Migrating our old SQL Server to the modern platform

PASS SQL ARGENTINA

Javier Villegas

DBA Manager at Mediterranean Shipping Company

Involved with the Microsoft SQL Server since early versions

Specialization in SQL Server Administration, Performance Tuning and High Availability

Microsoft MVP Data Platform

MCP and MCTS

Technical Speaker

SQL PASS, 24 HOP, SQL Saturdays , PASS Marathon and PASS Virtual Groups,

vOpen, Microsoft AI Tour, GroupBy and DataPlatformGeeks

@sqlargentina

javier.ignacio.villegas@gmail.com

@javier_vill

javiervillegas

sql-javier-villegas.blogspot.com.ar

Microsoft® MVP Most Valuable Professional

Microsoft CERTIFIED AZURE DATABASE ADMINISTRATOR ASSOCIATE

Microsoft CERTIFIED AZURE DATA FUNDAMENTALS

Agenda

- Introduction
  - SQL Server 2008 / SQL Server 2008 R2 end of support
- Database Compatibility Level
  - Cardinal Estimator (CE) Versions
  - Trace-Flags
  - Scoped-Database Configuration

Agenda

- Options
  - Move to newer version of SQL Server
    - On-Prem SQL Server 2017/2019
  - Move to Azure VM running SQL Server On-Prem (IaaS)
  - Move to SQL Azure Database (SaaS)
  - Move to SQL Azure Managed Instance (SaaS)

Agenda

- Migration / Upgrade Tools and Services
  - SQL Upgrade Advisor
  - Backup/restore
    - Backup to URL
  - Import Export Wizard
  - Data Migration Assistant (DMA)
  - Database Experimentation Assistant (DEA)
  - SQL Server Migration Assistant (SSMA)
  - Azure Database Migration Service (DMS)
  - SSMS 18 - Database Upgrade Wizard

SQL SERVER END OF SUPPORT

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 mainstream support until CY 2017	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/

Agenda

- Post-Migration
  - Database Compatibility Level
  - Cardinal Estimator (CE) Versions
  - Trace-Flags
  - Scoped-Database Configuration
- Maintenance Tasks
  - Update Statistics
  - Index Rebuild
- HA/DR Options
- New Roles of DBAs

Microsoft

SQL Server 2008 and 2008 R2 end of support is coming

SQL Server 2008 and SQL Server 2008 R2 will no longer be supported by Microsoft starting in July 2019. Avoid challenges and vulnerabilities caused by end of support.

Why should you upgrade?

Mitigate risks with platform security and compliance
There will be no access to critical security updates, opening the potential for business interruptions and loss of data.

Upgrade to better cost efficiency
Maintaining legacy servers, firewalls, intrusion systems, and other tools can get expensive quickly.

Modernize to innovate
Grow your environments with data, analytics and the cloud.

More than an upgrade

With **SQL Server 2017** and **Azure SQL Database** Managed Instance, you don't just get an update – you get in-memory performance across workloads, mission-critical high availability and built-in security features to help protect your data at rest and in motion.

Learn more about SQL Server 2017 here

SQL

#1 OLTP performance[1]

#1 DW performance on 1TB[2]**, 10TB**[3]**, and 30TB**[4]

#1 OLTP price/performance[5]

#1 DW price/performance on 1TB[2]**, 10TB**[3]**, and 30TB**[4]

End of support offers for SQL Server 2008 and 2008 R2

Save up to 85% with Azure Hybrid Benefit

Save even more when you migrate your SQL Server 2008 or 2008 R2 to Azure SQL Database Managed Instances, a fully managed, "version-free" database-as-a service, or to Azure virtual machines with the Azure Hybrid Benefit.

Extended Security Updates for on-premises environments

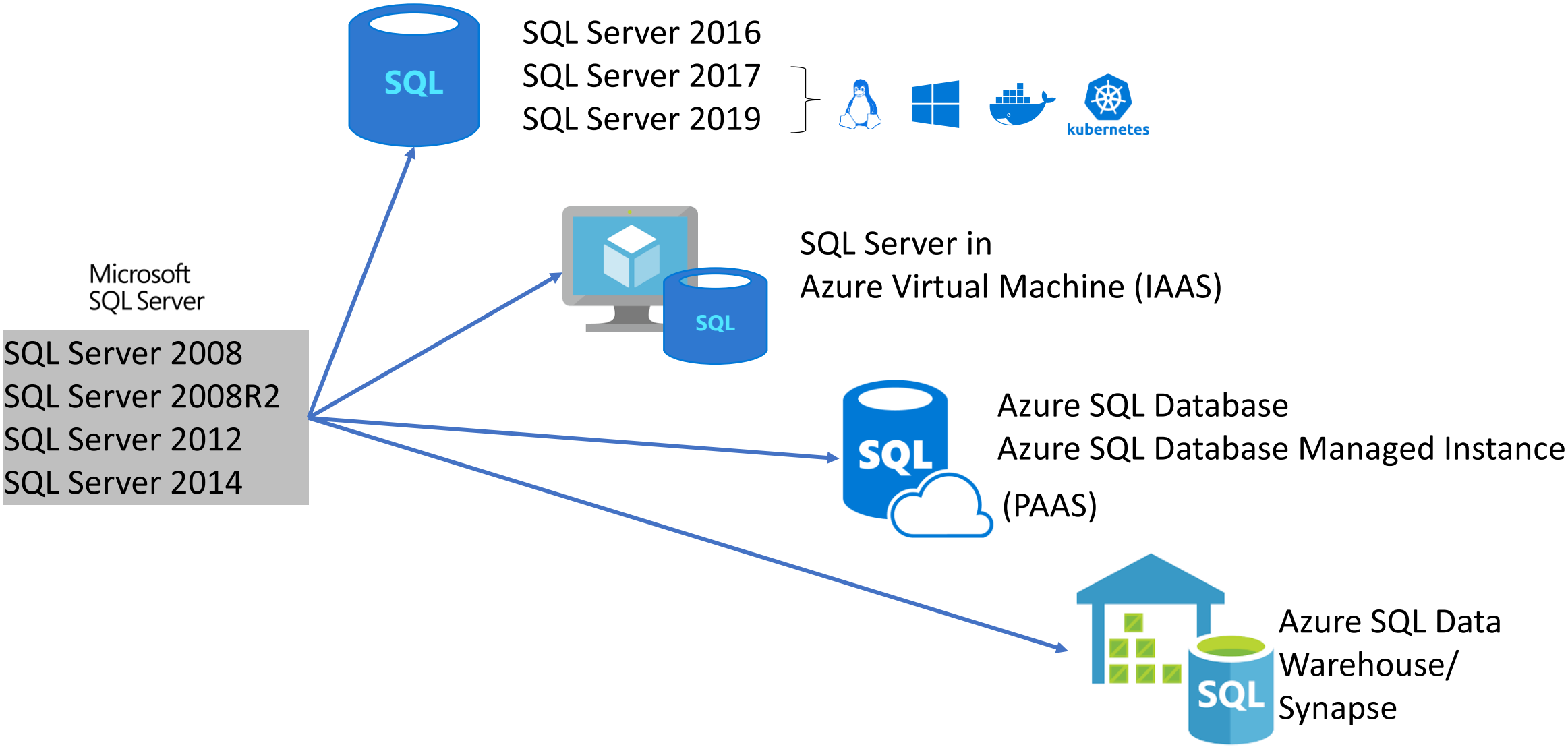
Customers with Software Assurance or subscription licenses may purchase Extended Security Updates for 3 years of security updates for SQL Server 2008 and 2008 R2. Customers can purchase Extended Security Updates for only the servers they need to cover.

Free Extended Security Updates in Azure

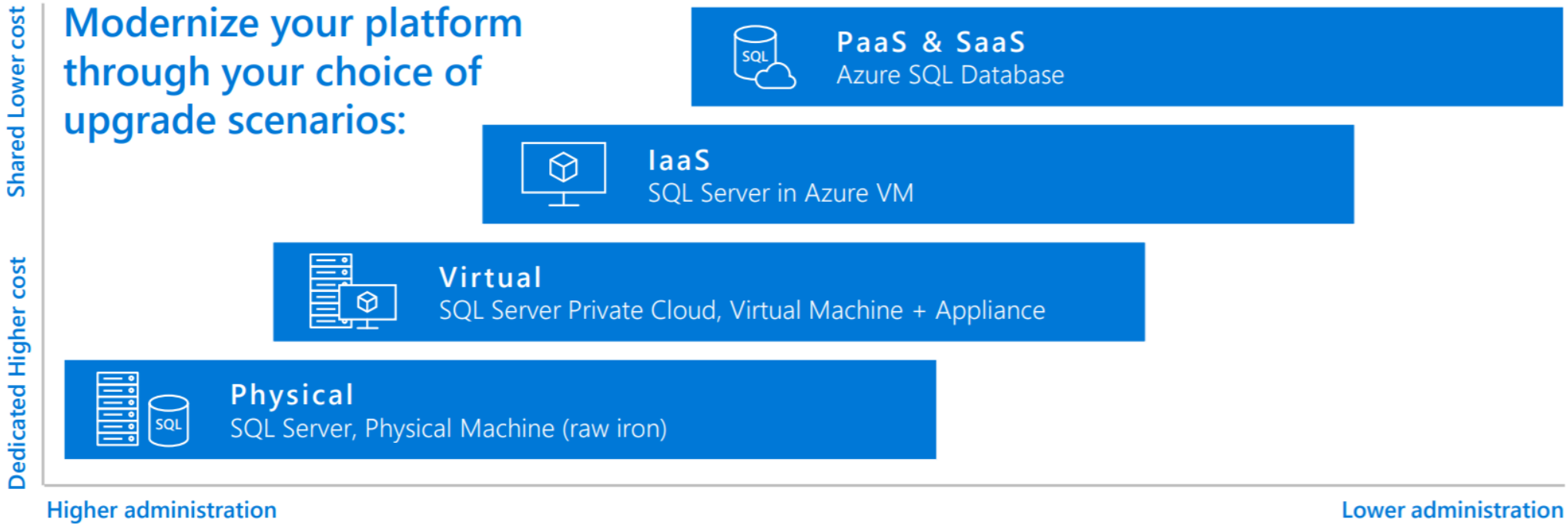
Lift and shift your SQL Server and Windows Server 2008 and 2008 R2 workloads to Azure with no application code change right away. This gives you more time to plan your upgrade to newer versions such as SQL Server 2017 or Windows Server 2016 running in Azure.

https://www.microsoft.com/en-us/sql-server/sql-server-2008

Azure Data Modernization Choices



Modernize your platform through your choice of upgrade scenarios:



All TPC Claims as of 1/19/2018. [1]http://www.tpc.org/4081; [2]http://www.tpc.org/3331; [3]http://www.tpc.org/3326; [4]http://www.tpc.org/3321; [5]http://www.tpc.org/4080

New features since 2008

OLTP Performance

- Hybrid Transactional and Analytical Processing with real-time insights using In-Memory OLTP and Clustered Columnstore
- In-Memory Data Warehouse with Clustered Columnstore Indexes
- Unparalleled scalability with Windows Server 2016, with 24TB memory and Windows Server 2016 max cores
- Enhanced Always On Availability Groups with easy GUI-based Wizard
- Automatic failover between three synchronous replicas. Up to eight secondary replicas
- Buffer Pool Extension to SSDs
- Adaptive Query Processing
- Resource Governor with CPU, Memory and IO Governance
- Delayed Durability of Memory-Optimized Tables
- Query optimization enhancements
- Query Store
- Local DB runtime (Express)
- Data-tier application component project FileTable build on FILESTREAM
- Remote Blob Storage with SharePoint 2016
- Statistical Semantic Search
- Contained Database Authentication
- Automatic Plan Correction

Security

- Always Encrypted
- Dynamic Data Masking
- Row-Level Security
- Auditing
- CC certification at EAL2 level for SQL Server 2016
- Backup encryption support
- Enhanced separation of duties
- Default schema for groups

Enterprise Information Management (EIM)

- Enhanced Integration Services
- Data Quality Services
- Master Data Services
- Extensible object model

Business Intelligence

- Enhanced connectors, new transformations, object-level security, ragged hierarchies**
- Mobile BI
- Enterprise-grade Analysis Services
- In-Memory analytics with Analysis Services Tabular Model
- Enhanced multidimensional models
- Modernized Reports and Dashboard with support for Mobile BI
- Create mobile reports using the SQL Server Mobile Report Publisher
- Consume with Power BI mobile apps
- HA for StreamInsight, complex event processing
- Import PowerPivot models into Analysis Services
- Advanced tabular mode
- Enhanced productivity and performance
- Power View
- Configurable reporting alerts
- Reporting as SharePoint Shared Service
- Master Data Services Add-in for Excel

Hybrid Cloud

- Disaster Recovery environment in Azure VMs using Always On Availability groups
- Stretch database
- Hybrid scenarios with SSIS
- Enhanced backup to Azure
- Easy migration to the cloud
- Secure backups to Azure with managed backups

Programmability Enhancements

- JSON support
- Spatial features, Full Globe and arcs
- Support for Temporal Tables
- Graph data support

Data Warehousing & Big Data

- Operational analytics
- Enhanced In-Memory DW with clustered ColumnStore
- Big Data integration with Hadoop and Azure Blob Storage
- Big Data Queries with T-SQL
- Enhanced database caching
- Direct Query

Advanced Analytics

- Machine Learning with built-in support for Python and R
- Operationalize Advanced Analytics where the data lives
- RRE APIs with full parallelism and no memory limits for scale/performance
- Built-in In-memory Advanced Analytics
- Advanced data mining

Platform

- Support for Linux (RHEL, SUSE, Ubuntu)
- Support for Docker containers and Kubernetes

The innovation of SQL Server 2016 and 2017

SQL Server 2016

- Query Store
- Polybase
- Temporal Tables
- JSON
- Always Encrypted
- Dynamic Data Masking
- Row Level Security
- R and Machine Learning
- It Just Runs Faster

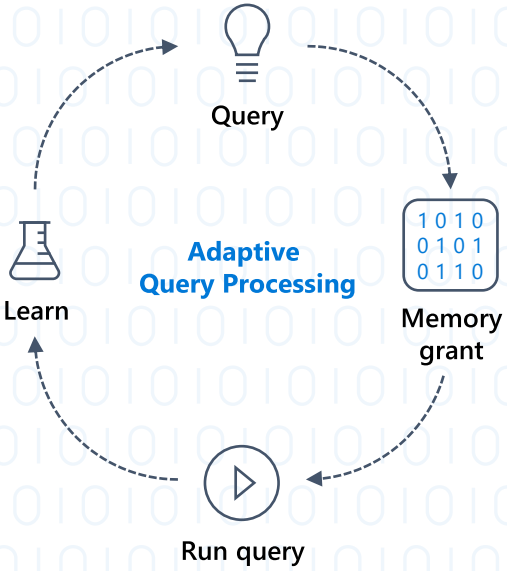
SQL Server 2017

- Linux and Containers
- Adaptive Query Processing
- Automatic Tuning
- *Clusterless* Availability Groups
- Graph database
- Python

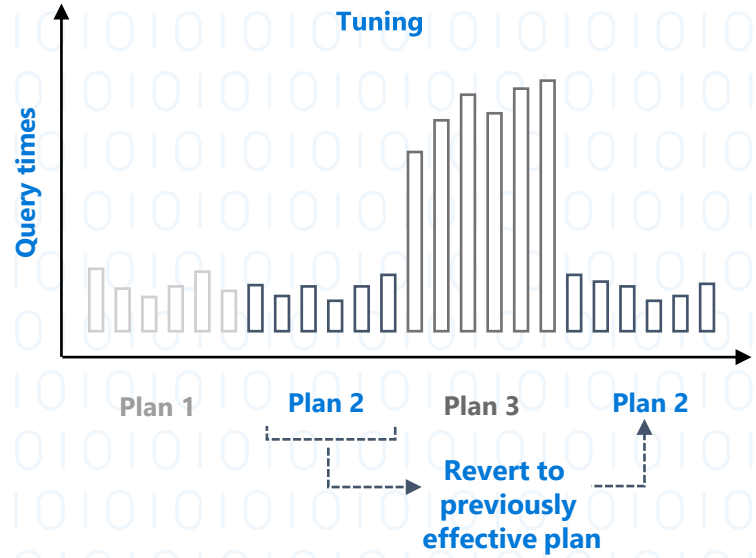
SQL SERVER 2017

Key New Functionality

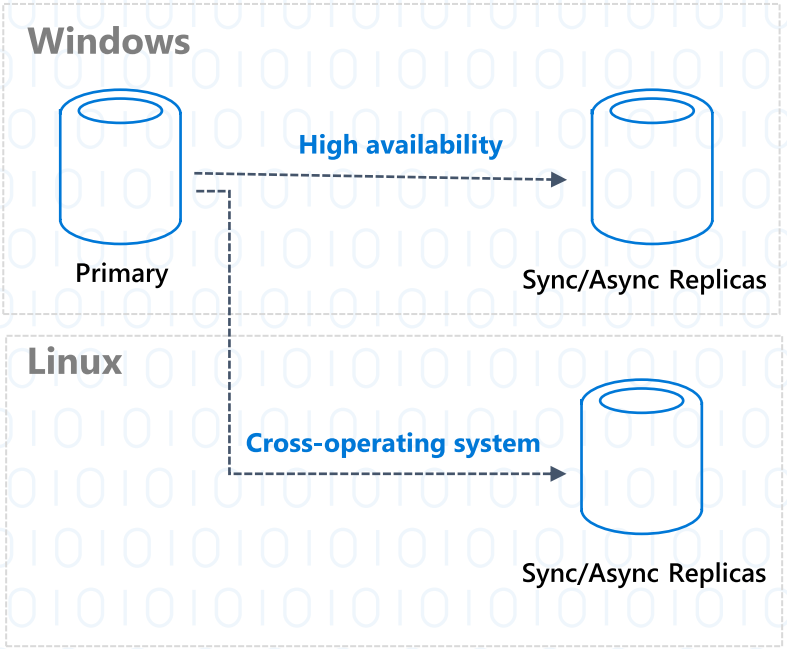
Platform of Choice



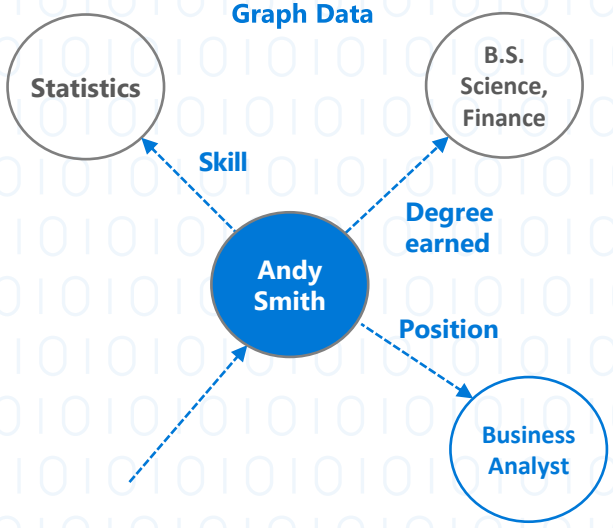
Automatic Tuning



"Clusterless" Availability Groups



Graph Data



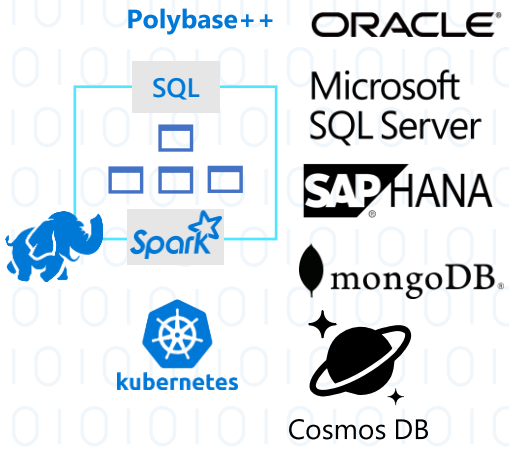
Built-in Machine Learning

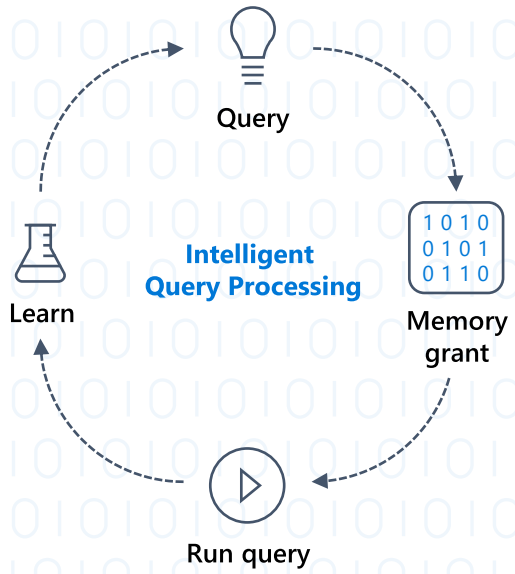
R and Python +
in-memory at massive scale

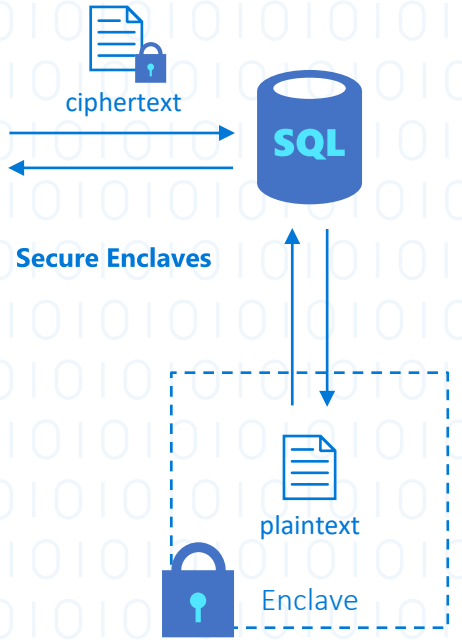
Native T-SQL scoring

SQL SERVER 2019

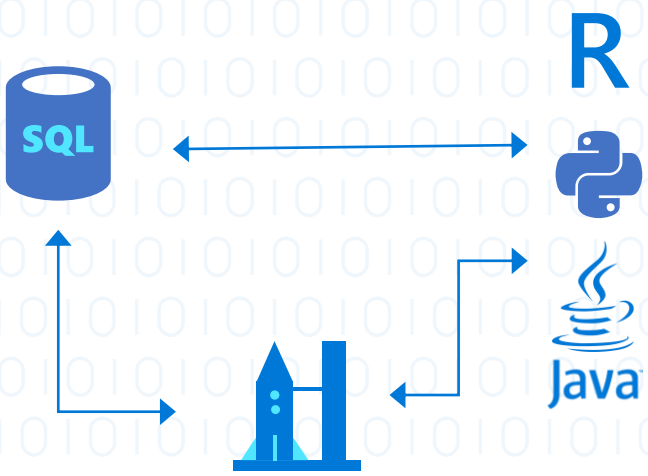
Key New Functionality





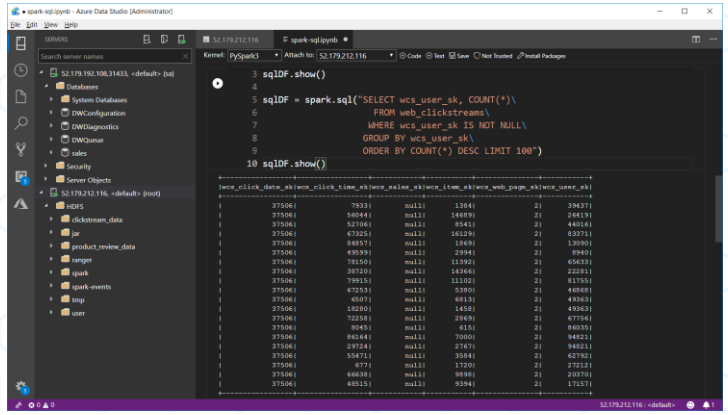


Built-in Machine Learning and Extensibility



Modern Platforms

Azure Data Studio



SQL Server 2017 on-premises

Build intelligent, mission-critical applications using a scalable, hybrid data platform for demanding workloads. Get started with a 180-day free trial of SQL Server 2017 on Windows.

Download free trial ↓

SQL Server in the cloud

Take advantage of the built-in high availability, security, and intelligence of Azure SQL Database, and use the familiar SQL engine without the complexity of infrastructure management. Start using SQL Database free in Azure.

Start free >

Install SQL Server 2019 on Windows, Linux, and containers

Windows

Run SQL Server 2019 database engine on Windows.

Choose your installation setup >

Linux

Run SQL Server 2019 database engine on Linux.

Choose your installation setup >

Docker

Run SQL Server 2019 database engine container image with Docker.

Choose your installation setup >

Big data analytics

Run SQL Server 2019 big data analytics container images with Kubernetes.

Choose your installation setup >

Opportunities with Azure

Scale Across Geographies

- Minimum infrastructure outlay
- 30 Azure regions with plans for 8 additional

Shrink and Grow on-demand

- Cater to different billing cycles (monthly / quarterly)
- Elastic auto-scale to meet desired performance

Lower Barriers of Entry

- Market to mid-market law firms via custom service bundles via Micro-services.
- Throttle payloads through tiered pricing and monthly subscriptions or Pay as you go.

More Data Residency Options

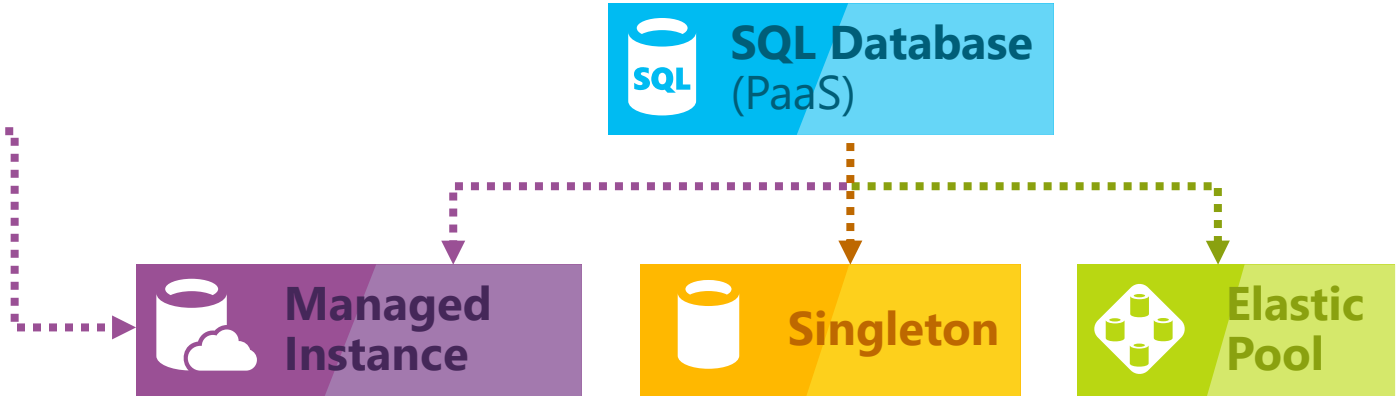
- Address International Compliance and Data residency concerns (GDPR etc.)
- Target 100+ Firm / Case Management systems

Realize untapped Potential

- Over 27k Thomson Reuters Legal Customers are yet to benefit from eBilling.

What is SQL Database Managed Instance?

A flavor of SQL DB designed to provide easy migration to fully managed PaaS



Unmatched app compatibility

- Fully-fledged SQL instance with nearly 100% compat with on-prem

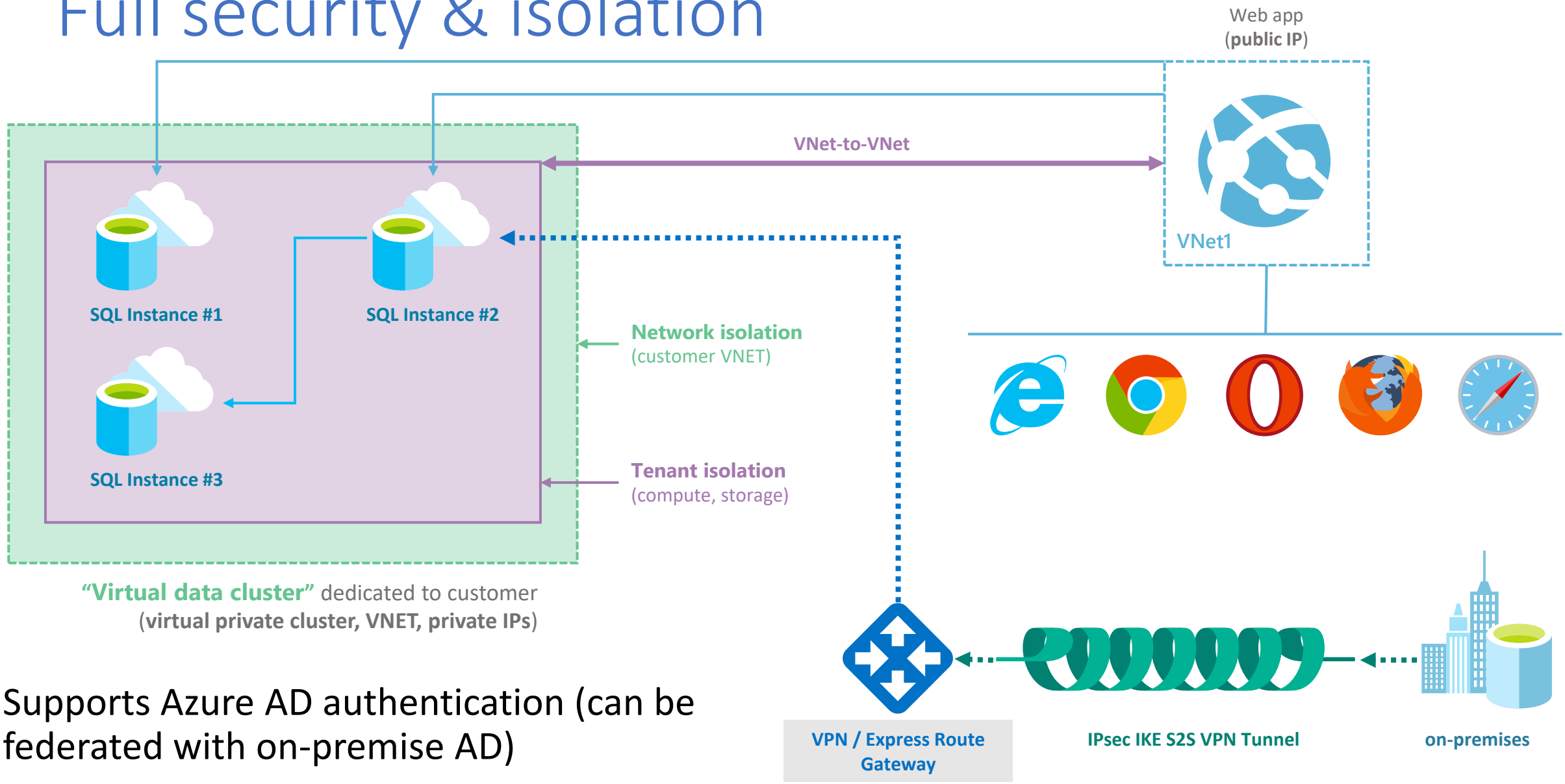
Unmatched PaaS capabilities

- Lowest TCO + rich Azure ecosystem
- Built-in auto management

Favorable business model

- Competitive
- Transparent
- Frictionless

Full security & isolation



Supports Azure AD authentication (can be federated with on-premise AD)

DEMO

PASS SQLARGENTINA

Why Upgrade?

Supportability

- SQL Server 2005 ended support in April 2016
- SQL Server 2008 & 2008 R2 (parts of 2012 as well) are now in the extended support portion of their lifecycle
- Extended Support phase – not possible for design changes

- Microsoft is planning on releasing a new version of SQL Server every 18-24 months
  - SQL Azure updated every 4 months
  - Industry is still managing SQL Server like they were in the 1990's
    - In a lot of cases even more risk averse
    - Need to be more agile

SQL Server Time Lines

- Microsoft is more aggressively releasing new version of SQL Server going forward

SQL Server Version	RTM Date	Delta
SQL Server 2016	US Summer 2016	< 24 months
SQL Server 2014	April 2014	25 months
SQL Server 2012	March 2012	23 months
SQL Server 2008 R2	April 2010	20 months
SQL Server 2008	August 2008	33 months
SQL Server 2005	November 2005	120 months
SQL Server 2000	November 2000	24 months
SQL Server 7.0	November 1998	-

Why Upgrade?

Features

- Many new features or existing features are enhanced in the new product.

Scalability

- Increased number of partitions for partitioned tables
- Increased amount of compute that can be accessed with the different versions
- Larger more complex and reliable configurations

Check for changes

- SQL Server documentation covers features that are deprecated or discontinued
- Do not miss ‘Breaking changes’
- Backward Compatibility documented for each component

Upgrade Blockers

- Vendor
  - Software not supported on later SQL Server editions
  - Learn from “mistakes of past”
- Risk
  - Database / code will break after
  - Unknown dependencies
  - A lot of risk is “perceived risk”
- Lack of Resources
  - IT Pro / Developers
  - Domain level knowledge
  - There are plenty of tools out there than can help analyze your current state

Planning

Use data from the Analysis stage

This will drive the sizing of the new system

Understand the needs of the business

Which teams will need to be involved in the cut-over

How long can they be without the system, this will help you decide on the actual migration approach

Planning to upgrade

- Preparing to Upgrade
  - Review upgrade documentation and resources
  - Document your resources and environment
  - Identify upgrade requirements
  - Decide on upgrade strategy
  - Upgrade High-Availability servers
  - Establish backup and rollback plans
  - Test the plan!!!

Interpretation

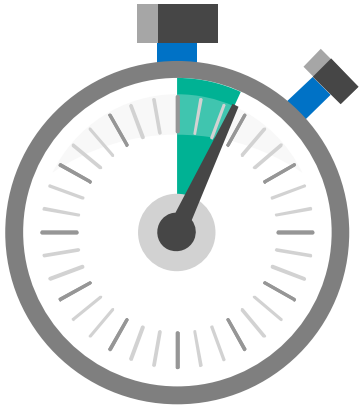
Understand what is important to you

Each SQL Server implementation is unique, there will be different performance metrics that matter to you

How will your system grow over time, look at historical baseline for trends to project requirements

How has the business changed over time and what are the plans for the future

What and How?



What to gather & analyze

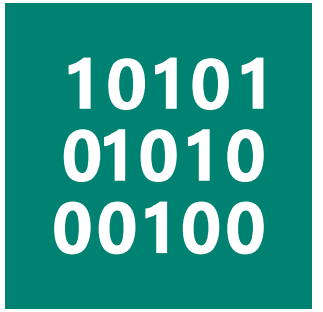
- Configuration to Counters
- Work to the initial scope and catalogue the estate
- Application and Business information :: Baselines

How to Analyze

- Handful of tools available to use
- Data Capture - wide period of time, multiple samples for the same period
- Include notable events in capture, like month end processing etc.
- Database Maintenance and Application Releases

CPU Sizing

- Parallelism
- Utilization
- Lower clock speed with more cache



Memory Sizing

- PLE – Page Life Expectancy & Buffer pool
- Leave room for growth
- Technology features – In-Memory OLTP

Storage Sizing

- Identify the IOPS requirements
- RAID vs SSD or PCI-e Flash
- Intelligent SAN or Dumb SAN?

Communications

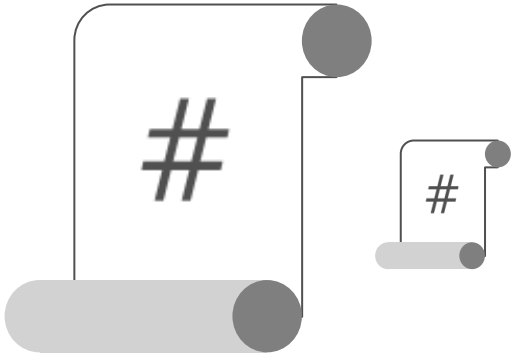
Users

- Communication plan is key and involve *affected* users *ahead* of time
- Notifications within the team
- Point of contact in the event of post-migration issues?

Who and when do key decisions get made as to whether to continue or rollback a migration on go-live day?

- Will all of the resources required be available and contactable
- Rollback plan - confidence to the business

Testing



What should I test?

- Old vs New
- Test the production hardware & pre-production systems
  - Create benchmark to test
  - Representative of queries in production system
  - Able to re-run after any configuration change
- Standardize on a deployment patch level
  - Document the suite of tests and automate
  - Each sub-system

Upgrade Strategy

Methods

Side-by-side

This involves building a new system alongside the existing one and moving databases to it.

In-Place upgrade

Upgrading the existing software to the new version, without the need to move the databases.

New Instance

Everything new from design to implementation stages.

Methods

Side-by-side

Allows for better testing of the new system prior to bringing it online.

There are more options when it comes to enacting a rollback in the event of an unsuccessful migration attempt.

Requires the system to be fully configured and checked against the existing system.

Will require that the application tier is redirected to new systems

Methods

In-Place upgrade

No need to move the databases to a new server

Server retains the same name for application connectivity

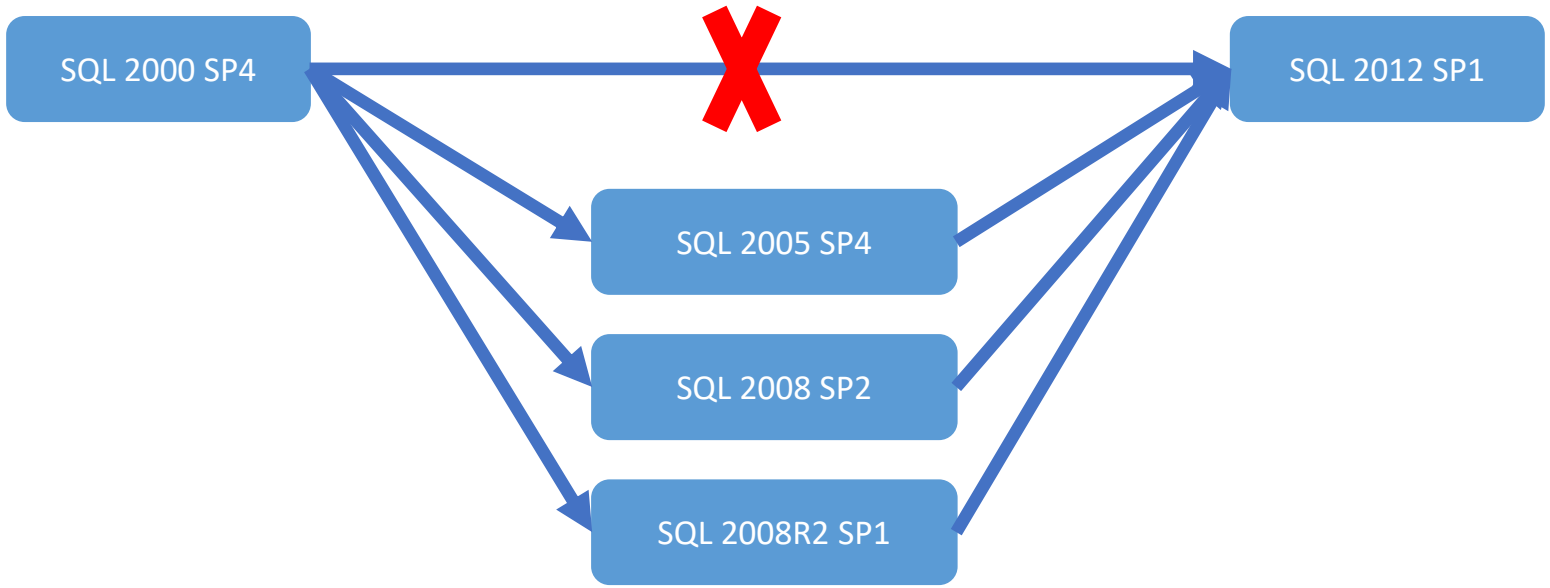
Configuration of the system is persisted to the new version

Difficult to rollback from

All applications must support the new version of SQL Server

The Path

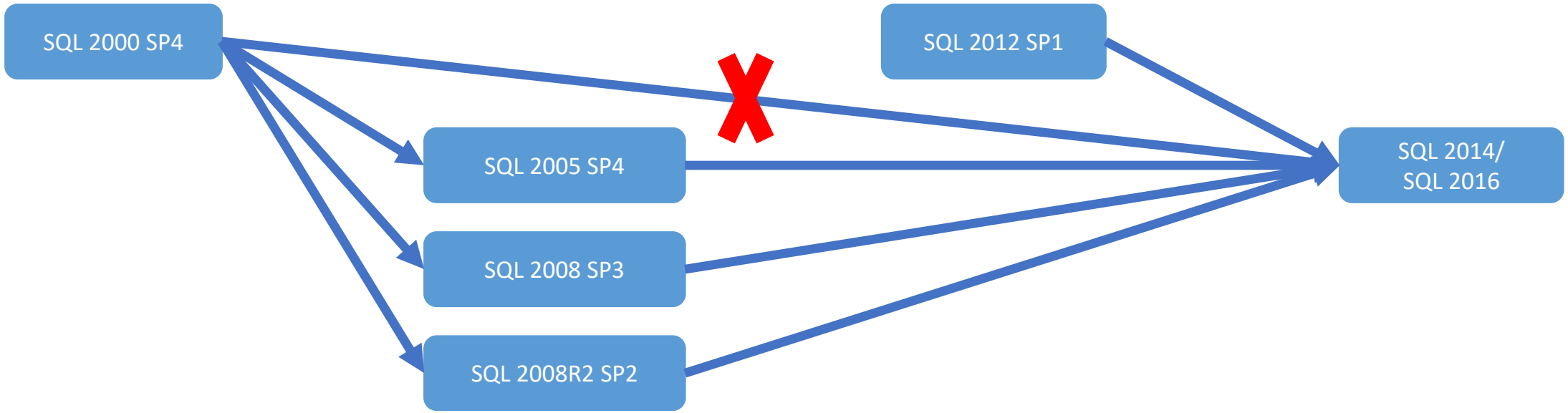
Upgrade Paths - 2012



The Path

Upgrade Paths

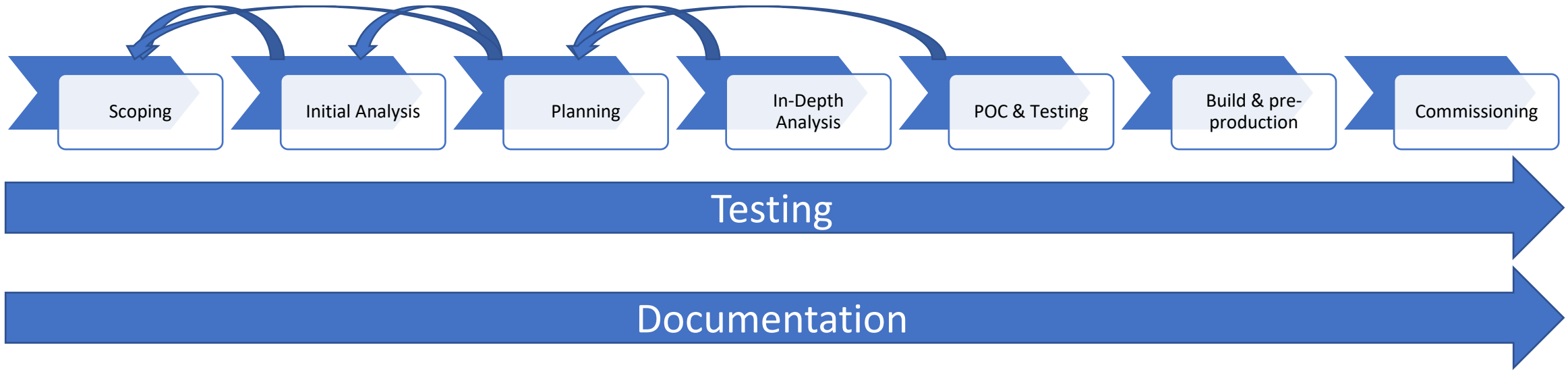
- Attach a SQL Server 2005 database (mdf/ldf files) to SQL Server 2016.
- Restore a SQL Server 2005 database to SQL Server 2016.
- Back up a SQL Server 2005 Analysis Services (SSAS) cube and restoring on SQL Server 2016.
- SQL Server 2005 → SQL Server 2016, the DB compatibility level will be changed from 90 to 100.



The Process

Upgrading SQL Server requires effort

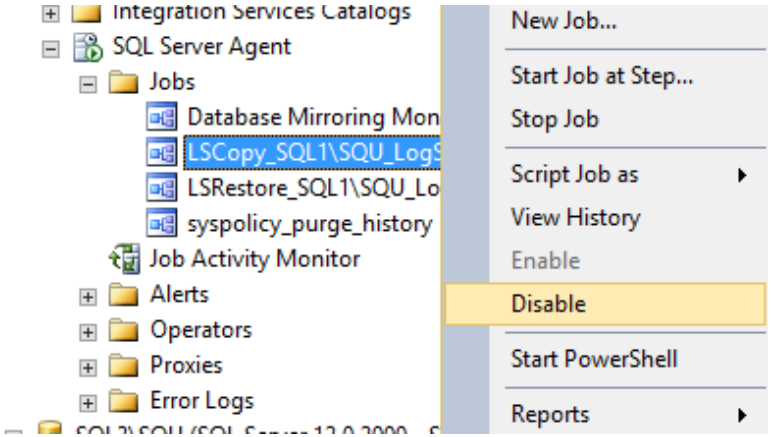
It is a multi-stage process that should be tackled in iterations feeding back into the process, flexing with what is discovered.



Upgrade – Log Shipping

Two methods – with or without a “role” change
The steps are:

- Disable log shipping jobs
- Upgrade secondary server(s)
- Upgrade monitor server if configured
- Re-enable log shipping jobs
- Catch up the secondary servers
- Pause primary server traffic
- Upgrade primary server



Upgrade – Mirroring

Mirroring is deprecated as of SQL Server 2012
Consider migrating to AlwaysOn Availability Groups

Upgrade – Failover Clustering - 1

Failover clustering features interact significantly with the Operating System
Windows Server 2012 R2 has major clustering improvements compared to Windows Server 2008
Therefore, in place upgrades not recommended

Upgrade – Failover Clustering - 2

However, it can be done

- Upgrade the secondary server(s) first
- Then, fail over to cause the database upgrades, and then upgrade the old primary node
- Special considerations for multi-node clusters – see the upgrade whitepaper - **http://tinyurl.com/SQLUpgradeDoc**

Upgrade – Failover Clustering - 3

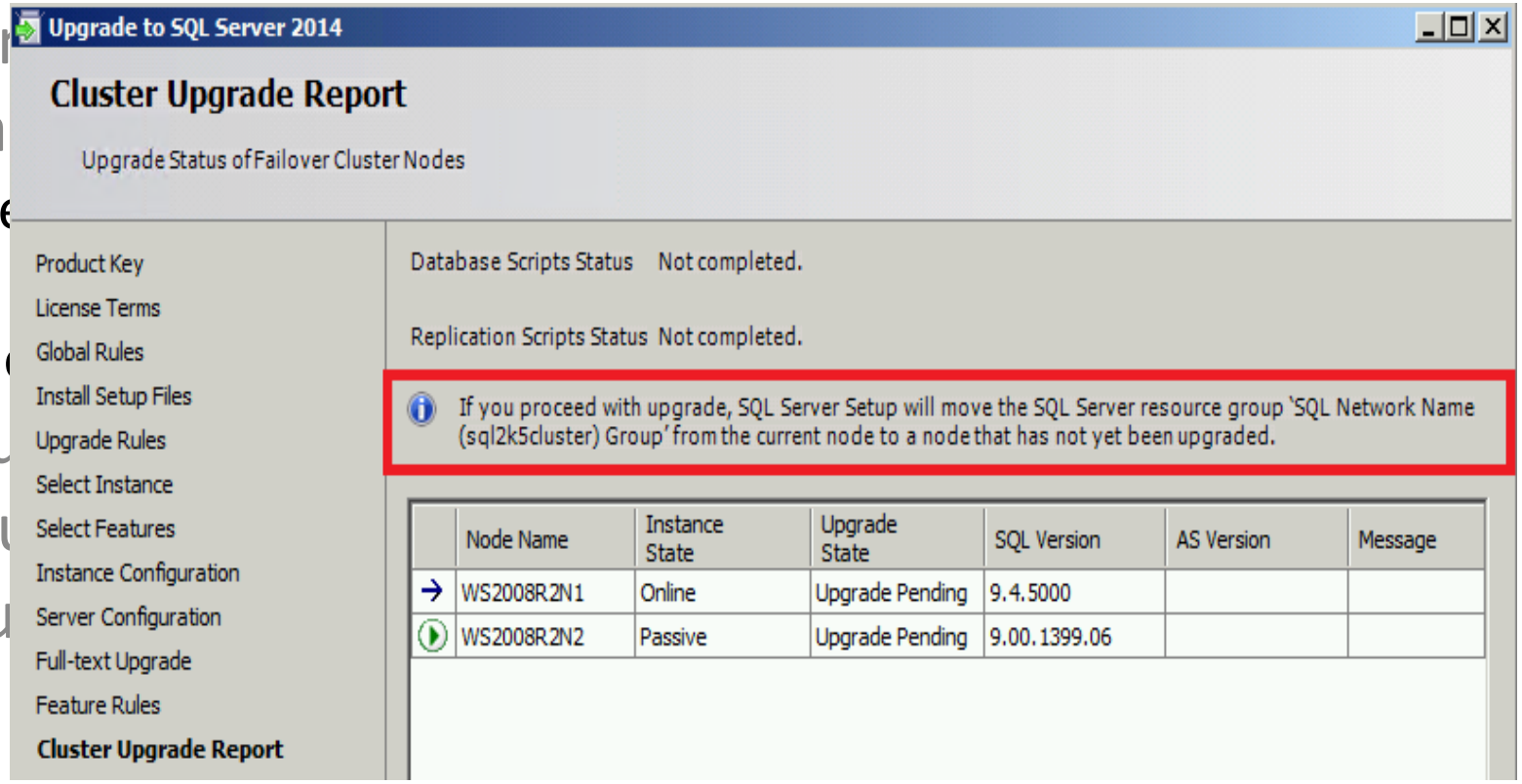
Each “cluster node” and i

Cluster upgrade will repla

- This will cause other cluste computer to go offline
- SQL Server Upgrade will d

If 50% or more of your clu automatically fail over you

Setup will warn you if you order



Upgrade - Replication

Always upgrade the distributors first

In-place upgrade generally recommended with replication to avoid re-sync costs

- Pausing the publication and ensuring all changes are pushed to subscribers is required before upgrade starts
- Disable replication-related SQL Server Agent jobs
- Upgrade distributor(s) → publishers → subscribers

Review upgrade paper and books online to validate your replication topologies are valid at every step

Upgrading your HA Solution

Consider AlwaysOn Availability Groups

- Replaces Database Mirroring (and some use cases for Log Shipping)
- Allows one automatic failover partner
- Allows 8 secondary copies, allowing read only and backups on secondaries

Consider Windows Server 2012 R2

- AlwaysOn heavily dependent upon Windows Clustering
- Windows Server 2012 R2 has huge clustering improvements

SSIS Upgrade Scenarios

Choose In-Place or Side-by-Side Upgrade

You can upgrade:

- ☑ SSIS 2005
- ☑ SSIS 2008
- ☑ SSIS 2008 R2
- ☑ SSIS 2012

Pre-2005 DTS packages must first be upgraded to one of the above versions

SSIS Backwards Compatibility

Deprecated features

🕐 Will be removed in a future release

Discontinued features

Examples
- 🚫 DTS
- 🚫 ActiveX scripting

More help

See additional references at the end of this slide deck

Behavior changes

Examples:
- ✋ SSIS 2005 VSA to 2008 VSTA scripting
- ✋ Can persist the cached reference table in Lookup transformation in SSIS 2008

SSIS Version Specific

The older the packages, the more potential problems

In-Place Scenario

Assumes upgrading both Integration Services and the Database Engine to SQL Server 2014

SSIS Upgrade Scenarios

Once completed…

Package are moved from the old SSIS package store to the new location

MSDB stored package are moved to the new SQL Server 2014/2016 instance

File system packages remain in place

No packages are automatically migrated to SSIS 2014/2016 package format

Moves log data, and folder metadata into the new SSIS 2014/2016 system tables

The older SSIS service continues to be available

Does not alter any SQL agent jobs (will continue to reference earlier dtexec utility

Side-by-Side Scenario

SSIS Upgrade Scenarios

Once completed…

Multiple SSIS platforms are available

Can use multiple editions of SSDT and BIDS

SSIS 2014 ‘dtexec’ can run earlier versions of SSIS packages

Allows for extensive testing

Recommended approach for refactoring earlier SSIS packages

The uninstallation of the old version can be done once the 2014/2016 migration is complete

Make best use of ETL Frameworks

May have custom ETL framework in current SSIS environment	Often used for consistent package configurations, logging, error handling and deployments
SSIS 2014 (as introduced in SSIS 2012) includes many 'Framework' features	May use to provide alternatives to an existing ETL framework
May consider 'refactoring' packages as a part of the SSIS upgrade	Can significantly increase the scope of an SSIS upgrade Project

Upgrading SSIS Projects

Project Conversion Wizard

Package Conversion Wizard

Project Model Conversion Wizard

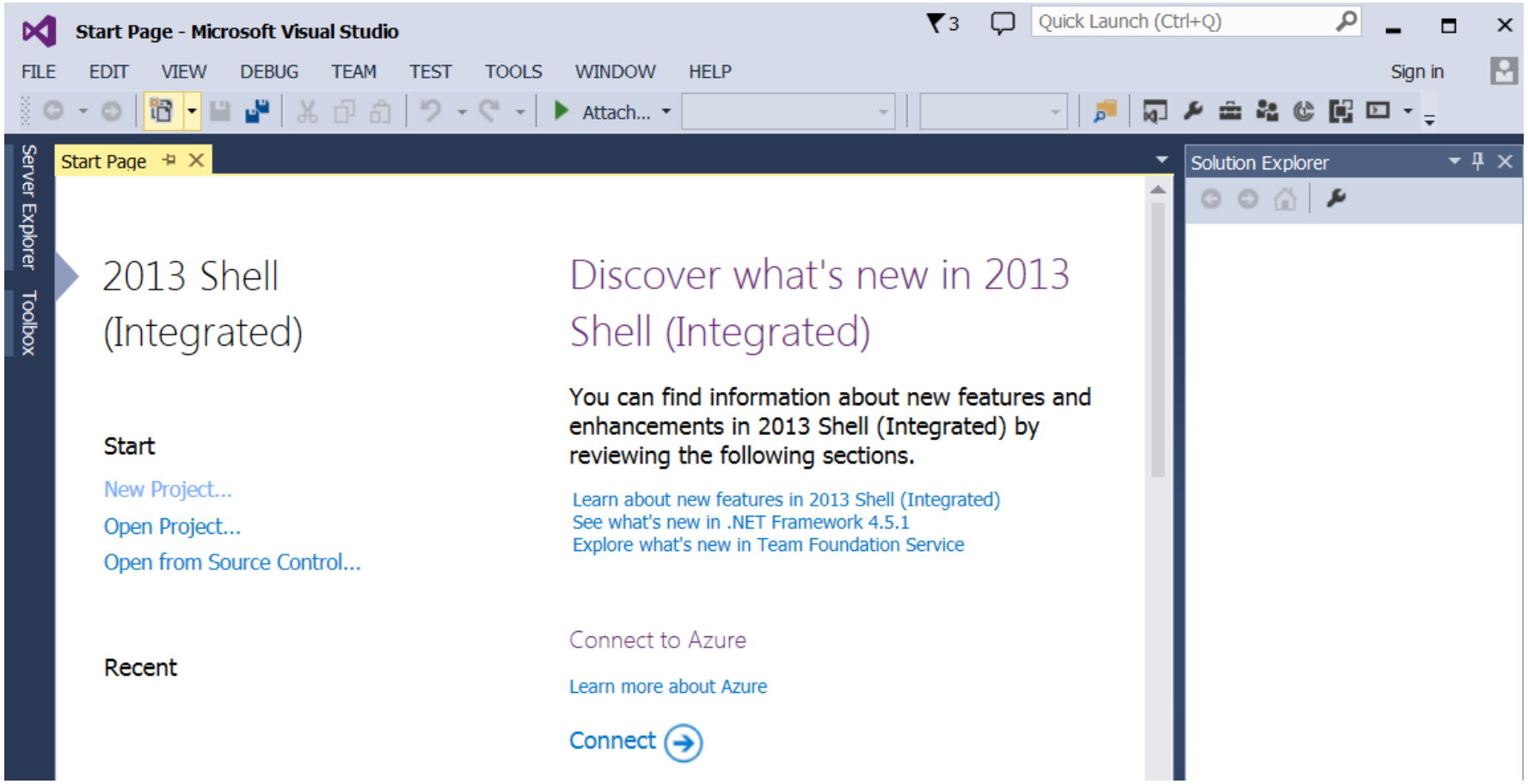
Converting to the Project Model is Optional

Project Conversion Wizard

Use a version of Visual Studio which supports the 2014 BI templates

Upgrades packages from previous SSIS versions - 2005, 2008, 2008 R2, and 2012

Start with Open Project…

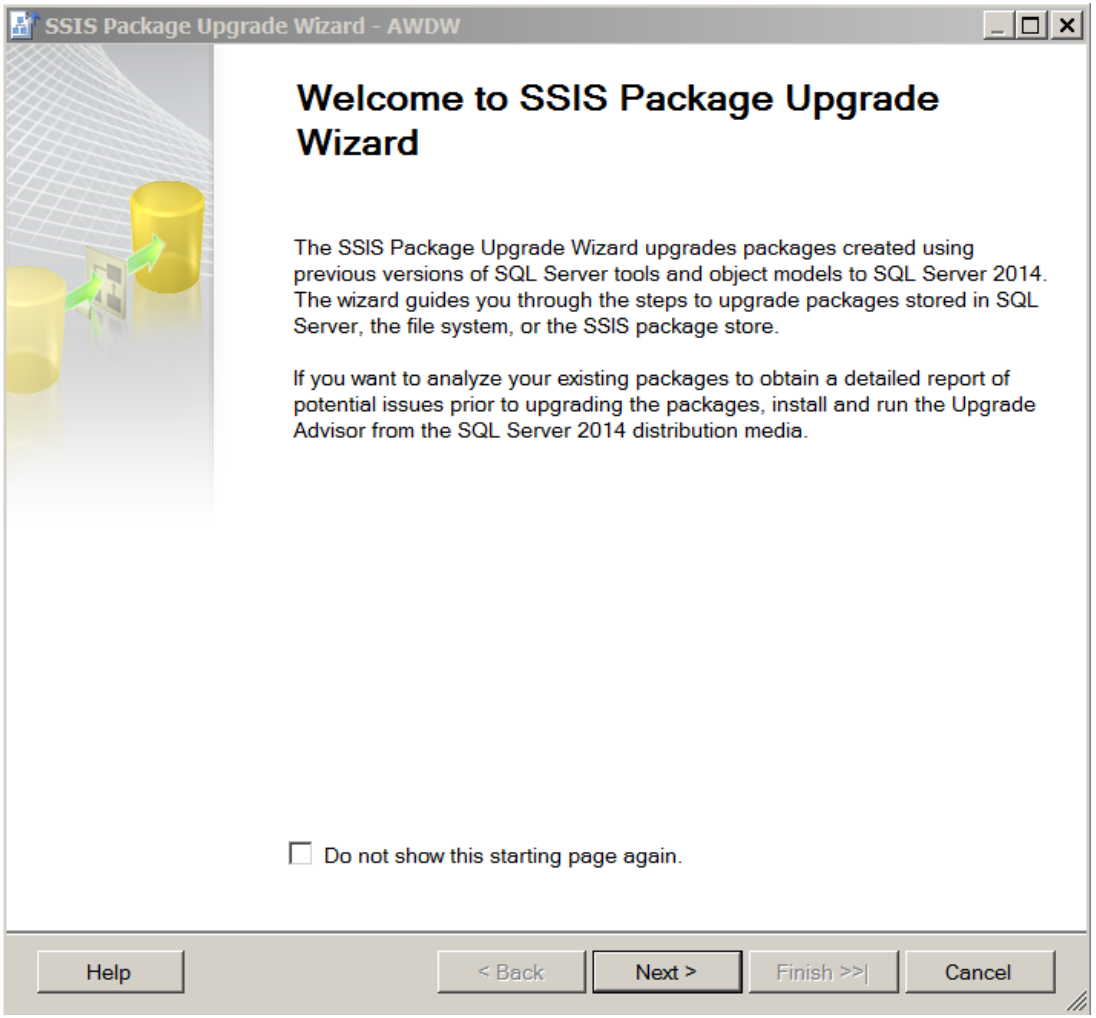


Package Conversion Wizard

Launched automatically from Project Conversion Wizard

Select Package Management Options

- ❑ Update connection providers
- ❑ Validate upgraded packages
- ❑ Creating new package IDs
- ❑ Continue on failure
- ❑ Ignore configurations

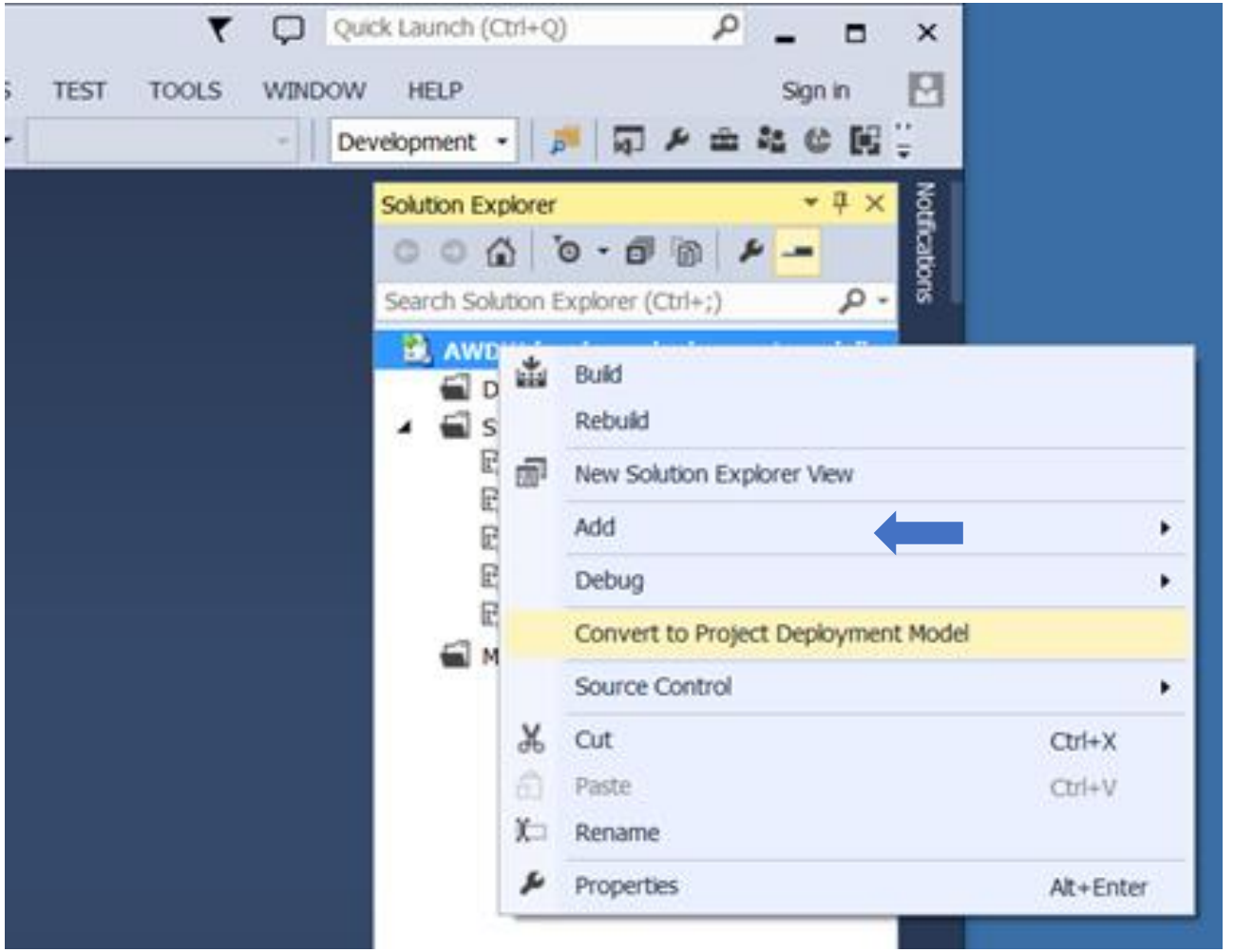


Project Model Conversion Wizard

Launched from the SSIS Solution Explorer

Project Deployment Model:

- Project is the unit of deployment
- Deployed to the SSISDB catalog
- Parameters are used to assign values to package properties
- Environment-specific parameters are stored in environment variables
- Offers catalog stored procedures and views



SSAS Upgrade Scenarios

Choose In-Place Upgrade or Side-by-Side Upgrade

You can upgrade:

- ☑ SSAS 2005
- ☑ SSAS 2008
- ☑ SSAS 2008 R2
- ☑ SSAS 2012

To SSAS 2014 Multidimensional mode using either the in-place or side-by-side method

Analysis Services Previous Versions (Multidimensional)

OLAP Services 7.0	Codename Sphinx (1998) Provided MOLAP, ROLAP, and HOLAP
SSAS 2000	Added distinct count measures Parent/child dimensions Improved aggregation design Data mining capabilities
SSAS 2005	Introduced UDM Increased scalability Added KPIs Improved data mining features
SSAS 2008	Improved wizards Improved VS tools for attribute relationships and aggregations Additional data mining models.
SSAS 2008 R2	No significant new features
SSAS 2014	Added Multidimensional Models to 'Power View'

In-Place Scenario

Select the Upgrade option in Set-up

Select the targeted instance to upgrade

Review summary actions and click Upgrade

Once completed…

The older SSAS engine and associated tools are removed

The older SSAS databases are moved, and do not need to be reprocessed

Only the SSAS 2014 instance will remain

To roll back in the event of a failure

Uninstall SSAS 2014

Reinstall the older version of SSAS

Restore the SSAS databases.

Can back-out at any point before clicking Upgrade

Side-by-Side Scenario

Install a new instance of SSAS 2014

Copy the databases using one of the following methods

- Backup and Restore
- Detach and Attach
- Script in XML/A and Run
- Deploy with Visual Studio

Once completed…

Both the previous version of SSAS, and SSAS 2014/2016 are available

Extended testing can be performed

The uninstallation of the old version can be done once the 2014 instance has been proven

Troubleshooting a Failed Upgrade

Review the setup logs that were created by the Setup application	Summary.txt Found in: %Program Files%\Microsoft SQL Server\120\Setup Bootstrap\Log
	Summary_[ComputerName]_[date]_[time].log Found in: %Program Files%\Microsoft SQL Server\120\Setup Bootstrap\Log\[date]_[time]

Post-Upgrade Tasks

Review upgraded databases

Expand the databases folder in SSMS

Browse the cubes

Verify database compatibility level

1050 – Multidimensional databases created in SSAS 2005, 2008, 2008 R2

1100 - Multidimensional databases created in SSAS 2012 or 2014

Summary

Upgrading to SSAS 2014 can be accomplished by using either an in-place upgrade or a side-by-side upgrade

Moving from earlier versions of SSAS provides performance and scalability improvements, along with a better set of developer tools

The in-place upgrade is more risky because it replaces the earlier version of SSAS

The side-by-side upgrade provides for two versions of SSAS run concurrently

A redesign should not be needed unless you migrate one or more cubes to the tabular model

Before you do an in-place upgrade, backups of all SSAS databases

Upgrade to Tabular?

- No migration path from multidimensional
- Power Pivot (Excel) model can be migrated to tabular server model
- Versions: 2012 (1100), 2012 SP1 (1103)

SSRS Upgrade Scenarios

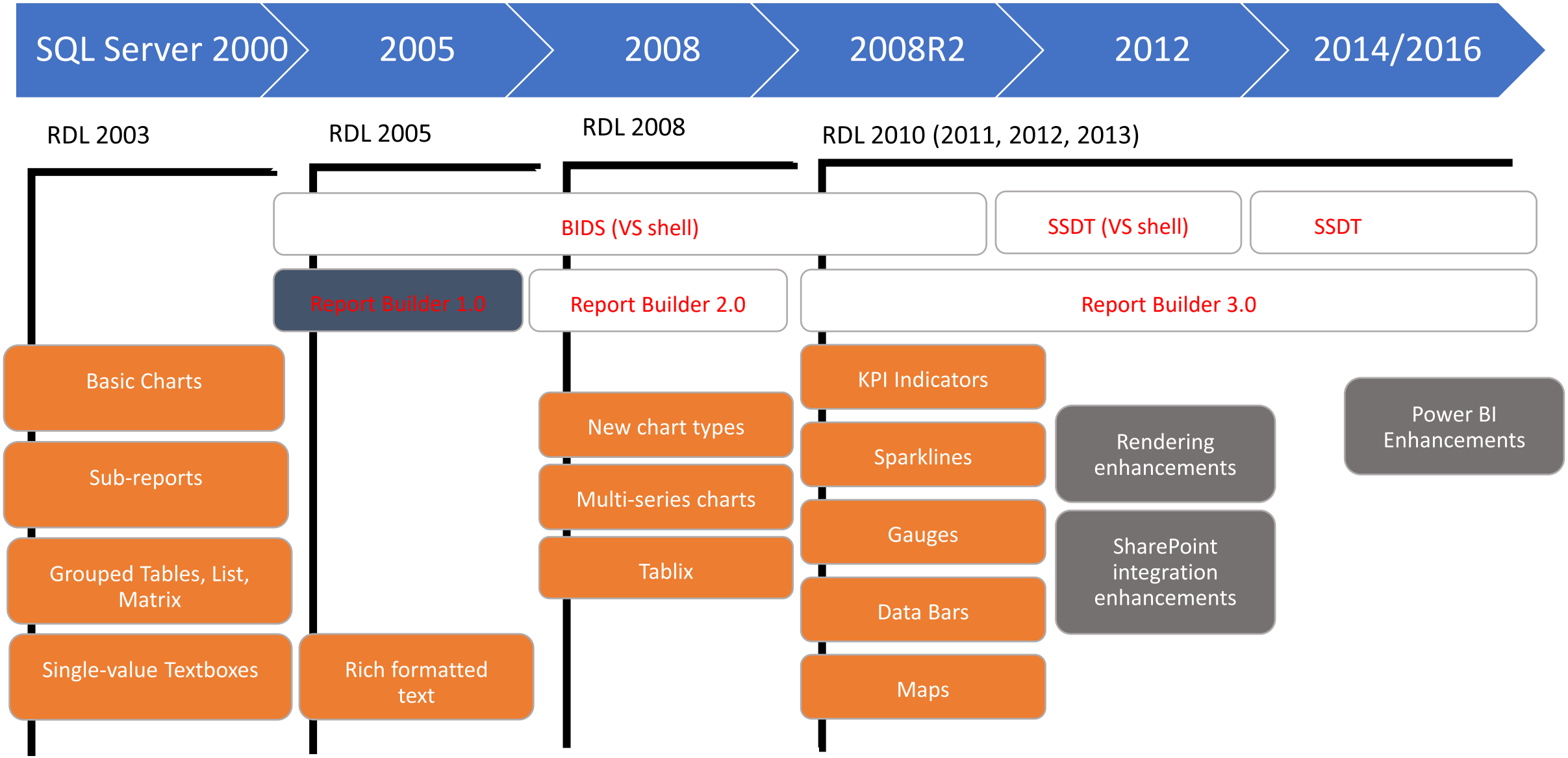
Choose In-Place Upgrade or Side-by-Side Upgrade

You can upgrade:

- ☑ SSRS 2005 (SP 4)
- ☑ SSRS 2008
- ☑ SSRS 2008 R2
- ☑ SSRS 2012

…to SSRS 2014 in either native or SharePoint integrated mode using either the in-place or side-by-side method

SSRS Versions & Feature Snapshot



Migrating From SSRS 2005

- Report server was IIS
- now http.sys in native mode
- No cross-platform migration (32 bit or 64 bit)
- Configuration settings (config files)
- Report models & Report Builder 1.0 reports are deprecated

Backup and Rollback Plan

Configuration Files

- Rsreportserver.config
- Rswebapplication.config
- Rssvrpolicy.config
- Rsmgrpolicy.config
- Reportingservicesservice.exe.config
- Web.config (for both the report server and Report Manager ASP.NET applications)
- Machine.config (for ASP.NET if you modified it for report server operations)

- Back up symmetric key
- Back up report server database(s):

  ReportServer contains all server content

  ReportServerTempDB contains no persistent objects & can be re-created

In-Place Upgrade

Select the targeted instance to upgrade

(optionally)
Migrate the database to 2008 R2+

Review summary actions and click Upgrade

Once completed…

The older report server and associated tools are removed

The older ReportServer database, configuration files & services and are moved

Only the SSRS 2014 instance will remain

To roll back in the event of a failure

Uninstall SSRS 2014

Reinstall the older version of SSRS

Restore the SSRS database & configuration files

Side-By-Side Upgrade

Install a new instance of SSRS 2014

Make copy of the SSRS project & deploy to the new report server

- Copy SSRS project
- Open & convert to new version
- Verify connectivity & formatting
- Deploy with Visual Studio

Once completed…

Both the previous version of SSRS, and SSRS 2014 are available

Extended testing can be performed

The uninstallation of the old version can be done once the 2014 instance has been proven

To roll back in the event of a failure

Uninstall SSRS 2014 server & database

Troubleshooting a Failed Upgrade

When looking for errors in the detail log, search for the following phrases:

- “Watson bucket”
- “Error: Exception has been”

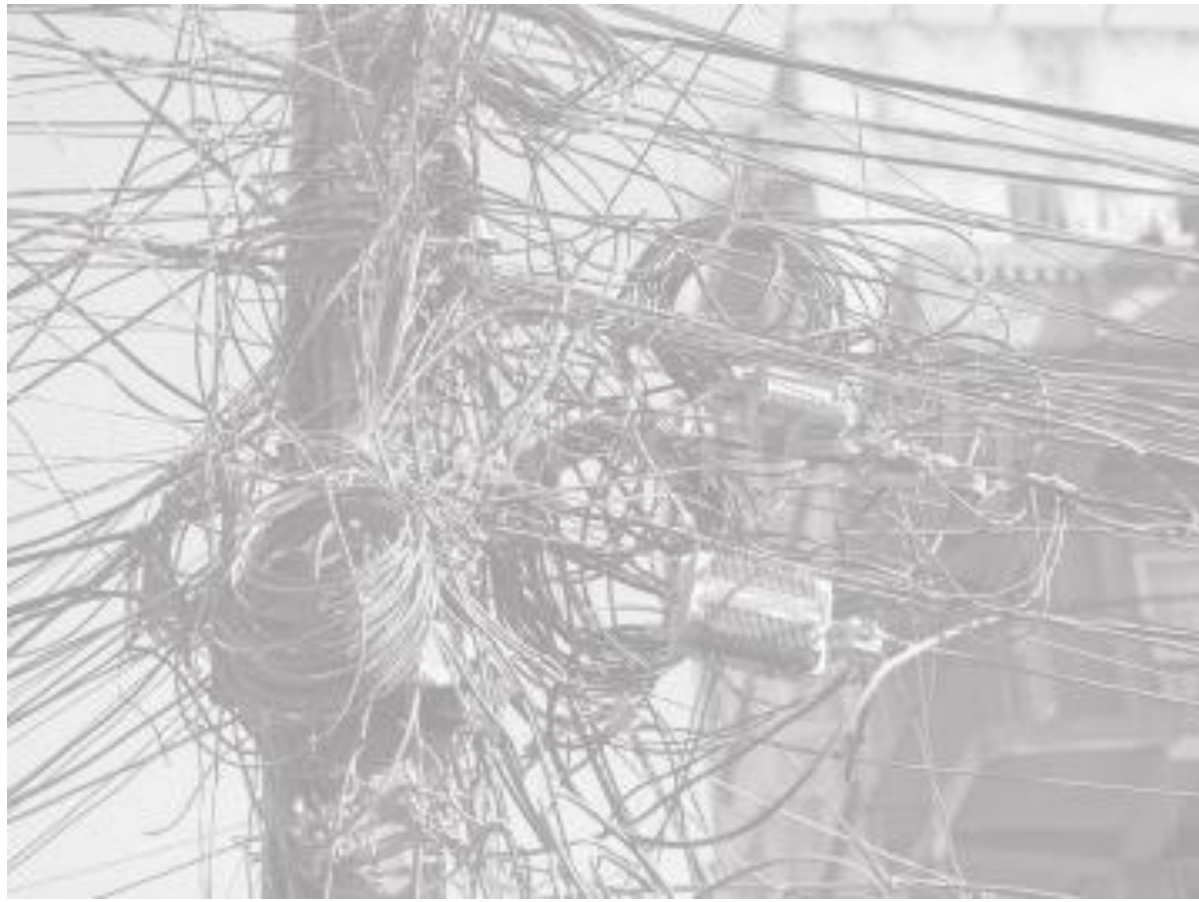
A typical Setup request goes through three execution phases:

- Global rules check
- Component update
- User-requested action

Each of these phases will generate detail and summary logs, with additional log files being generated as appropriate.

Setup is called at least three times per user-requested Setup action.
Typical log files generated are:

- Detail_GlobalRules.txt
- Detail_ComponentUpdate.txt
- Detail.txt

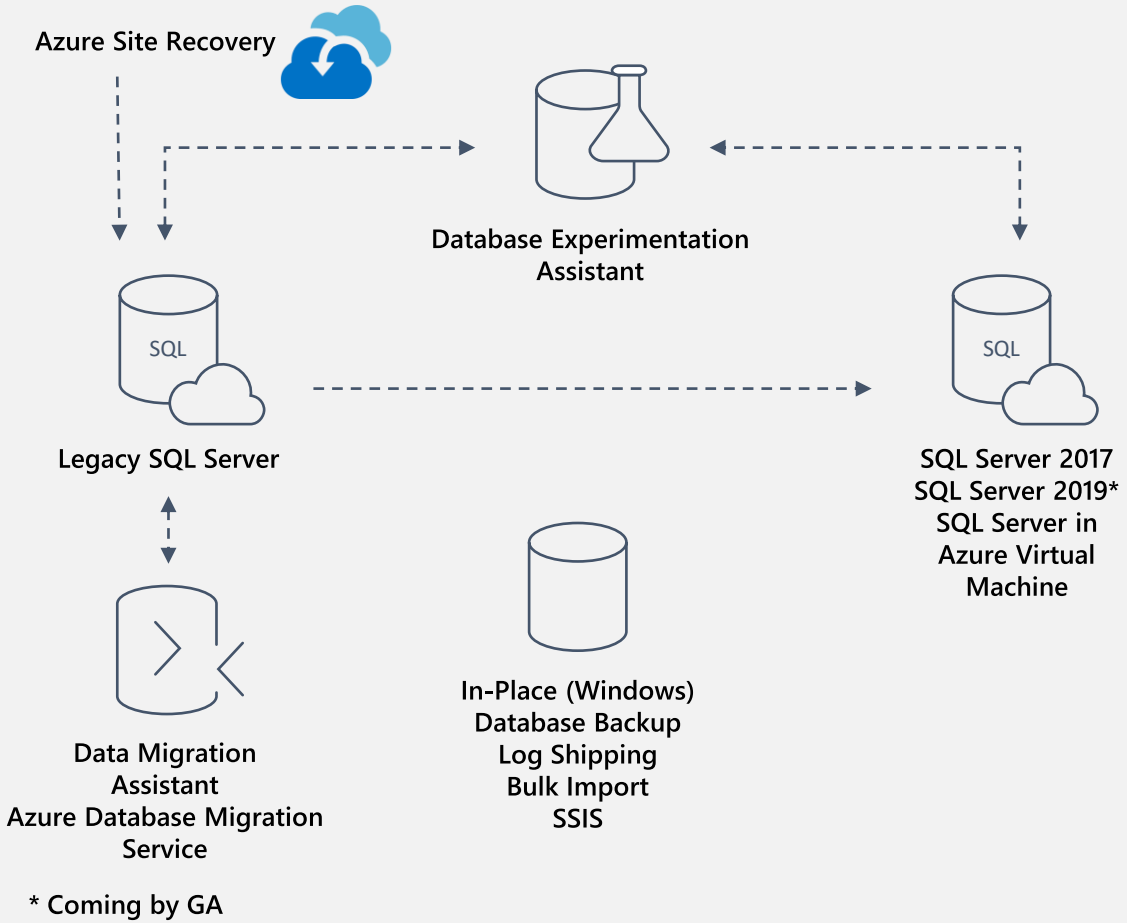


Summary

Upgrade from SSRS 2005 SP4+ to SSRS 2012 or 2014	In-place or side-by-side upgrade option
Backup SSRS content database, configuration files & symmetric key	Run the SQL Server Upgrade Advisor for SQL Server 2014 for potential issues
Upgrade SSRS project using Visual Studio	Upgrade SSRS instance using SQL Server setup upgrade option

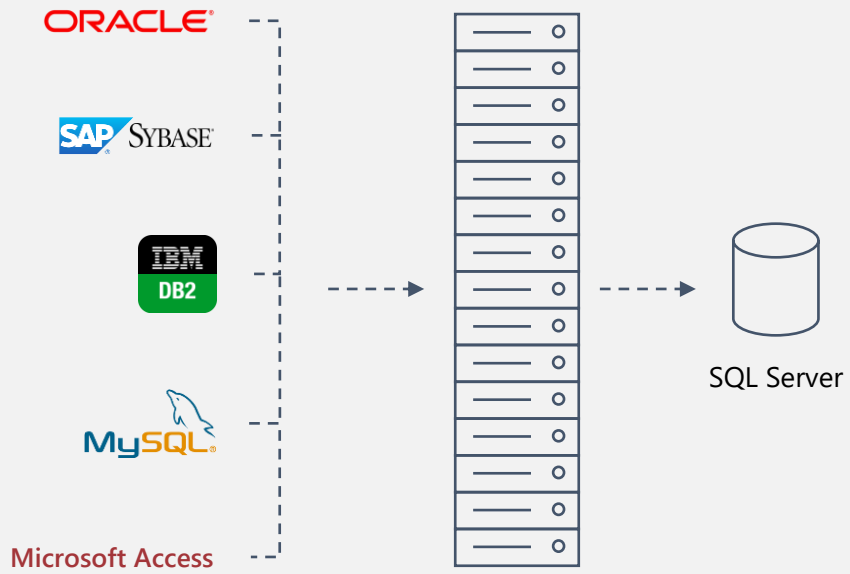
Migration to the Modern SQL Server

Migration from legacy SQL Server



* Coming by GA

Migration from external databases



SQL Server Migration Assistant (SSMA)

Database Compatibility Certification

Stop certifying for any given platform (Cloud, on-prem)!

Stop certifying for a named SQL Server version!

- Any certification process should be thought in terms of "*which target database compatibility level am I certifying to?*"

- Updated public documentation: http://aka.ms/dbcompat

Microsoft stands by DB Compat based certification

Microsoft Database Compatibility Level Protection

Full Functional protection once assessment tools runs clean with no errors.

Query Plan shape protection on comparable hardware.

Maintaining backward compatibility is very important to SQL Server team.

* More on this this Saturday @3pm in session “Modernizing your SQL Server the right way”

Compatibility Level

Product	Database Engine Version	Compatibility Level Designation	Supported Compatibility Level Values
SQL Server 2019	15	150	150, 140, 130, 120, 110, 100
SQL Server 2017 (14.x)	14	140	140, 130, 120, 110, 100
Azure SQL Database single database/elastic pool	12	130	150, 140, 130, 120, 110, 100
Azure SQL Database managed instance	12	130	150, 140, 130, 120, 110, 100
SQL Server 2016 (13.x)	13	130	130, 120, 110, 100
SQL Server 2014 (12.x)	12	120	120, 110, 100
SQL Server 2012 (11.x)	11	110	110, 100, 90
SQL Server 2008 R2	10.5	100	100, 90, 80
SQL Server 2008	10	100	100, 90, 80
SQL Server 2005 (9.x)	9	90	90, 80
SQL Server 2000	8	80	80

```
SELECT name, compatibility_level FROM sys.databases;
```

Remarks

For all installations of SQL Server, the default compatibility level is set to the version of the Database Engine.

Databases are set to this level unless the **model** database has a lower compatibility level.

When a database is upgraded from any earlier version of SQL Server, the database retains its existing compatibility level, if it is at least minimum allowed for that instance of SQL Server.

Upgrading a database with a compatibility level lower than the allowed level, automatically sets the database to the lowest compatibility level allowed.

This applies to both system and user databases.

Compatibility Levels and SQL Server Upgrades

Database compatibility level is a valuable tool to assist in database modernization, by allowing the SQL Server Database Engine to be upgraded, while keeping connecting applications functional status by maintaining the same pre-upgrade database compatibility level.

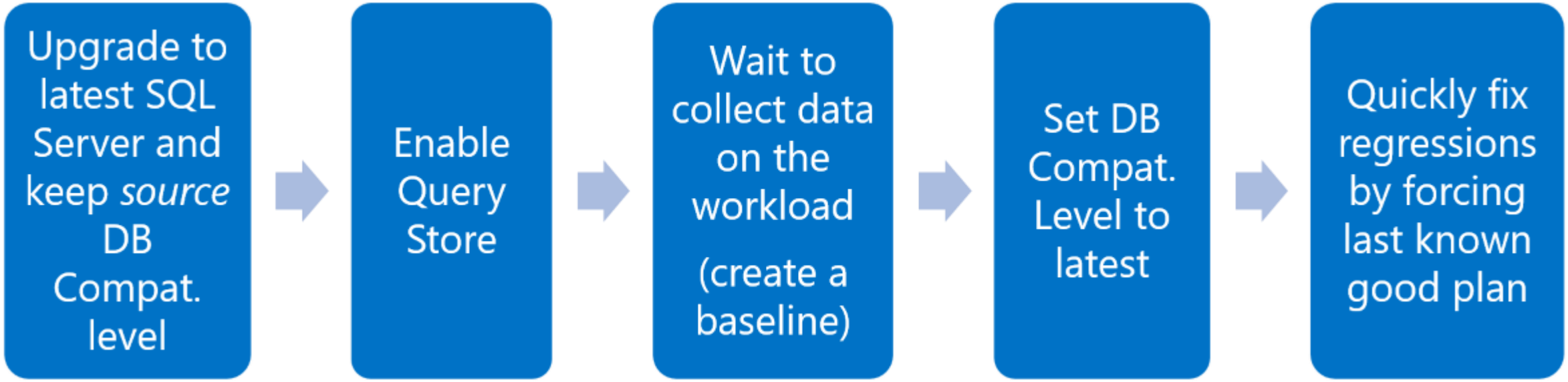
As long as the application does not need to leverage enhancements that are only available in a higher database compatibility level, it is a valid approach to upgrade the SQL Server Database Engine and maintain the previous database compatibility level. For more information on using compatibility level for backward compatibility, see the Using Compatibility Level for Backward Compatibilitylater in this article.

Using Compatibility Level for Backward Compatibility

The *database compatibility level* setting affects behaviors only for the specified database, not for the entire server. Database compatibility level provides only partial backward compatibility with earlier versions of SQL Server.

Starting with compatibility mode 130, any new query plan affecting features have been intentionally added only to the new compatibility level. This has been done in order to minimize the risk during upgrades that arise from performance degradation due to query plan changes. From an application perspective, the goal should still be to upgrade to the latest compatibility level at some point in time, in order to inherit some of the new features, as well as performance improvements done in the query optimizer space, but to do so in a controlled way. Use the lower compatibility level as a safer migration aid to work around version differences, in the behaviors that are controlled by the relevant compatibility level setting. For more details, including the recommended workflow for upgrading database compatibility level

Change the Database Compatibility Level and use the Query Store

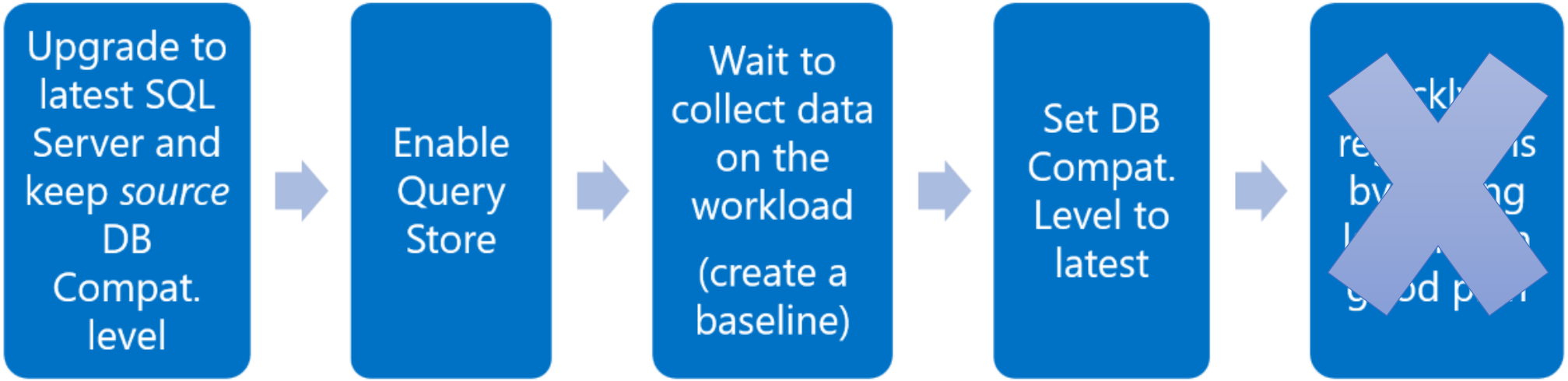


SSMS 18. Query Tuning Assistant

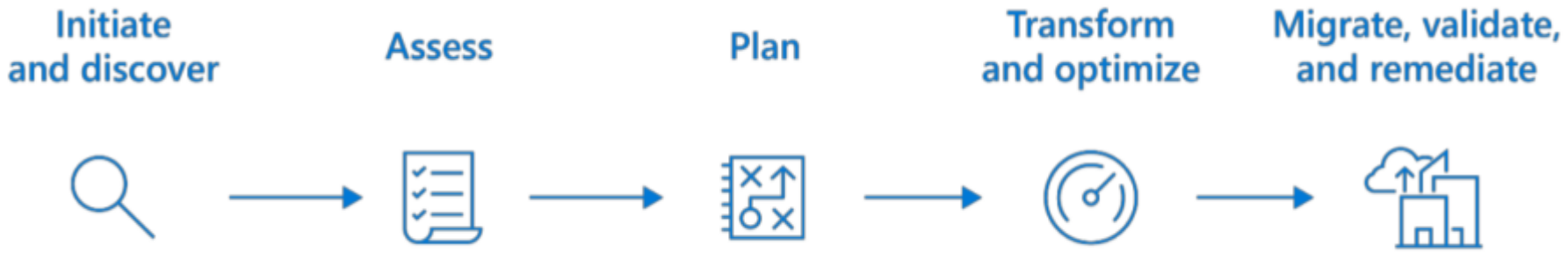
Query Tuning Assistant (QTA)

Crucial to uncover query performance issues with the workload, as it runs on the newer version of SQL Server Database Engine.

User needs to follow documented DB Compatibility upgrade procedure, QTA will guide through steps.



Data Migration Roadmap

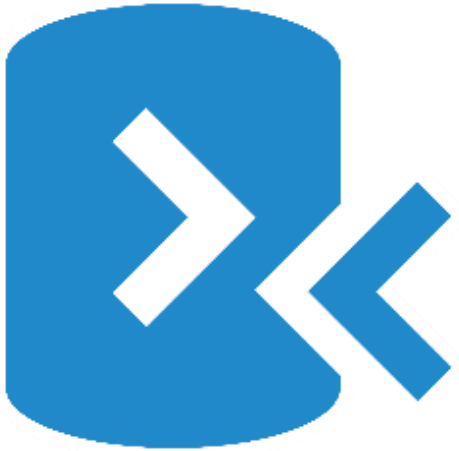


Data Migration Roadmap

- Initiate and discover
  - Understand your database footprint and potential approaches to migration
- Assess
  - Assess the discovered workload requirements and any dependencies
- Plan
  - Plan and describe the workloads to be migrated, the tool to be used for migration and the target platform for the workload
- Transform and optimize
  - Transform any workloads not currently compatible with modern data platforms. Optimize workloads to take advantage of new features
- Migrate, validate and remediate
  - Perform migration, validate successful migration, and remediate applications where required

Migration Tools & Services

Assess → Migrate



Data Migration Assistant

Rich assessments at scale

Feature recommendations

Schema conversions

Azure Database Migration Service

MS and non-MS source support

Built for scale and reliability

Built with enterprise security and privacy

MAP Toolkit

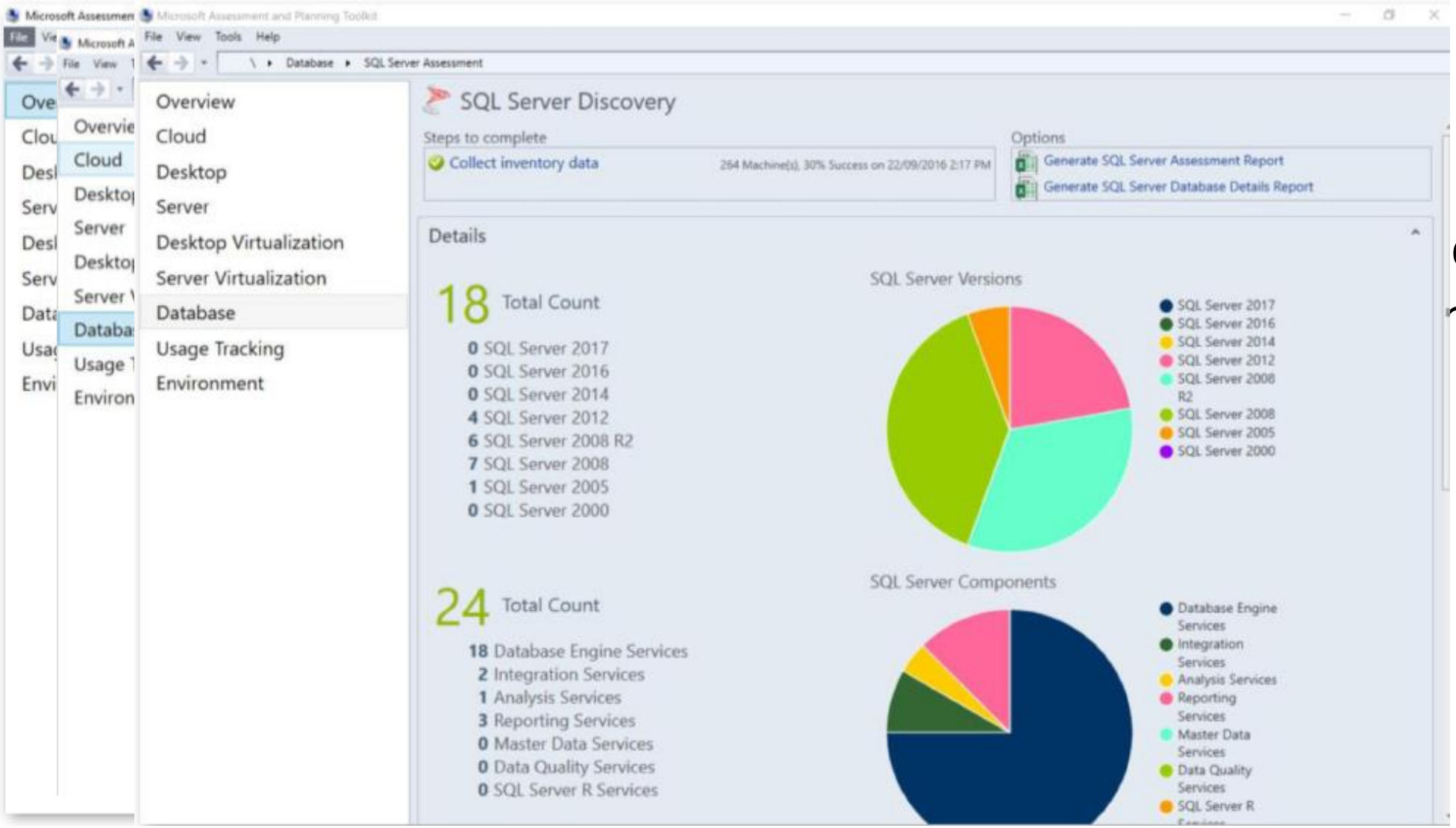
Free

The

appl

d

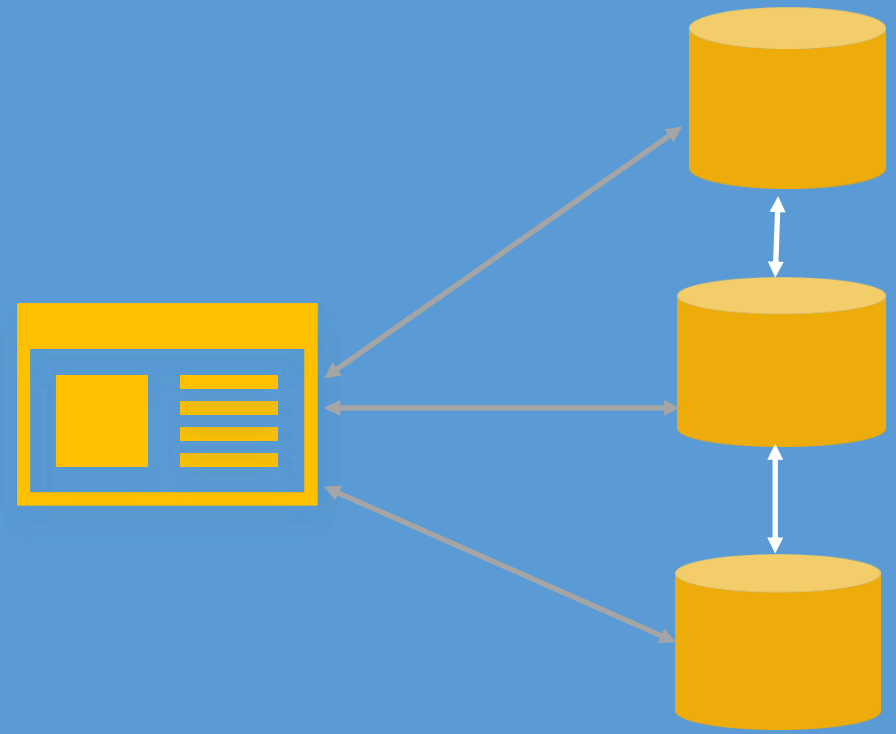
ns.



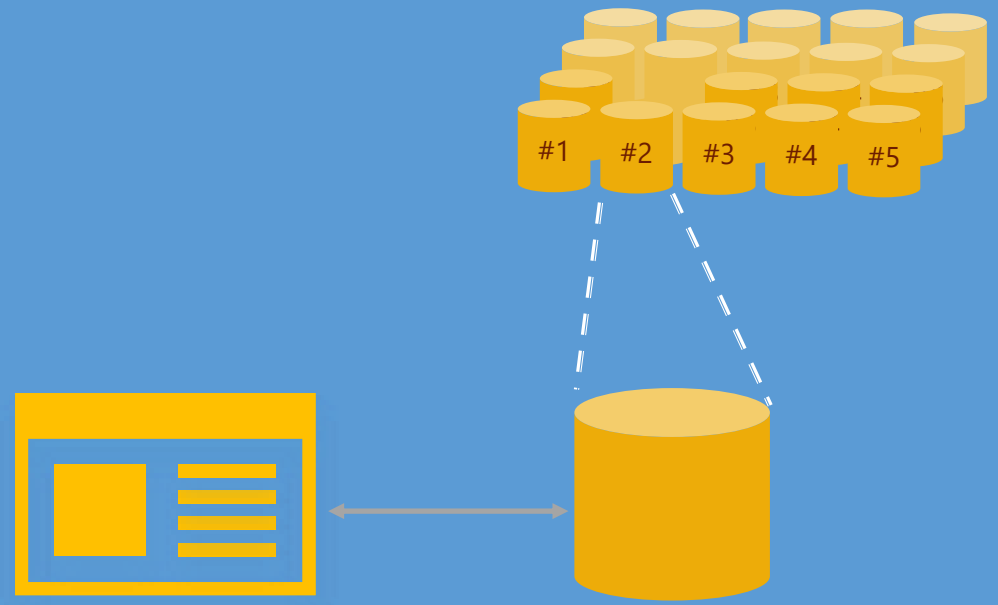
Application pattern

Discover → Assess → Convert

Legacy - Distributed

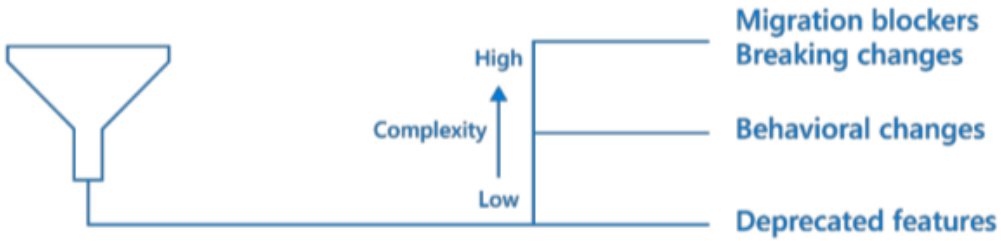


SOA - Single / Multi-Tenant



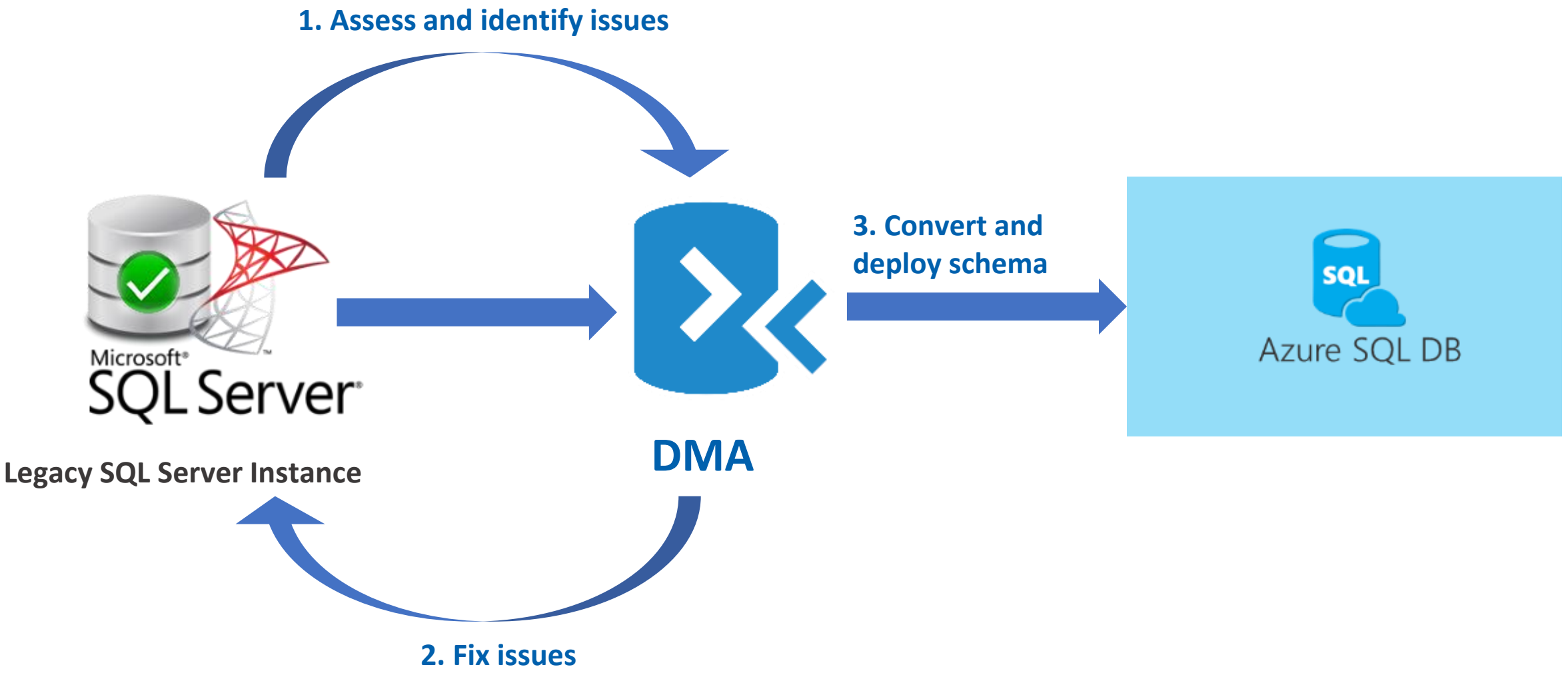
Asses and Convert

- The migration blockers
  - A migration will not be able to proceed until these issues are resolved.
- Breaking changes
  - A migration will be able to proceed but the workload will need to be fixed post-migration to be functional.
- Features to leverage
  - Available Azure features that when utilized can maximize the benefit of migrating to Azure services.
- Effort involved to fix issues
  - An estimate of the time and processes required to rectify the above highlighted issues.
  - To realize these goals, a closer examination of workloads is done with emphasis placed on the following areas:
    - Assess workloads for migration
    - Assess workload criteria
    - Assess database using Data Migration Assistant



Assess and Convert

Discover → Assess → Convert



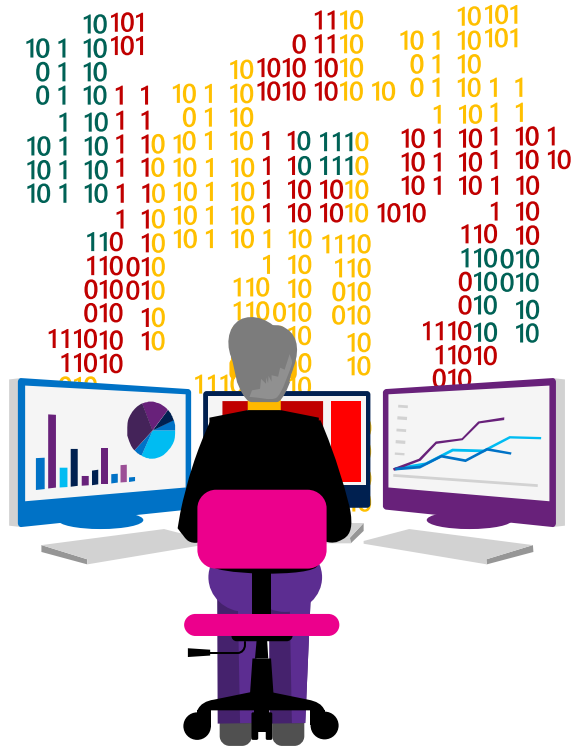
Assessment Recap

SQL PaaS Unsupported Data Types : NTEXT and TEXT columns needed to be converted to nvarchar(max)

External Tables : Cross database joins are not supported. Create external tables and remove the 3 part references(e.g. [dbo].[database].[table]) to make the schema compatible.

UTC Dates : Custom date function using AT TIME ZONE in all table defaults, Stored Procedures, functions etc.. to report time in regional format.

SQL Agent Jobs : Need to be converted to elastic Jobs or hosted in IaaS SQL.



NTEXT -> varchar

```
-- Change TEXT to VARCHAR(MAX)
ALTER TABLE billing_schedule ALTER COLUMN description varchar(max)
NULL
ALTER TABLE billing_schedule ALTER COLUMN notes varchar(max) NULL
ALTER TABLE contract ALTER COLUMN notes varchar(max) NULL
ALTER TABLE contract ALTER COLUMN warnings varchar(max) NULL
```

Custom date function

```
CREATE FUNCTION getlocaldate()
returns DATETIME
as
begin
     DECLARE @D AS datetimeoffset
     SET @D = CONVERT(datetimeoffset, getdate()) AT TIME ZONE 'Eastern
Standard Time’;
SELECT CONVERT(datetime, @D);
end
```

Cross database transactions

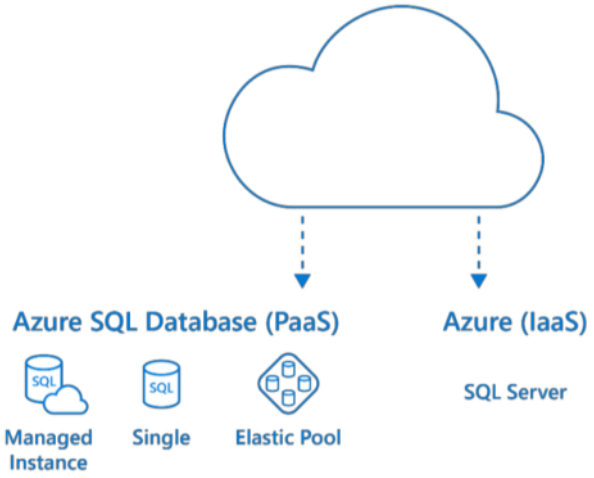
```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'strongpassword';
GO
CREATE DATABASE SCOPED CREDENTIAL [ebhDataServiceCred] WITH IDENTITY = 'username', SECRET =
'strongpassword';
GO
CREATE EXTERNAL DATA SOURCE [ebhDataService] WITH
(TYPE = RDBMS,
LOCATION = 'yourserver.database.windows.net',
DATABASE_NAME = 'YourExternalDatabase',
CREDENTIAL = ebhDataServiceCred,
)
GO

EXEC sp_execute_remote @data_source_name = N'ebhDataService', @stmt = @cmd;
GO
```

Plan

The goal is to establish two key things for each workload

- Target platform
  - This is the final location for each workload.
- One-time migration versus continuous sync
  - A one-time migration means that a workload can be taken offline, whereas a continuous sync means that the workload source database needs to be available during the migration.
    - Plan the target platform
    - How to choose the right target platform
    - Migrating SSAS, SSIS and SSRS to Azure or another platform
    - Plan the migration tool
    - Target platform selection examples



Transform and Optimize

- With a solid idea now of what is being migrated, where its migrating to, and how that will be achieved, the next step is to transform and optimize and make any required changes to the source environment to prepare for the upcoming migration phase. After completing the transform and optimize phase we will have
  - Schema compatible with target
    - The database schema will be in a supported state for the target platform and ready to migrate
  - Preparations complete for data migration
    - All errors will be rectified, and data is ready to be moved. Throughout the remainder of this section we will investigate how to transform the source data or mechanisms used to fix any issues and look into possible optimizations that can be made to take full advantage of the Azure SQL platform or SQL Server on-premises.

Transformation

- Update and check database schemas
- Implement any version upgrade requirements for the environment
- Remediation of any errors or warnings provided by the migration assessment tools
- Migrate existing integrated database services into Azure
- Handling SSIS workloads in the cloud

Optimization

- Assess what new features may be available on the target platform
- Re-structure workloads into more cost effective or performance effective sets
- Ensure workloads are right-sized

Migrate, Validate and Remediate

- Consider the maintenance windows that are available to the application and database targeted for migration
  - If they are critical workloads, they may only be able to go offline for a few minutes at a very specific point in time. Alternatively, a workload may be used for historical reporting purposes and can easily be taken offline most days of the week without impacting end-users. These differences will help decide which migration technique needs to be used.
- Start with low priority databases first
  - This can help ensure the migration process works and help gauge how long the migration is likely to take when you get to your more critical workloads.

Migrate, Validate and Remediate

- Fix compatibility issues outlined in Data Migration Assistant
  - These issues should be fixed during the Transform and Optimize phase, but validate that DMA no longer presents any remaining issues.
- Run a test migration with chosen tool
  - Before migrating the database, run a test migration of the database to confirm the amount of time the migration will take, and any issues encountered during the migration process.
- Test database for issues
  - When the test migration completes, perform validation steps to confirm that the data is migrated in full and check for any issues encountered on the Azure SQL platform.

Migrate, Validate and Remediate

- Repeat issue fixes until the database is fixed
  - For each issue discovered during testing, find a fix, and then retest. Keep repeating this test-fix cycle until all issues have been found and repaired.
- Re-write third-party applications for the cloud as needed
  - Third party applications may benefit from the Azure Application Architecture Guide as it discusses older onpremises versus cloud architecture models that could help optimize performance and decrease overhead with leaner coding approaches. Each application should be analyzed on a case by case basis to see if a lift and shift or a re-write in necessary.
- Test third-party applications
  - Confirm that any third-party applications will still function as expected in the cloud as each application is moved, including any dependencies.

Migrate, Validate and Remediate

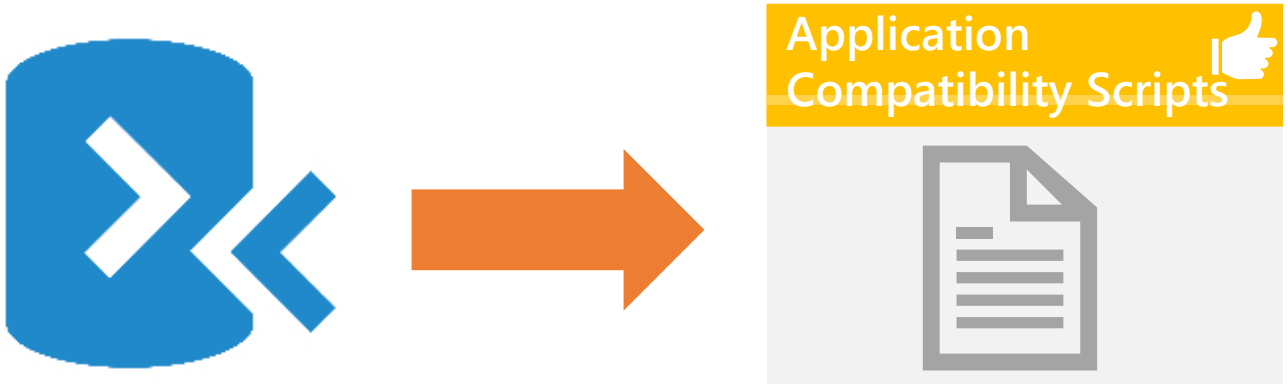
- Take old databases and application offline
  - Remember to take the source database and application offline before starting the migration process to avoid confusion and preserve the original data in case there is a need to refer to them or perform a roll-back.
- Create new disaster recovery and maintenance plans
  - Take the time to update your disaster recovery plans, as data has now moved locations and is accessed in a different manner. Consider improving disaster recovery plans by utilizing the geo-replication features of Azure to protect data that may previously have been too complex or costly to protect when on-premises. Maintenance plans will also need to be reviewed as Azure now performs many of those maintenance tasks automatically for you in the
  - background, removing the need to perform them manually.

Migrate, Validate and Remediate

- Understand your workload requirements as a starting point
  - Requirements might include storage size, storage throughput, and high availability.
- Create a plan to mitigate risk associated with downtime and compatibility issues
  - Many of the discussed points in this whitepaper will help to reduce the risk of errors during the migration. Conduct test migrations before doing the final migration by getting to know any errors before getting to more critical workloads and have a rollback plan prepared in case of an emergency.
- Continually iterate on your migration process
  - During the first migrations small changes will be found, documentation or processes will need to be created, or unnecessary migration steps will need to be removed. These findings should be fed back in to the migration process that you are following to optimize the remaining higher priority migrations.

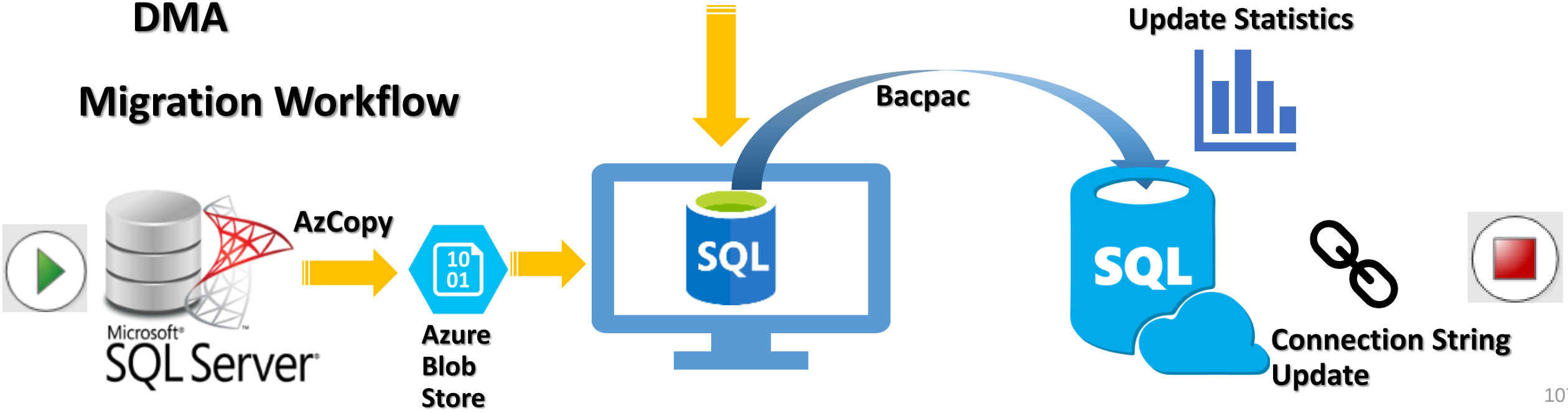
SQL PaaS Migration Workflow

Pre-Migration Tasks

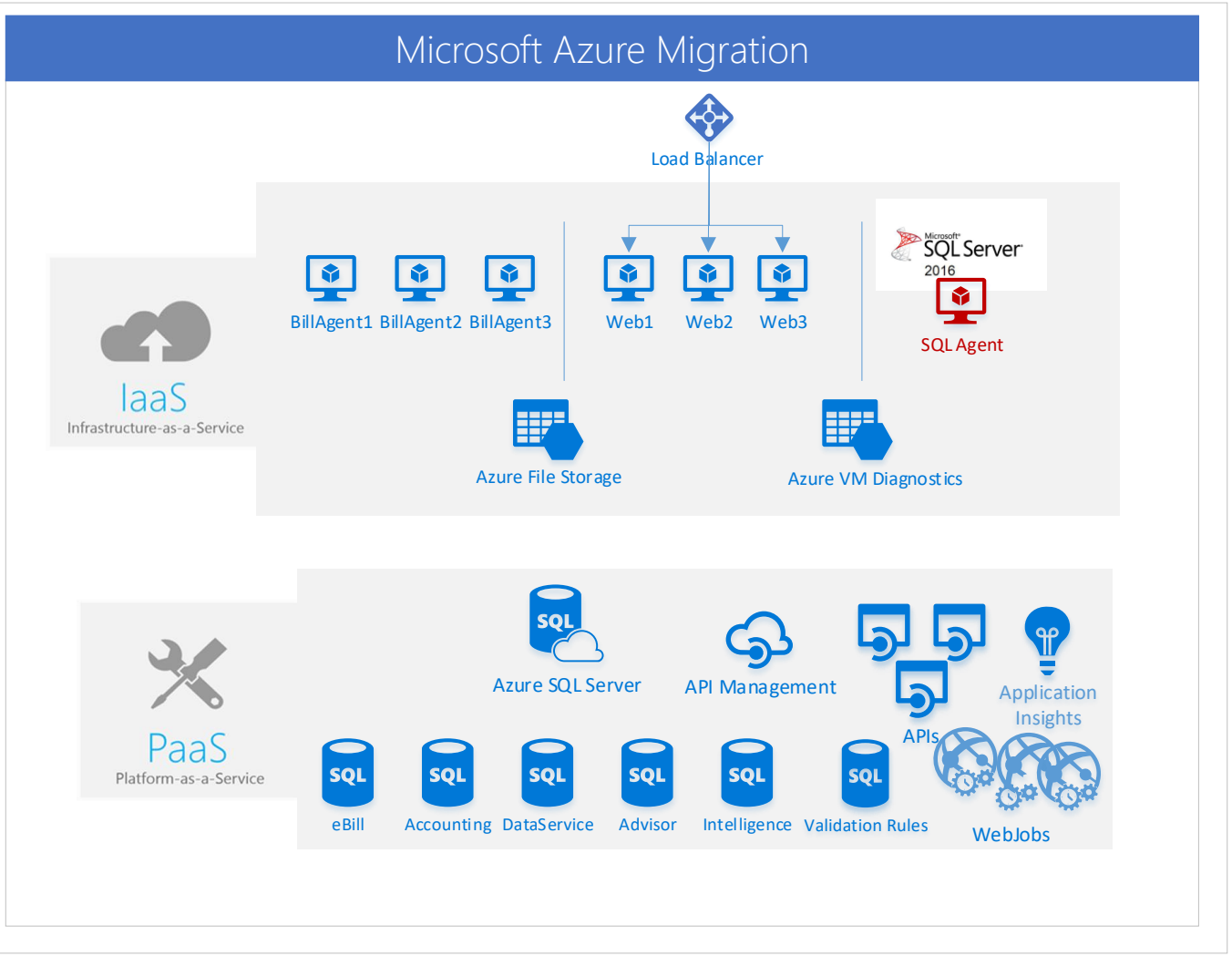


DMA

Migration Workflow



Post Migration Azure Footprint



IaaS : Lift and Shift

- All front end web servers and bill agent processors were migrated to Azure VMs and load balanced.
- VMs share storage with Azure file storage.
- Due to time constraints a new Azure VM with SQL Server was also provisioned to host SQL Agent Jobs.

PaaS : SQL Migration

- All SQL Databases were migrated to Azure SQL PaaS instances.
- Single Azure SQL Server hosts all databases in an elastic pool.

eBillingHub : SQL PaaS Migration Challenges

Application compatibility

Discover, Assess and Refactor App and SQL

Migration effort

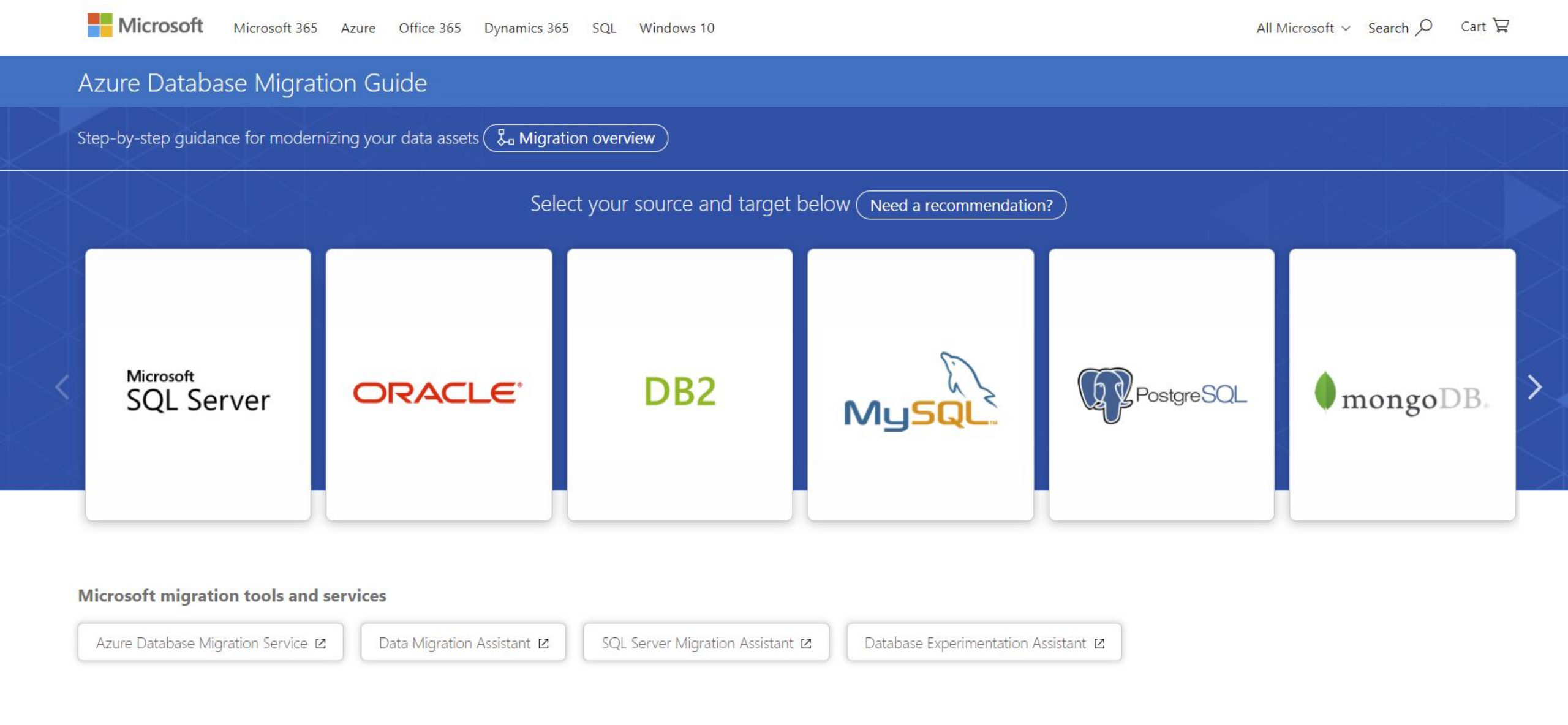
4+ Months of Engineering Time

Application downtime

48 Hours…

WAS NOT EASY…

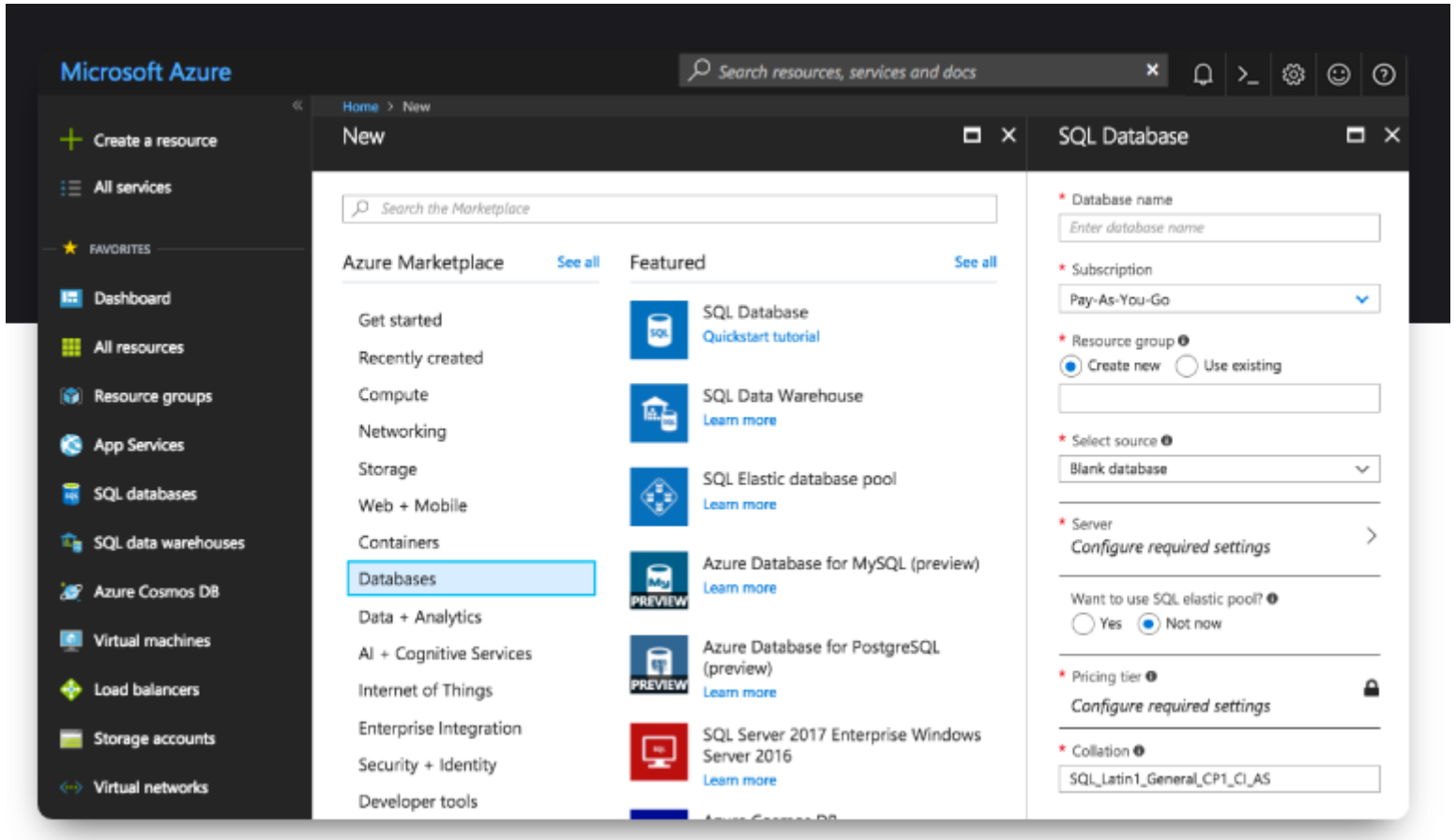
https://datamigration.microsoft.com/



What are your next steps?

1. Assess your SQL Server instances and databases
2. Are you affected by EOS for SQL Server 2008 and/or SQL Server 2008 R2?
3. Look at modernization choices
4. Try out SQL Server 2017, SQL Server 2019, Linux, and Containers
5. Do a Proof of Concept (POC)
6. Build out a semester and year plan. Adjust each semester
7. Discuss database compatibility certification with your developers or ISV
8. What do you want to move to the cloud?
9. Where are you biggest pain points?
10. Modernize your skills

Microsoft SQL Server on Azure Virtual Machines or use Azure SQL Database, a fully managed service



- As SQL Server 2008 and SQL Server 2008 R2 rapidly approach the end of Extended support from Microsoft on July 9, 2019, and with SQL Server 2014 also falling out of Mainstream support on July 9, 2019 (joining SQL Server 2012, which fell out of Mainstream support on July 11, 2017) , Organizations are migrating to a modern version of SQL Server.

- Modern version of SQL Server as SQL Server 2016 or later.

- SQL Server 2016 and newer actually have many useful new features that make them much better products than their predecessors.

- Migrating to a modern version of SQL Server also usually means using new, faster hardware and storage, running on a current version of Windows Server, which is also very beneficial, as long as you choose your new hardware and storage wisely.

- Despite all of this, there are a number of cases where organizations have migrated from a legacy version of SQL Server to a modern version of SQL Server on new hardware and a new operating system, and then be unpleasantly surprised by performance regressions once they are in Production.

- How can these performance regressions be occurring, and what steps can you take to help prevent them?

- The main culprit in most of these performance regressions is a combination of lack of knowledge, planning, and adequate performance testing.

- Unlike legacy versions of SQL Server, modern versions of SQL Server have several important performance-related configuration options that you need to be aware of, understand, and **actually test** with your workload.

- Most people who run into performance regressions have done what I call a “blind migration” where they simply restore their databases from the older version to the new version of SQL Server, with no meaningful testing of the performance effects of these different configuration options.

- What are these key configuration options that you need to be concerned with from a performance perspective?
- The most important ones include your database compatibility level
- The cardinality estimator version that you are using
- Your database-scoped configuration options
- What trace flags you are using

- Since SQL Server 2014, the database compatibility level affects the default cardinality estimator that the query optimizer will use

- Since SQL Server 2016, the database compatibility level also controls other performance related behavior by default.

- You have the ability override many of these database compatibility level-related changes with database scoped configuration options and query hints.

- There are actually a rather large number of different combinations of settings that you have to think about and test.

- What are you supposed to do?

Microsoft Database Experimentation Assistant

- In my ideal scenario, you would use the free Microsoft Database Experimentation Assistant (DEA) to capture a relevant production workload.
- This involves taking a full production database backup, then capturing a production trace that covers representative high priority workloads.
- While this is going on, I would run some of my SQL Server Diagnostic Queries to get some baseline metrics from your legacy instance.

Once you have done that, you can then restore that backup to your new environment, and replay the production trace multiple times in your new environment. Each time you do this (which also includes a fresh restore from that original full production database backup), you will use a different combination of these key configuration settings. You have to make the database configuration/property changes ***after*** each restore, but ***before*** you replay the DEA trace

Avoiding SQL Server Upgrade Performance Issues

By: Glenn Berry
Posted on: February 5, 2019 11:25 am

As SQL Server 2008 and SQL Server 2008 R2 rapidly approach the end of Extended support from Microsoft on July 9, 2019, and with SQL Server 2014 also falling out of Mainstream suppor on July 9, 2019 (joining SQL Server 2012, which fell out of Mainstream support on July 11, 2017)), I am seeing an increasing number of organizations that have been migrating to a moder version of SQL Server. I define a modern version of SQL Server as SQL Server 2016 or later.
I see this as a positive development overall, since SQL Server 2016 and newer actually have many useful new features that make them much better products than their predecessors. Migrating to a modern version of SQL Server also usually means using new, faster hardware and storage, running on a current version of Windows Server, which is also very beneficial, as long as you choose your new hardware and storage wisely.
Despite all of this, I have seen a decent number of cases where organizations have migrated from a legacy version of SQL Server to a modern version of SQL Server on new hardware and a new operating system, and then be unpleasantly surprised by performance regressions once they are in Production. How can these performance regressions be occurring, and what steps ca you take to help prevent them?
The main culprit in most of these performance regressions is a combination of lack of knowledge, planning, and adequate performance testing. Unlike legacy versions of SQL Server, moderı versions of SQL Server have several important performance-related configuration options that you need to be aware of, understand, and **actually test** with your workload. Most people whc run into performance regressions have done what I call a "blind migration" where they simply restore their databases from the older version to the new version of SQL Server, with no meaningful testing of the performance effects of these different configuration options.
So, what are these key configuration options that you need to be concerned with from a performance perspective? The most important ones include your database compatibility level, the cardinality estimator version that you are using, your database-scoped configuration options, and what trace flags you are using. Since SQL Server 2014, the database compatibility level affects the default cardinality estimator that the query optimizer will use. Since SQL Server 2016, the database compatibility level also controls other performance related behavior by defau
I have written more about this subject here:
The Importance of Database Compatibility Level in SQL Server
Compatibility Levels and Cardinality Estimation Primer

You have the ability override many of these database compatibility level-related changes with database scoped configuration options and query hints. There are actually a rather large number of different combinations of settings that you have to think about and test. So, what are you supposed to do?

Microsoft Database Experimentation Assistant

In my ideal scenario, you would use the free Microsoft Database Experimentation Assistant (DEA) to capture a relevant production workload. This involves taking a full production database backup, then capturing a production trace that covers representative high priority workloads. While this is going on, I would run some of my SQL Server Diagnostic Queries to get some baseline metrics from your legacy instance.
Once you have done that, you can then restore that backup to your new environment, and replay the production trace multiple times in your new environment. Each time you do this (whicl also includes a fresh restore from that original full production database backup), you will use a different combination of these key configuration settings. You have to make the database configuration/property changes ***after*** each restore, but ***before*** you replay the DEA trace.
The idea here is to see which combination of these configuration settings yields the best performance with your workload. Here are some relevant, likely combinations:

- Use the default native database compatibility level of the new version
- Use the default native database compatibility level of the new version and use the query optimizer hotfixes database-scoped configuration option
- Use the default native database compatibility level of the new version and use the legacy cardinality estimator database-scoped configuration option
- Use the default native database compatibility level of the new version and use the legacy cardinality estimator database-scoped configuration option and use the query optimizer hotfixes database-scoped configuration option
- Use the existing database compatibility level of the old version
- Use the existing database compatibility level of the old version and use the query optimizer hotfixes database-scoped configuration option

This level of DEA testing may not be practical if you have a large number of databases, but you should really try to do it on your most mission critical databases. Barring that, I would try to do as much testing of your most important stored procedures and queries as possible, using these different configuration settings.
Finally, if no adequate testing is possible you can follow Microsoft's recommended upgrade sequence (in your new production environment, after you go live), which is:

Questions ?

Migrating our old SQL Server to the modern platform


javier.ignacio.villegas@gmail.com


@javier_vill


/javiervillegas


sql-javier-villegas.blogspot.com.ar

Thank you!!

Gracias !!

SQL ARGENTINA

PASS

MUG

vopen.tech

PASS MARATHON

SPEAKER IDOL PASS

24 HOURS OF PASS

PASS GLOBAL SPANISH VIRTUAL CHAPTER

Microsoft AI Tour Argentina

GroupBy

2019 Global Azure BOOTCAMP

PASS SQLSATURDAY

DBA FUNDAMENTALS VIRTUAL GROUP

DEV DAY ARGENTINA

GLOBAL AI COMMUNITY ON VIRTUAL TOUR

POWERBI DAYS

@javier_vill javiervillegas