

sqlbits

2025

18-21 JUNE, LONDON EXCEL



SQL Projects and Source Control Integration for SQL in Fabric, Azure SQL, and SQL Server



Hamish Watson

DevOps Consultant, Morph iT Limited

CEO, MakeStuffGo



- 25+ years mucking about with 1s & 0s
- Bringing DevOps to databases (and the masses) is a personal passion
- Understanding data and cloud is a company driver
- Technologist who understands business value...
- #makeStuffGo



@theHybridDBA



<https://www.makestuffgo.com>



hamish@morphit.co.nz



Before we start...

- Make sure you have VS Code (free)
 - + the mssql extension
 - + the SQL Database Projects extension
- A (free) GitHub account
- .NET 8 SDK
- A database somewhere

What you signed up for...

This workshop is a deep dive on **database DevOps centered on the SQL database projects format**, where you will learn practical techniques for managing database changes whether your workload is operational, analytical, or somewhere in between. Our exploration begins with the foundational CI/CD capabilities for SQL projects, where you can ensure no matter how a **database is developed it is in source control** and can be verified before deployments to one or more environments. The same shared SQL projects format applies to the **whole Microsoft SQL family**, so we'll examine the advantages and special capabilities to understand when implementing DevOps practices for Fabric SQL, Azure SQL, or SQL Server. We conclude our workshop with some special topics at the core of **good data DevOps practices**, including security management, static data management, and coordinating changes with other workloads.

Ground rules

- Follow the conference code of conduct:
 - “Harassment-free conference experience”
 - “Deliberate intimidation”
 - “Sustained disruption”
- Your questions are welcome
 - No bad questions
 - May have to hold on a question due to schedule
- Try to leave space for all the knowledge in the room
- Practice the human side of DevOps - empathetic, collaborative

Agenda

9am

Intro

Foundations of SQL
projects

10am

SQL Variants

Advanced Topics Part
One

1pm

 Lunch 

2pm

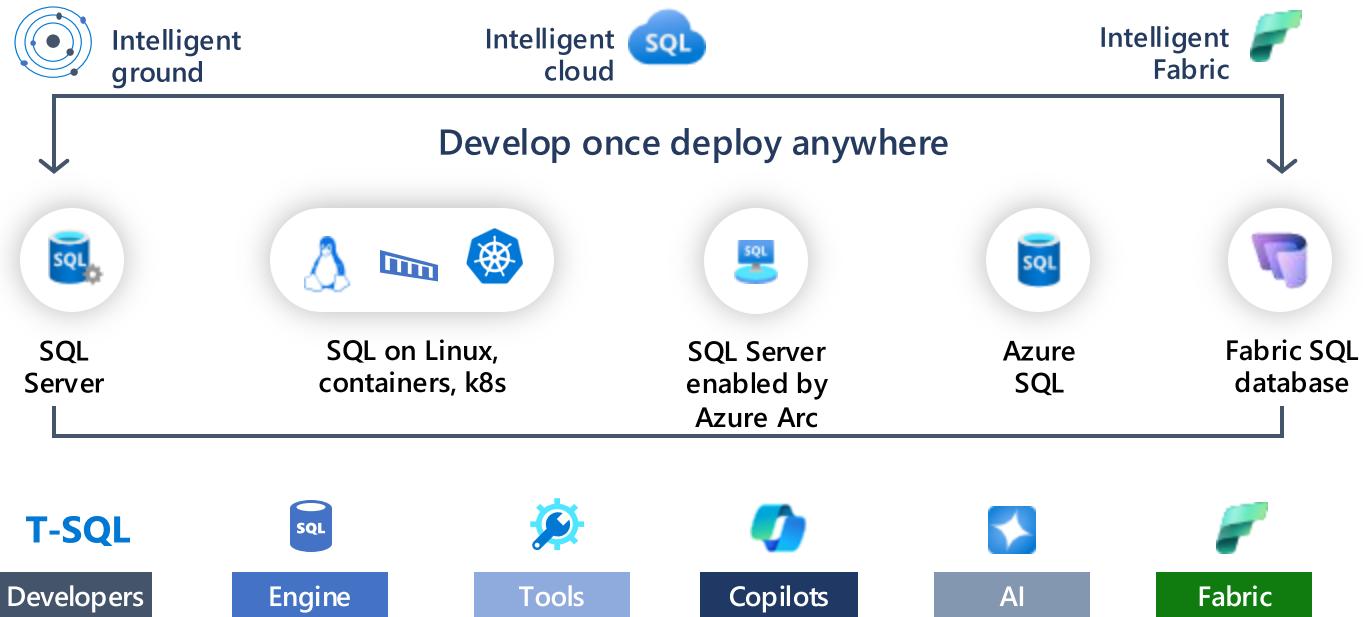
Advanced Topics Part
Two

Troubleshooting and
Optimizing DevOps
Workflows

5pm

Closing

Microsoft SQL - a whole range



Why DevOps?

- Issues identified as early as possible (build)
- Scalability and Flexibility
 - Add team members and know the releases will be consistent
- Consistent releases
- Increased Efficiency
- Removes single point of failure (person)
- Automated documentation
- Improved quality

Team looks like left side of this tree, flourishing, achieving more and more, people want to be in the team



Why Not DevOps?

- Errors identified at production time
- Team unclear on how to deploy
- ~~Legacy Systems~~
- Small/Simple projects
- Resistance to change
- Regulatory Constraints

Team ends up looking like this tree, unmotivated, confused

What is DevOps?



Tools



Process



People



Culture

<https://www.red-gate.com/blog/database-devops/the-four-pillars-of-devops>

There are a few tools out there.....

XebiaLabs
Deliver Faster

 Follow @xebialabs

DevSecOps Tools Periodic Table

1 En	Aja Atlassian Jira Align																		2 Os														
3 En	Daa Digital Agility	4 En	Tp Targetprocess																Gi Git														
11 En	Pv Planview	12 En	Br Broadcom Rally																Gh GitHub														
19 En	Aj Atlassian Jira	20 En	Dd Datadog	21 En	Bp Big Panda	22 Pd	In Instana	23 En	Acp AWS CodePipeline	24 En	Mt Microsoft Teams	25 Os	Rha Red Hat Ansible	26 En	Ht HashiCorp Terraform	27 Os	Dk Docker	28 En	Rho Red Hat OpenShift	29 Os	Lb Liquibase	30 Fm	Dp Delphix	31 En	Ud UrbanCode Deploy	32 Pd OpMx	33 Os HashiCorp Vault	34 En	Sy Snyk	35 Os Pd PagerDuty	36 Fm Abb Atlassian Bitbucket		
37 En	Sp Splunk	38 En	Ad AppDynamics	39 En	Kb Kibana	40 En	Dar DigitalLab Release	41 En	Ur UrbanCode Confluence	42 En	Ac Atlassian Confluence	43 Os	Ch Chef	44 En	Acf AWS CloudFormation	45 Os	Ku Kubernetes	46 En	Ak Amazon EKS	47 En	De Docker Enterprise	48 En	Rf RidgeIt Flyway	49 En	Ha Harness	50 En	Pi Pulumi	51 Os Sr SonarQube	52 En	Ff Micro Focus Fortify SCA	53 En	Azf Azure Functions	54 En Ci Compuware ISPW
55 En	Dt Dynatrace	56 En	Nr New Relic	57 En	Dh Docker Hub	58 En	Np npm	59 En	Ja JFrog Artifactory	60 En	So Stack Overflow	61 En	Sl Slack	62 Os	Hc HashiCorp Consul	63 Fm	Pu Puppet	64 En	Azk Azure AKS	65 En	Ae Amazon ECS	66 Fm	Qt Quest Road	67 Os	Sk Spinmaker	68 En	Od Octopus Deploy	69 En	Sb Synapsys Black Duck	70 En	Cx Checkmarx SAST	71 Fm He Heroku	72 En Al AWS Lambda
73 Os	Gr Grafana	74 Os	EI Elastic ELK Stack	75 Os	Yn Yarn	76 Os	Nu NuGet	77 Os	Snx Sonatype Nexus	78 Os	Mm Mattemost	79 En	Mr Miro	80 En	MI Mural	81 Os	Hp HashiCorp Packer	82 En	Gk Google GKE	83 Os	Hm Helm	84 Os	Fx Flux	85 Os	Tk Tekton	86 En	Acd AWS CodeDeploy	87 Os	Sn Snort	88 Fm Pbs PortSwigger Burp Suite	89 En Gf Google Firebase	90 Os Cf Cloud Foundry	
91 Os	Jn Jenkins	92 En	Azc Azure DevOps Code	93 Os	Glc GitLab CI	94 Os	Tr Travis CI	95 Fm	Cc Circle CI	96 Os	Mv Maven	97 Pd	Ab Atlasian Bamboo	98 En	Ga Github Actions	99 En	Acb AWS CodeBuild	100 En	Cf CodeFresh	101 En	Az Azure	102 En	Gc Google Cloud	103 En	Aws AWS	104 Os	Os OpenStack	105 Os Bg Backstage					
106 Fr	Tt Tricentis Tosca	107 Fr	Se Selenium	108 Fr	Ju JUnit	109 Pd	Sl Sauce Labs	110 En	Ct Compuware Topaz	111 En	Ap Appium	112 Os	Sq Squash TM	113 Fr	Cu Cucumber	114 Fr	Jm Jmeter	115 Pd	Dac Digital.ai Continuous Testing	116 En	Pa Parasoft	117 En	Da Digital.ai	118 En Pvz Planview Viz	119 En Pr Plutora	120 En Dai Digital.ai Intelligence							

91 Os	Jn Jenkins	92 En	Azc Azure DevOps Code	93 Os	Glc GitLab CI	94 Os	Tr Travis CI	95 Fm	Cc Circle CI	96 Os	Mv Maven	97 Pd	Ab Atlasian Bamboo	98 En	Ga Github Actions	99 En	Acb AWS CodeBuild	100 En	Cf CodeFresh	101 En	Az Azure	102 En	Gc Google Cloud	103 En	Aws AWS	104 Os	Os OpenStack	105 Os Bg Backstage
106 Fr	Tt Tricentis Tosca	107 Fr	Se Selenium	108 Fr	Ju JUnit	109 Pd	Sl Sauce Labs	110 En	Ct Compuware Topaz	111 En	Ap Appium	112 Os	Sq Squash TM	113 Fr	Cu Cucumber	114 Fr	Jm Jmeter	115 Pd	Dac Digital.ai Continuous Testing	116 En	Pa Parasoft	117 En	Da Digital.ai	118 En Pvz Planview Viz	119 En Pr Plutora	120 En Dai Digital.ai Intelligence		

What does success look like when working with SQL and DevOps?

- Hamish talking points
 - Reliable Database Deployments
 - Environment Consistency
 - Foundation of Monitoring & Observability
 - Auditors like DevOps
 - Faster Development & Deployment Cycles
 - SAFER Development & Deployment Cycles
 - Apps -> Infrastructure -> Databases -> Analytics
 - Developers, Data Engineers, DBAs collaborate
 - Governance and compliance are less boring

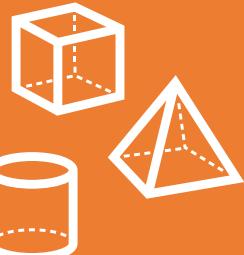
Database as code – SQL projects



Database settings



Pre-deployment script



Database objects
(tables, stored procs, indexes, etc.)



Post-deployment script

build

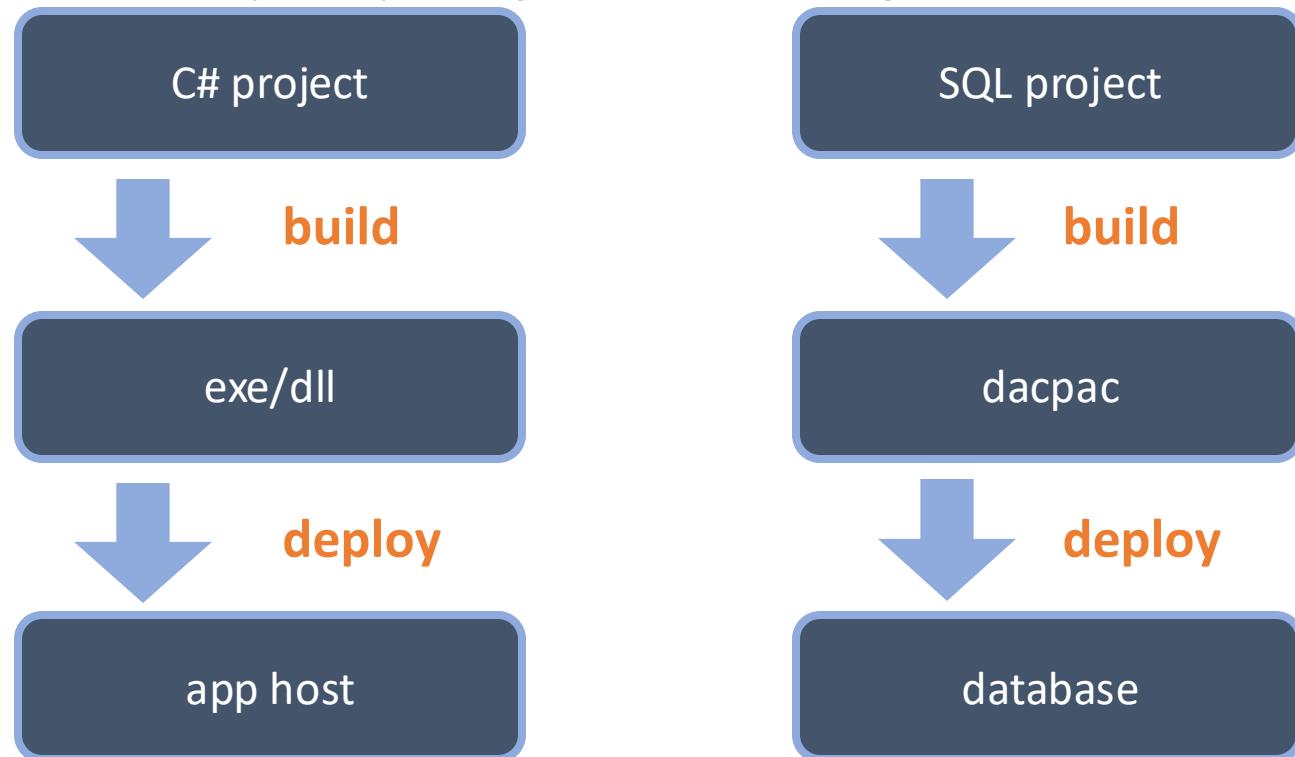
.dacpac



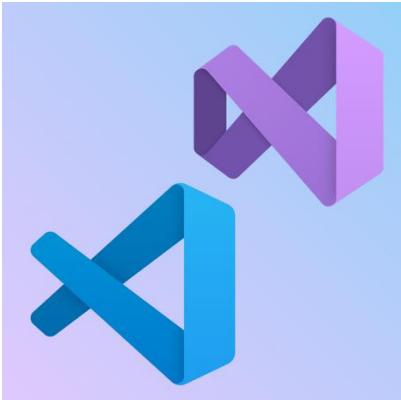
Foundations of CI/CD and SQL projects



Build and ship - project lifecycle

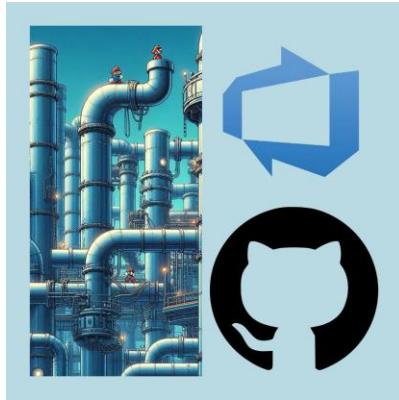


SQL projects tooling ecosystem



Visual Studio and VS Code

Develop, analyze, and compare database objects



GitHub and Azure DevOps tasks

Streamline SQL project deployment from CI/CD environments



SqlPackage CLI

Automate transposing between files and databases

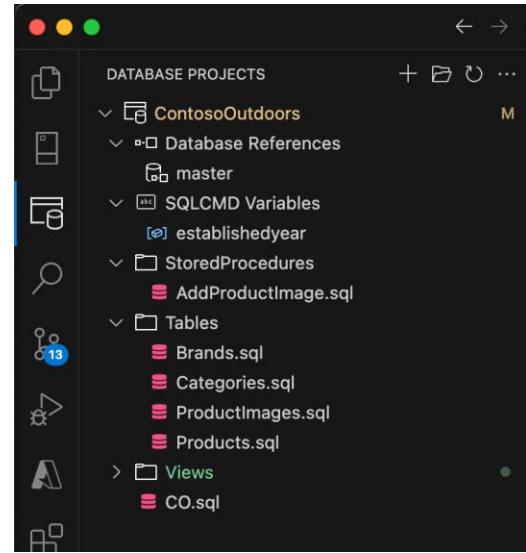
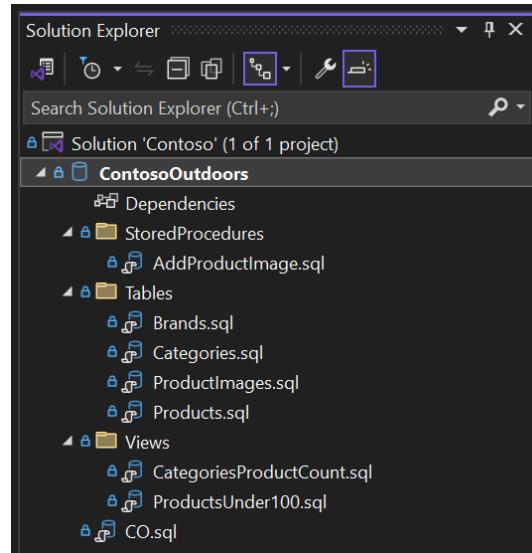
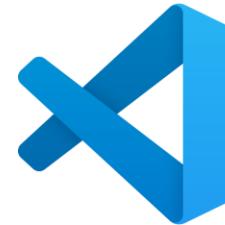
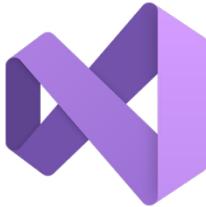


API, class, and model packages

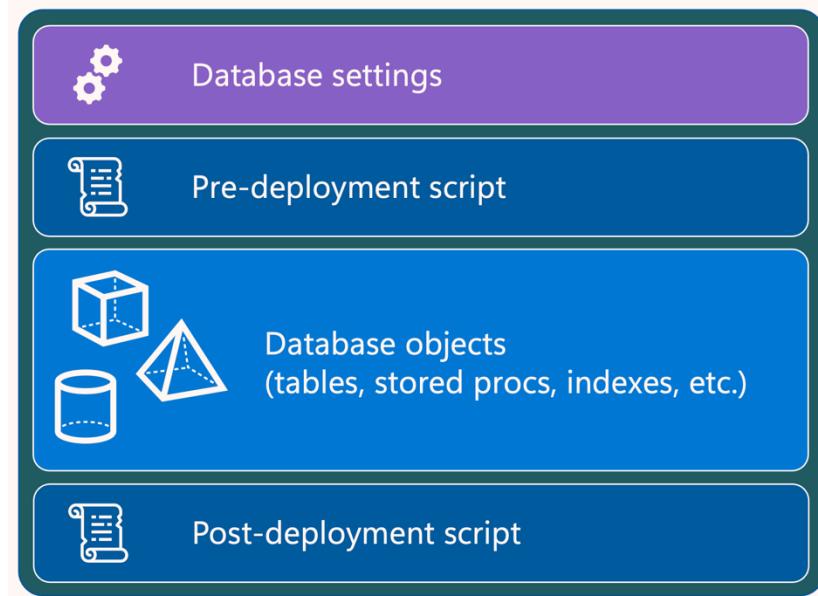
Extensibility for core APIs includes loading analysis rules and database parts from packages

Desktop tooling

- Add objects to projects
- Update projects from databases
- Modify project settings



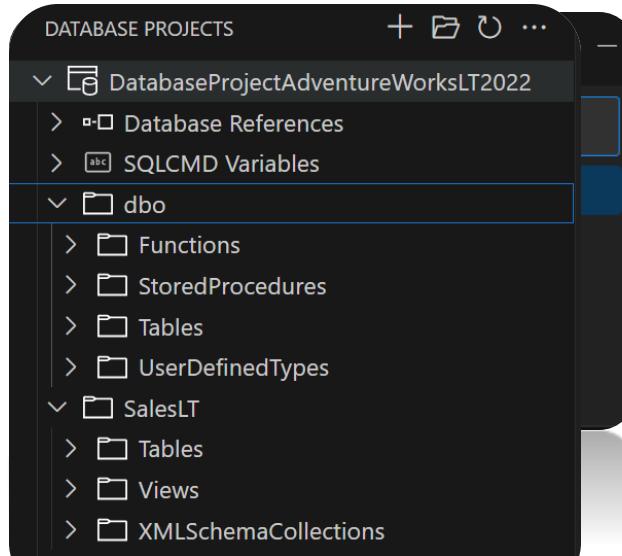
The .sqlproj file behind the scenes



- Automatically includes all *.sq*/files in the folders as objects
- Logins/users/roles/permissions == security objects
- You can store non-object scripts in same folders as <None /> items
- Set specific scripts to execute on every deployment as <PreDeploy /> and <PostDeploy /> items
- Project properties establish database settings

Considerations for .sqlproj

- Folder structure



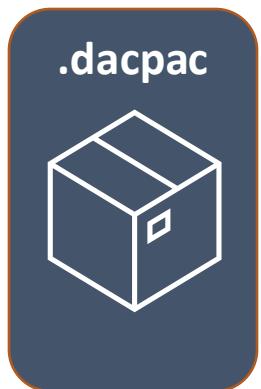
- Security objects

The screenshot shows the 'DataEngineers.sql' file in the 'DataEngineers' database. The code is as follows:

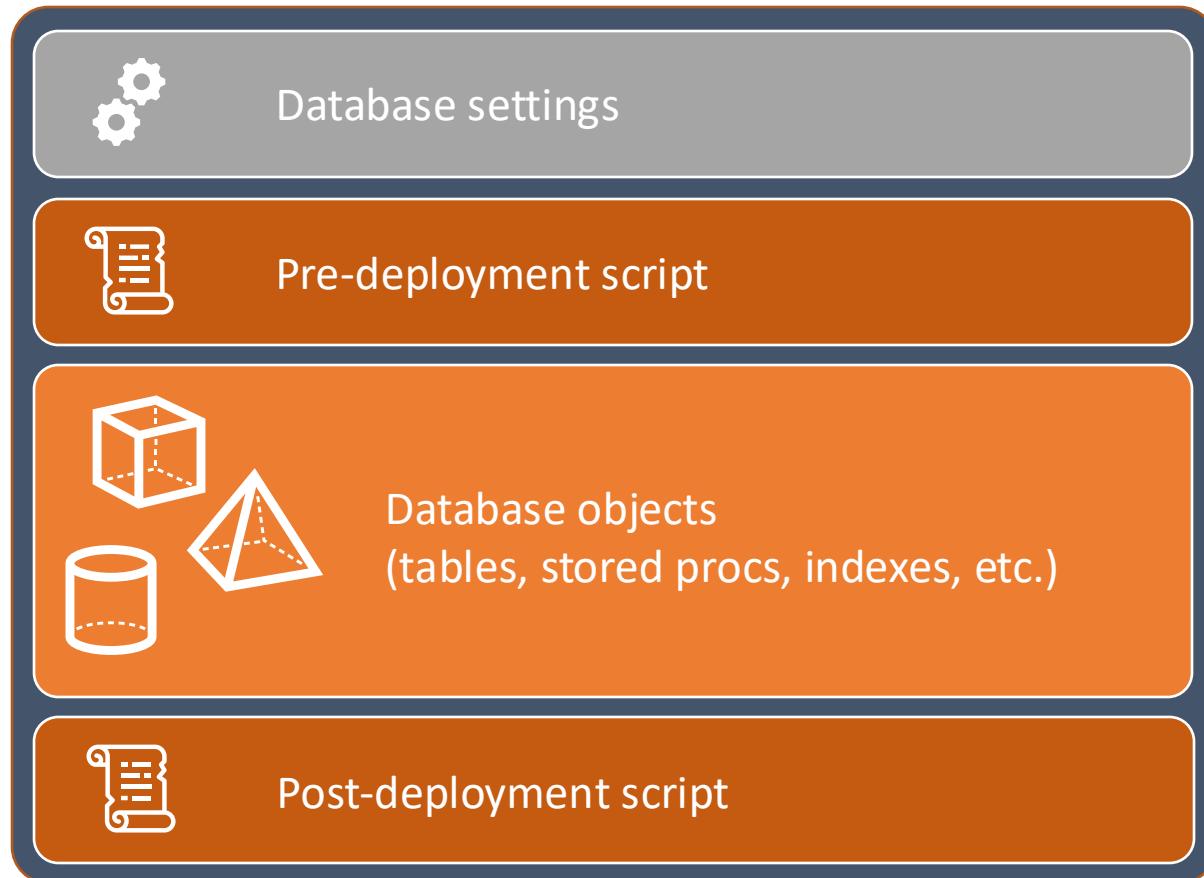
```
1 CREATE ROLE [DataEngineers]
2     AUTHORIZATION [dbo];
3 GO
4
5 ALTER ROLE [DataEngineers] ADD MEMBER [DemoAccount];
6 GO
```

Below the code, the 'Object Explorer' pane shows other files like SalesLT.sql and ExtendedProperties.sql.

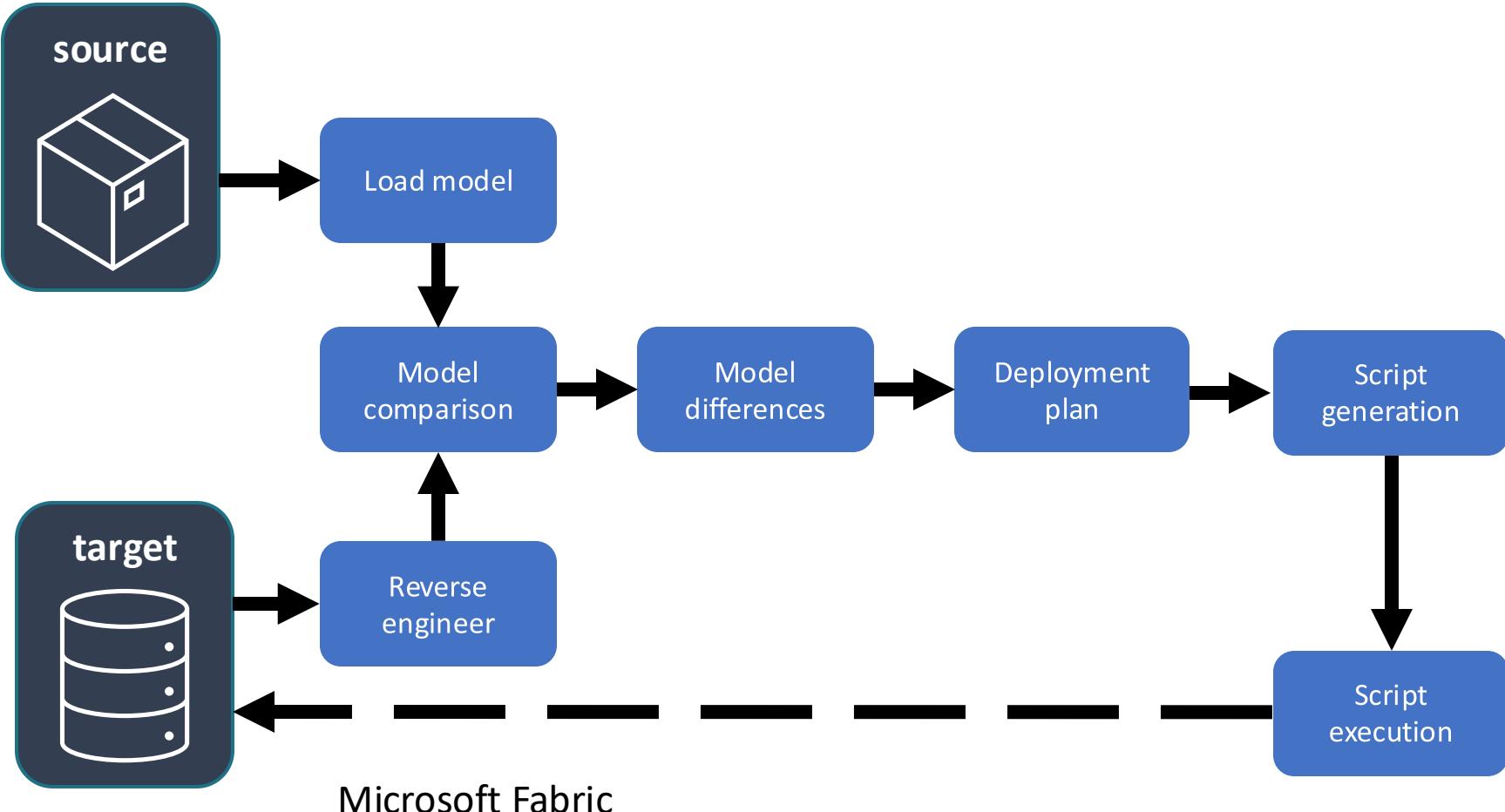
SqlPackage extract/publish



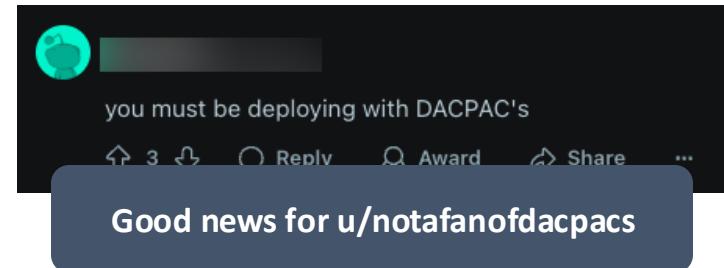
extract



Dacpac deployment behind the scenes



Deployment surprises



De-risk SqlPackage deployments with:

1. Script (outputs .sql) for review
2. Deploy report, enabling automatic actions

CI/CD tooling

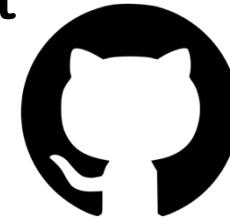
CLIs

- SqlPackage
- Sqlcmd
- ---
- Dbup
- App frameworks (EF Core)



Runtime hosts

- GitHub actions
- Azure DevOps pipelines
- ---
- Gitlab
- Bitbucket

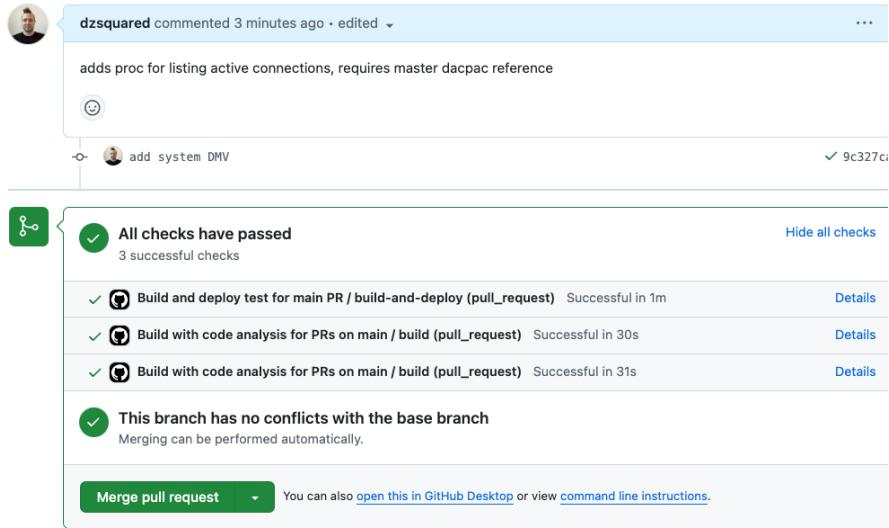


Managed environments

- Fabric



Continuous integration (CI)



A screenshot of a GitHub pull request page. At the top, a comment from user 'dzsquared' is shown, followed by a commit message 'adds proc for listing active connections, requires master dacpac reference'. Below this is a diff hunk 'add system DMV'. On the left, there's a green 'checks' icon with a checkmark. A summary box says 'All checks have passed' with '3 successful checks'. It lists three build steps: 'Build and deploy test for main PR / build-and-deploy (pull_request)', 'Build with code analysis for PRs on main / build (pull_request)', and 'Build with code analysis for PRs on main / build (pull_request)'. All three are marked as successful. Below this, a note says 'This branch has no conflicts with the base branch' and 'Merging can be performed automatically.' At the bottom, there's a 'Merge pull request' button and a note about opening it in GitHub Desktop or viewing command line instructions.



Is it valid?



Is it good?



Is it correct?

- Leveraging automation in development processes to improve quality during collaboration
- Goal: knowing if this is checked in, it meets our standards

SQL code analysis

- “out of the box” option with SQL projects includes 14 rules
 - Design
 - Naming
 - performance
- When enabled, it runs during build
 - Project property (in .sqlproj)
`<RunSqlCodeAnalysis>True</RunSqlCodeAnalysis>`
 - Build-time property
`dotnet build /p:RunSqlCodeAnalysis=true`



Extending rules

Code analysis rules are written in C#

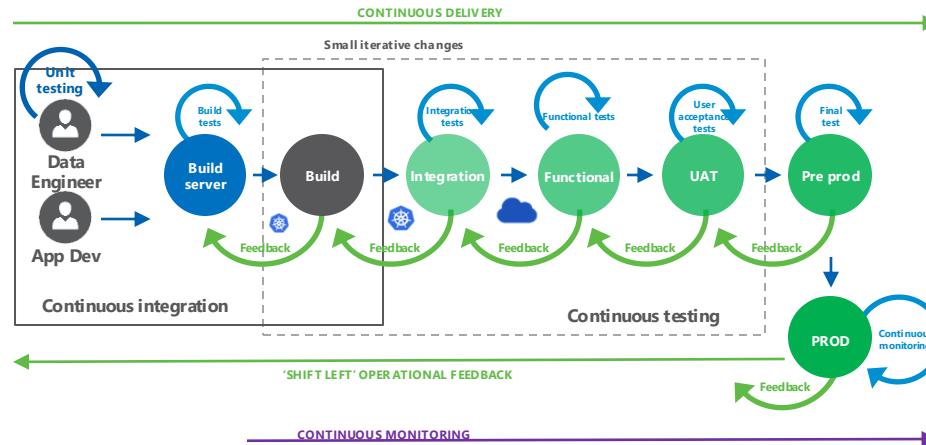
Rules have access to database model (including relationships)

Code analysis rule sets have been shared in the SQL community

Continuous deployment (CD)

- Deployment to where? (not necessarily prod)
- When code is checked in, it is deployed to *some environment* as soon as possible
- Goal: checked in code provides value quickly

CI/CD – create change continuously



How much git do you need to know?

Commit

Mark your current files as a point in the git timeline



Push

Send the local contents to a remote (like GitHub or Azure DevOps)
Branch-specific



Branch

Partition changes in different sections of the timeline

main

username/fix-1234

release/1_23



Merge/Pull request

Combine updates from one branch into another

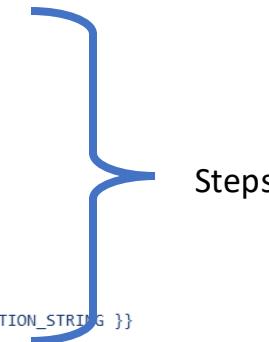


Anatomy of a pipeline

GitHub: workflow
Azure DevOps: pipeline

YAML = yet another markup language

```
1  # This workflow will build a .NET project
2  # For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-net
3
4  name: Deploy SQL project
5
6  on:
7    workflow_dispatch: } Trigger starts whole pipeline
8
9  jobs:
10   build: } Each job wraps steps + settings
11
12    runs-on: ubuntu-latest } Agent
13
14    steps:
15      - uses: actions/checkout@v4
16
17      - name: Setup .NET
18        uses: actions/setup-dotnet@v4
19        with:
20          dotnet-version: 8.0.x
21
22      - name: Restore dependencies
23        run: dotnet restore Wingtips/Wingtips.sqlproj
24
25      - name: Build
26        run: dotnet build Wingtips/Wingtips.sqlproj --no-restore
27
28      - name: Publish SQL project
29        run: |
30          SqlPackage /Action:Publish /SourceFile:Wingtips/bin/Debug/Wingtips.dacpac /TargetConnectionString:${{ secrets.SQL_CONNECTION_STRING }}
```



Steps

Pipeline anatomy

Trigger

- On demand
- On file changes
- On pull requests
- ...

Agent

- Cloud hosted
 - Windows or Linux
 - Comes with preset software
 - You can install software during pipeline
 - Automatically updated every couple of weeks
- Self-hosted (you host)

Steps

- Anything you can script can be a step
- Pre-created “tasks” or “actions” combine logic

Tips for success in today's labs

- Ask for help
- Remember you're learning two things at once
 - YAML syntax – double check indentation and docs
 - DevOps principles - consider the logical order of steps
- Go one step at a time

Lab 1 recap

1. Starting from an existing database, create a new project
2. Setup a publish pipeline such that new changes can be deployed (CD)
3. Setup a code analysis pipeline to help check new changes (CI)
4. Run a database change through the CI/CD

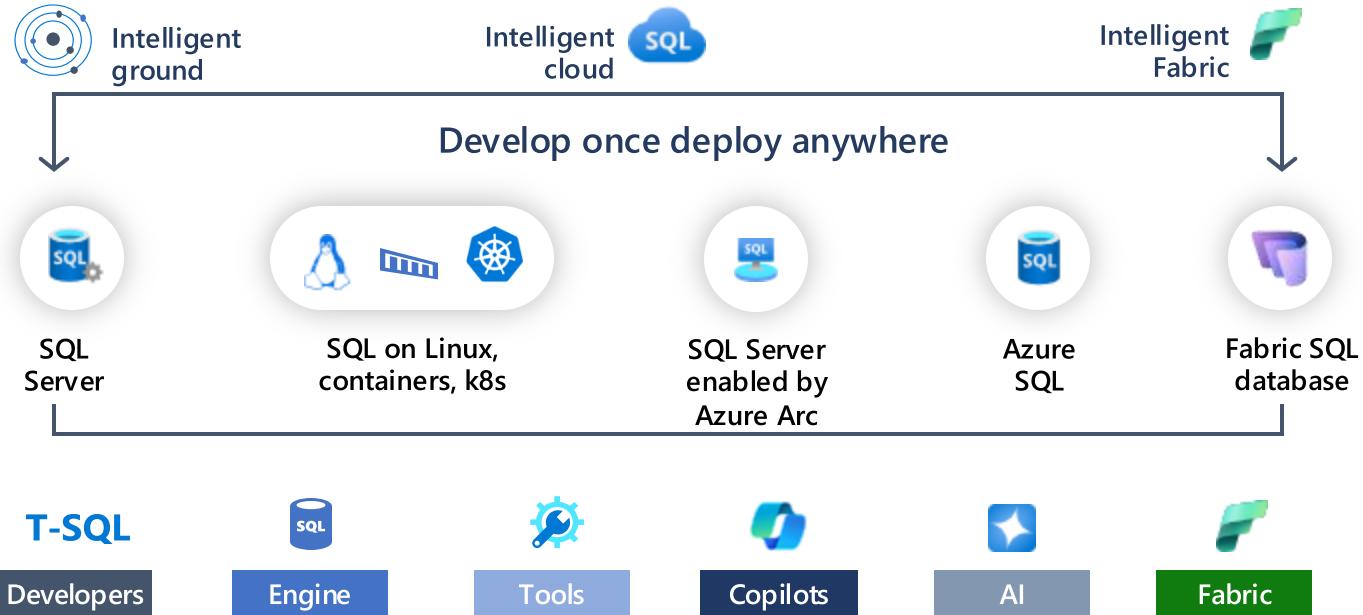
Lab One



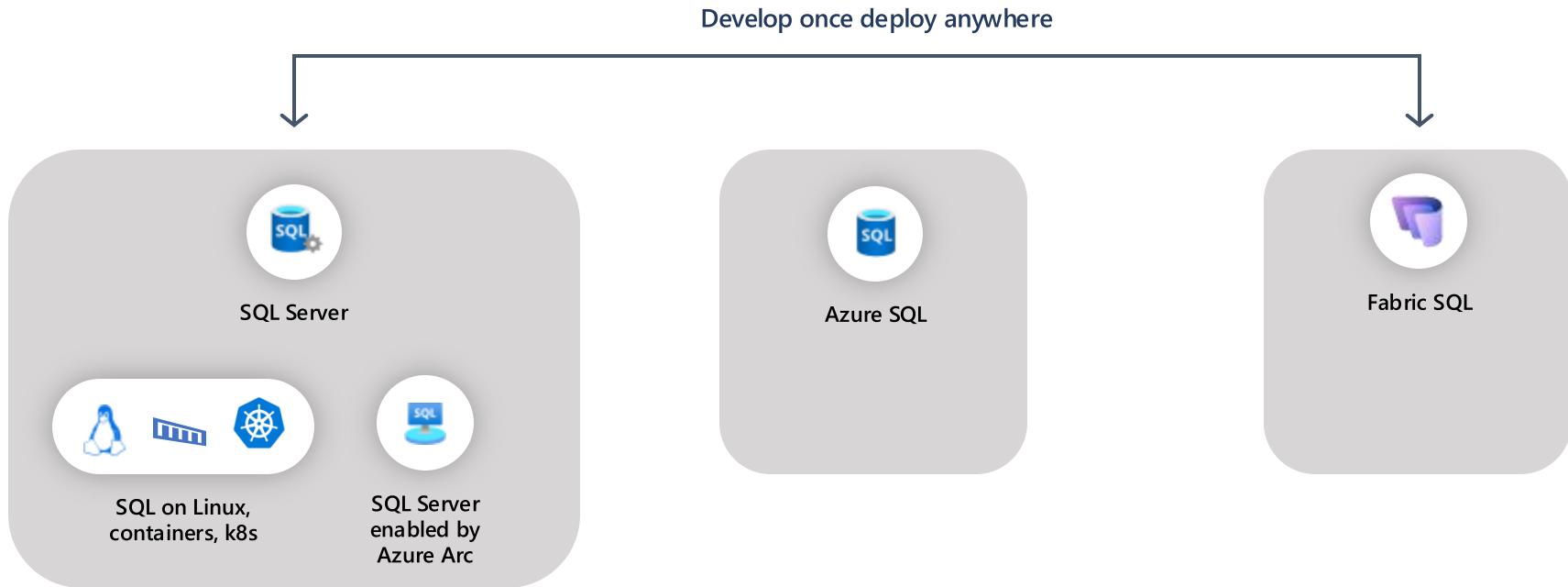
SQL projects across SQL variants



Microsoft SQL - a whole range

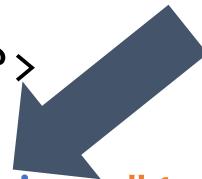
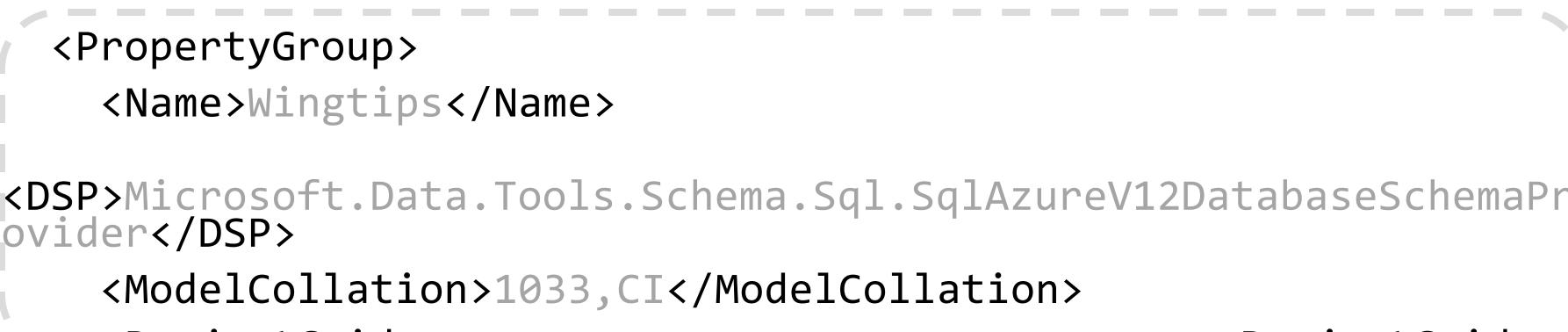


Grouping for SQL projects



Project properties

```
<?xml version="1.0" encoding="utf-8" ?>
<Project DefaultTargets="Build" >
    <Sdk Name="Microsoft.Build.Sql" Version="1.0.0" />
    <PropertyGroup>
        <Name>Wingtips</Name>
        <DSP>Microsoft.Data.Tools.Schema.Sql.SqlAzureV12DatabaseSchemaProvider</DSP>
        <ModelCollation>1033,CI</ModelCollation>
        <ProjectGuid>{00000000-0000-0000-0000-000000000000}</ProjectGuid>
    </PropertyGroup>
</Project>
```



Controls what features are available

Change tracking, query store, ledger, collation, etc

Target platform

- What syntax is valid at build time?
- Guards against accidentally attempting to deploy a dacpac against a potentially incompatible engine
- Microsoft.Data.Tools.Schema.Sql.?
 - SqlAzureV12DatabaseSchemaProvider
 - Sql160DatabaseSchemaProvider
 - SqIDwDatabaseSchemaProvider
 - SqIDwUnifiedDatabaseSchemaProvider
 - SqIDbFabricDatabaseSchemaProvider
- Publish to a different target platform with SqlPackage /p:AllowIncompatiblePlatform =true
- Build for a different target platform without updating the project file with dotnet build /p:DSP=*dspvalue*

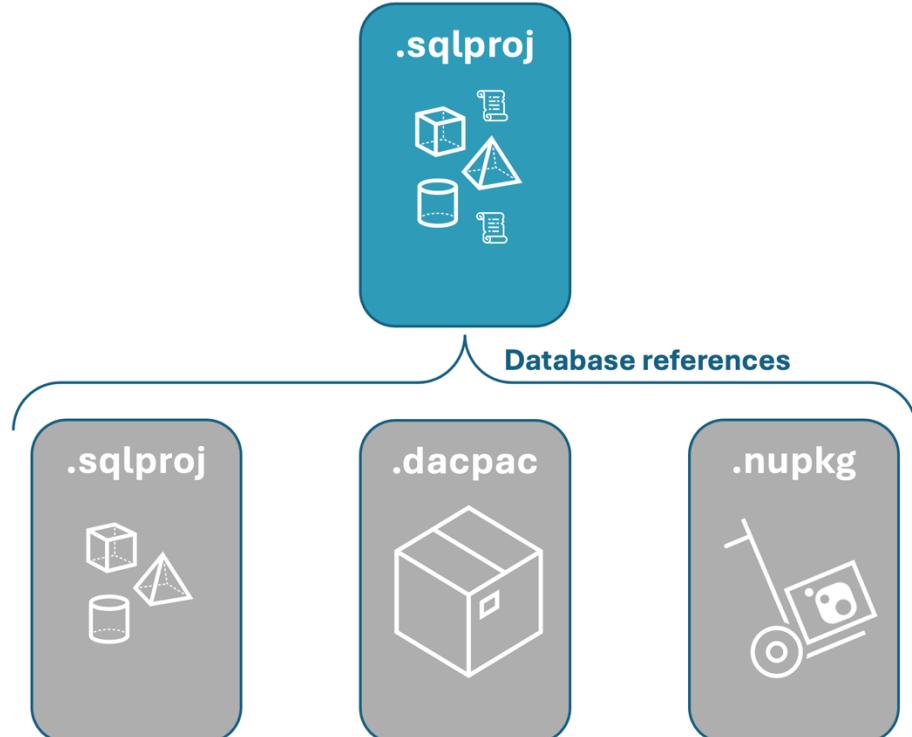
Curious about migration T-SQL compatibility?



SQL Server

SQL projects resolve references to additional databases through **Database references**

- Same database (more objects for that database) 
- Different database, same server (3-part naming) 
- Different database, different server (4-part naming for linked servers) 



Database reference types

Project references

- All files must be present for every build
- Mono-repo or consolidated repos only
- Quickly iterate
- Longer build times for outer project

Dacpac references

- Dacpac file must be copied to build location (including in pipelines)
- Can be difficult to verify you have the right dacpac and consistently use the same path
- Since dacpac is already compiled, build times are improved

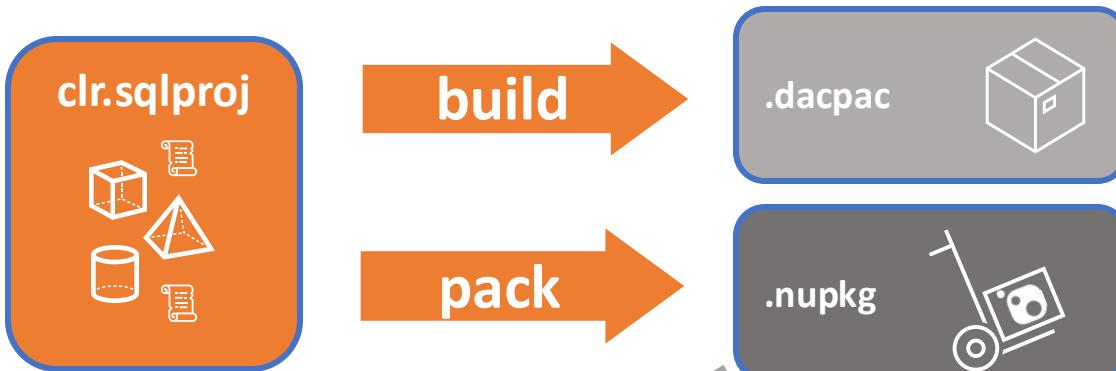
Package references

- “`dotnet pack`” creates a dacpac and packages it into a nupkg
- Referenced packages are automatically downloaded from the “package feed” where they were published
- Packages are referenced by version

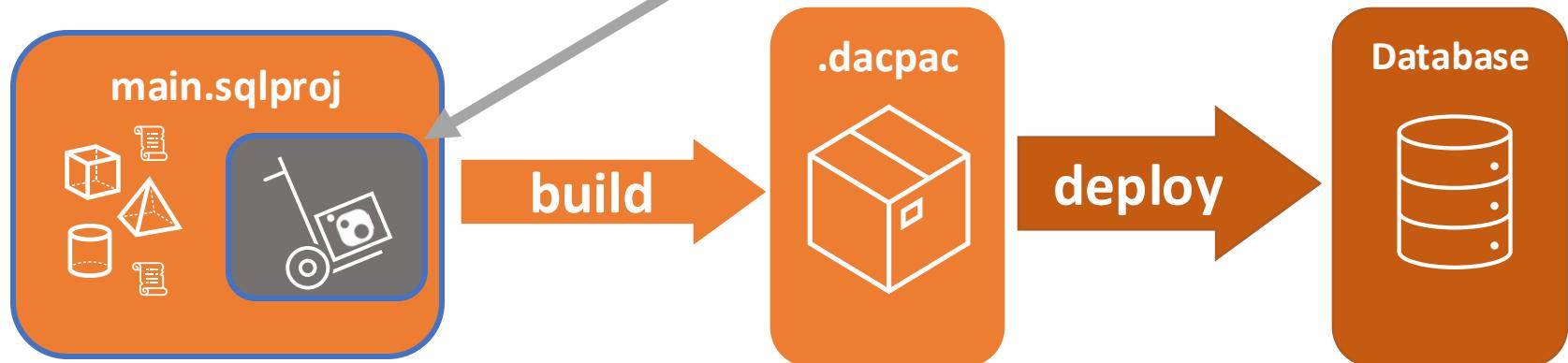
SQL Server - CLR objects

- SQL Server uses .NET Framework for CLR objects
- .NET Framework must be used to build the SQL project directly containing the CLR objects

.NET Framework



.NET



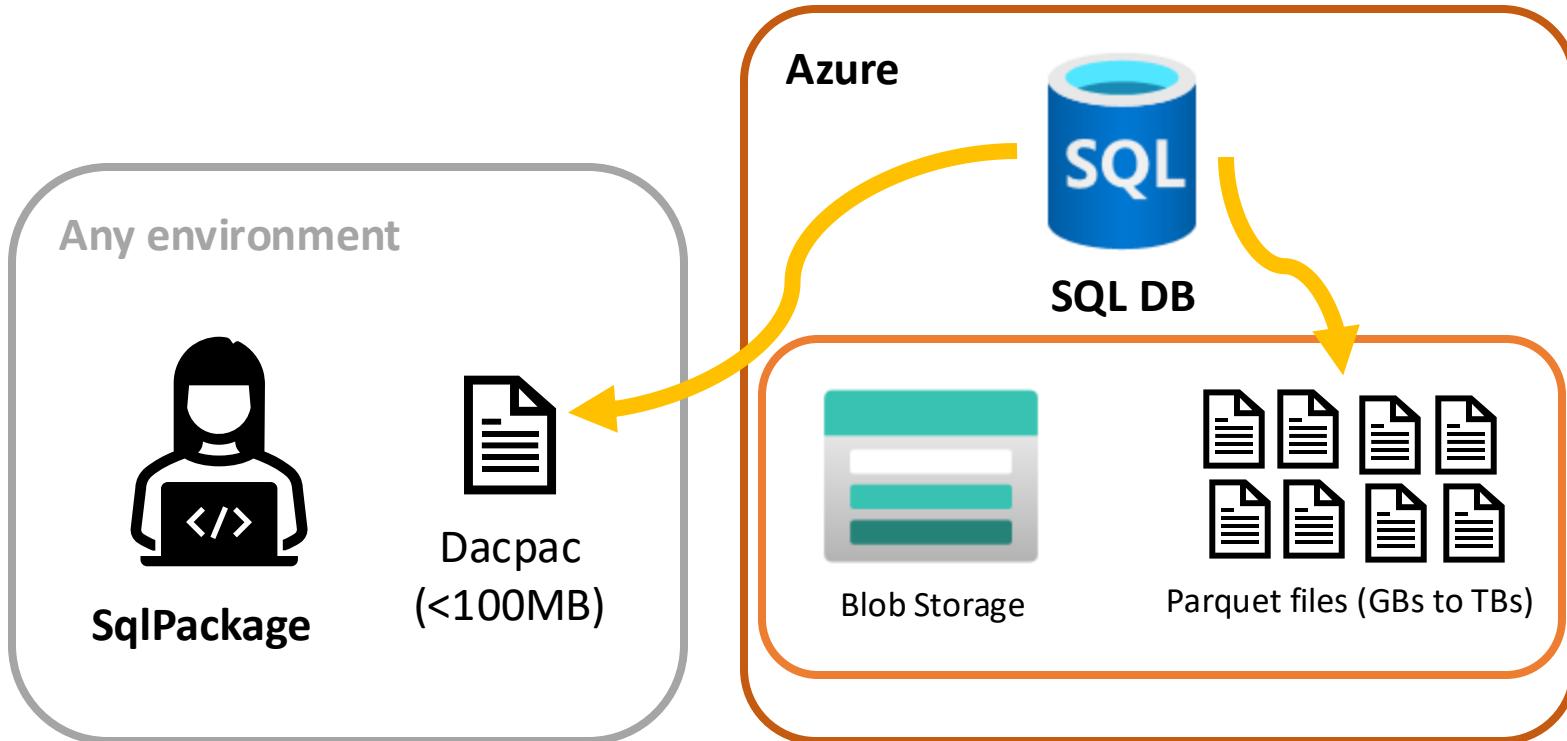
Azure SQL Database

- Reference master or system objects with master dacpac references
- 3-part naming not supported beyond system objects in the engine
- Database references can only be used for same-database references

Move data around with SqlPackage

- Bacpac contains data in contained bcp files by default
- Dacpac can include data in contained bcp files by default
- Dacpac can reference data in parquet files in Azure Blob Storage

Export (extract) to parquet in a fraction of the time



Azure SQL MI

“Always up to date” versionless

- Use Azure SQL DB target platform
- Allowed syntax will change over time

Version-pinned Azure SQL MI

- Use SQL Server target platform
- Syntax is limited to capabilities of the SQL Server version pinned

Both support 3-part naming and same-server database references

Fabric and SQL projects

Data warehouse

- Analytics, big big data
- Tons of aggregation queries



SQL database in Fabric

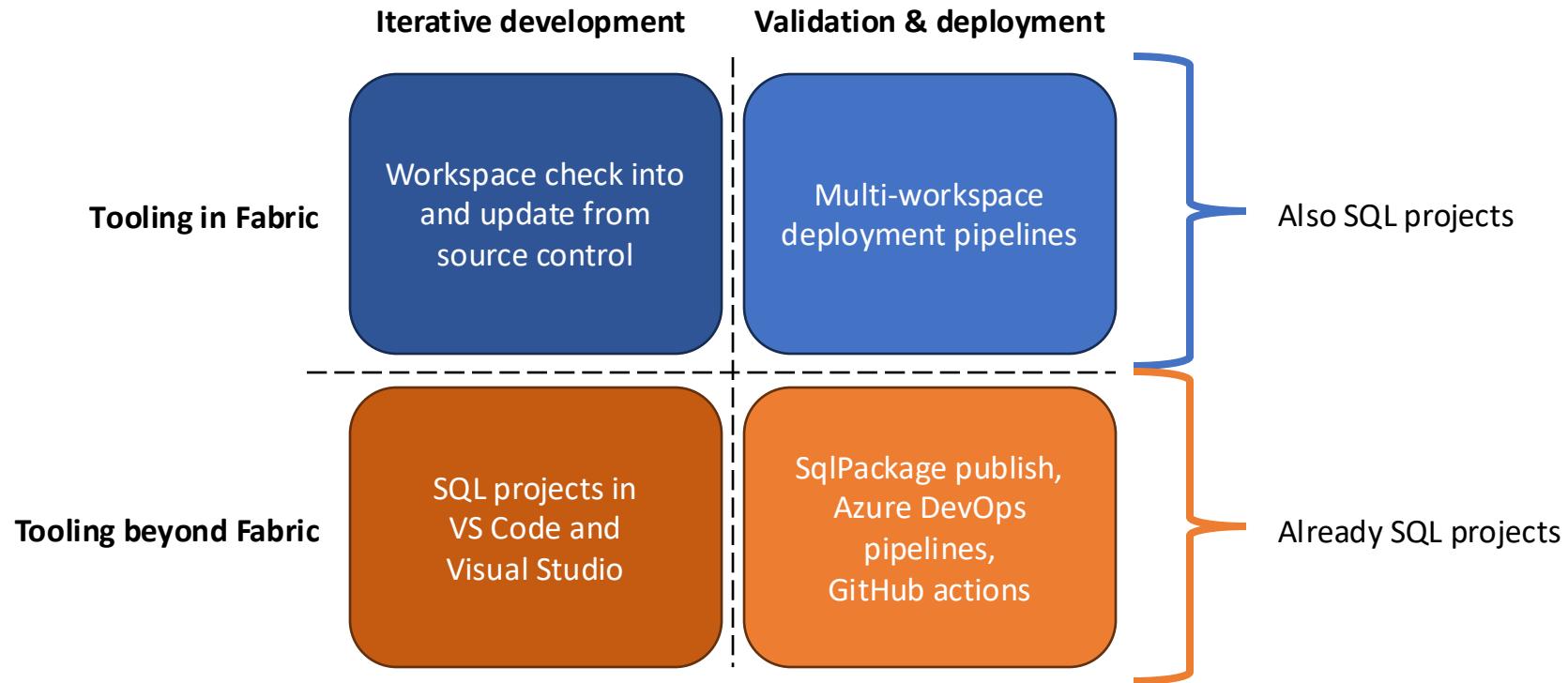
- OLTP, high performance CRUD
- Some analytical workloads
- (SQL Server, Azure SQL DB)
- No sql authentication



Lakehouse

- (Views)

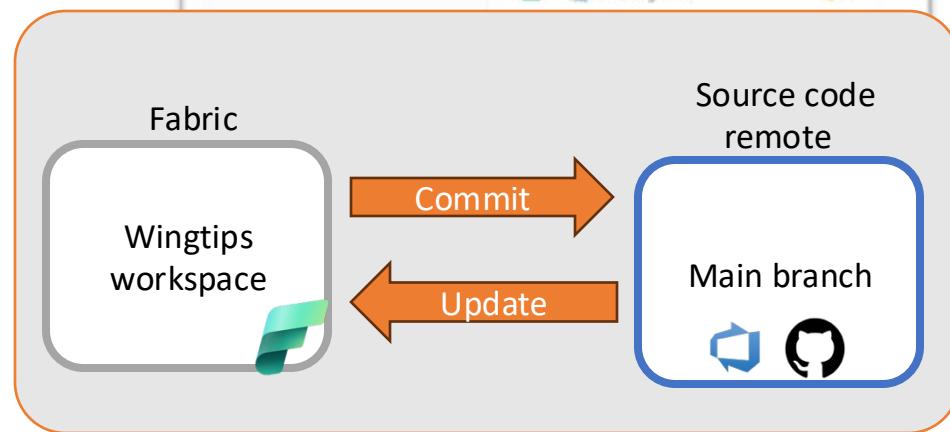
Fabric platform lifecycle management



Fabric source control integration

- GIT integration at workspace level
 - Commit workspace item definitions to source control
 - Update workspace items from source control definitions
- You can layer with deployment mechanisms
 - Fabric deployment pipelines
 - ADO Pipelines / GitHub actions

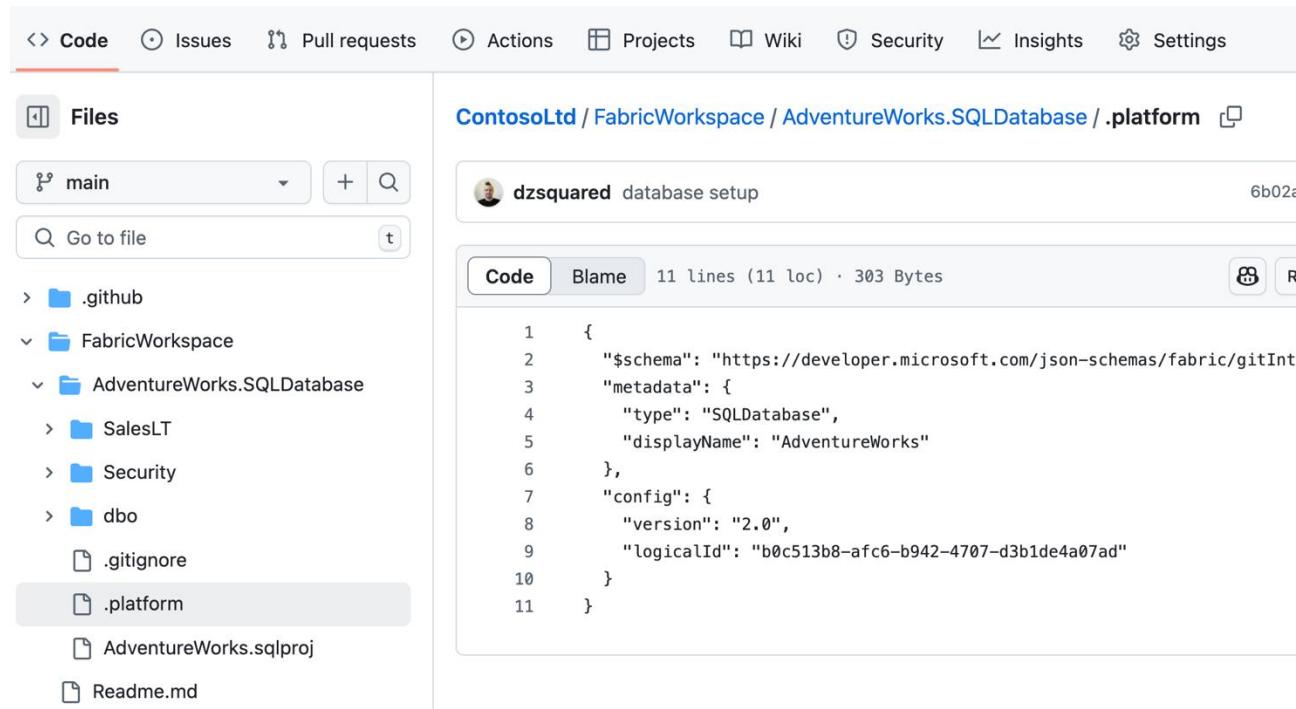
The screenshot shows the Microsoft Fabric Source Control interface. At the top, there are navigation links for 'Create app', 'Manage access', and 'Workspace settings'. Below that is a search bar with 'Filter by keyword' and a 'Source control' button. The main area is titled 'Source control' and shows 'Current branch: main'. It lists two items: 'Changes' (with a red dot) and 'Updates'. Under 'Changes', there is a card for 'early development work'. The 'Updates' section shows 'Item' and 'MessagesDB' with green checkmarks. On the right side of the interface, there are buttons for '... G X' and a refresh icon.



What's being written to source control?

The “item definition”

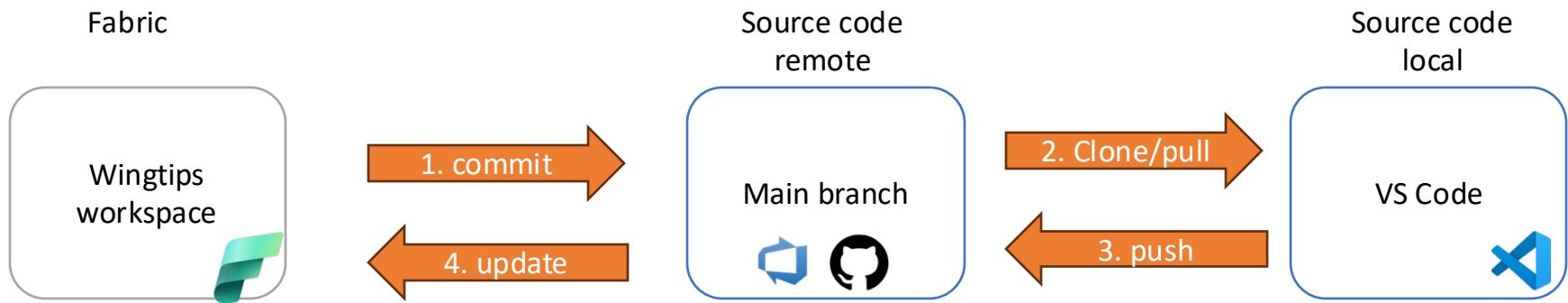
- **.platform metadata file**
- **SQL project**
- ***Shared queries***



A screenshot of a GitHub repository interface. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main area shows a tree view of files under the ".platform" folder. The ".platform" file is highlighted with a light gray background. The code content of the ".platform" file is displayed below:

```
1  {
2      "$schema": "https://developer.microsoft.com/json-schemas/fabric/gitItemDefinition.schema.json",
3      "metadata": {
4          "type": "SQLDatabase",
5          "displayName": "AdventureWorks"
6      },
7      "config": {
8          "version": "2.0",
9          "logicalId": "b0c513b8-afc6-b942-4707-d3b1de4a07ad"
10     }
11 }
```

Leverage source control integration





Workspaces



...



ContosoLtd

[Create deployment pipeline](#)[Create app](#)[Manage access](#)[Workspace settings](#)[+ New item](#)[New folder](#)[Import](#)[Source control](#)[Filter by keyword](#)[Filter](#)

	Name	Git status	Type	Task	Owner	Refreshed
	AdventureWorks	Uncommitted	SQL database	—	Drew Skwiers-Kob...	—
	AdventureWorks	—	Semantic model ...	—	ContosoLtd	3/29/2025, 8:01
	AdventureWorks	—	SQL analytics end...	—	Drew Skwiers-Kob...	—



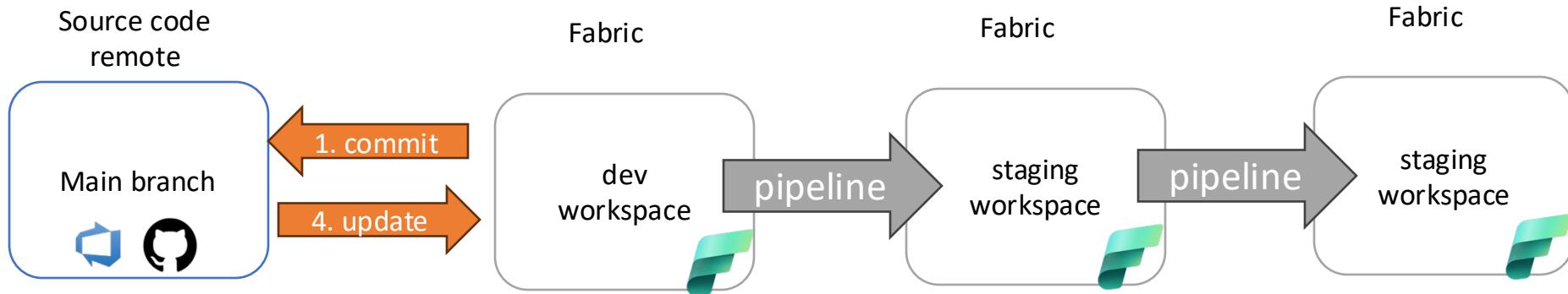
main



Last synced: 3/29/2025 at 8:05 AM 3719bacf

Fabric deployment pipelines

- Alternative to GitHub actions, Azure DevOps pipelines
- Applies definitions from one workspace to another
- Limited customization and logging
- Quick to setup
- Provide object-level view of what has changed





ContosoLtd

[+ New item](#)[New folder](#)[Import](#)[Create deployment pipeline](#)[Create app](#)[Manage access](#)[Workspace settings](#)[Source control](#)[Filter by keyword](#)[Filter](#)

	Name	Git status	Type	Task	Owner	Refreshed
	AdventureWorks	Synced	SQL database	—	Drew Skwiers-Kob...	—
	AdventureWorks	—	Semantic model ...	—	ContosoLtd	3/29/2025, 8:01
	AdventureWorks	—	SQL analytics end...	—	Drew Skwiers-Kob...	—





Microsoft Fabric

All your data. In one location. Organize. Collaborate. Create.

Explore the experience



Power BI

Find insights, track progress, and make decisions faster using rich visualizations.



Databases

Create operational SQL databases seamlessly for transactional workloads.



Data Factory

Solve complex data ingestion, transformation, and orchestration scenarios using cloud-scale data movement and data transformation services.



Industry Solutions

Use out-of-the-box industry data solutions and resources.



Real-Time Intelligence

Discover insights from your streaming data. Quickly ingest, index, and partition any data source or format, then query the data and create visualizations. You can also create alerts to flag anomalies.



Data Engineering

Create a lakehouse and operationalize your workflow to build, transform, and share your data estate.



Data Science

Unlock powerful insights using AI and machine learning technology.



Data Warehouse

Scale up your insights by storing and analyzing data in a secure SQL warehouse. Benefit from top-tier performance at petabyte scale in an open-data format.

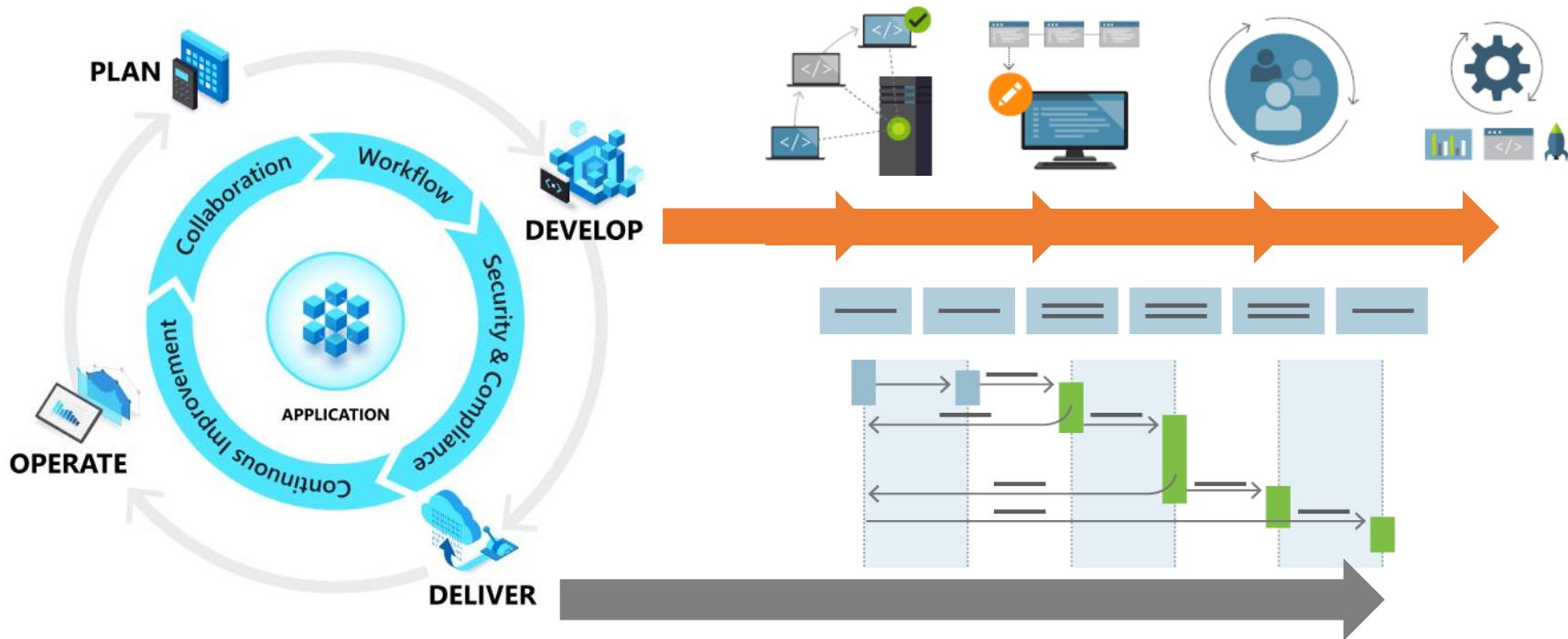


Read documentation



Explore community

Introduction to Continuous Integration & Continuous Delivery (CI/CD)



Fabric SQL DB CI/CD goals

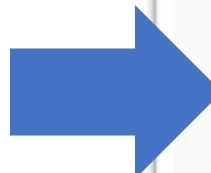
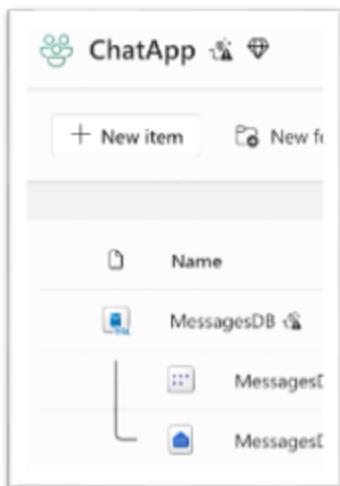


1. Get every database under source control
2. Make development databases self-service and disposable
3. Offer a low-friction mechanism to perform deployments and code quality checks without learning a new tool
4. Keep integration points open to more advanced CI/CD tools and workflows

Fabric - commit to source control

- Develop directly in the database
- Quickly iterate in early development
- Keep track of changes (commit along the way)

The screenshot shows the Fabric interface with the "Source control" tab selected. The top navigation bar includes links for "Create deployment pipeline", "Create app", "Manage access", and "Workspace settings". Below the navigation is a toolbar with "Source control" (marked with a red badge), "Filter by keyword", "Filter", and other icons. The main area displays a table with columns "Next refresh" and "Endorse". A row shows "1:53..." under "Next refresh" and "N/A" under "Endorse". To the right, a sidebar titled "Source control" shows the "Current branch: main". Under "Changes" (marked with a red badge), there is a list item "early development work". Below the changes list are sections for "Item" and "MessagesDB".



App Demo / Repos / Files / Chat App Demo

Chat App Demo

- ✓ DatastoreWorkspace
- ✓ MessagesDB.SQLDatabase
- ✓ chatter
 - StoredProcedures
 - checkUserToken.sql
 - GetCurrentlyActivePlaces.sql
 - getLatestMessages.sql
 - GetUserInfo.sql
 - joinPlace.sql
 - newMessage.sql
 - newUser.sql
 - RevokeMessage.sql
 - updateUser.sql
 - Tables
 - Messages.sql
 - PlaceMembership.sql
 - Places.sql
 - TextAnalysis.sql
 - Users.sql
 - Security
 - .platform
 - MessagesDB.sqlproj

File main / DatastoreWorkspace / MessagesDB.SQLDatabase / chatter / Tables / Messages.sql

Messages.sql

Contents History Compare Blame

```
1 CREATE TABLE [chatter].[Messages] (
2     [MessageId] INT IDENTITY (1, 1) NOT NULL,
3     [UserId] UNIQUEIDENTIFIER NOT NULL,
4     [PlaceId] INT NOT NULL,
5     [CreatedDateTime] DATETIME2 (7) NOT NULL,
6     [IsActive] BIT NOT NULL,
7     [MessageContents] NVARCHAR (MAX) NOT NULL,
8     PRIMARY KEY CLUSTERED ([MessageId] ASC),
9     CONSTRAINT [FK_Messages_PlaceId] FOREIGN KEY ([PlaceId]) REFERENCES [chatter].[Places] ([PlaceId]),
10    CONSTRAINT [FK_Messages_UserId] FOREIGN KEY ([UserId]) REFERENCES [chatter].[Users] ([UserId])
11 );
12
13
14 GO
15
16 CREATE NONCLUSTERED INDEX [IX_Messages_PlaceId_IsActive_CreatedDateTime]
17     ON [chatter].[Messages]([PlaceId] ASC, [IsActive] ASC, [CreatedDateTime] ASC);
18
19
20 GO
21
22
```

File main / DatastoreWorkspace

DatastoreWorkspace

Contents History

Graph Commit

early development work
78615567 · Drew Skwiers-Koballa · Today at 12:40 PM

Committing 1 item from workspace 164f249e-b429-4351-8fd5-3c83ef37ad76
98bd0f9a · Drew Skwiers-Koballa · Today at 12:14 PM

Creating directory DatastoreWorkspace
42a5ed0 · Drew Skwiers-Koballa · Today at 12:14 PM

The screenshot shows the Microsoft SQL Server Management Studio interface. It displays a file tree for a project named "Chat App Demo" under "DatastoreWorkspace". The "Tables" node is expanded, showing the "Messages.sql" file. The content of the "Messages.sql" file is displayed, containing the creation script for the "Messages" table and a non-clustered index. A large blue arrow points from the Power Apps studio interface on the left towards this interface. On the far right, there's a small preview window for the "DatastoreWorkspace" history.

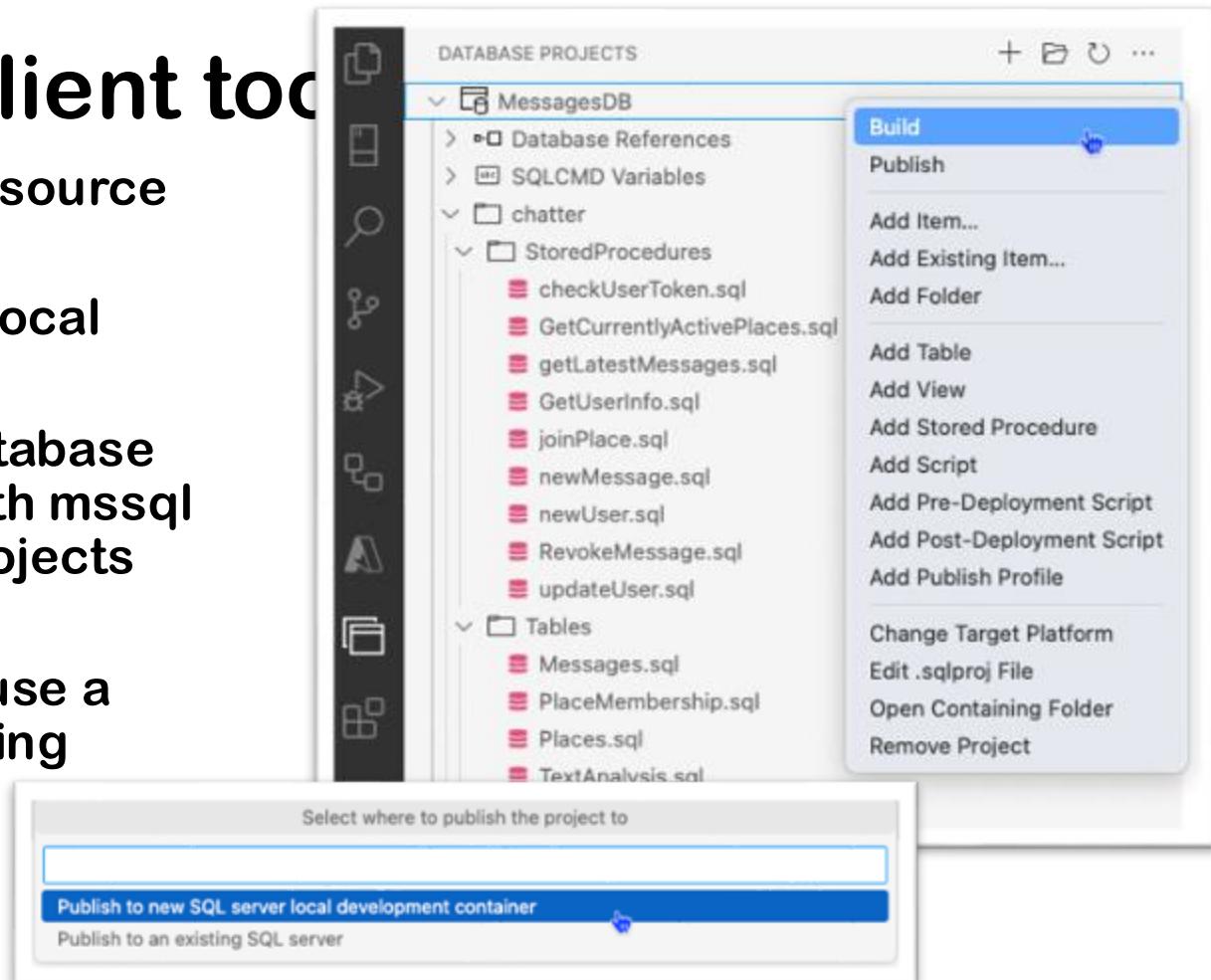
What's in the database code?

- SQL database project
- 1 file for each object
- Definitions as CREATE T-SQL statements
- Build process validates database model
- Portable and reusable artifact of database state



Develop in client tools

- Create new branch in source control
- Clone (copy) code to local machine
- Edit and build SQL database project in VS Code with mssql and SQL Database Projects extension
- Connect to Fabric or use a local instance for testing



Source control setup

1. Primary Fabric workspace where proof of concept (POC) work is taking place
2. Create an empty git repository (Azure DevOps or GitHub)
3. Connect workspace to main* branch of repo

*main or alternative branch based on your team's branching strategy

The screenshot shows the 'Git integration (Preview)' dialog box. At the top, it says 'Connect to Git to manage your code and back up your work.' with a 'Learn more' link. Below that, there's a 'Provider' section with 'Azure DevOps' selected and an 'AAD account' dropdown showing 'drskwier@microsoft.com'. A 'Log out' button is available. To the right, a 'Manage all accounts' link is shown. The main area contains fields for 'Organization' (set to 'drskwier'), 'Project' (set to 'Chat App Demo'), 'Git repository' (set to 'Chat App Demo'), 'Branch' (set to 'main'), and 'Git folder' (with an input field 'Enter name of folder'). At the bottom, there are 'Connect and sync' and 'Cancel' buttons.

Analyze

SQLDB Native Fabric ALM Test

https://msit.powerbi.com/groups/18492cb5-2b13-4ce9-b761-b52461706d06/list?experience=power-bi

Microsoft | Analyze Drew Sessions Finder

Search

Trial: 20 days left

Drew Sessions Finder

Create deployment pipeline

Create app

Manage access

Workspace settings

+ New item

New folder

Upload

Filter by keyword

Filter

Name

Type

Task

Owner

Refreshed

Next refresh

Endorsement

Sensitivity

Included in app

Name	Type	Task	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Included in app
Dataflow 1	Dataflow Gen2	—	Drew Skwiers...	11/1/2024, 9:52...	N/A	—	Confidential\...	○
find_sessions_wrapper	API for Graph...	—	Drew Skwiers...	—	—	—	Confidential\...	○
Sessions Finder Notebook	Notebook	—	Drew Skwiers...	—	—	—	Confidential\...	○
Sessions_database	SQL Database	—	Drew Skwiers...	—	—	—	Confidential\...	○
Sessions_database	Semantic mo...	—	Drew Session...	11/1/2024, 9:19...	N/A	—	Confidential\...	○
Sessions_database	SQL analytics...	—	Drew Skwiers...	—	—	—	Confidential\...	○

Home

Create

Browse

OneLake

Apps

Metrics

Monitor

Learn

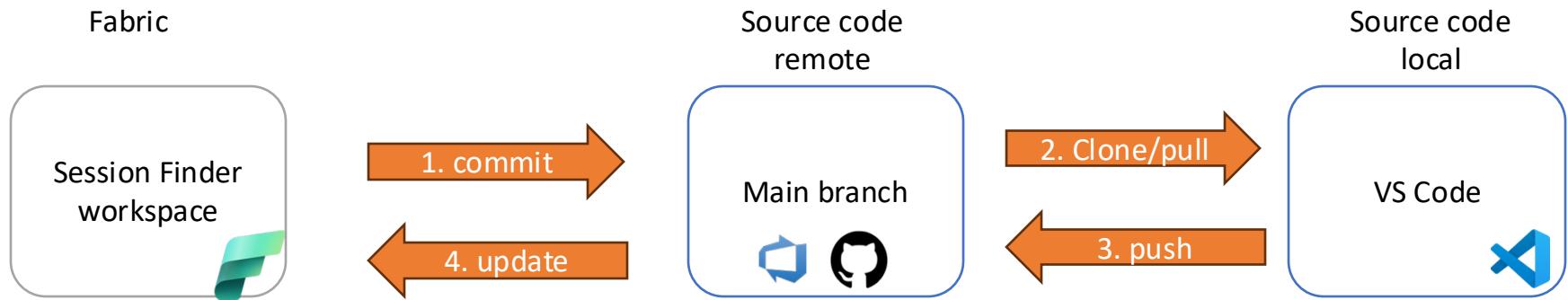
Workspaces

Drew Sessions ...

Sessions_database

Analyze

Source control integration in action



Microsoft | Analyze Drew Sessions Finder

Search

Trial: 20 days left

Create deployment pipeline Create app Manage access Workspace settings

+ New item New folder Upload Source control Filter by keyword Filter

Name	Git status	Type	Task	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Included in app
Dataflow 1	Unsupported	Dataflow G...	—	Drew Skwie...	11/1/2024, 9...	N/A	—	Confidentia...	○
find_sessions_wrapper	Synced	API for Gra...	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions Finder Notebook	Synced	Notebook	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions_database	Synced	SQL Databa...	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions_database	—	Semantic m...	—	Drew Sessio...	11/1/2024, 9...	N/A	—	Confidentia...	○
Sessions_database	—	SQL analyti...	—	Drew Skwie...	—	—	—	Confidentia...	○

Home Create Browse OneLake Apps Metrics Monitor Learn Workspaces Drew Sessions ... Sessions_database ...

Analyze

dev/sessions-finder Last synced: 11/1/2024 at 11:01 AM e72bf746

Analyze sessions_feedback.sql - Repos

https://msit.powerbi.com/groups/18492cb5-2b13-4ce9-b761-b52461706d06/list?experience=power-bi

Microsoft Analyze Drew Sessions Finder Search Trial: 20 days left

Drew Sessions Finder

Create deployment pipeline Create app Manage access Workspace settings

+ New item New folder Upload Source control Filter by keyword Filter

Name	Git status	Type	Task	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Included in app
Dataflow 1	Unsupported	Dataflow G...	—	Drew Skwie...	11/1/2024, 9...	N/A	—	Confidentia...	○
find_sessions_wrapper	Synced	API for Gra...	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions Finder Notebook	Synced	Notebook	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions_database	Synced	SQL Databa...	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions_database	—	Semantic m...	—	Drew Sessio...	11/1/2024, 9...	N/A	—	Confidentia...	○
Sessions_database	—	SQL analyti...	—	Drew Skwie...	—	—	—	Confidentia...	○

Home Create Browse OneLake Apps Metrics Monitor Learn Workspaces Drew Sessions ... Sessions_database Analyze

dev/sessions-finder Last synced: 11/1/2024 at 11:04 AM 37b14bba

analyze

sessions_feedback.sql - Repos

https://msdata.visualstudio.com/Database%20Systems/_git/SQLDB%20Native%20Fabric%20ALM%20Testing?path=/FabricWorkspace/Sessions_database.SQLDatabase dbo/Tables/sessions_feedback.sql...

Azure DevOps msdata / Database Systems / Repos / Files / SQLDB Native Fabric ALM Testing

Search

Database Systems

Overview

Boards

Repos

Files

Commits

Pushes

Branches

Tags

Pull requests

EzCab

EzCab-Test

Ownership

Pipelines

Test Plans

Artifacts

Compliance

Project settings

SQLDB Native Fabric ALM Testing

FabricWorkspace

DataflowsStagingLakehouse.Lakehouse

DataflowsStagingWarehouse.Warehouse

find_sessions_wrapper.GraphQLApi

Sessions Finder Notebook.Notebook

Sessions_database.SQLDatabase

dbo

Sequences

StoredProcedures

Tables

searched_text.sql

sessions_feedback.sql

sessions_speakers.sql

sessions.sql

speaker_profiles.sql

speakers.sql

Security

.gitignore

.platform

Sessions_database.sqlproj

Readme.md

dev/sessions-finder

Sessions_database.SQLDatabase dbo Tables sessions_feedback.sql

Edit

Contents History Compare Blame

You updated dev/sessions-finder 3m ago

Create a pull request

Full history (simplify merges) Author

Graph Commit Change Pull Request Status

Graph	Commit	Change	Pull Request	Status
	<p>improve feedback - datetime and... a359926e Drew Skwiers-Koballa Today at 11:22 AM</p> <p>adding session feedback 37b14bba Drew Skwiers-Koballa Today at 11:04 AM</p>	<p>edit</p> <p>add</p>		

Analyze Sessions_database.SQLDatabase

https://msit.powerbi.com/groups/18492cb5-2b13-4ce9-b761-b52461706d06/list?experience=power-bi

Microsoft | Analyze Drew Sessions Finder

Search

Trial: 20 days left

Create deployment pipeline Create app Manage access Workspace settings

+ New item New folder Upload Source control 1 Filter by keyword Filter

Drew Sessions Finder

Name Git status Type Task Owner Refreshed Next refresh Endorsement Sensitivity Included in app

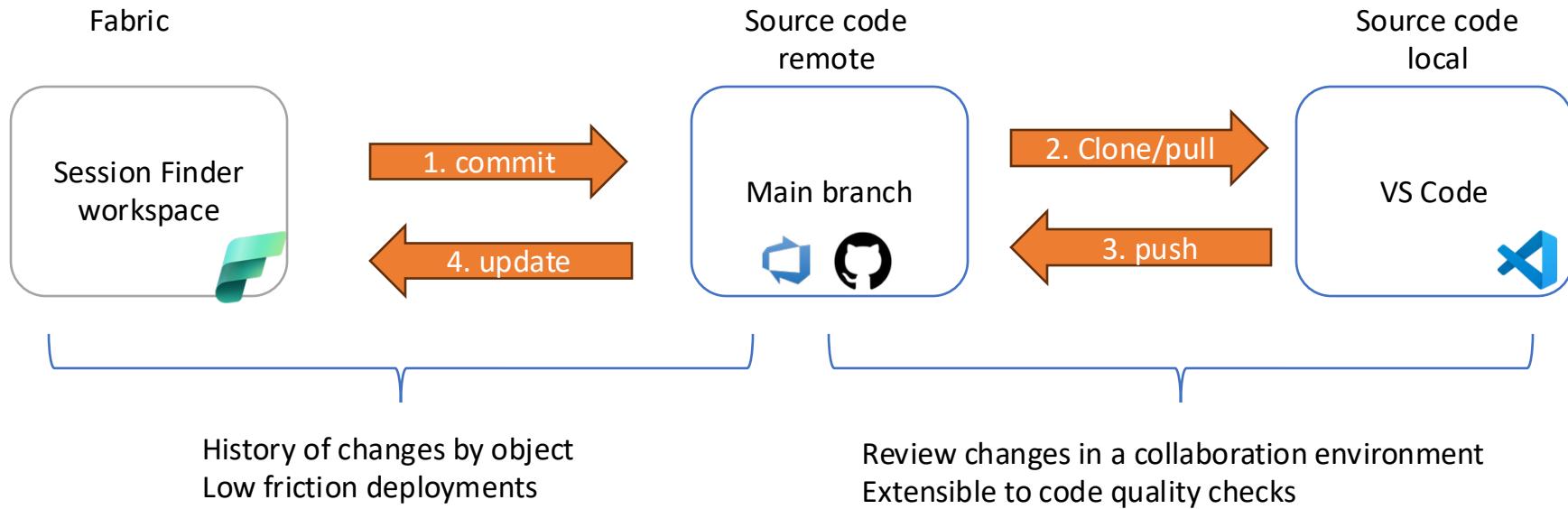
Name	Git status	Type	Task	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Included in app
Dataflow 1	Unsupported	Dataflow G...	—	Drew Skwie...	11/1/2024, 9...	N/A	—	Confidentia...	○
find_sessions_wrapper	Synced	API for Gra...	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions Finder Notebook	Synced	Notebook	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions_database	Uncommitted	SQL Databa...	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions_database	—	Semantic m...	—	Drew Sessio...	11/1/2024, 9...	N/A	—	Confidentia...	○
Sessions_database	—	SQL analyti...	—	Drew Skwie...	—	—	—	Confidentia...	○

Home Create Browse OneLake Apps Metrics Monitor Learn Workspaces Drew Sessions ... Sessions_database ...

Analyze

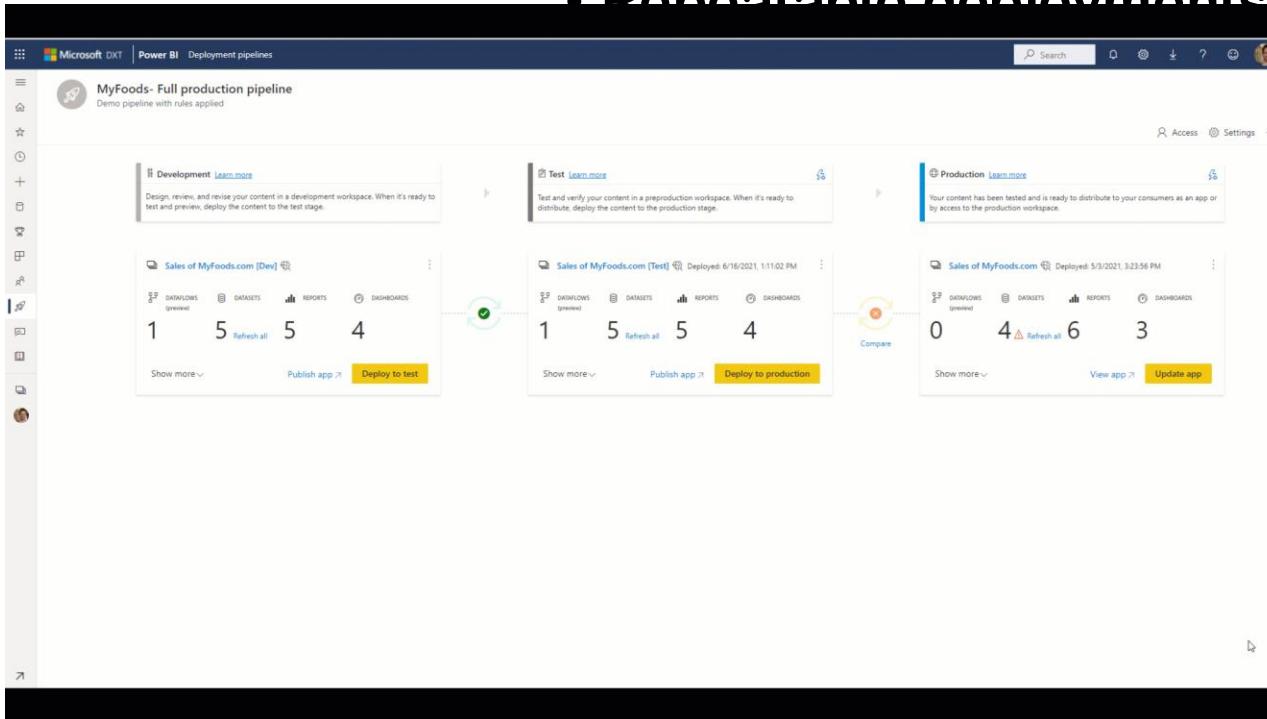
89 dev/sessions-finder Last synced: 11/1/2024 at 11:27 AM a359926e

Benefits from source control integration



Fabric Deployment pipelines

- Multi-workspace deployment process
 - Dev/Test/Prod environment automation
 - Repeatable deployments



Setup Azure DevOps identity



Azure DevOps
Pipelines



Azure

The image contains three side-by-side screenshots from the Azure portal:

- Screenshot 1:** "New service connection" dialog. It shows a list of connection types: Azure Classic, Azure Repos/Team Foundation Server, Azure Resource Manager (selected), Azure Service Bus, and Bitbucket Cloud. A search bar at the top says "Search connection types".
- Screenshot 2:** "New Azure service connection" dialog. It shows the "Authentication method" section with "Workload Identity federation (automatic)" selected (radio button is checked). Other options include Workload Identity federation (manual), Service principal (automatic), Service principal (manual), Managed identity, and Publish Profile. A "Next" button is visible at the bottom right.
- Screenshot 3:** "New Azure service connection" dialog (continued). It shows the "Scope level" section with "Subscription" selected (radio button is checked). Other options include Management Group and Machine Learning Workspace. It also shows fields for "Subscription" (drskwier-chat-app-hackathon), "Resource group" (drskwier-chat-app-hackathon), "Service connection name" (devops-demo, highlighted with a red box), and "Description (optional)".

The image shows a screenshot of the Azure portal for a service connection named "drskwier-Chat App Demo-d".

Overview:

- Display name: drskwier-Chat App Demo-d51 (highlighted with a green box)
- Application (client) ID : (empty)
- Object ID : (empty)
- Directory (tenant) ID : (empty)
- Supported account types : [My organization only](#)

Endpoints:

Preview features:

Setup Azure DevOps pipeline



Manage SQL security

X

Manage access

Add or remove users for this role.

Add people, groups or apps

drskwier-Chat App Demo-d51

drskwier-Chat App Demo-d51

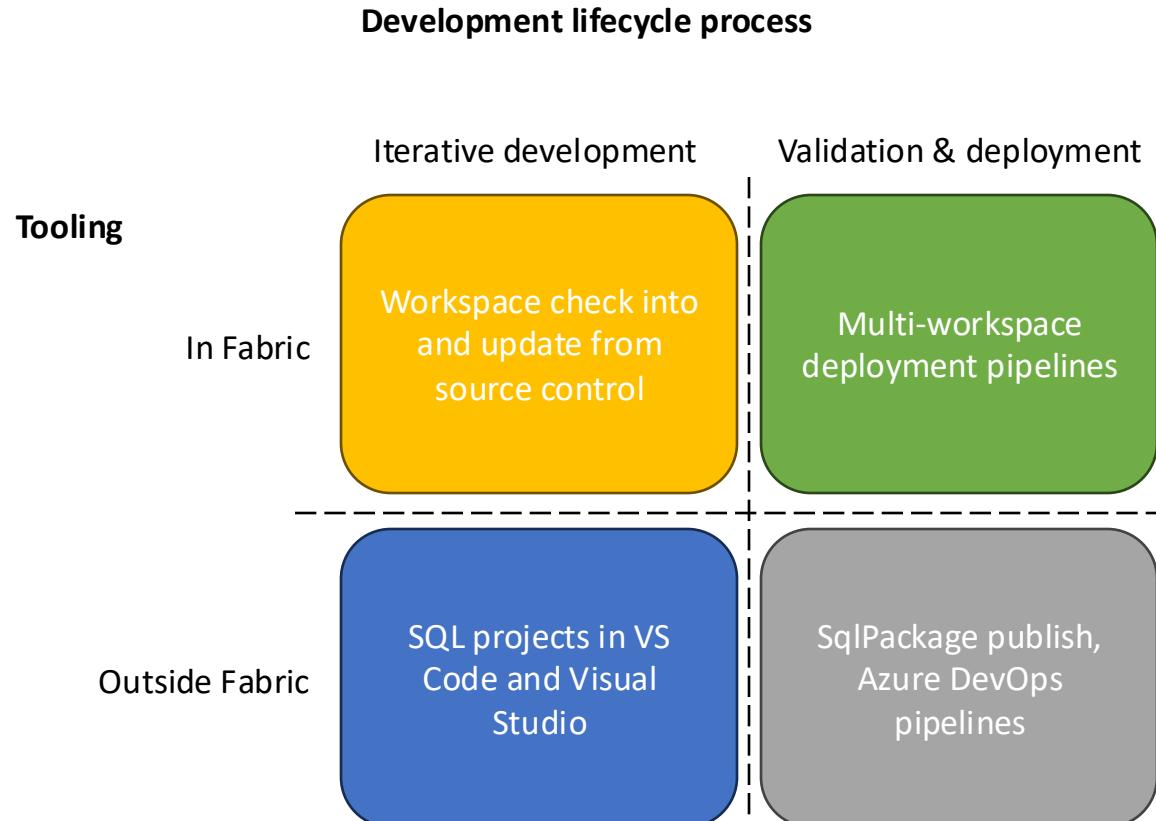
Add

Clear



```
30 # this connection settings would usually be in variables
31 Settings
32 - task: SqlAzureDacpacDeployment@1
33   inputs:
34     azureSubscription: 'devops-demo' (Red box)
35     AuthenticationType: 'aadAuthenticationIntegrated'
36     ServerName: 'fleet-srv-767c0a4b-beec-4d0a-9c10-333efe44de71-a08b9015aef7.database.windows.net'
37     DatabaseName: 'MessagesDB-91784e0c-92f1-4c30-b3c7-2e40acd5be9c'
38     deployType: 'DacpacTask'
39     DeploymentAction: 'Publish'
40     DacpacFile: '$(Pipeline.Workspace)/s/DatastoreWorkspace/MessagesDB.SQLDatabase/bin/Debug/MessagesDB.dacpac' (Red arrow pointing here)
41     IpDetectionMethod: 'AutoDetect'
42 
```

SQL database in Fabric CI/CD ecosystem



Fabric deployment pipelines



Analyze

FabricWorkspace - Repos

https://msit.powerbi.com/groups/18492cb5-2b13-4ce9-b761-b52461706d06/list?experience=power-bi

Microsoft | Analyze Drew Sessions Finder

Search

Trial: 20 days left

Create deployment pipeline Create app Manage access Workspace settings

+ New item New folder Upload Source control Filter by keyword Filter

Drew Sessions Finder

Home Create Browse OneLake Apps Metrics Monitor Learn Workspaces

Drew Sessions ... Sessions_database

Analyze

Name	Git status	Type	Task	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Included in app
Dataflow 1	Unsupported	Dataflow G...	—	Drew Skwie...	11/1/2024, 9...	N/A	—	Confidentia...	○
find_sessions_wrapper	Synced	API for Gra...	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions Finder Notebook	Synced	Notebook	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions_database	Synced	SQL Databa...	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions_database	—	Semantic m...	—	Drew Sessio...	11/1/2024, 9...	N/A	—	Confidentia...	○
Sessions_database	—	SQL analyti...	—	Drew Skwie...	—	—	—	Confidentia...	○

dev/sessions-finder Last synced: 11/1/2024 at 11:56 AM 6e98d029

Sessions_database - Analyze FabricWorkspace - Repos

https://msit.powerbi.com/groups/7e644d00-01a0-4fc8-921d-f1110312dc9f/sqldatabases/02e29660-2214-4957-b3b5-f534d191edda?experience=power-bi

Microsoft Sessions_database | Confidential\Microsoft Extended | Search Trial: 20 days left 43 SQL Database

Home Replication Security

Get data New Query Templates Open in New API for GraphQL Performance summary

Explorer

Search

> Sessions_database

Queries

Query, preview, or connect your data

Interact with your data by starting a query, creating database objects with a template, or connecting the database to your app with a GraphQL API. You can also preview your data by opening a table or view.

+ New

Home Replication Security

Create

Browse

OneLake

Apps

Metrics

Workspaces

Drew Sessions ...

Sessions_database

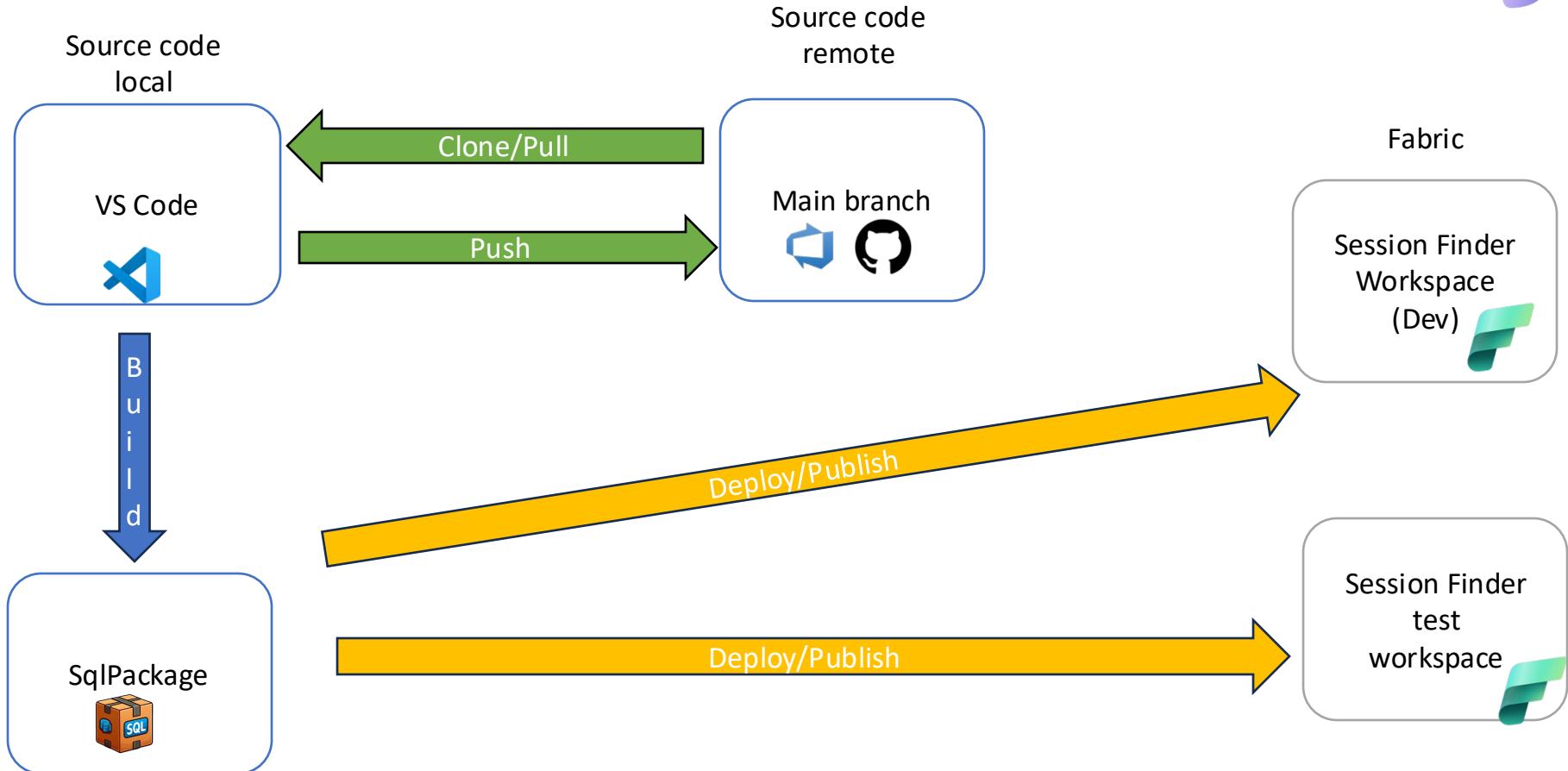
Drew Sessions ...

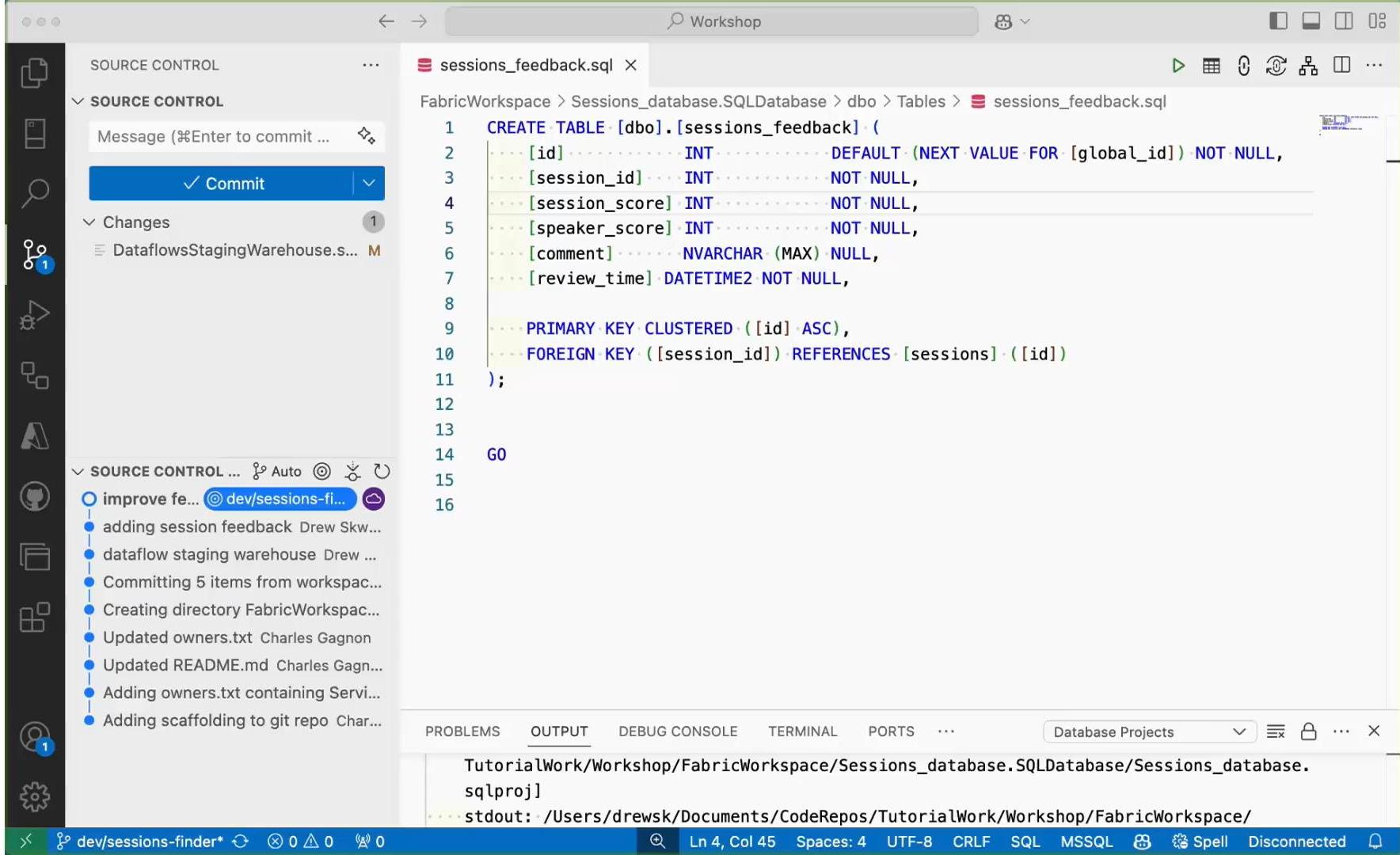
Sessions_database

... Analyze

The screenshot shows the Microsoft Fabric workspace interface. At the top, there are two tabs: "Sessions_database - Analyze" and "FabricWorkspace - Repos". The URL in the address bar is <https://msit.powerbi.com/groups/7e644d00-01a0-4fc8-921d-f1110312dc9f/sqldatabases/02e29660-2214-4957-b3b5-f534d191edda?experience=power-bi>. The top navigation bar includes "Microsoft", "Sessions_database", "Confidential\Microsoft Extended", a search bar, and a trial status message "Trial: 20 days left" with a notification badge "43". Below the top bar is a secondary navigation bar with links for "Home", "Replication", "Security", "Get data", "New Query", "Templates", "Open in", "New API for GraphQL", and "Performance summary". On the left side, there is a sidebar titled "Explorer" containing a "Search" input field and a tree view with nodes for "Sessions_database" and "Queries". The main content area features a large circular placeholder icon with two overlapping squares. Below the icon, the text "Query, preview, or connect your data" is displayed, followed by a descriptive paragraph about interacting with the data. At the bottom right of the main area is a green button labeled "+ New". The overall layout follows the Microsoft Fluent design system.

SqlPackage deployments





Analyze

FabricWorkspace - Repos

https://msit.powerbi.com/groups/18492cb5-2b13-4ce9-b761-b52461706d06/list?experience=power-bi

Microsoft | Analyze Drew Sessions Finder

Search

Trial: 20 days left

Create deployment pipeline Create app Manage access Workspace settings

+ New item New folder Upload Source control Filter by keyword Filter

Drew Sessions Finder

Name Git status Type Task Owner Refreshed Next refresh Endorsement Sensitivity Included in app

Name	Git status	Type	Task	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Included in app
Dataflow 1	Unsupported	Dataflow G...	—	Drew Skwie...	11/1/2024, 9...	N/A	—	Confidentia...	○
find_sessions_wrapper	Synced	API for Gra...	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions Finder Notebook	Synced	Notebook	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions_database	Synced	SQL Databa...	—	Drew Skwie...	—	—	—	Confidentia...	○
Sessions_database	—	Semantic m...	—	Drew Sessio...	11/1/2024, 9...	N/A	—	Confidentia...	○
Sessions_database	—	SQL analyti...	—	Drew Skwie...	—	—	—	Confidentia...	○

Home Create Browse OneLake Apps Metrics Monitor Learn Workspaces Drew Sessions ... Sessions_database Analyze

dev/sessions-finder Last synced: 11/1/2024 at 11:56 AM 6e98d029

A decorative background featuring various neon signs on a dark brick wall. On the left, there's a sign for "sqlbits 2025" with a blue-to-orange gradient. Above it are neon outlines of cherries, a flamingo, a lollipop, a lightning bolt, and several stars. On the right, there's a neon outline of a planet with a ring.

Lab Two

- Test the deployment with a SQL Server 2022 container in a pipeline
- Connect a SQL database in Fabric to a git repository

Advanced Topics



Watch out for these SQL projects pitfalls!

Managing users/roles

- Focus on managing DB roles
- Consider segmenting user assignment to roles in another cadence
- Change controls are ideally integrated into user assignment process

Schema compare gaps

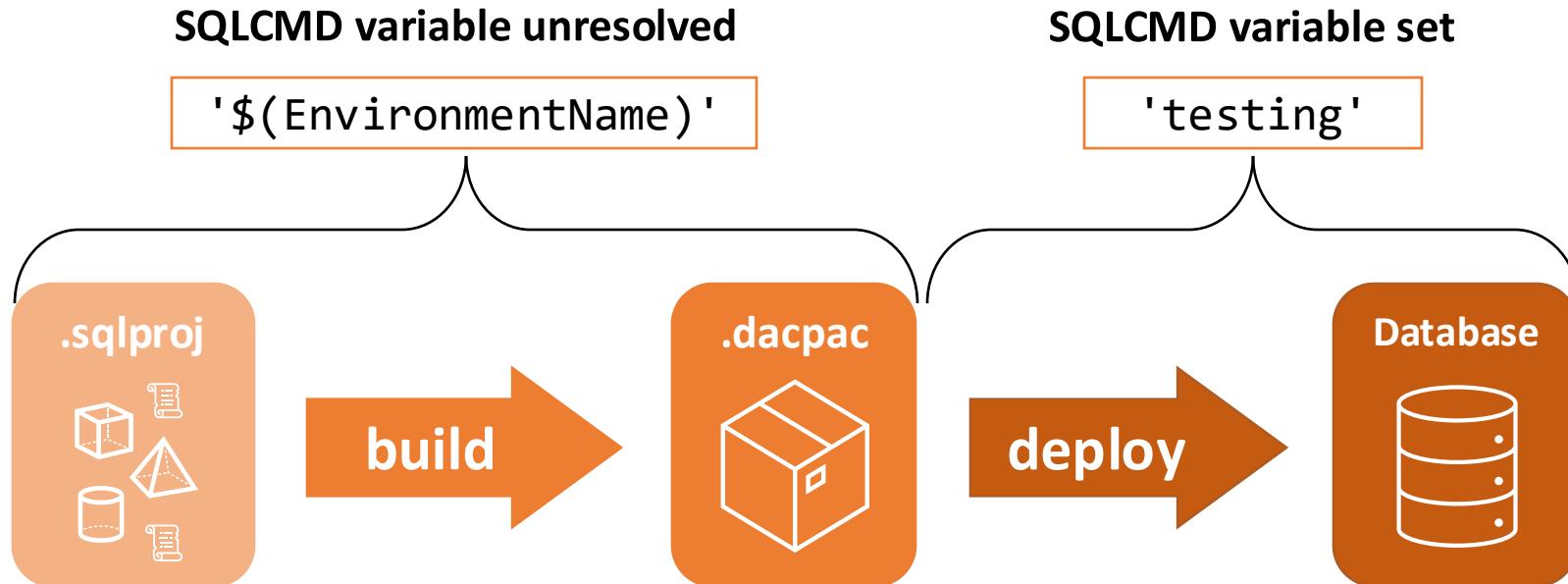
- SQL applies a default name to unnamed constraints
- Schema compare will display the object as having a difference when dealing with unnamed constraints and SQLCMD variables

Connectivity to DB

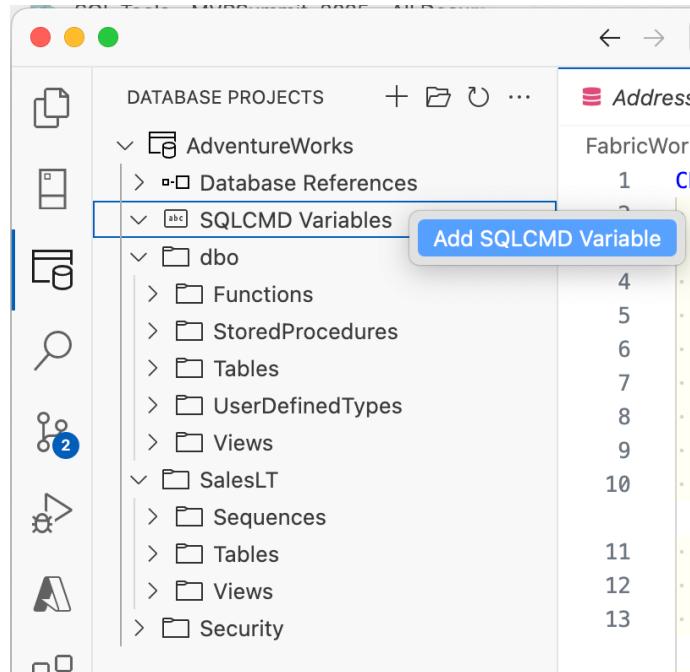
- Azure SQL DB open to all Azure services is not recommended except for early dev
- SQL Server instances usually have full network isolation
- Consider self-hosted runners for controlled environments

SQLCMD variables

- Placeholder in any SQL projects T-SQL code that is only applied during deployment
- Control IF/ELSE statements, fill in default values, set database name in 3-part naming



SQLCMD variables



.sqlproj file

```
<ItemGroup>
  <SqlCmdVariable
    Include="StudentPassword">
    <Value>$(SqlCmdVar__1)</Value>
    <DefaultValue>password</DefaultValue>
  </SqlCmdVariable>
</ItemGroup>
```

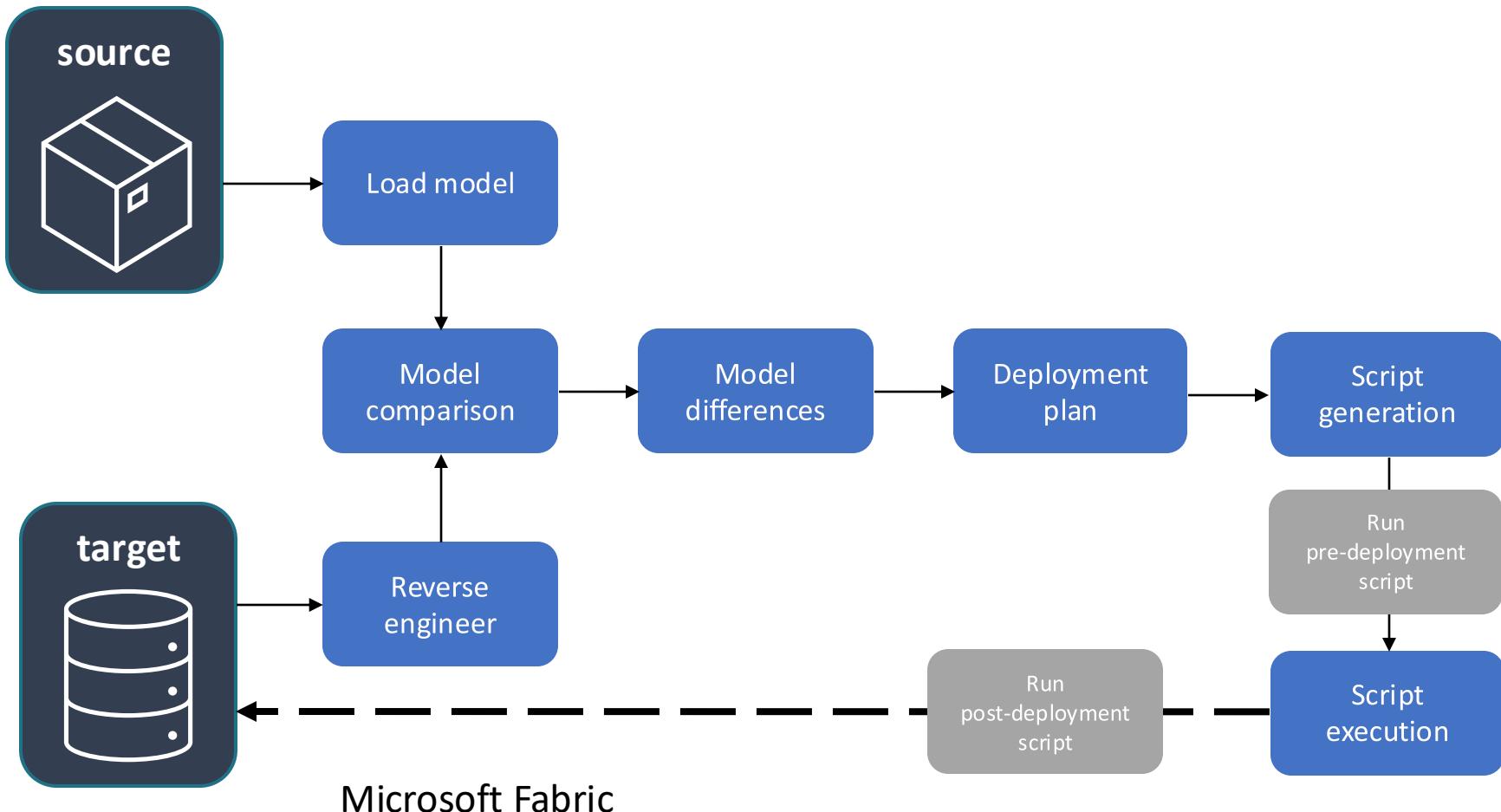
Use in scripts

```
CREATE USER [student] WITH
PASSWORD = '$(StudentPassword)';
```

Pre- and post-deployment scripts

- Non-object SQL files in a SQL project that are run with each deployment
- First line of flexibility with SQL projects to support more situations
- Partially enabled for Fabric source control integration (run once, will be removed on next commit from Fabric)

Dacpac deployment behind the scenes



Pre/post-deployment scripts in .sqlproj

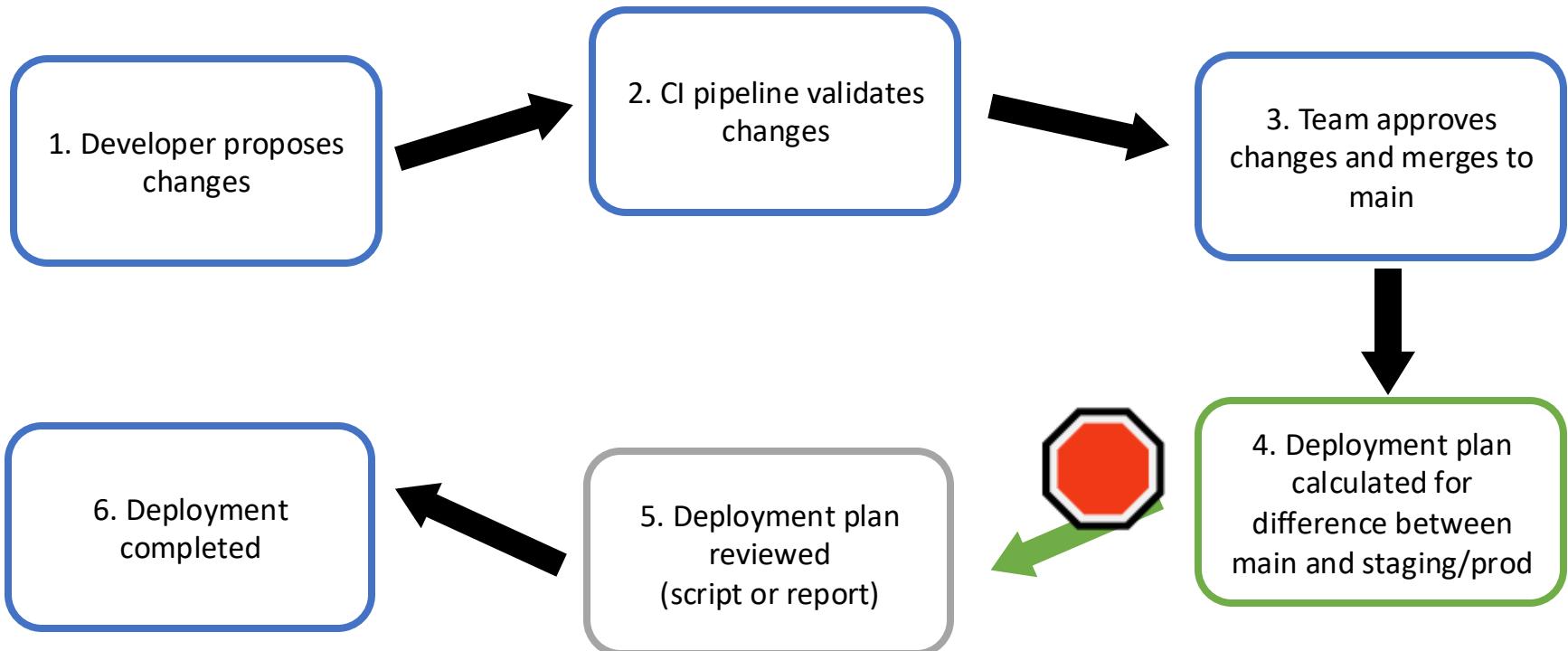
```
<?xml version="1.0" encoding="utf-8" ?>
<Project DefaultTargets="Build" >
  <Sdk Name="Microsoft.Build.Sql" Version="1.0.0" />
  <PropertyGroup>
    <Name>Wingtips</Name>

  <DSP>Microsoft.Data.Tools.Schema.Sql.SqlAzureV12DatabaseSchemaProvider</DSP>
    <ModelCollation>1033,CI</ModelCollation>
    <ProjectGuid>{00000000-0000-0000-0000-000000000000}</ProjectGuid>
  </PropertyGroup>
  <ItemGroup>
    <PreDeploy Include="MyPredeployScript.Sql" />
  </ItemGroup>
</Project>
```

Static data management

- (other than pre/post-deployment scripts...)
- BCP
- External tables
- Manual notebook execution
- Custom process with backup restore and data selection

Gate pipelines



Gate pipelines

- Automated does not have to be automatic
- Use GitHub environments or Azure DevOps approvals to gate a workflow

The screenshot shows the 'Summary' tab of an Azure DevOps pipeline named 'Deploy SQL project #4'. The pipeline has two steps: 'build' (green checkmark) and 'deploy' (yellow circle with a question mark). The 'deploy' step is currently 'Waiting' with a status message: 'dzsquared requested your review to deploy to ProductionDeployment'. Below the summary, there's a 'publish.yml' file snippet and a 'Deployment protection rules' section.

The screenshot shows the 'Checks for Deploy_WUS2' dialog in the Azure DevOps interface. It lists several checks:

- Approval**: Environment FabrikamFiber Feed WUS1 - Waiting for approval. A red box highlights the 'Approve' button.
- Approval**: Variable group AzureFunctionKeys - Waiting for approval.
- Business Hours Ren...**: Variable gr... AzureFunction... - In progress.
- Branch control**: Environment FabrikamFiber Feed WUS1 - Passed.

 The main pipeline summary shows a 'Deploy_Staging' stage in 'Waiting' status with 2 checks in progress.

Network access challenges

Any Azure allowed

- Azure SQL Database, SQL database in Fabric
- Effectively the same as public access
- Caution
- Early development loops

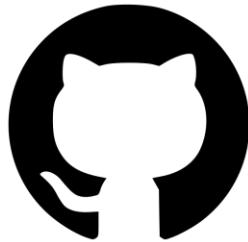
Firewall rules

- Add/remove just-in-time (JIT)
- Leverages EntraID access to Azure resource management

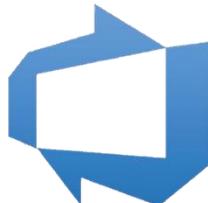
Private endpoints

- Isolated networks
- No incoming traffic allowed
- Deploy a pipeline runner in the network

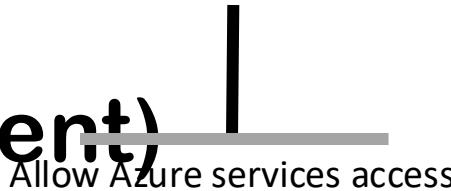
“Easy mode” (for development)



GitHub actions

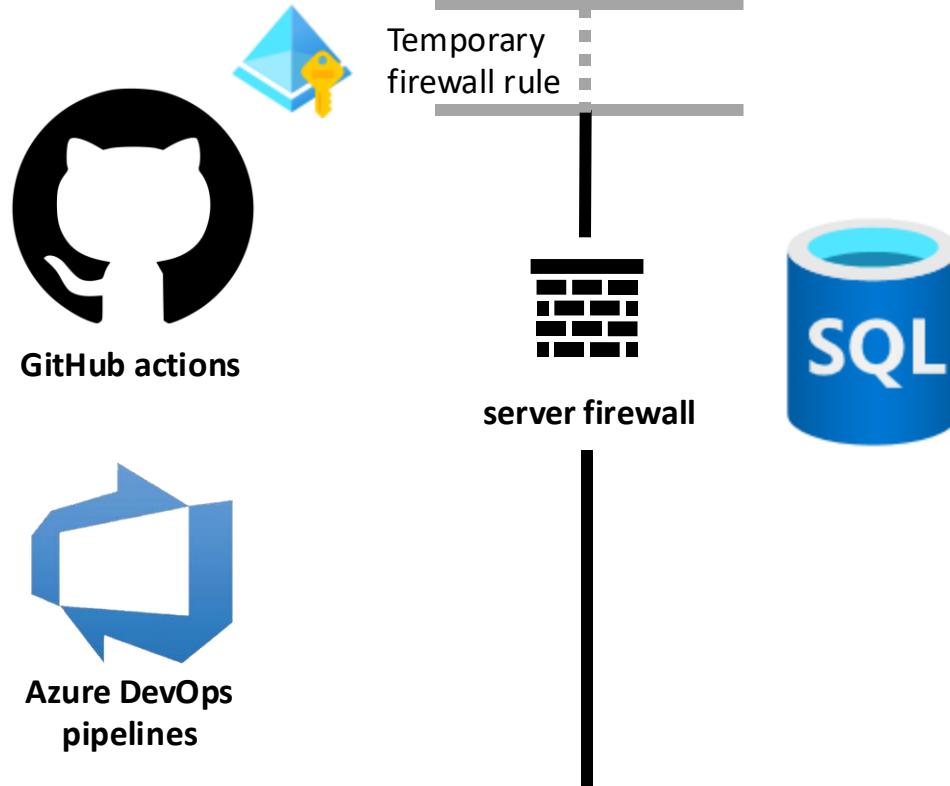


Azure DevOps
pipelines



- GitHub and Azure DevOps pipelines are hosted in Azure infra
- “Allow Azure services access” permits any pipeline to reach your database
- **Think about the consequences of that access**

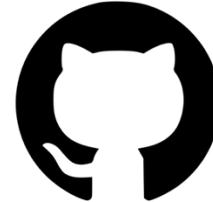
Navigate firewall



- Leverage managed identities as a best practice
- Add+remove a temporary firewall rule
- Tasks in GitHub and ADO incorporate a firewall rule for Azure SQL DB
- Use az cli or Azure PowerShell to directly modify firewall rules
- Azure SQL MI public networking can be done but is much more complex

105

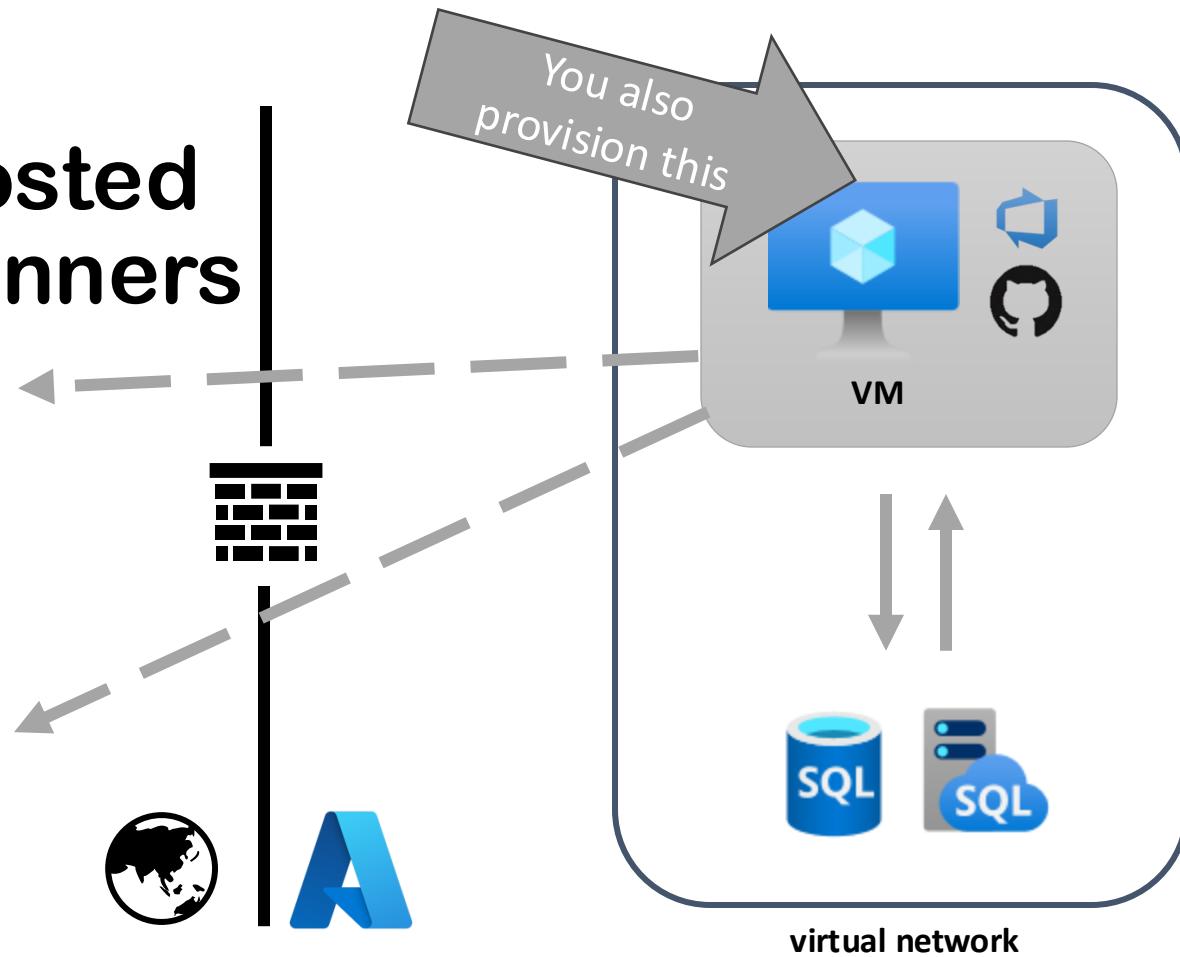
Network isolation use self-hosted pipeline runners



GitHub actions

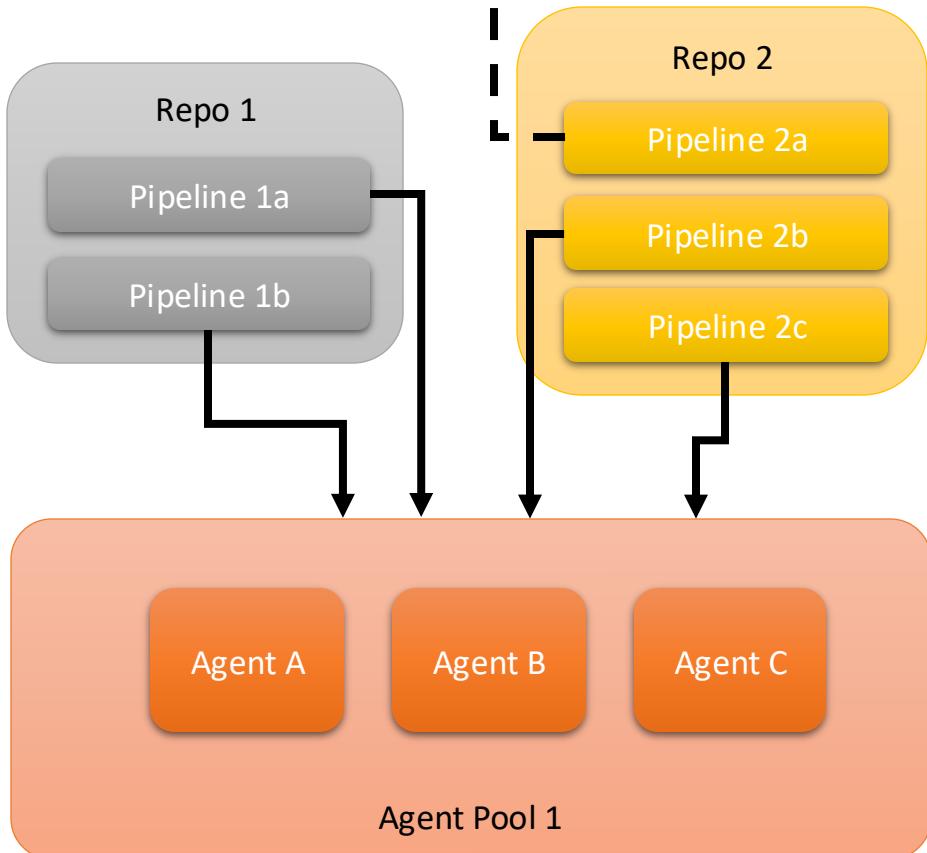


Azure DevOps
pipelines



Self-hosted automation environments

- Enables network isolation
- Runner only needs to be able to call out to check for pending jobs
- Same runner/agent can be used across many pipelines
- Groups of runners/agents are pooled to share workloads
- You manage the OS, installed software, availability, etc
- Commonly configured as VMs or containers



Add self-hosted runner to a single GitHub repo

The screenshot shows the GitHub Settings page for a repository named 'dzsquared/wingtips-devops-samples'. The left sidebar has a 'Runners' tab selected. The main content area is titled 'Runners / Add new self-hosted runner · dzsquared/wingtips-devops-samples'. It contains instructions for adding a self-hosted runner, mentioning the GitHub Actions Runner and the GitHub Customer Agreement. It shows three selection boxes for 'Runner image': macOS (unchecked), Linux (unchecked), and Windows (checked). Below that is an 'Architecture' dropdown set to 'x64'. A 'Download' section provides a command-line script for configuring the runner 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/v2.323.0/actions-runner-win-x64-2.323.0.zip -OutFile actions-runner-win-x64-2.323.0.zip
# Optional: Validate the hash
```

<https://github.com/Azure-Samples/container-apps-ci-cd-runner-tutorial>

The screenshot shows a Microsoft Visual Studio Code (VS Code) window with the following details:

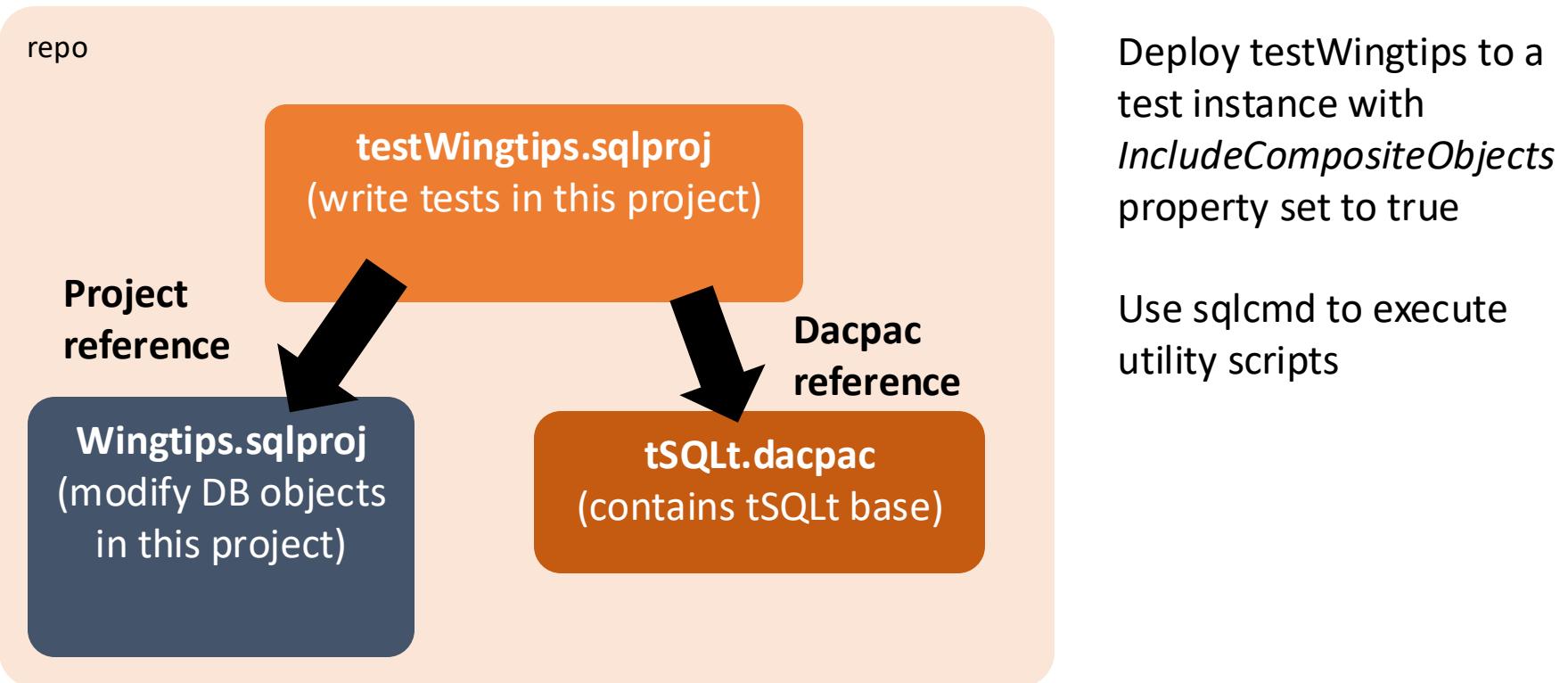
- Title Bar:** wingtips-devops-samples
- Explorer:** Shows the project structure under "WINGTIPS-DEVOPS-SAMPLES". Key files visible include `local-environment-update.yml`, `NewVenue.sql`, `Venues.sql`, `Dockerfile` (selected), `entrypoint.sh`, `testWingtips`, `Wingtips`, `bin`, `control`, `dbo` (containing `StoredProcedures` with files like `DeleteEvent.sql`, `DeleteVenue.sql`, `NewVenue.sql`, `ResetEventDates.sql`), `Tables` (containing `Countries.sql`, `Customers.sql`, `Events.sql`, `EventSections.sql`, `Sections.sql`), and `OUTLINE` and `TIMELINE`.
- Search:** A search bar at the top right.
- Editor:** The main editor area displays the `Dockerfile` content:FROM ghcr.io/actions/actions-runner:2.323.0
USER root
install dotnet sdk and sqlpackage
RUN apt-get update && apt-get install -y dotnet-sdk-8.0 && \
 # apt-get install sqlcmd && \
 dotnet tool install --tool-path /usr/local/bin/ Microsoft.SqlPackage
COPY entrypoint.sh ./entrypoint.sh
RUN chmod +x ./entrypoint.sh
USER runner
ENTRYPOINT ["./entrypoint.sh"]
- Bottom Status Bar:** Shows "Time Elapsed 00:00:05.59" and a status indicator for "Wingtips (main)".
- Bottom Icons:** Includes icons for GitHub, Git, and other development tools.

Unit testing – is it correct?

- Avoid unit tests on production environments
- Allows early detection of defects
- Increased confidence
- Test all the things
- Tests should be independent and small
- Automate as part of CI/CD



Writing, deploying, and running unit tests



feat/unit-testing ↻ ✘ 0 △ 0 ① 6 ⌂ Pull Request #6

wingtips-devops-samples

EXPLORER

OPEN EDITORS

- NewVenue.sql testWingtips/myTests
- testWingtips.sqlproj testWingtips
- build-sql2022.yml .github/workflows
- NewVenue.sql Wingtips/db/StoredPr...

WINGTIPS-DEVS-SAMPLES

- .github/workflows
- build-codeanalysis.yml
- build-sql2022.yml
- deployReportOutput.ps1
- publish.yml
- rogue-db.yml
- FabricWorkspace
- testWingtips
- Wingtips
- WingtipsFromLiveDB
- .gitignore

OUTLINE

TIMELINE

! .../myTests

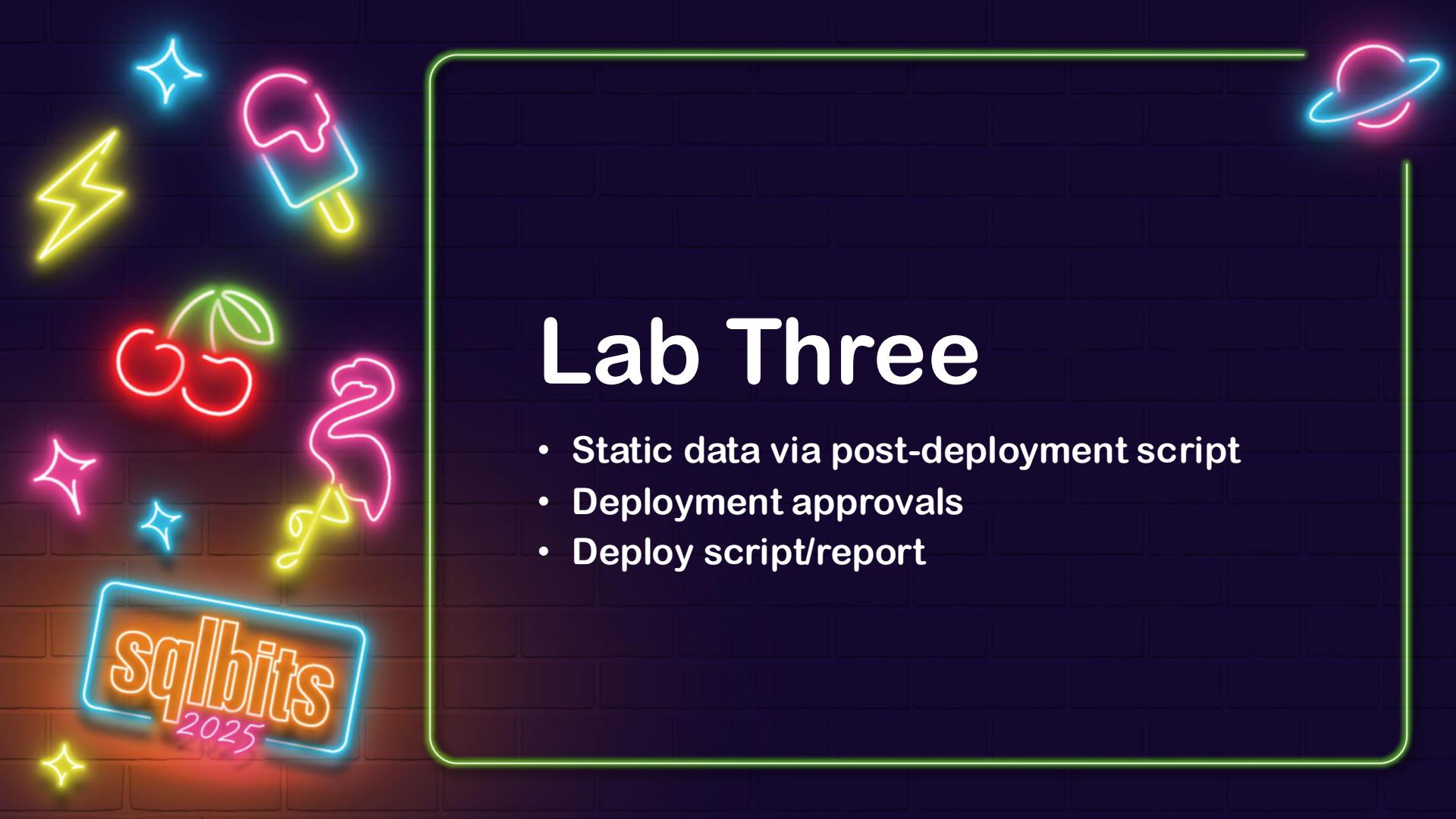
! build-sql2022.yml

! NewVenue.sql .../StoredProcedures

Wingtips > dbo > StoredProcedures > NewVenue.sql

```
1 -- Creates a new Venue plus a default sections and events.
2
3 CREATE PROCEDURE [dbo].[NewVenue]
4     @VenueId INT,
5     @VenueName NVARCHAR(128),
6     @VenueType NVARCHAR(30) = 'multipurpose',
7     @PostalCode NVARCHAR(20) = '98052',
8     @CountryCode CHAR(3) = 'USA'
9 AS
10    IF @VenueId IS NULL
11        BEGIN
12            RAISERROR ('Error. @VenueId must be specified', 11, 1)
13            RETURN 1
14        END
15
16    IF @VenueName IS NULL
17        BEGIN
18            RAISERROR ('Error. @VenueName must be specified', 11, 1)
19            RETURN 1
20        END
21
22
23    -- Insert Venue
24    INSERT INTO [dbo].Venues
25        ([VenueId], [VenueName], [VenueType], [AdminEmail], [CountryCode], [PostalCode])
26        VALUES
27        (@VenueId, @VenueName, @VenueType, 'admin@email.com', @CountryCode, @PostalCode)
28
29    RETURN 0
30
```

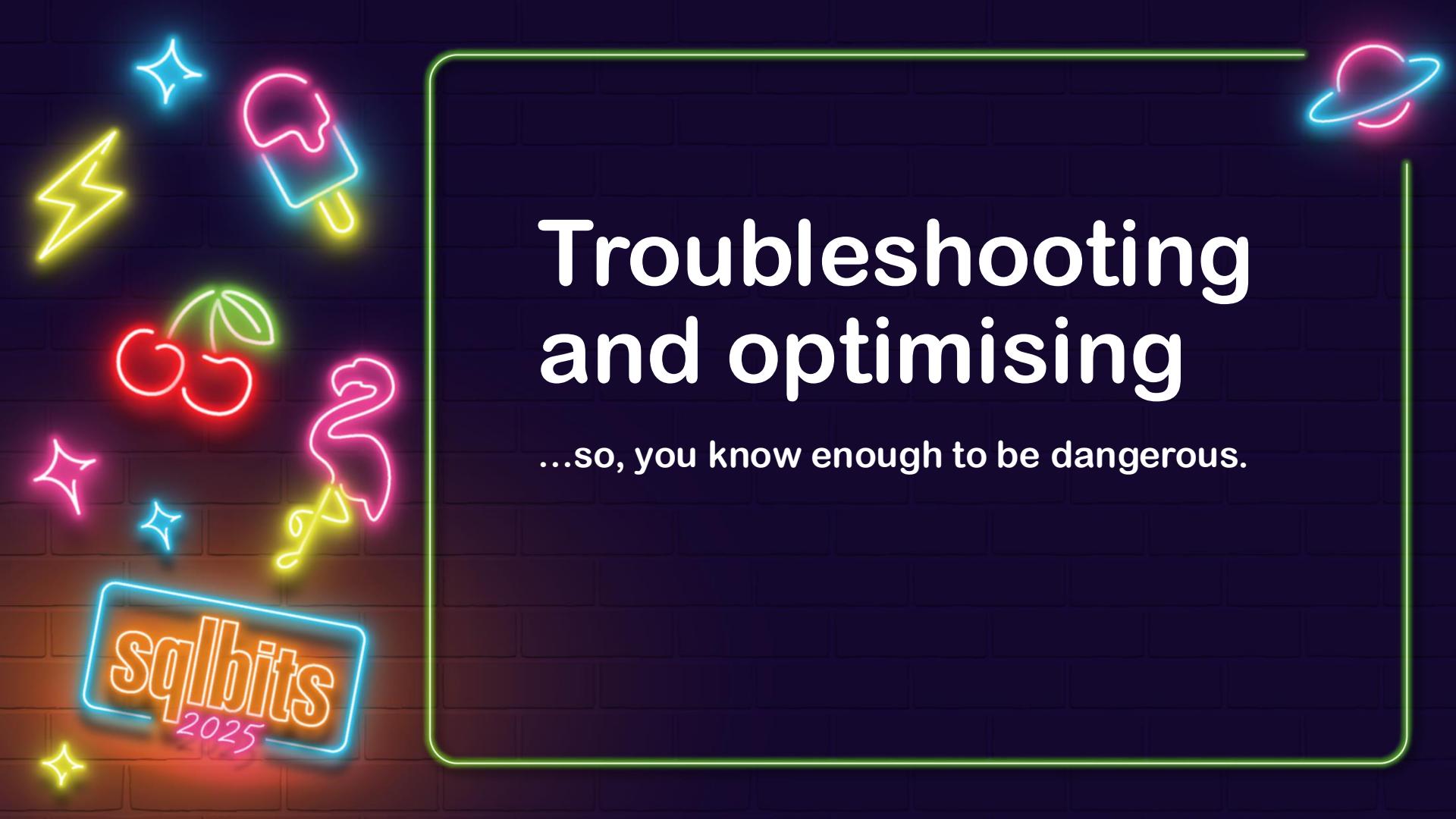
Drew Skwiers-Koballa (2 weeks ago) Ln 15, Col 1 Spaces: 4 UTF-8 with BOM LF {} SQL

The background features a dark blue brick wall with various neon signs. On the left, there's a large red and yellow cherry sign, a yellow lightning bolt, a blue star, a pink ice cream cone, a yellow star, and a pink flamingo. On the right, there's a blue planet with a ring and a green outline of the word 'sqlbits'. A large green rectangular border surrounds the central content area.

Lab Three

- Static data via post-deployment script
- Deployment approvals
- Deploy script/report

sqlbits
2025

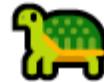
A dark blue brick wall background features several glowing neon signs. In the top left, there's a yellow lightning bolt, a blue starburst, a red cherry with a green leaf, a pink ice cream cone, a yellow starburst, and a yellow flamingo. In the bottom left, there's a blue starburst, a pink starburst, and a blue 'sqlbits 2025' logo. A thin green rectangular border surrounds the central text area.

Troubleshooting and optimising

...so, you know enough to be dangerous.

Pipeline best practices

- Specify environment version (avoid “windows-latest” and “ubuntu-latest”)
- If using a task, specify a version or a git commit hash (avoid latest)
- Watch for case sensitivity



My build is slow – what now?!

Unresolved references

- Completely unresolved references and case-mismatch delays the build
- Code sanitization takes time but pays off

Build fewer objects

- Project references extend the build time significantly
- Artifact/package references have much smaller impact
- Use project references only when contained objects change frequently and on same cadence

Complain

- No, seriously
- If you can share your project with Microsoft, reach out to

github.com/microsoft/dacfx

Branch management

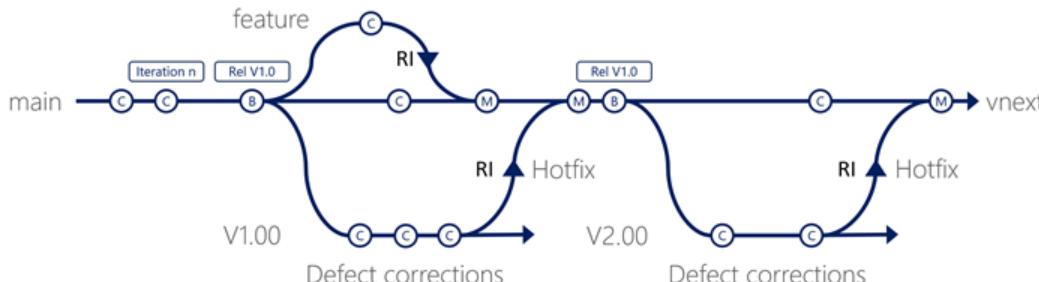
- Start simple ‘main’ branch



- Progress to introduce feature branching



- Work towards multiple branching strategies



Tasks vs script directly

Script

- Script steps offer complete control
- Barebones operations
- .NET, SqlPackage, sqlcmd, and any CLI can be used directly
- Bash, cmd, PowerShell

GitHub azure/sql-action

- Windows+Linux
- Azure + anywhere
- Adds/removes firewall rule
- Builds a sqlproj, deploys a dacpac + other SqlPackage commands

Contributors welcome

ADO

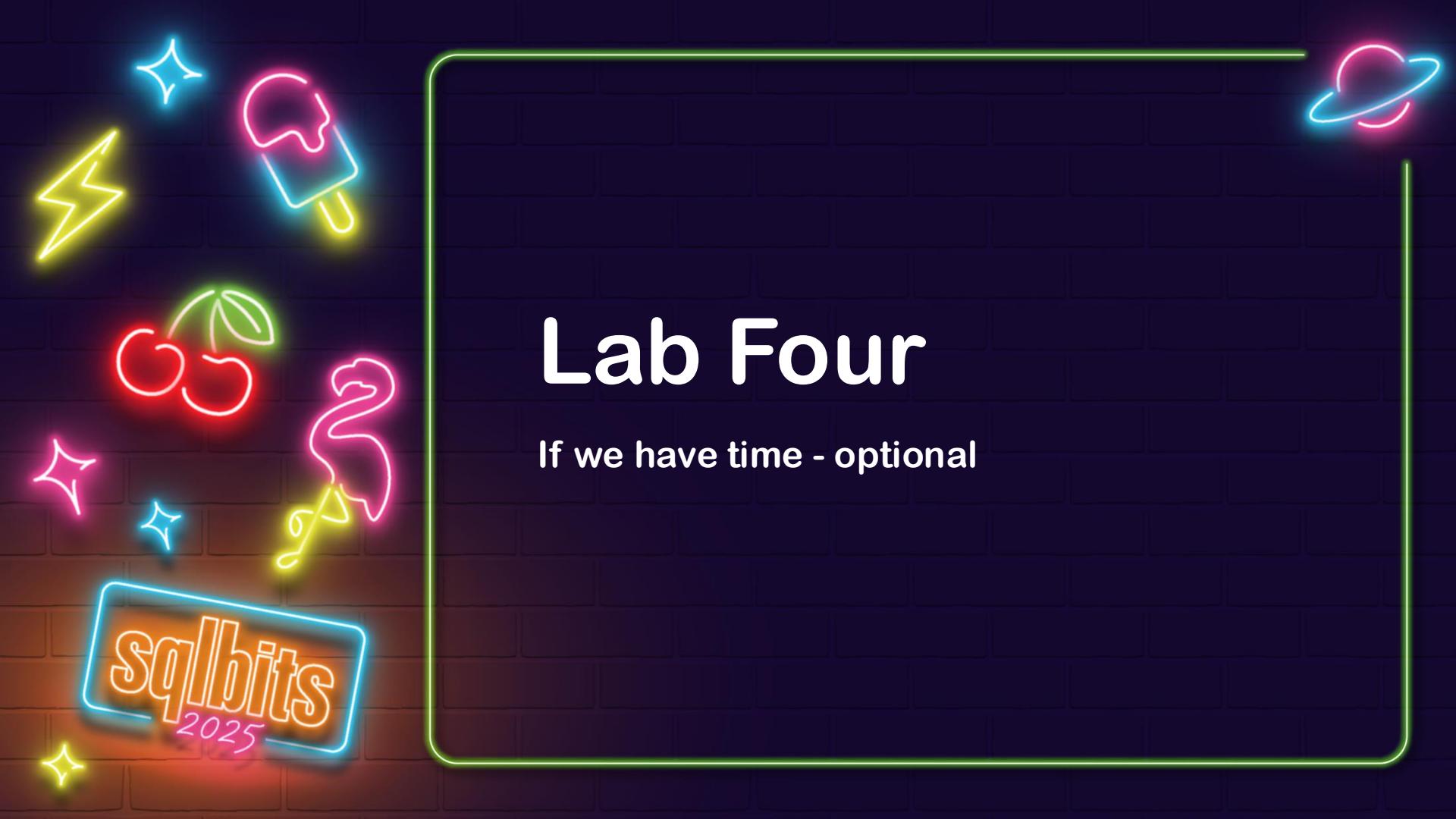
SqlAzureDacpacDeployment

- Windows only, Azure only
- Picks out the version of SqlPackage*
- Adds/removes firewall rule from Azure
- Deploys a dacpac + other



Upcoming renovation



A decorative background featuring various neon signs on a dark brick wall. On the left, there's a sign for 'sqlbits 2025' with a blue-to-orange gradient. Above it are neon cherry signs in red and yellow, a lightning bolt in yellow, a star in pink, and a planet with a ring in blue and pink. On the right, there's a neon flamingo in yellow and pink, a popsicle in pink and yellow, and a star in blue.

Lab Four

If we have time - optional

Wrap Up



What did we not talk about today?

- Publish profiles
- Deployment contributors
- Writing custom code analysis rules
- Refactorlog/refactoring

Resources

Documentation

- [Aka.ms/sqlprojects](https://aka.ms/sqlprojects)

Reach out for lab follow up:

drskwier@microsoft.com

More Community Members

- Kevin Chant
- Erik Ejlskov Jensen
- Kevin Pereira
- Olivier Van Steenlandt
- Erin Dempster

One last thing

Email: hamish@makesstuffgo.com for a
30-minute FREE consultation

UKT mornings are best for me

