Get to know Azure
Explore the world of cloud computing and learn what sets Azure apart

Azure Developer Series

Migrating a dotnetcore 2-tier application to Azure, using different architectures and DevOps best practices

Hands-On-Labs step-by-step guides

Prepared by:

Peter De Tender

CEO and Lead Technical Trainer
PDTIT and 007FFFLearning.com

@pdtit @007FFFLearning

Version: Sept 2019 – 1.0

Contents

Migrating a dotnetcore 2-tiered application to Azure using different architectures and DevOps best practices - Hands-On-Labs step-by-step

You are part of an organization that is running a dotnetcore e-commerce platform application, using Windows Server infrastructure on-premises today, comprising a WebVM running Windows Server 2012 R2 with Internet Information Server (IIS) and a 2nd SQLVM running Windows Server 2012 R2 and SQL Server 2014.

The business has approved a migration of this business-critical workload to Azure, and you are nominated as the cloud solution architect for this project. No decision has been made yet on what the final architecture should or will look like. Your first task is building a Proof-of-Concept in your Azure environment, to test out the different architectures possible:

- Infrastructure as a Service (IAAS)
- Platform as a Service (PAAS)
- Containers as a Service (CaaS)

At the same time, your CIO wants to make use of this project to switch from a more traditional mode of operations, with barriers between IT sysadmin teams and Developer teams, to a DevOps way of working. Therefore, you are tasked to explore Azure DevOps and determine where CI/CD Pipelines can assist in optimizing the deployment and running operations of this e-commerce platform, especially when deploying updates to the application.

As you are new to the continuous changes in Azure, you want to make sure this process goes as smooth as possible, starting from the assessment to migration to day-to-day operations.

Abstract and Learning Objectives

This workshop enables anyone to learn, understand and build a Proof of Concept, in performing a multi-tiered .Net Core web application using Microsoft SQL Server database, platform migration to Azure public cloud, leveraging on different Azure Infrastructure as a Service, Azure Platform as a Service (PaaS) and Azure Container offerings like Azure Container Instance (ACI) and Azure Kubernetes Services (AKS).

After an introductory module on cloud app migration strategies and patterns, students get introduced to the basics of automating Azure resources deployments using Visual Studio and Azure Resource Manager (ARM) templates. Next, attendees learn about the importance of performing proper assessments, and what tools Microsoft offers to help in this migration preparation phase. Once the application has been deployed on Azure Virtual Machines, students learn about Microsoft SQL database migration to SQL Azure PaaS, as well as deploying and migrating web applications to Azure Web Apps.

After these foundational platform components, the workshop will totally focus on the core concepts and advantages of using containers for running business workloads, based on Docker, Azure Container Registry (ACR), Azure Container Instance (ACI) and WebApps for Containers, as well as how to enable container orchestration and cloud-scale using Azure Kubernetes Service (AKS).

In the last part of the workshop, students get introduced to Azure DevOps, the new Microsoft Application Lifecycle environment, helping in building a CI/CD Pipeline to publish workloads using the DevOps principals and concepts, showing the integration with the rest of the already touched on Azure services like Azure Web Apps and Azure Kubernetes Services (AKS), closing the workshop with a module on overall Azure monitoring and operations and what tools Azure has available to assist your IT teams in this challenge.

The focus of the workshop is having a Hands-On-Labs experience, by going through the following exercises and tasks:

- Deploying a 2-tier Azure Virtual Machine (Webserver and SQL database Server) using ARM-template automation with Visual Studio 2019;
- Publishing a .NET Core e-commerce application to an Azure Web Virtual Machine and SQL DB Virtual Machine;
- Performing a proper assessment of the as-is Web and SQL infrastructure using Microsoft Assessment Tools;
- Migrating a SQL 2014 database to Azure SQL PaaS (Lift & Shift);
- Migrating a .NET Core web application to Azure Web Apps (Lift & Shift);
- Containerizing a .NET Core web application using Docker, and pushing to Azure Container Registry (ACR);
- Running Azure Container Instance (ACI) and WebApp for Containers;
- Deploy and run Azure Azure Kubernetes Services (AKS);

- Deploying Azure DevOps and building a CI/CD Pipeline for the subject e-commerce application;
- Managing and Monitoring Azure Kubernetes Services (AKS);

Requirements

Naming Conventions:

IMPORTANT: Most Azure resources require unique names. Throughout these steps you will see the word **"[SUFFIX]"** as part of resource names. You should replace this with your initials, guaranteeing those resources get uniquely named.

Azure Subscription:

Participants need a "pay-as-you-go", MSDN or other paid Azure subscription

a) In one of the Azure Container Services tasks, you are required to create an Azure AD Service Principal, wich typically requires an Azure subscription owner to log in to create this object. If you don't have the owner right in your Azure subscription, you could ask another person to execute this step for you.

b) The Azure subscription must allow you to run enough cores, used by the baseline Virtual Machines, but also later on in the tasks when deploying the Azure Container Services, where ACS agent and master machines are getting set up. If you follow the instructions as written out in the lab guide, you need 12 cores.

c) If you run this lab setup in your personal or corporate Azure payable subscription, using the configuration as described in the lab guide, the estimated Azure consumption costs for running the setups during the 2 days of the workshop is $20.

Other requirements:

Participants need a local client machine, running a recent Operating System, allowing them to:

- browse to https://portal.azure.com from a most-recent browser;
- establish a secured Remote Desktop (RDP) session to a lab-jumpVM running Windows Server 2016;

Alternative Approach:

Where the lab scenario assumes all exercises will be performed from within the lab-jumpVM, (since several tools will be installed on the lab-jumpVM or are already installed by default), participants could also execute (most, if not all…) steps from their local client machine.

The following tools are being used throughout the lab exercises:

- Visual Studio 2017 community edition (updated to latest version); this could also be Visual Studio 2019 community edition - latest version
- Docker for Windows (updated to latest version)
- Azure CLI 2.0 (updated to latest version)
- Kubernetes CLI (updated to latest version)
- SimplCommerce Open Source e-commerce platform example (http://www.simplcommerce.com)

Make sure you have these tools installed prior to the workshop, if you are not using the lab-jumpVM. You should also have full administrator rights on your machine to execute certain steps within using these tools.

Final Remarks:

VERY IMPORTANT: You should be typing all of the commands as they appear in the guide, except where explicitly stated in this document. Do not try to copy and paste from Word to your command windows or other documents where you are instructed to enter information shown in this document. There can be issues with Copy and Paste from Word or PDF that result in errors, execution of instructions, or creation of file content.

IMPORTANT: Most Azure resources require unique names. Throughout these steps you will see the word "[SUFFIX]" as part of resource names. You should replace this with your initials, guaranteeing those resources get uniquely named.

Lab 2: Performing Assessment of your as-is situation.

What you will learn

In this second lab, you focus on performing the necessary assessment phase in your simulated "on-premises" application landscape, by using Microsoft assessment tools:

- Microsoft Database Migration Assistant;
- Azure App Service Migration Assistant;

Time estimate

This lab is estimated to take **30min**, assuming your Azure subscription is already available, and you successfully completed Lab 1.

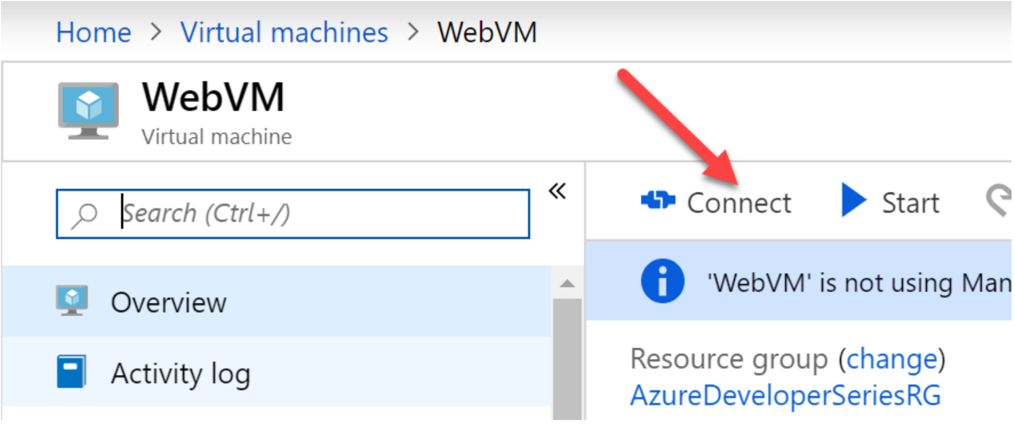
Task 1: Running a SQL Server assessment using Data Migration Assistant

In short, you will perform the following tasks:

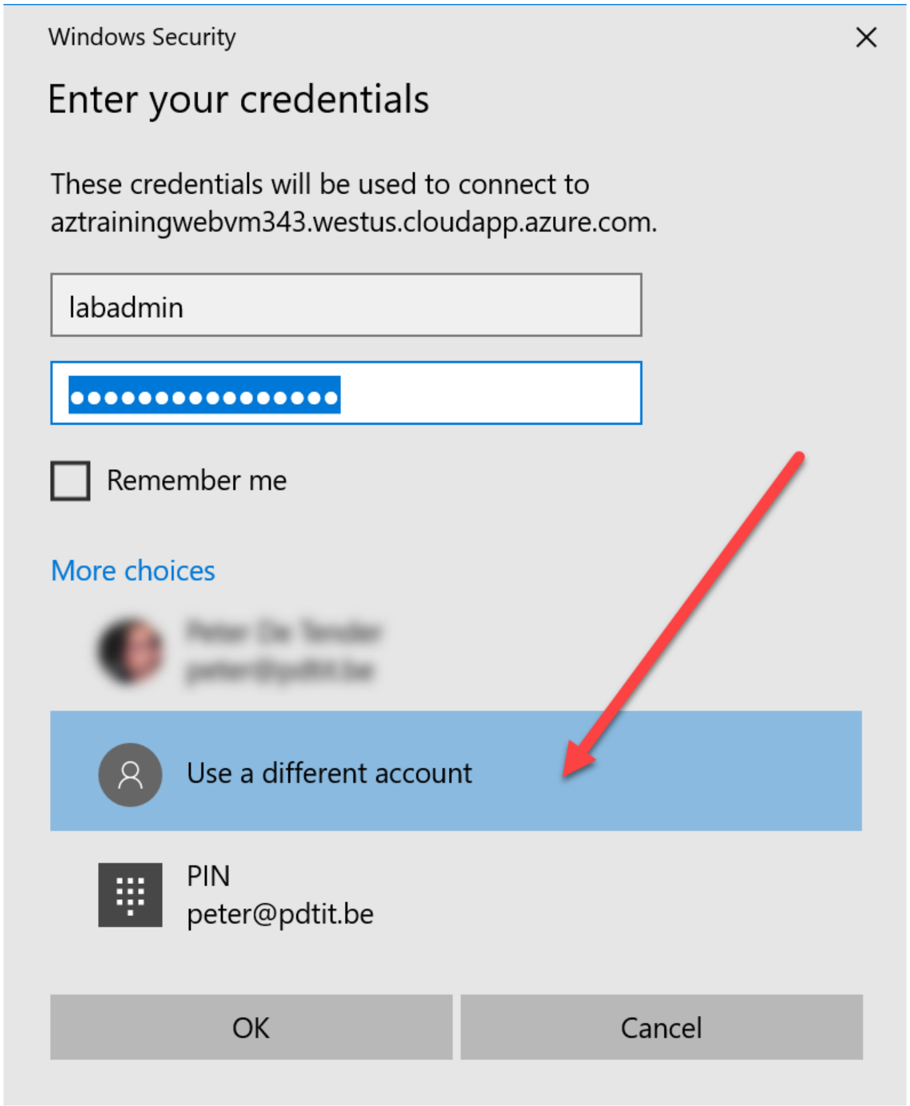
1. Install the Azure Database Migration Assistant on the Web VM
2. Perform an assessment of the to-be-migrated database

In this task, you download and install the Azure Database Migration Assistant. We are using the SQLVM in this lab, but you can run this from any Windows server in the same network as the SQL Virtual Machine, meaning, you don't have to install it on the Web Server VM itself.

1. Connect to the WebVM virtual machine using RDP, by **selecting** the WebVM from the **Virtual Machines** section in the Azure Portal followed by **selecting Connect**.



2. In the **Connect** blade, **click** the **Download RDP File.** Once downloaded, open the file. This will start the **Remote Desktop**, asking for credentials. Here, **select "Use a different account"** and provide the following credentials:
user account: **labadmin**
password: L@BadminPa55w.rd



Windows Security

Enter your credentials

These credentials will be used to connect to aztrainingwebvm343.westus.cloudapp.azure.com.

labadmin

●●●●●●●●●●●●●●●●●

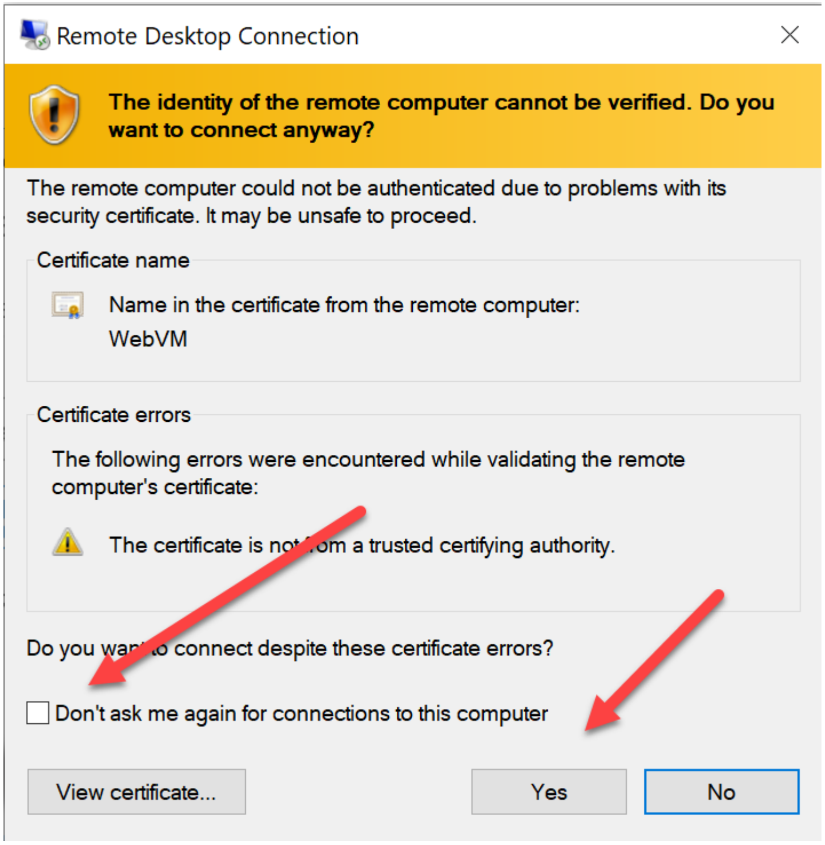
□ Remember me

More choices

Use a different account

PIN
peter@pdtit.be

3. When you are prompted for a certificate security warning, select Don't ask me again… and press OK to continue.



Remote Desktop Connection

The identity of the remote computer cannot be verified. Do you want to connect anyway?

The remote computer could not be authenticated due to problems with its security certificate. It may be unsafe to proceed.

Certificate name

Name in the certificate from the remote computer:
WebVM

Certificate errors

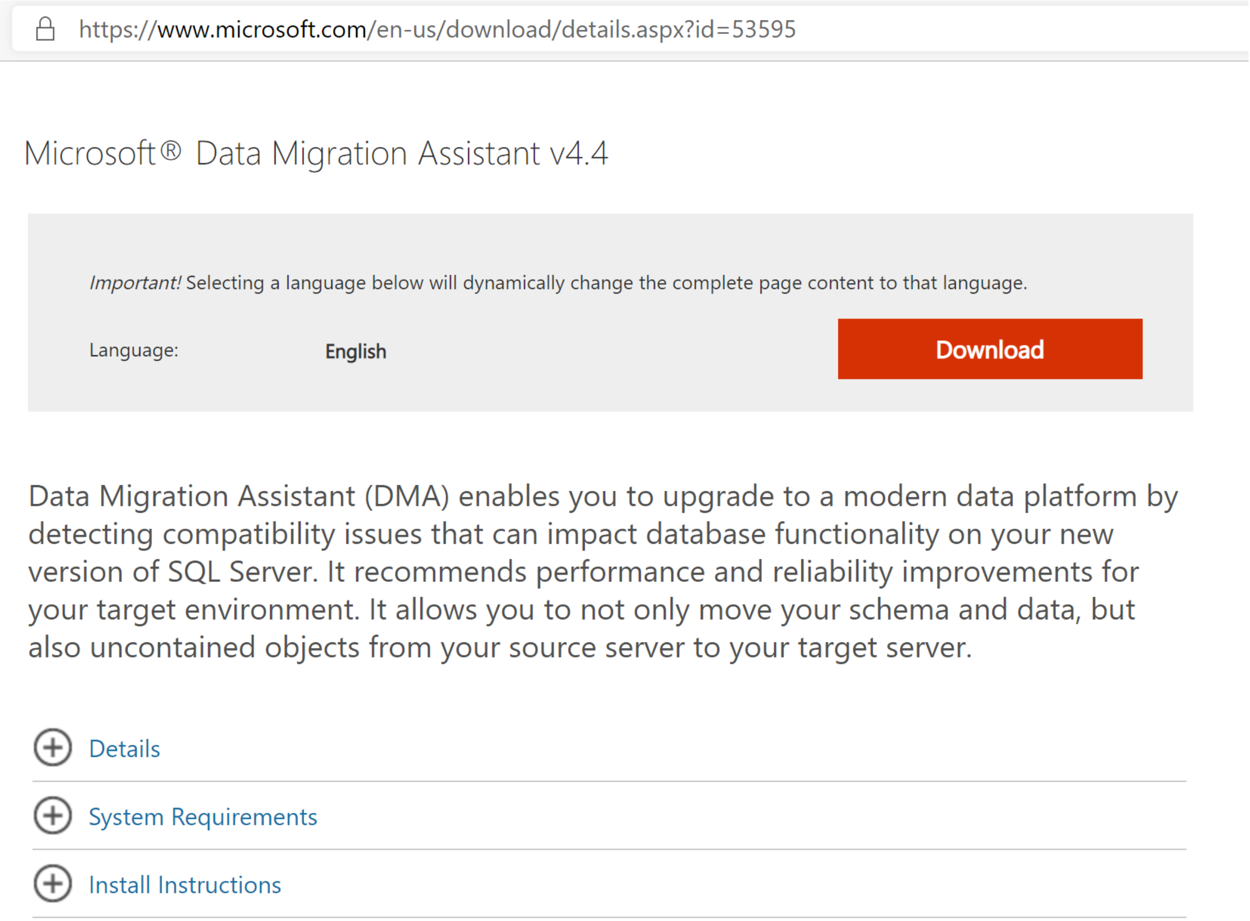
The following errors were encountered while validating the remote computer's certificate:

The certificate is not from a trusted certifying authority.

Do you want to connect despite these certificate errors?

□ Don't ask me again for connections to this computer

4. Once logged on to the desktop of the WebVM, open the browser, and search for Azure Database Migration Assistant download, or connect directly to the following URL: https://www.microsoft.com/en-us/download/details.aspx?id=53595



https://www.microsoft.com/en-us/download/details.aspx?id=53595

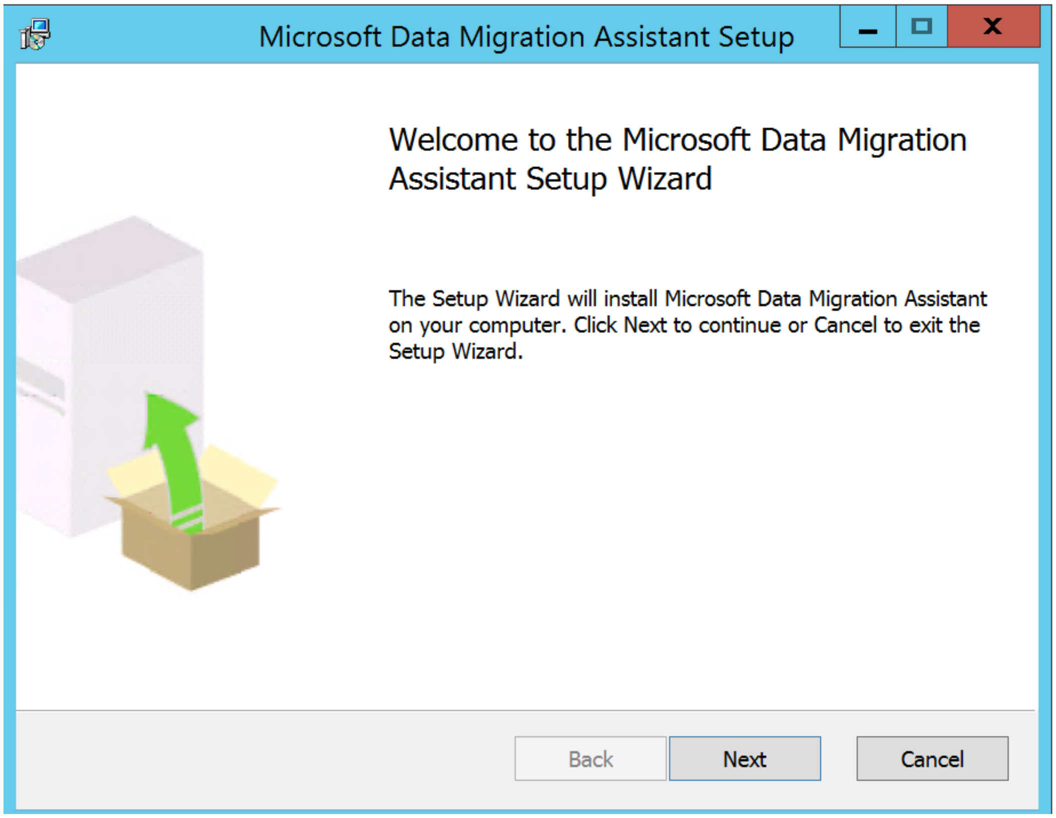
Microsoft® Data Migration Assistant v4.4

Important! Selecting a language below will dynamically change the complete page content to that language.

Language: English

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. It recommends performance and reliability improvements for your target environment. It allows you to not only move your schema and data, but also uncontained objects from your source server to your target server.

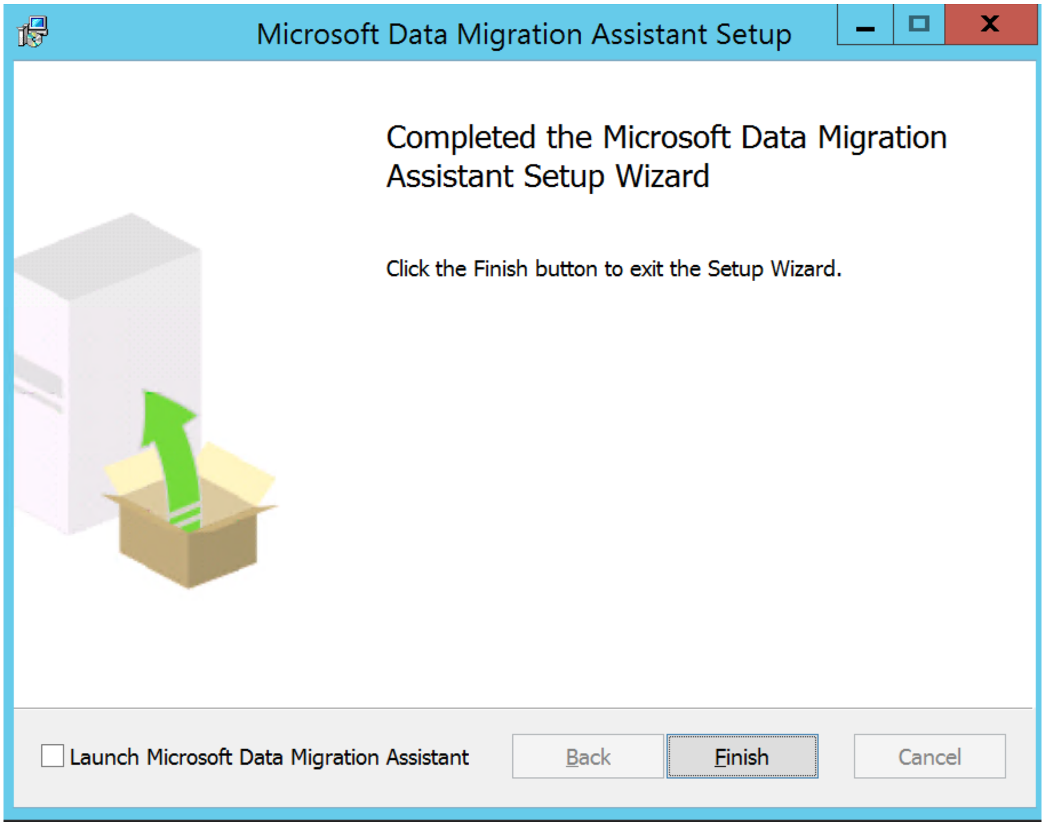
5. Once the download is complete, launch the DataMigrationAssistant.msi. Press Next to continue.



Welcome to the Microsoft Data Migration Assistant Setup Wizard

The Setup Wizard will install Microsoft Data Migration Assistant on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

6. Accept the license terms agreement, press Next and confirm by pressing the Install button. Wait for the install to complete successfully.



Microsoft Data Migration Assistant Setup

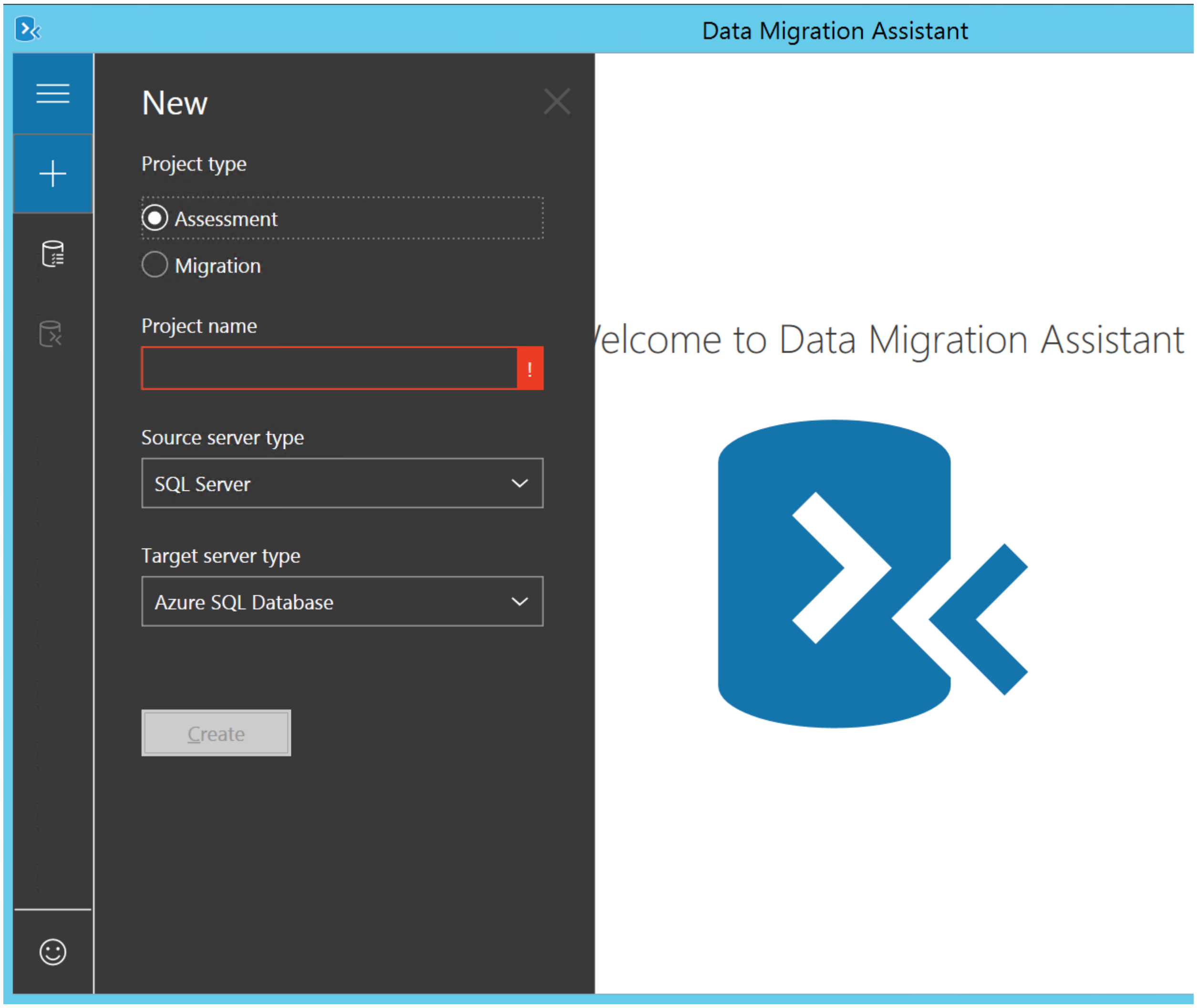
Completed the Microsoft Data Migration Assistant Setup Wizard

Click the Finish button to exit the Setup Wizard.

□ Launch Microsoft Data Migration Assistant

7. If you want to open the DMA tool, select “Launch Microsoft Data Migration Assistant”.

8. From Data Migration Assistant, select the + on the side to launch a new migration scenario.



New

Project type

● Assessment

○ Migration

Project name

Source server type

SQL Server

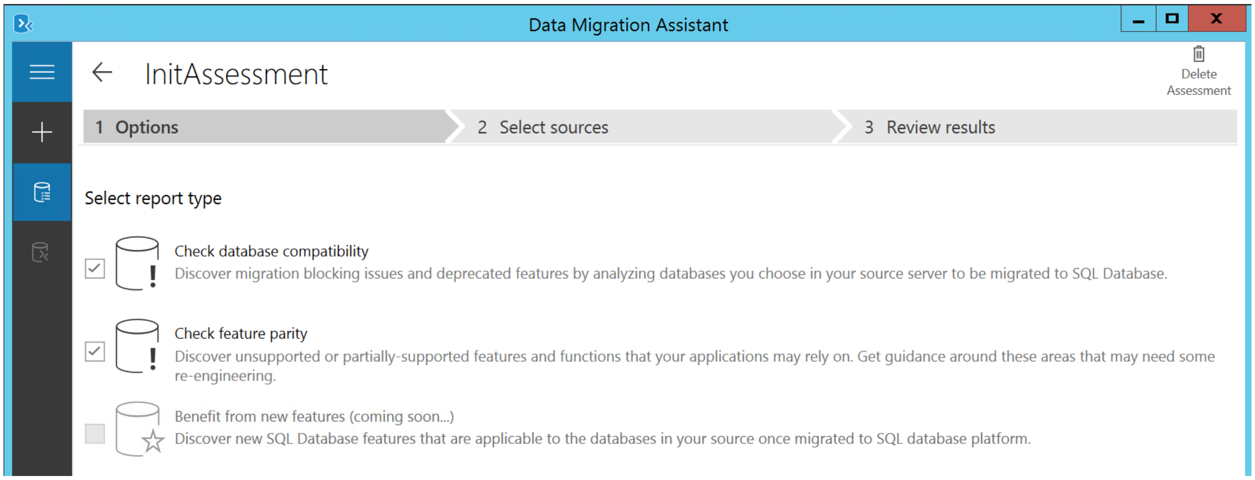
Target server type

Azure SQL Database

9. We start by running an assessment. Complete the wizard with the following parameters:

- Project type	Assessment
- Project Name	InitAssessment
- Source Server type	SQL Server
- Target Server type	Azure SQL Database

10. This launches the Data Migration Assistant selection window. Here, press Next to continue.



Select report type

☑ Check database compatibility
Discover migration blocking issues and deprecated features by analyzing databases you choose in your source server to be migrated to SQL Database.

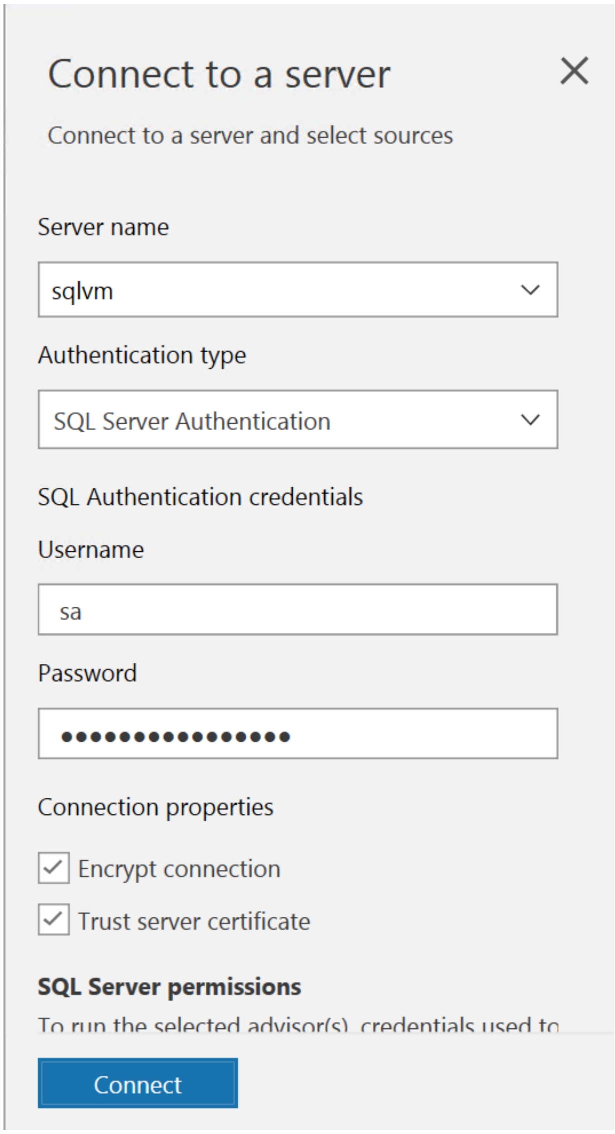
☑ Check feature parity
Discover unsupported or partially-supported features and functions that your applications may rely on. Get guidance around these areas that may need some re-engineering.

□ Benefit from new features (coming soon...)
Discover new SQL Database features that are applicable to the databases in your source once migrated to SQL database platform.

11. We now need to connect to our source SQL Server. Therefore, provide the following information in the wizard:

- Server Name	sqlvm
- Authentication type	SQL Server authentication
- Username	sa
- Password	L@BadminPa55w.rd

Also flag both options “Encrypt connection” and “Trust certificate”.



Server name: sqlvm

Authentication type: SQL Server Authentication

SQL Authentication credentials

Username: sa

Password: ●●●●●●●●●●●●●●●●

Connection properties

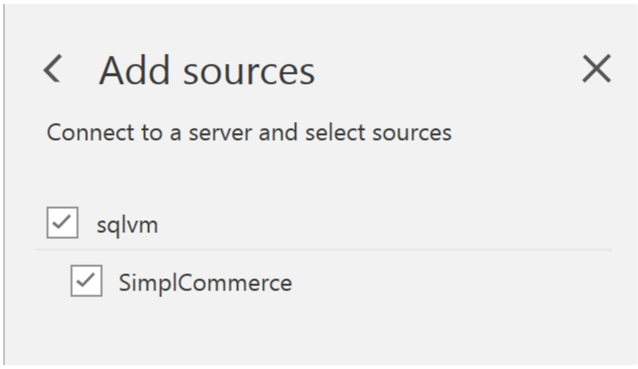
☑ Encrypt connection

☑ Trust server certificate

SQL Server permissions

To run the selected advisor(s), credentials used to

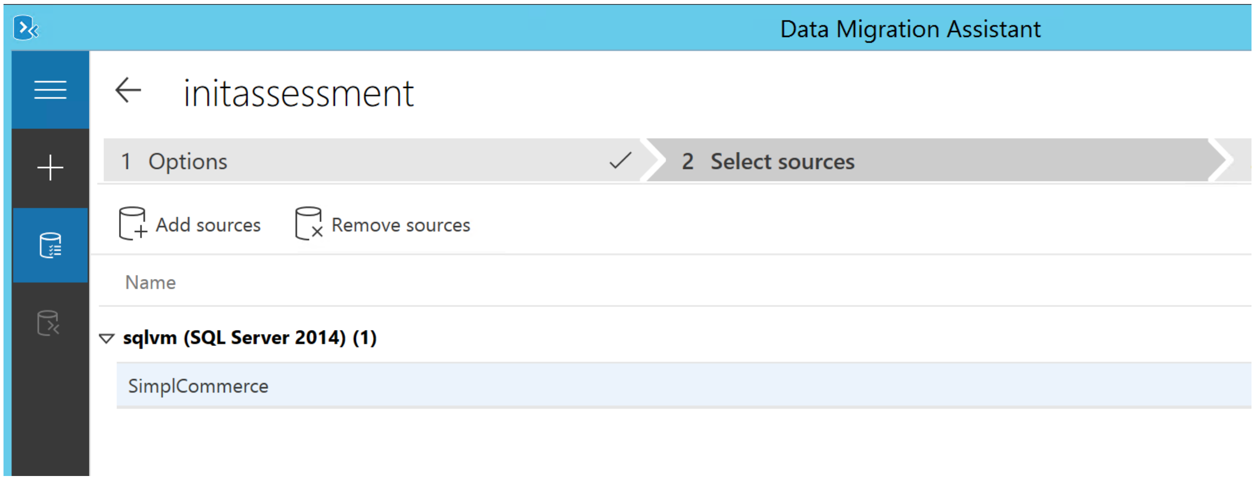
12. Press Connect to continue. This brings up the sources list.



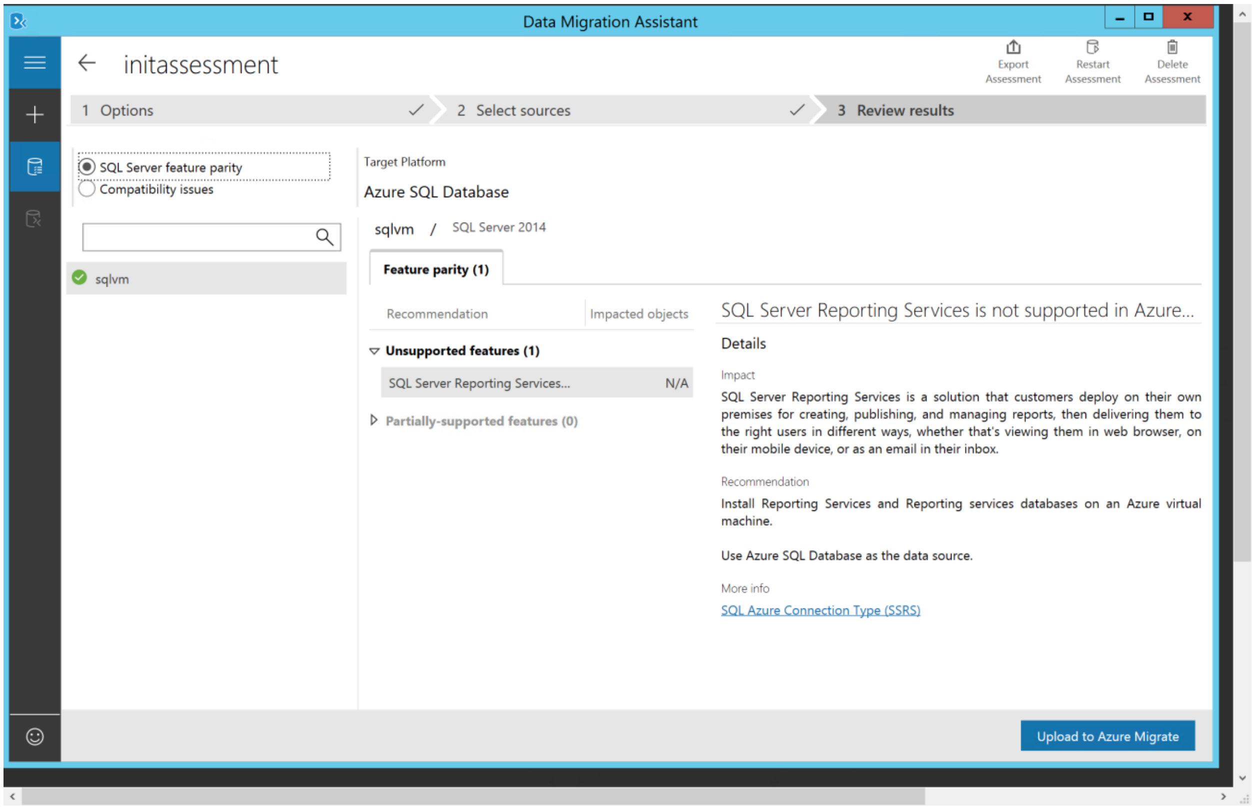
☑ sqlvm

☑ SimplCommerce

13. **Select** SimplCommerce as source database, and **select Add**, to add this database to the list



14. **Next, press "Start Assessment"** button. This runs the assessment and should take a few minutes to complete. Take note of the several recommendations under **unsupported features** and **partially supported features**.



SQL Server Reporting Services is not supported in Azure...

Details

Impact

SQL Server Reporting Services is a solution that customers deploy on their own premises for creating, publishing, and managing reports, then delivering them to the right users in different ways, whether that's viewing them in web browser, on their mobile device, or as an email in their inbox.

Recommendation

Install Reporting Services and Reporting services databases on an Azure virtual machine.

Use Azure SQL Database as the data source.

More info

SQL Azure Connection Type (SSRS)

15. **Once** you are familiar with the reported features, you can close the Data Migration Assistant.

This completes the task in which you deployed and ran the Data Migration Assistant to validate compatibility of your source SQL Server database with Azure SQL target.

In a next lab, you will reuse this tool to perform the actual database migration.

Task 2: Running a Web Server assessment using Azure App Service Migration Assistant

In short, you will perform the following tasks:

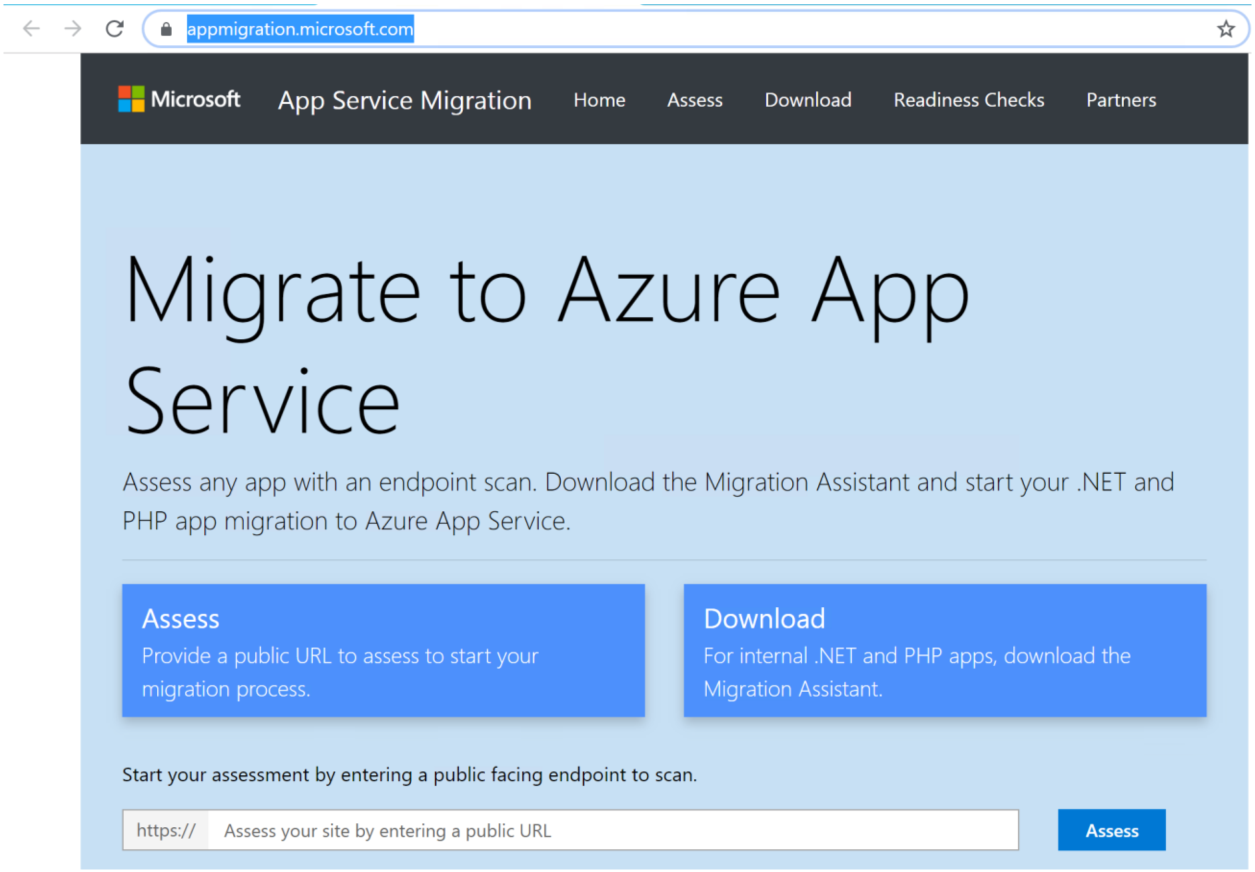
1. Install the Azure App Service Migration Assistant on the Web VM
2. Perform an assessment of the to-be-migrated web application

In this task, you download and install the Azure App Service Migration Assistant. We are using the WebVM directly in this lab, but you can run this from any Windows server in the same network as the WebVM Virtual Machine, meaning, you don't have to install it on the Web Server VM itself.

1. Connect to the WebVM virtual machine using RDP, by **selecting** the WebVM from the **Virtual Machines** section in the Azure Portal followed by **selecting Connect,** and authenticating with labadmin / L@BadminPa55w.rd as credentials.

2. From within the **WebVM**, open an internet browser, and connect to the following URL to download the latest version of the Azure App Service Migration Assistant:

   https://appmigration.microsoft.com/



Migrate to Azure App Service

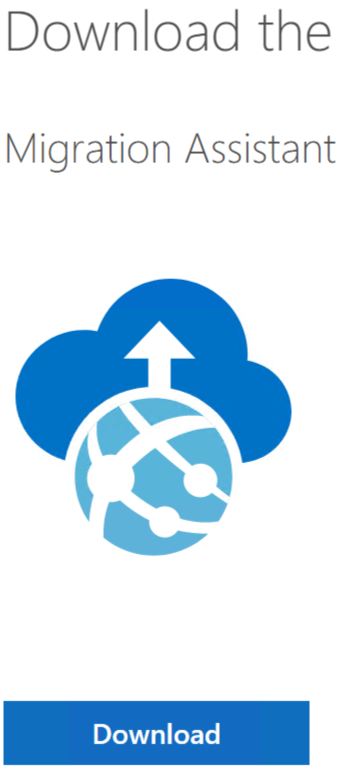
Assess any app with an endpoint scan. Download the Migration Assistant and start your .NET and PHP app migration to Azure App Service.

Assess
Provide a public URL to assess to start your migration process.

Download
For internal .NET and PHP apps, download the Migration Assistant.

Start your assessment by entering a public facing endpoint to scan.

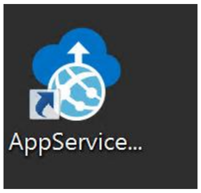
3. **Press** the **Download option,** to get redirect to the download page. Here, continue with **selecting** the download button.



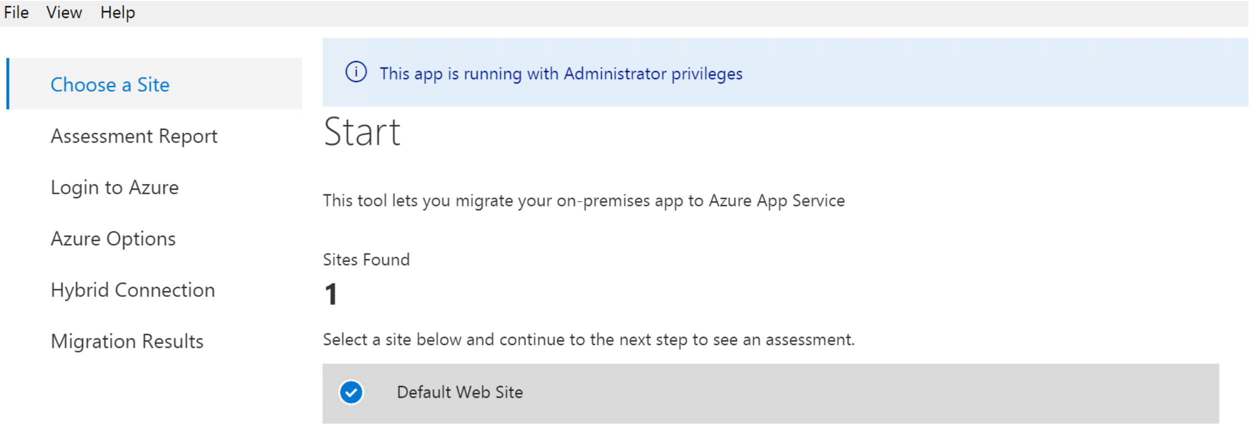
Download the Migration Assistant

Migration Assistant

4. **Once downloaded**, launch the **AppServiceMigrationAssistant.msi**, which will configure a shortcut on the desktop



5. **Launch** the AppServiceMigrationAssistant; This brings up a 5-step scenario. **Select Step 1 “Choose a site”**; here, notice it has found 1 site, “Default Web Site”.



This app is running with Administrator privileges

Start

This tool lets you migrate your on-premises app to Azure App Service

Sites Found

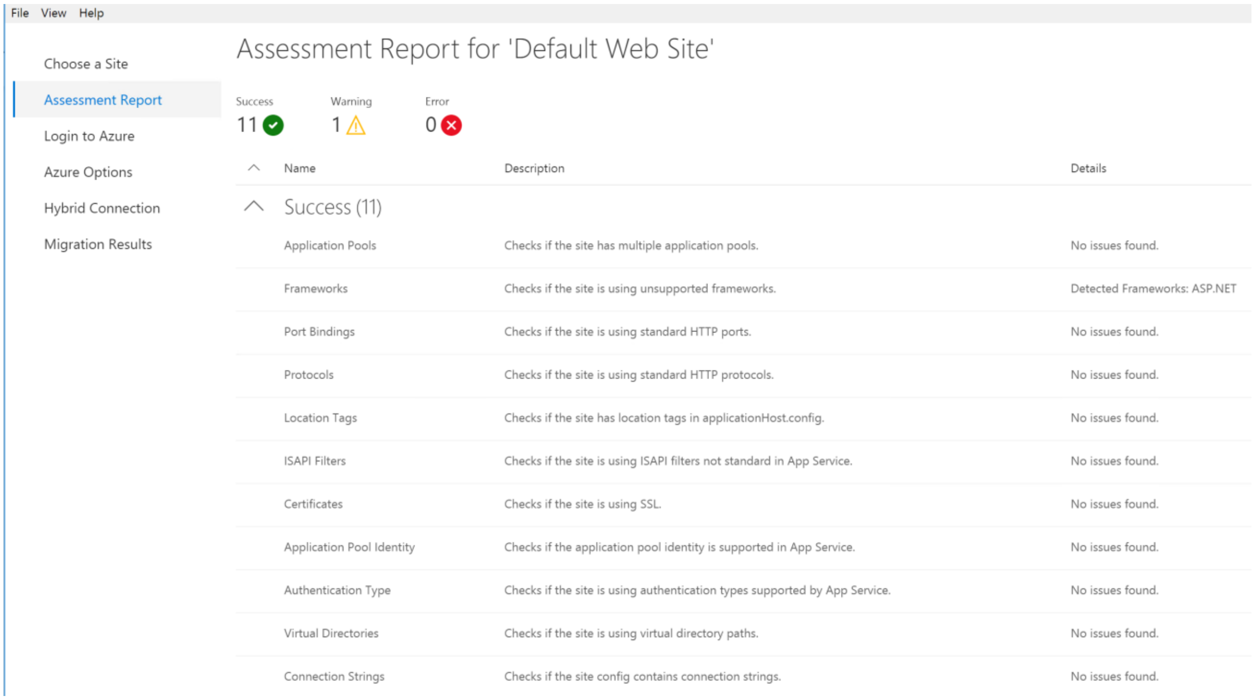
1

Select a site below and continue to the next step to see an assessment.

✓ Default Web Site

6. Select "Default Web Site" and press Next to continue

7. This results in a detailed assessment report of the web application. Browse through this report to become familiar with the gathered information.



Assessment Report for 'Default Web Site'

Success 11 ✓ Warning 1 ⚠ Error 0 ✕

Name	Description	Details
Success (11)		
Application Pools	Checks if the site has multiple application pools.	No issues found.
Frameworks	Checks if the site is using unsupported frameworks.	Detected Frameworks: ASP.NET
Port Bindings	Checks if the site is using standard HTTP ports.	No issues found.
Protocols	Checks if the site is using standard HTTP protocols.	No issues found.
Location Tags	Checks if the site has location tags in applicationHost.config.	No issues found.
ISAPI Filters	Checks if the site is using ISAPI filters not standard in App Service.	No issues found.
Certificates	Checks if the site is using SSL.	No issues found.
Application Pool Identity	Checks if the application pool identity is supported in App Service.	No issues found.
Authentication Type	Checks if the site is using authentication types supported by App Service.	No issues found.
Virtual Directories	Checks if the site is using virtual directory paths.	No issues found.
Connection Strings	Checks if the site config contains connection strings.	No issues found.

8. Notice we have no errors, but 1 warning, related to a connection string of our e-commerce application, where the tool wants to emphasize we need to update our application connection string before/after the migration.

9. This completes the task

Summary

In this lab, you deployed the Database Migration Assistant as well as the App Service Migration Assistant, to validate your existing e-commerce application environment, to be compatible with Azure Platform As A Service, as part of the assessment phase of your migration project.

In the next labs, you will reuse these tools to perform an actual migration of the SQL Server database to Azure SQL, as well as migrating the Web Application to Azure Web Apps.