Intro to Automated DB Deployments with GitHub

Erin Dempster (she/her)

Principal Cloud Engineer

Trean Corporation

#SQLSatJax

Erin Dempster

Microsoft® MVP Most Valuable Professional

She/Her

Principal Cloud Engineer

Trean Corporation

- SQL Server DBA
- Azure Administrator
- Azure DevOps Administrator

https://www.erindempster.com

@em_dempster

https://www.linkedin.com/in/erindempster

Speaker – PASS Summit, SQLBits +

Author – SQLServerCentral.com

SQLSATURDAY

#1068 - Jacksonville, FL - May 4th, 2024

Link to Today's Content

https://github.com/endlessautomation/SQLSat

Who’s in the Audience?

- DevOps Engineers?
- Database Developers?
- Database Administrators?
- IT Managers?

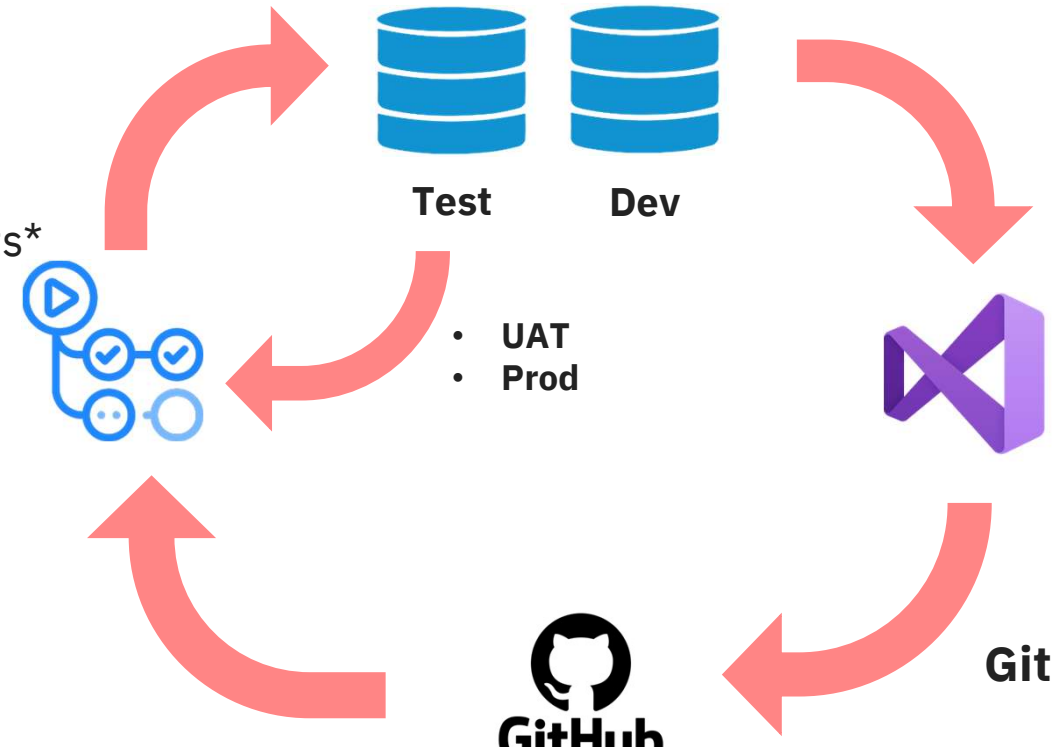
Who am I Expecting Today?

- Everyone, of course, especially
- Application Developers working with Databases
- Database Developers
- Have **some familiarity** with source control

Agenda

GitHub Actions

1. Build
2. Automated Tests*
3. Approve
4. Deploy
5. Repeat 3&4



Visual Studio

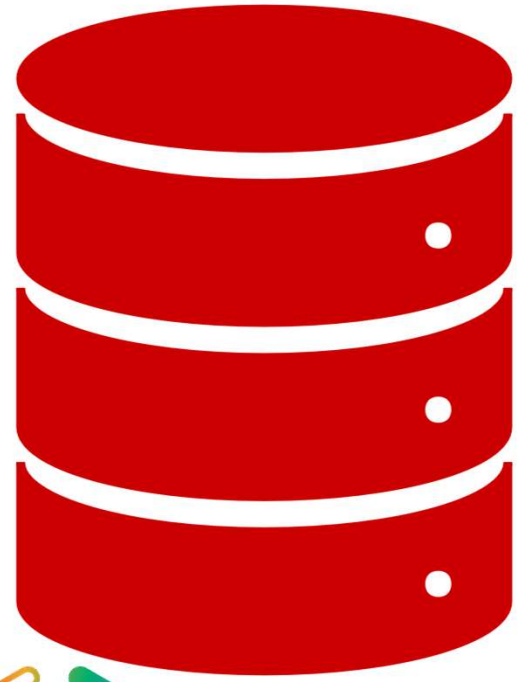
- Reverse Engineer
- Develop
- Unit Test
- Commit Changes

Introduction to Visual Studio

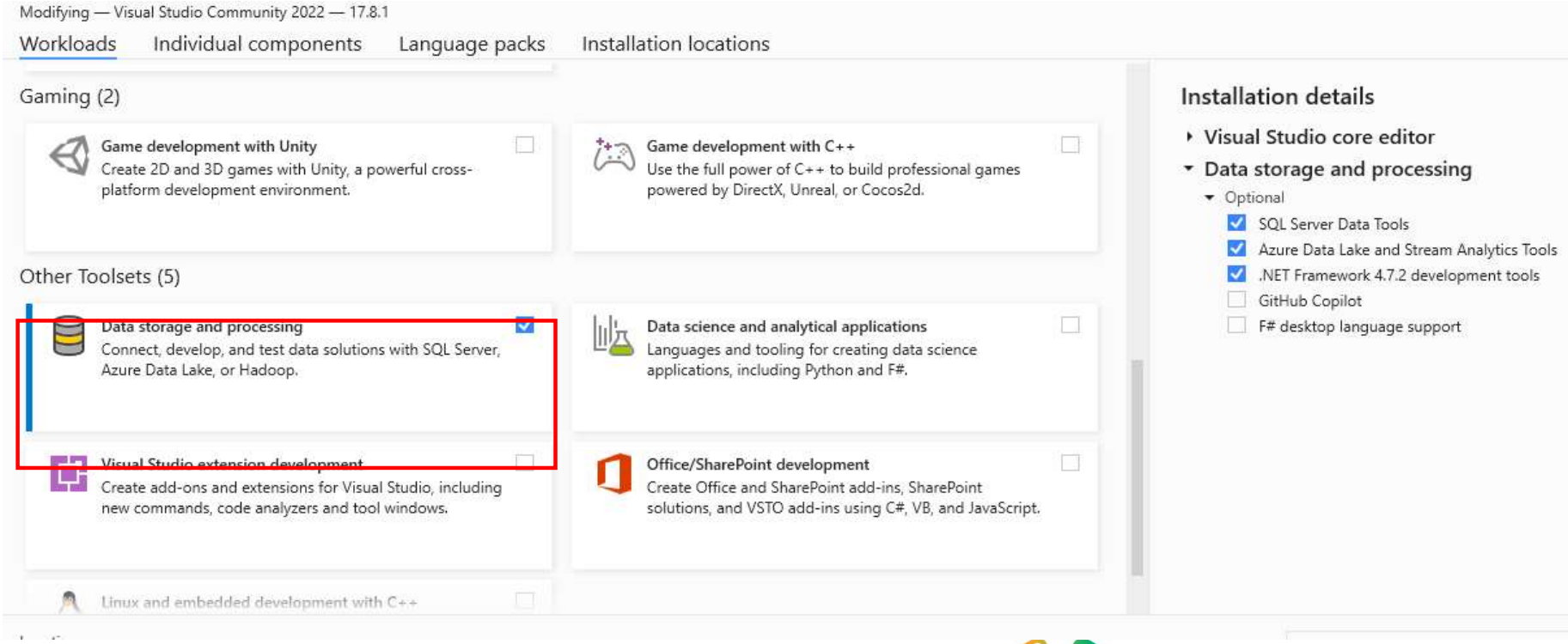
SQL Database Project

Visual Studio – SQL Database Project

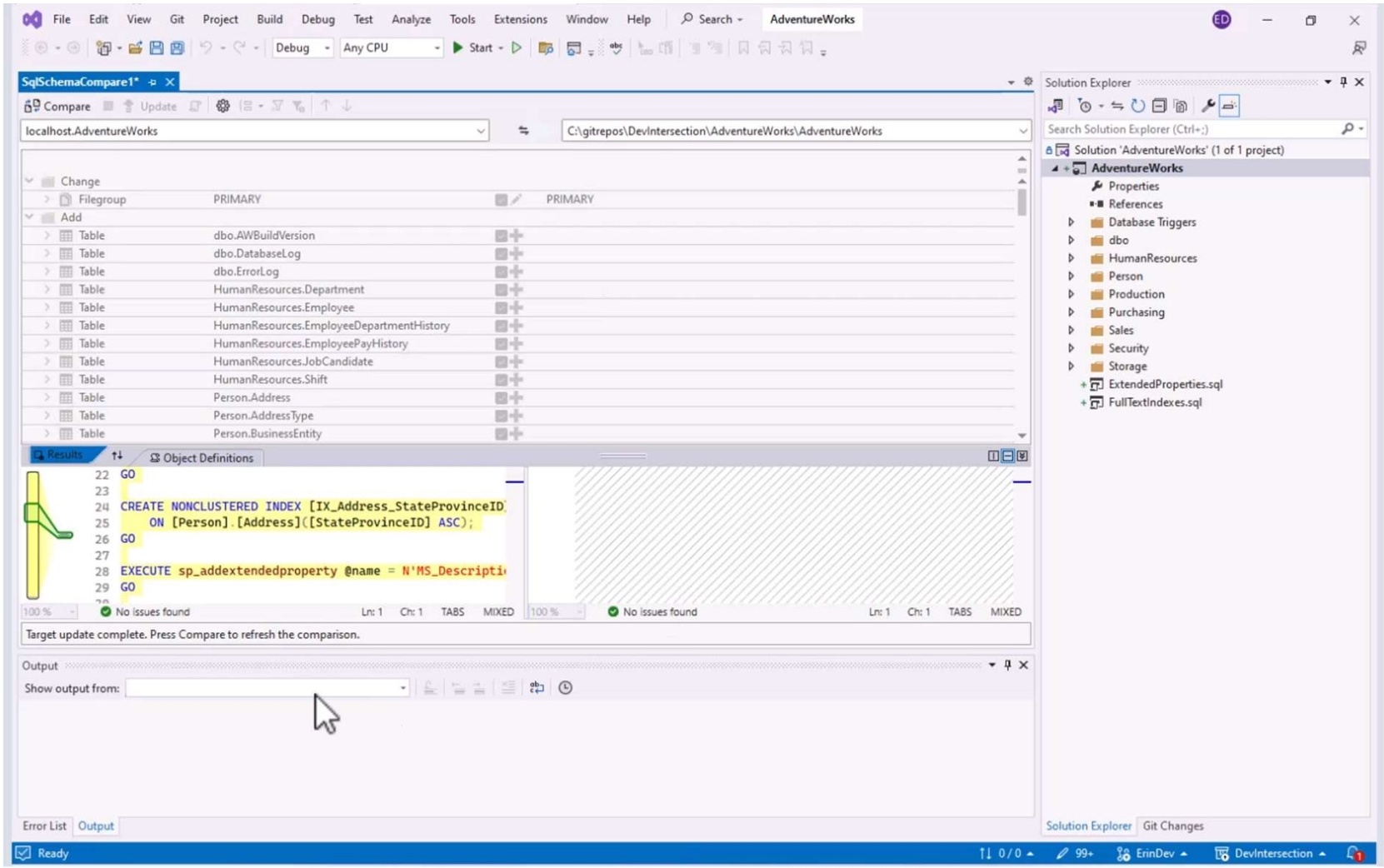
- Long-Time Support for SQL Databases
- Build Projects with msbuild.exe
- Creates a .DACPAC for Deployment



Visual Studio Installer



SQL Database Project Demo


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
AdventureWorks
Debug
Any CPU
Start
SqlSchemaCompare1*
Compare
Update
localhost.AdventureWorks
C:\gitrepos\DevIntersection\AdventureWorks\AdventureWorks
Change
Filegroup PRIMARY PRIMARY
Add
Table dbo.AWBuildVersion
Table dbo.DatabaseLog
Table dbo.ErrorLog
Table HumanResources.Department
Table HumanResources.Employee
Table HumanResources.EmployeeDepartmentHistory
Table HumanResources.EmployeePayHistory
Table HumanResources.JobCandidate
Table HumanResources.Shift
Table Person.Address
Table Person.AddressType
Table Person.BusinessEntity
Results
Object Definitions
22 GO
24 CREATE NONCLUSTERED INDEX [IX_Address_StateProvinceID
25 ON [Person].[Address]([StateProvinceID] ASC);
26 GO
28 EXECUTE sp_addextendedproperty @name = N'MS_Descripti
29 GO
No issues found
Ln: 1 Ch: 1 TABS MIXED
Target update complete. Press Compare to refresh the comparison.
Output
Show output from:
Error List
Output
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'AdventureWorks' (1 of 1 project)
AdventureWorks
Properties
References
Database Triggers
dbo
HumanResources
Person
Production
Purchasing
Sales
Security
Storage
ExtendedProperties.sql
FullTextIndexes.sql
Solution Explorer
Git Changes
Ready
ErinDev
DevIntersection

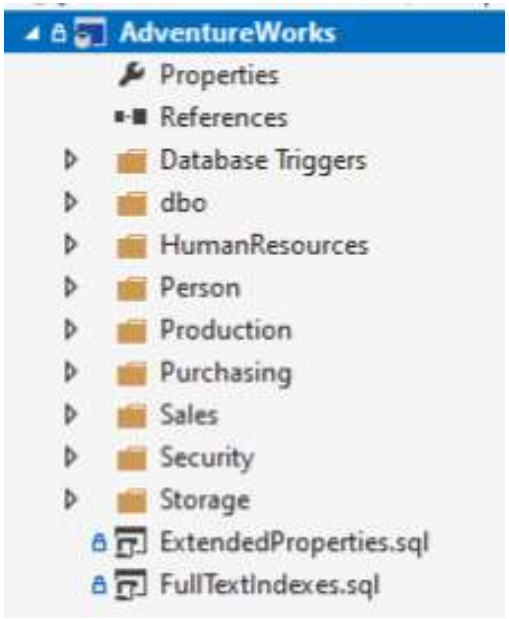
Introduction to GitHub

GitHub

- Public-focused Source Control Solution
- Code-focused
  - Tasks limited to Issue Logs
  - Often integration with Jira for task boards
- Acquired by Microsoft in 2019

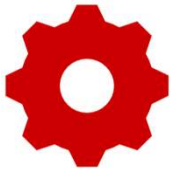
GitHub Actions

- Manages code builds and deployments
- Triggered from many events
  - Code check-ins (push)
  - Pull Requests (pull_request)
- Define **YOUR** workflow with YAML!



What is YAML?

"YAML Ain't Markup Language



Configuration Layout

Used by both GitHub Actions and Azure Pipelines

Links to GitHub Actions

- Workflow Syntax - https://tinyurl.com/4yz9bmjr
- Actions Marketplace - https://tinyurl.com/386ner8r

YAML - GitHub Actions vs Azure Pipelines

```yaml
name: Build AdventureWorks

on:
  push:
    branches: [main]
  workflow_dispatch:

jobs:
  build:
    runs-on: windows-latest
    steps:
    - uses: actions/checkout@v3

    - name: Add MSBuild to PATH
      uses: microsoft/setup-msbuild@v1.0.2

    - name: Build
      working-directory: ${{env.GITHUB_WORKSPACE}}
      run: msbuild /m /p:Configuration=Release Adventureworks/Adventureworks.sqlproj
```

```yaml
trigger:
- main

pool:
  vmImage: windows-latest

jobs:
- job: buildDACPAC
  displayName: Build Azure Data Studio DACPAC
  steps:
  Settings
  - task: DotNetCoreCLI@2
    inputs:
      command: 'build'
      projects: '**/Adventureworks.sqlproj'
```

GitHub Workflow - Triggers

```
3    on:
4      pull_request:
5        branches: [ "test" ]
6
7      workflow_dispatch:
```

Quick Example #1

- 2 Triggers
- Pull request going to ‘test’ branch
  - Workflow runs when Pull Request is opened
- Manual trigger, workflow_dispatch

```
3    on:
4      pull_request:
5        types: [ "closed" ]
6        branches: [ "dev" ]
7
8      workflow_dispatch:
```

Quick Example #2

- **2 Triggers**
- **Pull request going to ‘dev’**
  - **Workflow runs when PR is closed**
- **Manual trigger, workflow_dispatch**

GitHub Workflow - Triggers

```
3     on:
4       push:
5         branches: [ "dev" ]
```

Quick Example #3

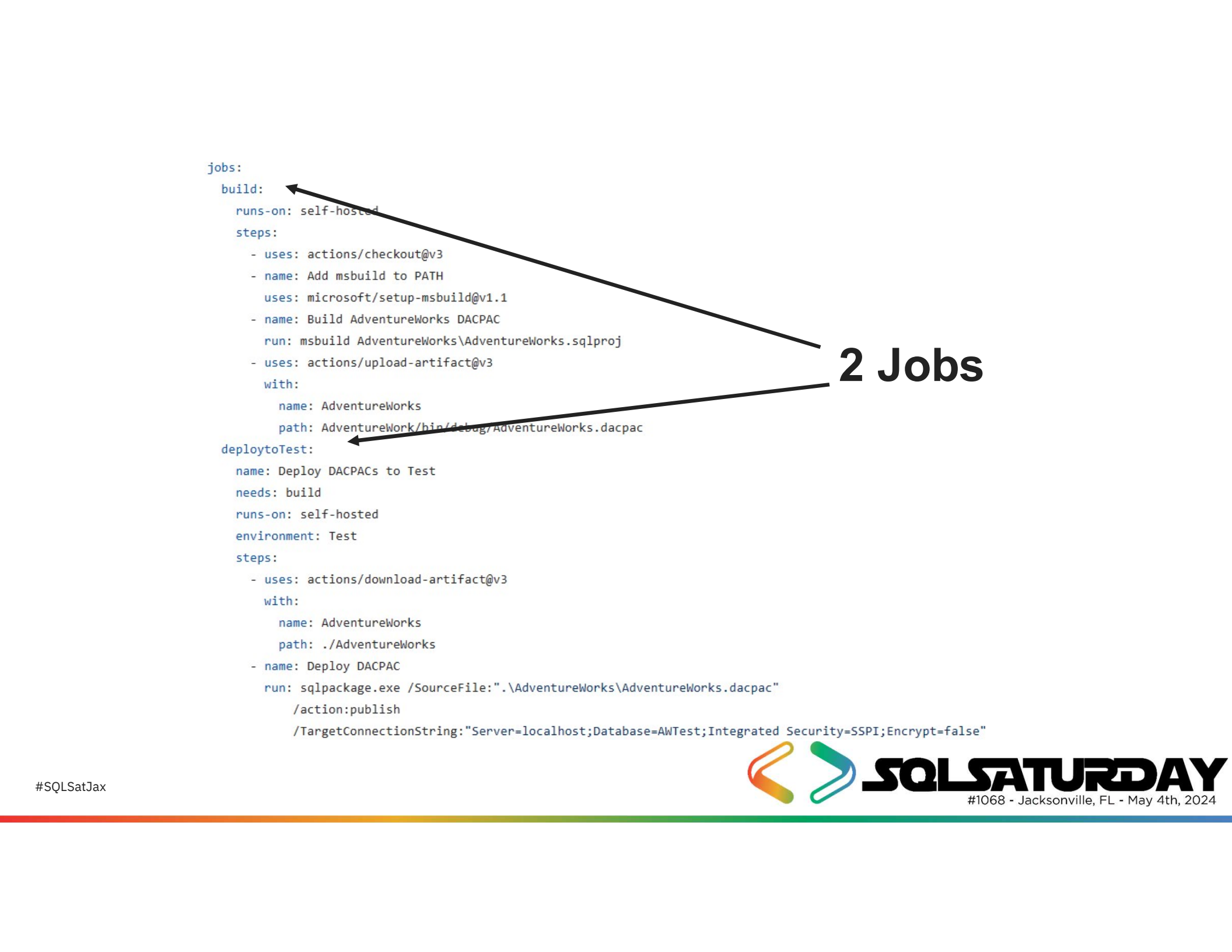
- 1 Trigger - Push code to ‘dev’ branch

Jobs

- Run Sequentially or in Parallel
- Inherit settings from the Workflow
- Override Workflow settings
  - Ex: runs-on

```yaml
jobs:
  build:
    runs-on: self-hosted
    steps:
      - uses: actions/checkout@v3
      - name: Add msbuild to PATH
        uses: microsoft/setup-msbuild@v1.1
      - name: Build AdventureWorks DACPAC
        run: msbuild AdventureWorks\AdventureWorks.sqlproj
      - uses: actions/upload-artifact@v3
        with:
          name: AdventureWorks
          path: AdventureWork/bin/debug/AdventureWorks.dacpac
  deploytoTest:
    name: Deploy DACPACs to Test
    needs: build
    runs-on: self-hosted
    environment: Test
    steps:
      - uses: actions/download-artifact@v3
        with:
          name: AdventureWorks
          path: ./AdventureWorks
      - name: Deploy DACPAC
        run: sqlpackage.exe /SourceFile:".\AdventureWorks\AdventureWorks.dacpac"
            /action:publish
            /TargetConnectionString:"Server=localhost;Database=AWTest;Integrated Security=SSPI;Encrypt=false"
```

```yaml
jobs:
  build:
    runs-on: self-hosted
    steps:
      - uses: actions/checkout@v3
      - name: Add msbuild to PATH
        uses: microsoft/setup-msbuild@v1.1
      - name: Build AdventureWorks DACPAC
        run: msbuild AdventureWorks\AdventureWorks.sqlproj
      - uses: actions/upload-artifact@v3
        with:
          name: AdventureWorks
          path: AdventureWork/bin/debug/AdventureWorks.dacpac
  deploytoTest:
    name: Deploy DACPACs to Test
    needs: build
    runs-on: self-hosted
    environment: Test
    steps:
      - uses: actions/download-artifact@v3
        with:
          name: AdventureWorks
          path: ./AdventureWorks
      - name: Deploy DACPAC
        run: sqlpackage.exe /SourceFile:".\AdventureWorks\AdventureWorks.dacpac"
             /action:publish
             /TargetConnectionString:"Server=localhost;Database=AWTest;Integrated Security=SSPI;Encrypt=false"
```



```yaml
jobs:
  build:
    runs-on: self-hosted
    steps:
      - uses: actions/checkout@v3
      - name: Add msbuild to PATH
        uses: microsoft/setup-msbuild@v1.1
      - name: Build AdventureWorks DACPAC
        run: msbuild AdventureWorks\AdventureWorks.sqlproj
      - uses: actions/upload-artifact@v3
        with:
          name: AdventureWorks
          path: AdventureWork/bin/debug/AdventureWorks.dacpac
  deploytoTest:
    name: Deploy DACPACs to Test
    needs: build
    runs-on: self-hosted
    environment: Test
    steps:
      - uses: actions/download-artifact@v3
        with:
          name: AdventureWorks
          path: ./AdventureWorks
      - name: Deploy DACPAC
        run: sqlpackage.exe /SourceFile:".\AdventureWorks\AdventureWorks.dacpac"
            /action:publish
            /TargetConnectionString:"Server=localhost;Database=AWTest;Integrated Security=SSPI;Encrypt=false"
```

Job Steps

```yaml
steps:
  - uses: actions/checkout@v3
  - name: Add msbuild to PATH
    uses: microsoft/setup-msbuild@v1.1
  - name: Build AdventureWorks DACPAC
    run: msbuild AdventureWorks\AdventureWorks.sqlproj
  - uses: actions/upload-artifact@v3
    with:
      name: AdventureWorks
      path: AdventureWork/bin/debug/AdventureWorks.dacpac
```

```yaml
jobs:
  build:
    runs-on: self-hosted
    steps:
      - uses: actions/checkout@v3
      - name: Add msbuild to PATH
        uses: microsoft/setup-msbuild@v1.1
      - name: Build AdventureWorks DACPAC
        run: msbuild AdventureWorks\AdventureWorks.sqlproj
      - uses: actions/upload-artifact@v3
        with:
          name: AdventureWorks
          path: AdventureWork/bin/debug/AdventureWorks.dacpac
  deploytoTest:
    name: Deploy DACPACs to Test
    needs: build
    runs-on: self-hosted
    environment: Test
    steps:
      - uses: actions/download-artifact@v3
        with:
          name: AdventureWorks
          path: ./AdventureWorks
      - name: Deploy DACPAC
        run: sqlpackage.exe /SourceFile:".\AdventureWorks\AdventureWorks.dacpac"
             /action:publish
             /TargetConnectionString:"Server=localhost;Database=AWTest;Integrated Security=SSPI;Encrypt=false"
```

Runners

```
build:
  runs-on: self-hosted
```

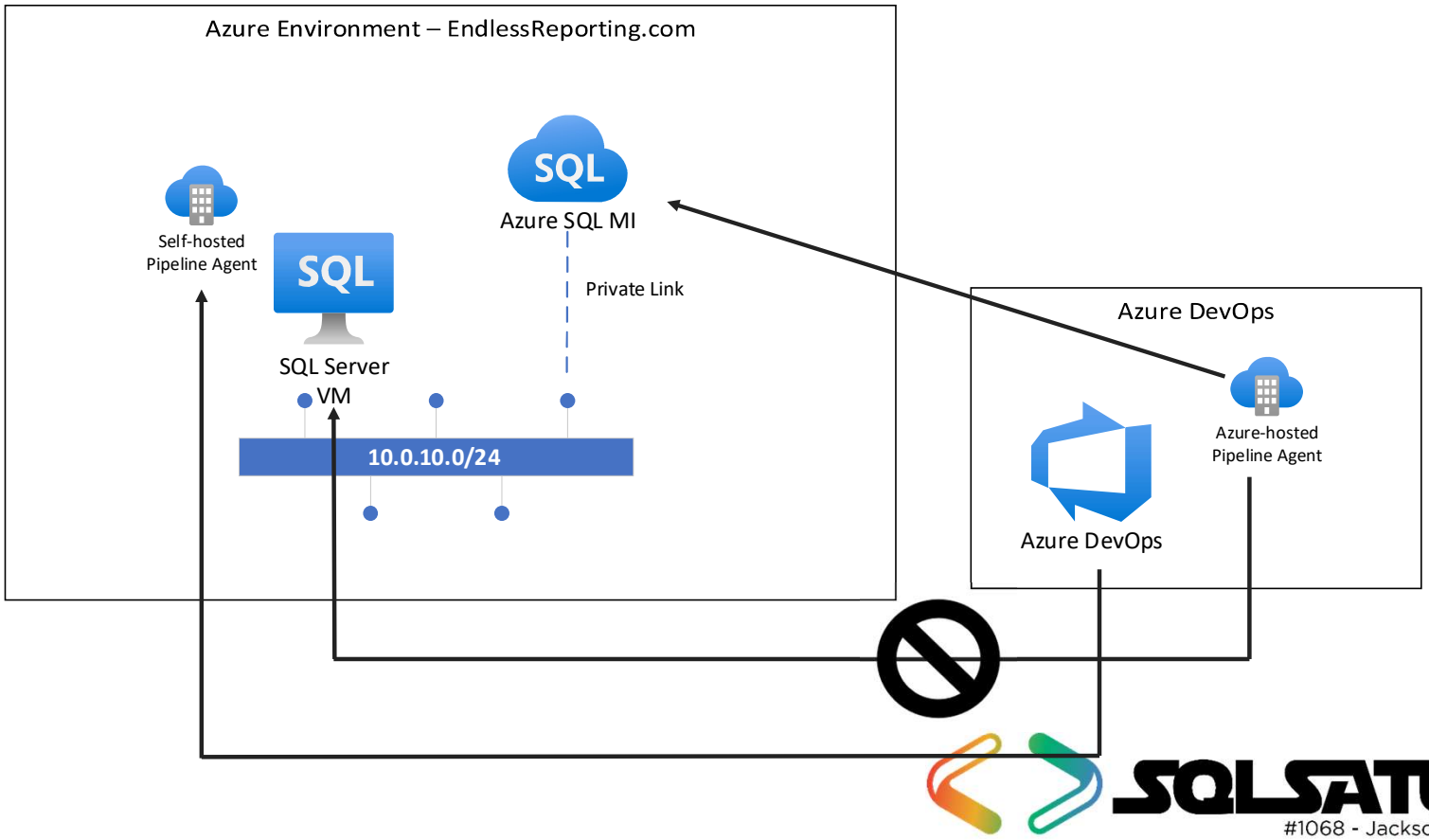
- Workflow Execution Agent
- GitHub-hosted
  - Windows
  - Ubuntu
- Self-hosted
  - Windows
  - Linux
  - Mac

← Much, much faster

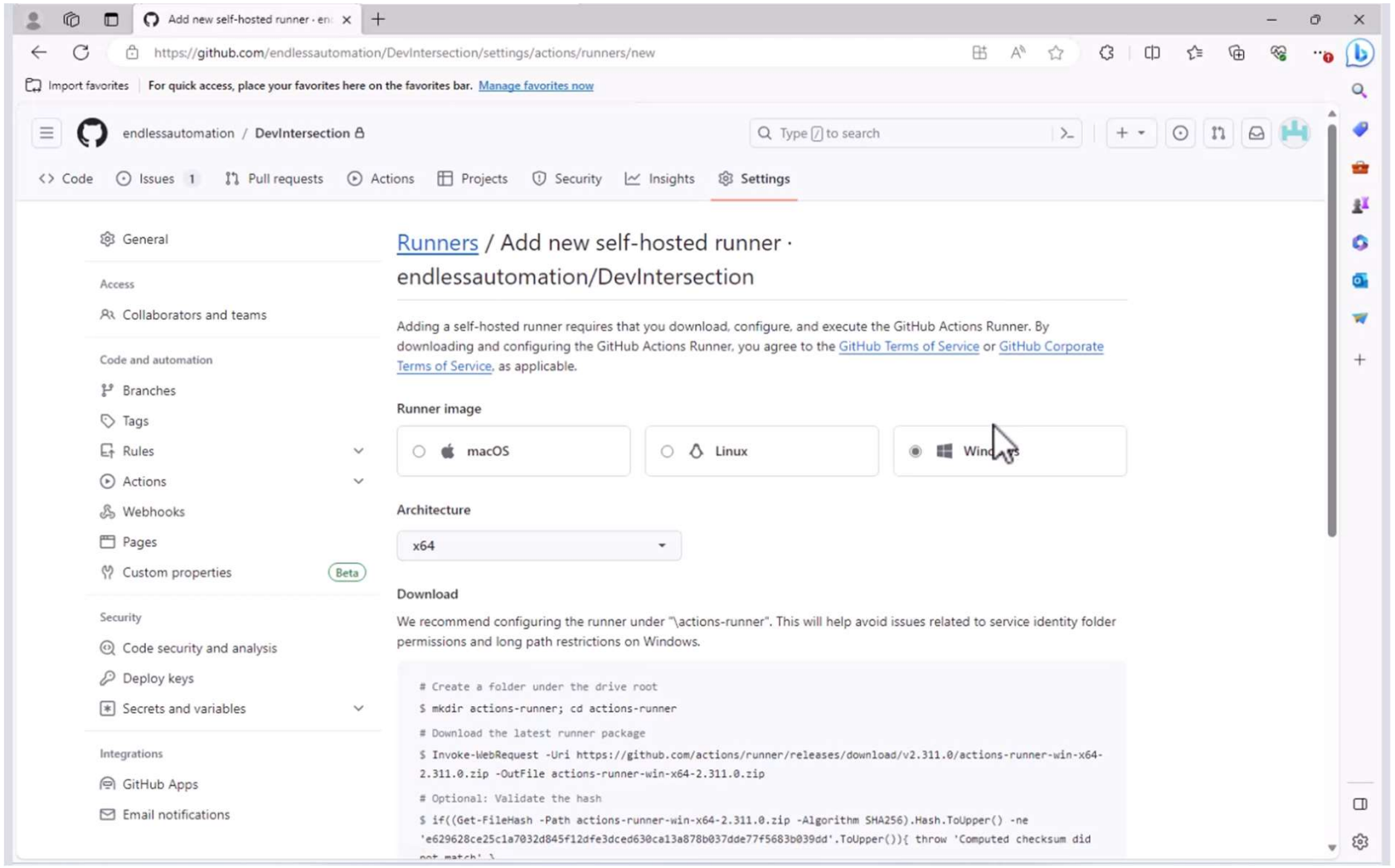
Network Communications

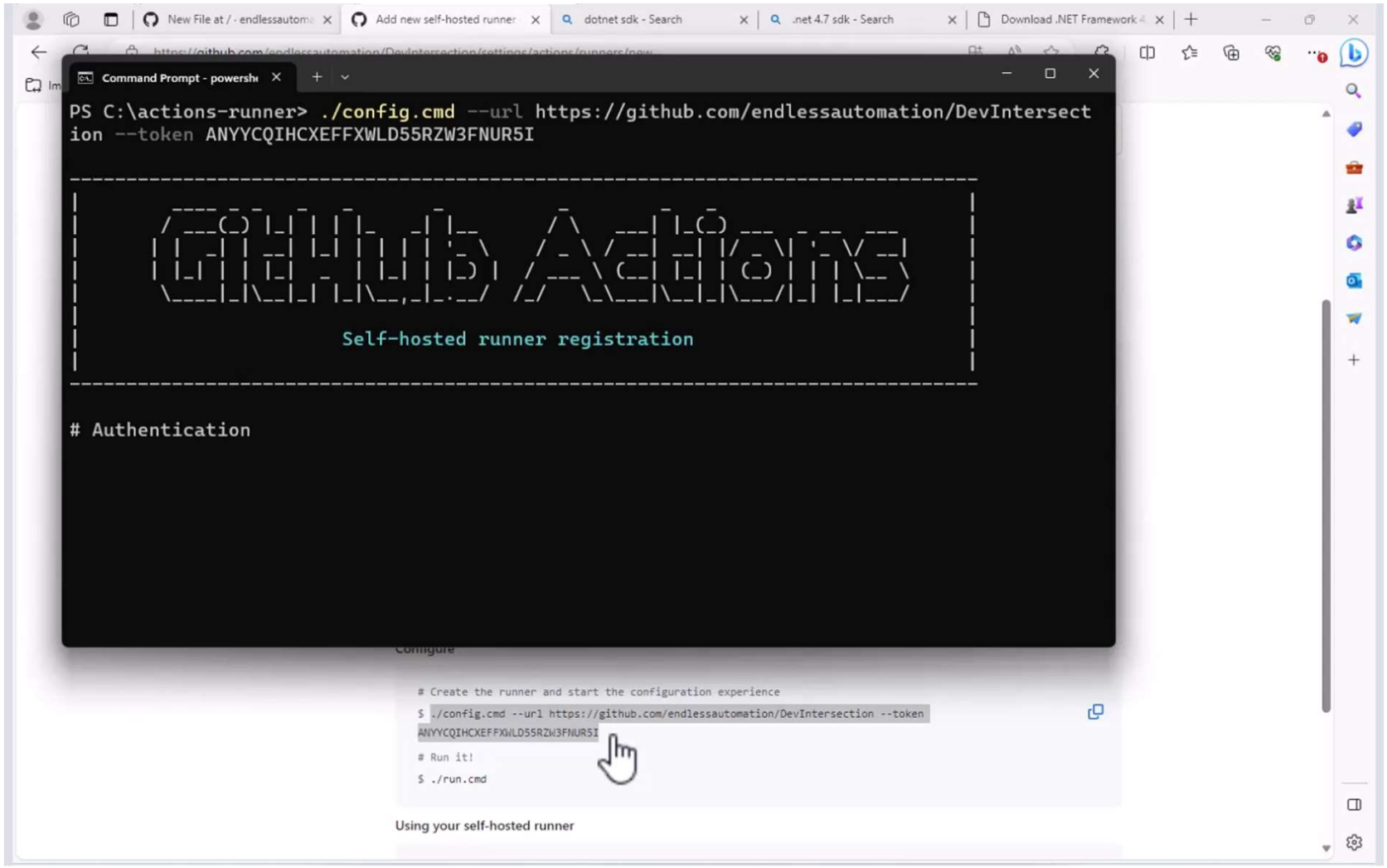
GitHub Action Workflows and Azure SQL DB/MI vs SQL Server Instances

Self-hosted Runners



GitHub Actions Self-Hosted Runner


Add new self-hosted runner · en
https://github.com/endlessautomation/DevIntersection/settings/actions/runners/new
Import favorites
For quick access, place your favorites here on the favorites bar. Manage favorites now
endlessautomation / DevIntersection
Type / to search
Code
Issues 1
Pull requests
Actions
Projects
Security
Insights
Settings
General
Access
Collaborators and teams
Code and automation
Branches
Tags
Rules
Actions
Webhooks
Pages
Custom properties
Beta
Security
Code security and analysis
Deploy keys
Secrets and variables
Integrations
GitHub Apps
Email notifications
Runners / Add new self-hosted runner · endlessautomation/DevIntersection
Adding a self-hosted runner requires that you download, configure, and execute the GitHub Actions Runner. By downloading and configuring the GitHub Actions Runner, you agree to the GitHub Terms of Service or GitHub Corporate Terms of Service, as applicable.
Runner image
macOS
Linux
Architecture
x64
Download
We recommend configuring the runner under "\actions-runner". This will help avoid issues related to service identity folder permissions and long path restrictions on Windows.
# Create a folder under the drive root
$ mkdir actions-runner; cd actions-runner
# Download the latest runner package
$ Invoke-WebRequest -Uri https://github.com/actions/runner/releases/download/v2.311.0/actions-runner-win-x64-2.311.0.zip -OutFile actions-runner-win-x64-2.311.0.zip
# Optional: Validate the hash
$ if((Get-FileHash -Path actions-runner-win-x64-2.311.0.zip -Algorithm SHA256).Hash.ToUpper() -ne 'e629628ce25c1a7032d845f12dfe3dced630ca13a878b037dde77f5683b039dd'.ToUpper()){ throw 'Computed checksum did


Command Prompt - powersh
PS C:\actions-runner> ./config.cmd --url https://github.com/endlessautomation/DevIntersection --token ANYYCQIHCXEFFXWLD55RZW3FNUR5I
Self-hosted runner registration
Authentication
Create the runner and start the configuration experience
$ ./config.cmd --url https://github.com/endlessautomation/DevIntersection --token ANYYCQIHCXEFFXWLD55RZW3FNUR5I
# Run it!
$ ./run.cmd
Using your self-hosted runner

GitHub Workflows Demo

Code Validation

- Does the code build?

- Does it meet basic tests?

Fix it before it gets to QA!

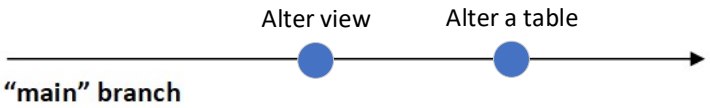
Code Validation

```yaml
name: Validate Development Branch

on:
  pull_request:
    branches: [ "dev" ]
  workflow_dispatch:

jobs:
  buildCode:
    name: Build SQL Projects

    runs-on: windows-latest
    steps:
    - uses: actions/checkout@v3
    - name: Add msbuild to PATH
      uses: microsoft/setup-msbuild@v1.1

    - name: Build AdventureWorks DACPAC
      run: msbuild AdventureWorks\AdventureWorks.sqlproj
```

Branching Code

Branching Strategy

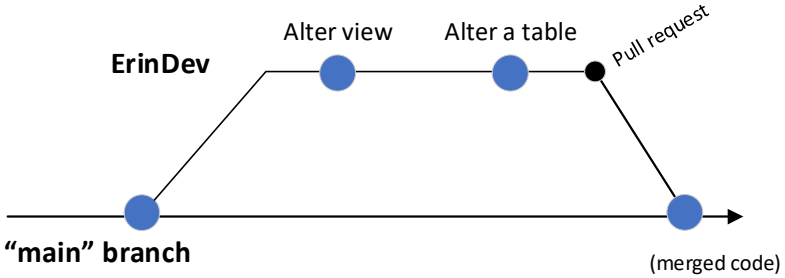
- main branch



- Harder to determine when to start workflow
  - Trigger after view change?
  - Trigger after table change?

Branching Strategy

- Developer branch and main

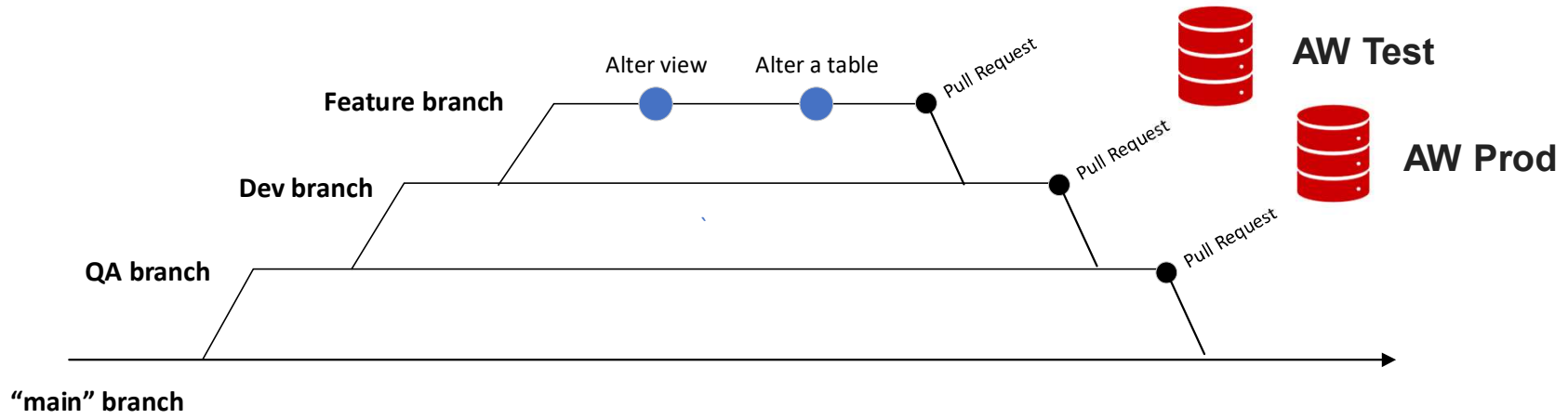


- Changes are merged into main together
- Pull Request (PR) can start automations

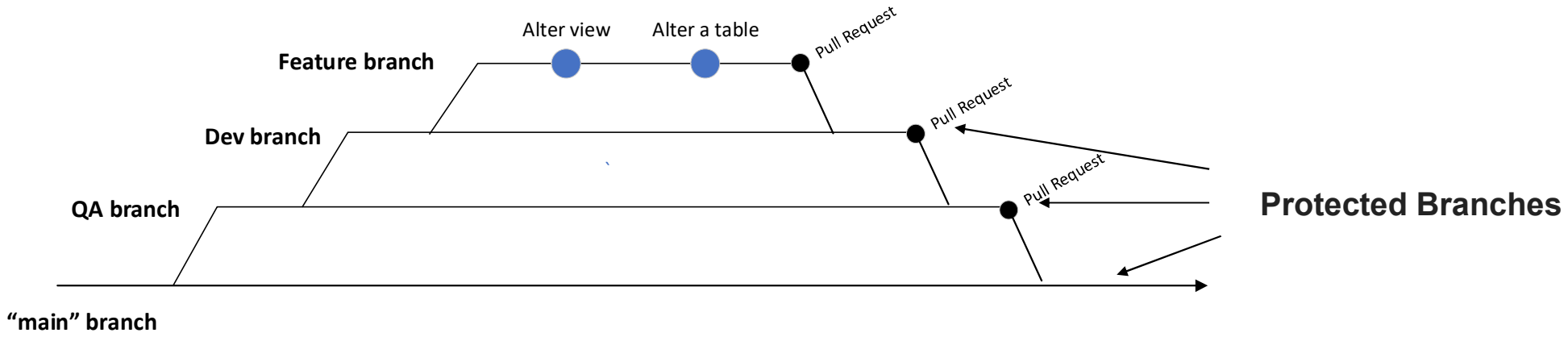
Pull Request

- Developer believes code is finished
- Changes are grouped together
- Associate work items
  - Requirements
  - Bugs
- Require approval by another team member

Potential Branching Strategy in GitHub



Potential Branching Strategy in GitHub



Pros and Cons

Pros

- Code collection
- Hotfixes from main to QA

Cons

- Lots of branches to maintain
- Pull requests – wrong target

Protecting Code and Deployments

Wrap-up

- Power Duo – Visual Studio and GitHub
- Manage Database Schema Alongside Application Code
- Build and deploy changes with little effort