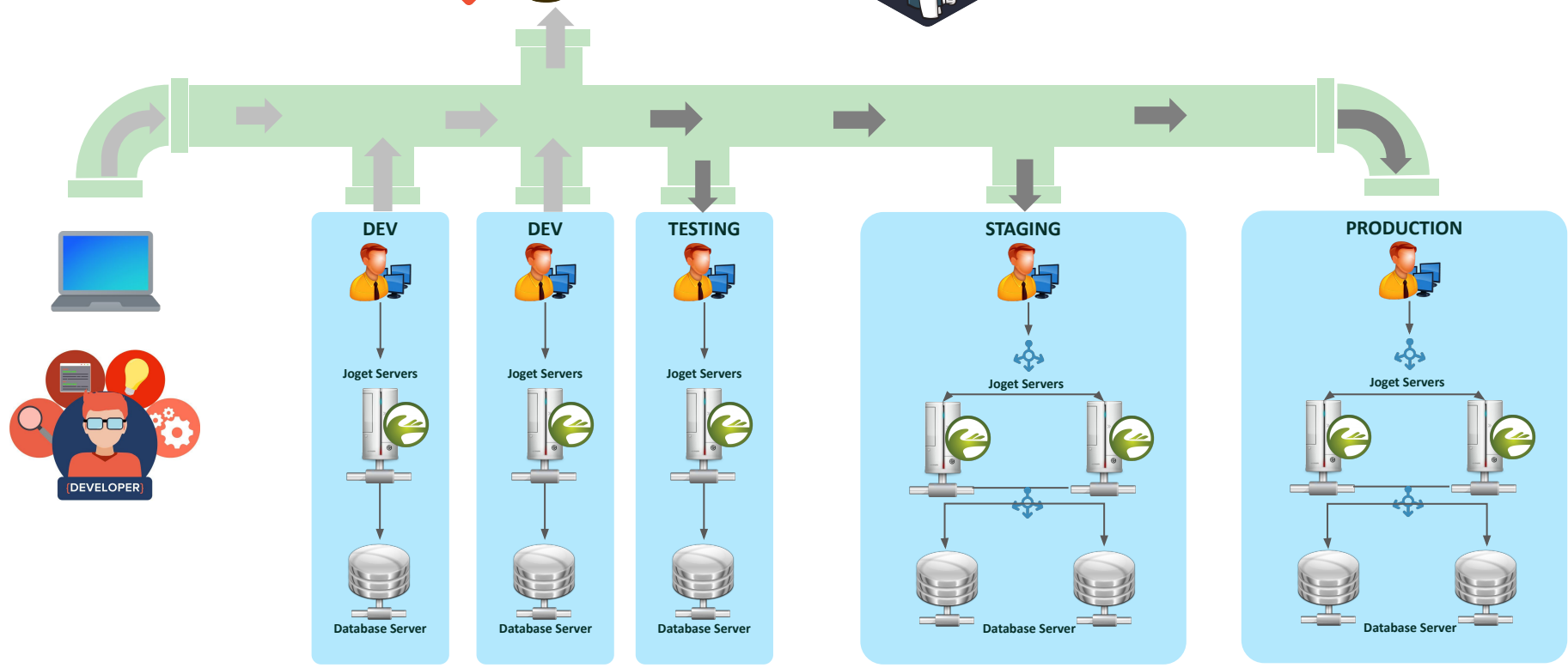
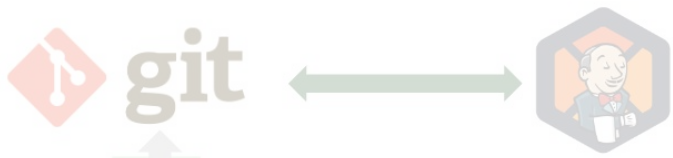


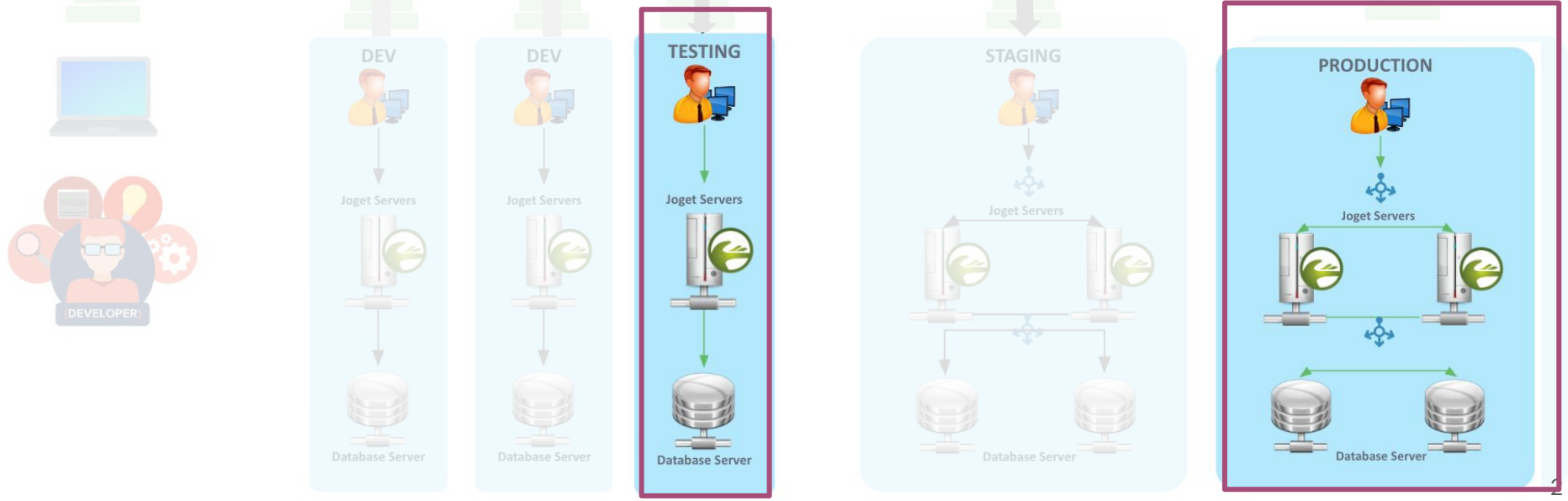
# Deployment / Environment Viewpoint



# Scenario 1

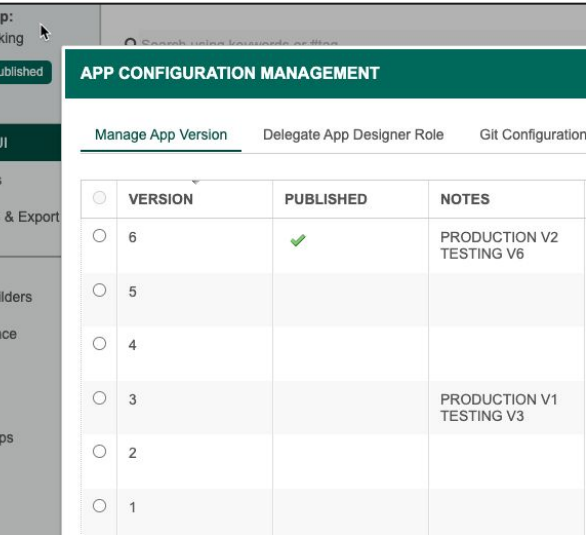


How to manage changes of a Joget App between 2 different Joget server

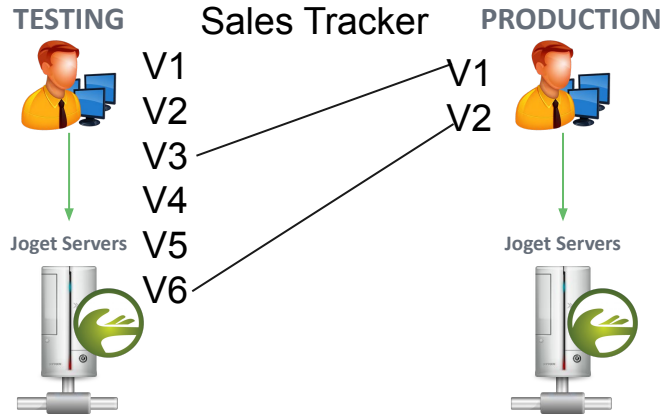
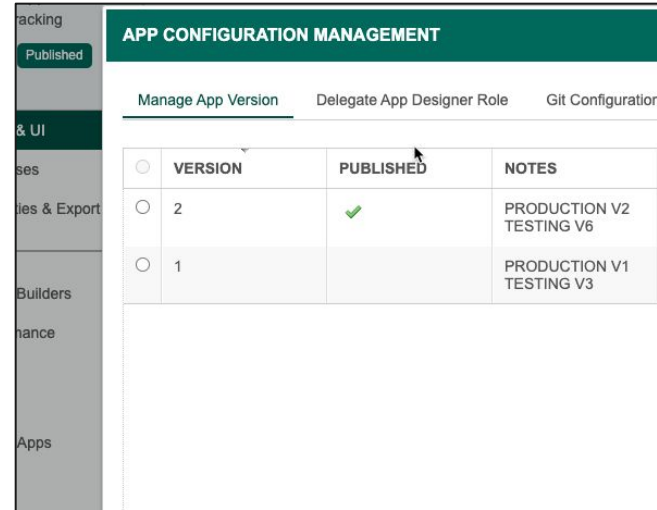


# Scenario 1

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



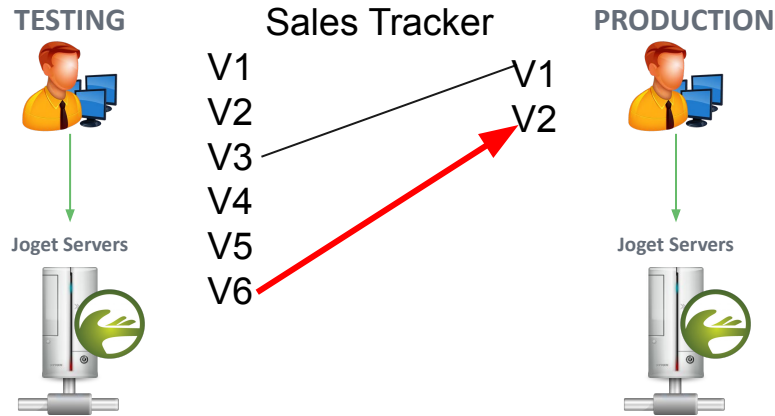
APP CONFIGURATION MANAGEMENT		
VERSION	PUBLISHED	NOTES
<input type="radio"/> 6	✓	PRODUCTION V2 TESTING V6
<input type="radio"/> 5		
<input type="radio"/> 4		
<input type="radio"/> 3		PRODUCTION V1 TESTING V3
<input type="radio"/> 2		
<input type="radio"/> 1		

APP CONFIGURATION MANAGEMENT		
VERSION	PUBLISHED	NOTES
<input type="radio"/> 2	✓	PRODUCTION V2 TESTING V6
<input type="radio"/> 1		PRODUCTION V1 TESTING 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

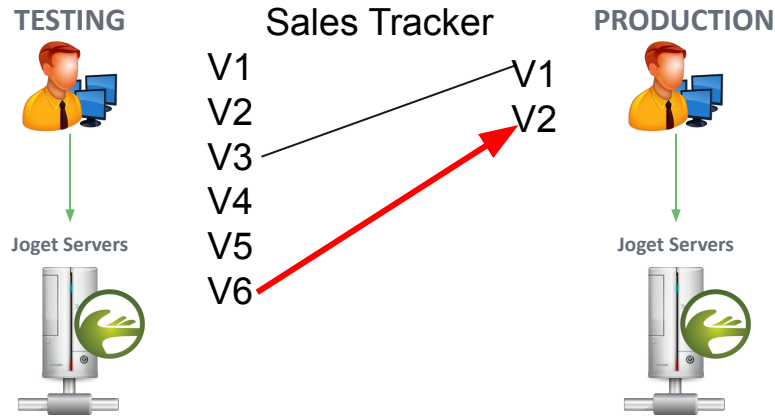
- Export Testing V6 and Import as Production V3
- If there is any running processes in Production V2, decision will need to be made whether to MIGRATE to Production V3.

To migrate, execute

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

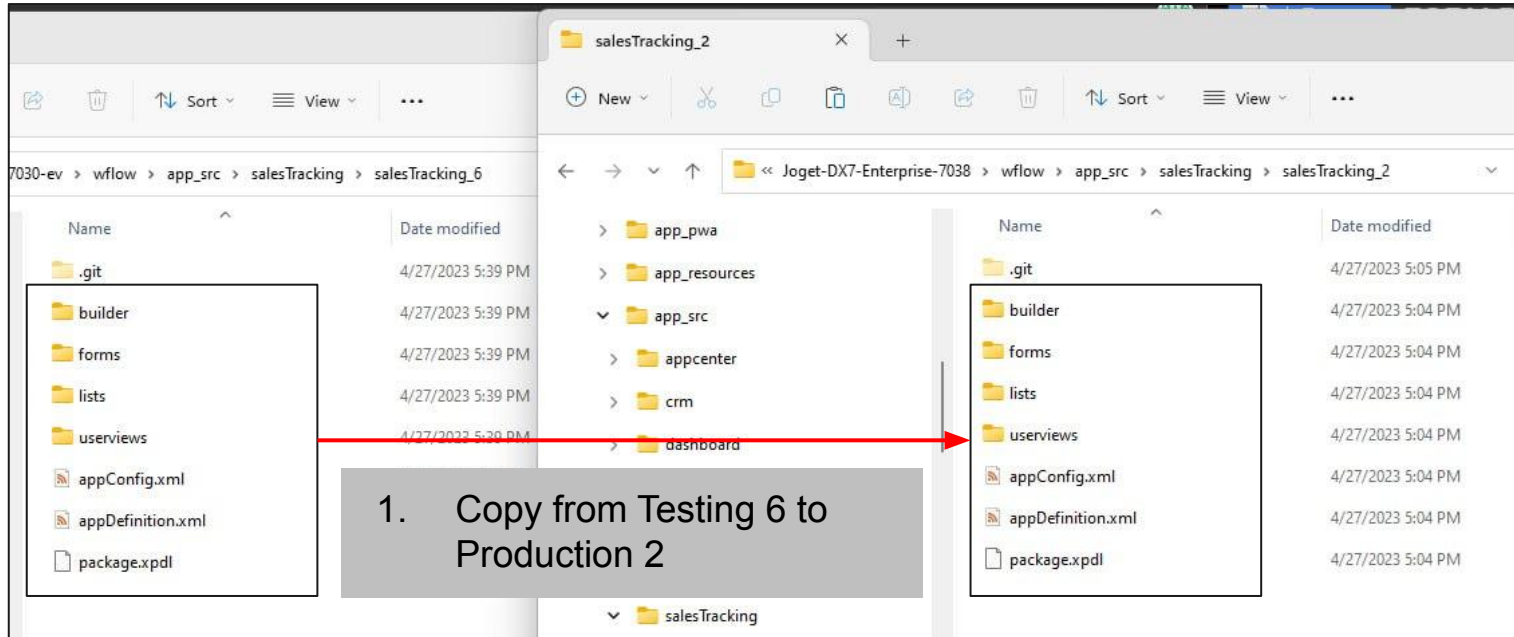
# Solution – Option 2

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



# Solution – Option 2

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

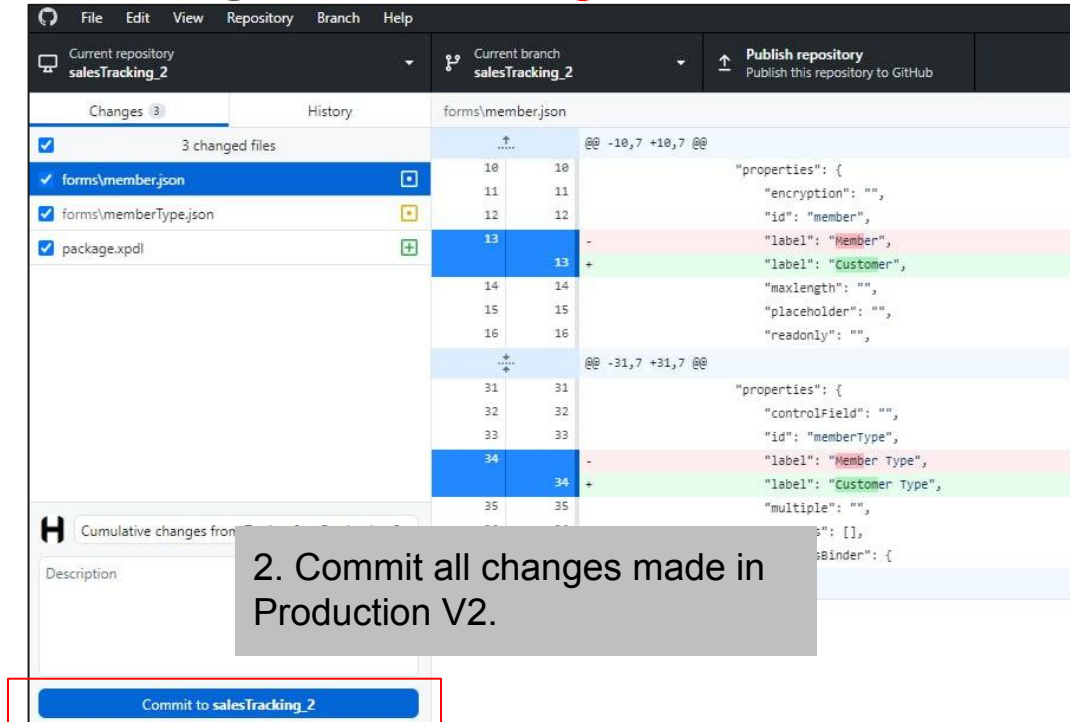


Note: appConfig.xml contains environment specific settings (dev, test, prod) such as counters, smtp settings that you may want to preserve.



# Solution – Option 2

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



File Edit View Repository Branch Help

Current repository salesTracking\_2 Current branch salesTracking\_2 Publish repository Publish this repository to GitHub

Changes (3) History forms/member.json

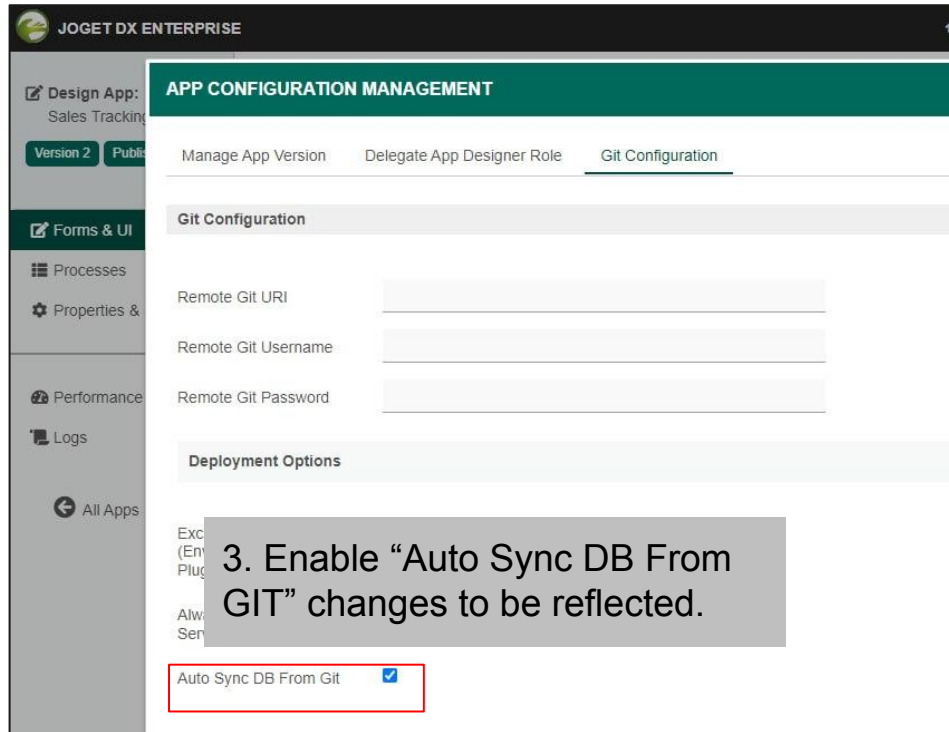
File	Line	Change	Code
forms/member.json	10	10	"properties": {
forms/member.json	11	11	"encryption": "",
forms/member.json	12	12	"id": "member",
forms/member.json	13	-	"label": "Member",
forms/member.json	13	+	"label": "Customer",
forms/member.json	14	14	"maxlength": "",
forms/member.json	15	15	"placeholder": "",
forms/member.json	16	16	"readonly": "",
forms/member.json	31	31	"properties": {
forms/member.json	32	32	"controlField": "",
forms/member.json	33	33	"id": "memberType",
forms/member.json	34	-	"label": "Member Type",
forms/member.json	34	+	"label": "Customer Type",
forms/member.json	35	35	"multiple": "",

2. Commit all changes made in Production V2.

Commit to salesTracking\_2

# Solution – Option 2

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



The screenshot shows the 'APP CONFIGURATION MANAGEMENT' interface in Joget DX Enterprise. The left sidebar contains navigation options: Design App (Sales Tracking), Forms & UI, Processes, Properties &, Performance, Logs, and All Apps. The main content area has three tabs: 'Manage App Version', 'Delegate App Designer Role', and 'Git Configuration'. Under 'Git Configuration', there are three input fields: 'Remote Git URI', 'Remote Git Username', and 'Remote Git Password'. Below these is a 'Deployment Options' section with a checkbox for 'Auto Sync DB From Git' which is checked. A grey callout box with the text '3. Enable “Auto Sync DB From GIT” changes to be reflected.' is overlaid on the checkbox. A red box highlights the checkbox and its label.

# 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.

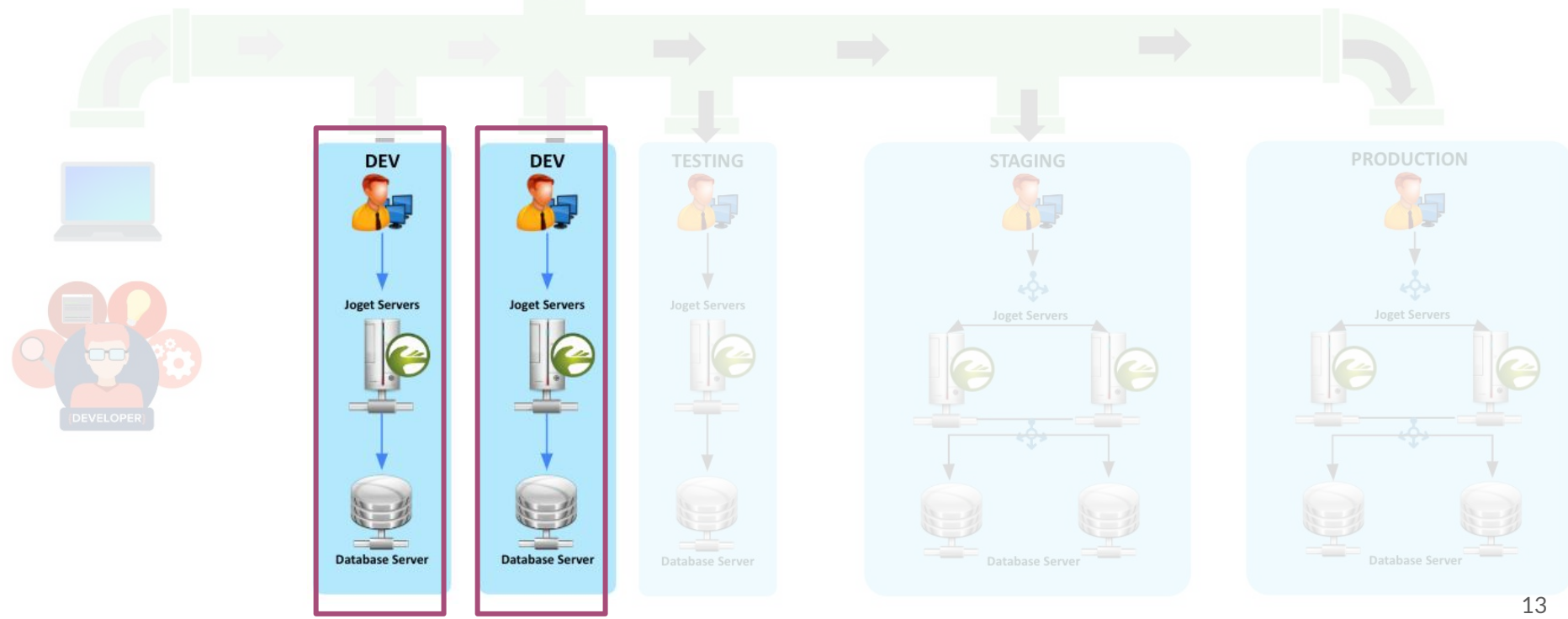
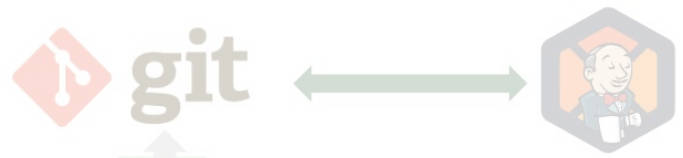
```
INFO 27 Apr 2023 17:45:07 org.joget.apps.app.service.AppServiceImpl$1 - Updating running processes for salesTracking from 2 to 3
INFO 27 Apr 2023 17:45:07 org.joget.apps.app.service.AppDevUtil - Sync complete for app {id=salesTracking, version=6, published=true}
INFO 27 Apr 2023 17:45:07 org.joget.apps.app.service.AppServiceImpl - Migrating Process Instance ID [12321_salesTracking_process1, 12322_salesTracking_process1] to new process version 3.
INFO 27 Apr 2023 17:45:07 org.joget.apps.app.service.AppServiceImpl$1 - Completed updating running processes for salesTracking from 2 to 3
INFO 27 Apr 2023 17:45:07 org.joget.apps.app.service.AppDevUtil - Commit to Git repo by admin: Update app definition salesTracking. _Update package salesTracking. _
INFO 27 Apr 2023 17:45:08 PackageEventLogger - UTCTime=1682588708323,EventType=packageUnloaded,PackageId=salesTracking,PackageVersion=2,EventPerformedBy=admin
```

# Scenario 2

- How to collaboratively design a Joget app?



# Option 1

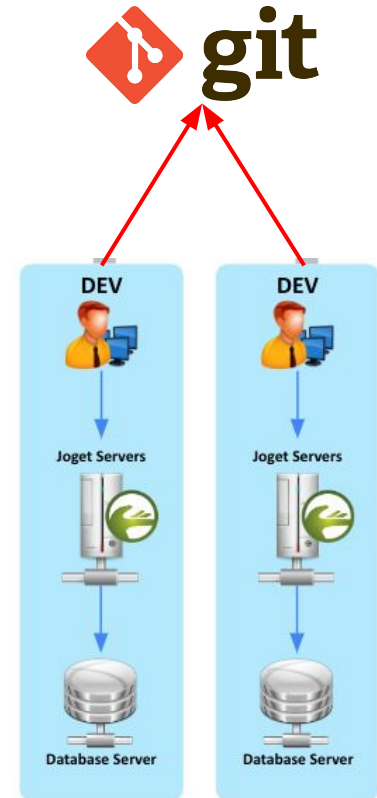


# Option 1

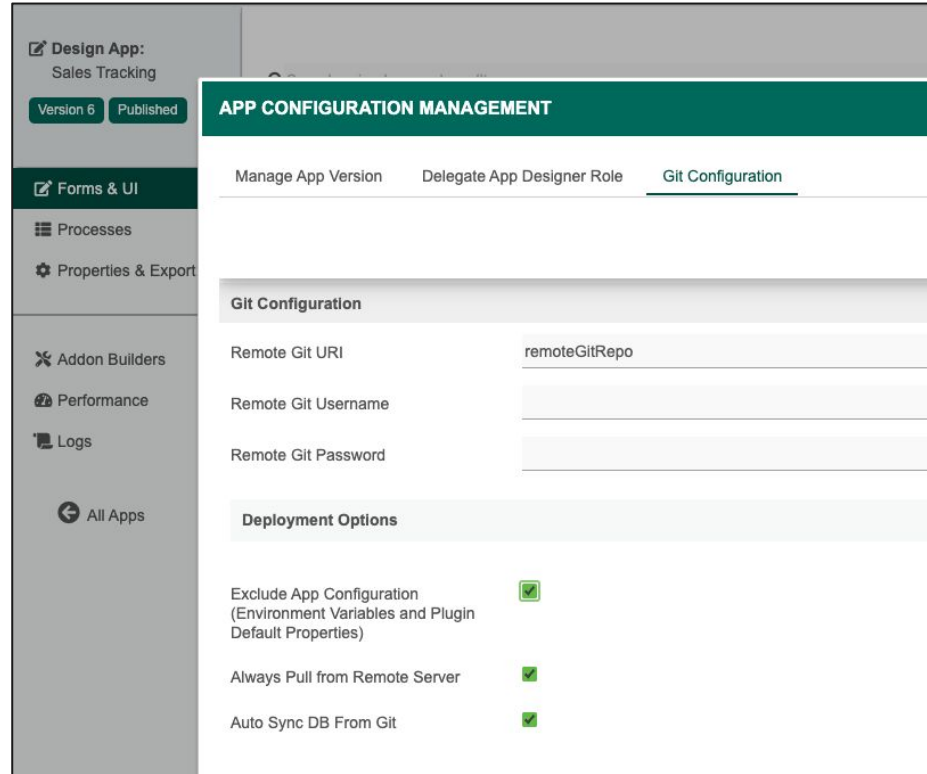
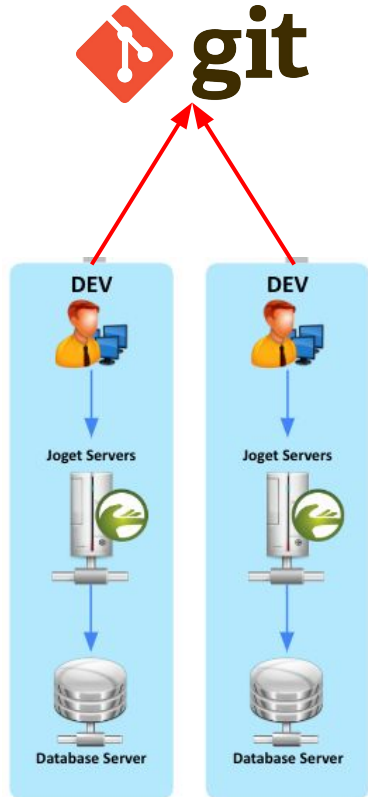
- Designer A runs Joget locally.
- Designer B runs Joget locally.
- Both work on the same app.

# 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



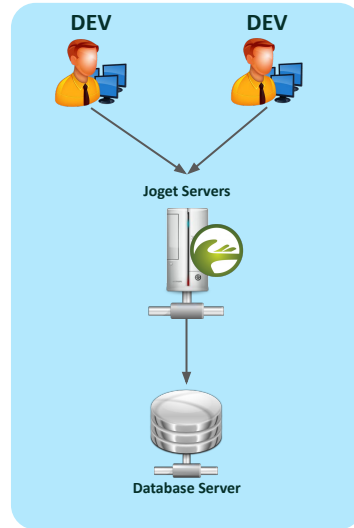
The screenshot shows the "APP CONFIGURATION MANAGEMENT" interface for a "Design App: Sales Tracking" (Version 6, Published). The "Git Configuration" tab is active, showing the following settings:

Configuration Item	Value
Remote Git URI	remoteGitRepo
Remote Git Username	
Remote Git Password	
Exclude App Configuration (Environment Variables and Plugin Default Properties)	<input checked="" type="checkbox"/>
Always Pull from Remote Server	<input checked="" type="checkbox"/>
Auto Sync DB From Git	<input checked="" type="checkbox"/>



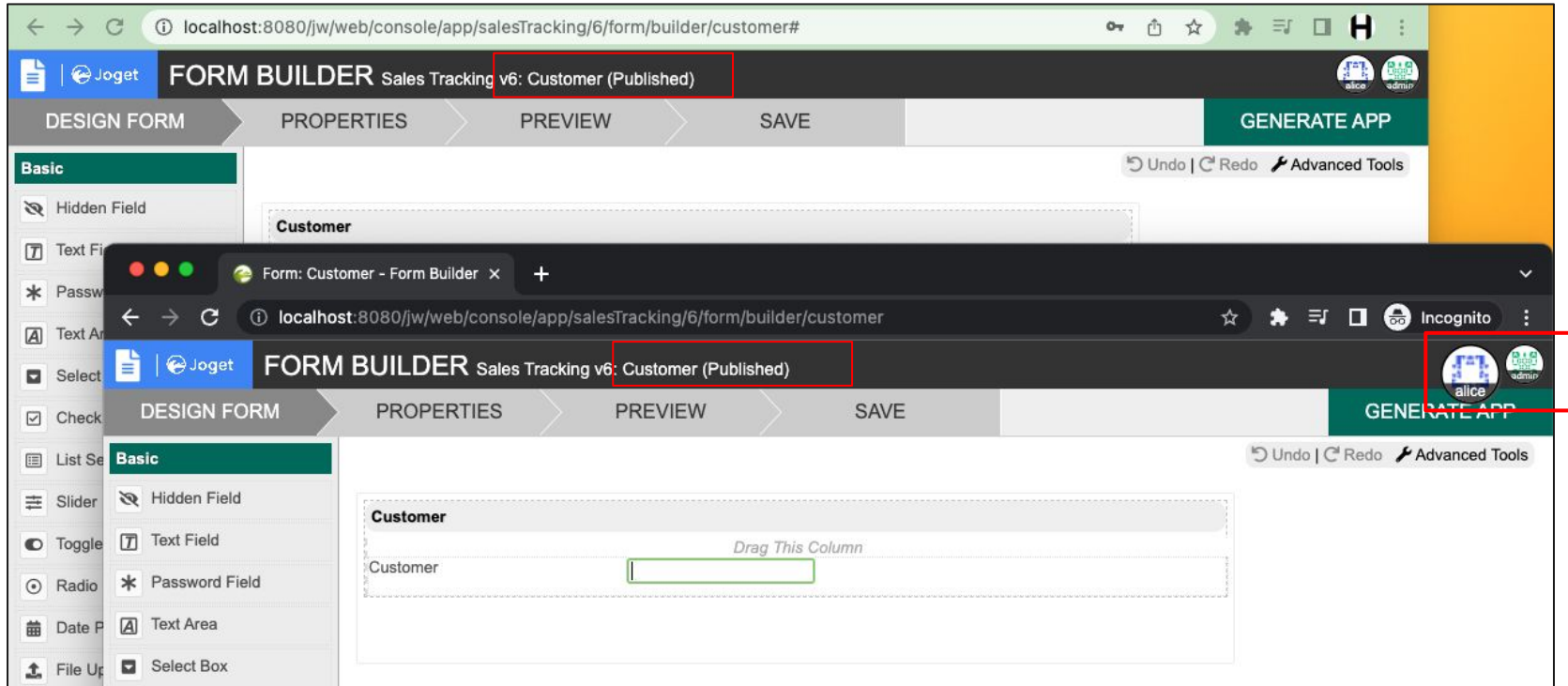
# Option 2

- All designers work on the same Joget server.



# Option 2

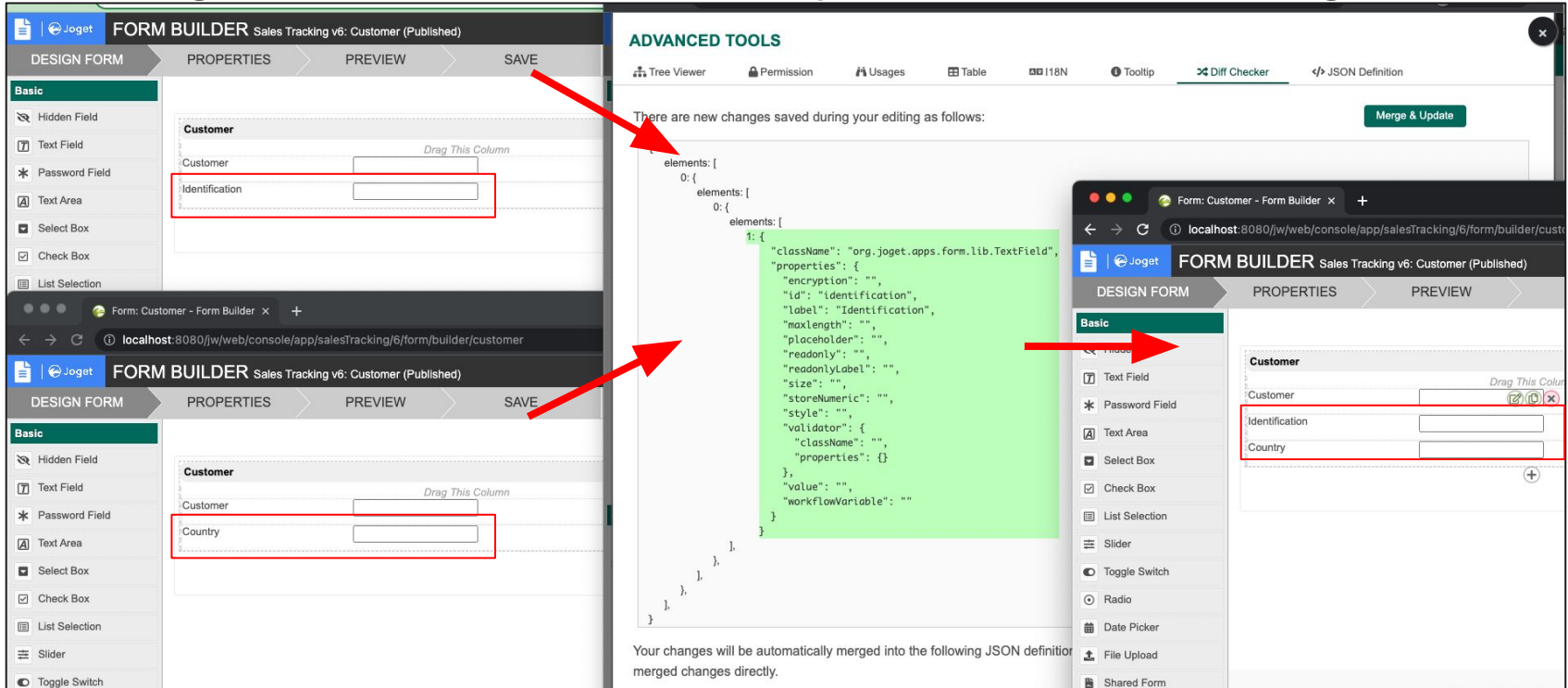
- Designers work on the same entity would be indicated on presence



The screenshot displays the Joget Form Builder interface in a web browser. The top window shows the 'FORM BUILDER Sales Tracking v6: Customer (Published)' interface with the 'DESIGN FORM' tab selected. The bottom window, titled 'Form: Customer - Form Builder', shows the same interface but with a 'Text Field' widget being dragged into the form. The user profile 'alice' is visible in the top right of both windows.

# Option 2

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



The image illustrates the process of merging changes in the Joget Form Builder. It shows three overlapping windows:

- Top Left Window:** The 'FORM BUILDER' interface for 'Sales Tracking v6: Customer (Published)'. The 'DESIGN FORM' tab is active. The 'Basic' widget palette is on the left. The form preview shows an 'Identification' field highlighted with a red box. A red arrow points from this box to the JSON definition.
- Top Right Window:** The 'ADVANCED TOOLS' panel. It displays a message: 'There are new changes saved during your editing as follows:'. Below this is a JSON definition snippet for the 'Identification' field, highlighted in green. A red arrow points from this JSON to the bottom right window. A 'Merge & Update' button is visible.
- Bottom Left Window:** Another instance of the 'FORM BUILDER' interface. The 'DESIGN FORM' tab is active. The form preview shows a 'Country' field highlighted with a red box. A red arrow points from this box to the JSON definition.
- Bottom Right Window:** A third instance of the 'FORM BUILDER' interface. The 'DESIGN FORM' tab is active. The form preview shows both 'Identification' and 'Country' fields highlighted with red boxes. A red arrow points from the 'Identification' field to the JSON definition.

At the bottom of the 'ADVANCED TOOLS' panel, a message states: 'Your changes will be automatically merged into the following JSON definition merged changes directly.'

# 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.