

BI4Dynamics Process Automation

How to automatically start and stop Azure Virtual Machine and Analysis Services

Last update: April 2023
Version 1.2
Revision 2.0

Table of Contents

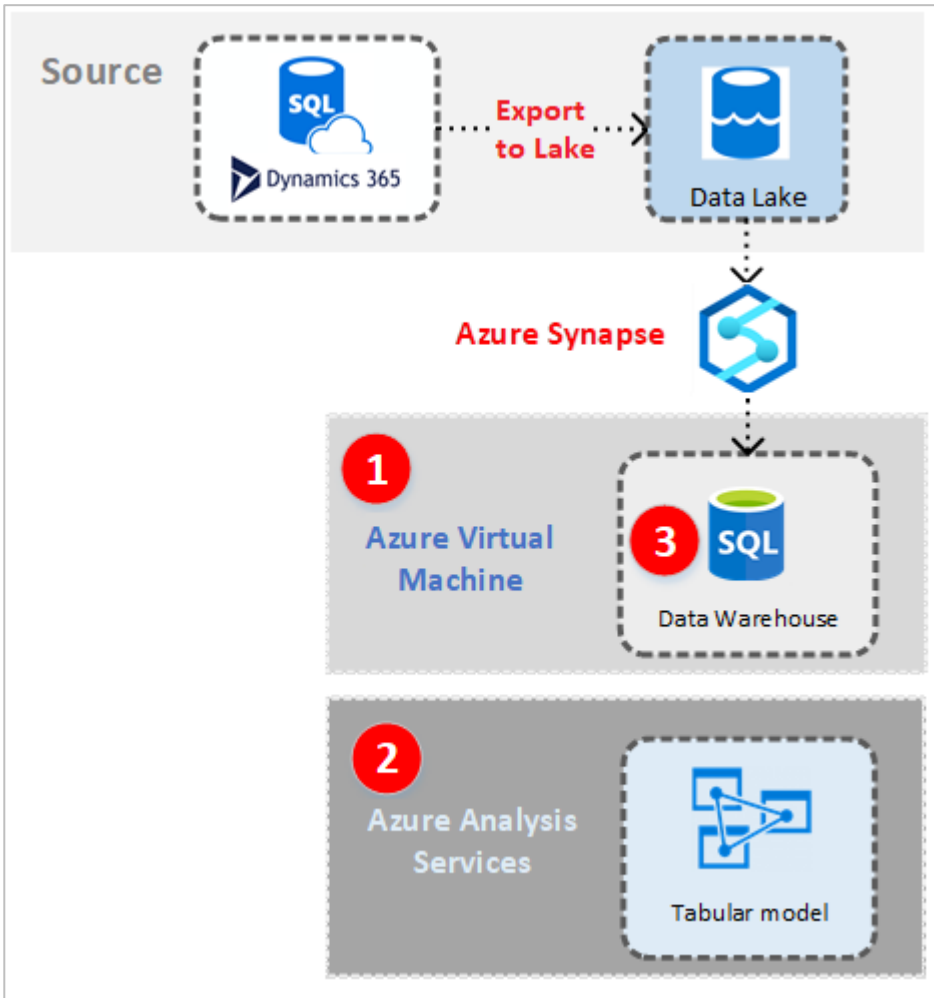
Introduction	3
1 Azure Virtual Machine: Start and Stop	4
1.1 Prerequisites.....	4
1.2 Start Virtual Machine	4
1.3 Stop (Deallocate) Virtual Machine.....	8
2 Azure Analysis Services: Start and Stop.....	10
2.1 Prerequisites.....	10
2.2 Instructions	10
3 SQL Server Agent: Start Job	16
3.1 Enable SQL Server agent.....	16
3.2 Setup SQL Server Agent	17
4 Timing examples	18

Introduction

Data processing automation allows to regularly process data without human involvement.

Process automation consists of scheduling following resources:

- 1) Azure Virtual Machine
- 2) Azure Analysis Services
- 3) SQL Server Agent job within Azure VM



1 Azure Virtual Machine: Start and Stop

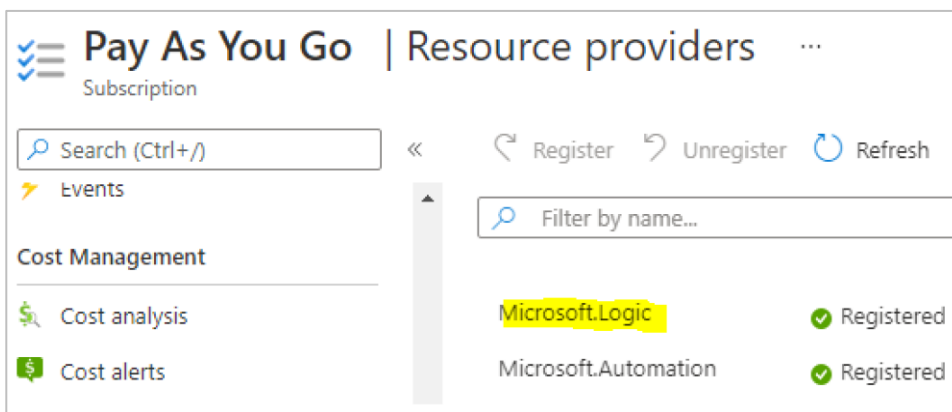
This automation process is for an Azure Virtual Machine. Through Logic apps virtual machine will run on a scheduled day and time. Logic app will automatically start and terminate the virtual machine after the processing is complete.

Note: We will create two logic apps, one for starting the VM and one for deallocating (stopping) it.

1.1 Prerequisites

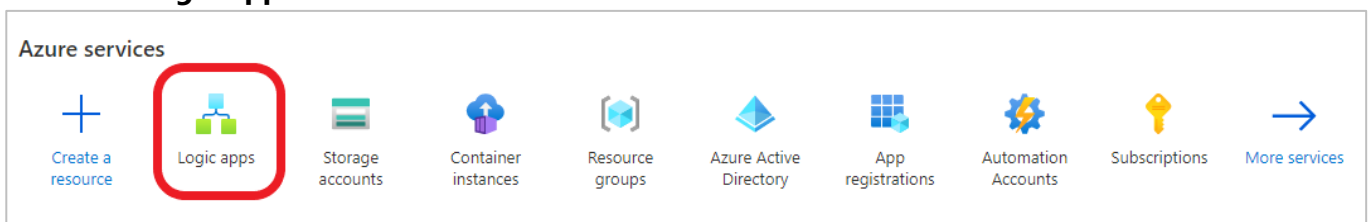
For this step you will need:

- Logic Apps (to be setup here).
- **Microsoft.Logic** and **Microsoft.Automation** resource providers registered for your subscription.

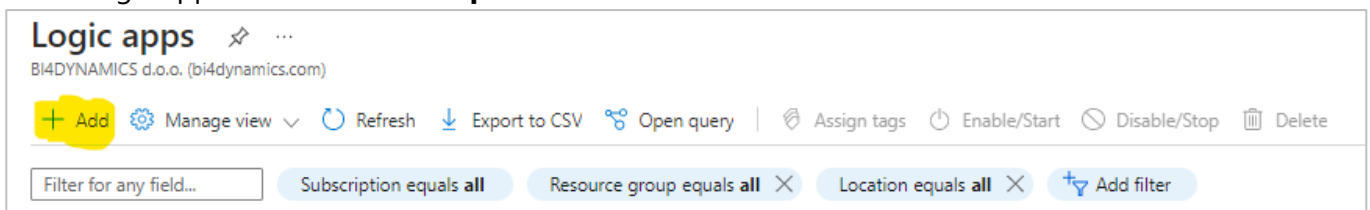


1.2 Start Virtual Machine

Search for **Logic Apps** in Azure.



Add a logic app and select **Consumption**.



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

Create Logic App

Basics Tags Review + create

Create a logic app, which lets you group workflows as a logical unit for easier management, deployment and sharing of resources. Workflows let you connect your business-critical apps and services with Azure Logic Apps, automating your workflows without writing a single line of code.

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ BI4DYNAMICS

Resource Group * ⓘ bidynamics
[Create new](#)

Instance Details

Type * Consumption Standard
 ⓘ Looking for the classic consumption create experience? [Click here](#)

Logic App name * start_vm ✓

Region * Southeast Asia

Enable log analytics * Yes No

⚠ There are no log analytics workspace resources in the selected subscription. In order to enable log analytics, either create a new log analytics workspace resource or switch to a subscription which already has one.

Review + create < Previous Next : Tags >

✓ Your deployment is complete

Deployment name: Microsoft.Web-LogicAppConsumption-Portal-8... Start time: 10/15/2021, 1:10:5
 Subscription: BI4DYNAMICS Correlation ID: e20a37c8-b05
 Resource group: bidynamics

Go to **Logic apps** and open the newly created application.


Logic apps

+ Add ⚙ Manage view ▾ ↻ Refresh ↓ Export to CSV 🔗 Open query

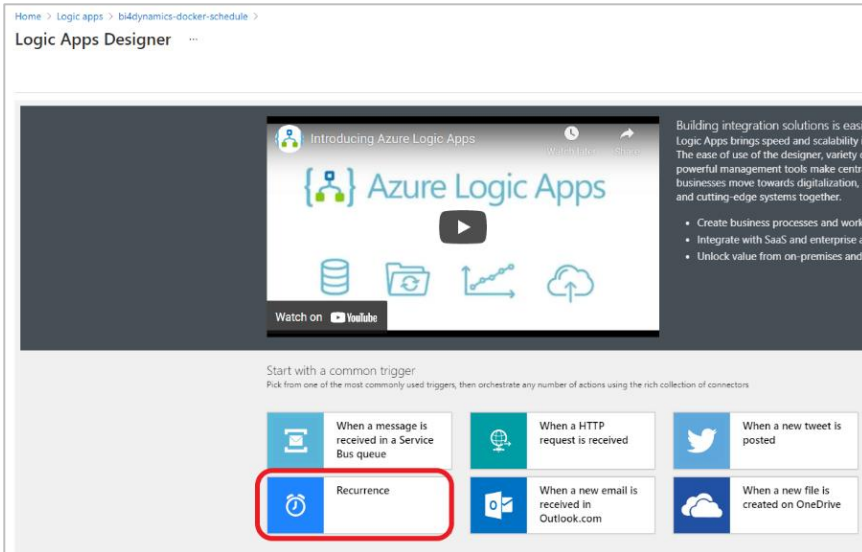
Filter for any field... Subscription == all Resource group == all ✕

Showing 1 to 3 of 3 records.

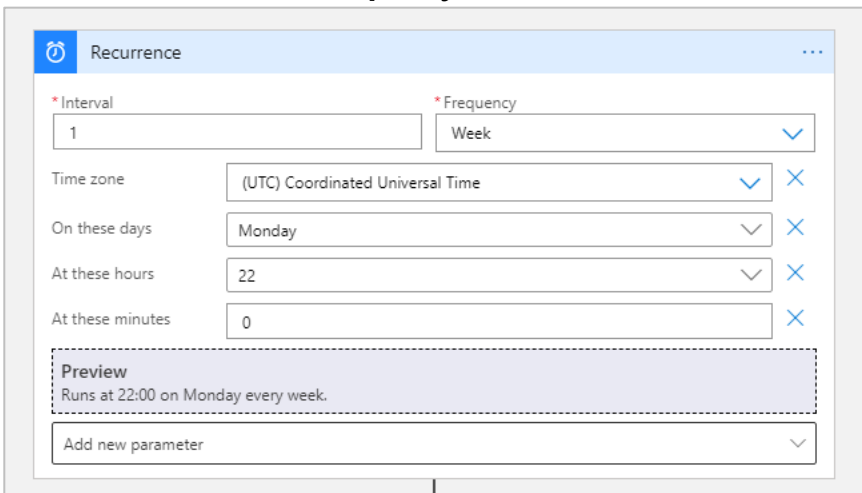
Name ↑↓

 start_vm

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

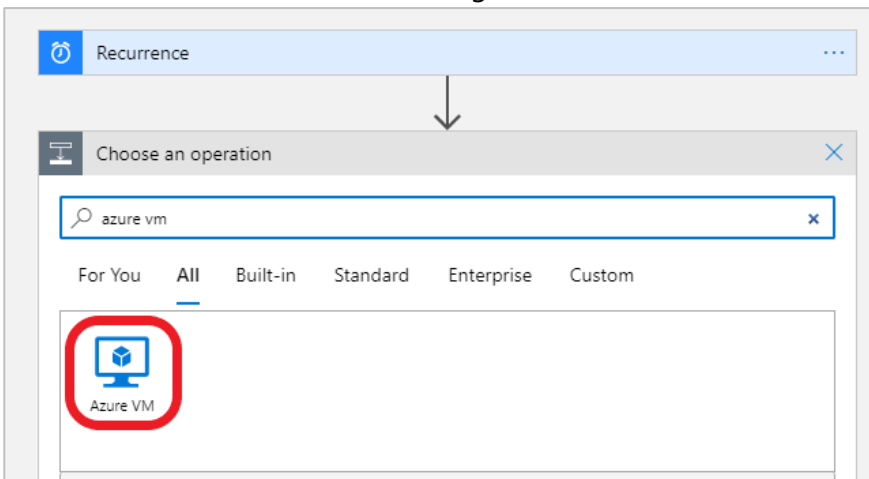


Select the **Interval** and **Frequency** at which virtual machine should start.

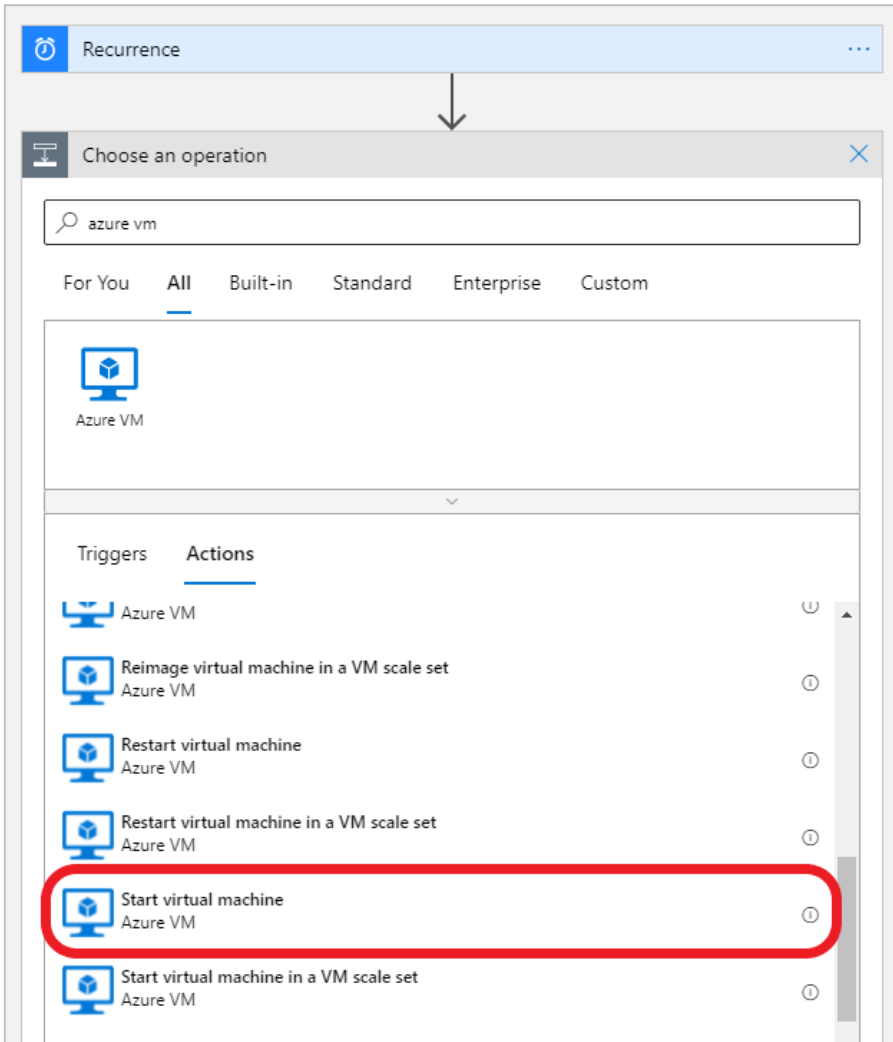


Note: If the selected Frequency is **Week**, you can add new parameters which set the days, hours, and minutes when the Virtual Machine should start.

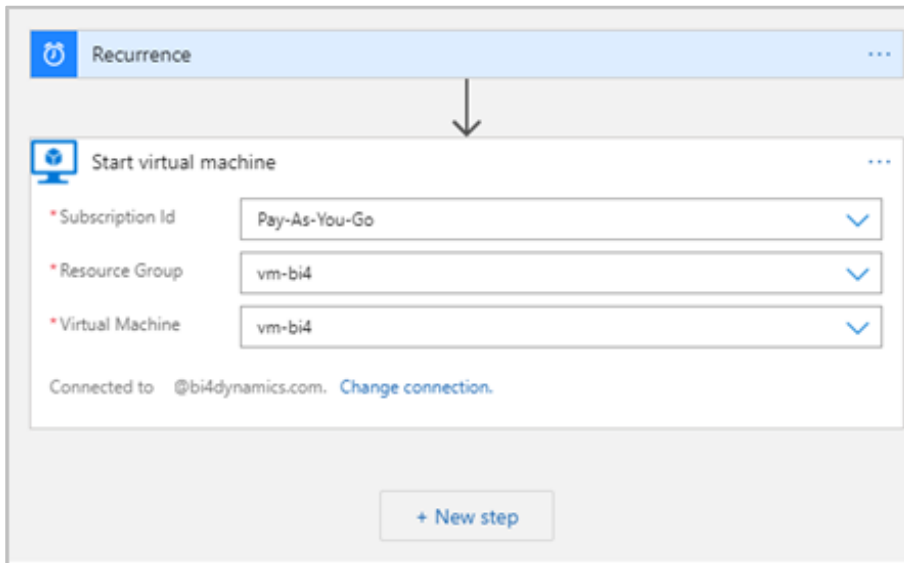
Search for **Azure VM** in search dialog and select it.



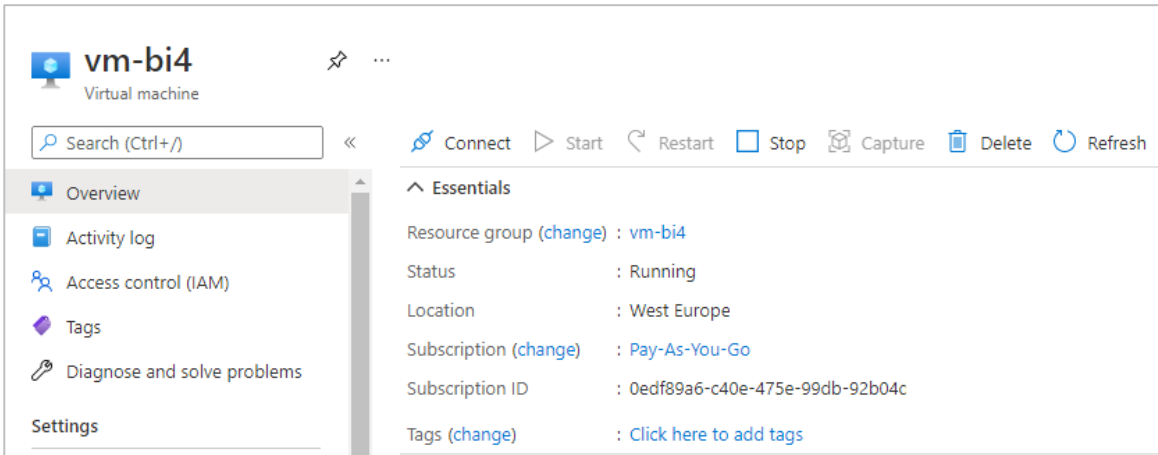
Select **Start virtual** machine option.



Insert values for Subscription ID, Resource Group and Virtual Machine name.



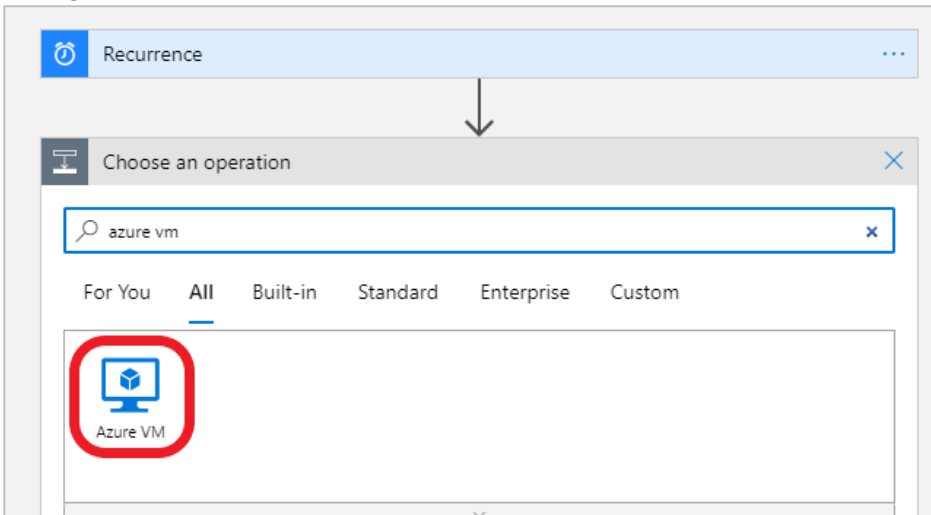
Next step is to **Save** and **Run** the application and go to **Virtual Machine** to check if it is **Running**.



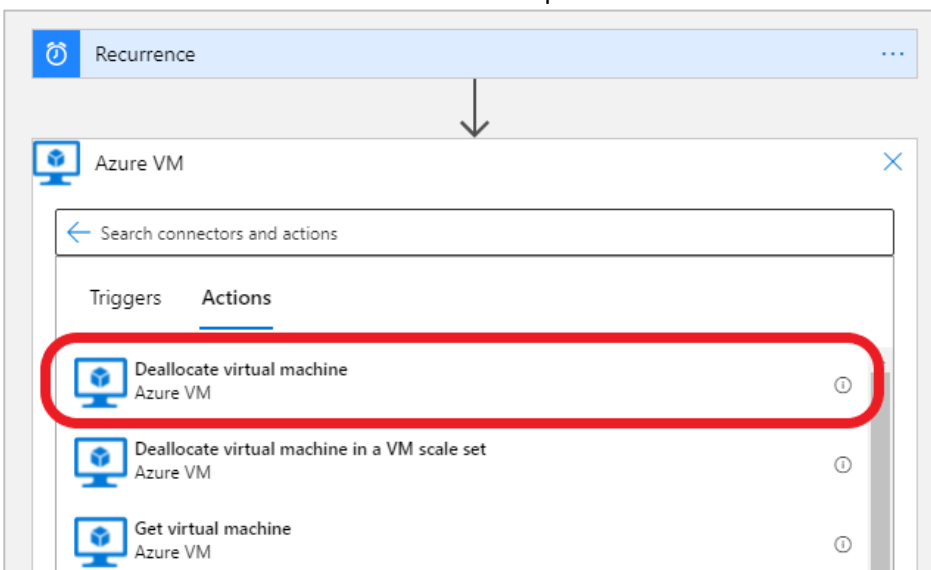
1.3 Stop (Deallocate) Virtual Machine

Process automation for Virtual Machine Deallocation is almost identical to Start VM Logic app.

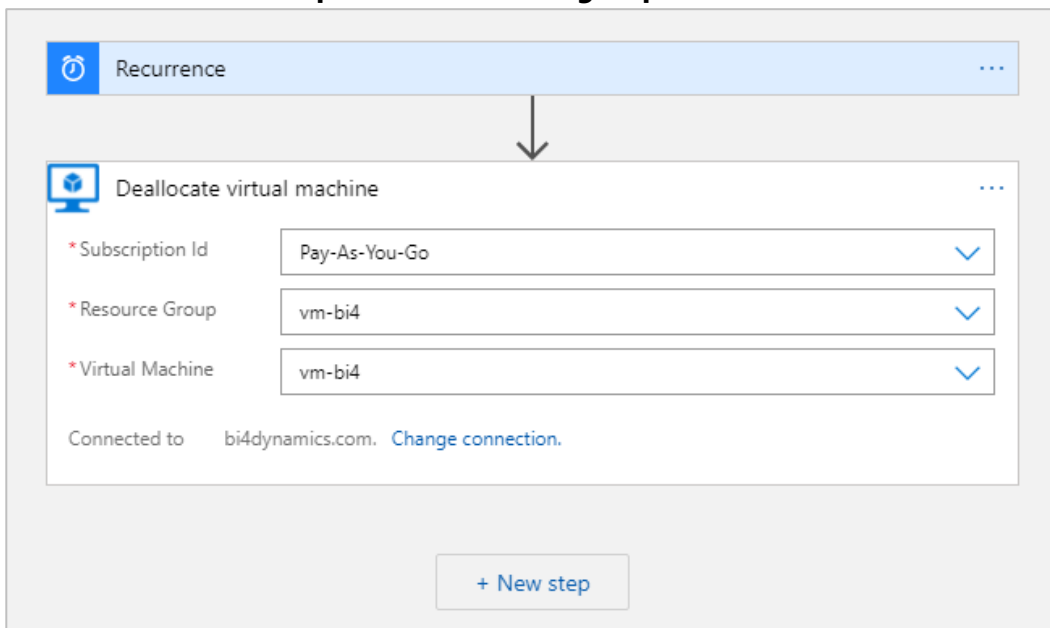
In **Logic Apps** select **Recurrence**, add a **new step** and search for **Azure VM**.



Next select **Deallocate virtual machine** option.



Insert values for **Subscription ID**, **Resource group** and **Virtual Machine** name.



Recurrence

Deallocation virtual machine

* Subscription Id: Pay-As-You-Go

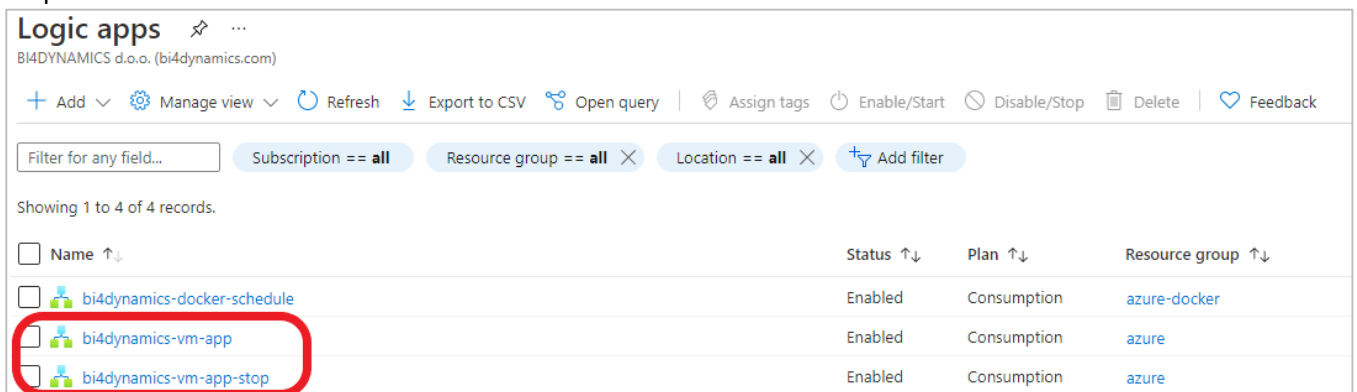
* Resource Group: vm-bi4

* Virtual Machine: vm-bi4

Connected to: bi4dynamics.com. [Change connection.](#)

+ New step

Click **Save** and exit Logic Apps Designer. Go to **Logic Apps** and check for apps VM start and VM stop.



Logic apps

BI4DYNAMICS d.o.o. (bi4dynamics.com)

+ Add Manage view Refresh Export to CSV Open query Assign tags Enable/Start Disable/Stop Delete Feedback

Filter for any field... Subscription == all Resource group == all Location == all Add filter

Showing 1 to 4 of 4 records.

Name	Status	Plan	Resource group
bi4dynamics-docker-schedule	Enabled	Consumption	azure-docker
bi4dynamics-vm-app	Enabled	Consumption	azure
bi4dynamics-vm-app-stop	Enabled	Consumption	azure

To check if Logic apps are properly working first run the start VM app, after the Virtual machine is running, run the stop VM app and check if it is allocated.

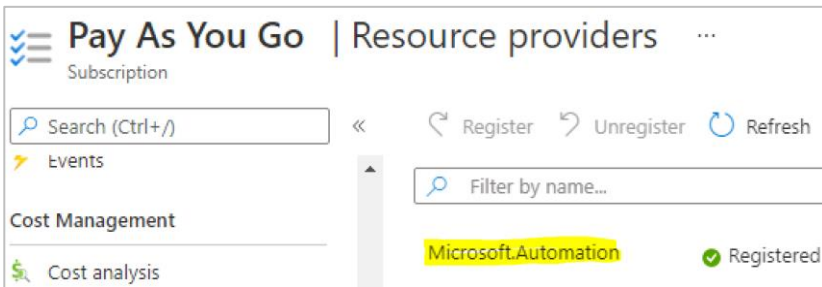
You have now successfully created a logic app that automatically starts the virtual machine at specified times and a logic app that automatically stops (deallocates) the virtual machine at specified times.

2 Azure Analysis Services: Start and Stop

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.

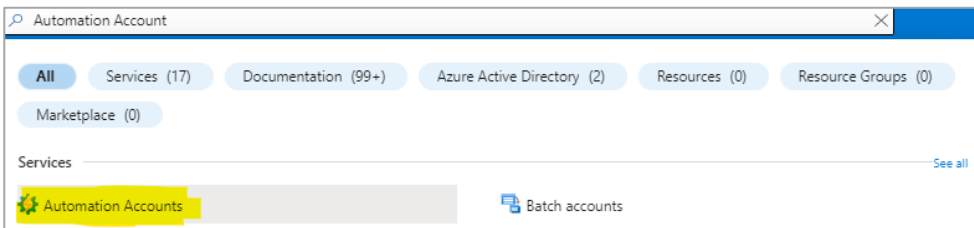
2.1 Prerequisites

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

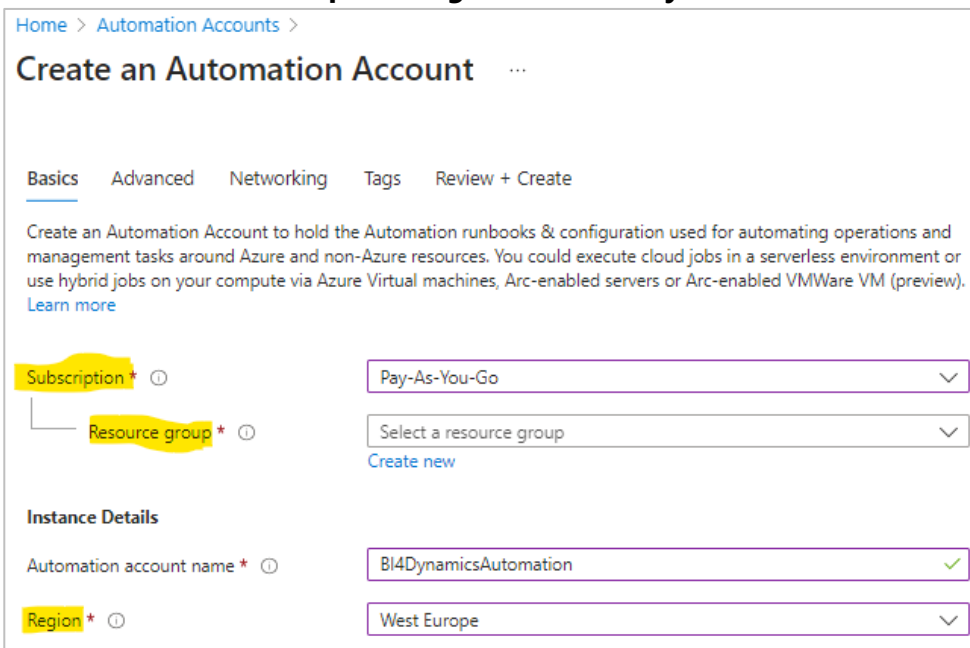


2.2 Instructions

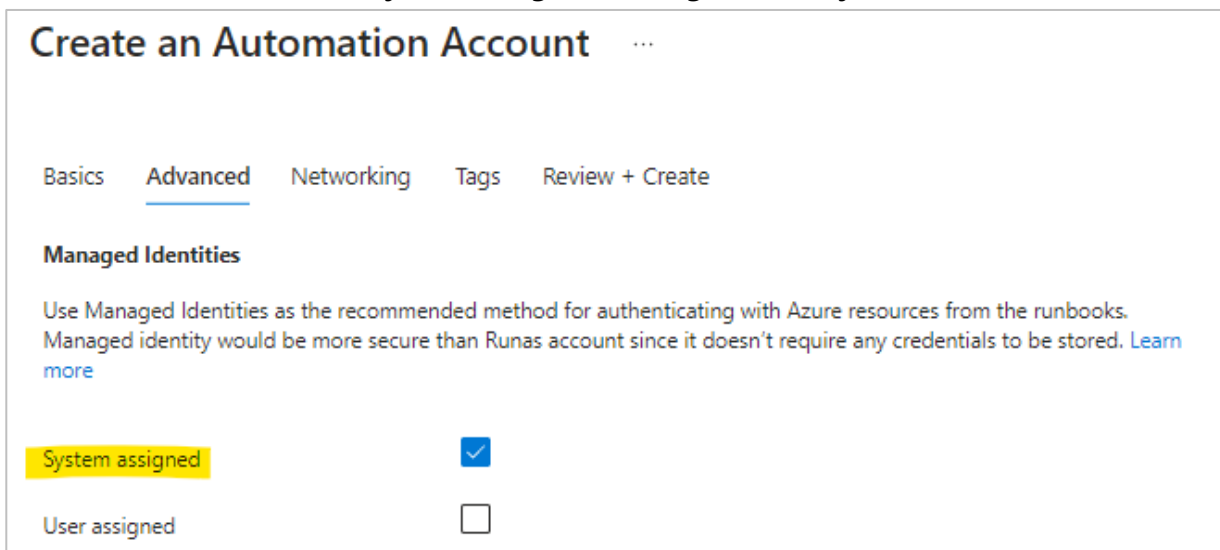
Go to the Azure Portal and search for **Automation Accounts**:



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**:



Create an Automation Account ...

Basics Advanced Networking Tags Review + Create

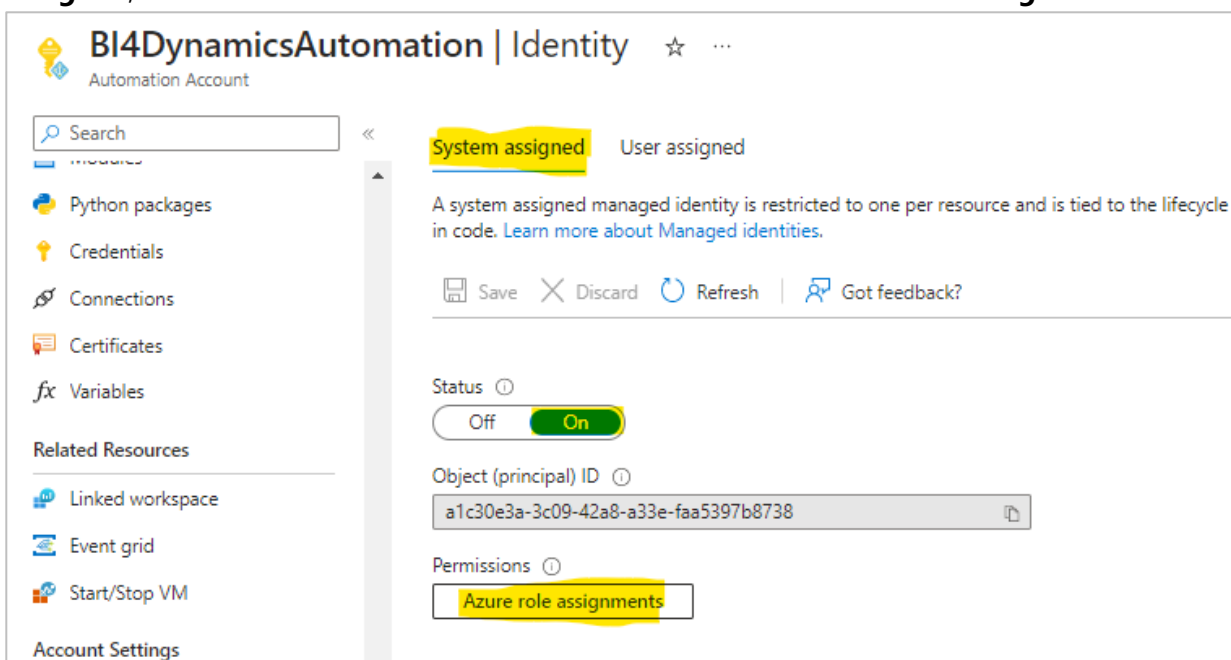
Managed Identities

Use Managed Identities as the recommended method for authenticating with Azure resources from the runbooks. Managed identity would be more secure than Runas account since it doesn't require any credentials to be stored. [Learn more](#)

System assigned

User assigned

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**:



BI4DynamicsAutomation | Identity ☆ ...

Automation Account

Search

Python packages

Credentials

Connections

Certificates

Variables

Related Resources

Linked workspace

Event grid

Start/Stop VM

Account Settings

System assigned User assigned

A system assigned managed identity is restricted to one per resource and is tied to the lifecycle in code. [Learn more about Managed identities.](#)

Save Discard Refresh Got feedback?

Status ⓘ

Off **On**

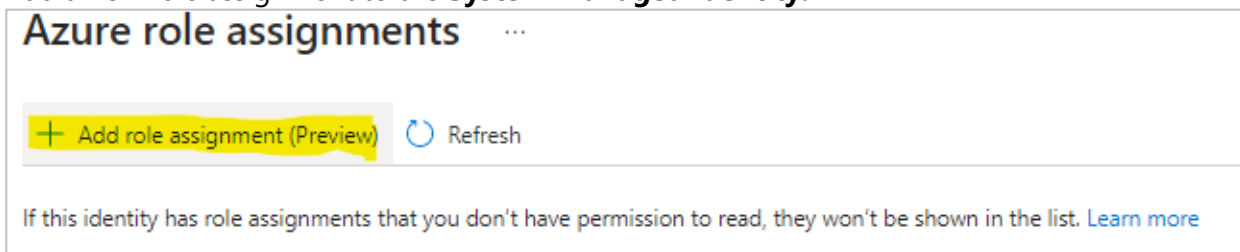
Object (principal) ID ⓘ

a1c30e3a-3c09-42a8-a33e-faa5397b8738

Permissions ⓘ

Azure role assignments

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



Azure role assignments ...

+ Add role assignment (Preview) Refresh

If this identity has role assignments that you don't have permission to read, they won't be shown in the list. [Learn more](#)

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**.

Add role assignment (Preview) [X]

Scope ⓘ
Resource group [v]

Subscription [v]

Resource group ⓘ
BI4Dynamics [v]

Role ⓘ
Contributor [v]

[Learn more about RBAC](#)

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**.

Create a runbook ...

Name * ⓘ Start_Stop_AAS [✓]

Runbook type * ⓘ PowerShell [v]

Runtime version * ⓘ 5.1 [v]

Description [v]

i During runbook execution, PowerShell modules targeting 5.1 runtime version will be used. Please make sure the required PowerShell modules are present in 5.1 runtime version.

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:

Edit PowerShell Runbook ...
Start_Stop_AAS

Save Publish Revert to published Test pane Feedback

> CMDLETS [1]

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.St
ate)"
    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** in which **Azure Analysis Services** are located, and **Managed Identity** was given the permissions to.
- Under the option **ANALYSSERVERNAME** insert the name of **Azure Analysis Services**.

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

The screenshot shows the 'Test Pane' interface. At the top, there are control buttons: 'Start' (highlighted in yellow), 'Stop', 'Suspend', 'Resume', 'View last test', and 'Refresh job streams'. Below the buttons is a 'Parameters' section with three input fields, each labeled with a parameter name and marked as mandatory: 'AASACTION', 'RESOURCEGROUPNAME', and 'ANALYSSERVERNAME'. Each field contains the text 'Enter a value'. To the right of the parameters is a black box with blue text that reads: 'Click 'Start' to begin the test run. Streams will display when the test completes.'

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

The screenshot shows the 'Edit PowerShell Runbook' interface. The title bar reads 'Edit PowerShell Runbook*' and 'Start_Stop_AAS'. Below the title bar are several buttons: 'Save' (highlighted in yellow), 'Publish' (highlighted in yellow), 'Revert to published', 'Test pane', and 'Feedback'. On the left side, there is a navigation pane with 'CMDLETS', 'RUNBOOKS', and 'ASSETS'. The main area displays PowerShell code for parameter validation:

```

1 # Parameters
2 [CmdletBinding()]
3 param(
4     [Parameter(Mandatory=$True,Position=0)]
5     [ValidateSet('Start','Stop')]
6     [string]$AasAction,
7 
```

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:

The screenshot shows the 'Schedule Runbook' interface. The title bar reads 'Schedule Runbook' and 'Start_Stop_AAS'. Below the title bar is a 'Schedule' section with the text 'Link a schedule to your runbook'. Below that is a 'Parameters and run settings' section with the text 'Configure parameters and run settings'.

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:

Name	Next run	Time zone	Status
Start	4/12/2023, 7:00 AM	Central European Time	✓ On
Stop	4/12/2023, 5:00 PM	Central European Time	✓ On

New Schedule ✕

Name *

Description

Starts *

Time zone

Recurrence

Recur every *

On these days Monday
 Tuesday
 Wednesday
 Thursday
 Friday
 Saturday
 Sunday

Set expiration

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

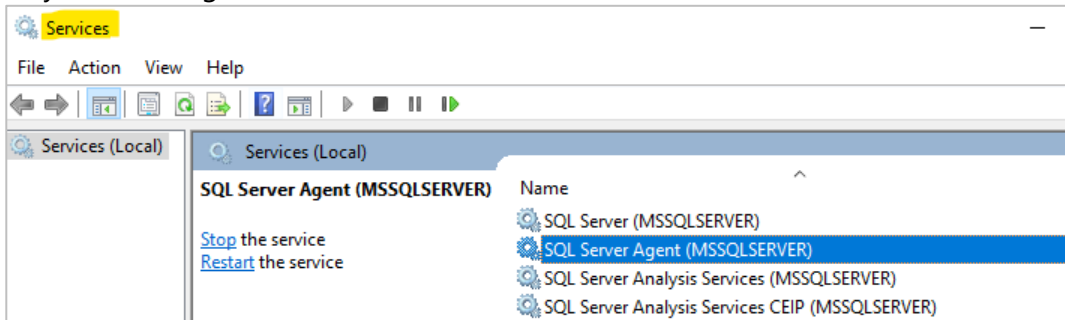
Runbook	Job created	Status	Ran on	Last status update
Start_Stop_AS	4/11/2023, 6:00:17 AM	✓ Completed	Azure	4/11/2023, 6:03:12 AM
Start_Stop_AS	4/10/2023, 6:00:17 PM	✓ Completed	Azure	4/10/2023, 6:01:52 PM
Start_Stop_AS	4/10/2023, 6:00:30 AM	✓ Completed	Azure	4/10/2023, 6:03:53 AM
Start_Stop_AS	4/9/2023, 6:00:21 PM	✓ Completed	Azure	4/9/2023, 6:03:31 PM
Start_Stop_AS	4/9/2023, 6:00:20 AM	✓ Completed	Azure	4/9/2023, 6:04:04 AM

3 SQL Server Agent: Start Job

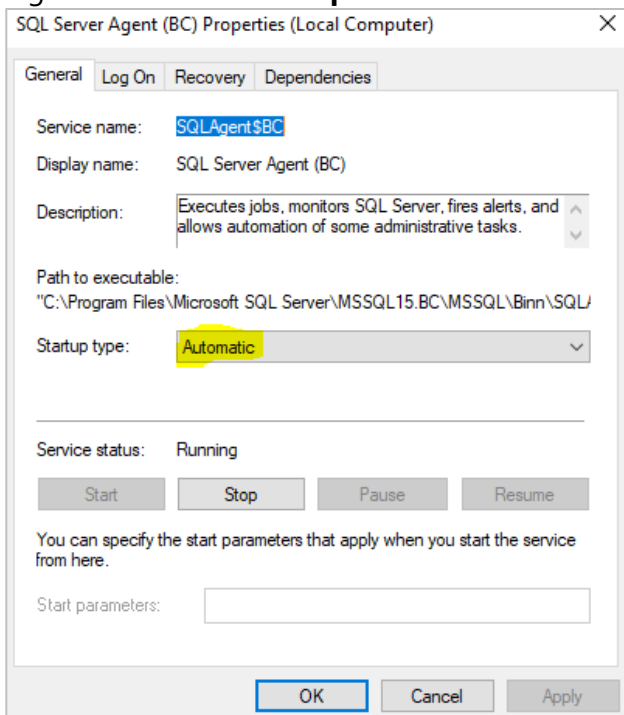
When Virtual machine is running, it is ready to process data. This process is triggered by SQL Server Agent feature, a part of SQL server.

3.1 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.



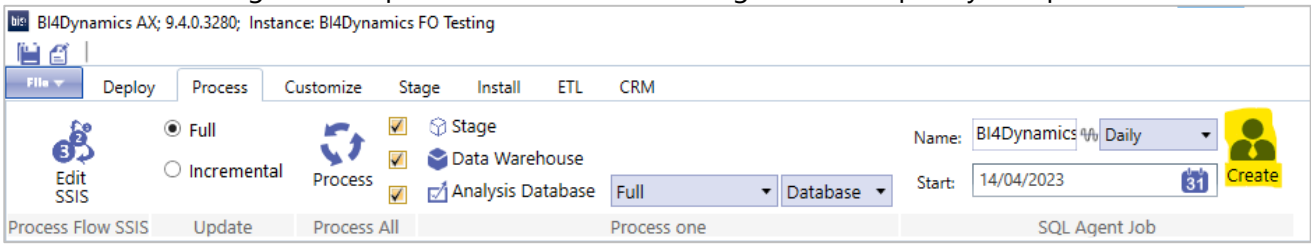
Right click and select **Properties** and set Start-up Type to Automatic.



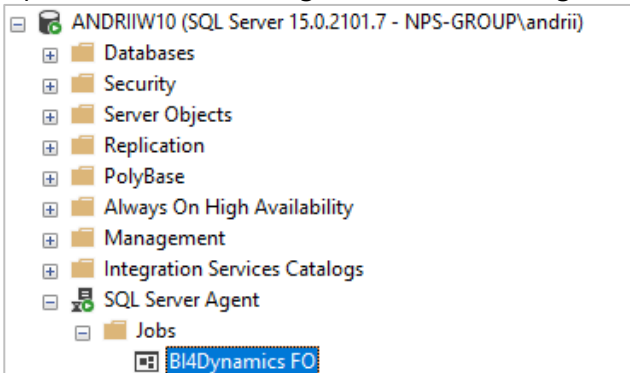
Note: make sure that user running service is a domain admin user (not a service) and has permissions needed to process data warehouse and analysis services. On VM this would be the VM admin user.

3.2 Setup SQL Server Agent

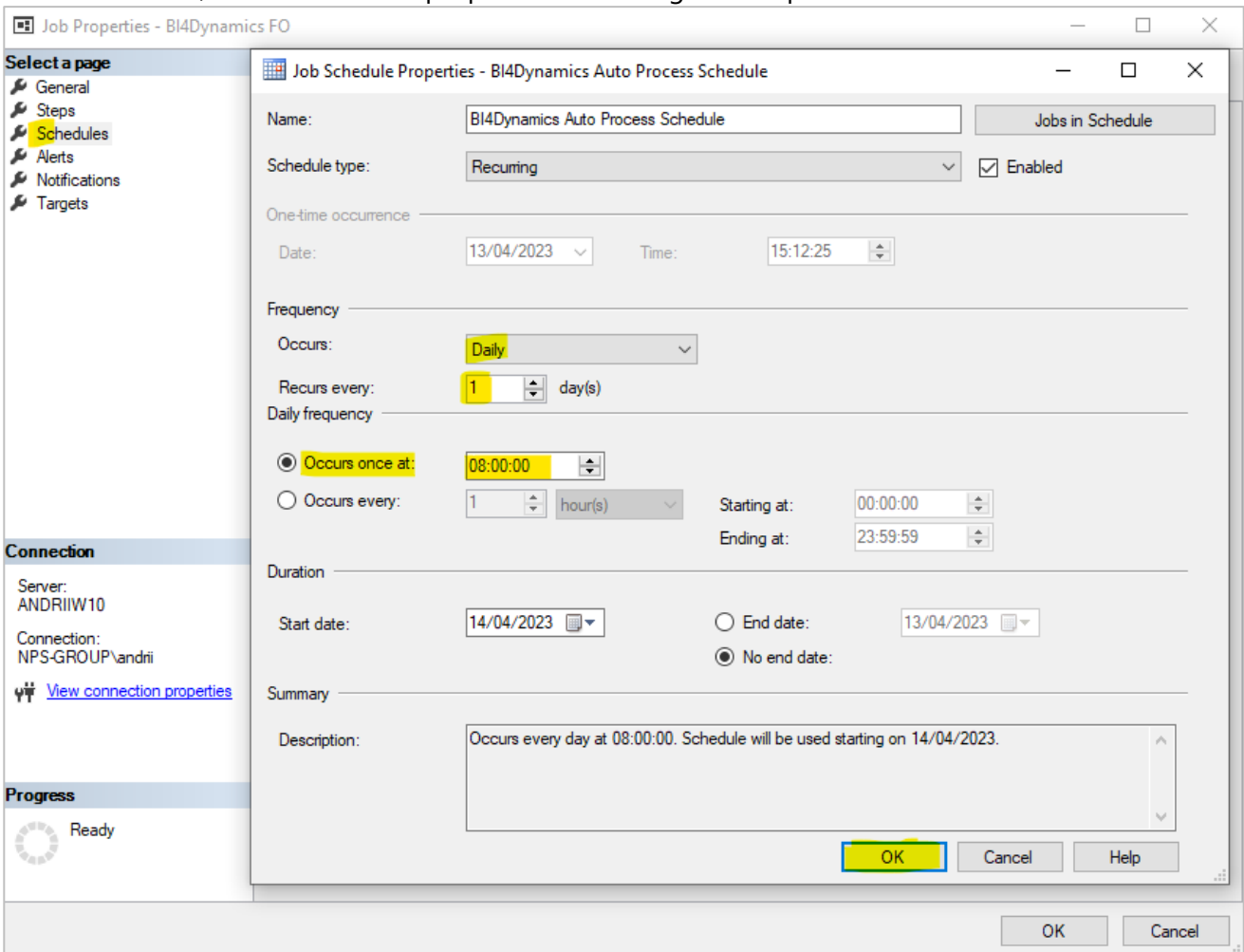
SQL Server agent conducts processing of stage, data warehouse and analytics, bringing new data to users. To set SQL Agent Job open **Process** tab, set SQL Agent Job frequency and press **Create**:



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**.



4 Timing examples

Here is an example of processing schedule for daily update:

Step	Step description	Start Time	Duration	Comment
1	Start Azure Virtual Machine	07:00	2-3 min	VM hosts data warehouse that must be ready when DW processing starts
2	Start Azure Analysis Services	07:00	2-3 min	Azure AS must be ready when DW processing start
3	Start SQL Server Agent	07:10	60 min	DW processing (data are in Azure AS)
4	Stop Azure Virtual Machine	08:30		Leave some buffer time after DW is processed and then stop VM.
5	Stop Azure Analysis Services	17:00		AAS will run during business hours when users are querying data.