Enable Self-Service Provisioning of Versioned Databases from Azure DevOps

AKA How to Build Workflow-Driven Database Provisioning in Azure DevOps

Tonie Huizer

promicro

Tonie Huizer

He/him

Freelance DevOps consultant

- linkedin.com/in/toniehuizer
- github.com/promicroNL/events
- www.promicro.nl

promicro

SeaQL.nl
Data, DevOps & SQL @ Sea!

redgate
COMMUNITY AMBASSADOR

Data Saturday Gothenburg 2025

Sponsors

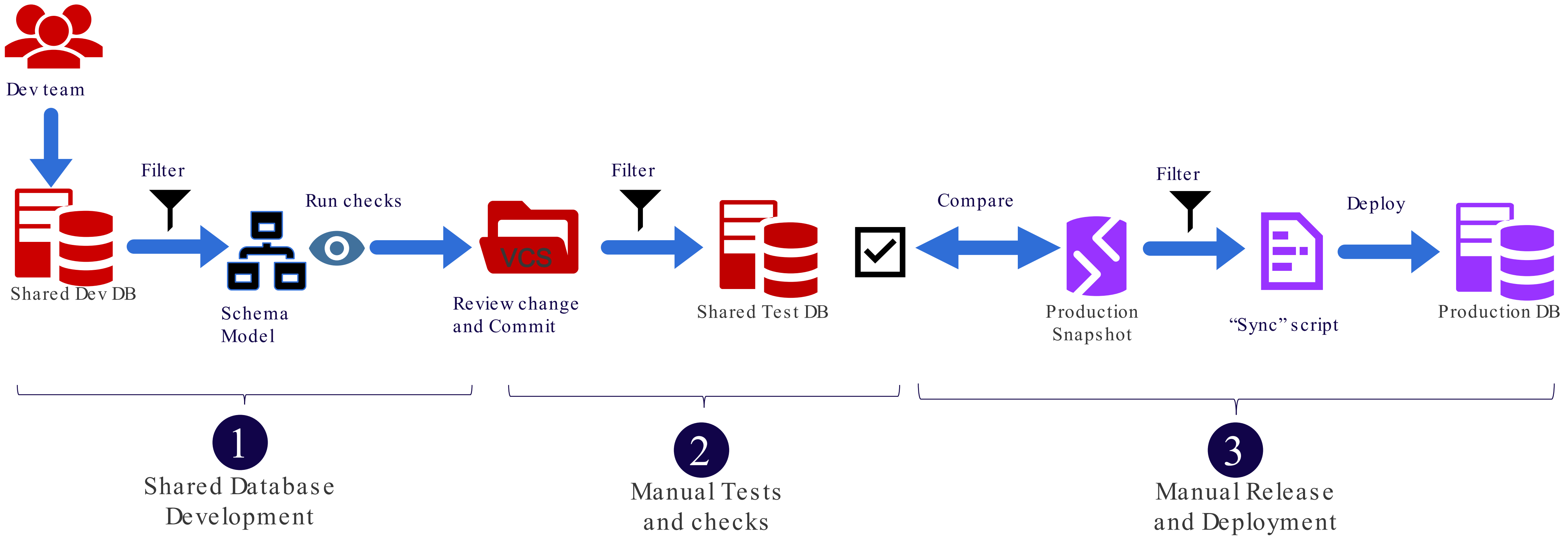
The after-party sponsor

What to expect of today's session

Segment
Welcome
Why Our Database DevOps Wasn’t Working (Yet)
What we changed: A Workflow-Driven Approach
Provisioning Should Follow the Work (DEMO)
PR workflow and house keeping (DEMO)
Wrap-up: Take aways & Q&A

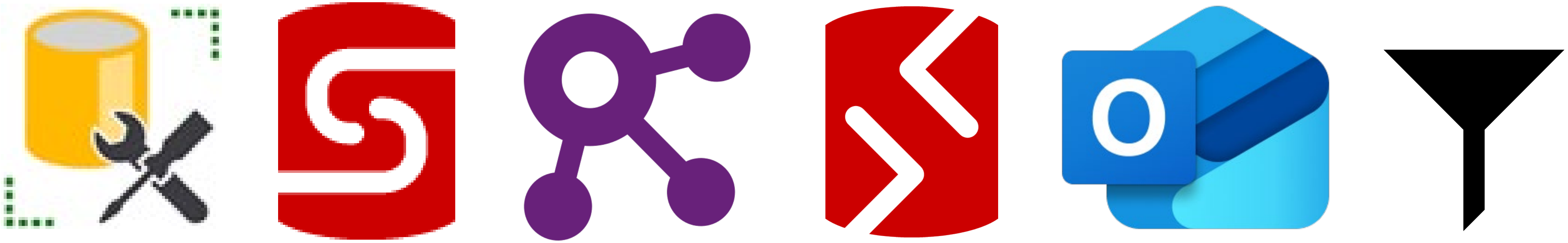
Why Our Database DevOps Wasn’t Working (Yet)

Our start: No real Database DevOps

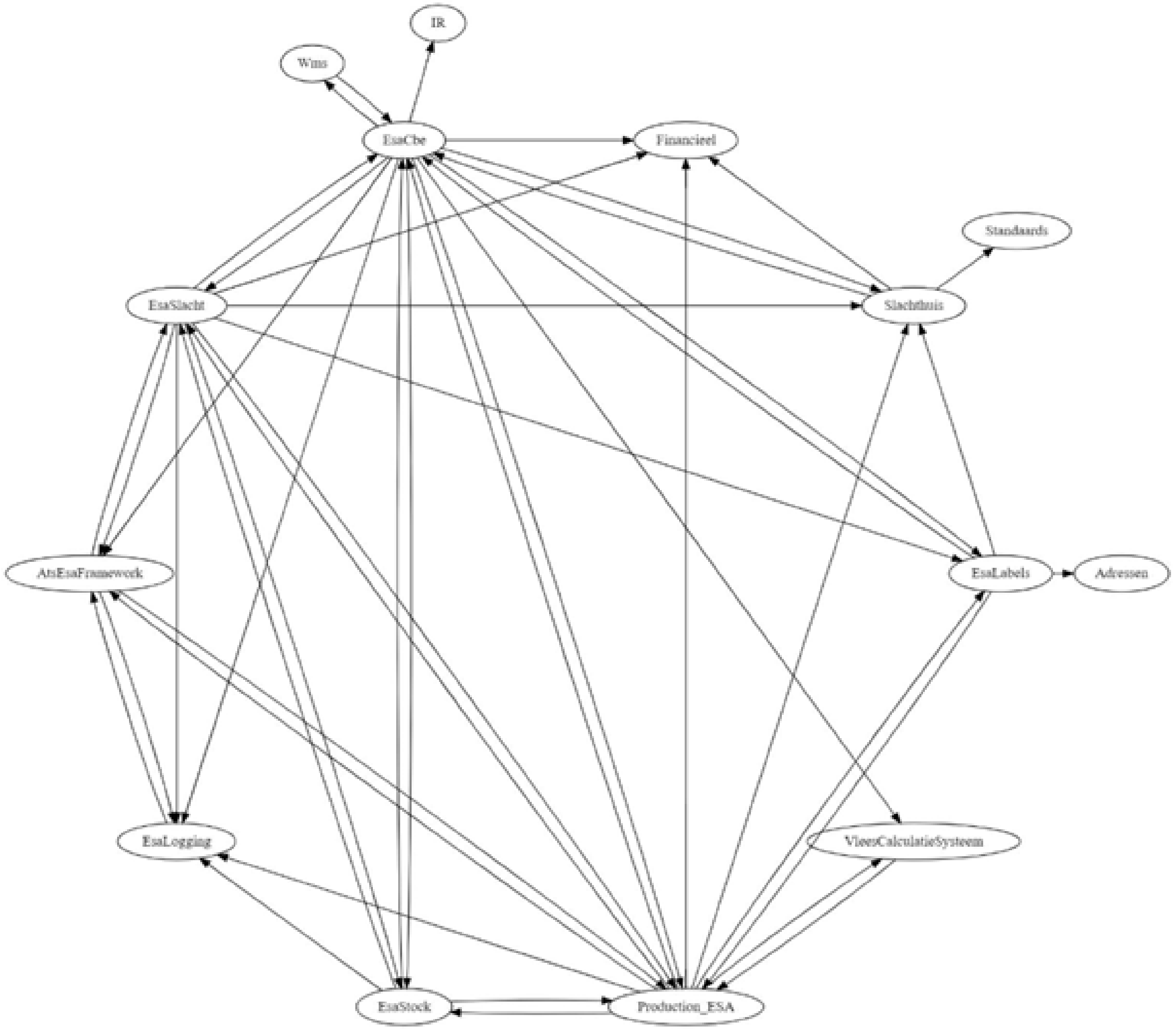


To sum up

- Shared database
- Source control
- Manual
- Repeated actions
- “DevOps” tools:



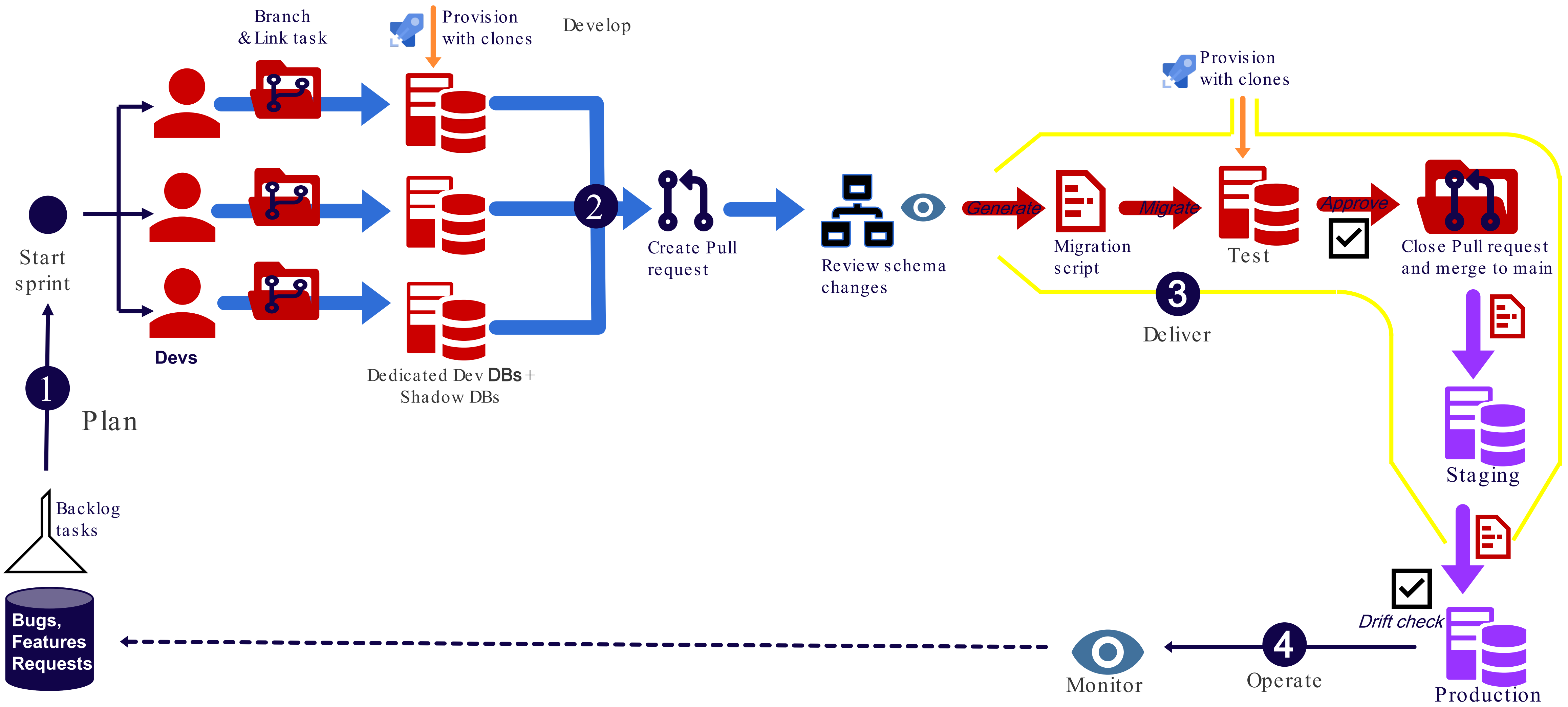
Database DevOps – the challenge



Database DevOps – the challenge

Our Solution: Workflow-Driven Provisioning

The result: Database DevOps



#DataSaturdayGothenburg-10

What did we change?

- Choose a VCS to collaborate and work in parallel
- Introduced dedicated disposable environments
- Turned code into immutable artifacts
- Automate delivery processes with approvals
- Tested in parallel, just like development
- Implement Pull Request Release workflow

Provisioning Should Follow the Work

A Simpler, Smarter Provisioning Model

1. Pick up a work item

From the
Azure DevOps
Board

2. Create a Git branch

Linked this work item
→ triggers build
→ provisions a database

3. Do you work

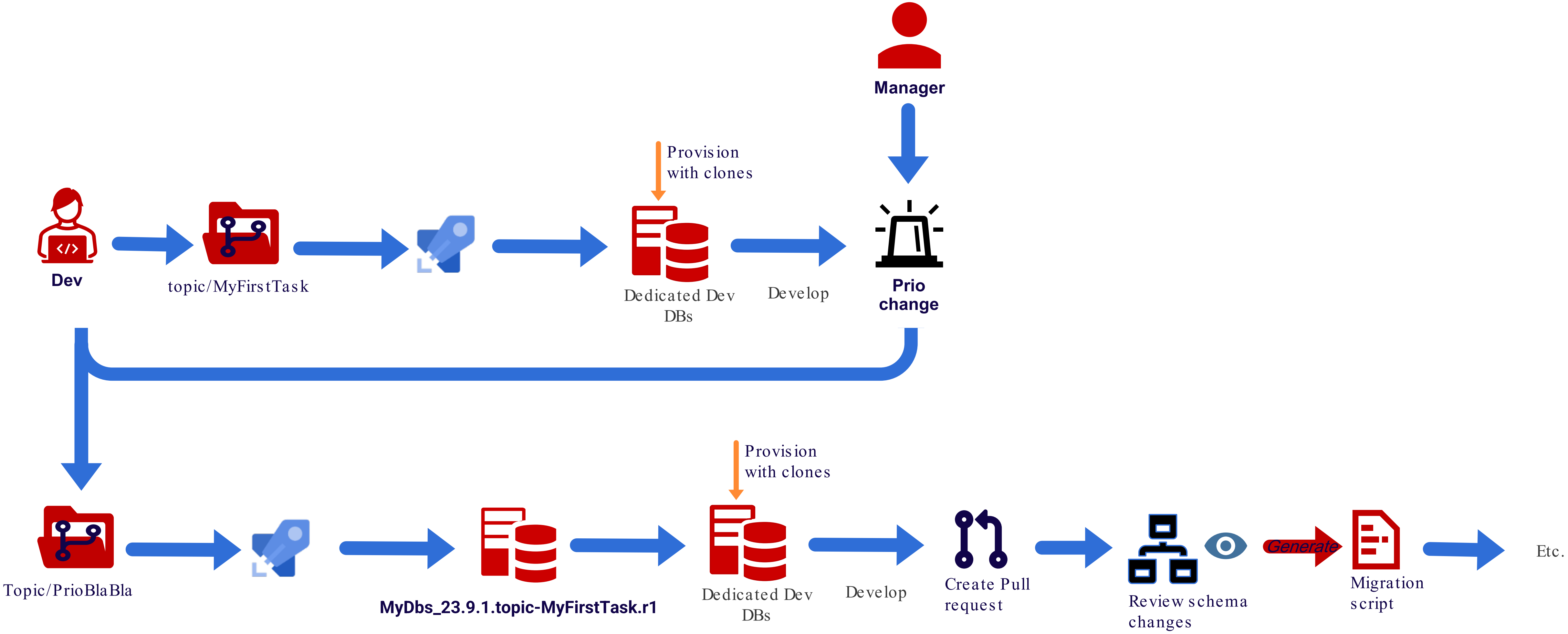
For Develop
→ approve release
for test → PR workflow

4. Dispose environment

Automatic teardown
For a clean
Dev/test environment

The concept of database stashing

- Changing priorities
- git stash but then for DBs
- Personal instance / environment
- Through Azure DevOps not on your system



Manager

Provision with clones

Dev

topic/MyFirstTask

Dedicated Dev DBs

Develop

Prio change

Provision with clones

Topic/PrioBlaBla

MyDbs_23.9.1.topic-MyFirstTask.r1

Dedicated Dev DBs

Develop

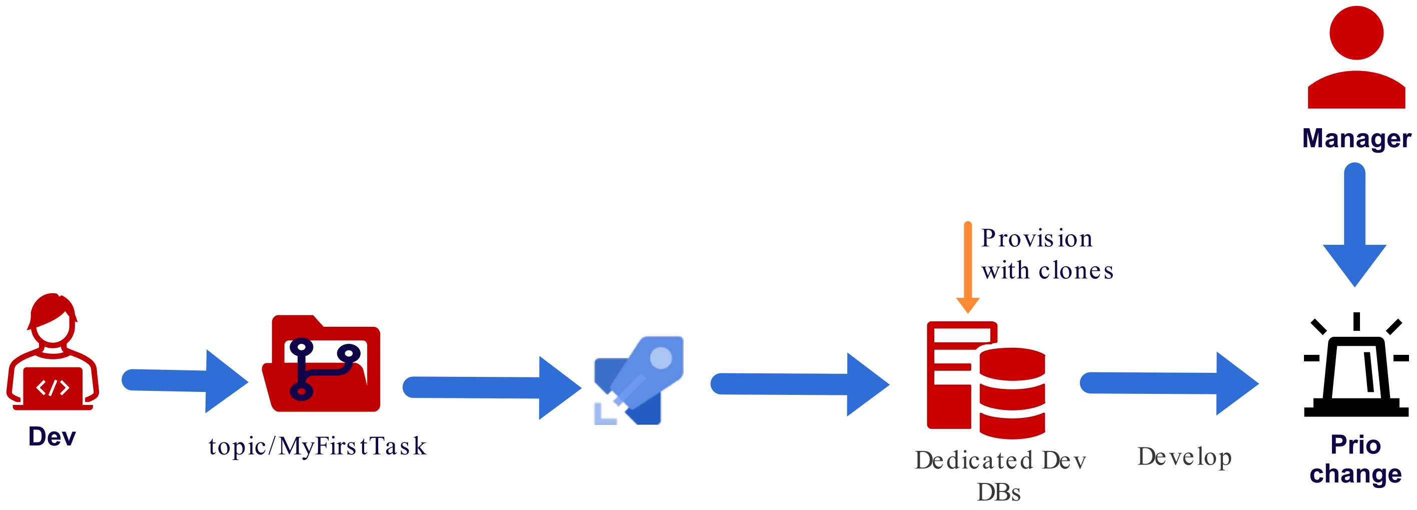
Create Pull request

Review schema changes

Generate

Migration script

Etc.



Manager

Provision
with clones

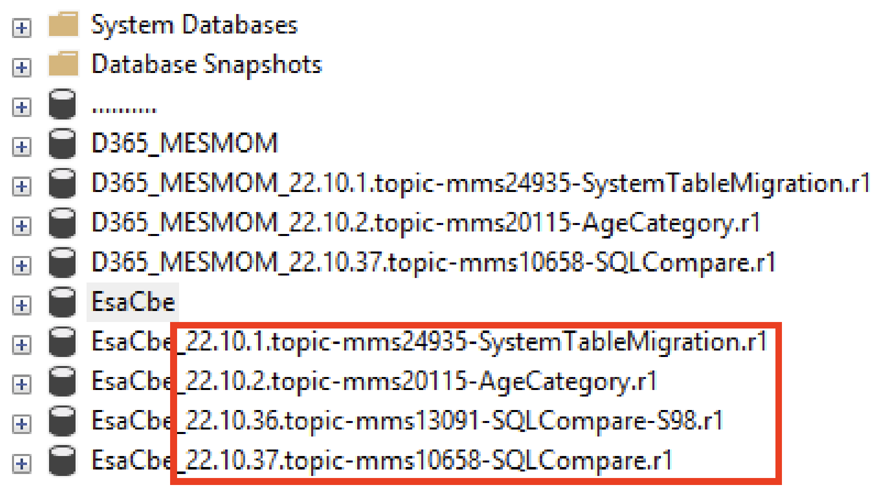
Dev

topic/MyFirstTask

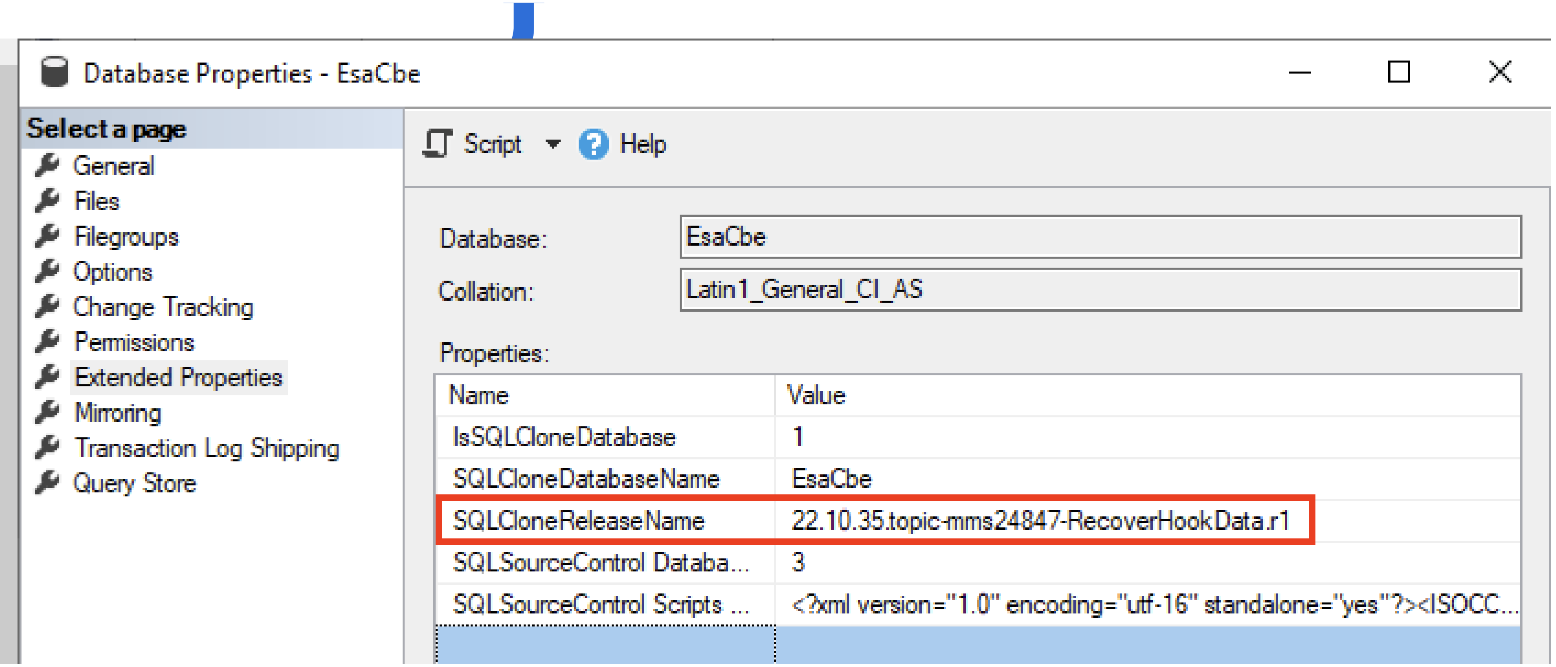
Dedicated Dev
DBs

Develop

Prio
change



- System Databases
- Database Snapshots
- ....
- D365_MESMOM
- D365_MESMOM_22.10.1.topic-mms24935-SystemTableMigration.r1
- D365_MESMOM_22.10.2.topic-mms20115-AgeCategory.r1
- D365_MESMOM_22.10.37.topic-mms10658-SQLCompare.r1
- EsaCbe
- EsaCbe_22.10.1.topic-mms24935-SystemTableMigration.r1
- EsaCbe_22.10.2.topic-mms20115-AgeCategory.r1
- EsaCbe_22.10.36.topic-mms13091-SQLCompare-S98.r1
- EsaCbe_22.10.37.topic-mms10658-SQLCompare.r1



Database Properties - EsaCbe

Database: EsaCbe

Collation: Latin1_General_CI_AS

Properties:

Name	Value
IsSQLCloneDatabase	1
SQLCloneDatabaseName	EsaCbe
SQLCloneReleaseName	22.10.35.topic-mms24847-RecoverHookData.r1
SQLSourceControl Databa...	3
SQLSourceControl Scripts ...	<?xml version="1.0" encoding="utf-16" standalone="yes"?><ISOCC...

Let’s Automate our tasks

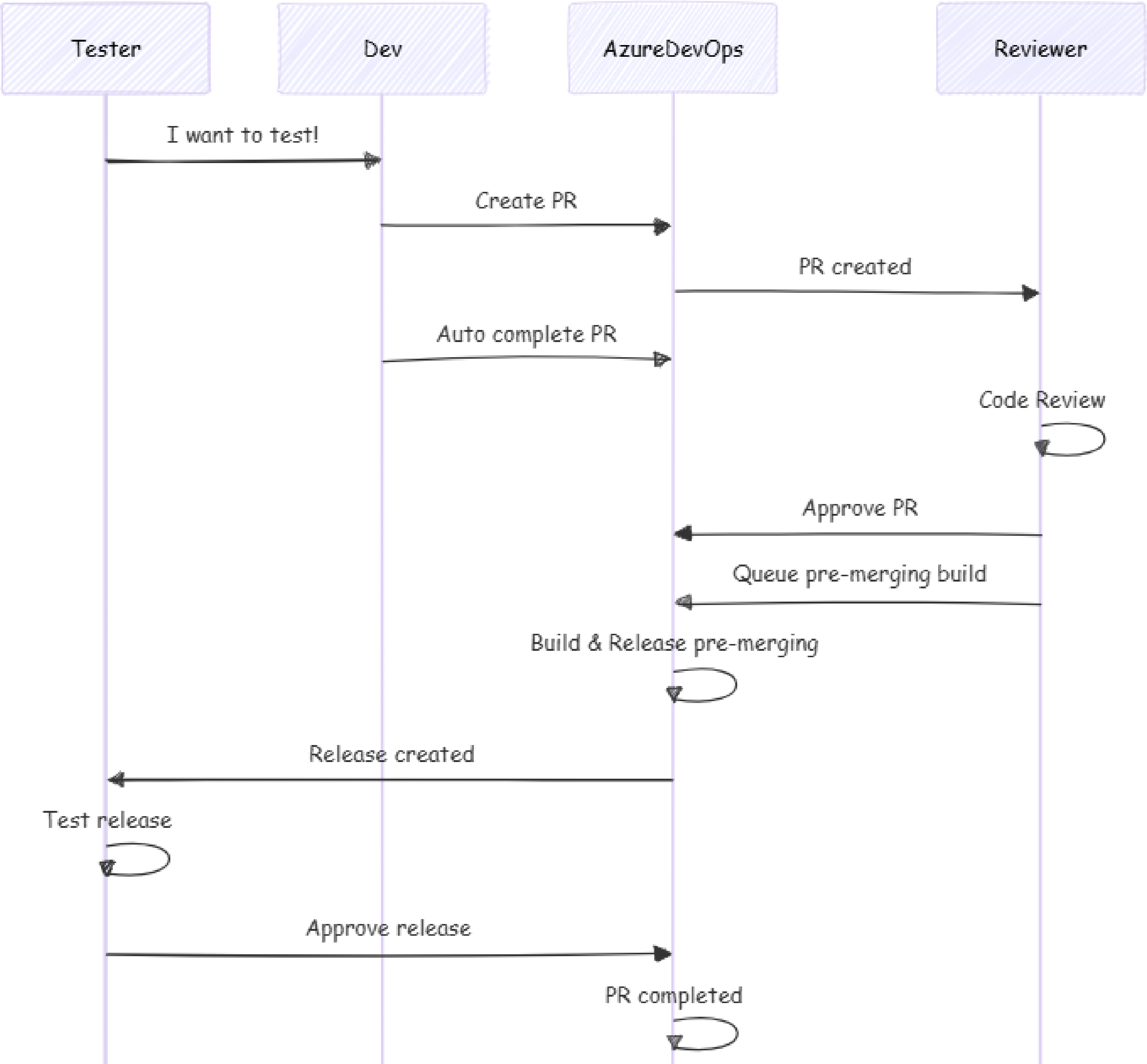
- Use pipelines, but first…
  - Start local to automate
  - Use verbose logging
  - Avoid inline PowerShell

Demo time

What did we just see?

- 00-utils.ps1 – Shared Utility Library
- 01-precheck.ps1 – Environment Validation
- 02-create-db.ps1 – Create Database or Unstash
- 03-apply-sql.ps1 – Apply Migrations
- 04-verify.ps1 – Post Migration Verification
- 05-stash.ps1 – Stash Database
    - Renames the database with the current build name, preserving it for later inspection or rollback.
    - Only stashes when the existing BuildName extended property matches

PR Workflow & House keeping


Tester
Dev
AzureDevOps
Reviewer
I want to test!
Create PR
PR created
Auto complete PR
Code Review
Approve PR
Queue pre-merging build
Build & Release pre-merging
Release created
Test release
Approve release
PR completed

Housekeeping for Provisioned Environments

- These environments are disposable
  - created per branch or PR
- Automatically deleted when:
  - The build is completed successfully
- Fully automated via scheduled pipelines

Housekeeping for Provisioned Environments

- Cleanup snapshots / images / etc:
  - Unused images (not linked to any active environment)
  - Outdated images
- Regular refresh jobs:
  - Run on weekends
  - Capture fresh production snapshots for the next workweek

Demo time

What did we just see?

- Configuration of branch policies:
  - PR workflow
- 06-resolve-approver-env.ps1 – Map Approver Environment
  - Queries Azure DevOps to find who approved
  - Sets Approver and TargetEnvironment for downstream job
- 07-cleanup-stash.ps1 – Remove Old Stashed DBs
  - Lists stashed databases (name + build suffix)
  - Checks each build’s status via Azure DevOps
  - Drops any that belong to completed builds to free up resource

Wrap-up:
Take aways & Q&A

Take aways

- No work item, no environment
- Use dedicated over shared
- Easily switch out the heart; the provisioning part
- Use scripts but only those that live in git
- Automation rocks…
  - but without logging it's too much black box

Questions?

Data Saturday Gothenburg 2025

Sponsors

Republify
PART OF UPER GROUP

b3

The after-party sponsor

24

Pagesplit

NEIGHBOUR
HOOD

Thank you

I appreciate the time you spent with me.

Please reach out if you have any questions!

Tonie Huizer

linkedin.com/in/toniehuizer

github.com/promicroNL/events

www.promicro.nl