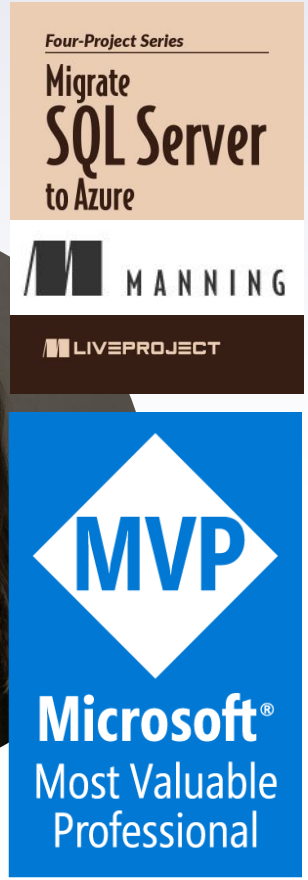
Migrate SQL Server to Azure MI Without Getting a Migraine

HELLO!

I am Josephine Bush

10+ years DBA experience
Microsoft Data Platform MVP

Manning liveProject Author
MBA IT Management
MS Data Analytics



@hellosqlkitty
sqlkitty.com

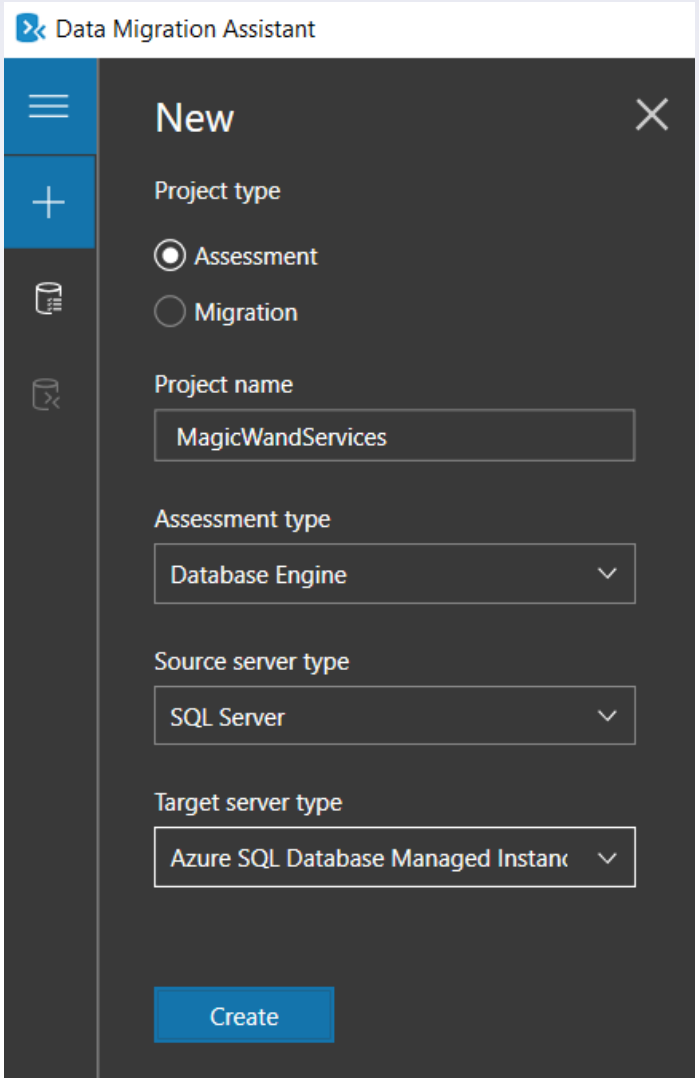
Database Migration Process

1 Prepare for Migration

- **Analyze/Audit**
  Using Microsoft Data Migration Assistant and SQL Server Audit
- **Clean Up**
  Remove unsupported and unused objects
- **Create Checklist**
  Catalogue items migrating to the cloud

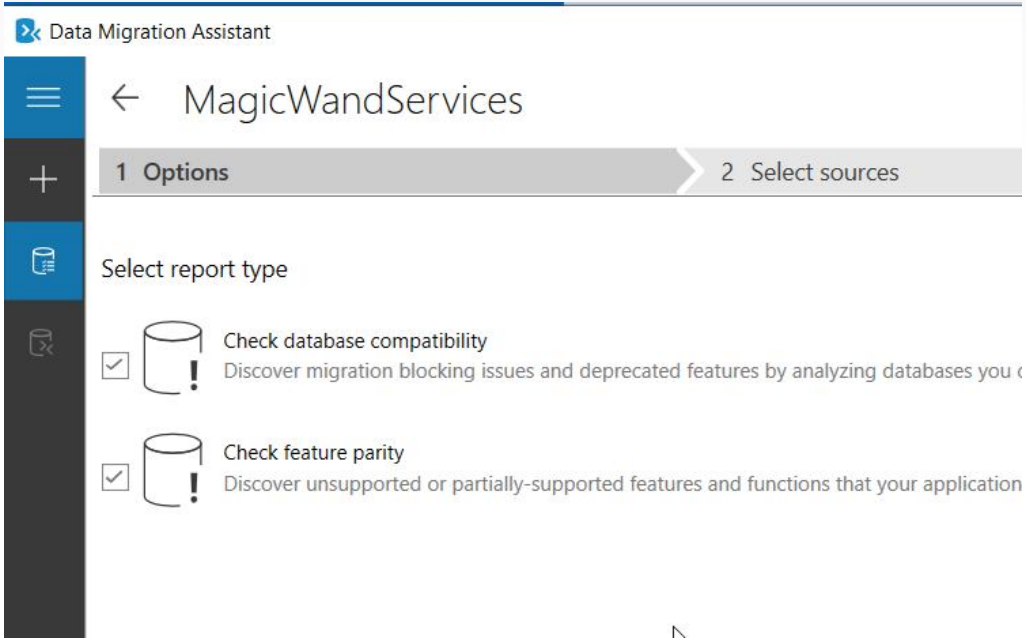
Microsoft Data Migration Assistant (DMA)

Enables you to upgrade to a modern data platform by detecting compatibility issues that can impact database functionality on your new version of SQL Server



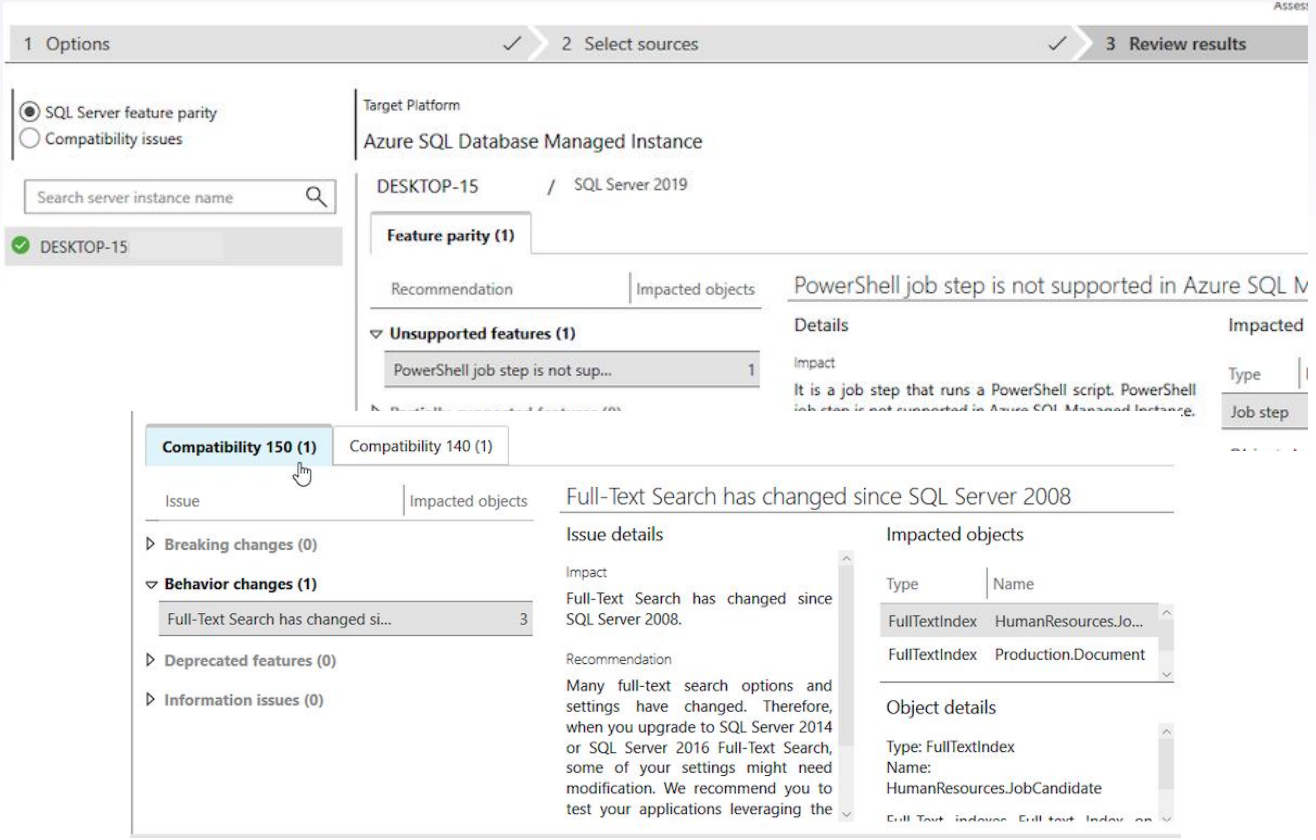
Note that the Microsoft Data Migration Assistant won't warn you about all issues you might encounter. **This is why it's always important to test your migrations from start to finish in a lower environment.**

DMA Configuration



Azure uses Azure Active Directory, as opposed to Active Directory. This means that when your users are logging in with a domain account, it will change the way they have to log in.

DMA Results


1 Options
2 Select sources
3 Review results
SQL Server feature parity
Compatibility issues
Search server instance name
DESKTOP-15
Target Platform
Azure SQL Database Managed Instance
DESKTOP-15 / SQL Server 2019
Feature parity (1)
Recommendation
Impacted objects
Unsupported features (1)
PowerShell job step is not sup... 1
PowerShell job step is not supported in Azure SQL M
Details
Impact
It is a job step that runs a PowerShell script. PowerShell
Impacted
Type
Job step
Compatibility 150 (1)
Compatibility 140 (1)
Issue
Impacted objects
Breaking changes (0)
Behavior changes (1)
Full-Text Search has changed si... 3
Deprecated features (0)
Information issues (0)
Full-Text Search has changed since SQL Server 2008
Issue details
Impact
Full-Text Search has changed since SQL Server 2008.
Recommendation
Many full-text search options and settings have changed. Therefore, when you upgrade to SQL Server 2014 or SQL Server 2016 Full-Text Search, some of your settings might need modification. We recommend you to test your applications leveraging the
Impacted objects
Type
Name
FullTextIndex HumanResources.Jo...
FullTextIndex Production.Document
Object details
Type: FullTextIndex
Name:
HumanResources.JobCandidate

SQL Server Audit Tips

- I use this on all prod servers to audit DDL and security changes
- For migrations, I may audit more to see if something is in use or not
- Be careful to not over audit

Prepare for Migration

- Analyze/audit
- Clean up
- Create checklist

2 Set Up Cloud Database

- **Create a Baseline**
  Check configuration settings with a SQL script
- **Create Azure SQL Managed Instance**
  Create MI via the Azure portal
- **Migrate Server-Level Objects**
  Using a PowerShell module named dbatools

SQL Managed Instance currently supports deployment only on the following types of subscriptions:

- Enterprise Agreement (EA)
- Pay-as-you-go
- Cloud Service Provider (CSP)
- Enterprise Dev/Test
- Pay-as-you-go Dev/Test
- Subscriptions with monthly Azure credit for Visual Studio subscribers

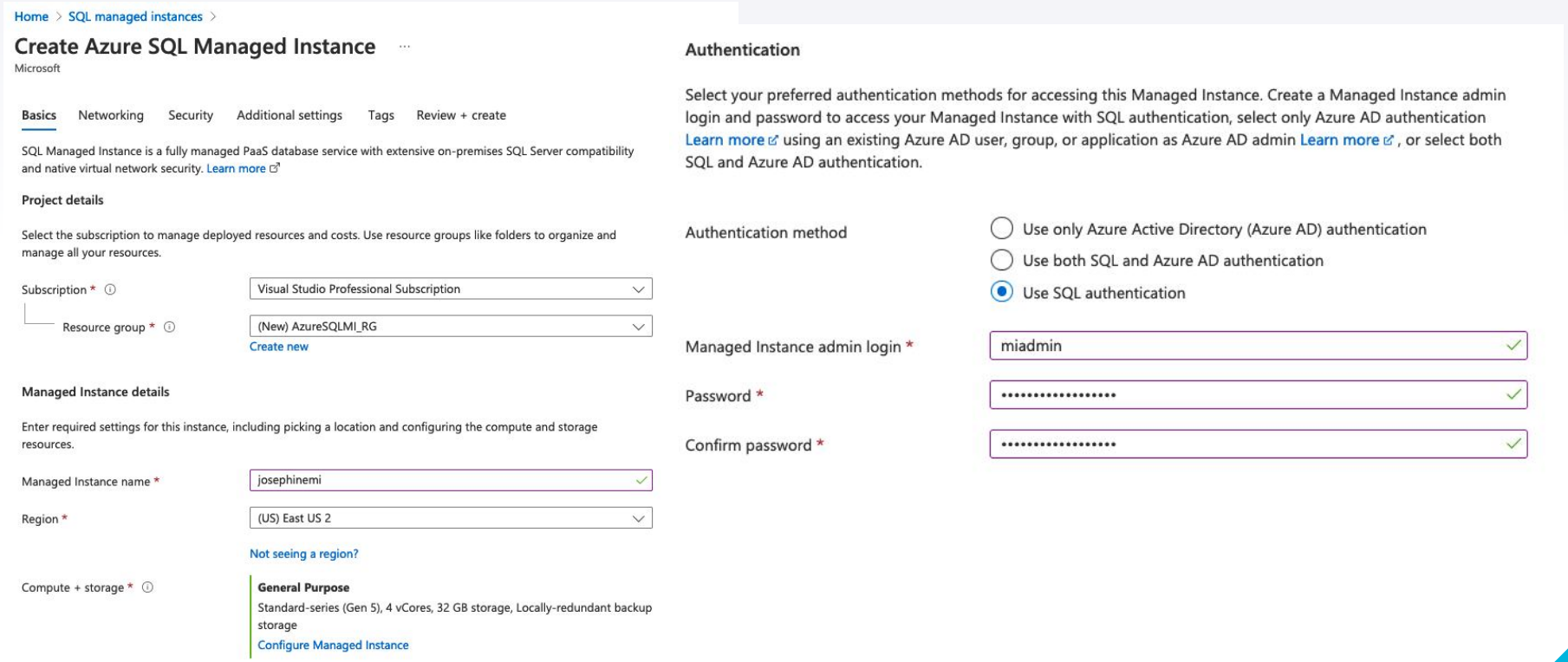
Create Configuration Baseline

```
/*setup for querying SQL Server configuration */
DECLARE @tf TABLE (TraceFlag nvarchar(35), status bit,global bit, session bit)
INSERT INTO @tf execute('DBCC TRACESTATUS(-1)');

DECLARE @config TABLE (
    name nvarchar(35),
    default_value sql_variant
)

/*not all of these settings are in all versions of sql server*/
INSERT INTO @config (name, default_value) VALUES
('access check cache bucket count',0),
('access check cache quota',0),
('ADR cleaner retry timeout (min)', 0),
('ADR Preallocation Factor', 0),
('Ad Hoc Distributed Queries',0),
('affinity I/O mask',0),
('affinity64 I/O mask',0),
('affinity mask',0),
('affinity64 mask',0),
('Agent XPs',0), --Changes to 1 when SQL Server Agent is started. Default value is
('allow filesystem enumeration', 0),
('allow polybase export', 0),
('allow updates',0),
('awe enabled',0),
('backup checksum default', 0),
```

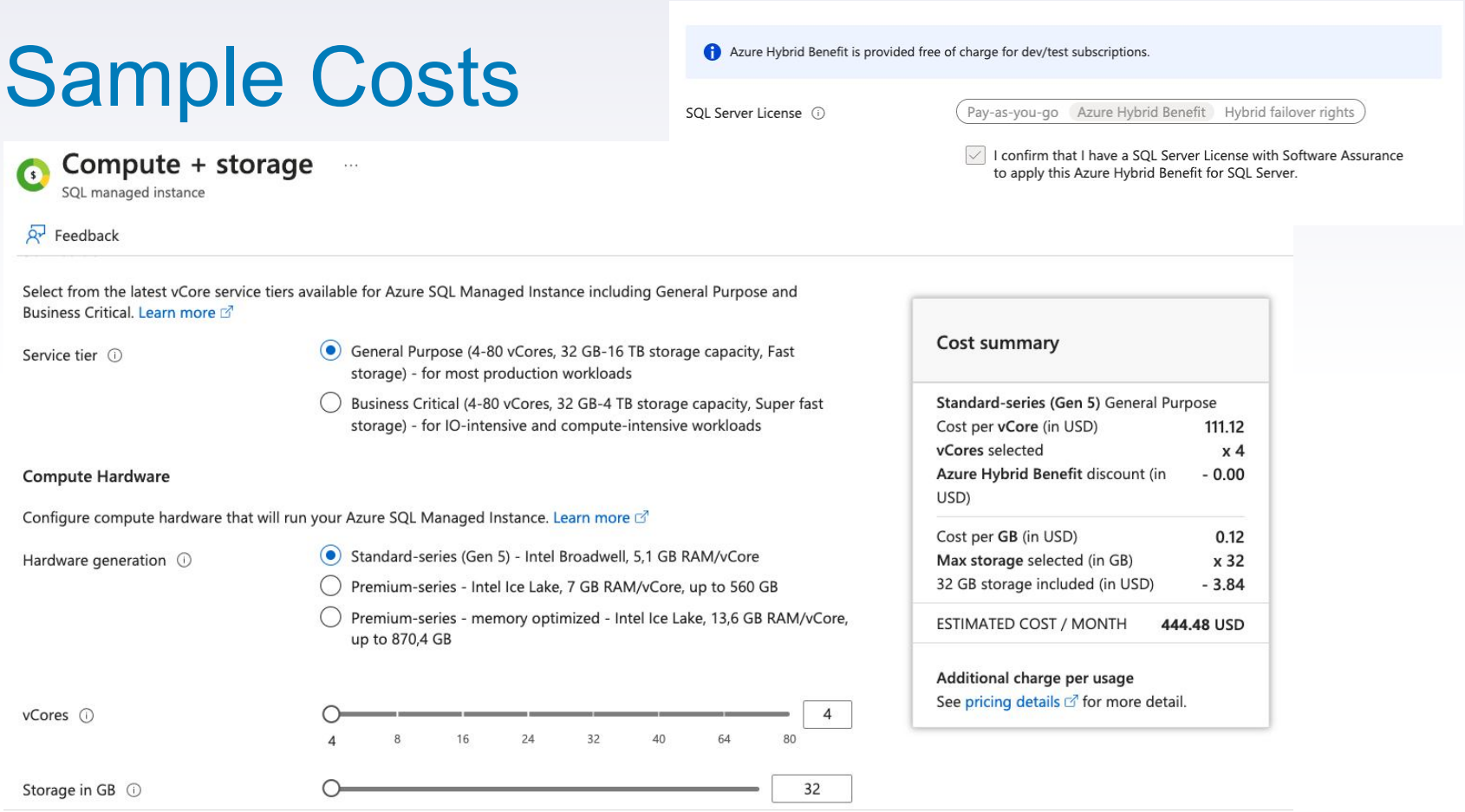
Very important information about costs

A single managed instance Standard-series Gen 5 will cost approximately $1/hour depending on your subscription

Create Managed Instance



Sample Costs

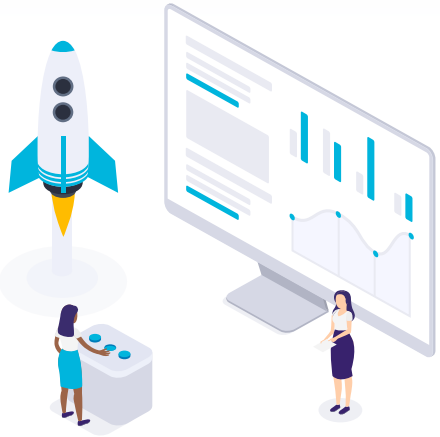


Azure SQL Managed Instance requires use of a dedicated vnet.

Make sure to check with your networking team to make sure you don’t configure this incorrectly.

Tips for Managed Instance

- It can take up to 6 hours to create. 30 min create is in preview, but not available in all subscriptions. Deployment of the first instance in the subnet takes longer than deploying into a subnet with existing managed instances.
- Start/stop is in preview, but not available in all subscriptions
- For testing, you can enable public endpoint, but don't do this for production
- To connect via SSMS with public endpoint: yourminame.**public**.xxxxxxxxxxxx.database.windows.net, 3342



Migrate Server-Level Objects

- Using dbatools
  - Install module in PowerShell

```
$scred = Get-Credential sa
$dcred = Get-Credential miadmin
$params = @{
  Source = "your_sql_server_name"
  Destination = "copy_your_mi_name_from_azure,3342"
  SourceSqlCredential = $scred
  DestinationSqlCredential =$dcred
}

Start-DbaMigration @params -Force -Exclude Databases -Verbose
```

Set Up Cloud Database

- Create baseline
- Create managed instance
- Migrate server-level objects

3 Migrate Databases To Azure

- **Set Up Database Migration Service (DMS)**
  Using Azure portal and Azure Data Studio (ADS)
- **Perform Online Migration**
  Using Azure Data Studio
- **Verify Migration**
  Check database configuration settings with a SQL script

Before You Set Up DMS

- Configure a VPN or ExpressRoute
- Register Microsoft.DataMigration
- Create a storage account
- Take a full backup of your on-premises database WITH CHECKSUM

VPN Setup

Connect to managed instance using Point-to-Site VPN connection

To setup environment using Azure Portal

Read documentation Quickstart: Configure a point-to-site connection to an Azure SQL Database managed instance from on-premises

To setup environment using PowerShell

1. Create VPN Gateway, configure it for Point-to-Site access and attach it to the virtual network. You could run PowerShell code shown below from your computer to automate this step.

```
$scriptUrlBase = 'https://raw.githubusercontent.com/Microsoft/sql-server-samples/master/samples/manage/azure-sql-db-managed-instance/attach-vpn-gateway'

$parameters = @{
    subscriptionId = '244eb28e-a9b8-42d4-9260-c0c553ae92e1'
    environmentName = 'AzureCloud'
    resourceGroupName = 'azuremi'
    virtualNetworkName = 'vnet-josephinemi'
    certificateNamePrefix  = 'cert4josephinemi'
}

Invoke-Command -ScriptBlock ([Scriptblock]::Create((iwr ($scriptUrlBase+'/attachVPNGateway.ps1?t='+ [DateTime]::Now.Ticks)).Content)) -ArgumentList $parameters, $scriptUrlBase
```

2. Install the VPN client and connect to virtual network
3. Run SSMS from the virtual machine and connect to Managed Instance. Below are the parameters needed to establish a connection

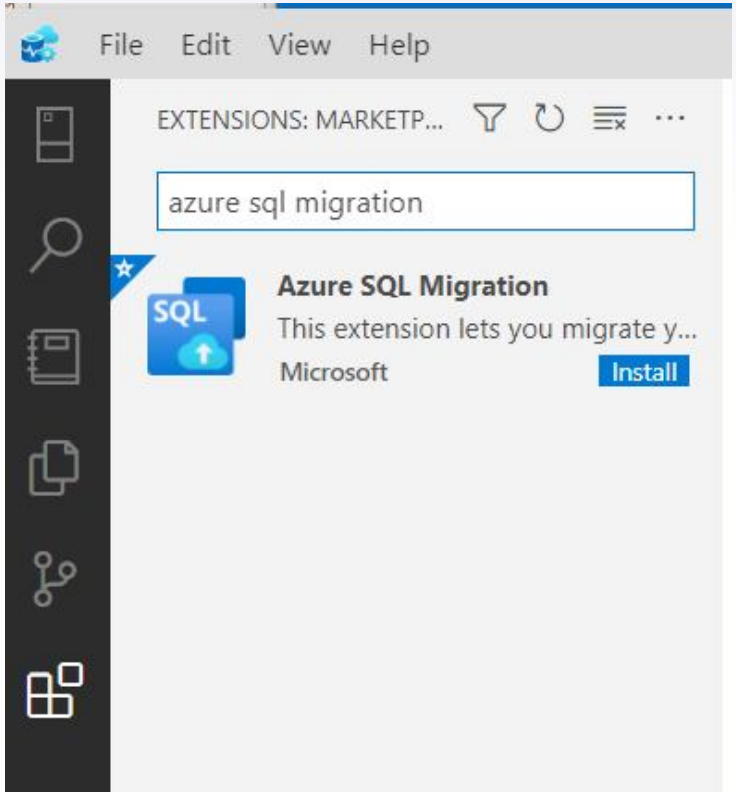
Login

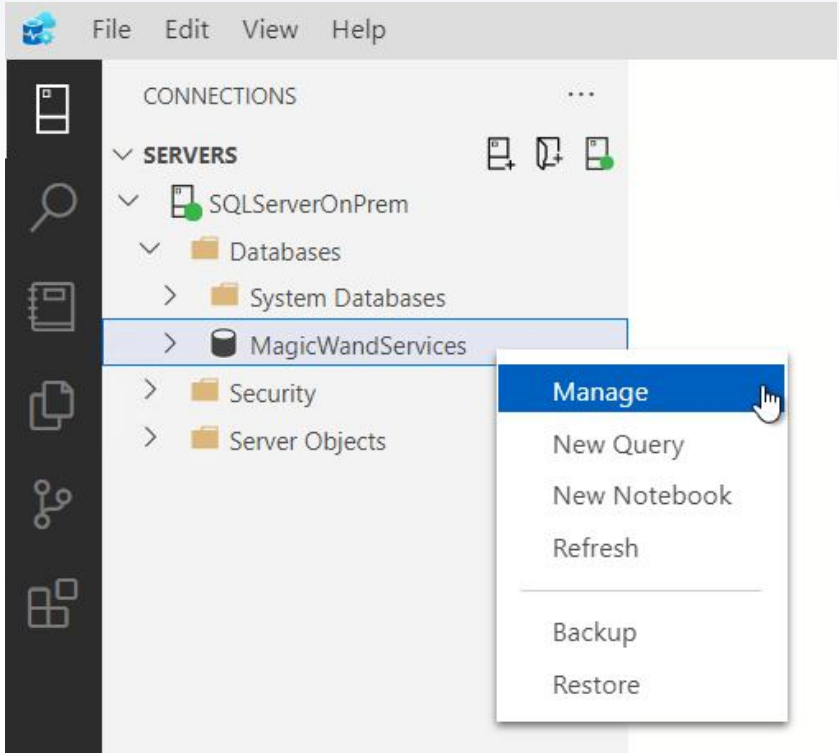
miadmin

Server name

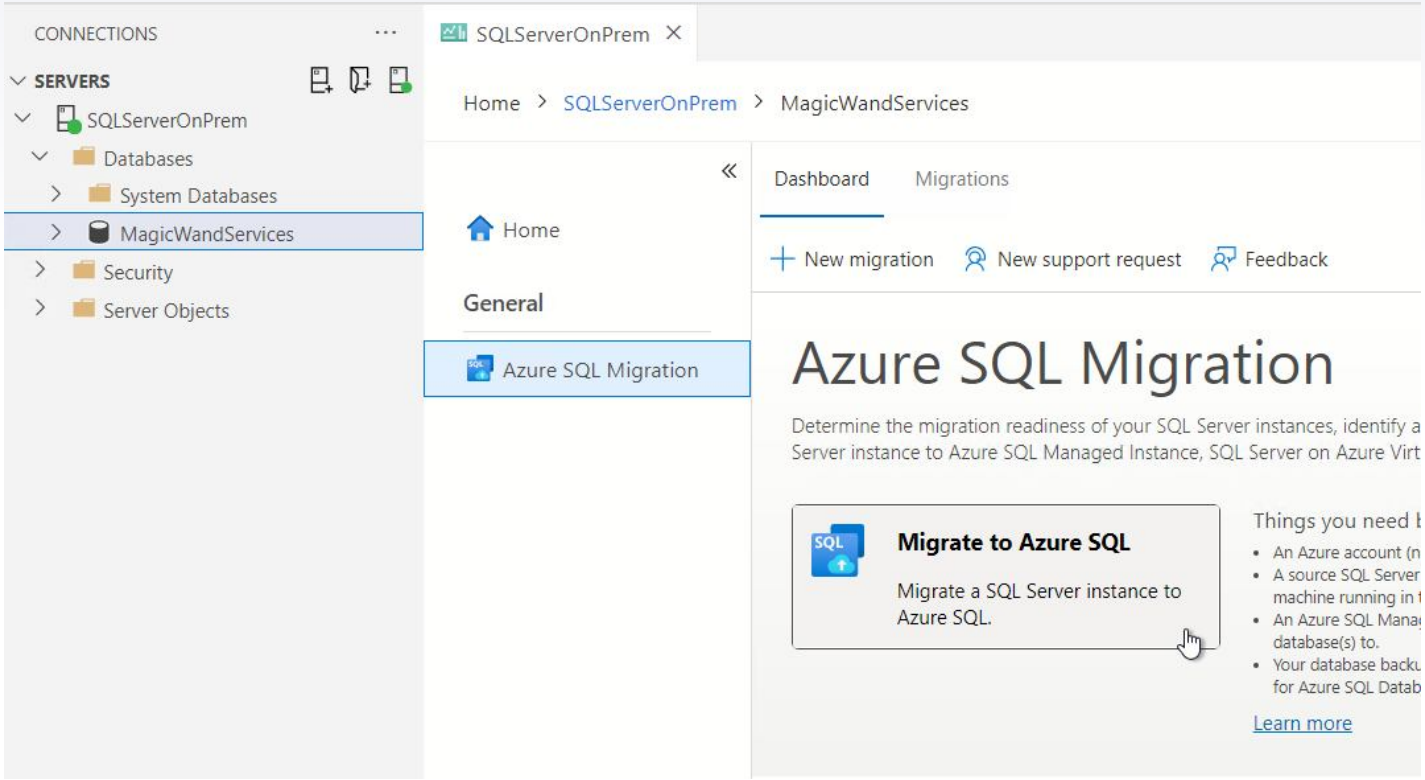
josephinemi.a57eaffdb3ca.database.windows.net

Setting up DMS in ADS





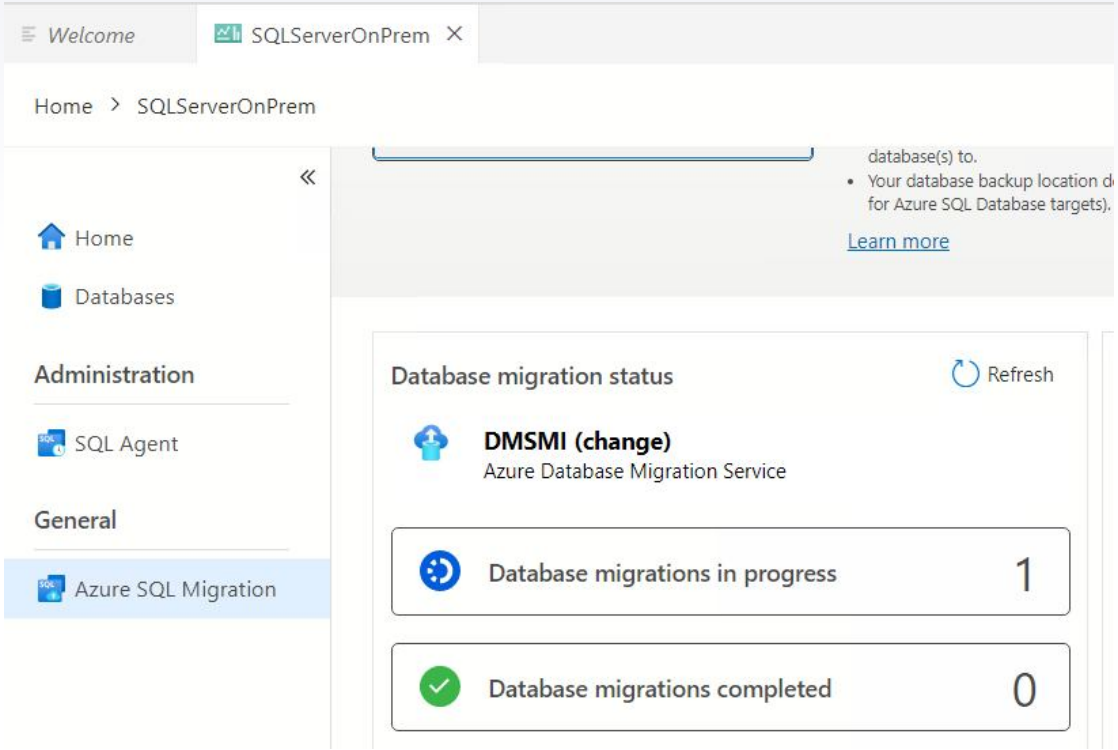
Set Up Migration in ADS



Steps to Set Up Migration

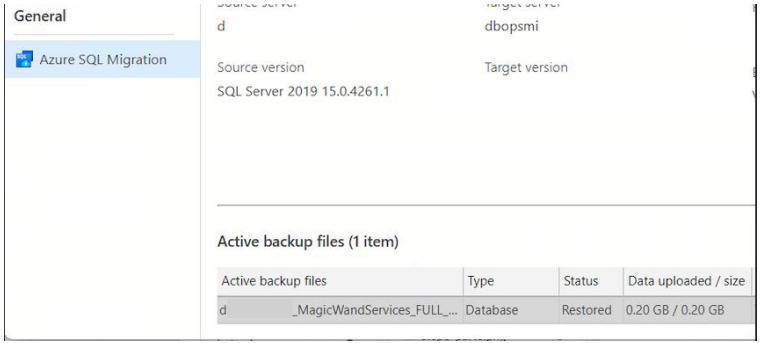
- Step 1: Databases for assessment
- Step 2: Assessment results and recommendations
- Step 3: Azure SQL target setup
- Step 4: Migration Mode
- Step 5: Data source configuration
- Step 6: Azure Database Migration Service – includes installing self hosted integration runtime
- Step 7: Summary

Migration Status in ADS



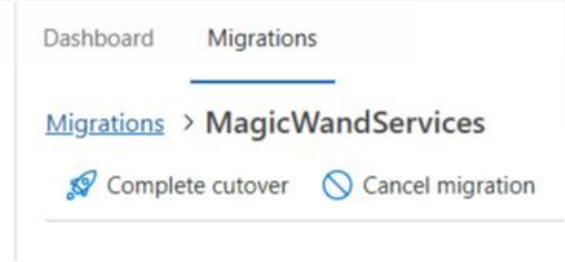
Begin Online Migration

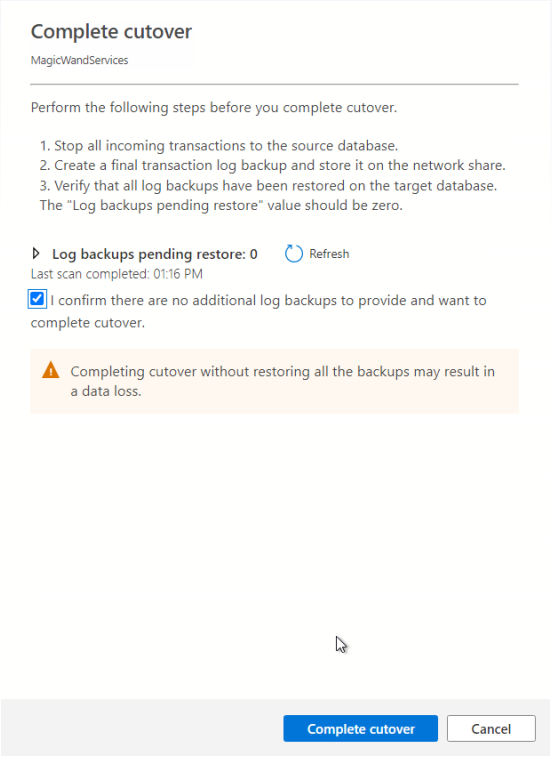
- Stop applications and users from using the on-premises database
- Take a final log backup WITH CHECKSUM
- Verify final backup is restored
- Don’t take your on-prem db offline



Complete Online Migration

- Click Complete Cutover





Verify Migration & Migrate DB Settings

```sql
DECLARE @dbname varchar(20);
SET @dbname = 'MagicWandServices';

DECLARE @config TABLE (
    name nvarchar(35),
    value sql_variant
)

INSERT INTO @config (name, value)
SELECT name, CASE
            WHEN value = 1 then 'ON'
            WHEN value = 0 then 'OFF'
            ELSE value
            END AS value
FROM sys.database_scoped_configurations
WHERE name <> 'MAXDOP'

INSERT INTO @config (name, value)
SELECT name, value
FROM sys.database_scoped_configurations
WHERE name = 'MAXDOP'


SELECT name = CONCAT('DBCONFIG:',dsc.name), dsc.value, is_value_default,
            'USE ' + @dbname +'; ALTER DATABASE SCOPED CONFIGURATION SET ' + dsc.name + '=' + convert(nvarchar(35), c.value) + ';' as SQLscript
FROM sys.database_scoped_configurations dsc
INNER JOIN @config c
ON c.name = dsc.name
WHERE is_value_default <> dsc.value
```

Migrate Databases to Azure

- Set up DMS
- Perform online migration
- Verify migration

4 Monitor and Audit

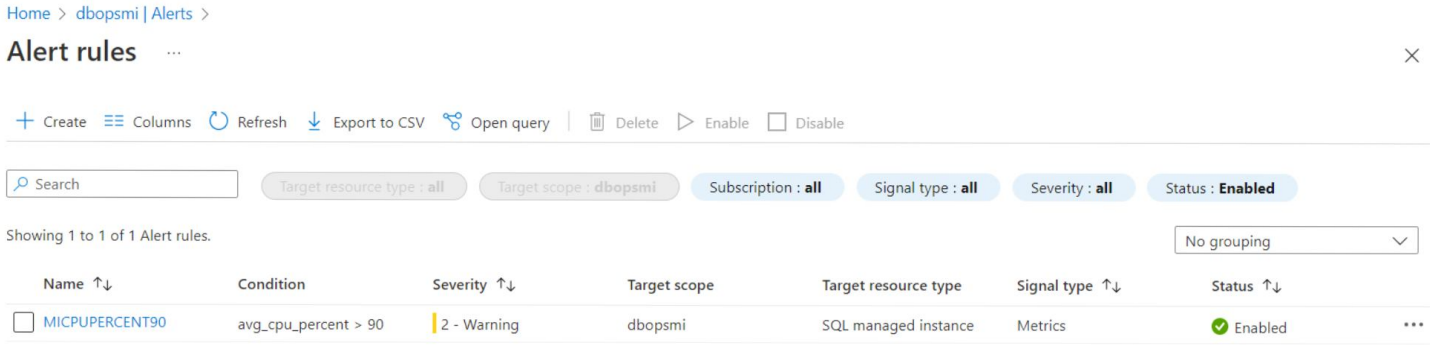
- **Configure Auditing**
  Using SQL Server Audit
- **Configure Alerting**
  Monitor and alert on things such as CPU usage
- **Security Check**
  Make sure your MI complies with CIS benchmarks

SQL Server Audit with MI tips

- Create a storage account to hold the audit files
- Use URL for audit destination
- Query that URL to get audit data

Monitoring/Alerting Tips

- Make sure to monitor and alert on things like
  - CPU usage
  - Memory usage
  - Storage space used
- Create an alert rule and alert action group



Security Check Tips

- Use the PowerShell module dbachecks
  - CIS benchmarks help you ensure your managed instance is as secure as possible.
  - Must install Pester module

```
$Date = Get-Date -Format "yyyy-MM-dd"
  Invoke-DbcCheck -Show Fails -SqlInstance "dbopsmi.public.rand0m1234.database.windows.net,3342" -SqlCredential (Get-Credential yourmiadmin) `
  -ExcludeCheck Backup, HADR, Domain, LogShipping, AgentServiceAccount, IdentityUsage, FutureFileGrowth, FKCKTrusted, GuestUserConnect, `
  ValidDatabaseOwner, InvalidDatabaseOwner, InstanceConnection, SqlEngineServiceAccount, TempDbConfiguration, BackupPathAccess, DefaultFilePath, `
  DAC, MaxMemory, OrphanedFile, ServerNameMatch, MemoryDump, SupportedBuild, DefaultBackupCompression, ErrorLog, CrossDBOwnershipChaining, DefaultTrace, `
   OLEAutomationProceduresDisabled, RemoteAccessDisabled, SystemFull, UserFull, UserDiff, Userrog `
   -Passthru | Convert-DbcResult | Set-DbcFile -FilePath C:\windows\temp\ -FileName DbcCheck_$Date -FileType csv
```

Monitor and Alert

- Configure auditing and alerting
- Security check

Migrating to MI

Prepare to Migrate

Analyze and audit your on-premises database server, clean up unused and unsupported objects, and prepare a checklist

Set Up Cloud Database

Create a baseline of your on-premises db server, create MI, and migrate server-level objects

Migrate Databases to Azure

Create DMS, do online migration, and verify migration

Monitor and Alert

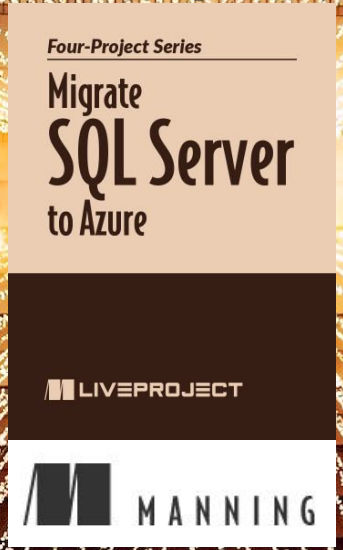
Configure auditing, alerting, and do a security check

100%

Total success!

Want the whole picture?

Get the Manning liveProject Series.



Resources

- Manning liveProject Series
- Migrate Without a Migraine GitHub Repository
- Managed Instance Features
- SQL Server Audit Presentations and Code
- dbatools PowerShell module
- dbachecks PowerShell module

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Illustrations by Sergei Tikhonov
- Photographs by Unsplash

THANKS!

Any questions?

You can find me at:

- @hellosqlkitty
- hellosqlkitty@gmail.com
- https://sqlkitty.com/