

Lifting & Shifting/Modernizing your ETL Workflows with SSIS in ADFv2

Sandy Winarko

Principal Program Manager, IM China

Sections

- Microsoft ETL/ELT Services in Azure
- ADF Concepts
- Enterprise Edition
- Custom Setup Interface
- Provisioning Methods
- Deployment Methods
- Execution Methods
- Scheduling Methods
- Monitoring Methods
- On-Demand/Just-In-Time Provisioning
- Resources

Microsoft ETL/ELT Services in Azure

Customer Insights

- SSIS is a traditional ETL tool that comes bundled with SQL Server on premises
 - Has been around for more than 10 years
 - Some customers have started to lift & shift their ETL workloads to the cloud to reduce their on-prem infra, but found managing Infrastructure as a Service (IaaS)/VMs challenging

Customer Insights

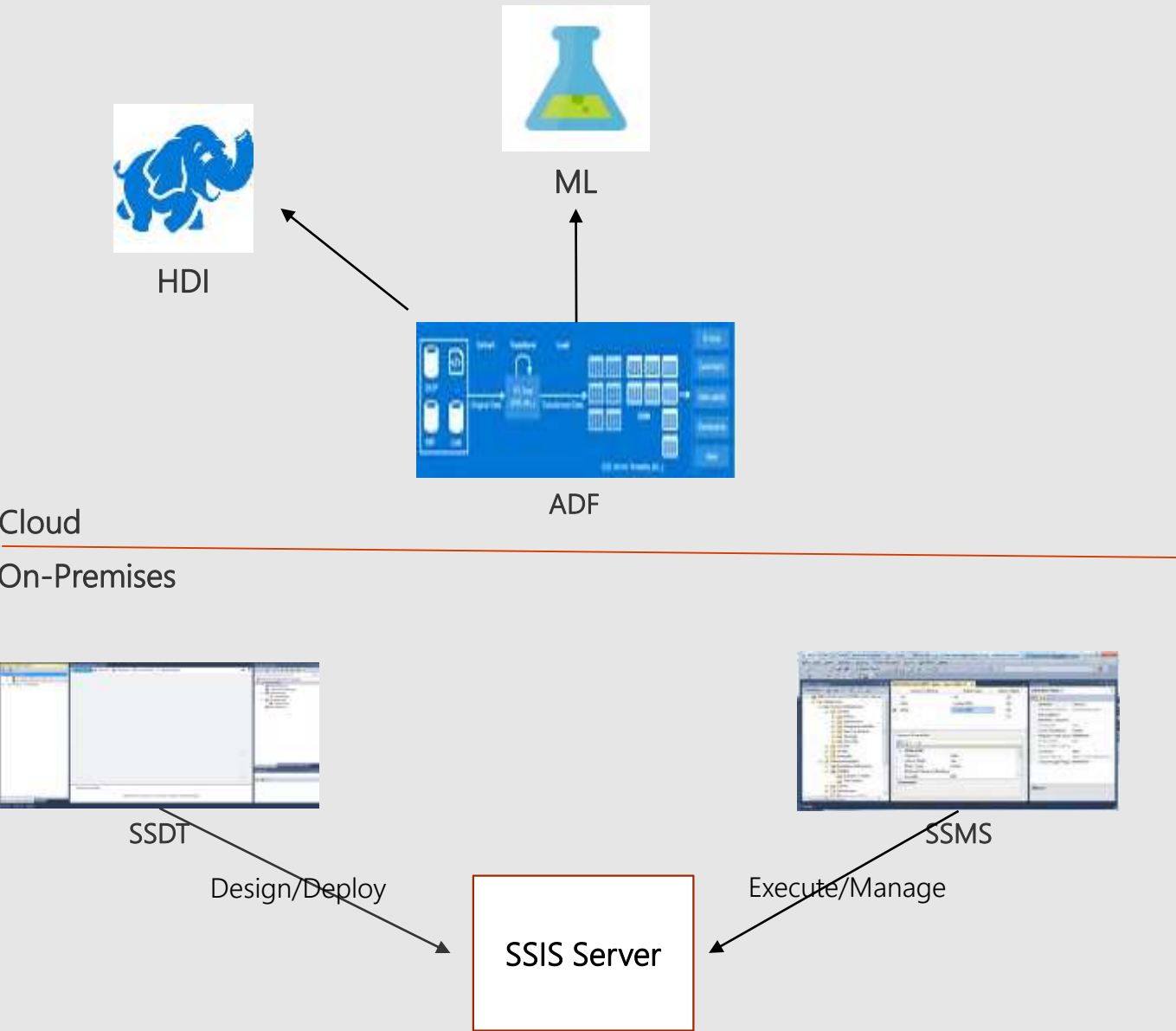
- SSIS is a traditional ETL tool that comes bundled with SQL Server on premises
 - Has been around for more than 10 years
 - Some customers have started to lift & shift their ETL workloads to the cloud to reduce their on-prem infra, but found managing Infrastructure as a Service (IaaS)/VMs challenging
- Azure Data Factory (ADF) is a modern ELT tool that moves/copies data and dispatches transformations for Big Data Analytics in the cloud
 - Some gaps in ELT workflows can be filled w/ code-free authoring of transformations/built-in tasks from SSIS
 - Some customers have started to combine ADF with SSIS on IaaS/VMs, but found managing IaaS/VMs challenging

Customer Insights

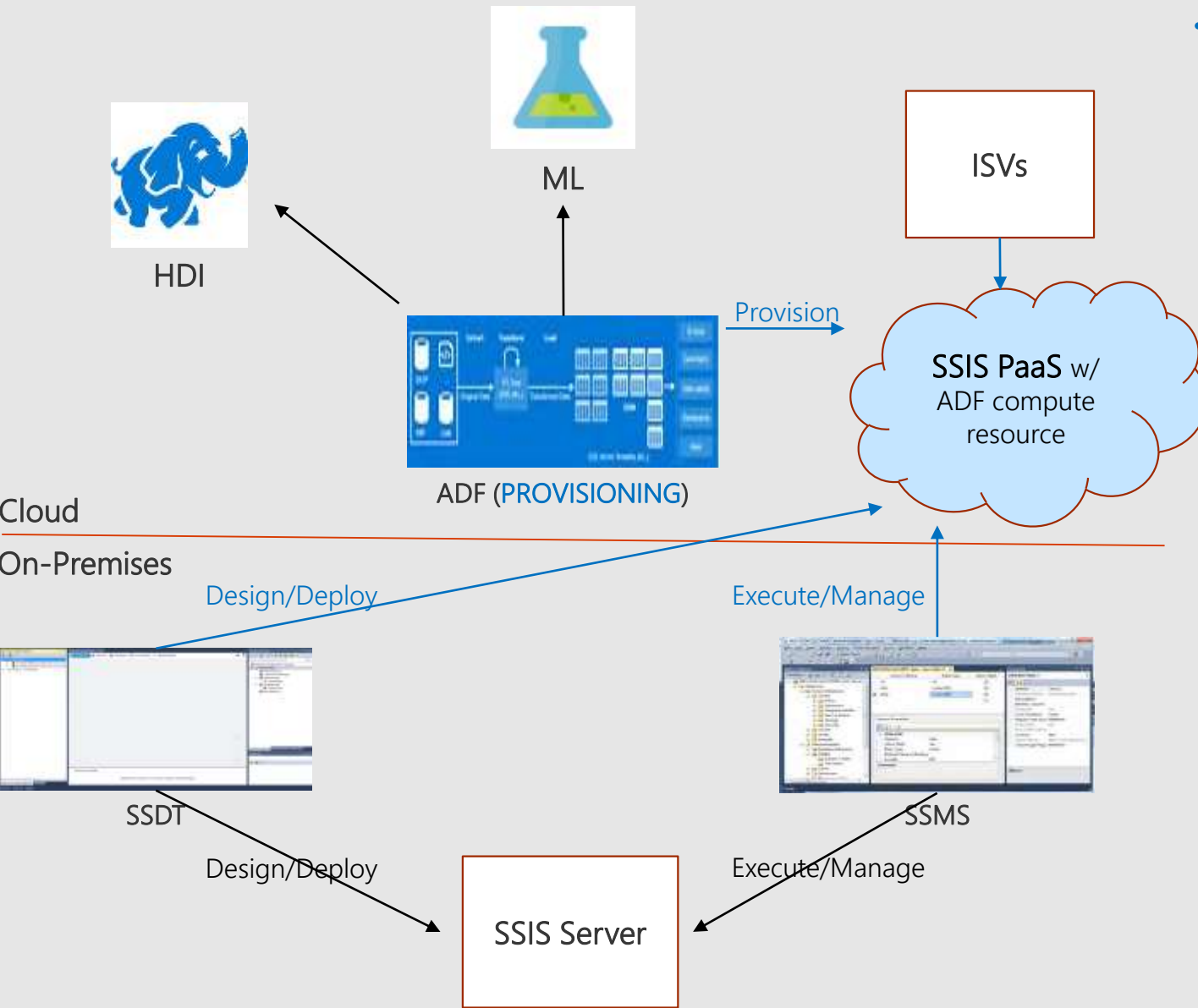
- SSIS is a traditional ETL tool that comes bundled with SQL Server on premises
 - Has been around for more than 10 years
 - Some customers have started to lift & shift their ETL workloads to the cloud to reduce their on-prem infra, but found managing Infrastructure as a Service (IaaS)/VMs challenging
- Azure Data Factory (ADF) is a modern ELT tool that moves/copies data and dispatches transformations for Big Data Analytics in the cloud
 - Some gaps in ELT workflows can be filled w/ code-free authoring of transformations/built-in tasks from SSIS
 - Some customers have started to combine ADF with SSIS on IaaS/VMs, but found managing IaaS/VMs challenging
- Evolution of a cloud-first product: SSIS on premises -> IaaS -> PaaS
 - The stage is set for SSIS PaaS...

Microsoft ETL/ELT Services in Azure

- We are building SSIS PaaS in ADFv2 phase by phase to offer a unified platform for Microsoft ETL/ELT services in the cloud

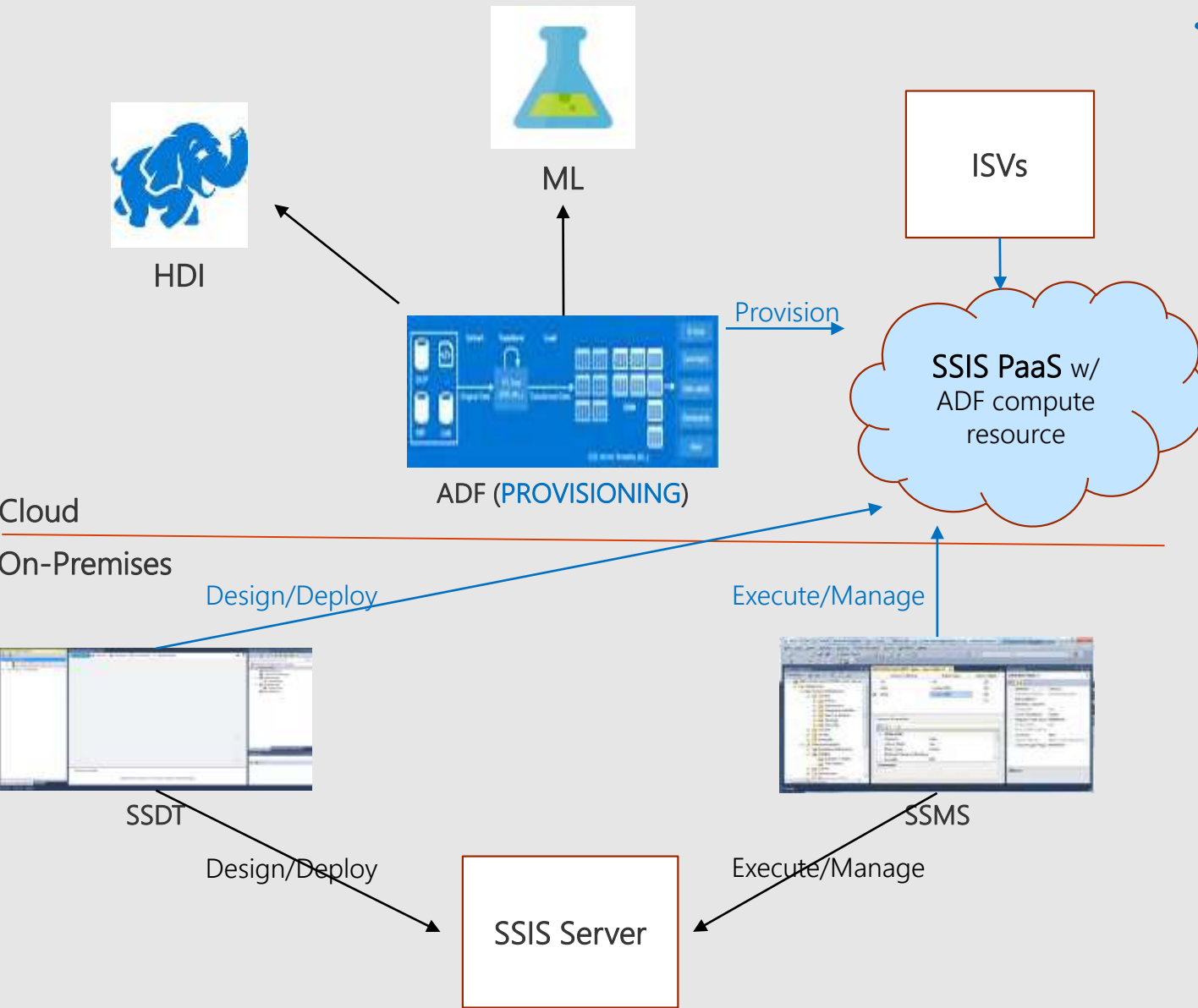


Microsoft ETL/ELT Services in Azure



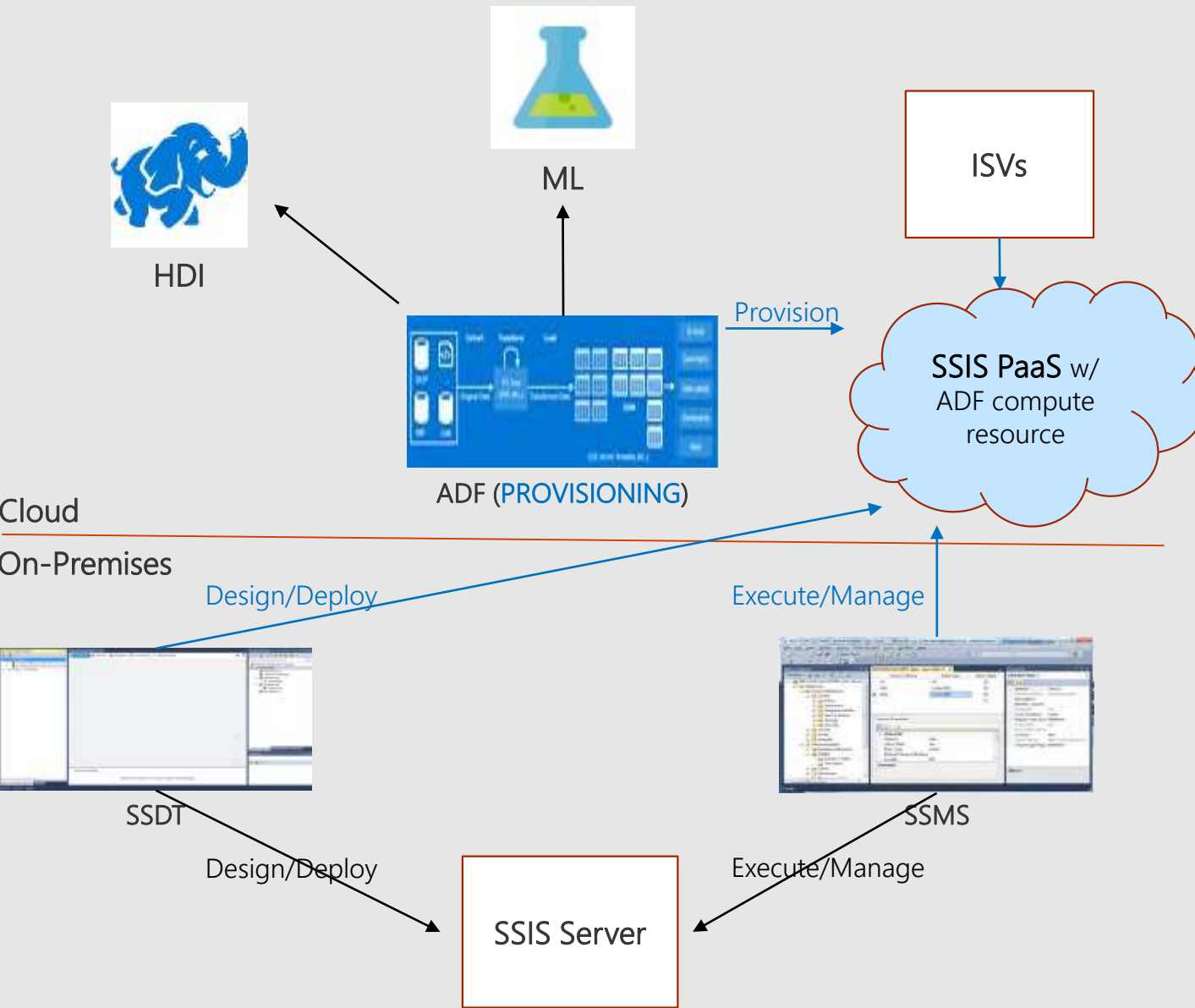
- **Phase 1:** Use ADFv2 to provision SSIS PaaS – Launched at MS Ignite (Sep'17)
 - Create ADFv2, if you have not done so already
 - Use ADFv2 App/SDK/API/PSH to provision SSIS PaaS w/ ADF compute resource called Azure-SSIS Integration Runtime (IR)
- Still use SQL Server Data Tools (SSDT) to design/deploy SSIS packages
- Still use SQL Server Management Studio (SSMS) to execute/manage SSIS packages
- Serve SSIS customers who want to move all/part of their on-premises workloads and just "lift & shift" many existing packages to Azure
- Independent Software Vendors (ISVs) can build extensions/Software as a Service (SaaS) on SSIS PaaS

Microsoft ETL/ELT Services in Azure



- Introducing Azure-SSIS IR: Managed cluster of Azure VMs (nodes) dedicated to run your SSIS packages and no other activities
 - You can scale it up/out by specifying the node size / number of nodes in the cluster
- You can bring your own Azure SQL Database (DB)/Managed Instance (MI) server to host the catalog of SSIS projects/packages (**SSISDB**) that will be attached to it
- You can join it to a Virtual Network (VNet) that is connected to your on-prem network to enable on-prem data access
- Once provisioned, you can enter your Azure SQL DB/MI server endpoint on SSDT/SSMS to deploy SSIS projects/packages and configure/execute them just like using SSIS on premises

Microsoft ETL/ELT Services in Azure



• Customer cohorts for Phase 1:

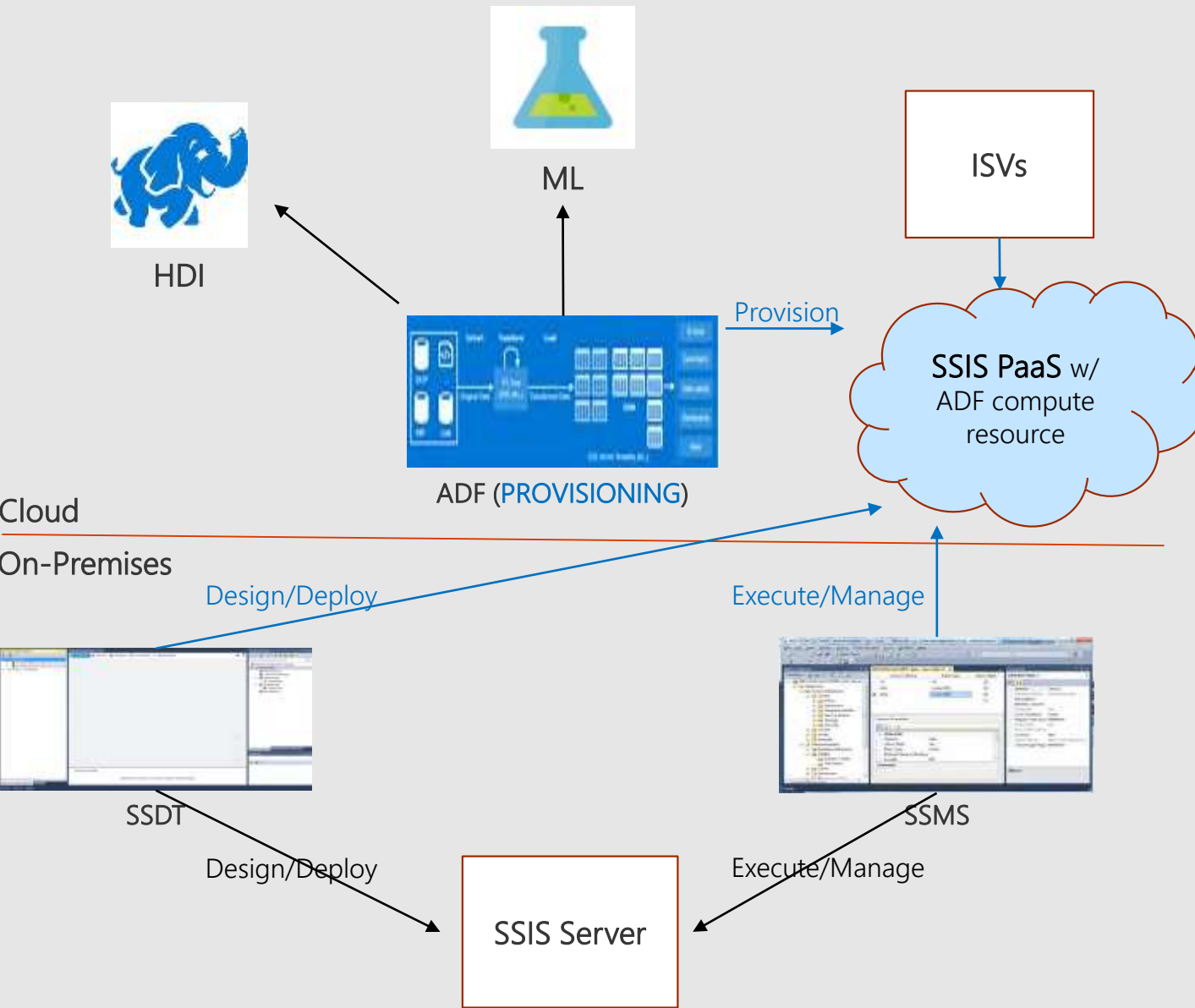
1. "SQL Migrators"

These are SSIS customers who want to retire their on-prem SQL Servers and migrate all apps + data ("complete/full lift & shift") into Azure SQL MI – For them, SSISDB can be hosted by Azure SQL MI inside their VNet

2. "ETL Cost Cutters"

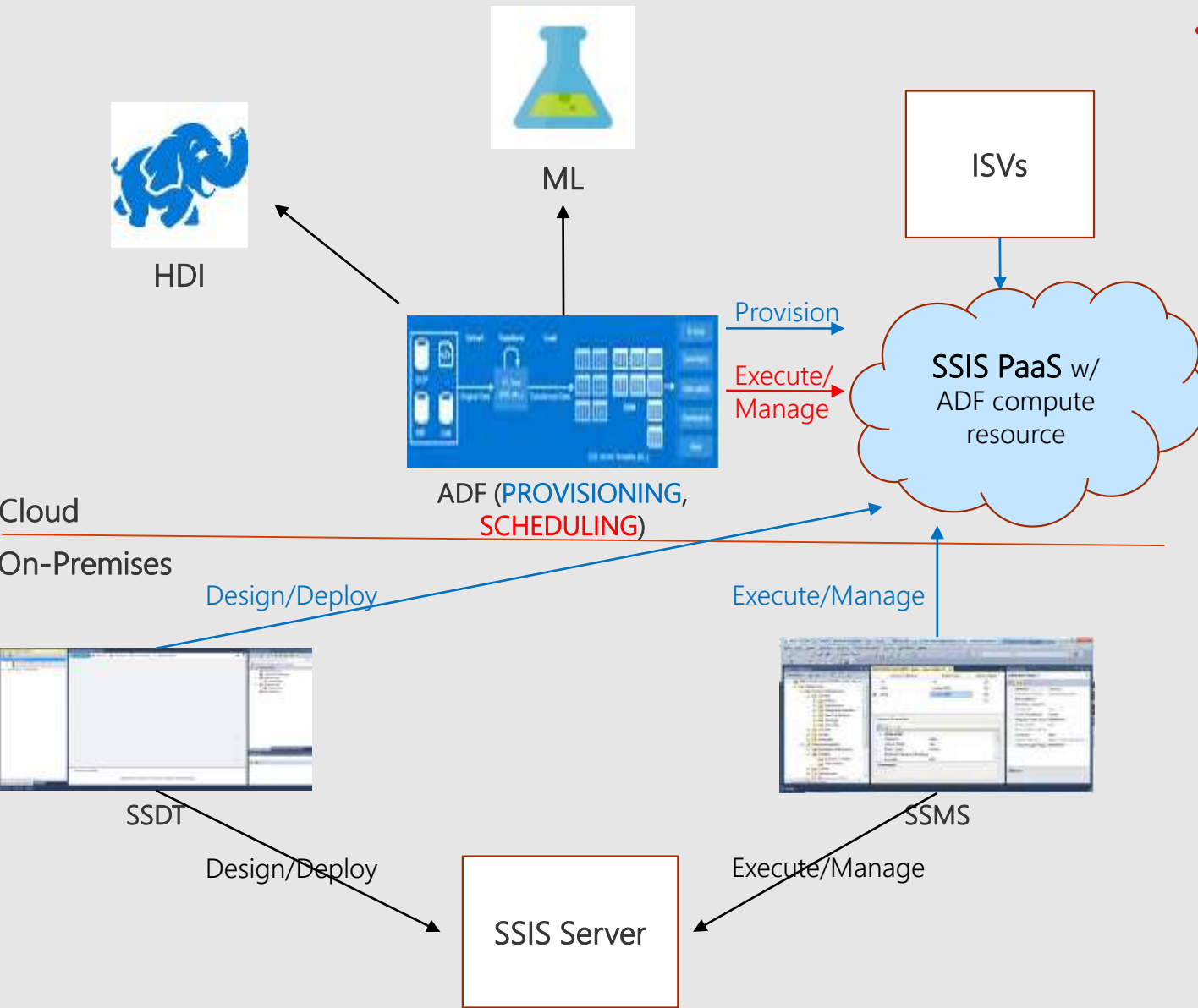
These are SSIS customers who want to lower their operational costs and gain High Availability (HA)/scalability for just their ETL workloads w/o managing their own infra ("partial lift & shift") – For them, SSISDB can be hosted by Azure SQL DB in the public network

Microsoft ETL/ELT Services in Azure



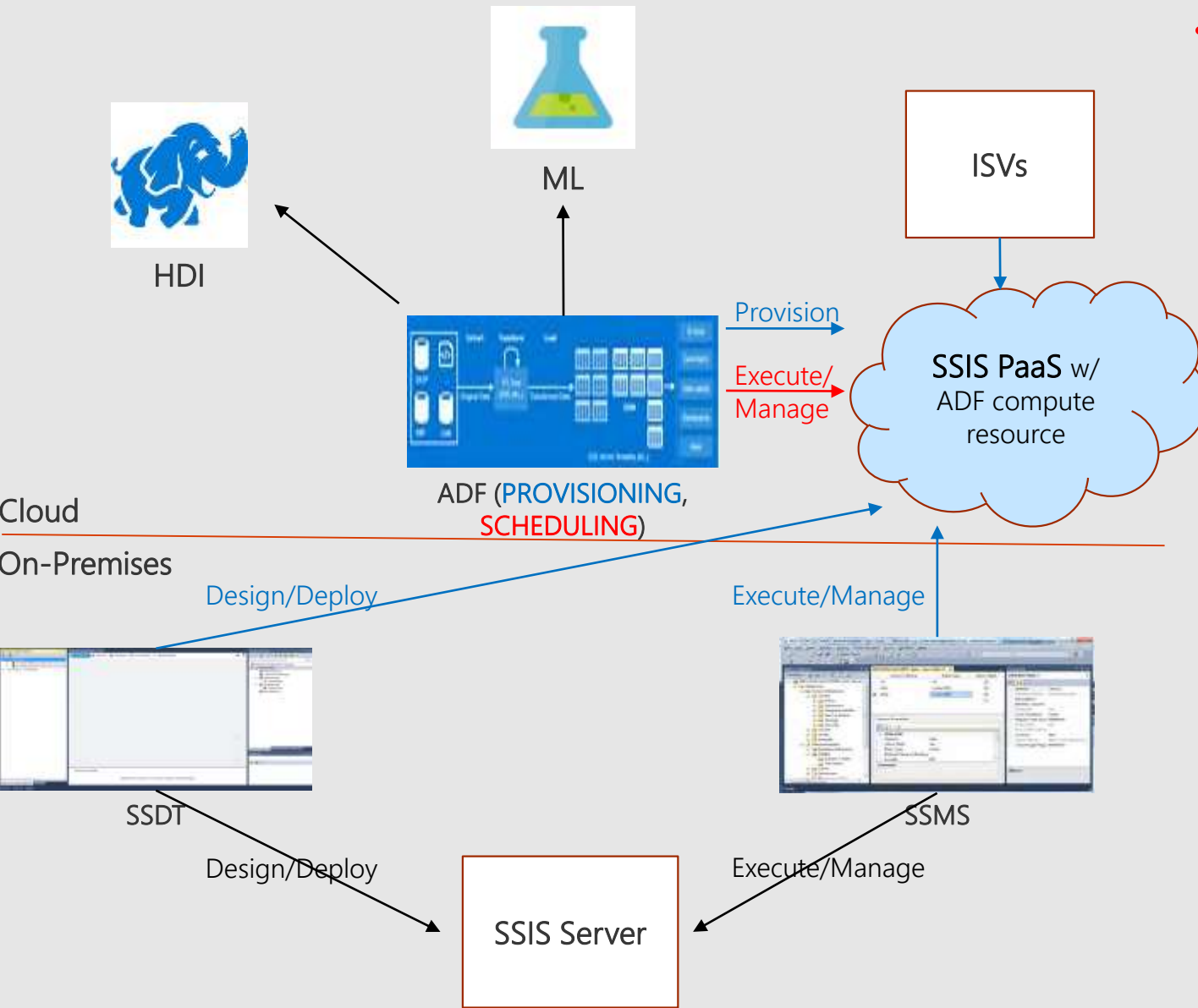
- LIVE now in Public Preview
 - Six locations: East US/East US 2/Central US/North Europe/West Europe/Australia East
 - Six node sizes: Av2/Dv2 series VMs
 - Classic VNet support
 - Standard edition/license
 - 24/7 live-site support
- **What's new**
 - Azure Resource Manager (ARM) VNet support
 - Enterprise edition/license (Private Preview)
 - Custom setup interface (Private Preview)
 - ADFv2 App/GUI web tool
- **Coming soon**
 - More locations, e.g. West US, UK South
 - More node sizes, e.g. Dv3/Ev3 series VMs
 - Azure Hybrid Use Benefit (AHUB)/Bring Your Own License (BYOL) support

Microsoft ETL/ELT Services in Azure



- **Phase 2:** Use ADFv2 to execute/manage SSIS packages deployed to SSIS PaaS as first-class activities – ETA Q2CY18 (TBD)
- Chain/group them with Azure HDInsight (HDI)/Machine Learning (ML)/other activities inside data pipelines
- Serve SSIS/ADF customers who want to combine ETL/ELT workflows on a single platform

Microsoft ETL/ELT Services in Azure



- Customer cohorts for Phase 2:

- 3. "ETL Modernizers"

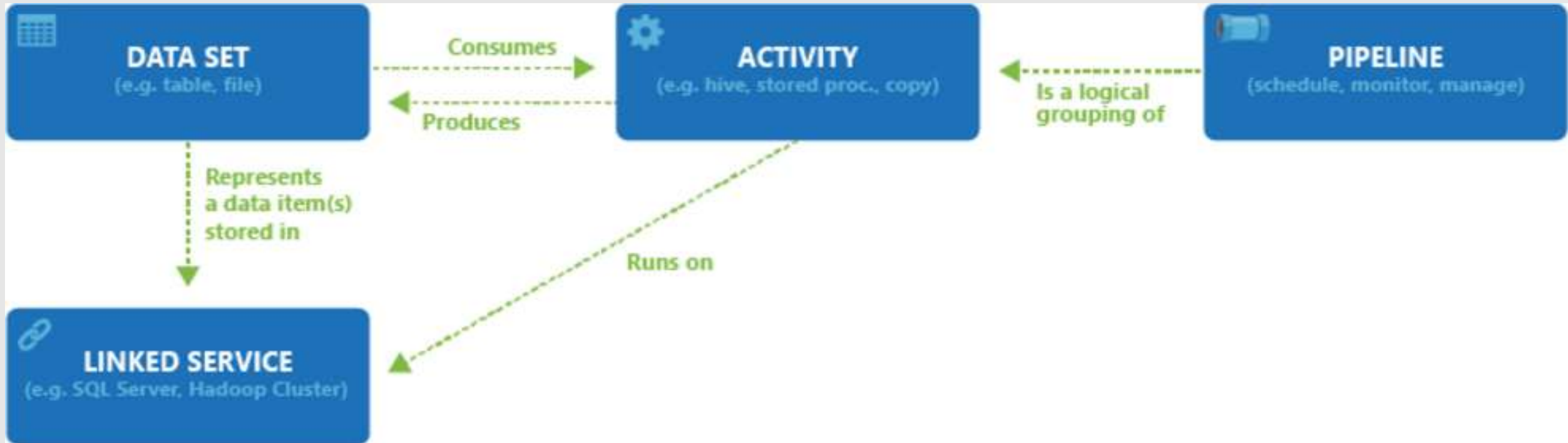
These are SSIS customers who want to modernize their workflows and explore Big Data Analytics in the cloud

- 4. "ELT Gap Fillers"

These are ADF customers who want to fill some gaps in their workflows w/ code-free authoring of transformations/built-in tasks from SSIS

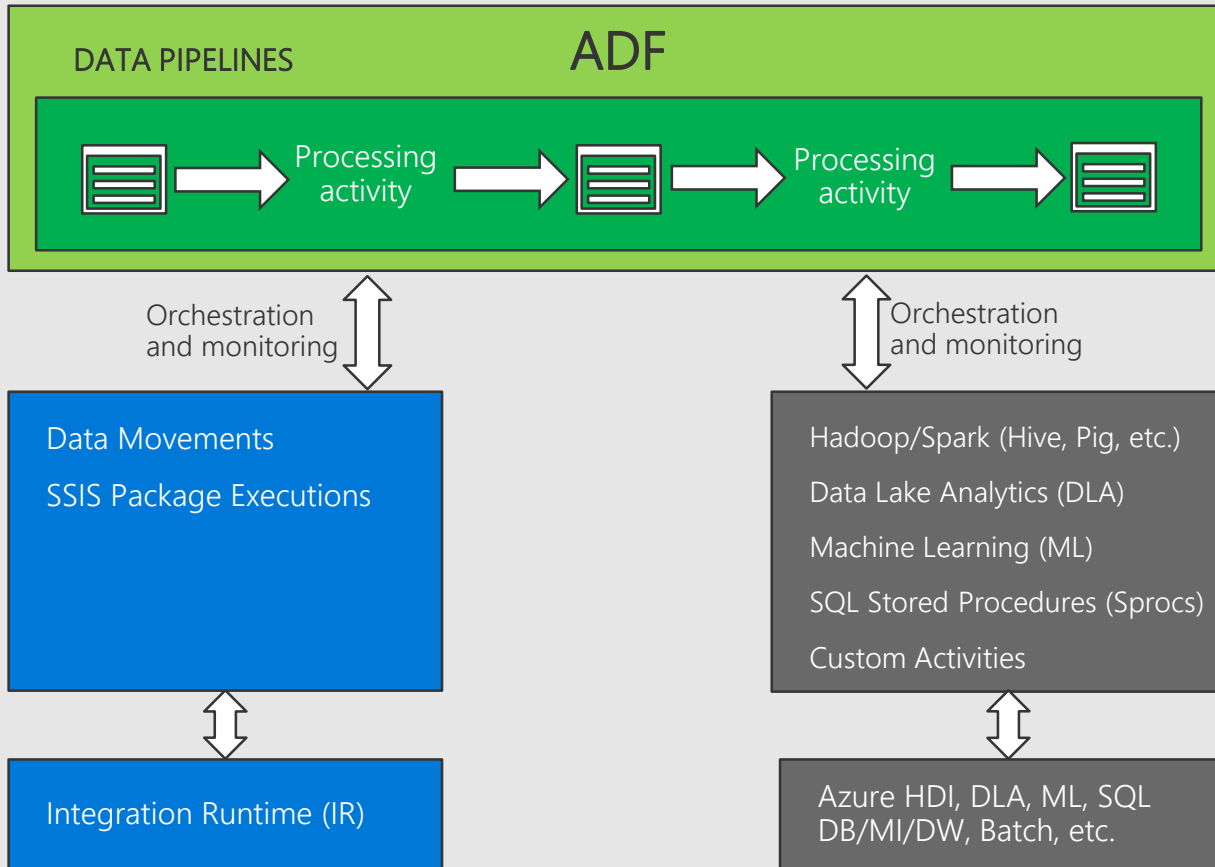
ADF Concepts

ADF Basic Concepts



- ADF is Microsoft's unified platform for ETL/ELT services in the cloud
- ADF allows you to build data pipelines and trigger/schedule their runs
- Data pipeline is a chain/group of activities to be performed on your data, e.g. data movements/transformations
- Activities take data sets, which are named references/pointers to data, as inputs/outputs
- Some activities target data store/compute resources allocated outside ADF, e.g. ADLS/HDI/ML/etc.
- Linked services represent those resources and provide the connection info for ADF to orchestrate activities targeting them

ADFv2 Integration Runtimes



- Integration Runtime (IR) is ADF compute resource that can perform data movements/transformations, including SSIS package executions
- Customers can deploy one/many instances of IR per ADF as required to run pipelines/process data
- IR can run in the public network, inside VNet, or behind corporate firewalls
- SSIS PaaS runs on Azure-SSIS IR, internal/native to ADF that provisions it, while SSISDB is hosted by your own Azure SQL DB/MI server, external to ADF
- Consequently, Azure-SSIS IR is billed under your ADF subscription, separately from SSISDB that is billed under your Azure SQL DB/MI subscription

Enterprise Edition

Enterprise Edition – Introduction

- Enterprise Edition of Azure-SSIS IR allows you to use advanced/premium features:
 - Change Data Capture (CDC) components
 - Oracle/Teradata/SAP BW connectors
 - SQL Server Analysis Services (SSAS)/Azure Analysis Services (AAS) connectors/transforms
 - Fuzzy Grouping/Lookup transforms
 - Term Extraction/Lookup transforms
- Some of these features will also require you to install additional components, essentially customizing your Azure-SSIS IR (via Custom Setup Interface)

Enterprise Edition – Features

Enterprise Features	Descriptions
CDC components	<ul style="list-style-type: none">• <u>CDC Source/Splitter/Control</u> Task are preinstalled on your Azure-SSIS IR Enterprise Edition• To connect to Oracle, you will also need to install <u>CDC Designer/Service</u> on another machine
Oracle connectors	<ul style="list-style-type: none">• <u>Oracle Connection Manager/Source/Destination</u> are preinstalled on your Azure-SSIS IR Enterprise Edition• You will also need to install Oracle Call Interface (OCI) driver, and if necessary configure Oracle Transport Network Substrate (TNS), on your Azure-SSIS IR (via <u>Custom Setup Interface</u>)
Teradata connectors	<ul style="list-style-type: none">• You will need to install <u>Teradata Connection Manager/Source/Destination</u> and Teradata Parallel Transporter (TPT) API + Teradata ODBC driver on your Azure-SSIS IR Enterprise Edition (via <u>Custom Setup Interface</u>)

Enterprise Edition – Features

Enterprise Features	Descriptions
SAP BW connectors	<ul style="list-style-type: none">• <u>SAP BW Connection Manager/Source/Destination</u> are preinstalled on your Azure-SSIS IR Enterprise Edition• You will also need to install SAP BW driver on your Azure-SSIS IR (via <u>Custom Setup Interface</u>)• These connectors support <u>SAP BW 7.0</u> or earlier versions• To connect to later versions, you can install SAP connectors from our partners (e.g. Theobald Software) on your Azure-SSIS IR (via <u>Custom Setup Interface</u>)

Enterprise Edition – Features

Enterprise Features	Descriptions
SSAS/AAS components	<ul style="list-style-type: none">• <u>Data Mining Model Training/Dimension Processing/Partition Processing Destinations and Data Mining Query Transform</u> are preinstalled on your Azure-SSIS IR Enterprise Edition• All support SSAS, but <u>only Partition Processing Destination supports AAS</u>• To connect to SSAS, you will also need to configure Windows Authentication credentials in your SSISDB• On top of these components, <u>Analysis Services Execute DDL/Processing and Data Mining Query Tasks</u> are also preinstalled on your Azure-SSIS IR Standard/Enterprise Edition
Fuzzy Grouping/Lookup Transforms	<ul style="list-style-type: none">• They are preinstalled on your Azure-SSIS IR Enterprise Edition and support <u>SQL Server/Azure SQL DB</u> for storing reference data
Term Extraction/Lookup Transforms	<ul style="list-style-type: none">• They are preinstalled on your Azure-SSIS IR Enterprise Edition and support <u>SQL Server/Azure SQL DB</u> for storing reference data

Enterprise Edition – Instructions

- Please send us your Azure subscription ID that you will use for provisioning/reconfiguring your Azure-SSIS IR with this feature – We will whitelist it for preview
- Please download and install our [private version of Azure PSH](#)
- When provisioning/reconfiguring your Azure-SSIS IR via PSH, execute Set-AzureRmDataFactoryV2IntegrationRuntime cmdlet with "Enterprise" as the value for new Edition parameter before starting your Azure-SSIS IR, e.g.

```
$MyAzureSsisIrEdition = "Enterprise"
```

```
Set-AzureRmDataFactoryV2IntegrationRuntime -DataFactoryName $MyDataFactoryName  
-Name $MyAzureSsisIrName -ResourceGroupName $MyResourceGroupName  
-Edition $MyAzureSsisIrEdition
```

```
Start-AzureRmDataFactoryV2IntegrationRuntime -DataFactoryName $MyDataFactoryName  
-Name $MyAzureSsisIrName -ResourceGroupName $MyResourceGroupName
```

Azure-SSIS IR Pricing

Node Size	Cores (CPU)	Memory (RAM)	Temp. Storage	Preview Prices	
				Standard	Enterprise
Standard_A4_v2	4	8.00 GiB	40 GiB	\$0.420/hour	\$0.956/hour
Standard_A8_v2	8	16.00 GiB	80 GiB	\$0.862/hour	\$1.935/hour
Standard_D1_v2	1	3.50 GiB	50 GiB	\$0.296/hour	\$0.832/hour
Standard_D2_v2	2	7.00 GiB	100 GiB	\$0.397/hour	\$0.933/hour
Standard_D3_v2	4	14.00 GiB	200 GiB	\$0.599/hour	\$1.136/hour
Standard_D4_v2	8	28.00 GiB	400 GiB	\$1.199/hour	\$2.271/hour

Custom Setup Interface

Custom Setup Interface – Introduction

- Custom Setup Interface allows you to alter the default operating configuration/environment (e.g. to start additional Windows services) and or install additional components (e.g. assemblies/drivers/extensions) on each node of your Azure-SSIS IR
- In general, it provides an interface to add your own setup steps during the provisioning/reconfiguration of your Azure-SSIS IR
- You can specify your custom setup by preparing a script + associated files, uploading them into a blob container in your Azure Storage account, and providing Shared Access Signature (SAS) Uniform Resource Identifier (URI) of your container when you provision/reconfigure your Azure-SSIS IR
- Each node of your Azure-SSIS IR will then download the script + files from your container and execute your custom setup with an elevated privilege
- Upon completion, each node will upload the standard output of execution and other logs into your container

Custom Setup Interface – Limitations

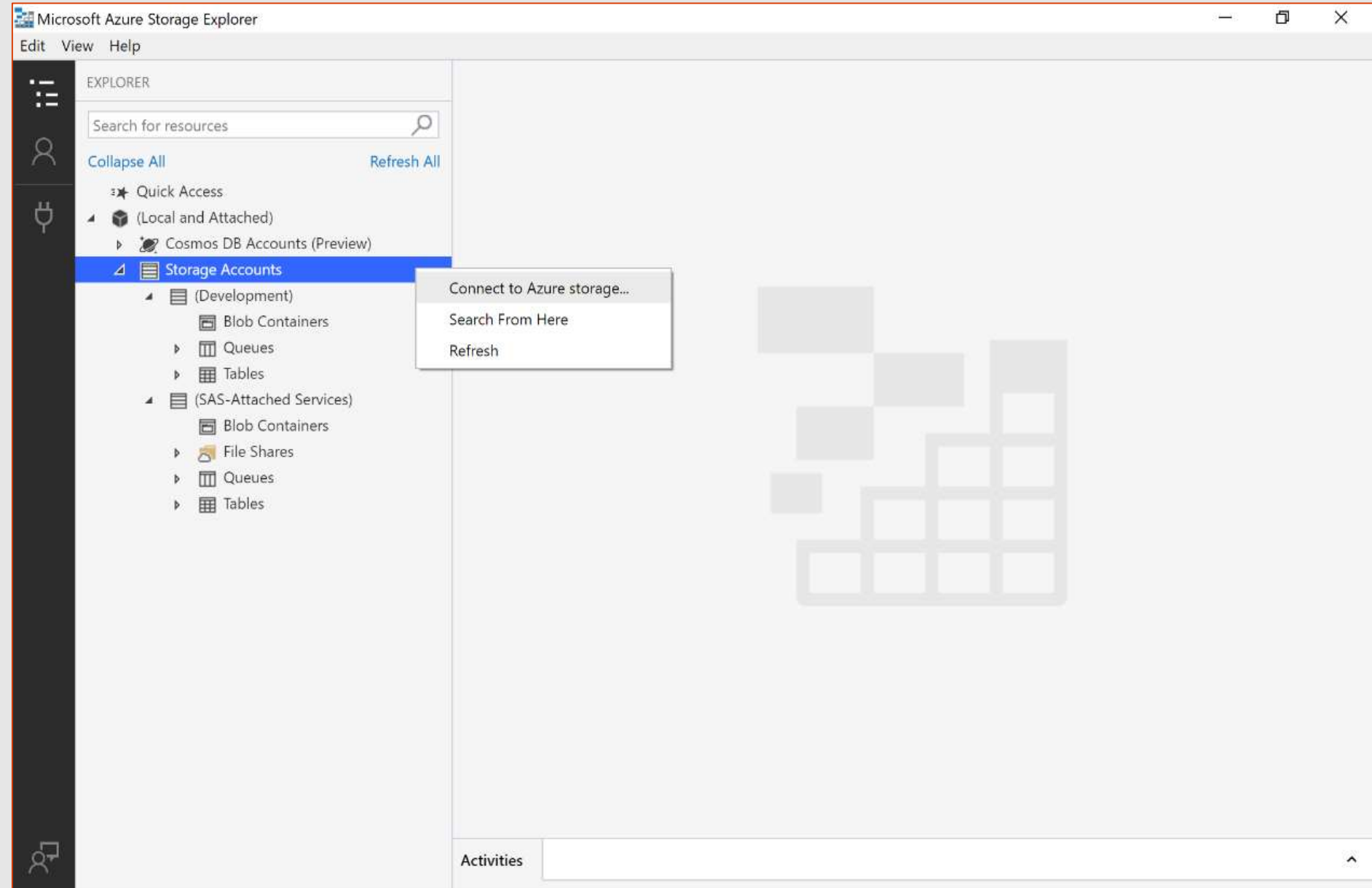
- We only support this feature in East US/North Europe/West Europe regions for now
- We only support the installation of free/unlicensed/Enterprise Edition components for now
 - We are in discussion with various Independent Software Vendor (ISV) partners to support the installation of premium/licensed components
- We do not support scripts invoking xcopy/robocopy tools directly to copy files for now
 - Please use `start /wait cmd /c "call install.cmd > %CUSTOM_SETUP_SCRIPT_LOG_DIR%\install.cmd.log"` as a workaround where install.cmd contains scripts invoking xcopy/robocopy tool
- We do not support scripts invoking gacutil tool to install assemblies in Global Assembly Cache (GAC) for now
 - Please use `gacinstall.cmd` provided in our Private Preview container instead

Custom Setup Interface – Instructions

- Please send us your Azure subscription ID that you will use for provisioning/reconfiguring your Azure-SSIS IR with this feature – We will whitelist it for preview
- Please download and install our [private version of Azure PSH](#)
- Please prepare your custom setup script and associated files (e.g. .bat/.cmd/.exe/.dll/.msi/.ps1 files)
 - You must have a script file named `"main.cmd"` that will be the entry point of your custom setup (see [Limitations](#) when preparing scripts)
 - If you want additional logs generated by some tools (e.g. msixec) to be also uploaded into your container, please specify our predefined environment variable, `CUSTOM_SETUP_SCRIPT_LOG_DIR`, as the log folder in your scripts (e.g. `msiexec /i xxx.msi /quiet /lv %CUSTOM_SETUP_SCRIPT_LOG_DIR%\install.log`)
- Download, install, and launch [Azure Storage Explorer app](#), if you have not done it already

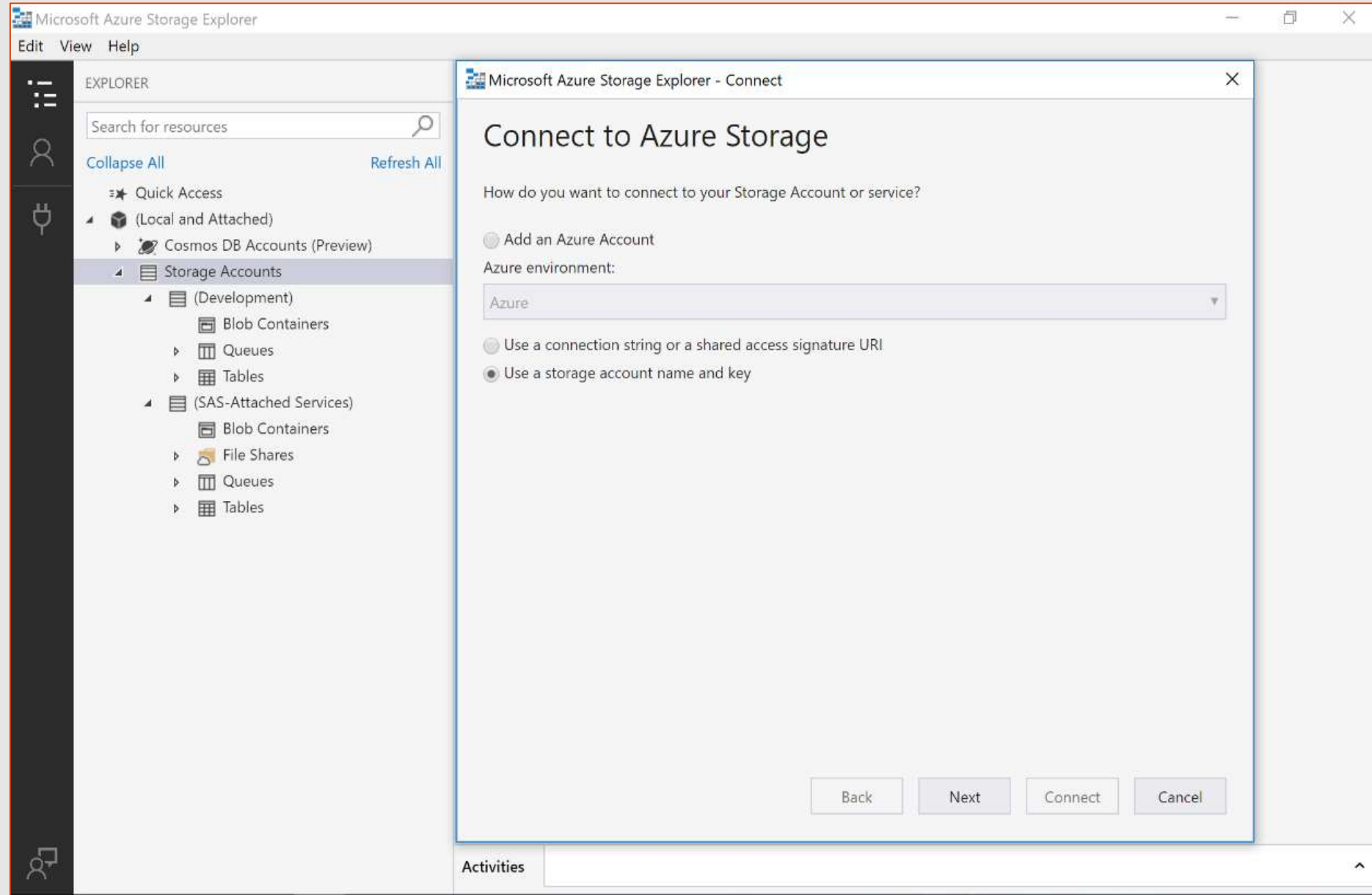
Custom Setup Interface – Instructions

- Under “(Local and Attached)” menu item, right-click “Storage Accounts”, and select “Connect to Azure storage...”



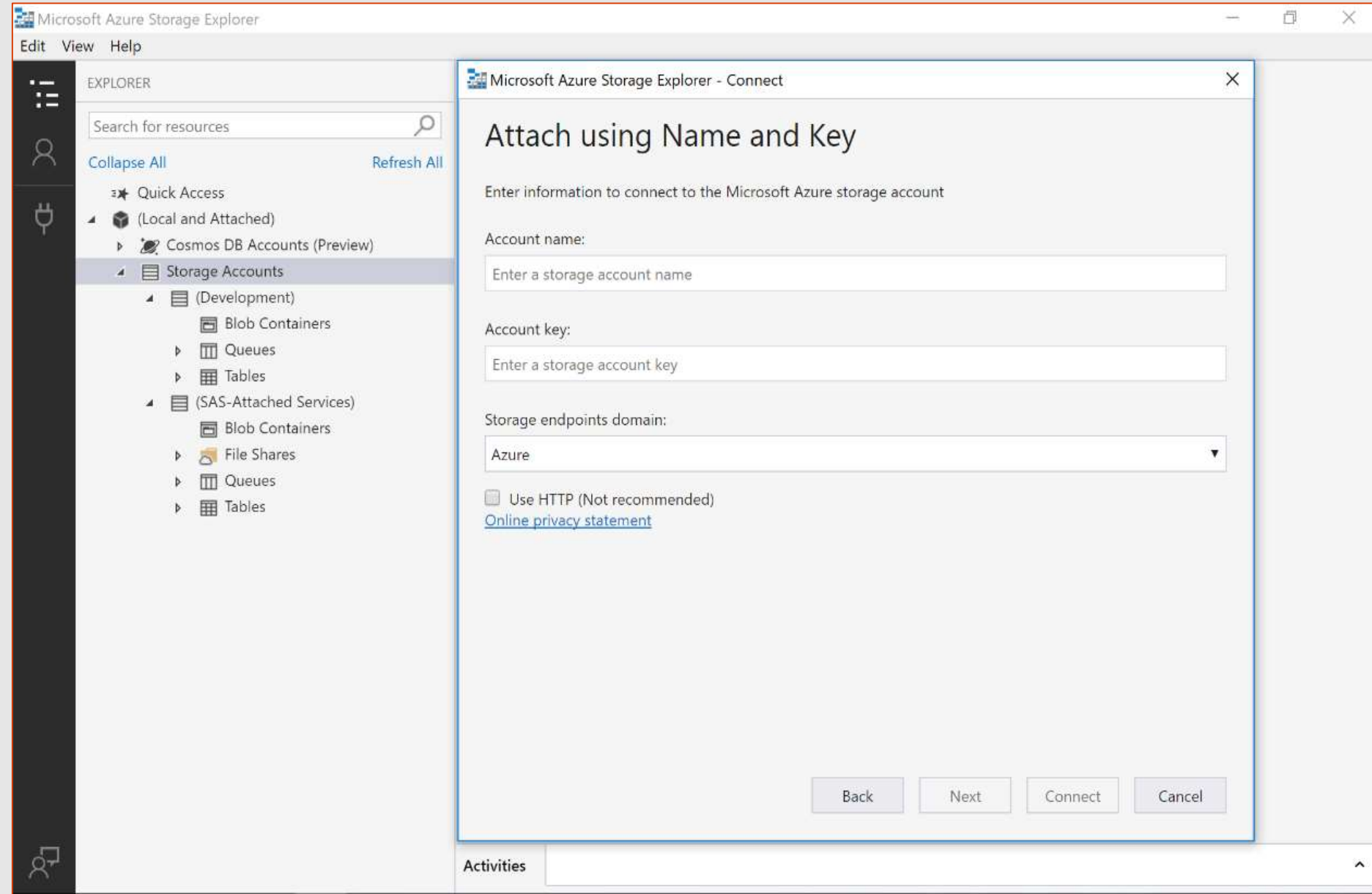
Custom Setup Interface – Instructions

- Select “Use a storage account name and key” and click “Next”



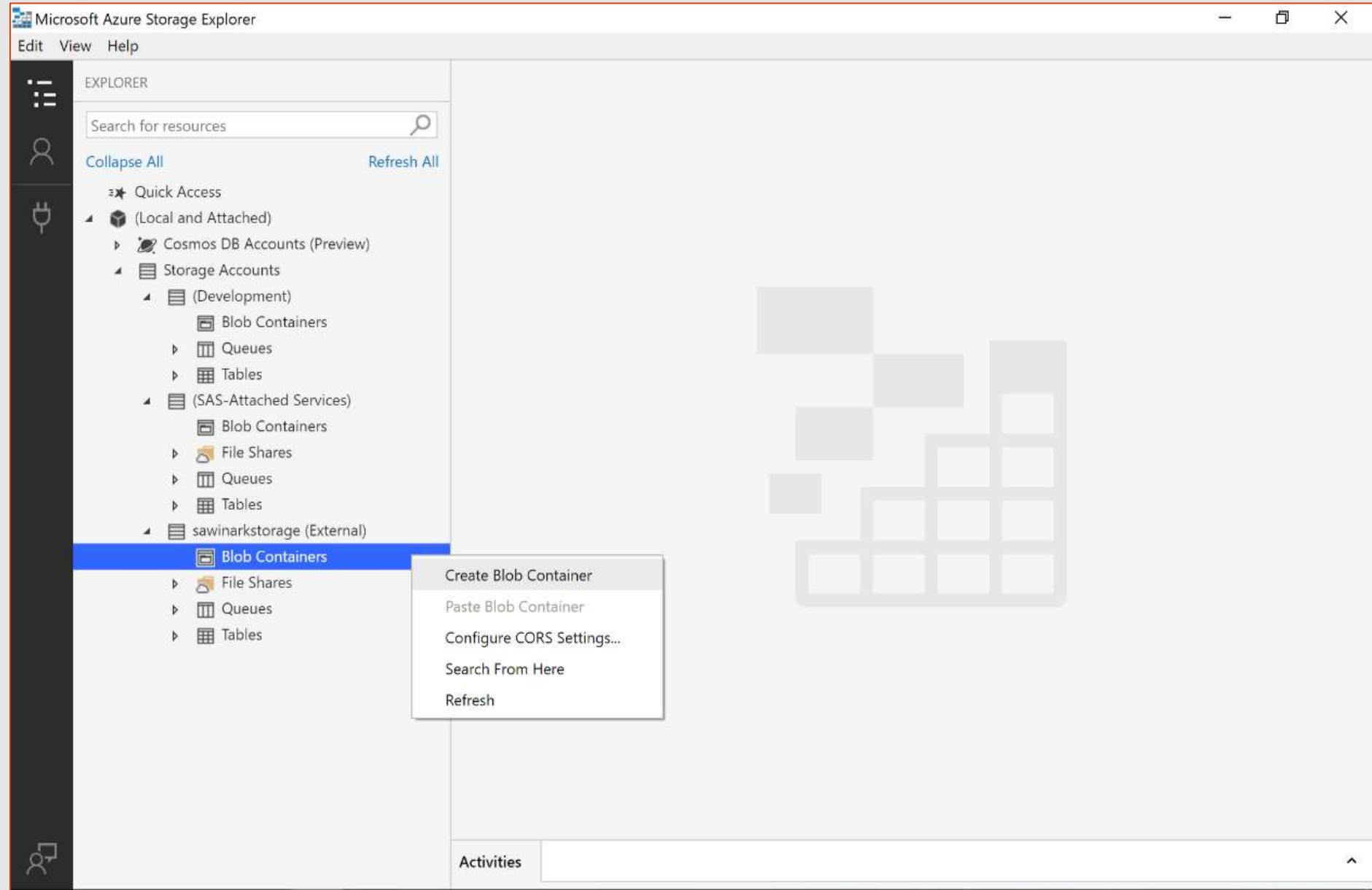
Custom Setup Interface – Instructions

- Enter your Azure Storage account name + key, click “Next”, and click “Connect”



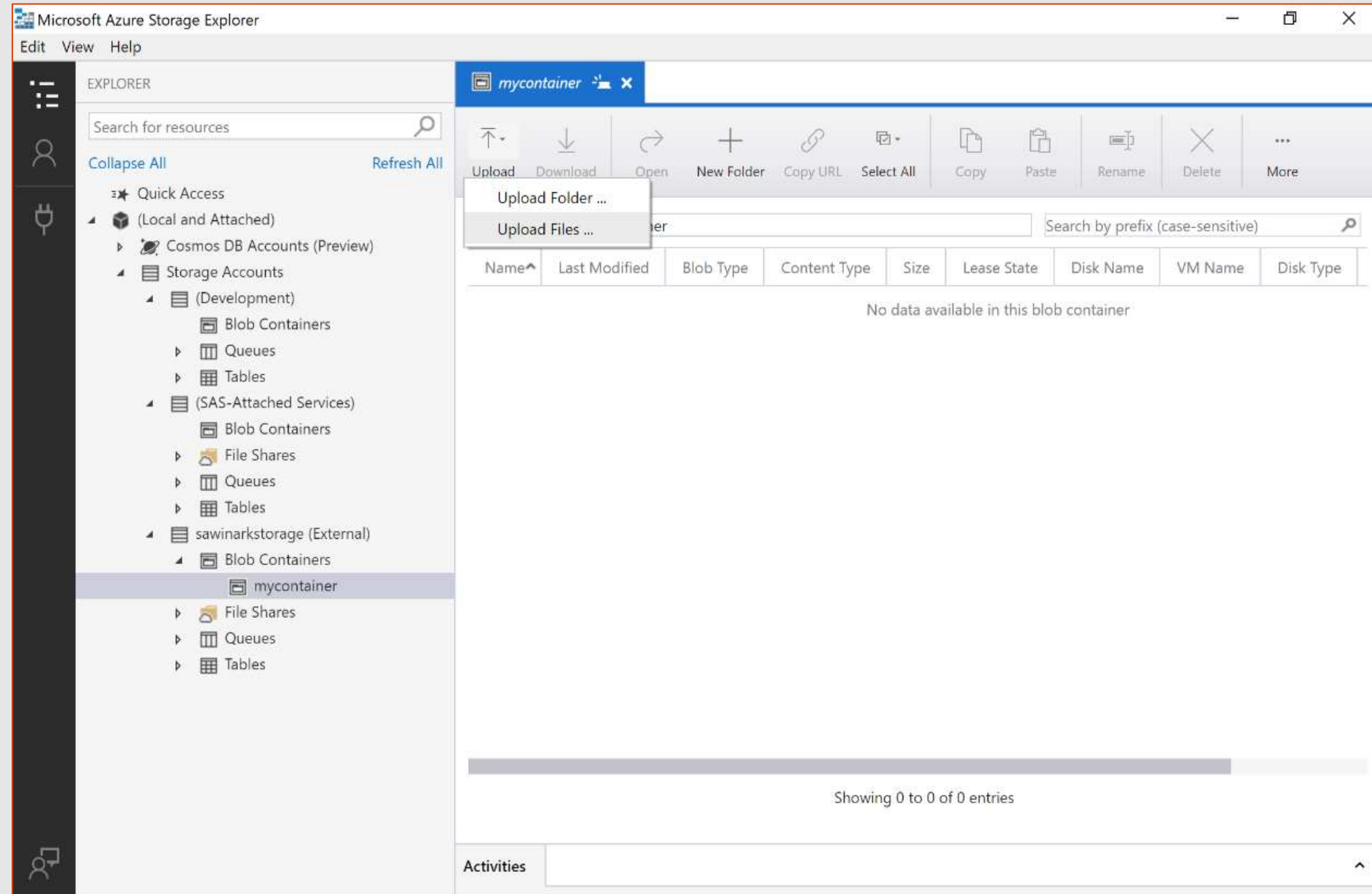
Custom Setup Interface – Instructions

- Under your connected Azure Storage account, right-click “Blob Containers”, select “Create Blob Container”, and name your new container



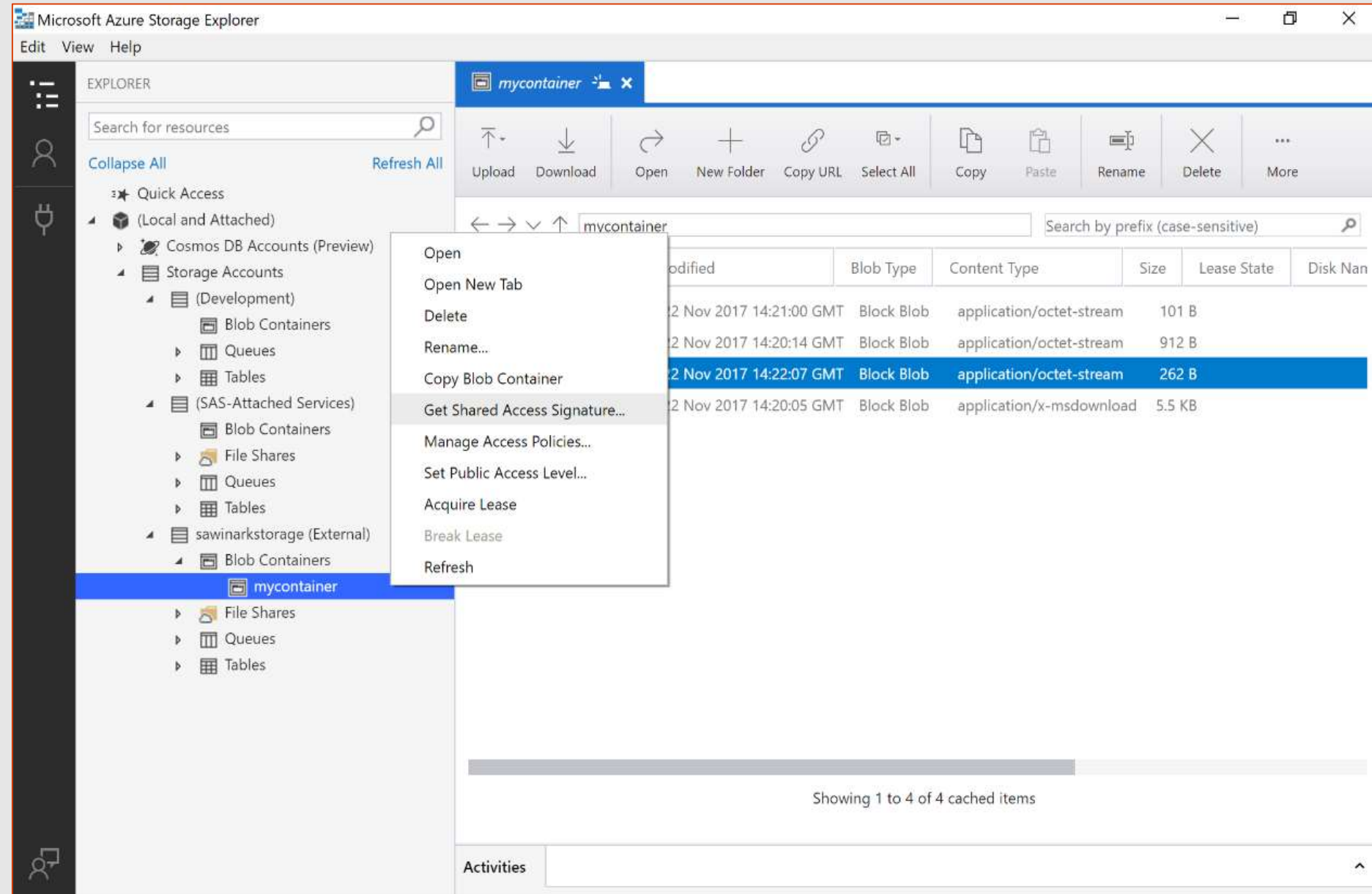
Custom Setup Interface – Instructions

- Select your container, upload your custom setup script + associated files, and make sure main.cmd is uploaded at the top level, not in any folder, of your container



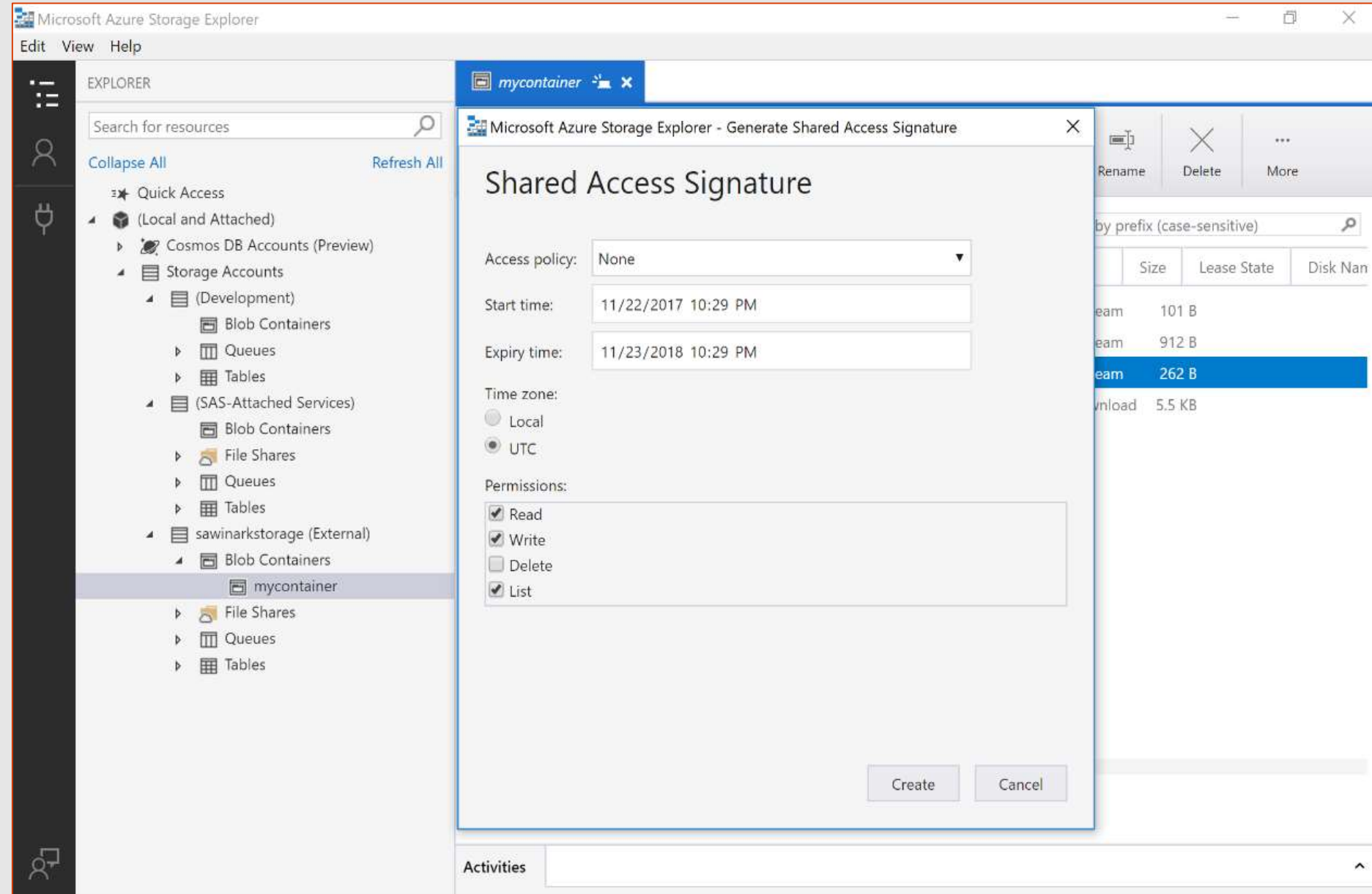
Custom Setup Interface – Instructions

- Right-click your container and select “Get Shared Access Signature...”



Custom Setup Interface – Instructions

- Create SAS URI of your container with sufficiently long expiry time and read + write + list permissions, since your custom setup script + associated files need to be redownloaded and re-executed whenever any node of your Azure-SSIS IR is reimaged, while write permission is required for uploading setup execution logs



Custom Setup Interface – Instructions

- Copy and save SAS URI of your container

The screenshot shows the Microsoft Azure Storage Explorer interface. On the left, the 'EXPLORER' pane displays a tree view of storage resources. Under 'Storage Accounts', the 'sawinarkstorage (External)' account is expanded, showing a 'Blob Containers' folder with a container named 'mycontainer' selected. The main pane shows the 'mycontainer' details. A dialog box titled 'Microsoft Azure Storage Explorer - Generate Shared Access Signature' is open in the foreground. The dialog has a title bar with a close button. The main content area is titled 'Shared Access Signature'. It contains three input fields: 'Container:' with the value 'mycontainer', 'URL:' with the value 'https://sawinarkstorage.blob.core.windows.net/mycontainer?st=201', and 'Query string:' with the value '?st=2017-11-22T22%3A29%3A00Z&se=2018-11-23T22%3A29%3A'. There are 'Copied' and 'Copy' buttons next to the URL and query string fields respectively. At the bottom of the dialog are 'Back' and 'Close' buttons. In the background, a table is partially visible with columns 'Size', 'Lease State', and 'Disk Nan'. The table has four rows: 'eam' with '101 B', 'eam' with '912 B', 'eam' with '262 B' (highlighted in blue), and 'ynload' with '5.5 KB'.

	Size	Lease State	Disk Nan
eam	101 B		
eam	912 B		
eam	262 B		
ynload	5.5 KB		

Custom Setup Interface – Instructions

- When provisioning/reconfiguring your Azure-SSIS IR via PSH, execute Set-AzureRmDataFactoryV2IntegrationRuntime cmdlet with SAS URI of your container as the value for new SetupScriptContainerSasUri parameter before starting your Azure-SSIS IR, e.g.

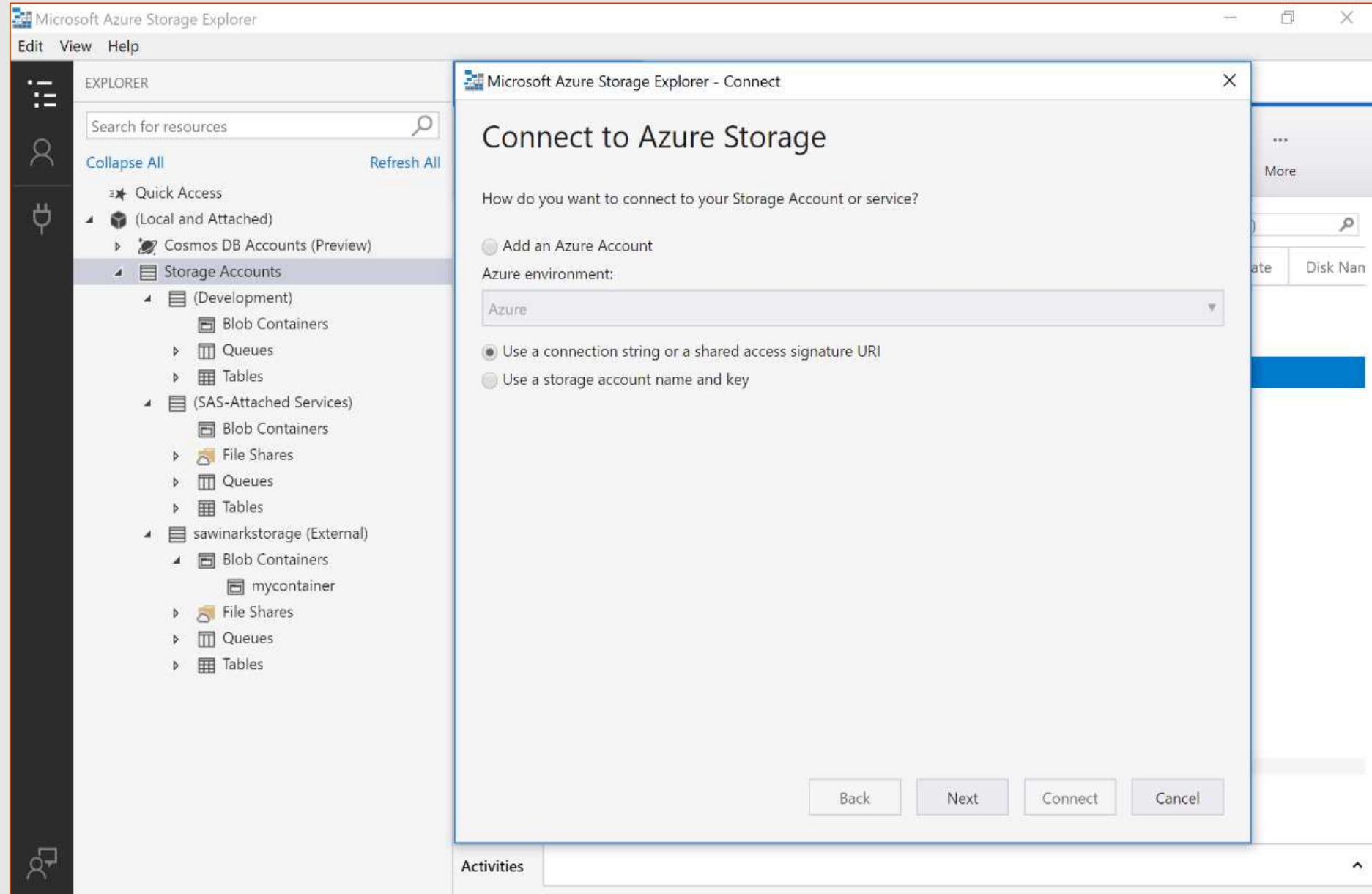
```
Set-AzureRmDataFactoryV2IntegrationRuntime -DataFactoryName $MyDataFactoryName  
-Name $MyAzureSsisIrName -ResourceGroupName $MyResourceGroupName  
-SetupScriptContainerSasUri $MySetupScriptContainerSasUri
```

```
Start-AzureRmDataFactoryV2IntegrationRuntime -DataFactoryName $MyDataFactoryName  
-Name $MyAzureSsisIrName -ResourceGroupName $MyResourceGroupName
```

- When your custom setup is completed/Azure-SSIS IR is started, you can find the standard output of main.cmd and other execution logs in main.cmd.log folder of your container

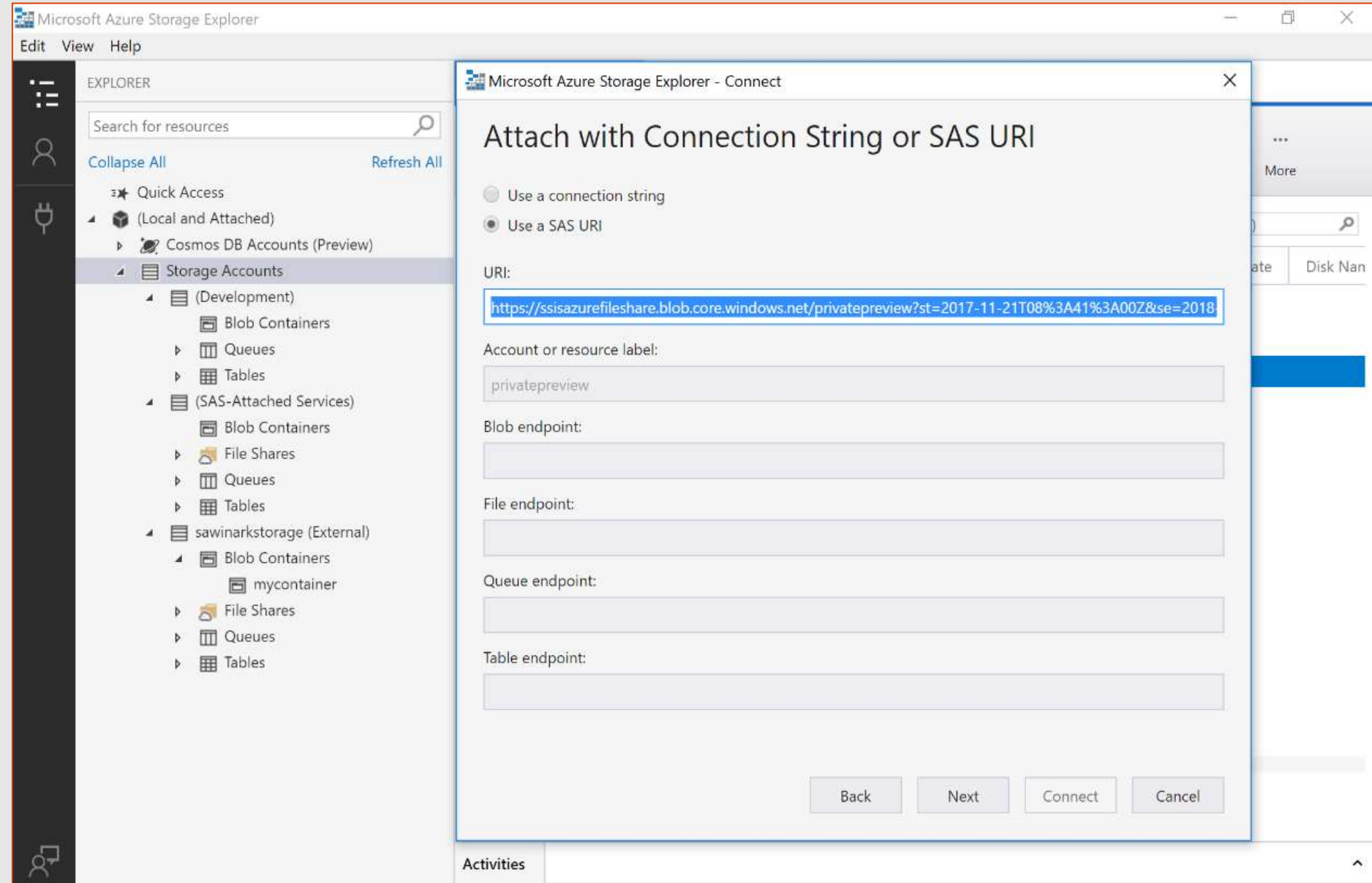
Custom Setup Interface – Examples

- Under “(Local and Attached)” menu item, right-click “Storage Accounts”, select “Connect to Azure storage...”, select “Use a connection string or a shared access signature URI”, and click “Next”



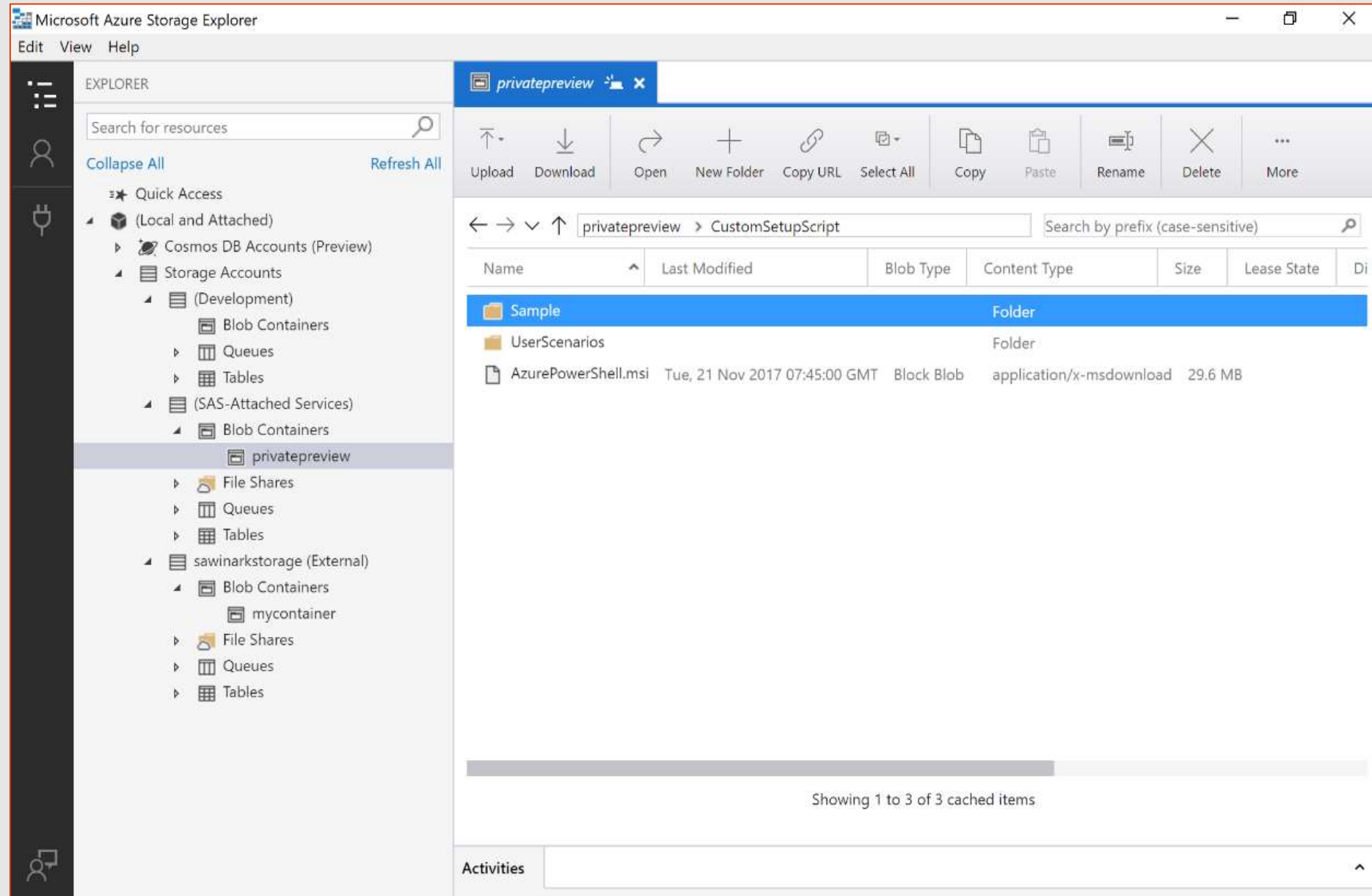
Custom Setup Interface – Examples

- Select “Use a SAS URI”, enter SAS URI of our Private Preview container (<https://ssisazurefileshare.blob.core.windows.net/privatepreview?st=2017-11-21T08%3A41%3A00Z&se=2018-11-22T08%3A41%3A00Z&sp=rl&sv=2017-04-17&sr=c&sig=kalXup1%2B4x49f57GKmUKTTFSZB%2FKRYlwN5tU%2F6Mk%2B4I%3D>), click “Next”, and click “Connect”



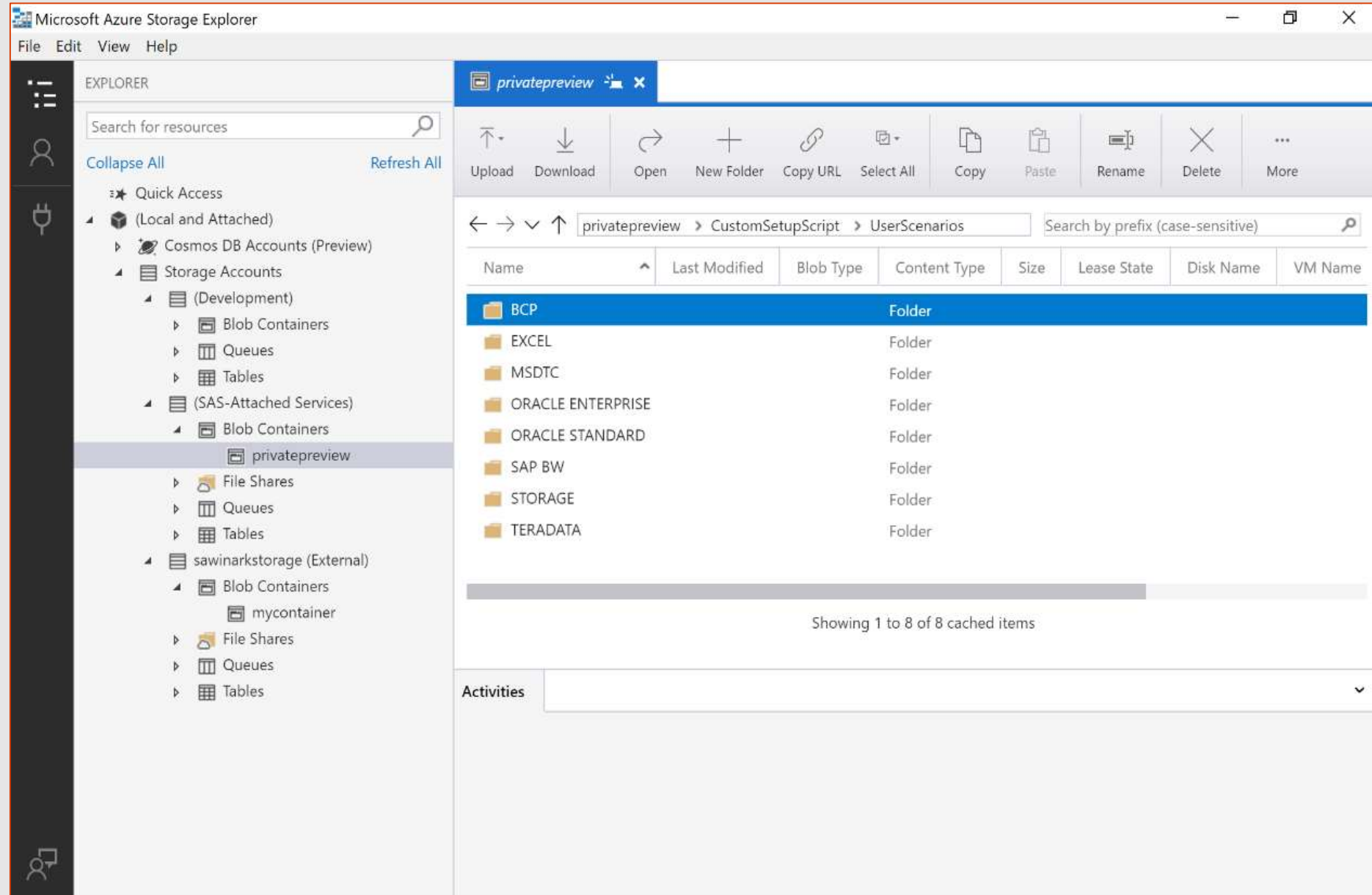
Custom Setup Interface – Examples

- Select our connected Private Preview container, double click “CustomSetupScript”, and you can find
 - “Sample” folder that contains a custom setup to install simple task that just sleeps for a few seconds on each node of your Azure-SSIS IR – It also contains `gacinstall.cmd` that replaces gacutil tool (see [Limitations](#))
 - “UserScenarios” folder that contains 8 custom setups for real user scenarios
 - `AzurePowerShell.msi`, which is our private version of Azure PSH



Custom Setup Interface – Examples

- Double click “UserScenarios” and you can find
 - “BCP” folder that contains a custom setup to install SQL Server command line utilities (**MsSqlCmdLnUtils.msi**), including bulk copy program (bcp), on each node of your Azure-SSIS IR
 - “EXCEL” folder that contains a custom setup to install Open Source assemblies (**DocumentFormat.OpenXml.dll**, **ExcelDataReader.DataSet.dll**, and **ExcelDataReader.dll**) on each node of your Azure-SSIS IR



Custom Setup Interface – Examples

- “MSDTC” folder that contains a custom setup to enable/start Microsoft Distributed Transaction Coordinator (MSDTC) service on each node of your Azure-SSIS IR
- “ORACLE ENTERPRISE” folder that contains a custom setup script (main.cmd) and silent install config file (client.rsp) to install Oracle OCI driver on each node of your Azure-SSIS IR Enterprise Edition, so you can use Oracle Connection Manager/Source/Destination – You need to first download winx64_12102_client.zip from [Oracle's site](#) and then upload it together with main.cmd and client.rsp into your container – If you use TNS to connect to Oracle, you also need to download tnsnames.ora file, then edit and upload it into your container, so it will be copied into Oracle installation folder during setup
- “ORACLE STANDARD” folder that contains a custom setup script (main.cmd) to install Oracle ODP.NET driver on each node of your Azure-SSIS IR, so you can use it with ADO.NET Connection Manager/Source/Destination – You need to first download ODP.NET_Managed_ODAC122cR1.zip from [Oracle's site](#) and then upload it together with main.cmd into your container

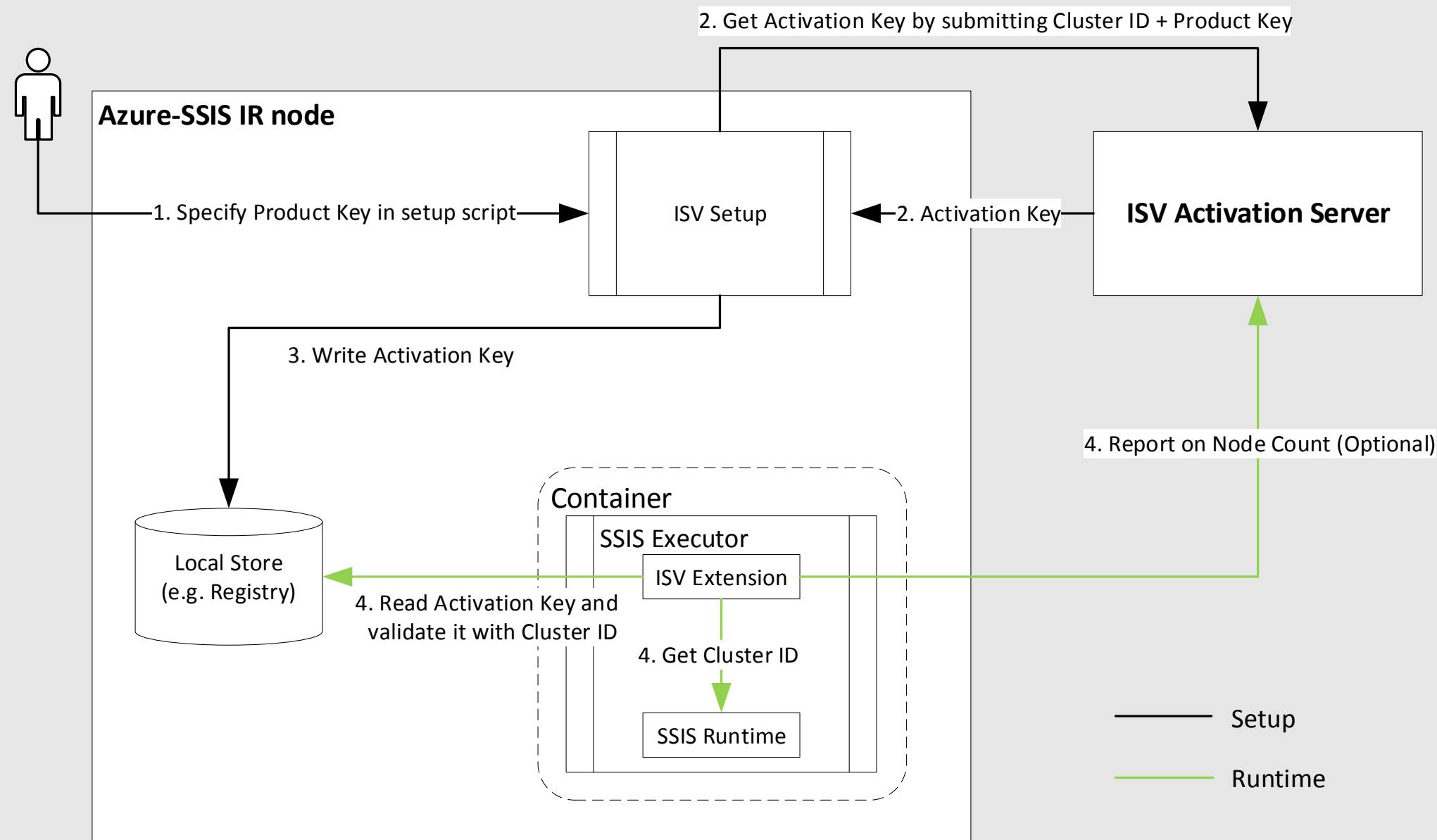
Custom Setup Interface – Examples

- “SAP BW” folder that contains a custom setup script (main.cmd) to install SAP .NET connector assembly (librfc32.dll) on each node of your Azure-SSIS IR Enterprise Edition, so you can use [SAP BW Connection Manager/Source/Destination](#) – You need to first upload the 64-bit/32-bit version of librfc32.dll from SAP installation folder into your container together with main.cmd that will then copy it into %windir%\System32/%windir%\SysWow64 folder, respectively, during setup
- “STORAGE” folder that contains a custom setup to install Azure PSH on each node of your Azure-SSIS IR, so you can deploy and run SSIS packages that execute [PSH scripts to manipulate your Azure Storage account](#) – You need to copy main.cmd + AzurePowerShell.msi + storage.ps1 to your container and use PowerShell.dtsx as a template for your packages, combining [Azure Blob Download Task](#) that downloads storage.ps1 as a modifiable PSH script and [Execute Process Task that executes it](#) on each node
- “TERADATA” folder that contains a custom setup script (main.cmd), its associated file (install.cmd), and installer package (.msi) files to install Teradata connectors + TPT API + ODBC driver on each node of your Azure-SSIS IR Enterprise Edition, so you can use [Teradata Connection Manager/Source/Destination](#) – You need to first download Teradata Tools and Utilities (TTU) 15.x zip file (e.g. TeradataToolsAndUtilitiesBase__windows_indep.15.10.22.00.zip) from [Teradata’s site](#) and then upload it together with the above .cmd/.msi files into your container

Custom Setup Interface – Licensing

- To support the installation of premium/licensed components from our ISV partners, we face challenges from the nature of Azure-SSIS IR:
 - The nodes of Azure-SSIS IR are volatile in the sense that they can be allocated/released at any time, e.g. customers can start/stop their nodes to manage the running cost or scale up/down through various node sizes as they see fit, so binding a component installation to any particular node by collecting machine-specific info that is traditionally used for on-premises installations, e.g. MAC address, CPU ID, etc. isn't viable
 - Customers can also scale in/out their Azure-SSIS IR, so that the number of nodes can shrink/expand at any time between 1 – 10 (or more by a special request)
- Consequently, we propose to introduce new system variables in SSIS runtime that can be referenced by ISV components as the unique/persistent info for Azure-SSIS IR, e.g. Cluster ID and Cluster Node Count
- ISVs can now bind their component installations to Azure-SSIS IR as a cluster, whose ID is invariant when customers start/stop, scale up/down, scale in/out, or reconfigure their Azure-SSIS IR in any way

Custom Setup Interface – Licensing



Custom Setup Interface – Licensing

- ISVs can offer their components in various SKUs/tiers (e.g. single node, up-to-5 nodes, up-to-10 nodes, etc.) and provide the corresponding Product Key when customers purchase any of them
- They can also provide Azure Storage blob container containing ISV Setup script and associated files that customers can copy & paste into their own container and modify with their own purchased Product Key (e.g. `IsvSetup.exe -pid xxxx-xxxx-xxxx`)
- Customers can then provision/reconfigure their Azure-SSIS IR with SAS URI of their container as parameter
- As Azure-SSIS IR is being provisioned/reconfigured, ISV Setup will be executed on each of its nodes to submit its Cluster ID and the purchased Product Key to ISV Activation Server that will generate an Activation Key
- After receiving an Activation Key, ISV Setup can store it locally on the node (e.g. in Registry)
- When customers use ISV Extension in their packages that run on a node, it will read the locally stored Activation Key and validate it with the node's Cluster ID
- It can also optionally report Cluster Node Count to ISV Activation Server

Custom Setup Interface – Licensing

```
// Validation code example
public override DTSExecResult Validate(Connections, VariableDispenser,
IDTSComponentEvents componentEvents, IDTSLogging log)
{
    Variables vars = null;
    variableDispenser.LockForRead("System::ClusterId");
    variableDispenser.LockForRead("System::ClusterNodeCount");
    variableDispenser.GetVariables(ref vars);

    // Validate Activation Key with ClusterId
    // Report on ClusterNodeCount

    vars.Unlock();

    return base.Validate(connections, variableDispenser, componentEvents, log);
}
```

Provisioning Methods

Provisioning Methods

- Azure-SSIS IR can be provisioned via ADFv2 App
- Azure-SSIS IR can be provisioned via PowerShell (PSH)/custom code using ADFv2 .NET SDK/API

Provisioning via ADFv2 App

The screenshot shows the Microsoft Azure portal interface. On the left, the 'New' blade is open, displaying the 'Data + Analytics' category. The 'Data Factory' option is highlighted with a red circle. A red callout bubble points to this option with the text 'Create ADFv2 if you have not done so already'.

The 'New data factory' form on the right contains the following fields:

- Name:** sawinark-ADFv2
- Subscription:** EIMS_TEST_EVEREST_1
- Resource Group:** Use existing (selected), sawinarkRG
- Version:** V2 (Preview)
- Location:** West Europe

At the bottom of the form, there is a checkbox for 'Pin to dashboard' and a 'Create' button.

Provisioning via ADFv2 App

Microsoft Azure

Report a bug Search resources, services and docs

Dashboard ▾ + New dashboard Edit dashboard Share Fullscreen Clone Delete

All resources ALL SUBSCRIPTIONS Refresh

sawinarkssisdb	SQL server
disinazure	SQL server
sawinarkWorker2	Virtual machine
sawinarkWorker1	Virtual machine
sawinarkMaster	Virtual machine
sawinarkClassicVNet	Virtual network (classic)
sawinarkARMVNet	Virtual network
SSISDB	SQL database
ScaleOutDemo-vnet	Virtual network
sawinarkSQLDB2	SQL database
ssisazurefileshare	Storage account
sawinarkSQLDB	SQL database
sawinarkMaster-ip	Public IP address

See more...

sawinarkSQLDB SQL DATABASE Online	sawinarkSQLDB2 SQL DATABASE Online	sawinarkstorage Available	ssisazurefileshare Available	sawinarkadls DATA LAKE STORE Running
sawinarkMaster Stopped	sawinarkWorker1 Stopped	sawinarkWorker2 Stopped	sawinarkClassicVNet	sawinarkARMVNet
sawinarkADF DATA FACTORY	sawinarkADFv1 DATA FACTORY	sawinarkADFv2 DATA FACTORY	sawinarkssisdb SQL SERVER Available	disinazure SQL SERVER Available

Open ADFv2 settings page

Provisioning via ADFv2 App

The screenshot displays the Microsoft Azure portal interface for a Data factory (ADFv2) named 'sawinarkADFv2'. The left sidebar shows the navigation pane with the 'Author & Monitor' link highlighted. The main content area shows the 'Settings' page for the Data factory, including details like Resource group (sawinarkRG), Location (westeurope), and Provisioning state (Succeeded). The 'Quick links' section contains links for Documentation and Author & Monitor. The 'Monitoring' section shows PipelineRuns and ActivityRuns. A red circle is drawn around the 'Author & Monitor' link, and a red arrow points from the text 'Open ADFv2 App' to it.

Microsoft Azure

Home > sawinarkADFv2 > Settings

sawinarkADFv2
Data factory

Delete

Essentials

Resource group
sawinarkRG

Location
westeurope

Provisioning state
Succeeded

Getting started
Quick start

Type
Data factory (V2)

Subscription name
EIMS_TEST_EVEREST_1

Subscription id
cb715d05-3337-4640-8c43-4f943c50d06e

All settings

Quick links

Documentation

Author & Monitor

Monitoring

PipelineRuns

ActivityRuns

Settings

Filter settings

SUPPORT + TROUBLESHOOTING

Diagnose and solve problems

Activity log

New support request

GENERAL

Properties

GETTING STARTED

Quick start

RESOURCE MANAGEMENT

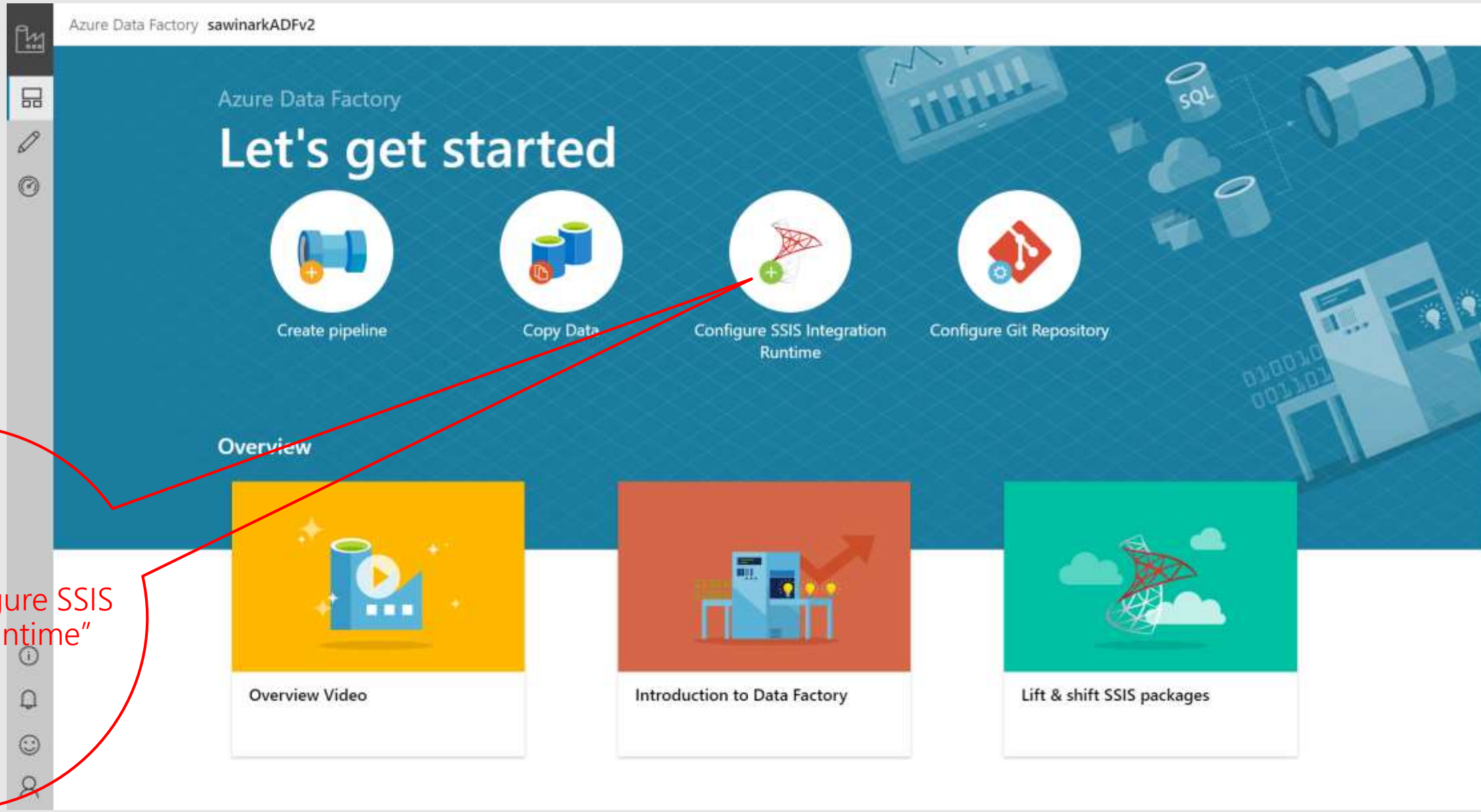
Tags

Locks

Users

Open ADFv2 App

Provisioning via ADFv2 App



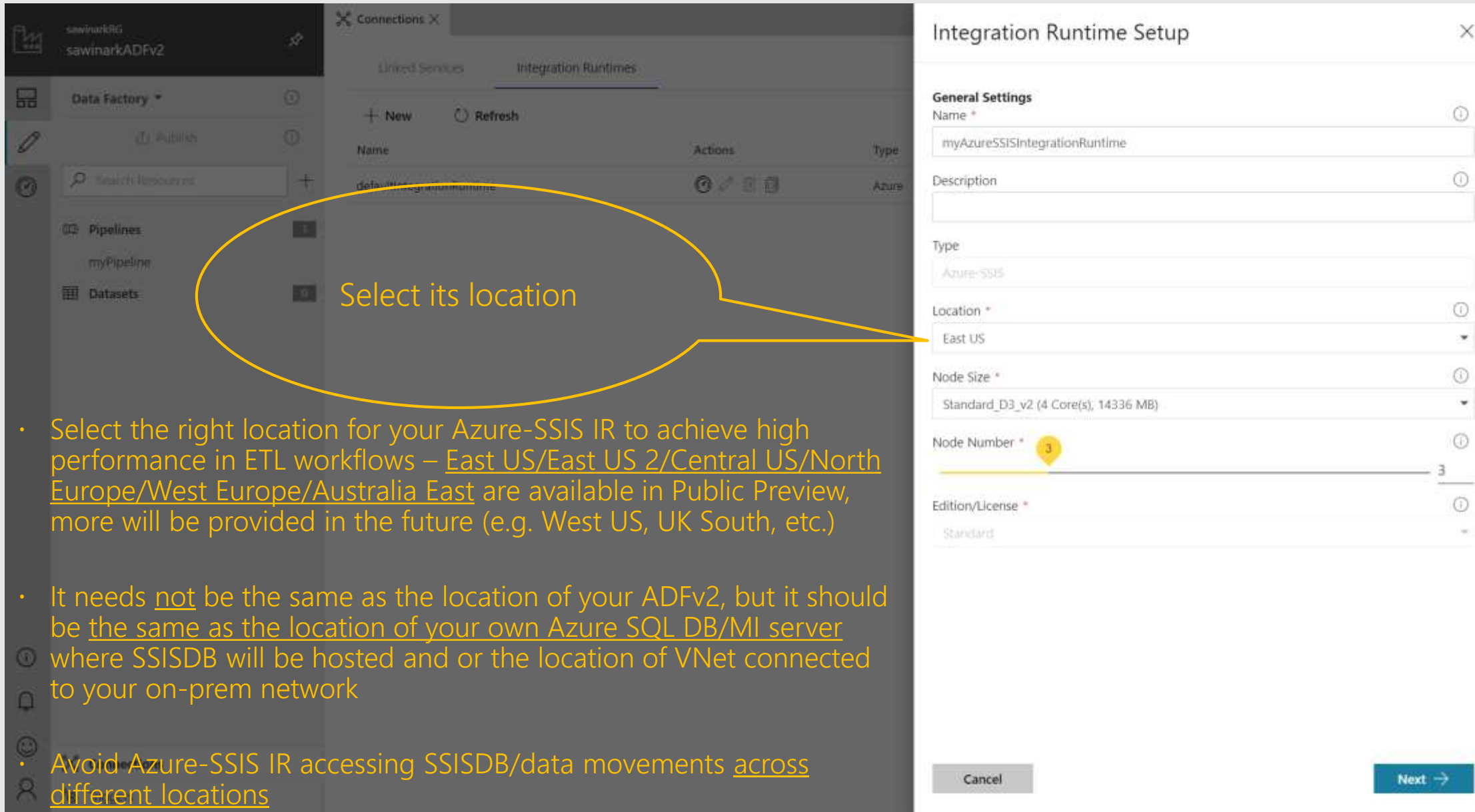
Provisioning via ADFv2 App

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the navigation pane shows 'Data Factory', 'Pipelines', and 'Datasets'. The main area is titled 'Integration Runtimes' and contains a table with columns 'Name', 'Actions', and 'Type'. A yellow callout bubble points to the 'Name' column, containing the text: 'Name and describe your Azure-SSIS IR'. On the right, the 'Integration Runtime Setup' dialog is open, showing the 'General Settings' section. The fields are filled with the following values:

- Name: myAzureSSISIntegrationRuntime
- Description: (empty)
- Type: Azure-SSIS
- Location: East US
- Node Size: Standard_D3_v2 (4 Core(s), 14336 MB)
- Node Number: 3
- Edition/License: Standard

At the bottom of the dialog, there are 'Cancel' and 'Next' buttons.

Provisioning via ADFv2 App



Select its location

- Select the right location for your Azure-SSIS IR to achieve high performance in ETL workflows – East US/East US 2/Central US/North Europe/West Europe/Australia East are available in Public Preview, more will be provided in the future (e.g. West US, UK South, etc.)
- It needs not be the same as the location of your ADFv2, but it should be the same as the location of your own Azure SQL DB/MI server where SSISDB will be hosted and or the location of VNet connected to your on-prem network
- Avoid Azure-SSIS IR accessing SSISDB/data movements across different locations

Integration Runtime Setup

General Settings

Name *
myAzureSSISIntegrationRuntime

Description

Type
Azure-SSIS

Location *
East US

Node Size *
Standard_D3_v2 (4 Core(s), 14336 MB)

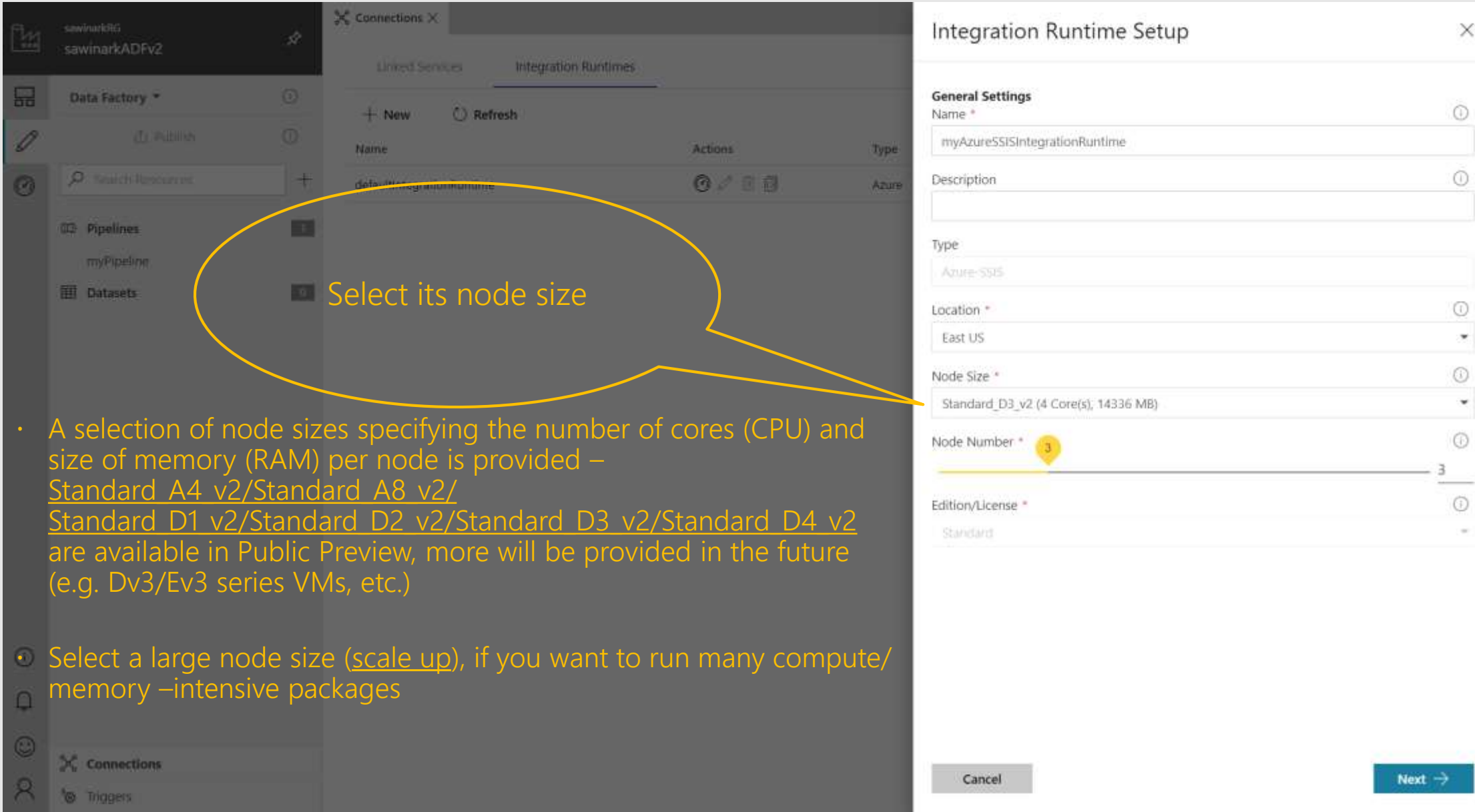
Node Number *
3

Edition/License *
Standard

Cancel

Next →

Provisioning via ADFv2 App



The screenshot shows the Azure Data Factory v2 console. In the background, the 'Integration Runtimes' list is visible, and a yellow callout bubble points to the 'Node Size' dropdown in the 'Integration Runtime Setup' dialog. The callout bubble contains the text 'Select its node size'.

Integration Runtime Setup

General Settings

Name * myAzureSSISIntegrationRuntime

Description

Type Azure-SSIS

Location * East US

Node Size * Standard_D3_v2 (4 Core(s), 14336 MB)

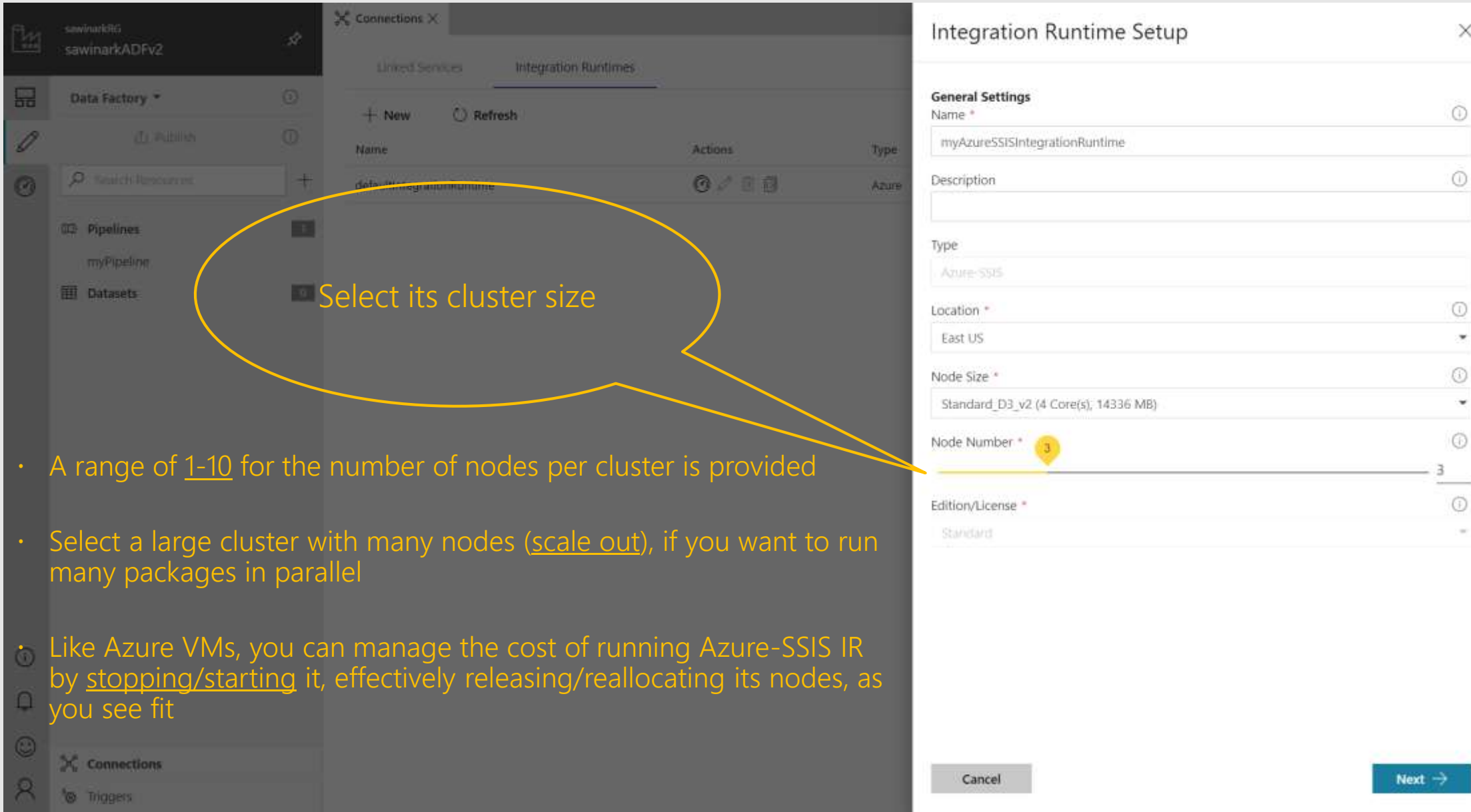
Node Number * 3

Edition/License * Standard

Cancel Next →

- A selection of node sizes specifying the number of cores (CPU) and size of memory (RAM) per node is provided –
Standard A4 v2/Standard A8 v2/
Standard D1 v2/Standard D2 v2/Standard D3 v2/Standard D4 v2
are available in Public Preview, more will be provided in the future (e.g. Dv3/Ev3 series VMs, etc.)
- Select a large node size (scale up), if you want to run many compute/memory –intensive packages

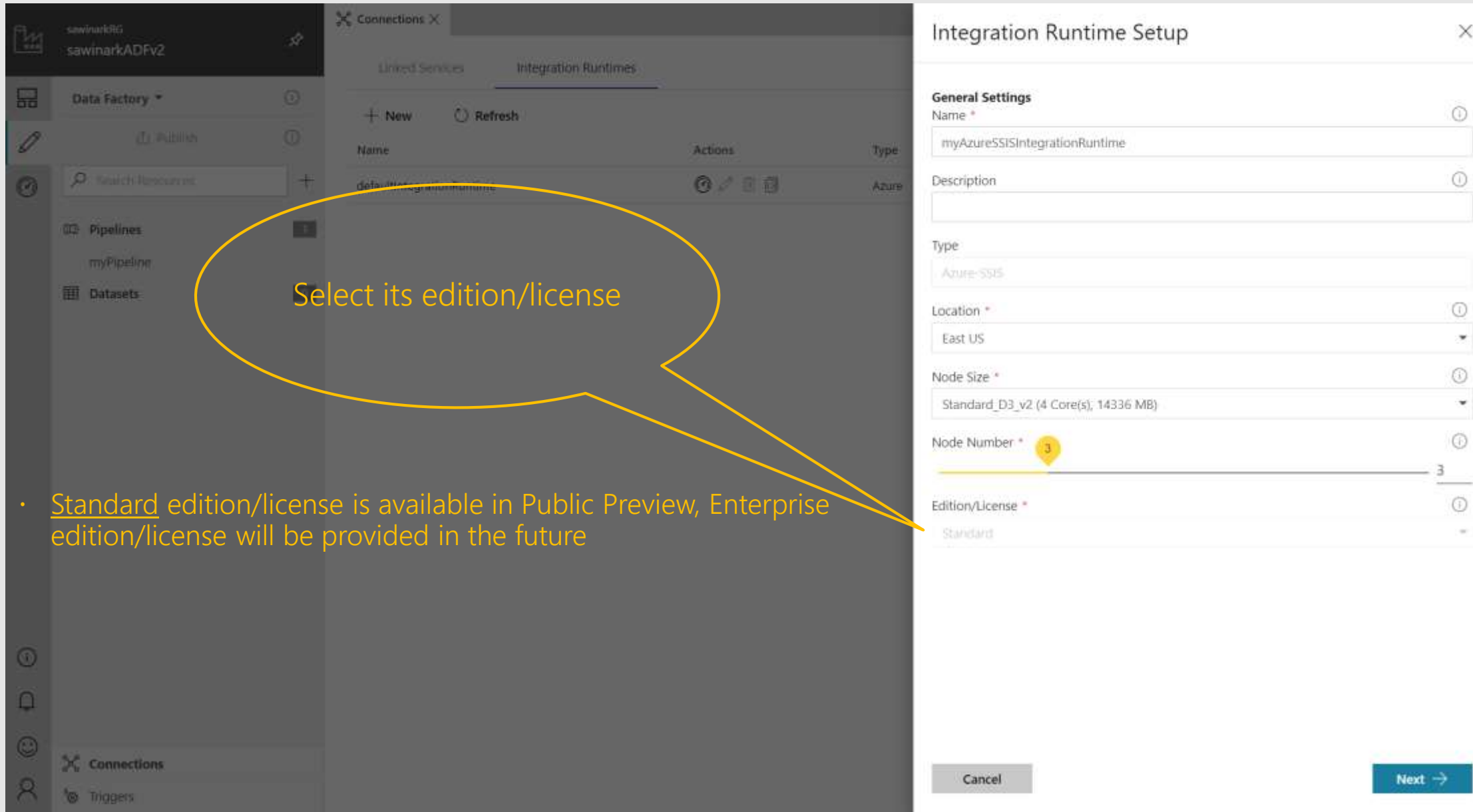
Provisioning via ADFv2 App



Select its cluster size

- A range of 1-10 for the number of nodes per cluster is provided
- Select a large cluster with many nodes (scale out), if you want to run many packages in parallel
- Like Azure VMs, you can manage the cost of running Azure-SSIS IR by stopping/starting it, effectively releasing/reallocating its nodes, as you see fit

Provisioning via ADFv2 App



Select its edition/license

- Standard edition/license is available in Public Preview, Enterprise edition/license will be provided in the future

Integration Runtime Setup

General Settings

Name *
myAzureSSISIntegrationRuntime

Description

Type
Azure-SSIS

Location *
East US

Node Size *
Standard_D3_v2 (4 Core(s), 14336 MB)

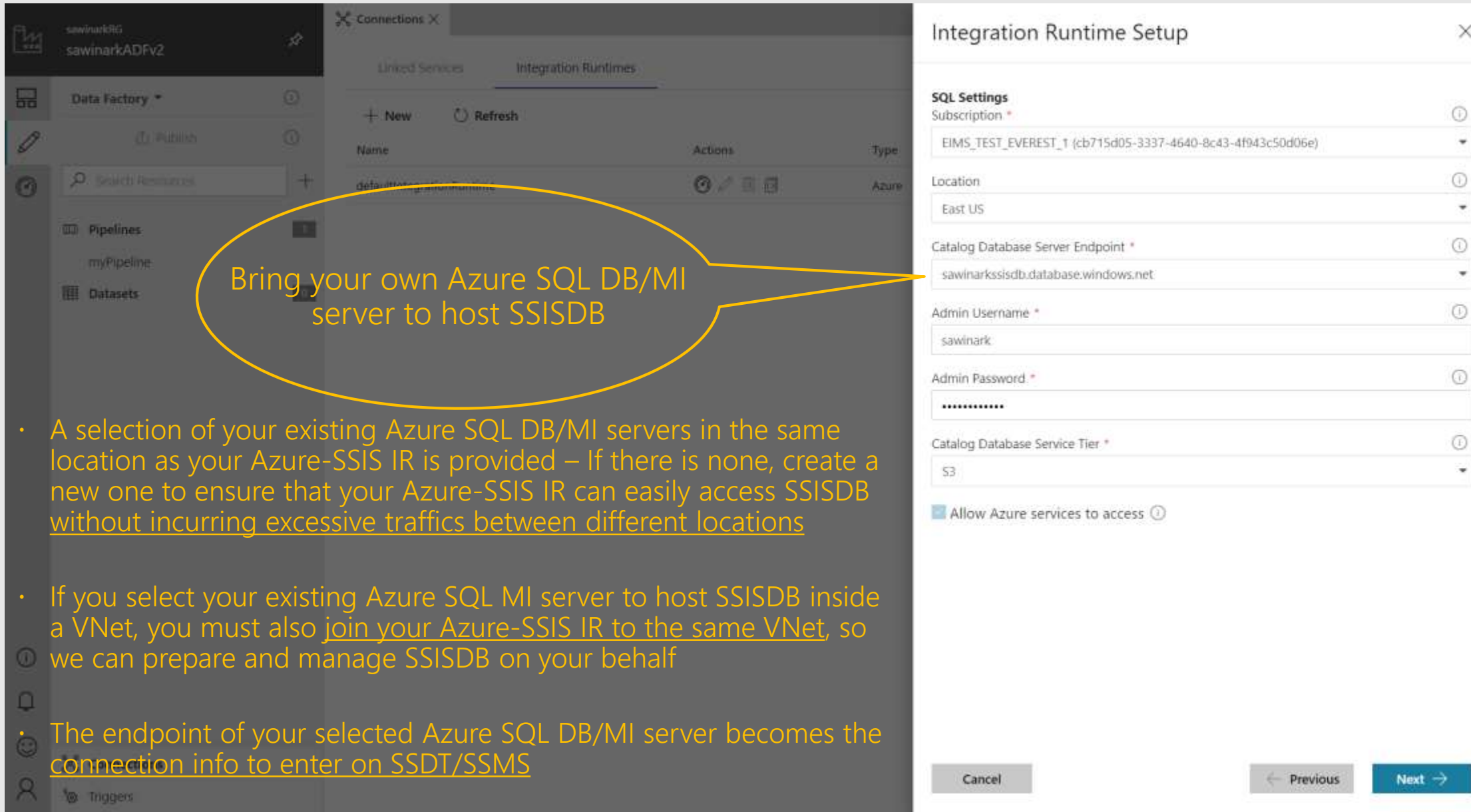
Node Number *
3

Edition/License *
Standard

Cancel

Next →

Provisioning via ADFv2 App



Bring your own Azure SQL DB/MI server to host SSISDB

- A selection of your existing Azure SQL DB/MI servers in the same location as your Azure-SSIS IR is provided – If there is none, create a new one to ensure that your Azure-SSIS IR can easily access SSISDB without incurring excessive traffics between different locations
- If you select your existing Azure SQL MI server to host SSISDB inside a VNet, you must also join your Azure-SSIS IR to the same VNet, so we can prepare and manage SSISDB on your behalf
- The endpoint of your selected Azure SQL DB/MI server becomes the connection info to enter on SSDT/SSMS

Integration Runtime Setup

SQL Settings

Subscription *
EIMS_TEST_EVEREST_1 (cb715d05-3337-4640-8c43-4f943c50d06e)

Location
East US

Catalog Database Server Endpoint *
sawinarkssisdb.database.windows.net

Admin Username *
sawinark

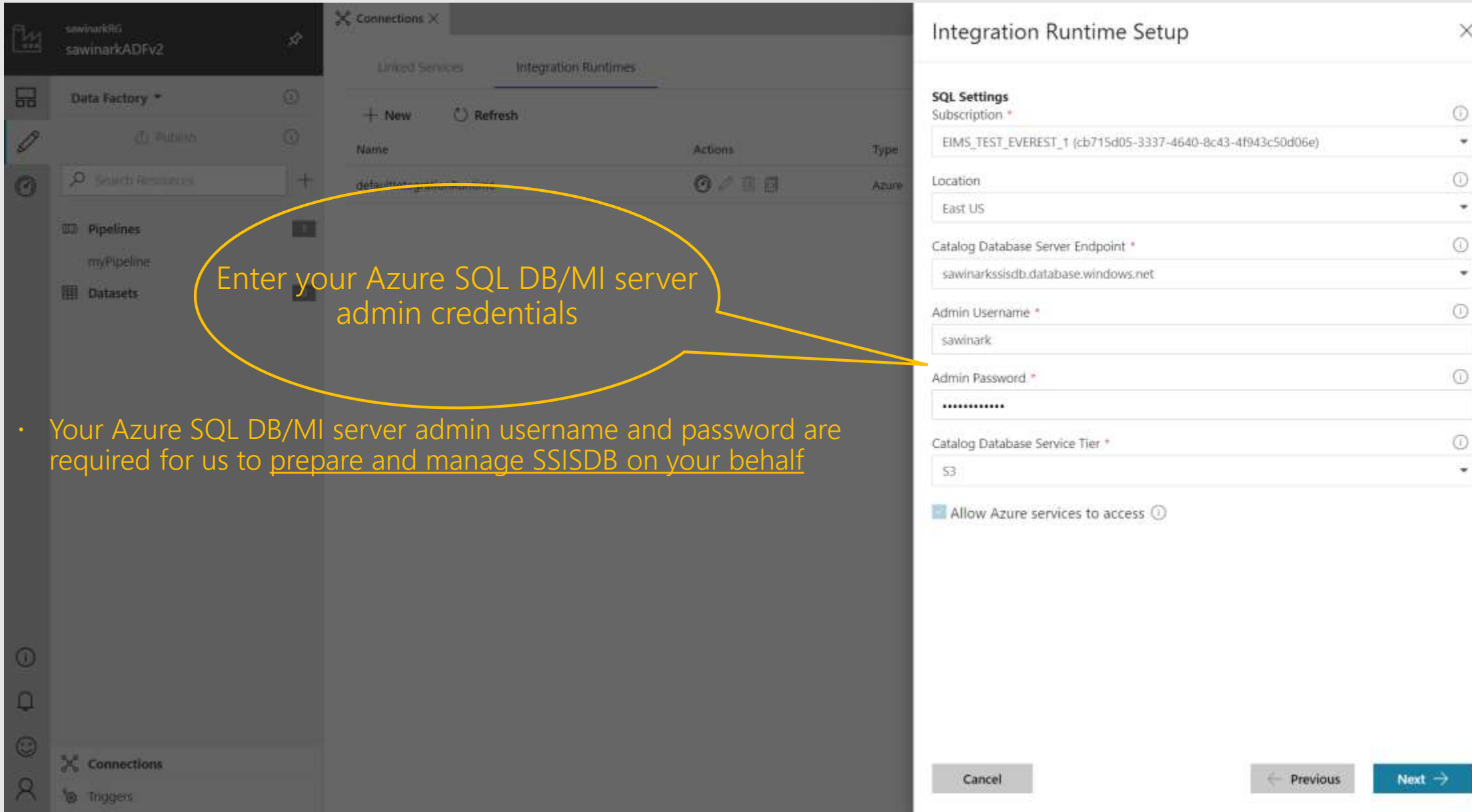
Admin Password *

Catalog Database Service Tier *
S3

☒ Allow Azure services to access ⓘ

Cancel Previous Next →

Provisioning via ADFv2 App



The screenshot displays the Azure Data Factory v2 console. On the left, the 'Connections' tab is active, showing a table of existing connections. A yellow callout bubble points to the 'Admin Username' and 'Admin Password' fields in the 'Integration Runtime Setup' dialog, with the text 'Enter your Azure SQL DB/MI server admin credentials'. The dialog is titled 'Integration Runtime Setup' and contains the following fields:

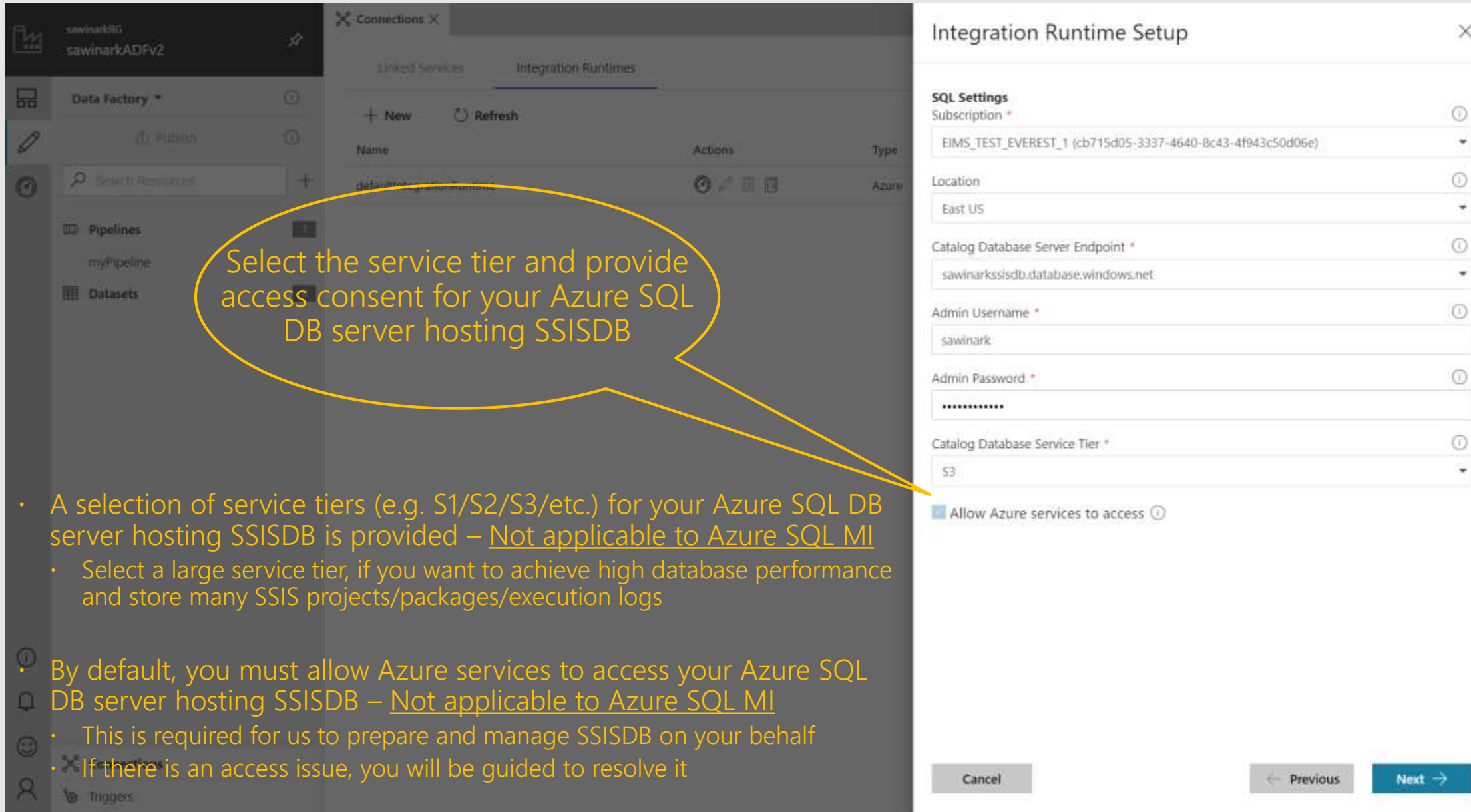
- SQL Settings**
 - Subscription: EIMS_TEST_EVEREST_1 (cb715d05-3337-4640-8c43-4f943c50d06e)
 - Location: East US
 - Catalog Database Server Endpoint: sawinarkssisdb.database.windows.net
 - Admin Username: sawinark
 - Admin Password: [Redacted]
 - Catalog Database Service Tier: S3
- ☒ Allow Azure services to access

At the bottom of the dialog, there are three buttons: 'Cancel', 'Previous', and 'Next'.

Name	Actions	Type
defaultintegrationruntime	[Icons]	Azure

- Your Azure SQL DB/MI server admin username and password are required for us to prepare and manage SSISDB on your behalf

Provisioning via ADFv2 App



Select the service tier and provide access consent for your Azure SQL DB server hosting SSISDB

- A selection of service tiers (e.g. S1/S2/S3/etc.) for your Azure SQL DB server hosting SSISDB is provided – Not applicable to Azure SQL MI
 - Select a large service tier, if you want to achieve high database performance and store many SSIS projects/packages/execution logs
- By default, you must allow Azure services to access your Azure SQL DB server hosting SSISDB – Not applicable to Azure SQL MI
 - This is required for us to prepare and manage SSISDB on your behalf
 - If there is an access issue, you will be guided to resolve it

Integration Runtime Setup

SQL Settings

Subscription *
EIMS_TEST_EVEREST_1 (cb715d05-3337-4640-8c43-4f943c50d06e)

Location
East US

Catalog Database Server Endpoint *
sawinarkssisdb.database.windows.net

Admin Username *
sawinark

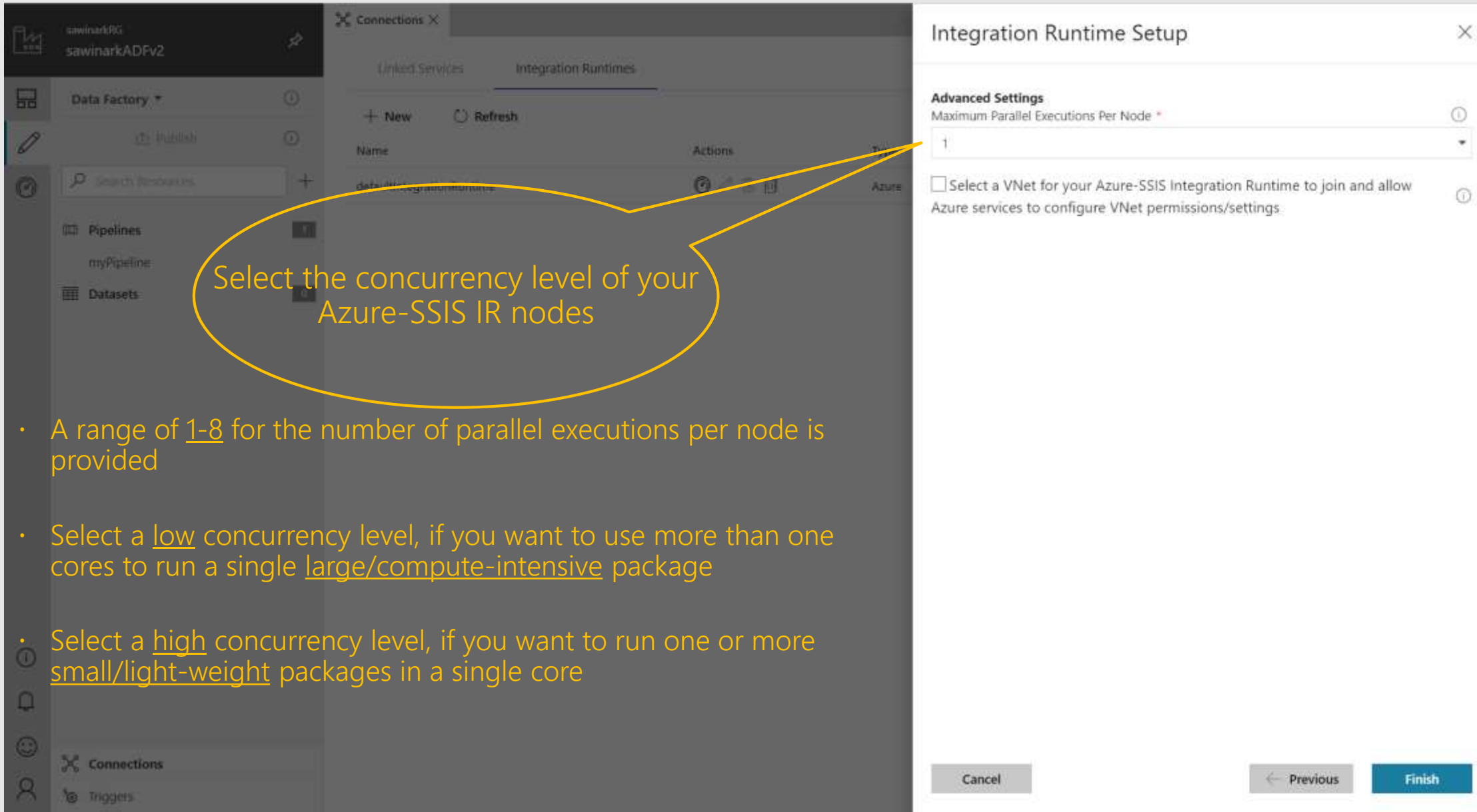
Admin Password *

Catalog Database Service Tier *
S3

☒ Allow Azure services to access ⓘ

Cancel Previous Next →

Provisioning via ADFv2 App



The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the navigation pane shows 'Data Factory', 'Pipelines', 'myPipeline', and 'Datasets'. The main area shows the 'Integration Runtimes' tab with a table listing the runtime. A yellow callout points to the 'Maximum Parallel Executions Per Node' dropdown in the 'Integration Runtime Setup' dialog, which is set to 1. The callout text reads: 'Select the concurrency level of your Azure-SSIS IR nodes'.

Integration Runtime Setup

Advanced Settings

Maximum Parallel Executions Per Node *

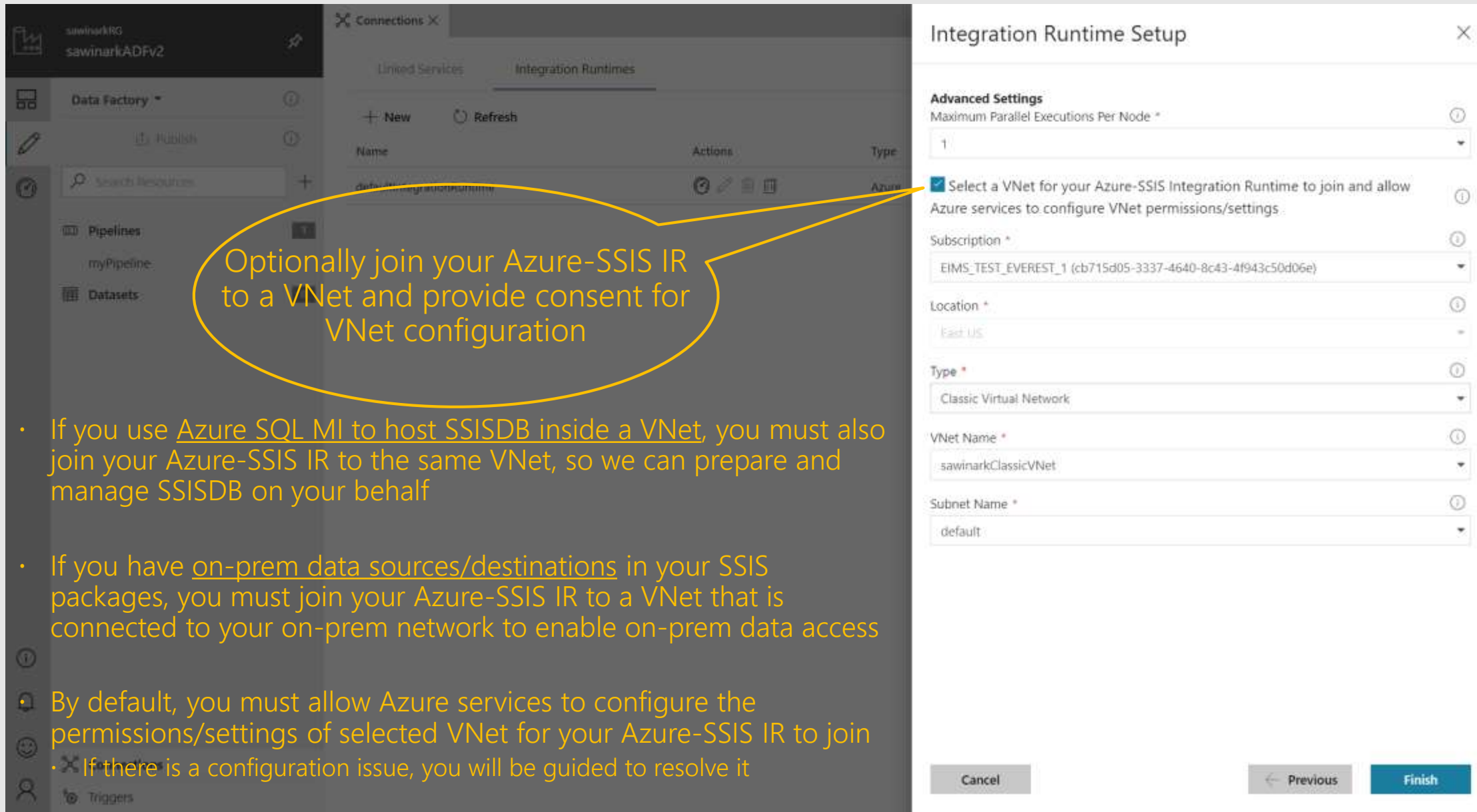
1

☐ Select a VNet for your Azure-SSIS Integration Runtime to join and allow Azure services to configure VNet permissions/settings.

Cancel Previous Finish

- A range of 1-8 for the number of parallel executions per node is provided
- Select a low concurrency level, if you want to use more than one cores to run a single large/compute-intensive package
- Select a high concurrency level, if you want to run one or more small/light-weight packages in a single core

Provisioning via ADFv2 App



Integration Runtime Setup

Advanced Settings

Maximum Parallel Executions Per Node *

1

☒ Select a VNet for your Azure-SSIS Integration Runtime to join and allow Azure services to configure VNet permissions/settings

Subscription *

EIMS_TEST_EVEREST_1 (cb715d05-3337-4640-8c43-4f943c50d06e)

Location *

East US

Type *

Classic Virtual Network

VNet Name *

sawinarkClassicVNet

Subnet Name *

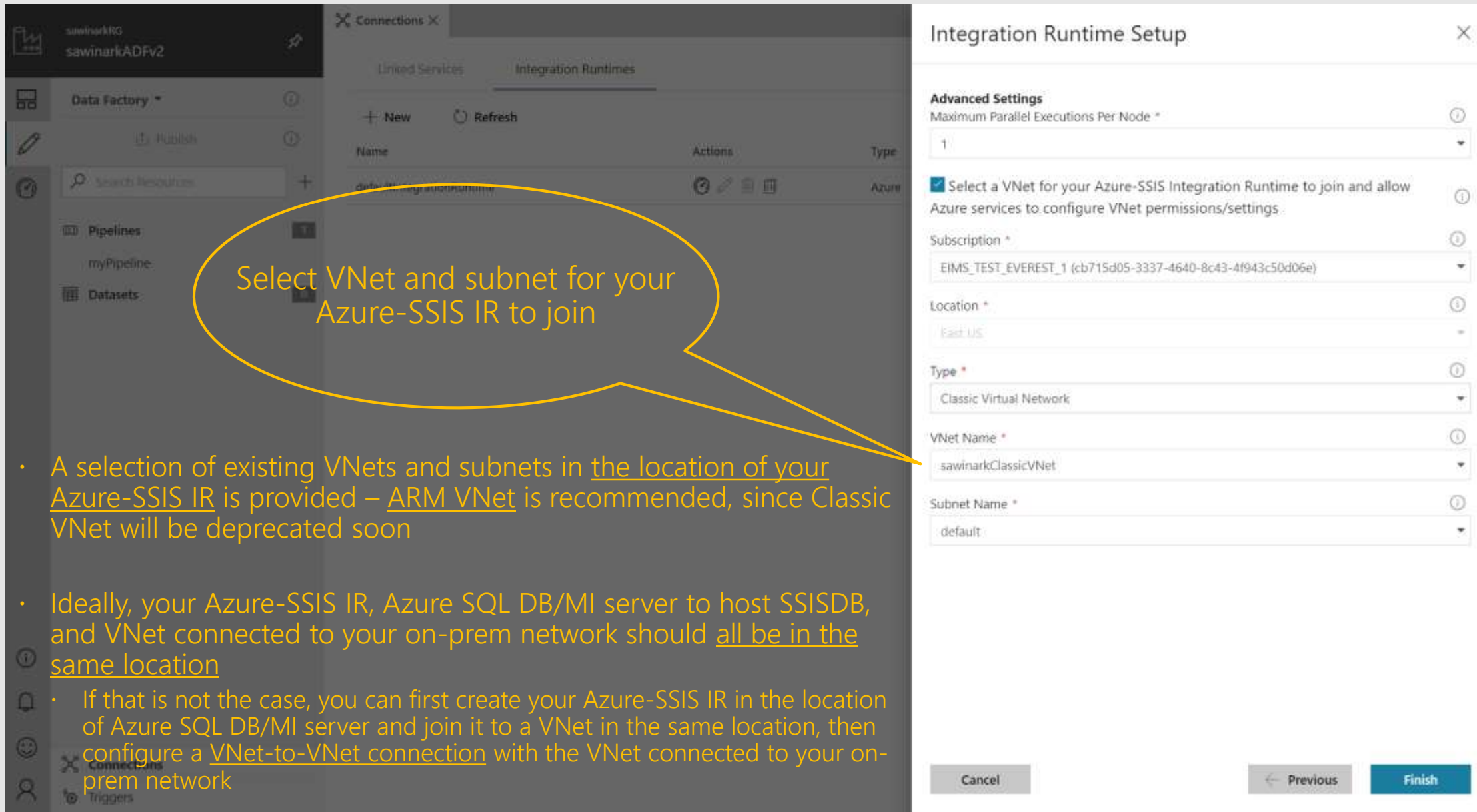
default

Cancel Previous Finish

Optionally join your Azure-SSIS IR to a VNet and provide consent for VNet configuration

- If you use Azure SQL MI to host SSISDB inside a VNet, you must also join your Azure-SSIS IR to the same VNet, so we can prepare and manage SSISDB on your behalf
- If you have on-prem data sources/destinations in your SSIS packages, you must join your Azure-SSIS IR to a VNet that is connected to your on-prem network to enable on-prem data access
- By default, you must allow Azure services to configure the permissions/settings of selected VNet for your Azure-SSIS IR to join
- If there is a configuration issue, you will be guided to resolve it

Provisioning via ADFv2 App



The screenshot displays the Azure Data Factory (ADFv2) console interface. On the left, the 'Data Factory' sidebar shows 'Pipelines', 'myPipeline', and 'Datasets'. The main area is titled 'Integration Runtimes' and includes a table with columns 'Name', 'Actions', and 'Type'. A yellow speech bubble points to the 'Select VNet and subnet for your Azure-SSIS IR to join' option in the 'Advanced Settings' section of the 'Integration Runtime Setup' dialog.

Integration Runtime Setup

Advanced Settings

Maximum Parallel Executions Per Node *

1

☒ Select a VNet for your Azure-SSIS Integration Runtime to join and allow Azure services to configure VNet permissions/settings

Subscription *

EIMS_TEST_EVEREST_1 (cb715d05-3337-4640-8c43-4f943c50d06e)

Location *

East US

Type *

Classic Virtual Network

VNet Name *

sawinarkClassicVNet

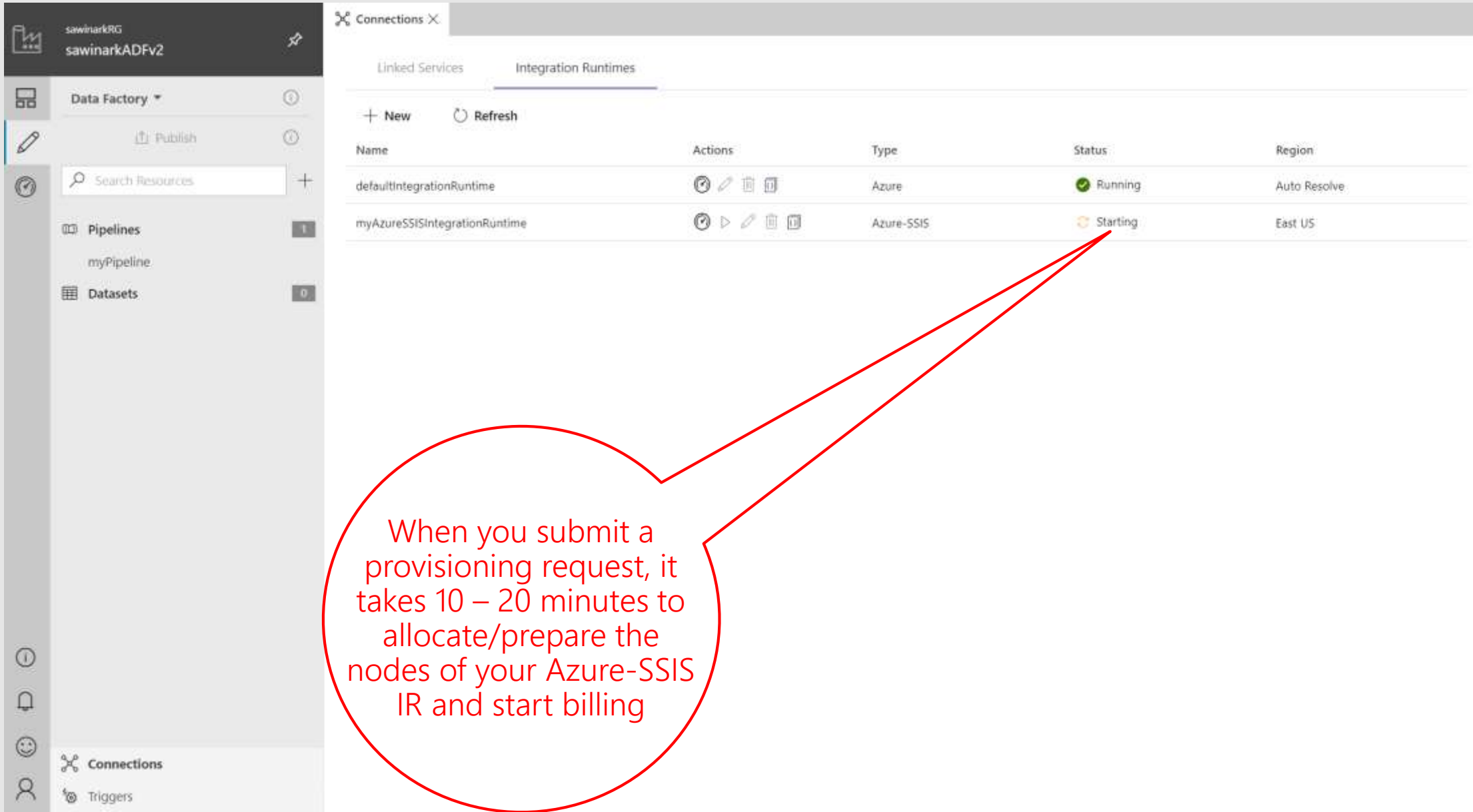
Subnet Name *

default










Cancel Previous Finish

- A selection of existing VNets and subnets in the location of your Azure-SSIS IR is provided – ARM VNet is recommended, since Classic VNet will be deprecated soon
- Ideally, your Azure-SSIS IR, Azure SQL DB/MI server to host SSISDB, and VNet connected to your on-prem network should all be in the same location
- If that is not the case, you can first create your Azure-SSIS IR in the location of Azure SQL DB/MI server and join it to a VNet in the same location, then configure a VNet-to-VNet connection with the VNet connected to your on-prem network

Provisioning via ADFv2 App



The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the navigation pane shows the 'Data Factory' view with options for 'Publish', 'Search Resources', 'Pipelines' (1 item), and 'Datasets' (0 items). The main pane is titled 'Connections' and has two tabs: 'Linked Services' and 'Integration Runtimes'. The 'Integration Runtimes' tab is active, showing a table of integration runtimes.

Name	Actions	Type	Status	Region
defaultIntegrationRuntime	   	Azure	Running	Auto Resolve
myAzureSSISIntegrationRuntime	    	Azure-SSIS	Starting	East US

A red callout bubble points to the 'Starting' status of the 'myAzureSSISIntegrationRuntime' with the following text:

When you submit a provisioning request, it takes 10 – 20 minutes to allocate/prepare the nodes of your Azure-SSIS IR and start billing

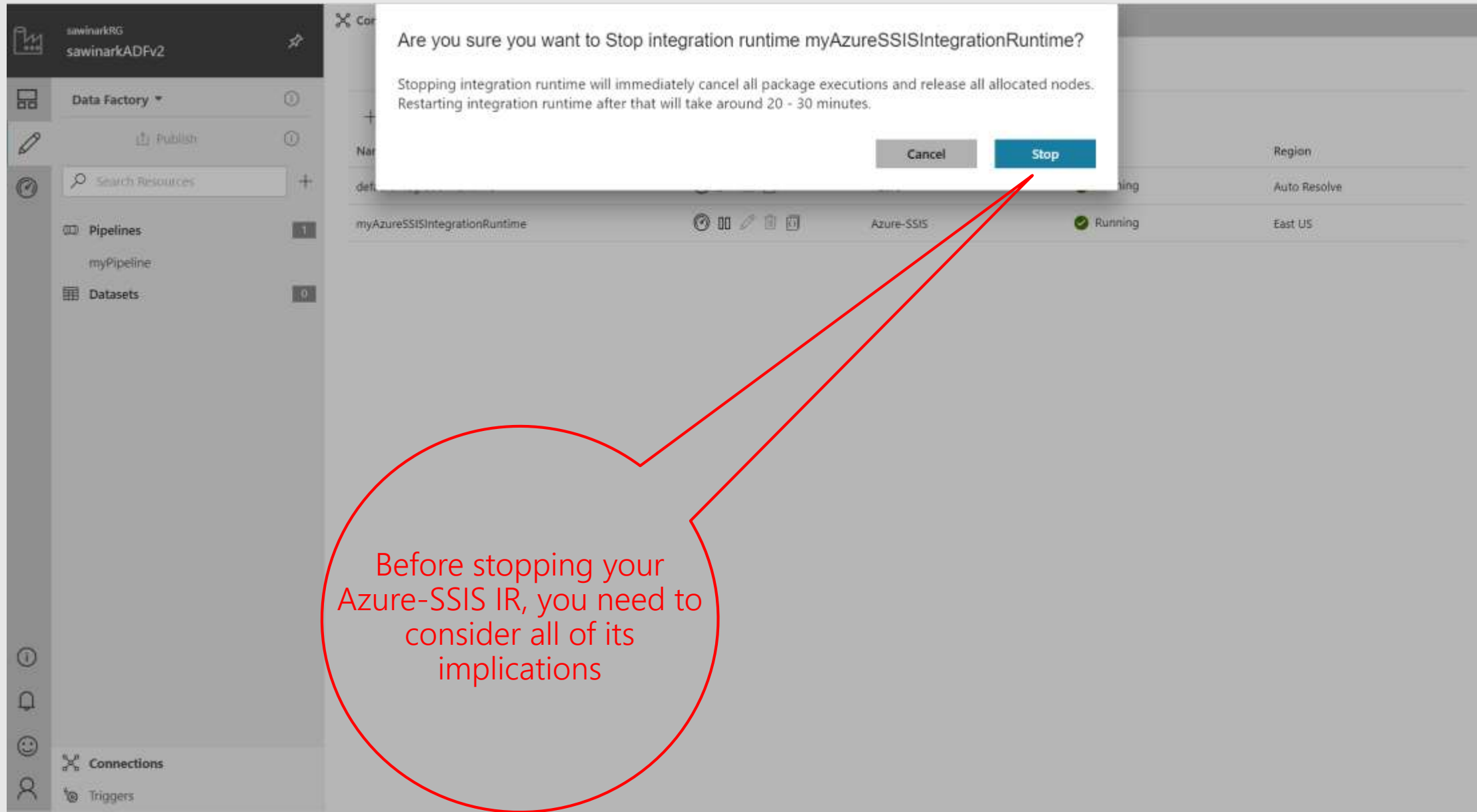
Provisioning via ADFv2 App

The screenshot shows the Azure Data Factory (ADFv2) interface. The left sidebar contains navigation options: 'Data Factory', 'Publish', 'Search Resources', 'Pipelines' (with 1 item), 'Datasets' (with 0 items), 'Connections', and 'Triggers'. The main area is titled 'Connections' and has two tabs: 'Linked Services' and 'Integration Runtimes'. The 'Integration Runtimes' tab is active, showing a table with the following data:

Name	Actions	Type	Status	Region
defaultIntegrationRuntime	[Stop] [Edit] [Refresh] [New]	Azure	Running	Auto Resolve
myAzureSSISIntegrationRuntime	[Stop] [Edit] [Refresh] [New]	Azure-SSIS	Running	East US

A red callout bubble points to the 'myAzureSSISIntegrationRuntime' row, containing the text: 'Once your Azure-SSIS IR is started, you can deploy SSIS packages to execute on it and you can stop it as you see fit'.

Reconfiguring via ADFv2 App



The screenshot shows the Azure Data Factory v2 interface. On the left, the navigation pane includes 'Data Factory', 'Publish', 'Search Resources', 'Pipelines' (with 1 item), 'myPipeline', 'Datasets' (with 0 items), 'Connections', and 'Triggers'. The main area displays the 'myAzureSSISIntegrationRuntime' resource, which is an 'Azure-SSIS' type, currently 'Running' in the 'East US' region. A confirmation dialog box is overlaid on the screen, asking 'Are you sure you want to Stop integration runtime myAzureSSISIntegrationRuntime?'. The dialog includes a warning: 'Stopping integration runtime will immediately cancel all package executions and release all allocated nodes. Restarting integration runtime after that will take around 20 - 30 minutes.' and two buttons: 'Cancel' and 'Stop'. A red circle with a line pointing to the 'Stop' button contains the text: 'Before stopping your Azure-SSIS IR, you need to consider all of its implications'.

Are you sure you want to Stop integration runtime myAzureSSISIntegrationRuntime?

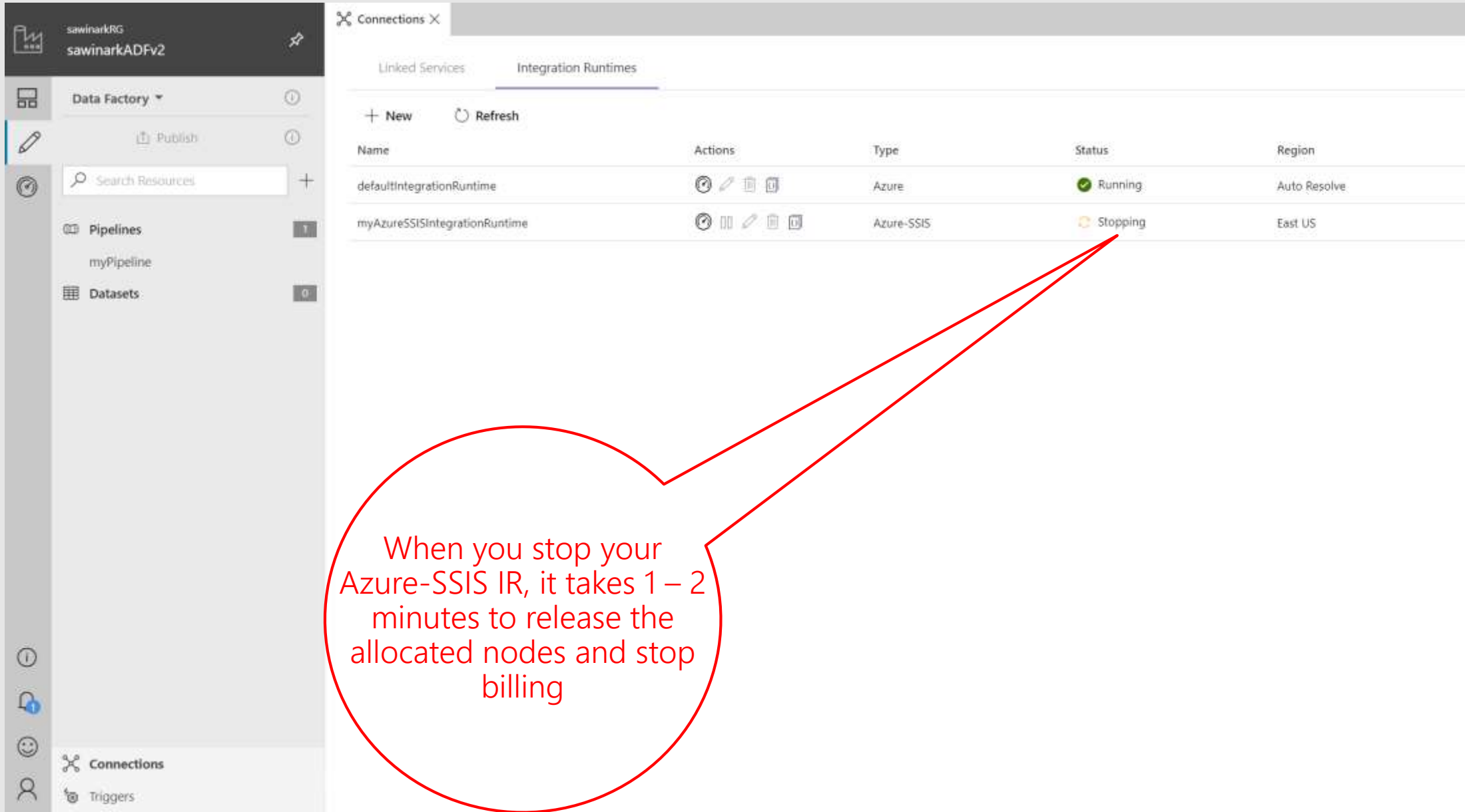
Stopping integration runtime will immediately cancel all package executions and release all allocated nodes. Restarting integration runtime after that will take around 20 - 30 minutes.

Cancel Stop

myAzureSSISIntegrationRuntime Azure-SSIS Running East US

Before stopping your Azure-SSIS IR, you need to consider all of its implications

Reconfiguring via ADFv2 App

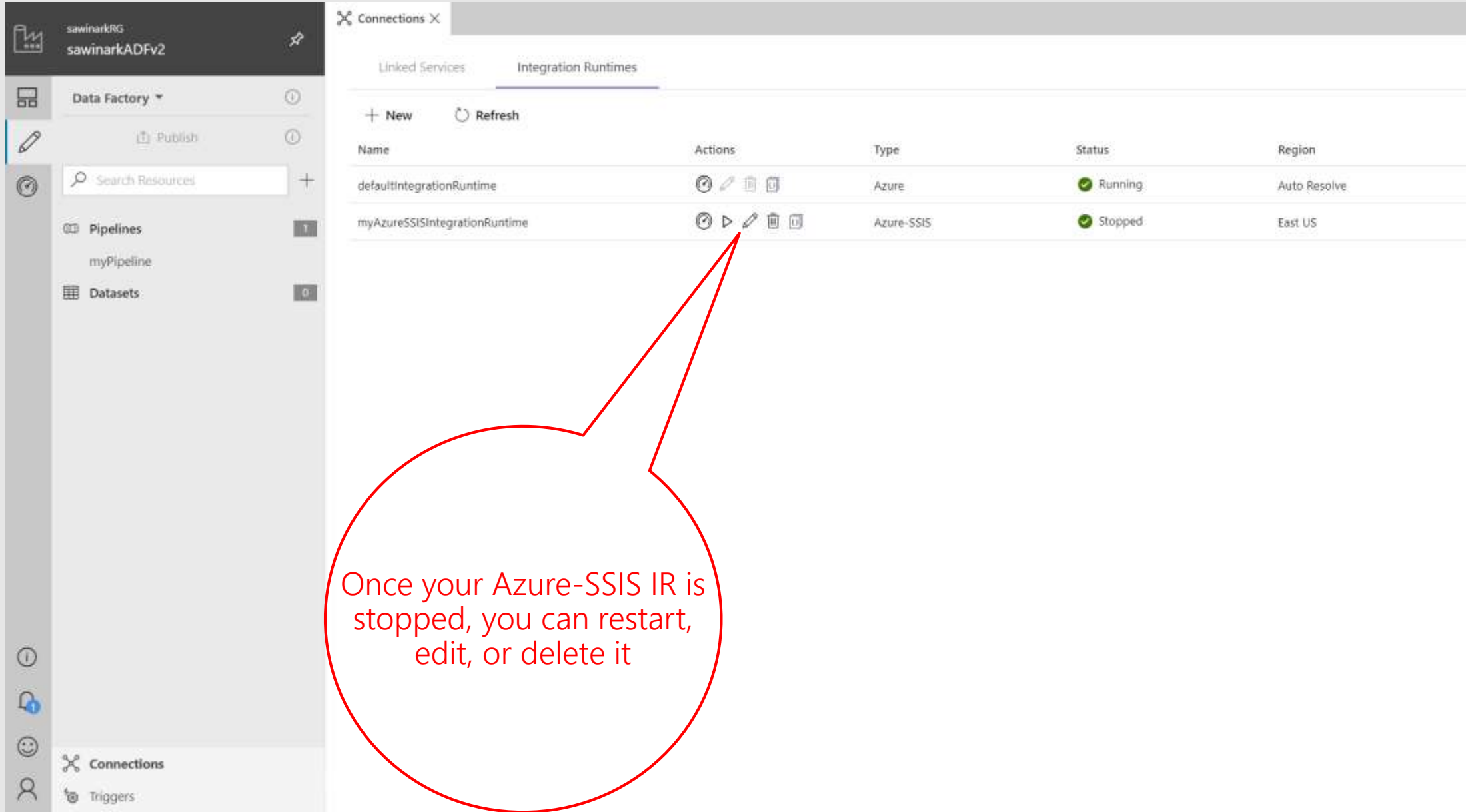


The screenshot shows the Azure Data Factory v2 interface. On the left is a sidebar with navigation icons and labels: 'Data Factory', 'Publish', 'Search Resources', 'Pipelines' (with 'myPipeline' listed), 'Datasets', 'Connections', and 'Triggers'. The main area is titled 'Connections' and has two tabs: 'Linked Services' and 'Integration Runtimes'. The 'Integration Runtimes' tab is active, displaying a table with the following data:










Name	Actions	Type	Status	Region
defaultIntegrationRuntime	[Icons: Stop, Edit, Delete, Refresh]	Azure	Running	Auto Resolve
myAzureSSISIntegrationRuntime	[Icons: Stop, Edit, Delete, Refresh]	Azure-SSIS	Stopping	East US

A red circle highlights the 'Stopping' status of the 'myAzureSSISIntegrationRuntime'. A red arrow points from this status to a text box that reads: 'When you stop your Azure-SSIS IR, it takes 1 – 2 minutes to release the allocated nodes and stop billing'.

Reconfiguring via ADFv2 App



The screenshot shows the Azure Data Factory v2 interface. On the left is a sidebar with navigation icons for Data Factory, Publish, Search Resources, Pipelines, Datasets, Connections, and Triggers. The main area is titled 'Connections' and has two tabs: 'Linked Services' and 'Integration Runtimes'. The 'Integration Runtimes' tab is active, showing a table of runtimes. A red circle highlights the 'myAzureSSISIntegrationRuntime' row, with a red arrow pointing to the 'Actions' column. A red callout bubble contains text explaining that once the Azure-SSIS IR is stopped, it can be restarted, edited, or deleted.

Name	Actions	Type	Status	Region
defaultIntegrationRuntime	   	Azure	Running	Auto Resolve
myAzureSSISIntegrationRuntime	    	Azure-SSIS	Stopped	East US

Once your Azure-SSIS IR is stopped, you can restart, edit, or delete it

Reconfiguring via ADFv2 App

The screenshot displays the Azure Data Factory v2 interface. On the left, the navigation pane shows 'Data Factory', 'Pipelines', and 'Datasets'. The main area is titled 'Connections' and 'Integration Runtimes'. A table lists the integration runtimes:

Name	Actions	Type
defaultIntegrationRuntime	[Icons]	Azure
myAzureSSISIntegrationRuntime	[Icons]	Azure-SSIS

The 'Integration Runtime Setup' dialog is open for 'myAzureSSISIntegrationRuntime'. It includes the following settings:

- General Settings**
- Name ***: myAzureSSISIntegrationRuntime
- Description**: [Empty field]
- Location ***: East US
- Node Size ***: Standard_D3_v2 (4 Core(s), 14336 MB)
- Node Number ***: 3 (indicated by a slider)
- Edition/License ***: Standard

At the bottom of the dialog are 'Cancel' and 'Next →' buttons.

When you edit your Azure-SSIS IR, only certain properties are editable, and when you finish, you can restart it

Provisioning via PSH

```
##### SSIS in ADFv2 specifications (please refer to SSIS in ADFv2 Public Preview documentation for field descriptions)
#####
# If your inputs contain PSH special characters, e.g. "$", please precede it with the escape character "`" like "`$".
$SubscriptionName = "[your Azure Subscription name]"
$ResourceGroupName = "[your Azure Resource Group name]"

# ADFv2 info
$DataFactoryName = "[your ADFv2 name]"
$DataFactoryLocation = "EastUS" # In Public Preview, only EastUS|EastUS2|WestEurope are supported for now

# Azure-SSIS Integration Runtime info - This is ADFv2 compute resource for running SSIS packages
$AzureSSISName = "[your Azure-SSIS Integration Runtime name]"
$AzureSSISDescription = "This is my Azure-SSIS Integration Runtime"
$AzureSSISLocation = "EastUS" # In Public Preview, only EastUS|EastUS2|CentralUS|NorthEurope|WestEurope|AustraliaEast
are supported for now
$AzureSSISNodeSize = "Standard_A4_v2" # In Public Preview, only
Standard_A4_v2|Standard_A8_v2|Standard_D1_v2|Standard_D2_v2|Standard_D3_v2|Standard_D4_v2 are supported for now
$AzureSSISNodeNumber = 2 # In Public Preview, only 1-10 nodes are supported for now
$AzureSSISEdition = "Standard" # Standard Edition is in Public Preview, Enterprise Edition is in Private Preview
$AzureSSISMaxParallelExecutionsPerNode = 4 # In Public Preview, only 1-8 parallel executions per node are supported for
now
```

Provisioning via PSH

Custom setup info

```
$SetupScriptContainerSasUri = "[SAS URI of your Azure Storage blob container for custom setup or leave it empty]" # Custom setup interface is in Private Preview
```

VNet info

```
$VnetId = "[your VNet resource ID or leave it empty]" # OPTIONAL: ARM VNet is recommended, since Classic VNet will be deprecated soon
```

```
#$VnetId = "/subscriptions/your Azure Subscription ID/resourceGroups/your Azure Resource Group name/providers/Microsoft.Network/virtualNetworks/your ARM VNet name" # ARM VNet example
```

```
#$VnetId = "/subscriptions/your Azure Subscription ID/resourceGroups/your Azure Resource Group name/providers/Microsoft.ClassicNetwork/virtualNetworks/your Classic VNet name" # Classic VNet example
```

```
$SubnetName = "[your subnet name or leave it empty]" # OPTIONAL: ARM VNet is recommended, since Classic VNet will be deprecated soon
```

SSISDB info

```
$SSISDBServerEndpoint = "[your Azure SQL Database server name.database.windows.net or your Azure SQL Managed Instance server endpoint]"
```

```
$SSISDBServerAdminUserName = "[your server admin username]"
```

```
$SSISDBServerAdminPassword = "[your server admin password]"
```

```
$SSISDBPricingTier = "[your Azure SQL Database pricing tier, e.g. S3, or leave it empty for Azure SQL Managed Instance]"
```

```
# Not applicable for Azure SQL Managed Instance
```

```
##### End of SSIS in ADFv2 specifications #####
```

Provisioning via PSH

```
##### Validate your Azure SQL Database/Managed Instance server #####
```

```
$SSISDBConnectionString = "Data Source=" + $SSISDBServerEndpoint + ";User ID="+ $SSISDBServerAdminUserName  
+";Password="+ $SSISDBServerAdminPassword
```

```
$sqlConnection = New-Object System.Data.SqlClient.SqlConnection $SSISDBConnectionString;
```

```
Try
```

```
{
```

```
    $sqlConnection.Open();
```

```
}
```

```
Catch [System.Data.SqlClient.SqlException]
```

```
{
```

```
    Write-Warning "Cannot connect to your Azure SQL DB logical server/Azure SQL MI server, exception: $_" ;
```

```
    Write-Warning "Please make sure the server you specified has already been created. Do you want to proceed? [Y/N]"
```

```
    $yn = Read-Host
```

```
    if(!($yn -ieq "Y"))
```

```
    {
```

```
        Return;
```

```
    }
```

```
}
```

```
##### Login and select your Azure subscription #####
```

```
Login-AzureRmAccount
```

```
Select-AzureRmSubscription -SubscriptionName $SubscriptionName
```

Provisioning via PSH

```
##### Automatically configure VNet permissions/settings for your Azure SSIS Integration Runtime to join
#####
# Register to Azure Batch resource provider
if(![string]::IsNullOrEmpty($VnetId) -and ![string]::IsNullOrEmpty($SubnetName))
{
    Register-AzureRmResourceProvider -ProviderNamespace Microsoft.Batch
    while(!(Get-AzureRmResourceProvider -ProviderNamespace
"Microsoft.Batch").RegistrationState.Contains("Registered"))
    {
        Start-Sleep -s 10
    }
    if($VnetId -match "/providers/Microsoft.ClassicNetwork/")
    {
        # Assign VM contributor role to Azure Batch
        $BatchObjectId = (Get-AzureRmADServicePrincipal -ServicePrincipalName "MicrosoftAzureBatch").Id
        New-AzureRmRoleAssignment -ObjectId $BatchObjectId -RoleDefinitionName "Classic Virtual Machine
Contributor" -Scope $VnetId
    }
}
```

Provisioning via PSH

```
##### Provision your ADFv2 + Azure SSIS Integration Runtime #####
```

```
New-AzureRmResourceGroup -Location $DataFactoryLocation -Name $ResourceGroupName
```

```
Register-AzureRmResourceProvider -ProviderNamespace Microsoft.DataFactory
```

```
$secpasswd = ConvertTo-SecureString $SSISDBServerAdminPassword -AsPlainText -Force
```

```
$serverCreds = New-Object
```

```
System.Management.Automation.PSCredential($SSISDBServerAdminUserName, $secpasswd)
```

```
Set-AzureRmDataFactoryV2 -ResourceGroupName $ResourceGroupName `
```

```
-Location $DataFactoryLocation `
```

```
-Name $DataFactoryName
```

Provisioning via PSH

```
Set-AzureRmDataFactoryV2IntegrationRuntime -ResourceGroupName $ResourceGroupName `
    -DataFactoryName $DataFactoryName `
    -Type Managed `
    -Name $AzureSSISName `
    -Description $AzureSSISDescription `
    -Location $AzureSSISLocation `
    -NodeSize $AzureSSISNodeSize `
    -NodeCount $AzureSSISNodeNumber `
    -Edition $AzureSSISEdition `
    -MaxParallelExecutionsPerNode $AzureSSISMaxParallelExecutionsPerNode `
    -SetupScriptContainerSasUri $SetupScriptContainerSasUri `
    -VnetId $VnetId `
    -Subnet $SubnetName `
    -CatalogServerEndpoint $SSISDBServerEndpoint `
    -CatalogAdminCredential $serverCreds `
    -CatalogPricingTier $SSISDBPricingTier

write-host("##### Starting #####")
Start-AzureRmDataFactoryV2IntegrationRuntime -ResourceGroupName $ResourceGroupName `
    -DataFactoryName $DataFactoryName `
    -Name $AzureSSISName `
    -Force

write-host("##### Completed #####")
write-host("If any cmdlet is unsuccessful, please consider using -Debug option for diagnostics.")
```

Reconfiguring via PSH

```
##### Reconfigure your Azure SSIS Integration Runtime, e.g. stopping/scaling out to 5 nodes/restarting  
#####
```

```
# Stopping your Azure-SSIS Integration Runtime will release its nodes and stop billing
```

```
#Stop-AzureRmDataFactoryV2IntegrationRuntime -DataFactoryName $DataFactoryName -Name $AzureSSISName -  
ResourceGroupName $ResourceGroupName
```

```
# Scaling out your Azure-SSIS Integration Runtime to 5 nodes
```

```
#Set-AzureRmDataFactoryV2IntegrationRuntime -DataFactoryName $DataFactoryName -Name $AzureSSISName -  
ResourceGroupName $ResourceGroupName -NodeCount 5
```

```
# Starting your Azure-SSIS Integration Runtime will allocate its nodes and start billing
```

```
#Start-AzureRmDataFactoryV2IntegrationRuntime -DataFactoryName $DataFactoryName -Name $AzureSSISName -  
ResourceGroupName $ResourceGroupName
```

```
##### Clean up #####
```

```
#Stop-AzureRmDataFactoryV2IntegrationRuntime -DataFactoryName $DataFactoryName -Name $AzureSSISName -  
ResourceGroupName $ResourceGroupName -Force
```

```
#Remove-AzureRmDataFactoryV2IntegrationRuntime -DataFactoryName $DataFactoryName -Name $AzureSSISName -  
ResourceGroupName $ResourceGroupName -Force
```

```
#Remove-AzureRmDataFactoryV2 -Name $DataFactoryName -ResourceGroupName $ResourceGroupName -Force
```

```
#Remove-AzureRmResourceGroup -Name $ResourceGroupName -Force
```


Deployment Methods

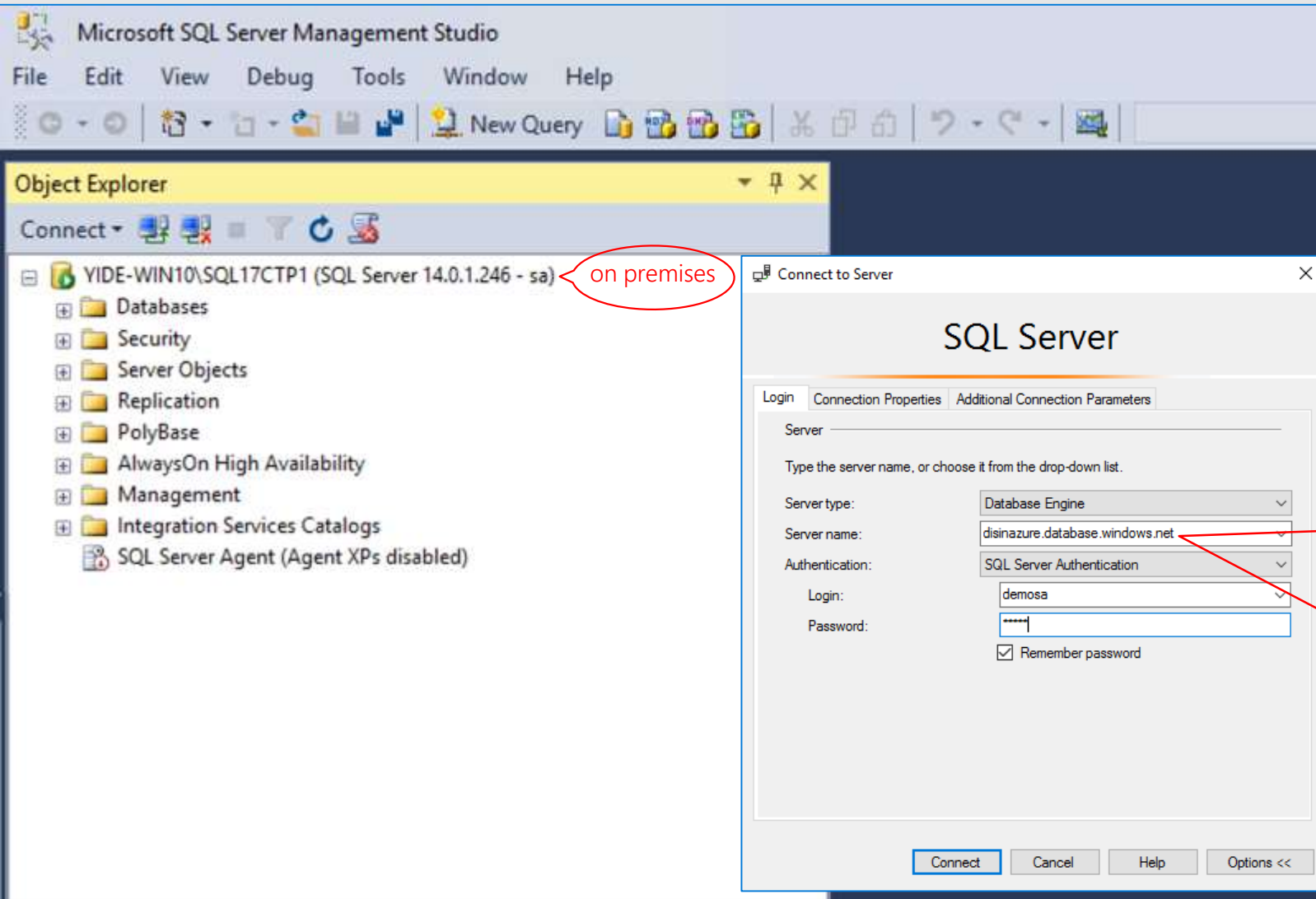
Deployment Methods

- SSIS PaaS supports the project deployment model used in SSIS 2012/later versions
 - Projects built in the legacy package deployment model used in SSIS 2008/earlier versions can be converted into this model via SSDT/SSMS using Integration Services Project Conversion Wizard
 - Packages built in SSIS 2008/earlier versions can be upgraded to the latest version supported by SSIS PaaS via SSDT/SSMS using SSIS Package Upgrade Wizard
 - In this model, the whole project needs to be deployed after any package changes – An incremental package deployment feature will be provided in the near future
 - Projects containing environment references/run-time parameters can be saved into project deployment files (.ispac extension)
 - Projects are deployed into SSISDB hosted by Azure SQL DB/MI server, packages are run by creating/starting jobs via SSISDB sprocs that will be executed on Azure-SSIS IR, and execution logs are written back into SSISDB

Deployment Methods

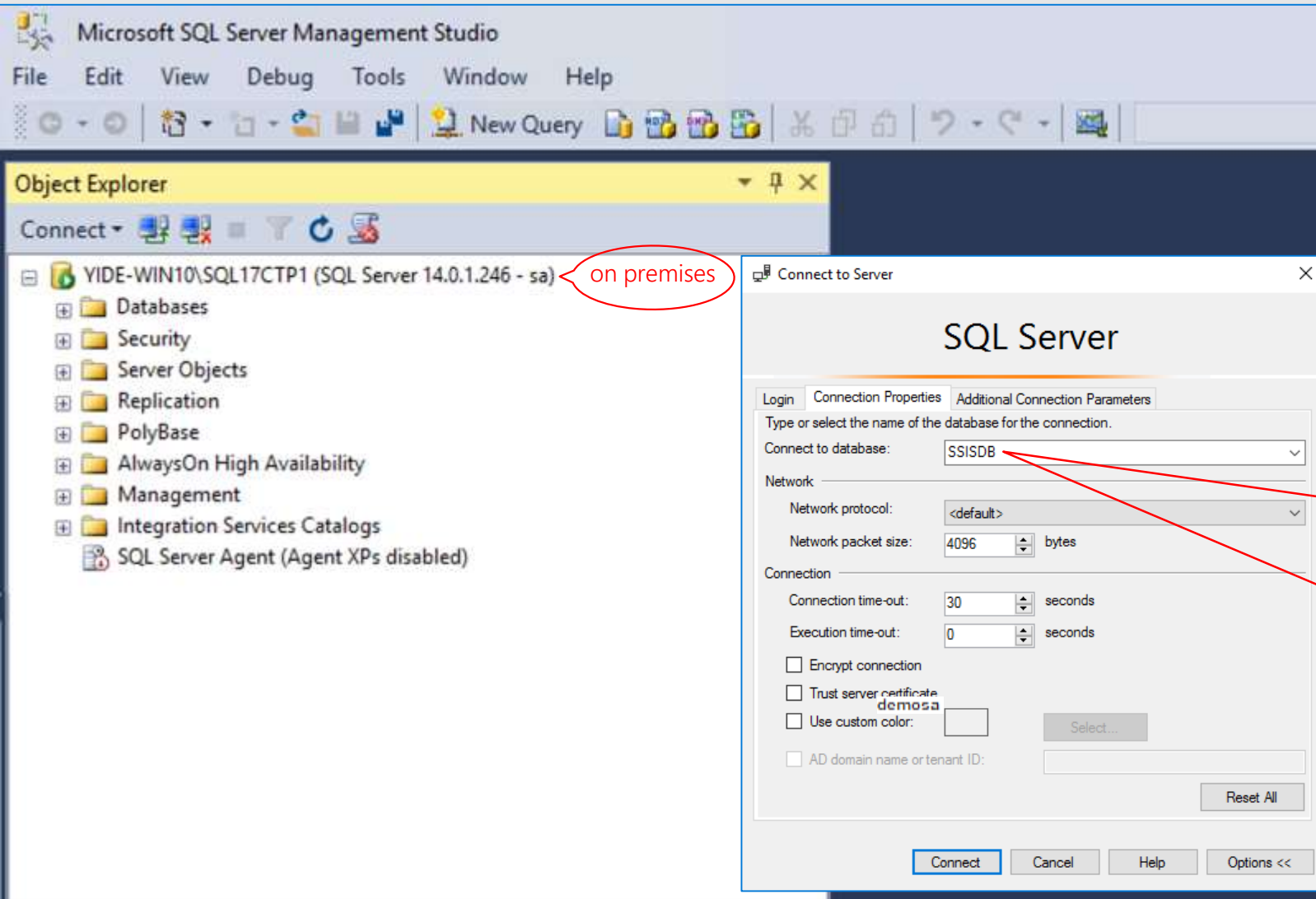
- SSIS projects can be deployed via SSDT/SSMS using Integration Services Deployment Wizard
- SSIS projects can be deployed via Command Line Interface (CLI)
 - Run isdeploymentwizard.exe from the command prompt (TBD)
- SSIS projects can be deployed via PSH/custom code using SSIS Managed Object Model (MOM) .NET SDK/API
 - Microsoft.SqlServer.Management.IntegrationServices.dll is installed in .NET Global Assembly Cache (GAC) with SQL Server/SSMS installation
- SSIS projects can be deployed via T-SQL scripts executing SSISDB sprocs
 - Execute SSISDB sproc [catalog].[deploy_project]

Deployment via SSMS



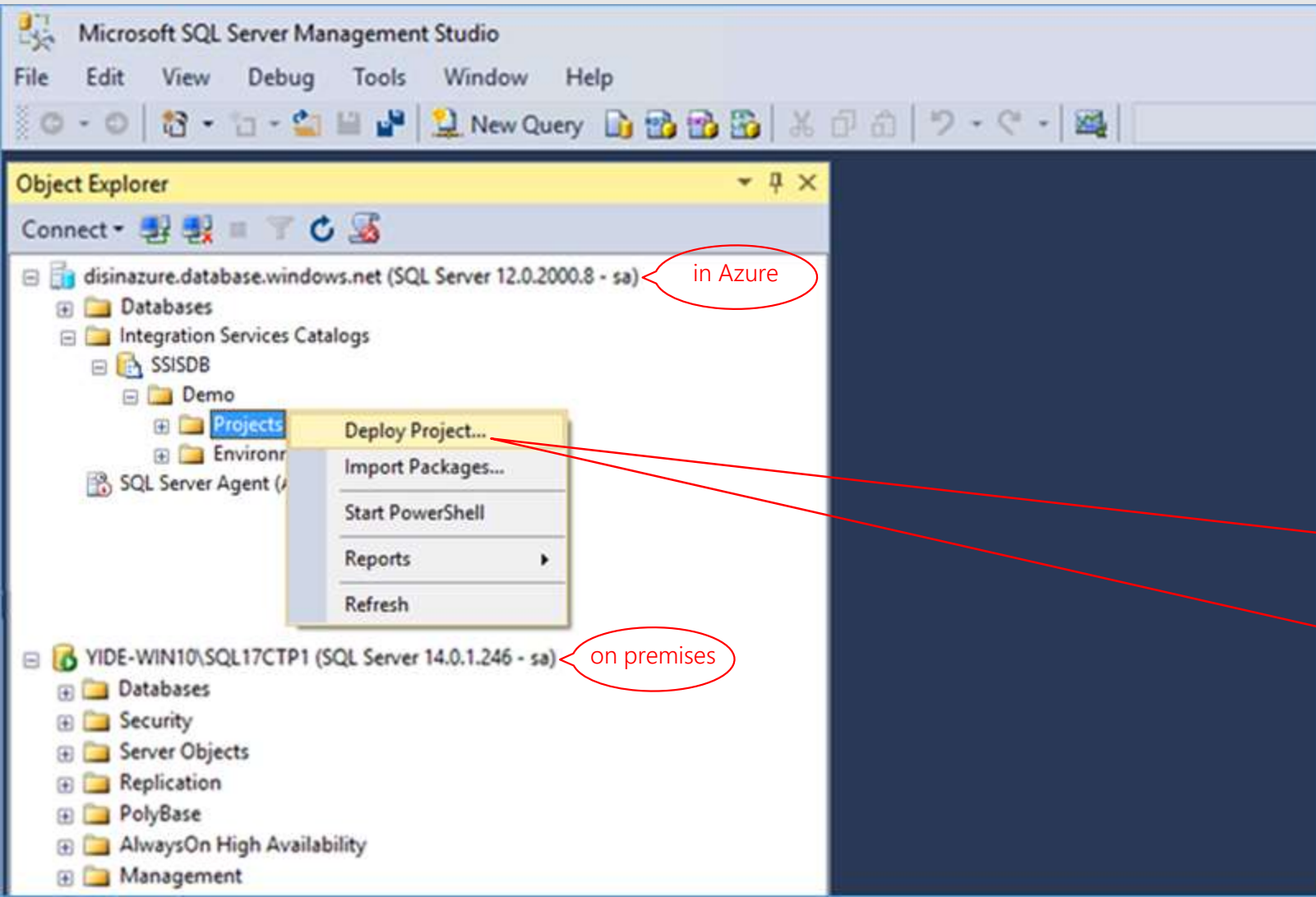
On SSMS, you can connect to SSIS PaaS using its connection info and SQL/AAD authentication credentials

Deployment via SSMS



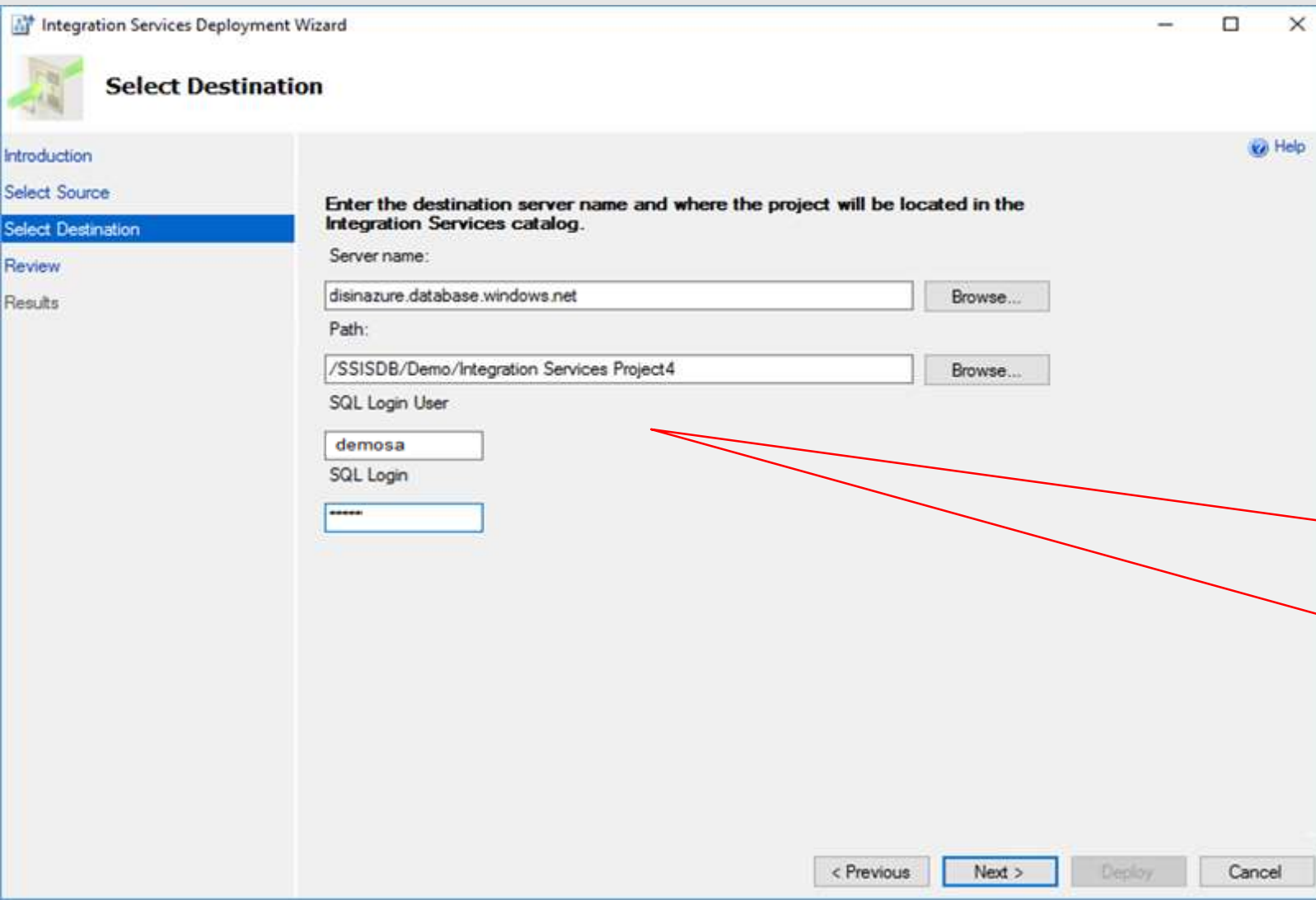
On "Connection Properties" tab, enter "SSISDB" as the target database to connect

Deployment via SSMS



Once connected, you can deploy projects/packages to SSIS PaaS from your local file system/SSIS on premises

Deployment via SSMS



The screenshot shows the 'Integration Services Deployment Wizard' window, specifically the 'Select Destination' step. The window has a title bar with standard Windows controls. On the left is a navigation pane with links: 'Introduction', 'Select Source', 'Select Destination' (highlighted), 'Review', and 'Results'. The main area contains the following fields and instructions:

- Instructions:** Enter the destination server name and where the project will be located in the Integration Services catalog.
- Server name:** A text box containing 'disinazure.database.windows.net' and a 'Browse...' button.
- Path:** A text box containing '/SSISDB/Demo/Integration Services Project4' and a 'Browse...' button.
- SQL Login User:** A text box containing 'demosa'.
- SQL Login:** A password text box with masked characters '*****'.

At the bottom of the wizard are four buttons: '< Previous', 'Next >' (highlighted with a blue border), 'Deploy', and 'Cancel'. A red callout bubble points to the 'SQL Login User' and 'SQL Login' fields.

On Integration Services Deployment Wizard, enter SSIS PaaS connection info and SQL authentication credentials

Execution Methods

Execution Methods

- SSIS packages can be directly executed as first-class SSIS activities in ADFv2 pipelines (Work in Progress)
 - For now, SSIS packages can be indirectly executed via ADFv2 Sproc Activity
- SSIS packages can be executed via SSMS
- SSIS packages can be executed via CLI
 - Run dtexec.exe from the command prompt (TBD)
- SSIS packages can be executed via PSH/custom code using SSIS MOM .NET SDK/API
 - Microsoft.SqlServer.Management.IntegrationServices.dll is installed in .NET GAC with SQL Server/SSMS installation
- SSIS packages can be executed via T-SQL scripts executing SSISDB sprocs
 - Execute SSISDB sprocs [catalog].[create execution] + [catalog].[set execution parameter value] + [catalog].[start execution]

Execution via ADFv2 Sproc Activity

- Create a linked service for Azure SQL DB/MI server hosting SSISDB
- Create a pipeline with SqlServerStoredProcedure activity
- Trigger/execute the pipeline on demand

Execution via ADFv2 Sproc Activity

The screenshot shows the Azure Data Factory (ADFv2) interface. On the left, the 'Connections' tab is selected in the sidebar. The main pane shows the 'Linked Services' tab. A table lists the linked services:

Name	Actions	Type
myLinkedService		Azure SQL Database

A red circle highlights the text 'Click on "Connections" and "Linked Services"' with a red arrow pointing to the '+ New' button in the 'Linked Services' tab.

Execution via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the 'Data Factory' sidebar shows the 'Connections' tab selected. The main area shows the 'Linked Services' tab with a table listing existing services. A yellow callout points to the '+ New' button and the 'Azure SQL Database' option in the 'New Linked Service' dialog.

Connections X

Linked Services | Integration Runtime

+ New

Name	Actions	Type
myLinkedService		Azure SQL Database

New Linked Service

Data Store | Compute

Search

 Azure File Storage	 Azure Key Vault	 Azure SQL Data Warehouse
 Azure SQL Database	 Azure Search	 Azure Table Storage
 Cassandra	 Concur (Beta)	 Couchbase (Beta)

Cancel | Continue

Click on "+ New" and "Azure SQL Database"

Execution via ADFv2 Sproc Activity

The screenshot shows the 'New Linked Service' dialog in the Azure Data Factory portal. The dialog is titled 'New Linked Service' and contains several input fields for configuring a linked service. A yellow callout circle is drawn around the 'Name' and 'Description' fields, with the text 'Name and describe your linked service' inside it. The 'Name' field is filled with 'myLinkedService2'. The 'Description' field is empty. The 'Connect via integration runtime' dropdown is set to 'Default'. The 'Account selection method' dropdown is set to 'From Azure subscription'. The 'Azure subscription' dropdown is set to 'EIMS_TEST_EVEREST_1 (cb715d05-3337-4640-8c43-4f943c50d06e)'. The 'Server name' dropdown is set to 'disinazure'. The 'Database name' dropdown is set to 'SSISDB'. The 'User name' field is filled with 'demosa'. The 'Password' field is filled with '*****'. At the bottom of the dialog, there are three buttons: 'Cancel', 'Test connection', and 'Save'. The 'Connections' tab is selected in the background, showing a table with one entry: 'myLinkedService' of type 'Azure SQL Database'.

Connections X

Linked Services Integration Runtimes

+ New

Name	Actions	Type
myLinkedService		Azure SQL Database

New Linked Service

Name *

myLinkedService2

Description

Connect via integration runtime *

Default

[New Integration Runtime](#)

Account selection method

From Azure subscription

Azure subscription

EIMS_TEST_EVEREST_1 (cb715d05-3337-4640-8c43-4f943c50d06e)

Server name *

disinazure

Database name *

SSISDB

User name *

demosa

Password *

Advanced ⓘ

Cancel Test connection Save

Name and describe your linked service

Execution via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory interface. On the left, the 'Connections' pane shows the 'myLinkedService' table. The main area shows the 'New Linked Service' dialog box. The dialog has the following fields:

- Name ***: myLinkedService2
- Description**: (empty)
- Connect via integration runtime ***: Default (selected)
- New Integration Runtime**: (link)
- Account selection method**: From Azure subscription
- Azure subscription**: EIMS_TEST_EVEREST_1 (cb715d05-3337-4640-8c43-4f943c50d06e)
- Server name ***: disinazure
- Database name ***: SSISDB
- User name ***: demosa
- Password ***: (masked with dots)
- Advanced**: (expandable section)
- Buttons**: Cancel, Test connection, Save

A yellow callout bubble points to the 'Connect via integration runtime' dropdown, containing the text: "Select ADFv2 Default IR to connect to your Azure SQL DB that is attached to Azure-SSIS IR".

Execution via ADFv2 Sproc Activity

The screenshot shows the 'New Linked Service' dialog in the Azure Data Factory portal. The dialog is for creating a new linked service of type 'Azure SQL Database'. A yellow callout bubble points to the 'Server name' and 'Database name' fields, with the text 'Select your Azure SQL DB server and SSISDB to connect'. The 'Server name' field is set to 'disinazure' and the 'Database name' field is set to 'SSISDB'. Other fields include 'Name' (myLinkedService2), 'Description' (empty), 'Connect via integration runtime' (Default), 'Account selection method' (From Azure subscription), 'Azure subscription' (EIMS_TEST_EVEREST_1), 'User name' (demosa), and 'Password' (masked with asterisks). The 'Advanced' section is collapsed. The 'Test connection' and 'Save' buttons are visible at the bottom right.

Connections

Linked Services Integration Runtimes

+ New

Name	Actions	Type
myLinkedService		Azure SQL Database

New Linked Service

Name *
myLinkedService2

Description

Connect via integration runtime *
Default

[New Integration Runtime](#)

Account selection method
From Azure subscription

Azure subscription
EIMS_TEST_EVEREST_1 (cb715d05-3337-4640-8c43-4f943c50d06e)

Server name *
disinazure

Database name *
SSISDB

User name *
demosa

Password *

Advanced ⓘ

Cancel Test connection Save

Select your Azure SQL DB server and SSISDB to connect

Execution via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) 'Connections' pane. The 'Linked Services' tab is active, showing a table with one entry: 'myLinkedService' of type 'Azure SQL Database'. A 'New Linked Service' dialog is open on the right, with the following fields and values:

- Name: myLinkedService2
- Description: (empty)
- Connect via integration runtime: Default
- Account selection method: From Azure subscription
- Azure subscription: EIMS_TEST_EVEREST_1 (cb715d05-3337-4640-8c43-4f943c50d06e)
- Server name: disinazure
- Database name: SSISDB
- User name: demosa
- Password: (masked with dots)

A yellow callout bubble points to the 'Password' field with the text: "Enter your Azure SQL DB server admin credentials".

At the bottom of the dialog, there are buttons for 'Cancel', 'Test connection', and 'Save'. An 'Advanced' section is also visible at the bottom left of the dialog.

Execution via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the navigation pane shows the 'Data Factory' tab with options for 'Publish', 'Search Resources', 'Pipelines', and 'Datasets'. The main area is titled 'Connections' and shows a table of linked services. A table with one row is visible:

Name	Actions	Type
myLinkedService	[Edit] [Delete] [Refresh]	Azure SQL Database

Overlaid on the right is the 'New Linked Service' dialog box. It contains the following fields and options:

- Name ***: myLinkedService2
- Description**: (empty text area)
- Connect via integration runtime ***: Default (dropdown menu)
- Account selection method**: From Azure subscription (dropdown menu)
- Azure subscription**: EIMS_TEST_EVEREST_1 (cb715d05-3337-4640-8c43-4f943c50d06e) (dropdown menu)
- Server name ***: disinazure (dropdown menu)
- Database name ***: SSISDB (dropdown menu)
- User name ***: demosa
- Password ***: (masked with dots)

At the bottom of the dialog, there is a green checkmark icon and the text 'Connection successful'. Below this, there are three buttons: 'Cancel', 'Test connection', and 'Save'. A yellow callout bubble originates from the 'Test connection' button and points to the text 'Test connection and save your linked service'.

Execution via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the 'Data Factory' sidebar is visible, featuring a search bar and a dropdown menu with options: 'Pipelines', 'myPipeline', 'Datasets', and 'Copy Data'. The main area shows the 'Connections' tab, which is divided into 'Linked Services' and 'Integration Runtimes'. The 'Linked Services' section contains a table with the following data:

Name	Actions	Type
myLinkedService	[Edit] [Delete] [Refresh]	Azure SQL Database
myLinkedService2	[Edit] [Delete] [Refresh]	Azure SQL Database

A red circle highlights the '+ New' button and the 'Pipeline' option in the dropdown menu. A red arrow points from the circle to the table of linked services.

Click on "+" and "Pipeline"

Execution via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) user interface. On the left, the 'Data Factory' sidebar shows the 'Pipelines' section with two items: 'myPipeline' and '* myPipeline2'. The main workspace is titled 'myPipeline2' and contains a search bar and a list of activity categories: Batch Service, Data Flow, Data Lake Analytics, General, HDInsight, and Iteration & Conditionals. Below this, the 'General' tab is selected, showing the 'Name' field set to 'myPipeline2' and an empty 'Description' field. A red circle highlights the 'Name' and 'Description' fields, with a red arrow pointing to it from the text 'Name and describe your pipeline'.

Connections X myPipeline2 X

Activities Search Activities

Batch Service

Data Flow

Data Lake Analytics

General

HDInsight

Iteration & Conditionals

General Parameters Output

Name * myPipeline2

Description

Name and describe your pipeline

Execution via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the 'Data Factory' sidebar shows the 'Pipelines' section with 'myPipeline2' selected. The 'Activities' pane on the right lists various activity types, including 'Execute Pipeline', 'Get Metadata', 'Lookup', 'Stored Procedure', and 'Web'. The 'Stored Procedure' activity is highlighted. In the main canvas, a 'Stored Procedure' activity named 'Stored Procedure1' is shown. Below the activity, the 'General' tab is active, showing the 'Name' field set to 'myPipeline2' and a 'Description' field. A red circle highlights the 'Stored Procedure' activity in the 'Activities' pane, and a red arrow points from it to the 'Stored Procedure1' activity in the canvas, indicating the drag-and-drop action.

Drag and drop "Stored Procedure" activity into your pipeline

Execution via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) user interface. On the left, the 'Data Factory' sidebar shows the 'Pipelines' section with a list of pipelines: 'myPipeline' and '* myPipeline2'. The 'Activities' pane on the right lists various activity types: 'Batch Service', 'Data Flow', 'Data Lake Analytics', 'General', 'HDInsight', and 'Iteration & Conditionals'. The 'Stored Procedure' activity is selected, and its configuration is shown in the main workspace. The 'General' tab is active, displaying the following fields:

- Name ***: MySprocActivity2
- Description**: (Empty text area)
- Timeout**: 7:00:00:00
- Retry**: 0
- Retry Interval**: 20

A red circle highlights the 'Name' and 'Description' fields, with a red arrow pointing to them from the text 'Name and describe your sproc activity'.

Execution via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the 'Data Factory' sidebar shows the 'Pipelines' section with 'myPipeline' and 'myPipeline2'. The 'myPipeline2' pipeline is selected, and the 'Stored Procedure' activity is highlighted in the 'Activities' pane. The main canvas shows the 'Stored Procedure' activity configuration for 'MySprocActivity2'. The 'SQL Account' tab is active, showing a 'Linked service' dropdown set to 'myLinkedService2'. A green checkmark indicates 'Connection successful'. Below the dropdown are buttons for 'Test connection', 'Edit', and '+ New'. A red circle highlights the 'Test connection' button, with a red arrow pointing to it from the text 'Select your linked service and test connection'.

Connections

myPipeline2

Activities

Search Activities

Batch Service

Data Flow

Data Lake Analytics

General

Execute Pipeline

Get Metadata

Lookup

Stored Procedure

Web

HDInsight

Iteration & Conditionals

Stored Procedure

MySprocActivity2

General

SQL Account

Stored Procedure 1

Parameters

Advanced

Connection successful

Linked service *

myLinkedService2

Test connection

Edit

+ New

Select your linked service and test connection

Execution via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the 'Data Factory' sidebar shows the 'Pipelines' section with 'myPipeline' and 'myPipeline2'. The 'Activities' pane on the right lists various activity types, including 'Execute Pipeline', 'Get Metadata', 'Lookup', 'Stored Procedure', and 'Web'. The 'Stored Procedure' activity is selected, and its configuration is shown in the 'Stored Procedure' tab. The 'Stored procedure name' is set to 'sp_executesql'. A red circle highlights this field, with a red arrow pointing to it from a text box that reads: 'Use system sproc "sp_executesql" to execute T-SQL script'. Below the 'Stored procedure name' field, there is an 'Import parameter' section with a table for 'Stored procedure parameters'. The table has columns for 'NAME', 'TYPE', and 'VALUE', and it currently shows 'No records found'.

Connections X myPipeline2 X

Activities Search Activities

Batch Service

Data Flow

Data Lake Analytics

General

Execute Pipeline

Get Metadata

Lookup

Stored Procedure

Web

HDInsight

Iteration & Conditionals

Validate Test Run Trigger

Code

Stored Procedure

MySprocActivity2

General SQL Account Stored Procedure Parameters Advanced

Details

Stored procedure name * sp_executesql Edit

Import parameter

Stored procedure parameters

+ New - Delete

NAME	TYPE	VALUE
No records found		

Execution via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the 'Data Factory' sidebar shows the 'myPipeline2' pipeline selected. The main canvas shows the 'Stored Procedure' activity configuration for 'MySprocActivity2'. The 'Stored procedure name' is set to 'sp_executesql'. Under 'Import parameter', a new parameter named 'stmt' of type 'string' has been added. The 'Value' field contains a T-SQL script to create and start an SSIS package execution.

Stored procedure name * ☒ Edit

Import parameter

Stored procedure parameters: 1

+ New | Delete

NAME	TYPE	VALUE
stmt	string	DECLARE @return_value INT, @exe_id BIGINT, @err_msg NVARCHAR(150) EXEC @return_value=[SSISDB].[catalog].[create_execution] @folder_name=N'demo', @project_name=N'ScaleOutProject', @package_name=N'SQLDBTable2toSQLDBTable2.dtsx', @use32bitruntime=0, @runinscaleout=1, @useanyworker=1, @execution_id=@exe_id OUT PUT EXEC [SSISDB].[catalog].[set_execution_parameter_value] @exe_id, @object_type=50, @parameter_name=N'SYNCHRONIZED', @parameter_value=1 EXEC [SSISDB].[catalog].[start_execution] @execution_id=@exe_id, @retry_count=0 IF(SELECT [status] FROM [SSISDB].[catalog].[executi

Add a sproc parameter "stmt" of type "string" with your T-SQL script to create/start SSIS package execution using SSISDB sprocs as its value

Execution via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the 'Data Factory' sidebar shows the 'myPipeline2' pipeline selected. The main canvas shows the 'myPipeline2' pipeline with a 'Stored Procedure' activity named 'MySprocActivity2'. The 'Trigger' menu is open, showing 'Trigger Now' and 'New/Edit' options. A red arrow points from the 'Trigger Now' button to a red circle containing the text 'Click on "Test Run"/"Trigger Now"'. The 'Stored Procedure' tab is selected, showing the 'Stored procedure name' as 'sp_executesql' and an 'Import parameter' table with columns NAME, TYPE, and VALUE.

NAME	TYPE	VALUE
stmt	string	DECLARE @return_v...

Execution via ADFv2 Sproc Activity

```
##### PSH script to invoke/trigger SSIS package executions in ADFv2 #####
```

```
# Create a linked service for your Azure SQL Database/Managed Instance server  
hosting SSISDB
```

```
Set-AzureRmDataFactoryV2LinkedService -ResourceGroupName $ResourceGroupName -  
DataFactoryName $DataFactoryName -Name "myLinkedService" -File  
"C:\ADF\SSIS\myLinkedService.json"
```

```
# Create a pipeline with sproc activity to execute your SSIS package(s)
```

```
Set-AzureRmDataFactoryV2Pipeline -ResourceGroupName $ResourceGroupName -  
DataFactoryName $DataFactoryName -Name "myPipeline" -DefinitionFile  
"C:\ADF\SSIS\myPipeline.json"
```

```
# Run your pipeline on demand
```

```
$myPipelineRun = Invoke-AzureRmDataFactoryV2Pipeline -ResourceGroupName  
$ResourceGroupName -DataFactoryName $DataFactoryName -PipelineName "myPipeline"
```

Execution via ADFv2 Sproc Activity

```
// JSON script to create a linked service for Azure SQL DB/MI server hosting SSISDB
```

```
{  
  "name": "myLinkedService",  
  "properties": {  
    "type": "AzureSqlDatabase",  
    "typeProperties": {  
      "connectionString": {  
        "type": "SecureString",  
        "value":  
"Server=tcp:YourAzureSQLDBServer.database.windows.net/YourAzureSQLMIServerEndpoint  
,1433;Database=SSISDB;User  
ID=YourUsername;Password=YourPassword;Trusted_Connection=False;Encrypt=True;Connec  
tion Timeout=30"  
      }  
    }  
  }  
}
```

Execution via ADFv2 Sproc Activity

// JSON script to create a pipeline with SqlServerStoredProcedure activity

```
{
  "name": "myPipeline",
  "properties": {
    "activities": [ {
      "name": "mySProcActivity",
      "description": "Sproc Activity to execute SSIS package(s)",
      "type": "SqlServerStoredProcedure",
      "linkedServiceName": {
        "referenceName": "myLinkedService",
        "type": "LinkedServiceReference"
      },
      "typeProperties": {
        "storedProcedureName": "sp_executesql",
        "storedProcedureParameters": {
          "stmt": {
            "value": "DECLARE @return_value INT, @exe_id BIGINT, @err_msg
NVARCHAR(150) EXEC @return_value=[SSISDB].[catalog].[create_execution] @folder_name=N'YourFolder', @project_name=N'YourProject',
@package_name=N'YourPackage', @use32bitruntime=0, @runinscaleout=1, @useanyworker=1, @execution_id=@exe_id OUTPUT EXEC
[SSISDB].[catalog].[set execution parameter value] @exe_id, @object type=50, @parameter name=N'SYNCHRONIZED', @parameter value=1 EXEC
[SSISDB].[catalog].[start_execution] @execution_id=@exe_id, @retry_count=0  IF(SELECT [status] FROM [SSISDB].[catalog].[executions]
WHERE execution id=@exe id)<>7 BEGIN SET @err_msg=N'Your package execution did not succeed for execution ID: ' + CAST(@exe_id AS
NVARCHAR(20)) RAISERROR(@err_msg,15,1) END"
          }
        }
      }
    } ]
  }
}
```

Execution via ADFv2 Sproc Activity

```
-- T-SQL script to create/start SSIS package execution using SSISDB sprocs
DECLARE @return_value int, @exe_id bigint, @err_msg nvarchar(150)

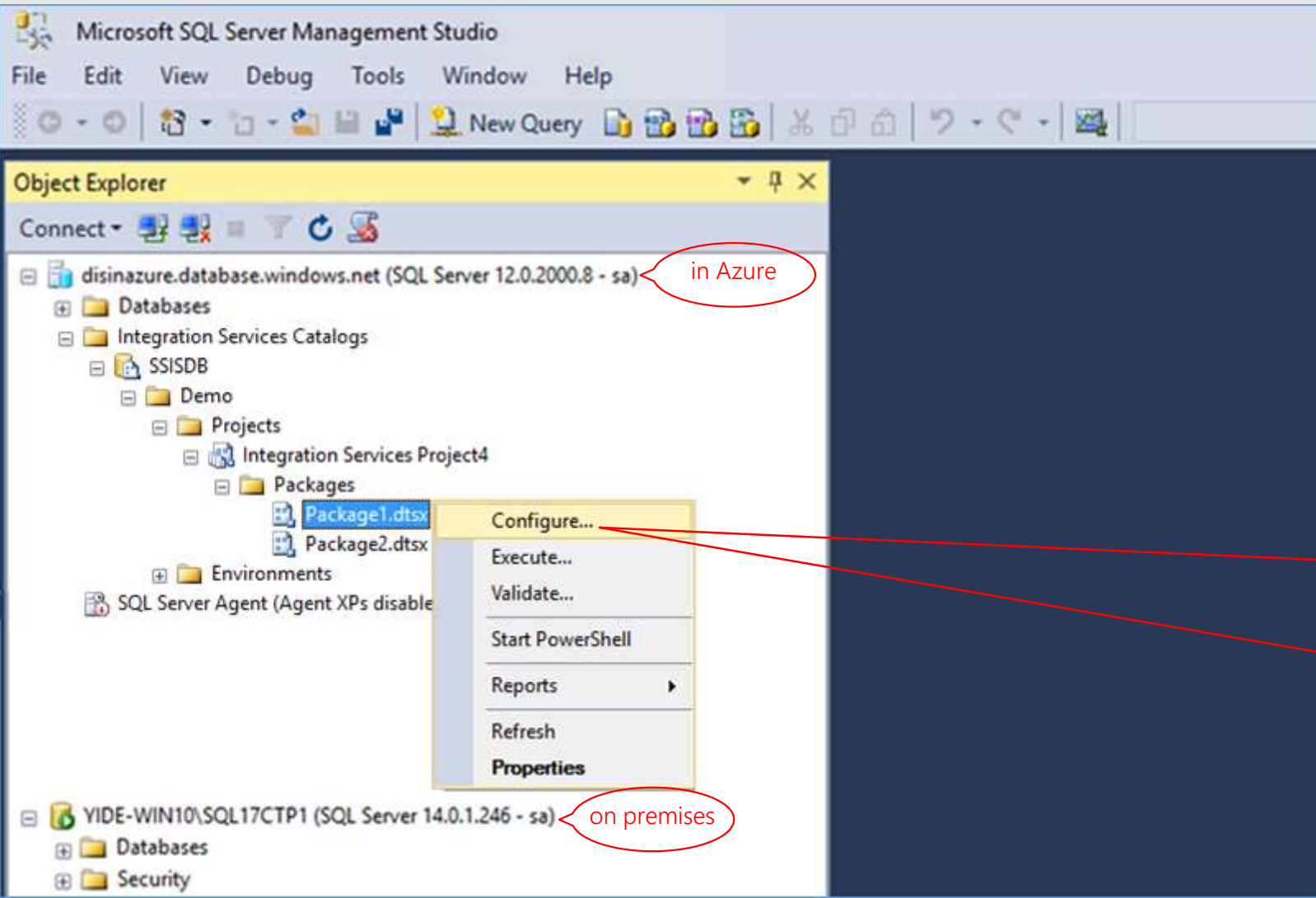
EXEC @return_value = [SSISDB].[catalog].[create_execution] @folder_name=N'YourFolder', @project_name=N'YourProject',
    @package_name=N'YourPackage', @use32bitruntime=0, @runincluster=1, @useanyworker=1,
    @execution_id=@exe_id OUTPUT

-- To synchronize SSIS package execution, set SYNCHRONIZED execution parameter
EXEC [SSISDB].[catalog].[set_execution_parameter_value] @exe_id, @object_type=50, @parameter_name=N'SYNCHRONIZED',
    @parameter_value=1

EXEC [SSISDB].[catalog].[start_execution] @execution_id = @exe_id, @retry_count = 0

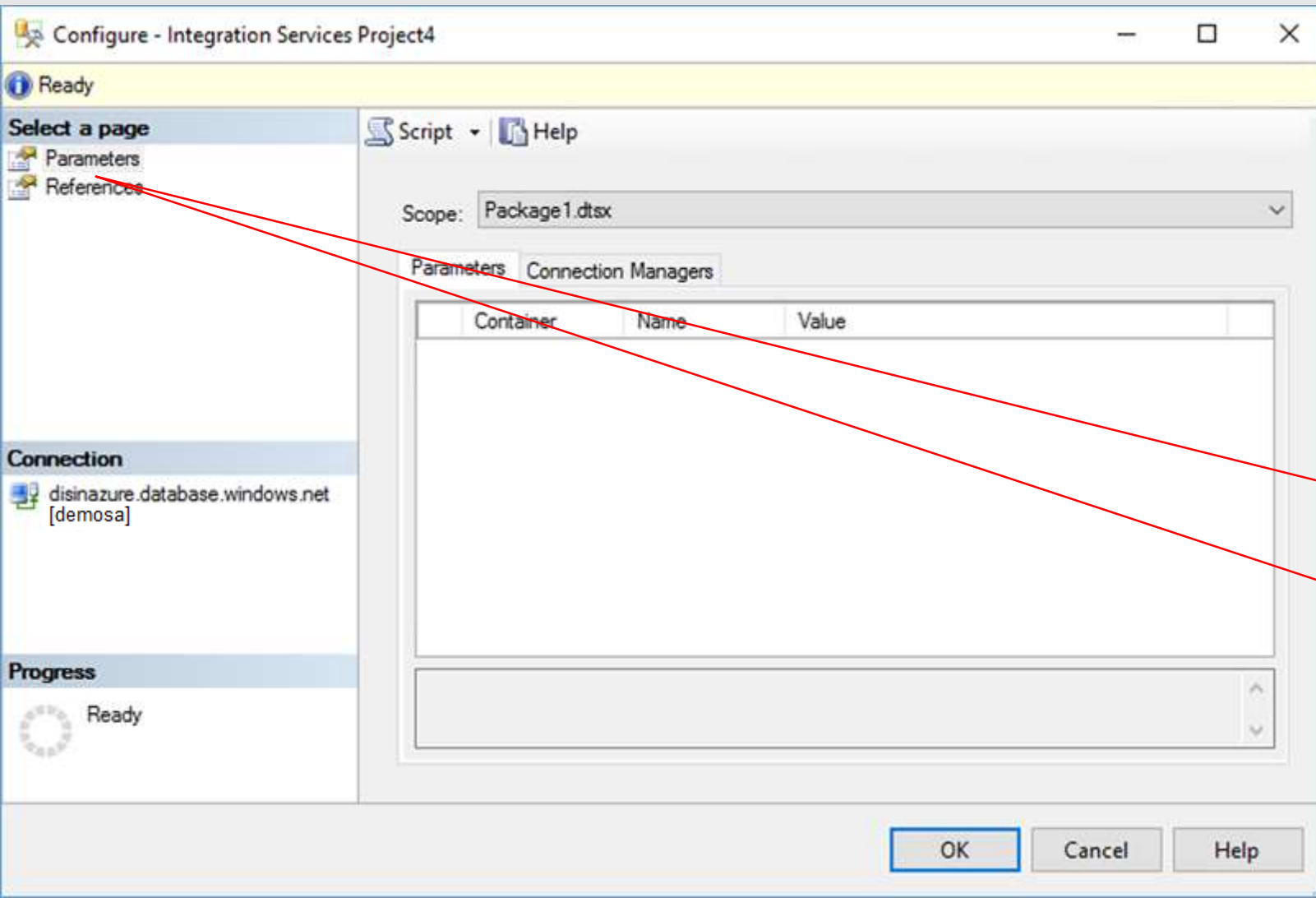
-- Raise an error for unsuccessful package execution, check package execution status = created (1)/running
(2)/canceled (3)/failed (4)/
-- pending (5)/ended unexpectedly (6)/succeeded (7)/stopping (8)/completed (9)
IF (SELECT [status] FROM [SSISDB].[catalog].[executions] WHERE execution_id = @exe_id)<>7
BEGIN
    SET @err_msg=N'Your package execution did not succeed for execution ID: '+ CAST(@execution_id as
    nvarchar(20))
    RAISERROR(@err_msg, 15, 1)
END
```

Execution via SSMS



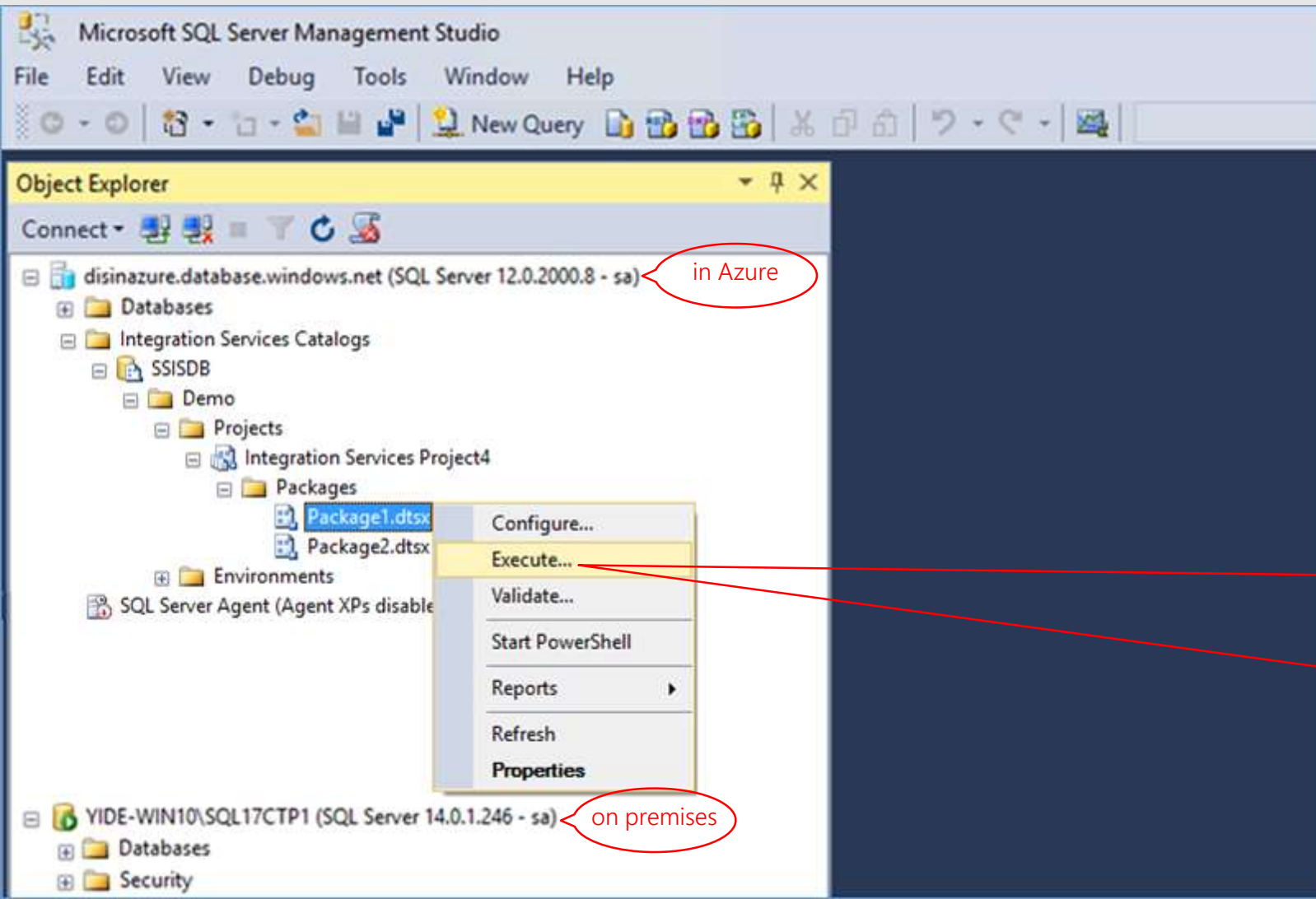
Once deployed, you can configure packages for execution on SSIS PaaS

Execution via SSMS



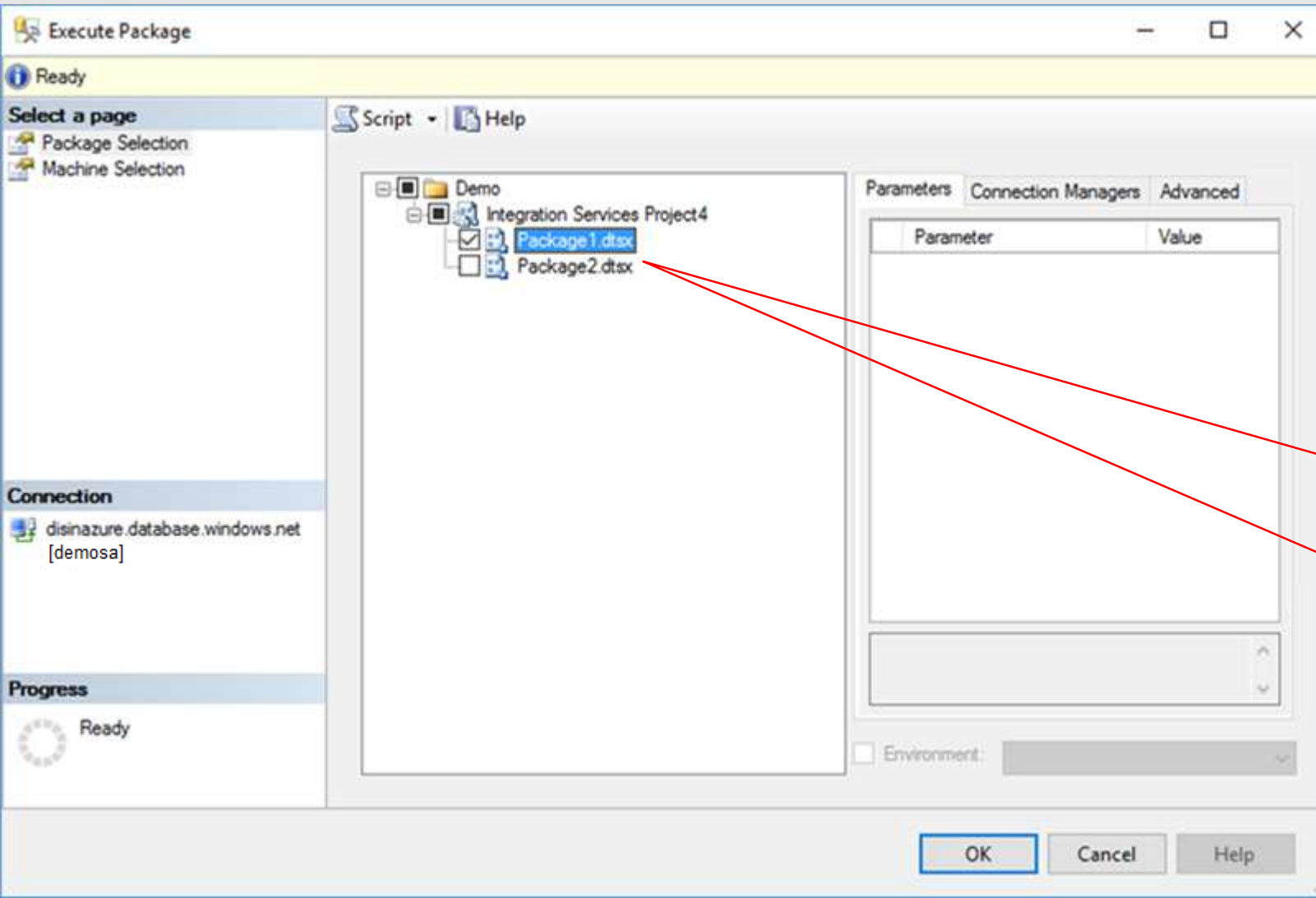
You can set package run-time parameters/environment references

Execution via SSMS



Once configured, you can execute packages on SSIS PaaS

Execution via SSMS



You can select some packages to execute on SSIS PaaS

Scheduling Methods

Scheduling Methods

- SSIS package executions can be directly scheduled as first-class SSIS activities in ADFv2 pipelines (Work in Progress)
 - For now, SSIS package executions can be indirectly scheduled via ADFv1/v2 Sproc Activity
- If you use Azure SQL MI server to host SSISDB
 - SSIS package executions can also be scheduled via Azure SQL MI Agent (Extended Private Preview)
- If you use Azure SQL DB server to host SSISDB
 - SSIS package executions can also be scheduled via Elastic Jobs (Private Preview)
- If you keep on-prem SQL Server
 - SSIS package executions can also be scheduled via on-prem SQL Server Agent

Scheduling via ADFv1 Sproc Activity

- Create a linked service for Azure SQL DB/MI server hosting SSISDB
- Create an output dataset that drives scheduling
- Create a pipeline with SqlServerStoredProcedure activity

Scheduling via ADFv1 Sproc Activity

- Create a linked service for Azure SQL DB/MI server hosting SSISDB

Microsoft Azure

Report a bug Search resources, services and docs

Home > sawinarkADFv1 > sawinarkADFv1 > Linked services/AzureSqlLinkedService

sawinarkADFv1 Data factory

Delete Move

Essentials

Resource group
sawinarkRG

Location
EastUS

Provisioning state
Succeeded

Data Integration App (INTERNAL PREVIEW)
Open link

Type
Data factory

Subscription name
EIMS_TEST_EVEREST_1

Subscription id
cb715d05-3337-4640-8c43-4f943c50d06e

All settings →

Actions

Author and deploy

Copy data (PREVIEW)

Monitor & Manage

Sample pipelines

Diagram

Metrics and operations

Contents

Datasets	Pipelines	Linked services
1	1	1
With errors 0	Scheduled 1 One time 0	Data Stores 1 Integration runtim... 0

See more

sawinarkADFv1

New data store More

Linked services

AzureSqlLinkedService

Datasets

Pipelines

Integration runtimes (Gateways)

Drafts

Linked services/AzureSqlLinkedService

Add activity Encrypt Clone Discard Deploy

```
{
  "name": "AzureSqlLinkedService",
  "properties": {
    "description": "",
    "hubName": "sawinarkadf1_hub",
    "type": "AzureSqlDatabase",
    "typeProperties": {
      "connectionString": "Data Source=tcp:disinazure.database.windows.net,1433;Initial Catalog=SSISDB;Int..."
    }
  }
}
```

Scheduling via ADFv1 Sproc Activity

// JSON script to create a linked service for Azure SQL DB/MI server hosting SSISDB

```
{
  "name": "AzureSqlLinkedService",
  "properties": {
    "description": "",
    "type": "AzureSqlDatabase",
    "typeProperties": {
      "connectionString": "Data
Source=tcp:YourAzureSQLDBServer.database.windows.net/YourAzureSQLMIServerEndpoint,
1433;Initial Catalog=SSISDB;User ID=YourUsername;Password=YourPassword;Integrated
Security=False;Encrypt=True;Connect Timeout=30"
    }
  }
}
```

Scheduling via ADFv1 Sproc Activity

- Create an output dataset that drives scheduling

Microsoft Azure

Home > sawinarkADFv1 > sawinarkADFv1 > Datasets/sprocsampleout

sawinarkADFv1 Data factory

Delete Move

Essentials

Resource group
sawinarkRG

Location
EastUS

Provisioning state
Succeeded

Data Integration App (INTERNAL PREVIEW)
[Open link](#)

Type
Data factory

Subscription name
EIMS_TEST_EVEREST_1

Subscription id
cb715d05-3337-4640-8c43-4f943c50d06e

[All settings](#)

Actions

Author and deploy

Copy data (PREVIEW)

Monitor & Manage

Sample pipelines

Diagram

Metrics and operations

Contents

Datasets	Pipelines	Linked services
1	1	1
With errors 0	Scheduled 1	Data Stores 1
	One time 0	Integration runtim... 0
		See more

sawinarkADFv1

New data store More

Linked services
AzureSqlLinkedService

Datasets
sprocsampleout

Pipelines

Integration runtimes (Gateways)

Drafts

Datasets/sprocsampleout

Add activity Encrypt Clone Discard Deploy

```
{
  "name": "sprocsampleout",
  "properties": {
    "published": false,
    "type": "AzureSqlTable",
    "linkedServiceName": "AzureSqlLinkedService",
    "typeProperties": {},
    "availability": {
      "frequency": "Minute",
      "interval": 15
    }
  }
}
```

Scheduling via ADFv1 Sproc Activity

```
// JSON script to create an output dataset that drives scheduling
{
  "name": "sprocsampleout",
  "properties": {
    "type": "AzureSqlTable",
    "linkedServiceName": "AzureSqlLinkedService",
    "typeProperties": {},
    "availability": {
      "frequency": "Hour",
      "interval": 1
    }
  }
}
```


Scheduling via ADFv1 Sproc Activity

- Create a pipeline with SqlServerStoredProcedure activity

Microsoft Azure

Home > sawinarkADFv1 > sawinarkADFv1 > Pipelines/SprocActivitySamplePipeline

sawinarkADFv1 Data factory

Delete Move

Essentials

Resource group
sawinarkRG

Location
EastUS

Provisioning state
Succeeded

Data Integration App (INTERNAL PREVIEW)
Open link

Type
Data factory

Subscription name
EIMS_TEST_EVEREST_1

Subscription id
cb715d05-3337-4640-8c43-4f943c50d06e

All settings →

Actions

Author and deploy

Copy data (PREVIEW)

Monitor & Manage

Sample pipelines

Diagram

Metrics and operations

Contents

Datasets	Pipelines	Linked services
1	1	1
With errors 0	Scheduled 1 One time 0	Data Stores 1 Integration runtim... 0

See more

sawinarkADFv1

New data store More

Linked services
AzureSqlLinkedService

Datasets
sprocsampleout

Pipelines
SprocActivitySamplePipeline

Integration runtimes (Gateways)
Drafts

Pipelines/SprocActivitySamplePipeline

Add activity Encrypt Clone Discard Deploy

```
{
  "name": "SprocActivitySamplePipeline",
  "properties": {
    "activities": [
      {
        "type": "SqlServerStoredProcedure",
        "typeProperties": {
          "storedProcedureName": "sp_executesql",
          "storedProcedureParameters": {
            "stmt": "DECLARE @return_value INT, @exe_id BIGINT, @err_msg NVARCHAR(150) EXEC @return_value = ..."
          }
        },
        "outputs": [
          {
            "name": "sprocsampleout"
          }
        ],
        "scheduler": {
          "frequency": "Minute",
          "interval": 15
        },
        "name": "SprocActivitySample"
      }
    ],
    "start": "2017-12-12T02:45:00Z",
    "end": "2017-12-12T03:30:00Z",
    "isPaused": false,
    "hubName": "sawinarkadf1_hub",
    "pipelineMode": "Scheduled"
  }
}
```

Scheduling via ADFv1 Sproc Activity

// JSON script to create a pipeline with SqlServerStoredProcedure activity

```
{
  "name": "SprocActivitySamplePipeline",
  "properties": {
    "activities": [ {
      "name": "SprocActivitySample",
      "type": "SqlServerStoredProcedure",
      "typeProperties": {
        "storedProcedureName": "sp_executesql",
        "storedProcedureParameters": {
          "stmt": "DECLARE @return_value INT, @exe_id BIGINT, @err_msg NVARCHAR(150) EXEC
@return_value=[SSISDB].[catalog].[create_execution] @folder_name=N'YourFolder', @project_name=N'YourProject', @package_name=N'YourPackage',
@use32bitruntime=0, @runinscaleout=1, @useanyworker=1, @execution_id=@exe_id OUTPUT EXEC [SSISDB].[catalog].[set_execution_parameter_value] @exe_id,
@object_type=50, @parameter_name=N'SYNCHRONIZED', @parameter_value=1 EXEC [SSISDB].[catalog].[start_execution] @execution_id=@exe_id, @retry_count=0
IF(SELECT [status] FROM [SSISDB].[catalog].[executions] WHERE execution_id=@exe_id)<>7 BEGIN SET @err_msg=N'Your package execution did not succeed for
execution ID: ' + CAST(@exe_id AS NVARCHAR(20)) RAISERROR(@err_msg,15,1) END"
        },
      },
      "outputs": [ {
        "name": "sprocsampleout"
      } ],
      "scheduler": {
        "frequency": "Hour",
        "interval": 1
      }
    } ],
    "start": "2017-04-02T00:00:00Z",
    "end": "2017-04-02T05:00:00Z",
    "isPaused": false
  }
}
```

Scheduling via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the 'Data Factory' sidebar shows the 'Pipelines' section with 'myPipeline2' selected. The main canvas shows the 'Triggers' tab for 'myPipeline2'. The 'Trigger' dropdown menu is open, showing 'Trigger Now' and 'New/Edit' options. A red arrow points from the 'New/Edit' option to the 'Stored Procedure' activity in the pipeline canvas. The 'Stored Procedure' activity is named 'MySprocActivity2'. Below the activity, the 'Stored Procedure' tab is selected, showing the 'Details' section with the 'Stored procedure name' set to 'sp_executesql'. The 'Import parameter' section shows a table with columns 'NAME', 'TYPE', and 'VALUE'. The table contains one row: 'stmt' (string) with the value 'DECLARE @return_v...'.

NAME	TYPE	VALUE
stmt	string	DECLARE @return_v...

Click on "Trigger" and
"New/Edit"

Scheduling via ADFv2 Sproc Activity

The screenshot shows the Azure Data Factory (ADFv2) interface. On the left, the navigation pane displays the 'Data Factory' section with 'Pipelines' and 'myPipeline2' selected. The main canvas shows the 'myPipeline2' pipeline with a 'Stored Procedure' activity. The 'Stored Procedure' activity is configured with the name 'sp_executesql' and a parameter 'stmt' of type 'string' with the value 'DECLARE @return_v...'. The 'Triggers' tab is active, showing a search bar and a list of triggers. A red arrow points from the 'Add Triggers' dialog to the 'myTrigger' trigger in the list.

Triggers X myPipeline2 X

Activities

Search Activities

Batch Service

Data Flow

Data Lake Analytics

General

Execute Pipeline

Get Metadata

Lookup

Stored Procedure

Web

HDInsight

Iteration & Conditionals

Connections

Triggers

Stored Procedure

MySprocActi

Details

Stored procedure name *

sp_executesql

Edit

Import parameter

Stored procedure parameters

New Delete

NAME	TYPE	VALUE
stmt	string	DECLARE @return_v...

The 'Add Triggers' dialog is shown, featuring a search bar and a list of triggers. The 'Choose trigger...' dropdown is selected, and the '+ New' button is highlighted. A red arrow points from the '+ New' button to the 'myTrigger' trigger in the list.

Add Triggers

Choose trigger...

Search

+ New

myTrigger

Click on "Choose trigger..."
and "+ New"

Scheduling via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the 'Data Factory' sidebar shows the 'myPipeline2' pipeline selected. The main canvas shows the 'myPipeline2' pipeline with a 'Stored Procedure' activity named 'MySprocAct'. The 'New Trigger' dialog box is open, showing the configuration for a new trigger named 'myTrigger2'. The dialog includes the following fields:

- Name:** myTrigger2
- Description:** (Empty text area)
- Type:** Schedule (selected), Tumbling Window
- Start Date (UTC):** 01/26/2018, 4:45 AM
- Recurrence:** Every Minute (selected), Every 15 Minute(s)
- End:** No End (selected), On Date
- End On (UTC):** 01/26/2018, 5:45 AM
- Activated:** ☒ Activated

A red circle highlights the 'Name' and 'Description' fields with the text 'Name and describe your trigger'.

Scheduling via ADFv2 Sproc Activity

The screenshot shows the Azure Data Factory (ADFv2) interface. On the left, the 'Data Factory' sidebar is visible with options like 'Publish All', 'Search Restrictions', 'Pipelines', 'myPipeline', 'myPipeline2', and 'Datasets'. The main area displays the 'myPipeline2' pipeline with a 'Stored Procedure' activity selected. The 'Stored Procedure' activity is configured with the name 'sp_execute_sql' and a parameter 'stmt' of type 'string' with the value 'DECLARE @return_v...'. The 'New Trigger' dialog box is open, showing the following configuration:

- Name:** myTrigger2
- Description:** (empty)
- Type:** ☒ Schedule ☐ Tumbling Window
- Start Date (UTC):** 01/26/2018, 4:45 AM
- Recurrence:** Every Minute, Every 15 Minute(s)
- End:** ☐ No End ☒ On Date
- End On (UTC):** 01/26/2018, 5:45 AM
- Activated:** ☒

A red circle highlights the 'Schedule' option under the 'Type' section, with a red arrow pointing to it from the text "Select a trigger of type 'Schedule'".

Scheduling via ADFv2 Sproc Activity

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the 'Data Factory' sidebar shows the 'myPipeline2' pipeline. The main canvas shows the 'myPipeline2' pipeline with a 'Stored Procedure' activity named 'MySprocAct'. The 'New Trigger' dialog box is open, showing the configuration for a new trigger named 'myTrigger2'. The dialog is configured as a 'Schedule' type with a recurrence of 'Every 15 Minute(s)'. The start date is set to '01/26/2018, 4:45 AM' and the end date is '01/26/2018, 5:45 AM'. The trigger is marked as 'Activated'. A red callout bubble points to the start date, recurrence, and end date fields with the text: 'Select the start date, recurrence, and end date of your trigger'.

Triggers **myPipeline2**

Activities:

Batch Service

Data Flow

Data Lake Analytics

General

Execute Pipeline

Get Metadata

Lookup

Stored Procedure

Web

HDInsight

Iteration & Conditionals

Connections

Triggers

Details

Stored procedure name:

Import parameter

Stored procedure parameters

+ New - Delete

NAME	TYPE	VALUE
stmt	string	DECLARE @return_v...

New Trigger

Name:

Description:

Type: ☒ Schedule ☐ Tumbling Window

Start Date (UTC):

Recurrence: Every Minute(s)

End: ☐ No End ☒ On Date

End On (UTC):

☒ Activated

Cancel Next

Select the start date, recurrence, and end date of your trigger

Scheduling via ADFv2 Sproc Activity

The screenshot shows the Azure Data Factory (ADFv2) interface. On the left, the 'Data Factory' sidebar is visible with options like 'Publish All', 'Search Restrictions', 'Pipelines', 'myPipeline', 'myPipeline2', and 'Datasets'. The main workspace displays the 'myPipeline2' pipeline with a 'Stored Procedure' activity named 'MySprocAct'. The 'Triggers' tab is active, showing a 'New Trigger' dialog box.

The 'New Trigger' dialog box contains the following fields and options:

- Name:** myTrigger2
- Description:** (empty text area)
- Type:** ☒ Schedule ☐ Tumbling Window
- Start Date (UTC):** 01/26/2018, 4:45 AM
- Recurrence:** Every Minute, Every 15 Minute(s)
- End:** ☐ No End ☒ On Date
- End On (UTC):** 01/26/2018, 5:45 AM
- Activated:** ☒ (highlighted with a red circle and arrow)

A red circle with an arrow points to the 'Activated' checkbox, with the text 'Activate your trigger' written in red.

Scheduling via ADFv2 Sproc Activity

Triggers X myPipeline2 X

Activities Validate Test Run Trigger

Search Activities

Batch Service

Data Flow

Data Lake Analytics

General

Execute Pipeline

Get Metadata

Lookup

Stored Procedure

Web

HDInsight

Iteration & Conditionals

Connections

Triggers

myPipeline

myPipeline2

Datasets

Search Resources

Publish All

Data Factory

myPipeline2 X

Stored Procedure

MySprocAct

General SQL Account Stored Procedure Parameters Advanced

Details

Stored procedure name: sp_executesql

Import parameter

Stored procedure parameters

New Delete

NAME TYPE VALUE

stmt string DECLARE @return_v...

New Trigger

Trigger Run Parameters

Name	Type	Value
This pipeline has no parameters		

Make sure to "Publish" for trigger to be activated after clicking "Finish"

Cancel Finish

Finish and publish your trigger

Scheduling via ADFv2 Sproc Activity

The screenshot shows the Azure Data Factory (ADFv2) interface. The left sidebar contains the following elements:

- Top bar: sawinarkRG, sawinarkADFv2
- Navigation icons: Data Factory, Pipelines, Datasets
- Search bar: Search Resources
- Bottom bar: Connections, Triggers

The main area displays the 'Triggers' tab for 'myPipeline2'. It shows a table with the following data:

Name	Action	Type	Number of pipelines
myTrigger	[Icons]	Schedule	1
myTrigger2	[Icons]	Schedule	1

A 'Pipelines' dialog box is open, showing the name 'myPipeline2'.

Click on "Triggers" to see your published triggers

Scheduling via ADFv2 Sproc Activity

```
##### PSH script to schedule SSIS package executions in  
ADFv2 #####
```

```
# Create a trigger to schedule your pipeline runs
```

```
Set-AzureRmDataFactoryV2Trigger -ResourceGroupName  
$ResourceGroupName -DataFactoryName $DataFactoryName -Name  
"myTrigger" -DefinitionFile "C:\ADF\SSIS\myTrigger.json"
```

```
# Start your trigger
```

```
Start-AzureRmDataFactoryV2Trigger -ResourceGroupName  
$ResourceGroupName -DataFactoryName $DataFactoryName -Name  
"myTrigger"
```

Scheduling via ADFv2 Sproc Activity

// JSON script to create a trigger for scheduling your pipeline runs

```
{
  "properties": {
    "name": "myTrigger",
    "type": "ScheduleTrigger",
    "typeProperties": {
      "recurrence": {
        "frequency": "Minute",
        "interval": 15,
        "startTime": "2017-12-22T04:30:00Z",
        "endTime": "2017-12-22T05:00:00Z"
      }
    },
    "pipelines": [{
      "pipelineReference": {
        "type": "PipelineReference",
        "referenceName": "myPipeline"
      },
      "parameters": {}
    }]
  }
}
```

Scheduling via Elastic Jobs

- Create Elastic Jobs DB
- Create Elastic Jobs account
- Create DB-scoped credentials
- Add Elastic Jobs target groups
- Add Elastic Jobs target group members
- Add jobs to schedule SSIS package executions
- Add job steps to create and start SSIS package executions
- Enable job schedules

Scheduling via Elastic Jobs

```
-- T-SQL script to set up Elastic Jobs for scheduling SSIS package execution
-- Create Elastic Jobs target group
EXEC jobs.sp_add_target_group 'TargetGroup'

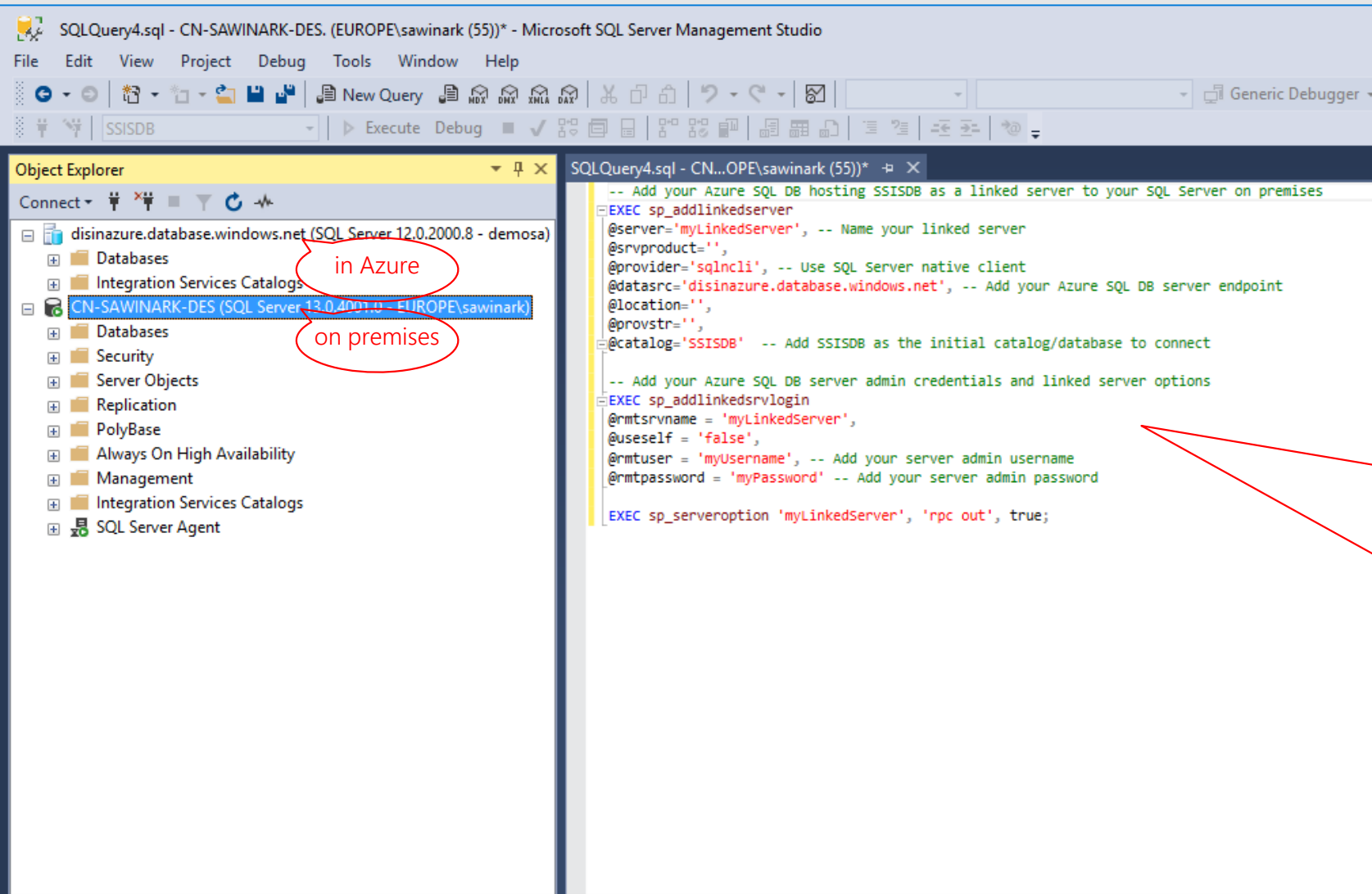
-- Add Elastic Jobs target group member
EXEC jobs.sp_add_target_group_member @target_group_name='TargetGroup', @target_type='SqlDatabase',
    @server_name='YourAzureSQLDBServer.database.windows.net',@database_name='SSISDB'

-- Add a job to schedule SSIS package execution
EXEC jobs.sp_add_job @job_name='ExecutePackageJob', @description='Description', @schedule_interval_type='Minutes',
    @schedule_interval_count=60

-- Add a job step to create/start SSIS package execution using SSISDB sprocs
EXEC jobs.sp_add_jobstep @job_name='ExecutePackageJob', @command=N'DECLARE @exe_id bigint
    EXEC [SSISDB].[catalog].[create_execution] @folder_name=N''YourFolder'', @project_name=N''YourProject'',
        @package_name=N''YourPackage'', @use32bitruntime=0, @runincluster=1, @useanyworker=1,
    @execution_id=@exe_id OUTPUT
    EXEC [SSISDB].[catalog].[start_execution] @exe_id, @retry_count=0', @credential_name='YourDBScopedCredentials',
    @target_group_name='TargetGroup'

-- Enable the job schedule
EXEC jobs.sp_update_job @job_name='ExecutePackageJob', @enabled=1, @schedule_interval_type='Minutes',
    @schedule_interval_count=60
```

Scheduling via On-Prem SQL Server Agent



Add your Azure SQL DB hosting SSISDB as a linked server to your SQL Server on premises

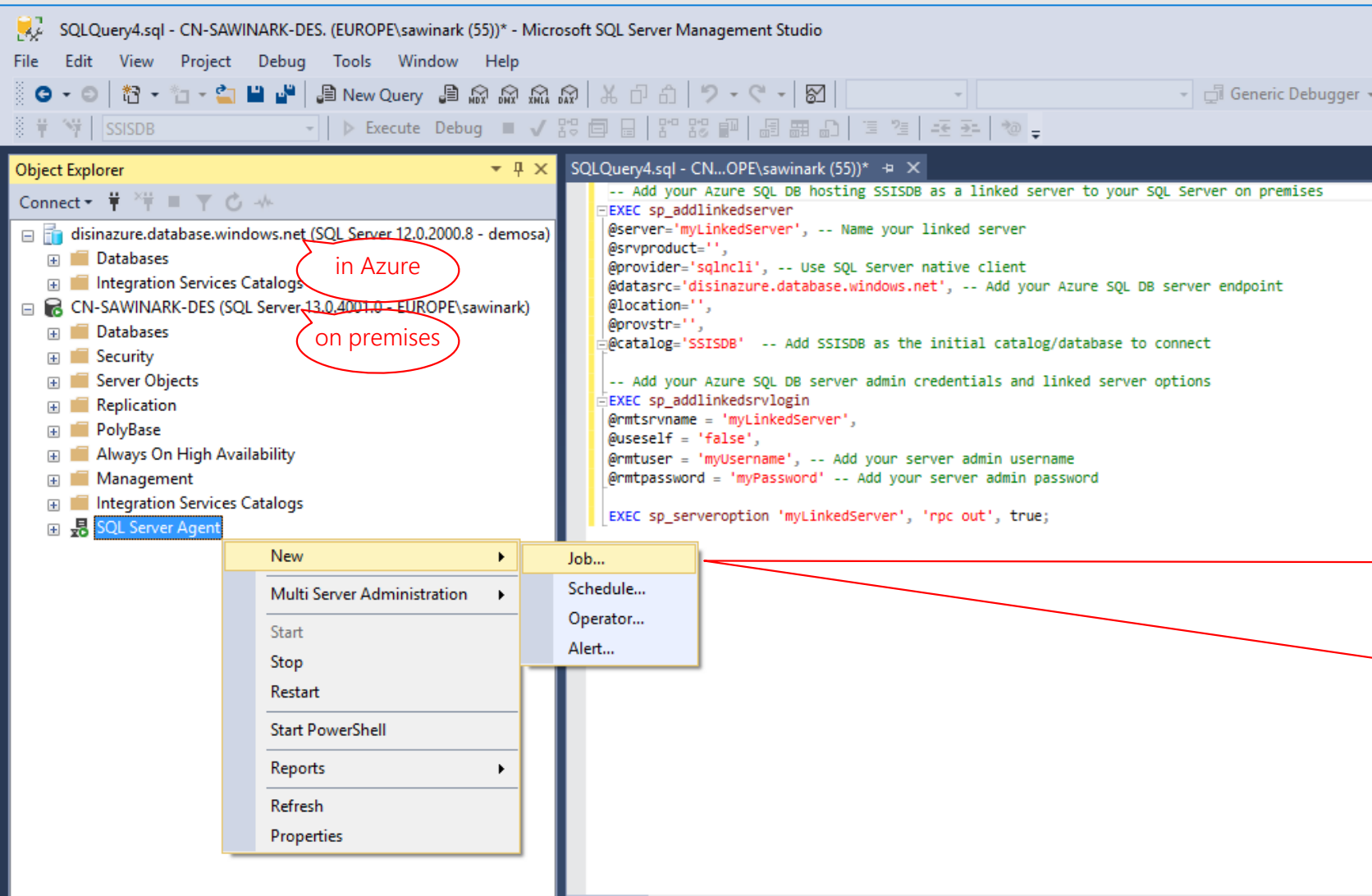
Scheduling via On-Prem SQL Server Agent

```
-- Add your Azure SQL DB hosting SSISDB as a linked server to your SQL Server on premises
EXEC sp_addlinkedserver
    @server='myLinkedServer', -- Name your linked server
    @srvproduct='',
    @provider='sqlncli', -- Use SQL Server native client
    @datasrc='disinazure.database.windows.net', -- Add your Azure SQL DB server endpoint
    @location='',
    @provstr='',
    @catalog='SSISDB' -- Add SSISDB as the initial catalog/database to connect

-- Add your Azure SQL DB server admin credentials and linked server options
EXEC sp_addlinkedsrvlogin
    @rmtsrvname = 'myLinkedServer',
    @useself = 'false',
    @rmtuser = 'myUsername', -- Add your server admin username
    @rmtpassword = 'myPassword' -- Add your server admin password

EXEC sp_serveroption 'myLinkedServer', 'rpc out', true;
```


Scheduling via On-Prem SQL Server Agent



Add T-SQL jobs to schedule SSIS package executions on SSIS PaaS via on-prem SQL Server Agent

Scheduling via On-Prem SQL Server Agent

The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the 'Jobs' folder expanded under the 'CN-SAWINARK-DES (SQL Server 13.0.4001.0 - EUROPE)' instance. The 'ExecuteSSISPackage' job is selected. The right pane shows the 'Job Properties - ExecuteSSISPackage' dialog box. The 'Job step list' table shows one step named 'ExecuteSSISPackage' of type 'Transact-SQL script (T-SQL)'. The 'Job Step Properties - ExecuteSSISPackage' dialog box is also open, showing the 'Step name' as 'ExecuteSSISPackage', 'Type' as 'Transact-SQL script (T-SQL)', 'Run as' as 'SQL Server Agent', 'Database' as 'SSISDB', and 'Command' as a T-SQL script to create/start SSIS package execution using SSISDB sprocs. The 'Connection' section shows the server as 'CN-SAWINARK-DES' and the connection as 'EUROPE\sawinark'.

SQLQuery4.sql - CN-SAWINARK-DES. (EUROPE\sawinark)

File Edit View Query Project Debug Tools

SSISDB Execute D

Object Explorer

Connect

disinazure.database.windows.net (SQL Server 12.0.20) in Azure

CN-SAWINARK-DES (SQL Server 13.0.4001.0 - EUROPE) on premises

Databases

Integration Services Catalogs

Jobs

SSIS Server Maintenance Job

syspolicy_purge_history

ExecuteSSISPackage

Job Activity Monitor

Alerts

Operators

Proxies

Error Logs

Job Properties - ExecuteSSISPackage

Select a page

General

Steps

Schedules

Alerts

Notifications

Targets

Script Help

Job step list:

St...	Name	Type	On Success	On Failure
1	ExecuteSSISPackage	Transact...	Quit the j...	Quit the job...

Job Step Properties - ExecuteSSISPackage

Select a page

General

Advanced

Script Help

Step name:

ExecuteSSISPackage

Type:

Transact-SQL script (T-SQL)

Run as:

SQL Server Agent

Database:

SSISDB

Command:

```
-- T-SQL script to create/start SSIS package execution using SSISDB sprocs
DECLARE @return_value INT, @exe_id BIGINT
EXEC @return_value=myLinkedServer.[SSISDB].[catalog].[create_execution]
EXEC myLinkedServer.[SSISDB].[catalog].[set_execution_parameter_value]
EXEC myLinkedServer.[SSISDB].[catalog].[start_execution] @execution_id
```

Open...

Select All

Copy

Paste

Parse

Connection

Server:

CN-SAWINARK-DES

Connection:

EUROPE\sawinark

View connection properties

T-SQL jobs can execute SSISDB sprocs in Azure SQL DB that has been added as a linked server to SQL Server on premises

Scheduling via On-Prem SQL Server Agent

-- T-SQL script to create/start SSIS package execution using SSISDB sprocs

DECLARE @return_value int, @exe_id bigint

```
EXEC @return_value = [YourLinkedServer].[SSISDB].[catalog].[create_execution]  
    @folder_name=N'folderName', @project_name=N'projectName',  
    @package_name=N'packageName', @use32bitruntime=0, @runincluster=1, @useanyworker=1,  
    @execution_id=@exe_id OUTPUT
```

```
EXEC [YourLinkedServer].[SSISDB].[catalog].[set_execution_parameter_value] @exe_id,  
    @object_type=50, @parameter_name=N'SYNCHRONIZED', @parameter_value=1
```

```
EXEC [YourLinkedServer].[SSISDB].[catalog].[start_execution] @execution_id=@exe_id
```

Scheduling via On-Prem SQL Server Agent

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the server hierarchy for 'disinazure.database.windows.net (SQL Server 12.0.20)' and 'CN-SAWINARK-DES (SQL Server 13.0.4001.0 - EUROPE)'. The 'Jobs' folder under 'CN-SAWINARK-DES' is expanded, showing 'SSIS Server Maintenance Job', 'syspolicy_purge_history', and 'ExecuteSSISPackage'. A red circle highlights 'ExecuteSSISPackage' with the text 'in Azure' and 'on premises'. The main pane shows the 'Job Properties - ExecuteSSISPackage' window. The 'Schedules' tab is selected, showing a 'Schedule list' table with one entry: 'MySSISSchedule' (ID 1009, Enabled Yes, Description 'Occurs every day every 15 minute(s)'). The 'Job Schedule Properties - MySSISSchedule' window is also open, showing the 'Name' as 'MySSISSchedule', 'Schedule type' as 'Recurring', and 'Enabled' checked. The 'Frequency' section shows 'Occurs: Daily', 'Recurs every: 1 day(s)', and 'Daily frequency' set to 'Occurs every: 15 minute(s)'. The 'Duration' section shows 'Start date: 12/25/2017' and 'End date: 12/26/2017'. The 'Summary' section shows the description: 'Occurs every day every 15 minute(s) between 12:00:00 AM and 11:59:59 PM. Schedule will be used between 12/25/2017 and 12/26/2017.'

ID	Name	Enabled	Description
1009	MySSISSchedule	Yes	Occurs every day every 15 minute(s)

in Azure
on premises

Schedule T-SQL jobs to execute SSISDB sprocs in Azure SQL DB server, triggering SSIS package executions on Azure-SSIS IR

Monitoring Methods

Monitoring Methods

- Azure-SSIS IR can be monitored via ADFv2 App
- Azure-SSIS IR can be monitored via PSH/custom code using ADFv2 .NET SDK/API
- SSIS package executions can be directly monitored as first-class SSIS activities in ADFv2 pipelines (Work in Progress)
 - For now, SSIS package executions can be indirectly monitored via ADFv2 Sproc Activity
- SSIS package executions can be monitored via SSMS

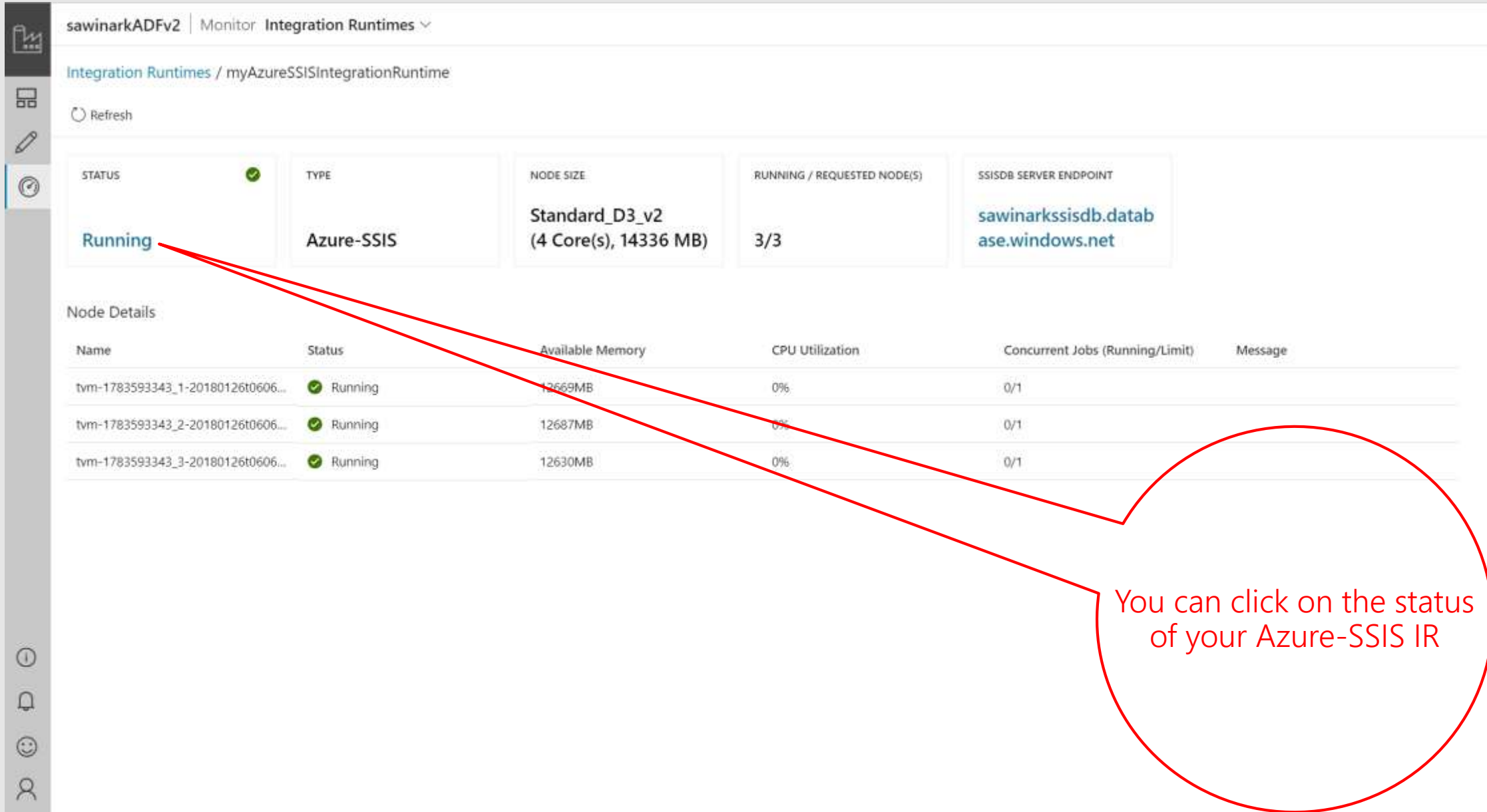
Monitoring via ADFv2 App

The screenshot displays the Azure Data Factory v2 portal interface. On the left, the navigation pane shows the 'Data Factory' section with options for 'Publish All', 'Search Resources', 'Pipelines' (containing 'myPipeline' and 'myPipeline2'), and 'Datasets'. The main area is titled 'Connections' and has two tabs: 'Linked Services' and 'Integration Runtimes'. The 'Integration Runtimes' tab is active, showing a table with two entries: 'defaultIntegrationRuntime' (Type: Azure, Status: Running, Region: Auto Resolve) and 'myAzureSSISIntegrationRuntime' (Type: Azure-SSIS, Status: Running, Region: East US). A red callout points to the 'Monitor' button located in the 'Actions' column for the 'myAzureSSISIntegrationRuntime' row.

Name	Actions	Type	Status	Region
defaultIntegrationRuntime		Azure	Running	Auto Resolve
myAzureSSISIntegrationRuntime	Monitor	Azure-SSIS	Running	East US

Once your Azure-SSIS IR is provisioned, you can monitor it

Monitoring via ADFv2 App



sawinarkADFv2 | Monitor Integration Runtimes ▾

Integration Runtimes / myAzureSSISIntegrationRuntime

Refresh

STATUS	TYPE	NODE SIZE	RUNNING / REQUESTED NODE(S)	SSISDB SERVER ENDPOINT
Running	Azure-SSIS	Standard_D3_v2 (4 Core(s), 14336 MB)	3/3	sawinarkssisdb.datab ase.windows.net

Node Details

Name	Status	Available Memory	CPU Utilization	Concurrent Jobs (Running/Limit)	Message
tvm-1783593343_1-20180126t0606...	Running	12669MB	0%	0/1	
tvm-1783593343_2-20180126t0606...	Running	12687MB	0%	0/1	
tvm-1783593343_3-20180126t0606...	Running	12630MB	0%	0/1	

You can click on the status of your Azure-SSIS IR

Monitoring via ADFv2 App

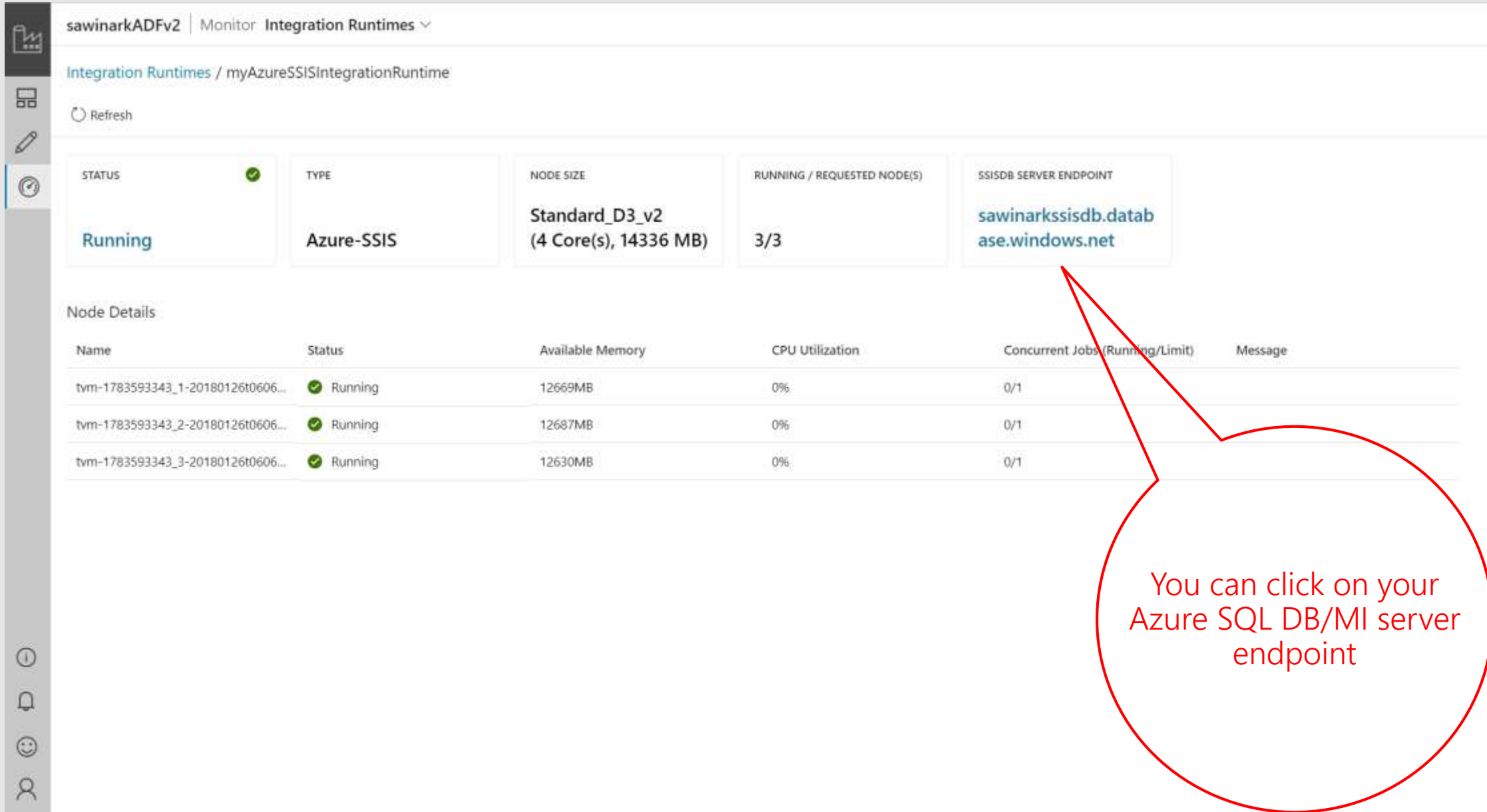
The screenshot displays the Azure Data Factory v2 Monitor interface. The top navigation bar shows 'sawinarkADFv2 | Monitor Integration Runtimes'. The main heading is 'Integration Runtimes / myAzureSSISIntegrationRuntime'. A 'Refresh' button is present. The main content area shows a summary card for the 'myAzureSSISIntegrationRuntime' with the following details:

- STATUS:** Running (indicated by a green checkmark)
- TYPE:** Azure-SSIS
- NODE SIZE:** Standard_D3_v2 (4 Core(s), 14336 MB)
- RUNNING / REQUESTED NODE(S):** 3/3
- SSISDB SERVER ENDPOINT:** sawinarkssisdb.datab ase.windows.net

Below the summary card, a modal window titled 'myAzureSSISIntegrationRuntime' is open. It contains 'Start' and 'Stop' buttons. The 'Resource ID' field is highlighted with a red box and a copy icon. The Resource ID is: `/subscriptions/cb715d05-3337-4640-8c43-4f943c`. A red callout bubble points to this field with the text: 'You can start/stop your Azure-SSIS IR and copy its resource ID'.

Available Memory	CPU Utilization	Concurrent Jobs (Running/Limit)	Message
12675MB	0%	0/1	
12688MB	0%	0/1	
12625MB	0%	0/1	

Monitoring via ADFv2 App



The screenshot displays the 'Integration Runtimes' monitoring page in the Azure Data Factory v2 portal. The page shows the status of the 'myAzureSSISIntegrationRuntime' as 'Running'. A callout bubble points to the 'SSISDB SERVER ENDPOINT' field, which contains the text 'sawinarkssisdb.datab' and 'ase.windows.net'.

Integration Runtimes / myAzureSSISIntegrationRuntime

Refresh

STATUS	TYPE	NODE SIZE	RUNNING / REQUESTED NODE(S)	SSISDB SERVER ENDPOINT
Running	Azure-SSIS	Standard_D3_v2 (4 Core(s), 14336 MB)	3/3	sawinarkssisdb.datab ase.windows.net

Node Details

Name	Status	Available Memory	CPU Utilization	Concurrent Jobs (Running/Limit)	Message
tvm-1783593343_1-20180126t0606...	Running	12669MB	0%	0/1	
tvm-1783593343_2-20180126t0606...	Running	12687MB	0%	0/1	
tvm-1783593343_3-20180126t0606...	Running	12630MB	0%	0/1	

You can click on your Azure SQL DB/MI server endpoint

Monitoring via ADFv2 App

sawinarkADFv2 | Monitor Integration Runtimes

Integration Runtimes / myAzureSSISIntegrationRuntime

Refresh

STATUS **TYPE** **NODE SIZE** **RUNNING / REQUESTED NODE(S)** **SSISDB SERVER ENDPOINT**

Running Azure-SSIS Standard_D3_v2 (4 Core(s), 14336 MB) 3/3 sawinarkssisdb.database.windows.net

Node Details

Name	Status	Available Memory
tvm-1783593343_1-20180126t0606...	Running	12675MB
tvm-1783593343_2-20180126t0606...	Running	12691MB
tvm-1783593343_3-20180126t0606...	Running	12636MB

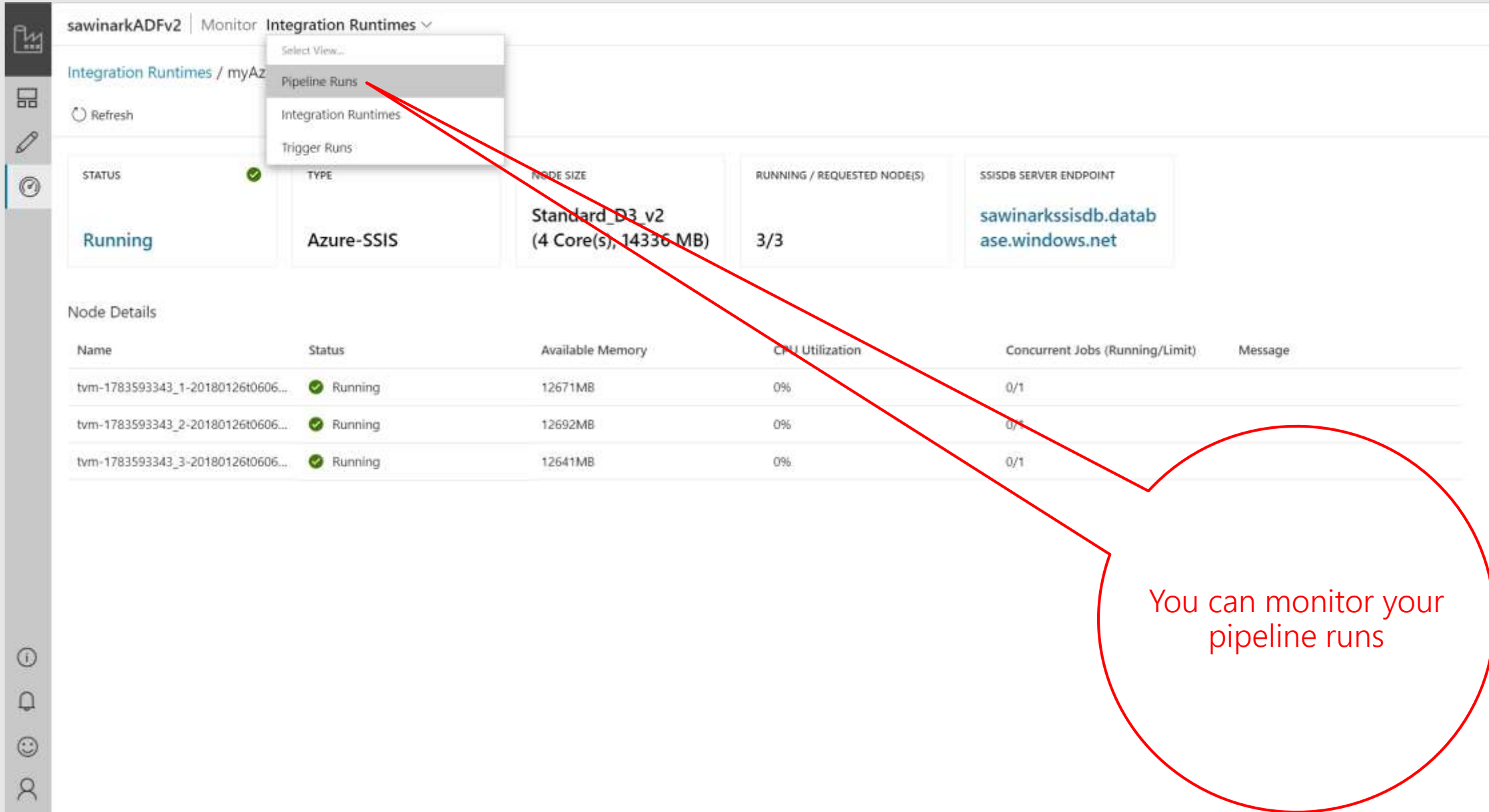
SSISDB Server Endpoint

sawinarkssisdb.database.windows.net

[See your Azure SQL Database/Managed Instance settings.](#)

You can copy your Azure SQL DB/MI server endpoint and monitor your SSISDB

Monitoring via ADFv2 App



The screenshot displays the Azure Data Factory v2 Monitor interface for a workspace named 'sawinarkADFv2'. The 'Integration Runtimes' section is active, showing a single runtime in a 'Running' status. A dropdown menu is open, with 'Pipeline Runs' selected. A red arrow points from this menu to a callout bubble.

Integration Runtimes / myAz

Refresh

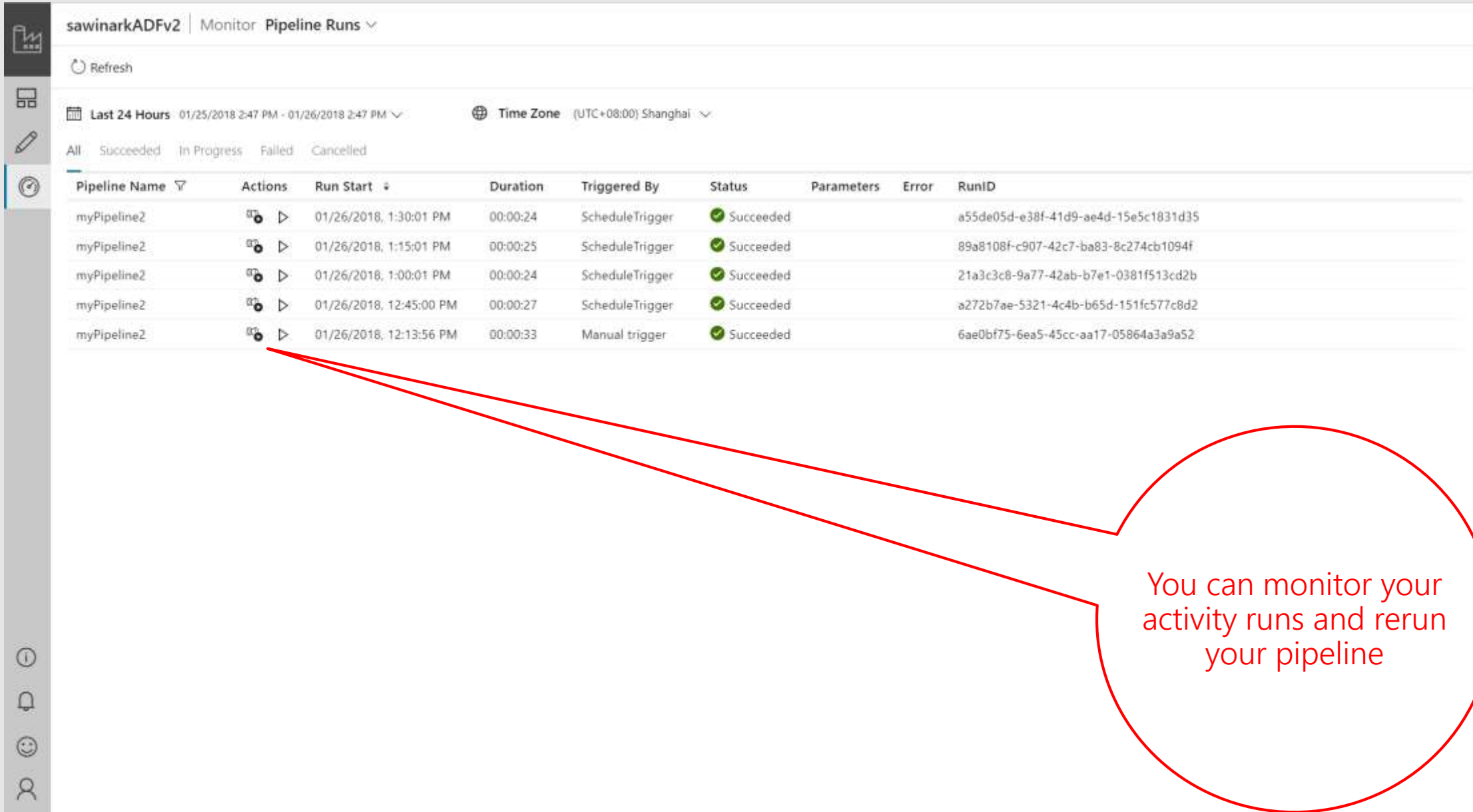
STATUS	TYPE	NODE SIZE	RUNNING / REQUESTED NODE(S)	SSISDB SERVER ENDPOINT
Running	Azure-SSIS	Standard_D3_v2 (4 Core(s), 14336 MB)	3/3	sawinarkssisdb.datab ase.windows.net

Node Details

Name	Status	Available Memory	CPU Utilization	Concurrent Jobs (Running/Limit)	Message
tvm-1783593343_1-20180126t0606...	Running	12671MB	0%	0/1	
tvm-1783593343_2-20180126t0606...	Running	12692MB	0%	0/1	
tvm-1783593343_3-20180126t0606...	Running	12641MB	0%	0/1	

You can monitor your pipeline runs

Monitoring via ADFv2 App


















sawinarkADFv2 | Monitor Pipeline Runs

Refresh

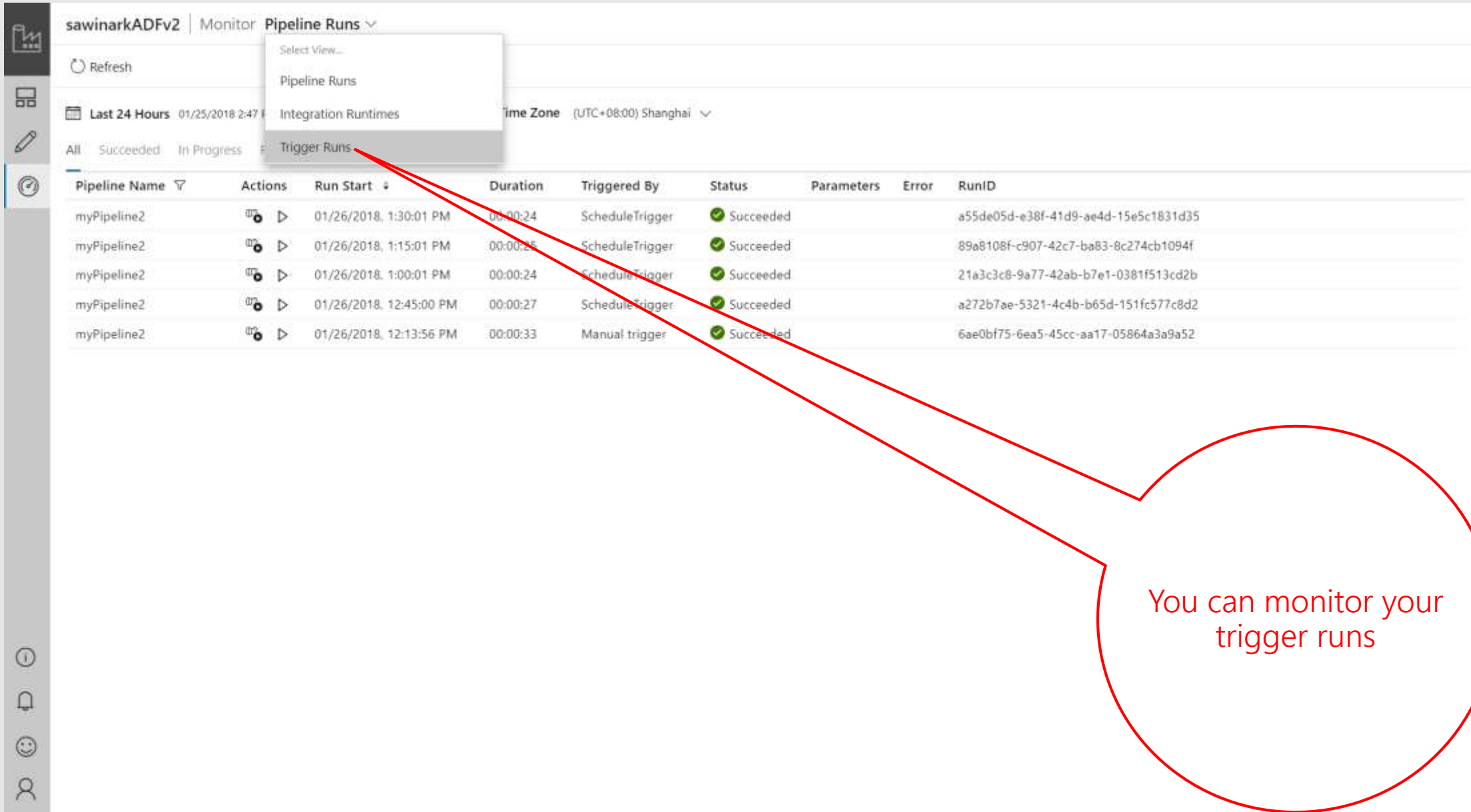
Last 24 Hours: 01/25/2018 2:47 PM - 01/26/2018 2:47 PM | Time Zone: (UTC+08:00) Shanghai

All Succeeded In Progress Failed Cancelled

Pipeline Name	Actions	Run Start	Duration	Triggered By	Status	Parameters	Error	RunID
myPipeline2	 	01/26/2018, 1:30:01 PM	00:00:24	ScheduleTrigger	 Succeeded			a55de05d-e38f-41d9-ae4d-15e5c1831d35
myPipeline2	 	01/26/2018, 1:15:01 PM	00:00:25	ScheduleTrigger	 Succeeded			89a8108f-c907-42c7-ba83-8c274cb1094f
myPipeline2	 	01/26/2018, 1:00:01 PM	00:00:24	ScheduleTrigger	 Succeeded			21a3c3c8-9a77-42ab-b7e1-0381f513cd2b
myPipeline2	 	01/26/2018, 12:45:00 PM	00:00:27	ScheduleTrigger	 Succeeded			a272b7ae-5321-4c4b-b65d-151fc577c8d2
myPipeline2	 	01/26/2018, 12:13:56 PM	00:00:33	Manual trigger	 Succeeded			6ae0bf75-6ea5-45cc-aa17-05864a3a9a52

You can monitor your activity runs and rerun your pipeline

Monitoring via ADFv2 App








sawinarkADFv2 | Monitor Pipeline Runs

Refresh


Last 24 Hours 01/25/2018 2:47 PM Integration Runtimes Time Zone (UTC+08:00) Shanghai





All Succeeded In Progress Trigger Runs





Pipeline Name	Actions	Run Start	Duration	Triggered By	Status	Parameters	Error	RunID
myPipeline2		01/26/2018, 1:30:01 PM	00:00:24	ScheduleTrigger	Succeeded			a55de05d-e38f-41d9-ae4d-15e5c1831d35
myPipeline2		01/26/2018, 1:15:01 PM	00:00:25	ScheduleTrigger	Succeeded			89a8108f-c907-42c7-ba83-8c274cb1094f
myPipeline2		01/26/2018, 1:00:01 PM	00:00:24	ScheduleTrigger	Succeeded			21a3c3c8-9a77-42ab-b7e1-0381f513cd2b
myPipeline2		01/26/2018, 12:45:00 PM	00:00:27	ScheduleTrigger	Succeeded			a272b7ae-5321-4c4b-b65d-151fc577c8d2
myPipeline2		01/26/2018, 12:13:56 PM	00:00:33	Manual trigger	Succeeded			6ae0bf75-6ea5-45cc-aa17-05864a3a9a52


You can monitor your trigger runs


Monitoring via ADFv2 App



















sawinarkADFv2 | Monitor Trigger Runs 

 Refresh

 Last 24 Hours: 01/25/2018 3:05 PM - 01/26/2018 3:05 PM 

 Time Zone (UTC+08:00) Shanghai 

Trigger Name	Trigger Type	Trigger Time	Status	Number of pipelines	Message	Properties	RunID
myTrigger2	ScheduleTrigger	01/26/2018, 1:30:01 PM	 Succeeded	1			08586846622844119879274308136
myTrigger2	ScheduleTrigger	01/26/2018, 1:15:01 PM	 Succeeded	1			08586846631840857539302112062
myTrigger2	ScheduleTrigger	01/26/2018, 1:00:01 PM	 Succeeded	1			08586846640842870323995507796
myTrigger2	ScheduleTrigger	01/26/2018, 12:45:00 PM	 Succeeded	1			08586846649851876216402474013

Monitoring via PSH

```
##### PSH script to monitor Azure-SSIS Integration Runtime and SSIS package executions in ADFv2 #####
```

```
# Query/monitor your Azure SSIS Integration Runtime
```

```
Get-AzureRmDataFactoryV2IntegrationRuntime -DataFactoryName $DataFactoryName -Name $AzureSSISName -ResourceGroupName $ResourceGroupName -Status
```

```
# Query/monitor your pipeline run
```

```
Get-AzureRmDataFactoryV2PipelineRun -ResourceGroupName $ResourceGroupName -DataFactoryName $DataFactoryName -PipelineRunId $myPipelineRun
```

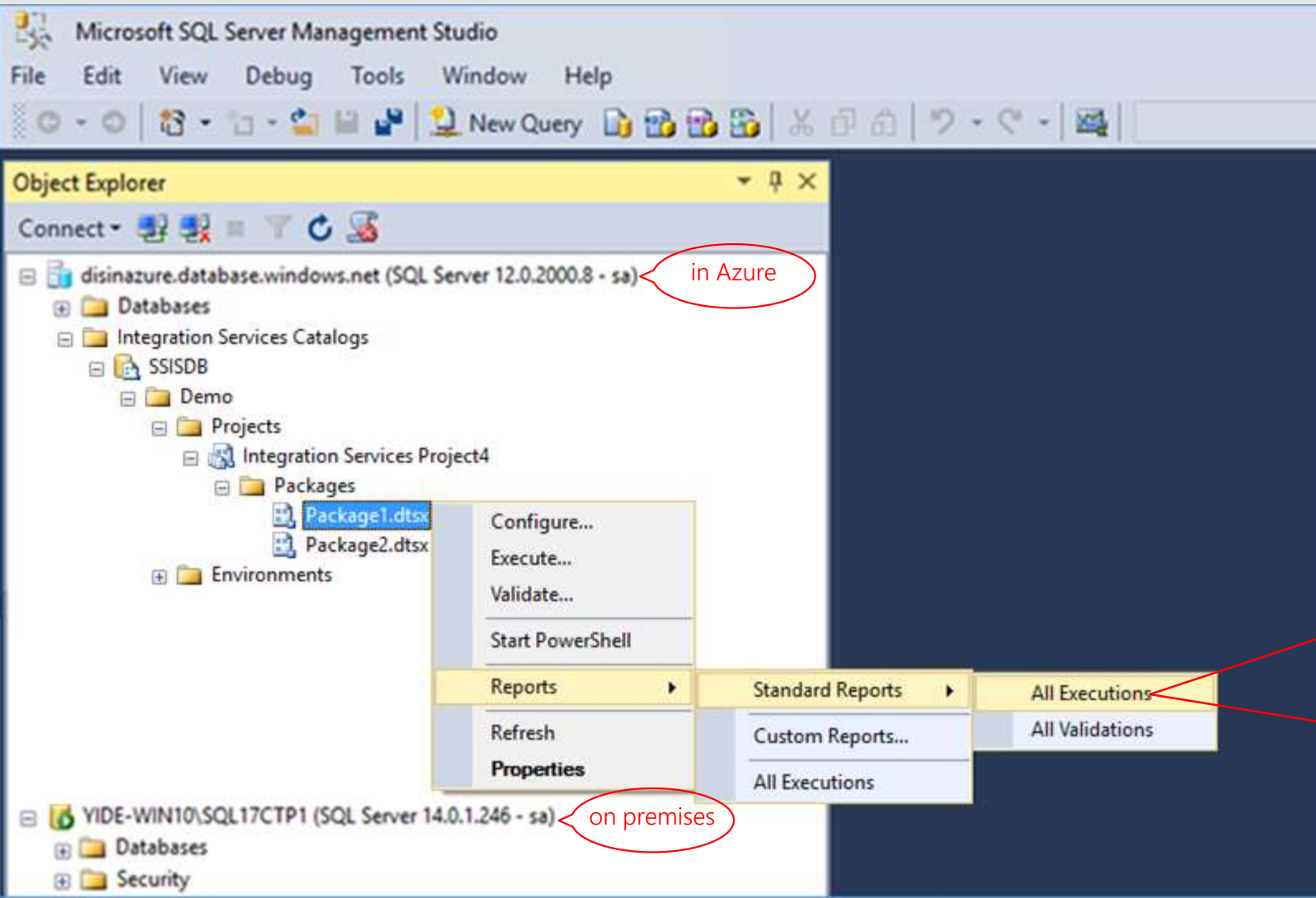
```
# Query/monitor your trigger
```

```
Get-AzureRmDataFactoryV2Trigger -ResourceGroupName $ResourceGroupName -DataFactoryName $DataFactoryName -Name "myTrigger"
```

```
# Query/monitor your trigger runs
```

```
Get-AzureRmDataFactoryV2TriggerRun -ResourceGroupName $ResourceGroupName -DataFactoryName $DataFactoryName -TriggerName "myTrigger" -TriggerRunStartedAfter "2017-12-22" -TriggerRunStartedBefore "2017-12-23"
```


Monitoring via SSMS



You can see reports of all package executions on SSIS PaaS

Monitoring via SSMS

Microsoft SQL Server Management Studio

File Edit View Debug Tools Window Help

Object Explorer

Connect

disinazure.database.windows.net (SQL Server 12.0.2000.8 - sa) **in Azure**

- Databases
- Integration Services Catalogs
 - SSISDB
 - Demo
 - Projects
 - Integration Services Project4
 - Packages
 - Package1.dtsx
 - Package2.dtsx
- Environments

YIDE-WIN10\SQL17CTP1 (SQL Server 14.0.1.246 - sa) **on premises**

- Databases
- Security
- Server Objects
- Replication
- PolyBase
- AlwaysOn High Availability
- Management
- Integration Services Catalogs
- SOL Server Agent (Agent XPs disabled)

All Executions - 1... - CN-SAWINARK-DES

SQL Server

All Executions

on disinazure.database.windows.net at 12/9/2016 11:22:45 AM

This report provides information about the Integration Services package executions that have been performed on the connected SQL Server instance.

Filter: Start time range: 12/3/2016 - 12/9/2016; Status: All; (4 more)

Execution Information

0	0	1	0
Failed	Running	Succeeded	Others

You can see package execution error messages

ID	Status	Report	Folder Name	Project Name	Package Name
5	Succeeded	Overview All Messages Execution Performance	Demo	Integration Services Project4	Package1.dtsx

On-Demand/Just-In-Time Provisioning

On-Demand/Just-In-Time Provisioning

- Starting/stopping Azure-SSIS IR can be invoked/scheduled via Azure Automation
- Starting/stopping Azure-SSIS IR can be invoked/scheduled using ADFv2 Web Activity that triggers Azure Automation via webhooks
- Starting/stopping Azure-SSIS IR can be done on demand/just in time before/after executing packages by chaining ADFv2 Web and Sproc Activities

On-Demand/Just-In-Time Provisioning

Microsoft Azure

Home > New > Add Automation Account

New

Search the Marketplace

Azure Marketplace

Featured

- Get started
- Recently created
- Compute
- Networking
- Storage
- Web + Mobile
- Containers
- Databases
- Data + Analytics
- AI + Cognitive Services
- Internet of Things
- Enterprise Integration
- Security + Identity
- Developer tools
- Monitoring + Management
- Add-ons
- Blockchain

Azure Migrate (preview)

Application Insights

Log Analytics

Automation

Backup and Site Recovery (OMS)

Intune App Protection

Scheduler

Dynatrace

Veeam Cloud Connect for the Enterprise

Add Automation Account

Name

sawinarkAutomation

Subscription

EIMS_TEST_EVEREST_1

Resource group

Create new Use existing

sawinarkRG

Location

East US 2

Create Azure Run As account

Yes No

The Run As account feature will create a Run As account and a Classic Run As account. Click here to learn more about Run As accounts.

Learn more about Automation pricing.

Pin to dashboard

Create

Create Automation account if you have not done so already

On-Demand/Just-In-Time Provisioning

Microsoft Azure

Report a bug

Search resources, services and docs

sawinark@microsoft.c...
MICROSOFT

Dashboard

+ New dashboard

Edit dashboard

Share

Fullscreen

Clone

Delete

All resources
ALL SUBSCRIPTIONS

Refresh

sawinarkRunbookRunbook

sawinarkAutomationAutomation Account

sawinarkssisdbSQL server

disinazureSQL server

sawinarkWorker2Virtual machine

sawinarkWorker1Virtual machine

sawinarkMasterVirtual machine

sawinarkClassicVNetVirtual network (classic)

sawinarkARMVNetVirtual network

SSISDBSQL database

ScaleOutDemo-vnetVirtual network

sawinarkSQLDB2SQL database

ssisazurefileshareStorage account

See more...

sawinarkSQLDB
SQL DATABASE

Online

sawinarkSQLDB2
SQL DATABASE

Online

sawinarkstorage

Available

ssisazurefileshare

Available

sawinarkadls
DATA LAKE STORE

Running

sawinarkMaster

Stopped

sawinarkWorker1

Stopped

sawinarkWorker2

Stopped

sawinarkClassicVNet

sawinarkARMVNet

sawinarkAutomation
AUTOMATION ACCOUNT

Active

sawinarkADF
DATA FACTORY

sawinarkADFv1
DATA FACTORY

sawinarkADFv2
DATA FACTORY

sawinarkssisdb
SQL SERVER

Available

disinazure
SQL SERVER

Available

Open Automation settings page

On-Demand/Just-In-Time Provisioning

The screenshot displays the Microsoft Azure portal interface for an Automation Account named 'sawinarkAutomation'. The left-hand navigation pane is expanded, showing various options under 'PROCESS AUTOMATION' and 'SHARED RESOURCES'. The 'Modules' option under 'SHARED RESOURCES' is highlighted with a red circle and an arrow pointing to it. The main content area shows the 'Job Statistics' section, which includes a donut chart with a large '0' in the center, indicating that there are no jobs currently running. A legend on the right side of the chart lists job statuses: QUEUED, RUNNING, COMPLETED, FAILED, STOPPED, and SUSPENDED. The 'Modules' link in the sidebar is the target of the instruction 'Click on "Modules"'.

On-Demand/Just-In-Time Provisioning

Microsoft Azure

Report a bug

Search resources, services and docs

sawinark@microsoft.c...

Home > sawinarkAutomation - Modules

sawinarkAutomation - Modules

Update management (Prev...

PROCESS AUTOMATION

Runbooks

Jobs

Runbooks gallery

Hybrid worker groups

Watcher tasks

SHARED RESOURCES

Schedules

Modules

Modules gallery

Credentials

Connections

Certificates

Variables

RELATED RESOURCES

Workspace

Unlink workspace

Event grid

Search (Ctrl+F)

Add a module

Update Azure Modules

Browse gallery

Refresh

NAME	LAST MODIFIED	STATUS	VERSION
Azure	1/17/2018, 7:26 AM	Available	1.0.3
Azure.Storage	1/17/2018, 7:33 AM	Available	1.0.3
AzureRM.Automation	1/17/2018, 7:31 AM	Available	1.0.3
AzureRM.Compute	1/17/2018, 7:30 AM	Available	1.2.1
AzureRM.Profile	1/17/2018, 7:30 AM	Available	1.0.3
AzureRM.Resources	1/17/2018, 7:31 AM	Available	1.0.3
AzureRM.Sql	1/17/2018, 7:32 AM	Available	1.0.3
AzureRM.Storage	1/17/2018, 7:32 AM	Available	1.0.3
Microsoft.PowerShell.Core	1/17/2018, 7:27 AM	Available	0.0
Microsoft.PowerShell.Diagnostics	1/17/2018, 7:27 AM	Available	
Microsoft.PowerShell.Management	1/17/2018, 7:28 AM	Available	
Microsoft.PowerShell.Security	1/17/2018, 7:28 AM	Available	
Microsoft.PowerShell.Utility	1/17/2018, 7:29 AM	Available	
Microsoft.WSMan.Management	1/17/2018, 7:29 AM	Available	
Orchestrator.AssetManagement.Cmdlets	1/17/2018, 7:33 AM	Available	1.0

If the latest AzureRM.Profile and AzureRM.DataFactoryV2 modules are not listed, click on "Browse gallery"

On-Demand/Just-In-Time Provisioning

The screenshot shows the Microsoft Azure PowerShell Gallery interface. The search bar at the top left contains the text 'datafactory'. Below the search bar, two modules are listed: 'AzureRM.DataFactories' and 'AzureRM.DataFactoryV2'. The 'AzureRM.DataFactoryV2' module is highlighted in blue. To the right of the search results, the details for the 'AzureRM.DataFactoryV2' module are displayed. The 'Import' button is located at the top of the details panel. A red circle is drawn around the search results, and a red arrow points from the circle to the 'Import' button.

Microsoft Azure

Report a bug

Search resources, services and docs

Home > sawinarkAutomation > Modules > Browse Gallery > AzureRM.DataFactoryV2

Browse Gallery

datafactory

AzureRM.DataFactories
Microsoft Azure PowerShell - DataFactories service cmdlets for Azure Resource Manager
Tags: Azure ResourceManager ARM DataFactory PSModule
Created by: azure-sdk
964690 downloads
Last updated: 2/1/2018

AzureRM.DataFactoryV2
Microsoft Azure PowerShell - DataFactories service cmdlets for Azure Resource Manager
Tags: Azure ResourceManager ARM DataFactory PSModule
Created by: azure-sdk
506764 downloads
Last updated: 2/1/2018

AzureRM.DataFactoryV2
PowerShell Module

Import

Microsoft Azure PowerShell - DataFactories service cmdlets for Azure Resource Manager

Created by: azure-sdk
Tags: Azure ResourceManager ARM DataFactory PSModule
Dependencies: AzureRM.Profile (≥ 4.2.0)
[View Source Project](#)

Version: 0.5.0
Downloads: 506,764
Last updated: 2/1/2018

Learn more
[View in PowerShell Gallery](#)
[Documentation](#)
[Licensing information](#)

Content

Search to filter items...

TYPE	NAME
Cmdlet	Set-AzureRmDataFactoryV2
Cmdlet	Update-AzureRmDataFactoryV2
Cmdlet	Get-AzureRmDataFactoryV2
Cmdlet	Remove-AzureRmDataFactoryV2
Cmdlet	Set-AzureRmDataFactoryV2LinkService
Cmdlet	New-AzureRmDataFactoryV2LinkServiceEncryptedCredential
Cmdlet	Get-AzureRmDataFactoryV2LinkService
Cmdlet	Remove-AzureRmDataFactoryV2LinkService
Cmdlet	Set-AzureRmDataFactoryV2Dataset
Cmdlet	Get-AzureRmDataFactoryV2Dataset

Search for the required modules and import them

On-Demand/Just-In-Time Provisioning

Microsoft Azure

Report a bug Search resources, services and docs

sawinarkAutomation - Modules

Automation Account

Search (Ctrl+F)

+ Add a module: Update Azure Modules: Browse gallery Refresh

✓ Azure modules have been updated.

NAME	LAST MODIFIED	STATUS	VERSION
Azure	2/7/2018, 12:28 AM	Available	5.1.1
Azure.Storage	2/7/2018, 12:26 AM	Available	4.1.0
AzureRM.Automation	2/7/2018, 12:26 AM	Available	4.2.0
AzureRM.Compute	2/7/2018, 12:26 AM	Available	4.2.0
AzureRM.DataFactoryV2	2/7/2018, 12:27 AM	Available	0.5.0
AzureRM.Profile	2/7/2018, 12:24 AM	Available	4.2.0
AzureRM.Resources	2/7/2018, 12:26 AM	Available	5.2.0
AzureRM.Sql	2/7/2018, 12:26 AM	Available	4.2.0
AzureRM.Storage	2/7/2018, 12:28 AM	Available	4.2.0
Microsoft.PowerShell.Core	1/17/2018, 7:27 AM	Available	0.0
Microsoft.PowerShell.Diagnostics	1/17/2018, 7:27 AM	Available	
Microsoft.PowerShell.Management	1/17/2018, 7:28 AM	Available	
Microsoft.PowerShell.Security	1/17/2018, 7:28 AM	Available	
Microsoft.PowerShell.Utility	1/17/2018, 7:29 AM	Available	
Microsoft.WSMan.Management	1/17/2018, 7:29 AM	Available	
Orchestrator.AssetManagement.Cmdlets	1/17/2018, 7:33 AM	Available	1.0

Update management (Prev...)

PROCESS AUTOMATION

- Runbooks
- Jobs
- Runbooks gallery
- Hybrid worker groups
- Watcher tasks

SHARED RESOURCES

- Schedules
- Modules
- Modules gallery
- Credentials
- Connections
- Certificates
- Variables

RELATED RESOURCES

- Workspace
- Unlink workspace
- Event grid

Click on "Runbooks"

On-Demand/Just-In-Time Provisioning

The screenshot shows the Microsoft Azure portal interface for the 'sawinarkAutomation - Runbooks' section. The left sidebar contains navigation links for Update management, Runbooks, Jobs, Runbooks gallery, Hybrid worker groups, Watcher tasks, Schedules, Modules, Modules gallery, Credentials, Connections, Certificates, Variables, and Workspace. The main content area displays a table of runbooks with columns for NAME, AUTHORIZING STATUS, LAST MODIFIED, and TAGS. A red arrow points to the '+ Add a runbook' button in the top left of the main content area. A red circle highlights the text 'Click on \"+ Add a runbook\"'.

NAME	AUTHORIZING STATUS	LAST MODIFIED	TAGS
AzureAutomationTutorial	✓ Published	2/6/2018, 11:51 PM	
AzureAutomationTutorialPython2	✓ Published	2/6/2018, 11:51 PM	
AzureAutomationTutorialScript	✓ Published	2/6/2018, 11:51 PM	
AzureClassicAutomationTutorial	✓ Published	2/6/2018, 11:51 PM	
AzureClassicAutomationTutorialScript	✓ Published	2/6/2018, 11:51 PM	

Click on "+ Add a runbook"

On-Demand/Just-In-Time Provisioning

The screenshot displays the Microsoft Azure portal interface for creating a new runbook. The 'Add Runbook' dialog is open, showing the 'Runbook' configuration page. The 'Name' field is set to 'sawinarkRunbook', the 'Runbook type' is 'PowerShell', and the 'Description' is 'Start/Stop Azure-SSIS IR'. A red callout bubble points to the 'Create' button with the text 'Create PSH runbook to start/stop your Azure-SSIS IR'.

Microsoft Azure

Home > sawinarkAutomation > Runbooks > Add Runbook > Runbook

Report a bug Search resources, services and docs

sawinark@microsoft.c... MICROSOFT

Add Runbook

Quick Create
Create a new runbook

Import
Import an existing runbook

Runbook

* Name [?]
sawinarkRunbook ✓

* Runbook type [?]
PowerShell ▼

Description
Start/Stop Azure-SSIS IR ✓

Create

Create PSH runbook to start/stop your Azure-SSIS IR

On-Demand/Just-In-Time Provisioning

Microsoft Azure

Home > sawinarkAutomation > Runbooks > sawinarkRunbook > Edit PowerShell Runbook*

Edit PowerShell Runbook*

Save Publish Revert to published Check in Test pane Feedback

CMDLETS
RUNBOOKS
ASSETS

```
1 Param
2 {
3     [Parameter (Mandatory= $true)]
4     [String] $ResourceGroupName,
5
6     [Parameter (Mandatory= $true)]
7     [String] $DataFactoryName,
8
9     [Parameter (Mandatory= $true)]
10    [String] $AzureSSISName,
11
12    [Parameter (Mandatory= $true)]
13    [String] $Operation
14 }
15
16 $connectionName = "AzureRunAsConnection"
17 try
18 {
19     # Get the connection "AzureRunAsConnection "
20     $servicePrincipalConnection=Get-AutomationConnection -Name $connectionName
21
22     "Logging in to Azure..."
23     Add-AzureRmAccount `
24         -ServicePrincipal `
25         -TenantId $servicePrincipalConnection.TenantId `
26         -ApplicationId $servicePrincipalConnection.ApplicationId `
27         -CertificateThumbprint $servicePrincipalConnection.CertificateThumbprint
28 }
29 catch {
30     if (!$servicePrincipalConnection)
31     {
32         $ErrorMessage = "Connection $connectionName not found."
33         throw $ErrorMessage
34     } else{
35         Write-Error -Message $_.Exception
36         throw $_.Exception
37     }
38 }
```

Enter PSH script, then save and publish your runbook

On-Demand/Just-In-Time Provisioning

```
##### PSH script for Automation runbook to start/stop Azure-SSIS IR #####
```

```
Param
```

```
(
```

```
    [Parameter (Mandatory= $true)]
```

```
    [String] $ResourceGroupName,
```

```
    [Parameter (Mandatory= $true)]
```

```
    [String] $DataFactoryName,
```

```
    [Parameter (Mandatory= $true)]
```

```
    [String] $AzureSSISName,
```

```
    [Parameter (Mandatory= $true)]
```

```
    [String] $Operation
```

```
)
```

```
$connectionName = "AzureRunAsConnection"
```

```
try
```

```
{
```

```
    # Get the connection "AzureRunAsConnection "
```

```
    $servicePrincipalConnection=Get-AutomationConnection -Name $connectionName
```

```
    "Logging in to Azure..."
```

```
    Add-AzureRmAccount `
```

```
        -ServicePrincipal `
```

```
        -TenantId $servicePrincipalConnection.TenantId `
```

```
        -ApplicationId $servicePrincipalConnection.ApplicationId `
```

```
        -CertificateThumbprint $servicePrincipalConnection.CertificateThumbprint
```

```
}
```

On-Demand/Just-In-Time Provisioning

```
catch {  
    if (!$servicePrincipalConnection)  
    {  
        $ErrorMessage = "Connection $connectionName not found."  
        throw $ErrorMessage  
    } else{  
        Write-Error -Message $_.Exception  
        throw $_.Exception  
    }  
}
```

```
if($Operation -eq "START" -or $operation -eq "start")
```

```
{
```

```
    "##### Starting #####"
```

```
    Start-AzureRmDataFactoryV2IntegrationRuntime -ResourceGroupName $ResourceGroupName -DataFactoryName $DataFactoryName -Name $AzureSSISName -Force
```

```
}
```

```
elseif($Operation -eq "STOP" -or $operation -eq "stop")
```

```
{
```

```
    "##### Stopping #####"
```

```
    Stop-AzureRmDataFactoryV2IntegrationRuntime -DataFactoryName $DataFactoryName -Name $AzureSSISName -ResourceGroupName $ResourceGroupName -Force
```

```
}
```

```
"##### Completed #####"
```


On-Demand/Just-In-Time Provisioning

The screenshot displays the Microsoft Azure portal interface for managing an Automation Runbook. The breadcrumb navigation at the top indicates the path: Home > sawinarkAutomation > Runbooks > sawinarkRunbook > Start Runbook. The left-hand navigation pane includes sections for Overview, Activity log, Tags, Diagnose and solve problems, RESOURCES (Jobs, Schedules, Webhooks), RUNBOOK SETTINGS (Properties, Description, Logging and tracing), SETTINGS (Locks, Automation script), and SUPPORT + TROUBLESHOOTING (New support request). The main content area features a toolbar with buttons for Start, View, Edit, Schedule, Webhook, Delete, Export, and Refresh. Below the toolbar, the 'Essentials' section lists metadata: Resource group (sawinarkRG), Account (sawinarkAutomation), Location (East US 2), and Subscription name (EIMS_TEST_EVEREST_1). To the right, a 'Status' box shows Published status, Runbook type (PowerShell Runbook), and Last modified (2/7/2018, 12:50 AM). The 'Details' section contains a 'Jobs' card with a checkmark icon. A red arrow originates from the 'Start' button and points to a large red circle on the right side of the screen, which contains the text 'Test your runbook'.

Microsoft Azure

Report a bug

Search resources, services and docs

sawinark@microsoft.c... MICROSOFT

Home > sawinarkAutomation > Runbooks > sawinarkRunbook > Start Runbook

sawinarkRunbook Runbook

Search (Ctrl+J)

Start View Edit Schedule Webhook Delete Export Refresh

Essentials

Resource group sawinarkRG

Account sawinarkAutomation

Location East US 2

Subscription name EIMS_TEST_EVEREST_1

Status Published

Runbook type PowerShell Runbook

Last modified 2/7/2018, 12:50 AM

Details

Jobs

Test your runbook

On-Demand/Just-In-Time Provisioning

Microsoft Azure

Home > sawinarkAutomation > Runbooks > sawinarkRunbook > Start Runbook

Start Runbook
sawinarkRunbook

Parameters

- * RESOURCEGROUPNAME ⓘ
sawinarkRG ✓
Mandatory; String
- * DATAFACTORYNAME ⓘ
sawinarkADFv2 ✓
Mandatory; String
- * AZURESSISNAME ⓘ
myAzureSSISIntegrationRuntime ✓
Mandatory; String
- * OPERATION ⓘ
STOP ✓
Mandatory; String

Run Settings

Run on Azure ⓘ

OK

Enter runbook parameters to start/stop your Azure-SSIS IR

On-Demand/Just-In-Time Provisioning

The screenshot displays the Microsoft Azure portal interface for a specific Runbook job. The breadcrumb navigation at the top indicates the path: Home > sawinarkAutomation > Runbooks > sawinarkRunbook > sawinarkRunbook 2/7/2018, 12:59 AM > All Logs.

Job Details (Essentials):

- Job id: d07379b2-f8bc-49ce-bc5d-f27a00415a55
- Created: 2/7/2018, 12:59 AM
- Job status: Completed
- Last Update: 2/7/2018, 1:00 AM
- Run As: User
- Runbook: sawinarkRunbook
- Ran on: Azure
- Source snapshot: View source snapshot

Overview Summary:

- Input: 4
- Output: 1
- Errors: 0
- Warnings: 0
- Exception: None

All Logs Table:

TIME	TYPE	DETAILS
2/7/2018, 12:59:47 AM	Output	Logging in to Azure...
2/7/2018, 12:59:51 AM	Output	
2/7/2018, 12:59:52 AM	Output	***** Stopping *****
2/7/2018, 1:00:59 AM	Output	***** Completed *****

A red circle highlights the 'All Logs' tab in the Overview section and the corresponding log entries in the 'All Logs' pane. A red arrow points from the text 'Check "Output"/"All Logs"' to the highlighted area.

On-Demand/Just-In-Time Provisioning

Microsoft Azure

Home > sawinarkAutomation - Runbooks > sawinarkRunbook - Schedules > Schedule Runbook

sawinarkRunbook - Schedules

Search (Ctrl+/)

+ Add a schedule Refresh

NAME	NEXT RUN	STATUS
No schedules found.		

Click on "Schedules" and "+ Add a schedule"

On-Demand/Just-In-Time Provisioning

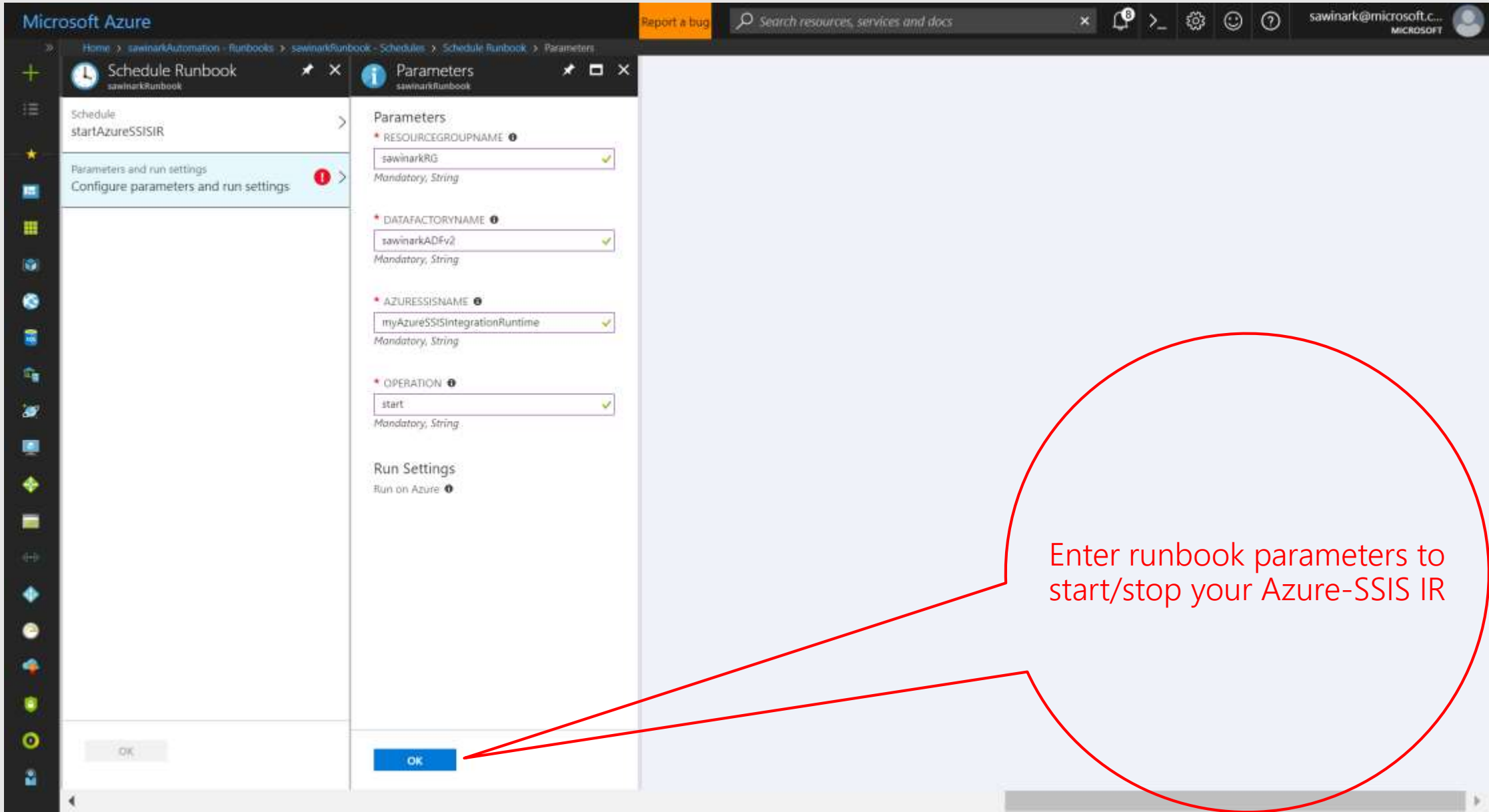
The screenshot displays the Microsoft Azure portal interface for configuring a new schedule for a Runbook. The breadcrumb navigation at the top indicates the path: Home > sawinarkAutomation > Runbooks > sawinarkRunbook > Schedules > Schedule Runbook > Schedule > New Schedule.

The interface is divided into three main sections:

- Left Panel:** Contains a 'Schedule Runbook' section with a link to 'Link a schedule to your runbook' and a 'Parameters and run settings' section with a warning icon and the text 'Configure parameters and run settings'.
- Middle Panel:** Displays a 'Create a new schedule' button and a message stating 'No schedules found.'
- Right Panel (Form):** Contains the following fields and options:
 - Name:** A text box containing 'startAzureSSISIR' with a green checkmark.
 - Description:** An empty text box.
 - Starts:** A date and time selector showing '2018-02-07' and '6:00:00 AM'.
 - Time Zone:** A dropdown menu set to 'China - China Time'.
 - Recurrence:** Two buttons, 'Once' and 'Recurring', with 'Recurring' selected.
 - Recur every:** A text box with '1' and a dropdown menu set to 'Day'.
 - Set expiration:** Two buttons, 'Yes' and 'No', with 'No' selected.
 - Expires:** A dropdown menu set to 'Never'.

A red callout bubble with the text 'Create a schedule for your runbook' points to the 'Create' button at the bottom right of the form.

On-Demand/Just-In-Time Provisioning



The screenshot displays the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo, a search bar, and the user's email 'sawinark@microsoft.c...'. The main content area is divided into two panes. The left pane, titled 'Schedule Runbook', shows a list of runbooks with 'startAzureSSISIR' selected. The right pane, titled 'Parameters', shows the configuration for this runbook. It includes four mandatory string parameters: 'RESOURCEGROUPNAME' (value: sawinarkRG), 'DATAFACTORYNAME' (value: sawinarkADFv2), 'AZURESSISNAME' (value: myAzureSSISIntegrationRuntime), and 'OPERATION' (value: start). Below these is the 'Run Settings' section with 'Run on Azure' selected. A red callout bubble with the text 'Enter runbook parameters to start/stop your Azure-SSIS IR' points to the 'OK' button at the bottom right of the 'Parameters' pane.

Microsoft Azure

Home > sawinarkAutomation > Runbooks > sawinarkRunbook > Schedules > Schedule Runbook > Parameters

Schedule Runbook
sawinarkRunbook

Schedule
startAzureSSISIR

Parameters and run settings
Configure parameters and run settings

Parameters
sawinarkRunbook

* RESOURCEGROUPNAME ⓘ
sawinarkRG ✓
Mandatory; String

* DATAFACTORYNAME ⓘ
sawinarkADFv2 ✓
Mandatory; String

* AZURESSISNAME ⓘ
myAzureSSISIntegrationRuntime ✓
Mandatory; String

* OPERATION ⓘ
start ✓
Mandatory; String

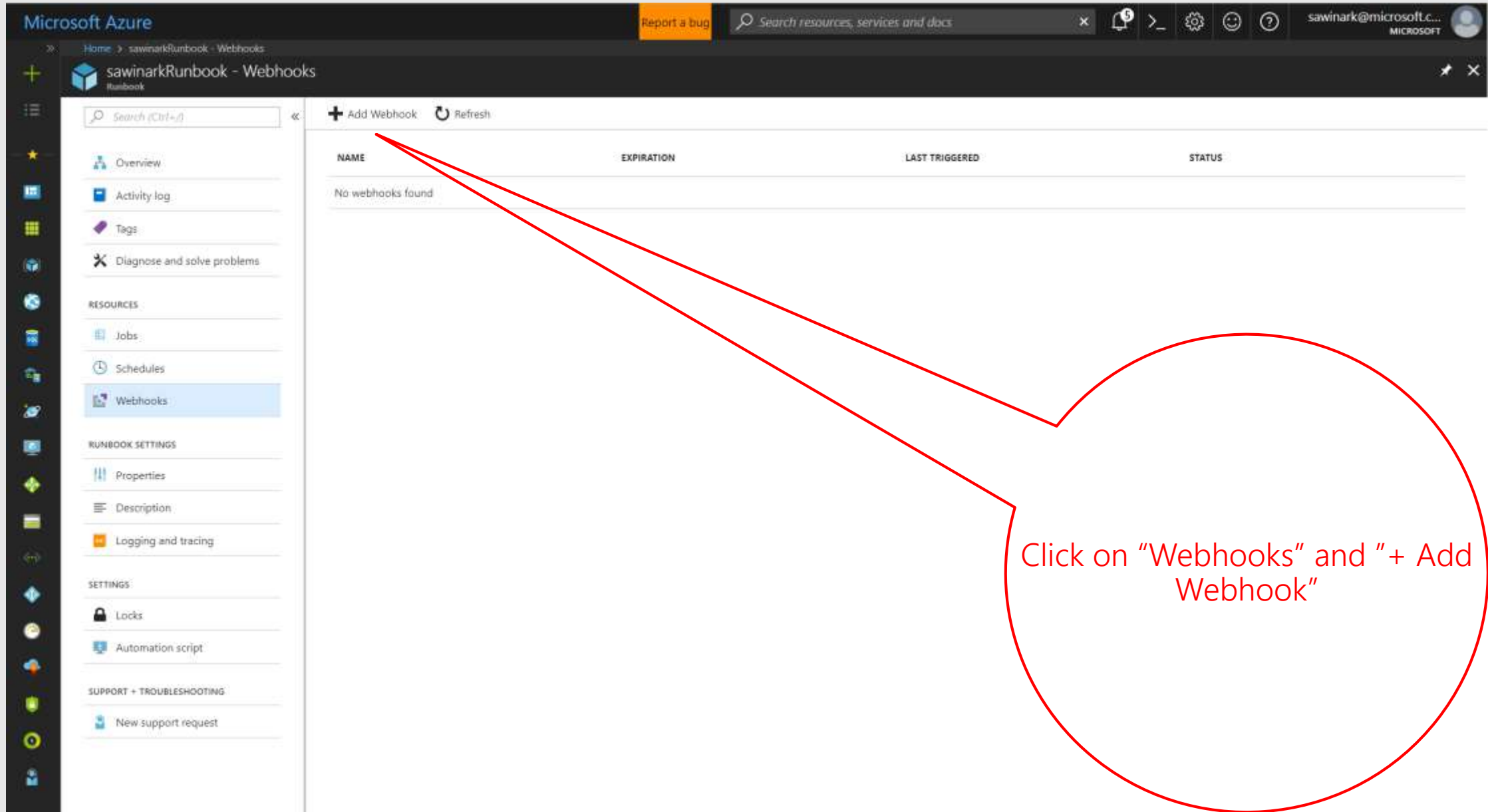
Run Settings
Run on Azure ⓘ

OK

OK

Enter runbook parameters to start/stop your Azure-SSIS IR

On-Demand/Just-In-Time Provisioning



The screenshot displays the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo, a 'Report a bug' button, a search bar, and user information for 'sawinark@microsoft.com'. The left sidebar shows the 'sawinarkRunbook - Webhooks' page, with the 'Webhooks' link highlighted under the 'RESOURCES' section. The main content area shows a table with columns 'NAME', 'EXPIRATION', 'LAST TRIGGERED', and 'STATUS'. The table is currently empty, displaying 'No webhooks found'. A red circle highlights the '+ Add Webhook' button and the 'Webhooks' link in the left sidebar, with a red arrow pointing from the circle to the button.

Click on "Webhooks" and "+ Add Webhook"

On-Demand/Just-In-Time Provisioning

Microsoft Azure

Home > sawinarkAutomation - Runbooks > sawinarkRunbook - Webhooks > Add Webhook > Create a new webhook

Report a bug Search resources, services and docs

9 > ? sawinark@microsoft.c... MICROSOFT

+

Webhook

Create new webhook

Parameters and run settings

Configure parameters and run settings

Start a runbook via a simple HTTP POST to a URL

For security, after creating a webhook its URL can't be viewed. Make sure to copy it before pressing "OK", and to store it securely. [Learn more](#)

* Name

startAzureSSISIR

* Enabled

Yes No

* Expires

2019-02-07 12:00:00 AM

URL

https://stevents.azure-automation.net/...

Create OK

Create webhooks for your runbook and save your URLs

On-Demand/Just-In-Time Provisioning

Microsoft Azure

Home > sawinarkAutomation > Runbooks > sawinarkRunbook > Webhooks > Add Webhook > Parameters

Add Webhook

Start a runbook via a simple HTTP POST to a URL

Webhook: startAzureSSISIR

Parameters and run settings
Configure parameters and run settings

Parameters

- * RESOURCEGROUPNAME ⓘ
sawinarkRG ✓
Mandatory, String
- * DATAFACTORYNAME ⓘ
sawinarkADFv2 ✓
Mandatory, String
- * AZURESSISNAME ⓘ
myAzureSSISIntegrationRuntime ✓
Mandatory, String
- * OPERATION ⓘ
start ✓
Mandatory, String

Run Settings
Run on Azure ⓘ

Create OK

Enter runbook parameters to start/stop your Azure-SSIS IR

On-Demand/Just-In-Time Provisioning

The screenshot displays the Azure Data Factory (ADFv2) interface. On the left, the 'Pipelines' section lists 'myPipeline', 'myPipeline2', and '* myPipeline3'. The 'Activities' pane on the right shows a 'Web' activity named 'startMyIR'. The 'General' tab of the activity configuration is active, showing fields for Name, Description, Timeout, Retry, and Retry interval. A red circle highlights the 'Description' field, with a red arrow pointing to it from the text 'Create ADFv2 pipeline with a web activity'.

Activities

Search Activities

Batch Service

Data Flow

Data Lake Analytics

General

Execute Pipeline

Get Metadata

Lookup

Stored Procedure

Web

HDInsight

Iteration & Conditionals

Web

startMyIR

Name *

Description

Timeout

7.00:00:00

Retry

0

Retry interval

30

Create ADFv2 pipeline with a web activity

On-Demand/Just-In-Time Provisioning

The screenshot displays the Azure Data Factory (ADF) interface. On the left, the 'Pipelines' section lists 'myPipeline', 'myPipeline2', and 'myPipeline3'. The 'Activities' pane on the right shows a 'Web' activity named 'startMyIR'. The 'Settings' tab for this activity is selected, showing the following configuration:

- URL:** `https://s1events.azure-automation.net/web`
- Method:** `POST`
- Headers:** No headers specified
- Body:** `("message": "Start my IR")`
- Datasets:** (Empty)
- Linkedservices:** `Select...`

A red circle highlights the 'URL' and 'Method' fields, with a red arrow pointing to them from a text box that reads: 'Enter the webhook URL to start your Azure-SSIS IR and select POST method'.

On-Demand/Just-In-Time Provisioning

The screenshot shows the Azure Data Factory portal interface. On the left, the 'Pipelines' section lists 'myPipeline', 'myPipeline2', and 'myPipeline3'. The 'myPipeline3' pipeline is selected, and its details are shown in the main pane. The pipeline consists of two activities: a 'Web' activity named 'startMyIR' and a 'Stored Procedure' activity named 'mySprocActivity'. The 'Stored Procedure' activity is highlighted with a red callout bubble containing the text: 'Add a sproc activity to execute your package(s)'.

The 'Stored Procedure' activity configuration is shown in the 'Stored Procedure' tab. The 'Stored procedure name' is set to 'sp_executesql'. The 'Import parameter' section shows a parameter named 'stmt' of type 'string'. The 'Stored procedure parameters' section shows a table with the following data:

NAME	TYPE	VALUE
stmt	string	

The 'Stored procedure parameters' section also includes a 'New' button and a 'Delete' button. The 'Stored procedure parameters' section is expanded, showing the following SQL code:

```
DECLARE @return_value INT, @exe_id BIGINT, @err_msg NVARCHAR(150) WHILE NOT EXISTS (SELECT * FROM [SSISDB].[catalog].[worker_agents] WHERE IsEnabled = 1 AND LastOnlineTime > DATEADD(MINUTE, -10, SYSDATETIMEOFFSET(0))) BEGIN WAITFOR DELAY '00:00:01'; END EXEC @return_value=[SSISDB].[catalog].[create_execution] @folder_name=N'demo', @project_name=N'ScaleOutProject', @package_name=N'SQLDBTable2toSQLDB2Table2.dtsx', @use32bitruntime=0, @runinscaleout=1, @useanyworker=1, @execution_id=@exe_id OUTPUT EXEC [SSISDB].[catalog].[set_execution_parameter_value] @exe_id, @object_type
```

On-Demand/Just-In-Time Provisioning

```
-- T-SQL script to create/start SSIS package execution using SSISDB sprocs
```

```
DECLARE    @return_value int, @exe_id bigint, @err_msg nvarchar(150)
```

```
-- Wait until Azure-SSIS IR is started
```

```
WHILE NOT EXISTS (SELECT * FROM [SSISDB].[catalog].[worker_agents] WHERE IsEnabled = 1 AND LastOnlineTime > DATEADD(MINUTE, -10, SYSDATETIMEOFFSET()))
```

```
BEGIN
```

```
    WAITFOR DELAY '00:00:01';
```

```
END
```

```
EXEC @return_value = [SSISDB].[catalog].[create_execution] @folder_name=N'YourFolder', @project_name=N'YourProject',  
    @package_name=N'YourPackage', @use32bitruntime=0, @runincluster=1, @useanyworker=1,  
    @execution_id=@exe_id OUTPUT
```

```
-- To synchronize SSIS package execution, set SYNCHRONIZED execution parameter
```

```
EXEC [SSISDB].[catalog].[set_execution_parameter_value] @exe_id, @object_type=50, @parameter_name=N'SYNCHRONIZED', @parameter_value=1
```

```
EXEC [SSISDB].[catalog].[start_execution] @execution_id = @exe_id, @retry_count = 0
```

```
-- Raise an error for unsuccessful package execution, check package execution status = created (1)/running (2)/canceled (3)/failed (4)/
```

```
-- pending (5)/ended unexpectedly (6)/succeeded (7)/stopping (8)/completed (9)
```

```
IF (SELECT [status] FROM [SSISDB].[catalog].[executions] WHERE execution_id = @exe_id)<>7
```

```
BEGIN
```

```
    SET @err_msg=N'Your package execution did not succeed for execution ID: '+ CAST(@execution_id as nvarchar(20))
```

```
    RAISERROR(@err_msg, 15, 1)
```

```
END
```

On-Demand/Just-In-Time Provisioning

The screenshot displays the Azure Data Studio interface for a Data Factory pipeline. The left sidebar shows the 'Pipelines' section with 'myPipeline3' selected. The main canvas shows a pipeline diagram with three activities: 'startMyIR' (Web), 'mySprocActivity' (Stored Procedure), and 'stopMyIR' (Web). The 'stopMyIR' activity is highlighted with a red circle. A red callout bubble points to the 'stopMyIR' activity with the text: 'Add a web activity to stop your Azure-SSIS IR on completion of the sproc activity'.

The pipeline diagram shows the following flow:

```
graph LR; startMyIR[Web: startMyIR] --> mySprocActivity[Stored Procedure: mySprocActivity]; mySprocActivity --> stopMyIR[Web: stopMyIR];
```

The 'stopMyIR' activity configuration is shown in the 'Settings' tab:

- URL: `https://s1events.azure-automation.net/web`
- Method: `POST`
- Headers: No headers specified
- Body: `["message": "Stop my IR"]`
- Datasets: (Empty)
- Linkedservices: `Select...`

On-Demand/Just-In-Time Provisioning

The screenshot displays the Azure Data Factory portal interface. On the left, the 'Pipelines' section lists 'myPipeline', 'myPipeline2', and 'myPipeline3'. The 'myPipeline3' tab is active, showing a workflow diagram with three activities: 'startMyIR' (WebActivity), 'mySprocActivity' (SqlServerStoredProcedure), and 'stopMyIR' (WebActivity). A red arrow points from the 'Trigger' button in the top toolbar to the 'Trigger Now' and 'New/Edit' options in the context menu. Below the diagram, the 'Output' tab is selected, showing a table of pipeline runs. A red circle highlights the 'Trigger/schedule your pipeline' text, which is a callout for the 'Trigger Now' option.

Trigger/schedule your pipeline

Name	Type	Run Start	Duration	Status	Actions	RunID
mySprocActi...	SqlServerSto...	02/20/2018 3:16 AM	00:15:32	In Progress		5e1b84c5-d4e3-45b3-8637-8a7a151f4a32
startMyIR	WebActivity	02/20/2018 3:16 AM	00:00:03	Succeeded		2365df05-f9ec-4ab8-b81d-5c360412ee0a

Resources

Contacts

- My email: sawinark@microsoft.com
- SSIS online survey:
https://www.surveybuilder.com/s/Dg6Bq?source_type=email
- SSIS Advisors Yammer Group:
https://www.yammer.com/azureadvisors/#/threads/inGroup?type=in_group&feedId=12090139&view=all

Documentations

- <https://docs.microsoft.com/en-us/azure/data-factory/tutorial-create-azure-ssis-runtime-portal>
- <https://docs.microsoft.com/en-us/azure/data-factory/join-azure-ssis-integration-runtime-virtual-network>
- <https://docs.microsoft.com/en-us/sql/integration-services/lift-shift/ssis-azure-connect-to-catalog-database>
- <https://docs.microsoft.com/en-us/sql/integration-services/lift-shift/ssis-azure-connect-with-windows-auth>
- <https://docs.microsoft.com/en-us/sql/integration-services/ssis-quickstart-deploy-ssms>
- <https://docs.microsoft.com/en-us/sql/integration-services/ssis-quickstart-run-ssms>
- <https://docs.microsoft.com/en-us/sql/integration-services/lift-shift/ssis-azure-schedule-packages>
- <https://docs.microsoft.com/en-us/azure/data-factory/how-to-invoke-ssis-package-stored-procedure-activity>
- <https://docs.microsoft.com/en-us/azure/data-factory/monitor-integration-runtime#azure-ssis-integration-runtime>
- <https://docs.microsoft.com/en-us/azure/data-factory/manage-azure-ssis-integration-runtime>
- <https://docs.microsoft.com/en-us/azure/data-factory/how-to-schedule-azure-ssis-integration-runtime>

