



Phase 8: Data Management & Deployment

⌚ Objective

The goal of this phase is to **manage data effectively** and **deploy Salesforce components safely** between environments.

This includes **data import/export, duplication control, and deployment automation** using Change Sets, Packages, or VS Code (SFDCX).

📤 1. Data Import Wizard

Description:

Data Import Wizard is a **browser-based tool** in Salesforce that helps you **import simple data** like Leads, Accounts, Contacts, or Custom Objects.

Use Case:

Used to upload **Customer** or **Complaint** records from Excel/CSV files.

Steps:

1. Go to **Setup → Data Import Wizard**.
2. Click **Launch Wizard**.
3. Choose the **Object** (e.g., Customer or Complaint).
4. Upload a **.CSV file**.
5. Map fields → Start Import.

Example:

Importing a CSV file with columns:

Customer_Name, Email, Phone, Address

After import, records automatically appear in **Customer__c**.

⚙️ 2. Data Loader

Description:

Data Loader is a **desktop application** used for **large data volumes** and **complex operations** (insert, update, upsert, delete, export).

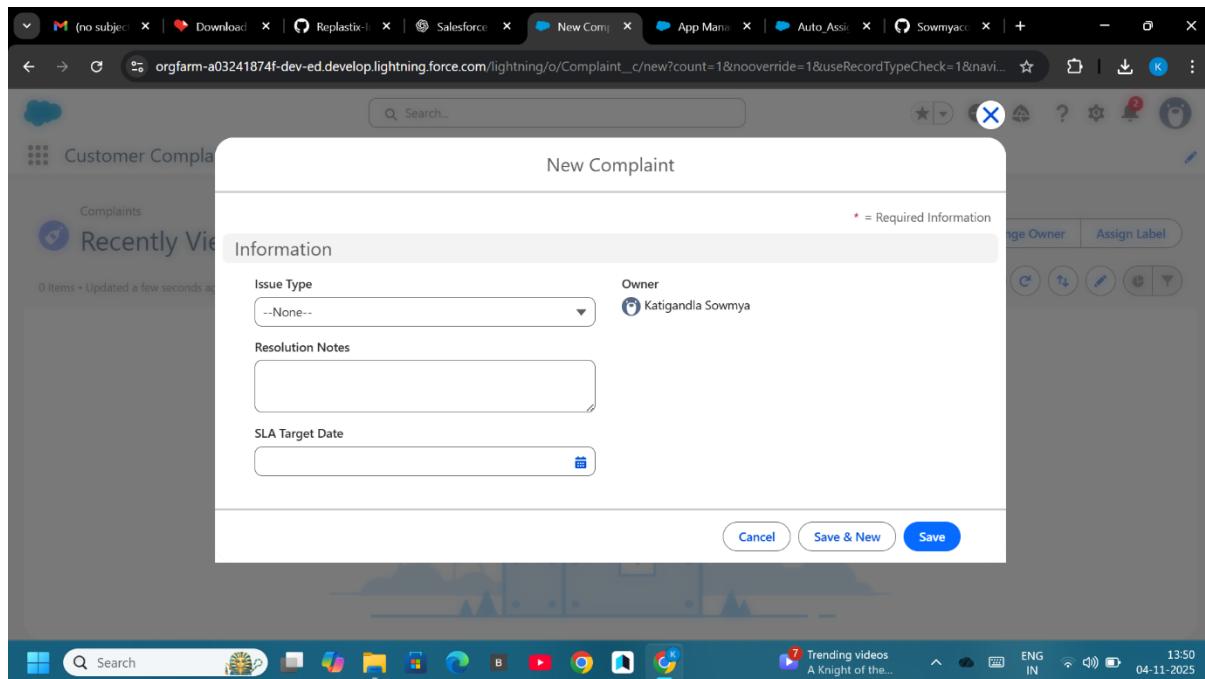
Use Case:

Used to import/export thousands of complaint records or migrate data between orgs.

Steps:

1. Download **Data Loader** from **Setup → Data Loader**.
2. Login with Salesforce credentials.
3. Choose operation (Insert/Update/Export/Delete).
4. Select Object (e.g., Complaint__c).

5. Upload CSV file and map fields.
6. Execute → Review success and error logs.



3. Duplicate Rules

Description:

Duplicate Rules prevent creating records with the same key values (like same Email or Phone).

Use Case:

Avoid duplicate customers in CRM.

Steps:

1. Setup → **Duplicate Rules** → New Rule.
2. Object: **Customer__c**.
3. Matching Rule: Email = Email.
4. Action: **Block or Alert** user when duplicate found.
5. Activate Rule.

4. Data Export & Backup

Description:

Salesforce allows **manual and scheduled backups** of data for recovery or migration.

Steps:

1. Setup → **Data Export**.
2. Click **Export Now** or **Schedule Export** (weekly/monthly).
3. Select Objects (Customer, Complaint, etc.).

4. Salesforce emails a **ZIP file** containing CSV data backup.

Best Practice:

Schedule weekly exports for safety.

5. Change Sets

Description:

Change Sets are the **standard Salesforce deployment tool** used to move metadata (fields, objects, flows, validation rules, etc.) between related orgs (e.g., Sandbox → Production).

Steps:

1. Go to **Setup** → **Outbound Change Sets** → **New**.
2. Add Components (Objects, Fields, Flows, Tabs, etc.).
3. Upload to Target Org (e.g., Production).
4. In Target Org → **Inbound Change Sets** → **Deploy**.

Use Case:

Deploy the **Customer Complaint CRM** app from Developer Org to Sandbox or Production.

6. Managed vs Unmanaged Packages

Description:

Type	Purpose	Usage
Managed Package	Locked, versioned, upgradable (AppExchange apps)	For production apps
Unmanaged Package	Editable, for learning and sharing	For student or project sharing

Use Case:

For this project, use an **Unmanaged Package** to share your app with your mentor or classmates.

Steps:

1. **Setup** → **Packages** → **New Package**.
2. Add components: Objects, Tabs, Reports, Dashboards, Flows.
3. Upload → Salesforce generates a link to share.

7. ANT Migration Tool

Description:

ANT (Apache Ant) is a **command-line deployment tool** used for automating metadata deployments between orgs.

Use Case:

Used in professional environments for CI/CD automation (Continuous Integration & Delivery).

Steps:

1. Install **Salesforce ANT Migration Tool**.
2. Create **build.xml** and **package.xml** files.
3. Run command:
4. ant retrieve
5. ant deploy
6. ANT retrieves or deploys components between orgs using API credentials.

8. VS Code & Salesforce DX (SFDX)

Description:

Salesforce DX (Developer Experience) with **Visual Studio Code** is a modern development approach for managing metadata and code.

Features:

- Source-driven development
- Version control (Git)
- Fast deployment to scratch orgs or sandboxes

Steps:

1. Install **VS Code** and **Salesforce Extensions Pack**.
2. Authenticate Org:
3. sfdx auth:web:login -d -a DevHub
4. Create Project:
5. sfdx force:project:create -n ComplaintCRM
6. Retrieve metadata:
7. sfdx force:source:retrieve -m ApexClass, CustomObject
8. Deploy changes:
9. sfdx force:source:deploy -p force-app

Use Case:

Move Apex classes, Flows, and LWC from local project to Salesforce org quickly.

Outcome

After completing Phase 8:

- Data is **accurate, secure, and backed up**.
- The **Customer Complaint CRM** can be **deployed or shared** easily between orgs.
- The project follows **professional DevOps and deployment standards**, ready for real-time implementation.