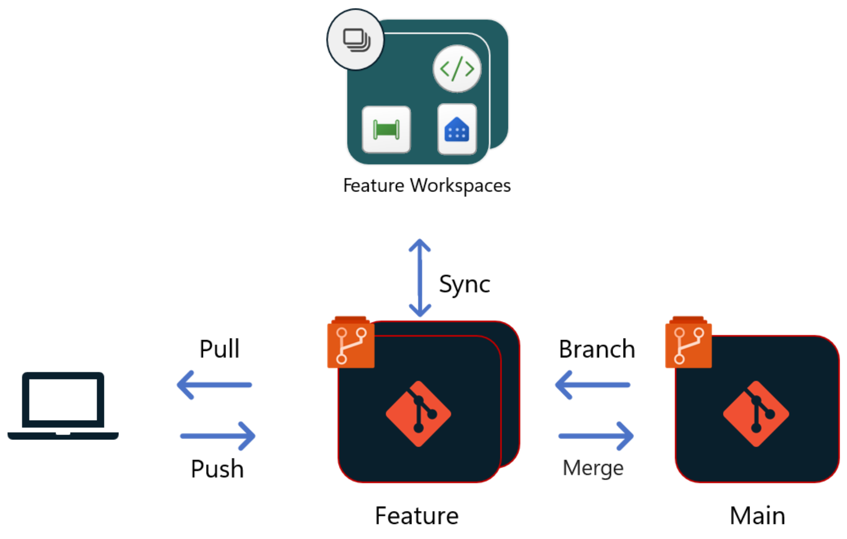
Custom “Branch Out to New Workspace” in Microsoft Fabric

Introduction

Branching out to a new workspace is a common activity in the development lifecycle within Fabric. This allows engineers to develop and test, in isolation, within a separate workspace connected to a feature branch as shown in the diagram below, and in our documentation here.



This setup can be achieved either by using the in-product capability via the user interface, or, programmatically via the REST APIs. Both of these methods may require a post process to reconfigure item level dependencies such as default lakehouses, or pipeline source/sink connection details. This document describes how to use the associated scripts in this repository to programmatically automate a custom branch out to new workspace process end-to-end using the REST APIs.

Custom Branch Out To New Workspace Scripts

Overview

Where above permissions cannot be granted to developers to utilize the built-in experience to create the new feature workspace these steps can be run using the Fabric REST APIs using an identity or set of identities or credentials which have permissions in DevOps to create new branches, and in Fabric to create new workspaces. **This document provides an overview of the supplies sample scripts to achieve this as well as setup and execution guidance.**

To automate the necessary steps, the process can be triggered from an Azure DevOps Release Pipeline by the administrator or a DevOps engineer. This is split into two steps:

1. Creation of the new workspace and new DevOps branch via release pipeline
2. Post activity tasks run via a Fabric Notebook

The release pipeline is defined by an Azure Devops yaml pipeline which runs each of these steps by invoking a Python script. The yaml pipeline passes various parameters and secrets (which are stored in an Azure Key Vault) to these scripts at run time. The secrets are used to store sensitive information, particularly for authentication purposes as there are a few options to consider from depending on which is the best fit for your organization:

1. Fabric using an access token generated via interactive user running notebook cell
2. Fabric using a dedicated Entra ID account username and password
3. Azure DevOps using service principal (client ID) and secret
4. Azure DevOps using Personal Access Token (PAT)

Certain Fabric REST APIs used in the CICD process do not currently support service principal, although this feature will be available in the future. In the meantime, there are two main interim solutions.
a.) Generate a token using notebookutils (code sample provided in Appendix A) prior to running the YAML pipeline, or,
b.) Leverage a dedicated Entra ID account without MFA, referred to as a “service account”.

If generating a token using notebookutils, the provided code will need to be run from a notebook by the user with sufficient permissions, which generates and stores a token in the linked key vault in a secret called ‘fabrictoken’. This needs to be done within one hour (expiry time) of triggering the yaml pipeline. Otherwise the keyvault secret can be used to simply store the username and password of the service account.

Note: When 'fabrictoken' is specified in the variable group this takes precedence over any username and password authentication method. Where 'azdopat' is specified this takes precedence over service principal authentication.

Understanding the scripts

This repository contains a set of files which supports a custom process to branch out to new workspace. It consists of the following files

Readme: Links to this document

AzDO folder containing yaml definition and python scripts:

> **Branch_out_workspace.yml**: definition of the release pipeline to use variable groups, accept parameters, install msal, and invoke two separate python scripts
>
> A scripts folder which contains:
>
> > **BranchOut-Feature-Workspace-Automation.py**: This python script is run in the context of an "admin" user and/or credentials (token/pat) which have permissions to create new branch, create new workspace and connect/sync to git.
> >
> > **Run_post_activity.py:** This script invokes the Fabric notebook to execute post described below.

Fabric folder containing a notebook:

> **Branch out to new workspace - Post Activity.ipynb:** This notebook executes a number of post activity steps such as:
>
> 1. Rebinding of notebooks, pipelines, semantic models and reports.
> 2. Configuring access to "source" data either via copy or shortcut (shortcut only supported in the lakehouse currently)

Configuration steps

Fabric

1. Create an Azure Key vault

Create a key vault and create a secret or secrets depending on the authentication option described above. If using the Fabric token method then the secret simply needs to be a placeholder with a dummy secret value at this stage.

2. Update the post activity notebook

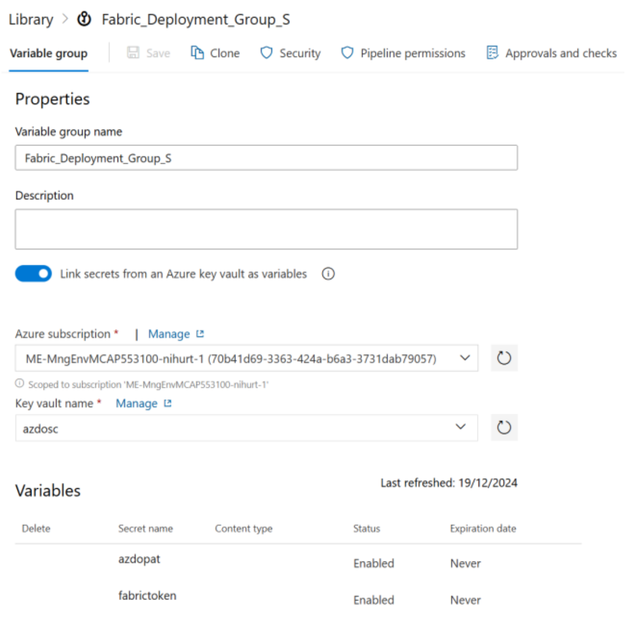
Upload the “Branch out to new workspace - Post Activity.ipynb” notebook to a workspace (My Workspace can also be used) in Fabric which the admin user has access to. Make note of notebook ID and the Workspace ID which the GUIDs that follows the “synapsenotebooks” and “groups” token respectively in the browser url.

```
https://app.fabric.microsoft..com/groups/<Workspace_ID>/synapsenotebooks/<Notebook_ID>?experience=data-engineering
```

Azure DevOps

1. Create an Azure DevOps project and initialize a main repo
2. Connect your ‘dev’ workspace to the repo and branch
3. Create a variable group under the Pipelines, Library, called Fabric_Deployment_Group_S to store sensitive information, and link the variable group to the key vault as shown below. Note the variables may be different depending on the authentication method chosen, as shown below.

When using Azure DevOps PAT and Fabric token ensure the following secrets are configured:



Properties

Variable group name: Fabric_Deployment_Group_S

Description:

Link secrets from an Azure key vault as variables

Azure subscription * | Manage: ME-MngEnvMCAP553100-nihurt-1 (70b41d69-3363-424a-b6a3-3731dab79057)

Scoped to subscription 'ME-MngEnvMCAP553100-nihurt-1'

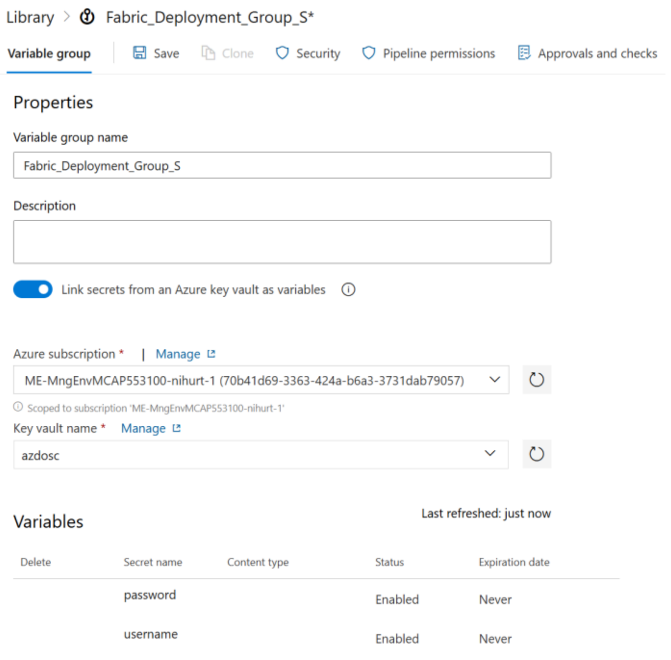
Key vault name * Manage: azdosc

Variables

Last refreshed: 19/12/2024

Delete	Secret name	Content type	Status	Expiration date
	azdopat		Enabled	Never
	fabrictoken		Enabled	Never

When using a service account ensure the following secrets are configured:


Library > Fabric_Deployment_Group_S*

Variable group | Save | Clone | Security | Pipeline permissions | Approvals and checks

Properties

Variable group name: Fabric_Deployment_Group_S

Description:

Link secrets from an Azure key vault as variables: On

Azure subscription * | Manage: ME-MngEnvMCAP553100-nihurt-1 (70b41d69-3363-424a-b6a3-3731dab79057)

Scoped to subscription 'ME-MngEnvMCAP553100-nihurt-1'

Key vault name * Manage: azdosc

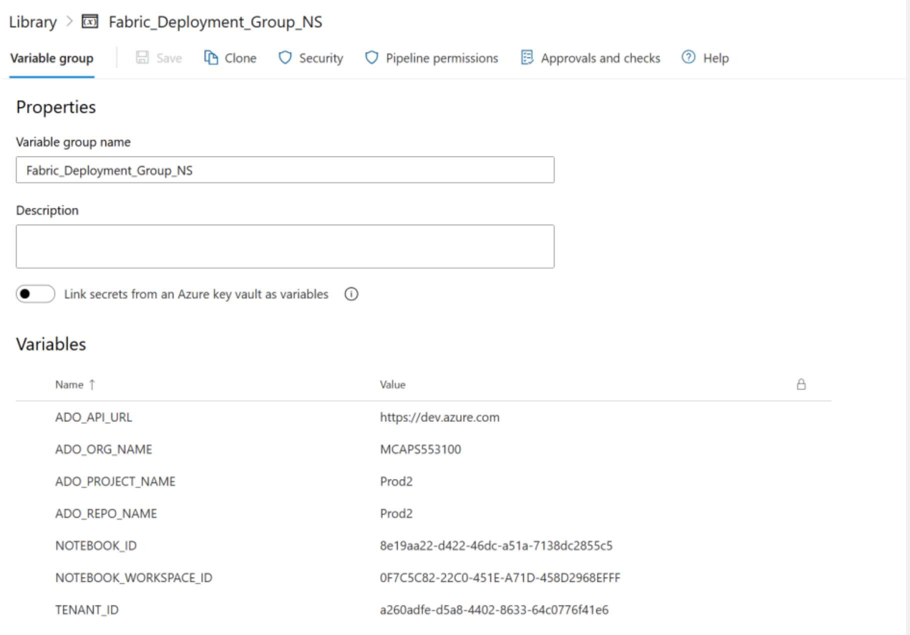
Variables

Last refreshed: just now

Delete	Secret name	Content type	Status	Expiration date
	password		Enabled	Never
	username		Enabled	Never

Commented [FN1]: @Nick Hurt - can you also attach a screen shot of variable library if they decide to use a Service account

4. Create another variable group called Fabric_Deployment_Group_NS and populate the required key value pairs as shown in the image below.



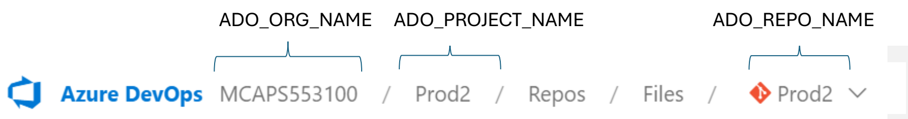
Variable group name: Fabric_Deployment_Group_NS

Description:

Name	Value
ADO_API_URL	https://dev.azure.com
ADO_ORG_NAME	MCAPS553100
ADO_PROJECT_NAME	Prod2
ADO_REPO_NAME	Prod2
NOTEBOOK_ID	8e19aa22-d422-46dc-a51a-7138dc2855c5
NOTEBOOK_WORKSPACE_ID	0F7C5C82-22C0-451E-A71D-458D2968EFFF
TENANT_ID	a260adfe-d5a8-4402-8633-64c0776f41e6

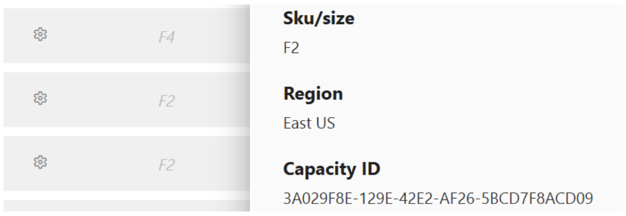
ADO_API_URL: normally will be https://dev.azure.com

ADO_ORG_NAME, ADO_PROJECT_NAME, ADO_REPO_NAME can be obtained when viewing the Repos page in Azure DevOps as shown below



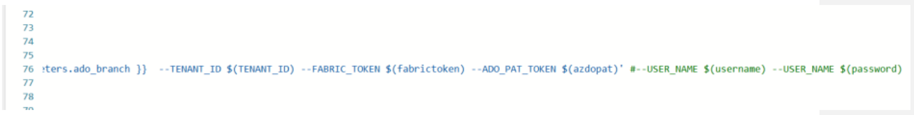
5. Navigate to the repos page and add / upload the YAML file to the root of the repo.
6. Edit the YAML file and review and change the pipeline and script parameters as necessary:
    a. The default parameters section contains relevant default values for example, the default capacity to be assigned to the new workspace. The capacity ID can be obtained in the Fabric Admin portal. Navigate to the capacities tab,

click on the actions icon to display the capacity details.



F4	**Sku/size**
	F2
F2	**Region**
	East US
F2	**Capacity ID**
	3A029F8E-129E-42E2-AF26-5BCD7F8ACD09

b. the parameters on line 76 are configured correctly depending which authentication method has been chosen:

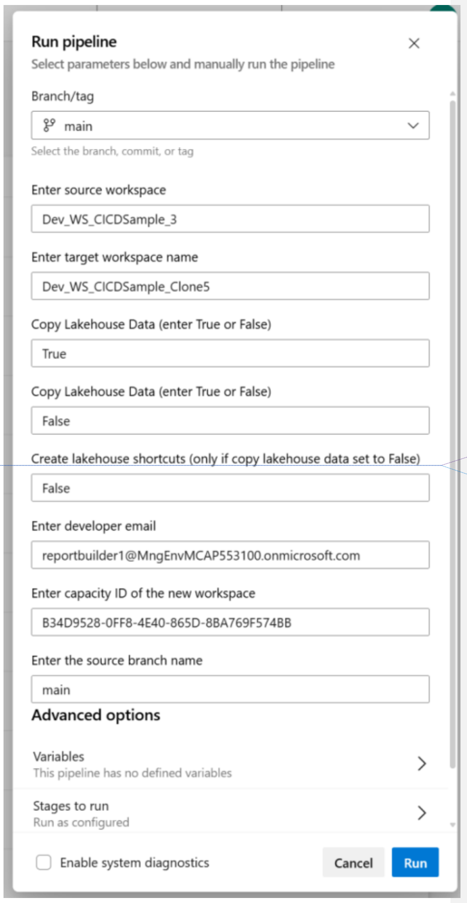


```
72
73
74
75
76 ters.ado_branch }}  --TENANT_ID $(TENANT_ID) --FABRIC_TOKEN $(fabrictoken) --ADO_PAT_TOKEN $(azdopat)' #--USER_NAME $(username) --USER_NAME $(password)
77
78
```

7. Add a new folder to the root of the repo called scripts and upload the associated python scripts.
8. Create a new release pipeline, chose Azure Repos Git, select the repository, chose Existing YAML file and select the uploaded YAML file.
9. Go back to the variable groups created in step 4 and click the pipeline permissions button to allow the pipeline to use the variable group.
10. Go back to the release pipeline created, click on the release pipeline name and click the “Run Pipeline” button.
11. This will display run-time parameters which can be modified as necessary:

a. Source workspace: Name of the dev workspace
b. Target workspace: Name of the new workspace to be created which will also serve as the new branch name
c. Copy Lakehouse data: Enter True or False depending on whether you wish to copy lakehouse data from the source workspace
d. Copy Warehouse data: Enter True or False depending on whether you wish to copy warehouse data from the source workspace
e. Create Lakehouse Shortcuts: Only applicable if Copy Lakehouse Data is False. This configures shortcuts pointing back to the source lakehouse
f. Swap connections in pipelines: Specify connections to be replaced using format (from1,to1),(from2,to2),...(fromN,toN) using either connection ID or name.
g. Enter Developer Email: Add the email address of the developer to be granted admin role on the new workspace
h. Enter Capacity ID: Enter the capacity ID of the new workspace if different from the default GUID.
i. Enter the branch name: Enter the source branch name which the new branch will be created from.
j. Click Run and monitor the release pipeline in Azure Devops and also the progress of the post activity notebook in the Fabric monitoring hub.

12. To debug and monitor the running YAML pipeline click on the “BranchOut” job


Run pipeline
Select parameters below and manually run the pipeline

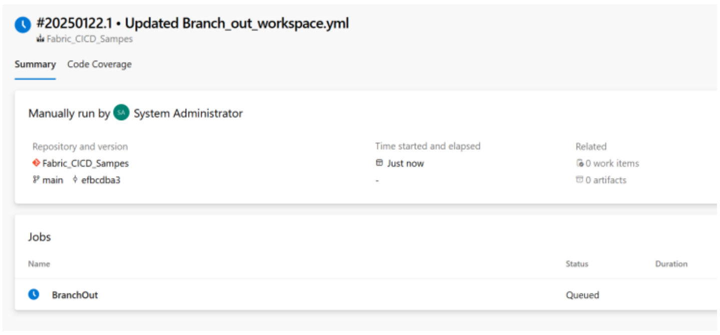
Branch/tag	main
Enter source workspace	Dev_WS_CICDSample_3
Enter target workspace name	Dev_WS_CICDSample_Clone5
Copy Lakehouse Data (enter True or False)	True
Copy Lakehouse Data (enter True or False)	False
Create lakehouse shortcuts (only if copy lakehouse data set to False)	False
Enter developer email	reportbuilder1@MngEnvMCAP553100.onmicrosoft.com
Enter capacity ID of the new workspace	B34D9528-0FF8-4E40-865D-8BA769F574BB
Enter the source branch name	main

Select the branch, commit, or tag

Advanced options
Variables
This pipeline has no defined variables
Stages to run
Run as configured
Enable system diagnostics

Commented [NS2]: @Nick Hurt how is this being done?

Commented [NH3R2]: It uses notebookutils filesystem copy to copy lakehouse data. For warehouse I have a process which creates a data pipeline on the fly which runs a parameterised copy. I use this in my BCDR accelerator notebooks


#20250122.1 • Updated Branch_out_workspace.yml
Fabric_CICD_Sampes

Summary | Code Coverage

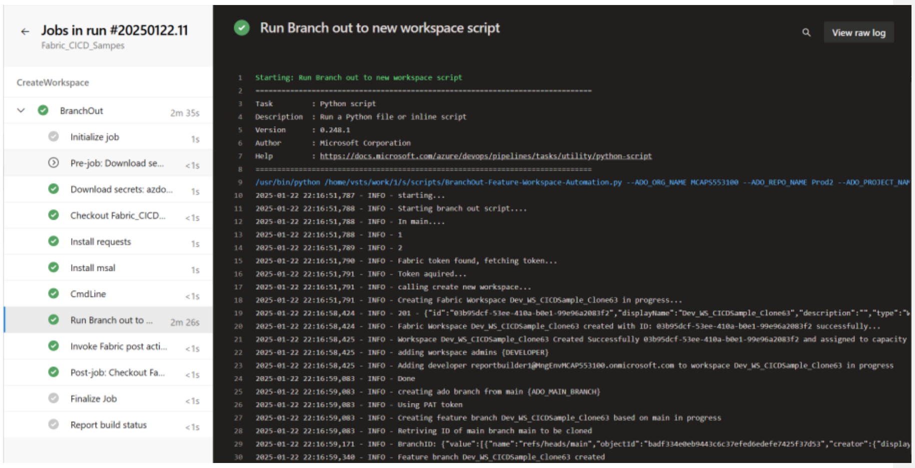
Manually run by System Administrator

Repository and version	Time started and elapsed	Related
Fabric_CICD_Sampes	Just now	0 work items
main efbcdba3	-	0 artifacts

Jobs

Name	Status	Duration
BranchOut	Queued	

Importantly ensure both python scripts complete successfully.


Jobs in run #20250122.11
Fabric_CICD_Sampes

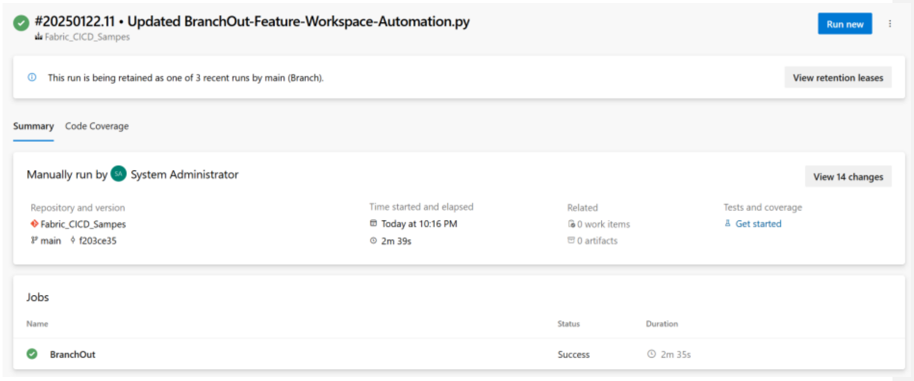
CreateWorkspace

BranchOut	2m 35s
Initialize job	1s
Pre-job: Download se...	<1s
Download secrets: azdo...	1s
Checkout Fabric_CICD...	<1s
Install requests	1s
Install msal	1s
CmdLine	<1s
Run Branch out to ...	2m 26s
Invoke Fabric post acti...	<1s
Post-job: Checkout Fa...	<1s
Finalize Job	<1s
Report build status	<1s

Run Branch out to new workspace script

```
Starting: Run Branch out to new workspace script
==============================================================================
Task         : Python script
Description  : Run a Python file or inline script
Version      : 0.248.1
Author       : Microsoft Corporation
Help         : https://docs.microsoft.com/azure/devops/pipelines/tasks/utility/python-script
==============================================================================
/usr/bin/python /home/vsts/work/1/s/scripts/BranchOut-Feature-Workspace-Automation.py --ADO_ORG_NAME MCAPS553100 --ADO_REPO_NAME Prod2 --ADO_PROJECT_NAM
2025-01-22 22:16:51,787 - INFO - starting...
2025-01-22 22:16:51,788 - INFO - Starting branch out script....
2025-01-22 22:16:51,788 - INFO - In main....
2025-01-22 22:16:51,788 - INFO - 1
2025-01-22 22:16:51,789 - INFO - 2
2025-01-22 22:16:51,790 - INFO - Fabric token found, fetching token...
2025-01-22 22:16:51,791 - INFO - Token aquired...
2025-01-22 22:16:51,791 - INFO - calling create new workspace...
2025-01-22 22:16:51,791 - INFO - Creating Fabric Workspace Dev_WS_CICDSample_Clone63 in progress...
2025-01-22 22:16:58,424 - INFO - 201 - {"id":"03b95dcf-53ee-410a-b0e1-99e96a2083f2","displayName":"Dev_WS_CICDSample_Clone63","description":"","type":"W
2025-01-22 22:16:58,424 - INFO - Fabric Workspace Dev_WS_CICDSample_Clone63 created with ID: 03b95dcf-53ee-410a-b0e1-99e96a2083f2 successfully...
2025-01-22 22:16:58,425 - INFO - Workspace Dev_WS_CICDSample_Clone63 Created Successfully 03b95dcf-53ee-410a-b0e1-99e96a2083f2 and assigned to capacity
2025-01-22 22:16:58,425 - INFO - adding workspace admins {DEVELOPER}
2025-01-22 22:16:58,425 - INFO - Adding developer reportbuilder1@MngEnvMCAP553100.onmicrosoft.com to workspace Dev_WS_CICDSample_Clone63 in progress
2025-01-22 22:16:59,083 - INFO - Done
2025-01-22 22:16:59,083 - INFO - creating ado branch from main {ADO_MAIN_BRANCH}
2025-01-22 22:16:59,083 - INFO - Using PAT token
2025-01-22 22:16:59,083 - INFO - Creating feature branch Dev_WS_CICDSample_Clone63 based on main in progress
2025-01-22 22:16:59,083 - INFO - Retriving ID of main branch main to be cloned
2025-01-22 22:16:59,171 - INFO - BranchID: {"value":[{"name":"refs/heads/main","objectId":"badf334e0eb9443c6c37efed6edefe7425f37d53","creator":{"display
2025-01-22 22:16:59,340 - INFO - Feature branch Dev_WS_CICDSample_Clone63 created
```

Azure DevOps success will be displayed on the previous job screen


#20250122.11 • Updated BranchOut-Feature-Workspace-Automation.py
Fabric_CICD_Sampes

This run is being retained as one of 3 recent runs by main (Branch).

Summary | Code Coverage

Manually run by System Administrator

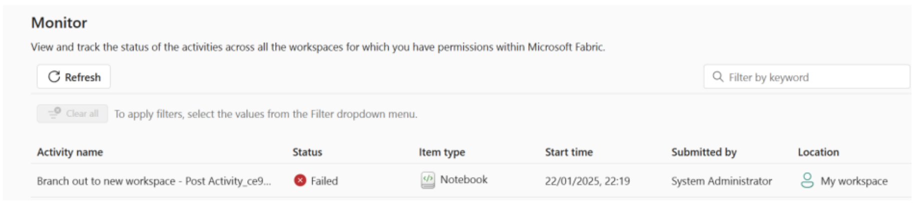
Repository and version	Time started and elapsed	Related	Tests and coverage
Fabric_CICD_Sampes	Today at 10:16 PM	0 work items	Get started
main f203ce35	2m 39s	0 artifacts	

Jobs

Name	Status	Duration
BranchOut	Success	2m 35s

13. The second python script asynchronously invokes the post activity notebook in Fabric, therefore navigate to the monitoring hub in Fabric to ensure the notebook

completes successfully. To debug or troubleshoot notebook errors, click on the failed activity name in the monitoring hub

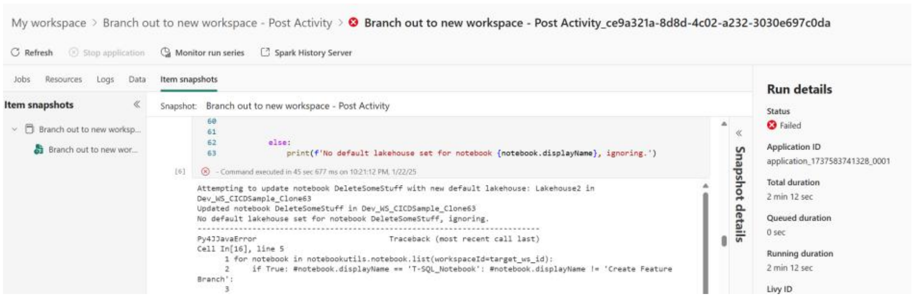

Monitor

View and track the status of the activities across all the workspaces for which you have permissions within Microsoft Fabric.

To apply filters, select the values from the Filter dropdown menu.

Activity name	Status	Item type	Start time	Submitted by	Location
Branch out to new workspace - Post Activity_ce9...	Failed	Notebook	22/01/2025, 22:19	System Administrator	My workspace

Click on item snapshots and review the output of the notebook.


My workspace > Branch out to new workspace - Post Activity > Branch out to new workspace - Post Activity_ce9a321a-8d8d-4c02-a232-3030e697c0da

Snapshot: Branch out to new workspace - Post Activity

```
            else:
                print(f'No default lakehouse set for notebook {notebook.displayName}, ignoring.')
```

[6] - Command executed in 45 sec 677 ms on 10:21:12 PM, 1/22/25

```
Attempting to update notebook DeleteSomeStuff with new default lakehouse: Lakehouse2 in
Dev_WS_CICDSample_Clone63
Updated notebook DeleteSomeStuff in Dev_WS_CICDSample_Clone63
No default lakehouse set for notebook DeleteSomeStuff, ignoring.
---------------------------------------------------------------------------
Py4JJavaError                             Traceback (most recent call last)
Cell In[16], line 5
      1 for notebook in notebookutils.notebook.list(workspaceId=target_ws_id):
      2     if True: #notebook.displayName == 'T-SQL_Notebook': #notebook.displayName != 'Create Feature
Branch':
      3
```

Run details

Status: Failed

Application ID: application_1737583741328_0001

Total duration: 2 min 12 sec

Queued duration: 0 sec

Running duration: 2 min 12 sec

Livy ID

After successfully running the pipeline the new workspace should set up and ready for the developer to access and start developing.

Commented [FN4]: @Nick Hurt - can you attach a sample successful run image of ADO pipeline here as a reference

Commented [FN5R4]: And If you have screen shot where these parameters are populated , pls attach that also for users to understand how these parameter looks like

Appendix A

```
import requests

nameOfKeyVault = 'azdosc'

secret_value = notebookutils.credentials.getToken('pbi')

secret_name = "fabrictoken"

access_token = notebookutils.credentials.getToken('keyvault')

url = f'https://{nameOfKeyVault}.vault.azure.net/secrets/{secret_name}?api-
version=7.3'

headers = {

    'Authorization': f'Bearer {access_token}',

    'Content-Type': 'application/json'

}

body = {

    'value': secret_value

}

response = requests.put(url, headers=headers, json=body)

if response.status_code == 200:

    print(f"Secret '{secret_name}' added to Key Vault.")

else:

    print(f"Failed to add secret: {response.status_code} - {response.text}")
```