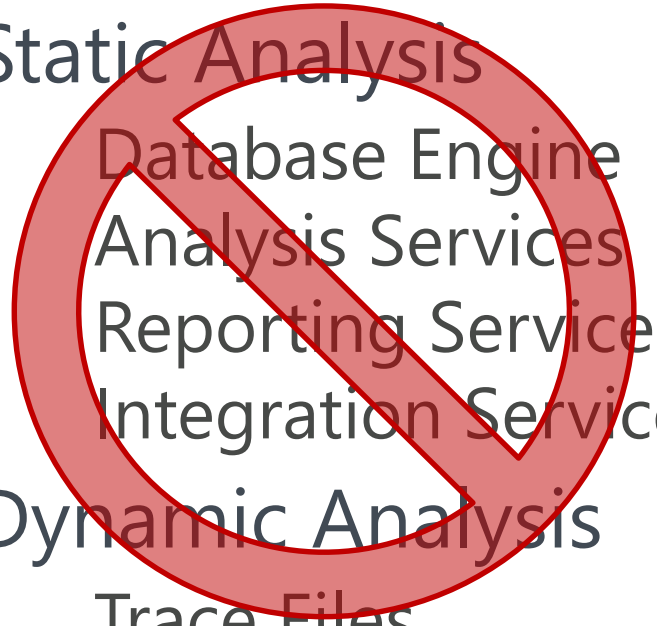
SQL Server Migration Assistant (SSMA)

SQL Server Data Tools for Visual Studio (SSDT)

SQL Server Management Studio (SSMS)

- "Export Data Tier Application"
- "Deploy Database to Microsoft Azure SQL Database" Wizard

Upgrade Advisor



- Static Analysis
  - Database Engine
  - Analysis Services
  - Reporting Services
  - Integration Services
- Dynamic Analysis
  - Trace Files
- Obsolete!
  - Use only if source database non supported
  - You can still find it in SQL Server Feature Packs (2014 latest)

Data Migration Assistant

On-premises instances assessment
- Blocking issues
- Partially supported or unsupported features

Discover issues that may affect compatibility
- Breaking Changes, Behavior Changes, Deprecated Features

Discover new features that may benefit after upgrading
- Performance, security, Storage

Azure SQL Database SKU Recommendations
- Performance counters analysis to suggest SKU Azure SQL Database

Performs the migration

Data Migration Assistant V5

Added SQL 2019 destinations (Windows & Linux)

Save and load assessment!!!

Assess SQL queries from external applications

- e.g. queries in C# code
- Access Migration Toolkit to generate JSON report

Assess SSIS in SSISDB and package store

More info

- https://docs.microsoft.com/en-us/sql/dma/dma-whatsnew

Versions supported by DMA

Source	Destination
SQL Server 2005	SQL Server 2012
SQL Server 2008	SQL Server 2014
SQL Server 2008 R2	SQL Server 2016
SQL Server 2012	SQL Server 2017 on Windows and Linux
SQL Server 2014	SQL Server 2019 on Windows and Linux
SQL Server 2016	Azure SQL Database
SQL Server 2017 on Windows	Azure SQL Database Managed Instance

Bulk Analysis

Repository instances/databases to assess in SQL/CSV

Automation with PowerShell

Import in reporting database

Reports with PowerBI

DMA Problems

- ~~Cannot save sessions!~~
  - ~~Only Assessment data in JSON format~~
- Azure SQL Database SKU Recommendations
  - Command Line Interface only
  - Currently not working…
- Bulk Analysis (still in V5?)
  - Cumbersome process
  - Errors
    - dmaDataCollector crashes with pubs, dmacmd.exe is a closed project
    - LoadWarehouse.sql needs to be fixed

Data Migration Assistant Resources

Docs

https://docs.microsoft.com/en-us/sql/dma/dma-overview

https://docs.microsoft.com/en-us/sql/dma/dma-consolidatereports

Blog

https://techcommunity.microsoft.com/t5/Microsoft-Data-Migration/bg-p/MicrosoftDataMigration

Database Experimentation Assistant

A/B testing solution for SQL Server

- Workload comparison
- e.g. upgrades, new indexes, etc.

Capture

- SQL Trace
- Extended Events (V2.6+)

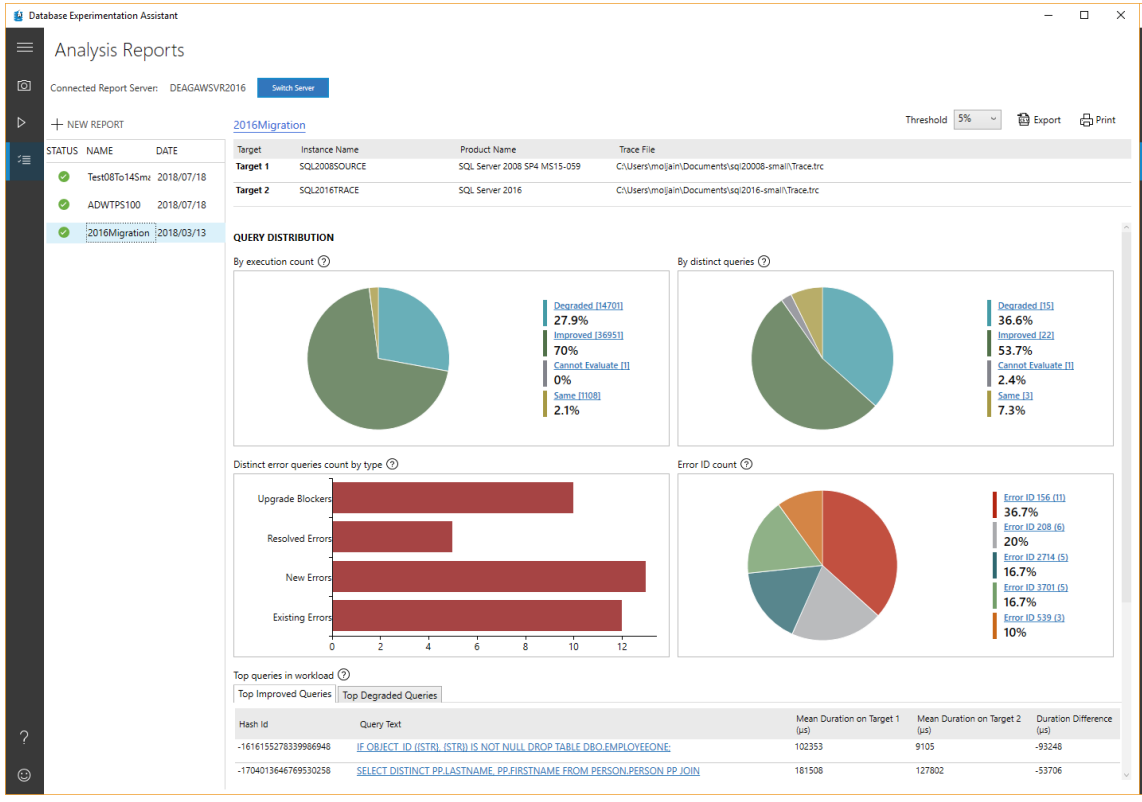
Replay

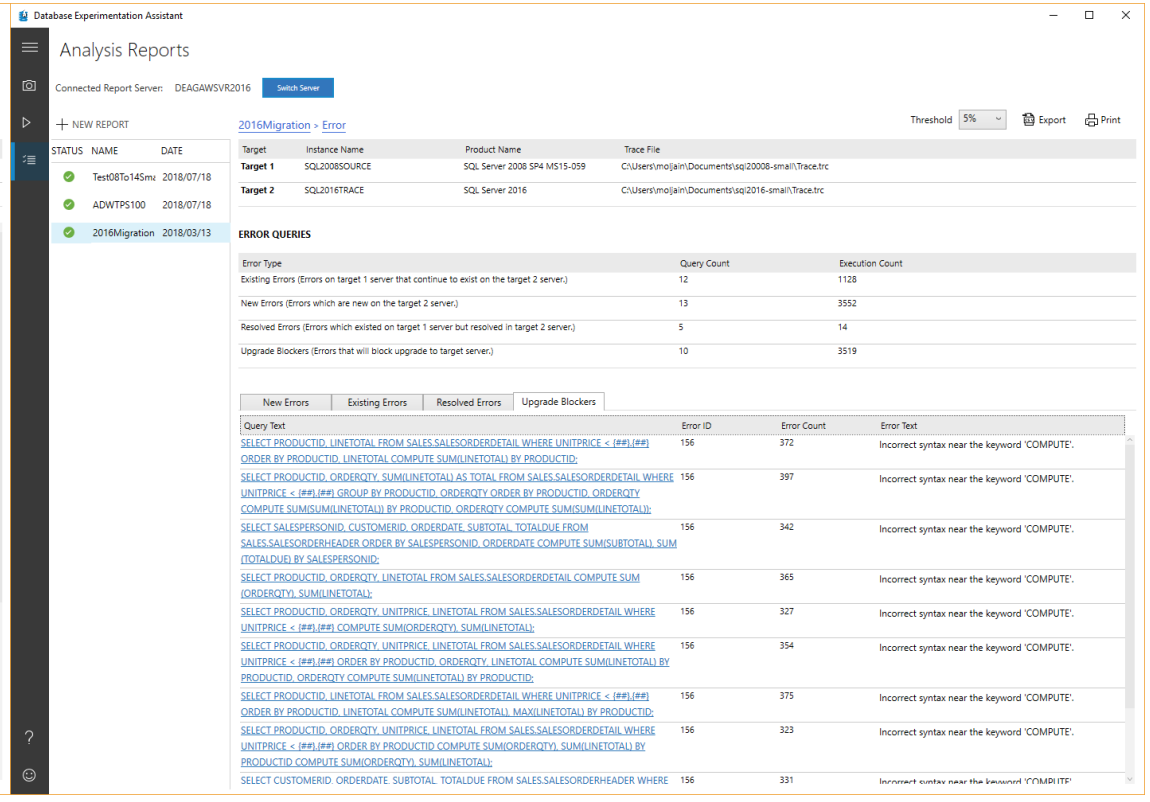
- DReplay integration

Analysis

- Performance improvements and regressions

DEA Analysis Example





Versions supported by DEA

Source

- SQL Server 2005+

Target

- SQL Server 2005+

Analysis

- SQL Server 2008+

Data Migration Service Resources

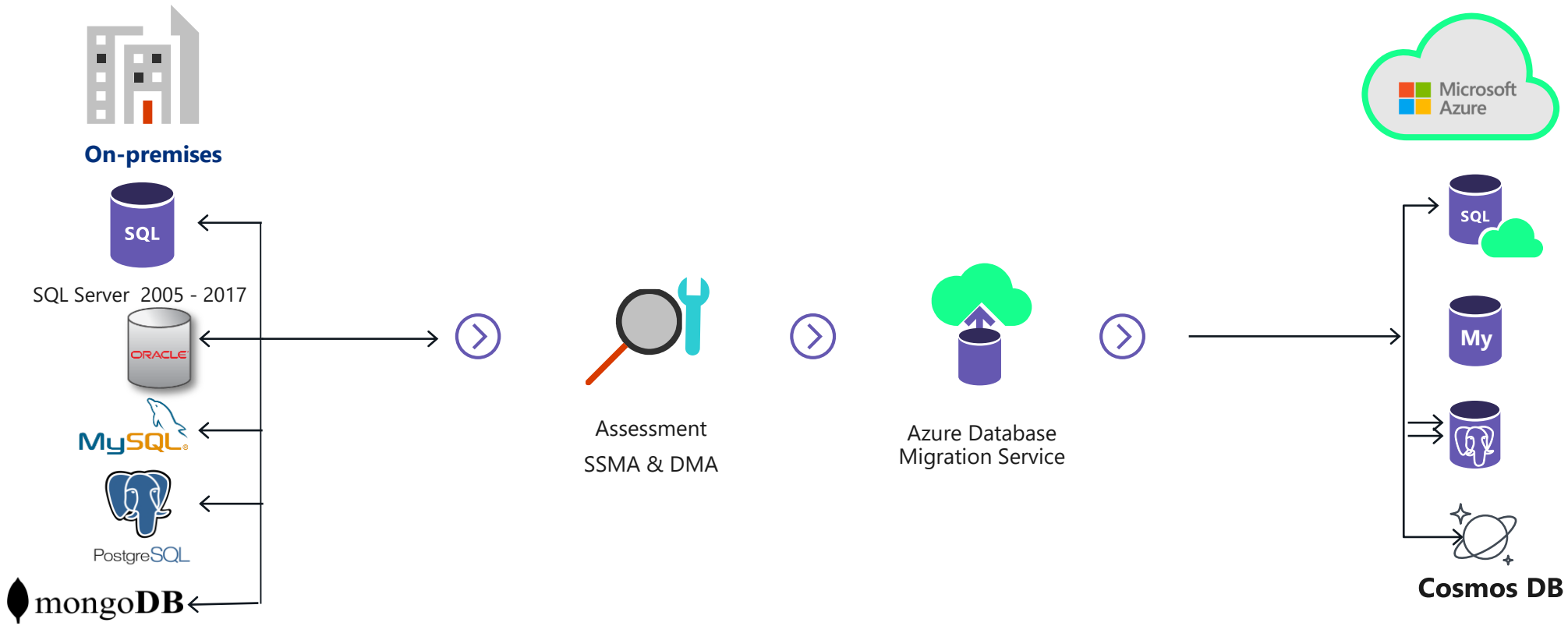
Docs

https://docs.microsoft.com/en-us/sql/dea/database-experimentation-assistant-overview

Blog

https://techcommunity.microsoft.com/t5/Microsoft-Data-Migration/bg-p/MicrosoftDataMigration

Azure Database Migration Service



DMS Migration Scenarios

SQL Server -> Azure SQL Database Offline

SQL Server -> Azure SQL Database Online

SQL Server -> Azure SQL Database Managed Instance Offline

SQL Server -> Azure SQL Database Managed Instance Online

MySQL -> Azure DB for MySQL Online

PostgreSQL -> Azure DB for PostgreSQL Online

Data Migration Service Resources

Docs

https://docs.microsoft.com/en-us/azure/dms/dms-overview

https://docs.microsoft.com/en-us/azure/dms/tutorial-sql-server-to-azure-sql

Blog

https://techcommunity.microsoft.com/t5/Microsoft-Data-Migration/bg-p/MicrosoftDataMigration

SQL Server Migration Assistant

Migration effort assessment

Code conversion

Data migration

Testing

Deployment

Platforms supported by SSMA

Source

Microsoft Access

DB2

MySQL

Oracle

SAP ASE (Sybase ASE)

Destination

SQL Server 2008

SQL Server 2008 R2

SQL Server 2012

SQL Server 2014

SQL Server 2016

SQL Server 2017 on Windows and Linux

SQL Server 2019 on Windows and Linux

Azure SQL Database

Azure SQL Database Managed Instance

Azure SQL Data Warehouse (Oracle only)

SSMA for Oracle: how it works (101)

- Creates two “helper” databases
  - sysdb
  - ssmatesterdb
- Stores metadata in the file system
  - data stored in xml compressed format
    - can rename to .gzip and uncompress it
  - shred xml into lot of files for caching
    - reports directory
      - can be safely deleted and re-generated
      - metadata reside in the original .mb files

SSMA for Oracle: sysdb helper database

- Provides
  - code infrastructure to emulate features
    - system functions
    - schema features (e.g. sequences)
    - engine features (e.g. autonomous transactions)
  - session bound data storage
  - emulated metadata storage (e.g. packages)
- Maintenance
  - protect metadata
  - performance related depends on workload

SSMA for Oracle: does it really work?

Yes, but don't expect miracles!

- Works well with simple schemas…
- Usually won't do all the work by itself

Very useful for

- assessing efforts
- assessing problems
- doing prototypes
- as a learning tool
- to translate/emulate some functionalities

Testing

Nothing can replace good testing!

- Hard to have tests with good coverage of the whole workload
- Sometimes is more cost effective to run fewer tests and fix the problems later…

SQL Server DReplay infrastructure

- Often impossible to capture and replay even a fraction of the workload

Workload Tools

Very good for side-by-side, real-time workload replay

- Captures XE batch completed and replay commands on-the-fly

Captures execution telemetry (cpu, read, writes, duration)

Reports to find regressions

Developed by a well-known MVP ☺

https://github.com/spaghettidba/WorkloadTools

DBA Tools

Lot of PowerShell cmdlets to make the DBA life easier

Many cmdlets target specifically migration/upgrade scenarios

- E.g copy logins with synch option

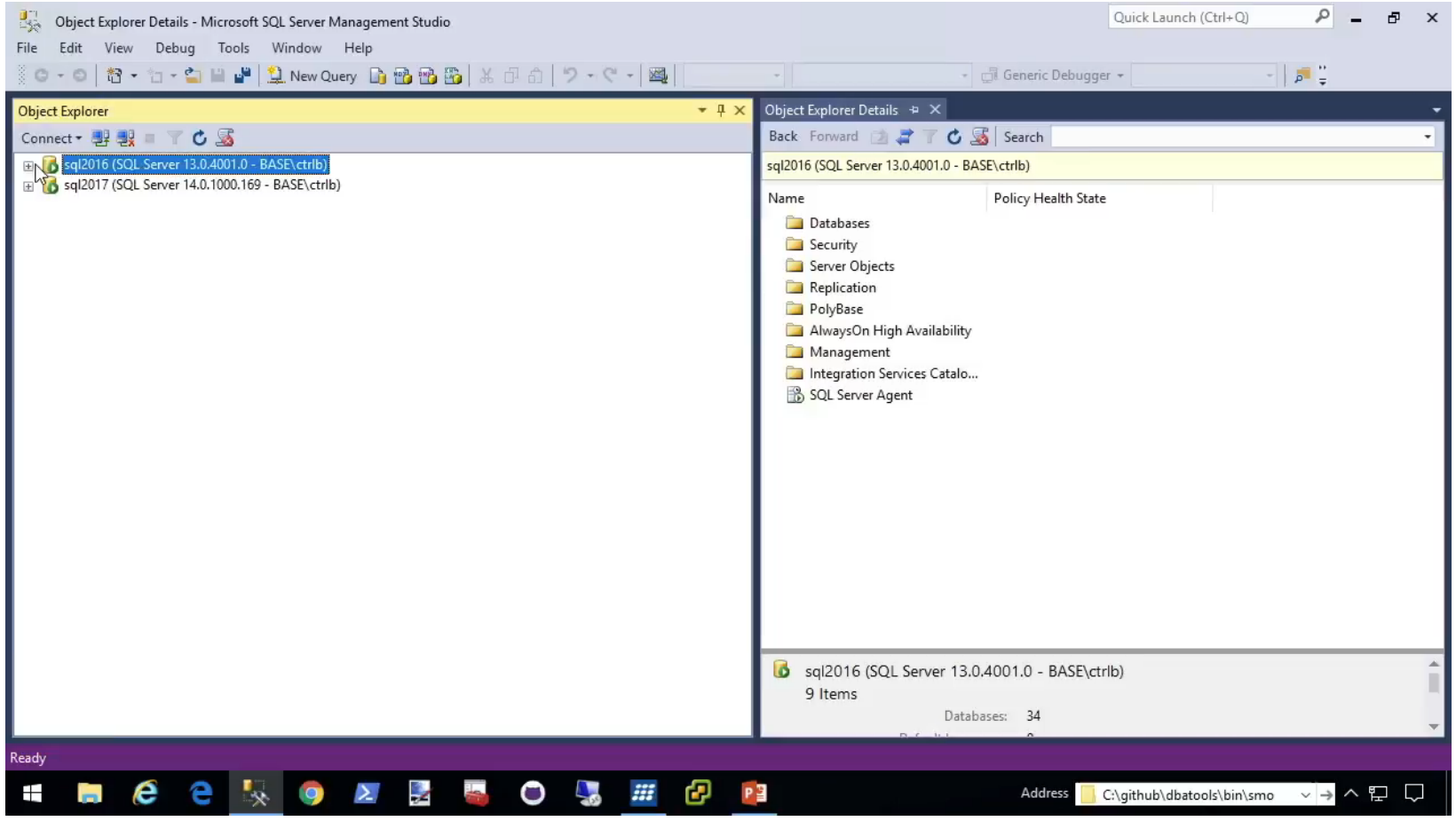
One stop cmdlet migrating an

- All instance-level assets (e.g. credentials, linked servers, etc.)
- Via backup/restore automation (just specify network share)

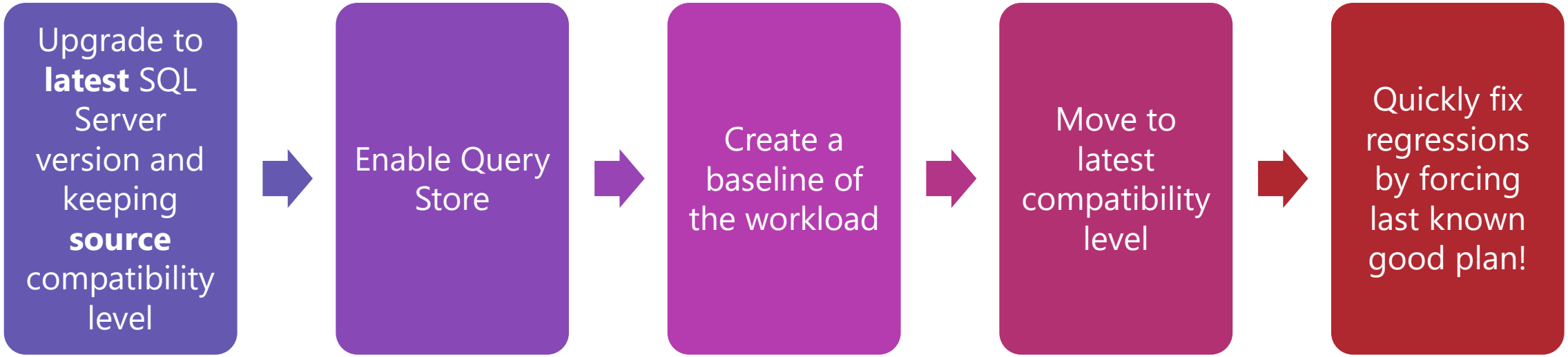
https://dbatools.io

Start-DbaMigration

https://youtu.be/hg8tovMRX2k



After the upgrade: remember the plan



Automatic Tuning

Automatic plan correction

- Automatically force last good plan when regression detected
- Minimum 10 CPU seconds improvement
- **sys.dm_db_tuning_recommendations** for manual corrections

SQL Server 2017 and Azure SQL Database

https://docs.microsoft.com/en-us/sql/relational-databases/automatic-tuning/automatic-tuning

Query Tuning Assistant

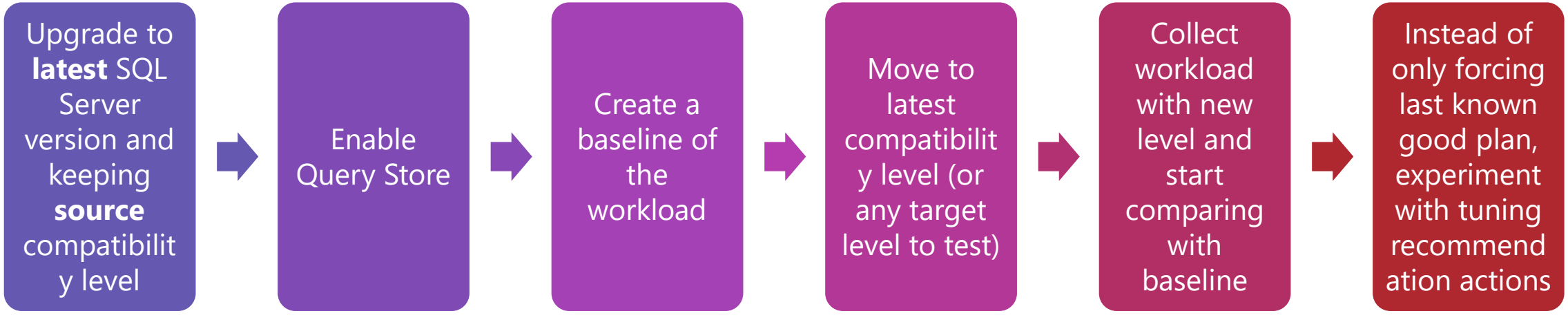
Well-defined workflow targeting patterns for query regressions

- Independence vs. Correlation
- Simple Containment vs. Base Containment
- Multi-statement table-valued function (MSTVF) fixed cardinality guess

SQL Server Management Studio Wizard-Based Tool

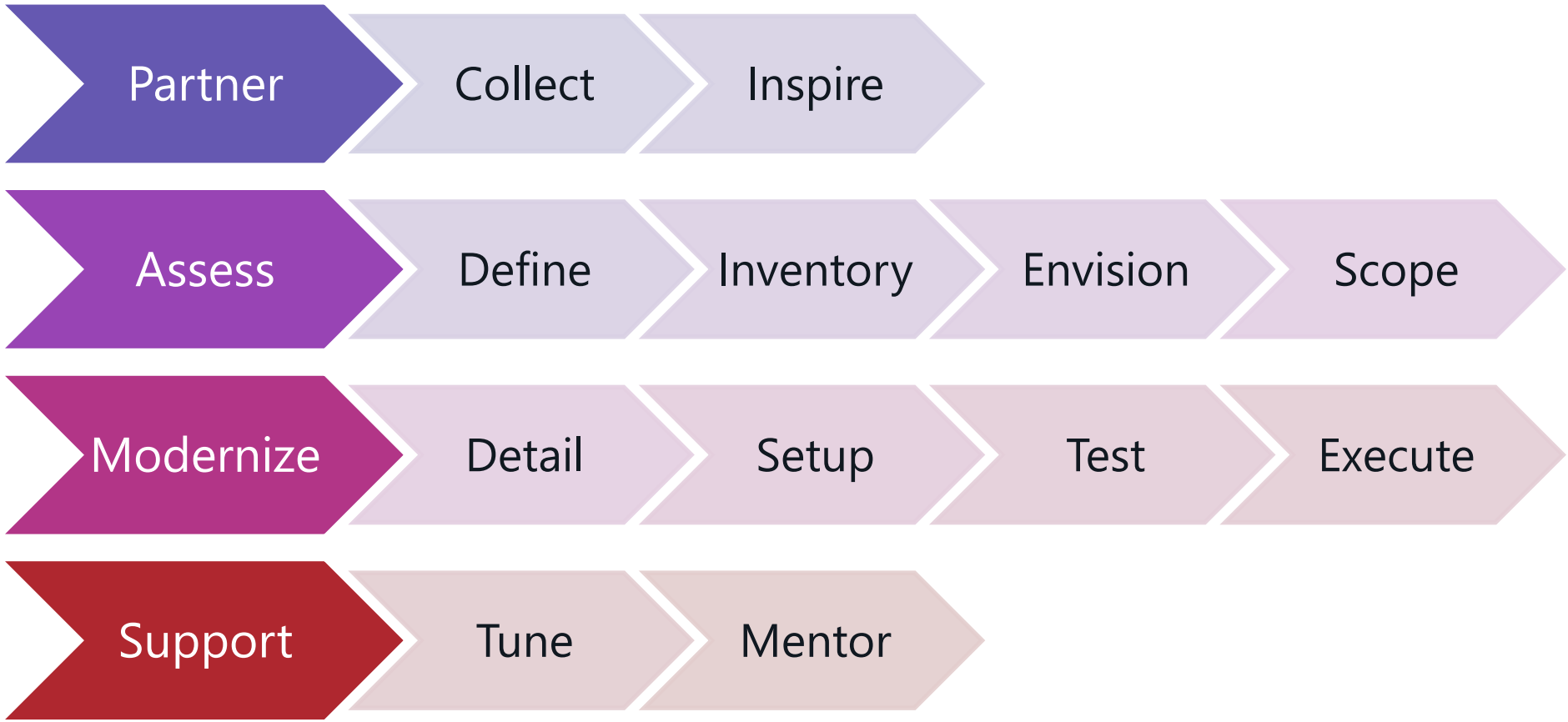
Can be automated via PowerShell

Query Tuning Assistant plan



https://docs.microsoft.com/en-us/sql/relational-databases/performance/upgrade-dbcompat-using-qta

Modernization Stream Phases in a SQL Project



Assessment phase

Define
- Finalize assessment's scope
- Finalize questionnaires to be sent to project representatives
- Activities kick-off

Inventory
- Detailed inventory of Windows Server & SQL Servers
- Static analysis of databases & SQL code
- Project representatives interviews (questionnaires)

Envision
- Consolidate & analyze inventory data
- Review configuration, performance, reliability and security best practices
- Prepare modernization scenarios and macro action-plan

Scope
- Discuss deliverables and scenarios with customer
- Finalize modernization scenarios
- Prepare next phases offering (Modernize & Support)

Assessment phase: reducing uncertainty

- Better planning/flexibility
  - Size better understood results in reduced expectations and scope
  - Allows to run several Modernization phases at different times
- Leaves the customer with something tangible
  - Detailed assets inventory
  - Performance/Security/Reliability automatic assessment
  - Static analysis of databases code
  - Possibility to rescope/change scenarios
  - Enough data to independently run the project

On-Premises to Azure PaaS/IaaS migrations

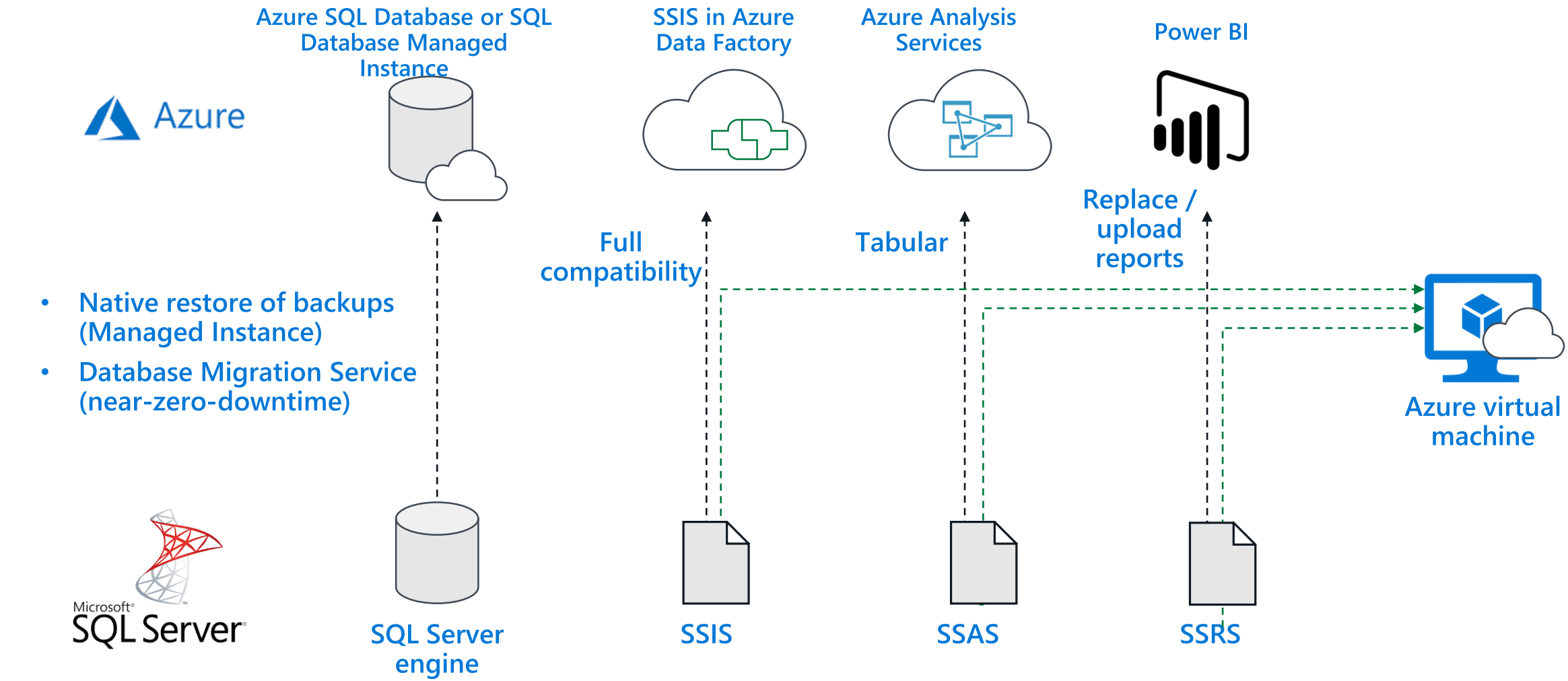
IaaS

- SQL Server on Azure VMs
- SQL Server on Azure Containers
- SQL Server on Azure Kubernetes

PaaS

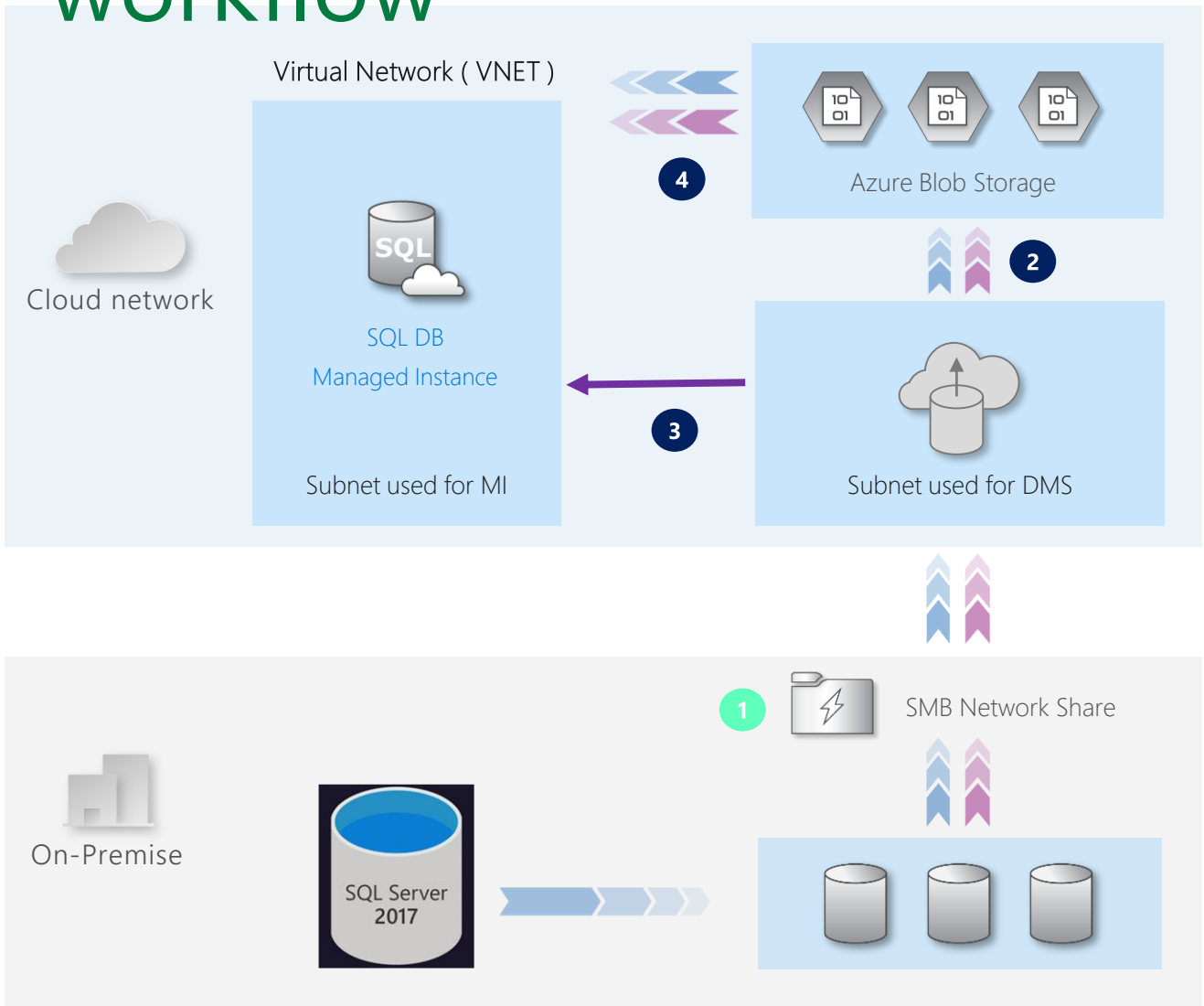
- Azure SQL Database
- Azure SQL Database Managed Instance

On-Premises to SQL Server PaaS migrations



- Native restore of backups (Managed Instance)
- Database Migration Service (near-zero-downtime)

On-Premise to Managed Instance online workflow



Legend

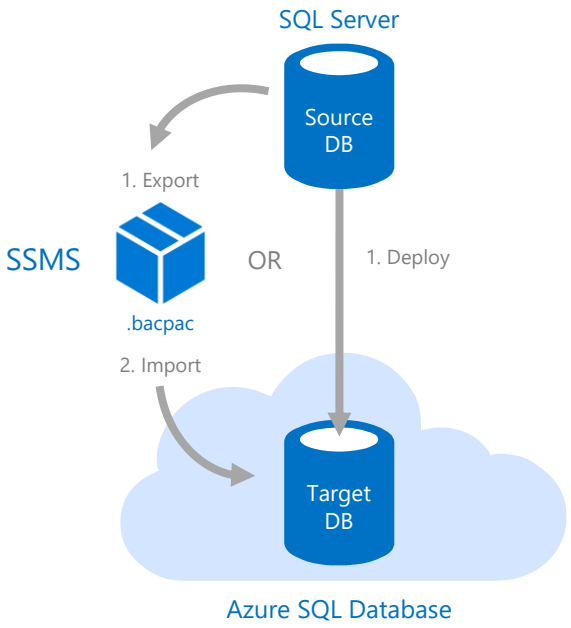
- Full Database backup files
- Transaction log backup files
- Site to site connectivity (VPN or ExpressRoute)

1. Provide existing backups in network share
2. DMS upload backup files to Azure storage
3. DMS initiate the migration to Azure SQL MI
4. Full backup restored and Transaction log backups continuously applied until cutover

Provide Tail-Log backup, initiate cutover in DMS and change the application connection strings

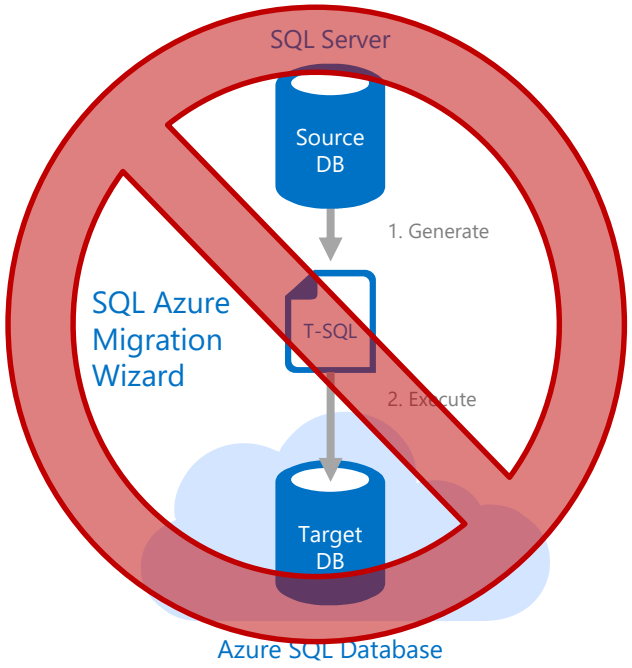
Other on-premises to PaaS strategies

Method 1



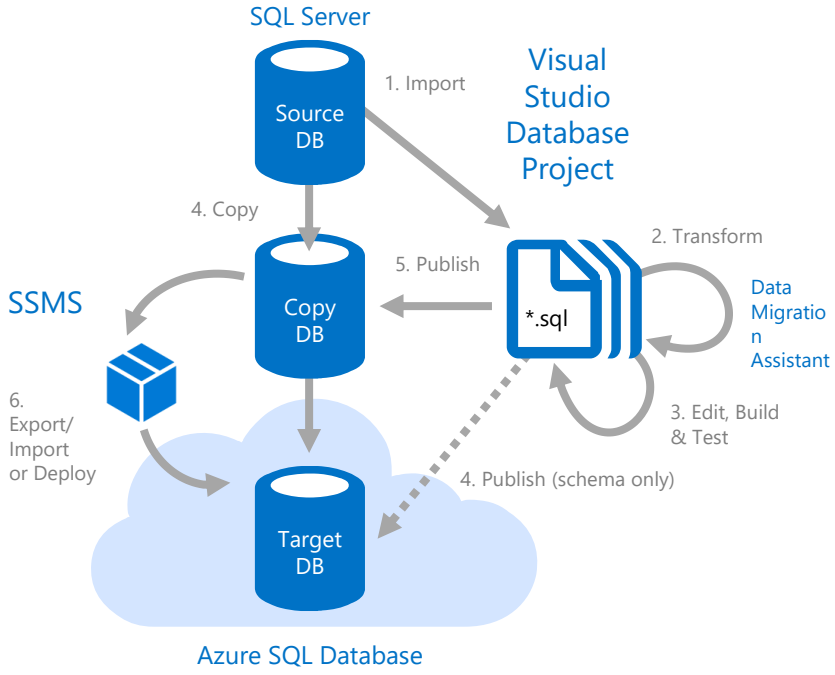
Compatible database: manual copy or deploy wizard in SSMS

Method 2



Database almost compatible: ~~SQL Azure Migration Wizard~~ DMA and SSMS

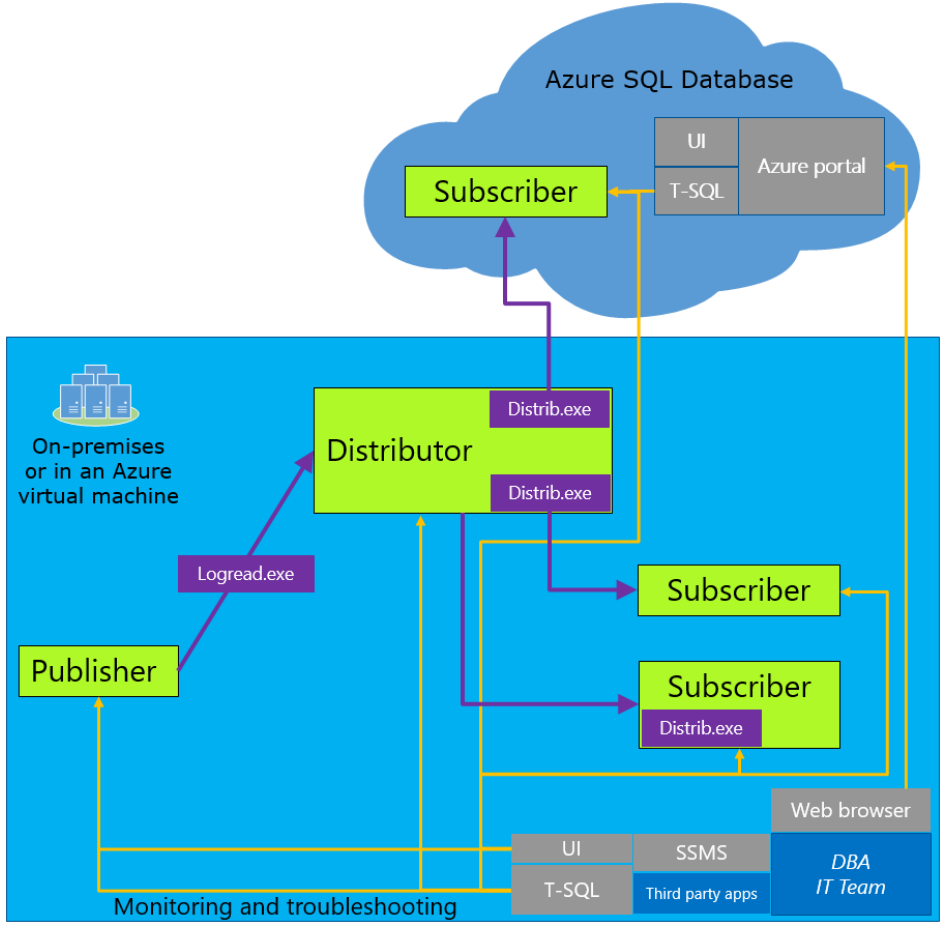
Method 3



Refactor with Visual Studio and Data Migration Assistant, final deploy with SSMS

Migration with minimal downtime

Transactional Replication



Data Migration Resources

Blog

https://techcommunity.microsoft.com/t5/Microsoft-Data-Migration/bg-p/MicrosoftDataMigration

Upgrade SQL Server

https://docs.microsoft.com/sql/database-engine/install-windows/upgrade-sql-server

Post-migration validation and optimization guide

https://docs.microsoft.com/sql/relational-databases/post-migration-validation-and-optimization-guide

Azure Database Migration Guide

https://datamigration.microsoft.com

Thanks!

#SQLSAT895