Calgary Power Platform User Group

Tuesday June 16, 2020

PBILab

POWER PlatformUG

IterationInsights

Power BI CI/CD with Azure DevOps

Luca Gualtieri, MCP
PBI Lab Inc
pbilab.com

PBILab POWER PlatformUG IterationInsights

Luca Gualtieri

CTO and Principal at PBI Lab Inc

- BI and Analytics consultant
- Power BI expert (MCP)
- Project management
- Co-founder of PBI Lab
- Passion for data

luca.gualtieri@pbilab.com
pbilab.com

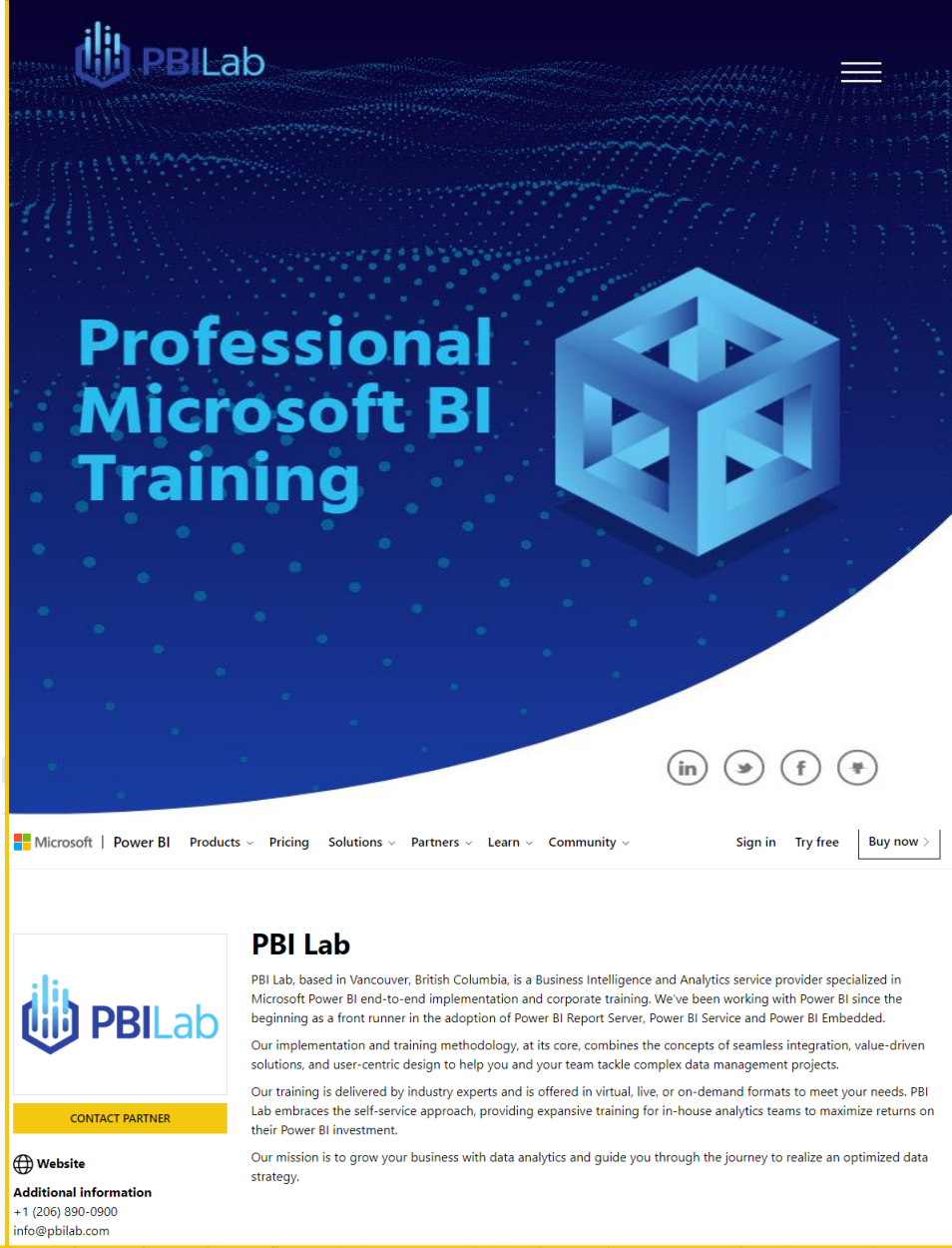
PBI Lab

BI and Analytics Consulting

- Microsoft Silver Partner in Data Analytics
- Microsoft Power BI Partner
- Microsoft Power BI Training Provider

pbilab.com

@pbilab



Session Objectives & Agenda

- New **Power BI Deployment Pipelines (preview)** feature
- **ALM** (Application Lifecycle Management): multi-tier developing process architecture for Power BI
- **Version Control** for Power BI with Visual Studio and Azure DevOps Git
- **Azure DevOps pipelines** for Power BI workspace creation and report deployment
- Summary & **QA**

Deployment Pipelines (preview)

New Deployment Pipeline in the Power BI Service

The tool is designed as a pipeline with three stages:

- Development
- Test
- Production

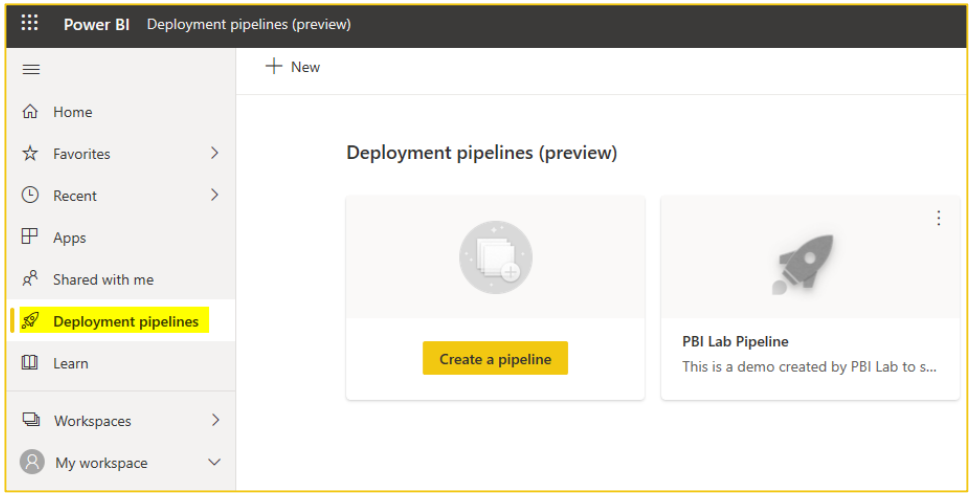
Requirements:

- You're a Power BI Pro user
- You belong to an organization that has Premium capacity
- You're an admin of a new workspace experience

New Deployment Pipeline in the Power BI Service

Functionalities:

- All content deployment
- Selective deployment
- Backwards deployment
- Dataset rules
- Comparing stages



Workspace assignment limitations:

- The workspace must be a new workspace experience
- You must be an admin of the workspace
- The workspace is not assigned to any other pipeline
- The workspace must reside on a premium capacity
- You cannot assign a workspace with Power BI samples to a pipeline stage

Deployment Pipelines (preview)
DEMO

Application Lifecycle Management

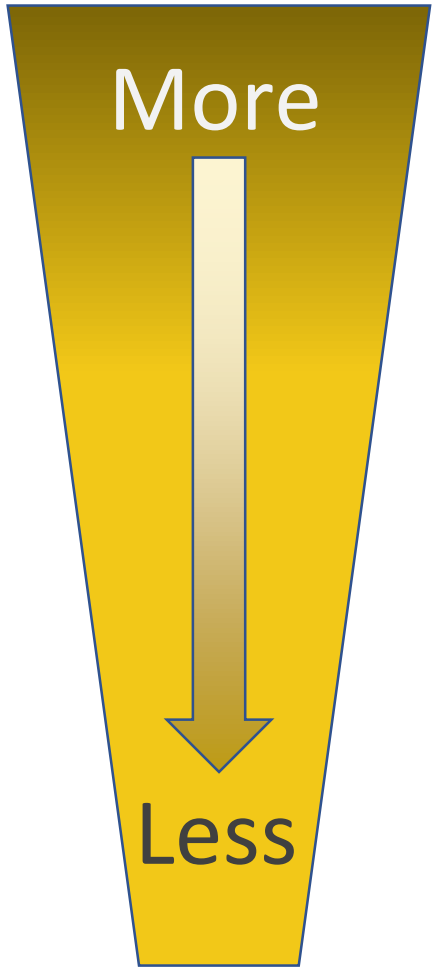
Multi-tier developing process architecture

- **DTAP** is an acronym for **Development**, **Testing**, **Acceptance**, and **Production**
- This **multi-tier** developing process is very common in IT
- Power BI, which is positioned as a **self-service data analytics tooling**, does not have a standard integrated capability to apply a multi-tier architecture

Multi-tier developing process architecture

- In some cases, the DTAP has **two additional stages** which are **Education** and **Backup**
- In this presentation we will only look at the most common four and we will **group Testing and Acceptance together**
- We will work with **three stages**: Development, Testing/Acceptance, and Production
- For simplicity we will call them **DEV**, **TST**, and **PRD**

Environments – Purpose



Performance & Stability

Production (PRD)

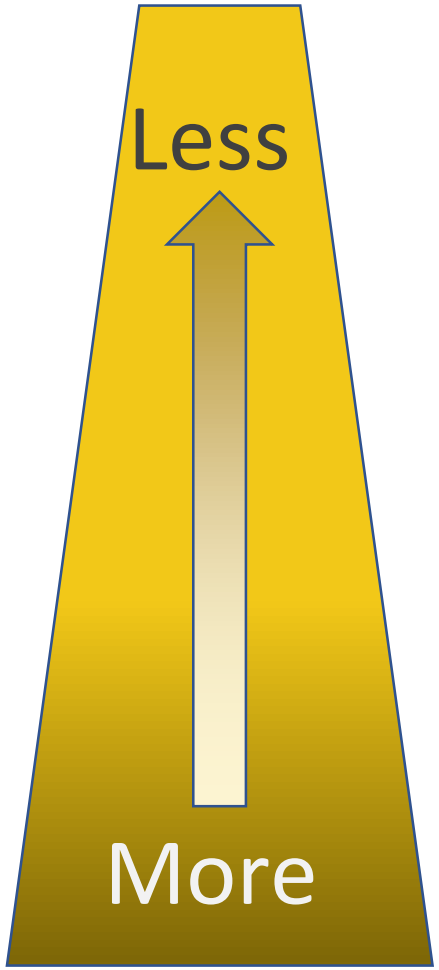
- Dedicated environment for consumers to access and engage with reports that are valid, secure and optimized

Testing/Acceptance (TST)

- Support & Developer collaboration on design, optimization and standards review
- Acceptance by the reviewers and business owners

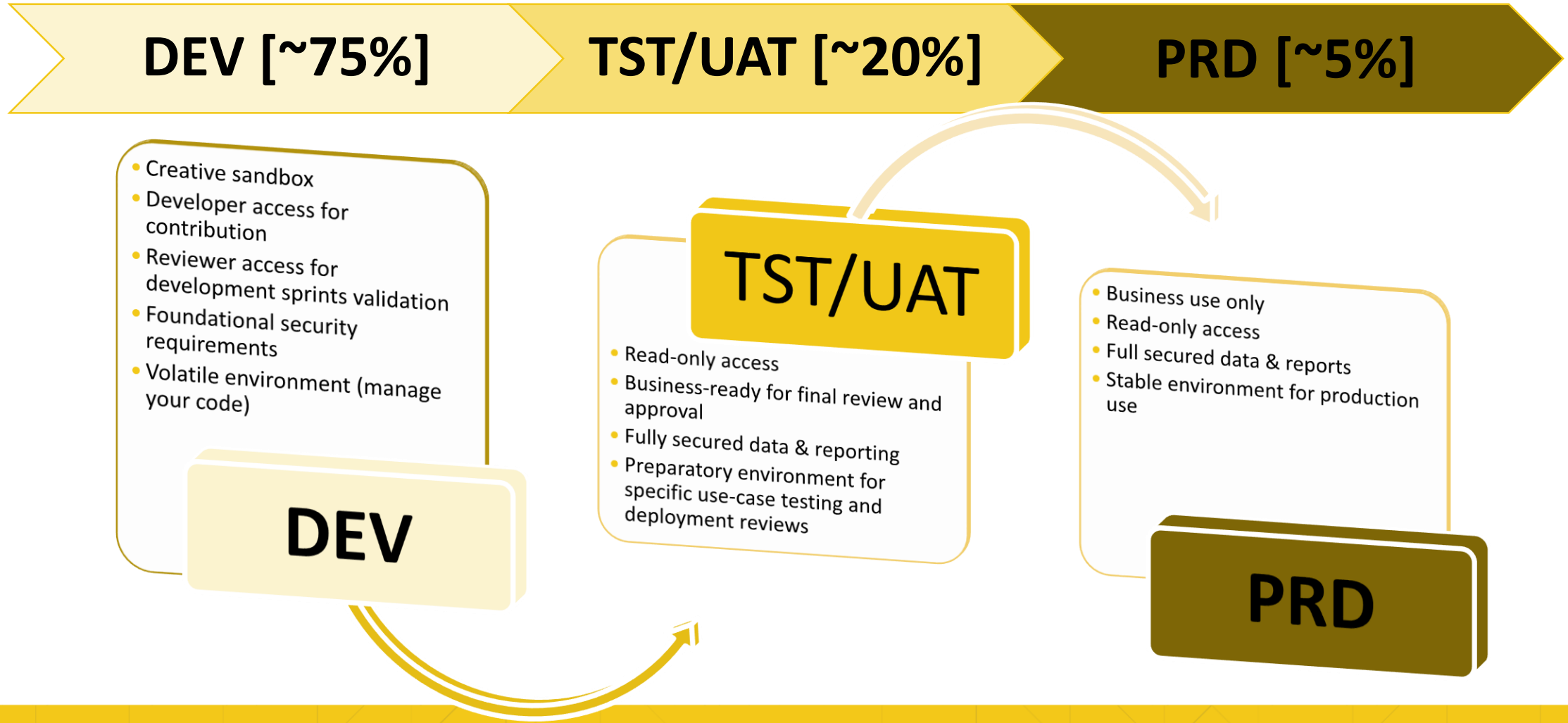
Development (DEV)

- Developer's sandbox to experiment and be creative with their ideas
- Ideal spot to start employing good design, standards and conventions
- Reviewer's working with developers to validate work



Flexibility & Adaptability

Environments – Function



DTAP: Development

- We use **Power BI Desktop** to create Power BI reports and data modeling in Power BI
- Power BI Desktop is our **development authoring** environment where we can do our data wrangling, transformations, modeling and of course create our report
- We can't collaborate and share our work with Power BI Desktop: we need to **publish our reports in Power BI Service**
- We need a development environment in Power BI Service
- **DEV workspace** available to **developers** with a "Contributor" role and to **reviewers** with a "Viewer" role

DTAP: Testing / Acceptance

- By creating a **TST workspace** and distributing the reports to a broader audience using a **TST App**
- Only **BI Administrators** can publish reports: they can deploy reports from a DEV workspace to a TST workspace using **automation** such as running a deployment pipeline in Azure DevOps
- Power Users and Reviewers can access the **TST App for review and acceptance**

DTAP: Production

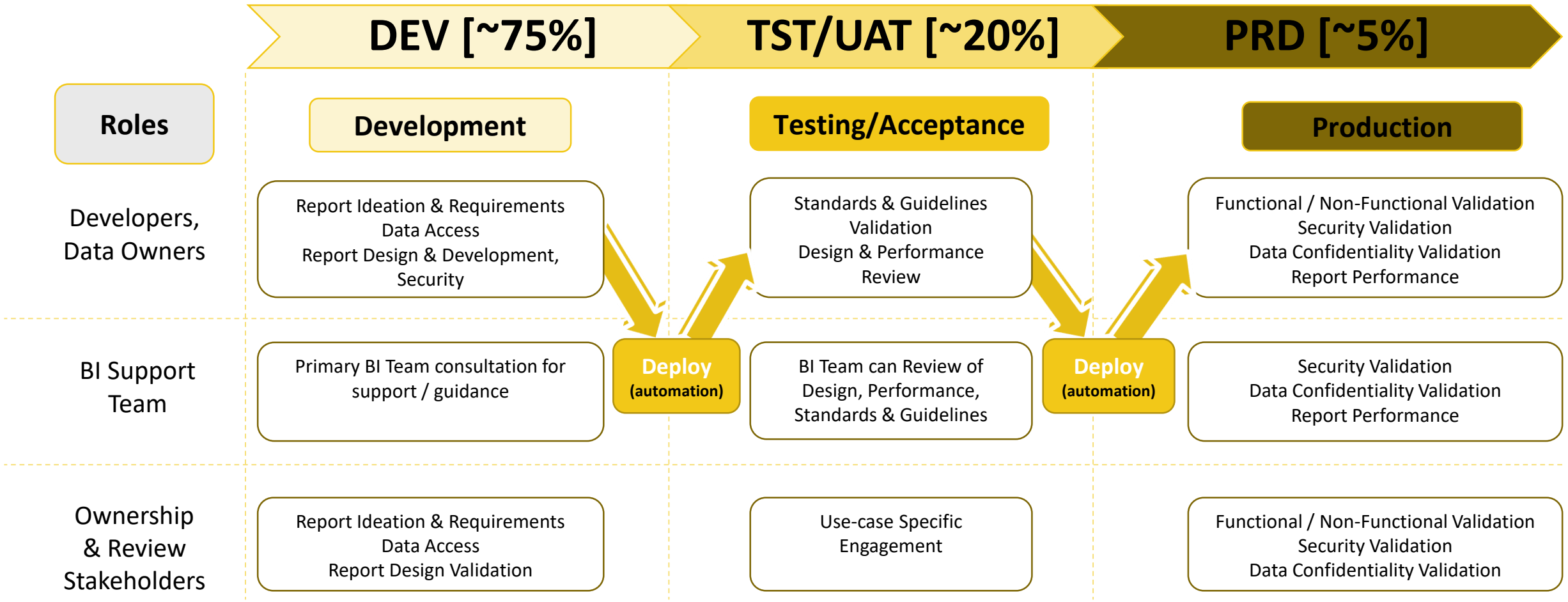
- After the testing is done and accepted by our business owners, we can push / deploy the content to a **production environment**, to be consumed by a broader audience

- We create a **PRD workspace** and distribute the reports to a broader audience on a **PRD App**

- We can do that by publishing the report to a **PRD App** and grant access to all end-users (“consumers”) with a “Viewer” role

- Only **BI Administrators** can publish reports: they can deploy reports from a DEV workspace to a TST workspace and finally to a PRD workspace using **automation** such as running a deployment pipeline in Azure DevOps

High-Level Deployment Guide



Summary of Environments

- Three environments with a combination of Workspaces and Apps

  - DEVELOPMENT ENVIRONMENT:
    - ✓ DEV WORKSPACE: available to Power Users, Reviewers and BI Admins
  - TEST ENVIRONMENT:
    - ✓ TST WORKSPACE: available to BI Admins
    - ✓ TST APP: available to Power Users, Reviewers and BI Admins
  - PRODUCTION ENVIRONMENT:
    - ✓ PRD WORKSPACE: available to BI Admins
    - ✓ PRD APP: available to Power Users, Consumers and BI Admins

Summary of Roles

- Four roles with different permissions by environment

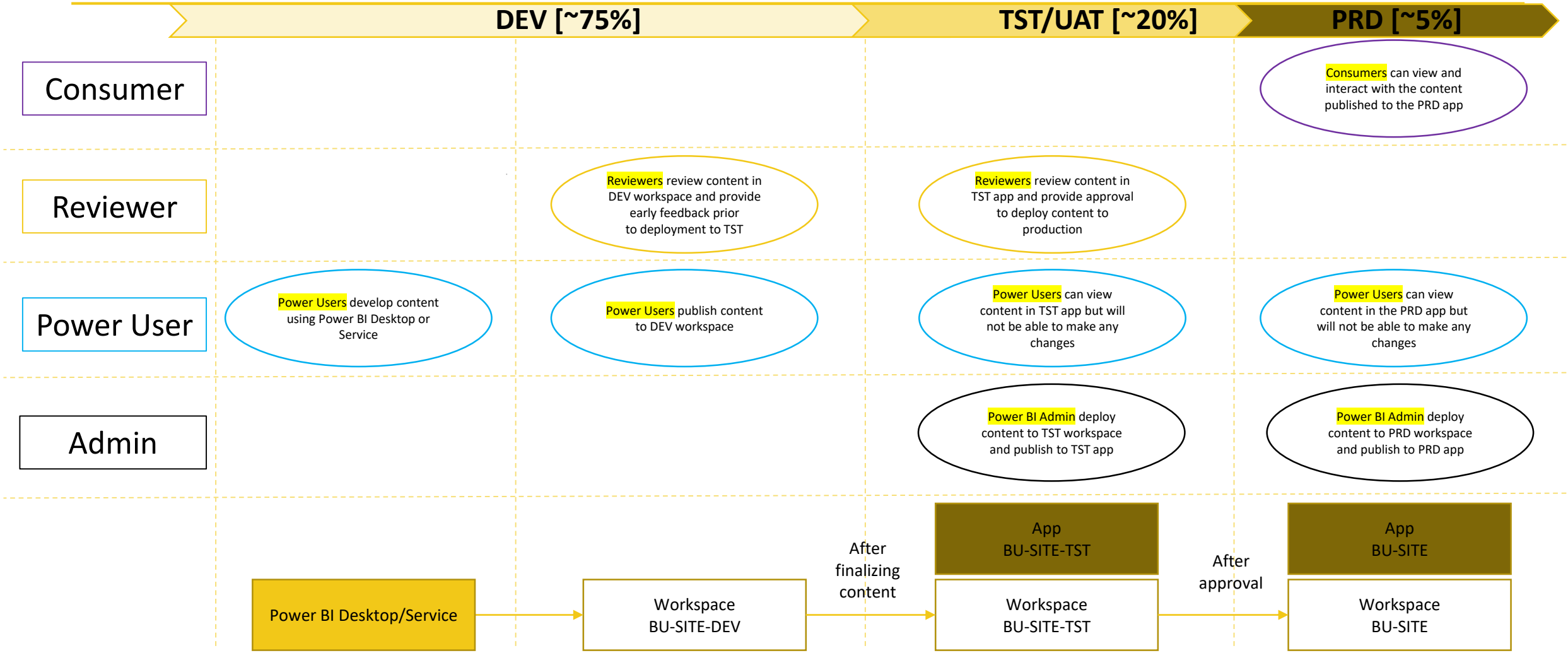
❑ Power Users:
- ✓ DEV WORKSPACE: contributors
- ✓ TST APP: viewers
- ✓ PRD APP: viewers

❑ Reviewers:
- ✓ DEV WORKSPACE: viewers
- ✓ TST APP: viewers

❑ Consumers:
- ✓ PRD APP: viewers

❑ BI Admin:
- ✓ admins in all the workspaces and apps

Workflow Overview

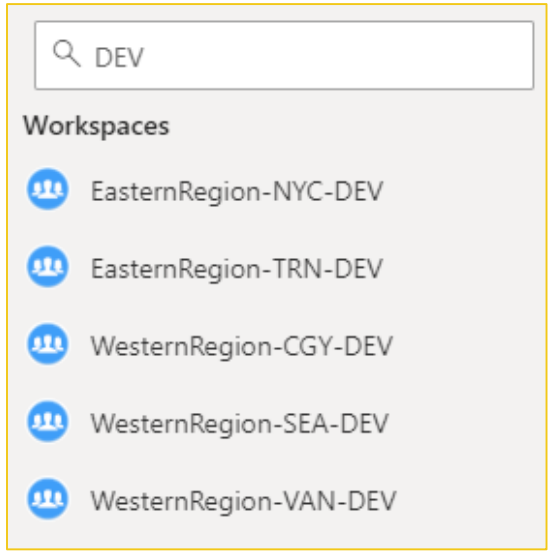


What is a DEV Workspace

Space for Development and Collaboration

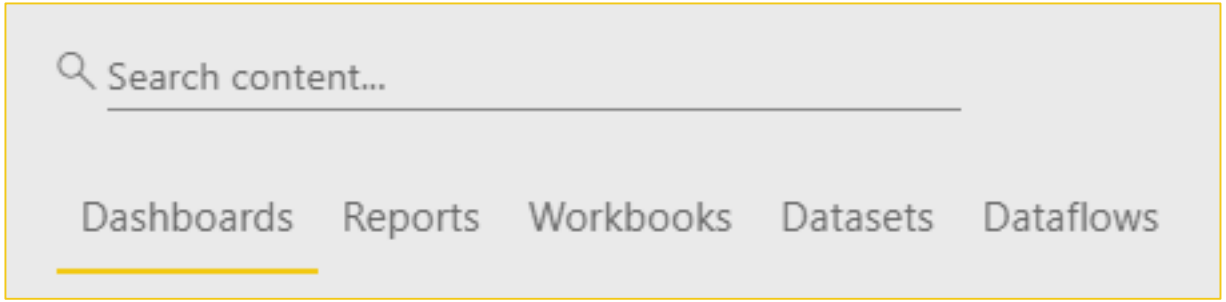
- In a **DEV** Power BI **workspace**, **Power Users** can **create** and **collaborate** on Power BI content with **Reviewers**
- A Power User, is able to contribute to his site-specific DEV workspace and in all the DEV workspaces where he has access as a Contributor (Power User)
- Power Users can submit a ticket/request to deploy the finished content the TST App for final review and UAT (User Acceptance Testing)
- Once the **content is approved in the TST App by Reviewers**, a new ticket/request is required to deploy the content to the official PRD App. The content will be available for the **Consumers** in the **PRD App**

What is a DEV Workspace



Workspaces are available in the Power BI Service

A Power User and/or Reviewer for a specific SITE or multiple SITES, can find the DEV workspaces available to them for contribution and collaboration in the Power BI Cloud service.

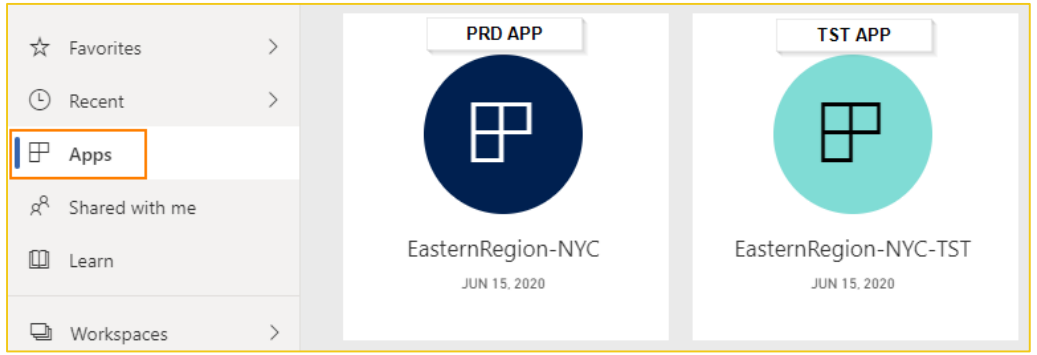


What are TST and PRD Apps

Spaces for Distribution, Acceptance and Consumption

- **App is for broader distribution** of content to Reviewers and Consumers
- Reviewers will review and interact with the content in TST App first to make sure everything is good for end users to consume, then they will approve to push the content to the **PRD App to be available to the Consumers**
- TST App is only for reviewing purpose. Reviewers will be able to view content in DEV workspaces in advance to provide early feedback before content is published to a TST App. All final reviews and approvals are done in the TST App
- The **PRD App is only for consumption purpose**. Consumers can view and interact but cannot change the content in the App

What is an App



Apps are available in the Power BI Service

As a Power User and Reviewer for a specific site or multiple sites, you will find the TEST apps available to you for reviewing and approval purpose in the Power BI Cloud service.

As a Power User and Consumer for a specific site or multiple sites, you will find the official (PROD) apps available to you for consumption purpose in the Power BI Cloud service.

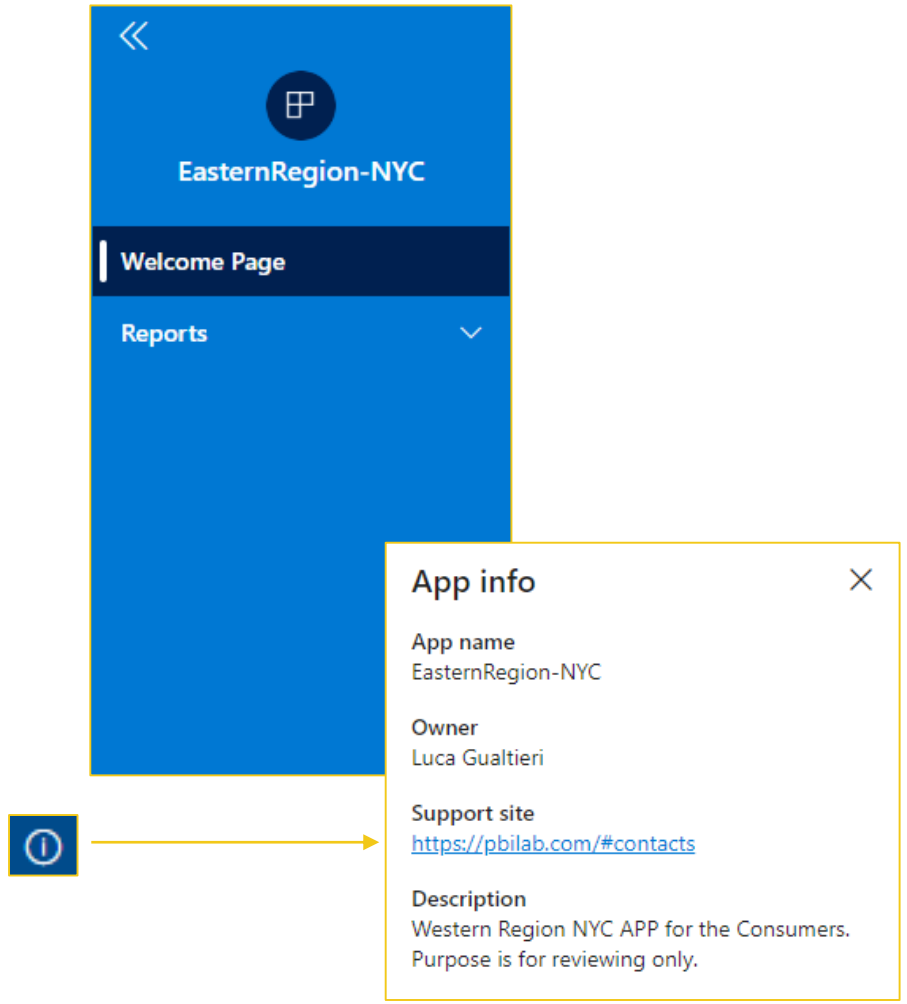
Customize an App

You can customized your App
- light blue app UI for the TST App
- dark blue app UI for the PRD App

Every App has a navigation pane on the left to enrich the user experience.

You can organize your App in sections:
- you can add a welcome page with valuable links
- you can organize your reports in sections

You can add your Support info



Workspace vs. App

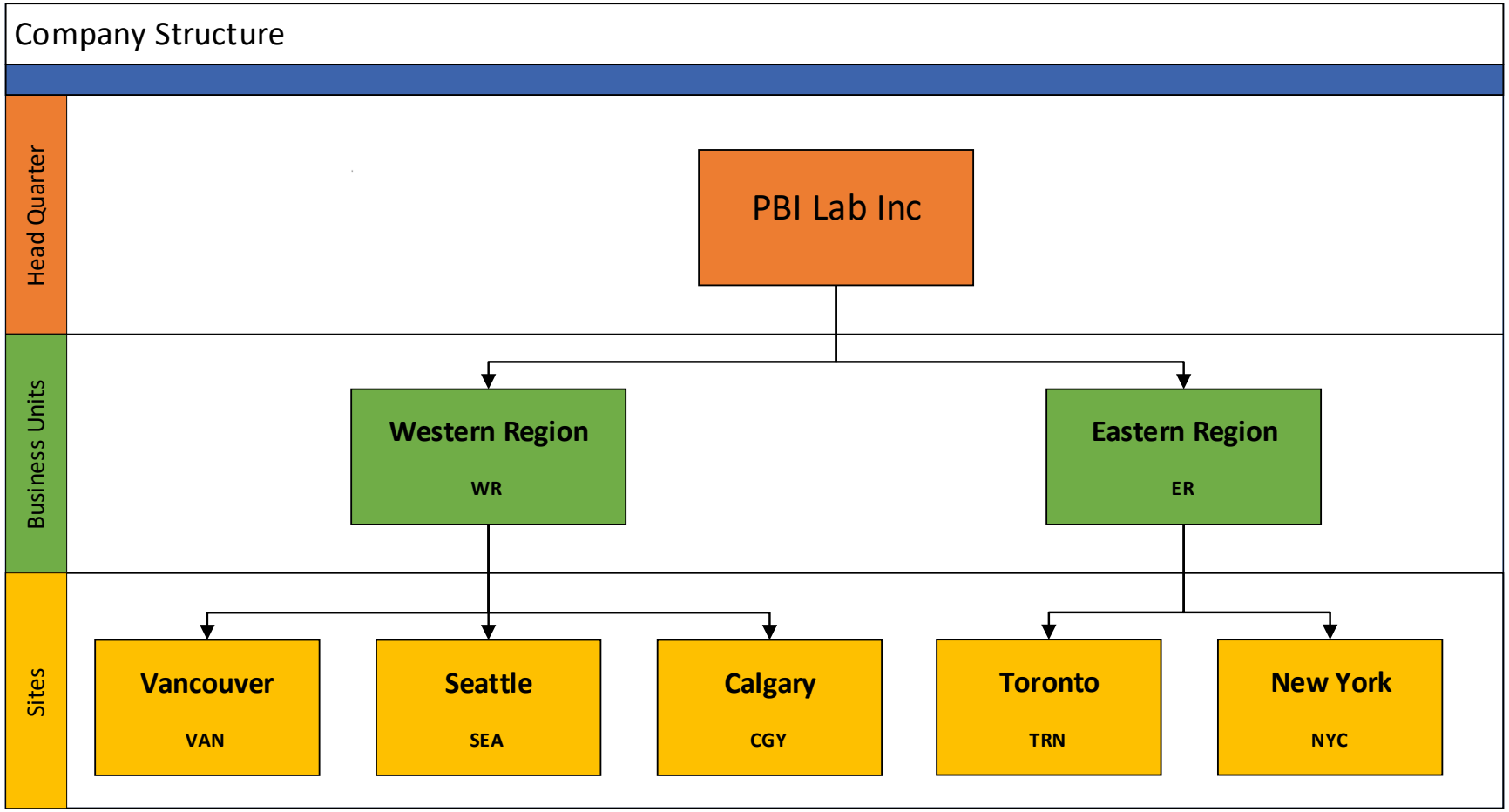
Workspace

- Create datasets, reports, dashboards
- In the workspace, Power Users (developers) can continue to develop content and decide which reports they want to be included in the App for Consumers
- Developers have access to the datasets and schedule refresh on the datasets
- Schedule refresh requires Gateway configuration

App

- For viewing and interacting with published content
- Consumers (users) can easily find their way around published content without getting confused by other work-in-progress items in the workspace
- Consumers can navigate through content more easily with the built-in navigation in the app
- Consumers get updates automatically in the app
- Consumers cannot modify content of the app
- Consumers do not have access to the underlying datasets

DEMO: PBI Lab fictitious organization



Application Lifecycle Management

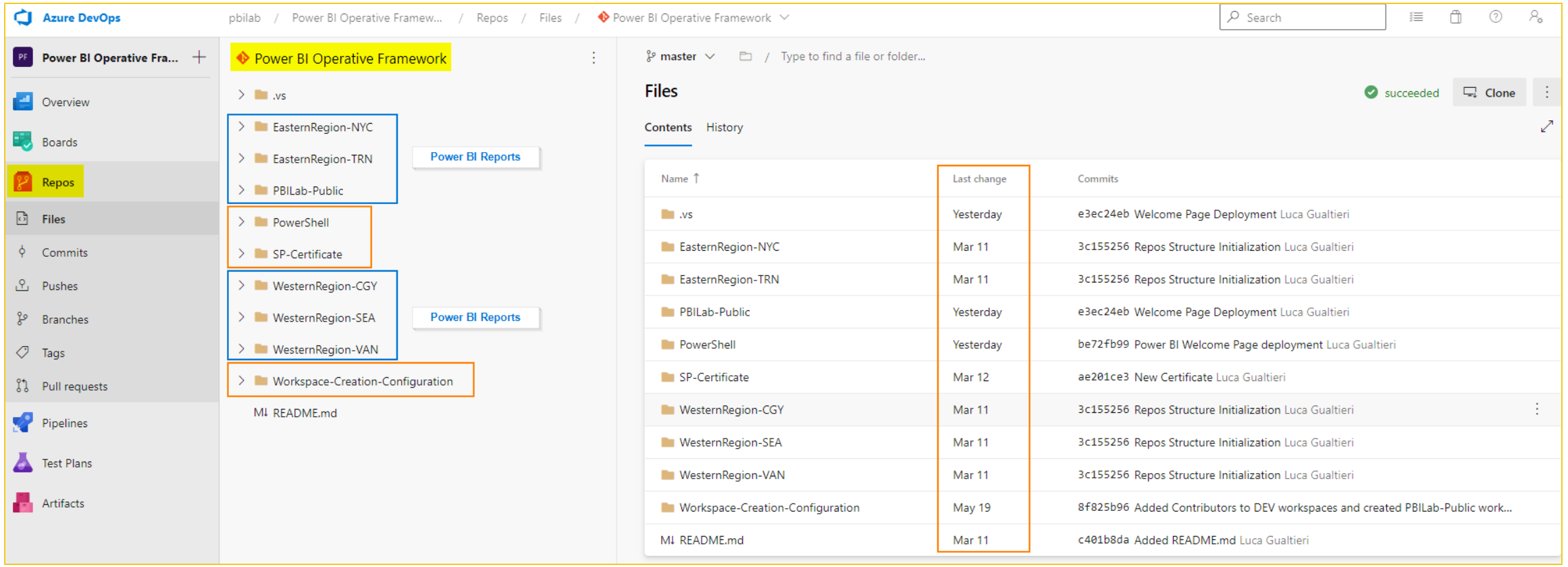
DEMO

Version Control with Azure Repo

Azure DevOps Repos

- **Azure Repos** is a set of version control tools that you can use to manage your code
- Use **version control** to save your work and coordinate code changes across your team
- Define **permission** on your Power BI Repos is important because Power BI reports in import mode might contain sensitive data
- Define **policies** to enforce who can create branches and the naming guidelines for the branches and automatically include the right reviewers for every code change

Azure DevOps Repos



Version Control with Azure Repo

DEMO

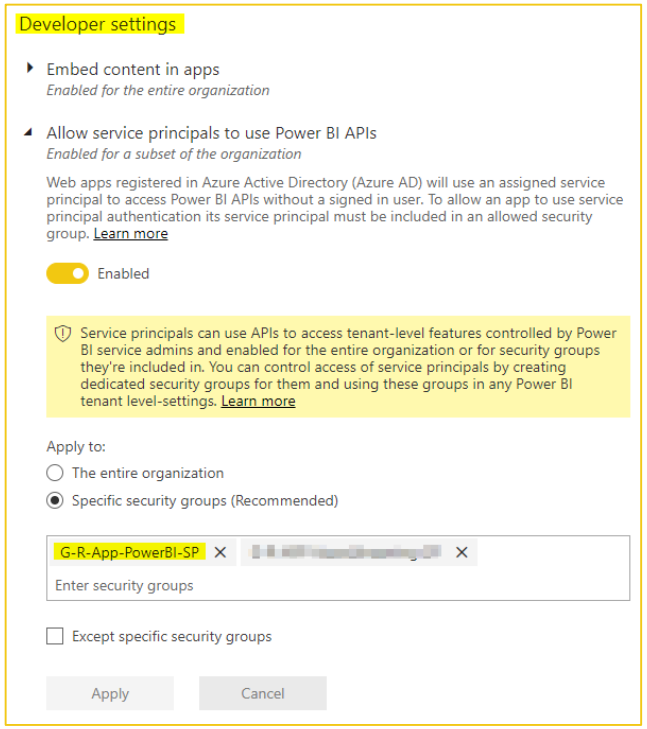
Azure DevOps Pipeline: Workspace Creation

Authentication for Power BI

- **Service principal** is a local representation of our AAD application for use in a specific tenant and will allow us to access resources or perform operations using Power BI API without the need for a user to sign in or have a Power BI Pro license

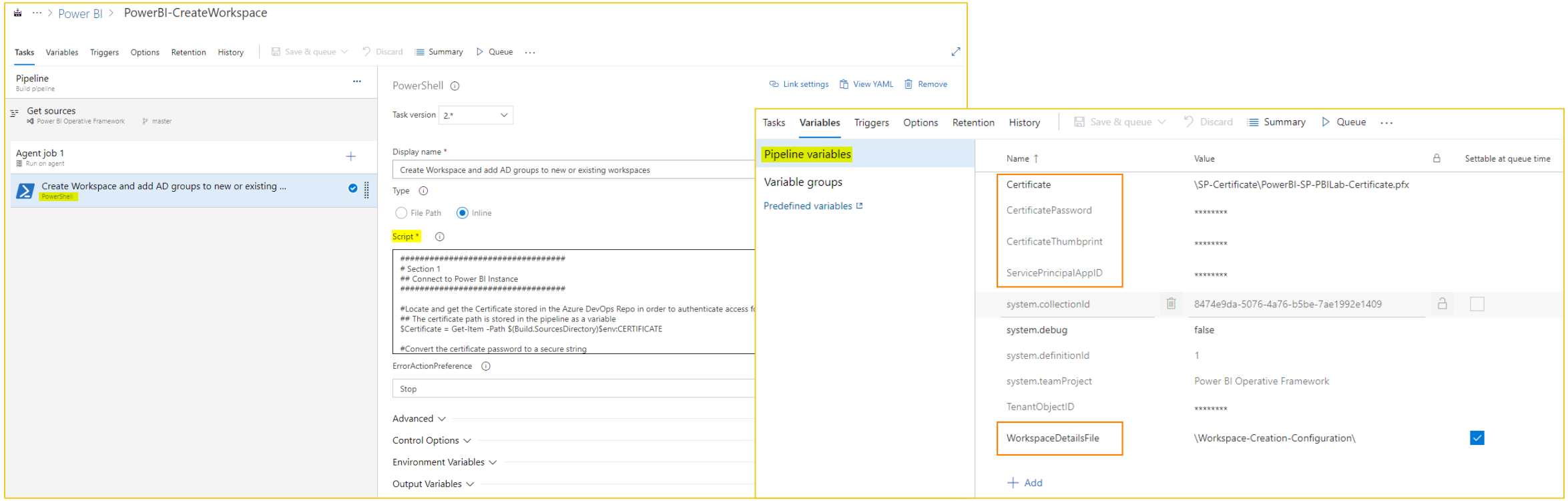
- Steps required:

  - ❑ Create an **Azure Active Directory (AD) App**
  - ❑ Assign the proper **API permissions**
  - ❑ Create an **Azure AD security group**
  - ❑ Enable the **Power BI service admin settings**



Azure DevOps Pipeline

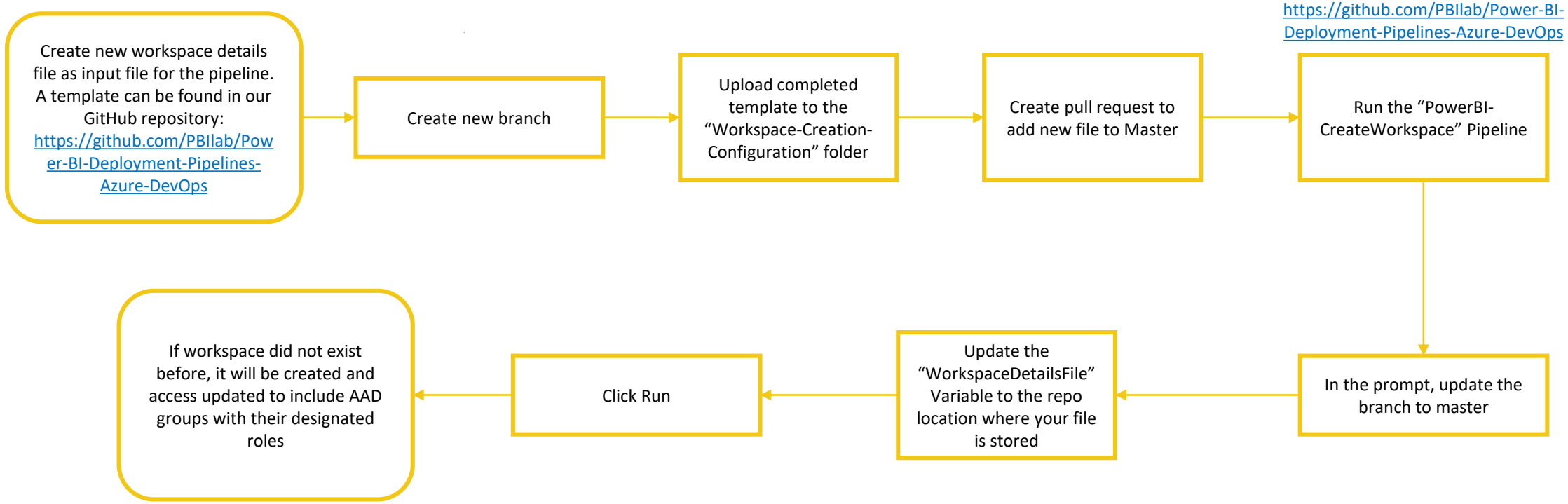
Azure DevOps Pipeline Name: **PowerBI-CreateWorkspace**



Workspace Creation and Security Configuration Flow

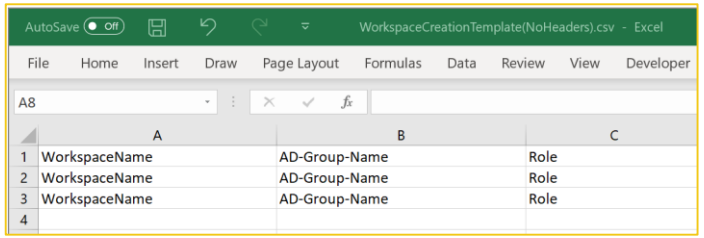
Azure DevOps Pipeline Name: **PowerBI-CreateWorkspace**

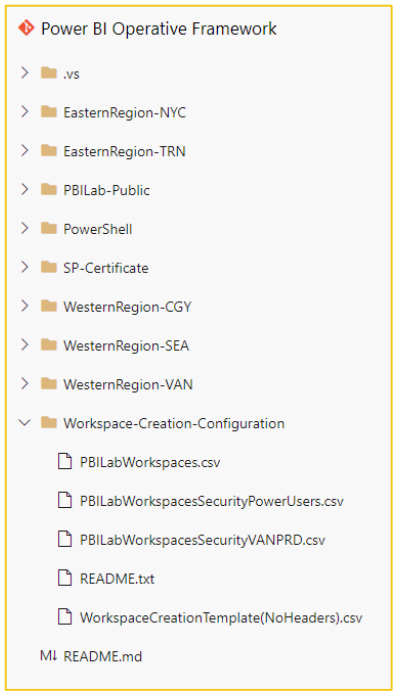
The PowerShell code for the pipeline can be found in our GitHub repository: https://github.com/PBIlab/Power-BI-Deployment-Pipelines-Azure-DevOps



Workspace Details Template

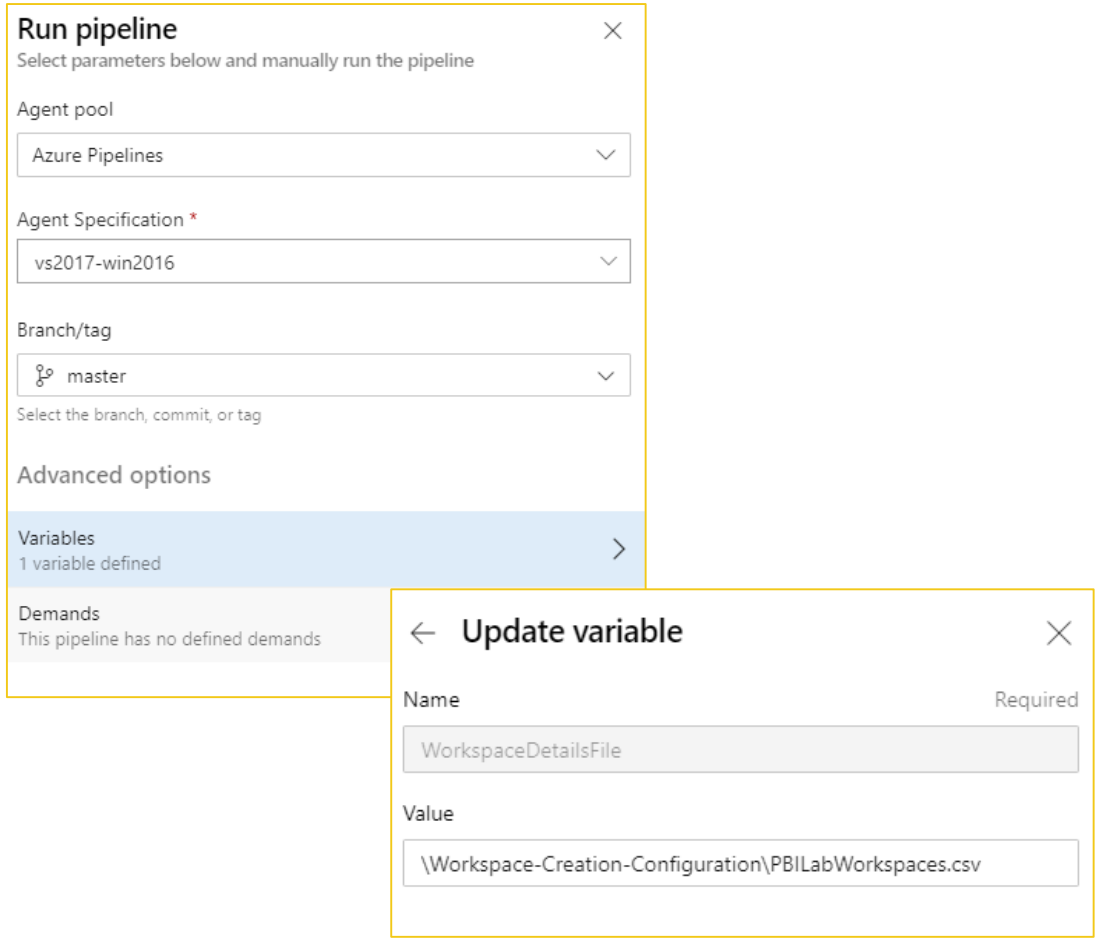
- Populate the template **WorkspaceCreationTemplate(NoHeaders)** with the information coming from the ticket / user request (if existing) and save as a csv; a template can be found in our GitHub repository: https://github.com/PBIlab/Power-BI-Deployment-Pipelines-Azure-DevOps
- Content should have no headers
  - 1st column = Workspace Name
  - 2nd column = Azure Active Directory (AAD) Group Name
    - ✓ Limited to AAD only; not include individuals' emails
  - 3rd column = Workspace Role
    - ✓ As of Jun 16th 2020, roles are limited to Admin, Member and Publisher only
    - ✓ As of Jun 16th 2020, AAD groups with Viewer role must be added manually
- Save the template file using the ticket number (if existing) as a reference
  - E.g. "TicketNumber.csv"
- If the workspace creation is not related to a ticket use a meaningful name
  - E.g. "PBILabWorkspaces.csv"
- Create a new local branch off master and Sync
- Upload the completed template to the "Workspace-Creation-Configuration" folder



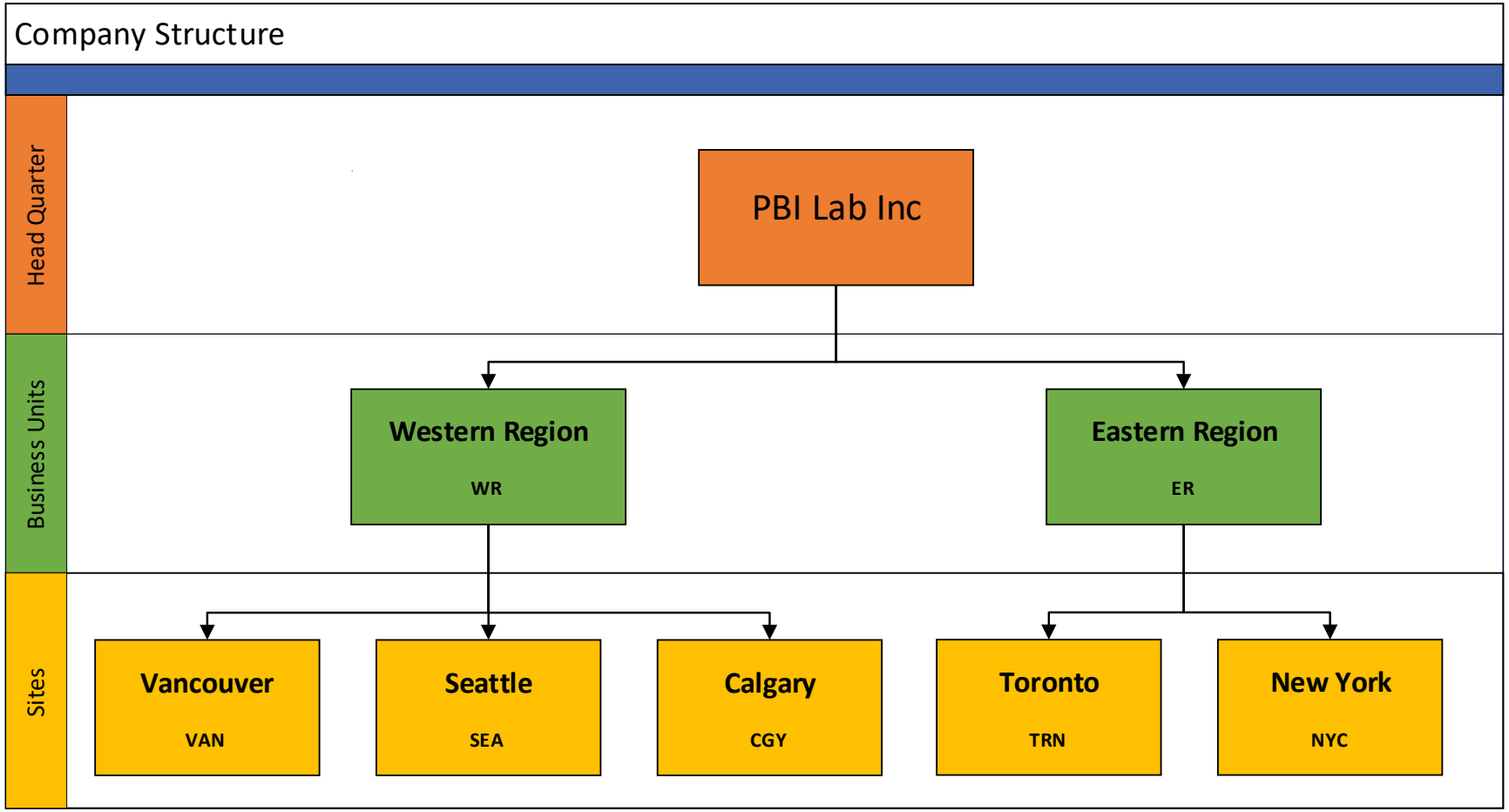


Execute Azure DevOps Pipeline

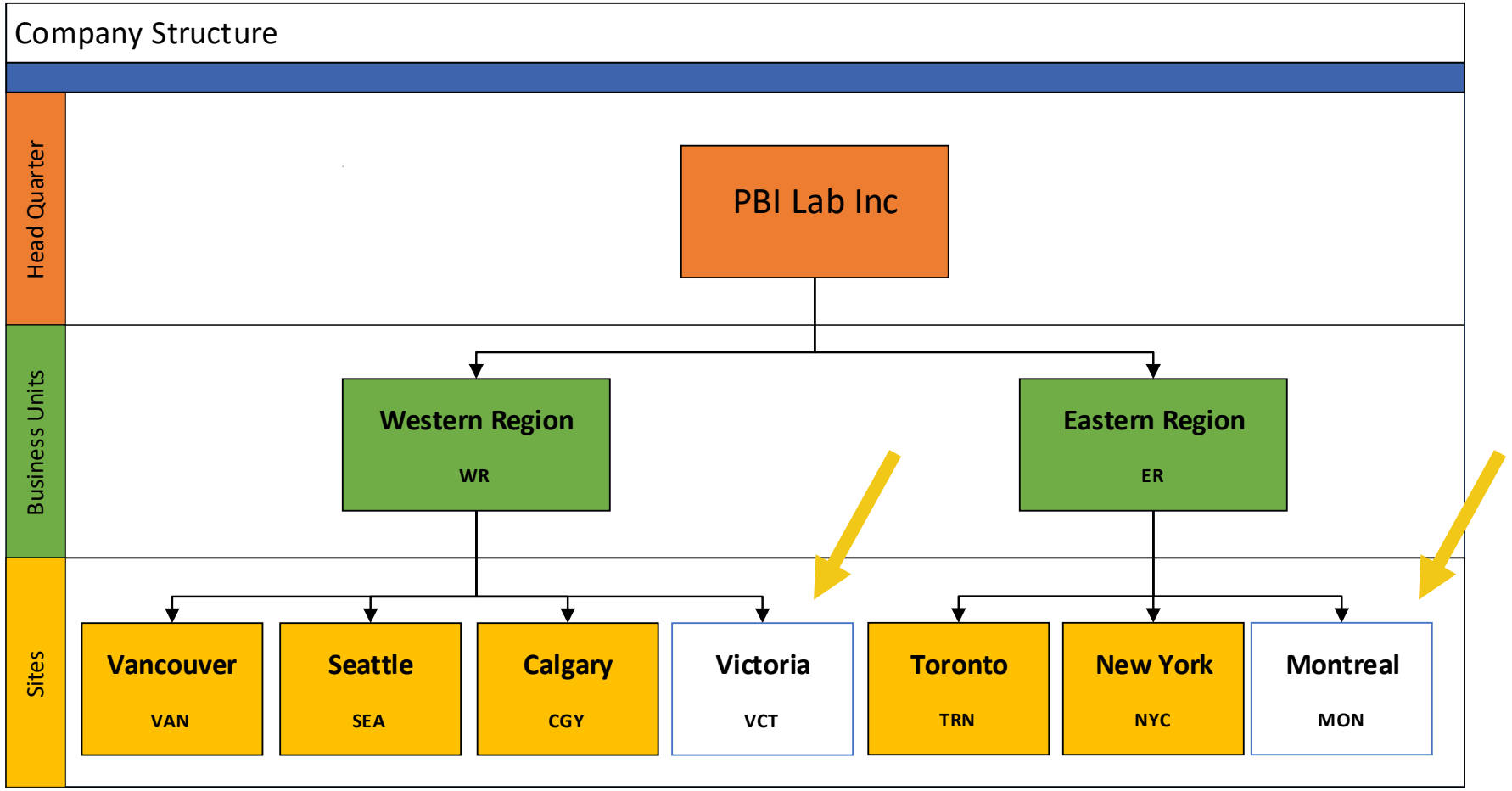
- Commit the template upload
- Push the local branch to remote
- Create a pull request to add the new file template to master
- Wait for the pull request to be reviewed and approved
- Run the **PowerBI-CreateWorkspace** Pipeline: the PowerShell code can be found in our GitHub repository: https://github.com/PBIlab/Power-BI-Deployment-Pipelines-Azure-DevOps
- Switch the Branch to master
- Update the “WorkspaceDetailsFile” Variable to the repo location where your file is stored
- Ensure that the location begins with \
- Ensure that the file name is included
- Click Run and wait for execution
- When completed check for errors and validate the existence of the workspaces in the service
- Add required Viewers manually



DEMO: PBI Lab fictitious organization



DEMO: PBI Lab Expansion



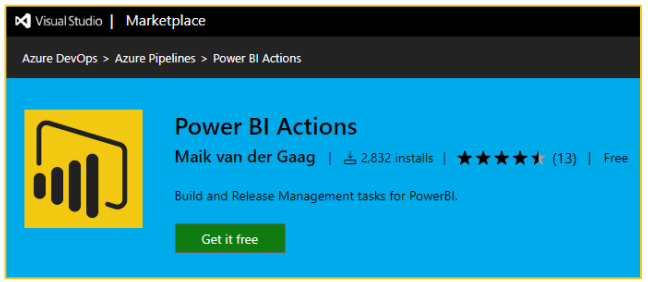
Azure DevOps Pipeline: Workspace Creation - DEMO

Azure DevOps Pipeline: Report Deployment

Authentication for Power BI and Azure DevOps plugin

- We will use the same **Service principal** created to run the workspace creation pipeline

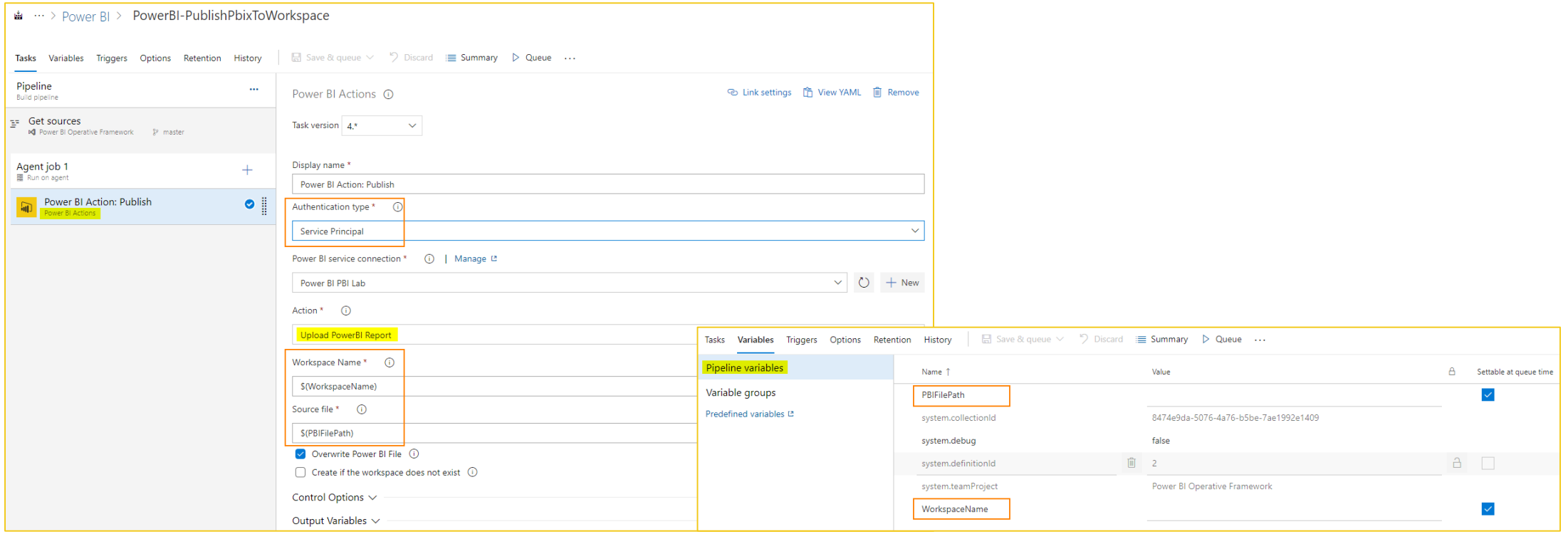
- For simplicity we will use a **plugin** called **Power BI Actions** that we can download from the Azure DevOps market place



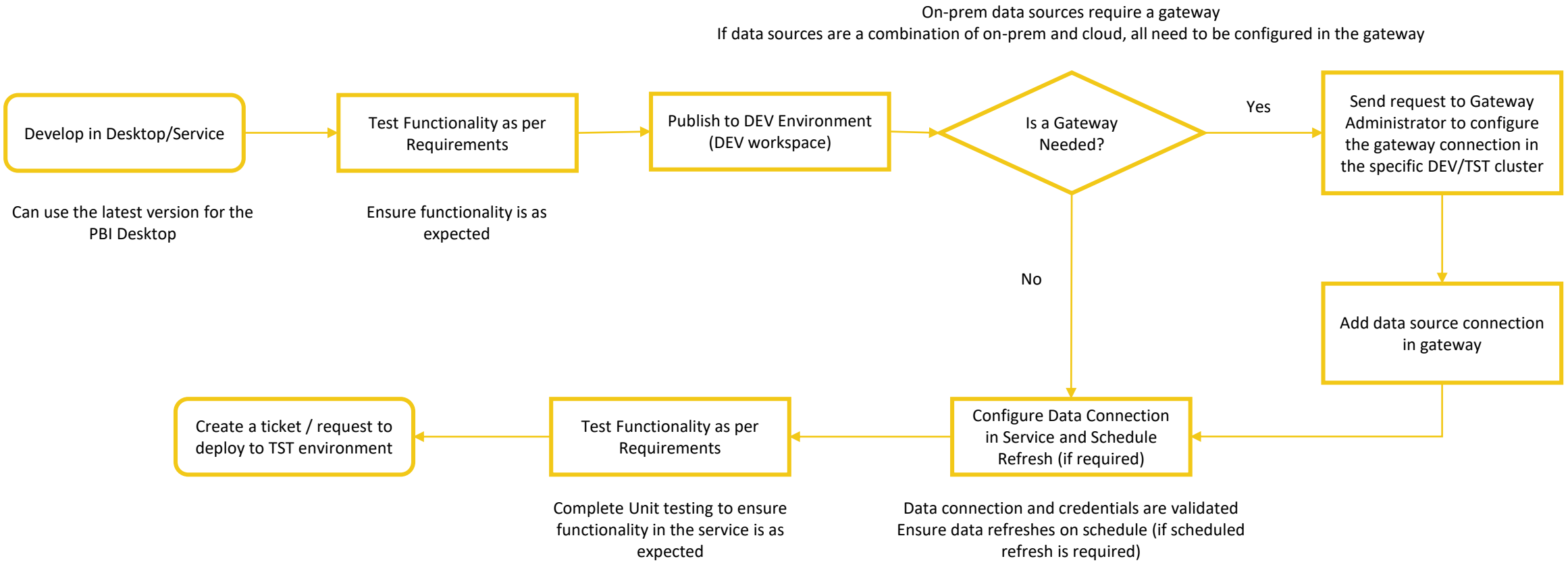
- The same results can be achieved by writing your own pipeline

Azure DevOps Pipeline

Azure DevOps Pipeline Name: PowerBI-PublishPbixToWorkspace

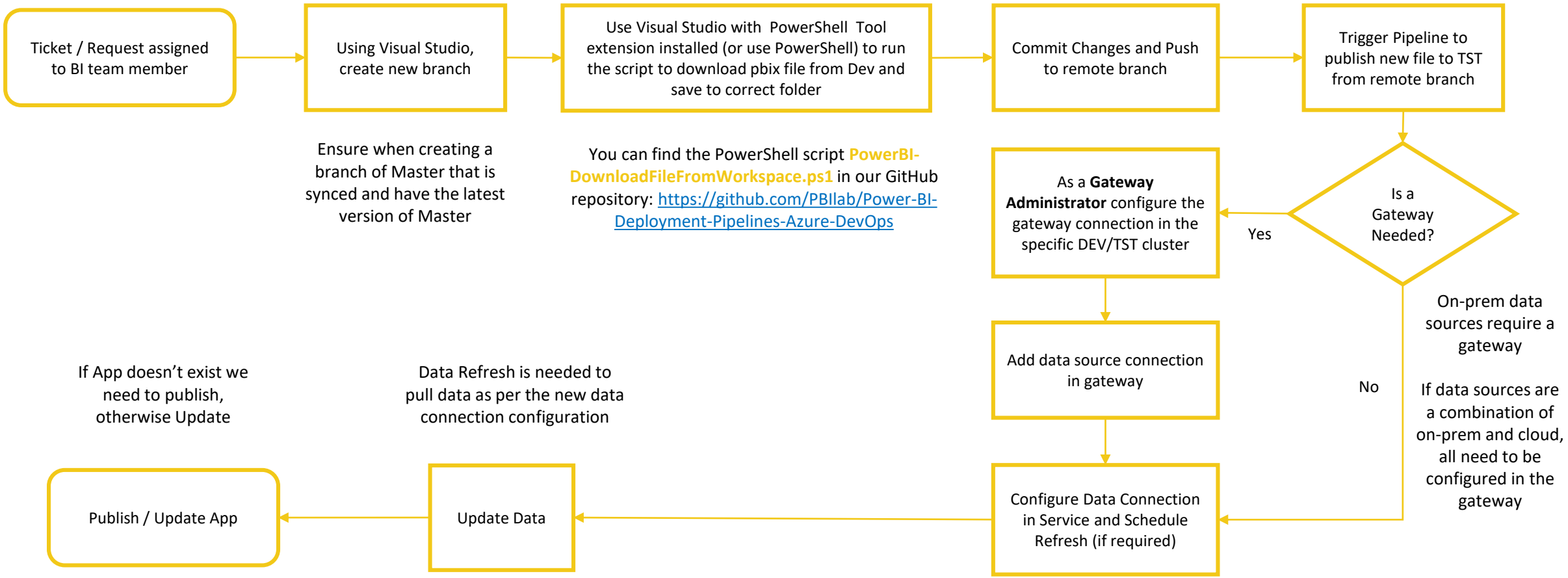


Development and Deployment to DEV Flow



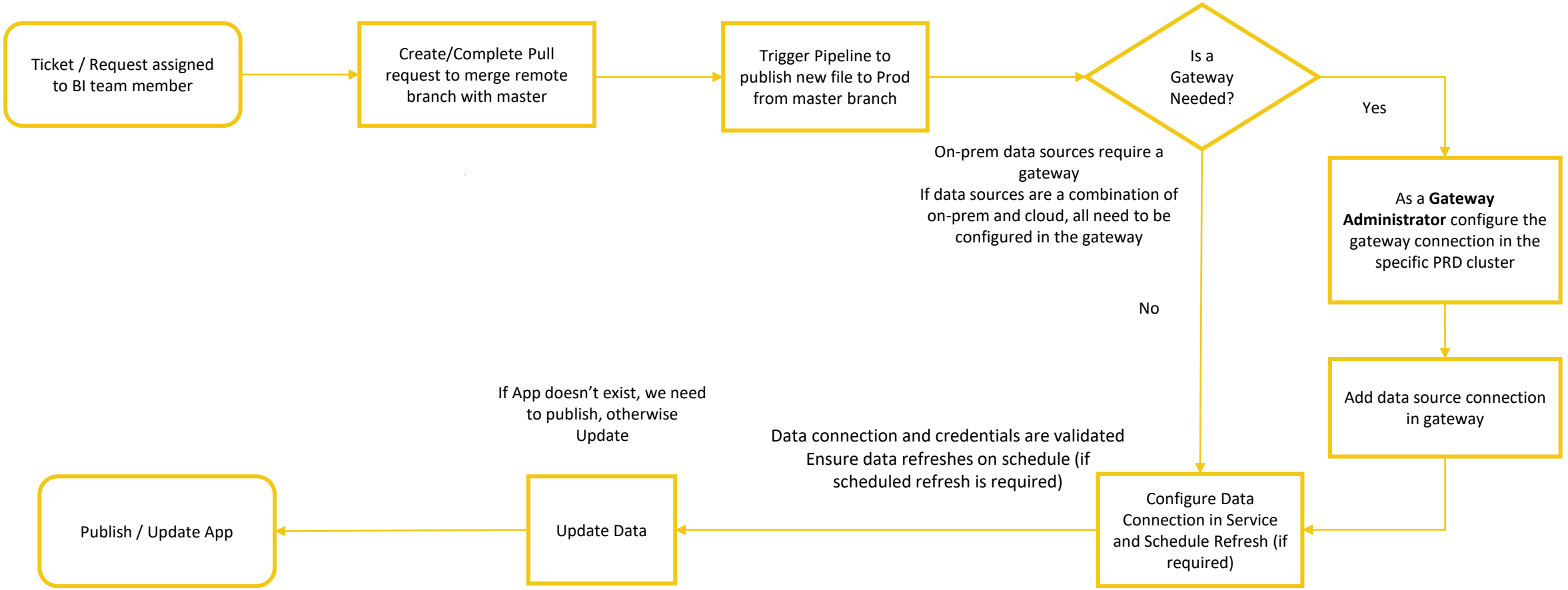
Audience: Power BI Power Users

Development to TST Flow



Audience: Power BI Support / Gateway Administrators

Development to PRD Flow



Audience: Power BI Support / Gateway Administrators

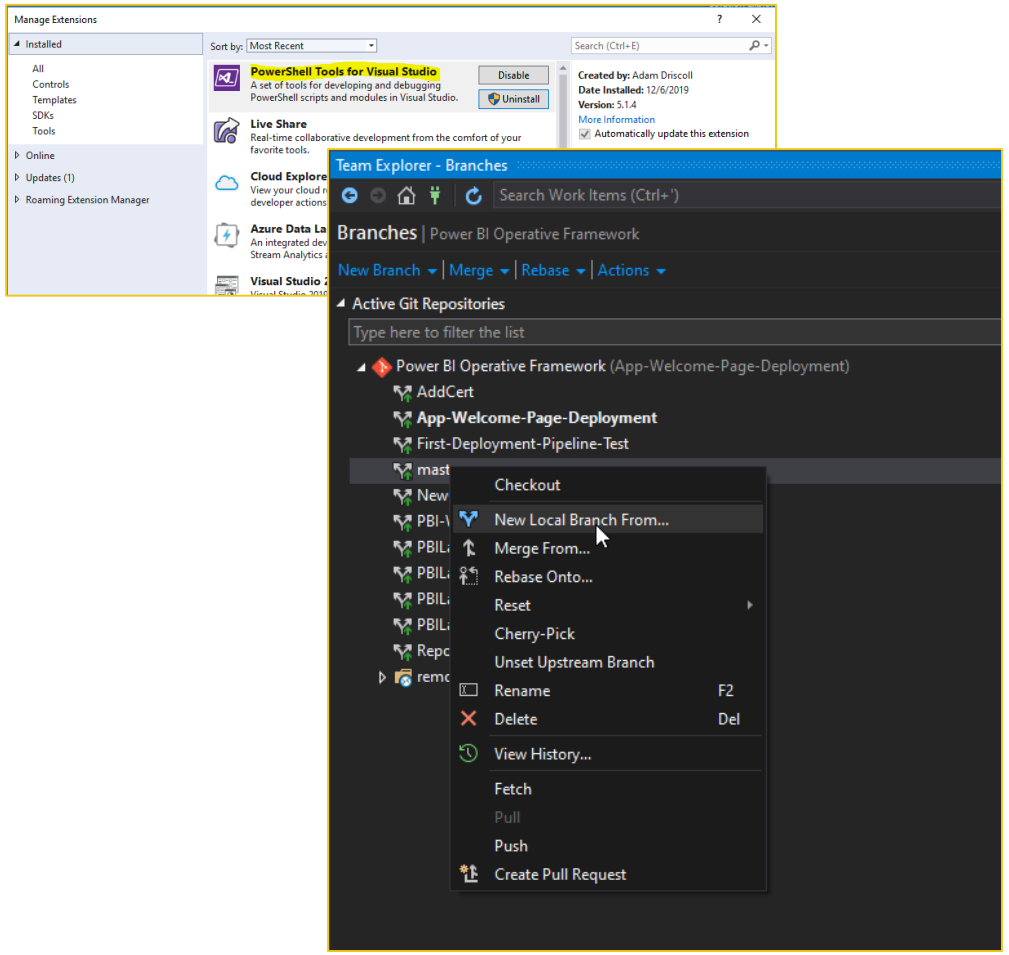
Create Local Branch

Prerequisite:

- PowerShell Tool extension must be installed in Visual Studio
  - *PowerShell outside Visual Studio can be used*
- .NET Framework above 4.7.0 (This only works from 4.7.1)
- Power Shell: MicrosoftPowerBIMgmt module installed
  - *Install-Module -Name MicrosoftPowerBIMgmt (Install as an Administrator)*

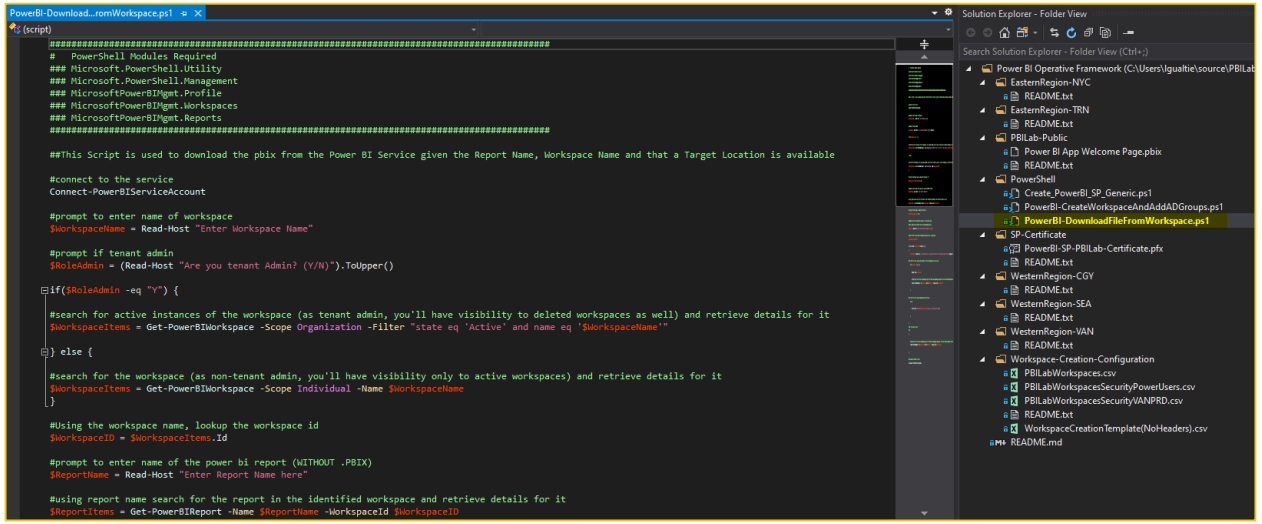
Using Visual Studio, create a local branch off master

- Use the ticket / request number in the name to associate the branch



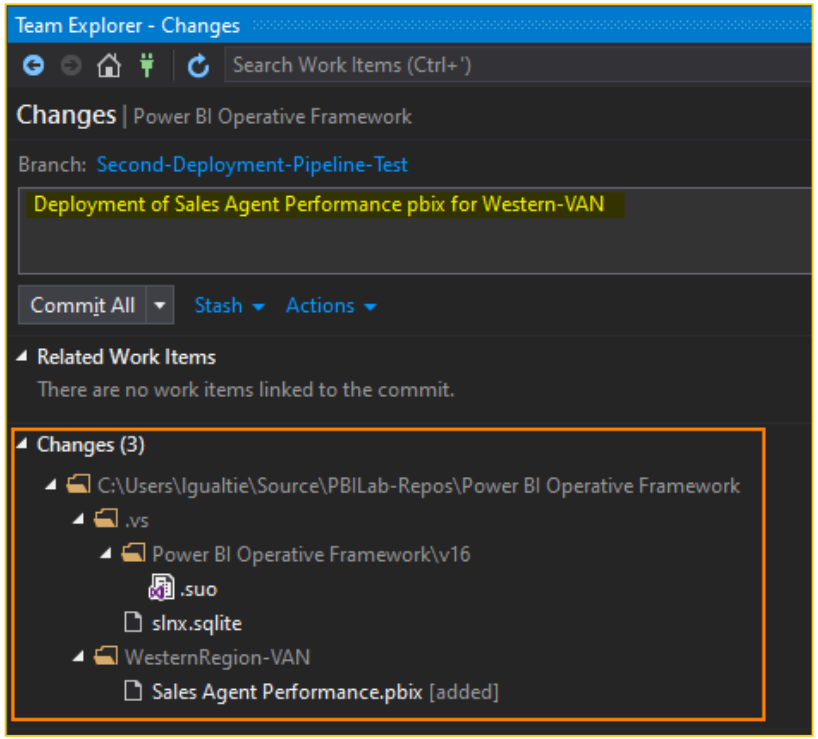
Download pbix to Local repo

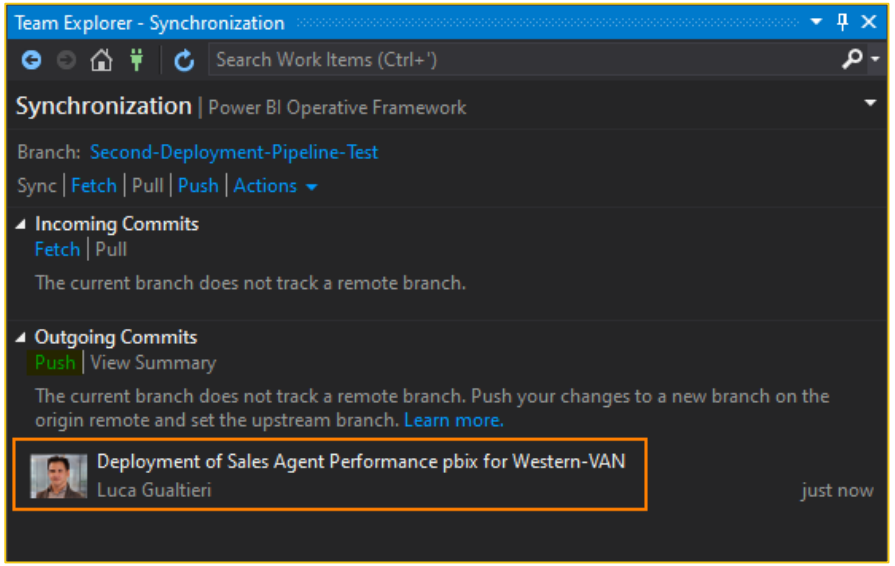
- Open the script **PowerBI-DownloadFileFromWorkspace.ps1** from the folder PowerShell
- Run in Visual Studio or PowerShell using your admin credentials
  - Only Tenant / Workspace Admins can run this
- Enter value in prompt for Workspace name where you want to download the report from
- Enter value in prompt for Report name you want to download (*)
- Enter value in prompt for location and filename that you'd like to save to (*)
- Downloaded file should be visible in the correct folder



Commit and Sync changes

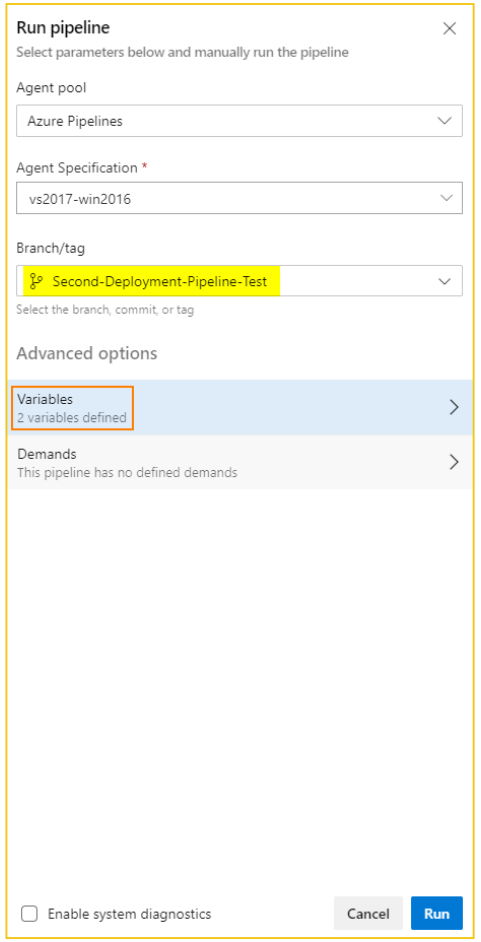
- After you've downloaded the pbix file in the local branch, **Repo the changes must be pushed** to the remote branch

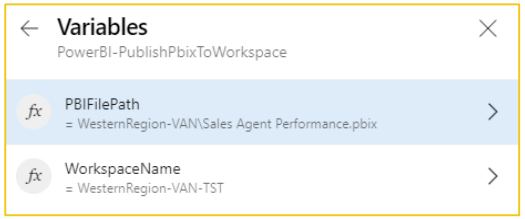




Run Pipeline to Publish to TST

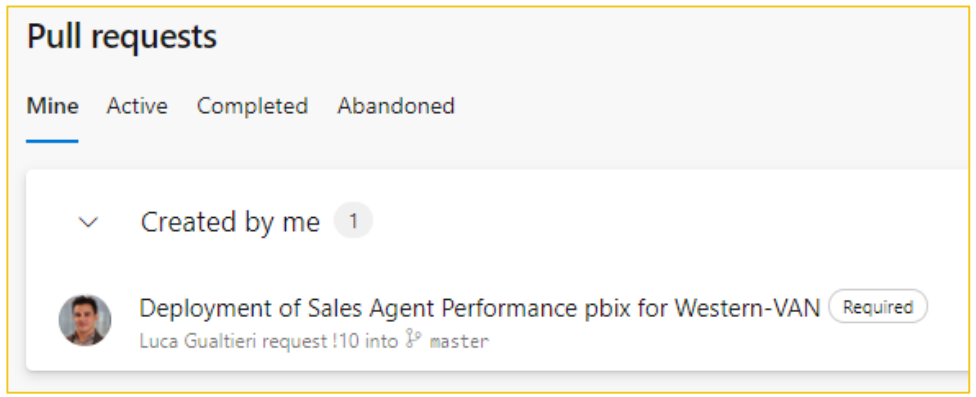
- Run the **PowerBI-PublishPbixToWorkspace** pipeline
- Update the branch/tag field to the remote branch you were working on
- Update the “PBIFilePath” variable to the file path where the pbix you wish to upload is stored
- Update the “WorkspaceName” variable to the workspace name where you want the file to be publish in power bi service
- Click Run and wait for execution

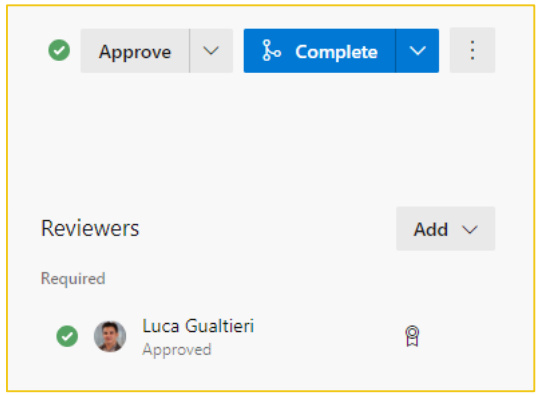


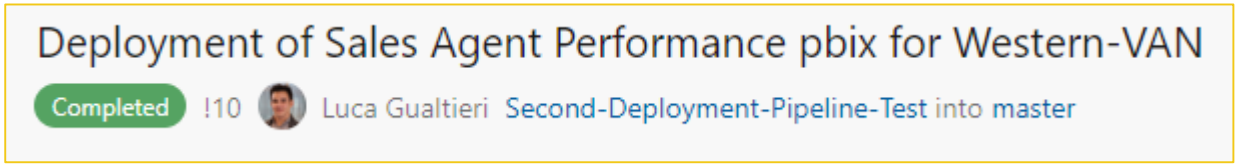


Merge with Master

- Create **Pull Request** to merge with master
  - You can create a Pull Request from the web page in Azure DevOps
  - You can create a Pull Request directly from Visual Studio and be redirected to Azure DevOps

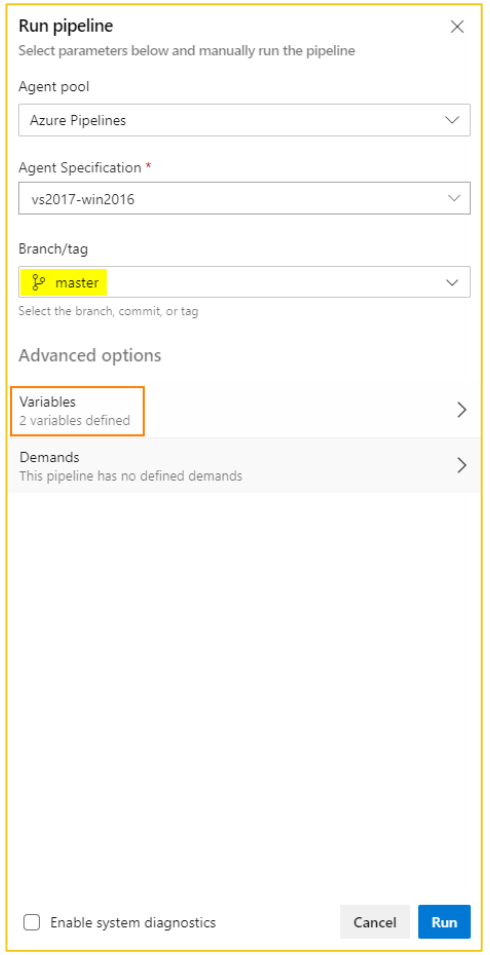


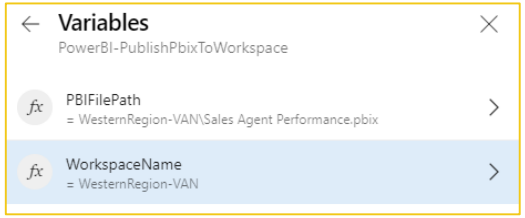




Run Pipeline to Publish to PROD

- Once the pull request has been approved and completed, Run the “PowerBI-PublishPbixToWorkspace” pipeline
- Update the branch to the master branch
- Update the “PBIFilePath” variable to the file path where the pbix you wish to upload is stored (*)
- Update the “WorkspaceName” variable to the workspace name where you want the file to be publish in power bi service (*)
- Click Run and wait for execution





Azure DevOps Pipeline: Report Deployment - DEMO

Resources

Connect with PBI Lab

pbilab.com

@pbilab

info@pbilab.com

luca.gualtieri@pbilab.com

Resources (1)

- PBI Lab GitHub Repository
https://github.com/PBIlab/Power-BI-Deployment-Pipelines-Azure-DevOps

- New deployment pipeline feature (preview) from Microsoft
https://docs.microsoft.com/en-us/power-bi/create-reports/deployment-pipelines-overview

- Create a repo in Azure DevOps from Visual Studio
https://docs.microsoft.com/en-us/azure/devops/repos/git/creatingrepo?view=azure-devops&tabs=visual-studio

- Service Principal
https://docs.microsoft.com/en-us/azure/active-directory/develop/app-objects-and-service-principals

Resources (2)

- Power BI PowerShell Reference
https://docs.microsoft.com/en-us/powershell/module/microsoftpowerbimgmt.admin/?view=powerbi-ps

- Azure DevOps Repos
https://azure.microsoft.com/en-ca/services/devops/repos/

- Azure DevOps Repos Permissions
https://docs.microsoft.com/en-us/azure/devops/organizations/security/permissions?view=azure-devops&tabs=preview-page

Resources (3)

- Azure DevOps branch policies
https://docs.microsoft.com/en-us/azure/devops/repos/git/branch-policies-overview?view=azure-devops

- Power BI Actions plugin for Azure DevOps
https://marketplace.visualstudio.com/items?itemName=maikvandergaag.maikvandergaag-power-bi-actions

- Power BI Action plugin wiki
https://github.com/MaikvanderGaag/msft-extensions/wiki

Q&A