**Phase 7 : Integration & External Access**

**Objective**

The goal of Phase 7 is to integrate Salesforce with the external **Learning Management System (LMS)** to achieve seamless synchronization of training assignments and completion records. This phase also ensures secure, real-time communication between systems through APIs, callouts, and platform events.

📸 *Insert Screenshot: Salesforce Setup Home (highlighting Integration section)*

**1. Named Credential**

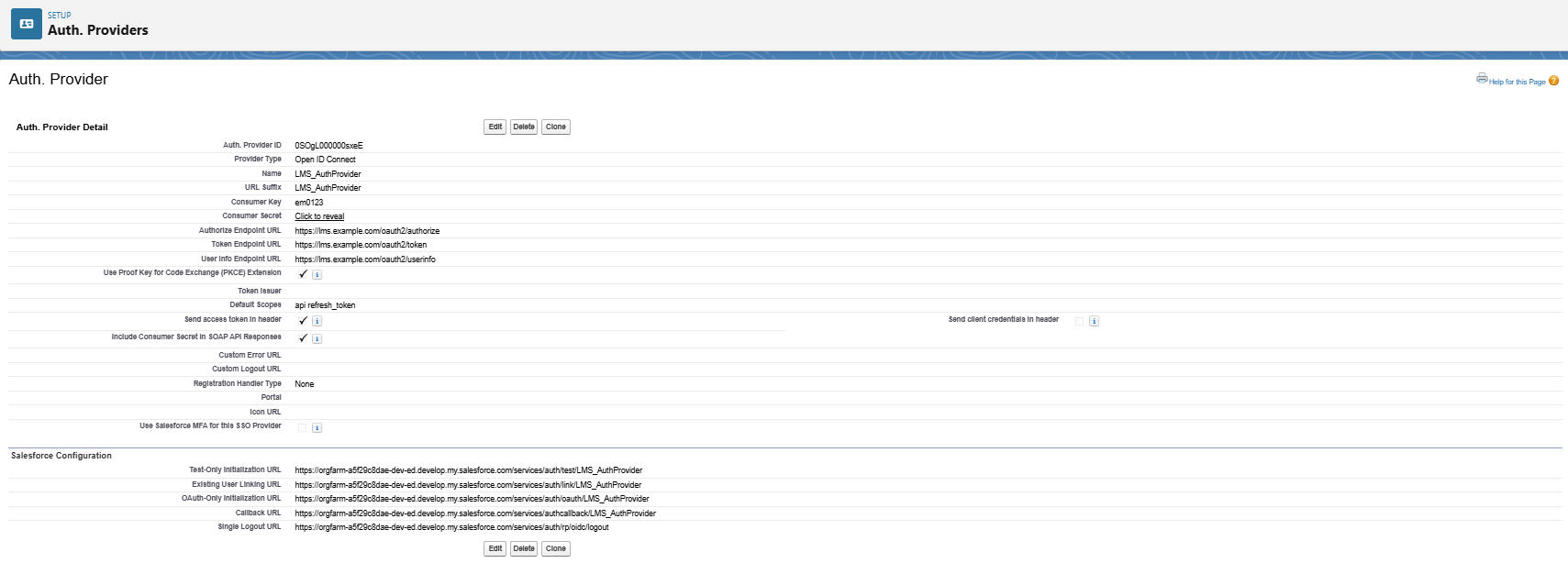
Setup → *Named Credentials* → New

* Label: **LMS\_NC**
* URL: https://lms.example.com/
* Identity Type: Named Principal
* Authentication: OAuth 2.0 or Password

**2. Auth Provider (if OAuth2)**

Setup → *Auth Providers* → New

* Provider Type: **OpenID Connect**
* Name: **LMS\_AuthProvider**
* Authorize & Token Endpoint URLs: provided by LMS



**3. Apex Class: Outbound Callout**

Apex Queueable class pushes new Assignments to LMS.

public with sharing class LMSPushJob implements Queueable, Database.AllowsCallouts {

private Id assignmentId;

public LMSPushJob(Id assignmentId){ this.assignmentId = assignmentId; }

public void execute(QueueableContext ctx) {

Assignment\_\_c a = [

SELECT Id, Name, Due\_Date\_\_c, Training\_Module\_\_r.External\_Module\_Id\_\_c,

Employee\_\_r.Email

FROM Assignment\_\_c WHERE Id = :assignmentId LIMIT 1

];

Map<String,Object> payload = new Map<String,Object>{

'employeeEmail' => a.Employee\_\_r.Email,

'moduleId' => a.Training\_Module\_\_r.External\_Module\_Id\_\_c,

'dueDate' => String.valueOf(a.Due\_Date\_\_c),

'assignmentId' => a.Id

};

HttpRequest req = new HttpRequest();

req.setEndpoint('callout:LMS\_NC/api/assignments');

req.setMethod('POST');

req.setHeader('Content-Type','application/json');

req.setBody(JSON.serialize(payload));

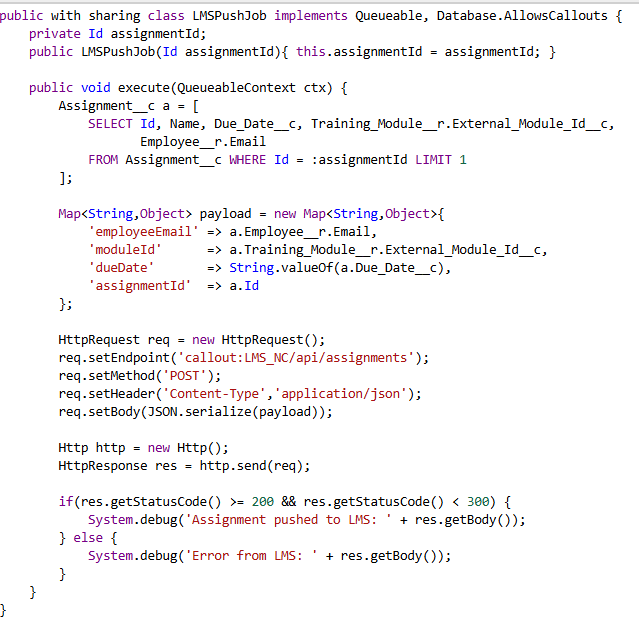
Http http = new Http();

HttpResponse res = http.send(req);

System.debug('Response: ' + res.getBody());

}

}



**4. Trigger to Enqueue Push**

trigger AssignmentTrigger on Assignment\_\_c (after insert) {

for(Assignment\_\_c a : Trigger.new) {

System.enqueueJob(new LMSPushJob(a.Id));

}

}

A screenshot of a computer

AI-generated content may be incorrect.

**5. Apex REST Service: Inbound Updates**

@RestResource(urlMapping='/lms/completion/\*')

global with sharing class LMSWebhook {

@HttpPost

global static void receiveCompletion() {

RestRequest req = RestContext.request;

Map<String,Object> payload =

(Map<String,Object>) JSON.deserializeUntyped(req.requestBody.toString());

String assignmentId = (String) payload.get('assignmentId');

Decimal score = payload.containsKey('score') ?

Decimal.valueOf(String.valueOf(payload.get('score'))) : null;

Assignment\_\_c a = [SELECT Id, Employee\_\_c FROM Assignment\_\_c

WHERE Id = :assignmentId LIMIT 1];

Completion\_\_c c = new Completion\_\_c(

Assignment\_\_c = a.Id,

Employee\_\_c = a.Employee\_\_c,

Score\_\_c = score,

Completion\_Date\_\_c = Date.today()

);

insert c;

a.Status\_\_c = 'Completed';

update a;

RestContext.response.statusCode = 200;

}

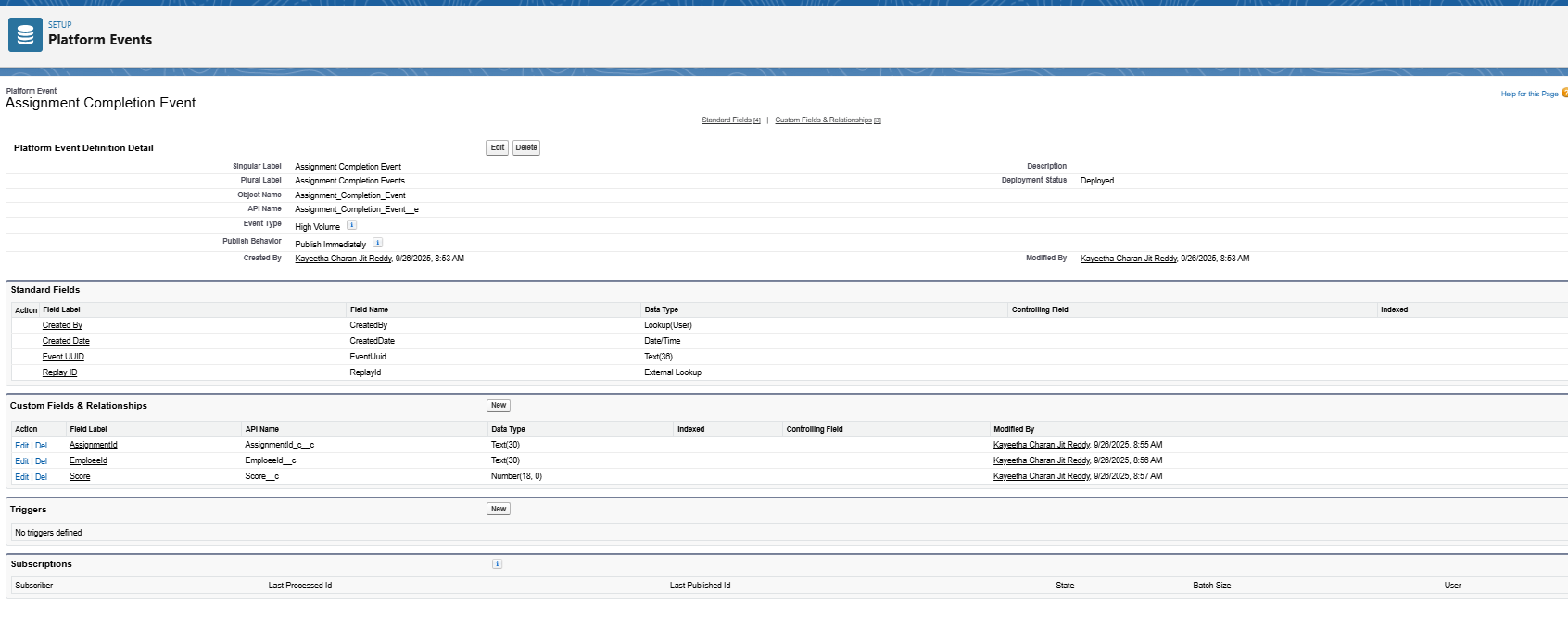
}



**6. Platform Event**

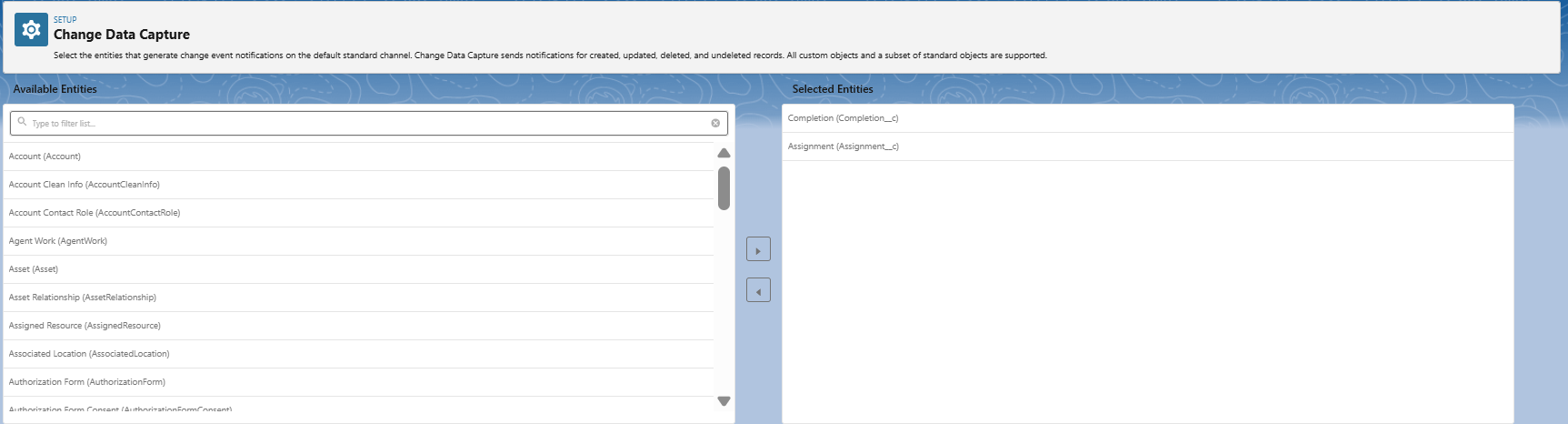
Create **AssignmentCompletion\_\_e** with fields:

* AssignmentId\_\_c (Text)
* EmployeeId\_\_c (Text)
* Score\_\_c (Number)



**7. Change Data Capture (CDC)**

Enable CDC for **Assignment\_\_c** and **Completion\_\_c** objects.



**Security Checklist**

* Use **Named Credentials** (no hardcoded URLs/keys).
* Protect REST endpoints with **OAuth2** or HMAC headers.
* Give the integration user **least privilege profile**.
* Monitor usage in **Company Information → API Requests**.

**Conclusion**

Phase 7 established secure two-way integration between Salesforce and the LMS. Assignments flow outbound, completions flow inbound, and platform events plus CDC keep systems in sync. This ensures a robust and enterprise-ready solution.