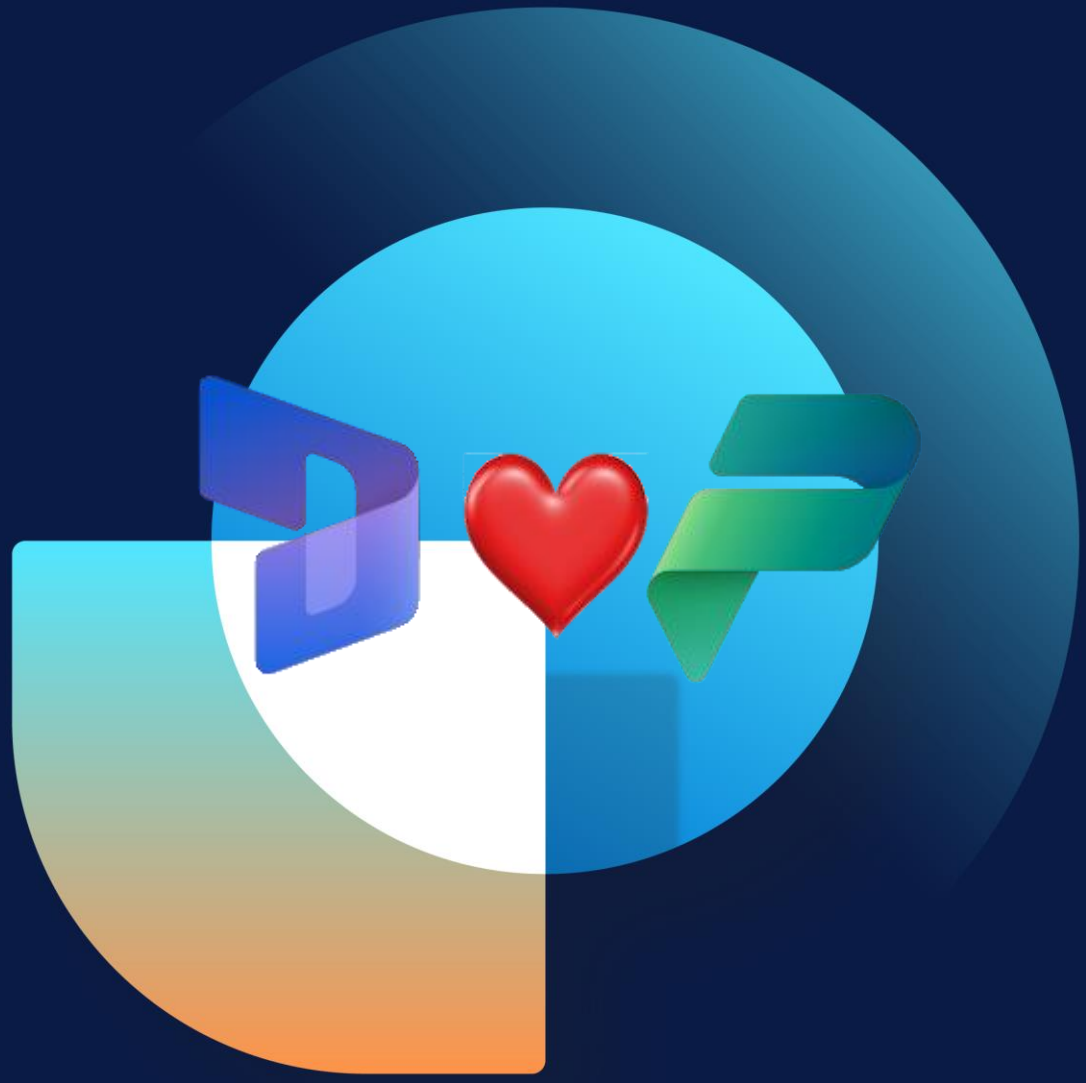
Microsoft Dynamics 365

Unified Dev ALM for Finance & Operations

Presenter(s):
Ankur Srivastava
Pankaj Thakur
Priyanka Sinha
Sourabh Namilikonda

Co-Presenter(s)/Q&A:
Peter Villadsen
Saurabh Kuchhal
Suresh Kotapalle
Takamitsu Endo



Pankaj Thakur

Principal Software Engineer

Email: Pankaj.Thakur@microsoft.com
LinkedIn: Pankaj Thakur | LinkedIn

Sourabh Namilikonda
Software Engineer II

Email: snamilikonda@microsoft.com

LinkedIn: Sourabh Namilikonda | LinkedIn

Ankur Srivastava

Senior Solution Architect

Email: ankur.srivastava@microsoft.com

LinkedIn: Ankur Srivastava | LinkedIn

Priyanka Sinha

Senior Solution Architect

Email: prsinha@microsoft.com

LinkedIn: Priyanka Sinha | LinkedIn

TechTalk Series

- Session 1 - Unified Admin Experience (YouTube Link)
- Session 2 – Unified Dev Experience (YouTube Link)
- *Session 3 – Unified Dev ALM*

Unified Dev ALM

Agenda

Objectives

Azure DevOps (ADO) and Git

Version Control

Branching strategy

Getting started with development and release

Continuous integration and deployment

Resources

Q&A

Microsoft Dynamics 365

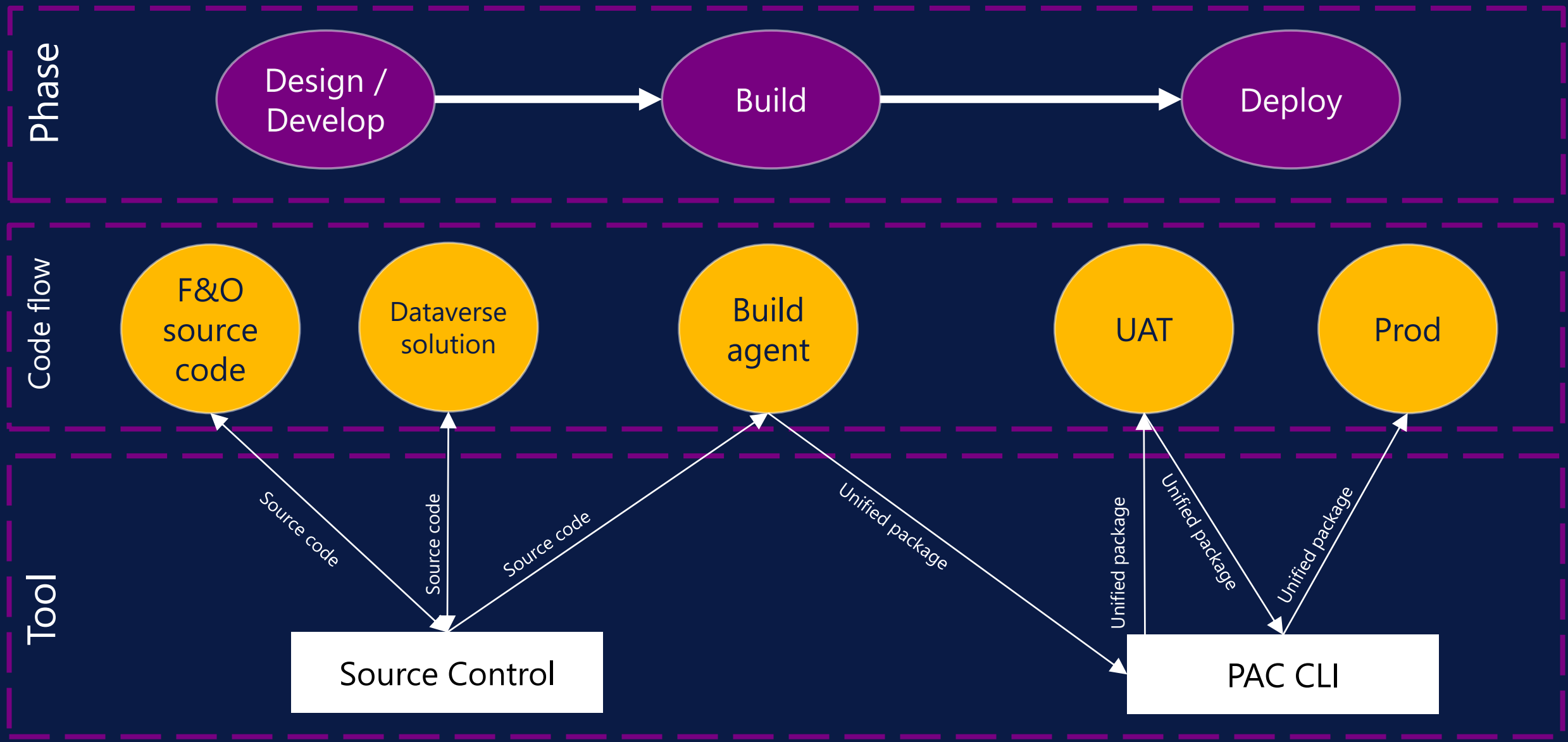
Objectives

Priyanka Sinha

What could you do with Dev ALM – F&O and Dataverse

- We recommend using Git for version control though we continue to support Team Foundation Version Control (TFVC) for Finance and Operations
- We support Azure Dev Ops for artifacts, build and release of F&O
- We provide Azure Dev Ops tasks towards builds, deployments and tests
- Consistent Development policies across the apps and development teams
- Auditing, approval, change management, signing, …

Unified Application Lifecycle Management



Microsoft Dynamics 365

Getting started on Azure DevOps (ADO)

Priyanka Sinha

Configure Azure DevOps

- Sign up to Azure DevOps, create an account, and create a new project.
- When setting up Azure DevOps, prefer to select Git as version control, as Git is widely recognized version control system in the software industry however Team Foundation Version Control (TFVC) can also be used.
- Setup "Disable creation of TFVC repositories" in **Organization setting > Repos > Repositories,** which will only affect the creation of new TFVC repositories and won't impact existing ones.
- Create an Azure DevOps team project.
- Create the recommended folder structure in your team project.
- Init repository

Configure your Azure DevOps organization and project

Configure Azure DevOps - Finance and Operations

- Install Dynamics 365 Finance and Operations Tools - Visual Studio Marketplace in your DevOps project
- Connect Visual Studio to your team project
- Map/Clone your Azure DevOps project to your local model store and projects folder

Configure Azure DevOps – Customer Engagement

- Install Power Platform Build Tools for Azure DevOps in your DevOps project
- Install Dynamics 365 Finance and Operations Tools - Visual Studio Marketplace in your DevOps project
- Create service connections for Dataverse environments with configured service principal

Additional details:

https://learn.microsoft.com/en-us/power-platform/alm/devops-build-tools
https://powerusers.microsoft.com/t5/Power-Apps-Community-Blog/Detail-Step-By-Step-Power-Platform-ALM-with-Azure-DevOps/ba-p/1976808

Microsoft Dynamics 365

Version Control

Ankur Srivastava

Version control using Git

Distributed Version Control System

Local Commit and Version Control Operations

Lightweight Branching

Easy Context Switching

Merge and Publish

Integration with Visual Studio and Azure DevOps

TFVC vs Git terminology

Operation	TFVC Workflow	Git Workflow
Create workspace	Create Workspace	N/A
Delete workspace	Delete Workspace	N/A
Code Download \ Configure your workspace	Create Workspace >> Map & Get	Create Repository >> Clone
Get Latest Code (First Time)	Get Latest Version	Clone
Get Latest Code (After First Time)	Get Latest Version	Pull
Code Commit	Check In	Commit + Push
Get Latest After Code Commit	Check In + Get Latest Version	Sync
Check Out a File for Editing	Check Out	Just start editing
Review Code	Code Review	Pull Request
Shelving Code	Shelveset	Stash
Files to be Included for Commit	Included Changes	Staged
Files to be Excluded for Commit	Exclude Changes	Unstage
Code Annotation	Annotate	Blame
View History	History	Log
Create Branch	Branch	Checkout -b
Delete Branch	N/A	Branch -d
Switch Branch	N/A	Checkout
Merge Changes	Merge	Merge
Resolve Conflicts	Resolve	Pull \ Merge >> Resolve Conflicts
Add Remote Repository	N/A	Add Remote
Remove Remote Repository	N/A	Remove Remote
List Remote Repositories	N/A	List Remote

Map TFVC actions to Git - Azure Repos | Microsoft Learn

Source Control Guidelines

- Default Target Branch Locked
- Merges Through Pull Requests (PRs)
- PRs Reference Work Items
- Consistent Commit History
- Branch Naming Conventions
- Clear Repository Documentation
- Secrets Management
- Public Repository Guidelines

Engineering Fundamentals Checklist - Engineering Fundamentals Playbook (microsoft.github.io)

Microsoft Dynamics 365

Branching

Ankur Srivastava

Branching strategy

Branch configurations can vary depending on the development team's preferences and the Dynamics 365 implementation lifecycle.

Team should consider minimum branching criteria –

- Isolation of untested development code (code that's in development)
- Isolation of in-development code from test-eligible code
- Isolation of in-test code from production code
- Support for Long Functional Testing Cycles

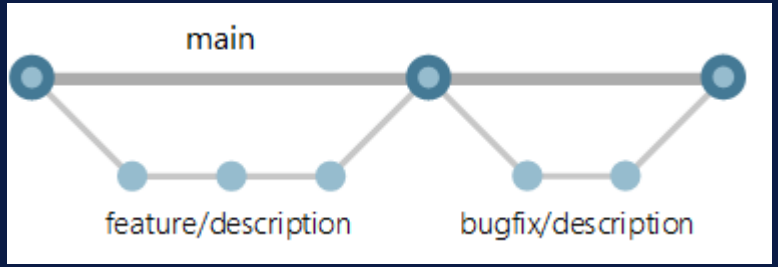
Ideally, the branching strategy should align with all requirements posed by the project plan. Scenarios can include –

- Developer isolation of features not to be pushed forward to production in the current phase
- Stabilized branches serving UAT/SIT (user acceptance testing / system integration testing) scenarios
- Hotfix branches enabling creating hotfix packages to be release to production

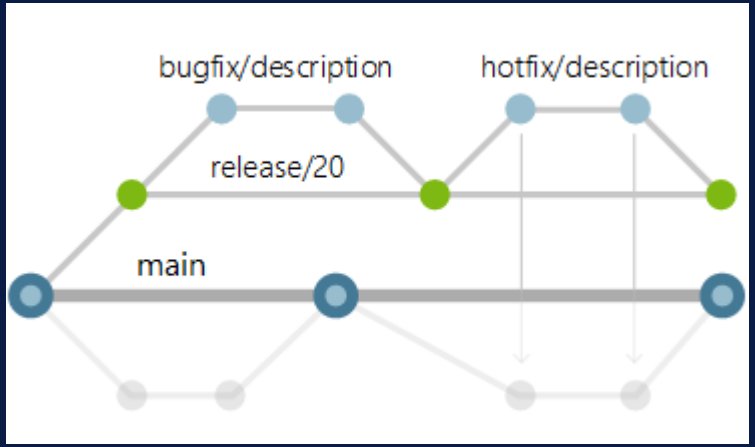
Branching in Git

- Use feature branches for all new features and bug fixes.
  - Develop your features and fix bugs in feature branches based off main branch
  - Even small fixes and changes should have their own feature branch.
- Review and merge code with pull requests
  - Merge branches through pull requests that pass your review process.
- Keep a high quality, up-to-date main branch.
  - The code in your main branch should pass tests, build cleanly, and always be current.
  - Set up a branch policy for your main branch
- Use release branches
  - Create a release branch from the main branch to manage releases
  - Create branches to fix bugs from the release branch and merge them back into the release branch in a pull request

Managing Development



Managing Release



Git branching guidance - Azure Repos | Microsoft Learn

Microsoft Dynamics 365

Development flow

Pankaj Thakur

Development flow (inner loop)

Enlist

- Developer clones the Azure DevOps repository on his development machine

Branch

- Developer creates a new branch off the main integration branch

Push

- Developer is ready to integrate and ship changes to the rest of the team, they push their local branch to a branch on the server and open a pull request

Pull request

- Use branch policies and ensure that branch policies are satisfied
- Review the code and approve the change

Merge

- Once the pull request satisfies all build policies and reviewers have signed off, the topic branch merges into the main integration branch, and the pull request is complete

Get started with Git and Visual Studio - Azure Repos

Work | UDEDevALM - Repos | Summary - Overview | Settings · Repositories (Helix | build-mergevalidation-build | MSDynEngCloudRuntime - A | Pipelines - Run 334 logs | Pipelines - All | Pipelines - Runs for build-me | Pipelines - Run 24.06.25.2 an | CloudRuntime3PCICDPipelin

https://dev.azure.com/ProgrammingModel/Helix

Attendee -- Finance... | First Party Forums.d... | POSTer | Sarif Viewer | LiveSite | 0531 | PL100 | PI400 | EngineeringSys | Prep1 | UnoDev | Work | OCI | Code | Learn | Microsoft | Mentoring | Samples | OP2OL | ODOP | Home | LT | PowerPLat | Read | TPrompt Chat | CS128 - Overview | CAPSLA | DocFx

Helix

Overview
Summary
Dashboards
Wiki
Boards
Repos
Pipelines
Test Plans
Artifacts
Compliance
Start Right

Project settings

Helix

Private

About this project

Help others to get on board!

Describe your project and make it easier for other people to understand it.

+ Add Project Description

Project stats

Period: Last 7 days

Repos

1
Pull requests opened

67
Commits and Changesets by 3 authors

Pipelines

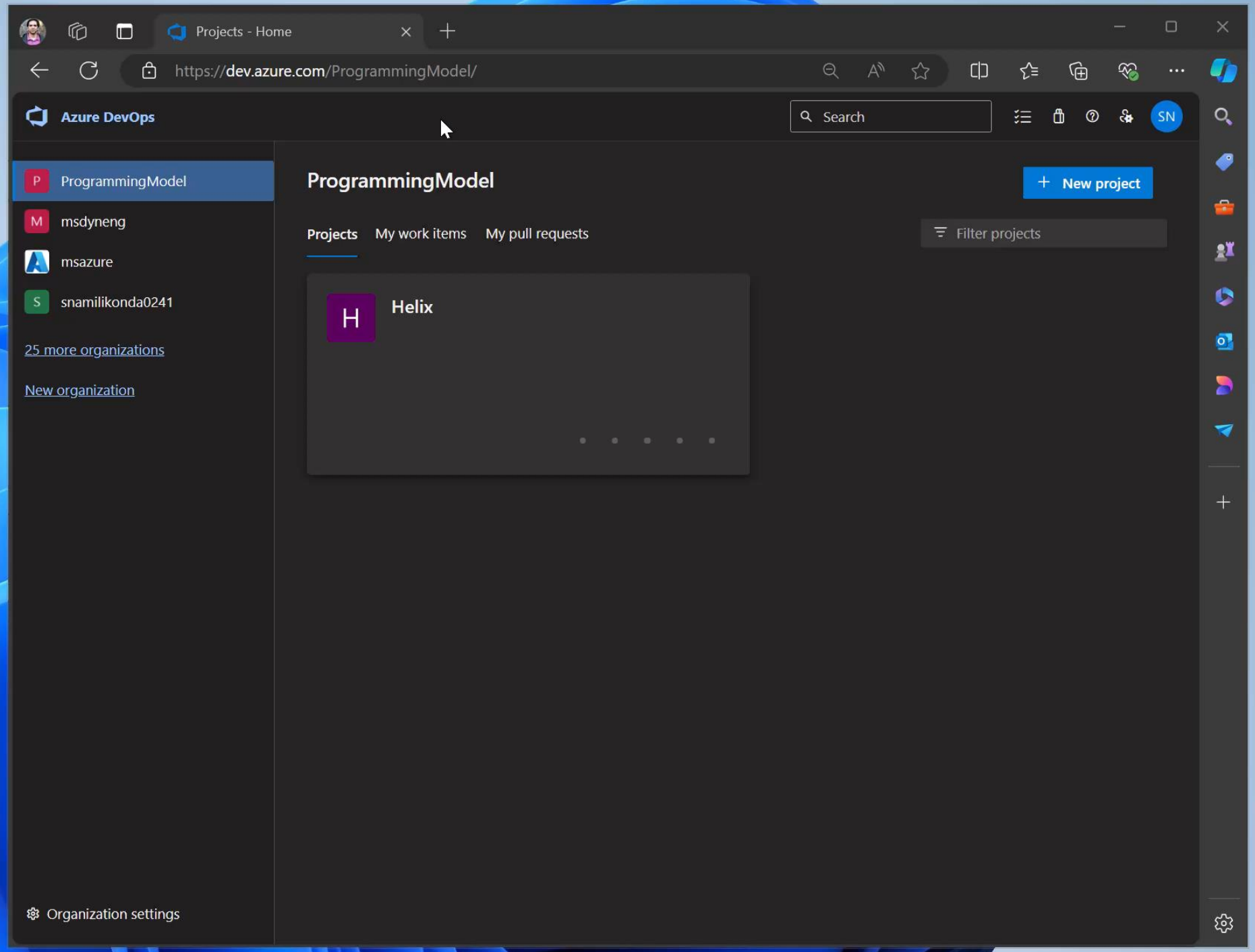
41%
Builds succeeded

43%
Deployments succeeded

Members 20

TS RUNE JL SN PT NG

clone the repository


Projects - Home
https://dev.azure.com/ProgrammingModel/
Azure DevOps
Search
SN
ProgrammingModel
msdyneng
msazure
snamilikonda0241
25 more organizations
New organization
Organization settings
ProgrammingModel
New project
Projects
My work items
My pull requests
Filter projects
Helix

Microsoft Dynamics 365

Continuous integration and deployment

Sourabh Namilikonda

Continuous Integration and Deployment (outer loop)

Install Build Tools

- Dynamics 365 Finance and Operations Tools
- Power Platform Build Tools

Create Service Connection

- Install Pac CLI Microsoft Power Platform CLI - Power Platform | Microsoft Learn
- Create connection to environment

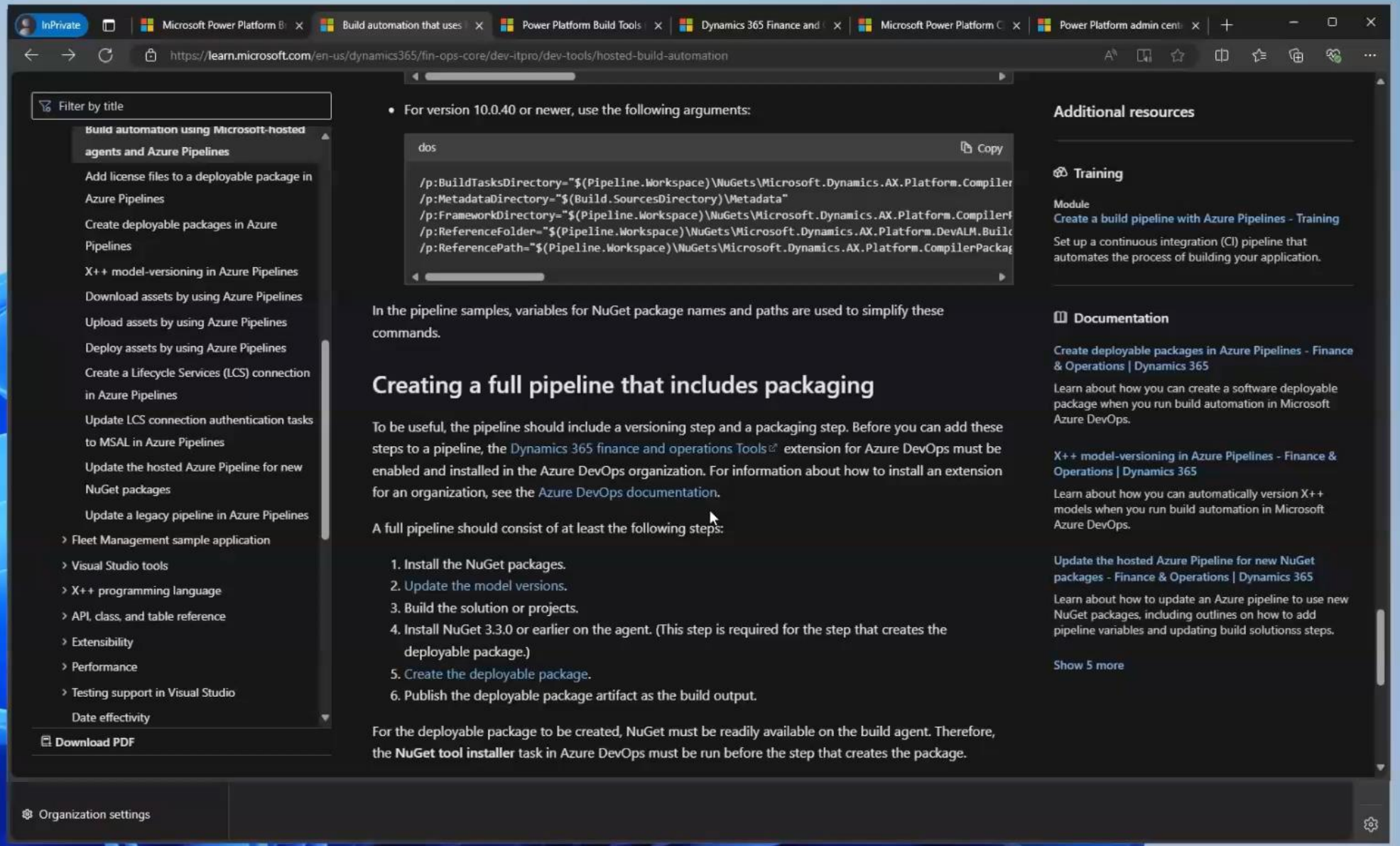
Create build pipelines

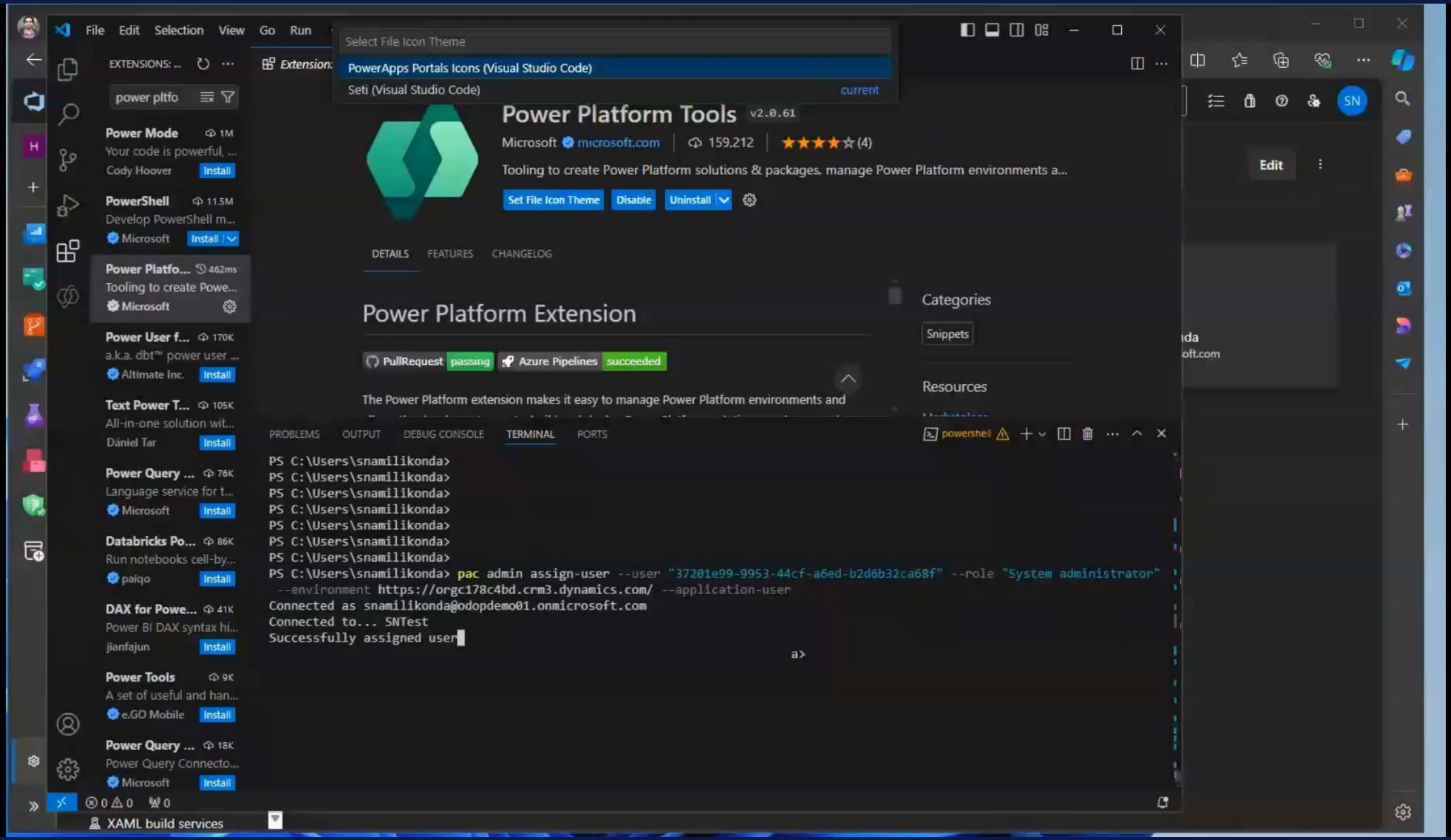
- Create/use existing build pipeline
- Continuous integration and deployment - Power Platform | Microsoft Learn

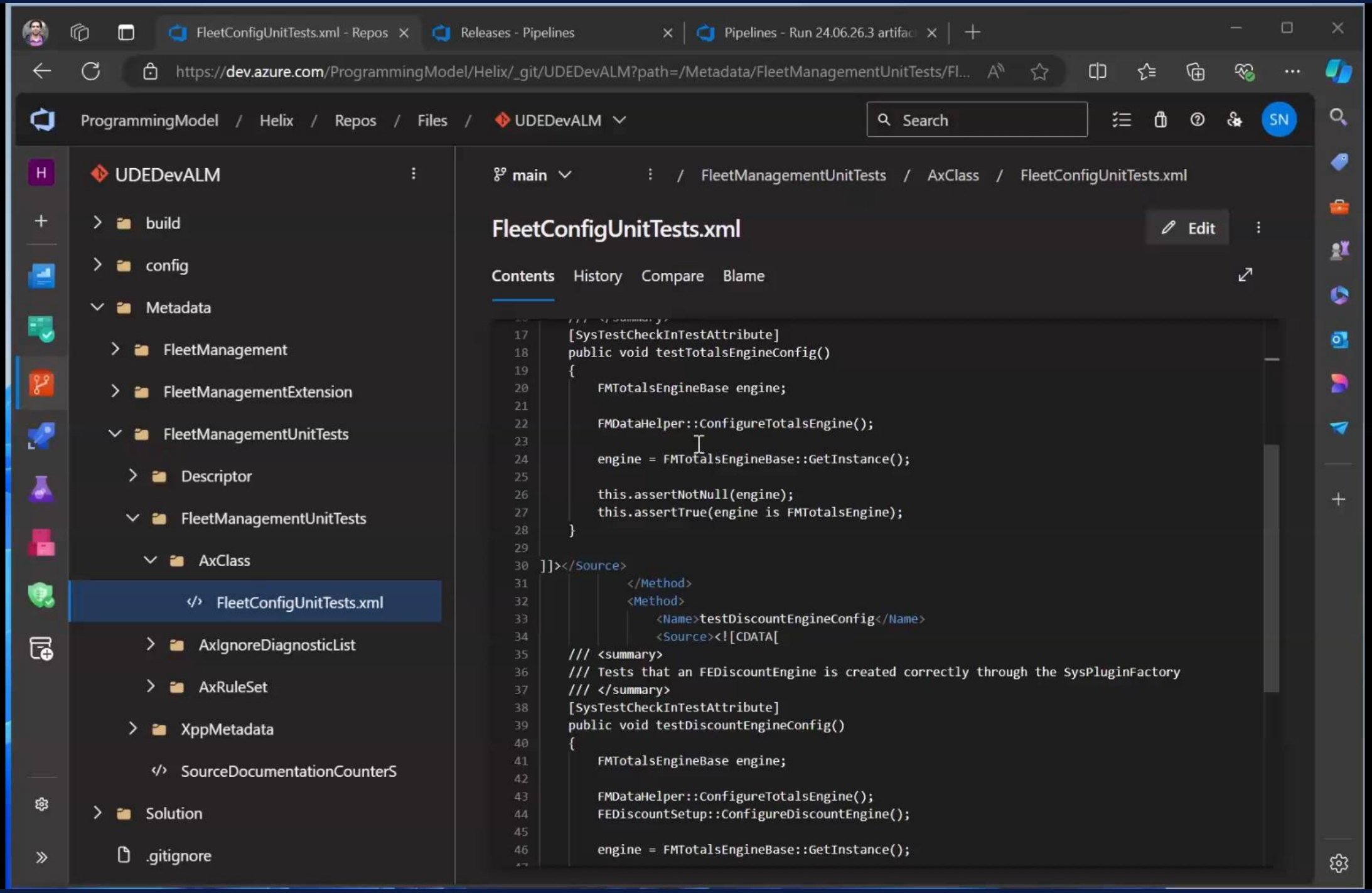
Create deployment pipelines

- Build tool tasks - Power Platform | Microsoft Learn

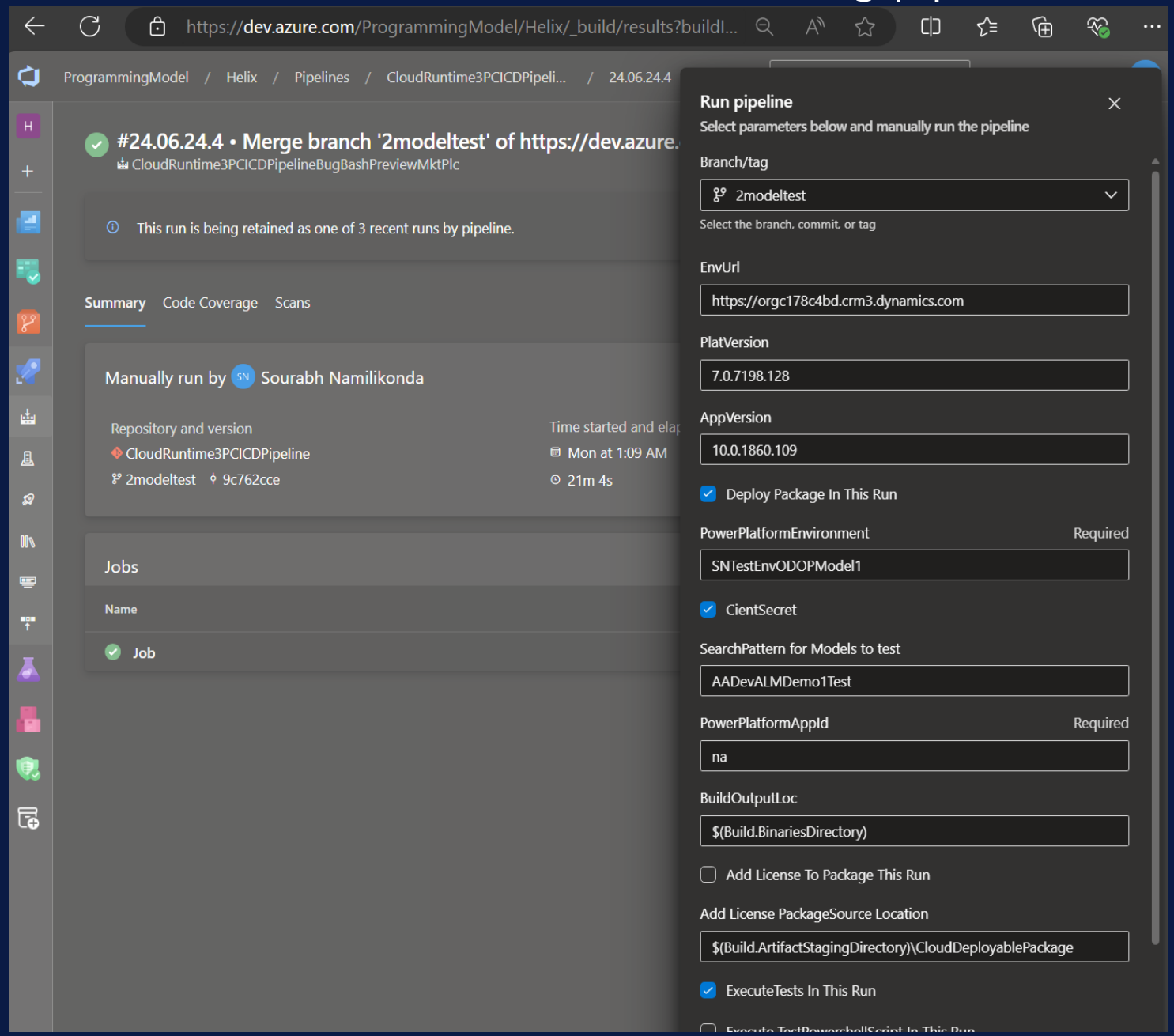
Get started with Git and Visual Studio - Azure Repos

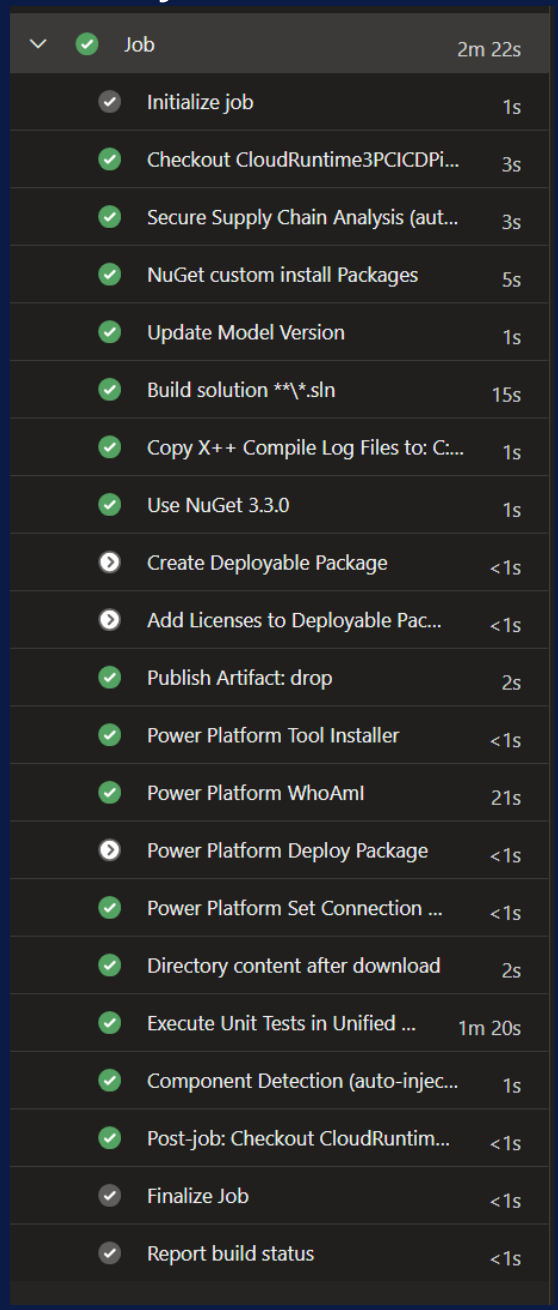

InPrivate
Microsoft Power Platform B
Build automation that uses
Power Platform Build Tools
Dynamics 365 Finance and
Microsoft Power Platform C
Power Platform admin cent
https://learn.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/dev-tools/hosted-build-automation
Filter by title
Build automation using Microsoft-hosted agents and Azure Pipelines
Add license files to a deployable package in Azure Pipelines
Create deployable packages in Azure Pipelines
X++ model-versioning in Azure Pipelines
Download assets by using Azure Pipelines
Upload assets by using Azure Pipelines
Deploy assets by using Azure Pipelines
Create a Lifecycle Services (LCS) connection in Azure Pipelines
Update LCS connection authentication tasks to MSAL in Azure Pipelines
Update the hosted Azure Pipeline for new NuGet packages
Update a legacy pipeline in Azure Pipelines
Fleet Management sample application
Visual Studio tools
X++ programming language
API, class, and table reference
Extensibility
Performance
Testing support in Visual Studio
Date effectivity
Download PDF
Organization settings
For version 10.0.40 or newer, use the following arguments:
dos
Copy
/p:BuildTasksDirectory="$(Pipeline.Workspace)\NuGets\Microsoft.Dynamics.AX.Platform.Compiler
/p:MetadataDirectory="$(Build.SourcesDirectory)\Metadata"
/p:FrameworkDirectory="$(Pipeline.Workspace)\NuGets\Microsoft.Dynamics.AX.Platform.CompilerF
/p:ReferenceFolder="$(Pipeline.Workspace)\NuGets\Microsoft.Dynamics.AX.Platform.DevALM.Build
/p:ReferencePath="$(Pipeline.Workspace)\NuGets\Microsoft.Dynamics.AX.Platform.CompilerPackag
In the pipeline samples, variables for NuGet package names and paths are used to simplify these commands.
Creating a full pipeline that includes packaging
To be useful, the pipeline should include a versioning step and a packaging step. Before you can add these steps to a pipeline, the Dynamics 365 finance and operations Tools extension for Azure DevOps must be enabled and installed in the Azure DevOps organization. For information about how to install an extension for an organization, see the Azure DevOps documentation.
A full pipeline should consist of at least the following steps:
1. Install the NuGet packages.
2. Update the model versions.
3. Build the solution or projects.
4. Install NuGet 3.3.0 or earlier on the agent. (This step is required for the step that creates the deployable package.)
5. Create the deployable package.
6. Publish the deployable package artifact as the build output.
For the deployable package to be created, NuGet must be readily available on the build agent. Therefore, the NuGet tool installer task in Azure DevOps must be run before the step that creates the package.
Additional resources
Training
Module
Create a build pipeline with Azure Pipelines - Training
Set up a continuous integration (CI) pipeline that automates the process of building your application.
Documentation
Create deployable packages in Azure Pipelines - Finance & Operations | Dynamics 365
Learn about how you can create a software deployable package when you run build automation in Microsoft Azure DevOps.
X++ model-versioning in Azure Pipelines - Finance & Operations | Dynamics 365
Learn about how you can automatically version X++ models when you run build automation in Microsoft Azure DevOps.
Update the hosted Azure Pipeline for new NuGet packages - Finance & Operations | Dynamics 365
Learn about how to update an Azure pipeline to use new NuGet packages, including outlines on how to add pipeline variables and updating build solutionss steps.
Show 5 more

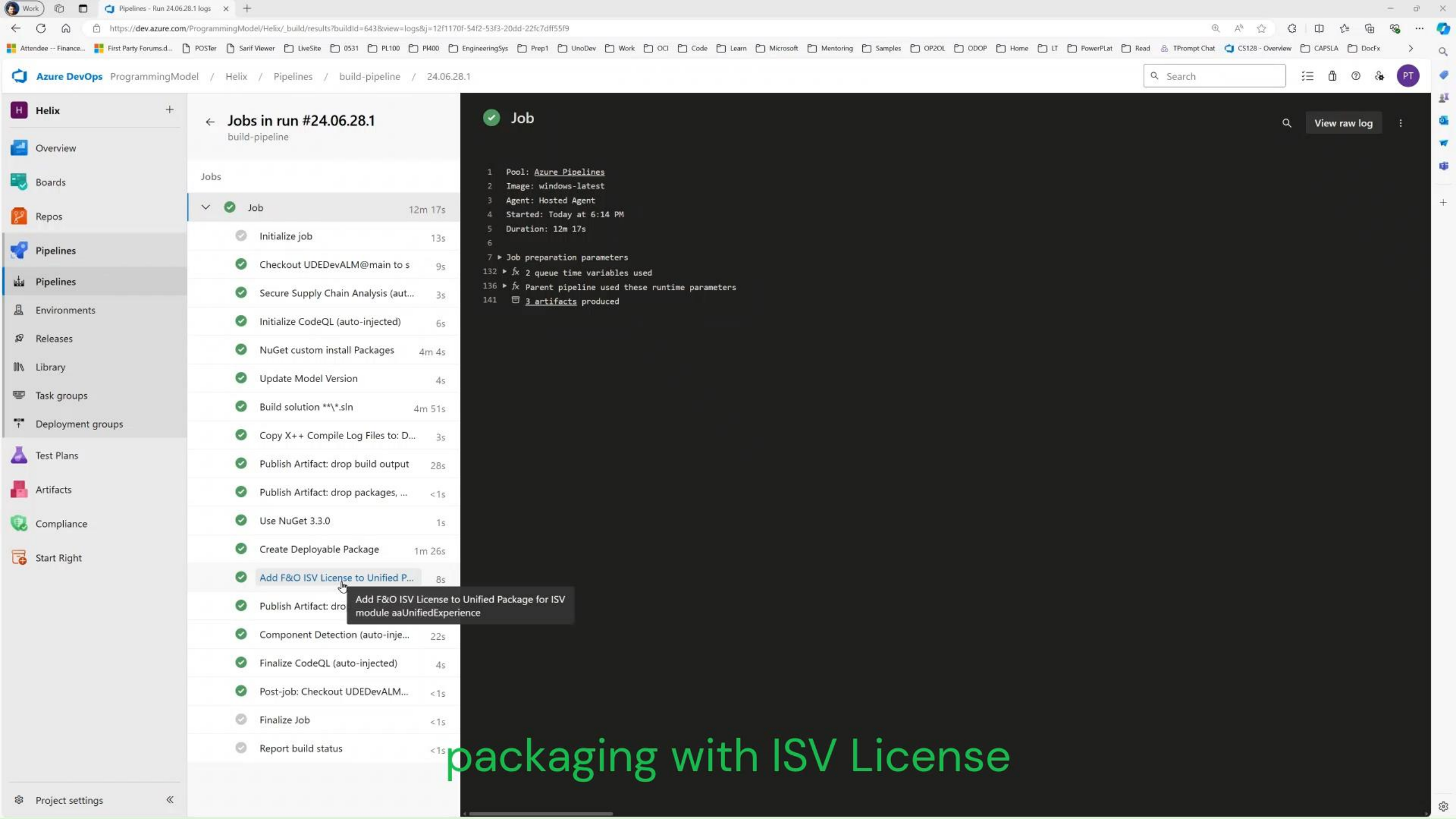

File Edit Selection View Go Run
Select File Icon Theme
PowerApps Portals Icons (Visual Studio Code)
Seti (Visual Studio Code) current
EXTENSIONS: ...
power pltfo
Power Mode 1M
Your code is powerful, ...
Cody Hoover Install
PowerShell 11.5M
Develop PowerShell m...
Microsoft Install
Power Platfo... 462ms
Tooling to create Powe...
Microsoft
Power User f... 170K
a.k.a. dbt™ power user ...
Altimate Inc. Install
Text Power T... 105K
All-in-one solution wit...
Dániel Tar Install
Power Query ... 76K
Language service for t...
Microsoft Install
Databricks Po... 86K
Run notebooks cell-by...
paiqo Install
DAX for Powe... 41K
Power BI DAX syntax hi...
jianfajun Install
Power Tools 9K
A set of useful and han...
e.GO Mobile Install
Power Query ... 18K
Power Query Connecto...
Microsoft Install
Power Platform Tools v2.0.61
Microsoft microsoft.com 159,212 (4)
Tooling to create Power Platform solutions & packages, manage Power Platform environments a...
Set File Icon Theme Disable Uninstall
DETAILS FEATURES CHANGELOG
Power Platform Extension
PullRequest passing Azure Pipelines succeeded
The Power Platform extension makes it easy to manage Power Platform environments and
Categories
Snippets
Resources
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
PS C:\Users\snamilikonda>
PS C:\Users\snamilikonda>
PS C:\Users\snamilikonda>
PS C:\Users\snamilikonda>
PS C:\Users\snamilikonda>
PS C:\Users\snamilikonda>
PS C:\Users\snamilikonda>
PS C:\Users\snamilikonda> pac admin assign-user --user "37201e99-9953-44cf-a6ed-b2d6b32ca68f" --role "System administrator"
--environment https://orgc178c4bd.crm3.dynamics.com/ --application-user
Connected as snamilikonda@odopdemo01.onmicrosoft.com
Connected to... SNTest
Successfully assigned user
a>
XAML build services
Edit
SN


FleetConfigUnitTests.xml - Repos
Releases - Pipelines
Pipelines - Run 24.06.26.3 artifac
https://dev.azure.com/ProgrammingModel/Helix/_git/UDEDevALM?path=/Metadata/FleetManagementUnitTests/Fl...
ProgrammingModel / Helix / Repos / Files / UDEDevALM
Search
SN
UDEDevALM
build
config
Metadata
FleetManagement
FleetManagementExtension
FleetManagementUnitTests
Descriptor
FleetManagementUnitTests
AxClass
FleetConfigUnitTests.xml
AxIgnoreDiagnosticList
AxRuleSet
XppMetadata
SourceDocumentationCounterS
Solution
.gitignore
main / FleetManagementUnitTests / AxClass / FleetConfigUnitTests.xml
FleetConfigUnitTests.xml
Edit
Contents
History
Compare
Blame
[SysTestCheckInTestAttribute]
public void testTotalsEngineConfig()
{
FMTotalsEngineBase engine;
FMDataHelper::ConfigureTotalsEngine();
engine = FMTotalsEngineBase::GetInstance();
this.assertNotNull(engine);
this.assertTrue(engine is FMTotalsEngine);
}
]]></Source>
</Method>
<Method>
<Name>testDiscountEngineConfig</Name>
<Source><![CDATA[
/// <summary>
/// Tests that an FEDiscountEngine is created correctly through the SysPluginFactory
/// </summary>
[SysTestCheckInTestAttribute]
public void testDiscountEngineConfig()
{
FMTotalsEngineBase engine;
FMDataHelper::ConfigureTotalsEngine();
FEDiscountSetup::ConfigureDiscountEngine();
engine = FMTotalsEngineBase::GetInstance();

Other mechanisms to execute tests using pipelines (sample demo only)





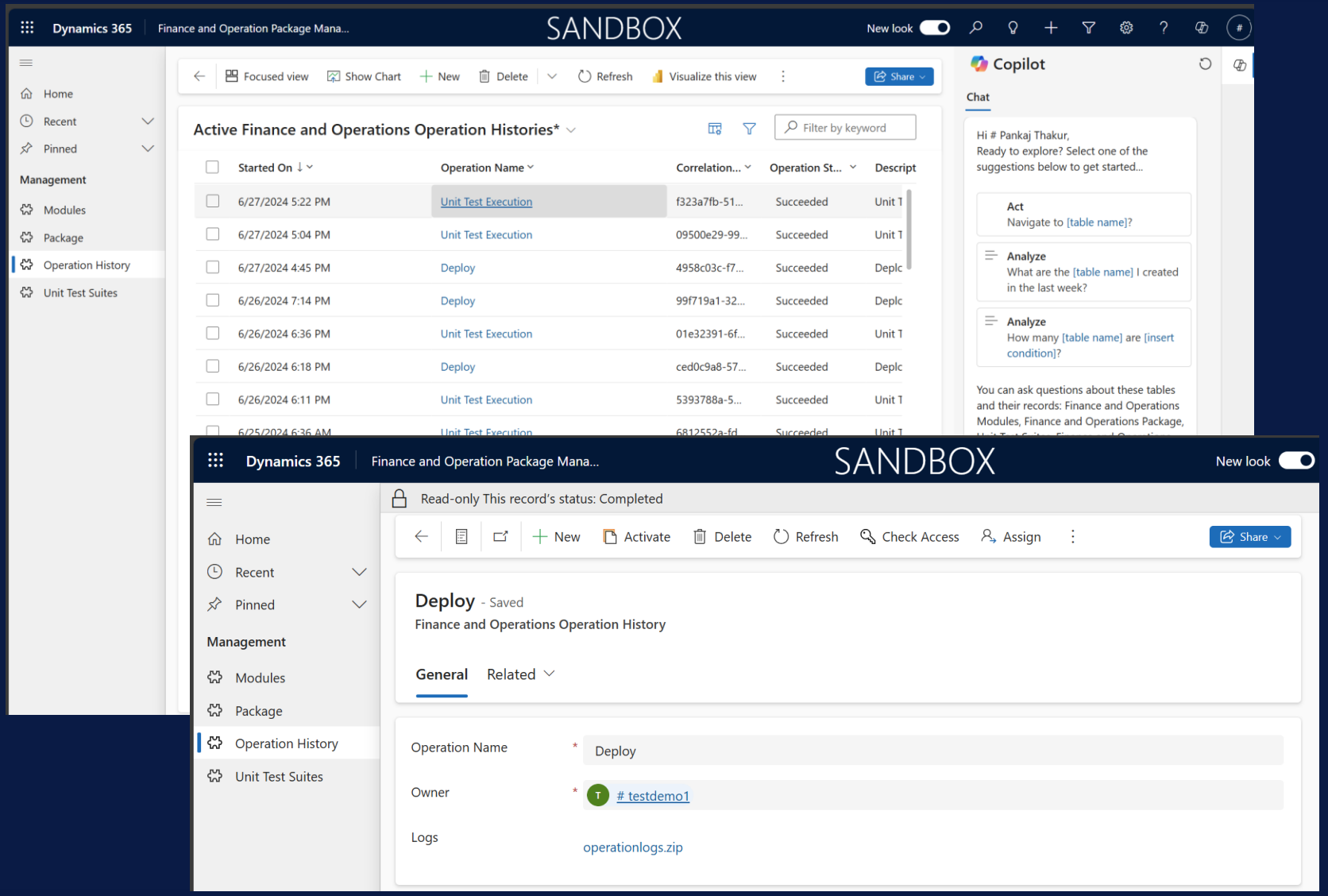

Azure DevOps ProgrammingModel / Helix / Pipelines / build-pipeline / 24.06.28.1
Search
PT
Helix
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Releases
Library
Task groups
Deployment groups
Test Plans
Artifacts
Compliance
Start Right
Project settings
Jobs in run #24.06.28.1
build-pipeline
Jobs
Job 12m 17s
Initialize job 13s
Checkout UDEDevALM@main to s 9s
Secure Supply Chain Analysis (aut... 3s
Initialize CodeQL (auto-injected) 6s
NuGet custom install Packages 4m 4s
Update Model Version 4s
Build solution ***.sln 4m 51s
Copy X++ Compile Log Files to: D... 3s
Publish Artifact: drop build output 28s
Publish Artifact: drop packages, ... <1s
Use NuGet 3.3.0 1s
Create Deployable Package 1m 26s
Add F&O ISV License to Unified P... 8s
Publish Artifact: dro
Component Detection (auto-inje... 22s
Finalize CodeQL (auto-injected) 4s
Post-job: Checkout UDEDevALM... <1s
Finalize Job <1s
Report build status <1s
Add F&O ISV License to Unified Package for ISV module aaUnifiedExperience
Job
View raw log
1 Pool: Azure Pipelines
2 Image: windows-latest
3 Agent: Hosted Agent
4 Started: Today at 6:14 PM
5 Duration: 12m 17s
7 Job preparation parameters
132 2 queue time variables used
136 Parent pipeline used these runtime parameters
141 3 artifacts produced
packaging with ISV License

Verify deployment history and logs

You can download the logs from your Dataverse organization:

- Login to the Dataverse organization
- Find **Finance and Operation Package Manager App** on the main page
- Select the app and then from left pane, select **Operation History**
- Open the respective record by selecting the **Operation Name** and download the operation logs (operationlogs.zip file)

- Note that when using Visual Studio , a link to download operation logs is available in the Visual Studio output pane.



Microsoft Dynamics 365

Resources

Priyanka Sinha

FAQs

- What is the recommendation to use Git or TFVC as version control?
  - Start using Git, it has become the preferred version control system in Azure Repos. While some still rely on TFVC and we don't intend to remove this feature set, we plan to phase out TFVC gradually for new projects and organizations, as well as for projects that currently don't use TFVC.

- How to migrate from TFVC to Git?
  - Refer to Import and migrate repositories from TFVC to Git - Azure Repos

- How can a fully deployable package (Lifecycle Services legacy package) be created?
  - Deployable package can be created from Azure DevOps pipelines in addition to the unified format.
  - Not possible from Visual Studio.

- How can a fully deployable package (Lifecycle Services legacy package) be converted into the new format to be compatible for deployment to environments?
  - Locate ModelUtil.exe inside the bin folder and run it from the command line to see usage.
  - Choose the option and provide the package zip and output location as parameters.

Frequently asked questions - Power Platform | Microsoft Learn

Resources

Viva Engage : Dynamics 365 and Power Platform Preview Programs : Private Preview: Online Development

Campaign: Fusion: ODOP - One Developer experience preview - Dynamics 365

Unified admin experience for finance and operations apps (preview) - Power Platform | Microsoft Learn

Unified developer experience for finance and operations apps (preview) - Power Platform | Microsoft Learn

Continuous integration and deployment

One Dynamics One Platform – TechTalk Series

What is Power Platform Tools for Visual Studio - Power Platform | Microsoft Learn

Frequently asked questions (preview) - Power Platform | Microsoft Learn

Resources

Azure Repos Git Documentation

Get started with Git and Visual Studio - Azure Repos

X++ in Git - Finance & Operations

Branching in Git

Git branching guidance - Azure Repos

Import and migrate repositories from TFVC to Git - Azure Repos

Git branch policies and settings - Azure Repos | Microsoft Learn

How Microsoft develops with DevOps - Azure DevOps

Summary and Key takeaways

- ✓ Significance of adopting modern Dev ALM practices
- ✓ Utilizing Git for version control
- ✓ Implementing effective branching strategies
- ✓ Understanding on Development and Release flow
- ✓ Git repository setup and working with git
- ✓ Embracing Continuous Integration and Deployment
- ✓ Creating classic and yaml pipelines for build, release and testing


QUESTIONS

Dankie Faleminderit **Shukran** Chnorakaloutioun Hvala Blagodaria

Děkuji **Tak** Dank u Tänan Kiitos **Merci** Danke Ευχαριστώ A dank

Mahalo ΠΤΙΛ. **Dhanyavād** Köszönöm Takk Terima kasih **Grazie** Grazzi

Thank you!

감사합니다 Paldies Choukrane Ačiū **Благодарам** ありがとうございました

谢谢 Баярлалаа **Dziękuję** Obrigado Mulţumesc **Спасибо** Ngiyabonga

Ďakujem Tack Nandri Kop khun **Teşekkür ederim** Дякую Хвала Diolch


Microsoft Dynamics 365