



# Introduction to Managed Instances

What, Why, How!

John Q. Martin, Data Platform MVP

PLATINUM SPONSOR



GOLD SPONSORS



SILVER SPONSORS



BRONZE SPONSORS





Image is in public domain:  
<https://www.publicdomainpictures.net/en/view-image.php?image=260306&picture=gdpr>



# John Q. Martin

## Bearded Data Platform Guy



[/johnqmartin](#)



[@SQLDiplomat](#)

### Data Platform MVP

Helping people understand many aspects of the Microsoft Data Platform and Azure.

### Learned a lot over the years

Working with Data Platform for over a decade, learning how not to use it as much as how to. Some lessons were learned the hard way!

### Ridden the Himalayas

Ridden over three of the highest motorable passes in the world on a Royal Enfield motorcycle through the Himalayas in India.

# Agenda

An agenda slide is highly recommended so attendees understand what you will be presenting and to minimize session hopping.

- What are Managed Instances?
- Why Should I Use Them?
- When should I Use Them?
- How do I migrate to them?



# What are Managed Instances?

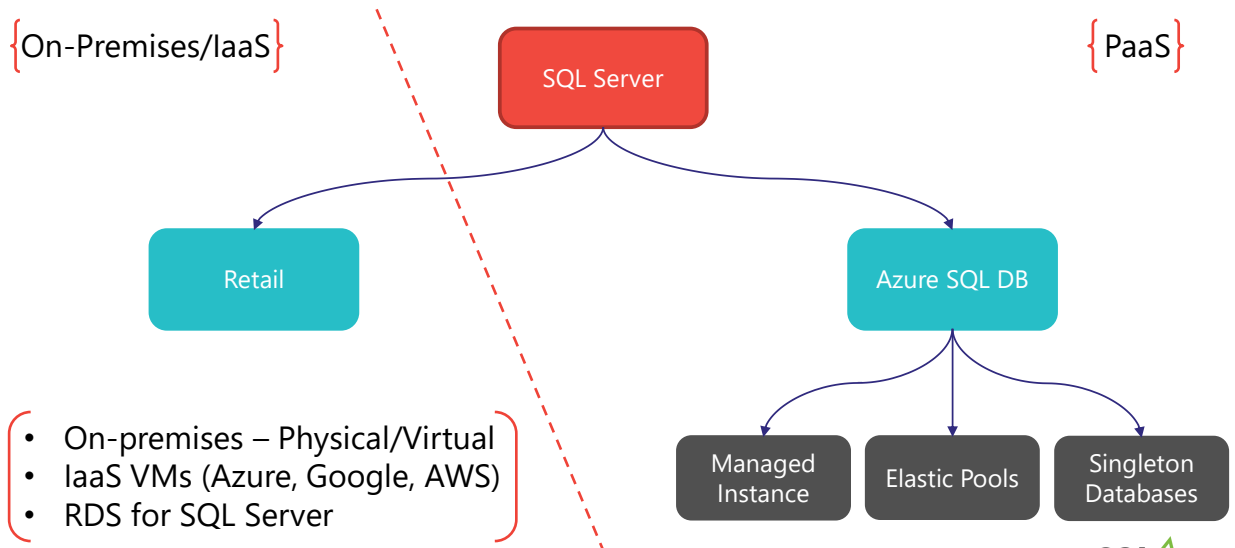
- Platform as a Service (PaaS) SQL Server Engine with a scope at the Instance Level.
- But we have Azure SQL Database Already...
  - No integrated Agent
  - No native Cross Database Queries
  - No control of files and filegroups
  - No Service Broker



# What are Managed Instances?

{ On-Premises/IaaS }

{ PaaS }



**SQL**  
RELAY

# Managed Instance Compared

	Managed Instance	Azure SQL DB	Elastic Pools
Scheduled Activity	<ul style="list-style-type: none"><li>• SQL Agent</li></ul>	<ul style="list-style-type: none"><li>• Elastic Database Jobs</li></ul>	<ul style="list-style-type: none"><li>• Elastic Database Jobs</li></ul>
Cross DB Queries	<ul style="list-style-type: none"><li>• Native Capability</li></ul>	<ul style="list-style-type: none"><li>• Elastic Query</li></ul>	<ul style="list-style-type: none"><li>• Elastic Query</li></ul>
Feature Scope	<ul style="list-style-type: none"><li>• Instance</li></ul>	<ul style="list-style-type: none"><li>• Database</li></ul>	<ul style="list-style-type: none"><li>• Database</li></ul>
Compute Measure	<ul style="list-style-type: none"><li>• vCore</li></ul>	<ul style="list-style-type: none"><li>• DTU/vCore</li></ul>	<ul style="list-style-type: none"><li>• DTU/vCore</li></ul>
Database Limit	<ul style="list-style-type: none"><li>• 100*</li></ul>	<ul style="list-style-type: none"><li>• Single Database</li></ul>	<ul style="list-style-type: none"><li>• Varies by tier</li></ul>
vNet Connectivity	<ul style="list-style-type: none"><li>• Native</li></ul>	<ul style="list-style-type: none"><li>• Via Service Endpoint</li></ul>	<ul style="list-style-type: none"><li>• Via Service Endpoint</li></ul>



Elastic Pools Limits Documentation - <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-vcore-resource-limits-elastic-pools>

The maximum number of databases on a Managed Instance has a hard limit at 100, however the actual number you can deploy can be below this if you have databases with multiple data files or the storage limit is reached by the volume of the data stored. Reference - <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance-resource-limits>



# Managed Instance Compared

	Managed Instance	Retail SQL Server	RDS for SQL Server
Scheduled Activity	<ul style="list-style-type: none"> <li>SQL Agent</li> </ul>	<ul style="list-style-type: none"> <li>SQL Agent</li> </ul>	<ul style="list-style-type: none"> <li>SQL Agent</li> </ul>
Cross DB Queries	<ul style="list-style-type: none"> <li>Native Capability</li> </ul>	<ul style="list-style-type: none"> <li>Native Capability</li> </ul>	<ul style="list-style-type: none"> <li>Native Capability</li> </ul>
Feature Scope	<ul style="list-style-type: none"> <li>Instance</li> </ul>	<ul style="list-style-type: none"> <li>Instance</li> </ul>	<ul style="list-style-type: none"> <li>Database</li> </ul>
Compute Measure	<ul style="list-style-type: none"> <li>vCore</li> </ul>	<ul style="list-style-type: none"> <li>Core Count</li> </ul>	<ul style="list-style-type: none"> <li>Core Count</li> </ul>
Database Limit	<ul style="list-style-type: none"> <li>100</li> </ul>	<ul style="list-style-type: none"> <li>32,767</li> </ul>	<ul style="list-style-type: none"> <li>30</li> </ul>
vNet Connectivity	<ul style="list-style-type: none"> <li>Native</li> </ul>	<ul style="list-style-type: none"> <li>IaaS VM/Hybrid VPN</li> </ul>	<ul style="list-style-type: none"> <li>Native to AWS</li> </ul>



## AWS Instance Constraints -

[https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_SQLServer.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_SQLServer.html)

- 30 Databases
- 16 TB per-instance (1TB for magnetic media)
- MSDB restricted access

# Managed Instance - Key Features

- **Replication**
  - Currently supports Transactional Replication
- **Security Features**
  - Always Encrypted/Dynamic Data Masking/Row Level Security
  - Transparent Database Encryption
  - Transport Layer Security (TLS)
- **Authentication**
  - Azure Active Directory & SQL Authentication



# Managed Instance - Key Features

- HA/DR
  - High Availability built-in
  - Business continuity allows for read-only access to secondary
  - Can backup/restore to Azure Blob Storage
- T-SQL
  - Cross Database Queries
  - Global Temp Tables
  - Linked Servers
  - CLR



# Managed Instance – Architecture

## General Purpose Tier

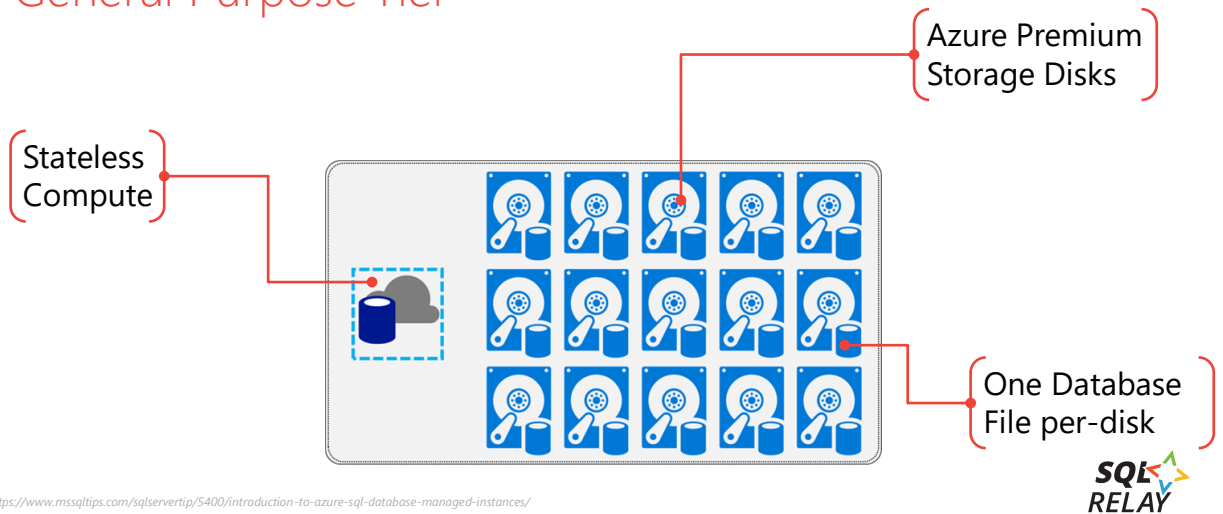


Image source for architecture layout from “Introduction to Azure SQL Database Managed Instances” by John Q. Martin (me) for MSSQLTips - <https://www.mssqltips.com/sqlservertip/5400/introduction-to-azure-sql-database-managed-instances/>

There are a number of scenarios where the number and size of databases will mean that you hit a limit. These are explained well in this blog post from Microsoft - <https://blogs.msdn.microsoft.com/sqlserverstorageengine/2018/08/20/reaching-azure-disk-storage-limit-on-general-purpose-azure-sql-database-managed-instance/>

# Azure Storage Fundamentals

- **Two different types**
  - Standard | Magnetic storage media
  - Premium | SSD based storage
- **Built in resilience**
  - Consistency checking, data replication, multiple replicas, error detection
- **Connectivity**
  - Network attached to compute layer



# Azure Premium Storage Information

Premium Disks Type	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk Size (GB)	128	256	512	1024	2048	4096	8192	16,384	32,767
IOPS per Disk	500	1,100	2,300	5,000	7,500	7,500	12,500	15,000	20,000
Throughput per disk (MB/s)	100	125	150	200	250	250	480	750	750

100 GB File = P10 Disk

500 GB File = P20 Disk

1.5 TB File = P40 Disk

Bigger files will give better throughput, even if they are empty.

Source: <https://docs.microsoft.com/en-us/azure/virtual-machines/windows/premium-storage>

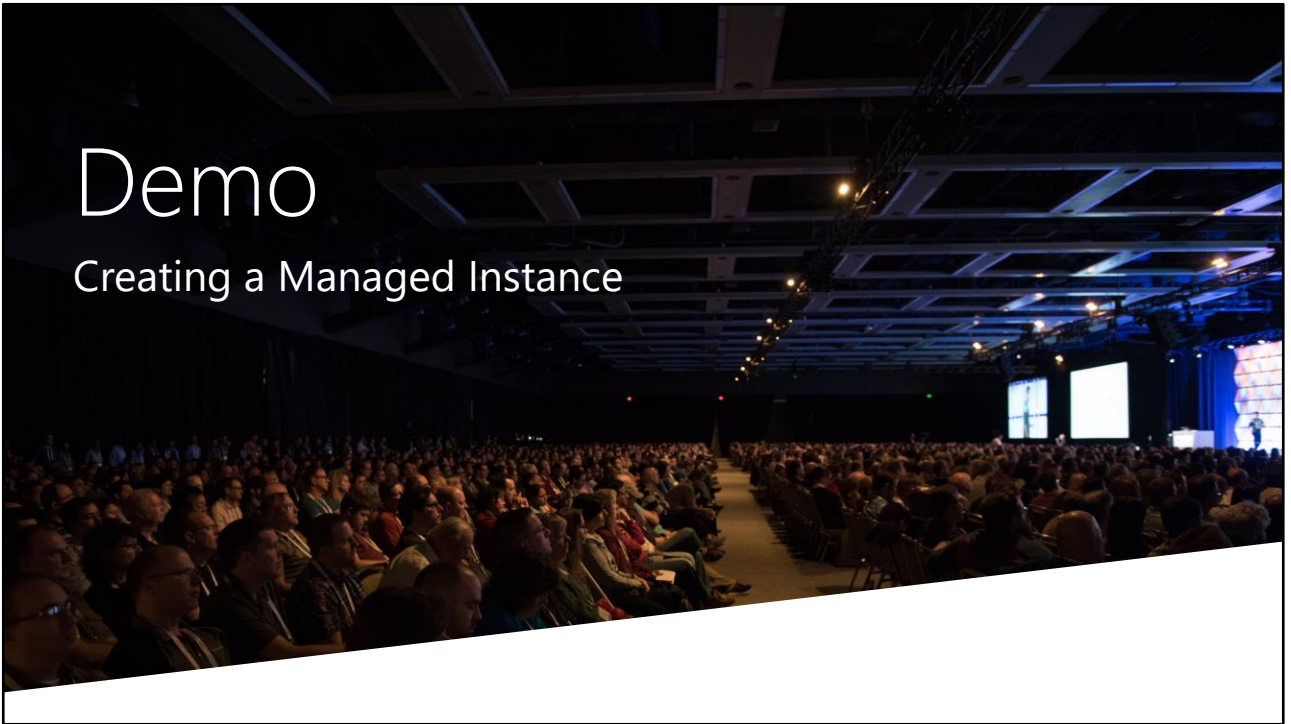


Performance chart for Azure Disks obtained from Microsoft Documentation :  
<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/premium-storage>

Storage performance for Managed Instance relies on the a number of factors, you can find out more information on how to achieve high performance in this blog post from Microsoft -<https://blogs.msdn.microsoft.com/sqlcat/2018/07/20/storage-performance-best-practices-and-considerations-for-azure-sql-db-managed-instance-general-purpose/>

# Demo

## Creating a Managed Instance



### Demo 1 – Azure Portal Experience

-----

1. Open Azure Portal
2. Create Resource Group
3. Create vNet
  1. Make sure vNet has been setup correctly with routing table entry
4. Create Managed Instance
5. Show Managed Instance blade
  1. Show existing one if there is still a delay on the creation of the MI resources
6. Show difference between General Purpose and Business Critical

### Demo 2 – Script Experience

-----

#### Azure CLI Script

---

```
az group create --name "weu-managedinstance-10" --location "West Europe"
```

#### PowerShell Script

---

# Managed Instance Tiers

	General Purpose	Business Critical <i>(Preview)</i>
vCore Count	<ul style="list-style-type: none"><li>• 8, 16, 24 (Gen 4)</li><li>• 8, 16, 24, 32, 40, 64, 80 (Gen 5)</li></ul>	<ul style="list-style-type: none"><li>• 8, 16, 24 (Gen 4)</li><li>• 8, 16, 24, 32, 40, 64, 80 (Gen 5)</li></ul>
Min Storage (GB)	<ul style="list-style-type: none"><li>• 32</li></ul>	<ul style="list-style-type: none"><li>• 32</li></ul>
Max Storage (TB)	<ul style="list-style-type: none"><li>• 8</li></ul>	<ul style="list-style-type: none"><li>• 1 (Gen 4)</li><li>• 4 (Gen 5)*</li></ul>
Memory per-vCore	<ul style="list-style-type: none"><li>• 5 – 7 GB</li></ul>	<ul style="list-style-type: none"><li>• 5 – 7 GB</li></ul>



The maximum size of the available storage for Business Critical will depend on the core count defined for the Managed Instance.

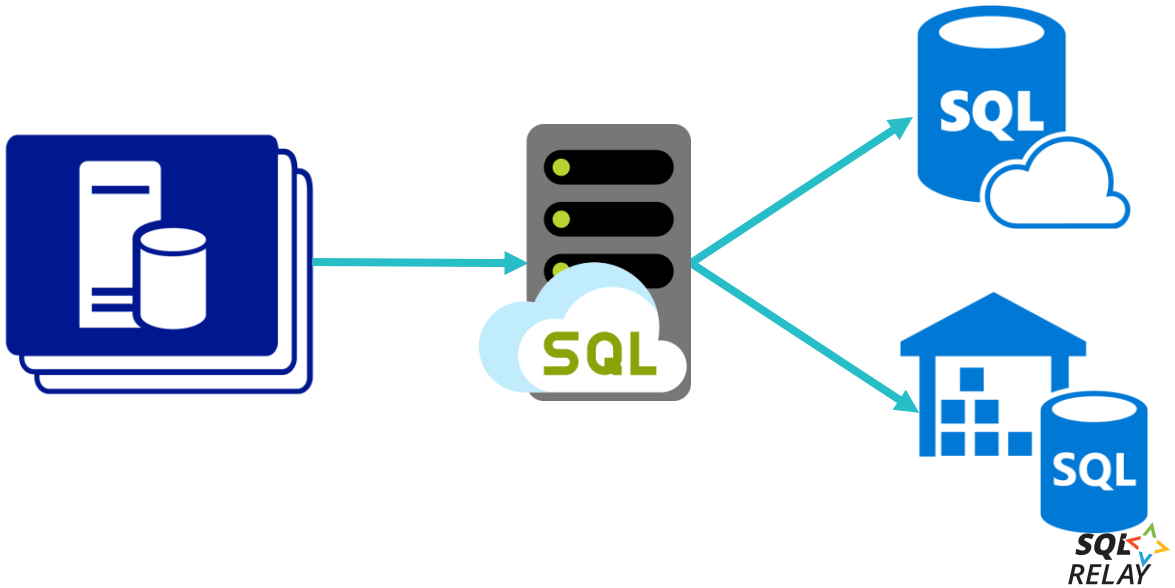
1 TB – 8, 16 vCores

2 TB – 24 vCores

4TB – 30, 40, 64, 80 vCores



# Cloud Migration Lifecycle



# Managed Instance Candidate Workloads

## Vendor Apps

SQL Server back end.

- DB Compatibility!

Needs to support SQL Server Authentication or AAD Authentication.

Where databases have multiple files and filegroups.

## In-House Apps

Can't be easily refactored to use Azure SQL Database.

Refactor to use AAD or fall-back to SQL Authentication.

Control needed for files and filegroups.

Need SQL Server Agent.

## New Cloud Apps

Where older design patterns are used.

Cross database queries that can't use Elastic Query.

- Linked Servers etc.

Need to leverage T-SQL surface area that is not available in SQL DB.



# Managed Instance SQL Workloads

## Vendor Apps

Windows Authentication Only.

> 100 Databases on an existing SQL Server.

Single file > 4TB

## In-House Apps

Filetable/filestream functionality.

> 100 Databases on an existing SQL Server.

Uses the In-Memory Engine

- Not available in General Purpose tier.

## New Cloud Apps

Really small database(s) with low resource use.

Small number of databases involved.

New Cloud Apps should be built on Azure SQL DB...

Seriously, no! Don't do it!!



# Migration Options

- **Backup & Restore to Azure Blob Storage**
  - Use Backup to URL from Retail SQL Server, PowerShell when not available.
  - Roll your own log shipping
- **Transactional Replication**
  - Minimise down-time needed for cut-over.
  - Use a Push Subscriber model for Managed Instance.
- **Azure Data Migration Service (DMS)**
  - Minimise down-time needed for cut-over.
  - Requires vNet connectivity for source, VPN or Express Route to Azure.



# Migration Preparation

- **Trace your Workloads**
  - Understand what features are in use and whether there needs to be updates in order to migrate.
- **Feature Utilization**
  - Is the feature-set in use compatible with those of Managed Instance?
- **Profile and Document the Environment**
  - Identify dependencies on the database tier, are they compatible with Managed Instance.
  - Understand the business needs for migration down-time tolerance.



# Migration Preparation

- **Identify instance scoped artefacts**
  - Migration of Jobs, Credentials, Logins, Linked Servers, etc.
  - Outside of the database these will need to be moved manually.
- **Map the new environment**
  - Placement of application components and connectivity requirements.
  - Hybrid or Cloud post-migration?
- **Is Azure Active Directory (AAD) needed?**
  - Do you need Azure AD Connect configured to sync users etc. to AAD to be able to use AAD Authentication?



One of the key things to understand with the database migration process if you are leveraging AAD for authentication is that the SID for the server level login will not be the same as on-premises. As with database migrations where SQL Authentication is being used, you will need to handle the mismatch in SID between the database user and Instance Login.

1. Rip and replace for Database level security mapping. This is only really viable if leveraging database roles for granular permissions, otherwise it can be difficult to map and then test that user level securables are removed and replaced as they should be.
2. Use ALTER USER to link database user to appropriate login from Azure Active Directory. This will link the database level user to the server login, performing the same functionality that used to be provided by sp\_change\_users\_login.

One of the key elements when it comes to authentication for the Managed Instance is that the application can support using AAD authentication. If this is not the case then there is a need for reverting to SQL Authentication.

Migration of Instance level objects can be performed via SSIS or other open source tools such as DBATools.

## Azure AD Connect

<https://docs.microsoft.com/en-us/azure/active-directory/connect/active-directory-aadconnect>

**DBA Tools**

<https://dbatools.io/>

# Useful Free Tools



Chrissy LeMaire | [@CL](#)



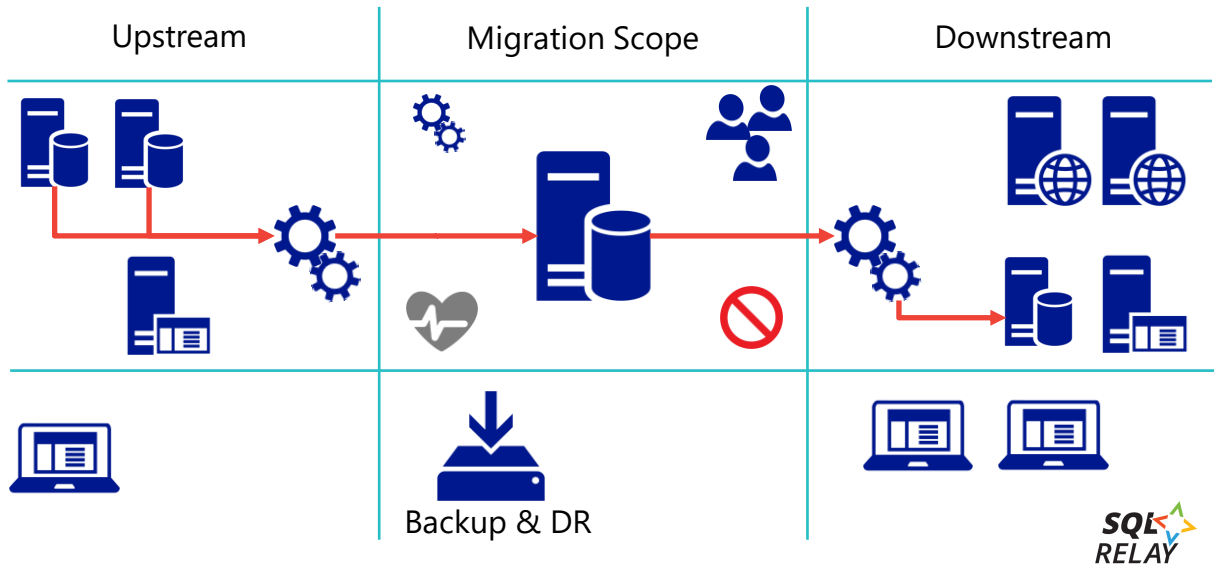
Gianluca Sartori  
[@Spaghettidba](#)

- **DBATools** - <https://dbatools.io>
  - PowerShell module that will help with scripting and automation for migration processes.
- **WorkloadTools** - <http://bit.ly/WorkloadTools>
  - GitHub project from Gianluca Sartori for analysis and replay of workloads.
  - Live replay from one server to another via XE or Trace Capture.
- **RML Utilities Suite**
  - Free tooling from Microsoft that has analysis and replay capabilities.





# Migration Preparation



SQL  
RELAY

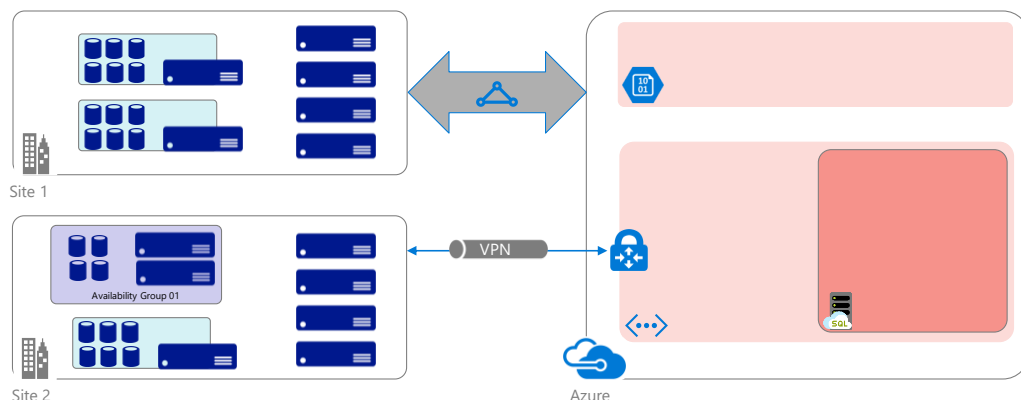
# Migration Preparation

- **Physical Database Architecture**
  - Understand files, filegroups, and log file properties.
- **Identify up-stream and down-stream dependencies**
  - Application connectivity, ETL pipelines, Authentication mechanisms.
- **Instance and external dependencies**
  - What will need to be migrated separately from the database, pre/post migration.
- **Tune workloads before moving them**



# Migrating to Managed Instance

## Backup and Restore



Migration of database elements from on-premises (or IaaS VMs) retail SQL Server to Azure SQL Database Managed Instance necessitates backup to URL from the source. This is then restored to the Managed Instance using the standard `RESTORE DATABASE` command, however there is no need for mapping storage paths etc. using `MOVE` as this is handled natively by the Managed Instance.

Once the database components are migrated to the Managed Instance, the application level components can be migrated if needed. Either creation of new IaaS VMs and installing the software as needed, or migrating existing VMs to Azure using Azure Migration Service.

Where the databases to be migrated are very large, there are options to mitigate lack of bandwidth and time. The Azure Import/Export service allows data to be stored on SATA disks that can then be sent to Microsoft and loaded to Azure Blob Storage. Leveraging this capability as part of the migration plan has a lot of potential to ease the pain of large database migrations.

Another option is to use Transactional Replication to seed the Azure SQL Database Managed Instance ahead of the migration and cut-over. Managed Instance supports being a Transactional Replication Subscriber.

There are many options that can be used to perform the migration, it is important to evaluate the options and select the one that is most appropriate for your environment.

### **Azure Migrate**

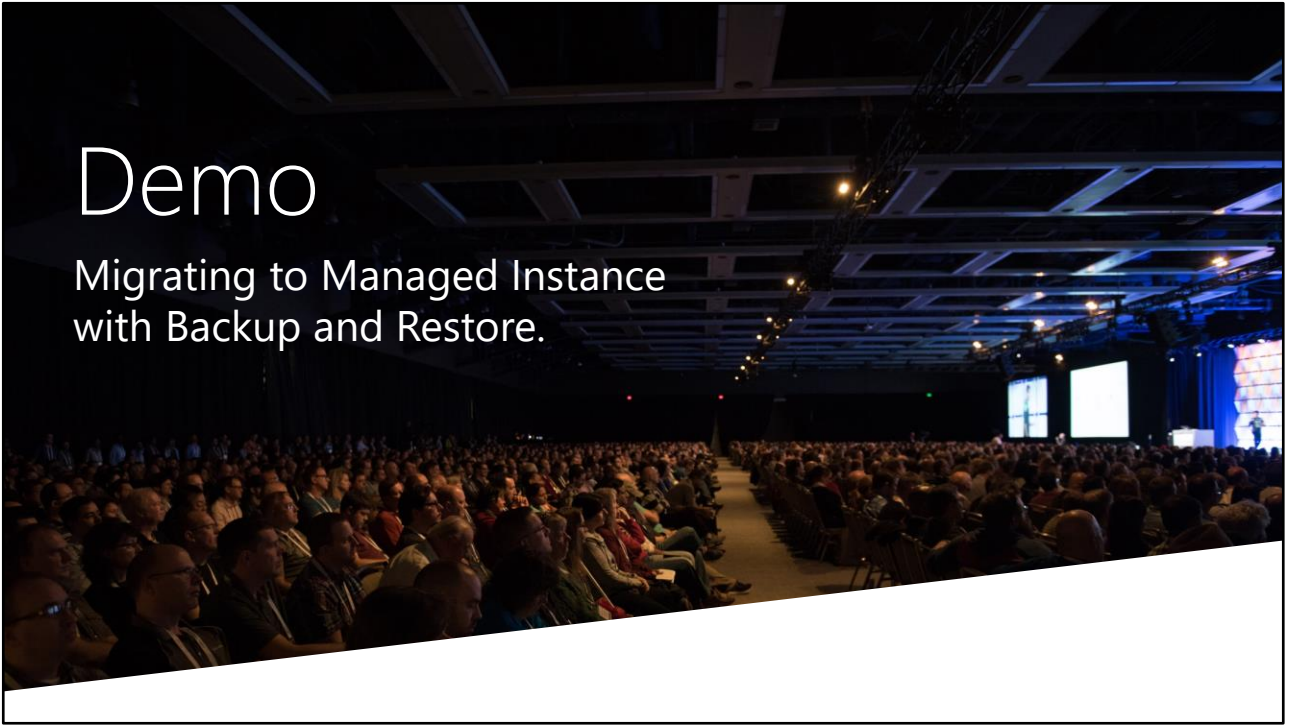
<https://azure.microsoft.com/en-gb/services/azure-migrate/>

**Azure Import/Export Service**

<https://azure.microsoft.com/en-us/services/storage/import-export/>

# Demo

Migrating to Managed Instance  
with Backup and Restore.



# Migrating to Managed Instance

## Transactional Replication

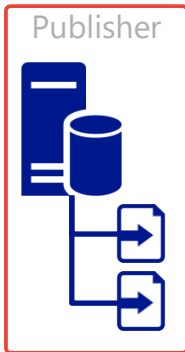
- **Co-locate Publisher and Distributor**
  - Typically have both roles on premises when used to migrate to Managed Instance.
- **Think about bandwidth for initialization**
  - Is it appropriate to use a snapshot or should you use a backup?
- **Instance and external dependencies**
  - What will need to be migrated separately from the database, pre/post migration.



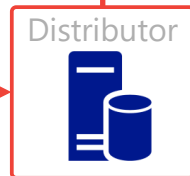
# Migrating to Managed Instance

## Transactional Replication

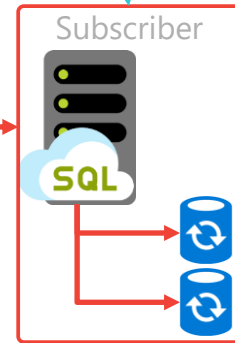
{ On-Premises/IaaS }



1



{ PaaS }



SQL  
RELAY

# Thank You



[@SQLDiplomat](https://twitter.com/SQLDiplomat)



[jqm-community@outlook.com](mailto:jqm-community@outlook.com)





PLATINUM SPONSOR



GOLD SPONSORS



SILVER SPONSORS



BRONZE SPONSORS





Please give us your feedback:

[sqlrelay.co.uk/feedback](https://sqlrelay.co.uk/feedback)

Thank you

