

CAF Ready Phase for SQL Migration

**List of Contents**

[1. Ready 3](#_Toc44672360)

[1.1 Azure Setup Guide 3](#_Toc44672361)

[1.1.1 Organizing Resources 3](#_Toc44672362)

[1.1.2 Managing Access 3](#_Toc44672363)

[1.1.3 Azure tools and Approaches 3](#_Toc44672364)

[1.1.4 Governance, Security and Compliance 4](#_Toc44672365)

[1.1.5 Monitoring and Reporting 5](#_Toc44672366)

[1.2 First Landing Zone 6](#_Toc44672367)

[1.2.1 Overview of Landing Zones 6](#_Toc44672368)

[1.2.2 Pros and Cons of Landing Zones 9](#_Toc44672369)

[1.2.3 Deciding Landing Zone 10](#_Toc44672370)

[1.2.4 Creating Landing Zones 19](#_Toc44672371)

[1.2.5 Migrating SSAS, SSIS and SSRS to an Azure 21](#_Toc44672372)

[1.2.6 Examples of Selecting Landing Zone 23](#_Toc44672373)

# Ready

In this section, we are going to prepare a cloud environment for planned changes. In simple words a landing zone to host SQL server that you plan to migrate from on-premises to cloud. This section will specifically discuss following

* Azure Setup Guide
* First Landing Zone

## Azure Setup Guide

Following are the features of the components that needed to build the Azure environment.

* Organizing Resources
* Manage Access
* Azure tools and Approaches
* Governance, Security and Compliance
* Monitoring and Reporting

### Organizing Resources

First step before SQL migration should be creating a management group hierarchy to simplify the management of your resources (Azure SQL Databases, SQL VM, SQL MI).

* Management groups: These groups are containers that help you manage access, policy, and compliance for multiple subscriptions. All subscriptions in a management group automatically inherit the conditions applied to the management group.
* Subscriptions: A subscription logically associates user accounts and the resources that were created by those user accounts. Each subscription has limits or quotas on the amount of resources you can create and use.
* Resource groups: we have to create a logical container into which Azure resources like databases, and storage accounts are deployed and managed.
* Resources: Resources are instances of services that you create, like virtual machines, storage, or SQL databases

### Managing Access

The next step in the process is to Manage who can access your Azure resources and subscriptions is an important part of your Azure governance strategy and assigning group-based access rights and privileges is a good practice.

### Azure tools and Approaches

Before migration we need to make sure that our source environment is supported and addressed all the possible migration option. Source Environment (On-premise SQL server) supported:

* SQL Server 2005
* SQL Server 2008 and SQL Server 2008 R2
* SQL Server 2012
* SQL Server 2014
* SQL Server 2016
* SQL Server 2017

**Data Migration Assistance tools:**

* [**Azure Database Migration Service (DMS):**](https://docs.microsoft.com/en-in/azure/dms/) Supports online (minimal downtime) as well as offline (one time) migrations at scale. The service uses the Data Migration Assistant to generate assessment reports that provide recommendations to guide you through the changes required prior to performing a migration.
* [**Data Migration Assistant (DMA):**](https://docs.microsoft.com/en-in/sql/dma/dma-overview?view=sql-server-2017)Its helps DMS to detect compatibility issues that can impact database functionality in your target Azure SQL Database. It also recommends performance and reliability improvements to DMS.
* [**Transactional Replication:**](https://docs.microsoft.com/en-in/sql/relational-databases/replication/administration/enhance-transactional-replication-performance?view=sql-server-2017)Transactional replication starts with a snapshot of the original database. After the initial snapshot is created, all changes to published objects and data in the original database (the publisher) are propagated to the new database (the subscriber) by the distributor, guaranteeing transactional consistency. This method is good when minimal downtime is required.
* [**Bulk Load:**](https://docs.microsoft.com/en-in/sql/t-sql/statements/bulk-insert-transact-sql?view=sql-server-2017)Transfer data to Azure SQL Database by using the bulk copy utility (bcp.exe) particularly helpful for older copies of SQL (back to 2000, select “Other Versions” for more options).

Microsoft also allows range of 3rd party tools to interact with azure for cloud migration. Below are few popular tools for reference:

|  |  |
| --- | --- |
| **Migration Tool** | **Features** |
| [Unifycloud](https://www.unifycloud.com/our-products/)  (Free Trial Available) | CloudRecon creates a cloud migration strategy and assessment by conducting a deep scan on your IT infrastructure, producing recommendations on which applications should migrate to the Cloud (SaaS, IaaS, or PaaS), and the roadmap behind it. CloudPilot analyze applications at the code level to determine cloud readiness and conduct migrations for Cloud-ready applications. |
| [Cloudamize](https://www.cloudamize.com/en/home/)  ([For pricing](https://www.digitalmarketplace.service.gov.uk/g-cloud/services/776640595250874)) | The Cloudamize SQL Migration & Modernization Planner eases and optimizes migrating on-premises server databases to Azure. Cloudamize is a cloud computing analytics platform that provides high precision analytics and powerful automation to improve the ease, speed, and accuracy of moving to the cloud. |
| Movere | Movere is a software as a service (SaaS) platform. It increases business intelligence by accurately assessing entire IT environments within a single day. |
| [Zerto](https://www.zerto.com/blog/cloud-migration/migrating-your-workloads-to-azure-as-easy-as-1-2-3/)  (Free Trial Available) | Zerto helps customers accelerate IT transformation by eliminating the risk and complexity of cloud adoption. By replacing multiple legacy solutions with a single IT Resilience Platform, Zerto is changing the way disaster recovery, backup and cloud are managed. |
| [MigVisor](https://www.migvisor.com/product/)  (Offer free trial) | MigVisor is a fully featured Database Management Software designed to serve Startups, SMEs. MigVisor scans your existing databases, identifies attributes and proprietary features currently in-use, and informs you of the fastest path to migrate your databases to cloud-native database solutions running in Azure. |

### Governance, Security and Compliance

After planning required resources for migration and managing to access to those resources, Data governance comes into picture. Once the governance strategy is ready, we can use the following azure services to automate the organization’s governance decisions:

* [**Azure blueprints:**](https://docs.microsoft.com/en-us/azure/governance/blueprints/overview)Azure Blueprints makes it possible for development teams to rapidly build and stand up new environments with trust they're building within organizational compliance with a set of built-in components -- such as networking -- to speed up development and delivery.

In our case let’s say we have created a blueprint for our migration process which will consist of all the artifacts (Resource group, policy assignment, role assignment, etc) required to create environment. Now at some point if you want to make any changes in this environment then blueprint is the only thing that you need to look. Blueprints can be saved and reuse for the purpose like creating new database.

* [**Azure Policy:**](https://docs.microsoft.com/en-us/azure/governance/policy/overview)Azure Policy is a service that you use to create, assign, and manage policies. These policies enforce rules on your resources, so those resources stay compliant with your corporate standards and service level agreements. In our case this service can be used to restrict regions for SQL databases.
* [**Azure security Center:**](https://docs.microsoft.com/en-us/azure/security-center/)Azure Security Center plays an important part in your governance strategy. It helps you stay on top of security because it:
* Provides a unified view of security across your workloads.
* Collects, searches, and analyzes security data from a variety of sources, which includes firewalls and other partner solutions.
* Provides actionable security recommendations to fix issues before they can be exploited.
* Can be used to apply security policies across your hybrid cloud workloads to ensure compliance with security standards

### Monitoring and Reporting

Azure offers many services that together provide a comprehensive solution for collecting, analyzing, and acting on telemetry from your applications and the Azure resources that support them. With Microsoft Azure monitoring, you can:

* Discover Azure Virtual machines under an Azure subscription.
* Find the number of VMs that are provisioned and running.
* View the status and configuration information of all the discovered resources.
* Collect and monitor performance metrics (CPU, Memory, Disk, Network, Thread & Process count, .Net CLR metrics).
* Collect and monitor Disk Utilization, Disk IO statistics, Network interface details by enabling Guest OS configuration.

Below are the tools that can be used for monitoring and reporting:

* Azure Monitor: Azure Monitor provides a single unified hub for all monitoring and diagnostics data in Azure. You can use it to get visibility across your resources. With Azure Monitor, you can find and fix problems and optimize performance. You also can understand customer behavior.
* Azure Service Health: Azure Service Health provides a personalized view of the health of the Azure services and regions you use.
* Azure Advisor: Azure Advisor is a free, personalized cloud consultant that helps you follow and implement best practices for Azure deployments.
* Azure Security Center: Azure Security Center also plays an important part in your monitoring strategy. It can help you monitor the security of your machines, networks, storage, data services, and applications.

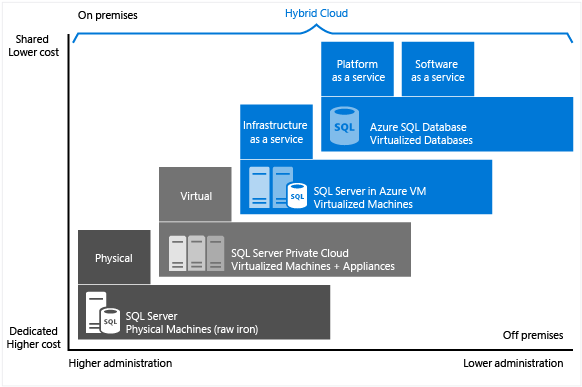
In our case these tools can also be extended to monitor On-premise SQL server which will help us to optimize the existing servers before we start the migration process.

## First Landing Zone

A landing zone is an environment for hosting your SQL server. Microsoft Azure provides both IaaS and PaaS options for building, deploying and managing SQL Server database applications in the cloud. The principle purpose of the landing zone is to be provisioned and prepared to host workloads being migrated from an on-premises environment (on-premises SQL server) into Azure (SQL DB, Managed Instance, Virtual Machines).

### Overview of Landing Zones

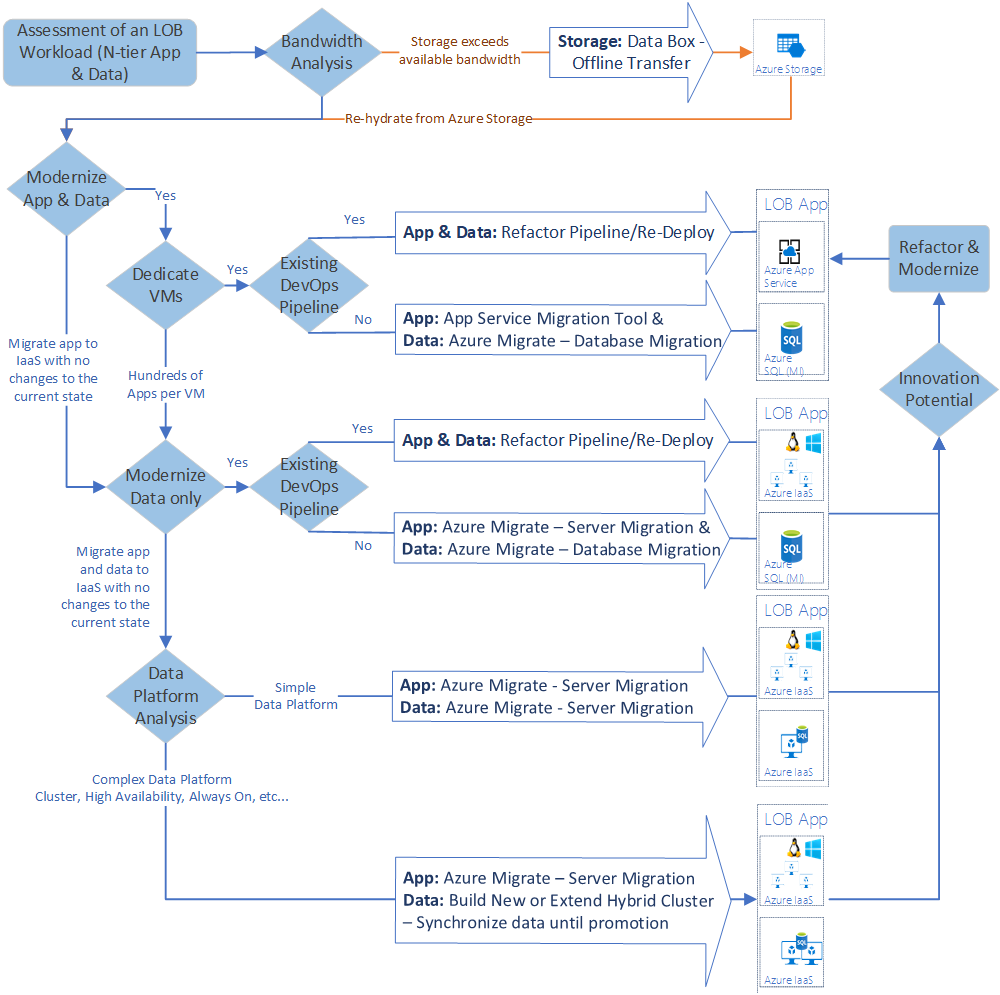
Let’s talk about various options available for creating and managing a SQL Server relational database in Azure with Azure SQL Database or Azure VMs. We will look at the differences, similarity, pros and cons of these landing zones.



Above diagram shows that each offering can be characterized by the level of administration you have over the infrastructure, and by the degree of cost efficiency.

**Migration Tools Decision Guide:** The below decision tree will help you to choose correct tools while migrating your on-premises application to azure. In the following diagram, the term modernize is used to reflect a decision to modernize an asset during migration and migrate the modernized asset to a PaaS platform. Before jumping to diagram following questions will help you to make decisions considering business goals, On-premises assets, Cost and ROI, etc

* Would modernization of the application platform during migration prove to be a wise investment of time, energy, and budget?
* Would modernization of the data platform during migration prove to be a wise investment of time, energy, and budget?
* Is your application currently running on dedicated virtual machines or sharing hosting with other applications?
* Will your data migration exceed your network bandwidth?
* Does your application make use of an existing DevOps pipeline?
* Does your data have complex data storage requirements?

[](https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/decision-guides/migrate-decision-guide/)

**SQL Databases:** A fully managed SQL database engine, based on the latest stable Enterprise Edition of SQL Server. This is a relational database-as-a-service (DBaaS) hosted in the Azure cloud that falls into the industry category of Platform-as-a-Service (PaaS). You can choose different database sizes from Basic (2GB, 5tps) to Premium (500GB, 735tps). Azure SQL database adheres to 99.99% availability service level agreement (SLA) set by Microsoft.

**SQL Managed Instances:** Azure SQL Database Managed Instance are another flavor and deployment option of Azure SQL Database. They are a managed Platform as a Service (PaaS) database offering, but with a far greater level of equality with the retail SQL Server product that we are all familiar with. Managed Instances give you capabilities that previously prevented many database systems being moved to Azure SQL Database, including cross database queries, lack of SQL Server Agent and other items.

**SQL Virtual Machines:**  SQL virtual machines fall into the industry category Infrastructure-as-a-Service (IaaS)and allows you to run SQL Server inside a fully managed virtual machine (VM) in the Azure cloud. Best for migrations and applications requiring OS-level access. SQL virtual machines are lift-and-shift ready for existing applications that require fast migration to the cloud with minimal changes or no changes.

SQL virtual machines offer full administrative control over the SQL Server instance and underlying OS for migration to Azure. The VM Machine in Azure can be accessed from any RDP (Remote Desktop Protocol) Client. It can be a Machine with the Windows OS installed (recommended), a Mac or any OS with RDP installed.

When migrating your data center from on-premises to Azure, you have two disk storage options available-

* Managed disks
* Unmanaged disks.

**Managed Disks:**

The managed disk provides enhanced manageability and high availability which provides the following features. it is available in several offerings: Standard HDD, Standard SSD, Premium SSD, and Ultra SSD.

* **Simple** - Abstracts underlying storage account/blob associated with the VM disks from customers. Eliminates the need to manage storage accounts for IaaS VMs
* **Secure by default** – Role-based access control, storage encryption by default and encryption using own keys
* **Storage account limits do not apply** – No throttling due to storage account IOPS limits
* **Big scale** - 20,000 disks per region per subscription
* **Better Storage Resiliency** - Prevents single points of failure due to storage Supports both Standard and Premium Storage disks

**Unmanaged Disks:**

* **Less availability:** Unmanaged disks do not protect against single storage scale unit outage
* **The upgrading process is complex:** If you want to upgrade from standard to premium on unmanaged disks, the process is very complex.
* Apart from this unplanned downtime, security is the downsides of the unmanaged disks. However, Cost differences between managed and unmanaged are based on your workload use case

### Pros and Cons of Landing Zones

Following are Pros and Cons of

* SQL Databases
* SQL Managed Instance
* SQL Virtual Machines

**SQL Databases:**

**Pros:**

* It is Simple and easy. If you need a SQL database, you deploy it in the Azure Portal and it’s ready. You do not need to wait for Azure infrastructure to be deployed.
* One can deploy an Azure SQL database and replicate it to another Azure region for disaster recovery in just a few minutes. It is faster than you will get SQL Server running in a virtual machine!
* Focus: You no longer are distracted by non-database activities. Its scalability, availability, built-in intelligent optimization, and advanced security allow you to focus on rapid development of your applications rather than managing virtual machines.
* It also preserves all platform-as-a-service capabilities, such as automatic patching and version updates, automated backups, and high availability. These capabilities help reduce management overhead and total cost of ownership.
* The connection is a straightforward process. If you have a browser, you can create, delete or administer your database. One can even create and administer Azure SQL database from your iPhone, your Android or any other similar device.
* Runs Latest SQL version, based on Enterprise edition

**Cons:**

* Backups fully managed. Also, most things are managed like backups, but which may be a problem if you want to log ship, migrate out, or do something more advanced.
* SQL agent don’t exist
* No cross-DB communication
* Can’t power down (so you can’t stop billing)

Use case that works well:

* When building new app
* Limitations make migrating existing apps close to impossible (that means no big clients will go here for existing heavy Databases)

**SQL Managed Instance:**

**Pros:**

* Provides the best of both (Azure SQL DB, SQL VM) offerings
* Infrastructure managed by Azure
* No more SQL patching
* Supports multiple dbs.
* Cross db communication supported
* SQL Agent available
* Lift & shift migrations possible (Fully-fledged SQL instance with nearly 100% compatibility with on-premises)
* Backups managed by azure

**Cons:**

* No Control over auto patching in SQL.
* Black box (Azure Set Lock pages in memory, maintenance volume GPOs on, Power savings) features set to Balanced mode

**SQL Virtual Machines:**

**Pros:**

* Familiarity: One who knows to work with SQL server, backup tools require, tackling problems related to Server will have the same experience in Virtual Machines
* Compatibility: When some application requires SQL Server you know that SQL Server on Windows Server will work.
* Flexibility: You can easily manage your system resources running SQL Server in a VM. Just add CPU, RAM, or storage resources to the VM according to your needs.
* You can easily create a SQL Server VM from the virtual machine image gallery, which allows you to choose the right version, edition, and operating system to fit your organization’s needs.
* Its resources aren’t shared with other Windows Azure tenants
* The VM Machine in Azure is more secure than the SQL Azure Database because you can configure and restrict the IPs that access to the VM Machines. The passwords can be protected using the OS security.

**Cons:**

* The VM Machine in Azure is as difficult as a local machine. You need to work on the OS Administration as well as the traditional DBA tasks
* Backups are your responsibility and under your control
* Unused, idle or over-provisioned resources may cost you a lot.

### Deciding Landing Zone

Assessment is an important activity which helps you decide on target landing zone from list above based on your source SQL Server system capabilities, features and code. We now know what workloads we’re dealing with, where they are, how big they are and what they’re used for. This data obtained from the initiate and discover phase can now be used as input into the second phase, assessment.

The data will need to be compiled and analyzed to achieve our goals for this phase, which is to identify:

**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.

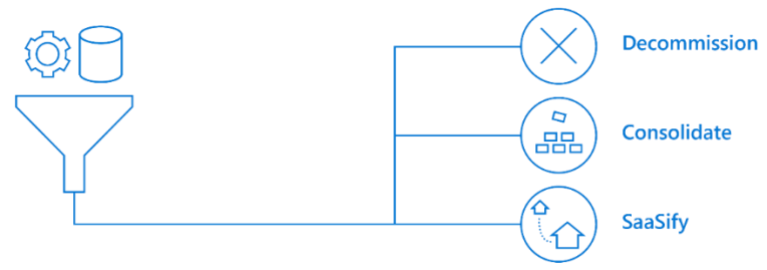
**Effort involved to fix issues:** An estimate of the time and processes required to rectify the above highlighted issues.

To realize these goals, a closer examination of workloads is done with emphasis placed on the following areas:

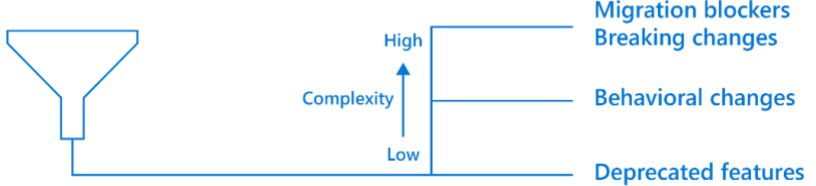
* Assess workloads for migration
* Assess workload criteria
* Assess database using Data Migration Assistant

#### Assess workloads for migration

* To establish a complete migration plan, a thorough assessment of your workloads prior to migration will help determine which databases will need to be migrated to the cloud as well as the quantities involved.
* Investigations should be made into whether on-premises applications in use now have a SaaS-based or hosted deployment model available, and if so consider moving to that platform to lower administrative costs.



* Proceed down the cloud migration path by looking to migrate any low effort, high impact databases. That is, look to segregate the discovered workloads according to their business impact.
* Workloads for applications used by a select number of users in the enterprise should have a smaller scope for disruption than applications used widely across the business.
* Non-critical workloads such as development, testing and training platforms would make good candidates for the first wave of migrations.
* Workloads can be further ranked by the severity of issues highlighted during the initiate and discover phase.
* Migration blockers or known breaking changes might require substantial remediation work, and position workloads well down the migration list.



* Behavioral changes might mean that some workloads need additional investigation and planning before they can make the transition to the cloud to fully understand any impacts.
* Any workloads making use of deprecated features should still be migratable but warrant investigation later to remove their dependency on those deprecated features.

#### Assessment Goals

Following are Assessment Goals

**Performance Requirement**:

Identify workloads which are a high or low user of resources, and gauge how many Azure resources will be required post-migration.

**Iaas VMs:**

If looking to transition to SQL Server on Azure IaaS VMs, this might simply amount to matching the number of compute cores currently allocated to those on the target platform.

**Azure SQL Databases:**

If moving to Azure SQL Databases this might require computing the number of Database Transaction Units (DTU) or virtual cores (vCores) needed for each database. Azure SQL Database provides two different models for measuring and purchasing compute: DTU-based and vCore-based.

* [Choose between the vCore and DTU purchasing models](https://docs.microsoft.com/en-us/azure/azure-sql/database/purchasing-models) - Azure SQL Database & SQL Managed Instance
* What are Database Transaction Units (DTUs)?
* [Service tiers in the DTU-based purchase model](https://docs.microsoft.com/en-us/azure/azure-sql/database/service-tiers-dtu) - Azure SQL Database performance is measured using the Database Transaction Unit or DTU, which is an aggregated metric of CPU, memory and I/O. DTU’s are useful for understanding the relative amount of resources allocated between different Azure SQL Databases as varying available performance levels are characterized by their allocated DTUs.
* [What are VCores (Virtual Cores)? vCore model overview - Azure SQL Database & SQL Managed Instance](https://docs.microsoft.com/en-us/azure/azure-sql/database/service-tiers-vcore?tabs=azure-portal)

[**Azure Database DTU Calculator**](http://dtucalculator.azurewebsites.net/)**:**

This calculator will help you determine the number of DTUs for your existing SQL Server database(s) as well as a recommendation of the minimum performance level and service tier that you need before you migrate to Azure SQL Database. Knowing the minimum service tier will allow you to get the performance you need while minimizing your costs.

**Compliance Requirement:**

Determine if there are any specific security or regulatory requirements. Microsoft’s Trusted Cloud initiative is built around the four foundational principles of security, privacy, compliance and transparency which is reflected in the platforms and services offered through Azure. Azure data centers comply with strict regulations and compliance standards, to help customers meet international data protection laws and industry requirements. Data residency laws might also mean that data for a given application must be kept in country or geographic region, restricting which Azure data centers can be utilized.

**Migration downtime:**

Understand the business requirements around the workload to be migrated. Is any downtime acceptable? This will impact the migration approach, toolsets used, and timeframes involved.

**Availability:**

Following on from migration downtime, what are the ongoing availability requirements for the workload? Azure SQL Databases are locally highly available as standard, with three copies of your database used to keep the data online and accessible during patching and transient hard failures. SQL Server on Azure VMs would require HA technologies such as Always on Failover Clustering, Always On availability groups, database mirroring or log shipping.

**Disaster Recovery:**

Understand RTO and RPO requirements. Establish replica out-of-region for minimal cost, the geo-replication feature protecting your database and application against wider regional failures. If an entire region experiences a service disruption, the locally redundant copies of your data would temporarily be unavailable. If you have enabled geo-replication, three additional copies of your Azure Storage blobs and tables are stored in a different region. In the event of a complete regional outage or a disaster in which the primary region is not recoverable, Azure remaps all of the DNS entries to the geo-replicated region. SQL Server on Azure IaaS VM doesn’t have readily available DR support and might require implementing SQL Server Enterprise Edition using Always On Availability Groups to meet aggressive RTO requirements for mission critical workloads. Lower priority workloads using Azure Site Recovery would normally suffice where protection at the virtual server level is acceptable.

For more info : <https://docs.microsoft.com/en-us/azure/virtual-machines/virtual-machines-disaster-recovery-guidance>

**Custom Workloads:**

There may be databases that have 3rd party tool integrations which are not currently supported on Azure SQL Database. The 3rd party vendor might need to be approached for a compatible version or alternate products considered.

**Cost Planning:**

Collect resource usage report (such as CPU, memory, and storage) to calculate and forecast for costs and expenditures. Typically, on-premises virtual machines are over-provisioned but only utilized under 20 percent. The goal for the Azure cloud model is to drive your virtual machines to at least 90 percent utilization while meeting performance and reliability goals. Therefore, the usage report will help us in selection of appropriate tier in azure cloud.

**Assess database using Data Migration Assistant**:

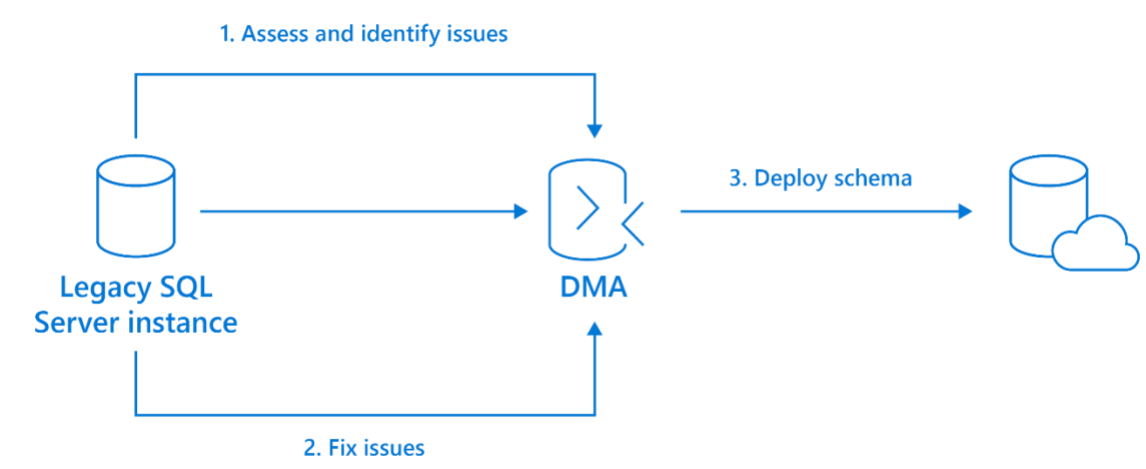
An overview of DMA is available at: <https://docs.microsoft.com/en-us/sql/dma/dma-overview>

DMA is another freely available download from the Microsoft website to help with the migration of on-premises SQL Server instances to Azure SQL Database or to a modern SQL Server instance hosted on an Azure Virtual Machine. DMA is installed and executed locally. DMA recommends performance and reliability improvements for your target environment and allows you to move your schema, data, and uncontained objects from your source server to your target server. It also provides recommendations on how to remediate those issues -

* Assess on-premises SQL Server instances migrating to Azure SQL Database
* Discover issues that can affect an upgrade to an on-premises SQL Server
* Discover new features in the target SQL Server platform that the database can benefit from after an upgrade
* Migrate **an on**-premises SQL Server instance to a modern SQL Server instance

**Note** - For large migrations (in terms of number and size of databases), we recommend that you use the Azure Database Migration Service, which can migrate databases at scale.

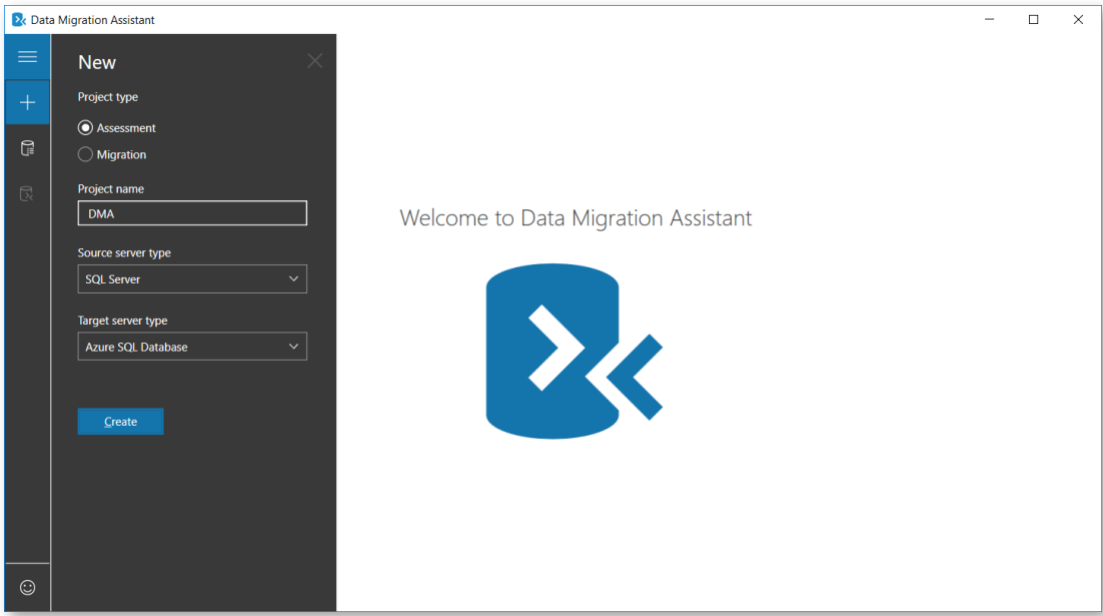
**Using DMA:**

When using DMA, it should first assess and identify issues in the source database that would prevent a successful migration. Armed with this information, you must fix the root cause or implement an alternate methodology for each highlighted issue. The assessment and fix processes are then repeated until the source database passes all DMA tests, at which point the schema of the source database can be deployed to the target database in the cloud with a high degree of confidence.

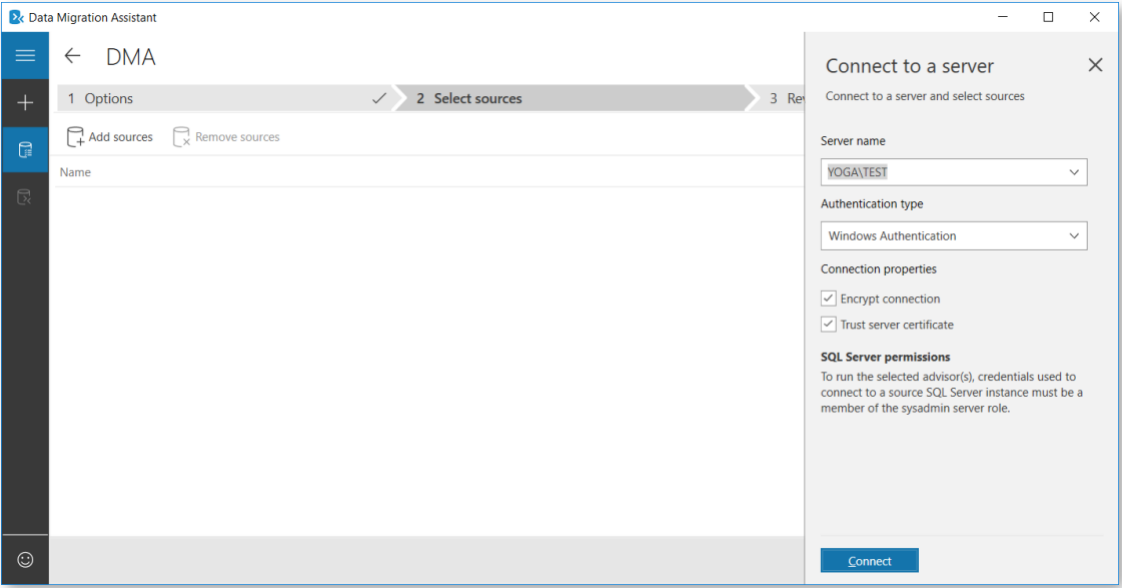
**Fig. Assessment and fixworkflow using DMA**

**Assessment steps using DMA –**

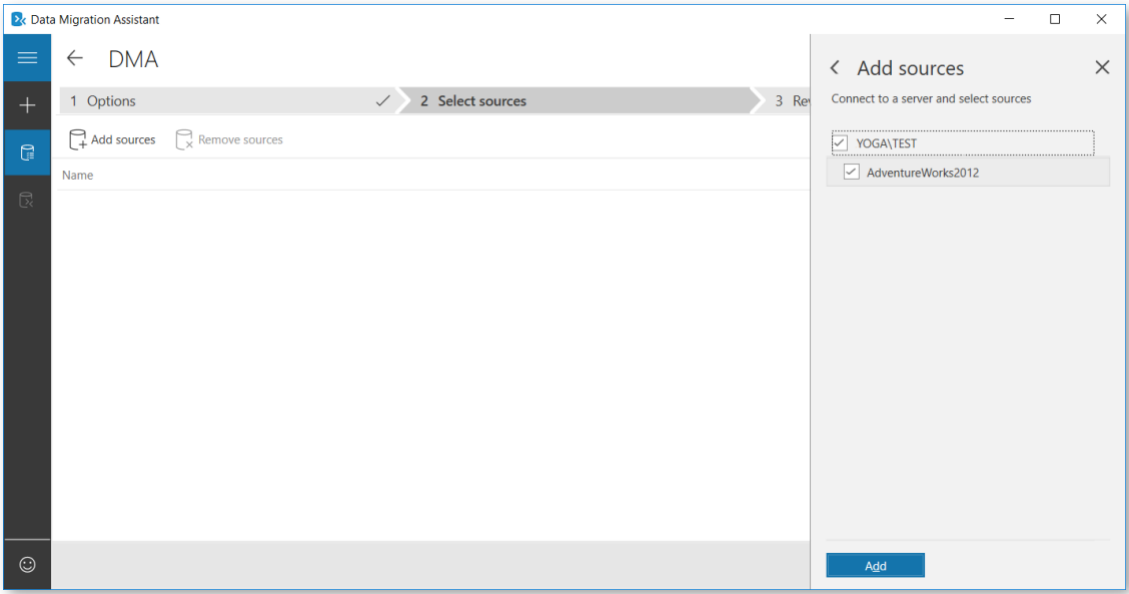
* To install DMA, download the latest version of the tool from the Microsoft Download Center, and then run the DataMigrationAssistant.msi file
* Please look at the Best Practices for running DMA
* Data Migration Assistant: Configuration settings
* Note - To run an assessment, you have to be a member of the SQL Server sysadmin role
* Create a New Assessment project.
* Open the Data Migration Assistant (DMA), and then begin creating a new assessment project. Select the New (+) icon, select the Assessment project type, specify a project name, select SQL Server as the source and Azure SQL Database as the target, and then select Create. Specify a project name, select SQL Server as the source server type, and then select Azure SQL Database as the target server type.



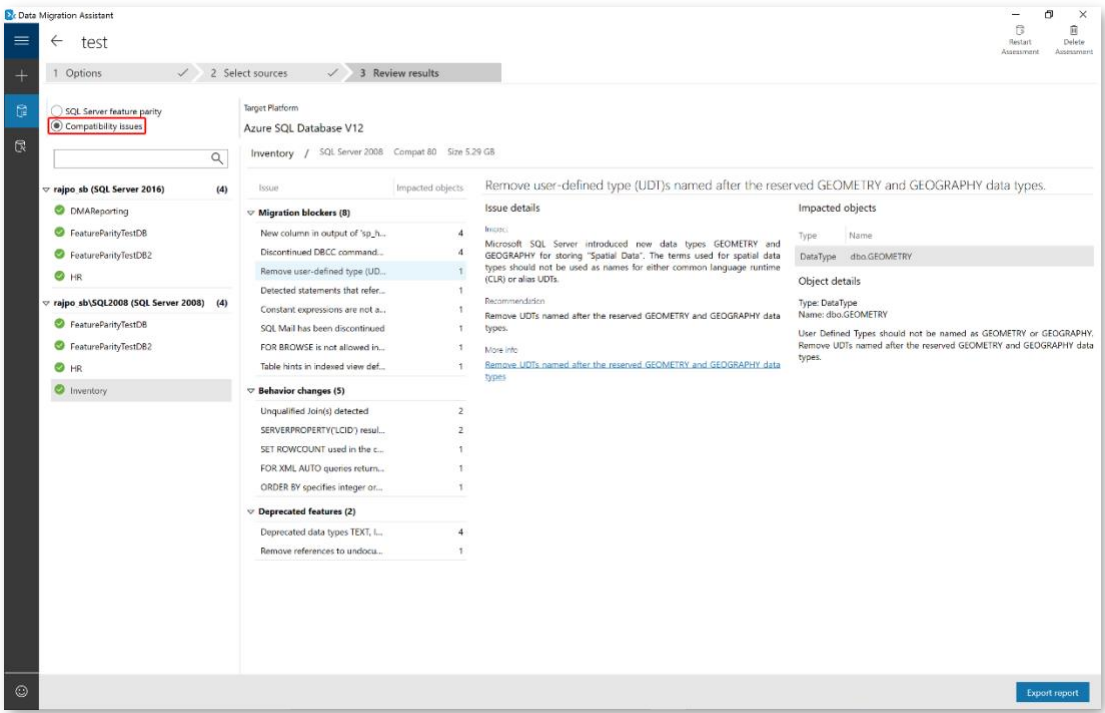
* Select the type(s) of assessment reports (Check database compatibility and Check feature parity) that you want to generate. Select one or both assessment report types (Check database compatibility and Check feature parity), and then select **Next**.
* In the connect to a server blade, specify the name of the SQL Server instance to connect to, specify the authentication type and connection properties, and then select **Connect**



* In the Add sources fly-out, select the database(s) that you want to assess, and then select **Add.**

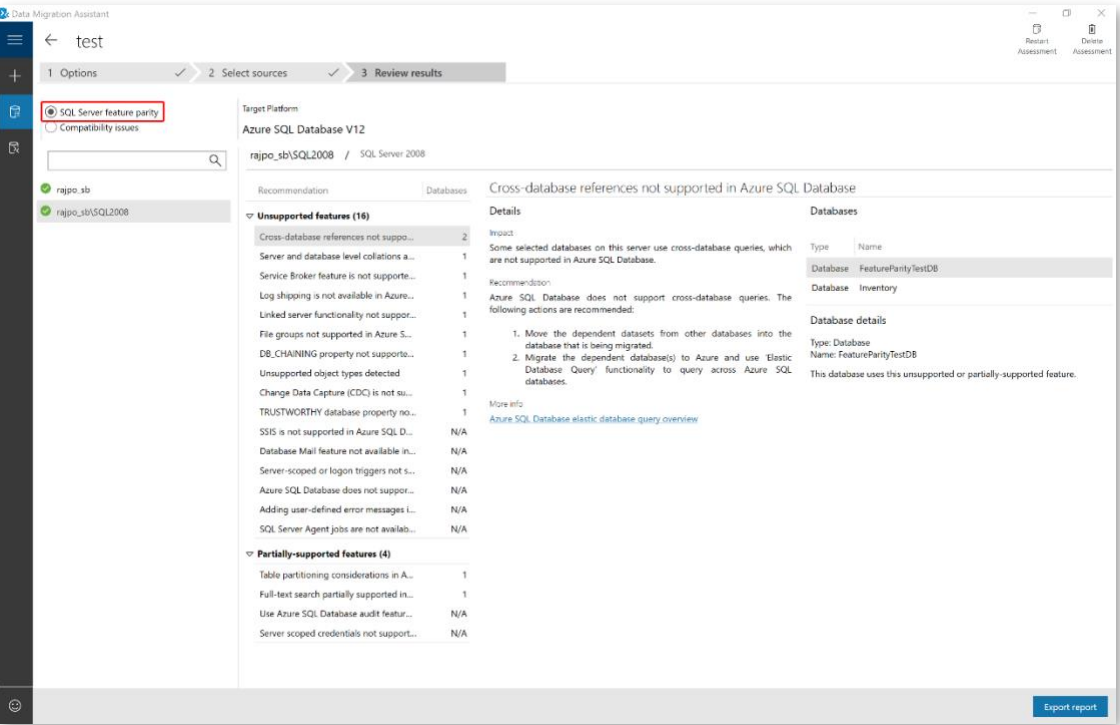


* Select Start Assessment: Now wait for the assessment results; the duration of the assessment depends on the number of databases added and the schema size of each database. Results will be displayed per database as soon as they are available.
* Select the database that has completed assessment, and then select Compatibility issues to review incompatible objects categorized under Migration blockers, Behavior changes, and Deprecated features.



**Fig: DMA Compatibility findings and Recommendation**

* Similarly, you can review recommendations across Performance, Storage, and Security areas. Feature recommendations cover a variety of features such as In-Memory OLTP and Columnstore, Stretch Database, Always Encrypted, Dynamic Data Masking, and Transparent Data Encryption.
* Select SQL Server feature parity to review the unsupported features or partially supported features in Azure SQL Database.

****

**Fig: DMA feature parity results and findings**

* DMA also provides a comprehensive set of recommendations, alternative approaches available in Azure, and mitigating steps.
* Review assessment results
* After all database assessments are complete, select Export report to export the results to either JSON or CSV file to analyze the data at your convenience.
* For additional information plz look at - Migrate On-Premises SQL Server using Data Migration Assistant.

Look for high level red flags –

* **The findings surfaced during the initiate and discover phase and the assessment tools used during the assessment phase should now be considered. Possible options to work around identified issues need to be identified or flagged as potential migration blockers if a workable solution cannot be found.**
* **Are you using features such as Database Mail, SQL Agent? Such features are available in Azure SQL Database Managed Instance, which is highly compatible with on-premises SQL Server.**
* **If the database is not using advanced SQL Server features such as MSDTC, MDS or QTS, then Azure SQL Database or Azure SQL Database elastic pools would be a good choice as Microsoft Operations takes care of the majority of infrastructure management drastically reducing administrative overhead costs.**
* **Are you looking to also migrate SSRS, SSAS or SSIS? Unfortunately, not all SQL Server components currently have an Azure data services equivalent.**
* **SSRS currently has no direct cloud-based equivalent, but reports could be rewritten based around Microsoft Power BI. Alternatively, SSRS can be deployed using SQL Server on an Azure VM.**
* **SSAS can be migrated** to Azure Analysis Services which is largely compatible with recent versions of SQL Server Analysis Services Enterprise Edition. Alternatively, SSAS can be deployed using SQL Server on an Azure VM.

This Channel 9 video shows you how to migrate on premises SSAS Tabular model to Azure Analysis Services - <https://channel9.msdn.com/series/Azure-Analysis-Services/AzureAnalysisServicesMovingModels>

SSIS packages can be migrated to SSIS runtime in Azure Data Factory PaaS service. For tutorials on this operation see - <https://docs.microsoft.com/en-us/azure/data-factory/how-to-migrate-ssis-job-ssms>

<https://docs.microsoft.com/en-us/azure/data-factory/create-azure-ssis-integration-runtime>. Alternatively, SSIS can be deployed using SQL Server on an Azure VM.

When looking to choose an appropriate target platform for each workload, there are four considerations to be made:

* Usage Scenarios
* Features
* Total Cost of Ownership

#### Choosing target platform by usage scenarios

Following are target platform by usage scenarios

**Azure SQL Database single databases:**

Azure SQL Database is ideally suited for customers developing new SaaS multi-tenant applications or intentionally transforming their existing on-premises applications into a SaaS multi-tenant application. There are enough differences between Azure SQL Database single databases and elastic pools and on-premises SQL Server that it is not usually trivial to lift-and-shift on-premises database workloads to Azure SQL Database. Similarly, third-party applications do not yet support the Azure SQL Database platform. This version of SQL Server is designed for a high level of uptime, with high availability coming as standard, which can be extended to provide geo-replicated topologies.

**Azure SQL Database Managed Instance:**

Managed Instances are good for customers looking to migrate several applications from on-premises or VM/hosted, self-built or ISV provided, with as low migration effort as possible. Additionally, these features make Managed Instances more desirable:

* High level of compatibility with on-premises SQL Server
* Support for isolation of workloads from the public internet using VNET support with private IP addresses and VPN to on-premises networks

**SQL Server on Azure VMs:**

Virtual machines can help customers that need to customize the operating system or the database server, as well as customers having specific requirements in terms of running third party apps side-by-side with SQL Server (on the same VM).

#### Choosing target platform by features

Following are target platform by features

**Azure SQL Database single databases:**

Azure SQL Database would be appropriate for use if the application surface area is database scoped. If the application uses some SQL features, then Azure SQL Database may not be appropriate as not all are yet available.

**Azure SQL Database Managed Instance:**

Would be appropriate for use if the application surface area is instance scoped and requires features not available in Azure SQL Database such as:

* SQL Agent
* MSDTC
* DQS
* MDS
* Database Mail
* Filestream
* Filetable
* Polybase

Additional features include:

* Support for Linked Servers
* Supports new Azure cloud services such as Threat Detection

**SQL Server on Azure IaaS:**

Use if application surface area is instance scoped and requires features not available in Azure SQL Database Managed Instance. Additionally, supports local instances of:

* SSRS
* SSAS
* SSIS

#### Choosing Target Platform by Cost

Following are target platform by cost

**Azure SQL Database:**

The Platform-as-a-Service nature of Azure SQL Database greatly reduces administration and management costs over the more traditional SQL Server on Azure IaaS topology, as most of the required work is completed silently in the background for you by Microsoft Operations. This is evident at scale where considerable savings in time and effort can be made.

Azure SQL Database Elastic Pools can offer considerable savings if used by multiple databases that have varying and unpredictable usage demands. The sharing of compute resources amongst all databases in the pool means that customers are not required to over-provision resources for all databases to meet their infrequent spikes in usage. Further savings are made on lowered server maintenance and administrative costs as most of the required work is completed silently in the background by Microsoft Operations.

**Azure SQL Database Managed Instance:**

Managed Instances is offered to those customers who want a fully managed service offering, where they can easily lift and shift their on-premises environment with minimal configuration changes. The environment offers a minimum of 8 cores and up to 8 TB of storage and sits in an isolated virtual network. This offering is great for customers that want to quickly get to the cloud and want to avoid the overhead of virtual machines.

**SQL Server on Azure VMs:**

VMs impose higher compute, storage, and management costs over the Azure SQL Database offerings but grants greater control across the SQL Server and infrastructure.

### Creating Landing Zones

This section will specifically discuss**-**

* SQL Databases
* Azure SQL Managed Instance
* Azure SQL Virtual Machines

#### SQL Databases

There are total 3 ways to create Azure SQL databases.

**Azure Portal:**

A single database is the quickest and simplest deployment option for Azure SQL Database. You manage a single database within a SQL Database server, which is inside an Azure resource group in a specified Azure region. In this you have create a new resource group and SQL server for the new database. Two option to create a database are as follows:

* Provisioned Database: A provisioned database is pre-allocated a fixed amount of compute resources, including CPU and memory, and uses one of two purchasing models (vCore- Based and DTU-based model).
* Serverless compute tier: The serverless compute tier is only available in the vCore-based purchasing model and has an auto-scaled range of compute resources, including CPU and memory.

The above option directly affects the performance and price of the deployed database. Below Link demonstrate step by step creation of Azure SQL Database using Azure Portal in detail.

LINK: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-single-database-get-started?tabs=azure-portal>

**Azure CLI and PowerShell:**

Though creation of Azure SQL Database using portal is the quickest and convenient way but, if you are writing a script to automate a task PowerShell or Azure CLI is a good option. Below Link demonstrate step by step creation of Azure SQL Database using Azure PowerShell and CLI.

LINK: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-single-database-get-started?tabs=azure-powershell> OR <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-single-database-get-started?tabs=azure-cli>

**Azure Resource Manager (ARM) Template**

Resource Manager Template is a JSON file that defines the infrastructure and configuration for your project. The template uses declarative syntax, which lets you state what you intend to deploy without having to write the sequence of programming commands like we write CLI.

The main benefits of the ARM API are that you can deploy several resources together in a single unit, in that the user declares the type of resource, what name to use and which properties it should have; the ARM API will then either create a new object that matches those details or change an existing object which has the same name and type to have the same properties. ARM Templates are what really gives us the ability to roll out Azure “Infrastructure as code”. ARM has advantage over CLI or PowerShell that you don’t need to check if the Database already exist or before trying to create it since the arm template will just update what’s missing in the resource group.

LINK: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-single-database-get-started-template?tabs=azure-powershell>

#### Azure SQL Managed Instance

It’s a service that **Azure** offers us to have a **SQL Server** instance, with all the advantages that it has, but without the need to manage our operating system or updates of new versions of both SQL Server and **Windows Update** patches.

There are 2 methods to deploy Azure SQL Managed Instance.

**Azure Portal:**

In the Azure portal, you configure the necessary parameters (username/password, number of cores, and max storage amount), and automatically create the Azure network environment without the need to know about networking details and infrastructure requirements. You just make sure that you have a subscription type (Enterprise Agreement, Pay-as-you-go, Cloud Service Provider, etc) that is currently allowed to create a managed instance. Below link walks you through the steps to create a Azure SQL Managed Instance using Azure portal and configuring the Vnet.

LINK: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance-get-started>

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance-configure-vnet-subnet>

**ARM template deployment:**

Azure API enables you to create Azure SQL Managed Instance using ARM templates. These are JSON objects that contain the definition of resources that should be created. You can send these objects to the Azure REST API to automate the creation of Azure SQL Managed Instance. Azure enables you to describe resources that you need using ARM templates and specify in the code what kind of resources you need and what infrastructure you want to build (so-called infrastructure as a code concept).

In order to create a new Azure SQL Managed Instance using ARM templates, you need to create an ARM JSON request. Once you create this JSON template you should save it to your local computer in some file and use this file as an input for PowerShell command that will execute it or Invoke it. Below Links will give you in detail procedure for deploying and creating azure SQL managed instance using ARM.

LINK: <https://blogs.msdn.microsoft.com/sqlserverstorageengine/2018/07/02/deploy-azure-sql-managed-instance-network-environment-using-arm/>

<https://blogs.msdn.microsoft.com/sqlserverstorageengine/2018/05/15/creating-azure-sql-managed-instance-using-arm-templates/>

#### Azure SQL Virtual Machines

There are 2 ways to create a SQL VM.

**Azure Portal:** In the azure portal while SQL Virtual Machines you can configure some basic settings (subscriptions, resource group, etc.), Disk type, virtual networks, monitoring, SQL server settings. Below link will walk you through the detail steps for creating SQL virtual Machines in windows as well as in Linux.

LINK: <https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sql/quickstart-sql-vm-create-portal>

**Azure PowerShell:** Creating your VM in Azure manually using the portal is a very pleasant experience. However, when you have a hundred and sometimes thousands of computers it is extremely exhausting and boring to create and configure it. With PowerShell, it is possible to automate several administrative tasks and create scripts to automatically create VMs, enable ports, download and create remote desktop files, administer services, etc. Below links explains detail procedure about creating and provision SQL virtual Machine using PowerShell.

LINK:

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sql/quickstart-sql-vm-create-powershell>

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sql/virtual-machines-windows-ps-sql-create>

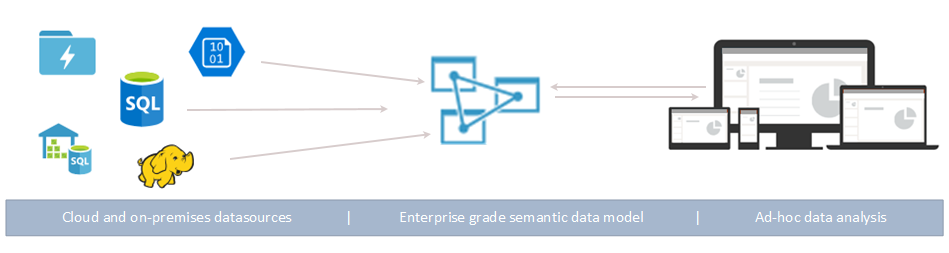
### Migrating SSAS, SSIS and SSRS to an Azure

This section discusses equivalent Azure fully managed services and VMs when migrating SSAS, SSIS, SSRS

#### Fully managed service offering

Migrating SSRS, SSAS or SSIS? Unfortunately, not all SQL Server components have an Azure data services equivalent currently.

**SQL Services Analysis Services (SSAS)**



SSAS can be migrated to [Azure Analysis Services](https://docs.microsoft.com/en-us/azure/analysis-services/analysis-services-overview) which is largely compatible with recent versions of SQL Server Analysis Services Enterprise Edition. Azure Analysis Services is a fully managed platform as a service (PaaS) that provides enterprise-grade data models in the cloud. Azure Analysis Services can be a good choice for large data models. Pay attention to the service tier, which corresponds to the hardware choice in the on-premises world. For more info see Azure Analysis Services videos on [Channel 9](https://channel9.msdn.com/series/Azure-Analysis-Services).

**SQL Server Integration Services (SSIS)**

Use the [Azure SSIS Integration Runtime](https://docs.microsoft.com/en-us/azure/data-factory/concepts-integration-runtime), which is the compute infrastructure used by Azure Data Factory. SSIS packages can be invoked using stored procedures in Azure Data Factory to provide true first-class support of SSIS package execution. With the release of Azure Data Factory v2, the Integration Runtime (IR) has been introduced. The IR is the compute infrastructure for Azure Data Factory, and it allows for several data integration capabilities:

* Data movement between different network environments
* Activity Dispatch
* SSIS package execution

With the Azure-SSIS IR, we can lift & shift existing SSIS projects and packages to the Azure cloud. There you can natively run your SSIS packages in a managed environment. The Azure-SSIS IR is a managed cluster of Azure VMs where you can execute SSIS packages. With the creation of the IR, you can specify the compute power of a node (scale up) and also how many nodes your cluster will have (scale out).

In the Azure-SSIS, you have access to an Integration Services catalog, just like you have on-premises. You have two options for storing your SSIS catalog:

* An Azure SQL database
* A SQL Server Managed Instance

The advantages of using Azure SQL DB is that it is lightweight and easy to set-up. The disadvantage is that it doesn’t have SQL Server Agent, but Managed Instance does. For more info see [Azure SSIS lift-and-shift](https://docs.microsoft.com/en-us/sql/integration-services/lift-shift/ssis-azure-lift-shift-ssis-packages-overview?view=sql-server-ver15) overview and [Azure Data Factory tutorial](https://docs.microsoft.com/en-us/azure/data-factory/tutorial-deploy-ssis-packages-azure).

**SQL Server Reporting Services (SSRS)**

SSRS currently has no direct cloud-based equivalent, but reports could be re-written using Microsoft Power BI or migrate to SSRS running on an Azure VM.

#### Infrastructure as a service

First, install services on an Azure VM and connect to Azure SQL Database or Managed Instance. Some reference links for SSRS and SSAS migrations:

* [SSAS multidimensional](https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/supported-data-sources-ssas-multidimensional?view=asallproducts-allversions&viewFallbackFrom=sql-server-ver15)
* [SSAS tabular](https://docs.microsoft.com/en-us/analysis-services/tabular-models/data-sources-supported-ssas-tabular?view=asallproducts-allversions&viewFallbackFrom=sql-server-ver15)

Basically, tabular model is something in between relational databases and multidimensional cubes. Similar to databases, tabular model supports tables with relations. Similar to cubes, the model supports measures and key performance indicators (KPIs).

**Why tabular solution can be chosen:**

* Easier for understanding and creating the model
* Works quicker than multidimensional cubes for queries based on columns
* Hardware, such as disks, is not important. However, tabular is a memory dependent solution and more memory will ensure better performance
* More efficient data compression about one-tenth of the size, whereas the compression of multidimensional is a third of the size of the original database

**Why multidimensional solution can be chosen:**

* Works better with a large amount of data – when we are talking about terabytes, it’s better to go with the multidimensional database. If your database requires more than 5 terabytes, multidimensional is the only option
* Performs better in terms of scalability
* Some features, such as aggregations or actions, are supported in the multidimensional model only

#### [SSRS data sources](https://docs.microsoft.com/en-us/sql/reporting-services/report-data/data-sources-supported-by-reporting-services-ssrs?view=sql-server-ver15) & [SSRS Connection Type](https://docs.microsoft.com/en-us/sql/reporting-services/report-data/sql-azure-connection-type-ssrs?view=sql-server-ver15)

When you deploy Reporting Services, a set of data processing extensions are automatically installed and registered on both the report authoring client and on the report server to provide access to a variety of data source types. Reporting Services installs the following data source types:

* Microsoft SQL Server
* Microsoft SQL Server Analysis Services for MDX, DMX, Microsoft Power Pivot, and tabular models
* Microsoft Azure SQL Database
* Oracle
* SAP BW
* Hyperion Essbase
* Microsoft SharePoint List
* Teradata
* OLE DB
* ODBC
* XML

[**SSRS connection type**](https://docs.microsoft.com/en-us/sql/reporting-services/report-data/sql-azure-connection-type-ssrs?view=sql-server-ver15)

Alternatively, it’s still possible to utilize an existing on-premises SSRS server to connect to your Azure SQL Database or Managed Instance for reporting purposes

### Examples of Selecting Landing Zone

In this section we’ll look at three examples of customer workloads and requirements and decide on an appropriate landing zone as well as the method of migration we’d use to get it there.

#### Azure SQL Database Single Database

The customer has an application that uses an on-premises SQL Server 2008 R2 installation. This application is 24x7 business critical with significant impact from any downtime during the year. This strict operational requirement has meant no scheduled maintenance windows are available and unscheduled maintenance is unacceptable. To facilitate this, the underlying infrastructure is designed for high availability with full redundancy across all components. The actual database has minimal growth per year, but an extremely high transaction rate that requires sizeable amounts of compute resources coupled with low latency/high throughput storage and networking. The redundancy requirements across all components means there are a lot of SQL Servers, virtual machines, storage, and networking to keep the DBAs and Sysadmins busy for significant amounts of time, which they’d rather spend on improving the performance and security of the application.

**Solution:**

In this scenario, utilizing the Azure SQL Database fully managed service offering platform would be beneficial as it removes the issue of managing compute and storage requirements. With Azure SQL Database including local high availability as standard for a 99.99% uptime SLA, and the possibility of using geo-located replicas for regional high availability and disaster recovery, the high uptime requirements should be easily met. Azure SQL Database’s premium performance tier is capable of 2ms IO latency with IO throughput measured at approximately 48 IOPS per DTU, a performance level on-par with enterprise flash-based SAN storage appliances. The lack of an allowed maintenance window means that it would not be possible to migrate the on-premises databases to Azure SQL Database using a backup-and-restore technique due to the sheer size of the data involved.

It would take too long to copy the backup files over the WAN connection. Instead, transactional replication would be used to synchronize the data in the background while keeping the source database online and available. Moving to Azure SQL Database would save costs on hardware and management overhead by removing the need to monitor, patch, and fix the numerous servers in the on-premises solution. The application would also benefit from Azure SQL Database’s built-in intelligent optimization and monitoring. Azure SQL Database Advisor can make recommendations of missing indexes that should be added or unused indexes that could be removed, to proactively improve application performance. Azure SQL Database Intelligent Insights analyzes SQL Database performance by comparing the current database workload with a historical baseline to highlight performance degradations and their possible causes. Threat Detection can be utilized through Azure Security Center, to detect and respond to potential threats as they occur.

#### Azure SQL Managed Instance

The customer has a custom-built application based on an on-premises SQL Server that contains sensitive intellectual property-related data. The application code has had some quirky development practices used in the past, which have caused compatibility issues over the years during upgrades from SQL Server 2000 to 2005 to 2008 to, finally, 2012. Any changes made to this application are costly as the development work has always been handled by a third-party development team. The application also does many cross-database queries for reporting and analysis reasons. A scheduled outage of the application would have a medium level impact on the business but would be acceptable with some forward planning. The customer is not convinced their current backup and recovery solution is reliable, with frequent failures occurring due to lack of free disk space or hung backup agent processes. The customer would like to remove the headaches of these operational tasks such as performing backups, patching, and version upgrades. The customer also has heard that the multi-tenant nature of cloud-based services means they will need yet another set of user credentials for each user to remember, which could cause extra overhead to users.

**Solution:**

The preferred platform of Azure SQL Database Single Databases doesn’t yet support all the features and levels of compatibility that a traditional on-premises SQL Server might, with one feature that it lacks being the ability to perform cross-database queries, something the customer has pointed out they need. Rather than have to reengineer the application to fit a solution that which will be both costly and time consuming, Azure SQL Database Managed Instance can be utilized to bring a high level of compatibility with on-premises SQL Server, while still enjoying many of the benefits of the cloud.

To migrate, a native SQL backup can be taken of the on-premises SQL Server databases, uploaded to Azure Blob storage, and restored straight into Azure SQL Database Managed Instance. Once on Azure SQL Database Managed Instance, the built-in automated backup and point-in-time restore capabilities can remove the headaches of ensuring reliable data protection, but the customer configurable backup retention and recovery means they still have control when needed. Furthering their data protection, Azure SQL Database Managed Instance’s native support of encryption means that valuable intellectual property data can be secured by encrypted data in transit and at rest using customer provided encryption keys.

Cost savings can be made on server maintenance and administration areas, with no patching and version upgrade overhead, so that administrators can fulfil higher priority tasks. Additional savings could be made by bringing their own SQL licenses with active Software Assurance using the Azure Hybrid Benefit for SQL Server. Lastly, by synchronizing their on-premises active directory to Azure Active Directory using the free Microsoft AADConnect directory synchronization tool, it’s possible to provide a single sign-on experience, so that Azure SQL Database Managed Instance databases are accessible using Windows user credentials without any additional login prompts being displayed. Managed Instances also adhere to compliance standards available to Azure SQL Server, so that customers do not need to maintain a lot of administrative overhead to keep up with new standards.

#### SQL Server on Azure Infrastructure as a Service (IaaS)

Our customer in this example has a custom application that makes use of the Filestream feature in SQL Server to store large sound files to disk and need to be read back at high speeds. Several third-party tools integrate with this SQL Server instance to provide advanced processing of the related meta-data and need to be installed locally. The vendors of these tools have yet to produce a version that works with Azure SQL Database.

**Solution:**

Unfortunately, the requirement for the Filestream feature rules out Azure SQL Database as it is not yet supported by this platform. Also, the need to install the third-party tools locally on the SQL Server rules out Azure SQL Database Managed Instance, whereas the full SQL Server instance is exposed to the end-user under Managed Instance, the underlying operating system is not. Therefore, in this example, the solution needs to use SQL Server on Azure VMs (IaaS), which offers a customized virtual environment to run SQL Server and includes full administrative rights to allow installation of third-party tools. The full specifications of SQL Server can be utilized, including support for up to 64TB of storage, as many databases per instance as needed, SQL Server transactional replication, Always On Availability Groups, Integration Services, Log Shipping to replicate data, and native SQL Server backups.

The downfall of using SQL Server on Azure VMs over Azure SQL Database is that many server maintenance and administration costs still exist, as does the need to manually configure and manage high availability, disaster recovery, and patching, creating extra administrative overhead. To migrate, Azure Site Recovery could be used to lift-and-shift the existing SQL Server to the Azure data center. This produces an exact replica of the server, complete with already installed and configured third-party tools which saves time and reduces the risk of mistakes when installing from scratch. The server data is synchronized to Azure in the background while the on-premises server remains online and available for service requests, with a minimal outage required to failover to the completely synchronized server image at an agreeable time.

#### Examples summary for selecting Landing Zone

The following tables summarize the scenarios from the previous examples where each of the target platforms are suited.

Common across Azure SQL platforms:

|  |  |  |
| --- | --- | --- |
| **Target Platform** | **Indicators to Look For** | **Benefits** |
| Azure SQL Database  Single Database  Azure SQL Database  Managed Instances | Current capacity or  management issues  Requires high availability | Compute and storage layers are provided and managed  Near limitless capacity available on demand  Premium tier available to meet higher performance and throughput requirements  Highly available as standard  Options available for regional high availability and disaster recovery protection  Azure manages backups, upgrades and patching  Azure provides automated analysis and recommendations for performance and security events.  Support for data encryption  Support for single sign on with Azure AAD |
| Azure SQL IaaS | Offers a customized  virtual environment to run SQL Server and includes full administrative rights to allow installation of third-party tools | Compatibility: When some application requires SQL Server, you know that SQL Server on Windows Server will work. Spend less time and expense of acquiring new on-premises hardware. Great option when migrating existing on-premises applications and Databases to Azure as-is. |