CI/CD for Azure SQL in DevOps

Making it Easier

Overall Process

1) Develop your Database Project in Visual Studio and check it in
2) Create a Build Pipeline
3) Create a Release Pipeline
4) Enable the CI/CD Trigger
5) Enable Pre-Deployment Approvals
6) Verify your DB Changes
7) Setup the DevOps Dashboard Badges

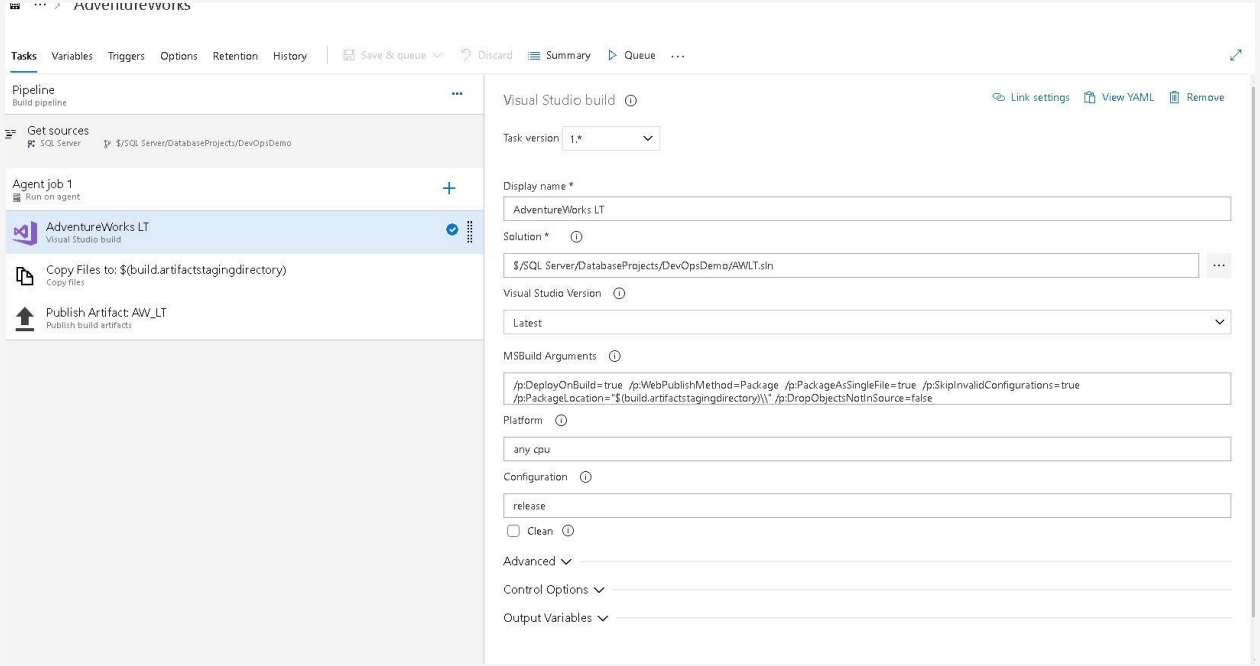
Creating a Build Pipeline

1. The Build Pipeline takes your source code and creates an **artifact**.

2. This requires that you store your code in a DevOps Repo. This Demo uses the TFVC Source Control provider.

3. Because this is a SQL instance, the build process simply runs SQLPACKAGE.EXE to create a DacPac as the artifact.

4. The DacPac becomes the input for the next step, the Release



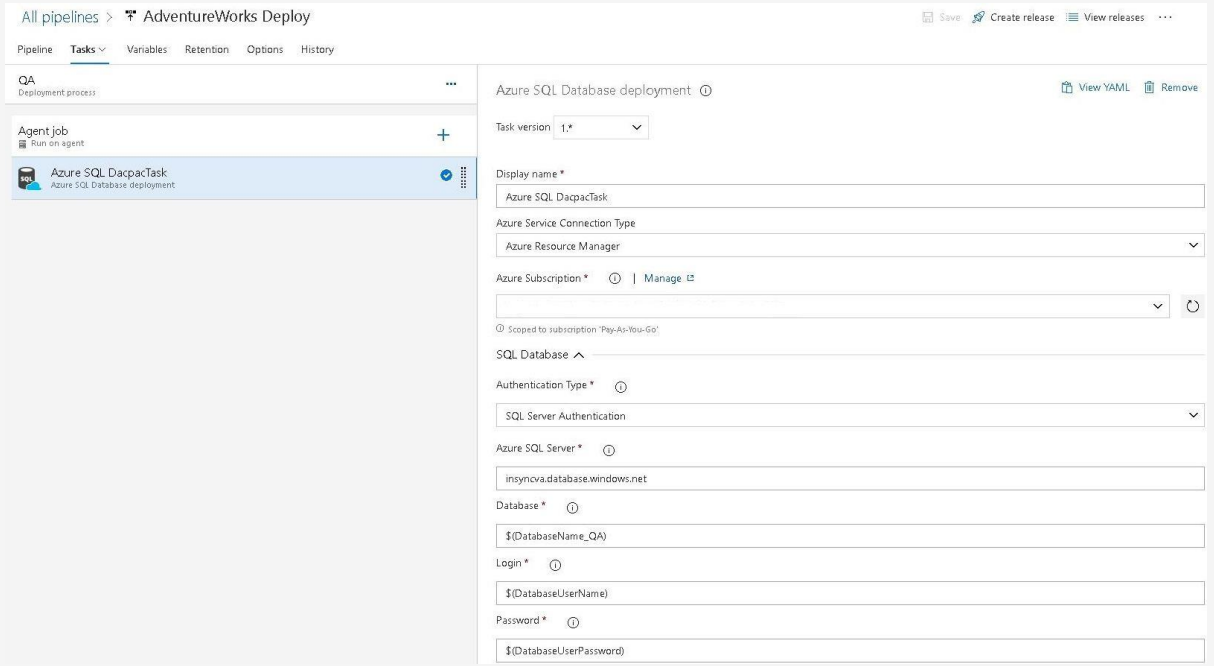
Creating a Release Pipeline

1. The Release Pipeline takes the DacPac and publishes it to an Azure SQL instance.

2. You can chain deployments to QA, DEV, Prod tiers on the same or separate SQL Servers

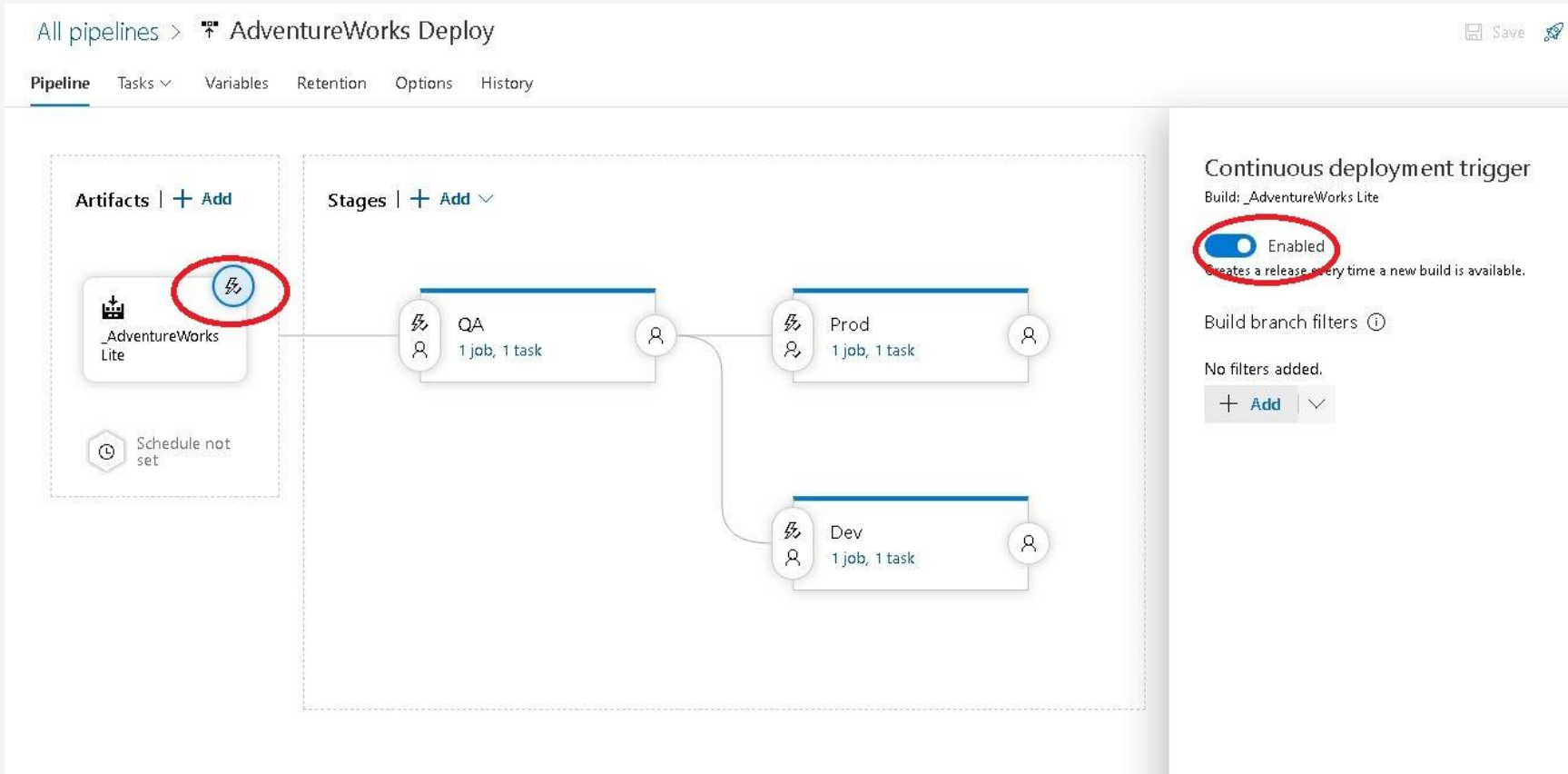
3. You can ask a Process Manager to Approve the Deployment

4. The system emails you and sets Project Dashboard build badges



CI/CD Trigger

The Continuous Integration/Continuous Deployment trigger is found on the Artifact object in the Release Pipeline



More questions about CI/CD in Azure SQL?

Azure Pipleline Docs

Gwalkey Github