Deploying Docker For Database DevOps Dominance

Dan Mallott

Dan Mallott

- Principal for West Monroe Partners
- In the industry since 2011

- Experience with SQL Server, Oracle, PostgreSQL, and Cassandra
- Been both a DBA and a developer
- Have a couple Microsoft certifications
- DataStax Certified Professional

- Twitter: @DanielMallott
- Github: https://github.com/danielmallott
- LinkedIn: https://www.linkedin.com/in/danielmallott/

What's On Deck?

Why Are We Here?

What Tools Can We Use?

Example Workflow

Question Time

Why Do We Want to Do This Anyway?

First, Some Definitions

- DevOps – A set of software development processes that combines software development and information technology operations to shorten the systems development lifecycle while delivering features, fixes, and updates frequently in close alignment with business objectives (Wikipedia)
- Continuous Integration (CI) – Integrating into a common branch frequently, building the code, and running automated unit tests to ensure the changes are valid
- Continuous Delivery (CD) – Automated release process that allows push button release of the CI artifacts

Why Use a CI/CD Process With Your DDL and DML?

- Your data definition language and data manipulation language is part of your codebase, just like the application code
- Practicing CI/CD in conjunction with storing your DDL and DML in source control allows your developers to see changes as they happen and participate in the process
- You will share a common development, build, and deployment mechanism with your application developers
- You can definitively answer the question “what was deployed?” and can recreate the database shape to any point in time
- Building and testing your database can help identify issues such as misspeelings or missing objects

The Toolset

Version Control for Database Code

State-based Tools

- State-based Tools generate a script to upgrade your database by comparing an existing database to the model stored in code
- This can be automated or be run manually by developers
- Examples include:
  - SQL Server Database Tools
  - Redgate SQL Source Control

Migration-based Tools

- Migration-based Tools assist in the creation of migration scripts that move the database from one version to the next
- This is usually done from a baseline version
- Examples include:
  - Flyway
  - Liquibase
  - Ruby on Rails Migrations
  - Other ORM Migration Tools

Unit Testing – Yes, Your Database Code

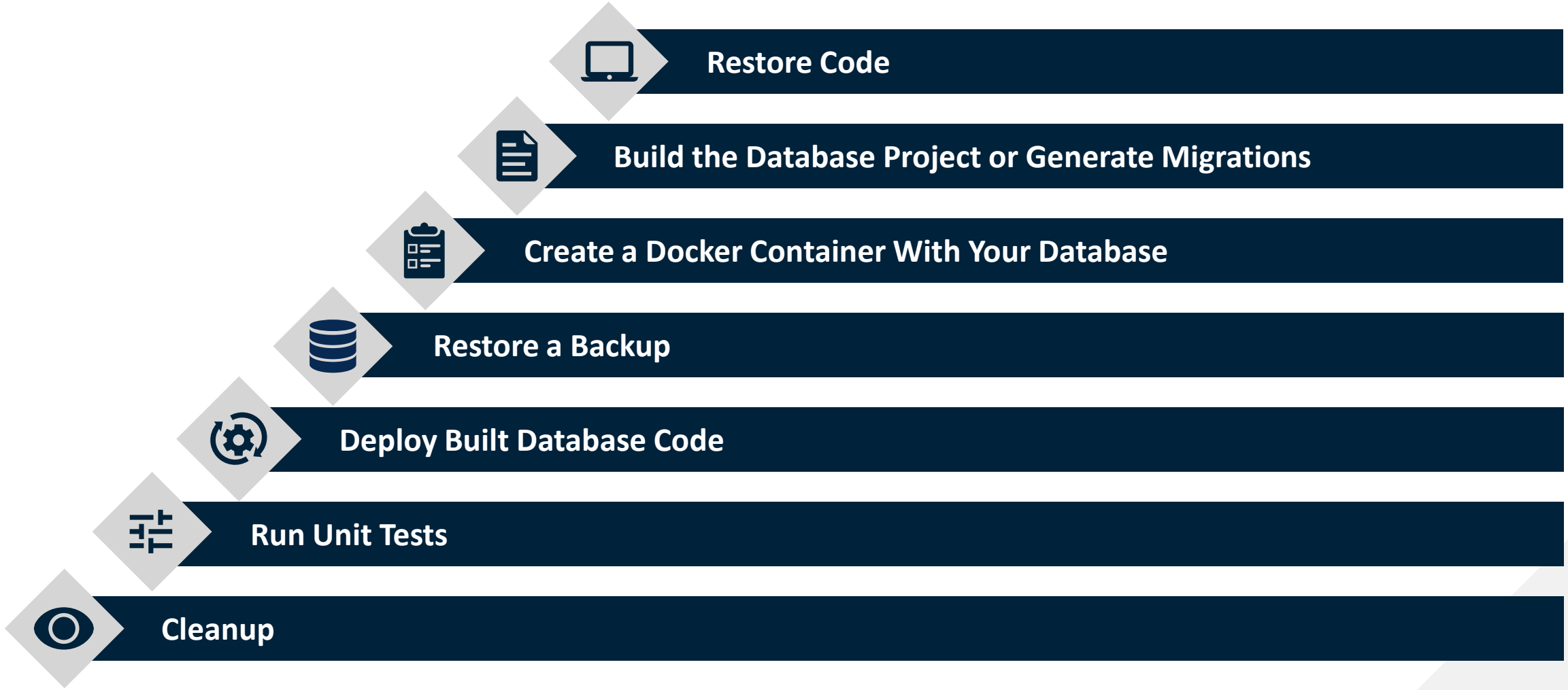
- If you have business logic in your database, it needs to be tested somehow
- The result of your DML can change (or stop working entirely) with DDL changes
- There are a variety of database unit testing frameworks available
  - tSQLt (SQL Server)
  - Redgate SQL Test (SQL Server)
  - DbFit (Cross-Platform)
  - NDbUnit (Cross-Platform)
  - DBTestDriven (SQL Server and Oracle)
- But…unit tests can only be effective when you run them, preferably in an automated fashion

Docker – For Databases?

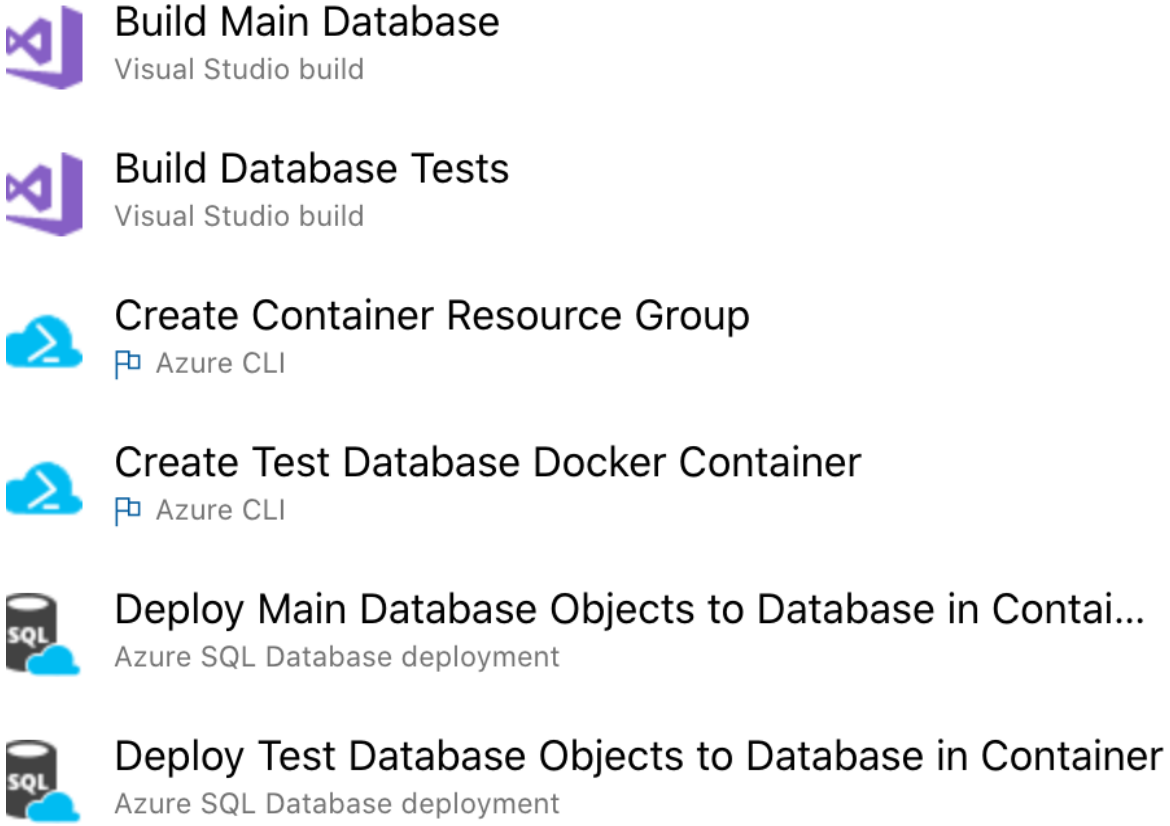
- What is Docker?
  - Docker is a tool that allows creation and deployment of containers
  - Containers allow you to host only the parts of the OS that are absolutely necessary for running an application
  - Essentially, Docker allows for the flexibility of Virtual Machines without the large overhead
- Most major database platforms have Docker containers available
  - SQL Server
  - PostgreSQL
  - MySQL
  - Oracle
- While these may not be appropriate for production workloads, we can use them to host a database for testing temporarily

Putting It All Together

Creating a DevOps Pipeline



An Example Pipeline



Additional Thoughts

Some (Potential) Pitfalls

- Handling data type changes
  - This potentially involves both a pre-deployment script (to copy data out of the affected table) and a post-deployment script (to copy data back into it)
  - These scripts will need to test whether the change is in place or not
- Large-scale refactorings
  - This is a place where you probably want to handle this out-of-band and create a new baseline
- Load testing
  - There are other tools to use for load testing
  - Load testing a database in isolation may not yield a true picture of where your application will have performance challenges

Questions?