bi4dynamics
Transforming your data into knowledge

BI4Dynamics Process Automation

How to automatically update data from BC Cloud, Data Lake to Analysis Services

BI server is On-Premises

Last update: August 2023
Version 2.0
Revision 1.2

Contents

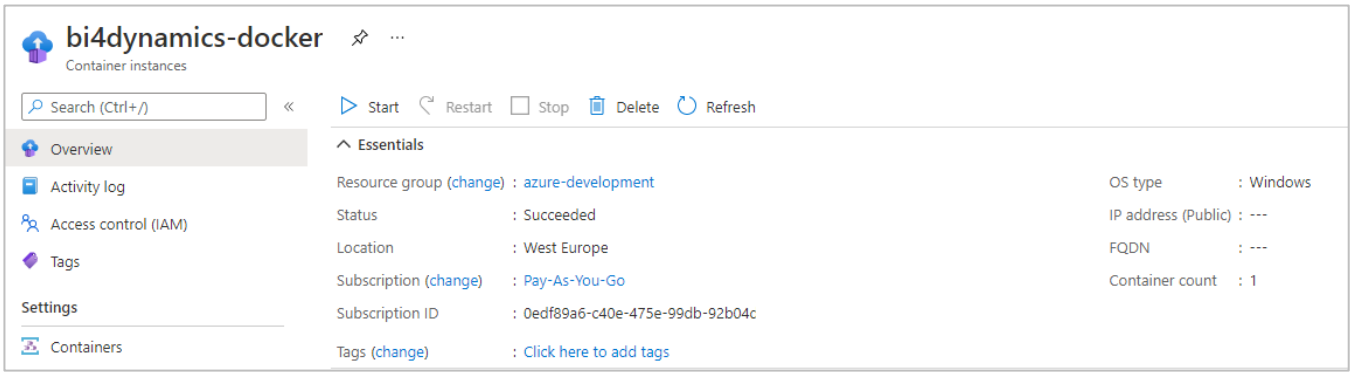
1 Process Automation #1 – Start Container Instance

1.1 Introduction

This automation process is for an Azure Container instance (Docker), which is a light virtual machine, based on BI4Dynamics image. These Container instances are used for running table export from BC to Blob storage. Through Logic apps docker will run on a scheduled day and time. Logic app will automatically start and terminate the docker after finishing the export.

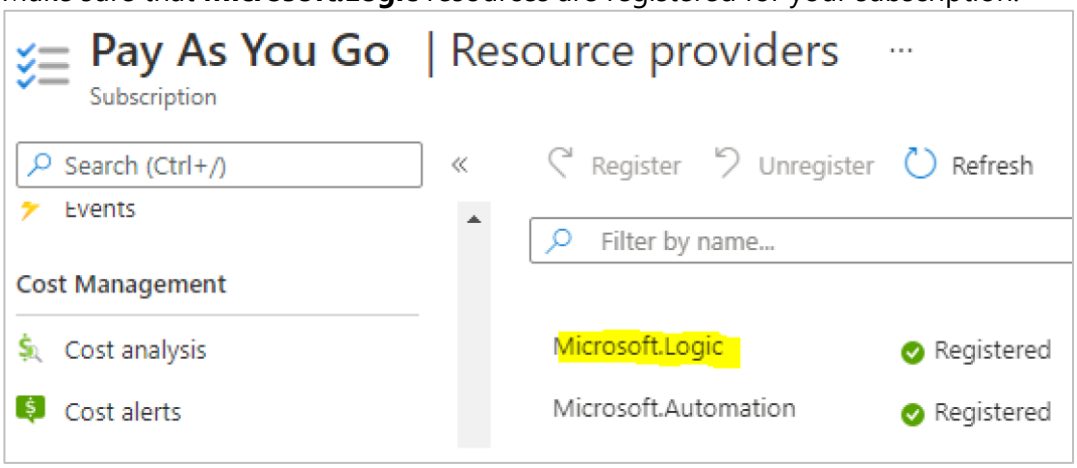
1.2 Prerequisite

For this manual you will need a working Container instance, which exports table data from BC to Blob storage. Picture below shows Container instance settings made as per instructions of BI4Dynamics in BI4Dynamics Infrastructure Installation (Local server + Azure resources).



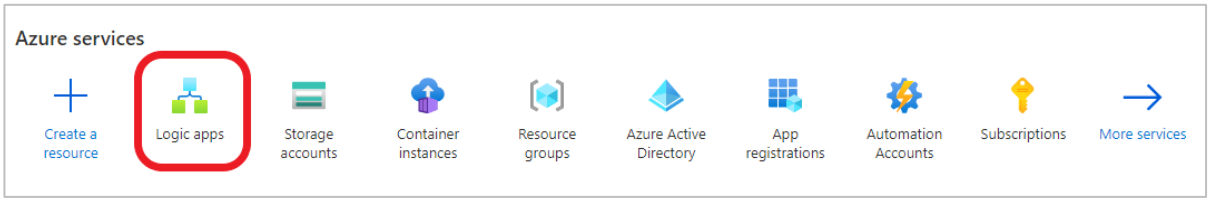
- Logic Apps (to be setup here).

Make sure that **Microsoft.Logic** resources are registered for your subscription.



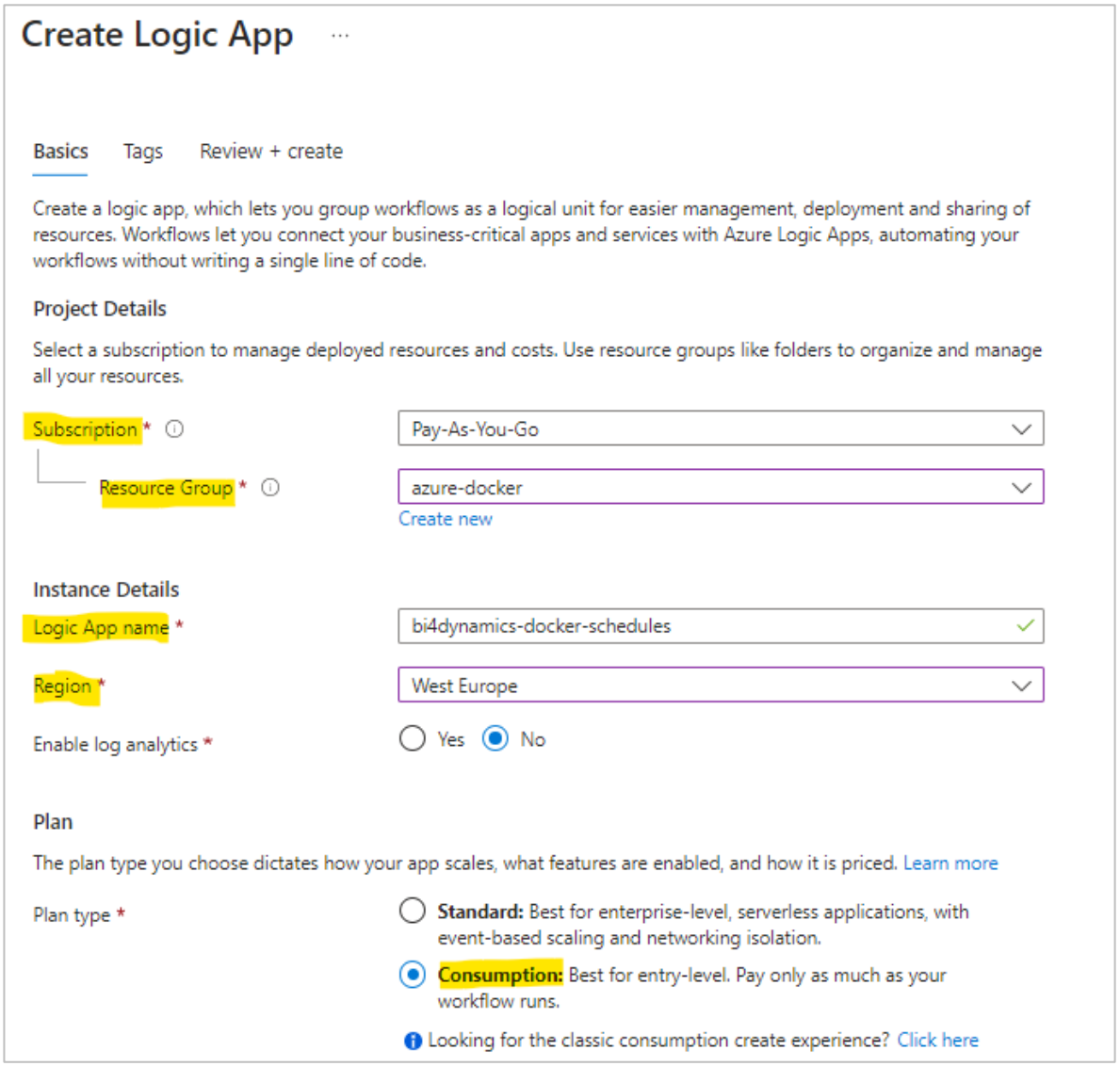
1.3 Setup Logic App

Search for **Logic Apps** in Azure.

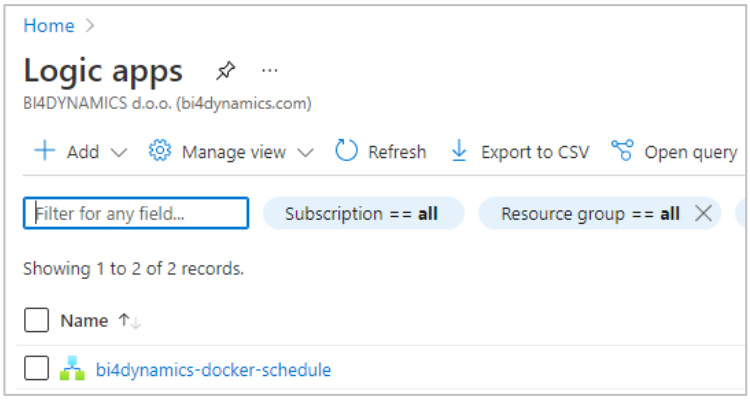


Add a new Logic App**.**

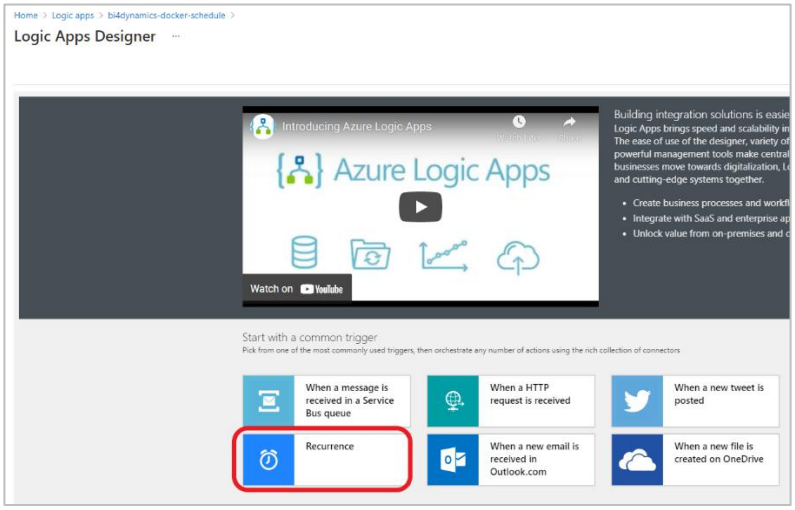
Enter **Subscription**, **Resource Group** and create a meaningful **Name** for your logic app.
Select the Region and choose **Consumption** as a Plan type.
Click **Review + create** and select **Create** in the next window.



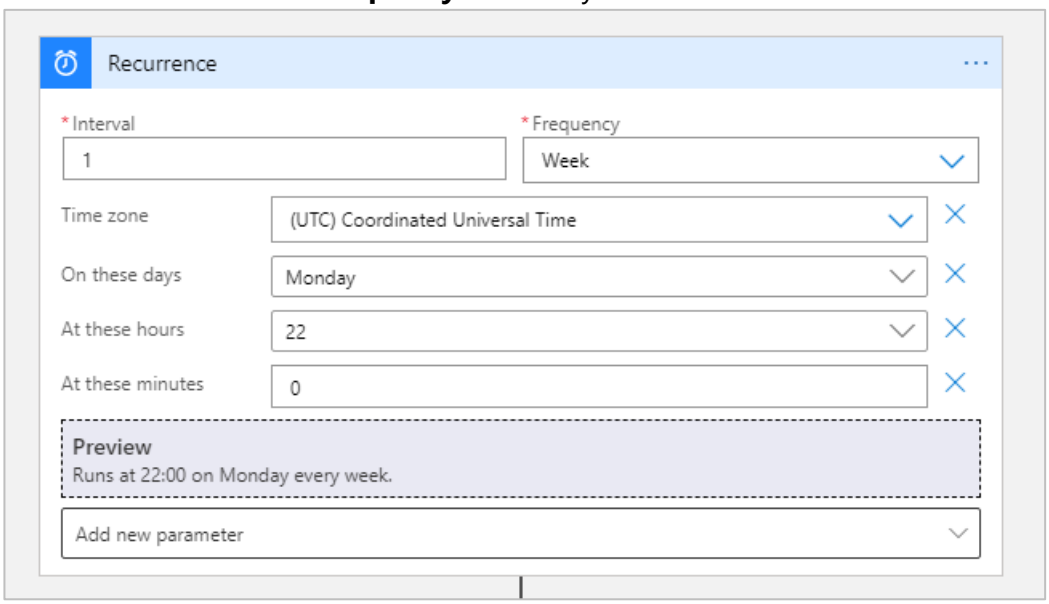
Once deployment is complete, go to **Logic apps** and open the newly created application.



Logic apps designer will open with premade templates to use.
Select **Recurrence** in the template or search for it in the search dialog.

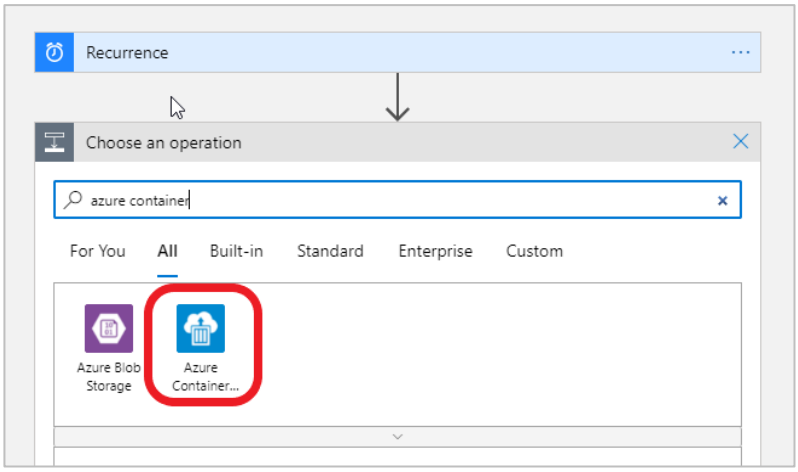


Select the **Interval** and **Frequency** at which you decide the docker should be run.

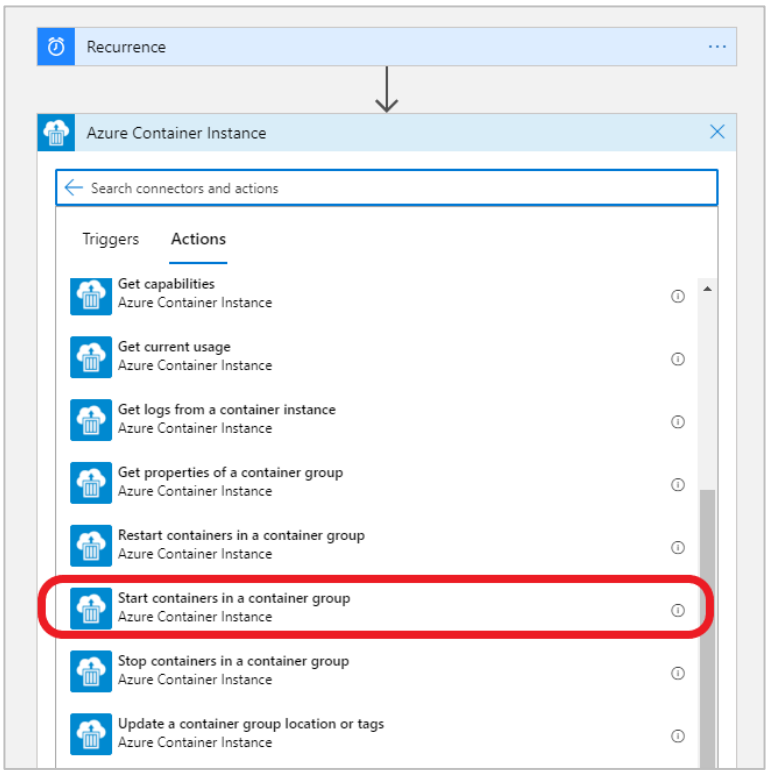


Note: If the selected Frequency is **Week**, you can add new parameters which set the days, hours, and minutes when the Virtual Machine should start.

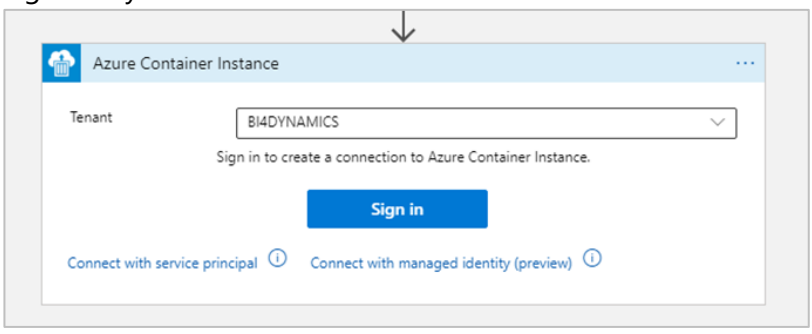
Click + **New step**, search for **Azure Container instance** and select it.



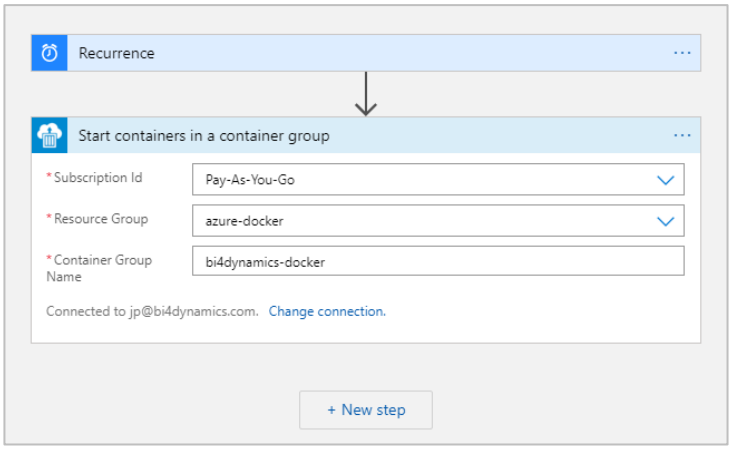
In the drop-down menu select Start containers in a container group.



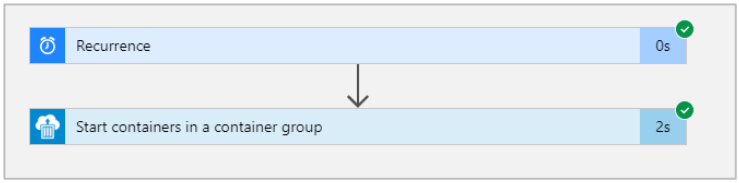
Sign into your **Tenant**.



Enter your **Subscription ID, Resource Group** and **Container Group Name** (docker).

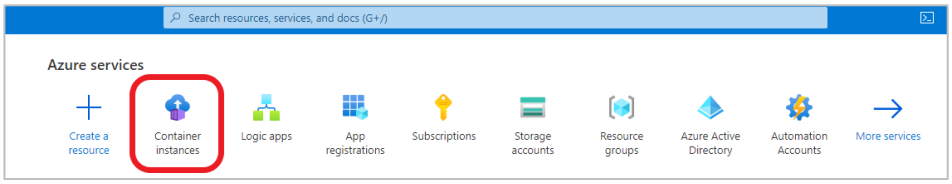


Click **Save** in the top left of the designer and press **Run** to test if the application is working correctly.

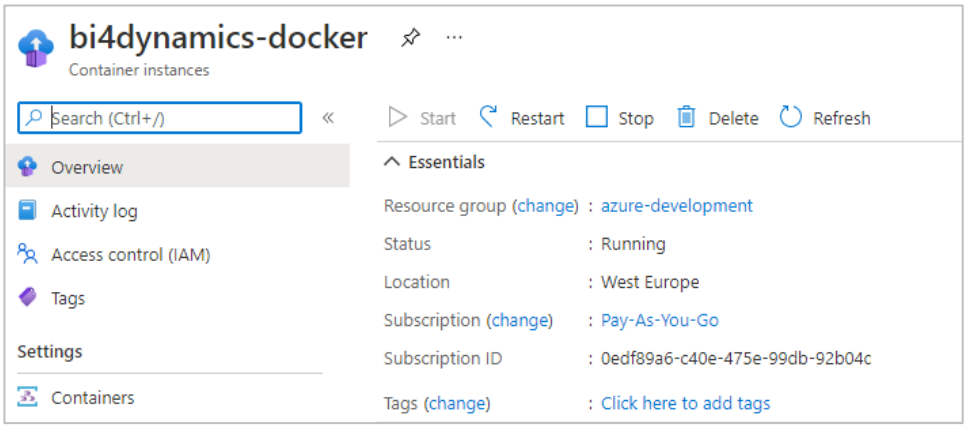


1.4 Test logic app

Go to Container instances.



Check the selected container instance (docker) if it is being **Created** or already **Running**.



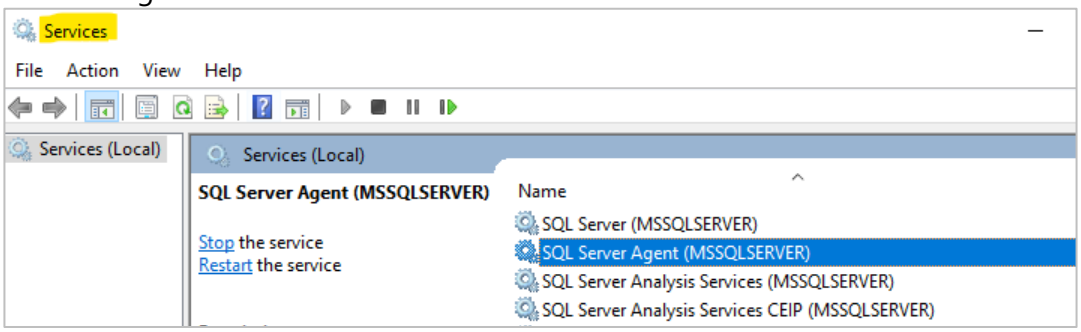
You have now successfully created and tested a logic app that automatically starts container instance at specific times.

2 Process Automation #3 – Start SQL server Agent (VM)

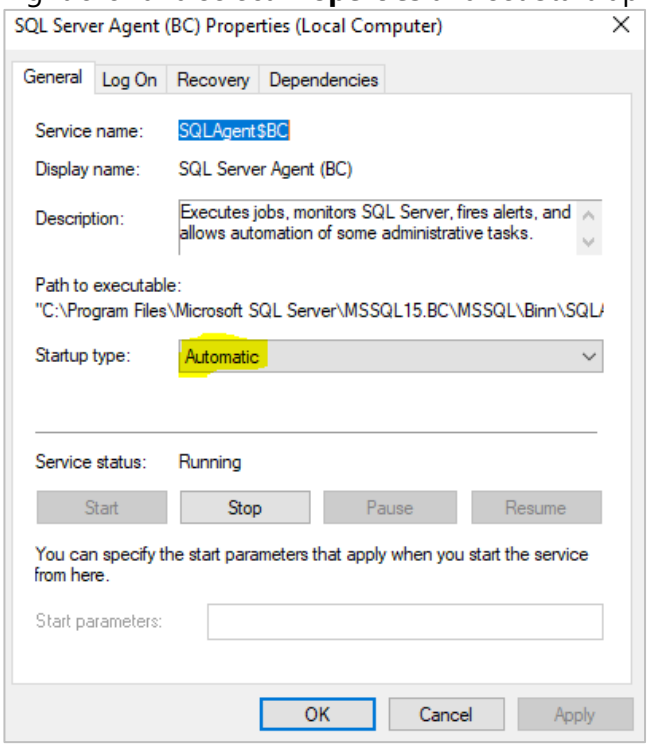
When Virtual machine is running, it is ready to process data. This process is triggered by SQL Server Agent feature, a part of SQL server.

2.1 Enable SQL Server agent

Go to **Services** and find the **SQL Server Agent** service. If you are using newly created Virtual Machine, it will probably be the only SQL Server Agent, but if you are running more SQL server engines, there may be more Agents.



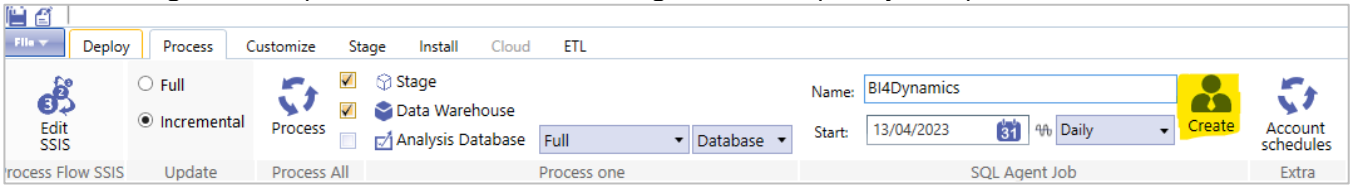
Right click and select **Properties** and set Start-up Type to Automatic.



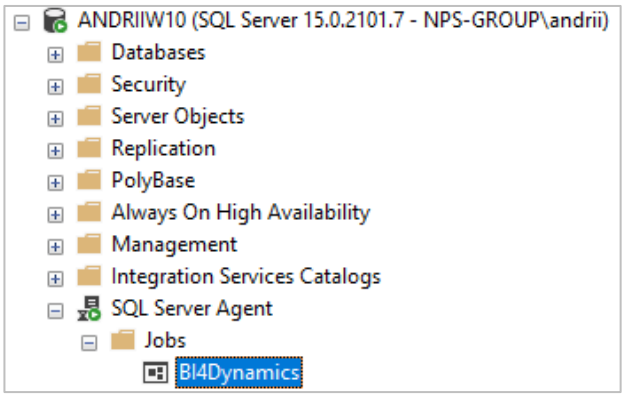
Note: make sure that user running service is a domain admin user (not a service) and has permissions needed to process data warehouse and analysis services. On VM this would be the VM admin user.

2.2 Setup SQL Server Agent

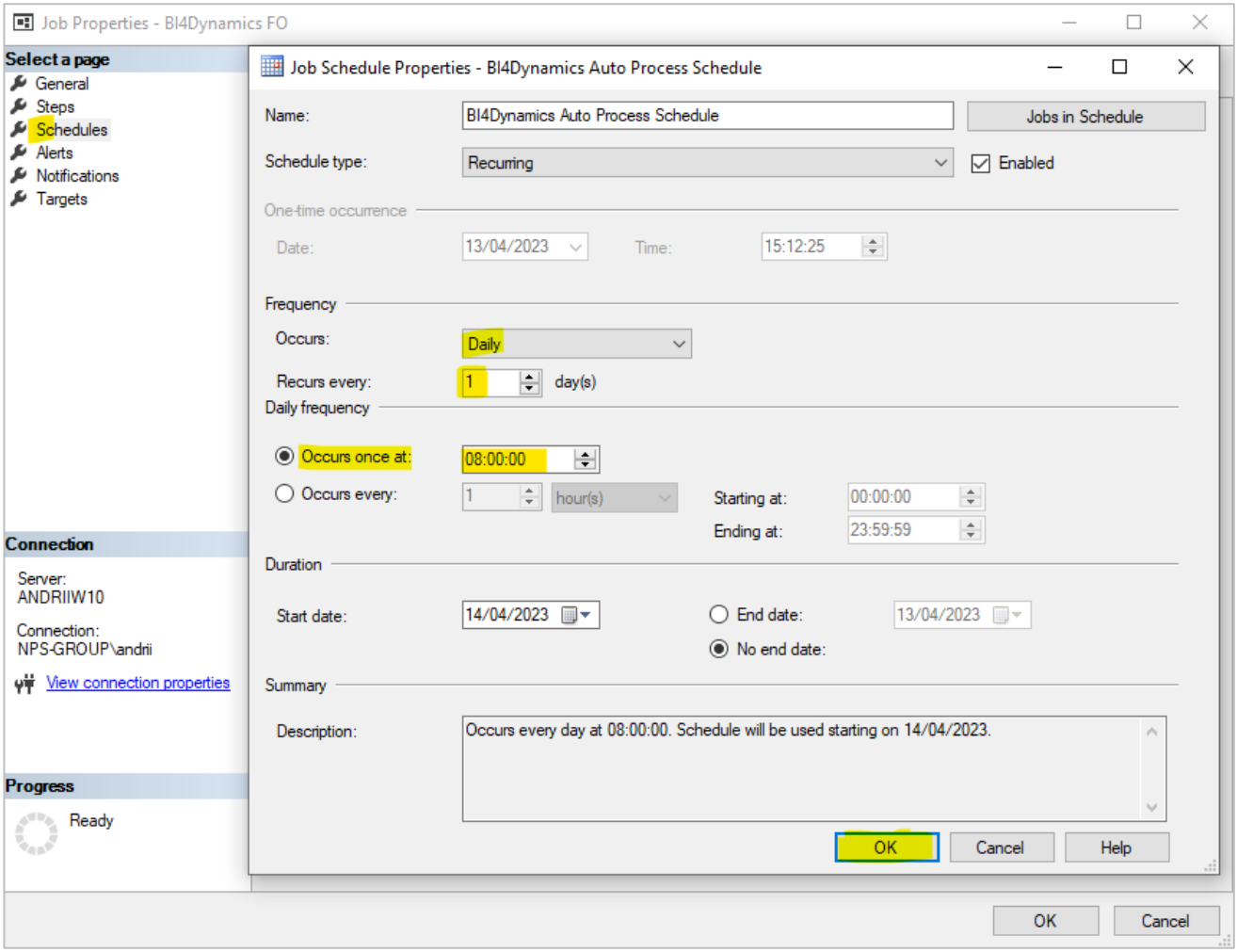
SQL Server agent conducts processing of stage, data warehouse and analytics, bringing new data to users. To set SQL Agent Job open ***Process*** tab, set SQL Agent Job frequency and press **Create**:



Open SQL Server Management Studio, navigate to ***Properties*** of the Agent Job that you have created.



Press ***Schedules***, click on **Edit**. Set properties for Job Agent and press ***OK***.

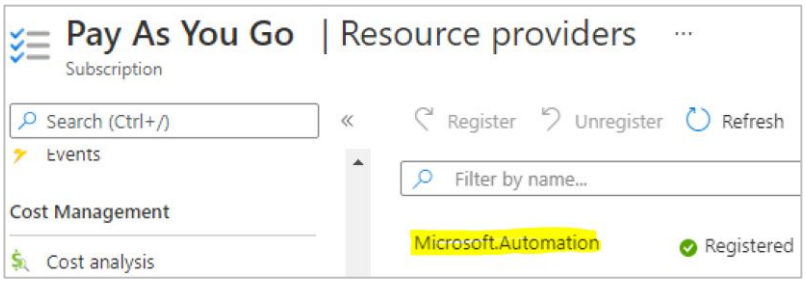


3 Process Automation #4 – Start and Stop Azure Analysis Services

This part of documentation is intended to explain the process of scheduling the work of Azure Analysis Services. It will allow the Analysis Services to start and stop on scheduled days and time based on the business requirements of the end-users.

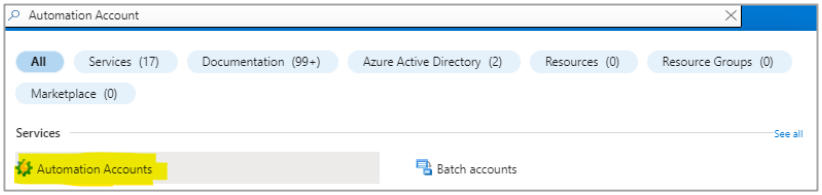
3.1 Prerequisities

- Azure Analysis Services: creation described in document *"Application Installation (Azure VM)"*.
- Registered resource **Microsoft.Automation** for the subscription.

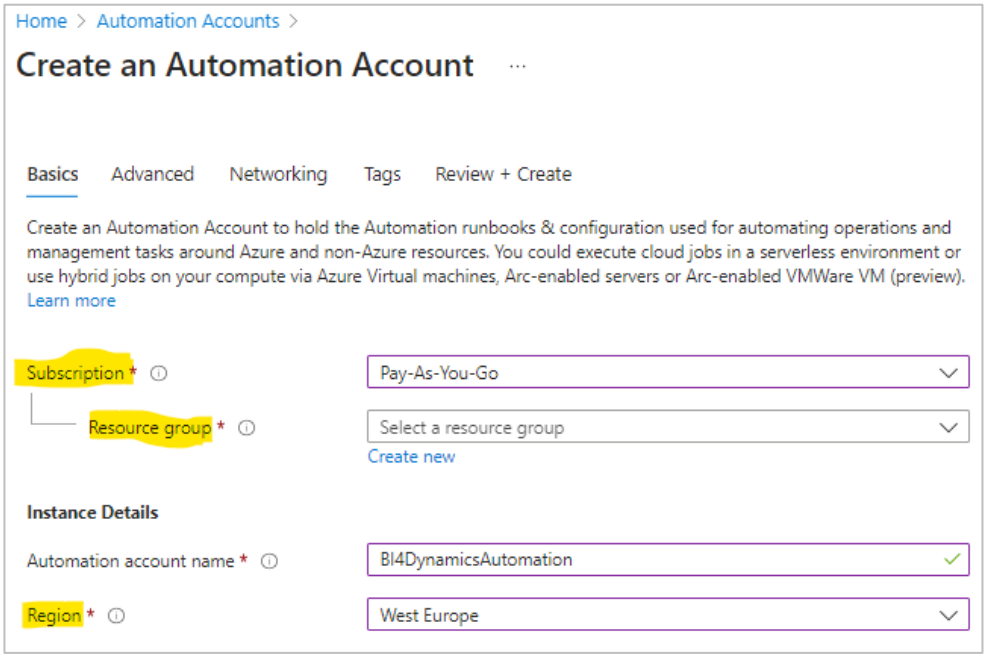


3.2 Instructions

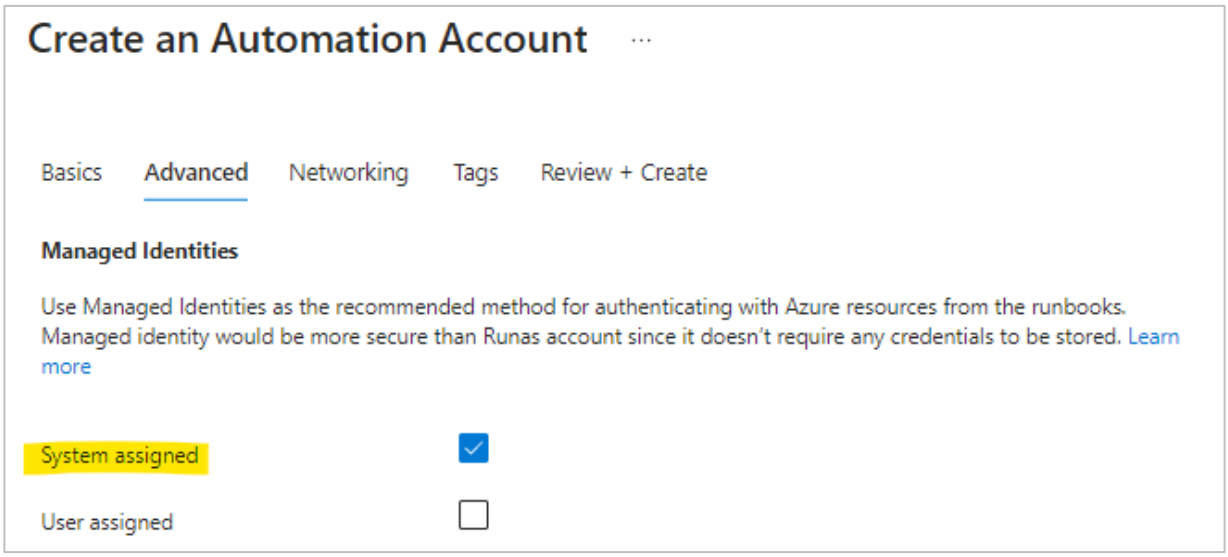
Go to the Azure Portal and search for **Automation Accounts**:



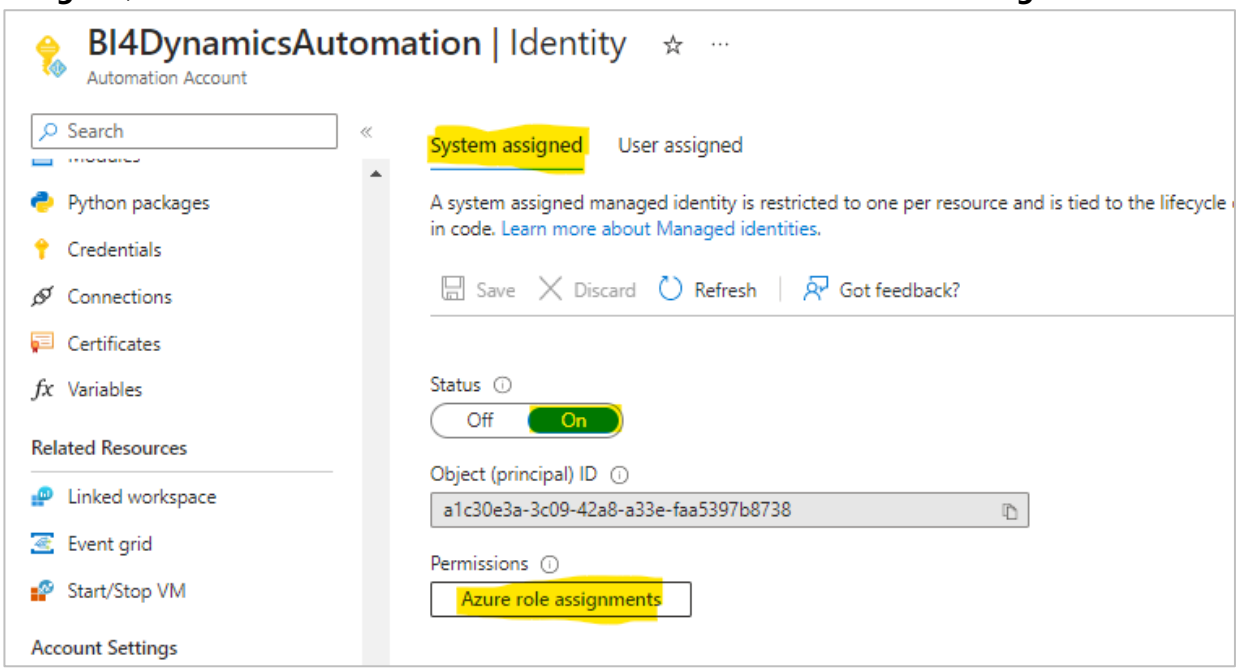
Create a new **Automation Account** under the same **Subscription** and in the same **Resource Group** and **Region** as the **Analysis Services**:



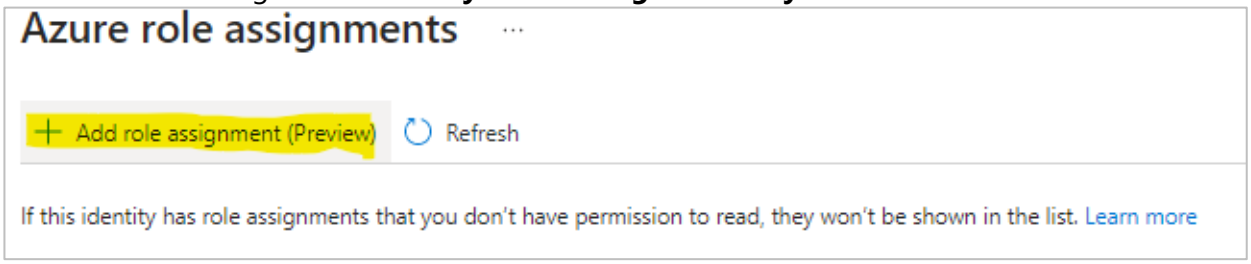
In the tab **Advanced**, select **System Assigned Managed Identity**:



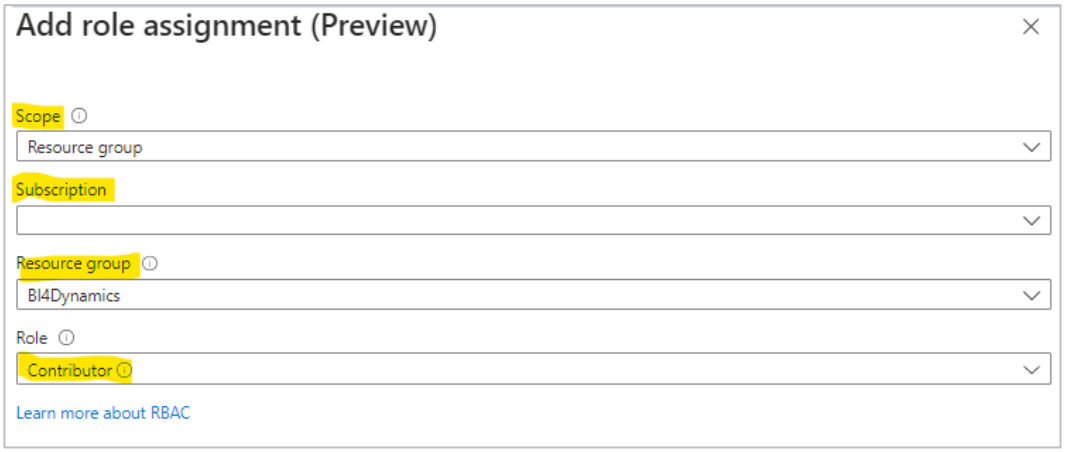
Once the **Automation Account** is created, go to **Account Settings > Identity**. In the tab **System Assigned**, make sure that the **Status** is set to **On** and click on **Azure Role Assignments**:



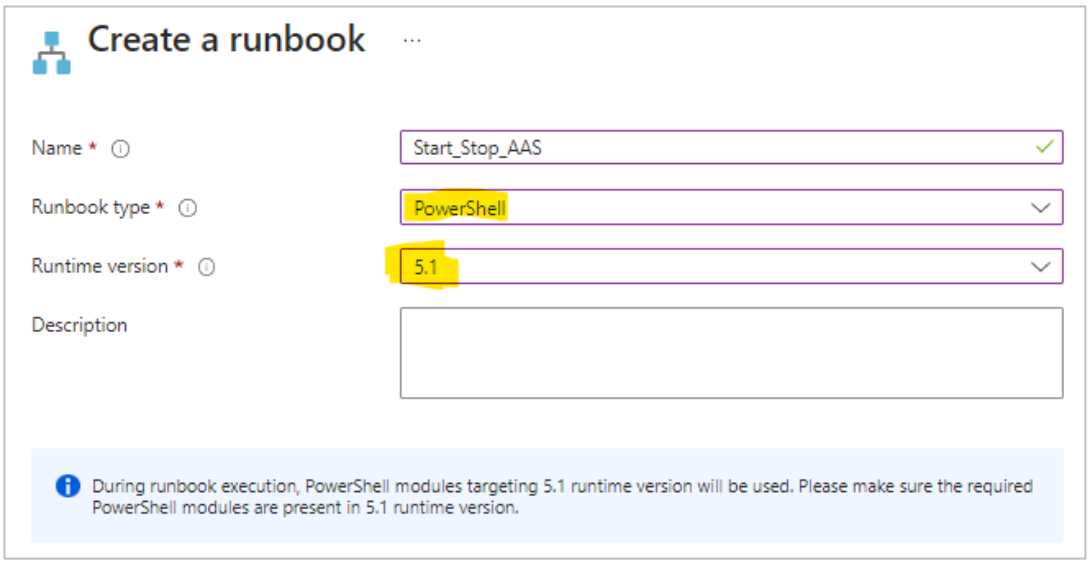
Add a new role assignment to the **System Managed Identity:**



Under the option **Scope** select the option **Resource Group**.
Specify the **Subscription** and **Resource Group** in which Analysis Services are located.
Under the option **Role** select **Contributor**.

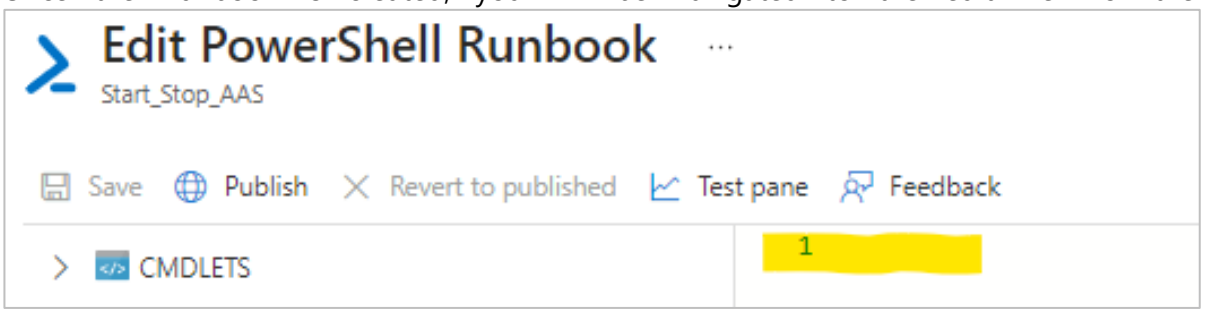


Once role assignment is added, navigate to **Process Automation > Runbooks** and create a new one.
Give to a new **Runbook** a meaningful name such as **Start_Stop_AAS**.
In the **Runbook Type** select the option **PowerShell**.
In the **Runtime Version** select the option **5.1**.



Important: Different **Runtime** Version might lead to the error in the execution of the script. Syntax for authentication might differ between PowerShell versions**.**

Once the Runbook is created, you will be navigated to the edit view of the Runbook:



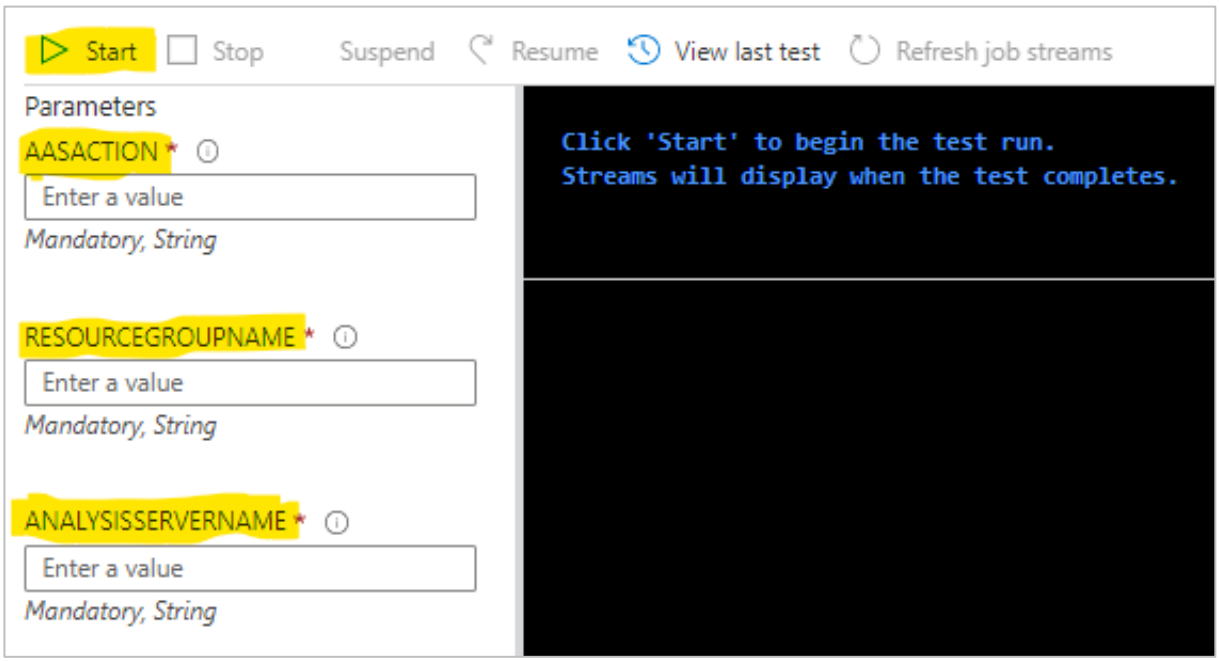
Insert the following script to the command lines:

```
# Parameters
[CmdletBinding()]
param(
    [Parameter(Mandatory=$True,Position=0)]
    [ValidateSet('Start','Stop')]
    [string]$AasAction,

    [Parameter(Mandatory=$True,Position=1)]
    [ValidateLength(1,100)]
    [string]$ResourceGroupName,

    [Parameter(Mandatory=$True,Position=2)]
    [ValidateLength(1,100)]
    [string]$AnalysisServerName
)
# Keep track of time
$StartDate=(GET-DATE)
# Log in to Azure with AZ (standard code)
Write-Verbose -Message 'Connecting to Azure'
# Name of the Azure Run As connection
$ConnectionName = 'AzureRunAsConnection'
try {
        $AzureContext = (Connect-AzAccount -Identity).context
    }
catch{
        Write-Output "There is no system-assigned user identity. Aborting.";
        exit
    }
# Getting the AAS for testing and logging purposes
$myAzureAnalysisServer = Get-AzAnalysisServicesServer -ResourceGroupName $ResourceGroupName -
Name $AnalysisServerName
if (!$myAzureAnalysisServer)
{
    Write-Error "$($AnalysisServerName) not found in $($ResourceGroupName)"
    return
}
else
{
    Write-Output "Current status of $($AnalysisServerName): $($myAzureAnalysisServer.State)"
}
# Check for incompatible actions
if (($AasAction -eq "Start" -And $myAzureAnalysisServer.State -eq "Succeeded") -Or ($AasAction -
eq "Stop" -And $myAzureAnalysisServer.State -eq "Paused"))
{
    Write-
Error "Cannot $($AasAction) $($AnalysisServerName) while the status is $($myAzureAnalysisServer.State
)"
    return
}
# Resume Azure Analysis Services
elseif ($AasAction -eq "Start")
{
    Write-Output "Now starting $($AnalysisServerName)"
    $null = Resume-AzAnalysisServicesServer -ResourceGroupName $ResourceGroupName -
Name $AnalysisServerName
}
# Pause Azure Analysis Services
else
{
    Write-Output "Now stopping $($AnalysisServerName)"
    $null = Suspend-AzAnalysisServicesServer -ResourceGroupName $ResourceGroupName -
Name $AnalysisServerName
}
# Show when finished
$Duration = NEW-TIMESPAN –Start $StartDate –End (GET-DATE)
Write-
Output "Done in $([int]$Duration.TotalMinutes) minute(s) and $([int]$Duration.Seconds) second(s)"
```

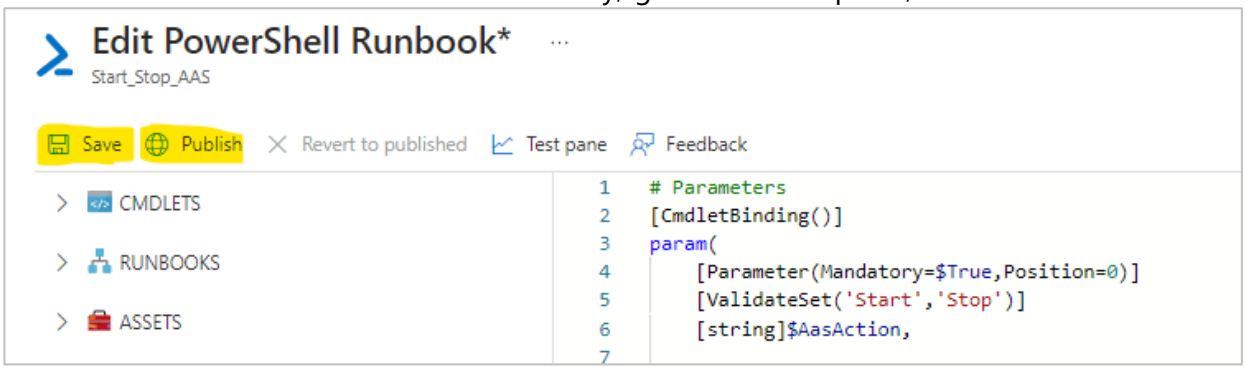
After that got to the **Test Pane** and fill in the required parameters:

- Under the option **AASACTION** write **Start** (if the Analysis Services are turned on - **Stop**).
- Under the option **RESOURCEGROUPNAME** insert the name of the **Resource Group.**
- Under the option **ANALYSISSERVERNAME** insert the name of **Azure Analysis Services.**

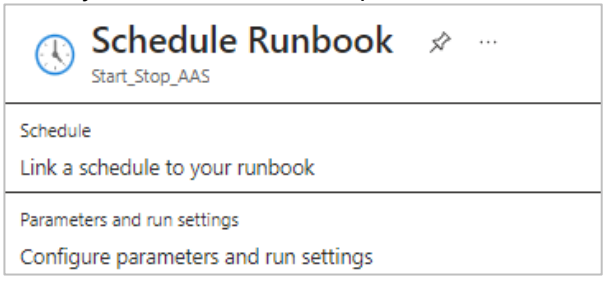
Once the parameters are inserted, click on **Start:**



Once the command was executed successfully, go to the Edit panel, **Save** and **Publish** the runbook:



Once it is published, click on the option **Link to the schedule.**
There you will need to set up the schedule and parameters for the Runbook Execution:



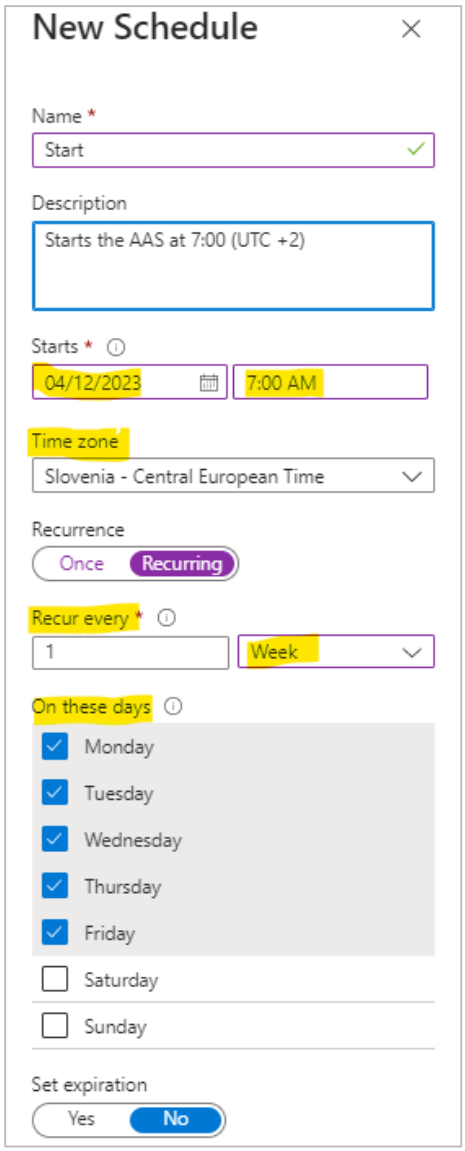
Parameters and run settings for the schedule should be configured similarly to the previous step.
Only **AASACTION** will differ based on the command of the schedule (Start or Stop).

Note:
The automation schedules for Start and Stop commands must be created separately.

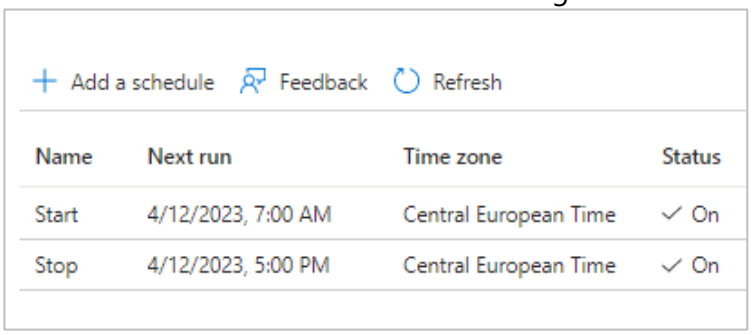
In the schedule settings provide a name to the new schedule. We suggest giving it a name corresponding to the executed command (Start or Stop). Also, you can provide additional details in the Description.

Important:
Do not forget to specify the correct **Time Zone** according to which the time of automation will be scheduled.

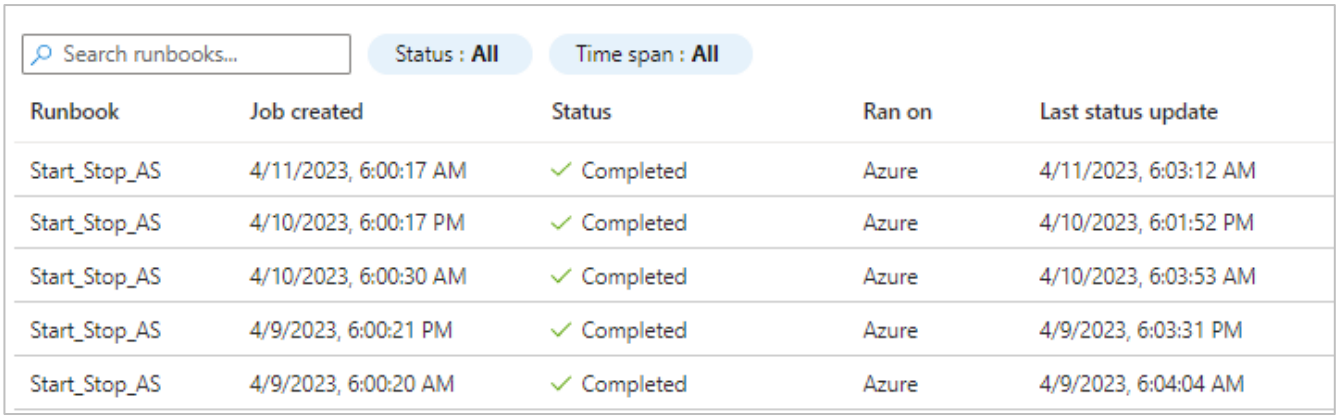
As a next step, change **Recurrence** from Once to Recurring and set up **Recur every** option to once a Day or Week. In case of week, the schedule can be set up at the specific days of the week so that Saturday and Sunday could be excluded from the automation as on the screenshot on the right.



The final schedules should look as following:



The execution of the schedules can be monitored in the **Process Automation > Jobs** tab:



4 Process Automation - Timing Schedule

Here is an example of processing schedule for daily update:

Step	Step description	Start Time	Duration	Comment
1	Start Container instance	22:00	45 min	BC export to data lake can run anytime after BC users are finishing their daily work. This process time can vary 30% (!) day by day, exporting same amount of data, in the after-office hours when no-one is using BC. Keep enough buffer time for next step.
2	Start Azure Analysis Services	07:00	2-3 min	Azure AS must be ready when DW processing start
3	Start SQL Server Agent	07:15	20 min	DW processing (data are in Azure AS)
4	Stop Azure Analysis Services	17:00		AAS will run during business hours when users are querying data.