

Phase 7 – Integration & External Access

1. Named Credentials

Use Case: Store authentication details securely for external systems to simplify callouts.

Explanation: Named Credentials let you define the endpoint URL and authentication in one place, so Apex or Flows can make callouts without storing credentials in code.

Steps:

1. Setup → Named Credentials → New Named Credential
2. Enter Name, URL of external service
3. Choose Authentication (e.g., Password, OAuth)
4. Save and use in Apex or External Services

Screenshot Placeholder: Named Credential setup page.

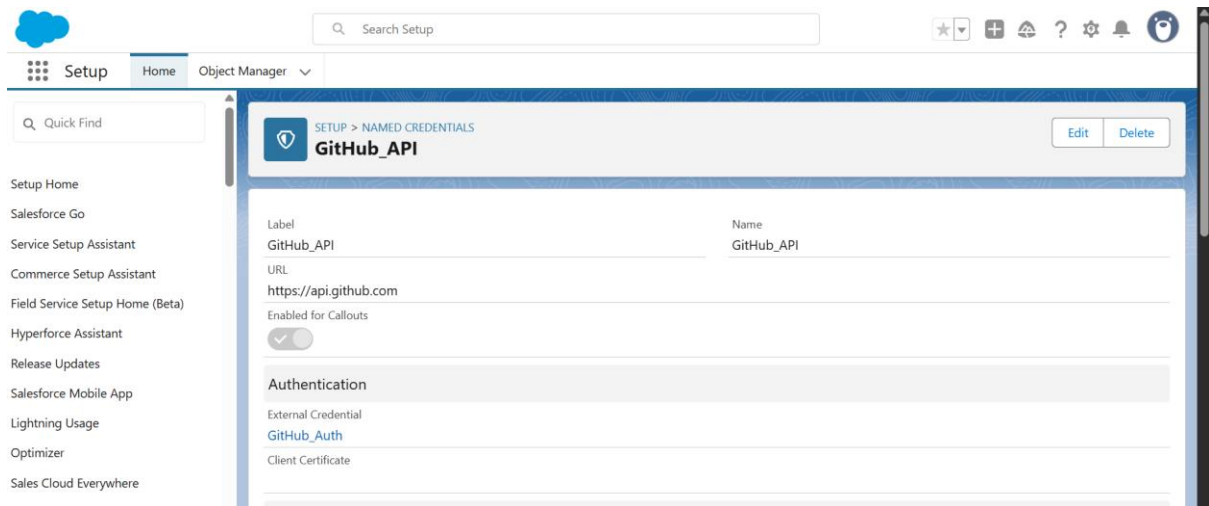
The image displays two screenshots of the Salesforce Setup interface, specifically the Named Credentials configuration page. The top screenshot shows the configuration for a Named Credential named 'GitHub_External_Cred'. The bottom screenshot shows the configuration for a Named Credential named 'GitHub_Auth'.

Top Screenshot: GitHub_External_Cred

- Label:** GitHub_External_Cred
- Name:** GitHub_External_Cred
- Description:**
- Created By Namespace:**
- Authentication:**
 - Authentication Protocol:** OAuth 2.0
 - Authentication Flow Type:** Authorization Code (Browser Flow)
- Client Identification:**
 - Client ID:**
 - Client Secret:**

Bottom Screenshot: GitHub_Auth

- Label:** GitHub_Auth
- Name:** GitHub_Auth
- Authentication Protocol:** OAuth 2.0
- Authentication Flow Type:** Browser Flow
- Scope:** read:user repo
- Identity Provider:**
 - External Auth Identity Provider:** GitHub_External_Cred
- Callout Options:**



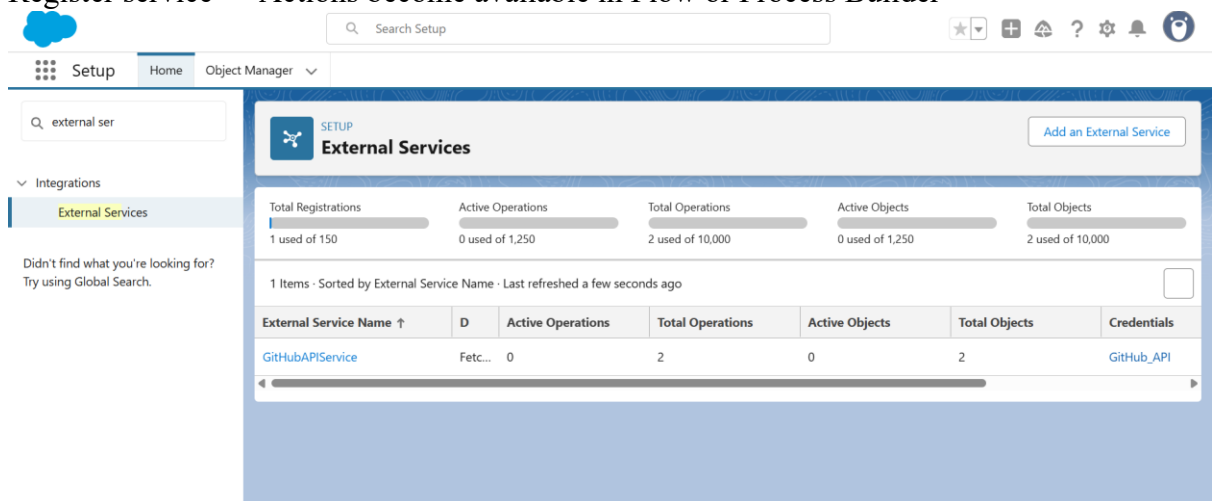
2. External Services

Use Case: Integrate ShipmentTrack with external APIs (e.g., courier tracking system).

Explanation: External Services allow Salesforce to invoke REST endpoints declaratively via Flow.

Steps:

1. Setup → External Services → New External Service
2. Provide Named Credential
3. Register service → Actions become available in Flow or Process Builder



3. Web Services (REST/SOAP)

Use Case: Enable external systems to query or update ShipmentTrack data.

Explanation: Expose Apex classes as REST or SOAP web services.

Example (REST Apex): `@RestResource(urlMapping='/ShipmentAPI/*')`

```
global with sharing class ShipmentRestService {  
    @HttpGet  
    global static Shipment__c getShipment(Id shipmentId){  
        return [SELECT Id, Name, Status__c FROM Shipment__c WHERE Id = :shipmentId];  
    }  
}
```

4. Callouts

Use Case: Fetch live shipment tracking info from courier APIs.

Explanation: Salesforce performs HTTP callouts to external endpoints to send or retrieve data.

Steps:

1. Ensure Remote Site Settings or Named Credential exist
2. Use HTTP classes in Apex to make callouts:

```
HttpRequest req = new HttpRequest();  
req.setEndpoint('https://api.courier.com/track/123');  
req.setMethod('GET');  
HttpResponse res = new Http().send(req);  
System.debug(res.getBody());
```

5. Platform Events

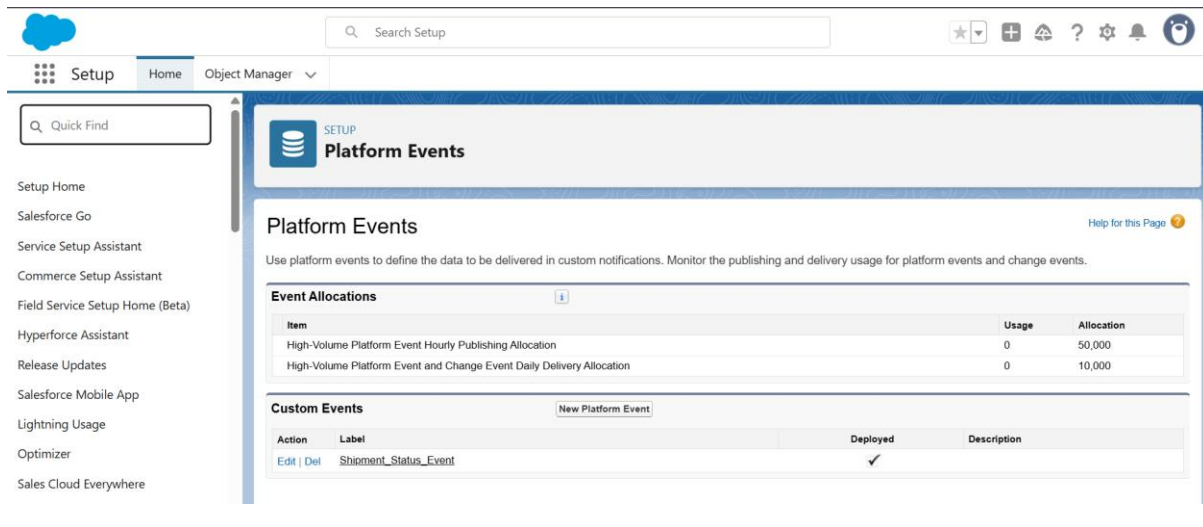
Use Case: Notify internal Salesforce users or systems when shipment status changes externally.

Explanation: Publish/subscribe model to push events in real time.

Steps:

Create a Platform Event

1. Go to Setup → Platform Events → New Platform Event
2. Name it: Shipment_Status_Event
3. Set Publish Behavior: Publish Immediately
4. Define fields: ShipmentId, Status, Timestamp
5. Subscribe via Trigger or Flow to update records



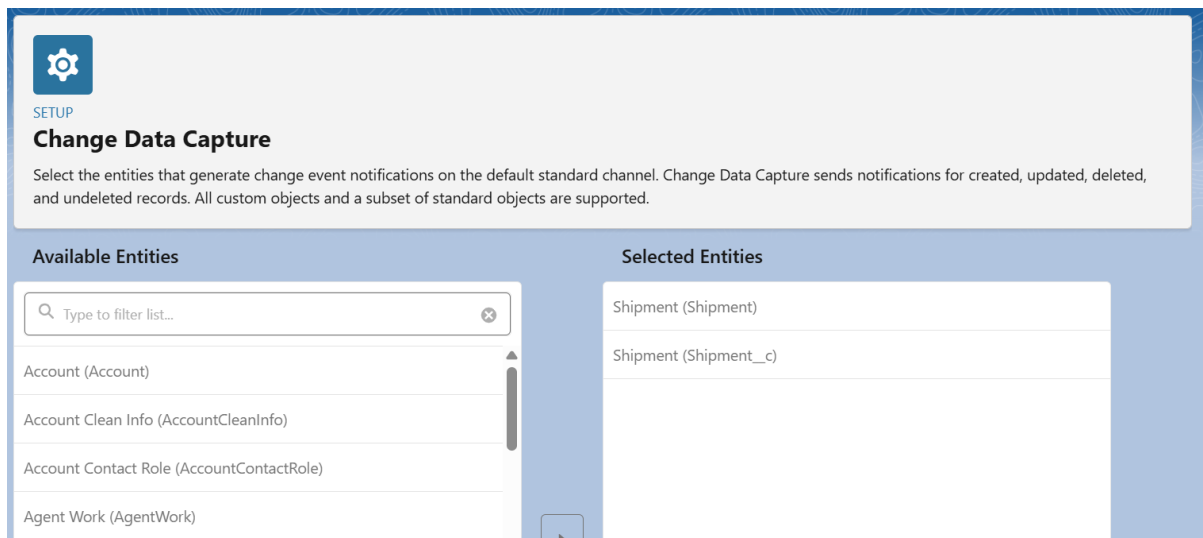
6. Change Data Capture (CDC)

Use Case: Track changes in Shipment object for external systems.

Explanation: Salesforce automatically publishes events whenever a record changes. Useful for integrations needing real-time updates.

Steps:

1. Setup → Change Data Capture → Enable for Shipment__c
2. Subscribe via CometD, Flow, or Apex Trigger



7. Salesforce Connect

Use Case: Access external shipment inventory without storing in Salesforce.

Explanation: Use External Objects to map external databases or APIs as if they were native objects.

Steps:

1. Setup → External Data Sources → New → Type (OData/Custom)

2. Configure connection → Sync external object → Add tab if needed

The screenshot shows the 'External Data Sources' setup page in Salesforce. At the top, there's a 'SETUP' button and the title 'External Data Sources'. Below this, a subtitle reads 'Access data in other Salesforce orgs as well as third-party databases and content systems.' There's a 'View:' dropdown set to 'All' and a 'Create New View' link. A navigation bar contains letters A through Z and 'Other', with 'All' selected. Below the navigation bar, there are two tabs: 'New External Data Source' (active) and 'Connect to Amazon DynamoDB'. A table lists existing data sources with columns: Action, Name, External Data Source, Type, and URL. One entry is visible: 'ShipmentInventory' with type 'Salesforce Connect: OData 4.0' and URL 'https://services.odata.org/V4/Northwind/Northwind.svc/'.

Action	Name	External Data Source	Type	URL
Edit Del	ShipmentInventory	ShipmentInventory	Salesforce Connect: OData 4.0	https://services.odata.org/V4/Northwind/Northwind.svc/

8. API Limits

Use Case: Monitor limits to prevent hitting governor limits during integration.

Explanation: Salesforce enforces daily and concurrent API limits per org.

Steps:

1. Setup → System Overview → Check API Usage
2. Track callouts, batch jobs, and integration limits

The screenshot shows the 'System Overview' page in Salesforce. At the top, there's a 'SETUP' button and the title 'System Overview'. Below this, a subtitle reads 'View key usage data for your org.' There's a 'Configure Messages' button. The page is divided into two main sections: 'Schema' and 'API Usage'. The 'Schema' section shows three rows of usage data: 'YOUR CUSTOM OBJECTS + YOUR CUSTOM SETTINGS' (9, 2% of 400), 'TOTAL CUSTOM OBJECTS + TOTAL CUSTOM SETTINGS' (9, 0% of 3,000), and 'YOUR CUSTOM METADATA TYPES' (0, 0% of 200). The 'API Usage' section shows 'API REQUESTS, LAST 24 HOURS' (0, 0% of 15,000).

Schema	API Usage
YOUR CUSTOM OBJECTS + YOUR CUSTOM SETTINGS 9 2% (maximum 400)	API REQUESTS, LAST 24 HOURS 0 0% (maximum 15,000)
TOTAL CUSTOM OBJECTS + TOTAL CUSTOM SETTINGS 9 0% (maximum 3,000)	
YOUR CUSTOM METADATA TYPES 0 0% (maximum 200)	

9. OAuth & Authentication

Use Case: Securely connect Salesforce to external services or allow apps to access Salesforce data.

Explanation: OAuth 2.0 provides token-based authentication without exposing passwords.

Steps:

1. Setup → App Manager → New Connected App
 2. Enable OAuth settings → Select scopes → Save
 3. Use Named Credential or custom code for authentication
-

10. Remote Site Settings

Use Case: Allow Salesforce to call external endpoints via HTTP callouts.

Explanation: Any URL that Salesforce calls must be registered in Remote Site Settings for security reasons.

Steps:

1. Setup → Remote Site Settings → New Remote Site
2. Enter Name, URL of external system → Save

The screenshot displays the Salesforce Setup interface. The top navigation bar includes the Setup icon, a search bar, and various utility icons. The left sidebar shows the Setup menu with options like Home, Object Manager, and a Quick Find search bar. The main content area is titled 'Remote Site Settings' and shows the details for a specific remote site named 'ShipmentAPI'. The details include the Remote Site URL 'https://api.shipmenttracker.com', a checkbox for 'Disable Protocol Security' which is unchecked, a description field, an 'Active' checkbox which is checked, and the 'Created By' field showing 'Gauni.Ladykar' on '9/29/2025, 8:57 PM'. There are 'Edit', 'Delete', and 'Clone' buttons for both the header and the details section. A 'Help for this Page' link is also visible.

Remote Site Detail		Edit	Delete	Clone
Remote Site Name	ShipmentAPI	Modified By Gauni.Ladykar, 9/29/2025, 8:57 PM		
Remote Site URL	https://api.shipmenttracker.com			
Disable Protocol Security	<input type="checkbox"/>			
Description				
Active	<input checked="" type="checkbox"/>			
Created By	Gauni.Ladykar, 9/29/2025, 8:57 PM	Edit	Delete	Clone