



Azure DevOps Simplified with Production Data

Paul Stanton, Co-founder Windocks
pauls@windocks.com



PASS

SQL DevOps Simplified with Production DBs

Three choices in SQL Server DevOps, and which CI server?

1. Build DBs from source, or use DB Clones w/migration scripts?
 - Source DB Clones from SQL Backups, Cloud storage, or storage arrays
2. Git or Git plus DB Source control (Red Gate, Liquibase, etc.)?
3. SQL Server containers or instances?

And, Azure DevOps, Jenkins, Team City, or other CI server?

This session: DB clones, Git w/migration scripts, Azure DevOps.
We'll use containers and discuss pros/cons of Instances.



PASS

SQL DevOps Simplified with Production DBs

Options for DB clones with Azure DevOps

- Windocks: SQL Server DB clones for containers or instances
- Red Gate SQL Clone for instances
- PowerShell scripting

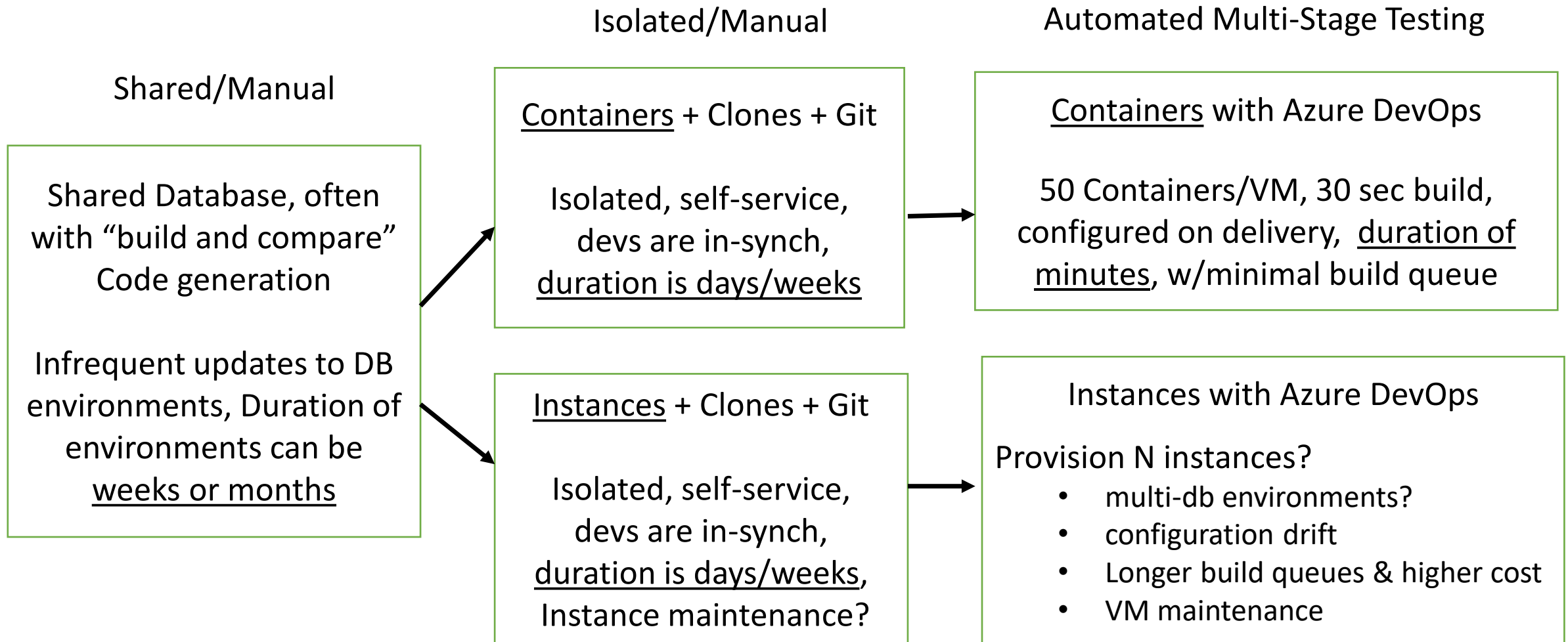
Why use DB clones?

- DB clones are a superior artifact, for higher quality releases
- Support multi-database environments (multiple DB builds are complex and slow)
- Faster, more reliable process compared to DB builds



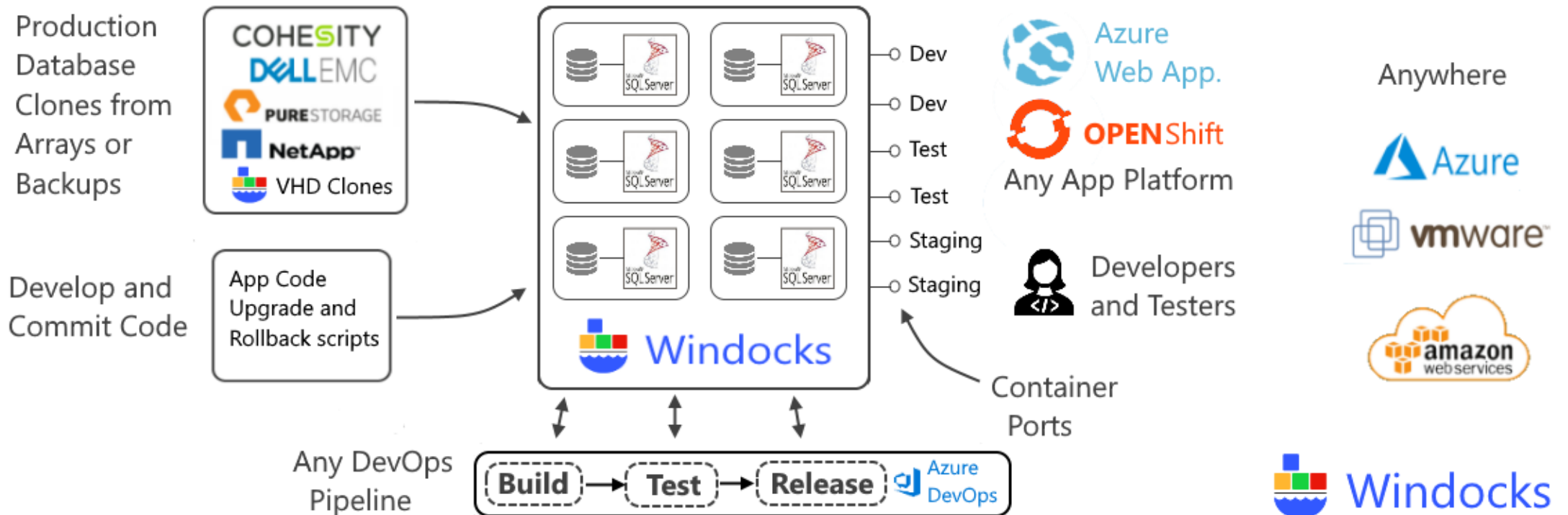
PASS

Plan for high volume CI testing





SQL DevOps simplified: production database clones with upgrade scripts applied, with SQL containers, on a single VM (in seconds).





PASS

Migration script ordering and management

A PowerShell script concatenates and orders migration scripts to produce an “ordered” script (FeatureA.sql). Orderscripts.sql is updated and committed for each migration script commit.

orderscripts.sql

Get-Content script1.sql, script2.sql, script3.sql | Set-Content FeatureA.sql



PASS

Combining DB clones with migration scripts

A dockerfile combines DB clone image with Git and script management. Each container includes the database clone, a clone of the repo in the \scripts folder, plus the “ordered” FeatureA.sql. Azure DevOps makes restful API calls to create containers, run tests, and integrate with front-end apps.

```
FROM mssql-2017
SETUPCLONING Full customers \\path\to\backup
COPY datamasking.sql .
RUN datamasking.sql
ENV USE_DOCKERFILE_TO_CREATE_CONTAINER=1
RUN git.exe clone https://url/repo scripts
RUN powershell.exe scripts\orderscripts.ps1
```



PASS

Control use of migration scripts at run time

Control which scripts are applied at run time with Docker commands. Azure DevOps restful API calls include this environment variable.

```
>docker create -e RUN="scripts\script1.sql" <dbcloneimage>
```

```
>docker create <dbcloneimage>
```

```
>docker create -e RUN="scripts\script1.sql, scripts\script2.sql" <dbimage>
```




PASS

Advantages of DB clones in DevOps

- DB clones are a superior artifacts for development than “built” databases
- Identify data and performance issues earlier
- Avoid DB build times, and support multi-database environments
- Combined with containers to realize additional benefits:
 - greater speed
 - reliability
 - security
 - And economy (scalability)



PASS

Setup a personal Azure DevOps lab

- A “ready for use” Azure DevOps YAML file provided by Windocks
- ~2 minute Azure DevOps setup
- The same VM and pipeline that we used today
- Free: for short-term (2 week) experience . . .
- Interested in a personal lab longer-term? Download a free Windocks Community edition server and continue indefinitely!

Download the YAML at: www.windocks.com/azure-devops