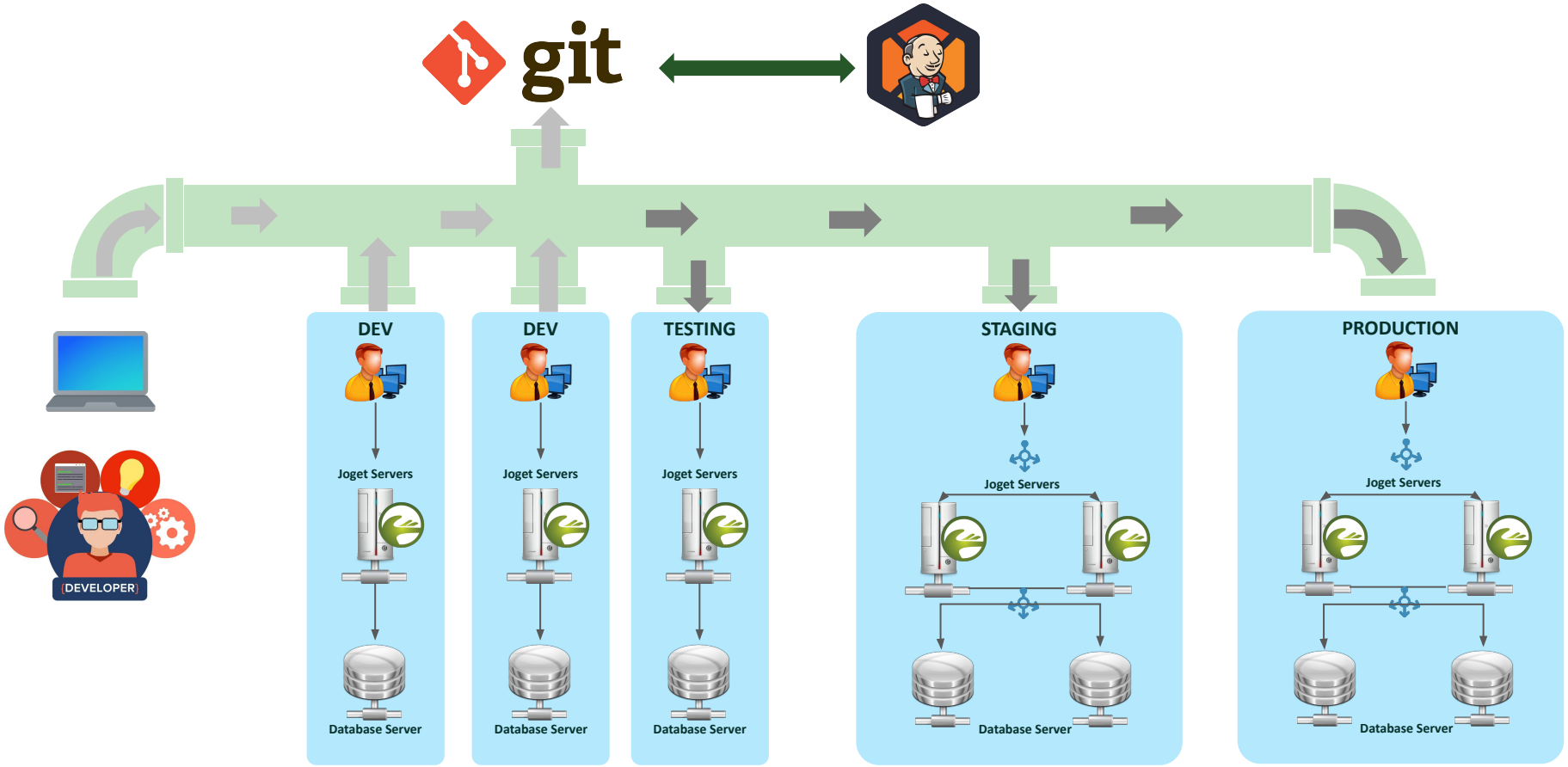
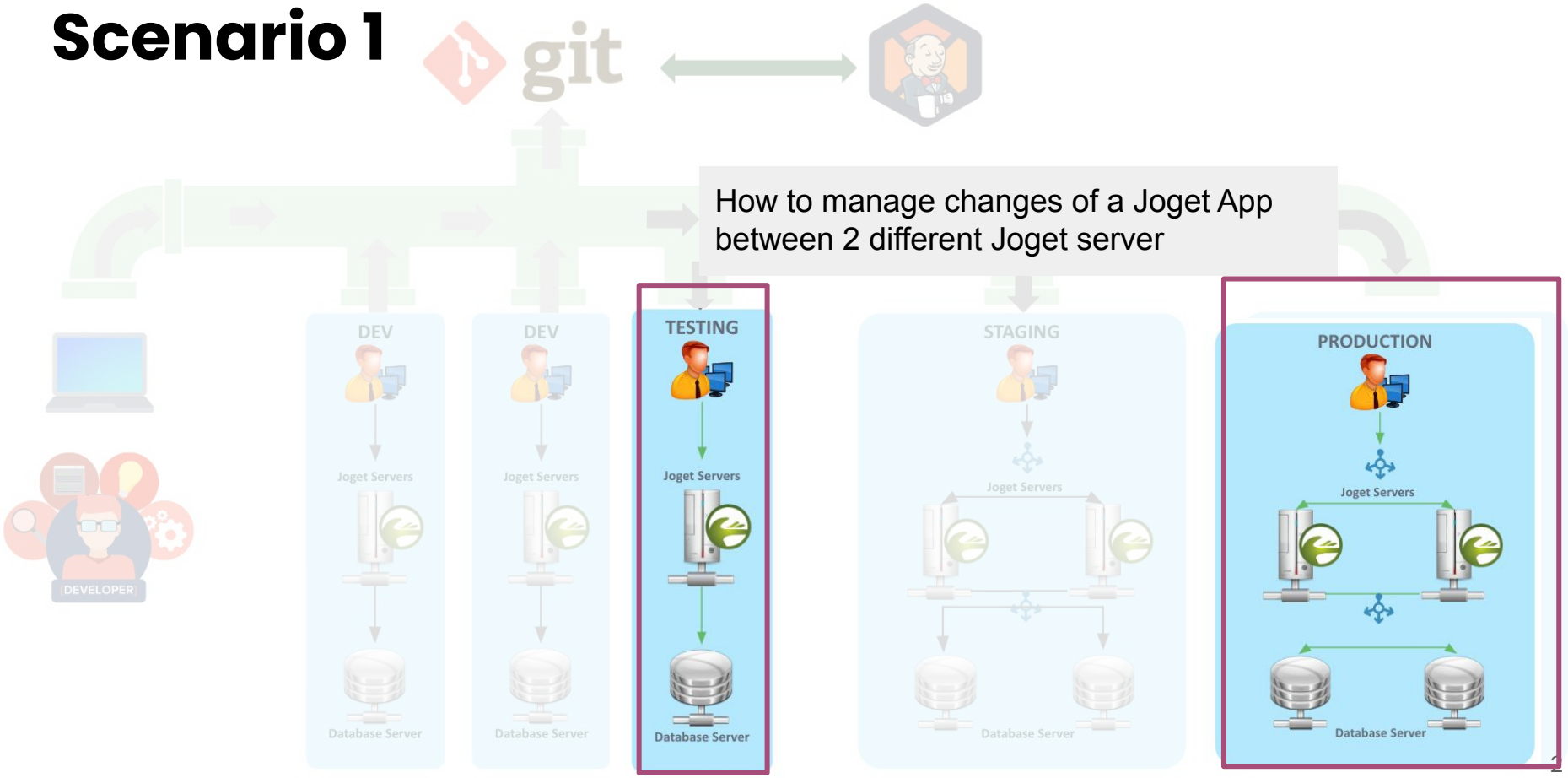
Deployment / Environment Viewpoint



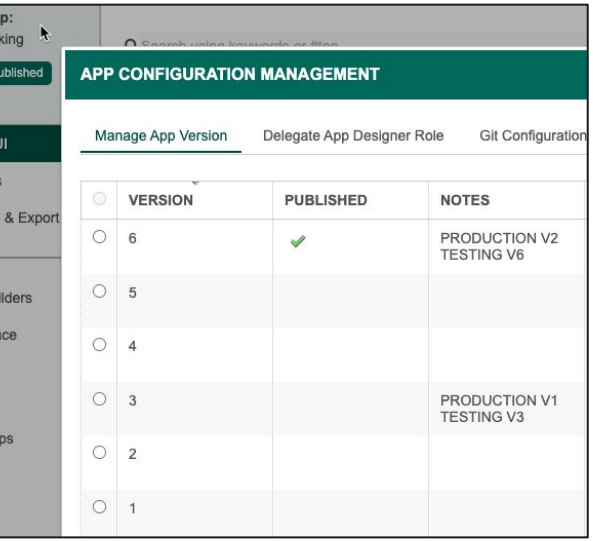
Scenario 1

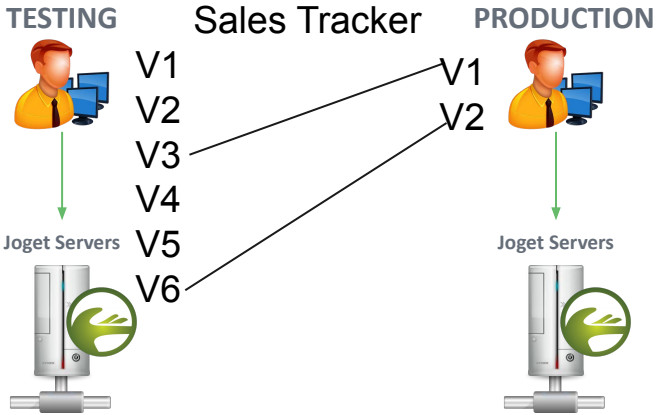
How to manage changes of a Joget App between 2 different Joget server

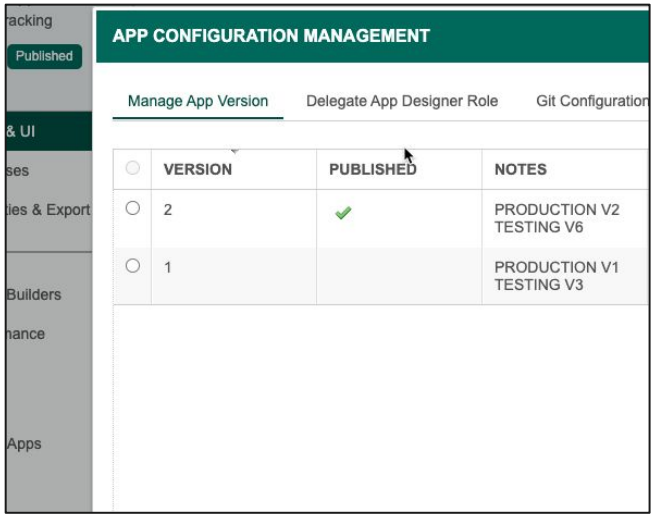


Scenario 1

- In Testing, we exported the app at V3 and imported into Production as V1.
- Likewise, V6 is exported from Testing into Production as V3.

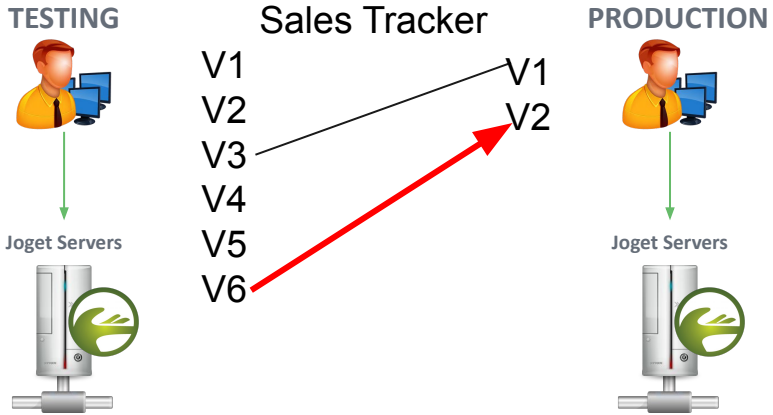






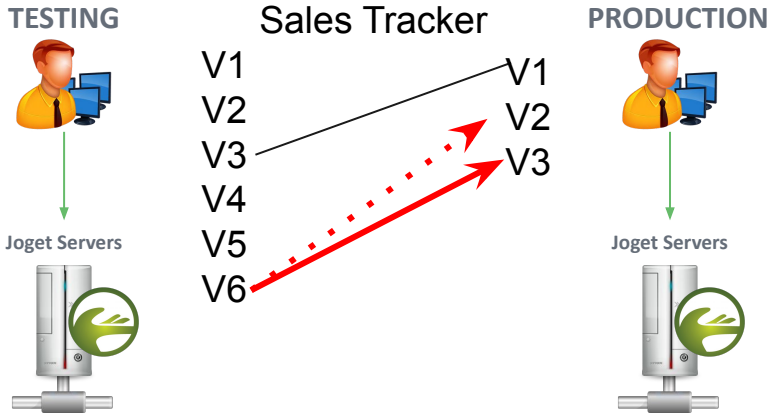
Scenario 1 Problem Statement

- Changes are made in **Testing V6**. How should we bring the changes to **Product V2**?



Solution – Option 1

- **Copy** from **Testing V6** and **merge** into **Production V3**.



Solution - Option 1

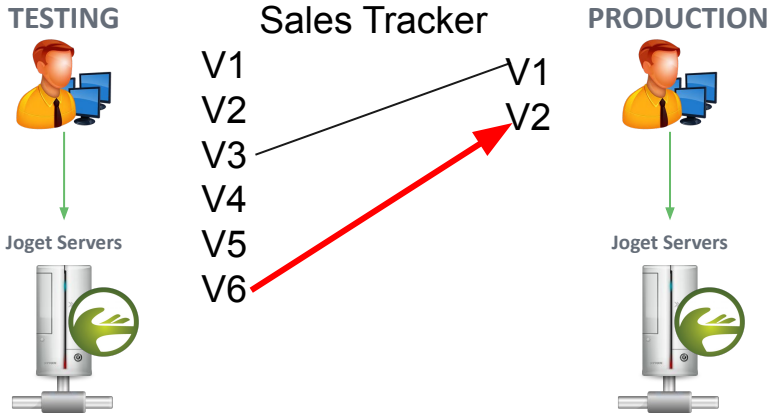
- Export Testing V6 and Import as Production V3
- If there is any running processes in Production V2, decision will need to made whether to MIGRATE to Production V3.

  To migrate, execute

```
AppService.updateRunningProcesses(packageId, oldProcessVersion,
newProcessVersion);
```

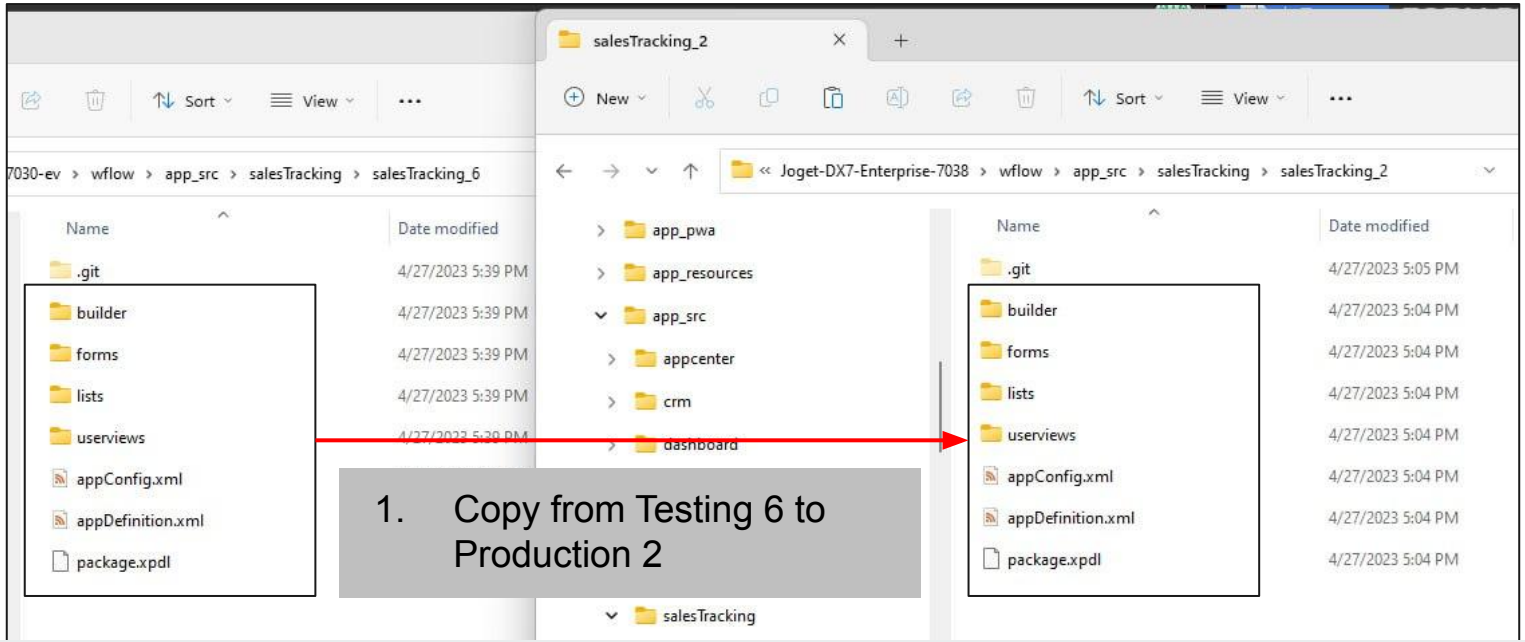
Solution – Option 2

- **Copy** from **Testing V6** and **merge** into **Production V2**.



Solution – Option 2

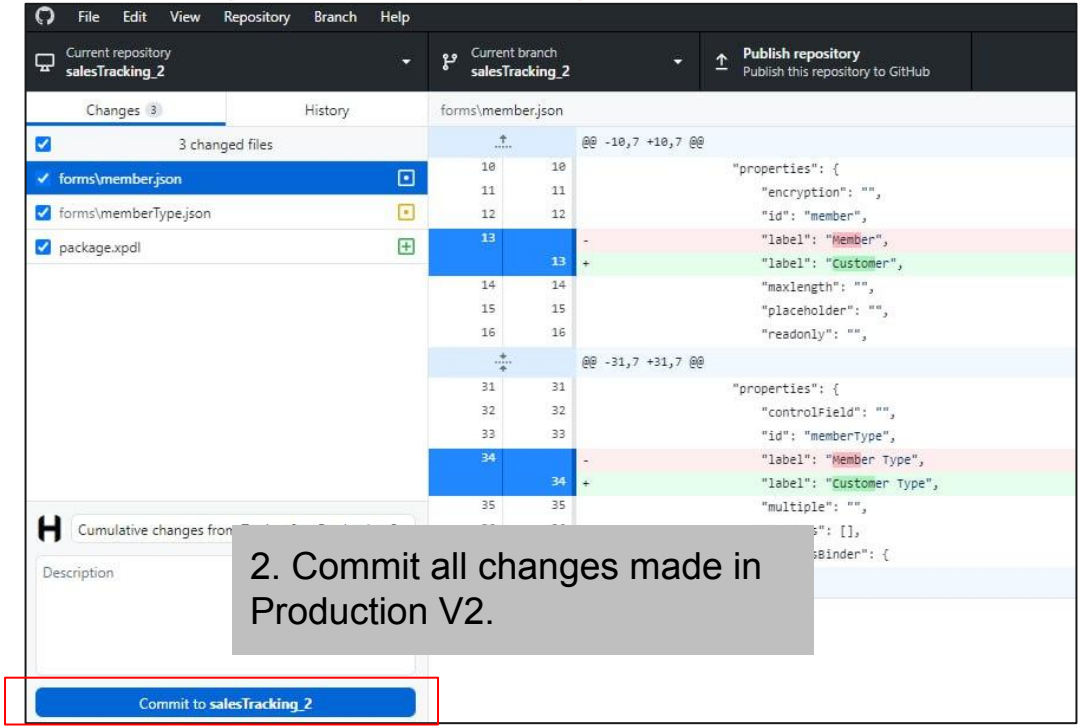
- **Copy** from **Testing V6** and **merge** into **Production V2**.



Note: appConfig.xml contains environment specific settings (dev, test, prod) such as counters, smtp settings that you may want to preserve.

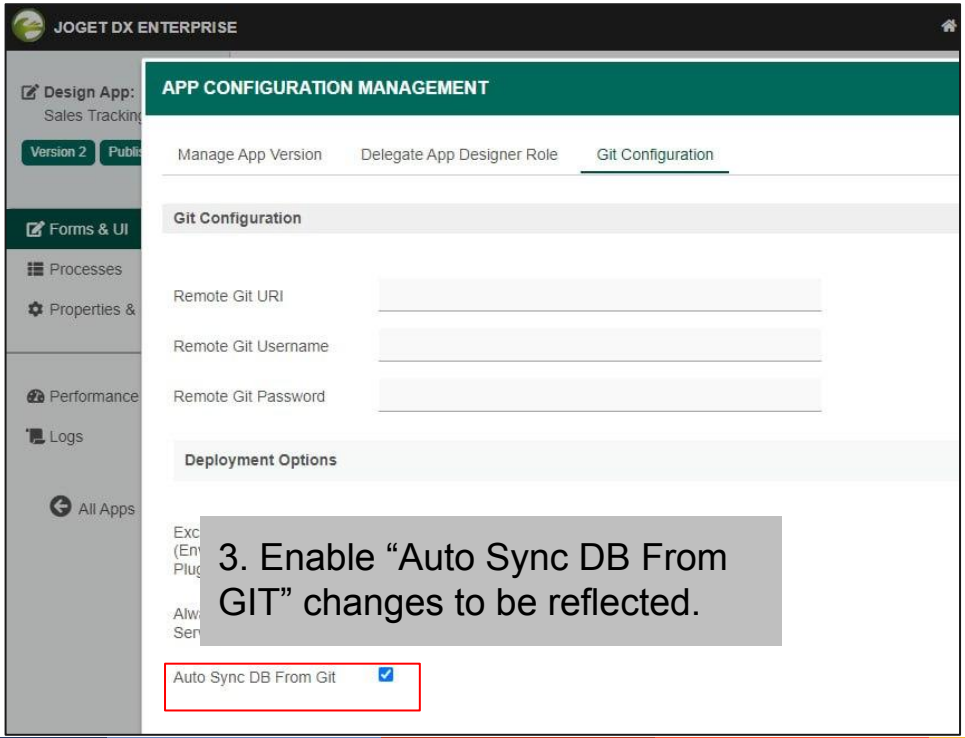
Solution – Option 2

- **Copy** from **Testing V6** and **merge** into **Production V2**.



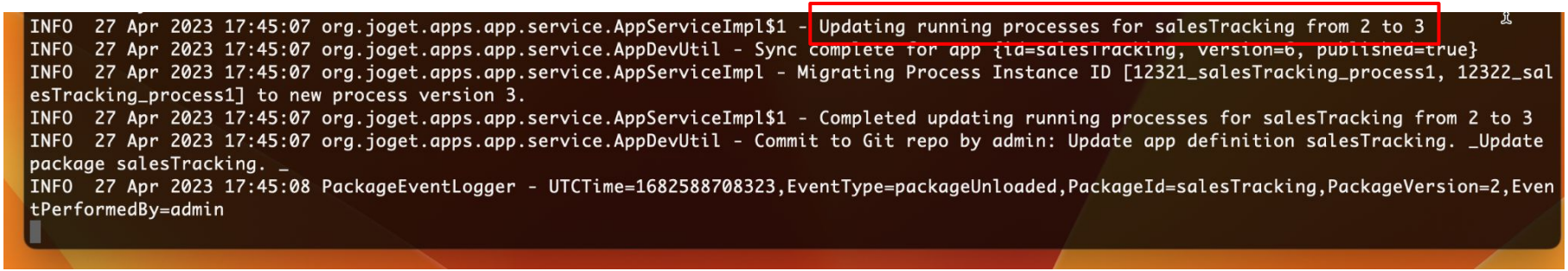
Solution – Option 2

- **Copy** from **Testing V6** and **merge** into **Production V2**.



Solution – Option 2

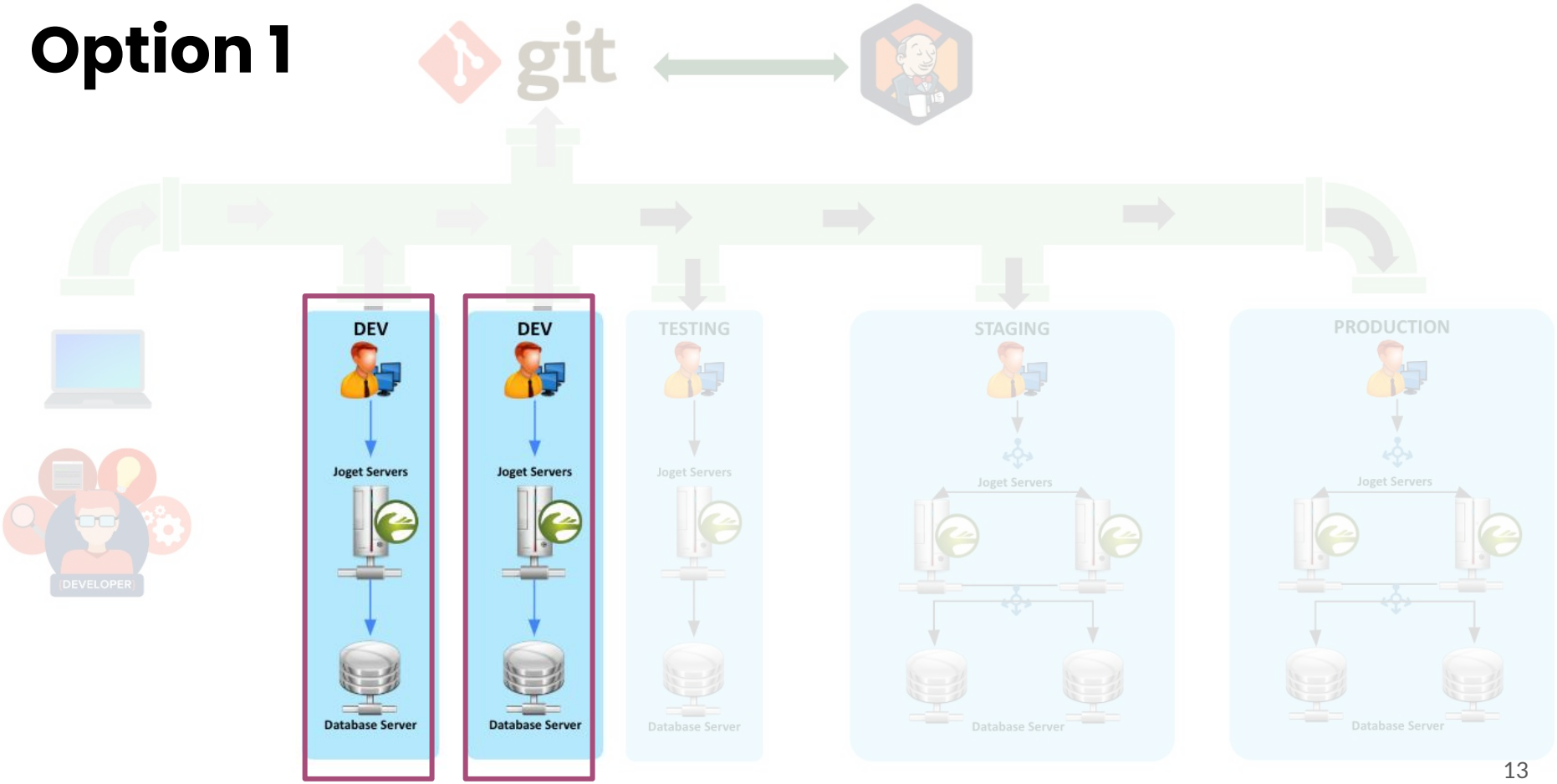
- **Copy** from **Testing V6** and **merge** into **Production V2**.
- If there is any process design change, any running processes in **Production V2** will be **automatically** migrated to the latest process design.



Scenario 2

- How to collaboratively design a Joget app?

Option 1

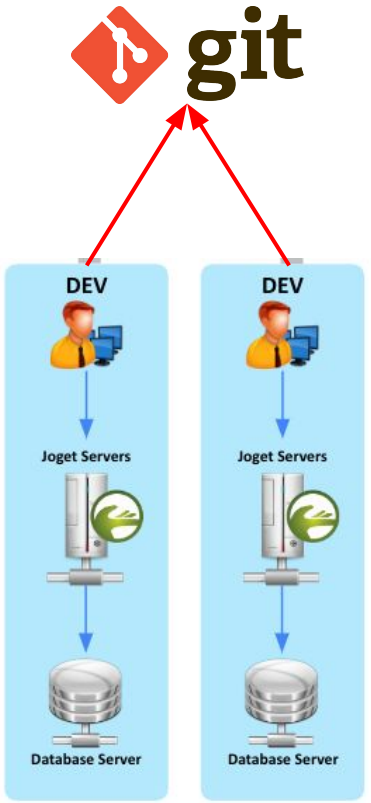

git
DEV
Joget Servers
Database Server
DEV
Joget Servers
Database Server
TESTING
Joget Servers
Database Server
STAGING
Joget Servers
Database Server
PRODUCTION
Joget Servers
Database Server
DEVELOPER

Option 1

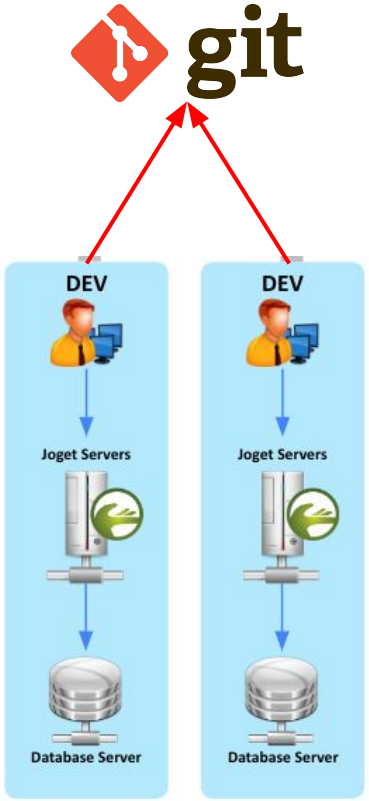
- Designer A runs Joget locally.
- Designer B runs Joget locally.
- Both work on the same app.

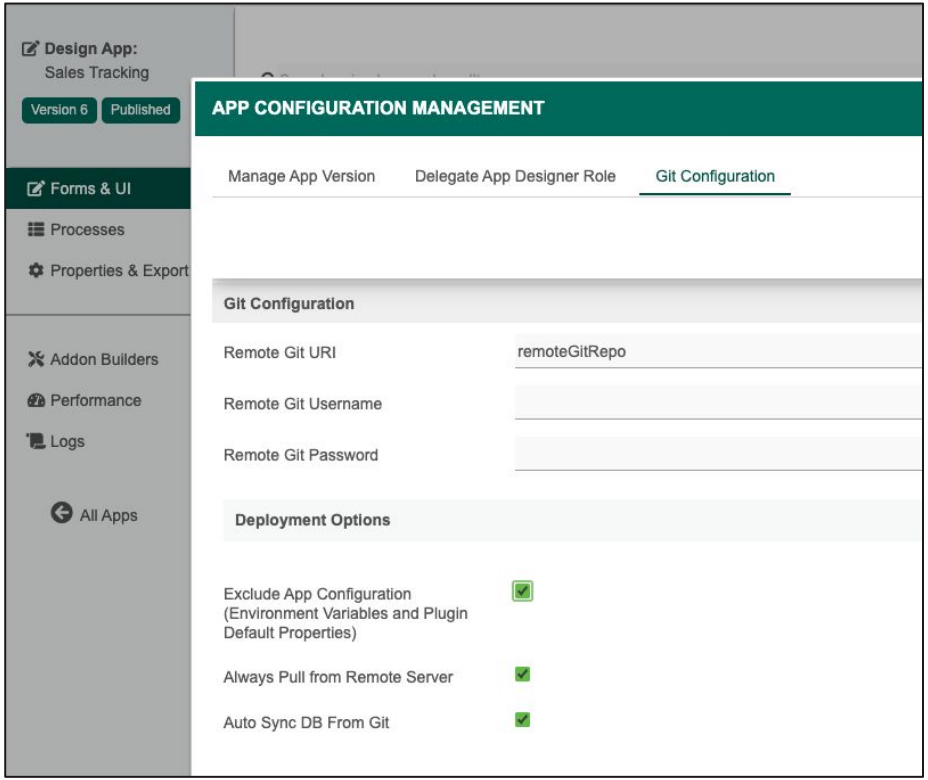
Option 1

- All designers point their app to a common git repository.
- Any changes made to the app will be **synced** to the remote git.



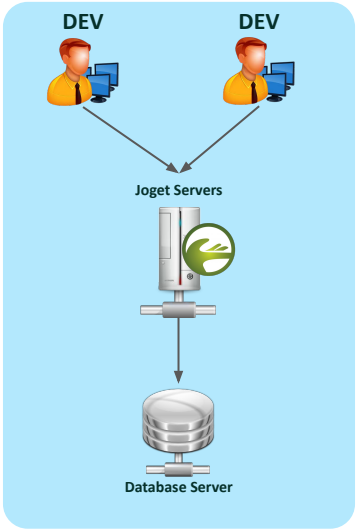
Option 1


git
DEV
DEV
Joget Servers
Joget Servers
Database Server
Database Server


Design App:
Sales Tracking
Version 6
Published
Forms & UI
Processes
Properties & Export
Addon Builders
Performance
Logs
All Apps
APP CONFIGURATION MANAGEMENT
Manage App Version
Delegate App Designer Role
Git Configuration
Git Configuration
Remote Git URI
remoteGitRepo
Remote Git Username
Remote Git Password
Deployment Options
Exclude App Configuration (Environment Variables and Plugin Default Properties)
Always Pull from Remote Server
Auto Sync DB From Git

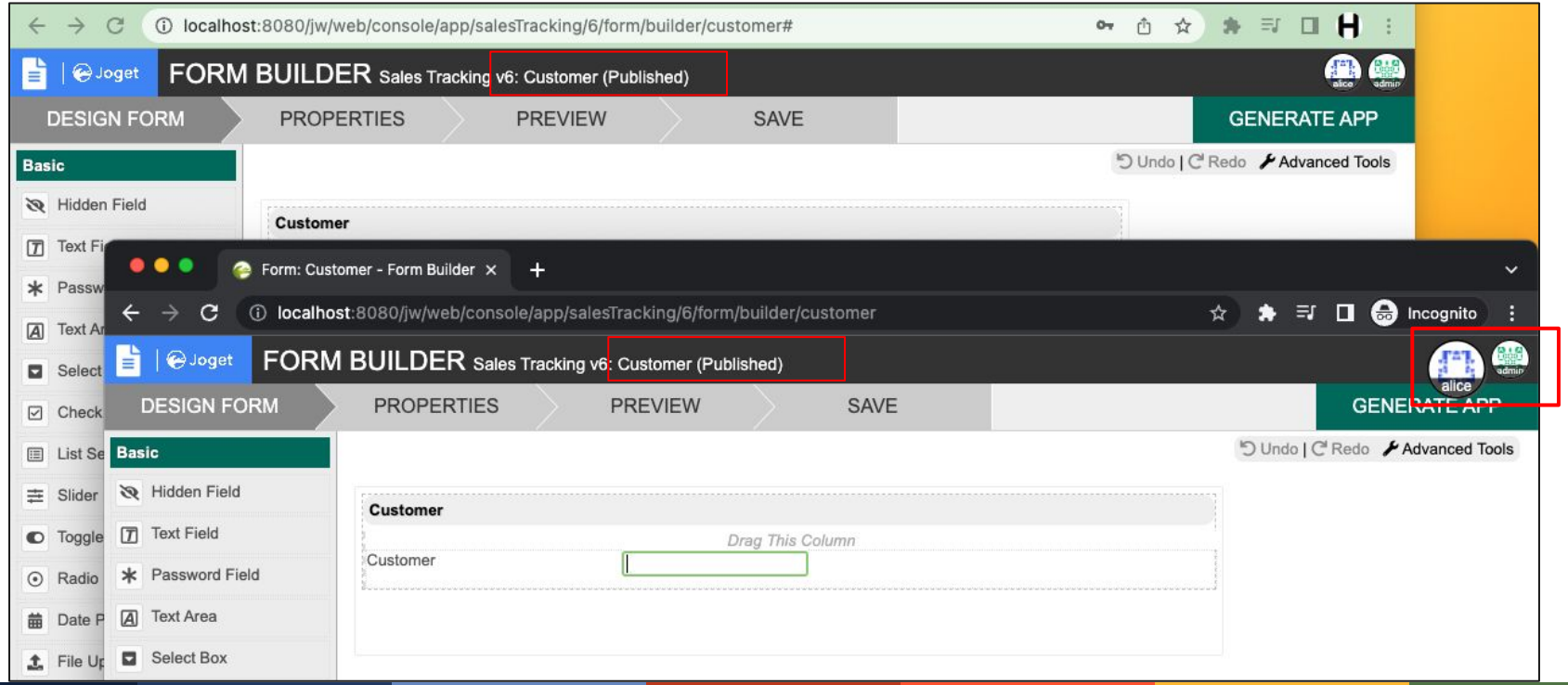
Option 2

- All designers work on the same Joget server.



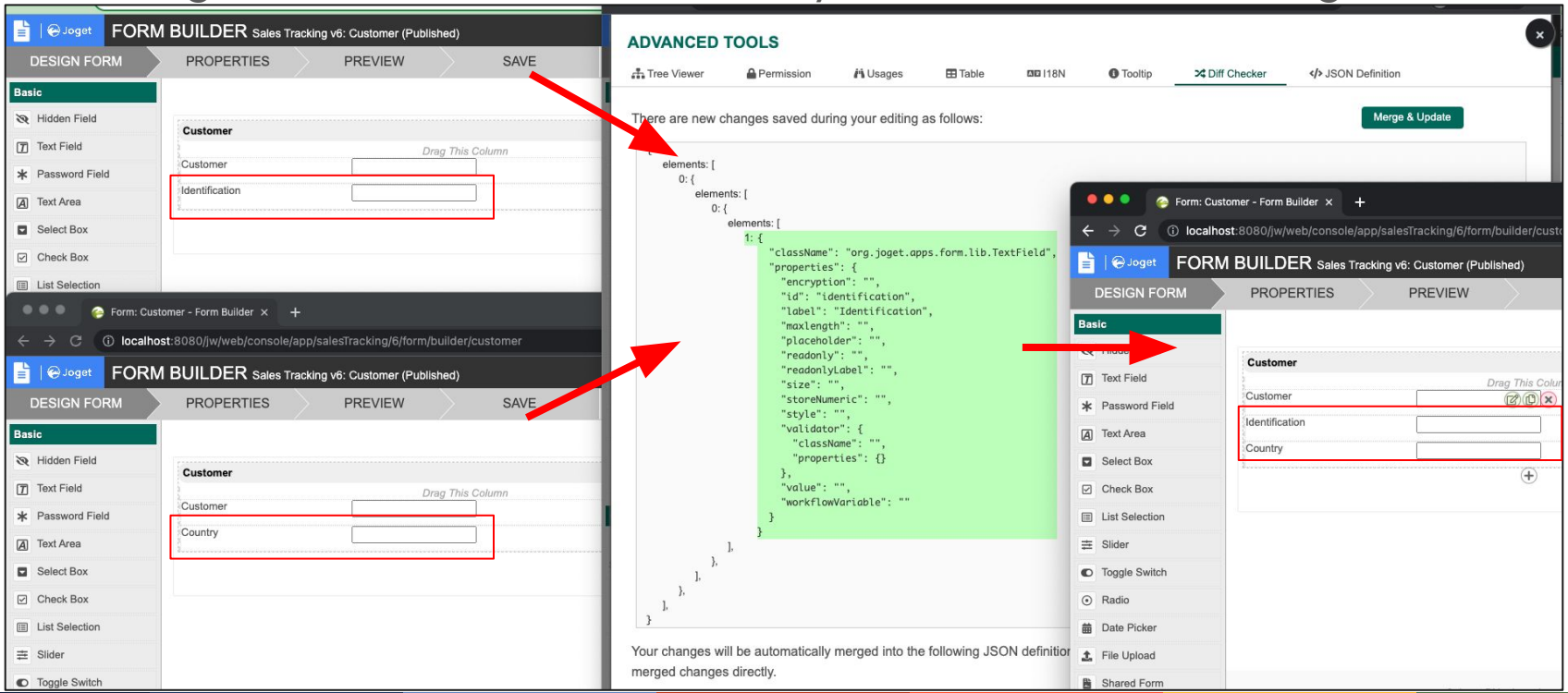
Option 2

- Designers work on the same entity would be indicated on presence



Option 2

- Changes made on the same entity (i.e. form) will be merged.



Summary

- We discussed various scenarios on how an Joget app can be developed collaboratively, maintained, and promoted to different environments.
- Some of these steps can be automated / adopted into your existing CI/CD tools.
- We hope that this provides useful insights for you to establish a CI/CD plan that would work best based on your needs.