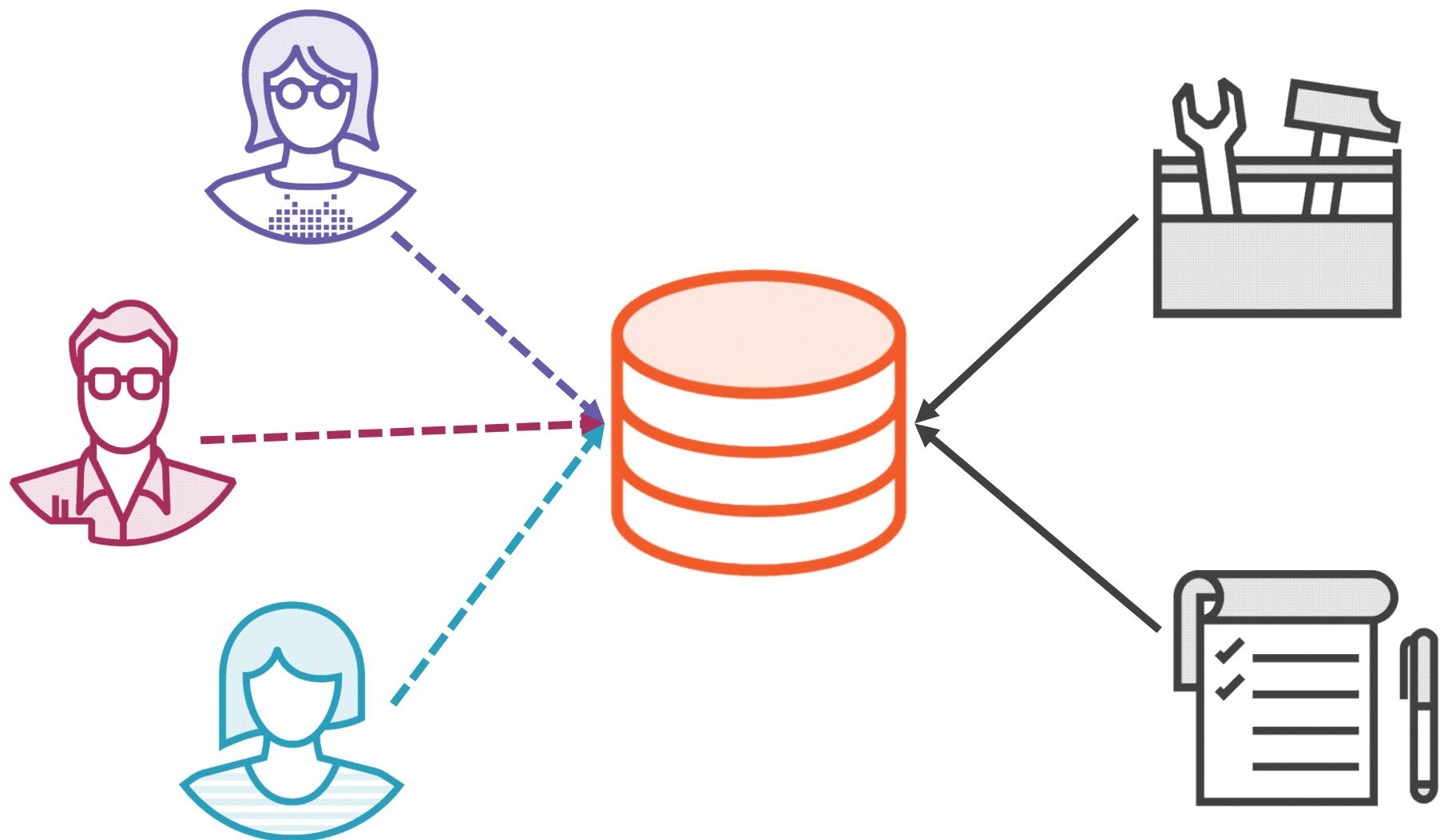


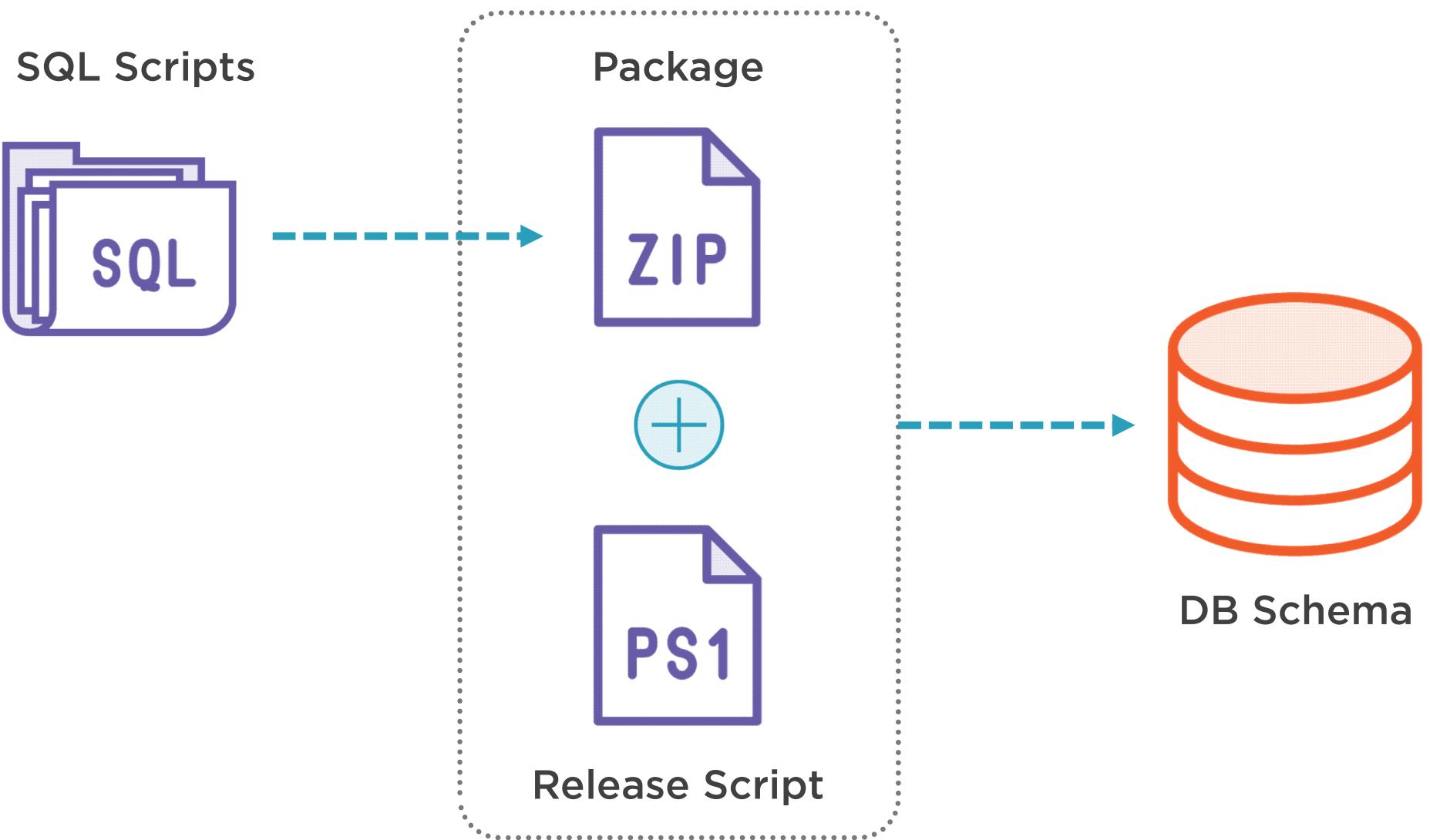
Running SQL Server Databases in Containers

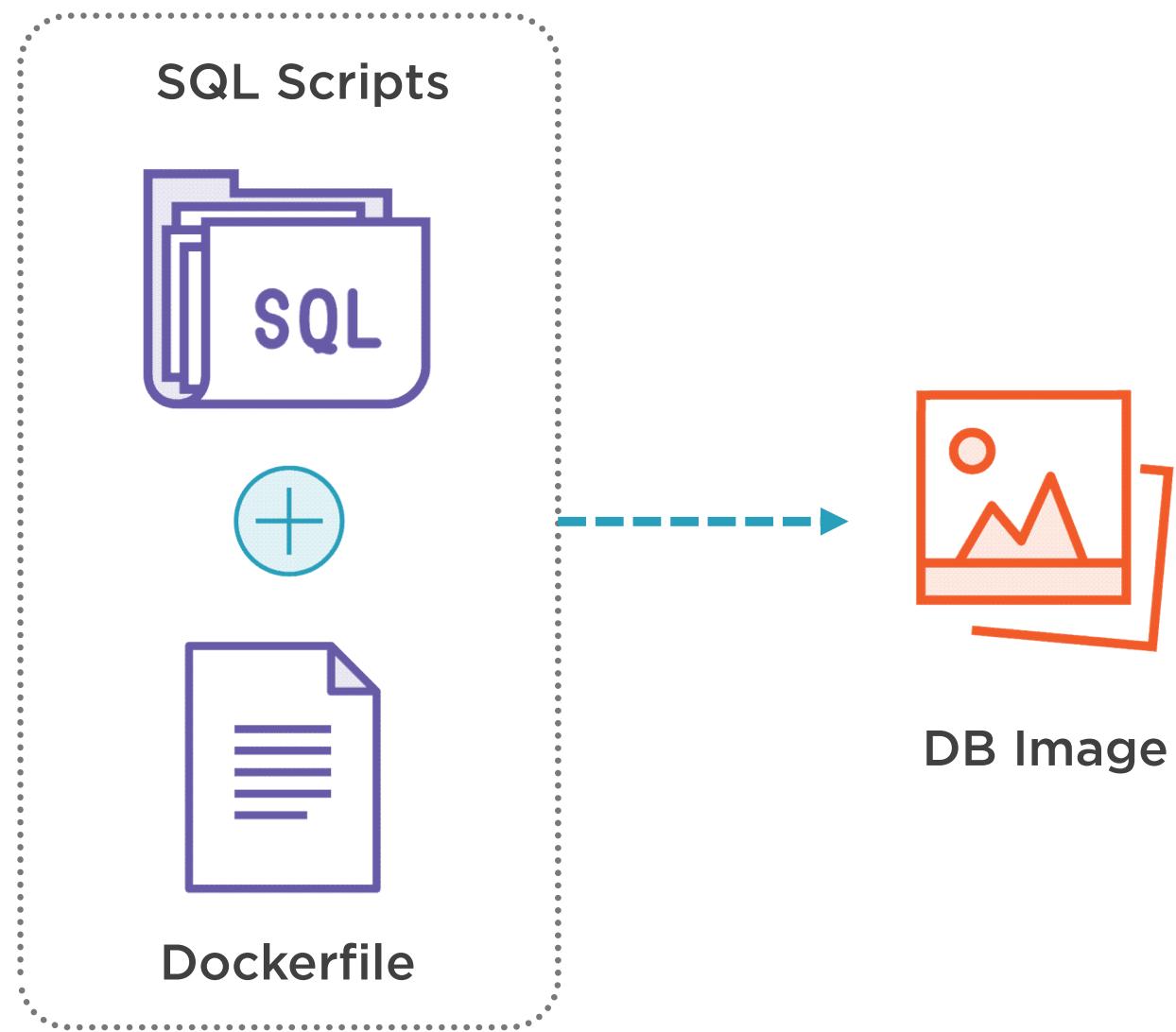


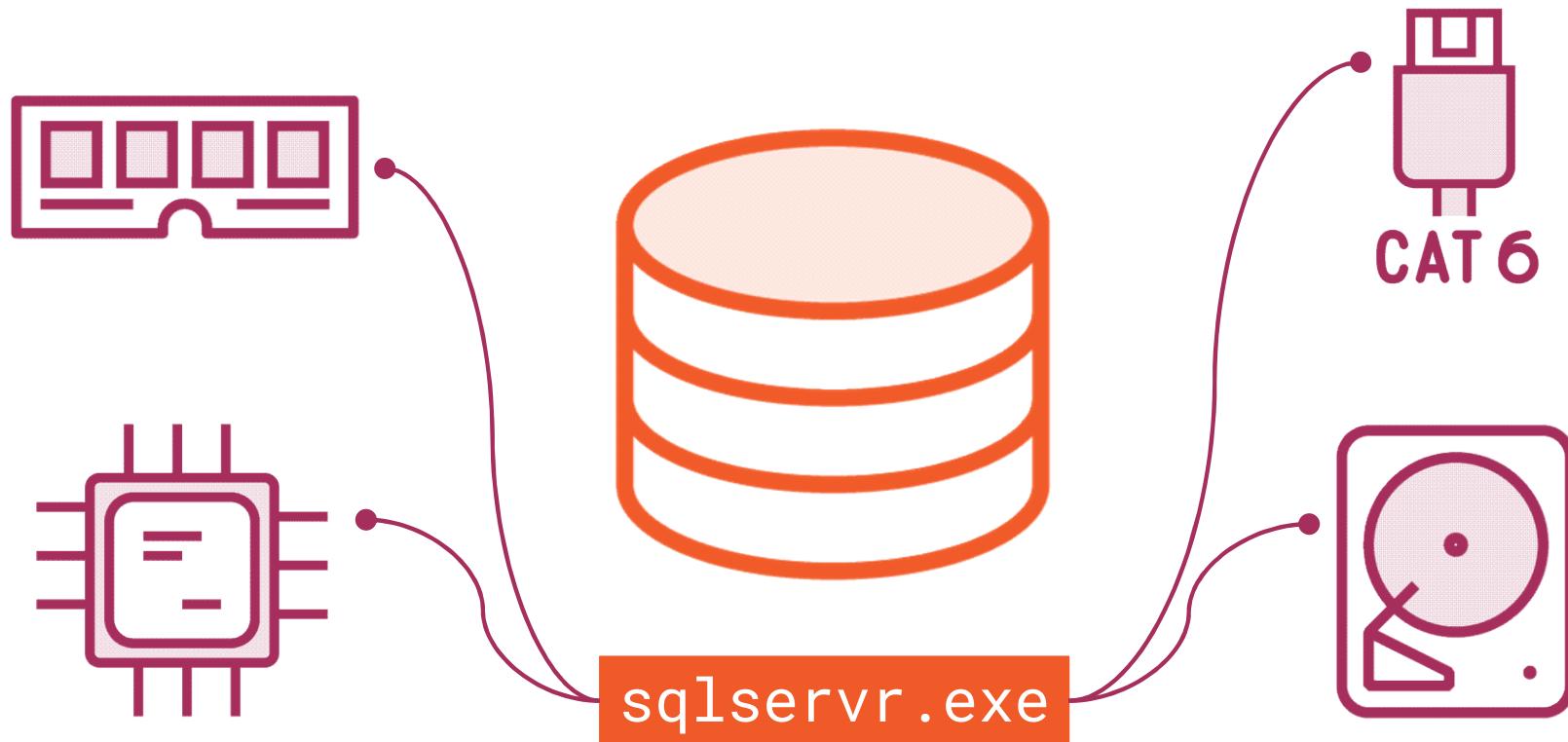
Elton Stoneman
DEVELOPER ADVOCATE

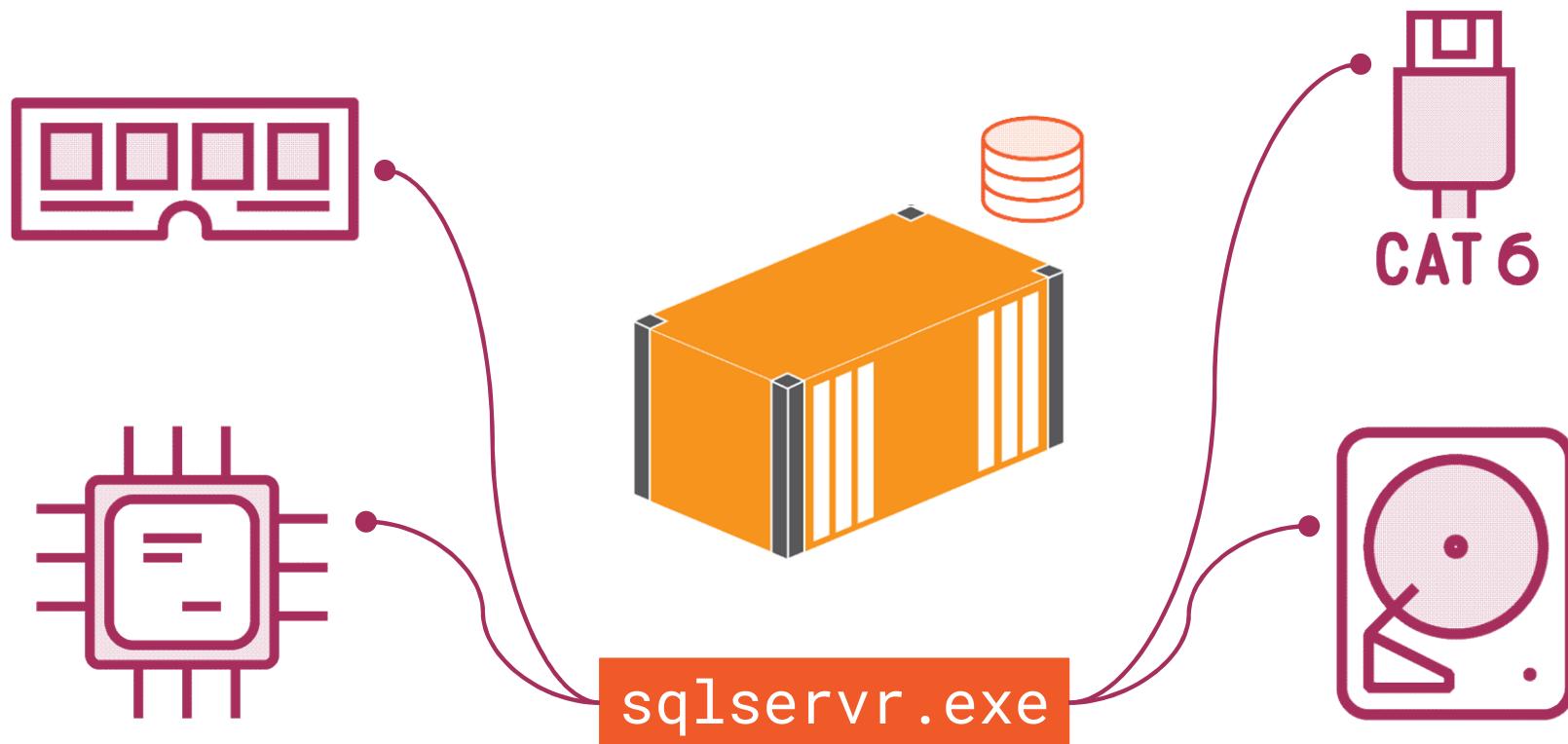
@EltonStoneman <https://blog.sixeyed.com>

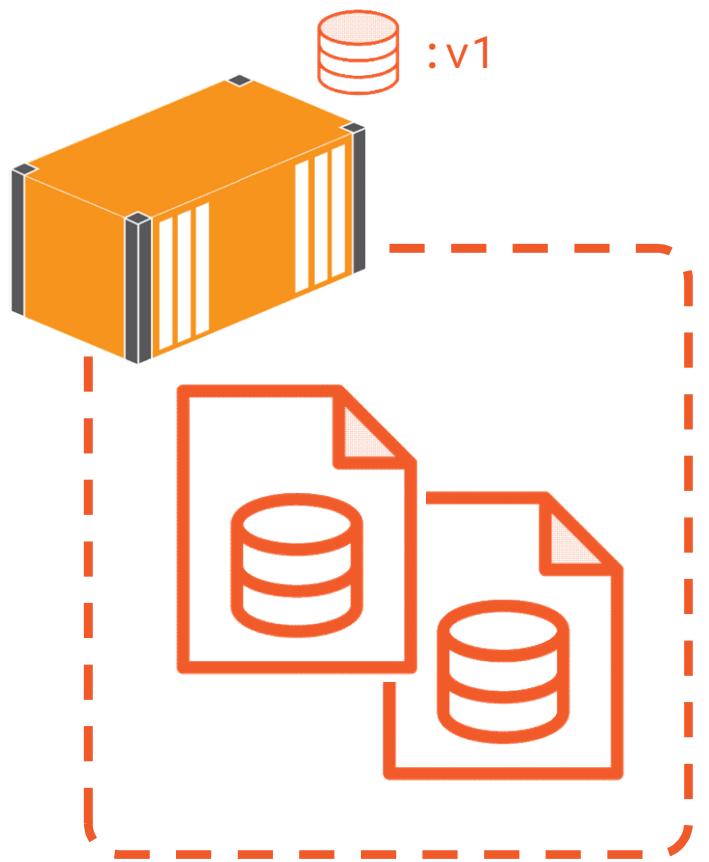


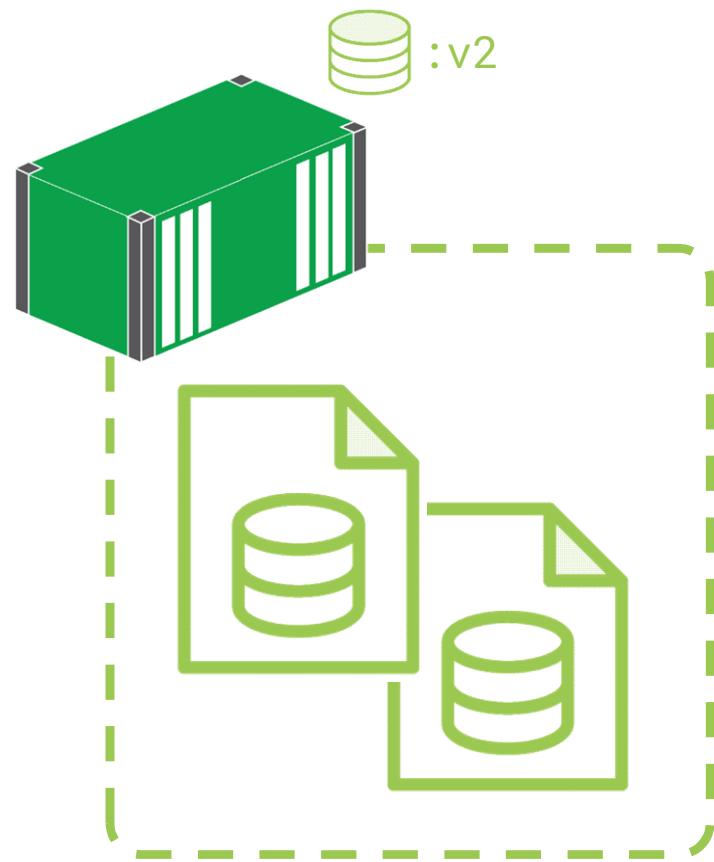
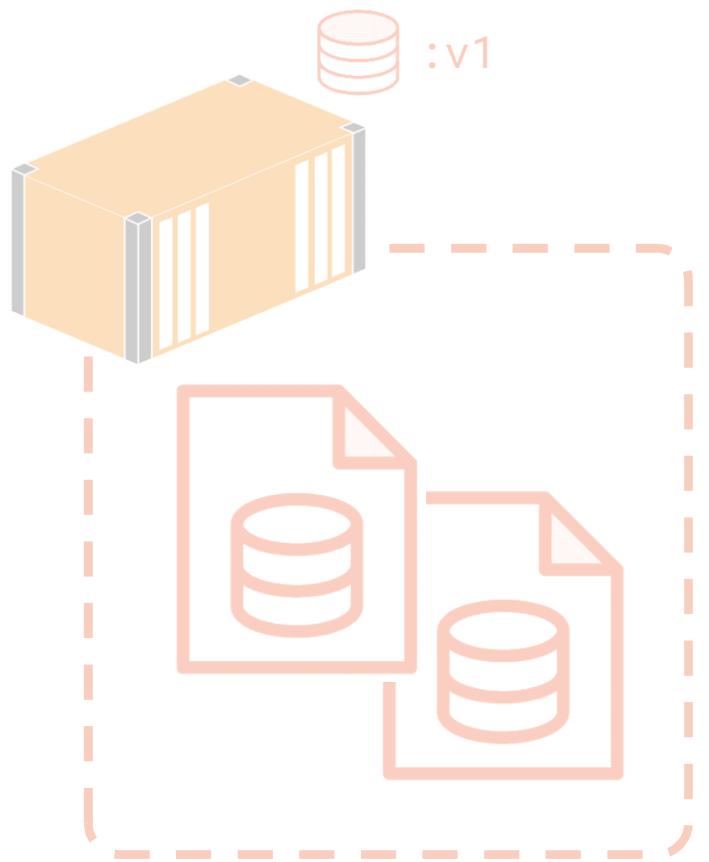




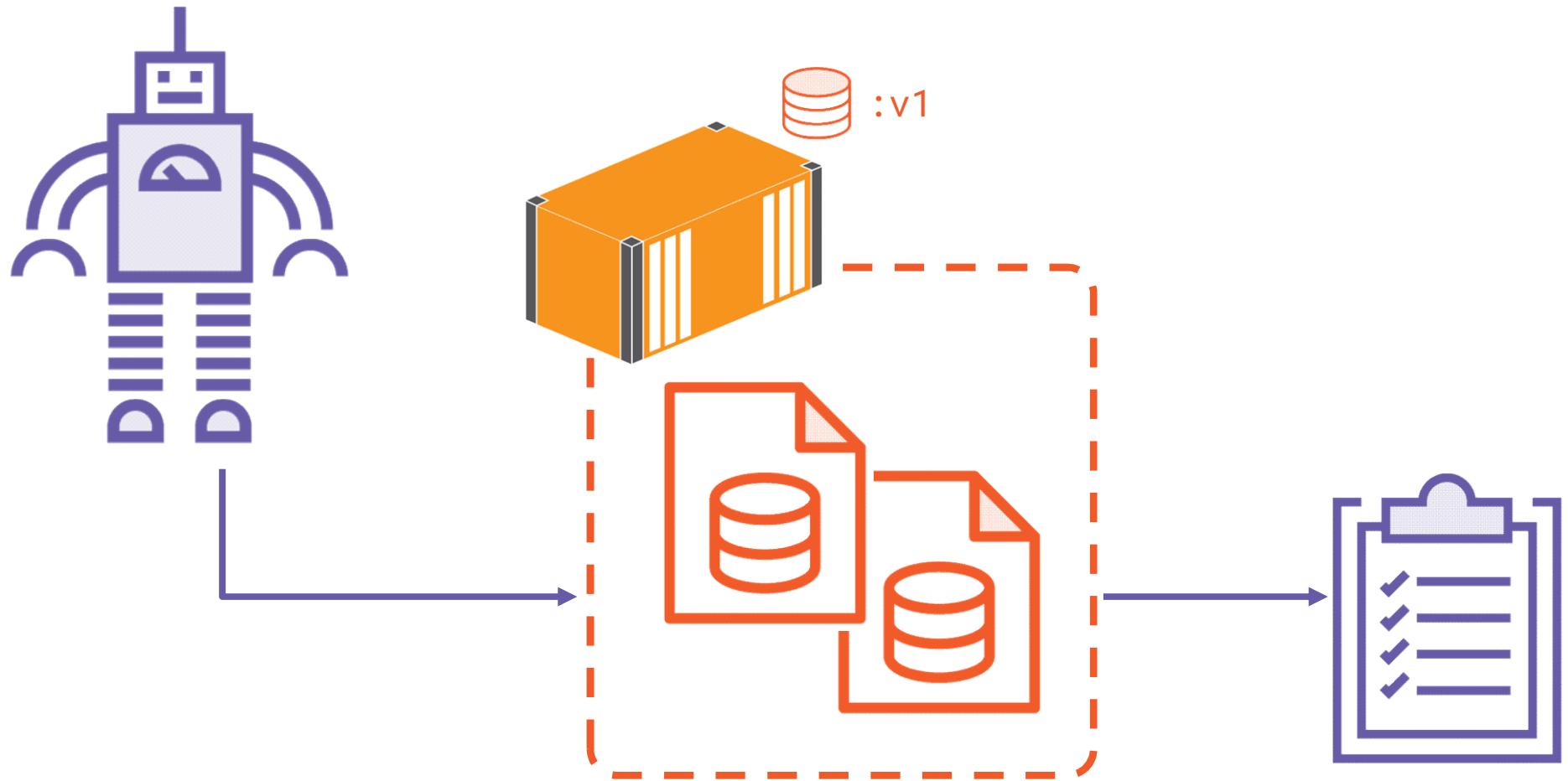


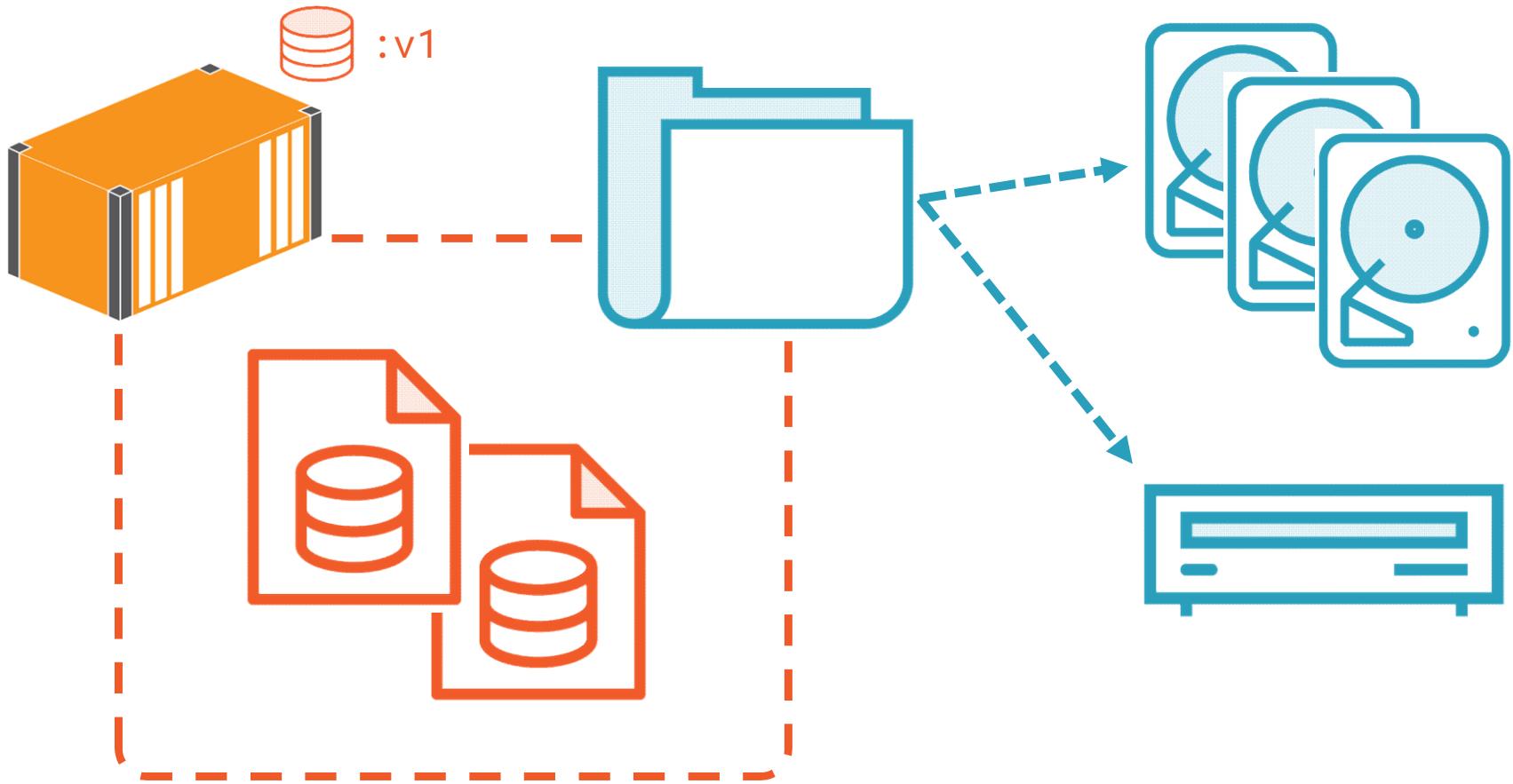


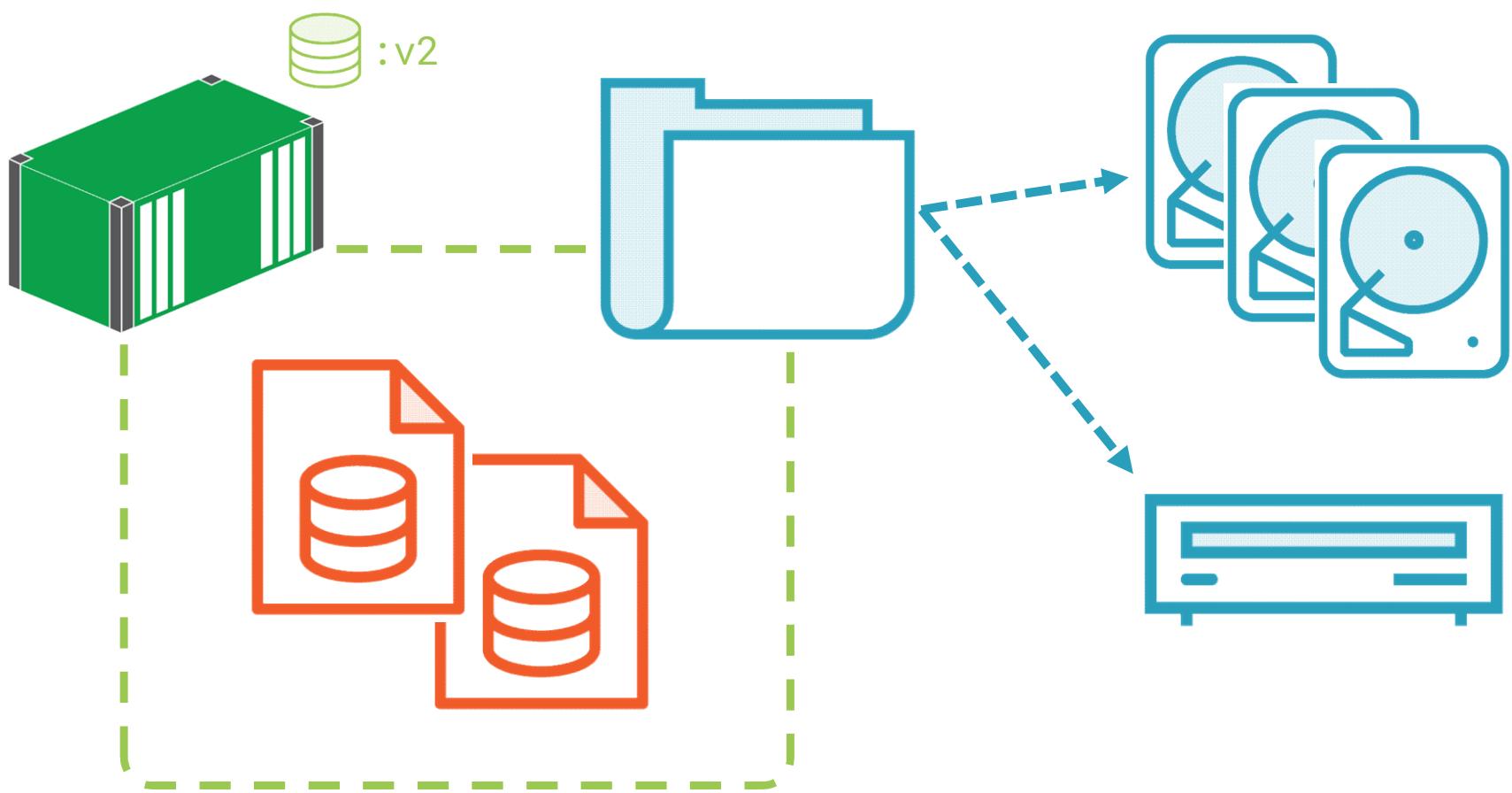


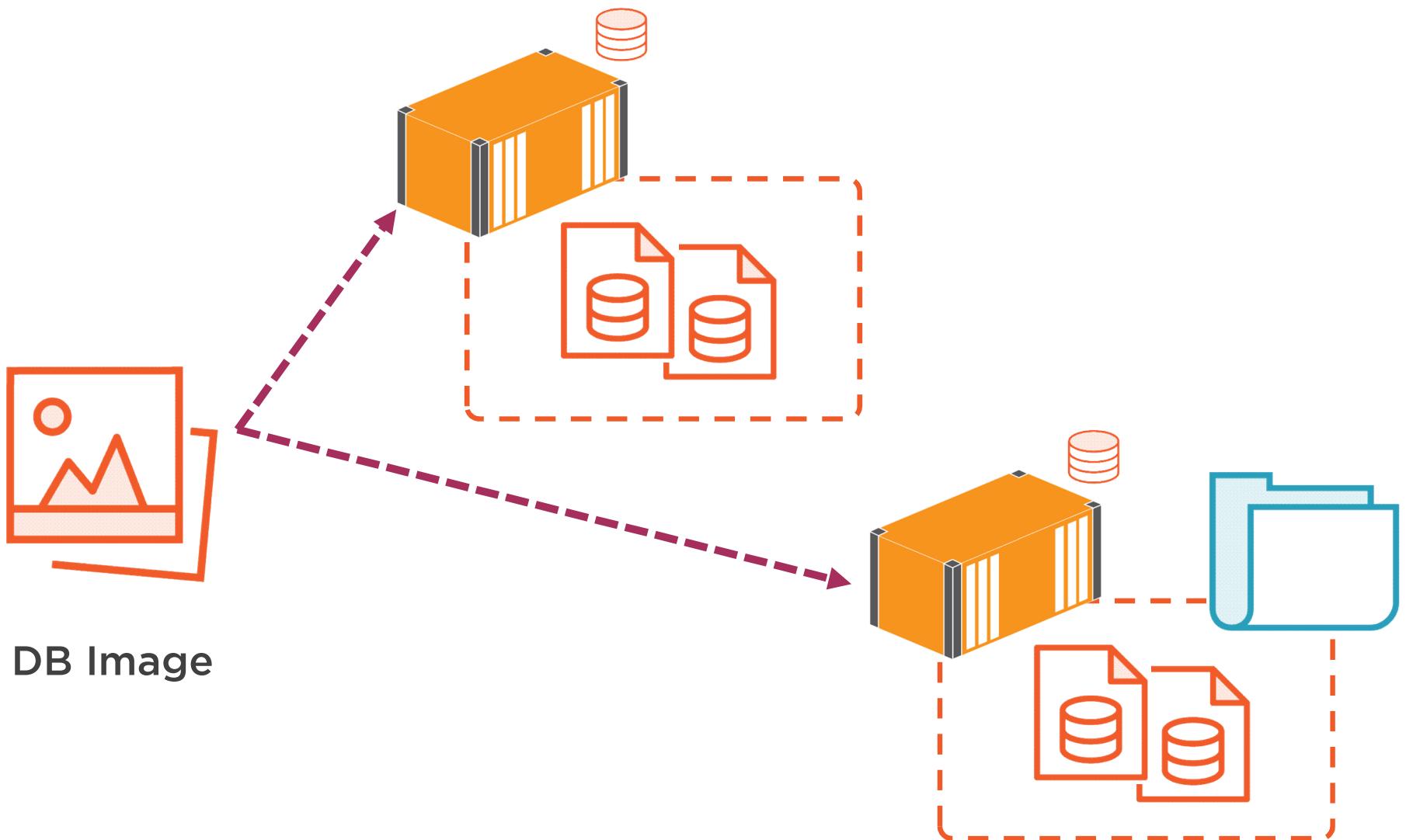


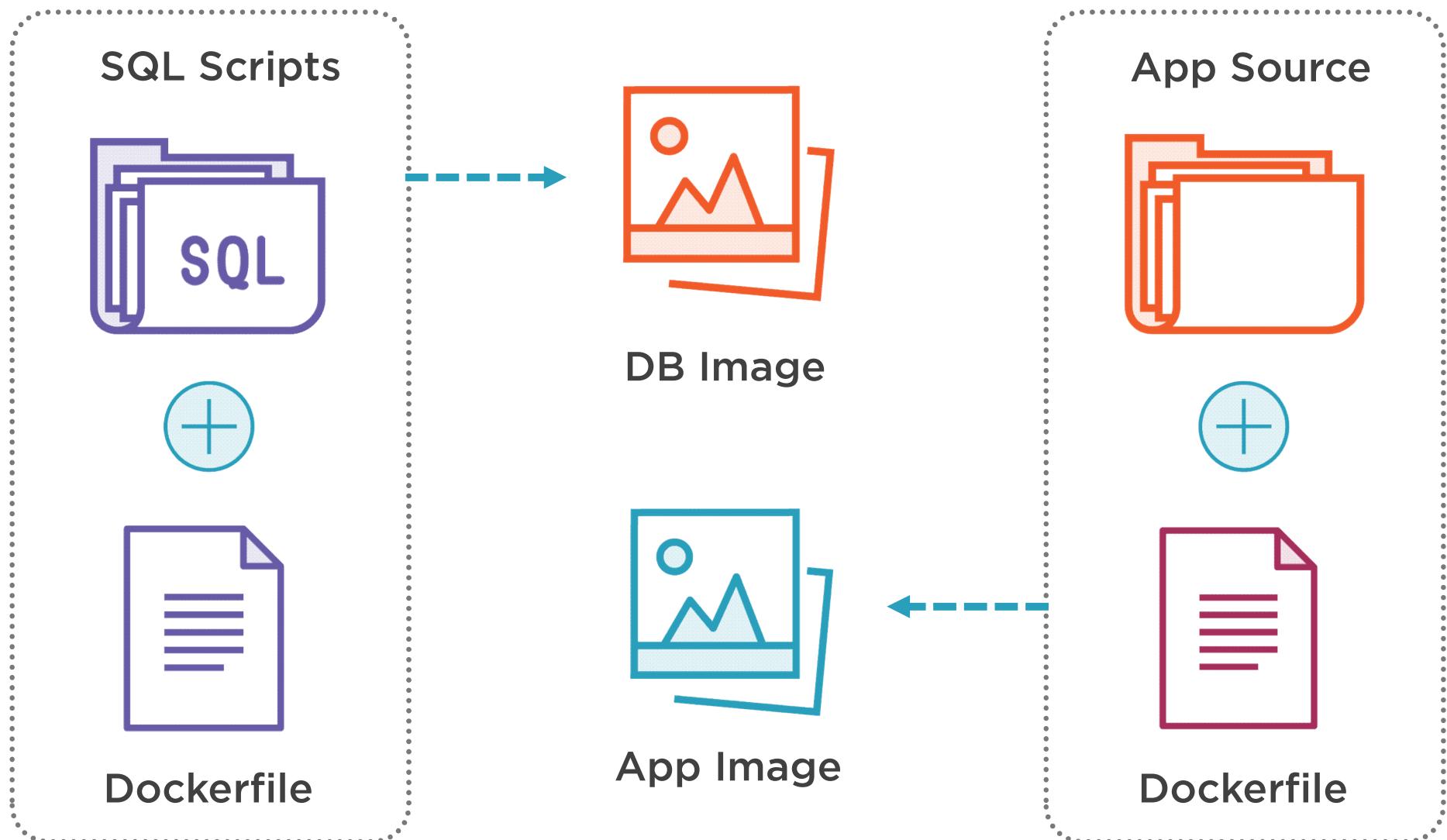


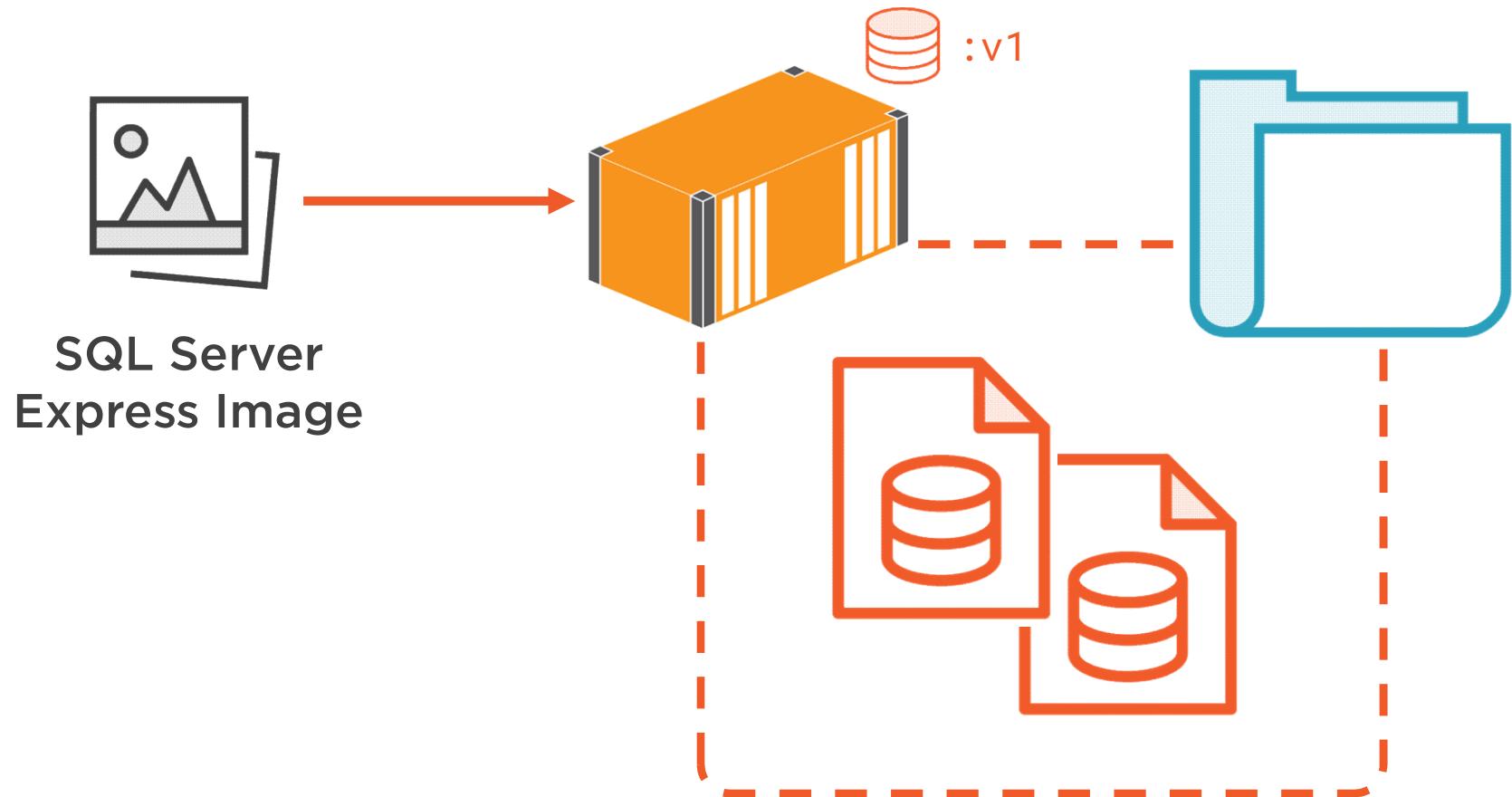


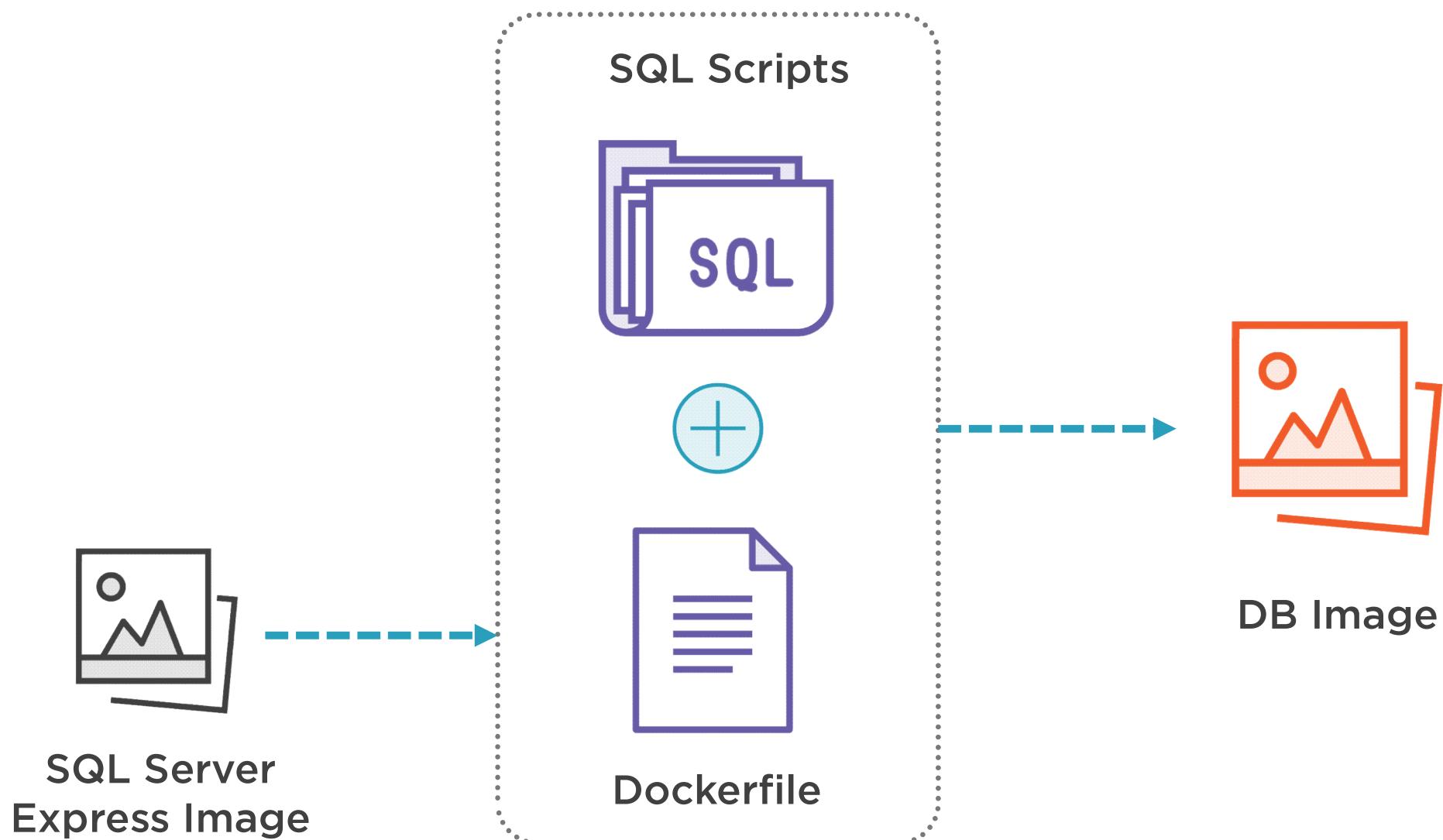




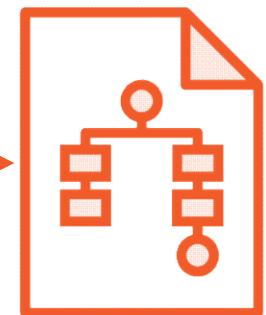






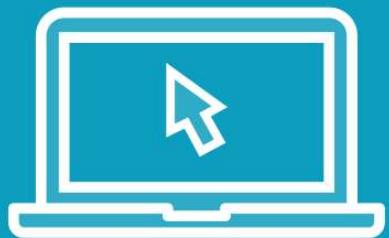


SQL Project



Dacpac

Demo



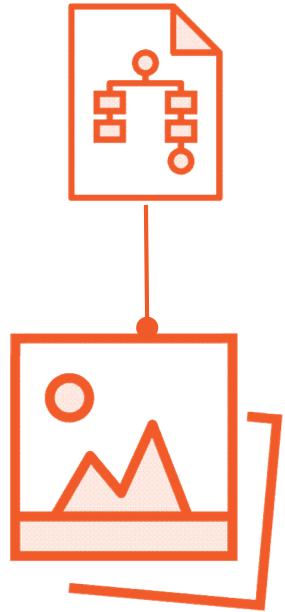
SQL Server Data Tools Projects

Building with Docker

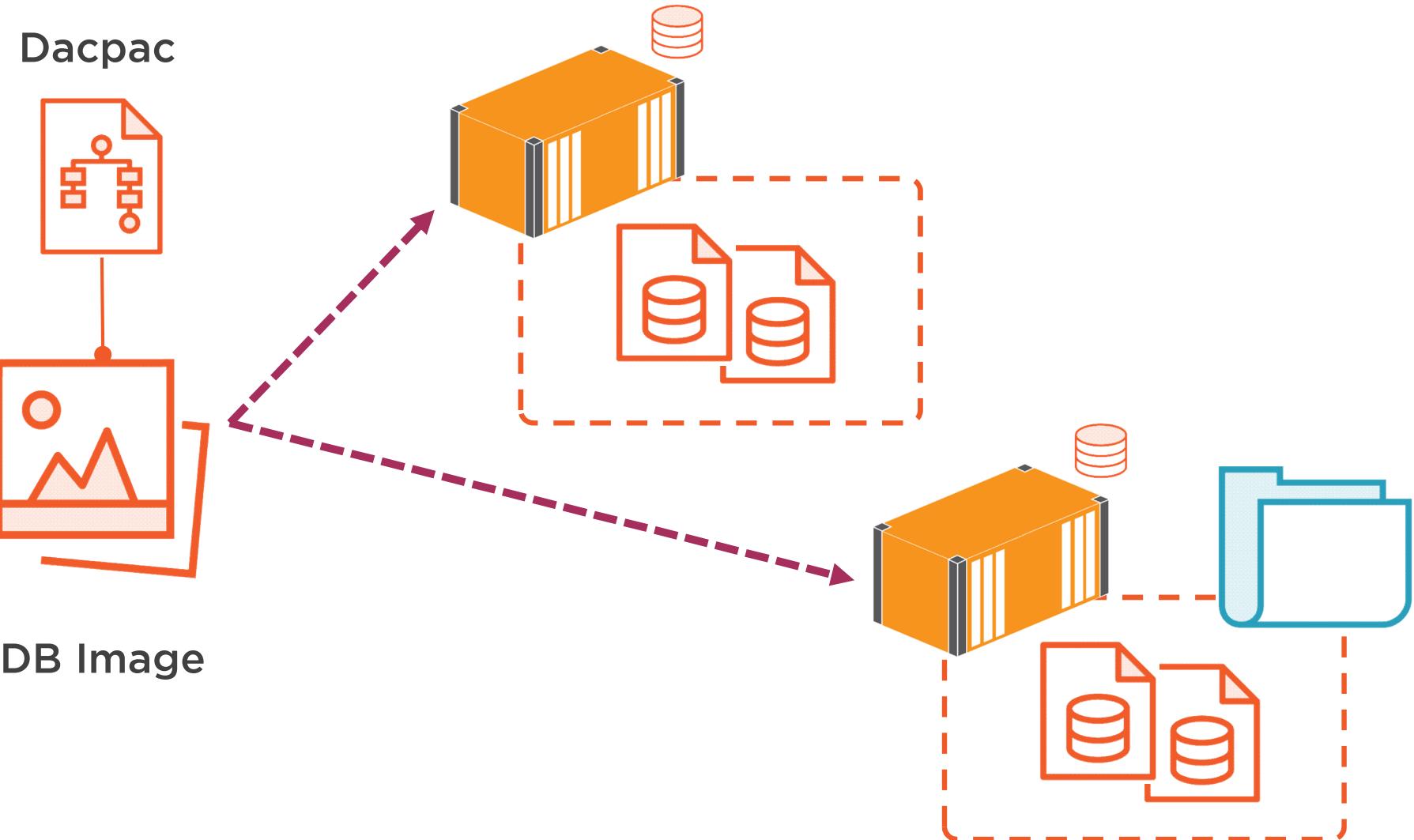
Writing the Dockerfile

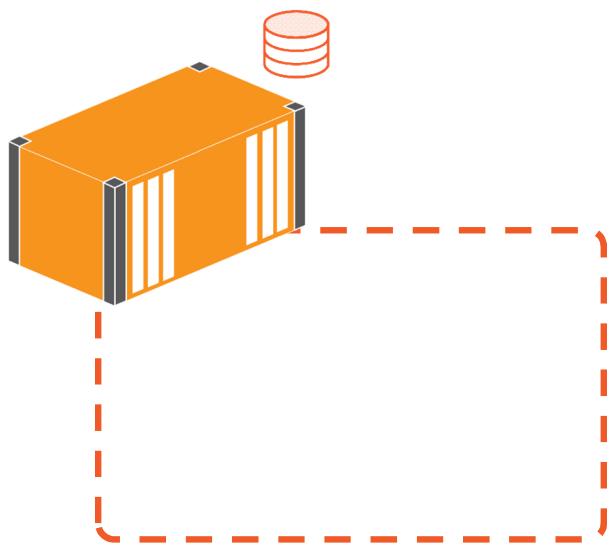
- Using a Dacpac

Dacpac



DB Image



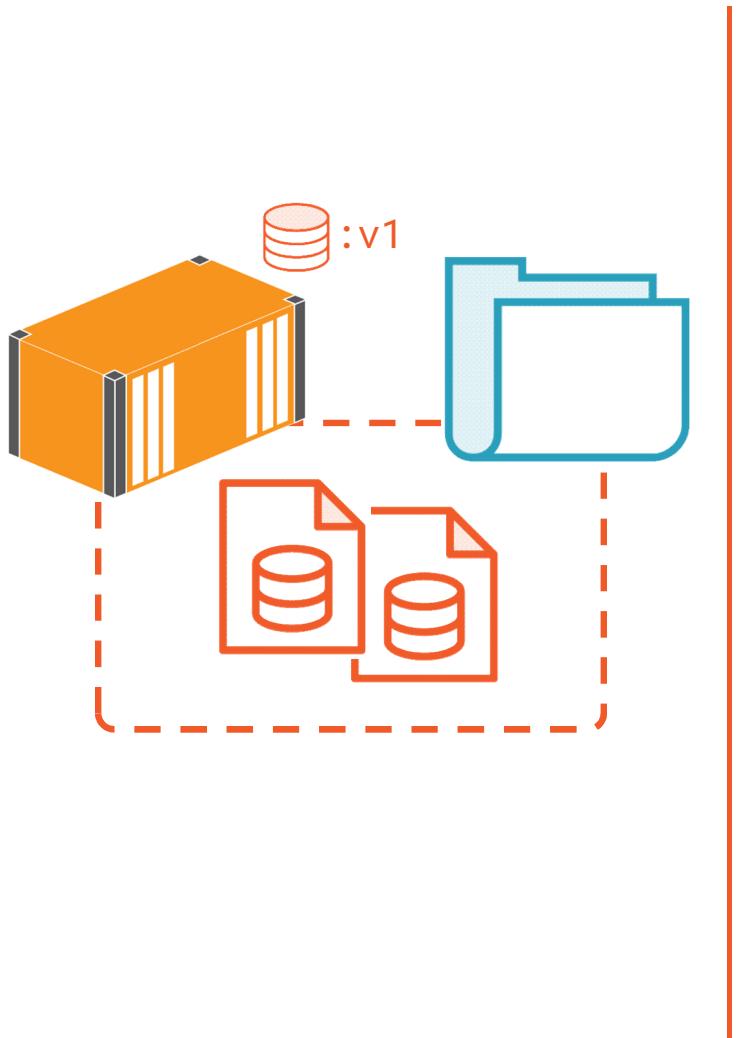


Workflow 1

- No existing database files

Generate SQL scripts

- Deploy schema
- Insert reference data

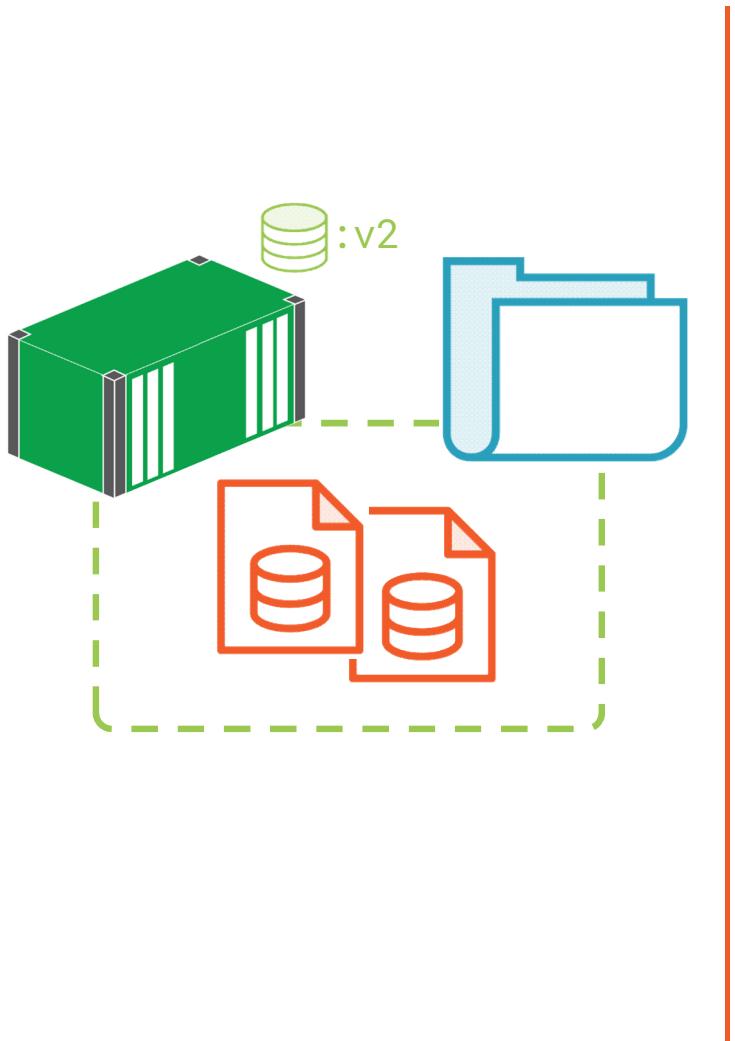


Workflow 2

- Existing database files
- Schema matches Dacpac

Attach database

- Attach files
- No schema updates

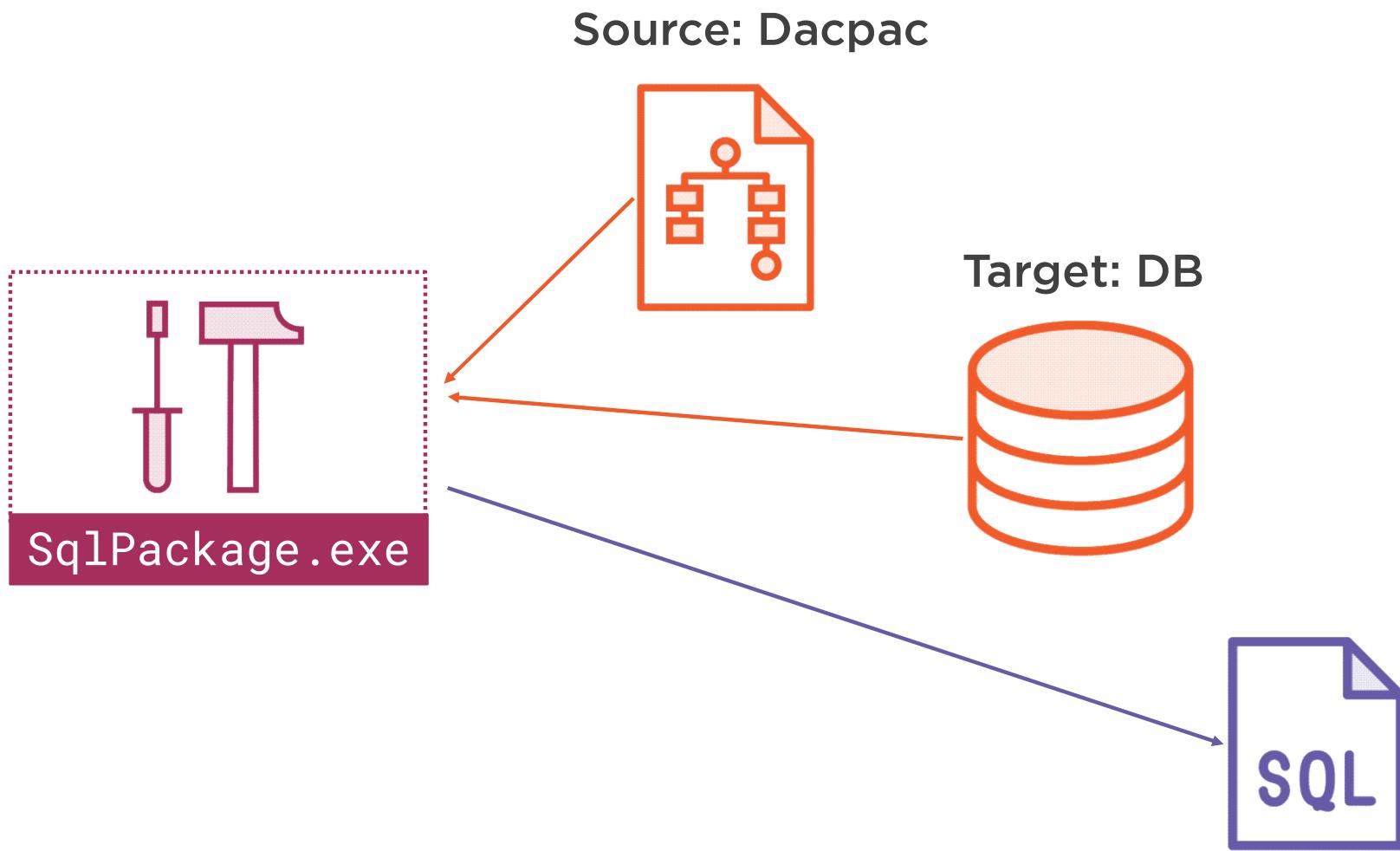


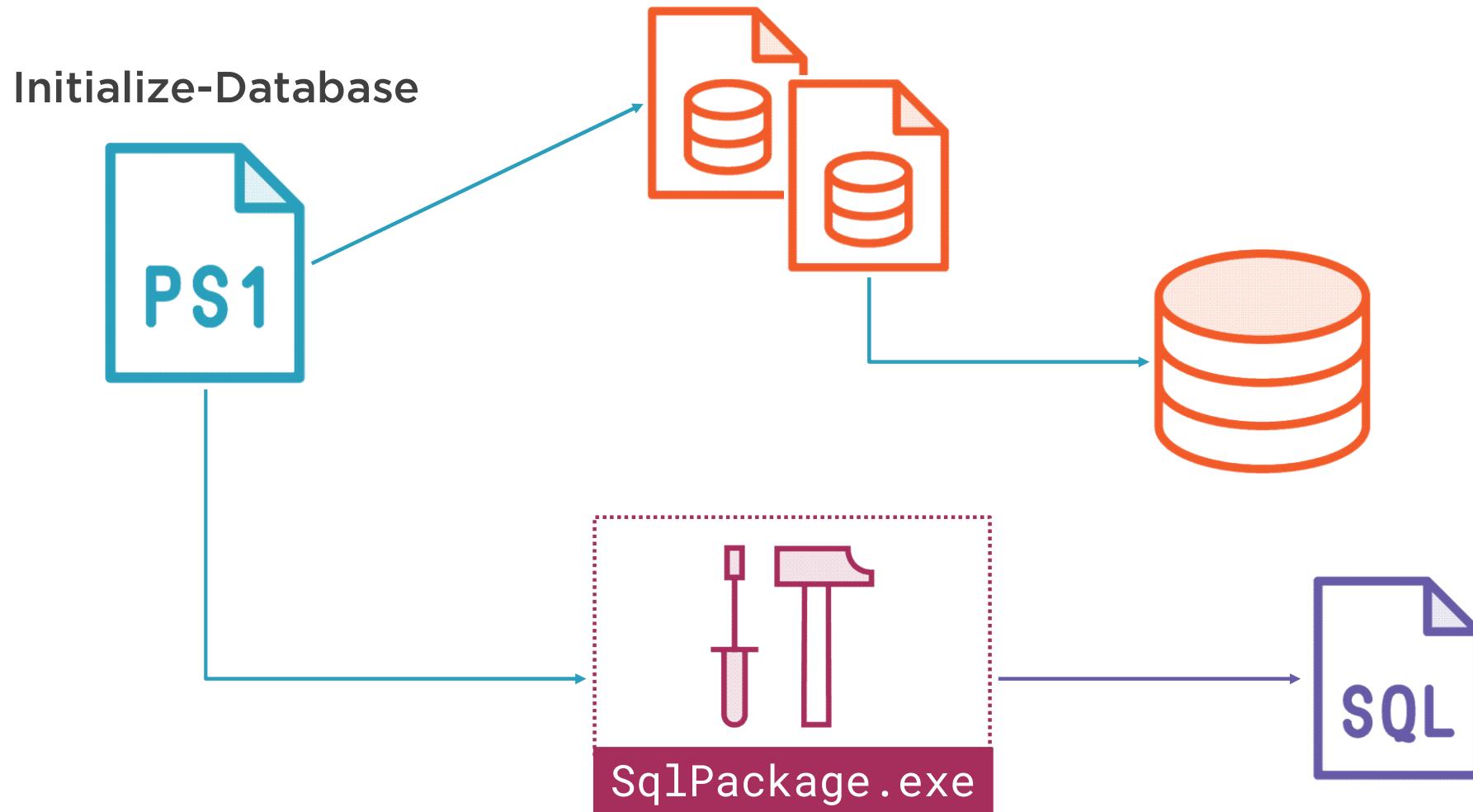
Workflow 3

- Existing database files
- Schema differs from Dacpac

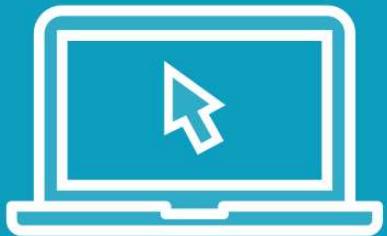
Upgrade database

- Attach files
- Compare schema
- Generate diff scripts





Demo

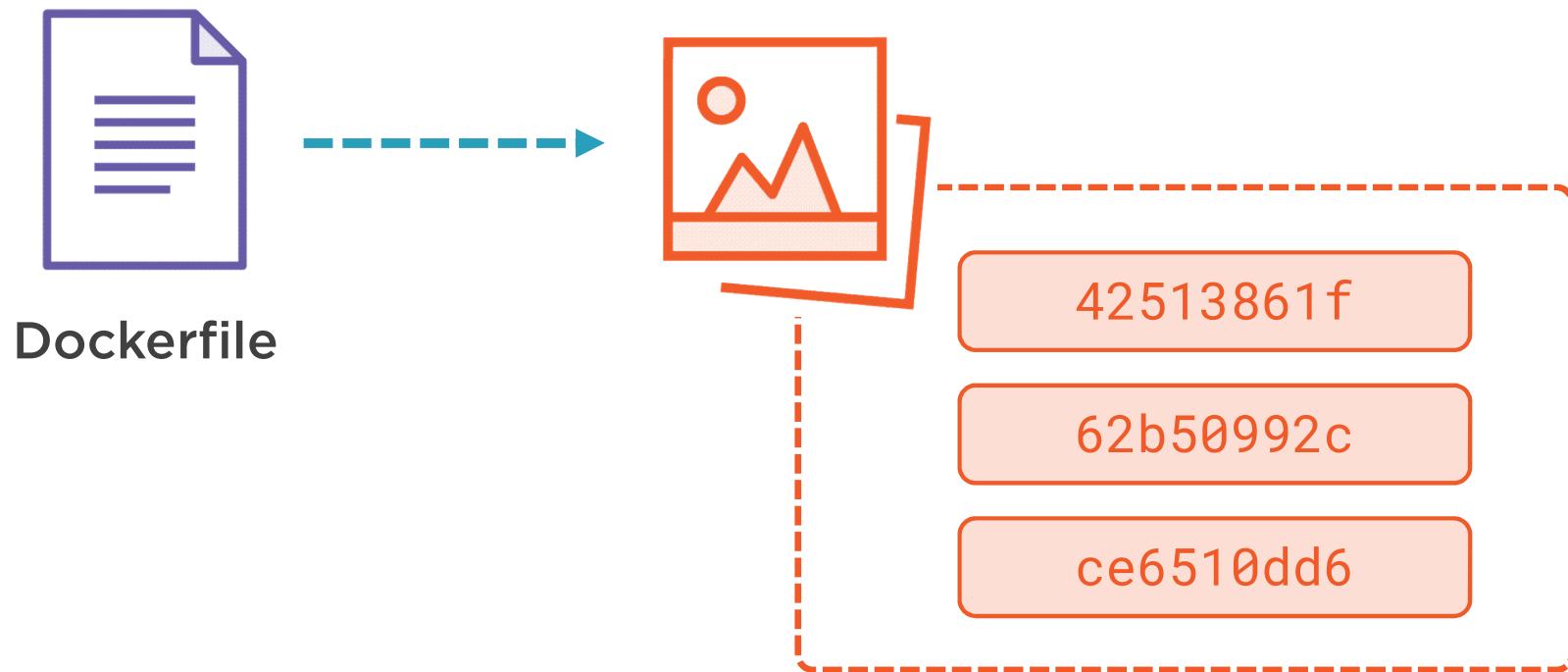


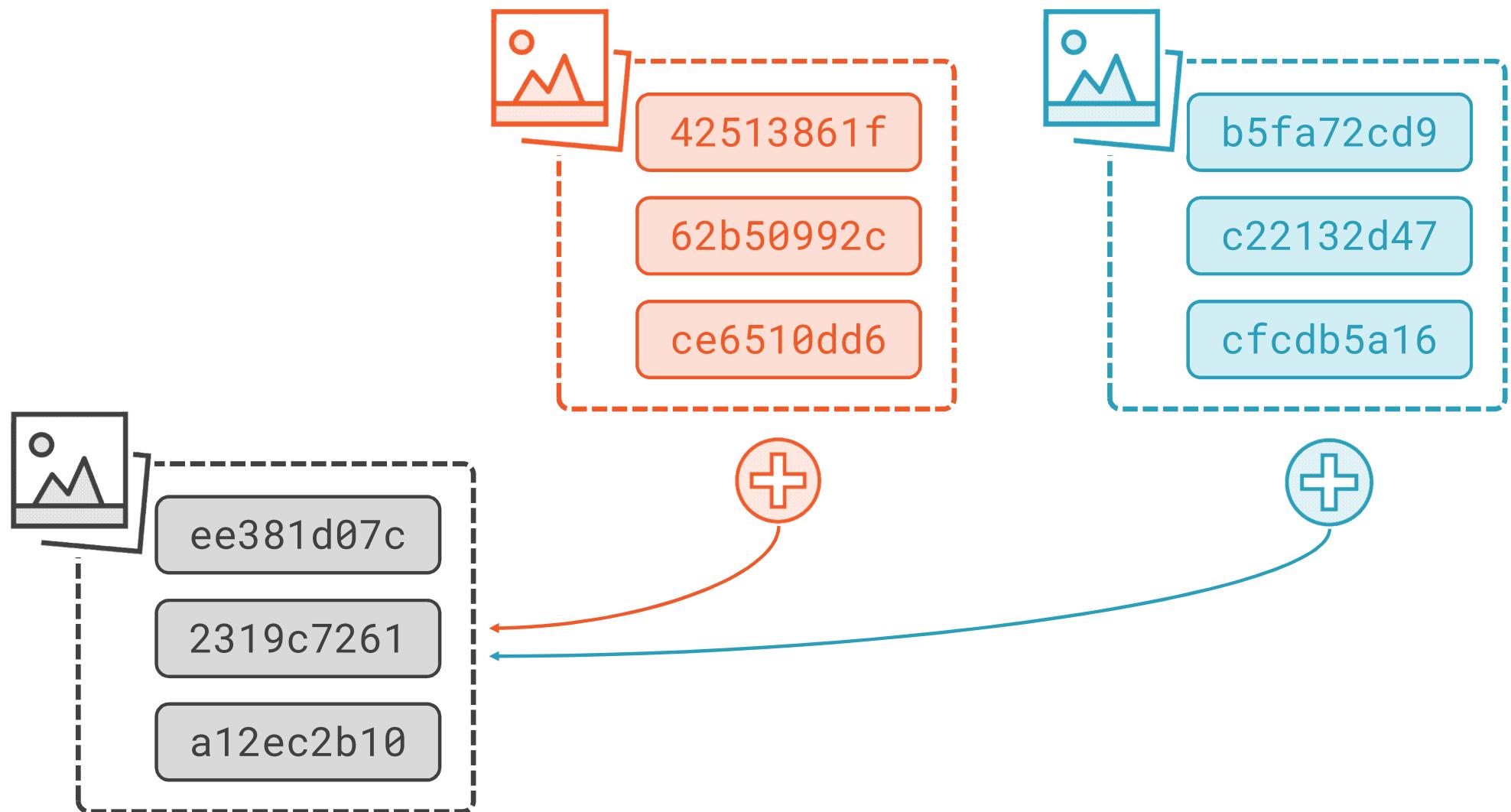
Initialize-Database script

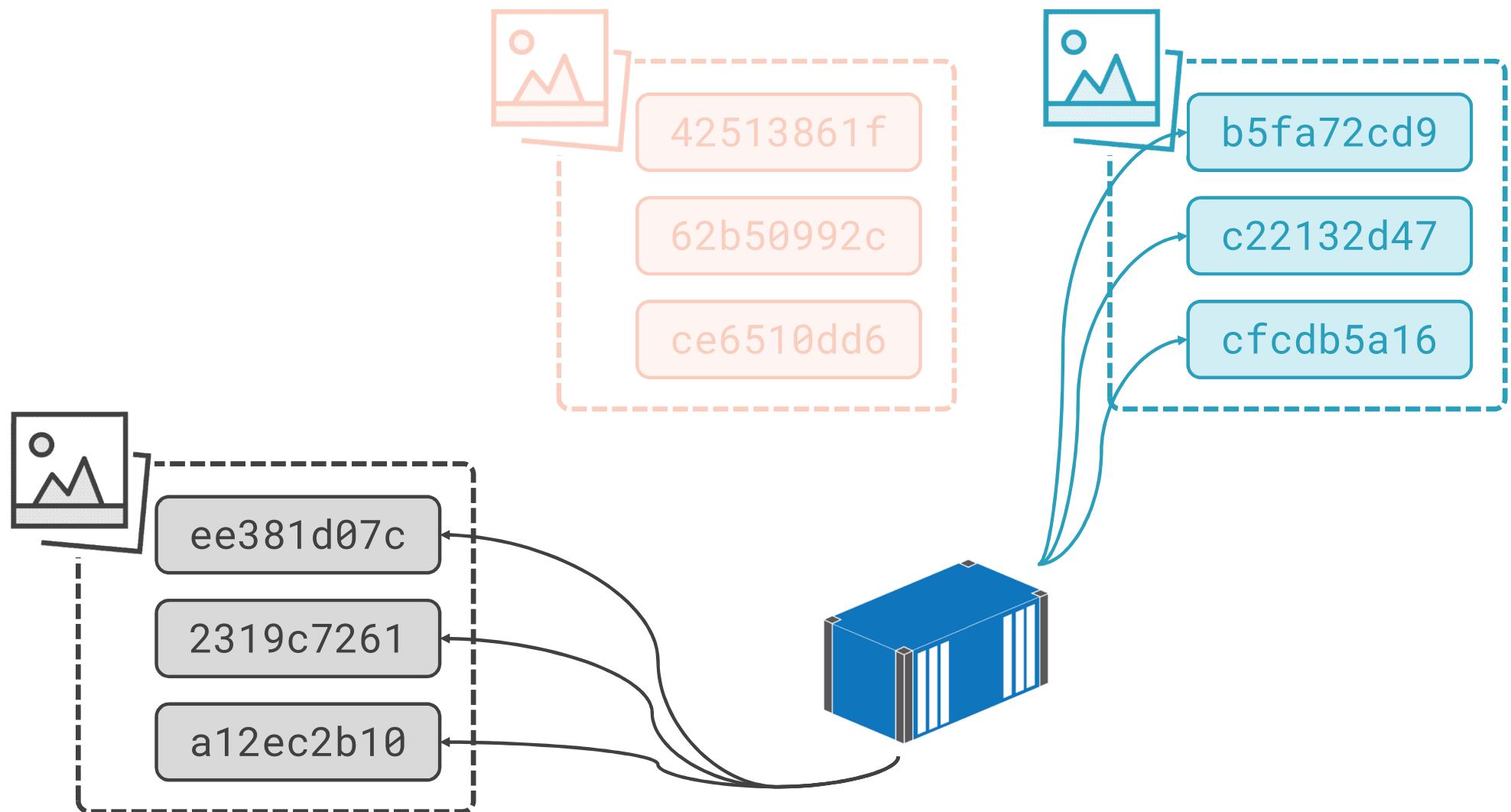
Check for database files

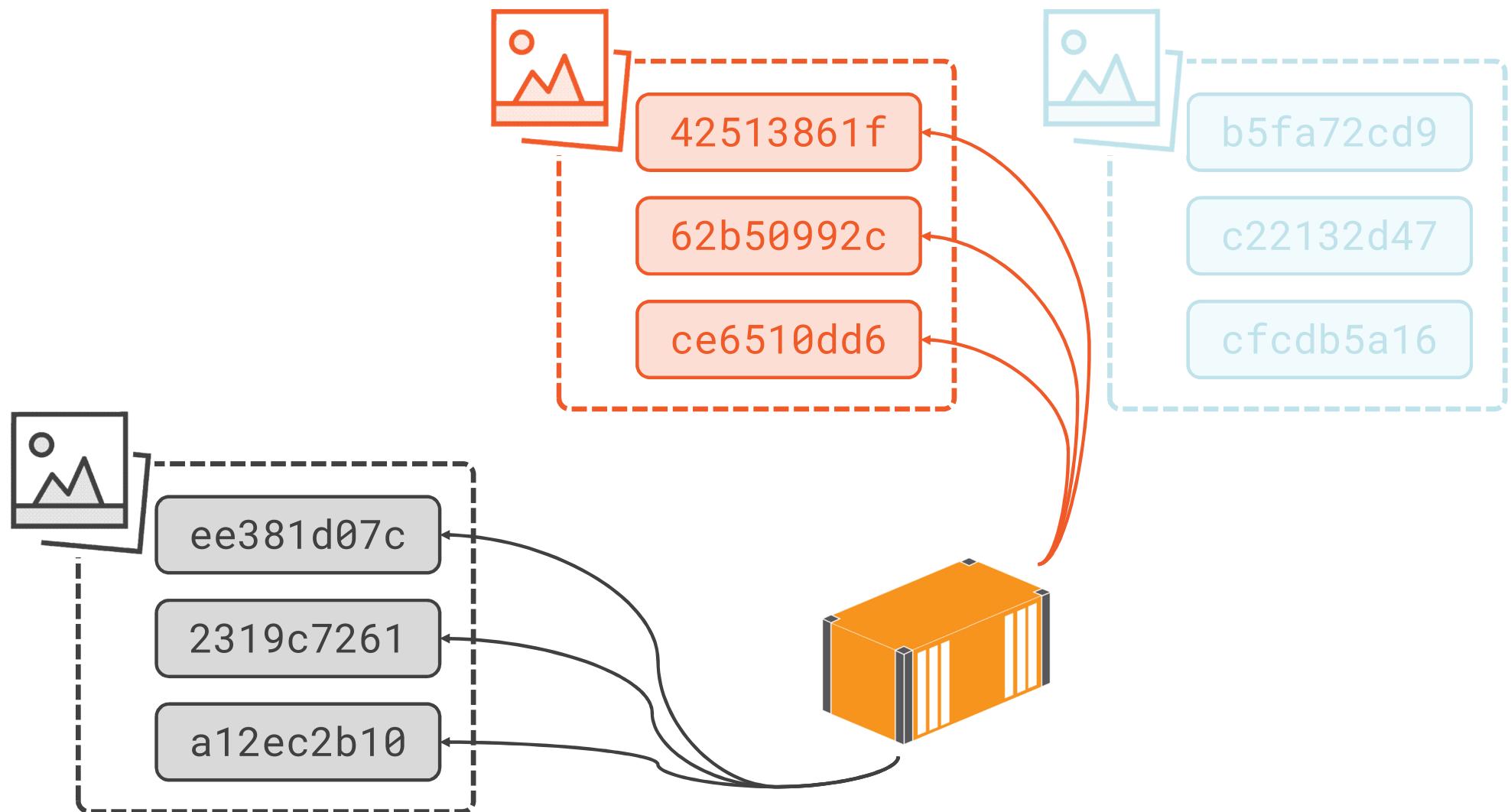
Run SqlPackage

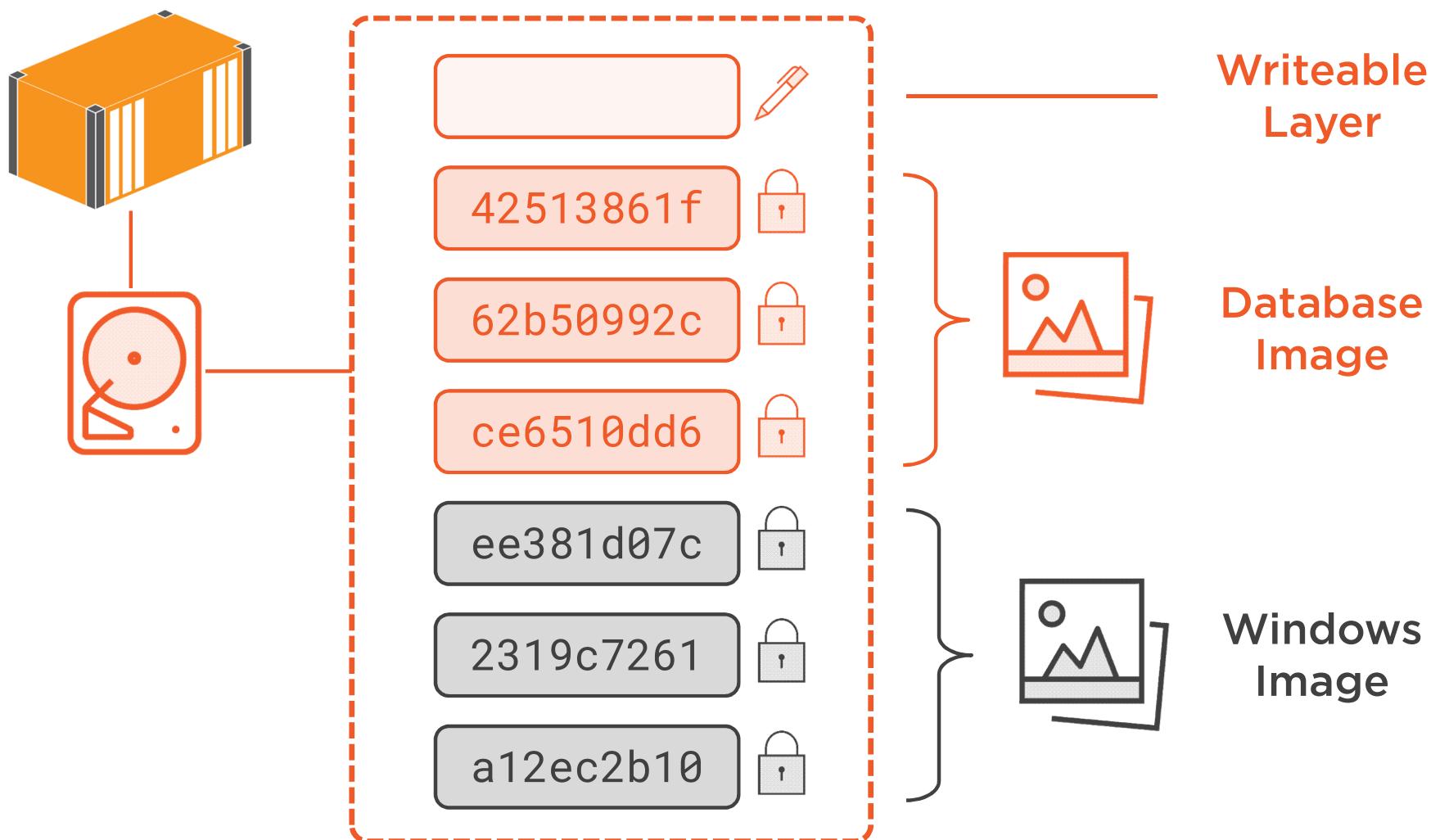
- Apply generated SQL scripts

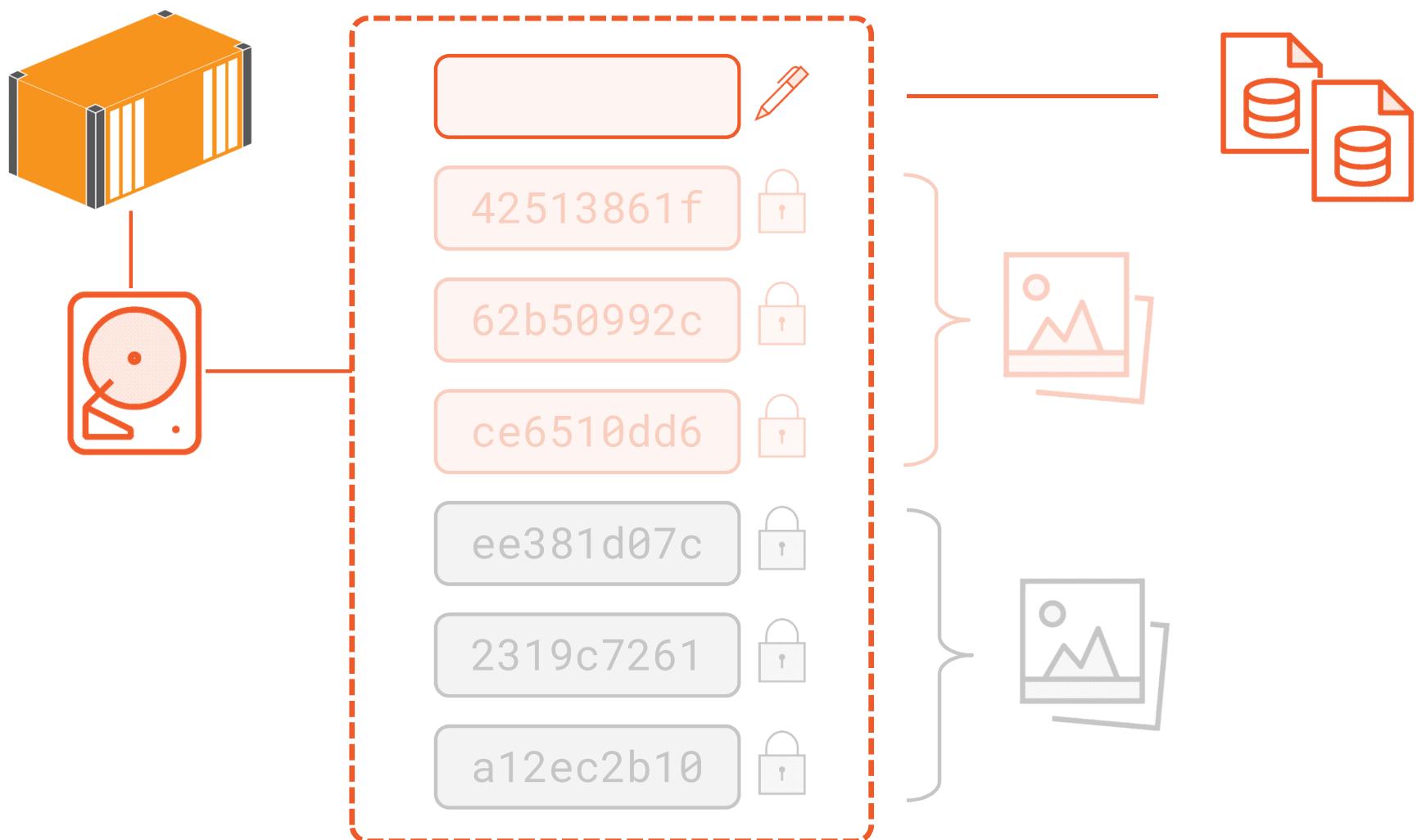


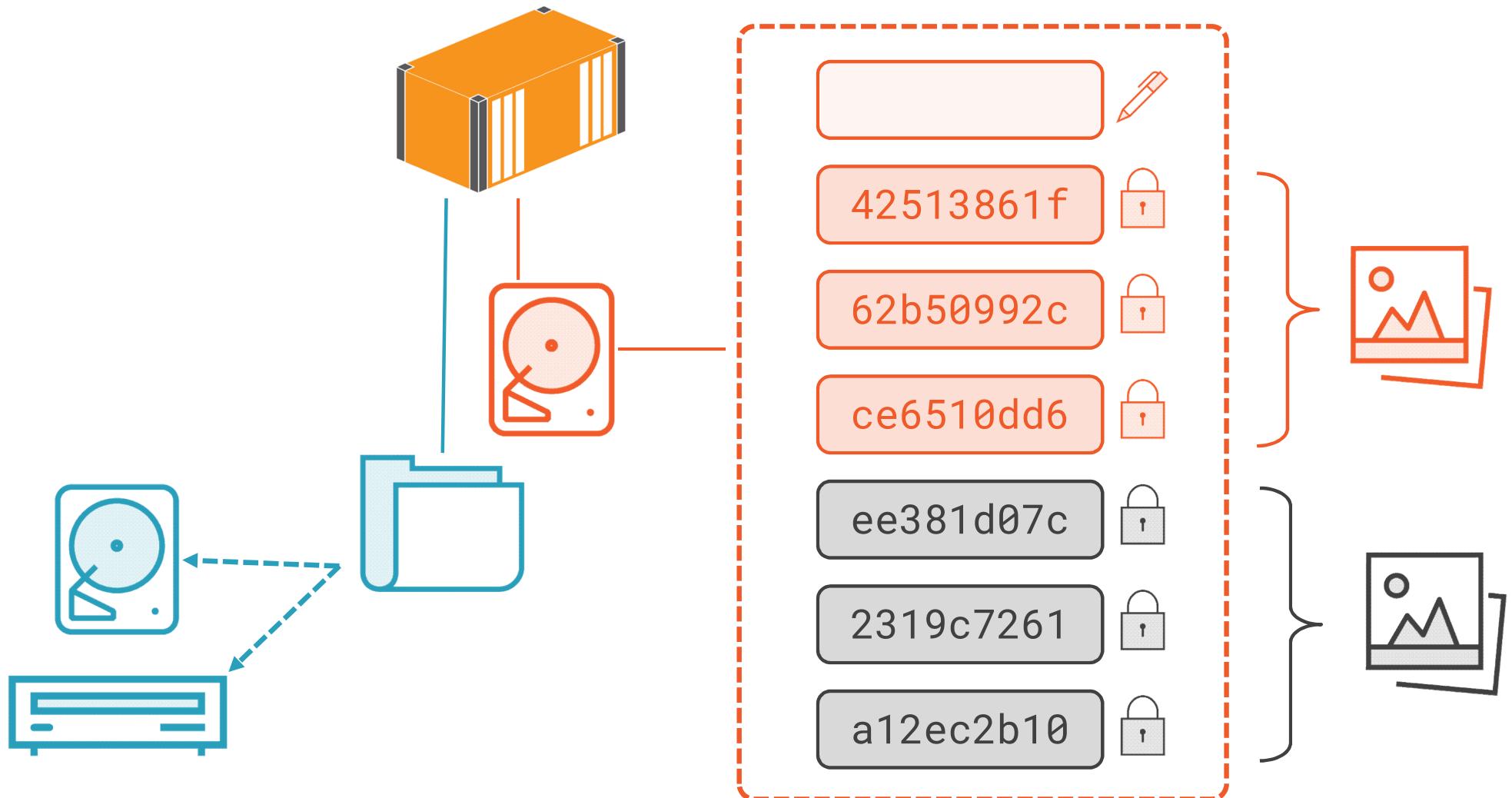




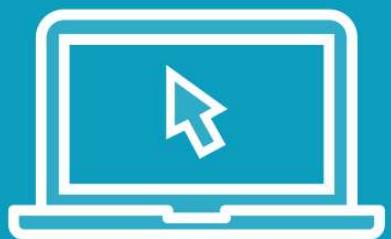








Demo



Disposable database containers

Persistent database containers

- Using Docker volumes

```
FROM sixeyed/msbuild:netfx-4.5.2-ssdt AS builder
WORKDIR C:\src\SignUp.Database
COPY src\SignUp.Database .
RUN msbuild SignUp.Database.sqlproj ...
```

Dockerfile for SQL Server

Stage 1: building the Dacpac

```
FROM microsoft/mssql-server-windows-express

ENV ACCEPT_EULA=Y \
    DATA_PATH="C:\data" \
    sa_password="DockerCon!!!"

VOLUME ${DATA_PATH}
```

Dockerfile for SQL Server

Stage 2: packaging the schema

```
WORKDIR C:\init  
COPY docker\db\Initialize-Database.ps1 .  
CMD powershell ./Initialize-Database.ps1  
COPY --from=builder C:\src\SignUp.Database\...
```

Dockerfile for SQL Server

Stage 2: packaging the schema

```
if ((Test-Path $mdfPath) -eq $true) {  
    $sqlcmd = "... CREATE DATABASE SignUp..."  
    # ...  
    Invoke-Sqlcmd -Query $sqlcmd  
}
```

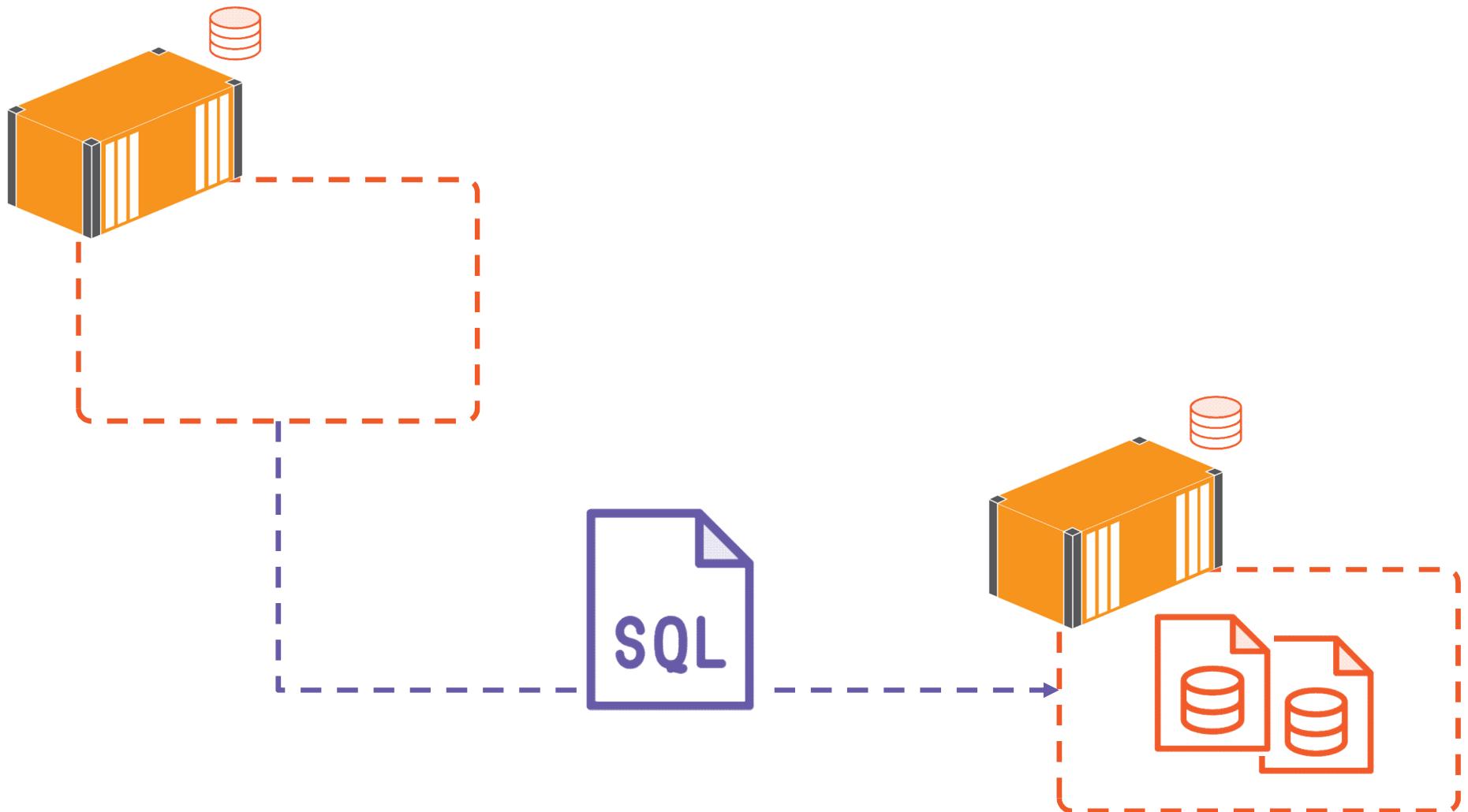
Database Initialization

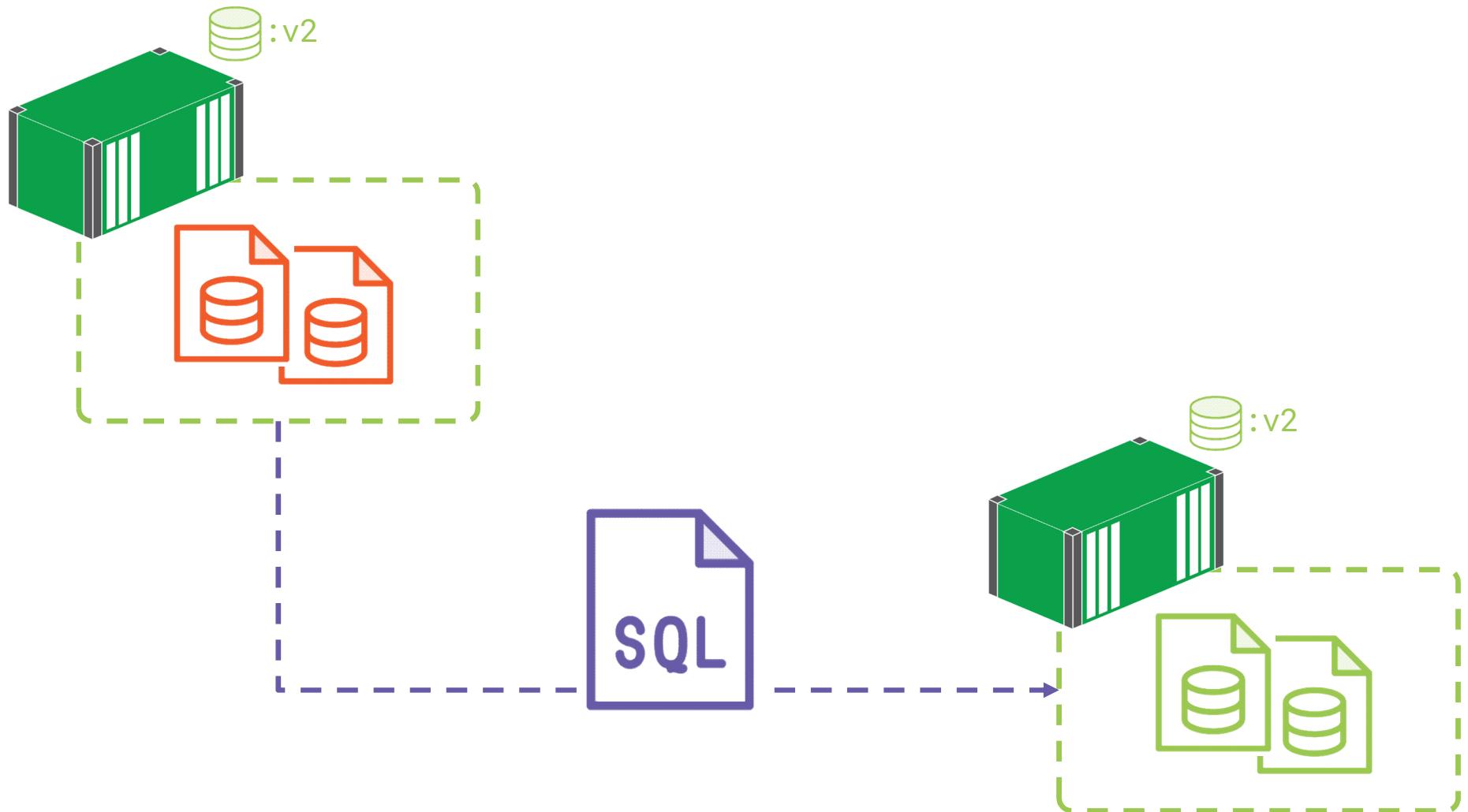
Attach existing data files

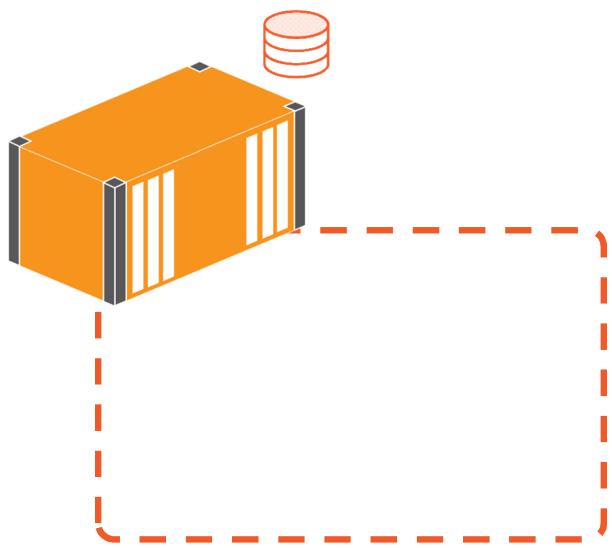
```
& $SqlPackagePath `'  
/sf:SignUp.Database.dacpac `'  
/a:Script /op:deploy.sql `'  
/tsn:.\SQLEXPRESS /tdn:SignUp /tu:sa
```

Database Initialization

Generate deployment SQL script







Workflow 1

- Disposable database

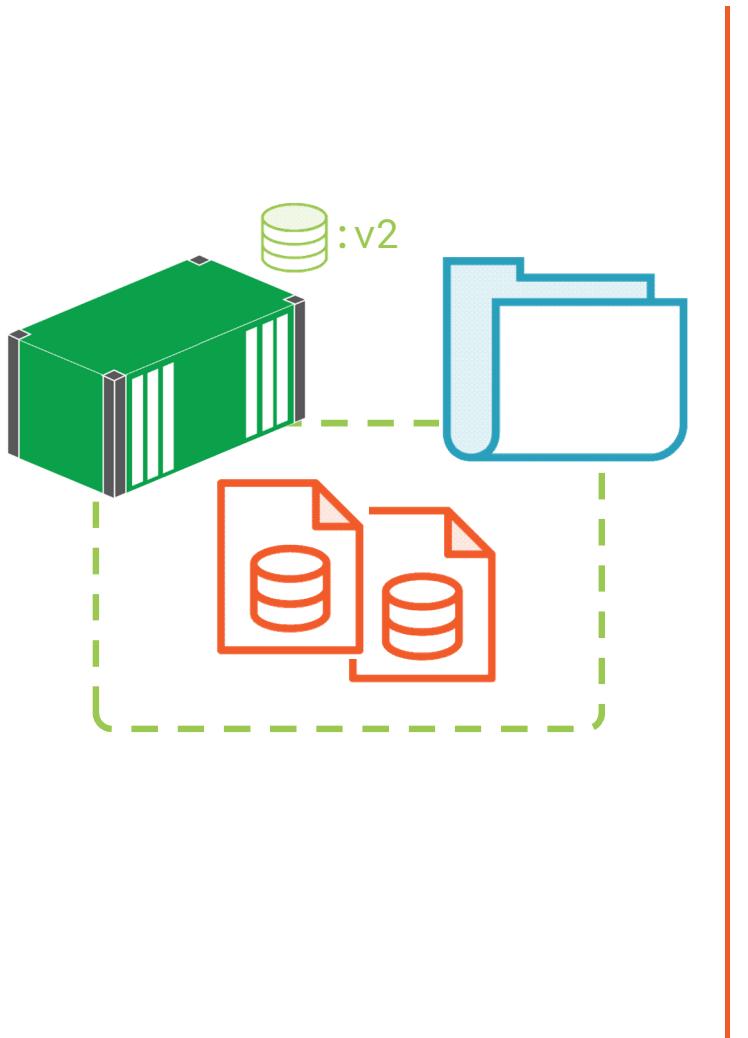




Workflow 2

- Windows & SQL Server updates



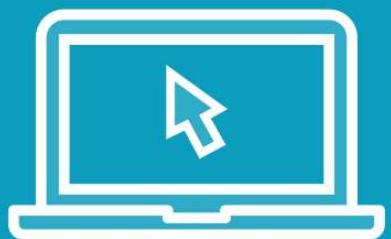


Workflow 3

- Database schema updates

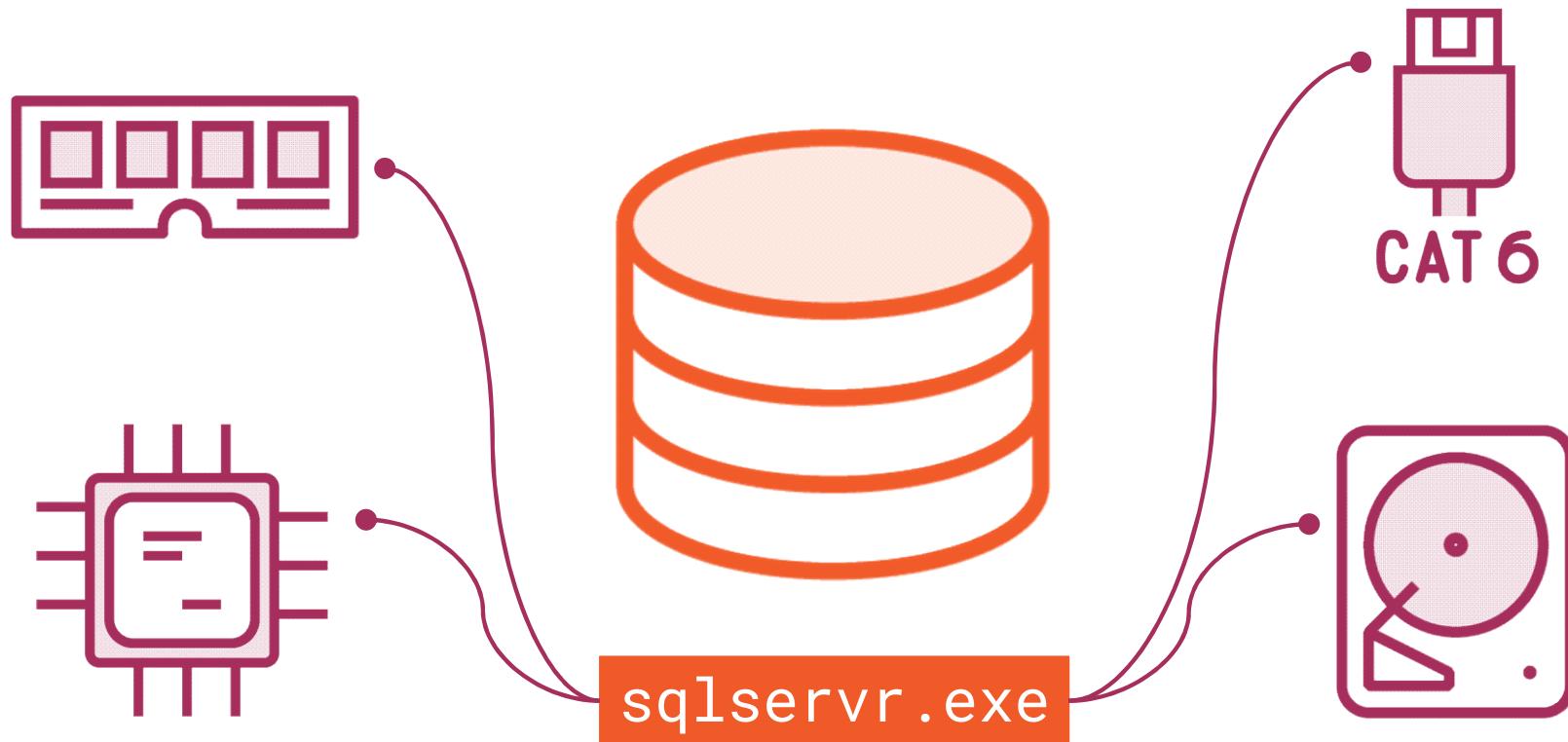


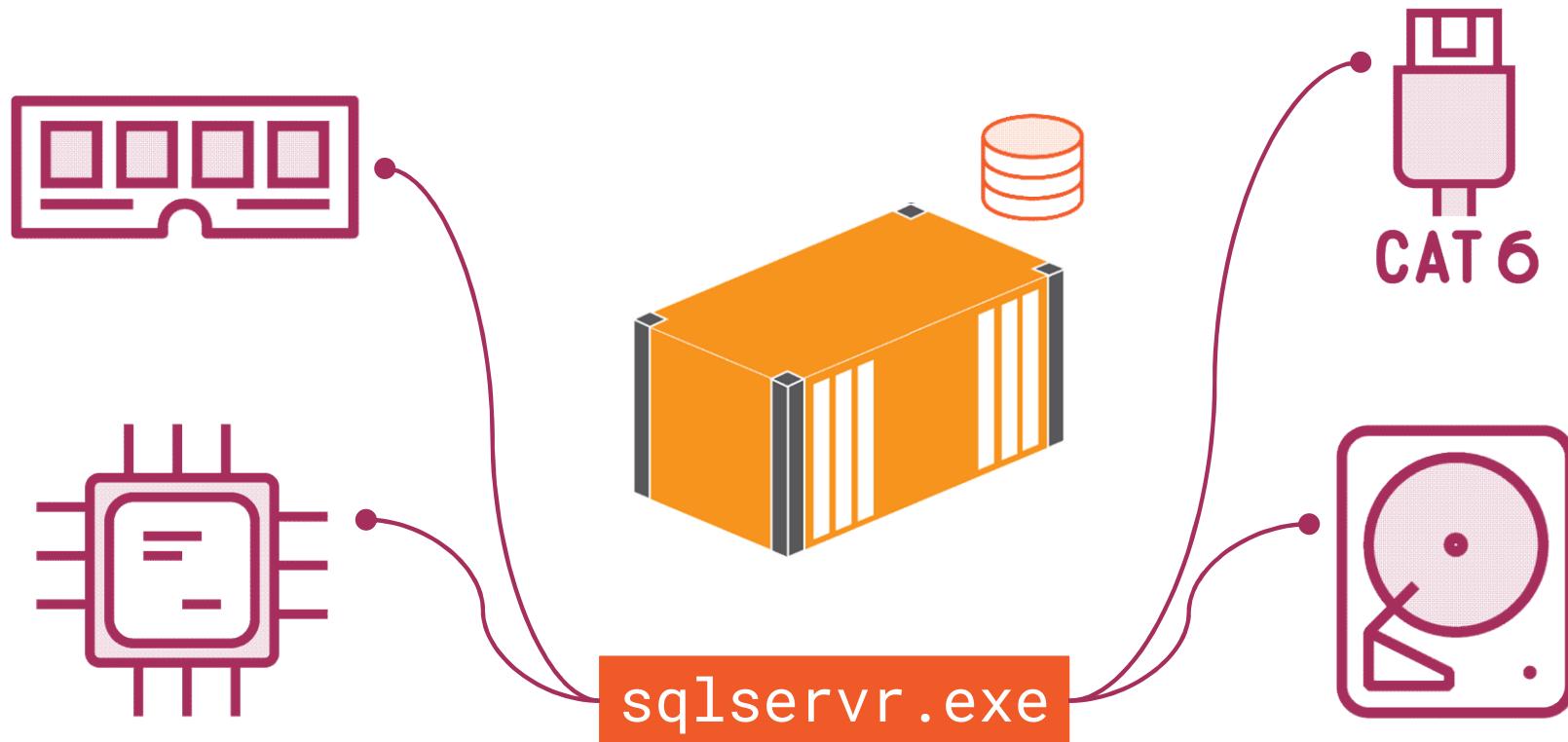
Demo

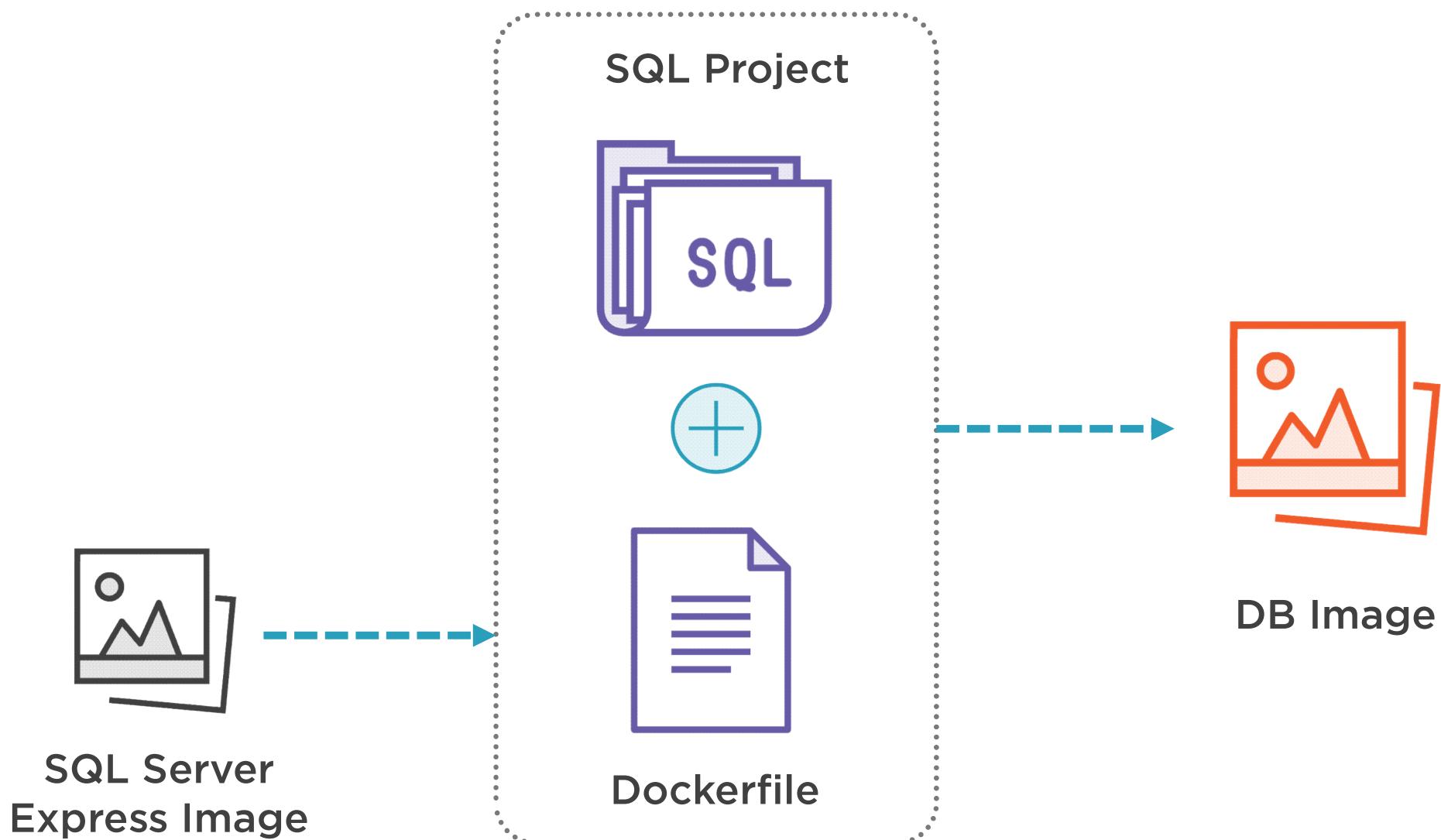


Database schema updates

- With Docker volumes

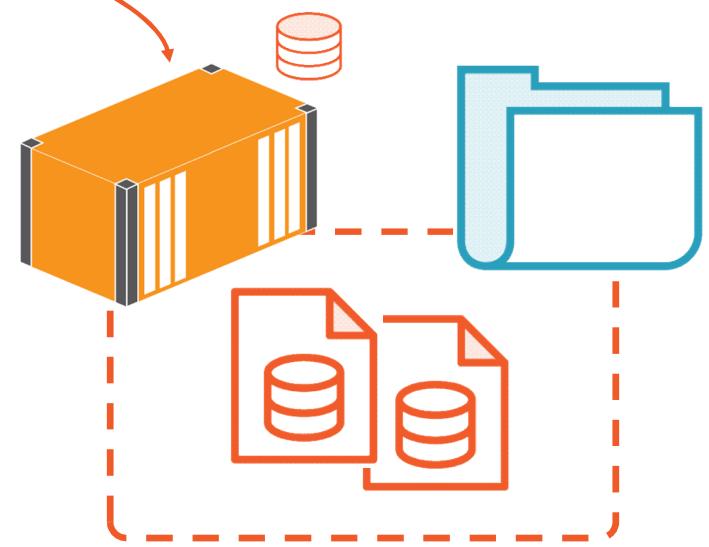




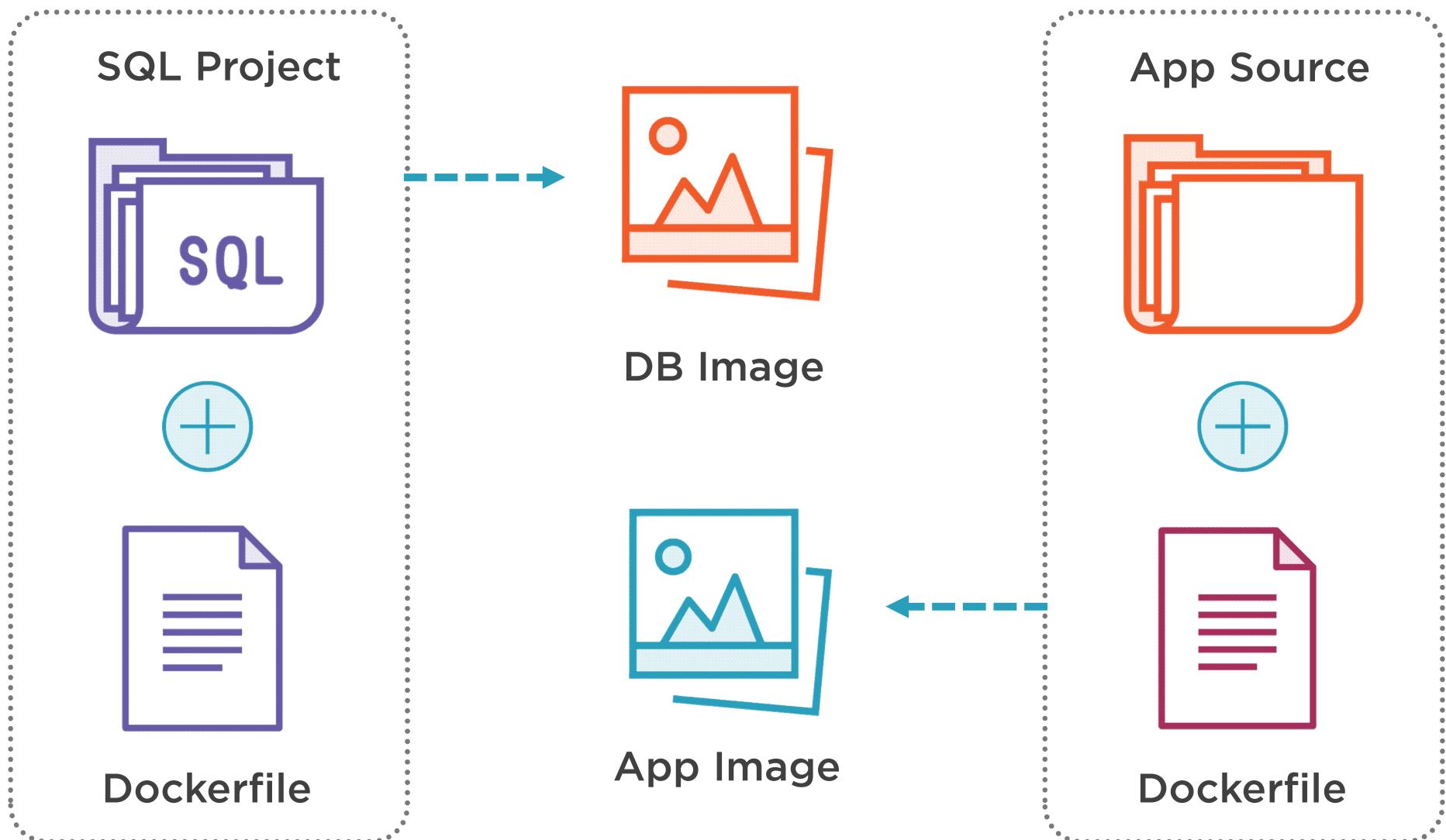


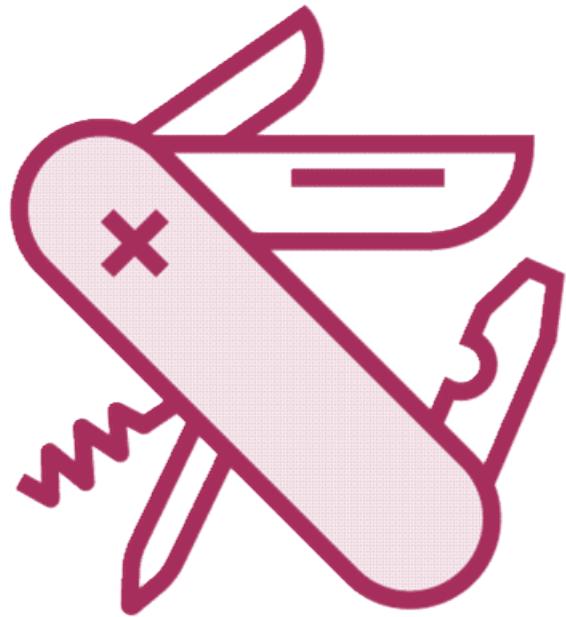


Disposable DB container



Persistent DB container





Docker is part of your toolkit

- Databases come too
- Containers mix well
- Connecting to external services

Coming Next



Scaling performance with messaging

- Connecting app & db containers
- Extracting features to new containers
- Publish events to a message queue