

Phase 7: Integration & External Access

1. Remote Site Settings

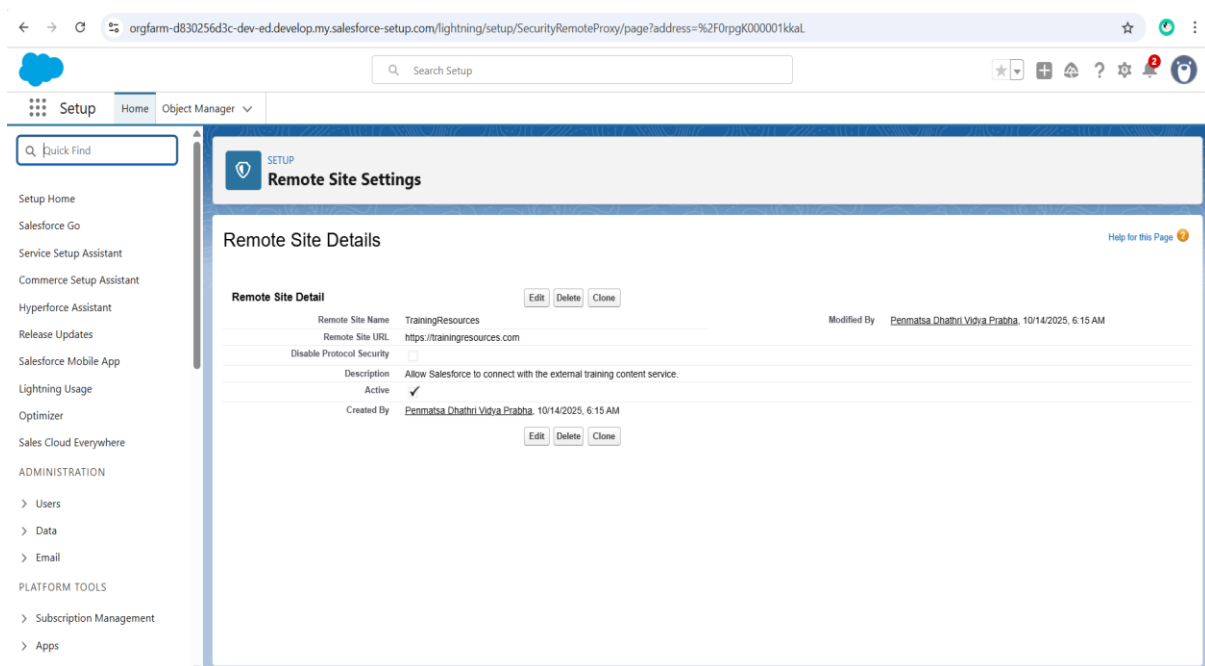
Use Case:

Our Corporate Training CRM app needs to fetch training resources or course materials from an external learning website (e.g., <https://trainingresources.com>).

To allow Salesforce to communicate with that external site securely, we must add it to Remote Site Settings.

Implementation Steps:

1. Go to Setup → Security → Remote Site Settings.
2. Click New Remote Site.
3. Fill:
 - Remote Site Name: TrainingResources
 - Remote Site URL: <https://trainingresources.com>
 - Description: Allow Salesforce to connect with the external training content service.
4. Click Save.



2. Named Credentials

Use Case:

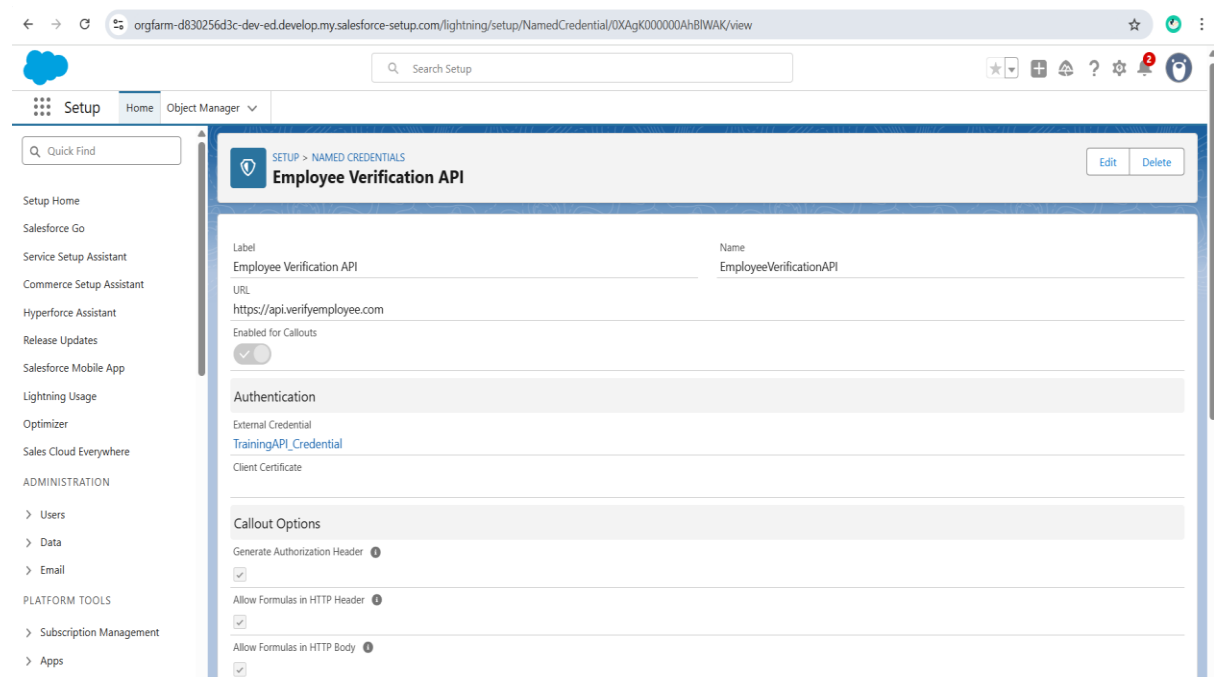
Instead of hardcoding credentials in Apex, we securely store API details for an external system (like a REST API for employee data verification).

Named Credentials simplify authentication and API callouts.

Implementation Steps:

1. Go to Setup → Named Credentials → New Named Credential.
2. Fill:
 - Label: Employee Verification API
 - Name: EmployeeVerificationAPI
 - URL: <https://api.verifyemployee.com>
 - External Credential: [TrainingAPI_Credential](#)
 - Identity Type: Named Principal
 - Authentication Protocol: Password Authentication or OAuth (if available).

3. Save.



The screenshot shows the Salesforce Setup interface for configuring a Named Credential. The browser address bar displays the URL: `orgfarm-d830256d3c-dev-ed.develop.my.salesforce-setup.com/lightning/setup/NamedCredential/0XAgK000000AhBIWAK/view`. The page title is "SETUP > NAMED CREDENTIALS" and the specific credential is "Employee Verification API".

The configuration details are as follows:

- Label:** Employee Verification API
- Name:** EmployeeVerificationAPI
- URL:** <https://api.verifyemployee.com>
- Enabled for Callouts:** ☒
- Authentication:**
 - External Credential:** [TrainingAPI_Credential](#)
 - Client Certificate:** (empty)
- Callout Options:**
 - Generate Authorization Header:** ☒
 - Allow Formulas in HTTP Header:** ☒
 - Allow Formulas in HTTP Body:** ☒

3. Apex REST Callout

Use Case:

When a registration record is created, the system calls an external service to check whether the employee's email is valid (example API endpoint).

Implementation:

Apex Class

```
public with sharing class EmployeeEmailValidator {  
    @future(callout=true)  
    public static void validateEmail(String email) {  
        Http http = new Http();  
        HttpRequest req = new HttpRequest();  
  
        req.setEndpoint('callout:EmployeeVerificationAPI/validate?email=' +  
            email);  
  
        req.setMethod('GET');  
        HttpResponse res = http.send(req);  
        if (res.getStatusCode() == 200) {  
            System.debug('Validation success: ' + res.getBody());  
        } else {  
            System.debug('Validation failed: ' + res.getStatus());  
        }  
    }  
}
```

Apex Trigger:

```
trigger RegistrationTrigger on Registration__c (after insert) {
```

```

for (Registration__c reg : Trigger.new) {

EmployeeEmailValidator.validateEmail(reg.Employee__r.Email__c);

}

}

```

The screenshot shows the Salesforce Developer Console with the `EmployeeEmailValidator.apex` class open. The class contains a static method `validateEmail` that makes an HTTP GET request to a custom endpoint. Below the code editor, the **Logs** tab is active, displaying a table of log entries.

User	Application	Operation	Time	Status	Read	Size
Permatasa Dhatthri Vidya Prabha	Unknown	ApexTestHandler	10/14/2025, 7:02:43 PM	Success	Unread	2.22 KB
Permatasa Dhatthri Vidya Prabha	Unknown	ApexTestHandler	10/14/2025, 7:02:43 PM	Success	Unread	528 bytes
Permatasa Dhatthri Vidya Prabha	Unknown	ApexTestHandler	10/14/2025, 7:02:09 PM	Success	Unread	528 bytes
Permatasa Dhatthri Vidya Prabha	Unknown	ApexTestHandler	10/14/2025, 7:02:09 PM	Success	Unread	2.22 KB

The screenshot shows the Salesforce Developer Console with the `RegistrationsTrigger.apex` class open. The class is a trigger that calls the `validateEmail` method from the `EmployeeEmailValidator` class. Below the code editor, the **Logs** tab is active, displaying a table of log entries.

User	Application	Operation	Time	Status	Read	Size
Permatasa Dhatthri Vidya Prabha	Unknown	ApexTestHandler	10/14/2025, 7:03:43 PM	Success	Unread	2.22 KB
Permatasa Dhatthri Vidya Prabha	Unknown	ApexTestHandler	10/14/2025, 7:03:43 PM	Success	Unread	528 bytes
Permatasa Dhatthri Vidya Prabha	Unknown	ApexTestHandler	10/14/2025, 7:02:09 PM	Success	Unread	528 bytes
Permatasa Dhatthri Vidya Prabha	Unknown	ApexTestHandler	10/14/2025, 7:02:09 PM	Success	Unread	2.22 KB

4. Platform Events

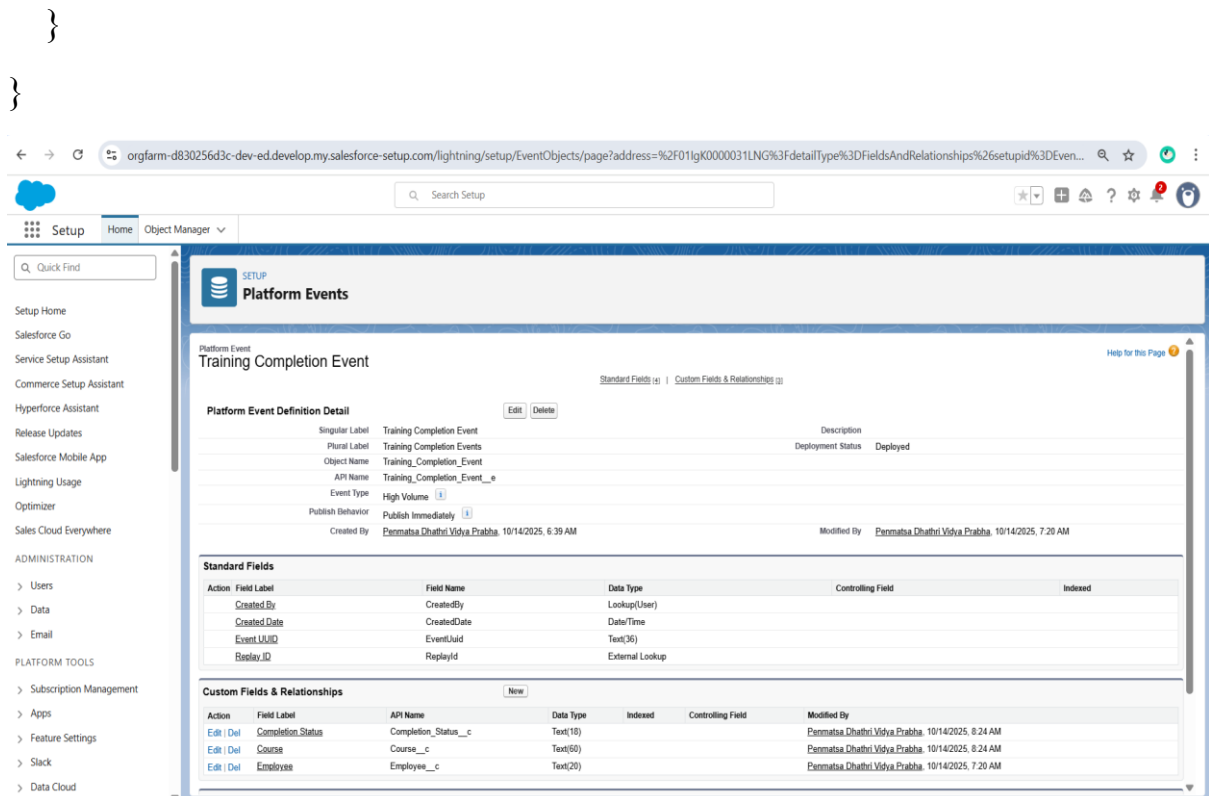
Use Case:

When a new registration is completed, send a **Platform Event** notification to external systems or apps (like Slack or an internal dashboard).

Implementation Steps:

1. Go to **Setup** → **Platform Events** → **New Platform Event**.
2. Name it `Training_Completion_Event`.
3. Add fields:
 - **Employee__c** (Text)
 - **Course__c** (Text)
 - **CompletionStatus__c** (Text)
4. Publish the event from Apex:

```
trigger RegistrationTrigger on Registration__c (after update) {  
    for (Registration__c reg : Trigger.new) {  
        // Check if completion is 100% and status is completed  
        if (reg.Completion__c == 100 && reg.Status__c != 'Completed') {  
            Training_Completion_Event__e eventMsg = new  
Training_Completion_Event__e(  
                Employee__c = reg.Employee__r.Name,  
                Course__c = reg.Course__r.Course_Name__c,  
                CompletionStatus__c = 'Completed'  
            );  
            Database.SaveResult sr = EventBus.publish(eventMsg);  
            System.debug('Event published: ' + sr.isSuccess());  
        }  
    }  
}
```



5. Change Data Capture

Use Case:

You want external apps to automatically get notified when a **Registration** record is updated or completed (for example, to sync with an external reporting system).

Implementation Steps:

1. Go to **Setup** → **Change Data Capture**.
2. Enable **Registration__c**, **Course__c**, and **Employee__c** objects.
3. Now Salesforce automatically streams events whenever these records change.

← → ↺

orgfarm-d830256d3c-dev-ed.develop.my.salesforce-setup.com/lightning/setup/CdcObjectEnablement/home

🔍 ☆ 🟢 ⋮

☁️

🔍 Search Setup

☆ 📄 🏠 ? ⚙️ 🔔 👤

⋮ Setup

Home

Object Manager ▾

🔍 Quick Find

Setup Home

Salesforce Go

Service Setup Assistant

Commerce Setup Assistant

Hyperforce Assistant

Release Updates

Salesforce Mobile App

Lightning Usage

Optimizer

Sales Cloud Everywhere

ADMINISTRATION

> Users

> Data

> Email

PLATFORM TOOLS

> Subscription Management

> Apps

> Feature Settings

> Slack

> Data Cloud

⚙️

SETUP

Change Data Capture

Select the entities that generate change event notifications on the default standard channel. Change Data Capture sends notifications for created, updated, deleted, and undeleted records. All custom objects and a subset of standard objects are supported.

Available Entities

🔍 Type to filter list...

Account (Account)

Account Clean Info (AccountCleanInfo)

Account Contact Role (AccountContactRole)

Agent Work (AgentWork)

ApiPrtcPolicy

Asset (Asset)

Asset Relationship (AssetRelationship)

Assigned Resource (AssignedResource)

Associated Location (AssociatedLocation)

Authorization Form (AuthorizationForm)

➡

⬅

Selected Entities

Course (Course_c)

Registration (Registration_c)

Employee (Employee_c)

Cancel

Save