

Deployment - Power BI Starter Kit

June 1, 2020

- [Solution Overview](#)
- [Deployment](#)
- [Technical Details](#)
- [Early Warning Metrics](#)

Table of Contents

- [Azure Deployment](#)
 - [Analysis Services](#)
 - [Automation](#)
 - [Automation Account](#)
 - [Credentials](#)
 - [Modules](#)
 - [Runbooks](#)
 - [Schedule](#)
 - [Office 365 Group Creation](#)
 - [Analytics Middle Tier Deployment](#)
 - [Tabular Model Deployment](#)
 - [Power BI Deployment](#)
- [Post Deployment Actions](#)

Azure Deployment

There are two main components in Azure - Analysis Services and Automation. Analysis Services is required for this solution, while Automation provides some useful cost-saving features and is completely optional.

- Analysis Services is the platform-as-a-service offering of SQL Server Analysis Services. This holds the definition of the data model and metric measurements and calculates the results for the Power BI report visualizations.
- Optional: Automation is a scripting framework to automate processes in Microsoft Azure. This is used to enable, disable, and process the Analysis Services model on a schedule.

Analysis Services

1. Log in to the [Microsoft Azure Portal](#).
2. Select the "+ Create a resource" button at the top of the left-navigation menu.
3. Search for "Analysis Services". The first result will be published by Microsoft.
4. Select the "Analysis Services" option and click Create.
5. Enter the name for the Analysis Services (AS) server. This will be the unique identifier for your server and will be used in the connection string.
6. Select the subscription that the server will be deployed to.
7. Choose an existing or new resource group to deploy the server to. A resource group is a collection of Azure services.

8. Select the location where the Analysis Services server will be hosted.
9. Select the tier for which the server will run on. For more information regarding which tier to choose, read the section of this documentation entitled 'Pricing Tiers'.
10. Select an Administrator from the Azure Active Directory - this can be an individual or group. This can be changed at any time, and for the purposes of successful deployment, leave it to the default value (your username).
11. Optional: You have the option to back up the server to an Azure Storage Account. By default, this is not enabled, and can be enabled anytime in the future.
12. Click "Create" to initiate the deployment of the Analysis Services server.
13. Once the deployment has finished, you can view the server in the list of resources in the Azure Portal.

Automation

Automation Account

1. Log in to the [Microsoft Azure Portal](#).
2. Select the "+ Create a resource" button at the top of the left-navigation menu.
3. Search for "Automation". The first result will be published by Microsoft.
4. Select the "Automation" option and click Create.
5. Enter the name for the Automation account. The name should represent a collection of like-minded tasks, such as "ed-fi-analytics-automation".
6. Select the subscription where the account will be deployed.
7. Choose an existing or new resource group. A resource group is a collection of Azure services.
8. Select the location where the Automation account will be hosted.
9. Click "Create" to initiate the creation of the Automation account.
10. Leave the default value of "Yes" for "Create Azure Run As account."
11. Click "Create" to initiate the deployment of the Automation module
12. Wait for the resource to be created before proceeding to the next steps.

Credentials

A set of credentials must be stored within the Azure Automation account in order to authenticate against both Azure and the Analysis Services server.

1. From the "All Resources" blade, navigate to the Azure Automation account you created.
2. Select "Credentials" under the "Shared Resources" category on the left-navigation menu within the resource panel.
3. Select "+ Add a credential" at the top of the panel.
4. Enter the following name for the credential: EdFiEWSAccount
 1. You can use a different name if you wish, but you'll need to adjust the runbook scripts accordingly.
5. Enter a username and password for an account within the Azure Active Directory.

(!) Note: This account must be an administrator on the Azure Analysis Services server and have access to the Azure Portal resource. For the ease of deployment, use the same account for all steps. For more information on Analysis Services administrator accounts, see [Tutorial: Configure server administrator and user roles](#).

6. Click "Create" to add the credential to the Automation account.
7. Proceed to the Modules section.

Modules

Azure Automation is backed by PowerShell, and this solution requires certain PowerShell modules to be configured on the Automation account.

1. Navigate to the Azure Automation account you created.
2. Select "Modules" under the "Shared Resources" category on the left-navigation menu within the resource panel.
3. The "Update Azure Modules" option has been deprecated. Use the following steps to ensure all the modules are up-to-date
 1. Click the "Clone or download" button on the [Azure Automation Modules](#) repository.
 2. Choose to download a zip file of Azure Automation Update module
 3. uncompress the .zip file to a local folder
 4. Click "+ Add a module" in the "Modules" section of your Azure Automation Account
 5. In the "Upload File" box navigate to the downloaded zip file
 6. Click "OK"
 7. Go to the "Runbooks" section from the left-menu and click on "import runbook"
 8. in the file chooser menu select the "Update-AutomationAzureModulesForAccount" file in the uncompressed version of your download, the other fields will be automatically populated.
 9. Click "ok"
 10. Click "publish" on the new runbook
 11. Click the "start" button on the new runbook
 12. Use the following parameters for the runbook and click "OK"

ResourceGroupName => Enter whichever Resource group you set your Automation account to use.

AutomationAccountName => Enter the name of the Automation Account.

The remaining fields should use the pre-selected "Default will be used" option.

13. This will ensure that we have the latest AzureRM modules installed in our Automation account.
14. Click "Refresh" to check on the status of the updates.
15. Once all the modules are updated, proceed with the next steps.
4. Select "Browse Gallery" in the Automation Account's "Module" section.
5. Search for "AzureRM.AnalysisServices". Select the result created by "azure-sdk".
6. Select "Import" at the top of the panel, and then click "Ok".
7. Select "Browse Gallery" again, and search for the "SqlServer" module. Select the result created by "matteot_msft".
8. Select "Import" at the top of the panel, and then click "Ok".
9. Proceed to the Runbooks section below.

Runbooks

A runbook is a script that is executed on-demand or based on a schedule. One or more runbooks associated with each Automation account.

1. Navigate to the Azure Automation account you created.
2. Select "Runbooks" under the "Process Automation" category on the left-navigation menu within the resource panel.
3. Select "+ Import a runbook" at the top of the panel.
4. Select the runbook file. There are two runbooks that should be imported:

Runbook Name	Description
AnalysisServicesSchedulerRunbook.ps1	Used to pause and resume the Analysis Services server on a schedule.
AnalysisServicesRefreshRunbook.ps1	Used to re-process the Analysis Services Tabular Model on a schedule. This will pull in the latest information from the ODS.

5. Select "PowerShell" for the "Runbook type".
6. Click "Create" to import the PowerShell script.
7. Click "Publish" on the top menu after the run book has been created.
8. Repeat steps 3 through 7 for each .ps1 file.
9. Once all Runbooks have been imported, we need to create a schedule for each one.
10. Open the [AnalysisServicesSchedulerRunbook](#) runbook.
11. Click "Edit" on the top of the panel. This will bring you into an editor for the PowerShell script.
12. Click "Publish" on the top of the panel, then "Yes" to verify the action. This will save and publish the runbook, and allow you to set a schedule for it.
13. Repeat steps 10 through 12 for each runbook.
14. Once both runbooks are published, proceed to the Schedule section below.

Schedule

A schedule is a time-based trigger to run an associated runbook. Additionally, you have the option to pass schedule-specific parameters to the runbook.

1. Navigate to the Azure Automation account you created.
2. Select "Runbooks" under the "Process Automation" category on the left-navigation menu within the resource panel.
3. Select the AnalysisServicesRefreshRunbook.
4. Select "Schedules" under the "Resources" category on the left-navigation menu.
5. Select "+ Add a schedule" at the top of the panel.

Note: If the "+ Add a schedule" button is greyed out, the runbook is not published. Follow steps 10 through 12 in the Runbooks section.

6. Select the "Schedule Link a schedule to your runbook" option, and then "+ Create a new schedule". For the purposes of this solution, three schedules are recommended:

Schedule Name	Description	Time	Associated Runbook
RefreshAnalysisServices	Refreshes the Analysis Services Tabular Model	7:15am	AnalysisServicesRefreshRunbook
ResumeAnalysisServices	Resumes the Analysis Services server	7:00am	AnalysisServicesSchedulerRunbook

Schedule Name	Description	Time	Associated Runbook
SuspendAnalysisServices	Pauses the Analysis Services server	7:00pm	AnalysisServicesSchedulerRunbook

- Set the schedule for the associated action. Make sure to enable "Recurrence" and set the interval at which this action will run (e.g. daily).
- Click "Create" to save the schedule.
- Select the "Parameters and run settings" option. Each runbook will have a specific set of parameters.

Runbook Name	Parameters
AnalysisServicesRefreshRunbook	<ul style="list-style-type: none"> servername - This is the name for the Azure Analysis Services server. databasename - This is the name of the Tabular Model. By default, it is "Ed-Fi-Data-Analytics". resourcegrouplocation - This is the resource group location for the Analysis Services server. It must be formatted in all lowercase with no spaces (e.g. South Central US becomes southcentralus).
AnalysisServicesSchedulerRunbook	<ul style="list-style-type: none"> action - This is either "resume" or "suspend", depending on the associated time. This will pause or resume the Analysis Service server. resourcegroupname - This is the name of the Azure Resource Group where the Analysis Services server is located. servername - This is the name of the Azure Analysis Services server.

- Click "Ok" twice to create and associate the schedule with the runbook.
- Repeat this process for each schedule.
- This completes the Azure Automation deployment. You can monitor the success or failure of each runbook in the Azure Portal.

Office 365 Group Creation

- Log in to the [Office 365 Admin Portal](#).
- Select the "Groups" tab in the left-navigation menu and select Groups.
- Repeat steps 4 through 11 for the two groups below: | Group Name | Group Id /

Group email address

Ed-Fi EWS Read All

edfiewsstudentauth

- Click "+ Add a Group" at the top of the page. A new panel will appear.
- Make sure the "Type" chosen is "Office 365 group".
- Enter the Name of the group.
- Enter the Id/email address of the group. This will be the alias of the group.
- Make sure "Privacy" is set to Public. This is required for Analysis Services row-level security.

9. Select an owner for the group. Select your username - this can be changed at a later date, and you will need access to the group in subsequent steps.
10. Disable the toggle for "Send copies of group conversations and events to group members' inboxes".
11. Click "Add" to create the group. It may take a minute or two to populate in the Office 365 Portal.

Analytics Middle Tier Deployment

The Analytics Middle Tier is a collection of views that simplify the process of retrieving data from the ODS for use in ad hoc or packaged analytics solutions. A basic installation might target the production ODS. For better overall performance, consider creating a replicated copy of the ODS and installing the Analytics Middle Tier into that copy. See Patterns and Practices for more information on this approach.

1. On a computer that can access the ODS database, download the latest release of the [Analytics Middle Tier](#) (must be signed-in on Github.com before clicking the link). The larger zip file, `EdFi.AnalyticsMiddleTier-win10.x64.zip`, contains all dependencies needed to run the application. The smaller file, `EdFi.AnalyticsMiddleTier.zip`, requires the .NET Core 2.1 runtime to be separately installed.
 1. If using a local computer, make sure that computer's IP address is allowed to access the SQL Server instance behind its firewall.
 2. Another option is to use Remote Desktop to connect to the database server (if on a Virtual Machine) and download and execute the program there.
2. Unzip the release.
3. Open a command prompt and CD to the directory containing the unzipped contents.
4. Assuming use of self-sufficient win10.x64 zip file, run the following command, substituting in a correct connection string for your ODS database:

```
.\EdFi.AnalyticsMiddleTier.Console.exe --connectionString "..."
```

5. To Install the row level security you will need to run:

```
.\EdFi.AnalyticsMiddleTier.Console.exe -c "... " -o RLS`
```

6. To Install the Early Warning System you will need to run:

```
.\EdFi.AnalyticsMiddleTier.Console.exe -c "... " -o EWS
```

Tabular Model Deployment

1. Open SQL Server Management Studio.
2. Navigate to File - Open and select the Model.xml file in the Ed-Fi Early Warning System directory.
3. You will be prompted to connect to an Analysis Services server. Enter the connection string for your Analysis Services instance. You can find this on the resource panel in the Azure Portal, and should be formatted like: `asazure://resourcegrouplocation.asazure.windows.net/servername`
4. Click "Connect". You'll be prompted to log in with your Azure AD account.

5. Now, we'll need to modify the connection to the ODS data source in the script before executing.
 1. Search the document for `address`. Replace `{Azure SQL Server name}` and `{Azure SQL database name}` with the name of your Azure SQL server running your Ed-Fi ODS database and the name of that database.
 2. Search the document for `credential`. Again replace `{Azure SQL Server name}` and `{Azure SQL database name}` with the name of your Azure SQL server running your Ed-Fi ODS database and the name of that database. Also replace `{SQL username}` and `{SQL password}` with the username and password for the SQL admin credentials on the Ed-Fi ODS database:
6. Search the document for `@domain.com`. There will be two matches toward the bottom of the document: one for `edfiwreadsall` and one for `edfiwstudentauth`.
7. Replace each of these instances with the correct domain for your Office 365 account. Each of these will correlate to an Office 365 Group that was created earlier.
8. Select "Execute" at the top of SSMS or press F5 to execute the script. The tabular model will be deployed to the Azure Analysis Services server.

Note: The Analysis Services model will not be populated with data until you complete the processing step in the Post Deployment Actions section.

Power BI Deployment

1. Open the Power BI folder in the supplied artifacts.
2. Open one of the .PBIX files in Power BI Desktop.
3. You will be prompted to log in to your Azure account.
4. You will receive an error message stating that your account does not have access to the server. This is correct - it is trying to authenticate your account to an invalid server (the one used for initial development of this model).
5. Select 'Edit' on the warning message and enter your Analysis Services connection string. Leave the Database Name alone - this is hardcoded in the Tabular Model.
 1. If you did not write this down after deploying the Azure components, you can view it in the Azure Portal. It will look something like as `asazure://{region}.asazure.windows.net/{ssas-name}`
 2. In the future you can return to the connection settings from the Home ribbon, under Edit Queries > Data Source Settings
6. Ensure the tabular data model name is set to "Ed-Fi-Data-Analytics" or whatever you listed above.
7. When prompted, select "Model". This is selecting the Tabular Model within the Analysis Services server.
8. Once connected, select "Publish" in the top ribbon menu. You may be prompted to log in to your Power BI account.
9. Select the workspace that you'd like to publish these reports to. This should match the context for each report.
 1. District and School PBIX should be published to the District Admins and School Admins workspaces.
 2. Teacher PBIX should be published to the Teachers workspace.
10. The reports should now be visible on the Power BI Web Service.
11. Before the reports can be completely used, you must complete the Post-Deployment Actions below.

Note: In order to view the reports on PowerBI.com, you must change the workspace from 'My Workspace' to one of the security groups.

Post Deployment Actions

Once all the Azure and Power BI components are fully deployed, the Tabular Model will need to be processed. This step is required; it will pull information into the Azure Analysis Services server from the ODS and allow the individuals to view the Power BI reports.

1. Process the Tabular Model database manually.
 1. Load SSMS and navigate to Connect - Analysis Services.
 2. Enter the Azure Analysis Services connection string. You'll be prompted to log in with your Azure AD account.
 3. Right-click the database and select "Process Database". For the first time, you'll want to use Process Full. This can take some time depending on your Analysis Services tier.
 1. If you find that you have a credential problem when connecting to the SQL Server, in SSMS, expand Ed-Fi > Connections. Double-click on Enterprise ODS. In the next dialog box, click in the Credentials property to edit the credentials.
 2. If you get an error referring to a data gateway, you can find install instructions at [Microsoft's Gateway Install Documentation](#)
2. Verify that your Power BI Workspaces are set to read-only. This ensures that your end users cannot modify the reports for other individuals.
 1. Log in to the Power BI Web Service.
 2. Open up the Workspace selection menu.
 3. Click on the three-dot menu next to each workspace. Select "Edit workspace".
 4. Change the drop-down menu under the Privacy subsection to "Members can only view Power BI content".
 5. Click Save to apply the new settings.
3. Add users to the appropriate Office 365 Groups. This can be done in the Office 365 Portal.
 1. Users added to **Ed-Fi EWS Student Auth** can only view the data for schools and sections associated with their user account in the **StudentDataAuthorization** table.
 2. Users added to **Ed-Fi EWS Read All** can view all data. Recommended only for use in testing, not production.