

# Azure Data Practicum

This document is best viewed in Microsoft Word, "Web Layout" view with Navigation pane on

This practicum is an opportunistic capture of knowledge gained while helping customers implement some very interesting solutions.



Content is organized into **product- | activity- focused objectives** and **discipline-specific sections**.

Consider navigating directly to those objectives that align with your focus and interest.

## Table of Contents

DevOps.....	3
Objective – Deploy Infrastructure-as-Code (feat. <b>DevOps + ARM Templates</b> ) .....	3
Objective - <b>Synapse + DevOps</b>   Source Control.....	16
Objective - <b>Synapse</b>   Schema Comparison .....	23
Migration .....	26
Objective - <b>Data Migration Assistant</b>   On-Prem SQL Server >> Azure SQL.....	26
Objective - <b>Linux + AzCopy</b>   Synchronize Unstructured Data, Server-to-Cloud.....	32
Objective - Synchronize Unstructured Data, On-Prem to On-Prem (feat. <b>File Sync</b> ) .....	38
Engineering .....	44
Objective - <b>Synapse</b>   Use External Data.....	44
Objective - <b>Synapse + Data Lake</b>   Simple Data Ingestion.....	53
Objective - <b>Synapse + SQL + Data Explorer</b>   Simple Data Pipeline.....	65
Objective - <b>Synapse</b>   Simple Conditional Pipeline .....	73
Objective - <b>Synapse (pipeline) + Data Explorer</b>   Incremental Load .....	79
Objective - <b>Synapse (data flow) + Delta Lake</b>   Incremental Load .....	88

Objective - <b>Databricks + Data Lake</b>   Mount Data Lake .....	130
Objective - <b>Databricks</b>   Localize Mounted Data .....	134
Objective - <b>Databricks</b>   Source from APIs.....	137
Objective - <b>Databricks</b>   Batch Upsert Data .....	141
Objective - <b>Data Explorer</b>   Query from On-Prem .....	144
Analysis .....	148
Objective - <b>Metrics Advisor + Data Explorer</b>   Onboard and Investigate.....	148
Application .....	155
Objective - <b>SQL + Power Apps + AI Builder</b>   Capture Data.....	155
Governance.....	178
Objective - <b>Purview</b>   Discover Data .....	178
Objective - <b>Purview</b>   Classify Data .....	192
Objective - <b>Purview</b>   Understand Lineage.....	198
Monitoring .....	203
Objective - <b>Synapse</b>   Audit Data Usage .....	203
Infrastructure.....	207
Application Registration (aka Service Principal).....	207
Command Line Interface (CLI).....	214
Data Lake .....	214
Data Migration Assistant.....	220
Data Share.....	220
Data Studio .....	233
Databricks .....	234
DevOps.....	240
Key Vault.....	244
Log Analytics .....	245
Metrics Advisor .....	246
Postman .....	247
Power Apps.....	247
Purview .....	247
Resource Group .....	253
SQL.....	254
Storage Account.....	256
Synapse .....	257
Virtual Machines.....	262

# DevOps

Objective(s) in this section speak to use of Azure DevOps, Data Studio, etc. for source control and release management.

## Objective – Deploy Infrastructure-as-Code (feat. DevOps + ARM Templates)

Use Case:

- Customer XYZ wants to automate infrastructure deployment for multiple environments {e.g., DEV, TEST, PROD}
- Customer XYZ want to protect their investment in infrastructure / configuration tuning with source control

Follow the instructions in this section to **deploy resources from DevOps using ARM Templates**.

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- DevOps (with project and repository)

### Step 2: Create Template

In this sub-section, we will create and deploy a bare-minimum template from scratch, manually deploy that template, and then export the deployed template to see what Azure adds.

Navigate to [Custom deployment - Microsoft Azure](#) and click on the “Build your own template in the editor” link.

The screenshot shows the Microsoft Azure 'Edit template' interface. The URL is https://ms.portal.azure.com/#create/Microsoft.Template. The page title is 'Edit template - Microsoft Azure'. The left sidebar shows 'Parameters (0)', 'Variables (1)', and 'Resources (1)' with one item: '[concat(variables('NamePrefix'),'sa') (Microsoft.Storage/storageAccount]'. The main area displays the following JSON code:

```
$schema: "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",  
contentVersion: "1.0.0.0",  
variables: {  
    NamePrefix: "practicum"  
},  
resources: [  
    {  
        type: "Microsoft.Storage/storageAccounts",  
        apiVersion: "2015-06-15",  
        name: "[concat(variables('NamePrefix'),'sa')]",  
        location: "[resourceGroup().location]",  
        dependsOn: [],  
        properties: {  
            accountType: "Standard_LRS"  
        }  
    }  
]
```

At the bottom, there are 'Save' and 'Discard' buttons.

On the “Edit template” screen, replace the default JSON with the following prepared, “bare minimum” JSON:

```
{  
    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",  
    "contentVersion": "1.0.0.0",  
    "variables": {},  
    "resources": []  
}
```

```

"contentVersion": "1.0.0.0",
"resources": [
    {
        "type": "Microsoft.Storage/storageAccounts",
        "apiVersion": "[providers('Microsoft.Storage','storageAccounts').apiVersions[0]]",
        "name": "[concat(resourceGroup().name,'sa')]",
        "location": "[resourceGroup().location]",
        "sku": {
            "name": "Standard_LRS"
        },
        "kind": "StorageV2"
    }
]
}

```

Visit the following documentation links to understand core template content:

- [Tutorial - Create & deploy template - Azure Resource Manager | Microsoft Docs](#)
- [Tutorial - Add resource to template - Azure Resource Manager | Microsoft Docs](#)

Special logic callouts...

NamePrefix	This variable will be used as prefix for all resource names {e.g., “practicumsa” for the storage account}
[concat...	Concatenation of the variable and acronym for the resource {i.e., “sa” for storage account}
Location	Use of “[resourceGroup().location]” ensures common location specification
AccountType	“Standard_LRS” is the cheapest option (and fine for demonstration) Every resource will have specific properties and codes best derived from online documentation

Review the pasted JSON and then click the **Save** button.

## Manual Deploy

The screenshot shows the 'Custom deployment' page in the Microsoft Azure (Preview) portal. At the top, there's a header bar with the Microsoft logo, a search bar, and user information. Below the header, the main content area has a dark background. The title 'Custom deployment' is displayed, along with a sub-instruction 'Deploy from a custom template'. There are three tabs at the top of the form: 'Select a template', 'Basics' (which is selected and underlined in blue), and 'Review + create'. The 'Template' section contains a card for a 'Customized template' with one resource. It includes 'Edit template' and 'Visualize' buttons. The 'Project details' section asks to select a subscription and resource group. The 'Subscription' dropdown is set to 'rchapler' and the 'Resource group' dropdown is set to 'practicumrg'. The 'Instance details' section asks to select a region, with '(US) West US 2' chosen. At the bottom, there are three buttons: 'Review + create' (highlighted in blue), '< Previous', and 'Next : Review + create >'. The entire interface is designed for creating a new resource using a custom template.

On the “Custom deployment” page, enter values for the following items:

Resource Group	Select existing or “Create new”
Region	Match resource group selection

Click the “Review + create” button, validate settings, and then click the **Create** button.

When deployment is complete, click the “Go to resource” button to review the created Storage Account.

## Export Template

Next, we will create a template by exporting the configuration of an existing resource.

Navigate to the newly created Storage Account, and then click “Export template” in the **Automation** group of the navigation.

```

1  {
2    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
3    "contentVersion": "1.0.0.0",
4    "parameters": {
5      "storageAccounts_practicumsa_name": {
6        "defaultValue": "practicumsa",
7        "type": "String"
8      }
9    },
10   "variables": {},
11   "resources": [
12     {
13       "type": "Microsoft.Storage/storageAccounts",
14       "apiVersion": "2021-06-01",
15       "name": "[parameters('storageAccounts_practicumsa_name')]",
16       "location": "westus2",
17       "properties": {
18         "allowBlobPublicAccess": false,
19         "allowFilePublicAccess": false,
20         "allowTablePublicAccess": false,
21         "allowQueuePublicAccess": false
22       }
23     }
24   ]
25 }

```

Worth noting...

- There is a lot more JSON than we included in the bare bones JSON!
- Even though so much was included, there are still items excluded as described in the red informational at the top {i.e., "...inventoryPolicies cannot be exported yet"}

Click the **Download** button and “ExportedTemplate-practicumrg.zip” will be written to the Downloads folder on your device.

Open the downloaded ZIP file and copy the “**template.json**” file to your Desktop.

Navigate to [Custom deployment - Microsoft Azure](#) and click on the “**Build your own template in the editor**” link.

Click “**Load file**”, navigate to the “**template.json**” file on your Desktop, and then click the **Open** button to edit.

## Upload Template

In this sub-section, we will upload our template to our DevOps repo.

Navigate to your instance of DevOps and then click on **Repos** in the navigation.

Click the **+** icon just to the right of your project name and select “**New repository**” from the resulting dropdown.

In the resulting “Create a repository” popout, enter a “**Repository name**” {e.g., “infrastructure”} and then click the **Create** button.

Initialize the new repository with a ReadMe file; you can delete this file once additional files have been uploaded.

Click the vertical ellipses button in the upper-right of the screen, and then “**Upload file(s)**” in the resulting dropdown.

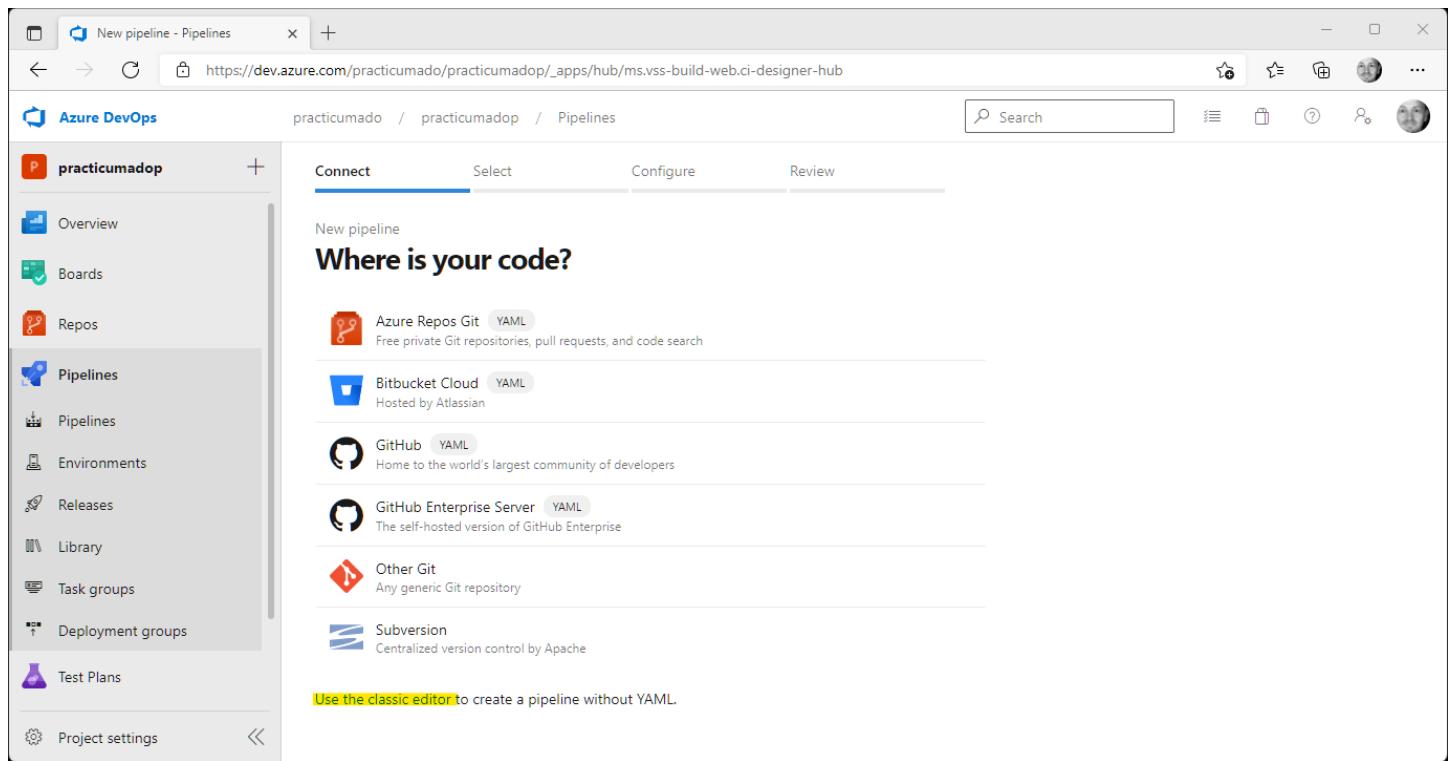
Drag-and-drop the **StorageAccount.json** template and click the **Commit** button.

## Step 3: Create Pipeline

In this sub-section, we will create, save, and run a DevOps Pipeline that uses our ARM template.

Navigate to your instance of DevOps and then click on **Pipelines** in the navigation.

Click the “**New Pipeline**” button.



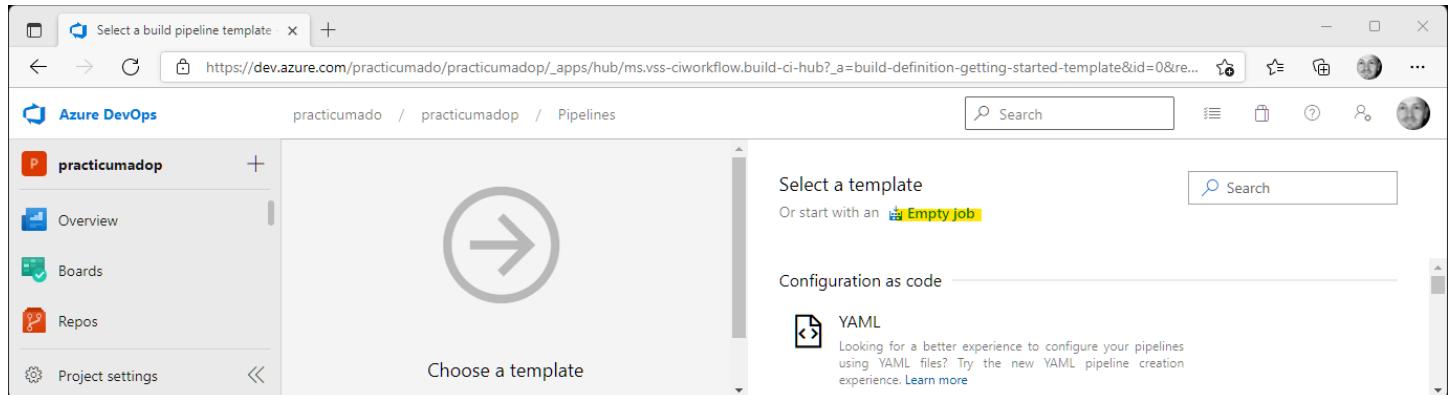
The screenshot shows the 'New pipeline' page in the Azure DevOps interface. The left sidebar shows the project 'practicumadop' and various navigation options like Overview, Boards, Repos, Pipelines, etc. The main area has a tab bar with 'Connect' selected, followed by 'Select', 'Configure', and 'Review'. Below the tabs, the heading 'Where is your code?' is displayed. A list of source providers is shown, each with a icon, name, and 'YAML' link: Azure Repos Git, Bitbucket Cloud, GitHub, GitHub Enterprise Server, Other Git, and Subversion. At the bottom of the list, there is a note: 'Use the classic editor to create a pipeline without YAML.'

On the **Connect** tab {i.e., “**New Pipeline ... Where is your code?**”} page, click the “**Use classic editor**” link.

On the “**Select a source**” popout, select “**Azure Repos Git**” and then enter values for the following items:

Team Project	Select your DevOps project
Repository	Select your DevOps repository {e.g., <b>infrastructure</b> }
Default Branch...	Confirm default selection, <b>main</b>

Click the **Continue** button.



The screenshot shows the 'Select a build pipeline template' page. The left sidebar is the same as the previous screenshot. The main area has a large circular arrow icon and the text 'Choose a template'. To the right, there is a section titled 'Select a template' with the sub-instruction 'Or start with an **Empty job**'. Below this, there is a 'Configuration as code' section with a 'YAML' link and a note: 'Looking for a better experience to configure your pipelines using YAML files? Try the new YAML pipeline creation experience. [Learn more](#)'.

Click “**Empty job**” on the “**Select a template**” popout.

The screenshot shows the Azure DevOps Pipelines interface. On the left, there's a sidebar with navigation links: Overview, Boards, Repos, Pipelines (which is selected), Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Project settings. The main area displays a pipeline named "practicumadop-Cl". The pipeline consists of a single step: "Get sources" (repository: "practicumadop\_infrastructure", branch: "main"). Below it is an "Agent job 1" step, which is currently set to "Run on agent". On the right side, there are configuration options: "Name" (set to "practicumadop-Cl"), "Agent pool" (set to "Azure Pipelines"), "Agent Specification" (set to "windows-2019"), and a "Parameters" section which is currently empty.

Confirm default values for **Name**, **Agent Pool** and **Agent Specification**.

This screenshot shows the same Azure DevOps Pipelines interface as the previous one, but with a focus on adding a new task. In the center, the pipeline structure is identical. On the right, under the "Add tasks" section, a search bar contains the text "arm". A card for "ARM template deployment" is displayed, showing its description: "Deploy an Azure Resource Manager (ARM) template to all the deployment scopes". There are "Add" and "Learn more" buttons at the bottom of the card. Below this, another card for "ARM Outputs" is partially visible.

Click the “+” button to the right of “**Agent Job 1**”.

Search for and then select “**ARM template deployment**”.

Click the **Add** button.

The screenshot shows the Azure DevOps Pipelines interface for a project named "practicumadop". A build pipeline named "practicumadop-CI" is selected. The pipeline structure is as follows:

- Get sources**: Repository "practicumadop\_infrastructure", branch "main".
- Agent job 1**: Run on agent.
- ARM Template deployment: Resource Group scope**: Selected.

The "ARM Template deployment: Resource Group scope" step is expanded, showing the following configuration:

- Task version**: 3.\*
- Display name**: ARM Template deployment: Resource Group scope
- Azure Details**:
  - Deployment scope**: Resource Group
  - Azure Resource Manager connection**: rchapler(91e9fddc-ef15-416c-9be1-085f8b1b46ed)
  - Subscription**: rchapler (91e9fddc-ef15-416c-9be1-085f8b1b46ed)
  - Action**: Create or update resource group
  - Resource group**: practicumrg
  - Location**: West US 2
- Template**:
  - Template location**: Linked artifact
  - Template**: StorageAccount.json
  - Template parameters**: (empty)
  - Override template parameters**: (empty)
  - Deployment mode**: Incremental

On the “**ARM template deployment**” popout, enter values for the following items:

<b>Deployment Scope</b>	Confirm default, “ <b>Resource Group</b> ”
<b>Azure Resource Manager...</b>	Select your subscription
<b>Subscription</b>	Select your subscription
<b>Action</b>	Select “ <b>Create or update resource group</b> ”
<b>Resource Group</b>	Select your resource group
<b>Location</b>	Match resource group selection

Template	Browse to and select the “ <b>StorageAccount.json</b> ” template created in the prior section
Deployment Mode	Confirm selection, <b>Incremental</b>

Click on the **Triggers** tab.

The screenshot shows the Azure DevOps interface for a pipeline named 'practicumadop-CI'. The left sidebar is open, showing various project management and development tools like Boards, Repos, Pipelines, Environments, and Test Plans. The main area is titled 'Continuous integration' and contains a single trigger named 'practicumdor\_infrastructure' which is 'Enabled'. Below it, there are sections for 'Scheduled' (no builds scheduled) and 'Build completion' (build when another build completes). On the right, under the 'practicumdor\_infrastructure' trigger, there are configuration options. Under 'Branch filters', the 'Type' is set to 'Include' and the 'Branch specification' is 'main'. Under 'Path filters', the 'Type' is set to 'Include' and the 'Path specification' is 'ARMTemplates'. There are also 'Add' buttons for both sections.

Check the “Enable continuous integration” checkbox; new input controls will surface.

Branch Filter	Confirm defaults, Type: <b>Include</b> and Branch Specification: <b>Main</b>
Path Filter	Add Type: <b>Include</b> and Path Specification: “ <b>ARMTemplates</b> ”

Click on the “Save & queue” dropdown button, and then click “Save & queue” in the resulting dropdown.

The screenshot shows the Azure DevOps interface for a project named 'practicumadop'. On the left, the navigation bar includes 'Overview', 'Boards', 'Repos', 'Pipelines' (which is selected), 'Environments', 'Releases', 'Library', 'Task groups', 'Deployment groups', 'Test Plans', and 'Artifacts'. The main area displays a pipeline named 'practicumadop\_CI' under the 'Continuous integration' section. It lists two triggers: 'Scheduled' (No builds scheduled) and 'Build completion' (Build when another build completes). A 'Run pipeline' dialog is open on the right, prompting for parameters. The 'Agent pool' is set to 'Azure Pipelines' and 'Agent Specification' is 'windows-2019'. The 'Branch/tag' dropdown shows 'main'. Under 'Advanced options', there is a 'Variables' section indicating '1 variable defined' and a 'Demands' section stating 'This pipeline has no defined demands'. At the bottom of the dialog are 'Cancel' and 'Save and run' buttons.

Confirm values on the resulting “Run pipeline” popout, and then click the “Save and run” button.

The screenshot shows the results of a pipeline run. The URL is https://dev.azure.com/practicumado/practicumadop/\_build/results?buildId=2&view=results. The pipeline name is 'practicumadop\_CI' and the run number is '2'. The summary indicates the run was 'Manually run by Rich Chapter'. The repository information shows it's from 'practicumadop\_CI' at commit 'af45098c'. The time started was 'Today at 6:06 AM' and it took '54s'. There are '0 work items' and '1 consumed'. The 'Jobs' section shows a single job named 'Agent job 1' with a status of 'Success' and a duration of '36s'. The 'Scans' tab is also visible.

## Confirm Success

On successful completion, you can expect to receive an email reporting “**Build succeeded**”.

Navigate to your Resource Group and confirm creation of the Storage Account described in your template.

# Objective Complete... congratulations!

## Additional Templates

This section includes simple ARM Template examples for various Azure resources; font is intentionally small.

*Note: These templates use the latest API version rather than a specific dated version; as such, schema might require update over time.*

### File Sync (aka Storage Sync Service)

```
{  
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",  
  "contentVersion": "1.0.0.0",  
  "resources": [  
    {  
      "type": "microsoft.storagesync/storageSyncServices",  
      "apiVersion": "[providers('Microsoft.StorageSync','storageSyncServices').apiVersions[0]]",  
      "name": "[concat(resourceGroup().name,'sss')]",  
      "location": "[resourceGroup().location]",  
      "properties": {  
        "incomingTrafficPolicy": "AllowAllTraffic"  
      }  
    }  
  ]  
}
```

### Storage Account

```
{  
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",  
  "contentVersion": "1.0.0.0",  
  "resources": [  
    {  
      "type": "Microsoft.Storage/storageAccounts",  
      "apiVersion": "[providers('Microsoft.Storage','storageAccounts').apiVersions[0]]",  
      "name": "[concat(resourceGroup().name,'sa')]",  
      "location": "[resourceGroup().location]",  
      "sku": {  
        "name": "Standard_LRS"  
      },  
      "kind": "StorageV2"  
    }  
  ]  
}
```

### Virtual Machine

```
{  
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",  
  "contentVersion": "1.0.0.0",  
  "resources": [  
    {  
      "type": "Microsoft.Network/publicIPAddresses",  
      "apiVersion": "[providers('Microsoft.Network','publicIPAddresses').apiVersions[0]]",  
      "name": "[concat(resourceGroup().name,'-pia')]",  
      "location": "[resourceGroup().location]",  
      "sku": {  
        "name": "Basic"  
      },  
      "properties": {  
        "publicIPAllocationMethod": "Dynamic",  
        "dnsSettings": {  
          "domainNameLabel": "[toLower(format('{0}-{1}', resourceGroup().name, uniqueString(resourceGroup().id, resourceGroup().name)))]"  
        }  
      }  
    },  
    {  
      "type": "Microsoft.Network/networkSecurityGroups",  
      "apiVersion": "[providers('Microsoft.Network','networkSecurityGroups').apiVersions[0]]",  
      "name": "[concat(resourceGroup().name,'-nsg')]",  
      "location": "[resourceGroup().location]",  
      "properties": {  
        "securityRules": [  
          {  
            "name": "default-allow-3389",  
            "properties": {  
              "protocol": "TCP",  
              "port": 3389,  
              "direction": "Inbound",  
              "priority": 100,  
              "access": "Allow",  
              "virtualNetworkRules": [  
                {  
                  "virtualNetwork": "[resourceGroup().name]",  
                  "subnets": [  
                    {  
                      "id": "[resourceGroup().id]"/subnets/{0}"  
                    }  
                  ]  
                }  
              ]  
            }  
          }  
        ]  
      }  
    }  
  ]  
}
```

```

        "priority": 1000,
        "access": "Allow",
        "direction": "Inbound",
        "destinationPortRange": "3389",
        "protocol": "Tcp",
        "sourcePortRange": "*",
        "sourceAddressPrefix": "*",
        "destinationAddressPrefix": "*"
    }
}
]
}
{
  "type": "Microsoft.Network/virtualNetworks",
  "apiVersion": "[providers('Microsoft.Network','virtualNetworks').apiVersions[0]]",
  "name": "[concat(resourceGroup().name,'-vn')]",
  "location": "[resourceGroup().location]",
  "properties": {
    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/16"
      ]
    },
    "subnets": [
      {
        "name": "Subnet",
        "properties": {
          "addressPrefix": "10.0.0.0/24",
          "networkSecurityGroup": {
            "id": "[resourceId('Microsoft.Network/networkSecurityGroups', concat(resourceGroup().name,'-nsg'))]"
          }
        }
      }
    ]
  },
  "dependsOn": [
    "[resourceId('Microsoft.Network/networkSecurityGroups', concat(resourceGroup().name,'-nsg'))]"
  ]
},
{
  "type": "Microsoft.Network/networkInterfaces",
  "apiVersion": "2021-02-01",
  "name": "[concat(resourceGroup().name,'-ni')]",
  "location": "[resourceGroup().location]",
  "properties": {
    "ipConfigurations": [
      {
        "name": "ipconfig1",
        "properties": {
          "privateIPAllocationMethod": "Dynamic",
          "publicIPAddress": {
            "id": "[resourceId('Microsoft.Network/publicIPAddresses', concat(resourceGroup().name,'-pia'))]"
          },
          "subnet": {
            "id": "[resourceId('Microsoft.Network/virtualNetworks/subnets', concat(resourceGroup().name,'-vn'), 'Subnet')]"
          }
        }
      }
    ]
  },
  "dependsOn": [
    "[resourceId('Microsoft.Network/publicIPAddresses', concat(resourceGroup().name,'-pia'))]",
    "[resourceId('Microsoft.Network/virtualNetworks', concat(resourceGroup().name,'-vn'))]"
  ]
},
{
  "type": "Microsoft.Compute/virtualMachines",
  "apiVersion": "[providers('Microsoft.Compute','virtualMachines').apiVersions[0]]",
  "name": "[resourceGroup().name]",
  "location": "[resourceGroup().location]",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2_v3"
    },
    "osProfile": {
      "computerName": "[resourceGroup().name]",
      "adminUsername": "Admin123",
      "adminPassword": "Password123!"
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2022-Datacenter",
        "version": "latest"
      },
      "osDisk": {
        "name": "[concat(resourceGroup().name,'-d-os')]",
        "createOption": "FromImage",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS"
        }
      }
    },
    "dataDisks": [

```

```
{
    "name": "[concat(resourceGroup().name,'-d-data')]",
    "diskSizeGB": 1023,
    "lun": 0,
    "createOption": "Empty"
}
],
},
"networkProfile": {
    "networkInterfaces": [
        {
            "id": "[resourceId('Microsoft.Network/networkInterfaces', concat(resourceGroup().name,'-ni'))]"
        }
    ]
},
"dependsOn": [
    "[resourceId('Microsoft.Network/networkInterfaces', concat(resourceGroup().name,'-ni'))]"
]
},
"outputs": {
    "hostname": {
        "type": "string",
        "value": "[reference(resourceId('Microsoft.Network/publicIPAddresses', concat(resourceGroup().name,'-pia'))).dnsSettings.fqdn]"
    }
}
}
```

## Objective - Synapse + DevOps | Source Control

*Note: These instructions also apply (with minor differences) to Azure Data Factory.*

Follow the instructions in this section to **connect Synapse to DevOps**, **create branches**, and **pull requests**.

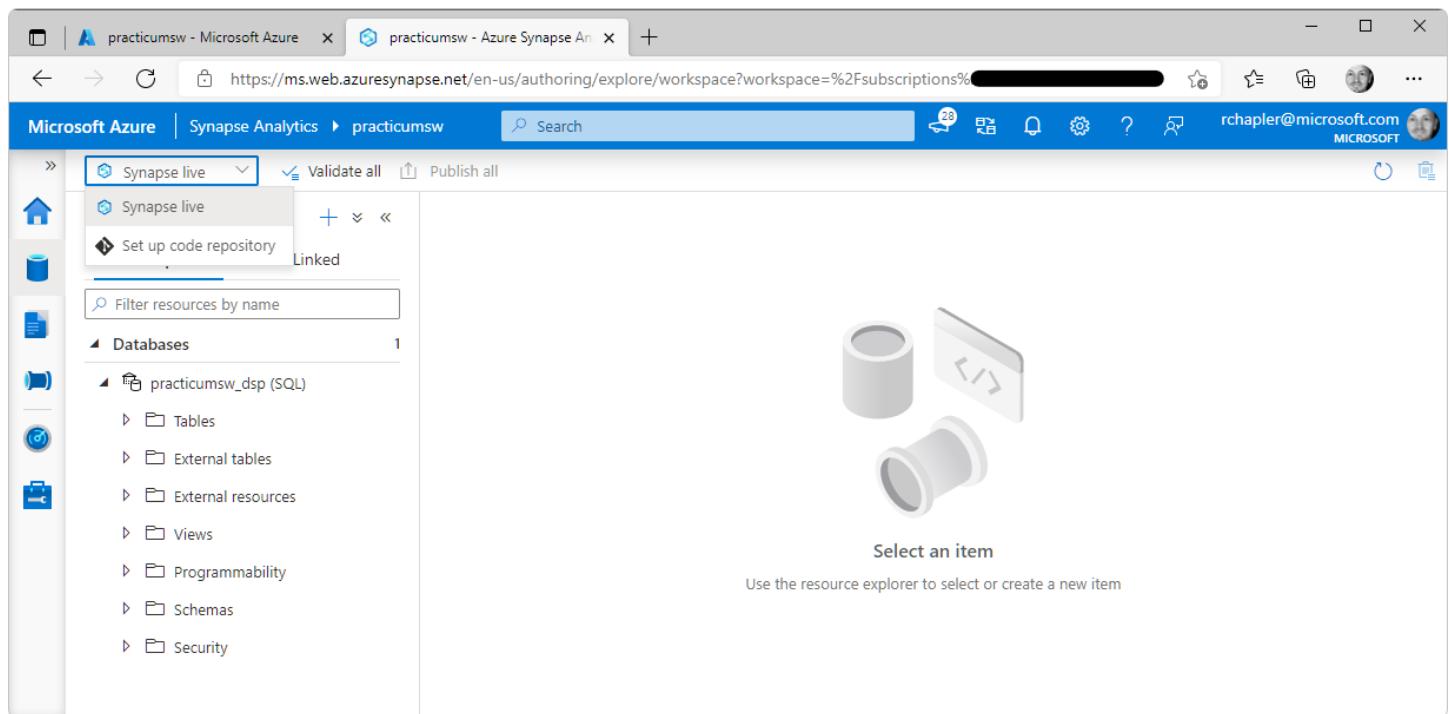
### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- DevOps
- Synapse (with Dedicated SQL Pool)

### Step 2: Setup Code Repository

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.



Click “**Synapse live**” and then select “**Set up code repository**” from the resulting dropdown.

The screenshot shows the Microsoft Azure portal interface. In the top navigation bar, there are two tabs: "practicumsw - Microsoft Azure" and "practicumsw - Azure Synapse Analytics". The URL in the address bar is <https://ms.web.azuresynthesize.net/en-us/authoring/explore/workspace?workspace=%2Fsubscriptions%2F...>. The main content area is titled "Configure a repository". It contains instructions: "Specify the settings that you want to use when connecting to your repository." Below this, there are two dropdown menus: "Repository type" set to "Azure DevOps Git" and "Azure Active Directory" set to "Microsoft". On the left side of the screen, there is a sidebar titled "Data" which lists "Workspace" and "Linked". Under "Workspace", there is a section for "Databases" which includes "practicumsw\_dsp (SQL)" with sub-options like "Tables", "External tables", "External resources", "Views", "Programmability", "Schemas", and "Security". At the bottom right of the configuration dialog are "Continue" and "Cancel" buttons.

On the “Configure a repository” popout, enter values for the following items:

---

**Repository Type** Select “**Azure DevOps Git**” from the dropdown

**Azure Active Directory** Select the value appropriate for your organization

---

Click the **Continue** button.

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. On the left, there's a sidebar with icons for Home, Data, Workspace, Databases, Tables, External tables, External resources, Views, Programmability, Schemas, and Security. The 'Data' section is selected. At the top, there are buttons for 'Set up code repository', 'Validate all', 'Publish all', and a search bar. The main area is titled 'Configure a repository' and shows settings for connecting to a repository. It includes fields for 'Azure DevOps organization name' (set to 'practicumado'), 'Project name' (set to 'practicumadop'), 'Repository name' (set to 'practicumadop'), 'Collaboration branch' (set to 'collaboration\_synapse'), 'Publish branch' (set to 'workspace\_publish'), and 'Root folder' (set to '/'). There's also an unchecked checkbox for 'Import existing resource'. At the bottom are 'Apply', 'Back', and 'Cancel' buttons.

On the second “Configure a repository” popout, enter values for the following items:

Azure DevOps Organization...	Select your DevOps account
Project Name	Select your DevOps project
Repository Name	Select the repo created with your DevOps project
Collaboration Branch	Dropdown, click “+ Create new”, enter an appropriate name in the resulting popup, and click <b>Create</b>
Publish Branch	Confirm default, <b>workspace_publish</b>
Root Folder	Confirm default, <b>/</b>
Import Existing Resources...	Unchecked

Click the **Apply** button.

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. On the left, there's a sidebar with icons for Home, Data, Databricks, Data Lake, Pipelines, and Data Flow. The main area shows a workspace named 'collaboration\_synapse' containing one database: 'practicumsw\_dsp (SQL)'. On the right, a 'Set working branch' dialog is open. It has two radio buttons: 'Create new' (unchecked) and 'Use existing' (checked). Below the radio buttons is a dropdown menu showing 'collaboration\_synapse'. At the bottom of the dialog is a blue 'Save' button.

Confirm values and then click the **Save** button.

When processing is complete, you will note that we are now working in the new branch.

### Step 3: Confirm Success

In this section, we will confirm success by simulating a deployment workflow {i.e., modification, and then pull request}.

Modify one or more of the following:

- SQL Script
- Spark Notebook
- Data Flow
- Pipeline

#### Pull Request

Next, we will promote changes from the collaboration branch to the master branch via Pull Request.

Click on the “{collaboration} branch” dropdown control and select “**Create pull request...**” from the resulting dropdown.

A third tab in your browser (pointing to <https://dev.azure.com...>) will open.

The screenshot shows the 'New pull request' interface in Azure DevOps. On the left, a sidebar lists project navigation options like Overview, Boards, Repos, Pull requests, Pipelines, Test Plans, Artifacts, and Compliance. The 'Pull requests' option is currently selected. The main content area displays a 'New pull request' form. At the top, it shows the source branch as 'collaboration\_synapse' and the target branch as 'master'. Below this, tabs for 'Overview', 'Files' (6), and 'Commits' (4) are visible. The 'Overview' tab is active. The main form fields include:

- Title:** A text input field containing 'My Pull Request'.
- Description:** A large text area with placeholder text 'Describe the code that is being reviewed'.
- Markdown supported:** A note indicating that Markdown is supported, with a drag & drop or file selection option.
- Link work items:** A note with a link icon.
- Reviewers:** A search bar labeled 'Search users and groups to add as reviewers'.
- Add required reviewers:** A button to add required reviewers.
- Work items to link:** A search bar labeled 'Search work items by ID or title'.
- Tags:** An empty input field.

A blue 'Create' button is located at the bottom right of the form.

On the second “New pull request” page, enter values for the following items:

Branches	{collaboration branch} into {master branch} ... you may have to create a master branch if one does not exist
Title	Enter a title that is meaningful for you (and aligned with your naming standards)

Enter values for items like **Reviewers** and “**Work items...**” as appropriate.

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the **Create** button.

The screenshot shows the Azure DevOps interface for a pull request. On the left, there's a sidebar with project navigation (Overview, Boards, Repos, etc.) and a 'Pull requests' section highlighted. The main area is titled 'My Pull Request' and shows an active pull request from 'Rich Chapler' to 'collaboration\_synapse' into 'master'. The pull request has a green checkmark indicating 'No merge conflicts' (last checked 'Just now'). Below this are tabs for 'Overview', 'Files', 'Updates', and 'Commits'. To the right, there are sections for 'Reviewers' (with a note 'Required' and 'No required reviewers'), 'Tags' (with a note 'No tags'), and 'Work items' (with a note 'No work items'). A comment box is open, showing a message from 'Rich Chapler' created 'Just now': 'Rich Chapler created the pull request'. A 'Show everything (1)' button is also visible.

At this point (and if specified), designated reviewers would be notified by email, and they will need to review and approve for the Pull Request to move forward.

Additional configuration can provide for validation of other gating criteria {e.g., inclusion of text patterns that might be secret like an account key}.

We have not included gating factors in this example.

Click the **Complete** button.

The screenshot shows the same Azure DevOps interface as before, but with a modal dialog box titled 'Complete pull request' overlaid. The dialog contains fields for 'Merge type' (set to 'Merge (no fast forward)'), a diagram illustrating the merge process, and 'Post-completion options' with three checked checkboxes: 'Complete associated work items after merging', 'Delete collaboration\_synapse after merging', and 'Customize merge commit message'. At the bottom of the dialog are 'Cancel' and 'Complete merge' buttons.

Confirm selections in the “Complete pull request” popout.

Click the “Complete merge” button.

The screenshot shows the Azure DevOps interface for a pull request titled "My Pull Request". The status bar at the top indicates "Completed" by Rich Chapler, merging "collaboration\_synapse" into "master". The "Overview" tab is selected. A message box shows "Rich Chapler completed this pull request Just now". Below it, a "Merged PR 2: My Pull Request" summary is shown, mentioning commit eda684b1 by Rich Chapler at "Just now". A "Show details" link is available. A green checkmark indicates "No merge conflicts Last checked Just now". The right side of the screen displays sections for "Reviewers" (Required: No required reviewers), "Optional" (No optional reviewers), "Tags" (No tags), and "Work items" (No work items). A comment input field says "Add a comment...". A recent activity log shows "Rich Chapler completed the pull request Just now".

*Note: Post-completion options such as “Cherry-pick” and Revert provide for operational control even after a code merge.*

*Objective Complete... congratulations!*

## Objective - Synapse | Schema Comparison

Follow the instructions in this section to **compare two database schemas** {e.g., old, and new}.

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Data Studio
- Synapse

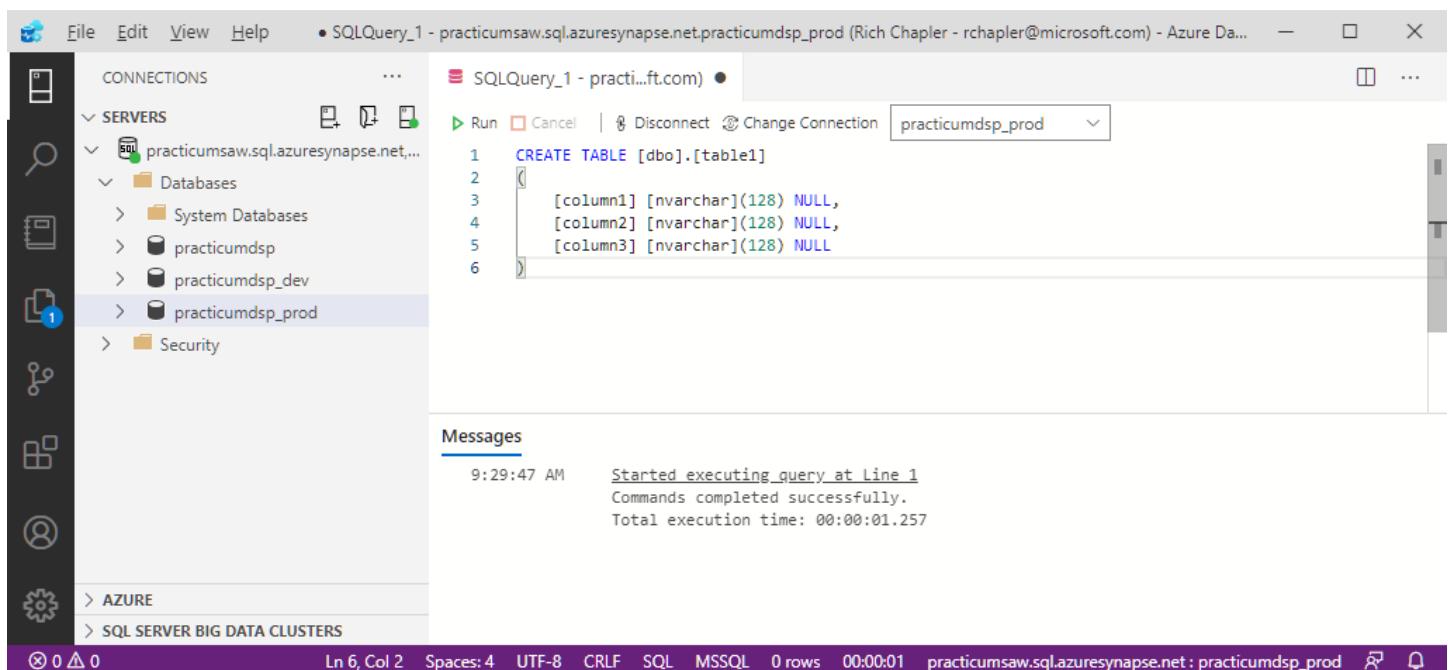
Although this example focuses on Synapse, the same solution applies to SQL.

### Step 2: Stage Resources

Navigate to your Synapse Workspace and follow the directions at [Instantiate Resources | Synapse | SQL Pool](#) to create two SQL Pools named **practicumdsp\_dev** and **practicumdsp\_prod**.

We will delete both SQL Pools at the end of this objective.

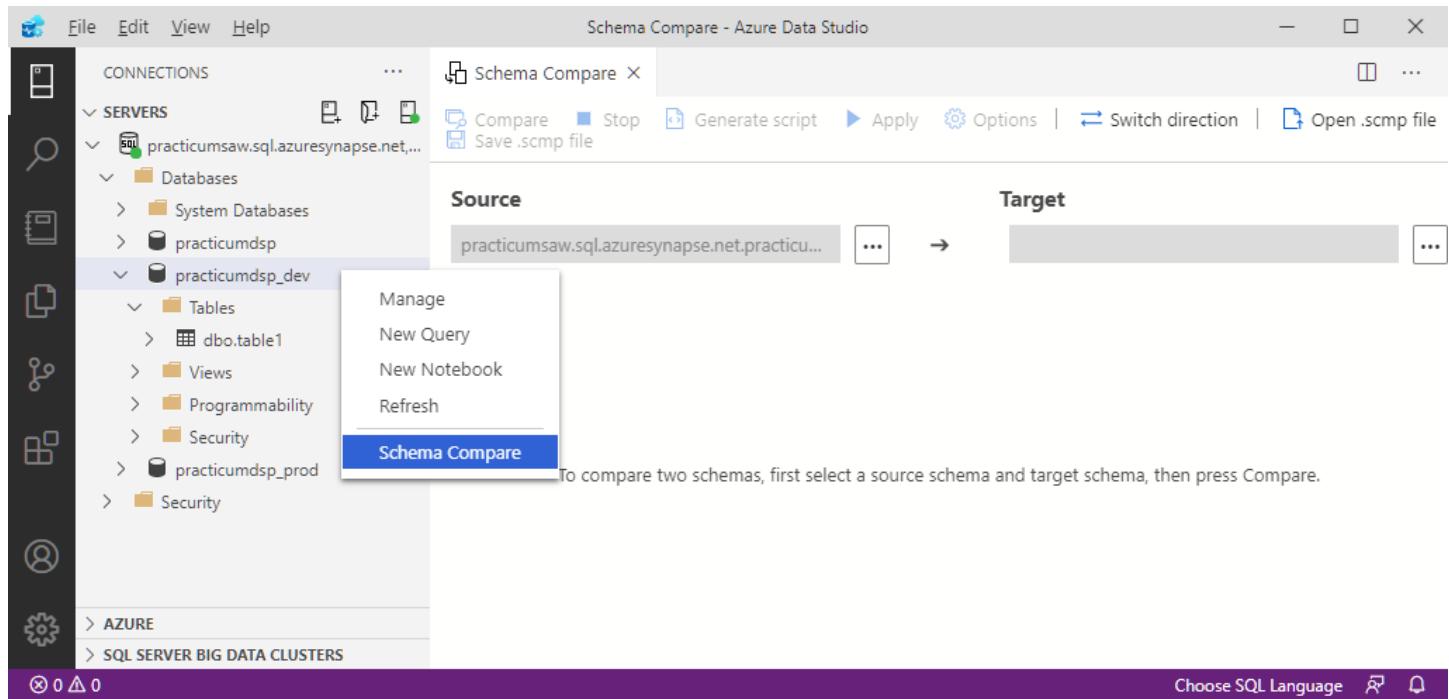
Open Azure Data Studio.



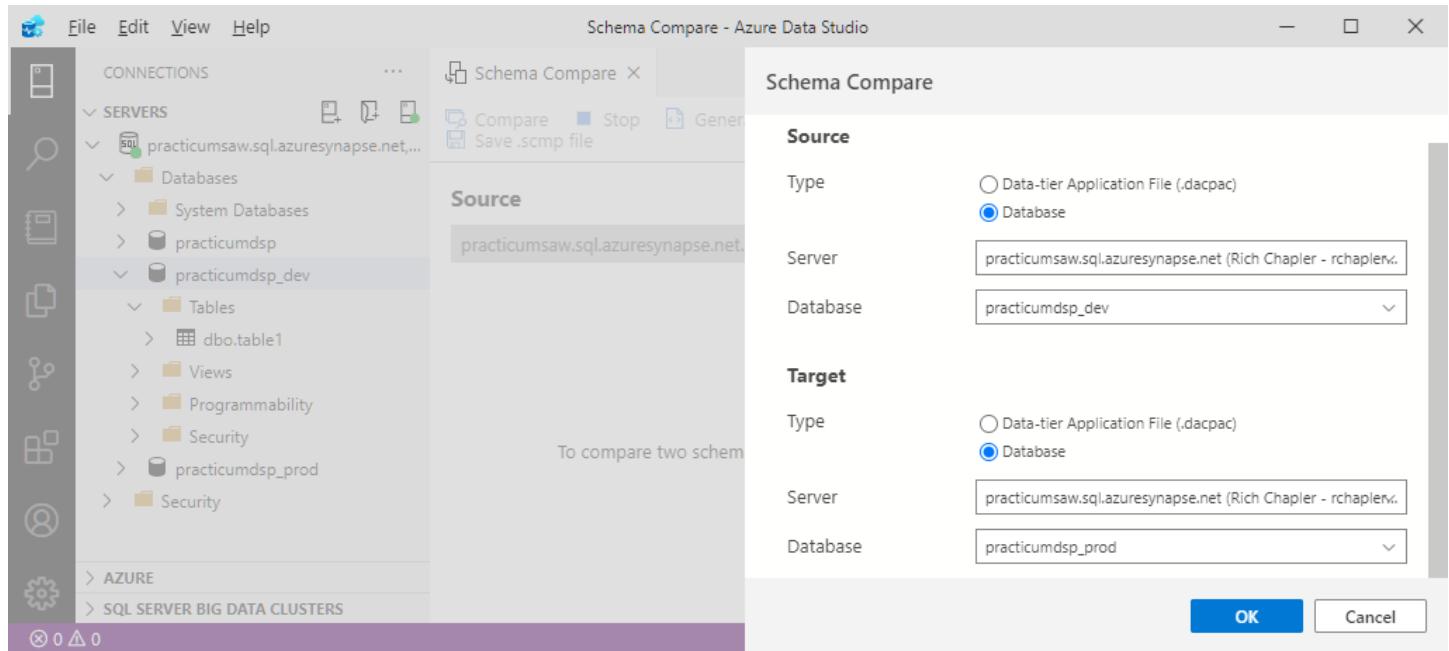
Execute the following T-SQL on **practicumdsp\_dev**:

```
CREATE TABLE [dbo].[table1]
(
    [column1] [nvarchar](128) NULL,
    [column2] [nvarchar](128) NULL,
    [column3] [nvarchar](128) NULL
)
```

### Step 3: Schema Compare



Right-click on the SQL Pool and click “Schema Compare” in the resulting dropdown.

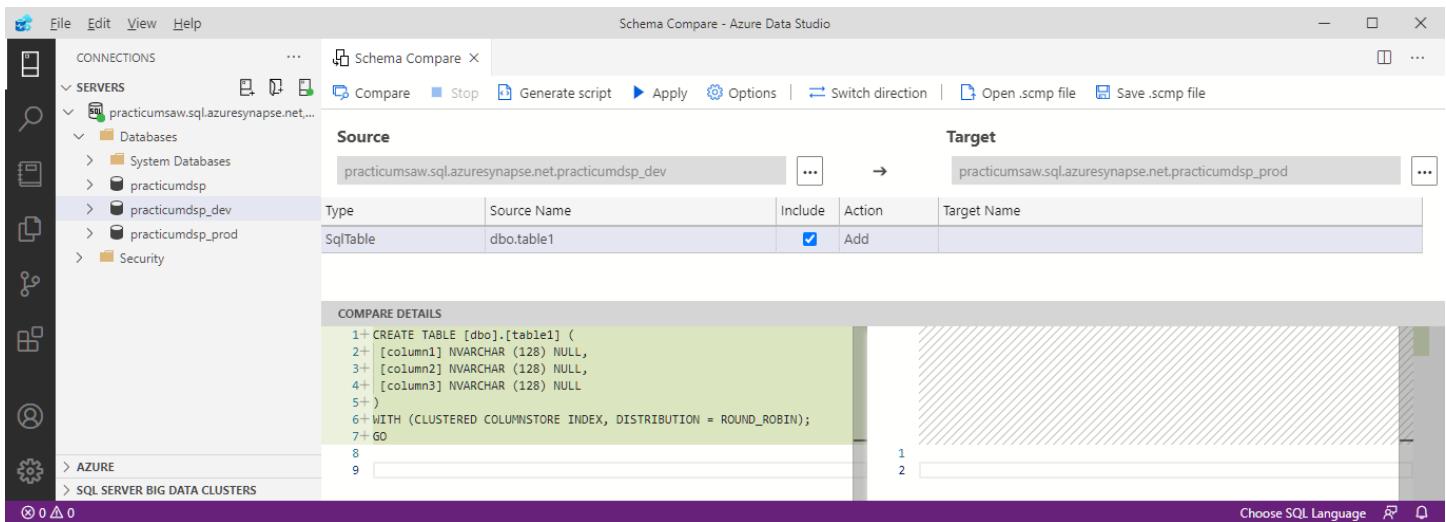


Click the ellipses button.

On the “Schema Compare” popout, enter values for the following items:

Source   Type	Select the <b>Database</b> radio button
Source   Server and Database	Confirm selection of your Synapse server and the <b>practicumdsp_dev</b> database
Target   Type	Select the <b>Database</b> radio button
Target   Server and Database	Confirm selection of your Synapse server and the <b>practicumdsp_prod</b> database

Click the **OK** button.

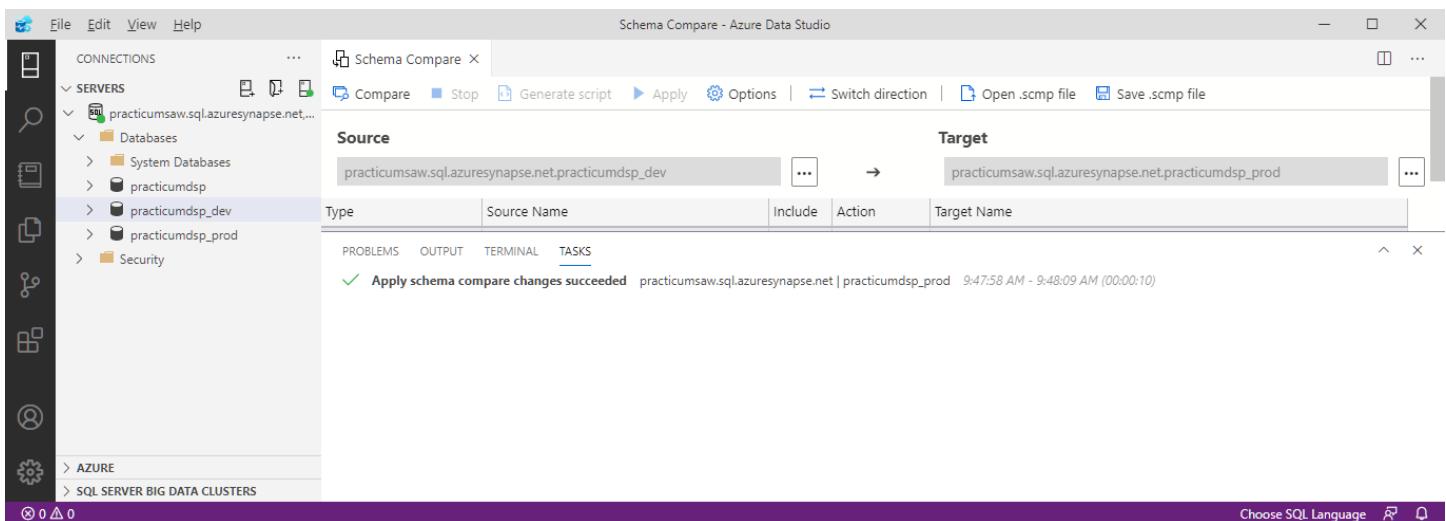


On the “Schema Compare” tab, click the **Compare** button.

We only created a table on the development database, so Schema Compare should surface that table as something missing from the production database.

Click on the row to see T-SQL for the identified item.

Click the **Apply** button to update the target.



*Objective Complete... congratulations!*

# Migration

Objective(s) in this section speak to one-time or recurring migration of data from on-prem databases.

## Objective - Data Migration Assistant | On-Prem SQL Server >> Azure SQL

### Use Case:

Customer XYZ wants to migrate some of many tables (schema and data) from an on-prem, SQL Server 2008 R2 database to Azure

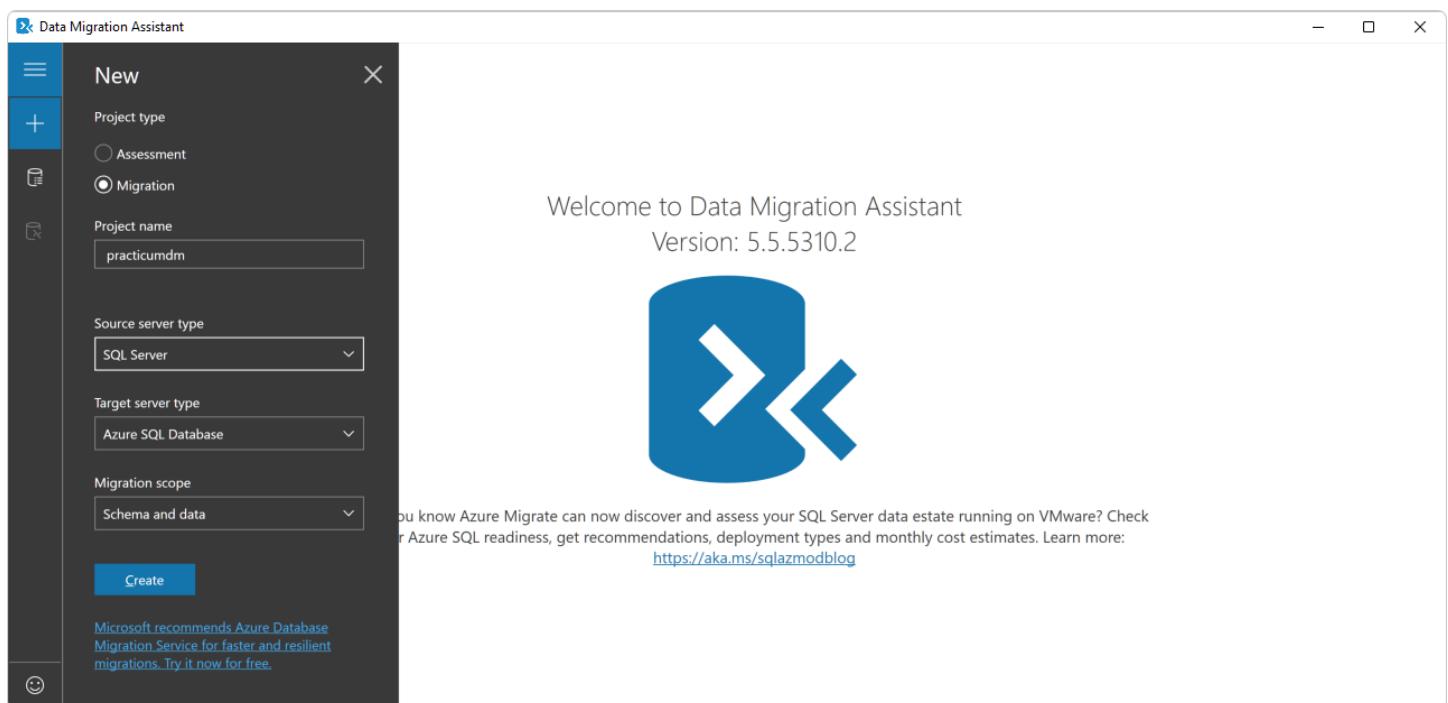
### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Data Migration Assistant
- SQL
- Virtual Machine (with “SQL Server 2008 R2...” image and **AdventureWorks** sample database)

### Step 2: Create Migration Project

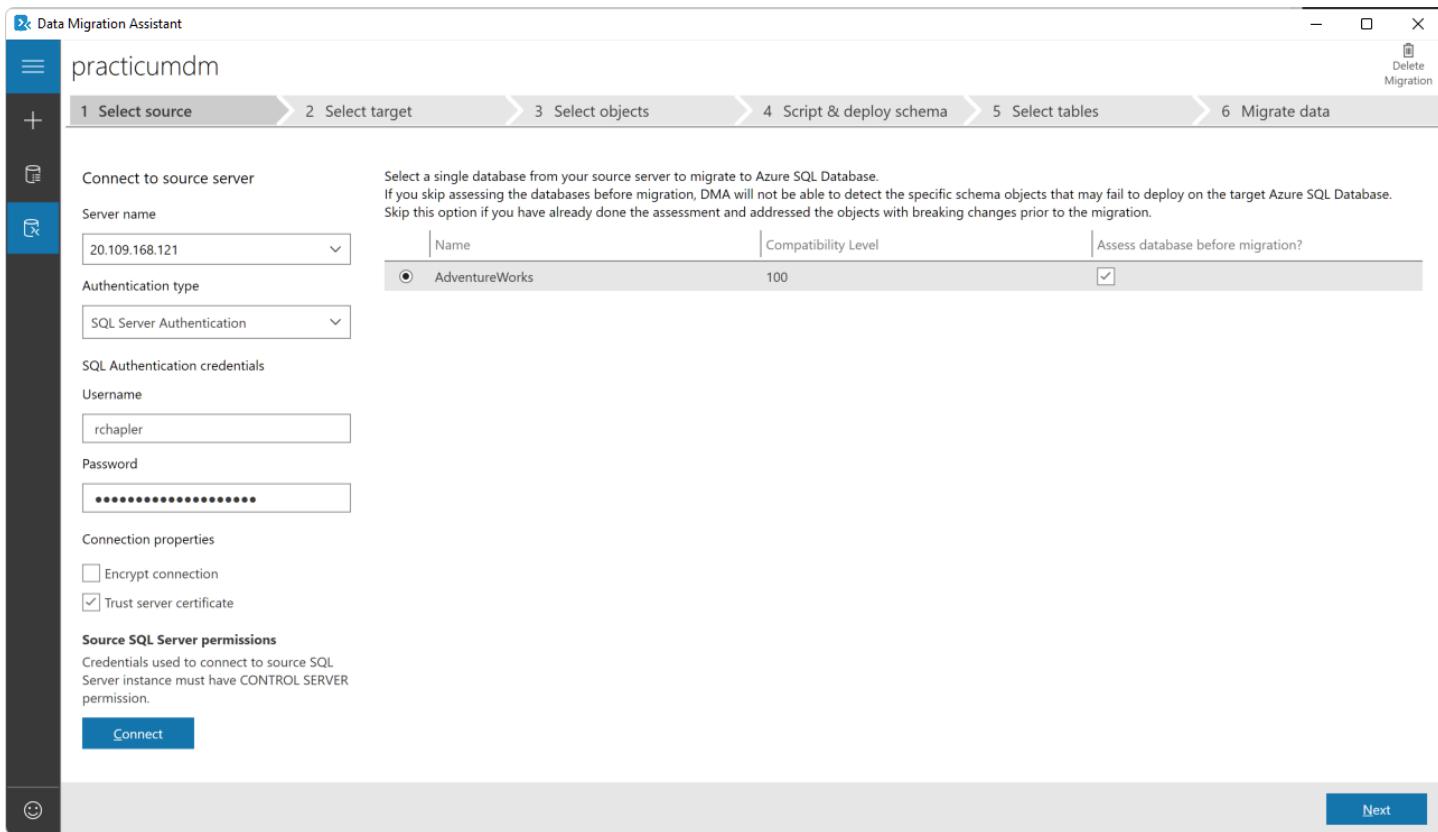
Open **Data Migration Assistant** and click **+** to create a new project



On the “Create Data Share” popout, enter values for the following items:

<b>Project Type</b>	Select the <b>Migration</b> radio button
<b>Project Name</b>	Enter a meaningful name aligned with standards
<b>Source Server Type</b>	Select “SQL Server”
<b>Target Server Type</b>	Select “Azure SQL Database”
<b>Migration Scope</b>	Select “Schema and data”

Click the **Create** button.



On the “**1. Select source**” page, enter values for the following items:

<b>Server Name</b>	Enter the “ <b>Public IP address</b> ” for your SQL Server 2008 virtual machine
<b>Authentication Type</b>	Select “ <b>Windows Authentication</b> ” and enter credentials
<b>Username</b>	
<b>Password</b>	
<b>Encrypt Connection</b>	Unchecked
<b>Trust Server Certificate</b>	Checked

Click the **Connect** button, confirm selection of **AdventureWorks** and then click the **Next** button.

Screenshot of the Data Migration Assistant "2. Select target" page. The page shows the configuration for connecting to a target Azure SQL Database. The server name is set to "practicumsds.database.windows.net", authentication type is "SQL Server Authentication", and the database name is "practicumsd". The compatibility level is set to 150. Connection properties include "Encrypt connection" (unchecked) and "Trust server certificate" (checked). A note about target permissions is present. The "Connect" button is visible at the bottom left.

On the “2. Select target” page, enter values for the following items:

<b>Server Name</b>	Enter a meaningful name aligned with standards
<b>Authentication Type</b>	Select “SQL Server Authentication” and enter credentials
<b>Username</b>	
<b>Password</b>	
<b>Encrypt Connection</b>	Unchecked
<b>Trust Server Certificate</b>	Checked

Click the **Connect** button, confirm selection of your SQL database, and then click the **Next** button.

 Data Migration Assistant

# practicumdm

1 Select source ✓ 2 Select target ✓ 3 Select objects 4 Script & deploy schema 5 Select tables 6 Migrate data

Source database AdventureWorks 20.109.168.121	Target database practicumsd practicumsd.database.windows.net	Assessment issues No collected objects with blocking issues No collected objects with other issues
---	--	--

Select the schema objects from your source database that you would like to migrate to Azure SQL Database.

- DDL Triggers**
  - ddlDatabaseTriggerLog
- Full-Text Catalogs**
  - AW2008FullTextCatalog
- Schemas**
  - HumanResources
  - Person
  - Production
  - Purchasing
  - Sales
- Stored Procedures**
  - dbo.uspGetBillOfMaterials
  - dbo.uspGetEmployeeManagers
  - dbo.uspGetManagerEmployees
  - dbo.uspGetWhereUsedProductID
  - dbo.usp.LogError
  - dbo.uspPrintError
  - dbo.uspSearchCandidateResumes
  - HumanResources.uspUpdateEmployeeHireInfo
  - HumanResources.uspUpdateEmployeeLogin

Select an object to view any issues found for that object

Back Generate SQL script

On the “3. Select objects” page, review schema objects, taking special note of any flagged items.

Click the “**Generate SQL script**” button in the bottom-right when you are ready to continue.

The screenshot shows the Data Migration Assistant interface with the title 'practicumdm'. The top navigation bar includes 'Data Migration Assistant', a back arrow, and a 'Delete Migration' button. Below the title, a progress bar indicates the migration process: '1 Select source' (green checkmark), '2 Select target' (green checkmark), '3 Select objects' (green checkmark), '4 Script & deploy schema' (blue arrow), '5 Select tables' (grey arrow), and '6 Migrate data' (grey arrow). The main area displays configuration details for the 'Source database' (AdventureWorks, IP 20.109.168.121) and 'Target database' (practicumsd, URL practicumsds.database.windows.net). It also shows 'Assessment issues' with a note: 'No selected objects with blocking issues' and 'No selected objects with other issues'. A message below states: 'This script was generated for the selected schema objects. Review the script, make edits if necessary, and click "Deploy schema" to deploy to Azure SQL Database. Any selected users were not scripted; these will be migrated separately upon clicking "Deploy schema." SQL logins associated with selected users will be recreated with strong, random passwords. You will need to change these passwords and enable them again on the target.' A 'Generated script' section contains the T-SQL migration script, and a 'Deployment results' section shows 2,236 commands executed with 0 errors.

Source database  
AdventureWorks  
20.109.168.121

Target database  
practicumsd  
practicumsds.database.windows.net

Assessment issues  
No selected objects with blocking issues  
No selected objects with other issues

This script was generated for the selected schema objects. Review the script, make edits if necessary, and click "Deploy schema" to deploy to Azure SQL Database. Any selected users were not scripted; these will be migrated separately upon clicking "Deploy schema." SQL logins associated with selected users will be recreated with strong, random passwords. You will need to change these passwords and enable them again on the target.

Generated script

Previous issue Next issue Save Copy

```
***** DMA Schema Migration Deployment Script Script Date: 11/12/2021 10:38:14 AM *****

***** Object: FullTextCatalog [AW2008FullTextCatalog] Script Date: 11/12/2021 10:38:08 AM *****
IF NOT EXISTS (SELECT * FROM sys.fulltextcatalogs ftc WHERE ftc.name = N'AW2008FullTextCatalog')
CREATE FULLTEXT CATALOG [AW2008FullTextCatalog] WITH ACCENT_SENSITIVITY = ON
AS DEFAULT

GO
***** Object: Schema [HumanResources] Script Date: 11/12/2021 10:38:08 AM *****
IF NOT EXISTS (SELECT * FROM sys.schemas WHERE name = N'HumanResources')
EXEC sys.sp_executesql N'CREATE SCHEMA [HumanResources]'

GO
IF NOT EXISTS (SELECT * FROM sys.fn_listextendedproperty(N'MS_Description', N'SCHEMA',N'HumanResources', N
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Contains objects related to employees and
GO
***** Object: Schema [Person] Script Date: 11/12/2021 10:38:08 AM *****
IF NOT EXISTS (SELECT * FROM sys.schemas WHERE name = N'Person')
EXEC sys.sp_executesql N'CREATE SCHEMA [Person]'

GO
IF NOT EXISTS (SELECT * FROM sys.fn_listextendedproperty(N'MS_Description', N'SCHEMA',N'Person', NULL,NUL
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Contains objects related to names and ad
```

Deployment results (2,236 commands executed, 0 errors)

Previous error Next error Export

Executed 2,230 of 2,236: ALTER AUTHORIZATION ON SCHE...
Command executed successfully.
Executed 2,231 of 2,236: /****** Indexes For Views Script Dat...
Command executed successfully.
Executed 2,232 of 2,236: IF NOT EXISTS (SELECT * FROM sys.in...
Command executed successfully.
Executed 2,233 of 2,236: IF NOT EXISTS (SELECT * FROM sys.fn...
Command executed successfully.
Executed 2,234 of 2,236: SET ARITHABORT ON SET CONCAT_N...
Command executed successfully.
Executed 2,235 of 2,236: IF NOT EXISTS (SELECT * FROM sys.in...
Command executed successfully.
Executed 2,236 of 2,236: IF NOT EXISTS (SELECT * FROM sys.fn...
Command executed successfully.

On the “4. Script & deploy schema” page, review the generated script.

Click the “Deploy schema” button when you are ready to execute the script and create schema on the target database.

Click the “Migrate data” button when you are ready to continue.

The screenshot shows the Data Migration Assistant interface at step 5: Select tables. The source database is AdventureWorks (20.109.168.121) and the target database is practicummsd (practicumsds.database.windows.net). A note at the top right suggests changing the Azure SQL Database performance level to P15. Below, a table lists 65/71 selected tables, including [HumanResources].[Department], [HumanResources].[Employee], [HumanResources].[EmployeeDepartmentHistory], [HumanResources].[EmployeePayHistory], [HumanResources].[JobCandidate], [HumanResources].[Shift], [Person].[Address] (selected), [Person].[AddressType] (selected), [Person].[BusinessEntity] (selected), and [Person].[BusinessEntityAddress] (selected). The 'Start data migration' button is visible at the bottom right.

Table name	Row count	Ready to move
[HumanResources].[Department]	16	OK
[HumanResources].[Employee]	290	OK
[HumanResources].[EmployeeDepartmentHistory]	296	OK
[HumanResources].[EmployeePayHistory]	316	OK
[HumanResources].[JobCandidate]	13	OK
[HumanResources].[Shift]	3	OK
<input checked="" type="checkbox"/> [Person].[Address]	19,614	OK
<input checked="" type="checkbox"/> [Person].[AddressType]	6	OK
<input checked="" type="checkbox"/> [Person].[BusinessEntity]	20,777	OK
<input checked="" type="checkbox"/> [Person].[BusinessEntityAddress]	19,614	OK

On the “5. Select tables” page, review the list of tables and exclude any that you do not wish to migrate.

Click the “Start data migration” button when you are ready to continue.

Data Migration Assistant

practicumdm

1 Select source ✓ 2 Select target ✓ 3 Select objects ✓ 4 Script & deploy schema ✓ 5 Select tables ✓ 6 Migrate data

Delete Migration

Server objects: 65 In-progress: 0 Successful: 65 Warnings: 0 Failed: 0

Source database: AdventureWorks  
Target database: practicumdsd  
Address: 20.109.168.121  
practicumsd.database.windows.net

Tables (65)

Status	Table name	Migration details
✓	[Person].[Address]	Migration successful. Duration: 0 hrs 0 mins 2 secs
✓	[Person].[AddressType]	Migration successful. Duration: 0 hrs 0 mins 1 secs
✓	[Person].[BusinessEntity]	Migration successful. Duration: 0 hrs 0 mins 1 secs
✓	[Person].[BusinessEntityAddress]	Migration successful. Duration: 0 hrs 0 mins 1 secs
✓	[Person].[BusinessEntityContact]	Migration successful. Duration: 0 hrs 0 mins 1 secs
✓	[Person].[ContactType]	Migration successful. Duration: 0 hrs 0 mins 0 secs
✓	[Person].[CountryRegion]	Migration successful. Duration: 0 hrs 0 mins 2 secs
✓	[Person].[EmailAddress]	Migration successful. Duration: 0 hrs 0 mins 1 secs
✓	[Person].[Password]	Migration successful. Duration: 0 hrs 0 mins 3 secs
✓	[Person].[Person]	Migration successful. Duration: 0 hrs 0 mins 30 secs
✓	[Person].[PersonPhone]	Migration successful. Duration: 0 hrs 0 mins 1 secs
✓	[Person].[PhoneNumberType]	Migration successful. Duration: 0 hrs 0 mins 0 secs

Migration complete. Duration: 0h 0m 41s

Objective Complete... congratulations!

## Objective - Linux + AzCopy | Synchronize Unstructured Data, Server-to-Cloud

**Use Case:** Customer XYZ shared the following requirements:

- **Unstructured data (images)** are regularly loaded to an **on-prem Linux server**
  - This configuration is owned by a separate group and cannot be changed to load directly to Azure
- Synchronization of new data must occur **as soon as it becomes available**

Follow the instructions in this section to **automate recurring synchronization of unstructured data {e.g., image files}** from an on-prem **Linux server to Azure Storage**.

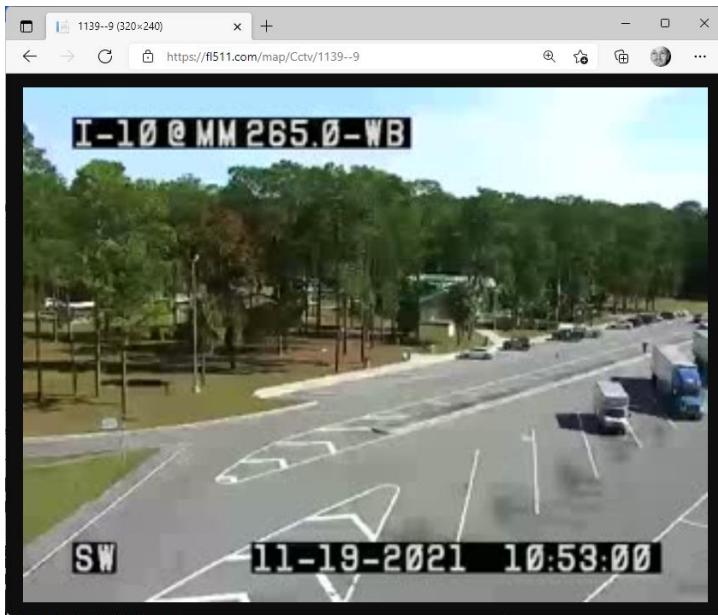
### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Storage Account with container named **images** and Shared Access Token (with **Read, Write, Delete, List** permissions)
- Virtual Machine (Linux/Ubuntu) ... intended to mimic the customers' on-prem configuration

### Step 2: Stage Sample Data

In this section, we will load three sample images from an online traffic cam source ([1139--9 \(320x240\) \(f1511.com\)](https://f1511.com/map/Cctv/1139--9)) onto our Linux virtual machine.



Navigate to Azure Cloud Shell.

#### Step 1: Connect using SSH

Update and execute the following command to connect to the server using SSH:

```
ssh -i ~/.ssh/id_rsa rchabler@{public ip address}
```

#### Step 2: Create Images Directory

Execute the following command to create a directory for our sample images:

```
mkdir images
```

#### Step 3: Get Sample Images

Execute the following command to get a sample image from the traffic cam source:

```
wget -O images/01.jpg https://f1511.com/map/cctv/1139--9
```

Update and execute the previous command to get additional, updated sample images (every ~minute); example:

```
 wget -O images/02.jpg https://f1511.com/map/cctv/1139--9  
 wget -O images/03.jpg https://f1511.com/map/cctv/1139--9
```

Execute the following commands to list sample files in the **images** directory:

```
 cd images  
 ls
```

## Step 3: Prepare AzCopy

In this section, we will download, install, and test AzCopy on our Linux virtual machine.

### Download AzCopy

Execute the following command to download the AzCopy installer:

```
 wget https://aka.ms/downloadazcopy-v10-linux
```

You can expect a response like:

```
--2021-11-23 16:21:50-- https://aka.ms/downloadazcopy-v10-linux  
Resolving aka.ms (aka.ms)... 23.63.47.52  
Connecting to aka.ms (aka.ms)|23.63.47.52|:443... connected.  
HTTP request sent, awaiting response... 301 Moved Permanently  
Location: https://azcopyvnext.azureedge.net/release20211027/azcopy_linux_amd64_10.13.0.tar.gz [following]  
--2021-11-23 16:21:50-- https://azcopyvnext.azureedge.net/release20211027/azcopy_linux_amd64_10.13.0.tar.gz  
Resolving azcopyvnext.azureedge.net (azcopyvnext.azureedge.net)... 104.86.182.19, 104.86.182.43,  
2600:1409:9800:21::17d8:93b8, ...  
Connecting to azcopyvnext.azureedge.net (azcopyvnext.azureedge.net)|104.86.182.19|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 11876012 (11M) [application/gzip]  
Saving to: 'downloadazcopy-v10-linux'  
  
downloadazcopy-v10-linux  
100%[=====] 11.33M --.  
.KB/s in 0.08s  
  
2021-11-23 16:21:50 (141 MB/s) - 'downloadazcopy-v10-linux' saved [11876012/11876012]
```

### Extract Download

Execute the following command to extract the downloaded GZIP file:

```
 tar -xvf downloadazcopy-v10-linux
```

You can expect a response like:

```
azcopy_linux_amd64_10.13.0/  
azcopy_linux_amd64_10.13.0/NOTICE.txt  
azcopy_linux_amd64_10.13.0/azcopy
```

### AzCopy Sync

Update and execute the following command to test file synchronization using AzCopy:

```
./azcopy_linux_amd64_10.13.0/azcopy sync "/home/rchapler/images"  
"https://practicumsa.blob.core.windows.net/images?sp=rw&st=2021-11-23T20:23:01Z&se=2022-10-  
01T03:23:01Z&spr=https&sv=2020-08-04&sr=c&sig=U3o98QJ1NJ8wAldkjBBJv8Eken4N3Rlrr50gLxsq6PE%3D" --recursive
```

You can expect a response like:

```
INFO: Any empty folders will not be processed, because source and/or destination doesn't have full folder support
```

```
Job ffac3e11-2fc7-ee4f-5216-95fd028b1a51 has started
Log file is located at: /home/rchapler/.azcopy/ffac3e11-2fc7-ee4f-5216-95fd028b1a51.log
```

3 Files Scanned at Source, 0 Files Scanned at Destination

Job ffac3e11-2fc7-ee4f-5216-95fd028b1a51 Summary

Files Scanned at Source: 3

Files Scanned at Destination: 0

Elapsed Time (Minutes): 0.0334

Number of Copy Transfers for Files: 3

Number of Copy Transfers for Folder Properties: 0

Total Number Of Copy Transfers: 3

Number of Copy Transfers Completed: 3

Number of Copy Transfers Failed: 0

Number of Deletions at Destination: 0

Total Number of Bytes Transferred: 58702

Total Number of Bytes Enumerated: 58702

Final Job Status: **Completed**

Navigate to the **images** container in your Storage Account and confirm file synchronization.

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
01.jpg	11/23/2021, 9:46:03 ...	Hot (Inferred)		Block blob	19.69 KiB	Available
02.jpg	11/23/2021, 9:46:03 ...	Hot (Inferred)		Block blob	18.56 KiB	Available
03.jpg	11/23/2021, 9:46:03 ...	Hot (Inferred)		Block blob	19.07 KiB	Available

## Step 4: Prepare Script

In this section, we will prepare a script and confirm success.

### Create Script with Vim

Execute the following command to open a text editor:

```
vim script.sh
```

In the editor, type **i** to change to **INSERT** mode.

Update and paste the following command lines into the text editor:

```
#!/bin/sh
```

```
./azcopy_linux_amd64_10.13.0/azcopy sync "/home/rchapler/images"
"https://practicumsa.blob.core.windows.net/images?sp=rw&st=2021-11-23T20:23:01Z&se=2022-10-
01T03:23:01Z&spr=https&sv=2020-08-04&sr=c&sig=U3o98QJ1NJ8wAldkjBBJv8Eken4N3Rlrr50gLxsq6PE%3D" --recursive
```

```
#!/bin/sh
./azcopy_linux_amd64_10.13.0/azcopy sync "/home/rchapler/images" "https://practicumsa.blob.core.windows.net/images?sp=rw&st=2021-11-23T20:23:01Z&se=2022-10-01T03:23:01Z&spr=https&sv=2020-08-04&sr=c&sig=U3o98QJ1NJ8wAldkjBBJv8Eken4N3Rlrr50gLxsq6PE%3D" --recursive
```

Switch to command mode by pressing the **ESC** key.

Press the : (colon) key to open the prompt bar in the bottom left corner of the window.

Type **x** after the colon and press the **Enter** key to save changes and exit.

Back in the Azure Cloud Shell, update and execute the following command to grant permissions to the new script:

```
chmod +x /home/rchapler/script.sh
```

Confirm Success

In the Cloud Shell, update and paste the following command line.

```
/home/rchapler/script.sh
```

Successful processing of the new script will produce a response like:

```
INFO: Any empty folders will not be processed, because source and/or destination doesn't have full folder support
```

```
Job cb947969-d113-624c-6ac6-56274d66baac has started
Log file is located at: /home/rchapler/.azcopy/cb947969-d113-624c-6ac6-56274d66baac.log
```

```
4 Files Scanned at Source, 0 Files Scanned at Destination
```

```
Job cb947969-d113-624c-6ac6-56274d66baac Summary
Files Scanned at Source: 4
Files Scanned at Destination: 4
Elapsed Time (Minutes): 0.0334
Number of Copy Transfers for Files: 0
Number of Copy Transfers for Folder Properties: 0
Total Number Of Copy Transfers: 0
Number of Copy Transfers Completed: 0
Number of Copy Transfers Failed: 0
Number of Deletions at Destination: 0
Total Number of Bytes Transferred: 0
Total Number of Bytes Enumerated: 0
Final Job Status: Completed
```

## Step 5: Schedule Cron Job

In this section, we will schedule a cron job and confirm success.

### Schedule Cron Job

In the Cloud Shell, execute the following command to **open the cron table** {i.e., job scheduler}:

```
crontab -e
```

In Crontab, update and paste the following command line.

```
* * * * * /home/rchapler/script.sh
```

Logic explained...

- **Schedule ... “\* \* \* \* \*”** (in this case, “every minute”)
- **Command ... “/home/rchapler/script.sh”**

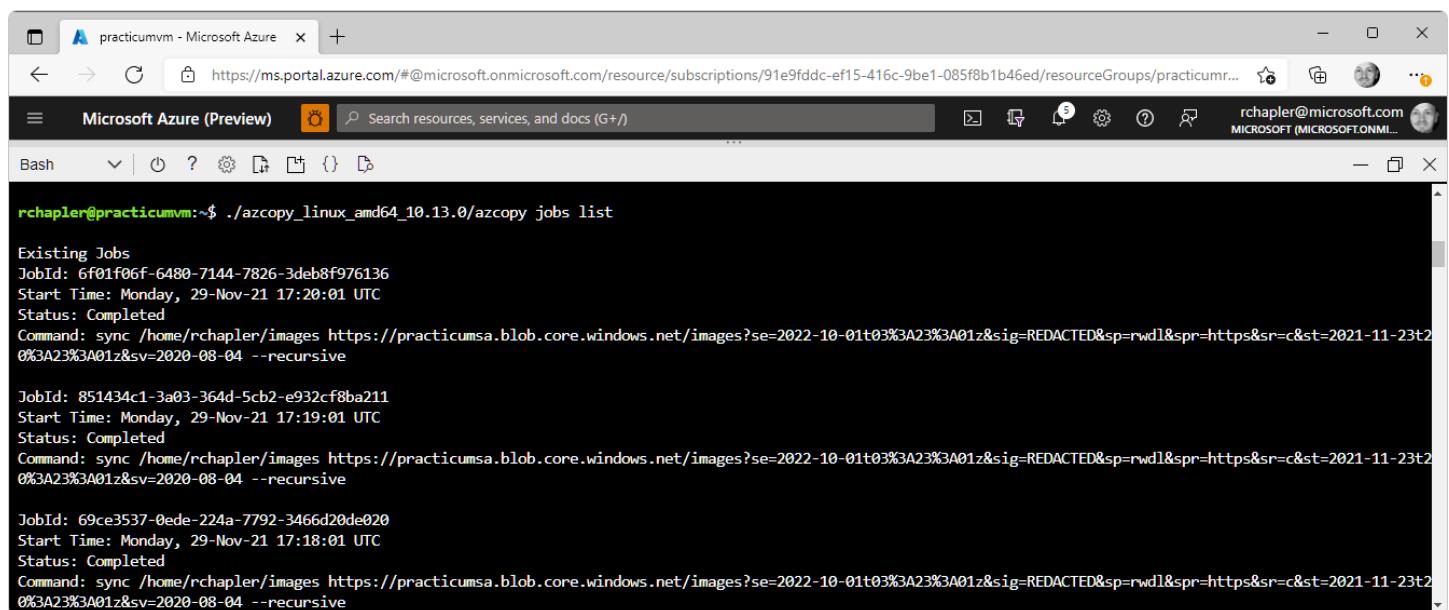
Press CTRL-X on the keyboard to exit crontab; save when prompted.

Back in the Cloud Shell, you should see the message “**crontab: installing new crontab**”

### Confirm Success

In the Cloud Shell, update and execute the following command line to review recent runs of AzCopy.

```
./azcopy_linux_amd64_10.13.0/azcopy jobs list
```



The screenshot shows a Microsoft Azure Cloud Shell interface. The terminal window title is "practicumvm - Microsoft Azure". The command entered was "./azcopy\_linux\_amd64\_10.13.0/azcopy jobs list". The output displays three completed AzCopy jobs. Each job entry includes the JobID, Start Time, Status, and the full command used to run the sync operation.

```
rchapler@practicumvm:~$ ./azcopy_linux_amd64_10.13.0/azcopy jobs list

Existing Jobs
JobID: 6f01f06f-6480-7144-7826-3deb8f976136
Start Time: Monday, 29-Nov-21 17:20:01 UTC
Status: Completed
Command: sync /home/rchapler/images https://practicumsa.blob.core.windows.net/images?se=2022-10-01t03%3A23%3A01z&sig=REDACTED&sp=rwdl&spr=https&sr=c&st=2021-11-23t20%3A23%3A01z&sv=2020-08-04 --recursive

JobID: 851434c1-3a03-364d-5cb2-e932cf8ba211
Start Time: Monday, 29-Nov-21 17:19:01 UTC
Status: Completed
Command: sync /home/rchapler/images https://practicumsa.blob.core.windows.net/images?se=2022-10-01t03%3A23%3A01z&sig=REDACTED&sp=rwdl&spr=https&sr=c&st=2021-11-23t20%3A23%3A01z&sv=2020-08-04 --recursive

JobID: 69ce3537-0ede-224a-7792-3466d20de020
Start Time: Monday, 29-Nov-21 17:18:01 UTC
Status: Completed
Command: sync /home/rchapler/images https://practicumsa.blob.core.windows.net/images?se=2022-10-01t03%3A23%3A01z&sig=REDACTED&sp=rwdl&spr=https&sr=c&st=2021-11-23t20%3A23%3A01z&sv=2020-08-04 --recursive
```

More recent executions are shown first {i.e., you will want to scroll the top of the results to see the most recent execution}.

## Step 6: Confirm Synchronization

Execute the following command to get a new sample image from the traffic cam source:

```
wget -O images/10.jpg https://f1511.com/map/cctv/1139--9
```

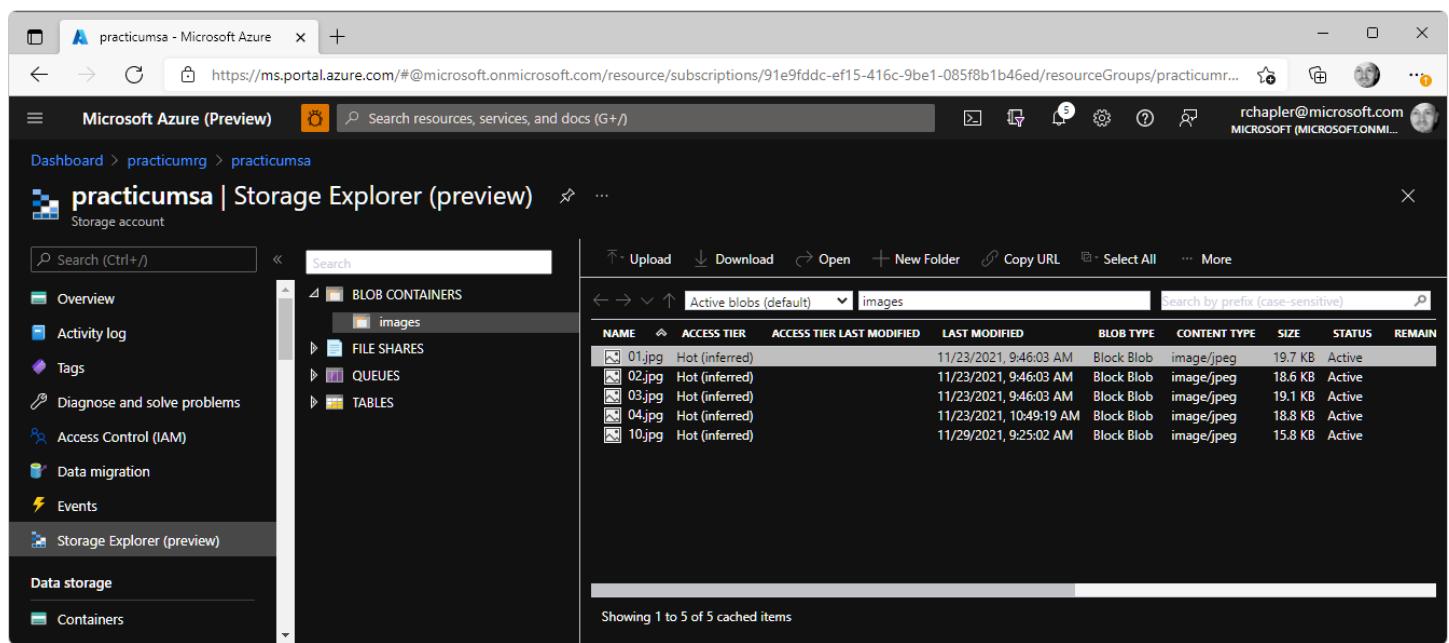
You can expect a response like:

```
--2021-11-29 17:24:45-- https://f1511.com/map/cctv/1139--9
Resolving f1511.com (f1511.com)... 172.67.4.44, 104.22.25.76, 104.22.24.76, ...
Connecting to f1511.com (f1511.com)|172.67.4.44|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16149 (16K) [image/jpeg]
Saving to: 'images/10.jpg'

images/10.jpg
100%[=====] 15.77K --.-KB/s in 0.06s

2021-11-29 17:24:45 (251 KB/s) - 'images/10.jpg' saved [16149/16149]
```

After waiting a minute for the scheduled synchronization, navigate to your storage account, and then **Storage Explorer**.



The screenshot shows the Microsoft Azure Storage Explorer interface. On the left, there's a sidebar with various options like Overview, Activity log, Tags, etc. The main area shows a tree view of storage resources under the 'practicumsa' storage account. A 'BLOB CONTAINERS' node has an 'images' folder expanded. Below the tree is a table listing blobs in the 'images' container. The table includes columns for NAME, ACCESS TIER, LAST MODIFIED, BLOB TYPE, CONTENT TYPE, SIZE, STATUS, and REMAIN. The table shows five files: 01.jpg, 02.jpg, 03.jpg, 04.jpg, and 10.jpg, all of which are 'Hot (inferred)' access tier, 'Block Blob' type, and 'image/jpeg' content type. The sizes range from 18.6 KB to 19.7 KB, and all are marked as 'Active'. At the bottom of the table, it says 'Showing 1 to 5 of 5 cached items'.

NAME	ACCESS TIER	LAST MODIFIED	BLOB TYPE	CONTENT TYPE	SIZE	STATUS	REMAIN
01.jpg	Hot (inferred)	11/23/2021, 9:46:03 AM	Block Blob	image/jpeg	19.7 KB	Active	
02.jpg	Hot (inferred)	11/23/2021, 9:46:03 AM	Block Blob	image/jpeg	18.6 KB	Active	
03.jpg	Hot (inferred)	11/23/2021, 9:46:03 AM	Block Blob	image/jpeg	19.1 KB	Active	
04.jpg	Hot (inferred)	11/23/2021, 10:49:19 AM	Block Blob	image/jpeg	18.8 KB	Active	
10.jpg	Hot (inferred)	11/29/2021, 9:25:02 AM	Block Blob	image/jpeg	15.8 KB	Active	

You should see the new file {i.e., "10.jpg"} included in the **images** container.

*Objective Complete... congratulations!*

## Objective - Synchronize Unstructured Data, On-Prem to On-Prem (feat. File Sync)

Follow the instructions in this section to migrate files using Azure File Sync and a storage account.

### Step 1: Understand Use Case

Pinpoint significant notes and requirements in the customer-provided information:

- Implementing a third-party Content Management System (CMS)
- Must migrate **petabytes of unstructured data**, including media, image, and Portable Document Format (PDF) files
- **Source** server is an **on-prem** server, and **destination** server is a **virtual machine**
- It is critical that **file metadata remains unchanged** during migration {e.g., Created On datetime}

### Step 2: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- File Sync (aka Storage Sync Services)
- Storage Account
- Virtual Machine 1... to mimic the customers' source server (on-prem)
- Virtual Machine 2... to mimic the customers' destination server (virtual machine)

Follow [Objective – Deploy Infrastructure-as-Code \(feat. DevOps + ARM Templates\)](#) instructions to create templates, create a pipeline using the following YAML, and then deploy necessary pre-requisites:

```
trigger:
  branches:
    include:
      - refs/heads/main
  paths:
    include:
      - ARMTemplates
jobs:
- job: sa
  displayName: 'Instantiate Storage Account'
  pool:
    vmImage: windows-latest
  steps:
    - task: AzureResourceManagerTemplateDeployment@3
      inputs:
        ConnectedServiceName: a775d3ec-e4f1-46ed-905b-ddd26a59356c
        subscriptionName: 91e9fddc-ef15-416c-9be1-085f8b1b46ed
        resourceGroupName: practicum
        location: West US
        csmFile: ARM Templates/StorageAccount.json
- job: sss
  displayName: 'Instantiate File Sync (aka Storage Sync Service)'
  pool:
    vmImage: windows-latest
  steps:
    - task: AzureResourceManagerTemplateDeployment@3
      inputs:
        ConnectedServiceName: a775d3ec-e4f1-46ed-905b-ddd26a59356c
        subscriptionName: 91e9fddc-ef15-416c-9be1-085f8b1b46ed
        resourceGroupName: practicum
        location: West US
        csmFile: ARM Templates/FileSync.json
- job: vm1
  displayName: 'Instantiate Virtual Machine #1'
  pool:
    vmImage: windows-latest
  steps:
    - task: AzureResourceManagerTemplateDeployment@3
      displayName: 'ARM Template deployment: practicummvm01'
      inputs:
        ConnectedServiceName: a775d3ec-e4f1-46ed-905b-ddd26a59356c
        subscriptionName: 91e9fddc-ef15-416c-9be1-085f8b1b46ed
        resourceGroupName: practicummvm01
        location: West US
        csmFile: ARM Templates/VirtualMachine.json
- job: vm2
  displayName: 'Instantiate Virtual Machine #2'
  pool:
    vmImage: windows-latest
  steps:
    - task: AzureResourceManagerTemplateDeployment@3
      displayName: 'ARM Template deployment: practicummvm02'
      inputs:
```

```
ConnectedServiceName: a775d3ec-e4f1-46ed-905b-ddd26a59356c
subscriptionName: 91e9fddc-ef15-416c-9be1-085f8b1b46ed
resourceGroupName: practicumvm02
location: West US
csmFile: ARM Templates/VirtualMachine.json
```

The screenshot shows the Azure DevOps Pipelines interface for a project named "practicumadop". A pipeline run titled "#20211222.3 Update Objective - Synchronize Unstructured Data, On-Prem to On-Prem" is currently active. The summary card indicates it was manually run by Rich Chapter at 5:32 AM today, taking 6m 53s. It shows 0 work items and 0 artifacts. The "Jobs" section lists four tasks: "Instantiate Storage Account" (Queued), "Instantiate File Sync (aka Storage Sync Service)" (Queued), "Instantiate Virtual Machine #1" (Success, 3m 29s), and "Instantiate Virtual Machine #2" (Running). The left sidebar shows other project navigation options like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, Artifacts, and Project settings.

Pipelines - Run 20211222.3

[https://dev.azure.com/practicumadop/\\_build/results?buildId=34&view=results](https://dev.azure.com/practicumadop/_build/results?buildId=34&view=results)

Azure DevOps / Pipelines / Synchronize Unstructured D... / 20211222.3

#20211222.3 Update Objective - Synchronize Unstructured Data, On-Prem to On-Prem

Manually run by Rich Chapter

View 2 changes

Name	Status	Duration
Instantiate Storage Account	Queued	
Instantiate File Sync (aka Storage Sync Service)	Queued	
Instantiate Virtual Machine #1	Success	3m 29s
Instantiate Virtual Machine #2	Running	2m 39s

## Step 3: Install Agent

Navigate to the first virtual machine, click the **Connect** button, and select **RDP** from the resulting dropdown.

On the resulting “**Connect with RDP**” page, click the “**Download RDP File**” button.

Click the “**Open file**” link under your RDP file in the **Downloads** dropdown.

Click the **Connect** button on the resulting “**Remote Desktop Connection**” popup.

In the “**Windows Security**” popup, click the “**More choices**” link, then “**Use a different account**” in the resulting choices, then click **OK**.

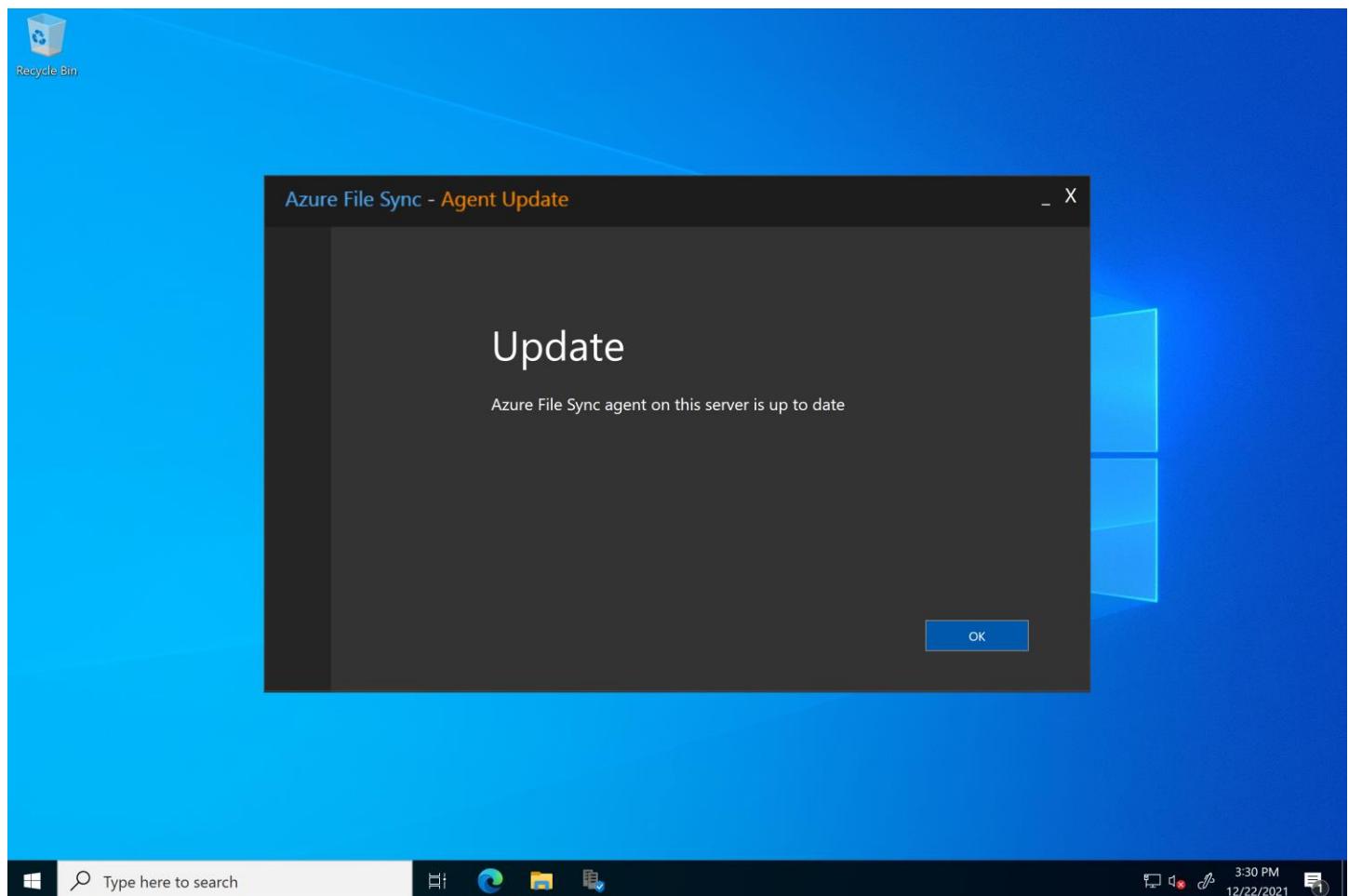
Back at the top of the “**Windows Security**” popup, enter credentials included in the YAML template.

**Note:** “*User name*” should include the name of the server {e.g., “practicumvm01\Admin123”}.

Once connected to the virtual machine, browse to [Download Azure File Sync Agent from Official Microsoft Download Center](#).

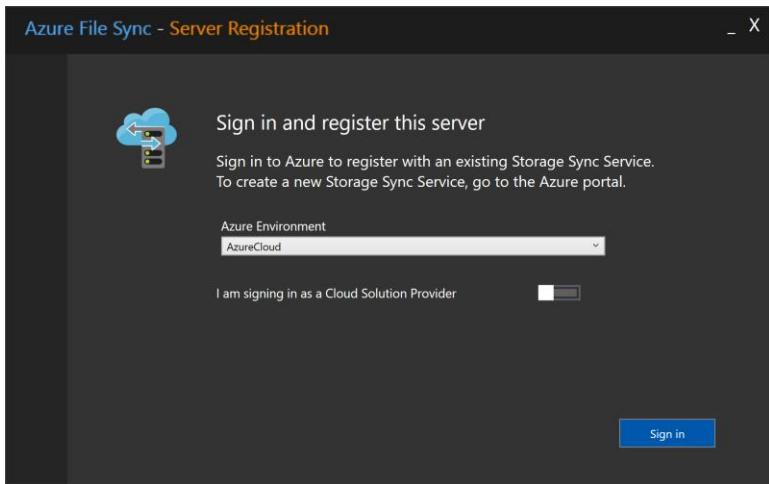
Click the **Download** button, and select the appropriate download {e.g., **StorageSyncAgent\_WS2022.msi**} and click the **Next** button.

Open the downloaded file and complete the installation.



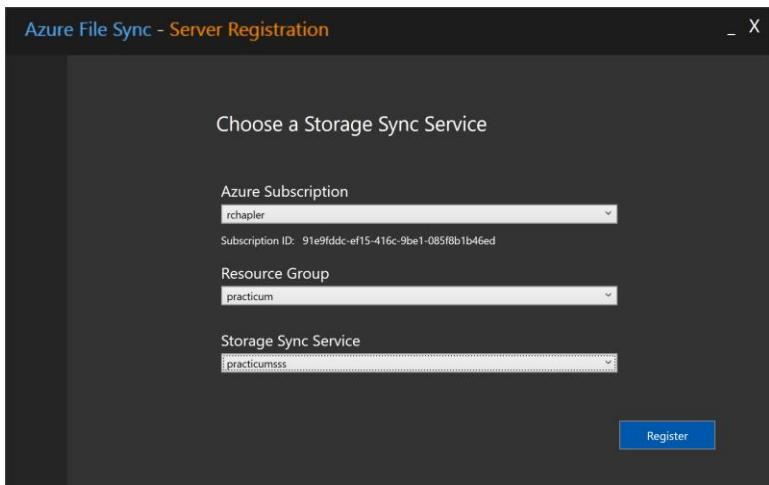
The Azure File Sync, server registration utility will open.

Click the **OK** button on the Update page.



Confirm “Azure Environment” selection and click the “Sign in” button on the “Sign in and register this server” page.  
Complete the sign in process.

*Note: You will likely have to add trusted sites and repeat the sign in process since this is a new, unconfigured server.*



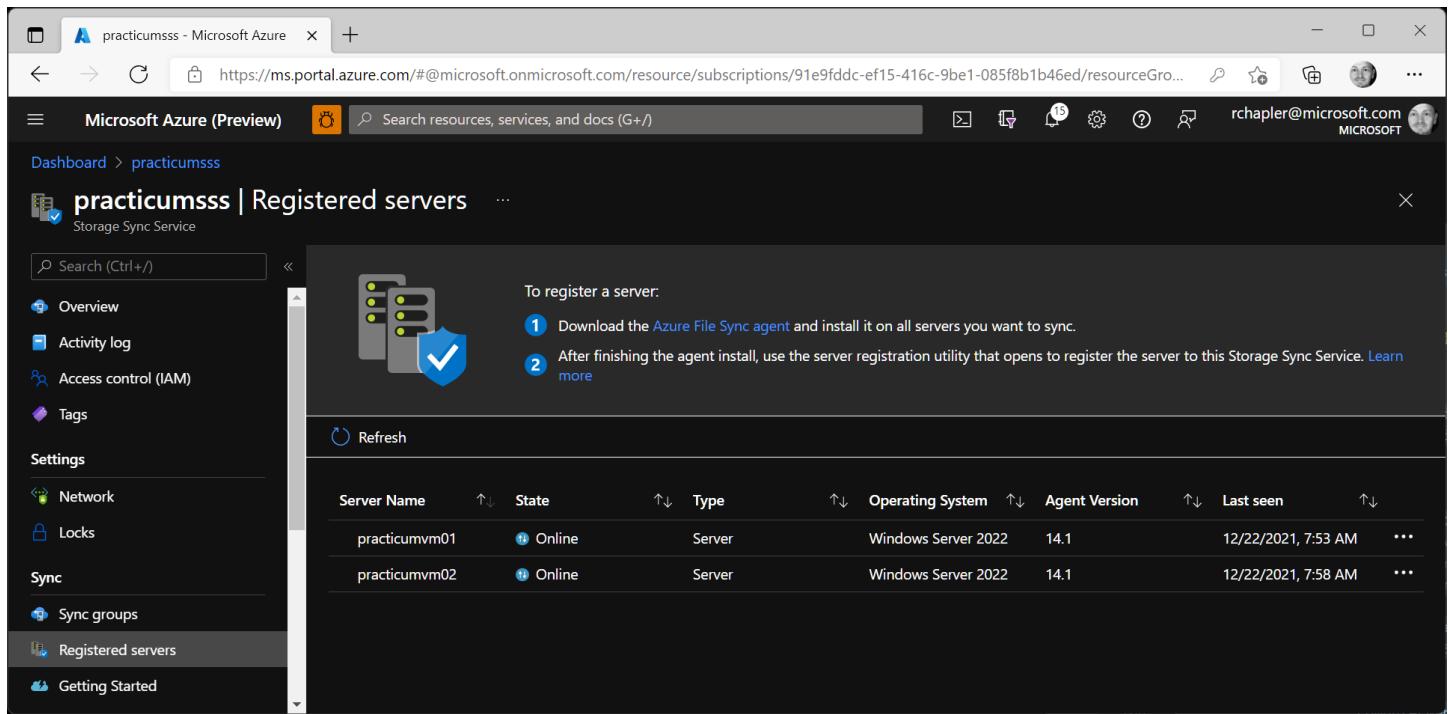
On the “Choose a Storage Sync Service” page, enter values for the following items:

Azure Subscription	Select your subscription
Resource Group	Select your resource group
Storage Sync Service	Select your storage sync service

Click the **Register** button and allow time for the “Network connectivity test” to reach “Passed. You are all set.”  
Close (or minimize) the “Remote Desktop Connection” window.

Repeat this process for the second virtual machine.

## Confirm Success



The screenshot shows the Microsoft Azure Storage Sync Service Registered servers page. The left sidebar includes options like Overview, Activity log, Access control (IAM), Tags, Network, Locks, Sync groups, Registered servers (which is selected and highlighted in grey), and Getting Started. The main content area displays a table of registered servers:

Server Name	State	Type	Operating System	Agent Version	Last seen	Actions
practicumvm01	Online	Server	Windows Server 2022	14.1	12/22/2021, 7:53 AM	...
practicumvm02	Online	Server	Windows Server 2022	14.1	12/22/2021, 7:58 AM	...

To register a server:

- 1 Download the Azure File Sync agent and install it on all servers you want to sync.
- 2 After finishing the agent install, use the server registration utility that opens to register the server to this Storage Sync Service. [Learn more](#)

Navigate to your instance of Storage Sync Service, then “**Registered Servers**” in the **Sync** group of the right-hand navigation. You should see your two registered servers.

## Step 4: Create File Share

Navigate to your Storage Account, then “**File Shares**” in the “**Data storage**” group of the right-hand navigation.

Click the “**+ File share**” button and enter a meaningful Name value in the resulting popout {e.g., practicumsafs}.

Click the **Create** button.

## Step 5: Create Sync Group

Navigate to your instance of Storage Sync Service, then “**Registered Servers**” in the **Sync** group of the right-hand navigation.

Click the “**+ Sync group**” button.

On the “**Choose a Storage Sync Service**” page, enter values for the following items:

<b>Sync Group Name</b>	Enter a meaningful name aligned with standards {e.g., practicumssssg}
<b>Subscription</b>	Select your subscription
<b>Storage Account</b>	Click the “ <b>Select storage account</b> ” button and select your storage account on the resulting screen
<b>Azure File Share</b>	Select your storage account, file share

Click the **Create** button.

## Step 6: Add Server Endpoint

Click into the new Sync Group and then click the “**Add server endpoint**” button at the top of the screen.

On the resulting “**Add server endpoint**” popout, enter values for the following items:

<b>Registered Server</b>	Select your first virtual machine
<b>Path</b>	Enter “C:\Temp”

Click the **Create** button and then repeat the process for the second virtual machine.

## Step 7: Confirm Success

Connect to your first virtual machine and then open File Explorer.

Navigate to C:\Temp, right-click and select New > Bitmap Image in the resulting dropdowns.

Take note of the metadata {e.g., **Created** datetime}.

Connect to your second virtual machine and then open File Explorer.

You should see a synchronized copy of the “New Bitmap Image.bmp” file and metadata will match.

Repeat this exercise with a file update {e.g., rename} and a file delete.

*Objective Complete... congratulations!*

# Engineering

Objectives in this section describe methods for extracting, transforming, and loading data.

## Objective - Synapse | Use External Data

Follow the instructions in this section to **explore methods of using data from data products external to Synapse**.

### Step 1: Instantiate Prerequisites

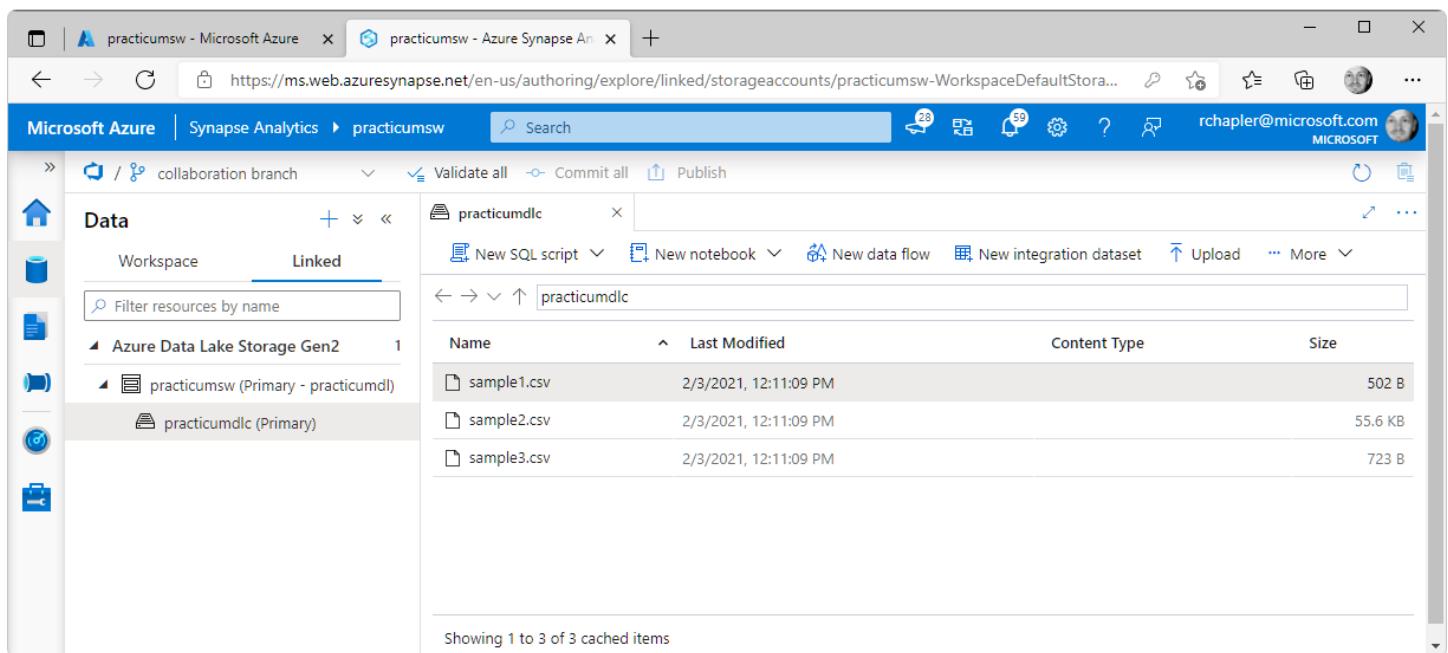
To complete this objective, you must instantiate the following resources:

- Data Lake (with sample data)
- Synapse (with Serverless SQL Database and Apache Spark Pool)

### Step 2: Explore Alternate Methods

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “Open Synapse Studio” rectangle.

Click the **Data** icon in the navigation.



The screenshot shows the Azure Synapse Analytics Data blade. On the left, there's a navigation sidebar with icons for Home, Data, Datasets, Pipelines, and Jobs. The 'Data' icon is selected. The main area has tabs for 'Workspace' and 'Linked'. Under 'Linked', it shows 'practicumdlc' which is expanded to show 'practicumsw (Primary - practicumdl)' and 'practicumdlc (Primary)'. A search bar at the top right contains 'practicumdlc'. Below the search bar is a table with three rows of data:

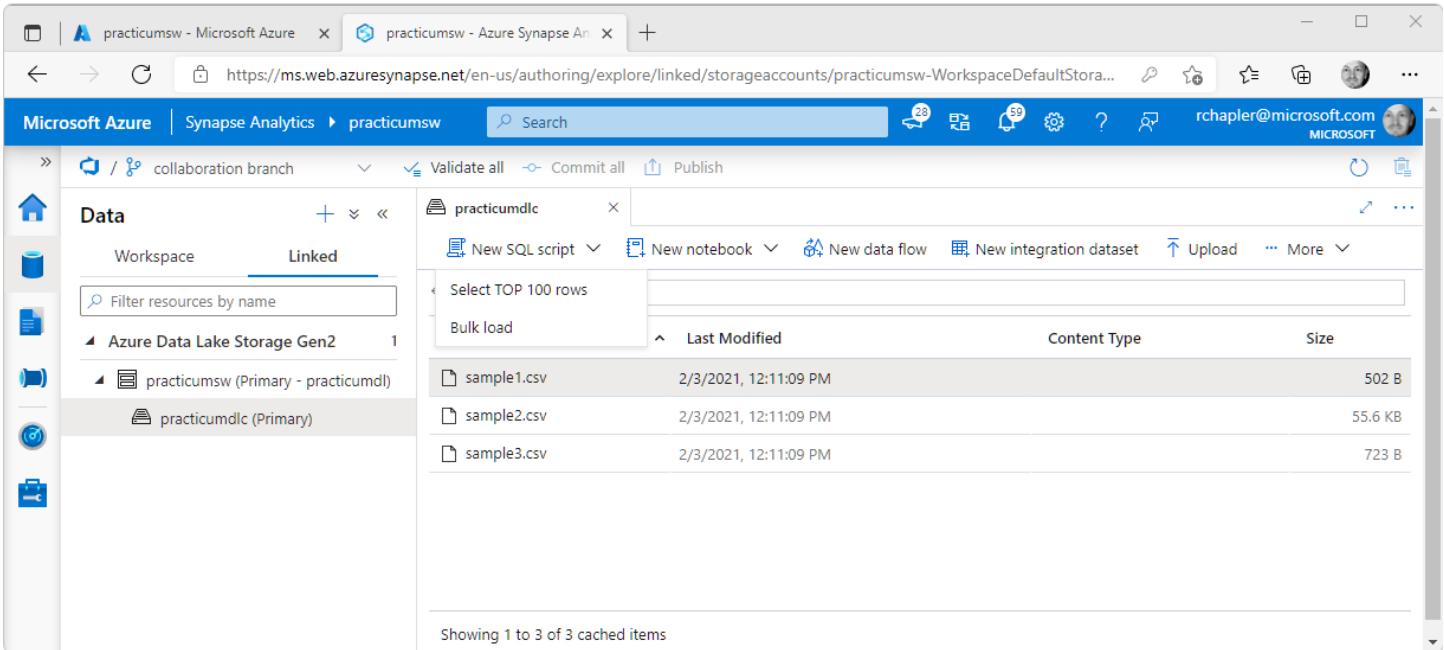
Name	Last Modified	Content Type	Size
sample1.csv	2/3/2021, 12:11:09 PM		502 B
sample2.csv	2/3/2021, 12:11:09 PM		55.6 KB
sample3.csv	2/3/2021, 12:11:09 PM		723 B

At the bottom of the table, it says 'Showing 1 to 3 of 3 cached items'.

Click the **Linked** tab and Expand navigation.

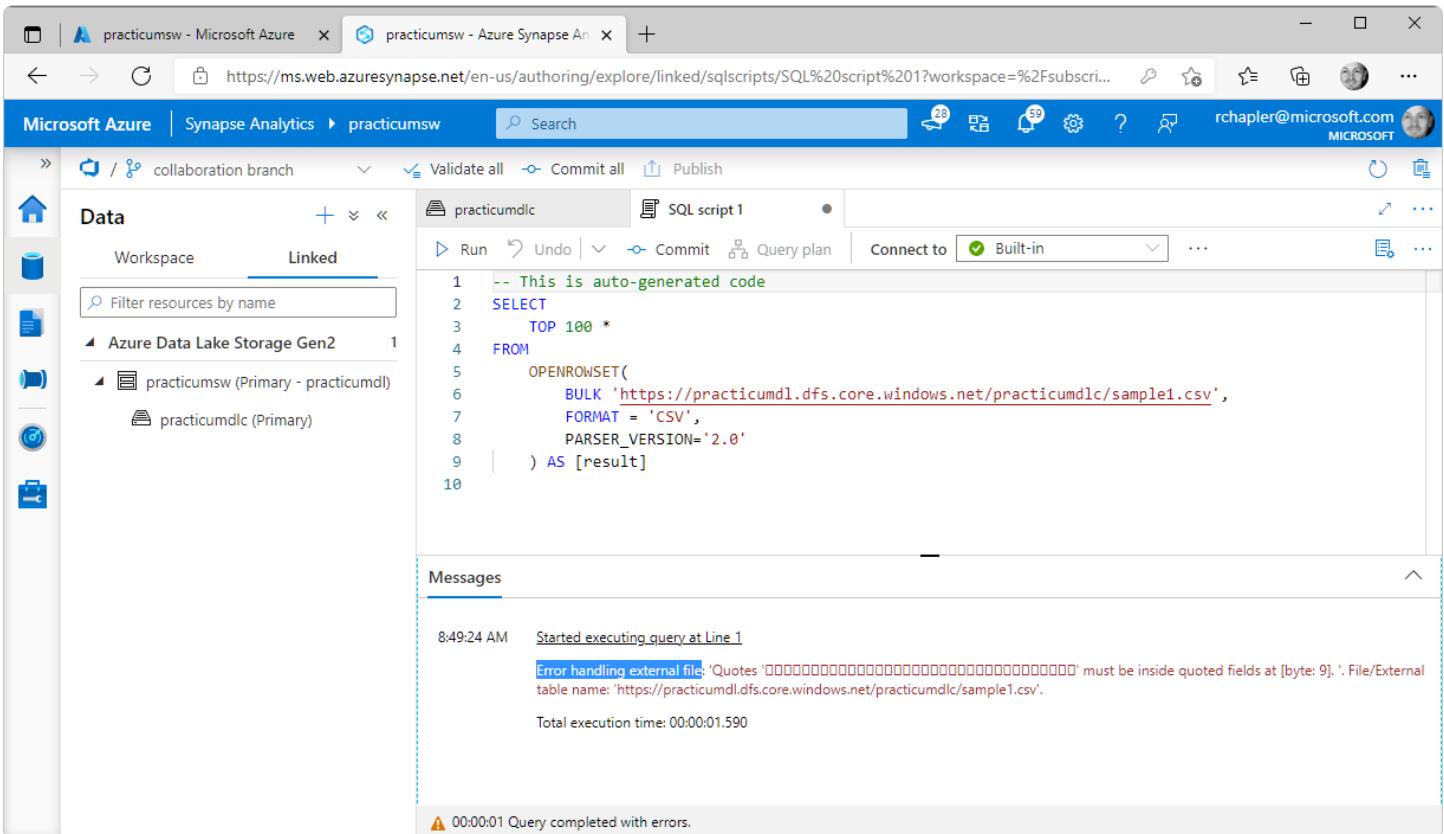
You will see that the Azure Data Lake Storage Gen2 specified during Synapse instantiation {i.e., practicumdlc} is surfaced with no additional configuration.

## Method 1: SQL Script (OPENROWSET)



The screenshot shows the Microsoft Azure Synapse Analytics Data blade. On the left, there's a navigation pane with icons for Home, Data, Workspace, and Linked. Under 'Linked', it shows 'practicumdlc' with three CSV files: 'sample1.csv', 'sample2.csv', and 'sample3.csv'. Each file has a timestamp of '2/3/2021, 12:11:09 PM' and a size: 502 B, 55.6 KB, and 723 B respectively. Below the table, it says 'Showing 1 to 3 of 3 cached items'.

Click the “New SQL script” button and “SELECT TOP 100 rows” from the resulting dropdown.



The screenshot shows the Microsoft Azure Synapse Analytics SQL Editor. The top bar shows the URL 'https://ms.web.azure-synapse.net/en-us/authoring/explore/linked/sqlscripts/SQL%20script%201?workspace=%2Fsubscriptions%2F...'. The editor interface includes tabs for Run, Undo, Commit, Query plan, and Connect to (set to Built-in). The main area contains the following SQL code:

```
1 -- This is auto-generated code
2 SELECT
3     TOP 100 *
4 FROM
5     OPENROWSET(
6         BULK 'https://practicumdl.dfs.core.windows.net/practicumdlc/sample1.csv',
7         FORMAT = 'CSV',
8         PARSE_DATE='2021-02-03'
9     ) AS [result]
10
```

Below the code, the 'Messages' section shows:

- 8:49:24 AM Started executing query at Line 1
- Error handling external file: 'Quotes '\u00e2\u20ac\u201d must be inside quoted fields at [byte: 9]. ' . File/External table name: 'https://practicumdl.dfs.core.windows.net/practicumdlc/sample1.csv'.
- Total execution time: 00:00:01.590

At the bottom, a message states: 00:00:01 Query completed with errors.

The resulting auto-generated code uses OPENROWSET(...) and a built-in, serverless SQL Pool to pull data directly from the CSV file in the ADLS container.

When we run the code (assuming you are using the same sample files, of course), we get an error about quote-handling.

Given the formatting of the sample file, we cannot run this without changes to the code.

Update the code to:

```

SELECT TOP 100 *
FROM OPENROWSET(
    BULK 'https://practicumdl.dfs.core.windows.net/practicumdlc/sample3.csv',
    FORMAT = 'CSV',
    PARSER_VERSION='2.0',
    HEADER_ROW = TRUE,
    FIELDQUOTE = ''
) AS [result]

```

... and then, re-Run...

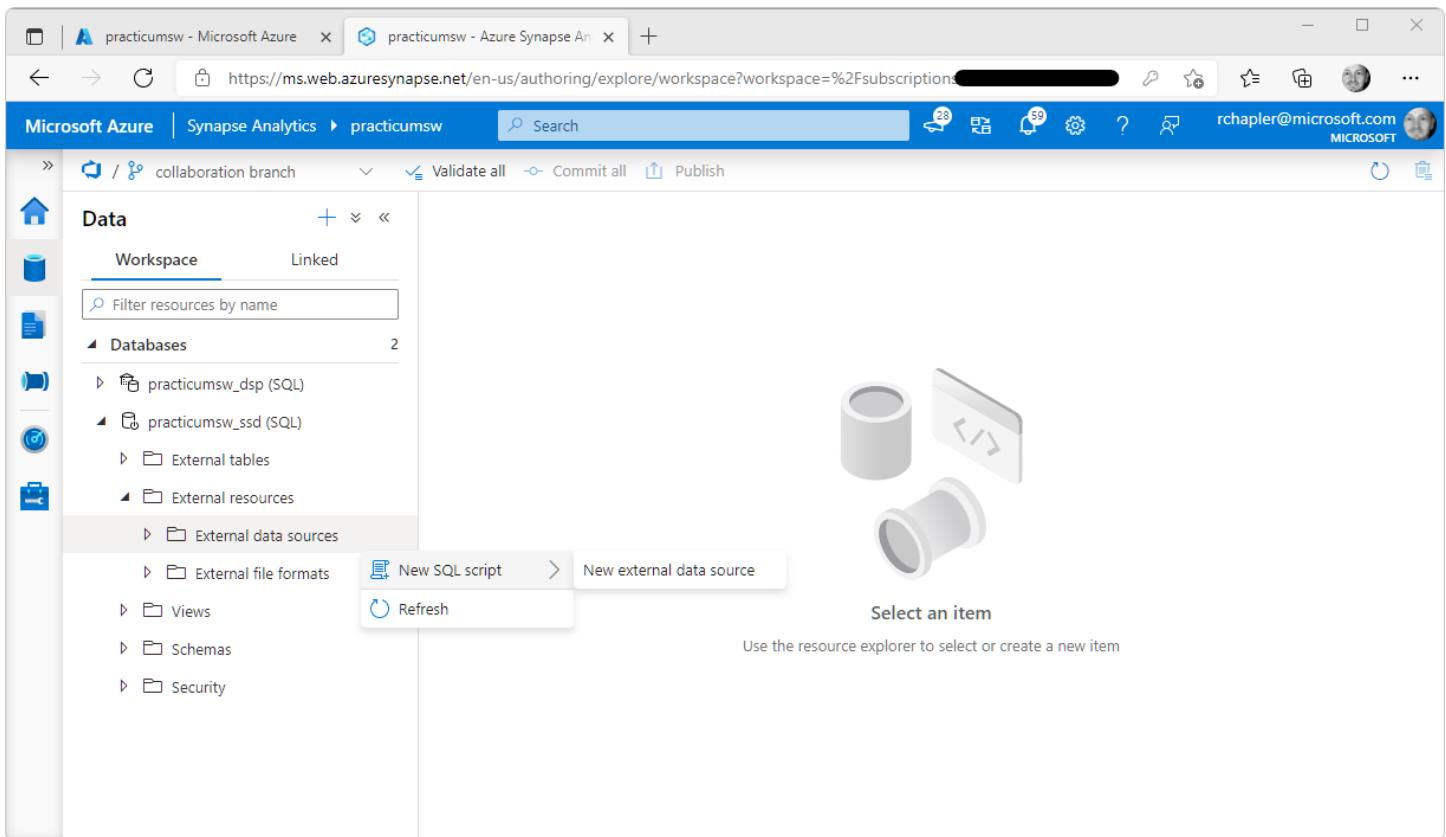
The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. On the left, there's a navigation sidebar with icons for Home, Data, Workspace, and Linked. The 'Data' section is selected, showing 'Azure Data Lake Storage Gen2' and 'practicumsw (Primary - practicumdl)' under 'Linked'. The main area has tabs for 'practicumdlc' and 'SQL script 1'. The 'SQL script 1' tab contains the T-SQL code provided above. Below the code, the 'Results' tab is selected, showing a table with two columns: 'Game Number' and 'Game Length'. The data rows are 1, 30; 2, 29; and 3, 31. At the bottom of the results pane, it says '00:00:11 Query executed successfully.'

## Method 2: External Table (Serverless SQL Database)

Click the **Workspace** tab.

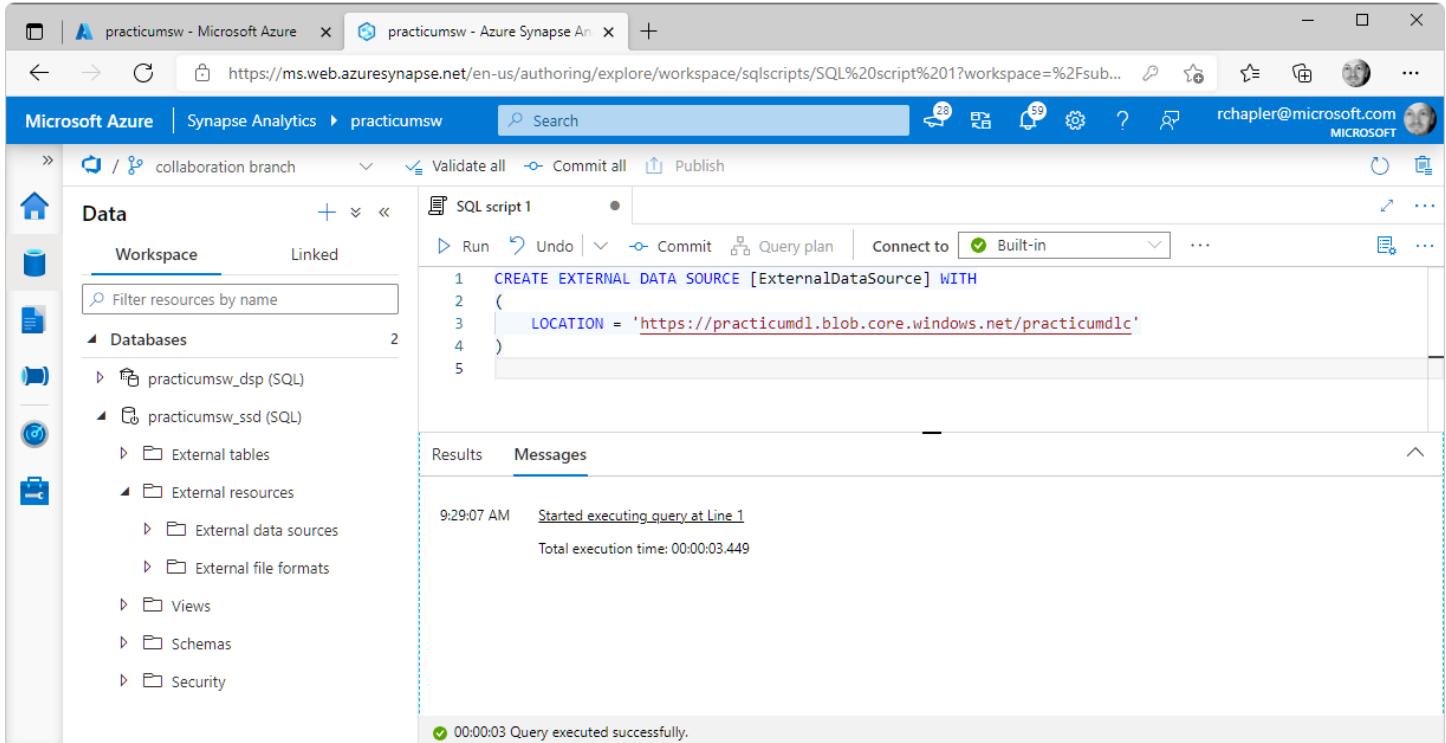
External Data Source

Expand navigation to and then right-click on “**External data sources**”.



The screenshot shows the Microsoft Azure Synapse Analytics workspace Data blade. In the left sidebar, under 'External resources', the 'External data sources' item is selected. A context menu is open over this item, with 'New SQL script' highlighted. Other options in the menu include 'New external data source' and 'Refresh'. The main pane displays a placeholder message: 'Select an item' with the sub-instruction 'Use the resource explorer to select or create a new item'. There are icons for databases and external tables.

Click “New SQL script”, and then “New external data source” in the dropdown.



The screenshot shows the Microsoft Azure Synapse Analytics workspace Data blade. The SQL script editor is open, displaying the following T-SQL code:

```
1 CREATE EXTERNAL DATA SOURCE [ExternalDataSource] WITH
2 (
3     LOCATION = 'https://practicumdl1.blob.core.windows.net/practicumdlc'
4 )
5
```

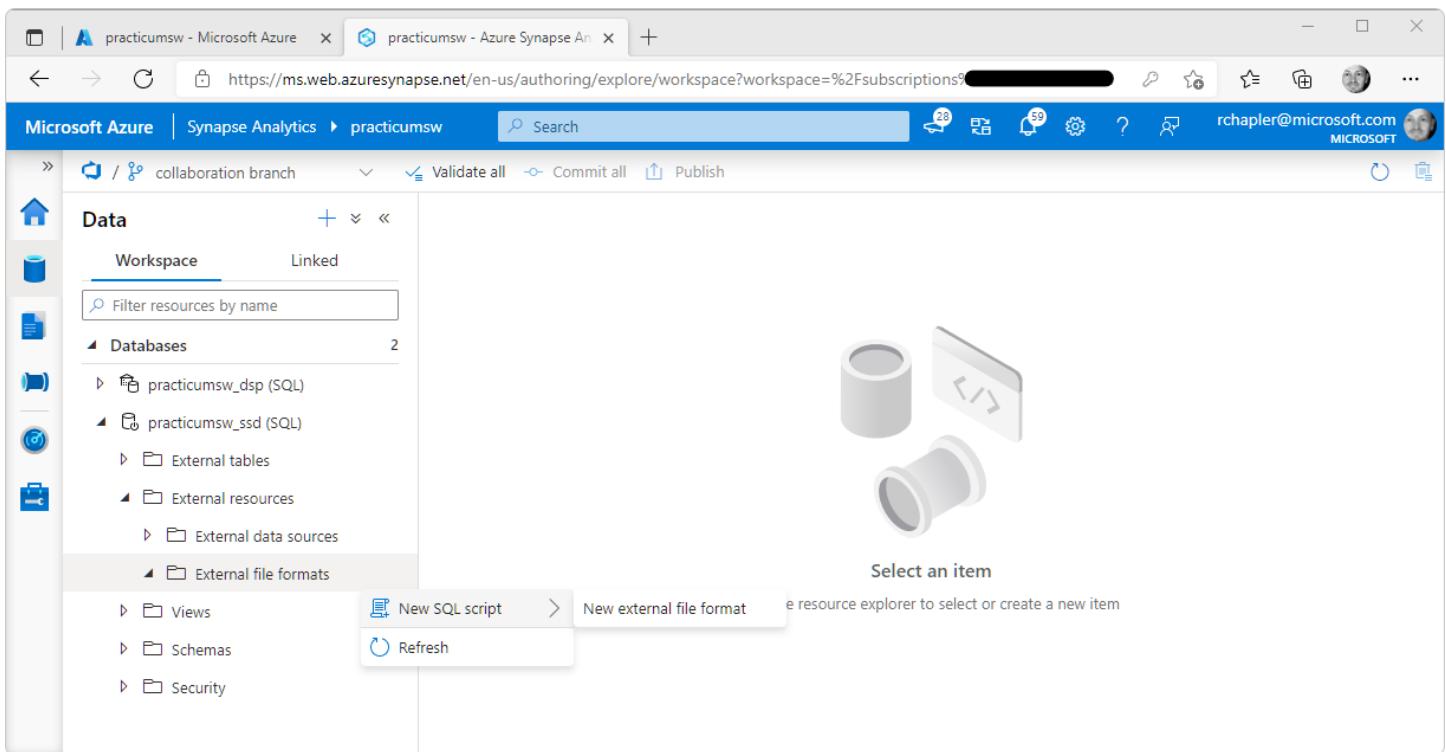
The results pane shows the execution log:

- 9:29:07 AM Started executing query at Line 1
- Total execution time: 00:00:03.449
- 00:00:03 Query executed successfully.

Replace <STORAGEACCOUNT> and <CONTAINER> in the auto-generated code and then Run.

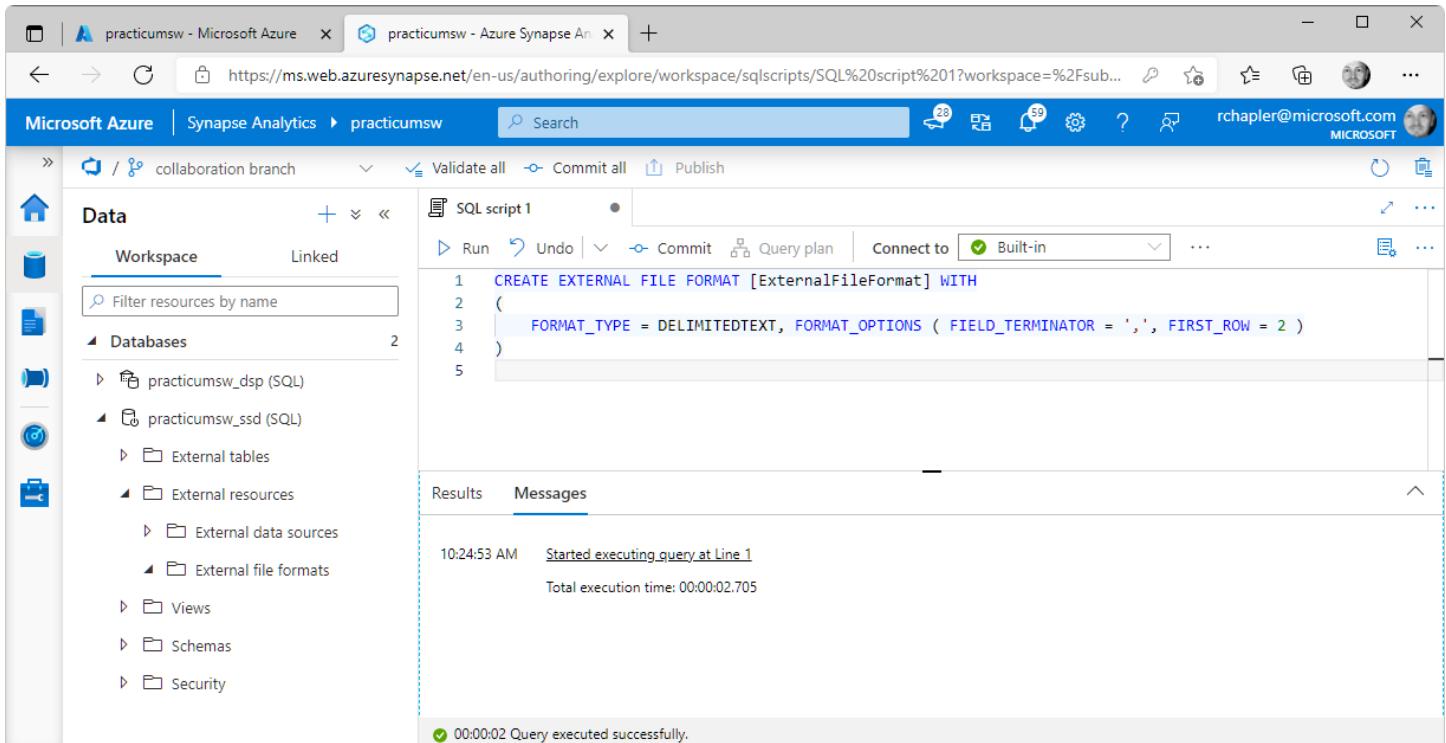
## External File Format

Expand navigation to and then right-click on “External file formats”.



The screenshot shows the Microsoft Azure Synapse Analytics workspace Data blade. The 'External resources' section is expanded, showing 'External data sources' and 'External file formats'. A context menu is open over 'External file formats', with 'New SQL script' highlighted.

Click “New SQL script”, and then “New external file format” in the dropdown.

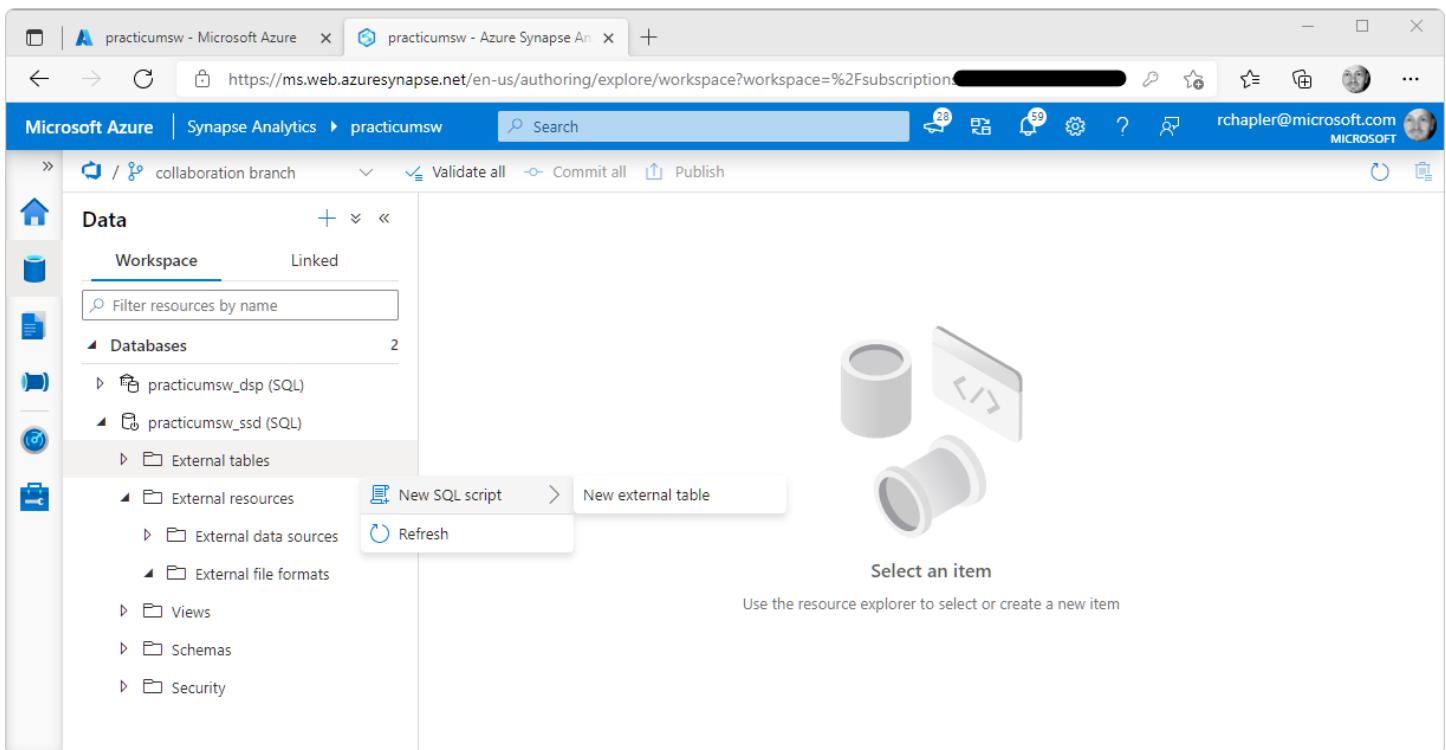


The screenshot shows the Microsoft Azure Synapse Analytics workspace Data blade. The 'External file formats' section is selected. A SQL script editor window is open, showing the creation of an external file format named 'ExternalFileFormat' with FORMAT\_TYPE set to DELIMITEDTEXT and FORMAT\_OPTIONS including FIELD\_TERMINATOR and FIRST\_ROW. The query runs successfully.

Append FORMAT\_OPTIONS arguments FIELD\_TERMINATOR and FIRST\_ROW to the default FORMAT\_TYPE argument and then Run.

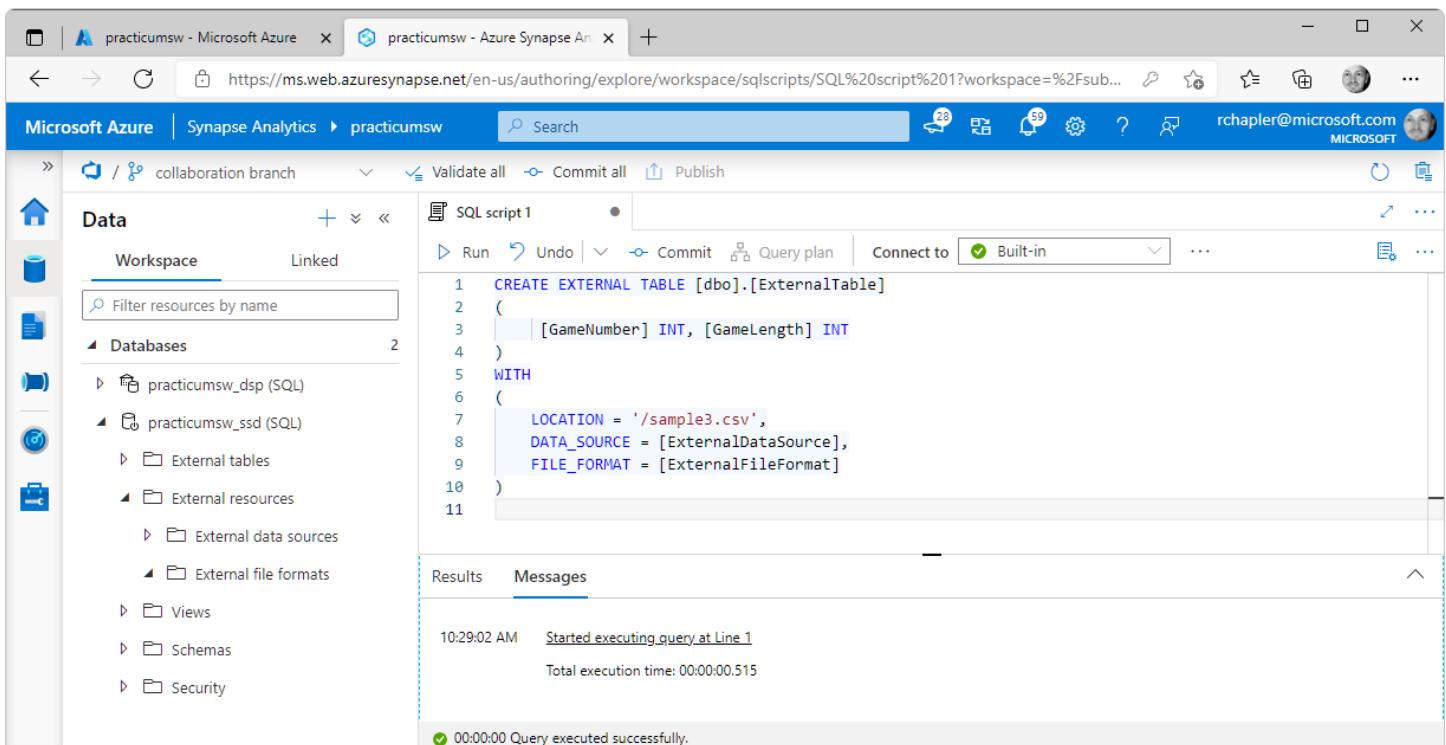
## External Table

Expand navigation to and then right-click on “External tables”.



The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. On the left, the 'Data' blade is open, showing a tree view of resources under 'Workspace'. The 'External tables' node is currently selected. A context menu is open at this node, with 'New SQL script' highlighted. Other options in the menu include 'New external table', 'Refresh', and a separator line. To the right of the menu, there is a placeholder area with icons for databases and tables, and a message saying 'Select an item'.

Click “New SQL script”, and then “New external table” in the dropdown.



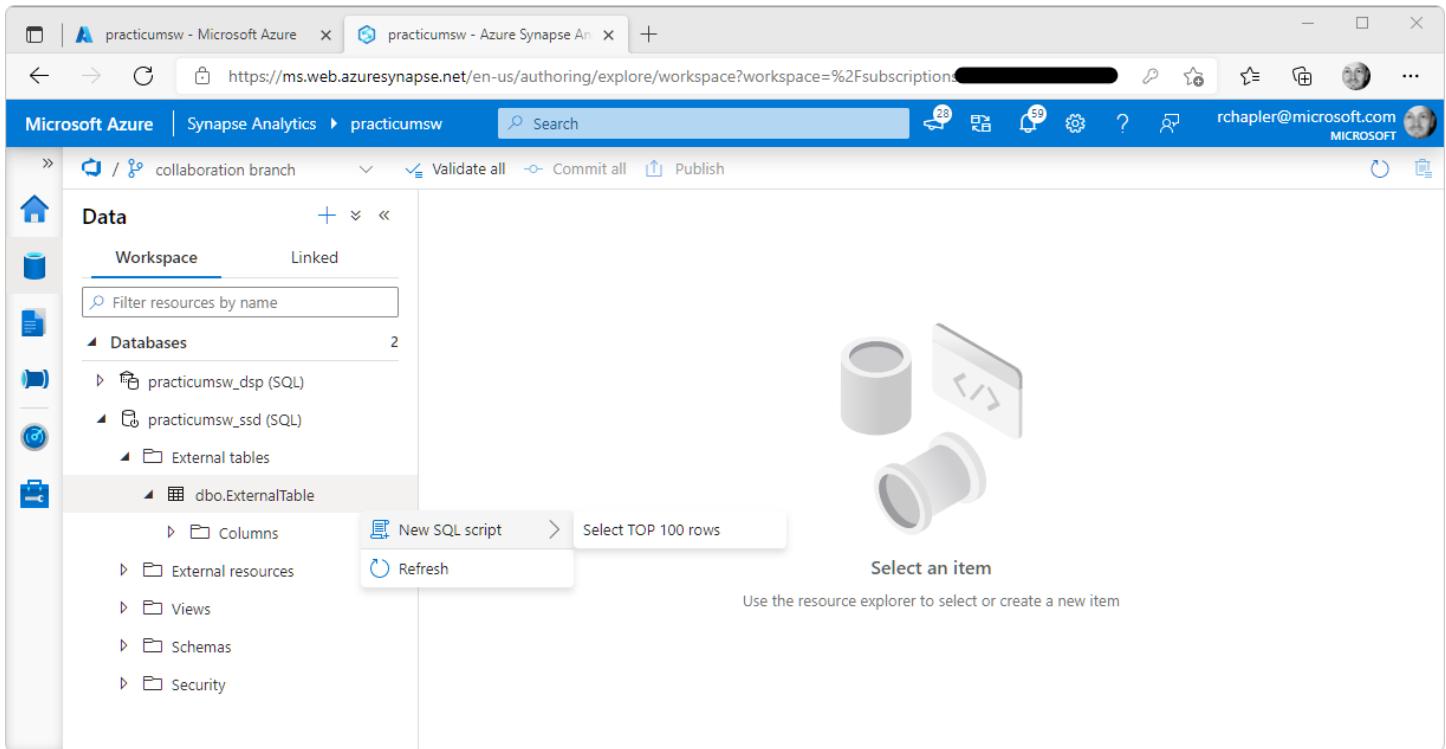
The screenshot shows the Microsoft Azure Synapse Analytics workspace interface with the 'SQL script' editor open. The editor displays a CREATE EXTERNAL TABLE T-SQL script. The script defines an external table named 'ExternalTable' with columns 'GameNumber' and 'GameLength'. It uses an external file format named 'ExternalFileFormat' and an external data source named 'ExternalDataSource'. The script is set to run against the 'Built-in' database. Below the editor, the 'Messages' tab shows the execution status: 'Started executing query at Line 1' and 'Total execution time: 00:00:00.515'. At the bottom, a success message is displayed: '00:00:00 Query executed successfully.'

Update the following items in the default code and then Run.

- **Schema Definition** ... replace `[Id] INT` with columns matching the external data source
- **LOCATION** ... replace `'/folder/file'` with values matching those in your container
- **DATA\_SOURCE** ... replace `[DataSource1]` with the name used in [Create External Data Source](#)
- **FILE\_FORMAT** ... replace `[FileFormat1]` with the name used in [Create External File Format](#)

## Confirm Success

Right-click on “External tables” and click Refresh.



Right-click on “dbo.ExternalTable”, click on “New SQL script”, and finally click on “Select TOP 100 rows” in the dropdown menu.

The screenshot shows the Microsoft Azure Synapse Analytics Data blade with a SQL script editor open. The script contains the following T-SQL code:

```
1 SELECT TOP (100) [GameNumber]
2 ,[GameLength]
3 FROM [dbo].[ExternalTable]
```

The 'Results' tab is selected, showing a table with two columns: 'GameNumber' and 'GameLength'. The data is as follows:

GameNumber	GameLength
1	30
2	29
3	31

At the bottom of the results pane, a message indicates: '00:00:01 Query executed successfully.'

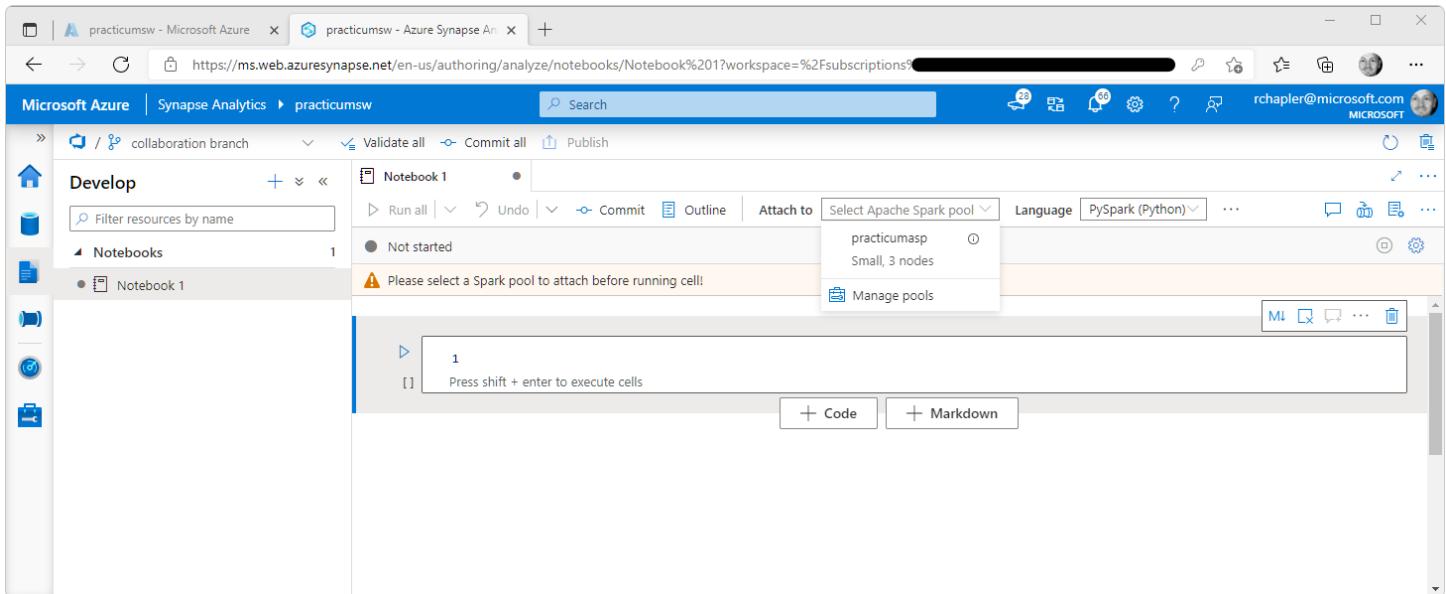
Click Run.

## Method 3: Spark Notebook (Python)

Click the **Develop** tab.

Click the + icon just above and to the right of the “**Filter resources by name**” input.

Select **Notebook** from the resulting dropdown.



Select your Apache Spark Pool from the “**Attach to**” dropdown.

Paste the following code to Cell 1 and then click “**Run Cell**”:

```
%pySpark  
theData = spark.read.load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/sample1.csv', format='csv', header=True)  
theData.show(10)  
  
print('Converted to Pandas...')  
print(theData.toPandas())
```

The screenshot shows a Microsoft Azure Synapse Analytics notebook titled "Notebook 1". The notebook contains the following PySpark code:

```
1 %%pyspark
2 theData = spark.read.load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/sample1.csv', format='csv', header=True)
3 theData.show(10)
4
5 print('Converted to Pandas...')
6 print(theData.toPandas())
7
```

The output of the code is displayed in a table:

Month	Average	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
May	0.1	0	0	1	1	0	0	0	2	0	0	0
Jun	0.5	2	1	1	0	0	1	1	2	2	0	1
Jul	0.7	5	1	1	2	0	1	3	0	2	2	1
Aug	2.3	6	3	2	4	4	4	7	8	2	2	3
Sep	3.5	6	4	7	4	2	8	5	2	5	2	5
Oct	2.0	8	0	1	3	2	5	1	5	2	3	0
Nov	0.5	3	0	0	1	1	0	1	0	1	0	1
Dec	0.0	1	0	1	0	0	0	0	0	0	0	1

The output also includes the message "Converted to Pandas..." followed by a Pandas DataFrame representation:

```
Month  Average  2005  2006  ...  2012  2013  2014  2015
0 May      0.1    0    0   ...   2    0    0    0
1 Jun      0.5    2    1   ...   2    2    0    1
2 Jul      0.7    5    1   ...   0    2    2    1
3 Aug      2.3    6    3   ...   8    2    2    3
4 Sep      3.5    6    4   ...   2    5    2    5
5 Oct      2.0    8    0   ...   5    2    3    0
6 Nov      0.5    3    0   ...   0    1    0    1
7 Dec      0.0    1    0   ...   0    0    0    1
```

[8 rows x 13 columns]

At the bottom of the notebook interface, there are two buttons: "+ Code" and "+ Markdown".

Objective Complete... congratulations!

## Objective - Synapse + Data Lake | Simple Data Ingestion

*Note: These instructions also apply (with minor differences) to Azure Data Factory.*

Follow the instructions in this section to load data from Data Lake to Synapse.

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Data Lake (with container and sample data)
- Synapse (with Dedicated SQL Pool)

### Step 2: Use “Copy Data tool”

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “Open Synapse Studio” rectangle.

Click the **Ingest** button.

#### Properties

The screenshot shows the Microsoft Azure Synapse Analytics Copy Data tool interface. The left sidebar lists steps: 1 Properties (selected), 2 Source, 3 Target, 4 Settings, and 5 Review and finish. The main area is titled 'Properties' and describes using the Copy Data Tool to perform a one-time or scheduled data load from 90+ data sources. It highlights the wizard experience for specifying loading settings and generating artifacts like pipelines, datasets, and linked services. Two task types are shown: 'Built-in copy task' (selected) and 'Metadata-driven copy task (Preview)'. The 'Built-in copy task' section indicates it will generate a single pipeline for copying data from 90+ sources. The 'Metadata-driven copy task (Preview)' section notes it requires metadata stored in external control tables for large-scale loads. Task cadence options are 'Run once now' (selected), 'Schedule', and 'Tumbling window'. Navigation buttons at the bottom include '< Previous', 'Next >', and 'Cancel'.

Confirm default select, “Built-in copy task” and “Run once now”, then click the “**Next >**” button.

## Source > Dataset

The screenshot shows the "Copy Data tool" interface in the Azure Synapse Analytics workspace. The left sidebar lists steps: Properties (selected), Source, Dataset, Configuration, Target, Settings, and Review and finish. The main panel is titled "Source data store" and contains the following fields:

- Source type:** Azure Data Lake Storage Gen2
- Connection:** practicumdl
- Integration runtime:** AutoResolveIntegrationRuntime
- File or folder:** practicumdlc/sample1.csv (with a "Browse" button)
- Options:** Binary copy (unchecked), Recursively (checked), Enable partition discovery (unchecked)
- Max concurrent connections:** (empty input field)
- Filter by last modified:** Start time (UTC) and End time (UTC) input fields

At the bottom are "Previous" and "Next >" buttons, and a "Cancel" button.

On the “Source data store” page, enter values for the following items:

**Source Type** Select “Azure Data Lake Storage Gen2”

**Connection** Click the “+ New connection” button

The screenshot shows the Microsoft Azure Synapse Analytics Copy Data tool interface. On the left, a navigation pane lists steps: Properties (selected), Source, Dataset, Configuration, Target, Settings, and Review and finish. The main area is titled "New connection (Azure Data Lake Storage Gen2)". It includes fields for Name (practicumdl), Description, Connect via integration runtime (AutoResolveIntegrationRuntime), Authentication method (Managed Identity), Account selection method (From Azure subscription selected), Azure subscription (rchapler), Storage account name (practicumdl), and Test connection (To linked service selected). At the bottom right, there are "Connection successful" and "Test connection" buttons.

On the “New connection (Azure Data Lake Storage Gen2)” page, enter values for the following items:

<b>Name</b>	Enter a meaningful name aligned with standards
<b>Connect via...</b>	Confirm default selection, <b>AutoResolveIntegrationRuntime</b>
<b>Account Selection Method</b>	Confirm default selection, “ <b>From Azure subscription</b> ”
<b>Azure Subscription</b>	Select your subscription
<b>Storage Account Name</b>	Select your instance of Data Lake

Click “**Test connection**” to confirm successful connection and then click the **Create** (or **Commit**) button.

Back on the “Source data store” page, click **Browse** to select container/file {e.g., “practicumdlc/sample1.csv”}.

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “**Next >**” button.

## Source > Configuration

The screenshot shows the 'Copy Data tool' interface in Microsoft Azure Synapse Analytics. The left sidebar lists five steps: Properties (checked), Source (selected), Dataset, Configuration, Target, Settings, and Review and finish. The main area is titled 'File format settings' and contains the following configuration:

- File format**: Text format (dropdown), Detect text format, Preview data button.
- Column delimiter**: Comma (,) (dropdown), Edit checkbox.
- Row delimiter**: Line feed (\n) (dropdown), Edit checkbox.
- First row as header**: Checked checkbox.
- Advanced**: Compression type (None dropdown).
- Additional columns**: New button.

At the bottom are 'Previous' and 'Next >' buttons, and a 'Cancel' button on the right.

On the “**File format settings**” page, confirm default configuration and the resulting data preview.

Make changes if required {e.g., in this example, I check “**First row as header**” to correct handling of incoming data}.

Click the “**Next >**” button.

## Target > Dataset

The screenshot shows the 'Copy Data tool' interface in the Microsoft Azure portal. The left sidebar lists steps: Properties (checked), Source (checked), Target (selected), Dataset, Configuration, Settings, and Review and finish. The main panel is titled 'Destination data store' and contains the following fields:

- Target type:** Azure Synapse Analytics (dropdown menu)
- Connection \***: Select... (dropdown menu) with a '+ New connection' button next to it.

At the bottom are 'Previous' and 'Next >' buttons, and a 'Cancel' button on the right.

On the “**Destination data store**” page, enter values for the following items:

**Source Type**      Select “**Azure Synapse Analytics**”

**Connection**      Click the “**+ New connection**” button

practicumsw - Microsoft Azure x practicumsw - Azure Synapse An x + <https://ms.web.azuresynthesize.net/en-us/authoring/orchestrate/pipeline/Pipeline%201?...> 28 4 ⚡ 🌐 ? 🔍 🗃 🏷️ 🎯 🏹 rchapler@microsoft.com MICROSOFT

Microsoft Azure | Synapse Analytics > practicum Search

Copy Data tool

Properties  
Source  
Target  
Dataset  
Configuration  
Settings  
Review and finish

Destination data store  
Specify the destination data store.  
Target type  
Connection \*

New connection (Azure Synapse Analytics)

Choose a name for your linked service. This name cannot be updated later.

Name \*

Description

Connect via integration runtime \*  [Edit](#)

Connection string [Azure Key Vault](#)

Account selection method [?](#)  
 From Azure subscription  Enter manually

Azure subscription

Server name \*

Database name \*

SQL pool \*  [Edit](#)

Authentication type \*

User name \*

Add dynamic content [Alt+Shift+D]

Password [Azure Key Vault](#)

AKV linked service \* [?](#)  [Edit](#)

Secret name \*

Secret version [?](#)

Additional connection properties [New](#)

Annotations [New](#)

Parameters [Advanced](#)

Previous Next Commit Cancel [Test connection](#)

On the “New connection (Azure Synapse Analytics)” popout, enter values for the following items:

Name	Enter a meaningful name aligned with standards
Connect via...	Confirm default selection, <b>AutoResolveIntegrationRuntime</b>
Account Selection Method	Confirm default selection, “From Azure subscription”
Azure Subscription	Select your subscription
Server Name	Select your instance of Synapse Workspace
Database Name	Select your Dedicated SQL Pool
Authentication Type	Confirm default value, “SQL authentication”
User Name	Enter the “SQL Server admin login” value used during instantiation of Synapse
AKV Linked Service	Select the name of the Linked Service created for the Key Vault
Secret Name	Enter the Secret Name used to capture the Synapse Workspace administrator password

Click “Test connection” to confirm successful connection and then click the **Create** (or **Commit**) button.

The screenshot shows the Microsoft Azure Copy Data tool interface. On the left, a vertical navigation pane lists steps: Properties (checked), Source (checked), Target (selected), Dataset, Configuration, Settings, and Review and finish. The main panel is titled "Destination data store". It asks to specify the destination data store, using an existing connection or creating a new one. The "Target type" is set to "Azure Synapse Analytics", the "Connection" is "practicumsa", and the "Integration runtime" is "AutoResolveIntegrationRuntime". Below this, a table maps a source table ("Azure Data Lake Storage Gen2 file") to a target table ("dbo.Sample1"). A checkbox for "Skip column mapping for all tables" is present. At the bottom are "Previous", "Next >", and "Cancel" buttons.

Back on the “Destination data source” page, enter values for target schema name {e.g., “dbo”} and target table name {e.g., “Sample1”}.

Click the “Next >” button.

## Target > Configuration

The screenshot shows the 'Copy Data tool' configuration interface in the Microsoft Azure portal. The left sidebar lists steps: Properties, Source, Target (selected), Dataset, Configuration, Settings, and Review and finish. The main area shows 'Table mappings (1)' for 'Source' (Azure Data Lake Storage Gen2 file) and 'Target' (dbo.Sample1). A 'Column mapping' table lists 12 columns from the source, all mapped to 'Month' in the target, maintaining their original string values. Below this is the 'Azure Synapse Analytics sink properties' section, which includes a 'Pre-copy script' field and an 'Advanced' link. Navigation buttons at the bottom include '< Previous', 'Next >', and 'Cancel'.

Source	Type	Destination	Type
Month	String	Month	String
"Average"	String	"Average"	String
"2005"	String	"2005"	String
"2006"	String	"2006"	String
"2007"	String	"2007"	String
"2008"	String	"2008"	String
"2009"	String	"2009"	String
"2010"	String	"2010"	String
"2011"	String	"2011"	String
"2012"	String	"2012"	String
"2013"	String	"2013"	String
"2014"	String	"2014"	String
"2015"	String	"2015"	String

Review default column mappings; correct if required.

Click the “**Next >**” button.

## Settings

The screenshot shows the 'Copy Data tool' settings page in the Microsoft Azure portal. The left sidebar lists steps: Properties (checked), Source (checked), Target (checked), Settings (selected), and Review and finish. The main area is titled 'Settings' and contains the following fields:

- Task name \***: CopyPipeline\_8th
- Task description**: (empty text area)
- Data consistency verification**: (checkbox is off)
- Fault tolerance**: (dropdown menu)
- Enable logging**: (checkbox is off)
- Enable staging**: (checkbox is off)
- Advanced** section:
  - Copy method**: Bulk insert (radio button selected)
  - Bulk insert table lock**: No (radio button selected)
- Data integration unit**: Auto (dropdown menu)
- Degree of copy parallelism**: (dropdown menu)

At the bottom are buttons for < Previous, Next >, and Cancel.

On the **Settings** page, enter values for the following items:

Enable Staging	Unchecked
Copy Method	Bulk insert

Click the “**Next >**” button.

## Summary

Screenshot of the Microsoft Azure Copy Data tool interface for creating a pipeline from Azure Data Lake Storage Gen2 to Azure Synapse Analytics.

The left sidebar shows the pipeline steps: Properties (checked), Source (checked), Target (checked), Settings (checked), Review and finish (selected), Review (unchecked), and Deployment (unchecked).

The main area displays the "Summary" step, indicating the pipeline will copy data from Azure Data Lake Storage Gen2 to Azure Synapse Analytics.

**Properties**

- Task name: CopyPipeline\_8th
- Task description: (empty)

**Source**

- Connection name: practicumdl
- Dataset name: SourceDataset\_8th
- Column delimiter: ,
- Row delimiter:
- Escape character: \
- Quote char: "
- First row as header: true
- File name: sample1.csv

**Target**

- Connection name: practicumsa
- Dataset name: DestinationDataset\_8th
- Table name: dbo.Sample1

**Copy settings**

- Timeout: 7.00:00:00
- Retry: 0
- Retry interval: 30
- Secure output: false
- Secure input: false

Buttons at the bottom: < Previous, Next >, Cancel.

Review configuration and then click the “**Next >**” button.

## Deployment

The screenshot shows the Microsoft Azure Copy Data tool interface. On the left, a vertical navigation bar lists steps: Properties, Source, Target, Settings, Review and finish, Review, and Deployment. The first four steps have a checkmark next to them. In the center, there's a diagram showing an arrow pointing from 'Azure Data Lake Storage Gen2' to 'Azure Synapse Analytics'. Below the diagram, the text 'Deployment complete' is displayed. To the right of the diagram, a table titled 'Deployment step' shows two rows: 'Creating datasets' and 'Creating pipelines', both with a status of 'Succeeded' and a green checkmark. At the bottom of the interface are three buttons: 'Finish', 'Edit pipeline', and 'Monitor'.

Monitor progress.

Click the **Finish** button on success.

## Step 3: Confirm Success

"CopyPipeline..." will be added to the list of **Pipelines** in the Synapse, **Integrate** section.

The screenshot shows the Microsoft Azure Synapse Analytics 'Integrate' section. On the left, a sidebar shows a 'collaboration branch' and lists 'Pipelines' with one item named 'CopyPipeline\_8th'. The main area displays the details of the 'CopyPipeline\_8th' pipeline. It shows the 'Activities' list with a single 'Copy data' activity named 'Copy\_8th'. Below the activities, there are tabs for 'Parameters', 'Variables', 'Settings', and 'Output'. The 'Output' tab is selected, showing a table with one row: Pipeline run ID: a084b7d3-1d69-4993-aad8-2f654c8962b0, Name: Copy\_8th, Type: Copy data, Run start: 2021-09-22T17:38:30.320, Duration: 00:00:10, Status: Succeeded, and Integration runtime: DefaultIntegrationRuntime. A green checkmark indicates success. At the bottom right, there are links for 'View debug run consumption' and 'View pipeline run history'.

Click **Debug** and confirm successful execution.

Click the **Data** icon in the navigation.

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar has a 'Data' icon selected. The main area displays a SQL script named 'SQL script 5' with the following code:

```
1 SELECT TOP (100) [Month]
2 , [ "Average"]
3 , [ "2005"]
```

The results pane shows a table with data for years 2006 through 2011. The table has columns for "2006", "2007", "2008", "2009", "2010", and "2011". The data is as follows:

	"2006"	"2007"	"2008"	"2009"	"2010"	"2011"
0	1	1	0	0	0	0
1	1	0	0	1	1	1
2	1	1	2	0	1	3
3	2	4	4	4	4	7

A message at the bottom of the results pane says "00:00:00 Query executed successfully."

Expand **Databases** > {Dedicated SQL Pool} > **Tables** and confirm that you can see your new table.

Right-click on your table, select “New SQL script”, and then “Select TOP 100 rows” in the dropdown menu.

*Objective Complete... congratulations!*

# Objective - Synapse + SQL + Data Explorer | Simple Data Pipeline

*Note: These instructions also apply (with minor differences) to Azure Data Factory.*

Follow the instructions in this section to load data from Data Explorer to Synapse.

## Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Data Explorer (with cluster, database, and sample Product table)
- SQL (with sample data)
- Synapse (with linked services)

## Step 2: Confirm Linked Services

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Select the **Manage** icon on the navigation and then “**Linked Services**” in the resulting navigation menu.

Confirm linked service instantiation for: 1) Key Vault, 2) SQL and 3) Data Explorer.

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar has a 'collab branch' selected. Under 'Linked services', there is a list of instantiated items:

Name	Type	Related	Annotations
practicumded	Azure Data Explorer (Kusto)	0	
practicumkv	Azure Key Vault	1	
practicumsd	Azure SQL Database	0	
practicumsw-WorkspaceDefaultSqlServer	Azure Synapse Analytics	0	
practicumsw-WorkspaceDefaultStorage	Azure Data Lake Storage Gen2	0	

*Note: you are likely to see system-generated linked services {e.g., “practicumsw-WorkspaceDefault...”} in addition to instantiated items.*

## Step 3: Create Pipeline

Click the **Integrate** icon in the navigation.

No items to show  
Try creating a new item using the + button above. [Learn more](#)

Select an item  
Use the resource explorer to select or create a new item

Click the + icon just above and to the right of the “**Filter resources by name**” input.

Select **Pipeline** from the resulting dropdown.

Pipeline 1

Activities

Source dataset \*

Select... New

Add a “**Copy data**” activity.

## Source

On the **Source** tab, click the “**+ New**” button.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline creation interface. On the left, the navigation pane is visible with 'Integrate' selected. Under 'Activities', 'Copy data' is highlighted. A search bar at the top right contains the text 'azure sql'. The main area displays two options under the 'All' tab: 'Azure SQL Database' and 'Azure SQL Database Managed Instance'. Both options feature a blue cylinder icon with the word 'SQL' on it. Below each icon is its name and a 'Continue' button. The 'Continue' button is highlighted with a blue border.

On the “**New integration dataset**” popout, search for and then select “**Azure SQL Database**”.

Click the **Continue** button.

The screenshot shows the Microsoft Azure Synapse Analytics interface. On the left, there's a sidebar with icons for Home, Pipelines, Databricks, Data Lake Analytics, Batch Service, Functions, Data Explorer, Copy Data, Data Flow, Move & Transform, Synapse, and Pipelines. The Pipelines section is expanded, showing 'Pipeline 1'. The main area is titled 'Set properties' for 'Pipeline 1'. It has fields for 'Name' (set to 'AzureSqlTable1'), 'Linked service' (set to 'practicumsd'), 'Connect via integration runtime' (set to 'AutoResolveIntegrationRuntime'), 'Table name' (set to 'SalesLT.Product'), and 'Import schema' (radio button selected for 'From connection/store'). At the bottom are 'OK', 'Back', and 'Cancel' buttons.

On the “Set properties” popout, enter values for the following items:

Name	Enter a meaningful name aligned with standards
Linked Service	Select your Azure SQL linked service
Table Name	Select “SalesLT.Product”
Import Schema	Confirm default selection, “From connection/store”

Leave all other settings with default values.

Click the **OK** button.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor interface. On the left, there's a sidebar with icons for Home, Pipelines, Databricks, Data Lake Analytics, General, HDInsight, Iteration & conditionals, and Machine Learning. The main area shows a pipeline named "Pipeline 1" with one activity: "Copy data1". The "Source" tab is selected, showing the configuration for the source dataset "AzureSqlTable1". Other tabs include "Sink", "Mapping", "Settings", and "User properties". At the bottom, there's a note: "Please preview data to validate the partition settings are correct before you trigger a run or publish the pipeline." A "New" button is also visible.

## Sink

Click on the **Sink** tab and then click the “**+ New**” button.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline creation interface. On the left, there's a sidebar with icons for Home, Pipelines, Databricks, Data Lake Analytics, and General. The main area shows a pipeline named 'Pipeline 1'. A search bar at the top right has 'Search' and a user profile 'rchapler@microsoft.com MICROSOFT'. Below the search bar, a button labeled 'New integration dataset' is visible. The central part of the screen displays a 'Select a data store' dialog with a search bar containing 'data explorer'. Below it, a list of data stores is shown, with 'Azure Data Explorer (Kusto)' highlighted. At the bottom of the dialog are 'Continue' and 'Cancel' buttons.

On the “**New integration dataset**” popout, search for and then select “**Azure Data Explorer (Kusto)**”.

Click the **Continue** button.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline creation interface. The left sidebar shows a pipeline named 'Pipeline 1'. The main area displays a 'Set properties' dialog for a dataset. The dialog fields include: 'Name' set to 'AzureDataExplorerTable1', 'Linked service' set to 'practicumded', 'Connect via integration runtime' set to 'AutoResolveIntegrationRuntime', and 'Table' set to 'None'. There are also 'Edit' and 'Advanced' options. At the bottom are 'OK', 'Back', and 'Cancel' buttons.

On the “**Set properties**” popout, enter values for the following items:

<b>Name</b>	Enter a meaningful name aligned with standards
<b>Linked Service</b>	Select your Azure Data Explorer linked service
<b>Table Name</b>	Select <b>Product</b>

Leave all other settings with default values.

Click the **OK** button.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor interface. On the left, the navigation pane is visible with sections like 'Integrate', 'Pipelines', and 'Pipeline 1'. The main workspace displays a pipeline named 'Pipeline 1' containing a single activity: 'Copy data'. This activity is highlighted with a red circle. Below the activity, there are tabs for 'General', 'Source', 'Sink', 'Mapping', 'Settings', and 'User properties'. The 'Sink' tab is selected. Under the 'Sink' tab, the 'Sink dataset' dropdown is set to 'AzureDataExplorerTable1'. Other fields include 'Database' (set to 'practicumded') and 'Ingestion mapping name' (empty). There are also buttons for 'Open', 'New', and 'Learn more'.

## Mapping

Click on the **Mapping** tab and then click the “Import schemas” button.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor. On the left, the navigation pane shows 'Integrate' and 'Pipelines'. Under 'Pipelines', 'Pipeline 1' is selected. The main area displays the 'Activities' section with a single 'Copy data' activity named 'Copy data1'. The 'Mapping' tab is active, showing type conversion settings for two columns:

Source	Type	Destination	Type
ProductNumber	nvarchar	ProductNumber	String
ListPrice	money	ListPrice	Decimal

Precision: 19 Scale: 4

## Step 4: Confirm Success

Click **Validate** to confirm that there are no errors, and then click **Debug** to confirm that the pipeline runs successfully.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor. The 'Output' tab is selected for Pipeline 1. It displays a successful pipeline run with the ID 77b27627-6862-4dea-87ba-da3a3e9586a5. The run started at 2021-10-04T20:59:21.848 and completed at 00:00:15 with a status of 'Succeeded' in the 'DefaultIntegrationRuntime'.

Name	Type	Run start	Duration	Status	Integration runtime
Copy data1	Copy data	2021-10-04T20:59:21.848	00:00:15	Succeeded	DefaultIntegrationRuntime

Objective Complete... congratulations!

# Objective - Synapse | Simple Conditional Pipeline

*Note: These instructions also apply (with minor differences) to Azure Data Factory.*

Follow the instructions in this section to **demonstrate conditional logic in a pipeline**.

## Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Synapse

## Step 2: Create Pipeline

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Click the **Integrate** icon in the navigation.

Click the **+** icon just above and to the right of the “**Filter resources by name**” input.

Select **Pipeline** from the resulting dropdown.

The screenshot shows the Microsoft Azure Synapse Analytics studio interface. The left sidebar has an 'Integrate' icon selected. Under 'Pipelines', there is one item named 'Pipeline 1'. The main area is titled 'Pipeline 1' and shows the 'Activities' tab. A sidebar on the right lists various activity types: Synapse, Move & transform, Azure Data Explorer, Azure Function, Batch Service, Databricks, Data Lake Analytics, General, HDInsight, Iteration & conditionals, and Machine Learning. Below the activities is a 'Variables' tab, which is currently active. It contains a table with two rows:

Name	Type	Default value
Flag	Boolean	Value
Result	String	Value

Click on the **Variables** tab.

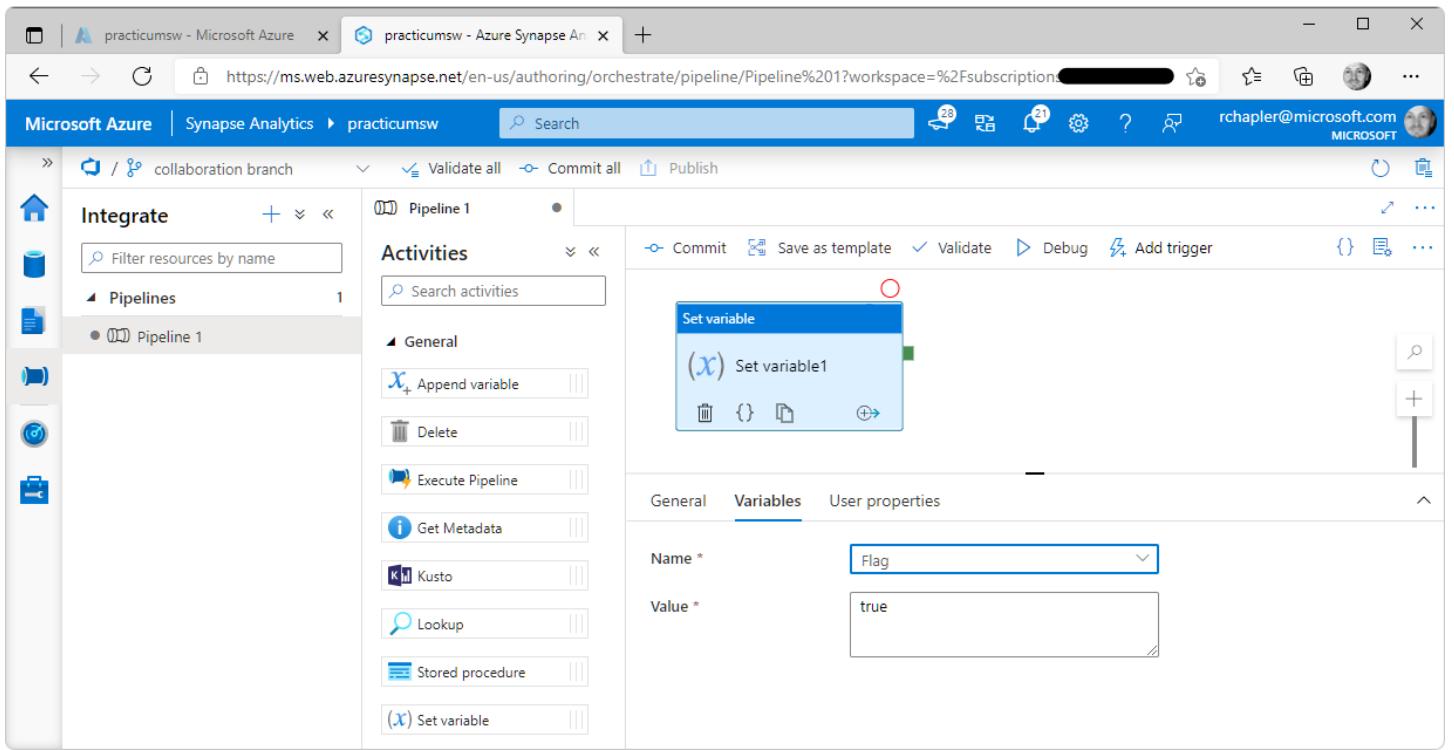
Click “**+ New**” to add each of the following variables:

**Flag** (Boolean) Will serve as a TRUE / FALSE trigger for the conditional result

**Result** (String) For capturing an anecdotal result {e.g., “Success!”}

Activity 1, Set Variable

Expand **General** in the **Activities** bar, and then drag-and-drop a “**Set variable**” component into the activity window.



On the **Variables** tab of the “**Set Variable...**” component, enter values for the following items:

Name	Select <b>Flag</b> from the dropdown
Value	Enter <b>true</b>

## Activity 2, If Condition

Expand “**Iteration & conditionals**” in the **Activities** bar.

Drag-and-drop an “**If Condition**” component into the pipeline.

The screenshot shows the Microsoft Azure Synapse Analytics Pipeline Editor. On the left, the navigation pane shows 'Integrate' selected, with 'Pipelines' expanded and 'Pipeline 1' selected. The main workspace displays a pipeline diagram. A 'Set variable' activity is connected to an 'If Condition' activity. The 'If Condition' activity has two branches: 'True' and 'False', both leading to 'No activities'. Below the activities, the 'Activities (0)' tab is selected. In the 'Expression' field, the expression '@variables('Flag')' is entered. The 'General' tab is also visible.

Create a dependency between the “**Set variable**” and “**If Condition**” components.

Click on the “**Activities (0)**” tab.

Click into the **Expression** textbox, and then enter the following: `@variables('Flag')`

Click on the pencil icon in the “**If Condition**” component, **True** selection.

The screenshot shows the Microsoft Azure Synapse Analytics Pipeline Editor. The left navigation pane shows 'Integrate' selected, with 'Pipelines' expanded and 'Pipeline 1' selected. The main workspace displays a pipeline diagram. A 'Set variable' activity is shown with its properties: 'Name' is 'Result' and 'Value' is '"Success!"'. This 'Set variable' activity is connected to an 'If Condition' activity. The 'If Condition' activity has two branches: 'True' and 'False', both leading to 'No activities'. The 'Variables' tab is selected for the 'Set variable' activity. The 'General' tab is also visible.

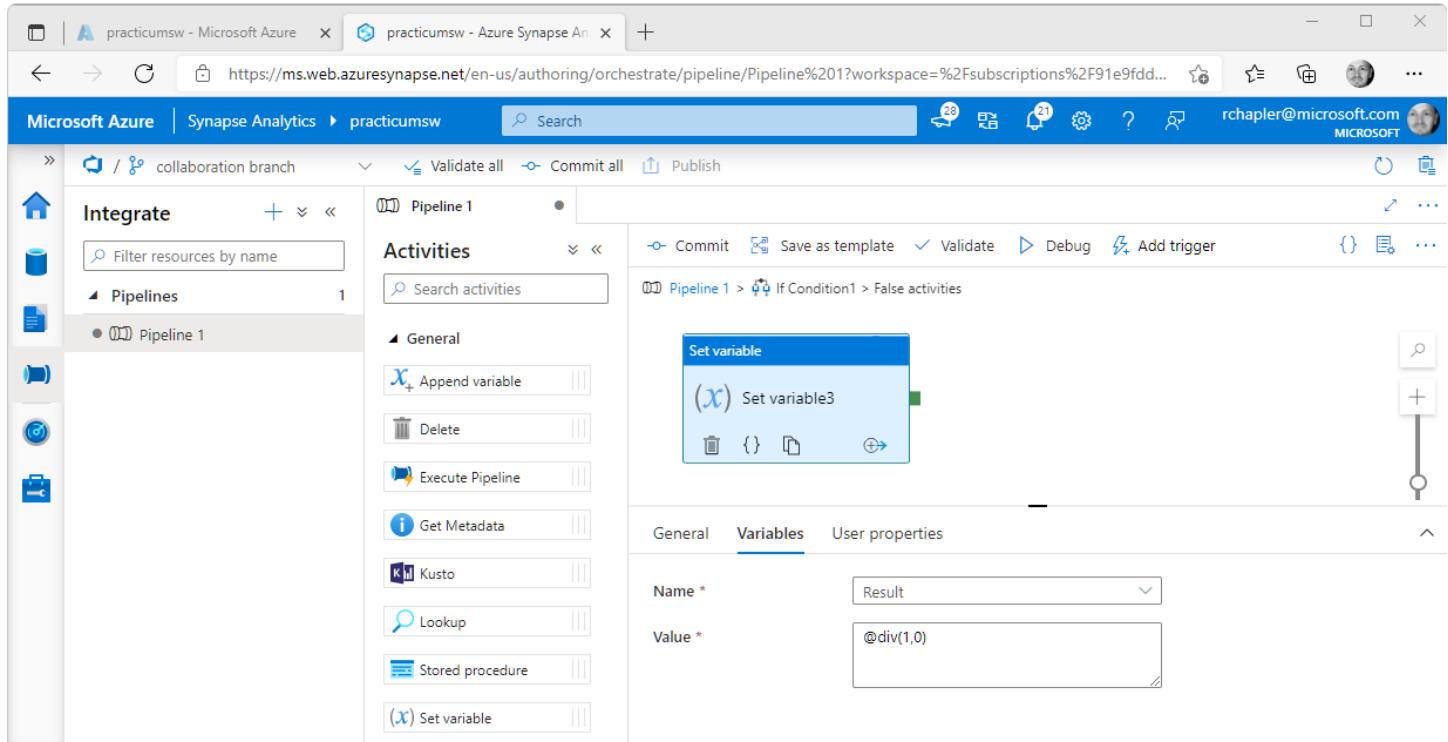
Expand **General** in the **Activities** bar, and then drag-and-drop a “Set variable” component into the activity window.

On the **Variables** tab of the “Set Variable...” component, enter values for the following items:

Name	Select <b>Result</b> from the dropdown
Value	Enter “Success!”

Use cookie crumbs to navigate back to the main window.

Click on the pencil icon in the “If Condition” component, **False** selection.



Expand **General** in the **Activities** bar, and then drag-and-drop a “Set variable” component into the activity window.

On the **Variables** tab of the “Set Variable...” component, enter values for the following items:

Name	Select <b>Result</b> from the dropdown
Value	Enter @div(1,0)... this will force an error result

### Step 3: Confirm Success

Click **Debug**.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor. On the left, the sidebar displays 'Integrate' and 'Pipelines'. Under 'Pipelines', 'Pipeline 1' is selected. The main workspace shows a pipeline flow starting with a 'Set variable' activity (labeled 'Set variable1') followed by an 'If Condition' activity. The 'If Condition' activity has two branches: 'True' (1 activities) and 'False' (No activities). Below the pipeline, the 'Output' tab is selected, showing a table of pipeline runs. The first run, 'Set variable2', is listed with a status of 'Succeeded'. The second run, 'If Condition1', is also 'Succeeded'. The third run, 'Set variable1', is also 'Succeeded'. The pipeline run ID is 08bb1dff-2b6d-4023-8ab7-750cbcff2423.

You should see **Succeeded** messages given current variable settings.

Next, we will make a pipeline change that will force pipeline failure.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor. The pipeline structure is identical to the previous screenshot, but the 'Set variable1' activity now has a red circle with a question mark icon above it, indicating an error. The 'Variables' tab is selected for the 'Set variable1' activity, showing 'Name' set to 'Flag' and 'Value' set to 'false'. The pipeline run table shows the 'Set variable1' run has failed.

Navigate to the “**Set variable1**” component, **Variables** tab.

Replace the **true** value with **false**.

Click **Debug**.

You will see **Failed** messages given current variable settings.

*Objective Complete... congratulations!*

# Objective - Synapse (pipeline) + Data Explorer | Incremental Load

Note: These instructions also apply (with minor differences) to Azure Data Factory and Azure Data Explorer.

Use Case:

- Customer XYZ wants to efficiently load data from an operational data source
  - The data source {e.g., Oracle} currently lacks a corresponding Data Flow connector, so you must use a Pipeline
- Customer XYZ wants to capture a time series of delete, insert, and update events and provide for “time travel” through data

Follow the instructions in this section to **establish incremental load using a Synapse Pipeline to move data from a SQL database to a Data Explorer database**.

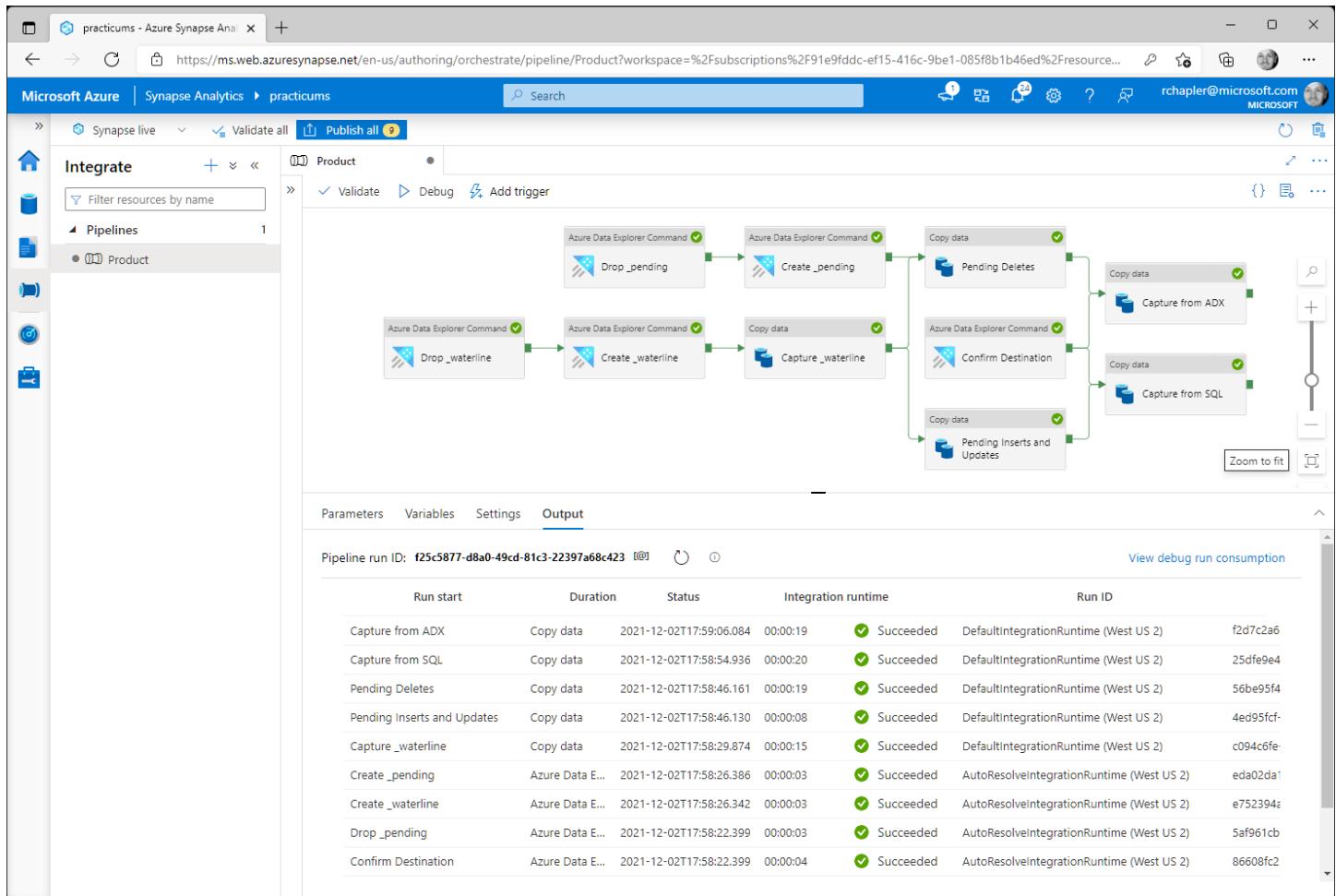
## Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- SQL (with sample database)
- Synapse with Data Explorer Pool, Data Explorer Database, and Linked Services for Data Explorer and SQL

## Step 2: Create Pipeline

When we are finished, our pipeline will function as snipped below.



Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Click the **Integrate** icon in the navigation, the **+** icon in the header and then “**Pipeline**” in the resulting dropdown.

## Ascertain Waterline

### Drop ...\_waterline

Execution of this component will **drop previously created waterline table and data** in Data Explorer.

Expand “**Azure Data Explorer**” in the **Activities** bar.

Drag-and-drop an “**Azure Data Explorer Command**” component into the activity window.

On the **Connection** tab of the new component, select the Data Explorer Linked Service from the dropdown.

On the **Command** tab of the new component, paste the following KQL in the Command textbox.

```
.drop table _product_waterline ifexists;
```

Click **Debug** and monitor result to confirm successful progress.

### Create ...\_waterline

Execution of this component will **create a new (and empty) waterline table** in Data Explorer.

Expand “**Azure Data Explorer**” in the **Activities** bar.

Drag-and-drop an “**Azure Data Explorer Command**” component into the activity window.

Create a dependency from the “**Drop Waterline**” component to the new component.

On the **Connection** tab of the new component, select the Data Explorer Linked Service from the dropdown.

On the **Command** tab of the new component, paste the following KQL in the Command textbox.

```
.create table _product_waterline ( _id:string, _dt:datetime );
```

Logic explained...

- The “**\_id**” field will be used as a string-based, generic capture of various identifier data types {e.g., int, guid, etc.} and situations where unique identification may require concatenation of multiple fields {e.g., “123-456”}
- The “**\_dt**” field will be used as a datetime-based, generic capture of waterline data that might come from various fields {e.g., “LastModified”, “ModifiedOn”, etc.}

Click **Debug** and monitor result to confirm successful progress.

### Capture ...\_waterline

Execution of this component will **localize a waterline dataset** we can use to identify deleted, inserted, or updated records.

**Capture of \_waterline, etc. data to database is necessary in Pipelines because unlike Data Flow, there is no in-flow use of data.**

Expand “**Move & Transform**” in the **Activities** bar, then drag-and-drop an “**Copy data**” component into the activity window.

Create a dependency from the “**Create Waterline**” component to the new component.

On the **Source** tab of the new component, enter values for the following items:

<b>Source Dataset</b>	Create a dataset using the <b>SQL</b> linked service and the “ <b>None</b> ” table {e.g., “ <b>practicumsd</b> ”}
<b>Use Query</b>	Select the <b>Query</b> radio button
<b>Query</b>	Paste the following T-SQL: <pre>SELECT ProductId _id,MAX(ModifiedDate) _dt FROM SalesLT.Product GROUP BY ProductId</pre>

No additional changes are required on this tab.

On the **Sink** tab of the new component, enter values for the following items:

---

<b>Sink Dataset</b>	Create a dataset using the <b>Data Explorer</b> linked service and the “ <code>_product_waterline</code> ” table {e.g., “ <code>practicumded_product_waterline</code> ”}
<b>Database and Table</b>	Confirm values

---

No additional changes are required on this tab.

On the **Mapping** tab of the new component, click the “**Import schemas**” button.

Click **Debug** and monitor result to confirm successful progress.

## Evaluate Change

### Drop ...\_pending

Execution of this component will **drop previously created “...pending” table and data** in Data Explorer.

Expand “**Azure Data Explorer**” in the **Activities** bar.

Drag-and-drop an “**Azure Data Explorer Command**” component into the activity window.

On the **Connection** tab of the new component, select the Data Explorer Linked Service from the dropdown.

On the **Command** tab of the new component, paste the following KQL in the Command textbox.

```
.drop table _product_pending ifexists;
```

Click **Debug** and monitor result to confirm successful progress.

### Create \_pending

Execution of this component will **create a new (and empty) “...pending” table** in Data Explorer.

Expand “**Azure Data Explorer**” in the **Activities** bar.

Drag-and-drop an “**Azure Data Explorer Command**” component into the activity window.

Create a dependency from the “**Drop Operation**” component to the new component.

On the **Connection** tab of the new component, select the Data Explorer Linked Service from the dropdown.

On the **Command** tab of the new component, paste the following KQL in the Command textbox.

```
.create table _product_pending ( _id:string, _op:string );
```

Logic explained...

- The “`_id`” field will be used as a string-based, generic capture of various identifier data types {e.g., int, guid, etc.} and situations where unique identification may require concatenation of multiple fields {e.g., “123-456”}
- The “`_op`” field will be used to flag rows for **insert**, **update**, or **delete** operations

Click **Debug** and monitor result to confirm successful progress.

## Capture Pending Deletes

Execution of this component will **capture pending delete operations** {i.e., items that exist in captured data but not in waterline data}.

Expand “**Move & Transform**” in the **Activities** bar, then drag-and-drop an “**Copy data**” component into the activity window.

Create a dependency from the “**Capture Waterline**” and “**Create Operation**” components to the new component.

On the **Source** tab of the new component, enter values for the following items:

---

Source Dataset	Create a dataset using the <b>Data Explorer</b> linked service and the <b>Product</b> table {e.g., “practicumded_Product”}
Use Query	Select the <b>Query</b> radio button
Query	Paste the following KQL: <pre>Product   join ( Product   summarize _dt = max( _dt ) by _id ) on _id, _dt   join kind = fullouter _product_waterline on _id   where isnull(_dt2) and _op != 'delete'   project _id, _op = 'delete'</pre>

Logic explained...

- The first **join** ensures that comparison to the most current version of the data
  - The second **join** is full outer to ensure that columns from both sides of the join are included
  - The first **where** clause filters to sink records that no longer exist at the source
  - The second **where** clause filters records where the latest operation was “delete”
  - The **project** clause surfaces the value for “\_id” and a fixed value for “\_op”
- 

No additional changes are required on this tab.

On the **Sink** tab of the new component, enter values for the following items:

---

Sink Dataset	Create and select a dataset using the <b>Data Explorer</b> linked service and the “_product_pending” table {e.g., “practicumded_product_pending”}
Database and Table	Confirm values

No additional changes are required on this tab.

On the **Mapping** tab of the new component, click the “Import schemas” button.

Click **Debug** and monitor result to confirm successful progress.

#### Activity 7: Capture Pending Inserts and Updates

Execution of this component will **capture pending insert operations** {i.e., items that exist in waterline data but not in captured data} and **capture pending update operations** {i.e., items that have a higher waterline at source than sink}.

Expand “**Move & Transform**” in the **Activities** bar, then drag-and-drop an “**Copy data**” component into the activity window.

Create a dependency from the “**Capture Waterline**” and “**Create \_pending**” components to the new component.

On the **Source** tab of the new component, enter values for the following items:

---

Source Dataset	Select <b>practicumded_Product</b>
Use Query	Select the <b>Query</b> radio button
Query	Paste the following KQL: <pre>Product_latest   join kind = rightanti _Product_waterline on _id   where isnull(_dt)   project _op = 'insert'   union ( Product_latest   join kind = inner _Product_waterline on _id   where _dt &lt; _dt1   project _id = _id1, _op = 'update' )</pre>

---

Logic (first part of UNION) explained...

- Use of “latest” from materialized view
- The **join** returns only records from the right that lack a match on the left
- The **where** clause filters to sink records that no longer exist at the source
- The **project** clause surfaces the value for “\_id” and a fixed value for “\_op”

Logic (second part of UNION) explained...

- Use of “latest” from materialized view
  - The **join** returns records from left and right with a matching “\_id” value
  - The **where** clause filters to source records with a higher waterline captured sink data
- The **project** clause surfaces the value for “\_id” and a fixed value for “\_op”
- 

No additional changes are required on this tab.

On the **Sink** tab of the new component, enter values for the following items:

<b>Sink Dataset</b>	Create a dataset using the <b>SQL</b> linked service and the “ <b>_product_pending</b> ” table {e.g., “ <b>practicumsd_product_pending</b> ”}
---------------------	--

Consider running the T-SQL below to create the table in advance of creating the sink dataset.

<b>Pre-Copy Script</b>	Paste the following T-SQL: <pre>DROP TABLE IF EXISTS _product_pending; CREATE TABLE _product_pending ( _id VARCHAR(256), _op VARCHAR(8) );</pre>
------------------------	---

No additional changes are required on this tab.

On the **Mapping** tab of the new component, click the “**Import schemas**” button.

Click **Debug** and monitor result to confirm successful progress.

## Capture Change

### Confirm Destination

Execution of this component will **confirm that the final data destination exists**.

Expand “**Azure Data Explorer**” in the **Activities** bar.

Drag-and-drop an “**Azure Data Explorer Command**” component into the activity window.

On the **Connection** tab of the new component, select the Data Explorer Linked Service from the dropdown.

On the **Command** tab of the new component, paste the following KQL in the Command textbox.

```
.create-merge table Product ( ProductId:int, ProductNumber:string, Name:string, ListPrice:decimal, ModifiedDate:datetime,
_id:string, _op:string, _dt:datetime );
```

Logic explained...

- Most fields will exactly match the source counterpart {e.g., ProductId, ProductNumber, Name, etc.}
- The “**\_id**” field will be used as a string-based, generic capture of various identifier data types {e.g., int, guid, etc.} and situations where unique identification may require concatenation of multiple fields {e.g., “123-456”}
- The “**\_op**” field will be used to flag rows for **delete**, **insert**, or **update** operations

Click **Debug** and monitor result to confirm successful progress.

## Capture Deletes (from Data Explorer)

Execution of this component will **write data associated with pending delete changes**.

Expand “**Move & Transform**” in the **Activities** bar, then drag-and-drop an “**Copy data**” component into the activity window. Create a dependency from the “**Confirm Destination**” and “**Pending Deletes**” components to the new component.

On the **Source** tab of the new component, enter values for the following items:

Source Dataset	Select <b>practicumded_Product</b>
Use Query	Select the <b>Query</b> radio button
Query	Paste the following parameterized KQL: <pre>Product   join ( Product   summarize _dt = max( _dt ) by _id ) on _id, _dt   join ( _product_pending   where _op == 'delete' ) on _id   project ProductId, ProductNumber, Name, ListPrice, ModifiedDate, _id, _dt, _op = _op1</pre>

Logic explained...

- The first **join** ensures that comparison to the most current version of the data
- The second **join** ensures inclusion of only rows with associated “delete” operations
- The **project** clause surfaces destination values and the correct “**\_op**” value

No additional changes are required on this tab.

On the **Sink** tab of the new component, enter values for the following items:

Sink Dataset	Select <b>practicumded_Product</b>
Database and Table	Confirm values

No additional changes are required on this tab.

On the **Mapping** tab of the new component, click the “**Import schemas**” button.

Click **Debug** and monitor result to confirm successful progress.

## Capture Inserts and Updates (from SQL)

Execution of this component will **write data associated with pending insert and update changes**.

Expand “**Move & Transform**” in the **Activities** bar, then drag-and-drop an “**Copy data**” component into the activity window. Create a dependency from the “**Confirm Destination**” and “**Pending Insert and Deletes**” components to the new component.

On the **Source** tab of the new component, enter values for the following items:

Source Dataset	Select <b>practicumsd_product_pending</b>
Use Query	Select the <b>Query</b> radio button
Query	Paste the following parameterized T-SQL: <pre>SELECT *, P.[ModifiedDate] [_dt] FROM [SalesLT].[Product] P INNER JOIN [dbo].[_product_pending] PP ON P.[ProductID] = PP.[_id]</pre>

No additional changes are required on this tab.

On the **Sink** tab of the new component, enter values for the following items:

Sink Dataset	Select <b>practicumded_Product</b>
--------------	------------------------------------

Database and Table	Confirm values
No additional changes are required on this tab.	

On the **Mapping** tab of the new component, click the “Import schemas” button.

Click **Debug** and monitor result to confirm successful progress.

## Step 3: Confirm Success

The first and most obvious confirmation is that your completed pipeline runs successfully.

Then, of course, confirm that data is correctly captured.

### Confirmation 1, Record Count

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Click the **Data** icon in the navigation, expand “**Data Explorer Databases...**”, and your Data Explorer Pool.

Right click on your Data Explorer Database {e.g., **practicumded**} and select “**New KQL script**” from the resulting dropdown.

Paste and then Run the following KQL query to validate debug runs {i.e. with a record count higher than 0}:

```
Product | summarize count()
```

### Confirmation 2, Delete Action

Navigate to your instance of SQL, click “**Query Editor**” in the navigation, and then login (whitelist your IP address, if required).

Execute the following T-SQL query to delete a record:

```
DELETE FROM [SalesLT].[Product] WHERE [ProductID]=( SELECT MIN( [ProductID] ) FROM [SalesLT].[Product] )
```

Navigate to the new pipeline in Synapse, and then click **Debug** and monitor result to confirm successful progress.

Execute the following KQL query to confirm that the deleted record is included in the Product table:

```
Product | where _op == 'delete'
```

Execute the following KQL query to confirm that you can see the full history for the deleted item:

```
Product | where ProductId == {deleted ProductId value}
```

Productid	ProductNumber	Name	ListPrice	ModifiedDate	_id	_operation	_dt
719	FR-R92R-48	HL Road Frame...	1431.5	2008-03-11T10:00:00.000Z	719	delete	2008-03-11T10:00:00.000Z
719	FR-R92R-48	HL Road Frame...	1431.5	2008-03-11T10:00:00.000Z	719	insert	2008-03-11T10:00:00.000Z

## Confirmation 3, Insert Action

Navigate to your instance of SQL, click “**Query Editor**” in the navigation, and then login (whitelist your IP address, if required).

Execute the following T-SQL query to insert a record:

```
INSERT INTO [SalesLT].[Product]
([Name],[ProductNumber],[Color],[StandardCost],[ListPrice],[SellStartDate],[rowguid],[ModifiedDate])
VALUES ('New Product ABC','ABC-123','Bronze',1000,1500,getdate(),newid(),getdate())
```

Navigate to the new pipeline in Synapse, and then click **Debug** and monitor result to confirm successful progress.

Execute the following KQL query to confirm that the inserted record is included in the Product table:

```
Product | where Name contains 'New Product'
```

## Confirmation 4, Update Action

Navigate to your instance of SQL, click “**Query Editor**” in the navigation, and then login (whitelist your IP address, if required).

Execute the following T-SQL query to update a record:

```
UPDATE SalesLT.Product SET ModifiedDate=GETDATE() WHERE ProductID=(SELECT MAX([ProductID]) FROM [SalesLT].[Product])
```

Navigate to the new pipeline in Synapse, and then click **Debug** and monitor result to confirm successful progress.

Execute the following KQL query to confirm that the updated record is included in the Product table:

```
Product | where Name contains 'New Product'
```

## Step 4: Create Trigger

Establish a recurring schedule for your completed pipeline.

Navigate to the new pipeline in Synapse, and then click the “**Add trigger**” button and click “**New/Edit**” in the resulting dropdown.

On the resulting popout, click the “**Choose trigger...**” dropdown and select “**+ New**”.

On the “**New trigger**” popout, enter values for the following items:

Name	Enter a meaningful name aligned with standards
Type	Select “ <b>Schedule</b> ”
Start Date	Confirm default value (consider using round hour / minute to avoid odd timing)
Time Zone	Confirm default value
Recurrence	Choose appropriate recurrence

No additional changes are required but be sure to confirm settings on the remaining tabs. Click the **Commit** button.

## Step 5: Create “\_Latest” Function

Simplify user experience {i.e., “I just always want to see the latest data!”} by preparing a Stored Function in Data Explorer.

```
.create-or-alter function product_latest ()  
{  
product | summarize _dt = max( _dt ) by _id  
| join kind = leftanti ( product | where _op == 'delete' | distinct _id ) on _id  
| join kind = rightsemi product on _id, _dt  
| project-away _id, _dt, _op  
}
```

*Objective Complete... congratulations!*

## Objective - Synapse (data flow) + Delta Lake | Incremental Load

*Note: These instructions also apply (with minor differences) to Azure Data Factory.*

Follow the instructions in this section to demonstrate incremental load using Synapse Data Flows and Delta Lake.

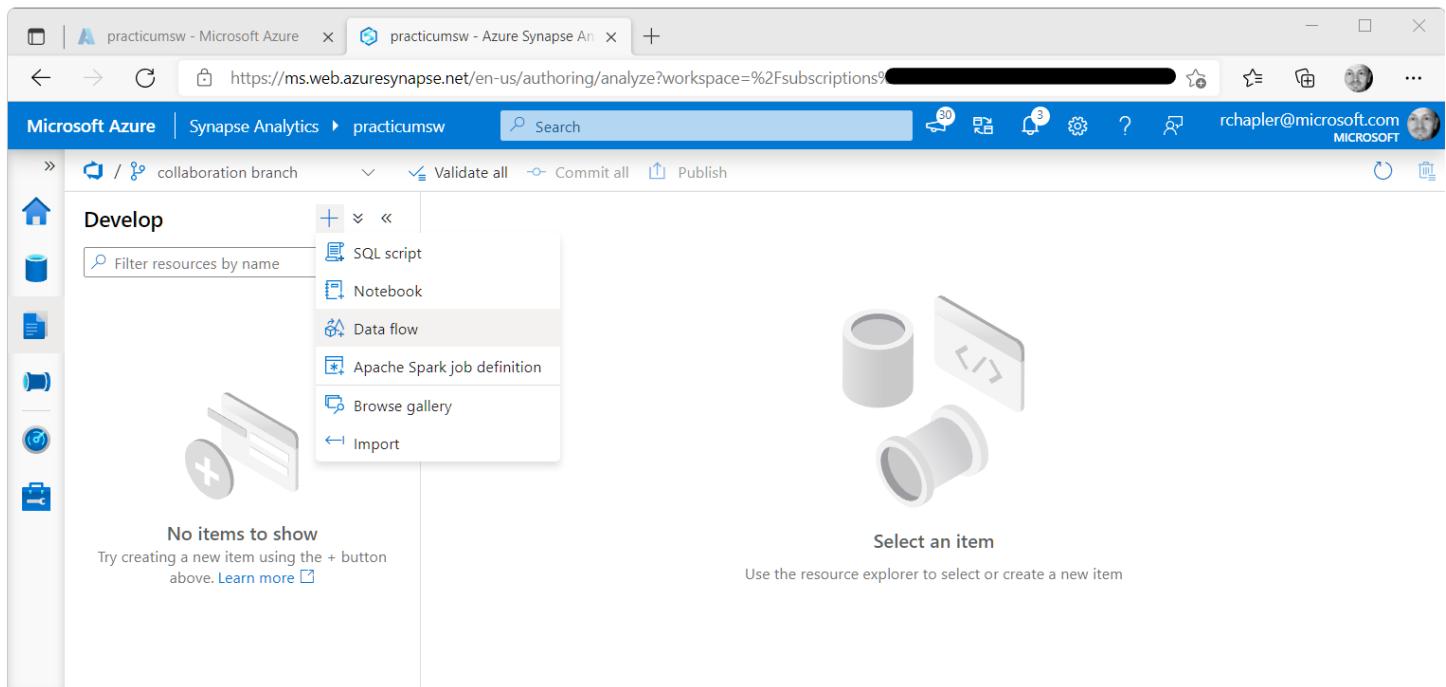
### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Data Lake
- SQL (with sample database)
- Synapse (with linked service for SQL and Apache Spark Pool)

### Step 2: Initial Load

This section describes how to load existing source data into Delta Lake (as a primer for Incremental Load).



Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

#### Create Data Flow

Click the **Develop** icon in the navigation.

Click the **+** icon just above and to the right of the “**Filter resources by name**” input.

Select “**Data flow**” from the resulting dropdown.

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow editor interface. At the top, there are two browser tabs: 'practicumsw - Microsoft Azure' and 'practicumsw - Azure Synapse Analytics'. The main window title is 'Microsoft Azure | Synapse Analytics > practicumsw'. The URL in the address bar is <https://ms.web.azuresynapse.net/en-us/authoring/analyze/dataflow/Dataflow1?workspace=%2Fsubscriptions%2F...>. The top navigation bar includes a search bar, notification icons (30, 23), and user information for rchaphler@microsoft.com.

The left sidebar shows a 'Develop' branch with a 'Data flows' section containing one item: 'Initial Load'. A 'Filter resources by name' input field is also present. The main workspace displays a 'Initial Load' pipeline with a single step labeled 'Add Source'. To the right of the pipeline is a 'Properties' panel for the 'Initial Load' data flow. The 'General' tab is selected, showing fields for 'Name' (set to 'Initial Load') and 'Description'. Below the pipeline, a 'Parameters' tab is visible with a '+ New' button.

On the **Properties** popout, enter values for the following items:

Name	Enter a meaningful name aligned with standards
------	--

Consider activating "**Data flow debug**" to avoid wait time later in the process.

Source (sample database)

Click on "**Add Source**" in the dashed rectangle.

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow blade. At the top, there are two tabs: 'practicumsw - Microsoft Azure' and 'practicumsw - Azure Synapse Analytics'. The URL in the address bar is <https://ms.web.azuresynapse.net/en-us/authoring/analyze/dataflow/Dataflow1?workspace=%2Fsubscriptions%2F...>. The main area displays a pipeline named 'Initial Load' under the 'Develop' branch. The pipeline consists of one step: 'practicumsd' (Integration dataset). Below the pipeline, the 'Source settings' tab is selected, showing configuration for the output stream name 'practicumsd', source type 'Inline', inline dataset type 'Azure SQL Database', linked service 'practicumsd' (with a successful test connection), and sampling settings.

On the “Source settings” tab, enter values for the following items:

<b>Output Stream Name</b>	Enter a meaningful name aligned with standards
<b>Source Type</b>	Select <b>Inline</b>
<b>Inline Dataset Type</b>	Select “ <b>Azure SQL Database</b> ”
<b>Linked Service</b>	Select <b>practicumsd</b>
<b>Sampling</b>	Confirm default setting, <b>Disable</b>

Click “Test connection” to confirm successful configuration.

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow blade. In the center, there is a component named "Initial Load". Below it, under the "Source options" tab, the "Input" section is selected. It shows the following configuration:

- Input: Table (radio button selected)
- Schema name \*: SalesLT
- Table name \*: Product
- Isolation level: Read uncommitted

At the top of the blade, there are tabs for "Source settings", "Source options" (which is active), "Projection", "Optimize", "Inspect", "Data preview", and "Description". There are also buttons for "Commit", "Validate", "Data flow debug" (which is turned on), and "Debug Settings".

On the “Source options” tab, enter values for the following items:

Input	Confirm default setting, <b>Table</b>
Schema Name	Enter <b>SalesLT</b>
Table Name	Enter <b>Product</b>

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow blade with the "Projection" tab selected. Under the "Projection" tab, the "Import schema" button is highlighted. Below the table, there is a list of columns and their types:

Column name	Type
ProductID	integer
Name	string
ProductNumber	string

On the **Projection** tab, click “Import schema” and then **Import** on the resulting popout.

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow interface. In the center, there is a data preview pane for a flow named 'Initial Load'. The preview shows 295 rows of data from a source named 'practicumsd'. The columns listed are ProductID, Name, ProductNumber, and Color. The data includes entries for various products like HL Road Frame - Black, Sport-100 Helmet, and Mountain Bike Saddle. At the top of the preview pane, there are tabs for Source settings, Source options, Projection, Optimize, Inspect, and Data preview. The 'Data preview' tab is selected. Below the tabs, there are buttons for Refresh, Typecast, Modify, Map drifted, Statistics, and Remove. The 'Refresh' button is highlighted.

Navigate to the “**Data preview**” tab and then click **Refresh**.

Click the **Commit** button.

## Sink (delta lake)

This screenshot is similar to the previous one but shows a 'sink' destination selected in the 'Destination' dropdown. The rest of the interface and data preview are identical to the first screenshot, showing the same 295 rows of product data from the source 'practicumsd'.

Click the + in the bottom right of the Data Lake source, and then search for and select **Sink** from the resulting popup list.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. At the top, there are two tabs: 'Initial Load' and 'Initial Load'. Below this, the 'Sink' tab is selected. The configuration includes:

- Output stream name:** practicumdl
- Incoming stream:** practicumsd
- Sink type:** Inline (selected)
- Inline dataset type:** Delta
- Linked service:** practicumsw-WorkspaceDefaultStorage (with 'Test connection' and 'Edit' buttons)
- Options:** Allow schema drift (checked), Validate schema (unchecked)

On the **Sink** tab, enter values for the following items:

<b>Output Stream Name</b>	Enter a meaningful name aligned with standards
<b>Incoming Stream</b>	Confirm default, <b>practicumsd</b>
<b>Sink Type</b>	Select <b>Inline</b>
<b>Linked Service</b>	Select <b>practicumsw-WorkspaceDefaultStorage</b>
<b>Options</b>	Confirm that " <b>Allow schema drift</b> " is checked

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. In the center, there's a diagram titled "Initial Load" showing a flow from a source dataset "practicumsd" to a sink dataset "practicumdlc". The sink dataset has 17 total columns. Below the diagram, the "Settings" tab is active, showing the following configuration:

- Folder path \***: practicumdlc / bronze/northwind/prc
- Compression type**: snappy
- Compression level**: Fastest
- Vacuum**: 0
- Table action**: None (radio button selected)
- Update method**:
  - Allow insert
  - Allow delete
  - Allow upsert
  - Allow update
- Delta options**:
  - Merge schema**:
  - Auto compact**:
  - Optimize write**:

On the **Settings** tab, enter values for the following items:

<b>File System</b>	Enter <b>practicumdlc</b>
<b>Folder Path</b>	Enter " <b>bronze/northwind/product</b> "
<b>Compression Type</b>	Select <b>snappy</b>
<b>Compression Level</b>	Select <b>Fastest</b>
<b>Auto Compact</b>	Checked
<b>Optimize Write</b>	Checked

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the **Commit** button.

## Create Pipeline

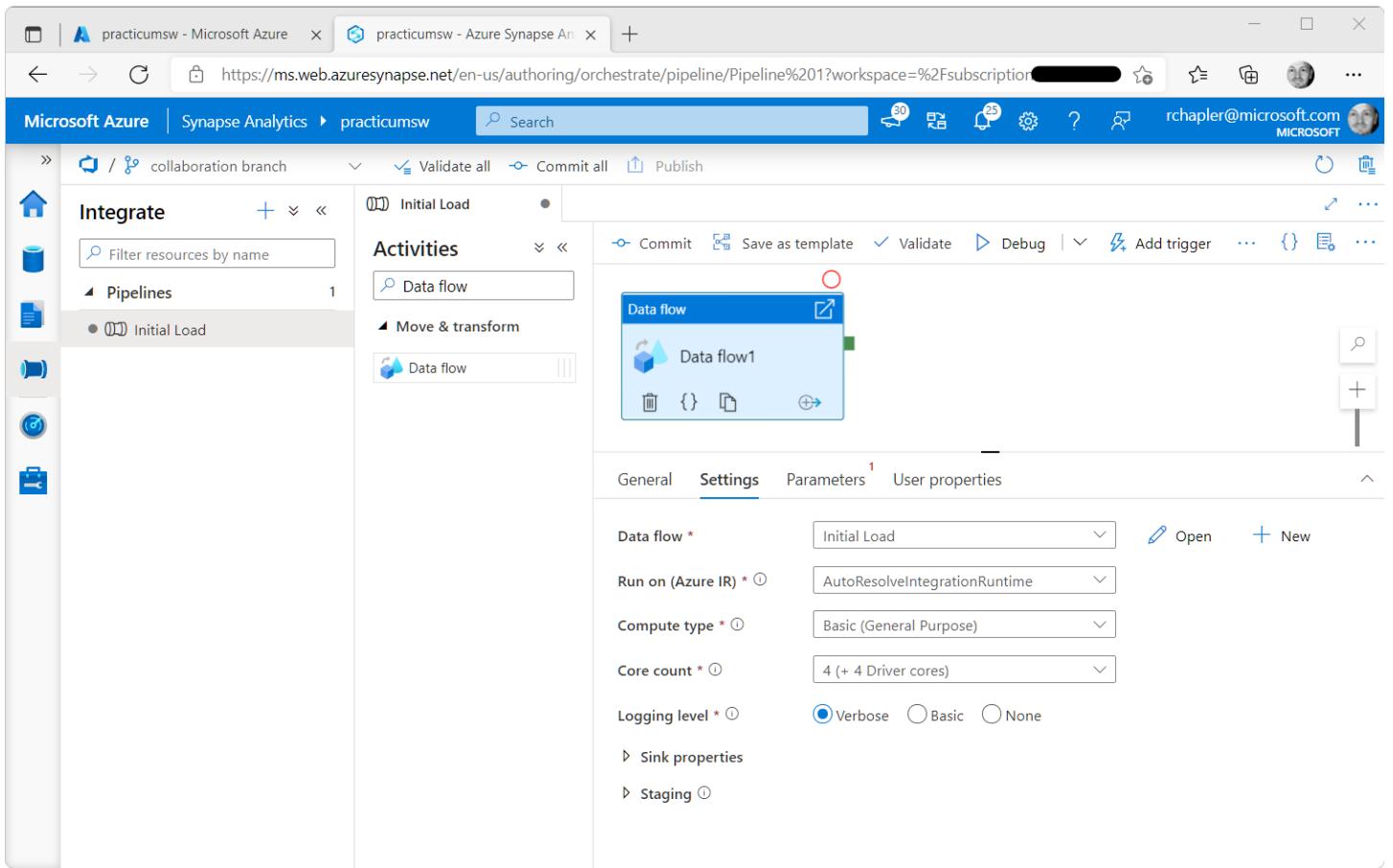
The screenshot shows the Microsoft Azure Synapse Analytics portal interface. The top navigation bar includes 'practicumsw - Microsoft Azure' and 'practicumsw - Azure Synapse Analytics'. The main area has a 'Microsoft Azure' header with 'Synapse Analytics' and 'practicumsw' sections, along with a search bar and various status indicators. On the left, there's a vertical sidebar with icons for Home, Collaboration branch, Validate all, Commit all, Publish, and a plus sign for creating new items. The main content area is titled 'Integrate' and displays a message 'No items to show' with a note to try creating a new item using the '+' button. A dropdown menu is open above the 'Filter resources by name' input field, listing 'Pipeline', 'Copy Data tool', 'Browse gallery', and 'Import from pipeline template'. To the right of the dropdown, there's a large icon of a cylinder and a screen with code snippets, and a message 'Select an item' with a note to use the resource explorer.

Click the **Integrate** icon in the navigation.

Click the + icon just above and to the right of the “**Filter resources by name**” input.

Select “**Pipeline**” from the resulting dropdown.

On the **Properties** popout, enter Name, “**Initial Load**”.



Search for, and then drag-and-drop a “**Data flow**” component into the activity window.

On the **Settings** tab of the “**Data flow...**” component, select your data flow.

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the **Commit** button.

Confirm Success

Debug Pipeline / Data Flow

Click **Debug** and confirm successful execution.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline interface. On the left, the navigation bar includes 'Microsoft Azure', 'Synapse Analytics', and 'practicumsw'. The main area displays the 'Initial Load' pipeline. The 'Activities' section lists various options like Synapse, Move & transform, Azure Data Explorer, etc. A single 'Data flow' activity named 'Data flow1' is selected. Below it, the 'Output' tab shows a table with one row: 'Data flow1' (Type: Data flow, Run start: 2021-10-13T13:51:32.702, Duration: 00:00:19, Status: Succeeded, Integration: AutoResolve).

## Query Delta Lake

Navigate to your instance of Synapse, and then click the **Develop** icon in the navigation.

Click the + icon just above and to the right of the “**Filter resources by name**” input.

Select “**Notebook**” from the resulting dropdown.

In your new notebook, select your Apache Spark pool (if required) from the “**Attach to**” dropdown.

The screenshot shows the Microsoft Azure Synapse Analytics notebook interface. The navigation bar includes 'Microsoft Azure', 'Synapse Analytics', and 'practicumsw'. The main area displays 'Notebook 1'. The code cell contains PySpark code: 

```
%pyspark
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
print(df.count())
```

. The output shows the command was executed successfully in 3 min 7 sec, with 1 executor and 4 cores. The count of records is 295. Buttons for '+ Code' and '+ Markdown' are at the bottom.

Run the following code to produce a count of records in the new Delta Lake:

```
%pyspark
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
print(df.count())
```

The screenshot shows the Microsoft Azure Synapse Analytics notebook interface. In the center, a notebook titled 'Notebook 1' is open, showing a single cell with the following Python code:

```
1 %%pyspark
2 df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
3 display(df.limit(3))
4
```

Below the code, a message indicates the command was executed successfully: **[2] ✓ 6 sec - Command executed in 5 sec 906 ms by rchapler on 7:24:44 AM, 10/13/21**. A link to 'Job execution' is provided.

The results of the command are displayed in a table view:

ProductID	Name	ProductNumber	Color	Sta
680	HL Road Frame - Black, 58	FR-R92B-58	Black	10!
707	Sport-100 Helmet, Red	HL-U509-R	Red	13.
708	Sport-100 Helmet, Black	HL-U509	Black	13.

Run the following code to display a few records in the new Delta Lake:

```
%%pyspark
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
display(df.limit(3))
```

## Step 3: Incremental Load

This section describes how to provide for recurring capture of changes to source data.

### Create Data Flow

Click the **Develop** icon in the navigation.

Click the **+** icon just above and to the right of the **Filter resources by name** input.

Select **Data flow** from the resulting dropdown.

Name your data flow **“Incremental Load”**

### Source (delta lake)

In this step, we will add the Delta Lake (created in the previous section) as a source.

Click on **“Add Source”** in the dashed rectangle.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. In the left sidebar, under the 'Develop' section, 'Incremental Load' is selected. The main area displays a data flow diagram with a single component named 'practicumndl'. Below the diagram, the 'Source settings' tab is active. The configuration includes:

- Output stream name:** practicumndl
- Source type:** Inline (selected)
- Inline dataset type:** Delta
- Linked service:** practicumsw-WorkspaceDefaultStorage (selected)
- Sampling:** Disable (selected)

On the “Source settings” tab, enter values for the following items:

<b>Output Stream Name</b>	Enter a meaningful name aligned with standards
<b>Source Type</b>	Select <b>Inline</b>
<b>Inline Dataset Type</b>	Select <b>Delta</b>
<b>Linked Service</b>	Select <b>practicumsw-WorkspaceDefaultStorage</b>
<b>Sampling</b>	Confirm default setting, <b>Disable</b>

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. In the left sidebar under 'Develop', 'Incremental Load' is selected. The main area displays an 'Incremental Load' configuration with a preview pane showing a single row from a file named 'practicumdlc'. The 'Source options' tab is active, showing the following settings:

- Folder path: practicumdlc / bronze/northwind/prc
- Allow no files found: Checked
- Compression type: snappy
- Compression level: Fastest
- Time travel: Disable (radio button selected)

On the “Source options” tab, enter values for the following items:

File System	Enter <b>practicumdlc</b>
Folder Path	Enter “ <b>bronze/northwind/product</b> ”
Allow No Files Found	<b>Checked</b>
Compression Type	Select <b>snappy</b>
Compression Level	Select <b>Fastest</b>
Time Travel	Confirm default, <b>Disable</b>

The screenshot shows the 'Import schema' dialog in the Microsoft Azure Synapse Analytics interface. On the left, there's a navigation sidebar with icons for Home, Databases, Tables, Scripts, Data flows, and Notebooks. The main area shows a 'Develop' branch with a 'practicumdl' dataset selected. The 'Source settings' tab is active, showing 'practicumdl' with '0 total' columns. A dropdown menu for 'Import schema' is open, listing various data types: Date, Time, Numerical whole number, Numerical fraction, Boolean true, and Boolean false. At the bottom right of the dialog are 'Import' and 'Cancel' buttons.

On the **Projection** tab, click “**Import schema**” and then **Import** on the resulting popout.

Navigate to the “**Data preview**” tab and then click **Refresh**.

The screenshot shows the 'Data preview' tab in the Microsoft Azure Synapse Analytics interface. The top navigation bar includes 'Validate all', 'Commit all', and 'Publish' buttons. Below the navigation is a toolbar with icons for Data flow debug (green checkmark), Debug Settings, and other options. The main area displays the 'practicumdl' dataset with '17 total' columns. The 'Data preview' tab is selected, showing a table with the following data:

ProductID	Name	ProductNumber	Color
680	HL Road Frame - Black, 58	FR-R92B-58	Black
707	Sport-100 Helmet, Red	HL-U509-R	Red
708	Sport-100 Helmet, Black	HL-U509	Black

Click the **Commit** button.

## Source (sample database for this exercise)

In this step, we will add the SQL Northwind sample database (our primary source for this exercise).

Click on “Add Source” in the dashed rectangle.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. The left sidebar shows 'Develop' mode with 'Data flows' selected. The main area shows two datasets: 'practicumdl' (Add source dataset) and 'practicumsd' (Columns: 0 total). The 'practicumsd' dataset is currently selected. The 'Source settings' tab is active, displaying the following configuration:

- Output stream name \***: practicumsd
- Source type \***: **Inline** (selected)
- Inline dataset type \***: Azure SQL Database
- Linked service \***: practicumsd (Connection successful)
- Sampling \***:  Disable

On the “Source settings” tab, enter values for the following items:

<b>Output Stream Name</b>	Enter a meaningful name aligned with standards
<b>Source Type</b>	Select <b>Inline</b>
<b>Inline Dataset Type</b>	Select “ <b>Azure SQL Database</b> ”
<b>Linked Service</b>	Select <b>practicumsd</b>
<b>Sampling</b>	Confirm default setting, <b>Disable</b>

Click “Test connection” to confirm successful configuration.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. On the left, there's a navigation pane with icons for Home, Databases, Notebooks, Data flows, and Incremental Load. The 'Data flows' section is expanded, showing two items: 'Incremental Load' and 'Initial Load'. The 'Incremental Load' item is selected. The main workspace displays two stages: 'practicumdl' (Incremental Load) and 'practicumsd' (Initial Load). The 'practicumdl' stage has an 'Add source dataset' button. Below the stages, there are tabs for 'Source settings', 'Source options' (which is selected), 'Projection', 'Optimize', 'Inspect', 'Data preview', and 'Description'. Under the 'Source options' tab, the 'Input' section is configured with 'Table' selected as the schema, 'SalesLT' as the schema name, 'Product' as the table name, and 'Read uncommitted' as the isolation level. There is also a 'Refresh' button.

On the “Source options” tab, enter values for the following items:

Input	Confirm default setting, <b>Table</b>
Schema Name	Enter <b>SalesLT</b>
Table Name	Enter <b>Product</b>

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow workspace. The left sidebar lists 'Develop' (1), 'Notebooks' (1), 'Data flows' (2), and 'Incremental Load' (selected). The main area displays the 'Incremental Load' pipeline. It consists of two stages: 'practicumdl' (Add source dataset) and 'practicumsd' (Columns: 17 total). The 'Projection' tab is selected, showing the schema:

Column name	Type
ProductID	integer
Name	string
ProductNumber	string
Color	string

On the **Projection** tab, click “**Import schema**” and then **Import** on the resulting popout.

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow interface. On the left, the navigation pane is visible with sections for Develop, Notebooks, Data flows, and Incremental Load. The 'Incremental Load' section is currently selected. In the main workspace, there are two datasets: 'practicumdl' (Add source dataset) and 'practicumsd' (Columns: 17 total). A join operation is being performed between them. The 'Data preview' tab is selected at the bottom, showing a table with three rows. The columns are ProductID, Name, ProductNumber, and Color. The data is as follows:

	ProductID	Name	ProductNumber	Color
+	680	HL Road Frame - Black, 58	FR-R92B-58	Black
+	707	Sport-100 Helmet, Red	HL-U509-R	Red

Navigate to the “**Data preview**” tab and then click **Refresh**.

Click the **Commit** button.

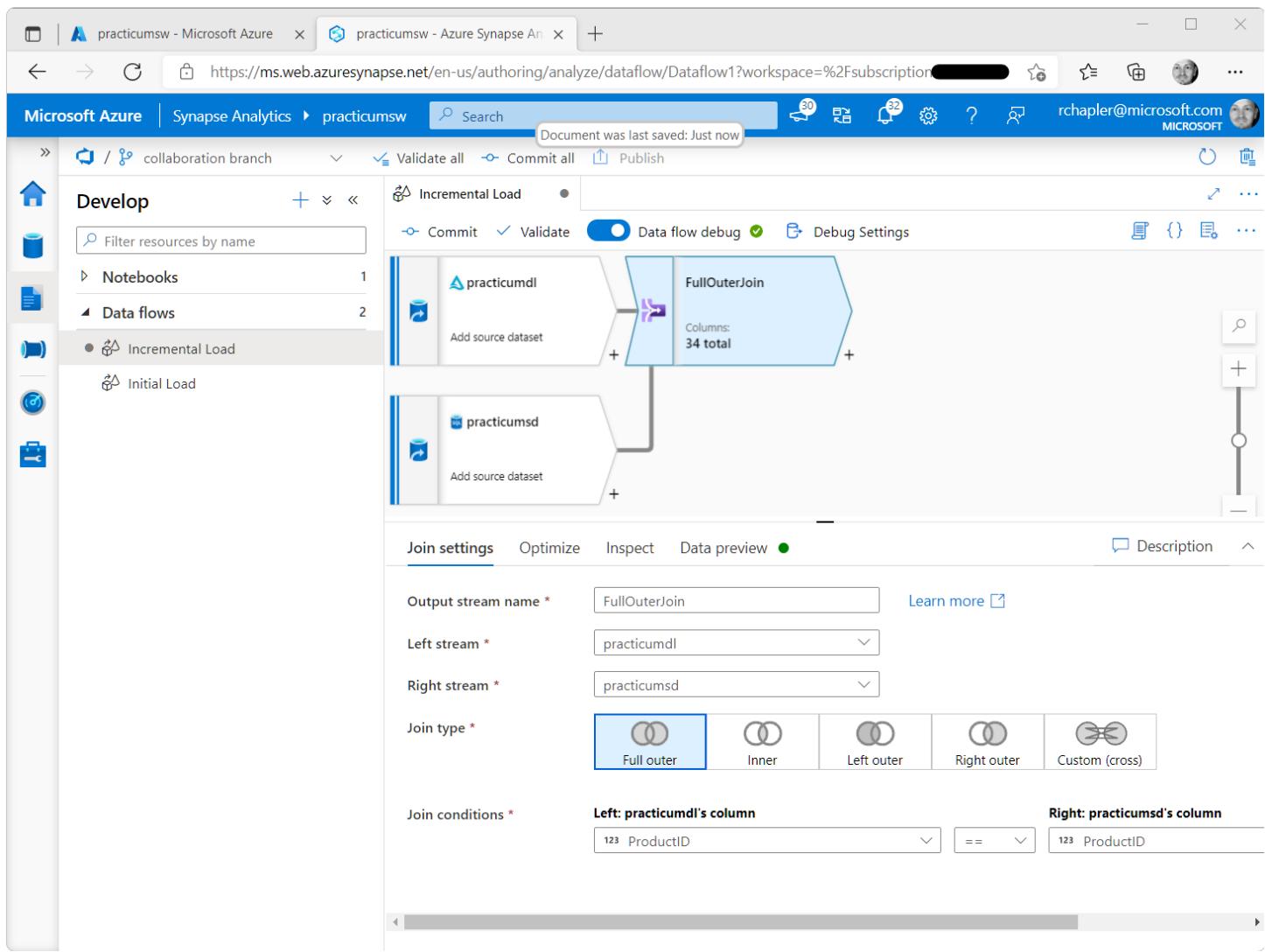
#### Join Sources

In this step, we will perform a full outer join between source datasets.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow interface. On the left, the navigation pane is visible with sections like 'Develop', 'Notebooks', 'Data flows', and 'Initial Load'. The 'Data flows' section is currently selected, showing a pipeline named 'Incremental Load'. The pipeline consists of two main components: 'practicumdl' (an Azure SQL Database source) and 'practicumsd' (another Azure SQL Database source). A 'join' operation is being performed between them. The 'Data preview' tab is selected, displaying the results of the join. The preview table has columns: ProductID, Name, ProductNumber, and Color. It contains two rows of data:

	ProductID	Name	ProductNumber	Color
+	680	HL Road Frame - Black, 58	FR-R92B-58	Black
+	707	Sport-100 Helmet, Red	HL-U509-R	Red

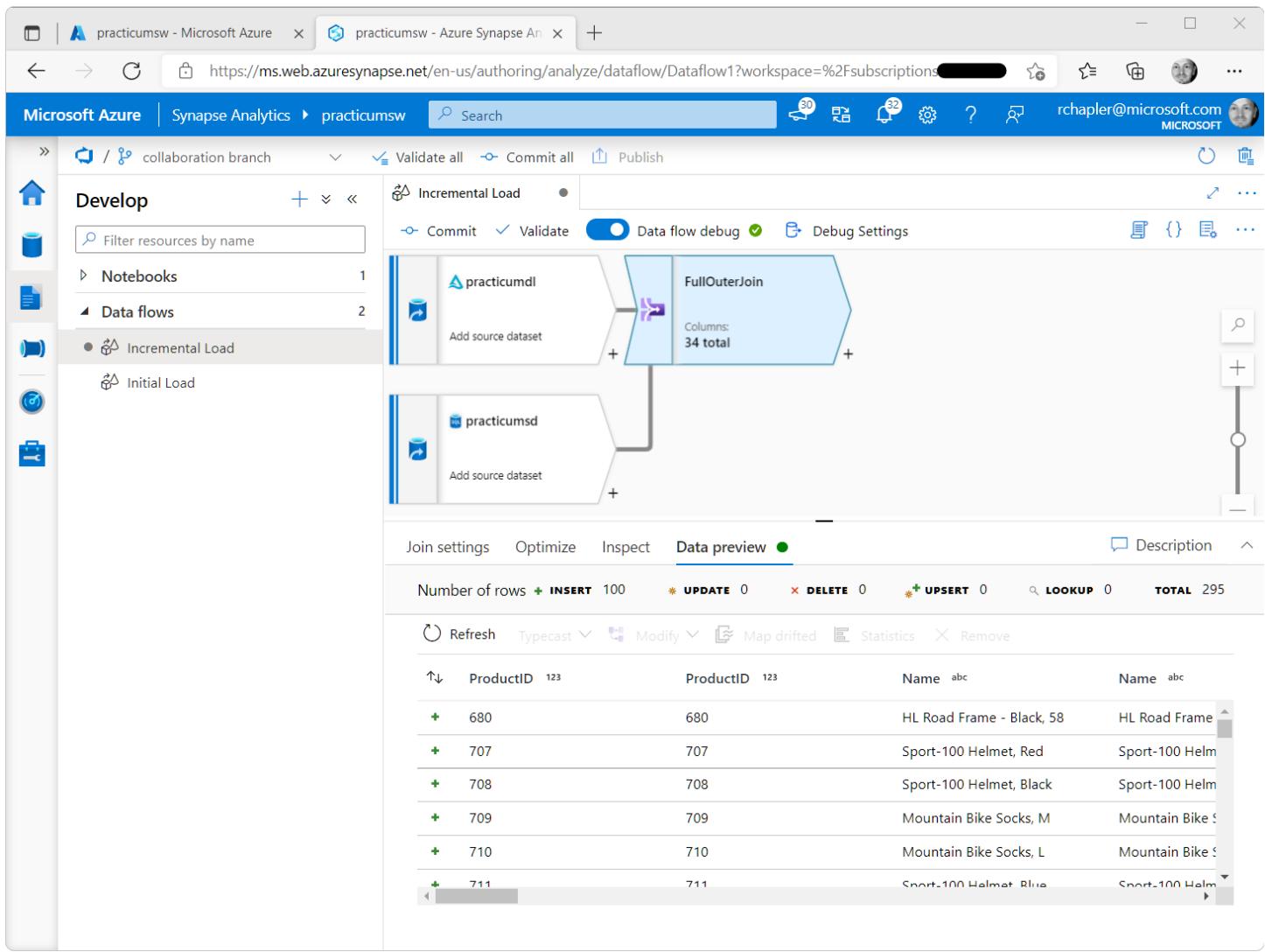
Click the **+** in the bottom right of the SQL source, and then select **Join** from the resulting popup list.



On the “Join settings” tab, enter values for the following items:

<b>Output Stream Name</b>	Enter a meaningful name aligned with standards
<b>Left Stream</b>	Select <b>practicumsd</b>
<b>Right Stream</b>	Select <b>AggregateWaterline</b>
<b>Join Type</b>	Select “ <b>Full outer</b> ”
<b>Join Conditions</b>	Select <b>ProductID</b> for both Left... and Right... column

Navigate to the “Data preview” tab and then click **Refresh**.



Click the **Commit** button.

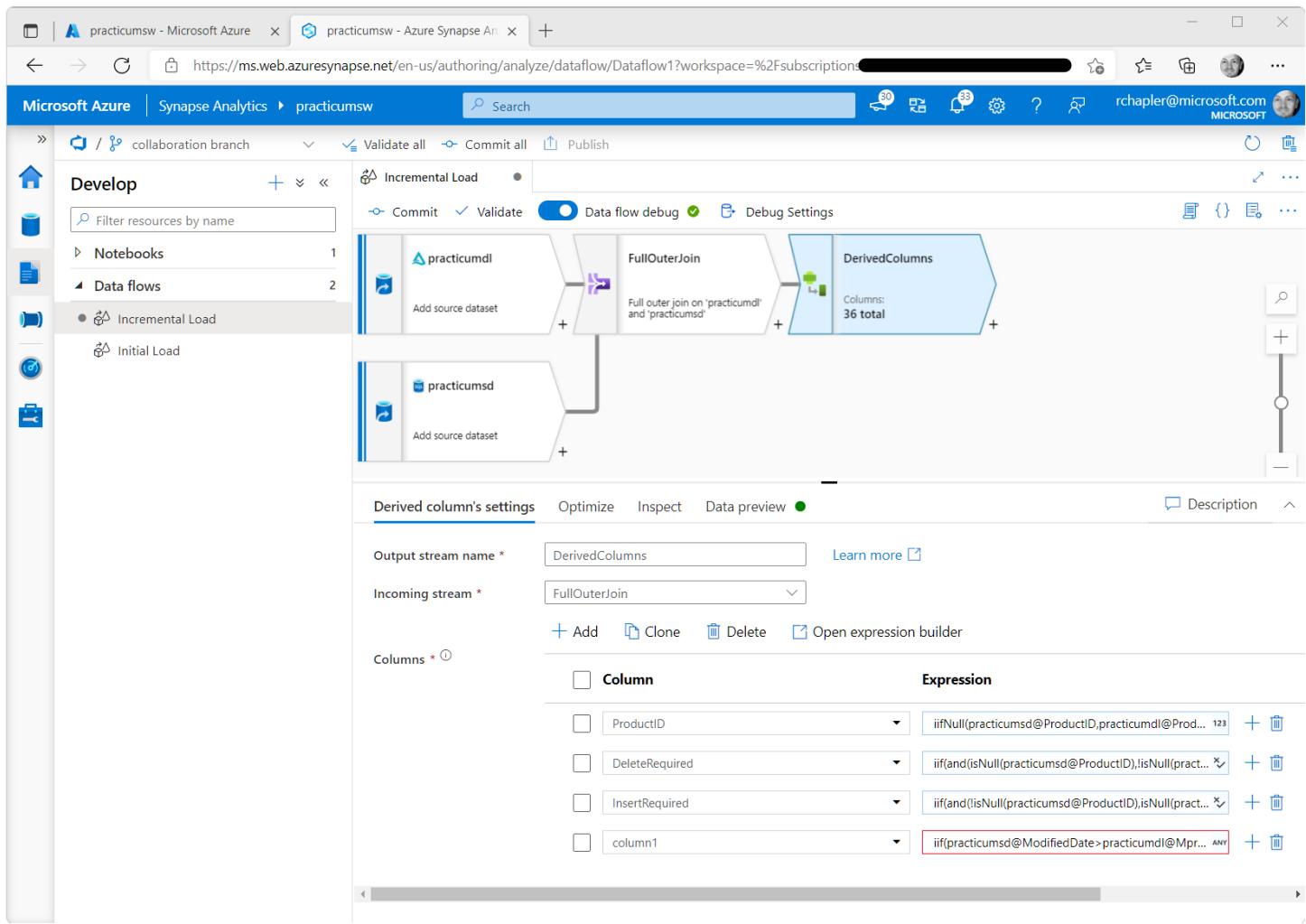
### Derive Columns

In this step, we will add columns designed to characterize required Delta Lake changes.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. On the left, the navigation pane is visible with sections like 'Develop', 'Notebooks', 'Data flows', and 'Initial Load'. The main workspace displays a data flow diagram titled 'Incremental Load'. It consists of two 'Add source dataset' components (labeled 'practicumdl' and 'practicumsd') connected to a 'FullOuterJoin' component. The 'FullOuterJoin' component has 34 total columns. A 'Schema modifier' panel is open on the right, showing a search bar for 'derived column' and a list item 'Derived Column'. Below the diagram, the 'Data preview' tab is selected, showing a table with the following data:

	ProductID	ProductID	Name	Name	ProductN
+	680	680	HL Road Frame - Black, 58	HL Road Frame - Black, 58	FR-R9,
+	707	707	Sport-100 Helmet, Red	Sport-100 Helmet, Red	HL-U5
+	708	708	Sport-100 Helmet, Black	Sport-100 Helmet, Black	HL-U5
+	709	709	Mountain Bike Socks, M	Mountain Bike Socks, M	SO-B9
+	710	710	Mountain Bike Socks, L	Mountain Bike Socks, L	SO-B9
+	711	711	Sport-100 Helmet, Blue	Sport-100 Helmet, Blue	HL-U5
+	712	712	AWC Logo Cap	AWC Logo Cap	CA-10

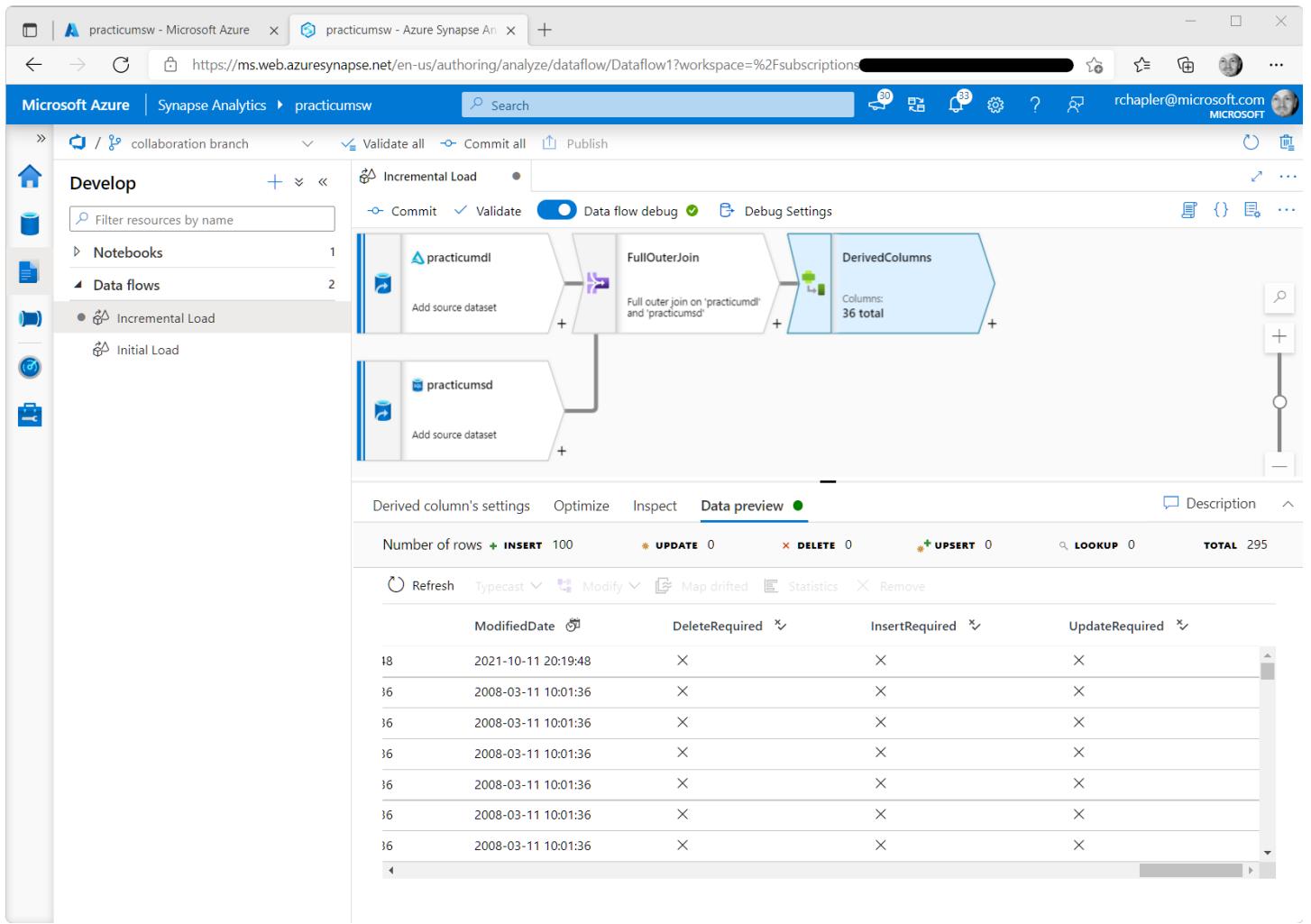
Click the **+** in the bottom right of the SQL source, and then select **Derived Column** from the resulting popup list.



On the “Derived column’s settings” tab, enter values for the following items:

<b>Output Stream Name</b>	Enter a meaningful name aligned with standards
<b>Incoming Stream</b>	Confirm default selection
<b>Columns</b>	<p><b>ProductID</b>  <code>iifNull(practicumsd@ProductID,practicumdl@ProductID)</code></p> <p><b>DeleteRequired</b>  <code>iif(and(isNull(practicumsd@ProductID),!isNull(practicumdl@ProductID)),true(),false())</code></p> <p><b>InsertRequired</b>  <code>iif(and(!isNull(practicumsd@ProductID),isNull(practicumdl@ProductID)),true(),false())</code></p> <p><b>UpdateRequired</b>  <code>iif(practicumsd@ModifiedDate&gt;practicumdl@ModifiedDate,true(),false())</code></p>

Navigate to the “Data preview” tab and then click **Refresh**.



Click the **Commit** button.

*Note: I included this step to provide an easy way to preview and provide for the delete, insert and update of data... arguably, this step could be consolidated into Filter or AlterRows.*

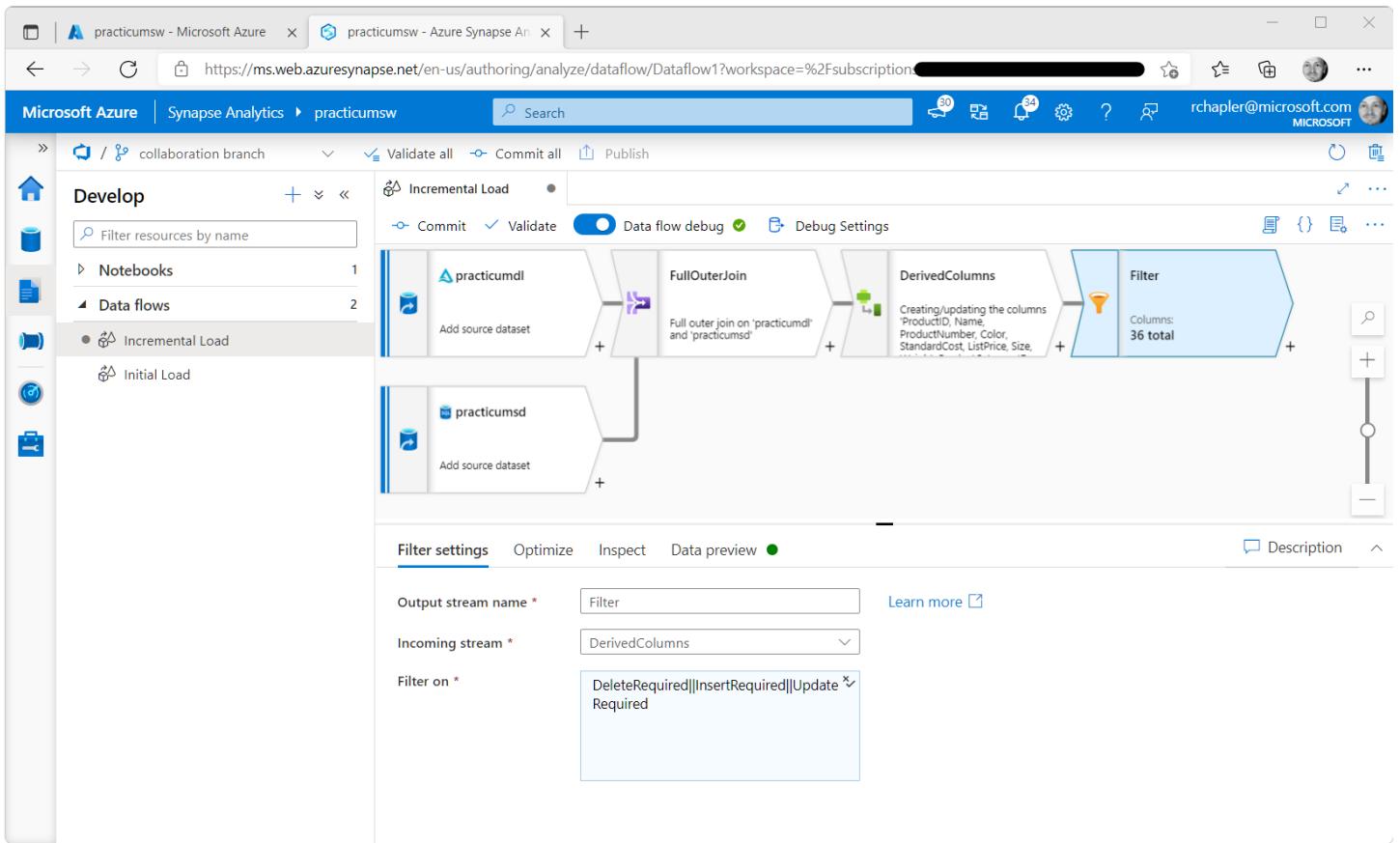
### Filter Non-Actionable

In this step, we will remove rows that have no required action.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. On the left, the navigation pane is visible with sections like 'Develop' and 'Data flows'. The main area displays a data flow named 'Incremental Load'. The flow consists of two source datasets: 'practicumdl' and 'practicumsd', which are joined via a 'FullOuterJoin' operation. The output of this join is then processed by a 'DerivedColumns' operation, resulting in 36 total columns. A 'Row modifier' step is present, followed by a 'Filter' step. The 'Data preview' tab is selected, showing a preview of 100 rows. The preview table has columns: ProductID, Name, Name, ProductNumber, and ProductN. The data includes various products like 'HL Road Frame - Black, 58', 'Sport-100 Helmet, Red', etc.

ProductID	Name	Name	ProductNumber	ProductN
680	HL Road Frame - Black, 58	HL Road Frame - Black, 58	FR-R92B-58	FR-R9
707	Sport-100 Helmet, Red	Sport-100 Helmet, Red	HL-U509-R	HL-U5
708	Sport-100 Helmet, Black	Sport-100 Helmet, Black	HL-U509	HL-U5
709	Mountain Bike Socks, M	Mountain Bike Socks, M	SO-B909-M	SO-B9
710	Mountain Bike Socks, L	Mountain Bike Socks, L	SO-B909-L	SO-B9
711	Sport-100 Helmet, Blue	Sport-100 Helmet, Blue	HL-U509-B	HL-U5
712	AWC Logo Cap	AWC Logo Cap	CA-1098	CA-10

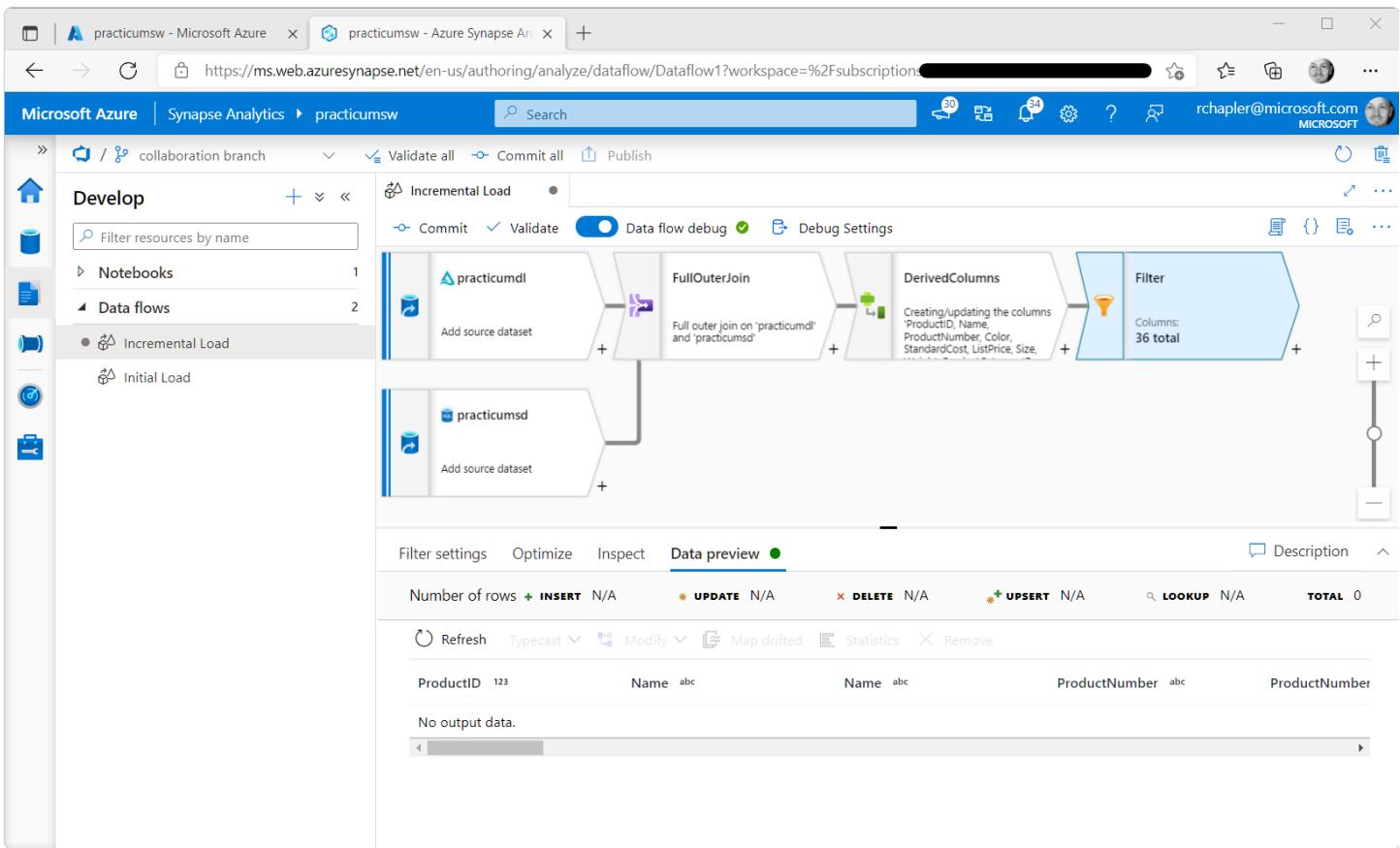
Click the **+** in the bottom right of the SQL source, and then select **Filter** from the resulting popup list.



On the “**Filter settings**” tab, enter values for the following items:

<b>Output Stream Name</b>	Enter a meaningful name aligned with standards
<b>Incoming Stream</b>	Confirm default selection
<b>Filter On</b>	<code>DeleteRequired    InsertRequired    UpdateRequired</code>

Navigate to the “**Data preview**” tab and then click **Refresh**.



Click the **Commit** button.

*Note: Because we are doing an incremental load immediately after an initial load, on a sample database which doesn't have unexpected change, we see no current results in the Data Preview.*

## Change Examples

To make the remaining steps a bit more interesting, we will force changes to the sample database.

Navigate to your Azure SQL Database, click the “**Query editor (preview)**” link in the navigation, and login.

## Delete Example

The screenshot shows the Microsoft Azure Query editor (preview) interface. The left sidebar lists various services: Overview, Activity log, Tags, Diagnose and solve problems, Quick start, and Query editor (preview). The Query editor (preview) option is selected. The main area displays a query window titled 'Query 1' containing the following T-SQL code:

```
1 DELETE FROM [SalesLT].[Product] WHERE [ProductID]=730
```

The 'Messages' tab shows the result: 'Query succeeded: Affected rows: 1'. A yellow banner at the bottom right indicates 'Query succeeded | 0s'.

Execute the following T-SQL:

```
DELETE FROM [SalesLT].[Product] WHERE [ProductID]=730
```

*Note: I chose 730 because it did not trigger constraints; any row deletion will work.*

## Insert Example

The screenshot shows the Microsoft Azure Query editor (preview) interface. The left sidebar lists various services: Overview, Activity log, Tags, Diagnose and solve problems, Quick start, and Query editor (preview). The Query editor (preview) option is selected. The main area displays a query window titled "Query 1". The query code is:

```
1 INSERT INTO [SalesLT].[Product]
2     ([Name],[ProductNumber],[Color],[StandardCost],[ListPrice],[SellStartDate],[rowguid],[ModifiedDate])
3 VALUES
4     ('New Product ABC','ABC-123','Bronze',1000,1500,getdate(),newid(),getdate())
```

The "Messages" tab shows the message: "Query succeeded: Affected rows: 1". A yellow banner at the bottom right indicates "Query succeeded | 0s".

Execute the following T-SQL:

```
INSERT INTO [SalesLT].[Product]
    ([Name],[ProductNumber],[Color],[StandardCost],[ListPrice],[SellStartDate],[rowguid],[ModifiedDate])
VALUES
    ('New Product ABC','ABC-123','Bronze',1000,1500,getdate(),newid(),getdate())
```

## Update Example

The screenshot shows the Microsoft Azure Query editor (preview) interface. The left sidebar lists various services: Overview, Activity log, Tags, Diagnose and solve problems, Quick start, and Query editor (preview). The Query editor (preview) option is selected. The main area displays a query window titled "Query 1". The query code is:

```
1 UPDATE [SalesLT].[Product]
2 SET [ListPrice] = 999.99, [ModifiedDate] = getdate()
3 WHERE ProductID=680
4
```

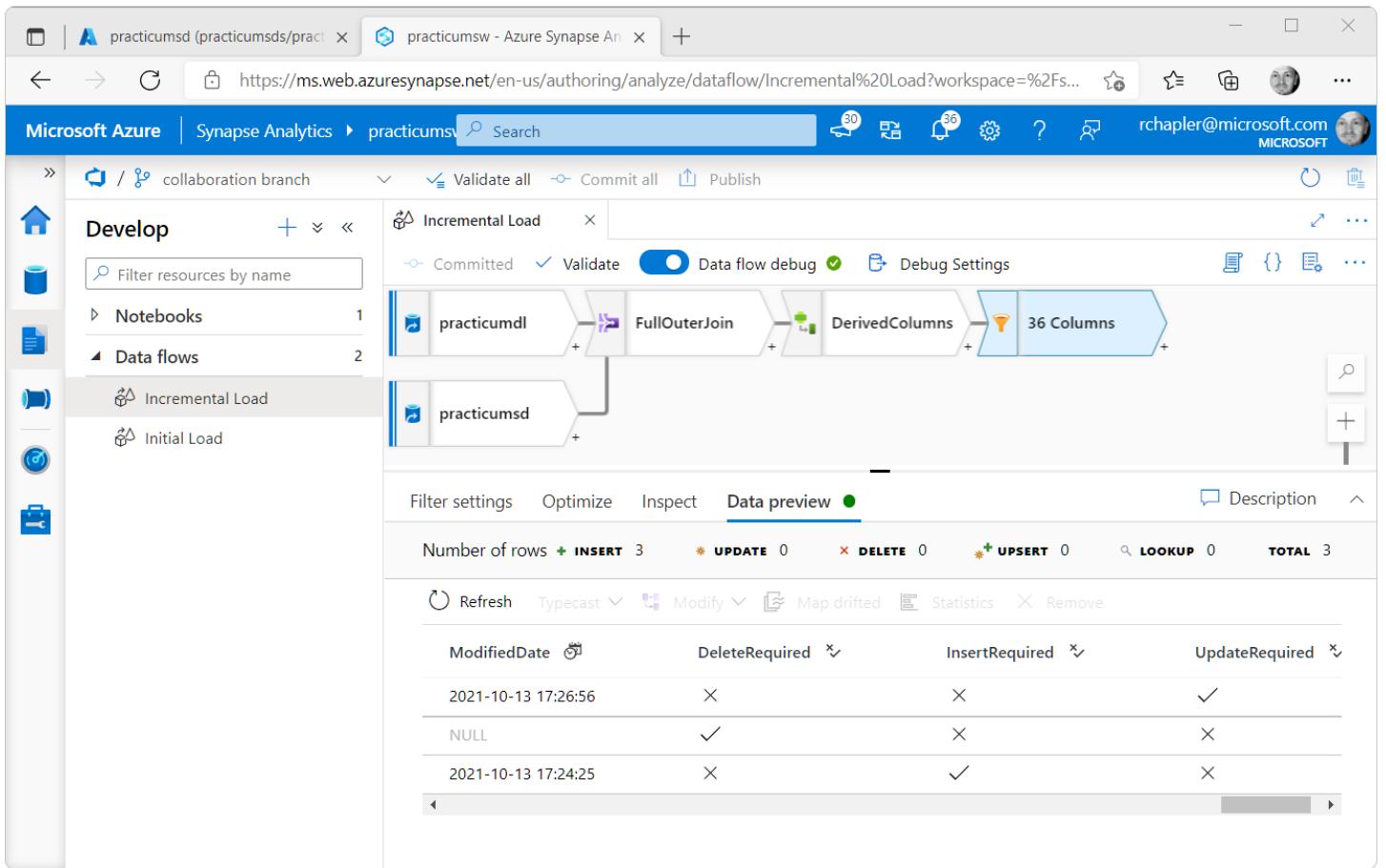
Below the query window, the "Messages" tab shows the message: "Query succeeded: Affected rows: 1". At the bottom of the screen, a yellow bar indicates "Query succeeded | 0s".

Execute the following T-SQL:

```
UPDATE [SalesLT].[Product]
SET [ListPrice] = 999.99, [ModifiedDate] = getdate()
WHERE ProductID=680
```

Confirm Changes

Return to the “**Incremental Load**” data flow and sequentially update with “**Data preview**” > **Refresh**.

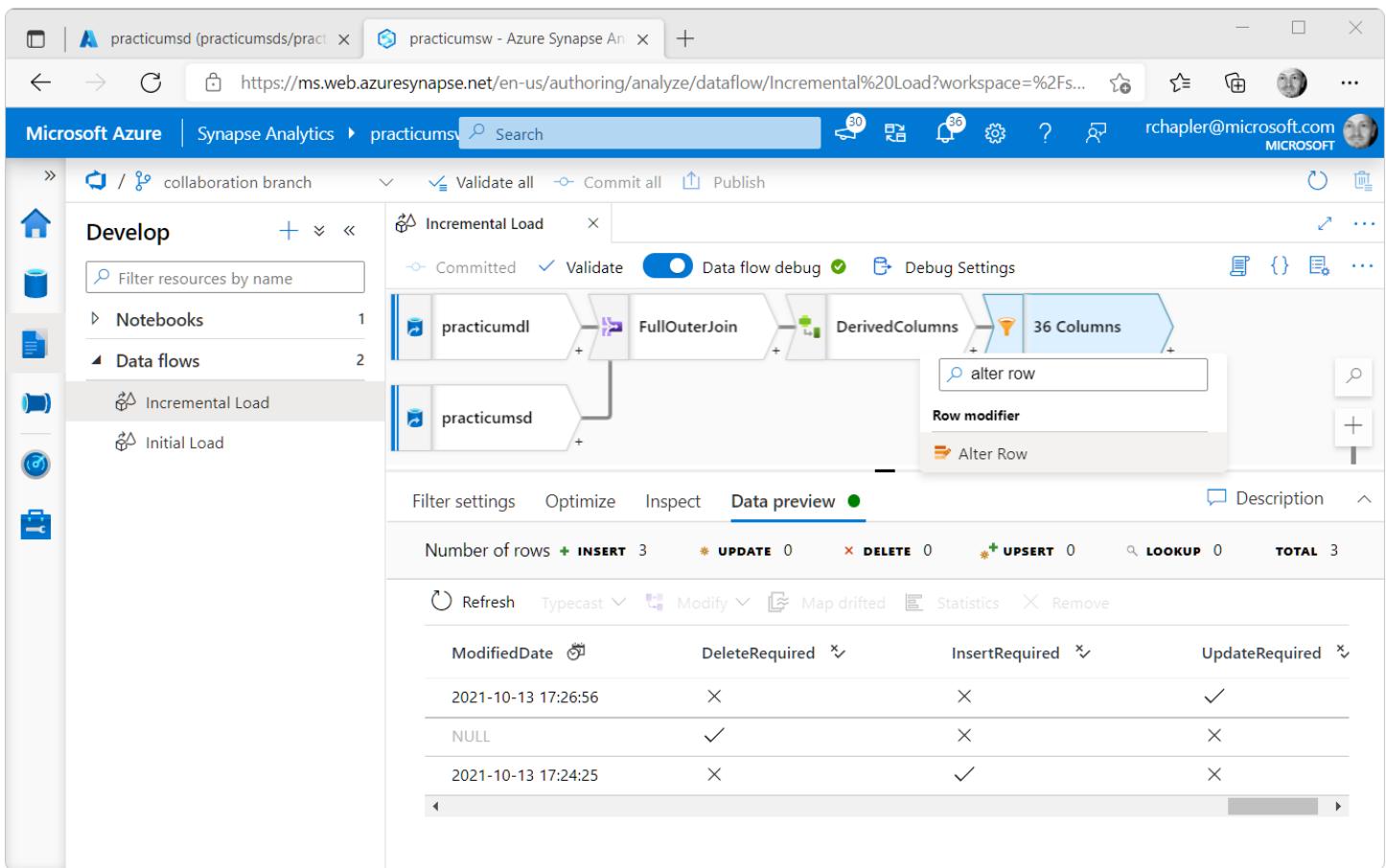


After update, you should see the three actions listed in **Filter > “Data preview”**.

## Finalize Data Flow

### Alter Row

In this step, we will apply delete, insert, and update policies to the rows.



Click the **+** in the bottom right of the SQL source, and then select “**Alter Row**” from the resulting popup list.

On the “**Alter row settings**” tab, enter values for the following items:

---

<b>Output Stream Name</b>	Enter a meaningful name aligned with standards
<b>Incoming Stream</b>	Confirm default selection
<b>Alter Row Conditions</b>	<p><b>"Delete if"</b> DeleteRequired</p> <p><b>"Insert if"</b> InsertRequired</p> <p><b>"Update if"</b> UpdateRequired</p>

---

Navigate to the “Data preview” tab and then click **Refresh**.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. On the left, there's a sidebar with 'Develop' selected, showing a tree view of resources: Notebooks (1), Data flows (2), and Incremental Load. The main area displays a data flow pipeline named 'Incremental Load'. The pipeline starts with a 'FullOuterJoin' between two sources: 'practicumdl' and 'practicumsd'. This is followed by 'DerivedColumns', 'Filter', and a final '36 Columns' sink. Below the pipeline, the 'Data preview' tab is active, showing a preview of the data with the following statistics:

	Number of rows	Operations
INSERT	1	
UPDATE	1	
DELETE	1	
UPsert	0	
Lookup	0	
<b>TOTAL</b>	<b>3</b>	

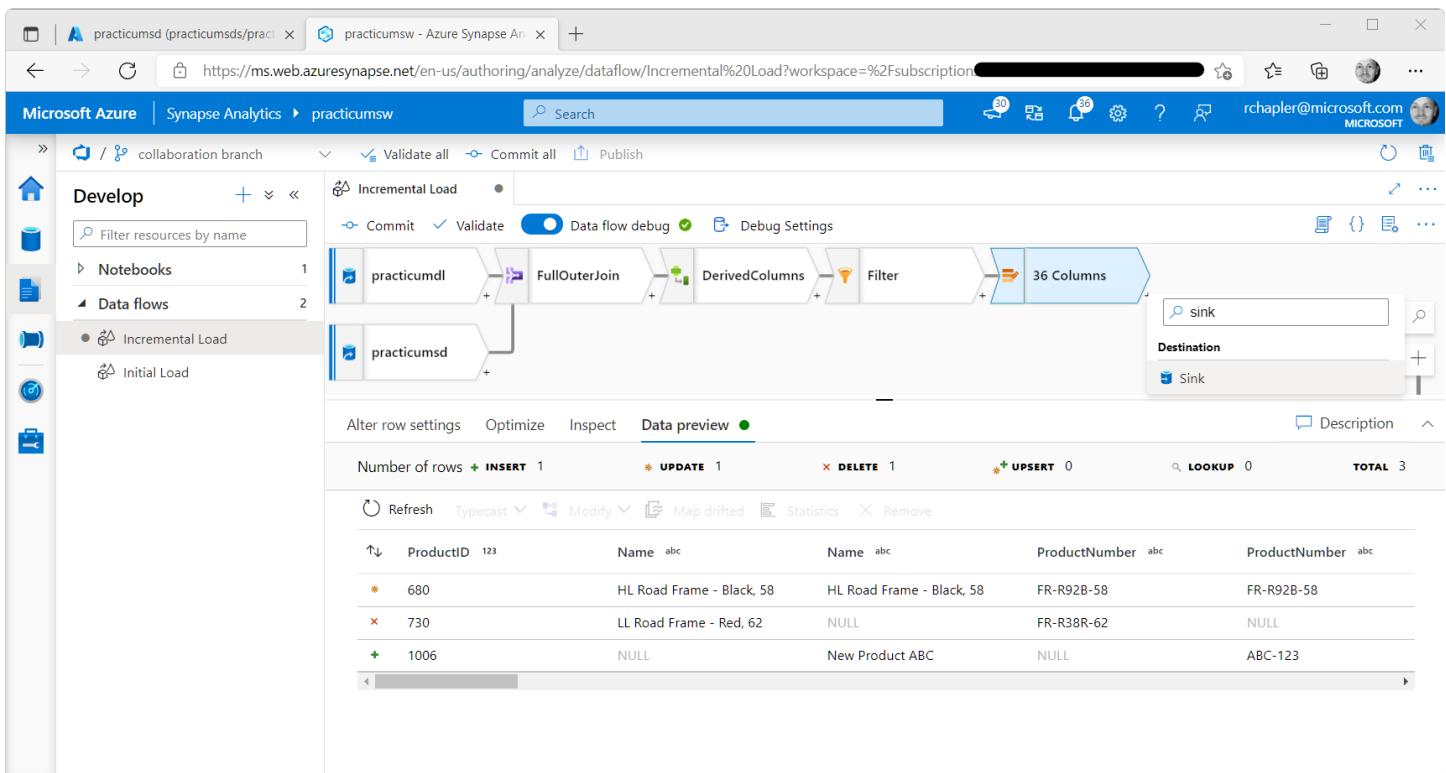
Below the statistics, a table shows the data preview with columns: ProductID, Name, Name, ProductNumber, and ProductNumber. The rows are:

ProductID	Name	Name	ProductNumber	ProductNumber
680	HL Road Frame - Black, 58	HL Road Frame - Black, 58	FR-R92B-58	FR-R92B-58
730	LL Road Frame - Red, 62	NULL	FR-R38R-62	NULL
1006	NULL	New Product ABC	NULL	ABC-123

Click the **Commit** button.

Sink (delta lake)

In this step, we finalize the data flow.



Click the + in the bottom right of the SQL source, and then select **Sink** from the resulting popup list.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade with the 'Sink' tab selected for the 'AlterRow' stage. The 'Sink' tab includes fields for 'Output stream name' (set to 'Sink'), 'Incoming stream' (set to 'AlterRow'), 'Sink type' (set to 'Inline'), 'Inline dataset type' (set to 'Delta'), 'Linked service' (set to 'practicumsw-WorkspaceDefaultStora...'), and 'Options' (checkboxes for 'Allow schema drift' and 'Validate schema').

On the **Sink** tab, enter values for the following items:

**Output Stream Name** Enter a meaningful name aligned with standards

**Sink Type** Select **Inline**

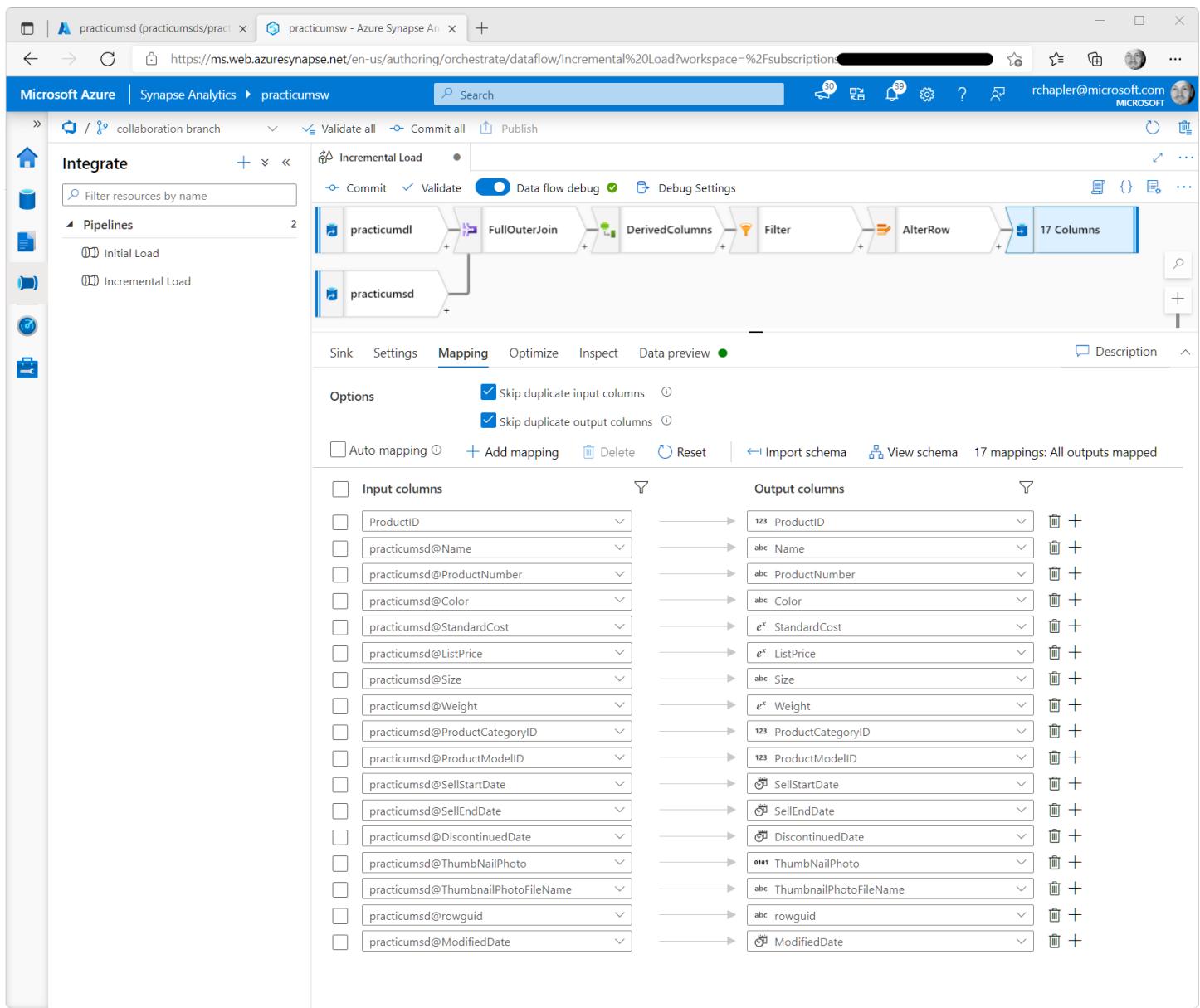
**Inline Dataset Type** Select **Delta**

The screenshot shows the Microsoft Azure Synapse Analytics studio interface. The left sidebar shows 'Integrate' pipelines, with 'Incremental Load' selected. The main area displays a data flow diagram with components like 'FullOuterJoin', 'DerivedColumns', 'Filter', 'AlterRow', and '20 Columns'. The 'Settings' tab is active for the sink, displaying the following configuration:

- Folder path \***: practicumdlc / bronze/northwind/prc
- Compression type**: snappy
- Compression level**: Fastest
- Vacuum**: 0
- Table action**: None (radio button selected)
- Update method**:
  - Allow insert
  - Allow delete
  - Allow upsert
  - Allow update
- Key columns \***:
  - List of columns
  - Custom expressionProductID
- Delta options**:
  - Merge schema
  - Auto compact
  - Optimize write

On the **Settings** tab, enter values for the following items:

<b>File System</b>	Enter <b>practicumdlc</b>
<b>Folder Path</b>	Enter " <b>bronze/northwind/product</b> "
<b>Compression Type</b>	Select <b>snappy</b>
<b>Compression Level</b>	Select <b>Fastest</b>
<b>Update Method</b>	"Allow insert", "Allow delete", and "Allow update" checked
<b>Key Columns</b>	Confirm default selection, "List of columns" and select <b>ProductID</b>
<b>Auto Compact</b>	Checked
<b>Optimize Write</b>	Checked

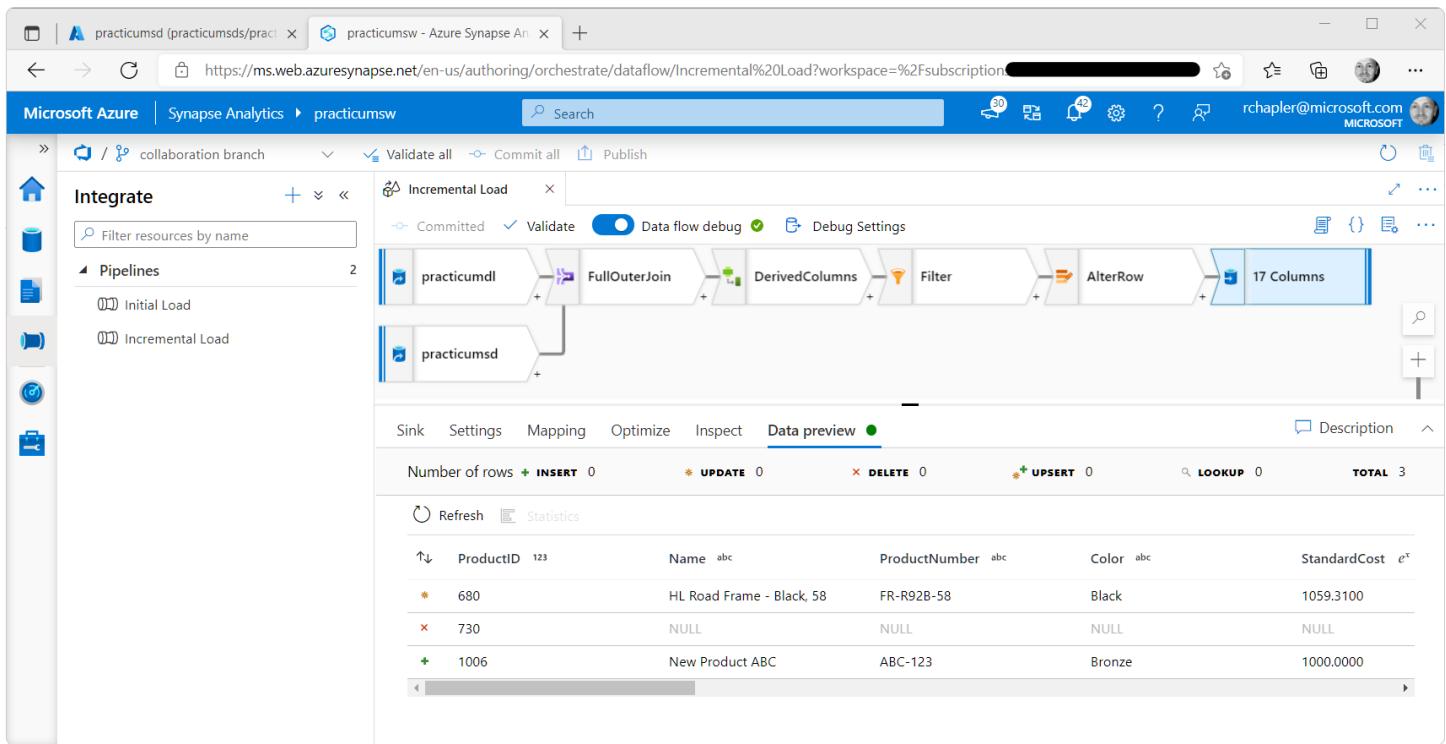


On the **Mapping** tab, un-check “Auto mapping”, click “Import schema” and then Import on the resulting popout.

Review the auto-generated list of columns and remove:

- Rows where “**Input columns**” begin with “**practicumdl...**” (need data changes from the source, not the waterline dataset)
- Derived Columns **DeleteRequired**, **InsertRequired**, and **UpdateRequired**

Navigate to the “**Data preview**” tab and then click **Refresh**.



Click the **Commit** button.

## Create Pipeline

Click the **Integrate** icon in the navigation.

Click the **+** icon just above and to the right of the “**Filter resources by name**” input.

Select “**Pipeline**” from the resulting dropdown.

On the **Properties** popout, enter Name, “**Incremental Load**”.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor interface. On the left, the sidebar displays 'Integrate' and 'Pipelines' sections, with 'Initial Load' and 'Incremental Load' selected. The main workspace shows an 'Incremental Load' activity window. Inside, a 'Data flow' component is selected, labeled 'Data flow1'. The 'Settings' tab is active, showing configuration for the data flow, including 'Run on (Azure IR)', 'Compute type', 'Core count', and 'Logging level'. Buttons for 'Open' and 'New' are visible. The top navigation bar includes tabs for 'Search', 'Commit all', 'Validate all', and 'Publish'.

Search for, and then drag-and-drop a “**Data flow**” component into the activity window.

On the **Settings** tab of the “**Data flow...**” component, select your data flow.

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the **Commit** button.

Confirm Success

Debug Pipeline / Data Flow

Click **Debug** and confirm successful execution.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline interface. On the left, the navigation bar includes 'Microsoft Azure', 'Synapse Analytics', and 'practicumsw'. The main area displays an 'Integrate' pipeline named 'Incremental Load'. Under 'Activities', there is a single 'Data flow' activity named 'Data flow1'. The 'Output' tab is selected, showing a table of pipeline runs. One run is listed: Pipeline run ID: 8524fdc2-bb62-4066-848b-49aac733c988, Name: Data flow1, Type: Data flow, Run start: 2021-10-13T20:30:16.012, Duration: 00:00:34, Status: Succeeded, and Integ: Auto. A message at the bottom right says 'View debug run consumption'.

## Review Produced Files

Click the **Data** icon in the navigation, and then the **Linked** tab.

Expand “**Azure Data Lake Storage Gen2**”, then **practicumsw** to **practicumdlc**.

Click to navigate to “**practicumdlc > bronze > northwind > product**”

The screenshot shows the Microsoft Azure Synapse Analytics Data blade. The navigation bar includes 'Microsoft Azure', 'Synapse Analytics', and 'practicumsw'. The 'Data' section is selected, and the 'Linked' tab is active. The left sidebar shows 'Workspace' and 'practicumsw (Primary - pr...)' under 'Azure Data Lake Storage...'. The main area shows a list of cached items in the 'practicumdlc' storage account, navigating through 'practicumdlc > bronze > northwind > product'. The table lists 158 items, including '\_delta\_log' and several parquet files, with columns for Name, Last Modified, Content Type, and Size.

Name	Last Modified	Content Type	Size
_delta_log	10/13/2021, 6:51:37 AM	Folder	
part-00000-6d4ac397-beee-4b62-9dfc-33458fb11fd6-c000.snappy.parquet	10/13/2021, 6:51:38 AM		147.3 KB
part-00000-82bf94ed-432a-4b12-958e-92309788c2b7-c000.snappy.parquet	10/13/2021, 1:30:25 PM		14.2 KB
part-00001-28ac0035-b7d0-48d7-b5ae-2cfb0e133fb0-c000.snappy.parquet	10/13/2021, 1:30:25 PM		13.9 KB
part-00003-55c643f4-1b5b-4b1d-a69b-f7f557e6f843-c000.snappy.parquet	10/13/2021, 1:30:25 PM		7.5 KB
part-00004-3817f4e7-f32f-45b9-a797-5f94050729e5-c000.snappy.parquet	10/13/2021, 1:30:25 PM		13.8 KB
part-00005-c61bb8c7-177b-4a4d-8b3e-0f5a9f019fc4-c000.snappy.parquet	10/13/2021, 1:30:25 PM		14.9 KB

Review the files produced by delta lake.

## Query Record Count

Navigate to your instance of Synapse, and then click the **Develop** icon in the navigation.

Click the + icon just above and to the right of the “**Filter resources by name**” input.

Select “**Notebook**” from the resulting dropdown.

In your new notebook, select your Apache Spark pool (if required) from the “**Attach to**” dropdown.

The screenshot shows the Microsoft Azure Synapse Analytics Notebook interface. The top navigation bar includes tabs for 'practicumsds/pract' and 'practicumsw - Azure Synapse An'. The URL is https://ms.web.azure-synapse.net/en-us/authoring/analyze/notebooks/Notebook%201?workspace=%2Fsubscriptions%2F... The main area is titled 'Develop' under 'Notebooks'. A search bar says 'Filter resources by name'. Below it, 'Notebook 1' is selected. The notebook content pane displays the following PySpark code:

```
1 %%pyspark
2 df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
3 print(df.count())
4
```

Cell [2] has a green checkmark and the status: "✓ 2 min 13 sec - Apache Spark session started in 1 min 28 sec 353 ms. Command executed in 43 sec 732 ms by rc...". It also shows "Job execution Succeeded" with "Spark 1 executors 4 cores". There are buttons for "View in monitoring" and "Open Spark UI". The output of the cell is "295". At the bottom of the notebook content are buttons for "+ Code" and "+ Markdown".

Run the following code to produce a count of records in the new Delta Lake:

```
%%pyspark
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
print(df.count())
```

*Note: Expected value is 295... we started with 295, deleted one and inserted one.*

## Confirm Delete Example

The screenshot shows the Azure Synapse Analytics interface. On the left, there's a navigation sidebar with icons for Home, Notebooks, Data flows, and others. The main area is titled 'Notebook 1' and shows a code cell with the following Python code:

```
1 %%pyspark
2 df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
3 display(df.where("ProductID == 730"))
4
```

Below the code, a message indicates a successful execution: [5] ✓ 11 sec - Command executed in 11 sec 23 ms by rchaper on 2:01:43 PM, 10/13/21. It also shows 'Job execution Succeeded' with 'Spark 1 executors 4 cores'. There are buttons for '+ Code' and '+ Markdown' at the bottom of the code cell.

Run the following code to confirm our update example:

```
%%pyspark
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
display(df.where("ProductID == 730"))
```

## Confirm Insert Example

The screenshot shows the same Azure Synapse Analytics interface. The notebook content remains the same as the previous example. After running the code, the interface displays a table view of the results. The table has columns: ProductID, Name, ProductNumber, and Color. One row is visible:

ProductID	Name	ProductNumber	Color
1006	New Product ABC	ABC-123	Bronze

Run the following code to confirm our update example:

```
%pyspark  
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')  
display(df.where("ProductID == 1006"))
```

## Confirm Update Example

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar has sections for 'Develop' (Notebooks, Data flows), 'Data engineering' (pipelines, triggers, scheduled datasets), and 'Machine learning' (ML datasets, ML pipelines). The main area shows 'Notebook 1' with the following code and execution results:

```
1 %%pyspark  
2 df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')  
3 display(df.where("ProductID == 680"))  
4
```

[3] ✓ 11 sec - Command executed in 10 sec 642 ms by rchapler on 1:54:40 PM, 10/13/21

> Job execution Succeeded Spark 1 executors 4 cores View in monitoring Open Spark UI

View	Table	Chart	Export results		
ProductID	Name	ProductNumber	Color	StandardCost	ListPrice
680	HL Road Frame - Black, 58	FR-R92B-58	Black	1059.3100	999.9900

Run the following code to confirm our update example:

```
%pyspark  
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')  
display(df.where("ProductID == 680"))
```

*Objective Complete... congratulations!*

## Objective - Databricks + Data Lake | Mount Data Lake

Follow the instructions in this section to establish connection to data in Azure Data Lake Storage.

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Data Lake (with “Storage Blob Data Reader” permissions set for your Application Registration)
- Databricks (with cluster, notebook, and secret scope)

### Step 2: Prepare Logic

Navigate to Databricks and the **practicumdbn** notebook.

Add a new cell and paste the following code:

```
configs = {"fs.azure.account.auth.type": "OAuth",
           "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
           "fs.azure.account.oauth2.client.id": dbutils.secrets.get( scope="practicumdbss", key="practicumar-clientid"),
           "fs.azure.account.oauth2.client.secret": dbutils.secrets.get( scope="practicumdbss", key="practicumar-clientsecret"),
           "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/" + dbutils.secrets.get(
scope="practicumdbss", key="practicumar-tenantid") + "/oauth2/token" }

adlsAccount = "practicumdl"
adlsContainer = "practicumdlc"
adlsFolder = ""
mountPoint = "/mnt/practicum"

if not any(mount.mountPoint == mountPoint for mount in dbutils.fs.mounts()):
    dbutils.fs.mount( source = "abfss://" + adlsContainer + "@" + adlsAccount + ".dfs.core.windows.net/" + adlsFolder,
    mount_point = mountPoint, extra_configs = configs )
```

Some callouts...

- **practicumdbss** ... refers to the Secret Scope
- **myClientId** ... refers to the Key Vault secret containing the “Application (client) ID”
- **myClientSecret** ... refers to the Key Vault secret containing the “Client Secret”
- **myTenantId** ... refers to the Key Vault secret containing the “Directory (tenant) ID”
- **practicumdl** ... refers to your Data Lake
- **practicumdlc** ... refers to the Data Lake Container
- **adlsFolder** ... placeholder / syntax for inclusion of a folder (null because it is not applicable in this instance)

Run “**Cmd 1**”.

The screenshot shows a Microsoft Azure Databricks notebook titled "practicumdbn - Databricks". The notebook interface includes a left sidebar with various icons for file operations, a top navigation bar with tabs for "practicumdb - Microsoft Azure" and "practicumdbn - Databricks", and a top right corner showing the user's email "rchapler@microsoft.com". The main workspace contains a code cell labeled "Cmd 1" containing Python code for mounting an ADLS Gen2 storage account. The code uses dbutils secrets to get OAuth2 client credentials and mounts the storage account at "/mnt/practicum". A note indicates that the command took 26.98 seconds and was run by rchapler@microsoft.com on 10/4/2021 at 10:27:15 AM. Below the code cell, there is a message "Shift+Enter to run".

```
1 configs = {"fs.azure.account.auth.type": "OAuth",
2             "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
3             "fs.azure.account.oauth2.client.id": dbutils.secrets.get(scope="practicumdbss", key="practicumar-clientid"),
4             "fs.azure.account.oauth2.client.secret": dbutils.secrets.get(scope="practicumdbss", key="practicumar-clientsecret"),
5             "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/" +
6             dbutils.secrets.get(scope="practicumdbss", key="practicumar-tenantid") + "/oauth2/token"}
7
8 adlsAccount = "practicumdl"
9 adlsContainer = "practicumdlc"
10 adlsFolder = ""
11 mountPoint = "/mnt/practicum"
12
13 if not any(mount.mountPoint == mountPoint for mount in dbutils.fs.mounts()):
14     dbutils.fs.mount( source = "abfss://" + adlsContainer + "@" + adlsAccount + ".dfs.core.windows.net/" + adlsFolder,
15     mount_point = mountPoint, extra_configs = configs )
```

## Quick Idea...

Consider organizing mount locations by associating them with a Databricks database entity; code example:

```
%sql
CREATE DATABASE myDatabase LOCATION "/mnt/practicum/myDatabase"
```

## Step 3: Confirm Success

Proof 1 ... Sample Files at Mount Point

*Note: Running this logic requires "Storage Blob Data Reader" permissions on your data lake.*

Add a cell below "**Cmd 1**". In cell "**Cmd 2**", paste and run the following code:

```
%fs
ls "/mnt/practicum"
```

practicumdbn - Microsoft Azure

practicumdbn - Databricks

Microsoft Azure | Databricks

practicumdbn (Python)

practicumdbc

Cmd 2

```
1 %fs
2 ls "mnt/practicum"
```

	path	name	size
1	dbfs:/mnt/practicum/sample1.csv	sample1.csv	502
2	dbfs:/mnt/practicum/sample2.csv	sample2.csv	56890
3	dbfs:/mnt/practicum/sample3.csv	sample3.csv	723

Showing all 3 rows.

You should see your sample files at the mount point.

## Proof 2 ... Schema and Data

In your Databricks notebook, add a new cell, then paste and run the following code:

```
df = spark.read.csv("dbfs:/mnt/practicum/sample1.csv")
df.printSchema()
df.show()
```

```
practicumdbn (Python)
practicumdbc

Cmd 3
1 df = spark.read.csv("dbfs:/mnt/practicum/sample1.csv")
2 df.printSchema()
3 df.show()

▶ (2) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [c0: string, c1: string ... 11 more fields]

root
|-- _c0: string (nullable = true)
|-- _c1: string (nullable = true)
|-- _c2: string (nullable = true)
|-- _c3: string (nullable = true)
|-- _c4: string (nullable = true)
|-- _c5: string (nullable = true)
|-- _c6: string (nullable = true)
|-- _c7: string (nullable = true)
|-- _c8: string (nullable = true)
|-- _c9: string (nullable = true)
|-- _c10: string (nullable = true)
|-- _c11: string (nullable = true)
|-- _c12: string (nullable = true)

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| _c0 | _c1 | _c2 | _c3 | _c4 | _c5 | _c6 | _c7 | _c8 | _c9 | _c10 | _c11 | _c12 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|Month| "Average"|"2005"|"2006"|"2007"|"2008"|"2009"|"2010"|"2011"|"2012"|"2013"|"2014"|"2015"|
| May | 0.1| 0| 0| 1| 1| 0| 0| 0| 2| 0| 0| 0 |
| Jun | 0.5| 2| 1| 1| 0| 0| 1| 1| 2| 2| 0| 1 |
| Jul | 0.7| 5| 1| 1| 2| 0| 1| 3| 0| 2| 2| 1 |
| Aug | 2.3| 6| 3| 2| 4| 4| 4| 7| 8| 2| 2| 3 |
| Sep | 3.5| 6| 4| 7| 4| 2| 8| 5| 2| 5| 2| 5 |
| Oct | 2.0| 8| 0| 1| 3| 2| 5| 1| 5| 2| 3| 0 |
| Nov | 0.5| 3| 0| 0| 1| 1| 0| 1| 0| 1| 0| 1 |
| Dec | 0.0| 1| 0| 1| 0| 0| 0| 0| 0| 0| 0| 1 |

Command took 5.57 seconds -- by rchapler@microsoft.com at 1/26/2021, 9:48:04 AM on practicumdbc
```

Shift+Enter to run [shortcuts](#)

Some observations:

- Schema ... the schema is not defined, so the resulting interpretation calls out each field as data type string.
- Resultset ... data formatting {e.g., first row as headers, double quotes, etc.} has not been applied.

*Objective Complete... congratulations!*

## Objective - Databricks | Localize Mounted Data

Follow the instructions in this section to **localize data from a Data Lake mount**.

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Data Lake
- Databricks

### Step 2: Prepare Logic

#### Why Localize?

*Databricks File System (DBFS) is a distributed file system mounted into a Databricks workspace and available on Databricks clusters. DBFS is an abstraction on top of scalable object storage and offers the following **benefits**:*

- Allows you to mount storage objects so that you can **seamlessly access data without requiring credentials**.
- Allows you to **interact with object storage using directory and file semantics instead of storage URLs**.
- Persists files to object storage, so you **will not lose data after you terminate a cluster**.

[Databricks File System \(DBFS\) — Databricks Documentation](#) | December 16, 2020

Though it is not listed here, many customers refer to the following benefits when justifying the decision to localize data from Azure Data Lake Storage to Azure Databricks:

- Performance Gains
- Decreased Compute Cost
- Leverage Benefits of Partitioning

#### The Logic

Navigate to Databricks and the **practicumdbn** notebook. Add a new cell and paste the following code:

```
import uuid
myUUID = uuid.uuid4().hex

spark.read.csv( "/mnt/practicum/sample1.csv" ).write.format( "delta" ).save( "/mnt/" + myUUID )
spark.sql( "CREATE TABLE IF NOT EXISTS myDeltaTable_fromADLS USING DELTA LOCATION '/mnt/" + myUUID + "/" )
```

The screenshot shows the Microsoft Azure Databricks workspace interface. On the left, there is a navigation sidebar with icons for Home, Workspace, Recents, Data, and Clusters. The 'Workspace' icon is highlighted with a red border. The main area displays a Python notebook titled 'practicumdbn (Python)'. A tab bar at the top shows 'practicumdb - Microsoft Azure' and 'practicumdb - Databricks'. The notebook contains the following code:

```
1 import uuid
2 myUUID = uuid.uuid4().hex
3
4 spark.read.csv( "/mnt/practicum/sample1.csv" ).write.format( "delta" ).save( "/mnt/" + myUUID )
5 spark.sql( "CREATE TABLE IF NOT EXISTS myDeltaTable_fromADLS USING DELTA LOCATION '/mnt/" + myUUID + "/" )
```

The output of the code execution is shown below the code editor:

```
▶ (5) Spark Jobs
Out[3]: DataFrame[]

Command took 13.51 seconds -- by rchapler@microsoft.com at 1/26/2021, 10:26:28 AM on practicumdbc
```

At the bottom of the code editor, there is a note: "Shift+Enter to run [shortcuts](#)".

Click on **Data** in the navigation. Click on the **default** database, then the **mydeltatable\_fromadls** table.

The screenshot shows the Microsoft Azure Databricks interface. The left sidebar has a dark theme with icons for Home, Workspace, Recents, Data (selected), Clusters, Jobs, Models, and Search. The main area is titled "Data". Under "Databases", "default" is selected. Under "Tables", "mydeltatable\_fromadls" is selected. The top navigation bar shows "practicumdb - Microsoft Azure" and "mydeltatable\_fromadls - Databricks". The URL is https://adb-2154823451042175.15.azuredatabricks.net/?o=2154823451042175#table/default/mydeltatable\_fromadls. The page title is "Table: mydeltatable\_fromadls". The table details show it was created on 2021-01-26 at 18:26:29 and last modified on 2021-01-26 at 18:26:32. It has no partition columns, 1 file, and a size of 3.54 kB. The schema table has columns: col\_name, data\_type, and comment. The sample data table has columns: \_c0, \_c1, \_c2, \_c3, \_c4, and \_c5.

Table: mydeltatable\_fromadls

mydeltatable\_fromadls

Description:  
Created at: 2021-01-26 18:26:29  
Last modified: 2021-01-26 18:26:32  
Partition columns:  
Number of files: 1  
Size: 3.54 kB

Schema:

	col_name	data_type	comment
1	_c0	string	
2	_c1	string	
3	_c2	string	
4	_c3	string	
5	_c4	string	
6	_c5	string	
7	_c6	string	
8	_c7	string	

Showing all 16 rows.

Sample Data:

	_c0	_c1	_c2	_c3	_c4	_c5
1	Month	"Average"	"2005"	"2006"	"2007"	"2008"
2	May	0.1	0	0	1	1
3	Jun	0.5	2	1	1	0
4	Jul	0.7	5	1	1	2
5	Aug	2.3	6	3	2	4
6	Sep	3.5	6	4	7	4
7	Oct	2.0	8	0	1	3
8	Nov	0.5	3	0	0	1

Showing all 9 rows.

Observation: Previously noted schema and resultset challenges remain present in the newly created table.

*Objective Complete... congratulations!*

## Objective - Databricks | Source from APIs

Follow the instructions in this section to [source data from a REST API](#).

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Databricks
- Key Vault

### Step 2: Stage Resources

For this exercise, we will use “**Current Weather**” data from <https://openweathermap.org/api>.

The screenshot shows the OpenWeatherMap API landing page at <https://openweathermap.org/api>. The page features a navigation bar with links for Guide, API, Pricing, Maps, FAQ, Partners, Blog, Marketplace, Sign in, and Support. Below the navigation bar, there's a search bar labeled "Weather in your city". The main content area has three main sections: "Weather API", "Current & Forecast weather data collection", "Current Weather Data", "Hourly Forecast 4 days", and "One Call API". Each section includes a "API doc" button and a "Subscribe" button. The "Current Weather Data" section lists benefits like access to over 200,000 cities and various data formats. The "Hourly Forecast 4 days" section highlights a 4-day forecast with 96 timestamps. The "One Call API" section details a single API call for current, forecast, and historical data.

Please [sign up](#) and use our fast and easy-to-work weather APIs for free. Look at our [monthly subscriptions](#) for more options rather than the Free account that we provide you. Read [How to start](#) first and enjoy using our powerful weather APIs.

## Current & Forecast weather data collection

### Current Weather Data

[API doc](#) [Subscribe](#)

- Access current weather data for any location including over 200,000 cities
- We collect and process weather data from different sources such as global and local weather models, satellites, radars and vast network of weather stations
- JSON, XML, and HTML formats
- Available for both Free and paid subscriptions

### Hourly Forecast 4 days

[API doc](#) [Subscribe](#)

- Hourly forecast is available for 4 days
- Forecast weather data for 96 timestamps
- Higher geographic accuracy
- JSON and XML formats
- Available for Developer, Professional and Enterprise accounts

### One Call API

[API doc](#) [Subscribe](#)

- Make one API call and get current, forecast and historical weather data
- **Minute forecast** for 1 hour
- **Hourly forecast** for 48 hours
- **Daily forecast** for 7 days
- **Historical data** for 5 previous days
- **National weather alerts**
- JSON format
- Available for both Free and paid subscriptions

I chose this as my sample source because they have a Free subscription, their data is relatable, and it is easy to use.

Navigate to <https://openweathermap.org/price>.

The screenshot shows the OpenWeatherMap Pricing page. At the top, there's a navigation bar with links for Guide, API, Pricing, Maps, FAQ, Partners, Blog, Marketplace, Sign In, and Support. Below the navigation is a search bar with the placeholder "Weather in your city". The main content area has a heading "Pricing" and a sub-heading "Current weather and forecasts collection". A table displays five subscription plans:

Free	Startup 40 USD / month	Developer 180 USD / month	Professional 470 USD / month	Enterprise 2,000 USD / month
<a href="#">Get API key</a>	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>
60 calls/minute <b>1,000,000 calls/month</b>	600 calls/minute 10,000,000 calls/month	3,000 calls/minute 100,000,000 calls/month	30,000 calls/minute 1,000,000,000 calls/month	200,000 calls/minute 5,000,000,000 calls/month
<b>Current Weather</b>	Current Weather	Current Weather	Current Weather	Current Weather

Click the “Get API key” button in the **Free** section and complete the “Create a New Account” process.

When you’re successfully created an account, making an API call for “**Current Weather**” data is as simple as:

```
api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}
```

The screenshot shows a browser window with the URL `api.openweathermap.org/data/2.5/weather?q=redmond&appid=...`. The page title is "Not secure". The response is a JSON object:

```
{"coord":{"lon":-122.1215,"lat":47.674}, "weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04d"}], "base":"stations", "main": {"temp":279.53,"feels_like":277.4,"temp_min":278.15,"temp_max":280.93,"pressure":1007,"humidity":65,"sea_level":1007,"grnd_level":1006}, "visibility":10000,"wind": {"speed":0.27,"deg":241}, "clouds": {"all":100}, "dt":1611870000, "sys": {"type":3,"id":2010401,"country":"US","sunrise":1611848385,"sunset":1611882185}, "timezone":-28800, "id":5808079, "name": "Redmond", "cod":200 }
```

### Step 3: Add Secret to Key Vault

Add your API Key to your instance of Key Vault.

The screenshot shows the Microsoft Azure portal interface for creating a secret. The page title is "Create a secret - Microsoft Azure". The URL in the address bar is <https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/91e9fddc-...>. The top navigation bar includes "Microsoft Azure (Preview)", a search bar, and user information for "rchapler@microsoft.com MICROSOFT (MICROSOFT.ONM...)".

The main form fields are:

- Upload options: Manual
- Name \* ⓘ: myOpenWeatherAPIKey
- Value \* ⓘ: (redacted)
- Content type (optional): (empty)
- Set activation date? ⓘ: (unchecked)
- Set expiration date? ⓘ: (unchecked)
- Enabled?: Yes (selected)

A blue "Create" button is located at the bottom left of the form.

## Step 4: Prepare Logic

Navigate to Databricks and the **practicumdbn** notebook. Add a new cell and paste the following code:

```
import requests
response = requests.get( 'http://api.openweathermap.org/data/2.5/weather?q=redmond&appid=' + dbutils.secrets.get(
scope="practicumdbss", key="myOpenWeatherAPIKey" ) )

df = spark.read.option( 'multiline', "true" ).json( sc.parallelize( [response.text] ) )
```

Run the cell.

The screenshot shows the Microsoft Azure Databricks workspace. On the left is a sidebar with icons for Home, Workspace, Recents, Data, Clusters, Jobs, Models, and Search. The main area has a header 'practicumdbn - Microsoft Azure' and 'practicumdbn - Databricks'. Below the header is a toolbar with back, forward, search, and other navigation icons. The main content area is a Python notebook titled 'practicumdbn (Python)'. It contains the following code:

```
1 import requests
2 response = requests.get( 'http://api.openweathermap.org/data/2.5/weather?q=redmond&appid=' + dbutils.secrets.get(
3     scope="practicumdbss", key="myOpenWeatherAPIKey" ) )
4 df = spark.read.option( 'multiline', "true" ).json( sc.parallelize( [response.text] ) )
```

Below the code, there is a detailed schema expansion for the DataFrame 'df':

- (1) Spark Jobs
- df: pyspark.sql.dataframe.DataFrame
  - base: string
  - clouds: struct
    - all: long
    - cod: long
  - coord: struct
    - lat: double
    - lon: double
  - dt: long
  - id: long
  - main: struct
    - feels\_like: double
    - grnd\_level: long
    - humidity: long
    - pressure: long
    - sea\_level: long

Another Python example (including POST, headers, and body):

```
import json
stringJSON = '{"data": [{"1212":0,"1227":0,..."ZZ9":0}]}'
theJSON = json.loads(stringJSON)
response = requests.post(
    'http://eedfb2cf-deb1-4260-971d-d7a6c308d9b2.eastus.azurecontainer.io/score',
    headers={'CONTENT-TYPE': 'application/json'},
    json = theJSON
)
response.json()
```

*Objective Complete... congratulations!*

## Objective - Databricks | Batch Upsert Data

Follow the instructions in this section to **provide for recurring upsert of data**.

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Data Lake
- Databricks

### Step 2: CREATE TABLE

Navigate to Databricks and the **practicumdbn** notebook.

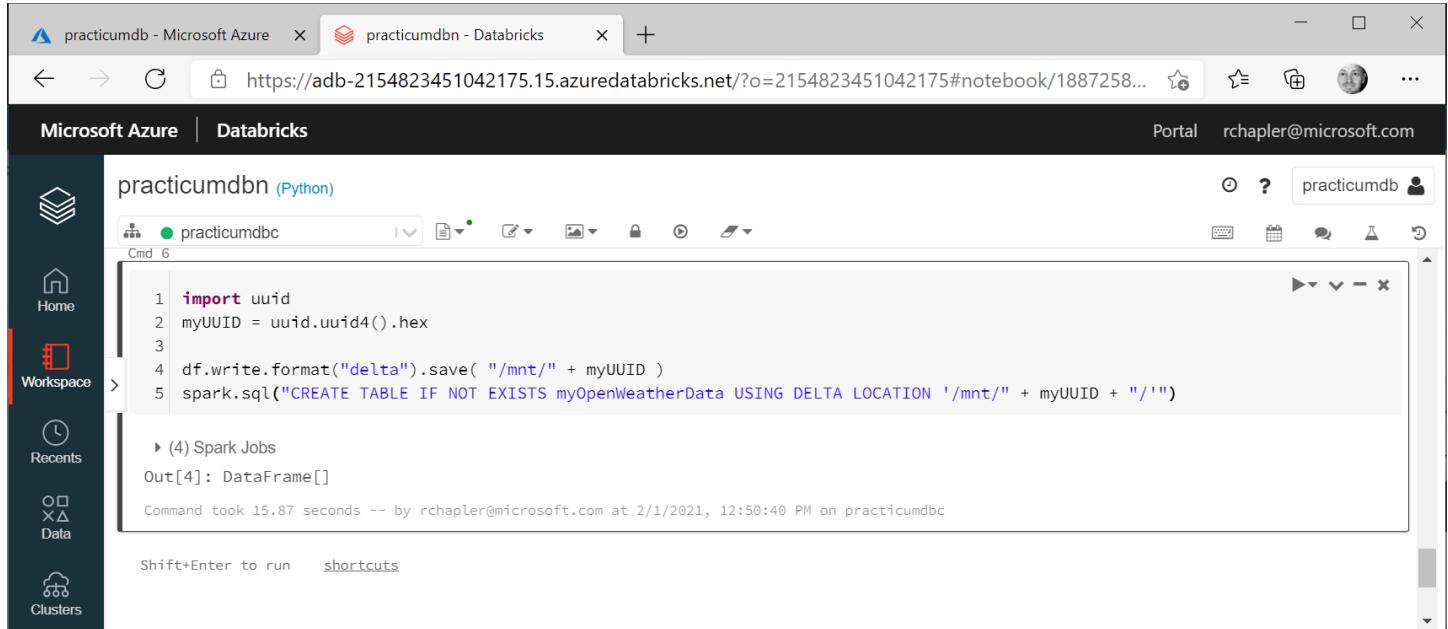
Add a new cell and paste the following code:

```
import uuid
myUUID = uuid.uuid4().hex

df.write.format("delta").save( "/mnt/" + myUUID )
spark.sql("CREATE TABLE IF NOT EXISTS myOpenWeatherData USING DELTA LOCATION '/mnt/" + myUUID + "'")
```

Notes:

- We are using dataframe **df** created in [Objective 3 ... The Logic](#)
- We are using format **delta** to provide for future UPSERT operations.



```
import uuid
myUUID = uuid.uuid4().hex

df.write.format("delta").save( "/mnt/" + myUUID )
spark.sql("CREATE TABLE IF NOT EXISTS myOpenWeatherData USING DELTA LOCATION '/mnt/" + myUUID + "'")
```

(4) Spark Jobs  
Out[4]: DataFrame[]  
Command took 15.87 seconds -- by rchapler@microsoft.com at 2/1/2021, 12:50:40 PM on practicumdbc

### Step 2: MERGE INTO

Navigate to Databricks and the **practicumdbn** notebook. Add a new cell and paste the following code:

```
import requests
import uuid

response = requests.get( 'http://api.openweathermap.org/data/2.5/weather?q=redmond&appid=' + dbutils.secrets.get(
    scope="practicumdbss", key="myOpenWeatherAPIKey" ) )
df = spark.read.option( 'multiline', "true" ).json( sc.parallelize( [response.text] ) )
```

```

myUUID = uuid.uuid4().hex
df.write.format("delta").save( "/mnt/" + myUUID )
spark.sql("DROP TABLE myOpenWeatherData_updates")
spark.sql("CREATE TABLE myOpenWeatherData_updates USING DELTA LOCATION '/mnt/" + myUUID + "'")

mySQL = "MERGE INTO myOpenWeatherData existing USING myOpenWeatherData_updates updates ON existing.id = updates.id AND
existing.dt = updates.dt"
mySQL += " WHEN MATCHED THEN UPDATE SET existing.main = updates.main"
mySQL += " WHEN NOT MATCHED THEN INSERT (id, dt, main) VALUES (id, dt, main)"

print( mySQL )

spark.sql( mySQL )

```

Run the cell.

The screenshot shows the Microsoft Azure Databricks interface. On the left, there's a sidebar with icons for Home, Workspace (selected), Recents, Data, Clusters, Jobs, Models, and Search. The main area has tabs for 'practicumdb - Microsoft Azure' and 'practicumdbn - Databricks'. The 'practicumdbn - Databricks' tab is active, showing a Python notebook titled 'practicumdbn (Python)'. The notebook contains the provided code for merging weather data. Below the code, the output shows the command was run by 'rchapler@microsoft.com' at 2/1/2021, 1:00:44 PM on cluster 'practicumdbc'. The output also includes a warning about the 'MERGE' command.

```

import requests
import uuid

response = requests.get( 'http://api.openweathermap.org/data/2.5/weather?q=redmond&appid=' + dbutils.secrets.get(
scope="practicumdbss", key="myOpenWeatherAPIKey" ) )
df = spark.read.option( 'multiline', "true" ).json( sc.parallelize( [response.text] ) )

myUUID = uuid.uuid4().hex
df.write.format("delta").save( "/mnt/" + myUUID )
spark.sql("DROP TABLE myOpenWeatherData_updates")
spark.sql("CREATE TABLE myOpenWeatherData_updates USING DELTA LOCATION '/mnt/" + myUUID + "'")

mySQL = "MERGE INTO myOpenWeatherData existing USING myOpenWeatherData_updates updates ON existing.id = updates.id AND
existing.dt = updates.dt"
mySQL += " WHEN MATCHED THEN UPDATE SET existing.main = updates.main"
mySQL += " WHEN NOT MATCHED THEN INSERT (id, dt, main) VALUES (id, dt, main)"

print( mySQL )

spark.sql( mySQL )

```

(14) Spark Jobs  
df: pyspark.sql.dataframe.DataFrame = [base: string, clouds: struct ... 12 more fields]  
MERGE INTO myOpenWeatherData existing USING myOpenWeatherData\_updates updates ON existing.id = updates.id AND existing.dt = up  
dates.dt WHEN MATCHED THEN UPDATE SET existing.main = updates.main WHEN NOT MATCHED THEN INSERT (id, dt, main) VALUES (id, dt,  
main)  
Out[8]: DataFrame[]  
Command took 16.50 seconds -- by rchapler@microsoft.com at 2/1/2021, 1:00:44 PM on practicumdbc

Shift+Enter to run [shortcuts](#)

### Step 3: Confirm Success

Navigate to Databricks and the **practicumdbn** notebook. Add a new cell and paste the following code:

```
%sql
SELECT * FROM myOpenWeatherData
```

Run the cell.

The screenshot shows the Microsoft Azure Databricks workspace. On the left is a sidebar with icons for Home, Workspace (selected), Recents, Data, Clusters, Jobs, Models, and Search. The main area has two tabs: "practicumdb - Microsoft Azure" and "practicumdbn - Databricks". The "practicumdbn - Databricks" tab is active, displaying a Python notebook cell and a command line interface (CLI) window.

**Python Notebook Cell:**

```
MERGE INTO myOpenWeatherData existing USING myOpenWeatherData_updates updates ON existing.id = updates.id AND existing.dt = updates.dt WHEN MATCHED THEN UPDATE SET existing.main = updates.main WHEN NOT MATCHED THEN INSERT (id, dt, main) VALUES (id, dt, main)
Out[8]: DataFrame[]
```

Command took 16.50 seconds -- by rchapler@microsoft.com at 2/1/2021, 1:00:44 PM on practicumdbc

**CLI Window (Cmd 8):**

```
1 %sql
2 SELECT * FROM myOpenWeatherData
```

▶ (3) Spark Jobs

	base	clouds	cod	coord	dt	id	main
1	stations	►{"all": 100}	200	►{"lat": 47.674, "lon": -122.1215}	1612211690	5808079	►{"feels_like": 280.49, "grnd_level": 1004, "h282.59, "temp_min": 281.48}
2	null	null	null	null	1612212650	5808079	►{"feels_like": 279.62, "grnd_level": 1005, "h282.59, "temp_min": 281.48}

Showing all 2 rows.

Command took 1.18 seconds -- by rchapler@microsoft.com at 2/1/2021, 1:22:13 PM on practicumdbc

Shift+Enter to run [shortcuts](#)

Note that the row created using MERGE INTO has a limited number of populated cells since only some of the possible fields were included in WHEN NOT MATCHED THEN INSERT (id, dt, main)... we can, of course, expand this to include all fields about which we care.

*Objective Complete... congratulations!*

## Objective - Data Explorer | Query from On-Prem

Follow the instructions in this section to **query from an on-prem client**.

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Azure CLI
- Data Explorer
- Postman

### Step 2: Get Token

In a Command Prompt window (with [Azure CLI](#) installed), execute the following command: `az login`

Use the resulting browser window to provide credentials.

In a Command Prompt window (with [Azure CLI](#) installed), execute the following command: `az account get-access-token --resource https://practicumdec.westus2.kusto.windows.net --query accessToken --output tsv`

Capture the returned token for use later in this section.



```
C:\Users\rchapler>az account get-access-token --resource https://practicumdec.westus2.kusto.windows.net --query accessToken --output tsv
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Im5PbzNaRHJPRFhFSzFqS1doWHNsSFJfs1hFZyIsImtpZCI6Im5PbzNaRHJPRFhFSzFqS1doWHNsSFJfs1hFZyJ9.eyJhdWQ
O1JodHlwczovL3ByYWNoaWN1bwRlYy53ZXN0dXMyLmt1c3RvLndpbmRvd3Mubm0iIwiAxNzIjoiaHR0cHM6Ly9zdHMu
d2luZG93cy5uZXQvNzJm0tG4YmYtODzms00MWFmlTkxyWnItM
mQ3Y2QwMTFkYjQ3LyIsImlhdcI6MTYxODk1NjI1NCwibmJmIjoNxNjE40TU2MjU0LCJleHAiOjE2MTg5NjAxNTQsImFjciI6IjEiLCJhaW8iOjBVVF
BdS84VEFBQUF5Y1NLemdTcmVRNm
Frcmh1Ly8xU3lxZmVXVUtyOEUwVh5SlgwUXZ3cGhwSVNOSFhuaXYrc25ROEp6YT
NVeWpTUw5QOU1J
RmdTeUpIMFBWW1M301Zydz09IiwiYWlyIjpbi
nJzYSi
sIm1mYSJdLCJhcHBpZCI
6IJA0YjA3NzK1LThkZGItNDYxYS1iYmV1LT
AyZj1lMWJmN2I0NiIsImFwcG1kYW NyIjoimCIsImRldmljZw1kIjoiMjc3MzgzMWEtMTQ2NS00N2NhLThiODItMDg5NWM1MGN1MwV1Iiwi
ZmfTa
wx5X25hbWU1o1JdaG
fwbGyIiwiZ2122W5fbmftzs16I1jpY2g1lCJpcGFkZHI0iI3N14xMjEuMTk0ljeZm
i
sIm5hbWU1o1Jsa
WnoIE
NoYBsZXi1lCJvaQo10jjyM
UyOTQ4Y
y01NzAxLTQyNjU
tODExZC02MzNhMzE0NjczZjAiLCJvb
nByZw1fc2lkIjoiUy0xLTUtmjEtMjEyNzUyMTE4NC0xNjA0MDEyOTIw
tE40Dc5Mj
c1MjctNDY2NzUyOCIsInB1aWQo1ixMD
A2N0ZGR
TgwMUFODDYxIiwi
cmgi
o1iwlkfSb0f2NGo1Y3ZH
R3IwR1JxeTE4MEJY111VjNzQVRialJwR3U
tNEMtZUdfZTBZYUFEVS41lCJzY3A1o1j1c2Vx2ltcG
Vyc29uYXRpb24iLCJ
zdWIoijsRk9zQ0xkdGFidHBoU01c3VytktqX3lqam84axo4s2zaM0Z4dHdnV3ZNIiwidG1kIjoiNzJm0tG4YmYtODzms00MWFmlTkxyWnItMmQ3Y2QwMTFkYjQ3Ii
widW5pcXVlX25h
bwUuio1JyY2hhcGx1ckbtaWnyb3NvZnQuY29tliwidXbuIjoi
cmNoYXBsZxJAbW1jcm9zb220LmNb
vSisInV0aS161ktqdV1hbEdYQ1Vhcnh
HajNBND
B3QUE1lCJ2ZxIIo1IxLjA1+Q.U6
yTE35AdDTQhJgUEjjUPFW6raM90RF07SzFya4azbPzrwKntGBrekyjLzqar5C-cJzPw9c8dp5Gn-g1Hw0cpXnZTbrA-Mv9-g4Gwsto3I1LxqXkH3bBmhGHqaDvQ3j1D-N_I1SU1whxj6F
JsgzfwB1NgCo1gt0nDyEW
HrfgvO31Pk
mwNTFLu3-hUqRQDxm1eu519Uf4cxMm00cxSaKegNs2ec0aYAUXcJgd2cdjV1EgbioL9ZdCCNJRB1AgHhJcxHVGafPBtyL03XL_Xngc-DpZdy
C6MMp1SF4-BEc72S9N1i4lJHNpGq-hy8DCB2_5bduHwXMGOTnf1wbq14PQ
C:\Users\rchapler>
```

## Step 3: Post API Request

Open a new request in Postman.

The screenshot shows the Postman application window. At the top, there's a menu bar with File, Edit, View, Help, Home, Workspaces, Reports, Explore, and a search bar. On the right side of the header are icons for cloud, file, gear, sign in, and create account. Below the header, a yellow banner says "Working locally in Scratch Pad. Switch to a Workspace". The main area shows a list of requests on the left and a detailed view of a selected POST request on the right. The request URL is https://practicumdec.westus2.kusto.windows.net/v2/rest/query. The method dropdown shows "POST". The "Authorization" tab is selected, showing a dropdown set to "Bearer Token". A tooltip explains that the authorization header will be automatically generated. The "Token" field contains a long JWT token: eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIn... . Other tabs like Params, Headers, Body, Pre-request Script, Tests, and Settings are visible at the bottom of the request view. At the very bottom of the window are buttons for Body, Cookies, Headers (9), Test Results, Status: 200 OK, Time: 288 ms, Size: 8.31 KB, Save Response, Find and Replace, Console, Runner, and Help.

Select **POST** from the dropdown. Enter a hostname (with the name you used for your cluster); example:

https://practicumdec.westus2.kusto.windows.net/v2/rest/query

On the **Authorization** tab, select “**Bearer Token**” from the dropdown and paste your previously copied token in the resulting interface.

This screenshot shows the same Postman interface as above, but with different configuration. The "Headers" tab is now selected, showing a list of custom headers: Accept: application/json, Accept-Encoding: deflate, Content-Type: application/json; charset=utf-8, and Host: help.kusto.windows.net. The "Authorization" tab is still visible but not selected. The status bar at the bottom indicates a successful response: Status: 200 OK, Time: 288 ms, Size: 8.31 KB, and Save Response.

On the **Headers** tab, enter the following items:

Accept	application/json
Accept-Encoding	deflate
Content-Type	application/json; charset=utf-8
Host	help.kusto.windows.net

The screenshot shows the Postman application interface. At the top, there are tabs for Home, Workspaces, Reports, and Explore, along with a search bar and user account options. Below the header, a yellow banner says "Working locally in Scratch Pad. Switch to a Workspace". The main area shows a POST request to "https://practicumdec.westus2.kusto.windows.net/v2/rest/query". The "Body" tab is selected, showing a raw JSON payload:

```
1 {  
2     "db": "practicumded",  
3     "csl": "StormEvents | take 5"  
4 }
```

Below the body, the status bar indicates "Status: 200 OK Time: 288 ms Size: 8.31KB".

On the **Body** tab, paste the following:

```
{  
    "db": "practicumded",  
    "csl": "StormEvents | take 5"  
}
```

Click the **Send** button.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', 'Help', 'Home', 'Workspaces', 'Reports', 'Explore', and a search bar. On the right of the top bar are 'Sign In' and 'Create Account' buttons. Below the navigation is a yellow banner with the text 'Working locally in Scratch Pad. Switch to a Workspace'. The main area shows a request card for a POST method to 'https://practicumdec.westus2.kusto.windows.net/v2/rest/query'. The body tab is selected, displaying a JSON response with line numbers from 1 to 24. The response content is as follows:

```
1  {
2    "FrameType": "DataSetHeader",
3    "IsProgressive": false,
4    "Version": "v2.0"
5  },
6  {
7    "FrameType": "DataTable",
8    "TableId": 0,
9    "TableKind": "QueryProperties",
10   "TableName": "@ExtendedProperties",
11   "Columns": [
12     {
13       "ColumnName": "TableId",
14       "ColumnType": "int"
15     },
16     {
17       "ColumnName": "Key",
18       "ColumnType": "string"
19     },
20     {
21       "ColumnName": "Value",
22       "ColumnType": "dynamic"
23     }
24   ]
}
```

You should see JSON with both schema and data included.

*Objective Complete... congratulations!*

# Analysis

Objective(s) in this section describe methods for analyzing data.

## Objective - Metrics Advisor + Data Explorer | Onboard and Investigate

**Use Case:** Customer XYZ shared the following requirements:

- Detect data anomalies for Metric 123
- When significant anomalies are detected, the solution should support investigation

Follow the instructions in this section to **on-board data feeds and investigate findings**.

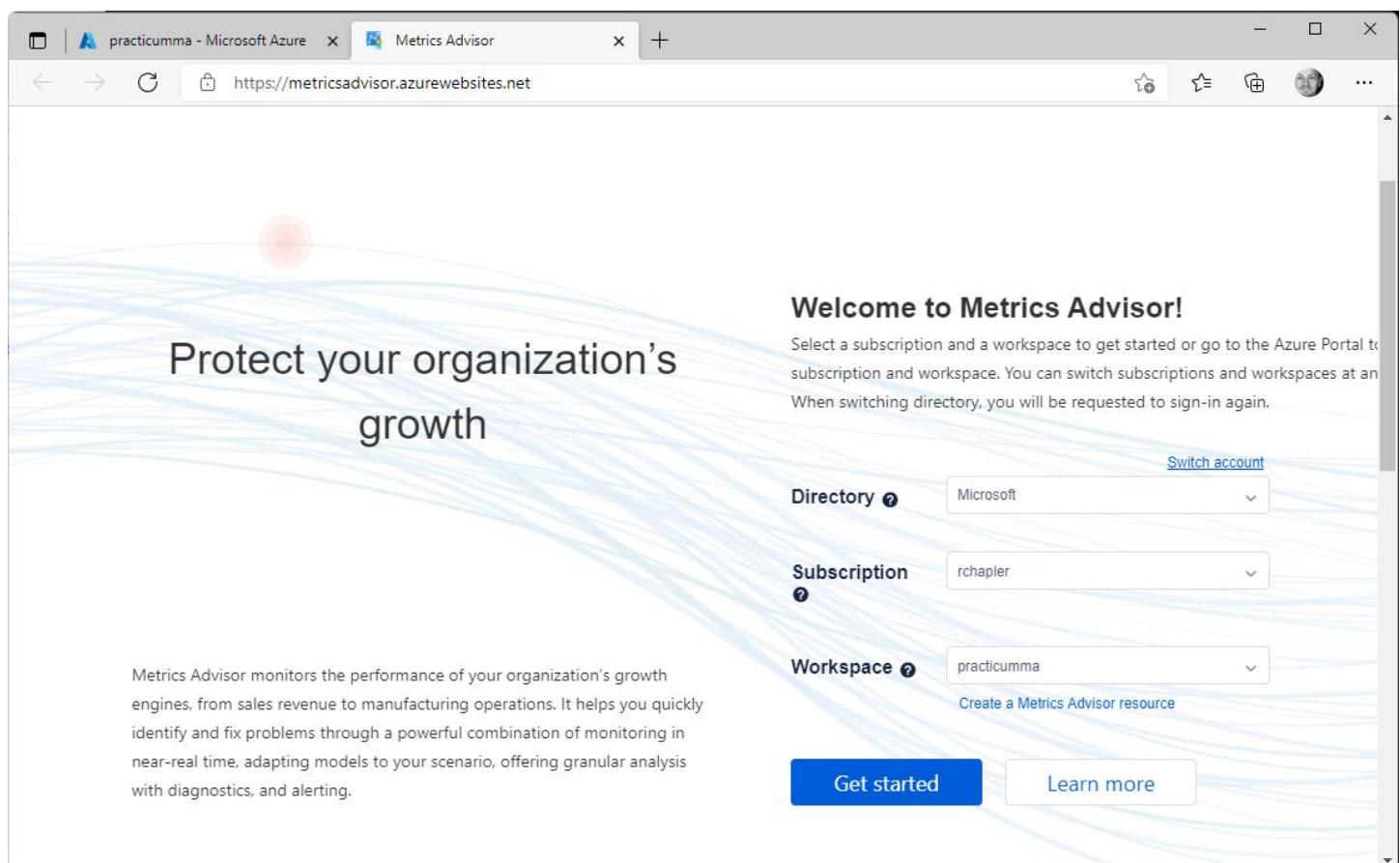
### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Data Explorer
- Metrics Advisor

### Step 2: Onboard Data Feed

Navigate to your instance of Metrics Advisor and then click the “**Go to your workspace**” link.



On the “**Welcome to Metrics Advisor!**” page, enter values for the following items:

Directory	Select your directory
Subscription	Select your subscription
Workspace	Select your instance of Metrics Advisor

Click the “Get started” button.

The screenshot shows the Microsoft Azure Metrics Advisor interface. On the left, a blue sidebar menu lists various options under "Monitor & Diagnostic" such as "Add data feed", "Metrics graph", "Settings", and "Onboarding". The main content area is titled "Data feeds" and displays three sequential steps:

- Step 1: Onboard time-series data** (Icon: A stack of servers with a gear and a checkmark). Description: "Ingest your time-series data from various data sources supported".
- Step 2: Tune configuration & subscribe anomaly alerts** (Icon: A chart with a gear and a warning sign). Description: "Fine tune Detection configurations to better serve real-world scenarios" and "Create a hook and subscribe real-time anomaly alerts".
- Step 3: Diagnose incidents** (Icon: A clipboard with a magnifying glass and a stethoscope). Description: "Identify key contributors with dimension tree" and "Chase down correlations with metrics graph".

A central call-to-action button at the bottom reads "Onboard my first data feed".

Click the “Onboard my first data feed” button.

The screenshot shows the 'Add data feed' page in the Microsoft Azure Metrics Advisor. The left sidebar has a blue header with the Metrics Advisor logo and a 'Collapse menu' button. Below the header, the sidebar lists various sections: Onboarding, Add data feed (which is selected and highlighted in blue), Monitor & Diagnostic, Data feeds, Incident hub, Metrics graph, Settings, API keys, Hooks, Credential entity, Help, Documentation, and Public community. The main content area has a white background. At the top, there's a 'Connection settings' section with a note about granting IP access. Below it are dropdowns for 'Source type' (set to 'Azure Data Explorer (Kusto)'), 'Granularity' (set to 'Daily'), and 'Ingest data since (UTC)' (set to '2007-01-01'). There's also a 'Import' and 'Start a tour' button. The next section is 'Authentication type', with a dropdown set to 'Managed Identity'. Under 'Connection string', there's a text input field containing the URL 'Data Source=https://practicumdec.westus2.kusto.windows.net;Initial Catalog=practicumdec'. Below this is a 'Tips' section with a note about specifying the connection string to access Azure Data Explorer. A 'Sample' section provides a template: 'Data Source=<URI Server>;Initial Catalog=<Database>'. There's a 'Learn more' link. The final section is 'Query', which includes an 'Important' note about running queries continuously and aggregating data by dimensions. It also includes a 'Sample' query for 'StormEvents' and a 'Load data' button.

On the “Add data feed” page, enter values for the following items:

Source Type	Select “ <b>Azure Data Explorer (Kusto)</b> ”
Granularity	Confirm default selection, <b>Daily</b>
Ingest Data Since (UTC)	Select a date appropriate for your data set
Authentication Type	Select “ <b>Managed Identity</b> ”
Connection String	Enter in form “Data Source=https://{{ADX Cluster}}.{region}.kusto.windows.net;Initial Catalog={{ADX Database}}
Query	Create and test a query in your database, then parameterize; example below:  <pre>StormEvents   summarize sum(DamageProperty) by startofday(StartTime), State   where StartTime &gt;= todatetime("@IntervalStart") and StartTime &lt; todatetime("@IntervalEnd")</pre>

Click the “Load Data” button.

The screenshot shows the Microsoft Azure Metrics Advisor interface for adding a data feed. On the left, a sidebar includes links for 'Credential entity', 'Help', 'Documentation', and 'Public community'. The main area has tabs for 'Query' and 'Schema configuration'. The 'Query' tab displays a DAX-like query:

```
StormEvents  
| summarize sum(DamageProperty) by startofday(StartTime), State  
| where StartTime >= todatetime("@IntervalStart") and StartTime < todatetime("@IntervalEnd")
```

The 'Schema configuration' tab shows sample data and schema mapping. The sample data table has columns: State, StartTime, and sum\_DamageProperty. The data rows are:

State	StartTime	sum_DamageProperty
NORTH CAROLINA	2007-01-01T00:00:00Z	0
WISCONSIN	2007-01-01T00:00:00Z	0
NEW YORK	2007-01-01T00:00:00Z	20000
ALASKA	2007-01-01T00:00:00Z	0
DELAWARE	2007-01-01T00:00:00Z	0
OKLAHOMA	2007-01-01T00:00:00Z	775000
INDIANA	2007-01-01T00:00:00Z	110000

The schema configuration table maps columns from the sample data to dimensions and measures. The columns are:

Column name	Display name	Column type	Select:
State	State	String	<input type="radio"/> Timestamp <input checked="" type="radio"/> Dimensions <input type="radio"/> Measure
StartTime	Event Date	String	<input checked="" type="radio"/> Timestamp <input type="radio"/> Dimensions <input type="radio"/> Measure
sum_DamageProperty	Count of Events	String	<input type="radio"/> Timestamp <input type="radio"/> Dimensions <input checked="" type="radio"/> Measure

A blue 'Verify schema' button is located at the bottom right of the schema configuration section.

Click the “Verify Schema” button.

practicumma - Microsoft Azure Metrics Advisor https://metricsadvisor.azurewebsites.net/add-data-feed

**Advanced settings** Help

**Automatic roll-up settings** Help

My data has already rolled up and the dimension value is represented by  NULL or Empty (Default)

I need the service to roll up my data by calculating  None and represent it by  Set roll-up columns

I do not need to perform root cause analysis into dimensions for my metrics.

**Ingestion options** Help

Ingestion time offset  hours Enter Max concurrency... Stop retrying after  hours Minimum retrying interval  hours

**Data feed not available alert** Help

Alert hooks  Select Grace period  hours Snooze  hours consecutive "data feed not available" alerts

By applying hooks, you can receive alerts. You can [create and manage hooks](#).

**Misc**

Missing points filling for anomaly detection model Help

Smart filling  Fill previous  Fill custom value:  0  No filling

Action link template Help

Tips: You can use these placeholders in the URL template %datafeed, %metric, %timestamp, %detect\_config, %tagset.

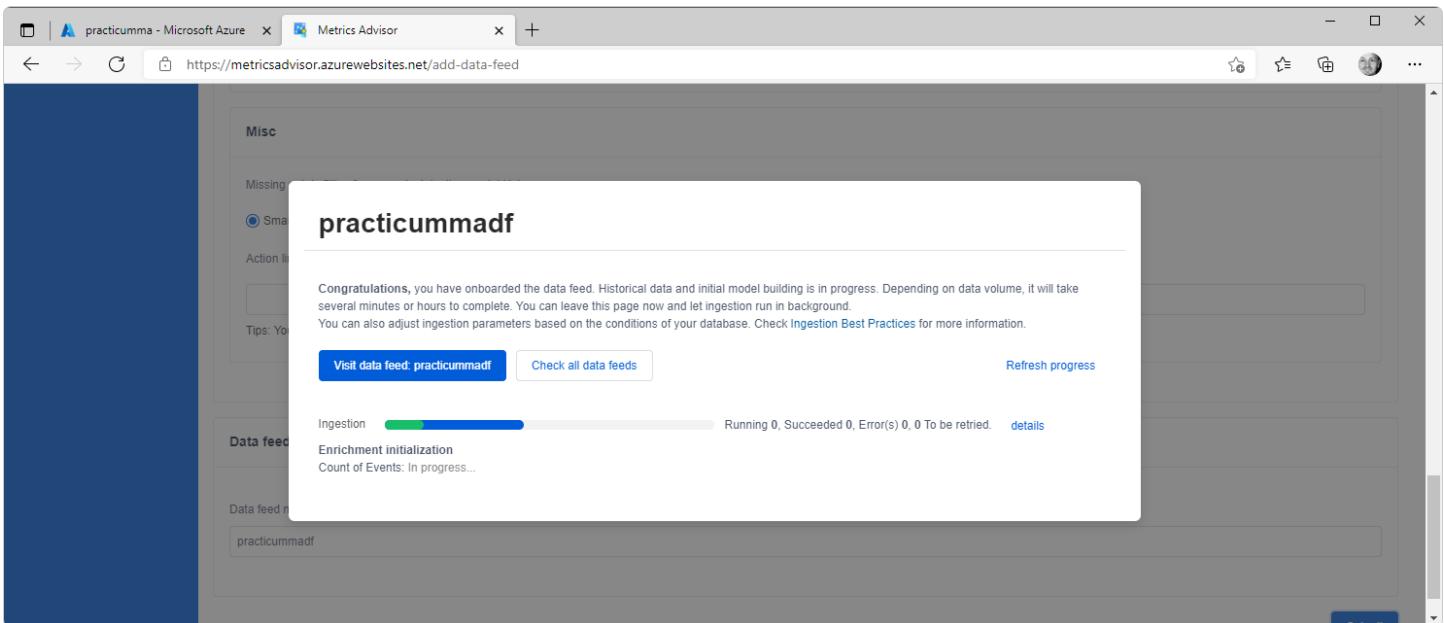
**Data feed name**

Data feed name  practicummadf

**Submit**

No changes are required to default values in “**Advanced Settings**”.

Enter a “**Data Feed Name**” and then click the Submit button.



You will receive a “**Congratulations...**” message.

Click the “**Visit data feed...**” button.

Ingestion Progress: Running 0, Succeeded 50, Error(s) 0, 0 To be retried.

Metrics: It might take 5-8 minutes to see all the metrics here after the data feed is created.

Metric name	Metric ID	Metric description	Series count	Start time	Actions
Count of Events	b2172aec-5fd0-4a0a-bb8d-5da630c4e217	<a href="#">Edit</a>	0	2007-01-01T00:00:00Z	

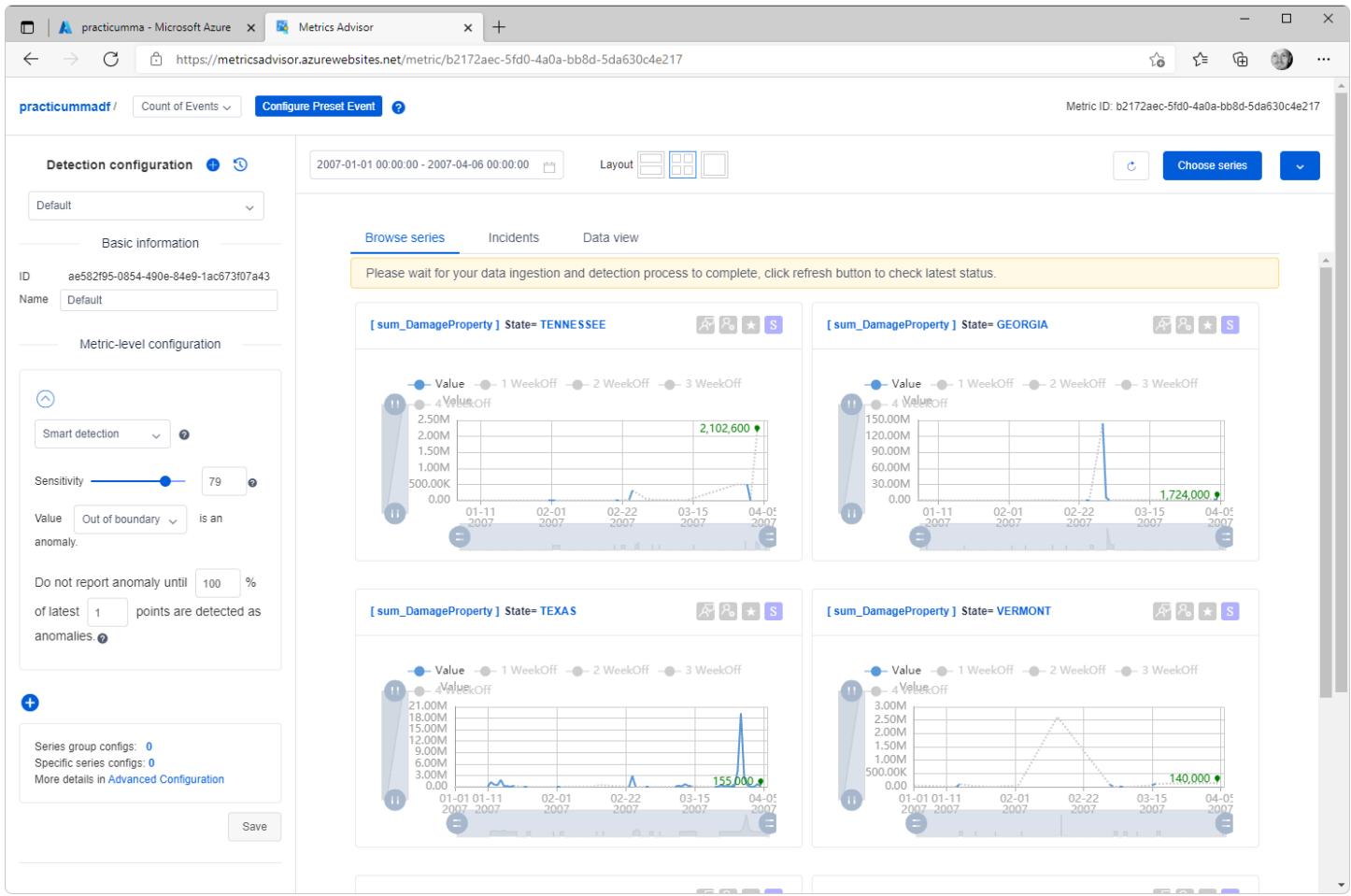
Add a new metric which is calculated by current metrics.

**Data feed Information** [Edit](#)

Data feed status	Active <a href="#">Toggle</a>
Name	practicummadf <a href="#">Edit</a>
ID	dbf98b41-f6f3-4fcc-8fdd-328b6be1a311
Created time	2021-09-21T15:06:28Z
Dimensions	State

Monitor progress on the data feed page.

Click into the Metric Name link to see analysis.



Objective Complete... congratulations!

# Application

Objective(s) in this section describe methods for using data products in user-facing applications, workflows, and artificial intelligence.

## Objective - SQL + Power Apps + AI Builder | Capture Data

**Use Case:** Company XYZ wants to produce a customer-facing app that can be used to capture images and metadata for an insurance claim {e.g., a vehicle with damage to a headlight}.

Follow the instructions in this section to **build an application that employs artificial intelligence with minimal code**.

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Azure SQL
- PowerApps

### Step 2: Create Target Table

Navigate to your instance of SQL.

The screenshot shows the Microsoft Azure portal interface for a SQL database named 'practicumsd'. The left sidebar contains navigation links for Overview, Activity log, Tags, Diagnose and solve problems, Quick start, and Query editor (preview). The main area is titled 'practicumsd (practicumsds/practicumsd) | Query editor (preview)' and features a large 'SQL' logo. Below it, the text 'Welcome to SQL Database Query Editor' is displayed. A 'SQL server authentication' form is shown, with 'Login' set to 'rchapler' and 'Password' masked. To the right, there is an 'Active Directory authentication' section with a 'Continue as rchapler@microsoft.com' button. A red error message at the bottom left states: 'Cannot open server 'practicumsd' requested by the login. Client with IP address [REDACTED] is not allowed to access the server. To enable access, use the Azure Portal or run sp\_set\_firewall\_rule on the master database to create a firewall rule for this IP address or address range. It may take up to five minutes for this change to take effect.' There is also a 'Whitelist IP [REDACTED] on server practicumsd' link. At the bottom center is a blue 'OK' button.

Navigate to “Query editor...” and login. Whitelist your IP address as appropriate.

The screenshot shows the Microsoft Azure Query editor (preview) interface. The left sidebar lists various services: Overview, Activity log, Tags, Diagnose and solve problems, Quick start, and Query editor (preview). The main area displays a query titled "Query 1" which has run successfully. The query code is:

```
1 CREATE TABLE dbo.myTable( Id INT NOT NULL IDENTITY(1,1) PRIMARY KEY, Name VARCHAR(64), Picture IMAGE )
```

The results pane shows the message "Query succeeded: Affected rows: 0".

Execute the following T-SQL:

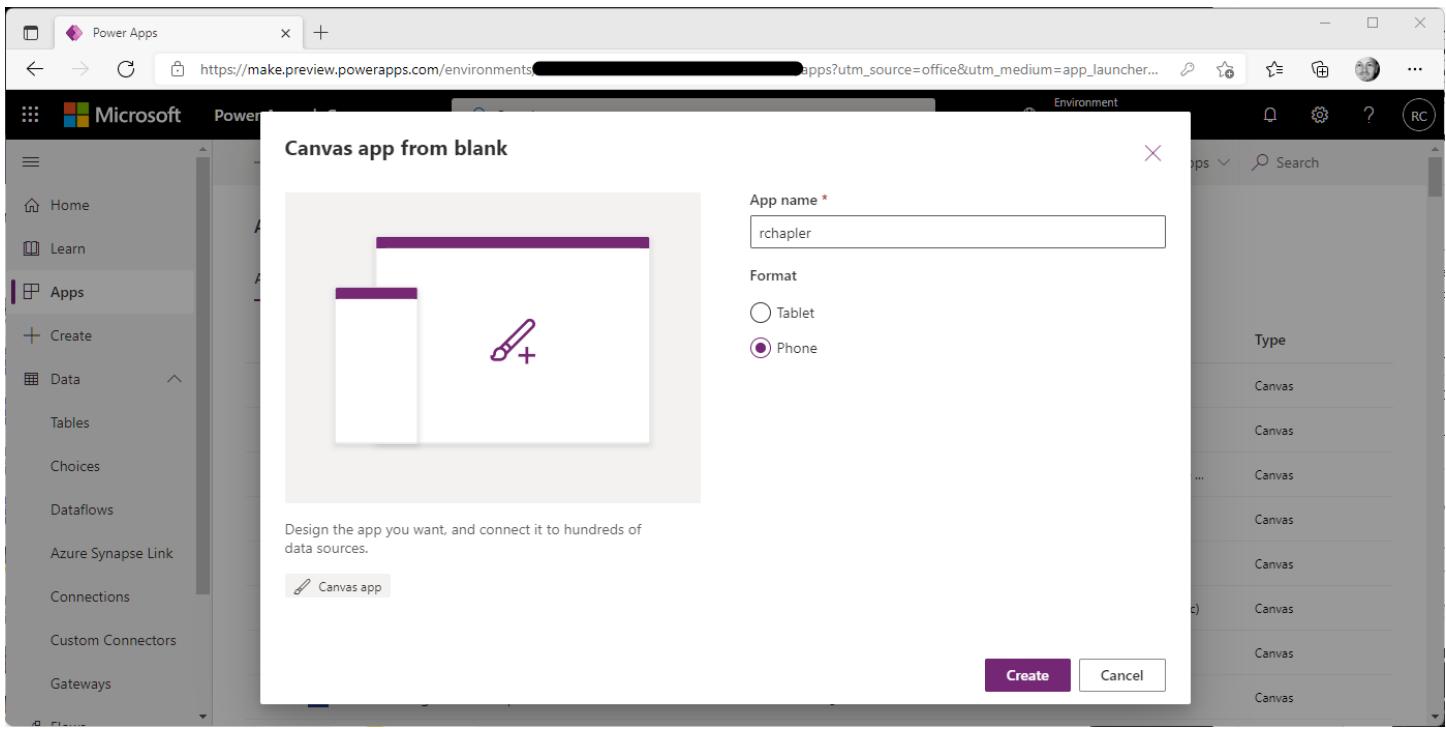
```
CREATE TABLE dbo.myTable( Id INT NOT NULL IDENTITY(1,1) PRIMARY KEY, Name VARCHAR(64), Picture IMAGE )
```

### Step 3: Create Canvas App

Navigate to Power Apps (<https://make.preview.Power Apps.com/>).

Click on **Apps** in the navigation.

At the top of the “**Apps**” page, click “+ New app” and select **Canvas** from the dropdown.



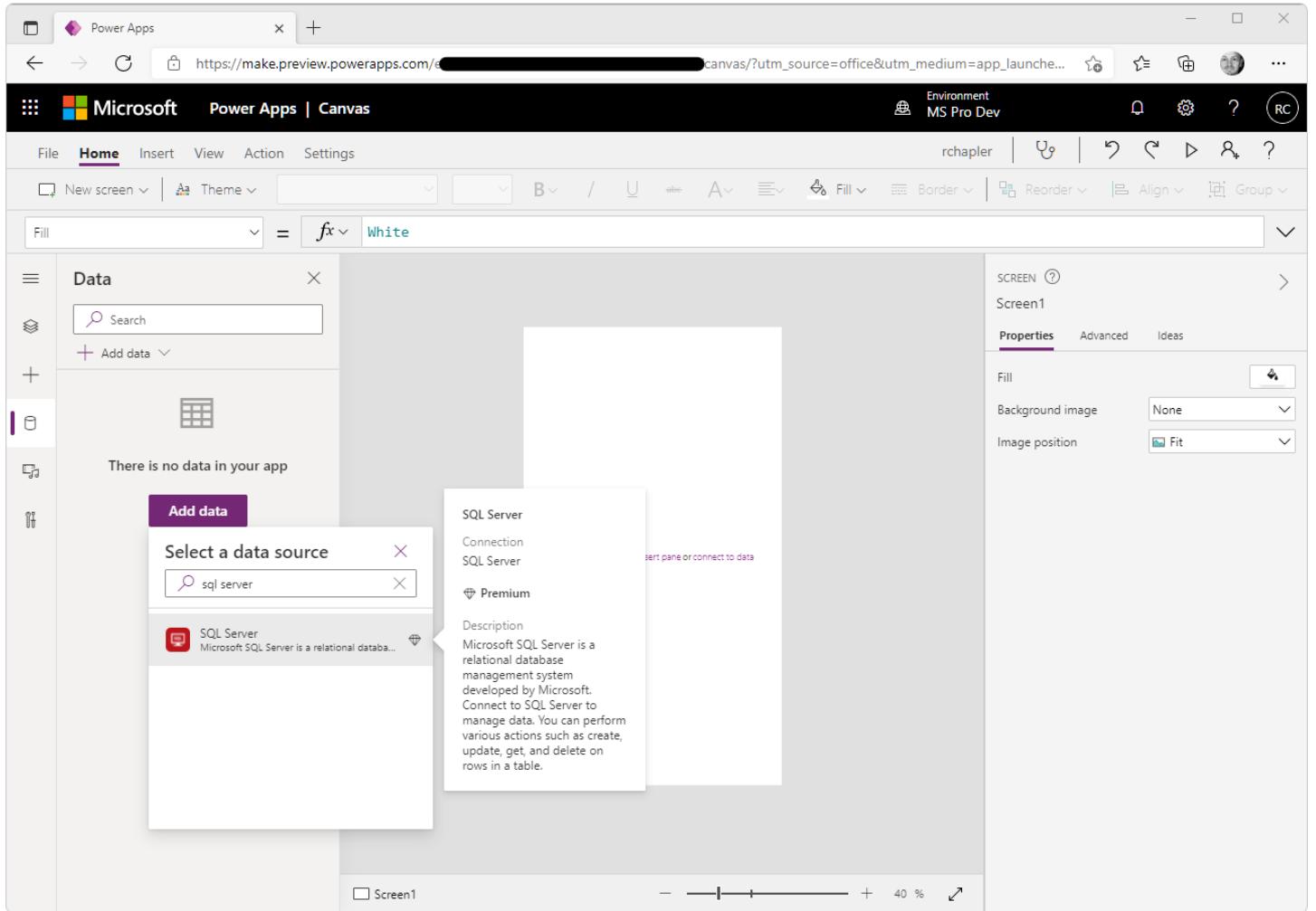
On the “**Canvas app from blank**” popup, enter the following items:

<b>App Name</b>	Enter a meaningful name aligned with standards
<b>Format</b>	Select the <b>Phone</b> radio button

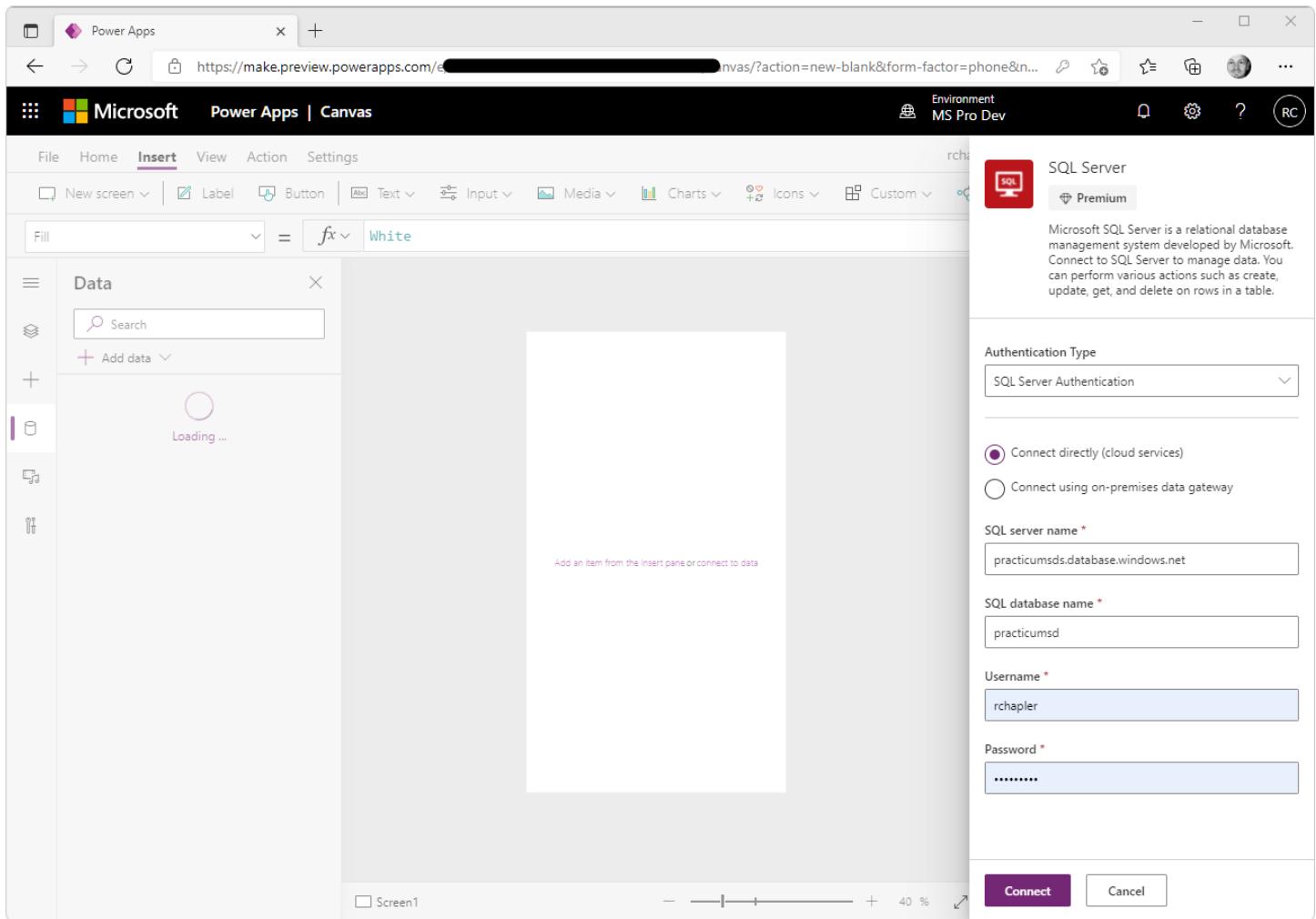
Click the **Create** button.

#### Step 4: Add Data

Click on **Data** in the navigation. Click on the “**Add data**” button.



In “Select a data source”, enter “sql server” in the search input and select “SQL Server” from the results.  
Click “+ Add a connection” on the resulting dropdown.



On the “SQL Server” popout, enter the following items:

**Authentication Type** SQL Server Authentication

**SQL Server Name** Enter the values employed during database server creation

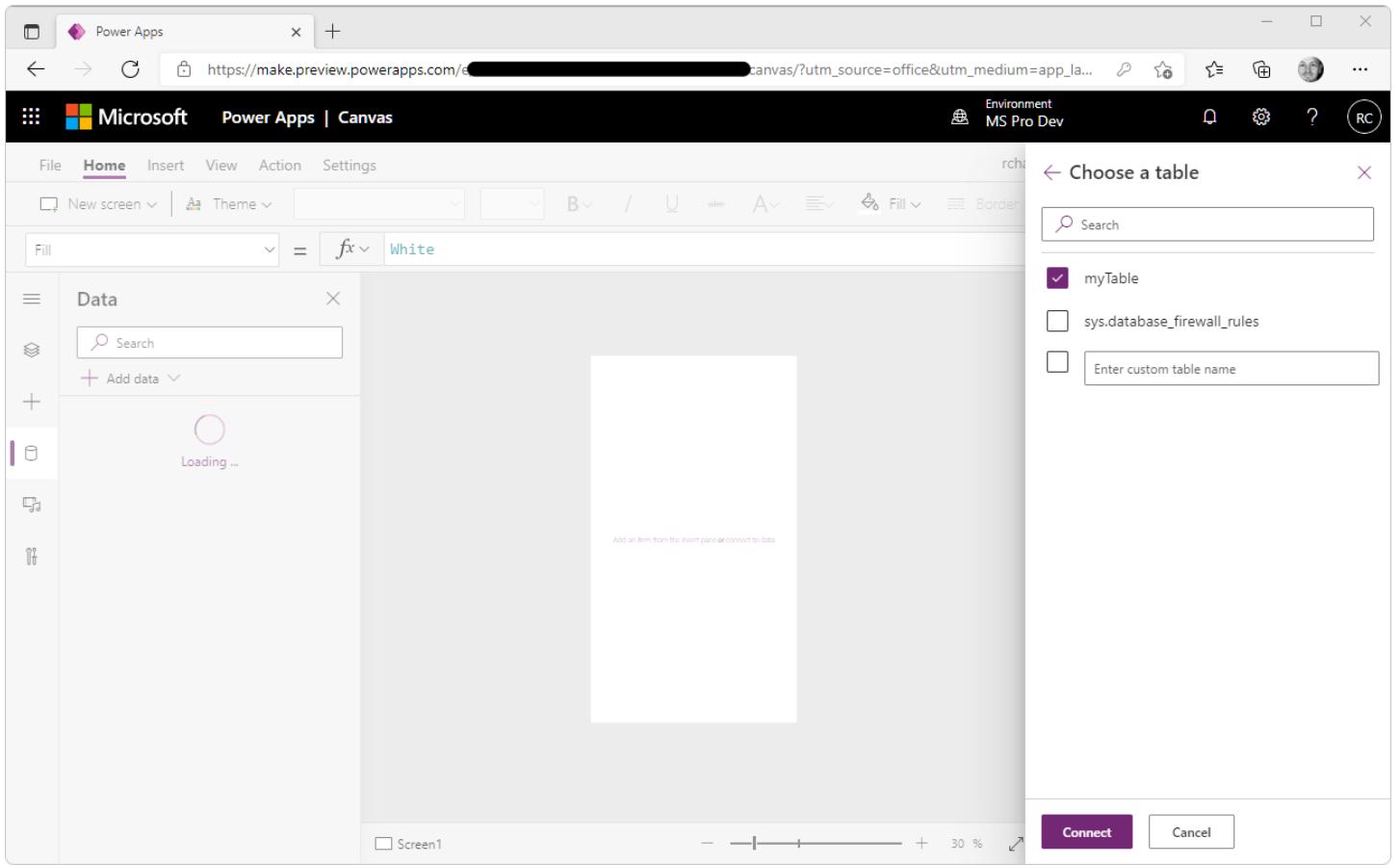
**SQL Database Name**

**Username**

**Password**

Click the **Connect** button.

*Note: Resolve error messages “We weren’t able to add this connection...” and “Client with IP address #.#.#.# is not allowed...”, by adding your IP address to the Azure SQL Server firewall.*



On the “Choose a table” popout, check the table created in [Create Target Table](#).

Click the **Connect** button.

## Step 6: Insert Object Detector

Click the “Add an item from the Insert pane” link.

On the resulting, left-hand **Insert** menu, expand “AI Builder” and select “Object detector”.

On the resulting, right-hand popout, click “+ New model”.

## Step 7: Create AI Model

The screenshot shows the Microsoft Power Apps | AI Builder interface. On the left, there is a navigation sidebar with options like Home, Learn, Apps, Create, Data, Flows, Chatbots, AI Builder (which is selected and highlighted in purple), Build, Models, and Solutions. The main content area has a heading "Enhance your business with AI" and a sub-section "Refine a model for your business needs". It displays five cards: "Category Classification" (document icon), "Entity Extraction" (document icon), "Form Processing" (document icon), "Object Detection" (camera icon), and "Prediction" (checkbox icon). Below the "Object Detection" card, there is a note: "Recognize and count things in images." At the top right, there are environment settings for "Environment MS Pro Dev" and a "Create" button.

On the new “Power Apps | AI Builder”, “Enhance your business with AI” page, click the “Object Detection” button.

Name the AI model and then click the **Create** button.

The screenshot shows the "Select your model's domain" page. On the left, there is a vertical list of steps: "Select domain" (checked), "Choose objects", "Add images", "Tag images", and "Model summary". The main content area has a heading "Select your model's domain" and a note: "When models focus on specific types of objects, they can be more accurate. If you don't see the right option, select **Common objects**. [Learn more](#)". Below this, there are three buttons: "Common objects" (highlighted in pink), "Objects on retail shelves", and "Brand logos". To the right, there is a preview image of a field with several combine harvesters. Three of the harvesters have purple bounding boxes around them, each labeled "Combine". At the bottom right of the preview image, there is a "Save and close" button. On the far right, there is a sidebar with "Quick tips" and "Get help or send feedback" sections. At the bottom left, there is a "Next" button.

On the “Select your model's domain” page, click the “Common objects” and then the “Next” button.

Select from database

### Choose objects for your model to detect

You can add them manually or select from your database. [Learn more](#)

Object names

Headlights

+ Add new object

Back Next 1 object name selected

Environment MS Pro Dev

Car Parts | Save and close

Quick tips

Select from database instead  
Save time and use objects in Dataverse, rather than creating them manually.

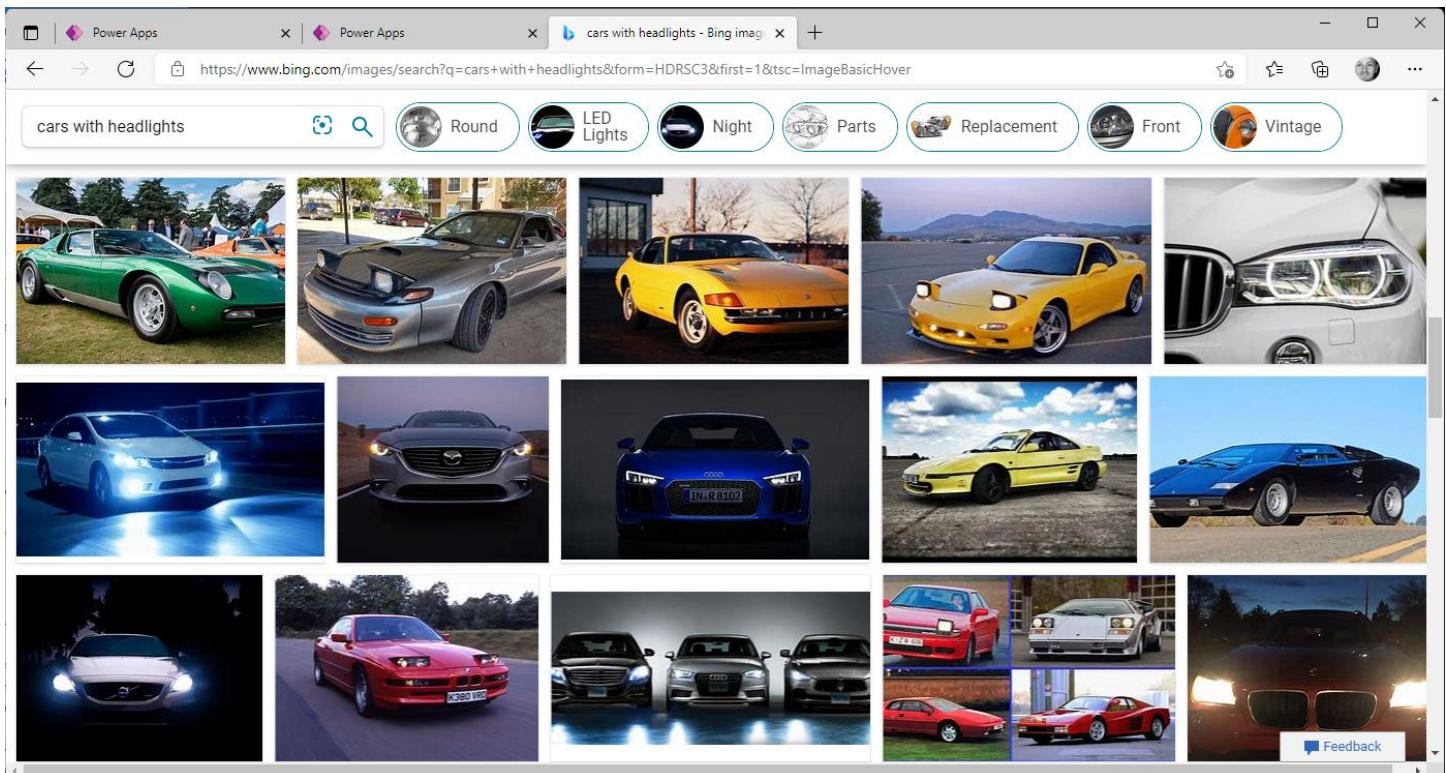
Switch to database

Get help or send feedback  
Get the answers you need, or tell us about your experiences.

Get help

On the “Choose objects for your model to detect” page, click “+ Add new object”, enter an object name and then click the “Next” button.

On the “Add example images for your objects” page, we will add image files that will be used to train the AI model. For this exercise, we can pull images of cars (with headlights) from a search engine.



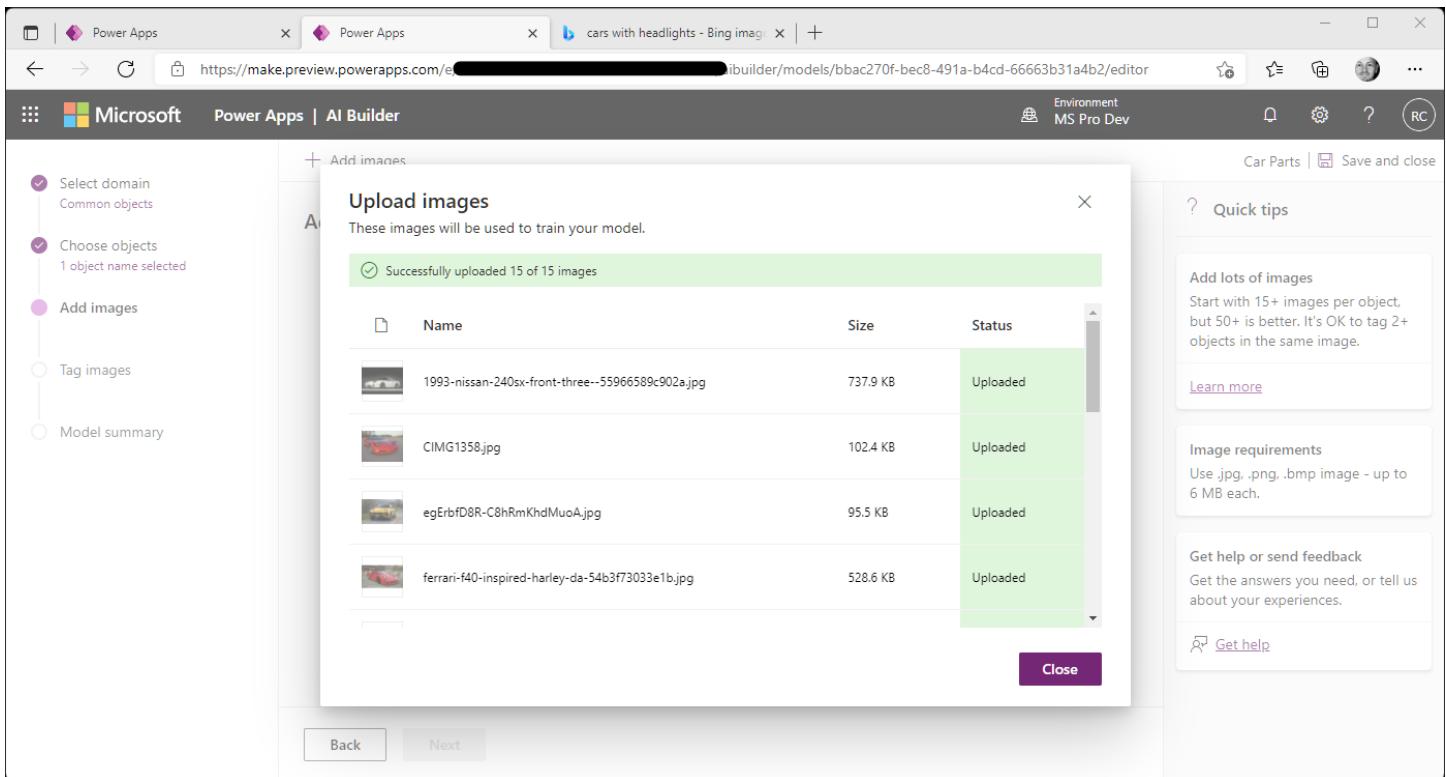
For each of the required 15 images, download an image file to a temporary folder on the Windows desktop.

The screenshot shows the Microsoft Power Apps AI Builder interface. On the left, a sidebar lists steps: 'Select domain' (checked), 'Common objects', 'Choose objects' (checked, showing '1 object name selected'), 'Add images' (highlighted in pink), 'Tag images', and 'Model summary'. The main area has a title 'Add example images for your objects' with a sub-section 'Find images containing your objects'. It says 'You need to add at least 15 images for each object you want to detect.' A large 'Add images' button is centered. To the right, there's a 'Quick tips' section with 'Add lots of images' (start with 15+ images per object, up to 50+), 'Image requirements' (use .jpg, .png, .bmp images up to 6 MB each), and 'Get help or send feedback' (link to Get help). Navigation buttons 'Back' and 'Next' are at the bottom.

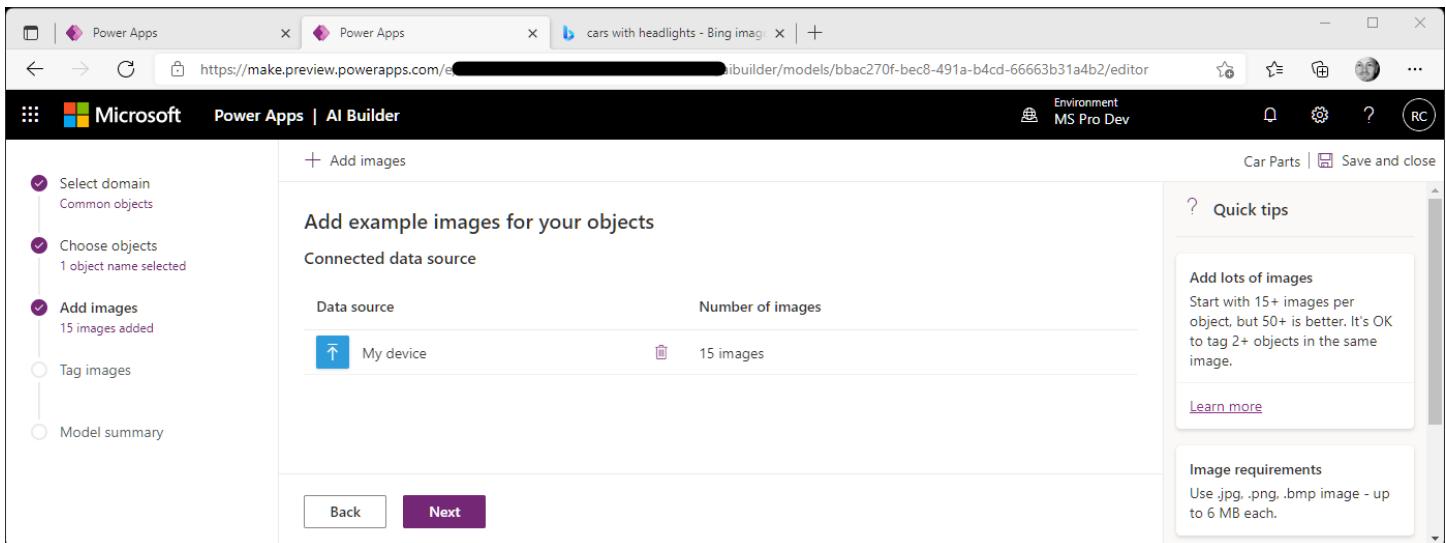
On the “Add example images for your objects” page, click the “Add images” button.

This screenshot shows the 'Add images' popout window. It has a title 'Add images' and a sub-section 'Select a data source'. It lists three options: 'Upload from local storage' (with an upward arrow icon), 'Azure Blob Storage' (with a cloud icon and a 'Premium' badge), and 'SharePoint' (with a SharePoint icon). The 'Upload from local storage' option is highlighted. The background of the popout shows the 'Add example images for your objects' page from the previous screenshot.

On the resulting “Add images” popout, click “Upload from local storage”. Select the Example Images downloaded to your desktop.



Review the items in the “Upload images” popup, then click the “Upload 15 images” button. Click Close.



Back on the “Add example images for your desktop” page, confirm “Number of images” shows 15 or more images, then click Next.

Power Apps | Microsoft Power Apps | AI Builder

Environment MS Pro Dev

Car Parts | Save and close

You will need to tag at least 15 images per object to continue.

Tag the objects in your images

All  

Model summary

Select domain Common objects

Choose objects 1 object name selected

Add images 15 images added

Tag images

Tagging requirements You must tag at least 15 images for each object. Tagging more than 50 images for each object could yield better results. You can have multiple tags per image.

0 Headlights

Want to add more images? Adding more images may increase model performance.

+ Add images

Get help or send feedback Get the answers you need, or tell us about your experiences.

Get help

Select the first image on the “Tag the objects in your images” page.

Power Apps | Microsoft Power Apps | AI Builder

https://make.preview.powerapps.com/e...

Don't use image Clear Discard changes Details Done tagging

Tagging progress

Image 1 of 15 You must tag at least 15 images for each object. Tagging more than 50 images for each object could yield better results. You can have multiple tags per image.

Tags applied 0 Headlights

Choose object

Search

Headlights

Back Next

Drag a rectangle around headlights in the image, then click the Headlights button on the “Choose object” popup.

Repeat for all taggable items, click > to move to through images, and then click the “Done tagging” button in the upper-right.

Power Apps | AI Builder

cars with headlights - Bing image

Environment: MS Pro Dev

Car Parts | Save and close

Tag the objects in your images

All

15 Headlights

Tagging requirements: You must tag at least 15 images for each object. Tagging more than 50 images for each object could yield better results. You can have multiple tags per image.

Want to add more images? Adding more images may increase model performance.

+ Add images

Get help or send feedback: Get the answers you need, or tell us about your experiences.

Get help

Back Next

The screenshot shows the Microsoft Power Apps AI Builder interface. On the left, there's a sidebar with a checklist: 'Select domain' (Common objects), 'Choose objects' (1 object name selected), 'Add images' (15 images added), 'Tag images' (15 images tagged), and 'Model summary'. The main area is titled 'Tag the objects in your images' and displays a 3x5 grid of 15 car images. Each image has a small 'NEW' label in the top-left corner and a checkmark in the center. To the right of the grid, there's a 'Quick tips' section with a heading 'Tagging requirements' and a note about tagging requirements. Below that is a section for adding more images with a '+ Add images' button. At the bottom, there are 'Back' and 'Next' buttons.

Click the **Next** button.

Screenshot of the Microsoft Power Apps AI Builder "Model summary" page.

The left sidebar shows completed steps: "Select domain" (Common objects), "Choose objects" (1 object name selected), "Add images" (15 images added), and "Tag images" (15 images tagged). The "Model summary" step is currently selected.

The main area displays the "Model summary" details:

- Overview**: Model type is Object Detection, owned by Rich Chapler, and the object type is Common objects.
- Image sources**: Data source is My device, with 15 images.
- Information to extract**: Object is Headlights, with 15 tags.

At the bottom are "Back" and "Train" buttons. The "Train" button is highlighted in purple.

The right sidebar includes "Quick tips", "What is training?", "What's next?", and "Get help or send feedback".

Review the “Model Summary” and then click the **Train** button.

Screenshot of the Microsoft Power Apps AI Builder "Model summary" page during training.

The left sidebar shows completed steps: "Select domain" (Common objects), "Choose objects" (1 object name selected), "Add images" (15 images added), and "Tag images" (15 images tagged). The "Model summary" step is currently selected.

The main area displays the "Model summary" details, including a progress indicator showing a laptop with a brain icon, and the text "Your model is training". It also states "This may take a while. You can close this window and come back later." A "Go to models" button is visible.

At the bottom are "Back" and "Training" buttons. The "Training" button is highlighted in purple.

The right sidebar includes "Quick tips", "What is training?", "What's next?", and "Get help or send feedback".

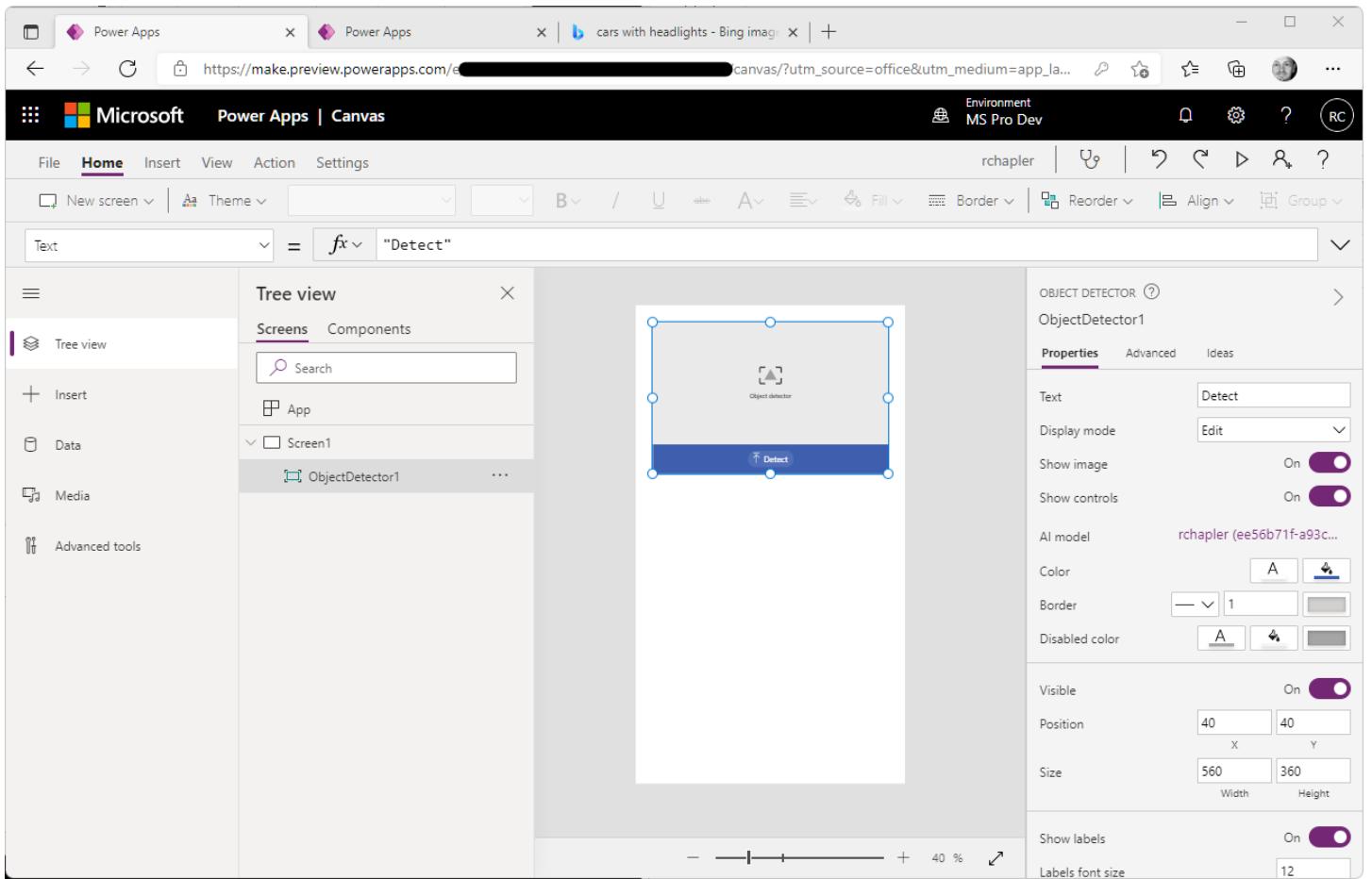
Click the “Go to models” button. After Status changes to **Trained**, navigate to the model.

The screenshot shows the Microsoft Power Apps AI Builder interface. On the left, a sidebar menu includes Home, Learn, Apps, Create, Data, Flows, Chatbots, AI Builder, Build, Models (which is selected), and Solutions. The main content area displays a 'Models > Car Parts' page for 'Object Detection'. A large circular progress bar indicates a 'Performance' of 46%. Below it, a callout box suggests improving the model by providing more training data. Two buttons at the bottom are 'Publish' (in purple) and 'Quick test'. To the right, there's a 'Details' section with 'Training date' (10:29:01 AM), 'Object type' (Common objects), and 'Number of objects' (1). A 'See more' link is also present. On the far right, a panel titled 'How your model is used' lists 'Power Automate' and 'Power Apps', both noting that the model isn't published.

In the snip above, we see that **Performance** {i.e., model performance expectation %} of 46 for the example images included in training.  
In the future, you can improve model quality by including more images and re-training the model.

Click the **Publish** button.

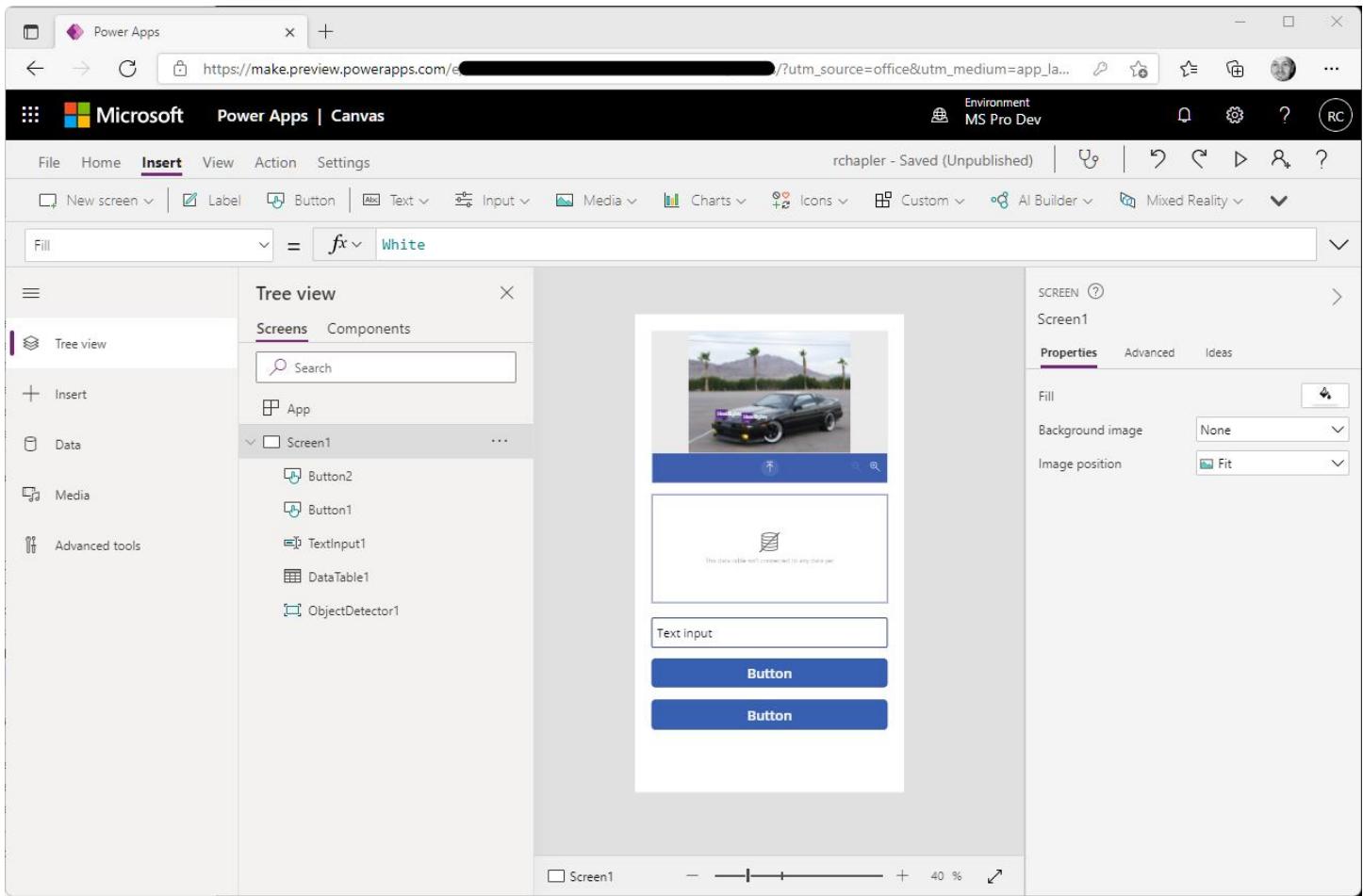
Return to Power Apps and complete the ObjectDetector1 component by selecting your newly created model.



*Note: Consider resizing the component to improve visibility.*

## Step 8: Insert / Configure Controls

Return by closing Preview Mode.

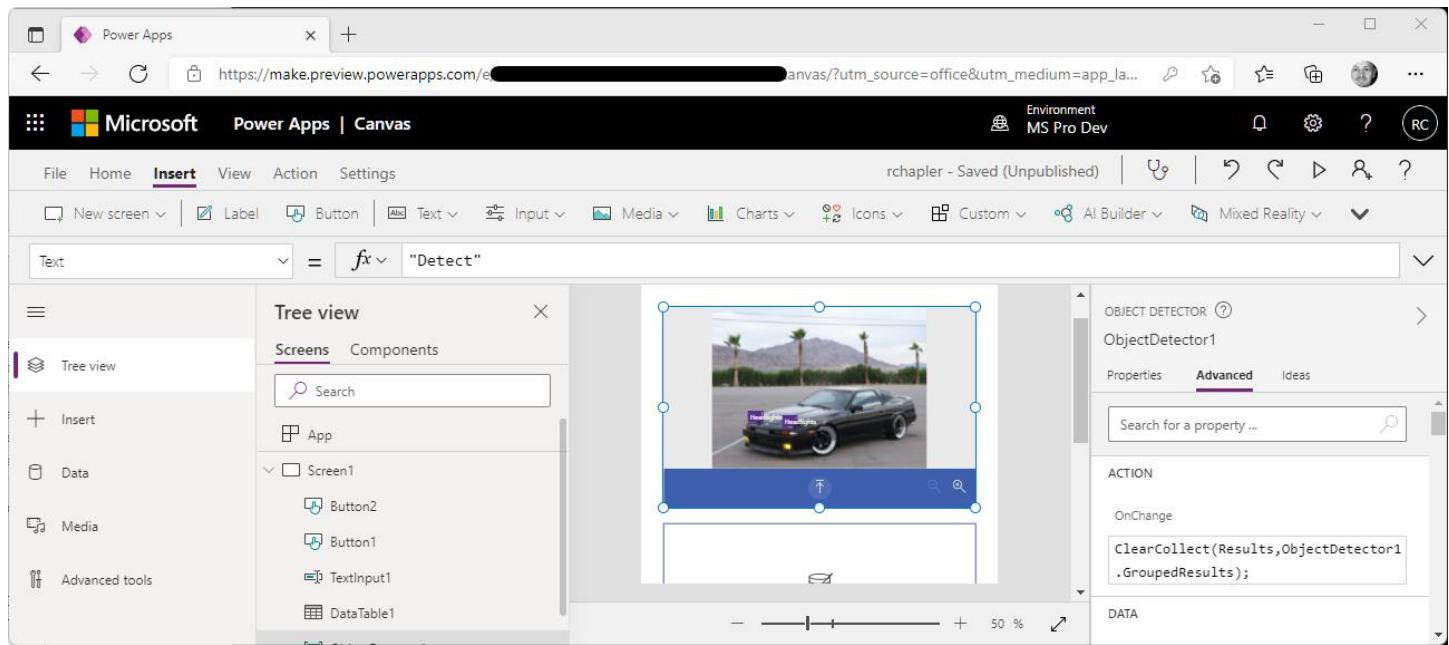


Insert the following controls:

- **Data Table** ... users will use this control to view the results generated by AI Builder
- **Text Input** ... users will use this control to enter comments about the analyzed image
- “Submit” **Button** ... users will use this control to save the image and comment to Azure SQL
- “Reset” **Button** ... users will use this control to clear previously entered values

You might note that I employed formatting {e.g., control width, border size, color, etc.} to enhance usability.

## ObjectDetector1

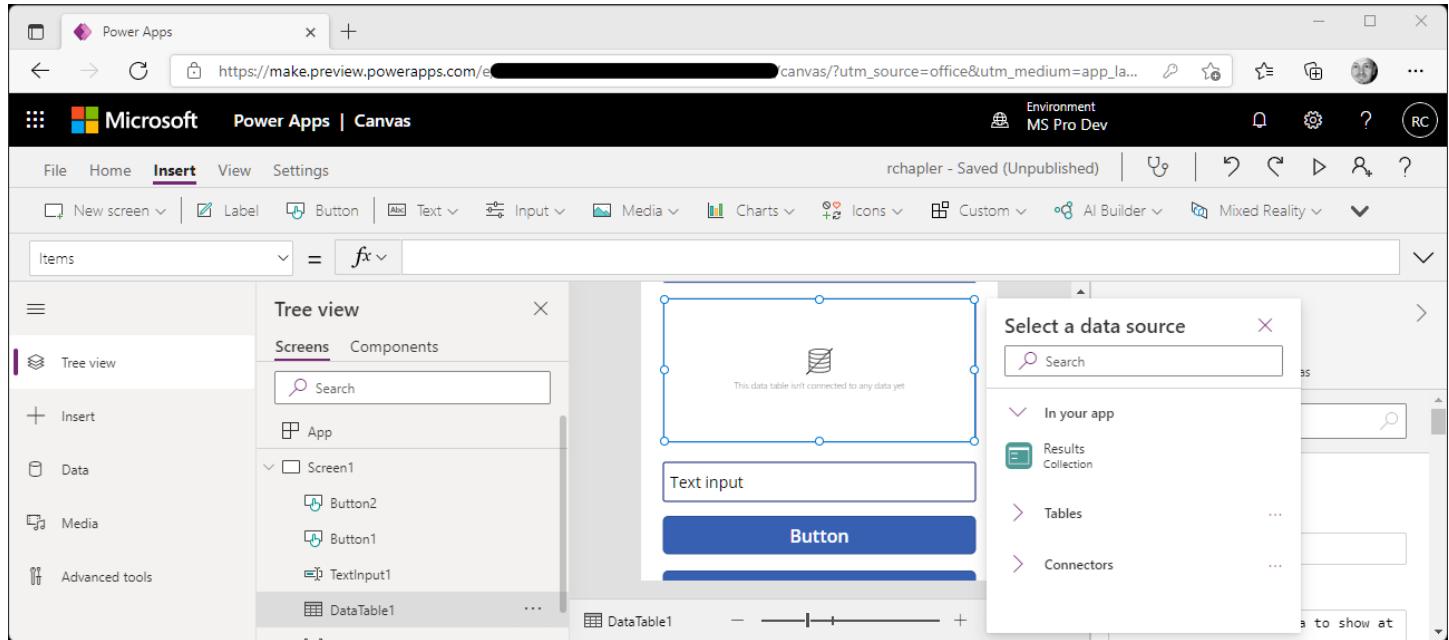


Click on **ObjectDetector1** in the left-hand **Tree View**. On the resulting right-hand popout, click the **Advanced** tab.

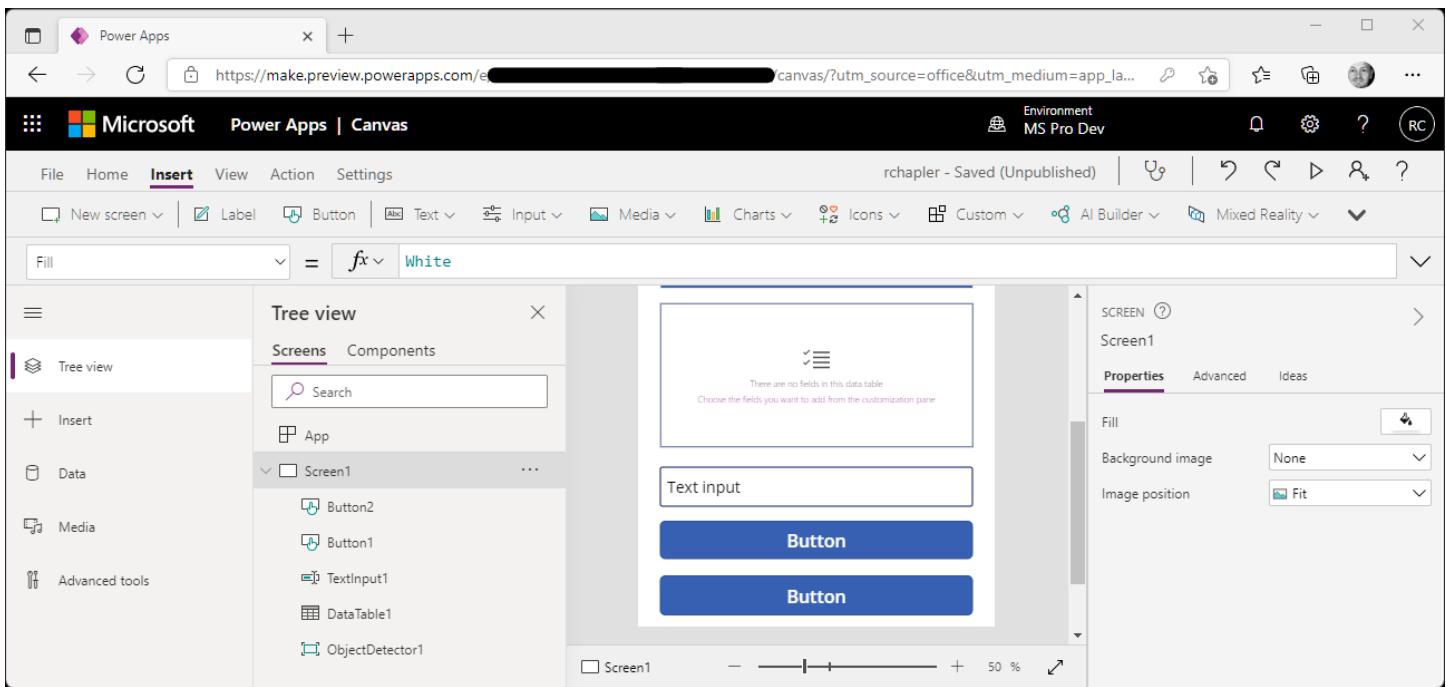
Paste the following logic into **Action | OnChange**:

```
ClearCollect(Results, ObjectDetector1.GroupedResults);
```

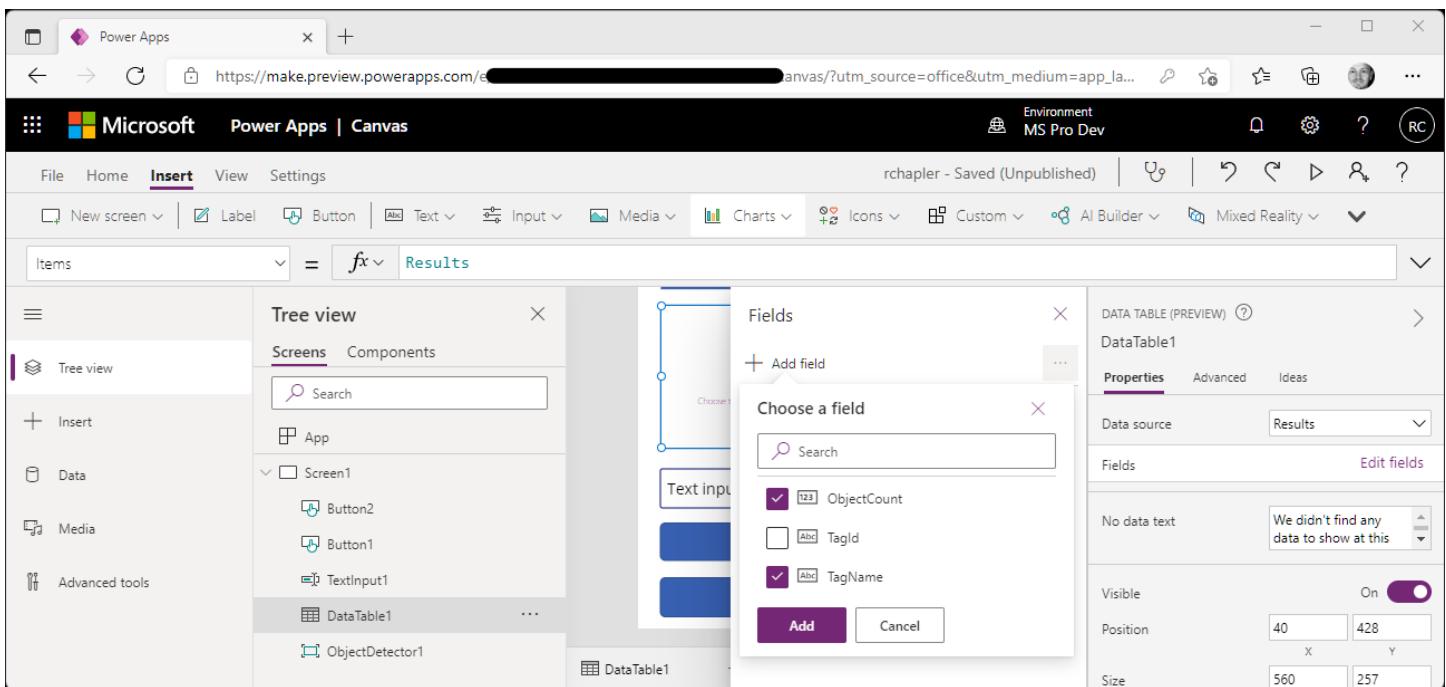
## DataTable1



Click on **DataTable1** in the left-hand **Tree View**. On the resulting popup, select "**In your app**" | **Results**.



On the updated control, click the “Choose the fields you want to add...” link.



On the resulting **Fields** popout, click “+ Add field”.

In the resulting “Choose a field” popup, check **TagName** and then **ObjectCount**.

Click the **Add** button.

## TextInput1

The screenshot shows the Microsoft Power Apps Canvas editor interface. On the left, there's a tree view of components under 'Screen1'. In the center, a 'Text input' component is selected, displaying a table with one row: 'Headlights' and '2'. To the right, a 'TEXT INPUT' popout shows the properties for 'TextInput1'. The 'Properties' tab is selected, showing the following settings:

Default	Text input
Format	Text
Hint text	"Enter Comments"
Font	Open Sans
Font size	21
Font weight	B Normal
Font style	/ U +
Text alignment	Left Center

Click on **TextInput1** in the left-hand **Tree View**.

On the resulting “**TEXT INPUT**” popover, **Advanced** tab, enter the following items:

Default	“”
Hint Text	“Enter Comments”

## Button1

The screenshot shows the Microsoft Power Apps Canvas editor interface. On the left, there's a tree view of components under 'Screen1'. In the center, a 'Button1' component is selected, displaying a button labeled 'Submit'. To the right, a 'BUTTON' popout shows the properties for 'Button1'. The 'Properties' tab is selected, showing the following settings:

Text	Submit
Display mode	Edit
Visible	On
Position	X: 40 Y: 819
Size	Width: 560 Height: 67
Padding	Top: 5 Bottom: 5

Click on **Button1** in the left-hand **Tree View**.

On the resulting right-hand popover, click the **Properties** tab, and enter the word “Submit” in the **Text** input.

On the resulting right-hand popout, click the **Advanced** tab,

Paste the following logic into **Action | OnSelect**:

```
Patch(myTable, Defaults(myTable), {Name:TextInput1.Text, Picture:ObjectDetector1.OriginalImage})
```

## Button2

The screenshot shows the Microsoft Power Apps Canvas editor interface. On the left, there's a tree view of the app structure under 'Screen1'. In the center, there's a canvas with a text input field labeled 'Enter Comments' and two buttons below it: a blue 'Submit' button and a blue 'Reset' button. The 'Reset' button is selected. On the right, a properties panel for the 'Reset' button is open, showing the 'Advanced' tab selected. Under 'ACTION', the 'OnSelect' section contains the logic: `Reset(TextInput1);`. Under 'DATA', the 'Text' section contains the value: `"Reset"`.

Click on **Button2** in the left-hand **Tree View**.

On the resulting right-hand popout, click the **Properties** tab, and enter the word “Reset” in the **Text** input.

On the resulting right-hand popout, click the **Advanced** tab,

Paste the following logic into **Action | OnSelect**:

```
Reset(TextInput1);
```

## Step 9: Confirm Success

Click **File** in the menu bar, then “**Save As**”, confirm the app name, and finally, click the **Save** button in the bottom-right.

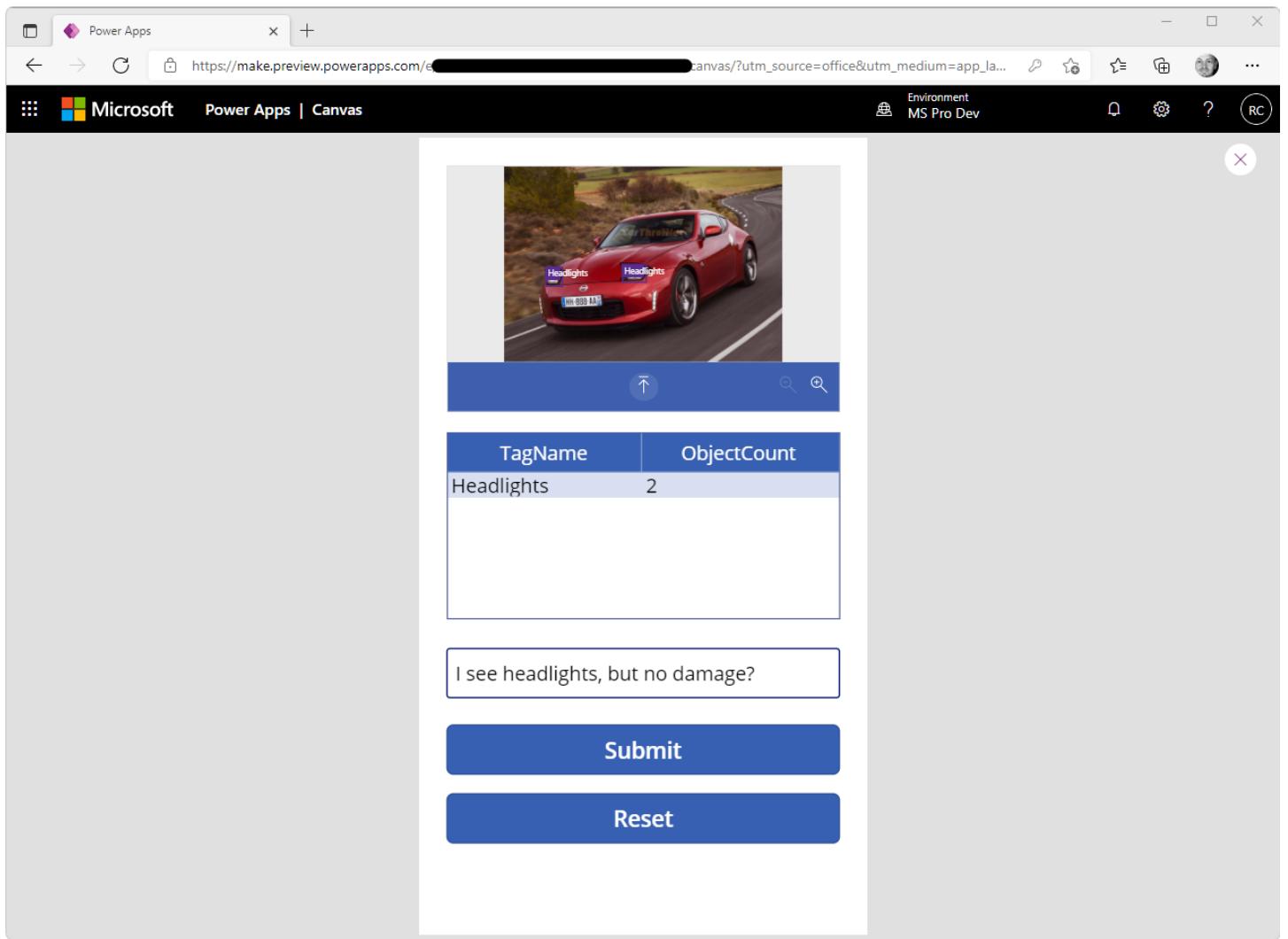
Return to the main page, click the “**Preview the app**” button {i.e., Play icon} in the upper-right.

The screenshot shows the Microsoft Power Apps interface in Canvas mode. The left sidebar displays a tree view of the app's components, including screens and components like Button2, Button1, TextInput1, DataTable1, and ObjectDetector1. The main canvas area shows a red sports car with bounding boxes labeled "Headlights". Below the image is a table with one row:

TagName	ObjectCount
Headlights	2

Below the table is a text input field labeled "Enter Comments" and two blue buttons labeled "Submit" and "Reset". The right side of the screen shows the "Screen" properties panel for "Screen1", which includes tabs for Properties (selected), Advanced, and Ideas. The "Properties" tab contains fields for ACTION (OnVisible, OnHidden), DATA (BackgroundImage, ContentLanguage), and DESIGN (Fill, ImagePosition). The "Fill" field is set to "White" and the "ImagePosition" field is set to "ImagePosition.Fit".

Your app will be presented in a phone simulation.



Click the **Detect** button and select an image file to test objection detection.

Review the resulting values in the data table.

Enter a comment in the text input and click the **Submit** button.

The screenshot shows the Microsoft Azure Query editor (preview) interface. On the left, there's a sidebar with various navigation links like Overview, Activity log, Tags, and Query editor (preview). The main area has a search bar and tabs for 'Query 1' and 'Query 2'. Under 'Query 2', a query is run against a database named 'dbo'. The results show two rows of data from a table named 'myTable'. The first row has an ID of 1 and a name of 'Nice car, but no apparent quality d...'. The second row has an ID of 2 and a name of 'I see headlights, but no damage?'. A message at the bottom indicates 'Query succeeded | 0s'.

ID	Name	Picture
1	Nice car, but no apparent quality d...	/9j/4AAQSkZJRgABAQEAAAAAAA...
2	I see headlights, but no damage?	/9j/2wBDAAYEBQYFBAYGBQYHBwY...

Navigate to **Query Editor** in the Azure SQL Database and confirm that your comment has been added as a new row.

Consider publishing your app and loading it on your phone as one additional confirmation of success.

*Objective Complete... congratulations!*

# Governance

Objective(s) in this section describe methods for cataloging and classifying data.

## Objective - Purview | Discover Data

Data discovery is “an organic process based on communal knowledge”, characterized by the following questions:

- **Search** ... “does the data that I need exist?”
- **Location** ... “how do I connect to Data Product X?”
- **Related** ... “are there related data products that I might find valuable?”
- **Contacts** ... “who are the owners and experts for Data Product X?”

In this section, we will configure Purview and successfully respond to these questions.

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Purview
- SQL

### Step 2: Register Source

Navigate to your instance of Purview in the Azure portal.

Click the “**Open Purview Studio**” button.

Click the **Data Map** icon on the navigation.

Click the **Register** button on the resulting page.

practicump - Microsoft Azure    Azure Purview Studio

https://ms.web.purview.azure.com/resource/practicump/main/datasource/registeredSources?feature.tenant=██████████

Microsoft Azure | Purview > practicump

Search assets

- > Sources
- Collections
- Source management
- Scan rule sets
- Integration runtimes
- Annotation management
- Classifications
- Classification rules

## Sources

Register Refresh

Map view

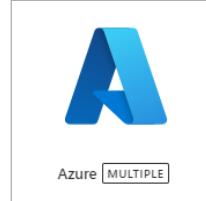
Filter by keyword

Showing 1 collection, 0 sources

## Register sources

Filter by keyword

All Azure Database Power BI Services and apps



Azure [MULTIPLE]



AWS account [MULTIPLE]



Azure Synapse Analytics [MULTIPLE]



Amazon S3



Azure Blob Storage



Azure Cosmos DB (SQL API)



Azure Data Explorer (Kusto)



Azure Data Lake Storage Gen1



Azure Data Lake Storage Gen2



Azure Database for MySQL



Azure Database for PostgreSQL



Azure Dedicated SQL Pool (formerly SQL DW)



Azure Files



Azure SQL Database



Azure SQL Database Managed Instance



Cassandra (Preview)



erwin (Preview)



Google BigQuery (Preview)



Hive Metastore



Looker (Preview)



Oracle



SAP ECC



SAP HANA

Click the **Azure** tab on the “Register sources” popout.

Click to check the “**Azure SQL Database**” option.

Click the **Continue** button.

The screenshot shows the Microsoft Azure Purview Studio interface. On the left, there's a sidebar with various management options like Sources, Collections, Source management, Scan rule sets, Integration runtimes, Annotation management, Classifications, and Classification rules. The main area is titled 'Sources' and shows a 'Register' button. A 'Register sources (Azure SQL Database)' dialog box is open on the right. It contains fields for 'Name' (set to 'practicumsd'), 'Azure subscription' (set to 'rchapler (91e9fddc-ef15-416c-9be1-085f8b1b46ed)'), 'Server name' (set to 'practicumsd'), 'Endpoint' (set to 'practicumsd.database.windows.net'), and 'Select a collection' (set to 'practicump'). A note below the collection field says 'All assets under this source will belong to the collection you select.' At the bottom of the dialog are 'Register', 'Back', and 'Cancel' buttons.

On the “Register sources (Azure SQL Database)” popout, enter values for the following items:

Name	Enter a meaningful name aligned with standards
Subscription	Select your subscription
Server Name	Enter the name of your SQL Database Server
Endpoint	Confirm the populated value
Select a Collection	Leave default value

Click the **Register** button.

### Step 3: Scan Data

Navigate to Sources, “Map view”.

The screenshot shows the Microsoft Azure Purview Studio interface. The left sidebar contains navigation links for Sources, Collections, Source management, Scan rule sets, Integration runtimes, Annotation management, Classifications, and Classification rules. The main area is titled 'Sources' and shows a list of registered sources. One source, 'practicump', is expanded, revealing its details. Below it, another source, 'practicumsd', is listed under 'practicump'. To the right, there is a detailed view of the collection structure, showing the relationship between 'practicump' and 'practicumsd'.

Click the “New scan” icon on the newly created item.

The screenshot shows the 'Scan "practicumsd"' dialog box. It includes fields for 'Name' (set to 'Scan-20210910'), 'Connect via integration runtime' (set to 'Azure AutoResolveIntegrationRuntime'), 'Server endpoint' (set to 'practicumsd.database.windows.net'), 'Database selection method' (radio button selected for 'From Azure subscription'), 'Database name' (set to 'practicumsds'), 'Credential' (set to 'practicumsds'), and 'Select a collection' (set to 'practicump'). At the bottom, there are 'Continue', 'Test connection', and 'Cancel' buttons.

On the “Scan...” popout, enter values for the following items:

Name	Enter a meaningful name aligned with standards
Database Selection Method	Select the “From Azure subscription” radio button
Database Name	Select your Azure SQL database
Credential	Select your “SQL Authentication” credential

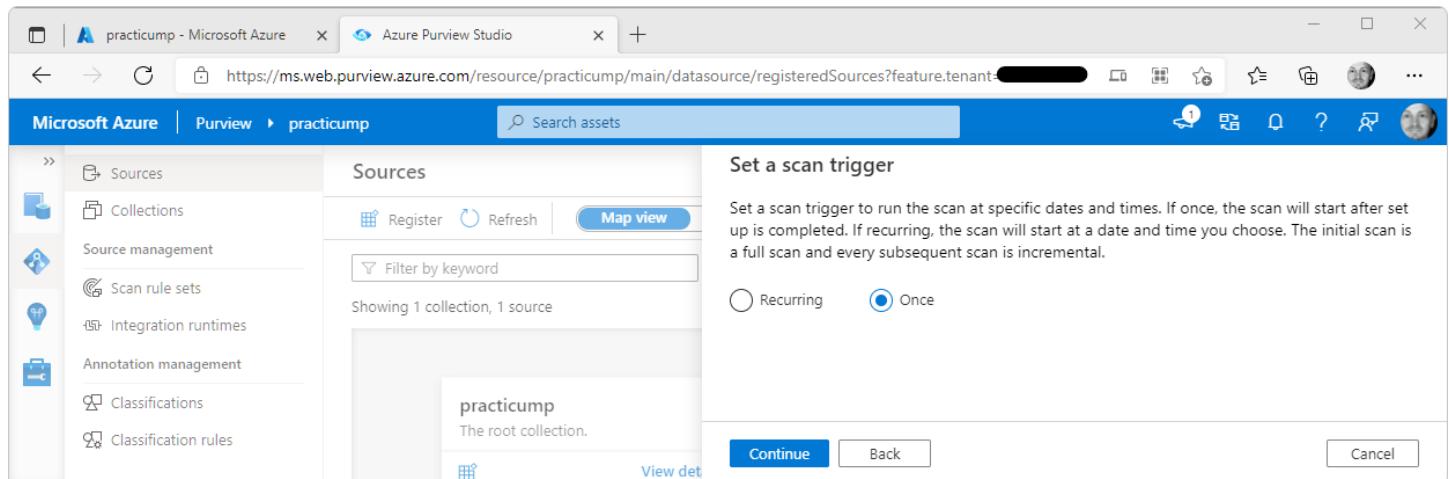
Click “Test connection” to confirm successful configuration and then click the **Continue** button.

The screenshot shows the Azure Purview Studio interface. On the left, a sidebar menu includes 'Sources', 'Collections', 'Source management', 'Scan rule sets' (which is selected), 'Integration runtimes', 'Annotation management', 'Classifications', and 'Classification rules'. The main area is titled 'Sources' and shows a single collection named 'practicump'. Below it is a sub-item 'practicumsd' which is identified as an 'Azure SQL Database'. To the right, a 'Scope your scan' section is open, containing a 'Refresh' button and a search bar. A list of items under 'practicumsd' is shown, each with a checkbox. All checkboxes are checked, indicating that all assets under the parent will be selected. The list includes: SalesLT.Customer, SalesLT.ProductModel, SalesLT.ProductDescription, SalesLT.Product, SalesLT.ProductModelProductDescription, SalesLT.ProductCategory, dbo.BuildVersion, dbo.ErrorLog, SalesLT.Address, SalesLT.CustomerAddress, SalesLT.SalesOrderDetail, SalesLT.SalesOrderHeader, SalesLT.vProductModelCatalogDescription, SalesLT.vProductAndDescription, and SalesLT.vGetAllCategories. At the bottom of the 'Scope your scan' section are 'Continue', 'Back', and 'Cancel' buttons.

Review and select desired data products from those listed in the “Scope your scan” popout and then click the **Continue** button.

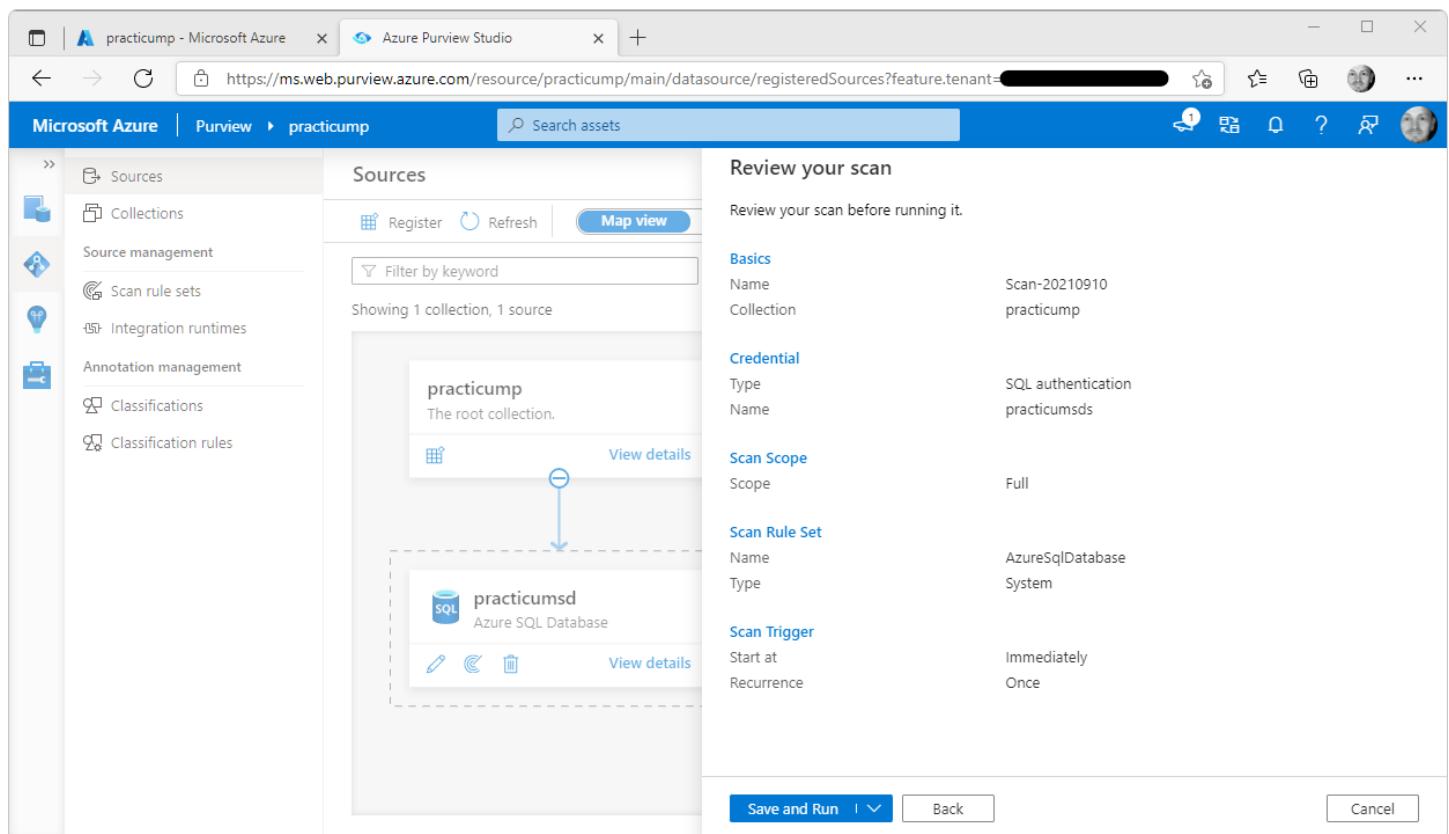
The screenshot shows the Azure Purview Studio interface. The sidebar menu is identical to the previous screenshot. The main area shows the 'practicump' collection and its sub-item 'practicumsd'. To the right, a 'Select a scan rule set' section is open. It features a '+ New scan rule set' button and a 'Refresh' button. Below this, a message says 'Select one scan rule set to be used by your scan.' A box highlights the 'AzureSqlDatabase' rule set, which is labeled as 'SYSTEM DEFAULT'. It is described as a 'Microsoft default scan rule set that includes all supported system classification rules'. A 'View detail' link is also present. At the bottom of the 'Select a scan rule set' section are 'Continue', 'Back', and 'Cancel' buttons.

Confirm selection of the default Scan Rule Set on the “Select a scan rule set” popout and then click the **Continue** button.



The screenshot shows the Azure Purview Studio interface. On the left, there's a sidebar with icons for Sources, Collections, Source management, Scan rule sets, Integration runtimes, Annotation management, Classifications, and Classification rules. The main area is titled "Sources" and shows a list with one item: "practicump". To the right, a modal window titled "Set a scan trigger" is open. It contains a description: "Set a scan trigger to run the scan at specific dates and times. If once, the scan will start after setup is completed. If recurring, the scan will start at a date and time you choose. The initial scan is a full scan and every subsequent scan is incremental." Below this, there are two radio buttons: "Recurring" and "Once". The "Once" button is selected. At the bottom of the modal are three buttons: "Continue", "Back", and "Cancel".

Click the **Once** radio button on the “Set a scan trigger” popout and then click the **Continue** button.



The screenshot shows the Azure Purview Studio interface. The left sidebar is identical to the previous screenshot. The main area shows the "practicump" collection. To the right, a modal window titled "Review your scan" is open. It has a heading "Review your scan before running it." Below this, there are several sections with details:

- Basics**: Name: Scan-20210910, Collection: practicump
- Credential**: Type: SQL authentication, Name: practicumstsds
- Scan Scope**: Scope: Full
- Scan Rule Set**: Name: AzureSqlDatabase, Type: System
- Scan Trigger**: Start at: Immediately, Recurrence: Once

At the bottom of the modal are three buttons: "Save and Run", "Back", and "Cancel".

Review scan settings and then click the “Save and Run” button.

Confirm Scan

Navigate to Sources, “Table view”.

**Sources**

Showing 1 source

Name	Source type	Collection	Source id	Scans	Registered on
AzureSqlDatabase-P3k	Azure SQL Data...	practicump	practicumsds.d...	1	09/10/21 01:59 PM

Note that the **Scans** column now shows 1.

Click on the **Name** link to drill through to details. Allow time for processing.

**practicumsd**

Azure SQL Database

New scan Edit source Delete source Refresh

**Overview** Scans

Source ID: practicumsd.database.windows.net Registered on 09/11/2021 03:18:28 AM

Scans 1 Scanned assets 17

Classified assets 15

Recent scans

Scan name	Last run status	Scan rule set	Last scan time
Scan-20210910	In progress	AzureSqlData...	09/11/21 03:25 AM

→ See all applied scans

Recent failed scans

Scan name	Status	Suggestions
No recent failed scan runs		

→ See all failed scan runs

Collection path  
practicump

Source hierarchy  
rchapter  
Subscription  
practicumrg  
Resource group

## Browse Assets

After scanning, it takes time for **Asset Insights** to reflect new data products.

We do not have to wait to browse data products, however.

Click the **Data Catalog** icon on the navigation.

The screenshot shows the Azure Purview Studio interface. The title bar reads "practicump - Microsoft Azure" and "Azure Purview Studio". The URL in the address bar is "https://ms.web.purview.azure.com/resource/practicump/?feature.tenant=[REDACTED]". The top navigation bar includes "Microsoft Azure", "Purview", and "practicump". On the left, there's a vertical sidebar with icons for Data Catalog, Data Governance, Data Quality, and Data Integration. The main area has a title "practicump" and a summary: "1 source", "No assets", and "No glossary terms". A search bar says "Search catalog". Below are three buttons: "Browse assets" (Explore assets by source type and collection), "Manage glossary" (Create and edit terms with custom term templates), and "Knowledge center" (Discover learning materials and tutorials). To the right is a network diagram showing various data assets connected.

Click on the “**Browse assets**” button.

The screenshot shows the "Browse assets" page. The title bar and URL are identical to the previous screenshot. The top navigation bar shows "Data catalog >". The main content area is titled "Browse assets" with a "Refresh" button. It offers filtering options: "By collection" (selected) and "By source type". A "Filter by keyword" input field is present. Below, it says "Showing 1 collection". A table lists one collection:

Name	Description	Assets	Collection admin
practicump	The root collection.	4	Rich Chapler

Click on the Root Collection link {i.e., name}.

The screenshot shows the Azure Purview Studio interface. The left sidebar has a tree view with icons for Data catalog, practicump (selected), Sub collection(s), Related, and Narrow results by (Classification, Contact, Label, Glossary term). The main area is titled 'Browse assets' with a 'By collection' filter selected. It shows 23 results, with the first few listed:

- Address**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/Address
- BuildVersion**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/dbo/BuildVersion
- Customer**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/Customer
- CustomerAddress**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/CustomerAddress
- dbo**: Azure SQL Schema, mssql://practicumsd.database.windows.net/practicumsds/dbo

At the bottom, there are navigation buttons for Previous, Page 1 of 1, and Next.

Explore results.

Click on the first item, **Address**.

The screenshot shows the Azure Purview Studio interface. The top navigation bar includes 'practicump - Microsoft Azure' and 'Azure Purview Studio'. The main content area displays an 'Address' asset, which is an 'Azure SQL Table'. The asset has an 'Edit' button, a 'Refresh' button, and a 'Delete' button. A blue 'Open in Power BI Desktop' button is also present. Below the asset name, there are tabs for 'Overview', 'Properties', 'Schema', 'Lineage', 'Contacts', and 'Related'. The 'Overview' tab is selected, showing the last update was on September 11, 2021, at 10:29 AM UTC by an automated scan. To the right of the asset details, there are sections for 'Collection path' (listing 'practicump'), 'Hierarchy' (showing the database structure from 'practicumsd.database.windows.net' down to 'Address'), and 'Glossary terms' (noting none exist). On the far left, a sidebar lists various data assets including 'Address', 'SalesLT', 'Customer', 'Product', 'Region', 'Employee', 'Order', and 'OrderDetail'.

Notice the automatic “**World Cities**” classification.

Drill-through the tabs {e.g., Properties, Schema, Lineage, etc.}.

#### Step 4: Confirm Success

Now that we have implemented the basics in Purview, can we answer the questions previously posed?

##### Search

Question: “**Does the data that I need exist?**”

Enter a phrase into the search box at the top center of the window {e.g., “address”}.

The screenshot shows the Azure Purview Studio interface with the URL <https://ms.web.purview.azure.com/resource/practicump/main/catalog/search?searchQueryKeyword=address&searchQueryLimit=2...>. The left sidebar has a 'Data catalog' section with filters for 'Collection', 'Classification', 'Contact', 'Label', and 'Glossary term'. The main area displays 'Search results for address' with two entries:

- Address**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/**Address**
- CustomerAddress**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/CustomerAddress

At the bottom, there are navigation buttons: < Previous, Page 1 of 1, and Next >.

Returned search results will include all data products that include the requested phrase.

## Location

Question: “**how do I connect to Data Product X?**”

Drill into the first search result, Address.

The screenshot shows the Azure Purview Studio interface with the URL <https://ms.web.purview.azure.com/resource/practicump/main/catalog/entity?guid=81afae99-9619-4559-aa95-b5f6f6f60000&secti...>. The main content area displays the 'Address' asset, which is an Azure SQL Table. The 'Overview' tab is selected, showing the following details:

- Collection path:** practicump
- Hierarchy:** A tree structure showing the database hierarchy:
  - practicumsd.database.windows.net (Azure SQL Server)
  - practicumsds (Azure SQL Database)
    - SalesLT (Azure SQL Schema)
      - Address (Azure SQL Table)

Other tabs visible include Properties, Schema, Lineage, Contacts, and Related. A note at the bottom right indicates the asset was updated on September 11, 2021, 10:29 AM UTC by automated scan.

On the **Overview** tab, you will see **Hierarchy** information which includes the Fully Qualified Domain Name of the Azure SQL Server.

## Related

Question: “**are there related data products that I might find valuable?**”

Continue in the search result for phrase “Address”.

The screenshot shows the Azure Purview Studio interface. The left sidebar has icons for Data Catalog, Purview, and other services. The main area shows a search result for "Address". The top navigation bar includes Microsoft Azure, Purview, practicum, address, and a search bar. Below the search bar are buttons for Edit, Refresh, Delete, and Open in Power BI Desktop. The main content area has tabs for Overview, Properties, Schema, Lineage, Contacts, and Related. The Related tab is selected, showing a list of 13 items under "Showing 1 to 13 of 13 items". The list includes: Address (Azure SQL Table), Customer (Azure SQL Table), CustomerAddress (Azure SQL Table), Product (Azure SQL Table), ProductCategory (Azure SQL Table), ProductDescription (Azure SQL Table), ProductModel (Azure SQL Table), ProductModelProductDescription (Azure SQL Table), SalesOrderDetail (Azure SQL Table), and SalesOrderHeader (Azure SQL Table). The status bar at the bottom right indicates the data was updated on September 11, 2021, at 10:29 AM UTC by automated scan.

On the **Related** tab, you will see a list of related data products (as identified by Purview).

## Contacts

Question: “who are the owners and experts for Data Product X?”

Continue in the search result for phrase “Address”.

The screenshot shows the Azure Purview Studio interface, similar to the previous one but with the Contacts tab selected. The main content area displays a message: "Here are the people you can contact about this asset:". Below this, there are two sections: "Experts" and "Owners". Under "Experts", it says "No experts for this asset.". Under "Owners", it says "No owners for this asset.". The rest of the interface is identical to the Related tab view, including the sidebar, top navigation, and status bar.

On the **Contacts** tab, you will see lists of **Experts** and **Owners** you can contact about the data product (as identified by Purview).

If, as in the snip above, there are no Contacts and you can add information not already captured, click the **Edit** button.

The screenshot shows the Azure Purview Studio interface for editing a data catalog entity. The URL in the browser is <https://ms.web.purview.azure.com/resource/practicump/main/catalog/entity/edit?guid=81afae99-9619-4559-aa95-b5f6f6f60000&...>. The page title is "Edit 'Address'". The "Contacts" tab is active. On the left, there are sections for "Experts" and "Owners", each with a search bar and a note about their roles. The "Experts" section shows one entry: Rich Chapter (Curator) with the email Richard.Chapter@microsoft.com. At the bottom are "Save" and "Cancel" buttons.

In either the **Experts** or the **Owners** dropdown, search for the user or group in your directory. Click **Save**.

*Objective Complete... congratulations!*

## Objective - Purview | Classify Data

Data classification is a process of annotating data products to help organizations answer the following questions:

- **Definition** ... “how does data map to a business context?” and “what is the intended use of Data Product X?”
- **Documentation** ... “where is the documentation of Data Product X?”
- **Vocabulary** ... “what else {e.g., synonyms, acronyms, etc.} might help me understand Data Product X?”
- **Standardization** ... “how do I classify all data products in a standard way?”

In this section, we will configure Purview and successfully respond to these questions.

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Purview
- SQL

### Step 2: Create Term

Navigate to your instance of Purview in the Azure portal.

Click the “Open Purview Studio” button.

Click the “Manage Glossary” button.

Click the “+ New term” button on the resulting page.

On the resulting popout, confirm selection of “**System default**” and then click the **Continue** button.

The screenshot shows the 'New term' creation page in the Azure Purview Studio. The URL in the browser is [https://ms.web.purview.azure.com/resource/practicump/main/catalog/term/new?feature.tenant=\[REDACTED\]](https://ms.web.purview.azure.com/resource/practicump/main/catalog/term/new?feature.tenant=[REDACTED]). The page has a blue header bar with the Microsoft Azure logo, 'Purview', and 'practicump'. Below the header is a search bar labeled 'Search assets'. The main content area has a title 'New term'. Under 'Term template', a dropdown menu shows 'System default'. To the right, there's a 'Status' dropdown set to 'Draft'. Below this, there are tabs for 'Overview' (which is selected), 'Related', and 'Contacts'. The 'Overview' tab contains several input fields:

- Name**: Product Number. A note below says 'Formal name: Product Number'.
- Definition**: The stock-keeping unit (SKU) that uniquely identifies a product.
- Parent**: A dropdown menu labeled 'Select...'.
- Acronym**: SKU.
- Resources**: A section with 'Resource name' (Wikipedia Definition) and 'Resource link' (https://en.wikipedia.org/wiki/Stock\_keeping\_unit). There's also a '+ Add a resource' button.

At the bottom of the page are two buttons: 'Create' (in blue) and 'Cancel'.

On the “**New term**” page, enter values for the following items:

<b>Name</b>	Enter a meaningful name aligned with standards
<b>Definition</b>	Select your subscription

Parent	Select a parent term, if applicable
Acronym	Capture abbreviation, if applicable
Resources	Click “+ Add a resource” to capture hyperlinks with descriptive content

Click the **Create** button.

### Step 3: Use Term

Search for “product” to identify data products that use terms like Product Number.

The screenshot shows the Azure Purview Studio interface with a search query of "product". The results page displays four items:

- Product**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/Product
- ProductModelProductDescription**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/ProductModelProductDescription
- ProductCategory**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/ProductCategory
- ProductDescription**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/ProductDescription

On the left sidebar, there are filters for Collection, Classification, Contact, Label, and Glossary term. The "Collection" filter has "practicump" selected. The "Sort by" dropdown is set to "Relevance".

Click **Product**.

The screenshot shows the Azure Purview Studio interface. In the top navigation bar, there are tabs for Microsoft Azure, Purview, and practicum. A search bar contains the query "product". The main content area displays the "Data catalog > Search results "product"" section. Under the "Product" category, which is identified as an "Azure SQL Table", there is a table named "Product". The "Schema" tab is selected. At the top of the schema view, there are buttons for Edit, Refresh, Delete, and "Edit columns". On the right, a blue button says "Open in Power BI Desktop". Below the buttons, there are tabs for Overview, Properties, Schema, Lineage, Contacts, and Related. The Schema tab is active. To the right of the tabs, it says "Updated on September 11, 2021 10:29 AM UTC by automated scan". The main table lists the columns of the "Product" table:

Column name	Classifications	Sensitivity label	Glossary terms	Data type	Asset description	Description
Color				nvarchar		
DiscontinuedDate				datetime		
ListPrice				money		
ModifiedDate				datetime		
Name				nvarchar		
ProductCategoryID				int		
ProductID				int		
ProductModelID				int		
ProductNumber				nvarchar		
rowguid				uniqueidentifier		

Click the **Schema** tab.

Click on the **ProductNumber** link.

The screenshot shows the Azure Purview Studio interface, similar to the previous one but focused on the "ProductNumber" column. The top navigation bar and search bar are identical. The main content area shows the "Data catalog > Search results "product"" section. Under the "Product" category, the "ProductNumber" column is selected. The "Overview" tab is selected. At the top of the overview view, there are buttons for Edit, Refresh, Delete, and "Edit columns". To the right, it says "Updated on September 11, 2021 10:29 AM UTC". The main content area displays the following details for the "ProductNumber" column:

Asset description	Collection path
No description for this asset.	practicump
Description	Hierarchy
-	
Classifications	Glossary terms
No classifications for this asset.	No glossary terms for this asset.
Fully qualified name	mssql://practicumsd.database.windows.net/practicumsds/SalesLT/Product#ProductNumber

Click the **Edit** button.

The screenshot shows the Azure Purview Studio interface. The title bar includes tabs for 'practicump - Microsoft Azure' and 'Azure Purview Studio'. The main navigation bar has 'Microsoft Azure' and 'Purview' with a 'practicump' dropdown. A search bar at the top right contains the text 'product'. The main content area is titled 'Edit "ProductNumber"'. On the left, there's a sidebar with icons for Overview, Contacts, and other data catalog items. The 'Overview' tab is selected. The main form has fields for 'Name' (ProductNumber), 'Asset description' (empty), 'Classifications' (Select...), and 'Glossary terms' (dropdown menu showing 'Product Number' with a checked checkbox). At the bottom are 'Save' and 'Cancel' buttons.

Select the new term, “**Product Number**” from the “**Glossary terms**” dropdown.

Click the **Save** button.

## Step 4: Confirm Success

Now that we have implemented the basics in Purview, can we answer the questions previously posed?

### Definition

Questions: “**how does data map to a business context?**” and “**what is the intended use of Data Product X?**”

The screenshot shows the Azure Purview Studio interface. The URL in the address bar is <https://ms.web.purview.azure.com/resource/practicump/main/catalog/term?section=overview&termGuid=52836041-66f9-43d5-8...>. The page title is "Azure Purview Studio". The main content area displays a term named "Product Number". The term is marked as "Term | System default" and is currently in "Draft" status. There are buttons for "Edit", "Delete", and "Refresh". The "Overview" tab is selected, showing the formal name "Product Number" and the parent term information ("No parent term for this term"). The "Definition" section contains the text: "The stock-keeping unit (SKU) that uniquely identifies a product.". Below this, the "Created on" and "Last updated" details are shown, both by "Rich Chapter" on 09/13/2021 at 07:37. The "Acronym" field is "SKU", and the "Catalog assets" section indicates "1 asset associated with "Product Number"" with a "View asset" link. The "Resources" section includes a "Wikipedia Definition" link.

This also demonstrates response to questions:

- **Documentation** ... “where is the documentation of Data Product X?
  - Links to documentation can be included in **Resources**
- **Vocabulary** ... “what else {e.g., synonyms, acronyms, etc.} might help me understand Data Product X?”

### Standardization

Question: “**how do I classify all data products in a standard way?**”

The screenshot shows the Azure Purview Studio interface for editing the schema of an 'Address' entity. The top navigation bar includes 'practicump - Microsoft Azure', 'Azure Purview Studio', and the URL 'https://ms.web.purview.azure.com/resource/practicump/main/catalog/entity/edit?guid=81afae99-9619-4559-aa95-b5f6f6f60000&...'. The main area is titled 'Edit "Address"' and has tabs for 'Overview', 'Schema' (which is selected), and 'Contacts'. A note at the top says, 'Making a manual update to the schema will prevent future scans on this asset from updating it.' The schema table lists columns: AddressID, AddressLine1, AddressLine2, City, CountryRegion, and ModifiedDate. The 'CountryRegion' column is currently selected. The table also includes 'Common Passwords' and 'Glossary terms' columns, and columns for 'Data type' and 'Asset description'. At the bottom are 'Save' and 'Cancel' buttons.

Use “**Column level classification**” options to characterize all country- / region-type schema elements as a standard classification.

*Objective Complete... congratulations!*

## Objective - Purview | Understand Lineage

Data lineage describes provenance {i.e., where data comes from, where it ends up and what happens to it along the way}.

Understanding data lineage enables stakeholders to answer the following question:

- **Accountability** (aka Impact Analysis) ... “who and what would be affected by a change to or problem with Data Product X?”

In this section, we will configure Purview and successfully respond to this question.

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Data Factory
- Data Share
- Purview
- SQL

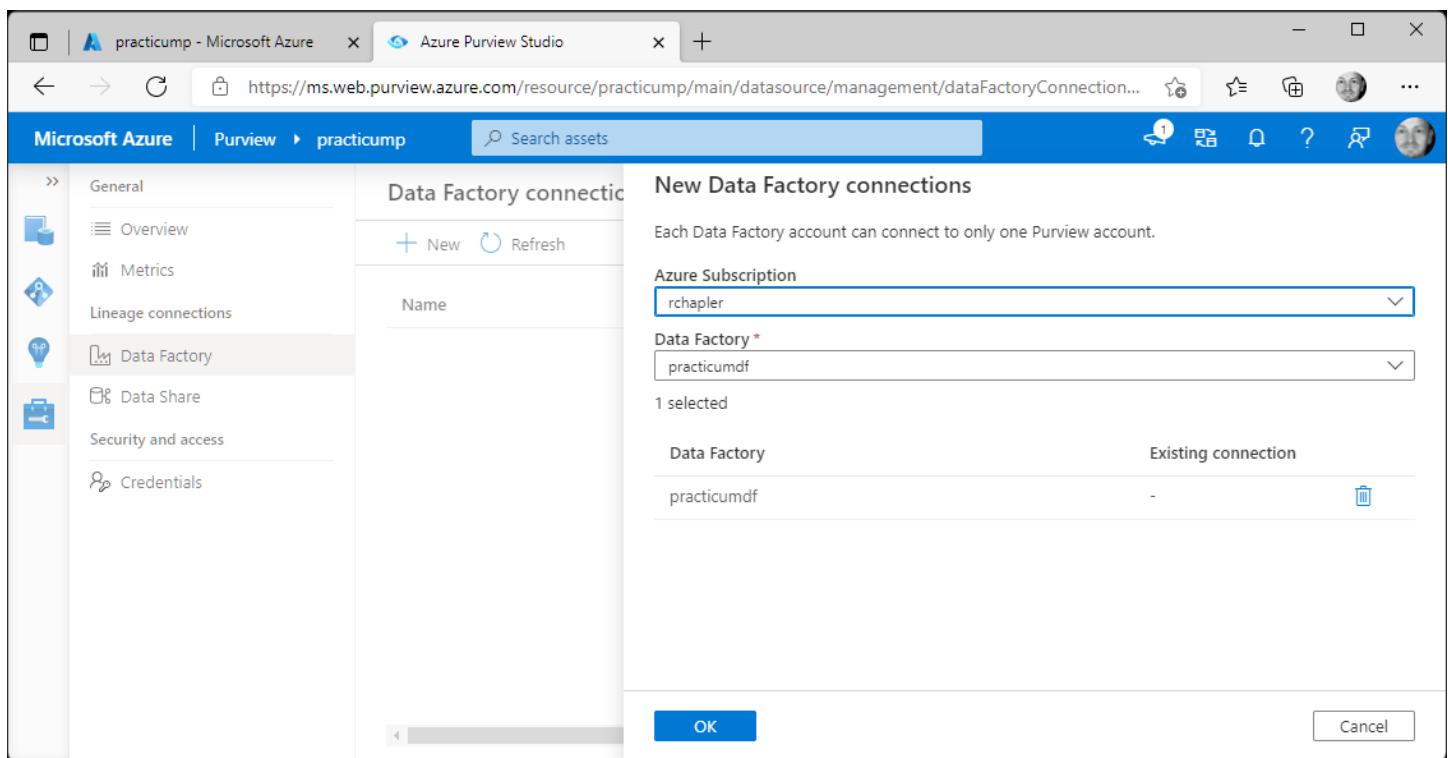
### Step 2: Configure Data Factory

Navigate to your instance of Purview in the Azure portal.

Click the “Open Purview Studio” button.

Click the **Management** icon on the navigation and then “**Data Factory**” from the resulting menu.

Click the “+ New” button on the resulting page.



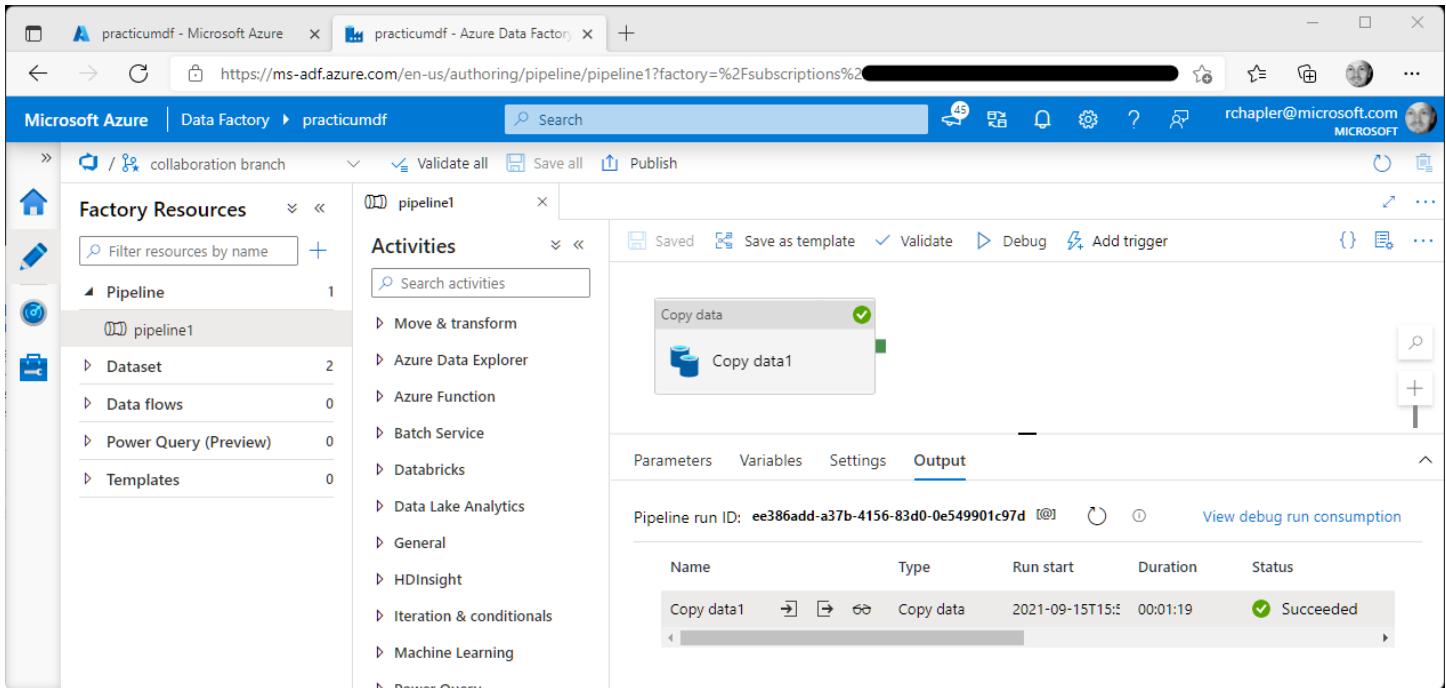
On the “**New Data Factory connections**” popout, enter values for the following items:

Azure Subscription	Select your subscription
Data Factory	Select your instance of data factory

Click the **OK** button. Confirm connection.

## Confirm Success

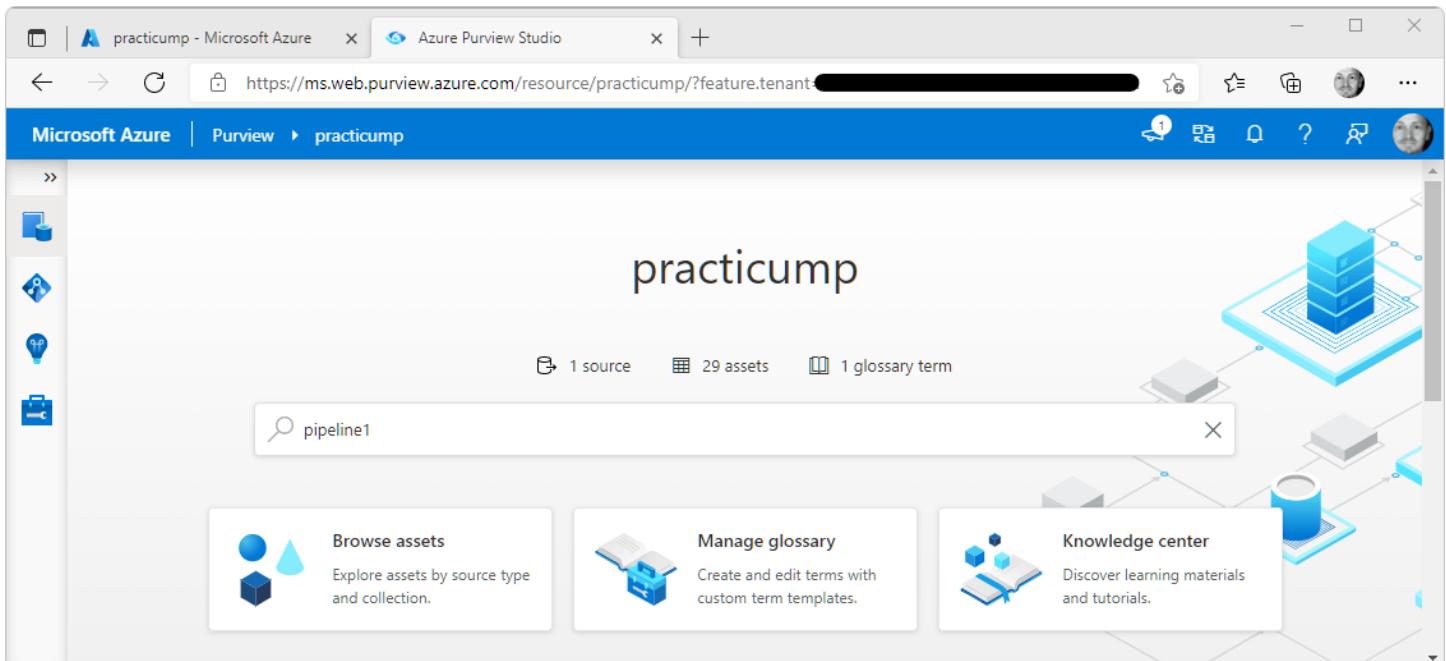
Purview will automatically detect data movement and capture data about a pipeline when it is executed in a connected data factory. Navigate to your instance of Data Factory and execute your sample pipeline.



The screenshot shows the Azure Data Factory pipeline execution results. The pipeline named 'pipeline1' contains one activity, 'Copy data1'. The activity has a green checkmark indicating success. Below the activities, the 'Output' tab is selected, showing a table of the run details:

Name	Type	Run start	Duration	Status
Copy data1	Copy data	2021-09-15T15:50:00Z	00:01:19	Succeeded

After successful execution, navigate to your instance of Purview.



The screenshot shows the Azure Purview Studio interface. The search bar at the top contains the text 'pipeline1'. The main area displays a network diagram of data assets. A callout box highlights the 'Browse assets' button, which is described as 'Explore assets by source type and collection.'

Search for "pipeline1".

The screenshot shows the Azure Purview Studio interface. The top navigation bar includes 'practicump - Microsoft Azure', 'Azure Purview Studio', and a search bar with the query 'pipeline1'. The main content area is titled 'Search results for pipeline1' and displays two results:

- Copy data1**  
Azure Data Factory Copy Activity  
/subscriptions/[REDACTED]/resourceGroups/practicumrg/providers/Microsoft.Dat...
- pipeline1**  
Azure Data Factory Pipeline  
/subscriptions/[REDACTED]/resourceGroups/practicumrg/providers/Microsoft.Dat...

On the left, there is a sidebar with filters for 'Collection' (selected 'practicump'), 'Classification', 'Contact', and 'Content type'. The bottom of the page shows pagination controls: '< Previous', 'Page 1 of 1', and 'Next >'.

Click on the “Copy data1” result.

Click on the **Lineage** tab.

The screenshot shows the Microsoft Purview Studio interface. The top navigation bar includes tabs for Microsoft Azure, Purview, and practicum. A search bar at the top right contains the query "pipeline1". Below the navigation, a breadcrumb trail shows "Data catalog > Search results "pipeline1"" and a title "Copy data1" with a subtitle "Azure Data Factory Copy Activity". On the left, a sidebar has icons for Data catalog, Purview, and practicum. The main content area has tabs for Overview, Properties, Lineage (which is selected), Contacts, and Related. A status message indicates it was updated on September 15, 2021, at 4:00 PM UTC by Azure Data Factory pipeline. A search bar at the top of the main content area is empty. The central part of the screen displays a lineage diagram. It features two "Product" assets, each represented by a blue box with a white icon. A curved arrow connects the two products, pointing from the first to the second. Between these arrows is a blue-bordered box containing the "Copy data1" activity. The box has a small icon of a cylinder with a pipe, the text "Copy data1", and "Factory: practicumdf Pipeline: pipeline1". At the bottom of this box are "See details" and "Open in Azure Data Factory" buttons. To the right of the lineage diagram is a vertical toolbar with icons for edit, refresh, delete, and other actions.

Continue to drill-through elements to better understand the capture of Lineage data.

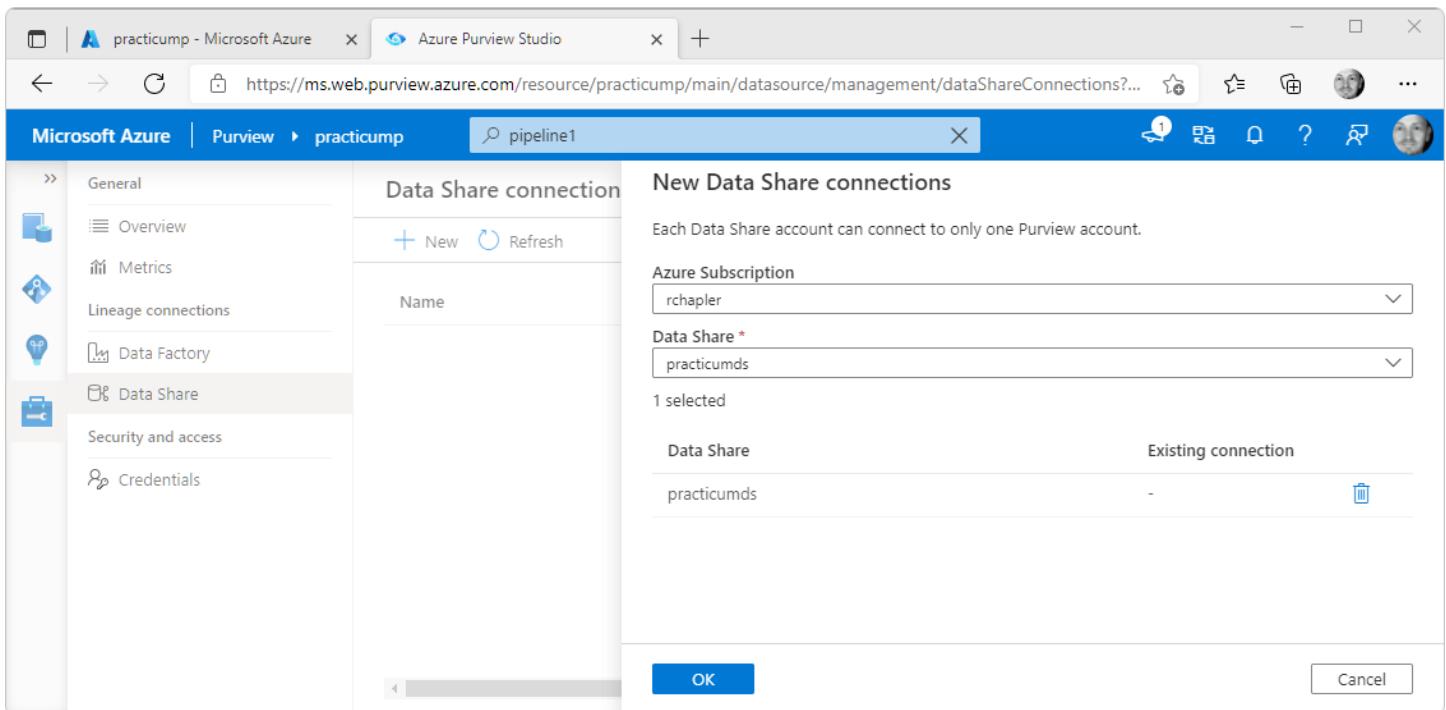
### Step 3: Configure Data Share

Navigate to your instance of Purview in the Azure portal.

Click the “Open Purview Studio” button.

Click the Management icon on the navigation and then “Data Share” from the resulting menu.

Click the “+ New” button on the resulting page.



On the “**New Data Factory connections**” popout, enter values for the following items:

Azure Subscription	Select your subscription
Data Share	Select your instance of data share

Click the **OK** button.

Article: [Connect to Azure Data Share - Azure Purview | Microsoft Docs](#)

*Note from this article... “For Data Share products to show in Purview, a snapshot job must be run after you connect your Data Share to Purview”*

*Objective Complete... congratulations!*

# Monitoring

## Objective - Synapse | Audit Data Usage

Follow the instructions in this section to support regular audit of and alerting on key data products.

### Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Synapse

### Step 2: Configure Auditing

Navigate to the **Overview** page in Synapse, Dedicated SQL Pool.

The screenshot shows the Microsoft Azure (Preview) portal interface. The URL in the address bar is <https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/91e9fddc-ef15-416c-9be1-...>. The main content area is titled "practicumdsp (practicumsaw/practicum dsp)" and is described as a "Dedicated SQL pool". On the left, there's a navigation menu with items like Overview, Activity log, Access control (IAM), Tags, Settings, Workload management, Maintenance schedule, Geo-backup policy, Connection strings, Properties, Locks, Security, Auditing, Data Discovery & Classification, Dynamic Data Masking, and Security Center. The "Auditing" item under Security is highlighted. The main panel shows resource details: Resource group (practicumsaw), Status (Online), Location (West US 2), Subscription (rchapler), Subscription ID (91e9fddc-ef15-416c-9be1-085f8b1b46ed), and Tags (Click here to add tags). Below these, there are three tabs: Notifications (0), Features (4), and Tasks (2). The "Features" tab is selected, showing two cards: "Transparent data encryption" (NOT CONFIGURED) and "Auditing" (NOT CONFIGURED). The "Auditing" card includes a description: "Track your SQL pool events and write them to an audit log in an Azure storage account." There are also "View Cost" and "JSON View" buttons at the top right of the main panel.

Click the **Auditing** button.

practicumdsp (practicumsaw/practicumdsp) | Auditing

Search (Ctrl+ /)

Save Discard View audit logs Feedback

Overview Activity log Access control (IAM) Tags

Settings Workload management Maintenance schedule Geo-backup policy Connection strings Properties Locks

If Azure SQL Auditing is enabled on the workspace, it will always apply to the SQL Pool, regardless of the SQL Pool settings.

View workspace settings

Workspace-level Auditing: **Disabled**

Azure SQL Auditing

Azure SQL Auditing tracks SQL Pool events and writes them to an audit log in your Azure storage account. [Learn more about Azure SQL Auditing](#)

Enable Azure SQL Auditing

Storage details >

practicumsa

Flip “Enable Azure SQL Auditing” switch to **ON** position. Click to configure “Storage details”.

Storage settings

Subscription rchabler

\*Storage account practicumsa

Retention (Days) 0

Storage access key Primary

OK

Save changes.

Click “View audit logs”.

Audit records - Microsoft Azure

Microsoft Azure (Preview)

Dashboard > practicumdsp (practicumsaw/practicumdsp) >

## Audit records

Refresh Filter Log Analytics View dashboard

Click here to learn more about methods for viewing & analyzing audit records.

Audit source: Workspace audit

Event time (UTC)	Principal name	Event type	Action status
2/3/2021 8:58:30 PM	##MS_InstanceCertificate##	BATCH COMPLETED	Succeeded
2/3/2021 8:56:26 PM	##MS_InstanceCertificate##	BATCH COMPLETED	Succeeded
2/3/2021 8:56:26 PM	##MS_InstanceCertificate##	DATABASE AUTHENTICATION SUCCEEDED	Succeeded

Click into one of “**BATCH COMPLETED**” audit records.

Audit record - Microsoft Azure

Microsoft Azure (Preview)

Search resources, services, and docs (G+)

Dashboard > practicumdsp (practicumsaw/practicumdsp) > Audit records >

## Audit record

practicumdsp

Event time (UTC)  
2/8/2021 7:32:50 PM

Event type  
BATCH COMPLETED

Server name  
practicumsaw

Database name  
practicumdsp

Application name  
SynapseSqlEditor

Principal name  
rchapler@microsoft.com

Client IP  
76.121.194.132

Status  
Succeeded

---

STATEMENT

```
SELECT
    s.name AS [schema_name],
    o.name AS [object_name], o.type AS [object_type], o.type_desc AS
    [object_type_desc],
    c.name AS [column_name], TYPE_NAME(c.system_type_id) AS
    [column_type]
FROM sys.schemas s
LEFT OUTER JOIN sys.all_objects o
    ON s.schema_id = o.schema_id
LEFT OUTER JOIN sys.all_columns c
    on c.object_id = o.object_id
```

Expand statement

Objective Complete... congratulations!

# Infrastructure

Objective(s) in this section describe instantiation of basic resources necessary to complete other exercises (using methods other than ARM Template / DevOps) and are presented alphabetically.

Consider instantiating only necessary resources {i.e., required for a specific exercise}.

## Application Registration (aka Service Principal)

To complete this objective, you must instantiate the following resources:

- Data Lake (with container)

Navigate to “Azure Active Directory” > “App Registrations”.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft logo, a search bar, and user information (rchapler@microsoft.com). Below the header, the URL https://ms.portal.azure.com/#blade/Microsoft\_AAD\_IAM/ActiveDirectoryMenuBlade/Registrations is visible. The main content area is titled "Microsoft | App registrations" under "Azure Active Directory". On the left, a sidebar titled "Manage" lists several options: Overview, Preview features, Diagnose and solve problems, Users, Groups, External Identities, Roles and administrators, Administrative units, Enterprise applications, Devices, App registrations (which is selected and highlighted in grey), and Identity Governance. The main pane shows a warning message: "If you are building an application for external users that will be distributed by Microsoft, you must register as a first party application to meet all security, privacy, and compliance policies. Read our decision guide". Below this, tabs for "All applications", "Owned applications" (which is selected and underlined in blue), and "Deleted applications (Preview)" are present. A search bar at the bottom of the main pane contains the placeholder text "Start typing a name or Application ID to filter these results". A message below the search bar states "This account isn't listed as an owner of any applications in this directory." and a blue button labeled "View all applications in the directory" is shown.

Click the “+ New Registration” button.

The screenshot shows the 'Register an application' page in the Microsoft Azure portal. The 'Name' field is filled with 'practicum'. Under 'Supported account types', the first option ('Accounts in this organizational directory only (Microsoft only - Single tenant)') is selected. The 'Redirect URI (optional)' field contains 'Web' and 'e.g. https://example.com/auth'. A note at the bottom states: 'By proceeding, you agree to the Microsoft Platform Policies'. A blue 'Register' button is visible.

Register an application

If you are building an application for external users that will be distributed by Microsoft, you must register as a first party application to meet all security, privacy, and compliance policies. [Read our decision guide](#)

\* Name

The user-facing display name for this application (this can be changed later).

practicum

Supported account types

Who can use this application or access this API?

Accounts in this organizational directory only (Microsoft only - Single tenant)  
 Accounts in any organizational directory (Any Azure AD directory - Multitenant)  
 Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)  
 Personal Microsoft accounts only

Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web e.g. https://example.com/auth

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the Microsoft Platform Policies

[Register](#)

On the “**Register an application**” page, enter values for the following items:

Name	Enter a meaningful name aligned with standards
------	--

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the **Register** button.

The screenshot shows the Microsoft Azure portal interface. The left sidebar has a dark theme with the following navigation items:

- Overview
- Quickstart
- Integration assistant
- Manage** (selected)
- Branding
- Authentication
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API

The main content area displays the application registration details for "practicumar". The "Essentials" section includes:

Setting	Value	Action
Display name	practicumar	Client credentials Add a certificate or secret
Application (client) ID	[REDACTED]	Redirect URIs Add a Redirect URI
Object ID	062d3984-d6b4-4f83-abcd-35d68d49a955	Application ID URI Add an Application ID URI
Directory (tenant) ID	[REDACTED]	Managed application in local directory practicumar
Supported account types	My organization only	

Make note of the “Application (client) ID” and “Directory (tenant) ID” values for use in the next section.

## Client Identifier

Navigate to the App Registration, Overview page.

The screenshot shows the Microsoft Azure portal interface, identical to the previous one but with different values in the "Essentials" section. The "Manage" item is still selected in the sidebar.

The "Essentials" section now shows the following values:

Setting	Value	Action
Display name	practicumar	Client credentials 0 certificate, 1 secret
Application (client) ID	93de551e-b998-4b1e-8930-3d9d8ac4757e	Redirect URLs Add a Redirect URI
Object ID	062d3984-d6b4-4f83-abcd-35d68d49a955	Application ID URI Add an Application ID URI
Directory (tenant) ID	[REDACTED]	Managed application in local directory practicumar
Supported account types	My organization only	

Copy the “Directory (tenant) ID” and “Application (client) ID” values.

## Key Vault Secrets, Tenant and Client Identifiers

Open a new tab and navigate to your instance of Key Vault.

Click **Secrets** in the **Settings** group of the navigation.

Click the “**+ Generate/Import**” button.

The screenshot shows the 'Create a secret' page in the Microsoft Azure (Preview) portal. The 'Name' field is set to 'practicumar-clientid'. The 'Value' field contains a redacted value. The 'Enabled' switch is set to 'Yes'. There are no tags assigned. The 'Create' button is visible at the bottom.

On the “**Create a secret**” page, enter values for the following items:

<b>Name</b>	Enter a meaningful name aligned with standards
<b>Value</b>	Paste the previously copied value

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the **Create** button.

Repeat for Tenant Id.

## Client Secret

In the **Manage** grouping, click “**Certificates & secrets**”.

Navigate to “**Certificates & secrets**” in the **Manage** group in the navigation.

The screenshot shows the Microsoft Azure portal interface. The URL in the address bar is [https://ms.portal.azure.com/#blade/Microsoft\\_AAD\\_RegisteredApps/ApplicationMenuBlade/Credentials/q...](https://ms.portal.azure.com/#blade/Microsoft_AAD_RegisteredApps/ApplicationMenuBlade/Credentials/q...). The top navigation bar includes the Microsoft Azure logo, a search bar, and a user profile for rchapler@microsoft.com.

The main content area is titled "practicumar | Certificates & secrets". On the left, there's a sidebar under "Manage" with the following items:

- Overview
- Quickstart
- Integration assistant
- Branding
- Authentication
- Certificates & secrets** (selected)
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators | Preview
- Manifest

**Certificates**

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

**Upload certificate**

Thumbprint	Start date	Expires	Certificate ID
No certificates have been added for this application.			

**Client secrets**

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

**New client secret**

Description	Expires	Value	Secret ID
No client secrets have been created for this application.			

Click the “+ New client secret” button, and then in the resulting “Add a client secret” popout, click the **Add** button.

**practicumar | Certificates & secrets**

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Upload certificate

Thumbprint	Start date	Expires	Certificate ID
No certificates have been added for this application.			

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

New client secret

Description	Expires	Value	Secret ID
Password uploaded on Fri Sep ...	3/24/2022	[REDACTED]	[REDACTED]

Copy Value.

## Key Vault Secret, Client Secret

Open a new tab and navigate to your instance of Key Vault.

Click **Secrets** in the **Settings** group of the navigation.

Click the “+ Generate/Import” button.

The screenshot shows the Microsoft Azure portal's "Create a secret" interface. Key fields include:

- Upload options: Manual
- Name: practicumar-clientsecret
- Value: (redacted)
- Content type (optional): (empty)
- Set activation date: (checkbox)
- Set expiration date: (checkbox)
- Enabled: Yes (selected)
- Tags: 0 tags

A blue "Create" button is at the bottom left.

On the “Create a secret” page, enter values for the following items:

Name	Enter a meaningful name aligned with standards
Value	Paste the previously copied value

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the **Create** button.

## Permissions, Data Lake

Navigate to your instance of data lake, your container and then “Access Control (IAM)” in navigation.

Click the “+ Add” button and then “Add role assignment” in the resulting dropdown.

The screenshot shows the Microsoft Azure (Preview) portal with the URL [https://ms.portal.azure.com/#blade/Microsoft\\_Azure\\_Storage/ContainerMenuBlade/accesscontrol/storage...](https://ms.portal.azure.com/#blade/Microsoft_Azure_Storage/ContainerMenuBlade/accesscontrol/storage...). The left sidebar shows a navigation path: Dashboard > practicumrg > practicuml > practicuml. The 'Access Control (IAM)' section is selected. The main content area displays the 'Add role assignment' dialog. The 'Role assignments' tab is selected. The 'Number of role assignments for this subscription' is 13, with a total limit of 2000. A table lists roles: Contributor, Owner, Reader, Storage Blob Data Contributor, Storage Blob Data Reader, and User Access Administrator. The 'Selected members:' section shows 'practicumar' selected. The 'Save' and 'Discard' buttons are at the bottom.

On the “Add role assignment” popout, enter values for the following items:

Role	Select “Storage Blob Data Reader”
Assign access to	Confirm default selection, “User, group, or service principal”
Select	Search for and select your application registration

Click the **Save** Button.

## Command Line Interface (CLI)

Browse to [Install the Azure CLI for Windows | Microsoft Docs](#)

Download, open, and install the current release of the Azure CLI.

## Data Lake

To complete this objective, you must instantiate the following resources:

- Resource Group
- Storage Account

**Create storage account**

**Basics**   [Networking](#)   [Data protection](#)   [Advanced](#)   [Tags](#)   [Review + create](#)

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below.

[Learn more about Azure storage accounts](#)

**Project details**

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

Subscription \*

Resource group \*  [Create new](#)

**Instance details**

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

Storage account name \*

Location \*

Performance  Standard  Premium

Account kind

Replication

[Review + create](#)   [< Previous](#)   [Next : Networking >](#)

On the “Create Storage Account” page, enter values for the following items:

<b>Subscription</b>	Select your subscription
<b>Resource Group</b>	Select your resource group
<b>Storage Account Name</b>	Enter a meaningful name aligned with standards
<b>Location</b>	Match resource group selection
<b>Performance</b>	Confirm default radio button selection, “Standard”
<b>Account Kind</b>	Confirm default selection, “StorageV2 (general purpose v2)”
<b>Replication</b>	Confirm default selection, “Read-access geo-redundant storage (RA-GRS)”

Navigate to the **Advanced** tab.

The screenshot shows the Microsoft Azure portal's 'Create storage account' interface. The 'Advanced' tab is active. In the 'Security' section, 'Allow Blob public access' is set to 'Enabled'. In the 'Data Lake Storage Gen2' section, 'Hierarchical namespace' is set to 'Enabled'. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next : Tags >'. The URL in the browser is <https://ms.portal.azure.com/#create/Microsoft.StorageAccount-ARM>.

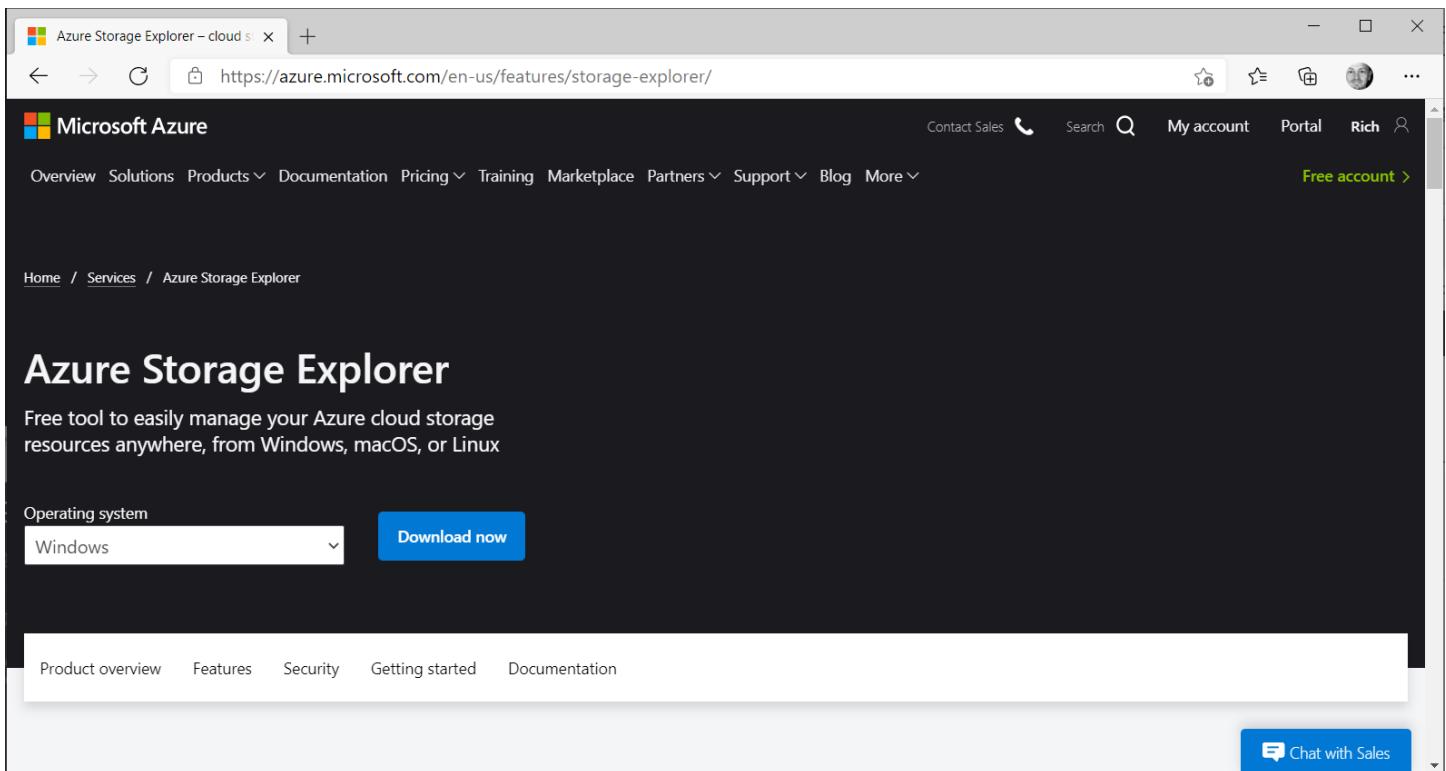
Click on the **Enabled** radio button in the “**Data Lake Storage Gen2**” grouping.

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “**Review + create**” button, review configuration, and then click the **Create** button.

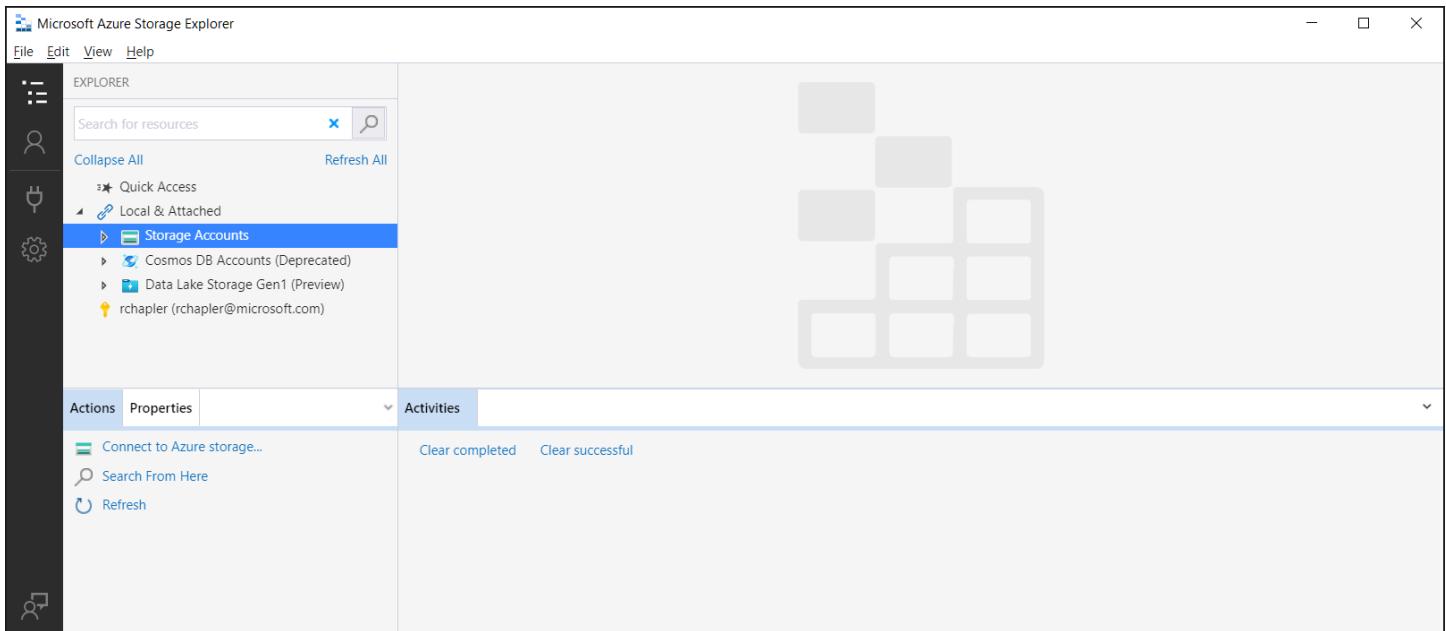
## Storage Explorer

Download and install the Microsoft Azure Storage Explorer app ([Azure Storage Explorer – cloud storage management | Microsoft Azure](#)).



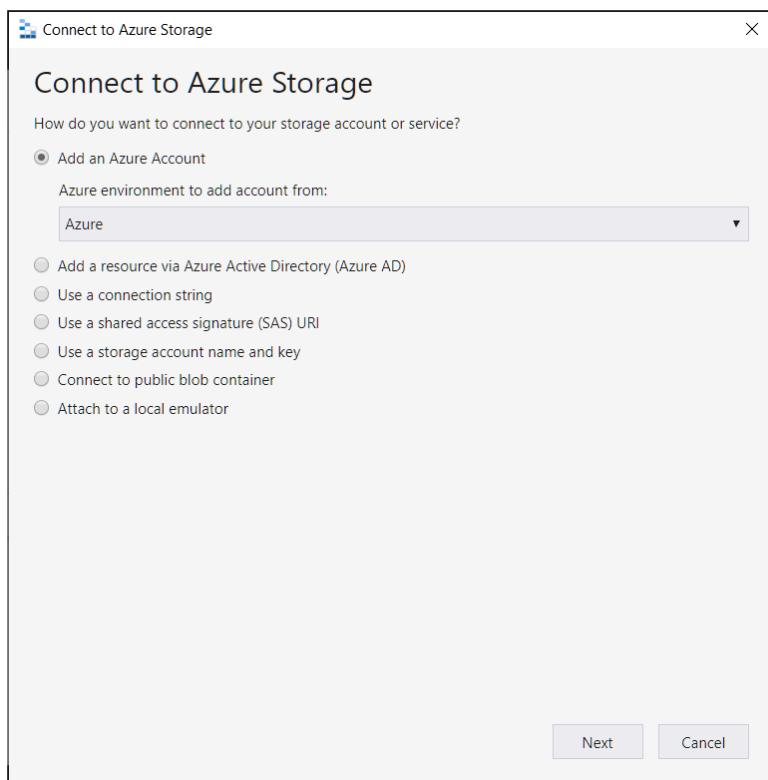
The screenshot shows the Microsoft Azure website with the URL <https://azure.microsoft.com/en-us/features/storage-explorer/>. The page title is "Azure Storage Explorer – cloud storage management". The top navigation bar includes links for Overview, Solutions, Products, Documentation, Pricing, Training, Marketplace, Partners, Support, Blog, More, and a "Free account" button. Below the navigation is a breadcrumb trail: Home / Services / Azure Storage Explorer. The main heading is "Azure Storage Explorer" with the subtext "Free tool to easily manage your Azure cloud storage resources anywhere, from Windows, macOS, or Linux". A dropdown menu for "Operating system" is set to "Windows", with a "Download now" button next to it. At the bottom of the page are links for Product overview, Features, Security, Getting started, and Documentation, along with a "Chat with Sales" button.

Launch the app and provide Azure credentials.



The screenshot shows the Microsoft Azure Storage Explorer application window. The menu bar includes File, Edit, View, and Help. The left sidebar has icons for Quick Access, Local & Attached, Storage Accounts (which is selected and highlighted in blue), Cosmos DB Accounts (Deprecated), Data Lake Storage Gen1 (Preview), and a user account (rchapler). The main pane displays a grid of placeholder icons for storage accounts. The bottom navigation bar has tabs for Actions, Properties, Activities (which is selected and highlighted in blue), and a status message: "Clear completed" and "Clear successful".

Click the “Connect to Azure storage...” link.



On the resulting popup, click the “Add an Azure Account” radio button. Click the **Next** button. Authenticate with Azure credentials.

## Container

Continue with the “Microsoft Azure Storage Explorer” app.

Use the search bar to focus on your Storage Account.

practicumdlc

Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Type	Size	Lease State
No data available in this blob container							

Showing 0 to 0 of 0 entries

Activities

Clear completed    Clear successful

Successfully created blob container 'practicumdlc'

Right-click on “Blob Containers” and click “Create Blob Container” in the resulting popup menu. Enter a meaningful name.

## Sample Data

Download sample files from a site like [Sample CSV Files Download - Get Examples Instantly \(filesamples.com\)](http://filesamples.com)

Sample CSV Files Download

CSV  
Comma Separated Values

Below you will find a selection of sample .csv document files for you to download. On the right there are some details about the file such as its size so you can best decide which one will fit your needs.

sample1.csv CSV / 502.00 B Download

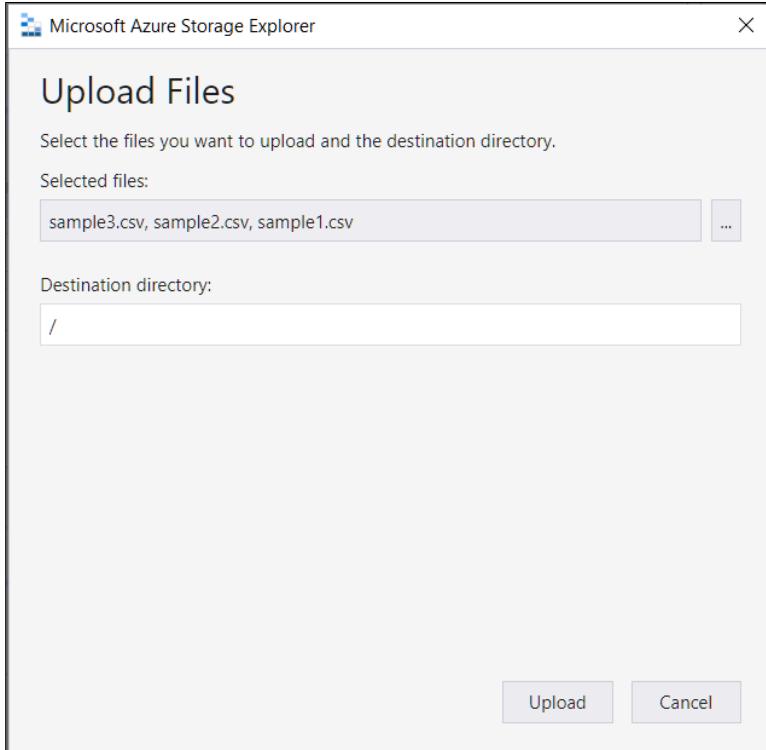
sample2.csv CSV / 55.56 KB Download

sample3.csv CSV / 723.00 B Download

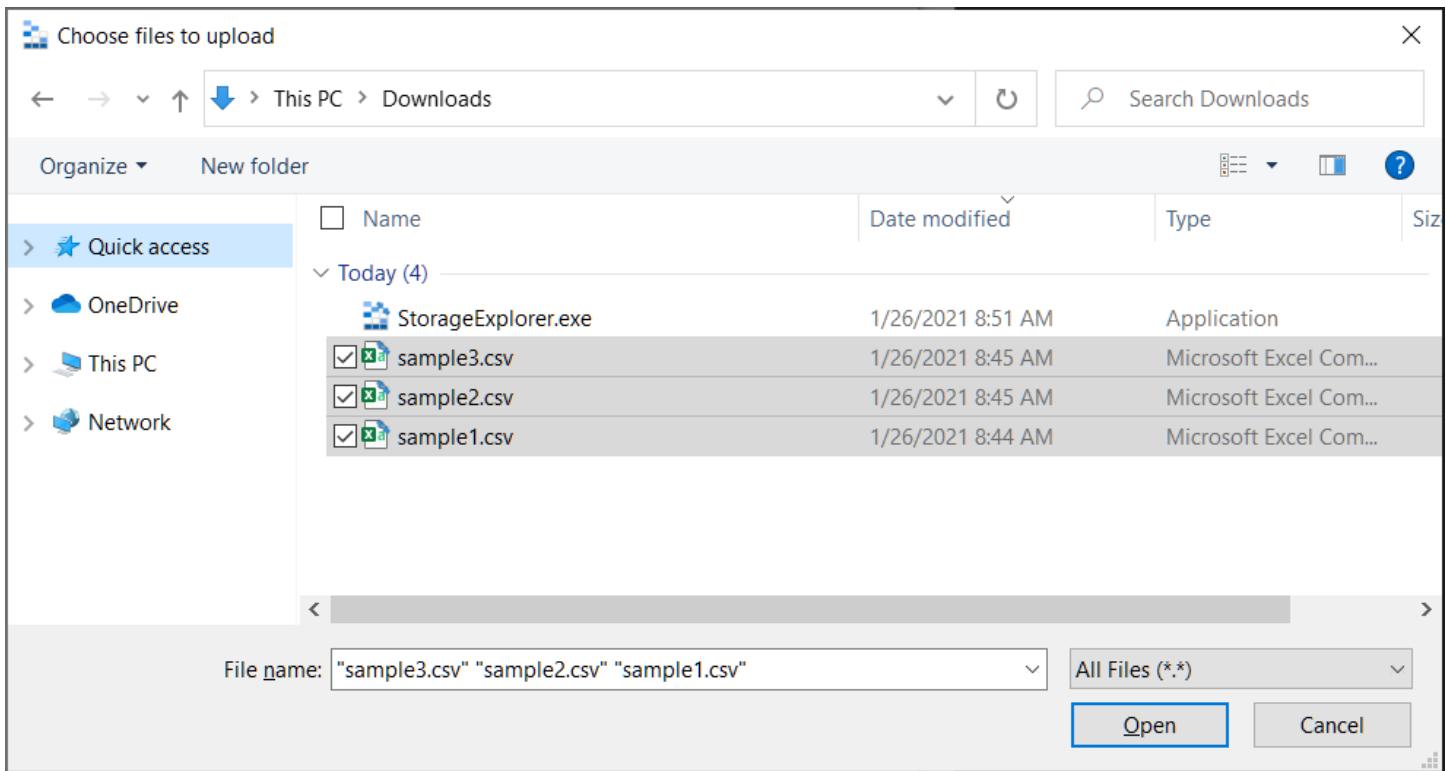
sample3.csv ... sample2.csv ... sample1.csv ... Show all X

Return to the “Microsoft Azure Storage Explorer” app and navigate to your container. Click the **Upload** button and select “Upload Files” in the resulting popup menu.

Click the ellipses button to the right of the “Selected Files” box.



Navigate to **Downloads** and select your sample data files.



Click the **Open** button.

On the “Upload Files” popup, click the **Upload** button. Allow time for file transfer.

The screenshot shows the Microsoft Azure Storage Explorer interface. The left sidebar shows a navigation tree with "practicumdl" selected. The main pane shows a blob container named "practicumdlc" with three files: "sample1.csv", "sample2.csv", and "sample3.csv". The "Actions" tab is active, displaying properties like URL, Custom Domain, Type, HNS Enabled, DFS Endpoint, Lease State, Lease Status, and Supports Tags. The "Properties" tab is also visible. At the bottom, a status message says "Transfer from 'C:\Users\rchapler\Downloads\' to 'practicumdlc/' complete: 3 items transferred (used SAS, discovery completed) Started at 2/3/2021, 12:11:00 PM Duration: 2 seconds".

## Data Migration Assistant

Browse to [Download Microsoft® Data Migration Assistant v5.5 from Official Microsoft Download Center](#).

Download, open, and install the “User Installer” for your platform.

## Data Share

To complete this objective, you must instantiate the following resources:

- Data Explorer (2 instances)
- Resource Group

The screenshot shows the Microsoft Azure portal interface for creating a Data Share. The top navigation bar includes the Azure logo, a title 'Create Data Share - Microsoft Az...', a search bar 'Search resources, services, and docs (G+)', and a user profile 'rchapler@microsoft.com MICROSOFT'. Below the navigation is a breadcrumb trail: Dashboard > practicumrg > Create a resource > Marketplace > Data Share > Create Data Share. The main content area is titled 'Create Data Share' with tabs for 'Basics', 'Tags', and 'Review + create'. The 'Basics' tab is active. Under 'Project details', it says 'Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.' It shows a subscription dropdown set to 'rchapler' and a resource group dropdown set to 'practicumrg' with a 'Create new' option. Under 'Instance details', it shows a location dropdown set to 'West US 2' and a name input field containing 'practicumds' with a green checkmark. At the bottom are buttons for 'Review + create', '< Previous', and 'Next : Tags >'.

On the “Create Data Share” page, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your resource group
Location	Match resource group selection
Name	Enter a meaningful name aligned with standards

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “Review + create” button, validate settings, and then click the **Create** button.

After deployment is complete, click the “Go to resource” button.

The screenshot shows the Microsoft Azure Data Share blade for the 'practicumds' resource. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, and Diagnose and solve problems under the 'Overview' section; Properties, Locks, Sent Shares, and Received Shares under 'Data Share'. The main content area displays the 'Essentials' section with details: Resource group (practicumrg), Location (West US 2), Subscription (rchapler), and Subscription ID. It also shows a 'Tags (change)' section with a link to 'Click here to add tags'. Below these are two large buttons: 'Start sharing your data' and 'View received shares'.

## Sample Share

*Note: To demonstrate sharing data to a target, you must instantiate a second ADX cluster.*

Click the “Start sharing your data” button.

On the resulting “Sent Shares” page, click the “+ Create” button.

The screenshot shows the Microsoft Azure (Preview) portal with the URL <https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions>. The user is logged in as rchapler@microsoft.com. The main content area displays the "practicumds | Sent Shares" page under "Microsoft.DataShare-20210915092919 > practicumds". A navigation bar at the top includes tabs for "1. Details", "2. Datasets", "3. Recipients", and "4. Review + Create". On the left, a sidebar lists "Overview", "Activity log", "Access control (IAM)", "Tags", "Diagnose and solve problems", "Settings" (with "Properties" and "Locks" options), "Data Share" (selected), "Sent Shares" (highlighted), "Received Shares", and "Feedback". Under "Monitoring", there are "Alerts", "Metrics", "Diagnostic settings", and "Logs". The "1. Details" tab is active, showing fields for "Share name" (containing "practicumdec"), "Share type" (set to "In-place" with a dropdown arrow), "Description" (with placeholder text "Enter description for the share"), and "Terms of use" (with placeholder text "Enter terms of use for the share"). At the bottom are "Cancel" and "Continue" buttons.

On the “Sent Shares” page, “1. Details” tab, enter values for the following items:

Share Name	Enter a meaningful name aligned with standards
Share Type	Select “In-place” from the dropdown menu

Click the **Continue** button.

practicumds | Sent Shares

Dashboard > Microsoft.DataShare-20210915092919 > practicumds

1. Details   2. Datasets   3. Recipients   4. Review + Create

Search (Ctrl+ /)

Overview  
Activity log  
Access control (IAM)  
Tags  
Diagnose and solve problems

Settings  
Properties  
Locks

Data Share  
Sent Shares

Add datasets

Select datasets to be shared. You must have permission to add role assignment to the data store. This permission exists in the Owner role. See Share Your Data tutorial for details

Previous Continue

On the “Sent Shares” page, “2. Datasets” tab, click the “Add datasets” button.

practicumds | Sent Shares

Dashboard > Microsoft.DataShare-20210915092919 > practicumds

1. Details   2. Datasets   3. Recipients

Select dataset type

Azure Data Explorer

Cancel Next

On the resulting “Select dataset type” popout, click the “Azure Data Explorer” button, then click the **Next** button.

The screenshot shows the Microsoft Azure (Preview) portal interface. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Properties, Locks, Data Share (Sent Shares selected), Received Shares, Feedback, Monitoring, Alerts, Metrics, Diagnostic settings, and Logs. The main content area shows a progress bar with steps 1. Details, 2. Datasets (selected), and 3. Recipients. A modal window titled "Azure Data Explorer" is open, prompting the user to "Select datasets". It lists "Subscriptions" (rchapler), "Resource groups" (practicumrg), and "Azure data explorer clusters" (practicumdec2). Navigation buttons "Previous" and "Next" are located at the bottom of the main page and the modal window.

On the resulting “**Azure Data Explorer**” popout, enter values for the following items:

<b>Subscriptions</b>	Select your subscription
<b>Resource Groups</b>	Select your resource group
<b>Azure Data Explorer Clusters</b>	Select your Data Explorer cluster
<b>Name</b>	Enter a meaningful name aligned with standards

Click the **Next** button.

The screenshot shows the Microsoft Azure portal interface for creating a Data Share. The main navigation bar includes 'Dashboard', 'Microsoft DataShare-20210915092919', and 'practicumds'. On the left sidebar, under 'Data Share', the 'Sent Shares' tab is selected. The main content area is divided into three steps: '1. Details', '2. Datasets' (which is active), and '3. Recipients'. A modal window titled 'Azure Data Explorer' is open, showing a list of datasets under the cluster 'practicumdec2'. One dataset, 'practicumdec2', is selected with a checkmark. At the bottom of the modal are 'Previous' and 'Next' buttons.

On the second "...Select datasets" popout, check the box next to your cluster and then click the **Next** button.

The screenshot shows the 'Azure Data Explorer' modal window from the previous step. It has a header 'Rename datasets' and instructions: 'Provide names for your datasets. This is the name that your data consumer will see when they accept the share. Dataset names must be unique.' Below this, there is a table with two columns: 'Dataset name' and 'Path'. A single row is present, showing 'practicumdec2' in both columns. At the bottom of the modal are 'Previous' and 'Add datasets' buttons.

On the "...Rename datasets" popout, confirm the automatically generated "Dataset name" value.

Click the "Add datasets" button.

practicums | Sent Shares

Dashboard > Microsoft.DataShare-20210915092919 > practicums

1. Details    2. Datasets    3. Recipients    4. Review + Create

Add Datasets    Edit dataset names    Remove All

Filter by name...

Datasets ↑	Type	Path
practicumdec2	Azure Data Explorer Cluster	practicumdec2/

Previous    Continue

Back on the “Sent Shares” page, “2. Datasets” tab, click the **Continue** button.

practicums | Sent Shares

Dashboard > Microsoft.DataShare-20210915092919 > practicums

1. Details    2. Datasets    3. Recipients    4. Review + Create

Enter email address of the recipients for the share. Please ensure you are using recipient's Azure login email.

Add recipient    Delete All    Update expiration for all

Email	Share expiration
rchapler@microsoft.com	Fri Oct 15 2021

Previous    Continue

On the “Sent Shares” page, “3. Recipients” tab, click the “Add recipient” button.

In the resulting interface, enter an **Email** value.

Click the **Continue** button.

practicumds - Microsoft Azure

Microsoft Azure (Preview)

Dashboard > Microsoft.DataShare-20210915092919 > practicumds

practicumds | Sent Shares

Search (Ctrl+ /)

1. Details    2. Datasets    3. Recipients    4. Review + Create

Share Contents  
Number of datasets: 1

Settings  
Name of data share: practicumdec  
Description: -  
Terms of use: -

Recipients  
Number of recipients: 1

Previous    Create

On the “Sent Shares” page, “4. Review + Create” tab, click the **Create** button.

The designated email recipient can expect to receive an email like the one below:

Azure Data Share invitation from Rich Chapler - Message (HTML) Search

File **Message** Help

Delete Respond Share to Teams Quick Steps Move Tags Editing Immersive Translate Zoom Dynamics 365 Insights Report Message Protection Reply with Meeting Poll FindTime

Teams Quick Steps Language Add-in Add-in Add-in Protection FindTime

Previous Item Next Item Quick Print

## Azure Data Share invitation from Rich Chapler

 Microsoft Azure  
To: Rich Chapler

If there are problems with how this message is displayed, click here to view it in a web browser.

 Microsoft Azure

### You're invited to access data from Microsoft

You're receiving this email because Rich Chapler from Microsoft wants to share the following data with you.

Share name: practicumdec

Description: NA

[View invitation >](#)

See detailed instructions on how to accept and configure your data share.  
If you're new to Azure, [create a free Azure subscription](#).

[f](#) [t](#) [y](#) [in](#)  
[Privacy Statement](#)  
Microsoft Corporation, One Microsoft Way, Redmond, WA 98052  
 Microsoft

Click the “View invitation >” link.

A Data Share Invitations - Microsoft

https://ms.portal.azure.com/#blade/Microsoft\_Azure\_DataShare/InvitationsBrowseBlade

Microsoft Azure (Preview) Search resources, services, and docs (G+/)

rchapler@microsoft.com MICROSOFT

Dashboard > Data Share Invitations

Refresh

Pending invitations sent to your Azure login email are listed. See [accept and receive data tutorial](#) for details.

Invitation	Sender	Company	Status	Received On
practicumdec	Rich Chapler	Microsoft	Pending	9/15/2021 11:02:56 AM

Click the **Invitation** link.

practicumdec - Microsoft Azure

https://ms.portal.azure.com/#blade/Microsoft\_Azure\_DataShare/InvitationsBrowseBlade

Microsoft Azure (Preview) Search resources, services, and docs (G+/)

rchapler@microsoft.com MICROSOFT

Dashboard > Data Share Invitations >

**practicumdec** ...

Invitation

From	Number of datasets
Rich Chapter	1

Company Microsoft

Description

Terms of use

**TARGET DATA SHARE ACCOUNT**

Subscription *	rchapler
Resource group *	practicumrg
Data share account *	practicumds
Received share name * ⓘ	practicumdec

Accept and configure Reject

On the resulting “...Invitation” page, enter values for the following items:

Subscriptions	Select your subscription
Resource Groups	Select your resource group
Data Share Account	Select your Data Share account
Received Share Name	Confirm default value

Click the “Accept and configure” button.

The screenshot shows the Microsoft Azure portal interface. The left sidebar has a dark theme with various navigation options. The main content area is titled "practicumds | Received Shares". It displays the following details:

Source share	Provider
practicumdec	Rich Chapler

Below this, it shows:

Provider company	Shared on
Microsoft	11:14:31 AM, 9/15/2021

Other details include:

Number of source datasets	Accepted by
1	Rich Chapler

Received share status: Active (indicated by a green checkmark).

Click the **Datasets** link.

On the resulting “Received Shares...” page, **Datasets** tab, check the box next to your Data Explorer cluster.

Click the “+ Map to target” button.

The screenshot shows the Microsoft Azure portal interface. In the top left, there's a browser bar with the URL <https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions>. The main navigation bar includes 'Microsoft Azure (Preview)', a search bar, and a user profile for rchapler@microsoft.com.

The left sidebar has a dark theme with several sections:

- Overview**
- Activity log**
- Access control (IAM)**
- Tags**
- Diagnose and solve problems**
- Settings**
  - Properties**
  - Locks**
- Data Share**
  - Sent Shares**
  - Received Shares** (selected)
  - Feedback**
- Monitoring**
  - Alerts**
  - Metrics**
  - Diagnostic settings**

The main content area shows the 'practicumds | Received Shares' blade. It displays a list of datasets under 'Received Shares': 'practicumdec' (selected), 'practicumdec2', 'practicumdec3', and 'practicumdec4'. The 'practicumdec' dataset is currently being mapped to a target data store.

A modal dialog titled 'Map datasets to target' is open. It shows the following configuration:

- Target data type:** Azure data explorer clusters
- Subscriptions:** rchapler
- Resource groups:** practicumrg
- location:** West US 2
- Dataset:** practicumdec
- Source Type:** Azure Data Explorer Cluster
- Source Path:** practicumdec

At the bottom of the modal are 'Cancel' and 'Map to target' buttons.

On the resulting “Map datasets to target” popout, enter values for the following items:

<b>Subscriptions</b>	Select your subscription
<b>Resource Groups</b>	Select your resource group
<b>Select Kusto clusters</b>	Select your Data Explorer cluster

Click the “Map to target” button.

A screenshot of the Microsoft Azure (Preview) web interface. The URL is https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions. The top navigation bar shows 'Microsoft Azure (Preview)' and a search bar. The main content area is titled 'practicumds | Received Shares'. On the left, there's a sidebar with 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Properties', 'Locks', 'Data Share' (selected), and 'Sent Shares'. The 'Data Share' section has a 'Search (Ctrl+ /)' input field. The 'Received Shares' section shows a breadcrumb 'Received Shares > practicumdec'. The 'Datasets' tab is active, displaying a table with one row:

Datasets	Source Type	Source Path	Status
practicumdec	Azure Data Explorer Cluster	practicumdec	Mapped

## Data Studio

Browse to [Download and install Azure Data Studio - Azure Data Studio | Microsoft Docs](#)

Download, open, and install the “User Installer” for your platform.

## Add Extensions

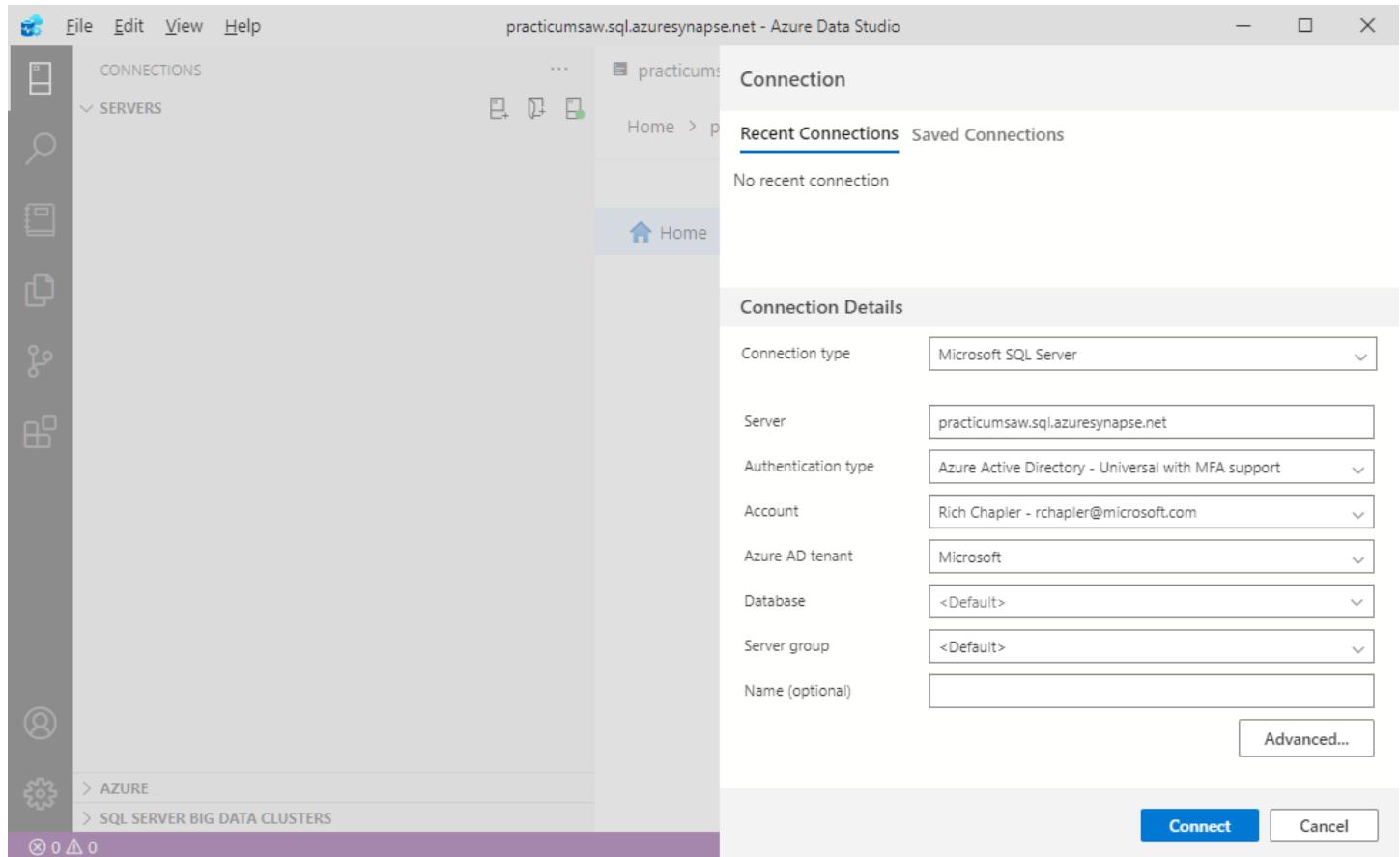
Launch Data Studio and then navigate to **Extensions** in the navigation.

A screenshot of the Azure Data Studio interface. The top menu bar includes File, Edit, View, Help, and a title bar 'Extension: SQL Server Schema Compare - Azure Data Studio'. The left sidebar has icons for file operations like Open, Save, Find, and Settings. The main area shows the 'EXTENSIONS: MARKETPLACE' view. A search bar at the top of the extension list contains the text 'schema compare'. Below it, a card for 'SQL Server Schema Compare 1.10.0' is shown, which is described as a 'Schema Compare tool for dacpac and databases' from Microsoft. An 'Install' button is visible. To the right of the extension list, a detailed view of the 'SQL Server Schema Compare' extension is displayed. It features a blue cylindrical icon with white geometric shapes. The title 'SQL Server Schema Compare' is followed by 'Microsoft | Repository | License | v1.10.0'. Below that, it says 'Schema Compare tool for dacpac and databases' and has another 'Install' button. A note states 'This extension is recommended by Azure Data Studio.' At the bottom, there's a section for 'Microsoft SQL Server Schema Compare for Azure Data Studio' with a list of included features: 'Schema Compare'. A note below it says 'The Schema Compare extension provides an easy to use experience to compare .dacpac and databases and apply the changes from source to target.' The bottom right corner of the interface has a 'Choose SQL Language' dropdown and some other small icons.

Search for “schema compare” and install the “SQL Server Schema Compare” extension.

## Create Connection

Navigate to **Connections** in the navigation.



Click the “Create Connection” icon (first of three) to the right of the **SERVERS** section header.

On the “Connection Details” popout, enter values for the following items:

<b>Connection Type</b>	Confirm default value, “Microsoft SQL Server”
<b>Server</b>	Paste the “Dedicated SQL endpoint” value copied in <a href="#">Instantiate Resources   Synapse</a>
<b>Authentication Type</b>	Select “Azure Active Directory – Universal with MFA support”
<b>Account</b>	Confirm credentials
<b>Azure AD Tenant</b>	Confirm tenant

Click the **Connect** button.

## Databricks

To complete this objective, you must instantiate the following resources:

- Resource Group

The screenshot shows the Microsoft Azure portal's 'Create an Azure Databricks workspace' wizard. The 'Basics' tab is active. Key configuration details are visible:

- Subscription:** rchapler
- Resource group:** practicumrg
- Workspace name:** practicumdb
- Region:** West US 2
- Pricing Tier:** Premium (+ Role-based access controls)

At the bottom, there are navigation buttons: 'Review + create' (highlighted in blue), '< Previous', and 'Next : Networking >'.

On the “Create an Azure Databricks workspace” page, enter values for the following items:

<b>Subscription</b>	Select your subscription
<b>Resource Group</b>	Select your resource group
<b>Workspace Name</b>	Enter a meaningful name aligned with standards
<b>Region</b>	Match resource group selection
<b>Pricing Tier</b>	Select “Premium...” to enable features that we will need for this exercise

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “Review + create” button, review configuration, and then click the **Create** button.

## New Cluster

Navigate to the **Overview** page in Databricks, click the “Launch Workspace” button, and login if required.

The screenshot shows the Azure Databricks landing page. On the left, there is a vertical sidebar with icons for Home, Workspace, Recents, Data, Clusters, Jobs, Models, and Search. The main content area features the Azure Databricks logo at the top. Below it are three main sections:

- Explore the Quickstart Tutorial**: An icon of a document with curly braces and a lightbulb. Description: Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.
- Import & Explore Data**: An icon of a dashed box with a cloud and a plus sign. Description: Quickly import data, preview its schema, create a table, and query it in a notebook.
- Create a Blank Notebook**: An icon of a document with curly braces and a plus sign. Description: Create a notebook to start querying, visualizing, and modeling your data.

Below these sections are three tabs: **Common Tasks**, **Recents**, and **Documentation**. The **Common Tasks** tab is active, showing links to New Notebook, Create Table, New Cluster, New Job, New MLflow Experiment, Import Library, and Read Documentation. The **Recents** tab shows a placeholder message: "Recent files appear here as you work." The **Documentation** tab lists Documentation, Release Notes, and Getting Started.

In the “Common Tasks” grouping, click “**New Cluster**”.

Create Cluster

New Cluster

Cancel Create Cluster

Cluster Name: practicumdbc

Cluster Mode: Standard

Pool: None

Databricks Runtime Version: Runtime: 7.4 (Scala 2.12, Spark 3.0.1)

Autopilot Options:

- Enable autoscaling
- Terminate after 120 minutes of inactivity

Worker Type: Standard\_DS3\_v2 (14.0 GB Memory, 4 Cores, 0.75 DBU)

Driver Type: Same as worker (14.0 GB Memory, 4 Cores, 0.75 DBU)

Min Workers: 2

Max Workers: 8

This Runtime version supports only Python 3.

Advanced Options

On the “Create Cluster” page, enter values for the following items:

**Cluster Name** Enter a meaningful name aligned with standards

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “Create Cluster” button.

## New Notebook

Return to the start page.

In the “Common Tasks” grouping, click “New Notebook”.

The screenshot shows the Azure Databricks interface. A modal window titled "Create Notebook" is open in the center. It contains fields for "Name" (set to "practicumdbn"), "Default Language" (set to "Python"), and "Cluster" (set to "practicumdbc"). Below the fields are two buttons: "Cancel" and "Create". The background shows the Databricks workspace with various sections like "Common Tasks", "Recents", and "Documentation".

Enter a meaningful name and click the **Create** button.

The screenshot shows the Azure Databricks notebook interface. A new notebook titled "practicumdbn (Python)" is open. The notebook has one command cell, "Cmd 1", which contains the number "1". The interface includes a toolbar at the top and a sidebar on the left with navigation links like Home, Workspace, and Recents.

Make note of the URL {i.e., <http://adb-21458...>} for use in the next section.

## Secret Scope

Navigate to your instance of Key Vault.

Click **Properties** in the **Settings** group of the navigation.

**practicumkv | Properties**

**Key vault**

**Name:** practicumkv

**Sku (Pricing tier):** Standard

**Location:** westus2

**Vault URI:** https://practicumkv.vault.azure.net/

**Resource ID:** /subscriptions/[REDACTED]/resourceGroups/practicumrg... [REDACTED]

**Subscription ID:** [REDACTED] [REDACTED]

**Subscription Name:** rchapler

**Directory ID:** [REDACTED] [REDACTED]

**Directory Name:** Microsoft

**Soft-delete:** Soft delete has been enabled on this key vault

**Days to retain deleted vaults:** 90

**Purge protection:**

Disable purge protection (allow key vault and objects to be purged during retention period)

Enable purge protection (enforce a mandatory retention period for deleted vaults and vault objects)

Make note of the values in “Vault URI” and “Resource ID”.

On a new tab, navigate to [https://\[databricksInstance\]#secrets/createScope](https://[databricksInstance]#secrets/createScope).

You will replace {databricksInstance} with the start of the URL in your workspace, from my exercise, example:

<https://adb-2154823451042175.15.azuredatabricks.net/#secrets/createScope>

Screenshot of the Microsoft Azure Databricks portal showing the "Create Secret Scope" page. The URL is https://adb-2154823451042175.15.azuredatabricks.net/?o=2154823451042175#secrets/createScope

The page displays the following fields:

- Scope Name:** practicumbdbss
- Manage Principal:** Creator
- Azure Key Vault:**
  - DNS Name:** https://practicumkv.vault.azure.net/
  - Resource ID:** /46ed/resourceGroups/practicumrg/providers/Microsoft.KeyVault/vaults/practicumkv

Buttons: Cancel, Create

On the “Create Secret Scope” page, enter values for the following items:

Scope Name	Enter a meaningful name aligned with standards
DNS Name	Paste the copied “Vault UI” value
Resource ID	Paste the copied “Resource ID” value

Click the **Create** button.

## DevOps

Browse to <https://azure.microsoft.com/en-us/services/devops/>

Screenshot of the Microsoft Azure DevOps Services homepage. The URL is https://azure.microsoft.com/en-us/services/devops/

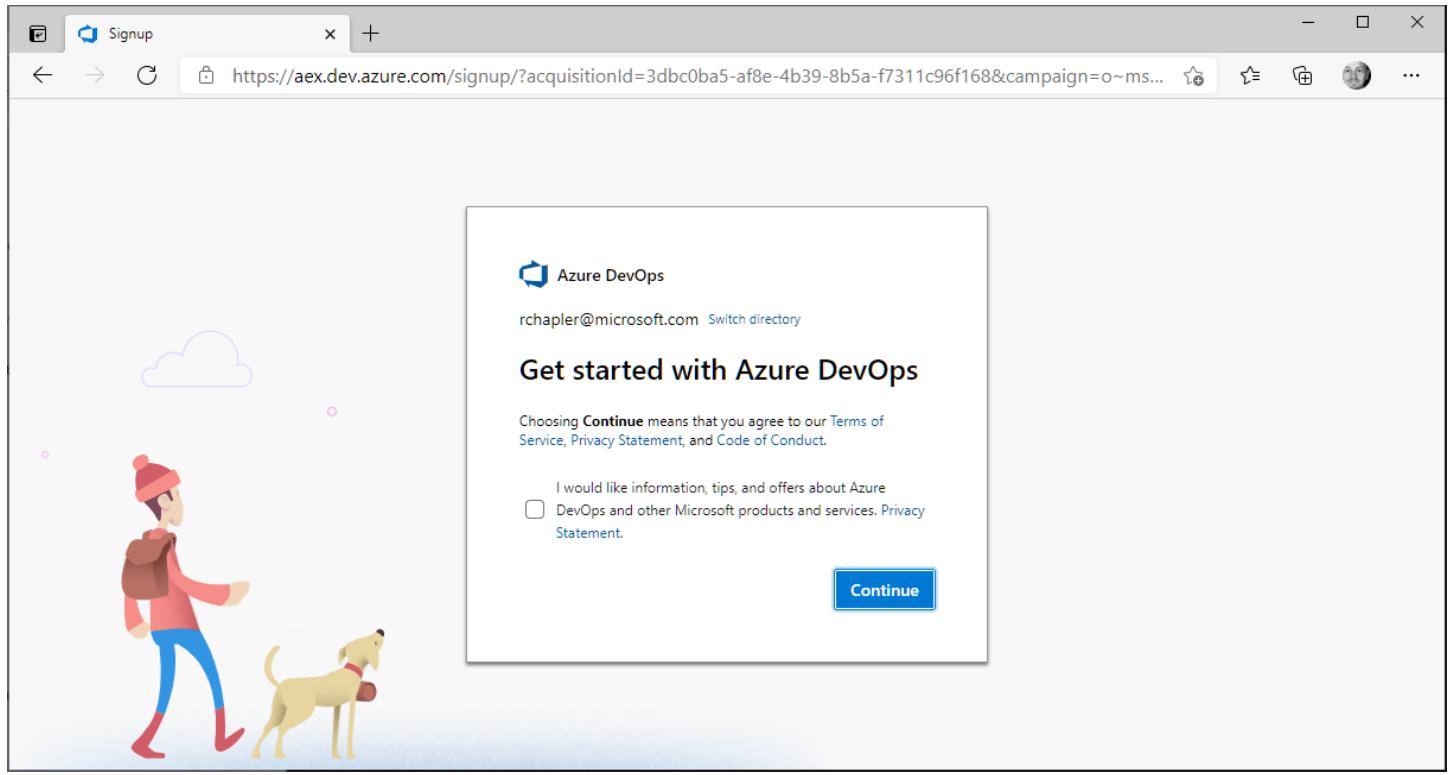
The page features a large illustration of a rocket launching from a platform with people working on it, set against a background of clouds.

Key elements include:

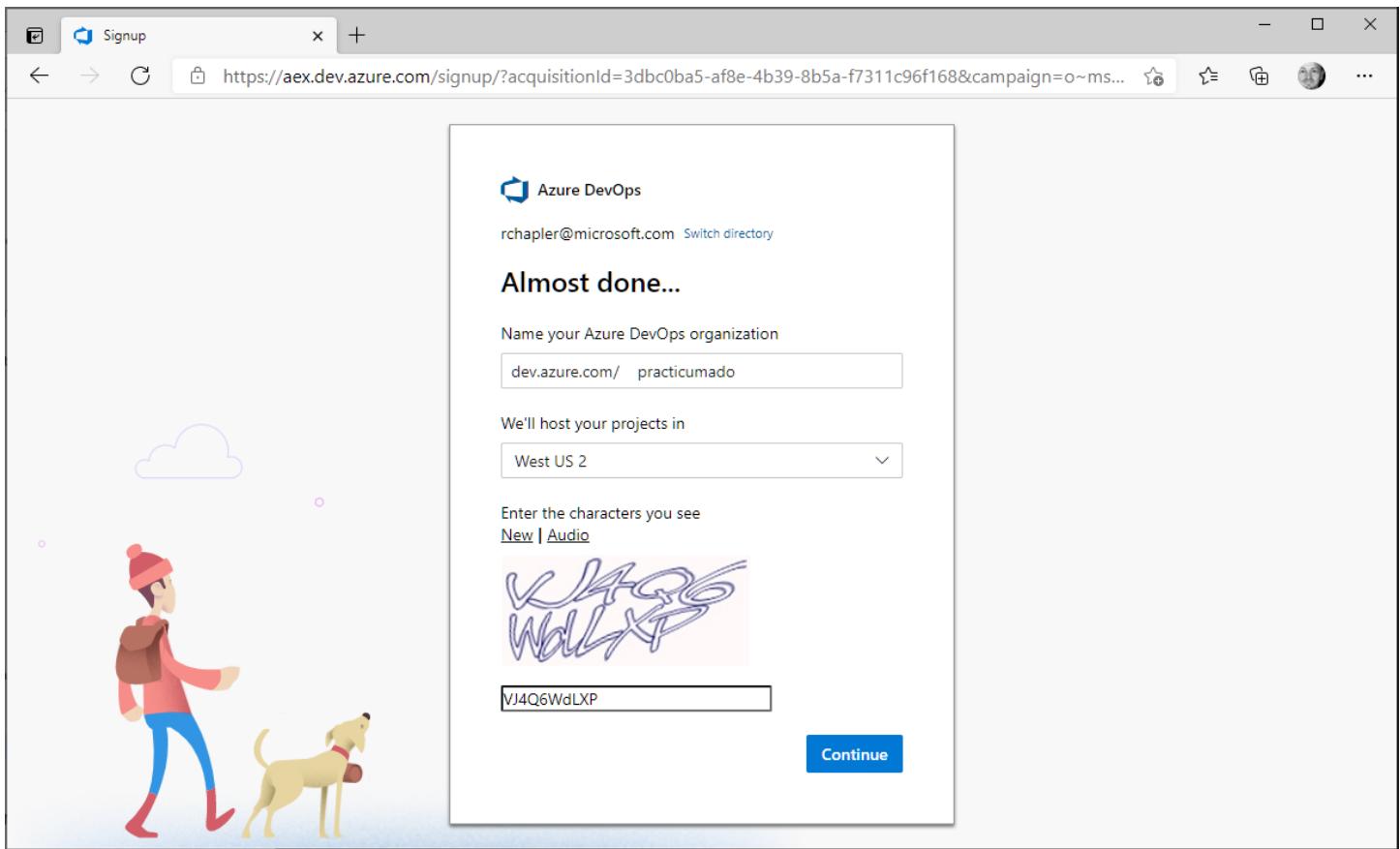
- Header:** Microsoft Azure DevOps Services | Microsoft
- Navigation:** Overview, Solutions, Products, Documentation, Pricing, Training, Marketplace, Partners, Support, Blog, More
- Breadcrumbs:** Home / Services / Azure DevOps
- Call-to-action buttons:** Start free, Start free with GitHub
- Text:** Plan smarter, collaborate better, and ship faster with a set of modern dev services.
- Links:** Already have an account?, Sign in to Azure DevOps >

Click the “Start free with GitHub” button.

On the resulting Azure DevOps page, click the “New organization” link on the navigation.



Click the **Continue** button.



On the “Almost done...” page, enter values for the following form items:

**Name your Azure DevOps organization** Enter a meaningful name aligned with standards

**We will host your projects in** Match resource group selection

Click the **Continue** button.

## Create Project

The screenshot shows the Azure DevOps 'Create a project to get started' page. On the left, there's a sidebar with organization names (practicumado, [REDACTED], [REDACTED], [REDACTED]), a 'New organization' link, and sections for 'What's new' (Sprint 184 release notes) and 'Organization settings'. The main area has a form for creating a project:

- Project name \***: practicumadop
- Description**: (empty)
- Visibility**:
  - Public**: Anyone on the internet can view the project. Certain features like TFVC are not supported.
  - Enterprise**: Members of your enterprise can view the project.
  - Private**: Only people you give access to will be able to view this project. This option is selected and highlighted with a blue border.
- Advanced** (button)
- + Create project** (button)

On the “Create a project to get started” page, enter values for the following form items:

<b>Project Name</b>	Enter a meaningful name aligned with standards
<b>Visibility</b>	Select the value that best aligns with your requirements

Click the “+ Create project” button.

When processing is complete, you will be navigated to a screen like the one snipped below.

The screenshot shows the Azure DevOps interface for the 'practicumadop' project. The left sidebar contains links for Overview, Summary (which is selected), Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main content area features a green header with the project name 'practicumadop'. Below the header is a cartoon illustration of a person working at a desk with a laptop, accompanied by a dog. The text 'Welcome to the project!' is displayed, followed by a question 'What service would you like to start with?'. Below this are buttons for Boards, Repos, Pipelines, Test Plans, and Artifacts, with 'Artifacts' being the active tab. At the bottom, there is a link to 'or manage your services'.

## Key Vault

### Step 1: Create Resource

On the “Create Key Vault” page, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your resource group
Key Vault Name	Enter a meaningful name aligned with standards {e.g., <b>practicumkv</b> }
Region	Match resource group selection
Pricing Tier	Confirm default selection, “Standard”

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “Review + create” button, review configuration, and then click the **Create** button.

## Log Analytics

The screenshot shows the Microsoft Azure portal in preview mode. The URL is https://ms.portal.azure.com/#create/Microsoft.LogAnalyticsOMS. The page title is "Create Log Analytics workspace". The top navigation bar includes "Microsoft Azure (Preview)", a search bar, and a user profile for rchabler@microsoft.com.

The main content area has a dark background. At the top, there are tabs: "Basics" (which is selected), "Pricing tier", "Tags", and "Review + Create". A blue info box states: "A Log Analytics workspace is the basic management unit of Azure Monitor Logs. There are specific considerations you should take when creating a new Log Analytics workspace. [Learn more](#)".

**Project details:** This section asks to select a subscription and resource group. The subscription is set to "rchabler" and the resource group is "practicumrg". There is also a "Create new" button.

**Instance details:** This section asks to enter a name and region. The name is "practicumlaw" and the region is "West US 2".

At the bottom, there are buttons for "Review + Create" (highlighted in blue), "« Previous", and "Next : Pricing tier >".

On the “Create Log Analytics workspace” page, enter values for the following items:

**Subscription** Select your subscription

**Resource Group** Select your resource group

**Name** Enter a meaningful name aligned with standards

**Region** Match resource group selection

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “Review + create” button, review configuration, and then click the **Create** button.

## Metrics Advisor

The screenshot shows the 'Create Metrics Advisor' page in the Microsoft Azure portal. At the top, the URL is https://ms.portal.azure.com/#create/Microsoft.CognitiveServicesMetricsAdvisor. The page title is 'Create Metrics Advisor'. The navigation path is: Dashboard > Resource groups > practicumrg > Create a resource > Marketplace > Metrics Advisor > Create Metrics Advisor.

**Basics**    Virtual network    Tags    Review + create

Embed AI-powered monitoring features to stay one step ahead of incidents no machine-learning expertise required. Metrics Advisor monitors the performance of your organization's growth engines, from sales revenue to manufacturing operations. It helps you quickly identify and fix problems through a powerful combination of monitoring in near-real time, adapting models to your scenario, offering granular analysis with diagnostics, and alerting. [Learn more](#)

**Project details**

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

Subscription \* ⓘ rchapler

Resource group \* ⓘ practicumrg

Create new

**Instance details**

Region \* ⓘ West US 2

Name \* ⓘ practicuma

Pricing tier \* ⓘ S0

[View full pricing details](#)

**Storage**

Bring your own storage \* ⓘ  Yes  
 No

The deployment could take up to 60 minutes to complete, although it normally finishes in less than 10 minutes.

I confirm I have read and understood the  notice below.

[Service Agreement & Terms](#)

[Review + create](#)    < Previous    Next : Virtual network >

On the “Create Metrics Advisor” page, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your resource group
Region	Match resource group selection
Name	Enter a meaningful name aligned with standards
Pricing Tier	Select S0
Bring Your Own Storage	Confirm default selection, “No”
I confirm...	Check the box

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “**Review + create**” button, review configuration, and then click the **Create** button.

## Add Permissions to Data Explorer

To complete this objective, you must instantiate the following resources:

- Data Explorer

Navigate to your instance of Data Explorer.

The screenshot shows the Microsoft Azure portal interface. On the left, the navigation menu includes 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Permissions' (which is selected), 'Query', 'Settings', 'Scale up', 'Scale out', and 'Configurations'. The main content area is titled 'practicumdec | Permissions' and shows a list of principals under the 'AllDatabaseAdmin' role. One principal, 'practicumdf', is listed as an 'App'. A modal dialog box titled 'New Principals' is open, showing a search result for 'practicumma' with its GUID. The 'Selected' section of the dialog contains the same item, with a 'Remove' button and a 'Select' button at the bottom.

Click **Permissions** in the navigation.

Click “**+ Add**” on the resulting page.

Select **AllDatabaseViewer** from the resulting dropdown.

Search for and then click to **Select** your instance of Metrics Advisor.

## Postman

Browse to [Download Postman | Try Postman for Free](#)

Select, download, and install the version of the Postman app appropriate for your system.

## Power Apps

If you don't have a working instance of Power Apps, you can get started at <https://Power Apps.microsoft.com/en-us/>

If you are already set up, click “**Sign In**”.

## Purview

To complete this objective, you must instantiate the following resources:

- Key Vault
- Resource Group

Create a Purview account to develop a data governance solution in just a few clicks. A storage account and eventhub will be created in a managed resource group in your subscription for catalog ingestion scenarios. [Learn more](#)

**Project details**

Subscription \* rchapler

Resource group \* practicumrg [Create new](#)

**Instance details**

Purview account name \* practicump

Location \* West US 2

**Managed resources**

Managed resource group name \* practicumpmrg

Storage account name *Name will be auto-generated during account creation.*

Event Hubs namespace name *Name will be auto-generated during account creation.*

[Review + Create](#) [Previous](#) [Next: Networking >](#)

On the “Create Purview account” page, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your resource group
Purview Account Name	Enter a meaningful name aligned with standards
Location	Match resource group selection
Managed Resource Group...	Self-explanatory

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “Review + Create” button, review configuration, and then click the **Create** button.

## Add Access Policy to Key Vault

Navigate to the Key Vault.

practicumkv - Microsoft Azure

Microsoft Azure (Preview)

Dashboard > practicumrg > practicumkv

Key vault

Search (Ctrl+ /)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Settings

- Keys
- Secrets
- Certificates
- Access policies

Resource group (change) practicumrg

Location West US 2

Subscription (change) rchapler

Subscription ID

Vault URI https://practicumkv.vault.azure.net/

Sku (Pricing tier) Standard

Directory ID

Directory Name Microsoft

Soft-delete Enabled

Purge protection Disabled

Tags (change)  
Click here to add tags

Click “Access Policies” in the **Settings** group of the navigation.

practicumkv - Microsoft Azure

Microsoft Azure (Preview)

Dashboard > practicumrg > practicumkv

Key vault

Search (Ctrl+ /)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Settings

- Keys
- Secrets
- Certificates
- Access policies

Enable Access to:

- Azure Virtual Machines for deployment ⓘ
- Azure Resource Manager for template deployment ⓘ
- Azure Disk Encryption for volume encryption ⓘ

Permission model

- Vault access policy
- Azure role-based access control

+ Add Access Policy

Current Access Policies

Name	Email	Key Permissions	Secret Permissions	Certificate Permissions	Action
APPLICATION					

Click the “+ Add Access Policy” link.

Principal

Select a principal

practicump  
d0206ad4-fa5d-4923-982a-4cf48bcf3559  
Selected

Selected items

practicump  
d0206ad4-fa5d-4923-982a-4cf48bcf3559

Add

Configure from template (optional)

Key permissions

Secret permissions

Certificate permissions

Select principal \*

Authorized application ⓘ

Get

List

Set

Delete

Recover

Backup

Restore

Select

On the “Add access policy” page, enter values for the following items:

**Secret Permissions** Select **Get** and **List** in the dropdown

**Select Principal** Search for, and then **Select** the managed identity for your instance of Purview

Click the **Add** button.

practicumkv | Access policies

Key vault

Search (Ctrl+ /)

Save Discard Refresh

Please click the 'Save' button to commit your changes.

Enable Access to:

Azure Virtual Machines for deployment ⓘ

Azure Resource Manager for template deployment ⓘ

Azure Disk Encryption for volume encryption ⓘ

Permission model

Vault access policy (selected)

Azure role-based access control

+ Add Access Policy

Current Access Policies

Name	Email	Key Permissions	Secret Permissions	Certificate Permissions	Action
APPLICATION	practicum	0 selected	2 selected	0 selected	<a href="#">Delete</a>

Click the **Save** button.

## Connect Key Vault

Navigate to Purview and click on the “**Open Purview Studio**” button.

Select the **Management** icon on the navigation.

Click **Credentials** in the “**Security and access**” group of the resulting menu.

Click the “**Manage Key Vault connections**” button on the resulting page.

Click the “**+ New**” button the resulting popout.

The screenshot shows the 'New Key Vault' dialog box within the Azure Purview Studio interface. The left sidebar shows 'General' selected under 'Credentials'. The main area displays a form with the following fields:

- Name \***: practicumkv
- Description**: (empty)
- Azure subscription**: rchapler [dropdown]
- Key Vault name \***: practicumkv
- Note**: You must grant the Purview managed identity access to your Azure Key Vault. [See more](#)

At the bottom are 'Create', 'Back', and 'Cancel' buttons.

On the “**New Key Vault**” popout, enter values for the following items:

<b>Name</b>	Enter a meaningful name aligned with standards
<b>Subscription</b>	Select your subscription
<b>Key Vault Name</b>	Select your instance of Key Vault

Click the **Create** button.

On the resulting “**Confirm granting access**” popup, review and then click the **Confirm** button.

Confirm the addition of your instance of Key Vault on the “**Manage Key Vault connections**” popout.

Click the **Close** button.

## Add New Credential

On the **Credentials** page, click the “**+ New**” button.

The screenshot shows the Microsoft Azure Purview Studio interface. In the top navigation bar, there are tabs for 'practicump - Microsoft Azure' and 'Azure Purview Studio'. The URL in the address bar is [https://ms.web.purview.azure.com/resource/practicump/main/datasource/management/credentials?feature.tenant=\[REDACTED\]](https://ms.web.purview.azure.com/resource/practicump/main/datasource/management/credentials?feature.tenant=[REDACTED]). The main content area has a blue header bar with icons for notifications, search, and user profile.

The left sidebar contains a navigation menu with the following items:

- General
- Overview
- Metrics
- Lineage connections
- Data Factory
- Data Share
- Security and access
- Credentials (selected)

The central panel is titled 'Credentials' and shows a message: 'Credentials are much like connection strings, which define how to connect to a data source.' Below this, it says 'Showing 0 items' and has a 'Name' filter field.

The right panel is titled 'New credential' and contains the following fields:

- Name \*: practicumsds
- Description: (empty text area)
- Authentication method \*: SQL authentication
- User name \*: rchabler
- Password \*: (empty text area)
- Key Vault connection \*: practicumkv
- Secret name \*: practicumsds-adminpassword
- Secret version: Use the latest version if left blank

At the bottom of the right panel are two buttons: 'Create' (highlighted in blue) and 'Cancel'.

On the resulting “**New Credential**” popout, enter values for the following items:

<b>Name</b>	Enter a meaningful name aligned with standards
<b>Authentication</b>	Select “SQL authentication”
<b>User Name</b>	Enter the “server admin login” value used during creation of the SQL Database Server
<b>Key Vault Connection</b>	Select your key vault
<b>Secret Name</b>	Select your secret

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the **Create** button.

## Resource Group

The screenshot shows the Microsoft Azure (Preview) portal with the URL <https://ms.portal.azure.com/#create/Microsoft.ResourceGroup>. The page title is "Create a resource group". The "Basics" tab is selected. In the "Project details" section, the "Subscription" dropdown is set to "rchapler" and the "Resource group" dropdown is set to "practicumrg". In the "Resource details" section, the "Region" dropdown is set to "(US) West US 2". At the bottom, there are navigation buttons: "Review + create", "< Previous", and "Next : Tags >".

On the “Create a Resource Group” page, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your resource group
Region	Select a region appropriate for your situation; take into consideration that: <ul style="list-style-type: none"><li>Some regions {e.g., West US and East US} see higher demand than others</li><li>Creation of resources in the same region offers best performance and lowest cost</li></ul>

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “Review + create” button, review configuration, and then click the **Create** button.

# SQL

## Step 1: Create Resource

The screenshot shows the Microsoft Azure portal's "Select SQL deployment option" page. At the top, there are four main options:

- SQL databases**: Best for modern cloud applications. Hyperscale and serverless options are available. Resource type dropdown: Single database. Buttons: Create, Hide details.
- SQL managed instances**: Best for most migrations to the cloud. Lift-and-shift ready. Resource type dropdown: Single instance. Buttons: Create, Show details.
- SQL virtual machines**: Best for migrations and applications requiring OS-level access. Lift-and-shift ready. Image dropdown. Buttons: Create, Show details.

Below these are three summary sections:

- Single database**: Single databases are a great fit for modern, cloud-born applications that need a fully managed database with predictable performance. **Featured capabilities:** Hyperscale storage (up to 100TB), Serverless compute, Easy management.
- Elastic pool**: Elastic pools provide a cost-effective solution for managing the performance of multiple databases with variable usage patterns. **Featured capabilities:** Resource sharing for cost optimization, Simplified performance management.
- Database server**: Database servers are used to manage groups of single databases and elastic pools. **Featured capabilities:** Access management, Backup management, Business continuity management.

On the “Select SQL deployment option” page, select the deployment option appropriate to your use case.

For practicum examples, we will typically select “Single database” in the “SQL databases” > “Resource type” dropdown.

Click the **Create** button.

On the “Create SQL Database” page, **Basics** tab, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your resource group
Database Name	Enter a meaningful name aligned with standards {e.g., <b>practicumsd</b> }
Server	Click the “Create new” link, enter values in the resulting “New server” popout and then click the <b>OK</b> button
Want to use SQL elastic pool?	Confirm default value, <b>No</b>
Compute + Storage	Click the “Configure database” link

On the resulting **Configure** page, enter values for the following items:

Service Tier	Confirm default value, “General Purpose...”
Compute Tier	Click the <b>Serverless</b> radio button
Max vCores	Confirm minimum value
Min vCores	Confirm minimum value
Enable Auto-Pause	Confirm default value, checked
Days   Hours   Minutes	Confirm default values, 0   1   0

<b>Data Max Size (GB)</b>	Set minimum appropriate value
<b>“...database zone redundant”</b>	Confirm default value, <b>No</b>

Review the “**Cost Summary**” and then click the **Apply** button.

Back on and further down on the “**Create SQL Database**” page, **Basics** tab, enter values for the following items:

<b>Backup Storage Redundancy</b>	Confirm default value, “Locally-redundant backup storage”
----------------------------------	---

If you want to add a sample database, navigate to the “**Additional settings**” tab and enter values for the following items:

<b>Use Existing Data</b>	Select the “Sample” option
--------------------------	----------------------------

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “**Review + create**” button, review configuration, and then click the **Create** button.

## Step 2: Prepare Key Vault

Create Secret for Admin Password

Navigate to your instance of Key Vault and then click **Secrets** in the **Settings** group of the navigation.

Click the “**+ Generate/Import**” button.

On the “**Create a secret**” page, enter values for the following items:

<b>Name</b>	Enter a meaningful name aligned with standards {e.g., “practicumsds-adminpassword”}
<b>Value</b>	Enter the value used when creating the SQL Database Server

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the **Create** button.

## Step 3: Configure Firewall

Navigate to your Azure SQL Server and click the “**Show firewall settings**” link on the **Overview** page.

Set “**Allow Azure services and resources to access this server**” to “**Yes**”.

Click the **Save** button.

## Storage Account

The screenshot shows the 'Create storage account' wizard in the Microsoft Azure portal. The current step is 'Basics'. The page title is 'Create storage account - Microsoft Azure'. The URL is https://ms.portal.azure.com/#create/Microsoft.StorageAccount-ARM. The top navigation bar shows 'Microsoft Azure (Preview)' and a search bar. Below the navigation is a breadcrumb trail: Dashboard > New > Marketplace > Storage account >. The main content area is titled 'Create storage account' with a back button.

**Basics** Networking Data protection Advanced Tags Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below.  
[Learn more about Azure storage accounts](#)

**Project details**

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

Subscription \* rchabler

Resource group \* practicumrg  
Create new

**Instance details**

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

Storage account name \* practicumsa

Location \* (US) West US 2

Performance Standard

Account kind StorageV2 (general purpose v2)

Replication Read-access geo-redundant storage (RA-GRS)

Review + create < Previous Next : Networking >

On the “Create Storage Account” page, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your resource group
Storage Account Name	Enter a meaningful name aligned with standards
Location	Match resource group selection
Performance	Confirm default radio button selection, “Standard”
Account Kind	Confirm default selection, “StorageV2 (general purpose v2)”
Replication	Confirm default selection, “Read-access geo-redundant storage (RA-GRS)”

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “Review + create” button, review configuration, and then click the **Create** button.

# Synapse

## Step 1: Instantiate Prerequisites

To complete this objective, you must instantiate the following resources:

- Key Vault
- Resource Group
- SQL

## Step 2: Create Resource

On the “Create Synapse workspace” page, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your resource group
Managed Resource Group	This additional Resource Group will hold ancillary resources created specifically for Synapse Because you are likely to have more than one managed resource group, consider using a consistent naming pattern {e.g., “practicummrg-s” for Synapse and “practicummrg-x” for X}
Workspace Name	Enter a meaningful name aligned with standards {e.g., <b>practicums</b> }
Region	Match resource group selection
Select Data Lake Storage Gen2	Confirm default, “From subscription”
Account Name	Select or create a data lake {e.g., <b>practicumdl</b> }
File System Name	Select or create a data lake file system {e.g., <b>practicumdlfs</b> }
Assign myself...	Check to assign necessary permissions

Click the “Next: Security >” button and enter values for “SQL Server Admin Login” and “SQL Password”.

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “Review + create” button, review configuration, and then click the **Create** button.

## Step 3: Prepare Key Vault

Create Secret for Admin Password

Navigate to your instance of Key Vault.

Click **Secrets** in the **Settings** group of the navigation and then click the “+ Generate/Import” button.

On the “Create a secret” page, enter values for the following items:

Name	Enter a meaningful name aligned with standards {e.g., “ <b>practicums-adminpassword</b> ”}
Value	Enter the value used when creating the Synapse Workspace

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the **Create** button.

Add Access Policy to Key Vault

Navigate to your instance of Key Vault and then click “Access Policies” in the **Settings** group of the navigation.

Click the “+ Add Access Policy” link and on the resulting “Add access policy” page, enter values for the following items:

Secret Permissions	Select <b>Get</b> and <b>List</b> in the dropdown
Select Principal	Search for, and then <b>Select</b> the managed identity for your Synapse Workspace

Click the **Add** button and then click the **Save** button.

## Step 4: Add Permissions to Data Explorer

To complete this objective, you must instantiate the following resources:

- Data Explorer

Navigate to your instance of Data Explorer.

Click **Permissions** in the navigation.

Click “+ Add” on the resulting page.

Select **AllDatabasesViewer** for read permissions and **AllDatabasesAdmin** for write, etc. permissions from the resulting dropdown.

Search for and then click to **Select** your instance of Synapse.

## Step 5: Create Databases

Data Explorer

Create Data Explorer Pool

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “Open Synapse Studio” rectangle.

Click the **Manage** icon in the navigation.

Select “**Data Explorer pools...**” in the “**Analytics pools**” grouping of the resulting navigation and then click the “**+ New**” button.

On the resulting “Create Data Explorer pool” popout, **Basics** tab, enter values for the following items:

<b>Data Explorer Pool Name</b>	Enter a meaningful name aligned with standards {e.g., <b>practicumdep</b> }
<b>Workload</b>	Select an appropriate option {e.g., “ <b>Compute optimized</b> ” as we will not have a large volume of data}
<b>Size</b>	Select an appropriate option {e.g., “ <b>Extra Small</b> ” as we will not demonstrate expensive computation}

Navigate to the “Create Data Explorer pool” popout, “**Additional settings**” tab, and then enter values for the following items:

<b>Autoscale</b>	Select an appropriate option {e.g., <b>Disabled</b> as we will not require Autoscale for this demonstration}
<b>Number of Instances</b>	Select an appropriate option (I usually choose the least expensive option for demonstrations)
<b>Estimated Price</b>	Confirm the estimated cost per hour that Azure generates based on your selected options

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “**Review + create**” button, review configuration, and then click the **Create** button.

Create Data Explorer Database

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “Open Synapse Studio” rectangle.

Click the **Data** icon in the navigation, the + button in the header and then “**Data Explorer database...**” in the resulting dropdown.

On the resulting “Create Data Explorer database...” popout, enter values for the following items:

<b>Pool Name</b>	Enter a meaningful name aligned with standards {e.g., <b>practicumdep</b> }
<b>Name</b>	Enter a meaningful name aligned with standards {e.g., <b>practicumded</b> }
<b>Default Retention Period...</b>	Confirm default, <b>365</b>
<b>Default Cache Period...</b>	Confirm default, <b>31</b>

Click the **Create** button.

## Sample 1, Product Table

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Click the **Data** icon in the navigation, expand “**Data Explorer Databases...**”, and your Data Explorer Pool.

Right click on your Data Explorer Database {e.g., **practicumded**} and select “New KQL script” from the resulting dropdown.

Paste and then Run the following KQL query to create a new table named Product:

```
.create table Product ( ProductId:int, ProductNumber: string, Name:string, ListPrice: decimal, ModifiedDate:datetime )
```

## Sample 2, StormEvents Data

Follow the instructions in the “Quickstart: Ingest sample data into Azure Data Explorer” article (<https://docs.microsoft.com/en-us/azure/data-explorer/ingest-sample-data>) to populate sample data that we can surface in Power Apps.

The screenshot shows the Azure Data Explorer interface. The left sidebar has three tabs: Data, Query (selected), and Dashboards (Preview). The main area shows a query editor with the following KQL code:

```
1 .create table StormEvents (StartTime: datetime, EndTime: datetime, EpisodeId: int, EventId: int, State: string, EventType: string)
2
3 .ingest into table StormEvents 'https://kustosamplefiles.blob.core.windows.net/sampledata/StormEvents.kusto'
4
5 StormEvents | sort by StartTime desc | take 10
```

Below the code, a table titled "Table 1" displays the results of the query. The columns are: StartTime, EndTime, EpisodeId, EventId, State, and EventType. The data is as follows:

StartTime	EndTime	EpisodeId	EventId	State	EventType
2007-12-31 23:53:00.0000	2007-12-31 23:53:00.0000	12,037	65,839	CALIFORNIA	High Wind
2007-12-31 23:53:00.0000	2007-12-31 23:53:00.0000	12,037	65,838	CALIFORNIA	High Wind
2007-12-31 22:30:00.0000	2007-12-31 23:59:00.0000	12,950	71,590	MICHIGAN	Winter Storm
2007-12-31 22:30:00.0000	2007-12-31 23:59:00.0000	12,950	71,588	MICHIGAN	Winter Storm
2007-12-31 22:30:00.0000	2007-12-31 23:59:00.0000	12,950	71,589	MICHIGAN	Winter Storm

Familiarize yourself with the resulting data for use in later sections.

## SQL

### Dedicated SQL Pool

Navigate to your instance of Synapse.

Click the “**+ New dedicated SQL pool**” button.

Enter a meaningful name in “**Dedicated SQL pool name**” and choose a pricing tier.

Click the “**Review + create**” button, review configuration, and then click the **Create** button.

## Serverless SQL Database

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Click the **Data** icon in the navigation.

Click the **Workspace** tab. Click the **+** button and “**SQL database**” in the resulting dropdown.

On the “**Create SQL database**” popout, enter values for the following items:

Select SQL Pool Type	Confirm default, <b>Serverless</b>
Database	Enter a meaningful name aligned with standards

Click the **Create** button.

## Step 6: Create Linked Services

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Click the **Manage** icon in the navigation.

Select “**Linked services**” in the “**External connections**” grouping of the resulting navigation and then click the “**+ New**” button.

Search for and then select the resource for which you are creating a linked service; examples:

- Azure Key Vault
- Azure Data Explorer (Kusto)
- Azure Data Lake Storage Gen2
- Azure SQL Database

Click the **Continue** button.

On the “**New linked service...**” popout, enter values for the following common items:

Name	Enter a meaningful name aligned with standards
Connect via...	Confirm default selection, “ <b>AutoResolveIntegrationRuntime</b> ”

And enter values for resource specific items...

### Data Explorer

Authentication Method	Select “ <b>Managed Identity</b> ”
Selection Method	Select “ <b>Enter manually</b> ”
Endpoint	Enter the URI for your Data Explorer Pool {e.g., <a href="https://practicumdep.practicums.kusto.azuresynapse.net">https://practicumdep.practicums.kusto.azuresynapse.net</a> }
Database	Select your Data Explorer Database

### Data Lake

Selection Method	Confirm default selection, “ <b>From Azure subscription</b> ”
Subscription	Select your subscription
Authentication Method	Confirm default selection, “ <b>Account key</b> ”
Storage Account Name	Select your Data Lake

### Key Vault

Selection Method	Confirm default selection, “ <b>From Azure subscription</b> ”
Subscription	Select your subscription
Azure Key Vault Name	Select your instance of Key Vault

## SQL

Selection Method	Confirm default selection, “From Azure subscription”
Subscription	Select your subscription
Server Name	Select your Azure SQL Server {e.g., <b>practicumsds</b> }
Database Name	Select your Azure SQL Database {e.g., <b>practicumsd</b> }
Authentication Type	Select “ <b>SQL authentication</b> ”
User Name	Enter the “ <b>server admin login</b> ” value used during instantiation of the SQL Database Server
AKV Linked Service	Select the name of the Linked Service created for the Key Vault
Secret Name	Enter the Secret Name used to capture the Azure SQL Server administrator password

Click “**Test connection**” to confirm successful connection and then click the **Create** (or **Commit**) button.

## Step 7: Create Apache Spark Pool

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Click the **Manage** icon and then “**Apache Spark pools**” in the navigation.

Click the “**New Apache Spark pool**” button.

On the “**New Apache Spark pool**” popout, enter values for the following items:

Apache Spark Pool Name	Enter a meaningful name aligned with standards
Isolated Compute	Confirm default, <b>Disabled</b>
Node Size Family	Confirm default, “ <b>Memory Optimized</b> ”
Node Size	Select “ <b>Small (4 vCores / 32 GB)</b> ”
Autoscale	Select <b>Disabled</b>
Number of Nodes	Slide to lowest possible value (to minimize demonstration cost)
Estimated Price	Review final “Est. cost per hour” and view pricing details, as desired

Click the “**Review + create**” button, validate settings, and then click the **Create** button.

# Virtual Machines

## Created using Portal

**Use Case:** Customer XYZ shared the following requirements:

- Mimic hard-to-replicate environments {e.g., on-prem SQL Server 2008 R2...}
- Create and configure using Portal

The screenshot shows the 'Create a virtual machine' page in the Microsoft Azure (Preview) portal. The 'Basics' tab is selected. The configuration includes:

- Subscription:** rchapler
- Resource group:** practicumrg
- Virtual machine name:** practicuumvm
- Region:** (US) West US 2
- Availability options:** No infrastructure redundancy required
- Security type:** Standard
- Image:** SQL Server 2008 R2 SP3 Standard on Windows Server 2008 R2 - Gen1
- Size:** Standard\_DS11\_v2 - 2 vcpus, 14 GiB memory (\$0.00/month)

At the bottom, there are buttons for 'Review + create' and 'Next : Disks >'.

On the “Create virtual machine” page, **Basics** tab, enter values for the following items:

<b>Subscription</b>	Select your subscription
<b>Resource Group</b>	Select your resource group
<b>Virtual Machine Name</b>	Enter a meaningful name aligned with standards
<b>Region</b>	Match resource group selection
<b>Availability Options</b>	Confirm default, “No infrastructure redundancy required”
<b>Security Type</b>	Confirm default, <b>Standard</b>
<b>Image</b>	Search for and select the “SQL Server 2008 R2...” image

Azure Spot Instance	Confirm default, unchecked
Size	Select the SKU appropriate to your use case requirement (I typically start with the cheapest option)

The screenshot shows the 'Create a virtual machine' wizard on the Microsoft Azure portal. The 'Inbound port rules' section is highlighted. A warning message in a yellow box states: '⚠️ This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.' Below this, under 'Select inbound ports', 'RDP (3389)' is selected. At the bottom of the page, there are navigation buttons: 'Review + create', '< Previous', and 'Next : Disks >'.

Further down on the “Create virtual machine” page, Basics tab, enter values for the following items:

Username	Self-explanatory
Password	
Confirm Password	
Public Inbound Ports	Confirm defaults, “Allow Selected Ports” and “RDP (3389)”
Select Inbound Ports	

The screenshot shows the Microsoft Azure 'Create a virtual machine' interface. The top navigation bar includes 'Create a virtual machine - Micros...', a search bar 'Search resources, services, and docs (G+)', and a user profile 'rchapler@microsoft.com MICROSOFT (MICROSOFT.ONM...)'.

The main page title is 'Create a virtual machine'. Below it, the 'SQL Server settings' tab is selected, indicated by a blue underline. The page displays several configuration sections:

- Security & Networking**: SQL connectivity is set to 'Public (Internet)' and port 1433 is specified.
- SQL Authentication**: SQL Authentication is enabled, with a login name of 'rchapler' and a password obscured.
- Azure Key Vault integration**: Integration is disabled.
- Storage configuration**: A note states: 'The default storage configuration for SQL virtual machines has changed, now including OLTP optimization and separate drives for data and log storage.'
- Storage**: Status is 'Not available'.
- SQL Server License**: A note says 'Save up to 43% with licenses you already own. Already have a SQL Server license? Learn more'. The 'No' radio button is selected.
- Automated patching**: Status is 'Enabled' at 'Sunday at 2:00'.
- Automated backup**: Status is 'Not available'.
- R Services(Advanced Analytics)**: SQL Server Machine Learning Services (In-Database) is disabled.

At the bottom, there are navigation buttons: 'Review + create' (highlighted in blue), '< Previous', and 'Next : Tags >'.

On the “Create virtual machine” page, “SQL Server settings” tab, enter values for the following items:

SQL Connectivity	Select “Public (Internet)”
------------------	----------------------------

Port	Confirm default, <b>1433</b>
SQL Authentication	Click to <b>Enable</b> and then confirm default (which will match your previously entered Administrator values)
Login Name	
Password	

No additional changes are required but be sure to confirm settings on the remaining tabs.

Click the “**Review + create**” button, validate settings, and then click the **Create** button.

## Configure SQL Server

**Connect** to the new virtual machine (using RDP) to complete image-specific configuration.

## SQL Server Browser

Enable SQL Server Browser to allow connection to the SQL Server.

Open **Services**, search for and double-click to open “**SQL Server Browser**”.

On the **General** tab, change “**Startup Type**” to **Automatic**, and then click the **Apply** button.

Click the **Start** button.

## Configure IE ESC

Disable Internet Explorer Enhanced Security Configuration to allow downloading to the demonstration VM.

Open **Server Manager** and click the “**Configure IE ESC**” link in the “**Server Summary**” > “**Security Information**” interface grouping.

On the “**Internet Explorer Enhanced Security Configuration**” popup, click **Off** under **Administrators** and **Users**, and then click the **OK** button.

## Sample Database

Browse to [AdventureWorks sample databases - SQL Server | Microsoft Docs](#).

Download the appropriate version of the AdventureWorks sample database {e.g., “AdventureWorks2008R2-oltp.bak”}.

Open **SQL Server Management Studio**, right-click on **Databases**, and click “**Restore Database**” in the resulting dropdown.

On the “**Restore Database**” popup, enter AdventureWorks in the “**To database**” text box, “**Destination for restore**” interface grouping.

On the “**Restore Database**” popup, click the “**From device**” radio button in the “**Source for restore**” interface grouping.

Click the **Ellipses** button and in the resulting “**Specify Backup**” popup, click the **Add** button.

Browse to the downloaded BAK file; you may have to copy the BAK from the Downloads folder to another location like c:\temp to make it accessible.

Back on the “**Restore Database**” popup, check the box next to new item in “**Select the backup sets to restore**” and then click the **OK** button.

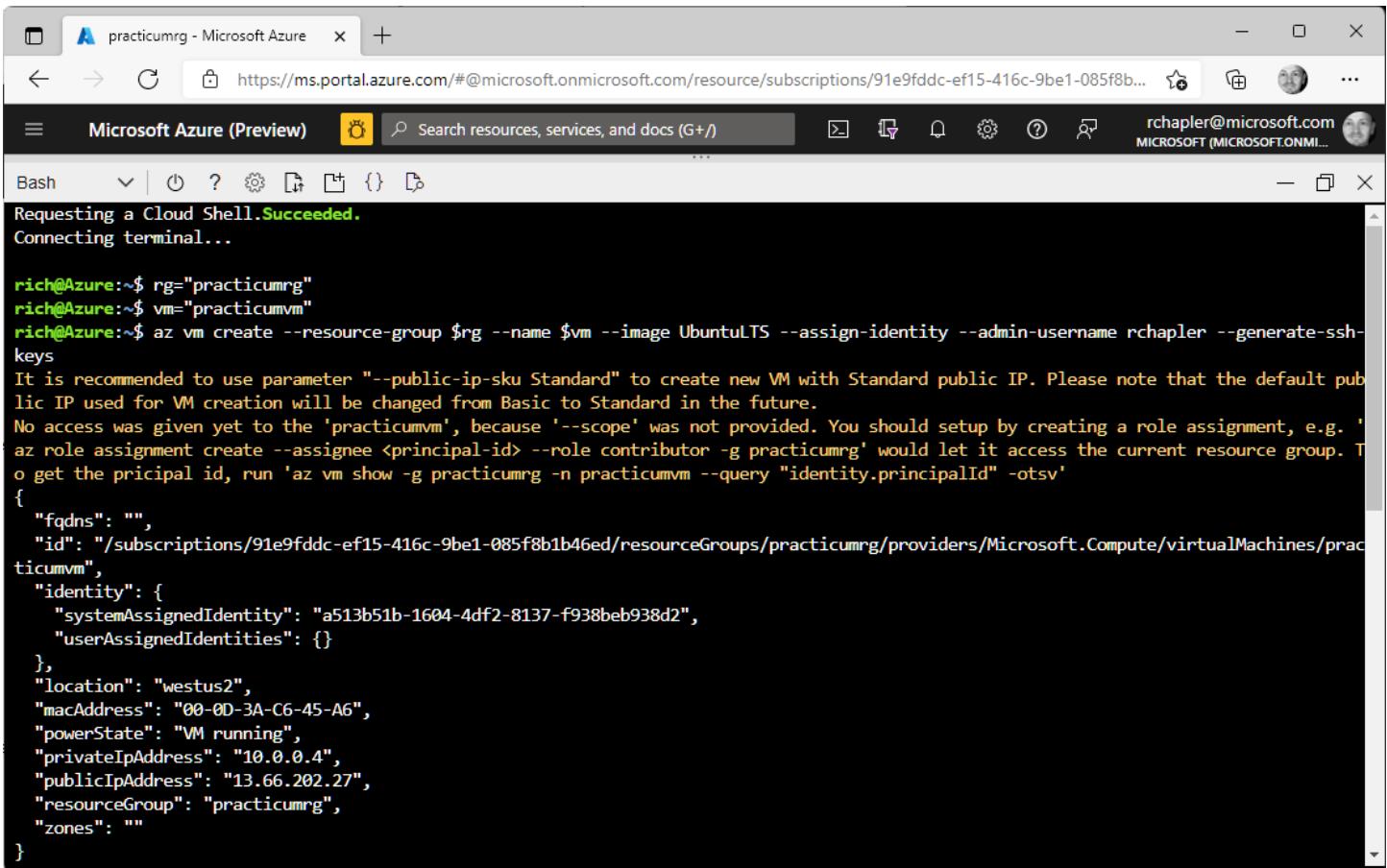
Upon successful completion, you should see a popup that says “**The restore of database ‘AdventureWorks’ completed successfully.**

## Created using Cloud Shell

**Use Case:** Customer XYZ shared the following requirements:

- Mimic hard-to-replicate environments {e.g., on-prem Ubuntu (Linux)}
- Create and configure using Cloud Shell

Navigate to Cloud Shell.



The screenshot shows a Microsoft Azure Cloud Shell window. The title bar says "practicumrg - Microsoft Azure". The URL in the address bar is "https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/91e9fddc-ef15-416c-9be1-085f8b...". The user is "rchapler@microsoft.com" from "MICROSOFT (MICROSOFT.ONM...)".

The terminal session starts with "Requesting a Cloud Shell. Succeeded." followed by "Connecting terminal...". The command entered is "az vm create --resource-group \$rg --name \$vm --image UbuntuLTS --assign-identity --admin-username rchapler --generate-ssh-keys". The output shows a warning about using the "Standard" public IP SKU and a note about role assignments. The JSON response for the VM creation is displayed, including fields like "fqdns", "id", "identity", "location", "macAddress", "powerState", "privateIpAddress", "publicIpAddress", "resourceGroup", and "zones".

```
Requesting a Cloud Shell. Succeeded.
Connecting terminal...

rich@Azure:~$ rg="practicumrg"
rich@Azure:~$ vm="practicumvm"
rich@Azure:~$ az vm create --resource-group $rg --name $vm --image UbuntuLTS --assign-identity --admin-username rchapler --generate-ssh-keys
It is recommended to use parameter "--public-ip-sku Standard" to create new VM with Standard public IP. Please note that the default public IP used for VM creation will be changed from Basic to Standard in the future.
No access was given yet to the 'practicumvm', because '--scope' was not provided. You should setup by creating a role assignment, e.g. 'az role assignment create --assignee <principal-id> --role contributor -g practicumrg' would let it access the current resource group. To get the principal id, run 'az vm show -g practicumrg -n practicumvm --query "identity.principalId" -otsv'
{
  "fqdns": "",
  "id": "/subscriptions/91e9fddc-ef15-416c-9be1-085f8b1b46ed/resourceGroups/practicumrg/providers/Microsoft.Compute/virtualMachines/practicumvm",
  "identity": {
    "systemAssignedIdentity": "a513b51b-1604-4df2-8137-f938beb938d2",
    "userAssignedIdentities": {}
  },
  "location": "westus2",
  "macAddress": "00-0D-3A-C6-45-A6",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "13.66.202.27",
  "resourceGroup": "practicumrg",
  "zones": ""
}
```

Update and execute the following command to set variable **rg** {i.e., the name of your resource group}:

```
rg="practicumrg"
```

Update and execute the following command to set variable **vm** {i.e., the desired name for the new virtual machine}:

```
vm="practicumvm"
```

Execute the following command to create your virtual machine:

```
az vm create --resource-group $rg --name $vm --image UbuntuLTS --assign-identity --admin-username rchapler --generate-ssh-keys
```

Successful execution will create the following resources:

- Virtual Machine
- Disk
- Network Security Group
- Public IP Address
- Network Interface
- Virtual Network

Update and execute the following command to connect to the server using SSH:

```
ssh -i ~/.ssh/id_rsa rchapler@[public ip address]
```

The screenshot shows a Microsoft Azure (Preview) terminal window titled "practicumrg - Microsoft Azure". The URL in the address bar is <https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/91e9fddc-ef15-416c-9be1-085f8b...>. The terminal is running a Bash session. The user has run the command `ssh -i ~/.ssh/id_rsa rchapler@13.66.202.27`. The terminal displays the following output:

```
rich@Azure:~$ ssh -i ~/.ssh/id_rsa rchapler@13.66.202.27
The authenticity of host '13.66.202.27 (13.66.202.27)' can't be established.
ECDSA key fingerprint is SHA256:2psYqDjJe2y74/P36tyrkln3Ha+WgNSdWgb9+UuA2gHs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '13.66.202.27' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1063-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Fri Nov 19 15:18:45 UTC 2021

System load:  0.15           Processes:      112
Usage of /:   4.6% of 28.90GB  Users logged in:    0
Memory usage: 5%            IP address for eth0: 10.0.0.4
Swap usage:   0%

0 updates can be applied immediately.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

rchapler@practicumvm:~$
```

You will know you are successful if your prompt has changed to your equivalent of `rchapler@practicumvm`.