Triggering Synapse Pipelines in DevOps

Created by	Ⓓ Devansh Gupta
Created time	@October 6, 2023 11:50 AM

STEP 1: Create a Service Connection
STEP 2: **Create an Azure DevOps Repo**
STEP 3: **Create an Azure DevOps Pipeline**
STEP 4: Add logging statements in your Synapse Pipeline (optional)
STEP 5: **Run the Azure DevOps Pipeline**
STEP 6: Monitoring the Pipeline Run

Creating a manual process to trigger a Synapse pipeline from Azure DevOps without creating a DevOps pipeline configuration from scratch.

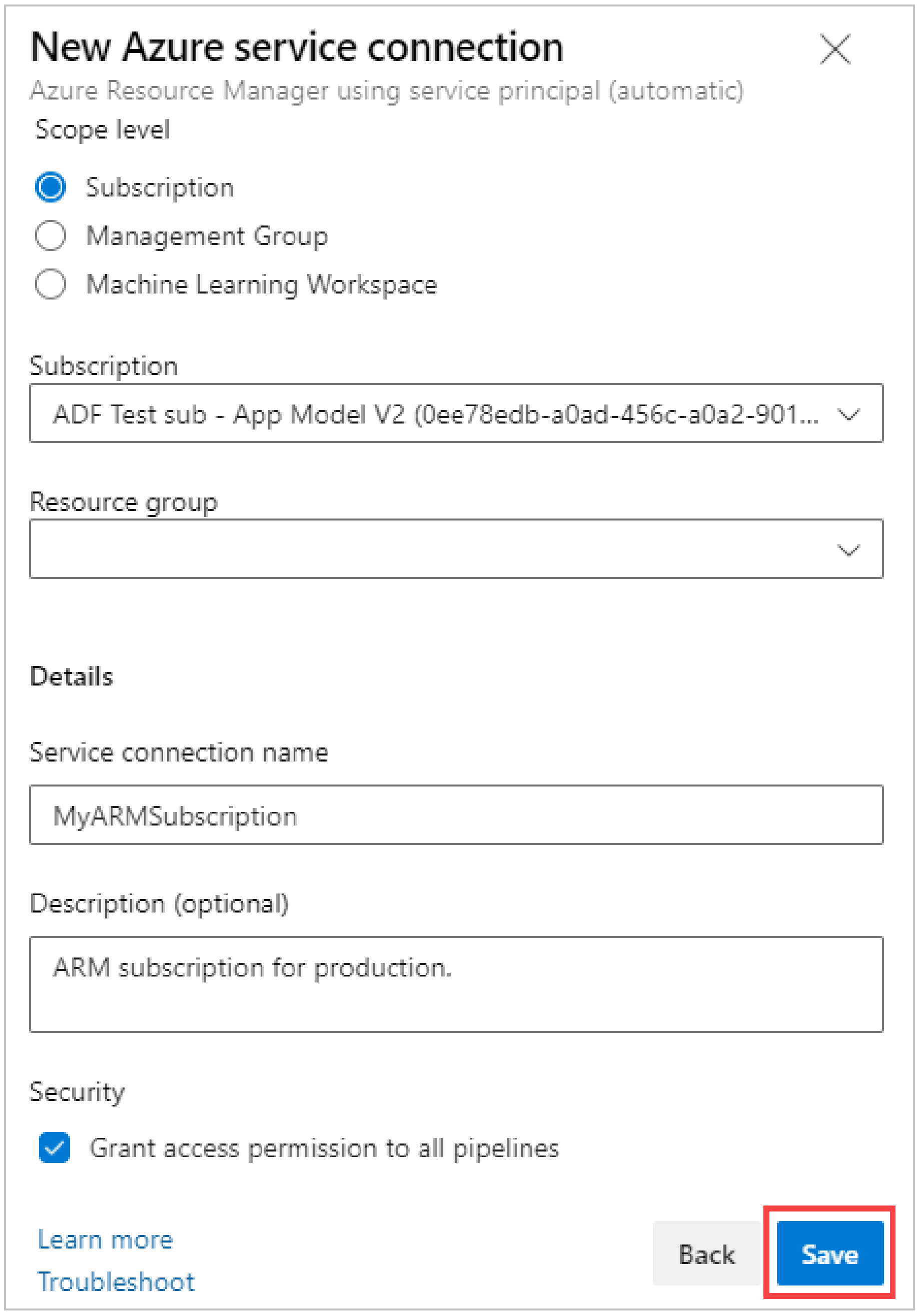
By running the trigger script within an Azure DevOps pipeline, you can automate the process of starting your Azure Synapse pipeline based on predefined triggers or events, and you can also benefit from the monitoring capabilities provided by Azure DevOps for your pipeline run.

STEP 1: Create a Service Connection

Complete the following steps to create a service connection for Azure Pipelines.

1. Sign in to your organization (`https://dev.azure.com/{yourorganization}`) and select your project.
2. Select **Project settings** > **Service connections**.
3. Select **+ New service connection**, select the type of service connection that you need,

and then select **Next**.
4. Choose an authentication method, and then select **Next**.
5. Enter the parameters for the service connection. The list of parameters differs for each type of service connection. For more information, see the list of service connection types and associated parameters.
6. Select **Save** to create the connection.

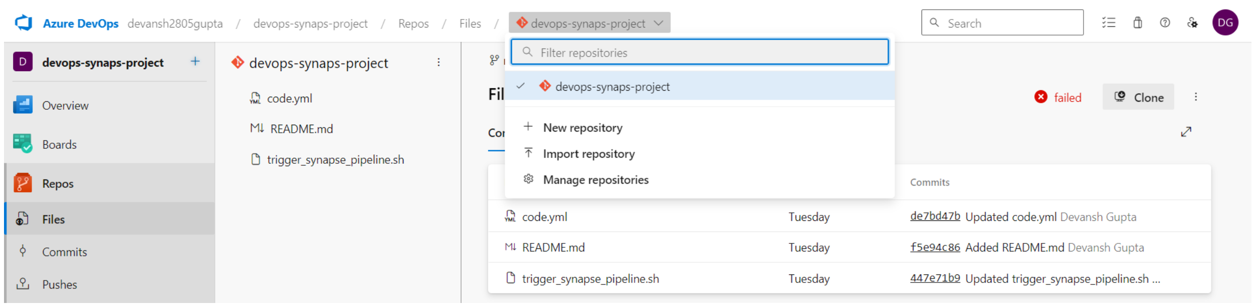

New Azure service connection
Azure Resource Manager using service principal (automatic)
Scope level
(●) Subscription
( ) Management Group
( ) Machine Learning Workspace
Subscription: ADF Test sub - App Model V2 (0ee78edb-a0ad-456c-a0a2-901...
Resource group:
Details
Service connection name: MyARMSubscription
Description (optional): ARM subscription for production.
Security
[✓] Grant access permission to all pipelines

Not selecting a resource group ensures that this Service Connection is connected to the whole subscription and can access any service that uses the same credentials.

STEP 2: Create an Azure DevOps Repo

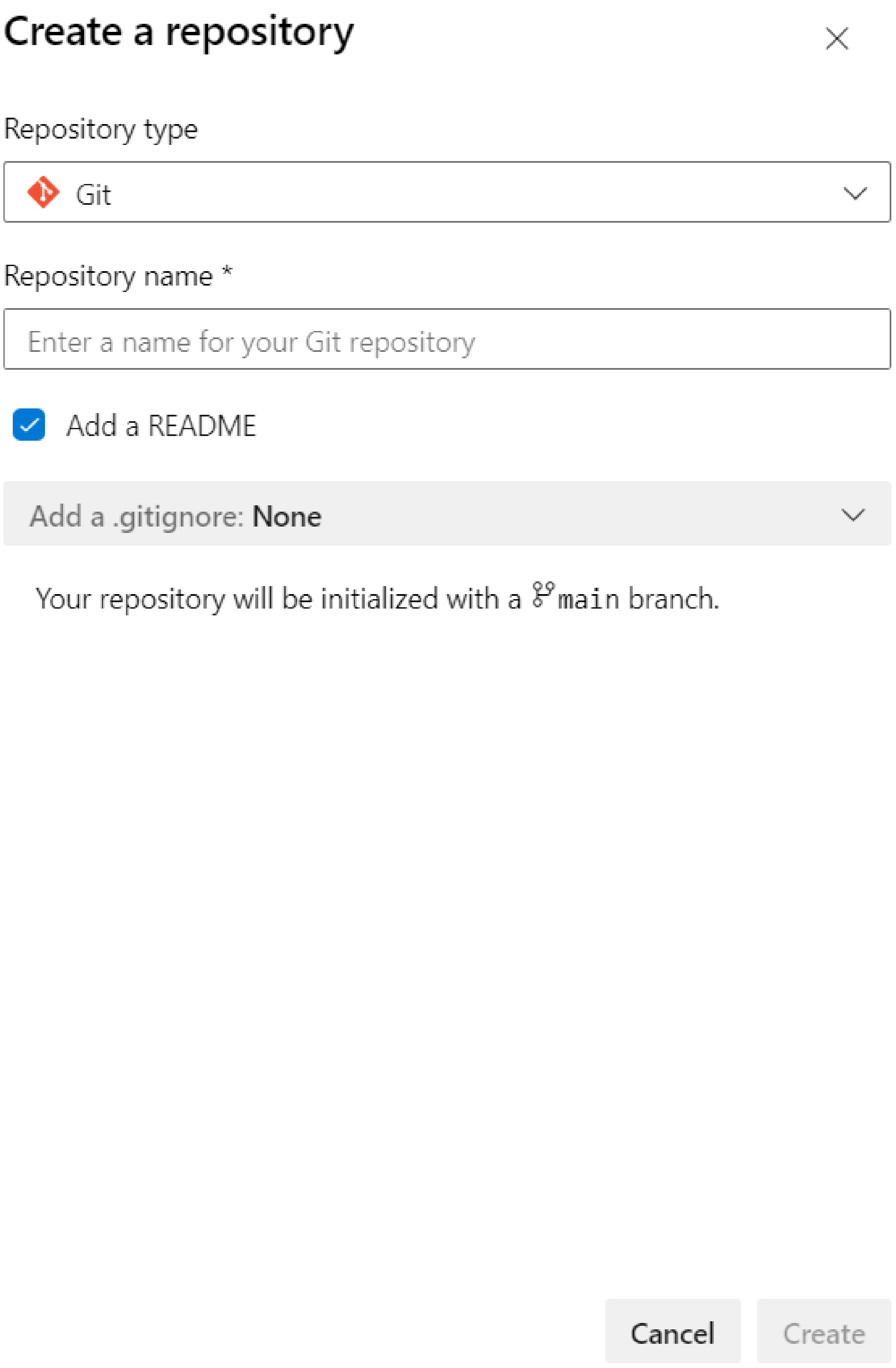
Create an Azure DevOps Repository which stores the YAML code which triggers the Azure Synapse Pipeline when run.

Go to then Repos Section → Files →



code.yml	Tuesday	de7bd47b Updated code.yml Devansh Gupta
README.md	Tuesday	f5e94c86 Added README.md Devansh Gupta
trigger_synapse_pipeline.sh	Tuesday	447e71b9 Updated trigger_synapse_pipeline.sh ...

Here select the +New Repository option and enter the Type and Name of the Repo.



Create a repository

Repository type

Git

Repository name *

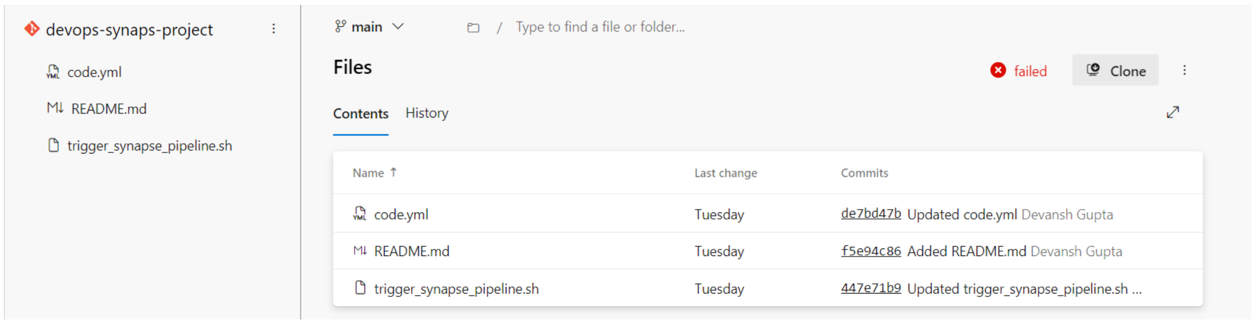
Enter a name for your Git repository

☑ Add a README

Add a .gitignore: None

Your repository will be initialized with a `main` branch.

Entering the details, you will create a New Repository in which you can add the following files:


devops-synaps-project

- code.yml
- README.md
- trigger_synapse_pipeline.sh

main / Type to find a file or folder...

Files

failed · Clone

Contents · History

Name ↑	Last change	Commits
code.yml	Tuesday	de7bd47b Updated code.yml Devansh Gupta
README.md	Tuesday	f5e94c86 Added README.md Devansh Gupta
trigger_synapse_pipeline.sh	Tuesday	447e71b9 Updated trigger_synapse_pipeline.sh ...

The following code can be used to create the code.yml file -

```
trigger:
  branches:
    exclude:
      - '*'

pool:
  vmImage: 'windows-latest'

steps:
- task: AzureCLI@2
  inputs:
    azureSubscription: 'YourAzureServiceConnection'
    scriptType: 'bash'
    scriptLocation: 'code.yml'
    scriptPath: 'scripts/trigger_synapse_pipeline.sh'
```

Replace `'YourAzureServiceConnection'`, `'scriptPath'`, and `'scripts/trigger_synapse_pipeline.sh'` with the appropriate values for your Azure service connection, script location, and script path.

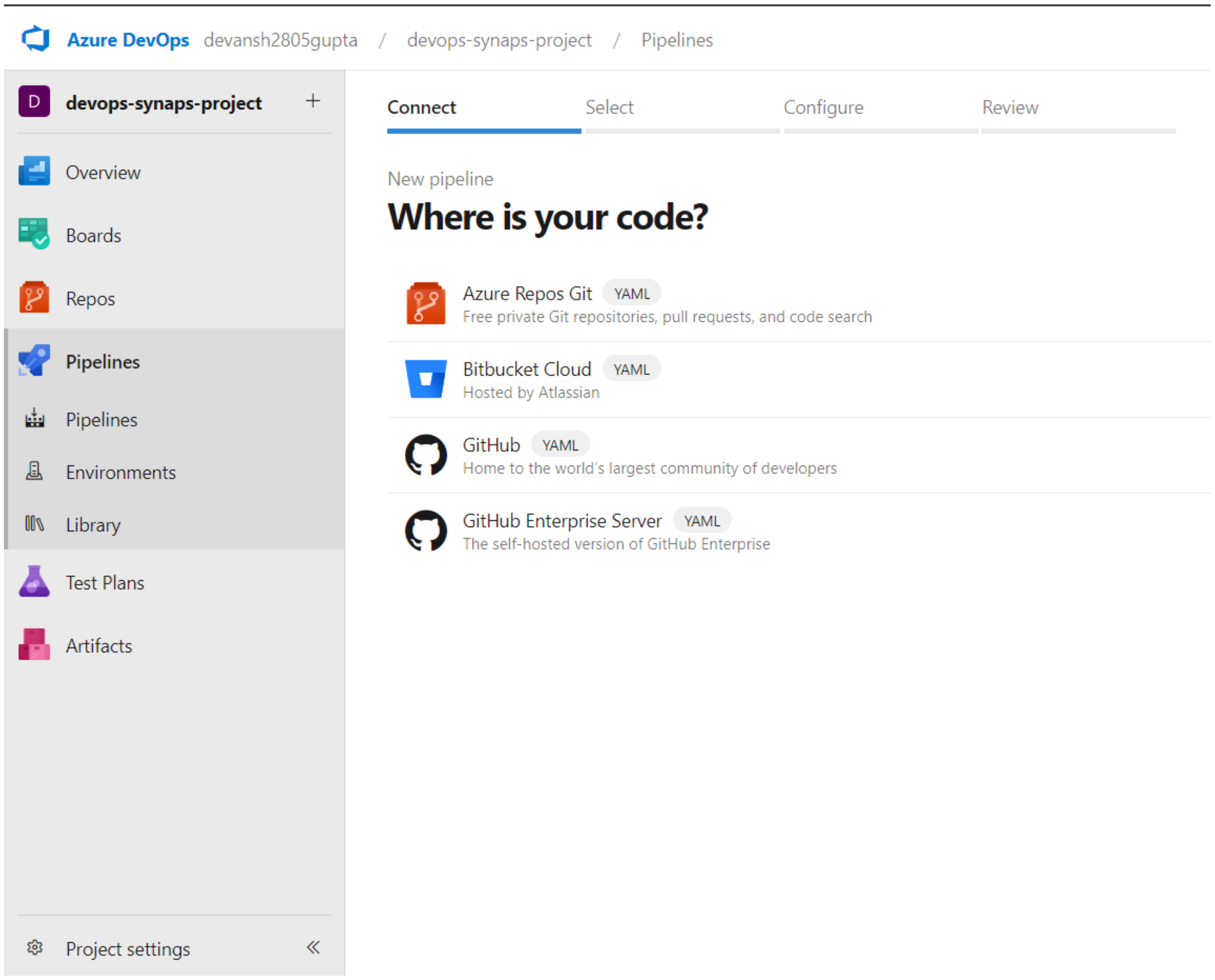
This file, creates the pipeline and when this is run, it triggers the run of Azure Synapse Pipeline which is already created by calling the Bash file **trigger_synapse_pipeline.sh:**

```
# Variables
synapseWorkspaceName="syn-project-new"     #Replace with your synapse workspace name
pipelineName="Pipeline1"                  #Replace with your synapse workspace pipeline name
  resourceGroup="rg-project"              #Replace with your resource group name
devOpsPAT="n7hznj7355x4h3utyyexcigrhmgwbvjvx5wcxjbugfjfxcd6llea" # Replace with your DevOps PAT

# Log in to Azure using a service connection or service principal
az login --service-principal -u $servicePrincipalId -p $servicePrincipalKey --tenant $tenantId

# Set the active Azure Synapse workspace
az synapse set --workspace-name $synapseWorkspaceName --resource-group $resourceGroup

# Trigger the Azure Synapse pipeline
az synapse pipeline create-run --workspace-name $synapseWorkspaceName --name $pipelineName

# Log out from Azure CLI (optional, depending on your setup)
az logout

# Return a message to the Azure DevOps pipeline
echo "Azure Synapse pipeline execution triggered successfully."
```

This completes the making of your Repository.

STEP 3: Create an Azure DevOps Pipeline

Go to the Pipelines section → +New Pipeline. You will get a screen like this:

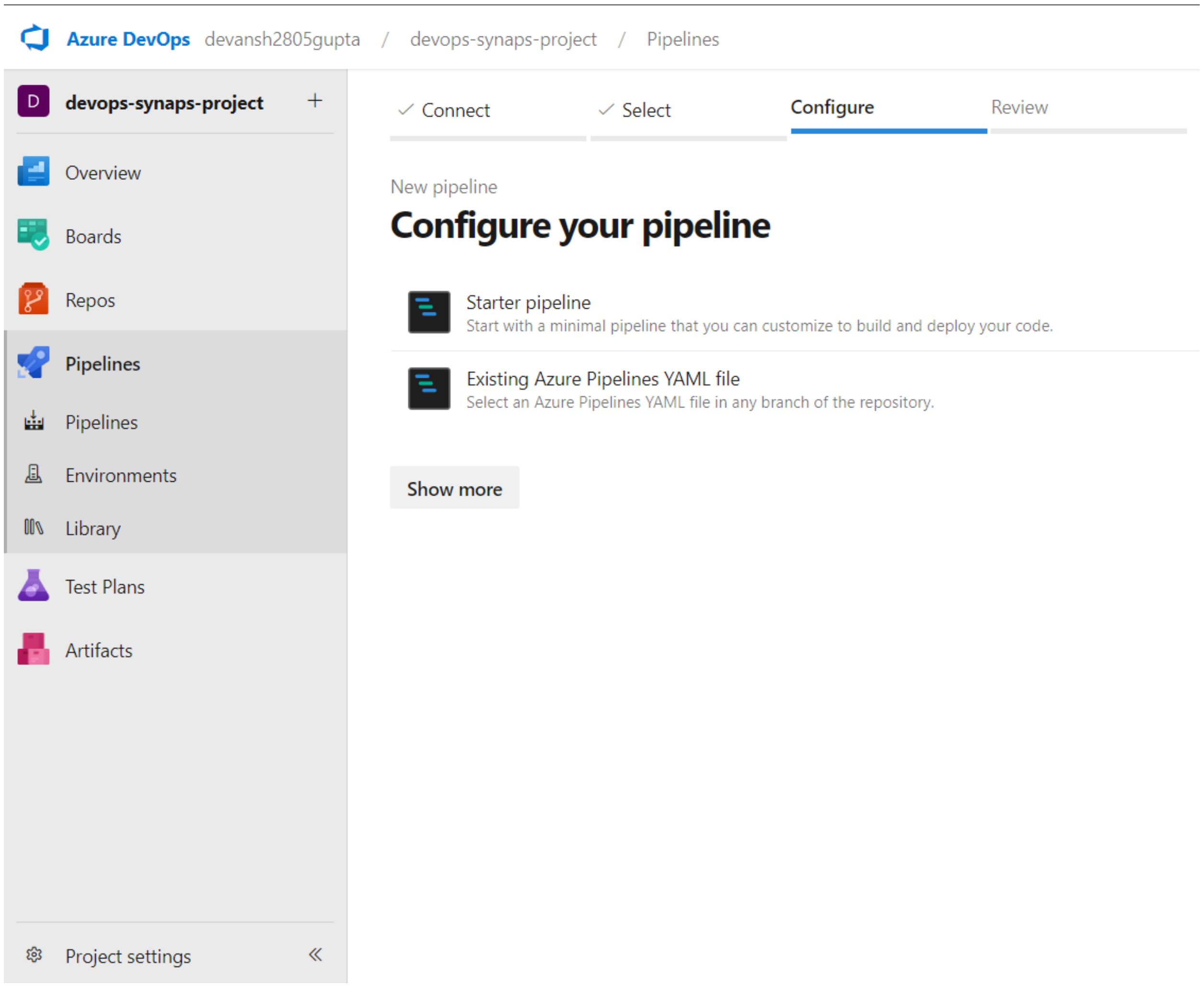


New pipeline

Where is your code?

- Azure Repos Git YAML — Free private Git repositories, pull requests, and code search
- Bitbucket Cloud YAML — Hosted by Atlassian
- GitHub YAML — Home to the world's largest community of developers
- GitHub Enterprise Server YAML — The self-hosted version of GitHub Enterprise

Select Azure Repos Git as your Repository is already created in DevOps using the Git format. You will get an option to select your repository.


Azure DevOps devansh2805gupta / devops-synaps-project / Pipelines

✓ Connect ✓ Select Configure Review

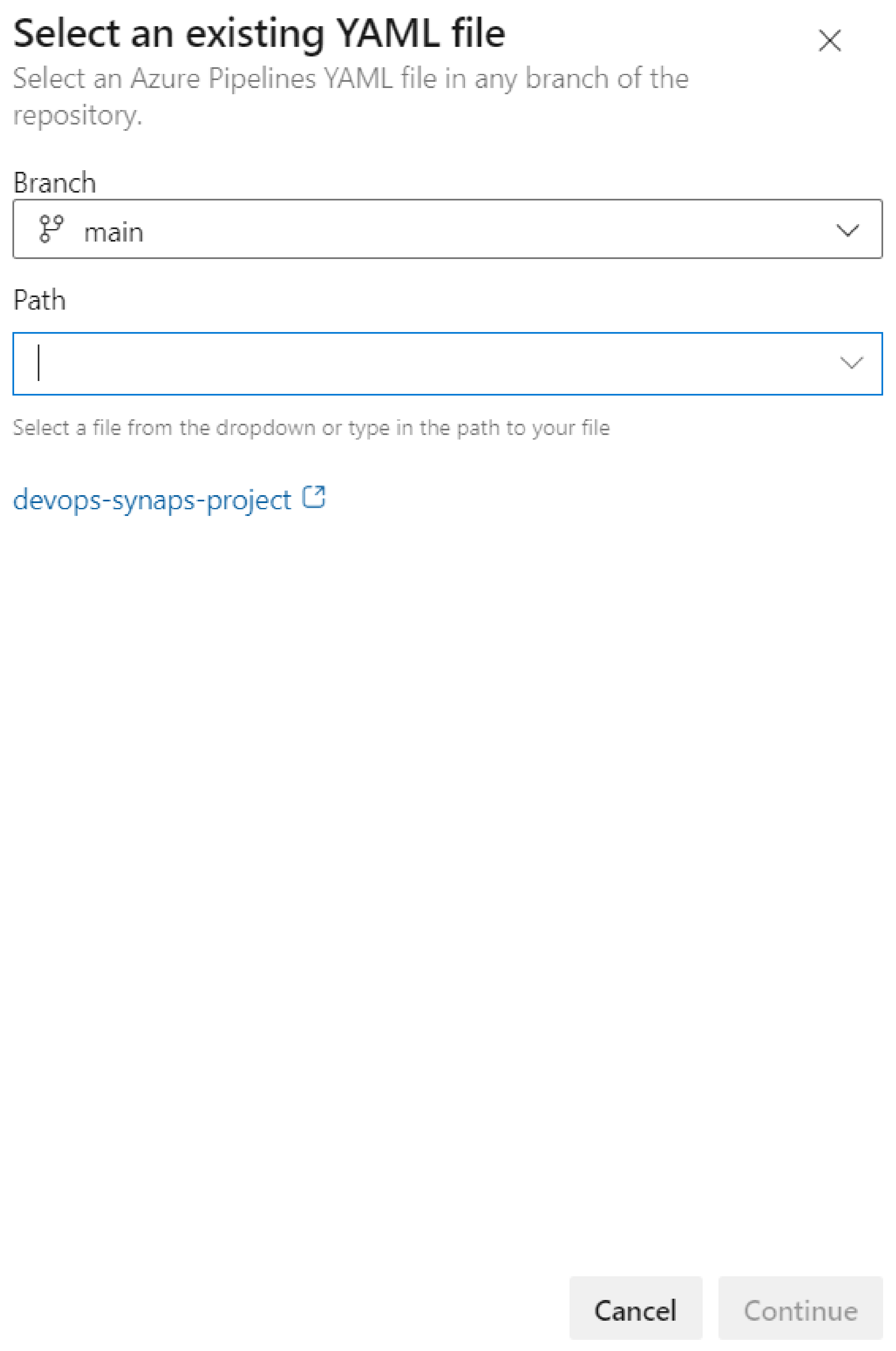
New pipeline

Configure your pipeline

Starter pipeline
Start with a minimal pipeline that you can customize to build and deploy your code.

Existing Azure Pipelines YAML file
Select an Azure Pipelines YAML file in any branch of the repository.

Here, select on Existing Azure Pipelines YAML file and this asks for the path of your YAML file.



Select an existing YAML file

Select an Azure Pipelines YAML file in any branch of the repository.

Branch: main

Path:

Select a file from the dropdown or type in the path to your file

devops-synaps-project

Select your YAML file code.yml by clicking on the dropdown. This creates your new pipeline.

STEP 4: Add logging statements in your Synapse Pipeline (optional)

To place logging statements in your Azure Synapse notebook, you can use the built-in capabilities of the notebook's chosen language, which is typically Python or PySpark. You can use print statements or logging libraries to capture and log information during notebook execution. Here's an example using Python:

```
import logging
# Configure the logger
logging.basicConfig(filename='notebook.log', level=logging.INFO)
# Log messages
logging.info("Starting data processing...")
data = [1, 2, 3, 4, 5]
logging.info(f"Data length: {len(data)}")
try:
  result = 10 / 0
except Exception as e:
  logging.error(f"Error: {str(e)}")
  logging.info("Data processing completed.")
```

In this example, the `logging` module is used to configure a logger and log messages to a file named `notebook.log`. You can customize the logging level and output format as needed.

We can place these logging statements (if your Synapse pipeline has notebook activity) to record the Run and these logs can be then stored in a Storage Container and be reviewed to check for errors and incomplete runs.

STEP 5: Run the Azure DevOps Pipeline

You can manually trigger the Azure DevOps pipeline or set up triggers based on events such as code commits, pull requests, or scheduled runs, depending on your requirements.

On running the above created pipeline, it will ask for some permissions. Authorize these permissions to run the pipeline. The pipeline runs and the you see that the Synapse Pipeline has also run.

You can verify this by checking the Storage Container that contains the logs and by verifying these logs for complete runs.

STEP 6: Monitoring the Pipeline Run

The method I described in the previous response allows you to trigger your Azure Synapse pipeline from Azure DevOps using a script, but it doesn't provide a direct mechanism for monitoring the pipeline run within Azure DevOps itself. Monitoring the pipeline run typically involves viewing the execution status, logs, and results within the Azure Synapse workspace or Azure Portal rather than within Azure DevOps.

Here's how you can monitor the pipeline run when using the method described:

1. **Azure Synapse Monitoring:**
   - After you trigger the Synapse pipeline from Azure DevOps, you would monitor the execution status, progress, and any issues within the Azure Synapse workspace.
   - You can check the status of the pipeline run, view logs, and review any error messages directly in Azure Synapse Studio or the Azure Portal.
2. **Azure DevOps Pipeline Run:**
   - Within Azure DevOps, you can monitor the status of the DevOps pipeline run that triggered the Synapse pipeline.
   - Azure DevOps will provide information about whether the pipeline run (i.e., the DevOps pipeline) was successful or not.
   - However, the details of the Synapse pipeline run itself, such as the progress of individual activities and the execution logs, won't be directly visible within Azure DevOps.

In summary, while you can use Azure DevOps to trigger the Synapse pipeline and monitor the status of the Azure DevOps pipeline run, the detailed monitoring of the Synapse pipeline's execution, including the status of individual activities and execution logs, is best done within the Azure Synapse workspace or Azure Portal. These tools are designed to provide comprehensive monitoring and management capabilities for Azure Synapse pipelines.