

# Contents

[Contents 1](#_Toc529954981)

[Prerequisites 2](#_Toc529954982)

[Managed Instance 2](#_Toc529954983)

[Azure VM 2](#_Toc529954984)

[Azure VM login 2](#_Toc529954985)

[SQL Server login 3](#_Toc529954986)

[Managed Instance login 3](#_Toc529954987)

[Demo - Migrate web app database 5](#_Toc529954988)

[Part 1 - Show a web app connected to an on-premises SQL Server database 5](#_Toc529954989)

[Part 2- Migrate the database using BACKUP / RESORE 6](#_Toc529954990)

[Part 3 - Migrate using Azure Database Migration Service 6](#_Toc529954991)

[Part 4 - Point web app to newly migrated database 9](#_Toc529954992)

[Demo - Threat detection 10](#_Toc529954993)

[To prepare (first time only): 10](#_Toc529954994)

[Demo flow and speaking points: 11](#_Toc529954995)

[Deploy to your own environment 11](#_Toc529954996)

# Prerequisites

For your demo environment you will need a general-purpose managed instance and a VM or on-premises machine connected to your VNet in Azure. These resources take considerable time to provision and setup. Plus, they will put a dent in your demo wallet (Azure subscription) if left one. It is recommended to put them all in one resource group to make it easy to delete.

### Managed Instance

QuickStart: Create an Azure SQL Database Managed Instance

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

**Note:** For the first instance in a subnet, deployment time is typically much longer than in case of the subsequent instances. Do not cancel deployment operation because it lasts longer than you expected. Creating the second Managed Instance in the subnet only takes a couple of minutes.

### Azure VM

QuickStart: Create a SQL Server 2017 Windows virtual machine in the Azure portal

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

**Note:** Deploy the VM in the same VNet as the Managed Instance, but in a separate subnet.

Install the following tools on the VM

* Latest version of SSMS. <https://docs.microsoft.com/en-us/sql/ssms>
* Visual Studio Community. <https://visualstudio.microsoft.com/downloads/>
* Latest version of Data Migration Assistant. <https://docs.microsoft.com/en-us/sql/dma/dma-overview>
* PowerShell. Install the latest version of powershellget and azurerm.sql on the VM

<https://www.powershellgallery.com/packages/PowerShellGet/> <https://www.powershellgallery.com/packages/AzureRM.Sql>

Unzip the following demo aps / scripts on the VM

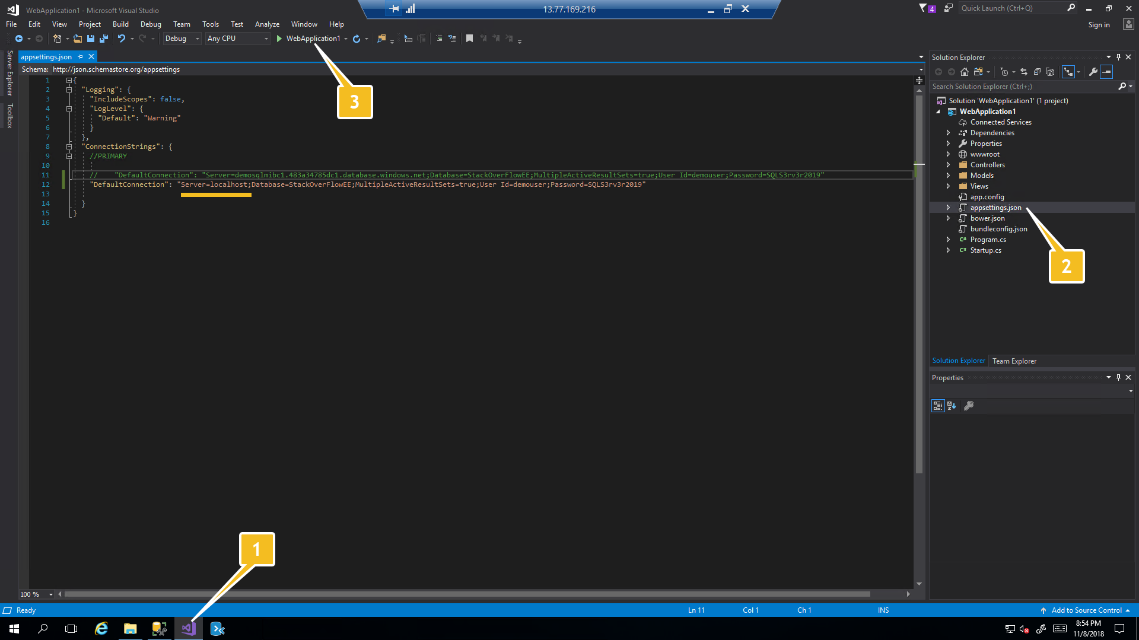
* StackOverFlow.zip
* scripts.zip

# Demo - Migrate web app database

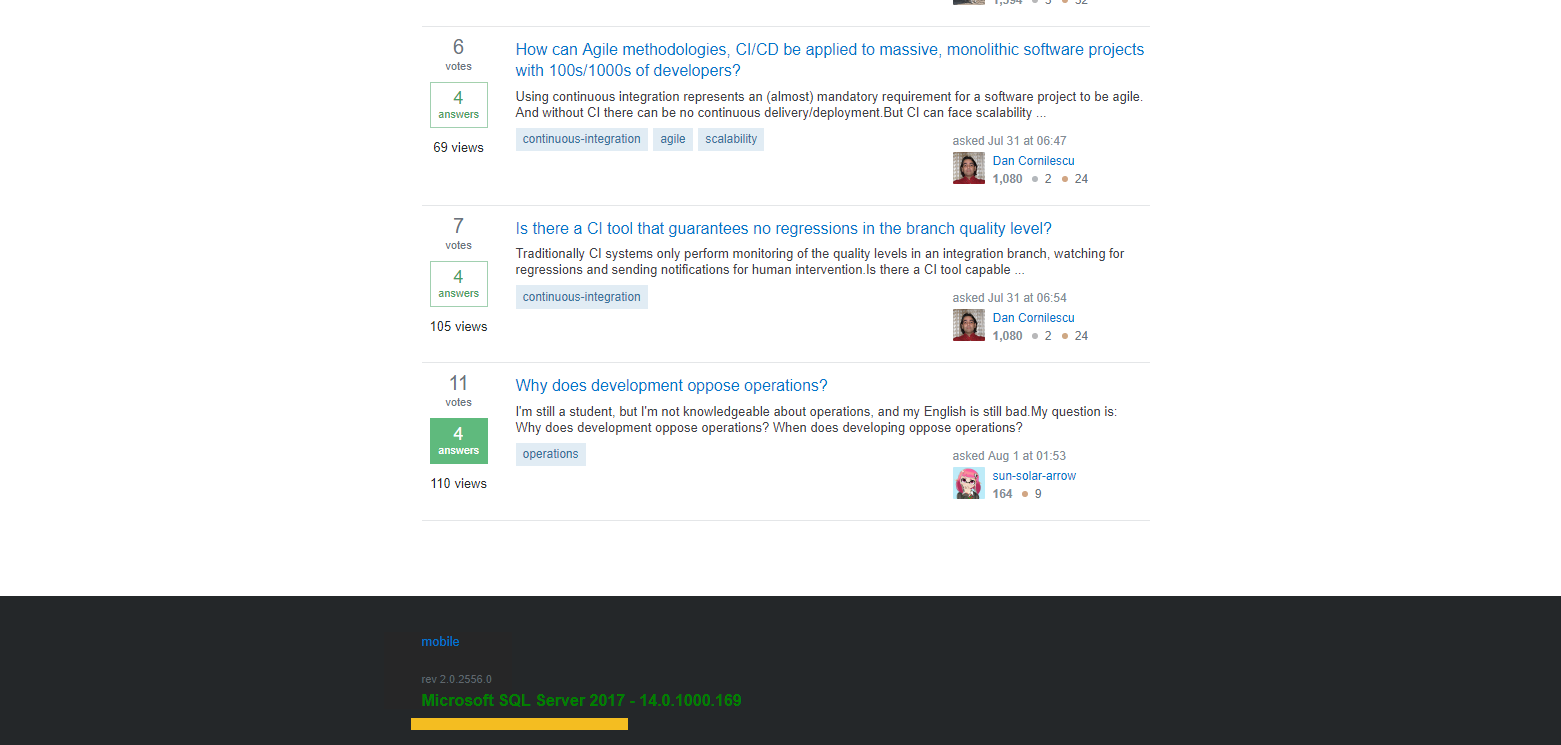
This demo migrates an on-premise SQL database to Azure SQL DB Managed Instance without the need to change the application apart from simply modifying the sql server connection string.

### Part 1 - Show a web app connected to an on-premises SQL Server database

1. Open visual studio and open project WebApplication1 included with StackOverFlow.zip
2. Expand the project in the solutions explorer and select **appsettings.json**
   1. Note the SQL server is the local host
3. Run the project (Ctrl + F5) or select run Webapplication1



Notice the bottom of the app show the SQL version (running on-premises)



1. Close the web app.

### Part 2- Migrate the database using BACKUP / RESORE

1. Open SSMS
2. Connect to local SQL Server
3. Run scripts to create shared access key (1 time only)
4. Run script to BACKUP ‘StackOverFlowEE’ database to Azure blob storage
5. Connect to local Managed Instance
6. Run script to create same shared access key on MI (1 time only)
7. Run script to RESTORE ‘StackOverFlowEE’ database to MI

Script 1 – Create SAS Credential

IF EXISTS (select \* from sys.credentials where name = N'https://storagebackupsfordemo.blob.core.windows.net/sqlbackups')

DROP CREDENTIAL [https://storagebackupsfordemo.blob.core.windows.net/sqlbackups] -- Example: [https://mystorage.blob.core.windows.net/mysqlbackups];

CREATE CREDENTIAL [https://storagebackupsfordemo.blob.core.windows.net/sqlbackups]

WITH IDENTITY = 'SHARED ACCESS SIGNATURE'

,SECRET = 'sv=2017-11-09&ss=b&srt=sco&sp=rwdlac&se=2019-02-01T02:46:01Z&st=2018-11-06T18:46:01Z&spr=https&sig=U7hMCPQXZSjVrKBCaRSih8O%2FzXGZAMPA4%2BdBwparQqE%3D';

GO

Script 2 - Backup database to blob storage

BACKUP DATABASE [StackOverFlowEE]

TO URL = N'https://storagebackupsfordemo.blob.core.windows.net/sqlbackups/StackOverFlowEE.bak'

WITH FORMAT, INIT

GO

Script 3 – Restore database from blob storage

IF EXISTS (select \* from sys.databases where name = 'StackOverFlowEE')

DROP DATABASE StackOverFlowEE

GO

RESTORE DATABASE StackOverFlowEE

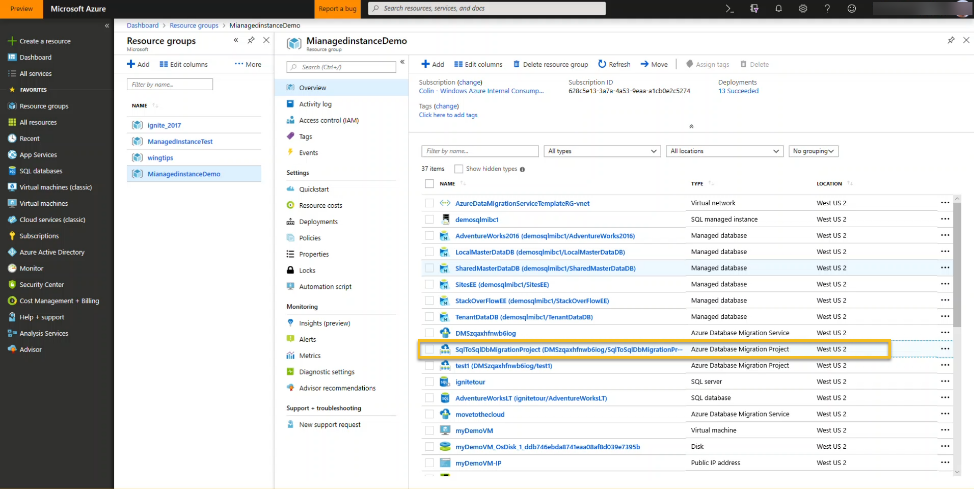
FROM URL = N'https://storagebackupsfordemo.blob.core.windows.net/sqlbackups/StackOverFlowEE.bak'

GO

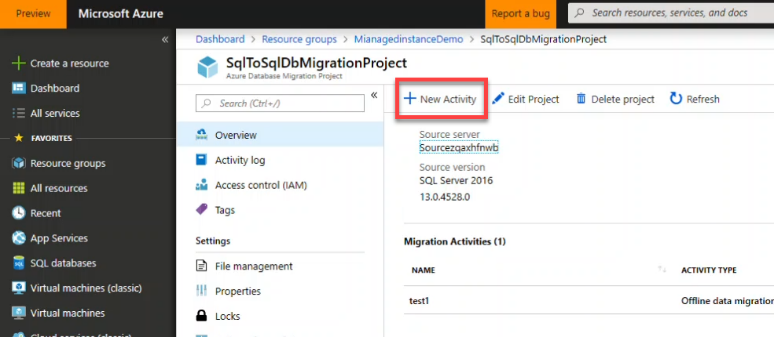
Reference link(s): <https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/sql-server-backup-and-restore-with-microsoft-azure-blob-storage-service?view=sql-server-2017>

### Part 3 - Migrate using Azure Database Migration Service

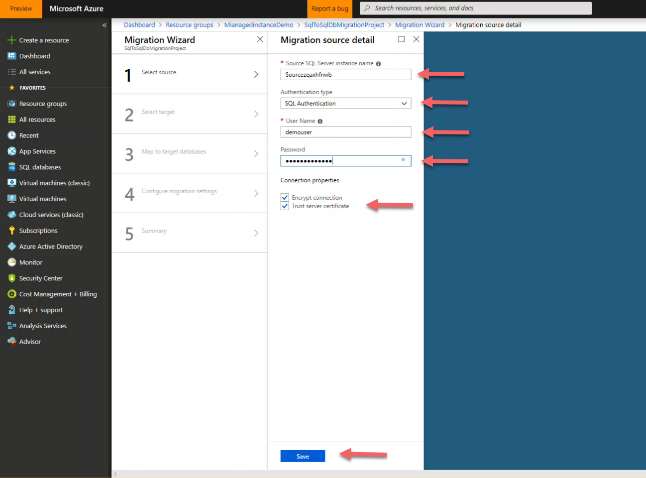
1. Select the Azure Data Migration project that has already been setup



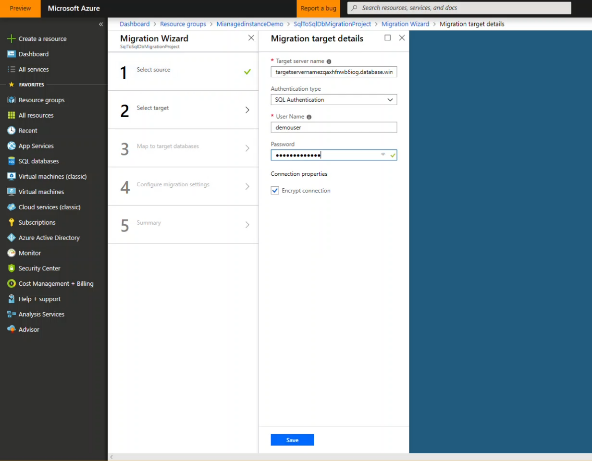
1. Select “New Activity”



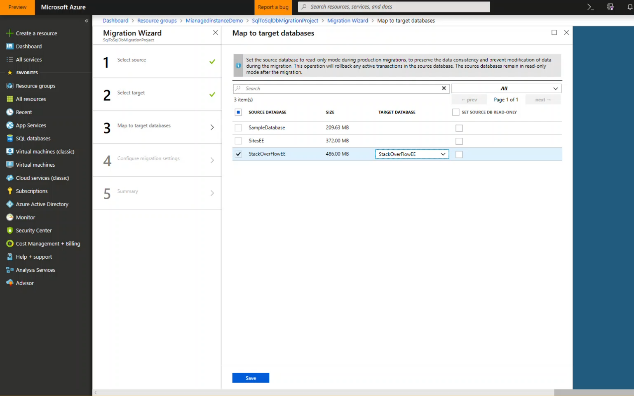
1. Enter the source details
   1. Source: “Sourcezqaxhfnwb”
   2. Authentication: “SQL Authentication”
   3. User: “demouser”
   4. Password: “SQLS3rv3r2019”



1. Enter the target details
   1. Target: “targetservernamezqaxhfnwb6iog.database.windows.net”
   2. Authentication: “SQL Authentication”
   3. User: “demouser”
   4. Password: “SQLS3rv3r2019



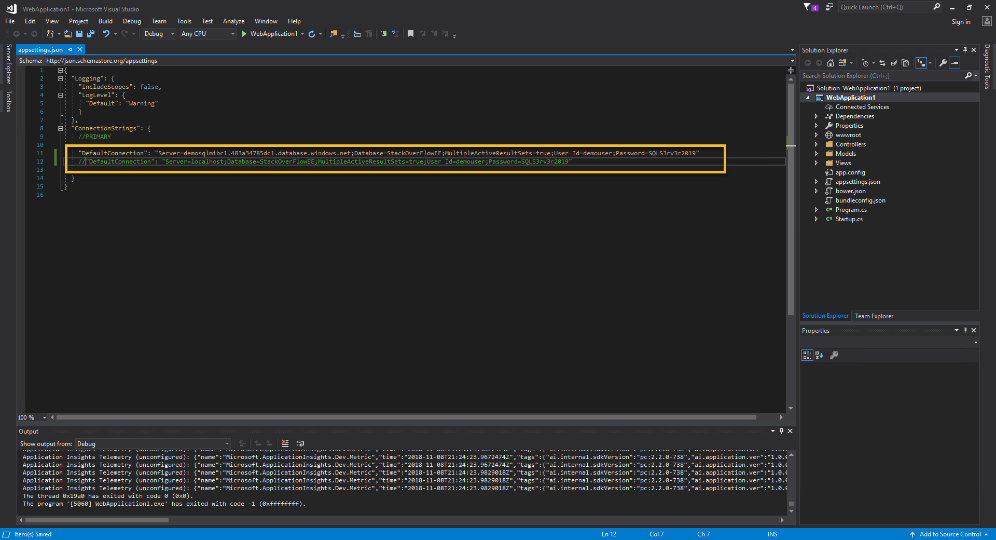
1. Map target database
   1. Select source database “StackOverFlowEE”
   2. Select target database “StackOverFlowEE”. This is pre-created schema created using Data Migration Assistance.



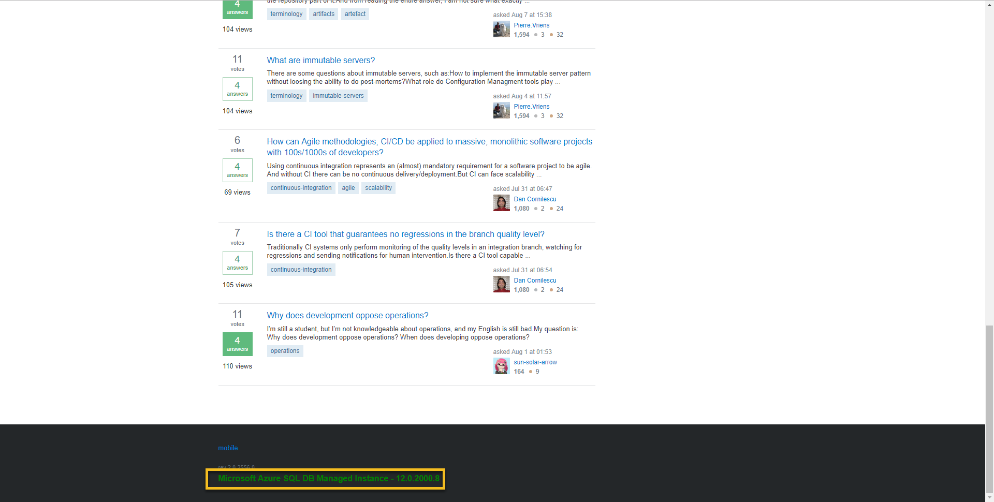
1. Select the table to migrate
2. Map target database
3. Select the tables to migrate
4. Run Migration
   1. Select do not validate database
   2. Run migration
   3. Refresh to see status of table migrated

### Part 4 - Point web app to newly migrated database

1. Simple change the connection string in the **appsettings.json** file to point to your Managed Instance and re-run the app. Comment out exist connection string and uncomment out one above that points to the Managed Instance



Scroll to the bottom of the app an note the SQL version is now Azure SQL DB Managed Instance

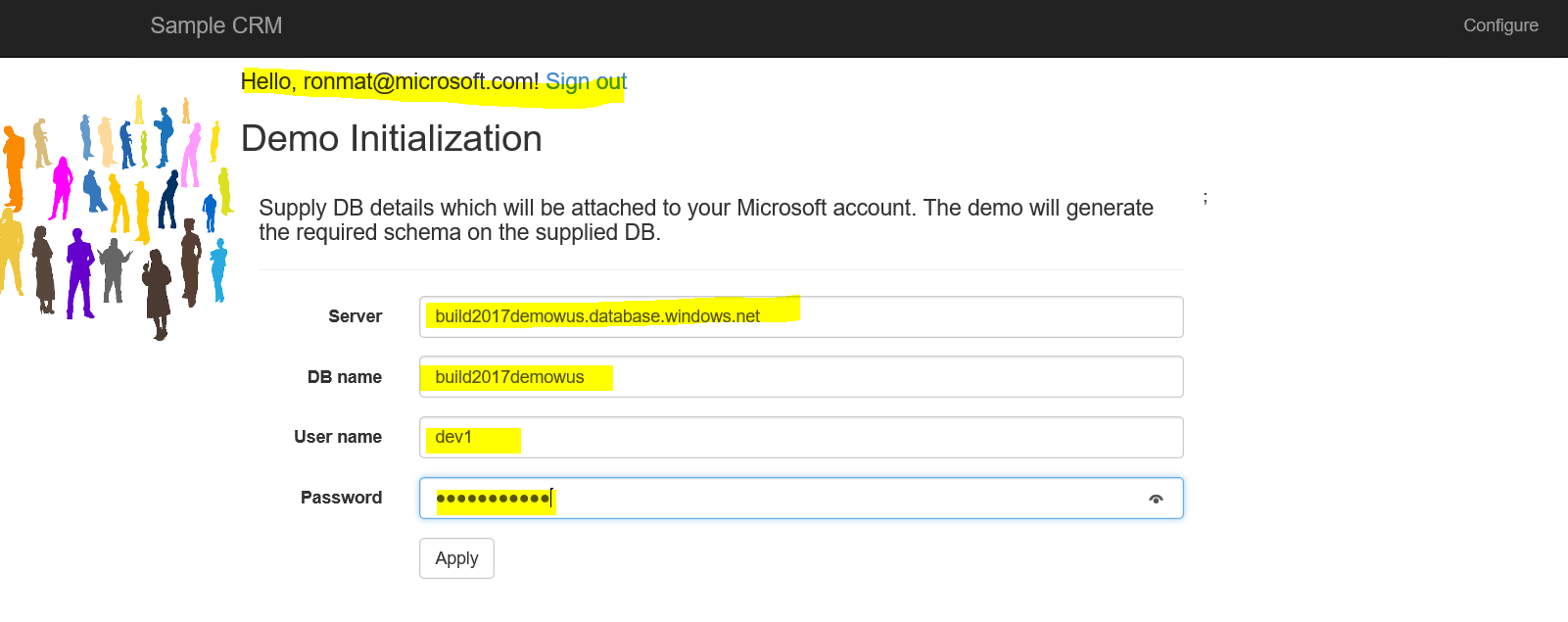


# Demo - Threat detection

### To prepare (first time only):

1. **Create an empty database** in SQL Database under your subscription
2. **Click on ‘Configure’** option in this portal <https://mybadwebsite.azurewebsites.net/Home/Init> , enter the details of the database you have created and click on “Apply”

\*\*The application will create a CRM scheme and will upload some CRM data into your database



E.g.

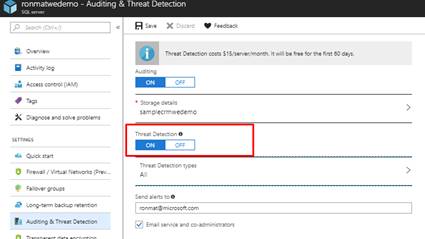
ignitetour.database.windows.net

AdventureWorksLT

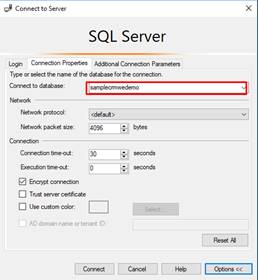
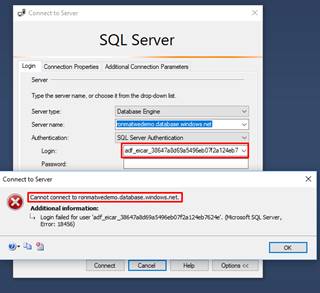
demouser

SQLS3rv3r2019

1. **Enable Auditing & Threat Detection** from Azure portal and set your real email to receive threat detection email notifications <https://ms.portal.azure.com>



1. **Simulate SQL injection attack** by injecting **' OR 1 = 1 –** text into the user field<https://mybadwebsite.azurewebsites.net/Home/Customers>  and check that you receive a SQLi alert email after 1-5 minutes (once every 30 minutes)
2. **Simulate behavioral alerts** by trying to log in using SSMS to your database using the following user adf\_eicar\_38647a8d69a5496eb07f2a124eb7624e and check that you get SQL behavioral alerts after 2 hours



### Demo flow and speaking points:

1. **Navigate to our demo app & use a SQL injection** technique to bypass the authentication: <https://mybadwebsite.azurewebsites.net/>

Log in as an app user. Use the credentials:

* + For Username: **'OR 1=1—**
  + For password: \*anything\*
* We successfully performed a SQL injection and broke into this app.

1. **Explore Threat Detection alerts**
2. Go to your Inbox and find a Threat Detection alert. It provides information on the

suspicious database activity that was detected, possible causes and recommended actions.

1. Click on ‘[View recent SQL alerts’ link](https://na01.safelinks.protection.outlook.com/?url=https:%2F%2Ftrack.azure.com%2Ff%2Fa%2FyjLpzvX_P1w6_CFto0DElw~~%2FAAAAAQA~%2FRgRcBjR_-vI1psE0GtQQgArQG5L6BrBlIUcm9ubWF0QG1pY3Jvc29mdC5jb20JUQQAAAAAR8d7IlNlbnREYXRlS2V5IjoiMjAxNzEyMDMwODAyMDYiLCJCYXRjaElkIjoiNzM5NWQ4MDktNDlkMC00YjIzLWE1MDQtZDliMjVhOTkxOWYxIiwiRW1haWxJZCI6IjYxYTAwZTBhLTdiMTQtNGJlMi1hNTVlLTQ1ZmI1ZGM1YjJjZSIsImJpbmRpbmciOiJuZXdkZWRpY2F0ZWQiLCJUZW1wbGF0ZUlkIjoidGVtXzRrTUo4bUtxZGY2UnBySGR5RHF3cFBxUCJ9Ew~~&data=02%7C01%7Cronmat%40microsoft.com%7C80fd6fa7b7f64fff8bb208d53a2428aa%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C636478849332527598&sdata=OIe2QT2bbCRKLg1DdhbyT0Laraxh0W2%2B4Z0V5mfYZiI%3D&reserved=0)  in the e-mail alert (or click on SQL security notifications tile in the database blade) , which launches Azure alerts blade which shows all the active threats on your database.
2. Clicking on a specific alert provides additional details and clear investigating and remediation steps for mitigating the threat
3. SQL Injection alerts

|  |  |
| --- | --- |
| Investigation steps | [View the vulnerable SQL statement](https://ms.portal.azure.com/) |
| Remediation steps | [Read more about SQL Injection threat and how to fix the vulnerable application code](https://www.microsoft.com/en-us/download/details.aspx?id=13759). |

1. Brute Force attack, GEO/User/Application Anomaly alerts

|  |  |
| --- | --- |
| Investigation steps | [View suspicious activity to validate whether this suspicious activity is expected.](https://na01.safelinks.protection.outlook.com/?url=https:%2F%2Ftrack.azure.com%2Ff%2Fa%2FhT5wZGoyjNhUUAMi8tqPIw~~%2FAAAAAQA~%~&data=02%7C01%7Cronmat%40microsoft.com%7Cb07b89932e744b98e6f408d52c70b971%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C636463785016759614&sdata=U2qWhkiT4k2n36sMOcQcAuvwnqptfzXzIk3RxMZsPO4%3D&reserved=0) |
| Remediation steps | [Lock down firewall as tightly as possible](https://ms.portal.azure.com/) and ensure you enforce the use of strong passwords and do not re-use them across multiple databases |

\*\* We are adding more and more sophisticated detection algorithms all the time.

### Deploy to your own environment

You can deploy to your own environment with the full source code for this demo application that’s published on github

<https://github.com/Microsoft/azure-sql-security-sample>