**Phase 8 – Data Management & Deployment**

# Purpose

Ensure data can be imported, exported, and moved safely between environments in Salesforce. This phase covers data tools, backup strategies, deployment methods, and packaging concepts.

# Quick Prerequisites

1. Admin access in both sandbox and production orgs.  
2. Install Salesforce CLI (sfdx), VS Code + Salesforce Extension Pack, Data Loader, and ANT Migration Tool.  
3. Confirm API access is enabled on target org.  
4. Decide deployment route (Change Sets or ANT/SFDX).

# Step 0 – Backup (Data Export & Backup)

• Setup → Data Management → Data Export → Export Now or Schedule Export.  
• Include attachments/files if needed.  
• Store encrypted copies securely.  
• Alternative: use Data Loader export or third-party backup tools.

# Step 1 – Duplicate Rules & Matching Rules

• Setup → Matching Rules → Create/Enable rules (e.g., match by Email).  
• Setup → Duplicate Rules → Configure object rules (Allow/Block/Report).  
• Test with sample imports.  
Tip: For large migrations, sometimes set to Report first.

# Step 2 – Data Import Wizard

• App Launcher → Data Import Wizard.  
• Supports up to 50,000 records.  
• Choose object (standard/custom), operation (Insert/Update/Upsert).  
• Upload CSV, map fields, configure dedupe, run import.  
• Download success/error logs and reprocess errors.

# Step 3 – Data Loader (Bulk/Automated)

• Install Data Loader.  
• Login using OAuth or username+password+token.  
• Choose Insert/Update/Upsert/Delete.  
• For Upsert: create/use External ID fields.  
• Supports Bulk API for large files; configure batch size.  
• Automation possible via CLI + process-conf.xml.

# Step 4 – Change Sets (Metadata Deployment)

• Sandbox → Setup → Outbound Change Sets → New.  
• Add components (Flows, Apex, Layouts, etc.).  
• Upload to production → Inbound Change Sets → Validate → Deploy.  
• Requires deployment connection.  
• Limitation: Not all metadata types supported.

# Step 5 – ANT Migration Tool & SFDX/VS Code

\*\*ANT Migration Tool:\*\*

1. Install Java + Apache Ant.  
2. Download Force.com Migration Tool.  
3. Configure build.properties (username, password+token).  
4. Run commands: `ant retrieve`, `ant deploy`.  
5. Use destructiveChanges.xml for deletions.

\*\*SFDX + VS Code:\*\*

1. Install Salesforce CLI + VS Code extensions.  
2. Authorize org: `sfdx auth:web:login -a MySandbox`.  
3. Retrieve metadata: `sfdx force:source:retrieve -m ApexClass:MyClass`.  
4. Deploy: `sfdx force:source:deploy -p force-app/main/default -u ProductionOrg`.  
5. Supports scratch orgs, CI/CD, 2GP/unlocked packaging.

# Step 6 – Unmanaged vs Managed Packages

• Unmanaged: For internal use, code visible, not upgradeable.  
• Managed: Namespace, upgradeable, secure, AppExchange distribution.  
• Use unmanaged for internal migrations, managed for commercial apps.

# Step 7 – Example End-to-End Flow

1. Backup production with Data Export.  
2. Set Duplicate Rules to Report mode.  
3. Upload sample via Data Import Wizard.  
4. Bulk load via Data Loader (with External IDs).  
5. Deploy metadata via Change Sets or SFDX/ANT.  
6. Post-deploy: import final data and validate.  
7. Document rollback plan.

# Step 8 – Rollback & Verification

• Metadata rollback: redeploy last known-good package.  
• Data rollback: re-import from Data Export backup.  
• Validate Change Sets/Deploys before final production deploy.

# Deployment Checklist

☑ Full data export completed  
☑ Duplicate Rules tested  
☑ Sample import verified  
☑ Metadata validated in production  
☑ Data migration plan ready  
☑ Stakeholders notified  
☑ Rollback plan documented

# Best Practices & Tips

• Always test imports and deployments in a sandbox before production.  
• Use External IDs wherever possible to maintain data integrity.  
• Maintain naming conventions for metadata (flows, fields, packages).  
• Document every change and maintain in version control.  
• Always validate change sets and deployments before applying to production.  
• Schedule backups before every major release.  
• Communicate downtime windows and rollback plans clearly to stakeholders.

# Example – Data Loader CSV Template (Task Import)

Below is a sample CSV structure for importing Tasks using Data Loader:  
  
Task\_External\_Id\_\_c,Subject,OwnerEmail,WhatExternalId,DueDate  
T001,Follow up with client,user1@company.com,ACC1001,2025-10-01  
T002,Send proposal,user2@company.com,ACC1002,2025-10-02  
T003,Schedule demo,user3@company.com,ACC1003,2025-10-03  
  
• Task\_External\_Id\_\_c is a custom External ID field.  
• WhatExternalId links the Task to the parent Account using the Account's External ID.  
• OwnerEmail can be mapped to the Task Owner (if Data Loader supports email-to-Id mapping).

# Example – ANT Migration Tool Files

build.properties:  
sf.username=admin@company.com  
sf.password=Password123TOKEN  
sf.serverurl=https://login.salesforce.com  
  
package.xml (basic):  
<?xml version="1.0" encoding="UTF-8"?>  
<Package xmlns="http://soap.sforce.com/2006/04/metadata">  
 <types>  
 <members>\*</members>  
 <name>ApexClass</name>  
 </types>  
 <version>59.0</version>  
</Package>

# Advanced SFDX Usage

• Create a new project: `sfdx force:project:create -n MyProject`  
• Create a scratch org: `sfdx force:org:create -s -f config/project-scratch-def.json -a Scratch1`  
• Push metadata to scratch org: `sfdx force:source:push`  
• Run Apex tests: `sfdx force:apex:test:run -u Scratch1 -c -r human`  
• Retrieve metadata from org: `sfdx force:source:retrieve -m CustomObject:MyObject\_\_c`  
• Convert source to metadata API format: `sfdx force:source:convert -d mdapi\_output\_dir`